

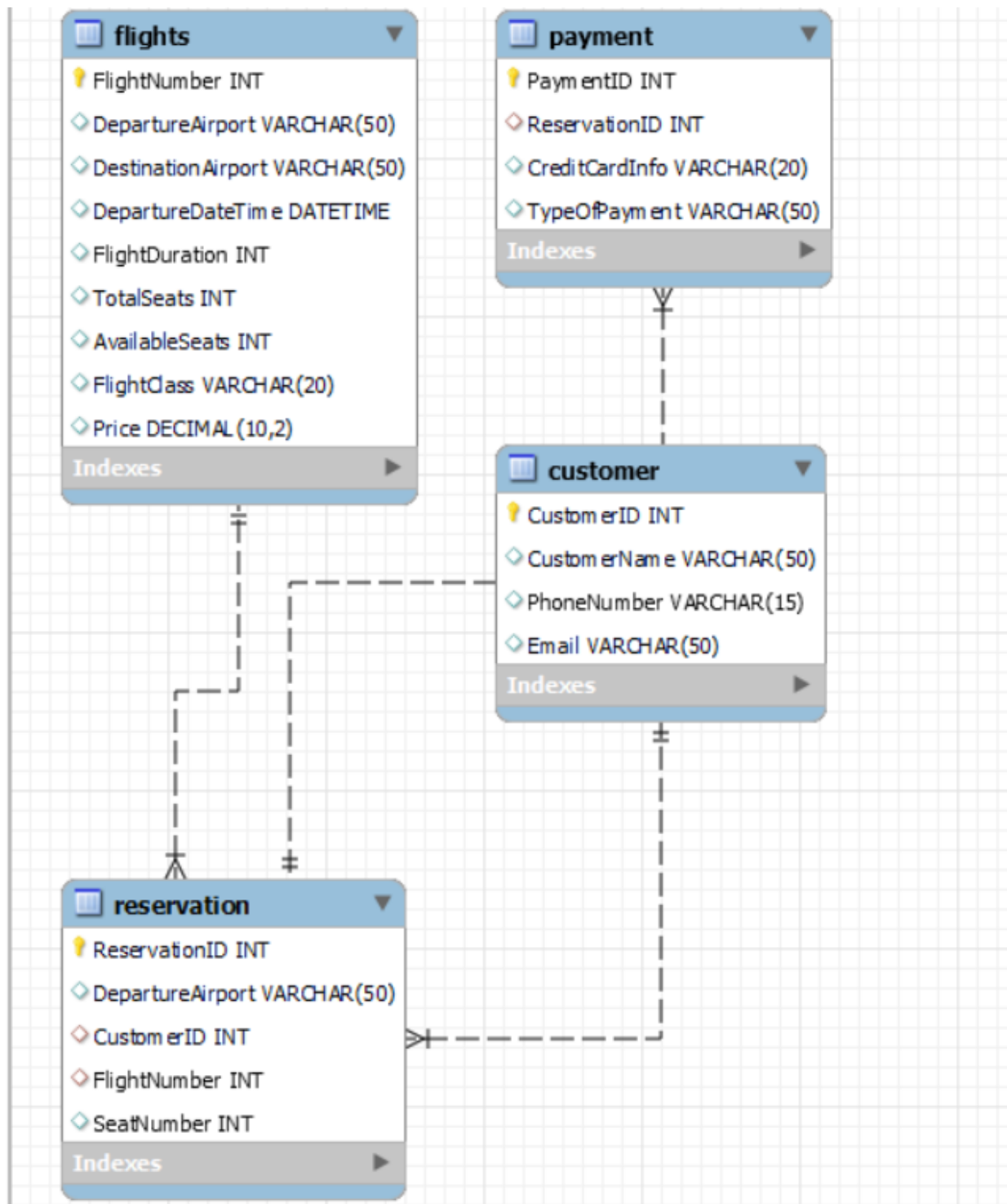
Term Project: Report 3

Database Structures

Iacopo Nohea Lenzi

Jasenia Guerra

Q0. Schema Diagram



Q1. Creating Six Queries:

Queries selected from HW2:

- 1) SELECT * FROM Flights WHERE DepartureAirport = 'JFK' UNION SELECT * FROM Flights WHERE DestinationAirport = 'JFK';
- 2) SELECT Customer.*, Reservation.* FROM Customer, Reservation WHERE Customer.CustomerID = Reservation.CustomerID;
- 3) SELECT DepartureAirport FROM Flights UNION SELECT DestinationAirport FROM Flights;
- 4) SELECT SUM(AvailableSeats) FROM Flights WHERE FlightNumber = '101';
- 5) SELECT * FROM Customer WHERE CustomerID IN (SELECT CustomerID FROM Reservation WHERE Reservationid > '2')
- 6) SELECT * FROM Reservation WHERE ReservationID IN (SELECT ReservationID FROM Payment WHERE typeofpayment > 'Paypal')

Python for selected Queries:

```
import datetime
import mysql.connector

# Establish a database connection
cnx = mysql.connector.connect(user='root', password='Database123',
host='localhost', database='test')
cursor = cnx.cursor()

# Define your SQL commands with host variables
queries_with_variables = {
    'query1': {
        'sql': "SELECT * FROM Flights WHERE DepartureAirport = %s UNION
SELECT * FROM Flights WHERE DestinationAirport = %s",
        'params': ('JFK', 'JFK')
    },
    'query2': {
        'sql': "SELECT Customer.*, Reservation.* FROM Customer, Reservation
WHERE Customer.CustomerID = Reservation.CustomerID AND Customer.CustomerID
= %s",
```

```

        'params': (5,)
    },
    'query3': {
        'sql': "SELECT DepartureAirport FROM Flights WHERE DepartureAirport =
%s UNION SELECT DestinationAirport FROM Flights WHERE DestinationAirport =
%s",
        'params': ('LAX', 'FLL')
    },
    'query4': {
        'sql': "SELECT SUM(AvailableSeats) FROM Flights WHERE FlightNumber
= %s",
        'params': ('101',)
    },
    'query5': {
        'sql': "SELECT * FROM Customer WHERE CustomerID IN (SELECT
CustomerID FROM Reservation WHERE Reservationid > %s)",
        'params': ('2',)
    },
    'query6': {
        'sql': "SELECT * FROM Reservation WHERE ReservationID IN (SELECT
ReservationID FROM Payment WHERE typeofpayment > %s)",
        'params': ('PayPal',)
    }
}

```

```

# Execute each query with its corresponding host variables
for query_name, query_info in queries_with_variables.items():
    cursor.execute(query_info['sql'], query_info['params']) # Execute the query
with host variables
    if query_name in ['query1', 'query2', 'query3', 'query5', 'query6']:
        # Fetch and print results for SELECT queries
        results = cursor.fetchall()
        print(f"Results for {query_name}:")
        for row in results:
            print(row)
    elif query_name == 'query4':
        # For the aggregation query, fetch and print the result
        total_seats = cursor.fetchone()
        print(f"Results for query4:\n Total available seats for airline 'AA':
{total_seats[0]}")

```

```
# Close the cursor and the database connection
cursor.close()
cnx.close()
```

Output for Python:

```
Results for query1:
(101, 'JFK', 'LAX', datetime.datetime(2023, 12, 3, 8, 0), Decimal('5.24'), 150, 145, 'Economy', Decimal('600.72'))
(104, 'ATL', 'JFK', datetime.datetime(2023, 11, 10, 10, 0), Decimal('3.23'), 120, 115, 'First Class', Decimal('399.99'))
(512, 'MIA', 'JFK', datetime.datetime(2023, 10, 24, 17, 41), Decimal('2.54'), 162, 18, 'Economy', Decimal('86.00'))
Results for query2:
(5, 'Justin Bieber', '123-456-7891', 'justin.bieb@email.com', 1, 6, 'LAX', 5, 5042, 2)
Results for query3:
('LAX',)
('FLL',)
Results for query4:
Total available seats for airline 'AA': 145
Results for query5:
(4, 'Taylor Swift', '555-555-5555', 't.swift@email.com', 1)
(5, 'Justin Bieber', '123-456-7891', 'justin.bieb@email.com', 1)
(6, 'Beyonce Knowles', '954-104-1395', 'beyonce.k@email.com', 1)
Results for query6:
(8, 'ATL', 4, 1111, 4)
```

Q2. Implementing Procedure and Function

Procedure containing aggregate function:

```
DELIMITER //
```

```
CREATE PROCEDURE GetTotalSeats()
BEGIN
    SELECT SUM(AvailableSeats) AS TotalSeats FROM Flights;
END //
```

```
DELIMITER;
```

Function returning a value:

```
DELIMITER //
```

```
CREATE FUNCTION AverageFlightPrice() RETURNS DECIMAL(10,2)
```

```

DETERMINISTIC
READS SQL DATA
BEGIN
    DECLARE result DECIMAL(10,2);
    SELECT AVG(Price) INTO result FROM Flights;
    RETURN result;
END //

DELIMITER ;

```

Python using both procedures:

```
import mysql.connector
```

```
# Connect to the MySQL database
```

```
cnx = mysql.connector.connect(user='root', password='Database123', host='localhost',
database='test')
```

```
cursor = cnx.cursor()
```

```
# Call the stored procedure
```

```
try:
```

```
    cursor.callproc('GetTotalSeats')
```

```
    for result in cursor.stored_results():
```

```
        print("Total available seats from stored procedure:", result.fetchone()[0])
```

```
except mysql.connector.Error as err:
```

```
    print("An error occurred: {}".format(err))
```

```
# Call the function
```

```
try:
```

```
    cursor.execute("SELECT AverageFlightPrice()")
```

```
    average_price = cursor.fetchone()[0]
```

```
    print("Average flight price:", average_price)
```

```
except mysql.connector.Error as err:
```

```
    print("Error occurred: {}".format(err))
```

```
# Close the cursor and database connection
```

```
cursor.close()
```

```
cnx.close()
```

Output for Python:

```
Total available seats from stored procedure: 700
Average flight price: 307.46
```

Q3. Implementing HW1 tasks

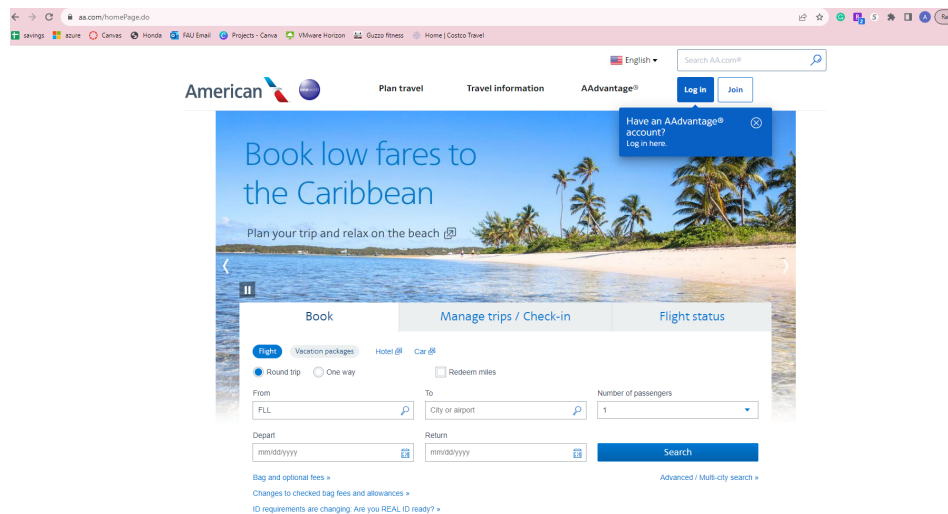
Part 1

For the implementation of tasks described in HW1 we have decided to go with the following.

- **Search Criteria:** In this feature the user will be able to enter the criteria for which they wish to choose a flight. There will be a round trip or single flight option, dates for departure and/or return option, a departure airport option, a destination airport option, and a number of passengers option.
- **Seat Selection:** This one is pretty straight forward, after having entered the search criteria and having selected the flight(s), the passenger will be allowed to pick whether they would like to choose a specific seat and cabin class.
- **Customer Details:** Finally, after having completing all the previous selections a prompt will pop up asking the customer to enter their personal information and payment method for their purchase.

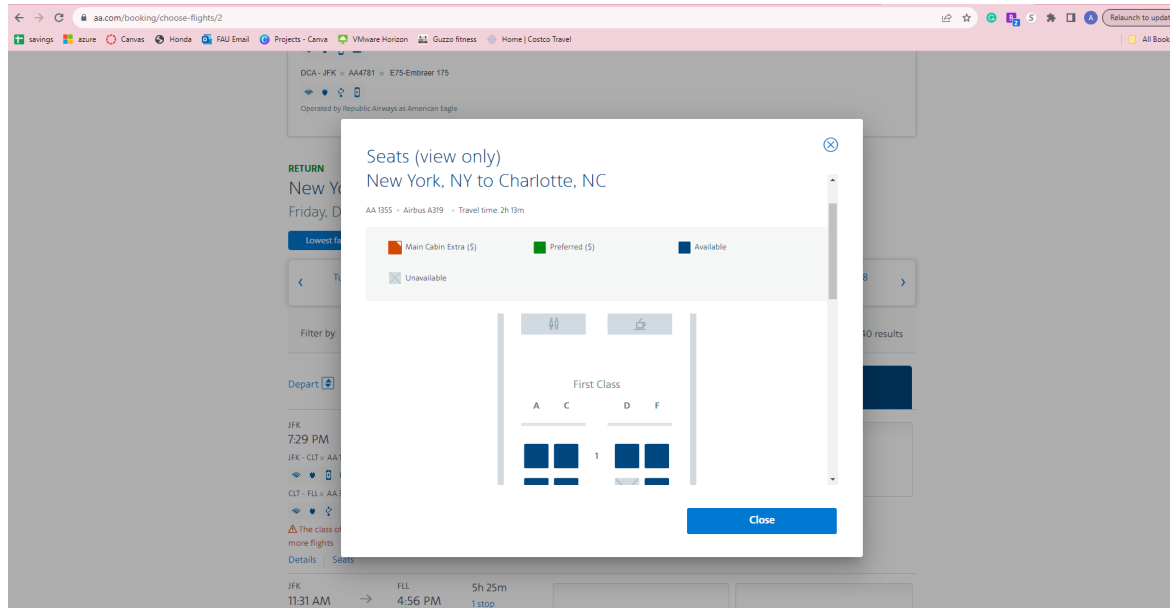
Below are the webpages that American Airlines currently uses:

Search Criteria:



The screenshot displays the American Airlines website's flight search interface. At the top, there's a navigation bar with the American Airlines logo, links for 'Plan travel', 'Travel information', and 'AAdvantage', along with 'Log in' and 'Join' buttons. A search bar is also present. Below the navigation bar, a large banner features the text 'Book low fares to the Caribbean' with a background image of a tropical beach. Underneath the banner, there's a section titled 'Book' with tabs for 'Flight', 'Vacation packages', 'Hotel', and 'Car'. The 'Flight' tab is selected. Within the 'Flight' section, there are options for 'Round trip' (selected) and 'One way', and a checkbox for 'Redeem miles'. The search form includes fields for 'From' (filled with 'FLL'), 'To' (with a placeholder 'City or airport'), 'Depart' (filled with 'mm/dd/yyyy'), and 'Return' (filled with 'mm/dd/yyyy'). A 'Number of passengers' dropdown is set to '1'. A blue 'Search' button is located at the bottom right of the form. Below the search form, there are links for 'Bag and optional fees', 'Changes to checked bag fees and allowances', and 'ID requirements are changing. Are you REAL ID ready?'. At the bottom right, there's a link for 'Advanced / Multi-city search'.

Seat and cabin Selection:



Customer Details:

Passenger details

Enter names as printed on each passenger's government-issued photo ID. [TSA Secure Flight rules](#)

We collect your personal data in accordance with applicable laws and regulations. Read how we use and protect your personal information. [American Airlines privacy policy](#)

Passenger 1

(*) Required

First name Middle name Last name

Loyalty program Loyalty number

Date of birth

Gender Country / region of residence

[Add Known Traveler number](#) [Add redress number](#) [Add special assistance](#)

Trip contact

(*) Required

Primary email Confirm primary email

Primary phone

Part 2

import mysql.connector

Connect to the MySQL database

cnx = mysql.connector.connect(user='root', password='', host='localhost',

database='P2_Schema')

cursor = cnx.cursor()


```

def search_criteria():
    round_trip = input("Round trip (yes/no): ")
    if round_trip == "yes":
        departure_date = input("Departure date (YYYY-MM-DD): ")
        return_date = input("Return date (YYYY-MM-DD): ")
    elif round_trip == "no":
        departure_date = input("Departure date (YYYY-MM-DD): ")
    departure_airport = input("Departure airport: ")
    destination_airport = input("Destination airport: ")
    num_passengers = int(input("Number of passengers: ")) # Convert to integer

    #SQL query to search for flights
    query = "SELECT * FROM Flights WHERE DepartureDateTime = %s AND
DepartureAirport = %s AND DestinationAirport = %s"
    cursor.execute(query, (departure_date, departure_airport, destination_airport))
    flights = cursor.fetchall()

```

```

print("\nAvailable Flights:\n")
for flight in flights:
    FlightNumber, DepartureAirport, DestinationAirport, DepartureDateTime,
    FlightDuration, TotalSeats, AvailableSeats, FlightClass, Price = flight
    print(f"Flight Number: {FlightNumber}")
    print(f"Departure Date and Time: {DepartureDateTime}")
    print(f"Departure Airport: {DepartureAirport}")
    print(f"Destination Airport: {DestinationAirport}")
    print(f"Available Cabin Class: {FlightClass}")
    print(f"Price: ${Price}")
print("\n*50) # Separator for better readability

```

Add more code to handle round trip, return date, etc.

```

def seat_selection():
    choose_seat = input("Do you want to choose a seat (yes/no): ")
    cabin_class = input("Cabin class (economy/business/first): ")

    if choose_seat.lower() == 'yes':
        # Add code for seat selection

```

```

        seat_number = input("Enter seat number: ")
        print(f"Seat {seat_number} selected in {cabin_class} class.\n")
    else:
        print(f"You have chosen {cabin_class} class without seat selection.\n")
    print("="*50) # Separator for better readability

def customer_details():
    name = input("Full name: ")
    email = input("Email address: ")
    phone = input("Phone number: ")
    flightPurchased = 1

    # SQL query to save customer details
    query = "INSERT INTO Customer (CustomerName, Email, PhoneNumber,
FlightPurchased) VALUES (%s, %s, %s, %s)"
    cursor.execute(query, (name, email, phone, flightPurchased))
    cnx.commit()
    print("Customer details saved successfully.")

# Example usage:
search_criteria()
seat_selection()
customer_details()

# Close the database connection
cursor.close()
cnx.close()

```

Part 3

```
● (base) iacopolenzi@Iacopos-MacBook-Air ~ % /Users/iacopolenzi/opt/anaconda3/bin/python /Users/iacopolenzi/Desktop/
HW3.py
Round trip (yes/no): no
Departure date (YYYY-MM-DD): 2023-12-03 08:00:00
Departure airport: JFK
Destination airport: LAX
Number of passengers: 1

Available Flights:

Flight Number: 101
Departure Date and Time: 2023-12-03 08:00:00
Departure Airport: JFK
Destination Airport: LAX
Available Cabin Class: Economy
Price: $600.72
=====
Do you want to choose a seat (yes/no): Yes
Cabin class (economy/business/first): Economy
Enter seat number: 32F
Seat 32F selected in Economy class.
=====
Full name: Iacopo Nohea Lenzi
Email address: ilenzi2017@fau.edu
Phone number: 1234567890
Customer details saved successfully.
```