**SQL Databases (CRUD Operations)**

CREATE DATABASE STUDENT;

SHOW DATABASES;

USE STUDENT;

SHOW TABLES;

CREATE TABLE STUDENT:

##RENAME DATABASE testDB TO tutorials\_DB;

SELECT DATABASE(); // to find the current working DB

CREATE TABLE CUSTOMERS (

ID INT PRIMARY KEY,

NAME VARCHAR(20),

AGE INT,

ADDRESS VARCHAR(50),

SALARY DECIMAL(10, 2)

);

INSERT INTO CUSTOMERS (ID, NAME, AGE, ADDRESS, SALARY)

VALUES

(1, 'Ramesh', 32, 'Ahmedabad', 2000.00),

(2, 'Khilan', 25, 'Delhi', 1500.00),

(3, 'kaushik', 23, 'Kota', 2000.00),

(4, 'Chaitali', 25, 'Mumbai', 6500.00),

(5, 'Hardik', 27, 'Bhopal', 8500.00),

(6, 'Komal', 22, 'Pune', 4500.00);

**SIMPLE CLONING:**

CREATE TABLE NEW\_CUSTOMERS SELECT \* FROM CUSTOMERS;

**SHALLOW CLONING:**

CREATE TABLE SHALL\_CUSTOMERS LIKE CUSTOMERS;

**DEEP CLONING: Shallow + Simple**

CREATE TABLE DEEP\_CUSTOMERS LIKE CUSTOMERS;

INSERT INTO DEEP\_CUSTOMERS SELECT \* FROM CUSTOMERS;

**TEMPORARY TABLE:**

CREATE TEMPORARY TABLE CUSTOMERS(

ID INT NOT NULL,

NAME VARCHAR (20) NOT NULL,

AGE INT NOT NULL,

ADDRESS CHAR (25) ,

SALARY DECIMAL (18, 2),

PRIMARY KEY (ID)

);

Just like normal tables you can insert data into a temporary table using the INSERT statement. Following query inserts 3 records into the above created temporary table −

INSERT INTO CUSTOMERS VALUES

(1, 'Ramesh', 32, 'Ahmedabad', 2000.00 ),

(2, 'Khilan', 25, 'Delhi', 1500.00 ),

(3, 'kaushik', 23, 'Kota', 2000.00 );

**DROP TEMPORARY TABLE CUSTOMERS;**

**ALTER TABLE:**

ALTER TABLE CUSTOMERS ADD Gender char(4);

UPDATE customers SET Gender = 'M' WHERE ID = 1;

UPDATE customers SET Gender = 'F' WHERE ID = 2;

UPDATE customers SET Gender = 'M' WHERE ID = 3;

UPDATE customers SET Gender = 'F' WHERE ID = 4;

UPDATE customers SET Gender = 'M' WHERE ID = 5;

UPDATE customers SET Gender = 'F' WHERE ID = 6;

UPDATE customers

SET Gender = CASE

WHEN ID = 1 THEN 'M'

WHEN ID = 2 THEN 'F'

WHEN ID = 3 THEN 'M'

WHEN ID = 4 THEN 'F'

WHEN ID = 5 THEN 'M'

WHEN ID = 6 THEN 'F'

END

WHERE ID IN (1, 2, 3, 4, 5, 6);

ALTER TABLE CUSTOMERS DROP COLUMN Gender;

ALTER TABLE CUSTOMERS RENAME COLUMN NAME to FULL\_NAME;

ALTER TABLE buyers

MODIFY COLUMN Address VARCHAR(25);

**ADD INDEX:**

1. ALTER TABLE customers

ADD INDEX full\_name\_index (FULL\_NAME);

DROP INDEX:

1. ALTER TABLE customers

DROP INDEX full\_name\_index;

show index from customers;

CREATE TABLE EMPLOYEES(

ID INT NOT NULL,

NAME VARCHAR (20) NOT NULL,

AGE INT NOT NULL,

ADDRESS CHAR (25),

SALARY DECIMAL (18, 2)

);

ALTER TABLE EMPLOYEES ADD CONSTRAINT MyPrimaryKey PRIMARY KEY(ID);

ALTER TABLE EMPLOYEES DROP PRIMARY KEY;

ALTER TABLE EMPLOYEES ADD CONSTRAINT CONST UNIQUE(NAME);

ALTER TABLE EMPLOYEES DROP CONSTRAINT CONST;

ALTER TABLE customers

MODIFY COLUMN ID INT NOT NULL;

ALTER TABLE customers

ADD PRIMARY KEY (ID);

ALTER TABLE customers

MODIFY COLUMN ID INT NOT NULL AUTO\_INCREMENT;

### Summary

* The AUTO\_INCREMENT attribute requires the column to be NOT NULL.
* If the column is not already NOT NULL, you need to modify it to be NOT NULL before adding the AUTO\_INCREMENT attribute.

**CREATE TABLE CUSTOMERS2** (

ID INT NOT NULL UNIQUE,

NAME VARCHAR (20) NOT NULL,

AGE INT NOT NULL,

ADDRESS CHAR (25),

SALARY DECIMAL (18, 2)

);

**CREATE TABLE CUSTOMERS11** (

ID INT NOT NULL UNIQUE,

NAME VARCHAR (20) DEFAULT 'Not Available',

AGE INT NOT NULL,

ADDRESS CHAR (25),

SALARY DECIMAL (18, 2)

);

CREATE TABLE **CUSTOMERS3**(

ID INT NOT NULL,

NAME VARCHAR (20) NOT NULL,

AGE INT NOT NULL,

ADDRESS CHAR (25),

SALARY DECIMAL (18, 2),

PRIMARY KEY (ID)

);

**SELECT**

SELECT ID, NAME, SALARY FROM CUSTOMERS;

SELECT CONCAT(NAME,' ',AGE) AS DETAILS, ADDRESS FROM CUSTOMERS ORDER BY NAME;

CREATE TABLE CUSTOMERS\_BACKUP AS SELECT \* FROM CUSTOMERS;

CREATE TABLE CUSTOMER\_DETAILS SELECT NAME, AGE, ADDRESS FROM CUSTOMERS;

SELECT \* INTO NameStartsWith\_K FROM CUSTOMERS WHERE NAME LIKE 'k%';

INSERT INTO BUYERS SELECT \* FROM CUSTOMERS;

INSERT INTO BUYERS SELECT \* FROM CUSTOMERS ORDER BY ID ASC LIMIT 3;

UPDATE CUSTOMERS SET ADDRESS = 'Pune' WHERE ID = 6;

UPDATE CUSTOMERS SET AGE = AGE+5, SALARY = SALARY+3000;

UPDATE CUSTOMERS SET ADDRESS = 'Goa', SALARY = 10000.00 WHERE NAME = 'Ramesh';

SELECT \* FROM CUSTOMERS

ORDER BY (CASE ADDRESS

WHEN 'DELHI' THEN 1

WHEN 'BHOPAL' THEN 2

WHEN 'KOTA' THEN 3

WHEN 'AHMEDABAD' THEN 4

WHEN 'Hyderabad' THEN 5

ELSE 100 END) ASC, ADDRESS DESC;

SELECT ID, NAME, SALARY FROM CUSTOMERS WHERE SALARY > 2000;

UPDATE CUSTOMERS set SALARY = SALARY+10000 where NAME = 'Ramesh';

SELECT \* from CUSTOMERS WHERE NAME IN ('Khilan', 'Hardik', 'Muffy');

SELECT \* from CUSTOMERS WHERE AGE NOT IN (25, 23, 22);

SELECT \* FROM CUSTOMERS WHERE NAME LIKE 'K\_\_\_%';

SELECT \* FROM CUSTOMERS

WHERE (AGE = 25 OR salary < 4500) AND (name = 'Komal' OR name = 'Kaushik');

SELECT TOP 4 \* FROM CUSTOMERS; // it will not work

SELECT \* FROM CUSTOMERS LIMIT 4;

SELECT \* FROM CUSTOMERS ORDER BY SALARY LIMIT 4;

SELECT \* FROM CUSTOMERS ORDER BY SALARY DESC LIMIT 4;

**SELECT \***

**FROM Customers**

**WHERE SALARY > (**

**SELECT 0.4 \* MAX(SALARY)**

**FROM Customers**

**)**

**ORDER BY SALARY DESC;**

**SELECT \***

**FROM CUSTOMERS**

**WHERE NAME LIKE 'K%'**

**LIMIT 2;**

DELETE FROM CUSTOMERS WHERE NAME LIKE 'K%' LIMIT 2;

SELECT \* FROM CUSTOMERS ORDER BY SALARY LIMIT 2;

**SELECT \* FROM CUSTOMERS**

**WHERE SALARY <= (**

**SELECT DISTINCT SALARY**

**FROM CUSTOMERS**

**ORDER BY SALARY**

**LIMIT 1 OFFSET 1**

**)ORDER BY SALARY;**

SELECT SALARY FROM CUSTOMERS ORDER BY SALARY;

SELECT DISTINCT SALARY FROM CUSTOMERS ORDER BY SALARY;

SELECT DISTINCT AGE, SALARY FROM CUSTOMERS ORDER BY AGE;

SELECT COUNT(DISTINCT AGE) as UniqueAge FROM CUSTOMERS;

SELECT DISTINCT AGE FROM CUSTOMERS;

UPDATE CUSTOMERS SET SALARY = NULL WHERE ID IN(6,4);

SELECT DISTINCT SALARY FROM CUSTOMERS ORDER BY SALARY;

SELECT \* FROM CUSTOMERS WHERE AGE = 25 ORDER BY NAME DESC;

SELECT SALARY FROM CUSTOMERS ORDER BY NAME LIMIT 4;

**GROUP BY**

SELECT AGE, COUNT(Name) FROM CUSTOMERS GROUP BY AGE;

SELECT AGE, MAX(salary) AS MAX\_SALARY FROM CUSTOMERS GROUP BY AGE;

SELECT ADDRESS, AVG(SALARY) as AVG\_SALARY FROM CUSTOMERS GROUP BY ADDRESS;

SELECT ADDRESS, AGE, SUM(SALARY) AS TOTAL\_SALARY FROM CUSTOMERS GROUP BY ADDRESS, AGE;

SELECT AGE, MIN(SALARY) AS MIN\_SALARY FROM CUSTOMERS GROUP BY AGE ORDER BY MIN\_SALARY DESC;

SELECT ADDRESS, AGE, MIN(SALARY) AS MIN\_SAL FROM CUSTOMERS GROUP BY ADDRESS, AGE HAVING AGE>24;

SELECT ADDRESS, AGE, MIN(SALARY) AS MIN\_SAL FROM CUSTOMERS GROUP BY ADDRESS, AGE HAVING AGE > 25;

SELECT ADDRESS, AGE, SUM(SALARY) AS TOTAL\_SALARY FROM CUSTOMERS GROUP BY ADDRESS, AGE HAVING TOTAL\_SALARY >=5000 ORDER BY TOTAL\_SALARY DESC;

SELECT AGE, COUNT(AGE) FROM CUSTOMERS GROUP BY AGE HAVING COUNT(age) > 2;

SELECT ADDRESS, AVG(SALARY) as AVG\_SALARY FROM CUSTOMERS GROUP BY ADDRESS HAVING AVG(SALARY) > 5240;

SELECT ADDRESS, MAX(SALARY) as MAX\_SALARY FROM CUSTOMERS GROUP BY ADDRESS HAVING MAX(SALARY) > 7000;

SELECT ID, NAME, SALARY FROM CUSTOMERS WHERE SALARY > 2000 AND age < 25;

SELECT \* FROM CUSTOMERS WHERE NAME LIKE 'k%' AND AGE >= 22 AND SALARY < 3742;

SELECT \* FROM CUSTOMERS WHERE NOT (SALARY > 4500 AND AGE < 26);

UPDATE CUSTOMERS SET salary = 55000 WHERE AGE > 27;

SELECT ID, NAME, SALARY FROM CUSTOMERS WHERE SALARY > 2000 OR age < 25;

SELECT \* FROM CUSTOMERS WHERE NAME LIKE '%l' OR SALARY > 10560 OR AGE < 25;

SELECT \* FROM CUSTOMERS WHERE (AGE = 25 OR salary < 4500) AND (name = 'Komal' OR name = 'Kaushik');

SELECT \* FROM CUSTOMERS WHERE SALARY LIKE '200%';

SELECT \* FROM CUSTOMERS WHERE NAME LIKE '%al%';

SELECT \* from customers WHERE NAME LIKE '\_\_m%';

**SELECT \***

**FROM customers**

**WHERE NAME LIKE 'k%'**

**AND (NAME LIKE 'kh%'**

**OR NAME LIKE 'ki%'**

**OR NAME LIKE 'ko%'**

**OR NAME LIKE 'km%'**

**OR NAME LIKE 'kl%'**

**OR NAME LIKE 'ka%'**

**OR NAME LIKE 'kn%');**

SELECT \* FROM customers WHERE NAME REGEXP '^[b-i]';

SELECT \* FROM customers WHERE NAME NOT REGEXP '^[b-k]';

SELECT \* FROM CUSTOMERS WHERE NAME LIKE 'C%i' OR NAME LIKE '%k';

select \* from customers WHERE NAME NOT LIKE 'K%';

select \* from employee WHERE BONUS\_PERCENT LIKE '%!%' escape '!';

select \* from employee WHERE BONUS\_PERCENT LIKE'2%!%%' escape '!';

select \* from CUSTOMERS WHERE NAME IN ('Khilan', 'Hardik', 'Muffy');

select \* from CUSTOMERS WHERE NAME = 'Khilan' OR NAME = 'Hardik' OR NAME = 'Muffy';

update customers set AGE = 30 where AGE IN (25, 27);

select \* from CUSTOMERS WHERE AGE NOT IN (25, 23, 22);

select \* from CUSTOMERS WHERE 2000 IN (SALARY);

select \* from CUSTOMERS WHERE NAME IN (SELECT NAME from CUSTOMERS WHERE SALARY > 2000);