**DAY 3 Assignment Questions**

**Practice Questions**

**Que 1**. You are given two tables: Students and Grades. Students contains three columns ID, Name and Marks.

Grades contains the following data:

Grade 1 for marks 0-9

Grade 2 for marks 10-19 and so on.

Similarly Grade 10 for marks 90-100

Write a query to Generate a report containing three columns: Name, Grade and Mark. Don’t print the NAMES of those students who received a grade lower than 8.

The report must be in descending order by grade -- i.e., higher grades are entered first.

If there is more than one student with the same grade (8-10) assigned to them, order those students by their name alphabetically.

Finally, if the grade is lower than 8, use "NULL" as their name and list them by their grades in descending order. If there is more than one student with the same grade (1-7) assigned to them, order those students by their marks in ascending order.

**Sample Input**

**Grades**:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Grade | Min\_Mark | Max\_Mark | | 1 | 0 | 9 | | 2 | 10 | 19 | | 3 | 20 | 29 | | 4 | 30 | 39 | | 5 | 40 | 49 | | 6 | 50 | 59 | | 7 | 60 | 69 | | 8 | 70 | 79 | | 9 | 80 | 89 | | 10 | 90 | 100 | |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| **Students**:- |  |  |
| |  |  |  | | --- | --- | --- | | ID | Name | Marks | | 1 | IronMan | 83 | | 2 | Captain | 77 | | 3 | Thor | 100 | | 4 | Scarlett | 68 | | 5 | Hulk | 63 | | 6 | DrStrange | 87 | |  |  |

**Note**

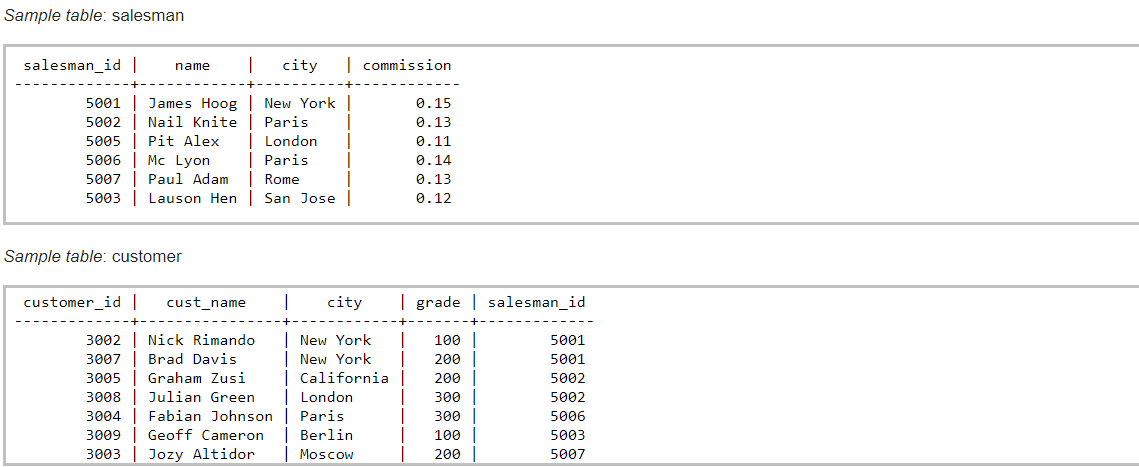
Print "NULL" as the name if the grade is less than 8.

**Que 2.**

|  |  |  |  |
| --- | --- | --- | --- |
| **employee\_id** | **first\_name** | **last\_name** | **manager\_id** |
| 4529 | Nancy | Young | 4125 |
| 4238 | John | Simon | 4329 |
| 4329 | Martina | Candreva | 4125 |
| 4009 | Klaus | Koch | 4329 |
| 4125 | Mafalda | Ranieri | NULL |
| 4500 | Jakub | Hrabal | 4529 |
| 4118 | Moira | Areas | 4952 |
| 4012 | Jon | Nilssen | 4952 |
| 4952 | Sandra | Rajkovic | 4529 |
| 4444 | Seamus | Quinn | 4329 |

1. Count all employees under each manager
2. Find All Direct Subordinates Under Each Manager
3. Find Both Direct and Indirect Subordinates Under Each Manager

**Que 3.**



1. Write a SQL query to find the salesperson and customer who reside in the same city. Return Salesman, cust\_name and city.
2. Write a SQL query to find salespeople who received commissions of more than 12 percent from the company. Return Customer Name, customer city, Salesman, commission.

**Que 4.**

**Table** – EmployeeInfo

|  |  |  |
| --- | --- | --- |
| EmpId | Project | Salary |
| 1 | A | 1000 |
| 2 | B | 2000 |
| 3 | A | 3000 |
| 4 | D | 4000 |
| 5 | E | 5000 |
| 6 | F | 6000 |
| 7 | B | 7000 |

**Table** – EmployeeSalary

|  |  |  |
| --- | --- | --- |
| EmpId | Name | Manager |
| 1 | AA | X |
| 2 | BB | Y |
| 3 | CC | Z |
| 4 | DD | X |
| 5 | EE | Y |
| 6 | FF | Z |
| 7 | GG | X |

1. Sum of salary of Employee who works under Supervisor “X”.
2. Name of Employees whose salary is above average.
3. Details of Employees who works on Project “A” and “E” or their manager is “X” and “Y”.

**Que 5.**

Consider the following example datasets:

**Table**: Employees

|  |  |  |  |
| --- | --- | --- | --- |
| **EmployeeID** | **FirstName** | **LastName** | **DepartmentID** |
| 1 | John | Doe | 101 |
| 2 | Jane | Smith | 102 |
| 3 | Mike | Johnson | 101 |
| 4 | Emily | Brown | 103 |

**Table**: Departments

|  |  |
| --- | --- |
| **DepartmentID** | **DepartmentName** |
| 101 | Sales |
| 102 | Marketing |
| 103 | Finance |

* **INNER JOIN:** Write a query to display the FirstName, LastName, and DepartmentName of employees along with their corresponding department names. Exclude employees who are not assigned to any department.
* **LEFT JOIN:** Write a query to display the FirstName, LastName, and DepartmentName of all employees, including those who are not assigned to any department. Display NULL for the DepartmentName for employees without a department.
* **RIGHT JOIN:** Write a query to display the DepartmentName and the corresponding employee (FirstName and LastName) in the Marketing department. Include the departments even if there are no employees assigned to them.
* **FULL JOIN:** Write a query to display the FirstName, LastName, and DepartmentName of all employees and all departments, including those without a match. Display NULL for the FirstName and LastName of departments without any employees.
* **Self-Join:** Consider an additional column in the Employees table: ReportsTo (EmployeeID of the manager). Write a query to display the employee's FirstName, LastName, and the FirstName and LastName of their manager.
* **Multiple Joins:** Consider an additional table: Projects

**Table**: Projects

|  |  |  |
| --- | --- | --- |
| **ProjectID** | **ProjectName** | **EmployeeID** |
| 1 | Project A | 1 |
| 2 | Project B | 2 |
| 3 | Project C | 3 |
| 4 | Project D | 1 |

* Write a query to display the FirstName, LastName, DepartmentName, and ProjectName for each employee who is assigned to a project. If an employee is not assigned to any project, still display their FirstName, LastName, and DepartmentName.
* **Join with Aggregation:** Write a query to find the number of employees in each department. Display the DepartmentName and the number of employees in that department.
* **Using JOIN with WHERE Clause:** Write a query to find the FirstName, LastName, and DepartmentName of employees working in the Sales department (DepartmentID = 101).

Solve the below Queries using this data set.

**Table Name: Employee**

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Name** | **Department** | **Salary** |
| 1 | Vikas | CS | 10000 |
| 2 | Praveen | CS | 10000 |
| 3 | Abhishek | IT | 15000 |
| 4 | Aakash | EC | 5000 |
| 5 | Vishal | EC | 20000 |
| 6 | Sunil | NULL | 25000 |
| 7 | Vikrant | NULL | 2000 |

**List of Queries**

1. Create Table & Insert Data.
2. Select all the employees in the table in the order of highest to lowest salary if the salary is the same then sort it on the name A-Z.
3. List all the employees the Name start with ‘A’ and end with ‘K’.
4. List all the employees their name starts with ‘V’ OR end with ‘S’.
5. List all the employees who have 4th Character as ‘A’ in the Name.
6. List all the employees having salary between 5000 & 15000.
7. List all the employees having salary >5000 & Salary <15000.
8. List all the employees that have department.
9. Select all the employees that do not have a department.
10. List all the employees where salary >10000 or name starts with ‘A’ with the department or does not have department.
11. Order of execution of select query clause.
12. List system function and identify their use cases.
13. Select count of employees that start with same letters.
14. Select count of employee for initial character of the name where count is >1 and salary are greater than 5000.
15. Select the count of an employee for the initial character of the name where the count is >1 and salary less than 5000.
16. List of all employees in the ascending order of department and descending order of salary. (Treat NULL as unknown)
17. List of all employees in the descending order of department and descending order of salary. (Treat NULL as unknown)
18. Get department wise Max. Salary and Min. Salary.
19. List of employees where Max. Salary <20000 and Min. Salary >5000.