Module 4.4 Practical Project Assignment

Name: Koushik Gaddam

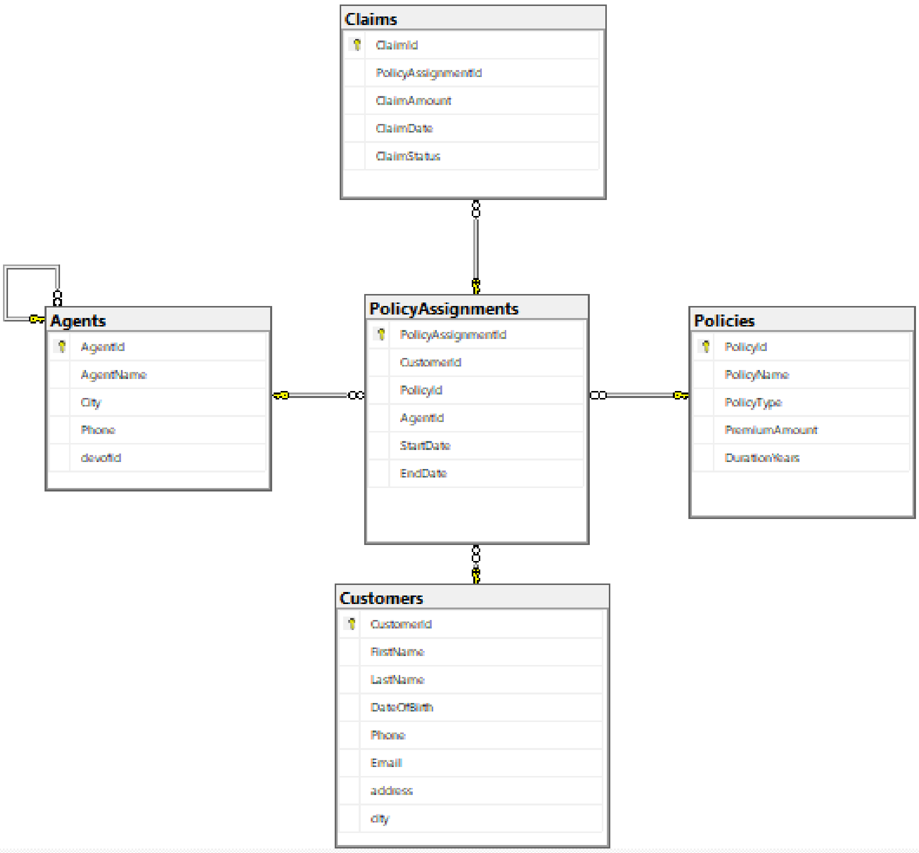
Email: koushikgaddam04@gmail.com

Date: 31/12/25

Creating a database:

```
Create database InsuranceDB;
```

Schema Diagram:



Create Table Commands:

```
create table customers(
        CustomerID int identity primary key,
        FirstName varchar(25),
        LastName varchar(25),
        DateOfBirth date,
        Phone varchar(15),
        email varchar(50));

create table Policies(
        PolicyID int identity primary key,
        PolicyName varchar(50),
        PolicyType varchar(50),
        PremiumAmount int,
        DurationYears int
);

create table Agents(
        AgentID int identity primary key,
        AgentName varchar(50),
        Phone float(10),
        City varchar(30)
)

create table PolicyAssignments(
        AssignmentID int identity primary key,
        CustomerID int,
        AgentID int,
        StartDate date,
        EndDate date
        constraint for_key_CustID Foreign Key (CustomerID) references
Customers(CustomerID),
        constraint for_key_AgentID Foreign Key (AgentID) references Agents(AgentID)
)

CREATE TABLE Claims (
   ClientID int IDENTITY PRIMARY KEY,
   AssignmentID int,
   claimDate date,
   claimAmount int,
   claimStatus int,
   CONSTRAINT for_key_assignID FOREIGN KEY (AssignmentID) REFERENCES
PolicyAssignments(AssignmentID)
);
```

Insert commands:

```
        insert into  Customers (FirstName, LastName, DateOfBirth, Phone, Email) values
        ('Koushik', 'Reddy', '1995-05-15', '9876543210', 'koushikreddy@gmail.com'),
        ('Rishik', 'Sharma', '2005-03-22', '8765432109', 'rishiksharma@gmail.com'),
        ('Madhumitha', 'Patel', '2010-07-08', '7654321098', 'madhumithapatel@gmail.com'),
        ('Tejaswini', 'Kumar', '1985-11-30', '6543210987', 'tejaswinikumar@gmail.com'),
        ('Krishna', 'Singh', '2008-09-12', '5432109876', 'krishnasingh@gmail.com'),
        ('Anirudh', 'Nair', '2015-01-25', '4321098765', 'anirudhnair@gmail.com'),
        ('Navaneeth', 'Joshi', '1990-06-18', '3210987654', 'navaneethjoshi@gmail.com'),
        ('Sujith', 'Menon', '2002-12-05', '2109876543', 'sujithmenon@gmail.com'),
        ('Rikhil', 'Iyer', '2018-08-14', '1098765432', 'rikhilier@gmail.com'),
```

```sql
('Pabitha', 'Bhat', '2003-02-28', '9988776655', 'pabithabhat@gmail.com'),
('Abhijeet', 'Verma', '1998-04-10', '9876543211', 'abhijeetverma@gmail.com'),
('Kishore', 'Pillai', '2007-08-03', '8765432110', 'kishorepillai@gmail.com'),
('Ganesh', 'Chopra', '2012-12-17', '7654321109', 'ganeshchopra@gmail.com'),
('Manikumar', 'Das', '2004-03-29', '6543211098', 'manikumardas@gmail.com'),
('Srinayani', 'Bansal', '2016-07-14', '5432109877', 'srinayanibansal@gmail.com'),
('Spoorthy', 'Gupta', '2009-06-20', '9876543222', 'spoorthygupta@gmail.com');

Insert into Agents (AgentName, Phone, City) values
('Ruthvik Gupta', '9123456789', 'Jaipur'),
('Koushik Desai', '9234567890', 'Mumbai'),
('Rishik Kapoor', '9345678901', 'Bengaluru'),
('Madhumitha Rao', '9456789012', 'Chennai'),
('Tejaswini Mehta', '9567890123', 'Nagpur'),
('Krishna Thakur', '9678901234', 'Pune'),
('Anirudh Malhotra', '9789012345', 'Jaipur'),
('Navaneeth Pillai', '9890123456', 'Bengaluru'),
('Sujith Verma', '9901234567', 'Delhi'),
('Rikhil Das', '9912345678', 'Kanpur');

Insert into PolicyAssignments(CustomerID, AgentID, PolicyID, StartDate, EndDate) values (1,1,1,'2024-01-15','2026-01-15'),
(2,2,2,'2024-02-20','2034-02-20'),
(3,3,3,'2024-03-10','2025-03-10'),
(4,4,4,'2024-04-05','2027-04-05'),
(5,5,1,'2024-05-12','2026-05-12'),
(5,6,7,'2024-06-18','2026-06-18'),
(5,7,9,'2024-07-22','2026-07-22'),
(6,8,2,'2024-08-30','2026-08-30'),
(7,9,5,'2024-09-14','2025-09-14'),
(8,10,8,'2024-10-25','2026-10-25'),
(9,1,10,'2024-11-03','2025-11-03'),
(10,2,11,'2024-12-08','2025-12-08'),
(11,3,6,'2025-01-14','2027-01-14'),
(12,4,12,'2025-02-19','2026-02-19'),
(13,5,3,'2025-03-25','2026-03-25'),
(16,6,4,'2025-04-01','2027-04-01');

Insert into Policies (PolicyName, PolicyType, PremiumAmount, DurationYears) values
('HealthGuard Pro', 'Health', 25000, 2),
('LifeSecure Plus', 'Life', 45000, 10),
('AutoShield Elite', 'Motor', 18000, 1),
('HomeProtect Max', 'Home', 12000, 3),
('TravelSafe Premium', 'Travel', 8000, 1),
('EduFuture Plan', 'Education', 30000, 5),
('CancerCare Shield', 'Health', 22000, 1),
('TermLife Basic', 'Life', 32000, 20),
('HeartCare Plan', 'Health', 28000, 3),
('CarComprehensive', 'Motor', 25000, 1),
('FamilyHealth Basic', 'Health', 15000, 1),

('BikeShield Pro', 'Motor', 7500, 1);


Insert into Claims (AssignmentID, ClaimDate, ClaimAmount, ClaimStatus) values

(1,'2024-11-10',50000,’Rejected’),

(2,'2024-12-05',250000, ‘Accepted’),

(3,'2025-01-18',35000, ‘Accepted’),

(4,'2025-02-22',80000, Rejected’),

(5,'2025-03-15',12000, ‘Accepted’),

(6,'2025-04-08',75000, ‘Accepted’),

(7,'2025-05-20',45000, Rejected’),

(8,'2025-06-12',28000, ‘Accepted’),

(9,'2025-07-25',95000, ‘Accepted’),

(10,'2025-08-30',15000, Rejected’),

(11,'2025-09-15',32000, ‘Accepted’),

(12,'2025-10-20',180000, Rejected’),

(13,'2025-11-25',45000, ‘Accepted’),

(1,'2025-12-01',65000, Rejected’),

(2,'2025-12-15',22000, ‘Accepted’);
```

Simple Querries:

- **Select policies where policy type is Health:**

  ```
  Select * from Policies where PolicyType = ‘Health’;
  ```

- **Select the number of policies where policy type is Life:**

  ```
  Select count(*) from Policies where PolicyType = ‘Life’;
  ```

- **Select the total Premium amount for policy type Motor:**

  ```
  Select sum(PremiumAmount) from Policies where PolicyType = ‘Motor’;
  ```

- **Display all the customer’s firstname, phone number and their email address:**

  ```
  Select FirstName, Phone, Email from Customers;
  ```

- **Display the customers who live in Bengaluru:**

  ```
  Select * from Customers where city = ‘Bengaluru ’;
  ```

Update Querries:

- **Increment Premium amount of policies by 10% for policy type Health:**

  ```
  Update Policies set PremiumAmount = PremiumAmount*1.1 where PolicyType=’Health’;
  ```

- **Update the value of Duration years of policy ID 3 to 8:**

  ```
  Update Policies set DurationYears=8 where PolicyID=3;
  ```

- **Update the city to Hyderabad for AgentID name Krishna:**

  ```
  Update Agents set city=’Hyderabad’ where AgentName=’Krishna’
  ```

- **Update the phone number to 9876543210 to the agentID 4:**

  ```
  Update Agents Phone=9876543210 where AgentID=4;
  ```

Alter Commands:

- **Drop a column city in Customers:**

  ```
  Alter table Customers drop column City;
  ```

- **Add a column City into Customers:**

  ```
  Alter table Customers add City varchar(50);
  ```

Delete Commands:

- **Delete records where policy type is education:**

  ```
  Delete from Policies where PolicyType=’Education’;
  ```

- **Remove the records where Premium Amount is less than 50000:**

  ```
  Delete from Policies where PremiumAmount < 50000;
  ```

Group By and Order by functions:

- **Calculate total Premium amount for every Policy type:**

  ```
  Select sum(PremiumAmount) from Policies group by PolicyType;
  ```

- **Calculate claimAmount and sort them by status:**

  ```
  select ClaimStatus, sum(ClaimAmount) from Claims group by
  ClaimStatus order by ClaimStatus;
  ```

- **Calculate average premium amount per policy type and display in descending order of average amount:**

  ```
  select PolicyType, avg(PremiumAmount) from Policies group by
  PolicyType order by AvgPremium desc;
  ```

- **Display number of policies by duration years, ordered by duration:**

  ```
  select DurationYears, count(PolicyID) as from Policies group by
  DurationYears order by DurationYears;
  ```

- **Show total premium amount by policy type, highest first:**

  ```
  select PolicyType, sum(PremiumAmount) from Policies group by
  PolicyType order by TotalPremium desc;
  ```

Having:

Agents with more than 3 claims:

```
select agentid,count(*) from claims group by agentid having count(*)>3;
```

Policies with total claims over 20,000:

```
select p.policyname,sum(c.claimamount) from policyassignments p join
claims c on p.policyid=c.policyid group by p.policyname having sum(c.claimamount)>20000;
```

Clients with average claim > 5000:

```
select clientid,avg(claimamount) from claims group by clientid having
avg(claimamount)>5000;
```

Joins:

- **Display customer names with their policy count:**

  ```
  select c.FirstName, c.LastName, count(pa.AssignmentID) as
  PolicyCount from Customers c join PolicyAssignments pa on
  c.CustomerID = pa.CustomerID group by c.CustomerID, c.FirstName,
  c.LastName order by PolicyCount desc;
  ```

- **Show agent names with total assignments:**

  ```
  select a.AgentName, count(pa.AssignmentID) as AssignmentCount
  from Agents a join PolicyAssignments pa on a.AgentID = pa.AgentID
  group by a.AgentID, a.AgentName order by AssignmentCount desc;
  ```

- **List agents with assignment dates:**

  ```
  select a.AgentName, pa.StartDate from Agents a join
  PolicyAssignments pa on a.AgentID = pa.AgentID;
  ```

- **Show policy names with claim amounts:**

  ```
  select p.PolicyName, cl.ClaimAmount from Policies p join
  PolicyAssignments pa on p.PolicyID = pa.PolicyID join Claims cl on
  pa.AssignmentID = cl.AssignmentID;
  ```

- **Show each agent and their development officer name:**

  ```
  select a.AgentName, p.AgentName as DevOfficer from Agents a left
  join Agents p on a.DevOfId = p.AgentID;
  ```

String Functions:

- **Concatenate agent name with ID:**

  ```
  select concat(agentname,' (ID:',agentid,')') from agents;
  ```

- **Show length of agent names:**

  ```
  select length(agentname), char_length(agentname) from
  agents;
  ```

- **Remove leading/trailing spaces:**

  ```
  select trim(agentname) from agents;
  ```

- **Replace 'Health' with 'Medical’:**

```
select replace(policyname,'Health','Medical') from
policyassignments;
```

Date Functions:

- **Show policies ending this year:**
  ```
  select PolicyName, EndDate from PolicyAssignments where
  year(EndDate) = 2025;
  ```

- **Display days between claim date and today:**
  ```
  select ClaimAmount, datediff(day, ClaimDate, getdate()) from Claims;
  ```

- **List claims filed in December:**
  ```
  select ClientID, ClaimDate from Claims where month(ClaimDate) = 12;
  ```

- **Show policies starting after June 2024:**
  ```
  select PolicyName, StartDate from PolicyAssignments where StartDate
  > '2024-06-01';
  ```

- **Show policies expiring within next 30 days with exact days remaining:**
  ```
  select PolicyName, EndDate, datediff(day, getdate(), EndDate) as
  DaysRemaining from PolicyAssignments where datediff(day, getdate(),
  EndDate) <= 30 and EndDate > getdate() order by DaysRemaining;
  ```

- **Add 6 months to policy start date and compare with end date:**
  ```
  select PolicyName, StartDate, dateadd(month, 6, StartDate),
  EndDate,case when dateadd(month, 6, StartDate) < EndDate then
  'Long Term' else 'Short Term' end from PolicyAssignments;
  ```

- **Find policies that started on weekdays only:**
  ```
  select PolicyName, StartDate, datename(weekday, StartDate) as
  DayOfWeek from PolicyAssignments where datepart(weekday,
  StartDate) between 2 and 6
  ```

Subquerries:

- **Show clients with policies:**

  ```
  select ClientName from Clients where ClientID in (select ClientID from PolicyAssignments);
  ```

- **Agents handling claims above average:**

  ```
  select AgentName from Agents where AgentID in (select AgentID from Claims where ClaimAmount > (select avg(ClaimAmount) from Claims));
  ```

- **Clients with above-average claims for their agent:**

  ```
  select c.ClientName, cl.ClaimAmount from Clients c join Claims cl on c.ClientID = cl.ClientID where cl.ClaimAmount > (select avg(ClaimAmount) from Claims c2 where c2.AgentID = cl.AgentID);
  ```

- **Agents with active policies:**

  ```
  select AgentName from Agents a where exists (select 1 from PolicyAssignments p where p.AgentID = a.AgentID and p.EndDate > getdate());
  ```

- **Policies with claims filed:**

  ```
  select PolicyName from PolicyAssignments p where exists (select 1 from Claims c where c.PolicyID = p.PolicyID);
  ```

Merging:

Sync PolicyAssignments with new policy staging data:

```
Merge PolicyAssignments as target
Using  (values
   (1, 'Health Pro', '2025-01-01', '2026-01-01', 1),
   (5, 'Auto Elite', '2025-03-01', '2026-03-01', 2)
) as source (PolicyID, PolicyName, StartDate, EndDate, AgentID)
On target.PolicyID = source.PolicyID
WHEN MATCHED THEN
   UPDATE SET
```

```
        PolicyName = source.PolicyName,
        StartDate = source.StartDate,
        EndDate = source.EndDate,
        AgentID = source.AgentID
WHEN NOT MATCHED BY TARGET THEN
    INSERT (PolicyID, PolicyName, StartDate, EndDate, AgentID)
    VALUES (source.PolicyID, source.PolicyName, source.StartDate, source.EndDate, source.AgentID);
WHEN NOT MATCHED BY SOURCE THEN
Delete;
```

Rollup: Hierarchical subtotals from right-to-left (detail → subtotals → grand total)

Claims summary by Agent, Month with subtotals

```
            SELECT
    AgentID,
    DATENAME(MONTH, ClaimDate) as MonthName,
    COUNT(*) as ClaimCount,
    SUM(ClaimAmount) as TotalClaims,
    GROUPING(AgentID) as IsAgentTotal,
    GROUPING(MONTH(ClaimDate)) as IsMonthTotal
FROM Claims
GROUP BY ROLLUP(AgentID, MONTH(ClaimDate), DATENAME(MONTH, ClaimDate))
ORDER BY AgentID, MONTH(ClaimDate);
```

Cube: All possible combinations of grouping columns (cross-tabulation).

Full claims analysis by Agent and Month:

```
SELECT
   ISNULL(AgentName, 'ALL AGENTS') as Agent,
   ISNULL(DATENAME(MONTH, ClaimDate), 'ALL MONTHS') as Month,
   COUNT(*) as ClaimCount,
   SUM(ClaimAmount) as TotalAmount
FROM Claims c
JOIN Agents a ON c.AgentID = a.AgentID
GROUP BY CUBE (a.AgentName, MONTH(ClaimDate))
ORDER BY AgentName, MONTH(ClaimDate);
```

Grouping Sets: Custom selection of specific grouping combinations.

Claims report with Agent totals, Month totals, AND grand total:

```
          SELECT
   ISNULL(AgentName, 'ALL AGENTS') as Agent,
   ISNULL(DATENAME(MONTH, ClaimDate), 'ALL MONTHS') as Month,
   COUNT(*) as ClaimCount,
   SUM(ClaimAmount) as TotalAmount,
   GROUPING_ID(AgentName, MONTH(ClaimDate)) as GroupingLevel
FROM Claims c
JOIN Agents a ON c.AgentID = a.AgentID
GROUP BY GROUPING SETS (
   (a.AgentName, MONTH(ClaimDate)),
   (a.AgentName),
   (MONTH(ClaimDate)),
   ()
)
```

```
ORDER BY Agent, Month;
```

Views:

Create a view to check for active policies:

```
create view activepolicies as select policyname,agentname,startdate,enddate from
policyassignments p join agents a on p.agentid=a.agentid where enddate > getdate();
```

Use the view:

```
select * from activepolicies;
```

Clustered Index: primary key automatically becomes clustered index in MySQL InnoDB - physically sorts table data.

```
create table claims (claimid int primary key auto_increment, clientid int, claimamount
decimal(10,2), claimdate date) engine=innodb;
```