Project 1 - Airline Database

ISTM 622 - 601

Group 1-3

Brian Newman, Bhavishya Tyagi, Chuchu Yao, Pooja Vaswani, Anjali Shukla

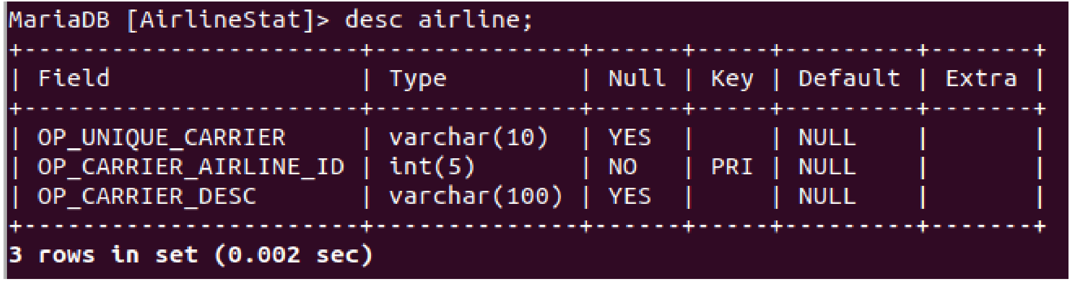
Create Database:

```
Create database AirlineStat;
Use AirlineStat;
```

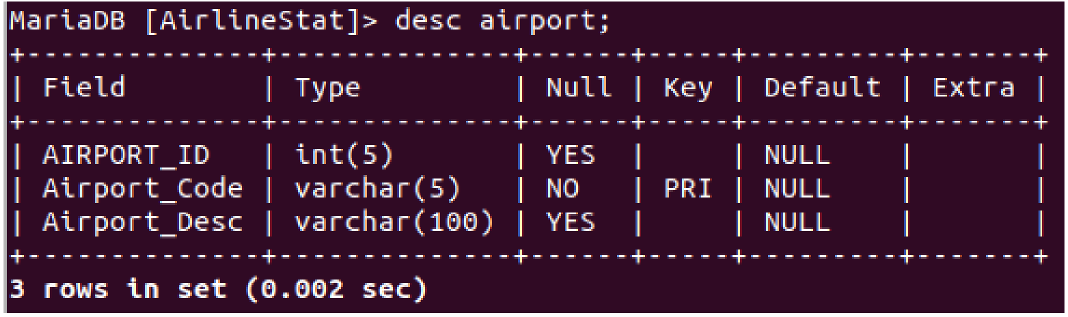
Create Table:

```
CREATE TABLE airline(
OP_UNIQUE_CARRIER  VARCHAR(10),
OP_CARRIER_AIRLINE_ID  INT(5),
OP_CARRIER_DESC VARCHAR(100),
 PRIMARY KEY (OP_CARRIER_AIRLINE_ID)
);


CREATE TABLE airport(
AIRPORT_ID INT(5),
Airport_Code VARCHAR(5),
Airport_Desc VARCHAR(100),
PRIMARY KEY (Airport_Code )
);


CREATE TABLE flight(
FL_DATE DATE,
OP_CARRIER_AIRLINE_ID INT(5),
OP_CARRIER_FL_NUM INT(5),
ORIGIN_Airport_Code VARCHAR(5),
DEST_Airport_Code VARCHAR(5),
CANCELLED INT(1),
DIVERTED INT(1),
ACTUAL_ELAPSED_TIME INT(6),
AIR_TIME INT(6),
FLIGHTS INT(1),
DISTANCE  INT(10) ,
CARRIER_DELAY   INT(10) ,
WEATHER_DELAY  INT(10) ,
NAS_DELAY INT(10) ,
SECURITY_DELAY  INT(10) ,
LATE_AIRCRAFT_DELAY  INT(10),
PRIMARY KEY (FL_DATE, OP_CARRIER_AIRLINE_ID , OP_CARRIER_FL_NUM, ORIGIN_Airport_Code,
DEST_Airport_Code, ACTUAL_ELAPSED_TIME),
FOREIGN KEY(OP_CARRIER_AIRLINE_ID ) REFERENCES airline(OP_CARRIER_AIRLINE_ID ),
FOREIGN KEY(ORIGIN_Airport_Code) REFERENCES airport(Airport_Code),
FOREIGN KEY(DEST_Airport_Code ) REFERENCES airport(Airport_Code)
);
```

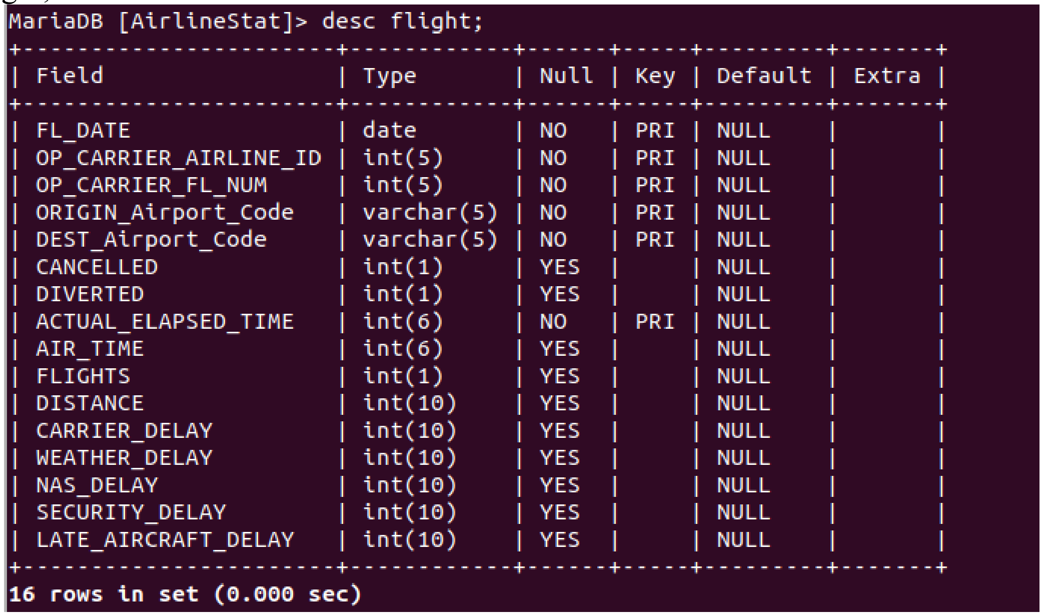
Show Structure:

- desc airline;
  - 

```
MariaDB [AirlineStat]> desc airline;
+-----------------------+--------------+------+-----+---------+-------+
| Field                 | Type         | Null | Key | Default | Extra |
+-----------------------+--------------+------+-----+---------+-------+
| OP_UNIQUE_CARRIER     | varchar(10)  | YES  |     | NULL    |       |
| OP_CARRIER_AIRLINE_ID | int(5)       | NO   | PRI | NULL    |       |
| OP_CARRIER_DESC       | varchar(100) | YES  |     | NULL    |       |
+-----------------------+--------------+------+-----+---------+-------+
3 rows in set (0.002 sec)
```

- desc airport;
  - 

```
MariaDB [AirlineStat]> desc airport;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| AIRPORT_ID   | int(5)       | YES  |     | NULL    |       |
| Airport_Code | varchar(5)   | NO   | PRI | NULL    |       |
| Airport_Desc | varchar(100) | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
3 rows in set (0.002 sec)
```

- desc flight;
  - 

```
MariaDB [AirlineStat]> desc flight;
+-----------------------+------------+------+-----+---------+-------+
| Field                 | Type       | Null | Key | Default | Extra |
+-----------------------+------------+------+-----+---------+-------+
| FL_DATE               | date       | NO   | PRI | NULL    |       |
| OP_CARRIER_AIRLINE_ID | int(5)     | NO   | PRI | NULL    |       |
| OP_CARRIER_FL_NUM     | int(5)     | NO   | PRI | NULL    |       |
| ORIGIN_Airport_Code   | varchar(5) | NO   | PRI | NULL    |       |
| DEST_Airport_Code     | varchar(5) | NO   | PRI | NULL    |       |
| CANCELLED             | int(1)     | YES  |     | NULL    |       |
| DIVERTED              | int(1)     | YES  |     | NULL    |       |
| ACTUAL_ELAPSED_TIME   | int(6)     | NO   | PRI | NULL    |       |
| AIR_TIME              | int(6)     | YES  |     | NULL    |       |
| FLIGHTS               | int(1)     | YES  |     | NULL    |       |
| DISTANCE              | int(10)    | YES  |     | NULL    |       |
| CARRIER_DELAY         | int(10)    | YES  |     | NULL    |       |
| WEATHER_DELAY         | int(10)    | YES  |     | NULL    |       |
| NAS_DELAY             | int(10)    | YES  |     | NULL    |       |
| SECURITY_DELAY        | int(10)    | YES  |     | NULL    |       |
| LATE_AIRCRAFT_DELAY   | int(10)    | YES  |     | NULL    |       |
+-----------------------+------------+------+-----+---------+-------+
16 rows in set (0.000 sec)
```

Combining CSV:

- This is because the files were split by month.

copy *.csv combinedData.csv

- This is command prompt on Windows.

Transforming the Data:

Powershell Script: Change Empty to 0

```
  $inFilePath  = "D:\Google Drive\_Fall  2019\ISTM 622\Group1-3\Newdataset\flightcase.csv"
  $outFilePath = "D:\Google Drive\_Fall  2019\ISTM 622\Group1-3\Newdataset\flightcaseEDIT.csv"

  Import-Csv $inFilePath | % {
     if (-not $_.CANCELLED     )
    { $_.CANCELLED     = 0 }
    if (-not $_.DIVERTED )
    { $_.DIVERTED  = 0 }
    if (-not $_.ACTUAL_ELAPSED_TIME  )
    { $_.ACTUAL_ELAPSED_TIME   = 0 }
    if (-not $_.AIR_TIME )
    { $_.AIR_TIME  = 0 }
    if (-not $_.FLIGHTS )
    { $_.FLIGHTS  = 0 }
    if (-not $_.DISTANCE )
    { $_.DISTANCE  = 0 }
    if (-not $_.CARRIER_DELAY)
    { $_.CARRIER_DELAY = 0 }
    if (-not $_.WEATHER_DELAY)
    { $_.WEATHER_DELAY = 0 }
    if (-not $_.NAS_DELAY)
    { $_.NAS_DELAY = 0 }
    if (-not $_.SECURITY_DELAY)
    { $_.SECURITY_DELAY = 0 }
    if (-not $_.LATE_AIRCRAFT_DELAY)
    { $_.LATE_AIRCRAFT_DELAY = 0 }
    $_   # echo all records, so they can be exported back to a file
  } | Export-Csv $outFilePath -NoType
```

Loading the Data:

```
LOAD DATA INFILE '/home/big/Desktop/dataset/airline.csv'
INTO TABLE airline
FIELDS TERMINATED BY ','
optionally enclosed BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES;

LOAD DATA INFILE '/home/big/Desktop/dataset/airport.csv'
INTO TABLE airport
FIELDS TERMINATED BY ','
optionally enclosed BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES;

LOAD DATA INFILE '/home/big/Desktop/dataset/Final17.csv'
INTO TABLE flight
FIELDS TERMINATED BY ','
optionally ENCLOSED BY '"'
```

```
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES;
```

Loading the Data:

```
LOAD DATA INFILE '/home/big/Desktop/dataset/Final18.csv'
INTO TABLE flight
FIELDS TERMINATED BY ','
optionally ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES;

LOAD DATA INFILE '/home/big/Desktop/dataset/Final19.csv'
INTO TABLE flight
FIELDS TERMINATED BY ','
optionally ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES;
```

Connecting to Database with Python:

- Run the commands on the cluster
  - sudo apt-get libmariadb-dev
  - CREATE USER 'root'@'10.20.0.167' IDENTIFIED BY '1234';
  - GRANT ALL PRIVILEGES ON *.* TO 'root'@'10.20.0.167';

HTML:

```
<!DOCTYPE html>
<html>
<head>
      <title>AirLine</title>
</head>
<!-- Adding some style to table -->
<style type="text/css">
   th:tr{color: blue;}
   tr:nth-of-type(2n){border: 1px solid black;background-color: rgba(150, 150, 150, 0.5);}
   td{padding: 8px 8px;border: 1px solid black;}
   body{background-color: powderblue;text-align: center;color: black;font-family: Arial,Helvetica,
     sans-serif;}
</style>
<body>
      <h1>Airline Stats </h1>
      <h2>Data is from January to June from 2017,2018,2019! </h2>
      <p>This is a dashboard!</p>

<table style="margin-left: 20px;">
<!-- Table headers for Query1 -->
   <th>
```

```
       <tr style="color: Black; font-weight: bold; ">
         <td>Average flight delays</td>
        </tr>
     <tr style="color: Black; ">
         <td>{{ query1 }} Minutes</td>
        </tr>
   </th>
</table>
<table style="margin-left: 20px;">
<!-- Table headers for Query2 -->
   <th>
       <tr style="color: Black; font-weight: bold; ">
         <td>Showing number of flights whose distance is greater than 1000 that were
             cancelled.</td>
        </tr>
     <tr style="color: Black; ">
         <td>{{ query2 }}</td>
        </tr>
   </th>
</table>
<table style="margin-left: 20px;">
<!-- Table headers for Query3 -->
   <th>
       <tr style="color: Black; font-weight: bold; ">
         <td>Showing number of flights whose distance is less than 1000 that were cancelled.</td>
        </tr>
     <tr style="color: Black; ">
         <td>{{ query3 }}</td>
        </tr>
   </th>
</table>
<table style="margin-left: 20px;">
<!-- Table headers for Query4 -->
   <th>
       <tr style="color: Black; font-weight: bold; ">
         <td colspan="2">Showing the number of cancelled flights from each location.</td>
         </tr>
       </th>
   <!-- For loop logic of jinja template -->
   {%for i in query4%}
   <!-- table rows -->
   <tr>
        <td>{{i[0]}}</td>
        <td>{{i[1]}}</td>
        {%endfor%}
   </tr>
</table>
<table style="margin-left: 20px;">
<p></p>
```

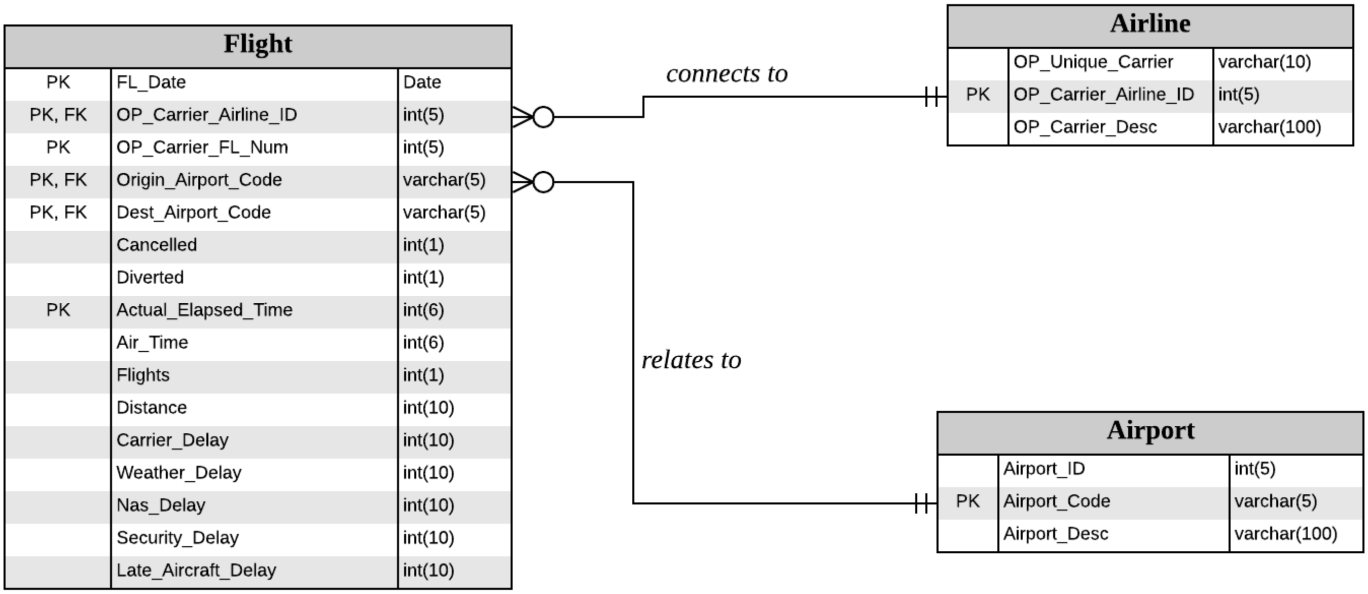
```
<!-- Table headers for Query5 -->
   <th>
       <tr style="color: Black; font-weight: bold;">
         <td colspan="5">Showing flight delayed by categories in minutes</td>
       </tr>
       <tr style="color: Black; font-weight: bold;">
           <td>Carrier Delay</td>
           <td>Weather Delay</td>
           <td>NAS Delay</td>
           <td>Security Delay</td>
           <td>Late Craft Delay</td>
         </tr>
   </th>
<!-- For loop logic of jinja template -->
{%for i in query5%}
   <!-- table rows -->
   <tr>
        <td>{{i[0]}}</td>
        <td>{{i[1]}}</td>
       <td>{{i[2]}}</td>
       <td>{{i[3]}}</td>
       <td>{{i[4]}}</td>
        {%endfor%}
   </tr>
</table>
<p></p>
<table style="margin-left: 20px;">
<!-- Table headers for Query6 -->
   <th>
       <tr style="color: Black; font-weight: bold;">
         <td colspan="5">Showing flight delayed by categories in numbers</td>
       </tr>
       <tr style="color: Black; font-weight: bold;">
           <td>Carrier Delay</td>
           <td>Weather Delay</td>
           <td>NAS Delay</td>
           <td>Security Delay</td>
           <td>Late Craft Delay</td>
         </tr>
   </th>
   <tr>
        <td>{{query6}}</td>
        <td>{{query7}}</td>
       <td>{{query8}}</td>
       <td>{{query9}}</td>
       <td>{{query10}}</td>
    </tr>
</table>
</body>
</html>
```

Python Application and Queries:

```
from flask import Flask, render_template, request
import pusher

app = Flask(__name__)

channels_client = pusher.Pusher(
 app_id='865352',
 key='2929f012f8448e1e7297',
 secret='3718786561642f32bace',
 cluster='us2',
 ssl=True
)
##################################################################################################
import mysql.connector as mariadb

mariadb_connection = mariadb.connect(user='root', password='1234', host='10.20.0.167',
database='AirlineStat')
cursor = mariadb_connection.cursor()

cursor.execute("Select
AVG(CARRIER_DELAY+WEATHER_DELAY+NAS_DELAY+SECURITY_DELAY+LATE_AIRCRAFT_DELAY ) from flight;")
query1 = cursor.fetchone()
query1 = query1[0]
print(query1)

cursor.execute("Select COUNT(CANCELLED) from flight where CANCELLED=1 and DISTANCE>1000;")
query2 = cursor.fetchone()
query2 = query2[0]
print(query2)

cursor.execute("Select COUNT(CANCELLED) from flight where CANCELLED=1 and DISTANCE<=1000;")
query3 = cursor.fetchone()
query3 = query3[0]
print(query3)

cursor.execute("Select ORIGIN_Airport_Code, COUNT(CANCELLED) FROM flight where CANCELLED=1 group
by ORIGIN_Airport_Code;")
query4 = cursor.fetchall()

cursor.execute("Select SUM(CARRIER_DELAY) as carrier_delay, SUM(WEATHER_DELAY) as
weather_delay,sum(NAS_DELAY) as nas_delay, "
              "sum(SECURITY_DELAY) as security_delay,sum(LATE_AIRCRAFT_DELAY) as late_craft_delay
From flight;")
query5 = cursor.fetchall()
```

```
cursor.execute("Select count(CARRIER_DELAY) as carrier_delay from flight where CARRIER_DELAY >0;")
query6 = cursor.fetchone()
query6 = query6[0]
print(query6)

cursor.execute("Select count(WEATHER_DELAY) as weather_delay from flight where WEATHER_DELAY >0;")
query7 = cursor.fetchone()
query7 = query7[0]
print(query7)

cursor.execute("select count(NAS_DELAY) as nas_delay from flight where NAS_DELAY>0;")
query8 = cursor.fetchone()
query8 = query8[0]
print(query8)

cursor.execute("select count(SECURITY_DELAY) as security_delay from flight where 
SECURITY_DELAY>0;")
query9 = cursor.fetchone()
query9 = query9[0]
print(query9)

cursor.execute("select count(LATE_AIRCRAFT_DELAY) as late_aircraft_delay from flight where 
LATE_AIRCRAFT_DELAY>0;")
query10 = cursor.fetchone()
query10 = query10[0]
print(query10)

##################################################################################################

@app.route('/')
def dashboard():
   # Display website
   #len = Length of list and then ports over
          return render_template('dashboard.html', query1 = query1, query2 = query2,
                                 query3 = query3, len = len(query4), query4 = query4,
                                 len1 = len(query5), query5 = query5, query6 = query6,
                                 query7 = query7, query8 = query8, query9 = query9,
                                 query10 = query10)
if __name__ == '__main__':
  app.run(debug=False)
```

ERD:



Flight

PK	FL_Date	Date
PK, FK	OP_Carrier_Airline_ID	int(5)
PK	OP_Carrier_FL_Num	int(5)
PK, FK	Origin_Airport_Code	varchar(5)
PK, FK	Dest_Airport_Code	varchar(5)
	Cancelled	int(1)
	Diverted	int(1)
PK	Actual_Elapsed_Time	int(6)
	Air_Time	int(6)
	Flights	int(1)
	Distance	int(10)
	Carrier_Delay	int(10)
	Weather_Delay	int(10)
	Nas_Delay	int(10)
	Security_Delay	int(10)
	Late_Aircraft_Delay	int(10)

Airline

	OP_Unique_Carrier	varchar(10)
PK	OP_Carrier_Airline_ID	int(5)
	OP_Carrier_Desc	varchar(100)

Airport

	Airport_ID	int(5)
PK	Airport_Code	varchar(5)
	Airport_Desc	varchar(100)

Cluster and Part 1 Demo:

- https://youtu.be/2XAHXM5ELnE