1. **Consider the following schema:**

**Suppliers (sid, sname, address)**

**Parts (pid, pname, color)**

**Catalog (sid, pid, cost)**

**Write the relational algebraic queries for the following:**

**i)Find the sids of suppliers who supply some red or green part**

**select**

**s.\***

**from Suppliers S, parts P,catalog C**

**where s.sid=c.sid and p.pid=c.pid**

**and p.color in('Red','Green');**

**select**

**unique(c.sid)**

**from parts P,catalog C**

**where p.pid=c.pid**

**and p.color in('Red','Green');**

**ii) Find the sids of suppliers who supply every red or green part**

**Query a: Find the number of red parts**

**select**

**Count(\*)**

**From parts**

**Where color=’Red’;**

**Query b: Find the number of red parts supplied by each supplier**

**select**

**Sid, count(\*)**

**From parts p, catalog c**

**Where p.pid=c.pid**

**And color=’Red’**

**Group by sid;**

**Select**

**sid   
From part p, catalog c**

**where p.pid=c.pid**

**and color=’Red’   
Group by sid   
Having count(\*)=(select  
 Count(pid)  
 From parts   
 Where color ='red')**

**Select**

**sid   
From part p, catalog c**

**where p.pid=c.pid**

**and color=’Red’   
Group by sid   
Having count(\*)=(select  
 Count(pid)  
 From parts   
 Where color ='red')  
Union   
Select   
Sid   
From catalog c ,parts p  
P. PID=c. Pid   
And p. Colour='green';**

**iii) Find the pids of parts supplied by at least two different suppliers.**

**Select  
PID   
From catalog   
Group by pid  
Having count(\*) >=2;**

1. **Consider the following schemas:**

**Sailors (sid, sname, rating, age)**

**Reserves (sid, bid, day)**

**Boats (bid, bname, color)**

**Write the following queries in relational algebra, tuple relational Calculus and**

**domain relational calculus:**

**a) Find the name of sailors who have reserved boat 103.**

**b) Find the names and ages of sailors with a rating above 7.**

**c) Find the names of sailors who have reserved a red boat.**

**d) Find the sname, bid, and day for each reservation.**

**e) Find the name of sailors who have reserved at least one boat.**

1. **Consider the SAILOR DATABASE**

**Sailors (sid:string, sname:string, rating:integer, age:real)**

**Boats (bid:integer, bname:string, color:string)**

**Reserves (sid:integer, bid:integer, day:date)**

**Based on the above schema, write the corresponding SQL queries for the following?**

1. **Find the colors of boats reserved by Lubber.**

**SELECT**

**Unique(B.color )**

**FROM Sailors S, Reserves R, Boats B**

**WHERE S.sid=R.sid AND R.bid=B.bid**

**AND S.sname= ‘Lubber’ ;**

1. **Find the names of sailors who have reserved at least one boat.**

**SELECT**

**Unique(S.sid, S.sname)**

**FROM Sailors S, Reserves R**

**WHERE S.sid=R.sid;**

1. **Find the names of sailors who have reserved a red or green boat.**

**Select  
S.sname  
From Sailors s,boats b, Reserves r  
Where s.sid=r.sid and** [**b.bid**](http://b.bid/)**=**[**r.bid**](http://r.bid/) **and b.color in ('Red','Green');**

1. **Find the names of the sailors who have reserved both a Red boat and a Green boat.**

**select  
S.sname  
From Sailors s, Boats b, Reserves r  
Where s.sid=r.sid and** [**b.bid**](http://b.bid/)**=**[**r.bid**](http://r.bid/) **And b.color='Red'  
Intersect  
Select  
S.sname  
From Sailors s, Boats b, Reserves r  
Where s.sid=r.sid and** [**b.bid**](http://b.bid/)**=**[**r.bid**](http://r.bid/) **and b.color='Green';**

1. **Find names of sailors who have reserved all boats.**

**Query a: Find the total number of boats**

**select**

**Count(\*)**

**From boats;**

**Query b: Find the number of boats reserved by each sailor**

**select**

**Sid, count(\*)**

**From reserves**

**Group by sid;**

**Query c:**

**Query b**

**Having count(\*)=(query a);**

**Select**

**Sid**

**From reserves**

**Group by sid**

**Having count(\*) = (select**

**Count(\*)**

**From parts);**

**Query d: find the names of sailors obtained in query c**

**select**

**sname**

**From sailors**

**Where sid in (Select**

**Sid**

**From reserves**

**Group by sid**

**Having count(\*) = (select**

**Count(\*)**

**From parts));**

1. **Consider the Bank Management System.**

**Account (account\_number, branch\_name, balance)**

**Branch (branch\_name, branch\_city, assets)**

**Customer (customer\_name, customer\_street, customer\_city)**

**Loan (loan\_number, branch\_name, amount)**

**Depositor (customer\_name, account\_number)**

**Borrower (customer\_name, loan\_number)**

**Based on the above schema, write the corresponding SQL queries for the**

**following?**

1. **For all customers who have a loan from the bank, find their names, loan numbers and loan amount.**

**Select**

**Customer\_name, L.loan\_number, amount**

**From Loan L, Borrower B**

**Where L.loan\_number=B.loan\_number;**

1. **Find the customer names, loan numbers, and loan amounts, for all loans at the Perryridge branch.**

**Select  
Customer\_name, a. Loan\_number, a. Loan\_amount  
From borrower b, loan a  
Where b. Loan\_number =a. Loan\_number  
And a.Branch\_name='perrybridge';**

1. **Find the names of all branches that have assets greater than those of at-least one branch located in Brooklyn.**

**select**

**branch\_name  
from branch  
where assests > (select**

**min(assests)**

**from branch**

**where brach\_name='Brooklyn');**

1. **Find the average account balance of those branches where the account balance is greater than Rs. 1200.**

**Solution 1: Find the average balance of all branches with account balance > 1200**

**select**

**Branch\_name, avg(balance)**

**From account**

**Where balance>1200**

**Group by branch\_name;**

**Solution 2:**

**Query a: Find the branches which maintain minimum account balance as 1200**

**Select**

**Branch\_name**

**From account**

**Group by branch\_name**

**Having min(balance)>=1200;**

**Query b: Find the average account balance for the branches obtained in query a**

**Select**

**Branch\_name, avg(balance)**

**From account**

**Where branch\_name in (query a)**

**Group by branch\_name;**

**Select**

**Branch\_name, avg(balance)**

**From account**

**Where branch\_name in (Select**

**Branch\_name**

**From account**

**Group by branch\_name**

**Having min(balance)>=1200)**

**Group by branch\_name;**

1. **Find the maximum across all branches of the total balance at each branch.**

**select**

**Max(Sum(balance))**

**From account**

**Group by branch\_name;**

**select  
max(total\_balance)  
from (select  
 sum(balance) total\_balance  
 from account  
 group by branch\_name);**

1. **Consider the Bank Management System.**

**account(account\_number, branch\_name, balance)**

**branch (branch\_name, branch\_city, assets)**

**customer (customer\_name customer\_street, customer\_city)**

**loan (loan\_number, branch\_name, amount)**

**depositor((customer\_name, account\_number)**

**borrower(customer\_name, loan\_number)**

**Answer the following queries using relational algebra operators.**

1. **List all branch names and their assets**

**select**

**Branch\_name, assets**

**From branch;**

1. **List all accounts of Brooklyn branch**

**select**

**Account\_number**

**From account**

**Where branch\_name=’Brooklyn’;**

1. **List all loans with amount > 1000.**

**select**

**Loan\_number**

**From loan**

**Where amount > 1000;**

1. **List all accounts of Perryridge branch with balance < 1000.**

**select**

**Account\_number**

**From account**

**Where branch\_name=’Perryridge’ and balance < 1000;**

1. **List Numbers of accounts with balances between 700 and 900**

**select**

**Count(\*)**

**From account**

**Where balance >700 and balance <900;**

1. **Consider following relations and write SQL queries for given statements.**

**Assume suitable constrains.**

**Instructor(ID, Name, Dept\_name , Salary)**

**Teaches(ID, Course\_id, Sec\_id, Semester(even/odd),Year)**

1. **Find the average salary of the instructors in computer department.**

**Select  
Avg(salary)  
From instructors  
Where dept\_name='Computer';**

1. **Find the number of instructors in each department who teach a course in even semester of 2016.**

**select**

**Dept\_name, count(\*)**

**From instructor t1, teaches t2**

**Where t1.id=t2.id**

**And semester=’Even’ and Year=’2016’**

**Group by dept\_name;**

1. **Find the names of instructor with salary amounts between 30000 and 50000.**

**select**

**name**

**From instructor**

**Where salary >= 30000 and 50000;**

**Where salary between 30000 and 50000;**

1. **Consider the following schemas:**

**Sailors (sid, sname, rating, age)**

**Reserves (sid, bid, day)**

**Boats (bid, bname, color)**

**Write the following queries in relational algebra, tuple relational Calculus and**

**domain relational calculus:**

1. **Find the name of sailors who have reserved boat 103.**

**Select**

**sname ,s.sid**

**from sailors s,reserves r,boats b**

**where s.sid=r.sid and b.bid=r.bid**

**and b.bid=103;**

**select**

**s.name**

**from Sailors s,Reserves r**

**where s.sid=r.sid**

**and r.bid=103 ;**

1. **Find the names and ages of sailors with a rating above 7.**

**select**

**sname, age**

**from Sailors**

**where rating > 7 ;**

1. **Find the names of sailors who have reserved a red boat.**

**select**

**sname ,s.sid**

**from sailors s,reserves r,boats b**

**where s.sid=r.sid and b.bid=r.bid**

**and color='red';**

**Select   
Sname   
From sailors   
Where sid in (select   
 Sid   
 From reserves   
 Where bid in (select   
 Bid   
 From boats   
 Where color ='red'));**

1. **Find the sname, bid, and day for each reservation.**

**select**

**sname, bid, day**

**from sailors s,reserves r**

**where s.sid=r.sid**

**order by day;**

1. **Find the name of sailors who have reserved at least one boat.**

**Select   
Sname   
From sailors   
Where sid in (select   
 Sid   
 From reserves);**

**select**

**unique(s.sid, sname)**

**from sailors s,reserves r**

**where s.sid=r.sid;**

1. **Consider the following schema:**

**Suppliers (sid, sname, address)**

**Parts (pid, pname, color)**

**Catalog (sid, pid, cost)**

**Write the relational algebraic queries for the following:**

**i)Find the sids of suppliers who supply some red or green part**

**ii) Find the sids of suppliers who supply every red or green part**

**iii) Find the pids of parts supplied by at least two different suppliers.**

1. **Consider the following database schema to write queries in SQL**

**Sailor(sid, sname, age, rating)**

**Boats(bid, bname, bcolor)**

**Reserves(sid, bid, day)**

1. **Find the sailors who have reserved a red boat**
2. **Find the names of the sailors who have reserved at least two boats**

**Select   
Sname   
From sailors   
Where sid in (select   
 Sid   
 From reserves   
 Group by sid   
 Having count(\*)>=2);**

**Select  
s.sid  
From sailors s,reserves r  
Where s.sid=r.sid  
Group by r.sid  
Having count(\*)>=2;**

1. **Find the colors of the boats reserved by ‘Mohan’.**

**select  
distinct(B.color)  
from Sailors s, Reserves r, boats B   
where s .sid=r .sid And b. Bid =s. Bid   
And s.Sname='mohan';**

**select**

**unique(color)**

**from boats**

**where bid in(select**

**bid**

**from reserves**

**where sid in(select**

**sid**

**from saliors**

**where sname='Mohan'));**

1. **Consider the following relation schema:**

**Sailors(sid: integer, sname: string, rating: integer, age: real)**

**Boat(bid: integer, bname: string, color: string)**

**Reserves(sid: integer, bid: integer, day: date)**

**Write the following queries in SQL.**

1. **Find the average age of the sailor who are eligible for voting for each rating level that has at least two sailors.**

**select**

**Rating, avg(age)**

**From sailors**

**Where age >=18**

**Group by rating**

**Having count(\*)>=2;**

1. **Find the name of sailors who have reserved both red and a green boat.**

**select**

**s.sid, sname**

**From sailors s, boat b, reserves r**

**Where s.sid=r.sid and b.bid=r.bid**

**And color=’Red’**

**INTERSECT**

**select**

**s.sid, sname**

**From sailors s, boat b, reserves r**

**Where s.sid=r.sid and b.bid=r.bid**

**And color=’Green’;**

1. **Find the sailor\_id of sailors who have reserved a red boat**

**select**

**s.sid**

**From sailors s, boat b, reserves r**

**Where s.sid=r.sid and b.bid=r.bid**

**And color=’Red’;**

1. **Consider the following database.**

**Employee (employee-name, street, city)**

**Works (employee-name, company-name, salary)**

**Company (company-name, city)**

**Manager (employee-name, manager-name)**

**Give an expression in the relational algebra, the tuple relational calculus, and the**

**domain relational calculus, for the following queries.**

1. **Find the names of all employees who work for “XYZ” bank.**

**select**

**Employee-name**

**From employee e, works w**

**Where e.employee-name=w.employee-name**

**And company-name=’XYZ’;**

1. **Find the street, city of the employee working under manager “Krishna”.**

**select**

**Street, city**

**From employee e, manager m**

**Where e.employee-name=m.employee-name**

**And manager-name=’Krishna’;**

1. **For the following relational database write the expressions in SQL.**

**Branch (branch name, Branchcity, Assets)**

**Customer (customername, customerstreet, customercity)**

**Loan (Branchname, loan number, Amount)**

**Borrower (customername, loan number)**

**Account (Branchname, Account number, balance)**

**Depositor (customername, Account number)**

1. **Find the names of all branches in Loan Schema?**

**select**

**branchname**

**From loan;**

1. **Find all customers having loan, account or both at bank?**

**Select**

**Customername**

**From borrower**

**Union**

**Select**

**Customername**

**From depositor;**

1. **Display customernames in alphabetical order who have a loan at the Newyork**

**branch?**

**select**

**customername**

**from borrower b,loan l**

**where b.loannumber=l.loannumber**

**and branchname='Newyork'**

**order by customername;**

1. **Find set of all customer names who have an account at the bank?**

**Select**

**Customername**

**From depositor;**

1. **Using the following relational database, give the expressions in SQL.**

**Branch(branch name, branch city, assets),**

**customer (customer name, customer street, customer city),**

**Loan (branch name, loan number, amount),**

**Borrower (customer name, Loan number),**

**Account (branch name, account number, balance),**

**Depositor (Customer name, account number)**

**i. find the names of all customers whose street address include**

**substring ’Main’**

**select**

**customername**

**from customer**

**where customer street like '%Main%';**

**ii. Find average balance for each customer who lives in Harrison and**

**having at least three accounts?**

**select**

**avg(a.balance), d.customername**

**from depositor d, account a , customer c**

**where a.accountnumber=d.accountnumber**

**and d.customername=c.customername**

**and customer street ='Harrison'**

**group by d.customername**

**having count(\*)>=3;**

**iii. Find the customers from all branches who have maximum loan or**

**deposit in that branch.**

**select  
Cname  
from loan l,borrower b  
where l.loannumber=b.loannumber  
and amount=(select  
 max(amount)  
 from loan)**

**UNION**

**select  
cname  
from account a,depositor d  
where a.accoutnumber=d.accountnumber  
and balance=(select  
 max(balance)  
 from account);**

**select   
Customer name   
from borrower b ,loan l  
where b.loannumber=l.loannumber   
Group by b.branchname   
Having b.amout=(select  
Max(amout)  
From loan  
Group by branchname**

**Union**

**Select   
Customername  
From deposit;**

**select**

**customername**

**From borrower b, loan l, (select**

**branchname, max(amount) ma**

**from loan**

**group by branchname) t**

**where b.loannumber=l.loannumber and t.branchname-l.branchname**

**union**

**select**

**customername**

**from deposit;**

**iv. Find all customers who have a loan at Newyork bank and whose**

**name is neither smith nor jones**

**select**

**Customer name**

**from borrower b ,loan l**

**where b.loannumber=l.loannumber   
And l.branchname='newyork'  
And customer name not in ('jones','smith');**

**v. Find all the customers who reside outside Newyork and having both**

**Loan and Deposit at Newyork branch.**

**Customer (customername, customerstreet, customercity)**

**Loan (Branchname, loan number, Amount)**

**Borrower (customername, loan number)**

**Account (Branchname, Account number, balance)**

**Depositor (customername, Account number)**

**select**

**c.customername**

**From customer c, loan l, borrower b, account a, depositor d**

**Where l.loannumer=b.loannumber**

**And a.accountnumber=d.accountnumber**

**And c.customername=b.customername**

**And c.customername=d.customername**

**And c.customercity !=’NewYork’**

**And l.branchname=’NewYork’ and a.branchname=’NewYork’;**

1. **Write the relational algebra, the tuple relational calculus expressions for the following relational database queries?**

**Sailor (sailor id, Boat id, sailorname, rating, age)**

**Reserves (Sailor id, Boat id, Day)**

**Boat (boat id, Boatname, color)**

1. **Find the sailorname, age of the youngest sailor?**

**Select**

**sailorname, age**

**from sailor**

**where age=(select**

**min(age)**

**from sailor);**

1. **Find the sailorname, color of the boat reserved of those sailors whose age is more than 35**

**select**

**sailorname, color**

**from sailor s, boat b**

**where s.boat id=b.boat id**

**and s.age >35;**

1. **Find the Number of reservations for each red boat on a particular day?**

**select   
bid,day,Count(bid)  
From reserves r,boats b   
Where r bid=**[**b.bid**](http://b.bid) **B.color =red  
Group by bid,day;**

1. **Find the average age of sailor for each rating level**

**select**

**rating , avg(age)**

**from sailor**

**group by rating;**

1. **Using the relational schema:**

**STUDENT (Stname, Stnum, Totalmarks, Average, Year, Semester)**

**HOSTEL (Stnum, Roomnum)**

**Represent the following queries in SQL.**

1. **Get the student details such as name and total marks of 2nd year 2nd semester Students**

**Select**

**stname,totalmarks**

**from student**

**where year=2 and semester=2;**

1. **Obtain the room number allotted to “Kumar”**

**select**

**roomnum**

**from STUDENT s, HOSTEL h**

**where s.stnum=h.stnum and s.stname='Kumar';**

1. **Obtain the names of students staying in room number 101 and secured more than 60% of marks.**

**select**

**stname**

**from student**

**where stnum in (select**

**stnum**

**from hostel**

**where roomnum=101 )**

**and average>60%;**

1. **Get the name and marks of students who are in the room number 101 in ascending order of name.**

**select**

**stname, Totalmarks**

**from student**

**where stnum in (select**

**stnum**

**from hostel**

**where roomnum=101 )**

**order by stname;**

1. **For the relational schema:**

**Emp(eid: integer, ename: string, age: integer, salary: real)**

**Works(eid: integer, did: integer, pcttime: integer)**

**Dept(did: integer, dname: string, budget: real, managerid: integer)**

**Answer the following:**

**i) Give an example of a foreign key constraint that involves the Dept relation. What are the**

**options for enforcing this constraint when a user attempts to delete a Dept tuple?**

**ii) Write the SQL statements required to create the preceding relations, including**

**appropriate versions of all primary and foreign key integrity constraints.**

**iii) Define the Dept relation in SQL so that every department is guaranteed to have a**

**manager.**

**iv) Write an SQL statement to add John Doe as an employee with eid = 101, age = 32 and salary = 15, 000.**

1. **Consider the following schema**

**Suppliers (sid:integer, sname:string, address:string)**

**Parts (pid: integer, pname:string, color:string)**

**Catalog (sid:integer, pid:integer, cost:real)**

**The catalog relation lists the prices charged for parts by suppliers. Write the following queries**

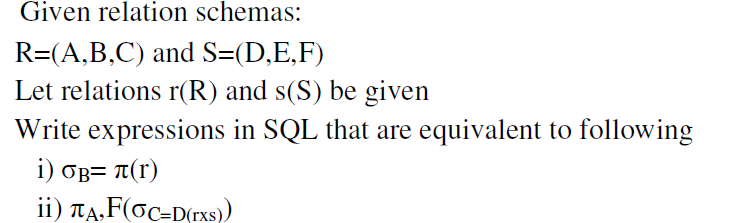
**in relational algebra, Tuple relational calculus.**

**i) Find the sid of suppliers who supply same red part and same green part?**

**ii) Find the names of suppliers who supply same red part?**

**iii) Find the sid of suppliers who supply every part?**

1. **…..**

****

1. **Select \* from R where B=x;**
2. **Select A, F from R, S where C=D;**
3. **Using the following schema answer the following queries in Relational Algebra:**

**Suppliers (sid, sname, saddress)**

**Parts (pid, pname, color)**

**Catalog (sid, pid, cost)**

1. **List the the names of suppliers who supply blue part**

**select**

**sname**

**from suppliers s, Parts p, catalog c**

**where s.sid=c.sid and p.pid=c.pid**

**and color='blue';**

1. **List the supplier details who supply “Bolts” colored “Red”**

**Select   
Sname   
From supplier s ,parts p,catalog c  
Where s.sid=c Sid And c.pid=p.pid   
And p.pname =bolt   
And p.color =red;**

1. **List the supplier details and part details of those whose cost is less than $30.**

**select  
s.\*,p.\*  
from supplier s,parts p,catalog c  
where s.sid=c.sid and p.pid=c.pid  
and c.cost<30;**

1. **List the part details that are supplied by supplier whose name starts with “S”.**

**select**

**p.pid, pname**

**from suppliers s,parts p,catalog c**

**where s.sid=c.sid and p.pid=c.pid**

**and s.sname like 'S%';**

**select**

**\***

**from parts**

**where pid in (select**

**pid**

**from catalog**

**where sid in (select**

**sid**

**from suppliers**

**where sname='s%'));**

1. **Using the following schema answer the following queries in SQL:**

**Suppliers(sid, sname, saddress)**

**Parts(pid, pname, color)**

**Catalog(sid, pid, cost)**

**i) List the the names of suppliers who supply blue part**

**ii) List the supplier details who supply “Bolts” colored “Red”**

**iii)List the supplier details and part details of those whose cost is less than $30.**

**iv)List the part details that are supplied by supplier whose name starts with “B”**

1. **List the parts that are supplied by supplier “HMT” whose cost is more than 5000.**

**select  
p.\*  
from supplier s,parts p,catalog c  
where s.sid=c.sid and p.pid=c.pid  
and s.sname='HMT'  
and c.cost>5000;**

1. **Using the following schema write the queries using Relational Algebra, Relational Calculus:**

**Sailors(sailor-id, sailor-name, sailor-rating, sailor-age)**

**Boats(boat-id, boat-name, boat-color)**

**Reserves (sailor-id, boat-id, booked\_date)**

1. **List all the boat details whose color is “Red”**

**select**

**\***

**from boats**

**where boats-color="Red";**

1. **List the Sailor details whose have not booked a boat on “10/10/2009”**

**select**

**s.\***

**from sailors s, reserves r  
where s.sid=r.sid and**

**booked\_date not like '10/10/2009’;**

**select**

**\***

**from sailors**

**where sailor-id NOT IN (select**

**sailor-id**

**from reserves**

**where booked\_date like ‘10/10/2009’);**

1. **List the Sailor details who are aged more than 50.**

**select**

**\***

**from sailors  
where sailor-age>50;**

1. **List the Sailors who booked Red boat on “10/10/2009”**

**select**

**sname**

**from Sailors s,boats b,Reserves r**

**where s.sid=r.sid and** [**b.bid**](http://b.bid)**=**[**r.bid**](http://r.bid)

**and r.booked\_date like "10/10/2009"**

**and boat\_color='Red’;**

1. **List the sailor whose rating is “A” and booked a “Red” boat**

**select**

**sailor-name**

**from sailors**

**where sailor-sid in (select**

**sailor-sid**

**from reserves**

**where boat-id in (select**

**boat-id**

**from boats**

**where boat-color ="Red"))**

**and sailor-rating="A" ;**

**select**

**sailor-name   
from sailors s,reserves r,boats b  
where s.sailor-id=r.sailor-id and b.sailor-id=s.sailor-id  
and sailor-rating='A'  
and boat-color='Red';**

1. **Using the following schema represent the following queries using Tuple relational calculus :**

**PROJECT (Projectnum, Project Name, Project Type, Project Manager)**

**EMPLOYEE ( Empnum, Empname)**

**ASSIGNED\_TO (Projectnum, Empnum)**

1. **Find Employee details working on a project name starts with ‘L’**

**select**

**\***

**from employee e, project p,assigned\_to a  
where e.enum=a.enum and a.projectnum=p.projectnum  
and project name like 'L%';**

**select**

**\***

**from employee**

**where empnum in (select**

**empnum**

**from Assigned\_to**

**where projectnum in (select**

**projectnum**

**from project**

**where project name="L%"));**

1. **List all the employee details who are working under project manager “Clevee”**

**select**

**\***

**from employee**

**where empnum in (select**

**empnum**

**from Assigned\_to**

**where projectnum in (select**

**projectnum**

**from project**

**where project manager="clevee"));**

**select**

**\***

**from employee e, project p, assigned\_to a  
where e.enum=a.enum and a.projectnum=p.projectnum  
and project manager=’clevee’;**

**iii. List the employees who are still not assigned with any project.**

**iv. List the employees who are working in more than one project.**

1. **Write SQL expressions for the following relational databases.**

**employee (empno, empname, gender, dob, doj, empadd, emptelno, empsalary, empdept, bonus, dor)**

**i. Find empname from emp schema table who joins at the same date?**

**ii. Find empadd, empname who are working in the same department?**

**iii. Find the salary from emp schema who are earing more than 6000?**

**iv. Find how many records are in table emp schema?**

**v. Find empno, empname, empdept from emp schema where empsalary is b/n.**

1. **For the following relational database, give the expressions in SQL.**

**Sailor schema (sailor id, sailorname, rating, Age)**

**Reserves (sailor id, Boat)id, day)**

**Boat Schema(Boat id, boatname, color)**

**i. find the name and age of the oldest sailors?**

**ii. Count the No.of sailors?**

**Select**

**Count(sailor id)**

**From Sailor;**

1. **Count the No.of different sailors names?**

**select**

**count(distinct(sailor name))**

**from sailor;**

1. **Find the names who are younger than the oldest sailors with rating of 10?**

**select**

**sailorname**

**from sailor**

**where age<(select**

**max(age)**

**from sailor where rating=10)**

**and rating=10;**

1. **For the following relational database write the expressions in SQL. [6+10]**

**flight (flight no, from, to, distance, departs arrives, price)**

**Aircraft (Aid, Aircraftname, cruising range)**

**Certified (Emp cd, Aid)**

**Employee (Emp id, Empname, salary)**

**i. Find the names of pilot whose salary are individually less than price of the cheapest route from ?Los Angles? to Honolulu ?**

**ii. Print the enames of pilots who can operate planes with cursing range greater than 5000 miles, but r not certified on any boxing aircraft?**

**iii. Find the names of pilots certified for some Boeing aircraft?**

**iv. Find the aids of all aircraft that can be used on routes Los Angles to**

**Chicago?**

**v. Compute the difference b/n the average salary of a pilot and the average salary of all employees?**