Table of Contents

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 MySQL - as opposed to Oracle SQL 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 the mysql command line 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 makes 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.

Optimizer: With a programming language such as Java or Visual Basic, the programmer defines the processing in a step-by-step fashion. With SQL you do not need to give such complete detailed steps. The optimizer decides on the exact statements to be run. The optimizer is a program that analyzes your SQL statement and generates the actual statements to be executed against the data. The optimizer is part of the dbms and is not part of the SQL language. The developer can specify optimizer hints that may influence the execution path that the optimizer uses.

4. Demos

These are examples of SELECT queries. They are based on the zoo_2015 table. You might have additional rows in your table. You can run the queries against your tables to see the actual output.

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 displayed in a text-style grid. 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.

Before you run any of these demos be sure you are in the correct database. The zoo_2015 table should be in the a_testbed database.

```
use a_testbed;
```

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_2015;
+---------+---------+
| z_name  | z_cost  |
+---------+---------+
| Sam     | 5000.00 |
| Abigail |  490.00 |
| Leon    | 5000.00 |
| Lenora  | 5000.00 |
| Sally   | 5000.25 |
| Huey    | 2500.25 |
| Dewey   | 2500.25 |
| Louie   | 2500.25 |
| NULL    |  490.00 |
| Dewey   | 3750.00 |
| Arnold  | 5000.00 |
```

```
| NULL    | 5000.00 |
| NULL    | 5000.00 |
| Artemis | 1500.00 |
| Diana   |  120.95 |
| Anders  |  490.00 |
| Anne    |  490.01 |
| Leon    | 1850.00 |
| NULL    | 1850.00 |
| NULL    | 1850.00 |
|         | 1850.00 |
+---------+---------+
21 rows in set (0.03 sec)
```

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_2015
WHERE z_cost > 3000;
+--------+--------------------+---------+
| z_name | Price more than 3K | z_type  |
+--------+--------------------+---------+
| Sam    |            5000.00 | Giraffe |
| Leon   |            5000.00 | Lion    |
| Lenora |            5000.00 | Lion    |
| Sally  |            5000.25 | Giraffe |
| Dewey  |            3750.00 | Giraffe |
| Arnold |            5000.00 | Giraffe |
| NULL   |            5000.00 | Giraffe |
| NULL   |            5000.00 | Giraffe |
+--------+--------------------+---------+
8 rows in set (0.00 sec)
```

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

```
SELECT z_dob, z_type, z_name
FROM zoo_2015
WHERE z_type = 'Armadillo';
+---------------------+-----------+---------+
| z_dob               | z_type    | z_name  |
+---------------------+-----------+---------+
| 2014-12-15 00:00:00 | Armadillo | Abigail |
| 2010-01-15 08:30:00 | armadillo | Anders  |
| 2010-01-15 08:30:00 | armadillo | Anne    |
+---------------------+-----------+---------+
3 rows in set (0.00 sec)
```

Demo 04: Sometimes a query does not return any rows. In that case, it does not display a header in this client

```
SELECT z_name, z_cost "Price more than 20K", z_type
FROM zoo_2015
WHERE z_cost > 20000;
Empty set (0.00 sec)
```

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.
- The column widths for the columns are determined by the data and column headers.

5. SQL syntax guidelines

Even though there is an SQL standard, individual database systems use different dialects of the language. The version used with MySQL differs somewhat from SQL used in SQL Server and SQL used in Microsoft Access and SQL used in Oracle and the version used with earlier versions of MySQL. 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 an_name, an_cost "Price more than 3K", an_type FROM zoo_2015 WHERE
an_cost > 3000;
```

  But it is a lot easier to read if you start the keywords SELECT, FROM, and WHERE 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 names 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 (with the proviso that if you are using a Linux O/S, keep the table names in lower case).
- You use commas to separate lists of items- such as the columns to be displayed.
- In MySQL, text comparisons are case-insensitive.
- 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 use commas or currency symbols in numbers
- Text literals are enclosed in quotes (')
- In MySQL, dates should be enclosed in quotes; the default syntax for date follows the pattern 2009-06-29.
- You can use comments in your SQL statement. There are two forms of comments
- The multi-line comment is delimited by /* comment */ Start with a slash, an asterisk and a space.
- The single line comment is indicated by two hyphens followed by a space. This is the ANSI standard comment.

5.1. Identifiers

A table exists within a database. (Actually a table exists within a schema- but MySQL uses the term database to refer to a schema). A column exists within a table. So if we want to refer to a column, we need a multi-part name.

The full name of the table is: `a_testbed.zoo_2015`

The full name of the attribute storing the names of our animals is: `a_testbed.zoo_2015.z_name`

This starts with the database name followed by a dot and the table name followed by a dot and the attribute name. This is also referred to as an absolute name.

If you are "in a database" then you can use a relative identifier and skip the database name if your table is in the current database. Assuming you are in the a_testbed database- you go "into" a database with the Use command.

The relative name of the table is: `zoo_2015`

The relative name of the attribute storing the names of our animals is: `zoo_2015.z_name`

When we are in a query that uses the zoo_2015 table in the From clause, we can refer to the attribute as `zoo_2015.z_name` or just as `z_name`

The identifier `zoo_2015.z_name` is referred to as a qualified column name; it includes the name of the table

- 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.

For many of the demo queries I will often use the absolute identifier, including the schema name, since that means you can run the query from within any database that you have permission to use.

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_2015;
```

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 mysql client, you could use a different delimiter, but for class we will stay with the default semicolon delimiter.

You can have a blank line in the midst of an SQL statement in the mysql client and the query will run. This is not true of all DBMS clients.