

Homework 4: SQL 2

Step 1: Start MariaDB (0 pts)

```
MySQL Client (MariaDB 11.7) × + v
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 11.7.2-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> |
```

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

Create New Relations

GoesOn(ssn: integer, id: integer)

SQL Statements	Output
<pre>CREATE TABLE GoesOn(ssn INTEGER, id INTEGER, PRIMARY KEY(ssn, id)); SHOW TABLES; DESCRIBE GoesOn;</pre>	<pre>MariaDB [hw4_database]> CREATE TABLE GoesOn(-> ssn INTEGER, -> id INTEGER, -> -> PRIMARY KEY(ssn, id) ->); Query OK, 0 rows affected (0.021 sec) MariaDB [hw4_database]> SHOW TABLES; +-----+ Tables_in_hw4_database +-----+ booking goeson travelagent traveler trip +-----+ 5 rows in set (0.009 sec) MariaDB [hw4_database]> DESCRIBE goeson; +-----+-----+-----+-----+-----+-----+ Field Type Null Key Default Extra +-----+-----+-----+-----+-----+-----+ ssn int(11) NO PRI NULL id int(11) NO PRI NULL +-----+-----+-----+-----+-----+-----+ 2 rows in set (0.051 sec)</pre>

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

SQL Statements	Output
<pre>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;</pre>	<pre>MariaDB [hw4_database]> CREATE TABLE Leg(-> Trip_id INTEGER, -> startLocation VARCHAR(50), -> endLocation VARCHAR(50), -> startDate DATE, -> endDate DATE, -> -> PRIMARY KEY(Trip_id, startLocation, endLocation) ->); Query OK, 0 rows affected (0.024 sec) MariaDB [hw4_database]> SHOW TABLES; +-----+ Tables_in_hw4_database +-----+ booking goeson leg travelagent traveler trip +-----+ 6 rows in set (0.002 sec) MariaDB [hw4_database]> DESCRIBE Leg; +-----+-----+-----+-----+-----+-----+ Field Type Null Key Default Extra +-----+-----+-----+-----+-----+-----+ Trip_id int(11) NO PRI NULL startLocation varchar(50) NO PRI NULL endLocation varchar(50) NO PRI NULL startDate date YES NULL endDate date YES NULL +-----+-----+-----+-----+-----+-----+ 5 rows in set (0.043 sec)</pre>

Owens(ssn: integer, passport_number: Integer, country: String)

SQL Statements	Output
<pre>CREATE TABLE Owens(ssn INTEGER, passport_number INTEGER, country VARCHAR(50), PRIMARY KEY(passport_number, country)); SHOW TABLES; DESCRIBE Owens;</pre>	<pre>MariaDB [hw4_database]> CREATE TABLE Owens(-> ssn INTEGER, -> passport_number INTEGER, -> country VARCHAR(50), -> -> PRIMARY KEY(passport_number, country) ->); Query OK, 0 rows affected (0.022 sec) MariaDB [hw4_database]> MariaDB [hw4_database]> SHOW TABLES; +-----+ Tables_in_hw4_database +-----+ booking goeson leg owns travelagent traveler trip +-----+ 7 rows in set (0.002 sec) MariaDB [hw4_database]> MariaDB [hw4_database]> DESCRIBE Owens; +-----+-----+-----+-----+-----+-----+ Field Type Null Key Default Extra +-----+-----+-----+-----+-----+-----+ ssn int(11) YES NULL passport_number int(11) NO PRI NULL country varchar(50) NO PRI NULL +-----+-----+-----+-----+-----+-----+ 3 rows in set (0.033 sec)</pre>

Passport(passport_number: integer, country: String, expirationDate: Date, holderName: String)

SQL Statements	Output
<pre>CREATE TABLE Passport(Passport_number INTEGER, country VARCHAR(50), expirationDate DATE, holderName VARCHAR(50), PRIMARY KEY(passport_number, country)); SHOW TABLES; DESCRIBE Passport;</pre>	<pre>MariaDB [hw4_database]> CREATE TABLE Passport(-> Passport_number INTEGER, -> country VARCHAR(50), -> expirationDate DATE, -> holderName VARCHAR(50), -> -> PRIMARY KEY(passport_number, country) ->); Query OK, 0 rows affected (0.020 sec) MariaDB [hw4_database]> MariaDB [hw4_database]> SHOW TABLES; +-----+ Tables_in_hw4_database +-----+ booking goeson leg owns passport travelagent traveler trip +-----+ 8 rows in set (0.001 sec) MariaDB [hw4_database]> MariaDB [hw4_database]> DESCRIBE Passport; +-----+-----+-----+-----+-----+-----+ Field Type Null Key Default Extra +-----+-----+-----+-----+-----+-----+ Passport_number int(11) NO PRI NULL country varchar(50) NO PRI NULL expirationDate date YES NULL holderName varchar(50) YES NULL +-----+-----+-----+-----+-----+-----+ 4 rows in set (0.027 sec)</pre>

Insert Data

SQL Statements	Output
<pre>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;</pre>	<pre>MariaDB [hw4_database]> 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); Query OK, 10 rows affected (0.016 sec) Records: 10 Duplicates: 0 Warnings: 0 MariaDB [hw4_database]> MariaDB [hw4_database]> SELECT * FROM GoesOn; +-----+-----+ ssn id +-----+-----+ 101 201 101 208 102 202 102 205 103 203 103 206 106 201 106 206 107 207 108 208 +-----+-----+ 10 rows in set (0.005 sec)</pre>

SQL Statements	Output
<pre>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;</pre>	<pre> MariaDB [hw4_database]> 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'); Query OK, 16 rows affected (0.010 sec) Records: 16 Duplicates: 0 Warnings: 0 MariaDB [hw4_database]> SELECT * FROM Leg; +-----+-----+-----+-----+-----+ Trip_id startLocation endLocation startDate endDate +-----+-----+-----+-----+-----+ 201 London Paris 2025-07-13 2025-07-20 201 New York London 2025-07-10 2025-07-12 202 Seoul Sydney 2025-08-06 2025-08-15 202 Tokyo Seoul 2025-08-01 2025-08-05 203 Berlin Rome 2025-09-09 2025-09-15 203 London Berlin 2025-09-05 2025-09-08 204 Atlanta New York 2025-06-18 2025-06-20 204 Miami Atlanta 2025-06-15 2025-06-17 205 Frankfurt Berlin 2025-10-07 2025-10-10 205 San Francisco Frankfurt 2025-10-01 2025-10-06 206 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 Istanbul Dubai 2025-09-13 2025-09-15 208 New York Istanbul 2025-09-10 2025-09-12 +-----+-----+-----+-----+-----+ 16 rows in set (0.001 sec) </pre>

SQL Statements	Output
<pre>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;</pre>	<pre>MariaDB [hw4_database]> 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'); Query OK, 8 rows affected (0.008 sec) Records: 8 Duplicates: 0 Warnings: 0 MariaDB [hw4_database]> MariaDB [hw4_database]> SELECT * FROM Owns; +-----+-----+-----+ ssn passport_number country +-----+-----+-----+ 101 5001 USA 102 5002 Canada 103 5003 UK 104 5004 Germany 105 5005 France 106 5006 Australia 107 5007 Japan 108 5008 India +-----+-----+-----+ 8 rows in set (0.001 sec)</pre>

SQL Statements	Output
<pre>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;</pre>	<pre>MariaDB [hw4_database]> INSERT INTO Passport(passport_number, country, expir ationDate, 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'); Query OK, 8 rows affected (0.002 sec) Records: 8 Duplicates: 0 Warnings: 0 MariaDB [hw4_database]> MariaDB [hw4_database]> SELECT * FROM Passport; +-----+-----+-----+-----+ Passport_number country expirationDate holderName +-----+-----+-----+-----+ 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 +-----+-----+-----+-----+ 8 rows in set (0.001 sec)</pre>

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
<pre>SELECT DISTINCT agent FROM Booking A, Traveler T WHERE A.traveler_ssn = T.ssn AND T.dob = '1985-06-12';</pre>	<pre>MariaDB [hw4_database]> SELECT DISTINCT agent -> FROM Booking A, Traveler T -> WHERE A.traveler_ssn = T.ssn -> AND T.dob = '1985-06-12'; +-----+ agent +-----+ Alice Brown +-----+ 1 row in set (0.015 sec)</pre>

2. [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
<pre>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');</pre>	<pre>MariaDB [hw4_database]> 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' ->); +-----+ dob +-----+ 1998-09-17 1999-02-25 +-----+ 2 rows in set (0.002 sec)</pre>

3. [10 Points] Retrieve the names of travel agents who have more experience than any agent whose name starts with 'John'.

SQL Statements	Output
<pre>SELECT name FROM TravelAgent A1 WHERE (A1.years_experience > ANY (SELECT years_experience FROM TravelAgent A2 WHERE A2.name LIKE 'John%'));</pre>	<pre>MariaDB [hw4_database]> SELECT name -> FROM TravelAgent A1 -> WHERE (-> A1.years_experience > ANY (-> SELECT years_experience -> FROM TravelAgent A2 -> WHERE A2.name LIKE 'John%') ->);</pre> <pre>+-----+ name +-----+ Alice Brown Daniel Lee Sarah Williams +-----+ 3 rows in set (0.027 sec)</pre>

4. [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
<pre>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;</pre>	<pre>MariaDB [hw4_database]> 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;</pre> <pre>+-----+ ssn +-----+ 101 102 103 106 +-----+ 4 rows in set (0.009 sec)</pre>

Name: Natalie Poche

Assignment: Homework 4 – SQL 2

Due: March 14, 2025

5. [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																														
<pre>SELECT B.agent, T.start_location, T.end_location FROM Trip T NATURAL JOIN (SELECT trip_id AS id, agent FROM Booking) B;</pre>	<pre>MariaDB [hw4_database]> SELECT B.agent, T.start_location, T.end_location -> FROM Trip T -> NATURAL JOIN (SELECT trip_id AS id, agent FROM Booking) B;</pre> <table><tr><th>agent</th><th>start_location</th><th>end_location</th></tr><tr><td>Alice Brown</td><td>New York</td><td>Paris</td></tr><tr><td>Alice Brown</td><td>New York</td><td>Dubai</td></tr><tr><td>Alice Brown</td><td>Chicago</td><td>Los Angeles</td></tr><tr><td>Daniel Lee</td><td>San Francisco</td><td>Berlin</td></tr><tr><td>John Smith</td><td>Tokyo</td><td>Sydney</td></tr><tr><td>John Smith</td><td>New York</td><td>Paris</td></tr><tr><td>Michael Johnson</td><td>London</td><td>Rome</td></tr><tr><td>Rachel Green</td><td>Boston</td><td>Dubai</td></tr><tr><td>Sarah Williams</td><td>Miami</td><td>New York</td></tr></table> <pre>9 rows in set (0.013 sec)</pre>	agent	start_location	end_location	Alice Brown	New York	Paris	Alice Brown	New York	Dubai	Alice Brown	Chicago	Los Angeles	Daniel Lee	San Francisco	Berlin	John Smith	Tokyo	Sydney	John Smith	New York	Paris	Michael Johnson	London	Rome	Rachel Green	Boston	Dubai	Sarah Williams	Miami	New York
agent	start_location	end_location																													
Alice Brown	New York	Paris																													
Alice Brown	New York	Dubai																													
Alice Brown	Chicago	Los Angeles																													
Daniel Lee	San Francisco	Berlin																													
John Smith	Tokyo	Sydney																													
John Smith	New York	Paris																													
Michael Johnson	London	Rome																													
Rachel Green	Boston	Dubai																													
Sarah Williams	Miami	New York																													

6. [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						
<pre>SELECT name FROM TravelAgent A1 WHERE A1.years_experience < ANY (SELECT A2.years_experience FROM TravelAgent A2 WHERE A2.years_experience > 10);</pre>	<pre>MariaDB [hw4_database]> SELECT name -> FROM TravelAgent A1 -> WHERE A1.years_experience < ANY (-> SELECT A2.years_experience -> FROM TravelAgent A2 -> WHERE A2.years_experience > 10 ->);</pre> <table><tr><th>name</th></tr><tr><td>Alice Brown</td></tr><tr><td>John Smith</td></tr><tr><td>Michael Johnson</td></tr><tr><td>Rachel Green</td></tr><tr><td>Sarah Williams</td></tr></table> <pre>5 rows in set (0.026 sec)</pre>	name	Alice Brown	John Smith	Michael Johnson	Rachel Green	Sarah Williams
name							
Alice Brown							
John Smith							
Michael Johnson							
Rachel Green							
Sarah Williams							

7. [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										
<pre>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;</pre>	<pre>MariaDB [hw4_database]> 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;</pre> <table><tr><th>id</th><th>start_location</th><th>end_location</th><th>start_date</th><th>end_date</th></tr><tr><td>201</td><td>New York</td><td>Paris</td><td>2025-07-10</td><td>2025-07-20</td></tr></table> <pre>1 row in set (0.009 sec)</pre>	id	start_location	end_location	start_date	end_date	201	New York	Paris	2025-07-10	2025-07-20
id	start_location	end_location	start_date	end_date							
201	New York	Paris	2025-07-10	2025-07-20							

8. [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;	<pre>MariaDB [hw4_database]> 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;</pre> <table><tr><th>name</th><th>years_experience</th><th>phone</th><th>trip_count</th></tr><tr><td>Alice Brown</td><td>12</td><td>123-456-7890</td><td>3</td></tr><tr><td>John Smith</td><td>8</td><td>234-567-8901</td><td>2</td></tr></table> <pre>2 rows in set (0.021 sec)</pre>	name	years_experience	phone	trip_count	Alice Brown	12	123-456-7890	3	John Smith	8	234-567-8901	2
name	years_experience	phone	trip_count										
Alice Brown	12	123-456-7890	3										
John Smith	8	234-567-8901	2										

9. [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;	<pre>MariaDB [hw4_database]> SELECT T.* -> FROM Traveler T -> LEFT JOIN GoesOn G on T.ssn = G.ssn -> WHERE G.ssn IS NULL;</pre> <table><tr><th>name</th><th>ssn</th><th>dob</th></tr><tr><td>James Miller</td><td>105</td><td>2000-08-14</td></tr><tr><td>Laura White</td><td>104</td><td>1995-04-23</td></tr></table> <pre>2 rows in set (0.109 sec)</pre>	name	ssn	dob	James Miller	105	2000-08-14	Laura White	104	1995-04-23
name	ssn	dob								
James Miller	105	2000-08-14								
Laura White	104	1995-04-23								

10. [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;	<pre>MariaDB [hw4_database]> 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;</pre> <table><tr><th>id</th><th>start_location</th><th>end_location</th><th>start_date</th><th>end_date</th></tr><tr><td>207</td><td>Boston</td><td>Dubai</td><td>2025-07-05</td><td>2025-07-18</td></tr></table> <pre>1 row in set (0.005 sec)</pre>	id	start_location	end_location	start_date	end_date	207	Boston	Dubai	2025-07-05	2025-07-18
id	start_location	end_location	start_date	end_date							
207	Boston	Dubai	2025-07-05	2025-07-18							