

600.415—Database Systems — Final Project Phase I

Matt Ziegelbaum & Parker Shelton
mziegelbaum@acm.jhu.edu, parker.shelton@jhu.edu

December 18, 2009

1 Project Members:

- Matt Ziegelbaum
- Parker Shelton

2 Target Domain

We plan to develop a system that allows users to make queries for information about international air travel. We plan on supporting queries regarding airlines, airports, and routes, as well as equipment (airplanes) used by the various airlines.

3 Questions

1. List all the airports in the database.
2. List all airlines in the database.
3. List all routes in the database.
4. List all airlines that fly between two cities.
5. List all cities an airline services.
6. List all airlines that service the exact same cities.
7. List all routes that an airline flies.
8. List the fewest number of stops between any two cities (including direct flights).
9. List all airlines that fly out of a city.

10. List all possible destinations from a given airport.
11. Compute the straight line distance between two airports.
12. Compute the time difference between two airports.
13. Compute the local time for arrival and departures of a flight given a departure time in UTC.
14. Compute the number flights an airport can service per hour/day/year.
15. Compute the number of passengers per route.
16. Estimate the cost to fly between two airports.
17. Estimate the fuel cost to fly between two airports.
18. Estimate the cost of the cheapest flight (or series of flights) between two airports.
19. List the airport with the highest/lowest elevation.

4 Design

```
CREATE TABLE Airports (  
    id INT NOT NULL primary key,  
    Name VARCHAR(255),  
    City VARCHAR(255),  
    Country VARCHAR(255),  
    IATA VARCHAR(3) NOT NULL unique key,  
    ICAO VARCHAR(4) NOT NULL unique key,  
    Latitude DECIMAL(9),  
    Longitude DECIMAL(9),  
    Altitude INT,  
    Timezone INT,  
    DST CHAR,  
    NumRunways INT  
);
```

```
CREATE TABLE Airlines (  
    id INT NOT NULL primary key,  
    Name VARCHAR(255),  
    Alias VARCHAR(255),  
    IATA VARCHAR(2),  
    ICAO VARCHAR(3) NOT NULL,  
    Callsign VARCHAR(255),
```

```

        Country VARCHAR(255),
        Active CHAR
    );

CREATE TABLE Routes (
    Airline VARCHAR(3) NOT NULL,
    AirlineID INT,
    Source VARCHAR(4),
    SourceID INT,
    Dest VARCHAR(4),
    DestID INT,
    Codeshare CHAR,
    Stops INT,
    Equipment VARCHAR(255),
    TicketPrice INT
);

CREATE TABLE Equipment (
    id VARCHAR(3) NOT NULL primary key,
    Name VARCHAR(255),
    Passengers INT,
    Fuel INT,
);

```

5 Sample SQL Statements

```

#7      CREATE PROCEDURE AirlineRoutes(IN airline1 VARCHAR(255))
        BEGIN
            SELECT R.* FROM Airlines A, Routes R
            WHERE (A.Name = airline1 OR A.IATA = airline1 OR A.ICAO = airline1)
                AND
                R.AirlineID = A.id;
        END

```

(The above takes either an IATA/FCC code, ICAO code, or full name.)

```

#9a     CREATE PROCEDURE AirlinesFromCity(IN city VARCHAR(255))
        BEGIN
            SELECT L.Name AS Airlines
            FROM Routes R INNER JOIN Airports A ON R.SourceID = A.ID
            INNER JOIN Airlines L ON L.ID = R.AirlineID WHERE A.City = city;

```

END

```
#9b CREATE PROCEDURE AirlinesToCity(IN city VARCHAR(255))
BEGIN
    SELECT L.Name AS Airlines
    FROM Routes R INNER JOIN Airports A ON R.DestID = A.ID
    INNER JOIN Airlines L ON L.ID = R.AirlineID WHERE A.City = city;
END
```

```
#11 CREATE PROCEDURE AirportDistance(IN airport1 INT, IN airport2 INT)
BEGIN
    SELECT SQRT(POW((A1.latitude - A2.latitude), 2)
        + POW((A1.longitude - A2.longitude), 2))
    AS Distance FROM Airport A1, Airport A2
    WHERE A1.id = airport1 AND A2.id = airport2;
END
```

(The above does not take into account the curvature of the earth.)

```
#12 CREATE PROCEDURE AirportTimeDifference(IN airport1 INT,
    IN airport2 INT, OUT diff INT)
    SELECT A1.Timezone - A2.Timezone INTO diff
    FROM Airport A1, Airport A2 WHERE
        A1.id = airport1 AND A2.id = airport2;
END
```

```
#19a CREATE PROCEDURE HighestElevationAirport(OUT aout INT)
    SELECT MAX(Altitude) INTO aout FROM Airports;
END
```

```
#19b CREATE PROCEDURE LowestElevationAirport(OUT aout INT)
    SELECT MIN(Altitude) INTO aout FROM Airports;
END
```

6 Database Population

We will load the database with values by importing from CSV files. The CSV files come from OpenFlights.org, AirlineRouteMapper (arm.64hosts.com), and OurAirports.com. Ad-

ditional data on equipment will be extracted from various web sources.¹

Ticket prices will be generated by an algorithm² at database population time.

7 User Interface

We plan on focussing on an advanced web GUI using the Google Maps API, JavaScript, and CSS. The Maps API will be used to display routes, pinpoint airports, and overlay data about routes, airports, and airlines.

8 Advanced Topics

1. Advanced GUI (See previous section)
2. Data mining – See database population.
3. Complex data extraction – We have a “dirty” data set, in that our data is not perfectly formatted and is missing information. We must extrapolate and add additional information from a number of different sources.

9 Database Platform

We will be developing with MySQL 5.1.x on OS X with Core 2 Duo 2.66Ghz.³ Our backend will be written in Ruby using the Sinatra web framework and the Ruby MySQL connector. The front end will be developed using HTML, CSS, and JavaScript. We will deploy on a Core 2 Duo 3.0Ghz with 8GB of RAM.⁴

¹Teh interwebs

²To be determined how the heuristic works.

³We both have almost matching Macbook Pros

⁴The JHU ACM Web server