- Views
- CTE
- Programming
- Ranking

1. Employees

```
CREATE TABLE Employees
(
   EmpId INT PRIMARY KEY,
   EmpName VARCHAR(100),
   DeptId INT,
   ManagerId INT NULL,
   JoinDate DATE,
   Salary DECIMAL(10,2)
);

INSERT INTO Employees VALUES
(1, 'Amit', 10, NULL, '2020-01-10', 65000),
(2, 'Neha', 10, 1,    '2022-02-15', 50000),
(3, 'Ravi', 20, 1,    '2023-03-12', 45000),
(4, 'Sana', 20, 3,    '2024-01-20', 42000),
(5, 'Karan', 30, 1,   '2021-07-18', 55000);
```

---

2. Departments

```
CREATE TABLE Departments
(
   DeptId INT PRIMARY KEY,
   DeptName VARCHAR(100)
);

INSERT INTO Departments VALUES
(10, 'IT'),
(20, 'HR'),
(30, 'Finance');
```

---

3. Sales

```
CREATE TABLE Sales
(
   SaleId INT PRIMARY KEY,
   EmpId INT,
   Region VARCHAR(50),
   SaleAmount DECIMAL(10,2),
   SaleDate DATE
);

INSERT INTO Sales VALUES
(1, 1, 'North', 100000, '2024-01-01'),
(2, 2, 'North',  90000, '2024-01-10'),
(3, 3, 'South', 120000, '2024-02-05'),
(4, 4, 'South', 120000, '2024-02-20'),
(5, 5, 'North', 110000, '2024-03-15');
```

---

4. Transactions

```
CREATE TABLE Transactions
(
   TransId INT PRIMARY KEY,
   AccountId INT,
   Amount DECIMAL(10,2),
   TransDate DATE
);

INSERT INTO Transactions VALUES
(1, 101, 1000, '2024-01-01'),
(2, 101, 2000, '2024-02-01'),
(3, 101, -500, '2024-03-01'),
(4, 102, 1500, '2024-01-15'),
(5, 102, -200, '2024-03-10');
```

Task-1

Write a query using **CASE** to categorize salary levels on Employees table:

- <20000 → Low
- 20000–50000 → Medium
- 50000 → High

Task -2

Declare a variable @Age.
Write logic using IF / ELSE:

- If Age < 18 → print “Minor”
- Else If Age between 18–60 → “Adult”
- Else → “Senior”

Task-3 Encrypted & Schema-Bound View

Create an **encrypted** and **schemabound** view that:

- Joins Employees, Departments, and Salaries tables
- Returns only employees who joined in the last 3 years
- Includes computed column: AnnualSalary = Salary * 12
- Prevents updates to base tables that break schema binding

Tasks

1. Create the view with WITH SCHEMABINDING, ENCRYPTION.
2. Try altering an underlying table column → observe the error.

Task-4— Complex Multi-Table View

Create a view that:

- Joins Employees + Sales
- Shows total sales per employee
- Shows rank based on total sales across company

Task-5— Simulate Error Capture

Write a block that:

- Attempts dividing by zero
- Catches the error
- Prints error details

Task-6— Nested TRY…CATCH With Custom Error

Validate salary:

- If salary < 1000, throw custom error using THROW.
- Declare variable to simulate salary

Task-7— Rank Employees by Region Sales

Task

- Compare Rank / Dense_Rank / Row_Number
- Identify top 2 per region

Task-8 -Using Sales table:

- First CTE: Filter only **last 1 year** sales
- Second CTE: Compute total sales per region
- Third CTE: Rank regions based on total sales
- Output **top 3 regions**

Task-8 Find Employees With Duplicate SalesAmount in Any Department

Task – 9

Perform Pagination and list all details from employees who’s page between 6 and 10