

SQL - Structured Query Language

CBSE
Vidyalekhan
AGE

- SQL is a standard language for storing, manipulating and retrieving data in databases.
- RDBMS stands for Relational Database Management System.
 - ↳ stored in database objects called Tables.
- A record is a horizontal entity in a table.
 - ↳ have fields
- SQL keywords are NOT case sensitive.
- The SELECT statement is used to select data from a database.

SELECT CustomerName, City FROM Customers;

- The SELECT DISTINCT statement is used to return only distinct values.

SELECT DISTINCT Country FROM Customers;

- By using the DISTINCT keyword in a function called COUNT, we can return the number of different countries.

SELECT COUNT(DISTINCT Country) FROM Customers;

- The WHERE clause is used to filter records.

SELECT * FROM Customers
WHERE Country = 'Mexico';

→ SQL Requires single quote around text values.
• However, numeric field should not be enclosed
in quotes.

→ Operators in the WHERE clause
= , > , < , >= , <= , <> , != , BETWEEN , LIKE , IN

→ The ORDER BY keyword is used to sort the results
in ascending or descending order.

```
SELECT * FROM Products  
ORDER BY Price;
```

→ DESC for descending order.

→ ORDER BY several columns

```
SELECT * FROM Customers  
ORDER BY Country , CustomerName;
```

→ The AND operator is used to filter records based on
more than one condition , like if you want to
return all customers from Spain that starts with
the letter 'C';

```
SELECT * FROM Customers  
WHERE Country = 'Spain' AND CustomerName  
LIKE 'C%';
```

→ The OR operator is used to filter records based on more
than one condition , like if you want to return all
customers from Germany but also those from Spain

```
SELECT * FROM Customers  
WHERE Country = 'Germany' OR  
Country = 'Spain';
```

- > The NOT operator is used in combination with other operators to give the opposite result, also called the negative result.

```
SELECT * FROM customers  
WHERE NOT Country = 'Spain';
```

- > The INSERT INTO statement is used to insert new records in a table.

```
INSERT INTO table-name (column1, column2, .... )  
VALUES (value1, value2, value3, ... );
```

- > A field with a NULL value is a field with no value.
 - > IS NULL
 - > IS NOT NULL

```
SELECT CustomerName, ContactName, Address  
FROM Customers  
WHERE Address IS NULL;
```

- > The UPDATE statement is used to modify the existing records in a table.

```
UPDATE table-name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;  
-> Update multiple records.
```

- > Be careful when updating records, If you omit the WHERE clause, ALL Records will be updated.

- > The DELETE statement is used to delete existing records in a table.

DELETE FROM table-name
WHERE condition;

- If you omit the WHERE clause, all records in the table will be deleted.
- To delete the table completely, use the DROP TABLE statement:

DROP TABLE Customers;

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

SELECT TOP 3 * FROM Customers;

- In MySQL it uses LIMIT.

SELECT * FROM Customers LIMIT 3;

- The MIN() function returns the smallest value of the selected column.
- The MAX() function returns the largest value of the selected column.

SELECT MIN(Price), MAX(Price)
FROM Products;

- The COUNT(*) function returns the number of rows that matches a specified criterion.

SELECT COUNT(*)
FROM Products;

- Ignore duplicates using DISTINCT keyword
- The SUM() function returns the total sum of numeric column.

SELECT SUM(Quantity) FROM OrderDetails;

- Sum with an expression

SUM(Quantity * 10)

- The AVG() function returns the average value of a numeric column.

SELECT AVG(Price) FROM Products;

- The SQL LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

- The percent sign % represents zero, one or multiple characters
- The underscore sign _ represents one, single characters

SELECT * FROM Customers
WHERE CustomerName LIKE 'a%';

- A wildcard character is used to substitute one or more characters in a string.
- The IN operator allows you to specify multiple values in a WHERE clause.
 - The IN operator is a shorthand for multiple OR conditions.

```
SELECT * FROM Customers  
WHERE Country IN ('Germany', 'France', 'UK');
```

- The BETWEEN operator selects values within a given range.
- The BETWEEN operator is inclusive: begin and end values are included.

```
SELECT * FROM Products  
WHERE Price BETWEEN 10 AND 20;
```

→ NOT BETWEEN

```
SELECT * FROM Products  
WHERE ProductName BETWEEN 'Carnarvon Tigers'  
AND 'Mozzarelli di Giovanni' ORDER BY ProductName;
```

→ BETWEEN Dates

→ SQL Aliases are used to give a table, or a column in a table, a temporary name.

```
SELECT CustomerID AS ID  
FROM Customers;
```

→ Actually, in most database languages, you can skip the AS keyword and get the same result.

→ Using alias with a space character

```
ProductName AS [My Great Products]  
AS "My Great Products";
```

→ Concatenate Columns

→ `SELECT customerName, CONCAT(Address, ', ',
PostuiCode, ', ', city, ', ', Country) AS Address
FROM customers;`

→ A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

```
SELECT Orders.OrderID, Customers.CustomerName,  
Orders.OrderDate  
FROM Orders  
INNER JOIN Customers  
ON Orders.CustomerID = Customers.CustomerID;
```

→ Different types of SQL Joins

- (INNER) JOIN : Returns records that have matching values in both tables.
- LEFT (OUTER) JOIN : Returns all records from the left table, and the matched records from the right table.
- RIGHT (OUTER) JOIN : Returns all records from the right table, and the matched records from the left table.
- FULL (OUTER) JOIN : Returns all records when there is a match in either left or right table.

```
SELECT column-name(s)  
FROM table1
```

```
LEFT JOIN table2  
ON table1.column-name = table2.column-name;
```

```
SELECT column-name(s)  
FROM table1  
RIGHT JOIN table2
```

ON table1.column-name = table2.column-name;

SELECT column-name(s)

FROM table1

FULL OUTER JOIN table2

ON table1.column-name = table2.column-name,

WHERE condition;

- A self join is a regular join, but the table is joined with itself.

SELECT A.CustomerName AS CustomerName1,
B.CustomerName AS CustomerName2,
A.City

FROM Customers A, Customers B

WHERE A.CustomerID <> B.CustomerID

AND A.City = B.City

ORDER BY A.City;

- The UNION operator is used to combine the result-set of two or more SELECT statements.

SELECT column-name(s) FROM table1

UNION

SELECT column-name(s) FROM table2;

- Every select statement within UNION must have the same number of columns

- The columns must also have similar data types

- The columns in every SELECT statement must also be in the same order.

- The UNION operator selects only distinct values by default. To allow duplicate values, use UNION ALL:

→ The GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each country".

```
SELECT column-name(s)
FROM table-name
WHERE condition
GROUP BY column-name(s)
ORDER BY column-name(s);
```

```
SELECT COUNT(customerID), Country
FROM Customers
GROUP BY Country;
```

→ The HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate functions.

```
SELECT COUNT(customerID), Country
FROM Customers
GROUP BY Country
HAVING COUNT(customerID) > 5;
```

→ The EXISTS operator is used to test for the existence of any record in a Subquery.

→ The EXISTS operator returns TRUE if the Subquery returns one or more records.

```
SELECT column-name(s)
FROM table-name
WHERE EXISTS
(SELECT column-name FROM table-name WHERE
condition);
```

- The ANY and ALL operators allow you to perform comparison between a single column value and a range of other values.
- The ANY means that the condition will be true if the operation is true for any of the values in the range.

```
SELECT column-name(s)
FROM table-name
WHERE column-name operator ANY / ALL
  (SELECT column-name
   FROM table-name
   WHERE condition);
```

- ALL means that the condition will be true only if the operation is true for all values in the range.

- The SELECT INTO statement copies data from one table into a new table.

```
SELECT *
INTO newtable [ IN external DB ]
FROM oldtable
WHERE condition;
```

- The SQL INSERT INTO SELECT statement copies data from one table and insert it into another table.

```
INSERT INTO table2
SELECT * FROM table1
WHERE condition;
```

- The CASE expression goes through conditions and

returns a value when the first condition is met. so, once a condition is true, it will stop reading and return result. If no conditions are true, it returns the value in the ELSE part.

CASE

```
WHEN condition1 THEN result1
WHEN condition2 THEN result2
ELSE result
END;
```

- The MySQL IFNULL() function lets you return an alternative value if an expression is NULL

```
SELECT ProductName, UnitPrice * (UnitsInStock
+ IFNULL(UnitsOnOrder, 0))
FROM Products;
```

→ COALESCE()

- A stored procedure is a prepared SQL code that you can save, so the code can be reused over and over again.

→ CREATE PROCEDURE procedure-name

AS

SQL-statement

GO;

EXEC procedure-name;

→ CREATE PROCEDURE SelectAllCustomers (@city nvarchar(30))
AS

SELECT * FROM Customers WHERE city = @city
GO;

EXEC selectAllCustomerss @city = 'London';

- Comments are used to explain sections of SQL statements or to prevent execution of SQL statement.

→ Single-line Comment

--

→ Multi-line Comments /*....*/

- SQL operators

- Arithmetic operators : +, -, *, /, %

- Bitwise operators : &, |, ^

- Comparison operators : =, >, <, >=, <=, <>

- Compound operators : +=, -=, *=, /=, %=, &=

^-=, |==

- Logical operators : ALL, ANY, BETWEEN, EXISTS
IN, LIKE, NOT, OR, SOME

⇒

- The CREATE DATABASE statement is used to create a new SQL Database.

CREATE DATABASE database-name;
SHOW DATABASES;

- The DROP DATABASE statement is used to drop an existing SQL database.

DROP DATABASE database-name;

- The CREATE TABLE statement is used to create a new table in a database.

CREATE TABLE table-name (
column1 datatype,

columns datatype,
....

);

→ Create table using another table.

```
CREATE TABLE new-table-name AS  
SELECT column1, column2, ...  
FROM existing-table-name  
WHERE ...;
```

→ The DROP TABLE Statement is used to drop an existing table in a database.

```
DROP TABLE table-name;
```

→ The TRUNCATE TABLE statement is used to delete data inside a table, but not the table itself.

④ TRUNCATE TABLE Shippers;

→ The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

→ is also used to add and drop various constraints on an existing table.

```
ALTER TABLE customers
```

```
ADD Email VARCHAR(255);
```

```
ALTER TABLE customers
```

```
DROP column Email;
```

```
ALTER TABLE table-name
```

```
RENAME column old-name to new-name;
```

ALTER TABLE table-name
MODIFY COLUMN column-name datatype;

- SQL constraints are used to specify rules for data in a table.
- column datatype constraint
- NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK, DEFAULT, CREATE INDEX
- The NOT NULL constraint enforces a column to accept NULL values.

ALTER TABLE Persons

MODIFY COLUMN Age int NOT NULL;

- The UNIQUE constraint ensures that all values in a column are different.

ALTER TABLE Persons

ADD UNIQUE (ID);

- The PRIMARY KEY constraint uniquely identifies each record in a table.
- A FOREIGN KEY constraint is used to a field in one table, that refers to the PRIMARY KEY in another table.

FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)

- The CHECK constraint is used to limit the value that can be placed in a column.
- The DEFAULT constraint is used to set a default value.

for a column.

City varchar(25) DEFAULT 'Sunday'

→ The CREATE INDEX statement is used to create indexes in table.

UNIQUE
CREATE INDEX index-name
ON table-name (column1, column2, ...);

→ DROP INDEX statement is used to delete an index in a table.

DROP INDEX index-name ON table-name;

→ Auto increment allows a unique number to be generated automatically when a new record is inserted into a table.

Person-ID int NOT NULL AUTO_INCREMENT,

ALTER TABLE Persons AUTO_INCREMENT = 100;

→ In SQL, a view is a virtual table based on the result-set of an SQL statement.

CREATE VIEW view-name AS
SELECT column1, column2, ...
FROM table-name
WHERE condition;

→ A view can be updated with the CREATE OR REPLACE VIEW statement.

→ DROP VIEW view-name;

- > SQL Injection is a code injection technique that might destroy your database.
- > Is one of the most common web hacking techniques.
- > If you want your web site to be able to store and retrieve data from a database, your web server should have access to a database system that uses the SQL language.