Dipesh Kafle Lab 1

DBMS LAB 1

Question 1

1.Consider the Following Database:
A software company wants to track project details
Employee(Empid , Empname, Address, Doj, Salary)
Empid as Primary key
Project (Projectno, Duration, Projectname).
Project no as Primary Key
Workson(Empid,Projno).
Empid as Foreign key references Employee
Projectno as Foreign key references Project
1. Display the Employee details in the descending order based on name.
2. Display the project details if project id is given.
3. Display the employee names starting with 'B'
4. Display the employee ID's working in a particular project
if project no is given.

Creating tables

```
--------------------------------------
--------------------------------------
-- creating tables

create table if not exists Employee(
    Empid int,
    Empname varchar(50),
    Address varchar(255),
    Doj Date,
    Salary float,
    PRIMARY KEY(Empid)
);

create table if not exists Project(
```

```
    Projectno int,
    Duration int,
    Projectname varchar(255),
    PRIMARY KEY(Projectno)
);


create table if not exists Workson(
    Empid int,
    Projno int,
    FOREIGN KEY(Empid)
        REFERENCES Employee(Empid),
    FOREIGN KEY(Projno)
        REFERENCES Project(Projectno),
    PRIMARY KEY(Empid,Projno)
);
```

Populating tables

```
----------------------------------------
----------------------------------------

-- populating tables

-- Employee table
insert into Employee values (1,'Dipesh Kafle', 'Nepal',CURRENT_DATE, 1000);
insert into Employee values (2,'Ram ', 'India',CURRENT_DATE, 2000);
insert into Employee values (3,'Sita', 'Nepal',CURRENT_DATE, 1500);
insert into Employee values (4,'Bharat', 'India',CURRENT_DATE, 500);

--  Project table
insert into Project values(1,10,'Find a job');
insert into Project values(2,1,'Marry Sita');
insert into Project values(3,1,'Marry Ram');

-- Workson table
insert into Workson values(1,1);
insert into Workson values(2,2);
insert into Workson values(3,3);
insert into Workson values(4,1);
```

1.1

```
-------------------------------------------------
-- Part 1
select * from Employee order by Empname DESC;
```

- Output

```
 empid |   empname    | address |    doj     | salary
-------+--------------+---------+------------+--------
     3 | Sita         | Nepal   | 2021-09-02 |   1500
     2 | Ram          | India   | 2021-09-02 |   2000
     1 | Dipesh Kafle | Nepal   | 2021-09-02 |   1000
     4 | Bharat       | India   | 2021-09-02 |    500
(4 rows)
```

1.2

```
-------------------------------------------------
-------------------------------------------------
-- Part2


-- valid
select * from Project where Projectno = 1;
-- valid
select * from Project where Projectno = 2;
-- valid
select * from Project where Projectno = 3;
-- not valid, will give empty
select * from Project where Projectno = 4;
```

- Output

```
select * from Project where Projectno = 1;
 projectno | duration | projectname
-----------+----------+-------------
         1 |       10 | Find a job
(1 row)

select * from Project where Projectno = 2;
 projectno | duration | projectname
-----------+----------+-------------
         2 |        1 | Marry Sita
(1 row)

select * from Project where Projectno = 3;
```

```
 projectno | duration | projectname
-----------+----------+-------------
         3 |        1 | Marry Ram
(1 row)

select * from Project where Projectno = 4;
 projectno | duration | projectname
-----------+----------+-------------
(0 rows)
```

1.3

```
select Empname from Employee where Empname like 'B%';
```

- Output

```
select Empname from Employee where Empname like 'B%';
 empname
---------
 Bharat
(1 row)
```

1.4

```
-- Part 4

select Empid from Workson where Projno=1;
select Empid from Workson where Projno=2;
select Empid from Workson where Projno=3;
```

- Output

```
select Empid from Workson where Projno=1;
 empid
-------
     1
     4
(2 rows)

select Empid from Workson where Projno=2;
 empid
-------
     2
(1 row)

select Empid from Workson where Projno=3;
 empid
```

```
-------
     3
(1 row)
```

Question 2

```
Student(Rollno, Name, Marks(of 6 subjects),total)
Rollno as Primary key
Department(Deptid, Deptname, HOD name) and
Deptid as Primary key
StudDep(Rollno, Deptid).
Rollno as foreign key references Student
Deptid as foreign key references DepartmentThe total field is updated automatically
1.Insert 10 student details and 3 department details. Insert
details in the studdep table.
2.Display the Student details if deptid is given
3.Display the department details if rollno is given
4.Display the student names who got total greater than 500
5.Display the HOD name of the CSE department
6.Display the student rollnos of the CSE department
```

Creating tables

```sql
-- schema for tables
create table if not exists Students (
    Rollno int,
    Name varchar(50) not null,
    Marks int [ 6 ] not null,
    total int generated always as (
        Marks[1]+Marks[2]+Marks[3]+Marks[4]+Marks[5]+Marks[6]
    ) STORED,
    PRIMARY KEY (Rollno)
);

create table if not exists Department (
    Deptid int,
    Deptname varchar (255) not null,
    HODName varchar(50) not null,
    PRIMARY KEY (Deptid)
);

create table if not exists StudDep (
    Rollno int,
    Deptid int,
    FOREIGN KEY (Rollno) REFERENCES Students (Rollno),
    FOREIGN KEY (Deptid) REFERENCES Department(Deptid),
```

```
    PRIMARY KEY(Rollno, Deptid)
);
```

2.1

```
-------------------------------------------------------------------------------
-- 2.1

-- 1) Insert 10 student details and 3 department details. Insert
-- details in the studdep table.
-------------------------------------------------------------------------------
-- Student details
--i
insert into
    Students(Rollno, Name, Marks)
values
    (106119001, 'Dipesh', Array [ 80,75, 81,87, 77,72 ]);

--ii
insert into
    Students(Rollno, Name, Marks)
values
    (106119002,'Ram', Array [ 86, 75,81, 87,77,77 ]);

--iii
insert into
    Students(Rollno, Name, Marks)
values
    (106119003,'Sita',Array [ 80,75, 81,92,77,80 ]);

--iv
insert into
    Students(Rollno, Name, Marks)
values
    (106119004,'Laxman',Array [ 90,75, 71,87,77,70 ]);

--v
insert into
    Students(Rollno, Name, Marks)
values
    (106119005,'Bharat',Array [ 91,90,95,95,95,95 ]);

--vi
insert into
    Students(Rollno, Name, Marks)
values
```

```
    (107119001,'Arjun',Array [ 81,90,85,98,75,95 ]);

--vii
insert into
    Students(Rollno, Name, Marks)
values
    (107119002,'Bheem',Array [ 81, 70,95,70,85,85 ]);

--viii
insert into
    Students(Rollno, Name, Marks)
values
    (108119001,'Vishnu',Array [ 91,90,75,85,87,95 ]);

--ix
insert into
    Students(Rollno, Name, Marks)
values
    (108119002,'Shiva',Array [ 94,70,85,95,95,95 ]);

-- x
insert into
    Students(Rollno, Name, Marks)
values
    (108119003,'Laxmi',Array [ 84,70,88,75,85,85 ]);

-- 3 department details
insert into
    Department
values
    (1, 'CSE', 'Rajeswari  Sridhar');

insert into
    Department
values
    (2, 'ECE', 'Prof X');

insert into
    Department
values
    (3, 'EEE', 'Prof Y');

-- studdep details
insert into
    studdep
values
```

```
    (106119001, 1);

insert into
    studdep
values
    (106119002, 1);

insert into
    studdep
values
    (106119003, 1);

insert into
    studdep
values
    (106119004, 1);

insert into
    studdep
values
    (106119005, 1);

insert into
    studdep
values
    (107119001, 2);

insert into
    studdep
values
    (107119002, 2);

insert into
    studdep
values
    (108119001, 3);

insert into
    studdep
values
    (108119002, 3);

insert into
    studdep
values
    (108119003, 3);
```

2.2

```
-------------------------------------------------------------------------------
-- 2.2

-- 2.Display the Student details if deptid is given
-------------------------------------------------------------------------------

select * from Students inner join Studdep
on
    students.Rollno=studdep.Rollno
where
    studdep.deptid=1;
select * from Students inner join Studdep
on
    students.Rollno=studdep.Rollno
where
    studdep.deptid=2;

select * from Students inner join Studdep
on
    students.Rollno=studdep.Rollno
where
    studdep.deptid=3;
```

- Output

```
select * from Students inner join Studdep
on
    students.Rollno=studdep.Rollno
where
    studdep.deptid=1;
  rollno   |  name  |        marks        | total |  rollno   | deptid
-----------+--------+---------------------+-------+-----------+--------
 106119001 | Dipesh | {80,75,81,87,77,72} |   472 | 106119001 |      1
 106119002 | Ram    | {86,75,81,87,77,77} |   483 | 106119002 |      1
 106119003 | Sita   | {80,75,81,92,77,80} |   485 | 106119003 |      1
 106119004 | Laxman | {90,75,71,87,77,70} |   470 | 106119004 |      1
 106119005 | Bharat | {91,90,95,95,95,95} |   561 | 106119005 |      1
(5 rows)

select * from Students inner join Studdep
on
    students.Rollno=studdep.Rollno
where
    studdep.deptid=2;
```

```
  rollno   | name  |        marks        | total |  rollno   | deptid
-----------+-------+---------------------+-------+-----------+--------
 107119001 | Arjun | {81,90,85,98,75,95} |   524 | 107119001 |      2
 107119002 | Bheem | {81,70,95,70,85,85} |   486 | 107119002 |      2
(2 rows)

select * from Students inner join Studdep
on
    students.Rollno=studdep.Rollno
where
    studdep.deptid=3;
  rollno   |  name  |        marks        | total |  rollno   | deptid
-----------+--------+---------------------+-------+-----------+--------
 108119001 | Vishnu | {91,90,75,85,87,95} |   523 | 108119001 |      3
 108119002 | Shiva  | {94,70,85,95,95,95} |   534 | 108119002 |      3
 108119003 | Laxmi  | {84,70,88,75,85,85} |   487 | 108119003 |      3
(3 rows)
```

2.3

```
-------------------------------------------------------------------------------
-- 2.3

-- 3.Display the department details if rollno is given
-------------------------------------------------------------------------------

select * from Department inner join Studdep
on
    Department.deptid=Studdep.deptid
where
    Studdep.rollno = 106119001;

select * from Department inner join Studdep
on
    Department.deptid=Studdep.deptid
where
    Studdep.rollno = 107119001;

select * from Department inner join Studdep
on
    Department.deptid=Studdep.deptid
where
    Studdep.rollno = 108119001;
```

- Output

```
select * from Department inner join Studdep
on
    Department.deptid=Studdep.deptid
where
    Studdep.rollno = 106119001;
 deptid | deptname |      hodname       |  rollno   | deptid
--------+----------+--------------------+-----------+--------
      1 | CSE      | Rajeswari  Sridhar | 106119001 |      1
(1 row)

select * from Department inner join Studdep
on
    Department.deptid=Studdep.deptid
where
    Studdep.rollno = 107119001;
 deptid | deptname | hodname |  rollno   | deptid
--------+----------+---------+-----------+--------
      2 | ECE      | Prof X  | 107119001 |      2
(1 row)

select * from Department inner join Studdep
on
    Department.deptid=Studdep.deptid
where
    Studdep.rollno = 108119001;
 deptid | deptname | hodname |  rollno   | deptid
--------+----------+---------+-----------+--------
      3 | EEE      | Prof Y  | 108119001 |      3
(1 row)
```

2.4

```
-------------------------------------------------------------------------------
-- 2.4

-- 4.Display the student names who got total greater than 500
-------------------------------------------------------------------------------

select Name , total from students where total > 500;
```

- Output

```
select Name , total from students where total > 500;
  name  | total
--------+-------
 Bharat |   561
```

```
 Arjun  |   524
 Vishnu |   523
 Shiva  |   534
(4 rows)
```

2.5

```
-------------------------------------------------------------------------------
-- 2.5

-- 5.Display the HOD name of the CSE department
-------------------------------------------------------------------------------

select HODName from Department where Deptname = 'CSE';
```

- Output

```
select HODName from Department where Deptname = 'CSE';
      hodname
--------------------
 Rajeswari  Sridhar
(1 row)
```

2.6

```
-------------------------------------------------------------------------------
-- 2.6

-- 6.Display the student rollnos of the CSE department
-------------------------------------------------------------------------------

select studdep.rollno, Department.deptname from Department inner join studdep
on
    studdep.deptid = Department.deptid
where
    Department.Deptname='CSE';
```

- Output

```
select studdep.rollno, Department.deptname from Department inner join studdep
on
    studdep.deptid = Department.deptid
where
    Department.Deptname='CSE';
  rollno   | deptname
-----------+----------
 106119001 | CSE
```

```
 106119002 | CSE
 106119003 | CSE
 106119004 | CSE
 106119005 | CSE
(5 rows)
```

Question 3

```
3.Consider the Following Database:
salesperson(ssn, name, start_year, dept_no)
ssn - Primary Key
trip(ssn, from_city, to_city, departure_date, return_date, trip_id))
ssn - Foreign key
trip_id - Primary key
salerep_expense(trip_id, expense_type,amount)
trip_id - Foreign key
The expense types are 'TRAVEL', 'STAY' and 'FOOD'
1.Give the details (all attributes of trip relation) for trips that
exceed Rs2000.2.Print the ssn of salesperson who took trips to chennai more than once
3.Print the total trip expenses incurred by the salesperson with ssn = 1000
4.Display the salesperson details in the sorted order based on name.
```

Creating tables

```
-------------------------------------------------------------
-- salesperson(ssn, name, start_year, dept_no)
-- ssn - Primary Key

-- trip(ssn, from_city, to_city, departure_date, return_date, trip_id))
-- ssn - Foreign key
-- trip_id - Primary key

-- salerep_expense(trip_id, expense_type,amount)
-- trip_id - Foreign key
-------------------------------------------------------------

create table if not exists salesperson(
    ssn int,
    name varchar(50),
    start_year int,
    dept_no int,
    Primary key(ssn)
);

create table if not exists trip(
    ssn int ,
```

```
    from_city varchar(100),
    to_city varchar(100),
    departure_date DATE,
    return_date DATE,
    trip_id int primary key,
    Foreign key(ssn) REFERENCES salesperson(ssn)
);

-- The expense types are 'TRAVEL', 'STAY' and 'FOOD'
create type exp_type as ENUM ('TRAVEL','STAY','FOOD');
create table if not exists salerep_expense(
    trip_id int ,
    expense_type exp_type,
    amount float,
    foreign key(trip_id) REFERENCES trip(trip_id),
    primary key(trip_id,expense_type)
);
```

Populating tables

```
-----------------------------------------------------------------
-- Inserting dummy data
-----------------------------------------------------------------
-- Filling salesperson db
insert into salesperson values(1,'Dipesh',2020,1);
insert into salesperson values(2,'Ram',2021,2);
insert into salesperson values(1000,'Shyam',2019,3);

-- Filling trip db
insert into trip values(1,'Kathmandu','Trichy','2019-07-22','2019-07-30',1);
insert into trip values(2,'Trichy','Chennai','2019-12-30','2020-12-31',2);
insert into trip values(2,'Trichy','Chennai','2019-12-30','2020-12-31',4);
insert into trip values(1000,'Pune','Chennai','2020-12-30','2021-01-02',3);

-- Filling salerep_expense db
insert into salerep_expense values(1,'TRAVEL',5000);
insert into salerep_expense values(1,'STAY',1000);
insert into salerep_expense values(1,'FOOD',300);
insert into salerep_expense values(2,'TRAVEL',1200);
insert into salerep_expense values(2,'FOOD',300);
insert into salerep_expense values(3,'TRAVEL',800);
insert into salerep_expense values(3,'FOOD',300);
insert into salerep_expense values(3,'STAY',1000);
insert into salerep_expense values(4,'TRAVEL',1200);
insert into salerep_expense values(4,'FOOD',300);
```

3.1

```
-------------------------------------------------------------
--3.1

-- 1. Give the details (all attributes of trip relation) for trips that
-- exceed Rs2000.
-------------------------------------------------------------
-- select * from (
--     select trip.* , sum(salerep_expense.amount) as cost
--     from trip inner join salerep_expense
--     on
--         trip.trip_id = salerep_expense.trip_id
--     group by trip.trip_id) as temp
-- where temp.cost > 2000;


select trip.* , sum(salerep_expense.amount) as cost
    from trip inner join salerep_expense
    on
        trip.trip_id = salerep_expense.trip_id
    group by trip.trip_id
    having sum(salerep_expense.amount) > 2000;
```

- Output

```
select trip.* , sum(salerep_expense.amount) as cost
    from trip inner join salerep_expense
    on
        trip.trip_id = salerep_expense.trip_id
    group by trip.trip_id
    having sum(salerep_expense.amount) > 2000;
 ssn  | from_city | to_city | departure_date | return_date | trip_id | cost
------+-----------+---------+----------------+-------------+---------+------
 1000 | Pune      | Chennai | 2020-12-30     | 2021-01-02  |       3 | 2100
    1 | Kathmandu | Trichy  | 2019-07-22     | 2019-07-30  |       1 | 6300
(2 rows)
```

3.2

```
-------------------------------------------------------------
-- 3.2

-- 2.Print the ssn of salesperson who took trips to chennai more than once
-------------------------------------------------------------
-- select temp2.ssn from
--     (select temp.ssn , count(temp.ssn) as to_chennai_trip_cnt
```

```
--     from (
--         select trip.ssn from trip  where trip.to_city = 'Chennai'
--         ) as temp
--     group by temp.ssn) as temp2
-- where temp2.to_chennai_trip_cnt > 1;

select temp.ssn from (
    select trip.ssn from trip where trip.to_city='Chennai'
) as temp group by temp.ssn having count(temp.ssn) > 1;
```

- Output

```
select temp.ssn from (
    select trip.ssn from trip where trip.to_city='Chennai'
) as temp group by temp.ssn having count(temp.ssn) > 1;
 ssn
-----
   2
(1 row)
```

3.3

```
-------------------------------------------------------------
-- 3.3

-- 3.Print the total trip expenses incurred by
-- the salesperson with ssn = 1000
-------------------------------------------------------------
select sum(tmp2.amount) from (select * from (
    select trip_id from trip
    where trip.ssn = 1000) as tmp inner join salerep_expense
    on tmp.trip_id = salerep_expense.trip_id) as tmp2;
```

- Output

```
select sum(tmp2.amount) from (select * from (
    select trip_id from trip
    where trip.ssn = 1000) as tmp inner join salerep_expense
    on tmp.trip_id = salerep_expense.trip_id) as tmp2;
 sum
------
 2100
(1 row)
```

3.4

```
-------------------------------------------------------------
-- 3.4
```

```
-- 4.Display the salesperson details in the sorted order
-- based on name.
-----------------------------------------------------------------
select * from salesperson order by name ;
```

- Output

```
select * from salesperson order by name ;
 ssn  |  name  | start_year | dept_no
------+--------+------------+---------
    1 | Dipesh |       2020 |       1
    2 | Ram    |       2021 |       2
 1000 | Shyam  |       2019 |       3
(3 rows)
```

Question 4

```
4.Consider the Following Database:
car(serial_no, model, manufacturer, price)
serial_no - Primary key
options(serial_no, option_name, price)
serial_no - Foreign key
sales(salesperson_id, serial_no, date, sale_price)
serial_no - Foreign key
salesperson_id - Foreign key
salesperson(salesperson_id, name, phone)
salesperson_id - Primary key
1.For the sales person named 'John' list the following information for all
the cars sold : serial no, manufacturer, sale_price
2.List the serial_no and model of cars that have no options
3.List the serial_no, model, sale_price for the cars that have optional parts.
4.Modify the phone no of a particular sales person
```

Creating tables

```
create table car(
    serial_no int,
    model varchar(255),
    manufacturer varchar(255),
    price float,
    primary key(serial_no)
);

create table options(
    serial_no int,
    option_name varchar(255),
```

```
    price float,
    foreign key(serial_no) REFERENCES car(serial_no)
);

create table salesperson(
    salesperson_id int,
    name varchar(50),
    phone varchar(10),
    primary key (salesperson_id)
);

create table sales(
    salesperson_id int,
    serial_no int,
    date DATE,
    sale_price float,
    foreign key(salesperson_id) REFERENCES salesperson(salesperson_id),
    foreign key(serial_no) REFERENCES car(serial_no)
);
```

Populating tables

```
-----------------------------------------------------------------------
-- dummy data
-----------------------------------------------------------------------

insert into car values(1,'Nano','Tata',150000);
insert into car values(2,'Model S','Tesla',6000000);

insert into options values(2,'Matte black coloring',6100000);
insert into options values(2,'Red coloring',6050000);

insert into salesperson values(1,'John','0123456789');
insert into salesperson values(2,'Dipesh','9876543210');

insert into sales values(2,1,CURRENT_DATE, 170000 );
insert into sales values(2,2,CURRENT_DATE, 6200000);
insert into sales values(1,1,CURRENT_DATE, 180000);
insert into sales values(1,1,CURRENT_DATE, 160000);
```

4.1

```
-----------------------------------------------------------------------
-- 4.1

-- 1.For the sales person named 'John' list the following information for all
```

```
-- the cars sold : serial no, manufacturer, sale_price
-----------------------------------------------------------------
select temp2.serial_no, car.manufacturer, temp2.sale_price  from
    (select sales.serial_no,sales.sale_price from (select salesperson_id from salesperson
        where
        name='John') as temp inner join sales
        on temp.salesperson_id = sales.salesperson_id) as temp2 inner join car
        on temp2.serial_no = car.serial_no;
```

- Output

```
select temp2.serial_no, car.manufacturer, temp2.sale_price  from
    (select sales.serial_no,sales.sale_price from (select salesperson_id from salesperson
        where
        name='John') as temp inner join sales
        on temp.salesperson_id = sales.salesperson_id) as temp2 inner join car
        on temp2.serial_no = car.serial_no;
 serial_no | manufacturer | sale_price
-----------+--------------+------------
         1 | Tata         |     180000
         1 | Tata         |     160000
(2 rows)
```

4.2

```
-----------------------------------------------------------------
-- 4.2

-- 2.List the serial_no and model of cars that have no options
-----------------------------------------------------------------
select temp.serial_no,car.model from (select serial_no from car
    EXCEPT
    select serial_no from options) as temp inner join car
    on car.serial_no = temp.serial_no;
```

- Output

```
select temp.serial_no,car.model from (select serial_no from car
    EXCEPT
    select serial_no from options) as temp inner join car
    on car.serial_no = temp.serial_no;
 serial_no | model
-----------+-------
         1 | Nano
(1 row)
```

4.3

```
-----------------------------------------------------------------------
-- 4.3

-- 3.List the serial_no, model, sale_price
-- for the cars that have optional parts.
-----------------------------------------------------------------------

select temp.serial_no,car.model,car.price from (select serial_no from car
    INTERSECT
    select serial_no from options) as temp inner join car
    on car.serial_no = temp.serial_no;
```

- Output

```
select temp.serial_no,car.model,car.price from (select serial_no from car
    INTERSECT
    select serial_no from options) as temp inner join car
    on car.serial_no = temp.serial_no;
 serial_no |  model  |  price
-----------+---------+---------
         2 | Model S | 6000000
(1 row)
```

4.4

```
-----------------------------------------------------------------------
-- 4.4

-- 4.Modify the phone no of a particular sales person
-----------------------------------------------------------------------

-- Before
select * from salesperson where name = 'Dipesh';

-- Update
update salesperson
set phone = '1235647890'
where name = 'Dipesh';

-- After
select * from salesperson where name = 'Dipesh';
```

- Output

```
select * from salesperson where name = 'Dipesh';
 salesperson_id |  name  |   phone
----------------+--------+------------
              2 | Dipesh | 9876543210
(1 row)

update salesperson
set phone = '1235647890'
where name = 'Dipesh';
UPDATE 1
select * from salesperson where name = 'Dipesh';
 salesperson_id |  name  |   phone
----------------+--------+------------
              2 | Dipesh | 1235647890
(1 row)
```