SQL(10.30-12)

SQL LANGUAGES/STATEMENTS

1.DATA DEFINATION LANGUAGE(DDL)

CREATE
ALTER
TRUNCATE
DROP

2.DATA MANIPULATION LANGUAGE(DML)

INSERT
UPDATE
DELETE

3.TRANSACTION CONTROL LANGUAGE(TCL)

COMMIT
ROLLBACK
SAVEPOINT

4.DATA CONTROL LANGUAGE(DCL)

GRANT
REVOKE

5.DATA QUERY LANGUAGE(DQL)

SELECT
PROJECTION
SELECTION

JOINS
DATA DEFINATION LANGUAGE(DDL)

1.CREATE

IT IS USED TO CREATE DATABASE AND ITS OBJECTS SUCH AS TABLES,VIEW,PROCEDURE,TRIGGER AND FUNCTION.

TO CREATE DATABASE

SYNTAX:

CREATE DATABASE DATABASE_NAME;

CREATE DATABASE DEMO;

TO VIEW ALL THE DATABASES PRESENT IN MYSQL

SYNTAX:

SHOW DATABASES;

TO ACCESS PARTICULAR DATABASE IN MYSQL

SYNTAX:

USE DATABASE_NAME;

USE DEMO;

TO CLEAR THE SCREEN IN MYSQL SERVER

SYNTAX:

```
SYSTEM CLS
```

TO LOGGED OUT FROM MYSQL SERVER

SYNTAX:

```
EXIT
```

TO VIEW TABLES PRESENT IN PARTICULAR DATABASE

SYNTAX:

```
SHOW TABLES;
```

TO CREATE TABLE

SYNTAX:

```
CREATE TABLE TABLE_NAME
(
COLUMN_NAME1 DATATYPE NULL/NOT NULL,
COLUMN_NAME2 DATATYPE NULL/NOT NULL,
'
'
'
COLUMN_NAMEN DATATYPE NULL/NOT NULL,
CONSTRAINT CONSTRAINT_NAME PRIMARY KEY(COLUMN_NAME),
CONSTRAINT CONSTRAINT_NAME UNIQUE(COLUMN_NAME),
CONSTRAINT CONSTRAINT_NAME CHECK(CONDITION),
CONSTRAINT CONSTRAINT_NAME FOREIGN KEY(COLUMN_NAME)REFERENCES PARENT_TABLE_NAME(COLUMN_NAME)
);
```

```
CREATE TABLE STUDENT
(
SID INT NOT NULL,
SNAME VARCHAR(10) NOT NULL,
PHONE BIGINT NOT NULL,
CONSTRAINT P_SID PRIMARY KEY(SID),
CONSTRAINT U_PH UNIQUE(PHONE),
CONSTRAINT C_PH CHECK(LENGTH(PHONE)=10)
);

BRANCH
------
BID INT NOT NULL PRIMARY KEY---P_BID
BNAME VARCHAR(10) NOT NULL
AREA VARCHAR(10) NOT NULL
PINCODE INT NOT NULL UNIQUE,CHECK(LENGTH(PINCODE)=6) U_PIN,P_PIN

CREATE TABLE BRANCH
(
BID INT NOT NULL,
BNAME VARCHAR(10) NOT NULL,
AREA VARCHAR(10) NOT NULL,
PINCODE INT NOT NULL,
CONSTRAINT P_BID PRIMARY KEY(BID),
CONSTRAINT U_PIN UNIQUE(PINCODE),
CONSTRAINT P_PIN CHECK(LENGTH(PINCODE)=6)
);

PRODUCT
-------
PID INT NOT NULL PRIMARY KEY---P_PID
PNAME VARCHAR(10) NOT NULL
```

```
PRICE DECIMAL(10,2) NULL CHECK(PRICE>0)---C_PRICE
QUANTITY INT NOT NULL

CREATE TABLE PRODUCT
(
PID INT NOT NULL,
PNAME VARCHAR(10) NOT NULL,
PRICE DECIMAL(10,2) NULL,
QUANTITY INT NOT NULL,
CONSTRAINT P_PID PRIMARY KEY(PID),
CONSTRAINT C_PRICE CHECK(PRICE>0)
);

TO VIEW STRUCTURE OF THE TABLE
-------------------------------
SYNTAX:
------
DESC TABLE_NAME;

DESC BRANCH;

TO VIEW ALL THE CONSTRAINT_NAME OF CONSTRAINTS
----------------------------------------------
STEP_1:

USE INFORMATION_SCHEMA;

STEP_2;

SELECT * FROM TABLE_CONSTRAINTS;

ORDERS
------
```

```
ORDER_ID INT NOT NULL PRIMARY KEY---P_ORD
ORDER_DATE DATE NOT NULL
DELIVERY_DATE DATE NOT NULL
ORDER_LOC VARCHAR(10) NOT NULL

CREATE TABLE ORDERS
(
ORDER_ID INT NOT NULL,
ORDER_DATE DATE NOT NULL,
DELIVERY_DATE DATE NOT NULL,
ORDER_LOC VARCHAR(10) NOT NULL,
CONSTRAINT P_ORD PRIMARY KEY(ORDER_ID)
);

SYNTAX:
-------
CREATE TABLE TABLE_NAME
(
COLUMN_NAME1 DATATYPE NULL/NOT NULL CONSTRAINT,
COLUMN_NAME2 DATATYPE NULL/NOT NULL CONSTRAINT,
'
'
'
'
COLUMN_NAMEN DATATYPE NULL/NOT NULL CONSTRAINT,
CONSTRAINT CONSTRAINT_NAME FOREIGN KEY(COLUMN_NAME)REFERENCES PARENT_TABLE_NAME(COLUMN_NAME)
);

FACULTY
-------
FID INT NOT NULL PRIMARY KEY
FNAME VARCHAR(10) NOT NULL
SUBJECT VARCHAR(10) NOT NULL
```

```
PHONE BIGINT NOT NULL CHECK(LENGTH(PHONE)=10),UNIQUE

CREATE TABLE FACULTY
(
FID INT NOT NULL PRIMARY KEY,
FNAME VARCHAR(10) NOT NULL,
SUBJECT VARCHAR(10) NOT NULL,
PHONE BIGINT NOT NULL UNIQUE,CHECK(LENGTH(PHONE)=10)
);

CUSTOMER
-------
CID INT NOT NULL PRIMARY KEY AUTO_INCREMENT
CNAME VARCHAR(10) NN
PHONE BIGINT NN UNIQUE,CHECK(LENGTH(PHONE)=10)
ADDRESS VARCHAR(10) NN
GENDER ENUM('MALE','FEMALE') NN
ORDER_ID INT NULL
BALANCE DECIMAL(10,2) NN CHECK(BALANCE>0) DEFAULT'0'
SID INT NN FOREIGN KEY ---F_SID

CREATE TABLE CUSTOMER
(
CID INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
CNAME VARCHAR(10) NOT NULL,
PHONE BIGINT NOT NULL UNIQUE,CHECK(LENGTH(PHONE)=10),
ADDRESS VARCHAR(10) NOT NULL,
GENDER ENUM('MALE','FEMALE') NOT NULL,
ORDER_ID INT NULL,
BALANCE DECIMAL(10,2) NOT NULL CHECK(BALANCE>0) DEFAULT'0',
SID INT NOT NULL,
CONSTRAINT F_SID FOREIGN KEY(SID)REFERENCES STUDENT(SID)
);
```

BANK

ACCOUNT_NUMBER INT NN PRIMARY KEY
IFSCE_CODE VARCHAR(20) NN UNIQUE,CHECK(LENGTH(IFSC_CODE)=11)
ACCOUNT_HOLDER_NAME VARCHAR(10) NN
BRANCH_NAME VARCHAR(10) NN
BID INT NN FOREIGN KEY---F_BID

```
CREATE TABLE BANK
(
ACCOUNT_NUMBER INT PRIMARY KEY,
IFSC_CODE VARCHAR(20) NOT NULL UNIQUE,CHECK(LENGTH(IFSC_CODE)=11),
ACCOUNT_HOLDER_NAME VARCHAR(10) NOT NULL,
BRANCH_NAME VARCHAR(10) NOT NULL,
BID INT NOT NULL,
CONSTRAINT F_BID FOREIGN KEY(BID)REFERENCES BRANCH(BID)
);
```

ALTER

IT IS USED TO MODIFY STRUCTURE OF THE TABLE.

TO ADD A NEW_COLUMN

SYNTAX:

```
ALTER TABLE TABLE_NAME
ADD COLUMN_NAME DATATYPE NULL/NOT NULL;
```

EMAIL VARCHAR(10) NOT NULL----STUDENT

```
ALTER TABLE STUDENT
ADD EMAIL VARCHAR(10) NOT NULL;
```

TO ADD COLUMN AFTER PARTICULAR COLUMN

SYNTAX:

```
ALTER TABLE TABLE_NAME
ADD COLUMN_NAME DATATYPE NULL/NOT NULL AFTER EXISTING_COLUMN_NAME;
```

BALANCE DECIMAL(3,1) NOT NULL---SID---STUDENT

```
ALTER TABLE STUDENT
ADD BALANCE DECIMAL(3,1) NOT NULL AFTER SID;
```

TO DROP COLUMN

SYNTAX:

```
ALTER TABLE TABLE_NAME
DROP COLUMN_NAME;
```

```
ALTER TABLE CUSTOMER
DROP ADDRESS;
```

TO CHANGE DATATYPE

SYNTAX:

```
ALTER TABLE TABLE_NAME
MODIFY EXISTING_COLUMN_NAME NEW_DATATYPE NULL/NOT NULL;
```

```
ALTER TABLE BANK
MODIFY ACCOUNT_HOLDER_NAME CHAR(10) NOT NULL;
```

TO CHANGE NULL/NOT NULL

```
---------------------
SYNTAX:

ALTER TABLE TABLE_NAME
MODIFY EXISTING_COLUMN_NAME EXISTING_DATATYPE NULL/NOT NULL;

ALTER TABLE CUSTOMER
MODIFY ORDER_ID INT NOT NULL;

TO CHANGE TABLE_NAME
------------------
SYNTAX:

ALTER TABLE TABLE_NAME
RENAME NEW_TABLE_NAME;

FACULTY_DATA

ALTER TABLE FACULTY
RENAME FACULTY_DATA;

TO CHANGE COLUMN_NAME
-------------------
SYNTAX:

ALTER TABLE TABLE_NAME
CHANGE EXISTING_COLUMN_NAME NEW_COLUMN_NAME EXISTING_DATATYPE NULL/NOT NULL;

ALTER TABLE BANK
CHANGE BRANCH_NAME BNAME VARCHAR(10) NOT NULL;

CONTACT_NUMBER
```

```
ALTER TABLE STUDENT
CHANGE PHONE CONTACT_NUMBER BIGINT NOT NULL;
```

TO ADD PRIMARY KEY CONSTRAINT

SYNTAX:

```
ALTER TABLE TABLE_NAME
ADD CONSTRAINT CONSTRAINT_NAME PRIMARY KEY(COLUMN_NAME);
```

TO ADD UNIQUE CONSTRAINT

SYNTAX:

```
ALTER TABLE TABLE_NAME
ADD CONSTRAINT CONSTRAINT_NAME UNIQUE(COLUMN_NAME);
```

TO ADD CHECK CONSTRAINT

SYNTAX:

```
ALTER TABLE TABLE_NAME
ADD CONSTRAINT CONSTRAINT_NAME CHECK(CONDITION);

ALTER TABLE BRANCH
ADD CONSTRAINT UNIQUE(AREA);
```

TO ADD FOREIGN KEY

SYNTAX:

```
ALTER TABLE TABLE_NAME
ADD CONSTRAINT CONSTRAINT_NAME FOREIGN KEY(COLUMN_NAME)REFERENCES PARENT_TABLE_NAME(COLUMN_NAME);
```

```
ALTER TABLE CUSTOMER
ADD CONSTRAINT FOREIGN KEY(ORDER_ID)REFERENCES ORDERS(ORDER_ID);

TO ADD DEFAULT CONSTRAINT FOR EXISTING COLUMN
---------------------------------------------
SYNTAX:

ALTER TABLE TABLE_NAME
MODIFY EXISTING_COLUMN_NAME EXISTING_DATATYPE NULL/NOT NULL DEFAULT'DEFAULT_VALUE';

ALTER TABLE PRODUCT
MODIFY QUANTITY INT NOT NULL DEFAULT'0';

TO ADD DEFAULT CONSTRAINT FOR NEW COLUMN
----------------------------------------
SYNTAX:

ALTER TABLE TABLE_NAME
ADD COLUMN_NAME DATATYPE NULL/NOT NULL DEFAULT'DEFAULT_VALUE';

BALANCE DECIMAL(10) NOT NULL---1000-FACULTY_DATA

ALTER TABLE FACULTY_DATA
ADD BALANCE DECIMAL(10) NOT NULL DEFAULT'1000';

TO ADD AUTO_INCREMENT FOR EXISTING COLUMN
-----------------------------------------
SYNTAX:

ALTER TABLE TABLE_NAME
MODIFY EXISTING_COLUMN_NAME EXISTING_DATATYPE NULL/NOT NULL AUTO_INCREMENT;
```

```
ALTER TABLE PRODUCT
MODIFY PID INT NOT NULL AUTO_INCREMENT;
```

TO ADD AUTO_INCREMENT FOR NEW COLUMN

SYNTAX:

```
ALTER TABLE TABLE_NAME
ADD COLUMN_NAME DATATYPE NULL/NOT NULL PRIMARY KEY AUTO_INCREMENT;
```

TO DROP PRIMARY KEY

SYNTAX:

```
ALTER TABLE TABLE_NAME
DROP PRIMARY KEY;
```

```
ALTER TABLE FACULTY_DATA
DROP PRIMARY KEY;
```

TO DROP DEFAULT

SYNTAX:

```
ALTER TABLE TABLE_NAME
ALTER COLUMN COLUMN_NAME
DROP DEFAULT;
```

```
ALTER TABLE FACULTY_dATA
ALTER COLUMN BALANCE
DROP DEFAULT;
```

TO DROP AUTO_INCREMENT

SYNTAX:

```
ALTER TABLE TABLE_NAME
MODIFY EXISTING_COLUMN_NAME EXISTING_DATATYPE NULL/NOT NULL;
```

```
ALTER TABLE CUSTOMER
MODIFY CID INT NOT NULL;
```

TO DROP OTHER CONSTRAINTS(UNIQUE,CHECK,FOREIGN KEY)

SYNTAX:

```
ALTER TABLE TABLE_NAME
DROP CONSTRAINT CONSTRAINT_NAME;
```

DATA MANIPULATION LANGUAGE(DML)

INSERT
UPDATE
DELETE

1.INSERT

TO INSERT RECORDS INTO EXISTING TABLE WE USE INSERT COMMAND.

SYNTAX_1(WHEN WE KNOW COLUMNS ALONG WITH THE ORDER OF THE COLUMNS WE USE SYNTAX_1)

```
INSERT INTO TABLE_NAME VALUES(V1,V2,....VN);
```

```
INSERT INTO PRODUCT VALUES(1,'LAPTOP',40000,1);
```

TO RETRIVE DATA FROM PRODUCT TABLE

```
SELECT * FROM PRODUCT;

INSERT INTO PRODUCT VALUES(2,'LIPSTICK',500,10);

INSERT INTO PRODUCT VALUES(3,'OLD MONK',200,5),(4,'BMW',100000,1);

INSERT INTO PRODUCT VALUES(5,'SHIRT',50000000,2);
```

SYNTAX_2(WHEN WE KNOW ONLY COLUMNS WE USE SYNTAX_2)

```
INSERT INTO TABLE_NAME(COL1,COL2,....COLN) VALUES(V1,V2,......VN);

INSERT INTO PRODUCT(PNAME,PRICE,PID,QUANTITY)VALUES('PAPER',10,6,1);

INSERT INTO PRODUCT(PNAME,PRICE,PID,QUANTITY)VALUES('KITKAT',120,7,3),('SILK',195,8,1);

INSERT INTO PRODUCT(PNAME,PRICE)VALUES('KINDER JOY',40);
```

**SYNTAX_3

```
INSERT INTO TABLE_NAME(SELECT STATEMENT);
```

UPDATE

TO MODIFY EXISTINGS RECORDS IN A TABLE WE USE UPDATE COMMAND

SYNTAX:

```
UPDATE TABLE_NAME
SET COL1=V1,COL2=V2,,,,,COLN=VN
```

```
[WHERE CONDITION];
```

```
UPDATE PRODUCT
SET PNAME='BIRYANI';
```

```
UPDATE PRODUCT
SET PNAME='PULAV'
WHERE PID=1;
```

```
UPDATE PRODUCT
SET PNAME='GHEE RICE',PRICE=80
WHERE PID=4;
```

DELETE

TO DELETE RECORDS FROM A TABLE WE USE DELETE COMMAND.

SYNTAX:

```
DELETE
FROM TABLE_NAME
[WHERE CONDITION];
```

```
DELETE
FROM PRODUCT
WHERE PID=9;
```

TRUNCATE

TO DELETE ALL THE RECORDS FROM TABLE WITHOUT AFFECTING TABLE STRUCTURE.

SYNTAX:

```
TRUNCATE TABLE TABLE_NAME;

TRUNCATE TABLE PRODUCT;

INSERT INTO PRODUCT(PNAME,PRICE)VALUES('KUSHKA',80);
```

DROP

TO DROP DATABASE AND ITS OBJECTS SUCH AS TABLE,VIEW,FUNCTION,PROCEDURE AND TRIGGER WE USE DROP COMMAND.

TO DROP TABLE

SYNTAX:

```
DROP TABLE TABLE_NAME;

DROP TABLE PRODUCT;
```

TO DROP DATABASE

```
DROP DATABASE DATABASE_NAME;

DROP DATABASE DEMO;

CREATE DATABASE AMAZON;

INSERT INTO EMP VALUES(1701,'RAHUL','MUKARJEE','1991-02-19','M','MANAGER',1702,' 2017-04-17',100000,NULL,111,NULL),
```

Order of Execution

> FROM
> SELECT

FROM Clause

> FROM Clause always execute first in query. It takes Table_name as argument.
> It goes to database, search for the table and put the table under execution.

SELECT Clause

> SELECT Clause executes after FROM Clause.
> It takes */Column_name/expressions as arguments.
> SELECT Clause goes to the table which is under execution, gets the data and display the output.

1.WQTD EMPLOYEE FIRST NAME AND SALARY FROM EMP TABLE???

```
SELECT FNAME,SAL
FROM EMP;
```

2.WQTD EMPLOYEE FIRST NAME,LAST NAME, DATE OF BIRTH AND DATE OF JOINING FRPM EMP TABLE??

```
SELECT FNAME,LNAME,DOB,DOJ
FROM EMP;
```

3.WQTD DETAILS OF EMPS FROM EMP TABLE??

```
SELECT *
FROM EMP;
```

4.WQTD EMP FNAME,LNAME,DEPT NO AND ANNUAL SALARY FROM EMP TABLE???

```
SELECT FNAME,LNAME,DNO,SAL*12
FROM EMP;
```

5.WQTD DETAILS OF EMPS ALONG WITH ANNUAL SALARY FROM EMP TABLE???

```
SELECT *,SAL*12
FROM EMP;
```

6.WQTD FNAME,LNAME,DOB AND SALARY WITH 20000 BONUS???

```
SELECT FNAME,LNAME,DOB,SAL+20000
FROM EMP;
```

7.WQTD FNAME,GENDER AND SALARY WITH 10% HIKE???

```
SELECT FNAME,GENDER,SAL+(SAL*10/100)
FROM EMP;
```

```
HIKE:
-----
COLUMN_NAME+(COLUMN_NAME * %HIKE/100)
```

```
DEDUCTION
---------
COLUMN_NAME-(COLUMN_NAME * %DEDUCTION/100)
```

8.WQTD FNAME,DOB,SAL AND ANNUAL SALARY WITH 8% HIKE???

```
SELECT FNAME,DOB,SAL,(SAL*12)+((SAL*12)*8/100)
FROM EMP;
```

9.WQTD FNAME,LNAME,JOB,DOB,SAL WITH 14 % DEDUCTION AND ANNUAL COMMISSION???

```
SELECT FNAME,LNAME,JOB,DOB,SAL-(SAL* 14/100),COMM*12
FROM EMP;
```

10.WQTD FNAME,DNO,MGR AND ANNUAL COMMISSION WITH 16% HIKE AND ANNUAL SAL WITH 4% DEDUCTION???

```
SELECT FNAME,DNO,MGR,(COMM*12)+((COMM*12)*16/100),(SAL*12)-((SAL*12)*4/100)
FROM EMP;
```

11.WQTD FNAME,JOB,SAL WITH 5% HIKE,COMM WITH 10% DEDUCTION,ANNUAL SALARY WITH 15% HIKE AND ANNUAL COMM WITH 20% DEDUCTION???

```
SELECT FNAME,JOB,SAL+(SAL*5/100),COMM-(COMM *10/100),(SAL*12)+((SAL*12)*15/100),(COMM*12)-((COMM*12)*20/100)
FROM EMP;
```

ALIASING

TO PROVIDE ALTERNATIVE NAME FOR COLUMNS IN RESULTANT TABLE WE USE ALIASING.

RULES

>WITH OR WITHOUT AS KEYWORD WE CAN WRITE ALIAS NAME.
>WE CAN USE MULTIPLE WORDS AS ALIAS NAME BY USING QUOTES OR BY CONNECTING IT WITH UNDER_SCORE

11.WQTD FNAME,JOB,SAL WITH 5% HIKE,COMM WITH 10% DEDUCTION,ANNUAL SALARY WITH 15% HIKE AND ANNUAL COMM WITH 20% DEDUCTION???

```
SELECT FNAME,JOB,SAL+(SAL*5/100)AS 'SAL WITH 5% HIKE' ,COMM-(COMM *10/100) 'COMM WITH 10% DEDUCTION',
(SAL*12)+((SAL*12)*15/100)AS 'ANNUAL SALARY WITH 15% HIKE',(COMM*12)-((COMM*12)*20/100)'ANNUAL COMM WITH 20% DEDUCTION'

FROM EMP;
```

12.WQTD FIRST NAME,LAST NAME AND ANNUAL SALARY FROM EMP TABLE???

```
SELECT FNAME AS 'FIRST NAME',LNAME 'LAST NAME',SAL*12 'ANNUAL SALARY'
FROM EMP;
```

13.WQTD DIFFERENT JOB ROLE PRESENT IN EMP TABLE???

```
SELECT DISTINCT JOB
FROM EMP;
```

DISTINCT

TO AVOID DUPLICATE VALUES FROM THE RESULTANT TABLE WE USE DISTINCT KEYWORD.

RULES

>EITHER * OR DISTINCT MUST BE THE FIRST ARGUMENT IN SELECT CLAUSE.

>WE CAN USE MULTIPLE COLUMNS ALONG WITH DISTINCT KEYWORD,IT WILL AVOID COMBINATION OF DUPLICATE VALUES.

14.WQTD UNIQUE DEPARTMENT NUMBERS FROM EMP TABLE??

```
SELECT DISTINCT DNO
FROM EMP;
```

15.WQTD UNIQUE COMBINATION OF SALARY AND DNO FROM EMP TABLE???

```
SELECT DISTINCT SAL,DNO
FROM EMP;
```

16.WQTD DETAILS OF EMPS WHOSE FNAME IS RASHMI???

```
SELECT *
FROM EMP
WHERE FNAME='RASHMI';
```

ORDER OF EXECUTION

1.FROM
2.WHERE
3.SELECT

17.WQTD FNAME AS FIRST NAME ,LNAME AS LAST NAME AND JOB FROM EMP TABLE IF THE EMP IS WORKING AS SALESMAN??

```
SELECT FNAME 'FIRST NAME',LNAME 'LAST NAME',JOB
FROM EMP
WHERE JOB='SALESMAN';
```

18.WQTD DETAILS OF EMPS IF THE EMPS ARE GETTING SALARY MORE THAN 45000???

```
SELECT *
FROM EMP
WHERE SAL>45000;
```

19.WQTD FNAME AS FIRST NAME,LNAME LAST NAME ,SAL AS SALARY AND DOB AS DATE OF BIRTH FROM EMP TABLE IF THE EMP BORN AFTER THE YEAR 1993??

```
SELECT FNAME 'FIRST NAME',LNAME 'LAST NAME',SAL SALARY,DOB 'DATE OF BIRTH'
FROM EMP
WHERE DOB>'1993-12-31';
OR
WHERE DOB>='1994-01-01';
```

20.WQTD FNAME,JOB,DOJ IF THE EMP HIRED BEFORE THE YEAR 2019???

```
SELECT FNAME,JOB,DOJ
FROM EMP
WHERE DOJ<'2019-01-01';
OR
WHERE DOJ<='2018-12-31'
```

21.WQTD DETAILS OF EMPS ALONG WITH ANNUAL SALARY IF THE EMPS ARE GETTING ANNUAL SALARY MORE THAN 500000???

```
SELECT *,SAL*12 'ANNUAL SALARY'
```

```
FROM EMP
WHERE (SAL*12)>500000;
```

22.WQTD DETAILS OF EMPS IF THE EMPS HIRED AFTER THE MONTH MARCH IN 2020???

```
SELECT *
FROM EMP
WHERE DOJ>'2020-03-31';
```

OPERATORS

1.ARITHMATIC OPERATORS(+,-,*,/,%)
2.RELATIONAL OPERATORS(<,>,<=,>=,=,!= OR <>)
3.LOGICAL OPERATORS(AND,OR,NOT)
4.SPECIAL OPERATORS(IN,NOT IN,IS,BETWEEN,NOT BETWEEN,LIKE,NOT LIKE)
5.SUBQUERY OPERATORS(ALL,ANY)

23.WQTD DETAILS OF EMPS IF THE EMPS ARE WORKING AS SALESMAN OR MANAGER???

```
SELECT *
FROM EMP
WHERE JOB='SALESMAN' OR JOB='MANAGER';
```

24.WQTD FNAME,LNAME,SAL,DOB FROM EMP TABLES IF THE EMPS BORN DURING THE YEAR 1995???

```
SELECT FNAME,LNAME,SAL,DOB
FROM EMP
WHERE DOB>='1995-01-01' AND DOB<='1995-12-31';
```

25.WQTD FNAME,JOB AND SALARY IF THE EMPS ARE WORKING AS DEVELOPER AND GETTING SALARY MORE THAN 50000???

```
SELECT FNAME,JOB,SAL
FROM EMP
```

```
WHERE JOB='DEVELOPER' AND SAL>50000;
```

26.WQTD DETAILS OF EMPS IF THE EMPS ARE WORKING AS SALESMAN OR DEVELOPER IN DEPT NUMBER 111??

```
SELECT *
FROM EMP
WHERE (JOB='SALESMAN' OR JOB='DEVELOPER') AND DNO=111;
```

27.WQTD DETAILS OF THE EMPS IF THE EMPS ARE WORKING AS SALESMAN,MANAGER,DEVELOPER OR DISPATCHER AND GETTING SALARY MORE THAN 45000???

```
SELECT *
FROM EMP
WHERE (JOB='SALESMAN' OR JOB='MANAGER' OR JOB='DEVELOPER' OR JOB='DISPATCHER') AND SAL>45000;
```

```
SELECT *
FROM EMP
WHERE JOB IN('SALESMAN','MANAGER','DEVELOPER','DISPATCHER') AND SAL>45000;
```

IN
--

IT IS A MULTI VALUE OPERATOR WHICH TAKES MULTIPLE VALUES AT THE RHS AND SINGLE VALUE AT THE LHS.

SYNTAX:

```
LHS RHS
COLUMN_NAME/EXPRESSION IN(V1,V2,....VN);
```

28.WQTD DETAILS OF THE EMPS IF THE EMPS ARE WORKING IN 110,111 OR 112 AND GETTING SALARY MORE THAN 30000 AND WORKING AS MANAGER OR SALESMAN???

```
SELECT *
```

```
FROM EMP
WHERE DNO IN(110,111,112) AND SAL>30000 AND JOB IN('MANAGER','SALESMAN');
```

29.WQTD FNAME,LNAME,JOB,DOJ IF THE EMP HIRED AFTER THE YEAR 2017 AND WORKING AS SALESMAN,MANAGER OR DEVELOPER???

```
SELECT FNAME,LNAME,JOB,DOJ
FROM EMP
WHERE DOJ>'2017-12-31' AND JOB IN('SALESMAN','MANAGER','DEVELOPER');
```

30.WQTD DETAILS OF THE EMPS ALONG WITH ANNUAL SALARY IF THE EMPS ARE NOT WORKING AS CEO AND GETTING ANNUAL SALARY MORE THAN 500000(WITHOUT USING SPECIAL OPERATOR)

```
SELECT *,SAL*12 'ANNUAL SALARY'
FROM EMP
WHERE JOB!='CEO' AND (SAL*12)>500000;
```

31.WQTD FNAME,LNAME,SAL,JOB IF THE EMPS ARE NOT WORKING AS SALESMAN,MANAGER,DEVELOPER (WITHOUT USING SPECIAL OPERATOR)

```
SELECT FNAME,LNAME,SAL,JOB
FROM EMP
WHERE JOB!='SALESMAN' AND JOB!='MANAGER' AND JOB!='DEVELOPER';
```

or

```
SELECT FNAME,LNAME,SAL,JOB
FROM EMP
WHERE JOB NOT IN('SALESMAN','MANAGER','DEVELOPER');
```

32.WQTD FNAME,DNO,SAL IF THE EMPS ARE NOT WORKING IN DNO 110,112 AND GETTING SALARY LESS THAN 100000??

```
SELECT FNAME,DNO,SAL
FROM EMP
```

```
WHERE DNO NOT IN(110,112) AND SAL<100000;
```

33.WQTD DETAILS OF EMPS WHO WERE BORN IN THE YEAR 1998 AND NOT WORKING AS MANAGER,SALESMAN AND HIRED DURING THE YEAR 2020 BUT NOT AS FEMALE??

```
SELECT *
FROM EMP
WHERE DOB>='1998-01-01' AND DOB<='1998-12-31' AND JOB NOT IN('MANAGER','SALESMAN') AND DOJ>='2020-01-01' AND
DOJ<='2020-12-31' AND GENDER!='F';
```

34.WQTD DETAILS OF EMPS IF THE EMPS ARE GETTING SALARY MORE THAN OR EQUAL TO 35000 AND LESS THAN OR EQUAL TO 50000???

```
SELECT *
FROM EMP
WHERE SAL>=35000 AND SAL<=50000;
```

BETWEEN

WHENEVER WE NEED TO INCLUDE SOME RANGE OF VALUES WE USE BETWEEN OPERATOR...

SYNTAX:

```
>= <=
COLUMN_NAME/EXPRESSION BETWEEN LOWER_RANGE_VALUE AND HIGHER_RANGE_VALUE;
```

```
SELECT *
FROM EMP
WHERE SAL BETWEEN 35000 AND 50000;
```

35.WQTD DETAILS OF EMPS THOSE WHO WERE BORN IN THE YEAR 1995????

```
SELECT *
FROM EMP
```

```
WHERE DOB BETWEEN '1995-01-01' AND '1995-12-31';
```

36.WQTD DETAILS OF EMPS WHO ARE GETTING SALARY MORE THAN 35000 AND LESS THAN 100000 AND WORKING AS SALESMAN,DEVELOPER OR MANAGER???

```
SELECT *
FROM EMP
WHERE SAL BETWEEN 35000+0.01 AND 100000-0.01 AND JOB IN('SALESMAN','DEVELOPER','MANAGER');
```

SAL DECIMAL(10,2)--MIN + VALUE 0.01

OR

```
SELECT *
FROM EMP
WHERE SAL>35000 AND SAL<100000 AND JOB IN('SALESMAN','DEVELOPER','MANAGER');
```

37.WQTD DETAILS OF EMPS WHO ARE NOT GETTING SALARY IN THE RANGE OF 35000 AND 45000??

NOT BETWEEN

WHENEVER WE NEED TO EXCLUDE SOME RANGE OF VALUES WE USE NOT BETWEEN.

SYNTAX:

```
< >
COLUMN_NAME/EXPRESSION NOT BETWEEN LOWER_RANGE_VALUE AND HIGHER_RANGE_VALUE;
```

37.WQTD DETAILS OF EMPS WHO ARE NOT GETTING SALARY IN THE RANGE OF 35000 AND 45000??

```
SELECT *
```

```
FROM EMP
WHERE SAL NOT BETWEEN 35000 AND 45000;
```

38.WQTD DETAILS OF EMPS THOSE WHO WERE NOT JONIED IN THE YEAR 2020???

```
SELECT *
FROM EMP
WHERE DOJ NOT BETWEEN '2020-01-01' AND '2020-12-31';
```

39.WQTD DETAILS OF THE EMPS WHO ARE GETTING SOME COMMISSION???

```
SELECT *
FROM EMP
WHERE COMM IS NOT NULL;
```

IS
--
TO CHECK WHETHER THE COLUMN IS NULL NOT NULL WE USE IS OPERATOR.

SYNTAX:

```
COLUMN_NAME/EXPRESSION IS NULL/NOT NULL;
```

40.WQTD DETAILS OF EMPS IF THE EMPS ARE ACTING AS CUSTOMER FOR THEIR COMPANY???

```
SELECT *
FROM EMP
WHERE CID IS NOT NULL;
```

41.WQTD DETAILS OF THE EMPS IF THEY HAVE REPORTING MANAGER???

```
SELECT *
FROM EMP
```

```
WHERE MGR IS NOT NULL;
```

42.WQTD DETAILS OF THE EMPS WHO ARE WORKING AS SALESMAN,DEVELOPER OR DISPATCHER BUT NOT IN THE DNO 110,112 AND GETTING SALARY IN THE RANGE OF 30000 AND 50000 BUT WERE NOT BORN IN THE YEAR 1995 AND THEY HAVE REPORTING MANAGER??

```
SELECT *
FROM EMP
WHERE JOB IN('SALESMAN','DEVELOPER','DISPATCHER') AND DNO NOT IN(110,112) AND SAL BETWEEN 30000 AND 50000 AND DOB NOT
BETWEEN '1995-01-01' AND '1995-12-31' AND MGR IS NOT NULL;
```

43.WQTD DETAILS OF EMPS IF THE EMPS ARE GETTING SALARY IN THE RANGE OF 40000 TO 100000 AND HIRED IN THE YEAR 2019 AND IN THE DATE OF 26????

```
SELECT *
FROM EMP
WHERE SAL BETWEEN 40000 AND 100000 AND DOJ
IN('2019-01-26','2019-02-26','2019-03-26','2019-04-26','2019-05-26','2019-06-26','2019-07-26','2019-08-26','2019-09-26','2019-10-26','2019-11-26','2019-12-26');
```

44.WQTD DETAILS OF EMPS IF THE FNAME IS STARTING WITH K???

```
select *
FROM EMP
WHERE FNAME LIKE 'K%';
```

45.WQTD DETAILS OF EMPS IF THE EMP LNAME IS ENDING WITH I??

```
SELECT *
FROM EMP
WHERE LNAME LIKE '%I';
```

46.WQTD FNAME,LNAME AND JOB IF THE FNAME ATLEAST 2A AND GETTING SALARY MORETHAN 25000??

```
SELECT FNAME,LNAME,JOB,SAL
FROM EMP
WHERE FNAME LIKE '%A%A%' AND SAL>25000;
```

47.WQTD FNAME,LNAME AND JOB IF THE JOB CONTAINS STRING 'MAN' IN IT??

```
SELECT FNAME,LNAME,JOB
FROM EMP
WHERE JOB LIKE '%MAN%';
```

48.WQTD DETAILS OF EMPS WHOSE FNAME STARTS WITH S OR A???

```
SELECT *
FROM EMP
WHERE FNAME LIKE 'S%' OR FNAME LIKE 'A%';
```

49.WQTD DETAILS OF EMPS IF THE EMP HIRED IN THE YEAR 2019????

```
SELECT *
FROM EMP
WHERE DOJ LIKE '2019%';
```

YYYY-MM-DD

50.WQTD FNAME,LNAME,DOB IF THE EMPS WERE BORN IN THE MONTH OF JAN,FEB OR MARCH???

```
SELECT FNAME,LNAME,DOB
FROM EMP
WHERE DOB LIKE '%01-%' OR DOB LIKE '%02___' OR DOB LIKE '____-03-__';
```

51.WQTD DETAILS OF EMPS THOSE WHO WERE JOINED IN THE DATE OF 12 OR 01???

```
SELECT *
```

```
FROM EMP
WHERE DOJ LIKE '%12' OR DOJ LIKE '%01';
```

52.WQTD FNAME FROM EMP TABLE IF THE FNAME IS NOT STARTING WITH S???

```
SELECT FNAME
FROM EMP
WHERE FNAME NOT LIKE 'S%';
```

NOT LIKE

IT IS USED FOR PATTERN MATCHING.

>IT WILL EXCLUDE THE DETAILS BASED ON THE PATTERN.

SYNTAX:

```
COLUMN_NAME/EXPRESSION NOT LIKE 'PATTERN_TO_MATCH';
```

53.WQTD DETAILS OF EMPS IF THE EMPS WERE NOT BORN DURING THE YEAR 1995??

```
SELECT *
FROM EMP
WHERE DOB NOT LIKE '1995%';
```

54.WQTD DETAILS OF EMPS WHOSE FNAME DOES NOT STARTS WITH S AND D???

```
SELECT *
FROM EMP
WHERE FNAME NOT LIKE 'S%' AND FNAME NOT LIKE 'D%';
```

55.WQTD FNAME,LNAME IF THE LNAME CONTAINS EXACTLY 3 CHARACTERS????

```
SELECT FNAME,LNAME
FROM EMP
WHERE LNAME LIKE '___';
```

56.WQTD DETAILS OF EMPS IF THE EMP FNAME LAST 3RD CHARACTER IS E AND LNAME LAST CHARACTER IS N AND WORKING AS SALESMAN OR MANAGER BUT NOT AS FEMALE??

```
SELECT *
FROM EMP
WHERE FNAME LIKE '%E__' AND LNAME LIKE '%N' AND JOB IN('SALESMAN','MANAGER') AND GENDER!='F';
```

57.WQTD DETAILS OF EMPS IF THE EMP FNAME LAST 2ND CHARACTER IS E LNAME 2ND CHARACTER IS H??

```
SELECT *
FROM EMP
WHERE FNAME LIKE '%E_' AND LNAME LIKE '_H%';
```

58.WQTD DETAILS OF EMPS FROM PENTAGON TABLE IF THE NAME CONTAINS ATLEAST 1 % IN IT??

```
CREATE TABLE PENTAGON
(
ID INT NOT NULL,
NAME VARCHAR(20) NOT NULL
);

INSERT INTO PENTAGON VALUE(1,'RAKS_HI%TH'),(2,'PR_ITHVI'),(3,'KI%RA%N');

SELECT *
FROM PENTAGON
WHERE NAME LIKE '%\%%';

\
--
```

IT IS USED TO REMOVE SPECIAL BEHAVIOUR FROM SPECIAL CHARACTER.

59.WQTD DETAILS OF EMPS FROM PENTAGON IF NAME CONTAINS ATLEAST 2 %???

```
SELECT *
FROM PENTAGON
WHERE NAME LIKE '%\%%\%%';
```

60.WQTD NAME FROM PENTAGON TABLE IF NAME 3RD CHARACTER IS _ (UNDERSCORE);

```
SELECT NAME
FROM PENTAGON
WHERE NAME LIKE '__\_%';
```

61.WQTD DETAILS FROM EMP TABLE IF FNAME STARTS WITH VOWELS???

```
SELECT *
FROM EMP
WHERE FNAME LIKE 'A%' OR FNAME LIKE 'E%' OR FNAME LIKE 'I%' OR FNAME LIKE 'O%' OR FNAME LIKE 'U%' ;
```

62.WQTD FNAME IF THE FNAME IS NOT ENDING WITH VOWELS???

```
SELECT FNAME
FROM EMP
WHERE FNAME NOT LIKE '%A' AND FNAME NOT LIKE '%E' AND FNAME NOT LIKE '%I' AND FNAME NOT LIKE '%O' AND FNAME NOT LIKE '%U';
```

1.AGGREGATE FUNCTION/GROUP FUNCTION

TYPES OF AGGREGATE FUNCTION

1.MAX()

TO OBTAIN MAXIMUM VALUE FROM THE GIVEN COLUMN WE USE MAX().

SYNTAX:

```
MAX(COLUMN_NAME/EXPRESSION);
```

EX:

```
SELECT MAX(SAL)
FROM EMP;
```

2.MIN()

TO OBTAIN MINIMUM VALUE FROM THE GIVEN COLUMN WE USE MIN().

SYNTAX:

```
MIN(COLUMN_NAME/EXPRESSION);
```

EX:

```
SELECT MIN(SAL)
FROM EMP;
```

3.AVG()

TO OBTAIN AVERAGE VALUE FROM THE GIVEN COLUMN WE USE AVG().

SYNTAX:

```
AVG(COLUMN_NAME/EXPRESSION);
```

EX:

SELECT AVG(SAL)
FROM EMP;

4.SUM()

TO OBTAIN TOTAL VALUE FROM THE GIVEN COLUMN WE USE SUM().

SYNTAX:

SUM(COLUMN_NAME/EXPRESSION);

EX:

SELECT SUM(SAL)
FROM EMP;

5.COUNT()

TO OBTAIN NUMBER OF VALUES PRESENT IN GIVEN COLUMN WE USE COUNT().

SYNTAX:

COUNT(*/COLUMN_NAME/EXPRESSION);

EX:

SELECT COUNT(*)
FROM EMP;

NOTE:

ONLY FOR COUNT() WE CAN USE * AS A ARGUMENT.

CHARACTERISTICS OF AGGREGATE FUNCTION

>IT TAKES N NUMBER OF INPUT AND GENERATE ONLY ONE OUTPUT.
>IT EXECUTES GROUP BY GROUP.
>WE CANT USE NORMAL COLUMNS ALONG WITH AGGREGATE FUNCTION.
>WE CAN USE ONLY ONE COLUMN AS A ARGUMENT FOR AGGREGATE FUNCTION.
>WE CANT NEST AGGREGATE FUNCTION.
>WE CANT USE AGGRGATE FUNCTION INSIDE WHERE CLAUSE.
>IT WILL IGNORES NULL VALUE.
**>WE CAN USE GROUP BY EXPRESSION ALONG WITH AGGREGATE FUNCTION.

63.WQTD MAXIMUM SALARY AND MINIMUM SALARY FROM EMP TABLE???

```
SELECT MAX(SAL),MIN(SAL)
FROM EMP;
```

64.WQTD TOTAL SALARY GIVEN TO ALL THE SALESMAN???

```
SELECT SUM(sAL)
FROM EMP
WHERE JOB='SALESMAN';
```

65.WQTD AVERAGE SALARY,TOTAL SALARY,MINIMUM SALARY AND MAXIMUM SALARY GIVEN TO THE EMPS WHOSE FNAME STARTS WITH K OR R????

```
SELECT AVG(sAL),SUM(SAL),MIN(SAL),MAX(SAL)
FROM EMP
WHERE FNAME LIKE 'K%' OR FNAME LIKE 'R%';
```

66.WQTD NUMBER OF EMPS WORKING AS SALESMAN OR MANAGER???.

```
select COUNT(*)
```

```
FROM EMP
WHERE JOB IN('SALESMAN','MANAGER');
```

67.WQTD NUMBER OF EMPS WORKING DNO 111 OR 113 AND FNAME STARTING WITH R BUT NOT AS FEMALE???

```
SELECT COUNT(*)
FROM EMP
WHERE DNO IN(111,113) AND FNAME LIKE 'R%' AND GENDER!='F';
```

68.WQTD NUMBER OF EMPS GETTING SALARY MORE THAN 50000 AND BORN DURING THE DATE 07 AND NOT GETTING ANY COMM???

```
SELECT COUNT(*)
FROM EMP
WHERE SAL>50000 AND DOB LIKE '%07' AND COMM IS NULL;
```

69.WQTD NUMBER OF EMPS WORKING IN EACH DEPT???

```
SELECT COUNT(*),DNO
FROM EMP
GROUP BY DNO;
```

ORDER OF EXECUTION

1.FROM
2.[WHERE]
3.GROUP BY
4.SELECT

70.WQTD TOTAL SALARY SPENT BY THE COMPANY IN EACH JOB ROLE IF THE JOB ROLES ARE SALESMAN,MANAGER OR DEVELOPER???

```
SELECT SUM(SAL),JOB
FROM EMP
WHERE JOB IN('SALESMAN','MANAGER','DEVELOPER')
```

```
GROUP BY JOB;
```

71.WQTD NUMBER OF EMPS WORKING IN EACH DEPT AND GETTING SALARY MORE THAN 32000 AND LESS THAN 50000??

```
SELECT COUNT(*),DNO
FROM EMP
WHERE SAL>32000 AND SAL<50000
GROUP BY DNO;
```

72.WQTD NUMBER OF DIFFERENT JOB ROLE PRESENT IN EMP TABLE???

```
SELECT COUNT(DISTINCT JOB)
FROM EMP;
```

73.WQTD MAXIMUM SALARY GIVEN TO EACH DEPT IF THE EMP IS GETTING SALARY MORE THAN 25000 AND MAXIMUM SALARY MORE THAN 50000???

```
SELECT MAX(sAL),DNO
FROM EMP
WHERE SAL>25000
GROUP BY DNO
HAVING MAX(SAL)>50000;
```

ORDER OF EXECUTION

1.FROM
2.[WHERE]
3.GROUP BY
4.HAVING
5.SELECT

74.WQTD AVERAGE SALARY AND TOTAL SALARY GIVEN TO EACH DEPT IF THE AVG SALARY OF THE DEPT IS MORE THAN 40000???

```
SELECT AVG(sAL),SUM(SAL),DNO
FROM EMP
GROUP BY DNO
HAVING AVG(SAL)>40000;
```

75.WQTD TOTAL SALARY AND NUMBER OF EMPS WORKING IN EACH JOB IF THERE ARE MORE THAN 2 EMPS WORKING IN EACH JOB ROLE???

```
SELECT SUM(sAL),COUNT(*),JOB
FROM EMP
GROUP BY JOB
HAVING COUNT(*)>2;
```

76.WQTD MAXIMUM SALARY AND MINIMUM SALARY GIVEN TO EACH DEPT IF THE DEPT CAONTAINS ATLEAST 3 EMPS WORKING IN IT AND SALARY MUST BE MORE THAN 32000???

```
SELECT MAX(SAL),MIN(SAL),DNO
FROM EMP
WHERE SAL>32000
GROUP BY DNO
HAVING COUNT(*)>=3;
```

77.WQTD MAXIMUM SALARY,MINIMUM SALARY ,TOTAL SALARY AND AVERAGE SALARY GIVEN TO EACH DEPT IF THE DEPT HAS MINIMUM SALARY MORE THAN 30000 AND MAXIMUM SALARY LESS THAN 500000???

```
SELECT MAX(SAL),MIN(SAL),SUM(SAL),AVG(SAL),DNO
FROM EMP
GROUP BY DNO
HAVING MIN(SAL)>30000 AND MAX(SAL)<500000;
```

78.WQTD NUMBER OF EMPS GETTING SAME SALARY????

```
SELECT COUNT(*),SAL
```

```
FROM EMP
GROUP BY SAL
HAVING COUNT(*)>1;
```

79.WQTD NUMBER OF EMPS HAVING SAME GENDER AND WORKING IN SAME DEPT????

```
SELECT COUNT(*),DNO,GENDER
FROM EMP
GROUP BY DNO,GENDER
HAVING COUNT(*)>1;
```

80.WQTD DETAILS OF EMPS ACCORDING TO THEIR SALARY MAXIMUM TO MINIMUM???

ORDER BY

TO ARRENGE THE RECORDS IN EITHER ASCENDING OR DESCENDING ORDER WE USE ORDER BY CLAUSE.

SYNTAX:

```
SELECT COLUMN_NAME/EXPRESSION
FROM TABLE_NAME
ORDER BY COLUMN_NAME ASC/DESC;
```

CHARACTERISTICS OF ORDER BY CLAUSE

>IT IS A LAST EXECUTABLE CLAUSE IN A QUERY.
>IT EXECUTES AFTER SELECT CLAUSE.
>BY DEFAULT IT WILL CONSIDER ASCENDING ORDER FOR THE COLUMNS.
>NORMALLY ALL THE RECORDS IN A TABLE ARRENGED IN ASCENDING ORDER BASED ON PRIMARY KEY COLUMN.
>WE CAN USE ALIAS NAME INSIDE ORDER BY CLAUSE.
>WE CAN USE MULTIPLE COLUMNS INSIDE ORDER BY CLAUSE.

order of EXECUTION

1.FROM
2.[WHERE]
3.GROUP BY
4.HAVING
5.SELECT
6.ORDER BY

80.WQTD DETAILS OF EMPS ACCORDING TO THEIR SALARY MAXIMUM TO MINIMUM???

```
SELECT *
FROM EMP
ORDER BY SAL DESC;
```

81.WQTD FNAME,LNAME AND JOB IF THE EMP IS WORKING AS SALESMAN OR MANAGER AND ARRENGE THE RECORDS ACCORDING TO ALPHABATICAL ORDER OF THEIR FNAME???

```
SELECT FNAME,LNAME,JOB
FROM EMP
WHERE JOB IN('SALESMAN','MANAGER')
ORDER BY FNAME ASC;
```

82.WQTD NUMBER OF EMPS WHO ARE GETTING SAME SALARY AND WORKING AS SAME JOB ROLE AND EMPS ARE GETTING SALARY MORE THAN 30000 AND ARRENGE THE SALARY IN ASCENDING ORDER???

```
SELECT COUNT(*),SAL,JOB
FROM EMP
WHERE SAL>30000
GROUP BY SAL,JOB
HAVING COUNT(*)>1
ORDER BY SAL ASC;
```

83.WQTD DETAILS OF FIRST 3 EMPS FROM EMP TABLE???

LIMIT

TO DISPLAY SPECIFIC NUMBER OF RECORDS FROM RESULTANT TABLE WE USE LIMIT.

SYNTAX:

```
SELECT COLUMN_NAME/EXPRESSION
FROM TABLE_NAME
LIMIT VALUE;
```

83.WQTD DETAILS OF FIRST 3 EMPS FROM EMP TABLE???

```
SELECT *
FROM EMP
LIMIT 3;
```

84.WQTD DETAILS OF FIRST 6 RECORDS FROM EMP TABLE???

```
SELECT *
FROM EMP
LIMIT 6;
```

85.WQTD FIRST RECORD DETAILS FROM EMP TABLE??

```
SELECT *
FROM EMP
LIMIT 1;
```

86.WQTD DETAILS OF 2ND RECORD FROM EMP TABLE???

```
SELECT *
```

```
FROM EMP
LIMIT 1 OFFSET 1;
```

OFFSET

TO SKIP/IGNORE SPECIFIC NUMBER OF RECORDS FROM RESULTANT TABLE WE USE OFFSET.

SYNTAX:

```
SELECT COLUMN_NAME/EXPRESSION
FROM TABLE_NAME
LIMIT VALUE OFFSET VALUE;
```

ORDER OF CLAUSES IN A QUERY

1.SELECT
2.FROM
3.WHERE
4.GROUP BY
5.HAVING
6.ORDER BY
7.LIMIT VALUE OFFSET VALUE

87.WQTD DETAILS OF 6TH AND 7TH RECORD FROM EMP TABLE???

```
SELECT *
FROM EMP
LIMIT 2 OFFSET 5;
```

88.WQTD DETAILS OF TOP 5 MAXIMUM SALARY HOLDERS???

```
SELECT *
FROM EMP
```

```
ORDER BY SAL DESC
LIMIT 5;
```

89.WQTD DETAILS OF LAST 5 RECORDS FROM EMP TABLE???

```
SELECT *
FROM EMP
ORDER BY EID DESC
LIMIT 5;
```

90.WQTD 3RD MAXIMUM SALARY FROM EMP TABLE???

```
SELECT DISTINCT SAL
FROM EMP
ORDER BY SAL DESC
LIMIT 1 OFFSET 2;
```

OR

```
SELECT SAL
FROM EMP
GROUP BY SAL
ORDER BY SAL DESC
LIMIT 1 OFFSET 2;
```

91.WQTD 4TH MINIMUM SALARY FROM EMP TABLE??

```
SELECT DISTINCT SAL
FROM EMP
ORDER BY SAL ASC
LIMIT 1 OFFSET 3;
```

92.WQTD 6TH MAXIMUM SALARY FROM EMP TABLE??

```
SELECT DISTINCT SAL
FROM EMP
ORDER BY SAL DESC
LIMIT 1 OFFSET 5;
```

CHARACTER FUNCTION

CHARACTERISTICS OF CHARACTER FUNCTION

>IT TAKES N NUMBER OF INPUTS AND GENERATE N NUMBER OF OUTPUTS.
>IT EXECUTES ROW BY ROW.
>WE CAN NEST CHARACTER FUNCTION.
>WE CAN USE CHARACTER FUNCTION INSIDE WHERE CLAUSE.

TYPES OF CHARACTER FUNCTION

1.LOWER()

TO CONVERT GIVEN STRING VALUE INTO LOWER CASE WE USE LOWER().

EX:

```
SELECT LOWER('PENTAGON');pentagon
```

2.UPPER()

TO CONVERT GIVEN STRING VALUE INTO UPPER CASE WE USE UPPER().

EX:

```
SELECT UPPER('pentagon');PENTAGON
```

3.LENGTH()

TO OBTAIN THE LENGTH OF GIVEN STRING VALUE WE USE LENGTH().

EX:

SELECT LENGTH('PENTAGON');8

SELECT LENGTH('PENTAGON SPACE');14

4.REVERSE()

TO DISPLAY GIVEN STRING VALUE IN REVERSE FORMAT WE USE REVERSE().

EX:

SELECT REVERSE('PENTAGON');NOGATNEP

5.CONCAT()

TO COMBINE TWO OR MORE STRING VALUES WE USE CONCAT().

SYNTAX:

CONCAT('STR1','STR2')

SAMPLE O/P

MR/MISS RAHUL YOUR SALARY IS 800 RS.

```
SELECT CONCAT('MR/MISS ',FNAME,' YOUR SALARY IS ',SAL,' RS.')
FROM EMP;
```

93.WQTD FNAME AND LNAME TOGETHER AS FULL NAME FROM EMP TABLE???

SAMPLE O/P

SHARUKH KHAN

```
SELECT CONCAT(FNAME,' ',LNAME) AS 'FULL NAME'
FROM EMP;
```

6.SUBSTR()

TO OBTAIN PART OF THE STRING FROM THE ORIGINAL STRING WE USE SUBSTR().

SYNTAX:

```
SUBSTR('ORIGINAL_STRING',POSITION,[LENGTH]);
```

CASE-1

```
B E N G A L U R U
1 2 3 4 5 6 7 8 9
```

```
SELECT SUBSTR('BENGALURU',4,3); GAL
SELECT SUBSTR('BENGALURU',5);ALURU
SELECT SUBSTR('BENGALURU',6,8);LURU
SELECT SUBSTR('BENGALURU',10,2);NULL
SELECT SUBSTR('BENGALURU',6,0);NULL
```

CASE-2

```
B E N G A L U R U
-9 -8 -7 -6 -5 -4 -3 -2 -1
```

```
SELECT SUBSTR('BENGALURU',-3,2);UR
SELECT SUBSTR('BENGALURU',-6);GALURU
SELECT SUBSTR('BENGALURU',-7,-2);NULL
SELECT SUBSTR('BENGALURU',-12,2);NULL
SELECT SUBSTR('BENGALURU',0,2);NULL
```

94.WQTD FNAME,LNAME FROM EMP TABLE IF THE FNAME STARTS WITH K???

```
SELECT FNAME,LNAME
FROM EMP
WHERE SUBSTR(FNAME,1,1)='K';
```

95.WQTD DETAILS OF EMPS WHOSE FNAME ENDS WITH R????

```
SELECT *
FROM EMP
WHERE SUBSTR(FNAME,-1,1)='R';
```

96.WQTD FNAME FROM EMP TABLE IF FNAME STARTS WITH S OR A???

```
SELECT FNAME
FROM EMP
WHERE SUBSTR(FNAME,1,1) IN('S','A');
```

97.WQTD FNAME,LNAME IF THE EMP FNAME STARTS WITH VOWELS AND EMP LNAME ENDS WITH VOWELS???

```
SELECT FNAME,LNAME
FROM EMP
WHERE SUBSTR(FNAME,1,1)IN('A','E','I','O','U') AND SUBSTR(LNAME,-1,1) IN('A','E','I','O','U');
```

98.WQTD DETAILS OF EMPS IF THE EMPS ARE GETTING SALARY MORE THAN 35000 AND GETTING SOME COMMISIION AND FNAME 2ND CHARACTER IS A???

```
SELECT *
FROM EMP
WHERE SAL>35000 AND COMM IS NOT NULL AND SUBSTR(FNAME,2,1)='A';
```

99.WQTD DETAILS OF THE EMPS WHOSE FNAME IS NOT STARTING WITH VOWELS???

```
SELECT *
FROM EMP
WHERE SUBSTR(FNAME,1,1) NOT IN('A','E','I','O','U');
```

100.WQTD DETAILS OF THE EMPS WHO BORN DURING THE YEAR 1995????

```
SELECT *
FROM EMP
WHERE SUBSTR(DOB,1,4)=1995;
```

YYYY-MM-DD
12345678910

101.WQTD FNAME,LNAME,DOJ IF THE EMP JOINED IN THE MONTH OF APRIL,MAY,JUNE OR JULY??

```
SELECT FNAME,LNAME,DOJ
FROM EMP
WHERE SUBSTR(DOJ,6,2)IN(04,05,06,07);
```

102.WQTD FNAME,JOB IF THE JOB IS STARTING WITH SAL OR MAN????

```
SELECT FNAME,JOB
FROM EMP
WHERE SUBSTR(JOB,1,3) IN('SAL','MAN');
```

103.WQTD DETAILS OF EMPS WHOSE REVERSED FNAME MATCHES WITH STRING NAMA????

```
SELECT *
FROM EMP
WHERE REVERSE(FNAME)='NAMA';
```

104.WQT EXTRACT INITIALS FROM FULL NAME????

```
SAMPLE O/P
----------

PUNEETH RAJKUMAR :P.R.
```

```
SELECT CONCAT(SUBSTR(FNAME,1,1),'.',SUBSTR(LNAME,1,1),'.')
FROM EMP;
```

105.WQTD FNAME,LNAME AND JOB TOGETHER IN BELOW FORMAT??

```
SAMPLE O/P
----------
KICCHA SUDEEP (HERO)
```

```
SELECT CONCAT(FNAME,' ',LNAME,' ','(',JOB,')')
FROM EMP;
```

7.REPLACE()

TO REPLACE SUBSTRING FROM NEW_STRING IN ORIGINAL STRING WE USE REPLACE().

SYNTAX:

```
REPLACE('ORIGINAL_STRING','SUBSTRING','NEW_STRING');
```

EX:

```
SELECT REPLACE('PENTAGON','PENT','HEX');HEXAGON

SELECT REPLACE('PENTAGON','N','S');PESTAGOS

SELECT REPLACE('PENTAGON','R','T');PENTAGON

SELECT REPLACE('PENTAGON','N','');PETAGO
```

106.WQT REPLACE A AND I WITH [A] AND [I] IN FNAME

KIRAN:K[I]R[A]N

```
SELECT REPLACE(REPLACE(FNAME,'A','[A]'),'I','[I]')
FROM EMP;
```

107.WQTD FNAME AND PASSWORD,THE PASSWORD MUST CONTAIN BELOW CONDITIONS:

i.FIRST 3 CHARACTERS OF FNAME
ii.LENGTH OF FNAME
iii.LAST 3 DIGITS OF EID

```
SELECT FNAME,CONCAT(SUBSTR(FNAME,1,3),LENGTH(FNAME),SUBSTR(EID,-3,3)) AS PASSWORD
FROM EMP;
```

108.WQTD FIRST HALF OF FNAME FROM EMP TABLE???

```
SELECT SUBSTR(FNAME,1,LENGTH(FNAME)/2)
FROM EMP;
```

109.WQTD 2ND HALF OF FNAME FROM EMP TABLE???

```
SELECT SUBSTR(FNAME,LENGTH(FNAME)/2+1)
FROM EMP;
```

110.WQTD FIRST HALF OF FNAME IN LOWER CASE AND 2ND HALF OF FNAME IN REVERSE FORMAT.

RAJU: raUJ

```
SELECT CONCAT(LOWER(SUBSTR(FNAME,1,LENGTH(FNAME)/2)),REVERSE(SUBSTR(FNAME,LENGTH(FNAME)/2+1)))
FROM EMP;
```

111.WQTD FNAME AND PASSWORD FOR ABHIJIT AND PRIYA,THE PASSWORD MUST CONTAIN BELOW CONDITIONS

i.2ND HALF OF FNAME
ii.LENGTH OF LNAME
iii.REVERSED LAST 3 DIGITS OF MGR
iV.LAST 2 DIGITS OF EID

```
SELECT FNAME,CONCAT(SUBSTR(FNAME,LENGTH(FNAME)/2+1),LENGTH(LNAME),REVERSE(SUBSTR(MGR,-3,3)),SUBSTR(EID,-2,2)) AS
PASSWORD
FROM EMP
WHERE FNAME IN('ABHIJIT','PRIYA');
```

112.WQTD NUMBER OF A PRESENT IN MALAYALAM???

```
SELECT LENGTH('MALAYALAM')-LENGTH(REPLACE('MALAYALAM','A',''));
```

113.WQTD FNAME FROM EMP TABLE IF FNAME CONTAINS EXACTLY 1 A????

```
SELECT FNAME
FROM EMP
WHERE LENGTH(FNAME)-LENGTH(REPLACE(FNAME,'A',''))=1;
```

NUMBER FUNCTION

CHARACTERISTICS OF NUMBER FUNCTION

>IT TAKES N NUMBER OF INPUTS AND GENERATE N NUMBER OF OUTPUTS.

>IT EXECUTES ROW BY ROW

>WE CAN NEST NUMBER FUNCTION

>WE CAN USE NUMBER FUNCTION INSIDE WHERE CLAUSE.

TYPES OF NUMBER FUNCTION

1.ABS()

TO CONVERT A NEGATIVE VALUE INTO POSITIVE VALUE WE USE ABS().

EX:

```
SELECT ABS(-45);45
SELECT ABS(45);45
```

2.MOD()

TO OBTAIN REMAINDER VALUE WE USE MOD().

SYNTAX:

```
MOD(M,N)
```

EX:

```
SELECT MOD(8,2);0
SELECT MOD(8,3);2
```

114.WQTD DETAILS OF EMPS THOSE WHO HAVE EVEN EID???

```
SELECT *
FROM EMP
WHERE MOD(EID,2)=0;
```

3.ROUND()

TO ROUND OFF THE VALUE UPTO SPECIFIED NUMBER OF DECIMAL PLACES WE USE ROUND().

SYNTAX:

```
ROUND(NUMBER,[DECIMAL_PLACE]);
```

EX:

```
SELECT ROUND(123.4);123

SELECT ROUND(123.4567,2);123.46

SELECT ROUND(123.456745,4);123.4567
```

115.WQTD AVERAGE SALARY OBTAINED IN EACH DEPT AND ROUND OFF THE VALUE UPTO 2 DECIMAL PLACE.

```
SELECT ROUND(AVG(SAL),2),DNO
FROM EMP
GROUP BY DNO;
```

4.CEIL

>IT WILL OBTAIN NEXT INTEGER VALUE FROM THE GIVEN DECIMAL VALUE(IF IT IS POSITIVE NUMBER)

>IT WILL OBTAIN CURRENT INTEGER VALUE FROM THE GIVEN DECIMAL VALUE(IF IT IS NEGATIVE NUMBER)

EX:

SELECT CEIL(4.5);5

SELECT CEIL(-4.5);-4

5.FLOOR()

>IT WILL OBTAIN CURRENT INTEGER VALUE FROM THE GIVEN DECIMAL VALUE(IF IT IS POSITIVE NUMBER)

>IT WILL OBTAIN NEXT INTEGER VALUE FROM THE GIVEN DECIMAL VALUE(IF IT IS NEGATIVE NUMBER)

EX:

SELECT FLOOR(3.6);3

SELECT FLOOR(-3.6);-4

6.TRUNCATE()

TO CUT OFF THE VALUE UPTO SPECIFIED NUMBER OF DECIMAL PLACES WITHOUT ROUNDING IT WE USE TRUNCATE().

EX:

SELECT TRUNCATE(123.456,2);123.45

SELECT TRUNCATE(123.45678,4);123.4567

7.POW()

TO OBTAIN POWER VALUE OF THE NUMBER WE USE POW().

EX:

SELECT POW(8,2);64

8.SQRT()

TO OBTAIN SQUARE ROOT VALUE OF A NON NEGATIVE NUMBER WE USE SQRT().

EX:

SELECT SQRT(64);8

SELECT SQRT(-64);NULL

DATE FUNCTION

CHARACTERISTICS OF DATE FUNCTION

>IT TAKES N NUMBER OF INPUTS AND GENRATE N NUMBER OF OUTPUTS.
>IT EXECUTES ROW BY ROW
>WE CAN NEST DATE FUNCTION
>WE CAN USE DATE FUNCTION INSIDE WHERE CLAUSE.

TYPES OF DATE FUNCTION

1.CURDATE()

TO OBTAIN CURRENT DATE FROM THE SYSTEM WE USE CURDATE().

EX:

SELECT CURDATE();2025-03-24

2.SYSDATE()

TO OBTAIN CURRENT DATE AND TIME FROM THE SYSTEM WE USE SYSDATE().

EX:

SELECT SYSDATE(); 2025-03-24 11:09:10

3.YEAR()

TO EXTRACT YEAR FROM THE GIVEN DATE EXPRESSION WE USE YEAR().

EX:

SELECT YEAR('2025-02-04');2025

4.MONTH()

TO EXTRACT MONTH FROM THE GIVEN DATE EXPRESSION WE USE MONTH().

EX:

SELECT MONTH('2020-10-12'); 10

5.DAY()

TO EXTRACT DATE FROM THE DATE EXPRESSION WE USE DAY().

EX:

SELECT DAY('2001-12-15');15

6.DATEDIFF()

TO OBTAIN DAY DIFFERENCE BETWEEN TWO DATE VALUES WE USE DATEDIFF().

EX:

SELECT DATEDIFF(CURDATE(),'2025-03-22'); 2

7.DATE_ADD()

TO ADD SOME TIME INTERVAL FOR A GIVEN DATE WE USE DATE_ADD().

SYNTAX:

DATE_ADD('DATE_VALUE',INTERVAL VALUE UNIT);

INTEVAL:IT IS A KEYWORD USED TO ADD TIME INTERVAL

VALUE:AMOUNT TIME WHICH WE WANT TO ADD

UNIT:UNIT OF TIME INTERVAL(EX:YEAR,MONTH,DAY,.....ETC)

116.WQT ADD 3 YEAR FOR BELOW DATE:

'2021-02-05'

SELECT DATE_ADD('2021-02-05',INTERVAL 3 YEAR);

117.WQT ADD 5 YEAR 6 MONTH FOR BELOW DATE

'1996-06-10'

```
SELECT DATE_ADD(DATE_ADD('1996-06-10',INTERVAL 5 YEAR),INTERVAL 6 MONTH);
```

8.DATE_SUB()

TO SUBTRACT SOME TIME INTERVAL FROM A GIVEN DATE WE USE DATE_SUB().

SYNTAX:

```
DATE_SUB('DATE_VALUE',INTERVAL VALUE UNIT);
```

118.WQT SUBTRACT 3 YEAR 10 MONTH AND 15 DAYS FROM BELOW DATE

'2025-03-24'

```
SELECT DATE_SUB(DATE_SUB(DATE_SUB('2025-03-24',INTERVAL 3 YEAR),INTERVAL 10 MONTH),INTERVAL 15 DAY);
```

119.WQTD DETAILS OF EMPS THOSE WHO WERE BORN IN THE YEAR 1995 AND IN THE MONTH APRIL???

```
SELECT *
FROM EMP
WHERE YEAR(DOB)=1995 AND MONTH(DOB)=04;
```

9.DATE_FORMAT()

TO EXTRACT INDIVIDUAL CHARACTERS FROM THE GIVEN DATE TIME EXPRESSION WE USE DATE_FORMAT().

SYNTAX:

```
DATE_FORMAT('DATE_VALUE','DATE_FORMAT_PATTERN');
```

```
YYYY: '%Y' SELECT DATE_FORMAT(SYSDATE(),'%Y');2025
YY : '%y' SELECT DATE_FORMAT(SYSDATE(),'%y');25
MONTH: '%M' SELECT DATE_FORMAT(SYSDATE(),'%M');MARCH
```

```
MM : '%m' SELECT DATE_FORMAT(SYSDATE(),'%m');03
MON : '%b' SELECT DATE_FORMAT(SYSDATE(),'%b');MAR
DAY : '%W' SELECT DATE_FORMAT(SYSDATE(),'%W');TUESDAY
DAY : '%a' SELECT DATE_FORMAT(SYSDATE(),'%a');TUE
DATE : '%d' SELECT DATE_FORMAT(SYSDATE(),'%d');25
HH24 : '%H' SELECT DATE_FORMAT(SYSDATE(),'%H');10
HH12 : '%h' SELECT DATE_FORMAT(SYSDATE(),'%h');10
MINUTE: '%i' SELECT DATE_FORMAT(SYSDATE(),'%i');52
SECOND: '%s' SELECT DATE_FORMAT(SYSDATE(),'%s');43
TIME : '%T' SELECT DATE_FORMAT(SYSDATE(),'%T');10:53:43
AM/PM: '%p' SELECT DATE_FORMAT(SYSDATE(),'%p');AM
TIME
AM/PM: '%r' SELECT DATE_FORMAT(SYSDATE(),'%r');10:54:39 AM
```

120.WQTD DATE-TIME EXPRESSION IN BELOW FORMAT FOR THE CURRENT DATE AND TIME:

'25-MAR-25 11 AM TUESDAY'

```
SELECT DATE_FORMAT(SYSDATE(),'%y-%b-%d %h %p %W');
```

121.WQTD DETAILS OF THE EMPS IF THE EMPS WERE JOINED IN THE MONTH OF MAY,JUNE OR JULY???

```
SELECT *
FROM EMP
WHERE MONTH(DOJ) IN(05,06,07);
```

OR

```
SELECT *
FROM EMP
WHERE DATE_FORMAT(DOJ,'%M')IN('MAY','JUNE','JULY');
```

122.WQTD DETAILS OF THE EMPS WHO WERE HIRED IN FRIDAY,SATURDAY OR SUNDAY??

```
SELECT *
FROM EMP
WHERE DATE_FORMAT(DOJ,'%W')IN('FRIDAY','SATURDAY','SUNDAY');
```

OR

```
SELECT *
FROM EMP
WHERE DATE_FORMAT(DOJ,'%a')IN('FRI','SAT','SUN');
```

123.WQTD FNAME AND DOB IN US DATE FORMAT???

US DATE FORMAT: MM-DD-YYYY

```
SELECT FNAME,DATE_FORMAT(DOB,'%m-%d-%Y')
FROM EMP;
```

124.WQTD NUMBER OF EMPS HIRED IN EACH YEAR AND DISPLAY THE NUMBERS MAXIMUM TO MINIMUM BASED ON THEIR COUNT????

```
SELECT COUNT(*),YEAR(DOJ)
FROM EMP
GROUP BY YEAR(DOJ)
ORDER BY COUNT(*) DESC;
```

125.WQTD DETAILS OF EMPS WHO WERE HIRED IN THE LEAP YEAR????

```
SELECT *
FROM EMP
WHERE MOD(YEAR(DOJ),4)=0;
```

126.WQTD FNAME AND CURRENT EXPERIENCE OF ALL THE EMPS INTERMS OF YEAR????

```
SELECT FNAME,YEAR(CURDATE())-YEAR(DOJ) AS 'CURRENT EXPERIENCE'
FROM EMP;
```

127.WQTD FIRST HIRED EMPLOYE DOJ???

```
SELECT DOJ
FROM EMP
ORDER BY DOJ ASC
LIMIT 1;
```

```
SELECT MIN(DOJ)
FROM EMP;
```

```
SELECT MAX(DOJ)
FROM EMP;
```

RULES

1.THE COLUMN SELECTED IN INNER QUERY AND THE COLUMN WRITTEN INSIDE OUTER QUERY SHOULD BE OF SAME DATATYPE.

2.IN INNER QUERY WE CAN SELECT ONLY ONE COLUMN.

128.WQTD DETAILS OF THE EMPS WHO ARE GETTING SALARY MORE THAN KARAN???

```
SELECT *
FROM EMP
WHERE SAL>(SELECT SAL
FROM EMP
WHERE FNAME='KARAN');
```

129.WQTD FNAME,LNAME,JOB AND SAL IF THE EMPS ARE WORKING SAME AS DHARANI'S JOB ROLE??

```
SELECT FNAME,LNAME,JOB,SAL
FROM EMP
WHERE JOB=(SELECT JOB
FROM EMP
WHERE FNAME='DHARANI');
```

130.WQTD DETAILS OF THE EMPS WHO WERE HIRED AFTER 2 YEAR OF THE FIRST EMPLOYEE JOINING DATE???

```
SELECT *
FROM EMP
WHERE DOJ>(SELECT DATE_ADD(MIN(DOJ),INTERVAL 2 YEAR)
FROM EMP);
```

131.WQTD DETAILS OF EMPS IF THE EMP FNAME 2ND CHARACTER IS A AND EMP LNAME LAST 2ND CHARACTER IS A AND GETTING SALARY LESS THAN SHIVANI???

```
SELECT *
FROM EMP
WHERE SUBSTR(FNAME,2,1)='A' AND SUBSTR(LNAME,-2,1)='A' AND SAL<(SELECT SAL
FROM EMP
WHERE FNAME='SHIVANI');
```

132.WQTD DETAILS OF EMPS IF THE EMPS ARE ACTING AS CUSTOMER FOR THEIR COMPANY AND HIRED AFTER RAHUL BUT NOT AS MALE????

```
SELECT *
FROM EMP
WHERE CID IS NOT NULL AND DOJ>(SELECT DOJ
FROM EMP
WHERE FNAME='RAHUL') AND GENDER!='M';
```

133.WQTD FNAME AND LNAME TOGETHER AS FULL NAME ALONG WITH SALARY,JOB AND DNO IF THE EMP IS WORKING IN SAME DEPT AS

KARAN'S DEPT AND GETTING SALARY MORE THAN PRIYA???

```
SELECT CONCAT(FNAME,' ',LNAME) AS 'FULL NAME',SAL,JOB,DNO
FROM EMP
WHERE DNO=(SELECT DNO
FROM EMP
WHERE FNAME='KARAN') AND SAL>(SELECT SAL
FROM EMP
WHERE FNAME='PRIYA');
```

134.WQTD DETAILS OF THE EMPS WHO ARE GETTING 2ND MAXIMUM SALARY???

```
SELECT *
FROM EMP
WHERE SAL =(SELECT DISTINCT SAL
FROM EMP
ORDER BY SAL DESC
LIMIT 1 OFFSET 1);
```

135.WQTD DETAILS OF THE EMPS WHO ARE ELDER THAN SHIVANI???

```
SELECT *
FROM EMP
WHERE DOB<(SELECT DOB
FROM EMP
WHERE FNAME='SHIVANI');
```

136.WQTD DETAILS OF EMPS WHO ARE GETTING COMM MORE THAN JAHNAVI AND YOUNGER THAN SAMEER???

```
SELECT *
FROM EMP
WHERE COMM>(SELECT COMM
FROM EMP
WHERE FNAME='JAHNAVI') AND DOB>(SELECT DOB
FROM EMP
WHERE FNAME='SAMEER');
```

137.WQTD DNAME OF THE EMP KARAN???

```
SELECT DNAME
FROM DEPT
WHERE DNO=(SELECT DNO
FROM EMP
WHERE FNAME='KARAN');
```

138.WQTD DETAILS OF THE EMPS WHO ARE WORKING IN ACCOUNTING DEPT???

```
SELECT *
FROM EMP
WHERE DNO= (SELECT DNO
FROM DEPT
WHERE DNAME='ACCOUNTING');
```

139.WQTD LOCATION OF THE CUSTOMER SALMAN???

```
SELECT LOCATION
FROM LOCATION
WHERE LID=(SELECT LID
FROM CUSTOMER
WHERE FIRST_NAME='SALMAN');
```

140.WQTD DETAILS OF THE EMPS WHO ARE WORKING AS SALESMAN OR MANAGER IN DELIVERY DEPT???

```
SELECT *
FROM EMP
WHERE JOB IN('SALESMAN','MANAGER') AND DNO=(SELECT DNO
FROM DEPT
WHERE DNAME='DELIVERY');
```

141.WQTD DETAILS OF THE EMPS WHO ARE WORKING IN ACCOUNTING OR TECHNICAL DEPT???

```
SELECT *
FROM EMP
WHERE DNO IN(SELECT DNO
FROM DEPT
WHERE DNAME IN('ACCOUNTING','TECHNICAL'));
```

```
ERROR:Subquery returns more than 1 row
```

TYPES OF SUBQUERY

1.SINGLE-ROW-SUBQUERY

IF INNER QUERY RETURNS SINGLE VALUE THEN WE CONSIDER THE QUERY AS SINGLE-ROW-SUBQUERY.

>HERE,WE CAN USE BOTH NORMAL OPERATORS(=,!=,>,<) AND SPECIAL OPERATORS(IN,NOT IN,ALL,ANY)

2.MULTI-ROW-SUBQUERY

IF INNER QUERY RETURNS MORE THAN ONE VALUE WE CONSIDER THE QUERY AS MULTI-ROW-SUBQUERY.

>HERE,WE CAN USE ONLY SPECAIL OPERATORS(IN,NOT IN,ALL,ANY)

142.WQTD WORKING LOCATION OF THE EMP PRIYA???

```
SELECT LOCATION
FROM LOCATION
WHERE LID IN(SELECT LID
FROM DEPT
WHERE DNO IN(SELECT DNO
FROM EMP
WHERE FNAME='PRIYA'));
```

143.WQTD EMP FNAME,JOB WHERE THEIR DEPT IS LOCATED IN HEBBAL????

```
SELECT FNAME,JOB
FROM EMP
WHERE DNO IN(SELECT DNO
FROM DEPT
WHERE LID IN(SELECT LID
FROM LOCATION
WHERE LOCATION='HEBBAL'));
```

144.WQTD NUMBER OF CUSTOMERS WHO ARE LIVING IN MUMBAI CITY???

```
SELECT COUNT(*)
FROM CUSTOMER
WHERE LID IN(SELECT LID
FROM LOCATION
WHERE CITY='MUMBAI');
```

145.WQTD LOCATION OF THE CUSTOMERS WHO ARE ACTING AS EMPLOYEES FOR THEIR COMPANY??

```
SELECT LOCATION
FROM LOCATION
WHERE LID IN(SELECT LID
FROM CUSTOMER
WHERE CID IN(SELECT CID
FROM EMP));
```

146.WQTD DETAILS OF THE CUSTOMERS WHO ORDERED SOME PRODUCT????

```
SELECT *
FROM CUSTOMER
WHERE ORDER_ID IS NOT NULL;
```

```
SELECT *
FROM CUSTOMER
WHERE PRODUCT_ID IS NOT NULL;
```

```
SELECT *
FROM CUSTOMER
WHERE ORDER_ID IN(SELECT ORDER_ID
FROM ORDERS);
```

```
SELECT *
FROM CUSTOMER
WHERE PRODUCT_ID IN(SELECT PRODUCT_ID
FROM PRODUCT);
```

147.WQTD DETAILS OF THE EMPS WHO DELIVERED THE PRODUCT TO THE CUSTOMER WHO BELONGS TO MUMBAI CITY????

```
SELECT *
FROM EMP
WHERE EID IN(SELECT EID
FROM ORDERS
WHERE ORDER_ID IN(SELECT ORDER_ID
FROM CUSTOMER
WHERE LID IN(SELECT LID
FROM LOCATION
WHERE CITY='MUMBAI')));
```

148.WQTD DETAILS OF THE EMPS WHO ARE WORKING IN VIJAYANAGAR LOCATION???

```
SELECT *
FROM EMP
WHERE DNO IN(SELECT DNO
FROM DEPT
WHERE LID IN(SELECT LID
FROM LOCATION
WHERE LOCATION='VIJAYANAGAR'));
```

149.WQTD DETAILS OF THE EMPS WHO ARE GETTING SALARY MORE THAN RASHMI???

```
SELECT *
FROM EMP
WHERE SAL>(SELECT SAL
FROM EMP
WHERE FNAME='RASHMI');
```

150.WQTD DETAILS OF THE EMPS WHO ARE GETTING SALARY MORE THAN ALL THE SALESMAN???

```
SELECT *
FROM EMP
WHERE SAL>(SELECT SAL
FROM EMP
WHERE JOB='SALESMAN');
```

```
ALL
---
```

IT IS A MULTIVALUE OPERATOR WHICH TAKES MULTIPLE VALUES AT THE RHS AND SINGLE VALUE AT THE LHS ALONG WITH RELATIONAL OPERATORS.

SYNTAX:

```
LHS RHS
COLUMN_NAME/EXPRESSION >/< ALL(V1,V2,.....,VN);
```

>IT WORKS ON AND CONDITION.

A=10000 T
B=7000 F

```
SAL>ALL(2000,5000,6000,8000);

10000>ALL(2000,5000,6000,8000);
7000>ALL(2000,5000,6000,8000);
```

150.WQTD DETAILS OF THE EMPS WHO ARE GETTING SALARY MORE THAN ALL THE SALESMAN???

```
SELECT *
FROM EMP
WHERE SAL>(SELECT MAX(SAL)
FROM EMP
WHERE JOB='SALESMAN');
```

OR

```
SELECT *
FROM EMP
WHERE SAL>ALL(SELECT SAL
FROM EMP
WHERE JOB='SALESMAN');
```

151.WQTD DETAILS OF THE EMPS WHO ARE GETTING SALARY MORE THAN ANY ONE OF THE SALESMAN?

ANY

IT IS A MULTIVALUE OPERATOR WHICH TAKES MULTIPLE VALUES AT THE RHS AND SINGLE VALUE AT THE LHS ALONG WITH RELATIONAL OPERATOR.

SYNTAX:

```
LHS RHS
COLUMN_NAME/EXPRESSION >/< ANY(V1,V2,....,VN);
```

>IT WORKS ON OR CONDITION.

```
A=10000 T
B=7000 T

SAL>ANY(2000,5000,6000,8000);

10000>ANY(2000,5000,6000,8000);
7000>ANY(2000,5000,6000,8000);
```

151.WQTD DETAILS OF THE EMPS WHO ARE GETTING SALARY MORE THAN ANY ONE OF THE SALESMAN?

```
SELECT *
FROM EMP
WHERE SAL>ANY(SELECT SAL
FROM EMP
WHERE JOB='SALESMAN');
```

OR

```
SELECT *
```

```
FROM EMP
WHERE SAL>(SELECT MIN(SAL)
FROM EMP
WHERE JOB='SALESMAN');
```

152.WQTD NUMBER OF EMPS HIRED IN SAME MONTH???

```
SELECT COUNT(*),MONTH(DOJ)
FROM EMP
GROUP BY MONTH(DOJ)
HAVING COUNT(*)>1;
```

153.WQTD DETAILS OF THE EMPS WHO HAS WORKING EXPERIENCE MORE THAN KIRAN BUT LESS THAN MURALI?????

```
SELECT *
FROM EMP
WHERE DATEDIFF(CURDATE(),DOJ)>(SELECT DATEDIFF(CURDATE(),DOJ)
FROM EMP
WHERE FNAME='KIRAN') AND DATEDIFF(CURDATE(),DOJ)<(SELECT DATEDIFF(CURDATE(),DOJ)
FROM EMP
OR WHERE FNAME='MURALI');
```

```
SELECT *
FROM EMP
WHERE DOJ<(SELECT DOJ
FROM EMP
WHERE FNAME='KIRAN') AND DOJ>(SELECT DOJ
FROM EMP
WHERE FNAME='MURALI');
```

154.WQTD PRODUCT DETAILS WHICH IS DELIVERED BY THE EMP JAHNAVI???

```
SELECT *
FROM PRODUCT
WHERE PRODUCT_ID IN(SELECT PRODUCT_ID
FROM ORDERS
WHERE EID IN(SELECT EID
FROM EMP
WHERE FNAME='JAHNAVI'));
```

ASSIGNMENT

1.WQTD DETAILS OF THE EMPS WHO ARE WORKING IN THE DEPT WHICH IS HAVING HIGHEST SALARY?

2.WQTD FNAME AND WORKING EXPERIENCE OF THE EMPS(INTERMS OF DAYS) WHO WORKED LONGER THAN AVERAGE TENURE AT THE COMPANY???

AVERAGE TENURE:AVERAGE WORKING EXPERIENCE OF ALL THE EMPS.
155.WQTD FULL NAME OF JAHNAVI'S MANAGER???

```
SELECT CONCAT(FNAME,' ',LNAME) AS 'FULL NAME'
FROM EMP
WHERE EID IN(SELECT MGR
FROM EMP
WHERE FNAME='JAHNAVI');
```

156.WQTD DNAME OF RASHMI'S MANAGER????

```
SELECT DNAME
FROM DEPT
WHERE DNO IN(SELECT DNO
FROM EMP
WHERE EID IN(SELECT MGR
FROM EMP
WHERE FNAME='RASHMI'));
```

157.WQTD WORKING LOCATION,CITY AND STATE OF AMAN'S MANAGER???

```
SELECT LOCATION,CITY,STATE
FROM LOCATION
WHERE LID IN(SELECT LID
FROM DEPT
WHERE DNO IN(SELECT DNO
FROM EMP
WHERE EID IN(SELECT MGR
FROM EMP
WHERE FNAME='AMAN')));
```

158.WQTD DETAILS OF RAHUL'S MANAGER'S MANAGER???

```
SELECT *
FROM EMP
WHERE EID IN(SELECT MGR
FROM EMP
WHERE EID IN(SELECT MGR
FROM EMP
WHERE FNAME='RAHUL'));
```

159.WQTD DETAILS OF THE EMPS WHO ARE REPORTING TO RAHUL???

```
SELECT *
FROM EMP
WHERE MGR IN(SELECT EID
FROM EMP
WHERE FNAME='RAHUL');
```

160.WQTD DETAILS OF EMPS WHO ARE REPORTING TO SAMEER'S MANAGER???

```
SELECT *
FROM EMP
WHERE MGR IN(SELECT EID
FROM EMP
WHERE EID IN(SELECT MGR
FROM EMP
WHERE FNAME='SAMEER'));
```

161.WQTD DEPT DETAILS OF THE EMPS WHO ARE REPORTING TO RAHUL'S MANAGER'S MANAGER?

```
SELECT *
FROM DEPT
WHERE DNO IN(SELECT DNO
FROM EMP
WHERE MGR IN(SELECT EID
FROM EMP
WHERE EID IN(SELECT MGR
FROM EMP
WHERE EID IN(SELECT MGR
FROM EMP
WHERE FNAME='RAHUL'))));
```

JOINS

TO RETRIEVE THE DATA FROM MULTIPLE TABLES SIMULTANEOUSELY WE USE JOINS.

TYPES OF JOINS

1.CROSS JOIN/CARTESIAN JOIN
2.INNER JOIN
3.OUTER JOIN:i.LEFT OUTER JOIN/LEFT JOIN

ii.RIGHT OUTER JOIN/RIGHT JOIN
4.SELF JOIN
5.NATURAL JOIN

1.CROSS JOIN

TO MERGE THE RECORDS OF ONE TABLE WITH THE RECORDS OF ANOTHER TABLE WE USE CROSS JOIN.

ANSI:AMERICAN NATIONAL STANDARD INSTITUTE

SYNTAX:

```
SELECT COLUMN_NAME/EXPRESSION
FROM TABLE_NAME T1 CROSS JOIN TABLE_NAME T2;
```

162.WQTD CARTESIAN PRODUCT FROM EMP AND DEPT TABLE???

```
SELECT *
FROM EMP E CROSS JOIN DEPT D;
```

DRAWBACK

IT WILL OBTAIN MORE NUMBER OF INVALID RECORDS/UNMATCHED RECORDS COMPARE TO VALID RECORDS/MATCHED RECORDS.

INNER JOIN

TO OBTAIN MATCHED RECORDS FROM DIFFERENT TABLES WE USE INNER JOIN.

SYNTAX:

```
SELECT COLUMN_NAME/EXPRESSION
FROM TABLE_NAME T1 INNER JOIN TABLE_NAME T2
```

```
ON JOIN_CONDITION;
```

ON:IT IS USED TO PASS JOIN CONDITION

JOIN CONDITION:IT IS USED TO JOIN MULTIPLE TABLES.

```
T1.COLUMN_NAME=T2.COLUMN_NAME
```

163.wqtd DETAILS FROM EMP AND DEPT TABLE??

```
SELECT *
FROM EMP E INNER JOIN DEPT D
ON E.DNO=D.DNO;
```

164.WQTD EMPLOYEE FNAME,JOB AND SAL ALONG WITH DNAME IF THE EMP IS GETTING SALARY MORE THAN 45000???

```
SELECT E.FNAME,E.JOB,E.SAL,D.DNAME
FROM EMP E INNER JOIN DEPT D
ON E.DNO=D.DNO
WHERE E.SAL>45000;
```

165.WQTD CUSTOMER FIRST_NAME ALONG WITH HIS LOCATION AND CITY???

```
SELECT C.FIRST_NAME,L.LOCATION,L.CITY
FROM CUSTOMER C INNER JOIN LOCATION L
ON C.LID=L.LID;
```

166.WQTD EMP FNAME,LNAME WHO ARE ACTING AS DELIVERY BOY???

```
SELECT E.FNAME,E.LNAME
FROM EMP E INNER JOIN ORDERS O
ON E.EID=O.EID;
```

167.WQTD EMP FNAME,LNAME,JOB ALONG WITH HIS DNAME IF THE EMPS ARE WORKING AS SALESMAN OR MANAGER????

```
SELECT E.FNAME,E.LNAME,E.JOB,D.DNAME
FROM EMP E INNER JOIN DEPT D
ON E.DNO=D.DNO
WHERE E.JOB IN('SALESMAN','MANAGER');
```

168.WQTD EMP FNAME,LNAME,DNAME IF THE EMP IS WORKING IN HR DEPT???

```
SELECT E.FNAME,E.LNAME,D.DNAME
FROM EMP E INNER JOIN DEPT D
ON E.DNO=D.DNO
WHERE D.DNAME='HR';
```

169.WQTD DETAILS OF THE EMPS ALONG WITH THEIR DNAME IF THE EMP BORN AFTER 1995???

```
SELECT E.*,D.DNAME
FROM EMP E INNER JOIN DEPT D
ON E.DNO=D.DNO
WHERE E.DOB>'1995-12-31';
OR
WHERE YEAR(E.DOB)>1995;
```

170.WQTD EMP FNAME,LNAME,JOB AND DNO ALONG WITH THEIR DNAME AND LID IF THE EMP IS WORKING AS SALESMAN,MANAGER OR DEVELOPER IN DNO 110,111 OR 112 AND HIS FNAME 3RD CHARACTER IS H???

```
SELECT E.FNAME,E.LNAME,E.JOB,E.DNO,D.DNAME,D.LID
FROM EMP E INNER JOIN DEPT D
ON E.DNO=D.DNO
WHERE E.JOB IN('SALESMAN','MANAGER','DEVELOPER') AND E.DNO IN(110,111,112) AND
SUBSTR(E.FNAME,3,1)='H';
OR
E.FNAME LIKE '__H%';
```

171.WQTD EMP FNAME AND PASSWORD,THE PASSWORD MUST CONTAIN BELOW CONDITIONS:
i.FIRST 3 CHARACTERS OF FNAME
ii.HIS DEPT LID
iii.HIS YEAR OF BIRTH

```
SELECT E.FNAME,CONCAT(SUBSTR(E.FNAME,1,3),D.LID,YEAR(E.DOB)) AS PASSWORD
FROM EMP E INNER JOIN DEPT D
ON E.DNO=D.DNO;
```

172.WQTD EMP FNAME,DNAME AND HIS DEPT LOCATION???

```
SELECT E.FNAME,D.DNAME,L.LOCATION
FROM EMP E INNER JOIN DEPT D
ON E.DNO=D.DNO INNER JOIN LOCATION L
ON D.LID=L.LID;
```

OR

```
SELECT E.FNAME,D.DNAME,L.LOCATION
FROM EMP E INNER JOIN DEPT D INNER JOIN LOCATION L
ON E.DNO=D.DNO AND D.LID=L.LID;
```

173.WQTD CUSTOMER FIRST_NAME,ORDER_ID,DELIVERY_dATE,PRICE AS TOTAL BILL,LOCATION,CITY AND STATE IF THE CUSTOMERS ORDERED SOME PRODUCT????

```
SELECT C.FIRST_NAME,C.ORDER_ID,O.DELIVERY_DATE,P.PRICE AS 'TOTAL BILL',L.LOCATION,L.CITY,L.STATE
FROM CUSTOMER C INNER JOIN ORDERS O INNER JOIN PRODUCT P INNER JOIN LOCATION L
ON C.ORDER_ID=O.ORDER_ID AND O.PRODUCT_ID=P.PRODUCT_ID AND C.LID=L.LID
WHERE C.ORDER_ID IS NOT NULL;
```

174.WQTD REVENUE GENERATED BY EACH STATE????

```
SELECT SUM(P.PRICE),L.STATE
FROM CUSTOMER C INNER JOIN PRODUCT P INNER JOIN LOCATION L
ON C.PRODUCT_ID=P.PRODUCT_ID AND C.LID=L.LID
GROUP BY L.STATE;
```

175.WQTD TOP 2 CUSTOMER FIRST_NAME ALONG WITH THEIR SPENT_MONEY..

```
FIRST_NAME PNAME PRICE
---------- ----- -----
A IPHONE 80000
B SAMSUNG 50000
C IQOO 30000
D OPPO 70000
A NOTHING 40000
```

```
SELECT C.FIRST_NAME,SUM(P.PRICE) AS SPENT_MONEY
FROM CUSTOMER C INNER JOIN PRODUCT P
ON C.PRODUCT_ID=P.PRODUCT_ID
GROUP BY C.FIRST_NAME
ORDER BY SPENT_MONEY DESC
LIMIT 2;
```

OUTER JOINS

1.LEFT OUTER JOIN

TO OBTAIN MATCHED RECORDS ALONG WITH UNMATCHED RECORDS FROM THE LEFT TABLE WE USE LEFT OUTER JOIN.

SYNTAX:

```
SELECT COLUMN_NAME/EXPRESSION
FROM TABLE_NAME T1 LEFT OUTER JOIN TABLE_NAME T2
```

```
ON T1.COLUMN_NAME=T2.COLUMN_NAME;
```

176.WQTD MATCHED AND UNMATCHED RECORDS FROM EMP TABLE???

EMP-LEFT
DEPT-RIGHT

```
SELECT *
FROM EMP E LEFT OUTER JOIN DEPT D
ON E.DNO=D.DNO;
```

RIGHT OUTER JOIN

TO OBTAIN MATCHED AND UNMATCHED RECORDS FROM RIGHT TABLE WE USE RIGHT OUTER JOIN.

SYNTAX:

```
SELECT COLUMN_NAME/EXPRESSION
FROM TABLE_NAME T1 RIGHT OUTER JOIN TABLE_NAME T2
ON T1.COLUMN_NAME=T2.COLUMN_NAME;
```

177.WQTD MATCHED AND UNMATCHED RECORDS FROM DEPT TABLE???

```
SELECT *
FROM EMP E RIGHT OUTER JOIN DEPT D
ON E.DNO=D.DNO;
```

178.WQTD LOCATION WHERE NO CUSTOMERS ARE LIVING?????

```
SELECT L.LOCATION
FROM LOCATION L LEFT OUTER JOIN CUSTOMER C
ON L.LID=C.LID
```

```
WHERE C.LID IS NULL;
```

```
SELECT L.LOCATION
FROM CUSTOMER C RIGHT OUTER JOIN LOCATION L
ON L.LID=C.LID
WHERE C.LID IS NULL;
```

179.WQTD DETAILS OF THE PRODUCTS WHICH ARE NOT ORDERED BY CUSTOMER???

```
SELECT P.*
FROM CUSTOMER C RIGHT OUTER JOIN PRODUCT P
ON C.PRODUCT_ID=P.PRODUCT_ID
WHERE C.CID IS NULL;
```

180.WQTD DETAILS OF THE LOCATION WHERE NO DEPT IS LOCATED???

```
SELECT L.*
FROM LOCATION L LEFT OUTER JOIN DEPT D
ON L.LID=D.LID
WHERE D.DNO IS NULL;
```

181.WQTD MONTHWISE REVENUE GENERATED IN THE YEAR 2022??????

```
SELECT SUM(P.PRICE) AS REVENUE,MONTH(O.ORDER_dATE)
FROM PRODUCT P INNER JOIN ORDERS O
ON P.PRODUCT_ID=O.PRODUCT_ID
WHERE YEAR(O.ORDER_DATE)=2022
GROUP BY MONTH(O.ORDER_dATE);
```

SELF JOIN

TO OBTAIN MATCHED RECORDS FROM SAME TABLE'S WE USE SELF JOIN.

SYNTAX:

```
SELECT COLUMN_NAME/EXPRESSION
FROM TABLE_NAME T1 JOIN TABLE_NAME T2
ON T1.COLUMN_NAME=T2.COLUMN_NAME;
```

182.wqtd EMPLOYEE AND HIS MANAGER DETAILS FROM EMP TABLE??

```
SELECT *
FROM EMP E1 JOIN EMP E2
ON E1.MGR=E2.EID;
```

183.WQTD EMP FNAME,SAL ALONG WITH THEIR MANAGER FNAME,SAL IF THE EMP IS GETTING SALARY MORE THAN 35000 AND MANAGER IS GETTING SALARY LESS THAN 500000???

```
SELECT E1.FNAME,E1.SAL,E2.FNAME,E2.SAL
FROM EMP E1 JOIN EMP E2
ON E1.MGR=E2.EID
WHERE E1.SAL>35000 AND E2.SAL<500000;
```

184.WQTD EMP FNAME,DOB ALONG WITH MANAGER FNAME AND DOB IF THE EMP IS ELDER THAN HIS MANAGER??

```
SELECT E1.FNAME,E1.DOB,E2.FNAME,E2.DOB
FROM EMP E1 JOIN EMP E2
ON E1.MGR=E2.EID
WHERE E1.DOB<E2.DOB;
```

185.WQTD EMPLOYEE FNAME,MANAGER FNAME ALONG WITH MANAGER DNAME?????

```
SELECT E1.FNAME,E2.FNAME,D2.DNAME
FROM EMP E1 JOIN EMP E2 INNER JOIN DEPT D2
ON E1.MGR=E2.EID AND E2.DNO=D2.DNO;
```

186.WQTD EMP FNAME,JOB,MANAGER FNAME,JOB ALONG WITH THIER DNAME IF THE EMP IS WORKING AS MANAGER OR SALESMAN AND EMP DNAME IS EITHER MARKETING OR DELIVERY AND MANAGER IS WORKING AS MANAGER???

```
SELECT E1.FNAME,E1.JOB,E2.FNAME,E2.JOB,D1.DNAME,D2.DNAME
FROM EMP E1 JOIN EMP E2 INNER JOIN DEPT D1 INNER JOIN DEPT D2
ON E1.MGR=E2.EID AND E1.DNO=D1.DNO AND E2.DNO=D2.DNO
WHERE E1.JOB IN('MANAGER','SALESMAN') AND D1.DNAME IN('MARKETING','DELIVERY') AND E2.JOB='MANAGER';
```

E1:EMP TABLE
E2:MANAGER TABLE
D1:EMPLOYEE DEPT TABLE
D2:MANAGER DEPT TABLE

187.WQTD EMP FNAME,DNAME AND DEPT LOCATION ALONG WITH MANAGER FNAME,DNAME AND DEPT LOCATION???

```
SELECT E1.FNAME,D1.DNAME,L1.LOCATION,E2.FNAME,D2.DNAME,L2.LOCATION
FROM EMP E1 JOIN EMP E2 INNER JOIN DEPT D1 INNER JOIN DEPT D2 INNER JOIN LOCATION L1 INNER JOIN LOCATION L2
ON E1.MGR=E2.EID AND E1.DNO=D1.DNO AND E2.DNO=D2.DNO AND D1.LID=L1.LID AND D2.LID=L2.LID;
```

188.WQTD DNAME OF MURALI'S MANAGER???

```
SELECT D2.DNAME
FROM EMP E1 JOIN EMP E2 INNER JOIN DEPT D2
ON E1.MGR=E2.EID AND E2.DNO=D2.DNO
WHERE E1.FNAME='MURALI';
```

189.WQTD NUMBER OF CUSTOMERS LIVING IN HYDERABAD CITY???

```
SELECT COUNT(*)
FROM CUSTOMER C INNER JOIN LOCATION L
ON C.LID=L.LID
WHERE L.CITY='HYDERABAD';
```

190.WQTD FNAME OF AMAN'S MANAGER'S MANAGER??

```
SELECT E3.FNAME
FROM EMP E1 JOIN EMP E2 JOIN EMP E3
ON E1.MGR=E2.EID AND E2.MGR=E3.EID
WHERE E1.FNAME='AMAN';
```

191.WQTD DNAME OF THE EMPS WHO ARE REPORTING TO RAHUL????

```
SELECT D1.DNAME
FROM EMP E1 JOIN EMP E2 INNER JOIN DEPT D1
ON E1.MGR=E2.EID AND E1.DNO=D1.DNO
WHERE E2.FNAME='RAHUL';
```

192.WQTD WORKING CITY OF THE EMPS WHO ARE GETTING 3RD MAXIMUM SALARY???

```
SELECT DISTINCT l.CITY,MAX(E.SAL)
FROM emp e
INNER JOIN dept d ON e.DNO = d.DNO
INNER JOIN location l ON d.LID = l.LID
JOIN emp e2 ON e.SAL < e2.SAL
GROUP BY E.SAL, l.CITY
HAVING COUNT(DISTINCT e2.SAL) = 2;
```

193.WQTD DETAILS OF THE EMPS WHO ARE GETTING SALARY MORE THAN KARAN????

```
SELECT E1.*
FROM EMP E1 JOIN EMP E2
ON E1.SAL>E2.SAL
WHERE E2.FNAME='KARAN';
```

194.WQTD 2ND MAXIMUM SALARY FROM EMP TABLE???

```
SELECT MAX(E1.SAL)
FROM EMP E1 JOIN EMP E2
ON E1.SAL<E2.SAL;
```

195.WQTD 3RD MINIMUM SALARY FROM EMP TABLE???

```
SELECT MIN(E1.SAL)
FROM EMP E1 JOIN EMP E2 JOIN EMP E3
ON E1.SAL>E2.SAL AND E2.SAL>E3.SAL;
```

196.WQTD 5TH MAXIMUM SALARY???

```
SELECT MAX(E1.SAL)
FROM EMP E1 JOIN EMP E2 JOIN EMP E3 JOIN EMP E4 JOIN EMP E5
ON E1.SAL<E2.SAL AND E2.SAL<E3.SAL AND E3.SAL<E4.SAL AND E4.SAL<E5.SAL;
```

197.WQTD DETAILS OF THE EMPS WHO ARE REPORTING TO RAHUL IF RAHUL IS GETTING SALARY MORE THAN KIRAN???

```
SELECT E1.*
FROM EMP E1 JOIN EMP E2 JOIN EMP E3
ON E1.MGR=E2.EID AND E2.SAL>E3.SAL
WHERE E2.FNAME='RAHUL' AND E3.FNAME='KIRAN';
```

NATURAL JOIN

TO OBTAIN MATCHED RECORDS FROM DIFFERENT TABLES BASED ON COMMEN ATTRIBUTE WE USE NATURAL JOIN.

SYNTAX:

```
SELECT COLUMN_NAME/EXPRESSION
FROM TABLE_NAME T1 NATURAL JOIN TABLE_NAME T2;
```

```
SELECT *
```

```
FROM EMP E NATURAL JOIN DEPT D; O/P MATCHED RECORDS
```

```
SELECT *
FROM EMP E NATURAL JOIN LOCATION D;O/P MERGED RECORDS
```

```
SELECT *
FROM EMP E NATURAL JOIN EMP D; O/P EMPTY SET
```

SET OPERATORS

1.UNION
2.UNION ALL
3.INTERSECTION
4.MINUS

A={1,2,3,4,5} B={5,6,7}

A UNION B={1,2,3,4,5,6,7}

A UNION ALL B={1,2,3,4,5,5,6,7}

A INTERSECTION B={5}

A-B={1,2,3,4}

B-A={6,7}
UNION

IT IS USED TO DISPLAY RECORDS FROM MULTIPLE TABLES VERTICALLY.

>HERE,IT WILL AVOID DUPLICATE VALUES FROM THE OUTPUT.

UNION ALL

IT IS USED TO DISPLAY RECORDS FROM MULTIPLE TABLES VERTICALLY.

>HERE,IT WILL INCLUDE DUPLICATE VALUES IN THE OUTPUT.

```
(SELECT FNAME
FROM EMP)
UNION
(SELECT FNAME
FROM EMP);O/P 13 RECORDS
```

```
(SELECT FNAME
FROM EMP)
UNION ALL
(SELECT FNAME
FROM EMP); 26 RECORDS
```

RULES

>WE SHOULD USE SEMICOLEN FOR THA LAST QUERY.
>WE SHOULD USE COMMEN BRACES FOR EACH QUERY
>NUMBER OF COLUMNS SHOULD BE SAME IN SELECT CLAUSE.

198.WQTD JOB,FNAME IN LOWER CASE IF EMPLOYEES ARE WORKING AS SALESMAN ELSE PRINT JOB,FNAME IN REVERSE FORMAT???

```
(SELECT JOB,LOWER(FNAME) AS FNAME
FROM EMP
WHERE JOB='SALESMAN')
```

```
UNION
(SELECT JOB,REVERSE(FNAME)
FROM EMP
WHERE JOB!='SALESMAN');
```

199.WQTD MATCHED AND UNMTACHED RECORDS FROM EMP AND DEPT TABLE???

EMP:LEFT
DEPT:RIGHT

```
(SELECT *
FROM EMP E LEFT OUTER JOIN DEPT D
ON E.DNO=D.DNO)
UNION
(SELECT *
FROM EMP E RIGHT OUTER JOIN DEPT D
ON E.DNO=D.DNO);
```

200.WQTD DETAILS OF 3RD AND 5TH RECORD FROM EMP TABLE???

```
(SELECT *
FROM EMP
LIMIT 1 OFFSET 2)
UNION
(SELECT *
FROM EMP
LIMIT 1 OFFSET 4);
```

CASE-STATEMENT

TO PASS CERTAIN CONDITIONS WHILE DISPLAYING THE RECORDS WE USE CASE-STATEMENT.

SYNTAX:

```
SELECT CASE
WHEN CONDITION_1 THEN VALUE_1
WHEN CONDITION_2 THEN VALUE_2
'
'
'
WHEN CONDITION_N THEN VALUE_N
ELSE DEFUALT_VALUE
END AS ALIAS_NAME

FROM TABLE_NAME;
```

201.WQTD FNAME AND IF THE GENDER='M' PRINT IT AS SIGMA_MALE AND IF THE GENDER='F' PRINT IT AS DADS LIL PRINCESS ELSE PRINT IT AS OTHER???

GENDER:ALIAS NAME

```
SELECT FNAME,CASE
WHEN GENDER='M' THEN 'SIGMA MALE'
WHEN GENDER='F' THEN 'DADS LIL PRINCES'
ELSE 'OTHER'
END AS GENDER
FROM EMP;
```

202.WQTD FNAME,DOJ AND IF THE EMP HIRED BEFORE 2020 PRINT IT AS SENIOR ELSE PRINT IT AS JUNIOR???

ROLE:ALIAS NAME

```
SELECT FNAME,DOJ,CASE
WHEN DOJ<'2020-01-01' THEN 'SENIOR'
ELSE 'JUNIOR'
END AS ROLE
```

FROM EMP
ORDER BY ROLE DESC;
OR
ORDER BY DOJ ASC;

TO DISABLE AUTOCOMMIT

SYNTAX:

SET AUTOCOMMIT=0;

basically,ALL THE DDL COMMANDS ARE AUTOCOMMIT COMMANDS.

TRANSACTION CONTROL LANGAUGE

1.COMMIT
2.ROLLBACK
3.SAVEPOINT

1.COMMIT

TO SAVE ALL THE TRANSACTIONS PERMENENTLY INSIDE THE DATABASE WE USE COMMIT.

SYNTAX:

COMMIT;

ROLLBACK

TO ROLL OUT THE OPERATIONS UPTO PREVIOUSELY USED COMMIT STATEMENT.

SYNTAX:

ROLLBACK;

>WE CAN GETBACK DELETED RECORDS BY USING ROLLBACK,IF THE COMMIT IS NOT USED AFTER DELETE OPERATION.

>WE CAN ROLL OUT THE OPERATIONS UPTO CERTAIN SAVEPOINTS.

SYNTAX_2

ROLLBACK TO SAVEPOINT_NAME;

SAVEPOINT

TO MARK ONE POSITION BETWEEN THE TRANSACTIONS WE USE SAVEPOINT.

>HERE,DATA WILL STORED TEMPORARILY,BUT NOT PERMENENTLY INSIDE THE DATABASE

SYNTAX:

SAVEPOINT SAVEPOINT_NAME;

DCL(DATA CONTROL LANGUAGE)

1.GRANT
2.REVOKE

1.GRANT

TO PROVIDE/GRANT THE PERMISSION OF THE DATA FROM ONE USER TO ANOTHER USER WE USE GRANT COMMAND.

SYNTAX:

GRANT SQL_STATEMENT ON TABLE_NAME TO 'USERNAME'@'HOST_NAME';

TO VIEW ALL THE USERS PRESENT IN MYSQL

STEP_1:

USE INFORMATION_sCHEMA;

STEP_2:

SELECT * FROM USER_ATTRIBUTES;

TO VIEW ACTIVE USER PRESENT IN MYSQL SERVER

SYNTAX:

SELECT USER();

TO CREATE USER IN MYSQL

SYNTAX:

CREATE USER 'USERNAME'@'HOST_NAME'IDENTIFIED BY'PASSWORD';

HOST:LOCALHOST,%

USERNAME:PENTAGON
HOST_NAME:LOCALHOST
PASSWORD:SQL

CREATE USER'PENTAGON'@'LOCALHOST'IDENTIFIED BY'SQL';

TO ACCESS MYSQL USER IN COMMAND PROMPT

SYNTAX:

```
mysql -u username -p
```

```
GRANT SELECT ON EMP TO'PENTAGON'@'LOCALHOST';
```

```
GRANT ALL ON EMP TO'PENTAGON'@'LOCALHOST';
```

ALL:TO PASS ALL THE PERMISSIONS AT A TIME.

REVOKE

TO GETBACK THE PERMISSION OF THE DATA FROM ANOTHER USER WE USE REVOKE.

SYNTAX:

```
REVOKE SQL_STATEMENT ON TABLE_NAME FROM 'USERNAME'@'HOST_NAME';
```

```
REVOKE DELETE ON EMP FROM'PENTAGON'@'LOCALHOST';
```

TO DROP USER FROM MYSQL

SYNTAX:

```
DROP USER 'USERNAME'@'HOSTNAME';
```

```
DROP USER'PENTAGON'@'LOCALHOST';
```

1.IS IT POSSIBLE TO CREATE DUPLICATE TABLE???
YES.....

EMP1:EMP

SYNTAX:

```
-------
CREATE TABLE TABLE_NAME(SELECT * FROM TABLE_NAME);

CREATE TABLE EMP1(SELECT * FROM EMP);
```

2.IS IT POSSIBLE TO CREATE DUPLICATE TABLE WITHOUT RECORDS???
YES......

EMP2:EMP

SYNTAX:

```
CREATE TABLE TABLE_NAME(SELECT * FROM TABLE_NAME WHERE FALSE_CONDITION);
```

EX:

```
CREATE TABLE EMP2(SELECT * FROM EMP WHERE FNAME='ROHIT SHARMA');
```

3.IS IT POSSIBLE TO ADD RECORDS FROM ONE TABLE TO ANOTHER TABLE??
YES...(THE TABLE STRUCTURES SHOULD BE SAME)

EMP2:MANAGERS RECORD FROM EMP TABLE

SYNTAX:

```
INSERT INTO TABLE_NAME(SELECT STATEMENT);
```

```
INSERT INTO EMP2(SELECT * FROM EMP WHERE JOB='MANAGER');
```

```
INSERT INTO EMP2(SELECT * FROM DEPT WHERE DNAME='MARKETING'); O/P:ERROR(TRUCTURE IS NOT SAME)
```

SUB-TABLE

SALESMAN_DATA: SALESMAN RECORDS FROM EMP TABLE

```
CREATE TABLE SALESMAN_DATA(SELECT * FROM EMP WHERE JOB='SALESMAN');
```

VIEW

>IT IS VIRTUAL TABLE
>IT DOESN'T OCCUPY ANY MEMORY INSIDE THE DATABASE.
>TO OVERCOME THE PROBLEM OF SUBTABLE WE USE VIEW.

SYNTAX:

```
CREATE VIEW VIEW_NAME AS(SELECT STATMENT);
```

SALES:SALESMAN RECORDS FROM EMP

```
CREATE VIEW SALES AS(SELECT * FROM EMP WHERE JOB='SALESMAN');
```

TO DROP VIEW

SYNTAX:

```
DROP VIEW VIEW_NAME;
```

RANKING FUNCTIONS

TO ASSIGN RANK FOR EACH RECORD WE USE RANKING FUNCTIONS.

SYNTAX:

```
SELECT RANKING_FUNCTION OVER([PARTITION BY COLUMN_NAME] ORDER BY COLUMN_NAME ASC/DESC)
FROM TABLE_NAME;
```

OVER:IT IS CLAUSE USED TO PASS RANKING FUNCTION IN SELECT CLAUSE.

PARTITION BY:IT IS USED TO CREATE GROUPS AND IT RESETS THE RANK AFTER EACH GROUP.

TYPES OF RANKING FUNCTIONS

1.ROW_NUMBER()
2.RANK()
3.DENSE_RANK()

1.ROW_NUMBER()

TO ASSIGN UNIQUE RANKS FOR ALL THE RECORDS WE USE ROW_NUMBER FUNCTION.

SYNTAX:

```
SELECT ROW_NUMBER() OVER([PARTITION BY COLUMN_NAME] ORDER BY COLUMN_NAME ASC/DESC)
FROM TABLE_NAME;
```

EX:

```
SELECT FNAME,DNO,SAL,ROW_NUMBER() OVER(PARTITION BY DNO ORDER BY SAL DESC) AS 'RANK'
FROM EMP;
```

```
SELECT FNAME,DNO,SAL,ROW_NUMBER() OVER(ORDER BY SAL DESC) AS 'RANK'
FROM EMP;
```

DRAWBACK:

FOR TIED RECORDS IT WILL ASSIGN DIFFERENT RANKS.

2.RANK()

TO ASSIGN RANKS FOR ALL THE RECORDS WE USE RANK FUNCTION
>HERE,IT WILL ASSIGN SAME RANKS FOR TIED RECORDS.

SYNTAX:

```
SELECT RANK() OVER([PARTITION BY COLUMN_NAME] ORDER BY COLUMN_NAME ASC/DESC)
FROM TABLE_NAME;
```

```
SELECT FNAME,DNO,SAL,RANK() OVER(ORDER BY SAL DESC) AS 'RANK'
FROM EMP;
```

DRAWBACK:

>IT WILL ASSIGN SAME RANK FOR TIED RECORDS BUT IT SKIPS THE NEXT RANKING NUMBER.

3.DENSE_RANK()

TO ASSIGN RANKS FOR ALL THE RECORDS WE USE DENSE_RANK() FUNCTION.
>HERE,IT WILL ASSIGN SAME RANK FOR TIED RECORDS ALSO IT REMAINS NEXT RANKING NUMBER SEQUENTIAL.

SYNTAX:

```
SELECT DENSE_RANK() OVER([PARTITION BY COLUMN_NAME] ORDER BY COLUMN_NAME ASC/DESC)
FROM TABLE_NAME;
```

```
SELECT FNAME,DNO,SAL,DENSE_RANK() OVER(ORDER BY SAL DESC) AS 'RANK'
FROM EMP;
```

203.WQTD EMP FNAME WHO ARE GETTING SALARY MORE THAN AVERAGE SALARY IN THEIR DEPT???

A:80000:110 A
B:20000:110

CORRELATED SUBQUERY

>HERE,BOTH OUTER QUERY AND INNER QUERY MUTUALLY DEPENDS ON EACH OTHER.
>THE INNER QUERY WILL EXECUTES FOR EACH RECORD OF OUTER QUERY TABLE

```
SELECT E1.FNAME,E1.SAL,E1.DNO
FROM EMP E1
WHERE E1.SAL>(SELECT AVG(E2.SAL)
FROM EMP E2
WHERE E1.DNO=E2.DNO);
```

204.WQTD EMP FNAME WHO ARE ELDER THAN THEIR MANAGERS???

```
SELECT E1.FNAME
```

```
FROM EMP E1
WHERE E1.DOB<(SELECT E2.DOB
FROM EMP E2
WHERE E1.MGR=E2.EID);
```

KEY ATTRIBUTES:THE ATTRIBUTES WHICH ARE ELIGIBLE TO BECOME PRIMARY KEY

NON KEY ATTRIBUT:THE ATTRIBUTES WHICH ARE NOT ELIGIBLE TO BECOME PRIMARY KEY

SUPER KEY ATTRIBUTE:IT IS A ATTRIBUTE OR COMBINATION OF ATTRIBUTES USED UNIQULY IDENTIFIED THE RECORDS

CANDIDATE KEY:IT IS smallest SUBSET AMONG SUPER KEY ATTRIBUTES
>IN A TABLE WE CAN HAVE MULTIPLE CANDIDATE KEY BUT SINGLE PRIMARY KEY
>ALL THE PRIMARY KEYS ARE CANDIDATE KEY,BUT ALL THE CANDIDATE KEY ARE NOT PRIMARY KEY

PRIMARY KEY:A ATTRIBUTE WHICH IS USED TO UNIQUELY IDENTIFIED THE RECORDS

FOREIGN KEY:A ATTRIBUTE WHICH IS USED TO ESTABLISH CONNECTION BETWEEN MULTIPLE TABLES

COMPOSITE KEY:IT IS A COMBINATION OF TWO OR MORE ATTRIBUTES AMONG SUPER KEY ATTRIBUTES

COMPOUND KEY:IF THE COMPOSITE KEY ATTRIBUTE CONTAINS ATLEAST ONE FOREIGN KEY THEN WE CAN CONSIDER IT AS COMPOUND KEY ATTRIBUTE

DEPENDENCY

IF ONE ATTRIBUTE IS DEPENDS ON ANOTHER ATTRIBUTE THEN THE PROCESS WILL BE KNOWN AS DEPENDENCY.

TYPES OF DEPENDENCY

1.TOTAL FUNCTIONAL DEPENDENCY
2.PARTIAL FUNCTIONAL DEPENDENCY
3.TRANSITIVE FUNCTIONAL DEPENDENCY

TO DROP PROCEDURE

DROP PROCEDURE PROCEDURE_NAME;

REDUNDENCY

THE PROCESS OF REPETATION OF DATA IS KNOWN AS REDUNDENCY.

ANOMALY

THE PROBLEMS OCCURES DUE TO DML OPERATIONS WILL BE KNOWN AS ANOMALY.

TYPES OF ANOMALY

1.INSERT ANOMALY
2.DELETE ANOMALY
3.UPDATE ANOMALY

,