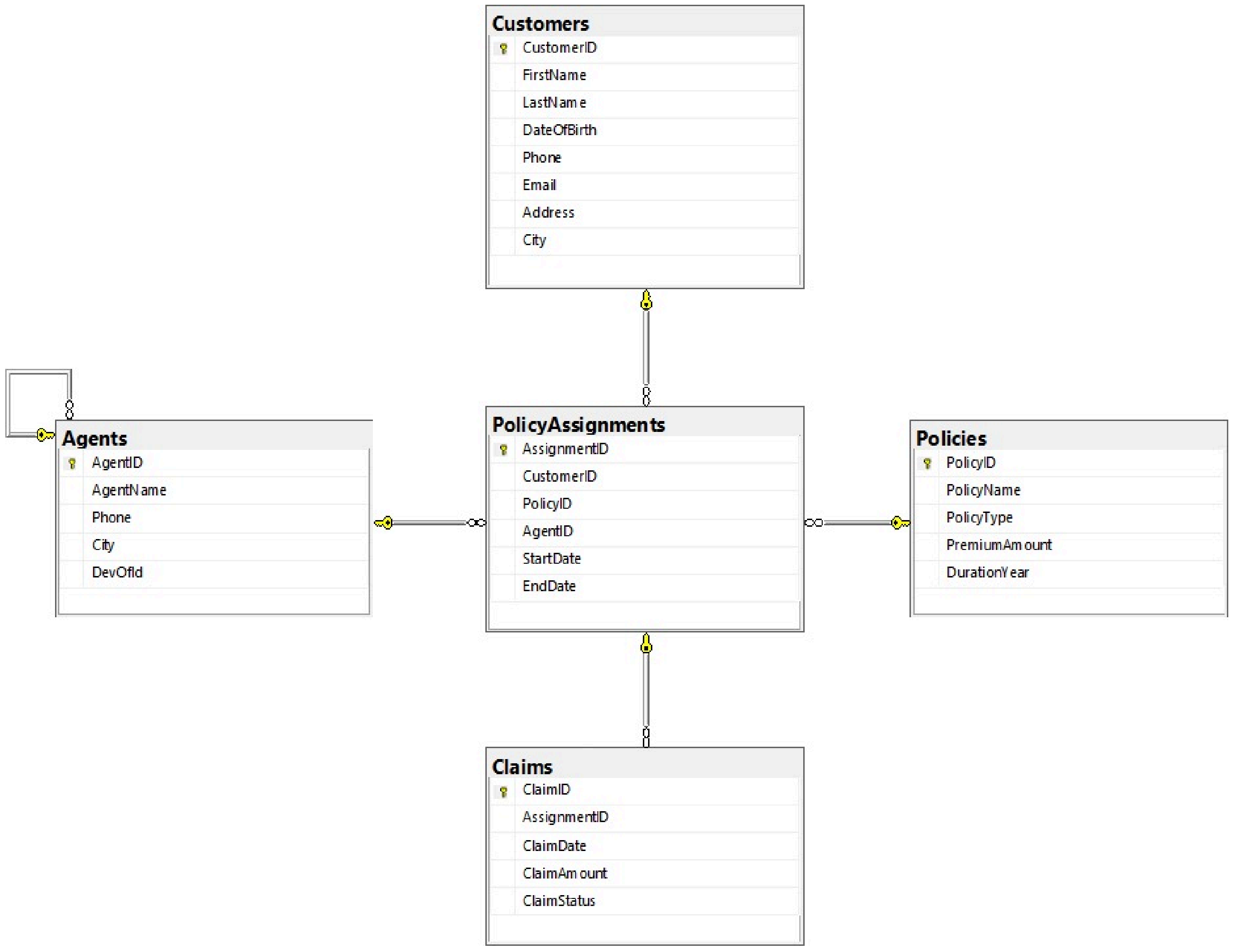
Module 4.4 - Practical Project Assignment

- By Ajay Lingampalli

1. Creating Insurance DB Database

```
CREATE DATABASE INSDB;
USE INSDB;
```

2. Database Diagram



3. Creating Tables in INSDB

Customer Table:

```
CREATE TABLE Customers (
   CustomerID INT IDENTITY PRIMARY KEY,
   FirstName VARCHAR(50) NOT NULL,
   LastName VARCHAR(50),
   DateOfBirth DATE,
   Phone VARCHAR(15),
   Email VARCHAR(100) UNIQUE
);
```

Policies Table:

```
CREATE TABLE Policies (
   PolicyID INT IDENTITY PRIMARY KEY,
   PolicyName VARCHAR(50),
   PolicyType VARCHAR(50),
   PremiumAmount INT,
   DurationYears INT
);
```

PolicyAssignments Table:

```
CREATE TABLE PolicyAssignments (
   AssignmentID  INT IDENTITY PRIMARY KEY,
   CustomerID INT,
   PolicyID INT,
   AgentID VARCHAR(50),
   StartDate DATE,
   EndDate DATE,
   FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
   FOREIGN KEY (PolicyID) REFERENCES Policies(PolicyID)
);
```

Claims Table:

```
CREATE TABLE Claims (
   ClaimID INT IDENTITY PRIMARY KEY,
   AssignmentID INT,
   ClaimDate DATE,
   ClaimAmount INT,
   ClaimStatus VARCHAR(50),
   FOREIGN KEY (AssignmentID) REFERENCES PolicyAssignments(AgentID)
);
```

Agents Table:

```
CREATE TABLE Agents (
AgentID INT IDENTITY PRIMARY KEY,
AgentName VARCHAR(50),
Phone VARCHAR(15),
City Varchar(50),
FOREIGN KEY (AgentID) REFERENCES PolicyAssignments(AssignmentID)
);
```

4. Insertions Into Tables

```
INSERT INTO Customers
VALUES
('Ajay', 'Rao', '2005-01-29', '6305284031', 'ajayrao1294@gmail.com');
INSERT INTO Customers
VALUES
('Ravi', 'Kumar', '1999-05-15', '9876543210', 'ravikumar@gmail.com');

INSERT INTO Policies
VALUES
('Health Plus', 'Health', 12000, 5),
('Life Secure', 'Life', 15000, 10);

INSERT INTO PolicyAssignments
VALUES
(1, 1, 'AGT101', '2024-01-01', '2029-01-01'),
(2, 2, 'AGT102', '2023-06-01', '2033-06-01');

INSERT INTO Claims
VALUES
(1, '2024-06-15', 50000, 'Approved'),
(2, '2024-08-10', 100000, 'Pending');

INSERT INTO Agents VALUES ('Ramesh','9876543210','Chennai'),
('Suresh','9123456780','Hyderabad');
```

INSERT INTO Agents VALUES ('Raju','9989723210','Banglore');

5.Select Commands

1.Select * from Customers;

2.Select CustomerID, PolicyID, StartDate, EndDate from PolicyAssignments;

3..Select * From Policies WHERE PolicyType= 'Health' OR PolicyType='Life' OR PolicyType='Motor';

4.Select * From Policies WHERE PolicyType IN('Health','Life','Motor');

5.Select * from Customers where DateOfBirth > '2001-01-01'and DateOfBirth < '2020-12-31';

6.Select * from Customers where DateOfBirth BETWEEN '2001-01-01'and '2020-12-31';

7.Select TOP 1 * From Claims ORDER BY ClaimDate DESC;

6.Update Command

1.UPDATE Policies Set PremiumAmount=PremiumAmount*1.10;

2.UPDATE Customers SET Phone='9876543210' WHERE CustomerID=1;

3.UPDATE Claims SET ClaimStatus='Approved' WHERE ClaimAmount>10000;

4.UPDATE Agents SET City='Hyderabad' WHERE AgentID=1;

7.Delete Command

1.DELETE FROM PolicyAssignments WHERE EndDate < GETDATE();

2.DELETE FROM Customers WHERE CustomerID=10;

3.DELETE FROM PolicyAssignments WHERE EndDate<'2020-01-01';

4.DELETE FROM Policies WHERE PremiumAmount<1000;

8.ALTER Commands

```
1.ALTER TABLE Customers ADD Address VARCHAR(100), City VARCHAR(50);
```

```
2.ALTER TABLE Agents ADD DevOfId INT;
```

```
3.ALTER TABLE Agents ADD CONSTRAINT FK_Agents_DevOf FOREIGN KEY (DevOfId)
REFERENCES Agents(AgentID);
```

9.Queries using Joins, Group By, Having etc.

```
 1.Select P.* FROM Policies P JOIN PolicyAssignments PA ON P.PolicyID = PA.PolicyID
WHERE PA.CustomerID = 5;
```

```
 2. Select C.CustomerID, C.FirstName, C.LastName, P.PolicyName, P.PolicyType
FROM Customers C JOIN PolicyAssignments PA ON C.CustomerID = PA.CustomerID
JOIN Policies P ON PA.PolicyID = P.PolicyID;
```

```
3.Select C.FirstName, C.LastName, 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;
```

```
4.Select C.FirstName, P.PolicyName, A.AgentName, PA.StartDate, PA.EndDate
FROM PolicyAssignments PA JOIN Customers C ON PA.CustomerID = C.CustomerID
JOIN Policies P ON PA.PolicyID = P.PolicyID JOIN Agents A ON CAST(A.AgentID AS
VARCHAR) = PA.AgentID;
```

```
5. 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;
```

```
6.Select C.FirstName, SUM(CL.ClaimAmount) AS TotalClaimAmount FROM Customers C
JOIN PolicyAssignments PA ON C.CustomerID = PA.CustomerID
JOIN Claims CL ON PA.AssignmentID = CL.AssignmentID
GROUP BY C.FirstName;
```

```
7.Select C.FirstName,SUM(CL.ClaimAmount) AS TotalClaimAmount FROM Customers C
JOIN PolicyAssignments PA ON C.CustomerID = PA.CustomerID
JOIN Claims CL ON PA.AssignmentID = CL.AssignmentID
GROUP BY C.FirstName HAVING SUM(CL.ClaimAmount) > 50000;
```

10.SubQueries

1.SELECT * FROM Customers WHERE CustomerID IN (SELECT CustomerID FROM PolicyAssignments);

2.SELECT * FROM Policies WHERE PolicyID NOT IN (SELECT PolicyID FROM PolicyAssignments);

3.SELECT * FROM Customers WHERE CustomerID NOT IN (SELECT CustomerID FROM PolicyAssignments WHERE AssignmentID IN (SELECT AssignmentID FROM Claims));

4.SELECT * FROM Policies WHERE PremiumAmount > (SELECT AVG(PremiumAmount) FROM Policies);

5.SELECT * FROM Claims WHERE ClaimAmount > (SELECT AVG(ClaimAmount) FROM Claims);

11. Aggregate Functions

1.SELECT AVG(PremiumAmount) FROM Policies;
2.SELECT COUNT(*) FROM Customers;

3.SELECT SUM(ClaimAmount) FROM Claims;

4.SELECT MAX(DurationYears) FROM Policies;

5.SELECT MIN(ClaimAmount) FROM Claims;

12.String & Date Functions

1.SELECT CONCAT(FirstName,' ',LastName) FROM Customers;

2.SELECT LEN(AgentName) FROM Agents;

3.SELECT YEAR(ClaimDate) FROM Claims;

4.SELECT DATEDIFF(DAY, StartDate, EndDate) FROM PolicyAssignments;

5.SELECT UPPER(City) FROM Agents;

13.SET OPERATIONS

1.SELECT FirstName FROM Customers UNION SELECT AgentName FROM Agents;

2.SELECT FirstName FROM Customers UNION ALL SELECT AgentName FROM Agents;

3. SELECT FirstName FROM Customers INTERSECT SELECT AgentName FROM Agents;

4. SELECT FirstName FROM Customers EXCEPT SELECT AgentName FROM Agents;

5.SELECT Phone FROM Customers UNION SELECT Phone FROM Agents;

14. CASE - ELSE QUERY

```
SELECT PolicyName,
CASE
WHEN PremiumAmount > 5000 THEN 'High Premium'
WHEN PremiumAmount BETWEEN 3000 AND 5000 THEN 'Medium Premium'
ELSE 'Low Premium'
END AS PremiumCategory
FROM Policies;
```

15.GROUP BY ROLLUP

SELECT CustomerID, COUNT(*) FROM PolicyAssignments GROUP BY ROLLUP(CustomerID);

16.GROUP BY CUBE

SELECT PolicyID, CustomerID, COUNT(*) FROM PolicyAssignments GROUP BY CUBE(PolicyID, CustomerID);

17.MERGE QUERY

MERGE Policies AS T USING (SELECT 1 AS PolicyID, 'Health' AS PolicyName, 'Medical' AS PolicyType, 5000 AS PremiumAmount, 5 AS DurationYears) AS S ON T.PolicyID=S.PolicyID WHEN MATCHED THEN UPDATE SET PremiumAmount=S.PremiumAmount WHEN NOT MATCHED THEN INSERT VALUES (S.PolicyName,S.PolicyType,S.PremiumAmount,S.DurationYears);

18.GROUP BY GROUPING SETS

SELECT CustomerID, PolicyID, COUNT(*) FROM PolicyAssignments GROUP BY GROUPING SETS ((CustomerID), (PolicyID));