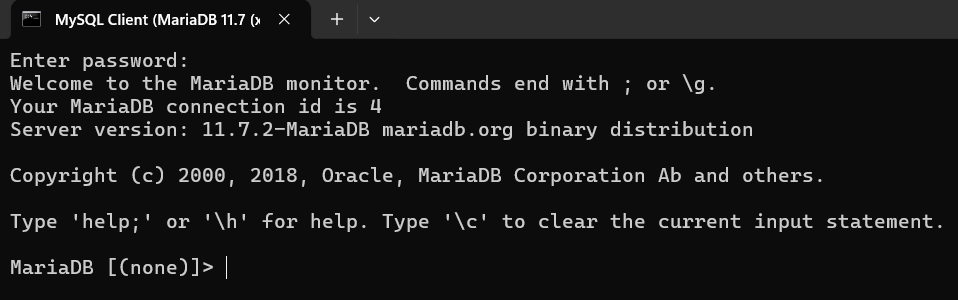
**Homework 4: SQL 2**

**Step 1: Start MariaDB (0 pts)**



**Step 2: Create New Relations and Insert Data (0 pts)**

**Create New Relations**

GoesOn(ssn: integer, id: integer)

|  |  |
| --- | --- |
| SQL Statements | Output |
| CREATE TABLE GoesOn(  ssn INTEGER,  id INTEGER,  PRIMARY KEY(ssn, id)  );  SHOW TABLES;  DESCRIBE GoesOn; |  |

Leg(Trip id: integer, startLocation: String, endLocation: String, startDate: Date, endDate: Date)

|  |  |
| --- | --- |
| SQL Statements | Output |
| CREATE TABLE Leg(  Trip\_id INTEGER,  startLocation VARCHAR(50),  endLocation VARCHAR(50),  startDate DATE,  endDate DATE,  PRIMARY KEY(Trip\_id, startLocation, endLocation)  );  SHOW TABLES;  DESCRIBE Leg; |  |

Owns(ssn: integer, passport\_number: Integer, country: String)

|  |  |
| --- | --- |
| SQL Statements | Output |
| CREATE TABLE Owns(  ssn INTEGER,  passport\_number INTEGER,  country VARCHAR(50),  PRIMARY KEY(passport\_number, country)  );  SHOW TABLES;  DESCRIBE Owns; |  |

Passport(passport\_number: integer, country: String, expirationDate: Date, holderName: String)

|  |  |
| --- | --- |
| SQL Statements | Output |
| CREATE TABLE Passport(  Passport\_number INTEGER,  country VARCHAR(50),  expirationDate DATE,  holderName VARCHAR(50),  PRIMARY KEY(passport\_number, country)  );  SHOW TABLES;  DESCRIBE Passport; |  |

**Insert Data**

|  |  |
| --- | --- |
| SQL Statements | Output |
| INSERT INTO GoesOn(ssn, id)  VALUES  (101, 201),  (101, 208),  (102, 202),  (102, 205),  (103, 203),  (103, 206),  (106, 206),  (106, 201),  (107, 207),  (108, 208);  SELECT \* FROM GoesOn; |  |

|  |  |
| --- | --- |
| SQL Statements | Output |
| INSERT INTO Leg(Trip\_id, startLocation, endLocation, startDate, endDate)  VALUES  (201, 'New York', 'London', '2025-07-10', '2025-07-12'),  (201, 'London', 'Paris', '2025-07-13', '2025-07-20'),  (202, 'Tokyo', 'Seoul', '2025-08-01', '2025-08-05'),  (202, 'Seoul', 'Sydney', '2025-08-06', '2025-08-15'),  (203, 'London', 'Berlin', '2025-09-05', '2025-09-08'),  (203, 'Berlin', 'Rome', '2025-09-09', '2025-09-15'),  (204, 'Miami', 'Atlanta', '2025-06-15', '2025-06-17'),  (204, 'Atlanta', 'New York', '2025-06-18', '2025-06-20'),  (205, 'San Francisco', 'Frankfurt', '2025-10-01', '2025-10-06'),  (205, 'Frankfurt', 'Berlin', '2025-10-07', '2025-10-10'),  (206, 'Chicago', 'Denver', '2025-09-10', '2025-09-11'),  (206, 'Denver', 'Los Angeles', '2025-09-12', '2025-09-12'),  (207, 'Boston', 'Dubai', '2025-07-05', '2025-07-10'),  (207, 'Dubai', 'Singapore', '2025-07-11', '2025-07-18'),  (208, 'New York', 'Istanbul', '2025-09-10', '2025-09-12'),  (208, 'Istanbul', 'Dubai', '2025-09-13', '2025-09-15');  SELECT \* FROM Leg; |  |

|  |  |
| --- | --- |
| SQL Statements | Output |
| INSERT INTO Owns(ssn, passport\_number, country)  VALUES  (101, 5001, 'USA'),  (102, 5002, 'Canada'),  (103, 5003, 'UK'),  (104, 5004, 'Germany'),  (105, 5005, 'France'),  (106, 5006, 'Australia'),  (107, 5007, 'Japan'),  (108, 5008, 'India');  SELECT \* FROM Owns; |  |

|  |  |
| --- | --- |
| SQL Statements | Output |
| INSERT INTO Passport(passport\_number, country, expirationDate, holderName)  VALUES  (5001, 'USA', '2030-01-01', 'David Harris'),  (5002, 'Canada', '2027-06-15', 'Sarah Connor'),  (5003, 'UK', '2026-12-30', 'Mike Johnson'),  (5004, 'Germany', '2028-05-10', 'Laura White'),  (5005, 'France', '2029-11-20', 'James Miller'),  (5006, 'Australia', '2027-03-25', 'Emma Watson'),  (5007, 'Japan', '2031-09-10', 'Chris Evans'),  (5008, 'India', '2025-07-05', 'Sophia Brown');  SELECT \* FROM Passport; |  |

**Step 3: SQL Query Exercises (100 pts)**

1. [10 Points] Find the name(s) of the travel agent(s) through whom the traveler with the date of birth ’1985-06-12’ has booked trips. The resulting relation must not include duplicated records.

|  |  |
| --- | --- |
| SQL Statements | Output |
| SELECT DISTINCT agent  FROM Booking A, Traveler T  WHERE A.traveler\_ssn = T.ssn  AND T.dob = '1985-06-12'; |  |

1. [10 Points] List the date of birth of travelers whose passport expires before January 1, 2027. Use a subquery in the WHERE clause to solve this question.

|  |  |
| --- | --- |
| SQL Statements | Output |
| SELECT dob  FROM Traveler T  WHERE T.ssn IN (  SELECT O.ssn  FROM Owns O  JOIN Passport P ON O.passport\_number = P.passport\_number  WHERE P.expirationDate < '2027-01-01'  ); |  |

1. [10 Points] Retrieve the names of travel agents who have more experience than any agent whose name starts with ’John’.

|  |  |
| --- | --- |
| SQL Statements | Output |
| SELECT name  FROM TravelAgent A1  WHERE (  A1.years\_experience > ANY (  SELECT years\_experience  FROM TravelAgent A2  WHERE A2.name LIKE 'John%')  ); |  |

1. [10 Points] Find the SSNs of travelers who both own a passport and have gone on multiple trips. Use INTERSECT to solve this question.

|  |  |
| --- | --- |
| SQL Statements | Output |
| SELECT T.ssn  From Owns O  JOIN Traveler T ON O.ssn = T.ssn  Group BY T.ssn  HAVING COUNT(O.passport\_number) > 0  INTERSECT  SELECT T.ssn  FROM GoesOn G  JOIN Traveler T ON G.ssn = T.ssn  GROUP BY T.ssn  HAVING COUNT(G.id) > 1; |  |

1. [10 Points] Find the start and end locations of trips booked by each travel agent. You must use natural join in your query.

|  |  |
| --- | --- |
| SQL Statements | Output |
| SELECT B.agent, T.start\_location, T.end\_location  FROM Trip T  NATURAL JOIN (  SELECT trip\_id AS id, agent  FROM Booking  ) B; |  |

1. [10 Points] Find the names of travel agents who have fewer years of experience than any agent with more than 10 years of experience.

|  |  |
| --- | --- |
| SQL Statements | Output |
| SELECT name  FROM TravelAgent A1  WHERE A1.years\_experience < ANY (  SELECT A2.years\_experience  FROM TravelAgent A2  WHERE A2.years\_experience > 10  ); |  |

1. [10 Points] Find the details of trips where the trip’s end date is after the traveler’s passport expiration date. You must use theta join in your query.

|  |  |
| --- | --- |
| SQL Statements | Output |
| SELECT T.\*  FROM Trip T  JOIN Booking B ON T.id = B.trip\_id  JOIN Owns O ON B.traveler\_ssn = O.ssn  JOIN Passport P ON O.passport\_number = P.passport\_number  WHERE T.end\_date > P.expirationDate; |  |

1. [10 Points] List all travel agents and the number of trips they have booked. Only show agents who have booked more than 1 trip. Use GROUP BY and HAVING to solve this question.

|  |  |
| --- | --- |
| SQL Statements | Output |
| SELECT A.\*, COUNT(B.trip\_id) AS trip\_count  FROM TravelAgent A  JOIN Booking B ON A.name = B.agent  GROUP BY A.name  HAVING COUNT(B.trip\_id) > 1; |  |

1. [10 Points] List the travelers who have not gone on any trips.

|  |  |
| --- | --- |
| SQL Statements | Output |
| SELECT T.\*  FROM Traveler T  LEFT JOIN GoesOn G on T.ssn = G.ssn  WHERE G.ssn IS NULL; |  |

1. [10 Points] Find all the trips booked by travel agents with fewer than 5 years of experience.

|  |  |
| --- | --- |
| SQL Statements | Output |
| SELECT T.\*  FROM Trip T  JOIN Booking B ON T.id = B.trip\_id  JOIN TravelAgent A ON B.agent= A.name  WHERE A.years\_experience < 5; |  |