PROGRAM - 1

Consider the following schema for a Library Database:

BOOK (Book_ID, Title, Pub_Name, Pub_Year)

BOOK_AUTHORS (Book_ID, Author_Name)

PUBLISHER (Pub_ID, Pub_Name, Address, Phone)

BOOK_COPIES (Book_ID, PGM_ID, No_of_Copies)

BOOK_LENDING (Book_ID, PGM_ID, Card_No, Date_Out, Due_Date)

LIBRARY_PROGRAM (PGM_ID, PGM_Name, Address)

Write SQL QUERIES to:

1) Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each program, etc.
2) Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017.
3) Delete a book in the BOOK table. Update the contents of other tables to reflect this data manipulation operation.
4) Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.
5) Create a view of all books and its number of copies that are currently available in the library.

Solution Queries:

Query to create tables:

- Publisher Table –

```
CREATE TABLE PUBLISHER(
PUB_ID INT PRIMARY KEY,
PUB_NAME VARCHAR(20) UNIQUE,
ADDRESS VARCHAR(20),
PHONE INT);
```

- Book Table –

```
CREATE TABLE BOOK(
BOOK_ID INT PRIMARY KEY,
TITLE VARCHAR(20),
PUB_NAME VARCHAR(20),
FOREIGN KEY (PUB_NAME) REFERENCES PUBLISHER (PUB_NAME) ON DELETE
CASCADE,
PUB_YEAR INT);
```

- Library Program Table –

```
CREATE TABLE LIBRARY_PROGRAM(
PGM_ID INT PRIMARY KEY,
PGM_NAME VARCHAR(20),
ADDRESS VARCHAR(30));
```

- Book Authors Table –

```
CREATE TABLE BOOK_AUTHORS(
BOOK_ID INT,
AUTHOR_NAME VARCHAR(20),
PRIMARY KEY(BOOK_ID, AUTHOR_NAME),
FOREIGN KEY(BOOK_ID) REFERENCES BOOK(BOOK_ID) ON DELETE CASCADE);
```

- Book Copies Table –

```
CREATE TABLE BOOK_COPIES(
BOOK_ID INT,
PGM_ID INT,
NO_OF_COPIES INT,
PRIMARY KEY(BOOK_ID, PGM_ID),
FOREIGN KEY (BOOK_ID) REFERENCES BOOK(BOOK_ID) ON DELETE CASCADE,
FOREIGN KEY (PGM_ID) REFERENCES LIBRARY_PROGRAM(PGM_ID) ON DELETE
CASCADE);
```

- Book Lending Table –

```
CREATE TABLE BOOK_LENDING(
BOOK_ID INT,
PGM_ID INT,
CARD_NO INT,
DATE_OUT DATE,
DUE_DATE DATE,
PRIMARY KEY(BOOK_ID,PGM_ID,CARD_NO),
FOREIGN KEY (BOOK_ID) REFERENCES BOOK(BOOK_ID) ON DELETE CASCADE,
FOREIGN KEY (PGM_ID) REFERENCES LIBRARY_PROGRAM(PGM_ID) ON DELETE
CASCADE);
```

Query to insert values into the table:

- Publisher table –

```
INSERT INTO PUBLISHER VALUES (501,'KVS','BANGALORE',9535616745);
INSERT INTO PUBLISHER VALUES (502,'WESTLAND','PUNE',8768916745);
INSERT INTO PUBLISHER VALUES (503,'RUPA','BANGALORE',6478989715);
INSERT INTO PUBLISHER VALUES (504,'GANGA','MUMBAI',9876985645);
INSERT INTO PUBLISHER VALUES (505,'HACHETTE','MATTUR',7013458745);
```

PUB_ID	PUB_NAME	ADDRESS	PHONE
501	KVS	BANGALORE	9535616745
502	WESTLAND	PUNE	8768916745
503	RUPA	BANGALORE	6478989715
504	GANGA	MUMBAI	9876985645
505	HACHETTE	MATTUR	7013458745

- Book Table –

```
INSERT INTO BOOK VALUES (001,'MCGRAW-HILL','GANGA',2001);
INSERT INTO BOOK VALUES (002,'MY ARTEMIS','KVS',2004);
INSERT INTO BOOK VALUES (003,'CHEMISTRY VOL 1','WESTLAND',2006);
INSERT INTO BOOK VALUES (004,'UPRISING','RUPA',2018);
INSERT INTO BOOK VALUES (005,'CHEMISTRY VOL 2','WESTLAND',2021);
```

BOOK_ID	TITLE	PUBLISHER_NAME	PUB_YEAR
1	MCGRAW-HILL	GANGA	2001
2	MY ARTEMIS	KVS	2004
3	CHEMISTRY VOL 1	WESTLAND	2006
4	UPRISING	RUPA	2018
5	CHEMISTRY VOL 2	WESTLAND	2021

- Library Program Table –

```
INSERT INTO LIBRARY_PROGRAM VALUES (101,'BOOK AXIS','BANGALORE');
INSERT INTO LIBRARY_PROGRAM VALUES (102,'BOOK SQUARE','PUNE');
INSERT INTO LIBRARY_PROGRAM VALUES (103,'CLAUS BOOKS','MUMBAI');
INSERT INTO LIBRARY_PROGRAM VALUES (104,'COMIC CON','PUNE');
INSERT INTO LIBRARY_PROGRAM VALUES (105,'FANDOM','BANGALORE');
```

PGM_ID	PGM_NAME	ADDRESS	PGM_ID
101	BOOK AXIS	BANGALORE	101
102	BOOK SQUARE	PUNE	102
103	CLAUS BOOKS	MUMBAI	103
104	COMIC CON	PUNE	104
105	FANDOM	BANGALORE	105

- Book Authors Table –

```
INSERT INTO BOOK_AUTHORS VALUES (001, 'ASHISH C');
INSERT INTO BOOK_AUTHORS VALUES (002, 'ANEESHA');
INSERT INTO BOOK_AUTHORS VALUES (003, 'ADITYA KUL C');
INSERT INTO BOOK_AUTHORS VALUES (004, 'SAQUIB M');
INSERT INTO BOOK_AUTHORS VALUES (005, 'ARJUN S');
```

BOOK_ID	AUTHOR_NAME
1	ASHISH C
2	ANEESHA
3	ADITYA KUL C
4	SAQUIB M
5	ARJUN S

- Book Copies Table –

INSERT INTO BOOK_COPIES VALUES (001, 102, 40);
INSERT INTO BOOK_COPIES VALUES (002, 101, 18);
INSERT INTO BOOK_COPIES VALUES (003, 104, 53);
INSERT INTO BOOK_COPIES VALUES (004, 103, 4);
INSERT INTO BOOK_COPIES VALUES (005, 105, 20);

BOOK_ID	PGM_ID	NO_OF_COPIES
1	102	40
2	101	18
3	104	53
4	103	4
5	105	20

- Book Lending Table –

INSERT INTO BOOK_LENDING VALUES (001, 101, 5001, '21-SEP-2021', '19-OCT-2021');
INSERT INTO BOOK_LENDING VALUES (001, 102, 5002, '07-JAN-2017', '18-MAY-2017');
INSERT INTO BOOK_LENDING VALUES (002, 102, 5003, '02-FEB-2017', '22-MAR-2020');
INSERT INTO BOOK_LENDING VALUES (003, 103, 5004, '14-SEP-2016', '08-OCT-2021');
INSERT INTO BOOK_LENDING VALUES (005, 104, 5005, '18-JUN-2020', '14-AUG-2021');
INSERT INTO BOOK_LENDING VALUES (002, 102, 5002, '07-JAN-2017', '18-MAY-2017');
INSERT INTO BOOK_LENDING VALUES (003, 102, 5002, '02-FEB-2017', '22-MAR-2020');
INSERT INTO BOOK_LENDING VALUES (004, 102, 5002, '14-MAR-2017', '08-MAY-2019');

BOOK_ID	PGM_ID	CARD_NO	DATE_OUT	DUE_DATE
1	101	5001	21-SEP-21	19-OCT-21
1	102	5002	07-JAN-17	18-MAY-17
2	102	5003	02-FEB-17	22-MAR-20
3	103	5004	14-SEP-16	08-OCT-21
5	104	5005	18-JUN-20	14-AUG-21
2	102	5002	07-JAN-17	18-MAY-17
3	102	5002	02-FEB-17	22-MAR-20
4	102	5002	14-MAR-17	08-MAY-19

Query for given questions:

1) SELECT LP.PGM_NAME, B.BOOK_ID,TITLE, PUB_NAME, AUTHOR_NAME, NO_OF_COPIES
FROM BOOK B, BOOK_AUTHORS BA, BOOK_COPIES BC, LIBRARY_PROGRAM LP, PUBLISHER P
WHERE B.BOOK_ID = BA.BOOK_ID AND
BA.BOOK_ID = BC.BOOK_ID AND
BC.PGM_ID = LP.PGM_ID
GROUP BY LP.PGM_NAME, B.BOOK_ID, TITLE,
PUB_NAME, AUTHOR_NAME, NO_OF_COPIES;

PGM_NAME	BOOK_ID	TITLE	PUB_NAME	AUTHOR_NAME	NO_OF_COPIES
CLAUS BOOKS	4	UPRISING	KVS	SAQUIB M	4
COMIC CON	3	CHEMISTRY VOL 1	GANGA	ADITYA KUL C	53
FANDOM	5	CHEMISTRY VOL 2	KVS	ARJUN S	20
FANDOM	5	CHEMISTRY VOL 2	GANGA	ARJUN S	20
BOOK AXIS	2	MY ARTEMIS	RUPA	ANEESHA	18

2) SELECT CARD_NO FROM BOOK_LENDING
WHERE DATE_OUT > '01-JAN-2017' AND
DATE_OUT < '01-JUN-2017'
GROUP BY CARD_NO
HAVING COUNT (*) > 3;

CARD_NO
5002

3) DELETE FROM BOOK WHERE BOOK_ID = 001;

BOOK_ID	TITLE	PUBLISHER_NAME	PUB_YEAR
2	MY ARTEMIS	KVS	2004
3	CHEMISTRY VOL 1	WESTLAND	2006
4	UPRISING	RUPA	2018
5	CHEMISTRY VOL 2	WESTLAND	2021

4) SELECT PUB_YEAR FROM BOOK;

PUB_YEAR
2004
2006
2018
2021

5) CREATE VIEW BOOKS_AVAILABLE AS
SELECT B.BOOK_ID, B.TITLE, C.NO_OF_COPIES
FROM LIBRARY_PROGRAM L, BOOK B, BOOK_COPIES C
WHERE B.BOOK_ID = C.BOOK_ID AND
L.PGM_ID=C.PGM_ID;
SELECT * FROM BOOKS_AVAILABLE

BOOK_ID	TITLE	NO_OF_COPIES
2	MY ARTEMIS	18
3	CHEMISTRY VOL 1	53
4	UPRISING	4
5	CHEMISTRY VOL 2	20

PROGRAM - 2

Consider the schema for College Database:

STUDENT (USN, SName, Address, Phone, Gender)

SEMSEC (SSID, Sem, Sec)

CLASS (USN, SSID)

COURSE (Subcode, Title, Sem, Credits)

IAMARKS (USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)

Write SQL queries to:

1) List all the student details studying in the fourth semester 'C' section.
2) Compute the total number of male and female students in each semester and in each section.
3) Create a view of Test1 marks of student USN '1DS22IS101' in all Courses.
4) Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students.
5) Categorize students based on the following criterion:
    - If FinalIA = 17 to 20 then CAT = 'Outstanding'
    - If FinalIA = 12 to 16 then CAT = 'Average'
    - If FinalIA< 12 then CAT = 'Weak'
    - Give these details only for 8th semester A, B, and C section students.

Solution Queries:

Query to create tables:

- Student Table –

```
CREATE TABLE STUDENT(
USN VARCHAR(20) PRIMARY KEY,
SNAME CHAR(20),
ADDRESS VARCHAR(50),
PHONE DECIMAL(10,0),
GENDER CHAR(10));
```

- SEMSEC Table –

```
CREATE TABLE SEMSEC(
SSID VARCHAR(20) PRIMARY KEY,
SEM INT,
SEC CHAR(2));
```

- Class Table –

```
CREATE TABLE CLASS(
USN VARCHAR(20),
SSID VARCHAR(20),
PRIMARY KEY (USN,SSID),
FOREIGN KEY (USN) REFERENCES STUDENT (USN) ON DELETE
CASCADE,
FOREIGN KEY (SSID) REFERENCES SEMSEC (SSID) ON DELETE
CASCADE);
```

- Course Table –

```
CREATE TABLE COURSE(
SUBCODE VARCHAR(20) PRIMARY KEY,
TITLE VARCHAR(20),
SEM INT,
CREDITS INT);
```

- IAMarks Table –

```
CREATE TABLE IAMARKS(
USN VARCHAR(20),
SUBCODE VARCHAR(20),
SSID VARCHAR(20),
PRIMARY KEY(USN,SUBCODE,SSID),
FOREIGN KEY(USN) REFERENCES STUDENT(USN) ON DELETE
CASCADE,
FOREIGN KEY(SUBCODE) REFERENCES COURSE(SUBCODE) ON
DELETE CASCADE,
FOREIGN KEY(SSID) REFERENCES SEMSEC(SSID) ON DELETE
CASCADE,
TEST1 INT,
TEST2 INT,
TEST3 INT,
FINALIA DECIMAL(5,2));
```

Query to insert values into the table:

- Student Table –

```
INSERT INTO STUDENT VALUES ('1DS19IS049','ARJUN','BANGALORE',
9535616756,'MALE');
INSERT INTO STUDENT VALUES ('1DS19IS022','ADITYA','BANGALORE',
7896716657,'MALE');
```

INSERT INTO STUDENT VALUES ('1DS19IS017','ANEESH','MATTUR', 7898676647,'MALE');
INSERT INTO STUDENT VALUES ('1DS19IS058','VARSHA','PUNE', 8978916756,'FEMALE');
INSERT INTO STUDENT VALUES ('1DS19IS007','RUPA','MANGALORE', 9535616756,'FEMALE');

USN	SNAME	ADDRESS	PHONE	GENDER
1DS19IS049	ARJUN	BANGALORE	9535616756	MALE
1DS19IS022	ADITYA	BANGALORE	7896716657	MALE
1DS19IS017	ANEESH	MATTUR	7898676647	MALE
1DS19IS058	VARSHA	PUNE	8978916756	FEMALE
1DS19IS007	RUPA	MANGALORE	9535616756	FEMALE

- SEMSEC Table –

INSERT INTO SEMSEC VALUES('4A01',4,'A');
INSERT INTO SEMSEC VALUES('4C02',4,'C');
INSERT INTO SEMSEC VALUES('8A01',8,'A');
INSERT INTO SEMSEC VALUES('8C03',8,'C');
INSERT INTO SEMSEC VALUES('5A01',5,'A');

SSID	SEM	SEC
4A01	4	A
4C02	4	C
8A01	8	A
8C03	8	C
5A01	5	A

- Class Table –

INSERT INTO CLASS VALUES('1DS19IS049','8A01');
INSERT INTO CLASS VALUES('1DS19IS022','8A01');
INSERT INTO CLASS VALUES('1DS19IS017','4C02');
INSERT INTO CLASS VALUES('1DS19IS058','4C02');
INSERT INTO CLASS VALUES('1DS19IS007','5A01');

USN	SSID
1DS19IS049	8A01
1DS19IS022	8A01
1DS19IS017	4C02
1DS19IS058	4C02
1DS19IS007	5A01

- Course Table –

```
INSERT INTO COURSE VALUES(1,'DBMS',5,4);
INSERT INTO COURSE VALUES(2,'PHYSICS',8,3);
INSERT INTO COURSE VALUES(3,'CHEMISTRY',8,4);
INSERT INTO COURSE VALUES(4,'CNCS',4,2);
INSERT INTO COURSE VALUES(5,'PP',5,3);
```

SUBCODE	TITLE	SEM	CREDITS
1	DBMS	5	4
2	PHYSICS	8	3
3	CHEMISTRY	8	4
4	CNCS	4	2
5	PP	5	3

- IAMarks Table –

```
INSERT INTO IAMARKS VALUES('1DS19IS049',2,'8A01',19,18,20,0);
INSERT INTO IAMARKS VALUES('1DS19IS058',4,'4C02',18,12,13,0);
INSERT INTO IAMARKS VALUES('1DS19IS022',2,'8A01',14,16,18,0);
INSERT INTO IAMARKS VALUES('1DS19IS007',1,'5A01',18,16,7,0);
INSERT INTO IAMARKS VALUES('1DS19IS017',4,'4C02',2,4,3,0);
```

USN	SUBCODE	SSID	TEST1	TEST2	TEST3	FINALIA
1DS19IS049	2	8A01	19	18	20	0.00
1DS19IS058	4	4C02	18	12	13	0.00
1DS19IS022	2	8A01	14	16	18	0.00
1DS19IS007	1	5A01	18	16	7	0.00
1DS19IS017	4	4C02	2	4	3	0.00

Query for given questions:

1)
```
SELECT S1.* FROM STUDENT S1, SEMSEC S2, CLASS C
WHERE S1.USN = C.USN AND C.SSID = S2.SSID
AND S2.SEM = 4 AND S2.SEC = 'C';
```

USN	SNAME	ADDRESS	PHONE	GENDER
1DS19IS017	ANEESH	MATTUR	7898676647	MALE
1DS19IS058	VARSHA	PUNE	8978916756	FEMALE

2)
```
SELECT S.GENDER, SS.SEM,SS.SEC ,COUNT(GENDER)
FROM STUDENT S, SEMSEC SS,CLASS C
WHERE S.USN = C.USN AND C.SSID = SS.SSID
GROUP BY S.GENDER, SS.SEM,SS.SEC;
```

GENDER	SEM	SEC	COUNT(GENDER)
FEMALE	4	C	1
MALE	8	A	2
MALE	4	C	1
FEMALE	5	A	1

3) CREATE VIEW IAMARKS_1 AS
SELECT SUBCODE , TEST1
FROM IAMARKS WHERE USN = '1DS19IS049';

SELECT * FROM IAMARKS_1;

SUBCODE	TEST1
2	19

4) UPDATE IAMARKS
SET FINALIA = (GREATEST(TEST1, TEST2, TEST3) +
((TEST1+TEST2+TEST3) - GREATEST(TEST1, TEST2, TEST3) -
LEAST(TEST1, TEST2, TEST3)))/2;

SELECT * FROM IAMARKS;

USN	SUBCODE	SSID	TEST1	TEST2	TEST3	FINALIA
1DS19IS049	2	8A01	19	18	20	19.5
1DS19IS058	4	4C02	18	12	13	15.5
1DS19IS022	2	8A01	14	16	18	17
1DS19IS007	1	5A01	18	16	7	17
1DS19IS017	4	4C02	2	4	3	3.5

5) SELECT SS.SEC,S.*,
(CASE WHEN IA.FINALIA BETWEEN 17 AND 20 THEN
'OUTSTANDING'
WHEN IA.FINALIA BETWEEN 12 AND 16 THEN 'AVERAGE'
ELSE 'WEAK' END) AS CAT FROM STUDENT S, SEMSEC SS, IAMARKS
IA, COURSE SUB
WHERE S.USN = IA.USN AND SS.SSID = IA.SSID AND
SUB.SUBCODE = IA.SUBCODE AND SUB.SEM = 8;

SEC	USN	SNAME	ADDRESS	PHONE	GENDER	CAT
A	1DS19IS049	ARJUN	BANGALORE	9535616756	MALE	OUTSTANDING
A	1DS19IS022	ADITYA	BANGALORE	7896716657	MALE	OUTSTANDING

Program - 3

Consider the schema for Company Database:

EMPLOYEE(SSN, Name, Address, Sex, Salary, SuperSSN, DNo)

DEPARTMENT(DNo, DName, MgrSSN, MgrStartDate)

DLOCATION(DNo,DLoc)

PROJECT(PNo, PName, PLocation, DNo)

WORKS_ON(SSN, PNo, Hours)

Write SQL queries to

1) Make a list of all project numbers for projects that involve an employee whose last name is 'Hegde', either as a worker or as a manager of the department that controls the project.
2) Show the resulting salaries if every employee working on the 'AI' project is given a 10 percent raise.
3) Find the sum of the salaries of all employees of the 'Accounts' department, as well as the maximum salary, the minimum salary, and the average salary in this department.
4) Retrieve the name of each employee who works on all the projects controlled by department number 5 (use NOT EXISTS operator).
5) For each department that has more than five employees, retrieve the department number and the number of its employees who are earning more than Rs.6,00,000 per month.

Solution Queries:

Query to create tables:

- Department Table –

```
CREATE TABLE DEPARTMENT (
   DNo INT PRIMARY KEY,
   DName VARCHAR(50),
   MgrSSN CHAR(9),
   MgrStartDate DATE
);
```

- Employee Table –

```
CREATE TABLE EMPLOYEE (
   SSN CHAR(9) PRIMARY KEY,
   Name VARCHAR(50),
   Address VARCHAR(100),
   Sex CHAR(1),
   Salary DECIMAL(10, 2),
   SuperSSN CHAR(9),
   DNo INT,
   FOREIGN KEY (SuperSSN) REFERENCES EMPLOYEE(SSN),
   FOREIGN KEY (DNo) REFERENCES DEPARTMENT(DNo)
);
```

- Dlocation Table –

```
CREATE TABLE DLOCATION (
   DNo INT,
   DLoc VARCHAR(50),
   PRIMARY KEY (DNo, DLoc),
   FOREIGN KEY (DNo) REFERENCES DEPARTMENT(DNo)
);
```

- Project Table –

```
CREATE TABLE PROJECT (
   PNo INT PRIMARY KEY,
   PName VARCHAR(50),
   PLocation VARCHAR(50),
   DNo INT,
   FOREIGN KEY (DNo) REFERENCES DEPARTMENT(DNo)
);
```

- Works_on Table –

```
CREATE TABLE WORKS_ON (
   SSN CHAR(9),
   PNo INT,
   Hours DECIMAL(5, 2),
   PRIMARY KEY (SSN, PNo),
   FOREIGN KEY (SSN) REFERENCES EMPLOYEE(SSN),
   FOREIGN KEY (PNo) REFERENCES PROJECT(PNo)
);
```

Query to insert values into the table:

- Department Table –

INSERT INTO DEPARTMENT VALUES (1, 'HR', '123456789', '01-JAN-2020');
INSERT INTO DEPARTMENT VALUES (2, 'Engineering', '987654321', '15-MAR-2019');
INSERT INTO DEPARTMENT VALUES (3, 'Marketing', '456789123', '10-JUN-2021');
INSERT INTO DEPARTMENT VALUES (4, 'Sales', '789123456', '23-NOV-2018');
INSERT INTO DEPARTMENT VALUES (5, 'Accounts', '321654987', '01-AUG-2017');

DNO	DNAME	MGRSSN	MGRSTARTDATE
1	HR	123456789	01-JAN-20
2	Engineering	987654321	15-MAR-19
3	Marketing	456789123	10-JUN-21
4	Sales	789123456	23-NOV-18
5	Accounts	321654987	01-AUG-17

- Employee Table –

INSERT INTO EMPLOYEE VALUES ('123456789', 'John Doe', '123 Elm St', 'M', 50000, NULL, 1);
INSERT INTO EMPLOYEE VALUES ('987654321', 'Jane Smith', '456 Oak St', 'F', 60000, '123456789', 2);
INSERT INTO EMPLOYEE VALUES ('456789123', 'Robert Hegde', '789 Pine St', 'M', 55000, '987654321', 3);
INSERT INTO EMPLOYEE VALUES ('789123456', 'Maria Johnson', '101 Maple St', 'F', 45000, '456789123', 4);
INSERT INTO EMPLOYEE VALUES ('321654987', 'Michael Hegde', '202 Birch St', 'M', 70000, '789123456', 5);

SSN	NAME	ADDRESS	SEX	SALARY	SUPERSSN	DNO
123456789	John Doe	123 Elm St	M	50000	-	1
987654321	Jane Smith	456 Oak St	F	60000	123456789	2
456789123	Robert Hegde	789 Pine St	M	55000	987654321	3
789123456	Maria Johnson	101 Maple St	F	45000	456789123	4
321654987	Michael Hegde	202 Birch St	M	70000	789123456	5

- Dlocation Table –

INSERT INTO DLOCATION VALUES (1, 'New York');
INSERT INTO DLOCATION VALUES (2, 'San Francisco');
INSERT INTO DLOCATION VALUES (3, 'Los Angeles');

INSERT INTO DLOCATION VALUES (4, 'Chicago');
INSERT INTO DLOCATION VALUES (5, 'Houston');

DNO	DLOC
1	New York
2	San Francisco
3	Los Angeles
4	Chicago
5	Houston

- Project Table –

INSERT INTO PROJECT VALUES (101, 'Project A', 'New York', 1);
INSERT INTO PROJECT VALUES (102, 'Project B', 'San Francisco', 2);
INSERT INTO PROJECT VALUES (103, 'AI', 'Los Angeles', 3);
INSERT INTO PROJECT VALUES (104, 'Project D', 'Chicago', 4);
INSERT INTO PROJECT VALUES (105, 'Project E', 'Houston', 5);

PNO	PNAME	PLOCATION	DNO
101	Project A	New York	1
102	Project B	San Francisco	2
103	AI	Los Angeles	3
104	Project D	Chicago	4
105	Project E	Houston	5

- Works_on Table –

INSERT INTO WORKS_ON VALUES ('123456789', 101, 20);
INSERT INTO WORKS_ON VALUES ('987654321', 102, 25);
INSERT INTO WORKS_ON VALUES ('456789123', 103, 30);
INSERT INTO WORKS_ON VALUES ('789123456', 104, 35);
INSERT INTO WORKS_ON VALUES ('321654987', 105, 40);
INSERT INTO WORKS_ON VALUES ('123456789', 103, 15);
INSERT INTO WORKS_ON VALUES ('456789123', 105, 10);

SSN	PNO	HOURS
123456789	101	20
987654321	102	25
456789123	103	30
789123456	104	35
321654987	105	40
123456789	103	15
456789123	105	10

Query for given questions:

1) SELECT DISTINCT P.PNo
   FROM PROJECT P
   LEFT JOIN WORKS_ON W ON P.PNo = W.PNo
   LEFT JOIN EMPLOYEE E1 ON W.SSN = E1.SSN
   LEFT JOIN DEPARTMENT D ON P.DNo = D.DNo
   LEFT JOIN EMPLOYEE E2 ON D.MgrSSN = E2.SSN
   WHERE E1.Name LIKE '%Hegde%' OR E2.Name LIKE '%Hegde%';

PNO
105
103

2) SELECT E.SSN, E.Name, E.Salary * 1.10 AS NewSalary
   FROM EMPLOYEE E
   JOIN WORKS_ON W ON E.SSN = W.SSN
   JOIN PROJECT P ON W.PNo = P.PNo
   WHERE P.PName = 'AI';

SSN	NAME	NEWSALARY
123456789	John Doe	55000
456789123	Robert Hegde	60500

3) SELECT SUM(E.Salary) AS TotalSalary,
      MAX(E.Salary) AS MaxSalary,
      MIN(E.Salary) AS MinSalary,
      AVG(E.Salary) AS AvgSalary
   FROM EMPLOYEE E
   JOIN DEPARTMENT D ON E.DNo = D.DNo
   WHERE D.DName = 'Accounts';

TOTALSALARY	MAXSALARY	MINSALARY	AVGSALARY
70000	70000	70000	70000

4) SELECT E.Name
   FROM EMPLOYEE E
   WHERE NOT EXISTS (
      SELECT P.PNo
      FROM PROJECT P
      WHERE P.DNo = 5
      AND NOT EXISTS (

```
        SELECT W.SSN
        FROM WORKS_ON W
        WHERE W.PNo = P.PNo AND W.SSN = E.SSN
    )
);
```

NAME
Robert Hegde
Michael Hegde

5)
```
SELECT D.DNo, COUNT(E.SSN) AS NumEmployees
FROM DEPARTMENT D
JOIN EMPLOYEE E ON D.DNo = E.DNo
WHERE E.Salary > 60000
GROUP BY D.DNo
HAVING COUNT(E.SSN) >= 1;
```

DNO	NUMEMPLOYEES
5	1