



# Master In SQL

## By Fireblaze AI School

# Index

- **Introduction to the Program**
- **Installation Of Oracle SQL**
- **Data Retrieving**
- **Restricting and Sorting Data**
- **Using Single-Row Functions to Customize Output**
- **Using Conversion Functions and Conditional Expressions**
- **Aggregated Data Using the Group Functions**
- **Displaying Data from Multiple Tables Using Joins**
- **Using Subqueries to Solve Queries**
- **Using the Set Operators**
- **Manipulating the Data**
- **Using DDL Statements to Create and Manage Tables**

# Manipulating Data

- Adding new rows in a table.
- INSERT statement.
- UPDATE statement
- DELETE statement
- TRUNCATE statement
- Database transaction

# Add New Row In Table

- Add new rows to a table by using the INSERT statement

- **Syntax:-**

INSERT INTO table [(column [, column...])] VALUES (value [, value...]);

- With this syntax, only one row is inserted at a time.
- You can add new rows to a table by issuing the INSERT statement.  
In the syntax: table column value Is the name of the table Is the name of the column in the table to populate Is the corresponding value for the column.

# Inserting New Rows

- Insert a new row containing values for each column.
- List values in the default order of the columns in the table.
- Optionally, list the columns in the INSERT clause.

## **Syntax:-**

Insert Into Table\_name (Column\_name1,Column\_2)

Values ( Value1, Value2);

# UPDATE Statement Syntax

Modify existing values in a table with the UPDATE statement:

Syntax:- UPDATE table

SET column = value [, column = value, ...]

[WHERE condition];

Update more than one row at a time (if required).

# Delete Statement

You can remove existing rows from a table by using the DELETE statement:

## **Syntax:-**

DELETE

[FROM] table

[WHERE condition];

# TRUNCATE Statement

**Removes all rows from a table, leaving the table empty and the table structure intact.**

- Is a data definition language (DDL) statement rather than a DML statement; cannot easily be undone.
- **Syntax:**  
Truncate Table Table\_Name;



# Database Transaction

With COMMIT and ROLLBACK statements, you can:

- Ensure data consistency
- Preview data changes before making changes permanent
- Group logically-related operations

# Commit, Savepoint, Rollback

Statement	Description
<b>COMMIT</b>	COMMIT ends the current transaction by making all pending data changes permanent
<b>SAVEPOINT name</b>	SAVEPOINT name marks a savepoint within the current transaction.
<b>ROLLBACK</b>	ROLLBACK ends the current transaction by discarding all pending data changes.