

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.

```
CREATE table Employees Table (  
    EMP_ID int NOT NULL,  
    FIRST NAME varchar(100),  
    LAST NAME varchar(100),  
    SALARY int,  
    JOINING DATE timestamp,  
    DEPARTMENT varchar(100).  
    PRIMARY KEY (EMP_ID)  
);
```

b. Add foreign key “EMP_REF_ID” in Variables Details and Designation Table that references “Emp_ID” in Employees Table.

```
CREATE table VariablesDetails(  
    EMP_REF_ID int,  
    VARIABLES_DATE timestamp,  
    VARIABLES_AMOUNT float,  
    FOREIGN KEY (EMP_REF_ID) REFERENCES EmployeesTable (EMP_ID)  
);
```

```
CREATE table DesignationTable(  
    EMP_REF_ID int,  
    EMP_TITLE varchar(100),  
    AFFECTED_FROM timestamp,  
    FOREIGN KEY (EMP_REF_ID) REFERENCES EmployeesTable (EMP_ID)  
);
```

INSERTATION

```
INSERT INTO Employees Table (
```

```
    EMP_ID,  
    FIRST_NAME,  
    LAST_NAME  
    SALARY,  
    JOINING_DATE,  
    DEPARTMENT,  
)
```

VALUES

```
(001, 'Manish', 'Agrawal', 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 Variables Details (

```
    EMP_REF_ID,
    VARIABLES_DATE,
    VARIABLES_AMOUNT
```

)

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 (

```
    EMP_REF_ID,
    EMP_TITLE,
    AFFECTED_FROM
```

)

VALUES

```
(1, 'Asst Manager', '2019-02-20 00:00:00'),
(2, 'Senior Analyst', '2019-01-11 00:00:00'),
(8, 'Senior Analyst', '2019-04-00 00:00:00')
(6, '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');
```

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

a) INNER JOIN: INNER JOINS are used to retrieve only common matching records.

Example:

```
SELECT e.EMP_ID, d.EMP_TITLE
FROM Employees Table e
INNER JOIN DesignationTable d ON e.EMP_ID = d.EMP_REF_ID;
```

b) LEFT JOIN: The LEFT JOIN command returns all rows from the left table, and the matching rows from the right table.

Example:

```
SELECT e.EMP_ID, d.EMP_TITLE
FROM EmployeesTable e
LEFT JOIN DesignationTable d ON e.EMP_ID = d.EMP_REF_ID
ORDER BY e EMP_ID;
```

c) RIGHT JOIN: The RIGHT JOIN command returns all rows from the right table, and the matching records from the left table.

Example:

```
SELECT e.FIRST_NAME, d.EMP_REF_ID
FROM Employees Table e
RIGHT JOIN DesignationTable d ON d.EMP_REF_ID = e.EMP_ID
ORDER BY d.EMP_REF_ID;
```

IV) CROSS JOIN: The CROSS JOIN keyword returns all records from both table.

Example:

```
SELECT * FROM EmployeesTable e
CROSS JOIN Designation_Table d;
```

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

```
SELECT FIRST_NAME, LAST_NAME, DEPARTMENT
FROM Employees Table
JOIN VariablesDetails ON EmployeesTable.EMP_ID = VariablesDetails.EMP_REF_ID
WHERE VariablesDetail.VARIABLES_AMOUNT = (
SELECT MAX(VARIABLES_AMOUNT) FROM Variables Details) OR VariablesDetails.VARIABLES_AMOUNT = (
SELECT MIN(VARIABLES_AMOUNT) FROM Variables Details);
```

c. What is cross join? Write a query to give an example of the same by performing it on the Employees table and Designation table.

The CROSS JOIN keyword returns all records from both table. Thus, using CROSS JOIN we will get all rows from Employees table and Designation table.

Example:

```
SELECT * FROM EmployeesTable e  
CROSS JOIN DesignationTable d;
```

d. What are the clauses used with Select statements and what are the preference orders of it?

Clauses help us to restrict and manage the data using valid constraints on the data in our database.

The SELECT statement has the following clauses:

1. SELECT
2. FROM
3. WHERE
4. GROUP BY
5. HAVING
6. ORDER BY
7. OFFSET
8. FETCH FIRST
9. UNION
10. INTERSECT
11. EXCEPT
12. WITH

3. What is the stored procedure?

A stored procedure is a prepared SQL code that we can save, so the code can be reused over and over again. So if you have an SQL query that you write over and over again, save it as a stored procedure, and then just call it to execute it.

a. Write a query to get the employee details who got their designations updated in the second half of the year 2019(July to December), sorted by the “variables_amount” (highest to lowest)

```
SELECT FROM EmployeesTable e  
LEFT JOIN DesignationTable d  
    ON e.EMP_ID = d.EMP_REF_ID  
LEFT JOIN VariablesDetails w  
    ON e.EMP_ID = w.EMP_REF_ID  
WHERE d.AFFECTED FROM BETWEEN '2019-07-01 00:00:00' AND '2019-12-31 00:00:00'  
ORDER BY w.VARIABLES_AMOUNT DESC;
```

b. Write a stored procedure to call the query that you have written for Q2.a

```
DELIMITER //  
CREATE PROCEDURE highest_least_variables()
```

```
BEGIN
    SELECT FIRST_NAME, LAST_NAME, DEPARTMENT
    FROM EmployeesTable
    JOIN VariablesDetails ON EmployeesTable.EMP_ID= VariablesDetails.EMP_REF_ID
    WHERE VariablesDetails.VARIABLES_AMOUNT = (
        SELECT MAX(VARIABLES_AMOUNT) FROM Variables Details)
    OR
        VariablesDetails.VARIABLES_AMOUNT = ( SELECT MIN(VARIABLES_AMOUNT) FROM
        VariablesDetails);
END//

DELIMITER;
```