a) Create the above tables and insert data as provided.

```
create table PROJECT ( code CHAR(4), type CHAR(2), title
VARCHAR(50),value INTEGER,   sponsor VARCHAR(50),pi INTEGER , CONSTRAINT
pr_pk PRIMARY KEY(code));

INSERT INTO PROJECT VALUES ('VSS1', 'S', 'VALIDATION OF CONTROL SYSTEMS',
2000000,'RLY', 1001);
INSERT INTO PROJECT VALUES ('SACP', 'S', 'SIDE CHANNEL ATTACKS', 5000000,
'DMI', 1002);
INSERT INTO PROJECT VALUES ('IALO', 'C', 'INDUSTRIAL AUTOMATION', 200000,
'BHL', 1003);
INSERT INTO PROJECT VALUES ('OCDE', 'C', 'OPERATIONS RESEARCH', 800000,
'HII', 1009);
INSERT INTO PROJECT VALUES ('BBDP', 'S', 'BLACK BODY RADIATION', 6000000,
'BHL', 1007);
INSERT INTO PROJECT VALUES ('SRC1', 'S', 'SOLID STATE DEVICES', 8000000,
'HAL', 1008);
INSERT INTO PROJECT VALUES ('LPDC', 'S', 'LOW POWER CIRCUITS', 3000000,
'DMI', 1006);
INSERT INTO PROJECT VALUES ('DCA1', 'C', 'SELF DRIVING CAR', 500000,
'GIL', 1003);
INSERT INTO PROJECT VALUES ('BHSM', 'S', 'BRIDGE HEALTH MONIROTING',
4000000,'RLY', 1004);
```

```
+------+------+-------------------------------+---------+---------+------
+
| code | type | title                         | value   | sponsor | pi
|
+------+------+-------------------------------+---------+---------+------
+
| BBDP | S    | BLACK BODY RADIATION          | 6000000 | BHL     | 1007
|
| BHSM | S    | BRIDGE HEALTH MONIROTING      | 4000000 | RLY     | 1004
|
| DCA1 | C    | SELF DRIVING CAR              |  500000 | GIL     | 1003
|
| IALO | C    | INDUSTRIAL AUTOMATION         |  200000 | BHL     | 1003
|
| LPDC | S    | LOW POWER CIRCUITS            | 3000000 | DMI     | 1006
|
| OCDE | C    | OPERATIONS RESEARCH           |  800000 | HII     | 1009
|
| SACP | S    | SIDE CHANNEL ATTACKS          | 5000000 | DMI     | 1002
|
| SRC1 | S    | SOLID STATE DEVICES           | 8000000 | HAL     | 1008
|
| VSS1 | S    | VALIDATION OF CONTROL SYSTEMS | 2000000 | RLY     | 1001
|
+------+------+-------------------------------+---------+---------+------
+
```

```
create table FACULTY(ec INTEGER, name VARCHAR(20), dept VARCHAR(10),dob
DATE,CONSTRAINT fa_pk PRIMARY KEY(ec));

INSERT INTO FACULTY VALUES (1001, 'BRYANT R.', 'CSE', '1975-01-28');
INSERT INTO FACULTY VALUES (1002, 'GIBBONS P.', 'CSE', '1979-02-15');
INSERT INTO FACULTY VALUES (1003, 'MANKOFF J.', 'MECH', '1972-02-07');
INSERT INTO FACULTY VALUES (1004, 'ROHDE G.', 'ECE', '1976-12-28');
INSERT INTO FACULTY VALUES (1005, 'MILLER G.', 'CSE', '1970-11-25');
INSERT INTO FACULTY VALUES (1006, 'KINGSFORD C.', 'EE', '1968-05-21');
INSERT INTO FACULTY VALUES (1007, 'BELLOCH G.', 'EE', '1980-03-03');
INSERT INTO FACULTY VALUES (1008, 'DANENBERG R.', 'ECE', '1972-04-16');
INSERT INTO FACULTY VALUES (1009, 'PAVLO A.', 'MECH', '1965-01-02');
INSERT INTO FACULTY VALUES (1010, 'GARLAN D.', 'CSE', '1980-05-16');
```

```
+------+--------------+------+------------+
| ec   | name         | dept | dob        |
+------+--------------+------+------------+
| 1001 | BRYANT R.    | CSE  | 1975-01-28 |
| 1002 | GIBBONS P.   | CSE  | 1979-02-15 |
| 1003 | MANKOFF J.   | MECH | 1972-02-07 |
| 1004 | ROHDE G.     | ECE  | 1976-12-28 |
| 1005 | MILLER G.    | CSE  | 1970-11-25 |
| 1006 | KINGSFORD C. | EE   | 1968-05-21 |
| 1007 | BELLOCH G.   | EE   | 1980-03-03 |
| 1008 | DANENBERG R. | ECE  | 1972-04-16 |
| 1009 | PAVLO A.     | MECH | 1965-01-02 |
| 1010 | GARLAN D.    | CSE  | 1980-05-16 |
+------+--------------+------+------------+
```

```
create table INVESTIGATOR ( code CHAR(4), ec INTEGER, CONSTRAINT FOREIGN
KEY in1_fk(code) REFERENCES PROJECT(code), CONSTRAINT FOREIGN KEY
in2_fk(ec) REFERENCES FACULTY(ec));
```

```
INSERT INTO INVESTIGATOR VALUES ('VSS1', 1001);
INSERT INTO INVESTIGATOR VALUES ('VSS1', 1002);
INSERT INTO INVESTIGATOR VALUES ('SACP', 1002);
INSERT INTO INVESTIGATOR VALUES ('SACP', 1010);
INSERT INTO INVESTIGATOR VALUES ('IALO', 1003);
INSERT INTO INVESTIGATOR VALUES ('IALO', 1007);
INSERT INTO INVESTIGATOR VALUES ('OCDE', 1009);
INSERT INTO INVESTIGATOR VALUES ('BBDP', 1007);
INSERT INTO INVESTIGATOR VALUES ('SRC1', 1008);
INSERT INTO INVESTIGATOR VALUES ('SRC1', 1001);
INSERT INTO INVESTIGATOR VALUES ('LPDC', 1006);
INSERT INTO INVESTIGATOR VALUES ('LPDC', 1010);
INSERT INTO INVESTIGATOR VALUES ('DCA1', 1003);
INSERT INTO INVESTIGATOR VALUES ('BHSM', 1004);
INSERT INTO INVESTIGATOR VALUES ('BHSM', 1001);
```

```
+------+------+
| code | ec   |
+------+------+
| VSS1 | 1001 |
| VSS1 | 1002 |
| SACP | 1002 |
| SACP | 1010 |
| IALO | 1003 |
| IALO | 1007 |
| OCDE | 1009 |
| BBDP | 1007 |
| SRC1 | 1008 |
| SRC1 | 1001 |
| LPDC | 1006 |
| LPDC | 1010 |
| DCA1 | 1003 |
| BHSM | 1004 |
| BHSM | 1001 |
+------+------+
```

```
create table STAFF(ec INTEGER ,name VARCHAR(50),code CHAR(4),dob
DATE,salary INTEGER,CONSTRAINT st_fk FOREIGN KEY(code) REFERENCES
PROJECT(code));

INSERT INTO STAFF VALUES (2001, 'ROBBIE KEANE', 'VSS1', '1990-05-16',
18000);
INSERT INTO STAFF VALUES (2002, 'ANSUMAN BANERJEE', 'VSS1', '1985-05-21',
18000);
INSERT INTO STAFF VALUES (2003, 'OMAR GONZALEZ', 'SACP', '1989-04-16',
20000);
INSERT INTO STAFF VALUES (2004, 'ALAN GORDON', 'IALO', '1989-05-03',
20000);
INSERT INTO STAFF VALUES (2005, 'ROBBIE KEANE', 'IALO', '1988-01-16',
18000);
INSERT INTO STAFF VALUES (2006, 'CHANDLER HOFFMAN', 'BBDP', '1988-07-17',
22000);
INSERT INTO STAFF VALUES (2007, 'PAUL POGBA', 'BHSM', '1990-08-16',
18000);
INSERT INTO STAFF VALUES (2008, 'SHINJI KAGAWA', 'LPDC', '1991-01-20',
18000);
```

```
+------+------------------+------+------------+--------+
| ec   | name             | code | dob        | salary |
+------+------------------+------+------------+--------+
| 2001 | ROBBIE KEANE     | VSS1 | 1990-05-16 |  18000 |
| 2002 | ANSUMAN BANERJEE | VSS1 | 1985-05-21 |  18000 |
| 2003 | OMAR GONZALEZ    | SACP | 1989-04-16 |  20000 |
| 2004 | ALAN GORDON      | IALO | 1989-05-03 |  20000 |
| 2005 | ROBBIE KEANE     | IALO | 1988-01-16 |  18000 |
| 2006 | CHANDLER HOFFMAN | BBDP | 1988-07-17 |  22000 |
| 2007 | PAUL POGBA       | BHSM | 1990-08-16 |  18000 |
| 2008 | SHINJI KAGAWA    | LPDC | 1991-01-20 |  18000 |
+------+------------------+------+------------+--------+
```

b) Sorting the order of appearance of results (using ORDER BY clause)

1) List the codes of consultancy projects in descending order of project value.

```
      select code from PROJECT order by value;
```

2) List all projects with title containing the word VLSI in ascending order of value.

```
      select * from PROJECT where title like '%VLSI%' order by value asc;
```

3) List all faculty members in ascending order of Dept, and within each category in ascending order of age.

```
      select * from FACULTY order by dept asc, dob asc;
```

4) List all staff members born between Jan 1, 1980 and Dec 31, 1985 in ascending order of name.

```
      select * from STAFF where dob between '1980-01-01' and '1985-12-31'
order by name asc;
```

c) Using group functions.

1) List the total number of staff working in projects.

```
      select count(*) as total_number_of_staff from STAFF;
```

2) List the total number of faculty members in the CSE, ECE and EE departments taken together.

```
      select count(*) as total_number_of_faculty_members from FACULTY
where dept != 'MECH';
```

3) List each project code with the number of staff in that project.

```
      select code,count(ec) from PROJECT natural left join STAFF group by
code;
```

4) List the maximum, minimum and average salary of staff working in the project having code MAFV.

```
      select code,avg(salary) as average_salary,max(salary) as
maximum_salary,min(salary) as minimum_salary from STAFF group by code
having code = 'MAFV';
```

5) List the sponsor names, and the maximum, minimum and average values of the projects sponsored by each sponsor.

```
      select sponsor ,max(value) as maximum,min(value) as
minimum,avg(value)as average from PROJECT group by sponsor;
```

6) List the codes of projects having less than 3 staff members.

```
      select code,count(ec) as number_of_staff from PROJECT natural left
join STAFF group by code having number_of_staff < 3;
```

7) List the top five projects with the highest value.

```
      select * from PROJECT order by value desc limit 5;
```

8) List the staff members earning more than the average salary.

```
       select * from STAFF where salary > (select avg(salary) from
STAFF);
```

d) Retrieving information from multiple tables

1) List the titles of projects where the PI is from the CSE department.

```
      select title from PROJECT where pi in(select ec from FACULTY where
dept = 'CSE');
```

2) List the names of staff members of projects for which the PI is from the CSE dept.

```
       select distinct name from STAFF where code in (select code from
PROJECT where pi in(select ec from FACULTY where dept = 'CSE'));
```

3) List the names of faculty members who are investigating projects having titles with the word VLSI in it.

```
      select name from PROJECT natural join INVESTIGATOR natural join
FACULTY where code in (select code from PROJECT where title like
'%VLSI%');
```

4) List the names of faculty members indicating whether they are investigating sponsored projects, consultancy projects, both, or none.

```
      select FACULTY.name, case when name in(select distinct name from
FACULTY where ec in(select ec from INVESTIGATOR where code in(select code
from PROJECT where type = 'S')) and name not in(select distinct name from
FACULTY where ec in(select ec from INVESTIGATOR where code in(select code
from PROJECT where type = 'C'))))then 'S' when name in(select distinct
name from FACULTY where ec in(select ec from INVESTIGATOR where code
in(select code from PROJECT where type = 'C')) and name not in(select
distinct name from FACULTY where ec in(select ec from INVESTIGATOR where
code in(select code from PROJECT where type = 'S')))) then 'C' when name
in (select distinct name from FACULTY where ec in(select ec from
INVESTIGATOR where code in(select code from PROJECT where type = 'C'))
and name in(select distinct name from FACULTY where ec in(select ec from
INVESTIGATOR where code in(select code from PROJECT where type = 'S'))))
then 'BOTH' when name in (select distinct name from FACULTY where name
not in(select distinct name from FACULTY where ec in(select ec from
INVESTIGATOR where code in(select code from PROJECT where type = 'C')) or
name in(select distinct name from FACULTY where ec in(select ec from
INVESTIGATOR where code in(select code from PROJECT where type = 'S')))))
then 'NOT IN ANY' end as status from FACULTY;
```

5) List the names of staff members who are working in the same project as ANSUMAN BANERJEE.

```
      select name from STAFF where code in (select code from STAFF where
name = 'ANSUMAN BANERJEE') and name != 'ANSUMAN BANERJEE';
```

e) Using set operations

1) List the names of faculty members who are investigating sponsored projects, but not any consultancy projects.

```
      select distinct name from PROJECT natural join INVESTIGATOR natural
join FACULTY where type = 'S' and name not in (select distinct name from
PROJECT natural join INVESTIGATOR natural join FACULTY where type = 'C');
```

2) List the names of faculty members who are investigating both sponsored projects as well as consultancy projects.

```
      select distinct name from PROJECT natural join INVESTIGATOR natural
join FACULTY where type = 'S' and name in (select distinct name from
PROJECT natural join INVESTIGATOR natural join FACULTY where type = 'C');
```

3) List the names of faculty and staff members in projects sponsored by INTEL.

```
       (select name from FACULTY where ec in (select pi from PROJECT
where sponsor = 'INTEL')) union (select name from STAFF where code in
(select code from PROJECT where sponsor = 'INTEL'));
```

f) SQL sub-queries

1)List the names of sponsor who have projects under PIs from all departments.

```
      select distinct S.sponsor from PROJECT as S where not exists(select
distinct dept from FACULTY where dept not in (select dept from FACULTY
where ec in (select pi from PROJECT as T  where T.sponsor = S.sponsor)));
```

2)List the projects where the youngest faculty member is a PI.

```
      select title from PROJECT where pi in(select ec from FACULTY where
dob = (select max(dob) from FACULTY));
```

3)List the departments from which no one is a PI of a consultancy project.

```
      select distinct S.dept from FACULTY as S where not exists(select ec
from FACULTY where dept = S.dept and ec in (select pi from PROJECT));
```

4)List the department having the maximum total value of projects under its faculty members (as PIs).

```
      a)create view Total as select dept,sum(value) as total_sum from
PROJECT join FACULTY where ec = pi group by dept;
      b)select dept from Total where total_sum = (select max(total_sum)
from Total);
```

5) List the names of faculty members in each department who are not associated with any projects.

```
      select name from FACULTY where ec not in (select ec from
INVESTIGATOR);
```

6) List the names of projects that involve all faculty members of a department.

```
      a)create view PNINF as select * from PROJECT natural join
INVESTIGATOR natural join FACULTY;
      b)select distinct S.code,S.dept from PNINF as S where not
exists(select ec from PNINF where dept = S.dept and ec not in (select
T.ec from PNINF as T where T.code = S.code));
```

7) List the departments where the oldest faculty member (in the dept) is also associated with the maximum number of projects (among other faculty members in the same dept).

```
      a)create view fac_data as (select ec,dept,dob from FACULTY);
      b)create view individual as (select dept,dob,count(*) as
ind_pro_count from fac_data as T,INVESTIGATOR where T.ec=INVESTIGATOR.ec
group by T.dept,T.ec);
      c)create view elderly as (select dept,min(dob) as eldest from
FACULTY,INVESTIGATOR where FACULTY.ec=INVESTIGATOR.ec group by dept);
      d)create view professional as(select T.ec,dept,dob,count(*) as
proCount from fac_data as T,INVESTIGATOR where T.ec=INVESTIGATOR.ec group
by T.dept,T.ec);
      e)create view eldest_person as (select TT.dept,eldest,YY.procount
as eld_pro_count from elderly as TT,professional as YY where
YY.dept=TT.dept and  eldest=dob);
      f)create view not_in as (select A.dept from individual as A,
eldest_person as B where A.dept=B.dept and A.dob!=B.eldest and
A.ind_pro_count>=B.eld_pro_count);

      g)select distinct dept from FACULTY where dept not in (select
distinct dept from not_in);
```

g) Application of Integrity constraints

1) Delete all the relevant records from the affected tables when a faculty with employee code EC = 1001 leaves the institute.

```
      a)delete from INVESTIGATOR where ec = 1001;
      b)delete from FACULTY where ec = 1001;
      c)update PROJECT set pi = null where pi = 1001;
```

2)Set the value to NULL for the affected columns of relevant records from the affected tables when a project with project code Code = 'VSS1' is terminated.

```
      a)delete from INVESTIGATOR where code = 'VSS1';
      b)update STAFF set code = null where code = 'VSS1';
      c)delete from PROJECT where code = 'VSS1';
```

3)Update the PI for the project with project code Code = 'SRC1' to 1012.

```
      update PROJECT set pi = 1012 where code = 'SRC1';
```

4)Apply CHECK constraints at the column level to ensure that value of any project is not less than Rs. 100,000/-.

```
      alter table PROJECT add check (value > 100000);
```

5)Apply CHECK constraints at the column level to ensure that every staff working on projects is at least 18 years old. (use system date)

```
      alter table STAFF add check (year(dob) < year(curdate())-18);
```