Day-1

1. Display the structure of an EMP table.

desc emp;

2. Display the structure of an DEPT table.

desc dept;

3. Display all the records of EMP table.

select * from emp;

4. Display all the records of DEPT table.

select * from dept;

5. Display only name of all employees.

select ename from emp;

6. Display Employee's Name and Salary.

select ename,salary from emp;

7. Display only unique departments(depno) from EMP table.

select distinct dept_id from dept;

8. Display Employees whose name starts with 'J'.

select * from emp where ename like 'j%';

9. Display all the employees Date of joining(HireDate) and salary(sal).

select hire_date,salary from emp;

10. Display all clerks(job) from EMP table.

select job from emp;

11. Display all employee's name and salary whose salary is more than 2000.

select ename,salary from emp where salary>2000;

12. Display all employees who are not in department number (deptno) 30.

select * from dept where not dept_id=30;

13. Display employees with their empno, ename and mgr (i.e manager's no).

14. Display managers number, job profile, department number and salary of employees allen,adams,jones and Blake.

15. Display only unique salary(sal) from EMP table.

select distinct salary from emp;

16. Display the location of department number 30.

select location from dept where dept_id=30;

17. Display the details of the department located in 'new york' city.

select * from dept where location='new york';

18. Display all the employees according to their names in sorted order.

select * from emp order by ename;

19. Display all the employees who are not salesman nor the manager.

```
select * from emp where not job='salesman' and not job='manager';
```

20.Display the details of a clerk who is getting salary more than 1000.

```
select * from emp where salary>1000;
```

DAY-2

1. **create below given table called "client_master" with the constrains and insert minimum 5 records in it.**

Column	Datatype and Size	Constraints
C_No	**Varchar (4)**	**Primary Key**
Client_Name	**Varchar (25)**	**Not Null**
Pincode	**Number (6)**	**Not Null**
Birth_Date	**Date**	**Not Null**
Occupation	**Varchar (15)**	**Not Null**

Add the following constraints with the table fields:

1. **Client_No must start with the capital letter 'C' only.**
2. **Pincode must be of exactly 6-digit number.**

C_No	Client_Name	Pincode	Birth_Date	Occupation
C001	**Kajal Oza**	**380036**	**14-july-1987**	**Manager**
C002	**Vraj Shah**	**380058**	**20-nov-1986**	**CEO**
C003	**Dhyan Dave**	**387903**	**17-June-1988**	**Doctor**
C004	**Mira Vaid**	**398006**	**5-jan-1976**	**Professor**
C005	**Preeti Patel**	**390040**	**28-feb-1971**	**Clerk**

```
CREAT TABLE Client_Master (
    C_NO varchar(4) check (C_NO like 'C%') primary key,
    Client_Name varchar(25) not null,
    Pincode number(6) not null,
    Birth_Date date not null,
    Occupation varchar(15) not null );
```

2. create following tables with constraints and records as shown in the example below.

Table: Product_Master

P_No	Description	Profit_Per	Unit	Qty_Hand	ReOrder	Sell_P	Cost_P
P001	T-Shirt	5	Piece	200	50	350	250
P002	Jeans	6	Piece	150	40	500	350
P003	Skirt	6	Piece	100	50	350	200
P004	Saree	3	Piece	100	20	800	600

Query:

```
CRATE TABLE Product_Master(
        P_NO varchar(4) check (P_NO like 'P%') primary key,
        Description varchar(10)  NOT NULL,
        Profit_Per number(1)  NOT NULL,
        Unit varchar(6) NOT NULL,
        Qty_Hand number(3)  NOT NULL,
        Reorder number(2)  NOT NULL,
        Sell_P number(3)  NOT NULL,
        Cost_P number(3)  NOT NULL  );
```

3. Create Table and Insert minimum 10 records in Salesman_Master Table as shown below. (Sample data are shown for your reference):

S_No	Name	Add1	Add2	City	Pin Code	State	Salary	Target	Sales
S001	Aman	3,Dev Appt	Vaghodia	Baroda		Gujarat	5000	100	50
S002	Omkar			Bhopal		MP	4500	200	150
S003	Raj	B-104	Verli	Mumbai	400002	Maharashtra	5500	200	200
S004	Ashish	1, Smruti	Ghoddol	Surat		Gujarat	4500	150	100

Query:

```
CREAT TABLE Salesman_Master(
        S_NO varchar(4) check(S_NO like 'S%') primary key,
        Name varchar(8) NOT NULL,
        Add1 varchar(10)  NULL,
        Add2 varchar(10)  NULL,
        City varchar(8) NOT NULL,
        Pincode number(6) NULL,
        State varchar(12)  NOT NULL,
        Salary number(6) NOT NULL,
        Target number(3) NOT NULL,
        Sales number(3) NOT NULL  );
```

4. Create Sales_Order and Sales_Order_Details table as shown below. Insert minimum 10 records in it.

Sales_Order

Fields	Description
Order_No	**Primary Key, Varchar (4), Starts with 'O'**
Client_No	**References Client_Master**
Order_Date	**Date**
Salesman_No	**Refers Salesman_Master**
Delivery_Type	**Free or Paid. Use Char(1) i.e 'F' or 'P'**
Bill_Paid	**Yes ('Y') or No ('No')**
Delivery_Date	**Date**
Order_Status	**'In Process', 'Pending', 'Fulfilled'**

Sales_Order_Details: Order_No and Product_No are the composite PK

Fields	Description
Order_No	**Refers Sales_Order Table**
Product_No	**Refers Product_Master**
Qty_Ordered	**Positive integer**
Qty_Dispatched	**Positive integer**
Product_Rate	**Product Selling Price * Qantity Delivered**

Query-1:

```
CREATE TABLE Sales_Order(
        Order_NO varchar(4) check(Order_NO like 'O%') primary key,
        Client_no varchar(4) references Client_Master(C_NO),
        Order_date date,
        Salesman_no varchar(4) references Salesman_Master,
        Delivery_type char(1) check (delivery_type in('f','p')),
        Bill_paid char(1) check(bill_paid in('y','n')),
        Delivery_date date,
        Order_status varchar(10) check(order_status in('in process','pending','fulfilled')));
```

Query-2:

```
create table Sales_Order_Details(
        Order_NO varchar(4) references Sales_Order,
        Product_no varchar(4) references Product_Master,
        Qty_ordered int,
        Qty_dispatched int,
        Product_rate int );
```

DAY-3

1. **Create a table called “Clients” from the “Client_Master” table.**

   create table client as select * from client_master

   output

   SQL> desc client

Name	Null?	Type
C_NO		VARCHAR2(10)
CLIENT_NAME	NOT NULL	VARCHAR2(25)
PINCODE	NOT NULL	NUMBER(6)
BIRTH_DATE	NOT NULL	DATE
OCCUPATION	NOT NULL	VARCHAR2(15)

2. **Insert all records of “Client_Master” into “Clients”**

   insert into client select * from client_master

   output:

C_NO	CLIENT_NAME	PINCODE	BIRTH_DAT	OCCUPATION
C001	kajal oza	380036	14-JUL-87	manager

3. **Create table “Products” from “Product_master”**

   create table products as select * from product_master

output:

```
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
P_NO                                               VARCHAR2(10)
DESCRIPTION                                        VARCHAR2(10)
PROFIT_PER                                         NUMBER(10)
UNIT                                               VARCHAR2(10)
QTY_HAND                                           NUMBER(10)
REORDER                                            NUMBER(10)
SELL_P                                             NUMBER(10)
COST_P                                             NUMBER(10)
```

4. **Copy only selected Product's information into "Products" table.**

   create table Productss as select P_NO, DESCRIPTION, unit,qty_hand from products

   output:

```
P_NO       DESCRIPTIO UNIT       QTY_HAND
---------- ---------- ---------- ----------
P001       T-Shirt    piece           200
```

5. **Create table "Salesman" from Salesman_Master" where "Salesman" table contains fields: Salesman number, Name, Address1, City, Target_Get, Achieved_Target**

   create table salesman as select s_no,name,add1,city,target from salesman_master

Output:

SQL> desc salesman

Name	Null?	Type
S_NO		VARCHAR2(10)
NAME		VARCHAR2(10)
ADD1		VARCHAR2(20)
CITY		VARCHAR2(20)
TARGET		NUMBER(10)

6. **Insert all records for the fields in new table from "Salesman_Master" table**

insert into salesman select s_no,name,add1,city,target from salesman_master

output:

Name	Null?	Type
S_NO	NOT NULL	VARCHAR2(10)
NAME		VARCHAR2(10)
ADD1		VARCHAR2(20)
ADD2		VARCHAR2(20)
CITY		VARCHAR2(20)
PINCODE		NUMBER(10)
STATE		VARCHAR2(10)
SALARY		NUMBER(10)
TARGET		NUMBER(10)
SALES		NUMBER(10)

7. **Create an "Order_Master" table form "Sales_Order" Tabl**

create table order_master as select * from sales_order

output:

Name	Null?	Type
ORDER_NO		VARCHAR2(4)
CLIENT_NO		VARCHAR2(10)
ORDER_DATE		DATE
SALESMAN_NO		VARCHAR2(10)
DELIVERY_TYPE		CHAR(1)
BILL_PAID		CHAR(1)
DELIVERY_DATE		DATE
ORDER_STATUS		VARCHAR2(10)

8. **Insert appropriate records in "Order_Master" from Sales_Order" table**

   insert into order_master select * from sales_order

   output:

Name	Null?	Type
ORDER_NO	NOT NULL	VARCHAR2(4)
CLIENT_NO		VARCHAR2(10)
ORDER_DATE		DATE
SALESMAN_NO		VARCHAR2(10)
DELIVERY_TYPE		CHAR(1)
BILL_PAID		CHAR(1)
DELIVERY_DATE		DATE
ORDER_STATUS		VARCHAR2(10)

9. **Create "Order_Detail" table from "Sales_Order_Detail" table**

   create table order_detail as select * from sales_order_details

output:

```
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
ORDER_NO                                           VARCHAR2(4)
PRODUCT_NO                                         VARCHAR2(10)
QTY_ORDERED                                        NUMBER(10)
QTY_DISPATCHED                                     NUMBER(10)
PRODUCT_RATE                                       NUMBER(10)
```

10. Insert appropriate records in "Order_Detail" from Sales_Order_ Detail" table.

insert into order_detail select * from sales_order_details

output:

```
ORDE PRODUCT_NO QTY_ORDERED QTY_DISPATCHED PRODUCT_RATE
---- ---------- ----------- -------------- ------------
O1   P001              5           4        1400
```

11. Destroy tables: Client_Master, Product_Master and Salesman_Master table using Truncate

TRUNCATE TABLE client_master

TRUNCATE TABLE Product_Master

TRUNCATE TABLE Salesman_Master

12. Destroy tables: "Sales_Order" and Sales_Order_Details using drop operation

DROP TABLE sales_order

DAY-4

1.Remove the product details column from Product_Master Table:

SQL> ALTER TABLE PRODUCT_MASTER DROP COLUMN DESCRIPTION;

Table altered.

2.Remove all Sales_Orders whose QtyOrdered = 1 in "Sales_Order_Detail" table:

SQL> DELETE FROM SALES_ORDER_DETAISS WHERE QTY_ORDERED=1;

1 row deleted.

3.Modify the OrderStatus to "Fulfilled" where the OrderStatus is "In Process" in Sales_Order Table.

SQL> UPDATE SALES_ORDER SET ORDER_STATUS ='FULFILED' WHERE ORDER_NO='O5';

1 row updated.

4.Add column City and State in Client_Master table.

SQL> ALTER TABLE CLIENT_MASTER ADD (CITY VARCHAR(20));

Table altered.

SQL> ALTER TABLE CLIENT_MASTER ADD (STATE VARCHAR(20));

Table altered.

5.Add a Client record for Client Name Ashwini Joshi, city = Banglore:

SQL> INSERT INTO CLIENT_MASTER VALUES

('C006','ASHWINI JOSHI',387903,'17-June-1988','Doctor','BANGLORE',NULL);

1 row created.

6.Change the city of the client “Ashwini Joshi” from “Bangalore” to “Chennai”:

SQL> UPDATE CLIENT_MASTER SET CITY='CHENNAI' WHERE CITY='BANGLORE';

1 row updated.

7.Change the Cost Price (Cost_P) of of a Skirt:

SQL> UPDATE PRODUCT_MASTER SET COST_P=400 WHERE DESCRIPTION='SKIRT';

1 row updated.

8.Increase the sell price of saree by Rs. 20:

SQL> UPDATE PRODUCT_MASTER SET SELL_P=SELL_P+20 WHERE P_NO='P004';

1 row updated.

9.Delete all Salesman from Salesman_Master table whose salary is more than 3500:

SQL> DELETE FROM SALESMANN_MASTERR WHERE SALARY >=3500;

5 rows deleted.

10.Delete all Products from Product_master where Quantity on Hand (Qty_Hand) is less than 50:

SQL> DELETE FROM PRODUCT_MASTER WHERE QTY_HAND<50;

0 rows deleted.

11.Add a column Mobile_No of type Number, in the Client_Master Table:

SQL> ALTER TABLE CLIENT_MASTER ADD (MOBILE_NO NUMBER(10));

Table altered.

DISPLAY TABLE:

SQL> DESC CLIENT_MASTER

Name

C_NO

CLIENT_NAME

PINCODE

BIRTH_DATE

OCCUPATION

CITY

STATE

MOBILE_NO

12.Change the name of Salesman_Master table to "SMan_Mst":

SQL> RENAME SALESMANN_MASTERR TO SMan_Mst;

Table renamed.

DISPLAY TABLE:

SQL> DESC SMan_Mst;

Name

S_NO

NAME

ADD1

ADD2

CITY

PINCODE

STATE

SALARY

TARGET

SALES

DAY-5

1. Display all the clients whose age is more than 25 years but less than 50 yesrs

```
select  CLIENT_NAME, to_char(sysdate, 'yyyy') - to_char(birth_date , 'yyyy')
 from CLIENT_MASTER
 where to_char(sysdate, 'yyyy') - to_char(birth_date , 'yyyy') > 25 and to_char(sysdate,
'yyyy') - to_char(birth_date , 'yyyy') < 50
```

```
CLIENT_NAME            TO_CHAR(SYSDATE,'YYYY')-TO_CHAR(BIRTH_DATE,'YYYY')
---------------------- ------------------------------------------------
Kajal Oza                                      35
Vraj Shah                                      36
Dhyan Dave                                     34
Mira Vaid                                      46
```

2. Display all the Salesman from Salesman_Master table whose “Sale” is more than his “Target” and “Target” is more than 100

```
 select  target,sales,name from salesman_master where  target > 100 and  sales >target;
```

```
 NAME                 TARGET      SALES
------------------------ ---------- ----------
Sanket                    200        300
```

3. Find the Salesman who are neither from Baroda nor from Surat

select name from salman_master where city Not in('baroda','surat');

SQL> select name from salman_master where city Not in('baroda','surat');

NAME

---

sanket

harsh

sachin

ajit

rahil

krunal

4. Display total number of clients using Client_Master Table.

select count(C_No) from Client_Master;

5. Display the highest salary s salesman is getting.

select max(Salary) from salesman_Master;

MAX(SALARY)

8000

6. Display all item names in upper case letters only.

```
SELECT UPPER(description) AS Uppercasedescription
FROM product_master;

UPPERCASEDESCRIPTION
--------------------
T-SHIRT
JEANS
SKIRT
SAREE
```

7. Display current date and time

8. Display average target given to the salesman

```
select avg(target) target from salman_master;
 AVG(TARGET)
 -----------
  139.7
```

9. Display the Birth Date (DOB in Student_Master Table) in a new format. (Eg. February 12, 1998)

```
select to_char(birth_date,'Month dd, yyyy') AS BIRTH_DATE from client_master;

BIRTH_DATE
------------------------------------------
July      14, 1987
November  20, 1986
June      17, 1988
January   05, 1976
February  28, 1971
```

10. Display Date of Joining (DOJ, Faculty_Master) of all faculties in DD/MM/YY format;

```
select to_char(birth_date,'dd/mm/yy') AS JOINING_DATE from client_master;


JOINING_
--------
14/07/87
20/11/86
17/06/88
05/01/76
28/02/71
```

11. Display Only Birth Date and Month of all the students from Student_Master Table

```
select to_char(birth_date,'dd-Month') AS BIRTH_DATE from client_master;
```

BIRTH_DATE

14-July

20-November

17-June

05-January

28-February

Employee_No	Name	Pincode	DOB	Address	Salary	Occupation
E100	Kapil	380036	14-july-1987	Bombay	10000	Manager
E101	Manish	385670	20-nov-1986	Pune	5000	CEO
E102	Ramesh	387903	17-June-1988	Goa	25000	Engineer
E103	Rohan	390009	15-jan-1976	Delhi	35000	Clerk
E104	Raj	398006	28-feb-2000	Abad	30000	Doctor

Create table Emp;

```
create table emp (
e_no varchar(4) primary key,
name varchar(10),
pincode number(6),
dob date,
add1 varchar(20),
sal number(6),
occuption varchar(15));
```

```
insert into emp values('E100','Kapil',380051,'24-nov-2001','Bombay',100000,'job');
insert into emp values('E101','Manish',382350,'20-jul-2005','Pune',50000,'CEO');
insert into emp values('E102','Ramesh',382360,'26-feb-1999','Goa',250000,'Engineer');
insert into emp values('E103','Rohan',382370,'20-jun-1995','Delhi',35000,'Clerk');
insert into emp values('E104','Raj',382380,'20-jan-2005','Ahmedabad',30000,'Doctor');
```

```
E_NO NAME       PINCODE DOB       ADD1                SAL OCCUPTION
---- ---------- ---------- --------- ------------------- ---------- ---------------
E100 Kapil       380051 24-NOV-01 Bombay               100000 job
E101 Manish      382350 20-JUL-05 Pune                 50000 CEO
E102 Ramesh      382360 26-FEB-99 Goa                  250000 Engineer
E103 Rohan       382370 20-JUN-95 Delhi                35000 Clerk
E104 Raj         382380 20-JAN-05 Ahmedabad            30000 Doctor
```

1. List the names of all employee having ‘a’ as the second letter in their names

```
select name from emp where name like'_a%';
```

```
NAME
----------
Kapil
Manish
Ramesh
Raj
```

2. List the employee whose occupation fist letter is ‘M’

```
select name,occuption from emp where occuption like'M%';
```

NAME OCCUPTION

---------- ----------

Kapil Manager

3. List the employees who have the second character as a or o

select name from emp where name like'_a%' or name like'_o%';

NAME

Kapil

Manish

Ramesh

Rohan

Raj

4. List the employee details of the named Rohan, Ramesh and Raj;

select * from emp where name IN('Rohan','Ramesh','Raj');

E_NO NAME PINCODE DOB ADD1 SAL OCCUPTION

---- ---------- ---------- --------- ---------- ---------- ----------

E102 Ramesh 382360 26-FEB-99 Goa 250000 Engineer

E103 Rohan 382370 20-JUN-95 Delhi 35000 Clerk

E104 Raj 382380 20-JAN-05 Ahmedabad 30000 Doctor

5. List all employee who stay in 'Banglore' or 'Pune'.

select name,add1 from emp where add1 like'Banglore' or add1 like'Pune';

NAME ADD1

---------- --------------------

Manish Pune

6. List the details of employee E102 and E104

select * from emp where e_no IN('E102','E104');

E_NO NAME PINCODE DOB ADD1 SAL OCCUPTION

---- ---------- ---------- --------- -------------------- ---------- ---------------

E102 Ramesh 382360 26-FEB-99 Goa 25000 Engineer

E104 Raj 382380 20-JAN-05 Ahmedabad 30000 Doctor

7. List the details of employee whose salary is greater than 5000 and less than 20000.

select * from emp where sal > 5000 and sal < 20000;

E_NO NAME PINCODE DOB ADD1 SAL OCCUPTION

---- ---------- ---------- --------- -------------------- ---------- ---------------

E100 Kapil 380051 24-NOV-01 Bombay 10000 Manager

8. List the name of employees who are not in the state 'Bombay'.

select name,add1 from emp where add1 NOT IN'Bombay';

NAME ADD1

---------- --------------------

Manish Pune

Ramesh Goa

Rohan Delhi

Raj Ahmedabad

9. List employee whose salary is less than 10000 and calculate a new incremented salary by 15% as Salary * .15

```
select sal + (sal*0.15) as New_Salary from emp where sal < 10000
```

```
NEW_SALARY
----------
     5750
```

10. Count the total number of employees

```
select count(e_no) from emp;
```

```
COUNT(E_NO)
-----------
         5
```

11. Calculate the average salary of all the employees.

```
select avg(Sal) as Average_Salary from emp;
```

```
AVERAGE_SALARY
--------------
        21000
```

12. Determine the maximum and minimum salary. Rename the output as max_salary and min_salary respectively

```
select max(sal) As Max_salary, min(sal) AS min_salary from emp;

MAX_SALARY MIN_SALARY
---------- ----------
    35000       5000
```

13. Count the number of employees having salary less than or equal to 15000.

```
 select count(e_no) from emp where sal <= 15000;

COUNT(E_NO)
-----------
         2
```

14. List the details of Employee month wise in DD/MM/YY format.

```
select to_char(DOB,'dd/mm/yy') AS birth_date from emp;

BIRTH_DA
--------
24/11/01
20/07/05
26/02/99
20/06/95
20/01/05
```

15. List the DOB in the format 'DD-Month-YY' e.g 12-February-91.ss

```
select to_char(DOB,'dd-MOnth-yy') AS birth_date from emp

BIRTH_DATE

-----------------------------------------

24-NOVEMBER -01

20-JULY     -05

26-FEBRUARY -99

20-JUNE     -95

20-JANUARY  -05
```

DAY-6

1. create applicant table

APPLICANT (AID, A_Name, City, B_Date)

ENTRANCE_TEST (ET_ID, ET_Name, Max_Score)

ETEST_DETAILS (AID, ETID, ETest_Date, Score)

```
create table applicant(
        A_ID varchar(5) primary key,
        A_name varchar(10),
        city varchar(10),
        B_date date);
```

```
insert into applicant values ('A001','Sanket','Ahmedabad','14-jul-2002');
insert into applicant values ('A002','Priya','Surat','21-feb-2001');
insert into applicant values ('A003','Vanita','Mumbai','07-jul-2003');
insert into applicant values ('A004','Amisha','Rishikesh','11-nov-2001');
insert into applicant values ('A005','Krishna','Ahmedabad','07-sep-2000');
```

```
A_ID  A_NAME     CITY       B_DATE
----- ---------- ---------- ---------
A001  Sanket     Ahmedabad  14-JUL-02
A002  Priya      Surat      21-FEB-01
A003  Vanita     Mumbai     07-JUL-03
A004  Amisha     Rishikesh  11-NOV-01
A005  Krishna    Ahmedabad  07-SEP-01
```

Create Entrance test table

```
create table entrance_test(
        Et_ID varchar(4) primary key,
        ET_name varchar(15),
        max_score number(4)
);

insert into entrance_test values('E001','Programming',100);
insert into entrance_test values('E002','Maths',100);
insert into entrance_test values('E003','Web Designing',70);
insert into entrance_test values('E004','Web DEvloping',80);
```

```
ET_I ET_NAME         MAX_SCORE
---- --------------- ----------
E001 Programming          100
E002 Maths                100
E003 Web Designing         70
E004 Web DEvloping         80
```

Create table Etest Details

```
create table Etest_detail23(
        AID varchar(5) references applicant(A_ID),
        ETID varchar(4) references entrance_test(ET_ID),
        ETEST_date date,
        score number(5),
        primary key(AID,ETID)
);
```

```
insert into Etest_detail values('A001','E004','15-jun-2022',70);
insert into Etest_detail values('A002','E002','18-jun-2022',85);
insert into Etest_detail values('A004','E003','19-jun-2022',90);
insert into Etest_detail values('A003','E001','20-jun-2022',70);
insert into Etest_detail values('A002','E004','28-jun-2022',65);
insert into Etest_detail values('A003','E003','18-jun-2022',87);
insert into Etest_detail values('A002','E002','18-jun-2022',87);
insert into Etest_detail values('A002','E001','21-jun-2022',90);
insert into Etest_detail values('A001','E003','20-jun-2022',82);
insert into Etest_detail values('A001','E001','15-jun-2022',95);
```

Table 2:

```
create table Etest_detail23(
        AID varchar(5) references applicant(A_ID),
        ETID varchar(4) references entrance_test(ET_ID),
        ETEST_date date,
        score number(5),
        primary key(AID,ETID)
);
```

```
insert into Etest_detail23 values('A001','E001','15-jun-2022',70);
insert into Etest_detail23 values('A001','E002','16-jun-2022',85);
insert into Etest_detail23 values('A001','E003','17-jun-2022',90);
insert into Etest_detail23 values('A001','E004','18-jun-2022',70);
insert into Etest_detail23 values('A002','E001','15-jun-2022',65);
insert into Etest_detail23 values('A002','E002','16-jun-2022',87);
```

```
insert into Etest_detail23 values('A002','E003','17-jun-2022',87);
insert into Etest_detail23 values('A002','E004','18-jun-2022',90);
insert into Etest_detail23 values('A003','E001','15-jun-2022',82);
insert into Etest_detail23 values('A003','E002','16-jun-2022',95);
insert into Etest_detail23 values('A003','E003','17-jun-2022',95);
insert into Etest_detail23 values('A003','E004','18-jun-2022',95);
insert into Etest_detail23 values('A004','E001','15-jun-2022',95);
insert into Etest_detail23 values('A004','E002','16-jun-2022',95);
insert into Etest_detail23 values('A004','E003','17-jun-2022',95);
insert into Etest_detail23 values('A004','E004','18-jun-2022',95);
```

```
AID   ETID ETEST_DAT     SCORE
----- ---- --------- ----------
A001  E004 15-JUN-22        70
A002  E002 18-JUN-22        85
A004  E003 19-JUN-22        90
A003  E001 20-JUN-22        70
```

1. How many applicants have appeared for each test

```
select AID,count(*) AS Test_No from etest_detail group by AID;
```

```
AID      TEST_NO
----- ----------
A001          2
A002          3
A003          2
```

A004 1

2. Display highest score for each test

select AID,Max(score) AS Highest_Score from etest_detail group by AID;

AID HIGHEST_SCORE

----- -------------

A004 90

A003 87

A001 87

A002 90

3. Display applicant's ID who appeared for more than 3 tests

select AID,count(*) AS Test_No from etest_detail group by AID having count(AID) > 3;

AID TEST_NO

----- ----------

A002 4

4. Calculate applicant's average score across all test they have appeared in

select ETID,AID,avg(score) AS Average_Score from etest_detail group by AID,ETID;

AID AVERAGE_SCORE

----- -------------

A004 90

A003 78.5

A001 78.5

A002 80.5

5. Display number of applicants by city

select city,count(City) AS Applicant_no from applicant group by City;

CITY APPLICANT_NO

---------- ----------

Ahmedabad 2

Mumbai 1

Surat 1

Rishikesh 1

6. Display ETID and Average score where average score is more than 50

select ETID,avg(Score) AS Average_score from etest_detail group by ETID having avg(score) > 50 order by(ETID) DESC;

ETID AVERAGE_SCORE

---- -------------

E001 80

E002 86

E003 86.3333333

E004 67.5

7. Count date wise total entrance test to be held

select Etest_date,Count(Etest_date) AS Etest_date from Etest_detail23 group by Etest_date;

ETEST_DAT ETEST_DATE

--------- ----------

16-JUN-22 4

17-JUN-22 4

18-JUN-22 4

15-JUN-22 4

I. Use the following tables and solve below given queries.

Distributor (Dno, DName, City, Phone)

Item (Item_No, Item_Name, Price, Weight)

Dist_Item (Dno, Item_No, Qty, Date)

create table Distributor(Dno varchar(4) primary key, Dname varchar(20), city varchar(10), Phone number(10))

insert into distributor values('D001','Sanket','Ahmedabad',7894586978);

insert into distributor values('D002','Harry Potter','Hogsward',9417554357);

insert into distributor values('D003','Ron Weasley','Hogsmeade',4082279747);

insert into distributor values('D004','Tea Post','LJ Campus',6356660734);

insert into distributor values('D005','Salim Babu','Sarkhej',9427953649);

insert into distributor values('D006','Harsh','Sarkhej',9427953679);

DNO DNAME CITY PHONE

---- -------------------- ---------- ----------

D002 Harry Potter Hogsward 9417554357

D003 Ron Weasley Hogsmeade 4082279747

D004 Tea Post LJ Campus 6356660734

D005 Salim Babu Sarkhej 9427953649

D001 Sanket Ahmedabad 7894586978

Item (Item_No, Item_Name, Price, Weight)

```
create table item(item_no varchar(4) primary key,item_name varchar(45),price
number(5),weight varchar(10));

insert into item values('i001','cold cofee',20,'200gm');

insert into item values('i002','hot coffe ',40,'50ml');

insert into item values('i003','burger',100,'500gm');

insert into item values('i004','nachos',200,'100gm');

insert into item values('i005','garlic bread',250,'150gm');
```

```
SQL> select * from item;

ITEM ITEM_NAME                                PRICE WEIGHT
---- ---------------------------------------- ---------- ----------
i001 cold cofee                                    20 200gm
i002 hot coffe                                     40 50ml
i003 burger                                       100 500gm
i004 nachos                                       200 100gm
i005 garlic bread                                 250 150gm
```

Dist_Item (Dno, Item_No, Qty, Date)

```
create table Dist_Item  (
                D_NO varchar(4) references Distributor(Dno),
                ItemNo varchar(4) references item(item_no),
                qty number(4),
                dist_date date,
                primary key(D_no,ItemNO)
);

insert into  Dist_Item
values('D001','i001',200,'24-nov-2001');
insert into  Dist_Item values('D001','i002',200,'25-nov-2001');
insert into  Dist_Item values('D001','i003',200,'26-nov-2001');
insert into  Dist_Item values('D001','i004',200,'27-nov-2001');
insert into  Dist_Item values('D001','i005',200,'28-nov-2001');
insert into  Dist_Item values('D002','i001',200,'24-nov-2001');
insert into  Dist_Item values('D002','i002',200,'25-nov-2001');
insert into  Dist_Item values('D002','i003',200,'26-nov-2001');
insert into  Dist_Item values('D002','i004',200,'27-nov-2001');
insert into  Dist_Item values('D002','i005',200,'28-nov-2001');
insert into  Dist_Item values('D003','i001',200,'24-nov-2001');
insert into  Dist_Item values('D003','i002',200,'25-nov-2001');
insert into  Dist_Item values('D003','i003',200,'26-nov-2001');
insert into  Dist_Item values('D003','i004',200,'27-nov-2001');
insert into  Dist_Item values('D003','i005',200,'28-nov-2001');
insert into  Dist_Item values('D004','i001',200,'24-nov-2001');
insert into  Dist_Item values('D004','i002',200,'25-nov-2001');
```

```
insert into  Dist_Item values('D004','i003',200,'26-nov-2001');

insert into  Dist_Item values('D004','i004',200,'27-nov-2001');

insert into  Dist_Item values('D004','i005',200,'28-nov-2001');

insert into  Dist_Item values('D005','i001',200,'24-nov-2001');

insert into  Dist_Item values('D005','i002',200,'25-nov-2001');

insert into  Dist_Item values('D005','i003',200,'26-nov-2001');

insert into  Dist_Item values('D005','i004',200,'27-nov-2001');

insert into  Dist_Item values('D005','i005',200,'28-nov-2001');D_NO ITEM       QTY
DIST_DATE

---- ---- ---------- ---------

D001 i001       200 24-NOV-01

D001 i002       200 25-NOV-01

D001 i003       200 26-NOV-01

D001 i004       200 27-NOV-01

D001 i005       200 28-NOV-01

D002 i001       200 24-NOV-01

D002 i002       200 25-NOV-01

D002 i003       200 26-NOV-01

D002 i004       200 27-NOV-01

D002 i005       200 28-NOV-01

D003 i001       200 24-NOV-01


D_NO ITEM       QTY DIST_DATE

---- ---- ---------- ---------

D003 i002       200 25-NOV-01

D003 i003       200 26-NOV-01

D003 i004       200 27-NOV-01

D003 i005       200 28-NOV-01

D004 i001       200 24-NOV-01

D004 i002       200 25-NOV-01
```

D004 i003 200 26-NOV-01

D004 i004 200 27-NOV-01

D004 i005 200 28-NOV-01

D005 i001 200 24-NOV-01

D005 i002 200 25-NOV-01

D_NO ITEM QTY DIST_DATE

---- ---- ---------- ---------

D005 i003 200 26-NOV-01

D005 i004 200 27-NOV-01

D005 i005 200 28-NOV-01

1. Display city wise total number of distributors

select city,count(*) AS Distributor_NO from Distributor group by City;

CITY DISTRIBUTOR_NO

---------- --------------

Hogsmeade 1

LJ Campus 1

Ahmedabad 1

Sarkhej 2

Hogsward 1

2. List distributors' no by who distributed more than 50 items in month of July

select

```
select D_no,to_char(dist_date,'Mon'), sum(qty) AS QTY from dist_item
group by D_no,dist_date having to_char(dist_date,'Mon') like 'Jul' and sum(qty) > 50 order by(d_no) ASC;
```

```
D_NO TO_CHAR(DIST       QTY
---- ------------ ----------
D001 Jul             200
D002 Jul             200
D003 Jul             200
D004 Jul             200
D005 Jul             200
```

3. List Item_No with more than 800 Qty delivered

```
select ItemNo,sum(qty) from dist_item group by Itemno,qty having sum(qty) > 800 order by(Itemno) asc;
```

```
ITEM   SUM(QTY)
---- ----------
i001      1000
i002      1000
i003      1000
i004      1000
i005      1000
```

4. List Dno who delivered more than 50 items for each month

```
select D_no,to_char(dist_date,'Mon'), sum(qty) AS QTY from dist_item
```

group by D_no,to_char(dist_date,'Mon') having sum(qty) > 50 order by(d_no) ASC;

D_NO TO_CHAR(DIST QTY

---- ---------- ----------

D001 Jul 200

D001 Nov 800

D002 Jul 200

D002 Nov 800

D003 Jul 200

D003 Nov 800

D004 Jul 200

D004 Nov 800

D005 Jul 200

D005 Nov 800

5. Display item details in descending order of price and ascending order of weight

select Item_No, Item_Name, Price, Weight from item order by price desc, weight;

ITEM ITEM_NAME PRICE WEIGHT

---- ---------- ---------- ----------

i005 garlic bread 250 150gm

i004 nachos 200 100gm

i003 burger 100 500gm

i002 hot coffe 40 50ml

i001 cold cofee 20 200gm

6. Show all distributors in alphabetical order of City and DName

select * from distributor order by city, dname;

DNO DNAME CITY PHONE

---- -------------------- ---------- ----------

D001 Sanket Ahmedabad 7894586978

D003 Ron Weasley Hogsmeade 4082279747

D002 Harry Potter Hogsward 9417554357

D004 Tea Post LJ Campus 6356660734

D006 Harsh Sarkhej 9427953679

D005 Salim Babu Sarkhej 9427953649

7. Calculate average quantity of items distributed on each day.

select D_no,avg(qty),to_char(dist_date,'dd/mm/yyyy') from dist_item group by d_no,to_char(dist_date,'dd/mm/yyyy'),qty order by(D_no);

D_NO AVG(QTY) TO_CHAR(DI

---- ---------- ----------

D001 200 24/11/2001

D001 200 25/07/2001

D001 200 26/11/2001

D001 200 27/11/2001

D001 200 28/11/2001

D002 200 24/11/2001

D002 200 25/07/2001

D002 200 26/11/2001

D002 200 27/11/2001

D002 200 28/11/2001

D003 200 24/11/2001

D_NO AVG(QTY) TO_CHAR(DI

---- ---------- ----------

D003 200 25/07/2001

D003 200 26/11/2001

D003 200 27/11/2001

D003 200 28/11/2001

D004 200 24/11/2001

D004 200 25/07/2001

D004 200 26/11/2001

D004 200 27/11/2001

D004 200 28/11/2001

D005 200 24/11/2001

D005 200 25/07/2001

D_NO AVG(QTY) TO_CHAR(DI

---- ---------- ----------

D005 200 26/11/2001

D005 200 27/11/2001

D005 200 28/11/2001

8. Find the weight wise average price of items.

```
select avg(price) from item where weight = '150gm';
```

```
AVG(PRICE)
----------
       300
```

DAY-7

create table student (rollno varchar(4) primary key,fname varchar(20),lname varchar(20),dob date);

insert into student values('R001','Manav','Patel','09-feb-2001');

insert into student values('R002','Sanket','Raval','26-apr-1944');

insert into student values('R003','Harsh','Patel','09-nov-2001');

insert into student values('R004','simmi','Virani','25-may-2005');

insert into student values('R005','Amisha','Raval','09-nov-2001');

ROLL	FNAME	LNAME	DOB
R001	Manav	Patel	09-FEB-01
R002	Sanket	Raval	26-APR-44
R003	Harsh	Patel	09-NOV-01
R004	simmi	Virani	25-MAY-05
R005	Amisha	Raval	09-NOV-01

COURSE (rollno,courseno,fees, coursename, max_marks, pass_marks)

create table course (roll_no varchar(4) references student(rollno),course_no varchar(5),fees number(4),course_name varchar(10), max_marks number(5), pass_marks number(5));

insert into course values('R001','C001',2500,'c',150,75);

insert into course values('R001','C002',2200,'html',200,150);

insert into course values('R002','C003',1900,'c++',70,59);

insert into course values('R002','C001',1200,'c',150,89);

insert into course values('R003','C004',8000,'java',120,100);

```
insert into course values('R003','C002',8000,'html',200,200);
insert into course values('R004','C001',8000,'c',150,59);
insert into course values('R004','C004',8000,'java',120,105);
insert into course values('R004','C003',8000,'java',120,105);
```

```
ROLL COURS       FEES COURSE_NAM  MAX_MARKS PASS_MARKS
---- ----- ---------- ---------- ---------- ----------
R001 C001        2500 c              150        75
R001 C002        2200 html           200        150
R002 C003        1900 c++            70         59
R002 C001        1200 c              150        89
R003 C004        8000 java           120        100
R003 C002        8000 html           200        200
R004 C001        8000 c              150        59
R004 C004        8000 java           120        105
```

1. Find all students whose marks is higher than the average marks of the students in their course.(using corelate sub query)

```
        select * from student where student.rollno IN (
        select roll_no from course,
        (select course_no,avg(pass_marks) rt from course group by course_no) ep
    where course.pass_marks > ep.rt and course.course_no = ep.course_no);
```

```
ROLL FNAME              LNAME              DOB
---- -------------------- -------------------- ---------
R001 Manav               Patel              09-FEB-01
R002 Sanket              Raval              26-APR-44
R003 Harsh               Patel              09-NOV-01
R004 simmi               Virani             25-MAY-05
```

2. List the student detail whose marks is same as ‘simmi’.

```
select * from course where pass_marks in (select pass_marks from course where roll_no =
(select rollno from student st where fname like'simmi') );
```

```
ROLL COURS       FEES COURSE_NAM  MAX_MARKS PASS_MARKS
---- ----- ---------- ---------- ---------- ----------
R004 C001        8000 c               150        59
R002 C003        1900 c++              70        59
R004 C004        8000 java            120        105
```

3. Display the name of students who have enrolled for ‘html’ course.

```
select * from student where rollno IN (
select roll_no from course where course_name like'html');
```

```
ROLL FNAME               LNAME               DOB
---- -------------------- -------------------- ---------
R001 Manav               Patel               09-FEB-01
R003 Harsh               Patel               09-NOV-01
```

4. Display details of students who have enrolled in courses

```
select * from student where rollno IN
(select distinct roll_no from course);
```

ROLL FNAME LNAME DOB

---- -------------------- -------------------- ---------

R001 Manav Patel 09-FEB-01

R002 Sanket Raval 26-APR-44

R003 Harsh Patel 09-NOV-01

R004 simmi Virani 25-MAY-05

5. Update max_marks =80 where courseno is C004

update course set max_marks = 80 where course_no = 'C004';

6. Delete those rows where pass_marks is less than 40.

delete from course where pass_marks < 40;

7. Give the name of the course in which maximum number of students are enrolled

select max(cnt) from

(select count(*) cnt,course_no from course group by course_no);

MAX(CNT)

3

8. Display the first name and last name of student who have enrolled for ‘c’ course.

select fname,lname from student where rollno IN(

select roll_no from course where course_name like 'c');

FNAME LNAME

---------- ----------

Manav Patel

Sanket Raval

simmi Virani

Employee (emp_no, fname, lname, dob,address)

create table emp1 (emp_no varchar(4) primary key,fname varchar(20),lname varchar(20),dob date,add1 varchar(20));

insert into emp1 values('E001','Harry','Potter','2-feb-2001','Hogsward');

insert into emp1 values('E002','Ron','Wasly','4-mar-2001','surat');

insert into emp1 values('E003','Ben','parker','2-jun-2008','New York');

insert into emp1 values('E004','Nobita','Nobi','3-feb-2008','Japan');

insert into emp1 values('E005','Sizuka','Nobi','8-jul-2001','japan');

insert into emp1 values('E006','Doremon','Nobi','9-dec-2001','japan');

EMP_ FNAME LNAME DOB ADD1

---- ---------- ---------- --------- ----------

E001 Harry Potter 02-FEB-01 Hogsward

E002 Ron Wasly 04-MAR-01 surat

E003 Ben parker 02-JUN-08 New York

E004 Nobita Nobi 03-FEB-08 Japan

E005 Sizuka Nobi 08-JUL-01 japan

Company(comp_id,emp_no,company_name,salary,department,designation)

```
create table company
(comp_id varchar(5), emp_no varchar(4) references emp1(emp_no),company_name
varchar(10),D_no varchar(5),
salary number(5),department varchar(10),designation varchar(15),primary
key(comp_Id,emp_no));

insert into company values('C001','E001','intas',2000,'Management','Manager');
insert into company values('C002','E002','infosys',90000,'HR','Director');
insert into company values('C003','E003','Wipro',90000,'Management','Manager');
insert into company values('C004','E004','Azilen',50000,'Finance','Accountant');
insert into company values('C005','E005','Wipro',80000,'marketing','salesman');
```

```
COMP_ EMP_ COMPANY_NA     SALARY DEPARTMENT DESIGNATION
----- ---- ---------- ---------- ---------- ---------------
C001  E001 intas           2000 Management Manager
C002  E002 infosys         90000 HR         Director
C003  E003 Wipro           90000 Management Manager
C004  E004 Azilen          50000 Finance    Accountant
C005  E005 Wipro           80000 marketing  salesman
```

Customer(cust_no,fname,lname,address)

```
create table customer(cust_n0 varchar(4) primary key,fname varchar(10),lname
varchar(10),add1 varchar(15));

insert into customer values('C001','ben','parker','new york');
insert into customer values('C002','Manav','Kakani','Ahmedabad');
```

```
insert into customer values('C003','Harsh','Patel','Surat');
insert into customer values('C004','Sanket','Raval','Bihar');
insert into customer values('C005','Vrajesh','Limbachiya','Godhra');
```

```
CUST FNAME      LNAME      ADD1
---- ---------- ---------- ---------------
C001 ben        parker     new york
C002 Manav      Kakani     Ahmedabad
C003 Harsh      Patel      Surat
C004 Sanket     Raval      BIhar
C005 Vrajesh    Limbachiya Godhra
```

1. **Find all employees whose salary is higher than the average salary of the employees in their departments.**

```
select emp_no,salary from company group by emp_no,salary having salary >
(select avg(salary) from company);
```

```
EMP_     SALARY
---- ----------
E005     80000
E002     90000
E003     90000
```

2. **List the employee detail whose salary is same as “Jay”**

```
select * from company where salary in (select salary from company where emp_no =
 (select Emp_no from emp1 where fname like'Ben') );
```

```
COMP_ EMP_ COMPANY_NA     SALARY DEPARTMENT DESIGNATION
----- ---- ---------- ---------- ---------- ---------------
C003  E003 Wipro          90000 Management Manager
C002  E002 infosys        90000 HR         Director
```

3. Display the name of employees who work in ‘mrkt’ department.

```
select fname from emp1 where emp_no =
(select Emp_no from company where department like 'marketing');
```

```
FNAME
--------
Sizuka
```

4. Display details of employees not belong to any department.

```
 select * from emp1 where emp_no NOT IN
 (select distinct emp_no from company);
```

```
EMP_ FNAME              LNAME              DOB       ADD1
---- -------------------- -------------------- --------- -----
E006 Doremon            Nobi               09-DEC-01 japan
```

5. Find out all the customers having same name as the employees.
(Using multi column subquery).

```
 select fname from emp1 where (select fname from customer) =
 (select fname from emp1);
```

6. Delete those rows where department=’HR’ ;

delete from company where department like'HR';

1 row deleted.

7. Update salary*0.10 where emp_no=105.

update company set salary=salary*0.10 where emp_no = 'E001';

II.

PRODUCTS (Prod_ID, Prod_Name, Supplier_ID, Cat_ID, Unit, Price)

Create table products(prod_id varchar(4) primary key,prod_name varchar(10),supplier_id varchar(10),

cat_id varchar(5), unit varchar(5), price number(5));

insert into products values('P001','Butter','A1','C01','KG',100);

insert into products values('P002','Milk','M1','C05','ML',25);

insert into products values('P003','Paneer','A1','C02','KG',200);

insert into products values('P004','lassi','CV2','C06','ML',50);

insert into products values('P005','ButterMilk','CV1','C03','Ltr',70);

insert into products values('P006','Cookie','CV1','C03','KG',70);

PROD PROD_NAME SUPPLIER_I CAT_I UNIT PRICE

---- ---------- ---------- ----- ----- ----------

P001 Butter A1 C01 KG 100

P003 Paneer A1 C02 KG 200

P004 lassi CV2 C06 ML 50

P005 ButterMilk CV1 C03 Ltr 70

P002 Milk M1 C05 ML 25

ORDER_DETAILS (OrderDetail_ID, Order_ID, Prod_Id, Quantity)

```
create table order_details(od_id varchar(5) primary key, o_id varchar(5),
                prod_id varchar(10) references products(prod_id), qty number(5));

insert into order_details values('OD01','O001','P001',50);
insert into order_details values('OD02','O005','P002',200);
insert into order_details values('OD03','O003','P003',150);
insert into order_details values('OD04','O002','P004',100);
insert into order_details values('OD05','O004','P005',300);
insert into order_details values('OD06','O001','P002',50);
```

OD_ID	O_ID	PROD_ID	QTY
OD01	O001	P001	50
OD02	O005	P002	200
OD03	O003	P003	150
OD04	O002	P004	100
OD05	O004	P005	300

1. **Lists the ProductName if ANY records in the Order_Details table has Quantity equal to 50**

```
select prod_name from products where prod_id IN(
select prod_id from order_details where qty = 50);
```

```
PROD_NAME
----------
Butter
Milk
```

2. **lists the ProductName if ALL the records in the OrderDetails table has Quantity more than 150**

```
 select prod_name from products where prod_id IN(
 select prod_id from order_details where qty > 150);
```

```
PROD_NAME
----------
Milk
ButterMilk
```

3. **Display Products which are not ordered**

```
select prod_name from products where prod_id NOT IN(
select distinct prod_id from order_details);
```

```
PROD_NAME
----------
```

Cookie

4. List the products which is ordered for more than 200 Quantity

```
select prod_name from products where prod_id IN(
select prod_id from order_details where qty > 200);
```

```
PROD_NAME
----------
ButterMilk
```

5. List the products whose Unit is Kg or Lt (Use IN operator)

```
select prod_name,unit from products where unit IN('KG','Ltr');
```

```
PROD_NAME  UNIT
---------- -----
Butter     KG
Paneer     KG
ButterMilk Ltr
Cookie     KG
```

DAY-8

I. Use following tables and solve given queries below it.

APPLICANT (AID, A_Name, City, B_Date)

select * from APPLICANT;

A_ID A_NAME CITY B_DATE

----- ---------- ---------- ---------

A001 Sanket Ahmedabad 14-JUL-02

A002 Priya Surat 21-FEB-01

A003 Vanita Mumbai 07-JUL-03

A004 Amisha Rishikesh 11-NOV-01

A005 Krishna Ahmedabad 07-SEP-01

ENTRANCE_TEST (ET_ID, ET_Name, Max_Score)

select * from entrance_test;

ET_I ET_NAME MAX_SCORE

---- --------------- ----------

E001 Programming 100

E002 Maths 100

E003 Web Designing 70

E004 Web DEvloping 80

ETEST_DETAILS (AID, ETID, ETest_Date, Score)

AID ETID ETEST_DAT SCORE

----- ---- --------- ----------

A001 E004 15-JUN-22 70

A002 E002 18-JUN-22 85

A004 E003 19-JUN-22 90

A003 E001 20-JUN-22 70

A002 E004 28-JUN-22 65

A003 E003 21-JUN-22 87

A001 E002 21-JUN-22 87

A002 E001 21-JUN-22 90

A002 E003 20-JUN-22 82

A001 E003 15-JUN-22 95

A001 E001 15-JUN-22 95

1. Display Entrance Test ID (ETID) Wise highest marks scored by any applicant

select etid,max(score) from etest_detail group by etid;

ETID MAX(SCORE)

---- ----------

E002 87

E004 70

E001 95

E003 95

2. Count ETID wise total number of applicants appeared for the test

select ETID,count(AID) from etest_detail group by ETID order by ETID;

ETID COUNT(AID)

---- ----------

E001 3

E002 2

E003 4

E004 2

3. Find the minimum number of applicants in the entrance test.

```
select min(cnt) from (select ETID,count(AID) cnt from etest_detail group by ETID order by
ETID);
```

```
 MIN(CNT)
----------
         2
```

4. Count city wise number of applicants registered

```
select City,count(A_ID) from applicant group by city;
```

```
CITY       COUNT(A_ID)
---------- -----------
Ahmedabad            2
Mumbai               1
Surat                1
Rishikesh            1
```

5. Display all the entrance test details for which the applicant “Sanket” appeared

```
select * from etest_detail where Aid = (
select A_id from applicant where A_name like 'Sanket') order by ETID;
```

```
AID   ETID ETEST_DAT      SCORE
----- ---- --------- ----------
A001  E001 15-JUN-22         95
A001  E002 21-JUN-22         87
```

A001 E003 15-JUN-22 95

A001 E004 15-JUN-22 70

II. Use the following tables and solve below given queries

Distributor (Dno, DName, City, Phone)

select * from Distributor;

DNO DNAME CITY PHONE

---- -------------------- ---------- ----------

D002 Harry Potter Hogsward 9417554357

D003 Ron Weasley Hogsmeade 4082279747

D004 Tea Post LJ Campus 6356660734

D005 Salim Babu Sarkhej 9427953649

D001 Sanket Ahmedabad 7894586978

D006 Harsh Sarkhej 9427953679

Item (Item_No, Item_Name, Price, Weight)

select * from Item;

ITEM ITEM_NAME PRICE WEIGHT

---- ---------- ---------- ----------

i001 cold cofee 20 200gm

i002 hot coffe 40 50ml

i003 burger 100 500gm

i004 nachos 200 100gm

i005 garlic bread 250 150gm

Dist_Item (Dno, Item_No, Qty, Date)

select * from Dist_Item;

D_NO ITEM QTY DIST_DATE

---- ---- ---------- ---------

D001 i001 200 24-NOV-01

D001 i002 200 25-JUL-01

D001 i003 200 26-NOV-01

D001 i004 200 27-NOV-01

D001 i005 200 28-NOV-01

D002 i001 200 24-NOV-01

D002 i002 200 25-JUL-01

D002 i003 200 26-NOV-01

D002 i004 200 27-NOV-01

D002 i005 200 28-NOV-01

D003 i001 200 24-NOV-01

D_NO ITEM QTY DIST_DATE

---- ---- ---------- ---------

D003 i002 200 25-JUL-01

D003 i003 200 26-NOV-01

D003 i004 200 27-NOV-01

D003 i005 200 28-NOV-01

D004 i001 200 24-NOV-01

D004 i002 200 25-JUL-01

D004 i003 200 26-NOV-01

D004 i004 200 27-NOV-01

D004 i005 200 28-NOV-01

D005 i001 200 24-NOV-01

D005 i002 200 25-JUL-01

D_NO ITEM QTY DIST_DATE

---- ---- ---------- ---------

D005 i003 200 26-NOV-01

D005 i004 200 27-NOV-01

D005 i005 200 28-NOV-01

1. Display all the distributor's name who supplies Item_No 5

```
select dname from Distributor where DNO IN(
select D_no from dist_item where itemno = 'i005');
```

DNAME

Harry Potter

Ron Weasley

Tea Post

Salim Babu

Sanket

2. Display the item which is distributed maximum time

```
select ITEM_NAME from item where item_no IN(
select itemno,sum(qty) from dist_item group by itemno);
```

3. Display all the items that are distributed by the distributor "Ron Weasley"

```
select * from dist_item where D_no = (
```

select dno from Distributor where dname like'Ron Weasley');

D_NO ITEM QTY DIST_DATE

---- ---- ---------- ---------

D003 i001 200 24-NOV-01

D003 i002 200 25-JUL-01

D003 i003 200 26-NOV-01

D003 i004 200 27-NOV-01

D003 i005 200 28-NOV-01

4. Display the Item_Name and Quantity that are received in month of July in 2021

select D_no,to_char(dist_date,'Mon'), sum(qty) AS QTY from dist_item

group by D_no,dist_date having to_char(dist_date,'Mon') like 'Jul';

D_NO TO_CHAR(DIST QTY

---- ------------ ----------

D003 Jul 200

D002 Jul 200

D004 Jul 200

D005 Jul 200

D001 Jul 200

5. Display all the items whose price is less than 1000 and received Qty more then 10

select * from dist_item where itemno IN(

select item_no,price from item group by item_no,price having price < 1000) and qty > 10;

DAY-9

1. **create table branch(bname varchar(10) primary key,city varchar(10));**

insert into branch values('Nikol','Ahmedabad');

insert into branch values('yogichowk','Surat');

insert into branch values('Verli','Mumbai');

insert into branch values('Sarkhej','Ahmedabad');

insert into branch values('Varachha','surat');

BNAME CITY

---------- ----------

Nikol Ahmedabad

yogichowk Surat

Verli Mumbai

Sarkhej Ahmedabad

Varachha surat

2. **create table customer(cust_no varchar(5) primary key,Cname varchar(10),city varchar(10));**

insert into customer values('C001','Manav','Ahmedabad');

insert into customer values('C002','Harry','Surat');

insert into customer values('C003','Ron','Ahmedabad');

insert into customer values('C004','Priya','Mumbai');

insert into customer values('C005','Eva','Mumbai');

CUST_ CNAME CITY

----- ---------- ----------

C001 Manav Ahmedabad

C002 Harry Surat

C003 Ron Ahmedabad

C004 Priya Mumbai

C005 Eva Mumbai

3. **create table deposite(Accno varchar(5) primary key,cust_no varchar(5) references customer(cust_no),Bname varchar(10) references branch(bname),amount number(6),Adate date);**

insert into deposite values('AC01','C001','Nikol',50000,'22-jul-2022');

insert into deposite values('AC02','C002','Varachha',10000,'21-feb-2022');

insert into deposite values('AC03','C003','Verli',70000,'21-jan-2022');

insert into deposite values('AC04','C004','Nikol',25000,'09-feb-2021');

insert into deposite values('AC05','C005','yogichowk',10000,'15-feb-2021');

ACCNO CUST_ BNAME AMOUNT ADATE

----- ----- ---------- ---------- ---------

AC02 C002 Varachha 10000 21-FEB-22

AC03 C003 Verli 70000 21-JAN-22

AC04 C004 Nikol 25000 09-FEB-21

AC05 C005 yogichowk 10000 15-FEB-21

4. **create table borrow(LoanNo varchar(5) primary key,cust_no varchar(5) references customer(cust_no),Bname varchar(10) references branch(bname),amount number(10));**

```
insert into borrow values('L001','C001','Nikol',30000);

insert into borrow values('L002','C002','Varachha',25000);

insert into borrow values('L003','C003','Verli',15000);
```

```
LOANN CUST_ BNAME          AMOUNT

----- ----- ---------- ----------

L001  C001  Nikol          30000

L002  C002  Varachha       25000

L003  C003  Verli          15000
```

1. Get the details of the customers ‘Ron’.

```
select Accno,loanno,Cname,deposite1.amount AS D_amount,borrow.amount AS
L_amount,city

from customer join deposite1

on

customer.cust_no = deposite1.cust_no

join borrow

on

customer.cust_no = borrow.cust_no

where cname like'Ron';
```

```
ACCNO LOANN CNAME       D_AMOUNT   L_AMOUNT CITY

----- ----- ---------- ---------- ---------- ----------

AC03  L003  Ron            70000      15000 Ahmedabad
```

2. **Give name of customer who are borrowers and depositors and having living city Ahmedabad**

```
select Accno,loanno,Cname,deposite1.amount AS D_amount,borrow.amount AS
L_amount,city
from customer join deposite1
on
customer.cust_no = deposite1.cust_no
join borrow
on
customer.cust_no = borrow.cust_no
where city like'Ahmedabad';
```

```
ACCNO LOANN CNAME       D_AMOUNT   L_AMOUNT CITY
----- ----- ---------- ---------- ---------- ----------
AC01  L001  Manav         50000      30000 Ahmedabad
AC03  L003  Ron           70000      15000 Ahmedabad
```

3. **Give city as their city name of customers having same living branch**

```
select cname,city,bname from customer join deposite1 on
customer.cust_no = deposite1.cust_no
where bname like'Nikol';
```

```
CNAME      CITY       BNAME
---------- ---------- ----------
Manav      Ahmedabad  Nikol
Priya      Mumbai     Nikol
```

desc emp1;

```
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
EMP_NO                                    NOT NULL VARCHAR2(4)
FNAME                                              VARCHAR2(20)
LNAME                                              VARCHAR2(20)
DOB                                                DATE
ADD1                                               VARCHAR2(20)
```

```
insert into emp1 values('E007','jiyan','Takamora','28-feb-2016','japan');
```

desc company;

```
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
COMP_ID                                   NOT NULL VARCHAR2(5)
EMP_NO                                    NOT NULL VARCHAR2(4)
COMPANY_NAME                                       VARCHAR2(10)
SALARY                                             NUMBER(5)
DEPARTMENT                                         VARCHAR2(10)
DESIGNATION                                        VARCHAR2(15)
```

```
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
CUST_NO                                   NOT NULL VARCHAR2(5)
CNAME                                              VARCHAR2(10)
CITY                                               VARCHAR2(10)
```

create table company1

(comp_id varchar(5), emp_no varchar(4) references emp1(emp_no),company_name varchar(10),D_no varchar(5),

salary number(5),d_loc varchar(10),Hire_date date,department varchar(10),designation varchar(15),primary key(comp_Id,emp_no));

insert into company1 values('C001','E001','intas','D001',2000,'Ahmedabad','26-jul-2020','Management','Manager');

insert into company1 values('C002','E002','infosys','D002',90000,'Surat','27-jul-2020','HR','Director');

insert into company1 values('C003','E003','Wipro','D003',90000,'New York','28-jul-2020','Management','Manager');

insert into company1 values('C004','E004','Azilen','D004',50000,'Wembaly','29-jul-2020','Finance','Accountant');

insert into company1 values('C006','E006','Azilen','D004',50000,'Wembaly','02-jul-2020','Finance','Clerk');

insert into company1 values('C005','E005','Wipro','D005',80000,'Ahmedabad','30-jul-2020','marketing','salesman');

insert into company1 values('C007','E007','Azilen','D004',50000,'Wembaly','02-jul-2020','Finance','supervisior');

4. Write a query to display the last name, department number, and department name for all employees

select fname,lname,d_no,department from emp1 join company1

on

emp1.emp_no = company1.emp_no;

FNAME LNAME D_NO DEPARTMENT

---------- ---------- ----- ----------

Harry Potter D001 Management

Ron Wasly D002 HR

Ben parker D003 Management

Nobita Nobi D004 Finance

Sizuka Nobi D005 marketing

5. Create a unique listing of all jobs that are in department D004. Include the location of the department in the output

select fname,lname,comp_id,department,designation from emp1 join company1

on emp1.emp_no = company1.emp_no where d_no like'D004';

FNAME LNAME COMP_ DEPARTMENT DESIGNATION

---------- ---------- ----- ---------- ----------

Nobita Nobi C006 Finance Clerk

Nobita Nobi C004 Finance Accountant

jiyan Takamora C007 Finance supervisior

6. Write a query to display the employee's name, department number, and department name for all employees who work in NEW YORK

select fname,lname,comp_id,d_no,department,designation,d_loc from emp1 join company1

on emp1.emp_no = company1.emp_no where d_loc like'New York';

FNAME LNAME COMP_ D_NO DEPARTMENT DESIGNATION D_LOC

---------- ---------- ----- ----- ---------- ---------- ----------

Ben parker C003 D003 Management Manager New Yor

7. Display the employee's last name and employee number along with their manager's last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#, respectively

8. Create a query to display the name and hire date of any employee hired after employee 'Harry'

```
select emp_no,Hire_date from company1 where hire_date > (select hire_date from
company1 where emp_no like'E001');
```

```
EMP_ HIRE_DATE
---- ---------
E002 27-JUL-20
E003 28-JUL-20
E004 29-JUL-20
E005 30-JUL-20
```

DAY-10

Worker (Id, Name, Wages_Per_Hr)

Job (Job_Id, Job_Type)

Assigned (W_Id, J_Id, Start_Date, Status, Total_Hrs)

```
create table Worker (w_id varchar(4) primary key,name varchar(10),wph  number(10));

insert into worker values('W001','sanket',500);
insert into worker values('W002','harsh',300);
insert into worker values('W003','manav',200);
insert into worker values('W004','harry',500);
insert into worker values('W005','coffee',800);
insert into worker values('W006','Damar',800);

create table job1(job_id varchar(4) primary key,job_type varchar(10));

insert into job1 values('J001','manager');
insert into job1 values('J002','clerk');

create table assigned(w_id varchar(4) references worker(w_id),j_id varchar(4) references
job1(job_id),start_date date,status varchar(20),total_hrs number(10));

insert into assigned values('W001','J001','11-feb-2020','Done',10);
insert into assigned values('W002','J001','12-feb-2020','Done',10);
insert into assigned values('W003','J001','13-feb-2020','Pending',12);
insert into assigned values('W004','J002','14-feb-2020','in progress',7);
insert into assigned values('W005','J002','15-feb-2020','Pending',6);
insert into assigned values('W006','J002','15-feb-2020','Done',6);
```

1. Display all the workers' Id assigned for Job 1 and 2 both (J_Id =1 and J_Id=2)

```
select * from worker join assigned on
worker.w_id = assigned.w_id where j_id like'J001'
Union
select * from worker join assigned on
worker.w_id = assigned.w_id where j_id like'J002';
```

QCSJ NAME	WPH	QCSJ	J_ID	START_DAT	STATUS	TOTAL_HRS
W001 sanket	500	W001	J001	11-FEB-20	Done	10
W002 harsh	300	W002	J001	12-FEB-20	Done	10
W003 manav	200	W003	J001	13-FEB-20	Pending	12
W004 harry	500	W004	J002	14-FEB-20	in progress	7
W005 coffee	800	W005	J002	15-FEB-20	Pending	6

2. List the workers who are working for J_Id =1 and not for J_Id = 2

```
select * from worker join assigned on
worker.w_id = assigned.w_id where j_id NOT IN(select j_id from assigned where j_id
like'J002');
```

QCSJ NAME	WPH	QCSJ	J_ID	START_DAT	STATUS	TOTAL_HRS
W001 sanket	500	W001	J001	11-FEB-20	Done	10
W002 harsh	300	W002	J001	12-FEB-20	Done	10
W003 manav	200	W003	J001	13-FEB-20	Pending	12

3. **Display the workers Id who are working for more than 6 hours and status is pending. And Union them all with the worker's Id who are working for less than 6 hours and status is completed**

```
select * from worker join assigned on
worker.w_id = assigned.w_id where total_hrs > 6 and status like'Pending'
union
select * from worker join assigned on
worker.w_id = assigned.w_id where total_hrs <= 6 and status like'Done'
```

QCSJ	NAME	WPH	QCSJ	J_ID	START_DAT	STATUS	TOTAL_HRS
W003	manav	200	W003	J001	13-FEB-20	Pending	12
W006	Damar	800	W006	J002	15-FEB-20	Done	6

III. Use following tables to solve below given queries.

FYRankers (Enrol_No, Name, SPI)

SYRankers (Enrol_No, Name, SPI)

```
create table FY(Enrol_no varchar(10) primary key, name varchar(10), spi number(5,2));

insert into FY values('FE001','Manav',6.5);
insert into FY values('FE002','Sanket',5.5);
insert into FY values('FE003','Harsh',4.5);
insert into FY values('FE004','Ron',7.9);
insert into FY values('FE005','salim',8);
```

```
create table SY(Enrol_no varchar(10) primary key, name varchar(10), spi number(5,2));

insert into SY values('SE001','Coffee',6.5);
insert into SY values('SE002','Harry',5.5);
insert into SY values('SE003','Damar',4.5);
insert into SY values('SE004','Ron',7.9);
insert into SY values('SE005','salim',3);
```

1. Display the name of the student who is ranker in 'FY' or 'SY'

```
select * from FY where spi = (select max(spi) from FY)
union
select * from SY where spi = (select max(spi) from SY);
```

```
ENROL_NO   NAME          SPI
---------- ---------- ----------
FE004      Ron           7.9
SE004      Ron           7.9
```

2. Display the name of the student who is ranker in 'FY' or 'SY' including duplicate data

3. **Display the name of the students who is ranker in 'FY' or 'SY' and having SPI more than 7 select * from FY where spi > 7 union select * from SY where spi > 7;**

ENROL_NO NAME SPI

---------- ---------- ----------

FE004 Ron 7.9

SE004 Ron 7.9

4. Display the name of the student who is ranker in both FY and SY

```
select * from FY where spi = (select max(spi) from FY)
union
select * from SY where spi = (select max(spi) from SY);
```

ENROL_NO NAME SPI

---------- ---------- ----------

FE004 Ron 7.9

SE004 Ron 7.9

5. Display the name of the student who is ranker in FY but not in SY

```
select * from FY where spi = (select max(spi) from FY)
minus
select * from SY where spi = (select max(spi) from SY);
```

ENROL_NO NAME SPI

---------- ---------- ----------

FE005 salim 8

6. Display the name of the student who is ranker in SY but not in FY

```
select * from SY where spi = (select max(spi) from SY)
minus
select * from FY where spi = (select max(spi) from FY);
```

```
ENROL_NO   NAME            SPI
---------- ---------- ----------
SE004      Ron             7.9
```

DAY-11

create table emp (emp_no varchar(10) primary key,e_name varchar(10),city varchar(10));

insert into emp values('E001','sanket','ahmedabad');

insert into emp values('E002','manav','baroda');

insert into emp values('E003','harsh','surat');

insert into emp values('E004','harry','rajkot');

insert into emp values('E005','coffee','mumbai');

create table dept (dept_no varchar(10) primary key,emp_no varchar(10) references emp(emp_no),dept_name varchar(10));

insert into dept values('D001','E001','marketing');

insert into dept values('D002','E002','HR');

insert into dept values('D003','E003','loan');

insert into dept values('D004','E004','HOD');

insert into dept values('D005','E005','finance');

1. Create a view called Emp_View from Employee table.

create view emp_view as select * from emp;

SQL> select * from emp_view;

EMP_NO E_NAME CITY

---------- ---------- ----------

E001 sanket ahmedabad

E002 manav baroda

E003 harsh surat

E004 harry rajkot

E005 coffee mumbai

2. Renaming the columns of Emp view.

create view emp_view2 AS select emp_no"empno",E_name"Ename",city"Add1" from emp;

empno Ename Add1

---------- ---------- ----------

E001 sanket ahmedabad

E002 manav baroda

E003 harsh surat

E004 harry rajkot

E005 coffee mumbai

3. Select Employee Name where dept_name is ‘Marketing’ or ‘Loan’.

```
select e_name,dept_name from emp join dept on
emp.emp_no = dept.emp_no where dept_name like'marketing'
union
select e_name,dept_name from emp join dept on
emp.emp_no = dept.emp_no where dept_name like'loan';
```

E_NAME DEPT_NAME

---------- ----------

harsh loan

sanket marketing

4. Update Name=’Ron’ where name is ‘harry’.

update emp set e_name = 'Ron' where e_name = 'harry';

EMP_NO E_NAME CITY

---------- ---------- ----------

E001 sanket ahmedabad

E002 manav baroda

E003 harsh surat

E004 Ron rajkot

E005 coffee mumbai

5. Delete a record where name is ‘coffee’.

DELETE FROM emp WHERE e_name like 'coffee';

EMP_NO E_NAME CITY

---------- ---------- ----------

E001 sanket ahmedabad

E002 manav baroda

E003 harsh surat

E004 Ron rajkot

6. Remove a view Emp from database.

```
drop view emp_view1;
```

View dropped.

Branch (bno, bname)

Address (addno, bno,type, Addr1, Addr2, City, State, Pincode)

```
create table branch(bno varchar(10) primary key,bname varchar(10));

insert into branch values('B001','vejalpur');
insert into branch values('B002','nikol');
insert into branch values('B003','naroda');
insert into branch values('B004','bopal');
insert into branch values('B005','sarkhej');

create table address(addno varchar(20) primary key,bno varchar(10) references
branch(bno),type varchar(10),addr1 varchar(10),addr2 varchar(10),city varchar(10),state
varchar(10),pincode number(6));

insert into address values('A001','B001','H','xyz','pqr','ahmedabad','gujarat',380021);
insert into address values('A002','B002','B','xyz','pqr','surat','gujarat',380022);
insert into address values('A003','B003','H','xyz','pqr','rajkot','gujarat',380023);
insert into address values('A004','B004','B','xyz','pqr','mumbai','gujarat',380023);
```

insert into address values('A005','B005','H','xyz','pqr','baroda','gujarat',380024);

SQL> select * from address;

ADDNO	BNO	TYPE	ADDR1	ADDR2	CITY	STATE	PINCODE
A001	B001	H	xyz	pqr	ahmedabad	gujarat	380021
A002	B002	B	xyz	pqr	surat	gujarat	380022
A003	B003	H	xyz	pqr	rajkot	gujarat	380023
A004	B004	B	xyz	pqr	mumbai	gujarat	380023
A005	B005	H	xyz	pqr	baroda	gujarat	380024

2. type field will have value ‘H’ or ‘B’ (H-Head, B-Branch);

```
select bname,type from branch join address on
branch.bno = address.bno where type like'H'
union
select bname,type from branch join address on
branch.bno = address.bno where type like'B';
```

BNAME	TYPE
bopal	B
naroda	H
nikol	B
sarkhej	H
vejalpur	H

4. Create a view Branch_Master from Brach and Address Table.

create view branch_master2 AS select branch.bno,bname,addno,type, Addr1, Addr2, City, State, Pincode from branch join address on branch.bno = address.bno;

BNO	BNAME	ADDNO	TYPE	ADDR1	ADDR2	CITY	STATE	PINCODE
B001	vejalpur	A001	H	xyz	pqr	ahmedabad	gujarat	380021
B002	nikol	A002	B	xyz	pqr	surat	gujarat	380022
B003	naroda	A003	H	xyz	pqr	rajkot	gujarat	380023
B004	bopal	A004	B	xyz	pqr	mumbai	gujarat	380023
B005	sarkhej	A005	H	xyz	pqr	baroda	gujarat	380024

5. Update Pincode=400079 where bno=102. update branch_master2 set pincode = 400079 where bno = 'B002';

BNO	BNAME	ADDNO	TYPE	ADDR1	ADDR2	CITY	STATE	PINCODE
B001	vejalpur	A001	H	xyz	pqr	ahmedabad	gujarat	380021
B002	nikol	A002	B	xyz	pqr	surat	gujarat	400079
B003	naroda	A003	H	xyz	pqr	rajkot	gujarat	380023
B004	bopal	A004	B	xyz	pqr	mumbai	gujarat	380023
B005	sarkhej	A005	H	xyz	pqr	baroda	gujarat	380024

Delete records where bno=102;

delete branch_master2 where bno = 'B005';

BNO BNAME ADDNO TYPE ADDR1 ADDR2 CITY STATE PINCODE

---------- ---------- -------------------- ---------- ---------- ---------- ---------- ---------- ----------

B001 vejalpur A001 H xyz pqr ahmedabad gujarat 380021

B002 nikol A002 B xyz pqr surat gujarat 400079

B003 naroda A003 H xyz pqr rajkot gujarat 380023

B004 bopal A004 B xyz pqr mumbai gujarat 380023

6. Remove a view Branch_Master from database.

drop view branch_master2;

View dropped.

DAY-12

I. Emloyee (EmpNo, Ename, Salary, Designation)

Dept (EmpNo, DeptNo)

1. Display all rows for salary greater than 5000.

Query:

```
select * from employee1 where salary>5000;
```

2. Display the deptno for the name ‘shyam’.

Query:

```
select deptno from dept1 where empno in (select empno from employee1 where
ename='shyam');
```

3. Add a new column DeptName in Dept table.

Query:

```
alter table employee1 add deptname varchar(10);
```

4. Change the designation of ename=’ram’ from ‘clerk’ to ‘senior clerk’.

Query:

```
update employee1 set designation='senior clerk' where ename='ram';
```

5. Find the total salary of all the rows.

Query:

```
select sum(salary) from employee1;
```

6. Display EmpNo, Ename, DeptNo, DeptName.

Query:

```
select employee1.empno,employee1.ename,dept1.deptno from employee1,dept1 where
employee1.empno in (select empno from dept1 where employee1.empno=dept1.empno);
```

7. Drop the table Employee.

Query:

drop table employee;

----------END OF FIRST QUERY----------

II. Student (StuNo,sname,marks,college)

Course (StuNo,CourseId)

1. Display all rows for student greater than 80.

Query:

select * from student where marks>80;

2. Display the CouseId for the name ‘shyam’.

Query:

select courseid from course where stuno in (select stuno from student where sname='shyam');

3. Add a new column CollegeName in Course table.

Query:

alter table course add collegename varchar(15);

4. Change the college of sname=’ram’ from ‘LJ’ to ‘new LJ’.

Query:

update student set college='new LJ' where sname='ram';

5. Find the total marks of all the rows.

Query:

select sum(marks) from student;

6. Display StuNo,sname,CoursId,CourseName.

Query:

select student1.stuno,student1.sname,course1.courseid from student1,course1 where student1.stuno in (select stuno from course1 where student1.stuno=course1.courseid);

----------END OF SECOND QUERY----------

III. BOOK_CATALOG (book_code, title, ISBN_No, Publisher_Name, yr_of_release, total_copies)

MEMBER (member_code, member_name, mem_ship_dt)

ISSUE (Issue_id, member_code, book_code, issue_date, issue_ret_dt)

1. Create the above tables with appropriate key constraints.

Query:

```
create table book_catalog(
book_code varchar(6) primary key,
title varchar(20) not null,
isbn_no number(3) not null,
publisher_name varchar(15) not null,
yr_of_release date not null,
total_copies number(6) not null);

create table member(
```

member_code varchar(6) primary key,

member_name varchar(10) not null,

mem_ship_dt date not null);

create table issue(

issue_id number(3) primary key,

member_code varchar(6) references member,

book_code varchar(6) references book_catalog,

issue_date date not null,

issue_ret_dt date not null);

2. **Publisher name should be entered in capital letters.**

Query:

3. **Display the book details which contain ‘Database’ somewhere in the book title .**

Query: select * from book_catalog where title like '%Database%';

4. **Display the member and book details for books issued between 1st January 2014 and 30th March 2015.**

Query:

5. **Display book details whose all copies are issued.**

Query:

6. **Display the book details of ‘Pearson’ publications.**

Query:

select * from book_catalog where publisher_name='Pearson';

7. **Display those books having name Database and SQL.**

Query:

select * from book_catalog group by book_code having title='Database' or title='SQL';

8. **Create a table LIBRARY_USER having the same structure of MEMBER table with no records.**

Query:

create table library_USER as select * from member;

9. **Drop table Library_USER from database.**

Query:

drop table Library_USER;

----------END OF THIRD QUERY----------

IV.

STUDENT (rollno, fname, lname, dob)

COURSE (courseno, coursename, max_marks, pass_marks)

1. **Create the above tables with appropriate key constraints.**

Query:

create table student(

rollno number(4) primary key,

fname varchar(10) not null,

lname varchar(10) not null,

dob date not null);

create table course(

courseno number(3) primary key,

coursename varchar(16) not null,

max_marks number(3) not null,

pass_marks number(3) check (pass_marks>0));

2. Marks cannot be less than 0.

Query:

pass_marks number(3) check (pass_marks>0)

3. Display the names of students who have last name like ‘Patel’, ‘Shah’ or ‘Desai’.

Query:

select fname from student where lname='patel' or lname='shah' or lname='desai';

4. Display the names of students who have not failed in any subject .

Query:

5. Display the age of all the students.

Query:

select dob from student;

----------END OF FOURTH QUERY----------

V.

CUSTOMER (cno, cust_name)

ITEM (item_no, item_name, item_price, stock)

CUST_ITEM (cust_no, item_no, qty_purchased, date_of_trans)

1. **Create the above tables with appropriate key constraints.**

Query:

2. **Qty_Purchased cannot be 0.**

Query:

3. **Retrieve the name of customers who have purchased the costliest item from the item list.**

Query:

4. **Display the total item price.**

Query:

5. **Alter table ITEM and add column item class, which can have values as A, B or C. 6. Display those customers who have purchased ‘chair’.**

Query:

6. **Display total number of items purchased by each customer. 6. Display the customer details whose name start with ‘n’.**

Query:

DAY-13

1. Print a static string "Hello Every One…!" using an anonymous PLSQL block and execute

```
begin
   dbms_output.put_line('Hello Every One.....');
end;
```

Hello Every One.....

2. Write a PLSQL block to display a greeting message like: "Hi!! Today is 3rd November 2021, Friday

```
declare
        day1 varchar(10);
begin
        dbms_output.put('Hi!! Today is ' || sysdate());
        day1 := to_char(sysdate(),'day');
        dbms_output.put_line(', ' || day1);
end;
```

Hi!! Today is 05-AUG-22, friday

3. Declare a string variable to store student's name, define three integer variables to store marks of 3 subjects (out of 50) of that student. Write a PLSQL code to calculate total of all three subjects and print the result in percentage.

```
declare
        Sname varchar(10);
        Mark1 number;
        Mark2 number;
        Mark3 number;
```

```
        addition number;
        percentage number(4,2);
begin
        Sname := '&Name';
        dbms_output.put_line('The name is ' || Sname);
        Mark1 := &Mark;
        Mark2 := &Mark;
        Mark3 := &Mark;
        dbms_output.put_line('The Mark1 is ' || Mark1);
        dbms_output.put_line('The Mark2 is ' || Mark2);
        dbms_output.put_line('The Mark3 is ' || Mark3);
        addition := Mark1 + Mark2 + Mark3;
        dbms_output.put_line('The sum is ' || addition);
        percentage := addition/150*100;
        dbms_output.put_line('The Percentage is ' || percentage);
end;
```

4. Write a program to divide 2 numbers and if the denominator if 0 then handle the exception

```
declare
        no1 number;
        no2 number;
        answer number;
begin
        no1 := &noA;
        no2 := &noB;
        dbms_output.put_line('The NO1 is ' || no1);
     dbms_output.put_line('The NO2 is ' || no2);
        if no2!=0 then
        answer := no1/no2;
```

```
        dbms_output.put_line('The Division is ' || answer);
        else
        dbms_output.put_line('The division is not possible');
        end if;
end;

--------------------------------------------------------------------------------
declare
        no1 number;
        no2 number;
        answer number;
        no_divide_zero exception;
begin
        no1 := &noA;
        no2 := &noB;
        dbms_output.put_line('The NO1 is ' || no1);
      dbms_output.put_line('The NO2 is ' || no2);
        if no2!=0 then
        answer := no1/no2;
        dbms_output.put_line('The Division is ' || answer);
        elsif no2=0 then
        RAISE no_divide_zero;
        end if;
exception
        when no_divide_zero then
        dbms_output.put_line('please input valid denominator');
end;
```

5. Write a user defined exception for above program 3 where if marks are less than 0 then appropriate error message must be shown as exception.

```
declare
        Sname varchar(10);
        Mark1 number;
        Mark2 number;
        Mark3 number;
        addition number;
        percentage number(4,2);
        no_marks Exception;
begin
        Sname := '&Name';
        dbms_output.put_line('The name is ' || Sname);
        Mark1 := &Mark;
        Mark2 := &Mark;
        Mark3 := &Mark;
        dbms_output.put_line('The Mark1 is ' || Mark1);
        dbms_output.put_line('The Mark2 is ' || Mark2);
        dbms_output.put_line('The Mark3 is ' || Mark3);
        if Mark1!=0 and Mark2!=0 and Mark3!=0 then
        addition := Mark1 + Mark2 + Mark3;
        dbms_output.put_line('The sum is ' || addition);
        percentage := addition/150*100;
        dbms_output.put_line('The Percentage is ' || percentage);
        else
        RAISE no_marks;
        end if;
Exception
        when no_marks then
```

```
        dbms_output.put_line('please enter valid marks');

end;
```

6. Write a PLSQL block to find the largest of three numbers

```
declare
        A number;
        B number;
        C number;
begin
        A := &noa;
        B := &nob;
        C := &noc;
        if A > B and A > C then
        dbms_output.put_line('A is the largest Number ' || A);
        elsif B > A and B > C then
        dbms_output.put_line('B is the largest Number ' || B);
        else
        dbms_output.put_line('C is the largest Number ' || C);
        end if;
end;
```

DAY-14

1. Write a PLSQL block to print all the prime numbers between 1 to 50.

```
declare
        i number;
        counter number;
        k number;
        n number;
begin
        for n in 1 .. 100
        loop
                counter := 0;
                k := n/2;
                for i in 2..k
                loop
                        if(mod(n, i) = 0) then
                                counter := 1;
                        end if;
                end loop;
                if(counter = 0) then
                        dbms_output.put_line(n || ' is a prime no');
                end if;
        end loop;
end;
```

2. Display all the integer numbers between 4 to 40 which are divisible by 3 using “Exit When” statement.

```
declare
```

```
        i number;
        j number;
begin
        i := 4;
        loop
                i := i + 1;
                if (i/3) then
                        dbms_output.put_line(i);
                end if;
        exit when i <= 40;
        end loop;
end;
```

DAY-15

Use following tables and write below given PL/SQL blocks.

PRODUCTS (Prod_ID, Prod_Name, Supplier_ID, Cat_ID, Unit, Price)

ORDER_DETAILS (OrderDetail_ID, Order_ID, Prod_Id, Quantity)

```
create table product(prod_id varchar(5) primary key,prod_name varchar(20),supplier_id
varchar(10),cat_id number(5),
                    unit number(5), price number(5));

insert into product values('P001','milk','S001',01,100,30);
insert into product values('P002','suagar','S002',02,200,40);
insert into product values('P003','beans','S003',03,300,50);
insert into product values('P004','coco-powder','S004',04,400,60);
insert into product values('P005','ice-cream','S005',05,500,70);

create table order_detail(orderdetail_id varchar(4)primary key,ordear_id
number(4),prod_id varchar(5)references product(prod_id),
quantity number(10));

insert into order_detail values('O001',001,'P001',100);
insert into order_detail values('O002',002,'P002',150);
insert into order_detail values('O003',003,'P003',300);
insert into order_detail values('O004',004,'P004',500);
insert into order_detail values('O005',005,'P005',600);
```

1. Write a PLSQL block to display total number of products ordered in Order_ID = 3

```
declare
```

```
    qty order_detail.quantity%type;
begin
    select quantity into qty from order_detail where ordear_id = 003;
    dbms_output.put_line(qty);
end;
```

2. Write a PLSQL block to update the price (actual price + 5) of product with Id = 2

```
declare
    pri product.price%type;
begin
    update product set price = 45 where prod_id = 'P002';
    select price into pri from product where prod_id = 'P002';
    dbms_output.put_line(pri);
end;
```

3. Write a PLSQL block to delete the products of Cat_Id = 3

```
begin
    delete from product where cat_id = 03;
end;
```

4. Write a PLSQL block to insert any product whose cat_id = 3

```
declare

begin
    insert into product values(
```

5. Write a PLSQL block to display Supplier_Id and their total number of products they supply

```
declare
	cursor csup is select * from product;
	cursor corder(varpid order_detail.prod_id%type) is select * from order_detail where prod_id = varpid;
begin
	for varcsup in csup
	loop
		DBMS_OUTPUT.PUT_LINE('Supplier id: ' || varcsup.supplier_id);
		DBMS_OUTPUT.PUT('product id: ' || varcsup.prod_id || '-->');
			for varcorder in corder(varcsup.prod_id)
			loop
				dbms_output.put_line('Quantity - ' || varcorder.quantity);
			end loop;
	end loop;
end;
```

DAY-16

Create a table Student (R_No, Name, Sub1, Sub2, Sub3, Total, Grade)

```
create table student(R_no number(5),Name varchar(10),sub1 number(3),sub2
number(3),sub3 number(3));

insert into student values(01,'Manav',50,59,70);

insert into student values(02,'Harsh',79,59,70);
insert into student values(03,'akshy',89,25,30);
insert into student values(04,'simmi',50,87,40);
insert into student values(05,'sanket',50,22,12);
insert into student values(06,'Diya',50,50,50);
insert into student values(07,'Disha',40,45,45);
insert into student values(08,'vikas',12,22,22);
insert into student values(09,'vanita',69,96,40);
insert into student values(10,'amisha',49,59,59);
```

1. Write a PLSQL block to

calculate and update the Total for each and every student.

```
declare
        cursor cstud is select * from student;
        tot number;
begin
        for varcstud in cstud
        loop
                tot := 0;
                dbms_output.put_line('Roll_no - ' || varcstud.r_no);
                dbms_output.put_line('Sub1 - ' || varcstud.sub1);
```

```
            dbms_output.put_line('Sub2 - ' || varcstud.sub2);
            dbms_output.put_line('Sub3 - ' || varcstud.sub3);
            tot := tot + varcstud.sub1 + varcstud.sub2 + varcstud.sub3;
            dbms_output.put_line('Total Marks - ' || tot);
            dbms_output.put_line('--------------------');
            update student set total = tot where r_no = varcstud.r_no;
        end loop;
end;
```

2. Calculate the grade of all students, based to total (>70 AA, >60 A, >50 B, >35 C, else Fail)

```
declare
        cursor cstud is select * from student;
        grd varchar(5);
begin
        for varcstud in cstud
        loop
            dbms_output.put_line('total - ' || varcstud.total);
            if varcstud.total > 280 then
                grd := 'AA';
            elsif varcstud.total > 250 then
                grd := 'A';
            elsif varcstud.total > 200 then
                grd := 'B';
            elsif varcstud.total > 150 then
                grd := 'C';
            else
                grd := 'FAIL';
```

```
                end if;
                update student set grade = grd where r_no = varcstud.r_no;
        end loop;
end;
```

3. Write a Cursor to find the first 3 rankers based on the total marks.

```
declare
 name student.name%type;
 Marks student.Total%type;
 Grade student.grade%type;
 cursor v1 is
   select name, Total, Grade from student order by total desc;
begin
 open v1;
 loop
    fetch v1 into name, Marks, Grade;
    exit when v1%rowcount > 4;
    dbms_output.put_line(lpad(name,10)||'  '||lpad(Marks,5)||'  '||lpad(Grade,5));
 end loop;
 close v1;
end;
```

output:

```
Harsh    208      B
vanita    205      B
Manav    179      C
simmi    177      C
```

(II)

```
create table f_master(f_no number(5),fname varchar(10),salary number(10));

insert into f_master values(01,'Urja',70000);
insert into f_master values(02,'Dhaval',70000);
insert into f_master values(03,'Bhavin',90000);
insert into f_master values(04,'Nilam',80000);
insert into f_master values(05,'Jinal',15000);
```

1. Add a Salary and Bonus column in the Faculty_Master Table and calculate the bonus of each faculty of “MCA” department which is based on the 5% of their salary. If the salary in less than 25000, then raise the exception.

```
declare
        cursor c1 is select * from f_master;
        bns number(10);
        salbns number(10);
begin
        bns := 0;
        for c in c1
        loop
                dbms_output.put_line('Salary --> ' || c.salary);
                        if c.salary > 25000 then
                                bns := c.salary * 0.05;
                        end if;
                dbms_output.put_line(bns);
```

```
                salbns := bns + c.salary;
                dbms_output.put_line(salbns);
                update f_master set bonus = salbns where f_no = c.f_no;
                commit;
                bns := 0;
        end loop;
end;
```

F_NO	FNAME	SALARY	BONUS
1	Urja	70000	73500
2	Dhaval	70000	73500
3	Bhavin	90000	94500
4	Nilam	80000	84000
5	Jinal	15000	15000

2. Display name of 2 faculties getting maximum bonus.

```
declare
 name f_master.fname%type;
 bonus f_master.bonus%type;
 cursor c1 is
    Select fName, Bonus from f_master
    where bonus is not null order by bonus desc;
Begin
  open c1;
  loop
```

```
    fetch c1 into name, bonus;
    exit when c1%rowcount > 2;
    dbms_output.put_line('faculty name - ' || name);
    dbms_output.put_line('Bonus - ' || bonus);
    dbms_output.put_line('---------');
  end loop;
  close c1;
end;
```

```
faculty name - Bhavin
Bonus - 4500
---------
faculty name - Nilam
Bonus - 4000
---------
```

(III)

supplier (sid, sname, contactnum)

parts (pid, pname, color, unitrate)

catalog (sid, pid, qty) [primary key(sid,pid)]

```
create table supplier(sid number(3) primary key,sname varchar(9));

insert into supplier values(1,'manav');
insert into supplier values(2,'sanket');
insert into supplier values(3,'harsh');
insert into supplier values(4,'vanita');
insert into supplier values(5,'diya');
```

parts (pid, pname, color, unitrate)

create table parts(pid number(3) primary key,pname varchar(10),color varchar(10),rate number(10));

insert into parts values(1,'break','black',700);

insert into parts values(2,'liner','black',450);

insert into parts values(3,'mirror','silver',500);

insert into parts values(4,'Tier','black',900);

insert into parts values(5,'seat','white',1200);

catalog (sid, pid, qty) [primary key(sid,pid)]

create table catalog1 (sid number(3) references supplier(sid),pid number(3) references parts(pid),qty number(5));

insert into catalog1 values(1,2,200);

insert into catalog1 values(1,3,400);

insert into catalog1 values(2,1,90);

insert into catalog1 values(3,4,120);

insert into catalog1 values(3,5,28);

insert into catalog1 values(5,1,30);

Table :- 1

SID	SNAME	CNUMBER
1	manav	
2	sanket	
3	harsh	
4	vanita	
5	diya	

Table :- 2

PID	PNAME	COLOR	RATE
1	break	black	700
2	liner	black	450
3	mirror	silver	500
4	Tier	black	900
5	seat	white	1200

Table :- 3

SID	PID	QTY
1	2	200
1	3	50
2	1	90
3	4	120
3	5	28
5	1	30

```
declare
  total number;
  Grandtotal number := 0;
  name supplier.sname%type;
  Cursor c1(name supplier.sname%type) is select parts.pid, parts.Pname, qty, rate from catalog1 join parts
  on
  parts.pid = catalog1.pid
  join supplier
  on
  supplier.Sid = catalog1.Sid
```

```
  where Supplier.sname Like name;
begin
  name := '&Supplier_Name';
  for v1 in c1(name)
  loop
  total := v1.QTY * v1.rate;
  Grandtotal := Grandtotal + total;

dbms_output.put_line(lpad(v1.PID,10)||lpad(v1.PNAME,10)||lpad(v1.qty,10)||lpad(v1.rate
,10)||lpad(Total,10)||lpad(Grandtotal,10));
  end loop;
end;
```

DAY-17

1. Write a procedure which will take Faculty ID as an input and will display all the information of that faculty

```
create or replace procedure fmaster
(fno IN f_master.f_no%type)
IS
name f_master.fname%type;
sal f_master.salary%type;
Begin
        select fname,salary into name,sal from f_master where f_no = fno;
        dbms_output.put_line(name);
        dbms_output.put_line(sal);
exception
        when NO_DATA_FOUND then
        dbms_output.put_line('Error - Faculty Number Not Found');
end;
```

2. Write a stored procedure that uses an INOUT parameter and an IN parameter. The user will supply 'M' or 'F' through IN parameter (emp_gender) to count a number of male or female from Employee table. The INOUT parameter (mfgender) will return the result to a user.

```
create table emp(eno number(4),ename varchar(10),gender varchar(1),salary number(5),
        check (gender in('M','F')));

insert into emp values(1,'Manav','M',50000);
insert into emp values(2,'Sanket','M',70000);
insert into emp values(3,'Harsh','M',80000);
insert into emp values(4,'Diya','F',5000);
```

```
insert into emp values(5,'Isha','F',9000);
insert into emp values(6,'Mayur','M',9000);

create or replace procedure empgen
(
 gen IN OUT emp.gender%type
)
IS
begin
select count(gender) into gen from emp where gender = gen;
end;

declare
        gn emp.gender%type;
begin
        gn := '&gender';
        empgen(gn);
        dbms_output.put_line(gn);
end;
```

3. Write a procedure which will take minimum limit and maximum limit of salary and the execution of the procedure will display name of the employees having salary between the range.

```
create or replace procedure psalary
(fno IN f_master.f_no%type)
IS
 name f_master.fname%type;
```

```
 sal f_master.salary%type;
begin
        select fName,Salary into name,sal from f_master where f_no = fno;
        if (sal > 15000 and sal <75000) then
        dbms_output.put_line(name);
        dbms_output.put_line(sal);
        else
        RAISE NO_DATA_FOUND;
        end if;
exception
        when NO_DATA_FOUND then
        dbms_output.put_line('Error - DATA NOT FOUND');
end;



create or replace procedure psalary
(fno IN f_master.f_no%type)
IS
 name f_master.fname%type;
 sal f_master.salary%type;
begin
        select fName,Salary into name,sal from f_master where f_no = fno;
        if (sal > 15000 and sal <75000) then
        dbms_output.put_line(name);
        dbms_output.put_line(sal);
        else
        RAISE NO_DATA_FOUND;
        end if;
exception
```

```
        when NO_DATA_FOUND then
        dbms_output.put_line('Error - DATA NOT FOUND');
end;

create or replace procedure psalary
(fno IN f_master.f_no%type)
IS
 name f_master.fname%type;
 sal f_master.salary%type;
begin
        select fName,Salary into name,sal from f_master where f_no = fno;
        if (sal > 15000 and sal <75000) then
        dbms_output.put_line(name);
        dbms_output.put_line(sal);
        else
        RAISE NO_DATA_FOUND;
        end if;
exception
        when NO_DATA_FOUND then
        dbms_output.put_line('Error - DATA NOT FOUND');
end;
```

III.

```
Movie (movie_id, movie_name, date_of_release)

create table movie(movie_id number(10) primary key,movie_name varchar(15),dor date);
```

insert into movie values(1,'ek_villen','31-jul-2022');

insert into movie values(2,'rakshabandhan','11-aug-2022');

insert into movie values(3,'kgf2','27-feb-2022');

insert into movie values(4,'Sonic','31-july-2022');

insert into movie values(5,'Nadi_dosh','31-july-2022');

Screen (screen_id, location, max_capacity)

create table screen(s_id number(5) primary key,location varchar(10),m_cap number(5));

insert into screen values(1,'Nikol',200);

insert into screen values(2,'CTM',100);

insert into screen values(3,'Raipur',80);

insert into screen values(4,'Bapunagar',100);

Current (movie_id,screen_id, date_of_arrival, date_of_closure)

create table current1(m_id number(10) references movie(movie_id),s_id number(5) references screen(s_id),doa date,doc date);

insert into current1 values(1,1,'31-jul-2022','7-aug-2022');

insert into current1 values(1,3,'31-jul-2022','15-aug-2022');

insert into current1 values(2,4,'11-aug-2022','20-aug-2022');

insert into current1 values(3,4,'27-feb-2022','3-apr-2022');

insert into current1 values(5,2,'31-jul-2022','15-aug-2022');

insert into current1 values(4,1,'31-jul-2022','12-aug-2022');

insert into current1 values(4,2,'12-aug-2022','30-aug-2022');

1. **Consider the above table and write a function to return the movie name which arrived today.**

```
create or replace function fdate
(mname OUT movie.movie_name%type,
```