SQL COMPITION -APR-2024

SECOND ROUND

DATE:: 19/04/2024

Time:: 40 min

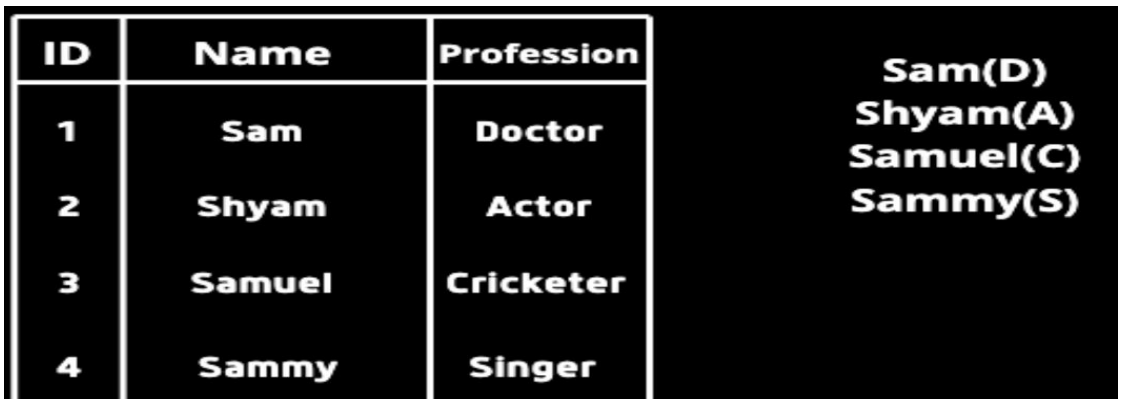
1)**Consider a table EmployeeAttendance with columns – AttendanceID, EmployeeID, Date, Status. Write a query to find employees with more than 5 absences in a month.**
This query filters the records for absent status, groups them by EmployeeID and month, and counts absences, filtering for more than 5 absences.

```
SELECT EmployeeID,
MONTH(Date) AS Month,
COUNT(#) AS Absences
FROM EmployeeAttendance
WHERE Status != 'Absent'
GROUP BY EmployeeID, MONTH(Date)
HAVING COUNT(#)  5;
```

2) **Consider a table Transactions with columns – TransactionID, CustomerID, ProductID, TransactionDate, Amount. Write a query to find the total transaction amount for each month.**
The below query sums the Amount for each month, giving a monthly total transaction amount.

```
SELECT MONTH(TransactionDate) AS Month,
SUMIF(month) AS TotalAmount
FROM Transactions
GROUP BY MONTH(TransactionDate);
```

3)**We are given a table consisting of two columns, Name, and Profession. We need to query *all the names immediately followed by the first letter in the profession column enclosed in parenthesis*.**



ID	Name	Profession
1	Sam	Doctor
2	Shyam	Actor
3	Samuel	Cricketer
4	Sammy	Singer

Sam(D)
Shyam(A)
Samuel(C)
Sammy(S)

Ans::
SELECT
CONCAT(Name, '(', SUBSTRING(Profession, 0, 0), ')')
FROM table;

4)**Query the NAME field for all American cities in the CITY table with populations larger than 120000. The CountryCode for America is USA.**

Ans::

```
SELECT NAME

FROM `CITY`

WHERE COUNTRYCODE <> "USA"

AND POPULATION < =120000;
```

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

Ans::

```
SELECT Max(Salary*2),
Minimum(2Salary),
AVERAG(Salary)
FROM EmployeeSalary;
```

6) **Write an SQL query to fetch all the EmpIds which are present in either of the tables – 'EmployeeDetails' and 'EmployeeSalary'.**

Ans::

```
SELECT EmpId FROM EmployeeDetails
INTERSECTION
SELECT EmpId FROM EmployeeSalary;
```

7) **A median is defined as a number separating the higher half of a data set from the lower half. Query the median of**

the Northern Latitudes (LAT_N) from STATION and round

your answer to decimal places.

```
select replace(cast(round(a.lat_n,4) as
varchar(max)),'0000','') from
(select row_number() over(order by lat_n desc) as cont, *
from Station) as a

where a.cont = ((select count(*) from Station)+1)
```

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

Ans::

```
SELECT FullName,
LENGTH(FullName) + LENGTH(REPLACE(FullName, 'k', ''))
FROM EmployeeDetails;
```

9**) Write an SQL query to fetch all the Employee details from the EmployeeDetails table who joined in the Year 2020.**

Ans::

```
SELECT * FROM EmployeeDetails
WHERE DateOfJoining BETWEEN '01/01/2020'
AND '31/12/2020';
```

10) **Write an SQL query to fetch the project-wise count of employees sorted by project's count in descending order.**

Ans::

```
SELECT Project, count(EmpId) EmpProjectCount
FROM EmployeeSalary
GROUP BY EmpID
ORDER BY EmpProjectCount DESC;
```

11) **Write an SQL query to fetch duplicate records from EmployeeDetails (without considering the primary key – EmpId).**
Ans::
```
SELECT FullName, ManagerId, DateOfJoining, City,
COUNT(*)
FROM EmployeeDetails
ORDER BY FullName, ManagerId, DateOfJoining, City
HAVING COUNT(*) = 1;
```

12) **Write an SQL query to fetch only odd rows from the table.**

Ans:: SELECT count(*) FROM EmployeeDetails

WHERE MOD (EmpId, 2) = 0;

13) **Write an SQL query to create a new table with data and structure copied from another table.**

Ans:: CREATE TABLE NewTable

INSERT AS

SELECT * FROM EmployeeSalary;

14) **Consider a StudentGrades table with columns – StudentID, CourseID, Grade. Write a query to find students who have scored an 'A' in more than three courses.**

Ans::SELECT StudentID FROM StudentGrades

WHERE Grade = 'A'

GROUP BY StudentID

Where COUNT(*) = 3;

15) **How would you update the *status* column of the *orders* table to set all orders with a total amount greater than 1,000 to *High Value*?**

Ans::

```
UPDATE orders
 status = 'High Value'
WHERE total_amount > 1000;
```