Access Tutorial 5: Basic Queries using SQL

5.1 Introduction: The difference between QBE and SQL

Query-By-Example (QBE) and Structured Query Language (SQL) are both well-known, industry-standard languages for extracting information from relational database systems. The advantage of QBE (as you saw in Tutorial 4) that it is graphical and relatively easy to use. The advantage of SQL is that it has achieved nearly universal adoption within the relational database world.

With only a few exceptions (which you probably will not encounter in this assignment) QBE and SQL are completely interchangeable. If you understand the underlying concepts (projection, selection, sorting, joining, and calculated fields) of one, you understand the underlying concepts of both. In fact, in Access you can switch between QBE and SQL versions of your queries with the click of a mouse.

Although you normally use QBE in Access, the ubiquity of SQL in organizations necessitates a brief overview.

5.2 Learning objectives

- □ What is the difference between QBE and SQL?
- □ How do I create an SQL query?

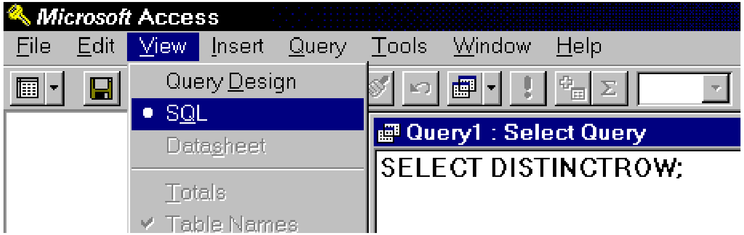
5.3 Tutorial exercises

In this section, you will create a few simple queries in SQL.

- Create a new query but close the “show table” dialog box with out adding tables.
- Select *View > SQL* to switch to the SQL editor as shown in Figure 5.1.

Last update: 22-Aug-1997

FIGURE 5.1: Open a query in SQL mode



SELECT DISTINCTROW;

5.3.1 Basic SQL queries

A typical SQL statement resembles the following: `SELECT DeptCode, CrsNum, Title FROM Courses WHERE DeptCode = “COMM”;`

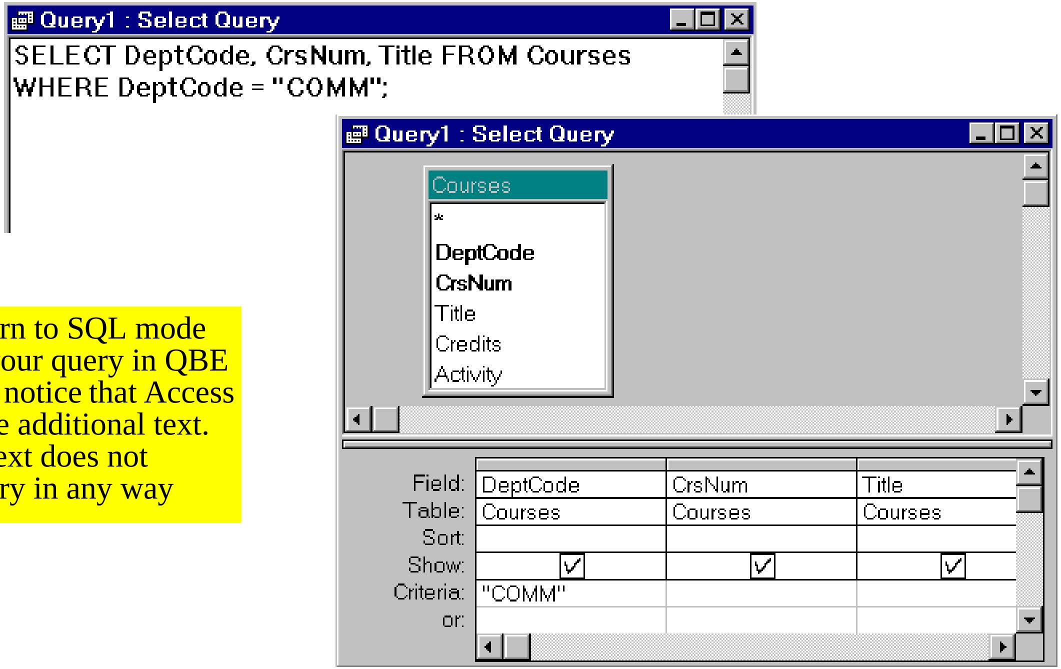
There are four parts to this statement:

1. `SELECT <field`$_1$`, field`$_2$`, …, field`$_n$`>` … — specifies which fields to project (the `DISTINCTROW` predicate shown in Figure 5.1 is optional and will not be discussed in this tutorial);
2. … `FROM <table>` … — specifies the underlying table (or tables) for the query;
3. … `WHERE <condition`$_1$` AND/OR condition`$_2$`, …, AND/OR condition`$_n$`>` — specifies one or more conditions that each record must satisfy in order to be included in the results set;
4. `;` (semicolon) — all SQL statements must end with a semicolon (but if you forget it, Access will add it for you).

These can now be put together to build an SQL query:

- Type the following into the SQL window: `SELECT DeptCode, CrsNum, Title FROM Courses WHERE DeptCode = “COMM”;`
- Select *View > Datasheet* to view the results.
- Select *View > Query Design* to view the query in QBE mode, as shown in Figure 5.2.
- Save your query as `qryCoursesSQL`.

FIGURE 5.2: The SQL and QBE views are interchangeable.



```
SELECT DeptCode, CrsNum, Title FROM Courses
WHERE DeptCode = "COMM";
```

Field:	DeptCode	CrsNum	Title
Table:	Courses	Courses	Courses
Sort:			
Show:	☑	☑	☑
Criteria:	"COMM"		
or:			

When you return to SQL mode after viewing your query in QBE mode, you will notice that Access has added some additional text. This optional text does not change the query in any way

5.3.2 Complex WHERE clauses

You can use AND, OR, and NOT conditions in your WHERE clauses in a straightforward manner.

- Change your query to the following to get all Commerce courses with more than three credits:

```
SELECT DeptCode, CrsNum, Title
FROM Courses
WHERE DeptCode = “COMM” AND Credits
 > 3
```

⚠ Note that since `DeptCode` is a text field, its criterion must be a string (in this case, the literal string “COMM”). However, `Credits` is a numeric field and its criterion must be a number (thus, there cannot be quotation marks around the 3).

5.3.3 Join queries

Join queries use the same elements as a basic select query. The only difference is that the FROM statement is replaced with a statement that describes the tables to be joined and the relationship (i.e., foreign key) between them:

$\texttt{... FROM table}_1\texttt{ INNER JOIN table}_2\texttt{ ON}$
$\texttt{table}_1\texttt{.field}_1\texttt{ = table}_2\texttt{.field}_2\texttt{ ...}$

Note that since both tables contain the fields `DeptCode` and `CrsNum`, the `<table name>.<field name>` notation must be used to remove any ambiguity.

- Create a new SQL query containing the text:

```
SELECT Courses.DeptCode,
 Courses.CrsNum, Courses.Title,
 Sections.CatalogNum
FROM Courses INNER JOIN Sections ON
 Courses.CrsNum = Sections.CrsNum
```

```
    AND Courses.DeptCode =
    Sections.DeptCode
  WHERE Courses.DeptCode=”COMM”;
```

5.4 Discussion

Although the syntax of SQL is not particularly difficult, writing long SQL queries is tedious and error-prone. For this reason, you are advised to use QBE for the assignment.

In the real world, however, when you say you know something about databases, it usually implies you know the “data definition” and “data manipulation” aspects of SQL in your sleep. If you plan to pursue a career in information systems, a comprehensive SQL reference book can be a worthwhile investment.