(Airline Ticket Reservation System)

FINAL PROJECT REPORT

CSD 3204-Relational Database & SQL

GROUP G7

Sujal Patel (C0735603)
Himanshu Patel (C0735691)
Shubham (C0737342)
Deepak Punia (C0739472)
Bhagat Singh (C0735947)
Jattinder Kaur (C0737459)

The aim of this project is to develop an application which would facilitate the reservation of online air tickets through an effective and yet simple GUI for a normal passenger intending to travel in airways.
The project is basically targeted at those people who would like to travel through air. Apart from reserving tickets, through this system a passenger can compare fares 'from' various cities 'to' various cities.

You will focus on the backend using Oracle. Which mean you have design and implement database for Airline Ticket Reservation System
The design of the database is an art-work and is up to the student, however, it should provide the following facilities:

Flight:
Each flight has flight number, aircraft, source, destination, air-line and flight time.

Reservation:
Passenger, ticket, airline, source, destination, etc.

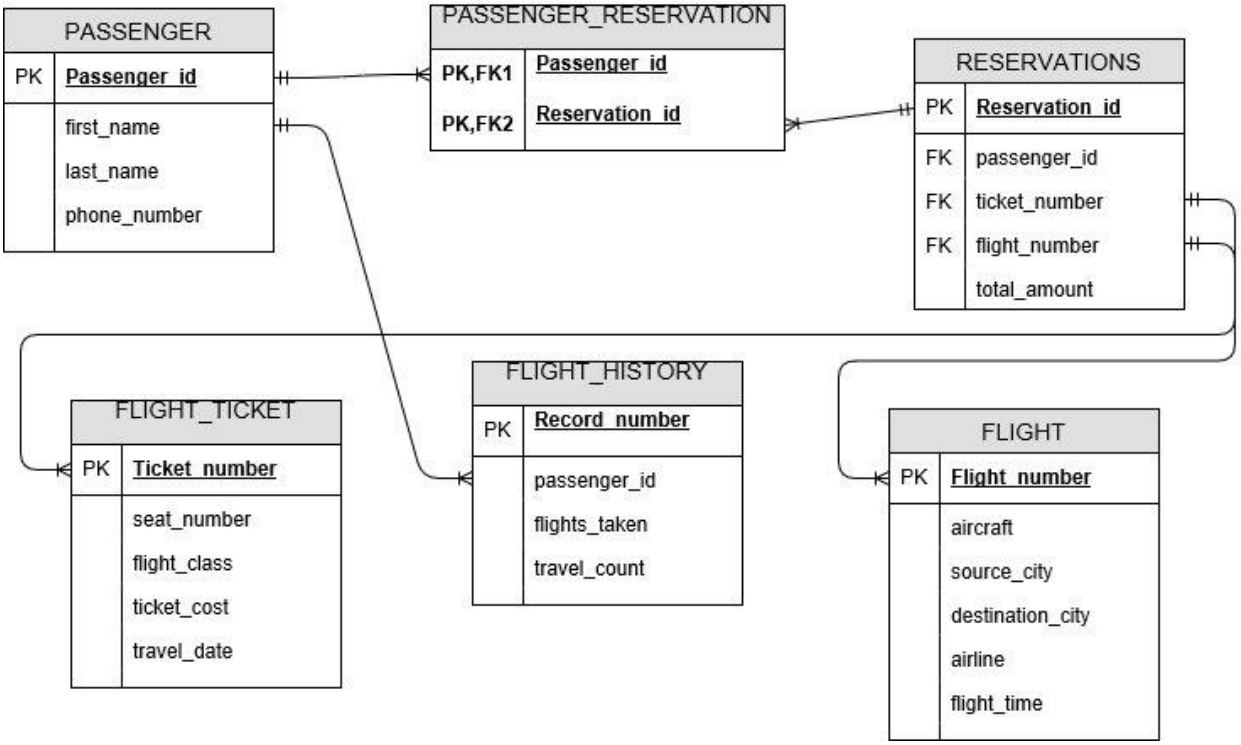
Ticket:
To be designed to fulfill the requirements of the queries.

Passenger:
To be designed to fulfill the requirements of the queries.

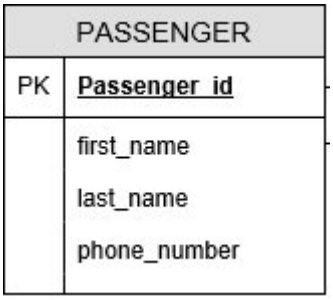
Note:
More than one airline travel from one city to the other city. Also, you may add more tables if required.

Part 1:

1) Create ERD for database
    a. Define the entities, the attributes and the relationship



(1.) **Passenger:** Details of passenger

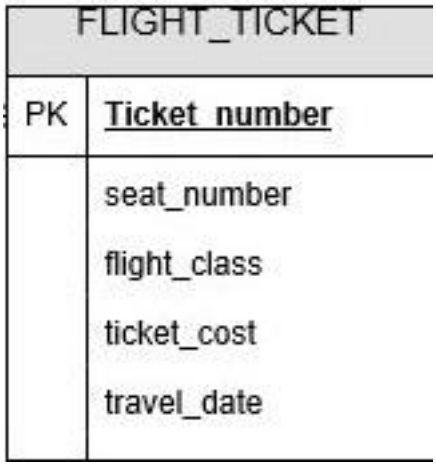


Description of Entities

passenger_id	Unique identification number
first_name	First name of passenger
last_name	Last name of passenger
phone_number	Contact of passenger

Primary Key: Passenger_id
Attributes: first_name, last_name, phone_number

(2.) **Flight_ticket :** Details of passenger

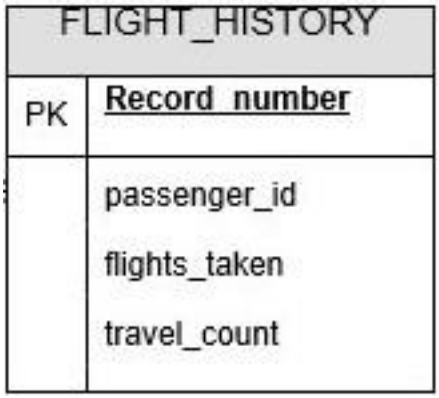


Description of Entities

ticket_number	Number of ticket
seat_number	Number of the seat
flight_class	Type of seating
ticket_cost	Price of ticket
travel_date	Date of flight

Primary Key: Ticket_number
Attributes: seat_number,class,cost,travel_date

(3.) **Flight_history :** Details of record



Description of Entites

Record_number	Unique identification number
Passenger_id	Unique identification number of passenger

Flights_taken	Number of flights taken by passenger
Travel_count	Total number of travels

Primary Key: Record_number
Attributes: passenger_id,flights_taken,travel_count

(4.) **Reservations:** Details of Reservation

RESERVATIONS	
PK	**Reservation_id**
FK	passenger_id
FK	ticket_number
FK	flight_number
	total_amount

Description of Entities

Reservation_id	Unique identification number
Passenger_id	Unique identification number of passenger
Ticket_number	Number of ticket
Flight_number	Number of flight
Total_amount	Total price paid

Primary Key: Reservation_id
Foreign Key: passenger_id, ticket_number, flight_number
Attributes: total_amount

(5.) **Flight:** Details of Flight

FLIGHT	
PK	**Flight_number**
	aircraft
	source_city
	destination_city
	airline
	flight_time

Description of Entites

Flight_number	Unique identification number
Aircraft	Name of aircraft
Source_city	Source city of travel
Destination_city	Destination city of travel
Airline	Name of airline
Flight_time	Time of flight

Primary key: Flight_number
Attributes: aircraft,source,destination,airline,flight_time

Relationship between tables:

(1.) Relationship between Passenger and Reservations:

It is a many to many relationship.
Each passenger has maximum many and minimum one reservation.
Each reservation has maximum many and minimum one passenger.

(2.) Relationship between Reservations and Tickets:

It is a one to many relationship.
Each Reservation has maximum many and minimum one Ticket.
Each Ticket has only one reservation.

(3.) Relationship between Reservations and Flights:

It is a one to many relationship.
Each Reservation has maximum many and minimum one Flights.
Each Flights has only one reservation.

(4.) Relationship between Passenger and Record:

It is a one to many relationship.
Each Record has maximum many and minimum one Passenger.
Each Passenger has only one Record.

2. Perform normalization (3th normal form)

Database Normalization is a technique of organizing the data in the database. Normalization is a systematic approach of decomposing tables to eliminate data redundancy and anomalies like insertion, deletion.
It is used for mainly two purposes:

- Eliminating redundant (useless) data.
- Ensuring data dependencies make sense i.e. data is logically stored.

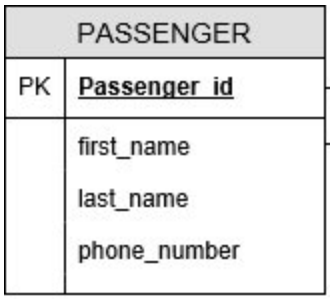
CRITERIA for each NORMAL FORM:

1. First NORMAL form(1NF):
   - Each cell (intersection of row and column) of the table must be single valued i.e. no multivalued columns.
   - Each column contains data for a single attribute of the thing being described.
   - A primary key has been defined.
   - All columns in the table are dependent on primary key.

2. Second NORMAL form(2NF):
   - Table must be in first normal form(1NF).
   - There are no partial dependencies i.e. each non-key attribute depends on the entire primary key.
3. Third NORMAL form(3NF):
   - Table must be in second normal form(2NF).
   - Table has no transitive dependencies i.e. every attribute must depend only on the primary key and not another non-key column.

   Below we have all the tables of our project. Now, we are going to apply above normalization criteria one by one on our tables.

(1.) Passenger:



- Primary key is defined, Hence, it is in 1NF.
- No, partial dependencies- it is in 2NF.
- No, transitive dependencies- it is in 3NF.

(2.) Flight_Ticket:

FLIGHT_TICKET	
PK	**Ticket_number**
	seat_number
	flight_class
	ticket_cost
	travel_date

- Primary key is defined, Hence, it is in 1NF.
- No, partial dependencies- it is in 2NF.
- No, transitive dependencies- it is in 3NF.

(3.) Flight_History:

FLIGHT_HISTORY	
PK	**Record_number**
	passenger_id
	flights_taken
	travel_count

- Primary key is defined, no multivalued columns, no repeating groups.Hence, it is in 1NF.

Now, the above table is not in 2NF because there is partial dependency i.e. non key is dependent on part of but not the entire composite key. here, Flights_taken and Travel_count depends on passenger_id but has no connection with Record_number. Thus, Flights_taken and travel_count non-key columns do not depend on entire composite key.

To remove the partial dependencies in Flight_History table, the Passenger table is created.

Passenger

Primary Key		
Passenger_id	Flights_taken	Travel_count

Flight_History

Primary_Key	Primary_Key
Record_number	Passenger_id

- No, partial dependencies- it is in 2NF.
- No, transitive dependencies- it is in 3NF.

(4.) Flight:

FLIGHT	
PK	**Flight_number**
	aircraft
	source_city
	destination_city
	airline
	flight_time

- Primary key is defined, Hence, it is in 1NF.
- No, partial dependencies- it is in 2NF.
- No, transitive dependencies- it is in 3NF.

(5.) Reservations:

RESERVATIONS	
PK	**Reservation_id**
FK	passenger_id
FK	ticket_number
FK	flight_number
	total_amount

- Primary key is defined, no multivalued columns, no repeating groups.Hence, it is in 1NF.

Now, the above table is not in 2NF because there is partial dependency i.e. non key is dependent on part of but not the entire composite key. here, Ticket_number, Flight_number and Total_amount depends on passenger_id but has no connection with Reservation_id. Thus, non-key columns Ticket_number, Flight_number and Total_amount do not depend on entire composite key.

To remove the partial dependencies in manufacturer table,the company table is created.

Passenger

Primary Key			
Passenger_id	Ticket_number	Flight_number	Total_ammount

Reservations

Primary Key	Primary_Key
Reservation_id	Passenger_id

- No, partial dependencies- it is in 2NF.
- No, transitive dependencies- it is in 3NF.

3. Create Database Tables

a. Construct CREATE statements for each table

```
CREATE TABLE flight
  (
    "Flight Number" VARCHAR(50) PRIMARY KEY,
    "Aircraft"      VARCHAR(30),
    "Source"        VARCHAR(30),
    "Destinantion"  VARCHAR(30),
    "Air-Line"      VARCHAR(30),
    “Flight time” VARCHAR(30)
  );

CREATE TABLE flight_history
(
  record_number    NUMBER(5) PRIMARY KEY,
  passenger_id     NUMBER(5) NOT NULL,
  flights_taken    NUMBER(30) ,
  travel_count     NUMBER(30)
 );

CREATE TABLE reservation
(
    reservation_id    NUMBER(5) PRIMARY KEY,
    passenger_id      NUMBER(5) NOT NULL,
    ticket_number     NUMBER(5) NOT NULL,
    flight_number     NUMBER(5) NOT NULL,
    total_amount      NUMBER(20)
    );

CREATE TABLE passenger
(
   passenger_id     NUMBER(5) PRIMARY KEY,
   first_name       VARCHAR2(15) NOT NULL,
   last_name        VARCHAR2(15) NOT NULL,
   phone_number     NUMBER(15)
     );

CREATE TABLE flight_ticket
(
    ticket_number    NUMBER(5) PRIMARY KEY,
    seat_number      VARCHAR(5) NOT NULL,
    flight_class     VARCHAR(20) NOT NULL,
    ticket_cost      NUMBER(15) NOT NULL,
     travel_date      DATE
     );
```

b. Construct INSERT statements and populate each table with at least 3 rows

```
INSERT INTO flight
("Flight Number", "Aircraft", "Source", "Destinantion", "Air-Line", "Flight time")
VALUES ('11', 'Boeing 385', 'Paris', 'Mumbai', 'Delta Airlines','9');
INSERT INTO flight
("Flight Number", "Aircraft", "Source", "Destinantion", "Air-Line","Flight time")
VALUES ('13', 'Boeing 380', 'Delhi', 'Toronto', 'Delta Airlines', '16');
INSERT INTO flight
("Flight Number", "Aircraft", "Source", "Destinantion", "Air-Line","Flight time")
VALUES ('17', 'Boeing 400', 'Mumbai', 'Amstradam', 'Air France', '9');
INSERT INTO flight
("Flight Number", "Aircraft", "Source", "Destinantion", "Air-Line","Flight time")
VALUES ('12', 'Boeing 454', 'Amstradam', 'Toronto', 'Air France','8');
INSERT INTO flight
("Flight Number", "Aircraft", "Source", "Destinantion", "Air-Line", "Flight time")
VALUES ('15', 'Boeing 777', 'Hong Kong', 'San Fransisco', 'United', '8');
INSERT INTO flight
("Flight Number", "Aircraft", "Source", "Destinantion", "Air-Line", "Flight time")
VALUES ('14', 'Boeing 777-300ER', 'Toronto', 'London', 'Air India','15');
INSERT INTO flight
("Flight Number", "Aircraft", "Source", "Destinantion", "Air-Line", "Flight time")
VALUES ('18', 'Boeing 787-8', 'London', 'Delhi', 'Air India','16');
INSERT INTO flight
("Flight Number", "Aircraft", "Source", "Destinantion", "Air-Line", "Flight time")
VALUES ('109', 'Boeing 787', 'Chicago', 'Barcelona', 'American Airlines','6');
INSERT INTO flight
("Flight Number", "Aircraft", "Source", "Destinantion", "Air-Line", "Flight time")
VALUES ('108', 'Embraer 175', 'Toronto', 'Chicago', 'American Airlines','3');
INSERT INTO flight
("Flight Number", "Aircraft", "Source", "Destinantion", "Air-Line","Flight time")
VALUES ('104', 'Etihad Airways 211', 'Delhi', 'Toronto', 'Virgin Australia','16');

INSERT INTO reservation
(reservation_id, passenger_id, ticket_number, flight_number, total_amount)
VALUES(01,1004,5120,11,300);
INSERT INTO reservation
(reservation_id,passenger_id,ticket_number,flight_number,total_amount)
VALUES(02,1007,5100,13,500);
INSERT INTO reservation
(reservation_id,passenger_id,ticket_number,flight_number,total_amount)
VALUES(03,1005,5170,17,900);
```

```
INSERT INTO reservation
(reservation_id,passenger_id,ticket_number,flight_number,total_amount)
VALUES(04,1009,5110,12,1200);
INSERT INTO reservation
(reservation_id,passenger_id,ticket_number,flight_number,total_amount)
VALUES(05,1006,5180,15,700);
INSERT INTO reservation
(reservation_id,passenger_id,ticket_number,flight_number,total_amount)
VALUES(06,1002,5160,14,900);
INSERT INTO reservation
(reservation_id,passenger_id,ticket_number,flight_number,total_amount)
VALUES(07,1001,5190,18,1100);
INSERT INTO reservation
(reservation_id,passenger_id,ticket_number,flight_number,total_amount)
VALUES(08,1003,5160,109,700);
INSERT INTO reservation
(reservation_id,passenger_id,ticket_number,flight_number,total_amount)
VALUES(09,1000,5140,108,1500);
INSERT INTO reservation
(reservation_id,passenger_id,ticket_number,flight_number,total_amount)
VALUES(10,1008,5130,104,800);

INSERT INTO passenger
(passenger_id,first_name,last_name,phone_number)
VALUES(1000,'Daenerys','Targaryen',6478303456);
INSERT INTO passenger
(passenger_id,first_name,last_name,phone_number)
VALUES(1001,'Jon','Snow',5163140745);
INSERT INTO passenger
(passenger_id,first_name,last_name,phone_number)
VALUES(1002,'Tyrion','Lannister',4163400234);
INSERT INTO passenger
(passenger_id,first_name,last_name,phone_number)
VALUES(1003,'Arya','Stark',3455780942);
INSERT INTO passenger
(passenger_id,first_name,last_name,phone_number)
VALUES(1004,'Cersei','Lannister',6473102167);
INSERT INTO passenger
(passenger_id,first_name,last_name,phone_number)
VALUES(1005,'Petyr','Baelish',5193003764);
INSERT INTO passenger
(passenger_id,first_name,last_name,phone_number)
VALUES(1006,'Jaime','Lannister',4163047045);
INSERT INTO passenger
(passenger_id,first_name,last_name,phone_number)
VALUES(1007,'Samwell','Tarly',5193154704);
```

```
INSERT INTO passenger
(passenger_id,first_name,last_name,phone_number)
VALUES(1008,'Eddard','Stark',6479925792);
INSERT INTO passenger
(passenger_id,first_name,last_name,phone_number)
VALUES(1009,'Joffrey','Baratheon',4162684567);

INSERT INTO flight_ticket
(ticket_number,seat_number,flight_class,ticket_cost,travel_date)
VALUES(5100,'K32','economy',300,'25-04-19');
INSERT INTO flight_ticket
(ticket_number,seat_number,flight_class,ticket_cost,travel_date)
VALUES(5110,'L27','buisness',900,'4-06-19');
INSERT INTO flight_ticket
(ticket_number,seat_number,flight_class,ticket_cost,travel_date)
VALUES(5120,'M15','first class',1500,'3-02-19');
INSERT INTO flight_ticket
(ticket_number,seat_number,flight_class,ticket_cost,travel_date)
VALUES(5130,'A15','buisness',800,'9-01-19');
INSERT INTO flight_ticket
(ticket_number,seat_number,flight_class,ticket_cost,travel_date)
VALUES(5140,'R10','economy',400,'5-08-18');
INSERT INTO flight_ticket
(ticket_number,seat_number,flight_class,ticket_cost,travel_date)
VALUES(5150,'E21','economy',500,'2-09-18');
INSERT INTO flight_ticket
(ticket_number,seat_number,flight_class,ticket_cost,travel_date)
VALUES(5160,'S15','economy',450,'9-08-18');
INSERT INTO flight_ticket
(ticket_number,seat_number,flight_class,ticket_cost,travel_date)
VALUES(5170,'E19','first_class',1200,'25-11-18');
INSERT INTO flight_ticket
(ticket_number,seat_number,flight_class,ticket_cost,travel_date)
VALUES(5180,'B12','business',900,'20-12-18');
INSERT INTO flight_ticket
(ticket_number,seat_number,flight_class,ticket_cost,travel_date)
VALUES(5190,'C17','economy',400,'20-11-18');

INSERT INTO flight_history
(record_number,passenger_id,flights_taken,travel_count)
VALUES(1,1005,2,1);
INSERT INTO flight_history
(record_number,passenger_id,flights_taken,travel_count)
VALUES(2,1001,4,2);
INSERT INTO flight_history
```

```
(record_number,passenger_id,flights_taken,travel_count)
VALUES(3,1007,3,2);
INSERT INTO flight_history
(record_number,passenger_id,flights_taken,travel_count)
VALUES(4,1003,5,3);
INSERT INTO flight_history
(record_number,passenger_id,flights_taken,travel_count)
VALUES(5,1008,3,2);
```

4. Constraints:

a. Identify Business Rules/Database Constraints.

NOT NULL - Ensures that a column cannot have a NULL value
UNIQUE - Ensures that all values in a column are different
PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
FOREIGN KEY - Uniquely identifies a row/record in another table
CHECK - Ensures that all values in a column satisfies a specific condition
DEFAULT - Sets a default value for a column when no value is specified
INDEX - Used to create and retrieve data from the database very quickly

Foreign Key:

```
ALTER TABLE flight_history
ADD FOREIGN KEY (passenger_id) REFERENCES passenger(passenger_id);
```

Check:

```
ALTER TABLE flight
ADD CHECK (source_city='Delhi');
```

b. Implement the constraints into the database creation statements.

```
CREATE TABLE flight_ticket(
        ticket_number    NUMBER(5) PRIMARY KEY,
        seat_number      VARCHAR(5) NOT NULL,
        flight_class     VARCHAR(20) NOT NULL,
        ticket_cost      NUMBER(15) NOT NULL,
        travel_date      DATE );
```

Part 2:

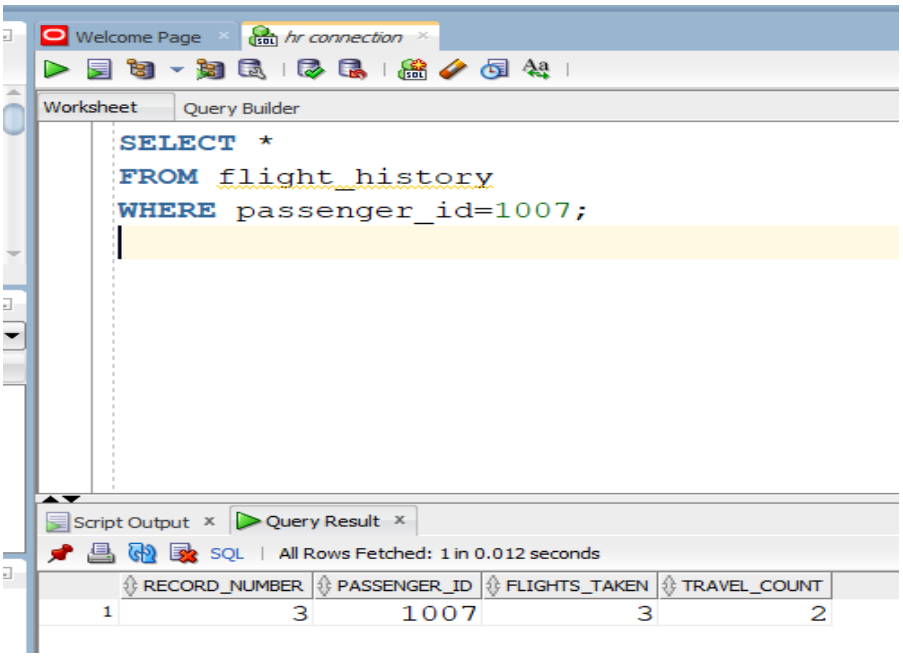
Queries:
Implement queries for following:

1) Data entry into tables that you have created.

----------Done Above----------

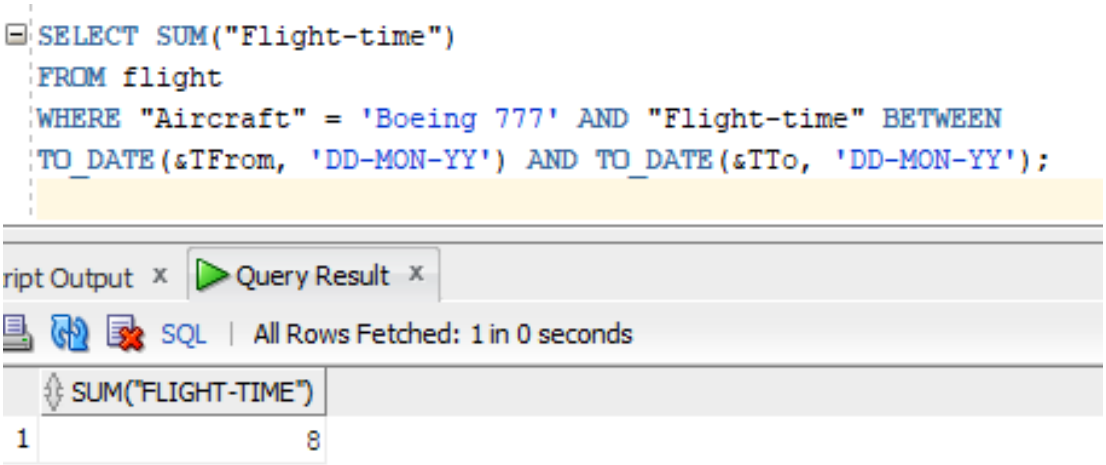
2) Travel history of a specific passenger.

```
SELECT *
FROM flight_history
WHERE passenger_id=1007;
```



3) Total hours that a specific aircraft has served during a specific time interval.

```
SELECT SUM("Flight-time")
FROM flight
WHERE "Aircraft" = 'Boeing 777' AND "Flight-time"
BETWEEN
TO_DATE(&TFrom, 'DD-MON-YY') AND
TO_DATE(&TTo, 'DD-MON-YY');
```



4) Total number of aircrafts belonging to a specific airline.

```
SELECT COUNT("Aircraft")
FROM flight
WHERE "Flight-time" BETWEEN TO_DATE(&TFrom,
'DD-MON-YY')
```

AND TO_DATE(&TTo, 'DD-MON-YY');

```
SELECT COUNT("Aircraft")
FROM flight
WHERE "Flight-time" BETWEEN TO_DATE(&TFrom, 'DD-MON-YY')
AND TO_DATE(&TTo, 'DD-MON-YY');
```

ript Output × | Query Result ×

SQL | All Rows Fetched: 1 in 0 seconds

	COUNT("AIRCRAFT")
1	10

5) Total number of hours that a specific passenger has travelled during a specific time interval.

SELECT SUM(f."Flight time"), p.passenger_id
FROM flight f
JOIN reservation r
ON f."Flight Number" = r.flight_number
JOIN passenger p
ON r.passenger_id = p.passenger_id
HAVING p.passenger_id = 1007 AND BETWEEN
TO_DATE(&TFrom, 'DD-MON-YY') AND
TO_DATE(&TTo, 'DD-MON-YY');
GROUP BY p.passenger_id;

```
SELECT SUM(f."Flight time"), p.passenger_id
    FROM flight f
    JOIN reservation r
    ON f."Flight Number" = r.flight_number
   JOIN passenger p
   ON r.passenger_id = p.passenger_id
   HAVING p.passenger_id = 1007 AND "Flight-time" BETWEEN TO_DATE(&TFr
   GROUP BY p.passenger_id;
```

Script Output | Query Result

SQL | All Rows Fetched: 1 in 0 seconds

	SUM(F."FLIGHT-TIME")	PASSENGER_ID
1	16	1007

6) Total number of hours that a specific aircraft has served during a specific time interval.

SELECT SUM(“Flight time”)
FROM flight
WHERE “Aircraft” = 'Boeing 777' AND WHERE "Flight-time"
BETWEEN TO_DATE(&TFrom, 'DD-MON-YY')
AND TO_DATE(&TTo, 'DD-MON-YY');

```
SELECT SUM("Flight time")
FROM flight
WHERE "Aircraft" = 'Boeing 777' AND WHERE "Flight-time"
BETWEEN TO_DATE(&TFrom, 'DD-MON-YY')
AND TO_DATE(&TTo, 'DD-MON-YY');
```

Script Output | Query Result

SQL | All Rows Fetched: 1 in 0 seconds

	SUM("FLIGHT-TIME")
1	8

7) List of all passengers who flew to a specific city during a specific time interval.

```
SELECT *
FROM passenger
Where "Flight-time"
BETWEEN TO_DATE(&TFrom, 'DD-MON-YY')
AND TO_DATE(&TTo, 'DD-MON-YY');
```

```
SELECT *
FROM passenger
Where "Flight-time"
BETWEEN TO_DATE(&TFrom, 'DD-MON-YY')
AND TO_DATE(&TTo, 'DD-MON-YY');
```

cript Output × | Query Result ×

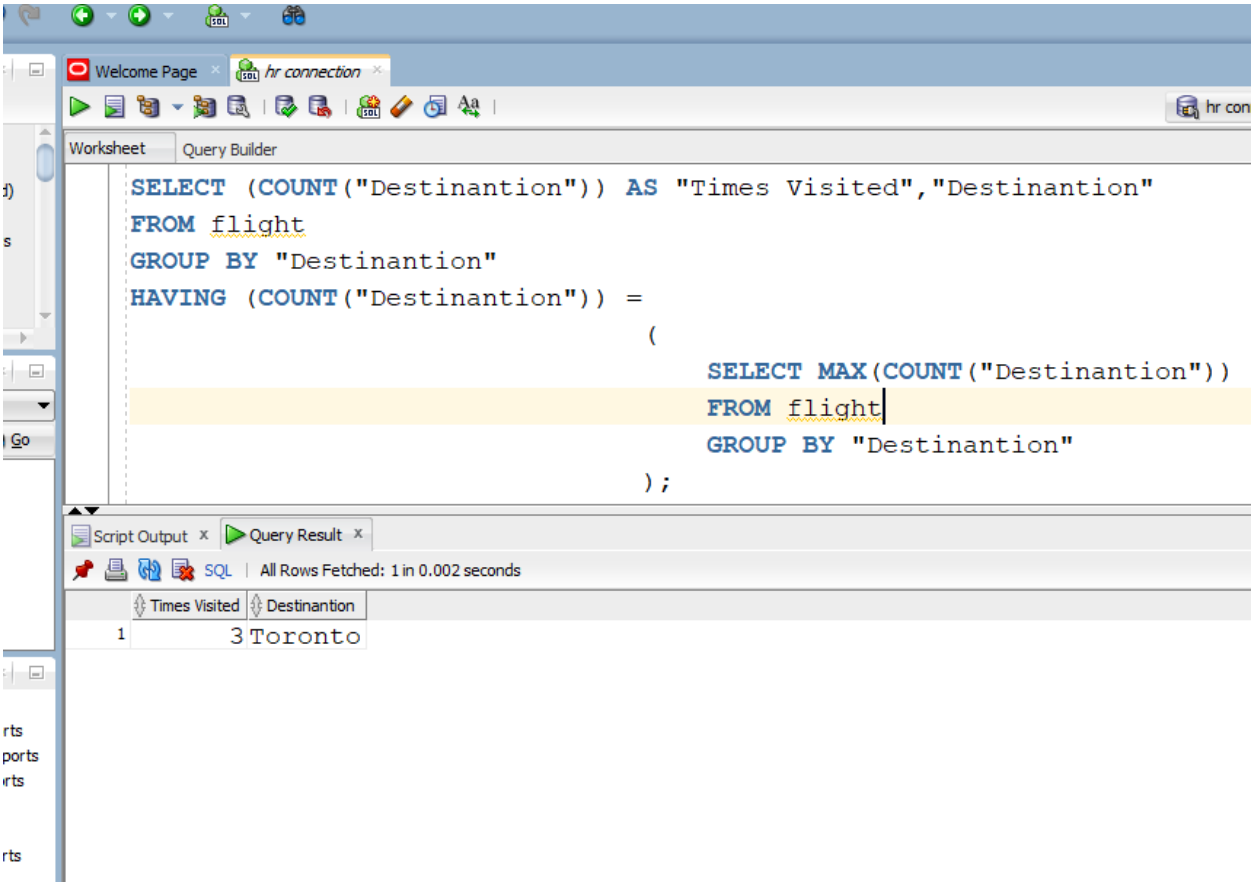
SQL | All Rows Fetched: 10 in 0 seconds

	PASSENGER_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBER
1	1000	Daenerys	Targaryen	6478303456
2	1001	Jon	Snow	5163140745
3	1002	Tyrion	Lannister	4163400234
4	1003	Arya	Stark	3455780942
5	1004	Cersei	Lannister	6473102167
6	1005	Petyr	Baelish	5193003764
7	1006	Jaime	Lannister	4163047045
8	1007	Samwell	Tarly	5193154704
9	1008	Eddard	Stark	6479925792
10	1009	Joffrey	Baratheon	4162684567

8) Most visited city during the last month.

```
SELECT (COUNT("Destinantion")),"Destinantion"
FROM flight
GROUP BY "Destinantion"
HAVING (COUNT("Destinantion")) =
                    (
```

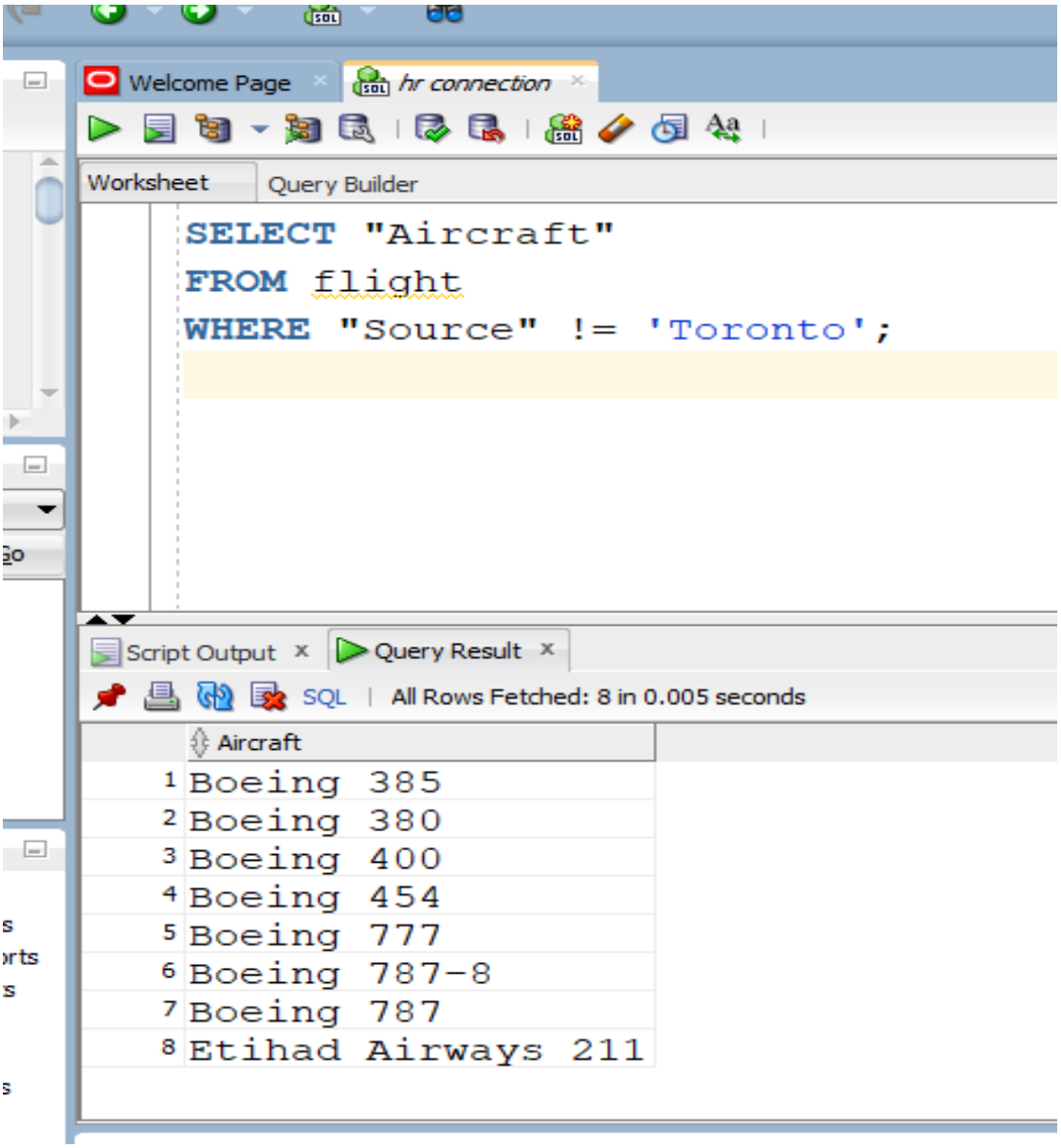
```
                    SELECT
MAX(COUNT("Destinantion"))
                    FROM flight
                    GROUP BY "Destinantion"
                );
```



9) List of aircrafts that have **not** been in used from a specific source location.

```
SELECT "Aircraft"
FROM flight
```

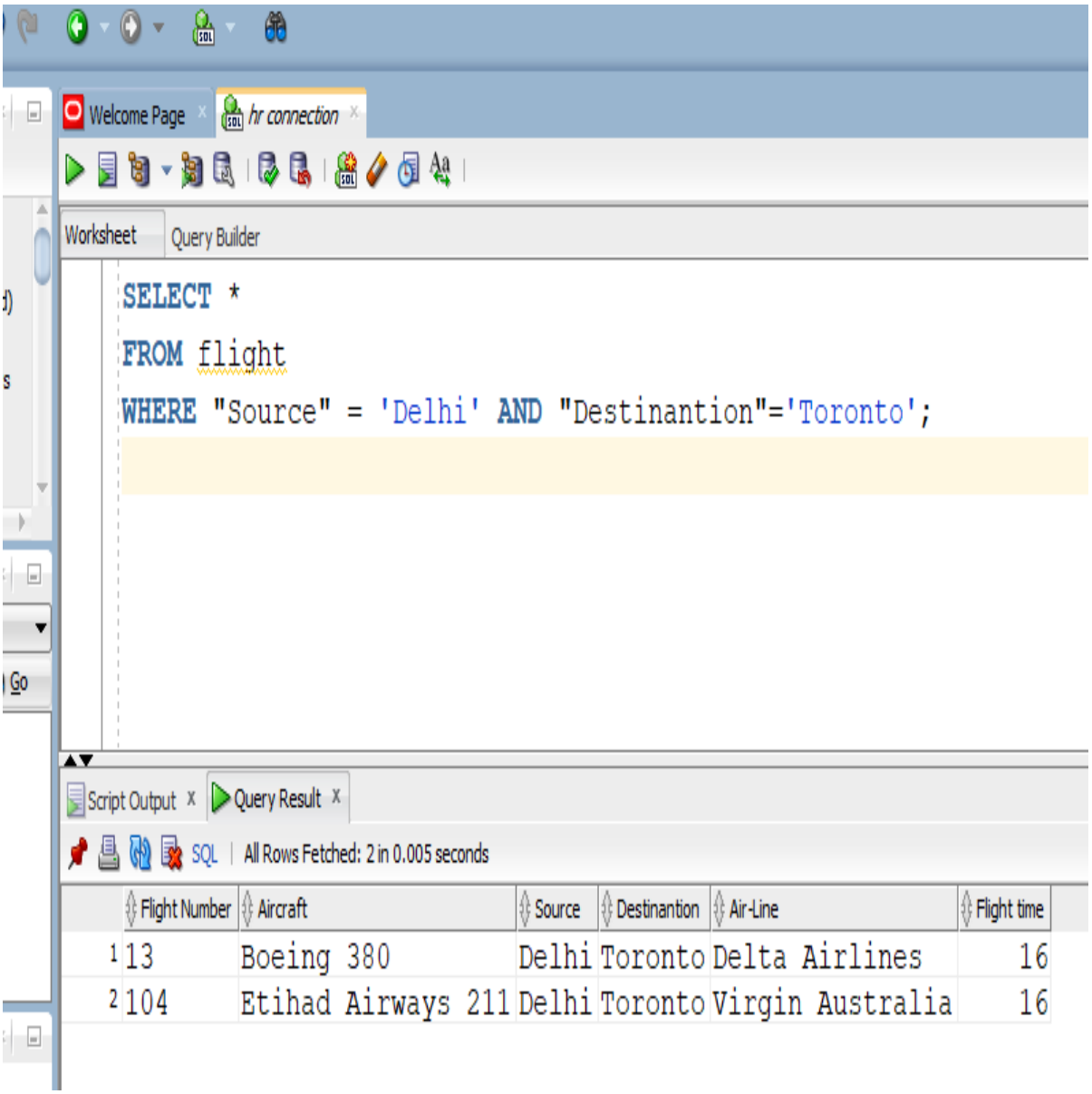
WHERE "Source" != 'Toronto';



10) List of airlines that run flight from a specific source to a destination.

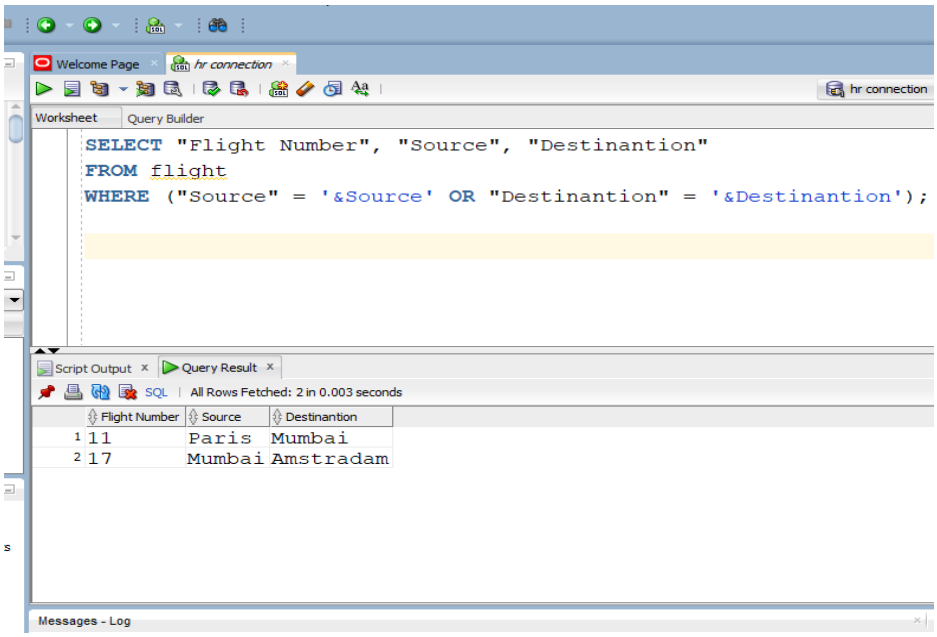
SELECT *
FROM flight

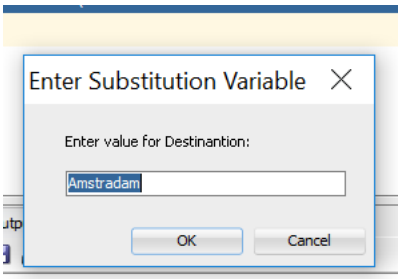
WHERE “Source” = 'Toronto' AND “Destination”='Delhi';

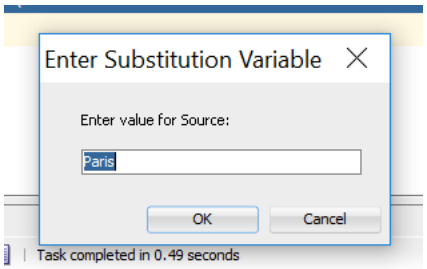


11) The list of all options that a passenger can have when travelling from a source to a destination. This includes a connecting flight, for instance, a passenger is travelling from Toronto to Dehli and there is no direct flight, therefore, you have to find the options for this passenger.

SELECT "Flight Number", "Source", "Destinantion"
FROM flight
WHERE ("Source" = '&Source' OR "Destinantion" = '&Destinantion');







12)What is the minimum number of hours that it will take for a passenger to travel from a source city to a destination city. Again, consider the connecting flights as mentioned in item number 9, e.g. travelling from Toronto to Dehli.

```
SELECT MIN("Flight time")
FROM flight
WHERE "Source" = 'Toronto' AND
"Destinantion"='Chicago';
```