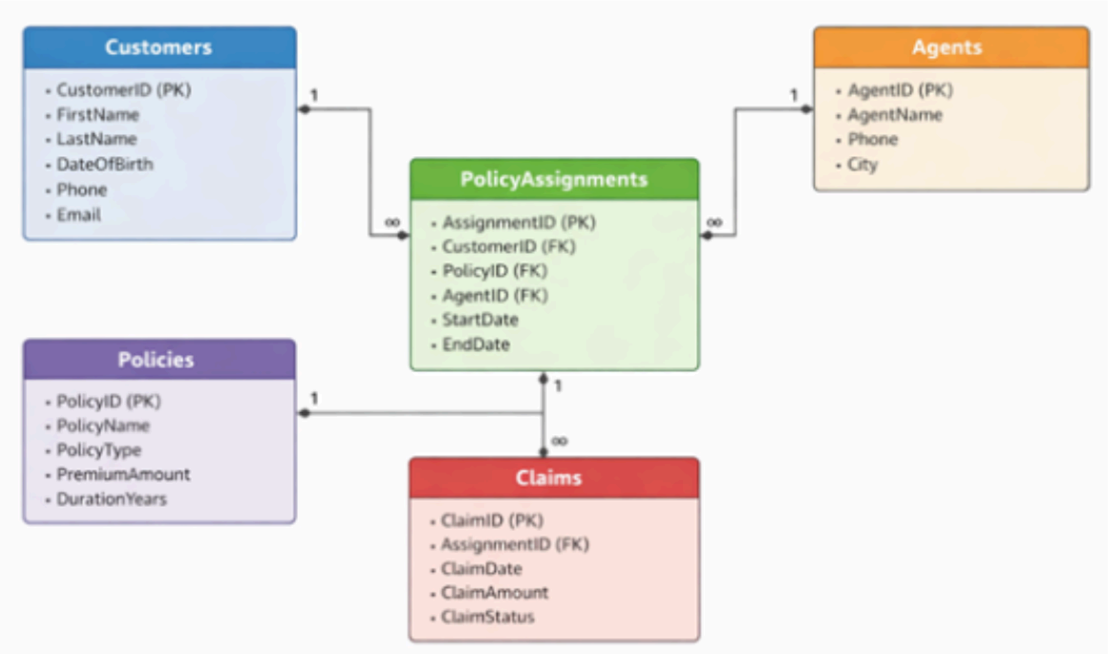
Module 4.4 Practical project assignment
RATNA JASHWANTH



DATABASE CREATION
Creation and use of database

```
create database InsuranceDB;

use InsuranceDB;
```

Creation of tables

```
create table Customers(
        CustomerID int identity 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 primary key,
        PolicyName varchar(20),
        PolicyType varchar(20),
        PremiumAmount decimal(10,2),
        DurationYear int
)

create table Agents(
        AgentID int identity primary key,
        AgentName varchar(20),
        Phone varchar(15),
        City varchar(20)
)

create table PolicyAssignments(
        AssignmentID int identity primary key,
        CustomerID int,
        PolicyID int,
        AgentID int,
        StartDate date,
        EndDate date,
        constraint fk_customerid foreign key(CustomerID) references
Customers(CustomerID),
        constraint fk_policyid foreign key(PolicyID) references Policies(PolicyID),
        constraint fk_agentid foreign key(AgentID) references Agents(AgentID),
)

create table Claims(
        ClaimID int identity primary key,
        AssignmentID int,
        ClaimDate date,
        ClaimAmount decimal(10,2),
        ClaimStatus varchar(20),
        constraint fk_policyID_claims foreign key(AssignmentId) references
PolicyAssignments(AssignmentID)
```

INSERTION OF VALUES

```
INSERT INTO Customers (FirstName, LastName, DateOfBirth, Phone, Email)
VALUES
('Amit', 'Sharma', '1998-04-12', '9876543210', 'amit@gmail.com'),
('Priya', 'Verma', '2000-09-25', '9123456789', 'priya@gmail.com'),
('Rahul', 'Kumar', '1997-01-18', '9988776655', 'rahul@gmail.com'),
('Sneha', 'Reddy', '1999-06-30', '9012345678', 'sneha@gmail.com'),
('Arjun', 'Mehta', '1996-11-05', '8899776655', 'arjun@gmail.com');

INSERT INTO Policies (PolicyName, PolicyType, PremiumAmount, DurationYear)
VALUES
('Life Secure', 'Life', 15000.00, 20),
('Health Plus', 'Health', 8000.50, 5),
('Car Shield', 'Vehicle', 6200.75, 1),
('Home Protect', 'Property', 12000.00, 10),
('Travel Safe', 'Travel', 3500.00, 1);

INSERT INTO Agents (AgentName, Phone, City)
VALUES
('Ravi Kumar', '9876543210', 'Hyderabad'),
('Anita Singh', '9123456789', 'Bengaluru'),
('Suresh Rao', '9988776655', 'Chennai'),
('Neha Patel', '9012345678', 'Ahmedabad'),
('Vikram Shah', '8899776655', 'Mumbai');

INSERT INTO PolicyAssignments (CustomerID, PolicyID, AgentID, StartDate, EndDate)
VALUES
(1, 1, 1, '2023-01-01', '2043-01-01'),
(2, 2, 2, '2024-03-15', '2029-03-15'),
(3, 3, 3, '2024-06-01', '2025-06-01'),
(4, 4, 4, '2022-09-10', '2032-09-10'),
(5, 5, 5, '2024-01-01', '2025-01-01');
```

```
INSERT INTO Claims (AssignmentID, ClaimDate, ClaimAmount, ClaimStatus)
VALUES
(1, '2024-02-10', 25000.00, 'Approved'),
(2, '2024-05-18', 12000.50, 'Pending'),
(3, '2024-07-01', 45000.00, 'Rejected'),
(4, '2023-12-22', 18000.75, 'Approved'),
(5, '2024-08-05', 9000.00, 'Pending');
```

BASIC COMMANDS

1. display all customers

```
select * from customers;
```

2. display only firstname and email of customers

```
select firstname, email from customers;
```

3. display all policies

```
select * from policies;
```

4. display agent names and cities

```
select agentname, city from agents;
```

5. display all claims

```
select * from claims;
```

Data querying

1. find customers born after 1998

```
select * from customers where dateofbirth > '1998-01-01';
```

2. find policies with premium greater than 10000

```
select * from policies where premiumamount > 10000;
```

3. find claims with status = 'approved'

```
select * from claim where claimstatus = 'approved';
```

4. display agents from mumbai

```
select * from agents where city = 'mumbai';
```

5. display policies ordered by premium (high to low)

```
select * from policies order by premiumamount desc;
```

ALTER COMMANDS

1. add city column to customers

```
alter table customers add city varchar(20);
```

2. add status column to policies

```
alter table policies add status varchar(20);
```

3. increase email size

```
alter table customers alter column email varchar(50);
```

4. add commission column to agents

```
alter table agents add commission decimal(8,2);
```

Subqueries

1. customers who have taken a policy

```
select * from customers where customerid in (select customerid from policyassignments);
```

2. policies assigned to customers

```
select * from policies where policyid in (select policyid from policyassignments);
```

3. customers who made claims

```
select * from customers where customerid in (select customerid from policyassignments
where assignmentid in
(select assignmentid from claims));
```

4. highest premium policy

```
select * from policies where premiumamount = (select max(premiumamount) from policies);
```

5. policies not assigned

```
select * from policies where policyid not in (select policyid from policyassignments);
```

Joins

1. customer and policy details

```
select c.firstname, p.policyname from customers c
join policyassignments pa on c.customerid = pa.customerid
join policies p on pa.policyid = p.policyid;
```

2. customer and agent details

```
select c.firstname, a.agentname from customers c
join policyassignments pa on c.customerid = pa.customerid
join agents a on pa.agentid = a.agentid;
```

3. policy and claim details

```
select p.policyname, cl.claimamount from policies p
join policyassignments pa on p.policyid = pa.policyid
join claims cl on pa.assignmentid = cl.assignmentid;
```

4. agent with customers

```
select a.agentname, c.firstname from agents a
join policyassignments pa on a.agentid = pa.agentid
join customers c on pa.customerid = c.customerid;
```

Functions

1. total premium amount

```
select sum(premiumamount) as total_premium from policies;
```

2. average claim amount

```
select avg(claimamount) as avg_claim from claims;
```

3. count number of customers

```
select count(*) as total_customers from customers;
```

4. uppercase customer names

```
select upper(firstname) as name from customers;
```

5. length of policy name

```
select policyname, len(policyname) as length from policies;
```

Date & time functions

1. year of birth

```
select firstname, year(dateofbirth) as birth_year from customers;
```

2. policies expiring in 2025

```
select * from policyassignments where year(enddate) = 2025;
```

3. days between policy start and end

```
select datediff(day, startdate, enddate) as total_days from policyassignments;
```

4. month of claim

```
select claimid, month(claimdate) as claim_month from claims;
```

Set operators

Union

1. customers who either have an active policy or have raised a claim

```
select customerid from policyassignments
where enddate >= getdate()
union
select pa.customerid from claims c
join policyassignments pa
on c.assignmentid = pa.assignmentid;
```

2. policies that are health type or have claims raised

```
select policyid from policies
where policytype = 'health'
union
select pa.policyid from policyassignments pa
join claims c
on pa.assignmentid = c.assignmentid;
```

Union all

3. count how many times customers appear in assignments + claims

```
select customerid from policyassignments
union all
select pa.customeridfrom claims c
join policyassignments pa
on c.assignmentid = pa.assignmentid;
```

4. policy participation across modules

```
select policyid from policyassignments
union all
select policyid from policies;
```

Intersect

5. customers who have a policy AND have filed a claim

```
select customerid from policyassignments
intersect
select pa.customerid from claims c
join policyassignments pa
on c.assignmentid = pa.assignmentid;
```

6. policies that are assigned AND claimed

```
select policyid from policyassignments
intersect
select pa.policyid from claims c
join policyassignments pa
on c.assignmentid = pa.assignmentid;
```

Except

7. customers who have policies but never filed a claim

```
select customerid from policyassignments
except
select pa.customerid from claims c
join policyassignments pa
on c.assignmentid = pa.assignmentid;
```

8. policies assigned but no claims raised

```
select policyid from policyassignments
except
select pa.policyid from claims c
join policyassignments pa
on c.assignmentid = pa.assignmentid;
```

Case else

1. classify premium

```
select policyname,
case
   when premiumamount < 5000 then 'low'
   when premiumamount between 5000 and 15000 then 'medium'
   else 'high'
end as premium_category from policies;
```

2. Classify policy duration

```
select policyname,
case
```

```
    when durationyear <= 1 then 'short term'
    when durationyear between 2 and 10 then 'medium term'
    else 'long term'
end as duration_category from policies;
```

3. premium eligibility status

```
select policyname,
case
    when premiumamount <= 8000 then 'economy'
    when premiumamount between 8001 and 15000 then 'standard'
    else 'premium'
end as policy_level from policies;
```

Rollup

1. Total premium by policytype & policyName

```
select policyname, policytype, sum(premiumamount) as total_premium
from policies
group by rollup(policyname, policytype)
order by Policyname, PolicyType;
```

2. Total premium by policytype & durationyear

```
select policytype, durationyear, sum(premiumamount) as total_premium from policies
group by rollup(policytype, durationyear)
order by policytype, durationyear;
```

3. Count of policies by policytype & status

```
select policytype, status, count(*) as policy_count from policies
group by rollup(policytype, status)
order by policytype, status;
```

Cube

1. Total premium by policyType, policyName, durationYear

```
select policyname, policytype, durationYear, sum(premiumamount) as total_premium from
policies
group by grouping sets(policyname, policytype, DurationYear)
order by Policyname, PolicyType;
```

2. Total premium by policytype, durationyear

```
select policytype, durationyear, sum(premiumamount) as total_premium from policies
group by cube(policytype, durationyear)
order by policytype, durationyear;
```

3. Policy count by policytype, status

```
select policytype, status, count(*) as policy_count from policies
group by cube(policytype, status)
order by policytype, status;
```

Grouping sets

1. Total premium by policyType, policyName, durationYear

```
select policyname, policytype, durationYear, sum(premiumamount) as total_premium
from policies
group by grouping sets(policyname, policytype, DurationYear)
order by Policyname, PolicyType;
```

2. Total premium by policytype and durationyear

```
select policytype, durationyear, sum(premiumamount) as total_premium from policies
group by grouping sets
((policytype), (durationyear))
order by policytype, durationyear;
```

3. policy count by policyname and status

```
select policyname, status, count(*) as policy_count from policies
group by grouping sets
((policyname),(status))
order by policyname, status;
```