DATABEAT ASSESSMENT
AARUSHI SINGH

Section-2

1) Write a SQL query to create these tables in your database and insert the data into these tables with the following characteristics:
a. Add the primary key “Emp_ID” to the Employees Table. Also, mention what are the constraints used in SQL.
b. Add foreign key “EMP_REF_ID” in Variables Details and Designation Table that references “Emp_ID” in Employees Table

Sol:-

First **creating employees table**-

```
use assignment;

Create table Employees(
EMP_ID INT(5) primary KEY,                    // 1) a) already assigned primary key
FIRST_NAME VARCHAR(10),
LAST_NAME VARCHAR(10),
SALARY INT(7),
JOINING_DATE DATE,
DEPARTMENT VARCHAR(8)
);

insert into Employees values(001,"Manish","Agarwal",700000,"2019-04-20","HR");
insert into Employees values(002,"Niranjan","Bose",20000,"2019-02-11","DA");
insert into Employees values(003,"Vivek","Singh",100000,"2019-01-20","DA");
insert into Employees values(004,"Asutosh","Kapoor",700000,"2019-03-20","HR");
insert into Employees values(005,"Vihaan","Banerjee",300000,"2019-06-11","DA");
insert into Employees values(006,"Atul","Diwedi",400000,"2019-05-11","Account");
insert into Employees
values(007,"Satyendra","Tripathi",95000,"2019-03-20","Account");
insert into Employees values(008,"Pritika","Bhatt",80000,"2019-02-11","DA");
```

Second **creating Variables table**-

```
Create table Variables(
EMP_REF_ID INT(3),
VARIABLES_DATE DATETIME,
VARIABLES_AMT INT(8),
FOREIGN KEY(EMP_REF_ID) REFERENCES Employees(EMP_ID)

// 1) b) adding foreign key

);
```

```
insert into Variables values(1,"2019-02-20 00:00:00",15000);
insert into Variables values(2,"2019-06-11 00:00:00",30000);
insert into Variables values(3,"2019-02-20 00:00:00",42000);
insert into Variables values(4,"2019-02-20 00:00:00",14500);
insert into Variables values(5,"2019-06-11 00:00:00",23500);
```

Third **creating Designation table**-

```
Create table Designation(
EMP_REF_ID INT(3),
EMP_TITLE VARCHAR(25),
AFFECTED_FROM DATE,
foreign key (EMP_REF_ID) references Employees(EMP_ID)

// 1) b) adding foreign key

);

insert into Designation values(1,"Asst. Manager","2019-02-20");
insert into Designation values(2,"Senior Analyst","2019-01-11");
insert into Designation values(8,"Senior Analyst","2019-04-06");
insert into Designation values(5,"Manager","2019-10-06");
insert into Designation values(4,"Asst. Manager","2019-12-06");
insert into Designation values(7,"Team Lead","2019-06-06");
insert into Designation values(6,"Team Lead","2019-09-06");
insert into Designation values(3,"Senior Analyst","2019-08-06");
```

CONSTRAINTS - Constraints limit the type of data that can go in a table. They can be applied at column level as well as table level.
Some of the commonly used constraints are -

a) Primary Key
b) Foreign Key
c) Not Null
d) Unique, etc.

2) Name the four different types of joins? Give examples of each by performing all the joins on the Employees table and Designation Table.
Sol-

TYPES OF JOINS-

a) Inner
b) Left

c) Right
d) Full

>> INNER - Returns all the rows from both the tables as long as the condition satisfies

Eg:- 1) Select * from Employees inner Join Designation on
Employees.EMP_ID = Designation.EMP_REF_ID;

2) Select FIRST_NAME, SALARY, EMP_TITLE from Employees inner Join Designation on Employees.EMP_ID = Designation.EMP_REF_ID;

>> LEFT - Returns all the rows from left side of the table present in Join

Eg:- 1) Select Employees.FIRST_NAME, Employees.SALARY from Employees left Join Designation on Employees.EMP_ID = Designation.EMP_REF_ID;

>> RIGHT - Returns all the rows from left side of the table present in Join

Eg:- 1)Select D.EMP_REF_ID, D.EMP_TITLE from Employees E right Join Designation D
On Employees.EMP_ID = Designation.EMP_REF_ID;

>> FULL - Returns all rows from both side of table

Eg:- 1) Select * from Employees Full Join Designation on
Employees.EMP_ID = Designation.EMP_REF_ID;

2) a) Write a query to get the employee details(columns - full name and department) of those who received the highest and the least variables
Sol:-

```
Select E.FIRST_NAME, E.LAST_NAME, E.Department from Employees E Join Variables V
on E.EMP_ID = V.EMP_REF_ID
Where V.VARIABLES_AMT in
( (select max(VARIABLES_AMT) from Variables), (select min(VARIABLES_AMT) from
Variables));
```

2) b)

```
select D.EMP_TITLE from Designation D join Employees E
on D.EMP_REF_ID = E.EMP_ID
join Variables V
on V.EMP_REF_ID = E.EMP_ID
where E.SALARY + V.VARIABLES_AMT
in(max(E.SALARY + V.VARIABLES_AMT), min(E.SALARY + V.VARIABLES_AMT));
```

2) c) Cross Join - Results in Cartesian Product, where it returns the number of rows in first table multiplied by the number of rows in second table.

Eg:- select * from Employees cross join Designation;

2) d) The clauses used with select statements are :

- Where
- Group By
- Having
- Order by

The preference order of above clauses are:-

1) Where
2) Group By
3) Having
4) Order By

3) Stored Procedure - It acts as a function, where you can write a query once and reuse it again and again.

3) a)

```
select D.EMP_REF_ID, E.FIRST_NAME, E.LAST_NAME, E.SALARY,
E.JOINING_DATE, E.DEPARTMENT
from Designation D
join Variables V
on D.EMP_REF_ID = V.EMP_REF_ID
join Employees E
on E.EMP_ID = V.EMP_REF_ID
where month(D.AFFECTED_FROM) > 6
order by V.VARIABLES_AMT;
```

3) b)

```
Use assignment;
Drop procedure if exists ‘SAMPLE1’;
Delimiter $$
Use ‘assignment’ $$

Create procedure ‘Sample1’
Begin
Select E.FIRST_NAME, E.LAST_NAME, E.Department from Employees E Join Variables V
on E.EMP_ID = V.EMP_REF_ID
Where V.VARIABLES_AMT in
( (select max(VARIABLES_AMT) from Variables), (select min(VARIABLES_AMT) from
Variables));
```

```
End $$;
delimiter;
```