NAME- SANMAY DAS ENROLMENT NO. - 2011200001045 DEPARTMENT- CSE

**CREATING DEPARTMENT TABLE**

create table department1( dname varchar2(50),

dnumber number(1) primary key, mgrstartdate date);

INSERT INTO DEPARTMENT1 VALUES ('Reasearch', 5, date'1988- 05-22');

INSERT INTO DEPARTMENT1 VALUES ('Administration', 4, date'1995-01-01');

INSERT INTO DEPARTMENT1 VALUES ('Headquarters', 1, date'1981-06-19');

**CREATING TABLE EMPLOYEE**

create table employee( fname varchar2(20), minit char,

lname varchar2(20),

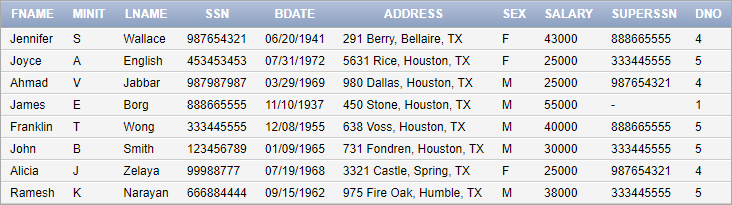
ssn number(9) primary key, bdate date,

address varchar2(100), sex char,

salary int,

superssn number(9) references employee(ssn),

dno number(1) references department1(dnumber));



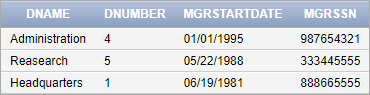
alter table department1 add mgrssn number(9) references employee(ssn);

update department1 set mgrssn = 333445555 where dnumber = 5;

update department1 set mgrssn = 987654321 where dnumber = 4;

update department1 set mgrssn = 888665555 where dnumber = 1;

select \* from department1;



**CREATING TABLE DEPT\_LOCATION**

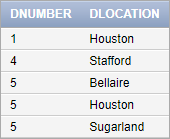
create table dept\_locations(

dnumber number(1) references department1(dnumber), dlocation varchar2(50),

primary key(dnumber, dlocation));

INSERT INTO DEPT\_LOCATIONS VALUES (1,'Houston'); INSERT INTO DEPT\_LOCATIONS VALUES (4,'Stafford'); INSERT INTO DEPT\_LOCATIONS VALUES (5,'Bellaire'); INSERT INTO DEPT\_LOCATIONS VALUES (5,'Sugarland'); INSERT INTO DEPT\_LOCATIONS VALUES (5,'Houston');

select \* from dept\_locations;



**CREATING TABLE PROJECT**

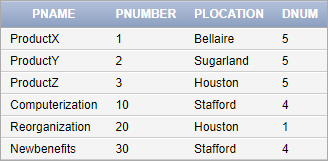
create table project( pname varchar2(50), pnumber int primary key, plocation varchar2(50), dnum number(1),

foreign key(plocation, dnum) references dept\_locations(dlocation, dnumber));

INSERT INTO PROJECT VALUES ('ProductX', 1,'Bellaire', 5); INSERT INTO PROJECT VALUES ('ProductY', 2,'Sugarland', 5); INSERT INTO PROJECT VALUES ('ProductZ', 3,'Houston', 5);

INSERT INTO PROJECT VALUES ('Computerization', 10,'Stafford', 4);

INSERT INTO PROJECT VALUES ('Reorganization', 20,'Houston', 1);

INSERT INTO PROJECT VALUES ('Newbenefits', 30,'Stafford', 4); select \* from project;

**CREATING TABLE WORKS\_ON**

create table works\_on(

essn number(9) references employee(ssn), pno int references project(pnumber), hours number(3, 1),

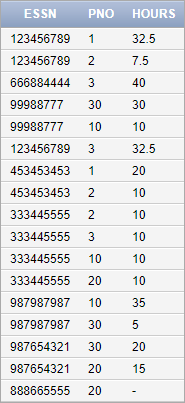
primary key(essn, pno));

INSERT INTO WORKS\_ON VALUES (123456789,1,32.5); INSERT INTO WORKS\_ON VALUES (123456789,2,7.5);

insert into works\_on values(123456789,3,32.5);

INSERT INTO WORKS\_ON VALUES (666884444,3,40.0); INSERT INTO WORKS\_ON VALUES (453453453,1,20.0); INSERT INTO WORKS\_ON VALUES (453453453,2,10.0); INSERT INTO WORKS\_ON VALUES (333445555,2,10.0); INSERT INTO WORKS\_ON VALUES (333445555,3,10.0); INSERT INTO WORKS\_ON VALUES (333445555,10,10.0); INSERT INTO WORKS\_ON VALUES (333445555,20,10.0); INSERT INTO WORKS\_ON VALUES (99988777,30,30.0); INSERT INTO WORKS\_ON VALUES (99988777,10,10.0); INSERT INTO WORKS\_ON VALUES (987987987,10,35.0); INSERT INTO WORKS\_ON VALUES (987987987,30,5.0); INSERT INTO WORKS\_ON VALUES (987654321,30,20.0); INSERT INTO WORKS\_ON VALUES (987654321,20,15.0); INSERT INTO WORKS\_ON VALUES (888665555,20,null);

select \* from works\_on;



**CREATING TABLE DEPENDENT**

create table dependent(

essn number(9) references employee(ssn), dfname varchar2(30),

dlname varchar2(30), sex char,

bdate date,

relationship varchar2(20),

primary key(essn, dfname, dlname));

INSERT INTO DEPENDENT VALUES (333445555,'Alice', 'Wong', 'F', date'1986-04-05','DAUGHTER');

INSERT INTO DEPENDENT VALUES (333445555,'Theodore', 'Smith', 'M', date'1983-10-25','SON');

INSERT INTO DEPENDENT VALUES (333445555,'Joy', 'Smith', 'F', date'1958-05-03','SPOUSE');

INSERT INTO DEPENDENT VALUES (987654321,'Abner', 'Nelson', 'M', date'1942-02-28','SPOUSE');

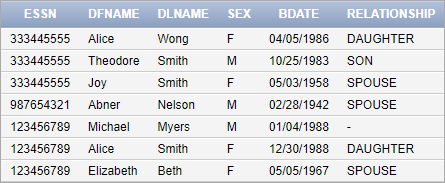
INSERT INTO DEPENDENT VALUES (123456789,'Michael', 'Myers', 'M', date'1988-01-04','');

INSERT INTO DEPENDENT VALUES (123456789,'Alice', 'Smith', 'F', date'1988-12-30','DAUGHTER');

INSERT INTO DEPENDENT VALUES (123456789,'Elizabeth', 'Beth'

,'F', date'1967-05-05','SPOUSE');

select \* from dependent;

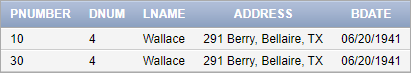


# For every project located in 'Stafford', list the project number, the controlling department number, and the department

**manager's last name, address, and birth date.**

Query

select pnumber, dnum, lname, address, bdate from (select pnumber, dnum from project where plocation='Stafford') natural join (select dnumber as dnum, mgrssn as ssn from department1) natural join (select ssn, lname, bdate, address from employee);



# Retrieve the name of each employee who works on all the projects controlled by department number 5.

Query

Select fname || ' ' || minit || ' ' || lname as name from ((select essn as ssn from works\_on minus (select essn from ((select \* from ((select distinct essn from works\_on) cross join (select pnumber from project where dnum = 5))) minus (select essn, pno from works\_on)))) natural join employee);

Output



1. **Make a list of all project numbers for projects that involve an employee whose last name is 'Smith', either as a worker or as a manager of the department that controls the project.**

Query

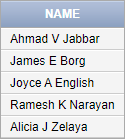
select pnumber from project natural join (select mgrssn, dnumber as dnum from department1) where pnumber in (select pno from works\_on natural join (select ssn as essn from employee where lname = 'Smith')) or mgrssn in (select ssn from employee where lname = 'Smith');

Output

# Retrieve the names of employees who have no dependents.

Query

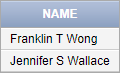
select fname || ' ' || minit || ' ' || lname as name from employee where ssn not in (select essn from dependent);

Output

# List the names of managers who have at least one dependent.

Query

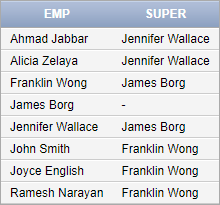
select fname || ' ' || minit || ' ' || lname as name from employee where ssn in (select mgrssn from department1 natural join (select essn as mgrssn from dependent));

Output

# For each employee, retrieve the employee's first and last name and the first and last name of his or her immediate supervisor.

Query

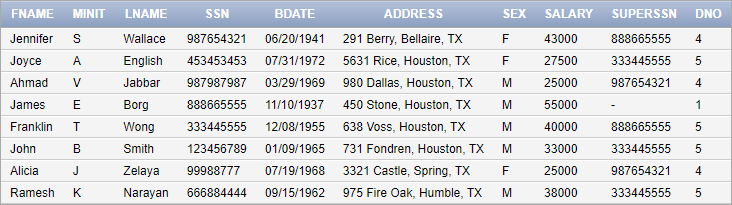
select e.fname || ' ' || e.lname as emp, s.fname || ' ' || s.lname as super from (employee e cross join employee s) where e.superssn = s.ssn union select fname || ' ' || lname, '' from employee where superssn is null;

Output

# Show the resulting salaries if every employee working on the 'ProductX' project is given a 10 percent raise.

Query

update employee set salary = salary + 0.1\* salary where ssn in (select essn from works\_on where pno = (select pnumber from project where pname = 'ProductX'));

select \* from employee; Output

# Retrieve a list of employees and the projects they are working on, ordered by department and, within each department, ordered alphabetically by last name, first name.

Query

select fname || ' ' || lname as name, pname, dname from employee cross join department1 cross join project cross join

works\_on where ssn = essn and pno = pnumber and dnum = dnumber order by dname, lname, fname;

Output

# Retrieve the names of all employees who do not have supervisors.

Query

select fname || ' ' || lname as name from employee where superssn is null;

Output

# Retrieve the name of each employee who has a dependent with the same last name as the employee.

Query

select fname || ' ' || lname as name from ((select fname, lname, ssn as essn from employee) natural join dependent) where lname = dlname;

Output

# Retrieve the social security numbers of all employees who work on project numbers 1, 2.

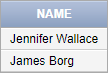
Query

select distinct essn from works\_on where pno in (1, 2); Output

# Returns the names of employees whose salary is greater than the salary of all the employees in department 5:

Query

select fname || ' ' || lname as name from employee where salary > all(select salary from employee where dno = 5);

Output

# Find the sum of the salaries of all employees, the maximum salary, the minimum salary, and the average salary.

Query

select sum(salary) as sum, max(salary) as max, min(salary) as min, avg(salary) as avg from employee;

Output

# Find the sum of the salaries of all employees of the 'Research' department, as well as the maximum salary, the minimum salary, and the average salary in this department.

Query

select sum(salary) as sum, max(salary) as max, min(salary) as min, avg(salary) as avg from employee where dno = (select dnumber from department1 where dname = 'Reasearch');

Output

# Retrieve the names of all employees who have two or more dependents.

Query

select fname || ' ' || lname as name from employee where ssn in (select essn from dependent group by essn having count(\*)

>= 2);

Output

# Count the total number of employees whose salaries exceed $40,000 in each department, but only for departments where more than five employees work.

Query

select dno, count(\*) as count from employee where salary > 40000 group by dno having count(\*) > 5;

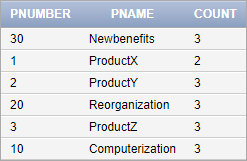
Output



# For each project, retrieve the project number, the project name, and the number of employees who work on that project.

Query

select pnumber, pname, count from ((select pno as pnumber, count(\*) as count from works\_on group by pno) natural join project);

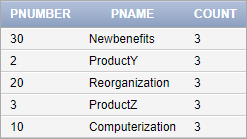
Output

# For each project on which more than two employees work, retrieve the project number, the project name, and the number of employees who work on the project.

Query

select pnumber, pname, count from ((select pno as pnumber, count(\*) as count from works\_on group by pno having count(\*)

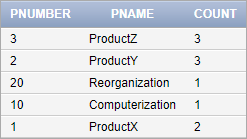
> 2) natural join project);

Output

# For each project, retrieve the project number, the project name, and the number of employees from department 5 who work on the project.

Query

select pnumber, pname, count from ((select pno as pnumber, count(\*) as count from works\_on where essn in (select ssn from employee where dno = 5) group by pno) natural join project);

Output

# For each department that has more than five employees, retrieve the department number and the number of its employees who are making more than

**$40,000.**

Query SELECT

Department1.dnumber, COUNT(Employee.salary) as count

FROM

Department1 LEFT JOIN Employee

ON Department1.dnumber = Employee.dno AND Employee.salary > 40000

WHERE

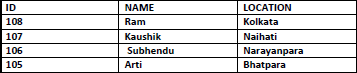
Department1.dnumber IN ( SELECT Employee.dno FROM Employee

GROUP BY Employee.dno HAVING COUNT(\*) > 5

)

GROUP BY Department1.dnumber;

**ASSIGNMENT-6**

* 1. Create the following Table: (ID number(3),Name varchar2(20),location varchar2(10)**Value of ID should be Auto generated**(using Sequence)

CREATE SEQUENCE seq\_1 START WITH 108

INCREMENT BY -1

MAXVALUE 1000 NOCYCLE;

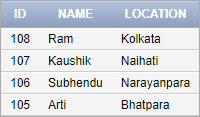
create table tab1( id number(3), name varchar2(20),

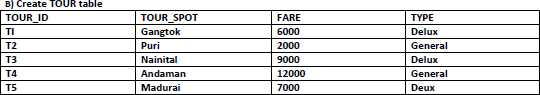
location varchar2(20));

insert into tab1 values(seq\_1.nextval, 'Ram', 'Kolkata'); insert into tab1 values(seq\_1.nextval, 'Kaushik', 'Naihati');

insert into tab1 values(seq\_1.nextval, 'Subhendu', 'Narayanpara'); insert into tab1 values(seq\_1.nextval, 'Arti', 'Bhatpara');

select \* from tab1 order by ID desc;





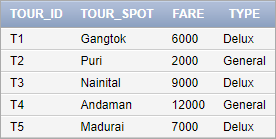
Create table Tour ( Tour\_id varchar2(5),

Tour\_Spot varchar2(20), Fare number(10),

Type varchar2(20)); desc tour

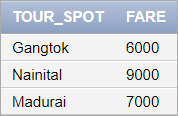
Insert into tour values('T1', 'Gangtok', 6000 , 'Delux'); Insert into tour values('T2', 'Puri', 2000 , 'General'); Insert into tour values('T3', 'Nainital', 9000 , 'Delux');

Insert into tour values('T4', 'Andaman', 12000 , 'General'); Insert into tour values('T5', 'Madurai', 7000 , 'Delux'); select \* from tour;



1. **Create a view TOURVIEW for deluxe type tour containing two fields, tour-spotnames and fares**

create view Tourview as( select Tour\_spot,Fare from Tour where Type='Delux'); select \* from tourview;



1. **Find all the Tour spots for fare greater than 8000 and Delux type tour from a) TOUR table,**
2. **TOURVIEW view**
   1. select tour\_spot from tour where fare>=8000;



* 1. select Tour\_spot from Tourview where Fare>=8000;

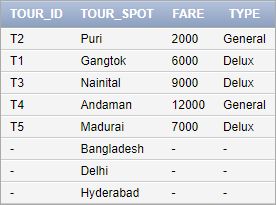


1. **Insert three rows to TOURVIEW.the location will be Bangladesh, Delhi, Hyderabad.**

insert into Tourview Values('Bangladesh',''); insert into Tourview Values('Delhi',''); insert into Tourview Values('Hyderabad','');

1. **Display these Records.Are they seen in TOURVIEW? Are they seen in TOUR Table?**

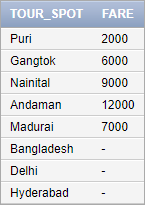
No they cannot be seen in Tourview.

Yes they are seen in Tour Table. Select \* from tour;

1. **Make them Seen in TOURVIEW.**

drop view tourview

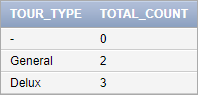
create view Tourview as(select Tour\_spot,Fare from Tour); select \* from Tourview;



1. **Create a view Tour\_type\_num containing two fields i.e. tour\_type and its total number of count.**

create view Tour\_type\_num (Tour\_Type ,Total\_Count)as select type,count(type) from tour GROUP BY type;

select \* from Tour\_type\_num;



1. **Try to insert records via this view. Errors!!,Explain why?**

A VIEW in SQL is like a virtual table that contains data from one or multiple tables. It does not hold any data and does not exist physically in the database. We can say that it’s contents are totally based on the base table. So whenever we try to insert any record via view the actual changes takes place on the base table and not on the view table directly .Toview the changes in the View we need to create a view after the changes to notice it.

**ASSIGNMENT-7**

**Customer(Cust id : integer, cust\_name: string) Item(item\_id: integer, item\_name: string, price: integer)**

**Sale(bill\_no: integer, bill\_data: date, cust\_id: integer, item\_id: integer, qty\_sold: integer)**

1. **Create the tables with the appropriate integrity constraints**

create table Customer(Cust\_id number NOT NULL primary key, cust\_name varchar2(20));

desc customer

create table Item(item\_id number NOT NULL primary key, item\_name varchar2(10),

price number NOT NULL); desc Item

create table Sale(bill\_no number NOT NULL, bill\_date date,

cust\_id number references Customer(cust\_id), item\_id number references Item(Item\_id), qty\_sold number);

desc sale

1. **Insert around 10 records in each of the tables**

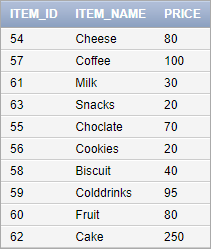
Insert into Customer Values (1, 'Rohit'); Insert into Customer Values (2, 'Ishan'); Insert into Customer Values (3, 'Virat'); Insert into Customer Values (4, 'Risab'); Insert into Customer Values (5, 'Hardik'); Insert into Customer Values (6, 'Jadeja'); Insert into Customer Values (7, 'Yuzi'); Insert into Customer Values (8, 'Shami');

Insert into Customer Values (9, 'Umran'); Insert into Customer Values (10, 'Bhuvi');

select \* from customer;

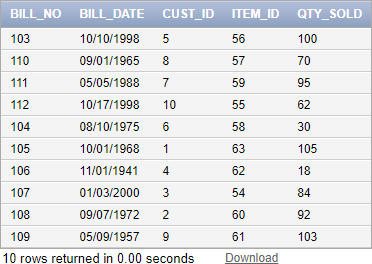


Insert into Item Values (54, 'Cheese',80); Insert into Item Values (55, 'Choclate',70); Insert into Item Values (56, 'Cookies',20); Insert into Item Values (57, 'Coffee',100); Insert into Item Values (58, 'Biscuit',40); Insert into Item Values (59, 'Colddrinks',95); Insert into Item Values (60, 'Fruit',80); Insert into Item Values (61, 'Milk',30); Insert into Item Values (62, 'Cake',250); Insert into Item Values (63, 'Snacks',20);

select \* from Item;

Insert into Sale Values (103, date'1998-10-10',5,56,100); Insert into Sale Values (104, date'1975-08-10',6,58,30); Insert into Sale Values (105, date'1968-10-01',1,63,105);

Insert into Sale Values (106, date'1941-11-01',4,62,18); Insert into Sale Values (107, date'2000-01-03',3,54,84); Insert into Sale Values (108, date'1972-09-07',2,60,92); Insert into Sale Values (109,date'1957-05-09',9,61,103); Insert into Sale Values (110,date'1965-09-01',8,57,70); Insert into Sale Values (111,date'1988-05-05',7,59,95); Insert into Sale Values (112,date'1998-10-17',10,55,62);

select \* from Sale;

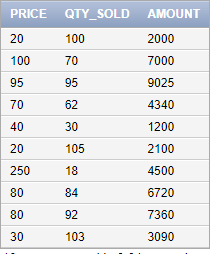
1. **List all the bills for the current date with the customer names and item numbers**

select c.cust\_name,i.item\_id,s.bill\_no from customer c, item i, Sale s where c.cust\_id=s.cust\_id and s.bill\_date=to\_char(sysdate);



1. **List the total Bill details with the quantity sold, price of the item and the final amount**

select i.price,s.qty\_sold,(i.price \* s.qty\_sold)Amount from item I,sales s where i.tem\_id=s.item\_id;



1. **List the details of the customer who have bought a product which has a price>200**

select c.cust\_id, c.cust\_name from customer c, sale s, item i where i.price>200 and c.cust\_id=s.cust\_id and i.item\_id=s.item\_id;



1. **Give a count of how many products have been bought by each customer**

select cust\_id, count(item\_id) from sale group by cust\_id;

1. **Give a list of products bought by a customer having cust\_id as 5**

select i.item\_name from item i, sale s where s.cust\_id=5 and i.item\_id=s.item\_id;



1. **List the item details which are sold as of today**

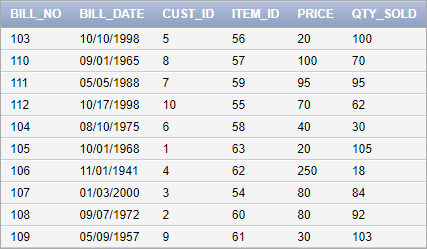
select c.cust\_name,i.item\_id,s.bill\_no from customer c, item i, Sale s where c.cust\_id=s.cust\_id and s.bill\_date=to\_char(sysdate);



1. **Create a view which lists out the bill\_no, bill\_date, cust\_id, item\_id, price,qty\_sold, amount**

create view Shopping as (select s.bill\_no,s.bill\_date,c.cust\_id,i.item\_id,i.price,s.qty\_sold from customer c,sale s,item I where c.cust\_id=s.cust\_id and i.item\_id=s.item\_id);

select \* from shopping;



1. **Create a view which lists the daily sales date wise for the last one week**

create view week as(select \* from sale where bill\_date between date'1998-10-10' and date'1998-10-17');

select \* from week;

