Homework #7

CIS 4301 - Spring 2025

Submission Format

Submit a soft copy of your solution via e-Learning (`http://elearning.ufl.edu`) by the end of the day (23:59 / 11:59 PM) on Wednesday, April 23rd. Save your solution as a PDF file and name it `hw7.pdf`. Include your name, assignment number, and due date at the top of the file.

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
-- Traveler
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');


-- TravelAgent
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');


-- Trip
INSERT INTO Trip (id, start_location, end_location, start_date, end_date) VALUES
(201, 'New York', 'Paris', '2022-07-10', '2022-07-20'),      -- Past trip
(202, 'Tokyo', 'Sydney', '2023-08-01', '2023-08-15'),        -- Past trip
(203, 'London', 'Rome', '2025-04-16', '2025-05-30'),         -- Ongoing trip
(204, 'Berlin', 'Tokyo', '2025-04-16', '2025-04-28'),        -- Ongoing trip
(205, 'Miami', 'New York', '2025-11-22', '2025-11-25'),      -- Future trip
(206, 'Madrid', 'Dubai', '2026-12-01', '2026-12-15'),        -- Future trip
(207, 'Beijing', 'Hong Kong', '2026-01-10', '2026-01-20'),   -- Future trip
(208, 'Los Angeles', 'Paris', '2026-02-15', '2026-02-25');   -- Future trip
```

```sql
-- Passport
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');


-- Owns
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');


-- Booking
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);
```

```
-- GoesOn
INSERT INTO GoesOn (ssn, id) VALUES
(101, 201),
(102, 202),
(103, 203),
(104, 204),
(105, 205),
(106, 206),
(107, 207),
(108, 208);

-- Leg
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');
```

Part 1: Create Standard Views (70 points)

Using the schema provided, write SQL statements to create the following 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.

2. **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.

3. **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`.

4. **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`.

5. **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`.

Part 2: Scenarios and Queries (30 points)

Answer the following questions using views created in Part 1. Submit the SQL queries and their corresponding outputs.

1. **(5 points)** List all travel agents who have booked trips for travelers on trips starting after today.
2. **(5 points)** Identify the country with the highest number of passport holders.
3. **(10 points)** Find the traveler(s) who have the most upcoming trips.
4. **(10 points)** List all active trips (as per `ActiveTrips`) with their corresponding travel agents.

What to Submit

For this assignment, include the following in your submission:

- **Part 1: SQL Commands for Standard Views**
  - Submit the SQL commands used to create each view.
  - Submit the SQL queries used to retrieve data from each view.
  - Include screenshots showing the output of each query.
- **Part 2: SQL Queries and Outputs for Scenarios**
  - Submit the SQL queries written to answer each of the four scenarios.
  - Include screenshots showing the output for each query.

Ensure all SQL commands and screenshots are clearly labeled and formatted for readability.