

# *: What SQL knowledge do managers need for decision support queries?*

by Dan Power

Editor, DSSResources.com

Structured Query Language (SQL) is considered difficult for most managers to learn and use. SQL is a complex language with many capabilities, but managers can learn to manipulate and retrieve data in a decision support query. The knowledge of SQL needed by a manager is limited. The decision support question or query to the database must be structured however to conform to the SQL command language rules.

An SQL command is a statement composed of a sequence of key words, identifiers, delimiters and constants. So what are the important facts about writing an SQL query?

First, begin all decision support queries with the key word SELECT. This command key word tells the database program you want to find and retrieve data. You often need to end the query with a semi-colon (;).

Second, list the data fields you want to retrieve. If the field name is unique, the name is adequate. If the name is used in multiple tables, then specify the table name and then the field name using the format tablename.fieldname. If you want multiple fields, then separate the names using commas. The asterisk \* can be used to retrieve all fields.

Third, if you want only distinct values from the fields preface the list of field names with the keyword DISTINCT, the default is ALL.

Fourth, the program needs to be told the tables where the data is located. The keyword FROM precedes the list of tables.

```
SELECT DISTINCT column_name(s)
```

```
FROM table_name
```

## *: What SQL knowledge do managers need for decision support queries?*

Fifth, if only part of the data is needed you can use the keyword WHERE to limit the dataset used to provide results. Operators including equals (=), greater than (>), BETWEEN, LIKE and IN are used to specify a condition. AND and OR operators filter records based on more than one condition. Use single quotes around text values in a condition. The IN keyword allows you to specify discrete values in WHERE criteria, BETWEEN specifies a search range.

```
SELECT column_name(s)
```

```
FROM table_name
```

```
WHERE column_name operator value
```

```
AND/OR column_name operator value
```

Sixth, if the data that is retrieved needs to be arranged in an ascending or descending order then the keyword phrase ORDER BY can be used to indicate the field used to order the data.

```
SELECT column_name(s)
```

```
FROM table_name
```

```
ORDER BY column_name(s) [ASC|DESC]
```

Seventh, if data in a field needs to be aggregated or summarized you can use keywords like AVG(), SUM() and COUNT() to create new results. Use these analytical functions to create a new result field.

```
SELECT AVG(column_name1) FROM table_name
```

Eighth, in some situations it is necessary to group data prior to applying functions. Also, one may want to limit output based on

## *: What SQL knowledge do managers need for decision support queries?*

calculated values. The keywords GROUP BY and HAVING are used in these situations.

```
SELECT column_name1, SUM(column_name2)
```

```
FROM table_name
```

```
GROUP BY column_name1
```

```
HAVING (arithmetic function condition)
```

Ninth, SQL joins are used to query data from two or more tables, based on a relationship between certain columns in these tables. There are a number of ways to specify joining 2 or more tables. The easiest way to join 2 tables is to use the WHERE keyword and the equal operator.

```
SELECT column_name3, SUM(column_name2)
```

```
FROM table_name1, table_name2
```

```
WHERE table_name1.column_name1 = table_name2.column_name1
```

```
GROUP BY table_name1.column_name3
```

Tenth, you can nest queries, meaning having multiple queries to answer a single question. The result of one query is used by another query and you can include nested queries in WHERE and HAVING clauses. If you need to nest queries to retrieve the desired data, consider asking for help.

SQL is not case sensitive; semicolon is the standard way to separate SQL statements; SQL aliases, alternate names, can be defined for database tables and database table columns using the keyword AS.

```
SELECT column_name AS column_alias
```

## *: What SQL knowledge do managers need for decision support queries?*

FROM table\_name;

The basic SQL syntax is:

SELECT [ALL/DISTINCT] column\_list, [SUM(column\_name)]

FROM table\_name1[, table\_name2]

[WHERE conditional expression]

[GROUP BY column\_name]

[HAVING conditional expression]

[ORDER BY column\_name(s) [ASC|DESC]]

Square brackets [] indicate optional elements.

### References

A gentle introduction to SQL, URL <http://sqlzoo.net/> (test queries)

PostgreSQL 8.5 Developer Documentation, Chapter 4. SQL Syntax, <http://developer.postgresql.org/pgdocs/postgres/sql-syntax.html>

SQL Tutorial - Learn SQL, URL <http://www.sql-tutorial.net>

SQL Syntax URL <http://www.1keydata.com/sql/sql-syntax.html>

SQL Tutorial, Version 2.1, URL <http://www.firstsql.com/tutor.htm> (Advanced)

Page 4/5

(c) 2021 Daniel J. Power, Power Enterprises <power@dssresources.com>

URL: <http://dssresources.com/faq/index.php?action=artikel&cat=&id=191&artlang=en>

*: What SQL knowledge do managers need for decision support queries?*

w3schools.com, SQL Tutorial, URL <http://www.w3schools.com/SQL/default.asp>

\*\*\*\*\*

Example Query

What employees worked more than 240 hours?

```
SELECT Employee, SUM (Hours)
```

```
FROM EmployeeHours
```

```
GROUP BY Employee
```

```
HAVING SUM (Hours) > 240
```

Author: Daniel Power

Last update: 2011-03-18 09:48