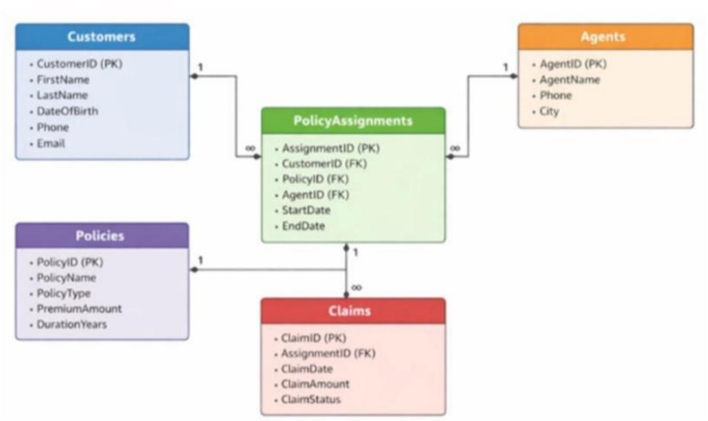
Module 4.4 Practical Project Assignment

-Gurumetkal Bharath Simha Reddy

Database Schema:



Database Creation:

```
create database insuranceDB;
```

Creation of Tables:

```
create table Customers(CustomerID int identity primary key,FirstName varchar(30) not null,LastName
varchar(30) not null,DateOfBirth date,Phone varchar(20) not null,Email varchar(20) not null unique);

create table Policies(PolicyID int identity primary key,PolicyName varchar(30) not null,PolicyType
varchar(30) not null,PremiumAmount decimal(10,2) not null,DurationYears int not null);


create table Agents(AgentID int identity primary key,AgentName varchar(30) not null,Phone varchar(20)
not null,City varchar(20) not null);

create table PolicyAssignments(AssignmentID int primary key,CustomerID int not null,PolicyID int not
null,AgentID int,StartDate date not null,EndDate date not null,CONSTRAINT FK_PA_Customers
 FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
 CONSTRAINT FK_PA_Policies
 FOREIGN KEY (PolicyID) REFERENCES Policies(PolicyID),
 CONSTRAINT FK_PA_Agents
 FOREIGN KEY (AgentID) REFERENCES Agents(AgentID));


create table Claims(ClaimID int identity primary key,AssignmentID int not null,ClaimDate date not
null,ClaimAmount decimal(10,2) not null,ClaimStatus varchar(20) not null,constraint FK_Claims foreign
key (AssignmentID) references PolicyAssignments(AssignmentID));
```

Insertion Commands:

```
INSERT INTO Customers (FirstName, LastName, DateOfBirth, Phone, Email) VALUES
('Bharath','Gurumetkal','2003-10-05','9566225566','bharath@gmail.com'),
('abhi','ram','2004-09-05','6655225566','abhi@gmail.com'),
('sai','sesh','2001-05-22','8899669966','sai@gmail.com'),
('harshini','gupta','2005-06-06','9566225566','harsh@gmail.com');


INSERT INTO Policies (PolicyName, PolicyType, PremiumAmount, DurationYears) VALUES
('Health Secure','Health',10000.00,1),
('Life Secure','Life',25000.00,2),
('Child Future','Education',12000.00,3),
('Retirement Plus','Pension',30000.00,1),
('Home Safety','Property',20000.00,2);


insert into Customers (FirstName, LastName, DateOfBirth, Phone, Email) values('ramesh','rao','2005-10-10','8989896633','ramesh@gmail.com');

INSERT INTO Agents (AgentName, Phone, City) VALUES
('Rahul', '9876543210', 'Delhi'),
('sai','9123456780','Mumbai'),
('Ram','8933456780','Pune'),
('Shyam','6655889900','Hyderabad')

insert into Agents values('Abhi','9988663355','Bangalore');

INSERT INTO PolicyAssignments VALUES
(1, 1, 1, 1, '2023-01-01', '2024-01-01'),
(2, 2, 2, 2, '2023-03-15', '2025-03-14'),
(3, 3, 3, 1, '2022-06-01', '2025-05-31');

INSERT INTO PolicyAssignments VALUES
(4, 5, 4, 2, '2024-01-01', '2025-01-01'),
(5, 5, 5, 1, '2023-09-01', '2024-09-01');



INSERT INTO Claims (AssignmentID, ClaimDate, ClaimAmount, ClaimStatus) VALUES
(1, '2023-06-10', 5000.00, 'Approved'),
(2, '2024-01-05', 8000.00, 'Pending'),
(3, '2023-11-20', 3000.00, 'Rejected');
```

Basic Select Commands:

```
select * from Customers;


select * from Agents;

select * from PolicyAssignments;

select * from Claims;
```

Querying Data:

```
select CustomerID,PolicyID,StartDate,EndDate from PolicyAssignments;

select * from Policies where PolicyType='Health';

select * from Policies where PremiumAmount>10000 and DurationYears=1;
```

```sql
select distinct City from Agents;

select * from Policies where PolicyType='Life' or PolicyType='Health' or PolicyType='Motor';

select * from Policies where PolicyType IN ('Life','Health','Motor');

select * from Customers where DateOfBirth between '2001-01-01' AND '2020-12-31';

select * from Claims where ClaimStatus='Rejected';

select * from Agents where City like '_a%';

select MAX(ClaimAmount) as highest_claimamount,MIN(ClaimAmount) as  highest_claimamount from
Claims;

select top 1 * from Claims order by ClaimDate desc;

update Policies set PremiumAmount=PremiumAmount*1.10 where PolicyType='Health';

delete from PolicyAssignments where EndDate <GETDATE();

select count(ClaimStatus) as rejected_count from Claims where ClaimStatus='Rejected';

SELECT
    PolicyID,
    PolicyName,
    PremiumAmount,
    PremiumAmount * 0.06 AS LocalTaxes,
    PremiumAmount + (PremiumAmount * 0.06) AS PremiumAmountWithTax,
    PremiumAmount / 12.0 AS MonthlyPremiumAmount
FROM Policies;
```

Alter Commands:

```sql
ALTER table Customers ADD Address varchar(50),City varchar(30);

alter table Agents Add DevOfId int;

alter table Agents
Add constraint fk_Agents_DevofId
foreign key (DevOfId)
references Agents(AgentID);
```

Functions:

```sql
--Display LastName in UPPERCASE
select upper(LastName) from Customers;

--Find customers whose LastName length is more than 6 characters.
select * from Customers where len(LastName)>6;

--Display the first 3 characters of FirstName.
select SUBSTRING(FirstName,1,3) from Customers;

--Replace the word ‘Life’ with ‘Term Life’ in PolicyType.
select replace(PolicyType,'Life','Term Life') from Policies;

--Find customers whose City contains ‘del’
select * from Agents where City like '%del%';
```

```
--Remove leading and trailing spaces from customer FirstName
select trim(FirstName) from Customers;

--Display PolicyType and its character length.
select PolicyType ,len(PolicyType) from Policies;

---Extract last 4 characters from PolicyType

select right(FirstName,3) from Customers;
```

Date and Time:

```
--  Get current date and current date-time
SELECT
    GETDATE() AS Current_DateTime,          -- returns current date & time
    CAST(GETDATE() AS DATE) AS Today_Date;  -- returns only current date

--  Calculate age of customers using DateOfBirth
SELECT
    CustomerID,
    FirstName,
    DateOfBirth,
    DATEDIFF(YEAR, DateOfBirth, GETDATE()) AS Age
FROM Customers;

--  Display policies that started in the current year
SELECT *
FROM PolicyAssignments
WHERE YEAR(StartDate) = YEAR(GETDATE());

--  Display policies expiring within the next 6 months
SELECT
    AssignmentID,
    CustomerID,
    PolicyID,
    EndDate
FROM PolicyAssignments
WHERE EndDate BETWEEN GETDATE() AND DATEADD(MONTH, 6, GETDATE());

--  Display duration of each policy in number of days
SELECT
    AssignmentID,
    StartDate,
    EndDate,
    DATEDIFF(DAY, StartDate, EndDate) AS Duration_Days
FROM PolicyAssignments;

--  Display claims filed in the year 2023
SELECT *
FROM Claims
WHERE YEAR(ClaimDate) = 2023;

--  Display month name in which each claim was raised
SELECT
    ClaimID,
    ClaimDate,
    DATENAME(MONTH, ClaimDate) AS Claim_Month
FROM Claims;

--  Display year and month of policy start date
SELECT
    AssignmentID,
    YEAR(StartDate) AS Start_Year,
    MONTH(StartDate) AS Start_Month
FROM PolicyAssignments;

-- Display number of policies started in each year
SELECT
    YEAR(StartDate) AS Start_Year,
    COUNT(*) AS No_of_Policies
FROM PolicyAssignments
GROUP BY YEAR(StartDate)
ORDER BY Start_Year;
```

JOINS:

```
select
    p.PolicyID,
    p.PolicyName,
    p.PolicyType,
    p.PremiumAmount,
    pa.StartDate,
    pa.EndDate
from PolicyAssignments pa
JOIN Policies p
    on pa.PolicyID = p.PolicyID
where pa.CustomerID = 5;

select
    c.FirstName,
    c.LastName,
    p.PolicyID,
    p.PolicyName,
    p.PolicyType,
    p.PremiumAmount,
    pa.StartDate,
    pa.EndDate
from Customers c
JOIN PolicyAssignments pa
    on c.CustomerID = pa.CustomerID
JOIN Policies p
    on pa.PolicyID = p.PolicyID
order by c.CustomerID;

select
c.FirstName,c.LastName,cl.ClaimID,cl.ClaimDate,cl.ClaimAmount,cl.ClaimStatus
from Claims cl
JOIN PolicyAssignments pa
 on cl.AssignmentID = pa.AssignmentID
JOIN Customers c
on pa.CustomerID = c.CustomerID;

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 pa.AgentID = a.AgentID;

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;
```

```
select c.CustomerID,c.FirstName,c.LastName,p.PolicyName,p.PolicyType
from Customers c
left join PolicyAssignments pa
on c.CustomerID = pa.CustomerID
left join Policies p
on pa.PolicyID = p.PolicyID;

select c.CustomerID,c.FirstName,c.LastName
from Customers c left join PolicyAssignments pa
on c.CustomerID = pa.CustomerID
left join Claims cl on pa.AssignmentID = cl.AssignmentID
where cl.ClaimID IS NULL;

select c.FirstName+' ' + c.LastName as Customer_name,sum(cl.ClaimAmount) as Total_claim_amount
from Customers c join PolicyAssignments pa on c.CustomerID = pa.CustomerID
join Claims cl on pa.AssignmentID = cl.AssignmentID
group by c.FirstName, c.LastName;


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

select a.AgentName,count(pa.PolicyID) as policy_count from Agents a
left join PolicyAssignments pa on a.AgentID = pa.AgentID group by a.AgentID, a.AgentName;
```

Sub Queries:

```
--  Customers who have taken at least one policy
SELECT *
FROM Customers
WHERE CustomerID IN (
   SELECT CustomerID
   FROM PolicyAssignments
);

--  Customers who have NOT taken any policy
SELECT *
FROM Customers
WHERE CustomerID NOT IN (
   SELECT CustomerID
   FROM PolicyAssignments
);

--  Customers who have filed at least one claim
SELECT *
FROM Customers
WHERE CustomerID IN (
   SELECT CustomerID
   FROM PolicyAssignments
   WHERE AssignmentID IN (
      SELECT AssignmentID
      FROM Claims
   )
);

--  Policies with premium greater than the average premium
SELECT *
FROM Policies
WHERE PremiumAmount > (
   SELECT AVG(PremiumAmount)
```

```sql
    FROM Policies
);

-- Customers who have taken the policy with the highest premium
SELECT *
FROM Customers
WHERE CustomerID IN (
    SELECT CustomerID
    FROM PolicyAssignments
    WHERE PolicyID = (
        SELECT PolicyID
        FROM Policies
        WHERE PremiumAmount = (
            SELECT MAX(PremiumAmount)
            FROM Policies
        )
    )
);

-- Customers who have filed claims with amount greater than average claim amount
SELECT *
FROM Customers
WHERE CustomerID IN (
    SELECT CustomerID
    FROM PolicyAssignments
    WHERE AssignmentID IN (
        SELECT AssignmentID
        FROM Claims
        WHERE ClaimAmount > (
            SELECT AVG(ClaimAmount)
            FROM Claims
        )
    )
);

-- Agents who are assigned to at least one policy
SELECT *
FROM Agents
WHERE AgentID IN (
    SELECT AgentID
    FROM PolicyAssignments
);

-- Agents who are NOT assigned to any policy
SELECT *
FROM Agents
WHERE AgentID NOT IN (
    SELECT AgentID
    FROM PolicyAssignments
    WHERE AgentID IS NOT NULL
);

-- Customers who have taken more than one policy
SELECT *
FROM Customers
WHERE CustomerID IN (
    SELECT CustomerID
    FROM PolicyAssignments
    GROUP BY CustomerID
    HAVING COUNT(PolicyID) > 1
);

-- Policy assignments whose duration is longer than the average duration
```

```
SELECT *
FROM PolicyAssignments
WHERE DATEDIFF(DAY, StartDate, EndDate) > (
    SELECT AVG(DATEDIFF(DAY, StartDate, EndDate))
    FROM PolicyAssignments
);
```

Set Operations:

```
--  UNION

SELECT FirstName AS Name
FROM Customers
UNION
SELECT AgentName
FROM Agents;

--  UNION ALL

SELECT FirstName AS Name
FROM Customers
UNION ALL
SELECT AgentName
FROM Agents;

--  INTERSECT

SELECT FirstName AS Name
FROM Customers
INTERSECT
SELECT AgentName
FROM Agents;

--  EXCEPT

SELECT FirstName AS Name
FROM Customers
EXCEPT
SELECT AgentName
FROM Agents;
```

Case Else:

```
--  Categorize policies based on premium amount
SELECT
    PolicyName,
    PremiumAmount,
    CASE
        WHEN PremiumAmount >= 25000 THEN 'High Premium'
        WHEN PremiumAmount BETWEEN 15000 AND 24999 THEN 'Medium Premium'
        ELSE 'Low Premium'
    END AS Premium_Category
FROM Policies;


--  Categorize claim status meaning
SELECT
    ClaimID,
    ClaimAmount,
    CASE
        WHEN ClaimStatus = 'Approved' THEN 'Paid'
        WHEN ClaimStatus = 'Pending' THEN 'In Process'
        ELSE 'Not Paid'
    END AS Claim_Result
FROM Claims;


--  Check whether policy is active or expired
SELECT
    AssignmentID,
    EndDate,
    CASE
        WHEN EndDate >= GETDATE() THEN 'Active'
        ELSE 'Expired'
    END AS Policy_Status
FROM PolicyAssignments;


--  Categorize customers by age group
SELECT
    CustomerID,
    FirstName,
    CASE
        WHEN DATEDIFF(YEAR, DateOfBirth, GETDATE()) < 18 THEN 'Minor'
        WHEN DATEDIFF(YEAR, DateOfBirth, GETDATE()) BETWEEN 18 AND 40 THEN 'Adult'
        ELSE 'Senior'
    END AS Age_Group
FROM Customers;

--  Display discount eligibility based on policy duration
SELECT
    PolicyName,
    DurationYears,
    CASE
        WHEN DurationYears >= 3 THEN 'Eligible for Discount'
        ELSE 'Not Eligible'
    END AS Discount_Status
FROM Policies;
```

Roll Up:

```
--  Total premium amount by PolicyType with grand total
SELECT
    PolicyType,
    SUM(PremiumAmount) AS Total_Premium
FROM Policies
GROUP BY ROLLUP (PolicyType);


--  Number of policy assignments per customer with grand total
SELECT
    CustomerID,
    COUNT(*) AS No_of_Policies
FROM PolicyAssignments
GROUP BY ROLLUP (CustomerID);


--  Claims amount by status with grand total
SELECT
    ClaimStatus,
    SUM(ClaimAmount) AS Total_Claim_Amount
FROM Claims
```

```
GROUP BY ROLLUP (ClaimStatus);


--  Policies count by duration with grand total
SELECT
    DurationYears,
    COUNT(*) AS Policy_Count
FROM Policies
GROUP BY ROLLUP (DurationYears);


--  Policy assignments per year with grand total
SELECT
    YEAR(StartDate) AS Start_Year,
    COUNT(*) AS No_of_Assignments
FROM PolicyAssignments
GROUP BY ROLLUP (YEAR(StartDate));
```

Cube:

```
--  Premium summary by PolicyType and DurationYears
SELECT
   PolicyType,
   DurationYears,
   SUM(PremiumAmount) AS Total_Premium
FROM Policies
GROUP BY CUBE (PolicyType, DurationYears);
```

```
--  Claim amount summary by ClaimStatus and Year
SELECT
   ClaimStatus,
   YEAR(ClaimDate) AS Claim_Year,
   SUM(ClaimAmount) AS Total_Claim
FROM Claims
GROUP BY CUBE (ClaimStatus, YEAR(ClaimDate));
```

```
--  Policy assignments by Customer and Agent
SELECT
   CustomerID,
   AgentID,
   COUNT(*) AS No_of_Assignments
FROM PolicyAssignments
GROUP BY CUBE (CustomerID, AgentID);
```

```
--  Policies by PolicyType and Duration
SELECT
   PolicyType,
   DurationYears,
   COUNT(*) AS Policy_Count
FROM Policies
GROUP BY CUBE (PolicyType, DurationYears);
```

```
--  Claims by status and assignment
SELECT
   ClaimStatus,
   AssignmentID,
   SUM(ClaimAmount) AS Claim_Total
FROM Claims
GROUP BY CUBE (ClaimStatus, AssignmentID);
```

Grouping Sets:

```
--  Total premium by PolicyType AND grand total
SELECT
   PolicyType,
   SUM(PremiumAmount) AS Total_Premium
FROM Policies
GROUP BY GROUPING SETS (
   (PolicyType),   -- subtotal by PolicyType
   ()              -- grand total
);
```

```
--  Number of policy assignments by Customer and by Agent
SELECT
   CustomerID,
    AgentID,
    COUNT(*) AS No_of_Assignments
FROM PolicyAssignments
GROUP BY GROUPING SETS (
    (CustomerID),   -- subtotal per customer
    (AgentID)       -- subtotal per agent
);


--  Total claim amount by ClaimStatus and by Year
SELECT
    ClaimStatus,
    YEAR(ClaimDate) AS Claim_Year,
    SUM(ClaimAmount) AS Total_Claim_Amount
FROM Claims
GROUP BY GROUPING SETS (
    (ClaimStatus),              -- subtotal by status
    (YEAR(ClaimDate)),          -- subtotal by year
    ()                          -- grand total
);


--  Count policies by DurationYears and PolicyType separately
SELECT
    DurationYears,
    PolicyType,
    COUNT(*) AS Policy_Count
FROM Policies
GROUP BY GROUPING SETS (
    (DurationYears),    -- subtotal by duration
    (PolicyType)        -- subtotal by policy type
);


--  Policy assignments by Start Year and Customer
SELECT
    YEAR(StartDate) AS Start_Year,
    CustomerID,
    COUNT(*) AS No_of_Policies
FROM PolicyAssignments
GROUP BY GROUPING SETS (
    (YEAR(StartDate)),   -- subtotal by year
    (CustomerID),        -- subtotal by customer
    ()                   -- grand total
);
```