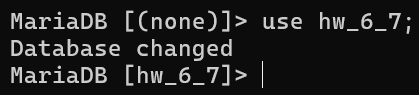
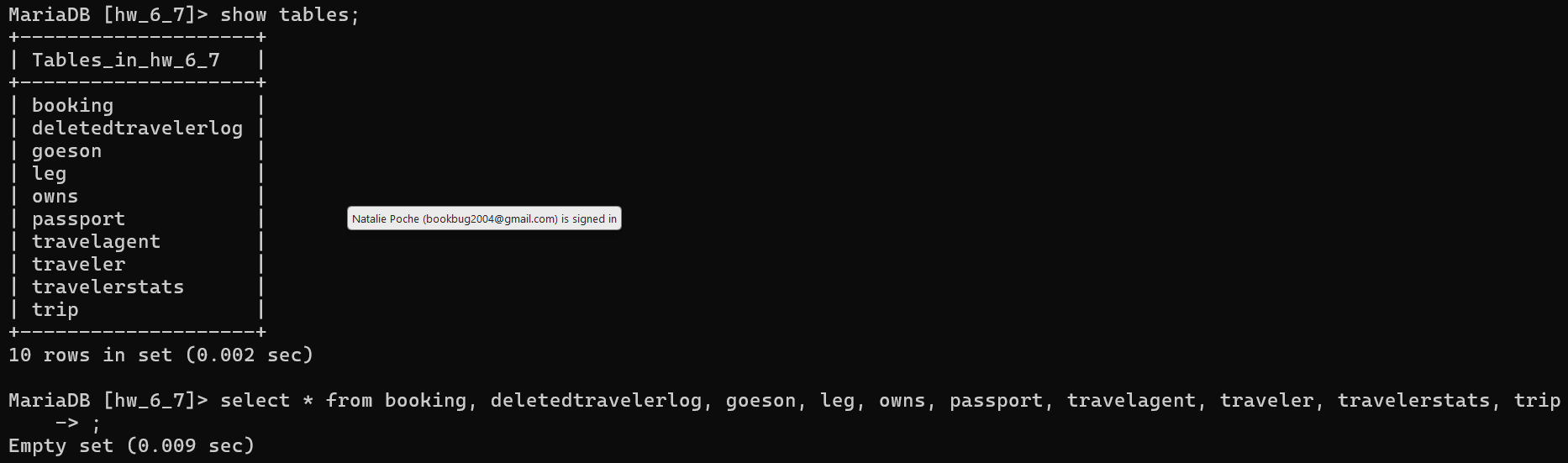
Homework 7

Problem Statement: This assignment uses the same database named HW 6 7 that you created in Homework 6. Before beginning this assignment, do the following:

* Use the HW 6 7 database. Do not create a new one.



* Delete all existing records from the tables to start with a clean slate.



* Repopulate the database with the new data provided in the INSERT statements below. These updated records will be used to complete all tasks in Homework 7.

|  |  |
| --- | --- |
| SQL Statements | SQL Implementation |
| INSERT INTO Traveler (name, ssn, dob)  VALUES ('John Doe', 101, '1985-06-12'),  ('Alice Brown', 102, '1992-03-05'),  ('Mike Johnson', 103, '1998-09-17'),  ('Lisa Turner', 104, '2000-12-22'),  ('Sarah Connor', 105, '2003-11-01'),  ('David Harris', 106, '1980-07-15'),  ('Emma Watson', 107, '1995-01-08'),  ('James Miller', 108, '1999-05-21');  SELECT \* FROM Traveler; |  |

|  |  |
| --- | --- |
| SQL Statements | SQL Implementation |
| INSERT INTO TravelAgent (name, years\_experience, phone)  VALUES  ('Emily Clark', 12, '123-456-7890'),  ('Robert Smith', 8, '234-567-8901'),  ('Anna Wilson', 15, '345-678-9012'),  ('Michael Davis', 10, '456-789-0123'),  ('Mary Johnson', 3, '567-890-1234'),  ('Sarah Williams', 18, '678-901-2345');  SELECT \* FROM TravelAgent; |  |

|  |  |
| --- | --- |
| SQL Statements | SQL Implementation |
| INSERT INTO Trip (id, start\_location, end\_location, start\_date, end\_date)  VALUES (201, 'New York', 'Paris', '2022-07-10', '2022-07-20'),  (202, 'Tokyo', 'Sydney', '2023-08-01', '2023-08-15'),  (203, 'London', 'Rome', '2025-04-16', '2025-05-30'),  (204, 'Berlin', 'Tokyo', '2025-04-16', '2025-04-28'),  (205, 'Miami', 'New York', '2025-11-22', '2025-11-25'),  (206, 'Madrid', 'Dubai', '2026-12-01', '2026-12-15'),  (207, 'Beijing', 'Hong Kong', '2026-01-10', '2026-01-20'),  (208, 'Los Angeles', 'Paris', '2026-02-15', '2026-02-25');  SELECT \* FROM Trip; |  |

|  |  |
| --- | --- |
| SQL Statements | SQL Implementation |
| INSERT INTO Passport (passport\_number, country, expirationDate, holderName)  VALUES  (3001, 'USA', '2025-11-30', 'John Doe'),  (3002, 'Canada', '2026-08-20', 'Alice Brown'),  (3003, 'UK', '2025-09-15', 'Mike Johnson'),  (3004, 'Australia', '2027-02-10', 'Lisa Turner'),  (3005, 'France', '2023-12-05', 'Sarah Connor'),  (3006, 'Germany', '2028-06-25', 'David Harris'),  (3007, 'USA', '2025-01-30', 'Emma Watson'),  (3008, 'Italy', '2025-08-20', 'James Miller');  SELECT \* FROM Passport; |  |

|  |  |
| --- | --- |
| SQL Statements | SQL Implementation |
| INSERT INTO Owns (ssn, passport\_number, country)  VALUES  (101, 3001, 'USA'),  (102, 3002, 'Canada'),  (103, 3003, 'UK'),  (104, 3004, 'Australia'),  (105, 3005, 'France'),  (106, 3006, 'Germany'),  (107, 3007, 'USA'),  (108, 3008, 'Italy');  SELECT \* FROM Owns; |  |

|  |  |
| --- | --- |
| SQL Statements | SQL Implementation |
| INSERT INTO Booking (agent, traveler\_ssn, trip\_id)  VALUES  ('Emily Clark', 101, 201),  ('Robert Smith', 102, 202),  ('Anna Wilson', 103, 203),  ('Michael Davis', 104, 204),  ('Emily Clark', 105, 205),  ('Sarah Williams', 106, 206),  ('Anna Wilson', 107, 207),  ('Emily Clark', 108, 208);  SELECT \* FROM Booking; |  |

|  |  |
| --- | --- |
| SQL Statements | SQL Implementation |
| INSERT INTO GoesOn (ssn, id)  VALUES  (101, 201),  (102, 202),  (103, 203),  (104, 204),  (105, 205),  (106, 206),  (107, 207),  (108, 208);  SELECT \* FROM GoesOn; |  |

|  |  |
| --- | --- |
| SQL Statements | SQL Implementation |
| INSERT INTO Leg (trip\_id, startLocation, endLocation, startDate, endDate)  VALUES  (201, 'New York', 'Madrid', '2022-07-10', '2022-07-11'),  (202, 'Tokyo', 'Kuala Lumpur', '2023-08-01', '2023-08-01'),  (203, 'London', 'Paris', '2025-04-16', '2025-04-17'),  (204, 'Berlin', 'Istanbul', '2025-04-16', '2025-04-16'),  (205, 'Miami', 'Atlantic City', '2025-11-22', '2025-11-22'),  (206, 'Madrid', 'Amman', '2026-12-01', '2026-12-02'),  (207, 'Beijing', 'Seoul', '2026-01-10', '2026-01-10'),  (208, 'Los Angeles', 'New York', '2026-02-15', '2026-02-15');  SELECT \* FROM Leg; |  |

Part 1: Create Standard Views

1. View: TopAgents (10 points) Create a view named TopAgents that lists all travel agents who have more than 10 years of experience. Include the name and years experience columns.

|  |  |
| --- | --- |
| SQL Statements | SQL Implementation |
| CREATE VIEW TopAgents AS  SELECT name, years\_experience  FROM TravelAgent  WHERE years\_experience > 10;  SELECT \*  FROM TopAgents; |  |

1. View: ActiveTrips (10 points) Create a view named ActiveTrips that includes details of all trips where the end date is later than the current date. Include the id, start location, end location, and end date columns.

|  |  |
| --- | --- |
| SQL Statements | SQL Implementation |
| CREATE VIEW ActiveTrips AS  SELECT id, start\_location, end\_location, end\_date  FROM Trip  WHERE end\_date > CURRENT\_DATE();  SELECT \*  FROM ActiveTrips; |  |

1. View: PassportHoldersByCountry (15 points) Create a view named PassportHoldersByCountry that lists the number of passport holders grouped by country. Include country and the count of passport holders as passport count.

|  |  |
| --- | --- |
| SQL Statements | SQL Implementation |
| CREATE VIEW PassportHoldersByCountry AS  SELECT P.country, COUNT(\*) AS passport\_count  FROM Passport P  GROUP BY P.country;  SELECT \*  FROM PassportHoldersByCountry; |  |

1. View: AgentBookings (15 points) Create a view named AgentBookings that lists each agent’s name along with the total number of trips they have booked. Include the agent and the total number of trips as trip count.

|  |  |
| --- | --- |
| SQL Statements | SQL Implementation |
| CREATE VIEW AgentBookings AS  SELECT B.agent, COUNT(\*) AS trip\_count  FROM Booking B  GROUP BY B.agent;  SELECT \*  FROM AgentBookings; |  |

1. View: UpcomingTripsByTraveler (20 points) Create a view named UpcomingTripsByTraveler that lists each traveler’s name, their upcoming trips (trips starting after today using CURRENT DATE() function), and the trip start and end dates. Include name, id, start date, and end date.

|  |  |
| --- | --- |
| SQL Statements | SQL Implementation |
| CREATE VIEW UpcomingTripsByTraveler AS  SELECT T.name, B.trip\_id, Tr.start\_date, Tr.end\_date  FROM Traveler T  JOIN Booking B ON B.traveler\_ssn = T.ssn  JOIN Trip Tr ON Tr.id = B.trip\_id  WHERE Tr.start\_date > CURRENT\_DATE();  SELECT \*  FROM UpcomingTripsByTraveler; |  |

Part 2: Scenarios and Queries

1. (5 points) List all travel agents who have booked trips for travelers on trips starting after today.

|  |  |
| --- | --- |
| SQL Statements | SQL Implementation |
| SELECT DISTINCT B.agent  FROM UpcomingTripsByTraveler T  JOIN Booking B ON B.trip\_id = T.trip\_id  WHERE T.start\_date > CURRENT\_DATE(); |  |

1. (5 points) Identify the country with the highest number of passport holders.

|  |  |
| --- | --- |
| SQL Statements | SQL Implementation |
| SELECT A.country  FROM PassportHoldersByCountry A  WHERE A.passport\_count >= (  SELECT MAX(\*)  ); |  |

1. (10 points) Find the traveler(s) who have the most upcoming trips.

|  |  |
| --- | --- |
| SQL Statements | SQL Implementation |
| SELECT A.name  FROM UpcomingTripsByTraveler A  GROUP BY A.name  HAVING COUNT(\*) = (  SELECT MAX(C.trip\_count)  FROM(  SELECT COUNT(\*) AS trip\_count  FROM UpcomingTripsByTraveler B  GROUP BY B.name  ) AS C  );  SELECT A.name, COUNT(\*)  FROM UpcomingTripsByTraveler A  GROUP BY A.name  HAVING COUNT(\*) >= ANY(  SELECT COUNT(\*)  FROM UpcomingTripsByTraveler B  GROUP BY B.name  HAVING COUNT(\*) > 0); |  |

1. (10 points) List all active trips (as per ActiveTrips) with their corresponding travel agents.

|  |  |
| --- | --- |
| SQL Statements | SQL Implementation |
| SELECT B.agent, A.id, A.start\_location, A.end\_location, A.end\_date  FROM ActiveTrips A, Booking Bo  JOIN AgentBookings B ON B.agent = Bo.agent  WHERE Bo.trip\_id = A.id; |  |