Project Report

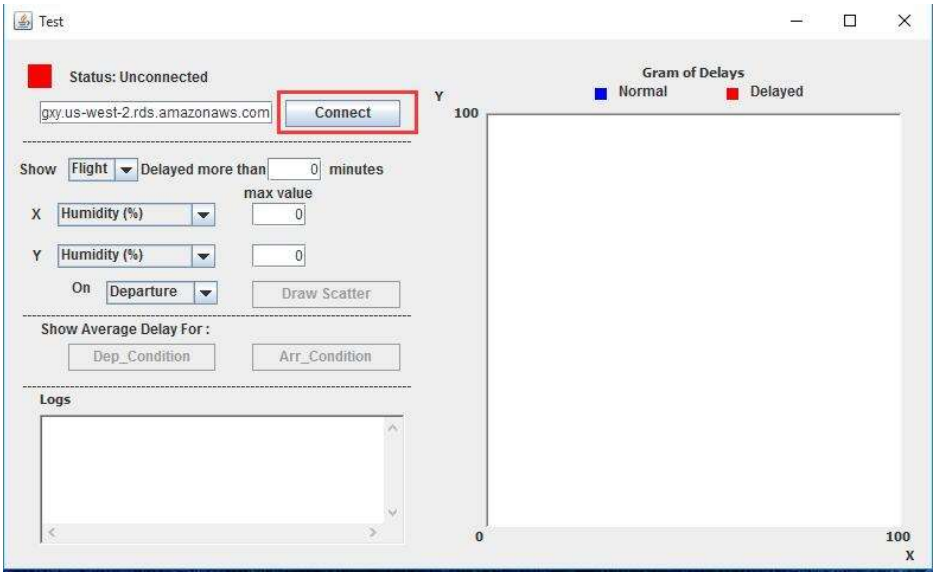
Mengna Lin 441399
Ziyang Liu 442873
Wenyan Cao 445783
Lina Su 445755

1. Readme – direction on how to run the project.

1.1 Front End:

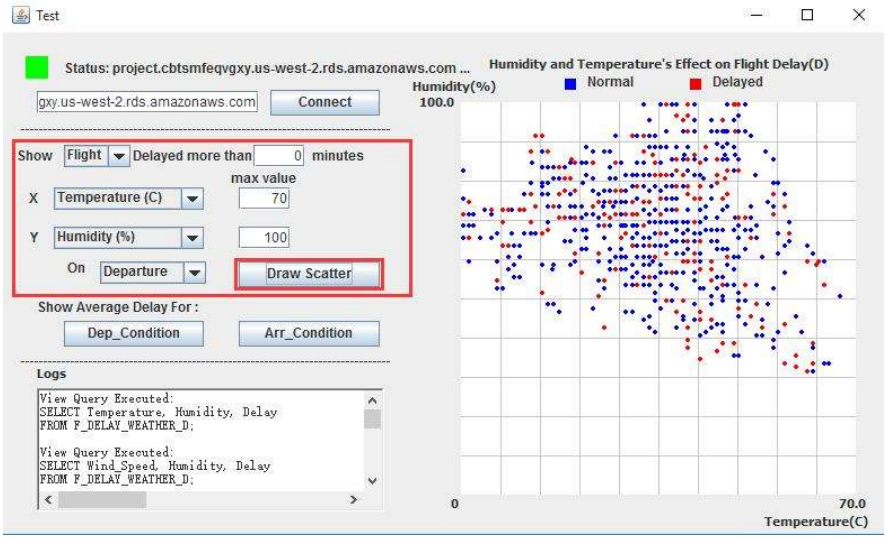
Our front program can be run on a machine with JRE installed.
(An runnable jar ProjectDeliverble.jar was uploaded to the Blackboard).

It can be run either by doubleclick, or use command line java -jar ProjectDeliverble.jar
After the launch, a interface should show. Clich “Connect” button to connect to our database.

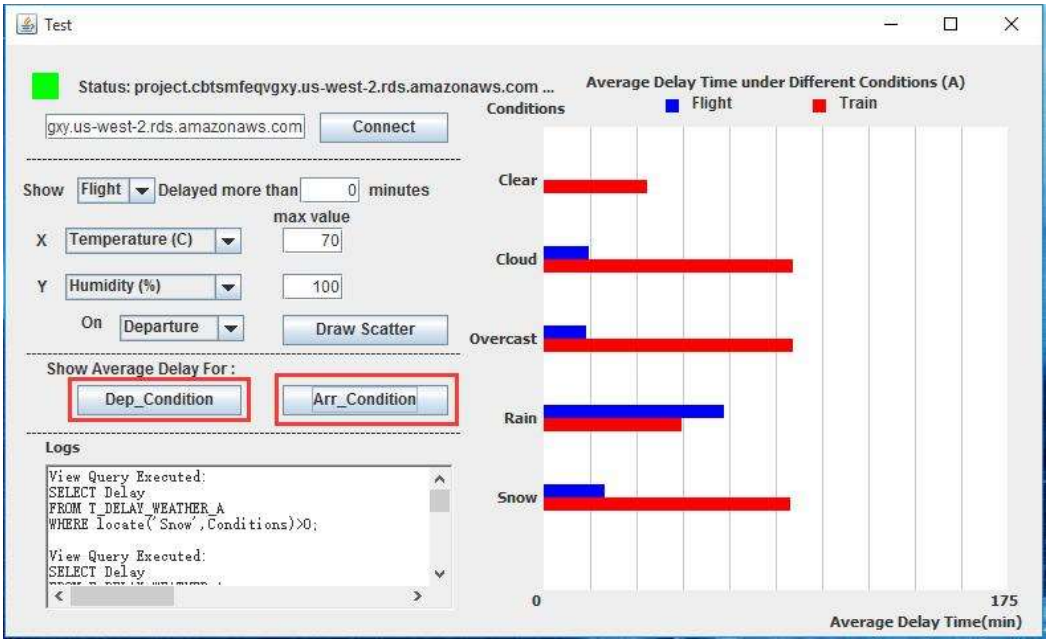


There should be no need to enter anything into the textbox on the left of the button.

To displace a scatter diagram for current data in the database, please select attributes, max values and the location, then click “Draw Scatter” button.



To displace a average delay data for current data in the database, click “xxx_Condition” button to show Average delays corresponds to depart or arrival location’s conditions.



1.2 Back End:

Our database can be accessed by any MySQL based console, application or API.
Such as MySQL Workbench (Recommended).

Or may be able to accessed through “Online Terminal” below:
http://www.tutorialspoint.com/mysql_terminal_online.php

The address & port is:

project.cbtsmfeqvgxy.us-west-2.rds.amazonaws.com:3306
Username: admin
Password: 12345678

OR (if not working):
Username: root
Password: 020736428526

The above users have admin access to the database.

Notes:
1) Sometimes the server cannot be connected. It may be in backup status.
2) After idle for some times, the connection could lost
3) The online database requires all table names are case sensitive. In our project, all table are in capital letters, like “FLIGHT”.

2. Project Description

2.1. Project Goal

The goal of our database is to analyze the weather’s effect on the reliability of different kinds of transportation. We mainly focus on helping people choose a better transportation way under different weather conditions. For example, under certain weather condition, useres can use the data we provide to predict the average delay of train or flight, so they can choose a better way of transport based on their own circumstances.

2.2. Data

2.2.1 The data we have:

We mainly have detail statistics of flight and train, which includes scheduled and actual arrival time, flight number and train number. Furthermore, we also have detail information of weather conditions, which includes humidity, visibility, precipitation, temperature, speed and direction of wind. We use city code to define the location, so time plus location would determine the weather information.

2.2.2 The data sources:

Flightstates flight delay
http://apps.bts.gov/xml/ontimesummarystatistics/src/dstat/OntimeSummaryArrivalsData.xml

Amtrak train delay
https://juckins.net/amtrak_status/archive/html/history.php

Weather Underground hourly weather data
https://www.wunderground.com/history/airport/KSFO/2016/1/1/DailyHistory.html

2.2.3 Size of data:

Hourly weather, flight, train and their operation record data within month 2016-01

2.2.4 Methods to scrape and clean data:

For Hourly Weather Data:
1. Downloaded the Comma Delimited Files for every day.

2. Use a .bat script “for %%i in (*.txt) do type %%i>>all.txt” To concaterate all those file into a single file
3. Load that file using excel, Change the format of incorperate data. (Such as replacing all “N/A” into ‘0’, Deleting unused columns, etc.)
4. Name the corrent sheet as “WEATHER” for script addressing later.
5. Use a excel function with the format below to generate our INSERT Statement:

```
=CONCATENATE("INSERT INTO WEATHER VALUES ('",
TEXT(INDIRECT("WEATHER!"&ADDRESS(ROW(),14)),"yyyy-mm-dd"),"','",
TEXT(INDIRECT("WEATHER!"&ADDRESS(ROW(),14)),"hh:mm"),"','STL',",
INDIRECT("WEATHER!"&ADDRESS(ROW(),2)),",",
INDIRECT("WEATHER!"&ADDRESS(ROW(),4)),",",
INDIRECT("WEATHER!"&ADDRESS(ROW(),6)),",",
INDIRECT("WEATHER!"&ADDRESS(ROW(),8)),",'",
INDIRECT("WEATHER!"&ADDRESS(ROW(),7)),"',",
INDIRECT("WEATHER!"&ADDRESS(ROW(),10)),",'",
INDIRECT("WEATHER!"&ADDRESS(ROW(),12)),"');")
```

6. Store the Generated INSERT statements.
   For example,
   {1:51 AM, 30.9, 23, 72, 30.4, 10, MNW, 9.2, - 0, ,Partly Cloudy, 240, 1/1/2016}
   yields
   "INSERT INTO WEATHER VALUES ('2016-01-02','07:51','STL',33.1,66,10,9.2,'West',0,'Clear');”

For Flight and Flight Operation Data:

1. Query info. of flights of a certain carrier in a given peroid, download its .csv file.
2. Format the data, convert “ “ into “” (delete duplicated spaces)
3. Name the sheet as “FLIGHT”
4. Use a Excel Function below to generate FLIGHT Insert Statements. The function (script) have the ability to calculate depart time from arrival time and duration. extract if the flight schedule pass a day from the information within the form.

   ```
   =CONCATENATE("INSERT INTO FLIGHT VALUES ('",.....)
   ……
   ```

   (Please refer to part 6.2 for details)
   For example,
   {AA,1/1/2015,96,N553AA,DFW,9:55,10:55,95,94,10,10:02,3,0,0,0,0,STL}
   yields
   INSERT INTO FLIGHT VALUES ('AA96','8:20','9:55','DFW','STL',0);

5. Use a Excel Function below to generate FLIGHT_Operation Insert Statements. The function (script) have the ability to calculate depart time from arrival time and duration and extract the name of flight delay reasons.

   ```
   =CONCATENATE("INSERT INTO FLIGHT_OPERATION VALUES ('",.....)
   ……
   ```

   (Please refer to part 6.2 for details)

   For example,
   {AA,1/1/2015,96,N553AA,DFW,9:55,10:55,95,94,10,10:02,3,0,0,0,0,STL}
   yields
   INSERT INTO FLIGHT_OPERATION VALUES ('AA756','2016-01-01','2016-01-01','13:36','15:30',NULL);

For Train and Train Operation Data:

1. For each train line, we need to goto Amtrak and search the depart and arrival information for a same train line base on their depart and arrive location. For example, for train 21, CHI to STL, we will fetch the departure information for CHI and arrival information to STL.
2. We concaterate the departure and arrival information together in Excel. (If there are no data missing, the data for a certain peroid of time should be well-alligned
3. We do some format on the data such as “2016-xx-xx xx:xx (day in a week)” to “2016-xx-xx”, and “##:##AM” to “##:## AM” (add a space).
4. Use the excel script below to generate our insert statements:

```
=CONCATENATE("INSERT INTO TRAIN_OPERATION VALUES ('",
INDIRECT("TRAIN!"&ADDRESS(ROW(),1)),
"','",
TEXT(INDIRECT("TRAIN!"&ADDRESS(ROW(),2)),"yyyy-mm-dd"),
"','",
TEXT(INDIRECT("TRAIN!"&ADDRESS(ROW(),8)),"yyyy-mm-dd"),
"','",
TEXT(INDIRECT("TRAIN!"&ADDRESS(ROW(),3)),"h:mm"),
"','",
TEXT(INDIRECT("TRAIN!"&ADDRESS(ROW(),9)),"h:mm"),
"');"
)
```

2.2.4 Filtering out unused data:

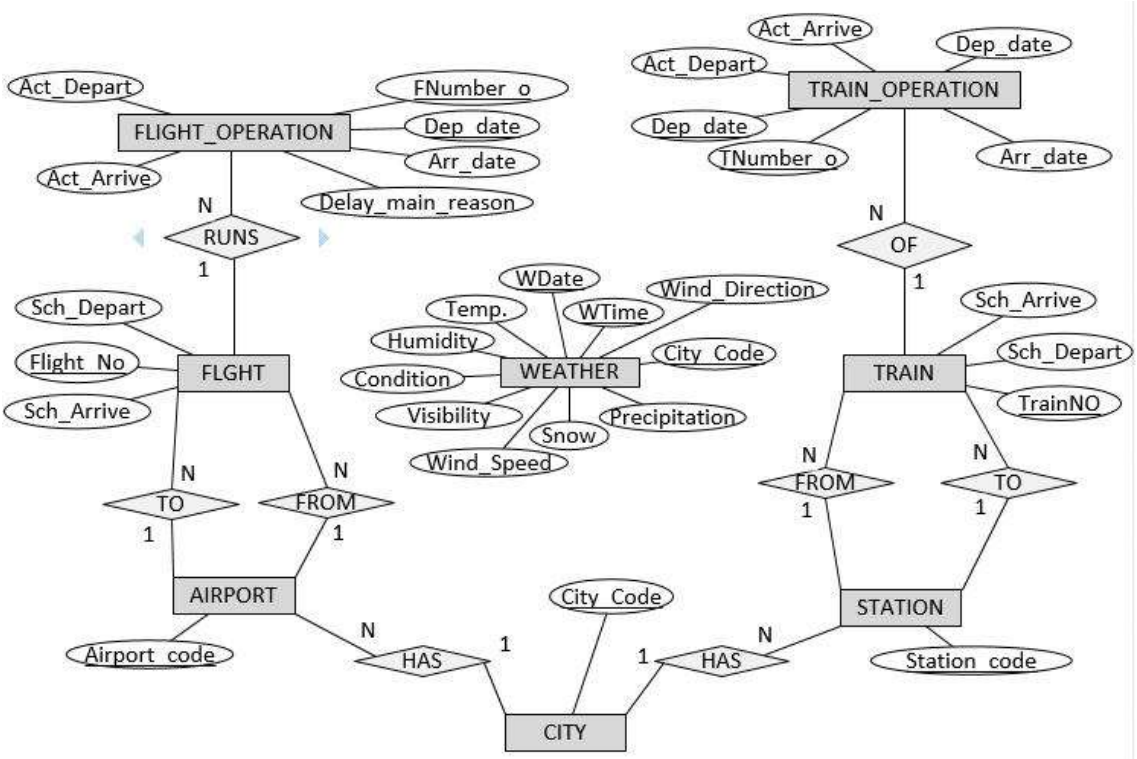
The databases contains a lot of flight or train data that corresponds to flights/ trains that we don’t really careabout. And some databases we use don’t have filter options. How we deal with those unused data depends of the number of tuples in an insert action. (We run scripts which contains INSERT statements to insert our data).

For those insert with a small amound (<500). We will run those INSERT statement anyway. The inserts of unused data will be blocked due to foreign key constrains. For example, if our table only contains airport STL and ORD, then all flight and flight operation records with a depart or destination airport other than “STL” and “ORD” will be blocked since the attribute “Dep_acode” and Arr_acode” have foreign key constrains from “AIRPORT.Airport_code”. If there is no such airport, there will be no such plane.

For those inserts with a large number (>500). We will use excel to remove unused data manually. Since within 10000 inserts, there could be only 500 that we need. In this case we choose to reduce the time for the server to process those unused inserts.

3. Data Modeling

3.1. ER Diagram



ER Diagram Descriptions

The ER Diagram above showed the model of our database. In this diagram, if one entity have more than one key attributes, that mean all those underlined attributes together form a primary key.

The ER Diagram are modified a lot of times during the project.
First, we pull all the “location” attributes as an entity to reduce the redundancy of the location informations.

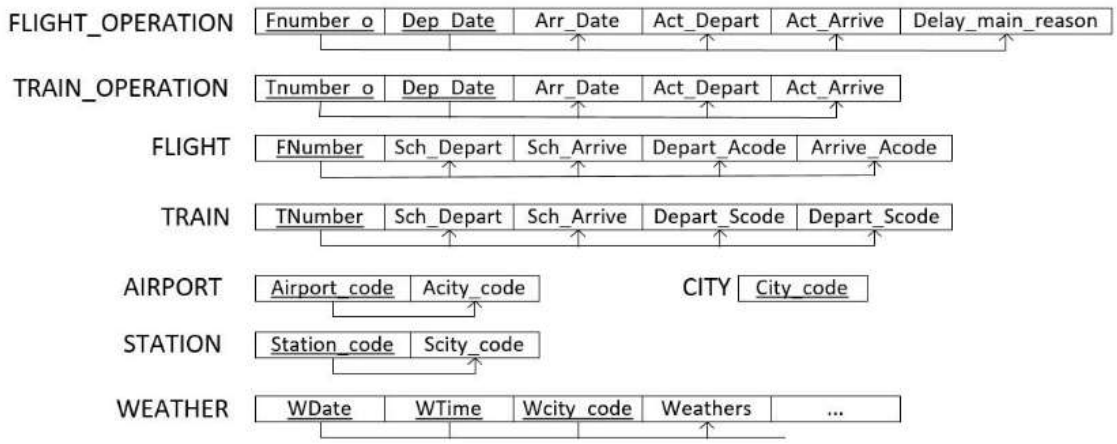
For the same reason, we pull out actual time attributes to “Operations”, leaving “FLIGHT” and “TRAIN” only static and fixed schedule data.

Then we splited “LOCATION” into “AIRPORT” and “STATION”. This is for establishing foreign key constrains. These constrains is good for us to control and control the number of locations we study.

We then extract CITY as a single entity from AIRPORT or STATION. This is for future update. There could be multiple airports or stations within a single city. Making CITY an entity will help us realize the support for those multiple airports or stations.

3.2 Normalized Table

Funtional Dependency between attriutes:



As shown about, there is no non-primary-key attributes offering functional dependency, and there is no non-prime atrributes partly or transitory depends on other attributs. So the database is in BCNF. This increase the complexity of selection, but reduced the data redundancy.

4. Programming

4.1 System description

Our program framework is basically a Java Swing program, plus the spporting package / APIs from MySQL to connect to the database.

Upon stablishing connection, the program will make a query to update a view which is closeley related to weather and traffics.

Then the later operations are mostly based on this view. In the later version, the create of view will be canceled from the application to reduce the load of server.

4.2 Sample SQL statements

4.2.1 View Update:

```
CREATE OR REPLACE VIEW DELAY_WEATHER AS
SELECT FNumber,
      Dep_date,
      Temperature,
      Humidity,
      Visibility,
      Wind_Speed,
      Conditions,
      Precipitation
      -TimeStampDiff(Minute,Act_Arrive,Sch_Arrive) AS Delayf_delay_weather_a
FROM flight JOIN Airport ON Depart_Acode=Airport_Code
          JOIN flight_operation ON FNumber=FNumber_o
          JOIN Weather ON ACity_code=WCity_code AND Dep_date=WDate AND HOUR(Act_Depart)=HOUR(WTime)
GROUP BY FNumber, Dep_Date
HAVING COUNT(*)>=1;
```

4.2.2 Query Delay Time

```
SELECT Humidity, Wind_speed, Delay
FROM F_DELAY_WEATHER_D;
```

4.2.3 Query Delays longer than 30 mins (Postgres Format)

```
SELECT TRAIN_OPERATION.tnumber,TRAIN_OPERATION.arr_date,Act_arrive-sch_arrive AS Time_Delayed
FROM TRAIN,TRAIN_OPERATION
WHERE tnumber=train_no
    AND Actual_arr-sch_arrive>'00:30:00'::time
```

4.3 Sample Script

Make query based on selection and draw result digram

Response Button “Execute”:

```
JButton btn_drawscatter = new JButton("Execute");
btn_drawscatter.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String X[] = ((String)select_factorX.getSelectedItem()).split("[\\(\\) ;]");
        double xmax = Double.valueOf(input_xmax.getText());
        String Y[] = ((String)select_factorY.getSelectedItem()).split("[\\(\\) ;]");
        double ymax = Double.valueOf(input_ymax.getText());
        String type = new String(((String)select_type.getSelectedItem()).substring(0,1));
        String loc = new String(((String)select_loc.getSelectedItem()).substring(0,1));
        drawScatterGram(type,loc,X[0],X[2],xmax,Y[0],Y[2],ymax);
    }
});
```

Query Execution: (SQL statments are created dynamically).

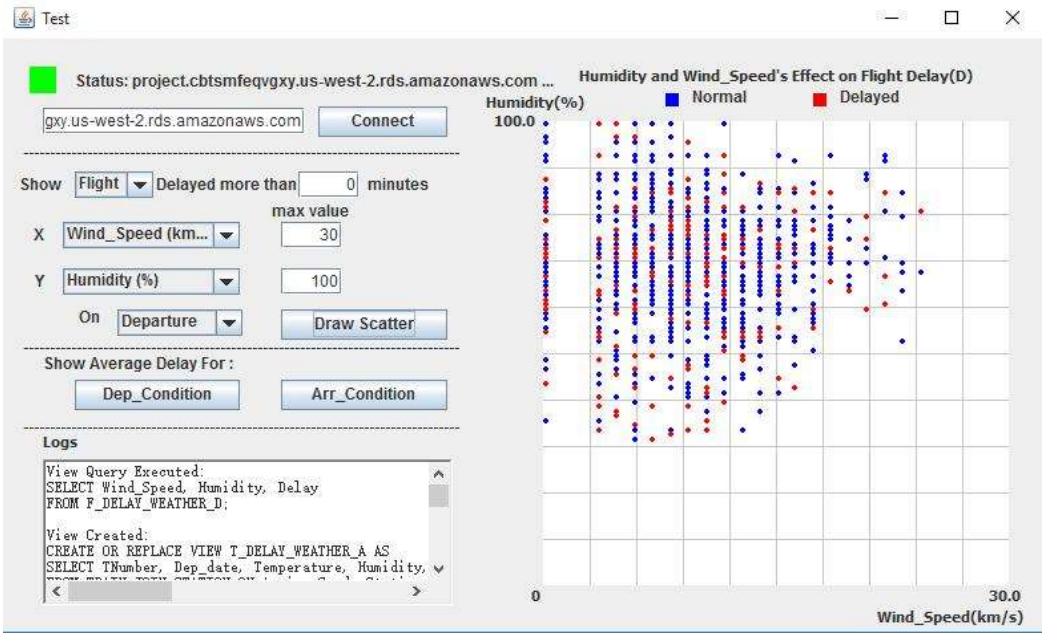
```
public void drawScatterGram(String type, String loc, String axisX, String unitX, Double maxX, String axi
    String sql = "SELECT "+axisX+", "+axisY +", Delay\n"
            +"FROM "+type+"_DELAY_WEATHER_"+loc;
    String type_name = new String();
    xlabel.setText(axisX+"("+unitX+")");
    ylabel.setText(axisY+"("+unitY+")");
    label_xmax.setText(maxX.toString());
    label_ymax.setText(maxY.toString());
    if(type.equals("T"))type_name="Train";
    else if(type.equals("F"))type_name="Flight";
    lblPleaseSelectItems.setText(axisY + " and "+axisX+"'s Effect on "+type_name+" Delay"+"("+loc+")");
    try
    {
        Statement statement = conn.createStatement();
        ResultSet rs = statement.executeQuery(sql);
        int w = panel_output.getWidth();
        double x = 0;
        double y = 0;
        int delay = 0;
        int thea = Integer.valueOf(input_delay.getText());
        Graphics2D dc = (Graphics2D) panel_output.getGraphics();
        dc.setColor(Color.WHITE);
        dc.fillRect(0, 0, w, w);
        dc.setColor(new Color(200,200,200));
        for(int i=w/10;i<w;i+=w/10)
        {
            dc.drawLine(0, i, w, i);
            dc.drawLine(i, 0, i, w);
        }
        while(rs.next())
        {
            x = rs.getDouble(axisX);
```

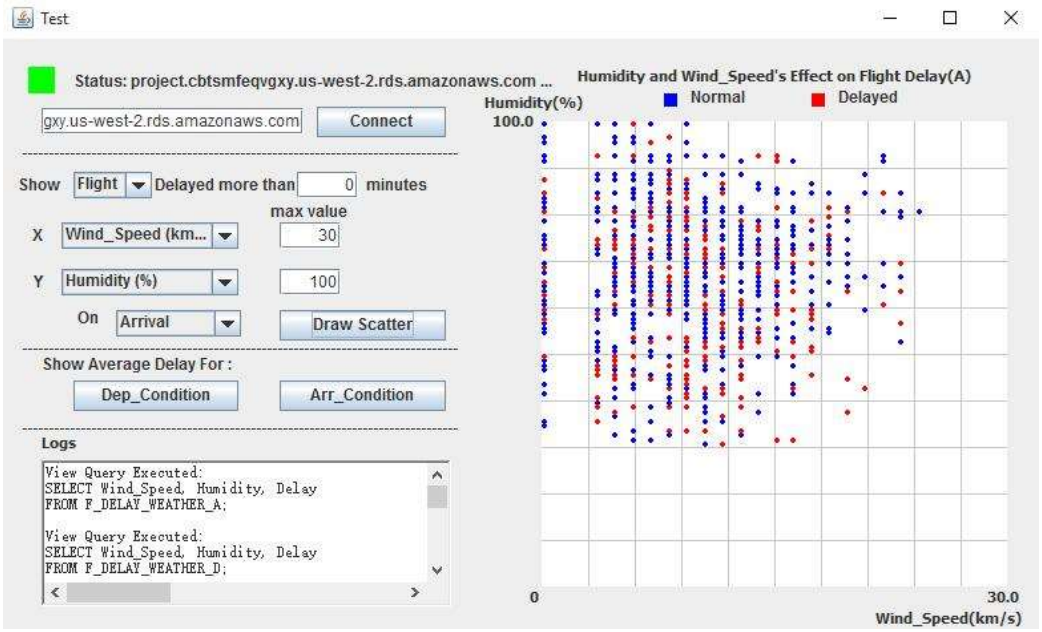
5. Conclusion

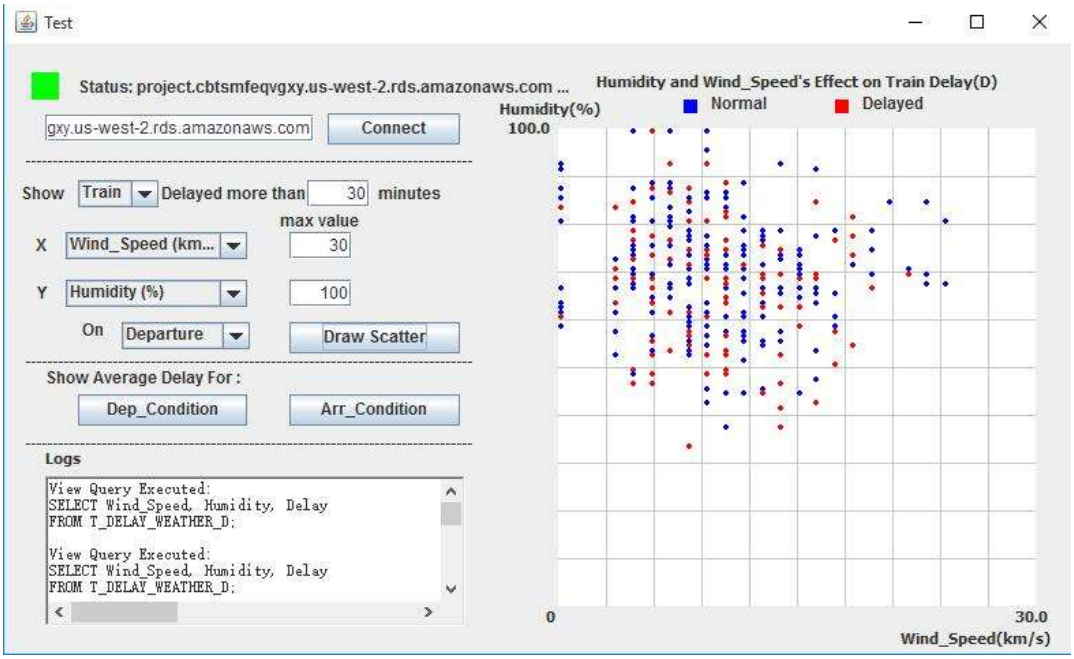
5.1 Data & Project Goal

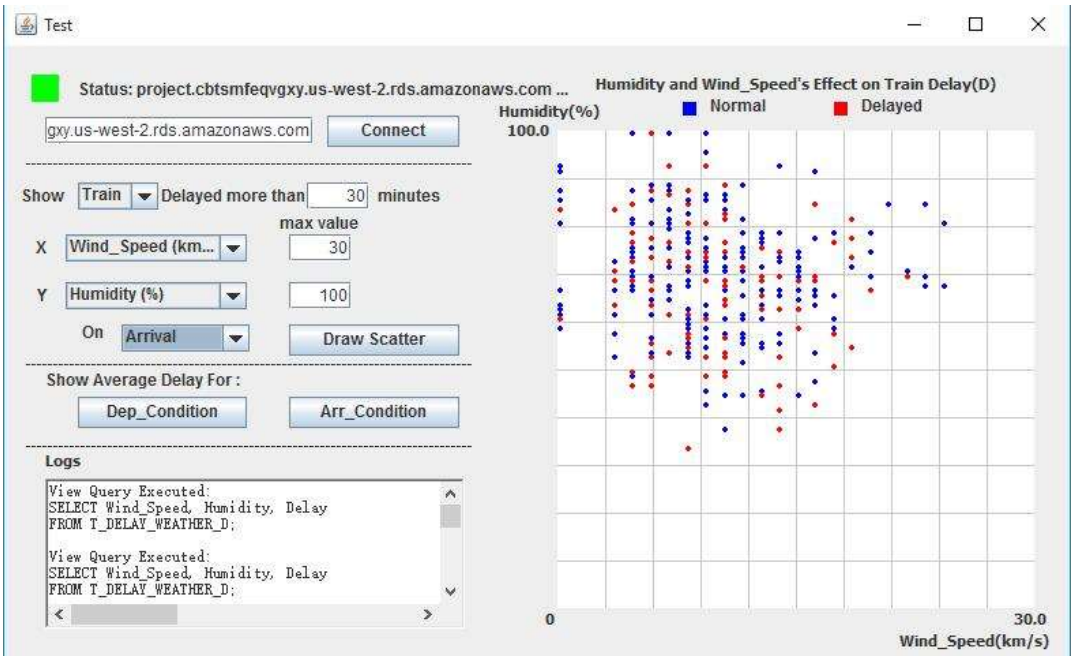
The data in our database can supported our project goal for some extend, but it is not very informative as we predicted.

From our diagrams, we can see that both flight delays as well as train delays shows some patterns corresopnd to different attributes of weather. And also, patterns for trains and flighs are different. This means flight and train will be influenced by weather in different ways. This means given a weather condition, there could be a “better way” to choose. People can use learning algorithms like logistic regression to produce a soft classfier, then apply the model on current weather data to choose which one is better. For example, 4 scatter point digrams below shows how the flight and train delay happens corresponds to wind speed and humidity at departrue and arrival location. The pattern is not as clear as we expected, but there do have some patterns and differences between these 4 diagrams.




Test
Status: project.cbtsmfeqvgxy.us-west-2.rds.amazonaws.com ...
gxy.us-west-2.rds.amazonaws.com
Connect
Show
Flight
Delayed more than
0
minutes
max value
X
Wind_Speed (km...
30
Y
Humidity (%)
100
On
Arrival
Draw Scatter
Show Average Delay For :
Dep_Condition
Arr_Condition
Logs
View Query Executed:
SELECT Wind_Speed, Humidity, Delay
FROM F_DELAY_WEATHER_A;
View Query Executed:
SELECT Wind_Speed, Humidity, Delay
FROM F_DELAY_WEATHER_D;
Humidity and Wind_Speed's Effect on Flight Delay(A)
Humidity(%)
Normal
Delayed
100.0
0
30.0
Wind_Speed(km/s)

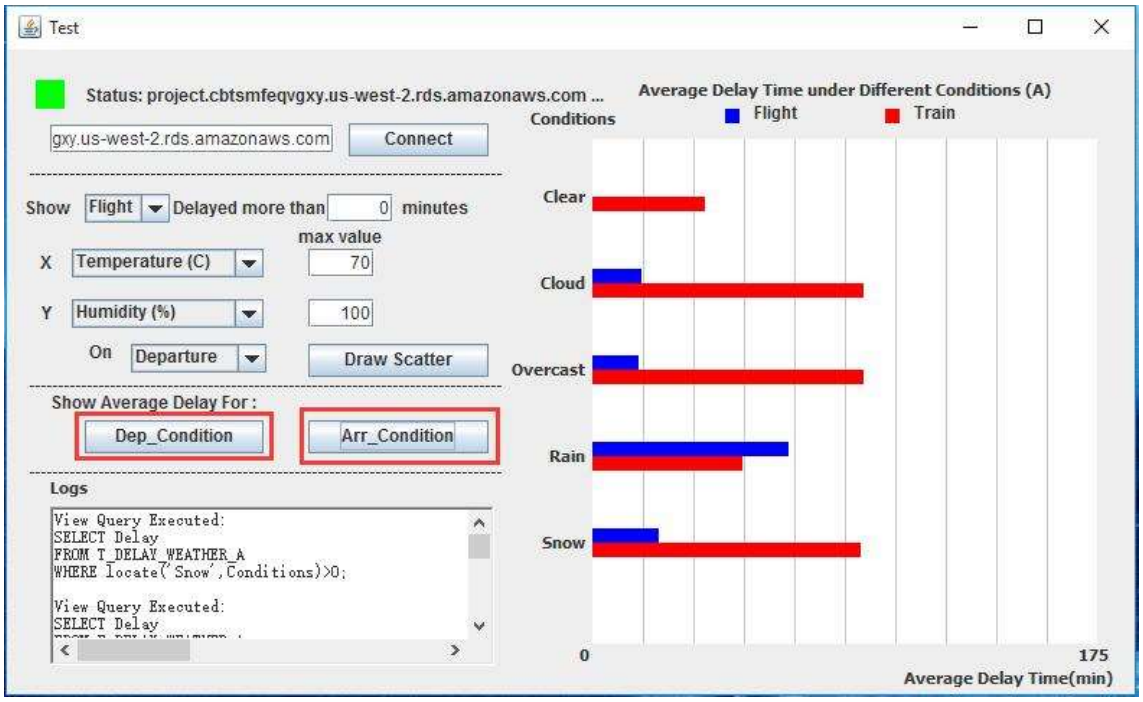

Test
Status: project.cbtsmfeqvgxy.us-west-2.rds.amazonaws.com ...
gxy.us-west-2.rds.amazonaws.com
Connect
Show
Train
Delayed more than
30
minutes
max value
X
Wind_Speed (km...
30
Y
Humidity (%)
100
On
Departure
Draw Scatter
Show Average Delay For :
Dep_Condition
Arr_Condition
Logs
View Query Executed:
SELECT Wind_Speed, Humidity, Delay
FROM T_DELAY_WEATHER_D;
View Query Executed:
SELECT Wind_Speed, Humidity, Delay
FROM T_DELAY_WEATHER_D;
Humidity and Wind_Speed's Effect on Train Delay(D)
Humidity(%)
Normal
Delayed
100.0
0
30.0
Wind_Speed(km/s)



As the figure shows, although there is no very clear patterns, but we can come to 2 conclutions:

1) Flight delay is more possible to have closer relations to weather conditions than trains
2) Low wind_speed during departure and high wind_speed during arrival may done some contribution to the flight delays
3) Flight delays may related to humidity

The unexpected part is the diagram of average delays corresponds to different conditions. For arrival conditions, it seems regular that the badder the condition, the longer the average delays. And train delays are much longer than flight delays (Thunderstorms didn't included because in current dataset there are very few trainw / planes depart / arrive exactly during a time when the conditons includes "Thunderstorm"). As shown below:



But the average delay times correspond to depart conditions seems abnormal.

But from these two diagram, we can see that flighs are far more reliable than trains. And fligh seldom delays for more than 30 minutes. As we continuely increase the amount of data, we expect more findings from those diagrams.

5.2 Additional Queries

Our database also support quires to show all the flight / train operation records that depart with a certain weather. For example, we can get all the flight number and depart date that depart in a clear weather:

```
SELECT FLIGHT_OPERATION.fnumber,FLIGHT_OPERATION.dep_date
FROM FLIGHT,FLIGHT_OPERATION,WEATHER,AIRPORT,CITY
WHERE fnumber=flight_no
        AND depart_acode=Airport_code
        AND AIRPORT.City_code=CITY.City_code
        AND AIRPORT.City_code=WEATHER.City_code
        AND WDate=dep_date
        AND EXTRACT(hour FROM WTime)=EXTRACT(hour FROM actual_dep)
        AND conditions='Clear';
```

When creating views, we can add more conditions to filter out part of the data. For example, the below query can create a view that include delay information and the corresponding weather attributes of the departing place. (But since weather will have effect on many delays and only a small subset of that are tagged “delay_main_reason=’weather’, this query is not included in our program

```
SELECT FNumber,
       Dep_date,
       Temperature,
       Humidity,
       Visibility,
       Wind_Speed,
       Conditions,
       Precipitation,
       -TimeStampDiff(Minute,Act_Arrive,Sch_Arrive) AS Delay
FROM FLIGHT JOIN AIRPORT ON Depart_Acode=Airport_Code
            JOIN FLIGHT_OPERATION ON FNumber=FNumber_o
            JOIN WEATHER ON ACity_code=WCity_code
                AND Dep_date=WDate
                AND HOUR(Act_Depart)=HOUR(WTime)
WHERE Delay_main_reason='Weather'
GROUP BY FNumber, Dep_Date
HAVING COUNT(*)>=1;
```

6. Appendix

6.1 Program Codes (Need external API Package : mysql-connector-java-5.1.38-bin.jar)

```
package stubs;

import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JPanel;
import java.awt.Color;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
```

```
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JLabel;
import javax.swing.SwingConstants;
import java.awt.TextArea;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.Graphics2D;

import javax.swing.border.BevelBorder;
import javax.swing.JScrollPane;
import javax.swing.JComboBox;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JTextField;
import java.awt.Component;
import javax.swing.Box;

public class TestFrame {

	private JFrame frmTest;
	static Connection conn;
	private JPanel panel_output;
	private JLabel lblConnStatus;
	static String host_add = "project.cbtsmfeqvgxy.us-west-2.rds.amazonaws.com";
	private JButton BtnConn;
	private TextArea text_result;
	private JLabel xlabel;
	private JLabel ylabel;
	private JLabel label_zero;
	private JLabel label_xmax;
	private JLabel label_ymax;
	private JPanel panel_red;
	private JPanel panel_blue;
	private JLabel lblDelayed;
	private JLabel lblNormal;
	private JLabel lblFnumber;
	private JLabel lblPleaseSelectItems;
	private JComboBox<String> select_factorY;
	private JTextField input_xmax;
	private JTextField input_ymax;
	private JTextField input_add;
	private JPanel led_conn;
	private String logstr = new String();
	private JTextField input_delay;
	private JLabel lblDelayedMoreThan;
	private JLabel lblMinutes;
	private JButton btn_drawscatter;
	private JButton btnAvgDelay;
	private JButton buttonColGram2;
	private JLabel labelc1;
	private JLabel labelc2;
	private JLabel labelc3;
	private JLabel labelc4;
	private JLabel labelc5;
	private JLabel lblShowAverageDelay;
	private JLabel label;
	private JLabel label_1;
	private JLabel label_2;
	/**
	 * Launch the application.
	 */
	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					TestFrame window = new TestFrame();
					window.frmTest.setVisible(true);

				} catch (Exception e) {
```

```
                                        e.printStackTrace();
                                }
                        }
                });

}

/**
 * Create the application.
 */
public TestFrame() {
        initialize();
}

/**
 * Initialize the contents of the frame.
 */
private void initialize() {
        frmTest = new JFrame();
        frmTest.setTitle("Test");
        frmTest.setBounds(100, 100, 798, 486);
        frmTest.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frmTest.getContentPane().setLayout(null);

        panel_output = new JPanel();
        panel_output.setBorder(new BevelBorder(BevelBorder.LOWERED, null, null, null, null));
        panel_output.setBackground(Color.WHITE);
        panel_output.setBounds(406, 62, 350, 350);
        frmTest.getContentPane().add(panel_output);

        lblConnStatus = new JLabel("Status: Unconnected");
        lblConnStatus.setBounds(54, 25, 365, 14);
        frmTest.getContentPane().add(lblConnStatus);

        input_add = new JTextField();
        input_add.setHorizontalAlignment(SwingConstants.RIGHT);
        input_add.setColumns(10);
        input_add.setBounds(30, 52, 197, 20);
        frmTest.getContentPane().add(input_add);

        input_add.setText(host_add);

        //====================== Connect to the database button ==================
        BtnConn = new JButton("Connect");
        BtnConn.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                        //Connect to the database
                        String add = input_add.getText();
                        String driver = "com.mysql.jdbc.Driver";
                        String url = new String();
                        if(add.length()>0)
                        {
                                url = "jdbc:mysql://"+add+":3306/traffic_weather";
                                host_add = add;
                        }
                        else
                                url = "jdbc:mysql://"+host_add+":3306/traffic_weather";
                        String user = "program";
                        String password = "12345678";

                        try{
                                Class.forName(driver);
                                conn = DriverManager.getConnection(url, user, password);
                                if(!conn.isClosed())
                                {
                                        showConnectStatus(true);
                                        logstr = "Connected.\n\n" + logstr;
                                        text_result.setText(logstr);
                                }
                        }
                        catch (Exception e1) {
```

```
                                e1.printStackTrace();
                        }
                        query_create_weaview();
                        btn_drawscatter.setEnabled(true);
                        btnAvgDelay.setEnabled(true);
                        buttonColGram2.setEnabled(true);
                }
});
BtnConn.setBounds(237, 51, 97, 23);
frmTest.getContentPane().add(BtnConn);

text_result = new TextArea();
text_result.setBounds(30, 317, 308, 110);
frmTest.getContentPane().add(text_result);

xlabel = new JLabel("X");
xlabel.setHorizontalAlignment(SwingConstants.RIGHT);
xlabel.setFont(new Font("Tahoma", Font.BOLD, 11));
xlabel.setBounds(575, 429, 192, 14);
frmTest.getContentPane().add(xlabel);

ylabel = new JLabel("Y");
ylabel.setFont(new Font("Tahoma", Font.BOLD, 11));
ylabel.setBounds(363, 41, 115, 14);
frmTest.getContentPane().add(ylabel);

label_zero = new JLabel("0");
label_zero.setFont(new Font("Tahoma", Font.BOLD, 11));
label_zero.setBounds(397, 413, 22, 14);
frmTest.getContentPane().add(label_zero);

label_xmax = new JLabel("100");
label_xmax.setHorizontalAlignment(SwingConstants.CENTER);
label_xmax.setFont(new Font("Tahoma", Font.BOLD, 11));
label_xmax.setBounds(724, 413, 59, 14);
frmTest.getContentPane().add(label_xmax);

label_ymax = new JLabel("100");
label_ymax.setHorizontalAlignment(SwingConstants.RIGHT);
label_ymax.setFont(new Font("Tahoma", Font.BOLD, 11));
label_ymax.setBounds(363, 55, 37, 14);
frmTest.getContentPane().add(label_ymax);

panel_red = new JPanel();
panel_red.setBackground(Color.RED);
panel_red.setBorder(null);
panel_red.setBounds(609, 41, 10, 10);
frmTest.getContentPane().add(panel_red);

panel_blue = new JPanel();
panel_blue.setBackground(Color.BLUE);
panel_blue.setBounds(498, 41, 10, 10);
frmTest.getContentPane().add(panel_blue);

lblDelayed = new JLabel("Delayed");
lblDelayed.setBounds(629, 37, 46, 14);
frmTest.getContentPane().add(lblDelayed);

lblNormal = new JLabel("Normal");
lblNormal.setBounds(518, 37, 46, 14);
frmTest.getContentPane().add(lblNormal);

lblFnumber = new JLabel("Logs");
lblFnumber.setFont(new Font("Tahoma", Font.BOLD, 11));
lblFnumber.setBounds(30, 297, 59, 14);
frmTest.getContentPane().add(lblFnumber);

lblPleaseSelectItems = new JLabel("Gram of Delays");
lblPleaseSelectItems.setFont(new Font("Tahoma", Font.BOLD, 11));
lblPleaseSelectItems.setHorizontalAlignment(SwingConstants.CENTER);
lblPleaseSelectItems.setBounds(406, 21, 350, 14);
```

```
                        frmTest.getContentPane().add(lblPleaseSelectItems);

                        JComboBox<String> select_factorX = new JComboBox<String>();
                        select_factorX.setModel(new DefaultComboBoxModel<String>(new String[] {"Humidity (%)", "Visibility (km)",
"Wind_Speed (km/s)", "Precipitation (m)", "Temperature (C)"}));
                        select_factorX.setBounds(45, 138, 133, 20);
                        frmTest.getContentPane().add(select_factorX);


                        select_factorY = new JComboBox<String>();
                        select_factorY.setModel(new DefaultComboBoxModel<String>(new String[] {"Humidity (%)", "Visibility (km)",
"Wind_Speed (km/s)", "Precipitation (m)", "Temperature (C)"}));
                        select_factorY.setBounds(45, 173, 133, 20);
                        frmTest.getContentPane().add(select_factorY);

                        input_xmax = new JTextField();
                        input_xmax.setText("0");
                        input_xmax.setHorizontalAlignment(SwingConstants.RIGHT);
                        input_xmax.setBounds(209, 138, 46, 20);
                        frmTest.getContentPane().add(input_xmax);
                        input_xmax.setColumns(10);

                        input_ymax = new JTextField();
                        input_ymax.setText("0");
                        input_ymax.setHorizontalAlignment(SwingConstants.RIGHT);
                        input_ymax.setColumns(10);
                        input_ymax.setBounds(209, 173, 46, 20);
                        frmTest.getContentPane().add(input_ymax);



                        JLabel lblMaxValue = new JLabel("max value");
                        lblMaxValue.setHorizontalAlignment(SwingConstants.CENTER);
                        lblMaxValue.setBounds(195, 123, 73, 14);
                        frmTest.getContentPane().add(lblMaxValue);

                        JLabel lblX = new JLabel("X");
                        lblX.setHorizontalAlignment(SwingConstants.CENTER);
                        lblX.setBounds(10, 141, 35, 14);
                        frmTest.getContentPane().add(lblX);

                        JLabel lblY = new JLabel("Y");
                        lblY.setHorizontalAlignment(SwingConstants.CENTER);
                        lblY.setBounds(10, 176, 35, 14);
                        frmTest.getContentPane().add(lblY);

                        JComboBox<String> select_loc = new JComboBox<String>();
                        select_loc.setModel(new DefaultComboBoxModel(new String[] {"Departure", "Arrival"}));
                        select_loc.setBounds(86, 204, 94, 20);
                        frmTest.getContentPane().add(select_loc);

                        JLabel lblOf = new JLabel("On");
                        lblOf.setHorizontalAlignment(SwingConstants.CENTER);
                        lblOf.setBounds(54, 203, 22, 14);
                        frmTest.getContentPane().add(lblOf);

                        JComboBox<String> select_type = new JComboBox<String>();
                        select_type.setModel(new DefaultComboBoxModel(new String[] {"Flight", "Train"}));
                        select_type.setBounds(54, 100, 59, 20);
                        frmTest.getContentPane().add(select_type);

                        JLabel lblType = new JLabel("Show");
                        lblType.setHorizontalAlignment(SwingConstants.CENTER);
                        lblType.setBounds(10, 103, 37, 14);
                        frmTest.getContentPane().add(lblType);

                        //=================================== Get Information and execute query
==========================================
                        btn_drawscatter = new JButton("Draw Scatter");
                        btn_drawscatter.addActionListener(new ActionListener() {
                                public void actionPerformed(ActionEvent e) {
```

```
                        String X[] = ((String)select_factorX.getSelectedItem()).split("[\\(\\) ;]");
                        double xmax = Double.valueOf(input_xmax.getText());
                        String Y[] = ((String)select_factorY.getSelectedItem()).split("[\\(\\) ;]");
                        double ymax = Double.valueOf(input_ymax.getText());
                        String type = new String(((String)select_type.getSelectedItem()).substring(0,1));
                        String loc = new String(((String)select_loc.getSelectedItem()).substring(0,1));
                        drawScatterGram(type,loc,X[0],X[2],xmax,Y[0],Y[2],ymax);
                }
});
btn_drawscatter.setBounds(209, 204, 125, 23);
frmTest.getContentPane().add(btn_drawscatter);
btn_drawscatter.setEnabled(false);

led_conn = new JPanel();
led_conn.setBackground(Color.RED);
led_conn.setForeground(Color.RED);
led_conn.setBounds(20, 21, 20, 20);
frmTest.getContentPane().add(led_conn);

input_delay = new JTextField();
input_delay.setText("0");
input_delay.setHorizontalAlignment(SwingConstants.RIGHT);
input_delay.setColumns(10);
input_delay.setBounds(222, 100, 46, 20);
frmTest.getContentPane().add(input_delay);

lblDelayedMoreThan = new JLabel("Delayed more than");
lblDelayedMoreThan.setHorizontalAlignment(SwingConstants.CENTER);
lblDelayedMoreThan.setBounds(111, 103, 115, 14);
frmTest.getContentPane().add(lblDelayedMoreThan);

lblMinutes = new JLabel("minutes");
lblMinutes.setHorizontalAlignment(SwingConstants.CENTER);
lblMinutes.setBounds(264, 103, 67, 14);
frmTest.getContentPane().add(lblMinutes);

btnAvgDelay = new JButton("Dep_Condition");
btnAvgDelay.setEnabled(false);
btnAvgDelay.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent arg0) {
                        int w = panel_output.getWidth();
                        int mul = 2;
                        Graphics2D dc = (Graphics2D) panel_output.getGraphics();
                        dc.setColor(Color.WHITE);
                        dc.fillRect(0, 0, w, w);
                        dc.setColor(new Color(200,200,200));
                        for(int i=w/10;i<w;i+=w/10)
                                dc.drawLine(i, 0, i, w);
                        lblPleaseSelectItems.setText("Average Delay Time under Different Conditions (D)");
                        lblNormal.setText("Flight");
                        lblDelayed.setText("Train");
                        xlabel.setText("Average Delay Time(min)");
                        ylabel.setText("Conditions");
                        label_xmax.setText(String.valueOf(w/mul));
                        label_ymax.setText("");
                        labelc1.setText("Clear");
                        labelc2.setText("Cloud");
                        labelc3.setText("Overcast");
                        labelc4.setText("Rain");
                        labelc5.setText("Snow");

                        drawColGram("Clear", "F", "D", 30, mul);
                        drawColGram("Clear", "T", "D", 40, mul);

                        drawColGram("Cloud", "F", "D", 90, mul);
                        drawColGram("Cloud", "T", "D", 100, mul);

                        drawColGram("Overcast", "F", "D", 150, mul);
                        drawColGram("Overcast", "T", "D", 160, mul);

                        drawColGram("Rain", "F", "D", 210, mul);
```

```
                        drawColGram("Rain", "T", "D", 220, mul);

                        drawColGram("Snow", "F", "D", 270, mul);
                        drawColGram("Snow", "T", "D", 280, mul);
                }
});
btnAvgDelay.setBounds(54, 257, 124, 23);
frmTest.getContentPane().add(btnAvgDelay);

buttonColGram2 = new JButton("Arr_Condition");
buttonColGram2.setEnabled(false);
buttonColGram2.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {
                int w = panel_output.getWidth();
                int mul = 2;
                Graphics2D dc = (Graphics2D) panel_output.getGraphics();
                dc.setColor(Color.WHITE);
                dc.fillRect(0, 0, w, w);
                dc.setColor(new Color(200,200,200));
                for(int i=w/10;i<w;i+=w/10)
                        dc.drawLine(i, 0, i, w);
                lblPleaseSelectItems.setText("Average Delay Time under Different Conditions (A)");
                lblNormal.setText("Flight");
                lblDelayed.setText("Train");
                xlabel.setText("Average Delay Time(min)");
                ylabel.setText("Conditions");
                label_xmax.setText(String.valueOf(w/mul));
                label_ymax.setText("");
                labelc1.setText("Clear");
                labelc2.setText("Cloud");
                labelc3.setText("Overcast");
                labelc4.setText("Rain");
                labelc5.setText("Snow");

                drawColGram("Clear", "F", "A", 30, mul);
                drawColGram("Clear", "T", "A", 40, mul);

                drawColGram("Cloud", "F", "A", 90, mul);
                drawColGram("Cloud", "T", "A", 100, mul);

                drawColGram("Overcast", "F", "A", 150, mul);
                drawColGram("Overcast", "T", "A", 160, mul);

                drawColGram("Rain", "F", "A", 210, mul);
                drawColGram("Rain", "T", "A", 220, mul);

                drawColGram("Snow", "F", "A", 270, mul);
                drawColGram("Snow", "T", "A", 280, mul);
        }
});
buttonColGram2.setBounds(209, 257, 125, 23);
frmTest.getContentPane().add(buttonColGram2);

labelc1 = new JLabel("");
labelc1.setHorizontalAlignment(SwingConstants.RIGHT);
labelc1.setFont(new Font("Tahoma", Font.BOLD, 11));
labelc1.setBounds(328, 95, 73, 14);
frmTest.getContentPane().add(labelc1);

labelc2 = new JLabel("");
labelc2.setHorizontalAlignment(SwingConstants.RIGHT);
labelc2.setFont(new Font("Tahoma", Font.BOLD, 11));
labelc2.setBounds(328, 155, 73, 14);
frmTest.getContentPane().add(labelc2);

labelc3 = new JLabel("");
labelc3.setHorizontalAlignment(SwingConstants.RIGHT);
labelc3.setFont(new Font("Tahoma", Font.BOLD, 11));
labelc3.setBounds(328, 215, 73, 14);
frmTest.getContentPane().add(labelc3);
```

```
		labelc4 = new JLabel("");
		labelc4.setHorizontalAlignment(SwingConstants.RIGHT);
		labelc4.setFont(new Font("Tahoma", Font.BOLD, 11));
		labelc4.setBounds(328, 275, 73, 14);
		frmTest.getContentPane().add(labelc4);

		labelc5 = new JLabel("");
		labelc5.setHorizontalAlignment(SwingConstants.RIGHT);
		labelc5.setFont(new Font("Tahoma", Font.BOLD, 11));
		labelc5.setBounds(328, 335, 73, 14);
		frmTest.getContentPane().add(labelc5);

		lblShowAverageDelay = new JLabel("Show Average Delay For :");
		lblShowAverageDelay.setHorizontalAlignment(SwingConstants.LEFT);
		lblShowAverageDelay.setBounds(30, 238, 282, 14);
		frmTest.getContentPane().add(lblShowAverageDelay);

		label = new JLabel("----------------------------------------------------------------------------------------");
		label.setHorizontalAlignment(SwingConstants.CENTER);
		label.setBounds(10, 225, 338, 14);
		frmTest.getContentPane().add(label);

		label_1 = new JLabel("----------------------------------------------------------------------------------------");
		label_1.setHorizontalAlignment(SwingConstants.CENTER);
		label_1.setBounds(10, 78, 338, 14);
		frmTest.getContentPane().add(label_1);

		label_2 = new JLabel("----------------------------------------------------------------------------------------");
		label_2.setHorizontalAlignment(SwingConstants.CENTER);
		label_2.setBounds(10, 285, 338, 14);
		frmTest.getContentPane().add(label_2);
	}

	public void showConnectStatus(boolean status){
		if(status==true)
		{
			this.lblConnStatus.setText("Status: "+host_add + " Connected");
			led_conn.setBackground(Color.GREEN);
		}
		else
			this.lblConnStatus.setText("Status: Unconnected");
	}

	//========================================== Generate and execute Query
==============================================
	public void drawScatterGram(String type, String loc, String axisX, String unitX, Double maxX, String axisY, String unitY,
Double maxY){
		String sql = "SELECT "+axisX+", "+axisY +", Delay\n"
				+"FROM "+type+"_DELAY_WEATHER_"+loc;
		String type_name = new String();
		xlabel.setText(axisX+"("+unitX+")");
		ylabel.setText(axisY+"("+unitY+")");
		label_xmax.setText(maxX.toString());
		label_ymax.setText(maxY.toString());
		if(type.equals("T"))type_name="Train";
		else if(type.equals("F"))type_name="Flight";
		lblPleaseSelectItems.setText(axisY + " and "+axisX+"'s Effect on "+type_name+" Delay"+"("+loc+")");
		labelc1.setText("");
		labelc2.setText("");
		labelc3.setText("");
		labelc4.setText("");
		labelc5.setText("");
		lblNormal.setText("Normal");
		lblDelayed.setText("Delayed");
		try
		{
			Statement statement = conn.createStatement();
			ResultSet rs = statement.executeQuery(sql);
			int w = panel_output.getWidth();
			double x = 0;
			double y = 0;
```

```
                        int delay = 0;
                        int thea = Integer.valueOf(input_delay.getText());
                        Graphics2D dc = (Graphics2D) panel_output.getGraphics();
                        dc.setColor(Color.WHITE);
                        dc.fillRect(0, 0, w, w);
                        dc.setColor(new Color(200,200,200));
                        for(int i=w/10;i<w;i+=w/10)
                        {
                                dc.drawLine(0, i, w, i);
                                dc.drawLine(i, 0, i, w);
                        }
                        while(rs.next())
                        {
                                x = rs.getDouble(axisX);
                                y = rs.getDouble(axisY);
                                if(x==0)x=0.01*maxX/w;
                                if(y==0)y=0.01*maxY/w;
                                delay = rs.getInt("Delay");
                                if(delay>thea)
                                        dc.setColor(Color.RED);
                                else
                                        dc.setColor(Color.BLUE);
                                dc.fillOval((int)(x*w/maxX), (int)(w-y*w/maxY), 4, 4);
                        }
                        logstr = "View Query Executed:\n"+sql+";\n\n" + logstr;
                        text_result.setText(logstr);
                }
                catch(SQLException e1) {
                        e1.printStackTrace();
                }
        }
        public void drawColGram(String condition, String Type, String Loc, int pos, int multiplier){
                String sql = "SELECT Delay\n"
                                +"FROM "+Type+"_DELAY_WEATHER_"+Loc+"\n"
                                + "WHERE locate('"+ condition +"',Conditions)>0";
                try
                {
                        Statement statement = conn.createStatement();
                        ResultSet rs = statement.executeQuery(sql);
                        int w = panel_output.getWidth();
                        double x = 0;
                        double sum = 0;
                        int num = 0;
                        Graphics2D dc = (Graphics2D) panel_output.getGraphics();

                        while(rs.next())
                        {
                                x = rs.getDouble("Delay");
                                sum = sum+x;
                                num ++;
                        }
                        x = sum/num;
                        if(Type.equals("F"))
                                dc.setColor(Color.BLUE);
                        else
                                dc.setColor(Color.RED);

                        dc.fillRect(0, pos, (int)x*multiplier, 10);
                        logstr = "View Query Executed:\n"+sql+";\n\n" + logstr;
                        text_result.setText(logstr);
                }
                catch(SQLException e1) {
                        e1.printStackTrace();
                }
        }
        public void query_create_weaview()
        {
                String sql = "CREATE OR REPLACE VIEW F_DELAY_WEATHER_D AS\n"
                                +"SELECT FNumber, Dep_date, Temperature, Humidity, Visibility, Wind_Speed,
Precipitation, Conditions, -TimeStampDiff(Minute,Act_Arrive,Sch_Arrive) AS Delay\n"
                                + "FROM FLIGHT JOIN AIRPORT ON Depart_Acode=Airport_Code\n"
```

```
                                        + "\t"+"JOIN FLIGHT_OPERATION ON FNumber=FNumber_o\n"
                                        + "\t"+"JOIN WEATHER ON ACity_code=WCity_code AND Dep_date=WDate AND
HOUR(Act_Depart)=HOUR(WTime)\n"
                                        + "GROUP BY FNumber, Dep_Date\n"
                                        + "HAVING COUNT(*)>=1;";

                        logstr = "View Created:\n"+sql+";\n\n" + logstr;
                        text_result.setText(logstr);

                        String sql2 = "CREATE OR REPLACE VIEW T_DELAY_WEATHER_D AS\n"
                                        +"SELECT TNumber, Dep_date, Temperature, Humidity, Visibility, Wind_Speed,
Precipitation, Conditions, -TimeStampDiff(Minute,Act_Arrive,Sch_Arrive) AS Delay\n"
                                        + "FROM TRAIN JOIN STATION ON Depart_Scode=Station_Code\n"
                                        + "\t"+"JOIN TRAIN_OPERATION ON TNumber=TNumber_o\n"
                                        + "\t"+"JOIN WEATHER ON SCity_code=WCity_code AND Dep_date=WDate AND
HOUR(Act_Depart)=HOUR(WTime)\n"
                                        + "GROUP BY TNumber, Dep_Date\n"
                                        + "HAVING COUNT(*)>=1;";

                        logstr = "View Created:\n"+sql2+";\n\n" + logstr;
                        text_result.setText(logstr);

                        String sql3 = "CREATE OR REPLACE VIEW F_DELAY_WEATHER_A AS\n"
                                        +"SELECT FNumber, Dep_date, Temperature, Humidity, Visibility, Wind_Speed,
Precipitation, Conditions, -TimeStampDiff(Minute,Act_Arrive,Sch_Arrive) AS Delay\n"
                                        + "FROM FLIGHT JOIN AIRPORT ON Arrive_Acode=Airport_Code\n"
                                        + "\t"+"JOIN FLIGHT_OPERATION ON FNumber=FNumber_o\n"
                                        + "\t"+"JOIN WEATHER ON ACity_code=WCity_code AND Arr_date=WDate AND
HOUR(Act_Arrive)=HOUR(WTime)\n"
                                        + "GROUP BY FNumber, Dep_Date\n"
                                        + "HAVING COUNT(*)>=1;";

                        logstr = "View Created:\n"+sql3+";\n\n" + logstr;
                        text_result.setText(logstr);

                        String sql4 = "CREATE OR REPLACE VIEW T_DELAY_WEATHER_A AS\n"
                                        +"SELECT TNumber, Dep_date, Temperature, Humidity, Visibility, Wind_Speed,
Precipitation, Conditions, -TimeStampDiff(Minute,Act_Arrive,Sch_Arrive) AS Delay\n"
                                        + "FROM TRAIN JOIN STATION ON Arrive_Scode=Station_Code\n"
                                        + "\t"+"JOIN TRAIN_OPERATION ON TNumber=TNumber_o\n"
                                        + "\t"+"JOIN WEATHER ON SCity_code=WCity_code AND Arr_date=WDate AND
HOUR(Act_Arrive)=HOUR(WTime)\n"
                                        + "GROUP BY TNumber, Dep_Date\n"
                                        + "HAVING COUNT(*)>=1;";

                        logstr = "View Created:\n"+sql4+";\n\n" + logstr;
                        text_result.setText(logstr);
                        try
                        {
                                Statement statement = conn.createStatement();
                                statement.executeUpdate(sql);
                                statement.executeUpdate(sql2);
                                statement.executeUpdate(sql3);
                                statement.executeUpdate(sql4);
                        }
                        catch(SQLException e1) {
                                e1.printStackTrace();
                        }
                }
}
```

6.2 Excel Scripts

WEATHER

```
=CONCATENATE("INSERT INTO WEATHER VALUES ('",
TEXT(INDIRECT("WEATHER!"&ADDRESS(ROW(),14)),"yyyy-mm-dd hh:mm"),
"','",
INDIRECT("WEATHER!"&ADDRESS(ROW(),15)),
```

"",",
INDIRECT("WEATHER!"&ADDRESS(ROW(),2)),
",",
INDIRECT("WEATHER!"&ADDRESS(ROW(),4)),
",",
INDIRECT("WEATHER!"&ADDRESS(ROW(),6)),
",",
INDIRECT("WEATHER!"&ADDRESS(ROW(),8)),
",'",
INDIRECT("WEATHER!"&ADDRESS(ROW(),7)),
"',",
INDIRECT("WEATHER!"&ADDRESS(ROW(),10)),
",'",
INDIRECT("WEATHER!"&ADDRESS(ROW(),12)),
"');"
)

FLIGHT

=CONCATENATE("INSERT INTO FLIGHT VALUES ('",
CONCATENATE(INDIRECT("FLIGHT!"&ADDRESS(ROW(),1)),INDIRECT("FLIGHT!"&ADDRESS(ROW(),3))),
"','",
IF(HOUR(INDIRECT("FLIGHT!"&ADDRESS(ROW(),6)))*60+MINUTE(INDIRECT("FLIGHT!"&ADDRESS(ROW(),6)))-INDIR
ECT("FLIGHT!"&ADDRESS(ROW(),8))>0,CONCATENATE(ROUNDDOWN((HOUR(INDIRECT("FLIGHT!"&ADDRESS(ROW
(),6)))*60+MINUTE(INDIRECT("FLIGHT!"&ADDRESS(ROW(),6)))-INDIRECT("FLIGHT!"&ADDRESS(ROW(),8)))/60,0),":",M
OD(HOUR(INDIRECT("FLIGHT!"&ADDRESS(ROW(),6)))*60+MINUTE(INDIRECT("FLIGHT!"&ADDRESS(ROW(),6)))-INDI
RECT("FLIGHT!"&ADDRESS(ROW(),8)),60)),CONCATENATE(ROUNDDOWN((HOUR(INDIRECT("FLIGHT!"&ADDRESS(R
OW(),6)))*60+MINUTE(INDIRECT("FLIGHT!"&ADDRESS(ROW(),6)))-INDIRECT("FLIGHT!"&ADDRESS(ROW(),8)))/60,0)+2
4,":",MOD(HOUR(INDIRECT("FLIGHT!"&ADDRESS(ROW(),6)))*60+MINUTE(INDIRECT("FLIGHT!"&ADDRESS(ROW(),6)))
-INDIRECT("FLIGHT!"&ADDRESS(ROW(),8)),60))),
"','",
TEXT(INDIRECT("FLIGHT!"&ADDRESS(ROW(),6)),"h:mm"),
"','",
INDIRECT("FLIGHT!"&ADDRESS(ROW(),5)),
"','",
INDIRECT("FLIGHT!"&ADDRESS(ROW(),18)),
"',",
IF(HOUR(INDIRECT("FLIGHT!"&ADDRESS(ROW(),6)))*60+MINUTE(INDIRECT("FLIGHT!"&ADDRESS(ROW(),6)))-INDIR
ECT("FLIGHT!"&ADDRESS(ROW(),8))>0,0,1),
");"
)

FLIGHT_OPERATION

=CONCATENATE("INSERT INTO FLIGHT_OPERATION VALUES ('",
CONCATENATE(INDIRECT("FLIGHT!"&ADDRESS(ROW(),1)),INDIRECT("FLIGHT!"&ADDRESS(ROW(),3))),
"','",
IF(HOUR(INDIRECT("FLIGHT!"&ADDRESS(ROW(),7)))*60+MINUTE(INDIRECT("FLIGHT!"&ADDRESS(ROW(),7)))-INDIR
ECT("FLIGHT!"&ADDRESS(ROW(),9))>0,TEXT(INDIRECT("FLIGHT!"&ADDRESS(ROW(),2)),"yyyy-mm-dd"),TEXT(INDIRE
CT("FLIGHT!"&ADDRESS(ROW(),2))-1,"yyyy-mm-dd")),
"','",
TEXT(INDIRECT("FLIGHT!"&ADDRESS(ROW(),2)),"yyyy-mm-dd"),
"','",
IF(HOUR(INDIRECT("FLIGHT!"&ADDRESS(ROW(),7)))*60+MINUTE(INDIRECT("FLIGHT!"&ADDRESS(ROW(),7)))-INDIR
ECT("FLIGHT!"&ADDRESS(ROW(),9))>0,
CONCATENATE(ROUNDDOWN((HOUR(INDIRECT("FLIGHT!"&ADDRESS(ROW(),7)))*60+MINUTE(INDIRECT("FLIGHT!"
&ADDRESS(ROW(),7)))-INDIRECT("FLIGHT!"&ADDRESS(ROW(),9)))/60,0),
IF(MOD(HOUR(INDIRECT("FLIGHT!"&ADDRESS(ROW(),7)))*60+MINUTE(INDIRECT("FLIGHT!"&ADDRESS(ROW(),7)))-I
NDIRECT("FLIGHT!"&ADDRESS(ROW(),9)),60)<10,":0",":"),
MOD(HOUR(INDIRECT("FLIGHT!"&ADDRESS(ROW(),7)))*60+MINUTE(INDIRECT("FLIGHT!"&ADDRESS(ROW(),7)))-IN
DIRECT("FLIGHT!"&ADDRESS(ROW(),9)),60)),CONCATENATE(ROUNDDOWN(((HOUR(INDIRECT("FLIGHT!"&ADDRES
S(ROW(),7)))+24)*60+MINUTE(INDIRECT("FLIGHT!"&ADDRESS(ROW(),7)))-INDIRECT("FLIGHT!"&ADDRESS(ROW(),9)))
/60,0),
IF(MOD(HOUR(INDIRECT("FLIGHT!"&ADDRESS(ROW(),7)))*60+MINUTE(INDIRECT("FLIGHT!"&ADDRESS(ROW(),7)))-I
NDIRECT("FLIGHT!"&ADDRESS(ROW(),9)),60)<10,":0",":"),
MOD(HOUR(INDIRECT("FLIGHT!"&ADDRESS(ROW(),7)))*60+MINUTE(INDIRECT("FLIGHT!"&ADDRESS(ROW(),7)))-IN
DIRECT("FLIGHT!"&ADDRESS(ROW(),9)),60))),
"','",

```
TEXT(INDIRECT("FLIGHT!"&ADDRESS(ROW(),7)),"h:mm"),
"",",
IF(MAX(INDIRECT("FLIGHT!"&ADDRESS(ROW(),13)):INDIRECT("FLIGHT!"&ADDRESS(ROW(),17)))>0,CONCATENATE(
"'",INDIRECT("FLIGHT!"&ADDRESS(1,MATCH(MAX(INDIRECT("FLIGHT!"&ADDRESS(ROW(),13)):INDIRECT("FLIGHT!"
&ADDRESS(ROW(),17))),INDIRECT("FLIGHT!"&ADDRESS(ROW(),13)):INDIRECT("FLIGHT!"&ADDRESS(ROW(),17)),0)+1
1)),"'"),"NULL"),
");"
)
```

TRAIN_OPERATION

```
=CONCATENATE("INSERT INTO TRAIN_OPERATION VALUES ('",
INDIRECT("TRAIN!"&ADDRESS(ROW(),1)),
"','",
TEXT(INDIRECT("TRAIN!"&ADDRESS(ROW(),2)),"yyyy-mm-dd"),
"','",
TEXT(INDIRECT("TRAIN!"&ADDRESS(ROW(),8)),"yyyy-mm-dd"),
"','",
TEXT(INDIRECT("TRAIN!"&ADDRESS(ROW(),3)),"h:mm"),
"','",
TEXT(INDIRECT("TRAIN!"&ADDRESS(ROW(),9)),"h:mm"),
"');"
)
```