SQL Assessment – Date 11/7/2022

Section 1:

1. Explain 3 table constraints with examples

UNIQUE is a table constraint that mandates all column values to be distinct and is useful for data such as social security numbers. Additionally, a UNIQUE column can accept exactly one null value.

NOT NULL is a table constraint that requires each row must have a value within that column. This is useful for data such as contact information or other types of identifiable information.

PRIMARY KEY is a table constraint that combines the features of UNIQUE and NOT NULL. Therefore, every row within the PRIMARY KEY column must have a distinct value and none of those values can be null. Moreover, it is by this constraint that SQL creates a cluster index and accordingly organizes the table data.

2. Explain different types of joins in SQL

RIGHT JOIN returns the values in one table that are present in a second table. Suppose there are two tables, Table A and Table B. Moreover, suppose that Table A and Table B share a common column name. Then, if it is necessary to append the data in Table A onto Table B. Then, a RIGHT JOIN parses the relevant information in Table B, based on the common column, and appends the relevant values from Table A.

LEFT JOIN is similar to RIGHT JOIN in that it requires the tables to share a common column. Then, for the instance of Table A and Table B, if it is necessary to append the data from Table B onto Table A, then a LEFT JOIN parses the relevant information in Table A, based on the common column, and appends the relevant values from Table B.

INNER JOIN can be considered in similarity to an intersection of sets. Therefore, an INNER JOIN considers the relevant data from both Table A and

Table B and combines the data based on the intersecting column of these two tables.

3. Explain Group by and Order by with example

“Group by” checks the queried table data and arranges them into subsets based on the non-cluster index specified in the “Group by” clause. Suppose the data in the table has different characteristics, such as a given location, then “Group by” arranges the data by that characteristic into subsets of the originally requested block of data. GROUP BY Location, for example.

“Order by” sorts the queried data in either ascending or descending order. Suppose the requested data contains a range of dates, then “Order by” allows either the latest (DESC) or earliest date (ASC) to be displayed at the top with sequential dates following.

Section 2:

Understand the below tables and write a query for Q1 to Q20

Table – EmployeeDetails

EmpId	FullName	ManagerId	DateOfJoining	City
121	John Snow	321	01/31/2019	Toronto
321	Walter White	986	01/30/2020	California
421	Kuldeep Rana	876	27/11/2021	New Delhi

Table – EmployeeSalary

EmpId	Project	Salary	Variable
121	P1	8000	500
321	P2	10000	1000
421	P1	12000	0

Q1 Write an SQL query to fetch the EmpId and FullName of all the employees working under the Manager with id – '986'.

SELECT EmpId, FullName FROM EmployeeDetails WHERE ManagerId = 986;

Q2 Write an SQL query to fetch the different projects available from the EmployeeSalary table.

SELECT Project FROM EmployeeSalary;

Q3 Write an SQL query to find the maximum, minimum, and average salary of the employees.

SELECT max(Salary), min(Salary), avg(Salary) FROM EmployeeSalary;

Q4 Write an SQL query to find the employee id whose salary lies in the range of 9000 and 15000

SELECT EmpId FROM EmployeeSalary WHERE Salary IN [9000, 15000];

Q5 Write an SQL query to fetch those employees who live in Toronto and work under the manager with ManagerId – 321

SELECT * FROM EmployeeDetails WHERE City = 'Toronto' AND ManagerId

= 321;

Q6 Write an SQL query to fetch all those employees who work on Projects other than P1

```
SELECT * FROM EmployeeSalary WHERE Project <> ‘P1’;
```

Q7 Write an SQL query to fetch all the EmpIds which are present in either of the tables – ‘EmployeeDetails’ and ‘EmployeeSalary’.

```
SELECT EmpId FROM EmployeeDetails WHERE EmpId IN

    (SELECT EmpId FROM EmployeeSalary);
```

Q8 Write an SQL query to fetch the EmpIds that are present in both the tables – ‘EmployeeDetails’ and ‘EmployeeSalary.

```
SELECT EmpId FROM EmployeeDetails

UNION

SELECT EmpId FROM EmployeeSalary;
```

Q9 Write an SQL query to fetch the employee's full names and replace the space with '-'.

SELECT REPLACE(FullName, ' ', '-') FROM EmployeeDetails

Q10 Write an SQL query to display both the EmpId and ManagerId together.

SELECT EmpId, ManagerId FROM EmployeeDetails;

Q11 Write an SQL query to find the count of the total occurrences of a particular character – 'n' in the FullName field

SELECT COUNT(FullName, 'n') FROM EmployeeDetails;

Q12 Write an SQL query to update the employee names by removing leading and trailing spaces.

UPDATE TABLE EmployeeDetails

SET FullName.strip()

Q13 Fetch all the employees who are not working on any project.

```
SELECT * FROM EmployeeSalary WHERE Project IS NULL;
```

Q14 Write an SQL query to fetch employee names having a salary greater than or equal to 5000 and less than or equal to 10000

```
SELECT FullName FROM EmployeeDetails WHERE EmpId IN

    (SELECT EmpId FROM EmployeeSalary WHERE Salary >= 5000 and
    Salary <= 10000)
```

Q15 Write an SQL query to fetch all the Employee details from the EmployeeDetails table who joined in the Year 2020

```
SELECT * FROM EmployeeDetails WHERE DateOfJoining LIKE
'%2020';
```

Q16 Write an SQL query to fetch all employee records from the EmployeeDetails table who have a salary record in the EmployeeSalary table.

```
SELECT * FRom EmployeeDetails WHERE EmpId IN
```

(SELECT EmpId FROM EmployeeSalary WHERE Salary IS NOT NULL);

Q17 Write a query to fetch employee names and salary records. Display the employee details even if the salary record is not present for the employee.

SELECT FullName, EmployeeSalary.Salary FROM EmployeeDetails WHERE EmpId IN

(SELECT * FROM EmployeeSalary);

Q18 Write an SQL query to fetch all the Employees who are also managers from the EmployeeDetails table.

SELECT * FROM EmployeeDetails WHERE EmpId IN

(SELECT ManagerId FROM EmployeeDetails);

Q19 Write an SQL query to find the nth highest salary from a table.

SELECT Salary FROM EmployeeSalary ORDER BY Desc;

Q20 Write SQL query to find the 3rd highest salary from a table without using the TOP/limit keyword

SELECT Salary FROM EmployeeSalary ORDER BY Desc;