MODULE 4.4-Practical Project Assignment ABHISHEK ALLABOINA

https://learn.microsoft.com/en-us/sql/t-sql/language-reference?view=sql-server-ver17

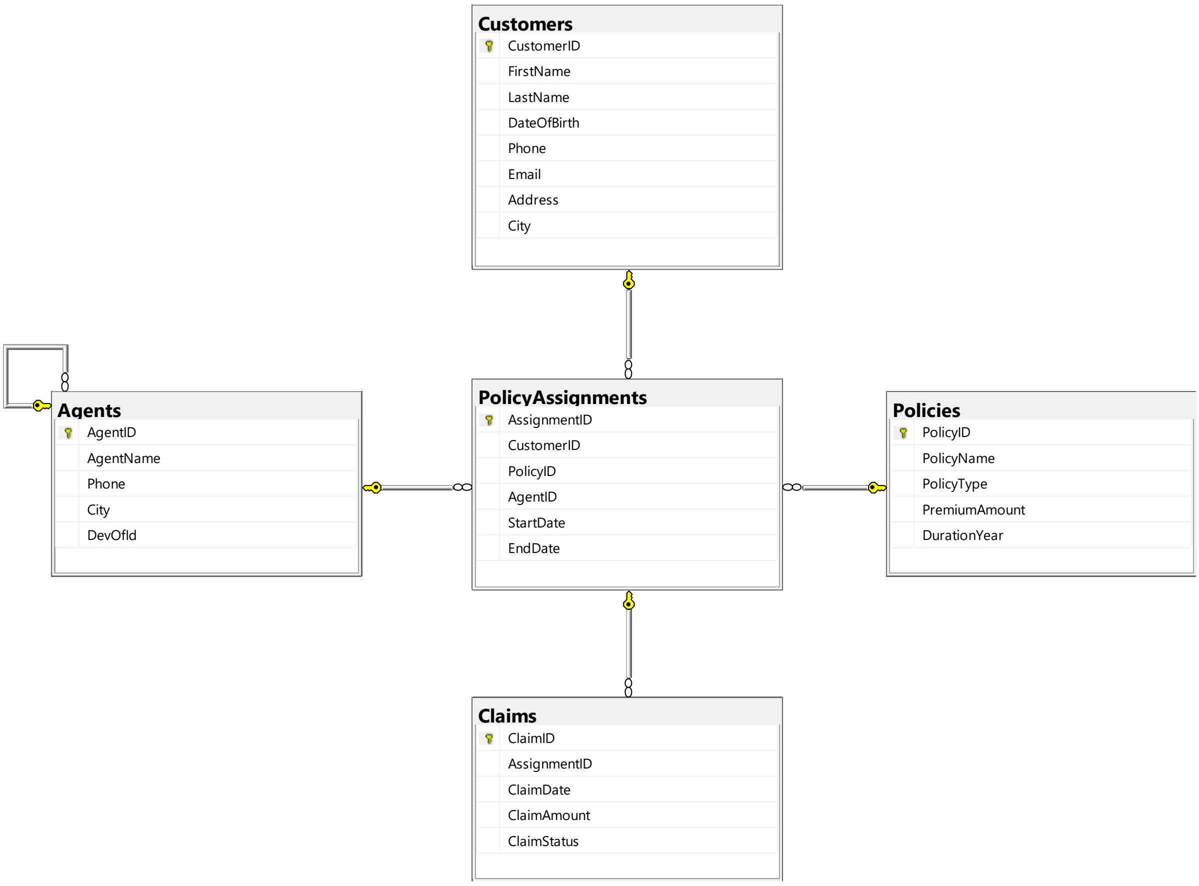
--1) Create Database command

```
CREATE DATABASE InsuranceDB;

USE InsuranceDB;
```

**--2) Create table commands for all the tables with constraints, relationships etc.**

```
CREATE TABLE Customers (
   CustomerID INT IDENTITY(1,1) CONSTRAINT PK_Customers PRIMARY KEY,
   FirstName VARCHAR(20) NOT NULL,
   LastName VARCHAR(20),
   DateOfBirth DATE,
   Phone VARCHAR(20),
   Email VARCHAR(20) UNIQUE
);

CREATE TABLE Policies (
   PolicyID INT IDENTITY(1,1) CONSTRAINT PK_Policies PRIMARY KEY,
   PolicyName VARCHAR(20),
   PolicyType VARCHAR(20),
   PremiumAmount DECIMAL(10,2),
   DurationYear INT
);

CREATE TABLE Agents (
   AgentID INT IDENTITY(1,1) CONSTRAINT PK_Agents PRIMARY KEY,
   AgentName VARCHAR(20),
   Phone VARCHAR(15),
   City VARCHAR(20)
);

CREATE TABLE PolicyAssignments (
   AssignmentID INT IDENTITY(1,1) CONSTRAINT PK_PolicyAssignments PRIMARY KEY,
   CustomerID INT,
   PolicyID INT,
   AgentID INT,
   StartDate DATE,
   EndDate DATE,

   CONSTRAINT FK_PolicyAssignments_Customers
       FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),

   CONSTRAINT FK_PolicyAssignments_Policies
       FOREIGN KEY (PolicyID) REFERENCES Policies(PolicyID),

   CONSTRAINT FK_PolicyAssignments_Agents
       FOREIGN KEY (AgentID) REFERENCES Agents(AgentID)
);


CREATE TABLE Claims (
   ClaimID INT IDENTITY(1,1) CONSTRAINT PK_Claims PRIMARY KEY,
   AssignmentID INT,
   ClaimDate DATE,
   ClaimAmount DECIMAL(10,2),
   ClaimStatus VARCHAR(20),

   CONSTRAINT FK_Claims_PolicyAssignments
       FOREIGN KEY (AssignmentID) REFERENCES PolicyAssignments(AssignmentID)
);
```

**SCHEMA DIAGRAM FOR INSURANCE DATABASE:**



**Insert commands for all tables.**

```
INSERT INTO Customers (FirstName, LastName, DateOfBirth, Phone, Email)
VALUES
('Abhi', 'Shek', '2004-07-03', '7075268421', 'abhishek@gmail.com'),
('Santhu', 'Chepuri', '2004-05-24', '9080706050', 'chepuri@gmail.com');
```

```
INSERT INTO Policies (PolicyName,PolicyType,PremiumAmount,DurationYear)
VALUES
('Life Term', 'Life', 15000.00, 20),
('Health Plus', 'Health', 12000.00, 10);


INSERT INTO Agents (AgentName, Phone, City)
VALUES
('Spoorthik', '7867867867', 'Karachi'),
('Idries', '7867867866', 'Rawalpindi');


INSERT INTO PolicyAssignments
(CustomerID, PolicyID, AgentID, StartDate, EndDate)
VALUES
(1, 1, 1, '2025-01-01', '2045-01-01'),
(2, 2, 2, '2025-06-01', '2035-06-01');


INSERT INTO Claims
(AssignmentID, ClaimDate, ClaimAmount, ClaimStatus)
VALUES
(1, '2026-02-15', 50000.00, 'Approved'),
(2, '2026-03-10', 30000.00, 'Pending');

INSERT INTO Customers (FirstName, LastName, DateOfBirth, Phone, Email)
VALUES
```

```
('Varun', 'Kumar', '1999-08-12', '9001112233', 'varun@gmail.com'),
('Bhanu', 'Prakash', '1998-03-25', '9002223344', 'bhanu@gmail.com'),
('Ramu', 'Reddy', '2000-12-05', '9003334455', 'ramu@gmail.com');


INSERT INTO Policies (PolicyName, PolicyType, PremiumAmount, DurationYear)
VALUES
('Life Term', 'Life', 18000.50, 25),
('Health Shield', 'Health', 9500.75, 12),
('Car Protect', 'Vehicle', 7200.00, 5);


INSERT INTO Agents(AgentName,Phone,City)
VALUES
('Anil', '9876543210', 'Delhi'),
('Suresh', '9223344556', 'Hyderabad');


INSERT INTO PolicyAssignments
(CustomerID, PolicyID, AgentID, StartDate, EndDate)
VALUES
(3, 3, 1, '2023-01-10', '2048-01-10'),
(4, 4, 2, '2024-06-15', '2036-06-15'),
(5, 5, 1, '2025-03-01', '2030-03-01');


INSERT INTO Claims
(AssignmentID, ClaimDate, ClaimAmount, ClaimStatus)
```

```
VALUES

(3, '2024-11-20', 45000.00, 'Approved'),

(4, '2025-07-05', 15000.00, 'Rejected'),

(5, '2025-09-18', 8000.00, 'Pending');



INSERT INTO Customers (FirstName, LastName, DateOfBirth, Phone, Email)

VALUES

('sangu', 'Rao', '2002-02-10', '9876543210', 'sangu@gmail.com'),

('Manu', 'pal', '1998-11-15', '9876543211', 'manu@gmail.com');




INSERT INTO Policies (PolicyName, PolicyType, PremiumAmount, DurationYear)

VALUES

('Motor Secure', 'Motor', 11000.00, 1),

('Family Health', 'Health', 20000.00, 1);


INSERT INTO Agents (AgentName, Phone, City)

VALUES

('Rakesh', '9112233445', 'Bangalore'),

('Ajay', '9887766554', 'Chandigarh');



INSERT INTO PolicyAssignments (CustomerID, PolicyID, AgentID, StartDate, EndDate)

VALUES

(1, 6, 3, '2024-01-01', '2025-01-01'),

(2, 7, 4, '2025-02-01', '2026-02-01'),
(6, 4, 3, '2023-05-10', '2035-05-10');
```

```
INSERT INTO Claims (AssignmentID, ClaimDate, ClaimAmount, ClaimStatus)

VALUES

(6, '2025-03-15', 60000.00, 'Approved'),

(7, '2025-05-20', 25000.00, 'Rejected');
```

**SELECT ,UPDATE ,DELETE COMMANDS:**

**1**: **Display policies having premium amount more than 10000 and DurationYears is 1**

```
Select * from Policies Where PremiumAmount>10000 AND DurationYear=1;
```

**2**: **List policies of type Life, Health, Motor use IN operator**

```
SELECT * FROM Policies

WHERE PolicyType IN ('Life','Health','Motor');
```

**3**:**Display list of customers born after January 1 st , 2001 and before December 31 st , 2020 using between operator**

```
SELECT * FROM Customers WHERE DateOfBirth BETWEEN'2001-01-01' AND '2020-12-31';
```

**4**: **Display records of Agents who stay in a city whose second letter is ‘a’**

```
SELECT * FROM Agents WHERE City LIKE '_a%';
```

**5**: **Display latest claim record**

```
select top 1* from Claims

order by ClaimDate;
```

**6**: **Increase premium amount to 10% for all health insurance policies**

```
SET PremiumAmount=PremiumAmount*1.10

WHERE PolicyType='Health';
```

**7: Delete the record of PolicyAssignments whose EndDate is before today&#39;s date.**

```
DELETE FROM PolicyAssignments

WHERE EndDate < GETDATE();
```

error:

```
Msg 547, Level 16, State 0, Line 236

The DELETE statement conflicted with the REFERENCE constraint
"FK_Claims_PolicyAssignments". The conflict occurred in database "InsuranceDB", table
"dbo.Claims", column 'AssignmentID'.

The statement has been terminated.
```

```
DELETE FROM Claims

WHERE AssignmentID IN (

  SELECT AssignmentID

  FROM PolicyAssignments

  WHERE EndDate < GETDATE()

);

DELETE FROM PolicyAssignments

WHERE EndDate < GETDATE();
```

**AGGREGATE FUNCTIONS:**

**8. Total number of customers**

```
SELECT COUNT(*) AS TotalCustomers

FROM Customers;
```

**9. Average premium amount for each policy type**

```
SELECT PolicyType, AVG(PremiumAmount) AS AvgPremium

FROM Policies

GROUP BY PolicyType;
```

**10. Total premium value of all policies**

```
SELECT SUM(PremiumAmount) AS TotalPremiumAmount

FROM Policies;
```

**11.Total claim amount by claim status**

```
SELECT ClaimStatus, SUM(ClaimAmount) AS TotalClaimAmount

FROM Claims

GROUP BY ClaimStatus;
```

**12. Maximum and minimum premium amount**

```
SELECT

   MAX(PremiumAmount) AS MaxPremium,

   MIN(PremiumAmount) AS MinPremium

FROM Policies;
```

## Date & Time Functions

**13. Extract year from customer date of birth**

```
SELECT FirstName, YEAR(DateOfBirth) AS BirthYear

FROM Customers;
```

**14. Extract month from policy start date**

```
SELECT AssignmentID, MONTH(StartDate) AS StartMonth

FROM PolicyAssignments;
```

**15. Calculate customer age**

```
SELECT FirstName,

     DATEDIFF(YEAR, DateOfBirth, GETDATE()) AS Age

FROM Customers;
```

**16. Policy duration in years using dates**

```
SELECT AssignmentID,

     DATEDIFF(YEAR, StartDate, EndDate) AS PolicyDurationYears

FROM PolicyAssignments;
```

**17. Claims raised in the last 1 year**

```
SELECT *

FROM Claims

WHERE ClaimDate >= DATEADD(YEAR, -1, GETDATE());
```

**18. Get month name of claim date**

```
SELECT ClaimID,

    DATENAME(MONTH, ClaimDate) AS ClaimMonth

FROM Claims;
```

## QUERIES USING JOINS, HAVING ,GROUPBY

**19.List all Policies for a CustomerId 5**

```
SELECT p.* FROM Policies p

JOIN

PolicyAssignments pa

ON p.PolicyID = pa.PolicyID WHERE pa.CustomerID = 5;
```

**20: Display FirstName, PolicyName, AgentName, StartDate and EndDate from their respective tables**

```
select FirstName,PolicyName,AgentName,StartDate,EndDate

FROM PolicyAssignments pa

join Customers c on pa.CustomerID=c.CustomerID

join Policies p on pa.PolicyID=p.PolicyID

join Agents a on pa.AgentID=a.AgentID;
```

**21: Display claims report with FirstName, PolicyName, ClaimAmount, ClaimStatus, and ClaimDate from their respective tables**

```
SELECT c.FirstName, p.PolicyName,
    cl.ClaimAmount, cl.ClaimStatus, cl.ClaimDate
FROM Claims cl
JOIN PolicyAssignments pa ON cl.AssignmentID = pa.AssignmentID
JOIN Customers c ON pa.CustomerID = c.CustomerID
JOIN Policies p ON pa.PolicyID = p.PolicyID;
```

**22: Display records of Customers with or without Policies**

```
SELECT c.CustomerID,c.FirstName,p.PolicyName,p.PolicyID
FROM Customers c
LEFT JOIN PolicyAssignments pa ON c.CustomerID = pa.CustomerID
LEFT JOIN Policies p ON pa.PolicyID = p.PolicyID;
```

**23: Show names and total claim amount of Customers With Claim Amount greater than 50000 (Use HAVING Clause)**

```
select concat(c.FirstName,c.LastName) as FullName,
sum(cl.ClaimAmount) as TotalClaim
from Customers c
  join PolicyAssignments pa on pa.CustomerID=c.CustomerID
join Claims cl on cl.AssignmentID=pa.AssignmentID
group by c.FirstName,c.LastName
having sum(cl.ClaimAmount)>50000;
```

**24: Display list with Agent Wise Policy Count**

```
SELECT a.AgentName,COUNT(pa.AgentID) AS PolicyCount
FROM Agents a JOIN PolicyAssignments pa ON
a.AgentID=pa.AgentID group by a.AgentID,a.AgentName;
```

## SUBQUERIES:

**25. Customers who have NOT taken any policy**

```
SELECT *

FROM Customers

WHERE CustomerID NOT IN (

   SELECT CustomerID

   FROM PolicyAssignments

);
```

**26. Policies with premium higher than the average premium**

```
SELECT *

FROM Policies

WHERE PremiumAmount > (

   SELECT AVG(PremiumAmount)

   FROM Policies

);
```

**27. Policy with the highest premium amount**

```
SELECT *

FROM Policies

WHERE PremiumAmount = (

   SELECT MAX(PremiumAmount)

   FROM Policies

);
```

**28. Agents who have not handled any claims**

```
SELECT *

FROM Agents

WHERE AgentID NOT IN (

   SELECT AgentID

   FROM PolicyAssignments PA
   JOIN Claims C ON PA.AssignmentID = C.AssignmentID
);
```

**29. Customers who have more than one policy**

```
SELECT *
FROM Customers
WHERE CustomerID IN (
   SELECT CustomerID
   FROM PolicyAssignments
   GROUP BY CustomerID
   HAVING COUNT(*) > 1
);
```

## CASE, ROLLUP, CUBE & GROUPING

**30. Categorize policies based on premium amount (CASE)**

```
SELECT PolicyName, PremiumAmount,
      CASE
         WHEN PremiumAmount < 10000 THEN 'Low Premium'
         WHEN PremiumAmount BETWEEN 10000 AND 15000 THEN 'Medium Premium'
         ELSE 'High Premium'
      END AS PremiumCategory
FROM Policies;
```

**31. Count policies by premium category**

```
SELECT
   CASE
      WHEN PremiumAmount < 10000 THEN 'Low'
      WHEN PremiumAmount BETWEEN 10000 AND 15000 THEN 'Medium'
```

```
        ELSE 'High'
    END AS PremiumCategory,
    COUNT(*) AS PolicyCount
FROM Policies
GROUP BY
    CASE
        WHEN PremiumAmount < 10000 THEN 'Low'
        WHEN PremiumAmount BETWEEN 10000 AND 15000 THEN 'Medium'
        ELSE 'High'
    END;
```

**32.Identify subtotal rows using GROUPING function**

```
SELECT
    ClaimStatus,
    SUM(ClaimAmount) AS TotalClaimAmount,
    GROUPING(ClaimStatus) AS IsSubtotal
FROM Claims
GROUP BY ROLLUP (ClaimStatus);
```

**33. Claim count by Year and Status (Cube)**

```
SELECT
    YEAR(ClaimDate) AS ClaimYear,
    ClaimStatus,
    COUNT(*) AS ClaimCount
FROM Claims
GROUP BY CUBE (YEAR(ClaimDate), ClaimStatus);
```

**SET OPERATIONS:**

34: List all unique phone numbers from Customers and Agents

```
SELECT Phone FROM Customers
UNION
SELECT Phone FROM Agents;
```

**35: Customers who have policies AND have raised claims**

```
SELECT CustomerID
FROM PolicyAssignments
INTERSECT
SELECT PA.CustomerID
FROM PolicyAssignments PA
JOIN Claims C ON PA.AssignmentID = C.AssignmentID;
```

**36:Customers who have policies but never raised a claim**

```
SELECT CustomerID
FROM PolicyAssignments

EXCEPT

SELECT PA.CustomerID
FROM PolicyAssignments PA
JOIN Claims C ON PA.AssignmentID = C.AssignmentID;
```