# Using SQL to Load a Star Schema

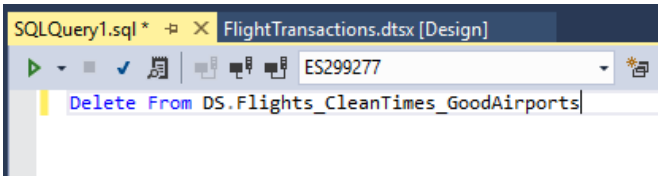
You will use ***SQL action queries on SQL Server Management Studio (SSMS) instead of SQL Server Integration Services (SSIS)*** to load your flights star schema. In practice, organizations either use ***stored procedures containing action queries or integration services like SSIS*** when performing data transformations and loads. The stored procedures would be saved in the database. This assignment will give you practice with SQL action queries that could be embedded in stored procedures.

## Useful SQL and SQL Server Settings

You will likely have reason to use one or more of these SQL expressions.

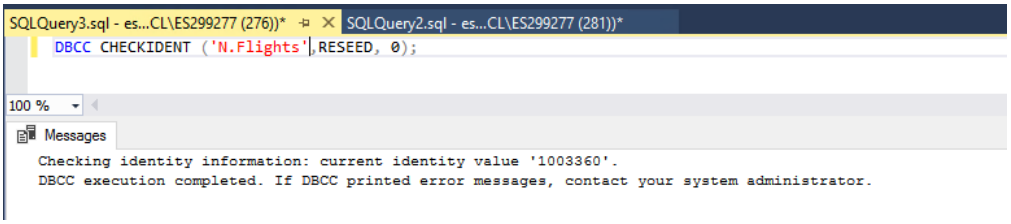
### Delete All Data in A Table

If you load a table and need to reload it, delete all data from it first.




### Reset the Identity Column Sequence

If you load a table that has an identity PK column like W.Flights and need to reload it, you can reset the identity sequence to restart at zero so that the first PK value is 1 (after you delete all the table data). In the example below, the "current identity value '1003360'" is the last identity value in the example before it was reset to zero.

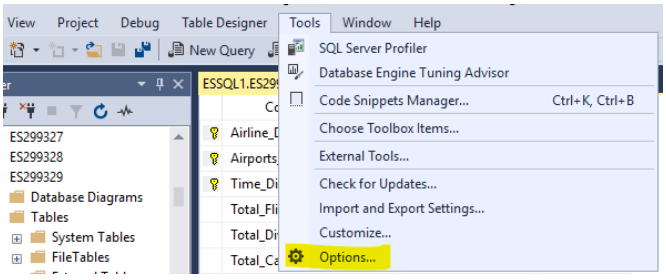



### Delete a Table

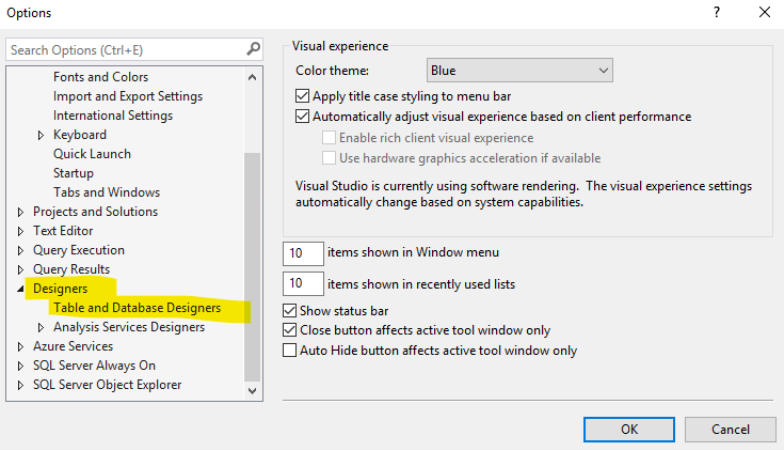
You do not need to use SQL. You can just right-click a table to delete it. Be careful, it is easy to delete a table accidentally.

### Change a Table

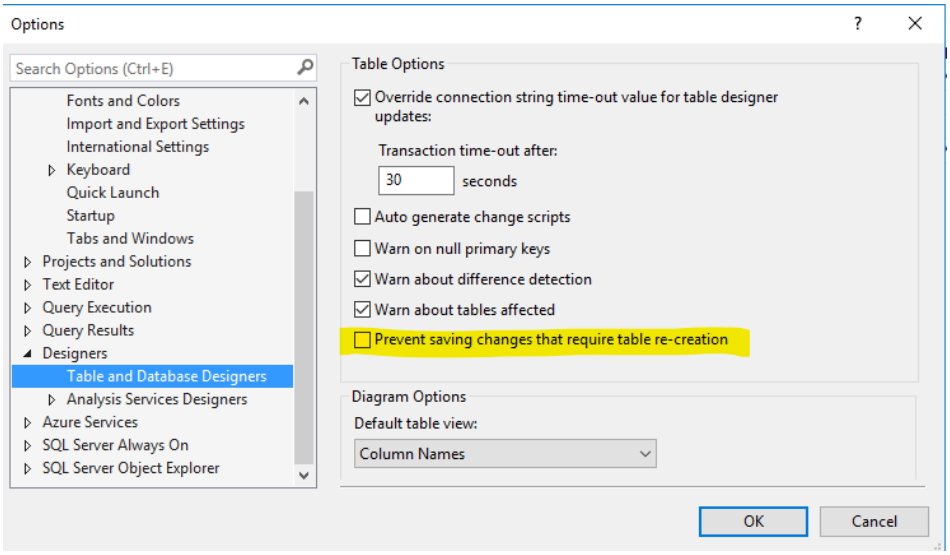
If you need to change a table, it is often best to ***remove all data and relationships and save first***. Also, you can change your Designer setting to allow you to make table changes that require that the table be dropped and recreated. To do this, use SQL Server Tools -> Options.



Then choose Designers -> Tables and Database Designers




Uncheck the box that prevents saving changes that require table re-creation and click OK.




# Create SS Schema, Tables and Relationships

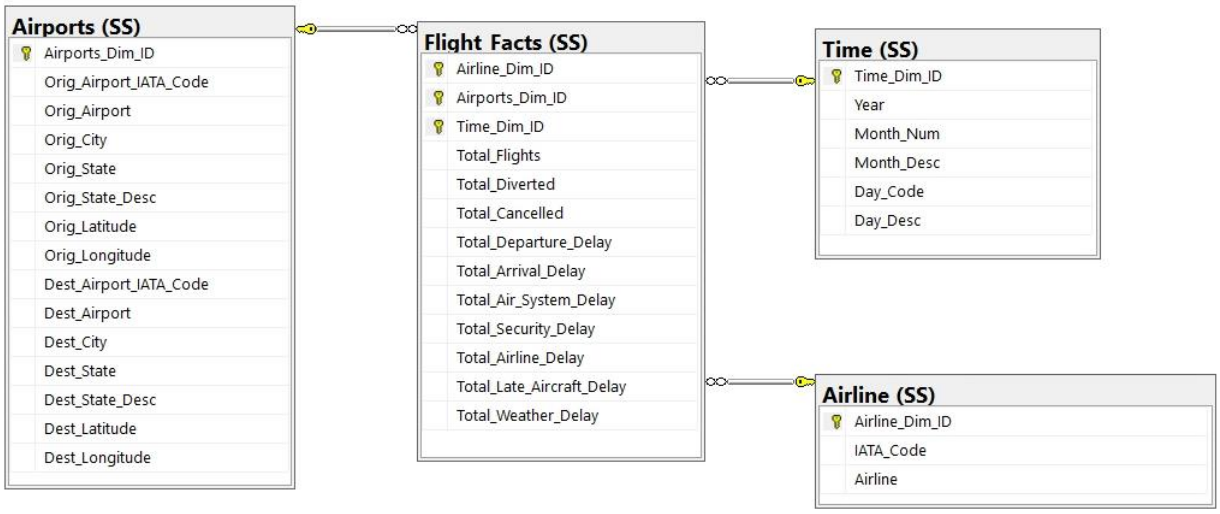
Create a new schema called “SS”. Create tables and relationships in the SS schema that are identical to those shown in the attached flights star schema ERD.

***Notes:***

- Be sure to ***view table properties*** when you are creating tables so you can select the ***correct schema (SS)***.
- The ***dimension table PK columns*** should all ***be identity*** columns***.***
- The ***fact table primary key*** is a ***composite PK*** combining the dimension table FKs and no identity columns.
- To create a composite PK, select all columns that make up the PK at once, then right-click and select Set Primary Key.
- All table columns are required.

Q1: There are two relationships between W.Airports and W.Flights. Why is there only one relationship between the SS.Airports dimension table and the SS.Flight_Facts table?

Create a database diagram in SQL Server for the SS schema. It should look similar to the following.




Q2: Snip/paste your SS diagram (with the database name showing) into the document.

# Load Dimensional Table Data

You will select data from the operational data schema (W) and load it into the dimension and fact tables. You must insert data into the dimensional tables before the fact table due to the foreign key relationships

**NOTES:**

1. You will use a SQL action statement with a syntax pattern of ***INSERT INTO … SELECT*** where you construct the SELECT statement used to extract data from one set of tables and insert it into a dimension or fact table.
2. You will ***test the SELECT statement before using it to insert data by commenting out the INSERT INTO*** portion.
3. If you get an error on the INSERT INTO, you may need to add a return after the line.
4. The INSERT INTO … SELECT syntax assumes that the ***SELECT columns are in the same order as the target table columns***. You must check this assumption before each load or you will have the wrong data in some columns and/or errors.

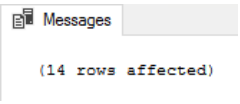
## SS.Airline Dimension Table Load

The syntax to load the SS.Airline dimension table is shown below with the INSERT INTO statement commented out. If you run it as is, you will select the 14 airlines.

```
--INSERT INTO SS.Airline
SELECT * from W.Airlines
```

### Q3: Explain what the SELECT statement does. What will be inserted into SS.Airline?

Remove the dashes that comment out the INSERT INTO statement and run it. You will see the following message.




Then select all rows from SS.Airline. Your results should look like the following, including the identity column PK values.

Results | Messages

| | Airline_Dim_ID | Airline_IATA_Code | Airline |
|---|---|---|---|
| 1 | 1 | AA | American Airlines Inc. |
| 2 | 2 | AS | Alaska Airlines Inc. |
| 3 | 3 | B6 | JetBlue Airways |
| 4 | 4 | DL | Delta Air Lines Inc. |
| 5 | 5 | EV | Atlantic Southeast Airlines |
| 6 | 6 | F9 | Frontier Airlines Inc. |
| 7 | 7 | HA | Hawaiian Airlines Inc. |
| 8 | 8 | MQ | American Eagle Airlines Inc. |
| 9 | 9 | NK | Spirit Air Lines |
| 10 | 10 | OO | Skywest Airlines Inc. |
| 11 | 11 | UA | United Air Lines Inc. |
| 12 | 12 | US | US Airways Inc. |
| 13 | 13 | VX | Virgin America |
| 14 | 14 | WN | Southwest Airlines Co. |

## SS.Airports Dimension Table Load

The syntax to load the SS.Airports dimension table requires joining normalized operational tables that describe airports (W.Airports, W.States) with the W.Flights table. The airport dimension table contains ***all routes (pairings of origin and destination airports) flown in 2015***.

The maximum number of possible routes would be all possible permutations of airports rather than combinations of airports since order (origin, destination) matters. You could use a permutations formula where n=322 airports and k=2 (two airports for each route) to determine the maximum number of routes as n!/(n-k)! = 322!/(322-2)! = 322!/320! = 322*321 = ***103,362 possible routes***.

Fortunately, you will not need to load all possible routes. You can run a query against the W.Flights table to see how many distinct routes it contains (***4696 routes***). This query counts the distinct concatenations of origin and destination airport IATA codes in W.Flights. When you are loading data, it is important to know how many values to expect.

```
SELECT Count(DISTINCT Origin_Airport + Destination_Airport) as 'Total Routes'
FROM W.Flights
```

Results | Messages

| | Total Routes |
|---|---|
| 1 | 4696 |

The syntax to load the operational data into the Airports dimension table is shown below with the INSERT INTO statement commented out. Table aliases are used to streamline the SQL syntax. ***Make sure you understand what you are doing with the syntax.***

```
--INSERT INTO SS.Airports
SELECT DISTINCT F.Origin_Airport, OA.Airport, OA.City, OA.State, OS.State_Desc,
       OA.Latitude, OA.Longitude,
       F.Destination_Airport, DA.Airport, DA.City, DA.State, DS.State_Desc,
       DA.Latitude, DA.Longitude
FROM W.Flights AS F
-- W.Airports must be joined twice since there are S.Airports_Dim table contains
origin/destination airport pairs
JOIN W.Airports AS OA ON F.Origin_Airport = OA.Airport_IATA_Code
JOIN W.Airports AS DA ON F.Destination_Airport = DA.Airport_IATA_Code
-- W.States is joined twice to differentiate between the origin and destination states
JOIN W.States OS ON OA.State = OS.State_Code
JOIN W.States DS ON DA.State = DS.State_Code
```

Q4: Explain what the SELECT statement does. What will be inserted into SS.Airports?

***Before you load the SS.Airports*** table, check the SELECT statement with the INSERT INTO statement commented out to make sure it is working properly. Your results should look similar to the following.

Results | Messages

| | Origin_Airport | Airport | City | State | State_Desc | Latitude | Longitude | Destination_Airport | Airport | City | State | State_Desc | Latitude | Longitude |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ABE | Lehigh Valley International Airport | Allentown | PA | Pennsylvania | 40.65236 | -75.44040 | ATL | Hartsfield-Jackson Atlanta International Airport | Atlanta | GA | Georgia | 33.64044 | -84.42694 |
| 2 | ABQ | Albuquerque International Sunport | Albuquerque | NM | New Mexico | 35.04022 | -106.60919 | ATL | Hartsfield-Jackson Atlanta International Airport | Atlanta | GA | Georgia | 33.64044 | -84.42694 |
| 3 | ABQ | Albuquerque International Sunport | Albuquerque | NM | New Mexico | 35.04022 | -106.60919 | ORD | Chicago O'Hare International Airport | Chicago | IL | Illinois | 41.97960 | -87.90446 |
| 4 | ACY | Atlantic City International Airport | Atlantic City | NJ | New Jersey | 39.45758 | -74.57717 | RSW | Southwest Florida International Airport | Ft. Myers | FL | Florida | 26.53617 | -81.75517 |
| 5 | AGS | Augusta Regional AirportÂ (Bush Field) | Augusta | GA | Georgia | 33.36996 | -81.96450 | ATL | Hartsfield-Jackson Atlanta International Airport | Atlanta | GA | Georgia | 33.64044 | -84.42694 |
| 6 | ANC | Ted Stevens Anchorage International Airport | Anchorage | AK | Alaska | 61.17432 | -149.99619 | BET | Bethel Airport | Bethel | AK | Alaska | 60.77978 | -161.83800 |
| 7 | ATL | Hartsfield-Jackson Atlanta International Airport | Atlanta | GA | Georgia | 33.64044 | -84.42694 | AGS | Augusta Regional AirportÂ (Bush Field) | Augusta | GA | Georgia | 33.36996 | -81.96450 |
| 8 | ATL | Hartsfield-Jackson Atlanta International Airport | Atlanta | GA | Georgia | 33.64044 | -84.42694 | ANC | Ted Stevens Anchorage International Airport | Anchorage | AK | Alaska | 61.17432 | -149.99619 |
| 9 | ATL | Hartsfield-Jackson Atlanta International Airport | Atlanta | GA | Georgia | 33.64044 | -84.42694 | DSM | Des Moines International Airport | Des Moines | IA | Iowa | 41.53493 | -93.66068 |
| 10 | ATL | Hartsfield-Jackson Atlanta International Airport | Atlanta | GA | Georgia | 33.64044 | -84.42694 | ELM | Elmira/Corning Regional Airport | Elmira | NY | New York | 42.15991 | -76.89144 |
| 11 | ATL | Hartsfield-Jackson Atlanta International Airport | Atlanta | GA | Georgia | 33.64044 | -84.42694 | ICT | Wichita Dwight D. Eisenhower National AirportÂ (... | Wichita | KS | Kansas | 37.64996 | -97.43305 |
| 12 | ATL | Hartsfield-Jackson Atlanta International Airport | Atlanta | GA | Georgia | 33.64044 | -84.42694 | ILM | Wilmington International Airport | Wilmington | NC | North Carolina | 34.27061 | -77.90256 |
| 13 | ATL | Hartsfield-Jackson Atlanta International Airport | Atlanta | GA | Georgia | 33.64044 | -84.42694 | SHV | Shreveport Regional Airport | Shreveport | LA | Louisiana | 32.44663 | -93.82560 |
| 14 | AUS | Austin-Bergstrom International Airport | Austin | TX | Texas | 30.19453 | -97.66987 | BOS | Gen. Edward Lawrence Logan International Airport | Boston | MA | Massachusetts | 42.36435 | -71.00518 |
| 15 | AUS | Austin-Bergstrom International Airport | Austin | TX | Texas | 30.19453 | -97.66987 | BWI | Baltimore-Washington International Airport | Baltimore | MD | Maryland | 39.17540 | -76.66820 |
| 16 | AUS | Austin-Bergstrom International Airport | Austin | TX | Texas | 30.19453 | -97.66987 | PDX | Portland International Airport | Portland | OR | Oregon | 45.58872 | -122.59750 |

Query executed successfully. | essql1.walton.uark.edu (13.... | GACL\ES299277 (58) | ES299277 | 00:00:12 | 4696 rows

Once the SELECT statement works correctly, remove the dashes that comment out the INSERT INTO statement and load the SS.Airports table. Your results will show 4696 rows affected since you added them to the table.

Messages

```
(4696 rows affected)
```

Select all from the SS.Airports table and your results should look similar to the following. ***Make sure your identity PK column is working properly.***

Results | Messages

| | Airports_Dim_ID | Orig_Airport_IATA_Code | Orig_Airport | Orig_City | Orig_State | Orig_State_Desc | Orig_Latitude | Orig_Longitude | Dest_Airport_IATA_Code | Dest_Airport | Dest_City | Dest_State | Dest_State_Desc | Dest_Latitude | Dest_Longitude |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | ABQ | Albuquerque International Sunport | Albuquerque | NM | New Mexico | 35.04022 | -106.60919 | SAN | San Diego International AirportÂ (Lindbergh Field) | San Diego | CA | California | 32.73356 | -117.18966 |
| 2 | 2 | AEX | Alexandria International Airport | Alexandria | LA | Louisiana | 31.32737 | -92.54856 | IAH | George Bush Intercontinental Airport | Houston | TX | Texas | 29.98047 | -95.33972 |
| 3 | 3 | ALB | Albany International Airport | Albany | NY | New York | 42.74812 | -73.80298 | IAD | Washington Dulles International Airport | Chantilly | VA | Virginia | 38.94453 | -77.45581 |
| 4 | 4 | ALB | Albany International Airport | Albany | NY | New York | 42.74812 | -73.80298 | MSP | Minneapolis-Saint Paul International Airport | Minneapolis | MN | Minnesota | 44.88055 | -93.21692 |
| 5 | 5 | ANC | Ted Stevens Anchorage International Airport | Anchorage | AK | Alaska | 61.17432 | -149.99619 | LAS | McCarran International Airport | Las Vegas | NV | Nevada | 36.08036 | -115.15233 |
| 6 | 6 | ATL | Hartsfield-Jackson Atlanta International Airport | Atlanta | GA | Georgia | 33.64044 | -84.42694 | ABQ | Albuquerque International Sunport | Albuquerque | NM | New Mexico | 35.04022 | -106.60919 |
| 7 | 7 | ATL | Hartsfield-Jackson Atlanta International Airport | Atlanta | GA | Georgia | 33.64044 | -84.42694 | BOS | Gen. Edward Lawrence Logan International Airport | Boston | MA | Massachusetts | 42.36435 | -71.00518 |
| 8 | 8 | ATL | Hartsfield-Jackson Atlanta International Airport | Atlanta | GA | Georgia | 33.64044 | -84.42694 | DTW | Detroit Metropolitan Airport | Detroit | MI | Michigan | 42.21206 | -83.34884 |
| 9 | 9 | ATL | Hartsfield-Jackson Atlanta International Airport | Atlanta | GA | Georgia | 33.64044 | -84.42694 | EGE | Eagle County Regional Airport | Eagle | CO | Colorado | 39.64257 | -106.91770 |
| 10 | 10 | ATL | Hartsfield-Jackson Atlanta International Airport | Atlanta | GA | Georgia | 33.64044 | -84.42694 | EYW | Key West International Airport | Key West | FL | Florida | 24.55611 | -81.75956 |
| 11 | 11 | ATL | Hartsfield-Jackson Atlanta International Airport | Atlanta | GA | Georgia | 33.64044 | -84.42694 | FWA | Fort Wayne International Airport | Fort Wayne | IN | Indiana | 40.97847 | -85.19515 |
| 12 | 12 | ATL | Hartsfield-Jackson Atlanta International Airport | Atlanta | GA | Georgia | 33.64044 | -84.42694 | SGF | Springfield-Branson National Airport | Springfield | MO | Missouri | 37.24433 | -93.38686 |
| 13 | 13 | AUS | Austin-Bergstrom International Airport | Austin | TX | Texas | 30.19453 | -97.66987 | ELP | El Paso International Airport | El Paso | TX | Texas | 31.80667 | -106.37781 |
| 14 | 14 | AUS | Austin-Bergstrom International Airport | Austin | TX | Texas | 30.19453 | -97.66987 | HRL | Valley International Airport | Harlingen | TX | Texas | 26.22851 | -97.65439 |
| 15 | 15 | AUS | Austin-Bergstrom International Airport | Austin | TX | Texas | 30.19453 | -97.66987 | SEA | Seattle-Tacoma International Airport | Seattle | WA | Washington | 47.44898 | -122.30931 |
| 16 | 16 | BIL | Billings Logan International Airport | Billings | MT | Montana | 45.80766 | -108.54286 | MSP | Minneapolis-Saint Paul International Airport | Minneapolis | MN | Minnesota | 44.88055 | -93.21692 |
| 17 | 17 | BLI | Bellingham International Airport | Bellingham | WA | Washington | 48.79275 | -122.53753 | HNL | Honolulu International Airport | Honolulu | HI | Hawaii | 21.31869 | -157.92241 |
| 18 | 18 | BMI | Central Illinois Regional Airport at Bloomington-N... | Bloomington | IL | Illinois | 40.47799 | -88.91595 | DTW | Detroit Metropolitan Airport | Detroit | MI | Michigan | 42.21206 | -83.34884 |
| 19 | 19 | BOS | Gen. Edward Lawrence Logan International Airport | Boston | MA | Massachusetts | 42.36435 | -71.00518 | BUF | Buffalo Niagara International Airport | Buffalo | NY | New York | 42.94052 | -78.73217 |

Query executed successfully. | essql1.walton.uark.edu (13.... | GACL\ES299277 (90) | ES299277 | 00:00:00 | 4696 rows

## SS.Time Dimension Table Load

You will load the Time dimension table from the W.Flights table joined with the various lookup tables. We would expect to see 96 distinct rows (1 year x 12 months x 8 days, allowing for the "unknown" designation). The syntax for the load is shown below (with the INSERT INTO commented out).

```
--INSERT INTO SS.Time
SELECT DISTINCT F.Year, F.Month, M.Month_Desc, F.Day, W.Day_Desc
FROM W.Flights AS F
JOIN W.Months As M on M.Month_Num=F.Month
JOIN W.Weekdays As W on W.Day_Code = F.Day
```

### Q5: Explain what the SELECT statement does. What will be inserted into SS.Time?

Run the SELECT query to test it before loading. Your results should similar to the following.

Results | Messages

| | Year | Month | Month_Desc | Day | Day_Desc |
|---|---|---|---|---|---|
| 1 | 2015 | 3 | March | 7 | Sunday |
| 2 | 2015 | 5 | May | 3 | Wednesday |
| 3 | 2015 | 5 | May | 6 | Saturday |
| 4 | 2015 | 6 | June | 3 | Wednesday |
| 5 | 2015 | 6 | June | 6 | Saturday |
| 6 | 2015 | 6 | June | 9 | Unknown |
| 7 | 2015 | 9 | September | 9 | Unknown |
| 8 | 2015 | 11 | November | 2 | Tuesday |
| 9 | 2015 | 11 | November | 5 | Friday |
| 10 | 2015 | 12 | December | 2 | Tuesday |
| 11 | 2015 | 12 | December | 5 | Friday |
| 12 | 2015 | 2 | February | 1 | Monday |
| 13 | 2015 | 4 | April | 1 | Monday |
| 14 | 2015 | 4 | April | 4 | Thursday |
| 15 | 2015 | 4 | April | 7 | Sunday |
| 16 | 2015 | 5 | May | 2 | Tuesday |

ark.edu (16.... | GACL\datn0128 (55) | DATN0128 | 00:00:00 | 96 rows

Remove the comment syntax and run the INSERT INTO to create the 96 rows.

Messages

```
(96 rows affected)
```

Select all rows from SS.Time and get the following results, including the identity PK column.

Results | Messages

| | Time_Dim_ID | Year | Month_Num | Month_Desc | Day_Code | Day_Desc |
|---|---|---|---|---|---|---|
| 1 | 1 | 2015 | 1 | January | 2 | Tuesday |
| 2 | 2 | 2015 | 3 | March | 2 | Tuesday |
| 3 | 3 | 2015 | 3 | March | 5 | Friday |
| 4 | 4 | 2015 | 5 | May | 5 | Friday |
| 5 | 5 | 2015 | 6 | June | 1 | Monday |
| 6 | 6 | 2015 | 8 | August | 1 | Monday |
| 7 | 7 | 2015 | 8 | August | 4 | Thursday |
| 8 | 8 | 2015 | 8 | August | 7 | Sunday |
| 9 | 9 | 2015 | 10 | October | 4 | Thursday |
| 10 | 10 | 2015 | 10 | October | 7 | Sunday |
| 11 | 11 | 2015 | 12 | December | 7 | Sunday |
| 12 | 12 | 2015 | 2 | February | 1 | Monday |
| 13 | 13 | 2015 | 4 | April | 1 | Monday |
| 14 | 14 | 2015 | 4 | April | 4 | Thursday |
| 15 | 15 | 2015 | 4 | April | 7 | Sunday |
| 16 | 16 | 2015 | 5 | May | 2 | Tuesday |

Query e... | essql1.walton.uark.edu (16.... | GACL\datn0128 (99) | DATN0128 | 00:00:00 | 96 rows

# Load Fact Table Data

You will join W.Flights with the dimension tables to load the Flight_Facts table. The syntax to load the operational data into the Flight_Fact table is shown below with the INSERT INTO commented out.

```
--INSERT INTO SS.Flight_FACTS
SELECT AL.Airline_Dim_ID, AP.Airports_Dim_ID, T.Time_Dim_ID,
       COUNT(*) AS Total_Flights,
       ISNULL(SUM(F.Diverted),0) As Total_Diverted,
       ISNULL(SUM(F.Cancelled),0) as Total_Cancelled,
       ISNULL(SUM(F.Departure_Delay),0) AS Total_Departure_Delay,
       ISNULL(SUM(F.Arrival_Delay),0) AS Total_Arrival_Delay,
       ISNULL(SUM(F.Air_System_Delay),0) AS Total_Air_System_Delay,
       ISNULL(SUM(F.Security_Delay),0) AS Total_Security_Delay,
       ISNULL(SUM(F.Airline_Delay),0) AS Total_Airline_Delay,
       ISNULL(SUM(F.Late_Aircraft_Delay),0) AS Total_Late_Aircraft_Delay,
       ISNULL(SUM(F.Weather_Delay),0) AS Total_Weather_Delay
FROM W.Flights F
JOIN SS.Airports AP
       ON F.Origin_Airport = AP.Orig_Airport_IATA_Code
       AND F.Destination_Airport = AP.Dest_Airport_IATA_Code
JOIN SS.Airline AL
       ON F.Airline = AL.Airline_IATA_Code
JOIN SS.Time T
       ON F.Year = T.Year
       AND F.Month = T.Month_Num
       AND F.Day_of_Week = T.Day_Code
GROUP BY AL.Airline_Dim_ID, AP.Airports_Dim_ID, T.Time_Dim_ID
```

### Q6: Explain what the SELECT statement does. What will be inserted into SS.Flight_Facts?

**Before you load the Flight_Facts table**, check the SELECT statement to make sure it is working properly. Your results should look similar to the following with 492,169 rows that will be added to the fact table. Notice that the fact table has ***significantly less rows*** that the operational table (W.Flights).

Results | Messages

| | Airline_Dim_ID | Airports_Dim_ID | Time_Dim_ID | Total_Flights | Total_Diverted | Total_Cancelled | Total_Departure_Delay | Total_Arrival_Delay | Total_Air_System_Delay | Total_Security_Delay | Total_Airline_Delay | Total_Late_Aircraft_Delay | Total_Weather_Delay |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 4 | 1513 | 36 | 18 | 0 | 0 | 406 | 264 | 76 | 0 | 267 | 80 | 0 |
| 2 | 10 | 2233 | 73 | 3 | 0 | 0 | -23 | -31 | 0 | 0 | 0 | 0 | 0 |
| 3 | 14 | 4594 | 64 | 4 | 0 | 0 | 72 | 103 | 41 | 0 | 35 | 27 | 0 |
| 4 | 12 | 727 | 85 | 11 | 0 | 0 | 28 | 83 | 133 | 0 | 43 | 0 | 0 |
| 5 | 12 | 4393 | 5 | 15 | 2 | 0 | 86 | 156 | 96 | 0 | 13 | 52 | 0 |
| 6 | 11 | 1123 | 52 | 19 | 2 | 1 | 646 | 171 | 92 | 0 | 80 | 88 | 0 |
| 7 | 4 | 4281 | 70 | 4 | 0 | 0 | 10 | 32 | 22 | 0 | 0 | 21 | 0 |
| 8 | 11 | 2791 | 45 | 8 | 0 | 0 | -8 | -42 | 0 | 0 | 0 | 0 | 0 |
| 9 | 12 | 4614 | 25 | 8 | 0 | 1 | -31 | -77 | 0 | 0 | 0 | 0 | 0 |
| 10 | 4 | 2613 | 77 | 15 | 0 | 0 | 156 | 64 | 17 | 0 | 110 | 0 | 0 |
| 11 | 3 | 3888 | 65 | 8 | 0 | 0 | -39 | -123 | 0 | 0 | 0 | 0 | 0 |
| 12 | 11 | 3708 | 57 | 37 | 0 | 0 | 860 | 479 | 568 | 0 | 19 | 348 | 0 |
| 13 | 1 | 737 | 6 | 15 | 0 | 0 | -3 | 28 | 110 | 0 | 9 | 0 | 16 |
| 14 | 14 | 1899 | 82 | 6 | 0 | 0 | 113 | -8 | 0 | 0 | 28 | 60 | 0 |
| 15 | 3 | 1460 | 83 | 16 | 0 | 0 | 487 | 537 | 328 | 8 | 65 | 182 | 0 |
| 16 | 10 | 501 | 83 | 4 | 0 | 0 | 78 | 4 | 0 | 0 | 0 | 41 | 0 |

Query executed successfully. | essql1.walton.uark.edu (16.... | GACL\datn0128 (55) | DATN0128 | 00:00:05 | 492,169 rows

**After you load the Flight_Fact table,** you will see the message below.

Messages

```
Warning: Null value is eliminated by an aggregate or other SET operation.

(492169 rows affected)
```

Select all rows and check the data to see if the load appears to be correct.

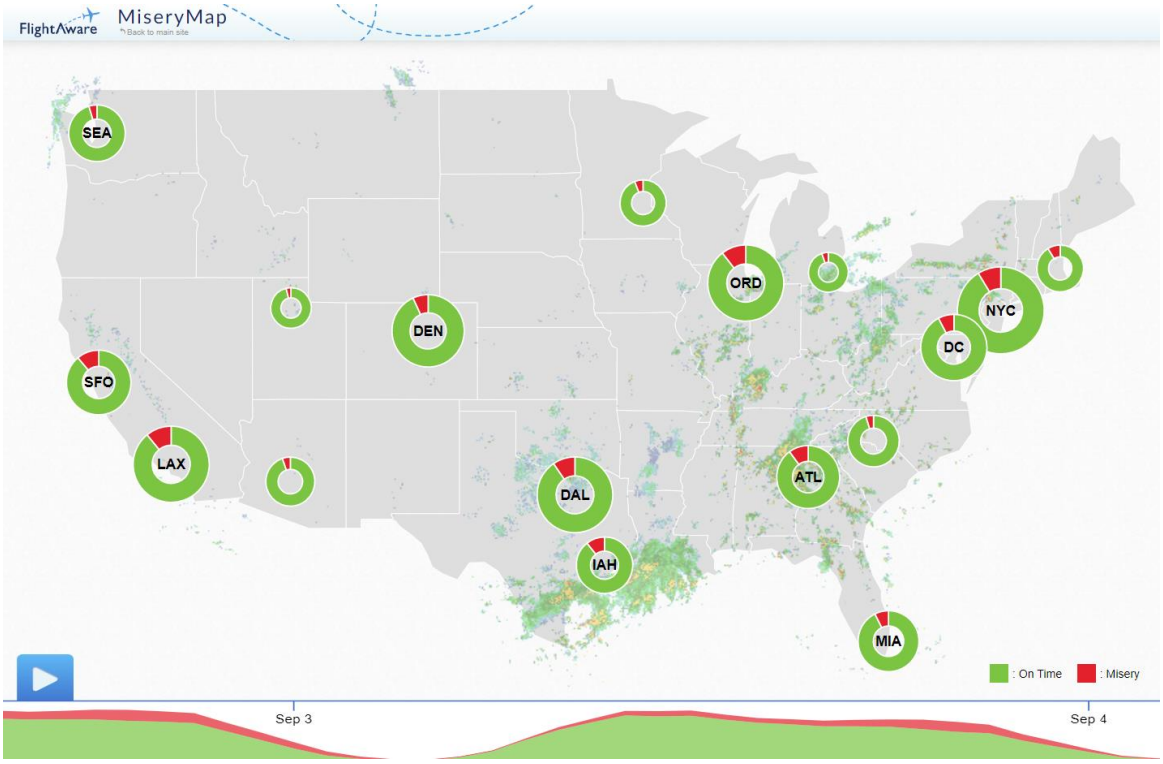
Results | Messages

| | Airline_Dim_ID | Airports_Dim_ID | Time_Dim_ID | Total_Flights | Total_Diverted | Total_Cancelled | Total_Departure_Delay | Total_Arrival_Delay | Total_Air_System_Delay | Total_Security_Delay | Total_Airline_Delay | Total_Late_Aircraft_Delay | Total_Weather_Delay |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 29 | 1 | 9 | 0 | 0 | 194 | 181 | 53 | 0 | 94 | 89 | 0 |
| 2 | 1 | 29 | 4 | 13 | 0 | 0 | 702 | 469 | 267 | 0 | 38 | 240 | 0 |
| 3 | 1 | 29 | 6 | 11 | 0 | 1 | 104 | 30 | 0 | 0 | 0 | 73 | 26 |
| 4 | 1 | 29 | 7 | 10 | 0 | 0 | 40 | 30 | 18 | 0 | 15 | 0 | 0 |
| 5 | 1 | 29 | 16 | 9 | 0 | 0 | -14 | -129 | 0 | 0 | 0 | 0 | 0 |
| 6 | 1 | 29 | 17 | 8 | 0 | 0 | -23 | -77 | 15 | 0 | 0 | 0 | 0 |
| 7 | 1 | 29 | 22 | 12 | 1 | 2 | 217 | 15 | 8 | 0 | 43 | 20 | 0 |
| 8 | 1 | 29 | 29 | 15 | 0 | 0 | 80 | -69 | 47 | 0 | 5 | 46 | 0 |
| 9 | 1 | 29 | 31 | 10 | 0 | 1 | 155 | 51 | 0 | 0 | 1 | 86 | 0 |
| 10 | 1 | 29 | 32 | 4 | 0 | 0 | 6 | -49 | 0 | 0 | 0 | 0 | 0 |
| 11 | 1 | 29 | 33 | 8 | 0 | 0 | 127 | 104 | 0 | 0 | 3 | 133 | 9 |
| 12 | 1 | 29 | 34 | 11 | 0 | 0 | 18 | -1 | 15 | 0 | 20 | 0 | 0 |
| 13 | 1 | 29 | 37 | 8 | 0 | 0 | 53 | 38 | 53 | 0 | 0 | 27 | 0 |
| 14 | 1 | 29 | 38 | 7 | 0 | 0 | -17 | 65 | 43 | 0 | 11 | 0 | 0 |
| 15 | 1 | 29 | 39 | 9 | 0 | 1 | 424 | 418 | 22 | 0 | 442 | 0 | 0 |
| 16 | 1 | 29 | 41 | 9 | 0 | 0 | -22 | -100 | 18 | 0 | 0 | 0 | 0 |

Query executed successfully. | essql1.walton.uark.edu (13.... | GACL\ES299277 (60) | ES299277 | 00:01:06 | 492169 rows

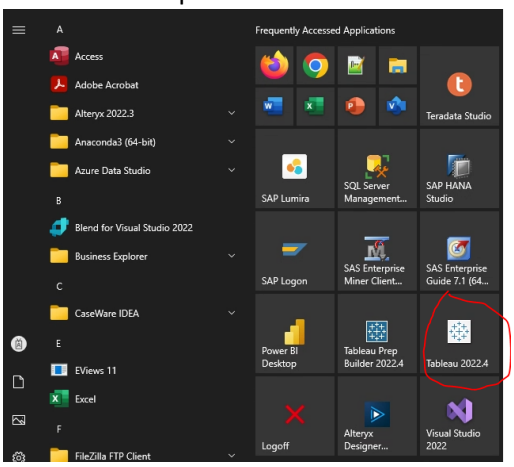
# Analyze Flights Data with Tableau

Now that you have created a data mart from the cleaned flights data in your warehouse, you will use Tableau to develop a visualization similar to that used in [FlightAware's Misery Map](FlightAware's Misery Map).



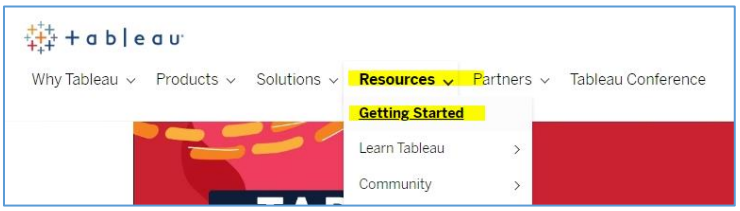
## Tableau Introduction

Tableau is a leading software for Analytics and Business Intelligence[1]. You will access Tableau on the remote desktop. It can take a little while to open.

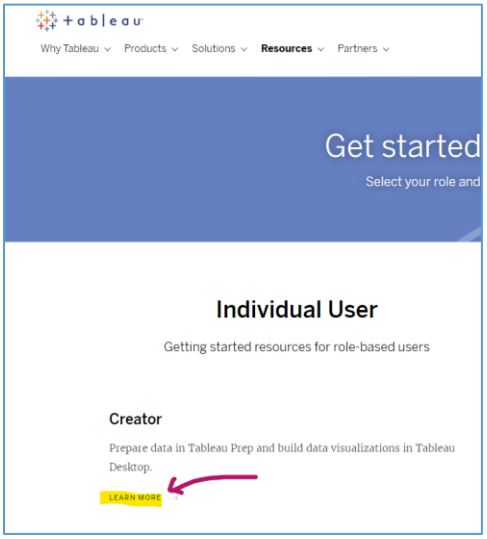



I suggest a short training on Tableau that you can access as follows. If you are interested in this tool, there are many training options.

1. Using a browser on the UArk virtual desktop, go to the Tableau web site (https://www.tableau.com/). Then click Resources and Getting Started.




2. Click Learn More under Creator.



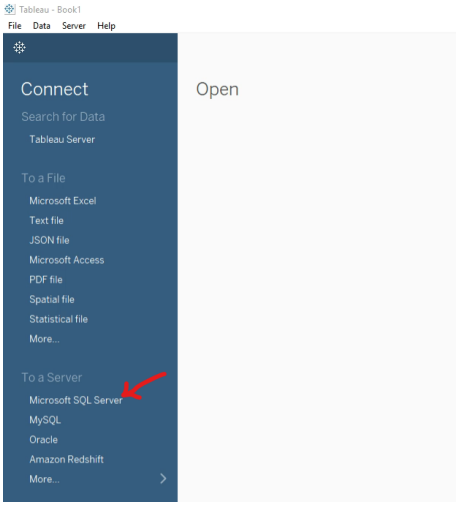

[1] Gartner. (2022, March 22). *Magic Quadrant for Analytics and Business Intelligence Platforms.* https://www.gartner.com/doc/reprints?id=1-29HD7D53&ct=220323&st=sb

3. Scroll down and click Build a Visualization under Build a data visualization.
    - This will require you to create an account. Choose Student in the Job Role options dropdown. (If you already have a Tableau username and password, you can sign it with it.)
    - ***You do <u>not</u> need to download Tableau since you have it on the UArk virtual desktop.***

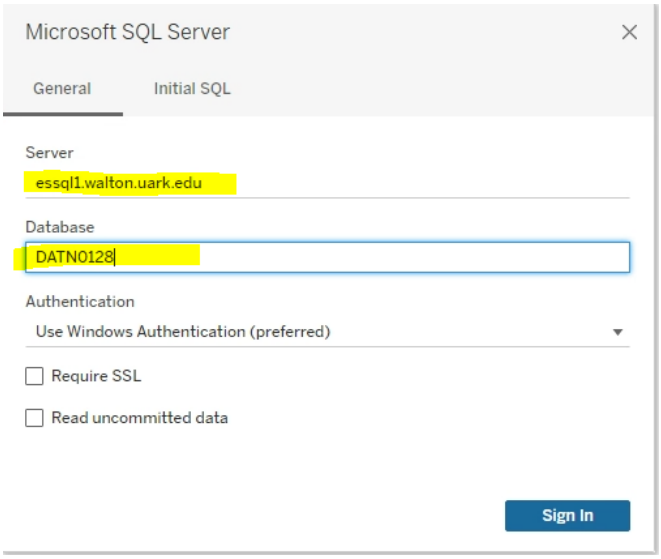
4. After you read the privacy notice, click Watch Video and the training video will start. It lasts about six minutes. You can also download the exercise and solution Tableau workbooks if you want to use them to follow along. Close Tableau afterward.

## Open Tableau and Connect to Your Database

Open Tableau again and choose Microsoft SQL Server from the list of data sources in the CONNECT panel.

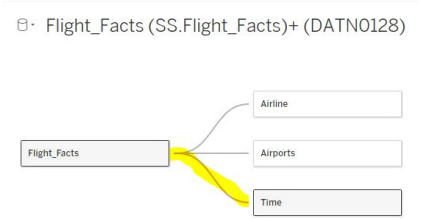


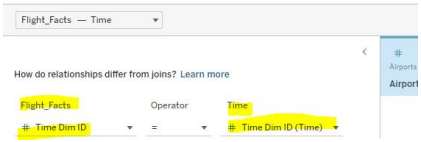

Enter the server information, your database and Windows Authentication for signing in.



## Create the Data Source

Drag your SS tables to the work area. Notice that your FK relationships are recognized. If you click on a relationship, you will see the ON criteria that matches FKs to PKs.




Save your Tableau file to the virtual desktop, naming it FlightsAnalysis.twb. ***Remember to save frequently on the desktop and upload to your OneDrive periodically.***

## Check Measurement Sums

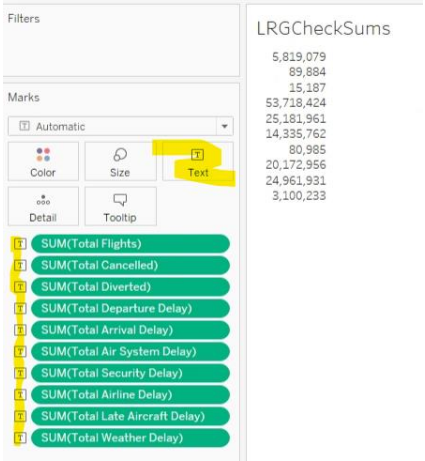
Right click the Sheet 1 tab and rename it *YourInitials*CheckSums (e.g., LRGCheckSums). ***You will not get credit for sheets that do not include your initials in the name.***

After you performed the transformation and load SSIS data flow into W.Flights, you performed a check sum of W.Flights measures against the values in DS.FlightsStaging. The W.Flights sums are shown below.

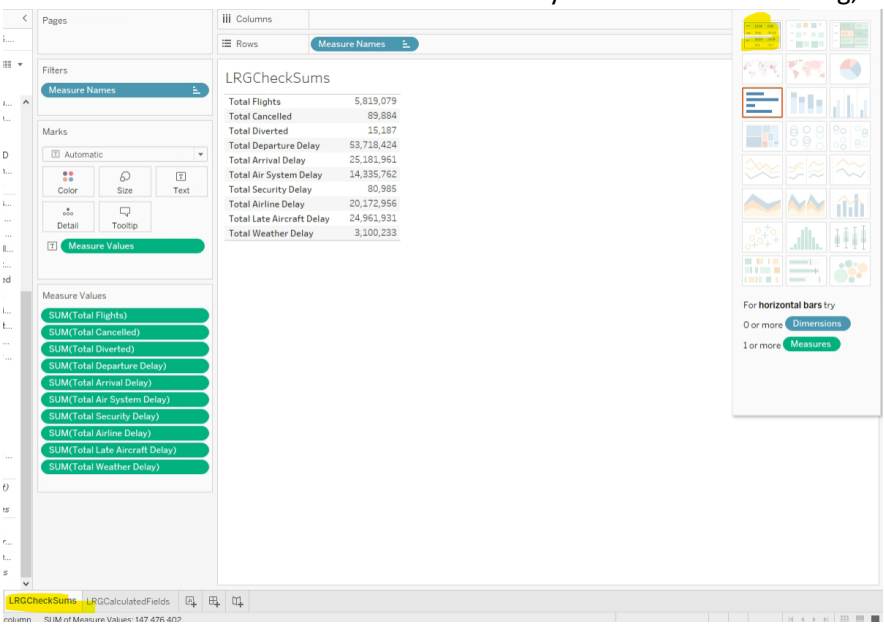
Results | Messages

| | # of Flights | # Cancelled | # Diverted | Total Departure Delay | Total Arrival Delay | Total Air System Delay | Total Security Delay | Total Airline Delay | Total Late Aircraft Delay | Total Weather Delay |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 5819079 | 89884 | 15187 | 53718424 | 25181961 | 14335762 | 80985 | 20172956 | 24961931 | 3100233 |

Since we did not eliminate any flights from the star schema, the check sums should have the same values. Our first Tableau analysis will check this. Drag all the measures to the Text mark. Your worksheet will look like the one below. Check to make sure your values match these.



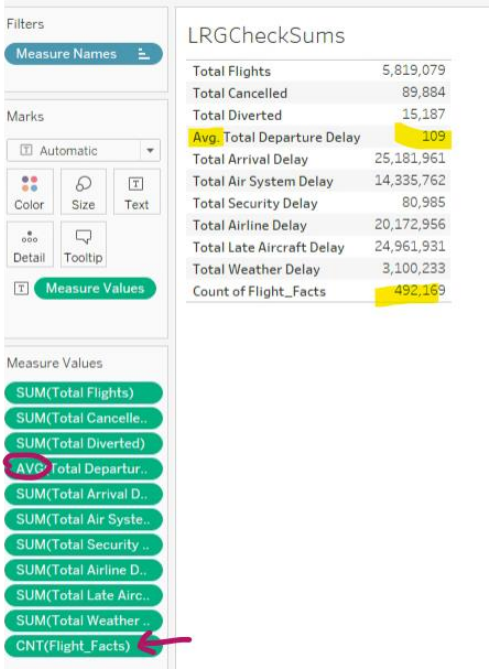
Then click Text Table in the Show Me tab and you will see the following, after rearranging the row order.



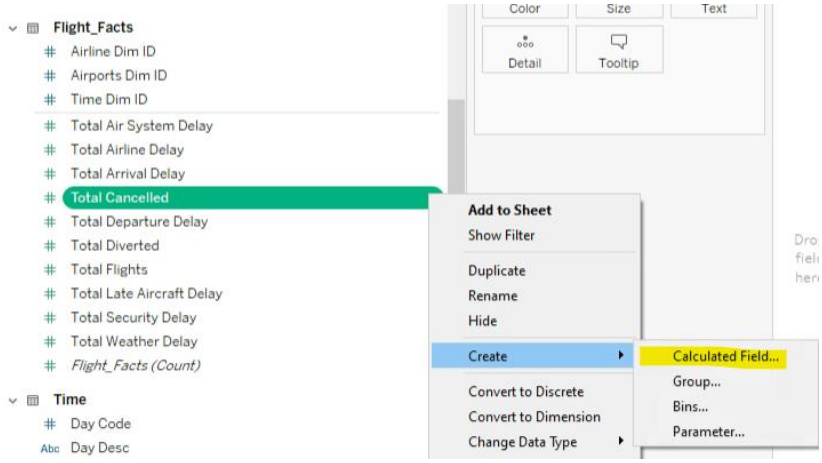

Q7: Snip and paste your CheckSums worksheet, including the tab, into your Word document.

## Create Calculated Fields

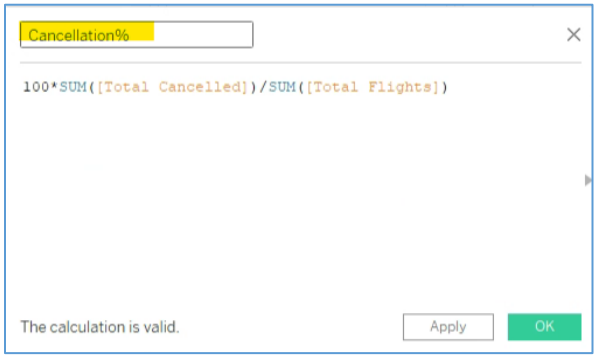
There is an option to change the SUM() to AVG() in the Measure Values. Since we have pre-aggregated facts from the star schema, this will not produce correct results. Using departure delay as an example, the average departure delay should be 53,718,424/5,819,079 (total departure delay/number of flights). This equals 9.2 minutes. If we use AVG(Total Departure Delay), we see 109 minutes. This is because there are 491,169 flight facts (53,718,424/492,169 = 109 minutes). For this reason, we will need to create calculated fields to obtain the correct values.



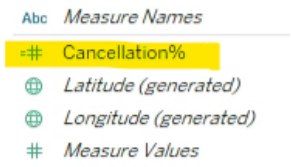
Right-click your check sums worksheet and duplicate it, renaming the copy, *YourInitials*CalculatedFields. Remove all the measures and expand the Total Cancelled options to create a calculated field.



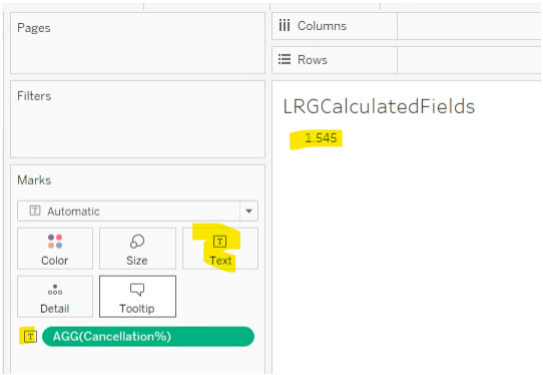

***Note:*** It does not matter which column you expand. It makes the formula a little easier to build by selecting a column used in it. Name the column Cancellation%, build the formula and click OK.




Notice that it appears as a calculated measure under the Measure Names.



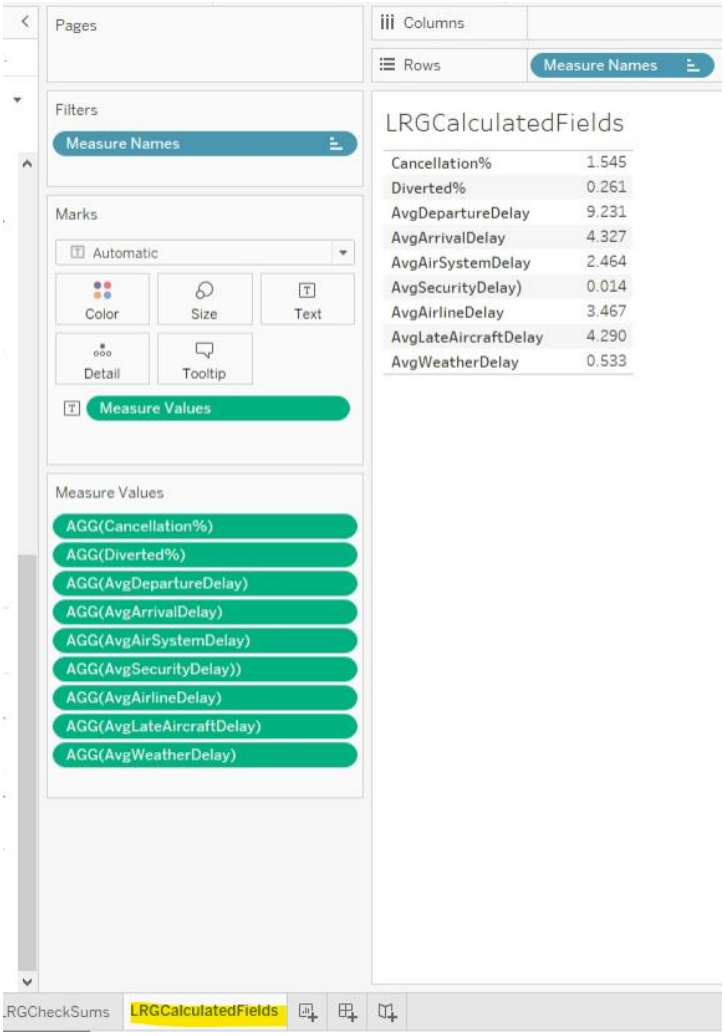

Drag it to the Text mark and you will see the following.



Using a similar approach, create the following calculated figures with names in parentheses.

- Percentage of flights diverted (Diverted%)
- Average departure delay minutes (AvgDepartureDelay)
- Average arrival delay minutes (AvgArrivalDelay)
- Average air system delay minutes (AvgAirSystemDelay)
- Average security delay minutes (AvgSecurityDelay)
- Average airline delay minutes (AvgAirlineDelay)
- Average late aircraft delay minutes (AvgLateAircraftDelay)
- Average weather delay minutes (AvgWeatherDelay)

Drag the remaining eight calculated figures to the Text mark, then create a Text Table. It should look like the image below, after rearranging the measure order. Make sure all your values match.



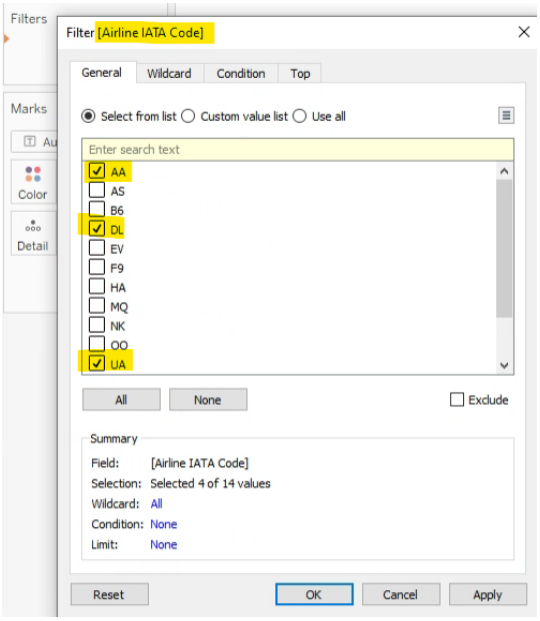

Q8: Snip and paste your calculated fields worksheet, including the tab, into your Word document.

Q9: Discuss what you learned about the airline performance from these measures.

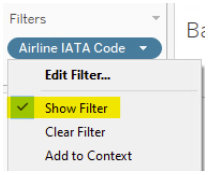
## Create BasicComparisons Sheet

Duplicate your calculated fields sheet and rename the copy, *YourInitials*BasicComparisons. Remove all measures except the cancellation percentage, diverted percentage, average departure delay and average arrival delay.

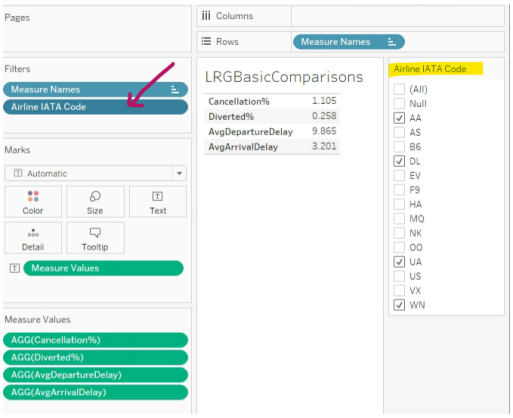
Drag Airline IATA Code to the Filters area and select AA (American Airlines), DL (Delta), UA (United Airlines) and WN (Southwest Airlines) and click OK. These four airlines hold the largest market shares.



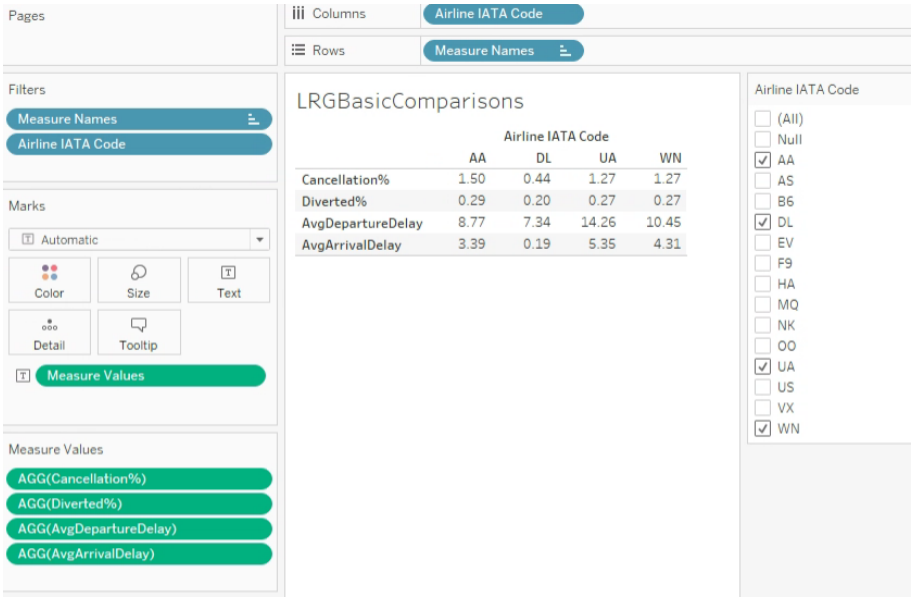

Then click Airline IATA Code in the Filters area and choose Show Filter.



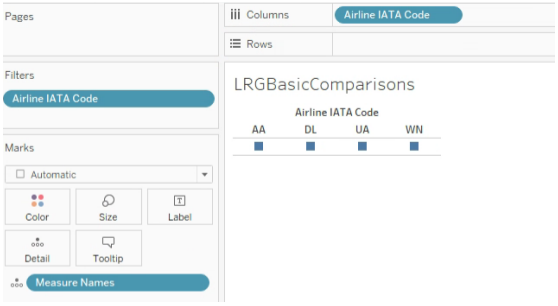

The filter values will show on the right.



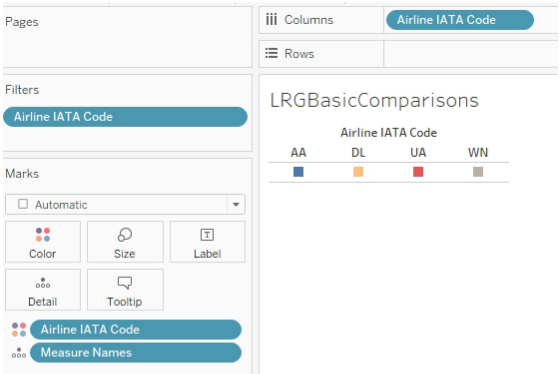
We will now create a comparative evaluation of these performance indicators. First, drag Airline IATA Code to the columns. This is an additional use of Airline IATA code beyond the filter. Your worksheet will now look like the following. You can add and subtract airlines from the comparison using the filter checkmarks on the right.



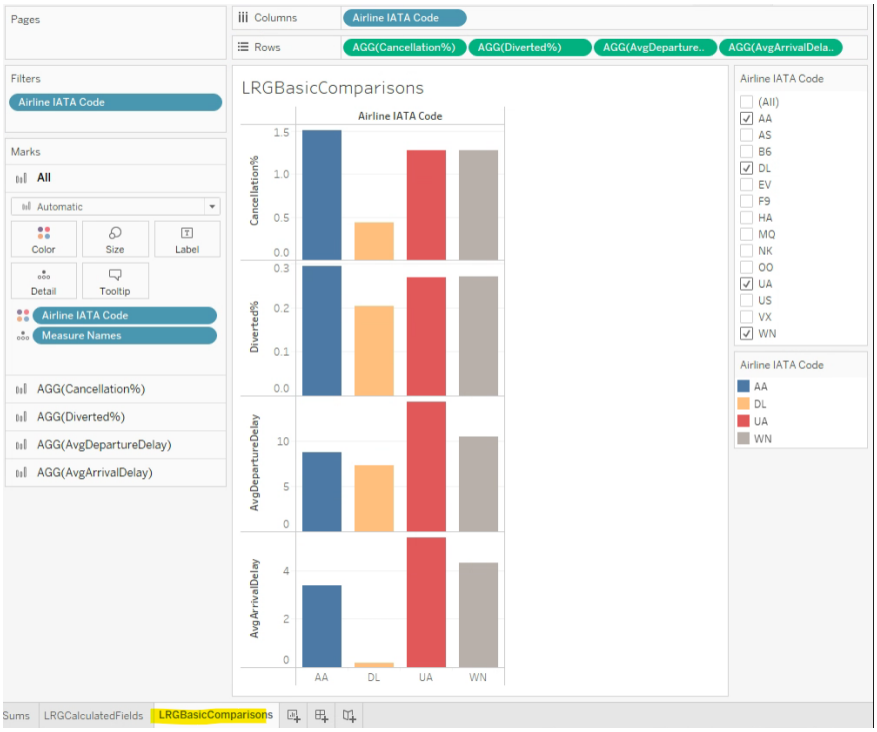

We will now prepare a comparative visualization of these performance indicators. Remove Measure Values from the Marks area and Measure Names from the Filters and Rows. This leaves only the four airline IATA codes. You will see a display similar to the one below.




As a third use of airline IATA codes, drag Airline IATA code to the colors mark. Notice that each airline has its own color now.



Drag the four calculated columns to the Rows. You should see a comparison for each measure like the one below.




Q10: Snip and paste your basic comparisons worksheet, including the tab, into your Word document.

Q11: Discuss what you learned about the four airlines' 2015 performance from this visualization.

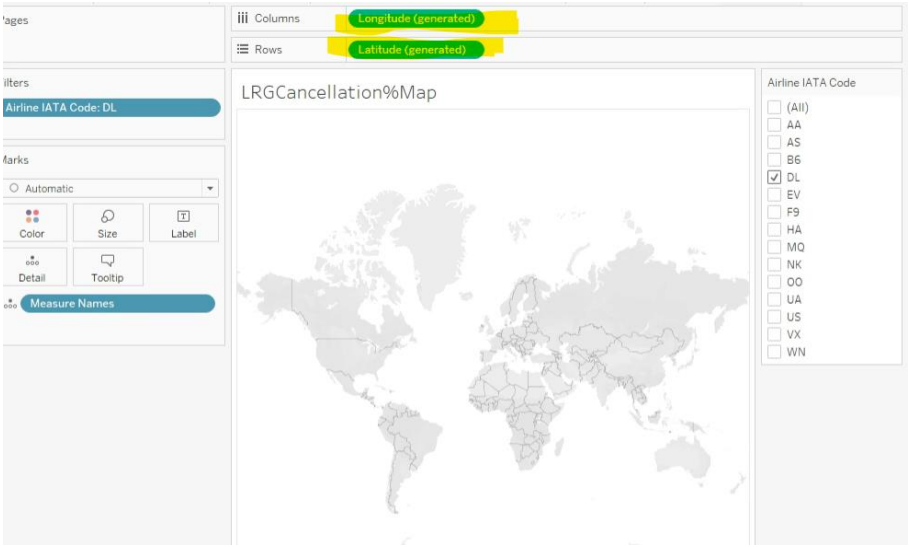
## Create Cancellation% Map Analysis

Duplicate your basic comparisons sheet and rename the copy, *YourInitials* Cancellation%Map. The map is intended for Delta Airlines managers to identify origin airports with the highest cancellation percentages.

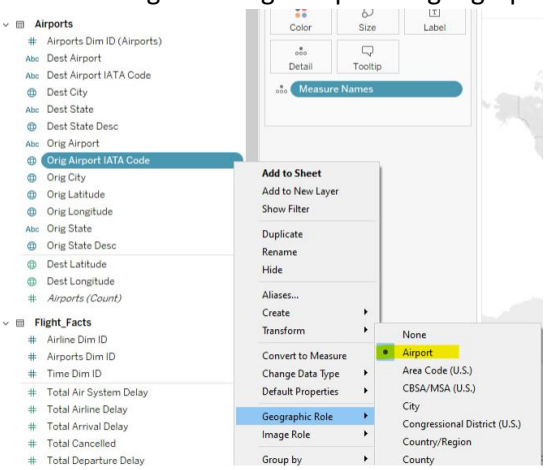
Do the following:

- Remove all airlines from the filter except DL
- Remove Airline IATA Code from the Marks area and Columns shelf
- Remove all measures from the Rows shelf
- Place Longitude (generated) on the Columns and Latitude (generated) on the Rows

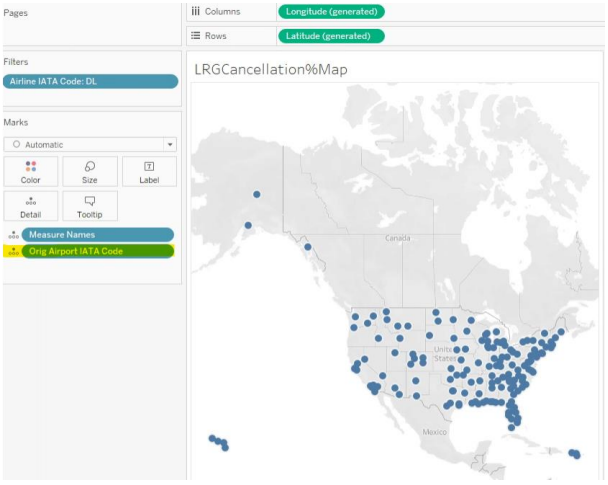
You will see a world map appear in the visualization area.



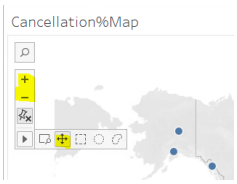

You will assign the origin airport its geographic role by expanding its options and choosing Airport.



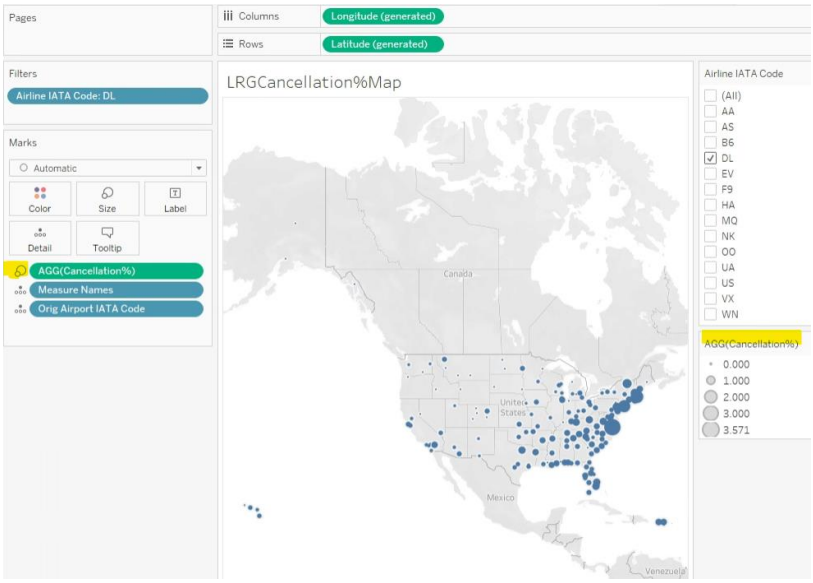

Then drag it to the Marks area and you will see the airports appear on the map.



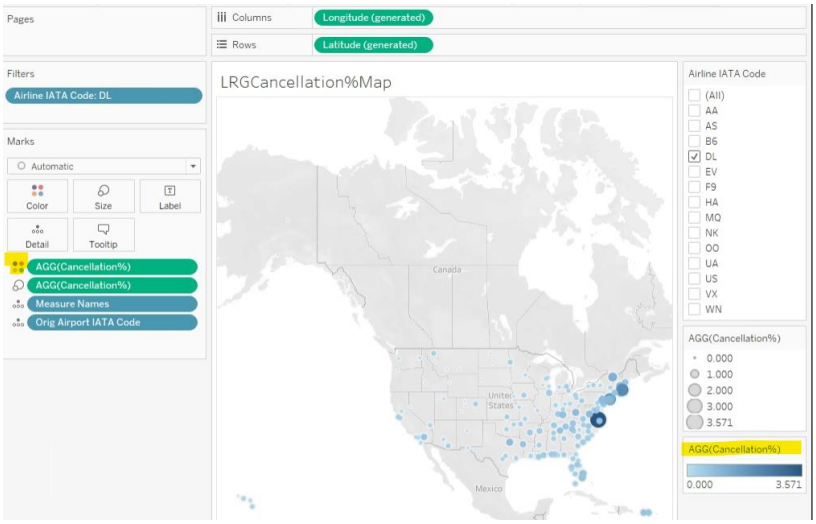
You can resize the map and use the Pan tool to move around it.



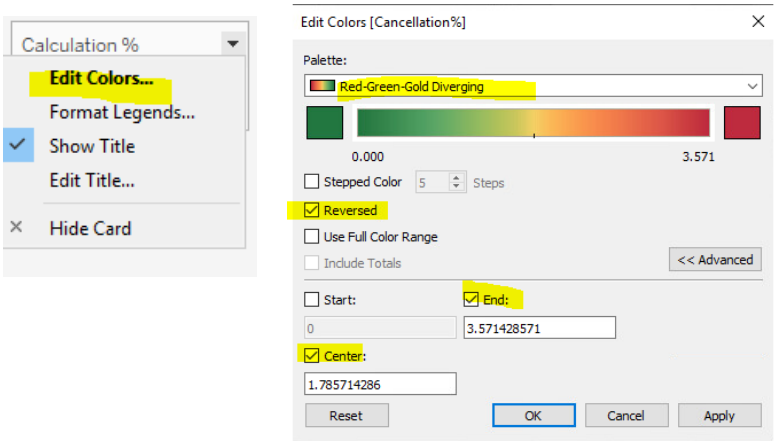

Drag Calculation% to the Size option and see how it changes the map and sheet. Notice that now there is a legend for the different sizes.



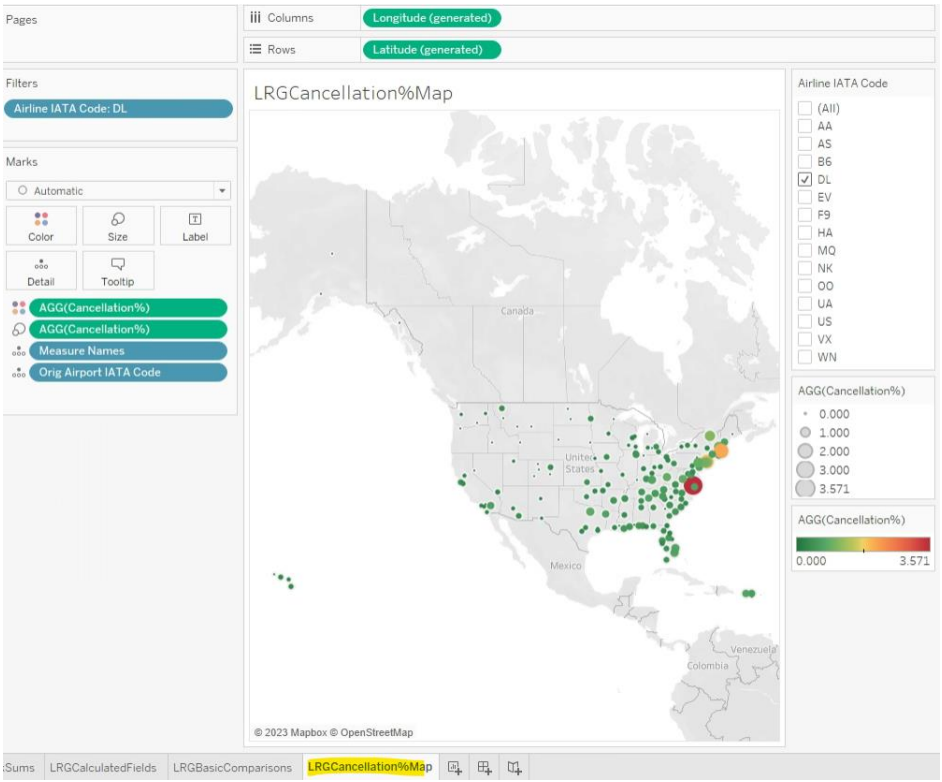

Also drag the Calculation % to the Color option. This adds a legend for the color shading that goes with the shapes



We will change the color code to a Green-Yellow-Red option. Use the color legend dropdown on the bottom right and choose Edit Colors. Choose Red-Green-Gold Diverging and click Reversed so that Green will correspond to small cancellation percentages. Then click Advanced and set the Start, End and Center values obtaining them from the data. (You could also specify these values rather than use the statistics. Then click OK.




Your results should look similar to the following, similar to the Misery Map. Notice that if you mouse over the airports, the airport code appears and the cancellation %.




Q12: Snip and paste your cancellation percentages worksheet, including the tab, into your Word document.

Q13: Determine Delta's problematic airports and their cancellation percentages. Include them in the response to this question.

## Create Your Own Analysis

There are many other analyses that would be useful to Delta Airlines managers. Think about measures or dimensions we have not used. Play with Tableau and develop a ***new analysis***, making sure that it would likely be useful for Delta Airlines managers. Use measures and/dimensions we have not used. You will be graded on the quality of new information it provides beyond the visualizations you've already created.

Q13: Snip and paste your own analysis worksheet, including the tab, into your Word document.

Q14: Describe the new information your visualization could provide Delta Airlines managers as well as how they might use it.