**SQL**

**What is SQL?**

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.

SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.

**Why SQL?**

SQL is widely popular because it offers the following advantages:

• Allows users to access data in the relational database management systems.

• Allows users to describe the data.

• Allows users to define the data in a database and manipulate that data.

• Allows to embed within other languages using SQL modules, libraries & pre-compilers.

• Allows users to create and drop databases and tables.

• Allows users to create view, stored procedure, functions in a database.

• Allows users to set permissions on tables, procedures and views.

**SQL Commands**

The standard SQL commands to interact with relational databases are CREATE, SELECT, INSERT, UPDATE, DELETE and DROP. These commands can be classified into the following groups based on their nature:

**DDL - Data Definition Language**

**Description & Command**

**CREATE** Creates a new table, a view of a table, or other object in the database.

**ALTER** Modifies an existing database object, such as a table.

**DROP** Deletes an entire table, a view of a table or other objects in the database.

**DML - Data Manipulation Language**

**Command Description**

**SELECT** Retrieves certain records from one or more tables.

**INSERT** Creates a record.

**UPDATE** Modifies records.

**DELETE** Deletes records

**DCL** - Data Control Language

**Command Description**

**GRANT** Gives a privilege to user.

**REVOKE** Takes back privileges granted from user.

Selecting Data

The select statement is used to query the database and retrieve selected data that match the criteria that you specify. Here is the format of a simple select statement:

select "column1"

[,"column2",etc]

from "tablename"

[where "condition"];

[] = optional

The column names that follow the select keyword determine which columns will be returned in the results. You can select as many column names that you'd like, or you can use a "\*" to select all columns.

The table name that follows the keyword from specifies the table that will be queried to retrieve the desired results.

The where clause (optional) specifies which data values or rows will be returned or displayed, based on the criteria described after the keyword where.

Conditional selections used in the where clause:

= Equal

Greater than

< Less than

= Greater than or equal

<= Less than or equal

<> Not equal to

**Inserting into a Table**

The insert statement is used to insert or add a row of data into the table.

To insert records into a table, enter the key words insert into followed by the table name, followed by an open parenthesis, followed by a list of column names separated by commas, followed by a closing parenthesis, followed by the keyword values, followed by the list of values enclosed in parenthesis.

The values that you enter will be held in the rows and they will match up with the column names that you specify.

Strings should be enclosed in single quotes, and numbers should not.

insert into "tablename"

(first\_column,...last\_column)

values (first\_value,...last\_value);

**Deleting Records**

The delete statement is used to delete records or rows from the table.

delete from "tablename"

where "columnname"

OPERATOR "value"

[and|or "column"

OPERATOR "value"];

**Creating Tables**

The create table statement is used to create a new table. Here is the format of a simple create table statement: create table "tablename"

("column1" "data type",

"column2" "data type",

"column3" "data type");

Format of create table if you were to use optional constraints:

create table "tablename"

("column1" "data type"

[constraint],

"column2" "data type"

[constraint],

"column3" "data type" [constraint]);

[ ] = optional

Note: You may have as many columns as you'd like, and the constraints are optional.

**Example:**

create table employee

(first varchar(15),

last varchar(20),

age number(3),

address varchar(30),

city varchar(20),

state varchar(20));

To create a new table, enter the keywords create table followed by the table name, followed by an open parenthesis, followed by the first column name, followed by the data type for that column, followed by any optional constraints, and followed by a closing parenthesis.

It is important to make sure you use an open parenthesis before the beginning table, and a closing parenthesis after the end of the last column definition.

Make sure you seperate each column definition with a comma. All SQL statements should end with a ";".

The table and column names must start with a letter and can be followed by letters, numbers, or underscores - not to exceed a total of 30 characters in length.

Do not use any SQL reserved keywords as names for tables or column names (such as "select", "create", "insert", etc).

Data types specify what the type of data can be for that particular column.

If a column called "Last\_Name", is to be used to hold names, then that particular column should have a "varchar" (variable-length character) data type.

**Here are the most common Data types:**

**char(size)** Fixed-length character string. Size is specified in parenthesis. Max 255 bytes.

\*\*varchar(size) \*\* Variable-length character string. Max size is specified in parenthesis.

**number(size)** Number value with a max number of column digits specified in parenthesis.

**Date** Date value

**Number(size,d)** Number value with a maximum number of digits of "size" total, with a maximum number of "d" digits to the right of the decimal.

\*\*What are constraints? \*\*

When tables are created, it is common for one or more columns to have constraints associated with them.

A constraint is basically a rule associated with a column that the data entered into that column must follow.

For example, a "unique" constraint specifies that no two records can have the same value in a particular column.

They must all be unique. The other two most popular constraints are "not null" which specifies that a column can't be left blank, and "primary key".

A "primary key" constraint defines a unique identification of each record (or row) in a table.

All of these and more will be covered in the future Advanced release of this Tutorial. Constraints can be entered in this SQL interpreter, however, they are not supported in this Intro to SQL tutorial & interpreter.

They will be covered and supported in the future release of the Advanced SQL tutorial - that is, if "response" is good.

It's now time for you to design and create your own table. You will use this table throughout the rest of the tutorial.

If you decide to change or redesign the table, you can either drop it and recreate it or you can create a completely different one.

The SQL statement drop will be covered later.

**Create Table Exercise**

You have just started a new company. It is time to hire some employees.

You will need to create a table that will contain the following information about your new employees: firstname, lastname, title, age, and salary. After you create the table, you should receive a small form on the screen with the appropriate column names.

If you are missing any columns, you need to double check your SQL statement and recreate the table. Once it's created successfully, go to the "Insert" lesson.

**IMPORTANT:**

When selecting a table name, it is important to select a unique name that no one else will use or guess.

Your table names should have an underscore followed by your initials and the digits of your birth day and month.

For example, Tom Smith, who was born on November 2nd, would name his table myemployees\_ts0211 Use this convention for all of the tables you create.

Your tables will remain on a shared database until you drop them, or they will be cleaned up if they aren't accessed in 4-5 days. If "support" is good, I hope to eventually extend this to at least one week.

When you are finished with your table, it is important to drop your table (covered in last lesson).