DAY-2 ASSIGNMENT

InsuranceDB:

1. CREATE DATABASE InsuranceDB;
   USE InsuranceDB;
2. Create table commands for all the tables with constraints, relationships etc.

```
CREATE TABLE Customers(
CustomerID INT PRIMARY KEY,
FirstName VARCHAR(50),
LastName VARCHAR(50),
DateOfBirth DATE,
Phone VARCHAR(20),
Email VARCHAR(50)
);

CREATE TABLE Policies(
PolicyID INT PRIMARY KEY,
PolicyName VARCHAR(50),
PolicyType VARCHAR(20),
PremiumAmount DECIMAL(10,2),
DurationYears VARCHAR(20)
);



CREATE TABLE Agents(
AgentID INT PRIMARY KEY,
AgentName VARCHAR(50),
Phone VARCHAR(20),
City VARCHAR(20)
);

CREATE TABLE PolicyAssignments(
AssignmentID INT PRIMARY KEY,
CustomerID INT,
PolicyID INT,
AgentID INT,
StartDate DATE,
EndDate DATE,
Constraint fk_customer_pa
FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
Constraint fk_policy_pa
FOREIGN KEY (PolicyID) REFERENCES Policies(PolicyID),
Constraint fk_agent_pa
FOREIGN KEY (AgentID) REFERENCES Agents(AgentID)
);

CREATE TABLE Claims(
ClaimID INT PRIMARY KEY,
AssignmentID INT,
ClaimDate DATE,
ClaimAmount DECIMAL(10,2),
ClaimStatus VARCHAR(10),
Constraint fk_pa_Claims
FOREIGN KEY (AssignmentID) REFERENCES PolicyAssignments(AssignmentID)
);
```

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

```sql
INSERT INTO Customers (CustomerID, FirstName, LastName, DateOfBirth,
Phone, Email)
VALUES
(1, 'Ravi', 'Kumar', '1990-05-12', '9876543210',
'ravi.kumar@example.com'),
(2, 'Priya', 'Sharma', '1988-11-22', '9876543211',
'priya.sharma@example.com'),
(3, 'Amit', 'Verma', '1995-03-15', '9876543212',
'amit.verma@example.com'),
(4, 'Sneha', 'Reddy', '1992-07-09', '9876543213',
'sneha.reddy@example.com');

INSERT INTO Customers (CustomerID, FirstName, LastName, DateOfBirth,
Phone, Email)
VALUES
(5, 'Chaitanya', 'Talasila', '2005-04-15', '9876543210',
'chaitanya@example.com');

INSERT INTO Policies (PolicyID, PolicyName, PolicyType, PremiumAmount,
DurationYears)
VALUES
(101, 'Health Secure', 'Health', 12000.00, '3 Years'),
(102, 'Life Shield', 'Life', 25000.00, '10 Years'),
(103, 'Motor Protect', 'Vehicle', 8000.00, '1 Year'),
(104, 'Senior Care', 'Health', 18000.00, '5 Years');

INSERT INTO Agents (AgentID, AgentName, Phone, City)
VALUES
(201, 'Anil Mehta', '9000011111', 'Hyderabad'),
(202, 'Kavya Rao', '9000011112', 'Mumbai'),
(203, 'Rohit Singh', '9000011113', 'Delhi');

INSERT INTO Agents (AgentID, AgentName, Phone, City)
VALUES
(204, 'Harshadh', '9000041111', 'Calicut');

INSERT INTO PolicyAssignments
(AssignmentID, CustomerID, PolicyID, AgentID, StartDate, EndDate)
VALUES
(301, 1, 101, 201, '2023-01-01', '2026-01-01'),
(302, 1, 102, 202, '2020-05-10', '2030-05-10'),
(303, 2, 103, 201, '2024-01-01', '2025-01-01'),
(304, 3, 104, 203, '2022-06-15', '2027-06-15'),
(305, 4, 101, 202, '2021-02-02', '2024-02-02');
SELECT * FROM PolicyAssignments;

INSERT INTO Claims
(ClaimID, AssignmentID, ClaimDate, ClaimAmount, ClaimStatus)
VALUES
(401, 301, '2024-03-12', 5000.00, 'Approved'),
(402, 303, '2024-04-15', 12000.00, 'Pending'),
(403, 305, '2023-10-10', 8000.00, 'Rejected'),
(404, 304, '2024-01-25', 15000.00, 'Approved'),
(405, 302, '2024-06-30', 30000.00, 'Pending');
```

4. Select commands

```
-- 1. View all records Customers table.
SELECT * FROM Customers;
```

	CustomerID	FirstName	LastName	DateOfBirth	Phone	Email
1	1	Ravi	Kumar	1990-05-12	9876543210	ravi.kumar@example.com
2	2	Priya	Sharma	1988-11-22	9876543211	priya.sharma@example.com
3	3	Amit	Verma	1995-03-15	9876543212	amit.verma@example.com
4	4	Sneha	Reddy	1992-07-09	9876543213	sneha.reddy@example.com
5	5	Chaitanya	Talasila	2005-04-15	9876543210	chaitanya@example.com

```
-- 2. View all records of PolicyAssignment table with CustomerId, PolicyId,
StartDate and EndDate columns only.
SELECT CustomerId, PolicyId, StartDate, EndDate FROM PolicyAssignments;
```

	CustomerId	PolicyId	StartDate	EndDate
1	1	101	2023-01-01	2026-01-01
2	1	102	2020-05-10	2030-05-10
3	2	103	2024-01-01	2025-01-01
4	3	104	2022-06-15	2027-06-15
5	4	101	2021-02-02	2024-02-02

```
-- 3. Display all policies of Health type.
SELECT PolicyName FROM Policies;
```

	PolicyName
1	Health Secure
2	Life Shield
3	Motor Protect
4	Senior Care

```
-- 4. Display policies having premium amount more than 10000 and DurationYears is
1.
SELECT PolicyName, PremiumAmount, DurationYears FROM Policies WHERE
PremiumAmount>1000 and DurationYears ='1 Year';
```

	PolicyName	PremiumAmount	DurationYears
1	Motor Protect	8800.00	1 Year

```
-- 5. Display unique city names from where agents belong to.
SELECT DISTINCT(city) FROM Agents;
```

	city
1	Calicut
2	Delhi
3	Hyderabad
4	Mumbai

```sql
-- 6. List policies of type Life, Health, Motor using OR clause.
SELECT PolicyName FROM Policies WHERE PolicyType='Life' OR PolicyType='Health' OR
PolicyType='Vehicle';
```

	PolicyName
1	Health Secure
2	Life Shield
3	Motor Protect
4	Senior Care

```sql
-- 7. List policies of type Life, Health, Motor using IN operator.
SELECT PolicyName FROM Policies WHERE PolicyType IN ('Life', 'Health',
'Vehicle');
```

	PolicyName
1	Health Secure
2	Life Shield
3	Motor Protect
4	Senior Care

```sql
-- 8. Display list of customers born after January 1st, 2001 and before December
31st, 2020 using >= and <= operators.
SELECT CustomerID, FirstName, LastName,DateOfBirth  FROM Customers WHERE
DateOfBirth>='2001-01-01' AND DateOfBirth<='2020-12-31';
```

	CustomerID	FirstName	LastName	DateOfBirth
1	5	Chaitanya	Talasila	2005-04-15

```sql
-- 9. Display list of customers born after January 1st, 2001 and before December
31st, 2020 using BETWEEN operator.
SELECT CustomerID, FirstName, LastName,DateOfBirth  FROM Customers WHERE
DateOfBirth BETWEEN '2001-01-01' AND '2020-12-31';
```

	CustomerID	FirstName	LastName	DateOfBirth
1	5	Chaitanya	Talasila	2005-04-15

```sql
-- 10. Display claims data where claim status is Rejected.
SELECT * FROM Claims WHERE ClaimStatus='Rejected';
```

	ClaimID	AssignmentID	ClaimDate	ClaimAmount	ClaimStatus
1	403	305	2023-10-10	8000.00	Rejected

```sql
-- 11. Display records of Agents who stay in a city whose second letter is 'a'.
SELECT * FROM Agents WHERE city LIKE '_a%';
```

	AgentID	AgentName	Phone	City
1	204	Harshadh	9000041111	Calicut

```sql
-- 12. Display highest and lowest ClaimAmount from Claims table.
SELECT MIN(ClaimAmount) as Lowest_claim, MAX(ClaimAmount) AS Highest_claim FROM
Claims;
```

	Lowest_claim	Highest_claim
1	5000.00	30000.00

```sql
-- 13. Display latest claim record.
SELECT * FROM Claims ORDER BY ClaimDate DESC OFFSET 0 ROWS FETCH NEXT 1 ROWS
ONLY;
```

	ClaimID	AssignmentID	ClaimDate	ClaimAmount	ClaimStatus
1	405	302	2024-06-30	30000.00	Pending

```sql
-- 14. Increase premium amount by 10% for all health insurance policies.
UPDATE Policies SET PremiumAmount=1.1*PremiumAmount
SELECT * FROM Policies;
```

	PolicyID	PolicyName	PolicyType	PremiumAmount	DurationYears
1	101	Health Secure	Health	13200.00	3 Years
2	102	Life Shield	Life	27500.00	10 Years
3	103	Motor Protect	Vehicle	8800.00	1 Year
4	104	Senior Care	Health	19800.00	5 Years

```sql
-- 15. Delete the record of PolicyAssignments whose EndDate is before today's
date.
DELETE FROM Claims
WHERE AssignmentID IN (
    SELECT AssignmentID
    FROM PolicyAssignments
    WHERE EndDate < GETDATE()
);
DELETE FROM PolicyAssignments WHERE EndDate<GetDate();
SELECT * FROM PolicyAssignments;
```

	AssignmentID	CustomerID	PolicyID	AgentID	StartDate	EndDate
1	301	1	101	201	2023-01-01	2026-01-01
2	302	1	102	202	2020-05-10	2030-05-10
3	304	3	104	203	2022-06-15	2027-06-15

```sql
-- 16. Display number of claims rejected.
SELECT COUNT(*) FROM Claims WHERE ClaimStatus='Rejected';
```

	(No column name)
1	1

```sql
-- 17. Display PolicyId, PolicyName, PremiumAmount along with computed fields not
in table -> 6% LocalTaxes, PremiumAmountWithTax and MonthlyPremiumAmount
considering PremiumAmount is Annual.
SELECT PolicyId, PolicyName, PremiumAmount, (0.06 *PremiumAmount) AS LocalTaxes,
ROUND(PremiumAmount+(0.06 *PremiumAmount),2) AS PremiumAmountWithTax,
ROUND((PremiumAmount + (PremiumAmount * 0.06)) / 12.0, 2) as MonthlyPremiumAmount
FROM Policies;
```

	PolicyId	PolicyName	PremiumAmount	LocalTaxes	PremiumAmountWithTax	MonthlyPremiumAmount
1	101	Health Secure	13200.00	792.0000	13992.0000	1166.00000000
2	102	Life Shield	27500.00	1650.0000	29150.0000	2429.17000000
3	103	Motor Protect	8800.00	528.0000	9328.0000	777.33000000
4	104	Senior Care	19800.00	1188.0000	20988.0000	1749.00000000

```sql
-- 18. Write a command to add Address and City Columns in the Customers table.
ALTER TABLE Customers
ADD Address VARCHAR(50), City VARCHAR(20)
SELECT * FROM Customers
```

	CustomerID	FirstName	LastName	DateOfBirth	Phone	Email	Address	City
1	1	Ravi	Kumar	1990-05-12	9876543210	ravi.kumar@example.com	NULL	NULL
2	2	Priya	Sharma	1988-11-22	9876543211	priya.sharma@example.com	NULL	NULL
3	3	Amit	Verma	1995-03-15	9876543212	amit.verma@example.com	NULL	NULL
4	4	Sneha	Reddy	1992-07-09	9876543213	sneha.reddy@example.com	NULL	NULL
5	5	Chaitanya	Talasila	2005-04-15	9876543210	chaitanya@example.com	NULL	NULL

```sql
-- 19. Write a command to add a new column named DevOfId (DevelopmentOfficerId)
in an existing Agents table.
ALTER TABLE Agents
ADD DevOfId INT;
```

	AgentID	AgentName	Phone	City	DevOfId
1	201	Anil Mehta	9000011111	Hyderabad	NULL
2	202	Kavya Rao	9000011112	Mumbai	NULL
3	203	Rohit Singh	9000011113	Delhi	NULL
4	204	Harshadh	9000041111	Calicut	NULL

```sql
-- 20. Write command to make the above DevOfId as a recursive foreign key to
AgentId as Parent.
ALTER TABLE Agents
ADD CONSTRAINT fk_agent_devof
FOREIGN KEY (DevOfId) REFERENCES Agents(AgentID);
```

5. Queries using Joins, Group By, Having etc.

```sql
-- 1. List all Policies for a CustomerId 4.
SELECT a.CustomerID,c.PolicyName
FROM Customers a JOIN PolicyAssignments b
ON a.CustomerID=b.CustomerID
JOIN Policies c
ON c.PolicyID=B.PolicyID
WHERE a.customerid=4;
```

	CustomerID	PolicyName
1	4	Health Secure

```sql
-- 2. View all customers with their policies.
SELECT a.CustomerID,c.PolicyName
FROM Customers a JOIN PolicyAssignments b
ON a.CustomerID=b.CustomerID
JOIN Policies c
ON c.PolicyID=B.PolicyID;
```

	CustomerID	PolicyName
1	1	Health Secure
2	1	Life Shield
3	2	Motor Protect
4	3	Senior Care
5	4	Health Secure

```sql
-- 3. View claims with customer name.
SELECT a.FirstName,a.LastName,c.ClaimID, c.ClaimDate, c.claimamount,c.claimstatus
FROM Customers a JOIN PolicyAssignments b
ON a.CustomerID=b.CustomerID
JOIN Claims c
ON b.AssignmentID=c.AssignmentID;
```

	FirstName	LastName	ClaimID	ClaimDate	claimamount	claimstatus
1	Ravi	Kumar	401	2024-03-12	5000.00	Approved
2	Ravi	Kumar	405	2024-06-30	30000.00	Pending
3	Priya	Sharma	402	2024-04-15	12000.00	Pending
4	Amit	Verma	404	2024-01-25	15000.00	Approved
5	Sneha	Reddy	403	2023-10-10	8000.00	Rejected

```sql
-- 4. Display FirstName, PolicyName, AgentName, StartDate and EndDate from their
respective tables.
SELECT a.FirstName,c.PolicyName,d.AgentName,b.StartDate,b.EndDate
FROM Customers a JOIN PolicyAssignments b
ON a.CustomerID=b.CustomerID
JOIN Policies c
ON c.PolicyID=B.PolicyID
JOIN Agents d
ON d.AgentID=b.AgentID;
```

	FirstName	PolicyName	AgentName	StartDate	EndDate
1	Ravi	Health Secure	Anil Mehta	2023-01-01	2026-01-01
2	Ravi	Life Shield	Kavya Rao	2020-05-10	2030-05-10
3	Priya	Motor Protect	Anil Mehta	2024-01-01	2025-01-01
4	Amit	Senior Care	Rohit Singh	2022-06-15	2027-06-15
5	Sneha	Health Secure	Kavya Rao	2021-02-02	2024-02-02

```sql
-- 5. Display claims report with FirstName, PolicyName, ClaimAmount, ClaimStatus, and ClaimDate from their respective tables.
SELECT a.FirstName,c.PolicyName,d.ClaimAmount,d.ClaimStatus,d.ClaimDate
FROM Customers a JOIN PolicyAssignments b
ON a.CustomerID=b.CustomerID
JOIN Policies c
ON c.PolicyID=B.PolicyID
JOIN Claims d
ON b.AssignmentID=d.AssignmentID;
```

	FirstName	PolicyName	ClaimAmount	ClaimStatus	ClaimDate
1	Ravi	Health Secure	5000.00	Approved	2024-03-12
2	Ravi	Life Shield	30000.00	Pending	2024-06-30
3	Priya	Motor Protect	12000.00	Pending	2024-04-15
4	Amit	Senior Care	15000.00	Approved	2024-01-25
5	Sneha	Health Secure	8000.00	Rejected	2023-10-10

```sql
-- 6. Display records of Customers with or without Policies.
SELECT a.CustomerID,a.FirstName,a.LastName,b.AssignmentID,b.PolicyID
FROM Customers a LEFT JOIN PolicyAssignments b
ON a.CustomerID=b.CustomerID;
```

	CustomerID	FirstName	LastName	AssignmentID	PolicyID
1	1	Ravi	Kumar	301	101
2	1	Ravi	Kumar	302	102
3	2	Priya	Sharma	303	103
4	3	Amit	Verma	304	104
5	4	Sneha	Reddy	305	101
6	5	Chaitanya	Talasila	NULL	NULL

```sql
-- 7. Display all Customers with NO Claims.
SELECT a.CustomerID,a.FirstName,a.LastName
FROM Customers a LEFT JOIN PolicyAssignments b
ON a.CustomerID=b.CustomerID
WHERE b.AssignmentID IS NULL;
```

	CustomerID	FirstName	LastName
1	5	Chaitanya	Talasila

```sql
-- 8. Show CustomerName with Total Claim Amount per Customer.
SELECT a.customerid,a.FirstName,a.lastname,SUM(c.claimamount)
FROM Customers a JOIN PolicyAssignments b
ON a.CustomerID=b.CustomerID
JOIN Claims c
ON b.AssignmentID=c.AssignmentID
GROUP BY a.customerid,a.FirstName,a.lastname;
```

	customerid	FirstName	lastname	(No column name)
1	1	Ravi	Kumar	35000.00
2	2	Priya	Sharma	12000.00
3	3	Amit	Verma	15000.00
4	4	Sneha	Reddy	8000.00

```sql
-- 9. Show names and total claim amount of Customers With Claim Amount > 50000
(Use HAVING Clause).
SELECT a.customerid,a.FirstName,a.lastname,SUM(c.claimamount)
FROM Customers a JOIN PolicyAssignments b
ON a.CustomerID=b.CustomerID
JOIN Claims c
ON b.AssignmentID=c.AssignmentID
GROUP BY a.customerid,a.FirstName,a.lastname
HAVING SUM(c.claimamount)>10000;
```

	customerid	FirstName	lastname	(No column name)
1	1	Ravi	Kumar	35000.00
2	2	Priya	Sharma	12000.00
3	3	Amit	Verma	15000.00

```sql
-- 10. Display list with Agent Wise Policy Count.
SELECT a.agentid,COUNT(*)
FROM agents a join policyassignments b
on a.agentid=b.agentid
GROUP BY a.agentid;
```

	agentid	(No column name)
1	201	2
2	202	2
3	203	1