**The British College**

**KATHMANDU**

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

**Faculty of Arts, Environment and Technology LBU Student Id:**

C7202324

**For checking by the student:**

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

Coursework title: Data warehouse Design and Development

Due Date:

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

Module tutor: (In TBC): Dibya Tara Shakya

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

Submission date& time: Date: 22 December,2020 Time: Before noon

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

**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: Date:

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

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

**Teacher's Feedback**

Table of Contents

[Data Warehouse 4](#_Toc59573153)

[What is OLAP and OLTP? 4](#_Toc59573154)

[OLAP 4](#_Toc59573155)

[OLTP 5](#_Toc59573156)

[Example of OLAP and OLTP 5](#_Toc59573157)

[OLAP 5](#_Toc59573158)

[OLTP 5](#_Toc59573159)

[Difference between OLAP and OLTP 6](#_Toc59573160)

[Star and Snowflake schemas: 7](#_Toc59573161)

[Star schema 7](#_Toc59573162)

[Snowflake schema: 7](#_Toc59573163)

[Key differences between star schema and snowflake schema: 8](#_Toc59573164)

[Project Overview 9](#_Toc59573165)

[Aim and objective: 10](#_Toc59573166)

[TASK1: 10](#_Toc59573167)

[Reports that supports the star schema 10](#_Toc59573168)

[STAR SCHEMA MODEL 11](#_Toc59573169)

[DATA DICTIONARY 12](#_Toc59573170)

[REPORT SCREENSHOT 16](#_Toc59573171)

[TASK2: 18](#_Toc59573172)

[SCREENSHOT OF APEX WORK 18](#_Toc59573173)

[FLYU\_DATA 18](#_Toc59573174)

[FLIGHT 2017 DATA 19](#_Toc59573175)

[FLIGHT 2018 20](#_Toc59573176)

[WEATHER DATA 20](#_Toc59573177)

[STAGING 21](#_Toc59573178)

[FOR CUSTOMER, COMPLAINT AND FLIGHT 21](#_Toc59573179)

[PROCEDURE AND SEQUENCE FOR STAGING TABLE 21](#_Toc59573180)

[STAGING TABLES ARE: 22](#_Toc59573181)

[STAGING\_COMPLAINT TABLE 22](#_Toc59573182)

[STAGING\_CUSTOMER TABLE 23](#_Toc59573183)

[STAGING\_FLIGHT TABLE 24](#_Toc59573184)

[DIM TABLE SCRIPT: 25](#_Toc59573185)

[TASK3: Task 1: Extract, Transform and Load (ETL) script to populate the Star Schema (DM) with data 26](#_Toc59573186)

[EXTRACTION: 26](#_Toc59573187)

[BAD TABLES AND THEIR PROCEDURE: 26](#_Toc59573188)

[BAD\_COMPLAINT 26](#_Toc59573189)

[BAD\_CUSTOMER 27](#_Toc59573190)

[BAD FLIGHTS 28](#_Toc59573191)

[CREAETING GOOD TABELS 29](#_Toc59573192)

[GOOD TABLE AND THEIR PROCEDURE ARE: 30](#_Toc59573193)

[PROCEDURE FOR GOOD DATAS 30](#_Toc59573194)

[COMPLAINT\_GOOD 31](#_Toc59573195)

[CUSTOMER\_GOOD 32](#_Toc59573196)

[FLIGHTS\_GOOD 32](#_Toc59573197)

[RECLEANING DATAS: 33](#_Toc59573198)

[CLEAN GOOD FLIGHTS 33](#_Toc59573199)

[CLEAN BAD CUSTOMER 33](#_Toc59573200)

[TRANSFORM: 34](#_Toc59573201)

[TRANSFORM\_CUSTOMER 34](#_Toc59573202)

[TRANSFORM\_FLIGHT 35](#_Toc59573203)

[TRANSFORM\_COMPLAINT 36](#_Toc59573204)

[LOAD: 37](#_Toc59573205)

[CUSTOMER DIM 37](#_Toc59573206)

[COPLAINT DIM 38](#_Toc59573207)

[FLIGHT DIM 39](#_Toc59573208)

[TIME DIM 40](#_Toc59573209)

[FLYU FACT TABLE 41](#_Toc59573210)

[GRANULARITY OF DATA: 42](#_Toc59573211)

[Time Dimension Table 42](#_Toc59573212)

[FlyU Fact Table 43](#_Toc59573213)

[TASK4: REPORTS GENERATED IN EXCEL 44](#_Toc59573214)

[Task5: Data warehouse approaches 47](#_Toc59573215)

[Bill Inmon – Top-down Data Warehouse Design Approach: 48](#_Toc59573216)

[Ralph Kimball – Bottom-up Data Warehouse Design Approach: 49](#_Toc59573217)

[Bill Inmon Model vs Ralph Kimball Model: 52](#_Toc59573218)

[Bibliography 53](#_Toc59573219)

Data Warehouse

A data warehouse (DW) is a collection of corporate information and data derived from operational systems and external data sources. A data warehouse is designed to support business decisions by allowing data consolidation, analysis and reporting at different aggregate levels. Data is populated into the DW through the processes of extraction, transformation and loading.

In a data warehouse, data from many heterogeneous sources is extracted into a single area, transformed according to the decision support system needs and stored into the warehouse. For example, a company stores information pertaining to its employees, their salaries, developed products, customer information, sales and invoices. The CEO might want to ask a question pertaining to the latest cost-reduction measures; the answers will involve analysis of all of this data. This is a main service of the data warehouse, i.e., allowing executives to reach business decisions based on all these disparate raw data items. (Techopedia Inc, 2016)

# What is OLAP and OLTP?

## OLAP

OLAP applies complex queries to large amounts of historical data, aggregated from OLTP databases and other sources, for data mining, analytics, and business intelligenceprojects. In OLAP, the emphasis is on response time to these complex queries. Each query involves one or more columns of data aggregated from many rows. OLAP databases and data warehouses give analysts and decision-makers the ability to use custom reporting tools to turn data into information. Query failure in OLAP does not interrupt or delay transaction processing for customers, but it can delay or impact the accuracy of business intelligence insights. (Talend SA, n.d.)

## OLTP

**OLTP** (On-line Transaction Processing) is involved in the operation of a particular system. OLTP is characterized by a large number of short on-line transactions (INSERT, UPDATE and DELETE). The main emphasis for OLTP systems is put on very fast query processing, maintaining data integrity in multi-access environments and an effectiveness measured by number of transactions per second. In OLTP database there is detailed and current data, and schema used to store transactional databases is the entity model (usually 3NF). It involves Queries accessing individual record like Update your Email in Company database.

# Example of OLAP and OLTP

## OLAP

A company might compare their mobile phone sales in September with sales in October, then compare those results with another location which may be stored in a seprate database.

## OLTP

An example of OLTP system is ATM center. Assume that a couple has a joint account with a bank. One day both simultaneously reach different ATM centers at precisely the same time and want to withdraw total amount present in their bank account.

However, the person that completes authentication process first will be able to get money. In this case, OLTP system makes sure that withdrawn amount will be never more than the amount present in the bank. The key to note here is that OLTP systems are optimized for transactional superiority instead data analysis.

# Difference between OLAP and OLTP

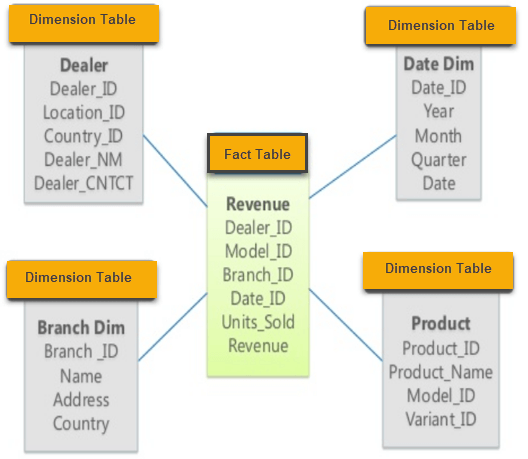
|  |  |  |
| --- | --- | --- |
| **Parameter** | **OLAP** | **OLTP** |
| Purpose | Designed for evaluation of enterprise measures by means of class and attributes. | Designed for actual time commercial enterprise operations. |
| Data source | Different OLTP databases become the source of data for OLAP. | OLTP and its transactions are the assets of data. |
| Reporting | Ad hoc, multidimensional, broadly focused reports and queries | Routine, periodic, narrowly focused reports |
| Resource requirements | Multiprocessor, large-capacity, specialized databases | Ordinary relational databases |
| Execution speed | Slow (resource intensive, complex, large-scale queries) | Fast (recording of business transactions and routine reports) |
| Query | |  | | --- | | Mostly select operations | | Insert, Update, and Delete information from the database. |
| Challenge | An OLAP cube is not an open SQL server data warehouse. Therefore, technical knowledge and experience is essential to manage the OLAP server. | Data Warehouses historically have been a development project which may prove costly to build. |

(Guru99, n.d.)

# Star and Snowflake schemas:

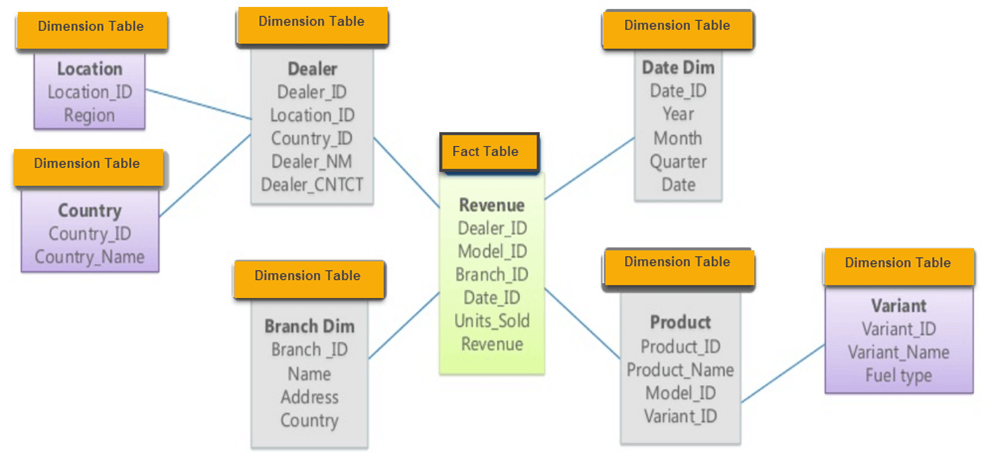
## Star schema

**Star Schema** in data warehouse, in which the center of the star can have one fact table and a number of associated dimension tables. It is known as star schema as its structure resembles a star. The Star Schema data model is the simplest type of Data Warehouse schema. It is also known as Star Join Schema and is optimized for querying large data sets. Example of Star schema:



## Snowflake schema:

**Snowflake Schema** in data warehouse is a logical arrangement of tables in a multidimensional database such that the ER diagram resembles a snowflake shape. A Snowflake Schema is an extension of a Star Schema, and it adds additional dimensions. The dimension tables are normalized which splits data into additional tables. Example of Snowflake schema:

 (Guru99, n.d.)

## Key differences between star schema and snowflake schema:

|  |  |
| --- | --- |
| **Star Schema** | **Snowflake Schema** |
| Hierarchies for the dimensions are stored in the dimensional table. | Hierarchies are divided into separate tables. |
| It contains a fact table surrounded by dimension tables. | One fact table surrounded by dimension table which are in turn surrounded by dimension table |
| In a star schema, only single join creates the relationship between the fact table and any dimension tables. | A snowflake schema requires many joins to fetch the data. |
| Simple DB Design. | Very Complex DB Design. |
| Denormalized Data structure and query also run faster. | Normalized Data Structure. |
| High level of Data redundancy | Very low-level data redundancy |
| Single Dimension table contains aggregated data. | Data Split into different Dimension Tables. |
| Cube processing is faster. | Cube processing might be slow because of the complex join. |

# Project Overview

Data warehouses are frequently thinking of as commercial enterprise talent structures created to assist with the everyday reporting wants of a business entity. They don’t have the equal real-time overall performance requirements (in widespread implementations) as OLTP records systems, and whereas OLTP systems will only incorporate the facts touching on to one small subset of the business, information warehouses seem to be to embody all records bearing on to the business.

FlyU is an airline company and they preserve information of the flights that depart every airport. They file deliberate departure and arrival times, actual departure and arrival times, from and to every airport. Information on which passengers are on every flight is recorded, some passengers will need a range of flights to attain their vacation spot – this is known as a trip and additionally includes important points of return flights. Also recorded is the lead and deputy pilot for the flight. The tail range is the identification for the area (similar to a car registration number).

Complaints raised by passengers are recorded in opposition to flights, passengers are required to increase a grievance in opposition to every flight if they omit connections due to delays.

The machine provides facts for the airline on flight delays which they use in their planning and customer complaints service. Data mart now wishes to enforce the data warehouse to analyze and file various measures/ facts to make sure they are reaching their targets and accomplishing their goals. They desire to check out three fundamental key performance indicators:

* **Deliver a quality service**
* **Ensure customer satisfaction**
* **Grow the company**

Among the three key performance indicators, the project focuses on building the warehouse whose primary aim to assist Data Mart in keeping up the latest development in technology.

## Aim and objective:

The primary aim of this project is to design, develop and implement a data ware house for FlyU Company that aids the company to be aware of the latest developments in technology and ensure that they can meet the demands of the technology market. The major objective of the project are:-

* To identify the major reports/ information the company would requires in order to be up-to-date with the trend in technological development.
* To design the star schema of the data warehouse based on the requirement of the reports to be produced.
* To implement and develop the data warehouse on the basis of designed star schema to set up database system as the OLTP source system
* To extract the data from source, transform the data as required based on the data warehouse design and load them into the data warehouse using ETL scripts
* To produce required report using Ms. excel.
* To create the dashboard using oracle apex for easy access to information and better readability of the reports.

# TASK1:

## Reports that supports the star schema

* Total number of complaint in a month.
* Total number of compensation amount per month.
* Total arrival delay per month.
* Total number of flights in a year.
* Total departure delay per month.

## STAR SCHEMA MODEL



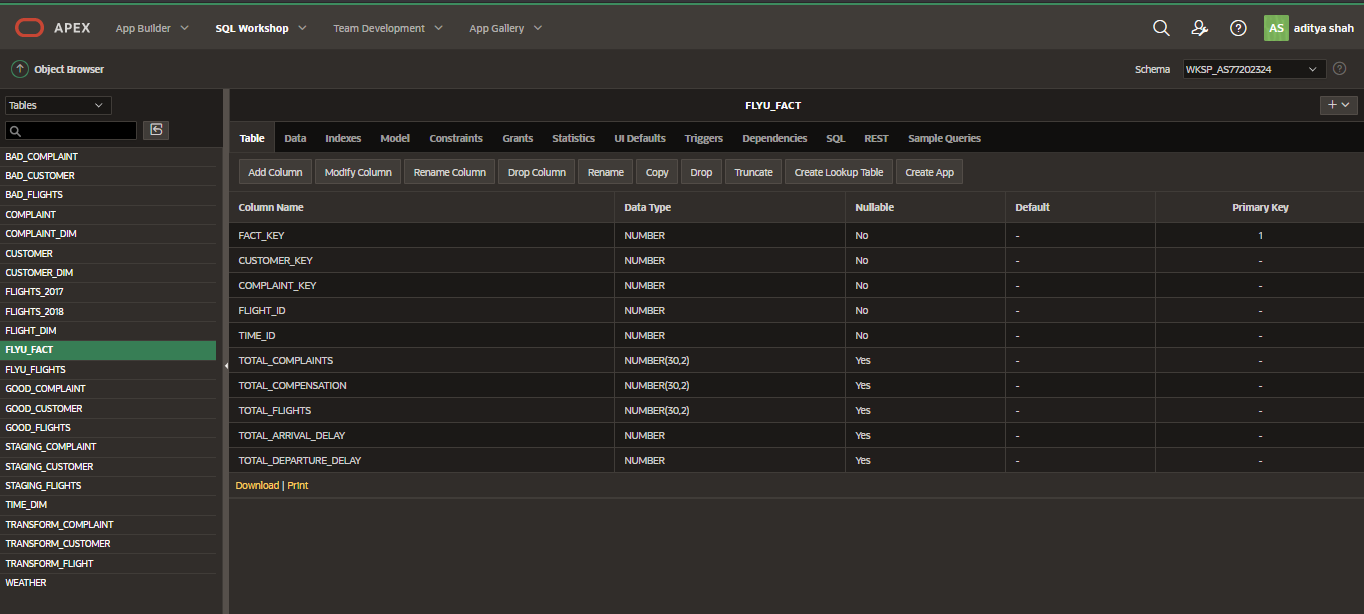
Star schema model design to support the report.

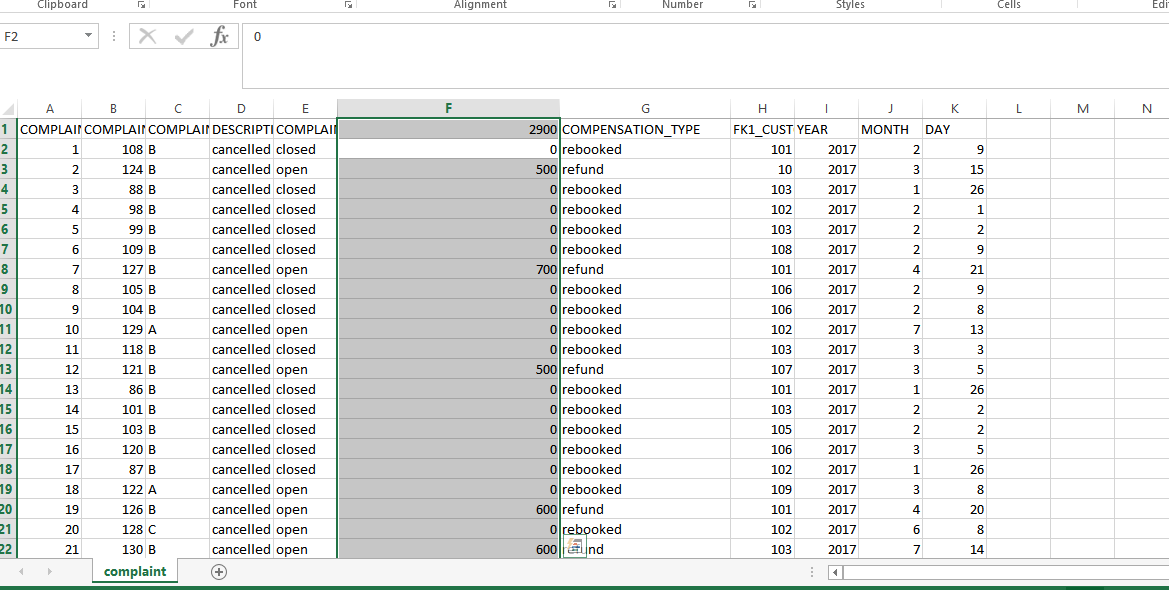
## DATA DICTIONARY

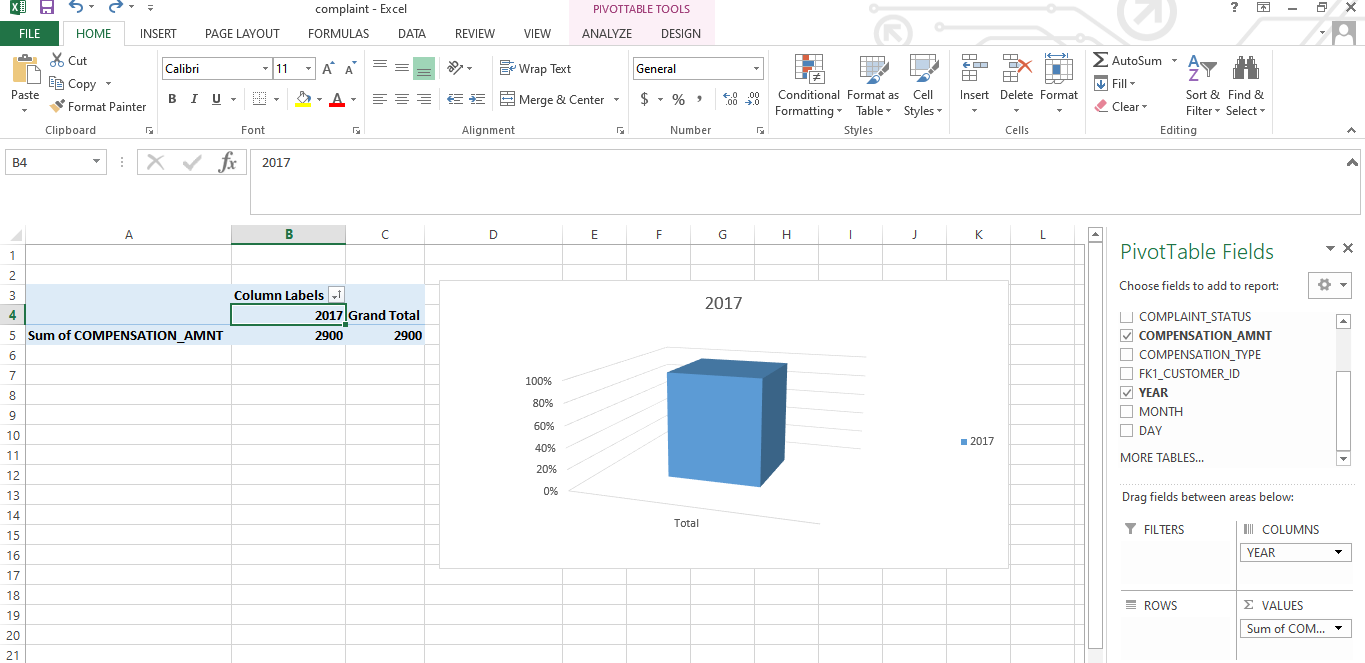
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Star schema table** | | | **Attribute name** | | | **Data Type** | | | **Key** | | | **DQ Source** | | | **Data source field (mapping)** | | | **Data quality Issues** | | | Transformation |
| **COMPLAINT\_DIM** | | | complaint\_key | | | Numeric | | | Yes | | | Automatically generated as primary key | | | n/a | | | n/a | | | Create a sequence complaint\_seq to support the generation of primary keys. |
| complaint\_id | | | number | | | No | | | Automatically generated as a surrogate key | | | n/a | | | n/a | | |  |
| complaint\_type | | | VARCHAR | | | No | | | FlyU\_flights.sql | | | FlyU\_flight.type | | | Some complaint types are missing. | | | n/a |
| description | | | VARCHAR | | | No | | | FlyU\_flights.sql | | | FlyU\_flights.description | | | Some description is missing. | | | n/a |
| complaint\_status | | | VARCHAR | | | No | | | FlyU\_flights.sql | | | FlyU\_flights.complaint\_status | | | No Data quality issues | | | n/a |
| compensation\_amnt | | | Number | | | No | | | FlyU\_flights.sql | | | FlyU\_flights.compensation\_amnt | | | No data quality issues. | | | n/a |
| compensation\_type | | | VARCHAR | | | No | | | FlyU\_flights.sql | | | FlyU\_flights.compensation\_type | | | Some compensation type data is missing. | | | n/a |
| Fk1\_Customer\_id | | | number | | | No | | | FlyU\_flights.sql | | | n/a | | | No data quality issues. | | | Foreign key |
|  | | | Year | | | Number | | | No | | | 2017\_3000.xlsx, Flight\_2018\_2000\_rows.xlsx | | | 2017\_3000.year and Flight\_2018\_2000\_rows.year | | | No data quality issues. | | | n/a |
|  | | | month | | | Numeric | | | No | | | Flight\_2018\_2000\_rows.xlsx | | | Flight\_2018\_2000\_rows.month | | | No data quality issues. | | | n/a |
|  | | | DAY | | | Number | | |  | | | Flight\_2018\_2000\_rows.xlsx | | | Flight\_2018\_2000\_rows.day | | | No data quality issues. | | | n/a |
| Definition: | | | The complaint dimension table holds data from the complaint table as given in the FlyU\_flights.sql script. The table includes data for description, complaint\_status etc. | | | | | | | | | | | | | | | | | | |
| Notes: | | |  | | | | | | | | | | | | | | | | | | |
| **Star schema table** | | | **Attribute name** | | | **Data Type** | | | **Key** | | | **DQ Source** | | | **Data source field (mapping)** | | | **Data quality Issues** | | | **Transformation** |
| **TIME\_DIM** | | | time\_id | | | Numeric | | | Yes | | |  | | | n/a | | | n/a | | | Create a sequence SEQUENCE\_DIM\_TIME to support the generation of primary key. |
| year | | | Numeric | | | No | | | 2017\_3000.xlsx, Flight\_2018\_2000\_rows.xlsx | | | 2017\_3000.year and Flight\_2018\_2000\_rows.year | | | No data quality issues. | | | n/a |
| month | | | Numeric | | | No | | | Flight\_2018\_2000\_rows.xlsx | | | Flight\_2018\_2000\_rows.month | | | No data quality issues. | | | n/a |
|  | | | Day | | | numeric | | |  | | | Flight\_2018\_2000\_rows.xlsx | | | Flight\_2018\_2000\_rows.day | | |  | | | n/a |
| Definition: | | | The time dimension table holds the intervals of time for which the data will be held. It is held at year level meaning now reports are available at a lower granularity. | | | | | | | | | | | | | | | | | | |
| Notes: | | |  | | | | | | | | | | | | | | | | | | |
| **Star schema table** | | | **Attribute name** | | | **Data Type** | | | **Key** | | | **DQ Source** | | | **Data source field (mapping)** | | | **Data quality Issues** | | | **Transformation** |
| **CUSTO**  **MER\_DIM** | | | customer\_key | | | Number | | | Yes | | | Automatically generated as primary key | | | n/a | | | n/a | | | Create a sequence customer\_seq to support the generation of primary key. |
| customer\_id | | | Numeric | | | No | | | Automatically generated as a surrogate key | | | n/a | | | n/a | | | Create a sequence customer\_id\_seq to support the generation of surrogate key. |
| CUSTOMER\_ZIP\_CODE | | | VARCHAR | | | No | | | FlyU\_flights.sql | | | FlyU\_flights.customer\_miles | | | No data quality issues | | | n/a |
| customer\_type | | | VARCHAR | | | No | | | FlyU\_flights.sql | | | FlyU\_flights.customer\_type | | | No data quality issues. | | | n/a |
|  | | | BUSINESS | | | VARCHAR | | |  | | | FlyU\_flights.sql | | | FlyU\_flights.business | | | n/a | | | n/a |
| Definition: | | | The customer dimension table holds the data from FlyU\_flights.sql script which contains data for customer\_type , customer\_zip\_ code and BUSINESS | | | | | | | | | | | | | | | | | | |
| Notes: | | |  | | | | | | | | | | | | | | | | | | |
| **Star schema table** | | **Attribute name** | | | **Data Type** | | | **Key** | | | **DQ Source** | | | **Data source field (mapping)** | | | **Data quality Issues** | | | **Transformation** | |
| **FLIGHT\_DIM** | | flight\_id | | | Numeric | | | Yes | | | Automatically generated as a primary key | | | n/a | | | n/a | | | Create a sequence flights\_seq to support the generation of primary keys. | |
| destination\_airport | | | INTEGER | | | No | | | 2017\_3000.xlsx, Flight\_2018\_2000\_rows.xlsx | | | 2017\_3000.DESTINATION\_AIRPORT and Flight\_2018\_2000\_rows.DESTINATION\_AIRPORT | | | No data quality issues. | | | n/a | |
| origin\_airport | | | INTEGER | | | No | | | 2017\_3000.xlsx, Flight\_2018\_2000\_rows.xlsx | | | 2017\_3000.ORIGIN\_AIRPORT and Flight\_2018\_2000\_rows.ORIGIN\_AIRPORT | | | No data quality issues. | | | n/a | |
| Year | | | Numeric | | | No | | | 2017\_3000.xlsx, Flight\_2018\_2000\_rows.xlsx | | | 2017\_3000.year and Flight\_2018\_2000\_rows.year | | | Some data is missing. | | | n/a | |
| Month | | | Numeric | | | No | | | 2017\_3000.xlsx, Flight\_2018\_2000\_rows.xlsx | | | Flight\_2018\_2000\_rows.Month | | | Some data is missing. | | | n/a | |
| flight\_number | | | Numeric | | | No | | | 2017\_3000.xlsx, Flight\_2018\_2000\_rows.xlsx | | | 2017\_3000.flight\_Number and Flight\_2018\_2000\_rows. flight\_Number | | | Some data is missing. | | | n/a | |
| Canclled\_Reason | | | VARCHAR | | | No | | | 2017\_3000.xlsx, Flight\_2018\_2000\_rows.xlsx | | | Flight\_2018\_2000\_rows.CANCELLATION\_REASON | | | Some data is missing. | | | n/a | |
| tail\_number | | | VARCHAR | | | No | | | 2017\_3000.xlsx, Flight\_2018\_2000\_rows.xlsx | | | 2017\_3000.TAIL\_NUMER and Flight\_2018\_2000\_rows.TAIL\_NUMBER | | | Some data is missing | | | n/a | |
| Data\_Source | | | VARCHAR | | | No | | | 2017\_3000.xlsx, Flight\_2018\_2000\_rows.xlsx | | | n/a | | | No data is missing | | | n/a | |
| arrival\_delay | | | INTEGER | | | No | | | 2017\_3000.xlsx, Flight\_2018\_2000\_rows.xlsx | | | 2017\_3000.ARRIVAL\_DELAY and Flight\_2018\_2000\_rows.ARRIVAL\_DELAY | | | Some data is missing | | | n/a | |
| cancelled | | | Numeric | | | No | | | FlyU\_flights.sql | | | 2017\_3000.CANCELLED and Flight\_2018\_2000\_rows.CANCELLED | | | All data are null. | | | n/a | |
| Definition: | | The flights dimension table holds data from 2015\_3000.xlsx, Flight\_2018\_2000\_rows.xlsx and FlyU\_flights\_5\_sept.sql data sources which contains data for arrival\_delay, tail\_number etc. | | | | | | | | | | | | | | | | | | | |
| Notes: | |  | | | | | | | | | | | | | | | | | | | |
| **Star schema table** | **Attribute name** | | | **Data Type** | | | **Key** | | | **DQ Source** | | | **Data source field (mapping)** | | | **Data quality Issues** | | | **Transformation** | | |
| **FLYU\_FACT** | flight\_Flyu\_key | | | INTEGER | | | Yes | | | Automatically generated as a primary key | | | n/a | | | n/a | | | Create a sequence outcome\_seq to support the generation of primary key. | | |
| customer\_key | | | Numeric | | | No | | | Automatically generated as a foreign key | | | n/a | | | n/a | | | Create a sequence customer\_seq to support the generation of primary key. | | |
| complaint\_key | | | Numeric | | | No | | | Automatically generated as a primary key | | | n/a | | | n/a | | | Create a sequence complaint\_seq to support the generation of primary key. | | |
| flight\_id | | | Numeric | | | No | | | Automatically generated as a primary key | | | n/a | | | n/a | | | Create a sequence flights\_seq to support the generation of primary key. | | |
| time\_id | | | INTEGER | | | No | | | Automatically generated as a primary key | | | n/a | | | n/a | | | Create a sequence time\_seq to support the generation of primary key. | | |
|  | total\_complaint | | | INTEGER | | | No | | | FlyU\_flights.sql | | |  | | | Some data missing. | | | n/a | | |
|  | total\_Compensation | | | INTEGER | | | No | | | FlyU\_flights.sql | | |  | | | Some data missing. | | | n/a | | |
|  | total\_flights | | | INTEGER | | | No | | | 2017\_3000.xlsx, Flight\_2018\_2000\_rows.xlsx | | |  | | | Some data missing. | | | n/a | | |
|  | total\_arrival\_delay | | | INTEGER | | | No | | | 2017\_3000.xlsx, Flight\_2018\_2000\_rows.xlsx | | |  | | | Some data missing. | | | n/a | | |
|  | Total\_departure\_delay | | |  | | |  | | |  | | |  | | |  | | |  | | |
| Definition: | The fact\_flight\_outcome fact table contains data from all the given sources. It contains the foreign key of all the dimension table in the star schema and also contains data like total\_cancelled, total compensation etc. | | | | | | | | | | | | | | | | | | | | |
| Notes: |  | | | | | | | | | | | | | | | | | | | | |

## REPORT SCREENSHOT

EXPECTED DATA







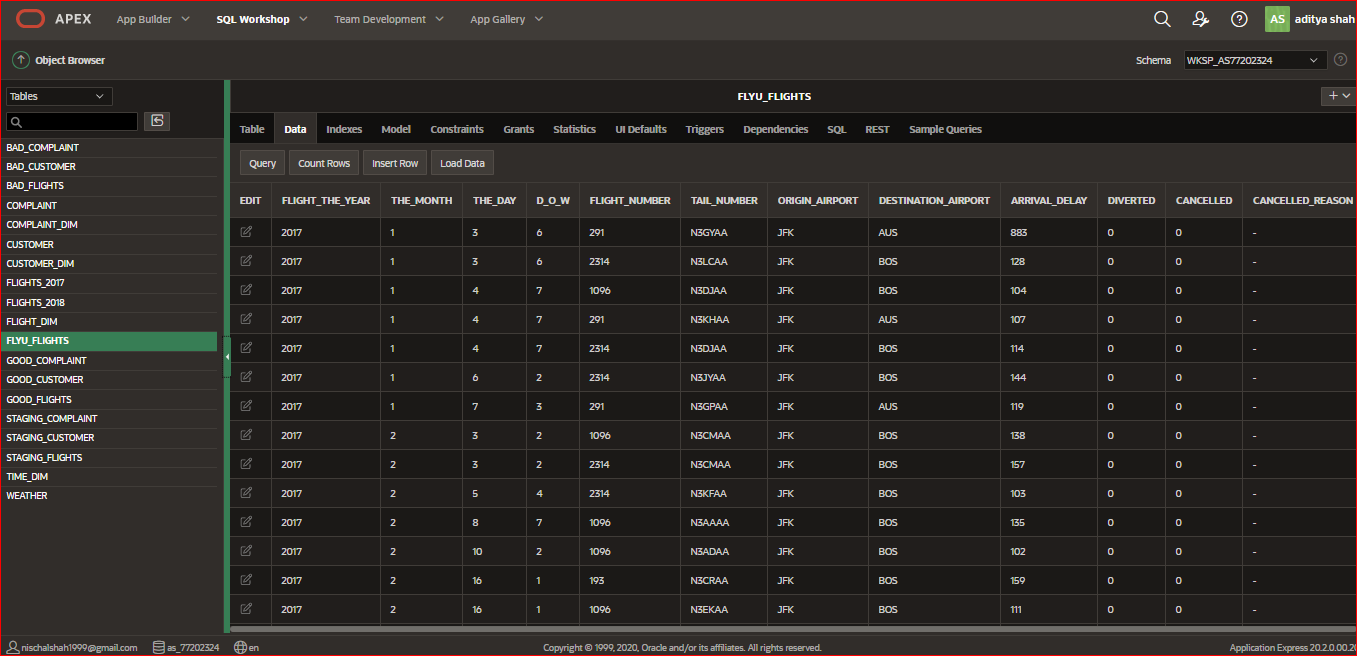
Total number of complaints in year 2017.

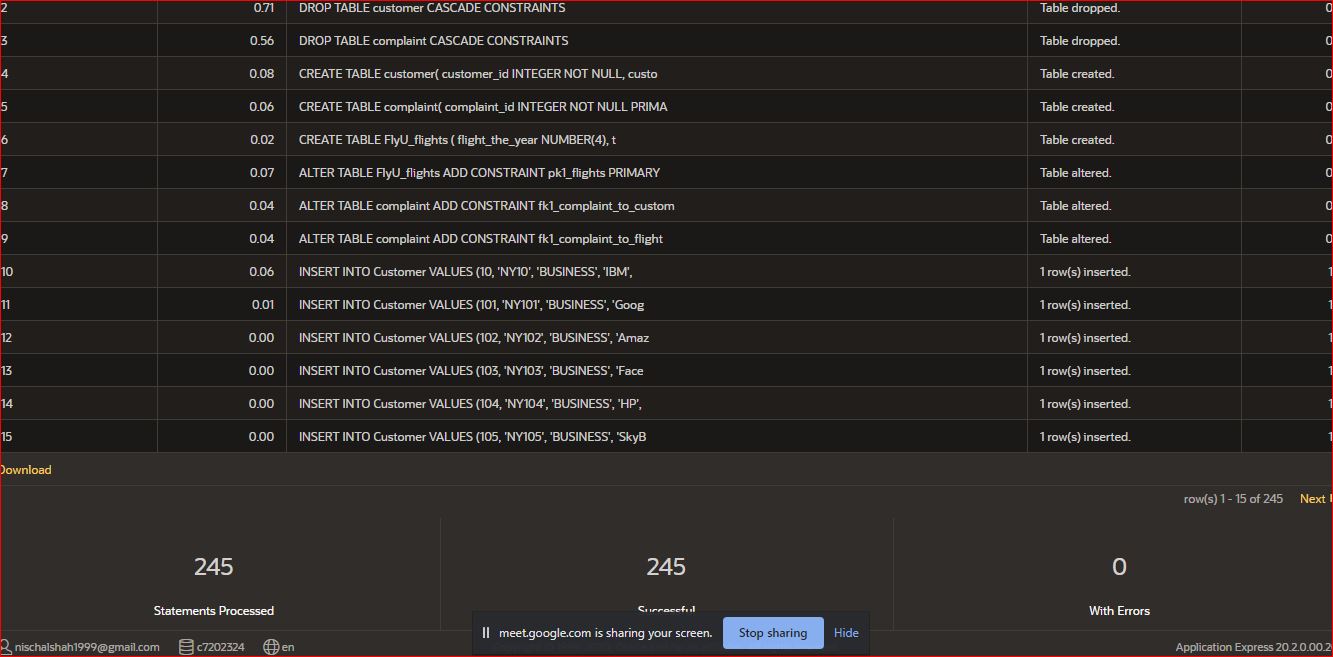
# TASK2:

## SCREENSHOT OF APEX WORK

## FLYU\_DATA

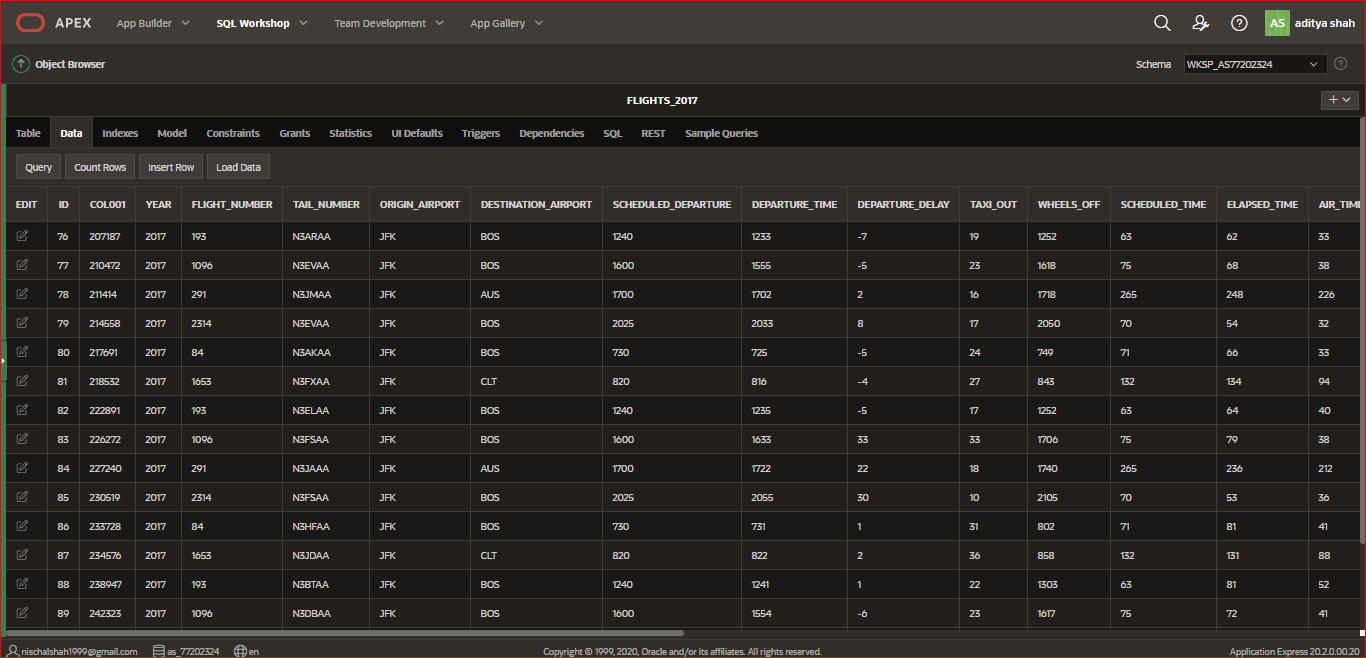
Uploading and running flyU flights script.



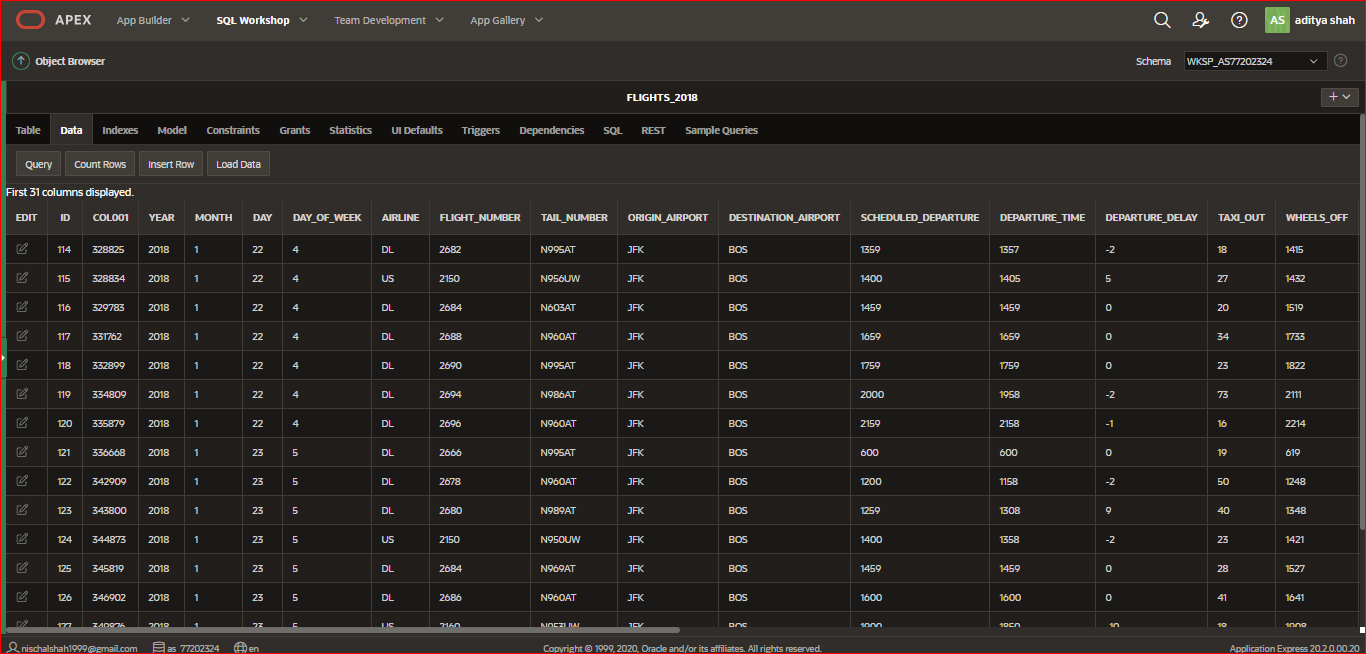


## FLIGHT 2017 DATA

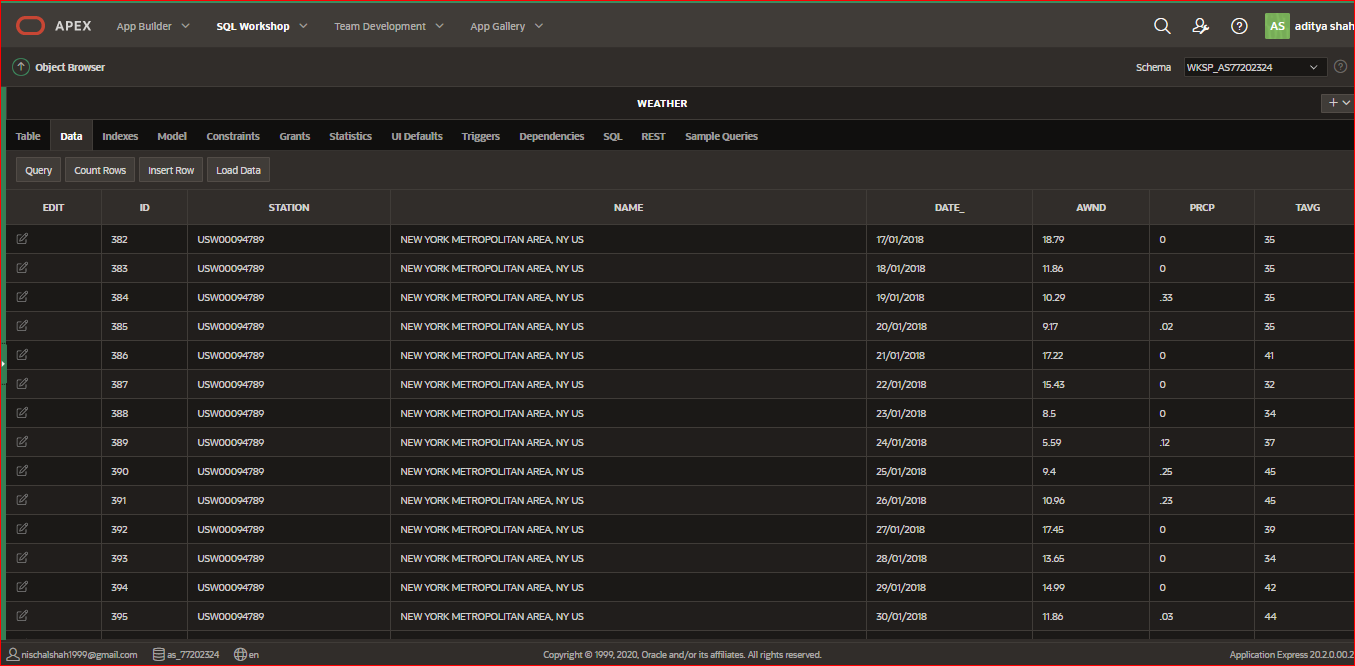
Uploaded and run the data of flights from source of 2017 excel sheet.



## FLIGHT 2018

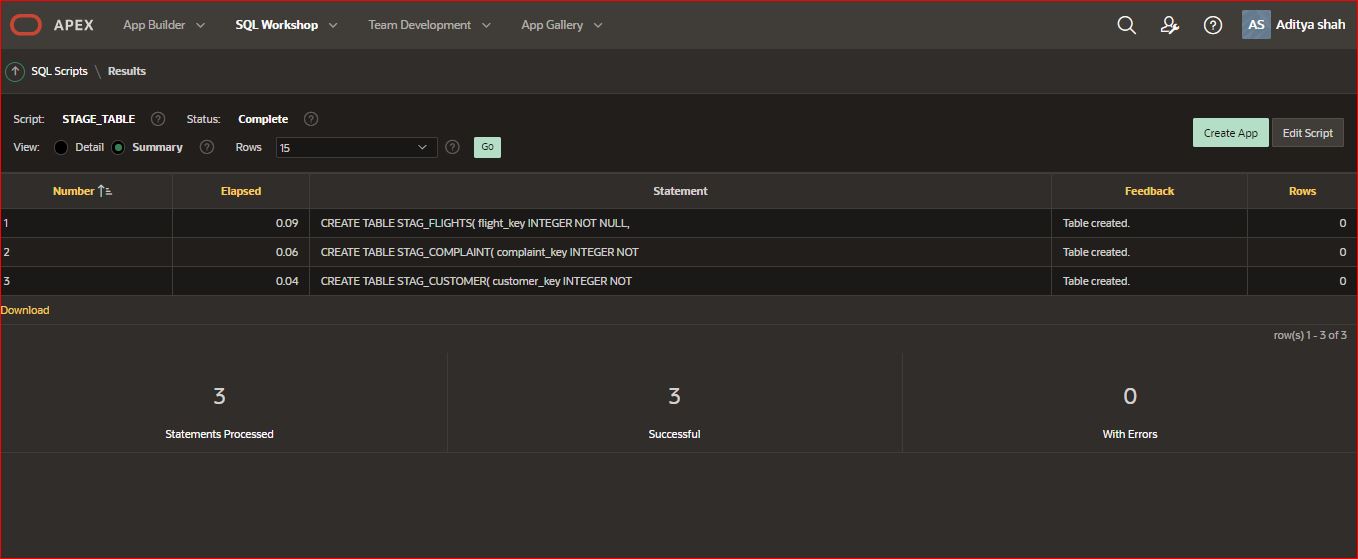
  
Uploaded and run the data of flights from source 2018 excel sheet.

## WEATHER DATA



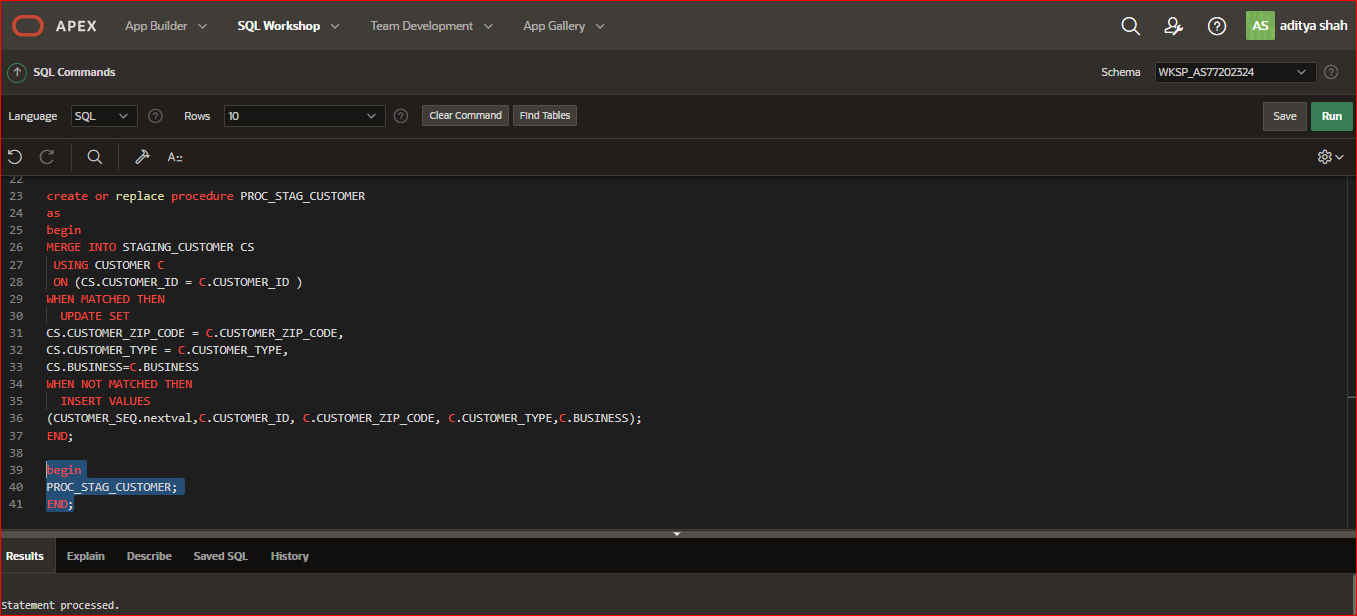
# STAGING

## FOR CUSTOMER, COMPLAINT AND FLIGHT



All three staging tables are created.

## PROCEDURE AND SEQUENCE FOR STAGING TABLE

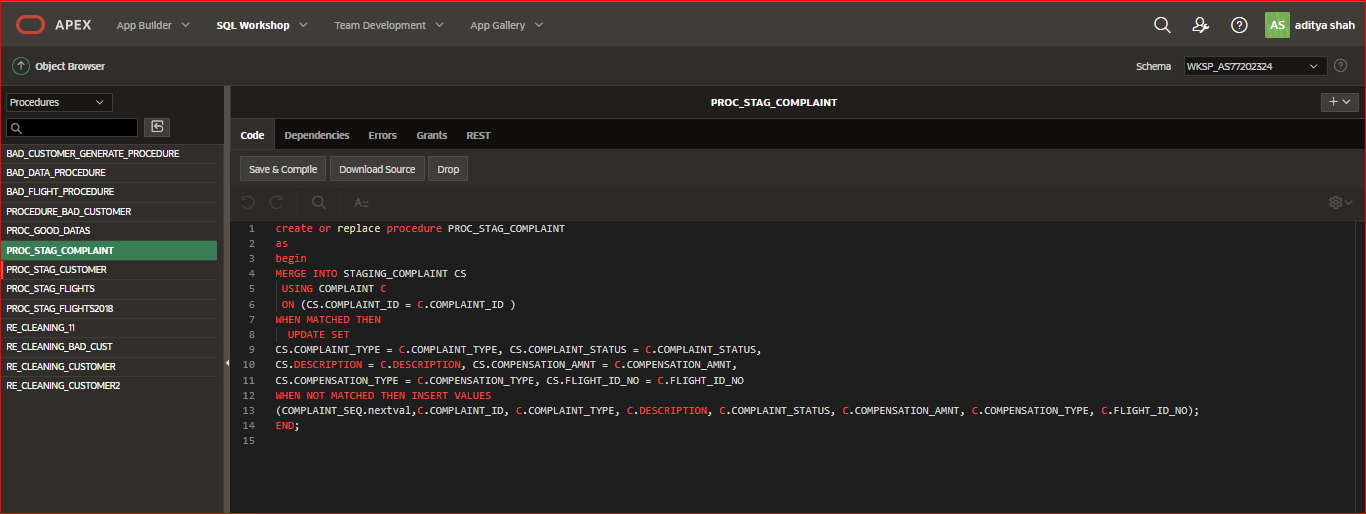


Procedure and sequence were created for all staging table.

# STAGING TABLES ARE:

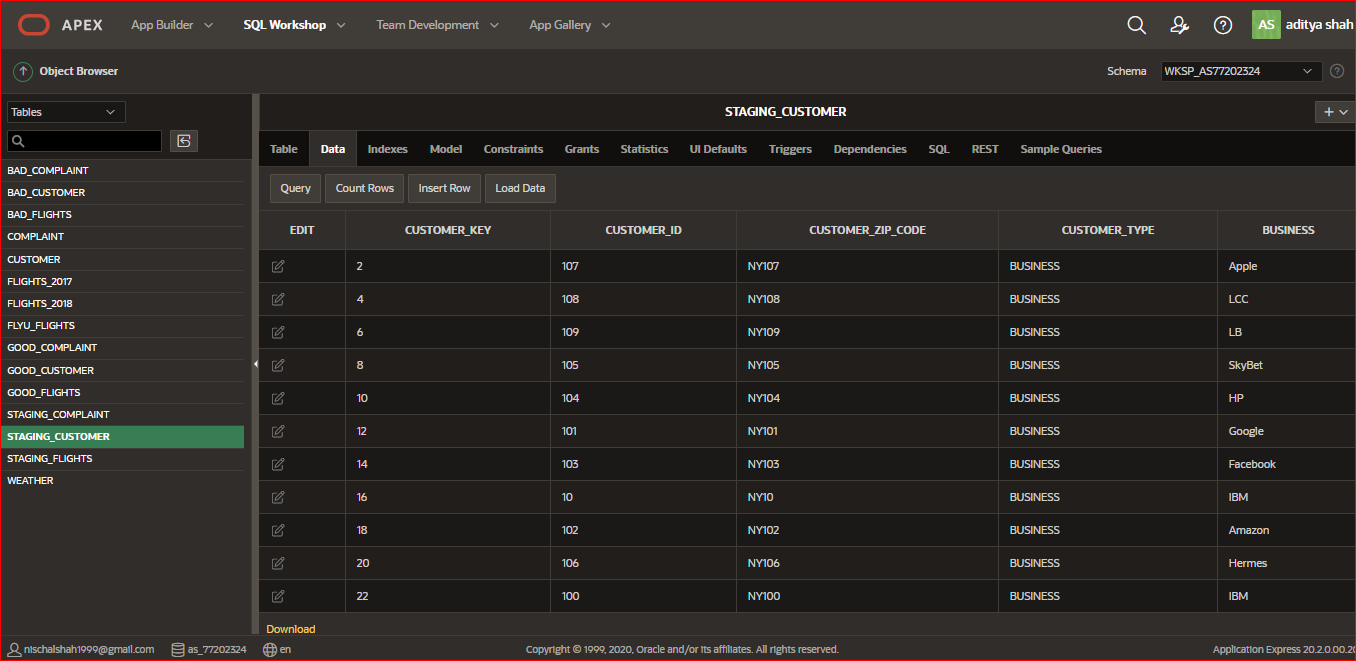
## STAGING\_COMPLAINT TABLE

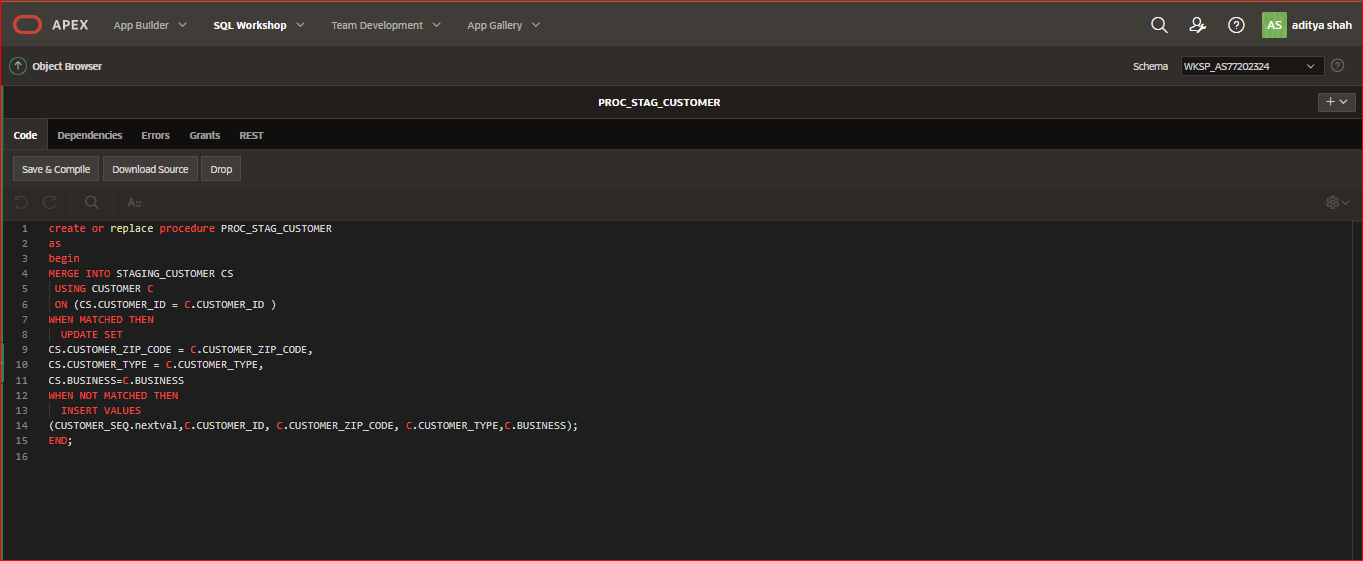




Staging complaint table with its data extracted from complaint table by creating procedure.

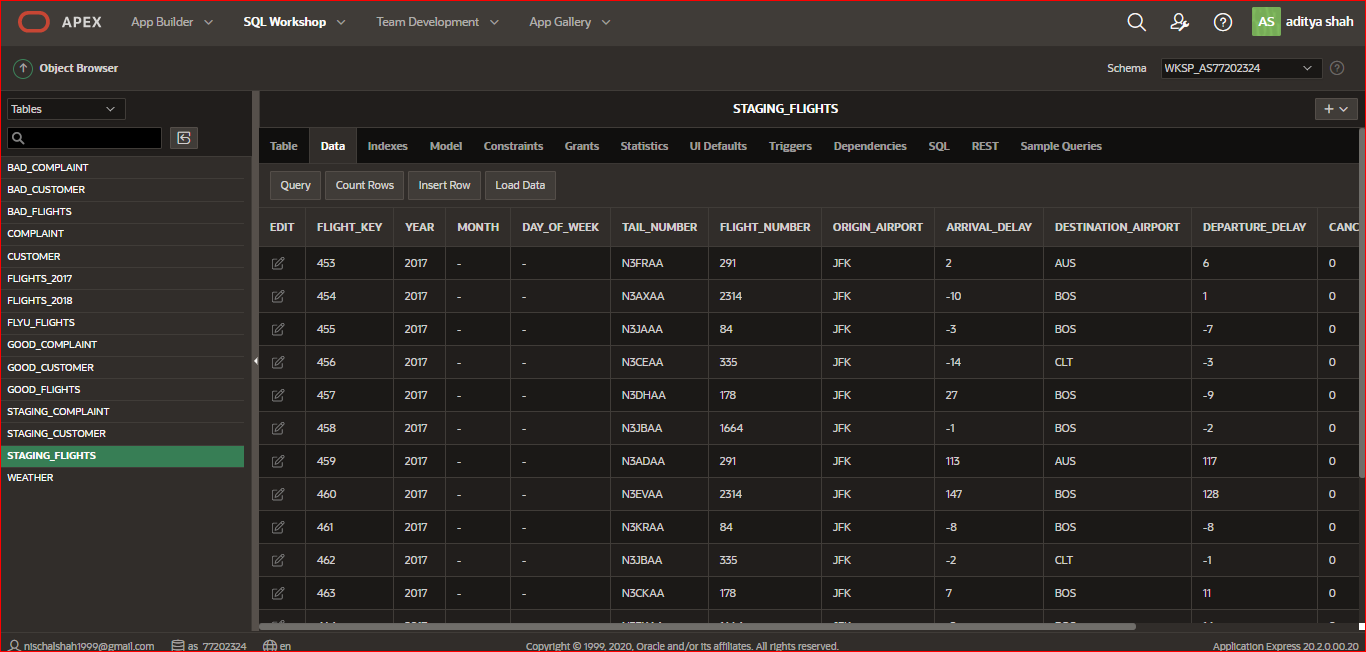
## STAGING\_CUSTOMER TABLE

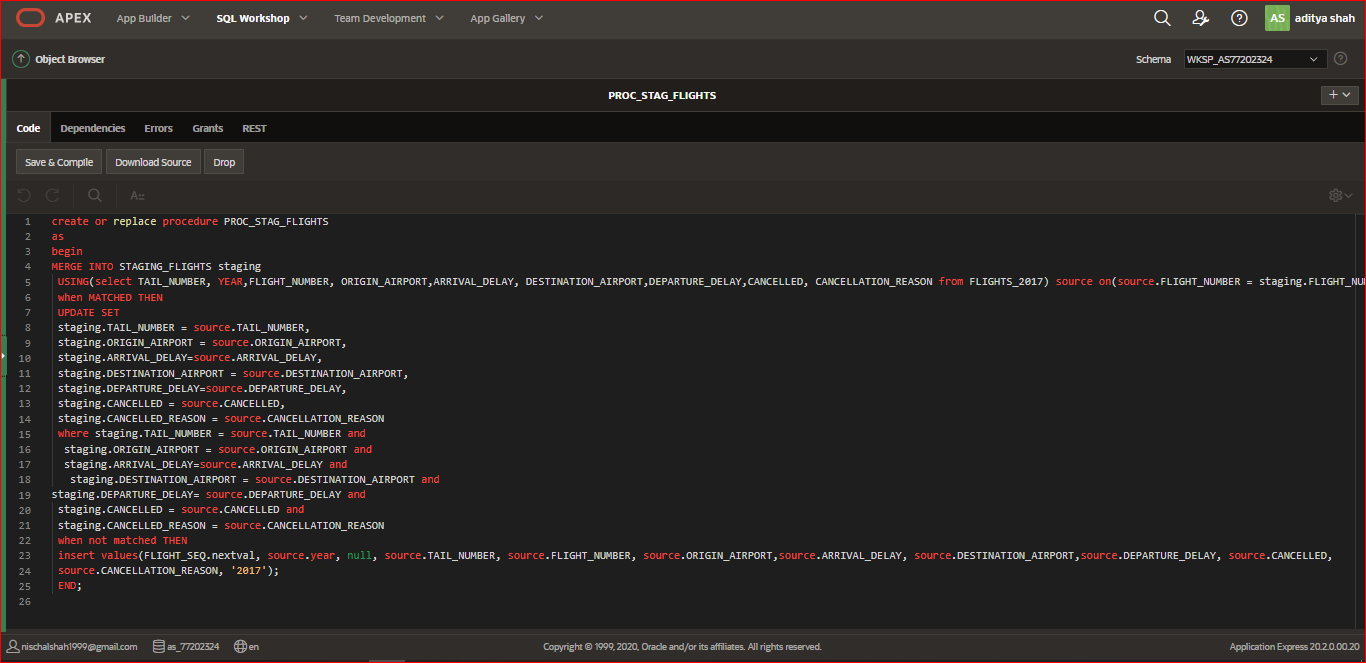




Staging customer table with respective data extracted from customer table using procedure.

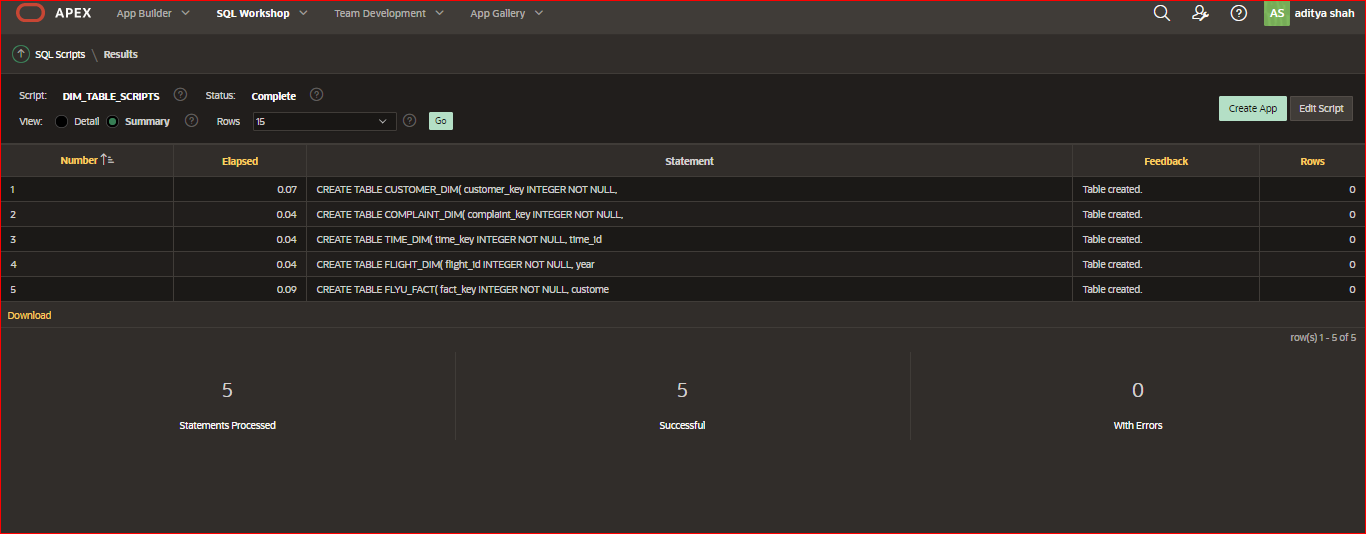
## STAGING\_FLIGHT TABLE





Staging flight table with data extracted from flyU flights table using procedure.

# DIM TABLE SCRIPT:



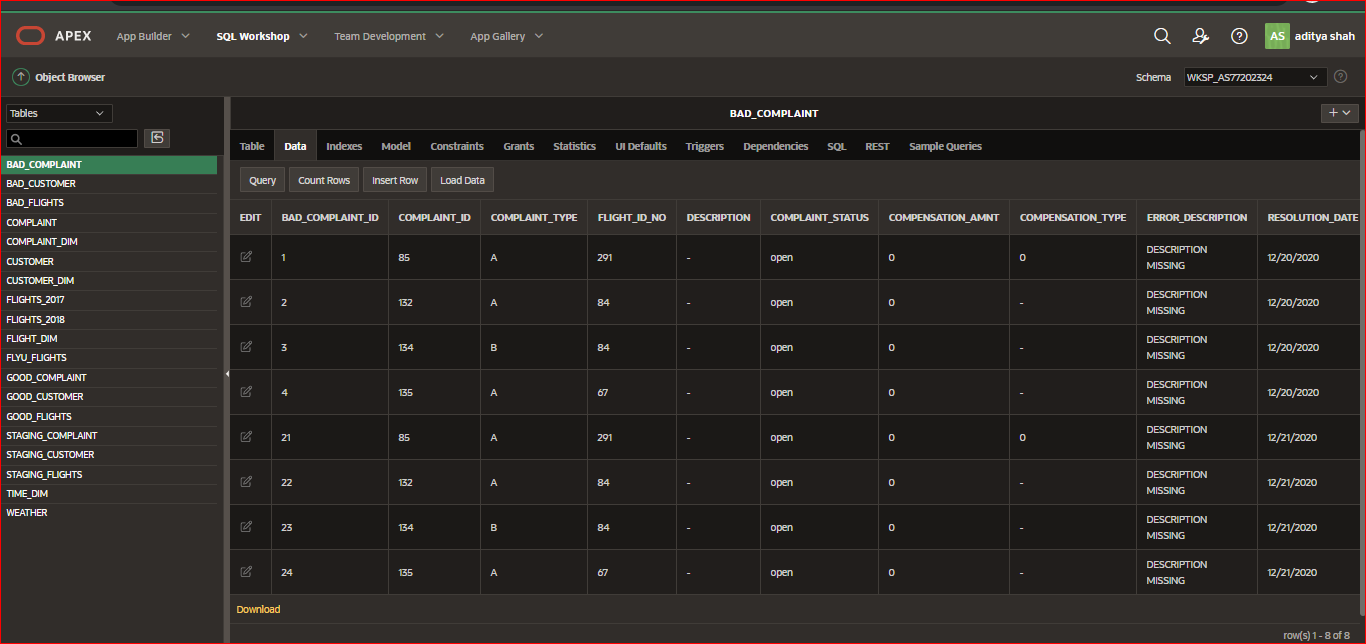
All three dimensional table are created with the sql script generated from QSEE.

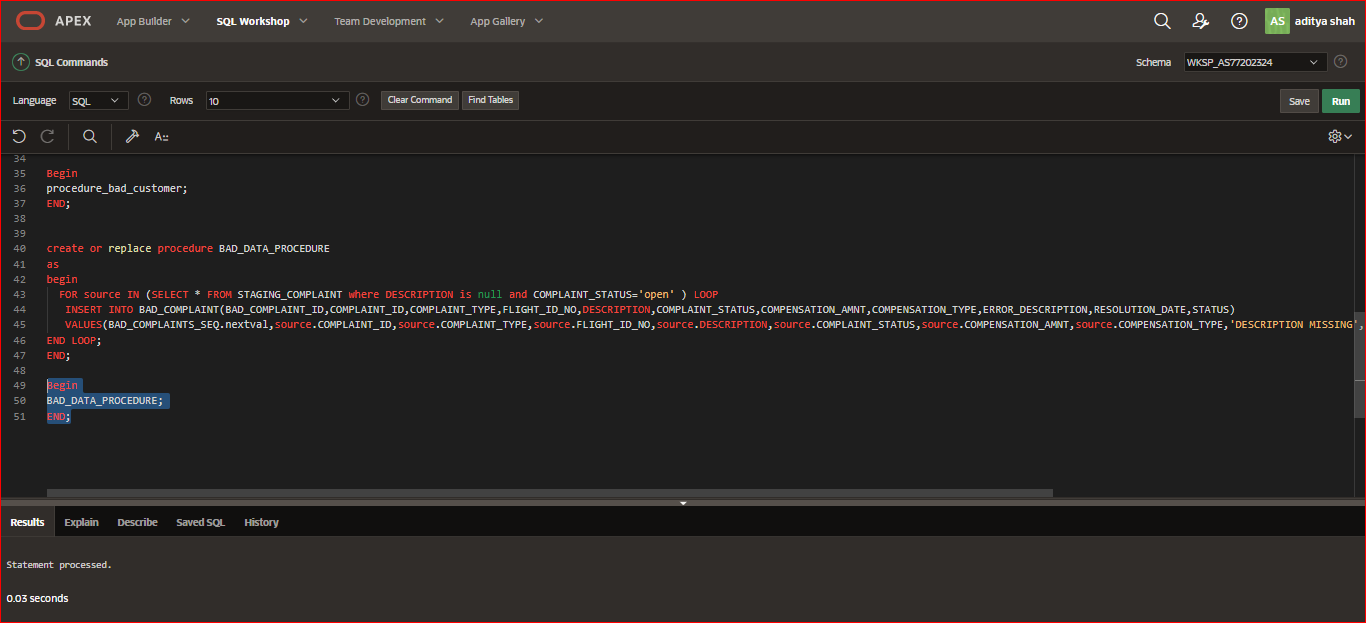
# TASK3: Task 1: Extract, Transform and Load (ETL) script to populate the Star Schema (DM) with data

# EXTRACTION:

## BAD TABLES AND THEIR PROCEDURE:

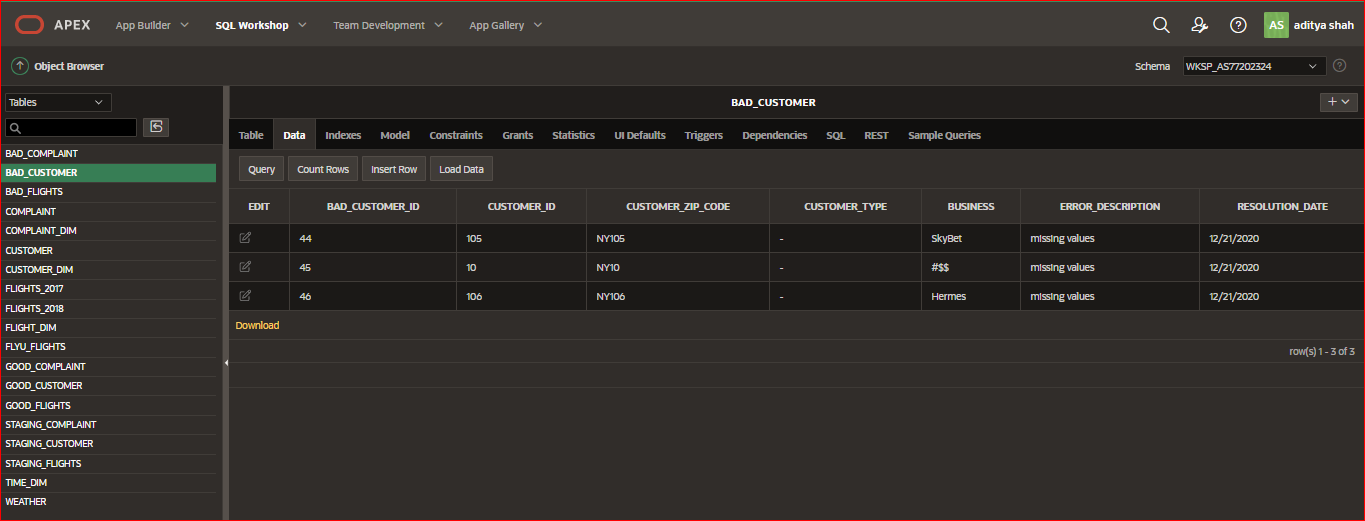
## BAD\_COMPLAINT

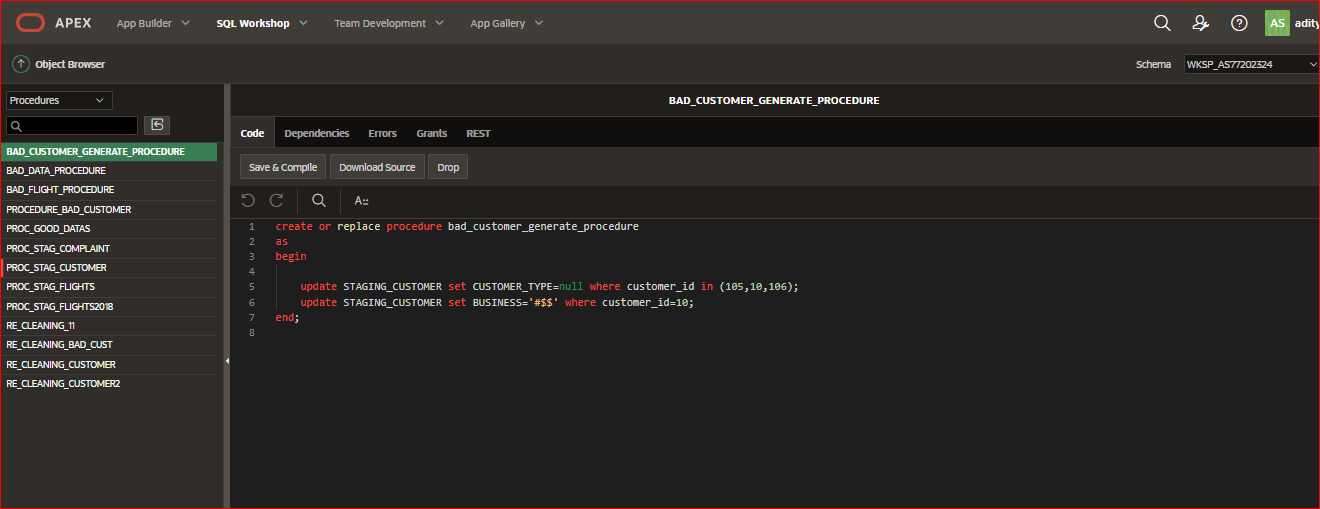




Bad data is extracted in bad complaint table from staging table by creating procedure.

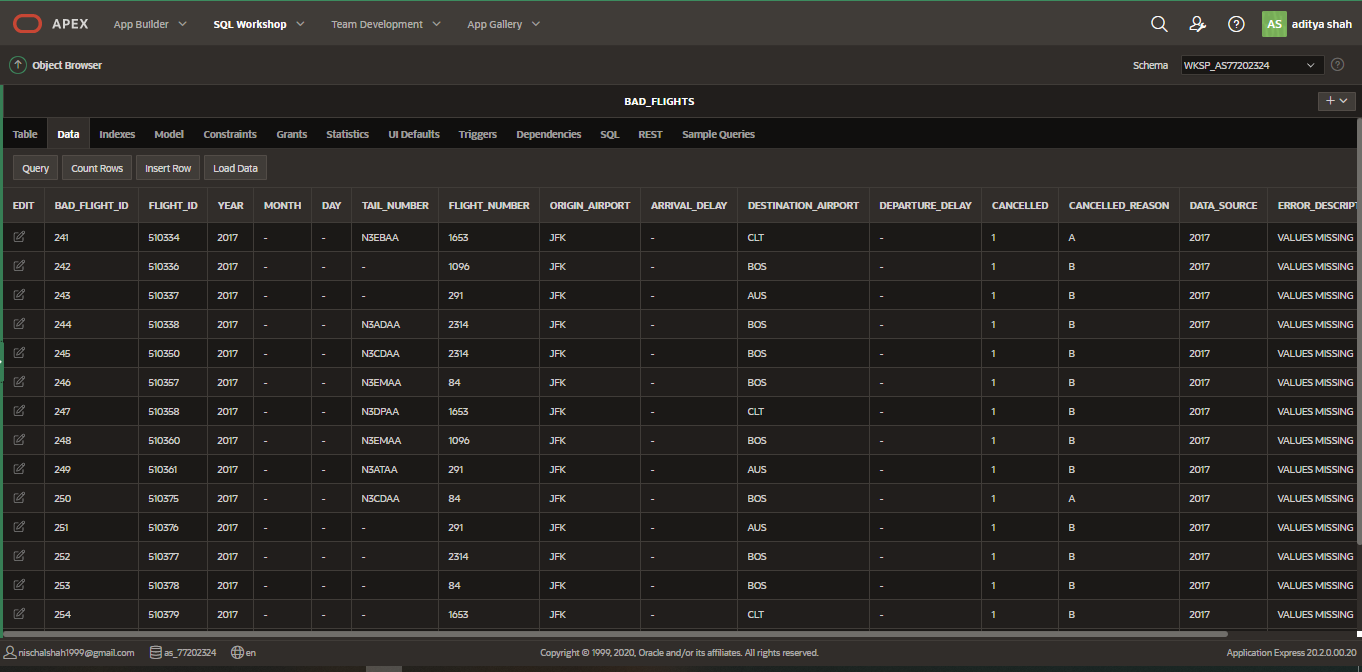
## BAD\_CUSTOMER

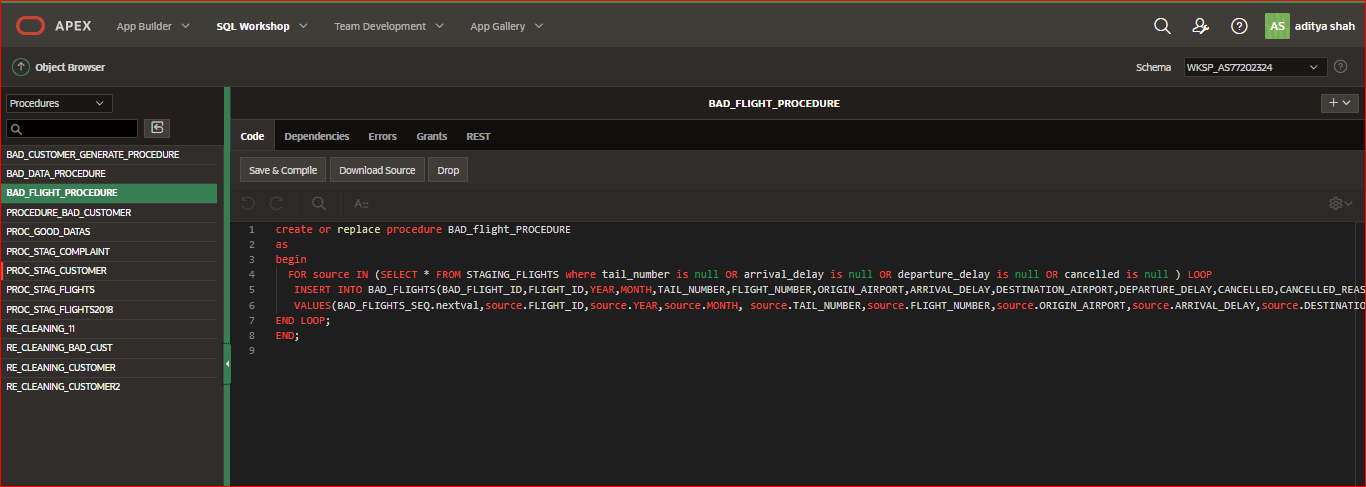




Procedure is generated to create bad data in bad customer table.

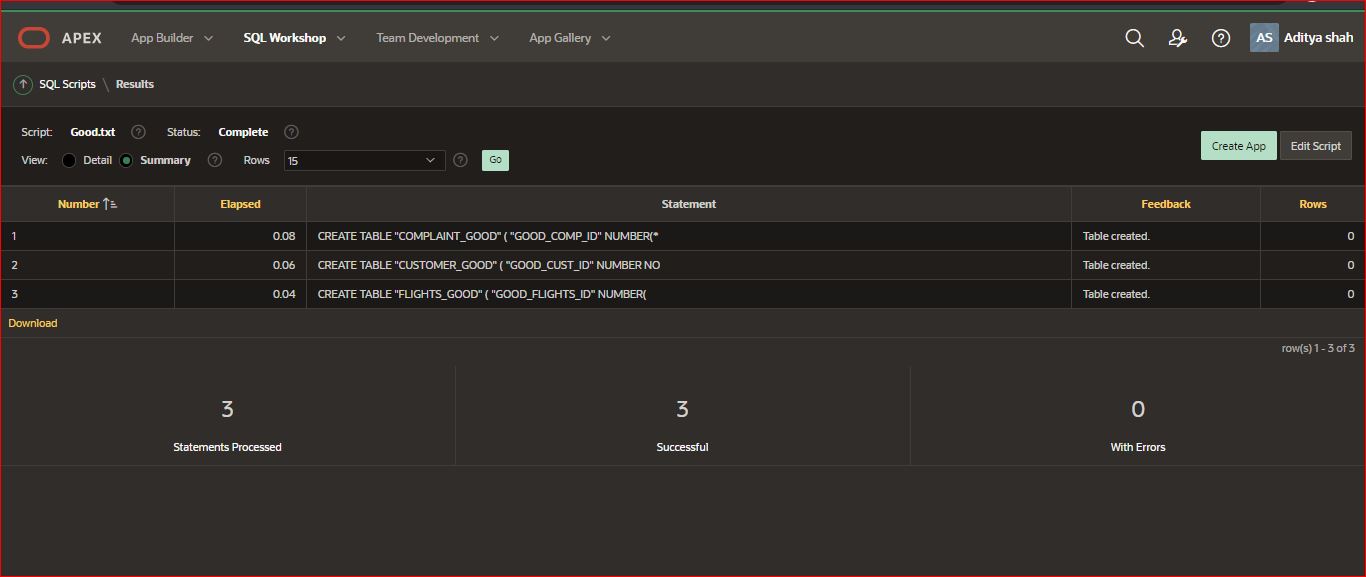
## BAD FLIGHTS





Created procedure for bad flights table to extract the bad data from staging table.

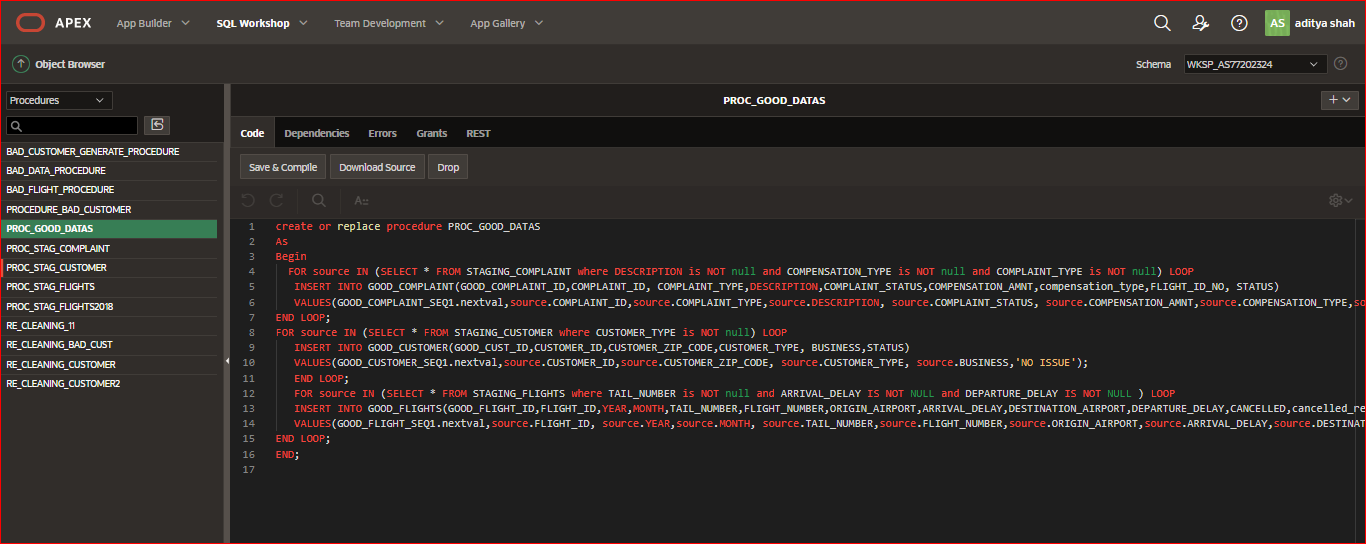
# CREAETING GOOD TABELS

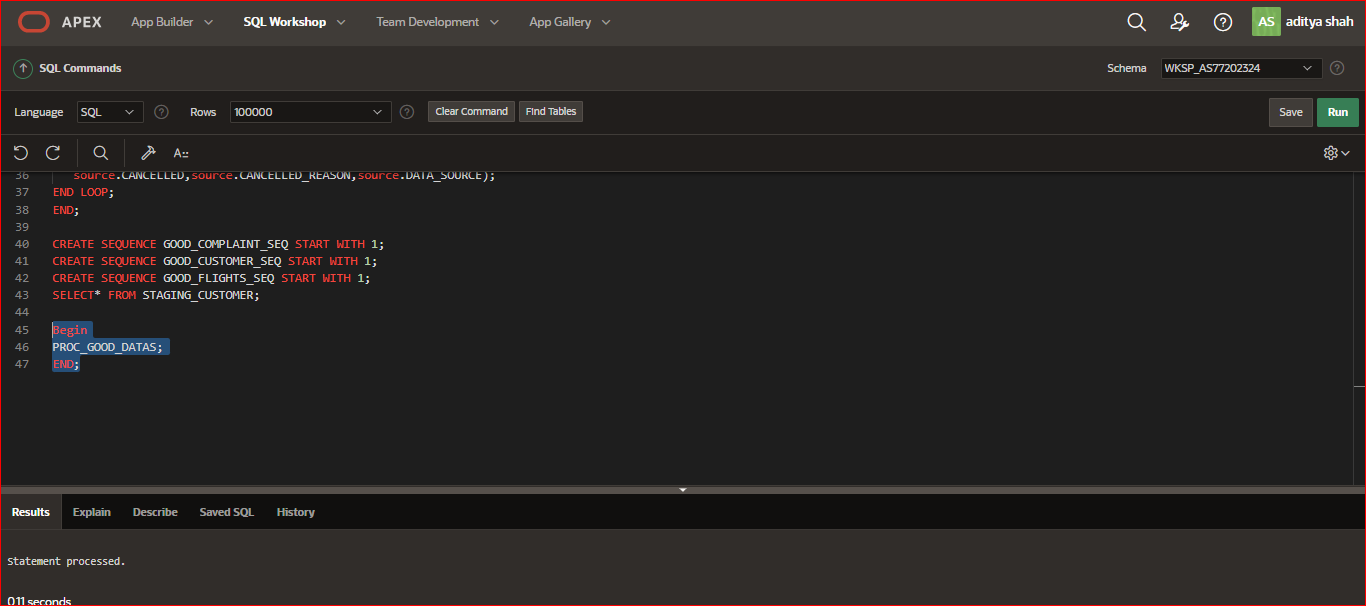


All three good tables are generated.

# GOOD TABLE AND THEIR PROCEDURE ARE:

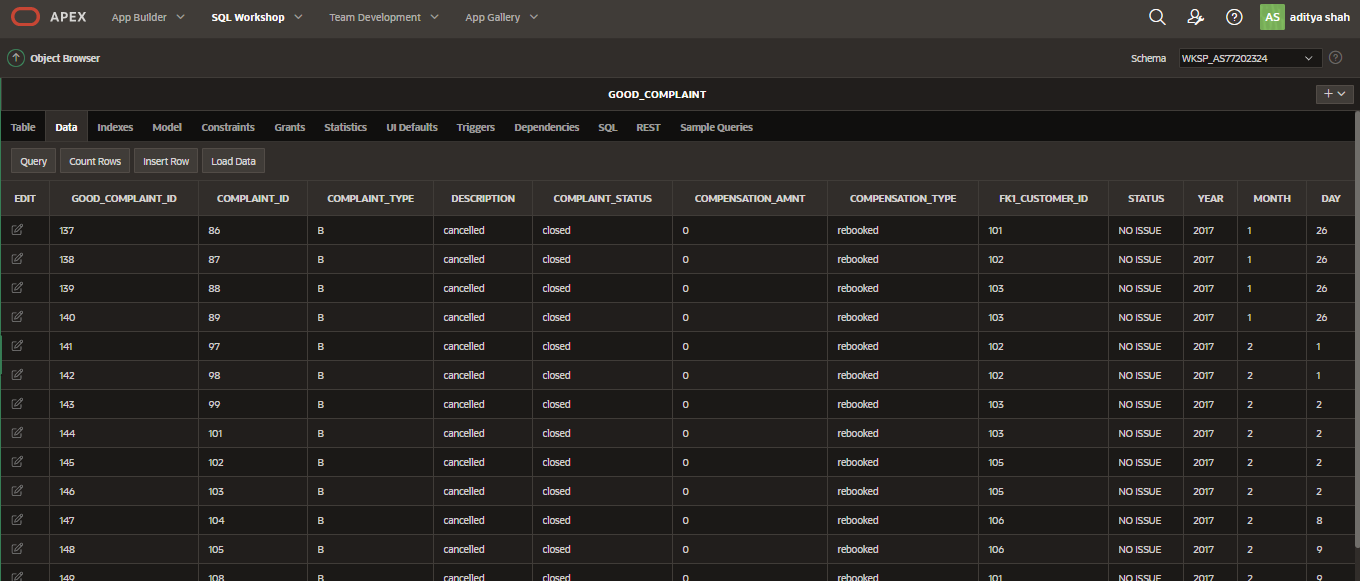
## PROCEDURE FOR GOOD DATAS





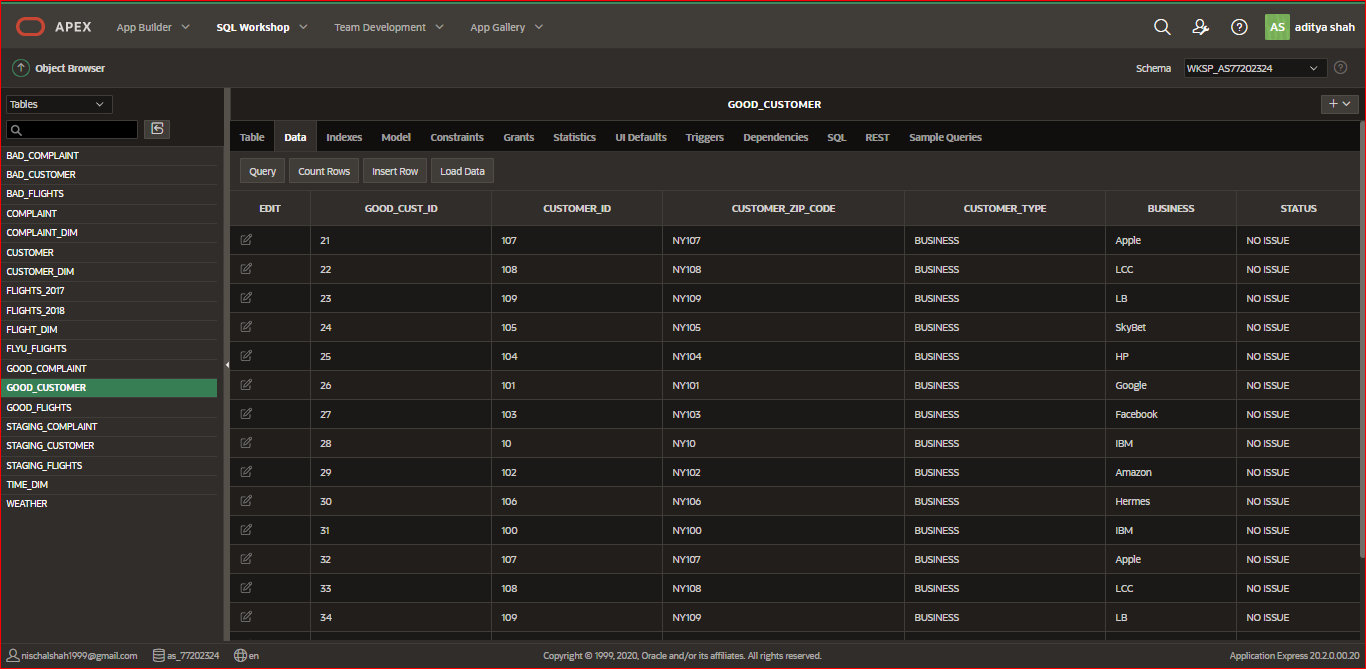
Procedure and sequence are created to extract the good data using loop for all good table from staging table.

## COMPLAINT\_GOOD



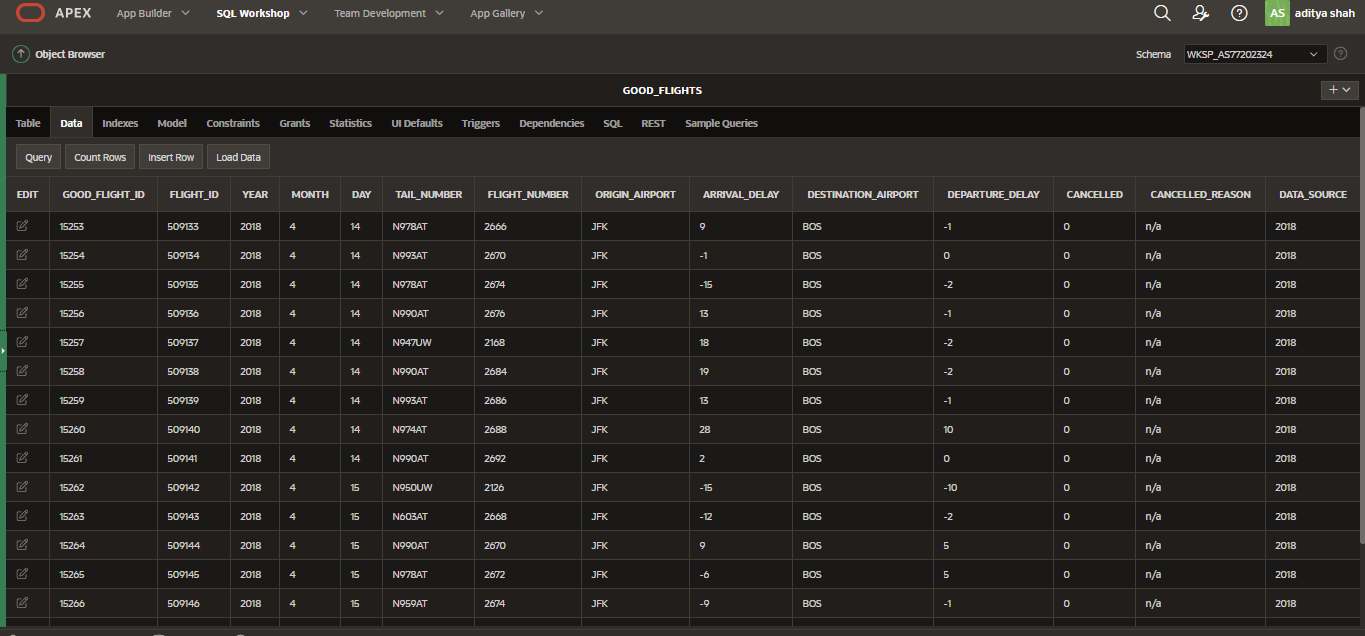
Good data are transferred to complaint\_good table.

## CUSTOMER\_GOOD



Good data are transformed to customer\_good table.

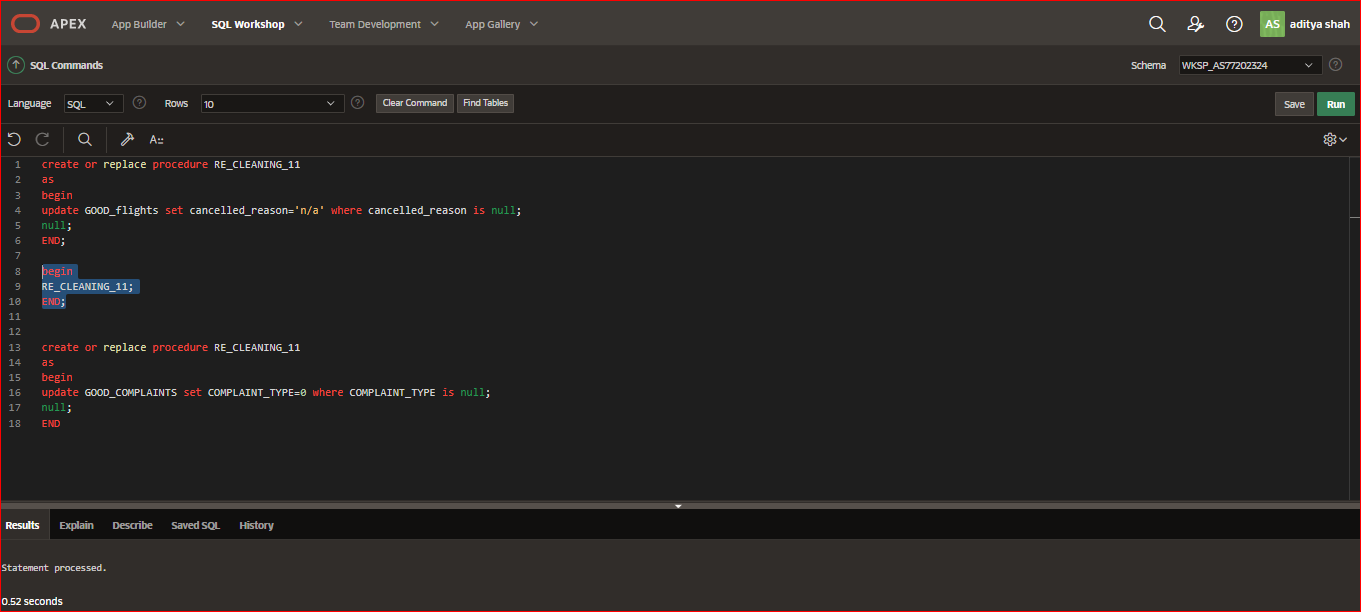
## FLIGHTS\_GOOD



Good data are transferred and store in flights\_good table.

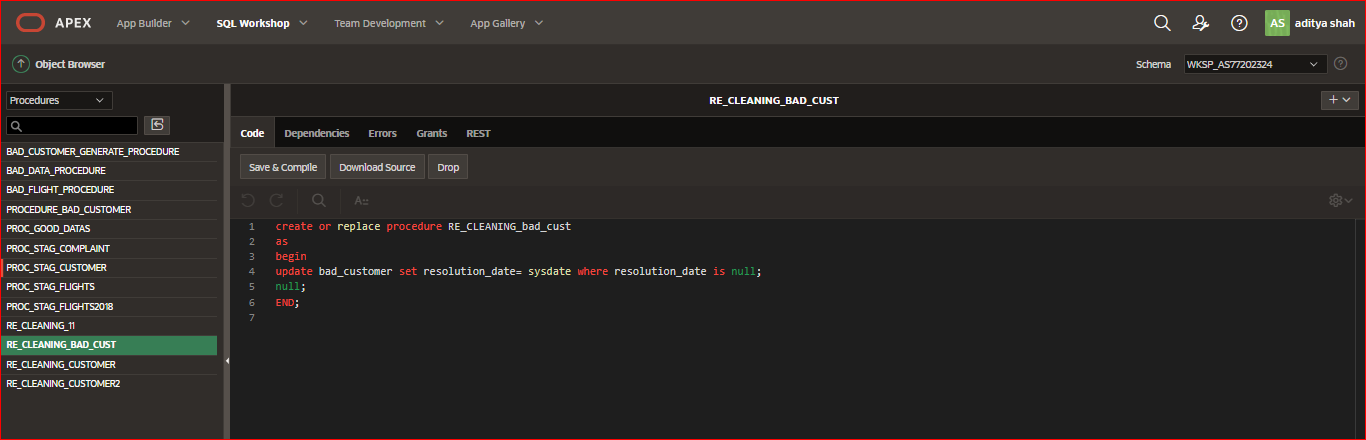
# RECLEANING DATAS:

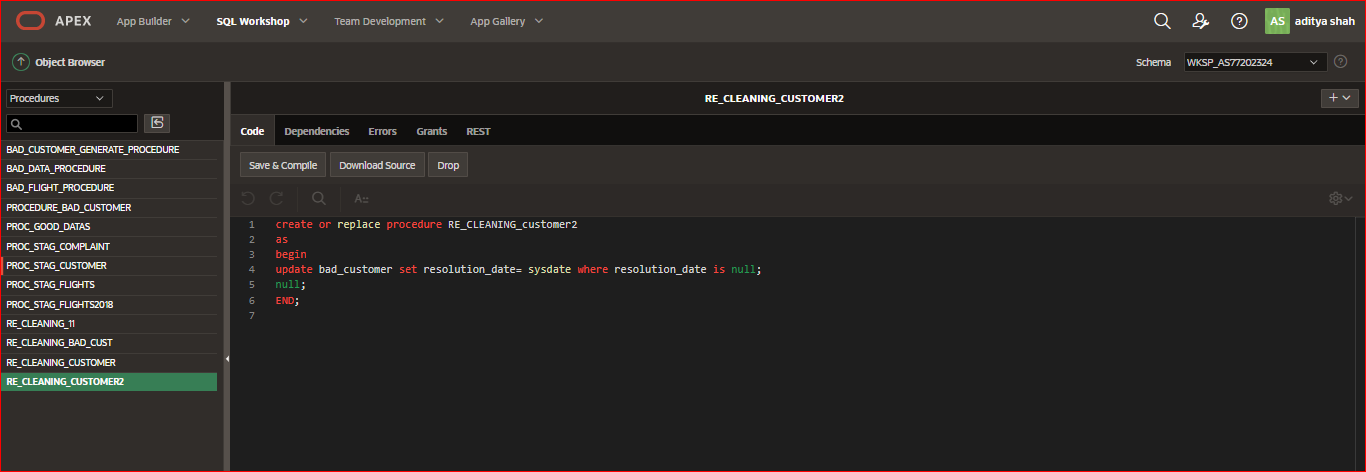
## CLEAN GOOD FLIGHTS



Recleaning the data in good flights table to remove null data by creating procedure to update the coloumn data.

## CLEAN BAD CUSTOMER

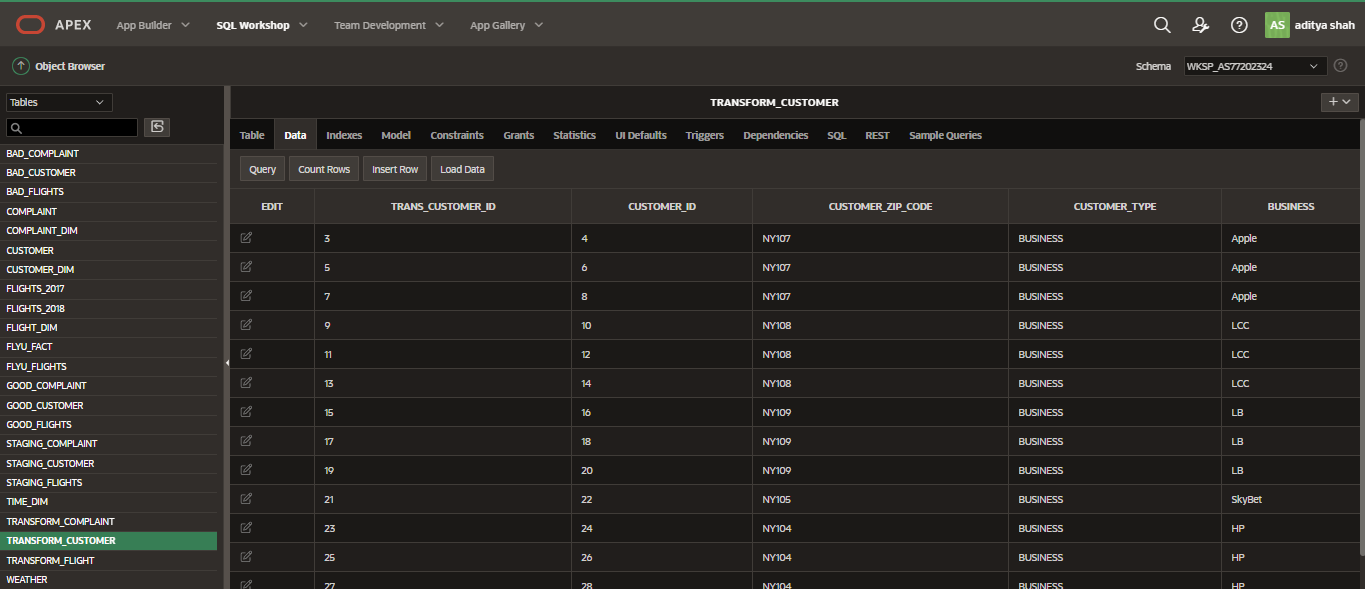


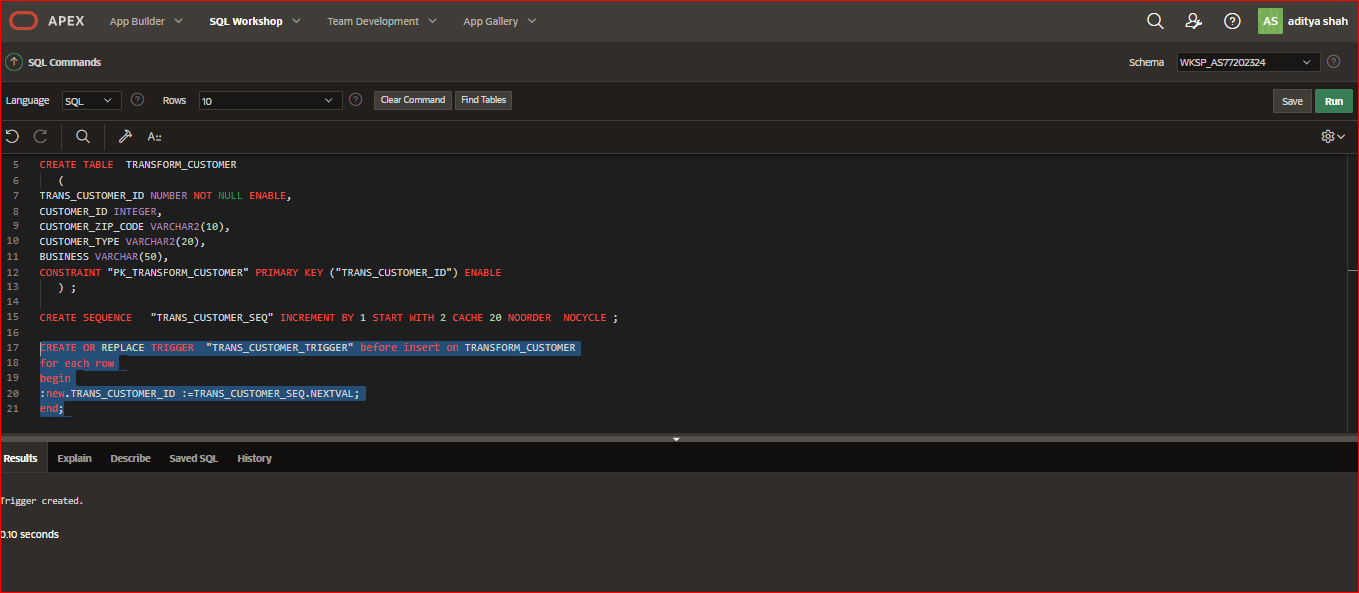


Recleaning the null value and adding error description and resolution date in bad customer table by updating table.

# TRANSFORM:

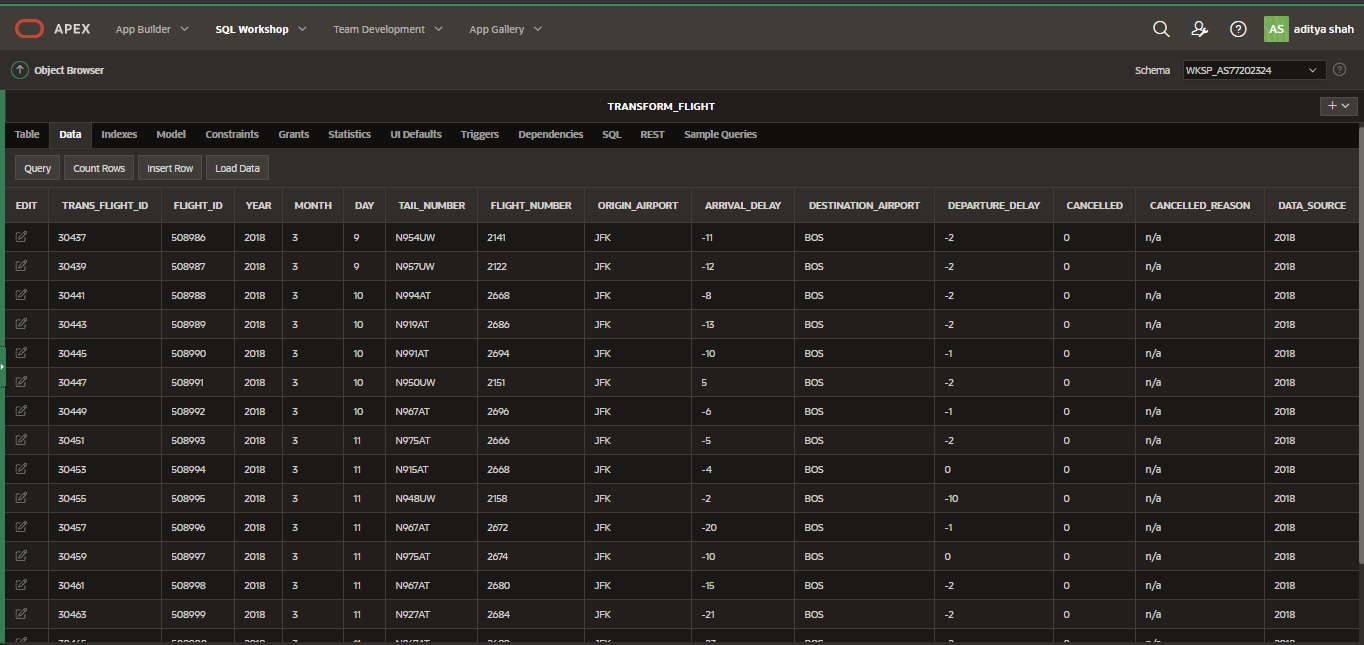
## TRANSFORM\_CUSTOMER

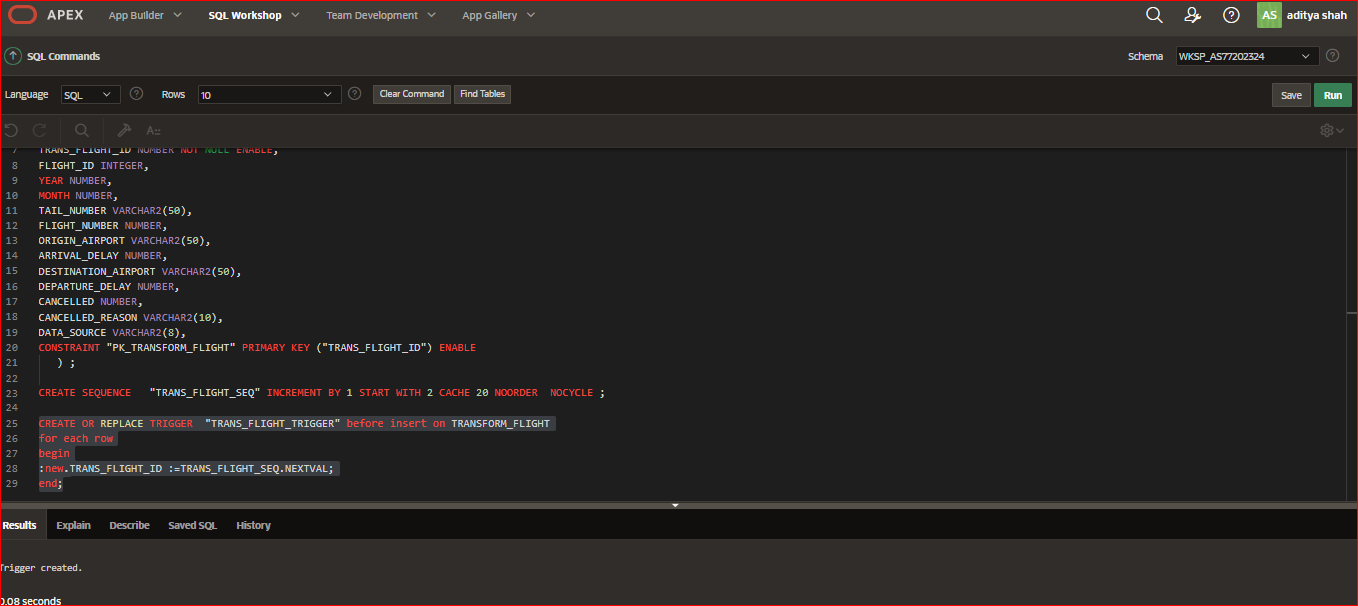




Good data are transform to the customer transform table from good customer table using merge statement.

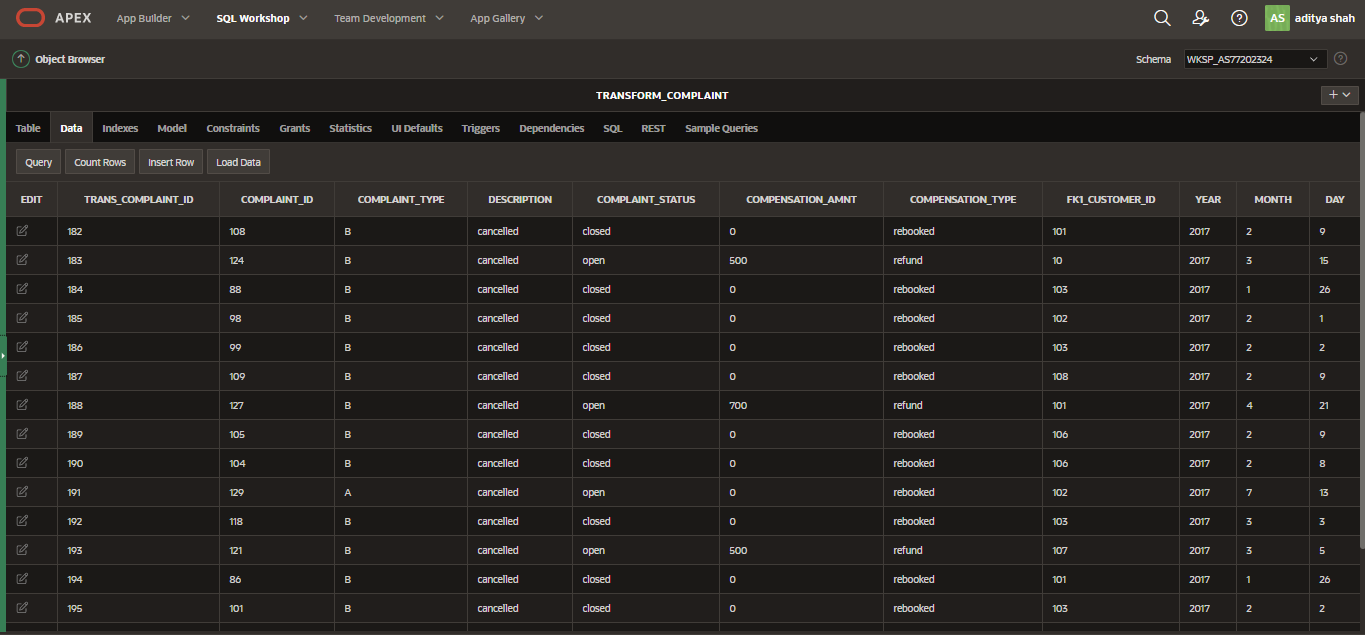
## TRANSFORM\_FLIGHT

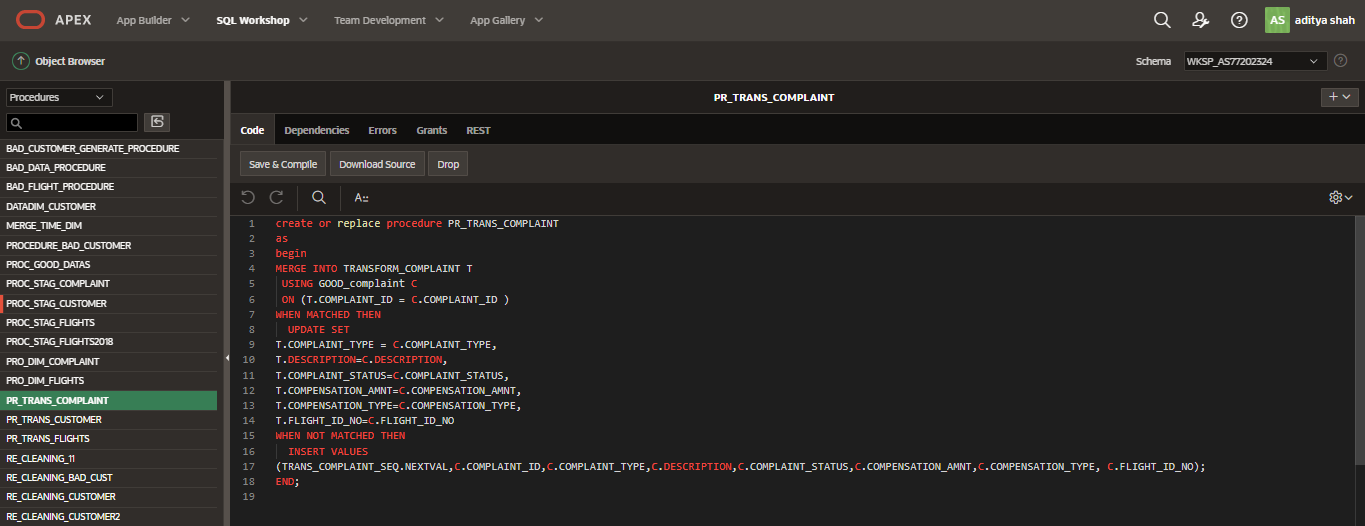




Good flights data are transformed to transformed flights table using procedure.

## TRANSFORM\_COMPLAINT

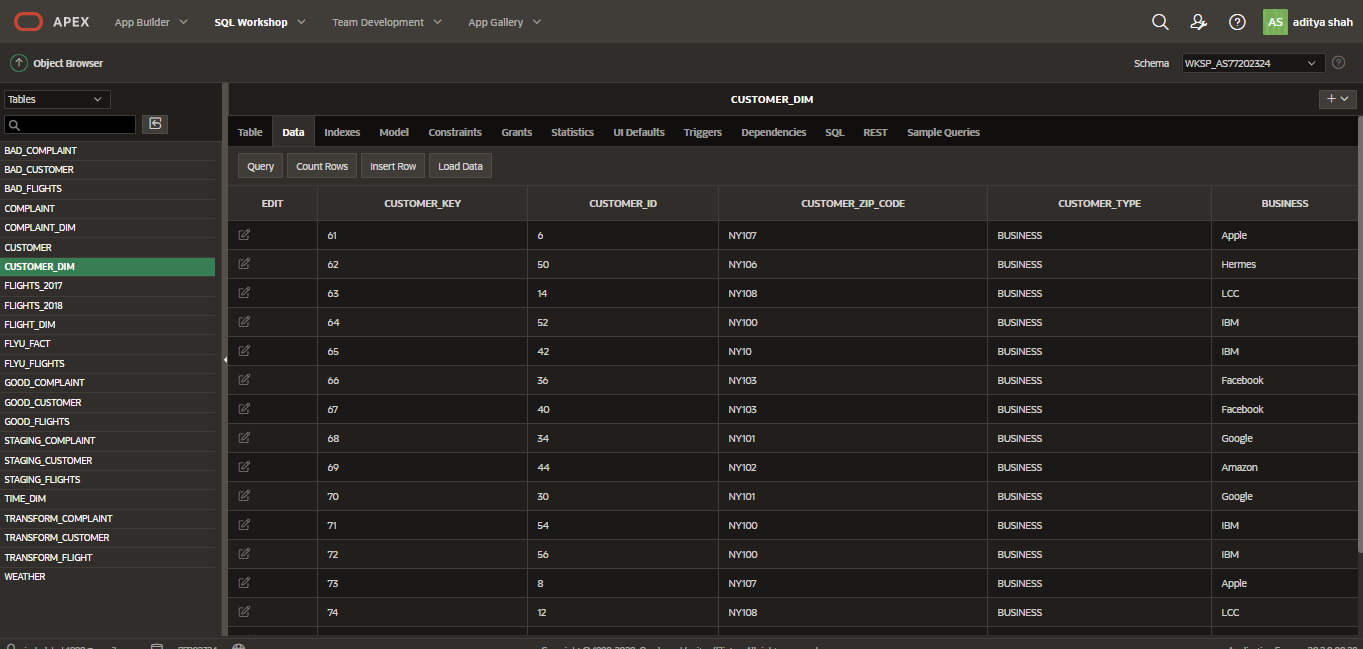


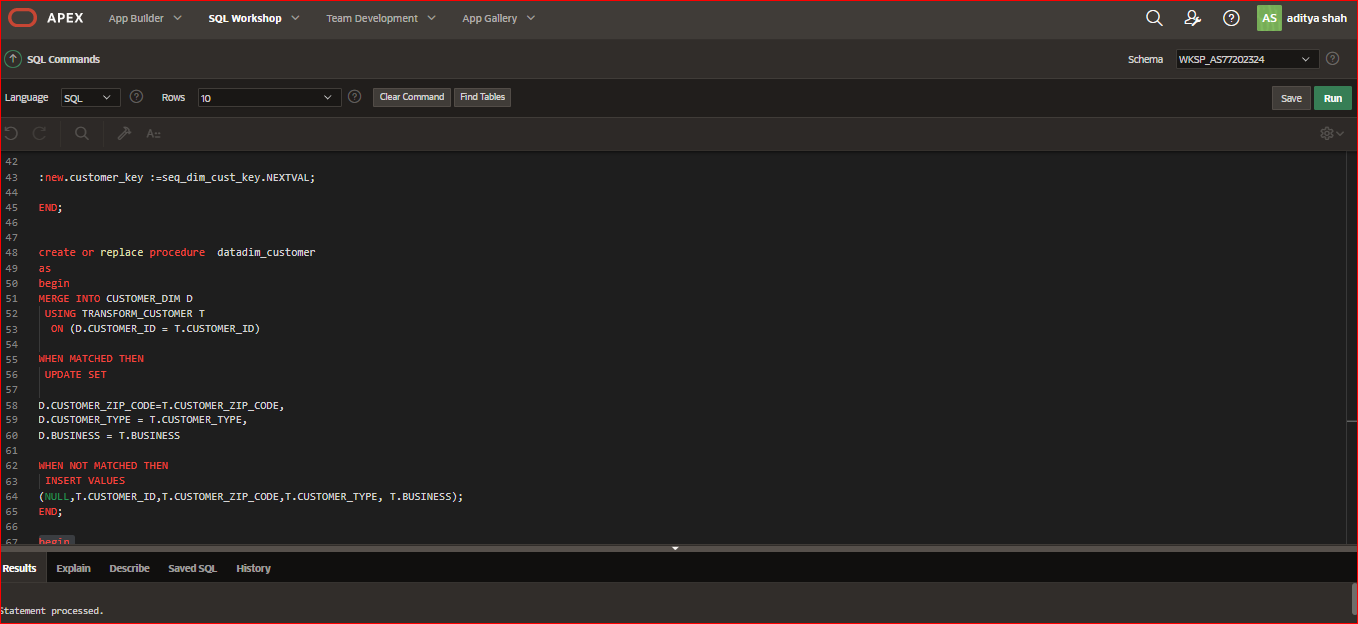


Transformation of good complaint data into transform complaint table by creating procedure.

# LOAD:

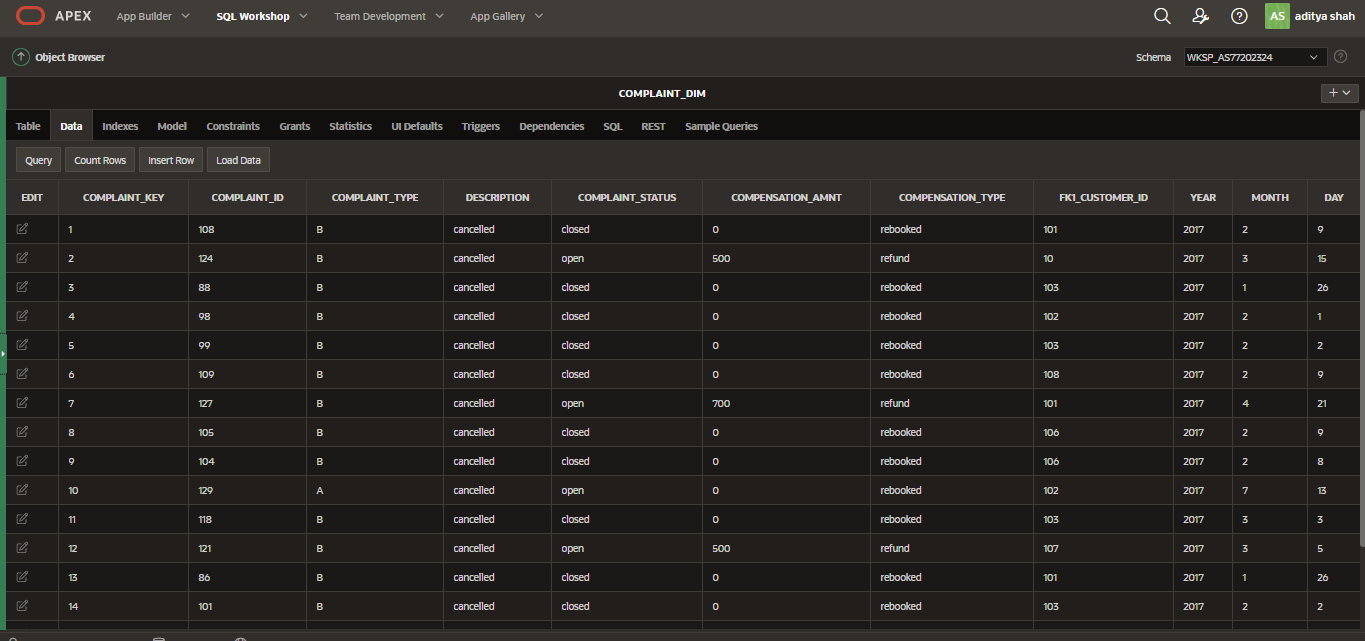
## CUSTOMER DIM

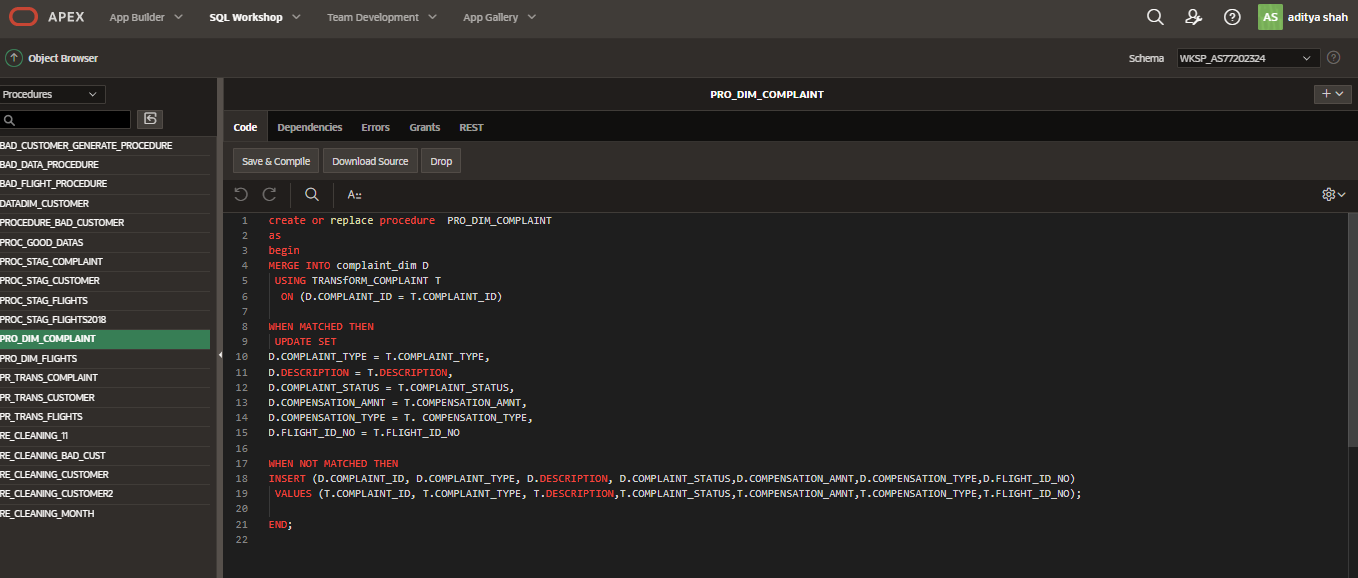




Loading good data from transform table by creating procedure and sequence.

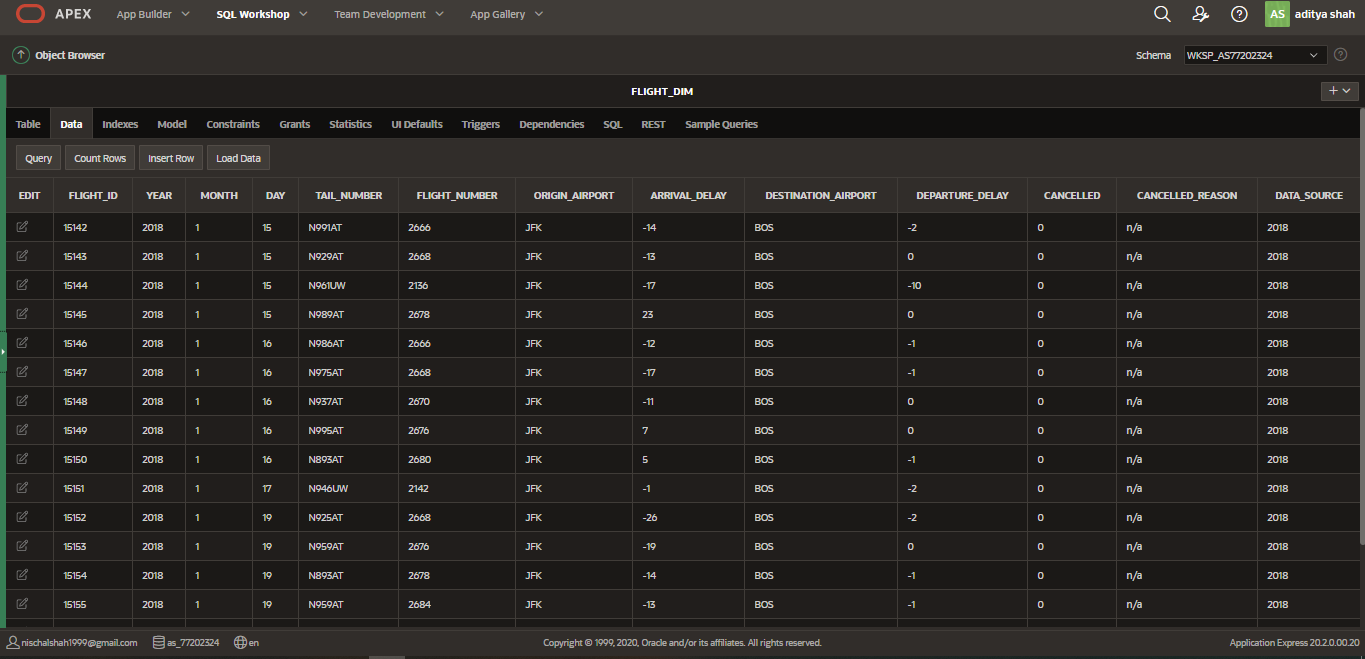
## COPLAINT DIM





Procedure in order to load clean data from good complaint table.

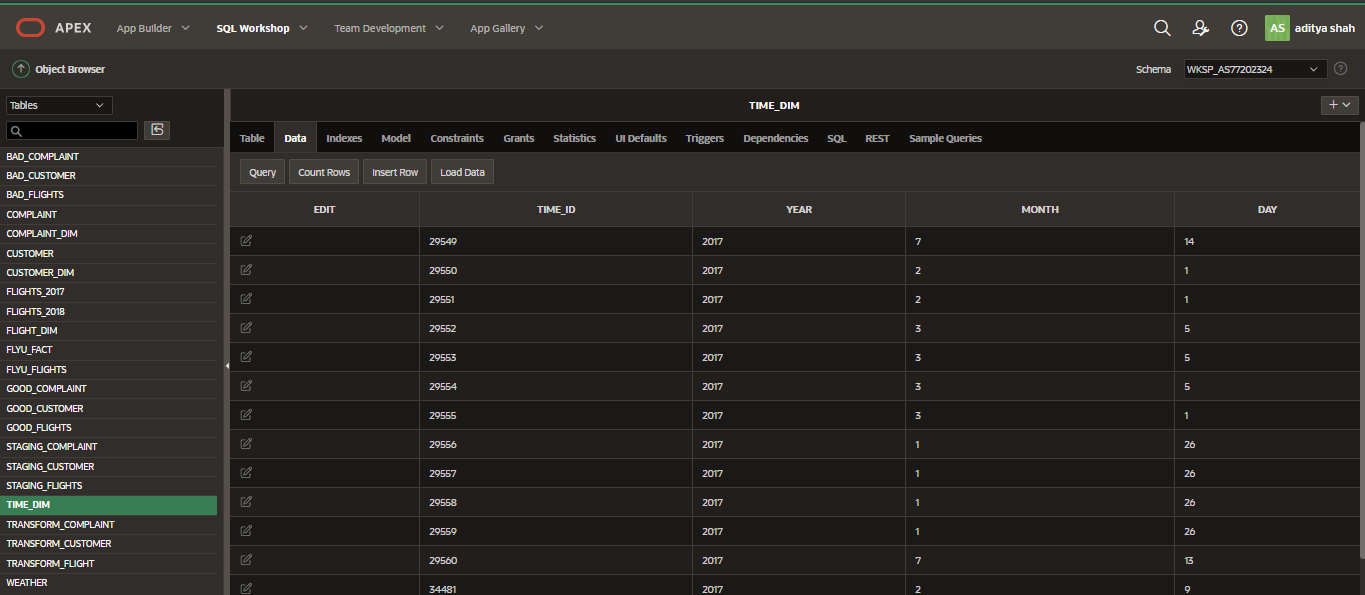
## FLIGHT DIM

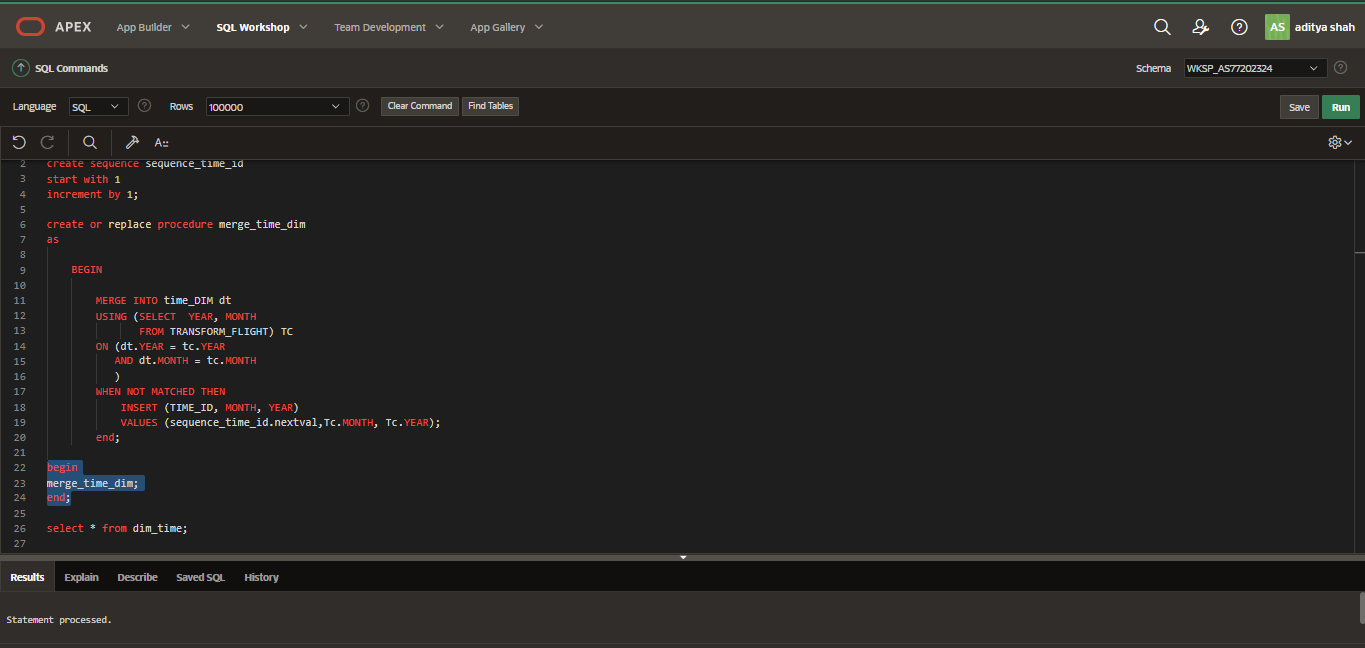




Clean data is loaded from good flights table to dim flight table using procedure.

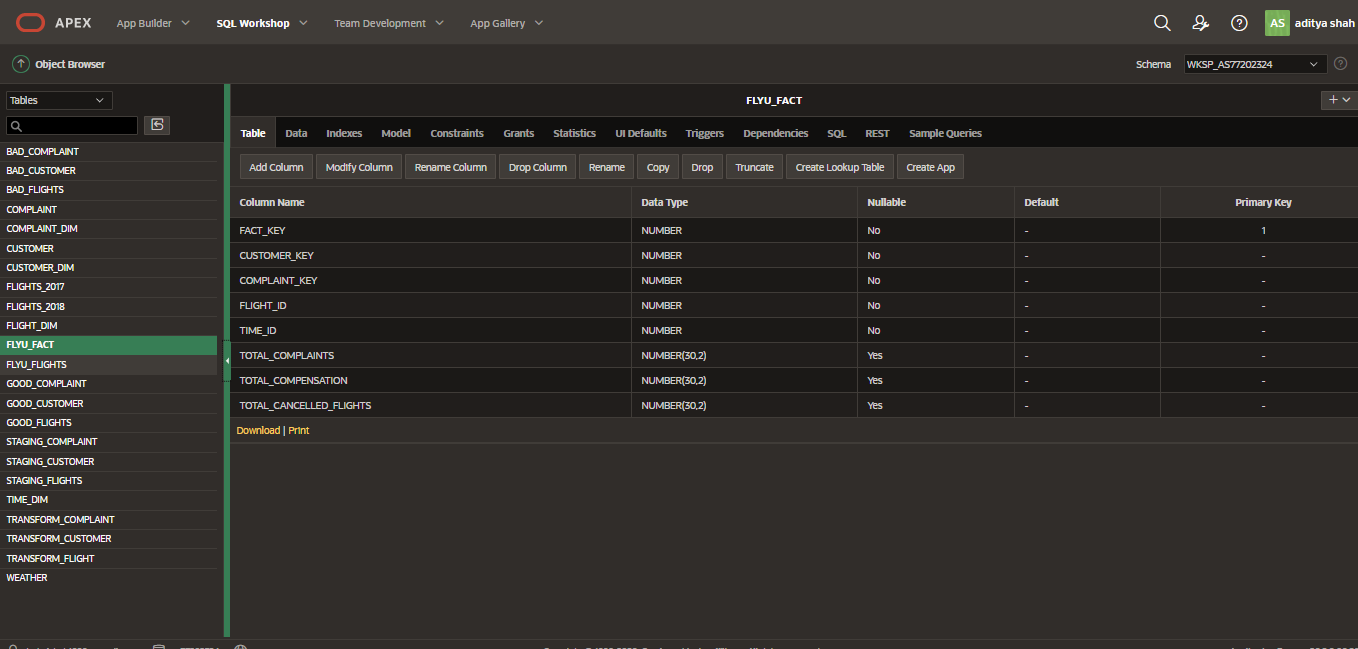
## TIME DIM

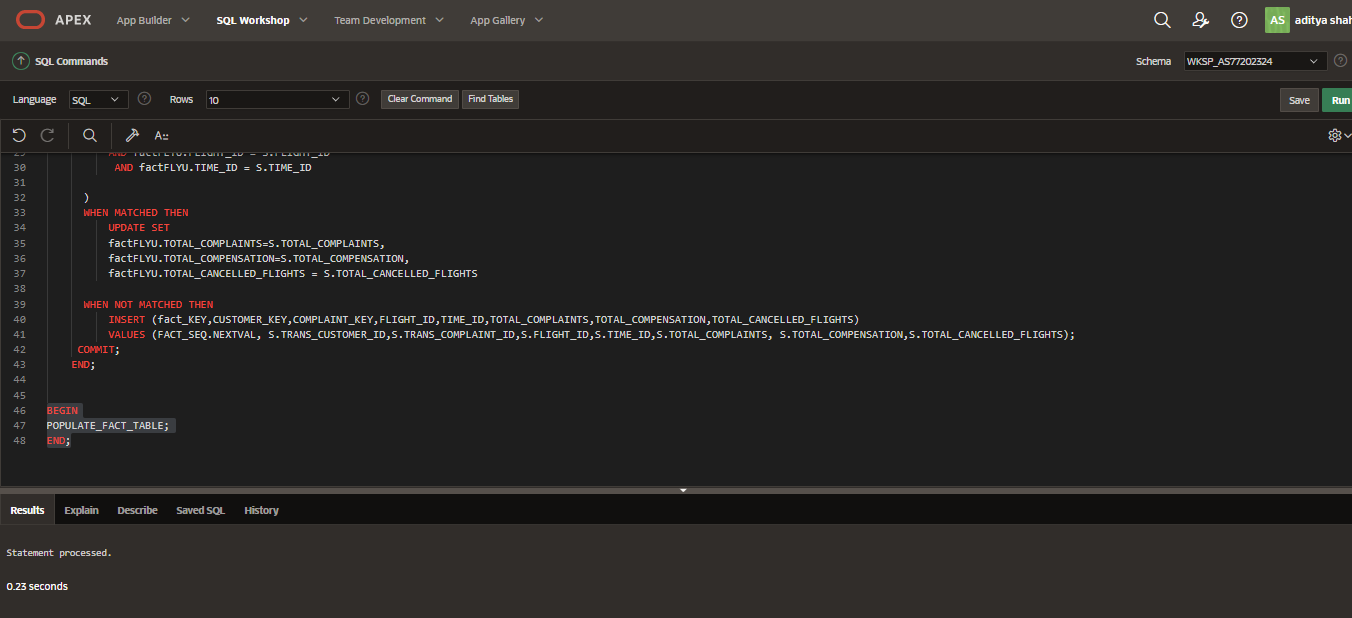




Clean time data loaded in time dimensional table from transform complaint table by generating procedure and sequence.

## FLYU FACT TABLE





Procedure to populate fact table with the appropriate data.

# GRANULARITY OF DATA:

Granularity is the lowest level of information or the level of detail considered during the storage of data. For example, the performance of an employee is considered as high level of granularity whereas daily performance of employee is considered to be low level of granularity.

## Time Dimension Table

|  |  |  |
| --- | --- | --- |
| **TIME\_DIM** | | |
| Attribute Name | Data Type | Key |
| Time\_id | INTEGER | PRIMARY |
| Year | INTEGER |  |
| Month | INTEGER |  |
| Day | INTEGER |  |

Table 1: Time Dimension Table

The time dimension table is granulated to store information on a monthly basis. For an airline corporation like FlyU retaining song of flights on a month-to-month groundwork is extra sensible. In order to analyze incoming flights on a month-to-month groundwork and to guide reports such as the range of delayed flights per month or number of cancelled flights per month, the time dimension desk is granulated to year, month, and day.

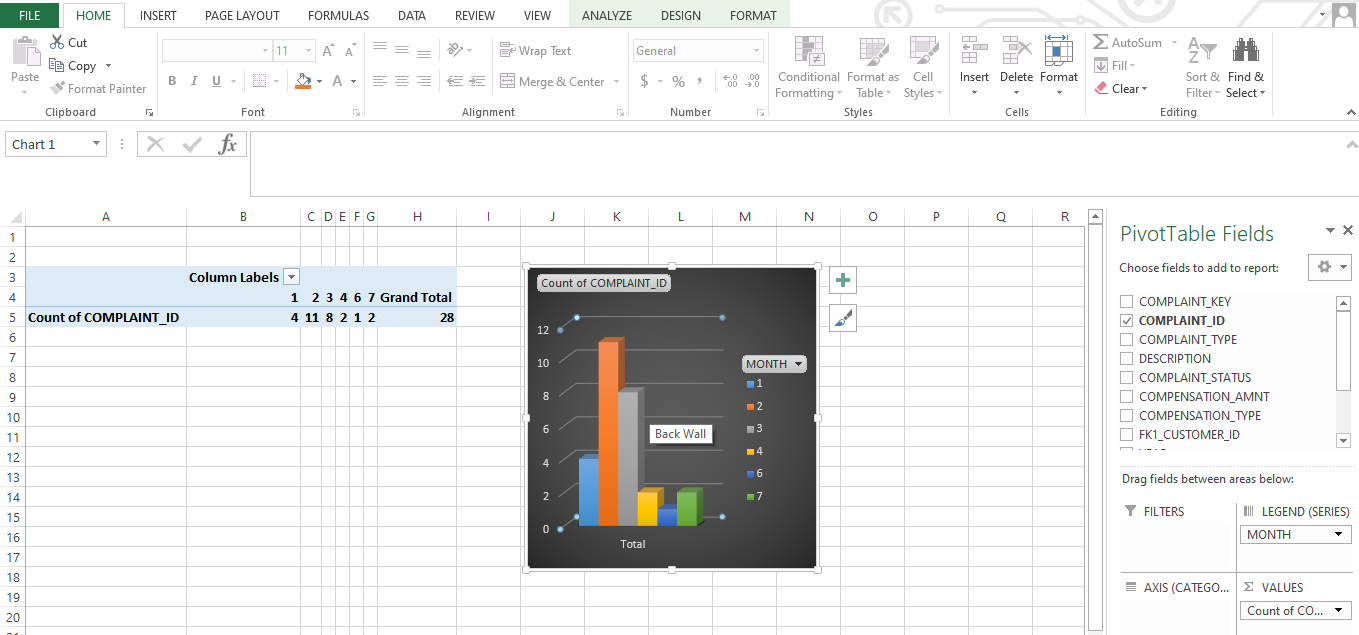
## FlyU Fact Table

|  |  |  |
| --- | --- | --- |
| **FLYU\_FACT** | | |
| Attribute Name | Data Type | Key |
| Fact\_key | INTEGER | PRIMARY |
| Customer\_key | INTEGER | FOREIGN KEY |
| Complaint\_key | INTEGER | FOREIGN KEY |
| flight\_id | INTEGER | FOREIGN KEY |
| Time\_id | INTEGER | FOREIGN KEY |
| total\_complaint | INTEGER |  |
| total\_compensation | INTEGER |  |
| total\_cancelled\_flights | INTEGER |  |

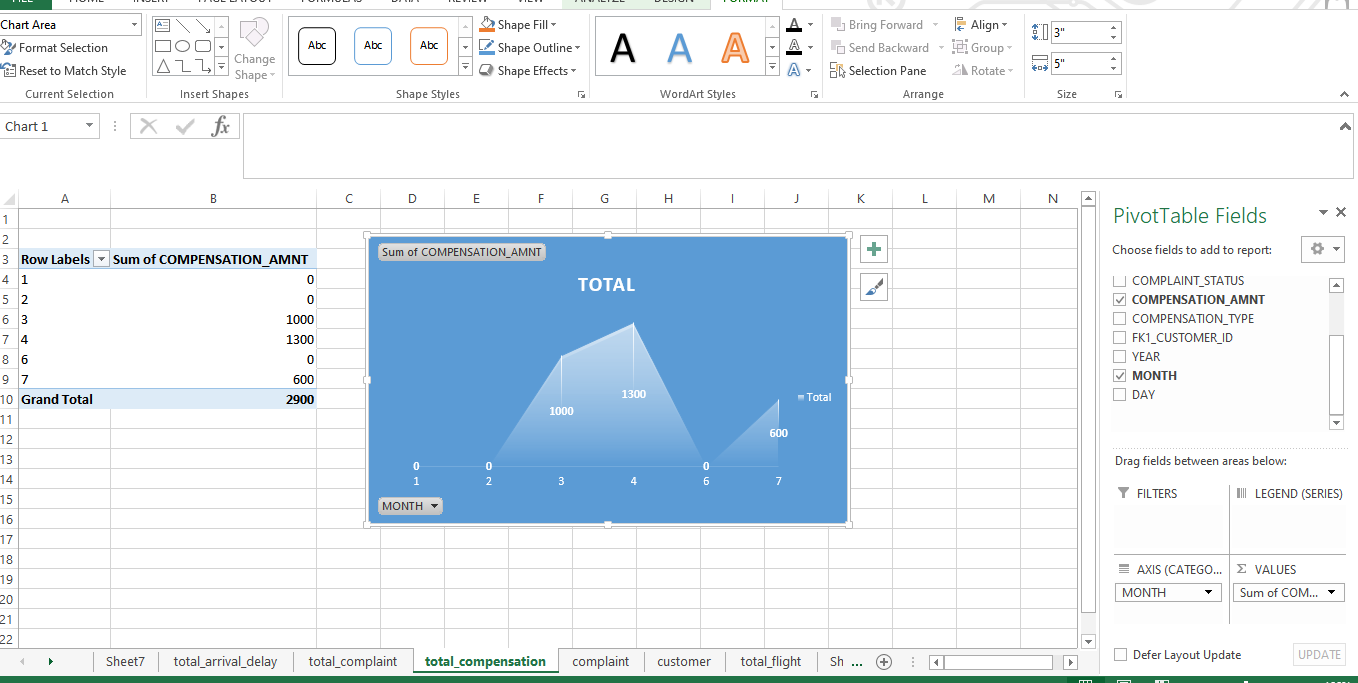
Table 2: Flight Outcome Fact Table

The fact table stores the records for each flight with the total complaint, total compensation, total flights cancelled.

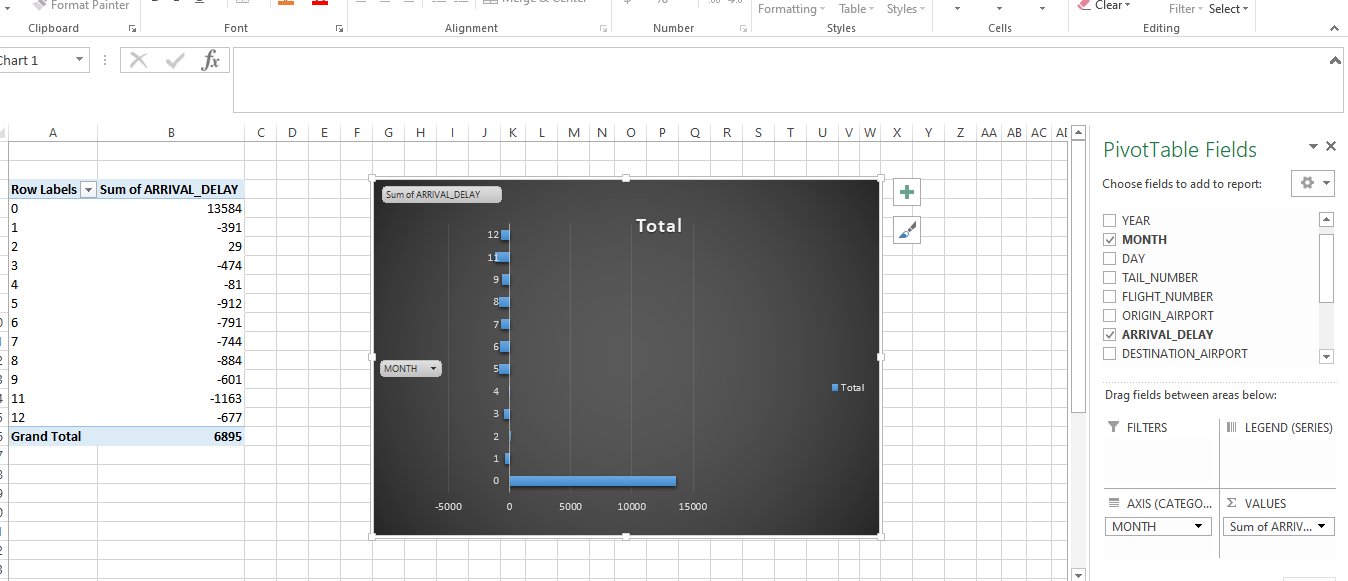
# TASK4: REPORTS GENERATED IN EXCEL



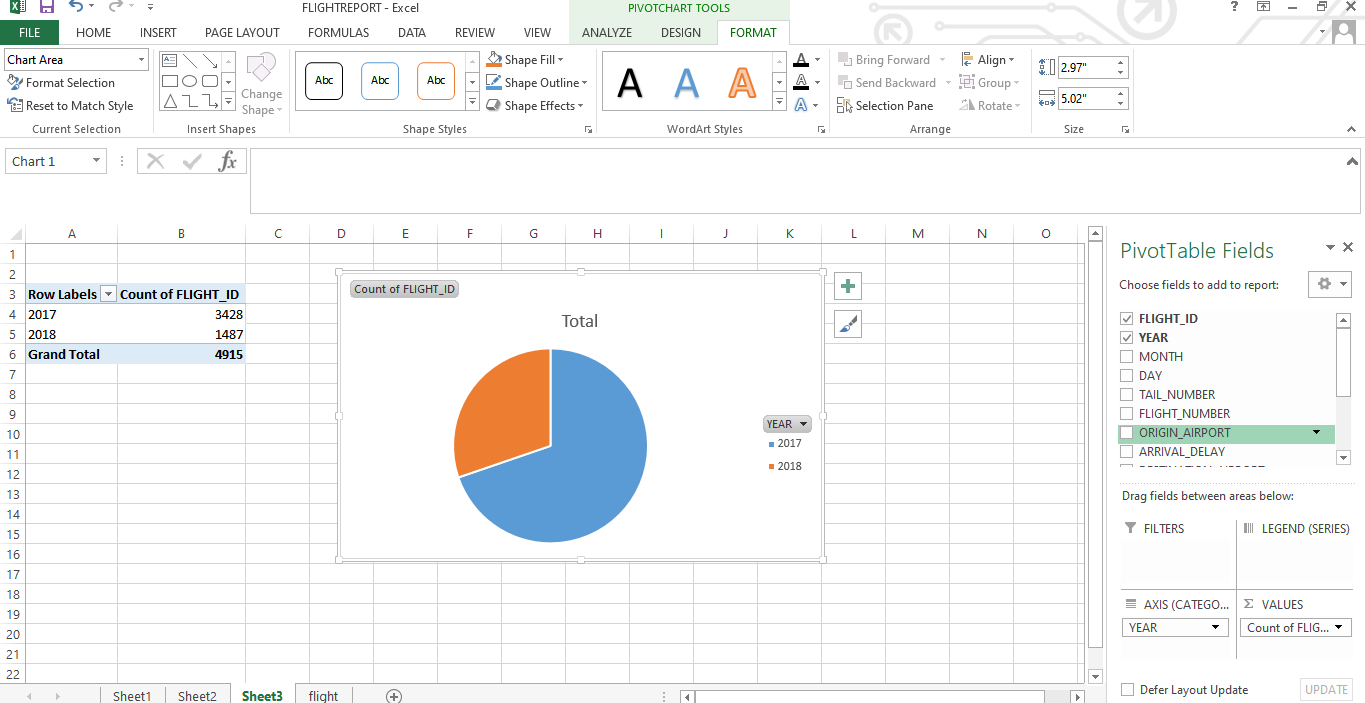
Total number of complant in a month.



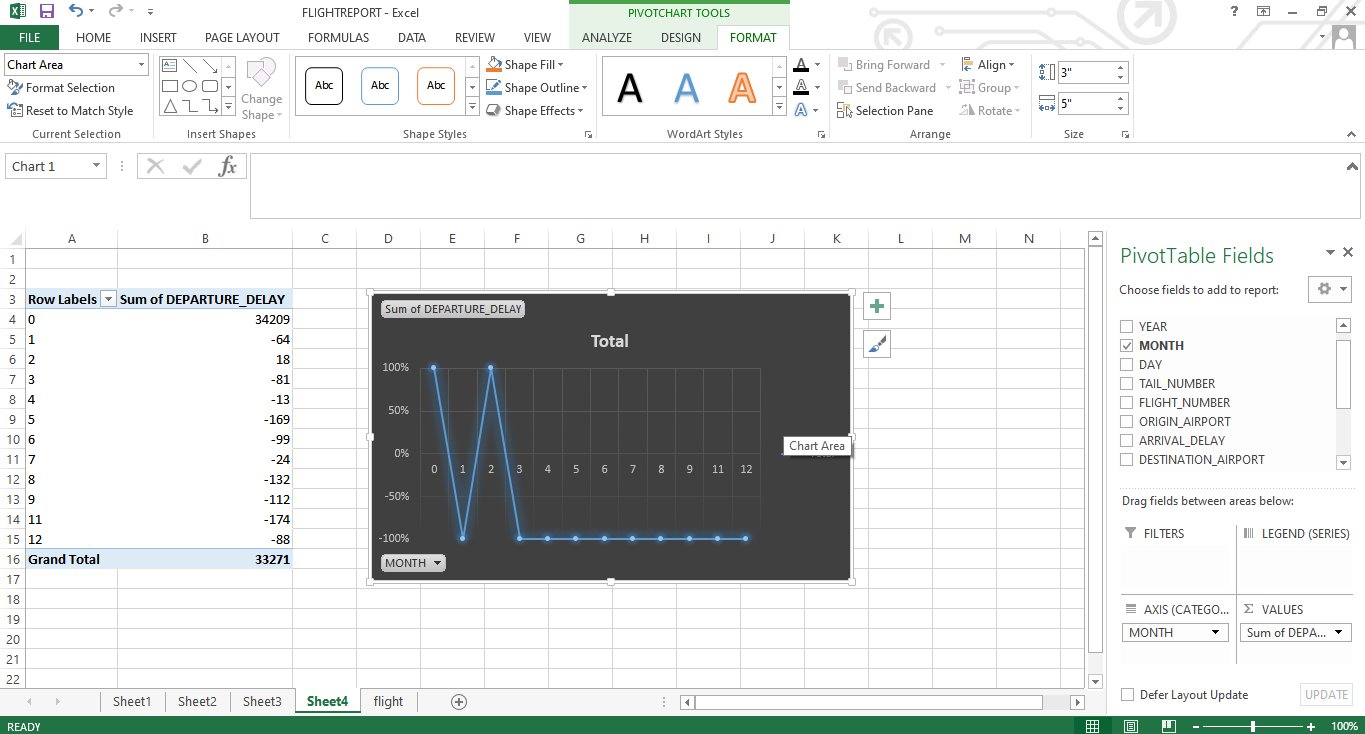
Total number of compensation amount per month.



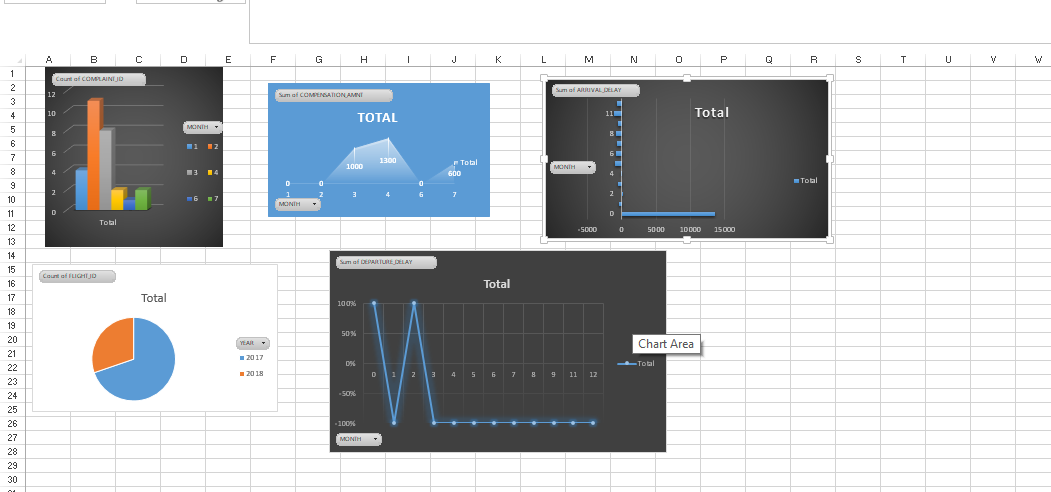
Total number of arrival delay with respect to month.



Total number of flight in a year.



Total number of departure per month.



Dashboard of all the reports generated.

# Task5: Data warehouse approaches

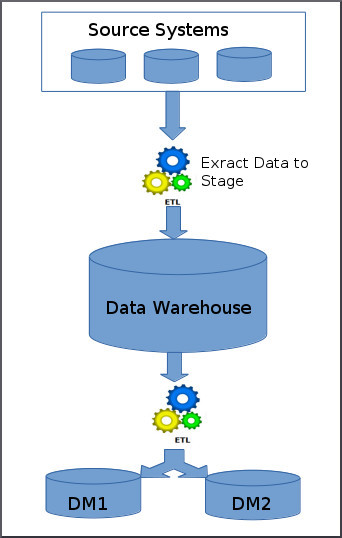
Data Warehouse design approaches are very important aspect of building data warehouse. Selection of right data warehouse design could save lot of time and project cost.

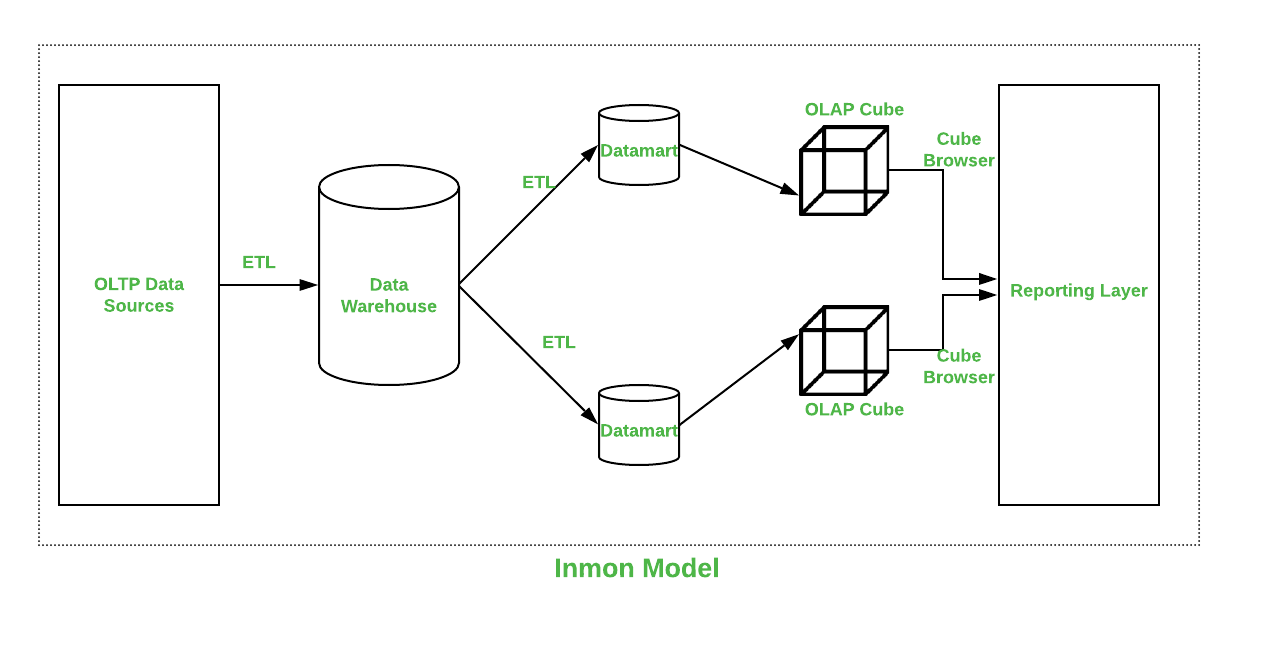
Data warehouse models provide advantages to an enterprise solely when the warehouse is viewed as the central hub of “all matters data” and not just a tool via which your operational reviews are produced. All operational systems ought to have two-way verbal exchange with the facts warehouse to feed statistics in and to acquire comments on how to improve operational efficiency. Any business change, such as an enlarge in expenses or reduction of supply/inventory have to first be prototyped and forecasted inside your records warehouse environment so that your enterprise can reliably predict and quantify the outcome

There are two different Data Warehouse Design Approaches normally followed when designing a Data Warehouse solution and based on the requirements of your project you can choose which one suits your particular scenario. These methodologies are a result of research from Bill Inmon and Ralph Kimball.

## Bill Inmon – Top-down Data Warehouse Design Approach:

“Bill Inmon” is sometimes also referred to as the “father of data warehousing”; his design methodology is based on a top-down approach. In the top-down approach, the data warehouse is designed first and then data mart are built on top of data warehouse.

The image depicts how the top-down approach works.



**Below are the steps that are involved in top-down approach:**

* Data is extracted from the various source systems. The extracts are loaded and validated in the stage area. Validation is required to make sure the extracted data is accurate and correct. You can use the ETL tools or approach to extract and push to the data warehouse.
* Data is extracted from the data warehouse in regular basis in stage area. At this step, you will apply various aggregation, summarization techniques on extracted data and loaded back to the data warehouse.
* Once the aggregation and summarization is completed, various data marts extract that data and apply the some more transformation to make the data structure as defined by the data marts.

**The key advantages of the Inmon approach are:**

* A truly corporate effort, an enterprise view of data
* Inherently architected- not a union of disparate Data marts
* Central rules and control
* May be developed fast using iterative approach

**Disadvantages of Inmon approach are:**

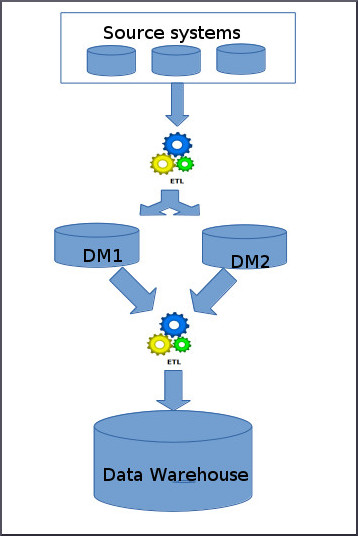
* Takes longer to build even with iterative method
* High exposure/risk to failure
* Needs high level of cross functional skills
* High outlay without proof of concept
* Difficult to sell this approach to senior management and sponsors

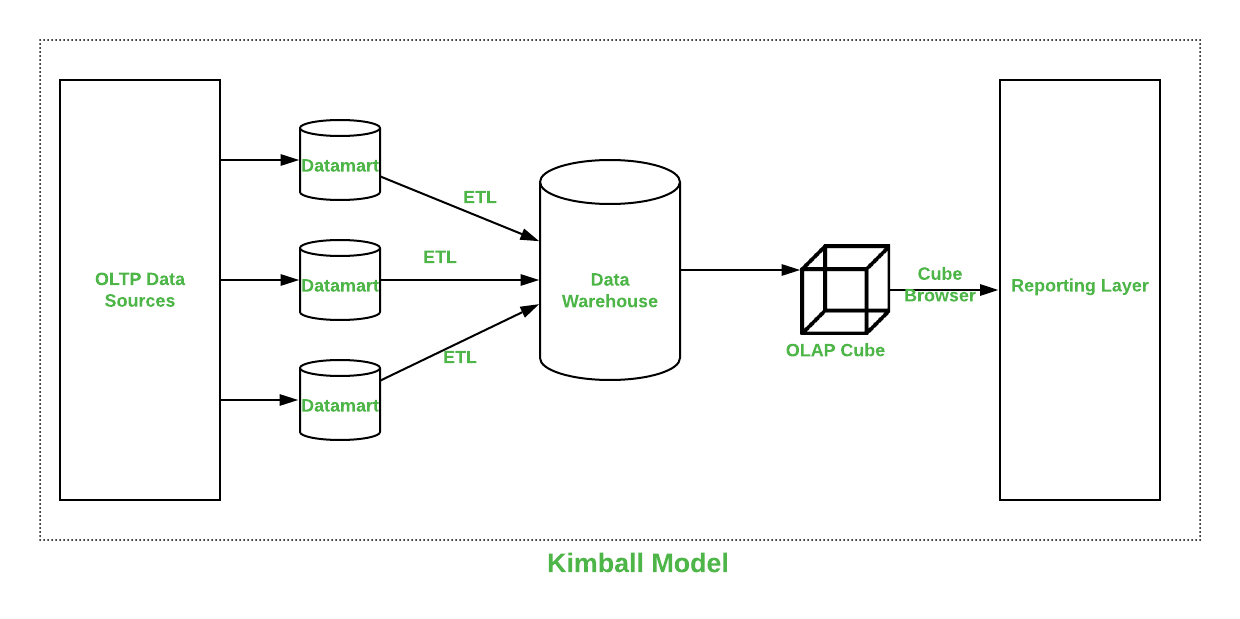
## Ralph Kimball – Bottom-up Data Warehouse Design Approach:

Ralph Kimball is a renowned author on the subject of data warehousing. His data warehouse design approach is called dimensional modelling or the Kimball methodology. This methodology follows the bottom-up approach.

As per this method, data marts are first created to provide the reporting and analytics capability for specific business process, later with these data marts enterprise data warehouse is created.

The image depicts how the bottom-up approach works.





Basically, Kimball model reverses the Inmon model i.e. Data marts are directly loaded with the data from the source systems and then ETL process is used to load in to Data Warehouse. The above image depicts how the top-down approach works.

**Below are the steps that are involved in bottom-up approach:**

* The data flow in the bottom up approach starts from extraction of data from various source system into the stage area where it is processed and loaded into the data marts that are handling specific business process.
* After data marts are refreshed the current data is once again extracted in stage area and transformations are applied to create data into the data mart structure. The data is the extracted from Data Mart to the staging area is aggregated, summarized and so on loaded into EDW and then made available for the end user for analysis and enables critical business decisions.

**Here are some of the advantages of the Kimball method:**

* Faster and easier implementation of manageable pieces
* Favorable ROI and proof of concept
* Less risk of failure
* Inherently incremental; can schedule important Data Marts first
* Allows project team to learn and grow

**Disadvantages of Kimball method:**

* Each DataMart has its own narrow view of data
* Permeates redundant data in every DataMart
* Difficult to integrate if the overall requirements are not considered in the beginning

## Bill Inmon Model vs Ralph Kimball Model:

| **PARAMETERS** | **KIMBALL** | **INMON** |
| --- | --- | --- |
| Introduced by | Introduced by Ralph Kimball. | Introduced by Bill 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. |
| Maintenance | Here maintenance is difficult. | Here maintenance is easy. |
| Data Model | It prefers data to be in De-normalized model. | It prefers data to be in normalized model. |

(manmeetjuneja5, 2020)

# Bibliography

Guru99, n.d. *guru99.* [Online]   
Available at: https://www.guru99.com/oltp-vs-olap.html  
[Accessed 30 November 2020].

manmeetjuneja5, 2020. *Geeksforfgeeks.* [Online]   
[Accessed 31 November 2020].

Oracle Corporation, n.d. *Oracle.* [Online]   
Available at: https://www.oracle.com/database/what-is-a-data-warehouse/  
[Accessed 30 November 2020].

Talend SA, n.d. *Stitch.* [Online]   
Available at: https://www.stitchdata.com/resources/oltp-vs-olap/  
[Accessed 30 November 2020].

Techopedia Inc, 2016. *Techopedia.* [Online]   
Available at: https://www.techopedia.com/definition/1184/data-warehouse-dw  
[Accessed 31 November 2020].