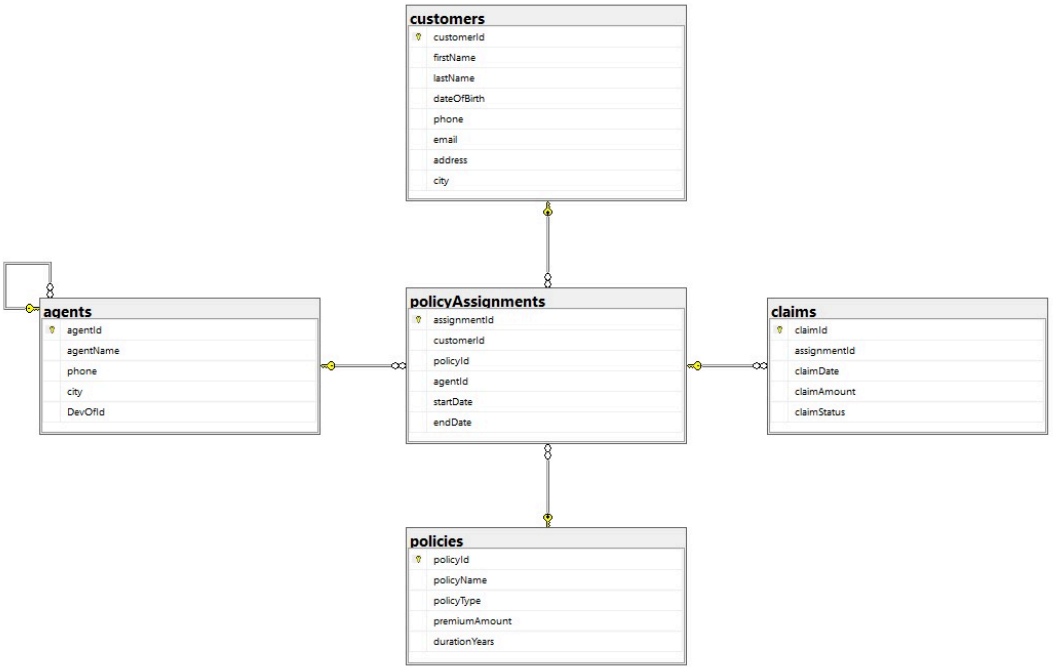
Module 4.4 Practical Project Assignment

1. Create a Database command.

Create database insurenceDB;

2.Database Diagram



3. Create table commands for all the tables with constraints, relationships etc.

```
create table Customers(
CustomerID int PRIMARY KEY,
FirstName varchar(100),
LastName varchar(100),
DateOfBirth date,
Phone varchar(20),
Email varchar(100) UNIQUE
);

create table Policies(
PolicyID int Primary key,
PolicyName varchar(100),
PolicyType varchar(100),
PremiumAmount DEcimal(10,2),
DurationYears int
);
```

```
create table Agents(
AgentID int Primary key,
AgentName varchar(50),
Phone varchar(20),
City varchar(50));

create table Claims(
ClaimsID int primary key,
AssignmentID int,
ClaimsDate date,
ClaimsAmount decimal(10,2),
ClaimStatus varchar(50),
);

create table PolicyAssignments(
AssignmentID int Primary key,
CustomerID int,
PolicyID int,
AgentID int ,
StartDate date,
EndDate date,
Constraint FK_Assign_Customer Foreign KEy(CustomerID)
References Customers(CustomerID),
Constraint FK_Assign_Policy Foreign KEy(PolicyID)
References Policies(PolicyID),
Constraint FK_Assign_Agent Foreign KEy(AgentID)
References Agents(AgentID),
);
```

4. Insert commands for all tables.

```
insert into  Customers
values
(101,'Manvitha','Reddy','2000-09-23','8999389997','manvithareddy23@gmail.com'),
(102,'Rajesh','Sharma','2000-04-15','895938967','rajeshsharma13@gmail.com')
(103,'Mohit','Oberoi','2001-04-17','8229389007','mohittob@gmail.com'),
(104,'Priya','Singh','2000-09-29','8922380017','priyas0929@gmail.com')
(105,'Sandhya','Rani','2002-09-03','8019389327','sandhya23@gmail.com')
(106,'Ayush','Mehta','2020-12-31','9948905679','ayushmmtaa@gmail.com');

insert into Agents
values
(1001,'Manoj','8897100567','Hyderabad'),
(1002,'Samanvi','9907100567','Mumbai');
(1003,'Krishna','8894600710','Pune'),
(1004,'Bhavani','9907132467','Mumbai')
(1005,'Arjun','980227857','Patna');
```

```
insert into Policies
values
(1,'Jeevan Sathi','Life Insurence',500000.00,5),
(2,'Acko','Car Insurence',100000.00,5)
(3,'Star Health','Health Insurence',250000.00,5),
(4,'Safar Yojana','Travel Insurence',100000.00,1)
(5,'Yolo Health','Health Insurence',450000.00,7)
(6,'Paw Care','Pet Insurence',15000.00,1);

insert into PolicyAssignments
values
(2001,101,1,1001,'2024-10-08','2029-10-08'),
(2002,102,2,1002,'2024-12-06','2029-12-06')
(2003,103,3,1003,'2024-07-07','2029-07-07'),
(2004,104,4,1004,'2024-02-26','2025-02-26');

insert into Claims
values
(1501,2001,'2024-10-08',50000.00,'Approved'),
(1502,2002,'2024-12-06',10000.00,'Rejected')
(1503,2003,'2024-07-07',25000.00,'Rejected'),
(1504,2004,'2024-02-26',9000.00,'Approved');
```

5. Select Commands

a.View all records of PolicyAssignment table with CustomerId, PolicyId, StartDate and EndDate columns only.

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

b.Display policies having premium amount more than 10000 and DurationYears is 1.

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

c.List policies of type Life, Health, Motor use OR clause.

```
select * from Policies where PolicyType='Life Insurence' or PolicyType='Car Insurence' or
PolicyType='Health Insurence';
```

d.List policies of type Life, Health, Motor use IN operator.

```
select *from Policies where PolicyType in ('Life Insurence','Car Insurence','Health
Insurence');
```

e.Display records of Agents who stay in a city whose second letter is ‘a’.

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

6.Group by, Having,Order by Commands

a.Show CustomerName with Total Claim Amount per Customer.

```
select
    c.FirstName,
    c.LastName,
    SUM(cl.ClaimsAmount) 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, c.LastName;
```

b.Show names and total claim amount of Customers With Claim Amount > 50000 (UseHAVING Clause).

```
select
    c.FirstName,
    c.LastName,
    SUM(cl.ClaimsAmount) 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, c.LastName
having SUM(cl.ClaimsAmount) > 50000;
```

c.Display list with Agent Wise Policy Count.

```
select
    a.AgentName,
    count(pa.PolicyID) AS PolicyCount
from Agents a
JOIN PolicyAssignments pa
on a.AgentID = pa.AgentID
group by a.AgentName;
```

d.Display latest claim record.

```
select top 1 * from Claims order by ClaimsDate desc;
```

e.Display agents ordered by City and AgentName.

```
select * from Agents order by City, AgentName;
```

7.Join Commands

a.View claims with customer name.

```
select c.CustomerID,c.FirstName,c.LastName,
       cl.ClaimsDate,cl.ClaimsAmount,cl.ClaimStatus
from Customers c
join  PolicyAssignments pa
   on c.CustomerID = pa.CustomerID
join Claims cl
    on pa.AssignmentID = cl.AssignmentID;
```

b.**Display FirstName, PolicyName, AgentName, StartDate and EndDate from their respective tables.**

```
Select
c.FirstName,
p.PolicyName,
a.AgentName,
pa.StartDate,
pa.EndDate
from
Customers c
join
PolicyAssignments pa
on c.CustomerID=pa.CustomerID
join Policies p
on pa.PolicyID=p.PolicyID
join Agents a
on pa.AgentID = a.AgentID;
```

c.Policies with no assignments.

```
Select
p.PolicyName,
p.PolicyType
from Policies p
left join
PolicyAssignments pa
on p.PolicyID=pa.PolicyID
left join Claims c
on pa.AssignmentID=c.AssignmentID
where
```

pa.AssignmentID is null;

d.Customers who purchased Life policy.

```
select
c.FirstName,
c.LastName
from Customers c
join PolicyAssignments pa
on c.CustomerID=pa.CustomerID
join Policies p
on pa.PolicyID=p.PolicyID
where p.PolicyType='Life insurence';
```

e.Display policy details along with claim amount (if any).

```
select
p.PolicyName,
p.PolicyType,
c.ClaimsAmount
from
Policies p
 join
PolicyAssignments pa
on p.PolicyID=pa.PolicyID
join Claims c
on pa.AssignmentID=c.AssignmentID;
```

8.Subqueries

a.Find customers who have taken premium above average.

```
select *from Customers c
join
PolicyAssignments pa
on
c.CustomerID =pa.CustomerID
join Policies p
on p.PolicyID=pa.PolicyID
where
p.PremiumAmount=(select avg(PremiumAmount) as avgPA from Policies);
```

b.Policies having the highest premium.

```
select *from Customers c
join
PolicyAssignments pa
on
c.CustomerID =pa.CustomerID
```

```
join Policies p
on p.PolicyID=pa.PolicyID
where
p.PremiumAmount=(select max(PremiumAmount) as avgPA from Policies);
```

c.Customers with maximum total premium.

```
 select
 c.FirstName , p.PremiumAmount
 from
 Customers c
 join
 PolicyAssignments pa
 on c.CustomerID=pa.CustomerID
 join Policies p
 on pa.PolicyID=p.PolicyID
 where p.PremiumAmount=(select max(PremiumAmount) from Policies);
```

d.Policies with claim amount greater than average claim.

```
select
p.PolicyName
from
Policies p
join
PolicyAssignments pa
on p.PolicyID=pa.PolicyID
join
Claims c
on pa.AssignmentID=c.AssignmentID
where c.ClaimsAmount>(select avg(ClaimsAmount) from Claims);
```

e.Find customers who have not taken any policy.

```
select FirstName, LastName
from Customers
where CustomerID not in (
    select CustomerID
    from PolicyAssignments
);
```

9.Date Functions

a.Calculate policy duration in days

```
select
    AssignmentId,
    DATEDIFF(DAY, StartDate, EndDate) AS PolicyDurationDays
```

```
from PolicyAssignments;
```

b.Display all active policies (where today is between StartDate and EndDate).

```
select *
from PolicyAssignments
where getdate() between StartDate and EndDate;
```

c.Display year and month of each claim.

```
select ClaimsID,
      year(ClaimsDate) as claim_year,
      month(ClaimsDate) as claim_month
from Claims;
```

d.Calculate customer age

```
select
   CustomerID,
   FirstName,
   DATEDIFF(YEAR, DateOfBirth, GETDATE()) AS Age
from Customers;
```

e.Show policy expiry month & year

```
select
   AssignmentID,
   MONTH(EndDate) AS ExpiryMonth,
   YEAR(EndDate) AS ExpiryYear
from PolicyAssignments;
```

10.String Functions

a.Display customer full name by concatenaϴng FirstName and LastName.

```
select
   CustomerID,
   CONCAT(FirstName, ' ', LastName) AS FullName
from Customers;
```

b.Display LastName in UPPERCASE

```
select
   CustomerID,
   FirstName,
   UPPER(LastName) as LastName_U
from Customers;
```

c.Display PolicyType and its character length.

```
select
   PolicyType,
   LEN(PolicyType) as CharacterLength
from Policies;
```

d.Reverse the FirstName

```
select
   FirstName,
   REVERSE(FirstName) as ReversedName
from Customers;
```

e.Extract username from Email (before @)

```
select
   Email,
   SUBSTRING(Email, 1, CHARINDEX('@', Email) - 1) AS EmailUserName
from Customers;
```

11.Set operations

a.Customers who have policies OR have claims

```
select CustomerID
from PolicyAssignments
UNION
select pa.CustomerID
from PolicyAssignments pa
Join Claims c
   on pa.AssignmentID = c.AssignmentID;
```

b.Customers who have policies AND have claims

```
select CustomerID
from PolicyAssignments
INTERSECT
select pa.CustomerID
from PolicyAssignments pa
Join Claims c
   on pa.AssignmentID = c.AssignmentID;
```

c.Policies that are either Health or Travel

```
select PolicyID
from Policies
where PolicyType = 'Health Insurence'
UNION
select PolicyID
```

```
from Policies
where PolicyType = 'Travel Insurence';
```

d.Common PolicyIDs appearing in both Assignments and Claims

```
select PolicyID
from PolicyAssignments
INTERSECT
select pa.PolicyID
from PolicyAssignments pa
Join Claims c
   on pa.AssignmentID = c.AssignmentID
```

e.Customers who have policies but are NOT agents

```
select CustomerID
from PolicyAssignments
EXCEPT
select AgentID
from Agents;
```

12.Rollback

a.Total claim amount by AssignmentID with grand total

```
select
   AssignmentID,
   SUM(ClaimsAmount) AS TotalClaimAmount
from Claims
group by ROLLUP(AssignmentID);
```

b.Total premium by DurationYears with overall total

```
select
   DurationYears,
   SUM(PremiumAmount) AS TotalPremium
from Policies
group by ROLLUP(DurationYears);
```

13.CUBE Commands

a.Total premium by PolicyType and PolicyID (all combinations)

```
select
   PolicyType,
   PolicyID,
   SUM(PremiumAmount) AS TotalPremium
from Policies
group by CUBE(PolicyType, PolicyID);
```

b.Total claims by AssignmentID and ClaimStatus

```
select
```

```
    AssignmentID,
    ClaimStatus,
    SUM(ClaimsAmount) AS TotalAmount
from Claims
group by CUBE(AssignmentID, ClaimStatus);
```

13.Grouping Sets commands

a.Total premium by PolicyType and PolicyID

```
select
    PolicyType,
    PolicyID,
    SUM(PremiumAmount) AS TotalPremium
from Policies
group by GROUPING SETS (
    (PolicyType),
    (PolicyID)
);
```

b.Total claims by Year only and Status only

```
select
    ClaimStatus,
    YEAR(ClaimsDate) AS ClaimYear,
    SUM(ClaimsAmount) AS TotalAmount
from Claims
Group by GROUPING SETS (
    (ClaimStatus),
    (YEAR(ClaimsDate)));
```

14. Merge Commands

a.MERGE to update ClaimStatus

```
MERGE Claims as target
using (SELECT 1503 AS ClaimID, 'Approved' AS ClaimStatus) AS source
on target.ClaimID = source.ClaimID
when matched then
    Update set target.ClaimStatus = source.ClaimStatus;
```

b.MERGE to update PremiumAmount only

```
MERGE Policies as target
using (SELECT 1 AS PolicyID, 520000 AS PremiumAmount) AS source
on target.PolicyID = source.PolicyID
when matched then
    Update settarget.PremiumAmount = source.PremiumAmount;
```