SQL ASSIGNMENT

1 A)

```
create database company;

use company;

create table EmployeesTable(

Emp_Id int NOT NULL,

First_Name varchar(250),

Last_Name varchar(250),

Salary int(10),

Joining_Date datetime,

Department varchar(10),

PRIMARY KEY(Emp_Id)

);

create table VariablesDetails(

Emp_Ref_Id int,

Variables_Date datetime,

Variables_Amount int(10),

foreign key(Emp_Ref_Id) references EmployeesTable(Emp_Id)

);

create table DesignationTable(

Emp_Ref_Id int,

Emp_Title varchar(250),

Affected_From datetime,

foreign key(Emp_Ref_Id) references EmployeesTable(Emp_Id)

);
```

insert into EmployeesTable values (001,'Manish','Agarwal', 700000,'2019-04-20 09:00:00','HR'),

(002,'Niranjan', 'Bose',20000,'2019-02-11 09:00:00','DA'),(003,'Vivek','Singh', 100000,'2019-01-20 09:00:00','DA'),

(004,'Asutosh', 'Kapoor', 700000,'2019-03-20 09:00:00','HR'),(005,'Vihaan','Banerjee',300000,'2019-06-11 09:00:00','DA'),

(006,'Atul','Diwedi', 400000,'2019-05-11 09:00:00','Account'),(007,'Satyendra','Tripathi', 95000, '2019-03-20 09:00:00','Account'),

(008,'Pritika','Bhatt',80000, '2019-02-11 09:00:00','DA');

insert into variablesdetails values (1,'2019-02-20 00:00:00',15000),(2,'2019-06-11 00:00:00',30000),(3,'2019-02-20 00:00:00', 42000),

(4 ,'2019-02-20 00:00:00', 14500),(5,'2019-06-11 00:00:00',23500);

insert into designationtable values (1,'Asst. Manager', '2019-02-20 00:00:00'),(2,'Senior Analyst','2019-01-11 00:00:00'),

(8, 'Senior Analyst' ,'2019-04-06 00:00:00'),(5,'Manager','2019-10-06 00:00:00'),(4,'Asst. Manager','2019-12-06 00:00:00'),

(7,'Team Lead', '2019-06-06 00:00:00'),(6,'Team Lead','2019-09-06 00:00:00'),(3,'Senior Analyst','2019-08-06 00:00:00');

1a A)

SQL Constraints are specified rules for the data that can be entered into a column or data table, here are the SQL Constraints

NOT NULL	It Ensures that column does not contain NULL Values
CHECK	It Ensures that values in a column satisfies a specific condition
UNIQUE	It Ensures that all values in a column are distinct
PRIMARY KEY	It uniquely identifies a row in the table. It is a combination of NOT NULL and UNIQUE constraints.
FOREIGN KEY	It is Same Like a Primary Key. But it is used to identify a row uniquely in another table
DEFAULT	It Assigns a default value for a column if no value is specified
CREATE INDEX	It is Used to Create and Retrieve Data from Databases Very Quickly

---

2 A) There are mainly 5 types of joins They are INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN, SELF JOIN

-- INNER JOIN

select emp_id,first_name,last_name,emp_title,salary from employeestable e INNER JOIN designationtable d on e.emp_id=d.emp_ref_id;

-- LEFT JOIN

select emp_id,first_name,last_name,emp_title,salary,department from employeestable e LEFT JOIN designationtable d on e.emp_id=d.emp_ref_id;

-- RIGHT JOIN

select emp_id,first_name,last_name,emp_title,salary,department from employeestable e RIGHT JOIN designationtable d on e.emp_id=d.emp_ref_id;

-- FULL JOIN is not Supported In MYSQL Instead we use CROSS JOIN Which have same functionality of FULL JOIN

select * from employeestable FULL JOIN designationtable on employeestable.Emp_Id=designationtable.Emp_Ref_Id;

-- SELF JOIN (getting the list of employees whose salary is less than current employee)

select f.First_Name,e.First_Name,e.salary from employeestable e,employeestable f where e.salary<f.salary;

2a A)

select concat(first_name,' ',last_name) as full_name,department from employeestable e inner join variablesdetails v on e.Emp_Id=v.Emp_Ref_Id

where Emp_Id = (select Emp_Ref_Id from variablesdetails where Variables_Amount=(select max(Variables_Amount) from variablesdetails)) or

Emp_Id = (select Emp_Ref_Id from variablesdetails where Variables_Amount=(select min(Variables_Amount) from variablesdetails));

2b A)

select emp_title,sum(amount) as amount from designationtable d, (select e.emp_id, e.salary+v.Variables_Amount as amount from employeestable e inner join variablesdetails v on e.Emp_Id=v.Emp_Ref_Id where year(v.variables_date)='2019') b

where d.emp_ref_id=b.emp_id group by Emp_Title order by amount desc limit 2;

2c A)

Cross Join is Similar to Full Join Where it returns all matched records from both tables + unmatched records from left table +unmatched records from right table it means returns all records from both tables

select * from employeestable CROSS JOIN designationtable on employeestable.Emp_Id=designationtable.Emp_Ref_Id;

2d A) Clauses that can be used with Select statement are from, where, group by, having, order by, Distinct,Limit

Order of writing is

- Distinct

- From
- Where
- Group By
- Having
- Order By
- Limit

Order of Execution is

- From
- Where
- Group by
- Having
- Select
- Distinct
- Order By
- Limit

---

3 A) A Stored Procedure Can be Treated as a Function in Programming wherein we can reuse the function as many times we want and the same goes here in case of Stored Procedure. In which we can call the Procedure repeatedly without Writing the Code again. A Stored Procedure can be parameterized or normal Procedure and in case of Parameterized Procedure the parameters must be passed during Procedure Call.

3a A)

select emp_id,first_name,last_name,salary,Joining_Date,department,v.Variables_Date,v.Variables_Amount from employeestable e inner join variablesdetails v on e.Emp_Id=v.Emp_Ref_Id

where Emp_Ref_Id in(select emp_ref_id from designationtable where month(Affected_From) between 7 and 12) order by Variables_Amount desc;

3b A)

DELIMITER //

CREATE PROCEDURE getemployeedetails()

BEGIN

select concat(first_name,' ',last_name) as full_name,department from employeestable e inner join variablesdetails v on e.Emp_Id=v.Emp_Ref_Id

where Emp_Id = (select Emp_Ref_Id from variablesdetails where Variables_Amount=(select max(Variables_Amount) from variablesdetails)) or

```
Emp_Id = (select Emp_Ref_Id from variablesdetails where Variables_Amount=(select min(Variables_Amount)
from variablesdetails));

END //

DELIMITER ;

call getemployeedetails();
```