CS 586/486 Introduction to Databases
Winter 2016 Quarter
Project: Airline Tracking System

Vinchhi Komal
Delaplane Connor
Litvinenko Max

Deliverable 3

Final Write-up: You are to implement your schema in a relational database and populate it with data. The preferred DBMS is Postgres. If you want to use a different DBMS, you must get advanced approval. You then need to translate your 20 questions into SQL and execute them on your database. Your write-up of these activities should include the following:

- Your ER diagram, showing any changes you made during the implementation process
- The CREATE TABLE statements for your database
- A brief description of how you populated the database
- A listing of the contents of all of your tables
- For each of your 20 questions, the question in English, its translation to SQL and the (full) answer to the query. (If you needed to change any of your original questions, also list the originals and why you needed to change or replace them.)

Please submit your write-up as a single PDF or as a link to a web page, which can in turn link to the different parts of the assignment.

GRADING

Submission 3 is worth 70 points, and the score will include the following factors, in addition to general correctness:

-
- Does the data used do a good job of demonstrating the adequacy of your schema and the correctness of your queries?
- Does your set of queries make use of all the parts of your schema and a range of SQL features?

The points breakdown for this submission is
I. Grammar and completeness: 15
II. Data entry – description and automation: 15
III. Data suitability: 10
IV: Queries – correct translation of English, coverage of schema and SQL features: 30

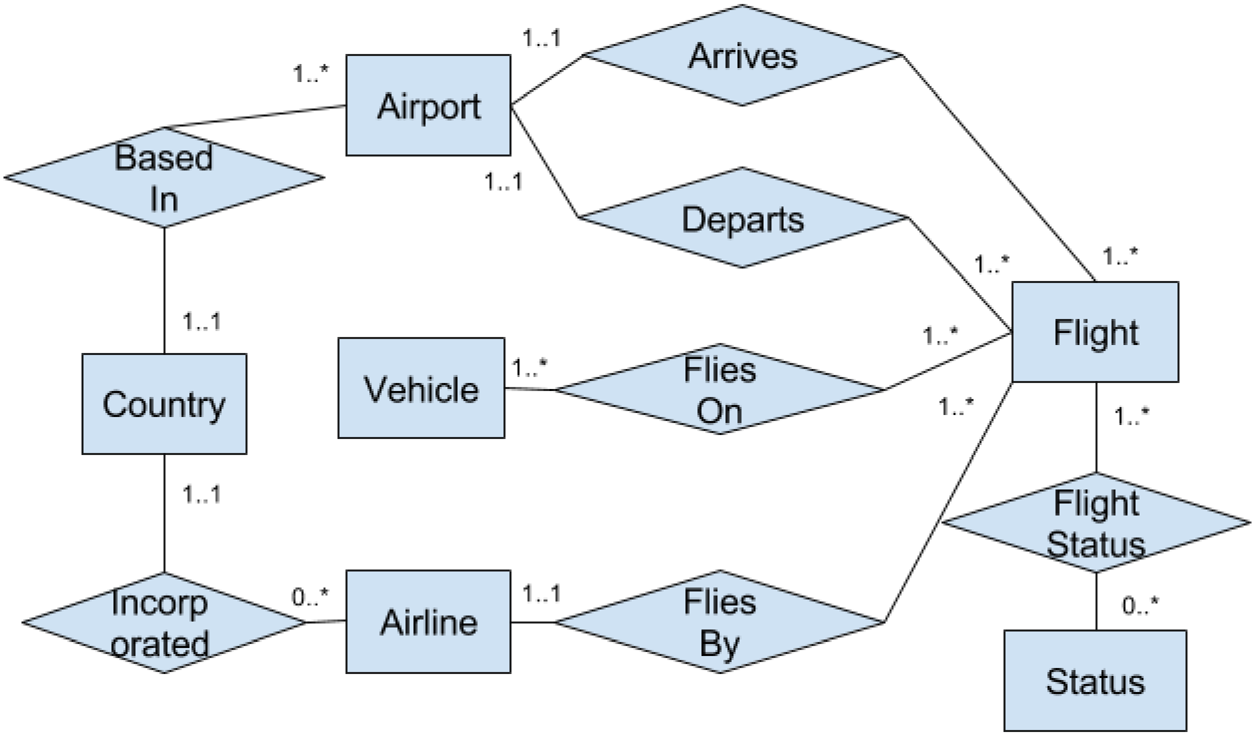
Table of Contents

1. Domain Description

Our domain for this project is an airline tracking system. The database will contain tables describing data for **airport**s, **airline**s, **flight**s, air **vehicle**s, and arrival/departure **flight status** (with foreign key to flight, status, and date columns). There will be an additional relation connecting the Airport relation to Airline, because an airline can service many airports and an airport can have many airlines that visit it. Flight information will include the flight #, airline company, current destination, and gate #. Because frequent updates to this database are expected as planes arrive/depart, run late, change gates, etc., the database will be implemented with the goal of maximum simplicity and efficiency for tuple modifications either manually or automatically. For this reason, among others, we are going to make the **FlightStatus** relation the primary “dynamic” relation, where data is expected to be modified frequently; the others should only need to be updated occasionally. NOTE: this is not a flight-booking system, but rather a general-purpose flight-tracking system.

Past deliverables can be found here:
https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/Deliverable1.pdf
https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/Deliverable2.pdf
https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/Deliverable3.pdf

2. ER Diagram



ER Diagram was well designed. Since our deliverable 2, we did not need to change the structure of it. All we did was add cardinalities.

1. Country (CountryId, CountryName)
2. Airport (AirportId, AirportName, City, CountryId, Alias, Latitude, Longitude, Altitude, TimeZone)
3. Airline (AirlineId, AirlineName, Alias, CountryId)
4. Flight (FlightId, FlightNo, AirlineId, FromAirportId, ToAirportId, NumOfStops, *WeekDays*, *DepartureTime*, *ArrivalTime*, *Distance*, *Duration*)
5. AirVehicle (PlaneId, TypeNum)
6. FliesOn (FlightAirVehicleId, FlightId, PlaneId)
7. Status(StatusId, *StatusName*)
8. FlightStatus (FlightStatusId, FlightId, StatusId, *OptionalNote*, *DepartureDateTime*, *ArrivingDateTime*, *InsertTime*)

*_Italic_ - means this data was NOT provided by our original source. We generated it using python.

3. CREATE TABLE Statements

https://github.com/maxpdx/cs468-FlightTracking/blob/master/sql/_CREATE_TABLE.sql

```
CREATE SCHEMA IF NOT EXISTS "AirlineTrackingSystem";

CREATE TABLE IF NOT EXISTS Country
(
  CountryId INT NOT NULL,
  CountryName VARCHAR(100),
  PRIMARY KEY(CountryId)
);

CREATE TABLE IF NOT EXISTS Airport
(
  AirportId INT NOT NULL,
  AirportName VARCHAR(255),
  City VARCHAR(100),
  CountryId INT,
  Alias VARCHAR(10),
  Latitude NUMERIC,
  Longitude NUMERIC,
  Altitude NUMERIC,
  TimeZone SMALLINT NOT NULL,
  PRIMARY KEY(AirportId),
  FOREIGN KEY (CountryId) REFERENCES Country(CountryId)
);

CREATE TABLE IF NOT EXISTS Airline
(
  AirlineId INT NOT NULL,
  AirlineName VARCHAR(255),
  Alias VARCHAR(50),
  CountryId INT,
  PRIMARY KEY(AirlineId),
  FOREIGN KEY (CountryId) REFERENCES Country(CountryId)
);

CREATE TABLE IF NOT EXISTS Flight
(
  FlightId INT NOT NULL,
  FlightNo INT NOT NULL,
  AirlineId INT NOT NULL,
  FromAirportId INT NOT NULL,
  ToAirportId INT NOT NULL,
  NumOfStops SMALLINT,
  WeekDays VARCHAR(15),
  DepartureTime TIME,
  ArrivalTime TIME,
  Distance SMALLINT,
  Duration TIME,
  PRIMARY KEY(FlightId),
  FOREIGN KEY(AirlineId) REFERENCES Airline(AirlineId),
  FOREIGN KEY(FromAirportId) REFERENCES Airport(AirportId),
  FOREIGN KEY(ToAirportId) REFERENCES Airport(AirportId)
);

CREATE TABLE IF NOT EXISTS AirVehicle
(
  PlaneId INT NOT NULL UNIQUE,
  TypeNum VARCHAR (50)
);

CREATE TABLE IF NOT EXISTS FliesOn
(
  FlightAirVehicleId INT NOT NULL,
  FlightId INT NOT NULL,
  PlaneId INT NOT NULL,
  FOREIGN KEY(FlightId) REFERENCES Flight(FlightId),
  FOREIGN KEY(PlaneId) REFERENCES Airvehicle(PlaneId)
);

CREATE TABLE IF NOT EXISTS Status
(
  StatusId INT NOT NULL,
  StatusName VARCHAR(100),
  PRIMARY KEY(StatusId)
);

CREATE TABLE IF NOT EXISTS FlightStatus
(
  FlightStatusId INT NOT NULL UNIQUE,
  FlightId INT NOT NULL,
  StatusId INT NOT NULL,
  OptionalNote VARCHAR(255),
  DepartureDateTime TIMESTAMP,
  ArrivingDateTime TIMESTAMP,
  InsertTime TIME,
  FOREIGN KEY(FlightId) REFERENCES Flight(FlightId),
  FOREIGN KEY(StatusId) REFERENCES Status(StatusId)
);
```

4. Data Population

As we planned, using Python, we generated INSERT INTO SQL queries by processing the CSV data files (from source: http://openflights.org/data.html) and generating some extra random data (repository of code: https://github.com/maxpdx/cs468-FlightTracking). However, after generating SQL files, we realized that phpPgAdmin doesn't support "Importing SQL files" (it supports CSV, XML, or Tabbed formats only). Copying and pasting SQL queries manually didn't work out well; with such a large number of queries, it was memory intensive and a PC would freeze for some time wanting to crash the chrome tabs. We wrote another python script to connect to the database and execute the queries over the script. After hours of running (~ 4 hrs of executing > 200K queries), we were able to (1) create new schema, (2) create tables, and (3) populate data. Thus we totally avoided manual table creation and manual data entries.

Number of SQL INSERT INTO statements per table AND a link to the full dataset of each table:

1. sql/country.sql [318]
   https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_tables/1.Country.csv

2. sql/airport.sql [8107]
   https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_tables/2.Airport.csv

3. sql/airline.sql [6048]
   https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_tables/3.Airline.csv

4. sql/flight.sql [66548]
   https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_tables/4.Flight.csv

5. sql/airVehicle.sql [167]
   https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_tables/5.AirVehicle.csv

6. sql/fliesOn.sql [92021]
   https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_tables/6.FliesOn.csv

7. sql/status.sql [7]
   https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_tables/7.Status.csv

8. sql/flightStatus.sql [34967]
   https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_tables/8.FlightStatus.csv

5. Questions & SQL Queries

1. What is the info for all flights departing today?
   (Original: What flights are departing today?)
   Reason for revising: We decided to make these questions more specific by asking for the *information* for each flight.

   ```
   SELECT *
   FROM Flight NATURAL JOIN FlightStatus
   WHERE DATE_TRUNC('day', DepartureDateTime) = CURRENT_DATE;
   ```

   https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_query_results/Q01.csv

2. What is the info for all flights arriving in the next 3 days?
   (Original: What flights are arriving in the next 3 days?)
   Reason for revising: We decided to make these questions more specific by asking for the *information* for each flight.

   ```
   SELECT *
   FROM Flight NATURAL JOIN FlightStatus
   WHERE DATE_TRUNC('day', ArrivingDateTime) >= CURRENT_DATE
   AND DATE_TRUNC('day', ArrivingDateTime) <= CURRENT_DATE + INTEGER '3';
   ```

   https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_query_results/Q02.csv

3. What is the distance to AMS from PDX?

   ```
   SELECT Distance
   FROM Flight, Airport A1, Airport A2
   WHERE A1.alias = 'AMS' AND A2.alias = 'PDX'
           AND Flight.ToAirportId = A1.AirportId
           AND Flight.FromAirportId = A2.AirportId;
   ```

   https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_query_results/Q03.csv

4. What are the durations of all flights to Seattle from San Francisco?
   (Original: What is the duration of the flight to St. Paul from San Francisco?)
   Reasons for revising:
   - We discovered that there was no data for flights traveling directly from San Francisco to St. Paul.
   - We realized that there could be multiple flights with different departure/arrival airports (and with different durations) all going from one city to another.

```
SELECT FlightId, Duration
FROM Flight, Airport A1, Airport A2
WHERE A1.City = 'Seattle' AND A2.City = 'San Francisco'
        AND Flight.ToAirportId = A1.AirportId
        AND Flight.FromAirportId = A2.AirportId;
```

   https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_query_results/Q04.csv

5. What flights were delayed today?

```
SELECT *
FROM Flight NATURAL JOIN (FlightStatus NATURAL JOIN Status)
WHERE StatusName = 'Delayed'
        AND (DATE_TRUNC('day', DepartureDateTime) = CURRENT_DATE
        OR DATE_TRUNC('day', ArrivingDateTime) = CURRENT_DATE);
```

   https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_query_results/Q05.csv

6. What flights are departing from Bangkok in the next 2 hours?
   (Original: What flights are departing from New York in the next 2 hours?)

   Reason for revising: Originals returned no tuples at time of execution.

```
SELECT F.*
FROM Flight F NATURAL JOIN FlightStatus, Airport A
WHERE A.City='Bangkok'
        AND A.airportId = F.FromAirportId
        AND DATE_TRUNC('day', DepartureDateTime) >= CURRENT_DATE
        AND DATE_TRUNC('day', DepartureDateTime) <= CURRENT_DATE +
                                                        INTERVAL '2 hours';
```

   https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_query_results/Q06.csv

7. What are all the different existing airline companies?

```
SELECT DISTINCT AirlineName
FROM Airline;
```

https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_query_results/Q07.csv

8. What flights are cancelled this week?

```
SELECT *
FROM Flight NATURAL JOIN (FlightStatus NATURAL JOIN Status)
WHERE StatusName = 'Cancelled'
  AND DATE_TRUNC('day', DepartureDateTime) >= CURRENT_DATE
  AND DATE_TRUNC('day', DepartureDateTime) <= CURRENT_DATE + INTEGER '7';
```

https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_query_results/Q08.csv

9. What flights are departing before noon tomorrow?

```
SELECT *
FROM Flight NATURAL JOIN FlightStatus
WHERE DepartureDateTime <
                        CURRENT_DATE + INTEGER '1' + INTERVAL '12 hours'
            AND DATE_TRUNC('day', DepartureDateTime) >= CURRENT_DATE;
```

https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_query_results/Q09.csv

10. What is the flight with the longest duration?

```
SELECT *
FROM Flight
WHERE Duration = (SELECT MAX(Duration) FROM Flight);
```

https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_query_results/Q10.csv

11. What airports have a flight scheduled for departure at 2:05 pm today?
    (Original: What airports have a flight scheduled for departure at 10:55 pm today?)
    Reason for revising: Originals returned no tuples at time of execution.

    ```
    SELECT A.AirportName, A.Alias
    FROM Airport A, (Flight NATURAL JOIN FlightStatus) F
    WHERE F.DepartureDateTime = CURRENT_DATE + TIME '14:05'
            AND A.AirportId = F.FromAirportId;
    ```

    https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_query_results/Q11.csv

12. What is the average travel distance for all flights departing from Columbus, OH?

    ```
    SELECT AVG(Distance)
    FROM Flight, Airport
    WHERE Airport.city = 'Columbus' AND Flight.FromAirportId = Airport.AirportId;
    ```

    https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_query_results/Q12.csv

13. What United Airlines flights are departing this week?
    (Original: What Alaska Airlines flights are departing this week?)
    Reason for revising: Originals returned no tuples at time of execution.

    ```
    SELECT *
    FROM Flight NATURAL JOIN FlightStatus NATURAL JOIN Airline
    WHERE AirlineName = ‘United Airlines’
            AND DepartureDateTime <= CURRENT_DATE + INTEGER ‘7’
            AND DepartureDateTime >= CURRENT_DATE;
    ```

    https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_query_results/Q13.csv

14. What is the vehicle id of the flight with ID 237?
    (Original: What is the vehicle id of a particular flight?)
    Reason for revising: Could not query original because it was not specific enough.

    ```
    SELECT PlaneId
    FROM Flight NATURAL JOIN (FliesOn NATURAL JOIN AirVehicle)
    WHERE FlightId = 237;
    ```

    https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_query_results/Q14.csv

15. What flights are arriving in or departing from Munich in the next hour?
(Original: What flights are arriving in Chicago in the next hour?)
Reason for revising:
    - Wanted to give an example of a more complicated query.
    - There were no flights meeting the original specification at the time of execution.

```
SELECT F.*
FROM (Flight NATURAL JOIN FlightStatus) AS F, Airport A
WHERE A.city = 'Munich'
AND(
        (F.FromAirportId = A.AirportId
        AND DATE_TRUNC('day', DepartureDateTime) = CURRENT_DATE
        AND DepartureDateTime <= CURRENT_TIMESTAMP + INTERVAL '1 hour')
        OR
        (F.ToAirportId = A.AirportId
        AND DATE_TRUNC('day', ArrivingDateTime) = CURRENT_DATE
        AND ArrivingDateTime <= CURRENT_TIMESTAMP + INTERVAL '1 hour')
);
```

https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_query_results/Q15.csv

16. What is the current status of Flight No. 2895?
(Original: What are the nearest and furthest departure/arrival times for all flights?)
Reason for revising: The original was deemed impossible to query because flights are repetitious.

```
SELECT StatusName, OptionalNote
FROM Flight NATURAL JOIN (FlightStatus NATURAL JOIN Status)
WHERE FlightStatusId = (
        SELECT MAX(FlightStatusId)
        FROM FlightStatus NATURAL JOIN Flight
        WHERE FlightNo = 457
        );
```

https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_query_results/Q16.csv

17. On what weekdays does Flight 1787 depart?
(Original: What flights are scheduled for bag pickup at carousel 3?)
Reason for revising Questions 17, 18 and 20: This question involved a theoretical attribute that we ended up not modeling in our end schema because we either found it to be unimportant information or we had no reasonably simple way to generate/fetch the data required.

```
SELECT WeekDays
FROM Flight
WHERE FlightNo = 1787;
```

https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_query_results/Q17.csv

18. How many airports are located in Germany?
(Original: What helicopters are departing today?)
Reason for revising Questions 17, 18 and 20: This question involved a theoretical attribute that we ended up not modeling in our end schema because we either found it to be unimportant information or we had no reasonably simple way to generate/fetch the data required.

```
SELECT COUNT(AirportId)
FROM Airport NATURAL JOIN Country
WHERE CountryName='Germany’;
```

https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_query_results/Q18.csv

19. How many flights visit PDX?

```
SELECT COUNT(FlightId)
FROM Flight, Airport
WHERE Airport.Alias = ‘PDX’ AND Flight.ToAirportId = Airport.AirportId;
```

https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_query_results/Q19.csv

20. How many flights have more than 1 stop?
(Original: What is the capacity of a flight 788 from Frontier Airline that arrives in next 3 days?)
Reason for revising Questions 17, 18 and 20: This question involved a theoretical attribute that we ended up not modeling in our end schema because we either found it to be unimportant information or we had no reasonably simple way to generate/fetch the data required.

```
SELECT COUNT(FlightId)
FROM Flight
WHERE NumOfStops >= 1;
```

https://github.com/maxpdx/cs468-FlightTracking/blob/master/Docs/CSV_query_results/Q20.csv

6. Project Analysis

We believe that the above tests show that our tracking system does what it was meant to do in a relatively efficient manner, such that query implementations can be kept relatively simple and response times are minimized with regards to the large data set we used.

Revised questions:

After implementing our database, we found that there were many alterations we wanted to make to our original questions from project 1. There were interpretations to many questions that we asked before creating the structure of the database. And that's what happens in real life, when a boss(client), who doesn't know the specifics of the database structure, comes up and asks to query some questions in english. Oftentimes, developers have to clear these questions to avoid ambiguities, confusions and wrong results printed. Most of these changes were made either because we decided that the originals were:

- not specific enough
- they required dynamic queries (e.g. anything involving the CURRENT_DATE or CURRENT_TIMESTAMP functions) and we wanted to provide evidence of each query returning a valid, non-empty set of results in our deliverable
- we ended up not having some data or changed our data structure from what we expected for better real world-like application.
- make sure we are using all of our major tables.

What can be improved:

We are proud of our such a huge data set, it was a great experience for us to work on such large set of data. However, we had decided to generate a lot of data which cause us to have some unrealistic results. (Ex1: Q4, where duration flights from San Francisco to Seattle range from 8.75 to almost 24 hours. Ex2: Q5, 'departure' and 'arriving' datetimes are completely irrelevant from previous statuses and insert time.)