**Introduction**

No company can survive without a large and accurate store of historical data, ranging from sales inventory to intellectual and personal property record. In the modern days effective and efficient data mining plays a vital role in business, and when it comes to data mining data warehouse is the core part of it. Data warehouse has the ability to look back at early trends and which have the accurate and properly formatted data and trend, Data warehouse play significant role for any company to make a better business decision. Let’s say if company suddenly needs to know the sales last 24 then rich historical data provided by the data warehouse can make this possible, not only this data warehouse can also add context to this histogram by listing all the key performance and trends that surround retrospective research. This kind of effective and easy overview can no matched by a legacy database. By enabling the historical overview data warehouse permit company to learn form the past trends and challenges. In essence, the benefit of data warehouse is continuous improvement. Data warehouse can also scale with business any growing company needs more and better data, and for this data warehouse can grow in more robust because of the ability of handling queries. The data are store in cloud or other third party by ensuring data security by using encryption and using specific protection set up.

The main benefit so having data warehouse in company is that it offers a weighty increase in competitive approach by enabling metric based, smarter decision on everything ranging from product release to the inventory level to main sales levels. It almost seems impossible for any business to compete in today’s market without the advanced of data warehouse.

# Data warehouse

Data warehouse is the data management system, which is specially designed to support and enable the business intelligence (BI) activities, especially analytics. It the process of collecting managing data from different sources to give meaningful business insights. It contains a large amount of data which are derived from a wide range of sources like transaction applications, application log etc. By using this huge amount of data, data analyst can analyze it, which allow organization to make right and valuable decision which improve their decision making. Data warehouse give result base on the data so it is considering as the organization ‘single source of truth’ (Anon., n.d.).

Designing the data warehouse is essential part of the business development, it makes organization to make better business decision. Once the data ware house is implemented on your organization it can benefit company in numerous ways. Given below are the some of the notable benefits of data warehouse.

## **Advantages of data warehouse**

* Improve the decision-making process
* Enable organizations to forecast with confidence
* It can enhance the business intelligence
* Enhanced data quality and consistency
* Generate high return on investment
* Business intelligence from multiple sources
* Increased Query and system performance.
* Data warehouse help to get holistic view of current standing and evaluate opportunities and risk.
* Data warehouse simplifies the flow of information through network by connecting all the related and no-related parties.
* Data warehouse keep all the data in one place and save user’s time to access the particular set of data, which can make rapid decision on enterprises actions.

## **Disadvantages of data warehouse**

* Bigger the organization more the data, and the extra time to load data warehouse run. Data which are generated by data warehouse require the participant of each department in the organization which bother with extra report work.
* Sometimes the standardization and similarity in the data formats lead to inflexibility and homogenization of data. Which further limit the data in the terms of establishing relation during aggregation and difficult to tune for query speed, and homogenization also cause loss of data.
* Sometimes centralizing data at once place cause lead to different as they hesitate to share their personal data within central repository. Which also raises ownership and security concern for few departments, so organization must sure that analysis of data is given to trusted individuals within the enterprise.
* Sometimes internal source of fueling data warehouse keep pushing problems which are undetected for years.

## **Characteristics of data warehouse**

**Integration**

Integration refers to the shared of entity to scale the all similar data from various database. Data warehouse is built by just integrating data from multiple sources of data. Moreover, it must have the reliable formats and codes, naming conventions. This integration of data warehouse benefits in the analyzation of data. Reliability of column scaling, encoding structure should be confirmed. This integration of data warehouse handles numerous subject related warehouses.

**Subject oriented**

Data warehouse is process and it is proposed to handle with specific theme, and this theme can be sales, marketing, distribution etc. Data warehouse focus to demonstrate and analyze of data to make the better decision. It also distributes easy a precise demonstration around the special theme just by eliminating data which is not required to make the decisions.

**Time variant**

Data is maintained in different time period such as weekly, monthly or annually. Data which are resided in the data warehouse is predictable with the particular time interval and delivers information from historic perspective. Another time variance is that once the data is store in data warehouse can not be modified, alter or updated.

**Non-volatile**

Data which are stored in data warehouse is permanent. Which means data can not be erased or deleted when new data is inserted.

# Data warehouse development methodologies

To design the data warehouse there are two methodologies they are Kimball and Inmon theory. Bill Inmon and Ralph Kimball sated different philosophies, for information collecting, information management, and also for the analytics for the decision support. Both theory approach problem from different viewpoints, techniques of designs and implementation of strategies. Both methodologies have their own advantages and disadvantages, based on our requirements we can use either Kimball or Inmon to develop the data warehouse, the more important is which one serves user at low redundancy. At first let’s discuss about the bill Inmon’s and Kimball theory then we will select the best one for our scenario.

## **Bill Inman’s theory**

This theory was introduced by the Bill Inmon. Inmon first start with centralized enterprise-wide data warehouse just by the multiple databases to the analytical needs of departments, which are later known by the data marts. Because of this reason Bill Inmon is consider as the top-down approach. The main core of Bill Inmon theory is enterprise data warehouse. The main central repository of the data combined from all the operations system of organization. This theory is the definitive of all the representation of business data, which means all the organizations choose the naming definitions of which data true and their conflicting values and all the other data cleaning operations are done before enters the warehouse. In this model data is stored in normalization form and also warehouse is not directly created, but data is fed into various data marts which are then filtered down to the subsets of particular needs. For example, sales department will have the data only which are used by the sales team. Applications which re retrieve the data will be joined to data marts. The main advantages of having these model is that it is in normalization form so it makes less data redundancy (Vidhya, n.d.).

### Advantages of Bill Inman’s theory

* In Inman’s theory it serves as the single source of truth for the enterprises, data marts and for all the data in Datawarehouse is integrated.
* Data update anomalies are avoided because of low redundancy, which make ETL process easier and less to failure.
* In this theory business procedures can be understood easily.
* When the business requirements change, it is easy to update data warehouse.
* It can handle wide range of reporting needs across the enterprise.

### Disadvantages of Bill Inman’s theory

* Implementation of the model can become complex over the time as it involves more table and joins query.
* As the resource required experts of data modeling which is hard to find and often expensive.
* More ETL is require as the data marts are built from the data warehouse.
* Initial set-up and delivery will take more time
* Specialist team are required to successfully manage the environment.

## **Ralph Kimball**

This model is introduced by the Ralph Kimball. This model starts with identifying the business process and questions that data warehouse to answer. Then these sets of information are being analyzed and then documented well. Data is collected using ETL process from multiple data sources called data marts and then it is loaded into common area called staging, then it is transformed into OLAP cube. This approach is also known by the bottom-up approach. This idea is when the data is being in normalized into star schema. This approach makes query writing fast and simple, and also get report very quickly. At first data from particular area are process into a star schema, and that Kimball data marts are connected with shared attributes and that form a dimensional data warehouse. The main advantages of having Kimball model is that we can get report very quickly in comparison to the Inmon and writing query is also very simple and fast (Vidhya, n.d.).

### Advantages of Ralph Kimball’s theory

* The first set up phase of data warehouse is easy and will be delivered quickly.
* Star schema of this theory can be understood easily by business users and easy to use for the reporting.
* It occupies less space in the database which make management system fairly easier.
* Even small team of developers is enough to keep the data warehouse performing effectively.
* It really works well for the department-wise metrics and KPI tracking, as the marts are geared towards department-wise reporting.
* Database engine will perform ‘star join; where cartesian product will create and the fact table will be queried for the selective rows.

### Disadvantages of Ralph Kimball’s theory

* Redundancy of data can cause data anomalies over the time.
* Adding column in fact table can cause performance issues, this is because fact tables are created in very deep, so if new columns are added the size of the table becomes much larger and will no perform well. Which make dimensional model hard because business requirements change.
* Integration of data in data warehouse can be complex procedures.
* It cannot handle all the enterprises because reporting needs because model is oriented towards business rather than enterprises as whole.

## **Key difference between Bill Inman’s and Ralph Kimball**

|  |  |
| --- | --- |
| **Ralph Kimball** | **Bill Inmon** |
| Relatively it takes less time to implement | It is quite complex and take more time to implement |
| This base theory is difficult to maintain | It is easy to maintain in comparison to Inman theory |
| It cost low in the initial phase because we only need data warehouse and cost remain same for the remaining phase. | Implementing this approach can cost high at initial phase, but the subsequent project development cost low. |
| Kimball based data architecture can be set up quickly in comparison to Bill Inman approach. | This theory base data warehouse set up require bit more time. |
| Kimball approach data warehouse require generalist team to implement. | To apply this based for the data warehouse we need a specialist team. |
| Kimball base data warehouse integration focused on the individual business area. | Inman based data warehouse integration nee enterprise wide data. |
| Kimball fallow bottom approaches | Inman fallow top-bottom approach |

# OLTP and OLAP

## **OLTP**

OLTP stands for online transaction processing system. OLTP record data in row and column form, the main focus of OLTP is to record, update and delete the data while transaction. In OLTP queries are short and simpler so it takes less time for processing and it also require a less space. The data of OLTP get update frequently. The example of OLTP data are web application, mobile application, ATM machine where short transactions modify status of our account. OLTP database store data in normalization form (3NF), these OLTP system become source of OLAP (techdifferences, n.d.).

**Benefits of using OLTP database system**

* Better for daily transactions of an organization.
* It can widen customer base of an organization just by simplifying individual process.

**Drawbacks of using OLTP database system**

* If it fallows hardware failure then online transactions get affect.
* OLTP permit multiple users to access and modify data at the same time which can created unpredictable situations.

## **OLAP**

OLAP stands for online analytical processing system. It is mainly use for analytical purpose. OLAP database stores historical data which are recorded in the OLTP. Using the OLAP system we can extract the information from a large amount of database and analyze it for decision making. If the transaction is failed in middle it will not harm data integrity as the user use OLAP system to retrieve from large database. User can fire the query again and extract the data for analyzing purpose. In OLAP system transaction are long so it takes more time for processing and requires more space in comparison to OLTP. The table in OLAP database are not normalized, also the data of OLAP system are never deleted. The example of OLAP database is viewing report of financial, budgeting, sales, marketing etc. (techdifferences, n.d.).

**Benefits of using OLAP database system**

* It can create single platform for all types of business analytical needs, that include budgeting, planning, forecasting and analysis.
* We can easily apply security restrictions over user and apply regulations to protect the sensitive data.
* It can help is consistency of calculations an information

**Drawbacks of using OLAP database system**

* In OLAP database system implementation and maintenance are dependent on the IT professional, the reason behind this is traditional OLAP need complicated modeling procedures.
* In OLAP, we need co-operation between employee of different department to be effective which might not always possible

## **Difference between OLAP and OLTP system**

|  |  |  |
| --- | --- | --- |
| **Parameters** | **OLTP** | **OLAP** |
| Table | Table in OLTP are in normalized form | Table in OLAP database are not in normalized |
| Source | OLTP and regular transaction are the source of OLTP | OLTP databases are the source of OLAP database |
| Query | Inset, update and delete are the normal query of OLTP | Mostly select operations |
| Method | OLTP use mostly traditional DBMS | OLAP uses data warehouse |
| Functionality | This is online database modifying system | OLAP is online Databse query management system. |
| Process | Generally online transactional system. It manages database modification. | OLAP is online analysis retrieving data system. |
| Characteristic | It is characterized by large number of short online transactions. | It is characterized by large volume of data |
| Uses | Help to control and run basic needed tasks | Help with planning, problem solving and to make the decision. |
| Operation | Allow read/write operation | Only read and rarely write |
| Purpose | Specially designed for real time business operations | Designed for analysis purpose of business measure by category and attributes. |

# Snowflake Schema

In general snowflake schema is an extension of star schema, the difference is that the dim table in snowflake schema is divided into more table, which create a snowflake pattern. Which means dimension tables are normalized which one dim table into additional dim tables. The main purpose of snowflake schema is to normalize the denormalized data in a star schema. Snowflake schema is multi-dimensional structure. Through this snowflake schema, we normalize dim tables by splitting tables into more tables until that dimension tables are completely normalized. Unlike in the star schema dimension tables of snowflake schema can of their own categories, the main idea behind this is to normalized dimension tables. Each dim tables can be described by one or more lookup tables. Each lookup tables are separated until the model is fully normalized. This process of normalizing star schema’s dimension tables is called the snowflaking (Smallcombe, 2020).

**Characteristics of snowflake schema**

* Snowflake schema make smaller disk space.
* Easier to implement a dimension is added to the schema
* Because of multiple tables performance of query is reduced.
* The main challenge of snowflake schema is that you need to perform relatively more maintenance efforts because of more lookup tables.

**Benefits of snowflake schema**

* There are some certain OLAP database tools which are used by data scientist for data analyzing and modeling which are particularly designed to work with snowflake.
* Normalize data normally get denormalized in star schema can reduce the disk space requirements, this is because you are converting long strings of non-numerical data to numerical keys which are vividly less taxing from a storage perspective.

**Challenges of snowflake schema**

* Snowflake schemas create complexity while normalizing the attributes of a star schema. This complexity requires more complicated join query. In offering a more effective way to store data, snowflake can result in performance declines while browsing complex joins.
* Because of complex join in snowflake schema it may result in slower data processing. The star schema is normally better for cube data processing.
* Snowflake schemas provide greater normalization and low risk of data corruption after insert and update commands, but they do not provide the level of transnational guarantee that offer by traditional and highly normalized database structure. So, when we have to load the data in snowflake, we should be very careful about the quality of information.

# Data dictionary

In general data dictionary refers to the collection of definitions, names, and attributes for data models and elements. Data dictionary define purpose and meaning of data elements within the context of a project and offer guidance on interpretation, representation and accepted meaning. It also provides the metadata about the data elements. Metadata of data dictionary can assist in defining the characteristics and scope of data elements along with the rules for their usage and application. Data is very important for many reasons like it prove assist in providing data inconsistencies across a project. It can make a data easier to analyze, not only this it can enforce the use of data standards. The main reason companies are using data dictionaries is to document and share data structure and some other information for all involved all database. Using share dictionary can ensure meaning, quality, and relevance for all data elements for all team members. Without using the data dictionary there is also high risk of losing crucial information in transition (ucmerced, n.d.).

# ETL

ETL stands for Extract, Transform and Load. It is the process in which ETL tool extracts the data from numerous data source systems, transforms data in staging area and then finally load it into the Data warehouse system. It is tempting to think create data warehouse is just simply to extract data from multiple source and load data in the data warehouse, but far from the truth it requires a complex ETL procedure, the ETL procedures require dynamic input from multiple stakeholders including testers, analysts, developers, top executives and is technically challenging. To maintain the value as tool for decision-making, Data warehouse system need to change with business changes. ETL (extraction, tranform, load) is recurrent process like weekly, monthly for a data warehouse and it needed to be automated, agile and should be well documented. Let us understand each of the ETL process in depth.

1. **Extraction**

This is the fist step of ETL. This is the extraction process, in this step data from multiple source which can be in various formats like XML, SQL, NO SQL, relational databases, into the staging area. It is necessary to extract the data from various source and store it into the stagging are first, not directly into the data warehouse because extracted data is in multiple formats and it can be corrupted too. So, loading the data directly into the data warehouse may cause problem and it may be very difficult to rollback. Therefore, this is one of the most important steps in ETL procedures.

1. **Transform**

This transformation process is second step of ETL process. In this stage, some rules or functions are applied on the extracted data to convert it into a single standard format. This transformation may involve following process.

* Filtering
* Cleaning
* Joining
* Splitting
* Sorting

1. **Load**

Third and final step of ETL procedures is loading. In this step transformed data is finally loaded into the data warehouse occasionally data is updated by loading into data warehouse very often and sometimes it is done after longer but regular intervals. Load time is depending upon the requirements and varies from system to system.

# Project overview

FlyU airline company keep the records of flight that leave each airport. They also kept the record of departure arrival times, along with actual arrival and departure time from and to each airport. They also kept the record of passenger on each flight. Some passenger needs number of flights to reach the destination, passenger can also raise complaint against flights if they have any connections. Here company want to analyze mainly on three topics

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

Here my role is an analyst/developer on data mart project to support the design analysis and collection of information relating of FlyU, I have to concern with company grow. To achieve this requirement, I have made some aim and objectives of the project, which are given below.

## **Project aim and objectives**

The main aim and objective of the project is to analyze the data so that company can make better decision to grow the companies. For this we have to design, develop and implement the data warehouse for the FlyU company.

# Reports to be supported by star schema

1. Number of customers per airport per year
2. Airlines with most complain
3. Number of flight and amount of minute that have arrived late
4. Number of cancelled flights each year
5. Number of flights that are diverted each year

Furthermore, the designed star schema would also support the report mention below at a much detailed such as:

# Star schema

Star schema is one of the most important and simplest schemas. This schema is widely used to develop and build data warehouse and dimensional data marts. Star schema consist of one or more than one fact tables indexing any number of dimensional tables. This schema is necessary for the snowflake schema and it is also efficient to handle basic queries. Star schema form a star shape having fact table at the center surrounded by the dimensional table.

The purpose of star schema is to separate numerical “fact” data form the descriptive or ‘dimensional” data. The fact table will have only numerical data like weight, price, speed, quantity, i.e., data which are in numerical format. Dimension table will have data like model, color, names, location employees name, salesperson name etc. i.e., descriptive data along with numerical information.

Fact data is organized into the fact table, and dimensional data is into dimension tables. Fact tables are the center of star schema in data warehouse. They allow to analyze the data as a single unit, and allow business systems to access the data together. Dim table manage and hold the numerical and non-numerical data which coverage through fact tables that make up the data warehouse. Fact table keep the track of numerical information related like they might include numeric values, foreign key that map additional dim tables. Dimension table normally list a surrogate primary key which maps to attributes related to the natural key. Let’s say we have dim table with the information related to store “dim\_store”. You can allocate an ID to each store and its row related non-numerical and other information like size, name, location, category etc. Wherever we list the store ID number on fact table “fact\_sales” will map to that specific row of store data on the “dim\_store” dim table (Smallcombe, 2020).

**Characteristics of star schema**

* Each dimension in star schema represent with only one-dimension table.
* Dim tables are connected with fact table using the foreign key
* Dim tables are not connected with each other
* Fact table would contain key and numeric value
* Star schema is easy to understand
* Star schema is supported by BI tools.
* Dim table are not normalized in star schema

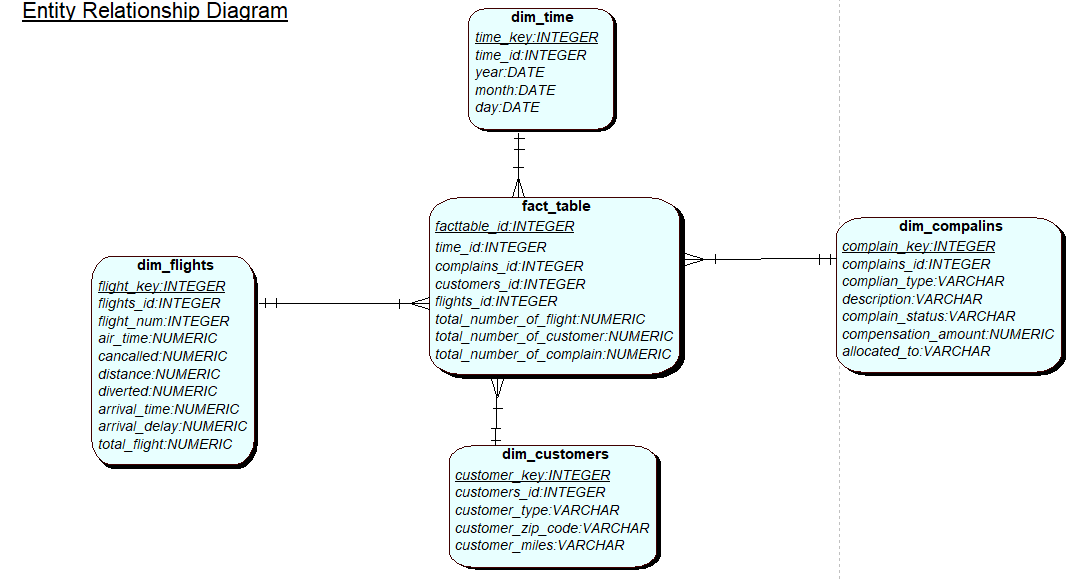
**Benefits of star schema**

* In this model all the data are connect through the fact table and multiple dimension tables are considered as the large table of information which make query simple and easy to perform
* It is easy to pull the busines reports in star schema
* It removes the bottlenecks of highly normalized schema, increase the query speed, and improve the performance of read only command.
* OLPA system can use star schemas to build OLAP cubes.

**Challenges of star schema**

* Because of denormalized form of star schema, it does not enforce in data integrity, though star schema use countermeasures to stop from anomalies, a simple update and inset can still cause data incongruities.
* Database designers shape and optimize star schemas for particular analytical needs. Because of deformalized data sets, they work with comparatively narrow set of simple queries. Relatively normalized schema allows a far wider variety of more complex analytical queries.

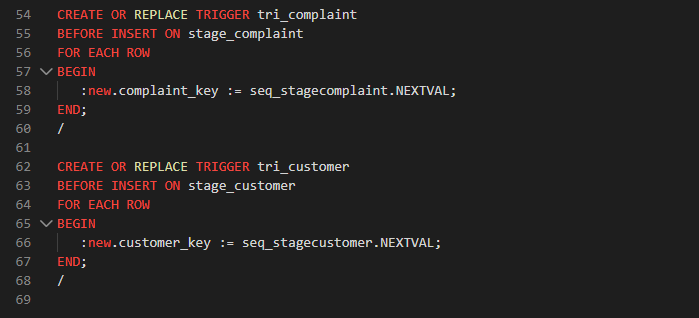
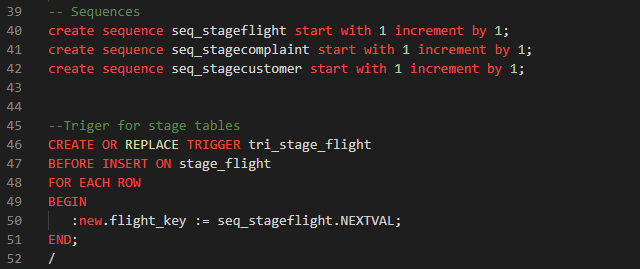
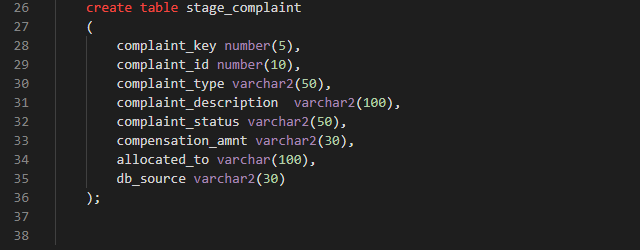
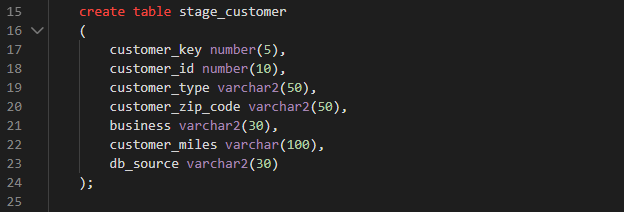
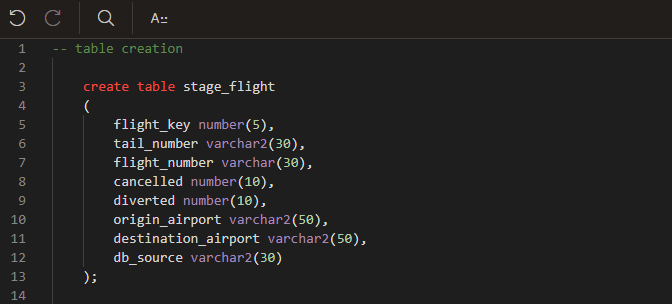
**Star schema of this project is given below:**



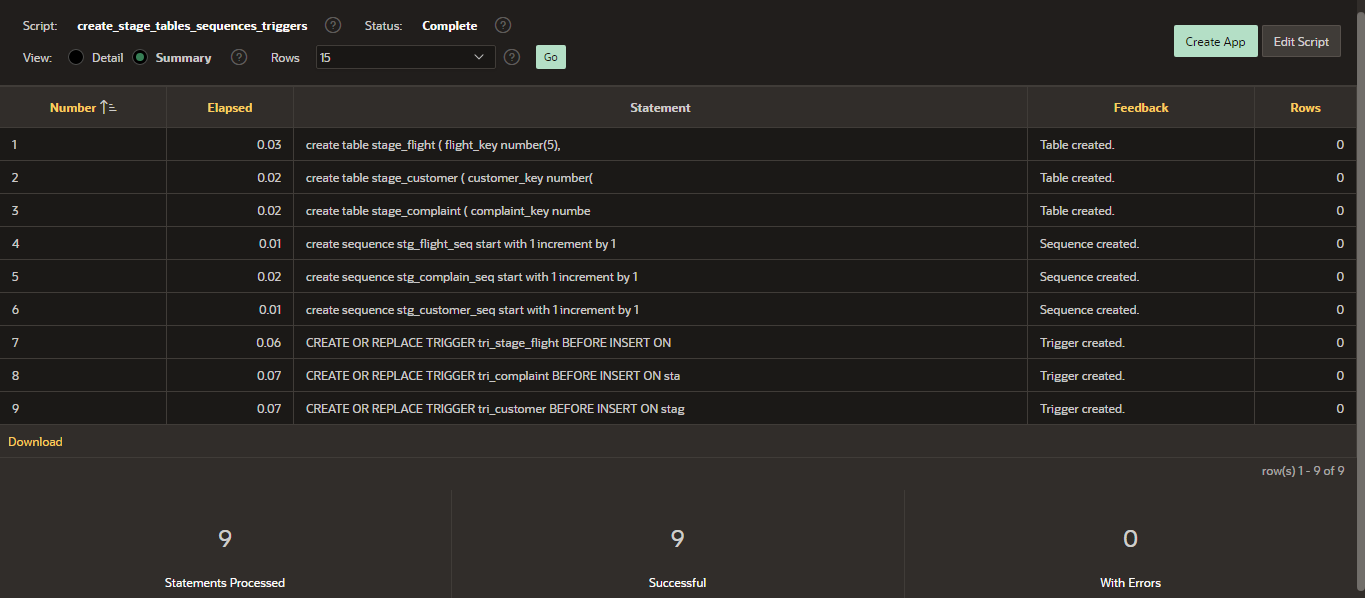
# Load data from sources

# Stagging

## **Stage table creation**



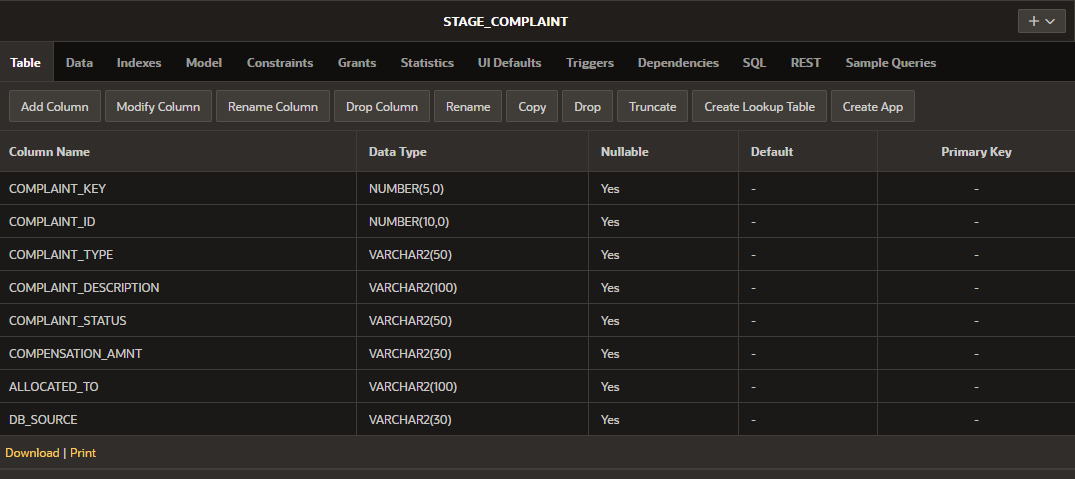
**Fig: script to create stage tables with require sequences and triggers**



**Fig: successfully created stage tables with required sequences and triggers**

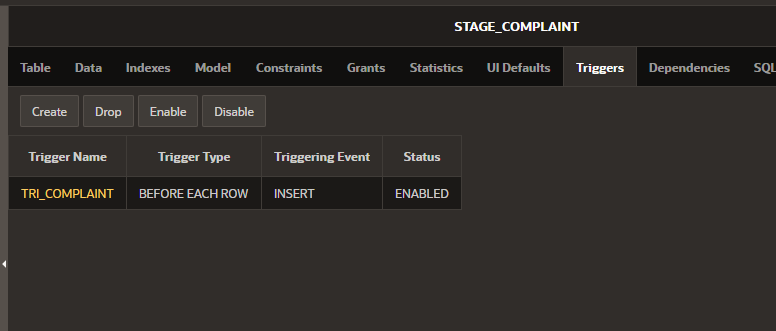
As seen in the above we have created all the stage tables and all the related sequences and triggers successfully. Stage tables are flight\_stage, stage\_complaint, stage\_customer. Sequences are stg\_flight\_seq, stg\_complain\_seq, stg\_customer\_seq. Triggers are tri\_stage\_flight, tri\_compalin, tri\_customer.

## **Stagging stage\_complaint table**

**Fig: Structure of stage\_complaint (stage table)**

The above shown image is the structure of stage\_complaint table. ‘STAGE\_COMPLAINT’ is the name of this table.

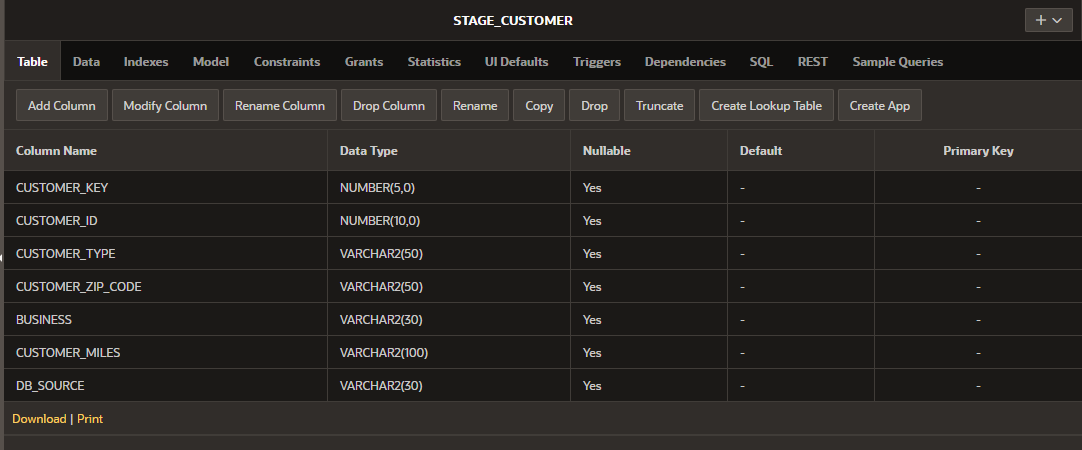
## **Trigger of stage\_complaint table**



**Fig: stage\_complaint trigger**

The image shown above is the trigger of stage\_complaint table. ‘TRI\_COMPLAINT’ is the name of trigger of stage\_complaint table.

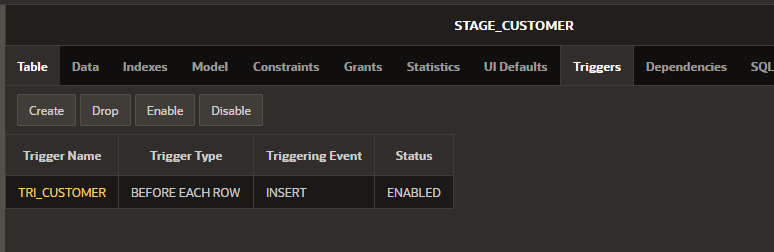
## **Staging stage\_customer table**



**Fig: Structure of stage\_customer table**

The above shown image is the structure of stage\_customer table. ‘STAGE\_CUSTOMER’ is the name of this table.

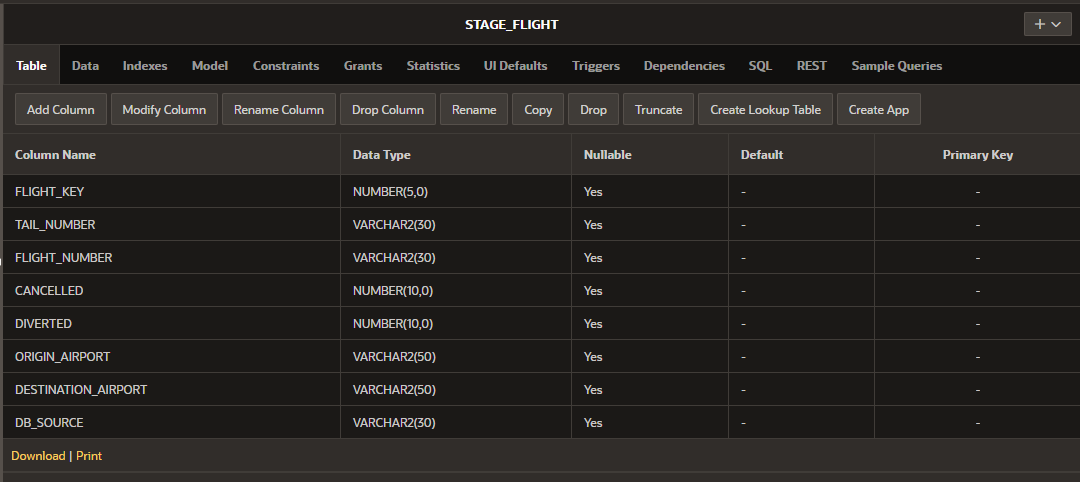
## **Trigger of stage\_customer table**



**Fig: trigger of stage\_customer table**

The image shown above is the trigger of stage\_customer table. ‘TRI\_CUSTOMER’ is the name of trigger of stage\_customer table.

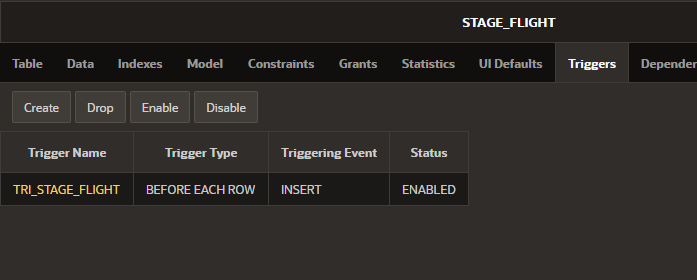
## **Stagging table of stage\_flight table**



**Fig: Structure of stage\_flight table**

The above shown image is the structure of stage\_flight table. ‘STAGE\_FLIGHT’ is the name of this table.

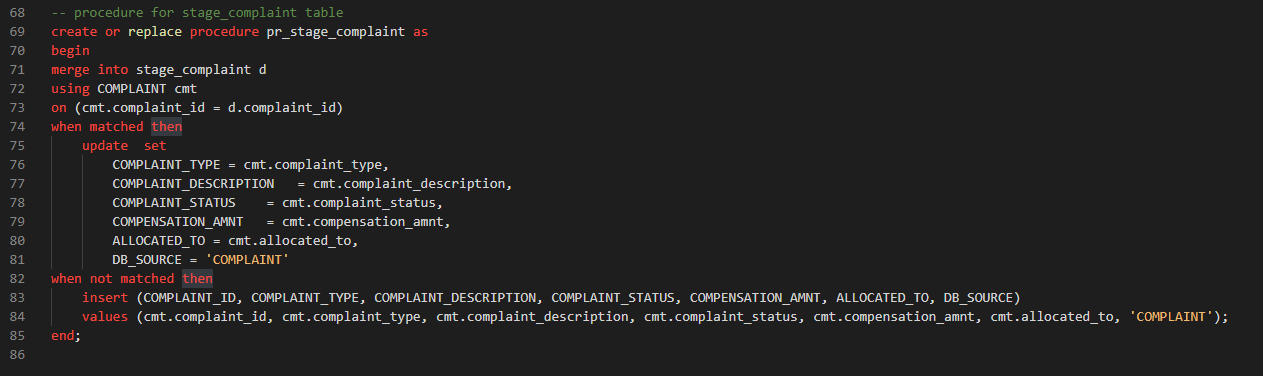
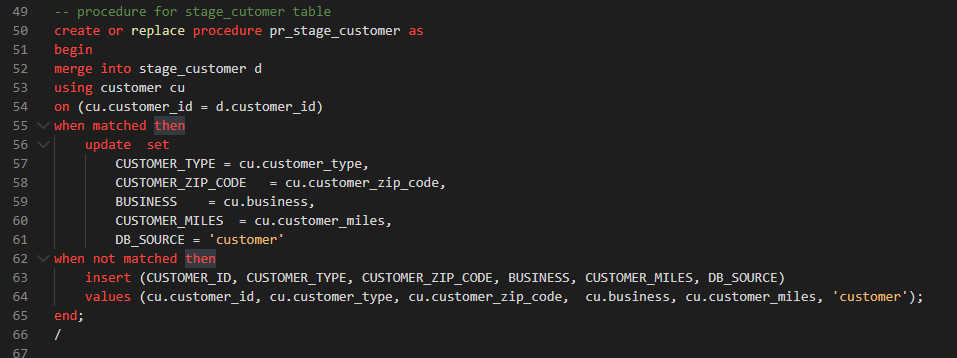
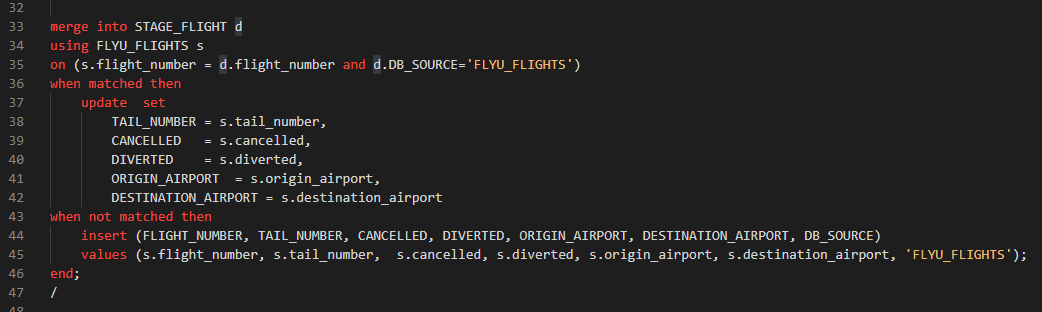
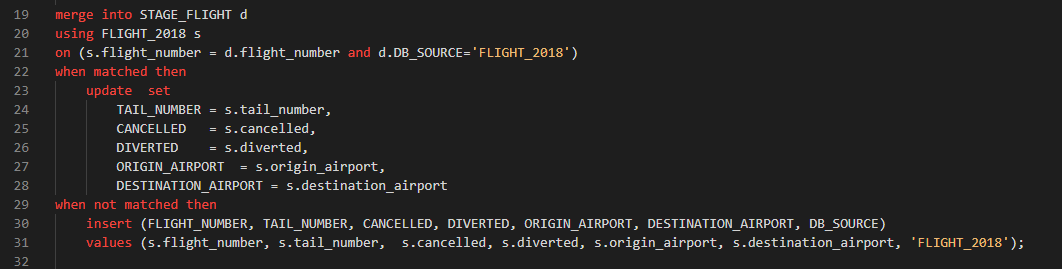
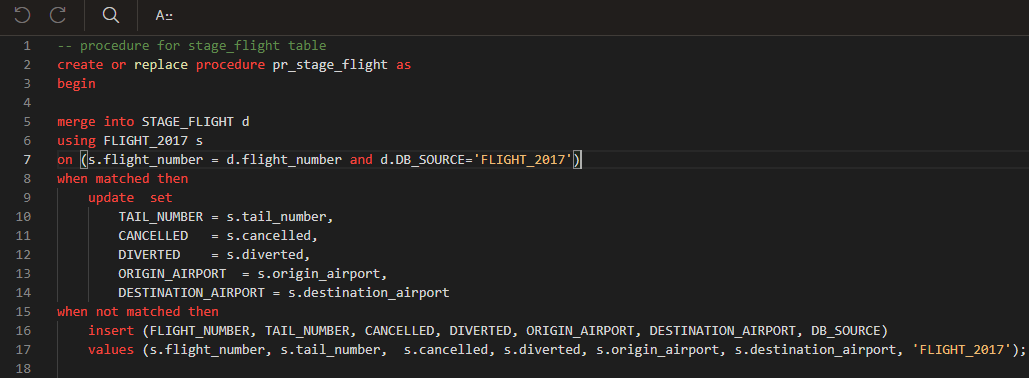
## **Trigger of stage\_flight table**



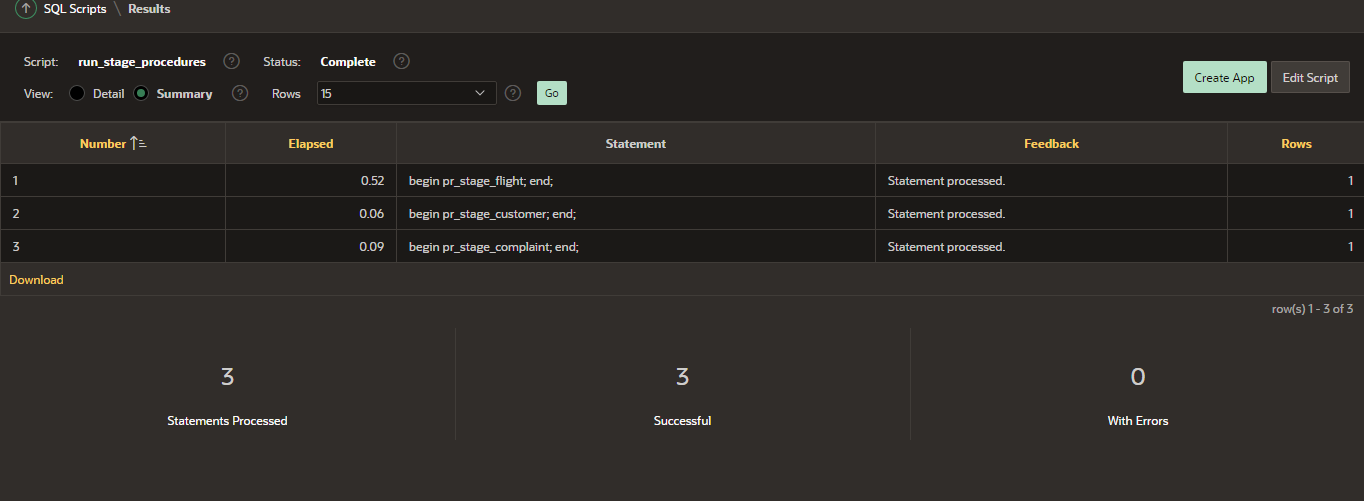
**Fig: Structure of stage\_complaint trigger**

The image shown above is the trigger of stage\_flight table. ‘TRI\_STAGE\_FLIGHT’ is the name of trigger of stage\_flight table

## **Create stable table procedures**

****

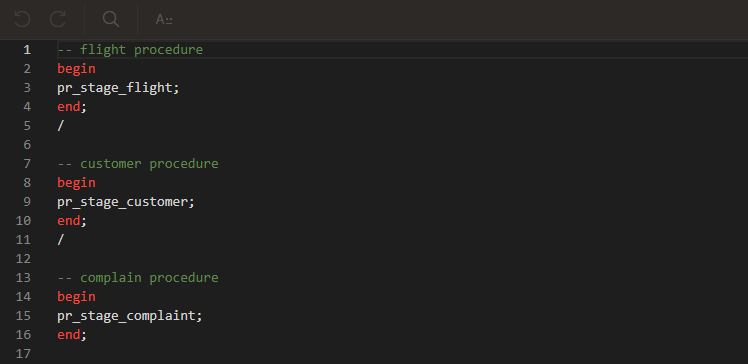
**Fig: script to create procedures for stage tables**



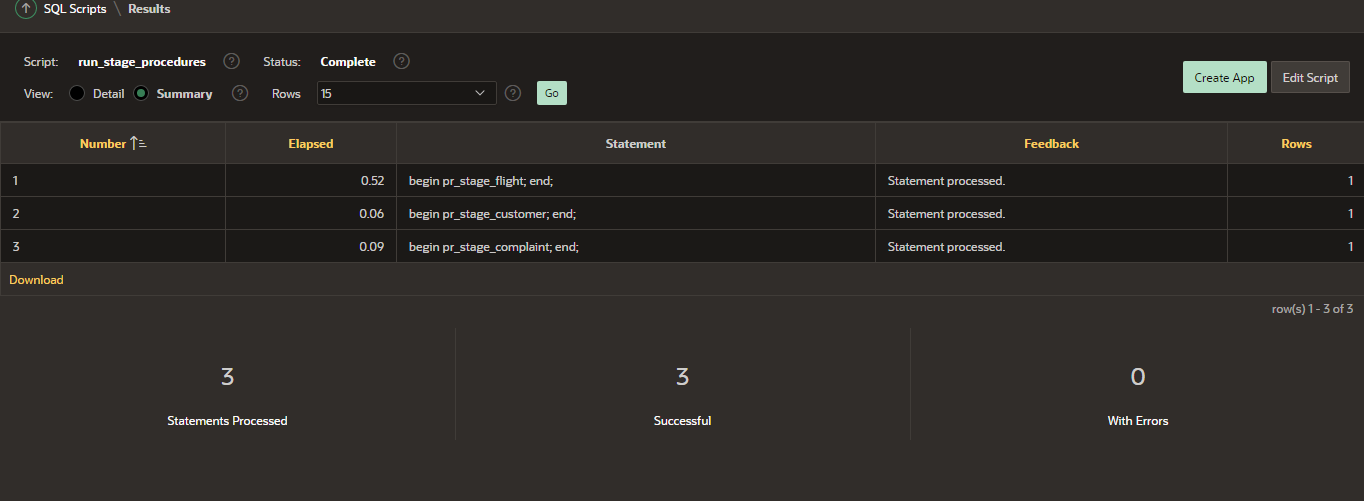
**Fig: successfully created stage procedures**

All the procedure of stage tables has been successfully created. Procedure names are pr\_stage\_flight, pr\_stage\_complaint, pr\_stage\_customer.

## **Run procedures of stage flight tables**



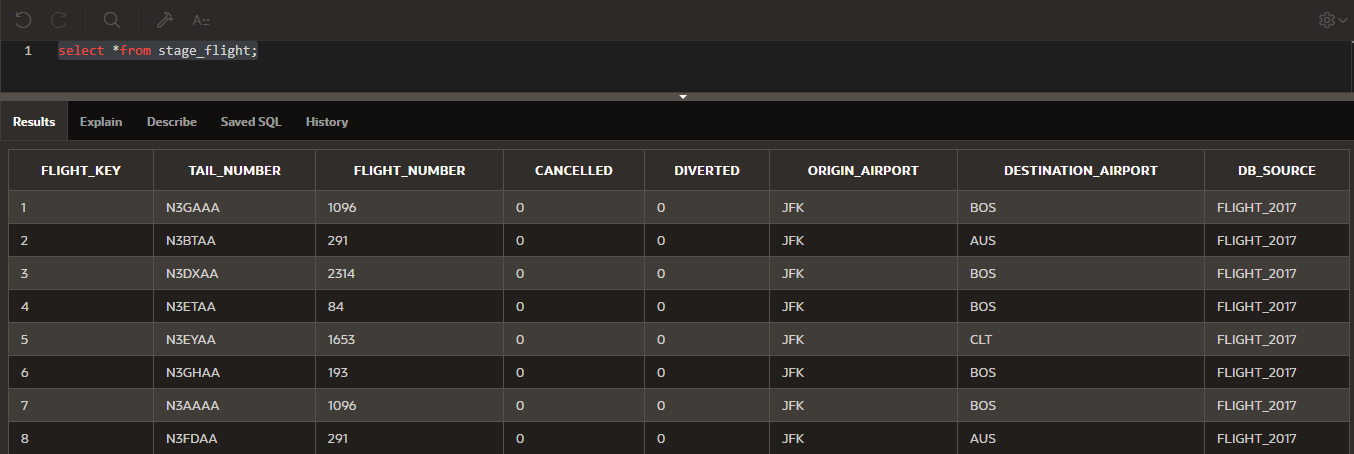
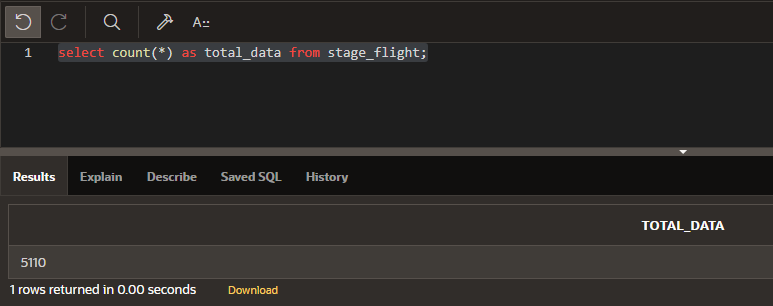
**Fig: script to run procedures**



**Fig: run procedures of stage flight tables**

Successfully run the procedures of stage flight

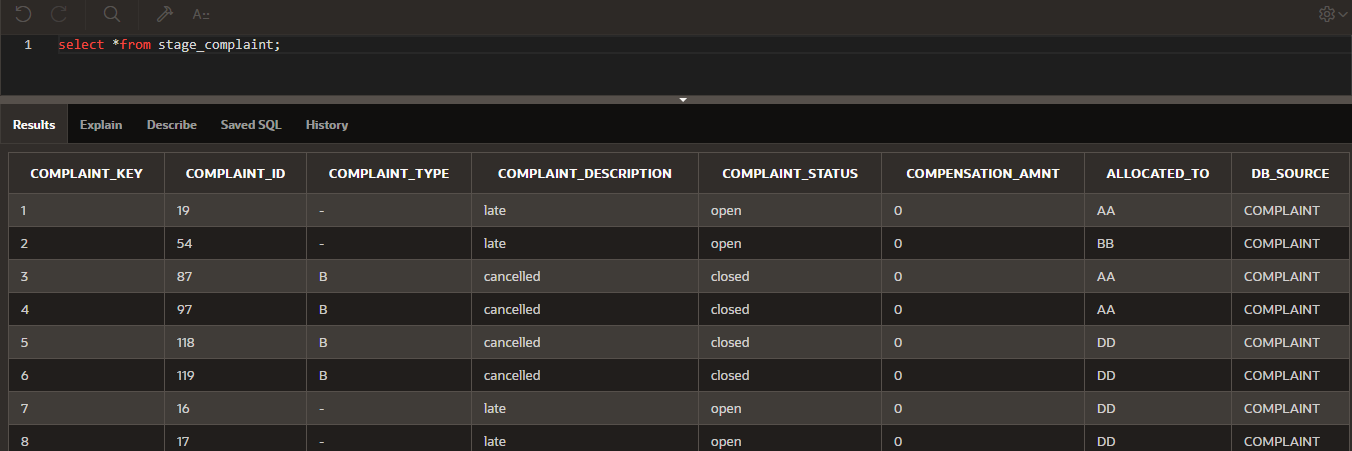
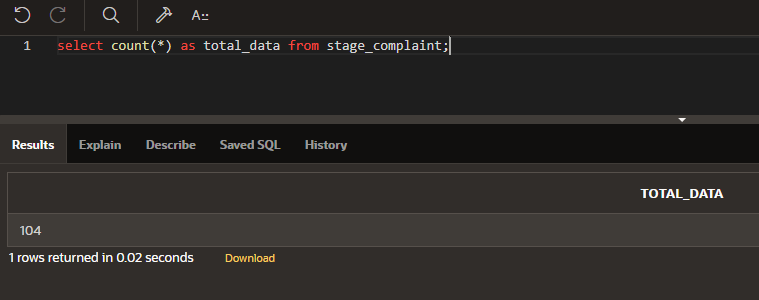
## **Data of stage\_flight table**



**Fig: count all the data of stage\_flight table**

We have successfully inserted 5,110 data into stage\_flight table from three data sources (flight\_2017, flight\_2018 and flyu\_flights).

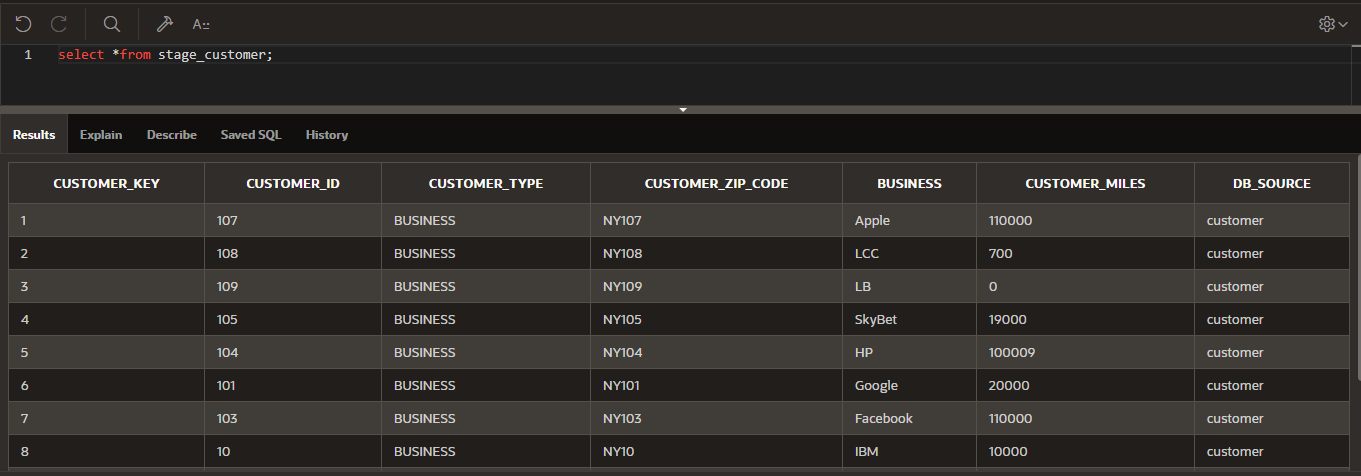
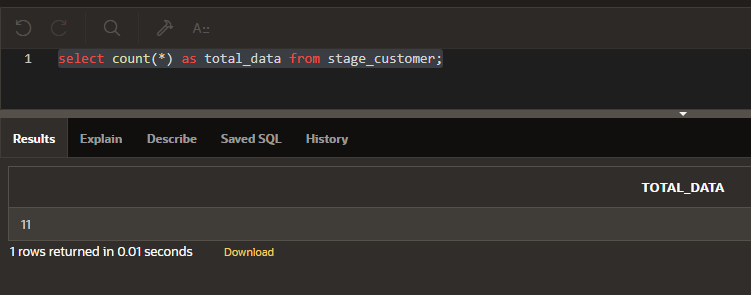
## **Data of stage\_complaint table**



**Fig: count all the data of stage\_complaint table**

We have successfully inserted 104 data into stage\_complaint table from complaint data source.

## **Data of stage\_customer table**



**Fig: count all the data of stage\_customer table**

We have successfully inserted 11 data into stage\_customer table from customer data source.

# Good data and Bad data

1. If tail\_number of planes is null then, it is bad data

Every plane has their own tail\_number

1. If diverted is greater than cancelled then it is bad data

If the flight is cancelled the there is no chance of flight diverted

1. Destination\_aiport name and origin\_airport name should be in alphabet
2. If complaint\_type is null then it is bad data
3. If complaint\_description is null then it is bad data

**Bad data in stage\_flight table**

* Tail\_number null
* if cancelled is one then diverted is one (make this)

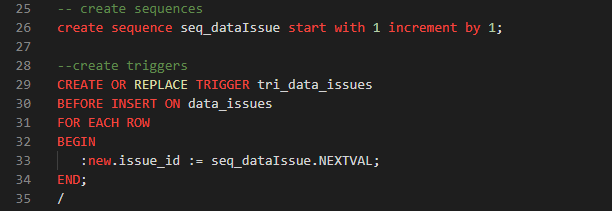
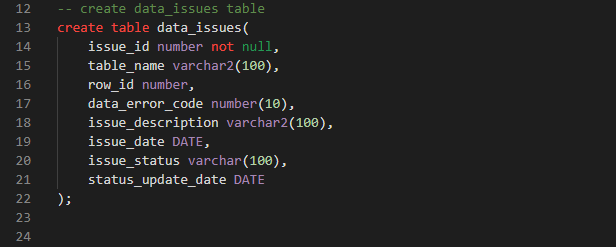
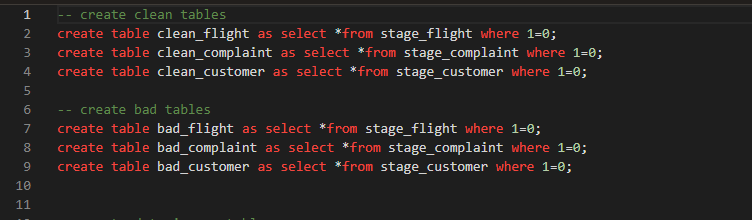
**Bad data in stage\_complaint table**

* Complaint\_type null
* Complaint\_description null

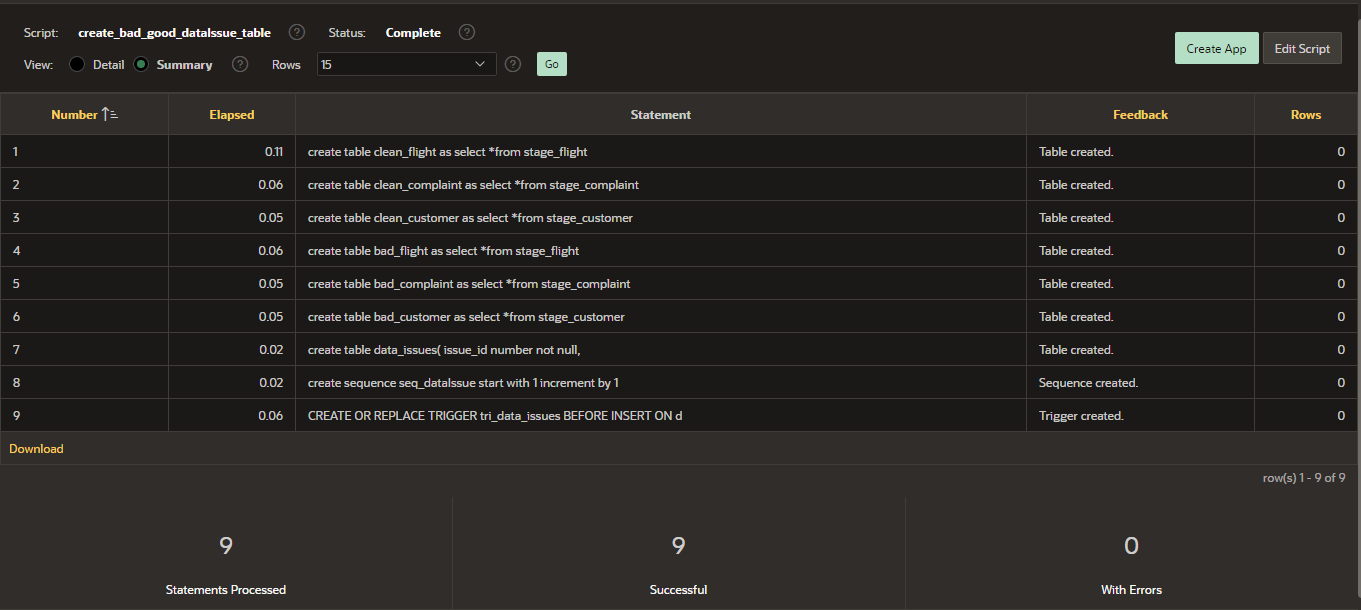
From business perspective – rules

* Every plane should have tail number
* If flight is cancelled then there can’t be flight diverted, which means if cancelled is 1 then diverted can’t be 1.
* Destination and origin airport name should be in alphabet
* If complaint\_type is null then there cannot be complaint description.
* Complaint\_type should be either A, B or C.
* Complaint\_allocated should be either to AA, BB, CC or DD.
* Airport name should be in alphabetic.
* Compensation amount should be only in number.

## **Create tables for good and bad data**



**Fig: script to create table for bad and good tables with require sequences and triggers**



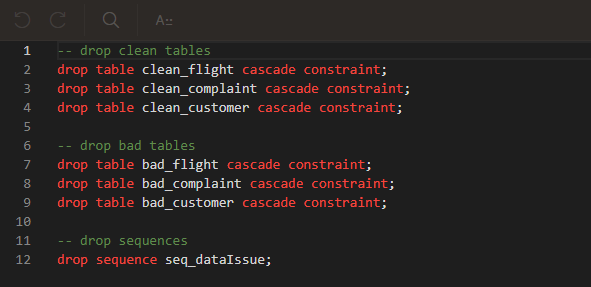
**Fig: successfully created table for good, bad data with related sequence and trigger**

Successfully created tables for good and bad data with related sequence and trigger. Tables are

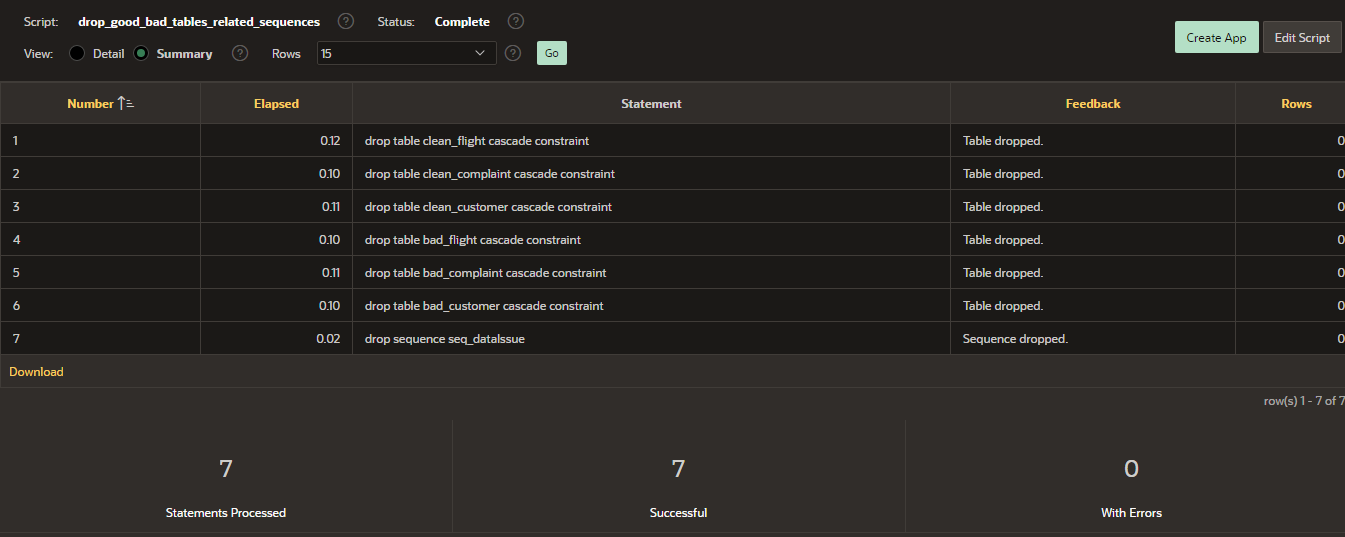
bad\_flight, clean\_flight, bad\_complaint, clean\_complaint, bad\_customer, clean\_customer. Sequence

seq\_dataIssue, and trigger tri\_data\_issue.

## **Drop script to drop bad and good data tables**



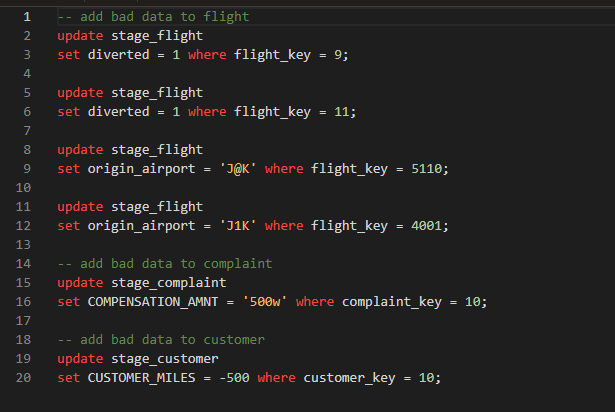
**Fig: script to drop good bad data tables and sequence**



**Fig: successfully dropped good, bad table and sequence.**

Successfully dropped all the tables with related sequence and trigger

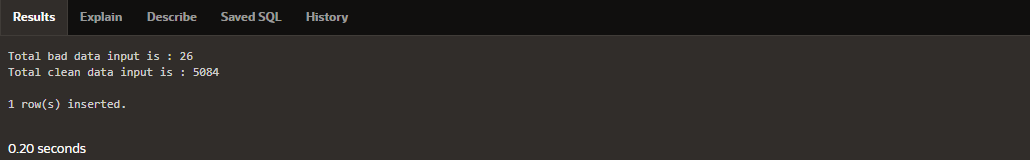
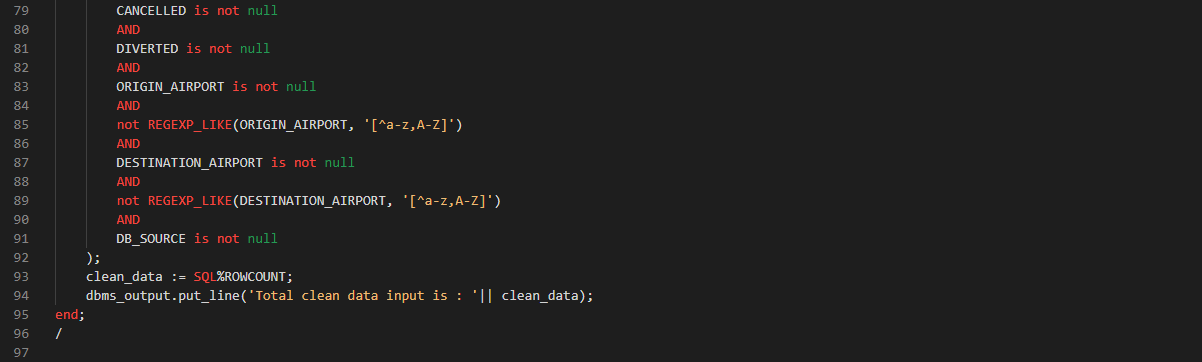
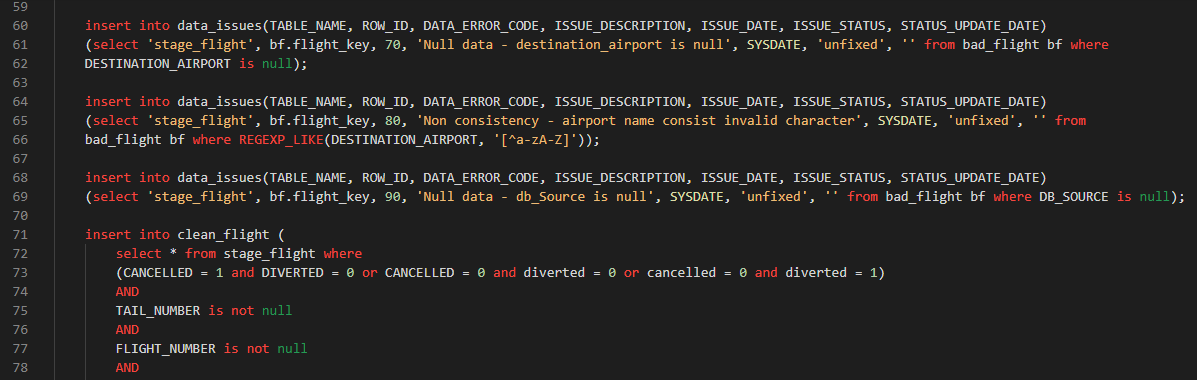
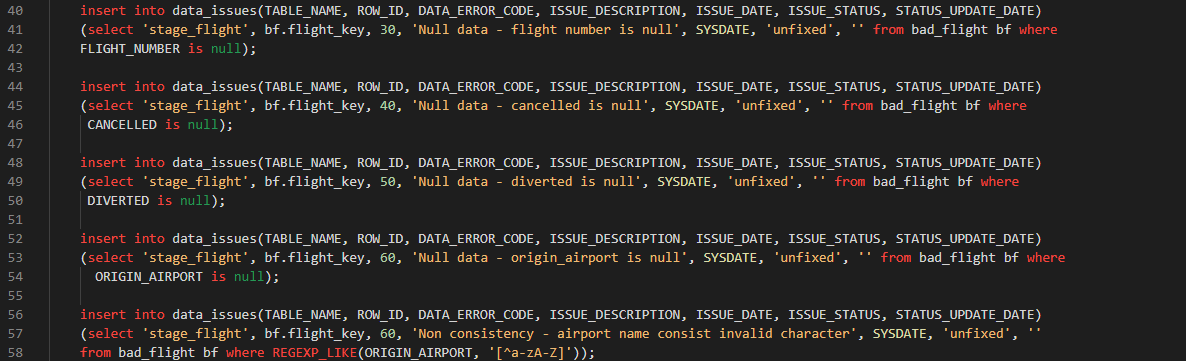
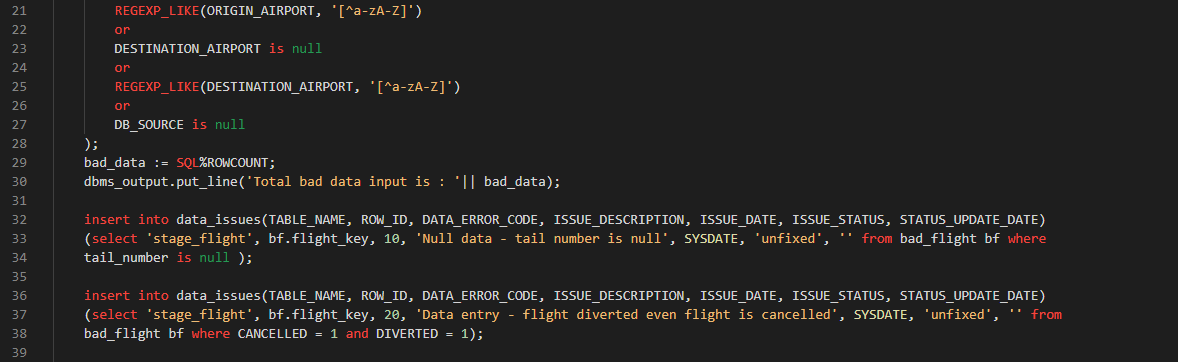
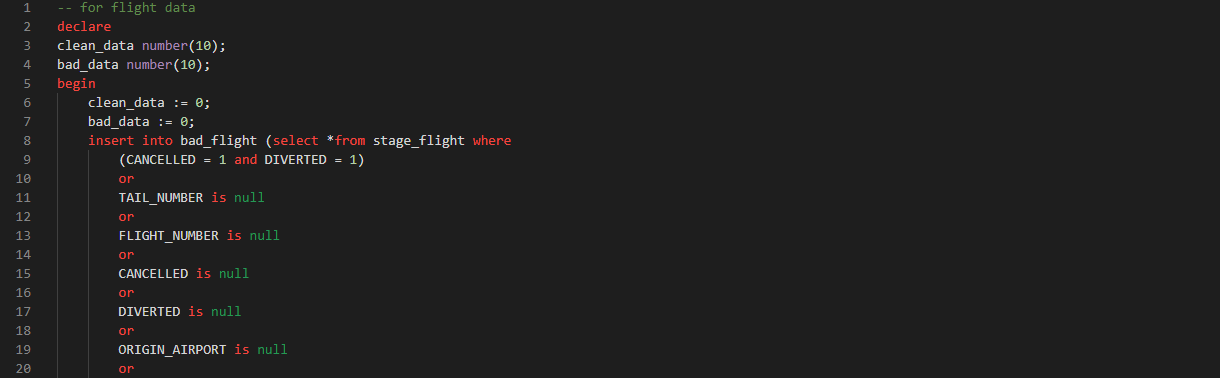
## **Add bad data to the table**



**Fig: script to add bad data in the tables.**

## **Insert good into clean tables and bad data into bad tables**

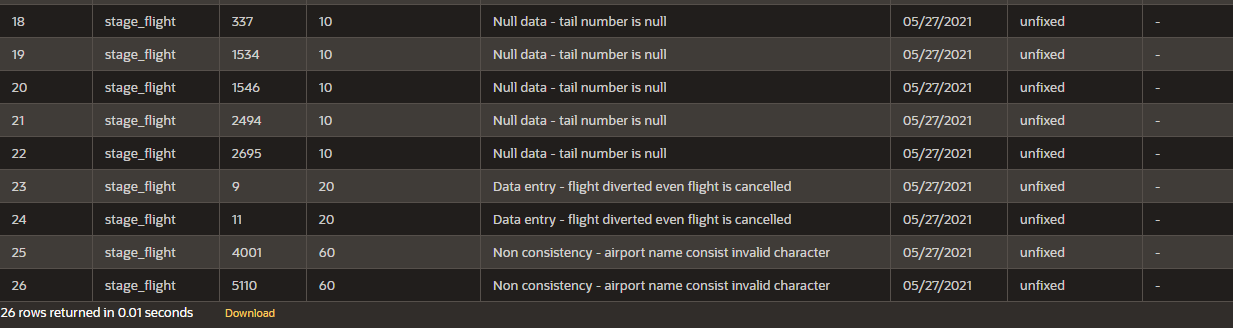
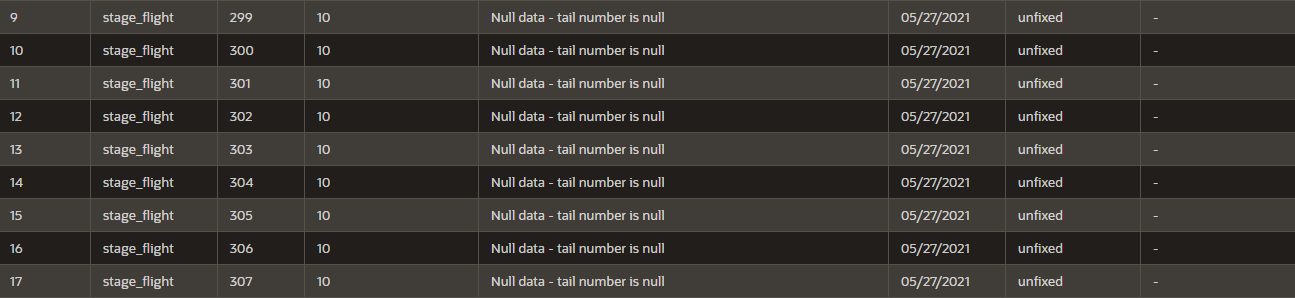
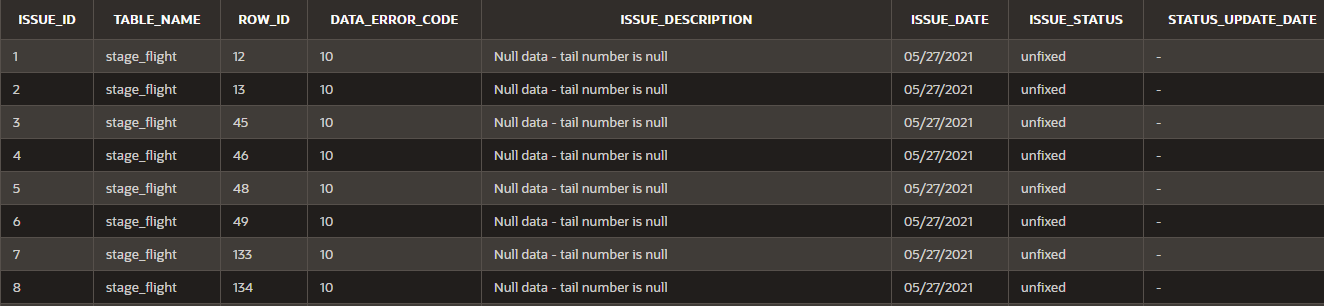
### Insert bad data of flight into bad\_flight table and good data into clean\_flight table



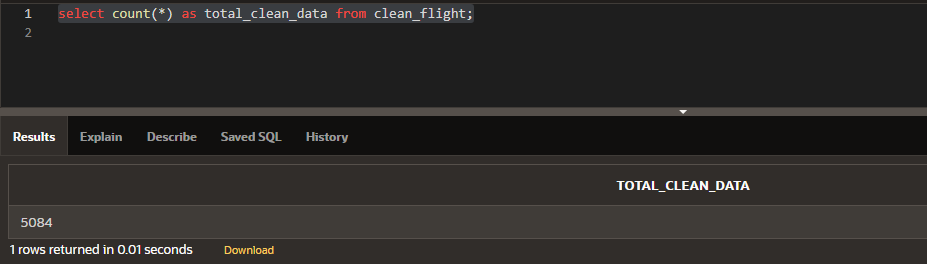
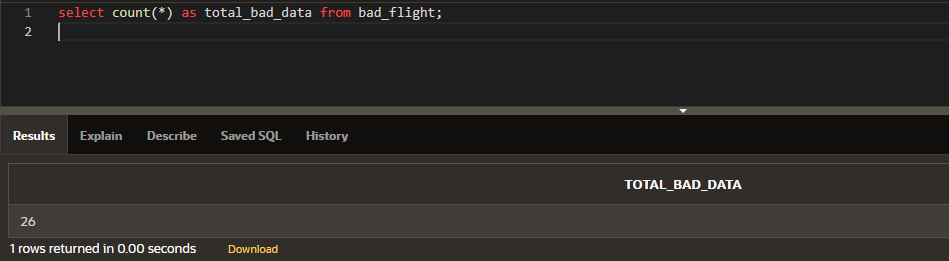
**Fig: script to insert bad data of flight into bad\_flight table and clean data into clean\_fligght table**

Above SQL code was written to insert good data into clean\_flight table and bad data into bad\_flight table. There were 26 bad data and 5084 clan data, and all the bad data is inserted into bad\_flight table while clean data is in clean\_flight table.

We have inserted the bad data into data\_issues table too. This table contain the all the information of bad data. Let’s see the detail of bad data from flight table.



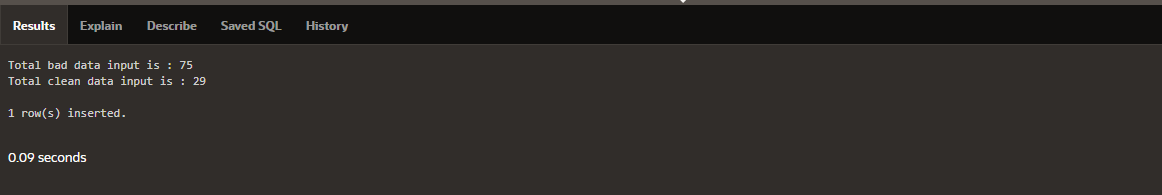
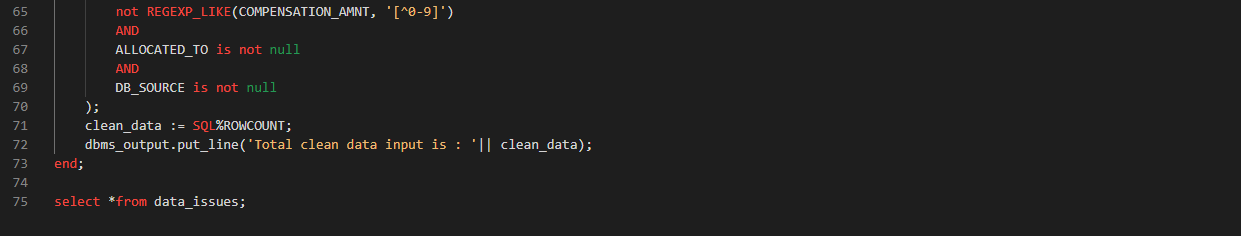
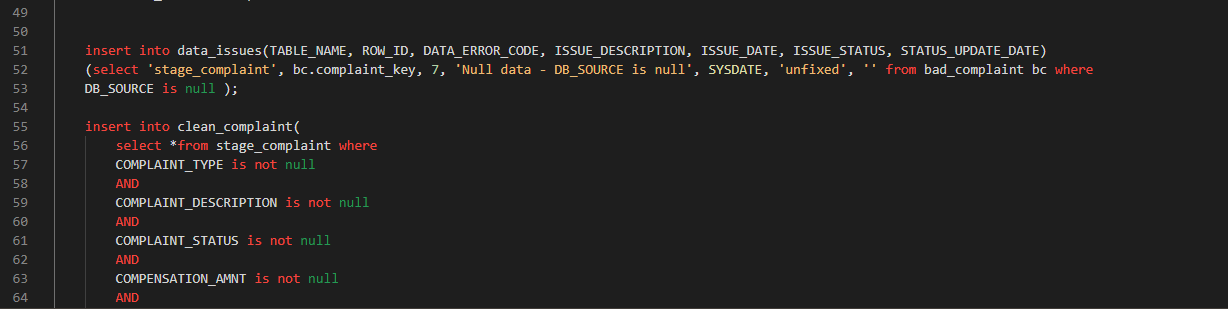
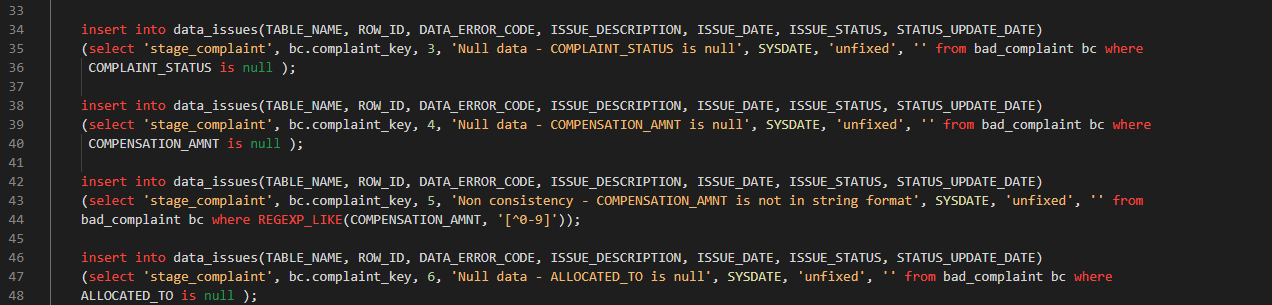
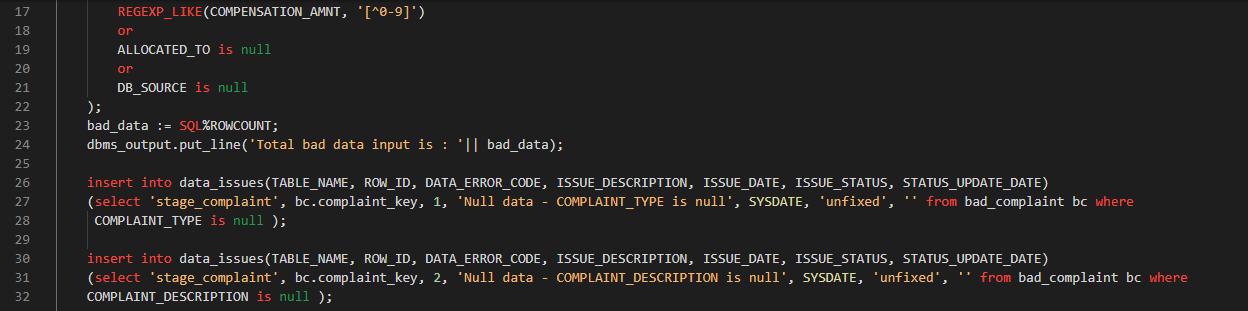
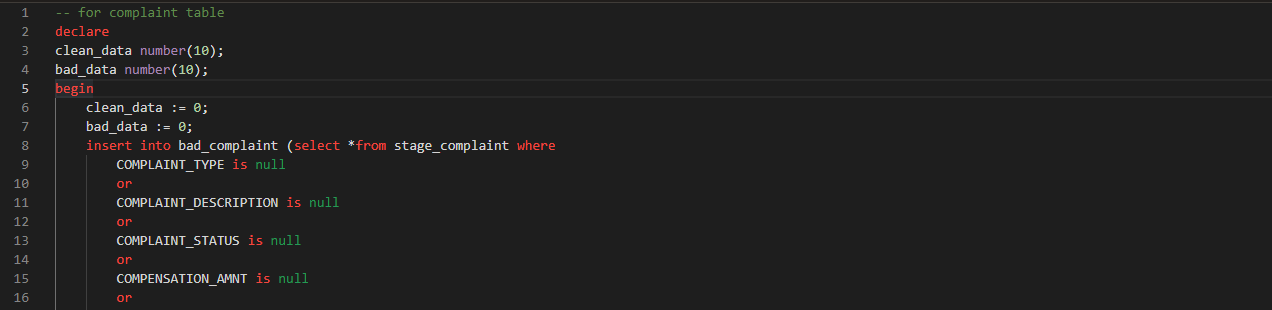
**Fig: information of bad data of flight table**

****

**Fig: total number of bad data and good data in bad\_flight and clean\_flight table**

Successfully inserted bad data into bad\_flight table and clean data into clean\_flight table.

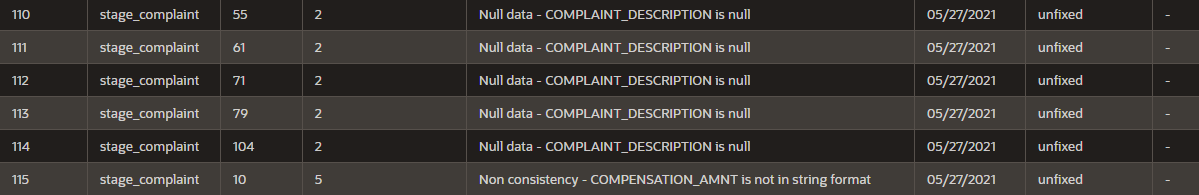
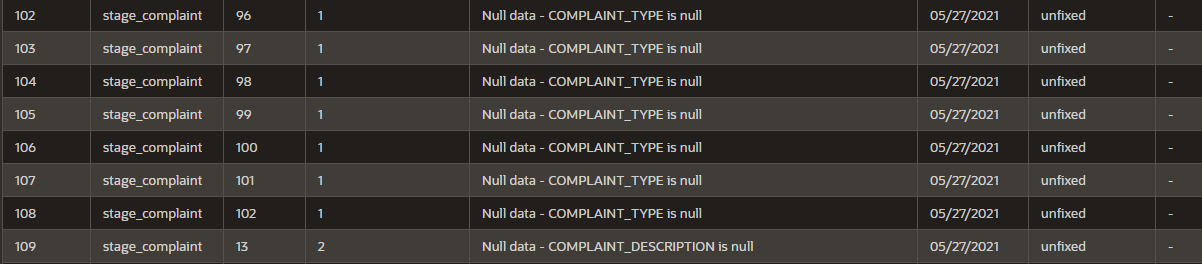
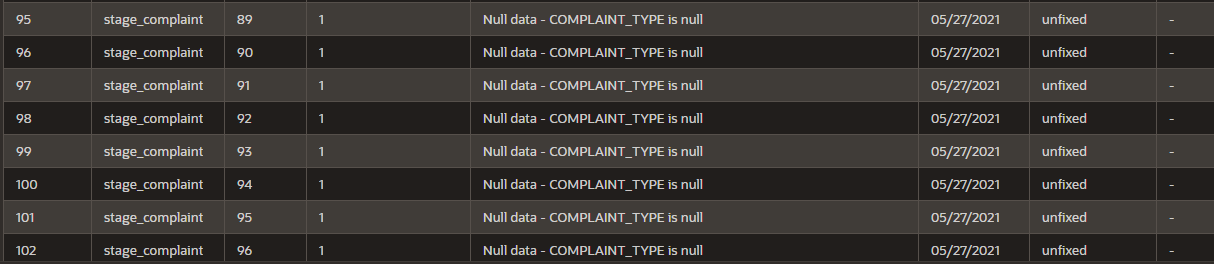
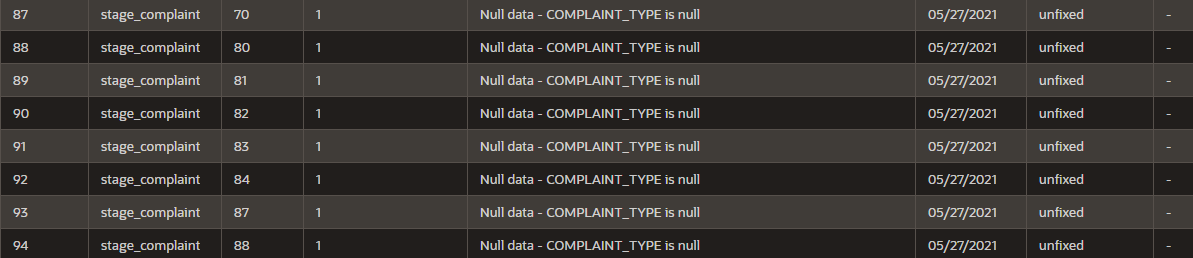
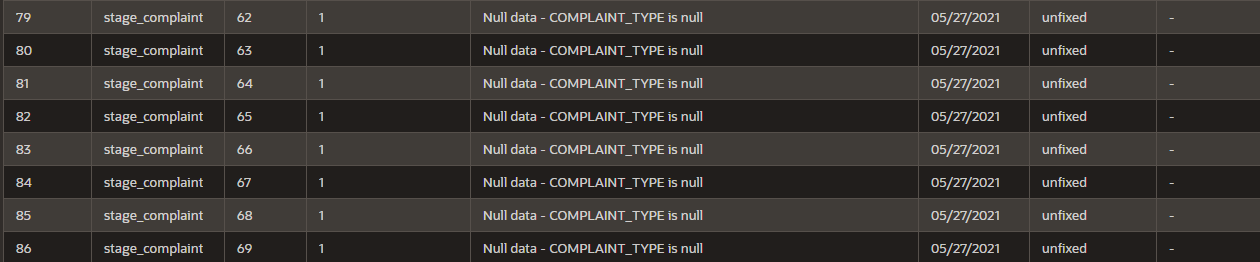
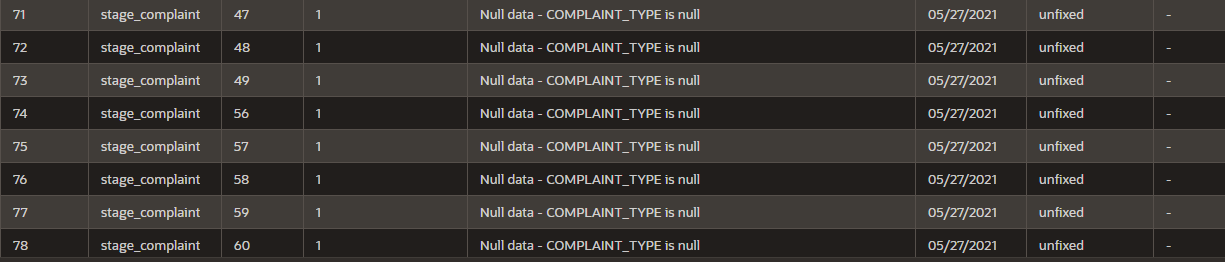
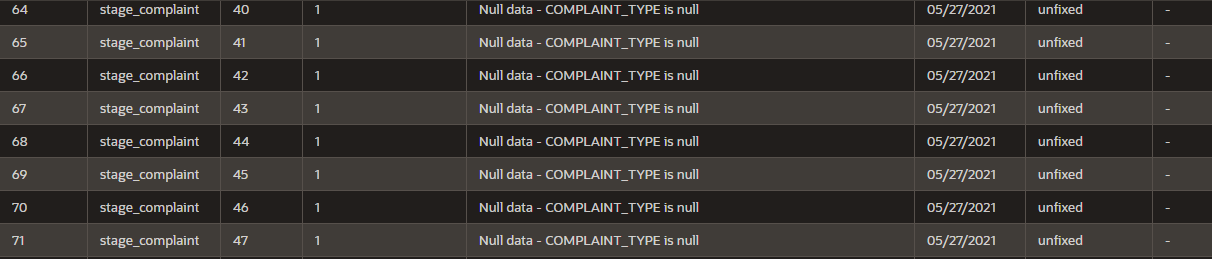
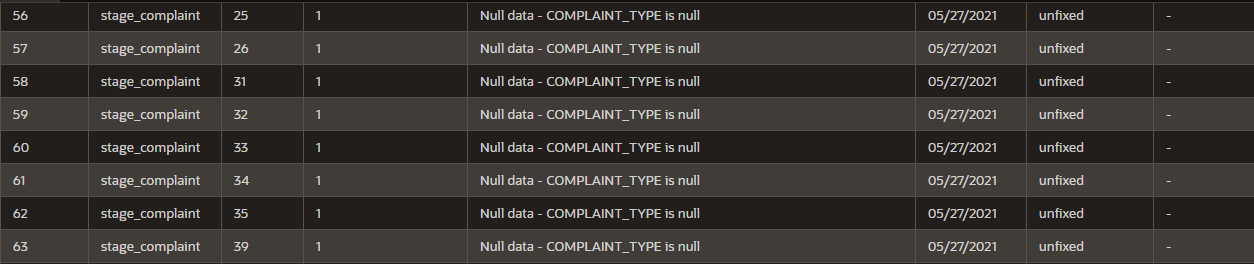
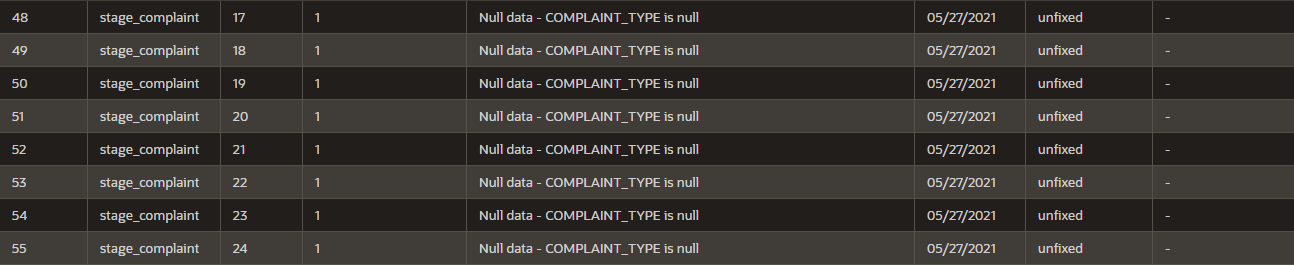
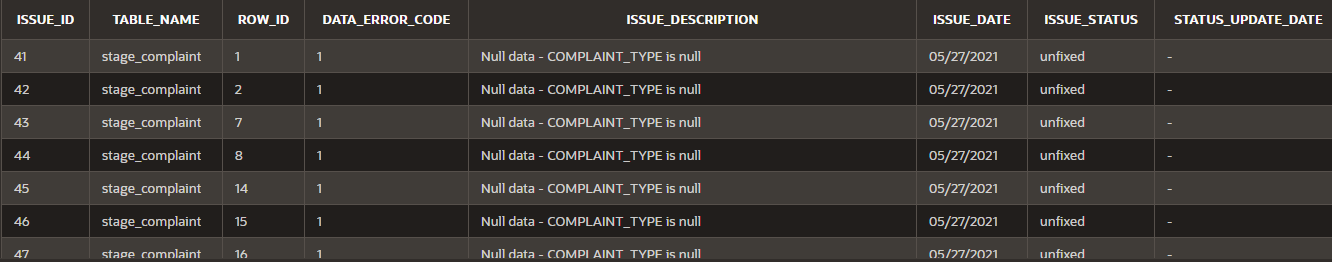
### Insert bad data of complaint into bad\_comlaint table and good data into clean\_Complaint table



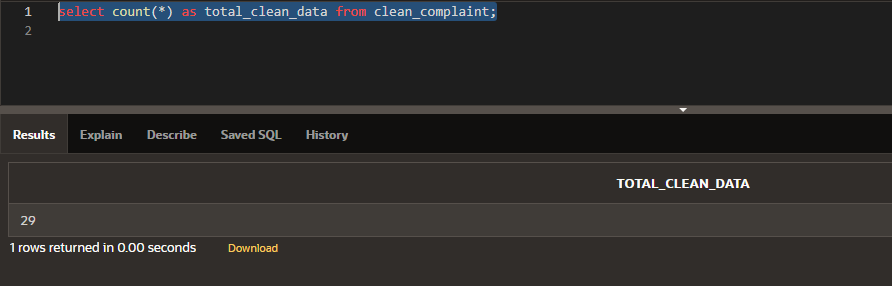
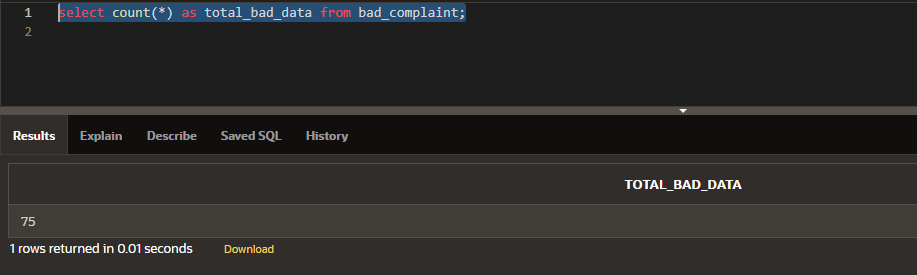
**Fig: script to insert bad data of complaint into bad\_compalint table and clean data into clean\_complaint table**

Above SQL code was written to insert good data of stage\_complaint into clean\_complaint table and bad data into bad\_complaint table. There were 75 bad data and 29 clan data, and all the bad data is inserted into bad\_complaint table while clean data is in clean\_complaint table.

We have inserted the bad data into data\_issues table too. This table contain the all the information of bad data. Let’s see the detail of bad data of stage\_complaint table.



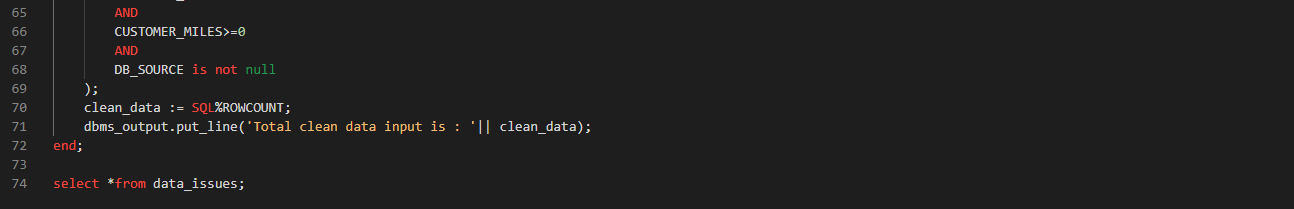
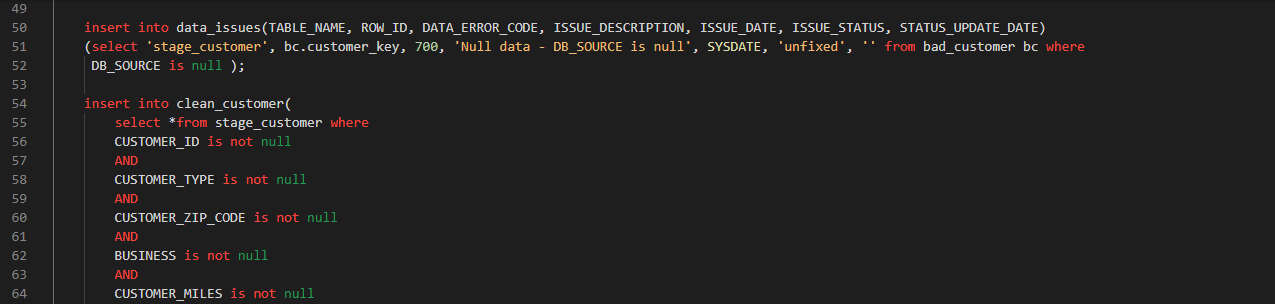
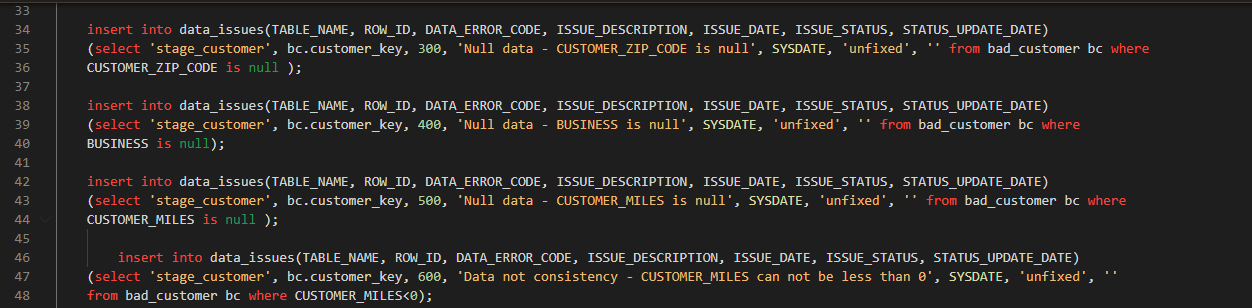
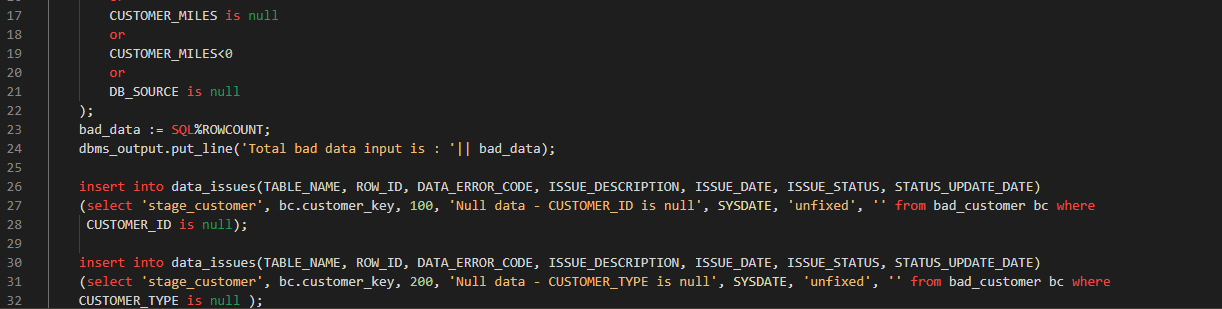
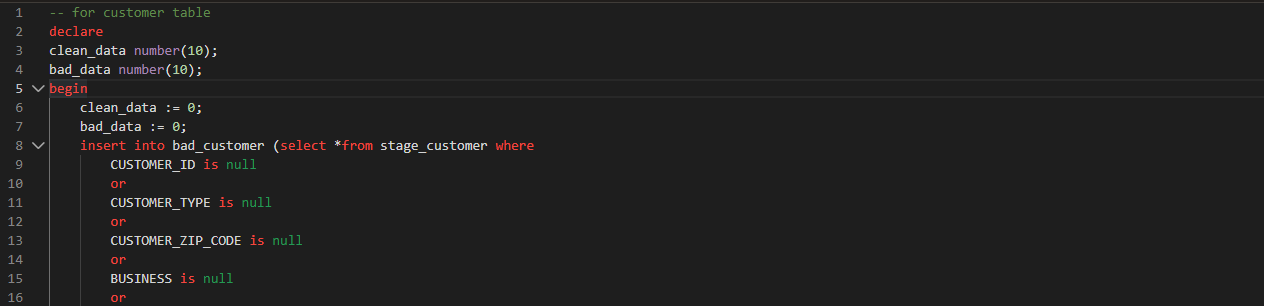
**Fig: information of bad data of complaint table**

****

**Fig: total number of bad data and good data in bad\_complaint and clean\_complaint table**

Successfully inserted bad data into bad\_complaint table and clean data into clean\_complaint table.

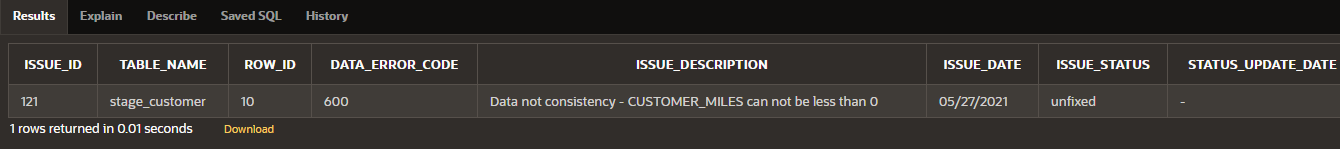
### Insert bad data of customer into bad\_customer table and good data into clean\_customer table



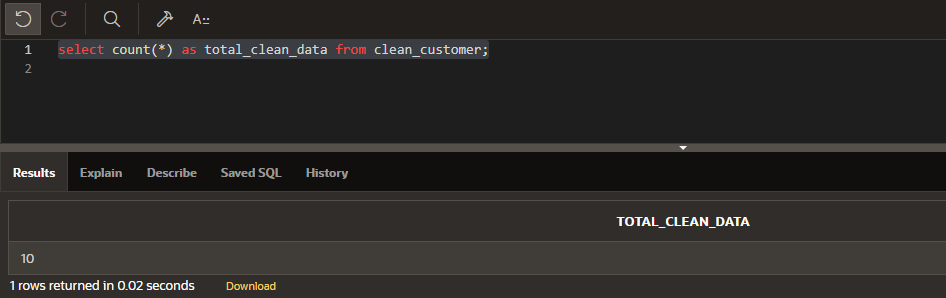
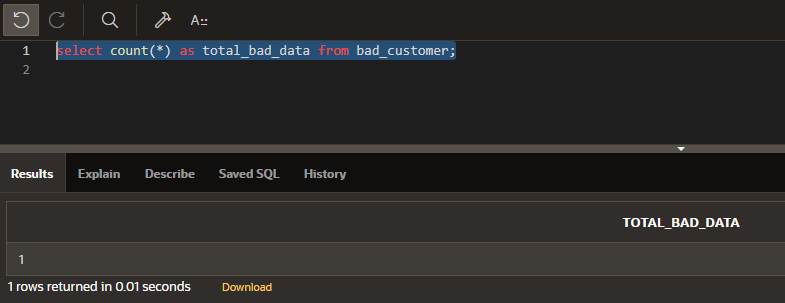
**Fig: script to insert bad data of customer into bad\_customer table and clean data into clean\_customer table**

Above SQL code was written to insert good data of stage\_customer into clean\_customer table and bad data into bad\_customer table. There were 1 bad data and 10 clan data, and all the bad data is inserted into bad\_customer table while clean data is in clean\_customer table.

We have inserted the bad data into data\_issues table too. This table contain the all the information of bad data. Let’s see the detail of bad data of stage\_customer table.



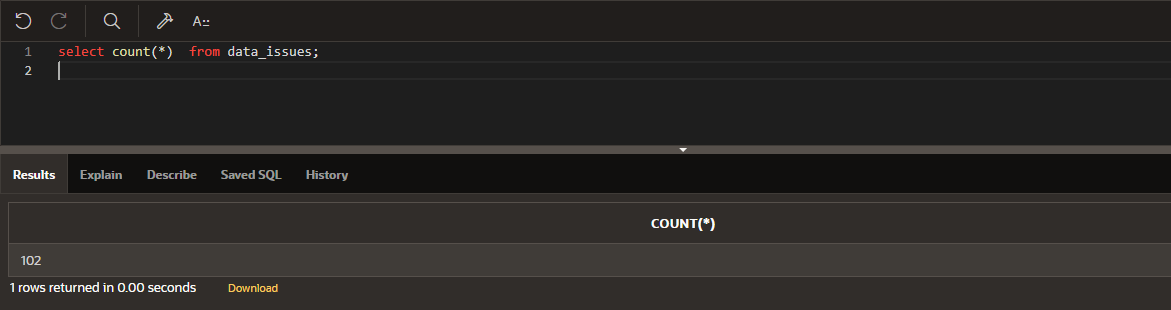
**Fig: information of bad data of customer table**

****

**Fig: total number of bad data and good data in bad\_customer and clean\_customer table**

Successfully inserted bad data into bad\_customer table and clean data into clean\_customer table.

## **Total data issue (bad data) in all tables**

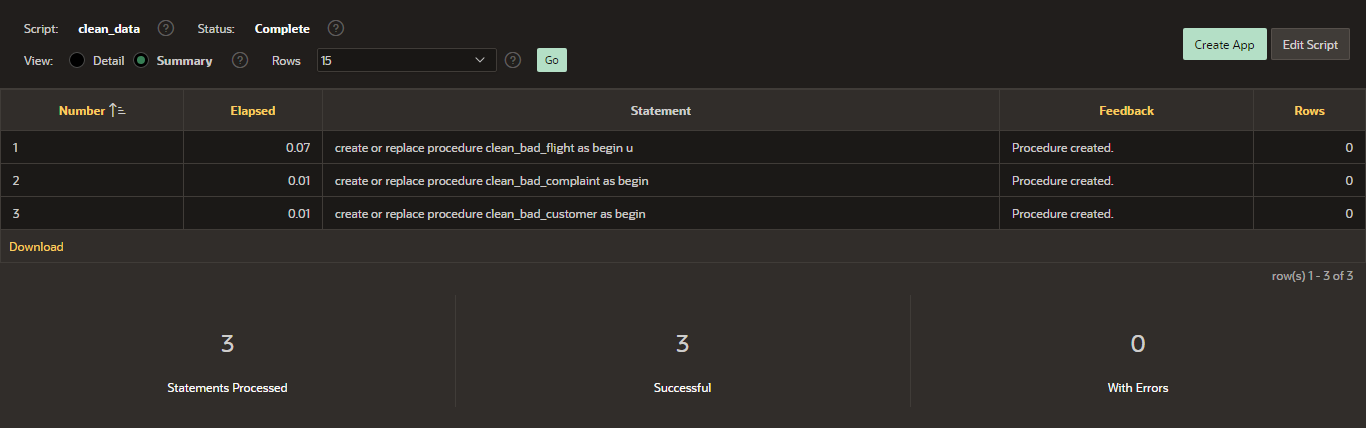
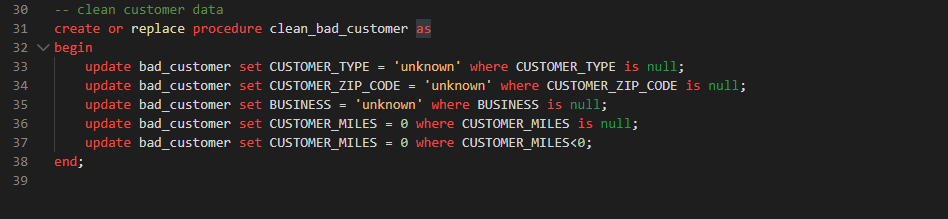
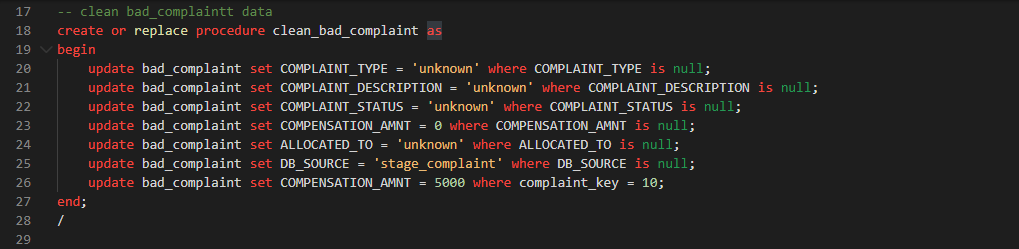
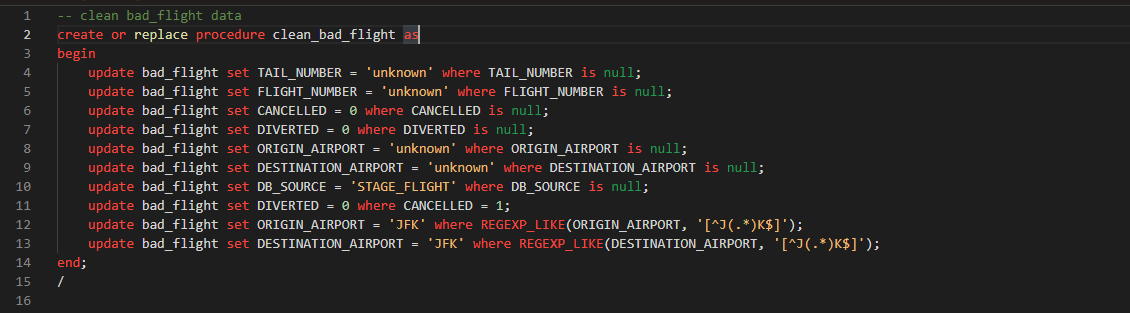


**Fig: total number of data issues**

In total we have issues with 102 data. Which means all together we have 102 bad data in all tables. These are the bad data from stage\_flight, stage\_complaint and stage\_customer tables. Now we have to fix issues of these data, which mean we will convert bad data into good data.

## **Fix bad data**

### Procedure to convert bad data

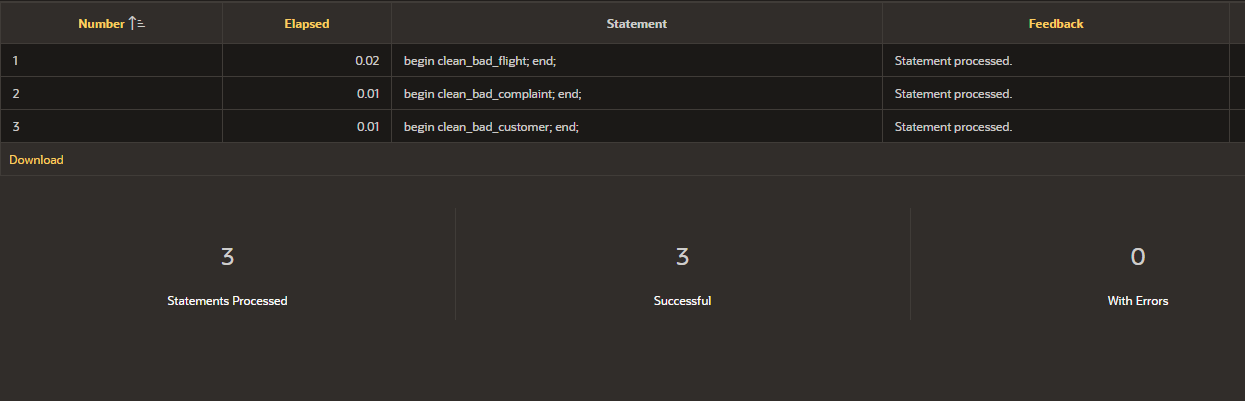
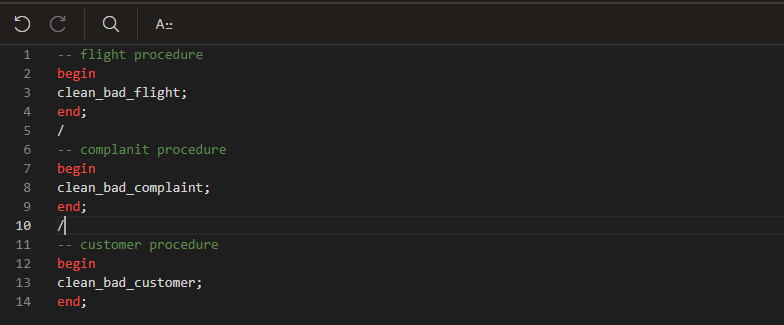


**Fig: Script to convert bad data**

To convert the bad data of flight, complaint and customer we have created three proc

Successfully created three procedure to convert the bad data of flight, customer and complaint table, clean\_flight\_data is created to convert the bad data of flight, clean\_compliant\_data to convert bad data of complaint, clean\_customer\_data to convert bad daa of customer.

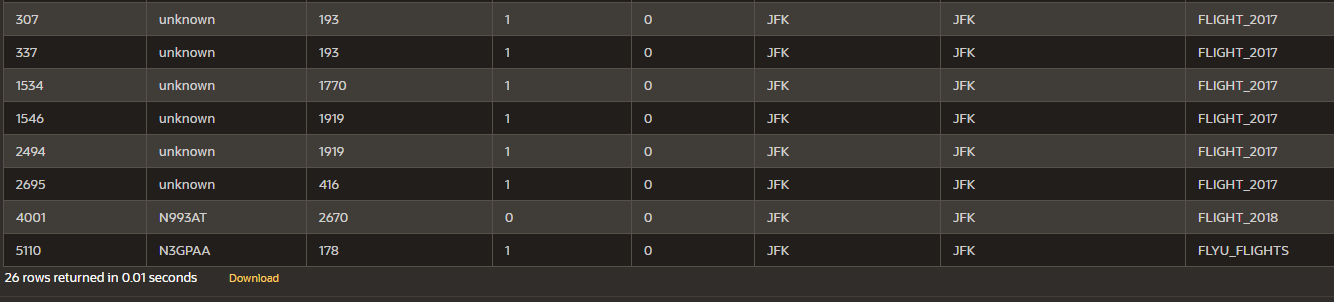
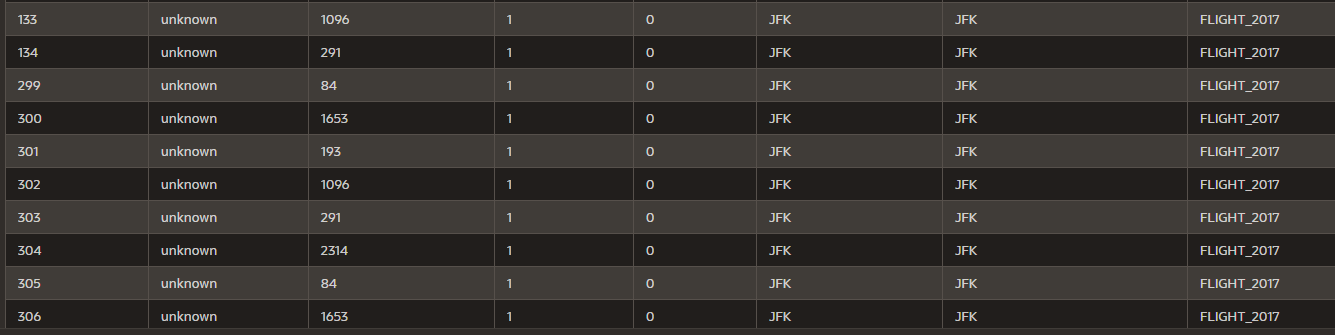
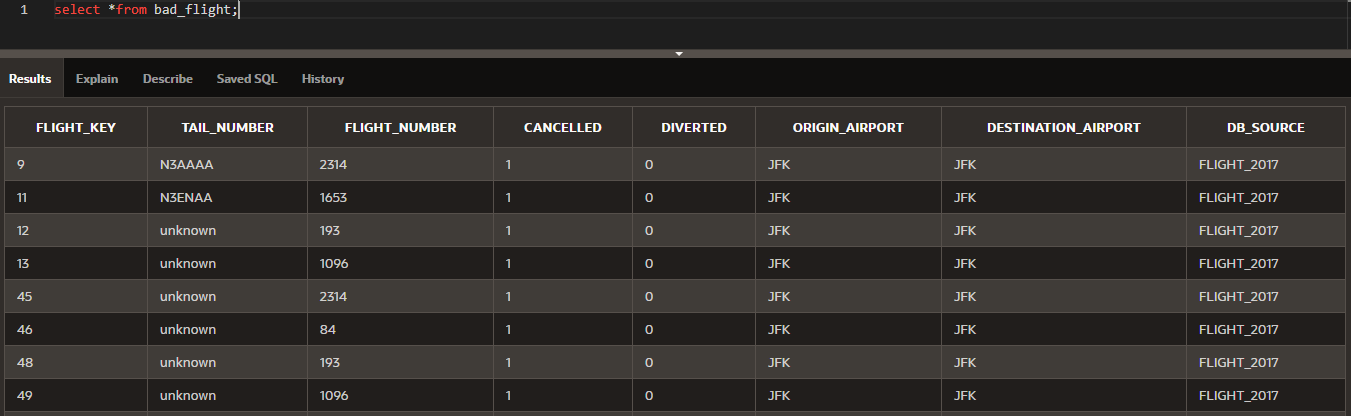
### Fire procedures

****

**Fig: fire clean\_data procedure**

Successfully fired the procedure to convert bad.

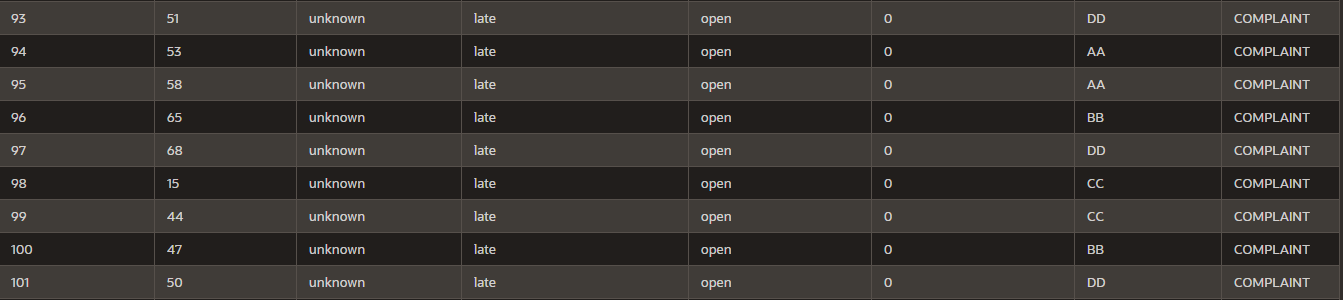
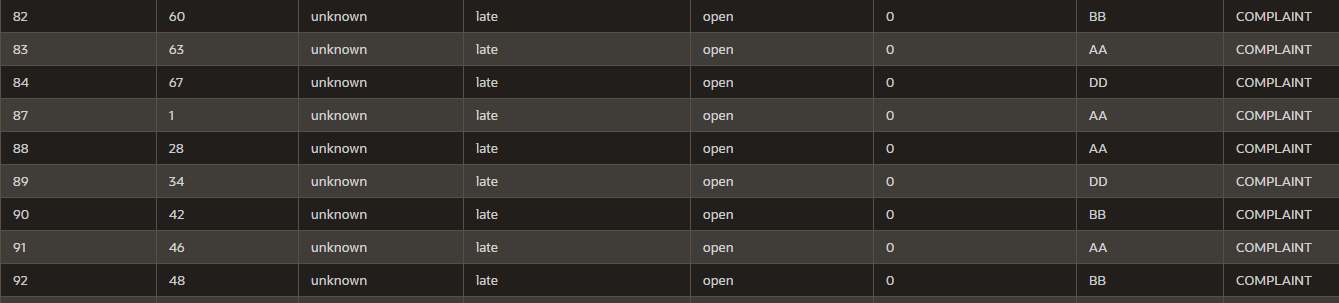
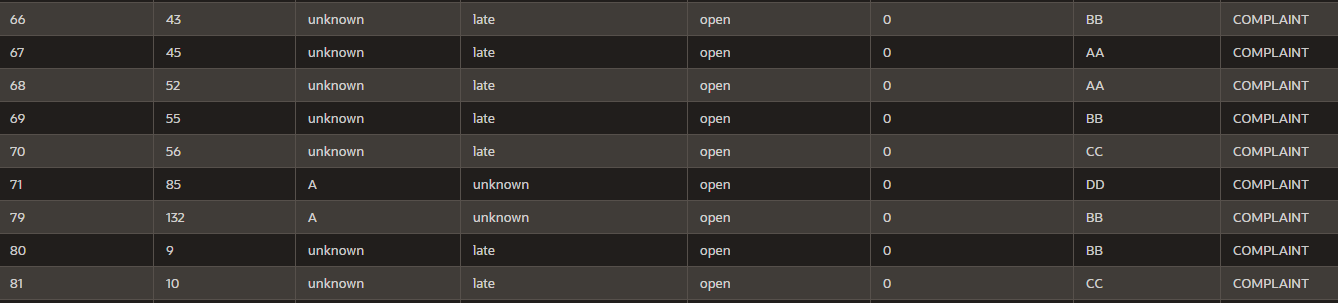
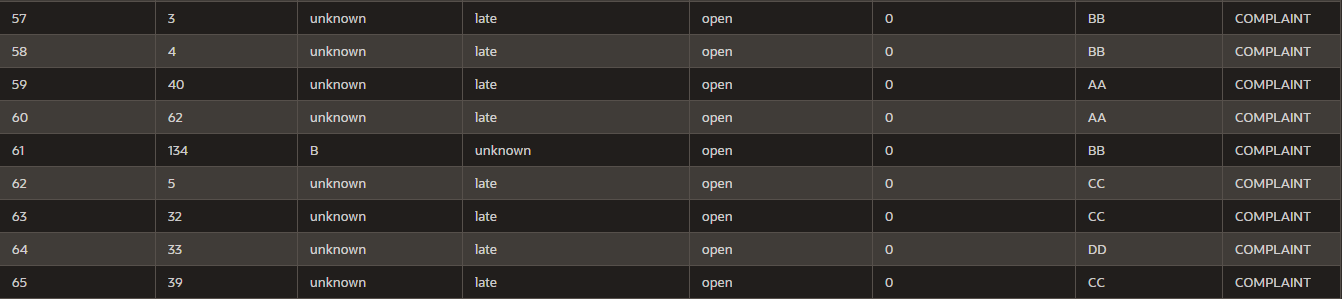
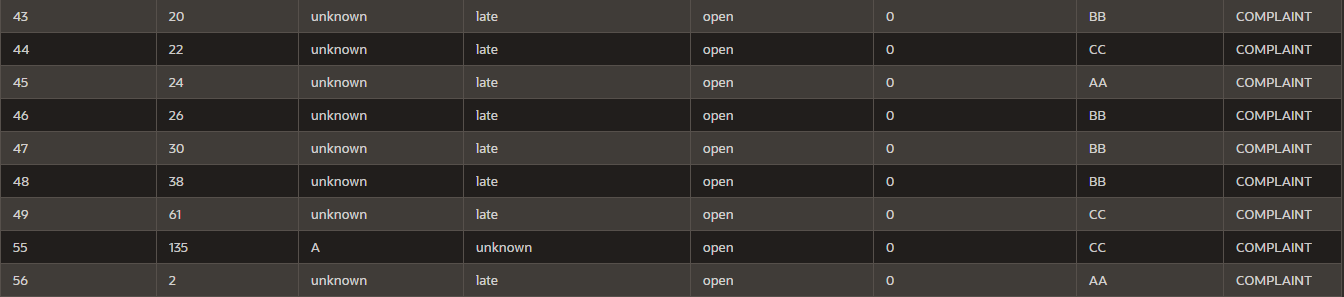
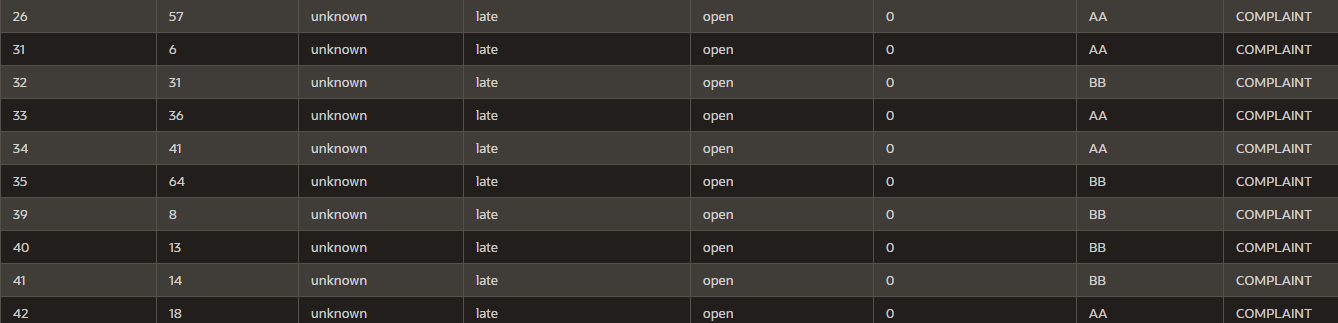
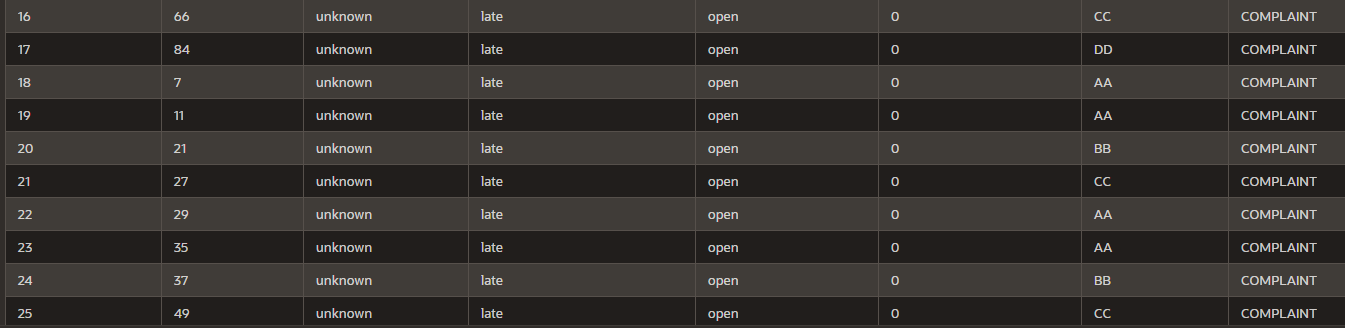
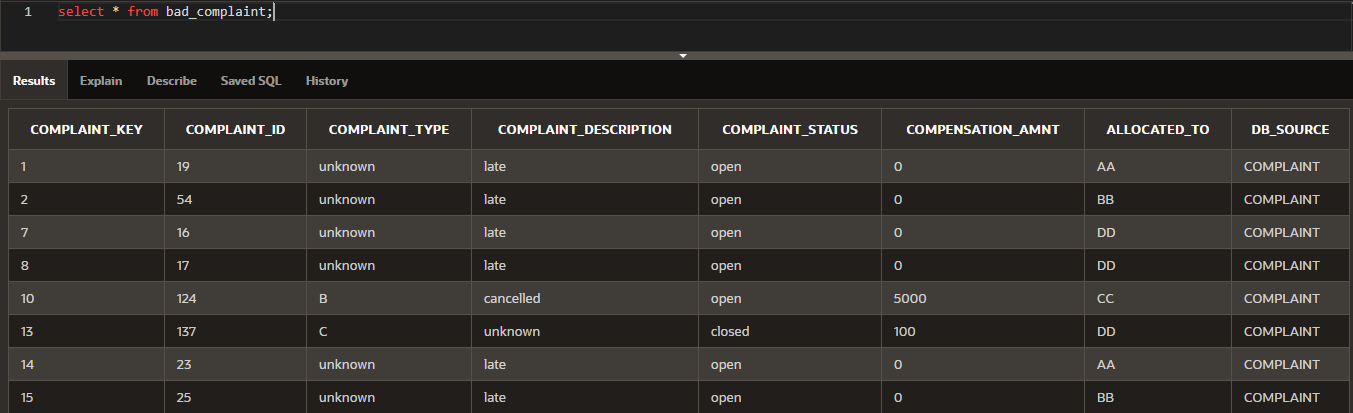
### Converted data of flight



**Fig: Converted bad data of flight**

As we see in the above, we have successfully converted bad data of flight into good data.

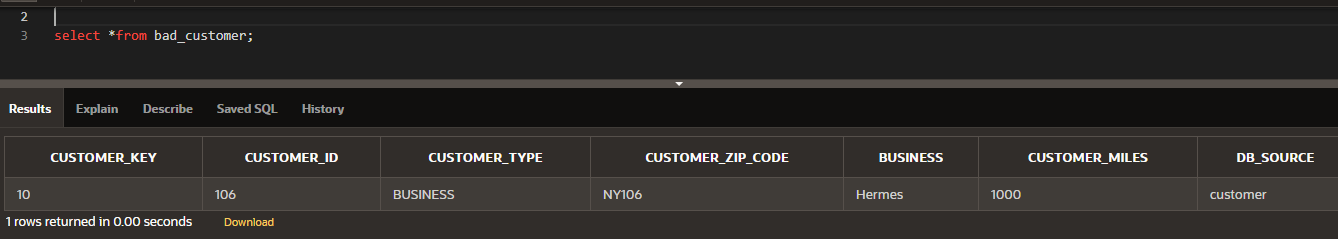
### Converted data of complaint



**Fig: Converted bad data of flight**

Successfully converted bad data into good data of complaint.

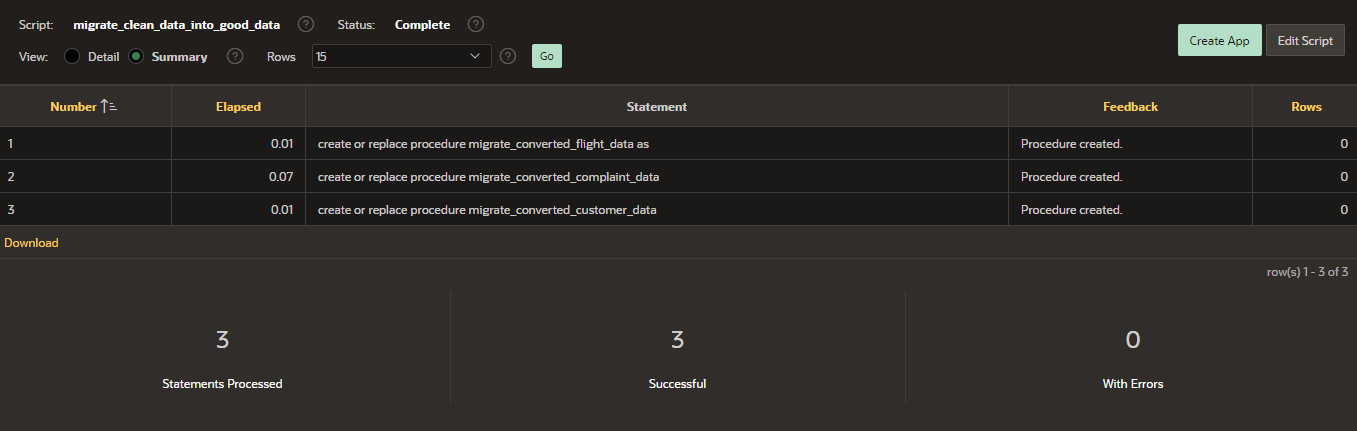
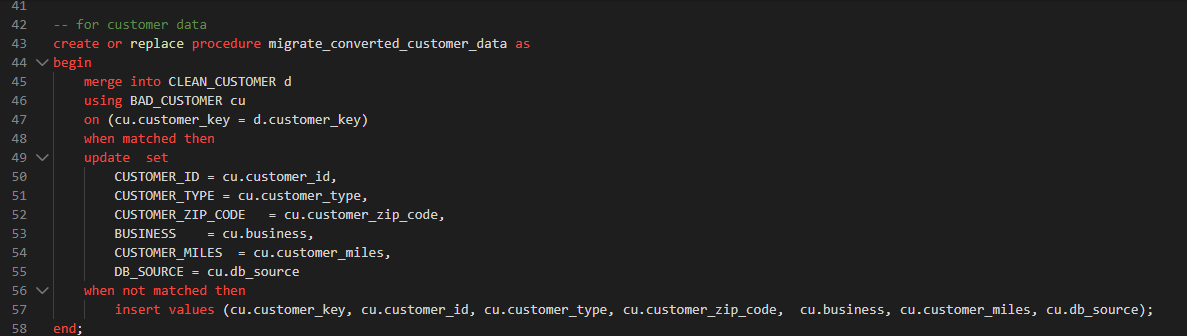
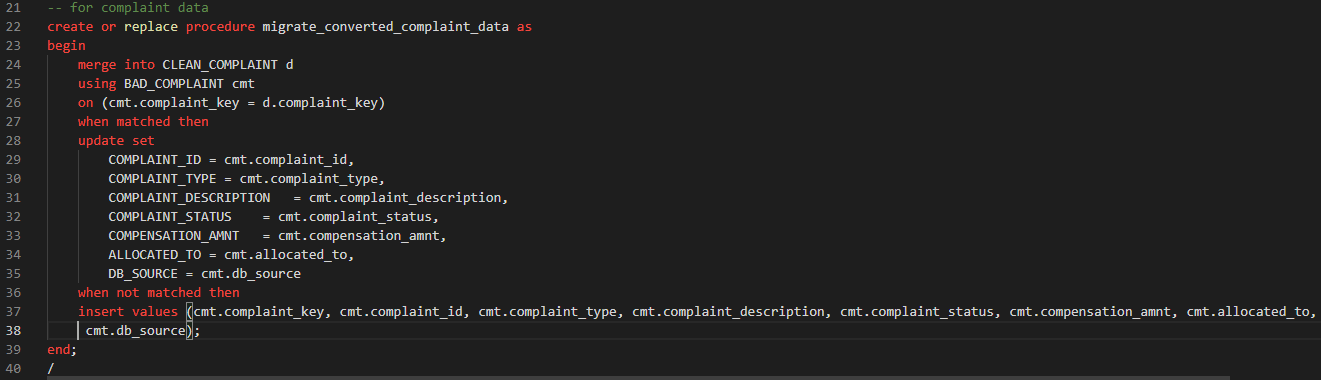
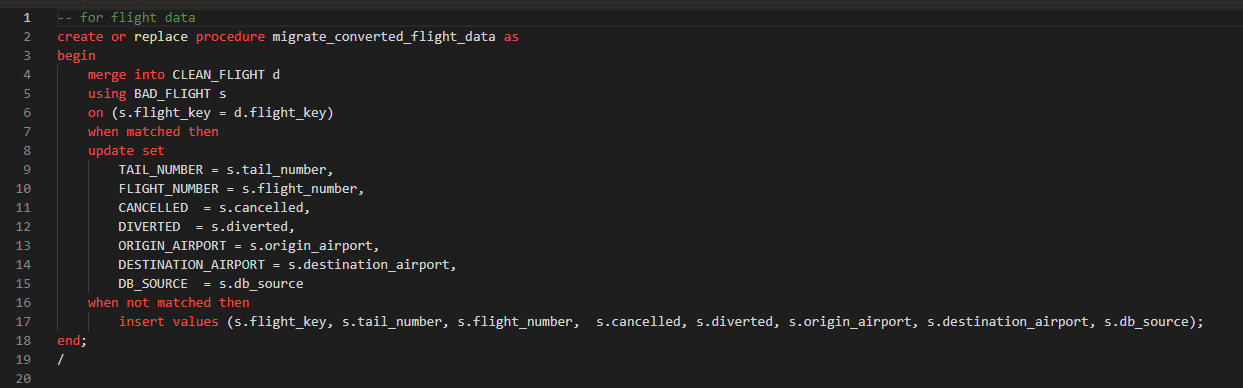
### Converted data of customer



**Fig: Converted bad data of flight**

Successfully converted bad data into good data of customer.

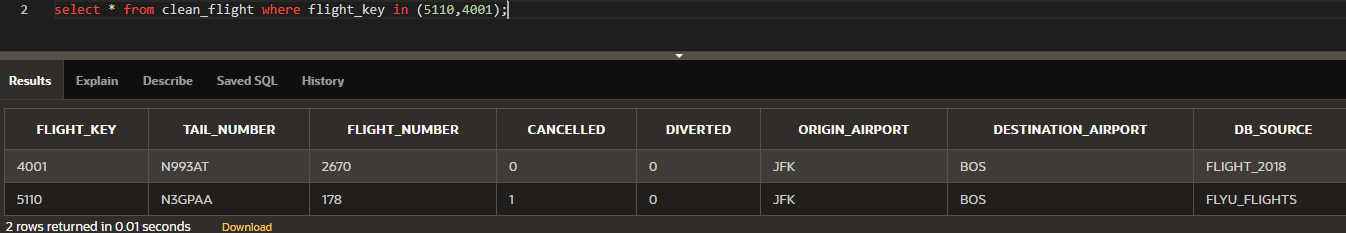
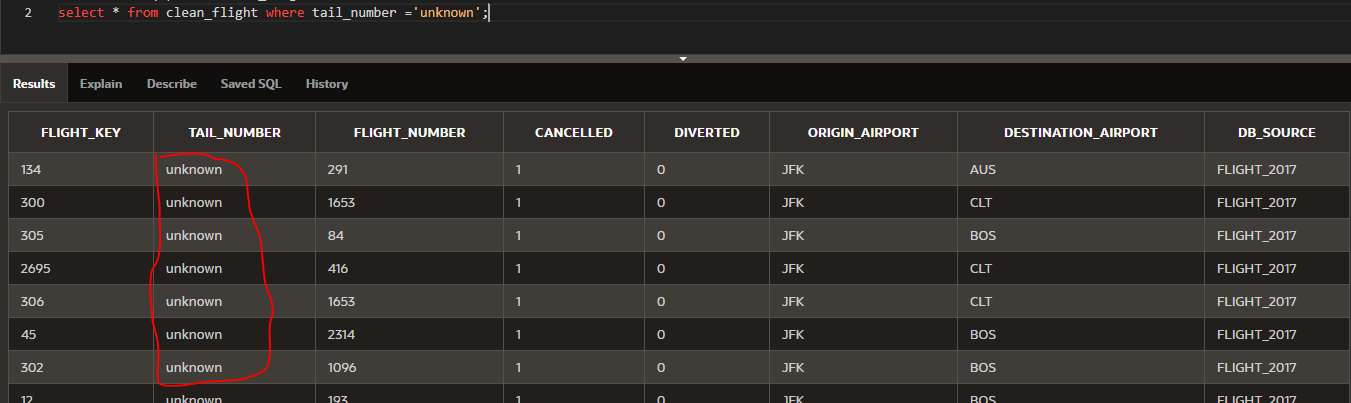
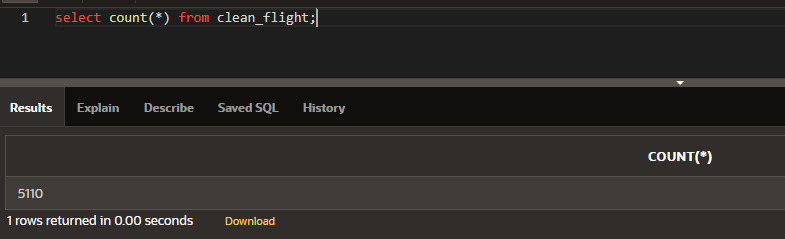
## **Migrate converted good data into clean table**



**Fig: migrate converted bad data into clean table.**

Successfully migrated the converted data into clean table.

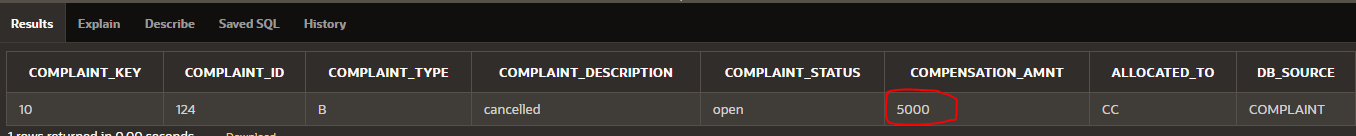
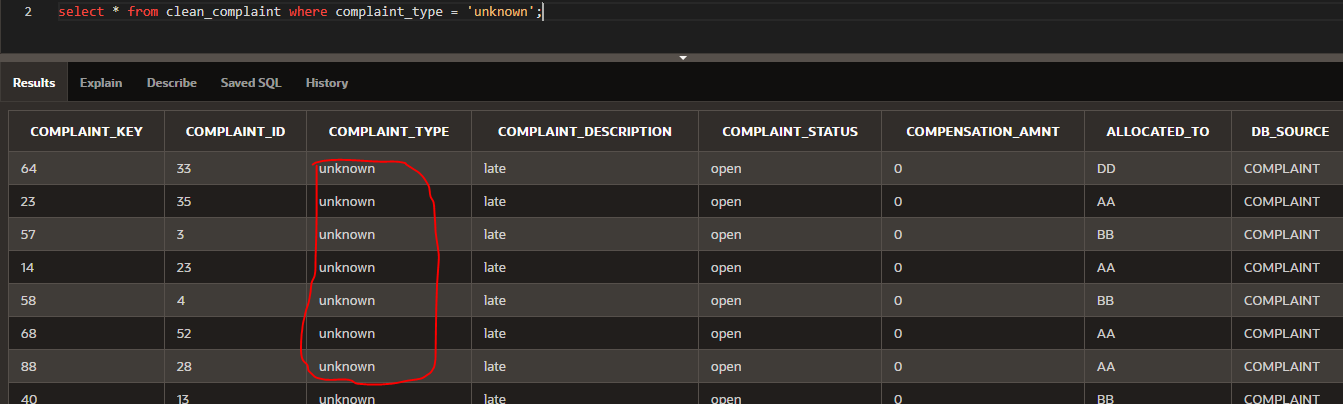
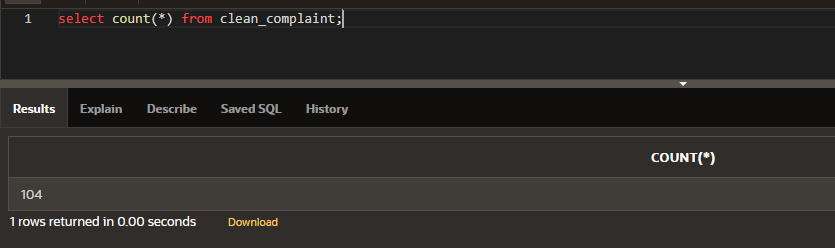
## **Clean flight bad data in clean flight table**



**Fig: clean flight bad data in clean flight table.**

As we see in the above, we have successfully converted bad data and migrate them into clean table. In clean table we have 5110 data which is equal to data is stagging table so there is not any data loss.

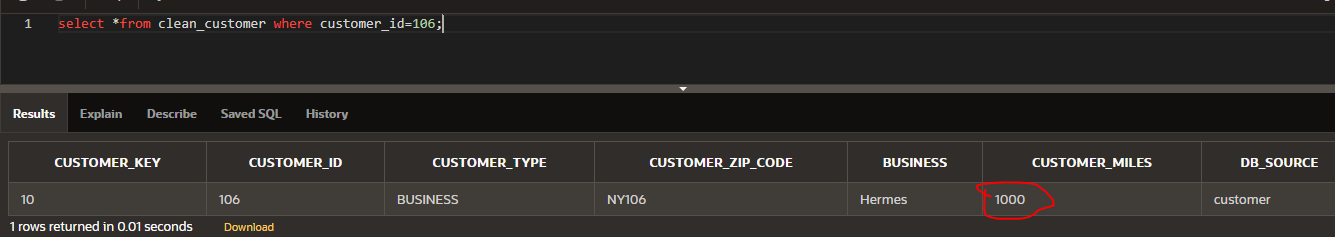
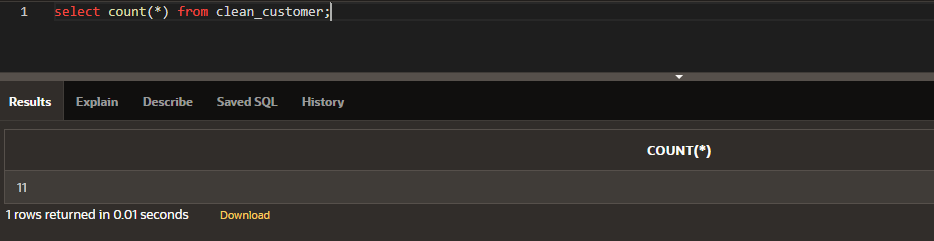
## **Clean complaint bad data in clean complaint table**



**Fig: clean complaint bad data in clean complaint table.**

As we see in the above, we have successfully converted bad data and migrate them into clean table. In clean table we have 104 data which is equal to data is stagging table so there is not any data loss.

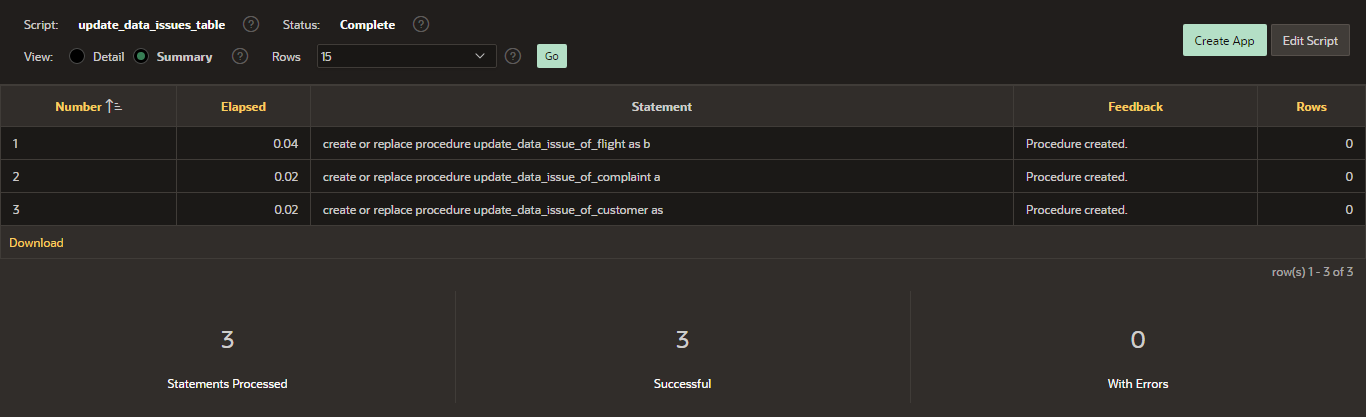
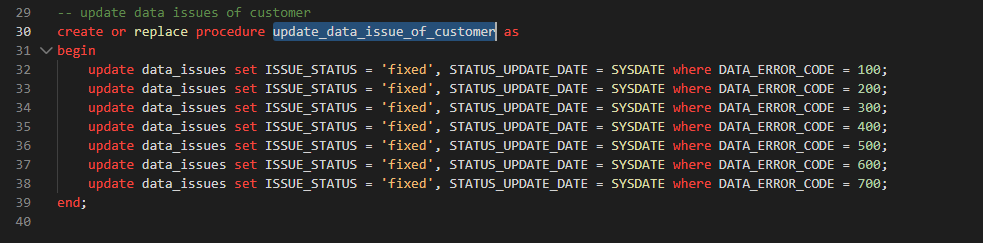
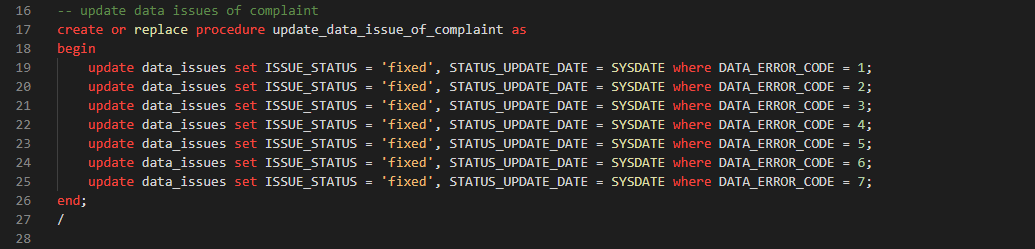
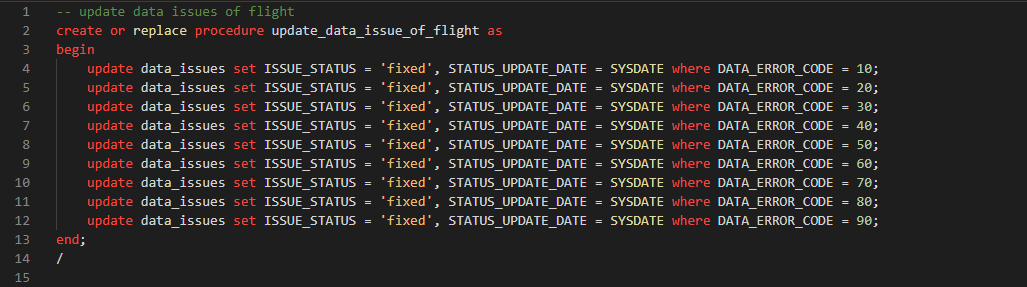
## **Clean complaint bad data in clean complaint table**



**Fig: clean customer bad data in clean customer table.**

As we see in the above, we have successfully converted bad data and migrate them into clean table. In clean table we have 11 data which is equal to data is stagging table so there is not any data loss.

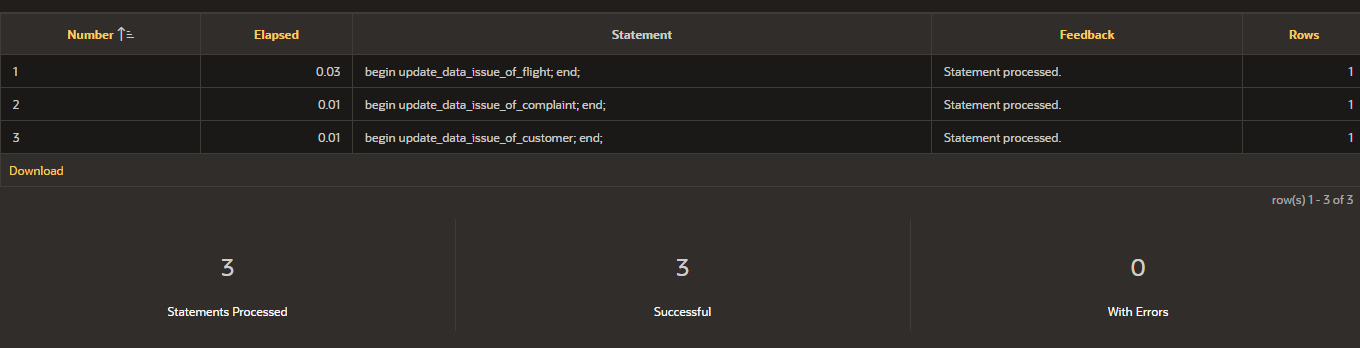
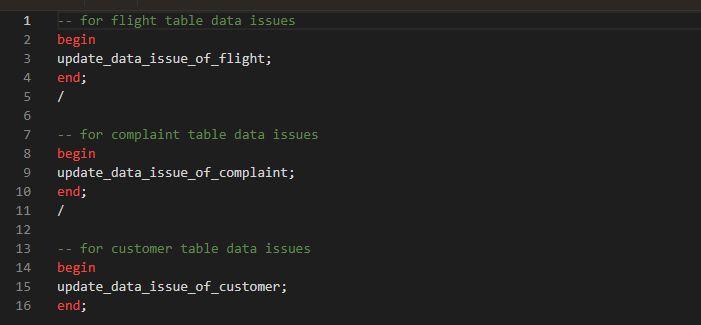
## **Update data\_issues table**



**Fig: script to create procedure to update data issues table.**

Successfully created procedures to update the information of data in data issues table. Procedures are update\_data\_issues\_of\_customer, update\_dat\_issues\_of\_complaint, update\_data\_issues\_of\_flight.

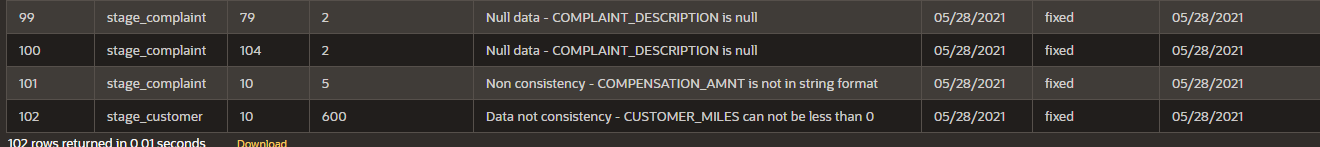
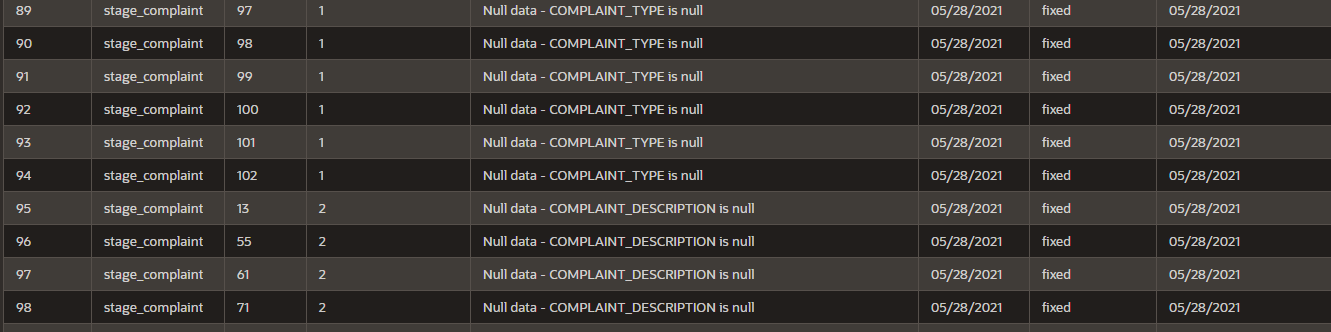
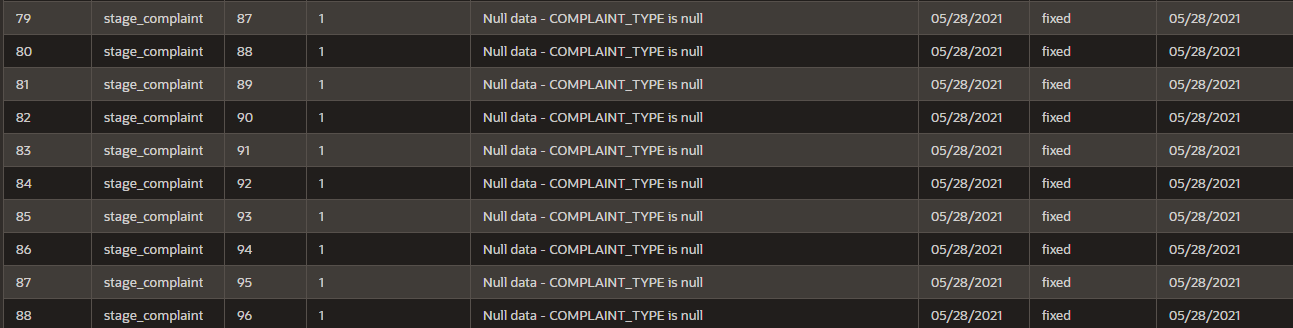
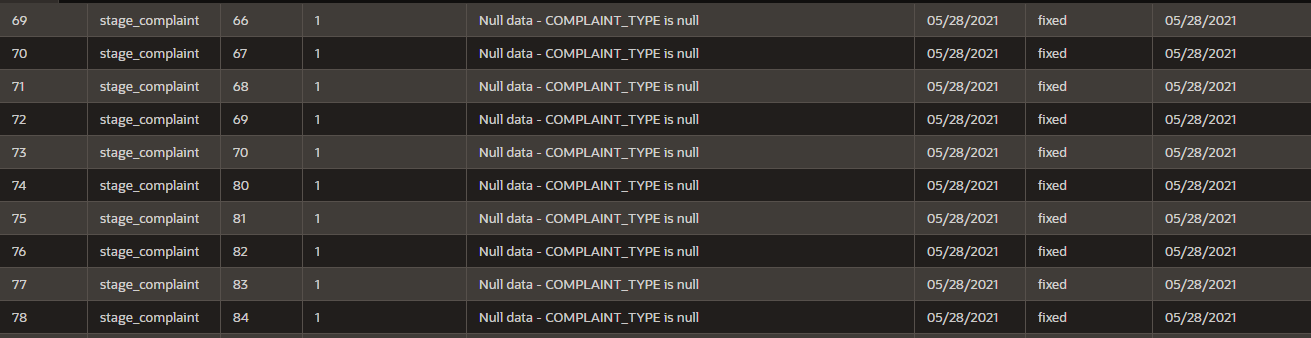
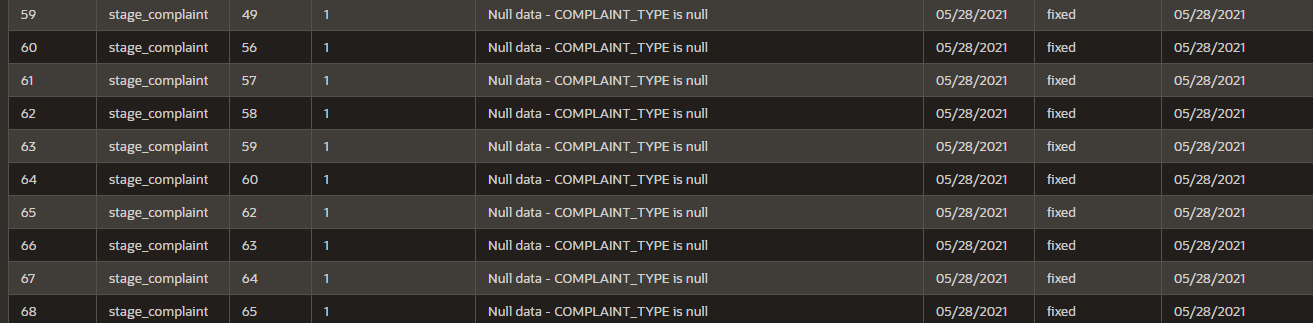
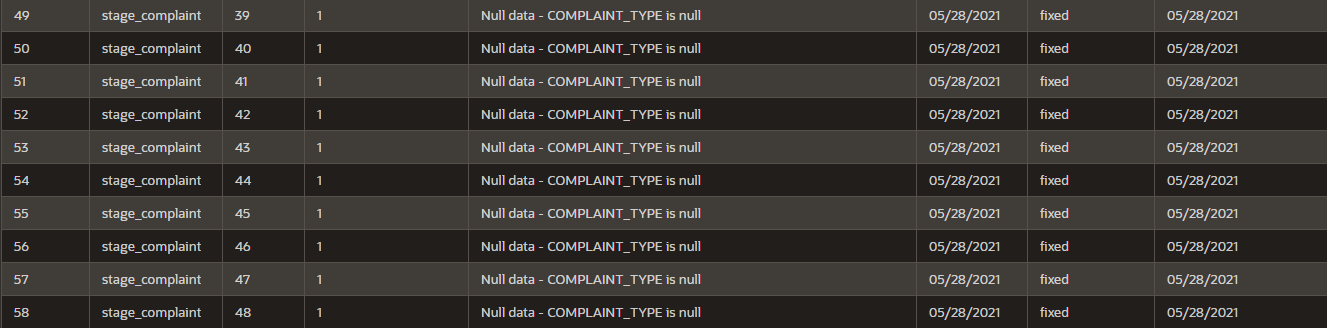
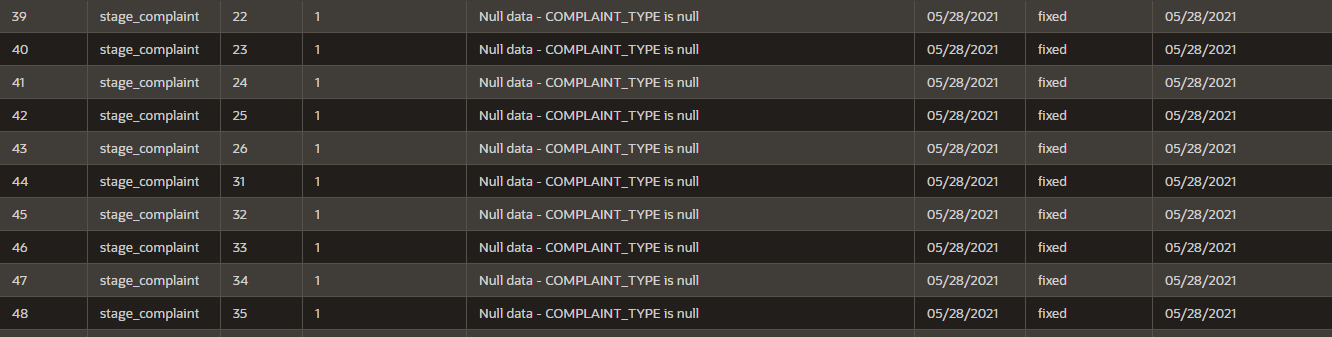
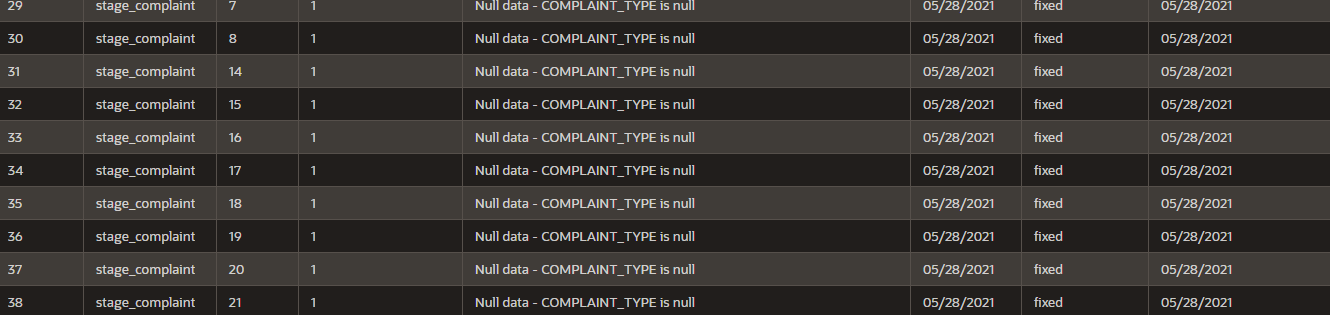
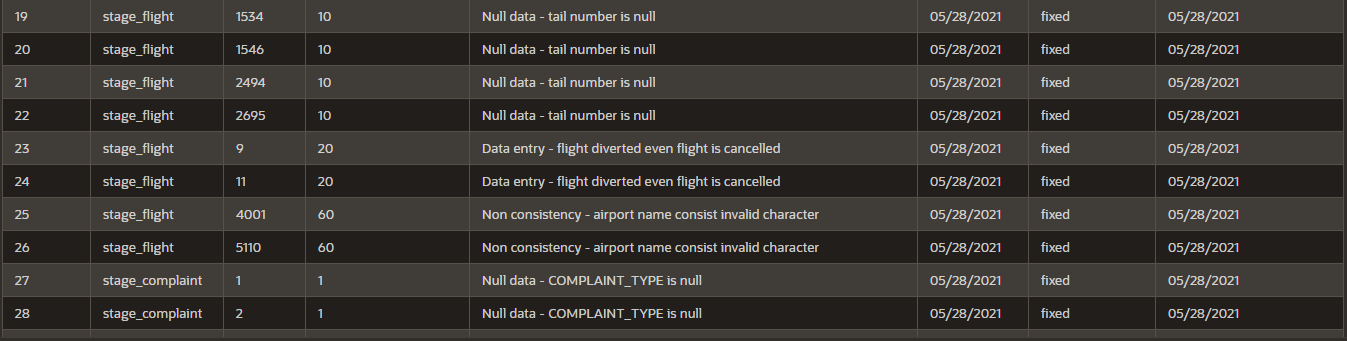
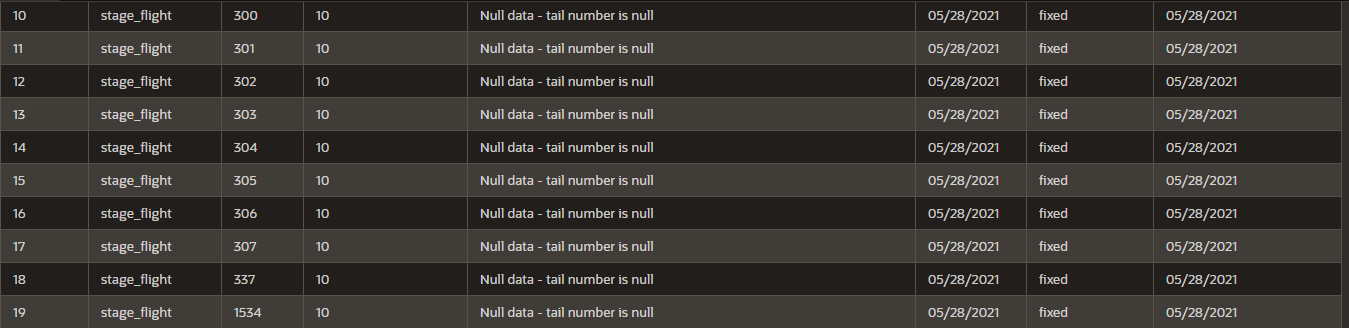
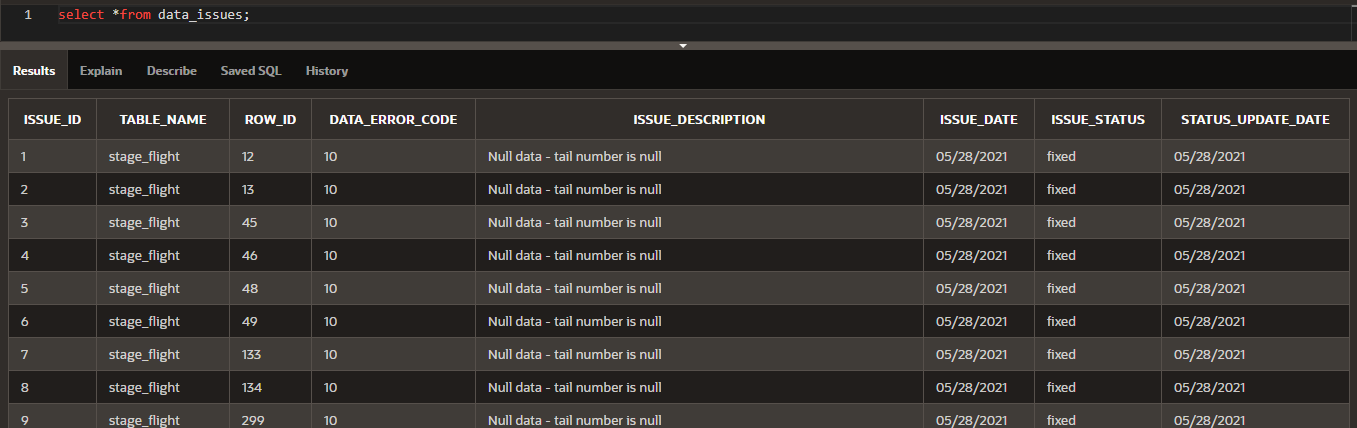
### Fire procedure of data\_issues table



**Fig: script to create procedure to update data issues table.**

Successfully fired procedures to update the information of data in data issues table. Fired procedures are update\_data\_issues\_of\_customer, update\_dat\_issues\_of\_complaint, update\_data\_issues\_of\_flight

### Data\_issues table after update



**Fig: Data issues fixed.**

Successfully updated the bad data information is data\_issues data table. So in future if anyone wants to know the error description, when it was fixed then they can checkout this table.