SQL

- SQL is a standard language for storing, manipulating and retrieving data in databases.
- SQL keywords are NOT case sensitive: select is the same as SELECT
- Some database systems require a semicolon at the end of each SQL statement.

What Can SQL do?

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views

Some of The Most Important SQL Commands

- **SELECT** - extracts data from a database
- **UPDATE** - updates data in a database
- **DELETE** - deletes data from a database
- **INSERT INTO** - inserts new data into a database
- **CREATE DATABASE** - creates a new database
- **ALTER DATABASE** - modifies a database
- **CREATE TABLE -** creates a new table
- **ALTER TABLE** - modifies a table
- **DROP TABLE** - deletes a table
- **CREATE INDEX** - creates an index (search key)
- **DROP INDEX** - deletes an index

SQL QUERIES

Create database

```
CREATE DATABASE databasename;

CREATE DATABASE SqlTutorial
```

Create table

```
CREATE TABLE table_name (
column1 datatype,
column2 datatype,
column3 datatype,
....
);
```

```
CREATE TABLE EmployeeDemographics(tablename)
(EmployeeID int,
 FirstName varchar(50),
 LastName varchar(50),
 Age int,
 Gender varchar(50)   )
```

Insert values into table

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

If you are adding values for all the columns of the table, you do not need to specify the column names in the SQL query. However, make sure the order of the values is in the same order as the columns in the table. Here, the INSERT INTO syntax would be as follows:

```
INSERT INTO table_name
VALUES (value1, value2, value3, ...);
```

```
INSERT INTO EmloyeeDemographics VALUES
(1001, 'Jim',30,'Male')
```

Select rows from table

1. SELECT all records in a table

```
SELECT * FROM tablename;
```

```
SELECT * FROM EmployeeDemographics
```

2. SELECT particular column from a table

```
SELECT column1,column2 FROM table_name;

select FirstName, LastName from EmployeeDemographics
```

3. SELECT TOP

The SELECT TOP clause is used to specify the number of records to return.

The SELECT TOP clause is useful on large tables with thousands of records. Returning a large number of records can impact performance.

```
SELECT TOP number * column_name(s)
FROM table_name
WHERE condition;

SELECT TOP 5 * FROM EmployeeDemographics
```

4. **SELECT DISTINCT values from one or more column**

The **SELECT DISTINCT** statement is used to return only distinct (different) or unique values in a specific column

```
SELECT DISTINCT (columnname) FROM table_name;

SELECT DISTINCT FirstName FROM EmployeeDemographics;
```

5. SELECT COUNT

Returns count of non null value from a specific column
The COUNT() function returns the number of rows that matches a specified criterion.

```
SELECT COUNT(column_name)
FROM table_name
WHERE condition;

SELECT COUNT(*) FROM EmployeeDemographics;

SELECT COUNT(LastName) FROM EmployeeDemographics; (Result will show no column name so use below query with AS )

SELECT COUNT(LastName) AS LastName FROM EmployeeDemographics;
```

6. SELECT records using where clause

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;

SELECT FirstName, LastName
FROM EmployeeDemographics
WHERE age= 40;
```