Introduction to Personal Oracle7

Personal Oracle7 for Windows 95 is a desktop version of the Oracle7 Database with a Windows 95 user interface called the Personal Oracle7 Navigator. Use Personal Oracle7 to:

- design/develop database objects, such as tables, views, indexes, etc.
- deploy database applications into any Oracle7 environment

Personal Oracle7 for Windows 95 is designed for two different user groups:

- those who have used other desktop database programs and want to upsize to a relational database with the power, flexibility, scalability, and multiuser features provided in an Oracle7 database
- developers and users who want the Oracle7 database on a desktop or laptop and who want to use distributed database features

The database in this product is a desktop version of the Oracle7 database. It has the same functionality as Oracle7 databases operating on midrange and mainframe computers, except that it cannot function as a server.

Applications developed on Personal Oracle7 can run with no reengineering on other Oracle7 platforms. Personal Oracle7 can support a maximum of 15 database connections.

This product includes the following components:

- Personal Oracle7 Navigator - This user interface allows you to browse, create, and modify database objects, such as tables, indexes, synonyms, snapshots, and views.
- Oracle7 DBMS - This version of the Oracle7 Database provides the power and sophistication of Oracle7 including advanced distributed features and symmetric replication..
- Oracle7 Utilities - This product includes the Export Utility, the Import Utility, and SQL*Loader. The Export Utility writes data from an Oracle7 database to an operating system file in order to archive, move, or restore data. The Import Utility reads data from files exported with Export into the Personal Oracle7 DBMS. SQL*Loader enables you to load data from foreign data sources - usually ASCII data files--into the Oracle7 Database.
- Oracle Call Interface (OCI) - OCI is an applications programming interface (API) for accessing an Oracle7 Database from a C program.
- SQL*Net - SQL*Net (the Oracle network interface) enables front-end application software running on client workstations to access, modify, share, and store data on Oracle7 database servers. This release of Personal Oracle7 contains SQL*Net Version 2 and includes support for the following protocols: TCP/IP, SPX, and Named Pipes.
- SQL*Plus - SQL*Plus for Windows 95 is a command language interface that enables you to use the SQL and PL/SQL database languages in a Windows 95 environment.

{button ,AL(`c_getting_started;IOC_Database_Objects_Overview;IDH_Oracle7_Navigator_Overview;IDH_Oracle7_Navigator_Tool_Bar;IDH_Oracle7_Navigator_Window')}   Related Topics

About Importing and Exporting

The Import and Export Utilities are command line programs used to import and export data from an Oracle database. Export writes data from an Oracle7 Database into a transportable operating system file. Import reads data from exported files back into an Oracle7 Database.

For information on using the Import and Export Utilities, refer to *Oracle7 Server Utilities,* Release 7.2, available online as an Adobe Acrobat-readable file.

{button ,AL(`getting_started;PRC_To_export_a_project')}   Related Topics

About SQL*Loader

The SQL*Loader Utility moves data from external files – usually ASCII text files – into tables in an Oracle database. SQL*Loader can load data from multiple datafiles of different file types. SQL*Loader is a command line program.

For information on using the SQL*Loader Utility, refer to *Oracle7 Server Utilities,* Release 7.2, available online as an Adobe Acrobat-readable file.

{button ,AL(`getting_started')} Related Topics

About SQL*DBA

The SQL*DBA Utility manages the operation and structure of an Oracle7 Database. You can perform a variety of database administration tasks with SQL*DBA. SQL*DBA is a command line program.

For information on using SQL*DBA, see the online help topics about SQL*DBA, beginning with Starting SQL*DBA. Also refer to *Oracle7 Server Utilities,* Release 7.2, available online as an Adobe Acrobat-readable file.

{button ,AL(`getting_started;SQL*DBA')} Related Topics

Getting Started with Personal Oracle7

Personal Oracle7 is a desktop relational database program that you can use to develop, create, and modify database objects, access information on remote databases, and deploy database applications into any Oracle7 environment.

Depending on what you want to do with this product, the following features may be useful:

Sample Data

Oracle has included some sample data that you can access by connecting to the local database. To do this, click Local Database on the Personal Oracle7 Navigator screen and then click a particular database object folder.

For example, click the Table folder and a list of sample tables appears. You can copy any of these tables and use the copy for your own application.

Create a Project

To group objects together for an application, you can create a Project. A Project is a way to conveniently collect database objects associated with a particular use or application.

For example, if you plan to track the products your company manufactures, you can create a Product Project. In this Project, you can create several tables related to your company's products, or you can drag and drop a table from the Local database.

You can also create a snapshot in your Project of a table on a remote database with information about a particular product. For information on how to do this, see To add a snapshot to a project. If you want to look at data from several tables, you can create a view in your Product Project. Also, within a Project, you can have a pointer to external files, such as a form associated with your application.

Within a Project, you can create, modify, and delete any database object to help you to store and organize data related to your particular plan.

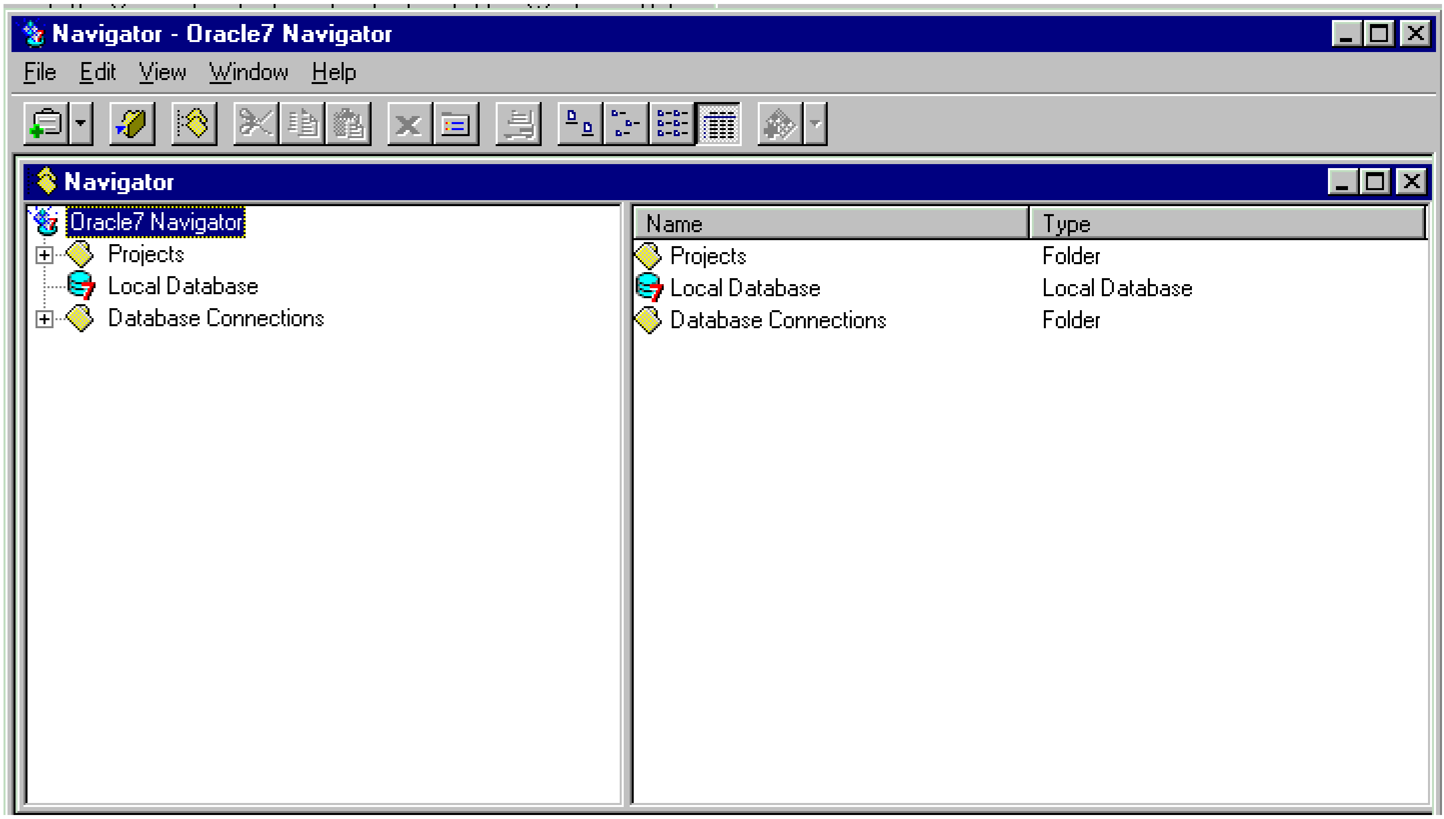
Create Database Objects

You can create new database objects such as tables, views, and indexes, or copy existing ones inside a Project or not.   By placing database objects inside a Project, you can organize the data related to your particular plan.

{button ,AL(`getting_started;r_windows;IOC_The_Oracle7_Server')}   Related Topics

About the Personal Oracle7 Navigator

The Personal Oracle7 Navigator is an application that you can use to access all your projects, database objects, and database connections. It functions much the same as the Windows 95 Explorer functions.



Name	Type
Projects	Folder
Local Database	Local Database
Database Connections	Folder

The following table shows some actions and results for the Navigator screen.

Action	**Result**
Click any object on the left side	The right side changes to reflect the contents of the selected object
Click button (+ or - sign next to an object or folder) on the left side	The left side expands or collapses to show the objects contained in the object or folder
Double-click any object on the right side	The right side changes to reflect the contents of the selected object, some default operation (like opening the object) is performed on the object
Drag an item from the right side and drop it onto an object or folder on the left side	The selected item is copied into the object or folder on the left side
Right click any object on either the left or the right side	A list appears showing the actions that can be performed on the object

Note: Not every object can be copied into an object folder. For example, a table cannot be copied into a User folder. An object cannot be copied to the Project folder, but it can be copied to a specific project that has been created. In these cases, the icon changes to a circle with a slash through it to let you know that you cannot perform this function.

You can use the Navigator to create, delete, move, view, and modify database objects. For a more complete description of what can be done with the Navigator, please refer to the online Help section, Step-by-Step Procedures.

You can also use it to view database objects by type (you can view all tables, or all snapshots) or by user (you can view all objects owned by Scott or by any other user). For help on doing this, refer to the following online Help topics: To view database objects by owner and To view database objects by type.

{button ,AL(`getting_started;r_windows')} Related Topics

About Projects

A Project is a related collection of information located on the local database. A Project can contain tables, views, indexes, etc., which are called database objects. A Project can also contain pointers to external files such as reports, spreadsheets, programs, and other applications. These database objects and nondatabase objects are what you use to store and organize the data related to your Project.

For example, if you plan to track the products your company manufactures, you can create a Product Project. In this Project, you can create several tables related to your company’s products, or you can copy a table on a remote database that has information about a particular product and paste it into your Project. After you paste it in, the object is also created in your local database. If you want to look at specific data from several tables, you can create a view in your Product Project. You can also put a form associated with the application into a Project.

Within a Project, you can create, modify, and delete any database object to store and organize the data related to your plan.

{button ,AL(`getting_started;p_projects')} Related Topics

The Word "Database"

In other desktop database programs, you can open and modify different database files containing separate collections of information.

For example, you might have a Bowling League database, with players' names and standings. Or, you might have the XYZ Company Product database, with information on the products your company manufactures, including the product name and identification number, the product cost, number of products in stock, number of products on order, and so on.

Oracle uses the word database to mean the physical and logical structure that you use to store information. You can organize many different collections of information and store them in this one database. When you install Personal Oracle7, you are given the choice of installing a database (known as the Starter database). This database consists of physical and logical structures that allow you to create, modify, and access data.

For example, within the Oracle7 database, you can create many different collections of information, known as database objects (tables, views, indexes, etc.) that refer to your bowling league, or to the XYZ company products.

Note: You can create a new database (one that is different from the Starter database) using Personal Oracle7, but Oracle does not recommend doing this unless you are an experienced Oracle user. Instead, we advise you to use the Starter database provided with this product.

{button ,AL(`dbconcepts')} Related Topics

Starting Up and Shutting Down the Database

By default, the Oracle7 DBMS starts up whenever you first access the Local Database in the Personal Oracle7 Navigator and it shuts down when you exit or log off Windows 95.

If you want to change this default action, you can:

- start the database when Windows 95 starts up, or when you log on to Windows 95
- manually start up and shut down the database
- not have the database start when you first access the Local Database in the Navigator

For information on how to perform these functions, refer the following online Help topics:

- To start the database at Windows 95 startup
- To start or stop the database manually
- To disable or enable automatic database startup

{button ,AL(`getting_started')} Related Topics

The Relational Database

A relational database uses tables to store information.

For example, you might want to track the products your company manufactures. In a relational database, you create several tables to store different pieces of information about your products, such as an inventory table, a manufacturing table, and a shipping table.

Each one of these tables includes a column for the product's part number. If several tables contain the same column with the same information, then the information in one table can refer to the information in another table.

Because the information in all of the tables is connected by this common column, you can easily find out how much product you have on hand, the product's manufacturing time, when a product shipped, and when and for what price the product was billed.

{button ,AL(`dbconcepts')}   Related Topics

The Local Database

The local database is the database that is made available to you when you start the Personal Oracle7 database and is the database you connect to when you select Local Database in the Personal Oracle7 Navigator. By default, this is the Starter database that comes with Personal Oracle7 for Windows 95, but it may also be a database that you have created. The Navigator uses a predefined user account to connect to the local database. A user account is a user name and a user password. Each user account is granted certain privileges and roles. The user name for the local database is PO7. The predefined user account has the necessary privileges to allow you to perform all database functions.

If you want to connect to your local database from another application such as Oracle Forms or SQL*Plus, you may be asked for the host string, alias, connect string, database, connect descriptor, or some similar label, depending on the application. In most cases, you can leave this field blank. This is because most applications connect to the local database by default.

Some applications, such as ODBC require that you enter something in this field. If you find that you cannot connect to the local database, enter 2: (this is the alias or connect string for the local database) in this field.

{button ,AL(`dbconcepts')}   Related Topics

Database Objects

A database object is anything defined and stored in a database. The following is a list of database objects: tables, views, indexes, synonyms, database links, roles, snapshots, and users.

{button ,AL(`dbconcepts;IOC_Database_Objects_Overview')}   Related Topics

Users, Passwords, Roles, and Privileges

Because the data you keep in your database may be confidential, Oracle has created powerful security features to ensure that unauthorized people cannot access your data. Users, passwords, roles, and privileges enable you to decide who has access to what part of your database.

When you first install Personal Oracle7 for Windows 95, certain default users and passwords are set up for your local database. By changing the default passwords, you can make your Personal Oracle7 database more secure. These defaults also have specific privileges associated with them, which allow you to perform specific database functions. Use these defaults until you establish users and passwords of your own. Please note that creating your own users and passwords helps to ensure the security of the data in your database.

User Name and User Password

A user name is a unique name by which a user is known to the Oracle7 database and to other Oracle users. A user name must be associated with a user password. The maximum length for the user name is 30 characters.

A user password is a word or phrase (no spaces or commas allowed) associated with a user name. The maximum length is 30 characters.

Privilege

A privilege is the right to execute a particular type of SQL statement or it is the right to access another user's object. You can grant privileges to individual users, or you can create a role and then grant privileges to that role. If you grant the role to one or more users, then they have the same privileges.

There are two types of privileges: system privileges and object privileges.

System privilege is the right to perform a particular action, or to perform a particular action on a type of object. An example of a system privilege is the right to delete the rows of any table in a database.

Object privilege is the right to perform a particular action on a specific object. An example of an object privilege is the right to delete rows from the EMP table.

Role

Defining a database role is a method of granting access privileges. You create a database object called a role, then grant privileges to that role, then grant that role to individual users. The Oracle7 database comes with several automatically defined roles, such as CONNECT, RESOURCE, DBA, EXP_FULL_DATABASE, and IMP_FULL_DATABASE.

Default User Names, User Passwords, and Roles

The following table shows the default user names, user passwords, and roles that exist for the local database.

Use these default values until you establish your own user name and user password. Please note that creating your own users and passwords helps to ensure the security of the data in your database.

Each role has certain database privileges. The user name, its associated password, and its assigned role is called a user account.

User Name	User Password	Roles
SCOTT	TIGER	CONNECT and RESOURCE
SYSTEM	MANAGER	DBA
SYS	SYS [Windows 95 product ID number excluding dashes]	CONNECT, RESOURCE, DBA, EXP_FULL_DATABASE, and IMP_FULL_DATABASE
DEMO	DEMO	CONNECT and RESOURCE
PO7	PO7	DBA

The Windows 95 product ID number can be found by clicking the Start button, Settings, Control Panel, then double clicking the System application. The product id number is on the General page under Registered to:

NOTE: The SYS user name gives you access to the structures that implement the database; therefore, it is possible to damage your database. Extreme care should be taken if it is ever necessary to use this user name.

The following table shows the three automatically defined roles and the privileges associated with them:

Role	Privilege
CONNECT	You can access the Oracle7 database
RESOURCE	You can access the Oracle7 database, and you can create tables, sequences, procedures, triggers, indexes, and clusters
DBA	You have all system privileges and you can grant all privileges to other users

{button ,AL(`dbconcepts;PRC_To_grant_roles_to_users;PRC_To_grant_system_privileges_to_roles;PRC_To_grant_roles_to_roles;PRC_To_grant_system_privileges_to_users;PRC_To_create_a_user;PRC_To_create_a_role;IDH_Create_Role_dialog_box;IDH_Create_User_dialog_box')} Related Topics

The Distributed Feature

This feature permits you to access data residing on other computers in different locations. The distributed feature includes:

- Database links - provide database-to-database access between the local and a remote database
- Distributed queries - allow retrieval of information from local and remote databases in one transaction
- Distributed updates - modify data in a local or remote database in one transaction
- Read-only and Updatable Snapshots - allow you to replicate data between databases
- Two-Phase commit - ensures that all database servers participating in a distributed transaction either all commit or all roll back a transaction statement.

Please refer to *Oracle7 Server Distributed Systems, Volumes I and II* (available online as an Adobe Acrobat-readable file) for additional information.

{button ,AL(`dbconcepts;IOC_About_Database_Links;PRC_To_create_a_database_link;PRC_To_create_a_database_connection;PRC_To_create_a_snapshot;IDH_Create_Snapshot_dialog_box;IOC_Replication')} Related Topics

Replication

The Replication feature of Oracle7 database enables you to create copies (known as snapshots) of data located on other databases. You can use these snapshots to keep abreast of current information and to change the data on the remote site, if necessary.   The table you are copying is known as the master table.

You can create two types of snapshots: Read-Only, which is also called publish-and-subscribe, or primary site-replication, and Updatable, which is also called update-anywhere replication.

For additional information about Replication, refer to the online Help topics related to Oracle7 Symmetric Replication

Read-Only Snapshot

When you create a read-only snapshot, you have created a copy (known as a replica) of a master table. You can create an unlimited number of read-only snapshots of the table.

This snapshot can be periodically refreshed to reflect the most recent state of the master table. During a snapshot refresh, any changes made to the master table since the snapshot was created or previously refreshed are reflected in the snapshot.

Updatable Snapshot

Updatable snapshots also let you create replicas of a master table. The difference is that all copies of data can be updated, and updates are propagated to all other copies. This gives you immediate local access to data, and allows systems to function autonomously even if the network fails.

Changes to local copies are stored locally and periodically forwarded either automatically or manually. To create an updatable snapshot, you must have an Oracle7 Server set up as a replication master site.

{button ,AL(`dbconcepts;IDH_Oracle7_Symmetric_Replication;PRC_To_create_a_snapshot;IDH_Create_Snapshot_dialog_box;IOC_About_Snapshots')}   Related Topics

Database Connections

Once you have logged on to a database, you have established an active database connection. In the Personal Oracle7 Navigator, Database Connections is a folder that you can use to hold a list of database connections that you have defined.

When you select a particular database connection that you have defined, you establish an active database connection. You need a database connection to work with either the local database or a database accessible to you via a network connection (known as a remote database). This gives you access to the data located on either the local or the remote database.

In this product, you already have a connection defined for the local database. When you click Local Database in the Navigator, you are establishing an active database connection to the local database.

If you want to establish an active connection to a remote database, you must first create the database connection. For more information on creating a remote database connection, please see Connecting to Local and Remote Databases. Once you have created a database connection to a remote database, you can select the database connection to establish an active database connection to a remote database. You can view database objects such as tables, views, indexes, etc. located on a remote database. You can copy these objects to a project or to your local database using drag and drop or copy and paste. Right now, a table is the only type of object that you can drag and drop from a remote database to your local database or to a Project.

In addition to creating a database connection to a remote database, you can also create an additional database connection to the local database.

{button ,AL(`dbconcepts;PRC_To_create_a_database_connection;PRC_To_create_a_table_on_the_Local_Database;IOC_Connecting_to_Local_and_Remote_Databases;IDH_Create_Database_Connection_dialog-box;IOC_About_Database_Links')} Related Topics

Connecting to Local and Remote Databases

Personal Oracle7 has the capability to connect to other databases. It does this either through a database connection or a database link.

Database Connection

When you have logged on to a database, you have established an active database connection. In the Personal Oracle7 Navigator, Database Connections is a folder that you can use to hold a list of database connections that you have defined. Database connections are primarily used to connect an application (e.g., the Personal Oracle7 Navigator) to a database.

When you select a particular database connection that you have defined, you establish an active database connection. A database connection is used to connect to one database at a time.

To create a database connection, you must identify the path used to get to the database to which you are trying to connect. This path can either be an alias, a connect string, a service name, or a connect descriptor.

Database Link

A Database Link is an object in the database that links one database to another database. You can use this link to get data from one or several remote databases. When you use a database link, it makes a database connection to a remote database.

This link resembles a map, telling your database how to get to the remote database. When you create a link, you must identify the path used to get from the database to which you are connected to the database to which you are trying to connect. This path can either be an alias, a connect string, a service name, or a connect descriptor.   It is the same information that you need to provide for a database connection.

Remote Database Connections

A remote database is one that is accessible to you via a network connection   If you want to connect to a remote database, you must know its alias or connect string.

Local Database Connections

If you select Local Database in the Personal Oracle7 Navigator, you are automatically connected to the local database.

If you want to connect to your local database from another application such as Oracle Forms or SQL*Plus, you may be asked for the host string, alias, connect string, database, connect descriptor, or some similar label, depending on the application. In most cases, you can leave this field blank. This is because most applications connect to the local database by default.

Some applications, such as ODBC require that you enter something in this field. If you find that you cannot connect to the local database, enter `2:`. (this is the alias or connect string for the local database) in this field.

{button ,AL(`dbconcepts;IDH_Create_Database_Connection_dialog_box;IOC_About_Database_Links;PRC_To_create_a_database_connection;IDH_Create_Database_Link_dialog_box')}   Related Topics

Connecting to Personal Oracle7

To connect to Personal Oracle7 from any application, you need to know the following:

User Name and User Password

A user name is a unique name by which a user is known to the Oracle7 database and to other Oracle users. A user name must be associated with a user password. The maximum length for the user name is 30 characters.

A user password is a word or phrase (no spaces or commas allowed) associated with a user name. Keep this password secret. The maximum length is 30 characters.

The starter database comes with some default user names and user passwords, which are shown in the following table. Use these default values until you establish your own user name and user password.

User Name	User Password
SCOTT	TIGER
SYSTEM	MANAGER
SYS	SYS [Windows 95 product id number excluding dashes]
DEMO	DEMO
PO7	PO7

The Windows 95 product ID number can be found by clicking the Start button, Settings, Control Panel, then double clicking the System application. The product id number is on the General page under Registered to:

Alias or Connect String

To connect to your local database from another application such as Oracle Forms or SQL*Plus, you may be asked for the host string, database, connect descriptor, or some similar label, depending on the application. In most cases, you can leave this field blank. This is because most applications connect to the local database by default.

Some applications, such as ODBC require that you enter something in this field. If you find that you cannot connect to the local database, enter 2: (this is the alias or connect string for the local database) in this field.

16-Bit Applications

To use 16-bit applications with this product, you need to install the most recent Oracle 16-bit Required Support Files. Please review the installation guide for Personal Oracle7 for Windows 95 or the Readme.doc file in the Personal Oracle7 for Windows 95 program group for information on installing these files.

The Oracle7 Server

Personal Oracle7 database shares many features with its "parent" software, the Oracle7 Server:

- the power of the Oracle7 relational database system
- full compatibility with all Oracle products

However, unlike Oracle7 servers--which support numerous users in large and sophisticated database environments--Personal Oracle7 has been designed especially for personal use by those who may or may not have experience administering a sophisticated database. Personal Oracle7 differs from the Oracle7 Server in the following areas:

- Personal Oracle7 includes an interface called the Personal Oracle7 Navigator for use on Windows 95 workstations. Most common database administration tasks can be performed with the Navigator.
- Personal Oracle7 has been preconfigured with common defaults allowing you to get your database up and running quickly.
- Personal Oracle7 comes with a "Starter" database configured to handle most common applications. Oracle recommends that you use this database for your applications.
- Personal Oracle7 can use database links, distributed queries, the distributed database option (one logical database stored on several computers), read-only and updatable snapshots, and two-phase commit. The parallel database (multiple instances that share a physical database) option is not available.

{button ,AL(`dbconcepts;IOC_Introduction_to_Personal_Oracle7;IOC_Getting_Started_with_Personal_Oracle7')}   Related Topics

Database Objects Overview

A database object is anything defined and stored in a database. The following is a list of database objects: tables, views, indexes, synonyms, database links, roles, snapshots, and users.

{button ,AL(`about_dbobjects;cr_dbobjects')}   Related Topics

About Tables

All data in a relational database is stored in tables. Every table has a table name and a set of columns and rows in which the data is stored. Each column is given a column name, a datatype (defining characteristics of the data to be entered in the column), and a width (quantity of space to allocate for data to be entered into the column).

Usually in a relational database, some of the columns in different tables contain the same information. In this way, the tables can refer to one another.

For example, you might want to create a database containing information about the products your company manufactures. In a relational database, you can create several tables to store different pieces of information about your products, such as an inventory table, a manufacturing table, and a shipping table. Each table would include columns to store data appropriate to the table (for example, the inventory table would include a column showing how much stock is on hand) and a column for the product’s part number.

This way, if you know a product’s part number, you could find out how much product you have on hand, how long the product takes to manufacture, and when the product shipped. This common column creates a relationship between all the different tables.

{button ,AL(`about_dbobjects;PRC_To_create_a_table_using_Table_Wizard;PRC_To_add_an_object_to_a_project_by_creating_a_new_object;PRC_To_add_an_object_to_a_project_from_the_local_database;PRC_To_add_an_object_to_a_project_from_another_project;PRC_To_add_a_table_to_a_project_from_a_remote_database;PRC_To_create_a_table_on_the_local_database;PRC_To_create_a_table_on_a_remote_database;PRC_To_view_data_in_a_table;PRC_To_edit_data_in_a_table;PRC_To_modify_a_table;PRC_To_modify_the_design_of_a_table;PRC_To_grant_object_privileges_on_tables_views_or_snapshots;IOC_The_Relational_Database;IOC_Database_Objects')} Related Topics

About Views

A view is a customized presentation of the data from one or more tables. Views derive their data from the tables on which they are based, which are known as base tables. All operations performed on a view actually affect the base tables of the view. You can use views for several purposes:

- to give you an additional level of table security by restricting access to a predetermined set of table rows and columns. For example, you can create a view of a table that does not include sensitive data (i.e., salary information).
- to hide data complexity. Oracle7 databases usually include many tables, and by creating a view combining information from two or more tables, you make it easier for other users to access information from your database. For example, you might have a view that is a combination of your Employee table and your Department table. A user looking at this view, which you have called emp_dept, only has to go to one place to get information, instead of having to access the two tables that make up this view.
- to present the data in a different perspective from that of the base table. Views provide a means to rename columns without affecting the base table.
- to store complex queries. For example, a query might perform extensive calculations with table information. By saving this query as a view, the calculations are performed only when the view is queried.

{button ,AL(`about_dbobjects;PRC_To_add_an_object_to_a_project_by_creating_a_new_object;PRC_To_add_an_object_to_a_project_from_the_local_database;PRC_To_add_an_object_to_a_project_from_another_project;PRC_To_create_a_view;IOC_Database_Objects;PRC_To_grant_object_privileges_on_tables_views_or_snapshots')} Related Topics

About Indexes

An index is used to quickly retrieve information from a database project. Just as book indexes help you retrieve specific information faster, a database index provides faster access to table data. Indexing creates an index file consisting of a list of records in a logical record order, along with their corresponding physical position in the table. You can use indexes to rapidly locate and display records, which is especially important with large tables, or with databases composed of many tables

Indexes are created on one or more columns of a table. Once created, an index is automatically maintained and used by the Oracle7 Database. Changes to table data (such as adding new rows, or deleting rows) are automatically incorporated into all relevant indexes.

To understand how an index works, suppose that you have created an employee table containing the first name, last name, and employee ID number of hundreds of employees and that you entered the name of each employee into the table as they were hired. Now, suppose you want to locate a particular record in the table. Because you entered information about each employee in no particular order, the DBMS must do a great deal of database searching to find the record.

If you create an index using the LAST_NAME column of your employee table, the DBMS has to do much less searching and therefore can return the results of a query very quickly.

{button ,AL(`about_dbobjects;PRC_To_add_an_object_to_a_project_by_creating_a_new_object;PRC_To_add_an_object_to_a_project_from_the_local_database;PRC_To_add_an_object_to_a_project_from_another_project;PRC_To_create_an_index;IOC_Database_Objects')} Related Topics

About Synonyms

A synonym is an alias for a table, view, sequence, or program unit. A synonym is not actually an object itself, but instead is a direct reference to an object. You can use a synonym to:

- mask the real name and owner of an object
- provide public access to an object
- mask the true location of a database object (For example, one table may be located on a database in Boston, and another table may be located on a database in New York. If you use synonyms, to the user, both tables appear to be located in one place)
- provide users or yourself with a simple name for a database object

Synonyms can be private or public. A private synonym is usable only by the person who owns the synonym. A public synonym is one that any user can employ to gain access to a database object.

When users who do not own a database object, such as a table, wish to reference that table, they can use a synonym to refer to the table. Database administrators most often create public synonyms that make the database objects available for system-wide use by database users.

{button ,AL(`about_dbobjects;PRC_To_add_an_object_to_a_project_by_creating_a_new_object;PRC_To_add_an_object_to_a_project_from_the_local_database;PRC_To_add_an_object_to_a_project_from_another_project;PRC_To_create_a_synonym;IOC_Database_Objects')} Related Topics

About Snapshots

A snapshot is a copy either of a full table or a subset of a table. Snapshots reflect the current state of the table you are copying, which is known as the master table.

You can create two types of snapshots: Read-Only, which is also called publish-and-subscribe replication, and Updatable, which is also called update-anywhere replication.

Read-Only Snapshot

When you create a read-only snapshot, you create a copy (known as a replica) of a master table. You can create an unlimited number of read-only snapshots of the table. This snapshot can be periodically refreshed to reflect the most recent state of the master table. During a snapshot refresh, changes made to the master table since the snapshot was created or previously refreshed are reflected in the snapshot.

Updatable Snapshot

Updatable snapshots also let you create replicas of a master table. The difference is that all copies of data can be updated, and updates are propagated to all other copies.

This gives you immediate local access to data, and allows systems to function autonomously even when the network fails. Changes to local copies are stored locally and then periodically forwarded. When data changes in the master table, the snapshot is also updated or refreshed automatically. In addition, any changes you make to data in a snapshot can be updated or ‘‘pushed up’’ to the master table.

{button ,AL(`about_dbobjects;PRC_To_add_a_snapshot_to_a_project;PRC_To_create_a_snapshot;PRC_To_grant_object_privileges_on_tables_views_or_snapshots;IOC_Database_Objects;IDH_Replication')}   Related Topics

About Database Links

A database link is an object that describes a path from one database to another. The path tells Oracle7 how to get from one database to another. When you use a database link, it makes a database connection to a remote database.

A database link requires that SQL*Net be running on each of the machines involved.

For example, if there are two computers, each running SQL*Net, with a database on each, a database link is used to link the first database (named Local) to the second database (named Remote).

Database links require the same information as a database connection:

- a database alias, connect string, or connect descriptor for a remote database
- name of a valid user account on the remote database
- password for the valid account in the remote database

{button ,AL(`about_dbobjects;PRC_To_add_an_object_to_a_project_by_creating_a_new_object;PRC_To_add_an_object_to_a_project_from_the_local_database;PRC_To_add_an_object_to_a_project_from_another_project;PRC_To_create_a_database_link;IOC_Database_Objects')}   Related Topics

About Users

Anyone who accesses an Oracle7 database is a user. Each user has a user name and a user password.

When you first install Personal Oracle7 for Windows 95, certain default user names and user passwords are set up for you. You can use these defaults until you establish user names and passwords of your own.

User Name

A unique name by which a user is known to the Oracle7 database and to other Oracle users. A user name must be associated with a user password. The maximum length of the user name is 30 characters.

The default user names are SCOTT, SYSTEM, SYS, DEMO, or PO7.

User Password

A secret word or phrase associated with a user name. The maximum length is 30 characters.

The default user passwords for the default user names are:

User Name	**User Password**
SCOTT	TIGER
SYSTEM	MANAGER
SYS	SYS [Windows 95 product id number excluding dashes]
DEMO	DEMO
PO7	PO7

The Windows 95 product ID number can be found by clicking the Start button, Settings, Control Panel, then double clicking the System application. The product id number is on the General page under Registered to:

{button ,AL(`about_dbobjects;PRC_To_add_an_object_to_a_project_by_creating_a_new_object;PRC_To_add_an_object_to_a_project_from_the_local_database;PRC_To_add_an_object_to_a_project_from_another_project;PRC_To_create_a_user;IOC_Database_Objects;PRC_To_create_a_role;IOC_Users_Passwords_and_Roles')} Related Topics

About Roles

A role is a method of granting privileges to work with database objects. The owner or creator of the object can grant these privileges, or the privilege can be granted to a role. Then, if that role is granted to a user, the user gets all privileges associated with that role. Users can have one or more roles granted to them.

{button ,AL(`about_dbobjects;PRC_To_add_an_object_to_a_project_by_creating_a_new_object;PRC_To_add_an_object_to_a_project_from_the_local_database;PRC_To_add_an_object_to_a_project_from_another_project;PRC_To_create_a_user;IOC_Database_Objects;PRC_To_create_a_role;IOC__Users_Passwords_and_Roles')}   Related Topics

To disable or enable automatic database startup

1 On the Windows 95 taskbar, click the Start button.

2 Click Run.

3 Do one of the following:

- To disable automatic database startup, type

```
oraautos off
```

- To enable automatic database startup, type

```
oraautos on
```

4 Click OK.

{button ,AL(`databases')} Related Topics

To create a database connection

1 In the Personal Oracle7 Navigator, right-click Database Connections.

2 Right-click New. The Create Database Connection dialog box appears.

3 Type the name for the database connection in the Name field.

4 Type the user name in the User Name field.

5 Type the user password in the Password field.

6 Type the connect descriptor for the database to which you are connecting in the Connect field.

7 If you want to connect to the database in the future without being prompted for the password, leave Save Password checked. If you want to be prompted for the password, uncheck Save Password.

8 If you are adding a new SQL*Net Version 2 database alias, click the Configure button. SQL*Net Easy Configure appears. Use this to configure your new database alias. When you have finished configuring your new database alias, return to this dialog box. The new database alias will appear in the Connect list.

Note

▶ If you don't have or don't know your username, password, or the alias or connect string of the database to which you are connecting, ask your database administrator. If you don't have a database administrator, ask the owner of the database to which you are trying to connect.

{button ,AL(`databases;IOC_Database_Connections;IOC_Connecting_to_Other_Databases;PRC_To_copy_a_table_from_a_remote_database_to_Personal_Oracle7;IDH_Create_Database_Connections_dialog_box')} Related Topics

To configure a SQL*Net connection

1 On the Windows 95 taskbar, click the Start button.

2 Point to Programs, and then point to Oracle for Windows 95. A menu of options appears.

3 Click SQL*Net Easy Configuration. Follow the directions in the dialog boxes for configuring a SQL*Net connection.

Note

▶ You can also configure a SQL*Net connection by clicking the Configure button in the Create Database Connection dialog box.

{button ,AL(`databases;IOC_Connecting_to_Other_Databases;IOC_Database_Connections;IDH_Create_Database_Connection_dialog_box')} Related Topics

To change database parameters

1 Start any text editor such as WordPad.

2 Open the database parameter file. For the Starter database, the database parameter file is called Initorcl.ora located in the Database directory on ORACLE_HOME.

3 Edit the appropriate database parameters.

4 Save the database parameter file.

5 Shut down and restart the database for the new parameters to take effect.

Note

▶ For more information on changing database parameters, see About Initialization Parameter Files and the *Oracle7 Server Administrator's Guide,* available online as an Adobe Acrobat-readable file.

{button ,AL(`databases;IDH_Configuration_Parameters')} Related Topics

To start or stop the database manually

1 On the Windows 95 taskbar, click the Start button.

2 Point to Programs, and then point to Personal Oracle7 for Windows 95. A menu of options appears.

3 Do one of the following:

- To start the database, click Start Database
- To stop the database, click Stop Database.

{button ,AL(`databases')} Related Topics

To start the database at Windows 95 startup

1. On the Windows 95 taskbar, right-click the Start button.
2. Click Open. The Start Menu window appears.
3. Double-click Programs. The Programs window appears.
4. Double-click Personal Oracle7 for Windows 95. The Personal Oracle7 for Windows 95 window appears.
5. Right-click Start Database, then click Copy.
6. Return to the Programs window, which you opened in step 3.
7. Right-click Startup, then click Paste.

After performing this task, the database will start whenever you start or log in to Windows 95.

{button ,AL(`databases')} Related Topics

To migrate data from Personal Oracle7 for Windows 3.1

See the Readme.doc file in the Personal Oracle7 for Windows 95 program group for information on performing this procedure.

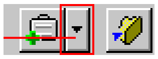
Note

▶ For information on database migration from an operating system other than Windows 3.1, see the *Oracle7 Server Migration* documentation (Part Number A19484-2), available online as an Adobe Acrobat-readable file.

{button ,AL(`databases')} Related Topics

To create a project

1 On the Personal Oracle7 Navigator toolbar, click the New Project/Connection button.

New Project/Connection button 

2 Click New Project. The Create Project dialog box appears.

3 Select the directory in which you want to save this project.

4 Type the name of the project in the File name field.

5 Click Save. A new project window appears.

Notes

- You can also create a new project by right-clicking the Projects folder in the Navigator and then clicking New.
- Create new objects in a project by using the New Database Objects tool bar button. Add existing objects to a project by dragging and dropping them from the local database, a remote database (tables only), or from another project.

{button ,AL(`projects;IOC_Getting_Started_with_Personal_Oracle7;IOC_About_Projects')} Related Topics

To add a local object to a project

Click any of the following for information on adding local objects to projects:

{button ,JI(`',`PRC_To_add_an_object_to_a_project_by_creating_a_new_object')} To add an object to a project by creating a new object

{button ,JI(`',`PRC_To_add_an_object_to_a_project_from_the_local_database')} To add an object to a project from the local database

{button ,JI(`',`PRC_To_add_an_object_to_a_project_from_another_project')} To add an object to a project from another project

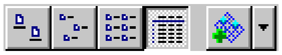
Note

▶ The only object you can add to a project from a remote database is a table. For information on how to do this, see To add a table to a project from a remote database.

{button ,AL(`IOC_About_Projects;p_projects')} Related Topics

To add an object to a project by creating a new object

1. On the left side of the Personal Oracle7 Navigator, click the project in which you want to create a database object. A list of objects already in the project appears on the right side of the Navigator.
2. On the toolbar, click the New Database Object button.

   
   New Database Object button

   A menu of commands to create new objects appears.
3. Click the kind of object you want to create. The dialog box for that kind of object appears.
4. Complete the options in the dialog box and click Finish or OK, as appropriate. The new object is created in the local database and the object appears in the list on the right side of the Navigator.

{button ,AL(`IOC_About_Projects;p_projects')}   Related Topics

To add an object to a project from the local database

1. On the left side of the Personal Oracle7 Navigator, click Local Database. A list of database object folders appears on the right side of the Navigator. These folders represent the objects in the local database, organized by object type.
2. Double-click the folder for the kind of object you want to add to the project. A list of all objects of that type appears on the right side of the Navigator. For example, double-click Index to display all the indexes in the local database.
3. From the list on the right side of the Navigator, drag the object you want to add to the project to that project on the left side of the Navigator.

Notes

- The steps in this topic assume that you are viewing database objects by type.
- To see that the object has been added to the project, click the project on the left side of the Navigator. The object appears in the list on the right.
- Dragging an object to a project does not remove the object from the database.
- The only database object you can add from a remote database is a table. For information on how to do this, see To add a table to a project from a remote database.

{button ,AL(`IOC_About_Projects;p_projects')} Related Topics

To add an object to a project from another project

1. On the left side of the Personal Oracle7 Navigator, click the project containing the object you want to add to another project. A list of the objects in the project you clicked appears on the right side of the Navigator.

2. From the list of objects in the source project (on the right side of the Navigator), drag the object you want to add to the destination project (on the left side of the Navigator).

Notes

- To see that the object has been added to the project, click the project on the left side of the Navigator. The object appears in the list on the right.
- Dragging an object from one project to another project does not remove the object from the first project.

{button ,AL(`IOC_About_Projects;p_projects')} Related Topics

To add a table to a project from a remote database

1 On the left side of the Personal Oracle7 Navigator, click Database Connections. A list of database connections appears on the right side of the Navigator.

2 On the right side of the Navigator, click the database connection to the database that contains the table you want to copy. A list of the database object folders in that database appears.

3 Double-click the Table folder for that database. A list of the tables in the database appears.

4 From the list of tables, drag the table to the desired project on the left side of the Navigator. The Copy Table dialog box appears.

5 Leave Table selected and click OK. The Copy Table As dialog box appears.

6 Type the name of the table in the Name field.

7 Select the owner's name in the Owner field.

8 Select Structure Only if you want to copy the table structure only. Select Structure and Data if you want to copy both the table definition and the data.

9 Click OK. The table is copied to the local database and is added to the project to which you dragged it.

Note

- The steps in this topic assume that you are viewing database objects by type.

{button ,AL(`projects;IOC_Database_Objects;PRC_To_copy_a_database_object_from_one_project_to_another_project;IOC_About_Database_Objects;IOC_About_Projects;PRC_To_view_database_objects_by_owner;PRC_To_view_database_objects_by_type;dbobjects;ICO_About_Tables;IOC_Database_Connections;IOC_Connecting_to_Other_Databases')} Related Topics

To add a snapshot to a project

1 On the left side of the Personal Oracle7 Navigator, click the database connection to the remote database that contains the table you want to create a snapshot of. A list of the database object folders in that database appears.

2 Double-click the Table folder for that database. A list of tables appears on the right side of the Navigator.

3 Drag the table you want to create a snapshot of to the desired project on the left side of the Navigator. The Copy Table dialog box appears.

4 Click Snapshot, then click OK. The Create Snapshot dialog box appears.

5 Click either Updatable or Read Only and, if necessary, enter the refresh rate.

6 Click OK. The Password dialog box appears.

7 Enter the owner's password for the table and click OK. A snapshot of the table is created in the local database and added to the project.

Note

- The steps in this topic assume that you are viewing database objects by type.

{button ,AL(`dbobjects;ICO_About_Tables;IOC_Database_Connections;IOC_Connecting_to_Other_Databases')} Related Topics

To delete a database object from a project

1 On the left side of the Personal Oracle7 Navigator, click the project containing the database object you want to delete. A list of the objects in the project appears on the right side of the Navigator.

2 Right-click the object to delete.

3 Click Delete, then select one of the options from the prompt:

- Click From Project Only to remove the object from the project but leave it in the database.
- Click From Project And Database to remove the object from the project and to delete it from the database.

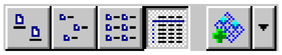
4 Click OK.

{button ,AL(`projects;IOC_About_Projects;PRC_To_view_database_objects_by_owner;PRC_To_view_database_objects_by_type;IOC_Database_Objects')} Related Topics

To add a file object to a project

1 On the left side of the Personal Oracle7 Navigator, click on the project to which you want to add a file object. A list of the objects in the project appears on the right side of the Navigator.

2 Click the New Database Object/File Object button on the right end of the toolbar.



New Database Object button

3 Click New File. The Browse dialog box appears.

4 Locate the file name of the file object you wish to add to the project.

5 Click the file name. Click Open. The file object appears with the list of database objects on the right side of the Navigator.

Note

- A file object can be almost any kind of file you want to add to your project, such as reports, spreadsheets, programs, other applications, slide presentations, sales reports, project notes, etc.

{button ,AL(`projects;IOC_About_Projects;PRC_To_view_database_objects_by_owner;PRC_To_view_database_objects_by_type;IOC_Database_Objects')} Related Topics

To view data in a table

1 In the Personal Oracle7 Navigator, right-click the table icon whose data you want to view.

2 Click Open. The table appears.

3 To close the window, click the X in the upper right corner of the window.

Note

▪ You can select tables in a project, in the local database, or tables that are available through a database connection. Click here for help on how to select any database object.

{button ,AL(`dbobjects;PRC_To_create_a_table_using_the_table_Wizard;PRC_To_create_a_table_manually;IOC_About_Tables;IDH_Create_Table_dialog_box')} Related Topics

To edit data in a table

1 In the Personal Oracle7 Navigator window, right-click the table icon whose data you want to edit.

2 Click Open. The table appears.

3 Do any of the following:

- Click the table cell you want to edit. Enter the new data.
- To add a row, right-click on the table and select Insert Before or Insert After. A new row is inserted. Enter data into the new row.
- To delete a row, click the row you want to delete. Right-click the table. Click Delete Row. A message appears asking if you want to delete the row. Click Yes.

When you move the cursor off the current row, all changes made to that row are committed to the database.

4 When you are finished editing your table, click the X in the upper right corner of the window to close the window. A message appears asking you if you want to commit the changes made to the table. Click Yes.

Note

- You can edit tables in a project, in the local database, or tables that are available through a database connection. Click here for help on how to select any database object.

{button ,AL(`dbobjects;PRC_To_create_a_table_using_the_table_Wizard;PRC_To_create_a_table_manually;IOC_About_Tables;IDH_Create_Table_dialog_box')} Related Topics

To modify the design of a table

1 In the Personal Oracle7 Navigator, right-click the table icon whose design you want to modify.

2 Click Design. The Design Table dialog box appears.

3 Edit the table grid. For more information on editing the table definition grid, see the Create Table dialog box.

4 When you are finished editing the table definition grid, click on OK. A message appears asking you if you want to commit the changes made to the table. Click Yes.

Note

- You can select tables in a project, in the local database, or tables that are available through a database connection. Click here for help on how to select any database object.

{button ,AL(`dbobjects;PRC_To_create_a_table_using_the_table_Wizard;PRC_To_create_a_table_manually;IOC_About_Tables;IDH_Create_Table_dialog_box')} Related Topics

To view a snapshot on the local database

1 In the Personal Oracle7 Navigator, right-click the snapshot. A list of options appears.

2 Click Open. The snapshot grid appears in a read-only mode, which means you cannot edit the data in the snapshot.

Note

- You can view snapshots in a project or on the local database. Click here for help on how to select any database object.

{button ,AL(`cr_dbobjects;IOC_About_Snapshots;IDH_Create_Snapshot_dialog_box;IDH_Replication;IDH_Oracle7_Symmetric_Replication;IOC_Database_Objects_Overview')} Related Topics

To edit an updatable snapshot on the local database

1 In the Personal Oracle7 Navigator, right-click the snapshot. A list of options appears.

2 Click Modify. A password dialog box appears.

3 Type the password for the snapshot owner. The snapshot grid appears.

4 Do any of the following:

- Click the table cell you want to edit. Enter the new data.
- To add a row, right-click on the table and select Insert Before or Insert After. A new row is inserted. Enter data into the new row.
- To delete a row, click the row you want to delete. Right-click the table. Click Delete Row. A message appears asking if you want to delete the row. Click Yes.

When you move the cursor off the current row, all changes made to that row are committed to the database.

5 When you are finished editing your snapshot, click the X in the upper right corner of the window to close the window. A message appears asking you if you want to commit the changes made to the snapshot. Click Yes.

Note

- You can edit snapshots in a project or on the local database. Click here for help on how to select any database object.

{button ,AL(`cr_dbobjects;IOC_About_Snapshots;IDH_Create_Snapshot_dialog_box;IDH_Replication;IDH_Oracle7_Symmetric_Replication;IOC_Database_Objects_Overview')} Related Topics

To manually refresh a snapshot on the local database

1 In the Personal Oracle7 Navigator, right-click the snapshot. A list of options appears.

2 Click Refresh. A password dialog box appears.

3 Type the password for the snapshot owner. The snapshot is refreshed.

Note

- You can manually refresh snapshots in a project or on the local database. Click here for help on how to select any database object.

{button ,AL(`cr_dbobjects;IOC_About_Snapshots;IDH_Create_Snapshot_dialog_box;IDH_Replication;IDH_Oracle7_Symmetric_Replication;IOC_Database_Objects_Overview')} Related Topics

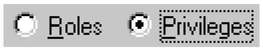
To grant system privileges to users

1 In the Personal Oracle7 Navigator, right-click the user to whom you want to grant system privileges.

2 Click Properties.

3 Click the Role/Privilege tab.

4 To grant system privileges to the user, click the Privileges button below the Remaining window.



○ Roles ● Privileges

5 In the Remaining window, click the system privilege that you want to grant.

6 Click the < button to grant the privilege.

7 If you want to grant more than one privilege, press the Ctrl (Control) key as you click each privilege, and then click the < button.

8 Click OK.

Notes

- Use the << button to grant all privileges.
- You can select users in a project, in the local database, or users that are available through a database connection. Click here for help on how to select any database object.

{button ,AL(`dbobjects;IOC_Users_Passwords_Roles_and_Privileges;PRC_To_create_a_user;IDH_Create_User_dialog_box;IOC_About_Users')} Related Topics

To grant system privileges to roles

1 In the Personal Oracle7 Navigator, right-click a user-created role to which you want to grant system privileges.

2 Click Properties.

3 Click the Role/Privilege tab.

4 To grant system privileges to a role, click the Privileges button below the Remaining window. A list of privileges appears.
▪

5 In the Remaining window, click the privilege you want to grant.

6 Click the ▪ button to grant the privileges.

7 If you want to grant more than one privilege, press the Ctrl (Control) key as you click each privilege, and then click the ▪ button.

8 Click OK.

Notes

- Use the << button to grant all privileges.
- You can select roles in a project, in the local database, or roles that are available through a database connection. Click here for help on how to select any database object.

{button ,AL(`dbobjects;IOC_Users_Passwords_Roles_and_Privileges;PRC_To_create_a_role;IDH_Create_Role_dialog_box;IOC_About_Roles')} Related Topics

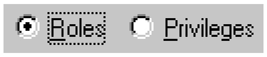
To grant roles to users

1 In the Personal Oracle7 Navigator, right-click a user to which you want to grant a role.

2 Click Properties.

3 Click the Role/Privilege tab.

4 Make sure that the Roles button below the Remaining window is selected.



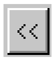
5 In the Remaining window, click the role that you want to grant.

6 Click the ▪ button to grant the role.

7 If you want to grant more than one role, press the Ctrl (Control) key as you click each role, and then click the ▪ button.

8 Click OK.

Notes

- Use the 

button to grant all privileges.
- You can select users in a project, in the local database, or users that are available through a database connection. Click here for help on how to select any database object.

{button ,AL(`dbobjects;PRC_To_create_a_user;IDH_Create_User_dialog_box;PRC_To_create_a_role;IOC_About_Roles;IOC_About_Users;IOC_Users_Passwords_Roles_and_Privileges')} Related Topics

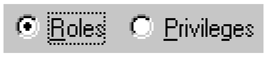
To grant roles to roles

1 In the Personal Oracle7 Navigator, right-click a user-created role to which you want to grant a role.

2 Click Properties.

3 Click the Role/Privilege tab.

4 Make sure that the Roles button below the Remaining window is selected.



5 In the Remaining window, click the role that you want to grant.

6 Click the bmc grantbut.bmp} button to grant the role.

7 If you want to grant more than one role, press the Ctrl (Control) key as you click each role, and then click the ▪ button.

8 Click OK.

Notes

▪ Use the

<< button to assign all privileges.

▪ You can select roles in a project, in the local database, or roles that are available through a database connection. Click here for help on how to select any database object.

{button ,AL(`dbobjects;PRC_To_create_a_role;IDH_Create_Role_dialog box;IOC_About_Roles;IOC_Users_Passwords_Roles_and_Privileges')} Related Topics

To grant object privileges on tables, views, or snapshots

1 In the Personal Oracle7 Navigator, right-click the table, view, or snapshot for which you want to grant object privileges.

2 Click Properties.

3 Click the Object Privilege tab.

4 When prompted, type the object owner's password and click OK.

5 In the Privilege For box, click User or Role.

6 From the list on the Privileges For box, click the user or role to which you want to grant object privileges.

7 In the Remaining window, click the object privilege you want to grant.

8 Click the ▪ button to grant the object privileges.

9 If you want to grant more than one privilege, press the Ctrl (Control) key as you click each privilege, and then click the ▪ button.

10 Click OK.

Notes

- Use the << button to assign all privileges.
- You can select tables, views, or snapshots in a project, in the local database, or tables, views, or snapshots that are available through a database connection. Click here for help on how to select any database object.

{button ,AL(`dbobjects;IOC_About_Tables;IOC_Users_Passwords_Roles_and_Privileges;IOC_About_Views;IOC_About_Snaphots;IOC_Database_Objects_Overview;IDH_Create_Table_dialog_box;IDH_Create_View_dialog_box;IDH_Create_Snapshot_dialog_box')} Related Topics

To change a user password

1 In the Personal Oracle7 Navigator, right-click the user whose password you want to change.

2 Click Properties.

3 Type the new password in the New field.

4 Type the new password again in the Confirm field.

5 Click OK.

Note

- You can select users in a project, in the local database, or users that are available through a database connection. Click here for help on how to select any database object.

{button ,AL(`dbobjects;IOC_Users_Passwords_Roles_and_Privileges')} Related Topics

To view database objects by owner

1. In the Personal Oracle7 Navigator, right-click either the Local Database or a specific database connection. A list of options appears.
2. Click Properties.
3. In the View By box, click the Object Owner button.
4. Click OK. A list of owners appears in the Navigator.
5. Click any owner to see a list of all the objects owned by that owner.

Note

- Your ability to see objects owned by a particular owner on a remote database is determined by the privileges that have been granted to you or to your role.

{button ,AL(`dbobjects;IOC_About_Database_Objects;IOC_Users_Passwords_and_Roles')} Related Topics

To view database objects by type

1. In the Personal Oracle7 Navigator, right-click either the Local Database or a specific database connection. A list of options appears.
2. Click Properties.
3. In the View By box, click the Object Type button.
4. Click OK.
5. In the left side of the Navigator, click a database object. If appropriate, a list of objects by Name and Owner appears on the right side of the Navigator window.

{button ,AL(`dbobjects;IOC_About_Database_Objects')}   Related Topics

To select a database object

Objects can be located in a Project, on the local database, or be available through a database connection.

If the object is located in a project:

1. Double-click the Projects folder. A list of projects appears.
2. Click the project name. A list of objects appears on the right side of the Navigator.

If the object is located in the Local Database

1. Double-click the word Local Database. A list of object folders appears on the left side of the Navigator.
2. Click the object folder. A list of objects appears on the right side of the Navigator.

If the object is available through a Database Connection

1. Double-click the Database Connections folder. A list of database connections appears.
2. Click the database connection name. A list of object folders appears on the right side of the Navigator.
3. Double-click the object folder. A list of objects appears on the right side of the Navigator.

Note

- The steps in this topic assume that you are viewing database objects by type.

{button ,AL(`cr_dbobjects;IOC_Database_Objects_Overview')} Related Topics

To create a table using Table Wizard

1. In the Personal Oracle7 Navigator, right-click the Table folder. A list of options appears.
2. Click New. The New Table dialog appears.
3. Click the Use table wizard option and click OK. The Table Wizard appears.
4. Follow the Table Wizard instructions. On the last screen, click the Finish button to create the table.

{button ,AL(`cr_dbobjects;IOC_About_Tables;IDH_Create_Table_dialog_box;IOC_Database_Objects_Overview;PRC_To_create a a_table_on_the_local_database')} Related Topics

To create a table on the local database

There are two ways to create a table on the local database: Create a new table, and Copy a table from the remote database.

Creating a new table

1 In the Personal Oracle7 Navigator, right-click the Table folder. A list of options appears.

2 Click New. The New Table dialog appears.

3 Click Create table manually and click OK. The Create Table dialog box appears.

4 Type the table name in the Name field.

5 Click the down-arrow in the Owner field and select the owner from the list.

6 Fill out the table grid.

7 Click OK. The new table appears in the list on the right side of the Navigator.

Copy table from a remote database

1 In the Personal Oracle7 Navigator, click the database connection to the database that contains the table you want to copy. A list of the database object folders in that database appears.

2 Double-click the Table folder for that database. A list of tables appears on the right side of the Navigator.

3 From the list of tables, drag the table to the Table object folder in the Local Database on the left side of the Navigator. The Copy Table As dialog box appears.

4 Type the name of the table in the Name field.

5 Select the owner’s name in the Owner field.

6 Select Structure Only if you want to copy the table structure only. Select Structure and Data if you want to copy both the table definition and the data.

7 Click OK. The table is copied to the local database.

Notes

- The steps in this topic assume that you are viewing database objects by type.
- See the Create Table dialog box for detailed information on filling out the table grid.
- You can select Table folders in the local database or Table folders that are available through a database connection. Click here for help on how to select any object folder.

{button ,AL(`cr_dbobjects;IOC_About_Tables;IOC_Create_Table_dialog_box;IOC_Database_Objects_Overview;PRC_To_copy_a_table_from_a_remote_database_to_Personal_Oracle7')} Related Topics

To create a table on a remote database

1. In the Personal Oracle7 Navigator, click the database connection to the database on which you want to create a new table. A list of the database object folders in that database appears.
2. Right-click the Table folder. A list of options appears.
3. Click New. The New Table dialog appears.
4. Click Create table manually and click OK. The Create Table dialog box appears.
5. Type the table name in the Name field.
6. Click the down-arrow in the Owner field and select the owner from the list.
7. Fill out the table grid.
8. Click OK. The new table appears in the list on the right side of the Navigator.

Note

- The steps in this topic assume that you are viewing database objects by type.

{button ,AL(`cr_dbobjects;IOC_About_Tables;IOC_Create_Table_dialog_box;IOC_Database_Objects_Overview;PRC_To_create_a_table_on_the_local_database')} Related Topics

To create a view

1 In the Personal Oracle7 Navigator, right-click the View folder.

2 Click New. The Create View dialog appears.

3 Type the name of the view in the Name field.

4 Click the down-arrow in the Owner field and select the owner from the list.

5 Click the down-arrow in the Database field and select the database that contains the table or views you will use to create this view. If you are creating a view on the local database for the user Scott or Demo, a dialog appears asking for the password.

6 Type the owner's password and click OK. Table and View appear in the Source Tables and Columns box.

7 Double-click either Table or View. A list of the owners appears.

8 Double-click the owner. A list of tables or views appears.

9 Double-click the table or view you want to use to create this view. A list of columns appears.

10 Double-click the column you want to use to create this view. The column appears in the Selected Columns grid.

11 Repeat Step 10 to finish creating your view. If you want to add columns from other tables and views, repeat the necessary steps.

12 To move a column up or down, or to remove a column, select a column in the grid and click the either the Up, Down, or Remove button.

13 When you have finished creating your view, click OK.

Note

▪ You can select View folders in the local database or View folders that are available through a database connection. Click here for help on how to select any object folder.

{button ,AL(`cr_dbobjects;IOC_About_Views;PRC_To_view_data_in_a_table;IDH_Create_View_dialog_box;IOC_Database_Objects_Overview')} Related Topics

To create an index

1 In the Personal Oracle7 Navigator, right-click the Index folder. A list of options appears.

2 Click New. The Create Index dialog box appears.

3 Type the name of the index in the Name field.

4 Click the down-arrow in the Owner field and select the owner from the list.

5 Click the down-arrow in the Table field and select a table from the list. The column names of the table appear in the table grid.

6 In the Order column, click the columns in the order in which you want the index built. If you want to clear all index ordering, double click on any column. You must order at least one of the index columns to create an index.

7 Click OK.

Note

▪ You can select Index folders in the local database or Index folders that are available through a database connection. Click here for help on how to select any object folder.

{button ,AL(`cr_dbobjects;IOC_About_Indexes;IOC_Database_Objects_Overview;IDH_Create_Index_dialog_box;')} Related Topics

To create a snapshot

1 From Database Connections in the Personal Oracle7 Navigator, click the database connection to the database that contains the table you want to use for your snapshot. A list of database object folders appears.

2 Click the Table folder. A list of tables appears.

3 Drag a table onto the Snapshot folder in the Local Database. The Create Snapshot dialog box appears.

4 Click either Updatable or Read Only and, if necessary, enter the refresh rate.

5 Click OK. The Password dialog box appears.

6 Enter the owner's password for the table and click OK. A snapshot of the table is created.

Notes

- The steps in this topic assume that you are viewing database objects by type.
- To create an updatable snapshot, you must have an Oracle7 Server set up as a replication master site. For additional information about Replication, refer to the online Help topics related to Oracle7 Symmetric Replication.

{button ,AL(`cr_dbobjects;IOC_About_Snapshots;IDH_Create_Snapshot_dialog_box;IDH_Replication;IDH_Oracle7_Symmetric_Replication;IOC_Database_Objects_Overview')} Related Topics

To create a database link

1. In the Personal Oracle7 Navigator, right-click the Database Link folder. A list of options appears.
2. Click New. The Create Database Link dialog box appears.
3. Type the name of the link in the Name field.
4. Click Public to make the database link public to all users, or click Private to select an owner for the database link from the user names in the drop-down list box.
5. Type the username in the Username field.
6. Type the password in the Password field.
7. Type the connect descriptor of the database in the Connect field.
8. Click OK.

Notes

- If you don’t have or don’t know your username, password, or the connect descriptor of the database to which you want to connect, ask your database administrator. If you don’t have a database administrator, ask the owner of the database to which you are trying to connect.
- You can select Database Link folders in the local database or Database Link folders that are available through a database connection. Click here for help on how to select any object folder.

{button ,AL(`cr_dbobjects;IOC_About_Database_Links;IDH_Create_Database_Links_dialog_box;IOC_Database_Objects_Overview;IOC_Connecting_to_Other_Databases')}   Related Topics

To create a role

1 In the Personal Oracle7 Navigator, right-click the Role folder. A list of options appears.

2 Click New. The Create Role dialog box appears.

3 Type the name of the role in the Name field.

4 If you want to protect the role with a password, click the Password Protected check box and type a password in the Password field.

5 Click OK.

Notes

- You can select Role folders in the local database or Role folders that are available through a database connection. Click here for help on how to select any object folder.
- In order for the new role to be useful, you must grant privileges or roles to your new role. See To grant system privileges to roles and To grant roles to roles.

{button ,AL(`cr_dbobjects;IOC_About_Roles;IOC_Database_Objects_Overview;IDH_Create_Role_Dialog_box;IOC_Users_Passwords_Roles_and_Privileges;PRC_To_grant_system_privileges_to_roles;PRC_To_grant_roles_to_users;PRC_To_grant_roles_to_roles')} Related Topics

To create a user

1. In the Personal Oracle7 Navigator, right-click the User folder. A list of options appears.
2. Click New. The Create User dialog box appears.
3. Type the name of the user in the Name field.
4. Type the new password for the user in the New field.
5. Type the new password again in the Confirm field.
6. Click OK.

Notes

- You can select User folders in the local database or User folders that are available through a database connection. Click here for help on how to select any object folder.
- When you create a new user, it is automatically assigned the CONNECT role. You may want to add more privileges and roles to this user. For information on how to do this, see To grant system privileges to users and To grant roles to users.

{button ,AL(`cr_dbobjects;IOC_Users_Passwords_Roles_and_Privileges;IOC_About_Users;IOC_Database_Objects_Overview;IDH_Create_User_dialog_box;PRC_To_grant_system_privileges_to_users;PRC_To_grant_roles_to_users')} Related Topics

To create a synonym

1. In the Personal Oracle7 Navigator, right-click the Synonym folder. A list of options appears.
2. Click New. The Create Synonym dialog box appears.
3. Type the name of the synonym in the Name field.
4. Click Public to make the synonym public to all users, or click Private to select an owner for the synonym from the user names in the drop-down list box.
5. Click the down arrow in the Database field to select the database that contains the table for which you are creating this synonym.
6. Click the down arrow in the Table field to select the table for which you are creating this synonym.
7. Click OK.

Notes

- You can select Synonym folders in the local database or Synonym folders that are available through a database connection. Click here for help on how to select any object folder.
- If you choose to make the synonym public, it does not appear under your name as owner. Instead, it appears under the Public owner.

{button ,AL(`cr_dbobjects;IOC_About_Synonyms;IOC_Database_Objects_Overview;IDH_Create_Synonym_dialog_box;')} Related Topics

To select a database object folder

Object folders can be located on the local database or through a database connection.

If the object is located in the Local Database

1 In the Personal Oracle7 Navigator, click the Local Database. A list of object folders appears on the right side of the Personal Oracle7 Navigator.

If the object is available through a Database Connection

1 In the Personal Oracle7 Navigator, double-click the Database Connections folder. A list of database connections appears.

2 Click the database connection name. A list of object folders appears on the right side of the Navigator.

Note

- The steps in this topic assume that you are viewing database objects by type.

{button ,AL(`cr_dbobjects;IOC_Database_Objects_Overview')} Related Topics

To export a project

1. In the Personal Oracle7 Navigator, right-click the project you want to export.
2. Click Export. The Export-Save As dialog box appears.
3. Use the Save In field to select the location of the exported file.
4. Type the file name in the File name field.
5. Click OK.

{button ,AL(`imp_exp;IOC_About_Projects;p_projects')}   Related Topics

Personal Oracle7 Navigator Overview

The Personal Oracle7 Navigator is an application that you can use to access, create, modify, and delete projects, database objects, and database connections. It functions much the same as the Windows 95 Explorer functions.

When you first access the Navigator, you see the following elements:

- Menu Bar
- Tool Bar
- Projects
- Local Database
- Database Connections

The following table shows some actions and results for the Navigator.

Action	**Result**
Click any object on the left side	The right side changes to reflect the contents of the selected object.
Click button (+ or - sign next to an object or folder) on the left side	The left side expands or collapses to show the objects contained in the object or folder
Double-click any object on the right side	The right side changes to reflect the contents of the selected object, some default operation (like opening the object) is performed on the object
Drag an item from the right side and drop it onto an object or folder on the left side	The selected item is copied into the object or folder on the left side
Right click any object on either the left or the right side	A list appears showing the actions that can be performed on the object

Note: Not every object can be copied into an object folder. For example, a table cannot be copied into a User folder. An object cannot be copied to the Project folder, but it can be copied to a specific project that has been created. In these cases, the icon changes to a circle with a slash through it to let you know that you cannot perform this function.

You can use the Navigator to create, delete, move, view, and modify database objects. For a more complete description of what can be done with the Navigator, please refer to the online Help section, Step-by-Step Procedures.

You can also use it to view database objects by type (you can view all tables, or all snapshots) or by user (you can view all objects owned by Scott or by any other user). For help on doing this, refer to the following online Help topics: To view database objects by owner and To view database objects by type.

{button ,AL(`windows;IOC_Introduction_to_Personal_Oracle7')} Related Topics

Personal Oracle7 Navigator Menu Bar

Most of the functions that can be performed from the Personal Oracle7 Navigator menu bar can also be performed by using either the Personal Oracle7 Navigator toolbar or by right-clicking on a particular object. The menu bar contains the following components:

File

New - Use this to create either a new project or a new database connection.

Open - Use this to open a project.

Close - Use this to close a project.

Properties - Use this to view the property sheet for the selected object.

Exit - Use this to shut down the Navigator.

Edit

Cut - Use this to cut database objects or generic objects from a project.

Copy - Use this to make a copy of a database object.

Paste - Use this to paste a copy of a database object.

Delete - Use this to delete a database object.

View

Main Window - Use this to access the Navigator.

Toolbar - Use this to toggle the toolbar off and on. When the check mark appears, the toolbar is visible.

Status Bar - Use this to toggle the status bar off and on. When the check mark appears, the status bar is visible.

Window

This contains Cascade, Tile, and Arrange Icons. Cascade arranges the open windows on top of one another with the title bar of each showing. Tile arranges the open windows side-by side in a tile pattern. Arrange Icons arranges all icons in even rows.

Help

Help Topics - Use this to see all the available Help topics, an Index for the Help topics, and Find, which is a full-text search feature.

Using Help - Use this to see the help on Help that has been provided by Windows 95.

Customer Support - Use this to view information about Oracle’s Customer Support.

About Navigator - Use this to check the version number of the Navigator.

{button ,AL(`windows;IOC_Introduction_to_Personal_Oracle7')}   Related Topics

Personal Oracle7 Navigator Toolbar

Most of the functions that can be performed from the Personal Oracle7 Navigator toolbar can also be performed by using either the Personal Oracle7 Navigator menu bar or by right-clicking on a particular object.

The following discusses the toolbar icons in order from left to right.

New Project/New Connection - This icon changes according to the last usage. You can use this to create either a new project or a new database connection.

New Project/Connection - Use this to create a new project or a new database connection.

Open Project - Use this to open a project.

Show Navigator Window - Use this to view the Navigator window.

Cut - Use this to cut the selected database object from a project.

Copy - Use this to copy the selected database object.

Paste - Use this to paste a database object.

Delete - Use this to delete the selected database object.

Properties - Use this to view the properties of the selected object

Large Icons - Use this to display objects as large icons on the right side of the Navigator.

Small Icons - Use this to display objects as small icons on the right side of the Navigator.

List - Use this to display objects in a list on the right side of the Navigator.

Details - Use this to display information about each object on the right side of the Navigator.

New Database Object - This icon changes according to the last usage. Use this to create a new database object (table, view, index, synonym, database link, snapshot, user, role) or a pointer to a file.

New Database/File Object - Use this to create a new database or file object. If you select this icon when you are in a Project, you can create any new database object or a pointer to a file. If you select this icon when you are in a database object folder, you can only create a database object of that type.

{button ,AL(`windows;IOC_Introduction_to_Personal_Oracle7')}   Related Topics

Personal Oracle7 Navigator Window

The Personal Oracle7 Navigator enables you to manage and create projects and database objects both on your local database or on a remote Oracle database. You can also create database connections to Oracle7 databases.

- Projects - A Project is a way to conveniently organize database objects associated with a particular use. For example, if you want to keep track of the products your company manufactures, you could create a Product Project. A Project can contain tables, views, indexes, etc., which are called database objects. A Project can also contain pointers to external files such as reports, spreadsheets, programs, and other applications. These database objects and nondatabase objects are what you use to store and organize the data related to your Project. Within a Project, you can create, modify, and delete any database object or nondatabase object that will help you to store and organize the data related to your Project.
- Local Database - The local database is the database that is made available to you when you start the Personal Oracle7 database and is the database you connect to when you select Local Database in the Personal Oracle7 Navigator. By default, this is the Starter database that comes with Personal Oracle7 for Windows 95, but it may also be a database that you have created. The Navigator uses a predefined user account to connect to the local database. A user account is a user name and a user password. Each user account is granted certain privileges and roles. The user name for the local database is PO7. The predefined user account has the necessary privileges to allow you to perform all database functions. If you click on the Local Database, a list of database object folders appears.
- Database Connections - Database Connection is a folder that contains a list of database connections that you have defined. When you select a database connection that you have defined, you establish an active database connection. You need a database connection to work with a database that is accessible to you via a network connection (this is known as a remote database). This gives you access to the data that is located on the remote database. In addition to creating a database connection to a remote database, you can also create an additional database connection to the local database.

{button ,AL(`windows;IOC_Introduction_to_Personal_Oracle7')}   Related Topics

Copy Table dialog box

Use this dialog box to select whether a table is to be copied as a table or as a snapshot.

This dialog box contains the following elements:

Table

Select this if you want the table to be copied as a table.

Snapshot

Select this if you want the table to be copied as a snapshot.

Copy Table As dialog box

This dialog box appears when you decide to copy a table as a table (instead of as a snapshot). You use this dialog box to rename the table and to decide if you want to copy the structure of the table or the structure and the data.

This dialog box contains the following elements:

Name

Type the new name of the table. This can be any name you choose.

Owner

Select the owner from the drop-down list.

Structure Only

Select this if you want to copy only the table design.

Structure and Data

Select this if you want to copy both the table design and the table data.

Create Database Connection dialog box

A database connection is a connection either to the local database or to a database that is accessible to you via a network connection (this is known as a remote database). Use the Create Database Connection dialog box to define new database connections. You need a database connection to work with either the local database or a remote database. This gives you access to the data that is located on either the local or the remote database. In this product, you already have a connection to the local database.

This dialog box contains the following elements:

Name

Type the name of the database connection. This can be any name that you choose.

Username

Type the user name you used to log into the database. If you haven’t established your own user name, you can use one of the following default user names: Scott, System, Sys, Demo, or PO7.

Password

Type the password for the user name. If you are using a default user name, the following is a list of the passwords associated with the default user names:

Username	**Password**
Scott	Tiger
System	Manager
Sys	Sys [Windows 95 product ID number excluding dashes]
Demo	Demo
PO7	PO7

The Windows 95 product ID number can be found by clicking the Start button, Settings, Control Panel, then double clicking the System application. The product id number is on the General page under Registered to:

Connect

Use this field to identify the path to the database to which you are trying to connect. This path can either be an alias, a connect string, a service name, or a connect descriptor. If you are unsure about what to enter in this field, contact the administrator of the database to which you are trying to connect.

Configure button

Click this button to configure your SQL*Net connection.

Save Password

Click this box to save the password.

{button ,AL(`dialogs;IOC_Database_Connections;IOC_Connecting_to_Other_Databases;PRC_To_create_a_database_connection;PRC_To_configure_an_SQL*Net_connection')} Related Topics

Type the name of the database connection. This can be any name that you choose.

Type the user name you used to log into the database. If you haven’t established your own user name, you can use one of the following default user names: Scott, System, Sys, Demo, or PO7.

Type the password for the user name. If you are using a default user name, the following is a list of the passwords associated with the default user names:

Username	Password
Scott	Tiger
System	Manager
Sys	Sys [Windows 95 product ID number excluding dashes
Demo	Demo
PO7	PO7

The Windows 95 product ID number can be found by clicking the Start button, Settings, Control Panel, then double clicking the System application. The product id number is on the General page under Registered to:

Use this field to identify the path to the database to which you are trying to connect. This path can either be an alias, a connect string, a service name, or a connect descriptor. If you are unsure about what to enter in this field, contact the administrator of the database to which you are trying to connect.

Click this button to configure your SQL*Net connection.

Click this box to save the password.

Create Database Link dialog box

A database link is an object in the database that allows you to access objects on a remote database.

This dialog box contains the following elements:

Name

Type the name of the new database link. This can be any name that you choose.

Public

Click this radio button to make the database link public to all users.

Private To

Click this radio button to select an owner for the database link from the user names in the drop-down list box. The drop-down list is a list of all existing users. The owner can be any of the existing users. If you want to create a database link owned by a nonexistent user, you must first create that user with the Create User dialog box. If you are creating a database link on the local database, a list of all existing owners appears. If you are creating a database link on a remote database, the only owner that appears is the name of the currently connected user.

Username

Type the user name under which the connection is made. This user name must be valid for the database to which you are trying to connect. If this is left blank, the user’s current login is used.

Password

Type the password associated with the above user name.

Connect

Use this field to identify the path to the database to which you are trying to connect. This path can either be an alias, a connect string, a service name, or a connect descriptor. If you are unsure about what to enter in this field, contact the administrator of the database to which you are trying to connect.

{button ,AL(`dialogs;PRC_To_create_a_database_link;IOC_Connecting_to_Other_Databases')} Related Topics

Type the name of the new database link. This can be any name that you choose.

Click this radio button to make the database link public to all users.

Click this radio button to select an owner for the database link. If you are creating a database link on the local database, a list of all potential owners appears. If you are creating a database link on a remote database, the only owner that appears is the name of the currently connected user.

Type the user name under which the connection is made. This user name must be valid for the database to which you are trying to connect. If this is left blank, the user’s current login is used.

Type the password associated with the above user name.

Use this field to identify the path to the database to which you are trying to connect. This path can either be an alias, a connect string, a service name, or a connect descriptor. If you are unsure about what to enter in this field, contact the administrator of the database to which you are trying to connect.

Create Index dialog box

Indexes are structures that you can create to facilitate your viewing and retrieval of data.

This dialog box contains the following elements:

Name

Enter the name of the new index that you want to create.

Owner

Select the owner for the index from the user names in the drop down list. The drop down list is a list of all existing users. The owner can be any of the existing users. If you want to create a view owned by a nonexistent user, you must first create that user with the Create User dialog box.

Table

Select the table for which you are creating an index from the drop down list of available tables.

Table Definition Grid

Specify the columns in the order in which you want the columns indexed. Clicking on a cell under Order makes the corresponding column the next one on which the index is built. Double clicking in the Order column clears all index ordering.

{button ,AL(`dialogs;PRC_To_create_an_index')} Related Topics

Enter the name of the new index that you want to create.

Select the owner for the index from the user names in the drop down list. The drop down list is a list of all existing users. The owner can be any of the existing users. If you want to create a view owned by a nonexistent user, you must first create that user with the Create User dialog box.

Select a table for which you are creating an index from the drop down list of available tables.

Specify the columns in the order in which you want the columns indexed. Clicking on a cell under Order makes the corresponding column the next one on which the index is built. Double clicking in the Order column clears all index ordering.

Create Role dialog box

A role is a set of privileges that you can define and then assign to one or more users or to other roles. If you need to change the privileges assigned to several users or roles, you can do so globally by changing the privileges assigned to the role.

This dialog box contains the following elements:

Name

Use this field for entering a new name for the role you are creating.

Password Protected

Click this box if you want a password to be associated with this role. If you click this box, enter the password for the new role. If you grant a password-protected role to a user, the user must provide the password when using the role. Passwords are optional for roles. However, only those users who have role privileges may use the role.

{button ,AL(`dialogs;IOC_Users_Passwords_Roles_and_Privileges;PRC_To_grant_system_privileges_to_roles;PRC_To_grant_roles_to_roles;PRC_To_create_a_role')} Related Topics

Use this field for entering a new name for the role you are creating.

Click this box if you want a password to be associated with this role. If you click this box, enter the password for the new role. Passwords are optional for roles.   However, only those users who have role privileges may use the role.

Create Snapshot dialog box

A snapshot is a full copy of a table. The table you are copying is known as the master table.

This dialog box contains the following elements:

Snapshot Type

- Updatable - Select this to create an updatable snapshot. An updatable snapshot is an updatable copy of the master table. Updatable snapshots make use of the Oracle symmetric replication facility, which allows multiple copies of data to be maintained at different sites in a distributed environment. It provides immediate, local access to data. Changes to local copies of data are stored locally and then forwarded either manually or automatically.
- Read-Only - Select this to create a read-only snapshot, which is a copy of a master table. Read-only snapshots can be used for queries only. The read-only snapshot changes to reflect any changes in the master table when the read-only snapshot is refreshed. Read-only snapshots are typically used when there is no worry if there is a discrepancy between the data being updated at different sites.

Next Refresh

The refresh function makes the snapshot reflect any changes that have been made to the master table since the snapshot was created or since the snapshot was last refreshed. Use the hours and minutes boxes to establish the amount of time between each snapshot refresh. If a refresh interval is not specified here, the snapshot needs to be refreshed manually.

{button ,AL(`dialogs;IOC_The_Distributed_Option;IOC_Replication;PRC_To_grant_object_privileges_on_a_table_view_or_snapshot;PRC_To_create_a_snapshot')} Related Topics

Select this to create an updatable snapshot. Updatable snapshots create multiple updatable copies of the same data. Updatable snapshots make use of the Oracle symmetric replication facility, which allows multiple copies of data to be maintained at different sites in a distributed environment. It provides immediate, local access to data. Changes to local copies of data are stored locally and then forwarded either manually or automatically.

Select this to create a read-only snapshot, which is a copy of a master table. Read-only snapshots can be used for queries only. The read-only snapshot changes to reflect any changes in the master table when the read-only snapshot is refreshed. Read-only snapshots are typically used when there is no worry if there is a discrepancy between the data being updated at different sites.

The refresh function makes the snapshot reflect any changes that have been made to the master table since the snapshot was created or since the snapshot was last refreshed. Use the hours and minutes boxes to establish the amount of time between each snapshot refresh. If a refresh interval is not specified here, the snapshot needs to be refreshed manually.

Currently there is no help for this topic.

Create Synonym dialog box

A synonym is an alternative name for a table.   Use the Create Synonym dialog to create a new synonym.

This dialog box contains the following elements:

Name

Type the name of the synonym that you want to create.

Public

Click this radio button to make the synonym public to all users.

Private To

Click this radio button to select an owner for the synonym from the user names in the drop-down list box. The drop-down list is a list of all existing users. The owner can be any of the existing users. If you want to create a synonym owned by a nonexistent user, you must first create that user with the Create User dialog box. If you are creating a synonym on the local database, a list of all existing owners appears. If you are creating a synonym on a remote database, the only owner that appears is the name of the currently connected user.

Database

Select a database from the Database drop down list box. This list contains the database to which you are currently connected and all available database links. The link to the local database is represented by <Personal Oracle7>.

Table

Select a table from the Table drop down list.

Note: If you choose to make the synonym public, it does not appear under your name as owner. Instead, it appears under the Public owner.

{button ,AL(`dialogs;PRC_To_create_a_synonym')}   Related Topics

Type the name of the synonym that you want to create.

Click this radio button to make the synonym public to all users.

Click this radio button to select an owner for the synonym from the user names in the drop-down list box. The drop down list is a list of all existing users. The owner can be any of the existing users. If you want to create a synonym owned by a nonexistent user, you must first create that user with the Create User dialog box. If you are creating a synonym on the local database, a list of all existing owners appears. If you are creating a synonym on a remote database, the only owner that appears is the name of the currently connected user.

Select a database from the Database drop down list box. This list contains the database to which you are currently connected and all available database links. The link to the local database is represented by <Personal Oracle7>.

Select a table from the Table drop down list.

Create Table dialog box

Use the Create Table dialog box to create or modify a table. Each row of the table definition grid represents a column in the table you are creating. (Table columns are sometimes known as “fields”.)

To create tables, you need to know something about the characteristics of the data you will be storing in the table. For example, you need to know if the data in a particular column is to be numerical only (such as a product ID number column), or if it will be a mix of alphabetical and numerical characters (such as an address column).

Note You can increase the width of any column by dragging the internal vertical lines of the column.

This dialog box contains the following elements:

Name

Type the name of the table you want to create.

Owner

Select the owner for the table from the user names in the drop down list.

Columns

Use the Table Definition Grid to define your new table. The grid contains the following columns:

Column Name

Enter the name of the column. Each column name must be unique within the table; the maximum length is 30 characters.

Type

Click the Type column to select a valid Oracle data type from the drop down list. Type is the type of data that will be placed in this column. For example, the data type NUMBER is used for a column in which the data will always be numerical. The data type for an address column would be CHAR because this will contain both numerical and alphabetical characters.

Oracle Data Types

Char

Alphanumeric data at least 1 character long and no more than 255 characters. Use this for fields that have a combination of characters and numbers, such as addresses.

Date

Oracle includes as part of the date the century, year, month, day, hour, minute, and second.

Long

Variable length alphanumeric strings up to 2 gigabytes in size.

Long Raw

Binary data up to 2 gigabytes in size.

Number

Stores numeric zero, positive, or negative fixed or floating point data. Use this for fields that will only contain numbers.

Raw

Binary data up to 255 bytes in size.

RowID

Hexadecimal string representing the unique address of a row in a table.

Varchar2

Alphanumeric data at least 1 character long and no more than 2,000 characters long. This is the same as the CHAR field except it accommodates more text input. You can use this field for entering memos or paragraphs.

Size

Enter the number of characters or bytes necessary to hold your data. For NUMBER data types, data size includes the total number of digits on **both** the left and right side of the decimal point.

Scale

For NUMBER data types, enter the number of digits to the right of the decimal point.

Default

This is the default value for the column that should be entered in case none is specified. This feature is useful if a column cannot be null.

Can be Null?

Click this button to specify whether or not this column can be null. This button is a toggle, and when an X appears on the button, the column can be null.

Unique?

Click this button to specify whether or not this column is unique. This button is a toggle, and when an X appears on the button, the row is unique. A unique key is one or more columns that are unique in each row of a table. A unique key can be used as a primary key. There can be one and only one row with each unique key value.

Primary Key

A set of columns used to enforce uniqueness of rows. The combination of column values is unique for each row in the table. The primary key is the most frequently used means of accessing rows. This field is also used to sort your table and to keep you from duplicating records. Primary keys can be made up of a single column or several columns and are used to locate specific rows of data. An example of a primary key would be a column in a table for a product ID number as long as no two products have the same ID number.

The number in this field designates the position of this column in the primary key that uniquely identifies a row in the table. A primary key must satisfy two conditions: (1) no two table rows can have the same value; and (2) the current column cannot be empty. Click on this column to append it to the primary key. The number that appears will be one greater than the highest primary key value in use for the current table. Double click in the column to clear all primary key designations.

Foreign Key

When you use a primary key from one table in another table, that field is called a foreign key. Enter a foreign key if you want the values of this column to refer to the primary or unique key values in a different table. The general syntax for specifying a foreign key is:

OWNER.TABLE(COLUMN NAME)

Check Condition

A check condition on a column or set of columns requires that a specific condition be true or unknown for every row of a table. For example, if you have a table with a column called SAL that shows the salaries of your employees , you could set a check condition of SAL > 500, which means that the value that you enter in the SAL column must be greater than $500. If you try to enter a value that is below $500, when you try to save this entry, you will get an error message. Enter a combination of one or two or more expressions and logical operators that evaluate to TRUE, FALSE, or unknown. You can make entries here when creating a new table. However, once a table has been saved, you cannot edit the check condition of existing columns. An example of a check condition statement follows:

ORDERID IS NOT NULL

CUSTID IS NOT NULL

TOTAL>=0

Remove Button

Click this to remove the selected column definition.

Up Button

Click this to move the selected column definition up one row.

Down Button

Click this to move the selected column definition down one row.

{button ,AL(`dialogs;PRC_To_edit_data_in_a_table;PRC_To_view_data_in_a_table;PRC_To_modify_the_design_in_a_table;PRC_To_grant_object_privileges_on_a_table_view_or_snapshot;PRC_To_create_a_table_using_table_wizard;PRC_To_create_a_table_manually;PRC_To_modify_a_table')} Related Topics

Type the name of the table you want to create.

Select the owner for the table from the user names in the drop down list.

Use the Table Definition Grid to define your table. Click the Help button in this dialog box for information on the entire dialog box, including detailed information on the Table Definition Grid.

Click this to remove the selected column definition.

Click this to move the selected column definition up one row.

Click this to move the selected column definition down one row.

Create User dialog box

Each user of the database must have an account, consisting of a user name and password. When you create a user account, the user is automatically granted the CONNECT privilege, which allows that user to connect to the database.

This dialog box contains the following elements:

Name

Use this field for entering a new user name for the user account you are creating.

New

Enter the password for the new account.

Confirm

Enter the password again to confirm it.

{button ,AL(`dialogs;IOC_Users_Passwords_Roles_and_Privileges;PRC_To_grant_system_privileges_to_users;PRC_To_grant_roles_to_users;PRC_To_create_a_user')} Related Topics

Use this field for entering a new user name for the user account you are creating.

Enter the password for the new account.

Enter the password again to confirm it.

Create View dialog box

Views are custom-tailored presentations of the data from one or more database tables. They enable you to examine data in arrangements that are different from the actual arrangement of the data in the table(s).

This dialog box contains the following elements:

Name

Enter a name for the view that you want to create.

Owner

Select the owner of the view from the user names in the drop down list. The drop down list is a list of all existing users. The owner can be any of the existing users. If you want to create a view owned by a nonexistent user, you must first create that user with the Create User dialog box.

Database

From the Database drop-down list, click the database that contains the tables and views you will use to create your view. This list contains the database to which you are currently connected and all available database links. The link to the local database is represented by <Personal Oracle7>.

After selecting the database, select either a table or a view, and then select a column or columns that you would like to be included in the view that you are creating.

Add button

Click this button to add the selected column to the Selected Columns box.

Selected Columns

Lists all columns that you have selected to become part of your view.

Column Name

The name of the column appears here.

New Name

If you want to rename a column, type the new column name here.

Type

The datatype of the column appears here.

Condition

If you want to set a condition for the column, enter the condition statement here. A condition on a column limits the number of rows that are used for your view. For example, if you have a table with a column called SAL that shows the salaries of your employees , you could set a condition of SAL > 5000, which means that for your view, you only want to see the salaries of employees which are greater than $5000.

Source

The name of the user and the name of the table from which this column came appears here.

Location

The name of the database from which this column came appears here.

Remove button

Click this button to remove a column from the Selected Columns box.

Up button

Click this button to move the selected column up one row.

Down button

Click this button to move the selected column down one row.

{button ,AL(`dialogs;PRC_To_grant_object_privileges_on_a_table_view_or_snapshot;PRC_To_create_a_view')} Related Topics

Enter a name for the view that you want to create.

Select the owner of the view from the user names in the drop down list. The drop down list is a list of all existing users. The owner can be any of the existing users. If you want to create a view owned by a nonexistent user, you must first create that user with the Create User dialog box.

From the Database drop-down list, click the database that contains the tables and views you will use to create your view. This list contains the database to which you are currently connected and all available database links. The link to the local database is represented by <Personal Oracle7>.

Click this button to add the selected column or table to the Selected Columns box.

Lists all columns that you have selected to become part of your view. Click the Help button in this dialog box for information on the entire dialog box, including detailed information on the Selected Columns grid.

Click this button to remove a column from the Selected Columns box.

Click this button to move the selected column up one row.

Click this button to move the selected column down one row.

Delete Object Dialog Box

The dialog appears if you try to delete an object from a Project.

This dialog box contains the following elements:

From Project Only

Select this if you want to delete the object only from the Project. The object still remains in the database.

From Project and Database

Select this if you want to delete the object from both the Project and the database.

Design Table dialog box

Use the Design Table dialog box to change the design of an existing table.

This dialog box contains the following elements:

Name

This field contains the name of the table.

Owner

This field contains the name of the owner of the table.

Columns

The Table Definition Grid is filled in. The grid contains the following columns:

Column Name

The name of the column is filled in. Each column name must be unique within the table; the maximum length is 30 characters.

Type

The Oracle data type is filled in. Type is the type of data that will be placed in this column. For example, the data type NUMBER is used for a column in which the data will always be numerical. The data type for an address column would be CHAR because this will contain both numerical and alphabetical characters. Click on the down arrow to display the list of data types.

Oracle Data Types

Char

Alphanumeric data at least 1 character long and no more than 255 characters. Use this for fields that have a combination of characters and numbers, such as addresses.

Date

Oracle includes as part of the date the century, year, month, day, hour, minute, and second.

Long

Variable length alphanumeric strings up to 2 gigabytes in size.

Long Raw

Binary data up to 2 gigabytes in size.

Number

Stores numeric zero, positive, or negative fixed or floating point data. Use this for fields that will only contain numbers.

Raw

Binary data up to 255 bytes in size.

RowID

Hexadecimal string representing the unique address of a row in a table.

Varchar2

Alphanumeric data at least 1 character long and no more than 2,000 characters long. This is the same as the CHAR field except it accommodates more text input. You can use this field for entering memos or paragraphs.

Size

The number of characters or bytes necessary to hold your data is filled in.   For NUMBER data types, data size includes the total number of digits on **both** the left and right side of the decimal point.

Scale

For NUMBER data types, the number of digits to the right of the decimal point is filled in.

Default

Default value (if any) is filled in. This is the default value for the column that should be entered in case none is specified. This feature is useful if a column cannot be null.

Can be Null?

This button is either on or off. Click this button to specify whether or not this row can be null. This button is a toggle, and when an X appears on the button, the row can be null.

Unique?

This button is either on or off. Click this button to specify whether or not this row is unique. This button is a toggle, and when an X appears on the button, the row is unique. A unique key is used to uniquely identify each record in a Oracle table. There can be one and only one row with each unique key value.

Primary Key

This field may or may not be filled in. A set of columns used to enforce uniqueness of rows. The combination of column values is unique for each row in the table. The primary key is the most frequently used means of accessing rows. This field is also used to sort your table and to keep you from duplicating records. Primary keys can be made up of a single column or several columns and are used to locate specific rows of data. An example of a primary key would be a column in a table for a product ID number as long as no two products have the same ID number.

The number in this field designates the position of this column in the primary key that uniquely identifies a row in the table. A primary key must satisfy two conditions: (1) no two table rows can have the same value; and (2) the current column cannot be empty. Click on this column to append it to the primary key. The number that appears will be one greater than the highest primary key value in use for the current table. Double click in the column to clear all primary key designations.

Foreign Key

This field may or may not be filled in. When you use a primary key field from one table in another table, that field is called a foreign key. Enter a foreign key if you want the values of this column to refer to the primary or unique key values in a different table. The general syntax for specifying a foreign key is:

OWNER.TABLE(COLUMN NAME)

Check Condition

This field may or may not be filled in. A check condition on a column or set of columns requires that a specific condition be true or unknown for every row of a table. For example, if you have a table with a column called SAL that shows the salaries of your employees , you could set a check condition of SAL > 500, which means that the value that you enter in the SAL column must be greater than $500. If you try to enter a value that is below $500, when you try to save this entry, you will get an error message. Enter a combination of one or two or more expressions and logical operators that evaluate to TRUE, FALSE, or unknown. You can make entries here when creating a new table. However, once a table has been saved, you cannot edit the check condition of existing columns. An example of a check condition statement follows:

ORDERID IS NOT NULL

CUSTID IS NOT NULL

TOTAL>=0

Remove Button

Click this to remove the selected column definition.

Up Button

Click this to move the selected column definition up one row.

Down Button

Click this to move the selected column definition down one row.

{button ,AL(`dialogs;PRC_To_edit_data_in_a_table;PRC_To_view_data_in_a_table;PRC_To_modify_the_design_in_a_table;PRC_To_grant_object_privileges_on_a_table_view_or_snapshot;PRC_To_create_a_table_using_table_wizard;PRC_To_create_a_table_manually;PRC_To_modify_a_table')} Related Topics

This field contains the name of the table.

This field contains the name of the owner of the table.

The Table Definition Grid is filled in. Use the Table Definition Grid to define your table. Click the Help button in this dialog box for information on the entire dialog box, including detailed information on the Table Definition Grid.

Click this to remove the selected column definition.

Click this to move the selected column definition up one row.

Click this to move the selected column definition down one row.

New Table Dialog Box

This dialog box appears when you try to create a new table. You use it to choose between creating a table using the table wizard and creating a table manually.

This dialog box contains the following elements:

Use table wizard

Select this if you want to create a table using the table wizard.

Create table manually

Select this if you want to create a table manually.

Password Dialog Box

This dialog box appears when a user password is needed for confirmation.

This dialog box contains the following element:

Password

Type the correct user password in order to perform the desired function.

About Initialization Parameter Files

The Starter database in Personal Oracle7 for Windows 95 uses database initialization parameters to establish specific database features each time an Oracle7 instance is started. Initialization parameters are located in INIT%ORACLE_SID%.ORA file located in the \ORAWIN95\DATABASE directory. To allow initialization parameters to be unique to a particular database, each database normally has its own initialization parameter file.

An initialization parameter file is an ASCII text file containing a list of parameters. Every instance has a corresponding initialization parameter file and ORACLE_SID parameter.

The initialization parameter filenames take the form `INIT`*`sid`*`.ORA`, where *sid* is the name of the instance. For example, the initialization parameter file for the Starter database is named INITORCL.ORA, because the Starter database SID is ORCL.

By changing initialization parameters values, you can specify features such as:

- the amount of memory the database uses
- whether to archive full online redo log files
- which control files currently exist for the database

Note: For a complete list and descriptions of all initialization parameters, see *Oracle7 Server Reference*, available online as an Adobe Acrobat-readable file.

{button ,AL(`initialization parameters')} Related Topics

Location of Initialization Parameter Files

The computer that starts the instance must have access to the appropriate initialization parameter files. The Starter database in Personal Oracle7 for Windows 95 uses the initialization parameter files located in \ORAWIN95\DATABASE, unless you use SQL*DBA to specify a file with the PFILE option at startup.

{button ,AL(`initialization parameters')}   Related Topics

Editing Initialization Parameter Files

To customize Personal Oracle7 database functions, you may need to edit the initialization parameter files. Use any ASCII text editor to modify the file.

Note: For a complete list and descriptions of all initialization parameters, see *Oracle7 Server Reference*, available online as an Adobe Acrobat-readable file.

{button ,AL(`initialization parameters')} Related Topics

The Sample Initialization Parameter File

The initialization parameter file (`INITORCL.ORA`) used by the Starter Database in \ORAWIN95\DATABASE. Use this file as a model for creating a new Personal Oracle7 database.

The annotated, sample initialization parameter file contains alternative values for the initialization parameters. These values and the annotations are preceded by comment signs (#), which prevent them from being processed. Activate a particular parameter by removing the preceding # sign. De-activate a particular parameter by adding a # sign.

For example, several initialization parameters are specified with three different values to create small, medium, or large System Global Areas (SGAs), respectively. By default, the parameter to create a small SGA is active:

```
db_block_buffers = 200___# SMALL
# db_block_buffers = 550__# MEDIUM
# db_block_buffers = 3200__# LARGE
```

To create a medium-sized SGA, comment out the small parameter definition and activate the medium:

```
# db_block_buffers = 200__# SMALL
db_block_buffers = 550___# MEDIUM
# db_block_buffers = 3200__# LARGE
```

{button ,AL(`initialization parameters')}   Related Topics

Initialization Parameters to Modify when Creating a New Database

Check the initialization parameters described in this section carefully if you decide to create a new database; they **cannot be modified** after database creation.

DB_NAME

Specifies the name of the database to be created. The database name is a string of eight characters or less. You cannot change the name of a database once it has been created.

CHARACTER_SET

Specifies the database NLS character set to use. This parameter can be set only when you create the database.

CONTROL_FILES

Designates the names and locations of all control files to be created and maintained. By default, Personal Oracle7 installs a single control file for the Starter Database, `CTL1ORCL.ORA`, in \ORAWIN95\DATABASE.

If you create your database with the CREATE DATABASE command, Personal Oracle7 creates the control file, `CTL1`*`sid`*`.ORA`, where *sid* is the SID of that database.

To reduce the risk of losing the control file due to disk drive failure, use at least two control files, each located on a separate storage device. The size of Personal Oracle7 control files varies according to the complexity of your database structure. The maximum size of an Personal Oracle7 control file is 2500 database blocks.

Note: See *Oracle7 Server Reference* for more information on initialization parameters.

{button ,AL(`initialization parameters')} Related Topics

Initialization Parameter Names and Values

Oracle7 Server Reference manual includes descriptions of all initialization parameters and instructions for setting and displaying their values. The default values and ranges for parameters with special characteristics under Personal Oracle7 are listed below.

Parameter Name	**Default Value**	**Range of Values**
BACKGROUND_DUMP_DEST	%RDBMSnn%\TRACE\	Any valid directory
COMMIT_POINT_STRENGTH	1	0 - 255
CONTROL_FILES	%ORACLE_HOME%\DATABASE\CTL1%ORACLE_SID%.ORA	Any valid path and filename.
DB_BLOCK_SIZE	2048	512 - 8192
DB_FILES	32	254, cannot exceed DB_BLOCK_BUFFERS
DB_FILE_MULTIBLOCK_READ_COUNT	4	1 - 32
DB_FILE_SIMULTANEOUS_WRITES	4	1 - 24
DISTRIBUTED_TRANSACTIONS	25 * TRANSACTIONS	0 - TRANSACTIONS
LOG_ARCHIVE_BUFFER_SIZE	127	1 - 127
LOG_ARCHIVE_BUFFERS	4	1 - 8
LOG_ARCHIVE_DEST	%RDBMSnn%\	Any valid directory
LOG_ARCHIVE_FORMAT	ARC%S.%T	A valid filename format
LOG_CHECKPOINT_INTERVAL	8000	2 - Unlimited
LOG_FILES	255	2 - 255
LOG_SMALL_ENTRY_MAX_SIZE	800	0 - Unlimited
NLS_LANGUAGE	AMERICAN	Any valid language
NLS_SORT	(Language dependent)	(Language dependent)
NLS_TERRITORY	AMERICA	Any valid territory
OPEN_CURSORS	50	1 - Unlimited
OS_AUTHENT_PREFIX	OPS$	any character string
PROCESSES	25	3 - Unlimited
REMOTE_LOGIN_PASSWORDFILE	shared	shared, exclusive, none
SHARED_POOL_SIZE	3,500,000	300K - Unlimited
SORT_AREA_SIZE	65536	0 - Unlimited
SORT_READ_FAC	20	0 - Unlimited
SORT_SPACEMAP_SIZE	512	0 - Unlimited
TEMPORARY_TABLE_LOCKS	SESSIONS	0 - Unlimited
TRANSACTIONS_PER_ROLLBACK_SEGMENT	30	1 - 255
USER_DUMP_DEST	%RDBMSnn%\TRACE	Any valid directory

{button ,AL(`initialization parameters')}   Related Topics

Configuration Parameters and the Windows 95 Registry

The Windows 95 Registry stores the Oracle configuration parameters. Oracle software uses configuration parameters to locate files and specify runtime parameters common to all Oracle products. When an Oracle program or application requires a translation for a particular configuration variable, Personal Oracle7 consults the associated parameter.

{button ,AL(`configuration parameters')} Related Topics

Editing Personal Oracle7 Configuration Parameters

You can set configuration parameters for Personal Oracle7 in two ways.

In the Windows 95 Registry

Oracle configuration parameters are stored in \HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE. Modify the values stored in this key by running the registry editor program, REGEDIT, included with Windows 95.

To run the Registry Editor,

1. Click the Windows 95 Start menu
2. Click Run. The Run dialog box appears.
3. Type `REGEDIT.` The Registry Editor appears.

At the Command Prompt

The syntax for setting any configuration parameter at the command prompt is:

```
   SET parameter_name=parameter_value
```

where *parameter_name* is the name of the parameter and *parameter_value* is the value you choose.

Note: A value set at the command prompt is valid for only that command prompt session. Any Oracle software you run during that session   recognizes the values set at the command prompt, and does not search for corresponding values in the Windows 95 Registry.

{button ,AL(`configuration parameters')}   Related Topics

General Configuration Parameters

The following parameters define characteristics of Personal Oracle7 and other Oracle products. Default means a value that various tools will use unless the tool finds a different value in the registry.

NLS_LANG

Operating System:	Windows 95
Default:	automatically set when you select a language during installation
Legal Values:	supported languages, territories, and character sets
Modify?	yes (but only with the installer)

This parameter includes three components:

- language
- territory
- character set

This parameter specifies the following information.

- language in which to display the messages
- format in which to display the date
- character set displayed.

The default value causes all messages to be displayed in English.

Note: Refer to *Oracle7 Server Reference* manual, available online as an Adobe Acrobat-readable file, and to the online Reference Help topic, National Language Support (NLS), for additional information about the NLS_LANG parameter and its components.

ORACLE_HOME

Operating System:	Windows 95
Default:	none (set during install)
Legal Values:	any valid directory on any drive
Modify?:	no

This parameter specifies the Oracle home directory in which Oracle products are installed. It is added to your Windows 95 registry when you first install Oracle products. This is the top directory in the Oracle directory hierarchy.

ORACLE_SID

Operating System:	Windows 95
Default:	none (set to `orcl` during install if the starter database is installed)
Legal Values:	any combination of up to 4 alphanumeric characters
Modify?:	yes

This parameter specifies the name of the Oracle7 DBMS instance on the host machine. The value of this parameter is the system identifier (SID) for the database.

ORA_*sid*_PWFILE

Operating System:	Windows 95
Default:	ORA%ORACLE_SID%.ORA
Legal Values:	any legal filename
Modify?:	yes

This parameter specifies the name of the password file containing the password for the database designated by the system identifier *sid*.

ORA_SQLDBA_MODE

Operating System:	Windows 95
Default:	SCREEN (set to LINE during install)
Legal Values:	LINE, SCREEN
Modify?:	yes

This parameter specifies the mode in which SQL*DBA starts from the command line.

PLSQL*nn*

Operating System:	Windows 95
Default:	none (set to %ORACLE_HOME%\PLSQL*nn* during install)

Legal Values:	any valid path
Modify?:	no

This parameter specifies the directory in which Personal Oracle7 looks for PL/SQL message files. Although it has no default, the Personal Oracle7 installation process sets the parameter as %ORACLE_HOME%\PLSQL*nn*, where *nn* indicates the version of PL*SQL being used.

RDBMS*nn*

Operating System:	Windows 95
Default:	none
Legal Values:	any valid path
Modify	no

This parameter defines the directory in which the Personal Oracle7 database files are located. Personal Oracle7 and the Oracle7 Utilities also use this directory to locate message and resource files. Although it has no default, the Personal Oracle7 installation process sets the parameter as %ORACLE_HOME%\RDBMS*nn*, where *nn* indicates the version of Personal Oracle7 being used.

API

Operating System:	Windows 95
Default:	none (set to %ORACLE_HOME%\dbs during install)
Legal Values:	any valid path
Modify?:	no

This parameter is used by the installer to find resource files.

COMPANY_NAME

Operating System	Windows 95
Default:	none (during install, set per user input)
Legal Values	any string
Modify?:	no

DATABASE_STARTUP

Operating System:	Windows 95
Default:	OFF (set to AUTO during install)
Legal Values:	AUTO or anything else to mean OFF
Modify?:	yes

This parameter is used to indicate whether or not the database should start automatically when a connection is first made to the database.

DBA_AUTHORIZATION

Operating System:	Windows 95
Default:	none (set to ORACLE during install if a database is installed)
Legal Values:	anything
Modify?:	yes

This parameter is used to specify the password for the database. If it matches the one in the password file, you will not be queried for one.

EXECUTE_SQL

Operating System:	Windows 95
Default:	none (set to PLUS32 during install)
Legal Values:	some executable that can interpret/execute SQL scripts.
Modify?:	yes

This parameter is used by tools to determine which application should be used to execute SQL scripts.

LISTENER_STARTUP

Operating System:	Windows 95
Default:	OFF (set to AUTO during install)
Legal Values:	AUTO or anything else to mean OFF
Modify?:	yes

This parameter is used to indicate whether or not the SQL*Net Listener will be started when the database is started.

MSHELP

Operating System:	Windows 95
Default:	none (set to %ORACLE_HOME%\MSHELP during install)
Legal Values:	any valid path
Modify?:	no

This parameter is used by some tools to find help files.

MSHELP_TOOLS

Operating System:	Windows 95
Default:	none (set to %ORACLE_HOME%\MSHELP during install)
Legal Values:	any valid path
Modify?:	no

This parameter is used by some tools to find help files.

NLSRTL*nn*

Operating System:	Windows 95
Default:	none (set to %ORACLE_HOME%\NLSRTL31 during install)
Legal Values:	any valid path
Modify?:	no

ORA_NLS

Operating System:	Windows 95
Default:	none (set to %ORACLE_HOME%\NLSRTL31\DATA during install)
Legal Values:	any valid path
Modify?:	no

This parameter is used to find character set information files.

ORACLE_GROUP_NAME

Operating System	Windows 95
Default:	set during install
Legal Values:	any string
Modify?:	no

PLUS*nn*

Operating System:	Windows 95
Default:	none (set to %ORACLE_HOME%\PLUS32 during install)
Legal Values:	any valid path
Modify?:	no

This parameter is used by SQL*PLUS to find its message files.

PRO*nn*

Operating System:	Windows 95
Default:	none (set to %ORACLE_HOME%\PRO17 during install)
Legal Values:	any valid path
Modify?:	no

This parameter is used by Pro*C and SQLLIB to find message files.

RDBMS_FILES

Operating System:	Windows 95
Default:	none (set to %ORACLE_HOME%\DATABASE during install)
Legal Values:	any directory
Modify?:	no

This parameter is used by the database tools to locate database files for backup and recovery when the database is not running.

SQLPATH

Operating System	Windows 95

Default:	any valid path
Legal Values:	any valid path or paths separated by semicolons
Modify?:	yes

This parameter is used by SQL*PLUS to locate SQL scripts.

Note: Individual Oracle tools can use additional configuration parameters; refer to the documentation for the tool you are using for more information on those parameters.

{button ,AL(`configuration parameters')} Related Topics

Networking Configuration Parameters

LOCAL

Operating System:	Windows 95
Default:	2: (Local Database)
Legal Values:	any valid SQL*Net alias, connect string, or connect descriptor

This parameter specifies the connect string to use when none is otherwise specified. The parameter enables you to define a "default" network connection. When a user tries to connect without specifying a connect string, Personal Oracle7 first tries to connect using the connect string defined by LOCAL (even if that connect string is a remote database). If it cannot find LOCAL, it defaults to the Local Database.

The default connects the user to the Local Database instance designated by the ORACLE_SID parameter on the local machine.

Note: You can reset this parameter on the command line at any time.

NET20

Operating System:	Windows 95
Default:	None
Legal Values:	%ORACLE_HOME%\NETWORK
Modify?:	no

This parameter specifies the SQL*Net Version 2 message files directory.

Note: Individual Oracle tools can use additional configuration parameters; refer to the documentation for the tool you are using for more information on those parameters.

{button ,AL(`configuration parameters')} Related Topics

SQL*DBA: Starting SQL*DBA

The following information applies to SQL*DBA under Windows 95. With SQL*DBA, you can perform database administration and execute SQL statements.

SQL*DBA is the tool which provides full and complete administration of your Personal Oracle7 database. You may use SQL*DBA only occasionally or you may not use SQL*DBA at all. However, SQL*DBA is available to you to perform specific database administration tasks. The following is information specific to SQL*DBA on Windows 95.

To start SQL*DBA:

At the DOS prompt type the command:

```
C:\>SQLDBA [MODE=SCREEN]
```

By default SQL*DBA operates in LINE mode. Use the MODE=SCREEN option if you wish to operate SQL*DBA in full screen character mode.

When running in screen mode, SQL*DBA uses the default terminal setting for a BIOS terminal. The key commands for this terminal type are listed below.

Function	**Key Commands**
[Show Keys]	[Ctrl]-K
[Help]	[Alt]-H
[Menu]	[F4]
[Accept]	[Enter]
[Cancel]	[Esc]
[List of Values]	[Ctrl]-L
[Select]	[Space Bar]
[Next Field],[Previous Field]	[Tab],[Shift]-[Tab]
[Rotate Window]	[F3]
[Up],[Down],[Left],[Right]	[Up Arrow],[Down Arrow],[Left Arrow],[Right Arrow]
[Scroll Up],[Scroll Down]	[Page Up],[Page Down]

Note: The key commands do not apply to SQL*DBA operating in line mode. For a complete description of SQL*DBA functions, see *Oracle7 Server Utilities* documentation available online as an Adobe Acrobat-readable file.

SQL*DBA examples shown through out this help documentation use line mode commands, rather than menus and dialog boxes.

{button ,AL(`SQL*DBA')} Related Topics

SQL*DBA: Hosting to Windows 95

SQL*DBA provides a special HOST statement that interrupts the current SQL*DBA session so that Windows 95 commands can be executed. The syntax of the HOST statement is:

```
SQLDBA>HOST [command]
```

or

```
SQLDBA>$ [command]
```

where command is a Windows 95 command.

If you include a Windows 95 command after the HOST statement, SQL*DBA performs that command and immediately returns to the SQL*DBA prompt.

If you do not specify a command, SQL*DBA hosts out to the Windows 95 command prompt. Return to SQL*DBA by entering the command:

```
C:\>EXIT
```

{button ,AL(`SQL*DBA')}   Related Topics

SQL*DBA: Defining Display Characteristics

SQL*DBA lets you define certain display and performance characteristics of the SQL*DBA session. You define these characteristics by using the SET command:

```
SQLDBA>SET parameter value
```

where parameter is the name of the characteristic to define. Many of the SQL*DBA session characteristics have Windows 95 defaults and ranges. See below for a description of these characteristics.

Name	**Windows 95 Default Value**	**Windows 95 Maximum**
ARRAYSIZE	20 rows	100 rows
CHARWIDTH	80 characters	65,535 characters
DATEWIDTH	9 characters	65,535 characters
LONGWIDTH	80 characters	65,535 characters
MAXDATA	20 kilobytes	65,535 characters
NUMWIDTH	10 digits	65,535 digits

Note: For more information on how individual parameters affect the SQL*DBA session, please refer to the *Oracle7 Server Utilities* documentation available online as an Adobe Acrobat-readable file.

{button ,AL(`SQL*DBA')} Related Topics

SQL*DBA: Saving Output from a SQL*DBA Session

SQL*DBA provides a function to record the output of a SQL*DBA session in a text file. The syntax for using this feature is:

```
SQLDBA>SPOOL [filename|OFF]
```

where filename is any valid Windows 95 filename.

If you specify no filename, SQL*DBA creates a file named SQLDBA.LOG in the directory in which you started SQL*DBA. To stop the output and close the file, use the OFF option.

{button ,AL(`SQL*DBA')}   Related Topics

Oracle Call Interface: Overview

The Oracle Call Interface (OCI) is an applications programming interface (API) for accessing an Oracle database from a C program. You make calls directly to the OCI functions from within your C program to direct the execution of your SQL statements.

Oracle supports client DLL implementation for 32-bit Windows applications. The following files are required:

- OCIW32.LIB and OCIW32.DLL for 32-bit applications

Win32s is currently not supported.

In the past, Oracle Client DLLs, with version numbers as part of their filenames, were used to implement the OCI layer. However, a DLL with the correct version number in the filename was not always available on the destination system.

The Oracle Client DLLs for 32-bit Windows applications no longer contain version numbers in their filenames and can dynamically load whatever client DLL is available. This new layer is implemented as a DLL and named OCIW32.DLL for 32-bit Windows applications. The new implementation effects the following items as indicated.

- the OCI interface will not change
- recompilation is not necessary
- no extra function calls are needed

{button ,AL(`OCI')} Related Topics

Oracle Call Interface: Implementation

At OCI application startup, the following occurs:

1. An attempt is made to translate ORAOCI (see Platform Specific Details). If this variable exists, its value is expected to be the name of an Oracle Client DLL. If ORAOCI does not exist, a search is made through a pre-defined list of known Oracle Client DLLs. If no Oracle Client DLLs can be located, an error is raised.
2. A table of function pointers (one for each OCI call) is filled in. Obsolete OCI functions are not immediately loaded since they are most likely not to be used. Instead, these are loaded on demand at runtime. No errors are raised when loading function pointers.
3. When a user calls a function, a check is made to see if a pointer exists. If it does, the function is called. If it does not, an error message informs the user of the application name, DLL, and function being called.

{button ,AL(`OCI')} Related Topics

Oracle Call Interface: Platform-Specific Details

When an OCI based application is loaded, an Oracle Client DLL must be located and loaded. The following directories are searched in the exact order listed.

For 32-bit Windows Applications

1. The directory in which the application is loaded.
2. The current directory.
3. If you are using Windows 95, the Windows system directory is searched.
4. If you are using Windows NT, the 32-bit Windows system directory, SYSTEM32, is searched.
5. For Windows NT only, the 16-bit Windows system directory, SYSTEM, is searched. There is no Win32 function that obtains the path of this directory, but it is searched.
6. The Windows directory.
7. The directories that are listed in the PATH environment variable.

When translating any variables, the following methods are used.

- 32-bit Windows Applications: ORAOCI can be set in the Windows Registration Database (under HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE) to point to the default Oracle Client DLL.

{button ,AL(`OCI')} Related Topics

Oracle Call Interface: Building OCI Applications

You can build OCI applications using the following techniques. You do not need to precompile the source code, because OCI applications contain no EXEC SQL statements.

OCI Header Files

To use OCI, you need to include OCI header files in your programs. The files are as follows:.

OCIAPR.H	Declares and gives prototype information for OCI functions. Use for ANSI C compatibility.
OCIDFN.H	Declares the cursor and logon data area structure. This is a common header file for OCI C sample programs.
OCIKPR.H	Declares the OCI functions. Use for K&R C compatibility.
ORATYPES.H	Defines C types used by OCI sample programs.

OCI Program Calls

WINOCI.C

This program is a GUI version of the sample program found in Pro*C. It adds new employee records to the demonstration personnel database. The program checks the integrity of the database. You can enter as many employee names as you want and perform the SQL commands by selecting the appropriate buttons in the Employee Record box.

The WINOCI program displays a window. Select Connect from the Oracle menu to display the logon dialog box. Type the userid SCOTT, the password TIGER, and the name of Personal Oracle7 (if nothing is entered, this defaults to the local database). Click OK to connect with the Oracle database.

The following lists the buttons and their definitions available in the program:

Select	Displays the first record from EMP
Fetch Next	Displays the next record from EMP
Delete	Deletes the current record
Update	Updates the record after changes to column values
Clear	Clears the display
Insert	Inserts a blank record into the current table
Commit	Commits changes made to the EMP table
Rollback	Undoes changes made to the EMP table since the last COMMIT
Exit	Exits the sample application

{button ,AL(`OCI')} Related Topics

Oracle Call Interface: Compiling OCI Applications

Once you have created an OCI application, you must compile the resulting C source file. To do this, switch to the development environment and build the application.

For 32-Bit Compilations

Use your product's default compiler options.

The Oracle OCI DLL is implemented using the cdel calling convention. Calls to the OCI must be made through this convention. This is the default option for MSVC++ and is specified by using the /GD compiler flag.

{button ,AL(`OCI')} Related Topics

Oracle Call Interface: Linking OCI Applications

For 32-Bit Applications

For linking 32-bit OCI applications, use the files listed below:

Library Name	**DLL Name**
OCIW32.DLL	OCIW32.LIB

Oracle Call Interface: Non-blocking Connections

A local non-blocking connection to Personal Oracle7 is made using a special SQL*Net TCP/IP loopback connection. Two requirements must be met to accomplish the connection:

1. Microsoft TCP/IP must be installed and configured.
2. Personal Oracle7 is installed after MS TCP/IP has been installed and configured.

If the above two requirements have not been met, see the installation documentation for information on configuration of incoming and TCP/IP loopback connections.

Loopback connection:

To make a TCP/IP loopback connection you must connect to Personal Oracle7 as if you were connecting to a remote database. This requires the following:

- You must use a special connect string or database alias.
- The database alias must have “127.0.0.1” as the TCP/IP host name.

The TCP-loopback connection is already configured to connect to the Personal Oracle7 database if you are using SQL*Net Easy Configuration to configure your SQL*Net connections. If you are not using SQL*Net Easy Configuration to configure your SQL*Net connections, consult with your network administrator to set up this SQL*Net configuration.

National Language Support (NLS) Introduction

Personal Oracle7 provides National Language Support (NLS), which allows users to interact with the database in their selected language. When you installed Personal Oracle7 for Windows 95, you selected a language, and the Installer records that language in the NLS_LANG configuration parameter.

Note: For additional information on the NLS_LANG parameter and NLS initialization parameters, refer to Appendix C in the *Oracle7 Server Administrator's Guide*, available online as an Adobe Acrobat-readable file.

NLS: NLS_LANG Parameter

When you install Personal Oracle7, you select a language for messages and text. The Installer supplies a value for the NLS_LANG parameter to specify the language selected.

The NLS_LANG parameter uses the following format:

```
NLS_LANG = LANGUAGE_TERRITORY.CHARACTER_SET
```

where language specifies the language and its conventions for displaying messages and day and month names, territory specifies the territory and its conventions for calculating week and day numbers, and character_set controls the character set used for displaying messages.

The default language at installation is American English. If you accept the default language, the following value is supplied for the NLS_LANG parameter:

```
AMERICAN_AMERICA.WE8ISO8859P1
```

NLS: Territory and Character Set Defaults

The default territories and character sets for each of the languages supported by Personal Oracle7 are listed below.

Language	Territory	Character Set
AMERICAN	AMERICA	WE8ISO8859P1
BRAZILIAN PORTUGUESE	BRAZIL	WE8ISO8859P1
CZECH	CZECH REPUBLIC	EE8MSWIN1250
DANISH	DENMARK	WE8ISO8859P1
DUTCH	THE NETHERLANDS	WE8ISO8859P1
FINNISH	FINLAND	WE8ISO8859P1
FRENCH	FRANCE	WE8ISO8859P1
GERMAN	GERMANY	WE8ISO8859P1
GREEK	GREECE	ELMSWIN1253
HUNGARIAN	HUNGARY	EE8MSWIN1250
ITALIAN	ITALY	WE8ISO8859P1
JAPANESE	JAPAN	JA16SJIS
KOREAN	KOREA	KO16KSC5601
NORWEGIAN	NORWAY	WE8ISO8859P1
POLISH	POLAND	EE8MSWIN1250
PORTUGUESE	PORTUGAL	WE8ISO8859P1
RUSSIAN	CIS	CL8MSWIN1251
SPANISH	SPAIN	WE8ISO8859P1
SWEDISH	SWEDEN	WE8ISO8859P1
TRADITIONAL CHINESE	TAIWAN	ZHT16BIG5
TURKISH	TURKEY	WE8ISO8859P9

NLS: Selecting Alternate Territories and Character Sets

Although the languages have default territories and character sets associated with them, any language, territory, and character set values can be mixed and included in a single NLS_LANG parameter. For messages and text to be displayed in a language, the NLS_LANG values must correspond to a fully installed language.

Refer to Appendix C in the Oracle7 Server Administrator's Guide for additional information about the NLS_LANG parameter and its components.

NLS: NLS_LANGUAGE and NLS_TERRITORY

Any of the language and territory components listed in the Territory and Character Set Default table can be used as values for the NLS_LANGUAGE and NLS_TERRITORY parameters respectively. Also, any of the territory values in the section Parameter Lists can be used for the NLS_TERRITORY parameter.

NLS: Parameter Lists

The following lists include all parameters for the supported languages, territories, and character sets for Personal Oracle7. Use these values independently within the NLS_LANG configuration parameter or use the language parameter to set the default values for territory and character set. Additionally, the language and territory values listed here may be used in the initialization parameters NLS_LANGUAGE and NLS_TERRITORY. Refer to the previous sections "NLS_LANG" and "NLS Initialization Parameters" for information on the usage of the following parameters.

Language

American
Arabic
Brazilian Portuguese
Czech
Danish
Dutch
Finnish
French
German
Greek
Hungarian
Italian
Japanese
Korean
Norwegian
Polish
Portuguese
Russian
Simplified Chinese
Slovak
Spanish
Swedish
Traditional Chinese
Turkish

Territories

America
Brazil
China
CIS
Czech Republic
Denmark
Finland
France
Germany
Greece
Hungary
Italy
Japan
Korea
Norway
Poland
Portugal
Spain
Sweden
Taiwan
The Netherlands
Turkey
United Arab Emirates

Character Sets

AL24UTFFSS
AR8ADOS710
AR8ADOS720
AR8APTEC715
AR8ARABICMAC
AR8ARABICMACS
AR8ASMO8X
AR8ISO8859P6
AR8MSWIN1256
AR8MUSSAD768
AR8NAFITHA711
AR8NAFITHA721
AR8SAKHR706
AR8SAKHR707
BG8MSWIN
BG8PC437S
CDN8PC863
CH7DEC
CL8ISO8859P5
CL8MACCYRILLIC
CL8MACCYRILLICS
CL8MSWIN1251
D7DEC
D7SIEMENS9780X
DK7SIEMENS9780X
E7DEC
E7SIEMENS9780X
EE8ISO8859P2
EE8MACCE
EE8MACCES
EE8MACCROATIAN
EE8MACCROATIANS
EE8MSWIN1250
EE8PC852
EEC8EUROPA3
EL8DEC
EL8ISO8859P7
EL8MACGREEK
EL8MACGREEKS
ET8MSWIN923
EL8MSWIN1253
F7DEC
F7SIEMENS9780X
HU8ABMOD
HU8CWI2
I7DEC
I7SIEMENS9780X
IS8PC861
IW8ISO8859P8
IW8MACHEBREW
IW8MACHEBREWS
IW8MSWIN1255
IW8PC1507
JA16EUC
JA16MACSJIS
JA16SJIS
JA16VMS
KO16KSC5601
KO16KSCCS
LA8PASSPORT
LT8MSWIN921
LT8PC772
LT8PC774
N7SIEMENS9780X
N8PC865
NDK7DEC

NE8ISO8859P10
NEE8ISO8859P4
RU8BESTA
RU8PC855
RU8PC866
S7DEC
S7SIEMENS9780X
SE8ISO8859P3
SF7ASCII
SF7DEC
TH8MACTHAI
TH8MACTHAIS
TH8TISASCII
TR7DEC
TR8DEC
TR8MACTURKISH
TR8MACTURKISHS
TR8MSWIN1254
TR8PC857
US8PC437
WE8DEC
WE8DG
WE8ISO8859P1
WE8ISO8859P9
WE8MACROMAN8
WE8MACROMAN8S
WE8NCR4970
WE8PC850
WE8PC860
WE8ROMAN8
ZHS16CGB231280
ZHT16BIG5
ZHT16CCDC
ZHT16DBT
ZHT32EUC
ZHT32SOPS
ZHT32TRIS

Introduction to Symmetric Replication

Your enterprise probably doesn't operate from a single room, a single building, or even a single continent. Most likely, you have users generating and demanding access to data from points both near and far. You have users working in many separate functional units across your enterprise, users that need computing systems that enable them to work together effectively wherever they might be.

Linking these operations together requires distributed computing systems that allow data to be shared in many different ways. These systems need to support client GUI front-end processing, application processing, and database processing where it makes the most sense to maximize performance and availability. Most of all, distributed computing systems need to be manageable.

Oracle offers the most advanced technologies for distributed computing, including:

- Client/server computing over almost any network or combination of networks
- Wireless communications
- Middleware services such as security and name servers
- Distributed database capabilities supporting both real-time and deferred data sharing
- Gateways to integrate legacy systems into your distributed environment

Summary

Building integrated, effective distributed computing systems is one of the great challenges of this decade. Distributed database technology is a fundamental component of a complete distributed solution. Oracle7 distributed database technology is:

Flexible to let you build distributed solutions that meet your business requirements. Whether your application requires remote access or replication, synchronous real-time operations or asynchronous deferred operations, the ability to query or update data or use procedure calls, the capabilities are there.

Manageable so that distributed systems are practical. Management tools, robust implementations to handle real-world stresses, and automatic and transparent features keep the complexity of distributed operations under control.

Integrated to improve manageability, maximize performance, and ensure compatibility with other database features. Oracle7 distributed technology is built-in-not added-on.

Oracle7 offers the most advanced distributed database technology in the industry. From the mid-1980s when Oracle introduced its first distributed capabilities, to today with the introduction of Symmetric Replication, Oracle has set the pace. Oracle's experience, resources and technical leadership ensure your success both now and in the future.

Symmetric Replication: Distributed Database Technology

Servers use distributed database technology to share data in many different ways. When a local server needs to share data that resides on a remote server, the local server can use distributed database technology to access the data on the remote system directly. Alternatively, the two servers can use distributed database technology to maintain copies, or replicates, of the data on both servers so that both can access it locally.

Added to this is the dimension of time. Do you need to access remote data in real-time or will deferred access suffice? Similarly, when replicated data is updated, do the updates need to be applied to other copies in real-time or can the updates be propagated on a deferred basis?

Real-time remote data access and real-time application of updates to replicated data is provided by synchronous distributed database technology. Deferred remote data access and deferred propagation of replicated data updates is provided by asynchronous distributed database technology. Understanding the tradeoffs between these two technologies is key to determining which to use to solve a particular business problem. Some business problems will require synchronous distributed technology. For others, asynchronous technology will be a better fit. In still other cases, both technologies will be needed to solve different aspects of the problem.

The tradeoffs between these two involve application integrity, complexity, performance, and availability. Synchronous technology ensures application integrity and minimizes complexity, but can be less available if the systems and networks involved are not reliable. It can also incur poor response time if network access between systems is slow. Asynchronous technology maximizes availability and minimizes response time, but can be more complex and requires careful planning and design to ensure application integrity.

The following are some examples of synchronous and asynchronous distributed technology.

Synchronous Distributed Technology Example

Say we have an order entry system and a shipping system. The systems share inventory data using synchronous distributed technology.

Since we are using synchronous technology, both systems have access to the most current, up-to-date inventory information. You could, for example, store a single copy of the inventory data on the shipping system and access the data remotely from the order entry system in real-time. Alternatively, you could use synchronous technology to maintain exact replicas of the inventory data on both systems. Any update to the data on either system would be applied synchronously, or in real-time, to the copy on the other system.

Orders can now be placed knowing exactly what is in inventory. Inventory items can be reserved as an integral part of the order-taking process. Shipments can be made knowing that the resulting decrements to inventory quantities will be immediately seen by the order entry system. In this way, synchronous technology provides simplicity and ensures application integrity.

To place orders or ship products, however, processing needs to take place across the network and on both systems, which may be slow. Also, the systems and network need to be reliable, or availability can suffer. If the shipping system or network goes down, for example, orders can't be placed. If remote access is being used, the order entry system can't access the inventory data on the shipping system. Similarly, if the inventory data is replicated, it can't be updated because the update can't be applied to both systems.

Asynchronous Distributed Technology Example

Now let's use asynchronous distributed technology to share inventory data between our order entry and shipping systems. First, we will place the inventory data on the shipping system. Then we will use asynchronous replication to give the order entry system access to this information.

Updates to the inventory data on the shipping system will be propagated and applied to a copy of the inventory data on the order entry system, but on a deferred basis. This copy will contain accurate information but will not always be current. It may, however, be current enough for many businesses. If inventory levels are high and/or easily replenished, then slightly out-of-date information may suffice. The chances of ordering an item which is out of stock will either be very low or will only cause small shipment delays.

Ensuring that asynchronous distributed technology provides adequate application integrity requires careful consideration. The advantage of asynchronous technology, however, is availability and response time performance. Transactions operate against local data only. They may initiate deferred operations which need to be propagated to other systems, but if these other systems are not available the propagation will be deferred until the systems come back up. In our example, orders can be placed even when the network or shipping system is down.

Symmetric Replication: Selecting Distributed Technology

Building successful distributed computing systems is one of the great challenges of this decade. To ensure success, the distributed database technology that you select should be:

Flexible-Distributed capability must fit your business-not the other way around. Whether an application requires remote access or replication, synchronous real-time operations or asynchronous deferred operations, the ability to query or update data or use procedure calls, the capabilities need to be there.

Manageable-Functionality without the ability to manage it is not usable. To minimize complexity, distributed operations should be as automatic and transparent as possible. Implementations should be robust to handle real-world stresses. Powerful management tools must be available.

Integrated-Distributed capability should be an integral component of the server. No additional, external components should be required that must be separately configured and administered. No performance penalty should be incurred requiring shared data to pass through extra components (extra "hops") before it reaches the database where it is needed. No incompatibilities between the server and the external components should limit functionality.

Symmetric Replication: Oracle7 Synchronous Capability

Oracle began offering synchronous distributed features in the mid-1980s. As Oracle's experience with distributed computing grew, Oracle's synchronous technology matured, culminating in the release of Oracle7. Oracle7 synchronous distributed technology is:

Flexible-Oracle7 synchronous distributed technology meets the widest range of business needs.

Distributed Queries-Data on multiple databases can be queried using the full functionality of the SQL standard SELECT statement. This includes selection, join, aggregation, and sorting operations optimized for maximum performance in a distributed environment.

Distributed Transactions-Data on multiple databases can be modified using the full functionality of SQL standard UPDATE, DELETE, and INSERT statements operating as transactions to ensure that either all modifications on all databases complete successfully or are all rolled back should failures occur.

Remote Procedure Calls-Oracle7 servers can execute remote PL/SQL procedures on other servers. The remote procedure execution operates within the same transaction again ensuring that either all modifications complete or are all rolled back.

Synchronous Replication-PL/SQL triggers can apply all modifications to tables in one database directly to replicate copies of those tables in other databases. All modifications are applied within the same transaction to ensure exact, point-in-time consistency of all copies.

Manageable-Oracle7 synchronous distributed operations are automatic, robust and transparent to minimize complexity.

Location Transparency-Applications can access data and execute procedures remotely as easily as they do locally. No special coding is required to specify data or procedure location. Applications merely specify data and procedures using logical names. Mapping from logical names to physical locations is done transparently. Data and procedures can be moved from one database to another without modifying application code.

Global Database Naming-Data can be uniquely identified and located in the distributed environment. External name services eliminate the burden of maintaining data location information within the database.

Commit Transparency-Applications can execute distributed transactions across multiple databases as easily as local transactions. The SQL standard COMMIT statement commits both local and distributed transactions transparently. No special coding is required.

Robust Protection Against Failures-Systems will fail and networks will fail. Oracle7 protects the integrity of distributed transactions automatically using a robust two-phase commit protocol. Complex two-phase commit logic does not need to be coded into applications.

XA Compliance-XA-compliant TP monitor facilities can be used to coordinate distributed transactions.

Status Monitoring-The status of distributed transactions can be easily obtained and monitored through standard data dictionary tables within the database using Oracle Server Managerª and other tools.

Integrated-Oracle7 distributed technology is built into the Oracle7 server.

Single Server-No additional, special servers need to be installed and maintained.

Direct Server-to-Server Communications-Distributed operations are performed automatically through direct server to server connections. Applications do not need additional connections to external servers to perform operations such as distributed queries. Distributed data access does not need to pass through extra servers impacting performance.

Compatibility-Operations such as procedure calls operate under the same transactional protections whether they operate locally or remotely.

Symmetric Replication: Oracle7 Asynchronous Capability

Oracle introduced its first asynchronous distributed capability in the initial Oracle7 release in late 1992. This feature, called read-only snapshots, provides a basic asynchronous replication capability. One site, the snapshot master, can be updated. All other replicates or snapshots are read-only. Incremental row changes are propagated on demand or at time-based intervals using a fast refresh mechanism. The snapshot refresh group feature ensures transactional consistency to maintain referential integrity between multiple snapshots. Read-only snapshots are also very easy to create and administer.

Oracle7 symmetric replication has the following features:

Basic and Advanced Replication Support-Applications can implement primary-site, dynamic, and shared-ownership models as well as fail-over configurations (see below).

Full Transactional Consistency-The referential integrity of replicated data is ensured.

Automatic Update Conflict Detection and Resolution-Resolution routines can be selected declaratively from a set of predefined standard routines such as most recent timestamp or site priority. Users can also define their own customized resolution rules.

Full and Subset Table Replication-All rows or only selected rows in a table can be replicated.

Event-and Demand-Based Replication Methods-Replicated row changes can be efficiently propagated either immediately or when they are demanded by the target system.

Deferred Remote Procedure Calls-Remote PL/SQL procedures can be executed in an asynchronous, or store-and-forward, manner.

Manageable-Oracle7 Symmetric Replication provides powerful management tools integrated into the database and as outside GUI-based client applications.

Replication Catalog-Provides a single, consolidated repository of meta data that defines the distributed/replicated environment, i.e., what database objects (tables, procedures, triggers, indexes, etc.) are replicated where and how they are being replicated. The replication catalog is itself replicated to multiple sites to ensure high availability and easy local access to authorized users.

Distributed Schema Management-Allows replicated environments to be defined and changed automatically at multiple sites by replicating and applying data definition language (DDL) commands. For example, operations such as adding an index or check constraint to a table everywhere it is replicated can be done automatically without tedious and error-prone manual operations.

SNMP Replication MIB-Allows system monitoring tools supporting the SNMP industry standard to monitor Oracle7 symmetric replication; defined as a Management Information Base (MIB) which extends the standard RDBMS MIB jointly defined by Oracle and other database vendors.

Oracle Server Manager-New Oracle Server Manager extensions provide an easy-to-use, GUI-based administration capability for Oracle7 Symmetric Replication. Administrators can easily query the replication catalog, examine internal replication engine components, initiate distributed schema management operations, and troubleshoot problems. Server Manager can be launched in context from system monitors and other tools.

SMTI Member Administration Tools-Additional distributed systems management tools supporting Oracle7 Symmetric Replication are available from Oracle partner companies supported through Oracle's SMTI (System Management Tools Initiative) program. Oracle7 Symmetric Replication users can continue to use the monitors and other administration tools they already have deployed.

Integrated-Oracle7 Symmetric Replication is built into the Oracle7 server as an internal, integrated facility to further improve manageability, maximize performance, and ensure compatibility with other database features.

Integration further improves manageability:

Single Server-No additional, external components must be configured, monitored and maintained.

Standard Components-Oracle7 Symmetric Replication is implemented using proven database components such as tables, views, and PL/SQL procedures that Oracle users are already familiar with.

Standard Backup/Recovery-Oracle7 Symmetric Replication is protected by Oracle7's standard backup and recovery mechanisms. No additional complex procedures involving extra external components are required. Database systems can be backed-up and recovered separately without the need to synchronize operations with other sites.

Integration maximizes performance:

Efficient Implementation-Oracle7 Symmetric Replication uses built-in, high-performance database components such as stored procedures and triggers to capture, propagate, and apply replicated row changes.

Direct Server-to-Server Propagation-Replicated data is propagated directly from server to server. There are no intermediary components to pass through, i.e., no extra "hops."

Proportional Processing-Replication processing is proportional to the amount of data that is actually being replicated, not to total system throughput. No "replication cost" is paid when updates are performed against tables that are not being replicated.

No Log Contention-No contention for log access can bottleneck high-throughput systems.

External replication server-based products must process each and every row change written into the log file to determine whether the change needs to be replicated or not. Contention for log access can bottleneck systems even when only a relatively small amount of data is actually being replicated. This problem is exacerbated because the server needs to be modified to write out a substantial amount of extra information for replication that is not needed for recovery purposes.

Integration ensures compatibility with other database features:

National Language Support-Oracle7 Symmetric Replication uses and is fully compatible with the Oracle7 National Language Support facility.

Parallel Data Query-Oracle7 Symmetric Replication is fully supported on systems using Oracle7 parallel data query capabilities.

External replication products only provide national language support and other standard database server capabilities when they reimplement the capability into the external components. Because of this, replication functionality tends to lag server functionality. Incompatibilities may also arise when new capabilities are provided by still other external components. For example, parallel features, such as parallel query capabilities, provided as an external mechanism outside the server may be incompatible with the external mechanisms supporting replication.

Symmetric Replication: Replication Concepts

Replication is a key element of an overall distributed computing solution, but also a demanding one. One particular challenge is understanding the range of potential uses for synchronous and asynchronous replication.

With synchronous replication, all copies of data are kept exactly synchronized and consistent. If any copy is updated, the update is immediately applied to all other copies within the same transaction. Synchronous replication is appropriate when this exact consistency is important to the business application.

With asynchronous replication, copies or replicates of data will become temporarily out of sync with each other. If one copy is updated, the change will be propagated and applied to the other copies as a second step, within separate transactions that may occur seconds, minutes, hours, or even days later. While copies can be temporarily out of sync, over time the data should converge to the same values at all sites.

Ensuring convergence in asynchronous replication environments is critical for nearly every application. What happens, though, if the same data element, e.g., the same column in the same row, is updated at two sites at the same time, or to be more precise within the same replication interval? This is known as an update conflict. To ensure convergence, update conflicts must be detected and resolved, so that the data element has the same value at every site. Alternatively, update conflicts may be avoided by limiting "ownership," or the right to update a given data element, to a single site.

Symmetric Replication: Usage Models

Asynchronous replication usage models are methodologies for conflict avoidance or conflict detection and resolution. Applications employing asynchronous replication use one or more of these methodologies to ensure data convergence. Applications include the logic necessary to understand, for example, that a conflict avoidance methodology is being employed and what data is owned and updatable at a given site. Similarly, if the business requirement demands that ownership be shared across multiple sites, the application needs to take into account the potential for update conflicts and their detection and resolution.

Oracle7 Symmetric Replication is designed to support replication models employing both conflict avoidance and conflict detection and resolution. Unlike other replication products, the underlying replication mechanisms do not limit users to a single replication model. It is called symmetric because all replicates are potentially updatable. Applications employing conflict avoidance models restrict updates to certain sites to prevent conflicts. Optionally, database facilities such as views and triggers can be defined as "guards" to enforce exclusive ownership. For shared-ownership models, Oracle7 Symmetric Replication provides built-in, automatic conflict detection and resolution.

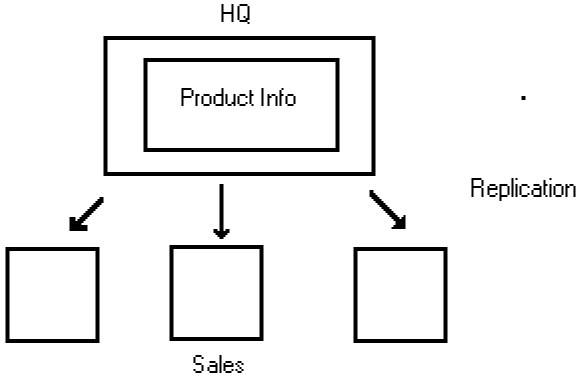
In the following section, we will describe in more detail two methods of conflict avoidance: primary-site ownership and dynamic ownership. We will then discuss the shared-ownership model and issues involving conflict detection and resolution. Finally, we will discuss the use of replication for fail-over configurations which combine elements of both conflict avoidance and conflict detection/resolution. We will provide examples of how each of these models can be applied to solve real business problems.

Primary-Site Ownership

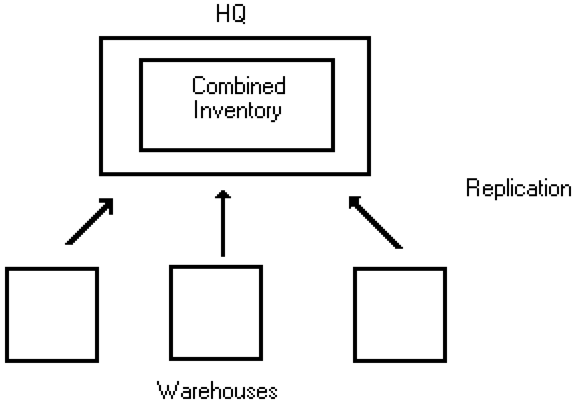
With primary-site ownership, asynchronously replicated data is "owned" by one site. Ownership means that the site may update the data. Other sites "subscribe" to the data owned by the primary site, which means that they have access to read-only copies on their local systems.

Primary-site replication has many uses, for example:

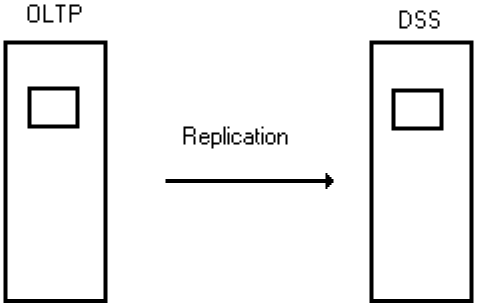
Distribution of centralized information-Product information such as price lists can be maintained at a corporate headquarters site and replicated to read-only copies maintained on order entry systems at remote sales offices.



Consolidation of Remote Information-Inventory data maintained on systems in a number of remote warehouse locations can be replicated to a consolidated read-only copy of the data at a corporate headquarters site.

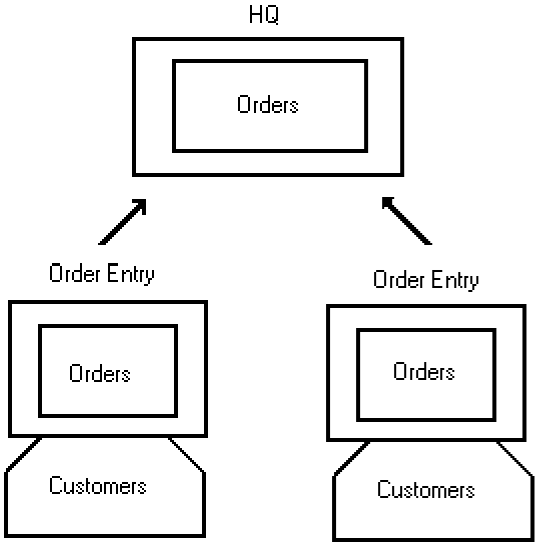


Offloading of OLTP Data for Decision Support System (DSS) Analysis-Data from one or more OLTP systems can be offloaded to a separate, local DSS for read-only analysis.



A primary site may own the data in an entire table, in which case other sites subscribe to read-only copies of all or some subset of that table. Alternatively, multiple sites may own distinct subsets or partitions of the table. Each site might own a distinct set of rows, i.e., a horizontal partition, or a distinct set of columns, i.e., a vertical partition, within a table. Other sites then subscribe to read-only copies of all or some further subsets of the partitions.

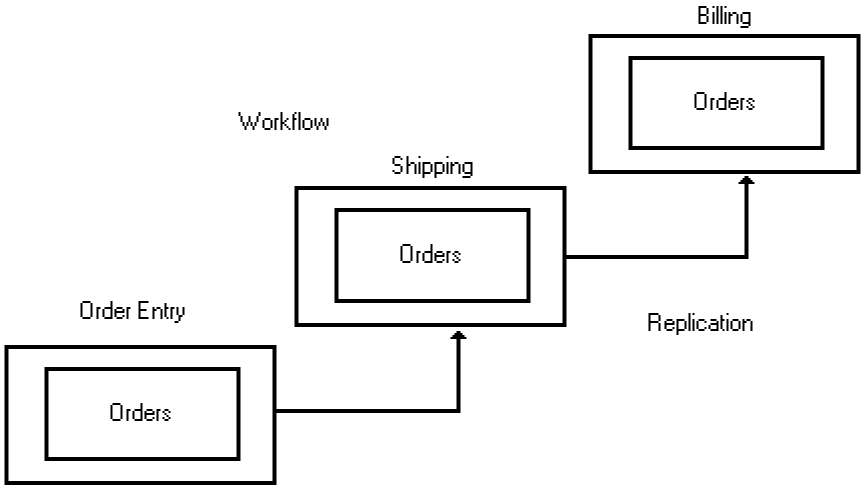
For example, a distributed order entry system could be implemented such that each order entry site in each sales office owns distinct horizontal partitions of tables, e.g., CUSTOMERS, ORDERS, and ITEMS tables, that contain the customer and order information for the customers serviced by each office. A central headquarters site could subscribe to the data owned by each order entry site to maintain a consolidated read-only copy of all orders and customer information across all the sales offices.



Dynamic Ownership

With dynamic ownership, the ability to update asynchronously replicated data moves from site to site while ensuring that only one site may update the data at any given point in time. For example, within an order processing system, the processing of orders typically follows a well ordered series of steps, e.g., orders are entered, approved, shipped, billed, collected, accounted for, and so on. Centralized systems allow the application modules that perform these steps to act on the same data contained in one integrated database. Each application module acts on an order, i.e., performs updates to the order data, when the state of the order indicates that the previous processing steps have been completed. For example, the application module that ships an order will do so only after the order has been entered and approved.

By employing a dynamic ownership replication methodology, such a system can be distributed across multiple sites and databases. Application modules can reside on different systems. For example, order entry and approval can be performed on one system, shipping on another, billing on another, and so on. Order data is replicated to a site when its state indicates that it is ready for the processing step performed by that site. Data may also be replicated to sites that need read-only access to the data. For example, order entry sites may wish to monitor the progression of processing steps for the orders they enter.



Shared Ownership

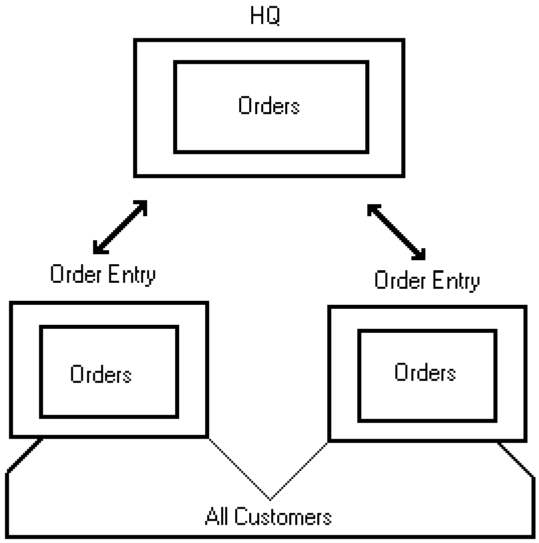
The replication models described thus far, primary site and dynamic ownership, share a common property: at any given point in time, only one site may update the data while the other sites have read-only access to replicated copies of the data.

In some situations, however, it is desirable to allow multiple sites to update the same data, potentially at the same time. For example, it may be desirable to replicate customer data across multiple sites and systems rather than maintain customer data centrally. In this case, different sites may need to update this data.

Update Conflicts

Say we are replicating customer data across sales office order entry sites and headquarters sites. One element of the customer data is the customer address. What happens if a customer's address is changed at both a sales office and a headquarters site at the same time?

This occurrence is known as an update conflict-asynchronously replicated data has become inconsistent because the replicated data can be updated at multiple sites. For some applications, though, temporary inconsistencies can be permitted as long as they can be detected and resolved to ensure that the replicated data will converge to a consistent state at all sites.



Sophisticated Uses of Shared Ownership

Shared ownership allows asynchronous replication to be employed where primary-site and dynamic-ownership models would be too restrictive. In cases where temporary inconsistencies can be permitted and conflict detection and resolution employed, it can offer unprecedented new capability.

For example, earlier we discussed how a distributed order entry system could be implemented using primary-site replication methodologies with horizontal partitioning. In this scenario, each sales office owned a distinct horizontal partition of the tables containing orders and customer information for the customers serviced by each office. Each sales office entered orders for its customers, but no others.

For some businesses, though, this is not the model. For example, a retail chain may have several stores in a metropolitan area. Customers may frequent the store closest to where they live, but go into other stores which will also want to take their orders. If multiple stores perform updates to the same customer and order data, however, update conflicts can occur. Sophisticated application developers can identify these conflicts and either select standard resolution routines or devise their own.

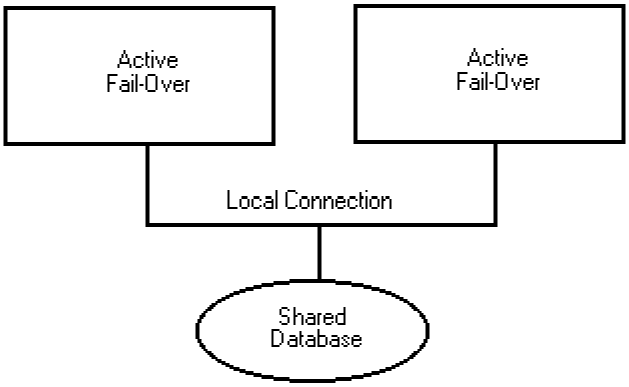
Fail-Over

Oracle7 Symmetric Replication is one of many technologies and approaches that can be employed to support fail-over configurations. A fail-over configuration protects against failures of a primary system by enabling continued processing on a secondary, fail-over system. Asynchronous replication can be used to maintain copies of a primary system's data on a fail-over system. Processing can continue against this replicated data if the primary system fails.

Oracle also provides two other options that should be considered when selecting a fail-over solution. These are: 1) the Oracle Parallel Server,[a] and 2) a standby database configuration. These options offer you alternative tradeoffs in terms of throughput capacity, ease of operation, potential for transaction loss and other data inconsistencies, limitations on the uses of the fail-over system, and the type of failure conditions that can be protected against.

Parallel Server

The Oracle Parallel Server provides fail-over capabilities in locally connected cluster or massively parallel environments. In these environments, it will usually be the preferred option.



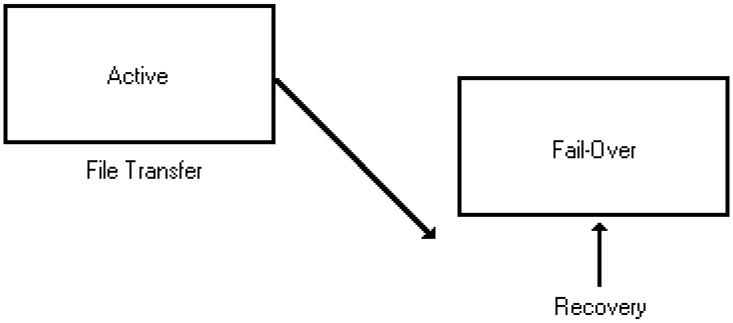
The Oracle Parallel Server allows multiple systems to share common access to a single database maintained on shared disk devices. If a system fails, an Oracle Parallel Server instance on one of the surviving systems will automatically recover any incomplete transactions. Applications that were running on the failed system can then resume processing on a surviving system against the same database.

The Oracle Parallel Server offers high-throughput capacity, ease of operation, and no potential for transaction loss. All systems in an Oracle Parallel Server environment can be fully utilized for both query and update access at any time. Oracle Parallel Server, however, does not provide protection against site failures that may render the entire locally connected environment inoperative, e.g., failures that might be caused by a power outage, fire, flood, or sabotage.

Standby Database Configuration

Another fail-over option is a standby database configuration. This approach protects against site failures and also offers high-throughput capacity. It uses an alternative form of asynchronous replication which employs operating system file transfer facilities to move the log files generated by the primary system to a fail-over site. The log files are then applied to the fail-over system using Oracle recovery mechanisms.

The fail-over system may be located in a geographically remote area to protect against site failures at the primary system's location. High throughput volumes are supported because file transfer does not incur log contention with the primary server and Oracle's recovery mechanisms are highly efficient.

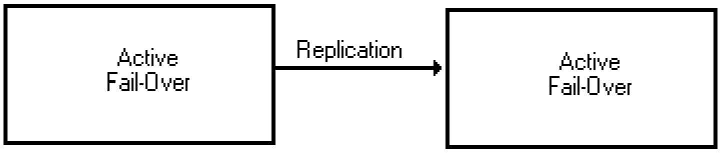


The fail-over system is maintained in a continuous recovery mode which means that it cannot be used for purposes other than fail-

over. Also, if the primary system's site should fail, it may not be possible to transfer the most recently generated log file or files to the fail-over site. Any transactions contained in these log files may be lost.
Oracle currently offers this solution as an Oracle Consulting service. It will be offered as a supported product feature later in 1995.

Oracle7 Symmetric Replication and Fail-Over

Oracle7 Symmetric Replication can also be used in a fail-over configuration to protect against site failures. Its advantage over the standby database approach is that the fail-over system or systems can be used for purposes other than fail-over.



Data that is being updated on the primary system can be queried on the fail-over system at any time, e.g., for reporting or decision support analysis. The same data on both the primary and fail-over systems can also be updated at any time if the two systems are employing a shared-ownership replication model with full update conflict detection and resolution.

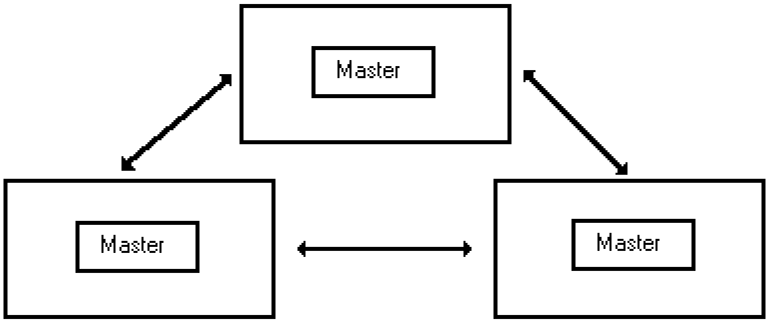
The throughput capacity is less than the standby database approach because Oracle7 Symmetric Replication does not use the Oracle recovery mechanisms to apply replicated data changes. Symmetric replication, like all fail-over solutions based on any form of asynchronous replication, also has the potential for transaction loss in some failure scenarios. Because of its asynchronous or store-and-forward nature, it is possible for replicated changes to have been stored on the primary system but not yet forwarded to the fail-over site when the failure occurs. These stored transactions may not be recoverable.

Symmetric Replication: Configurations

Oracle7 Symmetric Replication supports replication of both full tables and subsets of tables through two mechanisms: multiple masters and updatable snapshots. These two mechanisms can be combined in hybrid configurations to meet different needs.

Multiple Masters

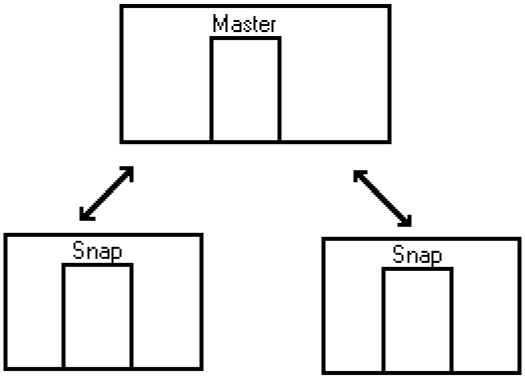
Multiple master replication supports full table, peer-to-peer replication between master tables. Master tables at all sites can be updated. Changes applied to any master table are propagated and applied directly to all other master tables. Failures of any one master site will not block propagation of changes between other master sites.



Multiple master replication uses deferred RPCs (described below) as the underlying transport mechanism to propagate and apply changes. Changes to multiple master tables are applied in a transactionally consistent manner to ensure data and referential consistency. Changes are propagated either immediately in an event-based manner, or at specified points in time when connectivity is available or when communications costs are lowest, e.g., during evening hours. If a remote system is unavailable, the deferred RPCs propagating changes to that system remain in their local queue for later execution.

Updatable Snapshots

Oracle has extended the initial Oracle7 snapshot mechanism to support Symmetric Replication. Snapshots, as well as the snapshot masters, can now be updated. Updates to snapshots are propagated and applied to snapshot masters using deferred RPCs as the underlying mechanism.

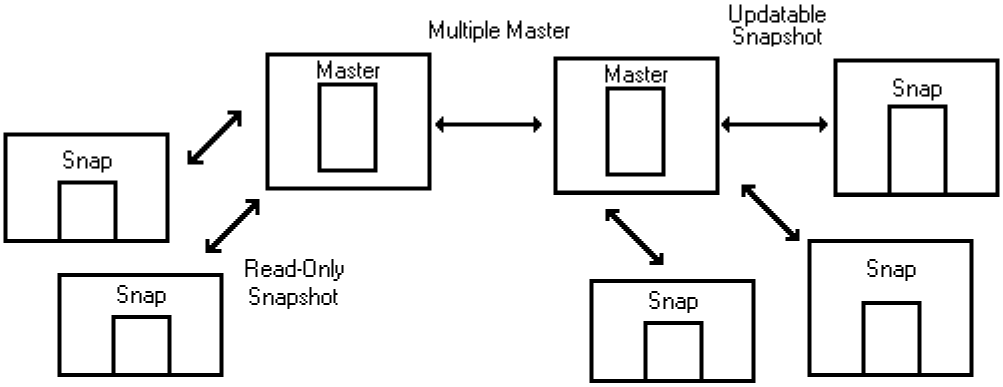


Snapshots can be defined to contain a full copy of a master table or a defined subset of the rows in the master table that satisfy a value-based selection criterion. Snapshots are refreshed from the master at time-based intervals or on demand. Any changes to the master table since the last refresh are then propagated and applied to the snapshot. Multiple snapshots are refreshed in a transactionally consistent manner to ensure data and referential integrity.

Hybrid Configurations

Multiple master replication and updatable snapshots can be combined in hybrid configurations to meet different needs. Specifically, snapshot masters can be replicated in multiple master configurations. This allows full-table and table subset replication to be combined in one system.

For example, multiple masters can replicate between two hub sites supporting two geographic regions. Read-only or updatable snapshots can be defined on the masters to replicate full tables or table subsets to sites within each region. This configuration allows the two master sites to function as fail-over sites for each other. An added benefit of this configuration is that snapshots can be re-mastered from the other hub site to provide an added measure of high availability.



Conflict Detection and Resolution

Oracle7 Symmetric Replication automatically detects update conflicts and invokes user specified conflict resolution routines to restore consistency and ensure data convergence. Conflicts are automatically detected by comparing the before image of the originally modified data to the current values of the data at replicate sites. If the values are different, a conflict has occurred.

Oracle7 Symmetric Replication includes a set of predefined, standard conflict resolution routines that users can declaritively select. Optionally, users can define their own customized routines. Predefined, standard routines include:

- Latest timestamp
- Minimum
- Earliest timestamp
- Maximum
- Priority group
- Average
- Site priority
- Overwrite
- Additive
- Discard

Users can define their own customized resolution routines using PL/SQL procedures. These customized routines can implement more application specific business rules. They might also invoke notification routines, e.g., generate e-mail messages or inserts into a transaction history table to notify users that a conflict occurred and how it was resolved.

Conflict resolution can be configured to invoke multiple resolution routines based on user-specified priorities. For example, a resolution routine could be specified to always execute first. If that routine is not able to resolve the conflict due to special circumstances that it detects, another routine can be specified to run second to handle those special circumstances.

Conflict detection and resolution is needed for the shared ownership replication model discussed above. It is also needed for dynamic ownership and fail-over models in certain configurations.

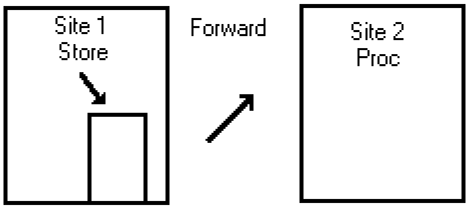
Column Groups

Conflict detection and resolution is based on column groups. A column group is a collection of logically related columns within a row that need to be dealt with as a unit to ensure consistency. An example would be a set of columns used to contain addresses in a customer table. If address information needs to be updated at multiple sites, a latest timestamp resolution method could be selected to ensure that the most recent address change is maintained should conflicts occur.

A row can have multiple column groups. For example, a column in the same customer table might contain information about customers' outstanding credit balances. This column may be incremented by a billing system and decremented by an accounts receivable system at different sites. Here, the additive resolution method could be used to resolve conflicts.

Symmetric Replication: Deferred RPCs

Deferred RPCs are a flexible, general purpose facility. They are used as a propagation mechanism for replication. The facility is also available for direct use to enable calls to remote PL/SQL procedures to be processed in an asynchronous or store-and-forward manner.



A local transaction initiates the execution of a deferred RPC by submitting a request to a local queue. Submission into the queue is done within the local transaction. Entries in the queue are then pushed to their target location(s) and executed as a second step within separate transactions.

If a remote system is unavailable when the deferred RPC queue is pushed, the entries for that target system remain in the queue for later propagation. The deferred RPC queue is durable, protected by the backup and recovery mechanisms of the Oracle7 server. This guarantees that the request will not be lost and can be propagated and executed when the target system becomes available.

Unlike other Oracle's vendors' offerings, deferred RPCs can be easily targeted to one or multiple remote systems. Also, multiple deferred RPCs submitted with the same local transaction are remotely executed together within the same transaction without requiring special coding.

Using Deferred RPCs

Deferred RPCs have many uses. Earlier we discussed how asynchronous replication could be used to propagate orders from an order entry system to a remote shipping system. Replication maintains copies of the orders on both systems. For some businesses, though, it may not be necessary to maintain copies of orders on the order entry system. All that is needed is a reliable, asynchronous mechanism to forward the orders to the shipping system.

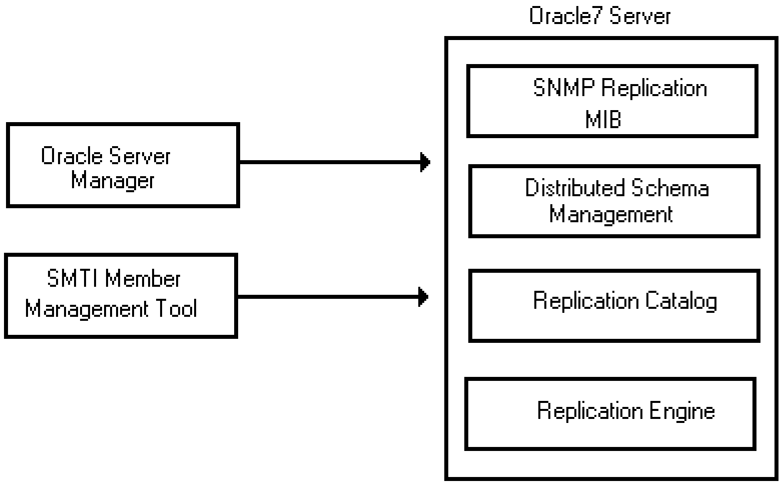
For example, a department store chain may perform all order processing, inventory, billing, distribution and other functions from a central processing location. At each store, a mechanism is needed to forward information about purchases to this central facility. There is no need, however, to maintain copies of this purchase information locally in each store. In such a situation, purchases could be forwarded from the store to the central facility using deferred RPCs. The deferred RPC mechanism guarantees delivery of the purchase information which is all the functionality that is required.

Procedural Replication

Deferred RPCs are also used for procedural replication. Occasionally it may be necessary to perform large numbers of updates in a serial, batch-oriented manner to replicated data. For example, perhaps once a quarter or once a year it may be necessary to purge old order data that may no longer need to be kept online. Replicating each individual row change across multiple sites could be very inefficient. Instead, a procedure can be executed at each replicate site to perform the updates directly. The distributed schema management facility (discussed in the next section) provided with Oracle7 Symmetric Replication allows you to easily set up and maintain remote procedures at multiple sites for procedural replication.

Symmetric Replication: Management Tools

Management tools are critical to successful distributed computing. Advanced replication functionality without the ability to manage it is not usable. Oracle provides management tools integrated into the database and as outside GUI-based client applications as is appropriate for the functionality they provide. Open interfaces and Oracle's SMTI program enable other vendors to enhance their management tool products to support Oracle7 Symmetric Replication.



Replication Catalog

The replication catalog provides a single, consolidated repository containing the meta data that defines the replicated environment. The replication catalog is itself replicated to multiple sites to ensure high availability and easy local access by authorized users wherever they may be located.

The replication catalog defines the database objects being replicated, the sites where they are replicated, and the mechanisms used to support their replication. Database objects include the tables that contain replicated data and the object definitions for other supporting components including indexes, views, procedures, triggers, and synonyms. An extension to Oracle's standard data dictionary, this meta data is open and accessible. It can easily be queried using standard SQL.

Other vendors who provide replication using an external replication server-based implementation do not provide a single, consolidated repository. Meta data and control information is scattered across multiple separate external servers, providing no system-wide view. Each replication server only maintains the information necessary for its operation.

Distributed Schema Management

Oracle7 Symmetric Replication's distributed schema management capability allows replicated environments to be defined and changed from a single control point. It automatically replicates and applies data definition language (DDL) commands to multiple sites. It also automatically generates the underlying mechanisms using database triggers and procedures to support replication.

Distributed schema management allows you to perform operations such as:

- Add an index or check constraint to a replicated table everywhere
- Add all the database objects in a replicated schema to a new site
- Alter a procedure or view definition everywhere

Distributed schema management operations are performed and controlled from one site called the master definition site. It automatically pushes DDL to all master sites and allows snapshot sites to pull down and apply DDL on demand. Because it uses and maintains the replication catalog, which is itself replicated to multiple sites, the master definition site can be changed in case of failure. Distributed schema management operations are initiated by GUI-based management tools from Oracle or other vendors through an open procedural API.

With other replication products, distributed schema management is a manual operation requiring system administrators to connect to multiple databases, replication servers, and other external components to execute DDL and configuration procedures. Configuration and maintenance of distributed schemas can involve hundreds of such operations. When done manually, this is complex, tedious, and highly error prone.

SNMP Replication MIB

Oracle7 Symmetric Replication can be monitored using the industry-standard Simple Network Monitor Protocol (SNMP). The Symmetric Replication MIB, or Management Information Base, extends the standard RDBMS MIB jointly defined by Oracle and other database vendors. SNMP-based tools such as HP's OpenView, Sun's SunNet Manager, and IBM's NetView/6000 can be used to monitor replication, database, operating system, and network activity from a single console. You can continue to use the monitoring tools you already have in place. Other replication products require that you install and maintain an additional monitoring system just for replication.

Oracle Server Manager

Oracle Server Manager provides an easy-to-use, GUI-based administration capability for Oracle7 Symmetric Replication as an extension to its capabilities for general database administration, and can be launched in context by other administration tools including system monitors.

Oracle Server Manager allows system administrators to easily:

- Query the replication catalogs and display SNMP replication MIB status information
- Examine internal replication engine components such as the deferred RPC queue and error log
- Initiate distributed schema management operations and monitor their execution across multiple sites
- Troubleshoot problems

SMTI Member Administration Tools

Additional distributed systems management tools are available from or under development by third-party management tool vendors. Oracle supports and encourages these third-party providers through its SMTI program. Oracle's SMTI program includes over 200 products from over 75 vendors, providing a wide range of administrative capabilities for Oracle.

The SMTI program fosters a close working relationship between Oracle and providers of complementary administration tools by providing:

- Early access to new product releases
- Technical assistance and training
- Cooperative development partnerships
- Joint support arrangements providing customers with a single point of contact

Oracle7 Symmetric Replication was designed to support open interfaces to further encourage SMTI tool development. These include:

- The SNMP replication MIB
- The Oracle7 Symmetric Replication procedural API for distributed schema management
- Standard SQL query access to the replication catalogs
- Standard SQL query access to underlying replication engine components such as the deferred RPC queue and error log implemented using standard relational tables

Symmetric Replication: Packaging

Oracle7 Replication is available in two packages. The first is the read-only snapshot feature which provides basic replication capabilities and is available with the Oracle7 Distributed Option. The second is the full capability of Symmetric Replication, as described herein, which is available with the Oracle7 Advanced Replication Option. The Advanced Replication Option requires the Distributed Option as a prerequisite. Oracle also recommends the Oracle Education and Consulting services available for Symmetric Replication.

Personal Oracle7 Security: Database Security

The Personal Oracle7 Server has many levels of security to keep stored data secure in a Personal Oracle7 database. Personal Oracle7 does not have the same level of security as data stored in an Oracle7 Server running on server level operating systems such as Novell NetWare.

When you first install Personal Oracle7 and the starter database you do not need to provide passwords. The starter database of Personal Oracle7 has been configured where the password is provided for you. To prevent others from accessing the database you should secure the database.

Secure your database through password protection. Password protection will prevent others from starting up or shutting down the database. Password protection also allows you to change the passwords on user accounts that only you know.

Personal Oracle7 Security: Database Password

Database password protection is provided through assignment of a password for a database.   A database password is required to startup and shutdown the database.   The starter database password has been provided for you.

To change the database password:

1. Delete the existing password file of the database. The starter database password file is PWDORCL.ORA.
2. Create a new password file using the password command line utility, orapwd72.

After changing the database password you may want the option of not providing a password each time you start your database. To start the database without providing the *new password* complete the following:

In the /HKEY_LOCAL_MACHINE/SOFTWARE/Oracle folder of the Windows 95 Registry, set the value DBA_AUTHORIZATION to the *new password*. To access the Windows 95 Registry:

1. On the Windows 95 taskbar, click the Start button.

1. Click Run. The Run dialog box appears.

1. Type `regedit` in the Open field.
2. Press Enter.

Note: Performing this function will make your database less secure.

Personal Oracle7 Security: User Account Password

The user accounts that come with the Personal Oracle7 starter database and there passwords are:

User Name	User Password	Roles
SCOTT	TIGER	CONNECT and RESOURCE
SYSTEM	MANAGER	DBA
SYS	SYS [Windows 95 product ID number excluding dashes]	CONNECT, RESOURCE, DBA, EXP_FULL_DATABASE, and IMP_FULL_DATABASE
DEMO	DEMO	CONNECT and RESOURCE
PO7	PO7	DBA

The Windows 95 product ID number can be found by clicking the Start button, Settings, Control Panel, then double clicking the System application. The product id number is on the General page under Registered to:

Each of these passwords should be changed. You can use the Personal Oracle7 Navigator to change these passwords. To change the user account passwords perform the following steps:

1 In the Oracle7 Navigator window, right-click the user whose password you want to change.

2 Click Properties.

3 Type the new password in the New field.

4 Type the new password again in the Confirm field.

5 Click OK.

Note: By default the navigator stores the password for the PO7 account in the Windows 95 Registry. In the property sheet of the local database folder, you must remove the check from the save password box.

Glossary

2:

Alias or connect string for the local database.

access privileges

Ability to connect to, read and/or write to a database. Access privileges are granted through user accounts. See also roles, user accounts.

account

An authorized user of an operating system or a database. Accounts are usually created and controlled by a system (or database) administrator. An account consists of a user name and a password.

active database connection

A database connection that is currently being used to access (connect to) a database. In Personal Oracle7, this is done when you select a database connection that you previously defined in the Database Connection folder in the Personal Oracle7 Navigator, or when you select a database link.

advanced distribution features

Database Links, Distributed Queries, Distributed Updates, Read-Only Snapshots, and Two-Phase Commits. See also distributed option.

alias

An alternative name for an existing network object, such as a host (server), or a set of parameters. In SQL, a temporary name assigned to a table, view, column, or value within a SQL statement, used to refer to that item later in the same statement.

Application Programming Interface (API)

A set of program functions or calls that allow an application to make use of, or communicate with, an underlying program or system.

ASCII data file

Plain text file. ASCII stands for American Standard Code for Information Interchange character set and is a convention for representing alphanumeric information using digital data.

backup

Creation of a copy (or copies) of existing data, to enable recovery of that data when subsequently required.

C:\ORAWIN

The default Oracle Home directory for Oracle products designed to run on the Windows 3.1 operating system.

C:\ORAWIN95

The default Oracle Home directory for Oracle products designed to run on the Windows 95 operating system.

C:\ORAWIN\DATABASE

The default location of the starter database files for Personal Oracle7 for Windows 95.

client

A user, software application, or computer that requests the services, data, or processing of another application or computer (the "server").

cluster

A database structure that contains one or more tables that all have one or more columns in common. Rows of one or more tables that share the same value in these common columns are physically stored together within the database.

column

A vertical space in a database table that represents a particular domain of data. A column has a column name and a specific datatype. For example, in a table of employee information, all of the employees' dates of hire would constitute one column.

column name

Alphanumeric designator of a column and the data it contains.

communications protocol

A set of implemented standards or rules governing the transmission of data across a network. A communications protocol includes detailed specifications for data transmission.

complex query

Also called a complex statement, an INSERT, UPDATE, DELETE, or SELECT statement that contains a subquery. The results of the subquery are used for further processing within the statement.

connect

1) A role that grants access to the Oracle database. 2) The act of accessing an Oracle database. 3) A SQL statement used to access an Oracle database.

connect descriptor

A specially formatted description, usually identified by an alias, of the destination for a network connection. Connect descriptors are used to connect to a database. Connect descriptors are constructed using a set of keywords and values. They are mapped to service names to provide more convenient reference.

connect string

The set of parameters, including a protocol, that SQL*Net uses to connect to a specific Oracle instance on the network.

data replication

Creating a copy of information from various sites of a distributed database in order to make access to the data faster for local clients.

database

A collection of data that is treated as a unit. Composed of logical and physical structures, databases are designed to store and retrieve related information.

database administrator

A person responsible for the operation, maintenance, and customization of an Oracle7 Server.

database application

A set of forms, menus, reports, and other components that satisfies a particular business function. An example of an application is an order entry system.

database connection

Access to the database and its data, accomplished by logging on to the database.

Database Connection

A folder in the Personal Oracle7 Navigator used to hold previously defined database connection. When you click on a previously defined connection, you make an active database connection.

database instance

A particular Oracle7 database such as the Starter database. Personal Oracle7 only supports one database instance. There is always a one-to-one correspondence between an Oracle instance and a set of Oracle data files.

database link

A database link is an object in the database that allows you to access objects on a remote database.

database object

A database object is anything defined and stored in a database. Database objects consist of tables, views, indexes, synonyms, database links, roles, snapshots, and users. Each of these objects define certain characteristics of your database project.

database owner

Usually the creator of a database, the person who can grant database privileges to other users.

database parameters

List of instance configuration particulars required to start an instance. See also instance.

database role

A database role is a method of granting access privileges. You create a database object called a role, then grant privileges to that role, then grant that role to individual users.

database server

The computer starting and running an Oracle database. Client machines connect to the database on the database server.

datatype

A standard form of data. The Oracle datatypes are CHAR, VARCHAR2, NUMBER, DATE, LONG, RAW, LONG RAW; and ROWID.

distributed database

A single, logical database that is physically located on two or more computers, connected via some form of communications network. An essential feature of a true distributed database is that the user and/or program work as if they had access to the whole database locally. All processing to give this impression is carried out by the database management system. See also distributed processing.

distributed option

Permits you to access data residing on different computers in different locations. The distributed option includes: Database Links, Distributed Queries, Distributed Updates, Read-Only Snapshots, and Two-Phase Commit.

distributed query

A query that selects data from databases on multiple nodes (database servers) of a network. A distributed query may reference data on nodes other than the local node, using joins, nested queries, or views. See also distributed database.

distributed update

An update that modifies data in a local or remote database in one transaction.

domain name

Identifier of a domain (A domain is a division of names guaranteed to be unique. Domains are related hierarchically in a naming model).

Export Utility

An Oracle utility used to write database data to operating system files external to the database. These files can then be used with the Import utility.

foreign key

One or more columns in one table whose values refer to the primary key values in another table. See also primary key.

foreign data source

Non-Oracle databases and other files containing data you want to load into an Oracle database.

host

A computer that provides a shared resource on a network.

host string

Also called connect descriptor, a specially formatted description (usually identified by an alias) of the destination for a network connection. Connect descriptors are constructed using a set of keywords and values. They are mapped to service names to provide more convenient reference.

Import Utility

An Oracle utility used to read operating system files that contain database data and were written by Export, in order to restore the data to an Oracle database.

index

A database object created to increase the performance of data retrieval. It provides a fast access path to data in a database. Indexes point directly to the location of the rows containing specified data.

INIT.ORA

A database system file containing a list of parameters that are read when the database system is started. The parameters identify the database and control system-wide resources.

instance

A particular Oracle7 database such as the Starter database. Personal Oracle7 only supports one database instance. There is always a one-to-one correspondence between an Oracle instance and a set of Oracle data files.

local database

The database that has been installed on your computer. It can be the starter database, which is the database that comes with Personal Oracle7, or it can be a database that you have created.

log on

Access a database using your assigned user name and password.

master table

The table you are copying from when you make a snapshot of a table.

migration

For this product, the act of installing Personal Oracle7 for Windows 95 and making your existing data available for use with Personal Oracle7 for Windows 95.

network connection

Electronic connection giving you the ability to access network resources from your local computer.

object

A named structure in the Oracle database, such as a table, index, or form. Also, a group of data in a form that you can copy, move, or delete in a single operation.

Oracle Call Interface

The Oracle Call Interface (OCI) is an applications programming interface (API) for accessing an Oracle database from a C program. You make calls directly to the OCI functions from within your C program to direct the execution of your SQL statements.

Oracle Developer/2000

A development tool that enables organizations to build systems that scale from workgroup to enterprise.

Oracle Named Pipes Protocol Adapter

A SQL*Net application interface that allows SQL*Net to communicate with a remote Oracle database via the Named Pipes Protocol.

Oracle SPX Protocol Adapter

A SQL*Net application interface that allows SQL*Net to communicate with a remote Oracle database via the SPX Protocol.

Oracle TCP/IP Protocol Adapter

A SQL*Net application interface that allows SQL*Net to communicate with a remote Oracle database via the TCP/IP Protocol.

Oracle7 DBMS

The database management system (DBMS) developed by Oracle Corporation. Components of the DBMS include the kernel and various utilities for use by database administrators and database users.

package

A package is an encapsulated collection of related procedures, functions, and other program objects stored together in the database.

parallel database

Multiple database instances that share a physical database.

password

This is the same as a user password. A secret word or phrase (30 character limit; spaces and commas are prohibited) associated with a user name. A password is used for data security and should be known only to its owner.

Personal Oracle7 Navigator

The Personal Oracle7 Navigator is the first screen you see when you open Personal Oracle7. It functions much like the Windows 95 Explorer. You use the Navigator to access all of your projects, database connections, and database objects.

PL/SQL database language

The Oracle procedural language extension of SQL. PL/SQL combines the ease and flexibility of SQL with the procedural functionality of a structured programming language, such as IF ...THEN, WHILE, and LOOP.

primary key

A set of columns used to enforce uniqueness of rows. The combination of column values is unique for each row in the table. The primary key is the most frequently used means of accessing rows. This field is also used to sort your table and to keep you from duplicating records.

privilege

A right to execute a particular type of SQL statement. There are two types of privileges: system privileges and object privileges. System privileges allow the user to perform a particular systemwide action, or to perform a particular action on a particular type of object; for example, the privilege to delete the rows of any table. Object privileges allow the user to perform a particular action on a specific object; for example, the privilege to delete rows in a specific table.

private synonym

A synonym for a database object defined by an Oracle user so that other selected database users can have access to that object. Contrast with public synonym.

procedure

A set of SQL and PL/SQL statements that are grouped together as a unit to solve a specific problem or perform a set of related tasks.

project

A related collection of information located either on the local database or on a remote database. An example of a project could be a Product Project that contains tables, views, indexes, etc. all related to the product that your company manufactures.

public synonym

A synonym for a database object defined by a database administrator so that all database users can have access to that object. Contrast with private synonym.

publish-and-subscribe replication

Part of the distributed option, allows you to replicate data from the master table. Also called Read-Only Snapshot.

read-only snapshot

A copy (known as a replica) of a master table. You can create an unlimited number of read-only snapshots of the table. This snapshot can be periodically refreshed to reflect the most recent state of the master table. During a snapshot refresh, any changes made to the master table since the snapshot was created or previously refreshed are reflected in the snapshot.

relational database

A database composed of tables of interconnected information. Data can be reorganized and re-presented in different ways depending on users' needs.

remote database

A database that is accessible to you via a network connection. Contrast with local database.

replica

A copy of a master table.

replication

A database feature which ensures that updates to copies of any data object are propagated to all other sites. Replication includes read-only snapshots, updatable snapshots, and symmetric replication.

replication master site

A location that contains master copies of information to be updated. Replication master sites are required for updatable snapshots.

role

A named group of related privileges that are granted to users or to other roles.

row

A collection of column information corresponding to a single record.

select

To fetch rows from one or more database tables using a query. Also, the SQL statement SELECT.

sequence

A sequence generates a serial list of unique numbers for numeric columns of a database's tables. For example, if two users want to insert new employee numbers into a table, the sequence automatically generates the correct value for each user.

server

Distributed processing uses more than one processor to divide the processing for a set of related jobs. This division is called client/server architecture, with the client being the front end and the server being the back end. The server runs Oracle software and handles the functions required for concurrent shared data access and it receives and processes SQL and PL/SQL statements originating from client applications.

service name

The name used to identify a SQL*Net server process. The services file on client and server machines maps each service name to a port number and protocol.

shut down

The process of stopping a running instance in order to make a database unavailable, including closing and dismounting a database if one has been mounted and opened. Contrast with start up.

snapshot

Local copy of a remote master table. Snapshots can be used to replicate (copy) all or part of a table, or to replicate the result of a query against multiple tables. Snapshots can either be updatable or read only. Both are copies of the master table. A read-only snapshot can be refreshed periodically to reflect any changes to the master table. Changes made to an updatable snapshot can be propagated to the master table and to any other copies.

snapshot refresh

Changes that have been made to the master table since the snapshot was created or previously refreshed are reflected during refresh.

SQL (Structured Query Language)

A data query, definition, manipulation language for relational databases. It is an internationally accepted standard for relational systems.

SQL*DBA Utility

Oracle utility that allows you to manage the operation of the Oracle7 Server. You can perform a variety of database administration tasks with SQL*DBA.

SQL*Loader Utility

An Oracle utility used to load data – usually ASCII data files – from operating system files into Oracle database tables.

SQL*Net

An Oracle product that works with Oracle Server and enables two or more computers that run the Oracle Server to exchange data through a third-party network. SQL*Net supports distributed processing and distributed database capability. SQL*Net is available in Version 2.

SQL*Plus

A SQL-based language interface for data manipulation, data definition, and the definition of access rights for an Oracle database. Often used as an end-user reporting tool.

starter database

The default database included with Personal Oracle7 for Windows 95. It has the same functionality as Oracle7 databases operating on midrange and mainframe computers, except that it cannot function as a server.

stored function

A set of PL/SQL statements you can call by name. Stored functions are very similar to procedures, except that a function returns a value to the environment in which it is called.

symmetric replication

This allows multiple copies of data to be maintained at different sites in a distributed system.

synonym

An alias for a table, view, sequence, or program unit.

table

The basic unit of storage in a relational database management system. A table consists of one or more units of information (rows), each of which contains the same kinds of values (columns).

table security

Ability to restrict access to a predetermined set of rows and columns of a table.

text editor

A program run by a computer's operating system used to create and edit text files. In Oracle, the text editor is often used to edit SQL files, output files, command files, or control files.

trigger

A stored procedure that is fired (implicitly executed) when an INSERT, UPDATE, or DELETE statement is issued against the associated table.

two-phase commit

A method used by a distributed DBMS (database management system) to ensure that a transaction is valid at all sites by the time it commits (to make a change to data permanent) or rolls back (to restore data to its prior state before the change was made) . All sites either commit or roll back together, no matter what errors occur in the network or on the machines tied together by the network.

updatable snapshot

A local copy of a remote master table that can be updated and can propagate updates to all other copies.

update-anywhere replication

A copy of a master table that can be updated and can propagate updates to all other copies. This is also known as an updatable snapshot.

user

Any person or group of people with some rights to access the database.

user account

Information about a user, including: user name, password, and role. The user account is used to grant access privileges to a user.

user name

The name by which a user is known to the Oracle server and to other users. Every user name is associated with a password, and both must be entered to connect to an Oracle database. See also account.

user password

A secret word or phrase (30 character limit; spaces and commas are prohibited) associated with a user name. A password is used for data security and should be known only to its owner.

view

Customized presentation of data from one or more tables.

2:

Alias or connect string for the local database.

access privileges

Ability to connect to, read and/or write to a database. Access privileges are granted through user accounts. See also roles, user accounts.

account

An authorized user of an operating system or a database. Accounts are usually created and controlled by a system (or database) administrator. An account consists of a user name and a password.

active database connection

A database connection that is currently being used to access (connect to) a database. In Personal Oracle7, this is done when you select a database connection that you previously defined in the Database Connection folder in the Personal Oracle7 Navigator, or when you select a database link.

advanced distribution features

Database Links, Distributed Queries, Distributed Updates, Read-Only Snapshots, and Two-Phase Commits. See also distributed option.

alias

An alternative name for an existing network object, such as a host (server), or a set of parameters. In SQL, a temporary name assigned to a table, view, column, or value within a SQL statement, used to refer to that item later in the same statement.

Application Programming Interface (API)

A set of program functions or calls that allow an application to make use of, or communicate with, an underlying program or system.

ASCII data file

Plain text file. ASCII stands for American Standard Code for Information Interchange character set, a convention for representing alphanumeric information using digital data.

backup

Creation of a copy (or copies) of existing data, to enable recovery of that data when subsequently required.

C:\ORAWIN

The default Oracle Home directory for Oracle products designed to run on the Windows 3.1 operating system.

C:\ORAWIN95

The default Oracle Home directory for Oracle products designed to run on the Windows 95 operating system.

C:\ORAWIN\DATABASE

The default location of the starter database files for Personal Oracle7 for Windows 95.

client

A user, software application, or computer that requests the services, data, or processing of another application or computer (the "server").

cluster

A database structure that contains one or more tables that all have one or more columns in common. Rows of one or more tables that share the same value in these common columns are physically stored together within the database.

column

A vertical space in a database table that represents a particular domain of data. A column has a column name and a specific datatype. For example, in a table of employee information, all of the employees' dates of hire would constitute one column.

column name

Alphanumeric designator of a column and the data it contains.

communications protocol

A set of implemented standards or rules governing the transmission of data across a network. A communications protocol includes detailed specifications for data transmission.

complex query

Also called a complex statement, an INSERT, UPDATE, DELETE, or SELECT statement that contains a subquery. The results of the subquery are used for further processing within the statement.

connect

1) A role that grants access to the Oracle database. 2) The act of accessing an Oracle database. 3) A SQL statement used to access an Oracle database.

connect descriptor

A specially formatted description, usually identified by an alias, of the destination for a network connection. Connect descriptors are used to connect to a database. Connect descriptors are constructed using a set of keywords and values. They are mapped to service names to provide more convenient reference.

connect string

The set of parameters, including a protocol, that SQL*Net uses to connect to a specific Oracle instance on the network.

data replication

Creating a copy of information from various sites of a distributed database in order to make access to the data faster for local clients.

database

A collection of data that is treated as a unit. Composed of logical and physical structures, databases are designed to store and retrieve related information.

database administrator

A person responsible for the operation, maintenance, and customization of an Oracle7 Server.

database application

A set of forms, menus, reports, and other components that satisfies a particular business function. An example of an application is an order entry system.

database connection

Access to the database and its data, accomplished by logging on to the database.

Database Connections

A folder in the Personal Oracle7 Navigator used to hold previously defined database connection. When you click on a previously defined connection, you make an active database connection.

database instance

A particular Oracle7 database such as the Starter database. Personal Oracle7 only supports one database instance. There is always a one-to-one correspondence between an Oracle instance and a set of Oracle data files.

database link

A database link is an object in the database that allows you to access objects on a remote database.

database object

A database object is anything defined and stored in a database. Database objects consist of tables, views, indexes, synonyms, database links, roles, snapshots, and users. Each of these objects define certain characteristics of your database project.

database owner

Usually the creator of a database, the person who can grant database privileges to other users.

database parameters

List of instance configuration particulars required to start an instance. See also instance.

database role

A database role is a method of granting access privileges. You create a database object called a role, then grant privileges to that role, then grant that role to individual users.

database server

The node or computer starting and running an Oracle database. Client machines connect to the database on the database server.

datatype

A standard form of data. The Oracle datatypes are CHAR, VARCHAR2, NUMBER, DATE, LONG, RAW,   LONG RAW; and ROWID.

distributed database

A single, logical database that is physically located on two or more computers, connected via some form of communications network. An essential feature of a true distributed database is that the user and/or program work as if they had access to the whole database locally. All processing to give this impression is carried out by the database management system. See also distributed processing.

distributed option

Permits you to access data residing on different computers in different locations. The distributed option includes: Database Links, Distributed Queries, Distributed Updates, Read-Only Snapshots, and Two-Phase Commits.

distributed query

A query that selects data from databases on multiple nodes (database servers) of a network. A distributed query may reference data on nodes other than the local node, using joins, nested queries, or views. See also distributed database.

distributed update

A update that modifies data in a local or remote database in one transaction.

domain name

Identifier of a domain (A domain is a division of names guaranteed to be unique. Domains are related hierarchically in a naming model).

Export Utility

An Oracle utility used to write database data to operating system files external to the database. These files can then be used with the Import utility.

folder

A directory or subdirectory in which files and other folders can be stored.

foreign key

One or more columns in one table whose values refer to the primary key values in another table. See also primary key.

foreign data source

Non-Oracle databases and other files containing data you want to load into an Oracle database.

host

A computer that provides a shared resource on a network.

host string

Also called connect descriptor, a specially formatted description (usually identified by an alias) of the destination for a network connection. Connect descriptors are constructed using a set of keywords and values. They are mapped to service names to provide more convenient reference.

Import Utility

An Oracle utility used to read operating system files that contain database data and were written by Export, in order to restore the data to an Oracle database.

index

A database object created to increase the performance of data retrieval. It provides a fast access path to data in a database. Indexes point directly to the location of the rows containing specified data.

INIT.ORA

A database system file containing a list of parameters that are read when the database system is started. The parameters identify the database and control system-wide resources.

instance

A particular Oracle7 database such as the Starter database. Personal Oracle7 only supports one database instance. There is always a one-to-one correspondence between an Oracle instance and a set of Oracle data files.

local database

The database that has been installed on your computer. It can be the Starter database, which is the database that comes with Personal Oracle7, or it can be a database that you have created.

log on

Access a database using your assigned user name and password.

master table

The table you are copying from when you make a snapshot of a table.

migration

For this product, the act of installing Personal Oracle7 for Windows 95   and making your existing data available for use with Personal Oracle7 for Windows 95.

network connection

Electronic connection giving you the ability to access network resources from your local computer.

object

A named structure in the Oracle database, such as a table, index, or form. Also, a group of data in a form that you can copy, move, or delete in a single operation.

Oracle Call Interface

The Oracle Call Interface (OCI) is an applications programming interface (API) for accessing an Oracle database from a C program. You make calls directly to the OCI functions from within your C program to direct the execution of your SQL statements.

Oracle Developer/2000

A development tool that enables organizations to build systems that scale from workgroup to enterprise.

Oracle Named Pipes Protocol Adapter

A SQL*Net application interface that allows SQL*Net to communicate with a remote Oracle database via the Named Pipes Protocol.

Oracle SPX Protocol Adapter

A SQL*Net   application interface that allows SQL*Net to communicate with a remote Oracle database via the SPX Protocol.

Oracle TCP/IP Protocol Adapter

A SQL*Net application interface that allows SQL*Net to communicate with a remote Oracle database via the TCP/IP Protocol.

Oracle7 DBMS

The database management system (DBMS) developed by Oracle Corporation. Components of the DBMS include the kernel and various utilities for use by database administrators and database users.

package

A package is an encapsulated collection of related procedures, functions, and other program objects stored together in the database.

parallel database

Multiple database instances that share a physical database.

password

This is the same as a user password. A secret word or phrase (30 character limit; spaces and commas are prohibited) associated with a user name. A password is used for data security and should be known only to its owner.

Personal Oracle7 Navigator

The Personal Oracle7 Navigator is the first screen you see when you open Personal Oracle7. It functions much like the Windows 95 Explorer. You use the Navigator to access all of your projects, database connections, and database objects.

PL/SQL database language

The Oracle procedural language extension of SQL. PL/SQL combines the ease and flexibility of SQL with the procedural functionality of a structured programming language, such as IF ...THEN, WHILE, and LOOP.

primary key

A set of columns used to enforce uniqueness of rows. The combination of column values is unique for each row in the table. The primary key is the most frequently used means of accessing rows. This field is also used to sort your table and to keep you from duplicating records.

privilege

A right to execute a particular type of SQL statement. There are two types of privileges: system privileges and object privileges. System privileges allow the user to perform a particular systemwide action, or to perform a particular action on a particular type of object; for example, the privilege to delete the rows of any table. Object privileges allow the user to perform a particular action on a specific object; for example, the privilege to delete rows in a specific table.

private synonym

A synonym for a database object defined by an Oracle user so that other selected database users can have access to that object. Contrast with public synonym.

procedure

A set of SQL and PL/SQL statements that are grouped together as a unit to solve a specific problem or perform a set of related tasks.

project

A related collection of information located either on the local database or on a remote database. An example of a project could be a Product Project that contains tables, views, indexes, etc. all related to the product that your company manufactures.

public synonym

A synonym for a database object defined by a database administrator so that all database users can have access to that object. Contrast with private synonym.

publish-and-subscribe replication

Part of the distributed option, allows you to replicate data from the master table. Also called Read-Only Snapshot.

read-only snapshot

A copy (known as a replica) of a master table. You can create an unlimited number of read-only snapshots of the table. This snapshot can be periodically refreshed to reflect the most recent state of the master table. During a snapshot refresh, any changes made to the master table since the snapshot was created or previously refreshed are reflected in the snapshot.

relational database

A database composed of tables of interconnected information. Data can be reorganized and re-presented in different ways depending on users’ needs.

remote database

A database that is accessible to you via a network connection.

replica

A copy of a master table.

replication

A database feature in which local copies of remote tables known as snapshots are created.

replication master site

A location that contains master copies of information to be updated. Replication master sites are required for updatable snapshots.

role

A named group of related privileges that are granted to users or to other roles.

row

A collection of column information corresponding to a single record.

select

To fetch rows from one or more database tables using a query. Also, the SQL statement SELECT.

sequence

A sequence generates a serial list of unique numbers for numeric columns of a database's tables. For example, if two users want to insert new employee numbers into a table, the sequence automatically generates the correct value for each user.

server

The provider of services requested by a client.

service name

The name used to identify a SQL*Net server process. The services file on client and server machines maps each service name to a port number and protocol.

shut down

The process of stopping a running instance in order to make a database unavailable, including closing and dismounting a database if one has been mounted and opened. Contrast with start up.

snapshot

Local copy of a remote master table. Snapshots can be used to replicate (copy) all or part of a table, or to replicate the result of a query against multiple tables. Snapshots can either be updatable or read only. Both are copies of the master table. A read-only snapshot can be refreshed periodically to reflect any changes to the master table. Changes made to an updatable snapshot can be propagated to the master table and to any other copies.

snapshot refresh

Changes that have been made to the master table since the snapshot was created or previously refreshed are reflected during refresh.

SQL (Structured Query Language)

A data query, definition, manipulation language for relational databases. It is an internationally accepted standard for relational systems.

SQL*DBA Utility

Oracle utility that allows you to manage the operation of any Oracle database. You can perform a variety of database administration tasks with SQL*DBA.

SQL*Loader Utility

An Oracle utility used to load data – usually ASCII data files – from operating system files into Oracle database tables.

SQL*Net

An Oracle product that works with Oracle Server and enables two or more computers that run the Oracle Server to exchange data through a third-party network. SQL*Net supports distributed processing and distributed database capability. SQL*Net is available in Version 2.

SQL*Plus

A SQL-based language interface for data manipulation, data definition, and the definition of access rights for an Oracle database. Often used as an end-user reporting tool.

starter database

The default database included with Personal Oracle7 for Windows 95. It has the same functionality as Oracle7 databases operating on midrange and mainframe computers, except that it cannot function as a server.

stored function

A set of PL/SQL statements you can call by name. Stored functions are very similar to procedures, except that a function returns a value to the environment in which it is called.

symmetric replication

This allows multiple copies of data to be maintained at different sites in a distributed system.

synonym

An alias for a table, view, sequence, or program unit.

table

The basic unit of data storage in a relational database management system. Every table has a table name and a set of columns and rows in which the data is stored. Each column is given a column name, a datatype (defining characteristics of the data to be entered in the column), and a width (quantity of space to allocate for data to be entered into the column).

table security

Ability to restrict access to a predetermined set of rows and columns of a table.

text editor

A program run by a computer's operating system used to create and edit text files. In Oracle, the text editor is often used to edit SQL files, output files, command files, or control files.

trigger

A stored procedure that is fired (implicitly executed) when an INSERT, UPDATE, or DELETE statement is issued against the associated table.

two-phase commit

A method used by a distributed DBMS (database management system) to ensure that a transaction is valid at all sites by the time it commits (to make a change to data permanent) or rolls back (to restore data to its prior state before the change was made) . All sites either commit or roll back together, no matter what errors occur in the network or on the machines tied together by the network.

updatable snapshot

A local copy of a remote master table that can be updated and can propagate updates to all other copies.

update-anywhere replication

A copy of a master table that can be updated and can propagate updates to all other copies. This is also known as an updatable snapshot.

user

Any person or group of people with some rights to access the database.

user account

Information about a user, including: user name, password, and role. The user account is used to grant access privileges to a user.

user name

The name by which a user is known to the Oracle server and to other users. Every user name is associated with a password, and both must be entered to connect to an Oracle database. See also account.

user password

A secret word or phrase (30 character limit; spaces and commas are prohibited) associated with a user name. A password is used for data security and should be known only to its owner.

view

Customized presentation of data from one or more tables.