

Management Information Systems, 15TH ED.

MANAGING THE DIGITAL FIRM

Kenneth C. Laudon • Jane P. Laudon

Chapter 6: Foundations of Business Intelligence: Databases and Information Management

Learning Track 2: Introduction to SQL

This Hands-On Guide is a brief introduction to Structured Query Language (SQL), the standard language for relational database management systems. SQL can be used for defining database structures, querying databases, and updating database data. If you are using Access, you will probably use its Query, Report, and Table-building tools rather than SQL for most of your querying and reporting work. However, you may want to use SQL to edit Access queries, and you can use your knowledge of SQL when you work with other database management systems. We will use tables from the HisNHers database developed in the previous appendix to illustrate how SQL works, focusing on the most important SQL commands for retrieving data. We will show you how to create SQL queries using sample data from the CUSTOMER Table and the APPOINTMENT Table illustrated here.

CUSTOMER TABLE WITH SAMPLE DATA

CUST ID	FNAME	MID NAME	LNAME	ADDR1	ADR2	CITY	STATE	ZIP	HOME PHONE	WORK PHONE	CELL PHONE	EMAIL	DOB	SEX	PERSONAL INFO	OLD CUST ID
1	Georgia	A	Brown	88 Main Street		Missoula	NY	10521	(914) 332-1234	(914) 222-8182	(912) 455-5555	gbrown@qqq.org	7/7/1951	F	WORKS AT M8	
2	Patrick		Malone	42 Sweetbriar L		Bocarain	NY	10598	(914) 333-4333	(211) 222-2121		PMAL@AOL.COM	6/12/1981	M		
3	Juliet	Q	Montague	11 Venice Plaza	Apt. 22	Bocarain	NY	10598	(914) 888-8181	(212) 921-0404	(212) 222-1221		12/25/1971	F		
4	Mary Ellen		Smith	Marian Lane		Catalan	NY	10520	(914) 821-3666				1/23/1959	F		
5	Mary Allen		Smith	22 September		Bocarain	NY	10598		(914) 999-1919		masmith@smith.com	4/25/1973	F		
6	Justin	J	Jumpup	6233 Ridgeview		Catalan	NY	10520				jjup@bigcompany.com	9/13/1966	M		M
7	Thomas	E	Brown	7 Schiff Ave	Bldg. C	Bocarain	NY	10598		(212) 333-4567	(912) 456-7823		2/27/1981	M	LOVE THE ME	
8	Zoe		Nother	91 Glendale		Bocarain	NY	10598	(212) 345-6789				3/14/1968	F		
9	Mary Ann		Smith	8 Race Road		Catalin	NY	10520	(941) 222-1888	(941) 765-3424			5/30/1083	F	Early gray show	
10	John Paul		Jones	66 Ocean View		Bocarain	NY	10598		(212) 459-7878	(430) 686-8565			M	very quiet guy	
12	Jeanne	J	Greene	9499 Threws Le		Bocarain	NY	10598								
13	Mark		MacGregor	43 Schiff Ave		Catalan	NY	10520								
15	Jim	Q	Brown	9 Penland Park		Missoula	NY	10521	(518) 333-3311				6/10/1979	M	handsome!	

*Number

continued

APPOINTMENT TABLE WITH SAMPLE DATA

CUSTID	SERVICE_ID	APPT_DATE	APPT_TIME	EMP_SS	PROD_NUM	APPT_COMMENT
1	2	11/5/2013	3:30 PM	123456789		
1	2	11/7/2013	2:00 PM	123456789		
2	2	11/5/2013	8:30 AM	987654321		
2	3	11/7/2013	3:00 PM	123456789		4
3	5	11/5/2013	10:00 AM	123456789		8
3	5	11/7/2013	11:30 AM	987654321		8
4	2	11/5/2013	3:00 PM	987654321		
4	1	11/7/2013	9:00 AM	123456789		6
5	3	11/5/2013	12:00 PM	123456789		3
5	3	11/5/2013	9:00 AM	123456789		2
6	5	11/5/2013	11:30 AM	987654321		
7	6	11/5/2013	10:30 AM	123456789		
7	2	11/7/2013	4:30 PM	987654321		
8	1	11/5/2013	1:00 PM	987654321		
9	4	11/5/2013	2:30 PM	123456789		8
10	2	11/5/2013	4:30 PM	123456789		
12	4	11/5/2013	9:30 AM	987654321		8
13	2	11/5/2013	8:30 AM	123456789		

Basic SQL Syntax Rules

SQL, like all computer languages has a particular syntax or grammar which must be followed in order for the commands to be understood and executed properly. Its most basic rules are:

1. Identifiers (names of tables, columns, and other objects) should contain between 1 and 30 characters. The identifiers can be upper or lower case, but no embedded spaces are allowed. For example, WORK PHONE would have to be written as WORKPHONE or WORK_PHONE.
2. SQL is not case sensitive, although SQL keywords such as SELECT or FROM are usually capitalized. Keywords have predefined meanings and cannot be used as identifiers.
3. SQL statements can take up more than one line (and there are no restrictions on the number of words per line or where to break a line). However, a new line is often started when a new clause in an SQL statement begins.
4. Commands begin with the SQL operator (e.g., CREATE or SELECT).
5. Field (column) names are separated from each other by a comma (e.g., SELECT FNAME, LNAME FROM tblCUSTOMER).
6. Field (column) names are separated from table names by a period (e.g., tblCUSTOMER.FNAME, tblCUSTOMER.LNAME). (Access uses this convention.) The name of the table does not have to be written before the name of the field from that table unless two tables used in the same SQL statement have identical field names (such as the CUSTID field in both

continued

the CUSTOMER and APPOINTMENT tables). Then the table name must be included (e.g., tblCUSTOMER.CUSTID and tblAPPOINTMENT.CUSTID).

7. Strings must be enclosed in quotation marks. A string is text for a collection of bits that is foreign to the database. The SQL statement to find the phone number for all of the customers whose last name is MacGregor would be written as:

```
SELECT HOMEPHONE
FROM tblCUSTOMER
WHERE LNAME = "MacGregor";
```

Remember too that a space is considered a character and must always be included in the quotation marks if it is needed for the comparison. "Mac Gregor" and "MacGregor" will not return the same Scotchman.

8. Nested operations must be enclosed in parentheses. One of the strengths of SQL is its ability to perform quite complex data manipulation. This is primarily due to its use of nesting. Commands within commands are enclosed in parentheses and the innermost ones are executed first.

Using the Select Statement

The most often used SQL command is SELECT, which returns rows of data from columns in the tables that the user would like to see in a result table. Following are a series of SELECT statements which demonstrate how to have all or specified columns returned in the result table. The names of the columns to be queried follow the keyword SELECT and the name of the table to use follows the keyword FROM.

SELECT every column and every row

```
SELECT *
FROM tblCUSTOMER;
```

The asterisk after SELECT causes every column to be included in the records from the table, tblCUSTOMER. The results show all the rows in the CUSTOMER Table as displayed earlier.

SELECT only some columns and every row

```
SELECT ADDR1, LNAME
FROM tblCUSTOMER;
```

Only the columns which are listed are returned from the CUSTOMER Table and they are presented in the order they are listed in the statement.

ADDR1	LNAME
88 Main Street	Brown
42 Sweetbriar Lane	Malone
11 Venice Plaza	Montague
Marian Lane	Smith
22 September Street	Smith
6233 Ridgeview	Jumpup
7 Schiff Ave	Brown
91 Glendale	Nother
8 Race Road	Smith
66 Ocean View	Jones
9499 Throws Le.	Greene
43 Schiff Ave	MacGregor

Sorting Data: Using Order By

The following examples show the ORDER BY statement which is used to sort the data returned by the SELECT statement.

Select only some columns and every row in sorted order by one field

```
SELECT LNAME, FNAME
FROM tblCUSTOMER
ORDER BY LNAME;
```

The ORDER BY command causes the records to be sorted in order by the contents of the column name following it.

LNAME	FNAME
Brown	Jim
Brown	Thomas
Brown	Georgia
Greene	Jeanne
Jones	John Paul
Jumpup	Justin
MacGregor	Mark
Malone	Patrick
Montague	Juliet
Nother	Zoe
Smith	Mary Ann
Smith	Mary Allen
Smith	Mary Ellen

Select only some columns and every row in sorted order by several fields

The returned records may be sorted in order by several fields. Each listed field after the ORDER BY clause is separated by a comma. First the records are sorted by last name, and if the last names are the same they are sorted by first name.

```
SELECT NAME, FNAME
FROM tblCUSTOMER
ORDER BY LNAME, FNAME;
```

LNAME	FNAME
Brown	Georgia
Brown	Jim
Brown	Thomas
Greene	Jeanne
Jones	John Paul
Jumpup	Justin
MacGregor	Mark
Malone	Patrick
Montague	Juliet
Nother	Zoe
Smith	Mary Allen
Smith	Mary Ann
Smith	Mary Ellen

Selecting Only Records That Meet Specified Criteria: Using Where

The WHERE clause determines which records are returned based on criteria described in the clause. It includes conditions for selecting specific rows in a table. Operators, such as = (equal), > (greater than), < (less than), and <> (not equal), as well as logical operators such as AND and OR are used to create the selection criteria. "IS NULL" is used to find fields that are empty (i.e., they contain no data). Remember that to a computer, a space is a character and a field into which only one space has been keyed is not considered NULL.

Select only records where the field SEX contains an F in sorted order by last name and first name.

```
SELECT LNAME, FNAME, SEX
FROM tblCUSTOMER
WHERE SEX = "F"
ORDER BY LNAME, FNAME;
```

Note that the letter F is a string and therefore must be enclosed in quotes.

LNAME	FNAME	SEX
Brown	Georgia	F
Montague	Juliet	F
Nother	Zoe	F
Smith	Mary Allen	F
Smith	Mary Ann	F
Smith	Mary Ellen	F

Select only the records where SEX does not contain an F in sorted order by last name and first name

```
SELECT FNAME, LNAME, SEX
FROM tblCUSTOMER
WHERE SEX <> "F"
ORDER BY LNAME, FNAME;
```

FNAME	LNAME	SEX
Jim	Brown	M
Thomas	Brown	M
John Paul	Jones	M
Justin	Jumpup	M
Patrick	Malone	M

Select only the records where SEX contains an F and the zip code is 10520 in sorted order by last name and first name.

```
SELECT LNAME, FNAME, SEX, ZIP
FROM tblCUSTOMER
WHERE SEX = "F" AND ZIP = "10520"
ORDER BY LNAME, FNAME;
```

The logical operator AND is used to select information that is contained within two different fields. Multiple AND and OR operators can be contained within one WHERE clause.

The zip code 10520 is enclosed in quotes because it is a string. The ZIP code field is a text field and even though its data may look numeric, it is stored as text characters and can only be found with a string. If the field had been defined as a numeric one, the quotes around 10520 would not be used.

LNAME	FNAME	SEX	ZIP
Smith	Mary Ann	F	10520
Smith	Mary Ellen	F	10520

Joining Multiple Tables

Data from two or more tables can be joined together to provide data from both tables where the contents of a field found in both tables matches. Both `tblCUSTOMER` and `tblAPPOINTMENT` are needed to see who has an appointment on a particular date. Since both tables contain the customer's ID, the contents of this field must match for the records to be returned.

Select the appointment date, time, and last name of all male customers with appointments.

```
SELECT APPT _ DATE, APPT _ TIME, LNAME
FROM tblCUSTOMER, tblAPPOINTMENT
WHERE tblCUSTOMER.CUSTID = tblAPPOINTMENT.CUSTID AND SEX = "M";
```

The preceding SQL syntax that works with most DBMS products. Access, however, uses a slightly different syntax for this particular operation. It would be:

```
SELECT tblAPPOINTMENT.APPT _ DATE,
tblAPPOINTMENT.APPT _ TIME, tblCUSTOMER.LNAME
FROM.tblCUSTOMER
INNER JOIN tblAPPOINTMENT ON tblCUSTOMER.CUSTID =
tblAPPOINTMENT.CUSTID
WHERE (((tblCUSTOMER.SEX) = "M"));
```

APPT_DATE	APPT_TIME	LNAME
11/5/2004	8:30 AM	Malone
11/7/2004	3:00 PM	Malone
11/5/2004	11:30 AM	Jumpup
11/5/2004	4:30 PM	Jones

COPYRIGHT NOTICE

Copyright © 2017 Kenneth Laudon and Jane Laudon.

This work is protected by United States copyright laws and is provided solely for the use of instructors in teaching their courses and assessing student learning. Dissemination or sale of any part of this work (including on the World Wide Web) will destroy the integrity of the work and is not permitted. The work and materials from this site should never be made available to students except by instructors using the accompanying text in their classes. All recipients of this work are expected to abide by these restrictions and to honor the intended pedagogical purposes and the needs of other instructors who rely on these materials.