DBMS ASSIGNMENTS

JADAVPUR UNIVERSITY

Name: ARKA DAS
Roll: 002210503046
Name: MD NAIMUR RAHAMAN
Roll: 002210503040

MCA 1st year 2nd Semester
Session: 2022 - 2024

Assignment: Set - 1

Assignment – 1

Question 1:

Create the following tables:

Tablename: EMPLOYEE
Structure :

EMP_CODE	char(16)
EMP_NAME	char(20)
DEPT_CODE	char(16)
DESIG_CODE	char(16)
SEX	char(1)
ADDRESS	char (25)
CITY	char (20)
STATE	char (20)
PIN	char (6)
BASIC	Number
JN_DT	Date

Primary key is EMP_CODE

Tablename : DESIGNATION
Structure :

DESIG_CODE	char(16)
DESIG_DESC	char(20)

Primary key is DESIG_CODE

Tablename : DEPARTMENT
Structure :

DEPT_CODE	char(16)
DEPT_NAME	char(20)

Primary key is DEPT_CODE.

Solution:

```
CREATE TABLE EMPLOYEE
(
      EMP_CODE varchar(16),
      EMP_NAME varchar(20),
      DEPT_CODE varchar(16),
      DESIG_CODE varchar(16),
      SEX varchar(1),
      ADDRESS varchar(25),
      CITY varchar(20),
      STATE varchar(20),
      PIN varchar(6),
      BASIC int(7),
      JN_DT DATE,
      primary key (EMP_CODE)
);
```

```
CREATE TABLE DESIGNATION
(
      DESIG_CODE varchar(16),
      DESIG_DESC varchar(20),
      primary key (DESIG_CODE)
);

CREATE TABLE DEPARTMENT
(
      DEPT_CODE varchar(16),
      DEPT_NAME varchar(20),
      primary key (DEPT_CODE)
);
```

Question 2:

Display the structure of each table

Solution:

```
desc EMPLOYEE;
MariaDB [assignment_1]> desc EMPLOYEE;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| EMP_CODE   | varchar(16) | NO   | PRI | NULL    |       |
| EMP_NAME   | varchar(20) | YES  |     | NULL    |       |
| DEPT_CODE  | varchar(16) | YES  |     | NULL    |       |
| DESIG_CODE | varchar(16) | YES  |     | NULL    |       |
| SEX        | varchar(1)  | YES  |     | NULL    |       |
| ADDRESS    | varchar(25) | YES  |     | NULL    |       |
| CITY       | varchar(20) | YES  |     | NULL    |       |
| STATE      | varchar(20) | YES  |     | NULL    |       |
| PIN        | varchar(6)  | YES  |     | NULL    |       |
| BASIC      | int(7)      | YES  |     | NULL    |       |
| JN_DT      | date        | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+

desc DESIGNATION;
MariaDB [assignment_1]> desc DESIGNATION;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| DESIG_CODE | varchar(16) | NO   | PRI | NULL    |       |
| DESIG_DESC | varchar(20) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+

desc DEPARTMENT;
MariaDB [assignment_1]> desc DEPARTMENT;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| DEPT_CODE | varchar(16) | NO   | PRI | NULL    |       |
| DEPT_NAME | varchar(20) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
```

Question 3:

Insert few rows in each table.
[While entering data in EMP table use DESIG_CODE which exists in DESIGNATION table and DEPT_CODE which is exists in DEPARTMENT table. In DESIGNATION table, assign code for Manager, Executive, officer, clerk and helper. In DEPARTMENT table, assign code for Personnel, Production, Purchase, Finance, Research departments]

Solution:

```
INSERT INTO DESIGNATION (DESIG_CODE, DESIG_DESC)
VALUES ('MGR', 'Manager'),
('EXE', 'Executive'),
('OFF', 'Officer'),
('CLK', 'Clerk'),
('HLP', 'Helper');

INSERT INTO DEPARTMENT (DEPT_CODE, DEPT_NAME)
VALUES ('PER', 'Personnel'),
('PRO', 'Production'),
('PUR', 'Purchase'),
('FIN', 'Finance'),
('RES', 'Research');

INSERT INTO EMPLOYEE (EMP_CODE, EMP_NAME, DEPT_CODE, DESIG_CODE, SEX, ADDRESS, CITY,
STATE, PIN, BASIC, JN_DT)
VALUES ('001', 'Adam Smith', 'PUR', 'MGR', 'M', '123 Main St', 'Berkeley', 'CA',
'90001', 50000, '2002-01-01');

INSERT INTO EMPLOYEE (EMP_CODE, EMP_NAME, DEPT_CODE, DESIG_CODE, SEX, ADDRESS, CITY,
STATE, PIN, BASIC, JN_DT)
VALUES ('002', 'Maddie Morris', 'PRO', 'EXE', 'F', '456 Elm St', 'Denver', 'CO',
'80001', 30000, '20015-03-10');

INSERT INTO EMPLOYEE (EMP_CODE, EMP_NAME, DEPT_CODE, DESIG_CODE, SEX, ADDRESS, CITY,
STATE, PIN, BASIC, JN_DT)
VALUES ('003', 'Edward Johnson', 'PUR', 'OFF', 'M', '789 Oak St', 'Atlanta', 'GA',
'30002', 40000, '2010-06-25');

INSERT INTO EMPLOYEE (EMP_CODE, EMP_NAME, DEPT_CODE, DESIG_CODE, SEX, ADDRESS, CITY,
STATE, PIN, BASIC, JN_DT)
VALUES ('004', 'Jenifer Martin', 'MGR', 'OFF', 'F', '245 Perl St', 'Nashville', 'TN',
'80001', 37010, '2008-01-24');

INSERT INTO EMPLOYEE (EMP_CODE, EMP_NAME, DEPT_CODE, DESIG_CODE, SEX, ADDRESS, CITY,
STATE, PIN, BASIC, JN_DT)
VALUES ('005', 'Sam Joe', 'FIN', 'CLK', 'M', '245 New St', 'Denver', 'CO', '80001',
35000, '2002-02-24');

INSERT INTO EMPLOYEE (EMP_CODE, EMP_NAME, DEPT_CODE, DESIG_CODE, SEX, ADDRESS, CITY,
STATE, PIN, BASIC, JN_DT)
VALUES ('006', 'Jessica Altman', 'PUR', 'HLP', 'F', '245 Old St', 'Atlanta', 'GA',
'80001', 45000, '2003-05-22');
```

```
INSERT INTO EMPLOYEE (EMP_CODE, EMP_NAME, DEPT_CODE, DESIG_CODE, SEX, ADDRESS, CITY,
STATE, PIN, BASIC, JN_DT)
VALUES ('007', 'Sarra Jhones', 'PRO', 'CLK', 'F', '160 Saint St', 'Nashville', 'TN',
'20001', 48000, '2001-06-18');

INSERT INTO EMPLOYEE (EMP_CODE, EMP_NAME, DEPT_CODE, DESIG_CODE, SEX, ADDRESS, CITY,
STATE, PIN, BASIC, JN_DT)
VALUES ('008', 'Maxine', 'FIN', 'CLK', 'F', '160 Perl St', 'Nashville', 'TN',
'20521', 48000, '2002-06-18');

MariaDB [assignment_1]> select * from employee;
```

EMP_CODE	EMP_NAME	DEPT_CODE	DESIG_CODE	SEX	ADDRESS	CITY	STATE	PIN	BASIC	JN_DT
001	Adam Smith	PUR	MGR	M	123 Main St	Berkeley	CA	90001	50000	2002-01-01
002	Maddie Morris	PRO	EXE	F	456 Elm St	Denver	CO	80001	0	0000-00-00
003	Edward Johnson	PUR	OFF	M	789 Oak St	Atlanta	GA	30002	40000	2010-06-25
004	Jenifer Martin	MGR	OFF	F	245 Perl St	Nashville	TN	80001	37010	2008-01-24
005	Sam Joe	FIN	CLK	M	245 New St	Denver	CO	80001	35000	2002-02-24
006	Jessica Altman	PUR	HLP	F	245 Old St	Atlanta	GA	80001	45000	2003-05-22
007	Sarra Jhones	PRO	CLK	F	160 Saint St	Nashville	TN	20001	48000	2001-06-18
008	Maxine	FIN	CLK	F	160 Perl St	Nashville	TN	20521	48000	2002-06-18

Question 4:

In EMP table insert few rows without DEPT_CODE and BASIC.

Solution:

```
INSERT INTO EMPLOYEE (EMP_CODE, EMP_NAME, DESIG_CODE, SEX, ADDRESS, CITY, STATE, PIN,
JN_DT)
VALUES ('009', 'Sophie Matrin', 'CLK', 'F', '246 Maple St', 'Boston', 'MA', '02101',
'2014-04-28');

INSERT INTO EMPLOYEE (EMP_CODE, EMP_NAME, DESIG_CODE, SEX, ADDRESS, CITY, STATE, PIN,
JN_DT)
VALUES ('010', 'Thomas Brown', 'HLP', 'M', '369 Pine St', 'San Francisco', 'CA',
'94101', '2016-10-12');
```

Question 5:

Find the rows with unassigned DEPT_CODE

Solution:

```
select * from employee where dept_code is NULL;
MariaDB [assignment_1]> select * from employee where dept_code is NULL;
```

EMP_CODE	EMP_NAME	DEPT_CODE	DESIG_CODE	SEX	ADDRESS	CITY	STATE	PIN	BASIC	JN_DT
009	Sophie Matrin	NULL	CLK	F	246 Maple St	Boston	MA	02101	NULL	2014-04-28
010	Thomas Brown	NULL	HLP	M	369 Pine St	San Francisco	CA	94101	NULL	2016-10-12

Question 6:

Find the rows with BASIC equal to zero

Solution:

```
select * from employee where basic = 0;
MariaDB [assignment_1]> select * from employee where basic = 0;
```

EMP_CODE	EMP_NAME	DEPT_CODE	DESIG_CODE	SEX	ADDRESS	CITY	STATE	PIN	BASIC	JN_DT
002	Maddie Morris	PRO	EXE	F	456 Elm St	Denver	CO	80001	0	0000-00-00

Question 7:

Find the rows with unassigned Basic.

Solution:

```
select * from employee where basic is NULL;
MariaDB [assignment_1]> select * from employee where basic is NULL;
```

EMP_CODE	EMP_NAME	DEPT_CODE	DESIG_CODE	SEX	ADDRESS	CITY	STATE	PIN	BASIC	JN_DT
009	Sophie Matrin	NULL	CLK	F	246 Maple St	Boston	MA	02101	NULL	2014-04-28
010	Thomas Brown	NULL	HLP	M	369 Pine St	San Francisco	CA	94101	NULL	2016-10-12

Question 8:

Find the average basic of the employees.

Solution:

```
select avg(basic) as 'Avg_Basic' from employee;
MariaDB [assignment_1]> select avg(basic) as 'Avg_Basic' from employee;
```

Avg_Basic
37876.2500

Question 9:

Replace the BASIC with 0 for the rows with unassigned Basic.

Solution:

```
update employee set basic = 0 where basic is NULL;
MariaDB [assignment_1]> update employee set basic = 0 where basic is NULL;
Query OK, 2 rows affected (0.004 sec)
Rows matched: 2  Changed: 2  Warnings: 0
```

Question 10:

Again, find the average Basic. (Note the difference of result obtained in Q.8 & Q.10.)

Solution:

```
select avg(basic) as 'Avg_Basic' from employee;
MariaDB [assignment_1]> select avg(basic) as 'Avg_Basic' from employee;
+------------+
| Avg_Basic  |
+------------+
| 30301.0000 |
+------------+
```

Question 11:

Delete the rows with unassigned DEPT_CODE

Solution:

```
delete from employee where dept_code is NULL;
MariaDB [assignment_1]> delete from employee where dept_code is NULL;
Query OK, 2 rows affected (0.008 sec)
```

Question 12:

Say, Net pay of an employee = Basic + HRA + DA where HRA is 50% of the Basic & DA is 40% of Basic. Show the employee name & Net pay for all employees.

Solution:

```
select emp_name, basic + (basic * 0.4) + (basic * 0.5) as "Net_Pay" from employee;
MariaDB [assignment_1]> select emp_name, basic + (basic * 0.4) + (basic * 0.5) as
"Net_Pay" from employee;
+----------------+---------+
| emp_name       | Net_Pay |
+----------------+---------+
| Adam Smith     | 95000.0 |
| Maddie Morris  |     0.0 |
| Edward Johnson | 76000.0 |
| Jenifer Martin | 70319.0 |
| Sam Joe        | 66500.0 |
| Jessica Altman | 85500.0 |
| Sarra Jhones   | 91200.0 |
| Maxine         | 91200.0 |
+----------------+---------+
```

Question 13:

Show the EMP_NAME & BASIC in the ascending order of DEPT_CODE. The employee name must appear in uppercase.

Solution:

```
select upper(emp_name), basic from employee order by dept_code;
MariaDB [assignment_1]> select upper(emp_name), basic from employee order by
dept_code;
+-----------------+-------+
| upper(emp_name) | basic |
+-----------------+-------+
| SAM JOE         | 35000 |
| Maxine          | 48000 |
| JENIFER MARTIN  | 37010 |
| MADDIE MORRIS   |     0 |
| SARRA JHONES    | 48000 |
| ADAM SMITH      | 50000 |
| EDWARD JOHNSON  | 40000 |
| JESSICA ALTMAN  | 45000 |
+-----------------+-------+
```

Question 14:

Find the employees who have joined after 1st January 2010.

Solution:

```
select * from employee where jn_dt > '2010-01-01';
MariaDB [assignment_1]> select * from employee where jn_dt > '2010-01-01';
+----------+---------------+-----------+------------+------+------------+---------+-------+-------+-------+------------+
| EMP_CODE | EMP_NAME      | DEPT_CODE | DESIG_CODE | SEX  | ADDRESS    | CITY    | STATE | PIN   | BASIC | JN_DT      |
+----------+---------------+-----------+------------+------+------------+---------+-------+-------+-------+------------+
| 003      | Edward Johnson | PUR       | OFF        | M    | 789 Oak St | Atlanta | GA    | 30002 | 40000 | 2010-06-25 |
+----------+---------------+-----------+------------+------+------------+---------+-------+-------+-------+------------+
```

Question 15:

Find, how many employees have joined in the month of January?

Solution:

```
select count(emp_code) as 'Joiners on January'  from employee where month(jn_dt) =
'01';
MariaDB [assignment_1]> select count(emp_code) as 'Joiners on January'  from employee
where month(jn_dt) = '01';
+--------------------+
| Joiners on January |
+--------------------+
|                  2 |
+--------------------+
```

Question 16:

Find the maximum & minimum Basic.

Solution:

```
select max(basic) as 'Max_Basic', min(basic) as 'Min_Basic' from employee;
MariaDB [assignment_1]> select max(basic) as 'Max_Basic', min(basic) as 'Min_Basic'
from employee;
+-----------+-----------+
| Max_Basic | Min_Basic |
+-----------+-----------+
|     50000 |         0 |
+-----------+-----------+
```

Question 17:

Find how many Female employees are there?

Solution:

```
select * from employee where sex = 'F';
MariaDB [assignment_1]> select * from employee where sex = 'F';
+----------+---------------+-----------+------------+------+--------------+-----------+-------+-------+-------+------------+
| EMP_CODE | EMP_NAME      | DEPT_CODE | DESIG_CODE | SEX  | ADDRESS      | CITY      | STATE | PIN   | BASIC | JN_DT      |
+----------+---------------+-----------+------------+------+--------------+-----------+-------+-------+-------+------------+
| 002      | Maddie Morris | PRO       | EXE        | F    | 456 Elm St   | Denver    | CO    | 80001 |     0 | 0000-00-00 |
| 004      | Jenifer Martin | MGR      | OFF        | F    | 245 Perl St  | Nashville | TN    | 80001 | 37010 | 2008-01-24 |
| 006      | Jessica Altman | PUR      | HLP        | F    | 245 Old St   | Atlanta   | GA    | 80001 | 45000 | 2003-05-22 |
| 007      | Sarra Jhones  | PRO       | CLK        | F    | 160 Saint St | Nashville | TN    | 20001 | 48000 | 2001-06-18 |
| 008      | Maxine        | FIN       | CLK        | F    | 160 Perl St  | Nashville | TN    | 20521 | 48000 | 2002-06-18 |
+----------+---------------+-----------+------------+------+--------------+-----------+-------+-------+-------+------------+
```

Question 18:

Replace CITY with existing value converted into uppercase for all rows.

Solution:

```
update employee set city = upper(city);
MariaDB [assignment_1]> update employee set city = upper(city);
Query OK, 8 rows affected (0.003 sec)
Rows matched: 8  Changed: 8  Warnings: 0

MariaDB [assignment_1]> select * from employee;
+----------+---------------+-----------+------------+------+--------------+-----------+-------+-------+-------+------------+
| EMP_CODE | EMP_NAME      | DEPT_CODE | DESIG_CODE | SEX  | ADDRESS      | CITY      | STATE | PIN   | BASIC | JN_DT      |
+----------+---------------+-----------+------------+------+--------------+-----------+-------+-------+-------+------------+
| 001      | Adam Smith    | PUR       | MGR        | M    | 123 Main St  | BERKELEY  | CA    | 90001 | 50000 | 2002-01-01 |
| 002      | Maddie Morris | PRO       | EXE        | F    | 456 Elm St   | DENVER    | CO    | 80001 |     0 | 0000-00-00 |
| 003      | Edward Johnson | PUR      | OFF        | M    | 789 Oak St   | ATLANTA   | GA    | 30002 | 40000 | 2010-06-25 |
| 004      | Jenifer Martin | MGR      | OFF        | F    | 245 Perl St  | NASHVILLE | TN    | 80001 | 37010 | 2008-01-24 |
| 005      | Sam Joe       | FIN       | CLK        | M    | 245 New St   | DENVER    | CO    | 80001 | 35000 | 2002-02-24 |
| 006      | Jessica Altman | PUR      | HLP        | F    | 245 Old St   | ATLANTA   | GA    | 80001 | 45000 | 2003-05-22 |
| 007      | Sarra Jhones  | PRO       | CLK        | F    | 160 Saint St | NASHVILLE | TN    | 20001 | 48000 | 2001-06-18 |
| 008      | Maxine        | FIN       | CLK        | F    | 160 Perl St  | NASHVILLE | TN    | 20521 | 48000 | 2002-06-18 |
+----------+---------------+-----------+------------+------+--------------+-----------+-------+-------+-------+------------+
```

Question 19:

Find in how many different cities various employees are residing?

Solution:

```
select count(distinct(city)) as 'Unique cities' from employee;
MariaDB [assignment_1]> select count(distinct(city)) as 'Unique cities' from
employee;
+---------------+
| Unique cities |
+---------------+
|             4 |
+---------------+
```

Question 20:

Display the employee information in the ascending order of DEPT_CODE and within a Department, it should be in the descending order of BASIC.

Solution:

```
select * from employee order by dept_code, basic desc;
MariaDB [assignment_1]> select * from employee order by dept_code, basic desc;
```

EMP_CODE	EMP_NAME	DEPT_CODE	DESIG_CODE	SEX	ADDRESS	CITY	STATE	PIN	BASIC	JN_DT
008	Maxine	FIN	CLK	F	160 Perl St	NASHVILLE	TN	20521	48000	2002-06-18
005	Sam Joe	FIN	CLK	M	245 New St	DENVER	CO	80001	35000	2002-02-24
004	Jenifer Martin	MGR	OFF	F	245 Perl St	NASHVILLE	TN	80001	37010	2008-01-24
007	Sarra Jhones	PRO	CLK	F	160 Saint St	NASHVILLE	TN	20001	48000	2001-06-18
002	Maddie Morris	PRO	EXE	F	456 Elm St	DENVER	CO	80001	0	0000-00-00
001	Adam Smith	PUR	MGR	M	123 Main St	BERKELEY	CA	90001	50000	2002-01-01
006	Jessica Altman	PUR	HLP	F	245 Old St	ATLANTA	GA	80001	45000	2003-05-22
003	Edward Johnson	PUR	OFF	M	789 Oak St	ATLANTA	GA	30002	40000	2010-06-25

Assignment: Set - 2

Assignment – 2

Question 1:

From the EMP table show the minimum, maximum and average basic for each department (show dept. Code).

Solution:

```
select dept_code, max(basic), min(basic), avg(basic) from employee group by
dept_code;
MariaDB [assignment_1]> select dept_code, max(basic), min(basic), avg(basic) from
employee group by dept_code;
+-----------+------------+------------+------------+
| dept_code | max(basic) | min(basic) | avg(basic) |
+-----------+------------+------------+------------+
| FIN       |      48000 |      35000 | 41500.0000 |
| MGR       |      37010 |      37010 | 37010.0000 |
| PRO       |      48000 |      32500 | 40250.0000 |
| PUR       |      50000 |      40000 | 45000.0000 |
+-----------+------------+------------+------------+
```

Question 2:

Find the number of female employees in each department (show dept. Code).

Solution:

```
select dept_code, count(emp_code) as 'Count of Female' from employee where sex = 'F'
group by dept_code;
MariaDB [assignment_1]> select dept_code, count(emp_code) as 'Count of Female' from
employee where sex = 'F' group by dept_code;
+-----------+-----------------+
| dept_code | Count of Female |
+-----------+-----------------+
| FIN       |               1 |
| MGR       |               1 |
| PRO       |               2 |
| PUR       |               1 |
+-----------+-----------------+
```

Question 3:

Find the city wise no. of employees for each department (show dept. Code).

Solution:

```
select distinct(dept_code), city, count(*) from employee group by city;
MariaDB [assignment_1]> select distinct(dept_code), city, count(*) from employee
group by city;
+-----------+-----------+----------+
| dept_code | city      | count(*) |
+-----------+-----------+----------+
| PUR       | ATLANTA   |        2 |
| PUR       | BERKELEY  |        1 |
| PRO       | DENVER    |        2 |
| MGR       | NASHVILLE |        3 |
+-----------+-----------+----------+
```

Question 4:

Show the designation wise no of employees who have joined in the year 2000 in each department. The listing should appear in the ascending order of no. of employees.

Solution:

```
select desig_code, dept_code, count(emp_code) from employee where year(jn_dt) =
'2002' group by desig_code;
MariaDB [assignment_1]> select desig_code, dept_code, count(emp_code) from employee
where year(jn_dt) = '2000' group by desig_code;
+------------+-----------+-----------------+
| desig_code | dept_code | count(emp_code) |
+------------+-----------+-----------------+
| CLK        | FIN       |               2 |
| MGR        | PUR       |               1 |
+------------+-----------+-----------------+
```

Question 5:

Find the department code wise total basic of male employees only for the departments for which such total is more than 50,000 and the listing should appear in the descending order of total basic.

Solution:

```
select dept_code, sum(basic) from employee where sex = 'M' group by dept_code having
sum(basic) > 50000;
MariaDB [assignment_1]> select dept_code, sum(basic) from employee where sex = 'M'
group by dept_code having sum(basic) > 50000;
+-----------+------------+
| dept_code | sum(basic) |
+-----------+------------+
| PUR       |      90000 |
+-----------+------------+
```

Question 6:

Show the employee name, Designation description and basic for all employees.

Solution:

```
select e.emp_name, d.desig_desc from employee e, designation d
where e.desig_code = d.desig_code;
MariaDB [assignment_1]> select e.emp_name, d.desig_desc from employee e,designation d
    -> where e.desig_code = d.desig_code;
+---------------+------------+
| emp_name      | desig_desc |
+---------------+------------+
| Adam Smith    | Manager    |
| Maddie Morris | Executive  |
| Edward Johnson | Officer    |
| Jenifer Martin | Officer    |
| Sam Joe       | Clerk      |
```

```
| Jessica Altman | Helper     |
| Sarra Jhones   | Clerk      |
| Maxine         | Clerk      |
+----------------+------------+
```

Question 7:

Show the employee name, Designation description, Department Name & Basic for all employees.

Solution:

```
select e.emp_name, d.desig_code, dp.dept_name, e.basic
from employee e, designation d, department dp
where e.desig_code = d.desig_code and e.dept_code = dp.dept_code;

MariaDB [assignment_1]> select e.emp_name, d.desig_code, dp.dept_name, e.basic
    -> from employee e, designation d, department dp
    -> where e.desig_code = d.desig_code and e.dept_code = dp.dept_code;
+---------------+------------+------------+-------+
| emp_name      | desig_code | dept_name  | basic |
+---------------+------------+------------+-------+
| Sam Joe       | CLK        | Finance    | 35000 |
| Maxine        | CLK        | Finance    | 48000 |
| Maddie Morris | EXE        | Production | 32500 |
| Sarra Jhones  | CLK        | Production | 48000 |
| Adam Smith    | MGR        | Purchase   | 50000 |
| Edward Johnson | OFF       | Purchase   | 40000 |
| Jessica Altman | HLP       | Purchase   | 45000 |
+---------------+------------+------------+-------+
```

Question 8:

Find the department Codes in which no employee works.

Solution:

```
select dept_code from department where dept_code
not in (select dept_code from employee);

MariaDB [assignment_1]> select dept_code from department where dept_code
    -> not in (select dept_code from employee);
+-----------+
| dept_code |
+-----------+
| PER       |
| RES       |
+-----------+
```

Question 9:

Find the department names where at least one employee works.

Solution:

```
select dept_name from department
where dept_code in
(select dept_code from employee);
```

```
MariaDB [assignment_1]> select dept_name from department
    -> where dept_code in
    -> (select dept_code from employee);
+------------+
| dept_name  |
+------------+
| Finance    |
| Production |
| Purchase   |
+------------+
```

Question 10:

Find the department names where at least 10 employees work.

Solution:

```
select d.dept_name from department d
where (select count(*) from employee e where e.dept_code = d.dept_code) > 9;
```

```
MariaDB [assignment_1]> select d.dept_name from department d
    -> where (select count(*) from employee e where e.dept_code = d.dept_code) > 9;
Empty set
```

Question 11:

Find the department code in which employee with highest Basic works.

Solution:

```
select dept_code, basic as MAX_BASIC from employee
where basic = (select max(basic) from employee);
```

```
MariaDB [assignment_1]> select dept_code, basic as MAX_BASIC from employee
    -> where basic = (select max(basic) from employee);
+-----------+-----------+
| dept_code | MAX_BASIC |
+-----------+-----------+
| PUR       |     50000 |
+-----------+-----------+
```

Question 12:

Find the Designation description of the employee with highest basic.

Solution:

```
select d.desig_desc, e.basic as MAX_BASIC
from employee e, designation d
where e.desig_code = d.desig_code and
e.basic = (select max(basic) from employee);
```

```
MariaDB [assignment_1]> select d.desig_desc, e.basic as MAX_BASIC
    -> from employee e, designation d
    -> where e.desig_code = d.desig_code and
    -> e.basic = (select max(basic) from employee);
+------------+-----------+
| desig_desc | MAX_BASIC |
+------------+-----------+
| Manager    |     50000 |
+------------+-----------+
```

Question 13:

Find the no. of managers in each department.

Solution:

```
select d.dept_name, count(*) as NO_OF_MANAGER
from employee e, department d where e.dept_code = d.dept_code
and e.desig_code in (select desig_code from designation
where desig_desc = 'Manager');
```

```
MariaDB [assignment_1]> select d.dept_name, count(*) as NO_OF_MANAGER
    -> from employee e, department d where e.dept_code = d.dept_code
    -> and e.desig_code in (select desig_code from designation
    -> where desig_desc = 'Manager');
+-----------+---------------+
| dept_name | NO_OF_MANAGER |
+-----------+---------------+
| Purchase  |             1 |
+-----------+---------------+
```

Question 14:

Find the maximum basic from EMP table without using MAX().

Solution:

```
select distinct basic as Max_Basic from employee e
where e.basic >= all(select basic from employee);
```

```
MariaDB [assignment_1]> select distinct basic as Max_Basic from employee e
    -> where e.basic >= all(select basic from employee);
```

```
+-----------+
| Max_Basic |
+-----------+
|     50000 |
+-----------+
```

Question 15:

Find the minimum basic from EMP table without using MIN().

Solution:

```
select distinct basic as Min_Basic from employee e
where e.basic <= all(select basic from employee);
```

```
MariaDB [assignment_1]> select distinct basic as Min_Basic from employee e
    -> where e.basic <= all(select basic from employee);
+-----------+
| Min_Basic |
+-----------+
|     32500 |
+-----------+
```

Question 16:

Find the name of the department with highest total basic. Do the same for highest average basic and maximum no. of employee.

Solution:

Part – 1

```
select d.dept_name, e.basic from employee e, department d
where e.dept_code = d.dept_code
and e.basic >= all(select basic from employee);
```

```
MariaDB [assignment_1]> select d.dept_name, e.basic from employee e, department d
    -> where e.dept_code = d.dept_code
    -> and e.basic >= all(select basic from employee);
+-----------+-------+
| dept_name | basic |
+-----------+-------+
| Purchase  | 50000 |
+-----------+-------+
```

Part - 2

```
select dept_name, basic as AVERAGE_BASIC from (
      select dept_name, avg(basic) as Basic from employee e, department d
      where e.dept_code = d.dept_code group by dept_name)
where basic = (select max(basic) from (
      select dept_name, avg(basic) as Basic from employee e, department d
      where e.dept_code = d.dept_code group by dept_name)
);
```

```
+-----------+-----------+
| dept_name | avg_basic |
+-----------+-----------+
| Purchase  | 45000     |
+-----------+-----------+
```

Part – 3

```
select dept_name, count(emp_code) as Count_of_Employee from employee, department
where employee.dept_code = department.dept_code
group by employee.dept_code order by count(emp_code) desc limit 1;

MariaDB [assignment_1]> select dept_name, count(emp_code) as Count_of_Employee from
employee, department
    -> where employee.dept_code = department.dept_code
    -> group by employee.dept_code order by count(emp_code) desc limit 1;
+-----------+-------------------+
| dept_name | Count_of_Employee |
+-----------+-------------------+
| Purchase  |                 3 |
+-----------+-------------------+
```

Question 17:

Insert same rows into EMP table with designation code not existing in DESIGNATION table.

Solution:

```
INSERT INTO EMPLOYEE (EMP_CODE, EMP_NAME, DEPT_CODE, DESIG_CODE, SEX, ADDRESS, CITY,
STATE, PIN, BASIC, JN_DT)
VALUES ('009', 'Adam Smith', 'PUR', 'Manager', 'M', '123 Main St', 'Berkeley', 'CA',
'90001', 50000, '2002-01-01');

INSERT INTO EMPLOYEE (EMP_CODE, EMP_NAME, DEPT_CODE, DESIG_CODE, SEX, ADDRESS, CITY,
STATE, PIN, BASIC, JN_DT)
VALUES ('010', 'Adam Smith', 'PUR', 'Worker', 'M', '123 Main St', 'Berkeley', 'CA',
'90001', 50000, '2002-01-01');

INSERT INTO EMPLOYEE (EMP_CODE, EMP_NAME, DEPT_CODE, DESIG_CODE, SEX, ADDRESS, CITY,
STATE, PIN, BASIC, JN_DT)
VALUES ('011', 'Adam Smith', 'PUR', 'Client', 'M', '123 Main St', 'Berkeley', 'CA',
'90001', 50000, '2002-01-01');

INSERT INTO EMPLOYEE (EMP_CODE, EMP_NAME, DEPT_CODE, DESIG_CODE, SEX, ADDRESS, CITY,
STATE, PIN, BASIC, JN_DT)
VALUES ('012', 'Adam Smith', 'PUR', 'Executive', 'M', '123 Main St', 'Berkeley',
'CA', '90001', 50000, '2002-01-01');
```

```
MariaDB [assignment_1]> select * from employee;
+----------+----------------+-----------+------------+------+--------------+-----------+-------+-------+-------+------------+
| EMP_CODE | EMP_NAME       | DEPT_CODE | DESIG_CODE | SEX  | ADDRESS      | CITY      | STATE | PIN   | BASIC | JN_DT      |
+----------+----------------+-----------+------------+------+--------------+-----------+-------+-------+-------+------------+
| 001      | Adam Smith     | PUR       | MGR        | M    | 123 Main St  | BERKELEY  | CA    | 90001 | 50000 | 2002-01-01 |
| 002      | Maddie Morris  | PRO       | EXE        | F    | 456 Elm St   | DENVER    | CO    | 80001 | 32500 | 0000-00-00 |
| 003      | Edward Johnson | PUR       | OFF        | M    | 789 Oak St   | ATLANTA   | GA    | 30002 | 40000 | 2010-06-25 |
| 004      | Jenifer Martin | MGR       | OFF        | F    | 245 Perl St  | NASHVILLE | TN    | 80001 | 37010 | 2008-01-24 |
| 005      | Sam Joe        | FIN       | CLK        | M    | 245 New St   | DENVER    | CO    | 80001 | 35000 | 2002-02-24 |
| 006      | Jessica Altman | PUR       | HLP        | F    | 245 Old St   | ATLANTA   | GA    | 80001 | 45000 | 2003-05-22 |
| 007      | Sarra Jhones   | PRO       | CLK        | F    | 160 Saint St | NASHVILLE | TN    | 20001 | 48000 | 2001-06-18 |
| 008      | Maxine         | FIN       | CLK        | F    | 160 Perl St  | NASHVILLE | TN    | 20521 | 48000 | 2002-06-18 |
| 009      | Adam Smith     | PUR       | Manager    | M    | 123 Main St  | Berkeley  | CA    | 90001 | 50000 | 2002-01-01 |
| 010      | Adam Smith     | PUR       | Worker     | M    | 123 Main St  | Berkeley  | CA    | 90001 | 50000 | 2002-01-01 |
| 011      | Adam Smith     | PUR       | Client     | M    | 123 Main St  | Berkeley  | CA    | 90001 | 50000 | 2002-01-01 |
| 012      | Adam Smith     | PUR       | Executive  | M    | 123 Main St  | Berkeley  | CA    | 90001 | 50000 | 2002-01-01 |
+----------+----------------+-----------+------------+------+--------------+-----------+-------+-------+-------+------------+
```

Question 18:

Delete the rows from EMP table with invalid DESIG_CODE.

Solution:

```
delete from employee where desig_code not in
(select desig_code from designation);
```

```
MariaDB [assignment_1]> delete from employee where desig_code not in
    -> (select desig_code from designation);
Query OK, 4 rows affected (0.005 sec)
```

```
MariaDB [assignment_1]> select * from employee;
+----------+----------------+-----------+------------+------+--------------+-----------+-------+-------+-------+------------+
| EMP_CODE | EMP_NAME       | DEPT_CODE | DESIG_CODE | SEX  | ADDRESS      | CITY      | STATE | PIN   | BASIC | JN_DT      |
+----------+----------------+-----------+------------+------+--------------+-----------+-------+-------+-------+------------+
| 001      | Adam Smith     | PUR       | MGR        | M    | 123 Main St  | BERKELEY  | CA    | 90001 | 50000 | 2002-01-01 |
| 002      | Maddie Morris  | PRO       | EXE        | F    | 456 Elm St   | DENVER    | CO    | 80001 | 32500 | 0000-00-00 |
| 003      | Edward Johnson | PUR       | OFF        | M    | 789 Oak St   | ATLANTA   | GA    | 30002 | 40000 | 2010-06-25 |
| 004      | Jenifer Martin | MGR       | OFF        | F    | 245 Perl St  | NASHVILLE | TN    | 80001 | 37010 | 2008-01-24 |
| 005      | Sam Joe        | FIN       | CLK        | M    | 245 New St   | DENVER    | CO    | 80001 | 35000 | 2002-02-24 |
| 006      | Jessica Altman | PUR       | HLP        | F    | 245 Old St   | ATLANTA   | GA    | 80001 | 45000 | 2003-05-22 |
| 007      | Sarra Jhones   | PRO       | CLK        | F    | 160 Saint St | NASHVILLE | TN    | 20001 | 48000 | 2001-06-18 |
| 008      | Maxine         | FIN       | CLK        | F    | 160 Perl St  | NASHVILLE | TN    | 20521 | 48000 | 2002-06-18 |
+----------+----------------+-----------+------------+------+--------------+-----------+-------+-------+-------+------------+
```

Question 19:

Find the name of the female employees with basic greater than the average basic of their respective department.

Solution:

```
select e.emp_name, e.basic from employee e, department d where
e.dept_code = d.dept_code and basic > (
      select avg(basic) from employee e, department d where
      e.dept_code = d.dept_code and e.sex = 'F'
) and e.sex = 'F';
```

```
MariaDB [assignment_1]> select e.emp_name, e.basic from employee e, department d
where
    -> e.dept_code = d.dept_code and basic > (
    -> select avg(basic) from employee e, department d where
    -> e.dept_code = d.dept_code and e.sex = 'F'
    -> ) and e.sex = 'F';
+----------------+-------+
| emp_name       | basic |
+----------------+-------+
| Jessica Altman | 45000 |
| Sarra Jhones   | 48000 |
| Maxine         | 48000 |
+----------------+-------+
```

Question 20:

Find the number of female managers.

Solution:

```
select count(*) as NO_OF_FEMALE_MANAGER from employee e, designation d
where e.desig_code = d.desig_code and e.sex = 'F'
and d.desig_code = "MGR";

MariaDB [assignment_1]> select count(*) as NO_OF_FEMALE_MANAGER from employee e,
designation d
    -> where e.desig_code = d.desig_code and e.sex = 'F'
    -> and d.desig_code = "MGR";
+----------------------+
| NO_OF_FEMALE_MANAGER |
+----------------------+
|                    1 |
+----------------------+
```

Assignment: Set - 3

Assignment – 3

Question 1:

In an organization, number of departments exists. Each department has a name & unique code. Number of employees work in each department. Each employee has unique employee code. Detailed information like name, address, city, basic, date of join are also stored. In a leave register for each employee leave records are kept showing leave type (CL/EL/ML etc.), from-date and to-date. When an employee retires or resigns then all the leave information pertaining to him are also deleted. Basic salary must be within Rs.5000 to Rs.9000. A department cannot be deleted if any employee record refers to it. Valid grades are A/B/C. Employee name must be in uppercase only. Default value for joining date is system date.

Design & implement the tables with necessary constraints to support the scenario depicted above.

Solution:

Creating Department table:

```
CREATE TABLE DEPARTMENT (
    DEPT_CODE VARCHAR(16) PRIMARY KEY,
    DEPT_NAME VARCHAR(50) NOT NULL
);
```

```
MariaDB [assignment_3]> DESC DEPARTMENT;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| DEPT_CODE | varchar(16) | NO   | PRI | NULL    |       |
| DEPT_NAME | varchar(50) | NO   |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
```

Creating Department table:

```
CREATE TABLE EMPLOYEE (
    EMP_CODE VARCHAR(16) PRIMARY KEY,
    EMP_NAME VARCHAR(50) CHECK (EMP_NAME = upper(EMP_NAME)),
    DEPT_CODE VARCHAR(16) NOT NULL,
    SEX CHAR(1) CHECK (SEX IN ('M', 'F')),
    ADDRESS VARCHAR(100),
    CITY VARCHAR(50),
    STATE VARCHAR(50),
    PIN INTEGER,
    BASIC INTEGER CHECK (BASIC BETWEEN 5000 AND 9000),
    JN_DT DATE DEFAULT (current_date()),
    GRADE CHAR(1) CHECK (GRADE IN ('A', 'B', 'C')),
      FOREIGN KEY (DEPT_CODE)
        REFERENCES DEPARTMENT (DEPT_CODE)
        ON DELETE RESTRICT
);
```

```
MariaDB [assignment_3]> DESC EMPLOYEE;
+-----------+-------------+------+-----+-----------+-------+
| Field     | Type        | Null | Key | Default   | Extra |
+-----------+-------------+------+-----+-----------+-------+
| EMP_CODE  | varchar(16) | NO   | PRI | NULL      |       |
| EMP_NAME  | varchar(50) | YES  |     | NULL      |       |
| DEPT_CODE | varchar(16) | NO   | MUL | NULL      |       |
| SEX       | char(1)     | YES  |     | NULL      |       |
| ADDRESS   | varchar(100)| YES  |     | NULL      |       |
| CITY      | varchar(50) | YES  |     | NULL      |       |
| STATE     | varchar(50) | YES  |     | NULL      |       |
| PIN       | int(11)     | YES  |     | NULL      |       |
| BASIC     | int(11)     | YES  |     | NULL      |       |
| JN_DT     | date        | YES  |     | curdate() |       |
| GRADE     | char(1)     | YES  |     | NULL      |       |
+-----------+-------------+------+-----+-----------+-------+
```

Creating Leave Record table:

```
CREATE TABLE LEAVE_RECORD (
    LEAVE_ID INTEGER PRIMARY KEY,
    EMP_CODE VARCHAR(16) NOT NULL,
    LEAVE_TYPE VARCHAR(10) CHECK (LEAVE_TYPE IN ('CL', 'EL', 'ML')),
    FROM_DATE DATE,
    TO_DATE DATE,
      FOREIGN KEY (EMP_CODE)
        REFERENCES EMPLOYEE (EMP_CODE)
        ON DELETE CASCADE
);

MariaDB [assignment_3]> DESC LEAVE_RECORD;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| LEAVE_ID   | int(11)     | NO   | PRI | NULL    |       |
| EMP_CODE   | varchar(16) | NO   | MUL | NULL    |       |
| LEAVE_TYPE | varchar(10) | YES  |     | NULL    |       |
| FROM_DATE  | date        | YES  |     | NULL    |       |
| TO_DATE    | date        | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
```

Inserting data in Department table:

```
INSERT INTO DEPARTMENT (DEPT_CODE, DEPT_NAME)
VALUES
      ('D001', 'Sales'),
      ('D002', 'Marketing'),
      ('D003', 'Finance');

MariaDB [assignment_3]> SELECT * FROM DEPARTMENT;
+-----------+-----------+
| DEPT_CODE | DEPT_NAME |
+-----------+-----------+
| D001      | Sales     |
| D002      | Marketing |
| D003      | Finance   |
+-----------+-----------+
```

Inserting data in Employee table:

```
INSERT INTO EMPLOYEE (EMP_CODE, EMP_NAME, DEPT_CODE, SEX, ADDRESS, CITY, STATE, PIN,
BASIC, GRADE)
VALUES
    ('E0010', 'John Doe', 'D001', 'M', '123 Main St', 'New York', 'NY', 12345, 6000,
'A'),
    ('E0011', 'Jane Smith', 'D001', 'F', '456 Elm St', 'Los Angeles', 'CA', 67890,
7000, 'B'),
    ('E0012', 'Michael Johnson', 'D002', 'M', '789 Oak St', 'Chicago', 'IL', 98765,
8000, 'C'),
    ('E0013', 'Emily Brown', 'D002', 'F', '321 Pine St', 'Houston', 'TX', 54321,
5500, 'A'),
    ('E0014', 'David Wilson', 'D003', 'M', '654 Cedar St', 'Miami', 'FL', 13579,
9000, 'B');

MariaDB [assignment_3]> SELECT * FROM EMPLOYEE;
+----------+----------------+-----------+------+-------------+-------------+-------+-------+-------+------------+-------+
| EMP_CODE | EMP_NAME       | DEPT_CODE | SEX  | ADDRESS     | CITY        | STATE | PIN   | BASIC | JN_DT      | GRADE |
+----------+----------------+-----------+------+-------------+-------------+-------+-------+-------+------------+-------+
| E0010    | John Doe       | D001      | M    | 123 Main St | New York    | NY    | 12345 |  6000 | 2023-06-10 | A     |
| E0011    | Jane Smith     | D001      | F    | 456 Elm St  | Los Angeles | CA    | 67890 |  7000 | 2023-06-10 | B     |
| E0012    | Michael Johnson | D002      | M    | 789 Oak St  | Chicago     | IL    | 98765 |  8000 | 2023-06-10 | C     |
| E0013    | Emily Brown    | D002      | F    | 321 Pine St | Houston     | TX    | 54321 |  5500 | 2023-06-10 | A     |
| E0014    | David Wilson   | D003      | M    | 654 Cedar St | Miami      | FL    | 13579 |  9000 | 2023-06-10 | B     |
+----------+----------------+-----------+------+-------------+-------------+-------+-------+-------+------------+-------+
```

Inserting data in Leave record table:

```
INSERT INTO LEAVE_RECORD (LEAVE_ID, EMP_CODE, LEAVE_TYPE, FROM_DATE, TO_DATE)
VALUES
    (1, 'E0010', 'CL', '2023-02-01', '2023-02-02'),
    (2, 'E0011', 'EL', '2023-03-15', '2023-03-20'),
    (3, 'E0012', 'ML', '2023-04-10', '2023-04-15'),
    (4, 'E0013', 'CL', '2023-05-10', '2023-05-10'),
    (5, 'E0014', 'EL', '2023-06-01', '2023-06-05');

MariaDB [assignment_3]> SELECT * FROM LEAVE_RECORD;
+----------+----------+------------+------------+------------+
| LEAVE_ID | EMP_CODE | LEAVE_TYPE | FROM_DATE  | TO_DATE    |
+----------+----------+------------+------------+------------+
|        1 | E0010    | CL         | 2023-02-01 | 2023-02-02 |
|        2 | E0011    | EL         | 2023-03-15 | 2023-03-20 |
|        3 | E0012    | ML         | 2023-04-10 | 2023-04-15 |
|        4 | E0013    | CL         | 2023-05-10 | 2023-05-10 |
|        5 | E0014    | EL         | 2023-06-01 | 2023-06-05 |
+----------+----------+------------+------------+------------+
```

Question 2:

Try to violate the constraints that you have implemented in the table & note, what happens. [Try with suitable INSERT/UPDATE/DELETE instruction]

Solution:

Violate foreign key integrity constraint:

```
INSERT INTO EMPLOYEE (EMP_CODE, EMP_NAME, DEPT_CODE, SEX, BASIC)
VALUES ('E002', 'john doe', 'D0010', 'M', 5500);

ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails
(`assignment_3`.`employee`, CONSTRAINT `employee_ibfk_1` FOREIGN KEY (`DEPT_CODE`)
REFERENCES `department` (`DEPT_CODE`))
```

Violate salary range constraint:

```
INSERT INTO EMPLOYEE (EMP_CODE, EMP_NAME, DEPT_CODE, SEX, BASIC)
VALUES ('E002', 'Jane Smith', 'D002', 'F', 9500);

ERROR 4025 (23000): CONSTRAINT `employee.BASIC` failed for `assignment_3`.`employee`
```

Violate grade range constraint:

```
INSERT INTO EMPLOYEE (EMP_CODE, EMP_NAME, DEPT_CODE, SEX, BASIC, GRADE)
VALUES ('E002', 'Jane Smith', 'D002', 'F', 9500, 'D');

ERROR 4025 (23000): CONSTRAINT `employee.BASIC` failed for `assignment_3`.`employee`
```

Violate salary range constraint:

```
INSERT INTO EMPLOYEE (EMP_CODE, EMP_NAME, DEPT_CODE, SEX, BASIC, GRADE)
VALUES ('E002', 'jane Smith', 'D002', 'F', 3000, 'C');

ERROR 4025 (23000): CONSTRAINT `employee.BASIC` failed for `assignment_3`.`employee`
```

Using update instruction:

```
select * from employee;
update employee
set DEPT_CODE = 'D010'
where emp_code = 'E0010';

ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails
(`assignment_3`.`employee`, CONSTRAINT `employee_ibfk_1` FOREIGN KEY (`DEPT_CODE`)
REFERENCES `department` (`DEPT_CODE`))
```

Using delete instruction:

```
DELETE FROM DEPARTMENT WHERE DEPT_CODE = 'D0010';

MariaDB [assignment_3]> DELETE FROM DEPARTMENT WHERE DEPT_CODE = 'D0010';
Query OK, 0 rows affected (0.001 sec)
```

Question 3:

a) Create a view showing employee code, name, dcode & Basic For a particular department.
b) Try to ensure a row into the view with valid department & also with invalid ones.
c) Find the newly inserted row in the table from which view was created.
d) Try to increment basic by Rs.100/.
e) Check it in the original table.
f) Delete the view.

Solution:

Part a:

```
MariaDB [assignment_3]> CREATE VIEW EmployeeView AS
    -> SELECT EMP_CODE, EMP_NAME, DEPT_CODE, BASIC
    -> FROM EMPLOYEE
    -> WHERE DEPT_CODE = 'D001';
Query OK, 0 rows affected

MariaDB [assignment_3]> SELECT * FROM EMPLOYEEVIEW;
+----------+------------+-----------+-------+
| EMP_CODE | EMP_NAME   | DEPT_CODE | BASIC |
+----------+------------+-----------+-------+
| E0010    | John Doe   | D001      |  6000 |
| E0011    | Jane Smith | D001      |  7000 |
+----------+------------+-----------+-------+
```

Part b:

```
MariaDB [assignment_3]> INSERT INTO EMPLOYEE (EMP_CODE, EMP_NAME, DEPT_CODE, SEX,
BASIC,
    -> VALUES ('E003', 'janey Smith', 'D001', 'F', 8000, 'C');

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds
to your MariaDB server version for the right syntax to use near 'VALUES
('E003', 'janey Smith', 'D001', 'F', 8000, 'C')' at line 2


MariaDB [assignment_3]> SELECT * FROM EMPLOYEEVIEW;
+----------+------------+-----------+-------+
| EMP_CODE | EMP_NAME   | DEPT_CODE | BASIC |
+----------+------------+-----------+-------+
| E0010    | John Doe   | D001      |  6100 |
| E0011    | Jane Smith | D001      |  7000 |
+----------+------------+-----------+-------+
```

Part c:

```
MariaDB [assignment_3]> SELECT * FROM EMPLOYEE, EMPLOYEEVIEW
    -> WHERE EMPLOYEE.EMP_CODE = EMPLOYEEVIEW.EMP_CODE;
+----------+------------+-----------+-----+-------------+-------------+-------+--------+--------+------------+-------+----------+------------+-----------+--------+
| EMP_CODE | EMP_NAME   | DEPT_CODE | SEX | ADDRESS     | CITY        | STATE | PIN    | BASIC | JN_DT      | GRADE | EMP_CODE | EMP_NAME   | DEPT_CODE | BASIC |
+----------+------------+-----------+-----+-------------+-------------+-------+--------+--------+------------+-------+----------+------------+-----------+--------+
| E0010    | John Doe   | D001      | M   | 123 Main St | New York    | NY    | 12345 |  6000 | 2023-06-10 | A     | E0010    | John Doe   | D001      |  6000 |
| E0011    | Jane Smith | D001      | F   | 456 Elm St  | Los Angeles | CA    | 67890 |  7000 | 2023-06-10 | B     | E0011    | Jane Smith | D001      |  7000 |
+----------+------------+-----------+-----+-------------+-------------+-------+--------+--------+------------+-------+----------+------------+-----------+--------+
```

Part d:

```
MariaDB [assignment_3]> UPDATE EMPLOYEE SET BASIC = BASIC + 100
      -> WHERE DEPT_CODE = 'D001';

Query OK, 2 rows affected
Rows matched: 2  Changed: 2  Warnings: 0
```

Part e:

```
MariaDB [assignment_3]> SELECT * FROM EMPLOYEE;
+----------+----------------+-----------+------+-------------+-------------+-------+-------+-------+------------+-------+
| EMP_CODE | EMP_NAME       | DEPT_CODE | SEX  | ADDRESS     | CITY        | STATE | PIN   | BASIC | JN_DT      | GRADE |
+----------+----------------+-----------+------+-------------+-------------+-------+-------+-------+------------+-------+
| E0010    | John Doe       | D001      | M    | 123 Main St | New York    | NY    | 12345 |  6100 | 2023-06-10 | A     |
| E0011    | Jane Smith     | D001      | F    | 456 Elm St  | Los Angeles | CA    | 67890 |  7100 | 2023-06-10 | B     |
| E0012    | Michael Johnson | D002     | M    | 789 Oak St  | Chicago     | IL    | 98765 |  8000 | 2023-06-10 | C     |
| E0013    | Emily Brown    | D002      | F    | 321 Pine St | Houston     | TX    | 54321 |  5500 | 2023-06-10 | A     |
| E0014    | David Wilson   | D003      | M    | 654 Cedar St | Miami      | FL    | 13579 |  9000 | 2023-06-10 | B     |
+----------+----------------+-----------+------+-------------+-------------+-------+-------+-------+------------+-------+
```

Part f:

```
DROP VIEW EmployeeView;
```

Question 4:

a) Create a view Showing empcode, name, deptname, basic, leave type, from date & to date.
b) Try to insert a row in the view. Check what happens?
c) Try to increment basic by Rs.100.
d) Delete the view.

Solution:

Part a:

```
CREATE VIEW EMPLOYEELEAVEVIEW AS
SELECT EMPLOYEE.EMP_CODE, EMPLOYEE.EMP_NAME, DEPARTMENT.DEPT_NAME, EMPLOYEE.BASIC,
LEAVE_RECORD.LEAVE_TYPE, LEAVE_RECORD.FROM_DATE, LEAVE_RECORD.TO_DATE
FROM EMPLOYEE, DEPARTMENT, LEAVE_RECORD
WHERE EMPLOYEE.DEPT_CODE = DEPARTMENT.DEPT_CODE
AND EMPLOYEE.EMP_CODE = LEAVE_RECORD.EMP_CODE;
```

```
MariaDB [assignment_3]> select * from EMPLOYEELEAVEVIEW;
+----------+----------------+-----------+-------+------------+------------+------------+
| EMP_CODE | EMP_NAME       | DEPT_NAME | BASIC | LEAVE_TYPE | FROM_DATE  | TO_DATE    |
+----------+----------------+-----------+-------+------------+------------+------------+
| E0010    | John Doe       | Sales     |  6100 | CL         | 2023-02-01 | 2023-02-02 |
| E0011    | Jane Smith     | Sales     |  7100 | EL         | 2023-03-15 | 2023-03-20 |
| E0012    | Michael Johnson | Marketing |  8000 | ML         | 2023-04-10 | 2023-04-15 |
| E0013    | Emily Brown    | Marketing |  5500 | CL         | 2023-05-10 | 2023-05-10 |
| E0014    | David Wilson   | Finance   |  9000 | EL         | 2023-06-01 | 2023-06-05 |
+----------+----------------+-----------+-------+------------+------------+------------+
```

Part b:

```
MariaDB [assignment_3]> insert into EmployeeLeaveView (EMP_CODE, EMP_NAME, DEPT_NAME,
BASIC, LEAVE_TYPE, FROM_DATE, TO_DATE)
    -> values
    -> ('E0015', 'jon snow', 'Sales', 7000, 'CL', '2023-02-01', '2023-02-02' );

ERROR 1393 (HY000): Can not modify more than one base table through a join view
'assignment_3.employeeleaveview'
```

Part c:

```
MariaDB [assignment_3]> UPDATE EMPLOYEE SET BASIC = BASIC + 100;

ERROR 4025 (23000): CONSTRAINT `employee.BASIC` failed for `assignment_3`.`employee`
```

Part d:

```
DROP VIEW EmployeeLeaveView;
```

Question 5:

a) Create a table having empcode , Name, deptname, & basic From the existing tables along with the records of the employee who are in a particular department (say, d1) and with a basic Rs. 7000/-
b) From the existing table, add the employees with the basic salary greater than or equal to 7000/-
c) Alter the table to add a net pay column.
d) Replace net pay with 1.5* Basic.
e) Try to remove the net pay column.
[It may require no. of steps]

Solution:

Part a:

```
CREATE TABLE FILTEREDEMPLOYEE AS
SELECT EMPLOYEE.EMP_CODE, EMPLOYEE.EMP_NAME, DEPARTMENT.DEPT_NAME, EMPLOYEE.BASIC
FROM EMPLOYEE, DEPARTMENT
WHERE EMPLOYEE.DEPT_CODE = DEPARTMENT.DEPT_CODE
AND DEPARTMENT.DEPT_CODE = 'D001'
AND EMPLOYEE.BASIC = 7000;

MariaDB [assignment_3]> SELECT * FROM FILTEREDEMPLOYEE;
+----------+------------+-----------+-------+
| EMP_CODE | EMP_NAME   | DEPT_NAME | BASIC |
+----------+------------+-----------+-------+
| E0011    | Jane Smith | Sales     |  7000 |
+----------+------------+-----------+-------+
```

Part b:

```
INSERT INTO FILTEREDEMPLOYEE (EMP_CODE, EMP_NAME, DEPT_NAME, BASIC)
SELECT EMPLOYEE.EMP_CODE, EMPLOYEE.EMP_NAME, DEPARTMENT.DEPT_NAME, EMPLOYEE.BASIC
FROM EMPLOYEE, DEPARTMENT
WHERE EMPLOYEE.DEPT_CODE = DEPARTMENT.DEPT_CODE AND BASIC >= 7000;
```

```
MariaDB [assignment_3]> SELECT * FROM FILTEREDEMPLOYEE;
+----------+-----------------+-----------+-------+
| EMP_CODE | EMP_NAME        | DEPT_NAME | BASIC |
+----------+-----------------+-----------+-------+
| E0011    | Jane Smith      | Sales     |  7000 |
| E0011    | Jane Smith      | Sales     |  7000 |
| E0012    | Michael Johnson | Marketing |  8000 |
| E0014    | David Wilson    | Finance   |  9000 |
+----------+-----------------+-----------+-------+
```

Part c:

```
ALTER TABLE FILTEREDEMPLOYEE
ADD COLUMN NET_PAY INTEGER;
```

```
MariaDB [assignment_3]> SELECT * FROM FILTEREDEMPLOYEE;
+----------+-----------------+-----------+-------+---------+
| EMP_CODE | EMP_NAME        | DEPT_NAME | BASIC | NET_PAY |
+----------+-----------------+-----------+-------+---------+
| E0011    | Jane Smith      | Sales     |  7000 |    NULL |
| E0011    | Jane Smith      | Sales     |  7000 |    NULL |
| E0012    | Michael Johnson | Marketing |  8000 |    NULL |
| E0014    | David Wilson    | Finance   |  9000 |    NULL |
+----------+-----------------+-----------+-------+---------+
```

Part d:

```
UPDATE FILTEREDEMPLOYEE
SET NET_PAY = 1.5 * BASIC;
```

```
MariaDB [assignment_3]> SELECT * FROM FILTEREDEMPLOYEE;
+----------+-----------------+-----------+-------+---------+
| EMP_CODE | EMP_NAME        | DEPT_NAME | BASIC | NET_PAY |
+----------+-----------------+-----------+-------+---------+
| E0011    | Jane Smith      | Sales     |  7000 |   10500 |
| E0011    | Jane Smith      | Sales     |  7000 |   10500 |
| E0012    | Michael Johnson | Marketing |  8000 |   12000 |
| E0014    | David Wilson    | Finance   |  9000 |   13500 |
+----------+-----------------+-----------+-------+---------+
```

Part e:

```
ALTER TABLE FILTEREDEMPLOYEE
DROP COLUMN NET_PAY;

MariaDB [assignment_3]> SELECT * FROM FILTEREDEMPLOYEE;
+----------+-----------------+-----------+-------+
| EMP_CODE | EMP_NAME        | DEPT_NAME | BASIC |
+----------+-----------------+-----------+-------+
| E0011    | Jane Smith      | Sales     |  7000 |
| E0011    | Jane Smith      | Sales     |  7000 |
| E0012    | Michael Johnson | Marketing |  8000 |
| E0014    | David Wilson    | Finance   |  9000 |
+----------+-----------------+-----------+-------+
```

Question 6:

Drop all the tables that you have created.

Solution:

```
DROP TABLE FILTEREDEMPLOYEE;
DROP TABLE LEAVE_RECORD;
DROP TABLE EMPLOYEE;
DROP TABLE DEPARTMENT;
```

Assignment: Set - 4

Assignment – 4

Question 1:

a) Create EMP table with ECODE (primary key), ENAME, DCODE, GRADE, BASIC & JN-DT as the columns. [Except BASIC & JN-DT, all columns are of char type and site of Grade is 1.]

b) Insert number of rows.

Solution:

```
CREATE TABLE EMP (
    ECODE VARCHAR(16) PRIMARY KEY,
    ENAME VARCHAR(50),
    DCODE VARCHAR(16),
    GRADE CHAR(1),
    BASIC INTEGER,
    JN_DT DATE
);

INSERT INTO EMP (ECODE, ENAME, DCODE, GRADE, BASIC, JN_DT)
VALUES
    ('E001', 'John Doe', 'D001', 'A', 5000, '2022-01-01'),
    ('E002', 'Jane Smith', 'D002', 'B', 6000, '2022-02-15'),
    ('E003', 'Mike Johnson', 'D001', 'A', 5500, '2022-03-10'),
    ('E004', 'Emily Davis', 'D002', 'B', 7000, '2022-04-05'),
    ('E005', 'David Wilson', 'D003', 'A', 6500, '2022-05-20'),
    ('E006', 'Sarah Thompson', 'D003', 'B', 5500, '2022-06-12'),
    ('E007', 'Michael Brown', 'D001', 'C', 4800, '2022-07-03');


MariaDB [assignment_4]> SELECT * FROM EMP;
+-------+----------------+-------+-------+-------+------------+
| ECODE | ENAME          | DCODE | GRADE | BASIC | JN_DT      |
+-------+----------------+-------+-------+-------+------------+
| E001  | John Doe       | D001  | A     |  5000 | 2022-01-01 |
| E002  | Jane Smith     | D002  | B     |  6000 | 2022-02-15 |
| E003  | Mike Johnson   | D001  | A     |  5500 | 2022-03-10 |
| E004  | Emily Davis    | D002  | B     |  7000 | 2022-04-05 |
| E005  | David Wilson   | D003  | A     |  6500 | 2022-05-20 |
| E006  | Sarah Thompson | D003  | B     |  5500 | 2022-06-12 |
| E007  | Michael Brown  | D001  | C     |  4800 | 2022-07-03 |
+-------+----------------+-------+-------+-------+------------+
```

Question 2:

Change the column heading as shown below, So that in subsequent SELECT statement newly set heading will be shown:

ECODE EMPLOYEE CODE

ENAME NAME

DCODE DEPT.CODE

JN-DT JONING DATE

Solution:

```
ALTER TABLE EMP
RENAME COLUMN ECODE TO `EMPLOYEE CODE`,
RENAME COLUMN ENAME TO `NAME`,
RENAME COLUMN DCODE TO `DEPT.CODE`,
RENAME COLUMN JN_DT TO `JOINING DATE`;
```

```
MariaDB [assignment_4]> SELECT * FROM EMP;
```

EMPLOYEE CODE	NAME	DEPT.CODE	GRADE	BASIC	JOINING DATE
E001	John Doe	D001	A	5000	2022-01-01
E002	Jane Smith	D002	B	6000	2022-02-15
E003	Mike Johnson	D001	A	5500	2022-03-10
E004	Emily Davis	D002	B	7000	2022-04-05
E005	David Wilson	D003	A	6500	2022-05-20
E006	Sarah Thompson	D003	B	5500	2022-06-12
E007	Michael Brown	D001	C	4800	2022-07-03

Question 3:

Set the format of columns as mentioned below, So that in subsequent SELECT statement, values appear in the specified format:

*format of BASIC is such that a value of 7000 will be shown as7,000

*Format of GRADE will be such that full column name appears in the display.

*For JN-DT format is such that 01-JAN-00 will be shown as JANURY 01, 2000

Solution:

```
SELECT
      ECODE,
      ENAME,
      DCODE,
      FORMAT(BASIC, 0) as BASIC,
      GRADE,
    date_format(JN_DT, '%M-%d-%Y') as JN-DT
FROM EMP;
```

```
MariaDB [assignment_4]> SELECT ECODE, ENAME, DCODE,
    -> FORMAT(BASIC, 0) AS BASIC,
    -> GRADE,
    -> DATE_FORMAT(JN_DT, '%M-%d-%Y') AS JN_DT
    -> FROM EMP;

+-------+----------------+-------+-------+-------+------------------+
| ECODE | ENAME          | DCODE | BASIC | GRADE | JN_DT            |
+-------+----------------+-------+-------+-------+------------------+
| E001  | John Doe       | D001  | 5,000 | A     | January-01-2022  |
| E002  | Jane Smith     | D002  | 6,000 | B     | February-15-2022 |
| E003  | Mike Johnson   | D001  | 5,500 | A     | March-10-2022    |
| E004  | Emily Davis    | D002  | 7,000 | B     | April-05-2022    |
| E005  | David Wilson   | D003  | 6,500 | A     | May-20-2022      |
| E006  | Sarah Thompson | D003  | 5,500 | B     | June-12-2022     |
| E007  | Michael Brown  | D001  | 4,800 | C     | July-03-2022     |
+-------+----------------+-------+-------+-------+------------------+
```

Question 4:

a) Show the display attributes of all the columns.
b) Show the display attributes of particular column.
c) Suppress the newly set attributes of JN-DT .Try a select statement.
d) Reset the newly set attributes of JN-DT
e) Reset the newly set attributes of all columns.
f) Shown the display attributes of all columns.

Solution:

```
DESC EMP;
MariaDB [assignment_4]> DESC EMP;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| ECODE | varchar(16) | NO   | PRI | NULL    |       |
| ENAME | varchar(50) | YES  |     | NULL    |       |
| DCODE | varchar(16) | YES  |     | NULL    |       |
| GRADE | char(1)     | YES  |     | NULL    |       |
| BASIC | int(11)     | YES  |     | NULL    |       |
| JN_DT | date        | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+


DESC EMP ENAME;
MariaDB [assignment_4]> DESC EMP ENAME;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| ENAME | varchar(50) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+


ALTER TABLE EMP MODIFY COLUMN JOINING_DATE DATE FORMAT 'YYYY-MM-DD';
ALTER TABLE EMP MODIFY COLUMN EMPLOYEE_CODE INT;
ALTER TABLE EMP MODIFY COLUMN NAME VARCHAR(20);
ALTER TABLE EMP MODIFY COLUMN DEPARTMENT_CODE CHAR(1);
ALTER TABLE EMP MODIFY COLUMN GRADE CHAR(1);
ALTER TABLE EMP MODIFY COLUMN BASIC FLOAT;
ALTER TABLE EMP MODIFY COLUMN JOINING_DATE DATE;

MariaDB [ASSIGNMENT_4]> DESC EMP;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| ECODE | int(1)      | NO   | PRI | NULL    |       |
| ENAME | varchar(20) | YES  |     | NULL    |       |
| DCODE | varchar(10) | YES  |     | NULL    |       |
| GRADE | char(1)     | YES  |     | NULL    |       |
| BASIC | float       | YES  |     | NULL    |       |
| JN_DT | date        | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
```