Structured Query Language (SQL)

Database is a collection of tables used to store data in terms of tables. A table consists of rows and columns which represents the attributes. There can be a relationship between the tables or may not be. SQL stands for Structured Query Language used for Creating, Reading, Updating, storing, and retrieving data in databases.

SQL Operations on Database

Execute	Queries
Create new	Databases, Tables, Records, Stored Procedures, Views
Update records of the Table	Update
Data Manipulation	Update
Set permissions for	Databases, Tables, Procedures, And Views

Create Database name = Flipkart with 3 tables (Customers, Orders and Shippings)

Flipkart	Customers
	Orders
	Shippings

Customers Table

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

Orders Table

order_id	item	amount	customer_id
1	Keyboard	400	4
2	Mouse	300	4
3	Monitor	12000	3
4	Keyboard	400	1
5	Mousepad	250	2

Shippings Table

shipping_id	status	customer
1	Pending	2
2	Pending	4
3	Delivered	3
4	Pending	5
5	Delivered	1

Structured Query Language (SQL)

SQL Commands

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
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

SQL CREATE TABLE Statement

It is used to create a table in the data base with the field names along with their datatypes.

Syntax	CREATE TABLE table_name (column1 datatype, column2 datatype, column3 datatype, ...);
Example	CREATE TABLE Customers (customer_id [int], first_name [varchar(100)], last_name [varchar(100)], age [int], country [varchar(100)]);
	CREATE TABLE Orders (order_id [integer], item [varchar(100)], amount [integer], customer_id [integer]);
	CREATE TABLE Shippings (shipping_id [integer], status [integer], customer [integer]);

SQL INSERT Statement

It is used to insert data for different types or dame type or both into the table of a database.

Syntax	INSERT INTO table_name (column1, column2, column3, ...columnN) VALUES (value1, value2, value3, ...valueN); or INSERT INTO table_name VALUES (value1, value2, value3, ...valueN);
Example	INSERT INTO Customers VALUES (1, "a1", "a11", 10,"US"); INSERT INTO Customers VALUES (2, "a2", "a22", 20," US "); INSERT INTO Customers VALUES (3, "a3", "a33", 30,"UK"); INSERT INTO Customers VALUES (4, "a4", "a44", 40,"UK"); INSERT INTO Customers VALUES (5, "a5", "a55", 50,"India"); INSERT INTO Orders VALUES (1, "Ram", 100, 4); INSERT INTO Orders VALUES (2, "Processor", 300, 4); INSERT INTO Orders VALUES (3, "Monitor", 2000, 3); INSERT INTO Orders VALUES (4, "Ram", 4000, 1); INSERT INTO Orders VALUES (5, "Ram", 4000, 1); INSERT INTO Shippings VALUES (1, " Pending ", 1); INSERT INTO Shippings VALUES (2, " Pending ", 4); INSERT INTO Shippings VALUES (3, " Delivered ", 2); INSERT INTO Shippings VALUES (4, " Delivered ", 3); INSERT INTO Shippings VALUES (5, " Pending ", 5);

Customers Table with data after Insert Statement

customer_id	first_name	last_name	age	country
1	a1	a11	10	US
2	a2	a22	20	US
3	a3	a33	30	UK
4	a4	a44	40	UK
5	a5	a55	50	India

Orders Table with data after Insert Statement

order_id	item	amount	customer_id
1	Ram	100	4
2	Processor	300	4
3	Monitor	2000	3
4	Ram	4000	1
5	Ram	4000	1

Shippings Table with data after Insert Statement

shipping _id	status	customer
1	Pending	1
2	Pending	4
3	Pending	2
4	Pending	3
5	Pending	5

Delete all Rows

DELETE	deletes a table in the database.
Syntax	DELETE FROM Table_Name WHERE Condition;
TRUNCATE	command deletes the data inside a table
Example	DELETE FROM Customers WHERE customer_id=1; DELETE FROM Customers WHERE customer_id=2; DELETE FROM Customers WHERE customer_id=3; DELETE FROM Customers WHERE customer_id=4; DELETE FROM Customers WHERE customer_id=5; DELETE FROM Orders WHERE order_id=1; DELETE FROM Orders WHERE order_id=2; DELETE FROM Orders WHERE order_id=3; DELETE FROM Orders WHERE order_id=4; DELETE FROM Orders WHERE order_id=5; DELETE FROM Shippings WHERE shipping_id=1; DELETE FROM Shippings WHERE shipping _id=2; DELETE FROM Shippings WHERE shipping _id=3; DELETE FROM Shippings WHERE shipping _id=4; DELETE FROM Shippings WHERE shipping _id=5;

SQL SELECT Statement

Syntax	SELECT column1, column2, ... columnN FROM table_name; SELECT * FROM table_name;
Example	SELECT customer_id, first_name, country FROM Customers; SELECT * FROM Customers;

SELECT * FROM Customers;

customer_id	first_name	last_name	age	country
1	a1	a11	10	US
2	a2	a22	20	US
3	a3	a33	30	UK
4	a4	a44	40	UK
5	a5	a55	50	India

SELECT * Orders Customers;

order_id	item	amount	customer_id
1	Ram	100	4
2	Processor	300	4
3	Monitor	2000	3
4	Ram	4000	1
5	Ram	4000	1

SELECT * Shippings Customers;

shipping _id	status	customer
1	Pending	1
2	Pending	4
3	Pending	2
4	Pending	3
5	Pending	5

Structured Query Language (SQL)

DROP TABLE

Definition	Used to delete the table.
Syntax	DROP TABLE TABLE-NAME;
Example	DROP TABLE Customers; DROP TABLE Orders; DROP TABLE Shippings;
Output	No tables

Count Distinct

Definition	Used to count number of rows of a table.
Syntax	SELECT COUNT(DISTINCT COLUMN-NAME) FROM TABLE_NAME;
Example	SELECT COUNT(*) FROM Customers; SELECT COUNT(*) FROM Orders; SELECT COUNT(*) FROM Shippings; Output:5

SQL WHERE Clause

Definition	Used to get records which satisfy a particular condition on a table.
Syntax	SELECT * FROM TABLE_NAME WHERE COLUMN_NAME=CONDITION;

Examples

SELECT * FROM Customers WHERE customer_id=1;

customer_id	first_name	last_name	age	country
1	a1	a11	10	US

SELECT * FROM Orders WHERE order_id =1;

order_id	item	amount	customer_id
1	Ram	100	4

SELECT * FROM Shippings WHERE shipping_id =1;

shipping _id	status	customer
1	Pending	1

Conditions

Operator	**Description**
BETWEEN	Between a certain range
=	Equal
>	Greater than
>=	Greater than or equal
<	Less than
<=	Less than or equal
!=	Not equal.
LIKE	Search for a pattern
IN	To specify multiple possible values for a column

ORDER BY

Definition	It is used to sort the result set in ascending / descending order.
Syntax	SELECT column1, column2, ... column FROM TABLE_NAME ORDER BY column1, column2, ... ASC|DESC;
Example	SELECT * FROM Customers ORDER BY age;

Customers Table

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

SELECT * FROM Customers ORDER BY age;

customer_id	first_name	last_name	age	country
1	a1	a11	10	US
2	a2	a22	20	US
3	a3	a33	30	UK
4	a4	a44	40	UK
5	a5	a55	50	India

AND Operator

Definition	It is filter records based on more than one condition.
Syntax	SELECT column1, column2, ... column FROM TABLE_NAME WHERE condition1 AND condition2 AND condition3 ...; condition;
Example	SELECT * FROM Customers WHERE customer_id = '3' AND country = "UK";

Customers Table

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

SELECT * FROM Customers WHERE customer_id = '3' AND country = "UK";

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

AND Operator

Definition	It is filter records based on more than one condition.
Syntax	SELECT column1, column2, ... column FROM TABLE_NAME WHERE condition1 OR condition2 OR condition3 ...; conditionn
Example	SELECT * FROM Customers WHERE customer_id = '3' AND country = "UK";

Customers Table

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

SELECT * FROM Customers WHERE customer_id = '3' or country = "UK";

customer_id	first_name	last_name	age	country
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK

NOT Operator

Definition	It is filter records based on more than one condition.
Syntax	SELECT column1, column2, ... column FROM TABLE_NAME WHERE NOT condition;
Example	SELECT * FROM Customers WHERE NOT customer_id = 3;

Customers Table

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

SELECT * FROM Customers WHERE NOT customer_id = 3;

customer_id	first_name	last_name	age	country
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK

UPDATE **Statement**

Definition	It is used to change the existing records in a table.
Syntax	UPDATE TABLE_NAME SET column1 = value1, column2 = value2, ... columnn = valuen WHERE condition;
Example	UPDATE Customers SET first_name= 'power', age =40 WHERE customer_id = 1;

Customers Table

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

UPDATE Customers SET first_name= 'power', age =40 WHERE customer_id= 1;

customer_id	first_name	last_name	age	country
1	power	Doe	40	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

TOP Clause

Definition	It is used to select the top rows upto a particular value.
Syntax	UPDATE TABLE_NAME SET column1 = value1, column2 = value2, ... columnn = valuen WHERE condition;
Example	UPDATE Customers SET first_name= 'power', age =40 WHERE customer_id = 1;

MIN() and MAX() **Functions**

Definition	It is used to select the top rows upto a particular value.
Syntax	SELECT MIN(column_name) FROM TABLE_NAME WHERE condition; SELECT MAX(column_name) FROM TABLE_NAME WHERE condition;
Example	SELECT MIN(customer_id) FROM Customers; -------1 SELECT MAX(customer_id) FROM Customers; -------5

COUNT () Function

Definition	It is used to select the top rows upto a particular value.
Syntax	SELECT COUNT(column_name) FROM TABLE_NAME WHERE condition;
Example	SELECT COUNT(age) FROM Customers;-------5

Structured Query Language (SQL)

Customers Table

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

SUM () Function

Definition	It is used to compute the sum of a numeric column.
Syntax	SELECT SUM(column_name) FROM TABLE_NAME WHERE condition;
Example	SELECT SUM(age) FROM Customers;----------180

AVG() Function

Definition	It is used to compute the average of a numeric column.
Syntax	SELECT avg(column_name) FROM TABLE_NAME WHERE condition;
Example	SELECT avg(age) FROM Customers;----------36

LIKE Operator

Definition	It is used to search for a particular pattern in a column.
Syntax	SELECT column1, column2, ...Columnn FROM TABLE_NAME WHERE columnn LIKE pattern;
Example	SELECT * FROM Customers WHERE country LIKE 'UK%';

Customers Table

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

SELECT * FROM Customers WHERE country LIKE 'UK%';

customer_id	first_name	last_name	age	country
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK

IN Operator

Definition	It specify multiple values in a WHERE clause.
Syntax	SELECT column_name(s) FROM TABLE_NAME WHERE column_name IN (value1, value2, ... valuen);
Example	SELECT * FROM Customers WHERE customer_id IN ('1', '2', '4');

Structured Query Language (SQL)

Customers Table

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
4	John	Reinhardt	25	UK

BETWEEN Operator

Definition	It selects values within a given range for a particular column.
Syntax	SELECT column_name(s) FROM TABLE_NAME WHERE column_name BETWEEN value1 AND value2;
Example	SELECT * FROM Customers WHERE age BETWEEN 20 AND 40;

Customers Table

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
4	John	Reinhardt	25	UK

SELECT * FROM Customers WHERE age BETWEEN 20 AND 40;

customer_id	first_name	last_name	age	country
2	a2	a22	20	US
3	a3	a33	30	UK
4	a4	a44	40	UK

SQL Joins (Inner Join, Outer Join, RIGHT Join)

Inner Join	It is used to join two tables based on a relationship between the columns.
Syntax	SELECT column_name(s) FROM TABLE_NAME1 INNER JOIN TABLE_NAME2 ON TABLE_NAME1.column_name = TABLE_NAME2.column_name;
Example	SELECT first_name, last_name, order_id, item FROM Customers INNER JOIN Orders ON Customers.customer_id = Orders.order_id ;

Customers Table with data after Insert Statement

customer_id	first_name	last_name	age	country
1	a1	a11	10	US
2	a2	a22	20	US
3	a3	a33	30	UK
4	a4	a44	40	UK
5	a5	a55	50	India

Orders Table with data after Insert Statement

order_id	item	amount	customer_id
1	Ram	100	4
2	Processor	300	4
3	Monitor	2000	3
4	Ram	4000	1
5	Ram	4000	1

SELECT first_name, last_name, order_id, item FROM Customers
INNER JOIN Orders ON Customers.customer_id = Orders.order_id;

first_name	last_name	order_id	item
a1	a11	1	Ram
a2	a22	2	Processor
a3	a33	3	Monitor
a4	a44	4	Ram
a5	a55	5	Ram

Left Join

Left Join	It is used to join two tables based on a relationship between the columns.
Syntax	SELECT column_name(s) FROM TABLE_NAME1 LEFT JOIN TABLE_NAME2 ON TABLE_NAME1.column_name = TABLE_NAME2.column_name;
Example	SELECT first_name, age, item, amount FROM Customers LEFT JOIN Orders ON Customers.customer_id = Orders.order_id;

SELECT first_name, age, item, amount FROM CustomersLEFT JOIN Orders
ON Customers.customer_id = Orders.order_id;

first_name	age	item	amount
a1	10	Ram	100
a2	20	Processor	300
a3	30	Monitor	2000
a4	40	Ram	4000
a5	50	Ram	4000

Right Join

Right Join	It is used to join two tables based on a relationship between the columns.
Syntax	SELECT column_name(s) FROM TABLE_NAME1 LEFT JOIN TABLE_NAME2 ON TABLE_NAME1.column_name = TABLE_NAME2.column_name;
Example	SELECT first_name, age, item, amount FROM Customers LEFT JOIN Orders ON Customers.customer_id = Orders.order_id;

Note: Not supported by latest versions.

Union

Definition	It is used to join two tables based on a relationship between the columns.
syntax	SELECT column_name(s) FROM TABLE_NAME1 UNION ALL SELECT column_name(s) FROM TABLE_NAME2;
Example	SELECT customer_id,first_name FROM Customers UNION ALL SELECT order_id,item FROM Orders;

SELECT customer_id,first_name FROM Customers
UNION ALL SELECT order_id,item FROM Orders;

1	a1
2	a2
3	a3
4	a4
5	a5
1	Ram

GROUP BY

Definition	It is used to join two tables based on a relationship between the columns.
syntax	SELECT column1, column2, ... columnN FROM TABLE_NAME GROUP BY column_name;
Example	SELECT customer_id,first_name, age, country FROM Customers GROUP BY Country

Customers Table

customer_id	first_name	last_name	age	country
1	a1	a11	10	US
2	a2	a22	20	US
3	a3	a33	30	UK
4	a4	a44	40	UK
5	a5	a55	50	India

SELECT customer_id,first_name, age, country FROM Customers
GROUP BY Country;

customer_id	first_name	age	country
2	a2	20	US
5	a5	50	India
3	a3	30	UK

1	a1	10	US

HAVING clause

Definition	It is used to join two tables based on a relationship between the columns.
syntax	SELECT column1, column2, ... columnN FROM TABLE_NAME GROUP BY column_name HAVING Condition;
Example	SELECT customer_id,first_name, age, country FROM Customers GROUP BY Country HAVING COUNT(customer_id) < 2;

SELECT customer_id,first_name, age, country FROM Customers GROUP BY Country HAVING COUNT(customer_id) < 2;

customer_id	first_name	age	country
2	a2	20	US
5	a5	50	India
1	a1	10	US

EXISTS Statement

Definition	It is used to test for the presence of any record in a subquery.
syntax	SELECT column1, column2, ... columnN FROM TABLE_NAME GROUP BY column_name HAVING Condition;
Example	SELECT customer_id FROM Customers WHERE EXISTS (SELECT first_name FROM Customers);
Output	Customer_id 1 2 3 4 5

SELECT INTO Statement

Definition	It is used to copy data from one table to other new table.
syntax	SELECT * INTO new_table_name FROM Old_table_name ;
Example	SELECT * INTO Customers2 FROM Customers1;

Stored Procedure

Definition	It is a saved prepared SQL code that that can be reused over and over again.
syntax	CREATE PROCEDURE procedure_name AS sql_statements_Block GO;

Execute a Stored Procedure	EXEC procedure_name;
Example	CREATE PROCEDURE customerslist AS SELECT * FROM Customers WHERE country = @US GO; Example EXEC customerslist @ country = 'London';

SQL Single Line Comments

Definition	Used for documentation purpose.
syntax	-- Comment_Name:
Example	-- this is comment section:

SQL Database

Create DB	Used to create a database
syntax	CREATE DATABASE Database_Name;
Example	CREATE DATABASE wisdommaterials;

Drop

Drop DB	Used to Drop a database
syntax	Drop DATABASE Database_Name;
Example	Drop DATABASE wisdommaterials;

Backup Database

Definition	Used to backup a database
syntax	BACKUP DATABASE Database_Name TO DISK = 'filepath';
Example	BACKUP DATABASE mydb TO DISK = 'C:\backups\test_New_DB.bak';

Alter

Definition	Used to Add / Remove, Rename, ALTER/ DATATYPE the columns
syntax	ALTER TABLE table_name ADD column_name datatype;
Example	ALTER TABLE table_name DROP column_name datatype; ALTER TABLE table_name RENAME COLUMN old_name to new_name; ALTER TABLE table_name ALTER COLUMN column_name datatype;

SQL Constraints

These are used to specify conditions for the data in a column level or table level.

Structured Query Language (SQL)

SQL constraints

NOT NULL	column cant have a NULL value
UNIQUE	Ensures that all values in a column are different
PRIMARY KEY	NOT NULL plus UNIQUE.
FOREIGN KEY	Prevents actions that would destroy links between tables
CHECK	Used to check the values in a column satisfies a specific condition
DEFAULT	Sets a default value for a column if no value is given of it.
CREATE INDEX	Index created for the fast retrieval of data from the database.

VIEW Statement

Definition	It is table created when you run an SQL statement.
syntax	CREATE VIEW view_name AS SELECT column1, column2, ... columnn FROM table_name WHERE condition;
Example	CREATE VIEW [CustomersVIEW] AS SELECT customer_id, first_name FROM Customers WHERE customer_id = '1'; SELECT * FROM [Customers];

Customers Table

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

SQL Data Types

Data type	Description	Holds up to
CHAR(size)	A FIXED length string (can contain letters, numbers, and special characters).	length 0 to 255in characters
VARCHAR(size)	It is a string can contain letters, numbers, and special characters.	0 to 65535
BINARY(size)	Equal to CHAR().	binary byte strings.
VARBINARY(size)	It stores binary byte strings.	255 bytes
TINYBLOB	For BLOBs (Binary Large Objects).	
TINYTEXT	Holds a string with a maximum length	255 characters
TEXT(size)	Holds a string with a maximum length.	
BLOB(size)	For BLOBs (Binary Large Objects).	65,535 bytes of data
MEDIUMTEXT	Holds a string.	maximum length of 16,777,215 characters
MEDIUMBLOB	For BLOBs (Binary Large Objects).	16,777,215 bytes of data