

Introduction to MS SQL

Lecture#1

Contents

- Install MS SQL Server
- What is SQL?

SQL Language

- SQL (Structured Query Language) is used to perform operations on the records stored in the database, such as updating records, inserting records, deleting records, creating and modifying database tables, views, etc.
- SQL is not a database system, but it is a query language.

Suppose you want to perform the queries of SQL language on the stored data in the database. You are required to install any database management system, for example, Oracle, MySQL, MongoDB, PostgreSQL, SQL Server,

What is SQL Server?

- SQL Server is a relational database management system (RDBMS) developed and marketed by Microsoft.
- As a database server, the primary function of the SQL Server is to store and retrieve data used by other applications.
- Similar to other RDBMS software, SQL Server is built on top of [SQL](#), a standard programming language for interacting with relational databases.
- SQL Server is tied to Transact-SQL, or T-SQL, the Microsoft's implementation of SQL that adds a set of proprietary programming constructs.

- SQL Server works exclusively on the Windows environment for more than 20 years.
- In 2016, Microsoft made it available on Linux. SQL Server 2017 became generally available in October 2016 that ran on both Windows and Linux.

Install MS SQL Server Management Studio

- Website for installing

<https://www.sqlservertutorial.net/install-sql-server/>

SQL Server components:

- SQL Server consists of two main components:
 - (1) Database Engine
 - (2) SQLLOS

Database Engine

- The core component of the SQL Server is the Database Engine.
- The Database Engine consists of a relational engine that processes queries and a storage engine that manages database files, pages, indexes, etc.
- The database objects such as [stored procedures](#), [views](#), and [triggers](#) are also created and executed by the Database Engine.

Relational Engine

- The Relational Engine contains the components that determine the best way to execute a query.
- The relational engine is also known as the **query processor**.
- The relational engine requests data from the storage engine based on the input query and processed the results.
- Some tasks of the relational engine include querying processing, memory management, thread and task management, buffer management, and distributed query processing.

Storage Engine

- The storage engine is in charge of storage and retrieval of data from the storage systems such as disks.

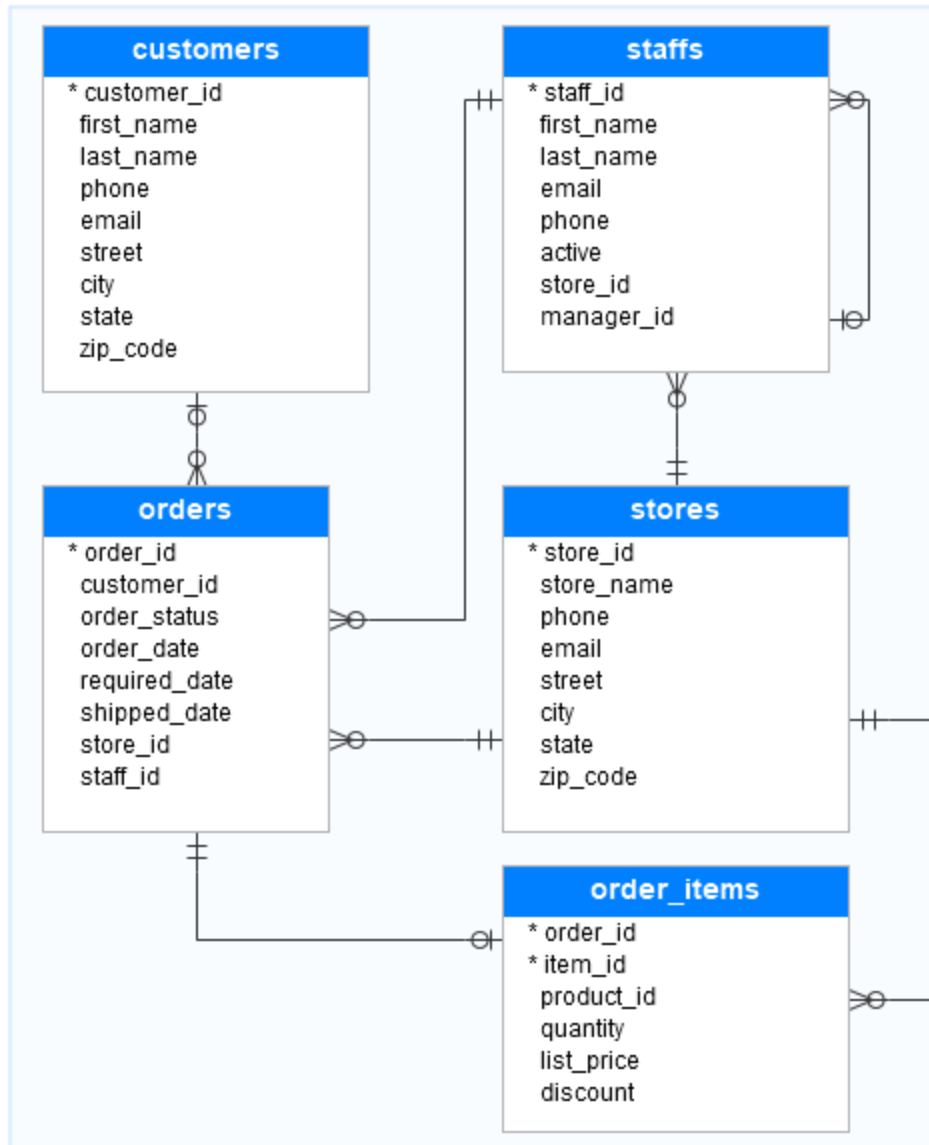
SQLLOS

- Under the relational engine and storage engine is the SQL Server Operating System or SQLLOS.
- SQLLOS provides many operating system services such as memory and I/O management. Other services include exception handling and synchronization services.

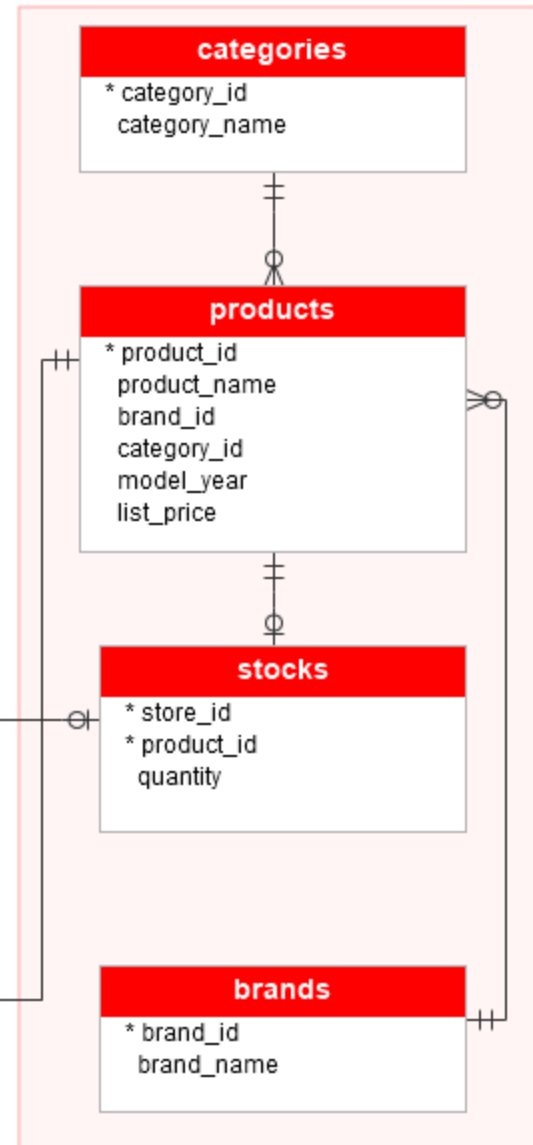
SQL Server Sample Database

- The following illustrates the BikeStores database diagram:
-

Sales



Production



- As you can see from the diagram, the BikeStores sample database has two schemas **sales** and **production**, and these schemas have nine tables.

Database Tables

- Table sales.stores
- The sales.stores table includes the store's information. Each store has a **store name**, **contact information** such as phone and email, and an **address** including street, city, state, and zip code.


```
CREATE TABLE sales.stores (  
    store_id INT IDENTITY (1, 1) PRIMARY KEY,  
    store_name VARCHAR (255) NOT NULL,  
    phone VARCHAR (25),  
    email VARCHAR (255),  
    street VARCHAR (255),  
    city VARCHAR (255),  
    state VARCHAR (10),  
    zip_code VARCHAR (5)  
);
```

Table sales.staffs

- The sales.staffs table stores the essential information of staffs including first name, last name. It also contains the communication information such as email and phone.
- A staff works at a store specified by the value in the store_id column. A store can have one or more staffs.
- A staff reports to a store manager specified by the value in the manager_id column. **If the value in the manager_id is null**, then the staff is the top manager.
- If a **staff no longer works** for any stores, the value in the active column is set to **zero**.

```
CREATE TABLE sales.staffs (  
    staff_id INT IDENTITY (1, 1) PRIMARY KEY,  
    first_name VARCHAR (50) NOT NULL,  
    last_name VARCHAR (50) NOT NULL,  
    email VARCHAR (255) NOT NULL UNIQUE,  
    phone VARCHAR (25),  
    active tinyint NOT NULL,  
    store_id INT NOT NULL,  
    manager_id INT,  
    FOREIGN KEY (store_id)  
    REFERENCES sales.stores (store_id)  
    ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (manager_id)  
    REFERENCES sales.staffs (staff_id)  
    ON DELETE NO ACTION ON UPDATE NO ACTION  
);
```

Table production.categories

- The production.categories table stores the bike's categories such as children bicycles, comfort bicycles, and electric bikes.

```
CREATE TABLE production.categories (  
    category_id INT IDENTITY (1, 1) PRIMARY KEY,  
    category_name VARCHAR (255) NOT NULL  
);
```

SQL Syntax

- <https://www.javatpoint.com/sql-syntax>

Types of SQL commands

Important points about the SQL syntax

- You can write the keywords of SQL in both uppercase and lowercase, but writing the SQL keywords in uppercase improves the readability of the SQL query.
- SQL statements or syntax are dependent on text lines. We can place a single SQL statement on one or multiple text lines.
- You can perform most of the action in a database with SQL statements.
- SQL syntax depends on relational algebra and tuple relational calculus.

Simple Example of SQL statement:

```
SELECT "column_name" FROM "table_name";
```

Each SQL statement begins with any of the SQL keywords and ends with the semicolon (;). The semicolon is used in the SQL for separating the multiple Sql statements which are going to execute in the same call. In this SQL tutorial, we will use the semicolon (;) at the end of each SQL query or statement.

Most Important SQL Commands and Statements

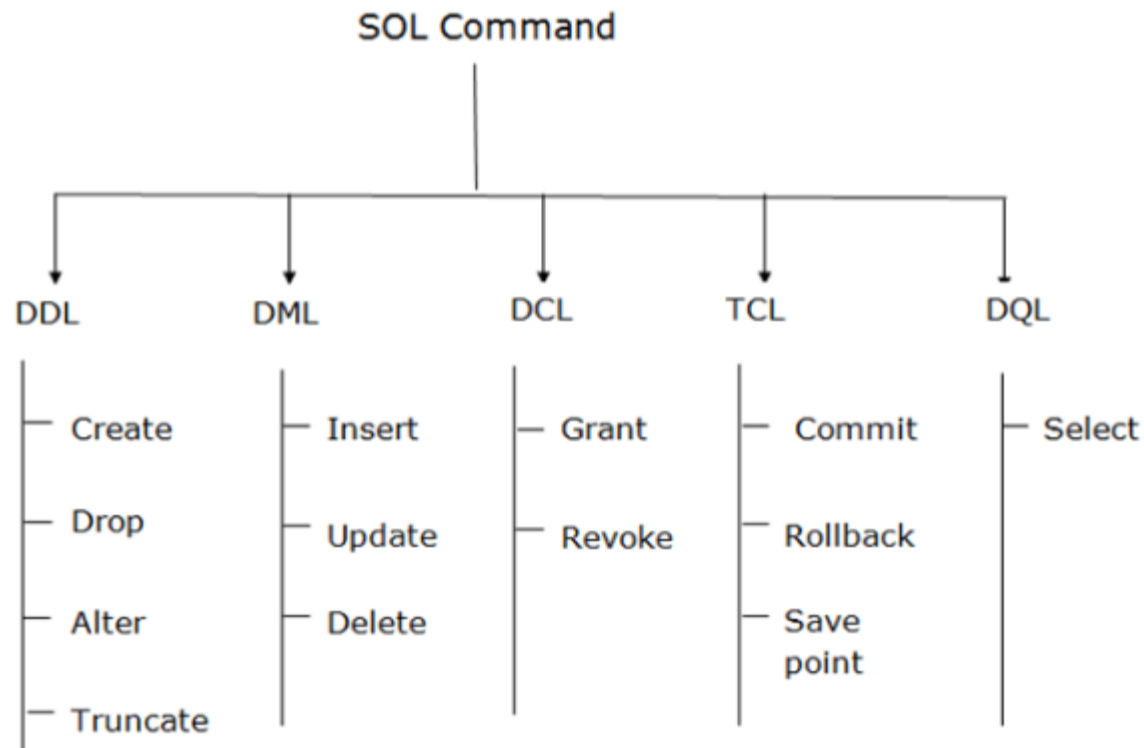
1. Select Statement
2. Update Statement
3. Delete Statement
4. Create Table Statement
5. Alter Table Statement
6. Drop Table Statement
7. Create Database Statement
8. Drop Database Statement
9. Insert Into Statement
10. Truncate Table Statement
11. Describe Statement
12. Distinct Clause
13. Commit Statement
14. Rollback Statement
15. Create Index Statement

Types of SQL commands

- SQL commands are instructions. It is used to communicate with the database. It is also used to perform specific tasks, functions, and queries of data.
- SQL can perform various tasks like create a table, add data to tables, drop the table, modify the table, set permission for users.

Types of SQL Commands

- There are five types of SQL commands: DDL, DML, DCL, TCL, and DQL.



Data Definition Language (DDL)

- DDL changes the structure of the table like creating a table, deleting a table, altering a table, etc.
- All the command of DDL are auto-committed that means it permanently save all the changes in the database

Here are some commands that come under DDL:

- CREATE
- ALTER
- DROP
- TRUNCATE

Commands under DDL

- It is used to create a new table in the database.

```
CREATE TABLE TABLE_NAME (COLUMN_NAME DATATYPES[,.....]);
```

DROP: It is used to delete both the structure and record stored in the table.

```
DROP TABLE table_name;
```



Commands under DDL

- **ALTER:** It is used to alter the structure of the database. This change could be either to modify the characteristics of an existing attribute or probably to add a new attribute.

To add a new column in the table

```
ALTER TABLE table_name ADD column_name COLUMN-definition;
```

- To modify existing column in the table:

```
ALTER TABLE table_name MODIFY(column_definitions...);
```

EXAMPLE

```
ALTER TABLE STU_DETAILS ADD(ADDRESS VARCHAR2(20));  
ALTER TABLE STU_DETAILS MODIFY (NAME VARCHAR2(20));
```


- **TRUNCATE:** It is used to delete all the rows from the table and free the space containing the table.

```
TRUNCATE TABLE table_name;
```

Example:

```
TRUNCATE TABLE EMPLOYEE;
```

◦ INSERT Data Manipulation Language

◦ UPDATE

◦ DELETE

- DML commands are used to modify the database. It is responsible for all form of changes in the database.
- The command of DML is not auto-committed that means it can't permanently save all the changes in the database. They can be rollback.

DML Commands

- **INSERT:** The INSERT statement is a SQL query. It is used to insert data into the row of a table.

Syntax:

```
INSERT INTO TABLE_NAME  
(col1, col2, col3,.... col N)  
VALUES (value1, value2, value3, .... valueN);
```

For example:

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

```
INSERT INTO javatpoint (Author, Subject) VALUES ("Sonoo", "DBMS");
```

- **UPDATE:** This command is used to update or modify the value of a column in the table.

Syntax:

```
UPDATE table_name SET [column_name1= value1,...column_nameN = valueN] [W
```

For example:

```
UPDATE students  
SET User_Name = 'Sonoo'  
WHERE Student_Id = '3'
```

- **DELETE:** It is used to remove one or more row from a table.

Syntax:

```
DELETE FROM table_name [WHERE condition];
```

For example:

```
DELETE FROM javatpoint  
WHERE Author="Sonoo";
```

Data Control Language

- DCL commands are used to grant and take back authority from any database user.
- Here are some commands that come under DCL:
 - Grant
 - Revoke

DCL Commands

- **Grant:** It is used to give user access privileges to a database.

```
GRANT SELECT, UPDATE ON MY_TABLE TO SOME_USER, ANOTHER_USER;
```

- **Revoke:** It is used to take back permissions from the user.

Example

```
REVOKE SELECT, UPDATE ON MY_TABLE FROM USER1, USER2;
```

Transaction Control Language

- TCL commands can only use with DML commands like INSERT, DELETE and UPDATE only.
- These operations are automatically committed in the database that's why they cannot be used while creating tables or dropping them.
- Here are some commands that come under TCL:
- COMMIT
- ROLLBACK
- SAVEPOINT

- **Commit:** Commit command is used to save all the transactions to the database.

Syntax:

```
COMMIT;
```

Example:

```
DELETE FROM CUSTOMERS  
WHERE AGE = 25;  
COMMIT;
```

- **Rollback:** Rollback command is used to undo transactions that have not already been saved to the database.

Syntax:

```
ROLLBACK;
```

Example:

```
DELETE FROM CUSTOMERS  
WHERE AGE = 25;  
ROLLBACK;
```

- **SAVEPOINT:** It is used to roll the transaction back to a certain point without rolling back the entire transaction.

Syntax:

```
SAVEPOINT SAVEPOINT_NAME;
```

Data Query Language

- DQL is used to fetch the data from the database.
- It uses only one command:
- **SELECT**
- **SELECT:** This is the same as the projection operation of relational algebra. It is used to select the attribute based on the conditions described by **WHERE** clause.

Syntax:

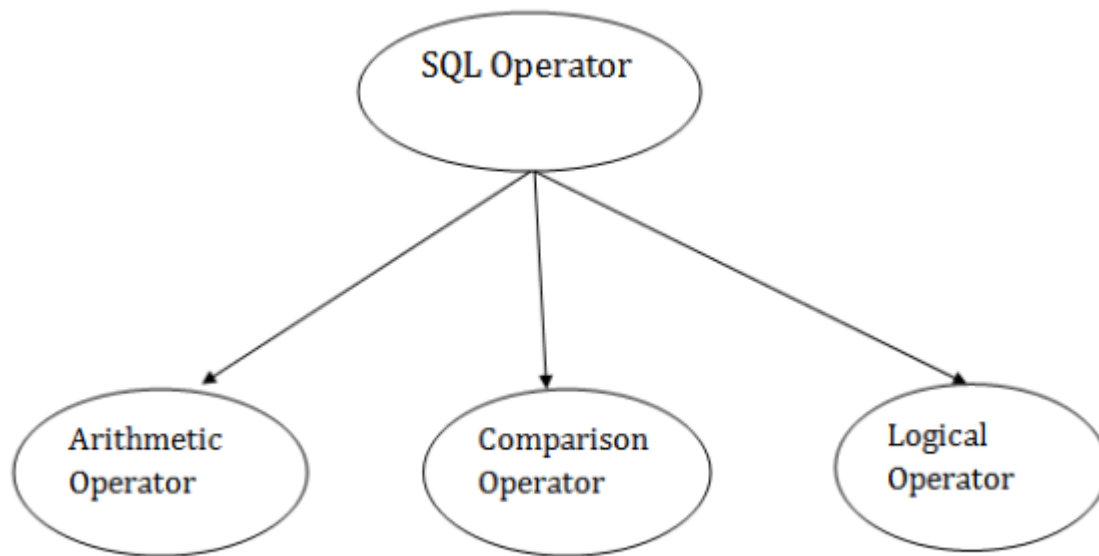
```
SELECT expressions  
FROM TABLES  
WHERE conditions;
```

For example:

```
SELECT emp_name  
FROM employee  
WHERE age > 20;
```

SQL Operator

- There are various types of SQL operator:
-



SQL Arithmetic Operators

- Let's assume 'variable a' and 'variable b'. Here, 'a' contains 20 and 'b' contains 10.

- | Operator | Description | Example |
|----------|---|-------------------|
| + | It adds the value of both operands. | a+b will give 30 |
| - | It is used to subtract the right-hand operand from the left-hand operand. | a-b will give 10 |
| * | It is used to multiply the value of both operands. | a*b will give 200 |
| / | It is used to divide the left-hand operand by the right-hand operand. | a/b will give 2 |
| % | It is used to divide the left-hand operand by the right-hand operand and returns remainder. | a%b will give 0 |

SQL Comparison Operators:

Operator	Description	Example
=	It checks if two operands values are equal or not, if the values are equal then condition becomes true.	(a=b) is not true
!=	It checks if two operands values are equal or not, if values are not equal, then condition becomes true.	(a!=b) is true
<>	It checks if two operands values are equal or not, if values are not equal then condition becomes true.	(a<>b) is true
>	It checks if the left operand value is greater than right operand value, if yes then condition becomes true.	(a>b) is not true
<	It checks if the left operand value is less than right operand value, if yes then condition becomes true.	(a<b) is true
>=	It checks if the left operand value is greater than or equal to the right operand value, if yes then condition becomes true.	(a>=b) is not true

<	It checks if the left operand value is less than right operand value, if yes then condition becomes true.	(a<b) is true
>=	It checks if the left operand value is greater than or equal to the right operand value, if yes then condition becomes true.	(a>=b) is not true
<=	It checks if the left operand value is less than or equal to the right operand value, if yes then condition becomes true.	(a<=b) is true
!<	It checks if the left operand value is not less than the right operand value, if yes then condition becomes true.	(a!=b) is not true
!>	It checks if the left operand value is not greater than the right operand value, if yes then condition becomes true.	(a!>b) is true

SQL Logical Operators

- There is the list of logical operator used in SQL:

Operator	Description
ALL	It compares a value to all values in another value set.
AND	It allows the existence of multiple conditions in an SQL statement.
ANY	It compares the values in the list according to the condition.
BETWEEN	It is used to search for values that are within a set of values.
IN	It compares a value to that specified list value.
NOT	It reverses the meaning of any logical operator.
OR	It combines multiple conditions in SQL statements.
EXISTS	It is used to search for the presence of a row in a specified table.
LIKE	It compares a value to similar values using wildcard operator.

SQL Table

- SQL Table is a collection of data which is organized in terms of rows and columns. In DBMS, the table is known as relation and row as a tuple.
- Table is a simple form of data storage. A table is also considered as a convenient representation of relations.

Let's see an example of the **EMPLOYEE** table:

EMP_ID	EMP_NAME	CITY	PHONE_NO
1	Kristen	Washington	7289201223
2	Anna	Franklin	9378282882
3	Jackson	Bristol	9264783838
4	Kellan	California	7254728346
5	Ashley	Hawaii	9638482678

In the above table, "EMPLOYEE" is the table name, "EMP_ID", "EMP_NAME", "CITY", "PHONE_NO" are the column names. The combination of data of multiple columns forms a row, e.g., 1, "Kristen", "Washington" and 7289201223 are the data of one row.

Operation on Table

1. Create table
2. Drop table
3. Delete table
4. Rename table

```
create table "table_name"  
("column1" "data type",  
"column2" "data type",  
"column3" "data type",  
...  
"columnN" "data type");
```

SQL Create Table

SQL create table is used to create a table in the database. To define the table, you should define the name of the table and also define its columns and column's data type.

Example

```
SQL> CREATE TABLE EMPLOYEE (  
EMP_ID INT NOT NULL,  
EMP_NAME VARCHAR (25) NOT NULL,  
PHONE_NO INT NOT NULL,  
ADDRESS CHAR (30),  
PRIMARY KEY (ID)  
);
```

If you create the table successfully, you can verify the table by looking at the message by the SQL server. Else you can use DESC command as follows:

SQL> DESC EMPLOYEE;

Field	Type	Null	Key	Default	Extra
EMP_ID	int(11)	NO	PRI	NULL	
EMP_NAME	varchar(25)	NO		NULL	
PHONE_NO	NO	int(11)		NULL	
ADDRESS	YES			NULL	char(30)

- o 4 rows in set (0.35 sec)

Drop table

A SQL drop table is used to delete a table definition and all the data from a table. When this command is executed, all the information available in the table is lost forever, so you have to very careful while using this command.

Syntax

```
DROP TABLE "table_name";
```

```
SQL> DROP TABLE EMPLOYEE;
```

SQL DELETE table

In SQL, DELETE statement is used to delete rows from a table. We can use WHERE condition to delete a specific row from a table. If you want to delete all the records from the table, then you don't need to use the WHERE clause.

Syntax

```
DELETE FROM table_name WHERE condition;
```


Suppose, the EMPLOYEE table having the following records:

EMP_ID	EMP_NAME	CITY	PHONE_NO	SALARY
1	Kristen	Chicago	9737287378	150000
2	Russell	Austin	9262738271	200000
3	Denzel	Boston	7353662627	100000
4	Angelina	Denver	9232673822	600000
5	Robert	Washington	9367238263	350000
6	Christian	Los angels	7253847382	260000

The following query will DELETE an employee whose ID is 2.

```
SQL> DELETE FROM EMPLOYEE  
WHERE EMP_ID = 3;
```

If you don't specify the WHERE condition, it will remove all the rows from the table.

```
DELETE FROM EMPLOYEE;
```

Now, the EMPLOYEE table would not have any records.

SQL SELECT Statement

- In SQL, the SELECT statement is used to query or retrieve data from a table in the database. The returned data is stored in a table, and the result table is known as result-set.

Syntax

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

Here, the expression is the field name of the table that you want to select data from.

Use the following syntax to select all the fields available in the table:

```
SELECT * FROM table_name;
```

To fetch the EMP_ID of all the employees, use the following query:

```
SELECT EMP_ID FROM EMPLOYEE;
```

EMP_ID
1
2
3
4
5

To fetch the EMP_NAME and SALARY, use the following query:

```
SELECT EMP_NAME, SALARY FROM EMPLOYEE;
```

EMP_NAME	SALARY
Kristen	150000
Russell	200000
Angelina	600000
Robert	350000
Christian	260000

To fetch all the fields from the EMPLOYEE table, use the following query:

```
SELECT * FROM EMPLOYEE
```

EMP_ID	EMP_NAME	CITY	PHONE_NO	SALARY
1	Kristen	Chicago	9737287378	150000
2	Russell	Austin	9262738271	200000
3	Angelina	Denver	9232673822	600000
4	Robert	Washington	9367238263	350000
5	Christian	Los angels	7253847382	260000

