



**The British College**  
KATHMANDU



**Coursework Submission Coversheet**  
(individual coursework only)

**Faculty of Arts, Environment and Technology**

**LBU Student Id:**

77202323

Please ensure all information is complete and correct and attach this form securely to the front of your work before posting it in a coursework collection box.

Award name: BSc Hons in Computing

Module code: 15781

Module name: Advance Database System (ADS)

Module run: 2020/21

Coursework title: Data warehouse Design and Development

Due Date: 22 December, 2020

Module leader: (In LBU): Jackie Campbell, Sanelia Lazarevski

Module tutor: (In TBC): Dibya Tara Shakya

**TURNITIN Checked:** YES  NO *(please circle)*

Submission date& time: Date: 22 December, 2020 Time: After evening

**Total Word Count: 15350 approx.**

**Total Number of Pages (including this front sheet): 117**

**In submitting this form with your assignment, you make the following declaration:**

I declare, that the coursework submitted is my own work and has not (either in whole or part) been submitted towards the award of any other qualification either at LBU or elsewhere. I have fully attributed/referenced all sources of information used during the completion of my assignment, and I am aware that failure to do so constitutes an assessment offence.

Signed: Avhimanhu Sapkota

Date: 22 December, 2020

**You are strongly advised to retain a second copy of your work in case of any query about the assignment.**

**For completion by the faculty:**

**This mark is provisional and subject to moderation and approval by the relevant examining board**

**Teacher's Feedback**

**Teacher's Signature:** \_\_\_\_\_

**Date:** \_\_\_\_\_

## Contents

Task 1: Data Warehouse (DW) Star Schema Design .....	4
Introduction to Data warehouse .....	4
Identify 3-5 reports that your star schema will support.....	4
Document the star schema (SS) design model to support the reports. ....	5
Use the data dictionary template from tutorials to document the data model of the project .....	8
Data Dictionary: Supports data sourcing and data integration .....	9
Data Dictionary: ETL required for star schema .....	14
Select one of the reports you have suggested. Illustrate the expected data in the star schema to support the report. ....	18
Task 2: Pre-processing: Gather the data sources prior to ETL.....	22
Collect only the data you require to populate the star schema into the data staging area.....	22
Star Schema Set Up (DM environment).....	30
Task 3: Extract, Transform and Load (ETL) script to populate the Star Schema (DM) with data .....	39
Extraction.....	39
Transformation .....	51
Transformation: Bad Data Identification, Segregation and Insert.....	51
Transformation: Good Data Identification and Insert .....	60
Transformation: Bad Data Cleaning and Insert into Clean Tables .....	66
Transformation: Into Transform Tables.....	76
Loading.....	84
Task 4: Data Analysis/OLAP/Mining Investigation.....	97
Data Analysis.....	97
Business Intelligence.....	97
Online Analytical Processing (OLAP) .....	97
Tableau: Reports, Charts and Dashboards.....	97
Additional Excel Sheet Reports .....	107
Drilled Down Reports.....	109
Task 5 .....	111
Introduction .....	111
Bill Inmon's Approach .....	111
Ralph Kimball's Approach .....	113
Key Differences between Ralph Kimball's Approach and Bill Inmon's Approach.....	114

Conclusion.....	114
References .....	116

## Task 1: Data Warehouse (DW) Star Schema Design

### Introduction to Data warehouse

Data Warehouse is a collection of integrated databases designed to support a Decision Support System (DSS). According to Bill Inmon, data warehouse is a collection of integrated, subject-oriented databases designed to support the DSS function, where each unit of data is non-volatile and relevant to some moment in time.

The prime purpose of Data Warehouse is to support decision making which will perhaps be used for evaluating future strategy. Thus, it is also used as the basics of business decision making, business intelligence solutions and data analytics dashboards and applications. It can be used to convert raw data into meaningful information which results in enhanced business and sustainable decision making, which can also be stated as Business Intelligence. It fundamentally combines information from several sources into one comprehensive database making data analysis, decision making and data mining possible. The proper analysis of data warehouse can benefit a company in product development, marketing, pricing strategy, historical analysis and customer satisfaction which further helps them to adapt changing business conditions. (*InvestingAnswers, 2020*).

Data warehouse is a large repository of data that contains information from heterogenous sources which can be used for various purposes within an organization. The data warehouse can be further divided into Data Marts in order to perform specific business-oriented analysis. Data Mart is basically a subset of data warehouse which contains prototypes of summarized data obtained, within an organization, for analysis on a specific section. (*Panoply.IO, 2020*)

Some more detail advantages of Data warehouse can be listed as:

- It allows business to quickly access data insights
- It helps to reduce time for analysis and visual data representations
- It helps users to analyze data from different time periods to identify trends (*Guru99, 2020*)

For instance, FlyU airline company, in the assignment case study, has recorded their data of flights and its details, customers, weather details and complaints. A data warehouse of the raw data of FlyU can be analysed, which may benefit them in quality service delivery, customer satisfaction and upliftment of the company by increasing sales and profits. However, data mart is concerned with the specific data analysis rather than analysing every aspect of FlyU.

Being assigned a role of an analyst/developer on a Data Mart project, we were supposed to choose one Key Performance Indicators (KPI) which were considered by FlyU airline company. As quoted by Tony Alessandra, ‘Being on par in terms of price and quality gets you into the game. Service wins the game.’ I believe that quality service is the key to organization’s sustainable growth. Thus, I have chosen KPI 1 which is concerned with the quality of their services and provide good flight experience to their customers.

Identify 3-5 reports that your star schema will support.

After the detail study and investigation of raw data provided via detailed flight information for 2017 and 2018, weather information, and FlyU complaints system, some reports, which will inform FlyU in order to meet their business objectives being based on quality services, have been identified as follows:

**1. Total number of flights cancelled on the basis of cancellation reasons.**

One of the factors determining quality of service in an airline company is ‘flight cancellation’. Therefore, in order to improve the quality service, FlyU must be aware of total number of flights that were cancelled along with their reason. The study of such data will make them prepared and hence they can resolve the cancellation reasons before the flight gets cancelled. For instance, if most of the flights have been cancelled due to computer glitches then the company can ensure that all the computer glitches are checked and tested before every flight is scheduled.

**2. List of airlines whose flights were delayed, in the year 2018, along with the total delayed time.**

Although FlyU is an airline company, they also facilitate their customers with other associate airlines. Hence, the report will help them identify airlines whose flights are being delayed along with total delayed time. As a result, for instance, they may cancel or shorten contracts with airline companies whose flights are frequently delayed which will ensure their quality service as most airlines whose flights are usually delayed will no more be their associate partners.

**3. List of top 10 aircraft delayed due to air system delay including their delayed time.**

Air system delay refers to delay usually caused because of some mechanical issues in an aircraft. Thus, this report will inform FlyU’s concerned officials to identify ten most problematic aircraft so that they can repair, discard or manage those problematic aircraft for future flights. Proper maintenance of aircrafts always results in quality service for any airline company.

**4. Total arrival delay time for each month in the year 2018**

As delay time affects the quality of service directly, the report identified will inform FlyU airline company regarding total arrival delay of each flight on the basis of months in the year of 2018. This will help them to make decisions on total number of flights to book in those months in the future.

**5. List of average PRCP, average AWND and average TAVG of weather on the basis of quarter of the year.**

The above report will inform FlyU about the average weather data on the basis of seasons (quarter) so that they can plan the number of flights to book in those quarter of the year being based on the weather predictions and historical data to ensure less flight delays and cancellation due to weather conditions.

Document the star schema (SS) design model to support the reports.

Data warehouse system is necessary in order to produce these reports. The foremost step to start building a data warehouse begins with designing a model of data warehouse. There are two main multidimensional

schemas designed to address the requirements of large databases. I propose star schema to design the data warehouse architecture.

Star schema is a data warehouse schema in which the centre of the star can have one fact table and many other associated dimension tables.

Some of the characteristics of star schema are:

- Every dimension is represented with one-dimension table which should contain attributes.
- Dimension table is joined to fact table using foreign key but they are not joined to each other.
- Dimension tables are not normalized

The other data warehouse design methodology is snowflake schema. It is a logical arrangement of tables, which is an extension of a Star Schema where dimension tables are normalized and can be connected to each other.

Some of the characteristics of snowflake schema are:

- It uses smaller disk space during data warehouse design
- Due to multiple tables connection, query implementation is reduced.
- It is easier to implement dimension tables along with the facts. (Guru99, 2020)

Hence, I propose star schema over snowflake schema as the data warehouse designing model because it is optimized for querying large data sets to generate appropriate reports. Furthermore, it is easy to understand and provides optimal disk usages which is more feasible and applicable. Whereas, snowflake schema will have many dimension tables resulting in complex database design which will require many joins to fetch the data. Therefore, I believe star schema is the best design methodology to design data warehouse for FlyU airlines company.

The foremost stage to design a star schema is to document the fact tables, dimensions, attributes, keys and measures which have been identified as follows:

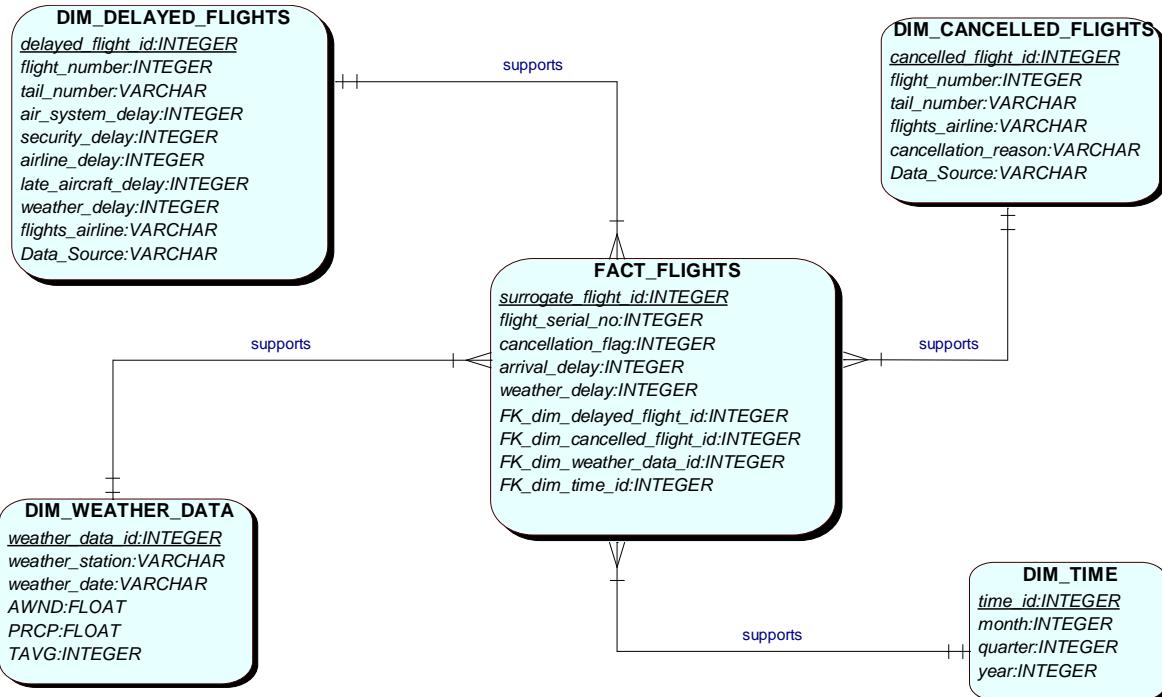
**Fact Table:** Flights

**Measures:** Flight\_number (alias: flight\_serial\_no), cancellation (alias: cancellation\_flag), departure\_delay, arrival\_delay, weather\_delay

**Dimension Tables and attributes:**

- Delayed\_Flights: delayed\_flight\_id, flight\_number, tail\_number, air\_system\_delay, security\_delay, airline\_delay, late\_aircraft\_delay, weather\_delay, flights\_airline.
- Cancelled\_Flights: cancelled\_flight\_id, flight\_number, tail\_number, flights\_airline, cancellation\_reason.
- Weather\_data: weather\_data\_id, weather\_station, AWND, PRCP, TAVG.
- Time: time\_id, month, quarter, year

The star schema model for FlyU airline company to improve quality of service:



The Star Schema above has **FACT\_FLIGHTS** as the fact table which has been derived from the data of Flights\_2017, Flights\_2018 and FlyU\_Flights. A surrogate key: surrogate\_flight\_id has been added because flights data provided did not have a specific primary key column. The measure flight\_serial\_no is alias name for flight\_number of the provided data. It is necessary to know the flight number of each flight hence it has been added as a measure for the fact. Similarly, cancellation\_flag is an alias name for cancellation data of the provided data, which stores 0 if flight was not cancelled and 1 if the flight is cancelled, which helps to ensure whether the flight was cancelled or not. The measure arrival\_delay has been identified in order to know whether each flight have been delayed or not. Finally, weather\_delay has been added as a measure to confirm if the flight has been delayed because of weather conditions.

The dimension table **DIM\_DELAYED\_FLIGHTS** has been derived from the data of Flights\_2017, and Flights\_2018. A surrogate key: delayed\_flight\_id has been added as primary key because the flights data provided did not have a specific primary key column. The dimension will store all the data of delayed flights including flight\_number, tail\_number, air\_system\_delay, security\_delay, airline\_delay, late\_aircraft\_delay, weather\_delay, and flights\_airline. All of the data has been added from the data sources provided however, flights\_airline is an alias name for airline provided in Flights\_2018.

The dimension table **DIM\_CANCELLED\_FLIGHTS** has also been derived from the data of Flights\_2017, Flights\_2018 and FlyU\_Flights. A surrogate key: cancelled\_flight\_id has been added as primary key because the flights data provided did not have a specific primary key column. The dimension will store all the data of cancelled flights including, flight\_number, tail\_number, flights\_airline and cancellation\_reason. All of the data added are from the data sources provided however, flights\_airline is an alias for airline provided in Flights\_2018.

The dimension table DIM\_WEATHER\_DATA has been derived from the data of Weather\_data. A surrogate key: weather\_data\_id has been added as primary key as the weather data provided did not have a specific primary key column. The dimension will store weather data including information of weather\_station, weather\_date, AWND, PRCP, and TAVG.

The dimension table DIM\_TIME stores the time data month, quarter (on the basis of seasons) and year which has been derived from all the data sources provided so that the reports can be filtered on the basis of month and quarter of each year.

[Use the data dictionary template from tutorials to document the data model of the project](#)

### **Granularity of data**

Firstly, granular data is simply known as detailed low-level data that is in pieces in order to be more defined and detailed. The granular data can be molded in any way which can help in producing meaningful information to add up to business intelligence and data analysis. In the context of data warehouse, granularity is a measure of the degree of detail presented in a fact table. Hence, the necessary data have been identified from the provided source and they have been documented below.

### **Data Dictionary**

A data dictionary is a description of data including data about the data such as datatypes, details of structure, and security restrictions. It results in higher-quality metadata as it is a disciplined and systematic approach to managing definitions (Dataversity.net, 2017). Data dictionary, in general, is data about the database. It contains information related to the database including what is in the database, where was the data brought from, data quality issues, and what actions were taken to resolve those issues. The data dictionary which describes the data of FlyU airline company has been documented as follows:

Data Dictionary: Supports data sourcing and data integration

<b>Data Source(s)</b>	<b>Field Name</b>	<b>Data Type</b>	<b>Key</b>	<b>DQ Source</b>	<b>Data Quality Check</b>	<b>Data quality Issues</b>	<b>Action Note</b>
Flights_2017	tail_number	Varchar2	No	Flights_2017	Incomplete data	Many data missing for aircraft's tail number	Add data with 'UNKNOWN' data for empty fields
Flights_2018		Varchar2	No	Flights_2018	N/A	N/A	Import the data without any changes
FlyU_Flights		Varchar2	No	FlyU_Flights	N/A	N/A	Import the data without any changes
Definition:	Column which stores the unique tail number of an aircraft.						
Notes:	tail_number exists in all data sources however one data source has missing values which may affect the report data as tail_number is required to identify aircraft.						
<b>Data Source(s)</b>	<b>Field Name</b>	<b>Data Type</b>	<b>Key</b>	<b>DQ Source</b>	<b>Data Quality Check</b>	<b>Data quality Issues</b>	<b>Action Note</b>
Flights_2017	flights_airline	No Data	No	Flights_2017	Incomplete data	Data not available	Add data with 'UNKNOWN' data for empty fields
Flights_2018		Varchar2	No	Flights_2018	N/A	N/A	Import the data without any changes
FlyU_flights		No Data	No	FlyU_flights	Incomplete data	Data not available	Add data with 'UNKNOWN' data for empty fields
Definition:	Column which stores the abbreviation of airline companies such as 'DL', 'US', 'EV', etc.						
Notes:	Flights_airline exists in flight_2018 only which is necessary to identify airlines. However, it does not exist in flight_2017 and FlyU_Flights. Although its value is not required, random data which clarifies null is better than keeping the column.						
<b>Data Source(s)</b>	<b>Field Name</b>	<b>Data Type</b>	<b>Key</b>	<b>DQ Source</b>	<b>Data Quality Check</b>	<b>Data quality Issues</b>	<b>Action Note</b>

Flights_2017	cancellation_reason	Varchar2	No	Flights_2017	Unclear data	'Char' data provided which is unclear	Change the char data with data which give proper reasons for cancellation	
Flights_2018		Varchar2	No	Flights_2018	Unclear data	'Char' data provided which is unclear	Change the char data with data which give proper reasons for cancellation	
FlyU_flights		Varchar2	No	FlyU_flights	Unclear data	'Char' data provided which is unclear	Change the char data with data which give proper reasons for cancellation	
Definition:	Column which stores the reason of flight cancellation.							
Notes:	Cancellation_reason exists in all data sources however they are meaningless as they contain reasons like 'A', 'B' or 'C'. Thus, a proper cancellation reason is necessary to display.							
Data Source(s)	Field Name	Data Type	Key	DQ Source	Data Quality Check	Data quality Issues	Action Note	
Flights_2017	air_system_delay	Number	No	Flights_2017	Inconsistent data	'0' for some rows while null value for some rows	Change null data with '0' to maintain consistency and to manage incomplete data	
Flights_2018					Incomplete data	Many rows contain null values if no-delay		
		Number	No	Flights_2018	Inconsistent data	'0' for some rows while null value for some rows	Change null data with '0' to maintain consistency and to manage incomplete data	
Definition:	Column which stores minutes as number which represents the delayed minutes due to air_system_delay.							
Notes:	Air_system_delay exists in both data which stores total minutes delayed because of the particular reason. It is necessary to calculate reports identified in task 1.							
Data Source(s)	Field Name	Data Type	Key	DQ Source	Data Quality Check	Data quality Issues	Action Note	

Flights_2017		Number	No	Flights_2017	Inconsistent data	'0' for some rows while null value for some rows	Change null data with '0' to maintain consistency and to manage incomplete data				
					Incomplete data	Many rows contain null values if no-delay					
Flights_2018	security_delay	Number	No	Flights_2018	Inconsistent data	'0' for some rows while null value for some rows	Change null data with '0' to maintain consistency and to manage incomplete data				
					Incomplete data	Many rows contain null values if no-delay					
Definition:	Column which stores minutes as number which represents the delayed minutes due to security_delay .										
Notes:	security_delay exists in both data which stores total minutes delayed because of the particular reason. It is necessary to calculate reports identified in task 1.										
Data Source(s)	Field Name	Data Type	Key	DQ Source	Data Quality Check	Data quality Issues	Action Note				
Flights_2017		Number	No	Flights_2017	Inconsistent data	'0' for some rows while null value for some rows	Change null data with '0' to maintain consistency and to manage incomplete data				
					Incomplete data	Many rows contain null values if no-delay					
Flights_2018	airline_delay	Number	No	Flights_2018	Inconsistent data	'0' for some rows while null value for some rows	Change null data with '0' to maintain consistency and to manage incomplete data				
Definition:	Column which stores minutes as number which represents the delayed minutes due to airline_delay .										
Notes:	airline_delay exists in both data which stores total minutes delayed because of the particular reason. It is necessary to calculate reports identified in task 1.										
Data Source(s)	Field Name	Data Type	Key	DQ Source	Data Quality Check	Data quality Issues	Action Note				

Flights_2017		Number	No	Flights_2017	Inconsistent data	'0' for some rows while null value for some rows	Change null data with '0' to maintain consistency and to manage incomplete data				
					Incomplete data	Many rows contain null values if no-delay					
Flights_2018	Late_aircraft_delay	Number	No	Flights_2018	Inconsistent data	'0' for some rows while null value for some rows	Change null data with '0' to maintain consistency and to manage incomplete data				
					Incomplete data	Many rows contain null values if no-delay					
Definition:	Column which stores minutes as number which represents the delayed minutes due to Late_aircraft_delay .										
Notes:	Late_aircraft_delay exists in both data which stores total minutes delayed because of the particular reason. It is necessary to calculate reports identified in task 1.										
Data Source(s)	Field Name	Data Type	Key	DQ Source	Data Quality Check	Data quality Issues	Action Note				
Flights_2017		Number	No	Flights_2017	Inconsistent data	'0' for some rows while null value for some rows	Change null data with '0' to maintain consistency and to manage incomplete data				
					Incomplete data	Many rows contain null values if no-delay					
Flights_2018	weather_delay	Number	No	Flights_2018	Inconsistent data	'0' for some rows while null value for some rows	Change null data with '0' to maintain consistency and to manage incomplete data				
Incomplete data	Many rows contain null values if no-delay	Change null data with '0' to maintain consistency and to manage incomplete data									
Definition:	Column which stores minutes as number which represents the delayed minutes due to weather_delay .										
Notes:	weather_delay exists in both data which stores total minutes delayed because of the particular reason. It is necessary to calculate reports identified in task 1.										
Data Source(s)	Field Name	Data Type	Key	DQ Source	Data Quality Check	Data quality Issues	Action Note				

Weather_Data	Weather_date	Varchar	No	Weather_Data	Inconsistent data	Date are stored as: 01/03/2017 and 2/3/2017 which creates issues while transferring	Extracting data using / as delimiter and changing those to same format.
Definition:	Column which stores date of weather as varchar.						
Notes:	Weather_date exists in weather_Data only however it is necessary to produce report number 5 identified in task 1.						

Data Dictionary: ETL required for star schema

Star schema table	Attribute name	Data Type	Key	DQ Source	Data source field (mapping)	Data quality issues	Transformation
DIM_DELAYED_FLIGHTS	delayed_flight_id	number	Yes	Automatically generated as surrogate key	N/A	N/A	Create a sequence and trigger to generate key and to assign them when a new row is inserted.
	flight_number	number	No	Flights_2017, Flights_2018	Flights_2017.flight_number Flights_2018.flight_number	N/A	Merge the data from both sources.
	tail_number	varchar	No	Flights_2017, Flights_2018	Flights_2017.tail_number Flight_2018.tail_number	Missing values in 2017	Clean by inserting 'UNKNOWN' where null and extract then insert data.
	air_system_delay	number	No	Flights_2017, Flights_2018	Flights_2017.air_system_delay Flight_2018.air_system_delay	Missing values	Clean by inserting 0 where null and extract then insert data.
	security_delay	number	No	Flights_2017, Flights_2018	Flights_2017.security_delay Flight_2018.security_delay	Missing values	Clean by inserting 0 where null and extract then insert data.
	airline_delay	number	No	Flights_2017, Flights_2018	Flights_2017.airline_delay Flight_2018.airline_delay	Missing values	Clean by inserting 0 where null and extract then insert data.
	late_aircraft_delay	number	No	Flights_2017, Flights_2018	Flights_2017.late_aircraft_delay Flight_2018.late_aircraft_delay	Missing values	Clean by inserting 0 where null and extract then insert data.

	weather_delay	number	No	Flights_2017, Flights_2018	Flights_2017. weather_delay Flight_2018. weather_delay	Missing values	Clean by inserting 0 where null and extract then insert data.
	flights_airline	varchar	No	Flights_2018	Flight_2018. airlines	Missing in Flights_2017	Clean by inserting 'UNKNOWN' for flights_2017
Definition:	Dim_delayed_flights holds the data of flights which has been delayed either during departure or during arrival. It has delay reasons along with delayed minutes which will be helpful to produce reports based on actual delayed time (in minutes).						
Notes:	Although, data from arrival and departure delay are stored, only the arrival delay data will be linked with fact table.						
Star schema table	Attribute name	Data Type	Key	DQ Source	Data source field (mapping)	Data quality issues	Transformation
DIM_CANCELLED_FLIGHTS	cancelled_flight_id	number	Yes	Automatically generated as surrogate key	N/A	N/A	Create a sequence and trigger to generate key and to assign them when a new row is inserted.
	flight_number	number	No	Flights_2017, Flights_2018, Flyu_flights	Flights_2017. flight_number Flights_2018. flight_number Flyu_flights. flight_number	N/A	Merge the data from all three sources.
	tail_number	varchar	No	Flights_2017, Flights_2018, Flyu_flights	Flights_2017. tail_number Flight_2018. tail_number Flyu_flights. tail_number	Missing values in 2017 and flyu_flights	Clean by inserting 'UNKNOWN' where null and extract then insert data.
	flights_airline	varchar	No	Flights_2018	Flight_2018. Airlines	Missing in Flights_2017	Clean by inserting 'UNKNOWN' for flights_2017

						and flyu_flights	
	cancellation_reason	varchar	No	Flights_2017, Flights_2018, Flyu_flights	Flights_2017. cancellation_reason Flight_2018. cancellation_reason Flyu_flights. cancelled_reason	Unclear data in all tables	Clean by self-analyzing and inserting clear reasoning consistently and extract then insert data.
Definition:	Dim_cancelled_flights holds the data of cancelled flights along with the reason. It will help in producing reports where cancelled flights and reasons are necessary.						
Notes:							
Star schema table	Attribute name	Data Type	Key	DQ Source	Data source field (mapping)	Data quality issues	Transformation
DIM_WEATHER_DATA	weather_data_id	number	Yes	Automatically generated as surrogate key	N/A	N/A	Create a sequence and trigger to generate key and to assign them when a new row is inserted.
	weather_station	varchar	No	Weather_data.	Weather_data. weather_station	N/A	Upload the data from the table into DM without any changes
	weather_date	date	No	Weather_data.	Weather_data. weather_date	Inconsistent data	Change the varchar value to date by making the data consistent and extract then insert data.
	AWND	float	No	Weather_data.	Weather_data. AWND	N/A	Upload the data from the table into DM without any changes
	PRCP	float	No	Weather_data.	Weather_data. PRCP	N/A	Upload the data from the table into DM without any changes

	TAVG	number	No	Weather_data.	Weather_data. TAVG	N/A	Upload the data from the table into DM without any changes
Definition:	Dim_weather_data holds the weather_data of 2017 and 2018 along with the months and day which will be useful in producing reports along with weather condition of that particular date.						
Notes:	Although the data of 2017 and 2018 are uploaded to dim_weather_data, data of 2018 and flyuflights will only be used in fact as 2017 has no month and day column to relate to weather_data.						
Star schema table	Attribute name	Data Type	Key	DQ Source	Data source field (mapping)	Data quality Issues	Transformation
DIM_TIME	time_id	number	Yes	Automatically generated as surrogate key	N/A	N/A	Create a sequence and trigger to generate key and to assign them when a new row is inserted.
	month	number	No	weather_Data	Weather_data. weather_date	year, month & day stored as single value date	Extract distinct month from weather_date and insert data.
	quarter	number	No	Check month and map with seasons to insert data	N/A	N/A	Create PL/SQL to identify month and map with seasons then insert data as 1, 2, 3, 4 being based on seasons
	year	number	No	weather_data Flights_2017, Flights_2018, FlyU_flights	Weather_data. weather_date, Flights_2017.year, Flights_2018.year, FlyU_flights.flight_the_year	year, month & day stored as single value date	Extract distinct year from weather_date, flights_2017, flights_2018 and flyu_flights and insert data.
Definition:	Dim_time holds the time interval of all the data inserted into the DM which will help to produce reports on various time stamps.						
Notes:	Two columns with only year has been inserted firstly in order to match the flights to dim_time whose month and year is not provided.						

Select one of the reports you have suggested. Illustrate the expected data in the star schema to support the report.

Firstly, to identify and produce expected reports, all the data in fact and dimensions have to be inserted. Hence, the expected Fact table and Dimension tables have been presented below as a prerequisite to the expected report with data.

#### Expected: DIM\_DELAYED\_FLIGHTS

DELAYED_FLIGHT_ID	FLIGHT_NUMBER	TAIL_NUMBER	AIR_SYSTEM_DELAY	SECURITY_DELAY	AIRLINE_DELAY	LATE_AIRCRAFT_DELAY	WEATHER_DELAY	FLIGHTS_AIRLINE	DATA_SOURCE
1	1096	N3ENAA	43	0	0	0	0	UNKNOWN	flights_2017
2	2314	N3ENAA	31	0	0	34	0	UNKNOWN	flights_2017
3	1096	N3HUAA	0	0	0	0	0	UNKNOWN	flights_2017
4	84	N3KUAA	26	0	0	0	8	UNKNOWN	flights_2017
5	1653	N3ETAA	0	0	0	20	0	UNKNOWN	flights_2017
6	193	N3GFAA	9	0	7	0	0	UNKNOWN	flights_2017
7	2314	N3HXAA	0	0	0	39	0	UNKNOWN	flights_2017
8	1096	N3DCAA	34	0	0	0	0	UNKNOWN	flights_2017
9	2314	N3DCAA	0	0	0	0	0	UNKNOWN	flights_2017
10	84	N3GUAA	0	0	0	0	28	UNKNOWN	flights_2017
11	2684	N956AT	0	0	0	0	0	DL	flights_2018
12	2162	N947UW	10	0	0	5	0	US	flights_2018
13	2694	N977AT	0	0	0	0	0	DL	flights_2018
14	2150	N956UW	0	0	0	0	0	US	flights_2018
15	2696	N961AT	0	0	0	0	0	DL	flights_2018
16	2152	N967UW	21	0	0	0	5	US	flights_2018
17	2136	N965UW	0	0	0	0	0	US	flights_2018
18	2686	N892AT	56	0	0	0	0	DL	flights_2018
19	2136	N956UW	0	0	0	0	0	US	flights_2018
20	2692	N953AT	63	0	10	0	0	DL	flights_2018

#### Expected: DIM\_CANCELLED\_FLIGHTS

CANCELLED_FLIGHT_ID	FLIGHT_NUMBER	TAIL_NUMBER	FLIGHTS_AIRLINE	CANCELLATION_REASON	DATA_SOURCE
1	291	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017
2	2314	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017
3	84	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017
4	1653	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017
5	193	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017
6	1096	N3DAAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017
7	2314	N3DAAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017
8	84	N3HSAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017
9	1653	N3DMAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017

10	193	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017
11	2172	N957UW	AA	AIR SYSTEM / MECHANICAL ISSUES	flights_2018
12	291	N3DNAA	UNKNOWN	ADVERSE WEATHER CONDITION	flyu_flights
13	193	N3FMAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flyu_flights
14	1096	N3AKAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flyu_flights
15	291	N3JFAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flyu_flights
16	2314	N3AKAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flyu_flights
17	1096	N3DAAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flyu_flights
18	2314	N3DAAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flyu_flights
19	84	N3HSAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flyu_flights
20	1096	N3LHAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flyu_flights

#### Expected: DIM\_WEATHER\_DATA

WEATHER_DATA_ID	WEATHER_STATION	WEATHER_DATE	AWNND	PRCP	TAVG
1	USW00094789	1/1/2017	17.2	0	33
2	USW00094789	1/2/2017	13.7	0	39
3	USW00094789	1/3/2017	9.17	0.94	36
4	USW00094789	1/4/2017	10.1	0.47	47
5	USW00094789	1/5/2017	19.7	0	43
6	USW00094789	1/6/2017	8.5	0.07	22
7	USW00094789	1/7/2017	21.5	0	22
8	USW00094789	1/8/2017	17.2	0	14
9	USW00094789	1/9/2017	19.7	0.07	26
10	USW00094789	1/10/2017	16.6	0	22

#### Expected: DIM\_TIME

TIME_ID	MONTH	QUARTER	YEAR
1	-	-	2017
2	-	-	2018
3	1	4	2017
4	2	4	2017
5	3	1	2017
6	4	1	2017
7	5	1	2017
8	6	2	2017

9	7	2	2017
10	8	2	2017
11	9	3	2017
12	10	3	2017
13	11	3	2017
14	12	4	2017
15	1	4	2018
16	2	4	2018
17	3	1	2018
18	4	1	2018
19	5	1	2018
20	6	2	2018
21	7	2	2018
22	8	2	2018
23	9	3	2018
24	10	3	2018
25	11	3	2018
26	12	4	2018

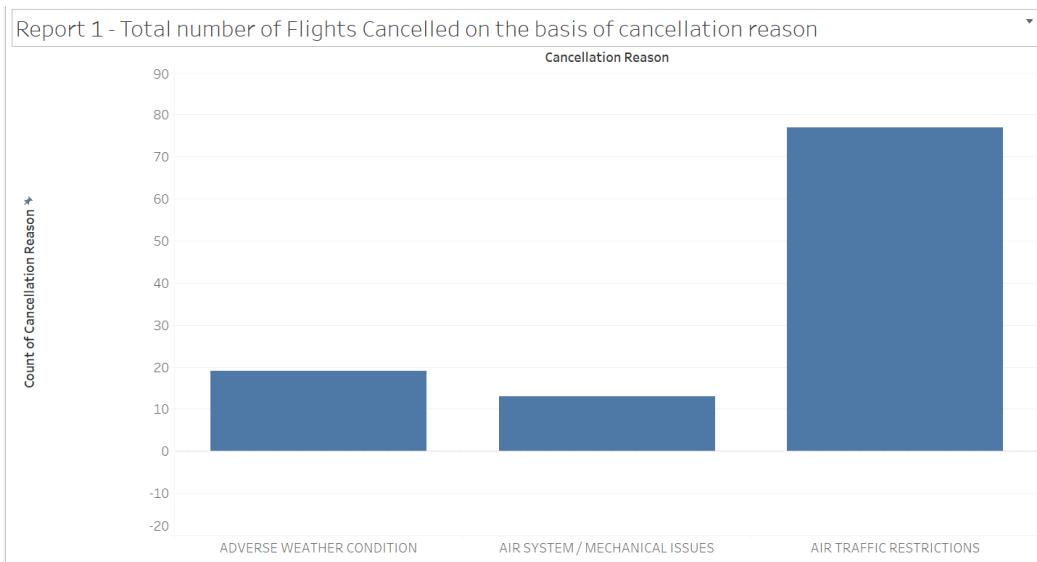
### Expected: Fact\_Flights

SURROGATE_FLIGHT_ID	FLIGHT_SERIAL_NO	CANCELLATION_FLAG	ARRIVAL_DELAY	WEATHER_DELAY	FK1_DELAYED_FLIGHT_ID	FK2_CANCELLED_FLIGHT_ID	FK3_TIMEME_ID	FK4_WEATHER_DATA_ID
1	1096	0	43	0	1	-	1	-
2	2314	0	65	0	2	-	1	-
3	1096	0	0	0	3	-	1	-
4	84	0	34	8	4	-	1	-
5	1653	0	20	0	5	-	1	-
6	193	0	16	0	6	-	1	-
7	2314	0	39	0	7	-	1	-
8	1096	0	34	0	8	-	1	-
9	2314	0	0	0	9	-	1	-
10	84	0	28	28	10	-	1	-
11	2684	0	0	0	11	-	2	10
12	2162	0	15	0	12	-	15	1
13	2694	0	0	0	13	-	17	2
14	2150	0	0	0	14	-	1	3
15	2696	0	0	0	15	-	1	4
16	2152	0	26	0	16	-	1	5
17	2136	0	0	0	17	-	1	6
18	2686	0	56	0	18	-	1	7
19	2136	0	0	0	19	-	1	8
20	2692	0	73	0	20	-	1	9
21	291	1	-	-	-	1	1	-

22	2314	1	-	-	-		2	1	-
23	84	1	-	-	-		3	1	-
24	1653	1	-	-	-		4	1	-
25	193	1	-	-	-		5	1	-
26	1096	1	-	-	-		6	1	-
27	2314	1	-	-	-		7	1	-
28	84	1	-	-	-		8	1	-
29	1653	1	-	-	-		9	1	-
30	193	1	-	-	-		10	1	-
31	2172	1	-	-	-		11	3	-
32	291	1	-	-	-		12	1	-
33	193	1	-	-	-		13	1	-
34	1096	1	-	-	-		14	1	-
35	291	1	-	-	-		15	1	-
36	2314	1	-	-	-		16	1	-
37	1096	1	-	-	-		17	1	-
38	2314	1	-	-	-		18	1	-
39	84	1	-	-	-		19	1	-
40	1096	1	-	-	-		20	1	-

Although most of the data have been inserted as it is supposed to be, some data may not be exactly similar to these however the datatype will be the same. Finally, I chose the first report to present the expected data, hence the expected report would be like:

### 1. Total number of flights cancelled on the basis of cancellation reasons.



## Task 2: Pre-processing: Gather the data sources prior to ETL

Collect only the data you require to populate the star schema into the data staging area

The first stage of data warehouse development is the data staging area. In order to first collect the data into data staging area, the original data provided has to be uploaded to the database. Hence, the excel sheets data (.xlsx) was converted to excel sheet (.csv) file. Then, these excel sheet data were uploaded to the apex oracle database via application imports. Finally, all the data from Flights\_2017, Flights\_2018, FlyU\_Flights and Weather\_Data has been uploaded into the apex oracle database.

Screenshots: Upload data of Flights\_2017 into the database.

The screenshot shows the 'Load Data' interface in Oracle Database. The 'Load To' section is set to 'New Table' with 'Table Owner' as 'WKSP\_ADSLEVEL6' and 'Table Name' as 'FLIGHTS\_2017'. The 'Configure' button is visible. Below it, under 'Settings', 'Column Headers' and 'First line contains headers' are checked. The preview section shows the first 200 rows of the CSV file, with a green checkmark indicating success. The message 'Table FLIGHTS\_2017 created with 3,503 rows!' is displayed. A 'View Table' button is at the bottom right.

1	YEAR	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE	DEPARTURE_TIME	DEPARTURE_DELAY	TAXI_OUT	
2	1256	2017	84	NSAXAA	JFK	BOS	730	734	4	17
3	6035	2017	195	NSDNAA	JFK	BOS	1255	1306	11	16
4	8723	2017	1096	NSAPAA	JFK	BOS	1545	1624	39	20

**FLIGHTS\_2017**

Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries													
	Add Column	Modify Column	Rename Column	Drop Column	Rename	Copy	Drop	Truncate	Create Lookup Table	Create App															
Column Name																									Primary Key
ID																									1
COLUMN_																									-
YEAR																									-
FLIGHT_NUMBER																									-
TAIL_NUMBER																									-
ORIGIN_AIRPORT																									-
DESTINATION_AIRPORT																									-
SCHEDULED_DEPARTURE																									-
DEPARTURE_TIME																									-
DEPARTURE_DELAY																									-
TAXI_OUT																									-
WHEELS_OFF																									-
SCHEDULED_TIME																									-
ELAPSED_TIME																									-
AIR_TIME																									-
DISTANCE																									-
WHEELS_ON																									-
TAKE_IN																									-
SCHEDULED_ARRIVAL																									-
ARRIVAL_TIME																									-
ARRIVAL_DELAY																									-
DIVERTED																									-
CANCELLED																									-
CANCELLATION_REASON																									-
AIR_SYSTEM_DELAY																									-
SECURITY_DELAY																									-
AIRLINE_DELAY																									-
LATE_AIRCRAFT_DELAY																									-
WEATHER_DELAY																									-

**FLIGHTS\_2017**

Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries													
	Query	Count Rows	Insert Row	Load Data																					
EDIT	ID	COLUMN_	YEAR	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE	DEPARTURE_TIME	DEPARTURE_DELAY	TAXI_OUT	WHEELS_OFF	SCHEDULED_TIME	ELAPSED_TIME	AIR_TIME	DISTANCE	WHEELS_ON								
156	406321	2017	291	-	JFK	AUS	1700	-	-	-	-	265	-	-	-	1521	-								
157	409332	2017	2314	-	JFK	BOS	2025	-	-	-	-	70	-	-	-	187	-								
158	412353	2017	84	-	JFK	BOS	730	-	-	-	-	71	-	-	-	187	-								
159	413163	2017	1653	-	JFK	CLT	820	-	-	-	-	132	-	-	-	541	-								
160	417397	2017	193	-	JFK	BOS	1240	-	-	-	-	63	-	-	-	187	-								
161	420674	2017	1096	NBENAA	JFK	BOS	1600	1634	54	28	1702	75	84	50	187	1752									
162	421620	2017	291	NBFAA	JFK	AUS	1700	1657	-3	17	1714	265	223	201	1521	1935									
163	424764	2017	2314	NBENAA	JFK	BOS	2025	2126	61	24	2150	70	74	45	187	2235									
164	427912	2017	84	NBICAA	JFK	BOS	730	725	-5	26	751	71	79	47	187	838									
165	428749	2017	1653	NBENAA	JFK	CLT	820	817	-3	23	840	132	117	85	541	1005									
166	433103	2017	193	NBKSAA	JFK	BOS	1240	1236	-4	32	1308	63	79	45	187	1351									
167	436485	2017	1096	NBHJAA	JFK	BOS	1600	1620	20	17	1637	75	60	35	187	1712									
168	437451	2017	291	NBKMAA	JFK	AUS	1700	1659	-1	19	1718	265	234	212	1521	1950									
169	440729	2017	2314	NBHJAA	JFK	BOS	2025	2025	0	29	2054	70	73	40	187	2154									
170	443746	2017	84	NBKJAA	JFK	BOS	730	738	8	46	824	71	97	43	187	907									

Download

Screenshots: Upload data of Flights\_2018 into the database.

The screenshot shows the 'Load Data' dialog box. At the top, it displays the file 'Flights\_2018\_2000\_rows.csv'. Below this, under 'Load To', the 'New Table' tab is selected. The 'Table Owner' is set to 'WKSP\_ADSLEVEL6' and the 'Table Name' is 'FLIGHTS\_2018'. A note says 'Please select the columns to load.' Below this, 'Primary Keys' are listed as 'SYS\_GUID' and 'Identity Column'. A checked checkbox 'Use Column Data Types' is present. In the 'Settings' section, 'Column Headers' are checked and 'First line contains headers' is selected. The preview section shows the first 200 rows of the CSV file, including columns: 1, YEAR, MONTH, DAY, DAY\_OF\_WEEK, AIRLINE, FLIGHT\_NUMBER, TAIL\_NUMBER, ORIGIN\_AIRPORT, and DESTINATION\_AIRPORT. Three sample rows are shown:

1	YEAR	MONTH	DAY	DAY_OF_WEEK	AIRLINE	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTINATION_AIRPORT
2	2018	1	2	5	EV	5038	N608QK	JFK	BHM
3	2018	1	5	1	DL	2666	N603AT	JFK	BOS

At the bottom of the preview section are 'Cancel' and 'Load Data' buttons. The 'Load Data' button is highlighted with a green checkmark.

**Table FLIGHTS\_2018 created with 1,488 rows!**

**View Table**

FLIGHTS_2018					
Column Name	Data Type	Nullable	Default		Primary Key
ID	NUMBER	No	'WISPAIRLEVELA'.'ISQ203_0477749'.'newfl'		1
COLUMN_	NUMBER	Yes	-		-
YEAR	NUMBER	Yes	-		-
MONTH	NUMBER	Yes	-		-
DAY	NUMBER	Yes	-		-
DAY_OF_WEEK	NUMBER	Yes	-		-
AIRLINE	VARCHAR(50)	Yes	-		-
FLIGHT_NUMBER	NUMBER	Yes	-		-
TAIL_NUMBER	VARCHAR(50)	Yes	-		-
ORIGIN_AIRPORT	VARCHAR(50)	Yes	-		-
DESTINATION_AIRPORT	VARCHAR(50)	Yes	-		-
SCHEDULED_DEPARTURE	NUMBER	Yes	-		-
DEPARTURE_TIME	NUMBER	Yes	-		-
DEPARTURE_DELAY	NUMBER	Yes	-		-
TAXI_OUT	NUMBER	Yes	-		-
WHEELS_OFF	NUMBER	Yes	-		-
SCHEDULED_TIME	NUMBER	Yes	-		-
ELAPSED_TIME	NUMBER	Yes	-		-
AIR_TIME	NUMBER	Yes	-		-
DISTANCE	NUMBER	Yes	-		-
WHEELS_ON	NUMBER	Yes	-		-
TAXI_IN	NUMBER	Yes	-		-
SCHEDULED_ARRIVAL	NUMBER	Yes	-		-
ARRIVAL_TIME	NUMBER	Yes	-		-
ARRIVAL_DELAY	NUMBER	Yes	-		-
DIVERTED	NUMBER	Yes	-		-
CANCELLED	NUMBER	Yes	-		-
CANCELLATION_REASON	VARCHAR(4000)	Yes	-		-
AIR_SYSTEM_DELAY	NUMBER	Yes	-		-
SECURITY_DELAY	NUMBER	Yes	-		-
AIRLINE_DELAY	NUMBER	Yes	-		-
LATE_AIRCRAFT_DELAY	NUMBER	Yes	-		-

FLIGHTS_2018					
Column Name	Data Type	Nullable	Default		Primary Key
ID	NUMBER	No	'WISPAIRLEVELA'.'ISQ203_0477749'.'newfl'		1
COLUMN_	NUMBER	Yes	-		-
YEAR	NUMBER	Yes	-		-
MONTH	NUMBER	Yes	-		-
DAY	NUMBER	Yes	-		-
DAY_OF_WEEK	NUMBER	Yes	-		-
AIRLINE	VARCHAR(50)	Yes	-		-
FLIGHT_NUMBER	NUMBER	Yes	-		-
TAIL_NUMBER	VARCHAR(50)	Yes	-		-
ORIGIN_AIRPORT	VARCHAR(50)	Yes	-		-
DESTINATION_AIRPORT	VARCHAR(50)	Yes	-		-
SCHEDULED_DEPARTURE	NUMBER	Yes	-		-
DEPARTURE_TIME	NUMBER	Yes	-		-
DEPARTURE_DELAY	NUMBER	Yes	-		-
TAXI_OUT	NUMBER	Yes	-		-
WHEELS_OFF	NUMBER	Yes	-		-
SCHEDULED_TIME	NUMBER	Yes	-		-
ELAPSED_TIME	NUMBER	Yes	-		-
AIR_TIME	NUMBER	Yes	-		-
DISTANCE	NUMBER	Yes	-		-
WHEELS_ON	NUMBER	Yes	-		-
TAXI_IN	NUMBER	Yes	-		-
SCHEDULED_ARRIVAL	NUMBER	Yes	-		-
ARRIVAL_TIME	NUMBER	Yes	-		-
ARRIVAL_DELAY	NUMBER	Yes	-		-
DIVERTED	NUMBER	Yes	-		-
CANCELLED	NUMBER	Yes	-		-
CANCELLATION_REASON	VARCHAR(4000)	Yes	-		-
AIR_SYSTEM_DELAY	NUMBER	Yes	-		-
SECURITY_DELAY	NUMBER	Yes	-		-
AIRLINE_DELAY	NUMBER	Yes	-		-
LATE_AIRCRAFT_DELAY	NUMBER	Yes	-		-

## Screenshots: Upload data of Weather\_Data into the database.

The screenshot shows the 'Load Data' dialog box. In the 'Load To' section, 'New Table' is selected, 'Table Owner' is set to 'WKSP\_ADSLLEVEL6', and the 'Table Name' is 'WEATHER\_DATA'. Under 'Primary Keys', 'SYS\_GUID' is chosen. A checked checkbox 'Use Column Data Types' is present. In the 'Settings' section, 'Column Headers' is checked, and the 'First line contains headers' option is selected. The 'Column Delimiter' is a comma (,), 'Enclosed By' is none, and 'File Encoding' is Western European ISO-8859-1. The 'Preview' section shows the first two rows of the CSV file: Row 1: STATION, NAME, DATE, AWND, PRCP, TAVG; Row 2: USW00094789, NEW YORK METROPOLITAN AREA, NY US, 1/1/2017, 17.22, 0, 33. Below the preview, there are 'Cancel' and 'Load Data' buttons. A large green checkmark is displayed above the message 'Table WEATHER\_DATA created with 730 rows!'. At the bottom right, there is a 'View Table' button.

**WEATHER\_DATA**

Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries	
Add Column	Modify Column	Rename Column	Drop Column	Rename	Copy	Drop	Truncate	Create Lookup Table	Create App				
Column Name	Data Type	Nullable	Default										Primary Key
ID	NUMBER	No	"WKSP_ADSLEVEL6"."SEQ\$\$_104978005".nextval										1
STATION	VARCHAR2(50)	Yes	-										-
NAME	VARCHAR2(255)	Yes	-										-
DATE_	VARCHAR2(50)	Yes	-										-
AWND	NUMBER	Yes	-										-
PRCP	NUMBER	Yes	-										-
TAVG	NUMBER	Yes	-										-

**WEATHER\_DATA**

EDIT	ID	STATION	NAME	DATE_	AWND	PRCP	TAVG
1	USW00094789	NEW YORK METROPOLITAN AREA, NY US		1/1/2017	17.22	0	33
2	USW00094789	NEW YORK METROPOLITAN AREA, NY US		2/1/2017	13.65	0	39
3	USW00094789	NEW YORK METROPOLITAN AREA, NY US		3/1/2017	9.17	.94	36
4	USW00094789	NEW YORK METROPOLITAN AREA, NY US		4/1/2017	10.07	.47	47
5	USW00094789	NEW YORK METROPOLITAN AREA, NY US		5/1/2017	19.69	0	43
6	USW00094789	NEW YORK METROPOLITAN AREA, NY US		6/1/2017	8.5	.07	22
7	USW00094789	NEW YORK METROPOLITAN AREA, NY US		7/1/2017	21.47	0	22
8	USW00094789	NEW YORK METROPOLITAN AREA, NY US		8/1/2017	17.22	0	14
9	USW00094789	NEW YORK METROPOLITAN AREA, NY US		9/1/2017	19.69	.07	26
10	USW00094789	NEW YORK METROPOLITAN AREA, NY US		10/1/2017	16.55	0	22
11	USW00094789	NEW YORK METROPOLITAN AREA, NY US		11/1/2017	10.98	0	26
12	USW00094789	NEW YORK METROPOLITAN AREA, NY US		12/1/2017	6.04	.7	36
13	USW00094789	NEW YORK METROPOLITAN AREA, NY US		13/01/2017	15.21	0	32
14	USW00094789	NEW YORK METROPOLITAN AREA, NY US		14/01/2017	11.63	0	24
15	USW00094789	NEW YORK METROPOLITAN AREA, NY US		15/01/2017	9.84	0	31

**Object Browser**

Tables Schema: WKSP\_ADSLEVEL6

Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries	
FLY_FLIGHTS	Query	Count Rows	Insert Row	Load Data									
FLY_FLIGHTS	EDIT	FLIGHT_THE_YEAR	THE_MONTH	THE_DAY	D_O_W	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTINATION_AIRPORT	ARRIVAL_DELAY	DIVERTED	CANCELLED	CANCELLED_REASON
FLY_FLIGHTS	1	2017	1	3	6	291	N8GYAA	JFK	AUS	883	0	0	-
FLY_FLIGHTS	2	2017	1	3	6	2314	N8LCAA	JFK	BOS	128	0	0	-
FLY_FLIGHTS	3	2017	1	4	7	1096	N8DJAA	JFK	BOS	104	0	0	-
FLY_FLIGHTS	4	2017	1	4	7	291	N8KHAA	JFK	AUS	107	0	0	-
FLY_FLIGHTS	5	2017	1	4	7	2314	N8DJAA	JFK	BOS	114	0	0	-
FLY_FLIGHTS	6	2017	1	6	2	2314	N87YAA	JFK	BOS	144	0	0	-
FLY_FLIGHTS	7	2017	1	7	3	291	N8GPAA	JFK	AUS	119	0	0	-
FLY_FLIGHTS	8	2017	2	3	2	1096	N8CMAA	JFK	BOS	158	0	0	-
FLY_FLIGHTS	9	2017	2	3	2	2314	N8CMAA	JFK	BOS	157	0	0	-
FLY_FLIGHTS	10	2017	2	5	4	2314	N8KFAA	JFK	BOS	105	0	0	-
FLY_FLIGHTS	11	2017	2	8	7	1096	N8AAAA	JFK	BOS	185	0	0	-
FLY_FLIGHTS	12	2017	2	10	2	1096	N8ADAA	JFK	BOS	102	0	0	-
FLY_FLIGHTS	13	2017	2	16	1	193	N8CRAA	JFK	BOS	159	0	0	-
FLY_FLIGHTS	14	2017	2	16	1	1096	N8EKA	JFK	BOS	111	0	0	-
FLY_FLIGHTS	15	2017	2	16	1	2314	N8JLAA	JFK	BOS	152	0	0	-

Although the data of SQL file which contains customer and complaint are not required to produce reports identified in task 1, they have been uploaded to apex oracle because they may be used for other references and evidence which may be necessary in other parts of data analysis.

## Screenshot: SQL Successfully uploaded and tables created.

The screenshot shows two pages from Oracle Application Express.

**Top Page:** A confirmation dialog titled "Run Script" asking for confirmation to run the script. It displays the following information:

Script Name	FlyUflightsData
Created	on 12/09/2020 11:20:39 AM by AV.SAPKOTA999@GMAIL.COM
Updated	on 12/09/2020 11:20:39 AM by AV.SAPKOTA999@GMAIL.COM
Number of Statements	245
Script Size in Bytes	26,294

Buttons: "Cancel" and "Run Now".

**Bottom Page:** A detailed log of the executed SQL statements. The table has columns: Number, Elapsed, Statement, Feedback, and Rows.

Number	Elapsed	Statement	Feedback	Rows
1	0.01	DROP TABLE FlyU.flights CASCADE CONSTRAINTS	ORA-00942: table or view does not exist ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_200200", line 626 ORA-06512: at "SYS.DBMS_SQL", line 1658 ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_200200", line 612 ORA-06512: at "APEX_200200.WWV_FLOW_DYNAMIC_EXEC", line 1748	-
2	0.00	DROP TABLE customer CASCADE CONSTRAINTS	ORA-00942: table or view does not exist ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_200200", line 626 ORA-06512: at "SYS.DBMS_SQL", line 1658 ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_200200", line 612 ORA-06512: at "APEX_200200.WWV_FLOW_DYNAMIC_EXEC", line 1748	-
3	0.00	DROP TABLE complaint CASCADE CONSTRAINTS	ORA-00942: table or view does not exist ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_200200", line 626 ORA-06512: at "SYS.DBMS_SQL", line 1658 ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_200200", line 612 ORA-06512: at "APEX_200200.WWV_FLOW_DYNAMIC_EXEC", line 1748	-
4	0.07	CREATE TABLE customer (customer_id INTEGER NOT NULL, cust_name VARCHAR(50))	Table created.	0
5	0.07	CREATE TABLE complaint (complaint_id INTEGER NOT NULL PRIMARY KEY, customer_id INTEGER NOT NULL, complaint_desc VARCHAR(255), complaint_date DATE)	Table created.	0
6	0.08	CREATE TABLE FlyU.flights ( flight_no NUMBER(4), flight_date DATE, destination VARCHAR(50), departure_time TIME, arrival_time TIME, duration NUMBER(3,1), fare NUMBER(10,2), airline VARCHAR(50))	Table created.	0
7	0.06	ALTER TABLE FlyU.flights ADD CONSTRAINT pk_flights PRIMARY	Table altered.	0
8	0.05	ALTER TABLE complaint ADD CONSTRAINT fk_complaint_to_customer FOREIGN KEY (customer_id) REFERENCES customer (customer_id)	Table altered.	0
9	0.05	ALTER TABLE flight ADD CONSTRAINT fk_flight_to_flyt FOREIGN KEY (flight_no) REFERENCES flights (flight_no)	Table altered.	0
10	0.06	INSERT INTO Customer VALUES (10, 'NY10', 'BUSINESS', 'IBM')	1 row(s) inserted.	1
11	0.00	INSERT INTO Customer VALUES (101, 'NY101', 'BUSINESS', 'Google')	1 row(s) inserted.	1
12	0.01	INSERT INTO Customer VALUES (102, 'NY102', 'BUSINESS', 'Amazon')	1 row(s) inserted.	1
13	0.00	INSERT INTO Customer VALUES (103, 'NY103', 'BUSINESS', 'Facebook')	1 row(s) inserted.	1
14	0.01	INSERT INTO Customer VALUES (104, 'NY104', 'BUSINESS', 'HP')	1 row(s) inserted.	1
15	0.00	INSERT INTO Customer VALUES (105, 'NY105', 'BUSINESS', 'SkyB')	1 row(s) inserted.	1

Summary statistics at the bottom:

245	242	3
Statements Processed	Successful	With Errors

Three errors have been identified in the first three statements because table has been dropped before it is created. This error is supposed to be solved when the script is run again. Hence, it can be safely discarded and can be counted as a successful upload.

**CUSTOMER**

Customer Data:

CUSTOMER_ID	CUSTOMER_ZIP_CODE	CUSTOMER_TYPE	BUSINESS	CUSTOMER_MILES
10	NY10	BUSINESS	IBM	10000
101	NY101	BUSINESS	Google	20000
102	NY102	BUSINESS	Amazon	50000
103	NY103	BUSINESS	Facebook	110000
104	NY104	BUSINESS	HP	10009
105	NY105	BUSINESS	SkyBet	19000
106	NY106	BUSINESS	Hermes	1000
107	NY107	BUSINESS	Apple	110000
108	NY108	BUSINESS	LCC	700
109	NY109	BUSINESS	LB	0
100	NY100	BUSINESS	IBM	10000

**COMPLAINT**

Complaint Data:

COMPLAINT_ID	FLIGHT_ID_NO	TAIL_NUMBER	THE_YEAR	THE_MONTH	THE_DAY	COMPLAINT_TYPE	DESCRIPTION	COMPLAINT_STATUS	ALLOCATED_TO	COMPENSATION_AMNT	COMPENSATION_TYPE	FK1_CUSTOMER_ID
1	291	NBGYAA	2017	1	3	-	late	open	AA	0	0	10
2	2314	NSLCAA	2017	1	3	-	late	open	AA	0	0	101
3	1096	NDJAA	2017	1	4	-	late	open	BB	0	0	103
4	291	NRKHAA	2017	1	4	-	late	open	BB	0	0	105
5	2314	NDJUAA	2017	1	4	-	late	open	CC	0	0	101
6	2314	NBYYAA	2017	1	6	-	late	open	AA	0	0	108
7	291	NGPAAA	2017	1	7	-	late	open	AA	0	0	107
8	1096	NBCMAA	2017	2	3	-	late	open	BB	0	0	10
9	2314	NBCMAA	2017	2	3	-	late	open	BB	0	0	106
10	2314	NRKFAA	2017	2	5	-	late	open	CC	0	0	101
11	1096	NSAAAA	2017	2	8	-	late	open	AA	0	0	102
12	1096	NSADAA	2017	2	10	-	late	open	AA	0	0	104
13	193	NBCRAA	2017	2	16	-	late	open	BB	0	0	107
14	1096	NBEKAA	2017	2	16	-	late	open	BB	0	0	108
15	2314	NBLJAA	2017	2	16	-	late	open	CC	0	0	109

## Star Schema Set Up (DM environment)

The star schema in task 1 was designed using QSEE which will also help in generating database script in SQL. The generated script was uploaded to the apex oracle database.

### Screenshot: Generated Script Run Report

The screenshot shows two pages from the Oracle Application Express (APEX) interface.

**Run Script Page:**

- Header: APEX, App Builder, SQL Workshop, Team Development, App Gallery.
- Breadcrumb: SQL Scripts > Run Script.
- User: Avhimantu Sapkota (av.sapkota999@gmail.com).
- Schema: WKSP\_ADSLEVEL0.
- Section: Run Script.
- Message: You have requested to run the following script. Please confirm your request.
- Table: Script details.

Script Name	DM_TableCreation
Created	on 12/09/2020 07:54:39 AM by AV.SAPKOTA999@GMAIL.COM
Updated	on 12/09/2020 07:54:39 AM by AV.SAPKOTA999@GMAIL.COM
Number of Statements	9
Script Size in Bytes	6,376

- Buttons: Cancel, Run Now.

**Summary Report Page:**

- Header: Script: DM\_TableCreation, Status: Complete.
- View: Summary.
- Rows: 15.
- Buttons: Create App, Edit Script.
- Table: Statement execution details.

Number	Elapsed	Statement	Feedback	Rows
1	0.14	CREATE TABLE FACT_FLIGHTS( surrogate_flight_id INTEGER NOT NULL, flight_id INTEGER);	Table created.	0
2	0.09	CREATE TABLE DIM_DELAYED_FLIGHTS( delayed_flight_id INTEGER);	Table created.	0
3	0.08	CREATE TABLE DIM_CANCELLED_FLIGHTS( cancelled_flight_id INT);	Table created.	0
4	0.07	CREATE TABLE DIM_TIME( time_id INTEGER NOT NULL, month INT);	Table created.	0
5	0.07	CREATE TABLE DIM_WEATHER_DATA( weather_data_id INTEGER NOT NULL);	Table created.	0
6	0.01	ALTER TABLE FACT_FLIGHTS ADD CONSTRAINT fk1_FACT_FLIGHTS_to_DIM_DELAYED_FLIGHTS;	ORA-00905: missing keyword ORA-06512 at "SYS.WWV_DBMS_SQL_APEX_200200", line 626 ORA-06512 at "SYS.DBMS_SYS_SQL", line 1658 ORA-06512 at "SYS.WWV_DBMS_SQL_APEX_200200", line 612 ORA-06512 at "APEX_20200.WWV_FLOW_DYNAMIC_EXEC", line 1749	-
7	0.00	ALTER TABLE FACT_FLIGHTS ADD CONSTRAINT fk2_FACT_FLIGHTS_to_DIM_CANCELLED_FLIGHTS;	ORA-00905: missing keyword ORA-06512 at "SYS.WWV_DBMS_SQL_APEX_200200", line 626 ORA-06512 at "SYS.DBMS_SYS_SQL", line 1658 ORA-06512 at "SYS.WWV_DBMS_SQL_APEX_200200", line 612 ORA-06512 at "APEX_20200.WWV_FLOW_DYNAMIC_EXEC", line 1749	-
8	0.00	ALTER TABLE FACT_FLIGHTS ADD CONSTRAINT fk3_FACT_FLIGHTS_to_DIM_TIME;	ORA-00905: missing keyword ORA-06512 at "SYS.WWV_DBMS_SQL_APEX_200200", line 626 ORA-06512 at "SYS.DBMS_SYS_SQL", line 1658 ORA-06512 at "SYS.WWV_DBMS_SQL_APEX_200200", line 612 ORA-06512 at "APEX_20200.WWV_FLOW_DYNAMIC_EXEC", line 1749	-
9	0.01	ALTER TABLE FACT_FLIGHTS ADD CONSTRAINT fk4_FACT_FLIGHTS_to_DIM_WEATHER_DATA;	ORA-00905: missing keyword ORA-06512 at "SYS.WWV_DBMS_SQL_APEX_200200", line 626 ORA-06512 at "SYS.DBMS_SYS_SQL", line 1658 ORA-06512 at "SYS.WWV_DBMS_SQL_APEX_200200", line 612 ORA-06512 at "APEX_20200.WWV_FLOW_DYNAMIC_EXEC", line 1749	-

- Summary: 9 statements processed, 5 successful, 4 with errors.
- Page footer: Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved. Application Express 20.2.0.0.20.

## Screenshot of script where errors were generated:

```

105 -- Now all the tables have been created the ALTER TABLE command is used to define some additional
106 -- constraints. These typically constrain values of foreign keys to be generated in some way
107 -- with the primary keys of related tables. Foreign key constraints can actually be specified
108 -- when each table is created, but doing so can lead to dependency problems within the script
109 -- i.e. tables may be referenced before they have been created. This method is therefore safer.
110
111 -- Alter table to add new constraints required to implement the "supports" relationship
112
113 -- This constraint ensures that the Foreign key of table "FACT_FLIGHTS"
114 -- correctly references the primary key of table "DIM_DELAYED_FLIGHTS"
115
116 ALTER TABLE FACT_FLIGHTS ADD CONSTRAINT fk1_FACT_FLIGHTS_to_DIM_DELAYED_FLIGHTS FOREIGN KEY(fk1_delayed_flight_id) REFERENCES DIM_DELAYED_FLIGHTS(delayed_flight_id) ON DELETE RESTRICT ON UPDATE RESTRICT;
117
118 -- Alter table to add new constraints required to implement the "supports" relationship
119
120 -- This constraint ensures that the Foreign key of table "FACT_FLIGHTS"
121 -- correctly references the primary key of table "DIM_CANCELLED_FLIGHTS"
122
123 ALTER TABLE FACT_FLIGHTS ADD CONSTRAINT fk2_FACT_FLIGHTS_to_DIM_CANCELLED_FLIGHTS FOREIGN KEY(fk2_cancelled_flight_id) REFERENCES DIM_CANCELLED_FLIGHTS(cancelled_flight_id) ON DELETE RESTRICT ON UPDATE RESTRICT;
124
125 -- Alter table to add new constraints required to implement the "supports" relationship
126
127 -- This constraint ensures that the Foreign key of table "FACT_FLIGHTS"
128 -- correctly references the primary key of table "DIM_TIME"
129
130 ALTER TABLE FACT_FLIGHTS ADD CONSTRAINT fk3_FACT_FLIGHTS_to_DIM_TIME FOREIGN KEY(fk3_time_id) REFERENCES DIM_TIME(time_id) ON DELETE RESTRICT ON UPDATE RESTRICT;
131
132 -- Alter table to add new constraints required to implement the "supports" relationship
133
134 -- This constraint ensures that the Foreign key of table "FACT_FLIGHTS"
135 -- correctly references the primary key of table "DIM_WEATHER_DATA"
136
137 ALTER TABLE FACT_FLIGHTS ADD CONSTRAINT fk4_FACT_FLIGHTS_to_DIM_WEATHER_DATA FOREIGN KEY(fk4_weather_data_id) REFERENCES DIM_WEATHER_DATA(weather_data_id) ON DELETE RESTRICT ON UPDATE RESTRICT;
138
139
140 ----- End of DDL file auto-generation
141
142 ----- End of DDL file auto-generation

```

The screenshot shows a SQL script in Oracle SQL Developer. Four specific lines of code are highlighted with red boxes and error markers, indicating syntax errors. The script is intended to generate DDL for tables and constraints, but contains mistakes such as missing closing parentheses or commas.

The auto-generated script had errors hence four errors were encountered. Thus, the code highlighted in the screenshot above were removed and drop table script was added in the same script for convenient table generation, every time the script is run.

The final script without any errors:

N.B. You can also find these codes in a separate file: 'Task 2 – DDL Script (Auto-generated from Star Schema Design)' in the attached folder: 'SQL Scripts'

---

```

-- Database table removal script
-- Auto-Generated by QSEE-SuperLite (c) 2001-2004 QSEE-Technologies Ltd.
-- note: spaces within table names have been replaced by underscores (_)
-- Target DB: SQL2
-- Entity Model: Entity Relationship Diagram
-- Use this script to drop the tables created by -
--   'C:\Users\AV.FUNKY-
YOGA.000\Documents\UnderGrad Documents\Level 6\Block 1\Advanced Database Systems\Assignment\Task 2\FLYU_DM
_Table1.sql'
-- Dropping a table has the effect of removing it from the Database. All existing data is lost
-- as well as the definition of the table itself (stored as meta-data). Dropping the tables allows
-- them to be re-created within the Database, even if they already exist.

-- Drop tables --
DROP TABLE FACT_FLIGHTS;
DROP TABLE DIM_DELAYED_FLIGHTS;
DROP TABLE DIM_CANCELLED_FLIGHTS;
DROP TABLE DIM_TIME;
DROP TABLE DIM_WEATHER_DATA;

-- End of DDL DROP file auto-generation

```

```

-----  

-- Database creation Script  

-- Auto-Generated by QSEE-SuperLite (c) 2001-2004 QSEE-Technologies Ltd.  

-- Verbose generation: ON  

-- note: spaces within table/column names have been replaced by underscores (_)  

-- Target DB: SQL2  

-- Entity Model: Entity Relationship Diagram  

-- To drop the tables generated by this script run -  

--   'C:\Users\AV.FUNKY-  

YOGA.000\Documents\UnderGrad Documents\Level 6\Block 1\Advanced Database Systems\Assignment\Task 2\FLYU_DM  

_Tables1_drop.sql'  

-----  

-----  

-- Table Creation --  

-- Each entity on the model is represented by a table that needs to be created within the Database.  

-- Within SQL new tables are created using the CREATE TABLE command.  

-- When a table is created its name and its attributes are defined.  

-- The values of which are derived from those specified on the model.  

-- Certain constraints are sometimes also specified, such as identification of primary keys.  

-- Create a Database table to represent the "FACT_FLIGHTS" entity.  

CREATE TABLE FACT_FLIGHTS(  

    surrogate_flight_id INTEGER NOT NULL,  

    flight_serial_no    INTEGER NOT NULL,  

    cancellation_flag  INTEGER NOT NULL,  

    arrival_delay      INTEGER NOT NULL,  

    weather_delay      INTEGER NOT NULL,  

    fk1_delayed_flight_id  INTEGER,  

    fk2_cancelled_flight_id INTEGER,  

    fk3_time_id        INTEGER NOT NULL,  

    fk4_weather_data_id INTEGER,  

    -- Specify the PRIMARY KEY constraint for table "FACT_FLIGHTS".  

    -- This indicates which attribute(s) uniquely identify each row of data.  

    CONSTRAINT pk_FACT_FLIGHTS PRIMARY KEY (surrogate_flight_id)  

);  

-- Create a Database table to represent the "DIM_DELAYED_FLIGHTS" entity.  

CREATE TABLE DIM_DELAYED_FLIGHTS(  

    delayed_flight_id  INTEGER NOT NULL,  

    flight_number      INTEGER NOT NULL,  

    tail_number VARCHAR(50) NOT NULL,  

    air_system_delay   INTEGER,  

    security_delay     INTEGER,

```

```

airline_delay    INTEGER,
late_aircraft_delay  INTEGER,
weather_delay    INTEGER,
flights_airline VARCHAR(20) NOT NULL,
Data_Source VARCHAR(20) NOT NULL,
-- Specify the PRIMARY KEY constraint for table "DIM_DELAYED_FLIGHTS".
-- This indicates which attribute(s) uniquely identify each row of data.
CONSTRAINT pk_DIM_DELAYED_FLIGHTS PRIMARY KEY (delayed_flight_id)
);

-- Create a Database table to represent the "DIM_CANCELLED_FLIGHTS" entity.
CREATE TABLE DIM_CANCELLED_FLIGHTS(
cancelled_flight_id INTEGER NOT NULL,
flight_number    INTEGER NOT NULL,
tail_number VARCHAR(50) NOT NULL,
flights_airline VARCHAR(20) NOT NULL,
cancellation_reason VARCHAR(120) NOT NULL,
Data_Source VARCHAR(20) NOT NULL,
-- Specify the PRIMARY KEY constraint for table "DIM_CANCELLED_FLIGHTS".
-- This indicates which attribute(s) uniquely identify each row of data.
CONSTRAINT pk_DIM_CANCELLED_FLIGHTS PRIMARY KEY (cancelled_flight_id)
);

-- Create a Database table to represent the "DIM_TIME" entity.
CREATE TABLE DIM_TIME(
time_id INTEGER NOT NULL,
month    INTEGER,
quarter  INTEGER,
year     INTEGER NOT NULL,
-- Specify the PRIMARY KEY constraint for table "DIM_TIME".
-- This indicates which attribute(s) uniquely identify each row of data.
CONSTRAINT pk_DIM_TIME PRIMARY KEY (time_id)
);

-- Create a Database table to represent the "DIM_WEATHER_DATA" entity.
CREATE TABLE DIM_WEATHER_DATA(
weather_data_id INTEGER NOT NULL,
weather_station VARCHAR(20) NOT NULL,
weather_date    DATE NOT NULL,
AWND    FLOAT(8) NOT NULL,
PRCP    FLOAT(8) NOT NULL,
TAVG    INTEGER NOT NULL,
-- Specify the PRIMARY KEY constraint for table "DIM_WEATHER_DATA".
-- This indicates which attribute(s) uniquely identify each row of data.
CONSTRAINT pk_DIM_WEATHER_DATA PRIMARY KEY (weather_data_id)
);

```

```
);
```

```
-- Alter Tables to add fk constraints --
-- Now all the tables have been created the ALTER TABLE command is used to define some additional
-- constraints. These typically constrain values of foreign keys to be associated in some way
-- with the primary keys of related tables. Foreign key constraints can actually be specified
-- when each table is created, but doing so can lead to dependency problems within the script
-- i.e. tables may be referenced before they have been created. This method is therefore safer.
-- Alter table to add new constraints required to implement the "supports" relationship
-- This constraint ensures that the foreign key of table "FACT_FLIGHTS"
-- correctly references the primary key of table "DIM_DELAYED_FLIGHTS"
ALTER TABLE FACT_FLIGHTS ADD CONSTRAINT fk1_FACT_FLIGHTS_to_DIM_DELAYED_FLIGHTS FOREIGN KEY(fk1_delayed_flight_id) REFERENCES DIM_DELAYED_FLIGHTS(delayed_flight_id);

-- Alter table to add new constraints required to implement the "supports" relationship
-- This constraint ensures that the foreign key of table "FACT_FLIGHTS"
-- correctly references the primary key of table "DIM_CANCELLED_FLIGHTS"
ALTER TABLE FACT_FLIGHTS ADD CONSTRAINT fk2_FACT_FLIGHTS_to_DIM_CANCELLED_FLIGHTS FOREIGN KEY(fk2_cancelled_flight_id) REFERENCES DIM_CANCELLED_FLIGHTS(cancelled_flight_id);

-- Alter table to add new constraints required to implement the "supports" relationship
-- This constraint ensures that the foreign key of table "FACT_FLIGHTS"
-- correctly references the primary key of table "DIM_TIME"
ALTER TABLE FACT_FLIGHTS ADD CONSTRAINT fk3_FACT_FLIGHTS_to_DIM_TIME FOREIGN KEY(fk3_time_id) REFERENCES DIM_TIME(time_id);

-- Alter table to add new constraints required to implement the "supports" relationship
-- This constraint ensures that the foreign key of table "FACT_FLIGHTS"
-- correctly references the primary key of table "DIM_WEATHER_DATA"
ALTER TABLE FACT_FLIGHTS ADD CONSTRAINT fk4_FACT_FLIGHTS_to_DIM_WEATHER_DATA FOREIGN KEY(fk4_weather_data_id) REFERENCES DIM_WEATHER_DATA(weather_data_id);
-----  
-- End of DDL file auto-generation  
-----
```

Finally, the script was run in the database again which successfully created the data mart tables.

Screenshots: Table created Report and All tables along with their columns.

The screenshot shows the Oracle Application Express interface. At the top, there's a navigation bar with links like 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'App Gallery'. Below the navigation bar, a search bar and user information are displayed. The main area is titled 'Run Script' with a large 'i' icon. It displays details about the script: 'Script Name: DM\_TableCreation', 'Created: 12/09/2020 07:54:59 AM by AV.SAPKOTA999@GMAIL.COM', 'Updated: 12/09/2020 08:04:22 AM by AV.SAPKOTA999@GMAIL.COM', 'Number of Statements: 14', and 'Script Size in Bytes: 7,438'. There are 'Cancel' and 'Run Now' buttons at the bottom. Below this, the 'Results' tab of the SQL Scripts page is shown. It lists 14 statements with their elapsed times and feedback. The statements include dropping existing tables and creating new ones with constraints. The results summary at the bottom indicates 14 statements processed with 0 errors.

Number	Elapsed	Statement	Feedback	Rows
1	0.65	DROP TABLE FACT_FLIGHTS	Table dropped.	0
2	0.56	DROP TABLE DIM_DELAYED_FLIGHTS	Table dropped.	0
3	0.54	DROP TABLE DIM_CANCELLED_FLIGHTS	Table dropped.	0
4	0.72	DROP TABLE DIM_TIME	Table dropped.	0
5	0.60	DROP TABLE DIM_WEATHER_DATA	Table dropped.	0
6	0.17	CREATE TABLE FACT_FLIGHTS(surrogate_flight_id INTEGER NOT NULL, flight_id INTEGER NOT NULL, flight_time_id INTEGER NOT NULL, month INT, year INT)	Table created.	0
7	0.09	CREATE TABLE DIM_DELAYED_FLIGHTS(delayed_flight_id INTEGER NOT NULL, flight_id INTEGER NOT NULL, month INT, year INT)	Table created.	0
8	0.07	CREATE TABLE DIM_CANCELLED_FLIGHTS(cancelled_flight_id INTEGER NOT NULL, flight_id INTEGER NOT NULL, month INT, year INT)	Table created.	0
9	0.12	CREATE TABLE DIM_TIME(time_id INTEGER NOT NULL, month INT, year INT)	Table created.	0
10	0.09	CREATE TABLE DIM_WEATHER_DATA(weather_data_id INTEGER NOT NULL, location_id INTEGER NOT NULL, month INT, year INT)	Table created.	0
11	0.04	ALTER TABLE FACT_FLIGHTS ADD CONSTRAINT R1_FACT_FLIGHTS_to_DIM_DELAYED_FLIGHTS FOREIGN KEY (delayed_flight_id) REFERENCES DIM_DELAYED_FLIGHTS(delayed_flight_id)	Table altered.	0
12	0.04	ALTER TABLE FACT_FLIGHTS ADD CONSTRAINT R2_FACT_FLIGHTS_to_DIM_CANCELLED_FLIGHTS FOREIGN KEY (cancelled_flight_id) REFERENCES DIM_CANCELLED_FLIGHTS(cancelled_flight_id)	Table altered.	0
13	0.06	ALTER TABLE FACT_FLIGHTS ADD CONSTRAINT R3_FACT_FLIGHTS_to_DIM_TIME FOREIGN KEY (month) REFERENCES DIM_TIME(month)	Table altered.	0
14	0.07	ALTER TABLE FACT_FLIGHTS ADD CONSTRAINT R4_FACT_FLIGHTS_to_DIM_WEATHER_DATA FOREIGN KEY (weather_data_id) REFERENCES DIM_WEATHER_DATA(weather_data_id)	Table altered.	0

**FACT\_FLIGHTS**

---

Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries
	Add Column	Modify Column	Rename Column	Drop Column	Rename	Copy	Drop	Truncate	Create Lookup Table	Create App		
Column Name		Data Type		Nullable		Default						Primary Key
SURROGATE_FLIGHT_ID		NUMBER		No		-						1
FLIGHT_SERIAL_NO		NUMBER		No		-						-
CANCELLATION_FLAG		NUMBER		No		-						-
ARRIVAL_DELAY		NUMBER		Yes		-						-
WEATHER_DELAY		NUMBER		Yes		-						-
FK1_DELAYED_FLIGHT_ID		NUMBER		No		-						-
FK2_CANCELLED_FLIGHT_ID		NUMBER		No		-						-
FK3_TIME_ID		NUMBER		No		-						-
FK4_WEATHER_DATA_ID		NUMBER		No		-						-

[Download](#) | [Print](#)

---

av.sapkota999@gmail.com ads\_level\_6 en Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved. Application Express 20.2.0.0.20

**DIM\_DELAYED\_FLIGHTS**

---

Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries
	Add Column	Modify Column	Rename Column	Drop Column	Rename	Copy	Drop	Truncate	Create Lookup Table	Create App		
Column Name		Data Type		Nullable		Default						Primary Key
DELAYED_FLIGHT_ID		NUMBER		No		-						1
FLIGHT_NUMBER		NUMBER		No		-						-
TAIL_NUMBER		VARCHAR2(50)		No		-						-
AIR_SYSTEM_DELAY		NUMBER		Yes		-						-
SECURITY_DELAY		NUMBER		Yes		-						-
AIRLINE_DELAY		NUMBER		Yes		-						-
LATE_AIRCRAFT_DELAY		NUMBER		Yes		-						-
WEATHER_DELAY		NUMBER		Yes		-						-
FLIGHTS_AIRLINE		VARCHAR2(20)		No		-						-
DATA_SOURCE		VARCHAR2(20)		No		-						-

[Download](#) | [Print](#)

---

av.sapkota999@gmail.com ads\_level\_6 en Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved. Application Express 20.2.0.0.20

DIM\_CANCELLED\_FLIGHTS

Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries	+ v
Actions													
	Add Column	Modify Column	Rename Column	Drop Column	Rename	Copy	Drop	Truncate	Create Lookup Table	Create App			
Column Name	Data Type			Nullable	Default			Primary Key					
CANCELLED_FLIGHT_ID	NUMBER			No	-			1					
FLIGHT_NUMBER	NUMBER			No	-			-					
TAIL_NUMBER	VARCHAR2(50)			No	-			-					
FLIGHTS_AIRLINE	VARCHAR2(20)			No	-			-					
CANCELLATION_REASON	VARCHAR2(120)			No	-			-					
DATA_SOURCE	VARCHAR2(20)			No	-			-					
<a href="#">Download</a>   <a href="#">Print</a>													

av.sapkota999@gmail.com [ads\\_level\\_6](#) en Copyright © 1999-2020, Oracle and/or its affiliates. All rights reserved. Application Express 20.2.0.0.0.20

DIM\_WEATHER\_DATA

Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries	+ v
Actions													
	Add Column	Modify Column	Rename Column	Drop Column	Rename	Copy	Drop	Truncate	Create Lookup Table	Create App			
Column Name	Data Type			Nullable	Default			Primary Key					
WEATHER_DATA_ID	NUMBER			No	-			1					
WEATHER_STATION	VARCHAR2(20)			No	-			-					
WEATHER_DATE	DATE			No	-			-					
AWND	FLOAT			No	-			-					
PRCP	FLOAT			No	-			-					
TAVG	NUMBER			No	-			-					
<a href="#">Download</a>   <a href="#">Print</a>													

av.sapkota999@gmail.com [ads\\_level\\_6](#) en Copyright © 1999-2020, Oracle and/or its affiliates. All rights reserved. Application Express 20.2.0.0.0.20

DIM\_TIME

Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries	+ v
	Add Column	Modify Column	Rename Column	Drop Column	Rename	Copy	Drop	Truncate	Create Lookup Table	Create App			
Column Name	Data Type		Nullable		Default						Primary Key		
TIME_ID	NUMBER		No		-						1		
MONTH	NUMBER		Yes		-						-		
QUARTER	NUMBER		Yes		-						-		
YEAR	NUMBER		No		-						-		

[Download](#) | [Print](#)

 avhimanhusapkota999@gmail.com  ads\_Level\_6  en

Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved.

Application Express 20.2.0.0.20

## Task 3: Extract, Transform and Load (ETL) script to populate the Star Schema (DM) with data

The data sources of FlyU Flights including Flights\_2017, Flights\_2018 and Weather\_Data has been uploaded into the database. Secondly, the data mart (data warehouse) framework has been created using the auto-generated SQL file. According to task 3, we are supposed to extract the data from data source and transform and load those data to the data mart tables. However, there are several stages within the ETL process. ETL is a process of extracting data from various sources, then transforming the data by applying various calculations and measures and finally loading the data into the data warehouse system.

### Extraction

The first step of ETL is Extraction where data is extracted from the source system into the staging area. All the data is collected from the OLTP database and all the required data is loaded into the staging tables. Thus, in case of FlyU airlines data has been provided on various basis. Thus, to store data into staging area staging tables has been created. In addition, sequence and trigger has also been created for each staging table. These sequences will increase the value stored in primary\_key sequentially starting from 1. The trigger has been created to change the value of sequence automatically before every insert in each table. The script used to create staging tables, their sequence and trigger:

N.B. You can also find these codes in a separate file: ‘Task 3 - Staging - Create Tables, Sequences and Triggers’ in the attached folder: ‘SQL Scripts’

```
-- STAGING AREA - TABLES --  
  
-- ALL SQL AND PL/SQL CODE REQUIRED FOR STAGING TABLE: DIM_DELAYED_FLIGHTS  
  
-- DROP TABLE STAGING_DELAYED_FLIGHTS BEFORE CREATION  
DROP TABLE STAGING_DELAYED_FLIGHTS;  
  
-- CREATE TABLE STAGING_DELAYED_FLIGHTS  
CREATE TABLE STAGING_DELAYED_FLIGHTS(  
    delayed_flight_id  INTEGER NOT NULL,  
    flight_number     INTEGER,  
    tail_number      VARCHAR(50),  
    air_system_delay  INTEGER,  
    security_delay    INTEGER,  
    airline_delay     INTEGER,  
    late_aircraft_delay INTEGER,  
    weather_delay     INTEGER,  
    flights_airline   VARCHAR(20),  
    Data_Source        VARCHAR(20) NOT NULL,
```

```

        CONSTRAINT pk_STAGING_DELAYED_FLIGHTS PRIMARY KEY (delayed_flight_id)
);

-- DROP SEQUENCE STAGING_DELAYED_FLIGHTS_SEQ BEFORE CREATION
DROP SEQUENCE STAGING_DELAYED_FLIGHTS_SEQ;

-- CREATE SEQUENCE STAGING_DELAYED_FLIGHTS_SEQ
CREATE SEQUENCE STAGING_DELAYED_FLIGHTS_SEQ INCREMENT BY 1 START WITH 1;

-- CREATE TRIGGER WHICH INCREASES THE VALUE OF PRIMARY KEY EVERY TIME A NEW DATA IS INSERTED.
CREATE OR REPLACE TRIGGER STAGING_DELAYED_FLIGHTS_TRIGGER
BEFORE INSERT ON STAGING_DELAYED_FLIGHTS

FOR EACH ROW
BEGIN
    :new.delayed_flight_id := STAGING_DELAYED_FLIGHTS_SEQ.NEXTVAL;
END;
/
-----



-----


-- ALL SQL AND PL/SQL CODE REQUIRED FOR STAGING TABLE: DIM_CANCELLED_FLIGHTS
-----


-- DROP TABLE STAGING_CANCELLED_FLIGHTS BEFORE CREATION
DROP TABLE STAGING_CANCELLED_FLIGHTS;

-- CREATE TABLE STAGING_CANCELLED_FLIGHTS
CREATE TABLE STAGING_CANCELLED_FLIGHTS(
    cancelled_flight_id INTEGER NOT NULL,
    flight_number INTEGER,
    tail_number VARCHAR(50),
    flights_airline VARCHAR(20),
    cancellation_reason VARCHAR(8),
    Data_Source VARCHAR(20) NOT NULL,

        CONSTRAINT pk_STAGING_CANCELLED_FLIGHTS PRIMARY KEY (cancelled_flight_id)
);

-- DROP SEQUENCE STAGING_CANCELLED_FLIGHTS_SEQ BEFORE CREATION
DROP SEQUENCE STAGING_CANCELLED_FLIGHTS_SEQ;

```

```

-- CREATE SEQUENCE STAGING_CANCELLED_FLIGHTS_SEQ
CREATE SEQUENCE STAGING_CANCELLED_FLIGHTS_SEQ INCREMENT BY 1 START WITH 1;

-- CREATE TRIGGER WHICH INCREASES THE VALUE OF PRIMARY KEY EVERY TIME A NEW DATA IS INSERTED.
CREATE OR REPLACE TRIGGER STAGING_CANCELLED_FLIGHTS_TRIGGER
BEFORE INSERT ON STAGING_CANCELLED_FLIGHTS

FOR EACH ROW
BEGIN
    :new.cancelled_flight_id := STAGING_CANCELLED_FLIGHTS_SEQ.NEXTVAL;
END;
/
-----
```

---

```

-- ALL SQL AND PL/SQL CODE REQUIRED FOR STAGING TABLE: DIM_WEATHER_DATA
-----
```

---

```

-- DROP TABLE STAGING_WEATHER_DATA BEFORE CREATION
DROP TABLE STAGING_WEATHER_DATA;

-- CREATE TABLE STAGING_WEATHER_DATA
CREATE TABLE STAGING_WEATHER_DATA(
    weather_data_id INTEGER NOT NULL,
    weather_station VARCHAR(20),
    weather_date     VARCHAR(50),
    AWND      FLOAT(8),
    PRCP      FLOAT(8),
    TAVG      INTEGER,
    CONSTRAINT pk_STAGING_WEATHER_DATA PRIMARY KEY (weather_data_id)
);

-- DROP SEQUENCE STAGING_WEATHER_DATA_SEQ BEFORE CREATION
DROP SEQUENCE STAGING_WEATHER_DATA_SEQ;

-- CREATE SEQUENCE STAGING_WEATHER_DATA_SEQ
CREATE SEQUENCE STAGING_WEATHER_DATA_SEQ INCREMENT BY 1 START WITH 1;

-- CREATE TRIGGER WHICH INCREASES THE VALUE OF PRIMARY KEY EVERY TIME A NEW DATA IS INSERTED.
CREATE OR REPLACE TRIGGER STAGING_WEATHER_DATA_TRIGGER
BEFORE INSERT ON STAGING_WEATHER_DATA

```

```
FOR EACH ROW
BEGIN
    :new.weather_data_id := STAGING_WEATHER_DATA_SEQ.NEXTVAL;
END;
/
----- END OF STAGING TABLE SCRIPT -----
```

## Screenshots: Staging table Script Run Reports, tables, sequences and triggers created.

SQL Scripts Run Script Schema: WKSP\_ADSLEVEL6

**Run Script**

You have requested to run the following script. Please confirm your request.

Script Name	ETL_Staging_Area
Created	on 12/09/2020 12:15:03 PM by AV.SAPKOTA999@GMAIL.COM
Updated	on 12/10/2020 09:35:51 AM by AV.SAPKOTA999@GMAIL.COM
Number of Statements	15
Script Size in Bytes	4970

av.sapkota999@gmail.com ads\_Level\_6 en Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved. Application Express 20.2.0.0.20

Script: ETL\_Staging\_Area Status: Complete

View:  Detail  Summary Rows: 15 Go

Create App Edit Script

Number	Elapsed	Statement	Feedback	Rows
1	0.87	DROP TABLE STAGING_DELAYED_FLIGHTS	Table dropped.	0
2	0.07	CREATE TABLE STAGING_DELAYED_FLIGHTS(delayed_flight_id	Table created.	0
3	0.03	DROP SEQUENCE STAGING_DELAYED_FLIGHTS_SEQ	Sequence dropped.	0
4	0.02	CREATE SEQUENCE STAGING_DELAYED_FLIGHTS_SEQ INCREMENT BY 1	Sequence created.	0
5	0.08	CREATE OR REPLACE TRIGGER STAGING_DELAYED_FLIGHTS_TRIGGER BE	Trigger created.	0
6	0.54	DROP TABLE STAGING_CANCELLED_FLIGHTS	Table dropped.	0
7	0.08	CREATE TABLE STAGING_CANCELLED_FLIGHTS(cancelled_flight	Table created.	0
8	0.02	DROP SEQUENCE STAGING_CANCELLED_FLIGHTS_SEQ	Sequence dropped.	0
9	0.03	CREATE SEQUENCE STAGING_CANCELLED_FLIGHTS_SEQ INCREMENT BY 1	Sequence created.	0
10	0.09	CREATE OR REPLACE TRIGGER STAGING_CANCELLED_FLIGHTS_TRIGGER	Trigger created.	0
11	0.57	DROP TABLE STAGING_WEATHER_DATA	Table dropped.	0
12	0.07	CREATE TABLE STAGING_WEATHER_DATA(weather_data_id INTG	Table created.	0
13	0.04	DROP SEQUENCE STAGING_WEATHER_DATA_SEQ	Sequence dropped.	0
14	0.06	CREATE SEQUENCE STAGING_WEATHER_DATA_SEQ INCREMENT BY 1 STAR	Sequence created.	0
15	0.09	CREATE OR REPLACE TRIGGER STAGING_WEATHER_DATA_TRIGGER BEFOR	Trigger created.	0

Download row(s) 1 - 15 of 15

15 Statements Processed 15 Successful 0 With Errors

av.sapkota999@gmail.com ads\_Level\_6 en Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved. Application Express 20.2.0.0.20

**STAGING\_DELAYED\_FLIGHTS**

Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries
	<a href="#">Add Column</a>	<a href="#">Modify Column</a>	<a href="#">Rename Column</a>	<a href="#">Drop Column</a>	<a href="#">Rename</a>	<a href="#">Copy</a>	<a href="#">Drop</a>	<a href="#">Truncate</a>	<a href="#">Create Lookup Table</a>	<a href="#">Create App</a>		
Column Name	Data Type		Nullable		Default		Primary Key					
CANCELLED_FLIGHT_ID	NUMBER		No		-		1					
FLIGHT_NUMBER	NUMBER		No		-		-					
TAIL_NUMBER	VARCHAR2(50)		No		-		-					
FLIGHTS_AIRLINE	VARCHAR2(20)		No		-		-					
CANCELLATION_REASON	VARCHAR2(8)		No		-		-					
DATA_SOURCE	VARCHAR2(20)		No		-		-					

[Download](#) | [Print](#)

**STAGING\_DELAYED\_FLIGHTS\_SEQ**

Object Details		Grants	Dependencies	SQL
<a href="#">Alter</a>	<a href="#">Drop</a>			
Min Value	1			
Max Value	99999999999999999999999999999999			
Increment By	1			
Cycle Flag	N			
Order Flag	N			
Cache Size	20			
Last Number	1			

**STAGING\_DELAYED\_FLIGHTS\_TRIGGER**

Object Details		Code	Errors	SQL
<a href="#">Compile</a>	<a href="#">Download</a>	<a href="#">Drop</a>	<a href="#">Disable</a>	
<b>Details</b>				
Object Status	VALID			
Trigger Status	ENABLED			
Trigger Type	BEFORE EACH ROW			
Triggering Event	INSERT			
Base Object Type	TABLE			
Base Object Owner	WKSP_ADSLEVEL6			
Base Object Name	STAGING_DELAYED_FLIGHTS			
Column Name	-			
Referencing Names	REFERENCING NEW AS NEW OLD AS OLD			
When Clause	-			
Description	STAGING_DELAYED_FLIGHTS_TRIGGER BEFORE INSERT ON STAGING_DELAYED_FLIGHTS FOR EACH ROW			
Action Type	PL/SQL			
<b>Columns</b>				
report error: ORA-50720: I/O data limit exceeded - call aborted				

[av.sapkota999@gmail.com](#) [ads\\_level\\_6](#) [en](#) Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved.

Application Express 20.2.0.0.20

**STAGING\_CANCELLED\_FLIGHTS**

Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries
	<a href="#">Add Column</a>	<a href="#">Modify Column</a>	<a href="#">Rename Column</a>	<a href="#">Drop Column</a>	<a href="#">Rename</a>	<a href="#">Copy</a>	<a href="#">Drop</a>	<a href="#">Truncate</a>	<a href="#">Create Lookup Table</a>	<a href="#">Create App</a>		
Column Name	Data Type		Nullable		Default		Primary Key					
CANCELLED_FLIGHT_ID	NUMBER		No		-		1					
FLIGHT_NUMBER	NUMBER		No		-		-					
TAIL_NUMBER	VARCHAR2(50)		No		-		-					
FLIGHTS_AIRLINE	VARCHAR2(20)		No		-		-					
CANCELLATION_REASON	VARCHAR2(8)		No		-		-					
DATA_SOURCE	VARCHAR2(20)		No		-		-					

[Download](#) | [Print](#)

av.sapkota999@gmail.com en Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved.

**STAGING\_CANCELLED\_FLIGHTS\_SEQ**

Object Details		Grants	Dependencies	SQL
<a href="#">Alter</a>	<a href="#">Drop</a>			
Min Value	1			
Max Value	99999999999999999999999999999999			
Increment By	1			
Cycle Flag	N			
Order Flag	N			
Cache Size	20			
Last Number	1			

Application Express 20.2.0.0.20

**STAGING\_CANCELLED\_FLIGHTS\_TRIGGER**

Object Details		Code	Errors	SQL
<a href="#">Compile</a>	<a href="#">Download</a>	<a href="#">Drop</a>	<a href="#">Disable</a>	
<b>Details</b>				
Object Status	VALID			
Trigger Status	ENABLED			
Trigger Type	BEFORE EACH ROW			
Triggering Event	INSERT			
Base Object Type	TABLE			
Base Object Owner	WKSP_ADSLEVEL6			
Base Object Name	STAGING_CANCELLED_FLIGHTS			
Column Name	-			
Referencing Names	REFERENCING NEW AS NEW OLD AS OLD			
When Clause	-			
Description	STAGING_CANCELLED_FLIGHTS_TRIGGER BEFORE INSERT ON STAGING_CANCELLED_FLIGHTS FOR EACH ROW			
Action Type	PL/SQL			
<b>Columns</b>				
report error: ORA-06520: 1/0 data limit exceeded - call aborted				

av.sapkota999@gmail.com en Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved.

Application Express 20.2.0.0.20

av.sapkota999@gmail.com en Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved.

Application Express 20.2.0.0.20

STAGING\_WEATHER\_DATA

Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries
	Add Column	Modify Column	Rename Column	Drop Column	Rename	Copy	Drop	Truncate	Create Lookup Table	Create App		
Column Name	Data Type	Nullable	Default								Primary Key	
ID	NUMBER	No	"WKSP_ADSLEVEL6"."ISEQ\$\$_104978005".nextval								1	
STATION	VARCHAR2(50)	Yes	-								-	
NAME	VARCHAR2(255)	Yes	-								-	
DATE_	VARCHAR2(50)	Yes	-								-	
AWND	NUMBER	Yes	-								-	
PRCP	NUMBER	Yes	-								-	
TAVG	NUMBER	Yes	-								-	

[Download](#) | [Print](#)

av.sapkota999@gmail.com ads\_level\_6 en Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved. Application Express 20.2.0.0.20

STAGING\_WEATHER\_DATA\_SEQ

Object Details		Grants	Dependencies	SQL
Alter	Drop			
Min Value	1			
Max Value	999999999999999999999999999999			
Increment By	1			
Cycle Flag	N			
Order Flag	N			
Cache Size	20			
Last Number	1			

STAGING\_WEATHER\_DATA\_TRIGGER

Object Details		Code	Errors	SQL
Compile	Download	Drop	Disable	
<b>Details</b>				
Object Status	VALID			
Trigger Status	ENABLED			
Trigger Type	BEFORE EACH ROW			
Triggering Event	INSERT			
Base Object Type	TABLE			
Base Object Owner	WKSP_ADSLEVEL6			
Base Object Name	STAGING_WEATHER_DATA			
Column Name	-			
Referencing Names	REFERENCING NEW AS NEW OLD AS OLD			
When Clause	-			
Description	STAGING_WEATHER_DATA_TRIGGER BEFORE INSERT ON STAGING_WEATHER_DATA FOR EACH ROW			
Action Type	PL/SQL			
<b>Columns</b>				
report error; ORA-00040: active time limit exceeded - call aborted				

av.sapkota999@gmail.com ads\_level\_6 en Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved. Application Express 20.2.0.0.20 av.sapkota999@gmail.com ads\_level\_6 en Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved. Application Express 20.2.0.0.20

By this time, all the staging tables along with sequences and triggers have been created successfully so, the next step of staging area is to insert only the necessary data into those staging tables. Thus, in order to insert data from multiple tables into the staging tables, procedures were created and executed using following codes:

N.B. You can also find these codes in a separate file: ‘Task 3 - Staging - Insert Data SQL and PLSQL’ in the attached folder: ‘SQL Scripts’

```
-- PROCEDURE WHICH INSERTS DATA INTO THE STAGING TABLES

-- PROCEDURE TO INSERT DATA INTO STAGING_DELAYED_FLIGHTS
CREATE OR REPLACE PROCEDURE PR_STAGING_DELAYED_FLIGHTS
AS
BEGIN
    -- SELECT AND INSERT ONLY REQUIRED DATA FROM FLIGHTS_2017 TO STAGING AREA
    FOR I IN (SELECT * FROM FLIGHTS_2017 WHERE DEPARTURE_DELAY > 0 OR AIRLINE_DELAY > 0) LOOP
        INSERT INTO STAGING_DELAYED_FLIGHTS (FLIGHT_NUMBER, TAIL_NUMBER, AIR_SYSTEM_DELAY, SECURITY_DELAY,
        AIRLINE_DELAY, LATE_AIRCRAFT_DELAY, WEATHER_DELAY, FLIGHTS_AIRLINE, DATA_SOURCE)
        VALUES (I.FLIGHT_NUMBER, I.TAIL_NUMBER, I.AIR_SYSTEM_DELAY, I.SECURITY_DELAY, I.AIRLINE_DELAY, I.LATE_AIRCRAFT_DELAY, I.WEATHER_DELAY, NULL, 'flights_2017');
    END LOOP;

    -- SELECT AND INSERT ONLY REQUIRED DATA FROM FLIGHTS_2018 TO STAGING AREA
    FOR I IN (SELECT * FROM FLIGHTS_2018 WHERE DEPARTURE_DELAY > 0 OR AIRLINE_DELAY > 0) LOOP
        INSERT INTO STAGING_DELAYED_FLIGHTS (FLIGHT_NUMBER, TAIL_NUMBER, AIR_SYSTEM_DELAY, SECURITY_DELAY,
        AIRLINE_DELAY, LATE_AIRCRAFT_DELAY, WEATHER_DELAY, FLIGHTS_AIRLINE, DATA_SOURCE)
        VALUES (I.FLIGHT_NUMBER, I.TAIL_NUMBER, I.AIR_SYSTEM_DELAY, I.SECURITY_DELAY, I.AIRLINE_DELAY, I.LATE_AIRCRAFT_DELAY, I.WEATHER_DELAY, I.AIRLINE, 'flights_2018');
    END LOOP;
END;
/

-- EXECUTE PROCEDURE PR_STAGING_CANCELLED_FLIGHTS
BEGIN
    PR_STAGING_DELAYED_FLIGHTS;
END;
/


-- PROCEDURE TO INSERT DATA INTO STAGING_CANCELLED_FLIGHTS
CREATE OR REPLACE PROCEDURE PR_STAGING_CANCELLED_FLIGHTS
AS
BEGIN
```

```

-- SELECT AND INSERT ONLY REQUIRED DATA FROM FLIGHTS_2017 TO STAGING AREA
FOR I IN (SELECT * FROM FLIGHTS_2017 WHERE CANCELLED = 1) LOOP
    INSERT INTO STAGING_CANCELLED_FLIGHTS (FLIGHT_NUMBER, TAIL_NUMBER, FLIGHTS_AIRLINE, CANCELLATION_REASON, DATA_SOURCE)
        VALUES (I.FLIGHT_NUMBER, I.TAIL_NUMBER, NULL, I.CANCELLATION_REASON, 'flights_2017');
END LOOP;

-- SELECT AND INSERT ONLY REQUIRED DATA FROM FLIGHTS_2018 TO STAGING AREA
FOR I IN (SELECT * FROM FLIGHTS_2018 WHERE CANCELLED = 1) LOOP
    INSERT INTO STAGING_CANCELLED_FLIGHTS (FLIGHT_NUMBER, TAIL_NUMBER, FLIGHTS_AIRLINE, CANCELLATION_REASON, DATA_SOURCE)
        VALUES (I.FLIGHT_NUMBER, I.TAIL_NUMBER, I.AIRLINE, I.CANCELLATION_REASON, 'flights_2018');
END LOOP;

-- SELECT AND INSERT ONLY REQUIRED DATA FROM FLYU_FLIGHTS TO STAGING AREA
FOR I IN (SELECT * FROM FLYU_FLIGHTS WHERE CANCELLED = 1) LOOP
    INSERT INTO STAGING_CANCELLED_FLIGHTS (FLIGHT_NUMBER, TAIL_NUMBER, FLIGHTS_AIRLINE, CANCELLATION_REASON, DATA_SOURCE)
        VALUES (I.FLIGHT_NUMBER, I.TAIL_NUMBER, NULL, I.CANCELLED_REASON, 'flyu_flights');
END LOOP;
END;
/

-- EXECUTE PROCEDURE PR_STAGING_CANCELLED_FLIGHTS
BEGIN
    PR_STAGING_CANCELLED_FLIGHTS;
END;
/
-----+
-----+


-- PROCEDURE TO INSERT DATA INTO STAGING_WEATHER_DATA
CREATE OR REPLACE PROCEDURE PR_STAGING_WEATHER_DATA
AS
BEGIN
    -- SELECT AND INSERT ALL DATA FROM FLIGHTS_2017 TO STAGING AREA
    FOR I IN (SELECT * FROM WEATHER_DATA) LOOP
        INSERT INTO STAGING_WEATHER_DATA (WEATHER_STATION, WEATHER_DATE, AWND, PRCP, TAVG)
            VALUES (I.STATION, I.DATE_, I.AWND, I.PRCP, I.TAVG);
    END LOOP;
END;
/
-- EXECUTE PROCEDURE PR_STAGING_WEATHER_DATA
BEGIN

```

```

PR_STAGING_WEATHER_DATA;
END;
/

```

----- END OF PROCEDURE CODES -----

Screenshots: Procedure Created and Executed along with data inserted into the tables.

The screenshot shows two consecutive pages from Oracle Application Express (APEX) illustrating the creation and execution of a PL/SQL script.

**Page 1: Run Script Confirmation**

This page displays a confirmation dialog for running a script named "ETL\_Staging\_Area\_Insert". The dialog includes the following details:

Script Name	ETL_Staging_Area_Insert
Created	on 12/10/2020 06:02:45 AM by AV.SAPKOTA999@GMAIL.COM
Updated	on 12/10/2020 09:45:08 AM by AV.SAPKOTA999@GMAIL.COM
Number of Statements	6
Script Size in Bytes	3,842

At the bottom, there are "Cancel" and "Run Now" buttons.

**Page 2: Execution Results**

This page shows the execution results for the script "ETL\_Staging\_Area\_Insert". The status is "Complete". The results table displays the following data:

Number	Elapsed	Statement	Feedback	Rows
1	0.18	CREATE OR REPLACE PROCEDURE PR_STAGING_DELAYED_FLIGHTS AS B	Procedure created.	0
2	0.53	BEGIN PR_STAGING_DELAYED_FLIGHTS; END;	Statement processed.	1
3	0.12	CREATE OR REPLACE PROCEDURE PR_STAGING_CANCELLED_FLIGHTS AS	Procedure created.	0
4	0.11	BEGIN PR_STAGING_CANCELLED_FLIGHTS; END;	Statement processed.	1
5	0.09	CREATE OR REPLACE PROCEDURE PR_STAGING_WEATHER_DATA AS BEGI	Procedure created.	0
6	0.29	BEGIN PR_STAGING_WEATHER_DATA; END;	Statement processed.	1

Below the table, summary statistics are provided:

- Statements Processed: 6
- Successful: 6
- With Errors: 0

**STAGING\_DELAYED\_FLIGHTS**

EDIT	DELAYED_FLIGHT_ID	FLIGHT_NUMBER	TAIL_NUMBER	AIR_SYSTEM_DELAY	SECURITY_DELAY	AIRLINE_DELAY	LATE_AIRCRAFT_DELAY	WEATHER_DELAY	FLIGHTS_AIRLINE	DATA_SOURCE
515	1919	N107US	152	0	0	0	0	0	-	flights_2017
516	67	N3GHAA	17	0	0	0	16	0	-	flights_2017
517	1737	N568UW	33	0	0	0	0	0	-	flights_2017
518	235	N87AA	-	-	-	-	-	-	-	flights_2017
519	1770	N704US	3	0	139	0	0	0	-	flights_2017
520	84	N5CCA	-	-	-	-	-	-	-	flights_2017
521	670	N507AY	0	0	24	0	0	0	-	flights_2017
522	1849	N840AW	-	-	-	-	-	-	-	flights_2017
523	2069	N104UW	-	-	-	-	-	-	-	flights_2017
524	199	N3GHAA	-	-	-	-	-	-	-	flights_2017
525	1919	N109UW	-	-	-	-	-	-	-	flights_2017
526	1723	N753US	-	-	-	-	-	-	-	flights_2017
527	1922	N737US	-	-	-	-	-	-	-	flights_2017
528	235	N3EPAA	0	0	0	40	0	0	-	flights_2017

av.sapkota999@gmail.com ads\_level\_6 en Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved. Application Express 20.2.0.0.20

**STAGING\_CANCELLED\_FLIGHTS**

EDIT	CANCELLED_FLIGHT_ID	FLIGHT_NUMBER	TAIL_NUMBER	FLIGHTS_AIRLINE	CANCELLATION_REASON	DATA_SOURCE
1	291	-	-	B	-	flights_2017
2	2314	-	-	B	-	flights_2017
3	84	-	-	B	-	flights_2017
4	1653	-	-	B	-	flights_2017
5	193	-	+	B	-	flights_2017
6	1096	N3DAAA	-	B	-	flights_2017
7	2314	N3DAAA	-	B	-	flights_2017
8	84	N3HSAA	-	B	-	flights_2017
9	1653	N3DMAA	-	B	-	flights_2017
10	193	-	-	B	-	flights_2017
11	1096	N3HAA	-	B	-	flights_2017
12	291	N3BMAA	-	B	-	flights_2017
13	2314	N3GRAA	-	B	-	flights_2017
14	2314	N3AAAA	-	B	-	flights_2017
15	84	N3ELAA	-	B	-	flights_2017

av.sapkota999@gmail.com ads\_level\_6 en Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved. Application Express 20.2.0.0.20

STAGING_WEATHER_DATA						
EDIT	WEATHER_DATA_ID	WEATHER_STATION	WEATHER_DATE	AWND	PRCP	TAVG
538	USW00094789	21/06/2018	16.3	.82	73	
539	USW00094789	22/06/2018	15	.08	60	
540	USW00094789	23/06/2018	18.6	1.81	54	
541	USW00094789	24/06/2018	19.5	.16	53	
542	USW00094789	25/06/2018	15.4	0	58	
543	USW00094789	26/06/2018	11.9	0	58	
544	USW00094789	27/06/2018	5.82	0	61	
545	USW00094789	28/06/2018	7.16	0	64	
546	USW00094789	29/06/2018	6.26	0	64	
547	USW00094789	30/06/2018	13.4	.28	66	
548	USW00094789	1/7/2018	9.4	0	61	
549	USW00094789	2/7/2018	9.62	0	58	
550	USW00094789	3/7/2018	8.5	0	62	
551	USW00094789	4/7/2018	9.4	0	66	
552	USW00094789	5/7/2018	11.6	0	65	

## Transformation

Data extracted from the source may or may not be useable or are raw as they are in their unique form hence, they are firstly supposed to be cleaned, mapped and then transform.

### Transformation: Bad Data Identification, Segregation and Insert

The next stage of the ETL process requires to address data quality issues. As documented in the data dictionary, many data quality issues have been identified. Hence, in this stage good quality data is transferred to clean table and bad quality data are migrated to bad table where those values are collectively managed and solved. After the successful addressment of bad data, they are further migrated to clean table.

Firstly, bad data tables were created, along with the sequence and triggers to insert primary key value, for each staging table using following codes:

N.B. You can also find these codes in a separate file: 'Task 3 - Transformation - Bad Table Create SQL' in the attached folder: 'SQL Scripts'

```
-- BAD DATA - TABLES --
```

---



---



---

```
-- ALL SQL AND PL/SQL CODE REQUIRED FOR BAD DATA TABLE: DIM_DELAYED_FLIGHTS
```

---



---

```
-- DROP TABLE BAD_DELAYED_FLIGHTS BEFORE CREATION
```

```
DROP TABLE BAD_DELAYED_FLIGHTS;
```

---

```
-- CREATE TABLE BAD_DELAYED_FLIGHTS
```

```
CREATE TABLE BAD_DELAYED_FLIGHTS(
```

```
    bad_delayed_flight_id  INTEGER NOT NULL,
```

```

delayed_flight_id  INTEGER NOT NULL,
flight_number     INTEGER,
tail_number       VARCHAR(50),
air_system_delay   INTEGER,
security_delay    INTEGER,
airline_delay     INTEGER,
late_aircraft_delay INTEGER,
weather_delay      INTEGER,
flights_airline   VARCHAR(20),
Data_Source        VARCHAR(20) NOT NULL,
error_description  VARCHAR(120),
error_status       VARCHAR(20),
resolution_date    DATE,
CONSTRAINT pk_BAD_DELAYED_FLIGHTS PRIMARY KEY (bad_delayed_flight_id)
);

-- DROP SEQUENCE BAD_DELAYED_FLIGHTS_SEQ BEFORE CREATION
DROP SEQUENCE BAD_DELAYED_FLIGHTS_SEQ;

-- CREATE SEQUENCE BAD_DELAYED_FLIGHTS_SEQ
CREATE SEQUENCE BAD_DELAYED_FLIGHTS_SEQ INCREMENT BY 1 START WITH 1;

-- CREATE TRIGGER WHICH INCREASES THE VALUE OF PRIMARY KEY EVERY TIME A NEW DATA IS INSERTED.
CREATE OR REPLACE TRIGGER BAD_DELAYED_FLIGHTS_TRIGGER
BEFORE INSERT ON BAD_DELAYED_FLIGHTS
FOR EACH ROW
BEGIN
: new.bad_delayed_flight_id := BAD_DELAYED_FLIGHTS_SEQ.NEXTVAL;
END;
/
-----  

-----  

-----  

-- ALL SQL AND PL/SQL CODE REQUIRED FOR BAD TABLE: DIM_CANCELLED_FLIGHTS
-----  

-----  

-- DROP TABLE BAD_CANCELLED_FLIGHTS BEFORE CREATION
DROP TABLE BAD_CANCELLED_FLIGHTS;

-- CREATE TABLE BAD_CANCELLED_FLIGHTS
CREATE TABLE BAD_CANCELLED_FLIGHTS(

```

```

bad_cancelled_flight_id INTEGER NOT NULL,
cancelled_flight_id INTEGER NOT NULL,
flight_number INTEGER,
tail_number VARCHAR(50),
flights_airline VARCHAR(20),
cancellation_reason VARCHAR(8),
Data_Source VARCHAR(20) NOT NULL,
error_description VARCHAR(120),
error_status VARCHAR(20),
resolution_date DATE,

CONSTRAINT pk_BAD_CANCELLED_FLIGHTS PRIMARY KEY (bad_cancelled_flight_id)
);

-- DROP SEQUENCE BAD_CANCELLED_FLIGHTS_SEQ BEFORE CREATION
DROP SEQUENCE BAD_CANCELLED_FLIGHTS_SEQ;

-- CREATE SEQUENCE BAD_CANCELLED_FLIGHTS_SEQ
CREATE SEQUENCE BAD_CANCELLED_FLIGHTS_SEQ INCREMENT BY 1 START WITH 1;

-- CREATE TRIGGER WHICH INCREASES THE VALUE OF PRIMARY KEY EVERY TIME A NEW DATA IS INSERTED.
CREATE OR REPLACE TRIGGER BAD_CANCELLED_FLIGHTS_TRIGGER
BEFORE INSERT ON BAD_CANCELLED_FLIGHTS

FOR EACH ROW
BEGIN
:new.bad_cancelled_flight_id := BAD_CANCELLED_FLIGHTS_SEQ.NEXTVAL;
END;
/

-----  

-----  

-----  

-----  

-----  

-- ALL SQL AND PL/SQL CODE REQUIRED FOR BAD TABLE: DIM_WEATHER_DATA
-----  

-----  

-----  

-- DROP TABLE BAD_WEATHER_DATA BEFORE CREATION
DROP TABLE BAD_WEATHER_DATA;

-- CREATE TABLE BAD_WEATHER_DATA
CREATE TABLE BAD_WEATHER_DATA(
bad_weather_data_id INTEGER NOT NULL,
weather_data_id INTEGER NOT NULL,
weather_station VARCHAR(20),

```

```

weather_date      VARCHAR(50),
AWND      FLOAT(8),
PRCP      FLOAT(8),
TAVG      INTEGER,
error_description VARCHAR(120),
error_status  VARCHAR(20),
resolution_date DATE,
CONSTRAINT pk_BAD_WEATHER_DATA PRIMARY KEY (bad_weather_data_id)
);

-- DROP SEQUENCE BAD_WEATHER_DATA_SEQ BEFORE CREATION
DROP SEQUENCE BAD_WEATHER_DATA_SEQ;

-- CREATE SEQUENCE BAD_WEATHER_DATA_SEQ
CREATE SEQUENCE BAD_WEATHER_DATA_SEQ INCREMENT BY 1 START WITH 1;

-- CREATE TRIGGER WHICH INCREASES THE VALUE OF PRIMARY KEY EVERY TIME A NEW DATA IS INSERTED.
CREATE OR REPLACE TRIGGER BAD_WEATHER_DATA_TRIGGER
BEFORE INSERT ON BAD_WEATHER_DATA
FOR EACH ROW
BEGIN
:new.bad_weather_data_id := BAD_WEATHER_DATA_SEQ.NEXTVAL;
END;
/
----- END OF BAD DATA TABLE CREATION -----

```

Screenshots: Bad data table, sequence and trigger creation run and successfully created report.

The screenshot shows a 'Run Script' confirmation dialog and a detailed execution report.

**Run Script Confirmation Dialog:**

- Script Name: ETL\_BadData\_Tables
- Created: on 12/11/2020 03:17:33 AM by AV.SAPKOTA999@GMAIL.COM
- Updated: -
- Number of Statements: 15
- Script Size in Bytes: 5,260

**Execution Report:**

Number	Elapsed	Statement	Feedback	Rows
1	0.90	DROP TABLE BAD_DELAYED_FLIGHTS	Table dropped.	0
2	0.07	CREATE TABLE BAD_DELAYED_FLIGHTS( bad_delayed_flight_id	Table created.	0
3	0.03	DROP SEQUENCE BAD_DELAYED_FLIGHTS_SEQ	Sequence dropped.	0
4	0.02	CREATE SEQUENCE BAD_DELAYED_FLIGHTS_SEQ INCREMENT BY 1 START	Sequence created.	0
5	0.10	CREATE OR REPLACE TRIGGER BAD_DELAYED_FLIGHTS_TRIGGER BEFORE	Trigger created.	0
6	0.61	DROP TABLE BAD_CANCELLED_FLIGHTS	Table dropped.	0
7	0.07	CREATE TABLE BAD_CANCELLED_FLIGHTS( bad_cancelled_flight	Table created.	0
8	0.02	DROP SEQUENCE BAD_CANCELLED_FLIGHTS_SEQ	Sequence dropped.	0
9	0.03	CREATE SEQUENCE BAD_CANCELLED_FLIGHTS_SEQ INCREMENT BY 1 STA	Sequence created.	0
10	0.08	CREATE OR REPLACE TRIGGER BAD_CANCELLED_FLIGHTS_TRIGGER BEFO	Trigger created.	0
11	0.54	DROP TABLE BAD_WEATHER_DATA	Table dropped.	0
12	0.06	CREATE TABLE BAD_WEATHER_DATA( bad_weather_data_id INTEG	Table created.	0
13	0.02	DROP SEQUENCE BAD_WEATHER_DATA_SEQ	Sequence dropped.	0
14	0.02	CREATE SEQUENCE BAD_WEATHER_DATA_SEQ INCREMENT BY 1 START WI	Sequence created.	0
15	0.08	CREATE OR REPLACE TRIGGER BAD_WEATHER_DATA_TRIGGER BEFORE IN	Trigger created.	0

**Summary Metrics:**

- Statements Processed: 15
- Successful: 15
- With Errors: 0

Henceforth after the creation of bad tables, bad data were identified from each staging table and were inserted into the bad tables using following code:

N.B. You can also find these codes in a separate file: 'Task 3 - Transformation - Bad Data Insert PLSQL' in the attached folder: 'SQL Scripts'

---

```
-- PL/SQL CODES TO IDENTIFY BAD DATA AND INSERT INTO BAD TABLE
```

---

```
-- PROCEDURE TO IDENTIFY AND INSERT BAD DATA INTO BAD_DELAYED_FLIGHTS
```

```

CREATE OR REPLACE PROCEDURE PR_BAD_DELAYED_FLIGHTS
AS
BEGIN
    -- IDENTIFYING BAD DATA AND INSERT INTO BAD TABLE
    FOR I IN (SELECT * FROM STAGING_DELAYED_FLIGHTS

        WHERE AIR_SYSTEM_DELAY IS NULL OR
        SECURITY_DELAY IS NULL OR
        AIRLINE_DELAY IS NULL OR
        LATE_AIRCRAFT_DELAY IS NULL OR
        WEATHER_DELAY IS NULL OR
        FLIGHTS_AIRLINE IS NULL
        ORDER BY DELAYED_FLIGHT_ID) LOOP

        INSERT INTO BAD_DELAYED_FLIGHTS (DELAYED_FLIGHT_ID, FLIGHT_NUMBER, TAIL_NUMBER, AIR_SYSTEM_DELAY,
        SECURITY_DELAY, AIRLINE_DELAY, LATE_AIRCRAFT_DELAY,
        WEATHER_DELAY, FLIGHTS_AIRLINE, DATA_SOURCE, ERROR_DESCRIPTION, ERROR_STATUS, RESOLUTI
        ON_DATE)
        VALUES (I.DELAYED_FLIGHT_ID, I.FLIGHT_NUMBER, I.TAIL_NUMBER, I.AIR_SYSTEM_DELAY, I.SECURITY_DELAY,
        I.AIRLINE_DELAY, I.LATE_AIRCRAFT_DELAY, I.WEATHER_DELAY, I.FLIGHTS_AIRLINE, I.DATA_SOURCE,
        'NULL VALUES IN AIR_SYSTEM_DELAY / SECURITY_DELAY / AIRLINE_DELAY / LATE_AIRCRAFT_DELAY /
        WEATHER_DELAY / FLIGHTS_AIRLINE', 'UNRESOLVED', NULL);
    END LOOP;
END;
/

-- EXECUTE PROCEDURE PR_BAD_DELAYED_FLIGHTS
BEGIN
    PR_BAD_DELAYED_FLIGHTS;
END;
/
-----
```

---

```

-- PROCEDURE TO IDENTIFY AND INSERT BAD DATA INTO BAD_CANCELLED_FLIGHTS
CREATE OR REPLACE PROCEDURE PR_BAD_CANCELLED_FLIGHTS
AS
BEGIN
    -- IDENTIFYING BAD DATA AND INSERT INTO BAD TABLE
    FOR I IN (SELECT * FROM STAGING_CANCELLED_FLIGHTS

        WHERE TAIL_NUMBER IS NULL OR
        FLIGHTS_AIRLINE IS NULL OR
        LENGTH(CANCELLATION_REASON) = 1
        ORDER BY CANCELLED_FLIGHT_ID) LOOP
```

```

        INSERT INTO BAD_CANCELLED_FLIGHTS (CANCELLED_FLIGHT_ID, FLIGHT_NUMBER, TAIL_NUMBER, FLIGHTS_AIRLINE,
        CANCELLATION_REASON, DATA_SOURCE,
        ERROR_DESCRIPTION, ERROR_STATUS, RESOLUTION_DATE)
        VALUES (I.CANCELLED_FLIGHT_ID, I.FLIGHT_NUMBER, I.TAIL_NUMBER, I.FLIGHTS_AIRLINE, I.CANCELLATION_REASON,
        I.DATA_SOURCE,
        'NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON', 'UNRESOLVED',
        NULL);
    END LOOP;
END;
/

-- EXECUTE PROCEDURE PR_BAD_CANCELLED_FLIGHTS
BEGIN
    PR_BAD_CANCELLED_FLIGHTS;
END;
/
-----


-- PROCEDURE TO IDENTIFY AND INSERT BAD DATA INTO BAD_WEATHER_DATA
CREATE OR REPLACE PROCEDURE PR_BAD_WEATHER_DATA
AS
BEGIN
    -- IDENTIFYING BAD DATA AND INSERT INTO BAD TABLE
    FOR I IN (SELECT * FROM STAGING_WEATHER_DATA WHERE LENGTH(WEATHER_DATE) != 10
              ORDER BY WEATHER_DATA_ID) LOOP

        INSERT INTO BAD_WEATHER_DATA (WEATHER_DATA_ID, WEATHER_STATION, WEATHER_DATE, AWND, PRCP, TAVG, ERROR_DESCRIPTION, ERROR_STATUS, RESOLUTION_DATE)
        VALUES (I.WEATHER_DATA_ID, I.WEATHER_STATION, I.WEATHER_DATE, I.AWND, I.PRCP, I.TAVG, 'INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY', 'UNRESOLVED', NULL);
    END LOOP;
END;
/
-----


-- EXECUTE PROCEDURE PR_BAD_WEATHER_DATA
BEGIN
    PR_BAD_WEATHER_DATA;
END;
/

```

Screenshot: Procedure Created and Executed along with data inserted into the tables.

The screenshot shows two pages from Oracle Application Express (APEX) illustrating the creation and execution of a SQL script named ETL\_BadData\_Insert.

**Page 1: Run Script Confirmation**

This page displays a confirmation message: "You have requested to run the following script. Please confirm your request." Below this is a table showing the script's details:

Script Name	ETL_BadData_Insert
Created	on 12/11/2020 03:14:07 AM by AV.SAPKOTA999@GMAIL.COM
Updated	on 12/11/2020 03:52:48 AM by AV.SAPKOTA999@GMAIL.COM
Number of Statements	6
Script Size in Bytes	3,734

At the bottom are "Cancel" and "Run Now" buttons.

**Page 2: Results Page**

This page shows the execution results of the script. The table has the following columns: Number, Elapsed, Statement, Feedback, and Rows.

Number	Elapsed	Statement	Feedback	Rows
1	0.15	CREATE OR REPLACE PROCEDURE PR_BAD_DELAYED_FLIGHTS AS BEGIN	Procedure created.	0
2	0.50	BEGIN PR_BAD_DELAYED_FLIGHTS; END;	Statement processed.	1
3	0.13	CREATE OR REPLACE PROCEDURE PR_BAD_CANCELLED_FLIGHTS AS BEG	Procedure created.	0
4	0.11	IN PR_BAD_CANCELLED_FLIGHTS; END;	Statement processed.	1
5	0.04	CREATE OR REPLACE PROCEDURE PR_BAD_WEATHER_DATA AS BEGIN	Procedure created.	0
6	0.13	BEGIN PR_BAD_WEATHER_DATA; END;	Statement processed.	1

Summary statistics at the bottom indicate: 6 Statements Processed, 6 Successful, and 0 With Errors.

BAD_DELAYED_FLIGHTS														
	BAD_DELAYED_FLIGHT_ID	DELAYED_FLIGHT_ID	FLIGHT_NUMBER	TAIL_NUMBER	AIR_SYSTEM_DELAY	SECURITY_DELAY	AIRLINE_DELAY	LATE_AIRCRAFT_DELAY	WEATHER_DELAY	FLIGHTS_AIRLINE	DATA_SOURCE	ERROR_DESCRIPTION	ERROR_STATUS	RESOLUTION_DATE
	121	121	84	NOKHAA	-	-	-	-	-	-	flights_2017	NULL VALUES IN AIR_SYSTEM_DELAY / SECURITY_DELAY / AIRLINE_DELAY / LATE_AIRCRAFT_DELAY / WEATHER_DELAY / FLIGHTS_AIRLINE	UNRESOLVED	-
	122	122	291	NBSCAA	-	-	-	-	-	-	flights_2017	NULL VALUES IN AIR_SYSTEM_DELAY / SECURITY_DELAY / AIRLINE_DELAY / LATE_AIRCRAFT_DELAY / WEATHER_DELAY / FLIGHTS_AIRLINE	UNRESOLVED	-
	123	123	1653	N3JLAA	-	-	-	-	-	-	flights_2017	NULL VALUES IN AIR_SYSTEM_DELAY / SECURITY_DELAY / AIRLINE_DELAY / LATE_AIRCRAFT_DELAY / WEATHER_DELAY / FLIGHTS_AIRLINE	UNRESOLVED	-
	124	124	1096	NSACAA	3	0	29	0	0	-	flights_2017	NULL VALUES IN AIR_SYSTEM_DELAY / SECURITY_DELAY / AIRLINE_DELAY / LATE_AIRCRAFT_DELAY / WEATHER_DELAY / FLIGHTS_AIRLINE	UNRESOLVED	-
	125	125	291	N3DCAA	0	0	2	17	0	-	flights_2017	NULL VALUES IN AIR_SYSTEM_DELAY / SECURITY_DELAY / AIRLINE_DELAY / LATE_AIRCRAFT_DELAY / WEATHER_DELAY / FLIGHTS_AIRLINE	UNRESOLVED	-

av.sapkota99@gmail.com [edit\\_level\\_6](#) [en](#) Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved. Application Express 20.2.0.0.20

BAD_CANCELLED_FLIGHTS														
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries		
Query	Count Rows	Insert Row	Load Data											
EDIT	BAD_CANCELLED_FLIGHT_ID	CANCELLED_FLIGHT_ID	FLIGHT_NUMBER	TAIL_NUMBER	FLIGHTS_AIRLINE	CANCELLATION_REASON	DATA_SOURCE	ERROR_DESCRIPTION				ERROR_STATUS	RESOLUTION_DATE	
<a href="#">1</a>	1	1	291	-	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON				UNRESOLVED	-	
<a href="#">2</a>	2	2	2314	-	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON				UNRESOLVED	-	
<a href="#">3</a>	3	3	84	-	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON				UNRESOLVED	-	
<a href="#">4</a>	4	4	1653	-	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON				UNRESOLVED	-	
<a href="#">5</a>	5	5	193	-	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON				UNRESOLVED	-	
<a href="#">6</a>	6	6	1096	N3DAAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON				UNRESOLVED	-	
<a href="#">7</a>	7	7	2314	N3DAAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON				UNRESOLVED	-	
<a href="#">8</a>	8	8	84	NBHSAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON				UNRESOLVED	-	
<a href="#">9</a>	9	9	1653	N3DMAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON				UNRESOLVED	-	
<a href="#">10</a>	10	10	193	-	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON				UNRESOLVED	-	
<a href="#">11</a>	11	11	1096	N3LHAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON				UNRESOLVED	-	
<a href="#">12</a>	12	12	291	N3BMAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON				UNRESOLVED	-	
<a href="#">13</a>	13	13	2314	N3GAAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON				UNRESOLVED	-	
<a href="#">14</a>	14	14	2314	N3AAAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON				UNRESOLVED	-	
<a href="#">15</a>	15	15	84	N3ELAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON				UNRESOLVED	-	

Download [row\(s\) 1- 15 of 109](#) [Next ▶](#)

BAD_WEATHER_DATA														
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries		
Query	Count Rows	Insert Row	Load Data											
EDIT	BAD_WEATHER_DATA_ID	WEATHER_DATA_ID	WEATHER_STATION	WEATHER_DATE	AWND	PRCP	TAVG	ERROR_DESCRIPTION				ERROR_STATUS	RESOLUTION_DATE	
<a href="#">218</a>	218	558	USW00094789	11/7/2018	15.7	0	56	INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY				UNRESOLVED	-	
<a href="#">219</a>	219	559	USW00094789	12/7/2018	6.04	0	61	INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY				UNRESOLVED	-	
<a href="#">220</a>	220	579	USW00094789	1/8/2018	12.1	.45	57	INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY				UNRESOLVED	-	
<a href="#">221</a>	221	580	USW00094789	2/8/2018	12.8	.09	54	INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY				UNRESOLVED	-	
<a href="#">222</a>	222	581	USW00094789	3/8/2018	11	.05	56	INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY				UNRESOLVED	-	
<a href="#">223</a>	223	582	USW00094789	4/8/2018	20.4	0	57	INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY				UNRESOLVED	-	
<a href="#">224</a>	224	583	USW00094789	5/8/2018	17.5	0	49	INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY				UNRESOLVED	-	
<a href="#">225</a>	225	584	USW00094789	6/8/2018	12.8	0	48	INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY				UNRESOLVED	-	
<a href="#">226</a>	226	585	USW00094789	7/8/2018	9.77	0	56	INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY				UNRESOLVED	-	
<a href="#">227</a>	227	586	USW00094789	8/8/2018	8.05	0	51	INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY				UNRESOLVED	-	
<a href="#">228</a>	228	587	USW00094789	9/8/2018	6.95	0	51	INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY				UNRESOLVED	-	
<a href="#">229</a>	229	588	USW00094789	10/8/2018	7.61	0	55	INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY				UNRESOLVED	-	
<a href="#">230</a>	230	589	USW00094789	11/8/2018	10.1	0	61	INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY				UNRESOLVED	-	
<a href="#">231</a>	231	590	USW00094789	12/8/2018	5.37	0	67	INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY				UNRESOLVED	-	
<a href="#">232</a>	232	610	USW00094789	1/9/2018	12.8	0	66	INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY				UNRESOLVED	-	

Download [row\(s\) 1- 15 of 270](#) [Next ▶](#)

av.sapkota99@gmail.com [edit\\_level\\_6](#) [en](#) Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved. Application Express 20.2.0.0.20

## Transformation: Good Data Identification and Insert

Secondly, good data tables are the tables which stores all the clean data from staging table or data which have been cleaned from bad tables. Good data tables were created, for each staging table, using following codes:

N.B. You can also find these codes in a separate file: ‘Task 3 - Transformation - Good Table Create SQL’ in the attached folder: ‘SQL Scripts’

```
-- GOOD DATA TABLES CREATION --  
  
-- ALL SQL AND PL/SQL CODE REQUIRED FOR GOOD DATA TABLE: DIM_DELAYED_FLIGHTS  
  
-- DROP TABLE GOOD_DELAYED_FLIGHTS BEFORE CREATION  
DROP TABLE GOOD_DELAYED_FLIGHTS;  
  
-- CREATE TABLE GOOD_DELAYED_FLIGHTS  
CREATE TABLE GOOD_DELAYED_FLIGHTS(  
    delayed_flight_id    INTEGER NOT NULL,  
    flight_number        INTEGER NOT NULL,  
    tail_number          VARCHAR(50) NOT NULL,  
    air_system_delay     INTEGER,  
    security_delay       INTEGER,  
    airline_delay        INTEGER,  
    late_aircraft_delay  INTEGER,  
    weather_delay        INTEGER,  
    flights_airline      VARCHAR(20) NOT NULL,  
    Data_Source          VARCHAR(20) NOT NULL,  
  
    CONSTRAINT pk_GOOD_DELAYED_FLIGHTS PRIMARY KEY (delayed_flight_id)  
);  
  
-- ALL SQL AND PL/SQL CODE REQUIRED FOR GOOD TABLE: DIM_CANCELLED_FLIGHTS  
  
-- DROP TABLE GOOD_CANCELLED_FLIGHTS BEFORE CREATION  
DROP TABLE GOOD_CANCELLED_FLIGHTS;  
  
-- CREATE TABLE GOOD_CANCELLED_FLIGHTS  
CREATE TABLE GOOD_CANCELLED_FLIGHTS(  
    cancelled_flight_id  INTEGER NOT NULL,
```

```

flight_number  INTEGER NOT NULL,
tail_number VARCHAR(50) NOT NULL,
flights_airline VARCHAR(20) NOT NULL,
cancellation_reason VARCHAR(120) NOT NULL,
Data_Source VARCHAR(20) NOT NULL,

CONSTRAINT pk_GOOD_CANCELLED_FLIGHTS PRIMARY KEY (cancelled_flight_id)
);

```

---



---



---

-- ALL SQL AND PL/SQL CODE REQUIRED FOR GOOD TABLE: DIM\_WEATHER\_DATA

---

-- DROP TABLE GOOD\_WEATHER\_DATA BEFORE CREATION

```
DROP TABLE GOOD_WEATHER_DATA;
```

-- CREATE TABLE GOOD\_WEATHER\_DATA

```
CREATE TABLE GOOD_WEATHER_DATA(
weather_data_id INTEGER NOT NULL,
weather_station VARCHAR(20) NOT NULL,
weather_date DATE NOT NULL,
AWND    FLOAT(8) NOT NULL,
PRCP    FLOAT(8) NOT NULL,
TAVG    INTEGER NOT NULL,
```

```
CONSTRAINT pk_GOOD_WEATHER_DATA PRIMARY KEY (weather_data_id)
);
```

---

----- END OF GOOD TABLE CREATION CODE -----

---

## Screenshots: Good data table run and successfully created report.

The screenshot shows the Oracle Application Express 'Run Script' dialog. At the top, it displays the script details: Script Name: ETL\_GoodData\_Tables, Created: 12/11/2020 04:28:08 AM by AV.SAPKOTA999@GMAIL.COM, Updated: 12/11/2020 05:15:25 AM by AV.SAPKOTA999@GMAIL.COM, Number of Statements: 6, and Script Size in Bytes: 3,416. Below this, there are 'Cancel' and 'Run Now' buttons. The 'Run Now' button is highlighted in green.

**Script: ETL\_GoodData\_Tables Status: Complete**

View:  Detail  Summary Rows: 15 Go

Number	Elapsed	Statement	Feedback	Rows
1	0.59	DROP TABLE GOOD_DELAYED_FLIGHTS	Table dropped.	0
2	0.08	CREATE TABLE GOOD_DELAYED_FLIGHTS( delayed_flight_id )	Table created.	0
3	0.52	DROP TABLE GOOD_CANCELLED_FLIGHTS	Table dropped.	0
4	0.01	CREATE TABLE GOOD_CANCELLED_FLIGHTS( cancelled_flight_id )	Table created.	0
5	0.51	DROP TABLE GOOD_WEATHER_DATA	Table dropped.	0
6	0.08	CREATE TABLE GOOD_WEATHER_DATA( weather_data_id INTEGER )	Table created.	0

Download row(s) 1 - 6 of 6

6 Statements Processed      6 Successful      0 With Errors

After the creation of good tables, good data were identified from each staging table and were inserted into the good tables using following code:

N.B. You can also find these codes in a separate file: 'Task 3 - Transformation - Good Data Insert PLSQL' in the attached folder: 'SQL Scripts'

```
-- PL/SQL CODES TO IDENTIFY CLEAN DATA AND INSERT INTO GOOD (CLEAN) TABLE

-- PROCEDURE TO IDENTIFY AND INSERT CLEAN DATA INTO GOOD_DELAYED_FLIGHTS
CREATE OR REPLACE PROCEDURE PR_GOOD_DELAYED_FLIGHTS
AS
BEGIN
    -- IDENTIFYING CLEAN DATA AND INSERT INTO GOOD TABLE
    FOR I IN (SELECT * FROM STAGING_DELAYED_FLIGHTS
               WHERE AIR_SYSTEM_DELAY IS NOT NULL AND
                     SECURITY_DELAY IS NOT NULL AND
                     AIRLINE_DELAY IS NOT NULL AND
                     LATE_AIRCRAFT_DELAY IS NOT NULL AND
                     WEATHER_DELAY IS NOT NULL AND
                     FLIGHTS_AIRLINE IS NOT NULL
               ORDER BY DELAYED_FLIGHT_ID) LOOP

        INSERT INTO GOOD_DELAYED_FLIGHTS (DELAYED_FLIGHT_ID, FLIGHT_NUMBER, TAIL_NUMBER, AIR_SYSTEM_DELAY,
                                         SECURITY_DELAY,
                                         AIRLINE_DELAY, LATE_AIRCRAFT_DELAY, WEATHER_DELAY, FLIGHTS_AIRLINE, DATA_SOURCE)
        VALUES (I.DELAYED_FLIGHT_ID, I.FLIGHT_NUMBER, I.TAIL_NUMBER, I.AIR_SYSTEM_DELAY, I.SECURITY_DELAY,
                I.AIRLINE_DELAY, I.LATE_AIRCRAFT_DELAY, I.WEATHER_DELAY, I.FLIGHTS_AIRLINE, I.DATA_SO
RCE);
    END LOOP;
END;
/

-- EXECUTE PROCEDURE PR_GOOD_DELAYED_FLIGHTS
BEGIN
    PR_GOOD_DELAYED_FLIGHTS;
END;
/


-- PROCEDURE TO IDENTIFY AND INSERT CLEAN DATA INTO GOOD_WEATHER_DATA
CREATE OR REPLACE PROCEDURE PR_GOOD_WEATHER_DATA
AS
BEGIN
```

```

-- IDENTIFYING CLEAN DATA AND INSERT INTO GOOD TABLE
FOR I IN (SELECT * FROM STAGING_WEATHER_DATA WHERE LENGTH(WEATHER_DATE) = 10
           ORDER BY WEATHER_DATA_ID) LOOP

    INSERT INTO GOOD_WEATHER_DATA (WEATHER_DATA_ID, WEATHER_STATION, WEATHER_DATE, AWND, PRCP, TAVG)
    VALUES (I.WEATHER_DATA_ID, I.WEATHER_STATION, TO_DATE(I.WEATHER_DATE, 'DD/MM/YYYY'), I.AWND, I.PRC
P, I.TAVG);
    END LOOP;
END;
/

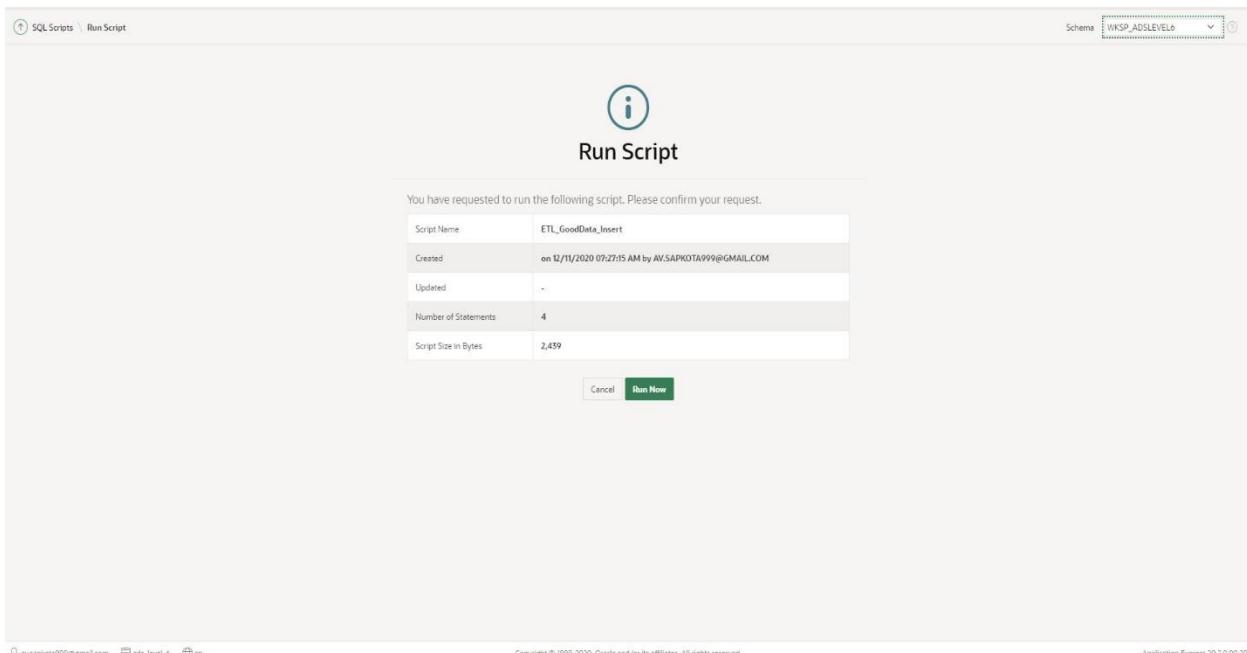
-- EXECUTE PROCEDURE PR_GOOD_WEATHER_DATA
BEGIN
    PR_GOOD_WEATHER_DATA;
END;
/

```

---

-----  
END OF GOOD DATA INSERT CODE -----  
-----

## Screenshots: Good Data Insert Script Run Report and Tables with data loaded.

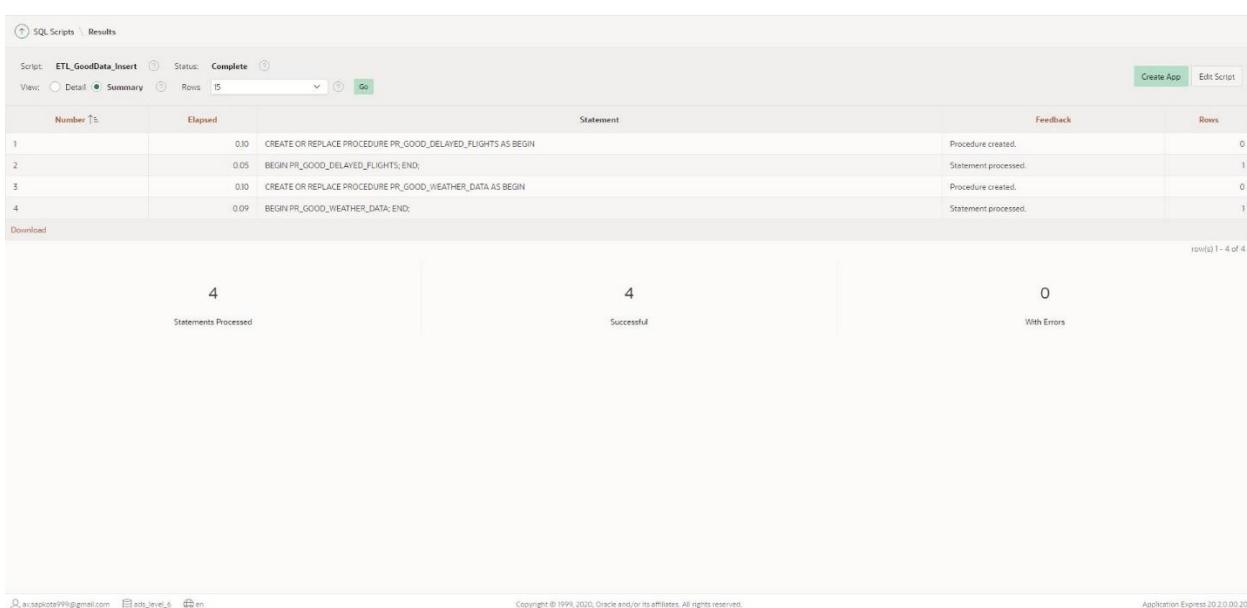


The screenshot shows a confirmation dialog titled "Run Script". It displays the following information:

- Script Name: ETL\_GoodData\_Insert
- Created: on 12/11/2020 07:27:15 AM by AV.SAPKOTA999@GMAIL.COM
- Updated: -
- Number of Statements: 4
- Script Size in Bytes: 2,439

At the bottom, there are "Cancel" and "Run Now" buttons.

---



The screenshot shows the results of the executed script "ETL\_GoodData\_Insert". The summary table is as follows:

Number	Elapsed	Statement	Feedback	Rows
1	0.10	CREATE OR REPLACE PROCEDURE PR_GOOD_DELAYED_FLIGHTS AS BEGIN	Procedure created.	0
2	0.05	BEGIN PR_GOOD_DELAYED_FLIGHTS; END;	Statement processed.	1
3	0.10	CREATE OR REPLACE PROCEDURE PR_GOOD_WEATHER_DATA AS BEGIN	Procedure created.	0
4	0.09	BEGIN PR_GOOD_WEATHER_DATA; END;	Statement processed.	1

Below the table, it says "Download" and "row(s) 1 - 4 of 4".

GOOD\_DELAYED\_FLIGHTS

Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries	
Query	Count Rows	Insert Row	Load Data										
EDIT	DELAYED_FLIGHT_ID	FLIGHT_NUMBER	TAIL_NUMBER	AIR_SYSTEM_DELAY	SECURITY_DELAY	AIRLINE_DELAY	LATE_AIRCRAFT_DELAY	WEATHER_DELAY	FLIGHTS_AIRLINE	DATA_SOURCE			
	1166	2162	N947UW	10	0	0	5	0	US	flights_2018			
	1170	2152	N967UW	21	0	0	0	5	US	flights_2018			
	1172	2686	N892AT	56	0	0	0	0	DL	flights_2018			
	1174	2692	N953AT	65	0	10	0	0	DL	flights_2018			
	1175	2680	N989AT	16	0	0	9	0	DL	flights_2018			
	1178	2164	N949UW	11	0	0	9	0	US	flights_2018			
	1180	2688	N927AT	53	0	9	0	0	DL	flights_2018			
	1183	2680	N919AT	5	0	0	10	0	DL	flights_2018			
	1186	2670	N927AT	40	0	0	0	0	DL	flights_2018			
	1195	2686	N923AT	10	0	1	9	0	DL	flights_2018			
	1196	2690	N919AT	17	0	0	9	0	DL	flights_2018			
	1197	2690	N922AT	47	0	0	0	10	DL	flights_2018			
	1199	5074	N76IND	22	0	5	0	0	EV	flights_2018			
	1200	2690	N927AT	54	0	0	10	0	DL	flights_2018			
	1201	2158	N948UW	16	0	0	10	0	US	flights_2018			

Download

row(s) 1 - 15 of 43 [Next ▶](#)[evisapkota999@gmail.com](#) [ads\\_Level\\_b](#) [en](#)

Copyright © 1999-2020, Oracle and/or its affiliates. All rights reserved.

Application Express 20.2.0.0.20

GOOD\_WEATHER\_DATA

Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries	
Query	Count Rows	Insert Row	Load Data										
EDIT	WEATHER_DATA_ID	WEATHER_STATION	WEATHER_DATE	AWND	PRCP	TAVG							
	13	USW0004789	13/01/2017	15.2	0	32							
	14	USW0004789	14/01/2017	11.6	0	24							
	15	USW0004789	15/01/2017	9.84	0	31							
	16	USW0004789	16/01/2017	18.3	0	34							
	17	USW0004789	17/01/2017	10.5	0	23							
	18	USW0004789	18/01/2017	6.26	1.2	36							
	19	USW0004789	19/01/2017	15.2	0	40							
	20	USW0004789	20/01/2017	12.1	0	39							
	21	USW0004789	21/01/2017	6.04	0	32							
	22	USW0004789	22/01/2017	8.05	0	35							
	23	USW0004789	23/01/2017	10.5	0	33							
	24	USW0004789	24/01/2017	11.4	.75	35							
	25	USW0004789	25/01/2017	15	0	39							
	26	USW0004789	26/01/2017	18.3	.33	28							
	27	USW0004789	27/01/2017	19	.31	25							

Download

row(s) 1 - 15 of 450 [Next ▶](#)[evisapkota999@gmail.com](#) [ads\\_Level\\_b](#) [en](#)

Copyright © 1999-2020, Oracle and/or its affiliates. All rights reserved.

Application Express 20.2.0.0.20

## Transformation: Bad Data Cleaning and Insert into Clean Tables

As cancelled flights did not have any good/clean data, no data has been inserted into GOOD\_CANCELLED\_FLIGHTS.

All the clean data has been inserted into the good tables however the bad data inserted into the bad tables from staging tables have not yet been cleaned. Hence, the following code was used to clean the

bad data from bad tables and to insert those clean data to good table. The code also has updated bad table's error status and resolution date.

N.B. You can also find these codes in a separate file: 'Task 3 - Transformation - Bad Data Cleaning and Insert Into Good Table' in the attached folder: 'SQL Scripts'

```
-- PL/SQL CODES TO IDENTIFY BAD DATA CLEAN THEM AND INSERT INTO GOOD TABLE

-- PROCEDURE TO IDENTIFY BAD DATA, CLEAN THEM AND INSERT CLEAN DATA INTO GOOD_DELAYED_FLIGHTS
-- ALSO UPDATE BAD_DELAYED_FLIGHTS ON ERROR_STATUS AND RESOLUTION_DATE
CREATE OR REPLACE PROCEDURE PR_CLEAN_BAD_DELAYED_FLIGHTS
AS
    V_AIRSYSTEMDELAY NUMBER;
    V_SECURITYDELAY NUMBER;
    V_AIRLINEDELAY NUMBER;
    V_LATEAIRCRAFTDELAY NUMBER;
    V_WEATHERDELAY NUMBER;
    V_FLIGHTSAIRLINE VARCHAR(120);
    CURSOR CUR_DELAYED_FLIGHTS IS SELECT * FROM BAD_DELAYED_FLIGHTS;

BEGIN
    FOR C IN CUR_DELAYED_FLIGHTS LOOP
        V_AIRSYSTEMDELAY := '';
        V_SECURITYDELAY := '';
        V_AIRLINEDELAY := '';
        V_LATEAIRCRAFTDELAY := '';
        V_WEATHERDELAY := '';
        V_FLIGHTSAIRLINE := '';

        -- CLEANING THE BAD DATA
        IF C.AIR_SYSTEM_DELAY IS NOT NULL THEN
            V_AIRSYSTEMDELAY := C.AIR_SYSTEM_DELAY;
        ELSE
            V_AIRSYSTEMDELAY := 0;
        END IF;

        IF C.SECURITY_DELAY IS NOT NULL THEN
            V_SECURITYDELAY := C.SECURITY_DELAY;
        ELSE
            V_SECURITYDELAY := 0;
        END IF;
    END LOOP;

```

```

IF C.AIRLINE_DELAY IS NOT NULL THEN
    V_AIRLINEDELAY := C.AIRLINE_DELAY;
ELSE
    V_AIRLINEDELAY := 0;
END IF;

IF C.LATE_AIRCRAFT_DELAY IS NOT NULL THEN
    V_LATEAIRCRAFTDELAY := C.LATE_AIRCRAFT_DELAY;
ELSE
    V_LATEAIRCRAFTDELAY := 0;
END IF;

IF C.WEATHER_DELAY IS NOT NULL THEN
    V_WEATHERDELAY := C.WEATHER_DELAY;
ELSE
    V_WEATHERDELAY := 0;
END IF;

IF C.FLIGHTS_AIRLINE IS NOT NULL THEN
    V_FLIGHTSAIRLINE := C.FLIGHTS_AIRLINE;
ELSE
    V_FLIGHTSAIRLINE := 'UNKNOWN';
END IF;

-- INSERTING THE CLEAN DATA
MERGE INTO GOOD_DELAYED_FLIGHTS GDF
USING (SELECT * FROM BAD_DELAYED_FLIGHTS WHERE DELAYED_FLIGHT_ID=C.DELAYED_FLIGHT_ID) BDF
ON (GDF.DELAYED_FLIGHT_ID = BDF.DELAYED_FLIGHT_ID)

WHEN MATCHED
THEN UPDATE SET
    GDF.FLIGHT_NUMBER = BDF.FLIGHT_NUMBER,
    GDF.TAIL_NUMBER = BDF.TAIL_NUMBER,
    GDF.DATA_SOURCE = BDF.DATA_SOURCE
WHEN NOT MATCHED
THEN
    INSERT (DELAYED_FLIGHT_ID, FLIGHT_NUMBER, TAIL_NUMBER, AIR_SYSTEM_DELAY, SECURITY_DELAY, AIRLINE_DELAY, LATE_AIRCRAFT_DELAY, WEATHER_DELAY, FLIGHTS_AIRLINE, DATA_SOURCE)
        VALUES (BDF.DELAYED_FLIGHT_ID, BDF.FLIGHT_NUMBER, BDF.TAIL_NUMBER, V_AIRSYSTEMDELAY, V_SECURITYDELAY, V_AIRLINEDELAY, V_LATEAIRCRAFTDELAY, V_WEATHERDELAY, V_FLIGHTSAIRLINE, BDF.DATA_SOURCE);

-- UPDATING BAD_DELAYED_FLIGHTS TABLE
UPDATE BAD_DELAYED_FLIGHTS SET ERROR_STATUS = 'RESOLVED', RESOLUTION_DATE = SYSDATE WHERE DELAYED_FLIGHT_ID = C.DELAYED_FLIGHT_ID;
END LOOP;

```

```

END;
/

-- EXECUTE PROCEDURE PR_CLEAN_BAD_DELAYED_FLIGHTS
BEGIN
    PR_CLEAN_BAD_DELAYED_FLIGHTS;
END;
/


-----



-- PROCEDURE TO IDENTIFY BAD DATA, CLEAN THEM AND INSERT CLEAN DATA INTO GOOD_CANCELLED_FLIGHTS
-- ALSO UPDATE BAD_CANCELLED_FLIGHTS ON ERROR_STATUS AND RESOLUTION_DATE
CREATE OR REPLACE PROCEDURE PR_CLEAN_BAD_CANCELLED_FLIGHTS
AS
    V_TAILNUMBER VARCHAR(20);
    V_FLIGHTSAIRLINE VARCHAR(20);
    V_CANCELLATIONREASON VARCHAR(120);
    CURSOR CUR_CANCELLED_FLIGHTS IS SELECT * FROM BAD_CANCELLED_FLIGHTS;

BEGIN

    FOR C IN CUR_CANCELLED_FLIGHTS LOOP
        V_TAILNUMBER := '';
        V_FLIGHTSAIRLINE := '';
        V_CANCELLATIONREASON := '';

        -- CLEANING THE BAD DATA
        IF C.TAIL_NUMBER IS NOT NULL THEN
            V_TAILNUMBER := C.TAIL_NUMBER;
        ELSE
            V_TAILNUMBER := 'UNKNOWN';
        END IF;

        IF C.FLIGHTS_AIRLINE IS NOT NULL THEN
            V_FLIGHTSAIRLINE := C.FLIGHTS_AIRLINE;
        ELSE
            V_FLIGHTSAIRLINE := 'UNKNOWN';
        END IF;

        IF C.CANCELLATION_REASON = 'A' THEN
            V_CANCELLATIONREASON := 'ADVERSE WEATHER CONDITION';
        ELSIF C.CANCELLATION_REASON = 'B' THEN
            V_CANCELLATIONREASON := 'AIR TRAFFIC RESTRICTIONS';
        END IF;
    END LOOP;
END;
/

```

```

    ELSE
        V_CANCELLATIONREASON := 'AIR SYSTEM / MECHANICAL ISSUES';
    END IF;

    MERGE INTO GOOD_CANCELLED_FLIGHTS GCF
    USING (SELECT * FROM BAD_CANCELLED_FLIGHTS WHERE CANCELLED_FLIGHT_ID = C.CANCELLED_FLIGHT_ID) BCF
    ON (GCF.CANCELLED_FLIGHT_ID = BCF.CANCELLED_FLIGHT_ID)

    -- INSERTING INTO GOOD_CANCELLED_FLIGHTS
    WHEN MATCHED
        THEN UPDATE SET
            GCF.FLIGHT_NUMBER = BCF.FLIGHT_NUMBER

    WHEN NOT MATCHED
        THEN
            INSERT (CANCELLED_FLIGHT_ID, FLIGHT_NUMBER, TAIL_NUMBER, FLIGHTS_AIRLINE, CANCELLATION_REASON, DATA_SOURCE)
            VALUES (BCF.CANCELLED_FLIGHT_ID, BCF.FLIGHT_NUMBER, V_TAILNUMBER, V_FLIGHTSAIRLINE, V_CANCELLATIONREASON, BCF.DATA_SOURCE);

    -- UPDATING BAD_CANCELLED_FLIGHTS TABLE
    UPDATE BAD_CANCELLED_FLIGHTS SET ERROR_STATUS = 'RESOLVED', RESOLUTION_DATE = SYSDATE WHERE CANCELLED_FLIGHT_ID = C.CANCELLED_FLIGHT_ID;
    END LOOP;
END;
/

-- EXECUTE PROCEDURE PR_CLEAN_BAD_CANCELLED_FLIGHTS
BEGIN
    PR_CLEAN_BAD_CANCELLED_FLIGHTS;
END;
/
-----



-- PROCEDURE TO IDENTIFY BAD DATA, CLEAN THEM AND INSERT CLEAN DATA INTO GOOD_WEATHER_DATA
-- ALSO UPDATE BAD_WEATHER_DATA ON ERROR_STATUS AND RESOLUTION_DATE
CREATE OR REPLACE PROCEDURE PR_CLEAN_BAD_WEATHER_DATA
AS
    V_WEATHERDATE DATE;
    CURSOR CUR_WEATHER_DATA IS SELECT * FROM BAD_WEATHER_DATA;

BEGIN
    FOR C IN CUR_WEATHER_DATA LOOP

```

```

-- CLEANING THE BAD DATA
V_WEATHERDATE := TO_DATE(C.WEATHER_DATE, 'DD/MM/YYYY');

-- INSERTING INTO GOOD_WEATHER_DATA
MERGE INTO GOOD_WEATHER_DATA GWD
USING (SELECT * FROM BAD_WEATHER_DATA WHERE WEATHER_DATA_ID = C.WEATHER_DATA_ID) BWD
ON (GWD.WEATHER_DATA_ID = BWD.WEATHER_DATA_ID)

WHEN MATCHED
THEN UPDATE SET
    GWD.WEATHER_STATION = BWD.WEATHER_STATION

WHEN NOT MATCHED
THEN
    INSERT(WEATHER_DATA_ID, WEATHER_STATION, WEATHER_DATE, AWND, PRCP, TAVG)
    VALUES(BWD.WEATHER_DATA_ID, BWD.WEATHER_STATION, V_WEATHERDATE, BWD.AWND, BWD.PRCP, BWD.TA
VG);

-- UPDATING BAD_WEATHER_DATA TABLE
UPDATE BAD_WEATHER_DATA SET ERROR_STATUS = 'RESOLVED', RESOLUTION_DATE = SYSDATE WHERE WEATHER_DATA_ID = C.WEATHER_DATA_ID;

V_WEATHERDATE := '';

END LOOP;
END;
/

-- EXECUTE PROCEDURE PR_CLEAN_BAD_WEATHER_DATA
BEGIN
    PR_CLEAN_BAD_WEATHER_DATA;
END;
/
----- END OF BAD DATA IDENTIFICATION, CLEANING AND INSERT INTO GOOD TABLES -----
-----
```

Screenshots: Procedure Statements Run Report along with Clean data Inserted into the good tables, bad tables updated and total number of rows in all three tables compared to staging tables.

The screenshot shows the 'Run Script' dialog from Oracle Application Express. The dialog title is 'Run Script' with an information icon. It displays the following details:

Script Name	ETL_CleaningData
Created	on 12/16/2020 01:33:41 AM by AV.SAPKOTA999@GMAIL.COM
Updated	on 12/16/2020 04:27:51 AM by AV.SAPKOTA999@GMAIL.COM
Number of Statements	6
Script Size in Bytes	5,704

At the bottom, there are 'Cancel' and 'Run Now' buttons.

The screenshot shows the 'Summary' report for the ETL\_CleaningData script. The report header includes the script name, status (Complete), and a link to 'View'. The summary table has columns: Number, Elapsed, Statement, Feedback, and Rows.

Number	Elapsed	Statement	Feedback	Rows
1	0.15	CREATE OR REPLACE PROCEDURE PR_CLEAN_BAD_DELAYED_FLIGHTS AS	Procedure created.	0
2	0.78	BEGIN PR_CLEAN_BAD_DELAYED_FLIGHTS; END;	Statement processed.	1
3	0.02	CREATE OR REPLACE PROCEDURE PR_CLEAN_BAD_CANCELLED_FLIGHTS AS	Procedure created.	0
4	0.06	BEGIN PR_CLEAN_BAD_CANCELLED_FLIGHTS; END;	Statement processed.	1
5	0.01	CREATE OR REPLACE PROCEDURE PR_CLEAN_BAD_WEATHER_DATA AS	Procedure created.	0
6	0.09	BEGIN PR_CLEAN_BAD_WEATHER_DATA; END;	Statement processed.	1

Below the table, there are three summary statistics: 6 Statements Processed, 6 Successful, and 0 With Errors.

GOOD\_DELAYED\_FLIGHTS

Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries	
Query	Count Rows	Insert Row	Load Data										
EDIT	DELAYED_FLIGHT_ID	FLIGHT_NUMBER	TAIL_NUMBER	AIR_SYSTEM_DELAY	SECURITY_DELAY	AIRLINE_DELAY	LATE_AIRCRAFT_DELAY	WEATHER_DELAY	FLIGHTS_AIRLINE	DATA_SOURCE			
1166	2162	N447UW	10	0	0	5	0	0	US	flights_2018			
1170	2152	N967UW	21	0	0	0	5	0	US	flights_2018			
1172	2686	N892AT	56	0	0	0	0	0	DL	flights_2018			
1174	2692	N953AT	65	0	10	0	0	0	DL	flights_2018			
1175	2680	N989AT	16	0	0	9	0	0	DL	flights_2018			
1178	2164	N949UW	11	0	0	9	0	0	US	flights_2018			
1180	2688	N927AT	53	0	9	0	0	0	DL	flights_2018			
1183	2680	N999AT	5	0	0	10	0	0	DL	flights_2018			
1186	2670	N923AT	40	0	0	0	0	0	DL	flights_2018			
1195	2686	N923AT	10	0	1	9	0	0	DL	flights_2018			
1196	2690	N910AT	17	0	0	9	0	0	DL	flights_2018			
1197	2690	N922AT	47	0	0	0	10	0	DL	flights_2018			
1199	5074	N76IND	22	0	5	0	0	0	EV	flights_2018			
1200	2690	N927AT	54	0	0	10	0	0	DL	flights_2018			
1201	2158	N948UW	16	0	0	10	0	0	US	flights_2018			

Download

row(s) 1 - 15 of more than 500 [Next ▶](#)[e.sapkota999@gmail.com](#) [edt\\_Level\\_6](#) [en](#)

Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved.

Application Express 20.2.0.0.20

BAD\_DELAYED\_FLIGHTS

Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries	
Query	Count Rows	Insert Row	Load Data										
EDIT	BAD_DELAYED_FLIGHT_ID	DELAYED_FLIGHT_ID	FLIGHT_NUMBER	TAIL_NUMBER	AIR_SYSTEM_DELAY	SECURITY_DELAY	AIRLINE_DELAY	LATE_AIRCRAFT_DELAY	WEATHER_DELAY	FLIGHTS_AIRLINE	DATA_SOURCE	ERROR_DESCRIPTION	ERROR_STATUS
121	121	121	84	N8KHAA	-	-	-	-	-	-	flights_2017	NULL VALUES IN AIR_SYSTEM_DELAY / SECURITY_DELAY / AIRLINE_DELAY / LATE_AIRCRAFT_DELAY / WEATHER_DELAY / FLIGHTS_AIRLINE	RESOLVED
122	122	122	291	N8BCAA	-	-	-	-	-	-	flights_2017	NULL VALUES IN AIR_SYSTEM_DELAY / SECURITY_DELAY / AIRLINE_DELAY / LATE_AIRCRAFT_DELAY / WEATHER_DELAY / FLIGHTS_AIRLINE	RESOLVED
123	123	123	1653	N3JLAA	-	-	-	-	-	-	flights_2017	NULL VALUES IN AIR_SYSTEM_DELAY / SECURITY_DELAY / AIRLINE_DELAY / LATE_AIRCRAFT_DELAY / WEATHER_DELAY / FLIGHTS_AIRLINE	RESOLVED
124	124	124	1096	N8ACAA	3	0	29	0	0	-	flights_2017	NULL VALUES IN AIR_SYSTEM_DELAY / SECURITY_DELAY / AIRLINE_DELAY / LATE_AIRCRAFT_DELAY / WEATHER_DELAY / FLIGHTS_AIRLINE	RESOLVED

[e.sapkota999@gmail.com](#) [edt\\_Level\\_6](#) [en](#)

Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved.

Application Express 20.2.0.0.20

STAGING_DELAYED_FLIGHTS											+ v	
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries
<a href="#">Query</a> <a href="#">Count Rows</a> <a href="#">Insert Row</a> <a href="#">Load Data</a>												
1,322												
Rows												

GOOD_DELAYED_FLIGHTS											+ v	
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries
<a href="#">Query</a> <a href="#">Count Rows</a> <a href="#">Insert Row</a> <a href="#">Load Data</a>												
1,322												
Rows												

GOOD_CANCELLED_FLIGHTS											+ v																																																																																																																
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries																																																																																																															
<a href="#">Query</a> <a href="#">Count Rows</a> <a href="#">Insert Row</a> <a href="#">Load Data</a>																																																																																																																											
<table border="1"> <thead> <tr><th>EDIT</th><th>CANCELLED_FLIGHT_ID</th><th>FLIGHT_NUMBER</th><th>TAIL_NUMBER</th><th>FLIGHTS_AIRLINE</th><th>CANCELLATION_REASON</th><th>DATA_SOURCE</th></tr> </thead> <tbody> <tr><td></td><td>1</td><td>291</td><td>UNKNOWN</td><td>UNKNOWN</td><td>AIR TRAFFIC RESTRICTIONS</td><td>flights_2017</td></tr> <tr><td></td><td>2</td><td>2314</td><td>UNKNOWN</td><td>UNKNOWN</td><td>AIR TRAFFIC RESTRICTIONS</td><td>flights_2017</td></tr> <tr><td></td><td>3</td><td>84</td><td>UNKNOWN</td><td>UNKNOWN</td><td>AIR TRAFFIC RESTRICTIONS</td><td>flights_2017</td></tr> <tr><td></td><td>4</td><td>1653</td><td>UNKNOWN</td><td>UNKNOWN</td><td>AIR TRAFFIC RESTRICTIONS</td><td>flights_2017</td></tr> <tr><td></td><td>5</td><td>193</td><td>UNKNOWN</td><td>UNKNOWN</td><td>AIR TRAFFIC RESTRICTIONS</td><td>flights_2017</td></tr> <tr><td></td><td>6</td><td>1096</td><td>NDAAA</td><td>UNKNOWN</td><td>AIR TRAFFIC RESTRICTIONS</td><td>flights_2017</td></tr> <tr><td></td><td>7</td><td>2314</td><td>NDAAA</td><td>UNKNOWN</td><td>AIR TRAFFIC RESTRICTIONS</td><td>flights_2017</td></tr> <tr><td></td><td>8</td><td>84</td><td>NHSAA</td><td>UNKNOWN</td><td>AIR TRAFFIC RESTRICTIONS</td><td>flights_2017</td></tr> <tr><td></td><td>9</td><td>1653</td><td>NBMAA</td><td>UNKNOWN</td><td>AIR TRAFFIC RESTRICTIONS</td><td>flights_2017</td></tr> <tr><td></td><td>10</td><td>193</td><td>UNKNOWN</td><td>UNKNOWN</td><td>AIR TRAFFIC RESTRICTIONS</td><td>flights_2017</td></tr> <tr><td></td><td>11</td><td>1096</td><td>NBLHAA</td><td>UNKNOWN</td><td>AIR TRAFFIC RESTRICTIONS</td><td>flights_2017</td></tr> <tr><td></td><td>12</td><td>291</td><td>NBBMAA</td><td>UNKNOWN</td><td>AIR TRAFFIC RESTRICTIONS</td><td>flights_2017</td></tr> <tr><td></td><td>13</td><td>2314</td><td>NBRAA</td><td>UNKNOWN</td><td>AIR TRAFFIC RESTRICTIONS</td><td>flights_2017</td></tr> <tr><td></td><td>14</td><td>2314</td><td>NSAAA</td><td>UNKNOWN</td><td>AIR TRAFFIC RESTRICTIONS</td><td>flights_2017</td></tr> <tr><td></td><td>15</td><td>84</td><td>NELAA</td><td>UNKNOWN</td><td>AIR TRAFFIC RESTRICTIONS</td><td>flights_2017</td></tr> </tbody></table>												EDIT	CANCELLED_FLIGHT_ID	FLIGHT_NUMBER	TAIL_NUMBER	FLIGHTS_AIRLINE	CANCELLATION_REASON	DATA_SOURCE		1	291	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017		2	2314	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017		3	84	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017		4	1653	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017		5	193	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017		6	1096	NDAAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017		7	2314	NDAAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017		8	84	NHSAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017		9	1653	NBMAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017		10	193	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017		11	1096	NBLHAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017		12	291	NBBMAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017		13	2314	NBRAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017		14	2314	NSAAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017		15	84	NELAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017
EDIT	CANCELLED_FLIGHT_ID	FLIGHT_NUMBER	TAIL_NUMBER	FLIGHTS_AIRLINE	CANCELLATION_REASON	DATA_SOURCE																																																																																																																					
	1	291	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017																																																																																																																					
	2	2314	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017																																																																																																																					
	3	84	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017																																																																																																																					
	4	1653	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017																																																																																																																					
	5	193	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017																																																																																																																					
	6	1096	NDAAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017																																																																																																																					
	7	2314	NDAAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017																																																																																																																					
	8	84	NHSAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017																																																																																																																					
	9	1653	NBMAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017																																																																																																																					
	10	193	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017																																																																																																																					
	11	1096	NBLHAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017																																																																																																																					
	12	291	NBBMAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017																																																																																																																					
	13	2314	NBRAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017																																																																																																																					
	14	2314	NSAAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017																																																																																																																					
	15	84	NELAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017																																																																																																																					
<a href="#">Download</a>																																																																																																																											
row(s) 1 - 15 of 109 <a href="#">Next ▶</a>																																																																																																																											

av.sapkota999@gmail.com edit\_level\_b en Copyright © 1999-2020, Oracle and/or its affiliates. All rights reserved. Application Express 20.2.0.0.20

BAD_CANCELLED_FLIGHTS											+ v																																																																																																																																																																																
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries																																																																																																																																																																															
<a href="#">Query</a> <a href="#">Count Rows</a> <a href="#">Insert Row</a> <a href="#">Load Data</a>																																																																																																																																																																																											
<table border="1"> <thead> <tr><th>EDIT</th><th>BAD_CANCELLED_FLIGHT_ID</th><th>CANCELLED_FLIGHT_ID</th><th>FLIGHT_NUMBER</th><th>TAIL_NUMBER</th><th>FLIGHTS_AIRLINE</th><th>CANCELLATION_REASON</th><th>DATA_SOURCE</th><th>ERROR_DESCRIPTION</th><th>ERROR_STATUS</th><th>RESOLUTION_DATE</th></tr> </thead> <tbody> <tr><td></td><td>1</td><td>1</td><td>291</td><td>-</td><td>-</td><td>B</td><td>flights_2017</td><td>NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON</td><td>RESOLVED</td><td>12/16/2020</td></tr> <tr><td></td><td>2</td><td>2</td><td>2314</td><td>-</td><td>-</td><td>B</td><td>flights_2017</td><td>NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON</td><td>RESOLVED</td><td>12/16/2020</td></tr> <tr><td></td><td>3</td><td>3</td><td>84</td><td>-</td><td>-</td><td>B</td><td>flights_2017</td><td>NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON</td><td>RESOLVED</td><td>12/16/2020</td></tr> <tr><td></td><td>4</td><td>4</td><td>1653</td><td>-</td><td>-</td><td>B</td><td>flights_2017</td><td>NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON</td><td>RESOLVED</td><td>12/16/2020</td></tr> <tr><td></td><td>5</td><td>5</td><td>193</td><td>-</td><td>-</td><td>B</td><td>flights_2017</td><td>NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON</td><td>RESOLVED</td><td>12/16/2020</td></tr> <tr><td></td><td>6</td><td>6</td><td>1096</td><td>NDAAA</td><td>-</td><td>B</td><td>flights_2017</td><td>NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON</td><td>RESOLVED</td><td>12/16/2020</td></tr> <tr><td></td><td>7</td><td>7</td><td>2314</td><td>NDAAA</td><td>-</td><td>B</td><td>flights_2017</td><td>NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON</td><td>RESOLVED</td><td>12/16/2020</td></tr> <tr><td></td><td>8</td><td>8</td><td>84</td><td>NHSAA</td><td>-</td><td>B</td><td>flights_2017</td><td>NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON</td><td>RESOLVED</td><td>12/16/2020</td></tr> <tr><td></td><td>9</td><td>9</td><td>1653</td><td>NBMAA</td><td>-</td><td>B</td><td>flights_2017</td><td>NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON</td><td>RESOLVED</td><td>12/16/2020</td></tr> <tr><td></td><td>10</td><td>10</td><td>193</td><td>-</td><td>-</td><td>B</td><td>flights_2017</td><td>NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON</td><td>RESOLVED</td><td>12/16/2020</td></tr> <tr><td></td><td>11</td><td>11</td><td>1096</td><td>NBLHAA</td><td>-</td><td>B</td><td>flights_2017</td><td>NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON</td><td>RESOLVED</td><td>12/16/2020</td></tr> <tr><td></td><td>12</td><td>12</td><td>291</td><td>NBBMAA</td><td>-</td><td>B</td><td>flights_2017</td><td>NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON</td><td>RESOLVED</td><td>12/16/2020</td></tr> <tr><td></td><td>13</td><td>13</td><td>2314</td><td>NBRAA</td><td>-</td><td>B</td><td>flights_2017</td><td>NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON</td><td>RESOLVED</td><td>12/16/2020</td></tr> <tr><td></td><td>14</td><td>14</td><td>2314</td><td>NSAAA</td><td>-</td><td>B</td><td>flights_2017</td><td>NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON</td><td>RESOLVED</td><td>12/16/2020</td></tr> <tr><td></td><td>15</td><td>15</td><td>84</td><td>NELAA</td><td>-</td><td>B</td><td>flights_2017</td><td>NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON</td><td>RESOLVED</td><td>12/16/2020</td></tr> </tbody></table>												EDIT	BAD_CANCELLED_FLIGHT_ID	CANCELLED_FLIGHT_ID	FLIGHT_NUMBER	TAIL_NUMBER	FLIGHTS_AIRLINE	CANCELLATION_REASON	DATA_SOURCE	ERROR_DESCRIPTION	ERROR_STATUS	RESOLUTION_DATE		1	1	291	-	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020		2	2	2314	-	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020		3	3	84	-	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020		4	4	1653	-	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020		5	5	193	-	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020		6	6	1096	NDAAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020		7	7	2314	NDAAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020		8	8	84	NHSAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020		9	9	1653	NBMAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020		10	10	193	-	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020		11	11	1096	NBLHAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020		12	12	291	NBBMAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020		13	13	2314	NBRAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020		14	14	2314	NSAAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020		15	15	84	NELAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020
EDIT	BAD_CANCELLED_FLIGHT_ID	CANCELLED_FLIGHT_ID	FLIGHT_NUMBER	TAIL_NUMBER	FLIGHTS_AIRLINE	CANCELLATION_REASON	DATA_SOURCE	ERROR_DESCRIPTION	ERROR_STATUS	RESOLUTION_DATE																																																																																																																																																																																	
	1	1	291	-	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020																																																																																																																																																																																	
	2	2	2314	-	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020																																																																																																																																																																																	
	3	3	84	-	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020																																																																																																																																																																																	
	4	4	1653	-	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020																																																																																																																																																																																	
	5	5	193	-	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020																																																																																																																																																																																	
	6	6	1096	NDAAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020																																																																																																																																																																																	
	7	7	2314	NDAAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020																																																																																																																																																																																	
	8	8	84	NHSAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020																																																																																																																																																																																	
	9	9	1653	NBMAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020																																																																																																																																																																																	
	10	10	193	-	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020																																																																																																																																																																																	
	11	11	1096	NBLHAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020																																																																																																																																																																																	
	12	12	291	NBBMAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020																																																																																																																																																																																	
	13	13	2314	NBRAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020																																																																																																																																																																																	
	14	14	2314	NSAAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020																																																																																																																																																																																	
	15	15	84	NELAA	-	B	flights_2017	NULL VALUES IN TAIL_NUMBER / FLIGHTS_AIRLINE AND UNCLEAR CANCELLATION_REASON	RESOLVED	12/16/2020																																																																																																																																																																																	
<a href="#">Download</a>																																																																																																																																																																																											
row(s) 1 - 15 of 109 <a href="#">Next ▶</a>																																																																																																																																																																																											

av.sapkota999@gmail.com edit\_level\_b en Copyright © 1999-2020, Oracle and/or its affiliates. All rights reserved. Application Express 20.2.0.0.20

Good_Weather_Data							
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults
Query	Count Rows	Insert Row	Load Data				
EDIT	WEATHER_DATA_ID	WEATHER_STATION	WEATHER_DATE	AWND	PRCP	TAVG	
13	USW00094789	01/13/2017	15.2	0	.32		
14	USW00094789	01/14/2017	11.6	0	.24		
15	USW00094789	01/15/2017	9.84	0	.31		
16	USW00094789	01/16/2017	18.3	0	.34		
17	USW00094789	01/17/2017	10.5	0	.23		
18	USW00094789	01/18/2017	6.26	1.62	.36		
19	USW00094789	01/19/2017	15.2	0	.40		
20	USW00094789	01/20/2017	12.1	0	.39		
21	USW00094789	01/21/2017	6.04	0	.32		
22	USW00094789	01/22/2017	8.05	0	.35		
23	USW00094789	01/23/2017	10.5	0	.33		
24	USW00094789	01/24/2017	11.4	.75	.35		
25	USW00094789	01/25/2017	15	0	.39		
26	USW00094789	01/26/2017	18.3	.55	.28		
27	USW00094789	01/27/2017	19	.31	.25		

BAD_WEATHER_DATA												
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries
Query	Count Rows	Insert Row	Load Data									
EDIT	BAD_WEATHER_DATA_ID	WEATHER_DATA_ID	WEATHER_STATION	WEATHER_DATE	AWND	PRCP	TAVG		ERROR_DESCRIPTION	ERROR_STATUS	RESOLUTION_DATE	
218	558	USW00094789	11/7/2018	15.7	0	.56		INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY	RESOLVED	12/16/2020		
219	559	USW00094789	12/7/2018	6.04	0	.61		INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY	RESOLVED	12/16/2020		
220	579	USW00094789	1/8/2018	12.1	.45	.57		INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY	RESOLVED	12/16/2020		
221	580	USW00094789	2/8/2018	12.8	.09	.54		INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY	RESOLVED	12/16/2020		
222	581	USW00094789	3/8/2018	11	.05	.56		INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY	RESOLVED	12/16/2020		
223	582	USW00094789	4/8/2018	20.4	0	.57		INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY	RESOLVED	12/16/2020		
224	583	USW00094789	5/8/2018	17.5	0	.49		INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY	RESOLVED	12/16/2020		
225	584	USW00094789	6/8/2018	12.8	0	.48		INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY	RESOLVED	12/16/2020		
226	585	USW00094789	7/8/2018	9.17	0	.56		INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY	RESOLVED	12/16/2020		
227	586	USW00094789	8/8/2018	8.05	0	.51		INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY	RESOLVED	12/16/2020		
228	587	USW00094789	9/8/2018	6.93	0	.51		INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY	RESOLVED	12/16/2020		
229	588	USW00094789	10/8/2018	7.61	0	.55		INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY	RESOLVED	12/16/2020		
230	589	USW00094789	11/8/2018	10.1	0	.61		INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY	RESOLVED	12/16/2020		
231	590	USW00094789	12/8/2018	5.37	0	.67		INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY	RESOLVED	12/16/2020		
232	510	USW00094789	1/9/2018	12.8	0	.66		INCONSISTENT DATE VALUES CHANGE TO DD/MM/YYYY	RESOLVED	12/16/2020		

The screenshot shows two database tables side-by-side. The top table is titled 'STAGING\_WEATHER\_DATA' and the bottom table is titled 'GOOD\_WEATHER\_DATA'. Both tables have a single row with the value '729' under the 'Rows' column. The interface includes standard navigation buttons like 'Query', 'Count Rows', 'Insert Row', and 'Load Data'.

## Transformation: Into Transform Tables

In order to transform the clean data from good tables, transformation tables have to be created. Thus, three transformation table for delayed flights, cancelled flights and weather data have been created using following code:

N.B. You can also find these codes in a separate file: 'Task 3 - Transformation - Transformation Table Create SQL' in the attached folder: 'SQL Scripts'

```
----- TRANSFORMATION DATA TABLES -----  
  
-- ALL SQL AND PL/SQL CODE REQUIRED FOR TRANSFORMATION TABLE: DIM_DELAYED_FLIGHTS  
  
-- DROP TABLE TRANSFORM_DELAYED_FLIGHTS BEFORE CREATION  
DROP TABLE TRANSFORM_DELAYED_FLIGHTS;  
  
-- CREATE TABLE TRANSFORM_DELAYED_FLIGHTS  
CREATE TABLE TRANSFORM_DELAYED_FLIGHTS(  
    delayed_flight_id    INTEGER NOT NULL,  
    flight_number       INTEGER NOT NULL,  
    tail_number        VARCHAR(50) NOT NULL,  
    air_system_delay    INTEGER NOT NULL,  
    security_delay     INTEGER NOT NULL,  
    airline_delay      INTEGER NOT NULL,  
    late_aircraft_delay INTEGER NOT NULL,  
    weather_delay      INTEGER NOT NULL,  
    flights_airline    VARCHAR(20) NOT NULL,  
    Data_Source         VARCHAR(20) NOT NULL,  
  
    CONSTRAINT pk_TRANSFORM_DELAYED_FLIGHTS PRIMARY KEY (delayed_flight_id)
```

```

);

-----  

-----  

-- ALL SQL AND PL/SQL CODE REQUIRED FOR TRANSFORMATION TABLE: DIM_CANCELLED_FLIGHTS  

-----  

-----  

-- DROP TABLE TRANSFORM_CANCELLED_FLIGHTS BEFORE CREATION
DROP TABLE TRANSFORM_CANCELLED_FLIGHTS;  

-- CREATE TABLE TRANSFORM_CANCELLED_FLIGHTS
CREATE TABLE TRANSFORM_CANCELLED_FLIGHTS(
    cancelled_flight_id INTEGER NOT NULL,
    flight_number      INTEGER NOT NULL,
    tail_number        VARCHAR(50) NOT NULL,
    flights_airline   VARCHAR(20) NOT NULL,
    cancellation_reason VARCHAR(120) NOT NULL,
    Data_Source        VARCHAR(20) NOT NULL,  

    CONSTRAINT pk_TRANSFORM_CANCELLED_FLIGHTS PRIMARY KEY (cancelled_flight_id)
);  

-----  

-----  

-----  

-- ALL SQL AND PL/SQL CODE REQUIRED FOR TRANSFORMATION TABLE: DIM_WEATHER_DATA  

-----  

-----  

-- DROP TABLE TRANSFORM_WEATHER_DATA BEFORE CREATION
DROP TABLE TRANSFORM_WEATHER_DATA;  

-- CREATE TABLE TRANSFORM_WEATHER_DATA
CREATE TABLE TRANSFORM_WEATHER_DATA(
    weather_data_id INTEGER NOT NULL,
    weather_station VARCHAR(20) NOT NULL,
    weather_date     DATE NOT NULL,
    AWND      FLOAT(8) NOT NULL,
    PRCP      FLOAT(8) NOT NULL,
    TAVG      INTEGER NOT NULL,  

    CONSTRAINT pk_TRANSFORM_WEATHER_DATA PRIMARY KEY (weather_data_id)
);  

-----  

-----  

END OF TRANSFORMATION TABLE CREATION CODE -----

```

## Screenshots: Transformation Tables created run and report

The screenshot shows the 'Run Script' dialog in Oracle Application Express. The title bar says 'Run Script'. The main area has a large blue 'i' icon and the text 'Run Script'. Below it, a message reads: 'You have requested to run the following script. Please confirm your request.' A table displays the script details:

Script Name	ETL_Transform_Tables
Created	on 12/16/2020 10:37:30 AM by AV.SAPKOTA999@GMAIL.COM
Updated	-
Number of Statements	6
Script Size in Bytes	3,557

Buttons at the bottom are 'Cancel' and 'Run Now'.

The screenshot shows the 'Results' page for the ETL\_Transform\_Tables script. The status is 'Complete'. The results table has columns: Number, Elapsed, Statement, Feedback, and Rows. The data is as follows:

Number	Elapsed	Statement	Feedback	Rows
1	0.59	DROP TABLE TRANSFORM_DELAYED_FLIGHTS	Table dropped.	0
2	0.11	CREATE TABLE TRANSFORM_DELAYED_FLIGHTS(delayed_flight_id INT)	Table created.	0
3	0.58	DROP TABLE TRANSFORM_CANCELLED_FLIGHTS	Table dropped.	0
4	0.07	CREATE TABLE TRANSFORM_CANCELLED_FLIGHTS(cancelled_flight_id INT)	Table created.	0
5	0.32	DROP TABLE TRANSFORM_WEATHER_DATA	Table dropped.	0
6	0.08	CREATE TABLE TRANSFORM_WEATHER_DATA(weather_data_id INT)	Table created.	0

Summary statistics at the bottom: 6 statements processed, 6 successful, 0 with errors.

Creation of transformation table is the prerequisites to insert data from clean tables. So, data were inserted into these transformation tables using the following code:

N.B. You can also find these codes in a separate file: 'Task 3 - Transformation - Insert Data into Transform Table PLSQL' in the attached folder: 'SQL Scripts'

```
-- PL/SQL CODES TO INSERT THE DATA FROM GOOD TABLES INTO TRANSFORMATION TABLES

-- PROCEDURE TO INSERT DATA FROM GOOD_DELAYED_FLIGHTS INTO TRANSFORM_DELAYED_FLIGHTS
CREATE OR REPLACE PROCEDURE PR_INSERT_TRANSFORM_DELAYED_FLIGHTS
AS
BEGIN
    -- MERGING ALL DATA INTO TRANSFORMATION TABLE
    MERGE INTO TRANSFORM_DELAYED_FLIGHTS TDF
    USING GOOD_DELAYED_FLIGHTS GDF
    ON (TDF.DELAYED_FLIGHT_ID = GDF.DELAYED_FLIGHT_ID)

    -- UPDATE DATA IF MATCHED
    WHEN MATCHED THEN
        UPDATE SET
            TDF.FLIGHT_NUMBER = GDF.FLIGHT_NUMBER,
            TDF.TAIL_NUMBER = GDF.TAIL_NUMBER,
            TDF.AIR_SYSTEM_DELAY = GDF.AIR_SYSTEM_DELAY,
            TDF.SECURITY_DELAY = GDF.SECURITY_DELAY,
            TDF.AIRLINE_DELAY = GDF.AIRLINE_DELAY,
            TDF.LATE_AIRCRAFT_DELAY = GDF.LATE_AIRCRAFT_DELAY,
            TDF.WEATHER_DELAY = GDF.WEATHER_DELAY,
            TDF.FLIGHTS_AIRLINE = GDF.FLIGHTS_AIRLINE,
            TDF.DATA_SOURCE = GDF.DATA_SOURCE

    -- INSERT DATA IF NOT MATCHED
    WHEN NOT MATCHED THEN
        INSERT (DELAYED_FLIGHT_ID, FLIGHT_NUMBER, TAIL_NUMBER, AIR_SYSTEM_DELAY, SECURITY_DELAY, AIRLINE_DELAY, LATE_AIRCRAFT_DELAY, WEATHER_DELAY, FLIGHTS_AIRLINE, DATA_SOURCE)
        VALUES (GDF.DELAYED_FLIGHT_ID, GDF.FLIGHT_NUMBER, GDF.TAIL_NUMBER, GDF.AIR_SYSTEM_DELAY, GDF.SECURITY_DELAY, GDF.AIRLINE_DELAY, GDF.LATE_AIRCRAFT_DELAY, GDF.WEATHER_DELAY, GDF.FLIGHTS_AIRLINE, GDF.DATA_SOURCE);

END;
/

-- EXECUTE PR_INSERT_TRANSFORM_DELAYED_FLIGHTS
BEGIN
    PR_INSERT_TRANSFORM_DELAYED_FLIGHTS;
```

```

END;
/
-----  

-----  

-- PROCEDURE TO INSERT DATA FROM GOOD_CANCELLED_FLIGHTS INTO TRANSFORM_CANCELLED_FLIGHTS
CREATE OR REPLACE PROCEDURE PR_INSERT_TRANSFORM_CANCELLED_FLIGHTS
AS
BEGIN
    -- MERGING ALL DATA INTO TRANSFORMATION TABLE
    MERGE INTO TRANSFORM_CANCELLED_FLIGHTS TCF
    USING GOOD_CANCELLED_FLIGHTS GCF
    ON (TCF.CANCELLED_FLIGHT_ID = GCF.CANCELLED_FLIGHT_ID)

        -- UPDATE DATA IF MATCHED
    WHEN MATCHED THEN
        UPDATE SET
            TCF.FLIGHT_NUMBER = GCF.FLIGHT_NUMBER,
            TCF.TAIL_NUMBER = GCF.TAIL_NUMBER,
            TCF.FLIGHTS_AIRLINE = GCF.FLIGHTS_AIRLINE,
            TCF.CANCELLATION_REASON = GCF.CANCELLATION_REASON,
            TCF.DATA_SOURCE = GCF.DATA_SOURCE

        -- INSERT DATA IF NOT MATCHED
    WHEN NOT MATCHED THEN
        INSERT (CANCELLED_FLIGHT_ID, FLIGHT_NUMBER, TAIL_NUMBER, FLIGHTS_AIRLINE, CANCELLATION_REASON, DATA_SOURCE)
        VALUES (GCF.CANCELLED_FLIGHT_ID, GCF.FLIGHT_NUMBER, GCF.TAIL_NUMBER, GCF.FLIGHTS_AIRLINE, GCF.CANCELLATION_REASON, GCF.DATA_SOURCE);

END;
/
-----  

-----  

-- EXECUTE PR_INSERT_TRANSFORM_CANCELLED_FLIGHTS
BEGIN
    PR_INSERT_TRANSFORM_CANCELLED_FLIGHTS;
END;
/
-----  

-----  

-- PROCEDURE TO INSERT DATA FROM GOOD_WEATHER_DATA INTO TRANSFORM_WEATHER_DATA
CREATE OR REPLACE PROCEDURE PR_INSERT_TRANSFORM_WEATHER_DATA
AS
BEGIN

```

```

-- MERGING ALL DATA INTO TRANSFORMATION TABLE
MERGE INTO TRANSFORM_WEATHER_DATA TWD
USING GOOD_WEATHER_DATA GWD
ON (TWD.WEATHER_DATA_ID = GWD.WEATHER_DATA_ID)

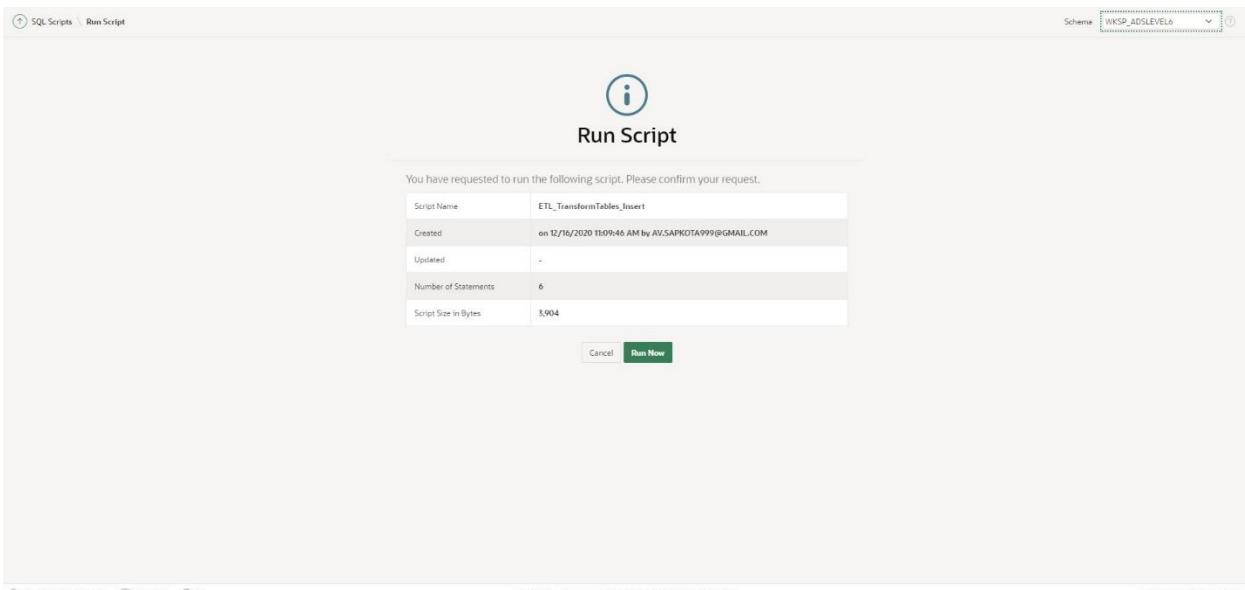
-- UPDATE DATA IF MATCHED
WHEN MATCHED THEN
UPDATE SET
    TWD.WEATHER_STATION = GWD.WEATHER_STATION,
    TWD.WEATHER_DATE = GWD.WEATHER_DATE,
    TWD.AWND = GWD.AWND,
    TWD.PRCP = GWD.PRCP,
    TWD.TAVG = GWD.TAVG

-- INSERT DATA IF NOT MATCHED
WHEN NOT MATCHED THEN
    INSERT (WEATHER_DATA_ID, WEATHER_STATION, WEATHER_DATE, AWND, PRCP, TAVG)
    VALUES (GWD.WEATHER_DATA_ID, GWD.WEATHER_STATION, GWD.WEATHER_DATE, GWD.AWND, GWD.PRCP, GWD.TAVG);

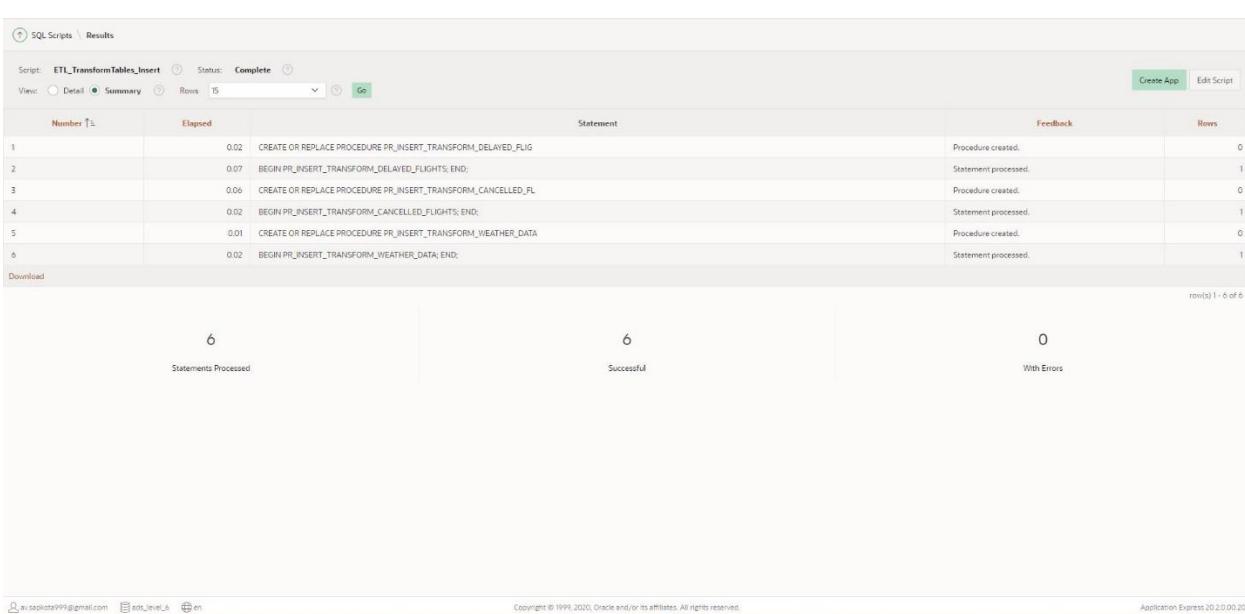
END;
/

-- EXECUTE PR_INSERT_TRANSFORM_WEATHER_DATA
BEGIN
    PR_INSERT_TRANSFORM_WEATHER_DATA;
END;
/
----- END OF INSERT INTO TRANSFORMATION TABLE CODES -----
-----
```

## Screenshots: Procedure Statements Run Report, Data Inserted into Transformation Table.



The screenshot shows a confirmation dialog titled "Run Script". It asks for confirmation to run a script named "ETL\_TransformTables\_Insert". The dialog provides details about the script: it was created on 12/16/2020 at 10:09:46 AM by AV.SAPKOTA999@GMAIL.COM, has 6 statements, and a size of 3,904 bytes. A "Run Now" button is prominently displayed.

The screenshot shows the results of running the "ETL\_TransformTables\_Insert" script. The status is "Complete". The results table lists 6 statements, all of which were successful. The statements include creating or replacing procedures for transforming delayed flights, cancelled flights, and weather data. The results summary indicates 6 statements processed, 0 errors, and 0 warnings.

**TRANSFORM\_DELAYED\_FLIGHTS**

Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries	
Query	Count Rows	Insert Row	Load Data										
EDIT	DELAYED_FLIGHT_ID	FLIGHT_NUMBER	TAIL_NUMBER	AIR_SYSTEM_DELAY	SECURITY_DELAY	AIRLINE_DELAY	LATE_AIRCRAFT_DELAY	WEATHER_DELAY	FLIGHTS_AIRLINE	DATA_SOURCE			
482	1922	N769US	B1	0	0	0	0	0	UNKNOWN	flights_2017			
483	235	N8KLAA	0	0	0	0	0	0	UNKNOWN	flights_2017			
765	1820	N583UW	0	0	0	0	0	0	UNKNOWN	flights_2017			
766	235	N8J7AA	0	0	0	0	0	0	UNKNOWN	flights_2017			
767	2057	N923US	0	0	0	0	0	0	UNKNOWN	flights_2017			
768	879	N545UW	0	0	144	0	0	0	UNKNOWN	flights_2017			
769	1445	N833AA	0	0	0	0	0	0	UNKNOWN	flights_2017			
770	67	N8MMAA	20	0	0	0	0	0	UNKNOWN	flights_2017			
771	1857	N572UW	8	0	0	0	10	0	UNKNOWN	flights_2017			
772	67	N83BAA	99	0	0	0	0	0	UNKNOWN	flights_2017			
773	1857	N881UW	0	0	0	0	0	0	UNKNOWN	flights_2017			
774	1831	N559UW	0	0	0	0	0	0	UNKNOWN	flights_2017			
775	1715	N755US	22	0	0	0	0	0	UNKNOWN	flights_2017			
776	67	N83KAA	109	0	0	0	0	0	UNKNOWN	flights_2017			
777	2019	N830AW	0	0	0	0	0	0	UNKNOWN	flights_2017			

Download

row(s) 1 ~ 15 of more than 500 [Next ▶](#)

av.sapkota99@gmail.com [Edit Level 6](#) [en](#)

Copyright © 1999-2020, Oracle and/or its affiliates. All rights reserved.

Application Express 20.2.0.0.20

---

**TRANSFORM\_CANCELLED\_FLIGHTS**

Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries	
Query	Count Rows	Insert Row	Load Data										
EDIT	CANCELLED_FLIGHT_ID	FLIGHT_NUMBER	TAIL_NUMBER	FLIGHTS_AIRLINE	CANCELLATION_REASON							DATA_SOURCE	
1	291	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017								
2	2314	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017								
3	84	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017								
4	1655	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017								
5	193	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017								
6	1096	N8DAAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017								
7	2314	N8DAAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017								
8	84	N8HSAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017								
9	1655	N8DMAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017								
10	195	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017								
11	1096	N8LHAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017								
12	291	N8MMAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017								
13	2314	N8GRAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017								
14	2314	N8AAAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017								
15	84	N8ELAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017								

Download

row(s) 1 ~ 15 of 109 [Next ▶](#)

av.sapkota99@gmail.com [Edit Level 6](#) [en](#)

Copyright © 1999-2020, Oracle and/or its affiliates. All rights reserved.

Application Express 20.2.0.0.20

TRANSFORM\_WEATHER\_DATA

Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries
	Query	Count Rows	Insert Row	Load Data								
	EDIT											
		WEATHER_DATA_ID	WEATHER_STATION	WEATHER_DATE	AWND	PRCP	TAVG					
1	625	USW0004789	09/16/2018	13.7	0	70						
2	626	USW0004789	09/17/2018	11	0	74						
3	627	USW0004789	09/18/2018	9.62	0	75						
4	628	USW0004789	09/19/2018	10.3	0	78						
5	629	USW0004789	09/20/2018	10.7	0	80						
6	630	USW0004789	09/21/2018	8.05	.31	75						
7	631	USW0004789	09/22/2018	8.28	1.09	72						
8	632	USW0004789	09/23/2018	6.71	.08	69						
9	633	USW0004789	09/24/2018	9.62	0	74						
10	634	USW0004789	09/25/2018	8.05	0	67						
11	635	USW0004789	09/26/2018	6.71	0	74						
12	636	USW0004789	09/27/2018	10.5	.32	72						
13	637	USW0004789	09/28/2018	13	.51	77						
14	638	USW0004789	09/29/2018	11.4	0	82						
15	639	USW0004789	09/30/2018	11	.02	79						

## Loading

Loading data into the target data warehouse is the last step of ETL process. Huge volume of data is inserted into fact and dim tables from transformation tables. Firstly, to start the loading of data from transformation table had to be inserted into dim tables, which has been performed using the following code:

N.B. You can also find these codes in a separate file: ‘Task 3 - Loading - Insert data into Dim table PLSQL’ in the attached folder: ‘SQL Scripts’

---

```
-- PL/SQL INSERT DATA INTO DIM TABLES
```

---

```
-- PROCEDURE TO INSERT DATA FROM TRANSFORMATION TABLES INTO DIM TABLES
CREATE OR REPLACE PROCEDURE PR_INSERT_INTO_DIM_TABLES
AS
    CURSOR Curr_Delayed_Flights IS SELECT * FROM TRANSFORM_DELAYED_FLIGHTS;
    CURSOR Curr_Cancelled_Flights IS SELECT * FROM TRANSFORM_CANCELLED_FLIGHTS;
    CURSOR Curr_Weather_Data IS SELECT * FROM TRANSFORM_WEATHER_DATA;

BEGIN
    -- INSERTING DATA INTO DIM_DELAYED_FLIGHTS
    FOR CDF IN Curr_Delayed_Flights LOOP
        INSERT INTO DIM_DELAYED_FLIGHTS (DELAYED_FLIGHT_ID, FLIGHT_NUMBER, TAIL_NUMBER, AIR_SYSTEM_DELAY,
SECURITY_DELAY, AIRLINE_DELAY, LATE_AIRCRAFT_DELAY, WEATHER_DELAY, FLIGHTS_AIRLINE, DATA_SOURCE)
        VALUES (CDF.DELAYED_FLIGHT_ID, CDF.FLIGHT_NUMBER, CDF.TAIL_NUMBER, CDF.AIR_SYSTEM_DELAY, CDF.SECURITY_DELAY, CDF.AIRLINE_DELAY, CDF.LATE_AIRCRAFT_DELAY, CDF.WEATHER_DELAY, CDF.FLIGHTS_AIRLINE, CDF.DATA_SOURCE);
    END LOOP;
END;
```

```

    END LOOP;

    -- INSERTING DATA INTO DIM_CANCELLED_FLIGHTS
    FOR CCF IN CURR_CANCELLED_FLIGHTS LOOP
        INSERT INTO DIM_CANCELLED_FLIGHTS (CANCELLED_FLIGHT_ID, FLIGHT_NUMBER, TAIL_NUMBER, FLIGHTS_AIRLINE, CANCELLATION_REASON, DATA_SOURCE)
        VALUES (CCF.CANCELLED_FLIGHT_ID, CCF.FLIGHT_NUMBER, CCF.TAIL_NUMBER, CCF.FLIGHTS_AIRLINE, CCF.CANCELLATION_REASON, CCF.DATA_SOURCE);
    END LOOP;

    -- INSERTING DATA INTO DIM_WEATHER_DATA
    FOR CWD IN CURR_WEATHER_DATA LOOP
        INSERT INTO DIM_WEATHER_DATA (WEATHER_DATA_ID, WEATHER_STATION, WEATHER_DATE, AWND, PRCP, TAVG)
        VALUES (CWD.WEATHER_DATA_ID, CWD.WEATHER_STATION, CWD.WEATHER_DATE, CWD.AWND, CWD.PRCP, CWD.TAVG);
    END LOOP;
END;
/

-- EXECUTE PROCEDURE PR_INSERT_INTO_DIM_TABLES
BEGIN
    PR_INSERT_INTO_DIM_TABLES;
END;
/
-----  

-----  

-- DROP SEQUENCE DIM_TIME_SEQ BEFORE CREATION
DROP SEQUENCE DIM_TIME_SEQ;

-- CREATE SEQUENCE DIM_TIME_SEQ
CREATE SEQUENCE DIM_TIME_SEQ INCREMENT BY 1 START WITH 1;

-- CREATE TRIGGER WHICH INCREASES THE VALUE OF PRIMARY KEY EVERY TIME A NEW DATA IS INSERTED.
CREATE OR REPLACE TRIGGER DIM_TIME_TRIGGER
BEFORE INSERT ON DIM_TIME

FOR EACH ROW
BEGIN
    :new.time_id := DIM_TIME_SEQ.NEXTVAL;
END;
/

```

```

-
- PL/SQL TO INSERT ONLY THE YEAR INTO DIM TIME SO THAT FLIGHTS DETAILS WHOSE MONTH IS NOT SPECIFIED CAN BE
LINKED.

CREATE OR REPLACE PROCEDURE PR_YEAR_INSERT_INTO_DIM_TIME
AS
CURSOR CUR_TIME IS (
    SELECT DISTINCT YEAR FROM FLIGHTS_2017
    UNION
    SELECT DISTINCT YEAR FROM FLIGHTS_2018
    UNION
    SELECT DISTINCT FLIGHT_THE_YEAR FROM FLYU_FLIGHTS
    UNION
    SELECT DISTINCT EXTRACT(YEAR FROM WEATHER_DATE) FROM TRANSFORM_WEATHER_DATA);

BEGIN
    FOR I IN CUR_TIME LOOP
        INSERT INTO DIM_TIME (YEAR) VALUES (I.YEAR);
    END LOOP;
END;
/

-- EXECUTE PROCEDURE PR_YEAR_INSERT_INTO_DIM_TIME
BEGIN
    PR_YEAR_INSERT_INTO_DIM_TIME;
END;
/


-- CREATE PROCEDURE TO INSERT DATA INTO DIM TIME TABLE
CREATE OR REPLACE PROCEDURE PR_INSERT_INTO_DIM_TIME
AS
V_YEAR NUMBER;
V_MONTH NUMBER;
V_QUARTER NUMBER;

CURSOR CUR_YEAR IS (
    SELECT DISTINCT YEAR FROM FLIGHTS_2017
    UNION
    SELECT DISTINCT YEAR FROM FLIGHTS_2018
    UNION
    SELECT DISTINCT FLIGHT_THE_YEAR FROM FLYU_FLIGHTS
    UNION
    SELECT DISTINCT EXTRACT(YEAR FROM WEATHER_DATE) FROM TRANSFORM_WEATHER_DATA
);

```

```

CURSOR CUR_MONTH IS SELECT DISTINCT EXTRACT(MONTH FROM WEATHER_DATE) MONTH FROM TRANSFORM_WEATHER_DATA
ORDER BY MONTH ASC;

BEGIN
    FOR CY IN CUR_YEAR LOOP
        -- CHECKS THE YEAR FROM DATA SOURCE
        IF (CY.YEAR = 2017) THEN
            V_YEAR := 2017;
        ELSE
            V_YEAR := 2018;
        END IF;

        -- CHECKS THE MONTH FROM DATA SOURCE AND IDENTIFIES WHICH QUARTER OF THE YEAR
        FOR CM IN CUR_MONTH LOOP
            V_MONTH := CM.MONTH;

            IF V_MONTH=3 OR V_MONTH=4 OR V_MONTH=5 THEN
                V_QUARTER := 1;
            ELSIF V_MONTH=6 OR V_MONTH=7 OR V_MONTH=8 THEN
                V_QUARTER := 2;
            ELSIF V_MONTH=9 OR V_MONTH=10 OR V_MONTH=11 THEN
                V_QUARTER := 3;
            ELSE
                V_QUARTER := 4;
            END IF;

            INSERT INTO DIM_TIME (MONTH, QUARTER, YEAR) VALUES (V_MONTH, V_QUARTER, V_YEAR);

            V_MONTH := '';
            V_QUARTER := '';
        END LOOP;

        V_YEAR := '';
    END LOOP;
END;
/

-- EXECUTE PROCEDURE PR_INSERT_INTO_DIM_TIME
BEGIN
    PR_INSERT_INTO_DIM_TIME;
END;
/
----- END OF LOADING DATA INTO DIM TABLES CODE -----
-----
```

## Screenshots: Script Run, Report and Data loaded into Dim tables:

The screenshot shows a confirmation dialog titled "Run Script". It displays the following information:

Script Name	ETL>LoadingIntoDim
Created	on 12/17/2020 03:27:43 AM by AV.SAPKOTA999@GMAIL.COM
Updated	on 12/18/2020 03:30:48 AM by AV.SAPKOTA999@GMAIL.COM
Number of Statements	9
Script Size in Bytes	5,005

Buttons at the bottom: "Cancel" and "Run Now".

The screenshot shows the execution summary for the script "ETL>LoadingIntoDim". Status: Complete. Rows: 15.

Number	Elapsed	Statement	Feedback	Rows
1	0.13	CREATE OR REPLACE PROCEDURE PR_INSERT_INTO_DIM_TABLES AS	Procedure created.	0
2	0.90	BEGIN PR_INSERT_INTO_DIM_TABLES; END;	Statement processed.	1
3	0.03	DROP SEQUENCE DIM_TIME_SEQ	Sequence dropped.	0
4	0.03	CREATE SEQUENCE DIM_TIME_SEQ INCREMENT BY 1 START WITH 1	Sequence created.	0
5	0.04	CREATE OR REPLACE TRIGGER DIM_TIME_TRIGGER BEFORE INSERT ON	Trigger created.	0
6	0.10	CREATE OR REPLACE PROCEDURE PR_YEAR_INSERT_INTO_DIM_TIME AS	Procedure created.	0
7	0.10	BEGIN PR_YEAR_INSERT_INTO_DIM_TIME; END;	Statement processed.	1
8	0.05	CREATE OR REPLACE PROCEDURE PR_INSERT_INTO_DIM_TIME AS V	Procedure created.	0
9	0.03	BEGIN PR_INSERT_INTO_DIM_TIME; END;	Statement processed.	1

Download

row(s) 1 - 9 of 9

9 Statements Processed      9 Successful      0 With Errors

The screenshot shows the final status of the script "ETL>LoadingIntoDim". Status: Complete. Rows: 15.

**DIM\_DELAYED\_FLIGHTS**

Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries	+ ▾
Query	Count Rows	Insert Row	Load Data										
EDIT	DELAYED_FLIGHT_ID	FLIGHT_NUMBER	TAIL_NUMBER	AIR_SYSTEM_DELAY	SECURITY_DELAY	AIRLINE_DELAY	LATE_AIRCRAFT_DELAY	WEATHER_DELAY	FLIGHTS_AIRLINE	DATA_SOURCE			
341	291	N835AA	17	0	24	0	0	0	UNKNOWN	flights_2017			
342	2314	N80HAA	0	0	5	39	0	0	UNKNOWN	flights_2017			
343	1653	N8B0AA	0	0	0	0	0	0	UNKNOWN	flights_2017			
344	1096	N8AIIIA	0	0	0	0	0	0	UNKNOWN	flights_2017			
345	1653	N8AHAA	0	0	0	0	0	0	UNKNOWN	flights_2017			
346	2314	N8JRAA	0	0	0	0	0	0	UNKNOWN	flights_2017			
347	1653	N8JRAA	0	0	0	0	0	0	UNKNOWN	flights_2017			
348	2314	N8AAAA	0	0	8	67	0	0	UNKNOWN	flights_2017			
349	1653	N8OKAA	34	0	73	0	0	0	UNKNOWN	flights_2017			
350	291	N83HAA	17	0	0	0	0	0	UNKNOWN	flights_2017			
351	2314	N8HLAA	14	0	4	66	0	0	UNKNOWN	flights_2017			
352	84	N8LSAA	0	0	23	0	0	0	UNKNOWN	flights_2017			
353	291	N8CGAA	0	0	0	0	0	0	UNKNOWN	flights_2017			
354	2314	N8BJAA	0	0	0	0	0	0	UNKNOWN	flights_2017			
355	1653	N8GLAA	0	0	0	0	0	0	UNKNOWN	flights_2017			

**DIM\_CANCELLED\_FLIGHTS**

Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL	REST	Sample Queries	+ ▾
Query	Count Rows	Insert Row	Load Data										
EDIT	CANCELLED_FLIGHT_ID	FLIGHT_NUMBER	TAIL_NUMBER	FLIGHTS_AIRLINE	CANCELLATION_REASON	DATA_SOURCE							
16		1653	N8ENAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017							
17		193	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017							
18		1096	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017							
19		291	N8BMAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017							
20		2314	N8FHA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017							
21		2314	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017							
22		84	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017							
23		1653	N8DAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017							
24		193	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017							
25		1096	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017							
26		2314	N8FFAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017							
27		1653	N8DUAA	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017							
28		1653	N8EBA	UNKNOWN	ADVERSE WEATHER CONDITION	flights_2017							
29		1096	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017							
30		291	UNKNOWN	UNKNOWN	AIR TRAFFIC RESTRICTIONS	flights_2017							

**DIM\_WEATHER\_DATA**

EDIT	WEATHER_DATA_ID	WEATHER_STATION	WEATHER_DATE	AWND	PRCP	TAVG
56	USW0004789	02/25/2017	107	0	28	
57	USW0004789	02/26/2017	128	0	28	
58	USW0004789	02/27/2017	103	0	25	
59	USW0004789	02/28/2017	828	0	23	
72	USW0004789	05/15/2017	805	0	37	
73	USW0004789	05/14/2017	895	111	43	
74	USW0004789	05/15/2017	201	0	45	
75	USW0004789	05/16/2017	872	0	41	
76	USW0004789	05/17/2017	183	.02	47	
77	USW0004789	05/18/2017	188	0	35	
78	USW0004789	05/19/2017	119	0	35	
79	USW0004789	05/20/2017	103	.33	35	
80	USW0004789	05/21/2017	917	.02	35	
81	USW0004789	05/22/2017	172	0	41	
82	USW0004789	05/23/2017	154	0	32	

Download [row\(s\) 1 - 15 of more than 500](#) Next ►

av.sapkota999@gmail.com ads\_Level\_5 en Copyright © 1999-2020, Oracle and/or its affiliates. All rights reserved. Application Express 20.2.0.0.20

**DIM\_TIME**

EDIT	TIME_ID	MONTH	QUARTER	YEAR
1	-	-	-	2017
2	-	-	-	2018
3	1	4	1	2017
4	2	4	1	2017
5	3	1	1	2017
6	4	1	1	2017
7	5	1	1	2017
8	6	2	1	2017
9	7	2	1	2017
10	8	2	1	2017
11	9	3	1	2017
12	10	3	1	2017
13	11	3	1	2017
14	12	4	1	2017
15	1	4	1	2018

Download [row\(s\) 1 - 15 of 26](#) Next ►

av.sapkota999@gmail.com ads\_Level\_5 en Copyright © 1999-2020, Oracle and/or its affiliates. All rights reserved. Application Express 20.2.0.0.20

Finally, data for fact table were calculated and were loaded into Fact table using the data and measures from dim tables, using the following code:

N.B. You can also find these codes in a separate file: 'Task 3 - Loading - Insert data into Fact Table PLSQL' in the attached folder: 'SQL Scripts'

```

-- DASH LINE
-----



-- DROP SEQUENCE FACT_FLIGHTS_SEQ BEFORE CREATION
DROP SEQUENCE FACT_FLIGHTS_SEQ;

-- CREATE SEQUENCE FACT_FLIGHTS_SEQ
CREATE SEQUENCE FACT_FLIGHTS_SEQ INCREMENT BY 1 START WITH 1;

-- CREATE TRIGGER WHICH INCREASES THE VALUE OF PRIMARY KEY EVERY TIME A NEW DATA IS INSERTED.
CREATE OR REPLACE TRIGGER FACT_FLIGHTS_TRIGGER
BEFORE INSERT ON FACT_FLIGHTS

FOR EACH ROW
BEGIN
    :new.surrogate_flight_id := FACT_FLIGHTS_SEQ.NEXTVAL;
END;
/


-- CREATE PROCEDURE TO INSERT DATA INTO FACT_FLIGHTS
CREATE OR REPLACE PROCEDURE PR_INSERT_INTO_FACT_FLIGHTS
AS
    V_ARRIVALDELAY NUMBER;
    V_YEAR NUMBER;
    V_MONTH NUMBER;
    V_DAY NUMBER;
    V_TIMEID NUMBER;
    V_WEATHERID NUMBER;
    CURSOR CR_DELAYED_FLIGHTS IS SELECT * FROM DIM_DELAYED_FLIGHTS ORDER BY DELAYED_FLIGHT_ID;
    CURSOR CR_CANCELLED_FLIGHTS IS SELECT * FROM DIM_CANCELLED_FLIGHTS ORDER BY CANCELLED_FLIGHT_ID;
    CURSOR CR_DIM_TIME IS SELECT * FROM DIM_TIME ORDER BY TIME_ID;
    CURSOR CR_WEATHER_DATA IS SELECT * FROM DIM_WEATHER_DATA ORDER BY WEATHER_DATA_ID;

    BEGIN

-- CALCULATES MEASURES, IDENTIFIES FOREIGN KEYS AND INSERTS DATA FROM DIM_DELAYED_FLIGHTS
        FOR CDF IN CR_DELAYED_FLIGHTS LOOP
            V_ARRIVALDELAY := CDF.AIR_SYSTEM_DELAY + CDF.SECURITY_DELAY + CDF.AIRLINE_DELAY + CDF.LATE_AIRCRAFT_DELAY + CDF.WEATHER_DELAY;

            IF CDF.DATA_SOURCE = 'flights_2017' THEN
                V_YEAR := 2017;
                SELECT TIME_ID INTO V_TIMEID FROM DIM_TIME WHERE YEAR=V_YEAR AND MONTH IS NULL;
            END IF;
        END LOOP;
    END;

```

```

        INSERT INTO FACT_FLIGHTS (FLIGHT_SERIAL_NO, CANCELLATION_FLAG, ARRIVAL_DELAY, WEATHER_DELAY, F
K1_DELAYED_FLIGHT_ID, FK3_TIME_ID)
        VALUES (CDF.FLIGHT_NUMBER, 0, V_ARRIVALDELAY, CDF.WEATHER_DELAY, CDF.DELAYED_FLIGHT_ID, V_
TIMEID);

    ELSE
        -- IDENTIFIES YEAR, MONTH AND DAY TO FIND FOREIGN KEY FOR TIME
        FOR I IN (SELECT YEAR, MONTH, DAY FROM FLIGHTS_2018
                  WHERE FLIGHT_NUMBER = CDF.FLIGHT_NUMBER AND
                        TAIL_NUMBER = CDF.TAIL_NUMBER AND
                        ARRIVAL_DELAY = V_ARRIVALDELAY) LOOP

            V_YEAR := I.YEAR;
            V_MONTH := I.MONTH;
            V_DAY := I.DAY;
            SELECT TIME_ID INTO V_TIMEID FROM DIM_TIME WHERE YEAR=V_YEAR AND MONTH=V_MONTH;
            -- IDENTIFIES WEATHER_ID FROM WEATHER DATA ON THE BASIS OF DATE AND TIME
            SELECT WEATHER_DATA_ID INTO V_WEATHERID FROM TRANSFORM_WEATHER_DATA WHERE EXTRACT(YEAR FRO
M WEATHER_DATE)=V_YEAR
                AND EXTRACT(MONTH FROM WEATHER_DATE)=V_MONTH AND EXTRACT(DAY FROM WEATHER_DATE
)=V_DAY;

        -- INSERTS ALL IDENTIFIED DATA, MEASURES, AND FOREIGN KEYS TO FACT TABLE FROM DIM DELAYED FLIGHTS
        INSERT INTO FACT_FLIGHTS (FLIGHT_SERIAL_NO, CANCELLATION_FLAG, ARRIVAL_DELAY, WEATHER_DELA
Y, FK1_DELAYED_FLIGHT_ID, FK3_TIME_ID, FK4_WEATHER_DATA_ID)
        VALUES (CDF.FLIGHT_NUMBER, 0, V_ARRIVALDELAY, CDF.WEATHER_DELAY, CDF.DELAYED_FLIGHT_ID, V_
TIMEID, V_WEATHERID);
        END LOOP;
    END IF;

    V_ARRIVALDELAY := '';
    V_YEAR := '';
    V_MONTH := '';
    V_DAY := '';
    V_TIMEID := '';
    V_WEATHERID := '';

END LOOP;

-- CALCULATES MEASURES, IDENTIFIES FOREIGN KEYS AND INSERTS DATA FROM DIM_CANCELLED_FLIGHTS
FOR CCF IN CR_CANCELLED_FLIGHTS LOOP

    IF CCF.DATA_SOURCE = 'flights_2017' THEN
        V_YEAR := 2017;
        SELECT TIME_ID INTO V_TIMEID FROM DIM_TIME WHERE YEAR=V_YEAR AND MONTH IS NULL;

```

```

    INSERT INTO FACT_FLIGHTS (FLIGHT_SERIAL_NO, CANCELLATION_FLAG, ARRIVAL_DELAY, WEATHER_DELAY,
FK2_CANCELLED_FLIGHT_ID, FK3_TIME_ID)
VALUES (CCF.FLIGHT_NUMBER, 1, 0, 0, CCF.CANCELLED_FLIGHT_ID, V_TIMEID);

ELSIF CCF.DATA_SOURCE = 'flights_2018' THEN
-- IDENTIFIES YEAR, MONTH AND DAY TO FIND FOREIGN KEY FOR TIME
FOR I IN (SELECT YEAR, MONTH, DAY FROM FLIGHTS_2018
WHERE FLIGHT_NUMBER = CCF.FLIGHT_NUMBER AND
TAIL_NUMBER = CCF.TAIL_NUMBER AND
CANCELLED=1) LOOP

V_YEAR := I.YEAR;
V_MONTH := I.MONTH;
V_DAY := I.DAY;
SELECT TIME_ID INTO V_TIMEID FROM DIM_TIME WHERE YEAR=V_YEAR AND MONTH=V_MONTH;
-- IDENTIFIES WEATHER_ID FROM WEATHER DATA ON THE BASIS OF DATE AND TIME
SELECT WEATHER_DATA_ID INTO V_WEATHERID FROM TRANSFORM_WEATHER_DATA WHERE EXTRACT(YEAR FRO
M WEATHER_DATE)=V_YEAR
AND EXTRACT(MONTH FROM WEATHER_DATE)=V_MONTH AND EXTRACT(DAY FROM WEATHER_DATE
)=V_DAY;
-- INSERTS ALL IDENTIFIED DATA, MEASURES, AND FOREIGN KEYS TO FACT TABLE FROM DIM CANCELLED FLIGHTS
INSERT INTO FACT_FLIGHTS (FLIGHT_SERIAL_NO, CANCELLATION_FLAG, ARRIVAL_DELAY, WEATHER_DELA
Y, FK2_CANCELLED_FLIGHT_ID, FK3_TIME_ID, FK4_WEATHER_DATA_ID)
VALUES (CCF.FLIGHT_NUMBER, 1, 0, 0, CCF.CANCELLED_FLIGHT_ID, V_TIMEID, V_WEATHERID);
END LOOP;

ELSE
-- CALCULATES MEASURES, IDENTIFIES FOREIGN KEYS AND INSERTS DATA FROM FLYU_FLIGHTS
FOR I IN (SELECT FLIGHT_THE_YEAR, THE_MONTH, THE_DAY FROM FLYU_FLIGHTS
WHERE FLIGHT_NUMBER = CCF.FLIGHT_NUMBER AND
TAIL_NUMBER = CCF.TAIL_NUMBER AND
CANCELLED=1) LOOP

V_YEAR := I.FLIGHT_THE_YEAR;
V_MONTH := I.THE_MONTH;
V_DAY := I.THE_DAY;
SELECT TIME_ID INTO V_TIMEID FROM DIM_TIME WHERE YEAR=V_YEAR AND MONTH=V_MONTH;

-- IDENTIFIES WEATHER_ID FROM WEATHER DATA ON THE BASIS OF DATE AND TIME
SELECT WEATHER_DATA_ID INTO V_WEATHERID FROM TRANSFORM_WEATHER_DATA WHERE EXTRACT(YEAR FRO
M WEATHER_DATE)=V_YEAR
AND EXTRACT(MONTH FROM WEATHER_DATE)=V_MONTH AND EXTRACT(DAY FROM WEATHER_DATE
)=V_DAY;

```

```

-- INSERTS ALL IDENTIFIED DATA, MEASURES, AND FOREIGN KEYS TO FACT TABLE
    INSERT INTO FACT_FLIGHTS (FLIGHT_SERIAL_NO, CANCELLATION_FLAG, ARRIVAL_DELAY, WEATHER_DELAY,
    FK2_CANCELLED_FLIGHT_ID, FK3_TIME_ID, FK4_WEATHER_DATA_ID)
        VALUES (CCF.FLIGHT_NUMBER, 1, 0, 0, CCF.CANCELLED_FLIGHT_ID, V_TIMEID, V_WEATHERID);
    END LOOP;
END IF;

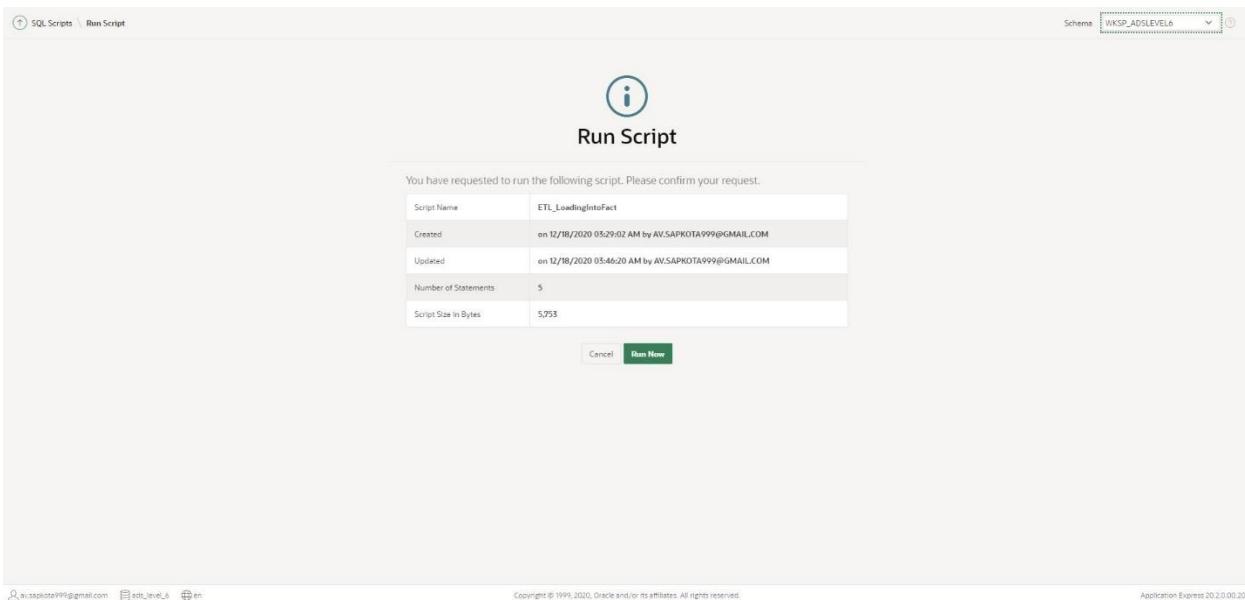
V_ARRIVALDELAY := '';
V_YEAR := '';
V_MONTH := '';
V_DAY := '';
V_TIMEID := '';
V_WEATHERID := '';

END LOOP;
END;
/

-- EXECUTE PROCEDURE PR_INSERT_INTO_FACT_FLIGHTS
BEGIN
    PR_INSERT_INTO_FACT_FLIGHTS;
END;
/
----- END OF INSERT DATA INTO FACT TABLE -----

```

## Screenshots: Script Run, report and Fact table with loaded data



The screenshot shows the 'Results' page for the ETL\_LoadingIntoFact script. The results table is as follows:

Number	Elapsed	Statement	Feedback	Rows
1	0.03	DROP SEQUENCE FACT_FLIGHTS_SEQ	Sequence dropped.	0
2	0.02	CREATE SEQUENCE FACT_FLIGHTS_SEQ INCREMENT BY 1 START WITH 1	Sequence created.	0
3	0.10	CREATE OR REPLACE TRIGGER FACT_FLIGHTS_TRIGGER BEFORE INSERT	Trigger created.	0
4	0.25	CREATE OR REPLACE PROCEDURE PR_INSERT_INTO_FACT_FLIGHTS AS	Procedure created.	0
5	0.61	BEGIN PR_INSERT_INTO_FACT_FLIGHTS; END;	Statement processed.	1

Summary statistics at the bottom:

- Statements Processed: 5
- Successful: 5
- With Errors: 0

FACT\_FLIGHTS

---

Table Data Indexes Model Constraints Grants Statistics UI Defaults Triggers Dependencies SQL REST Sample Queries

Query Count Rows Insert Row Load Data

EDIT	SURROGATE_FLIGHT_ID	FLIGHT_SERIAL_NO	CANCELLATION_FLAG	ARRIVAL_DELAY	WEATHER_DELAY	FK1_DELAYED_FLIGHT_ID	FK2_CANCELLED_FLIGHT_ID	FK3_TIME_ID	FK4_WEATHER_DATA_ID
1	1096	0	43	0	1	-	-	1	-
2	2314	0	65	0	2	-	-	1	-
3	1096	0	0	0	3	-	-	1	-
4	84	0	34	8	4	-	-	1	-
5	1653	0	20	0	5	-	-	1	-
6	195	0	16	0	6	-	-	1	-
7	2314	0	39	0	7	-	-	1	-
8	1096	0	34	0	8	-	-	1	-
9	2314	0	0	0	9	-	-	1	-
10	84	0	28	28	10	-	-	1	-
11	1096	0	188	0	11	-	-	1	-
12	2314	0	157	0	12	-	-	1	-
13	84	0	0	0	13	-	-	1	-
14	1096	0	60	0	14	-	-	1	-
15	291	0	0	0	15	-	-	1	-

Download

row(s) 1 - 15 of more than 500 Next ►

---

w.sapkota999@gmail.com ads\_Level\_6 en

Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved.

Application Express 20.2.0.0.20

## Task 4: Data Analysis/OLAP/Mining Investigation

### Data Analysis

Data analysis is a process of extracting, transforming, and demonstrating raw data and mining them to produce required and meaningful information which are relevant to business objectives. The primary purpose of data analysis is to extract useful information from data which will help making decisions in the future. Data analysis and data warehouse comes hand in hand as data warehouse designs and manages data into one database which can be analyzed to produce meaning results further supporting Business Intelligence. Data analysis is being used by everyone around who are good at decision making. For instance, the process of thinking what we did in the past to make decisions being based on the past actions is a data analysis in general terms. However, data analysis is a broad topic and which has various methods which are based on qualitative and quantitative data analysis. Furthermore, there are several phases that occur simultaneously in data analysis which includes: Data requirement specification, data collection, data processing, data cleaning, data analysis and communication (Tutorialspoint, 2020)

### Business Intelligence

Business Intelligence is a process of analyzing data to make business decision. It is a similar term as data analysis however data analysis is about extracting the data whereas business intelligence is the decision-making phase. Business Intelligence is a suite of software and services which transforms data into visual data organization formats. It supports and performs data analysis and create reports, charts, maps, graphs, and dashboards with detail information which supports decision making and strategic planning to increase revenue, improve operational efficiency and growth of the company. Some benefits of business intelligence include speed and improved decision-making, increased productivity, identify evolving market tendencies, develop business techniques, etc. (Tutorialspoint, 2020).

### Online Analytical Processing (OLAP)

Online Analytical Processing (OLAP) is a database software which allows users to extract, and analyze data and information from multiple databases. It supports 3-dimensional data warehousing. In general, the information and data loaded into OLAP system are extracted from Online Transaction Processing (OLTP) systems. Data warehousing comes under OLAP where data from various sources are organized and managed in such a way that reports and other visual representation of data is easy to produce and generate. There are four basic analytical operations in OLAP which includes Roll-up, Drill-down, slice and dice and pivot. Some advantages of OLAP systems are:

- It is a database platform for business planning, analysis, reporting and financing.
- It makes Information gathering and calculation consistent.
- It provides basic infrastructures for business modelling, data analysis, mining and business intelligence (Guru99, 2020).

### Tableau: Reports, Charts and Dashboards

Tableau Data Source has been imported from the data warehouse created after ETL process and the fact was uploaded into the data source of tableau. In addition, all the dim tables were uploaded into the tableau as a connection of fact flights. The connection of each dim table with fact flight was made possible because of the foreign keys from dim tables into the fact table.



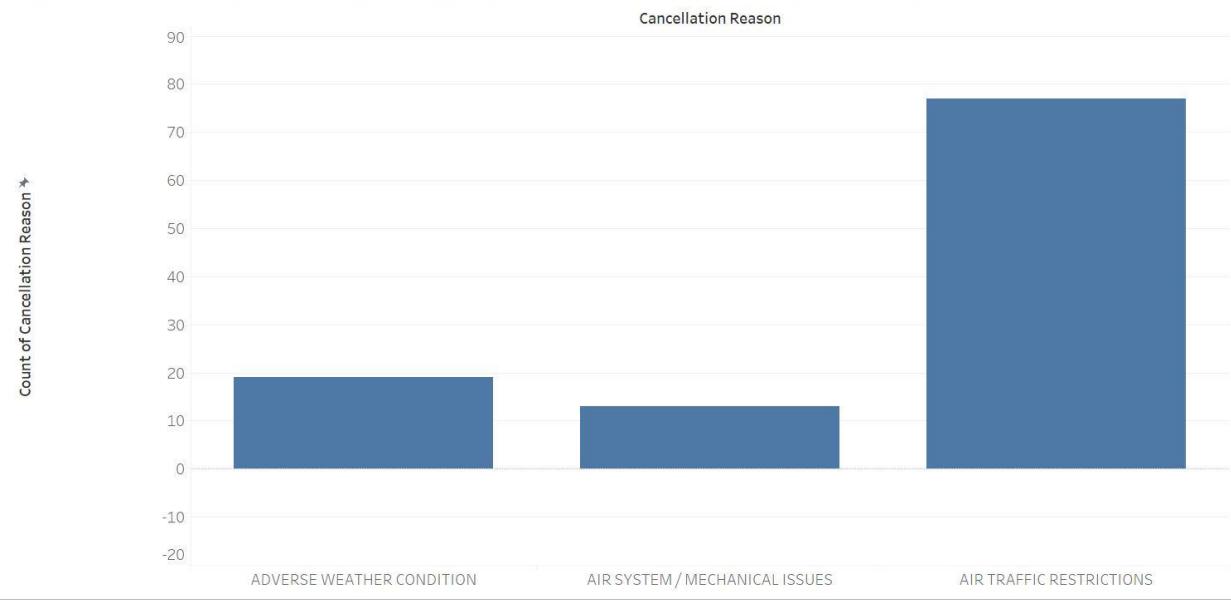
Henceforth, the facts and dimensions loaded into tableau were analysed to produce reports identified in task 1 of the assignment has been produced using pivot function and data visualization tools of the 'Tableau' software.

N.B. All the reports pasted in the document have been separately attached in a folder: 'Report' which contains excel sheet reports and tableau file containing all reports and dashboards designed for FlyU airlines company.

#### **Report 1: Total number of flights cancelled on the basis of cancellation reasons.**

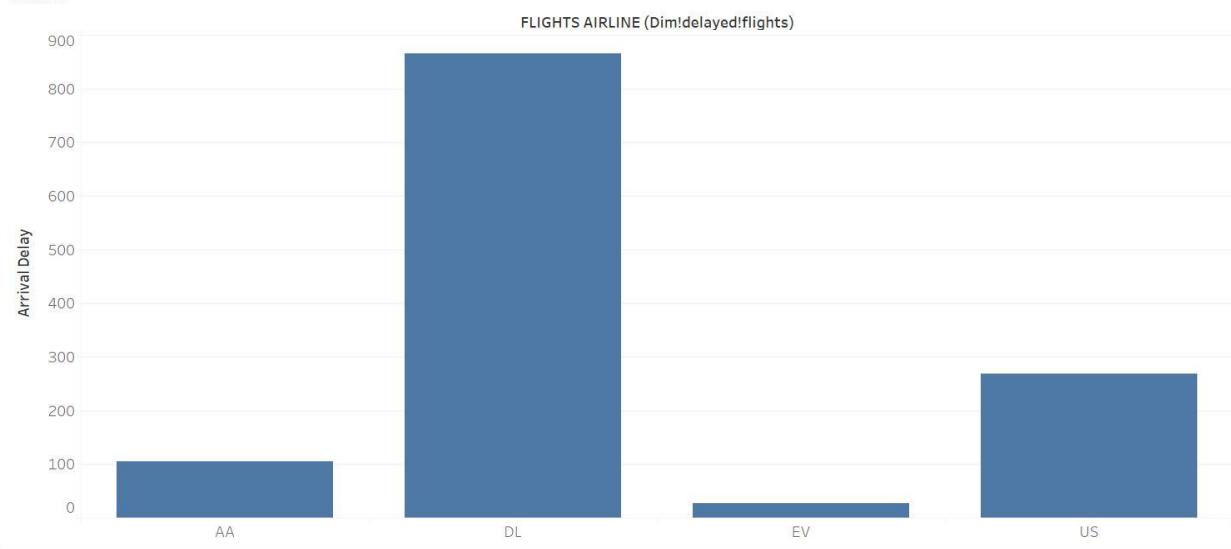
The report below shows distinct cancellation reason due to which flights were cancelled on the horizontal pane and the total number of flights cancelled on the vertical pane. This report can perhaps be useful in order to identify the cancellation reason due to which most of the flights have been cancelled. For instance, on the basis of the report presented above, most of the flights of FlyU airline company were cancelled due to air traffic restrictions, adverse weather condition and air system or mechanical issues in decreasing order. Thus, FlyU airline company can plan their flights, for future, in such time interval when the total number of flights to take off is low. Similarly, they can be aware about the weather condition and predictions before they place a flight for booking. Finally, they may as well repair every aircraft or replace the damaged aircraft before booking a new flight in the future which will eventually reduce the number of cancelled flights hence the quality of service of FlyU company will perhaps improve.

Report 1 - Total number of Flights Cancelled on the basis of cancellation reason



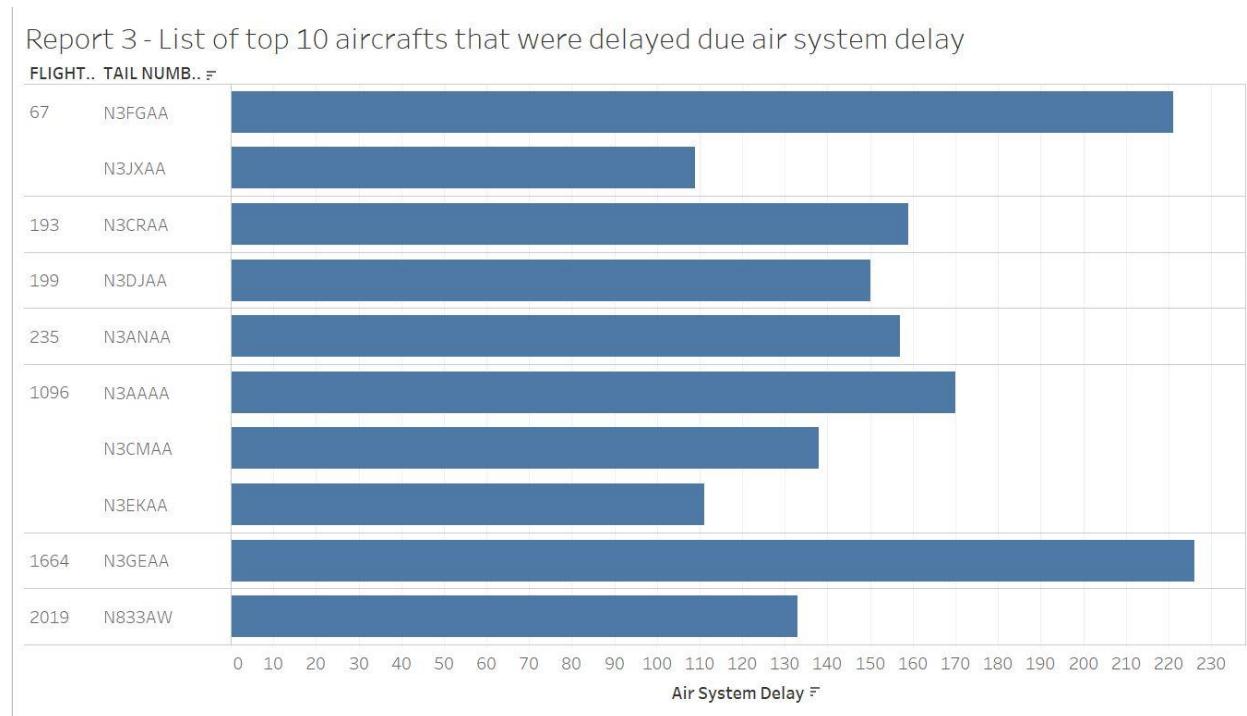
**Report 2: List of airlines whose flights were delayed, in the year 2018, along with total delayed time**

Report 2 - List of airlines whose flights were delayed, in the year 2018, along with the total delayed time.



The report above demonstrates total minutes delayed for arrival on the vertical pane and list of distinct airline companies on the horizontal pane. It is basically total arrival delay of each airline company. The report can be used to identify the airline company's whose flights are being delayed for more minutes. Hence, for instance, FlyU airline company can report and complain the associate airline companies in order to decrease the total arrival delay time which will eventually uplift the quality of service delivered by FlyU.

### **Report 3: List of top 10 aircraft delayed due to air system delay including their delayed time**

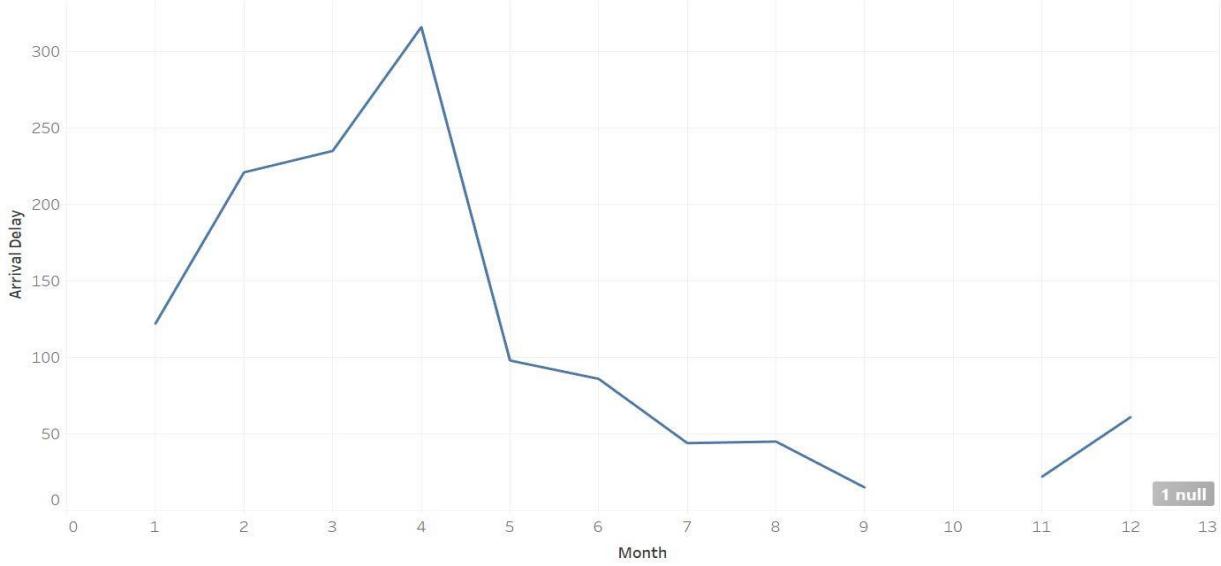


The report above represents the list of aircrafts' tail number along with the flight number on the vertical pane and total minutes each flight and aircraft were delayed due to air system delay on the horizontal pane. The report above can be used to analyse the aircraft's condition. For example, this is the list of top 10 aircraft delayed due to air system delay. Air system delay basically refers to the delay cause due to some mechanical issues on the aircraft. Hence FlyU can identify the top 10 aircraft which are problematic hence they can discard the aircraft in the future until it is completely issue free or can add on a new aircraft rather than using an old problematic aircraft.

### **Report 4: Total arrival delay time for each month in the year 2018**

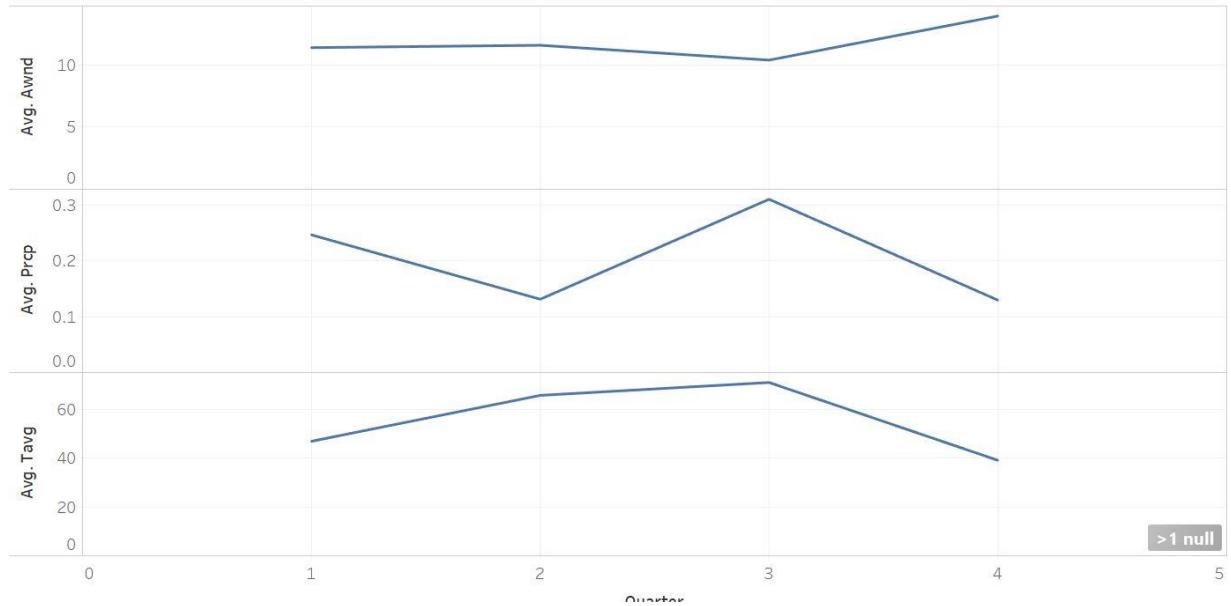
The report below portrays months of the year 2018 on the horizontal pane and total sum of arrival delay on the vertical pane. The data below can be analysed to find out the total minutes delayed in each month which can further be analysed with other data to make decisions on monthly basis. For instance, FlyU airline company can analyse the data below to plan the number of flights to take off every month as delay are high during first 5 months of the year whereas low during the second half of the year. Thus, they can book more flights for the second half of the year which will ensure increase revenue along with quality service.

Report 4 - Total arrival delay time for each month in the year 2018



#### Report 5: List of average PRCP, average AWND, average TAVG of weather data on the basis of quarter of the year

Report 5 - Average weather data (AWND, PRCP and TAVG) on the basis of quarter of the year 2018

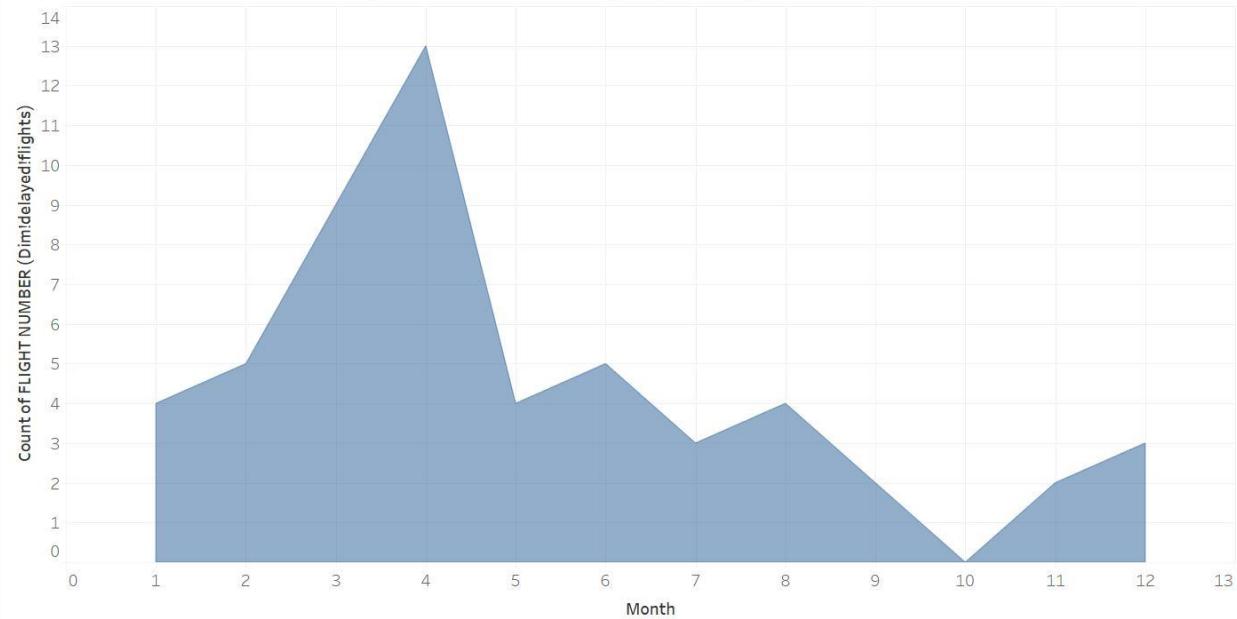


The report above represents average TAVG, average PRCP and average AWND, which are weather reports, on the vertical pane and quarter on the basis of seasons (fall, spring, autumn, and summer). The weather report above can be used to predict the weather. For instance, FlyU can book less flights when average TAVG, AWND and PRCP is high as the adverse weather can affect (delay or cancel) the flights which will decrease the number of delayed or cancelled flights ensuring the improved quality of service.

In addition to the reports identified on the task 1 on the basis of segregated KPI, some more reports were produced using the data managed and maintained in the data warehouse which would further help FlyU airline company to plan sustainable strategies and make profitable decisions. Therefore, the reports presented below were produced using the data warehouse designed and populated in the previous tasks however they can benefit the company on various basis to ensure quality service, customer satisfaction and also help to make decisions which will gradually grow the company.

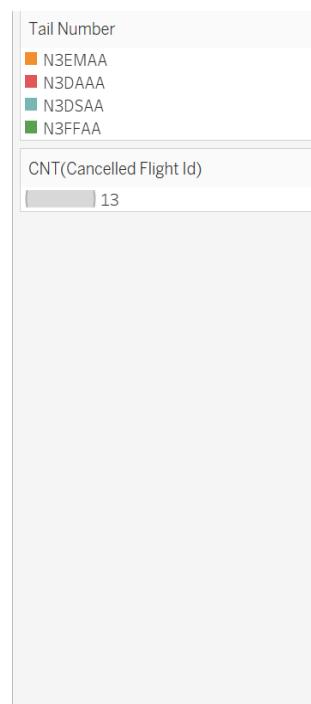
**Report 6: Total number of flights delayed in every month of the year 2018.**

Report 6 - Total number of flights that were delayed, in the year 2018, on the basis of months



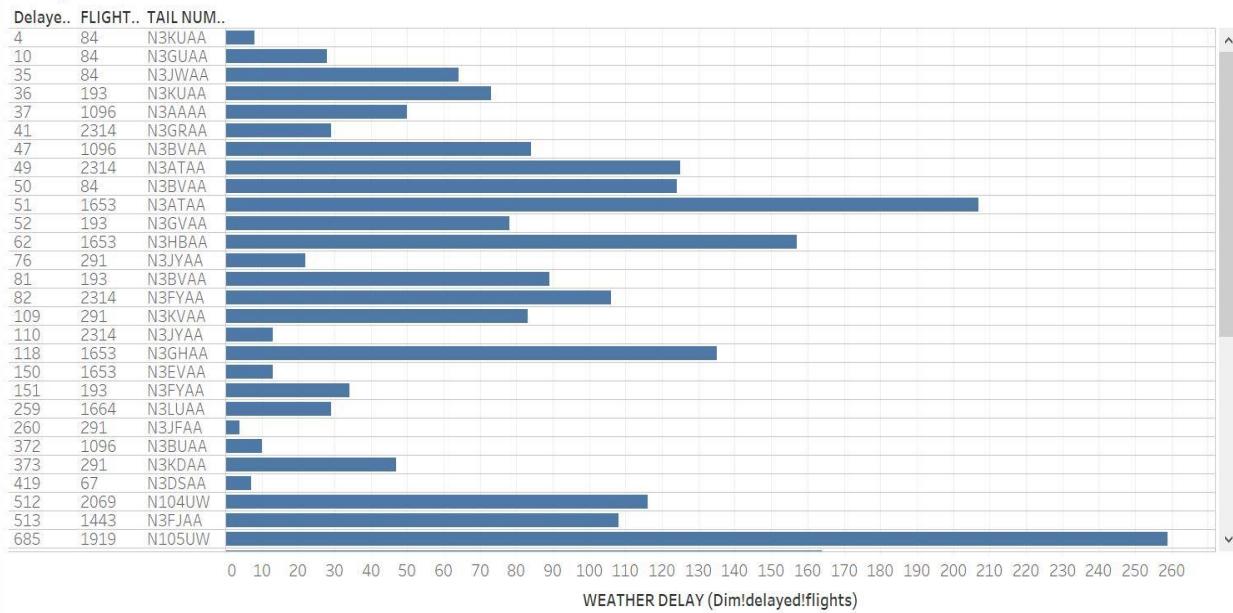
### Report 7: Top 5 aircraft that were cancelled along with the number of times they were cancelled.

Report 7 - List of top 5 aircraft that were cancelled along with the number of times they were cancelled



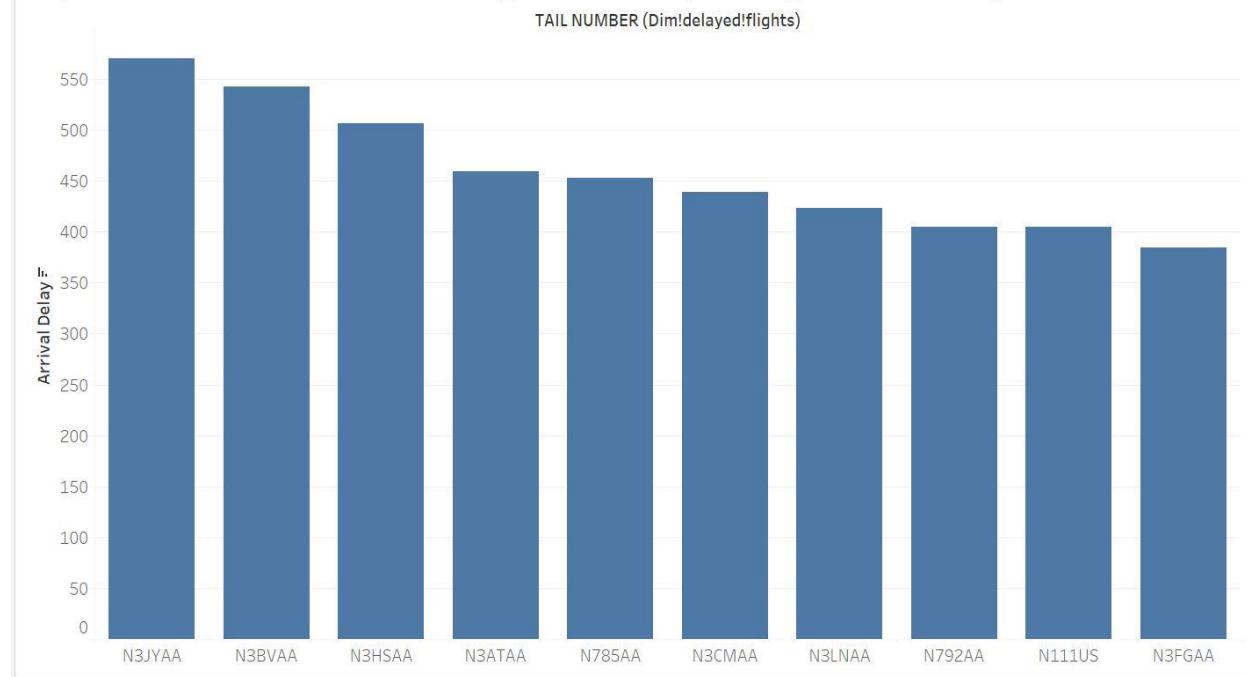
### Report 8: List of aircraft, on the basis of flight number, which was delayed due to weather condition along with the total delayed time in minutes.

Report 8 - List of flight number and tail number that were affected by weather delay along with the delay time



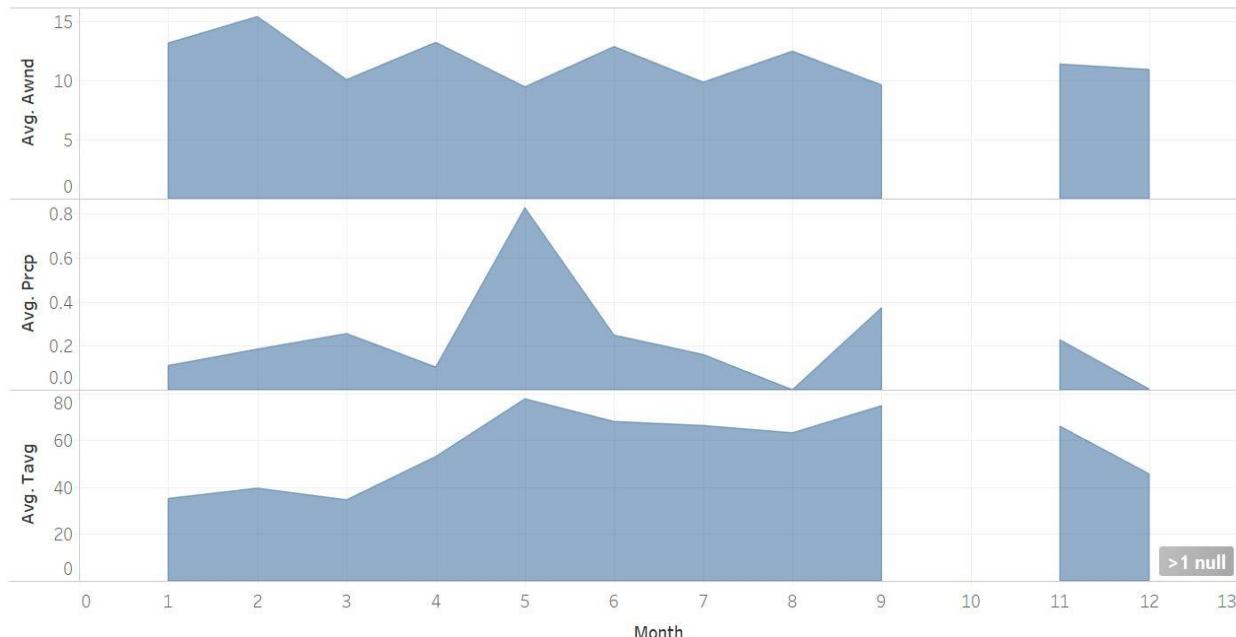
**Report 9: List of 10 aircraft whose flights were delayed on the basis of total minutes delayed for each aircraft.**

Report 9 - List of 10 aircrafts whose flights were delayed along with total delay time



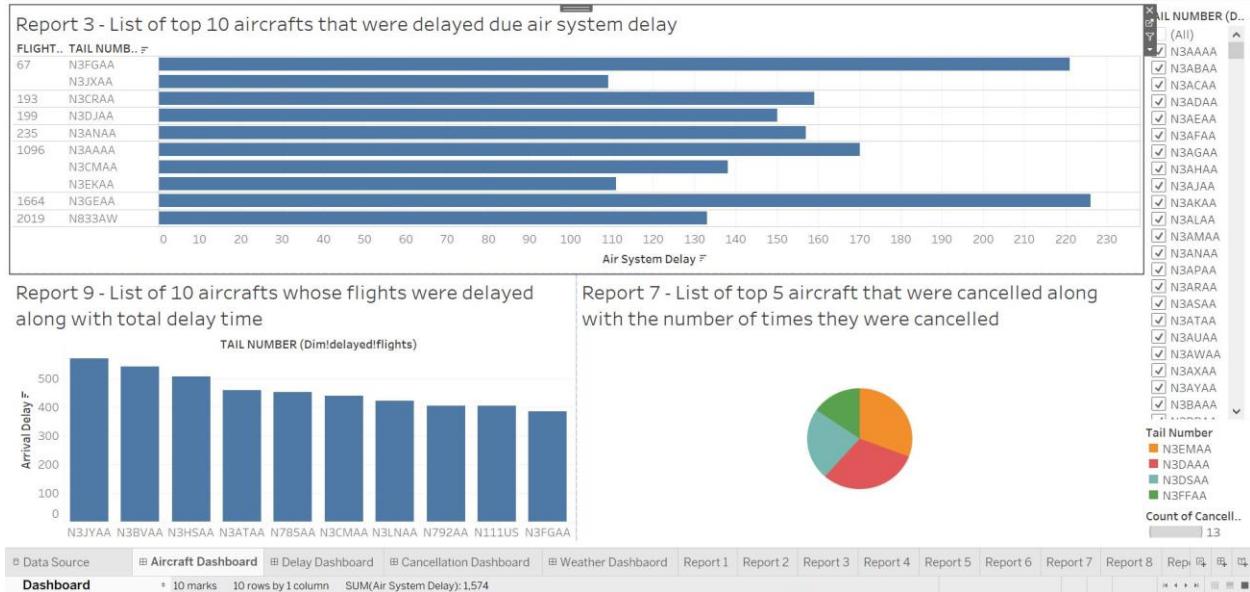
**Report 10: Average weather data on the basis of month of the year 2018**

Report 10 - Average weather data (AWND, PRCP and TAVG) on the basis of months of the year 2018



The reports identified and created above are useful and will change dynamically on every data changes. Hence, dashboards have been created which groups different reports on some condition or data which will make data analysis easier and more convenient for FlyU airline company.

## Aircraft Dashboard



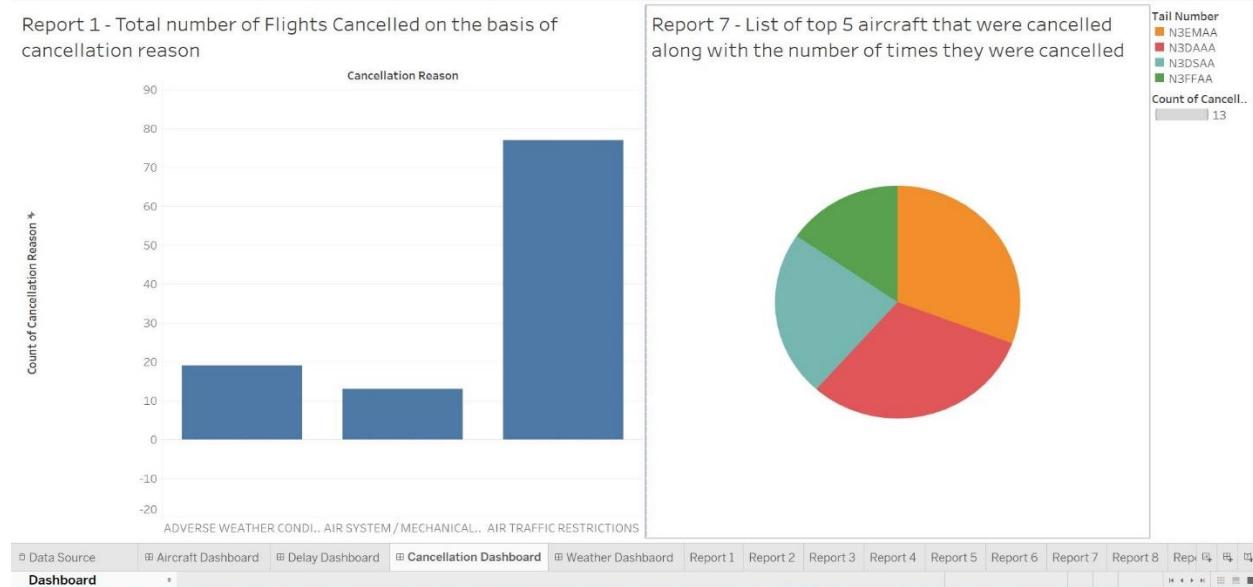
The aircraft dashboard shows the reports which are relevant to aircraft details, thus in order to analyze the data related to aircraft FlyU can directly browse the dashboard rather than viewing reports separately.

## Delay Dashboard



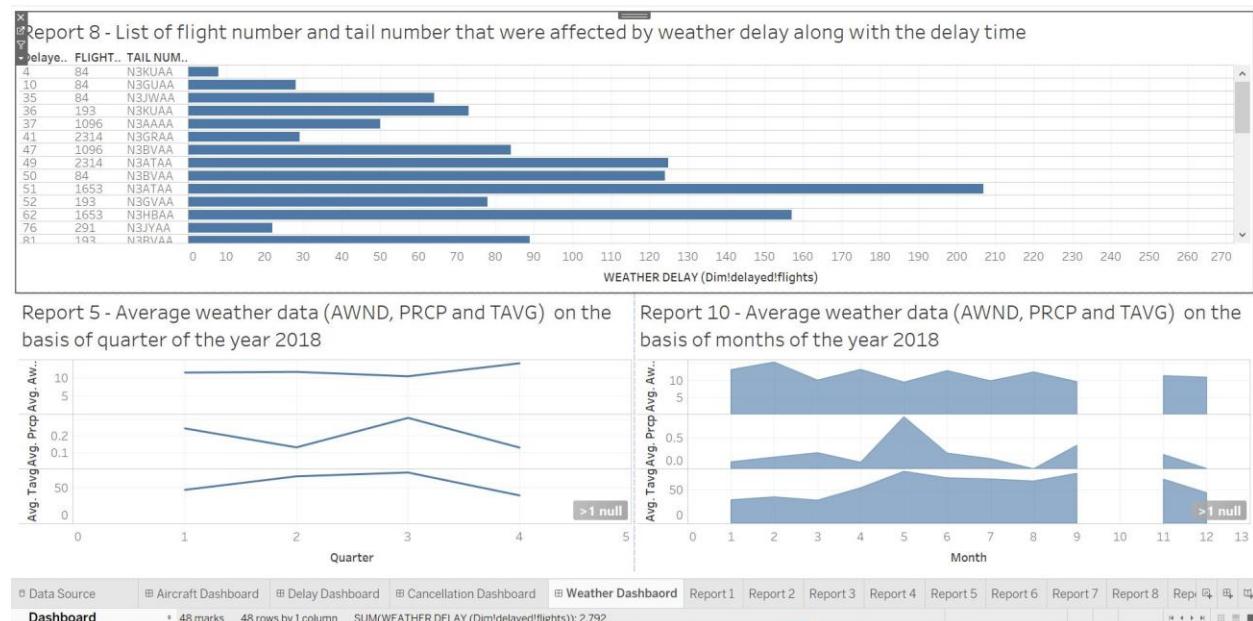
The delay dashboard shows all the reports related to flight delays. Therefore, FlyU can open the dashboard to view multiple reports at once to analyse delayed data.

## Cancellation Dashboard



Similarly, the cancellation dashboard shows all the report and chart which are relevant to flight cancellation.

## Weather Dashboard

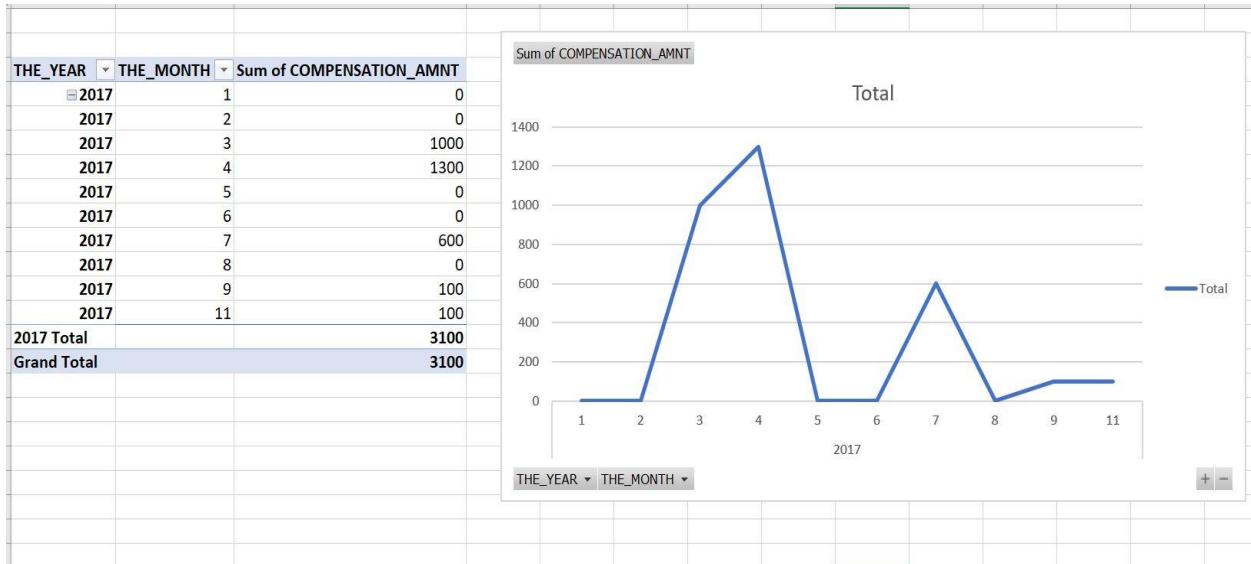


Finally, the weather dashboard shows all the data related to weather and flights which will help the company to analyse flights affected due to weather conditions in the past.

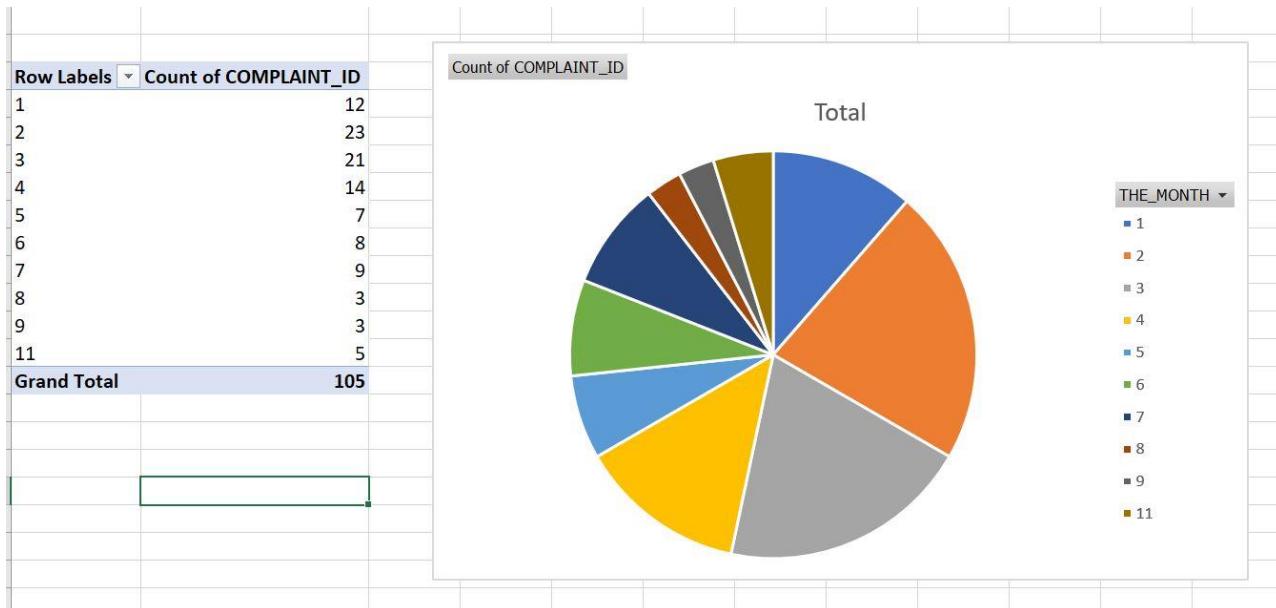
## Additional Excel Sheet Reports

Furthermore, although complaint data was not used in the data warehouse, it can also be used for data analysis. Hence, some more reports were generated using complaint data and other raw data provided by FlyU airline company using excel sheets and pivot table and data visualization functions of MS Excel.

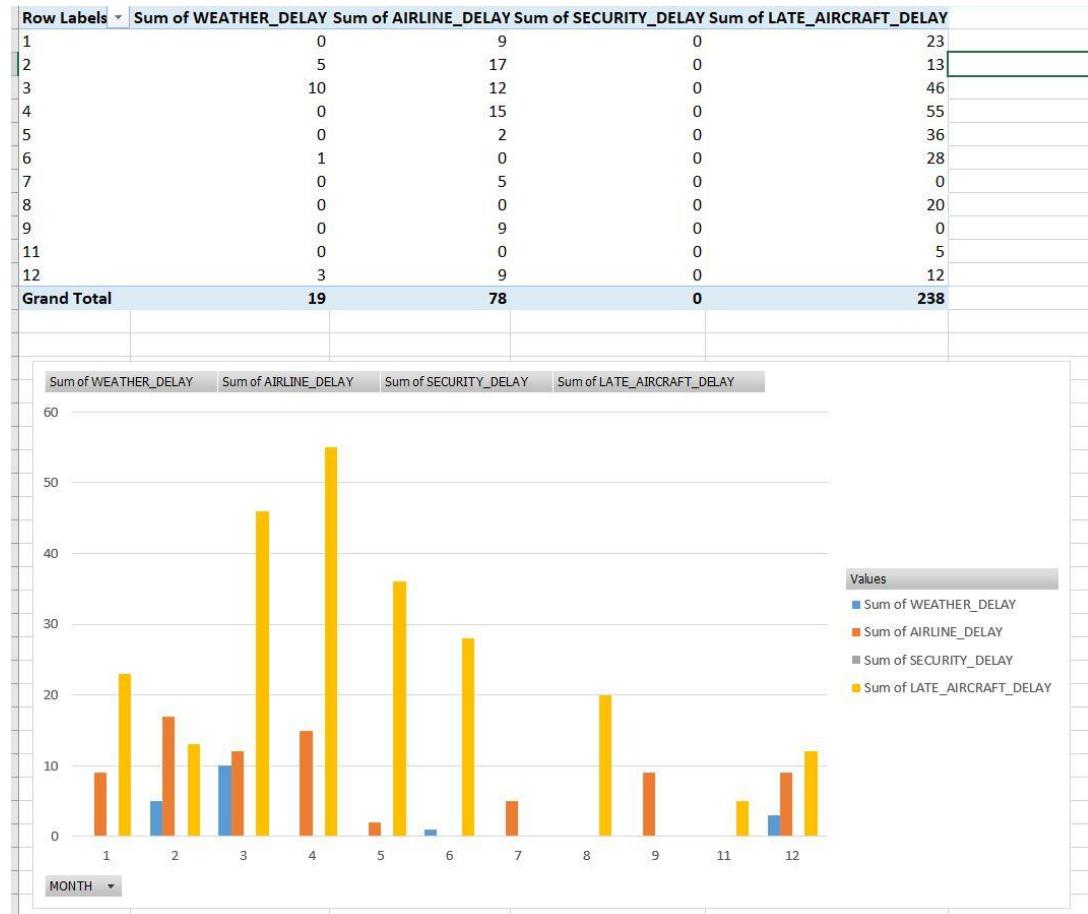
### Excel Report 1: Total sum of compensation amount paid per month.



### Excel Report 2: Total number of complaints reported per month.



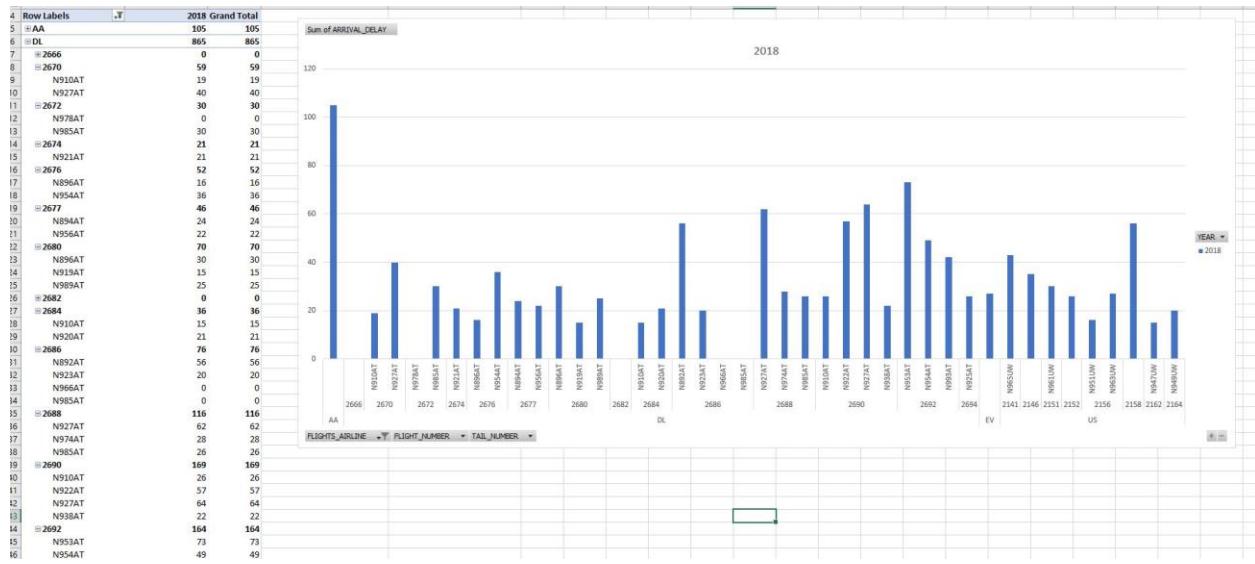
### Excel Report 3: Delay reasons along with total delay time per month.



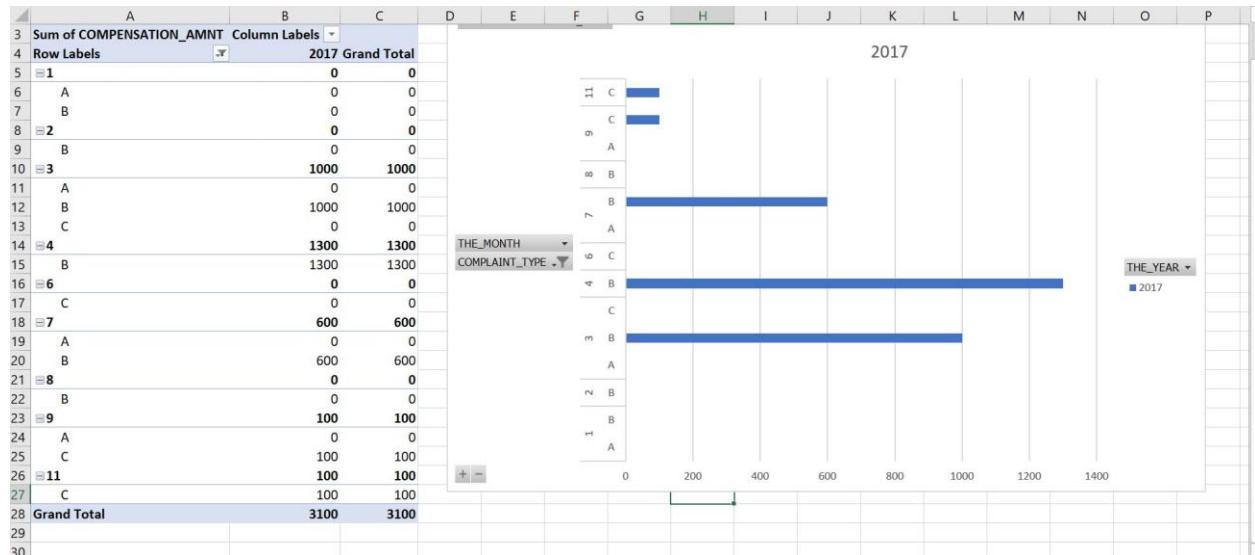
Finally, some more drilled down reports were produced, within the excel itself, in order to analyze multiple data fields at once from the size table which will make the data representation more convenient as multiple columns of data can be view on a same platform.

## Drilled Down Reports

**Sum of arrival delay with respect to airline, flight number and tail number.**



### **Drilled Down Report: Sum of compensation amount with respect to month and complaint type.**



**Drilled Down Report: Sum of air time travelled by each aircraft on the basis of flight number.**

Row Labels	Sum of AIR_TIME
⊕ 2131	790
⊖ 2138	85
N959UW	37
N967UW	48
⊖ 2140	46
N948UW	46
⊖ 2142	34
N946UW	34
⊖ 2150	190
N945UW	41
N949UW	39
N950UW	41
N956UW	35
N961UW	34
⊕ 2151	1139
⊖ 2152	801
N944UW	71
N946UW	37
N947UW	34
N948UW	80
N949UW	38
N951UW	72
N952UW	33
N953UW	36
N954UW	101
N955UW	37
N957UW	118
N958UW	32
N961UW	34
N967UW	78
⊕ 2694	1232
<b>Grand Total</b>	<b>4317</b>

## Task 5

Consider and reflect on data warehousing approaches with respect to FlyU. Draw on the models of Kimball and Inmon in considering solutions for supporting an enterprise (business) and the business processes and functions.

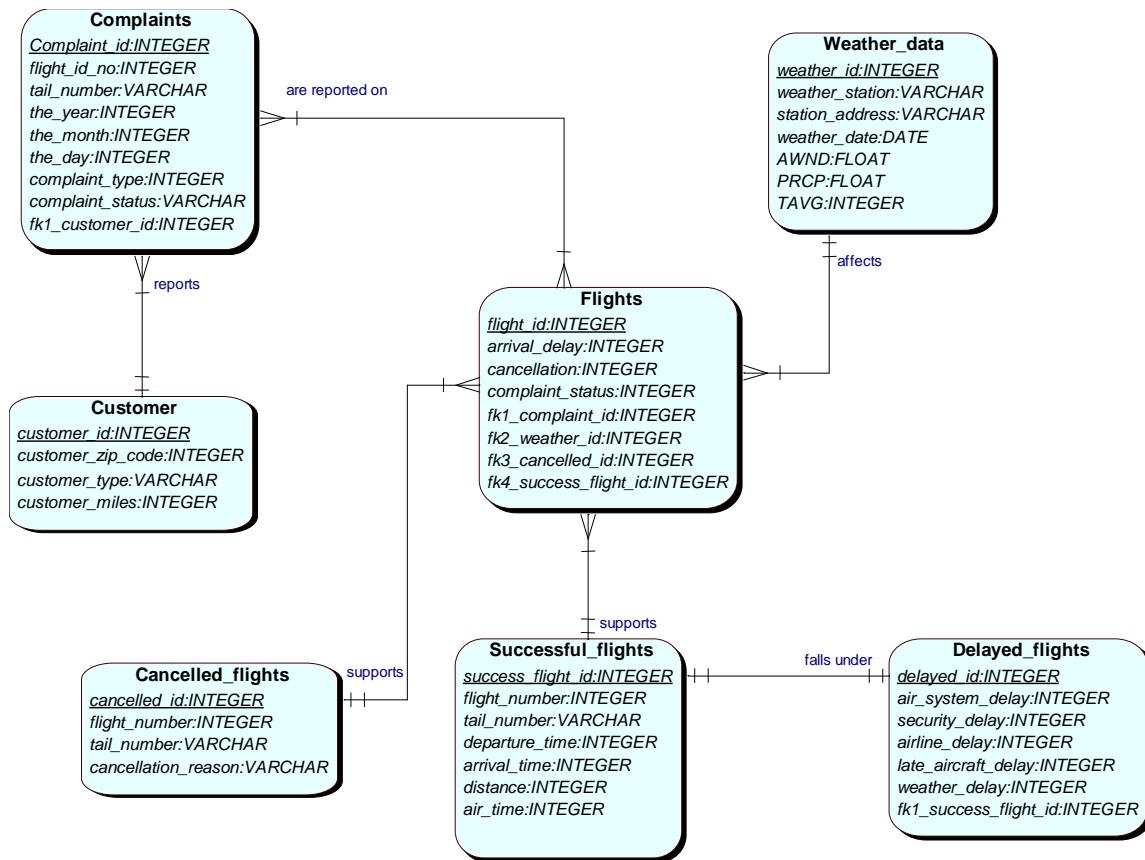
### Introduction

Data warehouse is a relational database management system construct to correlate broad business data from various sources into one comprehensive database which provides insights on businesses. As stated in the first task, the primary purpose of data warehouse is to produce meaningful information which can be used for sustainable business decision making and evaluating future strategies using Business Intelligence, data analytics dashboards and applications. It must be designed in order to support these features. There are two common approaches used to design a data warehouse: top-down approach and bottom-up approach proposed by Bill Inmon and Ralph Kimball respectively. Bill Inmon, the father of data warehousing, defines data warehouse as a collection of integrated, subject-oriented databases designed to support the DSS function, where each unit of data is non-volatile and relevant. In Inmon's approach a normalized data model is designed then the dimensional data marts, which contains data required for business processes, are created from within the data warehouse. In contrast, Ralph Kimball's introduced the bottom-up approach where the data identified first, which are further correlated to develop a data warehouse (Computerweekly, 2020)

The FlyU airline company, case study presented as assignment, is concerned to analyse their data in order to make business decisions which will perhaps enhance the quality of service they deliver, ensure customer satisfaction, and grow the company. FlyU has provided with a number of raw data sets which has the details of flights, customers, weather and complaints. Hence, data warehouse can be used to develop meaningful information which will help FlyU to make decisions to support their business objectives. Thus, FlyU case study can be used in order to reflect on data warehouse approaches proposed by Inmon and Kimball.

### Bill Inmon's Approach

The top-down approach proposed by Inmon to design a data warehouse, firstly, identifies the main subject area and entities of a business. In case of FlyU the main entities are: complaints, flights, weather\_data, cancelled\_flights, and successful\_flights Secondly, a logical model is created for every primary entity. The approach uses a normalized form of entity structure which results in identifying business requirements without any data duplication. Thirdly, the normalized physical model for the data warehouse is constructed. The physical model in case of FlyU airline company is as follows:



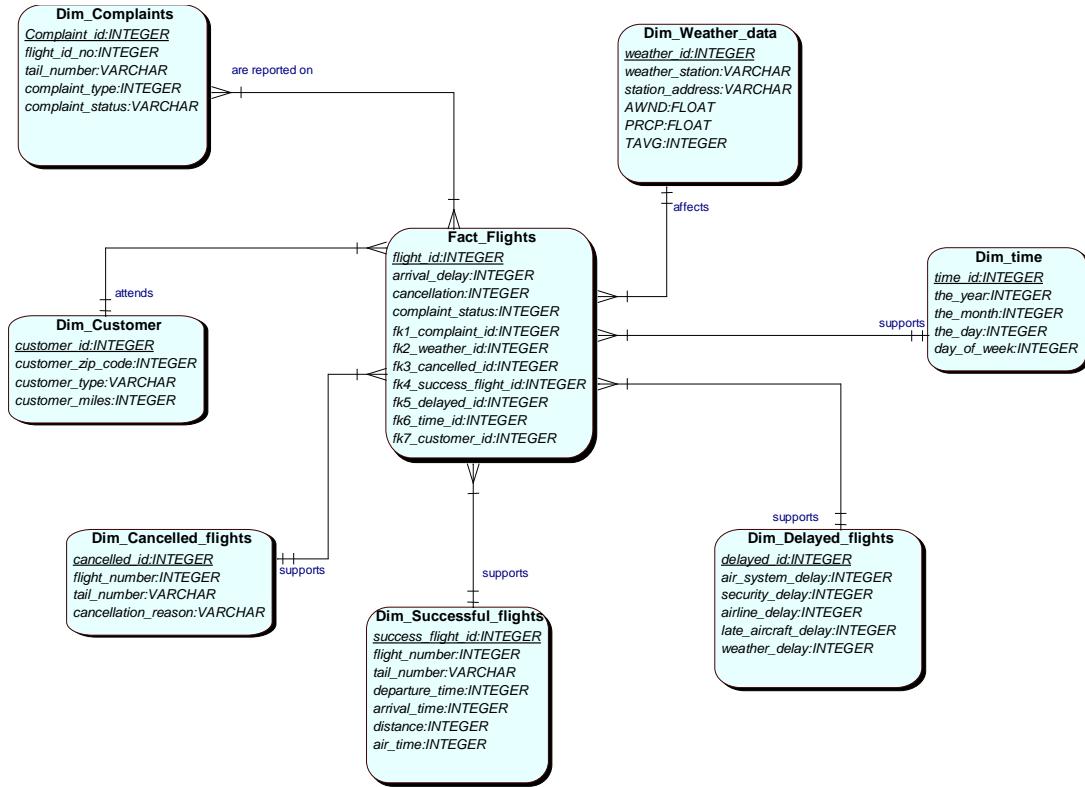
The ERD Model for FlyU airline company presented above supports Inmon's top-down approach. Firstly, the main entities were identified and their logical table was identified in reference to data provided by FlyU. Finally, the ERD was designed. In the data warehouse architecture above, flights is the main entity which correlates all other tables of the database. The data inside complaints, customer, cancelled\_flights, successful\_flights, delayed\_flights and weather\_data will be uploaded in normalized form which will reduce data redundancy. Also, Complaints, cancelled\_flights, successful\_flights and weather\_data have one to many relationship with flights because they are the dimensions which will relate with flights in order to produce reports to support business objectives of FlyU. However, customer has one to many relation with complaints as one customer can report multiple complaints and customer's details is only required if the data mart complaint is used to produce a report. Finally, delayed\_flights has one to one relation with successful\_flights as all delayed\_flights are successful. In addition, delayed\_flight will be used to produce report only if successful\_flights is chosen as data mart to support the business determinations. The data warehouse model above supports business objectives of FlyU flights as it can be used to produce reports and charts which will help the FlyU airline company to make decisions which will impact their business.

In Inmon's approach data in DW is integrated as it works as a unified source. Similarly, the possibility of data duplication is very low which makes the ETL process to be easy and less likely to fail. Also, the approach offers flexibility to update data in case of change of data in the original source. However, complexity to deal with multiple tables increases as time is inserted into each table. The increase in complexity results in more time, thus, it is a little more time consuming. Moreover, additional ETL

operation is required since data marts are created after data warehouse. Finally, in order to manage a data warehouse effectively, expertise in data warehousing is necessary (Astera, 2020)

### Ralph Kimball's Approach

Kimball's bottom-up approach is initiated with data marts. Firstly, data marts are identified on the basis of business objectives. The data marts identified are documented in a star schema. Star schema is the physical design model where one fact table is identified and surrounded by many other associated dimension tables. The star schema for FlyU airline company has been presented below. Secondly, the data sources provided are evaluated and filtered in accordance to the business objectives and data marts. These data sources are analysed via ETL process where data is uploaded to staging area from various sources. Good data and bad data are identified in the staging area and filtered. Then, good data are inserted into clean table and bad data are inserted into bad table where data are cleaned and finally inserted into clean table. Finally, data is segregated into multiple tables in the transformation phase and all the data from transformation phase are loaded to Data warehouse.



The ERD Model for FlyU airline company presented above supports Kimball's bottom-up approach. Firstly, the data marts which supports the business objectives are identified in this approach. However, all tables in the data source have been identified as data marts rather than breaking down for different KPI in order to portray the similar number of entities as compared to Inmon's approach. Fact\_Flights is the fact table with measures arrival\_delay, cancellation, and complaint\_status. Likewise, Dim\_Customer, Dim\_complaints, Dim\_cancelled\_flights, Dim\_delayed\_flights, Dim\_successful\_flights, dim\_weather\_data and Dim\_time are the dimensions to support the fact table. Dim\_time is a dimension which will correlate to every other dim tables via fact table. However, updating data is complex and there are possibilities of redundant data in the star schema presented above. The star schema will be referenced

to design data warehouse and the ETL process will be initiated before loading all the data into data warehouse. The data warehouse model above supports the business process of FlyU airlines as the data warehouse can produce reports which will determine and show their past reports on complaints, customer, delayed and cancelled flights which will perhaps help them make business decisions to uplift their service quality, customer satisfaction and also help them grow.

The Kimball's approach to data warehouse is fast and less time consuming as data are not normalized. Also, data retrieval from data warehouse is fast as data is divided into facts and dimensions. Furthermore, it emphasizes on individual business areas and processes which requires smaller team of designers and developers as query optimization is predictable, straightforward and well-disciplined. In contrast, some data irregularities can occur when data is updated in this approach as data is not normalized. Moreover, it may not support all BI requirements as it focuses on data marts rather than on the entire business processes (Astera, 2020).

### Key Differences between Ralph Kimball's Approach and Bill Inmon's Approach

PARAMETERS	KIMBALL	INMON
Approach	It has Bottom-Up Approach for implementation.	It has Top-Down Approach for implementation.
Data Integration	It focuses Individual business areas.	It focuses Enterprise-wide areas.
Building Time	It is efficient and takes less time.	It is complex and consumes a lot of time.
Cost	It has iterative steps and is cost effective.	Initial cost is huge and development cost is low.
Skills Required	It does not need such skills but a generic team will do job.	It needs specialized skills to make work.
Data Model	It prefers data to be in De-normalized model.	It prefers data to be in normalized model.
Data Store Systems	In this, source systems are highly stable.	In this, source systems have high rate of change.

GeeksforGeeks, 2020

### Conclusion

Data warehouse is the collection of raw business data which can be analyzed to generate meaningful information which will perhaps support decision making and business intelligence. There are two commonly used approaches to design a data warehouse introduced by Bill Inmon and Ralph Kimball. Bill Inmon's top-down approach identifies normalized entities then dimensional data marts are created using ER Model. This approach features integrated data with low data redundancy, has flexibility to change and update data and can handle wide reporting business ambitions although it is a complex, time consuming and expensive approach. On the other hand, Ralph Kimball's bottom-up approach firstly identifies data marts which are documented into star schema and loaded into data warehouse only after the ETL process where data are cleaned and divided into facts and dimensions. Kimball's approach is fast and less time consuming. In addition, data retrieval is easier and the approach is less expensive compared to Inmon's approach. However, updating data into data warehouse is complex and less flexible and the approach may not support all BI requirements. In conclusion, both the approaches are used widely in designing data

warehouse and both have their effectiveness and faultiness. However, to accomplish business objectives of FlyU airlines company Ralph Kimball's approach has been used in task 1 of the assignment.

Word Count: 1475 approx.

## References

*Data Warehousing Definition & Example / InvestingAnswers* (2020). Available at: <https://investinganswers.com/dictionary/d/data-warehousing> (Accessed: 7 December 2020).

*Data Mart vs. Data Warehouse* (2020). Available at: <https://panoply.io/data-warehouse-guide/data-mart-vs-data-warehouse/> (Accessed: 7 December 2020).

*Tony Alessandra Quote* (2020). Available at: <https://www.azquotes.com/quote/773686?ref=quality-service> (Accessed: 7 December 2020).

*What is Data Warehouse? Types, Definition & Example* (2020). Available at: <https://www.guru99.com/data-warehousing.html> (Accessed: 21 December 2020).

*Star and Snowflake Schema in Data Warehouse with Examples* (2020). Available at: <https://www.guru99.com/star-snowflake-data-warehousing.html> (Accessed: 7 December 2020).

*What Is a Data Dictionary? - DATAVERSITY* (2017). Available at: <https://www.dataversity.net/what-is-a-data-dictionary/#> (Accessed: 8 December 2020).

*Data Analysis - Process - Tutorialspoint* (2020). Available at: [https://www.tutorialspoint.com/excel\\_data\\_analysis/data\\_analysis\\_process.htm](https://www.tutorialspoint.com/excel_data_analysis/data_analysis_process.htm) (Accessed: 20 December 2020).

*Data Analysis - Process - Tutorialspoint* (2020). Available at: [https://www.tutorialspoint.com/excel\\_data\\_analysis/data\\_analysis\\_process.htm](https://www.tutorialspoint.com/excel_data_analysis/data_analysis_process.htm) (Accessed: 20 December 2020).

*What is OLAP? Cube, Operations & Types in Data Warehouse* (2020). Available at: <https://www.guru99.com/online-analytical-processing.html#4> (Accessed: 20 December 2020).

*Inmon or Kimball: Which approach is suitable for your data warehouse?* (2020). Available at: <https://www.computerweekly.com/tip/Inmon-or-Kimball-Which-approach-is-suitable-for-your-data-warehouse> (Accessed: 20 December 2020).

*Data Warehouse Concepts: Kimball vs. Inmon Approach* | Astera (2020). Available at:  
<https://www.astera.com/type/blog/data-warehouse-concepts/> (Accessed: 20 December 2020).

*Difference between Kimball and Inmon - GeeksforGeeks* (2020). Available at:  
<https://www.geeksforgeeks.org/difference-between-kimball-and-inmon/> (Accessed: 21 December 2020).