

# Überblick

Der Microsoft ODBC-Treiber für Oracle ermöglicht die Verbindung einer ODBC-kompatiblen Anwendung mit einer Oracle-Datenbank. Diese Version bietet verbesserte Leistung und Steuerungsfunktionen, einschließlich Zugriff auf PL/SQL-Pakete, XA/DTC-Integration und Oracle-Zugriff von Internet Information Server (IIS) aus. Der ODBC-Treiber entspricht den Open Database Connectivity (ODBC)-Angaben, die in der *ODBC Programmer's Reference, Version 2.0*, für Ihre Plattform beschrieben werden.

Diese Hilfedatei enthält Anleitungen zur Installation, Konfiguration und Verwendung des ODBC-Treibers für Oracle und umfaßt folgende Abschnitte:

- [Systemanforderungen](#)
- [Hinzufügen und Ändern von Datenquellen mit dem Setup-Programm](#)
- [Konfigurieren des ODBC-Treibers für Oracle](#)
- [Herstellen der Verbindung zu einer Datenquelle](#)
- [Attribute von Verbindungszeichenfolgen](#)
- [ODBC-Konformitätsstufen](#)
- [SQL-Konformitätsstufen](#)
- [Zuordnen von Datentypen](#)
- [Verwenden von Microsoft Transaction Server](#)
- [Verwenden von Microsoft Internet Information Server](#)
- [Verwenden der Betriebssystem-Authentifizierung](#)
- [Einschränkungen bei der Verwendung von treffermengengesteuerten Cursors](#)
- [Rückgabe von Datenfeldparametern von gespeicherten Prozeduren](#)
- [Hinweise zu API-Funktionen](#)
  - [Hinweise zur Thread-Sicherheit bei API-Funktionen](#)
  - [Zentrale API-Funktionen](#)
  - [API-Funktionen der Stufe 1](#)
  - [API-Funktionen der Stufe 2](#)
- [Tabelle mit Verbindungsoptionen](#)
- [Tabelle mit Anweisungsoptionen](#)
- [Tabelle mit Cursorarten und Gleichzeitigkeitskombinationen](#)
- [Fehlermeldungen](#)

## Systemanforderungen

Zur Verwendung des Microsoft ODBC-Treibers für Oracle muss Windows 95/98 oder Microsoft Windows NT und Oracle Client Software, Version 7.3, oder eine höhere Version auf Ihrem Windows-System installiert sein. Der Microsoft ODBC-Treiber für Oracle unterstützt nur SQL\*Net 2.3 oder höher. Weitere Informationen zu den Oracle-Produkten finden Sie in der Dokumentation zu Oracle.

# Hinzufügen und Ändern von Datenquellen mit dem Setup-Programm

Eine Datenquelle gibt den Pfad zu Daten an, die eine Netzwerkbibliothek, einen Server, eine Datenbank oder andere Attribute umfassen. In diesem Fall ist die Datenquelle der Pfad zu einer Oracle-Datenbank. Um die Verbindung zu einer Datenquelle herzustellen, wird die Windows-Registrierung vom Treiber-Manager auf bestimmte Verbindungsinformationen geprüft.

Der vom ODBC-Datenquellen-Administrator erstellte Registrierungseintrag wird vom ODBC-Treiber-Manager und den ODBC-Treibern verwendet. Dieser Eintrag enthält Informationen zu jeder Datenquelle und dem dazugehörigen Treiber. Bevor Sie die Verbindung zu einer Datenquelle herstellen können, müssen die Verbindungsinformationen der Registrierung hinzugefügt werden.

Um Datenquellen hinzuzufügen und zu konfigurieren, greifen Sie über die 32-Bit-ODBC-Systemsteuerung in Windows auf den ODBC-Administrator zu. Dieser aktualisiert dann die Verbindungsinformationen der Datenquelle. Während Sie Datenquellen hinzufügen, aktualisiert der ODBC-Administrator die Registrierungsinformationen.

## So fügen Sie eine Datenquelle für Windows hinzu

- 1 Doppelklicken Sie in der Windows-Systemsteuerung doppelt auf das Symbol **ODBC-Datenquellen**, um den ODBC-Administrator zu starten..
- 2 Wenn das Dialogfeld **ODBC-Datenquellen-Administrator** angezeigt wird, klicken Sie auf die **Schaltfläche Hinzufügen**. Das Dialogfeld **Neue Datenquelle** erstellen wird angezeigt.
- 3 Wählen Sie den **ODBC-Treiber** aus, und klicken Sie auf **Fertig stellen**. Das Dialogfeld **Microsoft ODBC für Oracle - Setup** wird angezeigt..
- 4 Geben Sie im Feld **Datenquellenname** den Namen der Datenquelle ein, auf die Sie zugreifen möchten.. Dabei ist jeder von Ihnen ausgewählte Name zulässig.
- 5 Geben Sie im Feld **Beschreibung** eine Beschreibung des Treibers ein.. In diesem optionalen Feld wird der Datenbanktreiber beschrieben, mit dem die Datenquelle verbunden ist. Dabei ist jeder von Ihnen ausgewählte Name zulässig.
- 6 Im Feld **Benutzername** geben Sie den Datenbank-Benutzernamen ein.. Der Benutzername ist Ihre Datenbank-Benutzer-ID.
- 7 Geben Sie im Feld **Server** die Verbindungszeichenfolge für die Oracle Server-Engine ein.. Die Verbindungszeichenfolge bezeichnet die Oracle Server-Engine, auf die Sie zugreifen möchten.
- 8 Klicken Sie auf **OK**, um diese Datenquelle hinzuzufügen..

**Anmerkung** Das Dialogfeld **Datenquellen** wird angezeigt, und der ODBC-Administrator aktualisiert die Registrierungsinformationen.. Der Benutzername und die Verbindungszeichenfolge, die Sie eingeben, werden die standardmäßigen Verbindungswerte dieser Datenquelle. Wenn Sie folglich unter Verwendung eines Dialogfelds oder einer Verbindungszeichenfolge eine Verbindung zu einer Datenquelle herstellen, werden diese Werte die Standardeinträge für die Datenquellenverbindung.

- 9 Klicken Sie auf Optionen, um weitere Angaben zum Oracle ODBC-Setup vorzunehmen.

## Übersetzung

Klicken Sie auf **Auswählen**, um einen geladenen Datenübersetzer zu wählen. Standardmäßig ist **Kein Übersetzer** ausgewählt.

## Leistungsverhalten

Mit der Option **REMARKS in Katalogfunktionen einfügen** wird festgelegt, ob der Treiber REMARKS-Spalten für die SQLColumns-Ergebnisgruppe zurückgibt.. Der ODBC-Treiber bietet schnelleren Zugriff, wenn dieser Wert nicht eingestellt ist.

Mit der Option **SYNONYMS in SQL-Spalten einfügen** wird festgelegt, ob der Treiber Spalteninformationen zurückgibt.

### **Anpassung**

Mit der Option **ODBC-Standard DayOfWeek erzwingen** wird festgelegt, ob die Ergebnisgruppe dem in ODBC angegebenen Wochentagformat entspricht (Sonntag=1; Samstag=7).

- 10 Klicken Sie auf **Hinzufügen**, um eine weitere Datenquelle hinzuzufügen, oder auf **Abbrechen**, um die Anwendung zu beenden..

### **So ändern Sie eine Datenquelle für Windows**

- 1 Starten Sie den ODBC-Administrator. Das Dialogfeld **Datenquellen** wird angezeigt..
- 2 Wählen Sie im Dialogfeld **Datenquellen** die Oracle-Datenquelle aus, die Sie ändern möchten, und klicken Sie auf **Konfigurieren**.. Das Dialogfeld **Microsoft ODBC für Oracle - Setup** wird angezeigt..
- 3 Ändern Sie die betroffenen Datenquellenfelder, und klicken Sie anschließend auf **OK**.

Wenn Sie die Informationen in diesem Dialogfeld geändert haben, aktualisiert der ODBC-Administrator die Registrierungsinformationen.

# Konfigurieren des ODBC-Treibers für Oracle

Wenn Ihnen die Datenumgebung des Oracle-Treibers bekannt ist, können Sie seine Leistung beeinflussen, indem Sie die Parameter der Datenquellenverbindung im Dialogfeld **ODBC-Datenquellen-Administrator** oder mit Hilfe der Parameter der Verbindungszeichenfolge korrekt einstellen. Das Dialogfeld enthält die folgenden Steuerungen, die zur Verbindungsherstellung mit einer Datenquelle unter Verwendung des Dialogfelds oder von Verbindungszeichenfolgen erforderlich sind:

- **Registerkarte "Benutzer-DSN"** Zeigt eine Liste der Datenquellennamen an, die nur für Ihren Computer gelten..
- **Registerkarte "System-DSN"** Zeigt ein Dialogfeld an, mit dem Sie eine Systemdatenquelle hinzufügen oder löschen können. Systemdatenquellen sind allen Benutzern auf dem lokalen Computer zugänglich.
- **Registerkarte "Datei-DSN"** Zeigt ein Dialogfeld an, mit dem Sie eine Dateidatenquelle zu dem lokalen Computer hinzufügen oder von diesem entfernen können. Dateidatenquellen können von allen Benutzern gemeinsam verwendet werden, die den gleichen Treiber installiert haben.
- **Registerkarte "ODBC-Treiber"** Zeigt eine Liste der installierten ODBC-Treiber an.
- **Registerkarte "Protokollieren"** Zeigt ein Dialogfeld an, in dem Sie festlegen können, wie der ODBC Treiber-Manager Aufrufe von ODBC-Funktionen protokolliert. Sie können die Protokollierung für jede installierte ODBC-Anwendung einzeln konfigurieren.
- **Registerkarte "Verbindungs-Pooling"** Zeigt ein Dialogfeld an, in dem Sie für jeden installierten Treiber Verbindungsoptionen auswählen können.
- **Registerkarte "Info"** Zeigt ein Dialogfeld an, in dem die installierten ODBC-Komponentendateien aufgelistet werden.

Nachdem Sie eine Datenquelle hinzugefügt haben, können Sie das Dialogfeld **ODBC-Datenquellen-Administrator** verwenden, um den Zugriff auf die Datenquelle zu konfigurieren.. Wählen Sie eine Datenquelle aus, und klicken Sie anschließend auf eine der Registerkarten, um die Informationen anzuzeigen oder zu bearbeiten.

Die maximale Standardanzahl (50) von Anweisungshandles wird vom Server festgelegt.

## Herstellen der Verbindung zu einer Datenquelle

Eine ODBC-Anwendung kann Informationen auf verschiedene Art und Weise weiterleiten. So kann der Benutzer beispielsweise vom Treiber immer zur Angabe der Verbindungsinformationen aufgefordert werden, oder es ist die Angabe einer Verbindungszeichenfolge für die Datenquellenverbindung erforderlich. Wie die Verbindung zu einer Datenquelle hergestellt wird, hängt von dem Verbindungsverfahren ab, das Ihre ODBC-Anwendung verwendet.

Normalerweise wird die Verbindung zu einer Datenquelle über das Dialogfeld **Datenquelle** hergestellt.. Wenn Ihre ODBC-Anwendung so konfiguriert ist, dass sie ein Dialogfeld nutzt, wird dieses angezeigt und fordert Sie zur Eingabe der erforderlichen Verbindungsinformationen für die Datenquelle auf.

Eine Verbindung zu einer Datenquelle kann auch mit der Verbindungszeichenfolge hergestellt werden.

### **So stellen Sie eine Verbindung zu einer Datenquelle mit einem Dialogfeld her**

- 1 Wenn das Dialogfeld **Datenquelle** angezeigt wird, wählen Sie eine Oracle-Datenquelle aus und klicken auf **Konfigurieren**. Das Dialogfeld zum **Herstellen einer Verbindung** wird angezeigt.
- 2 Geben Sie im Dialogfeld zum **Herstellen einer Verbindung** die erforderlichen Informationen ein, und klicken Sie auf **OK**.

Sobald die Informationen überprüft sind, kann Ihre Anwendung mit dem ODBC-Treiber auf die Informationen zugreifen, die die Datenquelle enthält.

## Attribute von Verbindungszeichenfolgen

Bei einigen Anwendungen ist u.U. eine Verbindungszeichenfolge erforderlich, die die Verbindungsinformationen der Datenquelle angibt, anstatt sie aus einem Dialogfeld abzurufen. Die Verbindungszeichenfolge enthält einige Attribute, die angeben, wie ein Treiber mit einer Datenquelle verbunden ist. Ein Attribut stellt eine bestimmte Information dar, die der Treiber benötigt, um die richtige Datenquellenverbindung herzustellen. Jeder Treiber kann eine andere Gruppe von Attributen enthalten, das Format der Verbindungszeichenfolge ist jedoch immer dasselbe. Eine Verbindungszeichenfolge hat das folgende Format:

```
"DSN=Datenquellennamen[;SERVER=Wert] [;PWD=Wert] [;UID=Wert]
[;Attribut>=<Wert>]"
```

**Anmerkung** Die Version des Microsoft ODBC-Treibers für Oracle unterstützt das geringfügig unterschiedliche Format von Verbindungszeichenfolgen der Version 1.

Geben Sie einen Datenquellennamen ein, wenn Sie nicht die Attribute UID, PWD, SERVER (oder CONNECTSTRING) und DRIVER angeben. Alle anderen Attribute sind optional. Wenn Sie kein Attribut angeben, wird standardmäßig das Attribut verwendet, das Sie in der Registerkarte **DSN** des Dialogfelds **ODBC-Datenquellen-Administrator** festgelegt haben.. Beim Attributwert muss u.U. die Groß- und Kleinschreibung beachtet werden.

Attribute für die Verbindungszeichenfolge:

Attribut	Beschreibung	Standardwert
DSN	Der Datenquellennamen. Dieser Name ist in der Registerkarte <b>ODBC-Treiber</b> des Dialogfelds <b>ODBC-Datenquellen-Administrator</b> aufgelistet.	""
PWD	Das Kennwort für den Oracle-Server, auf den Sie zugreifen möchten.	""
SERVER	Die Verbindungszeichenfolge für den Oracle-Server, auf den Sie zugreifen möchten.	""
UID	Der Oracle-Server-Benutzername. Je nach System ist dieses Attribut nicht optional – einige Datenbanken und Tabellen benötigen dieses Attribut u.U. aus Sicherheitsgründen.  Mit "/" können Sie die Betriebssystem-Authentifizierung von Oracle verwenden.	""
BUFFERSIZE	Die optimale Puffergröße, die beim Abrufen von Spalten verwendet wird.  Der Treiber optimiert das Abrufen, so dass ein Abruf von dem Oracle Server genügend Zeilen ausgibt, um einen Puffer dieser Größe zu füllen. Größere Werte lassen die Leistung zunehmen, wenn Sie große Datenmengen abrufen.	65535
SYNONYM_COLUMNS	Wenn dieser Wert wahr ist (1), gibt ein <b>SQLColumn()</b> API-Aufruf	1

Spalteninformationen zurück. Andernfalls gibt **SQLColumn( )** nur Spalten für Tabellen und Ansichten zurück. Der ODBC-Treiber bietet schnelleren Zugriff, wenn dieser Wert nicht eingestellt ist.

REMARKS	Ist dieser Wert wahr (1), gibt der Treiber <b>Anmerkungenspalten</b> für die <u>SQLColumns</u> -Ergebnisgruppe zurück. Der ODBC-Treiber bietet schnelleren Zugriff, wenn dieser Wert nicht eingestellt ist.	0
---------	---	---

StdDayOfWeek	Mit diesem Attribut wird der ODBC-Standard für den DAYOFWEEK-Skalar verwendet. Ist standardmäßig aktiviert; Benutzer, die jedoch die lokalisierte Version benötigen, können die Einstellung ändern, um die jeweilige Rückgabe von Oracle zu verwenden.	1
--------------	--	---

Beispiel für eine Verbindungszeichenfolge, die unter Verwendung des Oracle-Servers *mickey.world* die Oracle-Benutzerin *Klara* mit der Angestellten Datenquelle verbindet:

```
"DSN=Angestellten;UID=Klara;PWD=Geheim;SERVER=mickey.world"
```

Ein weiteres Beispiel für eine Verbindungszeichenfolge, die unter Verwendung des Oracle-Servers *Hochburg* und der Betriebssystem-Authentifizierung eine Verbindung zu der Gehaltslisten-Datenquelle herstellt:

```
"DSN=Gehälter;UID=/;PWD=;SERVER=Hochburg"
```



# ODBC-Konformitätsstufen

ODBC definiert zwei Konformitätsstandardtypen für Treiber: API-Konformitätsstandard und SQL-Grammatik-Konformitätsstandard. API-Konformität bezieht sich auf die Funktionen, die ein Treiber unterstützt. SQL-Konformität bezieht sich auf die SQL-Grammatik, die ein Treiber unterstützt. Jeder Konformitätsstandard setzt sich aus mehreren Stufen zusammen.

## API-Konformitätsstufe

Der ODBC-Treiber unterstützt die zentralen Funktionen und die API-Funktionen der Stufe 1. Der Treiber unterstützt auch die folgenden Funktionen der Stufe 2:

- [SQLBrowseConnect\( \)](#)
- [SQLDataSources\( \)](#)
- [SQLDescribeParam\( \)](#)
- [SQLExtendedFetch\( \)](#)
- [SQLForeignKeys\( \)](#)
- [SQLMoreResults\( \)](#)
- [SQLNativeSql\( \)](#)
- [SQLNumParams\( \)](#)
- [SQLPrimaryKeys\( \)](#)
- [SQLProcedureColumns\( \)](#)
- [SQLProcedures\( \)](#)
- [SQLSetPos\( \)](#)
- [SQLSetScrollOptions\( \)](#)

## Unterstützte Optionen

Der Treiber unterstützt die folgenden Optionen für die Funktionen **SQLGetConnectOption( )** und **SQLSetConnectOption( )** der Stufe 1::

- SQL\_ACCESS\_MODE (Nur [SQLGetConnectOption\( \)](#))
- [SQL\\_AUTOCOMMIT](#)
- [SQL\\_ODBC\\_CURSORS](#)
- [SQL\\_OPT\\_TRACEFILE](#)
- [SQL\\_OPT\\_TRACE](#)
- [SQL\\_TRANSLATE\\_DLL](#)
- [SQL\\_TRANSLATE\\_OPTION](#)
- [SQL\\_TXN\\_ISOLATION](#)

Der Treiber unterstützt die folgenden Optionen für die Funktionen **SQLGetStmtOption( )** und **SQLSetStmtOption( )** der Stufe 1:

- SQL\_BIND\_TYPE
- SQL\_CONCURRENCY
- SQL\_CURSOR\_TYPE
- SQL\_KEYSET\_SIZE
- SQL\_MAX\_ROW
- SQL\_ROWSET\_SIZE

# SQL-Konformitätsstufen

Der ODBC-Treiber unterstützt die minimale SQL-Grammatik und die zentrale SQL-Grammatik sowie die folgenden ODBC-Erweiterungen an SQL:

- Datentypen "Date", "Time" und "Timestamp"
- Inklusionsverknüpfungen links und rechts
- Numerische Funktionen:

abs	log	round	tan
ceiling	log10	second	truncate
cos	mod	sign	
exp	pi	sin	
floor	power	sqrt	

- Datenfunktionen:

curdate	dayofweek	monthname	second
curtime	dayofyear	minute	week
dayname	hour	now	year
dayofmonth	month	quarter	

- Zeichenfolgenfunktionen:

ascii	left	right	ucase
char	length	rtrim	
concat	ltrim	soundex	
lcase	replace	substring	

- Typenkonvertierungsfunktionen:

convert

- Systemfunktionen:

ifnull  
user

## Zuordnen von Datentypen

Der Oracle-Server unterstützt eine Reihe von Datentypen. Der ODBC-Treiber ordnet diese Datentypen den entsprechenden ODBC SQL-Datentypen zu. In der folgenden Tabelle werden die Oracle 7.3-Server-Datentypen und die entsprechenden ODBC SQL-Datentypen aufgelistet.

Oracle-Server-Datentyp	ODBC SQL-Datentyp
CHAR	SQL_CHAR
DATE	SQL_TIMESTAMP
FLOAT	SQL_DOUBLE
INTEGER	SQL_DECIMAL
LONG	SQL_LONGVARCHAR
LONG RAW	SQL_LONGVARBINARY
NUMBER	SQL_DECIMAL
RAW	SQL_VARBINARY
VARCHAR2	SQL_VARCHAR

**Anmerkung** MLSLABEL Oracle-Datentypen werden von ODBC SQL-Datentypen nicht unterstützt. Skalarwerte werden im Format der lokalen Zeichenumsetzungstabelle zurückgegeben.

In Oracle 8 ist die maximale Größe einer VARCHAR-Spalte von 2000 auf 4000 Bytes gestiegen. Die Oracle 7.3.x-Clientsoftware kann keine Parameterwerte binden, die größer als 2000 Bytes sind. Aus diesem Grund können Sie beim Erstellen einer Tabelle mit einer VARCHAR-Spalte, die größer als 2000 Bytes ist, auf Daten, die das 2000-Byte-Limit der Clientsoftware überschreiten, keine auf Parameter basierenden Einfügungen, Aktualisierungen und Löschungen oder Abfragen ausführen. Da sowohl der ODBC-Treiber für Oracle als auch der OLE DB-Provider für Oracle auf Parameter basierende Einfügungen, Aktualisierungen, Löschungen und Abfragen verwenden, melden sie in diesem Fall ORA-01026-Fehler. Daten, die sich innerhalb der durch die Oracle Clientsoftware aktivierten Limits befinden, werden angenommen. Um diese 2000-Byte-Grenze zu umgehen, müssen Sie Ihre Clientsoftware zu Oracle 8 (8.0.4.1.1c oder höher) aktualisieren.

## Verwenden von Microsoft Transaction Server

Sie können eine Oracle-Datenbank einrichten, so dass sie mit den Transaktionskomponenten von Microsoft Transaction Server (MTS) unter Windows NT und Windows 95/98 zusammenarbeitet. Damit eine Oracle-Datenbank mit MTS-Komponenten zusammenarbeitet, die Transaktionen unterstützen, müssen Sie eine Sicht mit dem Namen V\$XATRANS\$ erstellen. Um dieses Skript zu erstellen, müssen Sie ein von Oracle geliefertes Skript ausführen. Weitere Informationen entnehmen Sie bitte der Hilfe zu Microsoft Transaction Server oder Ihrer Oracle-Dokumentation.

## Verwenden von Microsoft Internet Information Server

Wenn bei der Verbindungsherstellung von einem Internet Information Server (IIS)-Skript aus Probleme auftreten (besonders ein ORA-12641-Fehler), fügen Sie der Datei SQLNET.ORA folgende Zeile hinzu.

```
SQLNET.AUTHENTICATION_SERVICES = (none)
```

Dadurch werden die Dienste zur Sicherung des Netzwerks deaktiviert, damit eine anonyme Authentifizierung durchgeführt werden kann.

## Verwenden der Betriebssystem-Authentifizierung

Die Betriebssystem-Authentifizierung von Oracle stützt sich auf das zu Grunde liegende Betriebssystem, um den Zugriff auf Datenbankkonten zu steuern. Benutzer brauchen bei Verwendung dieser Art der Anmeldung kein Kennwort einzugeben.

Wenn Sie unter Verwendung einer der folgenden Verbindungs-APIs eine Verbindung herstellen, können Sie diese Funktion nutzen, indem Sie "/" als Benutzer-ID, jedoch kein Kennwort angeben.

**SQLBrowseConnect**, **SQLConnect**, oder **SQLDriverConnect**.

## Einschränkungen bei der Verwendung von treffermengengesteuerten Cursors

Sie müssen eine einzelne ROWID-Spalte aus der abgefragten Tabelle abrufen können. Ein treffermengengesteuerter Cursor kann in Verknüpfungen, Abfragen oder Anweisungen mit DISTINCT-, GROUP BY-, UNION-, INTERSECT- oder MINUS-Klauseln nicht verwendet werden.



## Rückgabe von Datenfeldparametern von gespeicherten Prozeduren

In Oracle 7.3 können Sie auf einen PL/SQL-Datensatztyp nur von einem PL/SQL-Programm aus zugreifen. Wenn die Prozedur oder Funktion in einem Paket ein formales Argument enthält, das als PL/SQL-Datensatztyp definiert ist, kann dieses formale Argument nicht als Parameter gebunden werden. Verwenden Sie den PL/SQL TABLE-Typ im Microsoft ODBC-Treiber für Oracle, um Datenfeldparameter aus Prozeduren abzurufen, die die korrekten ESC-Folgen verwenden.

Verwenden Sie zum Aufrufen der Prozedur die folgende Syntax:

```
{call <package-name>.<proc-or-func>;  
(..., {Ergebnisgruppe <max-records-requested> ,<formal-array-param_1>;  
<formal-array-param_2>,...,<formal-array-param_n> }, ... ) }
```

### Anmerkung

- Der Parameter `<max-records-requested>` muss größer als oder genau so groß wie die Anzahl der in der Ergebnisgruppe vorhandenen Zeilen sein. Ist dies nicht der Fall, gibt Oracle einen Fehler zurück, der Ihnen vom Treiber übergeben wird.
- PL/SQL-Datensätze können nicht als Datenfeldparameter verwendet werden. Jeder Datenfeldparameter kann nur eine Spalte einer Datenbanktabelle repräsentieren.

Im folgenden Beispiel wird ein Paket mit zwei Prozeduren definiert, die jeweils unterschiedliche Ergebnisgruppen zurückgeben; anschließend werden zwei Möglichkeiten bereitgestellt, um Ergebnisgruppen vom Paket zurückzugeben:

### Paketdefinition:

```

CREATE OR REPLACE PACKAGE SimplePackage AS
TYPE t_id IS TABLE OF NUMBER(5)
    INDEX BY BINARY_INTEGER;

TYPE t_Course IS TABLE OF VARCHAR2(10)
    INDEX BY BINARY_INTEGER;

TYPE t_Dept IS TABLE OF VARCHAR2(5)
    INDEX BY BINARY_INTEGER;

PROCEDURE proc1
(
    o_id          OUT    t_id,
    ao_course      OUT    t_Course,
    ao_dept        OUT    t_Dept
);

TYPE t_pk1Type1 IS TABLE OF VARCHAR2(100) INDEX BY BINARY_INTEGER;
TYPE t_pk1Type2 IS TABLE OF NUMBER INDEX BY BINARY_INTEGER;
PROCEDURE proc2
(
    i_Arg1        IN     NUMBER,
    ao_Arg2        OUT    t_pk1Type1,
    ao_Arg3        OUT    t_pk1Type2
);

END SimplePackage;

CREATE OR REPLACE PACKAGE BODY SimplePackage AS
    PROCEDURE proc1 ( o_id OUT t_id,
        ao_course OUT t_Course, ao_dept OUT t_Dept    ) AS
    BEGIN
        o_id(1) := 200;
        ao_course(1) := 'M101';
        ao_dept(1) := 'EEE' ;

        o_id(2) := 201;
        ao_course(2) := 'PHY320';
        ao_dept(2) := 'ECE' ;

    END proc1;
    PROCEDURE proc2
    (
        i_Arg1        IN     NUMBER,
        ao_Arg2        OUT    t_pk1Type1,
        ao_Arg3        OUT    t_pk1Type2
    )
    AS
        i    NUMBER;
    BEGIN
        FOR i IN 1 .. i_Arg1 LOOP
            ao_Arg2(i) := 'Row Number ' || to_char(i);
        END LOOP;
        FOR i IN 1 .. i_Arg1 LOOP
            ao_Arg3(i) := i;
        END LOOP;
    END proc2;

```

```
END SimplePackage;
```

### So rufen Sie die Prozedur PROC1 auf

- 1 Alle Spalten sollen in einer Ergebnisgruppe zurückgegeben werden:

```
{ call SimplePackage.Proc1( {resultset 3, o_id , ao_course, ao_dept } ) }
```

- 2 Jede Spalte soll in einer eigenen Ergebnisgruppe zurückgegeben werden:

```
{call SimplePackage.Proc1( {resultset 3, o_id}, {resultset 3, ao_course},  
{resultset 3, ao_dept} ) }
```

Dadurch werden drei Ergebnisgruppen zurückgegeben, eine für jede Spalte.

### So rufen Sie die Prozedur PROC2 auf

- 1 Alle Spalten sollen in einer Ergebnisgruppe zurückgegeben werden:

```
{call SimplePackage.Proc2( 5 , {resultset 5, ao_Arg2, ao_Arg3} ) }
```

- 2 Jede Spalte soll in einer eigenen Ergebnisgruppe zurückgegeben werden:

```
{call SimplePackage.Proc2( 5 , {resultset 5, ao_Arg2}, {resultset 5,  
ao_Arg3} ) }
```

Stellen Sie sicher, dass Ihre Anwendungen alle Ergebnisgruppen mit **SQLMoreResults** API abrufen. Weitere Informationen erhalten Sie in der *ODBC Programmer's Reference*.

**Anmerkung** Bei der Version 2.0 von ODBC-Treiber für Oracle können Sie keine Oracle-Funktionen verwenden, die PL/SQL-Datenfelder zurückgeben, um Ergebnisgruppen zurückzugeben.

## Hinweise zu API-Funktionen

Der Microsoft ODBC-Treiber für Oracle unterstützt die zentralen API-Funktionen und API-Funktionen der Stufe 1 und 2. Diese Funktionen werden unter ODBC-Konformitätsstufen beschrieben.

- Die Konformität der zentralen Funktionsschnittstelle (CLI = Core Level Interface) bietet die Eigenschaften, die in der ISO CLI-Spezifikation definiert sind, sowie die in der X/Open CLI-Spezifikation definierten obligatorischen Eigenschaften.
- Die Konformität der Funktionen der Stufe 1 bietet eine CLI-Funktionalität sowie zusätzliche Leistungsmerkmale wie Transaktionen.
- Die Konformität der Funktionen der Stufe 2 bietet eine Funktionalität auf Stufe 1 sowie zusätzliche Leistungsmerkmale wie Lesezeichen, dynamische Parameter und asynchrone Ausführung von ODBC-Funktionen.

## Hinweise zur Thread-Sicherheit bei API-Funktionen

Der Microsoft ODBC-Treiber für Oracle ist Thread-geschützt. Oracle erlaubt jedoch keine aktiven, gleichzeitigen Mehrfach-Anweisungen an eine einzelne Verbindung. Der Treiber aktiviert diese Beschränkung. Anders ausgedrückt, in Multi-Thread-Anwendungen sperrt der Treiber, obwohl jeder Thread sich jederzeit in den Oracle ODBC-Treiber einwählen kann, jeden weiteren Thread vom Treiber auf der gleichen Verbindung, bis der ursprüngliche Thread den Treiber verlässt.

Der Treiber sperrt nicht, wenn es zwei Anweisungen auf zwei verschiedenen Verbindungen gibt. Wenn jedoch eine einzige Verbindung mit zwei Anweisungen existiert, besteht die Gefahr des Sperrrens.

## Zentrale API-Funktionen

Die Funktionen auf dieser Stufe beinhalten das Mindestmaß an Schnittstellenkonformität für ODBC-Treiber.

API-Funktion	Anmerkungen
<b>SQLAllocConnect</b>	Weist Arbeitsspeicher zu für das Verbindungshandle, <i>hdbc</i> , in der Umgebung durch <i>henv</i> gekennzeichnet. Der Treiber-Manager verarbeitet diesen Aufruf und ruft immer dann die Funktion <b>SQLAllocConnect</b> des Treibers auf, wenn <u>SQLConnect</u> , <b>SQLBrowseConnect</b> oder <u>SQLDriverConnect</u> aufgerufen werden.
<b>SQLAllocEnv</b>	Zeigt ein Dialogfeld mit Anforderungen für die Oracle Clientsoftware an und gibt anschließend SQL_NULL_HANDLE zurück. Wenn die Oracle Clientsoftware nicht installiert ist, wird mit dieser Funktion Arbeitsspeicher für ein Umgebungshandle, <i>henv</i> , zugeordnet und die ODBC-Schnittstelle auf Aufrufebene zur Verwendung durch eine Anwendung initialisiert.
<b>SQLAllocStmt</b>	Weist Arbeitsspeicher für ein Anweisungshandle zu und ordnet dem Anweisungshandle die durch <i>hdbc</i> angegebene Verbindung zu.. Der Treiber-Manager gibt diesen Aufruf an den Treiber weiter, der den Arbeitsspeicher für die <i>hstmt</i> -Struktur zuordnet.
<b>SQLBindCol</b>	Weist Speicherplatz für eine Ergebnisspalte zu und gibt den Ergebnistyp an.
<b>SQLCancel</b>	Bricht die Verarbeitung eines Anweisungshandle, <i>hstmt</i> , ab. In einigen Fällen erlaubt Oracle nicht, dass Sie eine in der Ausführung befindliche Anweisung abbrechen. Dies bedeutet, dass solch eine Anweisung solange fortgesetzt wird, bis Oracle den Vorgang abschließt. Dies ist dann der Zeitpunkt, zu dem die Ergebnissuche aus den Anweisungen vom ODBC-Treiber abgebrochen wird.
<b>SQLColAttributes</b>	Gibt Deskriptorinformationen für eine Spalte in der Ergebnisgruppe zurück. Deskriptorinformationen werden als Zeichenfolge, Deskriptor-unabhängiger 32-Bit-Wert oder Ganzzahlwert zurückgegeben.
<b>SQLConnect</b>	Stellt eine Verbindung zu einer Datenquelle her Um die Betriebssystem-Authentifizierung von Oracle zu verwenden, geben Sie "/" als <i>szUID</i> -Parameter und "" als <i>szAuthStr</i> -Parameter an.
<b>SQLDescribeCol</b>	Gibt Name, Typ, Genauigkeit, Skalierung und Zulässigkeit von NULL-Werten einer bestimmten Ergebnisspalte zurück.  <b>Anmerkung</b> <b>SQLDescribeCol</b> zeigt berechnete Spalten als SQL_VARCHAR an.
<b>SQLDisconnect</b>	Schließt die Verbindung. Wenn Verbindungs-Pooling für eine gemeinsam genutzte Umgebung aktiviert ist, und eine Anwendung <b>SQLDisconnect</b> an einer Verbindung in dieser Umgebung aufruft, wird die Verbindung in den Verbindungs-Pool zurückgegeben und ist weiterhin für andere Komponenten verfügbar, die die gleiche

gemeinsame Umgebung nutzen.

#### **SQLError**

Gibt Fehler- oder Statusinformationen zum letzten Fehler zurück. Der Treiber führt eine Liste der Fehler, die mit den Argumenten *hstmt*, *hdbc* und *henv* zurückgegeben werden können, je nachdem, wie der Aufruf von **SQLError** durchgeführt wird.. Nach jeder Anweisung wird die Fehlerschlange gelöscht. Diese Funktion ruft im Normalfall eine Oracle-Fehlermeldung ab und ist andernfalls leer.

#### **SQLExecDirect**

Führt eine neue, vorbereitbare SQL-Anweisung aus. Der Treiber verwendet die aktuellen Werte der Variablen zur Parametermarkierung, wenn die Anweisung Parameter enthält. Wenn Ihre Tabellen-, Ansicht- oder Feldnamen Leerstellen enthalten, sollten Sie sie durch Anführungszeichen eingrenzen. Wenn Ihre Datenbank beispielsweise eine Tabelle mit dem Namen "Meine Tabelle" und dem Feld "Mein Feld" enthält, grenzen Sie jedes Element des Bezeichners durch Anführungsstriche ein. Beispiel:

```
SELECT "Meine Tabelle". "Mein Feld1",;  
"Meine Tabelle"."Mein Feld2" FROM "Meine  
Tabelle"
```

#### **SQLExecute**

Führt eine vorbereitete SQL-Anweisung aus (eine Anweisung, die von SQLPrepare bereits vorbereitet wurde). Der Treiber verwendet die aktuellen Werte der Variablen zur Parametermarkierung, wenn die Anweisung Parameter enthält.

#### **SQLFetch**

Ruft eine Zeile aus einer Ergebnisgruppe in die Speicherorte ab, die durch vorangehende Aufrufe von SQLBindCol festgelegt wurden. Der Treiber wird für den Aufruf von SQLGetData für die ungebundenen Spalten vorbereitet..

#### **SQLFreeConnect**

Gibt Verbindungshandle und Arbeitsspeicher, der dem Handle zugeteilt wurde, frei.

#### **SQLFreeEnv**

Beendet den ODBC-Treiber und gibt den Arbeitsspeicher, der dem Treiber zugewiesen wurde, frei.

#### **SQLFreeStmt**

Beendet die mit einer bestimmten *hstmt* verknüpften Verarbeitung, schließt alle offenen Cursor, die mit der *hstmt* verknüpft sind, verwirft Ergebnisse in Bearbeitung und gibt optional alle Ressourcen frei, die mit dem Anweisungshandle verknüpft sind.

#### **SQLGetCursorName**

Gibt den Namen des Cursors zurück, der mit der angegebenen *hstmt* verknüpft ist.

#### **SQLNumResultCols**

Gibt die Spaltenanzahl im Ergebnisgruppen-Cursor zurück.

#### **SQLPrepare**

Bereitet eine SQL-Anweisung vor, indem Optimierung und Ausführung der Anweisung geplant werden. Die SQL-Anweisung wird zur Ausführung von SQLExecDirect kompiliert.

Wenn Ihre Tabellen-, Ansicht- oder Feldnamen Leerstellen enthalten, sollten Sie sie durch Anführungszeichen eingrenzen. Wenn Ihre Datenbank beispielsweise eine Tabelle mit dem Namen "Meine Tabelle" und ein Feld "Mein Feld" enthält, sollte jedes Element des Bezeichners folgendermaßen eingeschlossen

werden:

```
SELECT "Meine Tabelle"."Mein Feld" FROM "Meine  
Tabelle"
```

Informationen zur Verwendung von Ergebnisgruppen, die Felder als formale Parameter enthalten, finden Sie unter Rückgabe von Datenfeldparametern von gespeicherten Prozeduren.

**SQLRowCount**

Oracle bietet erst nach Abrufen der letzten Zeile die Möglichkeit, die Zeilen in einer Ergebnisgruppe zu zählen. Nach Abrufen der letzten Zeile wird -1 zurückgegeben.

**SQLSetCursorName**

Ordnet einen Cursornamen zu einem aktiven Anweisungshandle, *hstmt*.

**SQLSetParam**

In ODBC 2.x durch SQLBindParameter ersetzt.

**SQLTransact**

Fordert einen Commit- oder Rollback-Vorgang an für alle aktiven Vorgänge an allen Anweisungshandles (*hstmts*), die einer Verbindung zugeordnet sind, oder alle Verbindungen, die einer Umgebungshandle, *henv*, zugeordnet sind. Wenn im manuellen Modus ein Commit-Vorgang fehlschlägt, bleibt die Transaktion aktiv. Sie können entweder einen Rollback-Vorgang ausführen oder versuchen, den Commit-Vorgang erneut auszuführen. Schlägt der Commit-Vorgang im automatischen Modus fehl, wird der Rollback-Vorgang automatisch ausgeführt; die Transaktion kann nicht inaktiv sein.



## API-Funktionen der Stufe 1

Die Funktionen auf dieser Stufe bieten eine CLI-Konformität plus zusätzlicher Funktionalität, z.B. Transaktionsunterstützung.

API-Funktion	Anmerkungen
<b>SQLColumns</b>	Erstellt eine Ergebnisgruppe für eine Tabelle, die die Spaltenliste für die angegebene/n Tabelle/n darstellt. Wenn Sie Spalten für ein PUBLIC-Synonym anfordern, muss das SYNONYMCOLUMNS-Verbindungsattribut festgelegt sein und eine leere Zeichenfolge als <i>szTableOwner</i> -Argument angegeben sein. Bei der Rückgabe von Spalten für PUBLIC-Synonyme setzt der Treiber die Spalte TABLE NAME auf eine leere Zeichenfolge. Die Ergebnisgruppe enthält am Ende jeder Zeile eine zusätzliche Spalte, ORDINAL POSITION. Dieser Wert ist die Ordinalzahl der Spalte in der Tabelle.
<b>SQLDriverConnect</b>	Verbindet eine vorhandene Datenquelle. Weitere Details erhalten Sie unter <u>Attribute von Verbindungszeichenfolgen</u> .
<b>SQLGetConnectOption</b>	Gibt die aktuelle Einstellung der Verbindungsoption zurück. Diese Funktion wird teilweise unterstützt: Der Treiber unterstützt alle Werte für das <i>fOption</i> -Argument. Einige <i>vParam</i> -Werte für das <i>fOption</i> -Argument <u>SQL_TXN_ISOLATION</u> unterstützt er jedoch nicht. Weitere Informationen finden Sie in der <u>Tabelle mit Verbindungsoptionen</u> .
<b>SQLGetData</b>	Ruft den Wert eines einzelnen Felds im aktuellen Datensatz der angegebenen Ergebnisgruppe ab.
<b>SQLGetFunctions</b>	Gibt TRUE für alle unterstützten Funktionen zurück. Wird vom Treiber-Manager implementiert.
<b>SQLGetInfo</b>	Gibt Informationen, z.B. SQLHDBC, SQLUSMALLINT, SQLPOINTER, SQLSMALLINT und SQLSMALLINT *, zum ODBC-Treiber und zur Datenquelle zurück, die einem Verbindungshandle, <i>hdbc</i> , zugeordnet sind..
<b>SQLGetStmtOption</b>	Gibt aktuelle Einstellung der Anweisungsoption zurück. Weitere Informationen finden Sie unter <u>Tabelle mit Anweisungsoptionen</u> .
<b>SQLGetTypeInfo</b>	Gibt Informationen zu den Datentypen zurück, die von einer Datenquelle unterstützt werden. Der Treiber gibt die Informationen in einer SQL-Ergebnisgruppe zurück.
<b>SQLParamData</b>	Wird zusammen mit <u>SQLPutData</u> verwendet, um Parameterdaten zum Zeitpunkt der Anweisungsausführung anzugeben.
<b>SQLPutData</b>	Ermöglicht einer Anwendung das Senden von Daten für einen Parameter oder eine Spalte an den Treiber zum Zeitpunkt der Anweisungsausführung.
<b>SQLSetConnectOption</b>	Greift auf Optionen zu, die verschiedene Aspekte der Verbindung beeinflussen. Diese Funktion wird teilweise unterstützt: Der Treiber unterstützt alle Werte für das <i>fOption</i> -Argument. Einige <i>vParam</i> -Werte für das <i>fOption</i> -Argument <u>SQL_TXN_ISOLATION</u> unterstützt er jedoch nicht. Weitere Informationen finden Sie in

der Tabelle mit Verbindungsoptionen.

#### **SQLSetStmtOption**

Stellt die Optionen ein, die mit einem Anweisungshandle, *hstmt*, verknüpft sind. Weitere Informationen finden Sie unter Tabelle mit Anweisungsoptionen.

#### **SQLSpecialColumns**

Ruft die optimale Spaltengruppe ab, die eine Zeile in der Tabelle eindeutig angibt.

#### **SQLStatistics**

Ruft eine Liste von Daten zu einer Tabelle und den Indizes oder Kennzeichennamen, die der Tabelle zugeordnet sind, ab. Der Treiber gibt die Informationen in Form einer Ergebnisgruppe zurück.

#### **SQLTables**

Gibt eine Liste von Tabellennamen zurück, die durch den Parameter in der Anweisung **SQLTables** festgelegt werden. Wenn kein Parameter angegeben ist, werden die Tabellennamen zurückgegeben, die in der aktuellen Datenquelle gespeichert sind. Der Treiber gibt die Informationen in Form einer Ergebnisgruppe zurück.

Aufrufe des Aufzählungstyps empfangen keinen Ergebnisgruppeneintrag für Remote-Sichten oder lokale parameterisierte Sichten. Ein Aufruf von **SQLTables** mit einer eindeutigen Tabellennamenangabe findet jedoch eine Übereinstimmung für eine Ansicht, die mit diesem Namen vorliegt; dadurch kann die API Namenkonflikte überprüfen, bevor eine neue Tabelle erstellt wird.

PUBLIC-Synonyme werden mit einem TABLE\_OWNER-Wert von "" zurückgegeben.

SYS- oder SYSTEM-eigene VIEWS werden als SYSTEM VIEW gekennzeichnet.

## API-Funktionen der Stufe 2

Funktionen dieser Stufe bieten Schnittstellenkonformität der Stufe 1 sowie zusätzliche Leistungsmerkmale wie Unterstützung für Lesezeichen, dynamische Parameter und asynchrone Ausführung von ODBC-Funktionen.

API-Funktion	Anmerkungen
<b>SQLBindParameter</b>	Ordnet einen Puffer zu einer Parametermarkierung in einer SQL-Anweisung zu.
<b>SQLBrowseConnect</b>	Gibt aufeinander folgende Stufen von Attributen und Attributwerten zurück.
<b>SQLDataSources</b>	Listet Datenquellennamen auf. Wird vom Treiber-Manager implementiert.
<b>SQLDescribeParam</b>	<p>Gibt die Beschreibung einer Parametermarkierung zurück, die einer vorbereiteten SQL-Anweisung zugeordnet ist.</p> <p>Gibt eine beste Schätzung des Parametertyps zurück, die auf dem Analyseergebnis der Anweisung basiert. Falls der Parametertyp nicht bestimmt werden kann, gibt SQL_VARCHAR eine Länge von 2000 zurück.</p>
<b>SQLDrivers</b>	Wird vom Treiber-Manager implementiert.
<b>SQLExtendedFetch</b>	Ist vergleichbar mit <u>SQLFetch</u> , es werden jedoch mehrere Zeilen zurückgegeben, wobei für jede Spalte ein Datenfeld verwendet wird. Die Ergebnisspalte kann vorwärts durchblättert werden und auch rückwärts, wenn der Cursor als statisch definiert wurde anstatt als Vorwärtscursor. Bei Vorwärtscursoren mit standardmäßiger Spaltenbindung werden Spaltendaten aus Datengruppen, die größer sind als das Verbindungsattribut <code>BUFFER_SIZE</code> , direkt in die Datenpuffer geholt. Unterstützt keine Lesezeichen variabler Länge und unterstützt keine Abrufe der ersten Zeilen bei einem Offset (ungleich 0) eines Lesezeichens.
<b>SQLForeignKeys</b>	Gibt eine Liste der Fremdschlüssel in einer einzelnen Tabelle oder einer Liste von Fremdschlüsseln in anderen Tabellen zurück, die sich auf eine einzige Tabelle beziehen.
<b>SQLMoreResults</b>	<p>Stellt fest, ob noch Ergebnisse zu einem Anweisungshandle, <i>hstmt</i>, ausstehen, die <b>SELECT</b>-, <b>UPDATE</b>-, <b>INSERT</b>- oder <b>DELETE</b>-Anweisungen enthalten. Wenn dies der Fall ist, wird die Verarbeitung dieser Ergebnisse initialisiert.</p> <p>Oracle unterstützt Mehrfach-Ergebnisgruppen von gespeicherten Prozeduren nur, wenn {resultset...}-ESC-Folgen verwendet werden.</p>
<b>SQLNativeSql</b>	Informationen zur Verwendung erhalten Sie unter <u>Rückgabe von Datenfeldparametern von gespeicherten Prozeduren</u> .
<b>SQLNumParams</b>	Gibt die Anzahl der Parameter in einer SQL-Anweisung zurück. Die Anzahl der Parameter sollte der Anzahl von Fragezeichen in der SQL-Anweisung entsprechen, die an <u>SQLPrepare</u> weitergeleitet wird.

<b>SQLPrimaryKeys</b>	Gibt den Spaltennamen zurück, aus denen der Primärschlüssel für eine Tabelle besteht.
<b>SQLProcedureColumns</b>	Gibt eine Liste von Eingabe- und Ausgabeparametern, den Rückgabewert und die Spalten in der Ergebnisgruppe einer Prozedur und zwei weitere Spalten, OVERLOAD und ORDINAL_POSITION, zurück. OVERLOAD ist die Spalte OVERLOAD in der Tabelle ALL_ARGUMENTS der Datenbeschreibungssicht von Oracle. ORDINAL_POSITION ist die Spalte SEQUENCE in der Tabelle ALL_ARGUMENTS der Datenbeschreibungssicht von Oracle. Bei gepackten Prozeduren befindet sich die Spalte PROCEDURE NAME im Format <i>paketname.prozedurname</i> . Gibt nicht die Prozedurspalten eines erstellten Synonyms zurück, das auf eine Prozedur oder Funktion verweist.
<b>SQLProcedures</b>	<p>Gibt eine Liste von Prozeduren zurück, die sich in der Datenquelle befinden. Bei gepackten Prozeduren befindet sich die Spalte PROCEDURE NAME im Format <i>paketname.prozedurname</i>.</p> <p>Da in Oracle nicht zwischen gepackten Prozeduren und gepackten Funktionen unterschieden werden kann, gibt der Treiber SQL_PT_UNKNOWN für die Spalte PROCEDURE_TYPE zurück.</p>
<b>SQLSetPos</b>	Legt die Cursorposition in einem Rowset fest. Sie können <b>SQLSetPos</b> mit <u>SQLGetData</u> verwenden, um Zeilen von ungebundenen Spalten abzurufen, nachdem der Cursor in eine bestimmte Zeile innerhalb des Rowsets gesetzt wurde. Zeilen, die der Ergebnisgruppe unter Verwendung von <i>fOption</i> SQL_ADD hinzugefügt werden, werden nach der letzten Zeile der Ergebnisgruppe angehängt.
<b>SQLSetScrollOptions</b>	Legt die Optionen zur Steuerung des Cursorverhaltens fest, die das Verhalten von Cursors steuern, die einem Anweisungshandle, <i>hstmt</i> , zugeordnet sind. Genauere Informationen finden Sie unter <u>Tabelle mit Cursortypen und Gleichzeitigkeitskombinationen</u> .

## Tabelle mit Verbindungsoptionen

Diese Optionen ermöglichen die Anpassung der Datenbankverbindung innerhalb einer Anwendung.

Verbindungsoption	Anmerkungen
SQL_AUTOCOMMIT	Wenn Sie SQL_AUTOCOMMIT_OFF auswählen, muss Ihre Anwendung Commit- oder Rollback-Vorgänge deutlich mit <u>SQLTransact</u> übergeben oder zurücksetzen.
SQL_ODBC_CURSORS	Dieses Verbindungsattribut ist im Treiber-Manager implementiert.
SQL_OPT_TRACE	Dieses Verbindungsattribut ist im Treiber-Manager implementiert.
SQL_OPT_TRACEFILE	Dieses Verbindungsattribut ist im Treiber-Manager implementiert.
SQL_TRANSLATE_DLL	Gibt folgende Fehlermeldung zurück: "Treiber unterstützt Funktion nicht".
SQL_TRANSLATE_OPTION	Ein 32-Bit-Wert wird an die Übersetzungs-DLL übergeben.
SQL_TXN_ISOLATION	Der Treiber lässt nur SQL_TXN_READ_COMMITTED zu. Folgende <i>vParams</i> werden nicht unterstützt: SQL_TXN_READ_UNCOMMITTED SQL_TXN_REAPEATABL_READ SQL_TXN_SERIALIZABLE
SQL_ATTR_ENLIST_IN_DTC	Dieses ODBC 3.0-Verbindungsattribut ermöglicht die Verwendung des Oracle ODBC-Treibers in Distributed Transactions, die vom Microsoft Transaction Server koordiniert werden. Es stellt den Schnittstellenzeiger <i>plTransaction</i> für die Transaktion als <i>vParam</i> -Argument bereit.
SQL_ATTR_CONNECTION_DEAD	Dieses schreibgeschützte ODBC 3.5-Verbindungsattribut ermöglicht Ihnen die Bestimmung, ob die Verbindung zu dem Oracle-Server fehlgeschlagen ist. Kann nur <b>abgerufen</b> , aber nicht <b>festgelegt</b> werden.

## Tabelle mit Anweisungsoptionen

Diese Optionen ermöglichen die Anpassung einer bestimmten Ausführungsanweisung innerhalb einer Anwendung.

Anweisungsoptionen	Anmerkungen
SQL_BIND_TYPE	2.147.483.647 Byte bzw. verfügbarer Speicher darf nicht überschritten werden.
SQL_CONCURRENCY	Zulässige Werte können Sie in der <u>Tabelle mit Cursortypen und Gleichzeitigkeitskombinationen</u> nachlesen.
SQL_CURSOR_TYPE	Der Treiber ermöglicht keine SQL_CURSOR_DYNAMIC. Weitere Informationen erhalten Sie unter <u>SQLSetScrollOptions</u> . Zulässige Werte können Sie in der <u>Tabelle mit Cursortypen und Gleichzeitigkeitskombinationen</u> nachlesen.
SQL_GET_BOOKMARK	Gibt einen 32-Bit-Ganzzahlwert zurück, der als Lesezeichen für die aktuelle Datensatznummer dient. Kann nur <b>abgerufen</b> , aber nicht <b>festgelegt</b> werden.
SQL_KEYSET_SIZE	Kann nur auf 0 eingestellt werden.
SQL_MAX_ROWS	Die maximale Anzahl von Zeilen, die von einer Ergebnisgruppe zurückgegeben werden sollen.
SQL_ROW_NUMBER	Gibt eine 32-Bit-Ganzzahl zurück, die die Position der aktuellen Zeile innerhalb der Ergebnisgruppe angibt. Kann nur <b>abgerufen</b> , aber nicht <b>festgelegt</b> werden.
SQL_ROWSET_SIZE	4.294.967.296 Zeilen dürfen nicht überschritten werden; sie müssen jedoch für genügend virtuellen Arbeitsspeicher sorgen, damit der Computer Ihre Anfrage durchführen kann.
SQL_USE_BOOKMARKS	Unterstützt die Einstellung von SQL_USE_BOOKMARKS auf SQL_UB_ON und Darstellung von Lesezeichen vorgegebener Länge.

## Tabelle mit Cursortypen und Gleichzeitigkeitskombinationen

Cursortypen steuern die Funktionalität des Cursors, der dem Benutzer zur Verfügung steht. Gleichzeitigkeitsoptionen steuern die Aktualisierbarkeit und das Sperrverhalten einer Ergebnisgruppe.

<b>Cursortyp</b>	<b>Gleichzeitigkeit (zulässige Werte)</b>
SQL_CURSOR_FORWARD_ONLY	SQL_CONCUR_READ_ONLY
SQL_CURSOR_STATIC	SQL_CONCUR_READ_ONLY
SQL_CURSOR_KEYSET_DRIVEN*	SQL_CONCUR_READ_ONLY SQL_CONCUR_LOCK** SQL_CONCUR_VALUES

### Anmerkungen

- \* Informationen finden Sie unter Einschränkungen bei der Verwendung von treffermengengesteuerten Cursors.
- \*\* SQL\_CONCUR\_LOCK wird nur unterstützt, wenn die Verbindungsoption SQL\_AUTOCOMMIT auf SQL\_AUTOCOMMIT\_OFF gestellt ist.

# Fehlermeldungen

Wenn ein Fehler auftritt, gibt der Microsoft ODBC-Treiber für Oracle das SQLSTATE (ein ODBC-Fehlercode) und eine Fehlermeldung zurück. Der Treiber leitet diese Informationen sowohl aus Fehlern ab, die vom Treiber gefunden wurden, sowie aus Fehlern, die vom Oracle-Server zurückgegeben wurden.

## Vom Oracle ODBC-Treiber zurückgegebene Meldungen

Wenn eine Oracle-Fehlermeldung verfügbar ist, werden die folgenden Marken vorangestellt: [Microsoft], [ODBC Driver for Oracle] und [Oracle]. Andernfalls wird die Meldung ohne die [Oracle]-Marke zurückgegeben. Beispiele:

Oracle-Fehlermeldung:

```
[Microsoft][ODBC driver for Oracle][Oracle]ORA-nnnnn message-text
```

Oracle ODBC-Treiber-Fehlermeldung:

```
[Microsoft][ODBC driver for Oracle]
```



