

Table of Contents

1. Assignment Rules	1
2. A note about the demos	1
3. Characteristics of SQL.....	1
4. Demos	2
5. SQL syntax guidelines.....	4
5.1. Identifiers	5
6. Statement terminator	5

SQL is a language for working with relational databases. These are some of the general characteristic of the SQL language and how to write statements using the SQL language. There are some differences in writing SQL statements for the various dbms; for this class I will emphasize the features that all versions of SQL have in common so that you could more easily transfer your knowledge to another dbms. But we will address the specifics of using Oracle SQL - as opposed to MySQL or T-SQL.

1. Assignment Rules

I have posted a document for the Assignment rules which lists the rules in effect for all assignments starting with A02. This include files names, use of the template and the layout for the SQL queries.

2. A note about the demos

With most of the document files, I will also include a text file that includes the sql used for the demos. The purpose of the demo file is to give you sample sql queries that you could run and modify to try experiments. It also saves typing. You can open the demo file in a text editor and copy the sql into a client window. If you are using a gui client you can probably open the demo file in the gui client. The demo files are not intended to be run as a script- the queries are intended to be run one at a time and thought about.

Often I will display part or all of the result produced by the sql query in these documents. It is possible that the actual values may differ from the values that you get with the current dataset. The data set that I use when creating these document has the same tables as I provide, but sometimes the inserted data is different. That is ok since the query should work correctly on any data set that is currently in the tables.

The result that I post were created using either the SQL*Plus or the SQL Developer client. It is possible that if you use a different client that the result you see may have a different format- the data might be displayed in a different way. The display of a Null often differs with clients- with some clients, the cell is left empty; with other clients, the cell contains the display NULL or *Null* . Numeric values might be displayed with different numbers of digits after the decimal point. These are formatting issues- not logical issues.

3. Characteristics of SQL

- SQL works with tables (either base tables or virtual tables) and it produces virtual tables. This feature is called closure. Base tables are tables you create with the Create Table statement; the data in a base table is stored in persistent storage. A virtual table is a collection of rows and columns that the computer has in memory but it is not stored to persistent storage. The result of your query is a virtual table.
- SQL is a declarative language. You do not tell the SQL engine step by step how to produce the result. Instead you write a statement that describes the desired output table. The SQL statement that you write is passed on to the DBMS, which processes it and returns the result to the client for display.
- The same SQL statements can be used by end-users, database programmers, and database administrators. The same SQL statements can be used interactively, collected in batch files (script files), or embedded into application programs.
- There are several basic categories of SQL statements.

Query statements are the Select statement used to display data

Data Manipulation Language (DML) statements are used to manage the data within the database- this includes modifying the data.

Data Definition Language (DDL) statements are used to create and modify the design of the database objects- such as tables and relationships.

Transaction Control statements are used to make changes to the database permanent or to roll them back

Data Control Language (DCL) statements are used to assign privileges to users.

- SQL is a redundant language. There are often several different ways to accomplish the same goal.
- The SQL language has an ANSI standard; most implementations of SQL follow the standard to some degree but also add additional features and have some variations in the way that SQL is written. In this class I will emphasize the standard techniques but will also address dbms variations.

4. Demos

These are examples of SELECT queries. They are based on the zoo_2016 table. You might have additional rows in your table. You can run the queries against your tables to see the actual output. The alignment of the data might be different depending on the client you use.

The method I will usually use for displaying the SQL and the output is shown here. The SQL statement is presented in the Courier New font. The output is boxed. You should be able to copy and paste the SQL statements into your client and run them and then try variations on the SQL. In some cases, I have reduced the column widths and limited the number of rows displayed to save space. When I introduce new SQL keywords I will show them in caps; keywords we have already used will be in lower case or Initcase.

Demo 01: All rows are displayed. The attribute names are used for the column headers; the column width is determined by the data being displayed.

```
SELECT z_name, z_cost
FROM zoo_2016;
```

Z_NAME	Z_COST
Sam	5000
Abigail	490
Leon	5000
Lenora	5000
Sally Robinson	5000.25
Huey	2500.25
Dewey	2500.25
Louie	2500.25
	490
Dewey	3750
Arnold	5000
	5000
	5000
Geoff	5000
Anders	490
Anne	490.01
Leon	1850
	1850
	1850
	1850

20 rows selected.

Demo 02: Select specific columns; add a column alias, and add a criterion to limit the rows that are displayed.

```
SELECT z_name, z_cost "Price more than 3K", z_type
FROM zoo_2016
WHERE z_cost > 3000;
```

Z_NAME	Price more than 3K	Z_TYPE
Sam	5000	Giraffe
Leon	5000	Lion
Lenora	5000	Lion
Sally Robinson	5000.25	Giraffe
Dewey	3750	Giraffe
Arnold	5000	Giraffe
	5000	Giraffe
	5000	Giraffe
Geoff	5000	Giraffe

9 rows selected.

Demo 03: Select specific columns and add a criterion to limit the rows that are displayed.

```
SELECT
  z_dob
, z_type
, z_name
FROM zoo_2016
WHERE z_type = 'Giraffe';
```

Z_DOB	Z_TYPE	Z_NAME
15-MAY-14	Giraffe	Sam
15-MAY-14	Giraffe	Sally Robinson
06-JUN-13	Giraffe	Dewey
15-MAY-14	Giraffe	Arnold
15-MAY-13	Giraffe	
15-MAY-02	Giraffe	
15-MAY-02	Giraffe	Geoff

7 rows selected.

Demo 04: Rewrite the literal to match a different pattern. Oracle does string comparisons using case-specific rules.

```
SELECT z_dob, z_type, z_name
FROM zoo_2016
WHERE z_type = 'giraffe';
```

No rows selected

Demo 05: Sometimes a query does not return any rows. In that case, it does not display a header in the SQL*PLUS client.

```
SELECT z_name, z_cost "Price more than 20K", z_type
FROM zoo_2016
WHERE z_cost > 20000;
```

no rows selected

In the SQL Developer client, the display is the header but no data rows

Z_NAME	Price more than 20K	Z_TYPE
--------	---------------------	--------

Things to notice about the result table for the query result, so far:

- This is a text display.

- Column headers are the attribute names or aliases but may be truncated to fit the column width.
- Text columns are left justified; numeric columns are right justified in SQL*Plus and left justified in SQL Developer.
- Numeric values are not automatically formatted with a certain number of digits after the decimal point.
- The column widths for the text columns are determined by the defined data types.
- The column widths for the numeric columns are determined by the default value for numeric columns and the column header width.
- The appearance / format of the display is influenced by the client you are using.

5. SQL syntax guidelines

Even though there is an SQL standard, individual database systems use different dialects of the language. The version used with Oracle differs somewhat from SQL used in SQL Server and SQL used in Microsoft Access and SQL used in MySQL and from earlier versions of Oracle SQL. The SQL that you learn in this class will help you when you need to use SQL in other relational database systems.

- SQL is a free-form language. This means you could write query 2 as

```
SELECT z_name, z_cost "Price more than 2K", z_type FROM zoo_2016 WHERE z_cost > 2000;
```

But it is a lot easier to read if you start the keywords SELECT, FROM, and WHERE on new lines. For the class assignments, it is a rule that you start these keywords on new lines.
- SQL statements begin with a keyword - such as SELECT- and are composed of one or more clauses which begin with keywords such as FROM or WHERE. You should avoid using the SQL keywords as table or column names.
- Although you will commonly see SQL keywords written in upper case, you can use either upper or lower case for SQL keywords and for table and column names.
- You use commas to separate lists of items- such as the columns to be displayed.
- Literals
 - If you use a constant (a literal) in an SQL statement, it may need to be delimited.
 - Numbers do not use delimiters.
 - Do not put commas or dollar signs in numeric literals.
 - Text literals are enclosed in quotes (')
 - In Oracle, dates should be enclosed in single quotes; the default syntax for Oracle date follows the pattern 11-AUG-07. We can use other date formats.
- In Oracle, text comparisons are case-sensitive.
- You can use comments in your SQL statement. There are three forms of comments
 - The single line comment is indicated by two hyphens followed by a space. This is the ANSI standard comment.
 - The multi-line comment is delimited by /* comment */ Start with a slash, an asterisk and a space. It is essential to have a space after the /* or SQL*Plus will misinterpret that line.

To avoid problems with comments:

 - Do not put comments near the start of your SQL statement.
 - Do not put comments on the same line after the semicolon of your SQL statement.
 - Do not put a semicolon, hyphen, or ampersand (&) in your comments.

5.1. Identifiers

A table exists within a schema which exists within a database. (When you log into your Oracle account you are already in your schema). A column exists within a table. So if we want to refer to a column, we need a multi-part name.

Within your schema, the name of the table is: zoo_2016

The name of the attribute storing the names of our animals is: zoo_2016.z_name

When we are in a query that uses the zoo_2016 table in the From clause, we can refer to the attribute as zoo_2016.z_name or just as z_name. Using zoo_2016.z_name is called "qualifying" the column name.

- A qualified column name is one that includes the name of the table- such as zoo_2016.z_name.
- If your query uses only a single table, you do not need to qualify any column name.
- You need to qualify column names only if the SQL statement includes two or more tables and that column name appears in more than one of these tables.

If your query uses multiple tables, your query might be more efficient if you qualify all of the column names

Demo 06: Any of these will work

```
select z_name, z_cost
from zoo_2016;
```

```
select zoo_2016.z_name, z_cost
from zoo_2016;
```

```
select zoo_2016.z_name, zoo_2016.z_cost
from zoo_2016;
```

6. Statement terminator

Each client needs to have a way of knowing when your sql statement is complete and you want to run it. This might be indicated by a statement terminator. Often the client uses a semicolon for this. So you will commonly see SQL written as

```
Select * from zoo_2016;
```

The semicolon is not actually part of the statement but it might be required by the client in order to run the statement.

The ANSI default character used is the semicolon. In the SQL*Plus client, the use of the semicolon is required. You can have a blank line in the midst of an SQL statement in the SQL Developer client but not in the SQL*Plus client. Since SQL*Plus is used so commonly, avoid blank lines in your queries.