BookAndFly!

Flight Booking System (Effective and Accuracy Opportunity)

Phuc Nguyen, Suheaf Hussein, Isaac Keen, Karl Lee
INFO 330 Project Phase 1 - Group 12

Project Proposal

Even with its wide range of applications the airline industry struggles to give its clients a simple and effective booking experience. Numerous airlines each with its own platforms and procedures add to a chaotic/hectic environment that frequently results in mistakes, hold-ups, and poor user experience. We want to present a single platform for all flight bookings in an attempt to transform this part of the airline industry. which will offer insights into client preferences and travel history and promise a more seamless, error-free booking experience by consolidating flight options from multiple carriers into a single system.

Problem Identification:

- Multiple platforms: Inefficiencies and inconsistencies in booking are caused by a variety of airline platforms.
- Booking errors and delays: Different systems may cause overlapping, or misbookings.
- Data security and integrity: discrepancies in data and possible loss can be made more likely by disorganized platforms.

Information Needs:

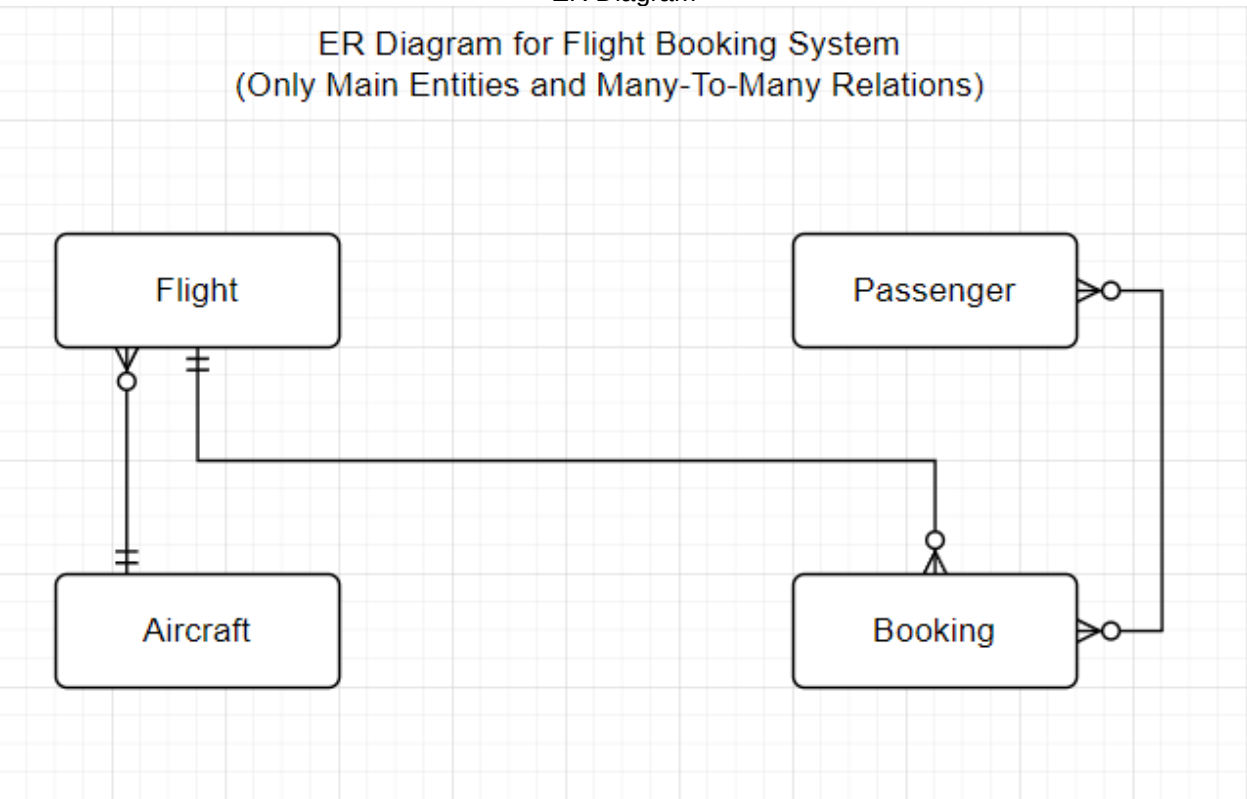
- Unified Traveler Profiles: An all-encompassing perspective of every passenger's flight records, inclinations, and input to improve service customization.
- Consolidated Flight Records: A single system that keeps track of every flight reservation, cancellation, and user-modified change.
- Real-time Flight Availability: To guarantee correct bookings, there is instant cross-checking of flight availability across airlines.
- Data integrity and protection from any breaches and data loss are ensured by backup **and security protocols.**

Entities List:

- Passenger: Details such as personal information, frequent flyer IDs, flight history, and preferences.
- Bookings: Capturing data related to all flight services booked by travelers.

- Flights: Comprehensive details about available flights, schedules, seating configurations, pricing, and any special provisions or offerings.
- Airlines: This table would encompass all the data related to the airlines that are part of the platform. This is vital since flights are connected to specific airlines, and different airlines might have varying rules, offers, loyalty programs, or partnerships.
- Aircraft foreign table
- Multiple flights foreign table

ER Diagram



Business Rules

1. Each Passenger can have 0 to many Bookings
2. Each Booking can have 0 to many passengers (family/multiple people)
3. Each Flight can have 0 to many Bookings
4. Each Booking is associated with 1 flight
5. Each Flight only associated with 1 aircraft
6. Each Aircraft can have multiple flights

Relational Schema:

1. Passenger(ID INT PRIMARY KEY, Name VARCHAR, FrequentFlyerID VARCHAR, ContactInfo VARCHAR, Preferences VARCHAR)

2. Booking(ID INT PRIMARY KEY, PassengerID INT, Date DATE, Status VARCHAR, TotalCost DECIMAL, FOREIGN KEY (PassengerID) REFERENCES Passenger(ID)

3. Flight(ID INT PRIMARY KEY, DepartureTime TIME, ArrivalTime TIME, Origin VARCHAR, Destination VARCHAR, Price DECIMAL

4. Airlines(ID INT PRIMARY KEY, Name VARCHAR, LoyaltyPrograms VARCHAR, PartnershipDetails VARCHAR)

5. Aircraft(ID INT PRIMARY KEY, Type VARCHAR, Capacity INT, AirlineID INT, FOREIGN KEY (AirlineID) REFERENCES Airlines(ID))

SQL Queries:

Note that the data we have is fairly limited and generated so there will be queries with less or no results due to amount and variance of the data available, however the queries will all return substantial information if given the data.

Name: Passenger Bookings

```
-- Lists all passengers along with the count of their bookings and total costs of the bookings--
WITH PassengerBookings AS (
    SELECT
        Passenger.ID AS PassengerID,
        Passenger.Name,
        COUNT(Booking.ID) AS BookingCount,
        SUM(Booking.TotalCost) AS TotalBookingCost
    FROM
        Passenger
        LEFT JOIN Booking ON Passenger.ID = Booking.PassengerID
    GROUP BY
        Passenger.ID, Passenger.Name
)

```

Results Messages

	PassengerID	Name	BookingCount	TotalBookingCost
1	1	John Doe	11	6350
2	2	Jane Smith	9	4800
3	3	Bob Johnson	9	5450
4	4	Alice White	8	4650
5	5	Charlie Brown	8	4850
6	6	Emily Davis	0	NULL
7	7	Frank Miller	0	NULL
8	8	Grace Turner	0	NULL
9	9	Henry Harris	0	NULL
10	10	Isabella Green	0	NULL
11	11	Jack Anderson	0	NULL
12	12	Katie Wilson	0	NULL
13	13	Liam Thomas	0	NULL
14	14	Mia Parker	0	NULL
15	15	Noah Wright	0	NULL

This query lists all the passengers in the database along with how many bookings they each have as well as the total costs of the bookings. The only passengers with bookings all have at least 8, with a minimum cost of 4650. The results imply for each passenger who books a flight, they will also book more (obviously not like real life).

Name: Average flight price for each Origin and Destination pair

```
18  WITH AvgFlightPrices AS (
19      SELECT
20          Origin,
21          Destination,
22          AVG(Price) AS AverageFlightPrice
23      FROM
24          Flight
25      GROUP BY
26          Origin, Destination
27  )
28
29  SELECT * FROM AvgFlightPrices;
30
```

Results Messages

	Origin	Destination	AverageFlightPrice
1	LHR	CDG	590
2	LAX	JFK	671
3	SFO	JFK	643
4	JFK	LAX	737
5	CDG	LHR	594
6	SFO	ORD	550
7	ORD	SFO	628

This query shows the average flight price for each pair of origin and destination. The most expensive being JFK to LAX and the cheapest SFO to ORD. This allows people to know which flights are expensive, not by the standard of date, but by specific location.

Name: High cost bookings

```
31  --Find passengers who have booked flights with costs higher than average costs of all bookings--
32  WITH AverageBookingCost AS (
33      SELECT
34          AVG(TotalCost) AS AvgTotalCost
35      FROM
36          Booking
37  )
38
39  SELECT
40      Passenger.ID AS PassengerID,
41      Passenger.Name,
42      Booking.TotalCost
43  FROM
44      Passenger
45      INNER JOIN Booking ON Passenger.ID = Booking.PassengerID
46      CROSS JOIN AverageBookingCost
47  WHERE
48      Booking.TotalCost > AverageBookingCost.AvgTotalCost;
49
```

Results Messages

	PassengerID	Name	TotalCost
1	1	John Doe	750
2	3	Bob Johnson	600
3	5	Charlie Brown	700
4	3	Bob Johnson	800
5	2	Jane Smith	650
6	4	Alice White	900
7	3	Bob Johnson	750
8	1	John Doe	600
9	2	Jane Smith	700
1…	5	Charlie Brown	800
1…	3	Bob Johnson	600
1…	5	Charlie Brown	700
1…	1	John Doe	800
1…	4	Alice White	600
1…	1	John Doe	750
1…	2	Jane Smith	900
1…	4	Alice White	650
1…	5	Charlie Brown	750

This query finds and lists all the passengers

```sql
--List all passengers who have a frequent flyer ID and their loyalty programs--
SELECT
    Passenger.ID AS PassengerID,
    Passenger.Name,
    Passenger.FrequentFlyerID,
    Airlines.Name AS AirlineName,
    Airlines.LoyaltyPrograms
FROM
    Passenger
    LEFT JOIN Booking ON Passenger.ID = Booking.PassengerID
    LEFT JOIN Flight ON Booking.ID = Flight.ID
    LEFT JOIN Airlines ON Flight.id = Airlines.ID
WHERE
    Passenger.FrequentFlyerID IS NOT NULL;

--Select total capacity and number of bookings for each aircraft type--
SELECT
    Aircraft.Type AS AircraftType,
```

Results | Messages

	PassengerID	Name	FrequentFlyerID	AirlineName	LoyaltyPrograms
1	1	John Doe	FF123	Delta Airlines	SkyMiles
2	1	John Doe	FF123	American Airlines	AAdvantage
3	1	John Doe	FF123	Singapore Airlines	KrisFlyer
4	1	John Doe	FF123	Qatar Airways	Privilege Club
5	1	John Doe	FF123	Etihad Airways	Etihad Guest
6	1	John Doe	FF123	*NULL*	*NULL*
7	1	John Doe	FF123	*NULL*	*NULL*
8	1	John Doe	FF123	*NULL*	*NULL*
9	1	John Doe	FF123	*NULL*	*NULL*
10	1	John Doe	FF123	*NULL*	*NULL*
11	1	John Doe	FF123	*NULL*	*NULL*
12	3	Bob Johnson	FF456	Air France	Flying Blue
13	3	Bob Johnson	FF456	Cathay Pacific	Marco Polo Club
14	3	Bob Johnson	FF456	KLM Royal Dutch	Flying Blue
15	3	Bob Johnson	FF456	*NULL*	*NULL*
16	3	Bob Johnson	FF456	*NULL*	*NULL*
17	3	Bob Johnson	FF456	*NULL*	*NULL*
18	3	Bob Johnson	FF456	*NULL*	*NULL*

```sql
--Select total capacity and number of bookings for each aircraft type--
SELECT
    Aircraft.Type AS AircraftType,
    Aircraft.Capacity AS TotalCapacity,
    COUNT(Booking.ID) AS NumberOfBookings
FROM
    Aircraft
    LEFT JOIN Airlines ON Aircraft.AirlineID = Airlines.ID
    LEFT JOIN Flight ON Airlines.ID = Flight.id
    LEFT JOIN Booking ON Flight.ID = Booking.ID
GROUP BY
    Aircraft.Type, Aircraft.Capacity;

--Total number of bookings for each airline, inclluding no bookings--
WITH AirlineBookings AS (
    SELECT
        Airlines.ID AS AirlineID,
```

Results | Messages

	AircraftType	TotalCapacity	NumberOfBookings
1	ATR 72	70	1
2	Bombardier CRJ-900	90	1
3	Embraer E190	100	1
4	Boeing 737	150	1
5	Airbus A320	180	1
6	Boeing 737 MAX	200	1
7	Boeing 757	200	1
8	Airbus A321neo	220	1
9	Airbus A330	250	1
10	Boeing 767	250	1
11	Boeing 787 Dreamliner	250	1
12	Airbus A350	300	1
13	Boeing 777	300	1
14	Boeing 747	400	1
15	Airbus A380	550	1

```sql
--Total number of bookings for each airline, including no bookings--
WITH AirlineBookings AS (
    SELECT
        Airlines.ID AS AirlineID,
        Airlines.Name AS AirlineName,
        COUNT(Booking.ID) AS BookingCount
    FROM
        Airlines
        LEFT JOIN Flight ON Airlines.ID = Flight.id
        LEFT JOIN Booking ON Flight.ID = Booking.ID
    GROUP BY
        Airlines.ID, Airlines.Name
)

SELECT * FROM AirlineBookings;

--List passengers and their total spending on flights--
WITH PassengerSpending AS (
   SELECT
      Passenger.ID AS PassengerID,
```

Results | Messages

	AirlineID	AirlineName	BookingCount
1	1	Delta Airlines	1
2	2	British Airways	1
3	3	American Airlines	1
4	4	Air France	1
5	5	Lufthansa	1
6	6	Emirates	1
7	7	Singapore Airlines	1
8	8	Qantas Airways	1
9	9	Cathay Pacific	1
10	10	Turkish Airlines	1
11	11	Qatar Airways	1
12	12	ANA All Nippon	1
13	13	KLM Royal Dutch	1
14	14	Virgin Atlantic	1
15	15	Etihad Airways	1

```sql
--List passengers and their total spending on flights--
WITH PassengerSpending AS (
    SELECT
        Passenger.ID AS PassengerID,
        Passenger.Name,
        SUM(Booking.TotalCost) AS TotalSpending
    FROM
        Passenger
        INNER JOIN Booking ON Passenger.ID = Booking.PassengerID
    GROUP BY
        Passenger.ID, Passenger.Name
)


SELECT * FROM PassengerSpending;

--Retrieve passengers with highest total cost of booking--
WITH MaxBookingCost AS (
    SELECT
        MAX(TotalCost) AS MaxTotalCost
```

Results Messages

	PassengerID	Name	TotalSpending
1	1	John Doe	6350
2	2	Jane Smith	4800
3	3	Bob Johnson	5450
4	4	Alice White	4650
5	5	Charlie Brown	4850

```sql
--Retrieve passengers with highest total cost of booking--
WITH MaxBookingCost AS (
    SELECT
        MAX(TotalCost) AS MaxTotalCost
    FROM
        Booking
)

SELECT
    Passenger.ID AS PassengerID,
    Passenger.Name,
    Booking.TotalCost
FROM
    Passenger
    INNER JOIN Booking ON Passenger.ID = Booking.PassengerID
    CROSS JOIN MaxBookingCost
WHERE
    Booking.TotalCost = MaxBookingCost.MaxTotalCost;

--Passengers who have booked round trip flights (flights with same or
```

Results Messages

	PassengerID	Name	TotalCost
1	4	Alice White	900
2	2	Jane Smith	900

```sql
--Passengers who have booked round trip flights (flights with same origin and destination)--
WITH RoundTripBookings AS (
    SELECT
        Booking.PassengerID,
        Booking.ID
    FROM
        Booking
        INNER JOIN Flight ON Booking.ID = Flight.ID
    WHERE
        Flight.Origin = Flight.Destination
)

SELECT DISTINCT
    Passenger.ID AS PassengerID,
    Passenger.Name
FROM
    Passenger
    INNER JOIN RoundTripBookings ON Passenger.ID = RoundTripBookings.PassengerID;

--Find the average price of flights for each airline--
```

Results Messages

	PassengerID	Name

```sql
--Find the average price of flights for each airline--
WITH AvgAirlinePrices AS (
    SELECT
        Airlines.ID AS AirlineID,
        Airlines.Name AS AirlineName,
        AVG(Flight.Price) AS AverageFlightPrice
    FROM
        Airlines
        LEFT JOIN Flight ON Airlines.ID = Flight.ID
    GROUP BY
        Airlines.ID, Airlines.Name
)

SELECT * FROM AvgAirlinePrices;

--Identify passengers who have booked the same flight multiple times--
WITH DuplicateBookings AS (
```

Results | Messages

	AirlineID	AirlineName	AverageFlightPrice
1	1	Delta Airlines	800
2	2	British Airways	450
3	3	American Airlines	600
4	4	Air France	700
5	5	Lufthansa	500
6	6	Emirates	750
7	7	Singapore Airlines	550
8	8	Qantas Airways	400
9	9	Cathay Pacific	900
10	10	Turkish Airlines	650
11	11	Qatar Airways	700
12	12	ANA All Nippon	800
13	13	KLM Royal Dutch	600
14	14	Virgin Atlantic	750
15	15	Etihad Airways	550

```sql
--Identify passengers who have booked the same flight multiple times--
WITH DuplicateBookings AS (
    SELECT
        PassengerID,
        Booking.ID AS BookingID,
        COUNT(Booking.ID) AS BookingCount
    FROM
        Booking
    GROUP BY
        PassengerID, Booking.ID
    HAVING
        COUNT(Booking.ID) > 1
)

SELECT * FROM DuplicateBookings;

--List flights with the highest total booking costs--
SELECT
```

Results Messages

	PassengerID	BookingID	BookingCount

```sql
--List flights with the highest total booking costs--
SELECT
    Flight.ID AS FlightID,
    Flight.Origin,
    Flight.Destination,
    SUM(Booking.TotalCost) AS TotalBookingCost
FROM
    Flight
    INNER JOIN Booking ON Flight.ID = Booking.ID
GROUP BY
    Flight.ID, Flight.Origin, Flight.Destination
ORDER BY
    TotalBookingCost DESC
```

Results | Messages

	FlightID	Origin	Destination	TotalBookingCost
1	12	SFO	JFK	900
2	31	LHR	CDG	900
3	25	ORD	SFO	800
4	19	LHR	CDG	800
5	9	JFK	LAX	800
6	40	LAX	JFK	800
7	45	SFO	JFK	750
8	3	ORD	SFO	750
9	13	LAX	JFK	750
10	30	ORD	SFO	750
11	34	SFO	JFK	750
12	24	LAX	JFK	700
13	8	LHR	CDG	700
14	16	SFO	JFK	700
15	43	CDG	LHR	700
16	39	SFO	JFK	700
17	10	CDG	LHR	650
18	33	JFK	LAX	650
19	28	SFO	JFK	600
20	22	SFO	JFK	600
21	4	SFO	JFK	600

```sql
--Calculate the percentage of bookings for each airline--
WITH BookingPercentages AS (
    SELECT
        Airlines.ID AS AirlineID,
        Airlines.Name AS AirlineName,
        COUNT(Booking.ID) * 100.0 / (SELECT COUNT(*) FROM Booking) AS BookingPercentage
    FROM
        Airlines
        LEFT JOIN Flight ON Airlines.ID = Flight.ID
        LEFT JOIN Booking ON Flight.ID = Booking.ID
    GROUP BY
        Airlines.ID, Airlines.Name
)
SELECT * FROM BookingPercentages;
```

Results | Messages

	AirlineID	AirlineName	BookingPercentage
1	1	Delta Airlines	2.222222222222
2	2	British Airways	2.222222222222
3	3	American Airlines	2.222222222222
4	4	Air France	2.222222222222
5	5	Lufthansa	2.222222222222
6	6	Emirates	2.222222222222
7	7	Singapore Airlines	2.222222222222
8	8	Qantas Airways	2.222222222222
9	9	Cathay Pacific	2.222222222222
10	10	Turkish Airlines	2.222222222222
11	11	Qatar Airways	2.222222222222
12	12	ANA All Nippon	2.222222222222
13	13	KLM Royal Dutch	2.222222222222
14	14	Virgin Atlantic	2.222222222222
15	15	Etihad Airways	2.222222222222

```sql
--Lists passengers without any bookings--
WITH PassengersWithoutBookings AS (
    SELECT
        Passenger.ID AS PassengerID,
        Passenger.Name
    FROM
        Passenger
        LEFT JOIN Booking ON Passenger.ID = Booking.PassengerID
    WHERE
        Booking.ID IS NULL
)

SELECT *
FROM PassengersWithoutBookings;

--Retrieves the average capacity of aircraft for each airline--
```

Results | Messages

	PassengerID	Name
1	6	Emily Davis
2	7	Frank Miller
3	8	Grace Turner
4	9	Henry Harris
5	10	Isabella Green
6	11	Jack Anderson
7	12	Katie Wilson
8	13	Liam Thomas
9	14	Mia Parker
10	15	Noah Wright

```sql
--Retrieves the average capacity of aircraft for each airline--
WITH AvgAircraftCapacity AS (
    SELECT
        Airlines.ID AS AirlineID,
        AVG(Aircraft.Capacity) AS AverageAircraftCapacity
    FROM
        Airlines
        LEFT JOIN Aircraft ON Airlines.ID = Aircraft.AirlineID
    GROUP BY
        Airlines.ID
)

SELECT *
FROM AvgAircraftCapacity;
```

Results | Messages

	AirlineID	AverageAircraftCapacity
1	1	150
2	2	180
3	3	250
4	4	550
5	5	300
6	6	250
7	7	400
8	8	200
9	9	300
10	10	250
11	11	100
12	12	90
13	13	220
14	14	200
15	15	70