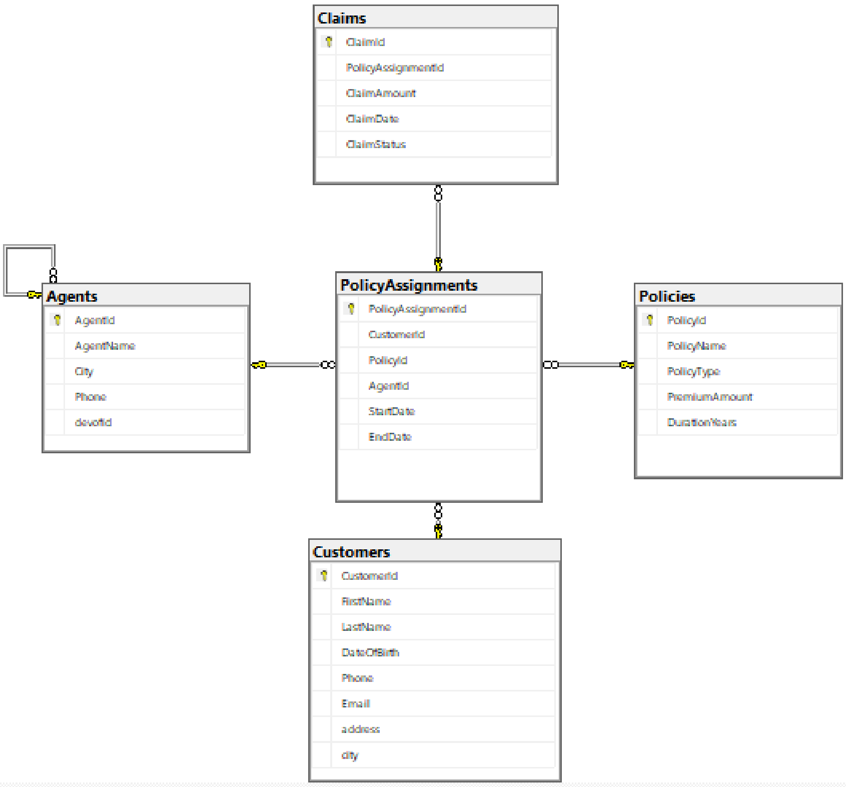
Module 4.4 Practical Project Assignment
BY KATTA SAI RUTVIK

Database creation

Create database insurancedb;

Schema diagram



Create tables

1.Customers

```
create table customers (
    customerid int identity(1,1) primary key,
    firstname varchar(50) not null,
    lastname varchar(50),
    dateofbirth date not null,
    phone varchar(15),
    email varchar(100) unique
);
```

2.Agents

```
create table agents (
    agentid int identity(1,1) primary key,
    agentname varchar(100) not null,
    city varchar(50),
    phone varchar(15),
    );
```

3.Policies

```
create table policies (
    policyid int identity(1,1) primary key,
    policyname varchar(100) not null,
    policytype varchar(20),
    premiumamount decimal(10,2) not null,
    durationyears int not null
);
```

4.PolicyAssignments

```
create table policyassignments (
    policyassignmentid int identity(1,1) primary key,
    customerid int not null,
    policyid int not null,
    agentid int not null,
    startdate date not null,
    enddate date,

     constraint fk_pa_customer
        foreign key (customerid) references customers(customerid),

    constraint fk_pa_policy
        foreign key (policyid) references policies(policyid),

    constraint fk_pa_agent
        foreign key (agentid) references agents(agentid)
);
```

5.claims

```
create table claims (
    claimid int identity(1,1) primary key,
    policyassignmentid int not null,
    claimamount decimal(12,2) not null,
    claimdate date not null,
    claimstatus varchar(20),

    constraint fk_claims_policyassignment
        foreign key (policyassignmentid)
        references policyassignments(policyassignmentid)
);
```

Insert Commands

1.Customers

```
insert into customers (firstname, lastname, dateofbirth, phone, email)
values
('Ravi', 'Kumar', '2003-06-15', '9876543210', 'ravi@gmail.com'),
('Anita', 'Sharma', '2005-09-20', '9876543211', 'anita@gmail.com'),
('Suresh', 'Reddy', '2001-02-10', '9876543212', 'suresh@gmail.com'),
('Priya', 'Mehta', '2010-11-18', '9876543213', 'priya@gmail.com'),
('Amit', 'Verma', '2008-01-05', '9876543214', 'amit@gmail.com'),
('Rahul', 'Singh', '2004-04-12', '9876543215', 'rahul@gmail.com'),
('Neha', 'Agarwal', '2006-07-19', '9876543216', 'neha@gmail.com'),
('Vikas', 'Jain', '2002-02-28', '9876543217', 'vikas@gmail.com'),
('Pooja', 'Malhotra', '2009-09-09', '9876543218', 'pooja@gmail.com'),
('Arjun', 'Mehta', '2001-12-15', '9876543219', 'arjun@gmail.com'),
('Sneha', 'Iyer', '2007-05-21', '9876543220', 'sneha@gmail.com'),
('Karthik', 'Rao', '2003-03-30', '9876543221', 'karthik@gmail.com');
```

2.Agents

```
insert into agents (agentname, city, phone)
values
('Rajesh Rao', 'hyderabad', '9123456780'),
('Sunita Devi', 'bangalore', '9123456781'),
('Mahesh Patel', 'ahmedabad', '9123456782'),
('Anil Kumar', 'chennai', '9123456783'),
('Kavya Nair', 'kochi', '9123456784'),
('Ramesh Gupta', 'delhi', '9123456785');
```

3. Policies

```
insert into policies (policyname, policytype, premiumamount, durationyears)
values
('Health Secure', 'health', 12000, 1),
('Life Protect', 'life', 25000, 10),
('Motor Safe', 'motor', 8000, 1),
('Health Plus', 'health', 18000, 2),
('Life Smart', 'life', 15000, 1),
('Health Basic', 'health', 9000, 1);
```

4.PolicyAssignments

```
insert into policyassignments (customerid, policyid, agentid, startdate, enddate)
values
(1, 1, 1, '2024-01-01', '2025-01-01'),
(2, 2, 2, '2023-06-15', '2033-06-15'),
(3, 3, 3, '2024-03-10', '2025-03-10'),
(4, 4, 4, '2022-11-20', '2024-11-20'),
(5, 5, 5, '2024-05-05', null),
(6, 6, 6, '2023-01-01', '2024-01-01');
```

5.Claims

```
insert into claims (policyassignmentid, claimamount, claimdate, claimstatus)
values
(1, 45000, '2024-06-10', 'approved'),
(1, 12000, '2024-08-15', 'rejected'),
(2, 60000, '2024-01-20', 'approved'),
(3, 30000, '2024-04-05', 'rejected'),
(4, 15000, '2023-12-12', 'approved'),
(5, 75000, '2024-09-01', 'approved');
```

Queries

Basic select

1. Display all customers

```
select * from customers;
```

2. Display agent name and city of all agents

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

3. Display policy name and premium amount

```
select policyname, premiumamount from policies;
```

4. Display customerid, policyid, startdate from policy assignments

```
select customerid, policyid, startdate from policyassignments;
```

5. Display claim amount and claim status

```
select claimamount, claimstatus from claims;
```

Where Clause

1. Display customers born after 2005

```
select * from customers where dateofbirth > '2005-01-01';
```

2. Display policies with premium amount greater than 10000

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

3. Display all health type policies

```
select * from policies where policytype = 'health';
```

4. Display claims with rejected status

```
select * from claims where claimstatus = 'rejected';
```

5. Display agents who belong to city hyderabad

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

Order and Distinct

1. Display all customers ordered by firstname
select * from customers order by firstname;

2. Display policies ordered by premium amount in descending order
select * from policies order by premiumamount desc;

3. Display agents ordered by city
select * from agents order by city;

4. Display claims ordered by claimdate (latest first)
select * from claims order by claimdate desc;

5. Display distinct cities from agents table
select distinct city from agents

Update

1. Increase premium amount by 10% for all health policies
update policies set premiumamount = premiumamount * 1.10 where policytype = 'health';

2. Update city to 'hyderabad' for agent named Rajesh Rao
update agents set city = 'hyderabad' where agentname = 'Rajesh Rao';

3. Update enddate for a policy assignment of customerid 1
update policyassignments set enddate = '2026-01-01' where customerid = 1;

4. Update claim status to approved for claimid 1
update claims set claimstatus = 'approved' where claimid = 1;

5. Update phone number for customer with firstname Anita
update customers set phone = '9999999999' where firstname = 'Anita';

Aggregate function

1. Display total number of customers
select count(*) from customers;

2. Display total premium amount of all policies
select sum(premiumamount) from policies;

3. Display average premium amount of policies

```
select avg(premiumamount) from policies;
```

4. Display highest and lowest claim amount

```
select max(claimamount), min(claimamount) from claims;
```

5. Display total number of policies assigned

```
select count(*) from policyassignments;
```

Date functions

1. Display current date and time

```
select getdate();
```

2. Display year from customer date of birth

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

3. Display customers born in the year 2005

```
select * from customers where year(dateofbirth) = 2005;
```

4. Display difference in days between claim date and today

```
select claimid, datediff(day, claimdate, getdate()) from claims;
```

5. Display month name of policy start date

```
select policyassignmentid, datename(month, startdate) from policyassignments;
```

Joins

1. Display customer names with their policy names (normal join)

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

2. Display customer names with their policy names (inner join)

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

3. Display all customers with or without policies (left join)

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

4. Display all agents with or without policy assignments (right join)

```
select a.agentname, p.policyname
from policyassignments pa
right join agents a on pa.agentid = a.agentid
left join policies p on pa.policyid = p.policyid;
```

5. Display agents with their development officers (self join)

```
select child.agentname, parent.agentname as development_officer
from agents child
left join agents parent on child.devofid = parent.agentid;
```

Alter

1. Increase size of firstname column in customers table

```
alter table customers alter column firstname varchar(100);
```

2. Increase size of policyname column in policies table

```
alter table policies alter column policyname varchar(150);
```

Groupby

1. Display number of customers in each city

```
select city, count(*)
from customers
group by city;
```

2. Display number of policies for each policy type

```
select policytype, count(*)
from policies
group by policytype;
```

3. Display total claim amount per customer

```
select c.firstname, sum(cl.claimamount)
from customers c
join policyassignments pa on c.customerid = pa.customerid
join claims cl on pa.policyassignmentid = cl.policyassignmentid
group by c.firstname;
```

4. Display number of policies assigned to each agent

```
select a.agentname, count(pa.policyassignmentid)
from agents a
join policyassignments pa on a.agentid = pa.agentid
group by a.agentname;
```

5. Display total premium amount for each policy type

```
select policytype, sum(premiumamount)
from policies
group by policytype;
```

Having

1. Display policy types having total premium amount greater than 30000

```
select policytype, sum(premiumamount)
from policies
group by policytype
having sum(premiumamount) > 30000;
```

2. Display claim statuses having more than one claim

```
select claimstatus, count()
from claims
group by claimstatus
having count() > 1;
```

3. Display policyassignmentids having total claim amount greater than 50000

```
select policyassignmentid, sum(claimamount)
from claims
group by policyassignmentid
having sum(claimamount) > 50000;
```

4. Display policy types having average premium amount greater than 15000

```
select policytype, avg(premiumamount)
from policies
group by policytype
having avg(premiumamount) > 15000;
```

5. Display policyassignmentids having more than one claim

```
select policyassignmentid, count()
from
group by policyassignmentid
having count() > 1;
```

Subqueries

1. Display customers whose date of birth is the earliest

```
select *
from customers
where dateofbirth = (select min(dateofbirth) from customers);
```

2. Display policy with the highest premium amount

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

3. Display claims having claim amount greater than average claim amount

```
select *
from claims
where claimamount > (select avg(claimamount) from claims);
```

4. Display customers whose total claim amount is greater than the overall average claim amount

```
select *
from customers
where customerid in (
select pa.customerid
from policyassignments pa
where pa.policyassignmentid in (
select policyassignmentid
from claims
group by policyassignmentid
having sum(claimamount) > (select avg(claimamount) from claims)
)
);
```

5. Display policies that are assigned to more than one customer

```
select *
from policies
where policyid in (
select policyid
from policyassignments
group by policyid
having count(*) > 1
);
```

Case

1. Display policies with premium category (low / medium / high)

```
select policyname,
case
when premiumamount < 10000 then 'low'
when premiumamount between 10000 and 20000 then 'medium'
else 'high'
end as premium_category
from policies;
```

2. Display claims with readable claim status

```
select claimid,
case
when claimstatus = 'approved' then 'claim accepted'
when claimstatus = 'rejected' then 'claim rejected'
else 'under review'
end as claim_result
from claims;
```

3. Display customers with age group based on date of birth

```
select firstname,
case
when year(dateofbirth) <= 2003 then 'above 21'
when year(dateofbirth) between 2004 and 2008 then 'teen'
else 'minor'
end as age_group
from customers;
```

4. Display policy assignments as active or expired

```
select policyassignmentid,
case
when enddate is null or enddate >= cast(getdate() as date) then 'active'
else 'expired'
end as policy_status
from policyassignments;
```

5. Display agents with workload category based on number of policies assigned

```
select agentid,
case
when count() = 1 then 'low load'
when count() between 2 and 3 then 'medium load'
else 'high load'
end as workload
from policyassignments
group by agentid;
```

Rollup

1. Display total premium amount by policy type with grand total

```
select policytype, sum(premiumamount)
from policies
group by rollup(policytype);
```

2. Display total claim amount by claim status with overall total

```
select claimstatus, sum(claimamount)
from claims
group by rollup(claimstatus);
```

3. Display number of policies assigned per agent with grand total

```
select agentid, count(policyassignmentid)
from policyassignments
group by rollup(agentid);
```

4. Display total claim amount per policyassignment with grand total

```
select policyassignmentid, sum(claimamount)
from claims
group by rollup(policyassignmentid);
```

5. Display number of customers per city with overall total

```
select city, count(*)
from customers
group by rollup(city);
```

Cube

1. Display total premium amount by policy type with all subtotals and grand total

```
select policytype, sum(premiumamount)
from policies
group by cube(policytype);
```

2. Display total claim amount by claim status with all combinations

```
select claimstatus, sum(claimamount)
from claims
group by cube(claimstatus);
```

3. Display number of policies assigned by agent with subtotals and grand total

```
select agentid, count(policyassignmentid)
from policyassignments
group by cube(agentid);
```

4. Display total claim amount by policyassignmentid with all totals

```
select policyassignmentid, sum(claimamount)
from claims
group by cube(policyassignmentid);
```

5. Display number of customers by city with all subtotals and grand total

```
select city, count(*)
from customers
group by cube(city);
```

Grouping set

1. Display total premium amount by policy type and grand total

```
select policytype, sum(premiumamount)
from policies
group by grouping sets ((policytype), ());
```

2. Display total claim amount by claim status and overall total

```
select claimstatus, sum(claimamount)
from claims
group by grouping sets ((claimstatus), ());
```

3. Display number of policies per agent and grand total

```
select agentid, count(policyassignmentid)
```

```
from policyassignments
group by grouping sets ((agentid), ());
```

Merge

Merge new customer data into customers table (insert if not exists, update if exists)

```
 merge customers as target
using (
    select 1 as customerid, 'Ravi' as firstname, 'Kumar' as lastname
    union all
    select 13, 'New', 'Customer'
) as source
on target.customerid = source.customerid

when matched then
    update set
        target.firstname = source.firstname,
        target.lastname = source.lastname

when not matched then
    insert (customerid, firstname, lastname)
    values (source.customerid, source.firstname, source.lastname);
```

View

1.Display customer name and policy name using a view

Create a view to display customer name and policy name

```
create view vw_customer_policy
as
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. Create a view to display claim amount and claim status

```
create view vw_claim_details
as
select claimamount, claimstatus
from claims;
```

Set Operations

1. **Display all customer first names and agent names using UNION**
   select firstname from customers
   union
   select agentname from agents;
2. **Display all customer first names and agent names including duplicates using UNION ALL**
   select firstname from customers
   union all
   select agentname from agents;
3. **Display customer first names that are also agent names using INTERSECT**
   select firstname from customers
   intersect
   select agentname from agents;
4. **Display customer first names that are not agent names using EXCEPT**
   select firstname from customers
   except
   select agentname from agents;
5. **Display policy ids that are assigned and also have claims using INTERSECT**
   select policyid from policyassignments
   intersectselect pa.policyid
   from policyassignments pa
   join claims cl on pa.policyassignmentid = cl.policyassignmentid;

String Operations

1. Display customer first names in uppercase
select upper(firstname) from customers;

2. Display agent names in lowercase
select lower(agentname) from agents;

3. Display length of each policy name
select policyname, len(policyname) from policies;

4. Display first 3 characters of customer first names
select substring(firstname, 1, 3) from customers;

5. Display customer first name and last name combined as full name
select concat(firstname, ' ', lastname) from customers

Clustered Index

1.Create a clustered index on customerid column of customers table

```
create clustered index idx_customers_customerid
on customers(customerid);
```

2 .Create a clustered index on policyassignmentid column of policyassignments table

```
create clustered index idx_policyassignments_id
on policyassignments(policyassignmentid);
```