SQL Queries and Database Objects

Lab:Writing SQL Queries using Employee-Department Tables

Refer to scripts present in Samples/Module2/PracticeQueries

Create Tables using script: **Emp_Dept.sql**

Simple Select queries

1. Calculate total salary of employee as Salary + commission and get employee name and Annual Salary.
2. Get all employees in department 10 and name starting with (a-c)
3. Sort employees by department no and names.
4. Get employees with job as salesman or clerk.
5. Get Top 3 Employees having Highest salaries

Aggregate Queries

6. Get total no of employees in each department.
7. Get Max ,Min, Avg, Sum of Salary for each department.
8. Get total no of employees in each department with salary more than 1500.
9. Get total no of employees in 10, 30 department.
10. Get Min,Max Salary in each department,Department should have more than 3 employees
11. Get Total no of employees hired in different years.

Joins

12. Get Employee name,salary and DepartmentName
13. Get all Employee and department details.Include all employees if if department is not assigned to them
14. Get employee name,manager name and their salaries
15. Get Department for which there is no employee
16. Get Manager name and total number of employees working under that manager.

Subqueries

17. Get details of employees with same salary as **'SCOTT'**
18. Get employees having same job as SCOTT or SMITH
19. Display the names of employees who earn salary more than that of Allen or Scott.
20. **Get details of employees under manager 'JONES'**

Co-related Sub Queries

21. Get all employees having salary greater than average salary of their own department.
22. Select Department details having
    a. At least one employee
    b. No employee

Table Values Subquery

23. Get empname,salary and average salary of department to which employee belongs to.

Advanced Queries:

Window Aggregate Functions

1. **Get Employee Details along with average Salary of their own Department.**

```
SELECT FKDeptId,Salary,EmpName, Avg(Salary) OVER (PARTITION BY FKDeptid) FROM Employee
```

2. **Get Employee Details with average Salary of their own Department and difference between Salary and Avg Salary.**

```
SELECT FKDeptId, Salary, EmpName, Avg(Salary) OVER (PARTITION BY FKDeptid),Salary -Avg(Salary
```

Window ranking Functions: ROW_NUMBER(),RANK(),DENSE RANK(),NTILE()

Give ranks to all Employees based on Salaries

```
Select Empname,Salary,Job,ROW_NUMBER() OVER (ORDER BY SALARY) AS RowNum From Employee
```

Give ranks to Employees for Each Job based on Salaries

```
Select Empname,Salary,Job,RANK() OVER (partition by job ORDER BY SALARY) AS RowNum
From Employee
```

Pivot: Pivoting in SQL Server rotates the display of data from row based orientation to column based orientation.

Original Query:

```
Select FKDeptId,job,Sum(Salary) as TotalSalary
From Employee
Group by Job,Fkdeptid
```

Using Pivot

```
WITH PIVOTTABLE AS
(
SELECT Job, FKDeptId, Salary
FROM Employee
)
Select Job, [10], [20], [30]
From PIVOTTABLE
PIVOT(SUM(Salary) FOR FKDeptId IN ([10],[20],[30])) AS T1
```

Database Objects:

View : Create view to show Employee Summary departmentwise

```
Create View vw_Empdeptsummary
as
	SELECT D.DeptName,MIN(Salary) as MinSal,MAX(Salary) as MaxSal,SUM(Salary) as
TotalSal,AVG(Salary) as AvgSal,count(*) as TotalEmp
	FROM Employee E
	INNER JOIN Department D
	ON E.FKDeptId=D.PKDeptId
	GROUP BY D.DeptName
```

Stored Procedure:Write a procedure to update employee salary based on department and return no of rows affected

```
CREATE PROCEDUE SpUpdateEmployee(@id int,@increment)
AS
       Update Employee
       Set Salary=Salary+@increment
      Where fkdeptId =@id
       return @@rowcount
```

To Execute:

Declare @no int
EXEC @no = SpUpdateEmployee 1111,1000
Print @no

Function:Create Function to get Total Salary of Department

```
Create Function fnGetTotalSalaryOfDeparment(@DeptId int) returns Money
AS
       Begin
                        Return(Select
                                    sum(Salary+isnull(Commission,0))
                        From employee
                Where FkDeptId = @DeptId)
       End
```