CSE 560 Project Milestone 1 (Mar. 6th, 2022)

1. Project Detail

- Project Name: Flight Delay Information
- Team Name: dbql kicks in
- Members:
    - Sungjoon Park (spark55)
    - Dennis Mulumbi Kyalo (dkyalo)
    - Sol Jang (soljang)

2. Problem Statement

- Description of the problem
    - Motivation

      The flight schedule is basically affected by various factors such as weather, aircraft safety check, administration, excess demand, etc. As such, flights are more likely to be delayed than on-time transportation such as trains. Unexpected changes in the flight schedule can cause inconvenience to customers using flights. In this regard, our main purpose is to build a database for our target users so that they can take flight delay information into account before they make decisions.
    - Problems
        - How to query the flight list in which conditions such as time, carrier, journey, weather, etc. match a customer's flight plan
        - How to enable customers to query seasonality of delay data in terms of 'day of week', 'month', 'time of day', etc.
        - How to query for comparison in which customers can compare various travel plans depending on the conditions in the first problem and choose the optimal one
- The comparison between DB systems and excel files
    - Data Purpose
        - While Excel files are great for numeric and text values in relatively low volume, DB systems expertly handle not only those but also easily incorporate other types of information, such as images and documents. In addition, DB systems can accommodate high volume and large file size data downloads like those from data loggers, GPS devices, cameras, drones, and other collection devices.
    - Data Volume
        - For long-term projects with numerous monitoring locations, millions of data points can be generated. Because DB systems more efficiently can store and handle volumes of information that would be unmanageable in

Excel files. In particular, spreadsheets have record limitations unlike DB systems and require a larger amount of hard-drive space for storage.

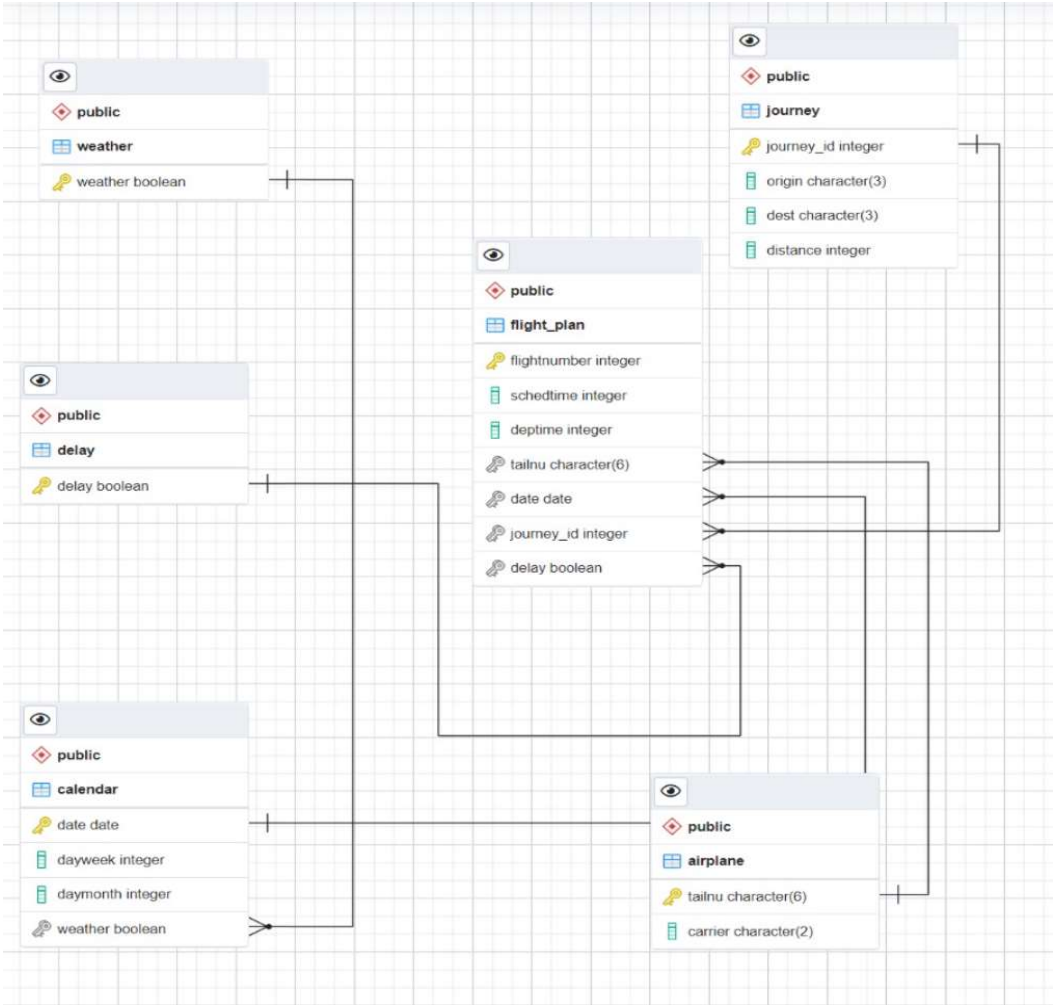
- Editing
  - Updates to databases are typically easier than spreadsheets, especially if the same information is maintained in multiple records or spreadsheets. In addition, a database can update records in bulk.
- Data Accessibility
  - Although data in spreadsheets can be sorted and filtered, a database has broad querying functionality that can retrieve all records matching select criteria, cross-reference records in multiple tables, and perform complex aggregate calculations across multiple tables.
  - When data are maintained in a centralized relational database, data is easily accessible for querying, analysis, and reporting. Since the database will enforce the same quality standards for any dataset, decisions can be made with confidence.
- Speed
  - Databases are designed to refer to information without loading all the information into memory, unlike spreadsheets.
- Data Integrity
  - Data integrity is a key difference between databases and spreadsheets. Relational databases follow standardized integrity rules to ensure that the data they contain are accurate and accessible.
- Redundancy
  - The database structure also avoids data redundancy. Frequent data in spreadsheets are copied multiple times and the same data are maintained in separate spreadsheet files, creating a nightmare to ensure accurate data when a change is required.
- Error Proliferation
  - Preventing and efficiently identifying data errors in spreadsheets is challenging. It is also much easier to accidentally overwrite or delete data in a spreadsheet than in a database.
- User Access and Security
  - Unlike spreadsheets, modern relational databases are designed for multiple users. Moreover, databases provide centralized data storage and offer better security. User permissions can be assigned to view data, edit data, and restrict access to privileged information.

3. <u>Target User</u>

- The user
  - People who book a flight (commuters, travelers)
  - Travel agencies
  - Logistics companies using air freight but without their own airplanes
- The administer

- Airport administrative organization
- Real-life scenario description
  - When travelers or travel agencies book flight tickets, they are worried that the aircraft would depart on time. They can reduce the probability that their flight is delayed by avoiding certain conditions such as certain time, carrier.
  - Moreover, customers like logistics companies will be able to prevent problems caused by aircraft delays in advance. For example, plans may be established in advance by considering the delay time in cases flight departure delays are frequent.
  - Regardless of whether a customer is an individual or a company, all customers will also want to know how long their departure time will be delayed on average. Through this database, on average, they can figure out how much delay will be and what the variance will be like.

4. E/R Diagram



5. Database Implementation

- Data Schemas
  - FLIGHT_PLAN TABLE
    - All information can be concentrated on this table. It brings information form 'AIRPLANE', 'JOURNEY', 'CALENDAR', and 'DELAY' tables using foreign keys. These are 'talinu', 'journey_id', 'date', and 'delay', which are primary keys in each table and all of them are one-to-many type of relationship.

Flight Delay/postgres@spark55

Query Editor | Query History | Explain | Notifications

```
1  CREATE TABLE FLIGHT_PLAN(
2      Flightnumber  INT NOT NULL,
3      Schedtime     INT NOT NULL,
4      Deptime       INT NOT NULL,
5      Tailnu        CHAR(6) NOT NULL,
6      Date          DATE NOT NULL,
7      Journey_ID    INT NOT NULL,
8      Delay         BOOLEAN NOT NULL,
9      PRIMARY KEY (Flightnumber),
10     FOREIGN KEY (Tailnu)
11         REFERENCES AIRPLANE(Tailnu)
12         ON DELETE CASCADE,
13     FOREIGN KEY (Date)
14         REFERENCES CALENDAR(Date)
15         ON DELETE CASCADE,
16     FOREIGN KEY (Journey_ID)
17         REFERENCES JOURNEY(Journey_ID)
18         ON DELETE CASCADE,
19     FOREIGN KEY (Delay)
20         REFERENCES DELAY(Delay)
21         ON DELETE CASCADE);
```

  - AIRPLANE TABLE
    - 'FLIGHT_PLAN' refers to this table using foreign key 'tailnu' by which users can have access to the information regarding each flight including its carrier and the tail number (airplane identifier).

Flight Delay/postgres@spark55

Query Editor | Query History | Explain | Notifications

```
1  CREATE TABLE AIRPLANE(
2      Tailnu  CHAR(6) NOT NULL,
3      Carrier CHAR(2) NOT NULL,
4      PRIMARY KEY (Tailnu));
```

  - JOURNEY TABLE
    - This table contains all currently operating air travel routes in the data set. Each flight information in 'FLIGHT_PLAN' table such as its origin and destination can be found using foreign key 'journey_id'.

Flight Delay/postgres@spark55

Query Editor | Query History | Explain | Notifications

```
CREATE TABLE JOURNEY(
    Journey_ID  INT NOT NULL,
    Origin      CHAR(3) NOT NULL,
    Dest        CHAR(3) NOT NULL,
    Distance    INT NOT NULL,
    PRIMARY KEY (Journey_ID));
```

- CALENDAR TABLE
  - This table has 1 foreign key ('weather'), and the type of relationship between tables is one-to-many. That is, each value of 'weather' attribute brought from the 'WEATHER' table can be found several times in the 'weather' attribute in the 'CALENDAR' table. This table delivers not only data information but also weather information to 'FLIGHT_PLAN' table through the channel set by foreign key 'weather'.

Flight Delay/postgres@spark55

Query Editor | Query History | Explain | Notifications

```
CREATE TABLE CALENDAR(
    Date       DATE NOT NULL,
    Dayweek    INT NOT NULL,
    Daymonth   INT NOT NULL,
    Weather    BOOLEAN NOT NULL,
    PRIMARY KEY (Date),
    FOREIGN KEY (Weather)
        REFERENCES WEATHER(Weather)
        ON DELETE CASCADE);
```

- WEATHER TABLE
  - It has weather categories. Users can see if the weather was bad or not on a specific day by foreign key 'weather'. Besides, they can even tell the weather when a particular flight was in operation through the connection of the three tables, 'WEATHER', 'CALENDAR', and 'FLIGHT_PLAN'.

Flight Delay/postgres@spark55

Query Editor | Query History | Explain | Notifications

```
CREATE TABLE WEATHER(
    Weather BOOLEAN NOT NULL,
    PRIMARY KEY (Weather));
```

- DELAY TABLE
  - It has 'delay' attribute. 'FLIGHT_PLAN' refers to it using foreign key 'delay' which indicates whether a flight is delayed or not.

Flight Delay/postgres@spark55

Query Editor | Query History | Explain | Notifications

```
CREATE TABLE DELAY(
    Delay   BOOLEAN NOT NULL,
    PRIMARY KEY (Delay));
```

- Attributes

Table	Attribute	Description	Datatype	Default Value	Null
FLIGHT_PLAN	Flightnumber	Flight identifier ex) 074, 1181	3 or 4-digit INT	·	Not Null
	Schedtime	Scheduled flight departure time ex) 1130 (11:30am), 2130 (9:30pm)	4-digit INT	·	Not Null
	Deptime	Actual flight departure time ex) 1135 (11:35am), 2137 (9:37pm)	4-digit INT	·	Not Null
	Tailnu	mentioned in 'AIRPLANE' table details			
	Date	mentioned in 'CALENDAR' table details			
	Journey_ID	mentioned in 'JOURNEY' table details			
	Delay	Mentioned in 'DELAY' table details			
AIRPLANE	Tailnu	Airplane identifier number ex) N225DL, N918DE	6-letter CHARACTER	·	Not Null
	Carrier	Carrier identifier to check the owned company ex) DL, DH	2-letter CHARACTER	·	Not Null
JOURNEY	Journey_ID	Flight route identifier ex) 1, 2, 3, …	INT	·	Not Null
	Origin	Origin airport ex) IAD, EWR	3-letter CHARACTER	·	Not Null
	Dest	Destination airport ex) JFK, LGA	3-letter CHARACTER	·	Not Null
	Distance	Distance between the origin and destination ex) 111, 291	3-digit INT	·	Not Null
CALENDAR	Date	Scheduled flight date ex) 1/24/2004, 3/1/2004	DATE	·	Not Null
	Dayweek	Day of the week of scheduled flight date ex) 1 (Monday), 7 (Sunday)	1-digit INT	·	Not Null
	Daymonth	Month of a row's corresponding date ex) 5 (5th), 23 (23th)	1 or 2-digit INT	·	Not Null
	Weather	mentioned in 'WEATHER' table details			
WEATHER	Weather	Weather information on a scheduled flight date ex) 0 (good for takeoff), 1 (not good for takeoff)	BOOLEAN	·	Not Null
DELAY	Delay	If a schedule is delayed ex) 0 (on time) or 1 (delay)	BOOLEAN	·	Not Null

- Primary and Foreign Keys
    - Primary keys for each table

Table	Primary key	Datatype
AIRPLANE	Talinu	CHAR(6)
WEATHER	Weather	BOOLEAN
JOURNEY	Journey_ID	INT
DELAY	Delay	BOOLEAN
CALENDAR	Date	DATE
FLIGHT_PLAN	Flightnumber	INT

- Foreign keys

Relationship	Foreign key	Reference
CALENDAR → WEATHER	Weather	WEATHER(Weather)
FLIGHT_PLAN → AIRPLANE	Tailnu	AIRPLANE(Tailnu)
FLIGHT_PLAN → CALENDAR	Date	CALENDAR(Date)
FLIGHT_PLAN → JOURNEY	Journey_ID	JOURNEY(Journey_ID)
FLIGHT_PLAN → DELAY	Delay	

- Actions taken on foreign keys when the corresponding primary key is deleted

Foreign key	Actions
Weather	Corresponding rows' Weather values automatically deleted
Tailnu	Corresponding rows' Tailnu values automatically deleted
Date	Corresponding rows' Date values automatically deleted
Journey_ID	Corresponding rows' Journey_ID values automatically deleted
Delay	Corresponding rows' Delay values automatically deleted

- Records Insertion (Three examples)

AIRPLANE TABLE

	tailnu [PK] character (6)	carrier character (2)
1	N940CA	OH
2	N405FJ	DH
3	N695BR	DL
4	N662BR	MQ
5	N698BR	UA
6	N687BR	US
7	N321UE	RU

JOURNEY TABLE

	journey_id [PK] integer	origin character (3)	dest character (3)	distance integer
1	1	BWI	JFK	184
2	2	DCA	LGA	213
3	3	IAD	EWR	229
4	4	DCA	LGA	214
5	5	BWI	JFK	228
6	6	IAD	EWR	213
7	7	BWI	JFK	184

CALENDAR TABLE

	date [PK] date	dayweek integer	daymonth integer	weather boolean
1	2004-01-01	4	11	false
2	2004-01-02	5	15	true
3	2004-01-03	2	20	true
4	2004-01-04	6	23	false
5	2004-01-05	7	17	false
6	2004-01-06	2	10	false
7	2004-01-07	5	12	true

WEATHER TABLE

	weather [PK] boolean
1	false
2	true