**Business Requirement Document: Medico Pvt. Ltd.**

****

**Medico Private limited**

**February 2022**

**1. Introduction**

**1.1 Project Summary**

**1.1.1 Objectives**

* Enhancing client’s business of Medical devices by loading the data into Snowflake database.
* Various department members will be able to see the progress & tracking of their performance from these dashboards from time to time and make changes in their processes accordingly.
* Business owners will be able to make decisions based on data backed opinions to deliver the required growth by optimizing business in areas where cost is high and return is low.

**1.1.2 Background**

A central data repository is necessary for simplifying the data analytical process for the client. Clients should be able to search through all the data and retrieve it as and when required. Data analytical solutions that will be provided to the client will help in making the right decisions for the business and understand how to grow the business by improving it in different areas.

**1.1.3 Business Drivers**

* Client is looking for having all the data in one place.
* Various internal teams within the client organization are going to be the end users of the dashboards.
* Client is responsible for managing all the data from various sources and dumping them at one place for further processing.

**1.2 Project Scope**

The scope of the project is to transform and load ERP, CRM, Feedback, Userlogin data for marketing leads, pipeline and opportunity processes. All the data will be used to create the dashboard views for interpreting the data and therefore make the changes in the organization accordingly. This transformed data is also used to perform machine learning algorithms to analyze the data and gather insights.

**1.2.1 Scope Functionality**

⦁ Security, Governance, and Data Protection

⦁ Standard and Extended SQL Support

⦁ Connectivity

⦁ Data Import and Export

⦁ Data Sharing

**1.3 System Perspective**

**1.3.1 Assumptions**

⦁ Data is accurate

⦁ Data is historical

⦁ Administrator will manage the data access control

⦁ Erroneous records are not considered for calculations

⦁ Price of the products will be in USD.

**1.3.2 Constraints**

⦁ Snowflake Training required for proper utilization

⦁ Data stored as PDF cannot be loaded

**2. Business Requirements**

The requirements in this project are prioritized as follows:

| **Value** | **Rating** | **Description** |
| --- | --- | --- |
| 1 | Critical | Most Urgent and most important requirements |
| 2 | High | Urgent but not as critical requirements and important requirements |
| 3 | Medium | Urgent but not as critical requirements and important requirements |
| 4 | Low | Low Important and not urgent requirements |
| 5 | Future | Future scope requirements. |

**2.1 Functional Requirements**

| Task Id | Priority | Task |
| --- | --- | --- |
| SF1 | 1 | Data Ingestion using Fivetran |
| SF2 | 2 | Data loading using Snowpipe |
| SF3 | 2 | DBT transformations |
| SF4 | 2 | Snowflake Development |
| SF5 | 1 | Views and table |
| SF6 | 3 | Data Sampling & Secure Data Sharing |
| SF7 | 4 | Query optimization, performance monitoring & Metadata |
| SF8 | 1 | Security in Snowflake |
| SF9 | 4 | Data Unloading |
| SF10 | 2 | TESTING |

**2.2. Non-Functional Requirements**

* The data must be either structured or semi structured
* The data file should not be in PDF format
* The data must be accurate

**3. Architecture Overview**

⦁ Client gets all their data in one place

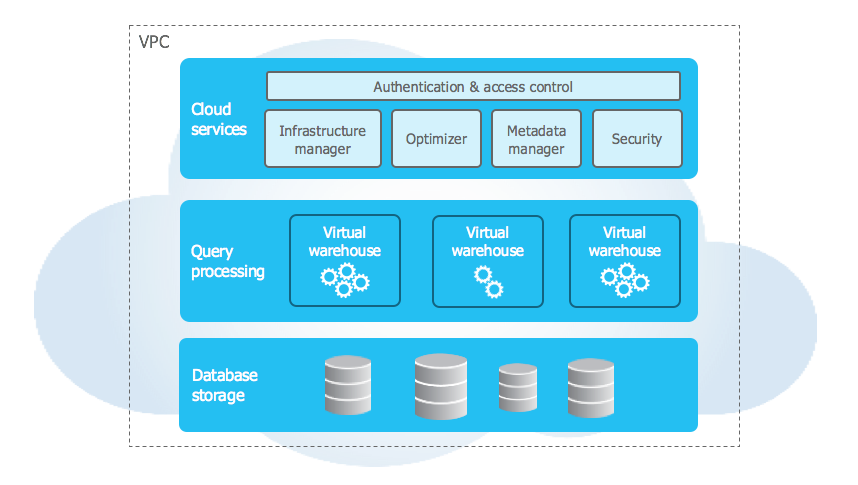
⦁ Data secured by top of line encryption

⦁ View product performance

⦁ Fail safe for data and time travel

**3.1 Architecture**

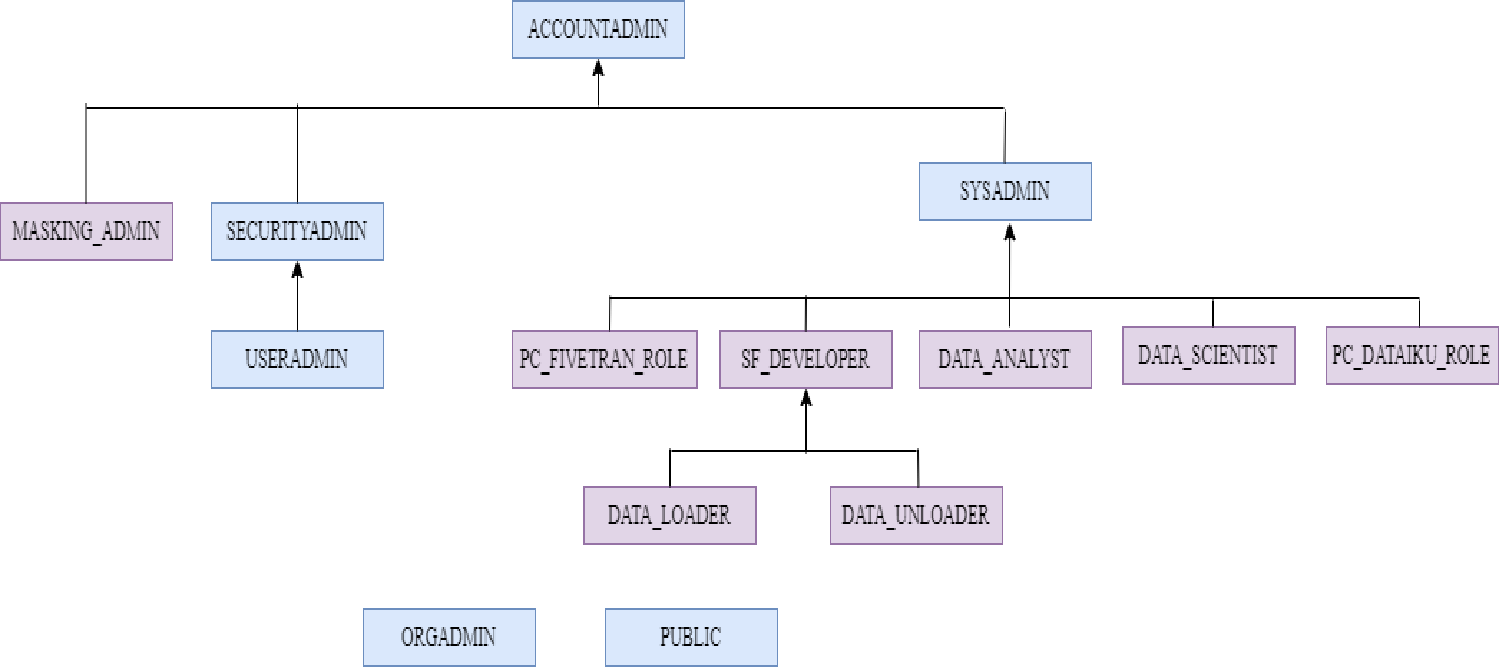
Snowflake’s architecture is a hybrid of traditional shared-disk and shared-nothing database architectures. This approach offers the data management simplicity of a shared-disk architecture, but with the performance and scale-out benefits of a shared-nothing architecture.



**3.2. Role Based Access Diagram**

Snowflake provides granular control over access to objects — who can access what objects, what operations can be performed on those objects, and who can create or alter access control policies.

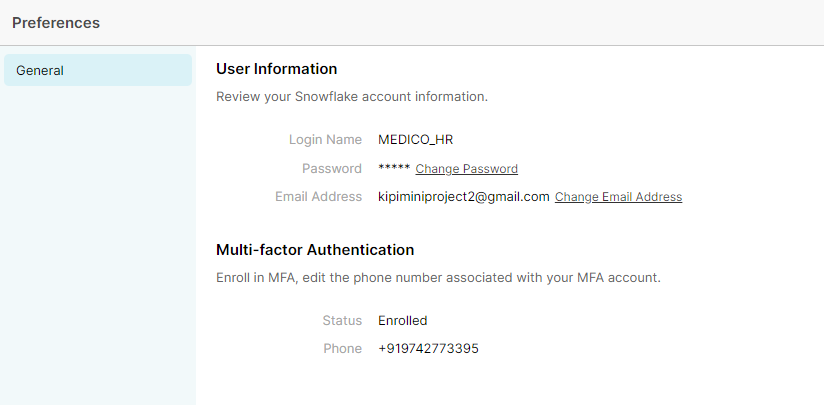
Roles are the entities to which privileges on securable objects can be granted and revoked. Roles are assigned to users to allow them to perform actions required for business functions in their organization



**3.2. Security in Snowflake**

Snowflake supports multi-factor authentication (i.e. MFA) to provide increased login security for users connecting to Snowflake. MFA support is provided as an integrated Snowflake feature, powered by the [Duo Security](http://www.duosecurity.com/) service, which is managed completely by Snowflake.

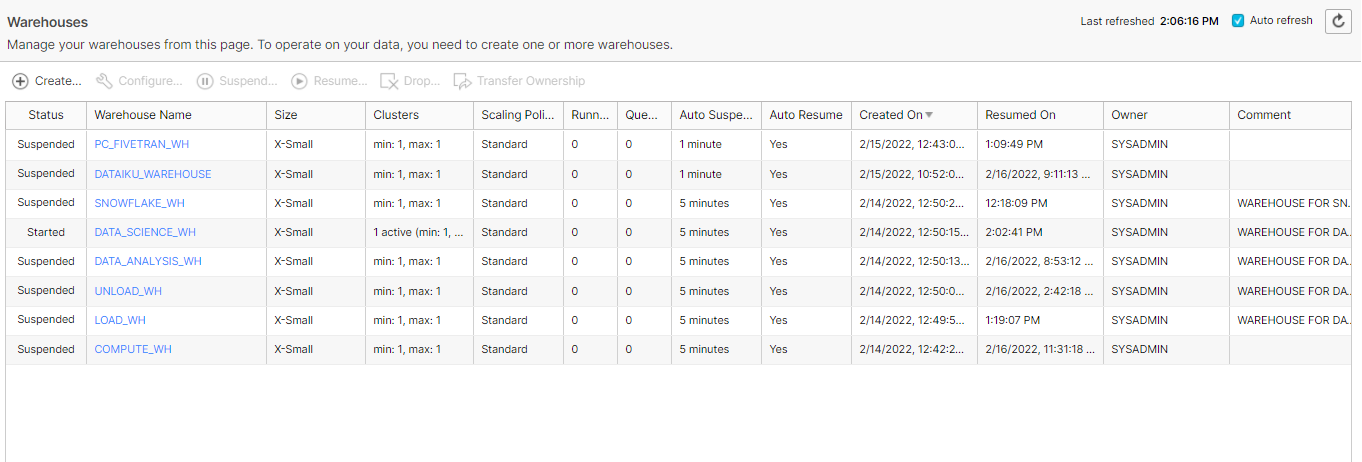
Users do not need to separately sign up with Duo or perform any tasks, other than installing the Duo Mobile application, which is supported on multiple smartphone platforms. MFA is enabled on a per-user basis; however, at this time, users are not automatically enrolled in MFA. To use MFA, users must enroll themselves.



**4. Implementation**

**4.1. Warehouses**

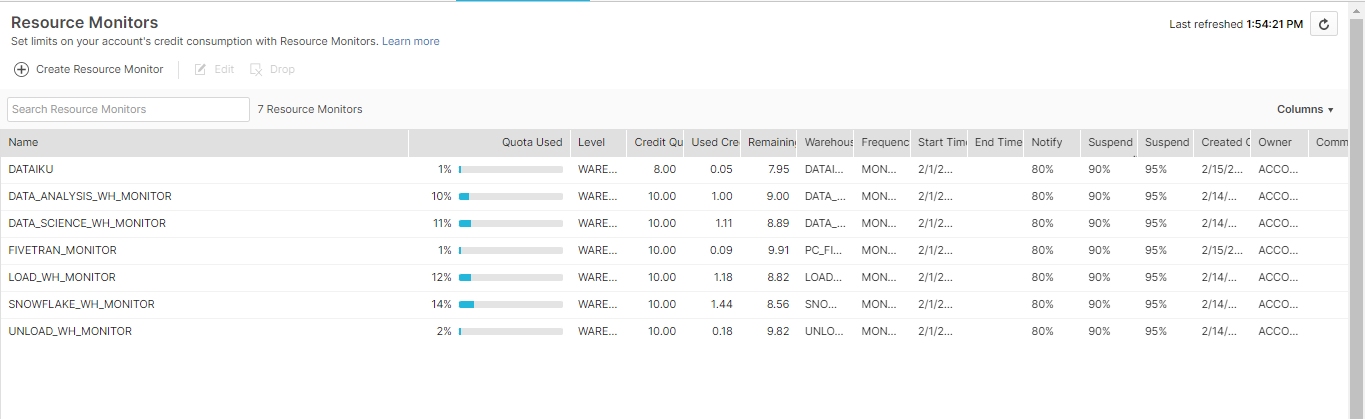
A virtual warehouse, often referred to simply as a “warehouse”, is a cluster of compute resources in Snowflake. A warehouse provides the required resources, such as CPU, memory, and temporary storage, to perform operations in a Snowflake session



**4.2. Resource monitor**

To help control costs and avoid unexpected credit usage caused by running warehouses, Snowflake provides resource monitors. Resource monitors can be used to impose limits on the number of credits that are consumed by:

* User-managed virtual warehouses
* Virtual warehouses used by cloud services



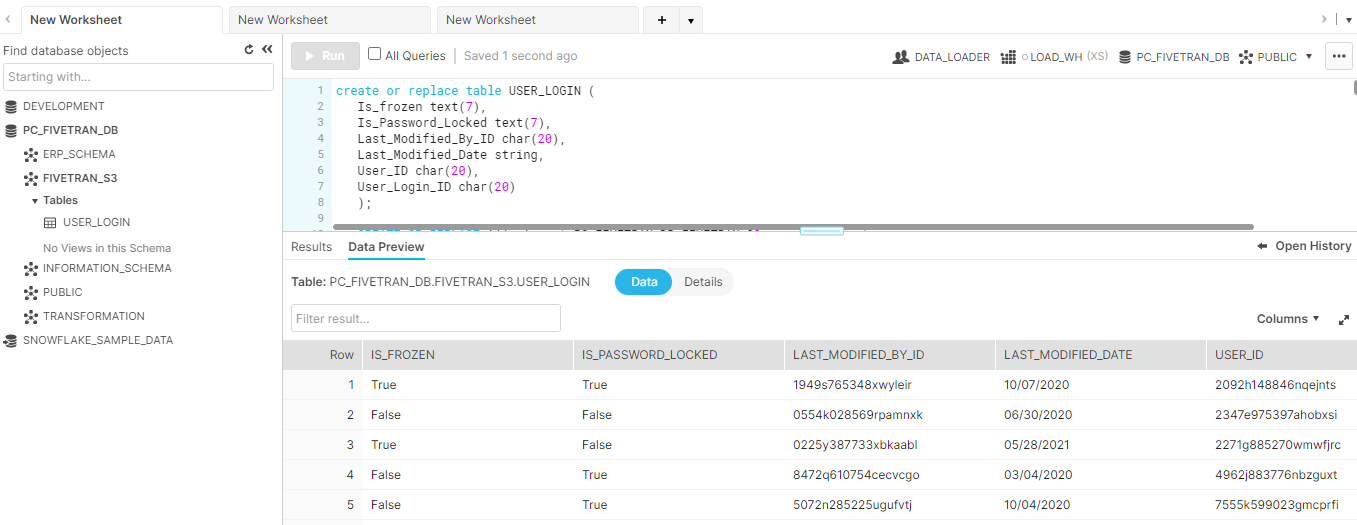
**4.3. Data loading**

1. **Fivetran**

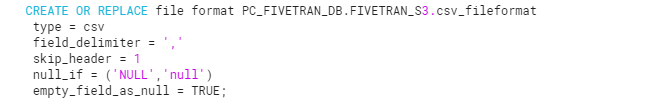
We are using Fivetran to ingest data from data lake to snowflake instance. The Fivetran BRD document will cover this topic in detail.

1. **Snowpipe:**

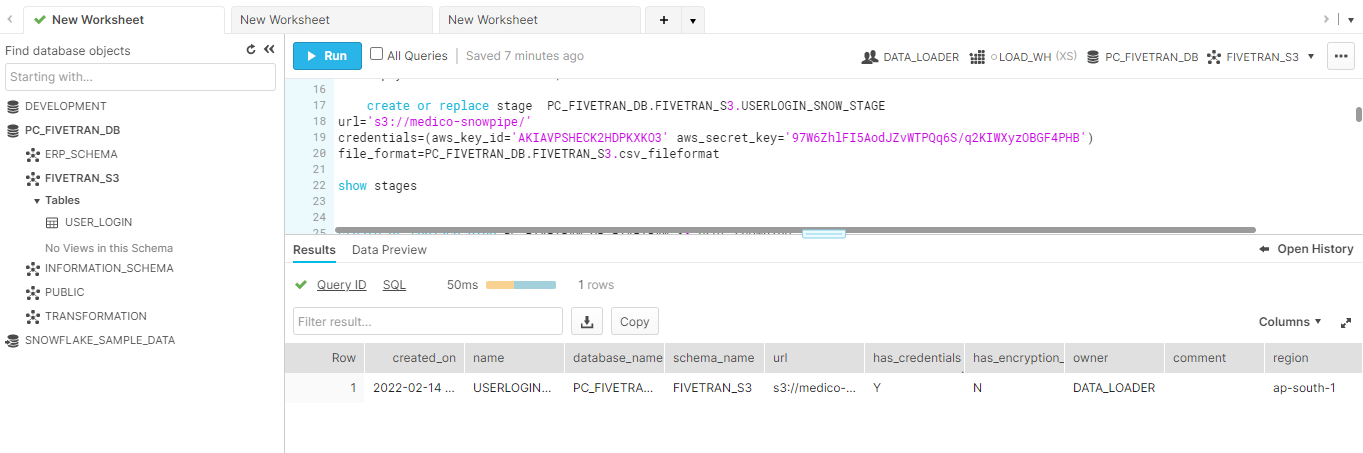
* Here we create a table user login for loading of data through snowpipe



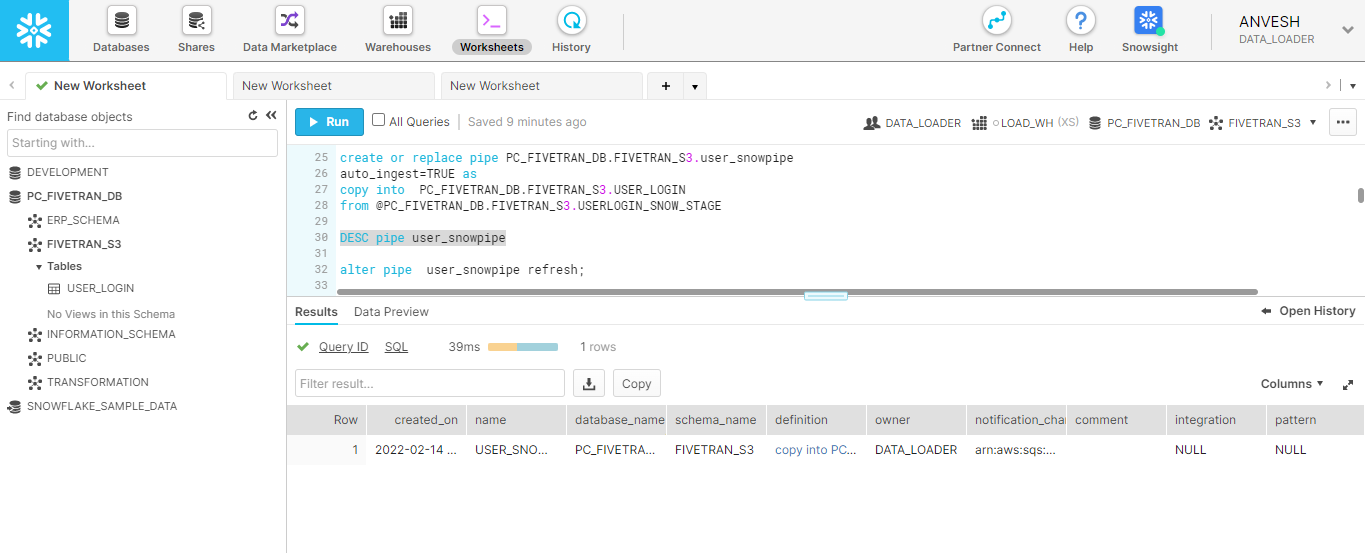
* Then we use a default file format



* Then we create a stage as snowpipe load data when it arrive in its staging area



* Then we create a snowpipe which copy data from the stage and insert it into user login table



1. **Semi structured data loading :**

We have loaded the json data from aws s3 bucket to our data lake using copy into command. We have parsed the json data in the copy command without storing it in the variant data column, as variant data type can hold only 16 mb of compressed data.



**4.4. DBT for data transformation**

DBT core is used for performing transformation. The destination for DBT transformed tables and views is TRANSFORMATION schema in Data Lake PC\_FIVETRAN\_DB.

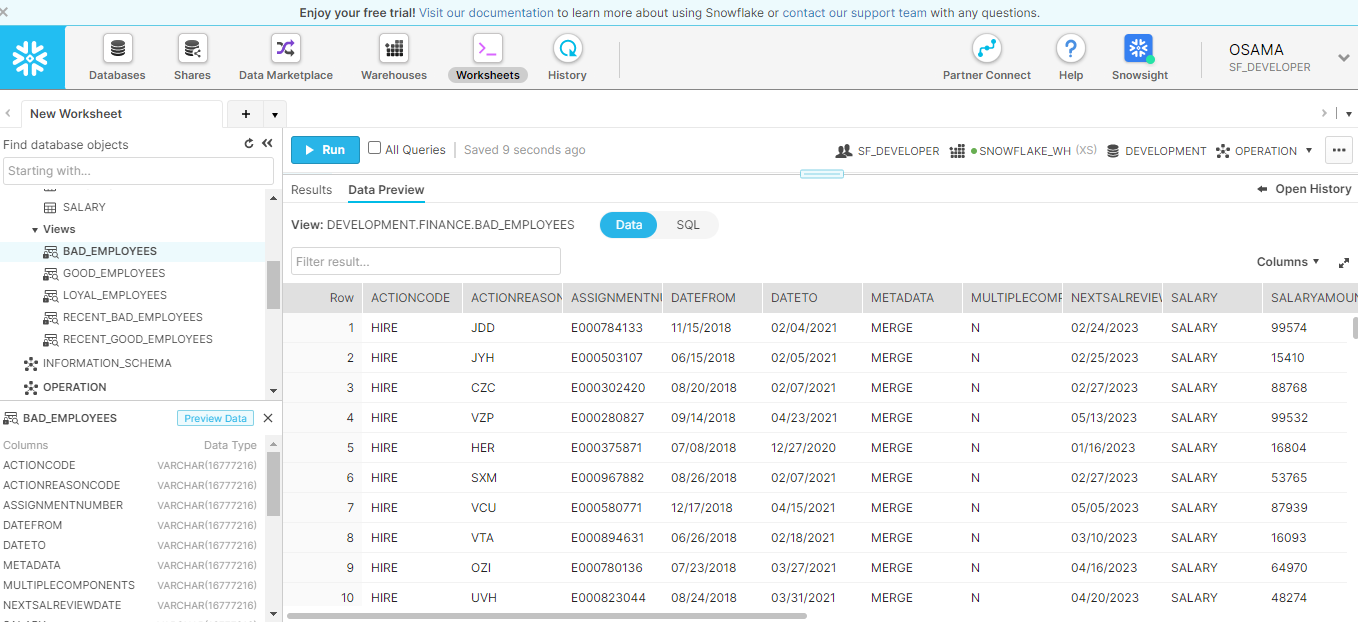
**4.5. Views and table**

A view allows the result of a query to be accessed as if it were a table. The query is specified in the Standard View.

In our snowflake instance we are using

**Secure views**

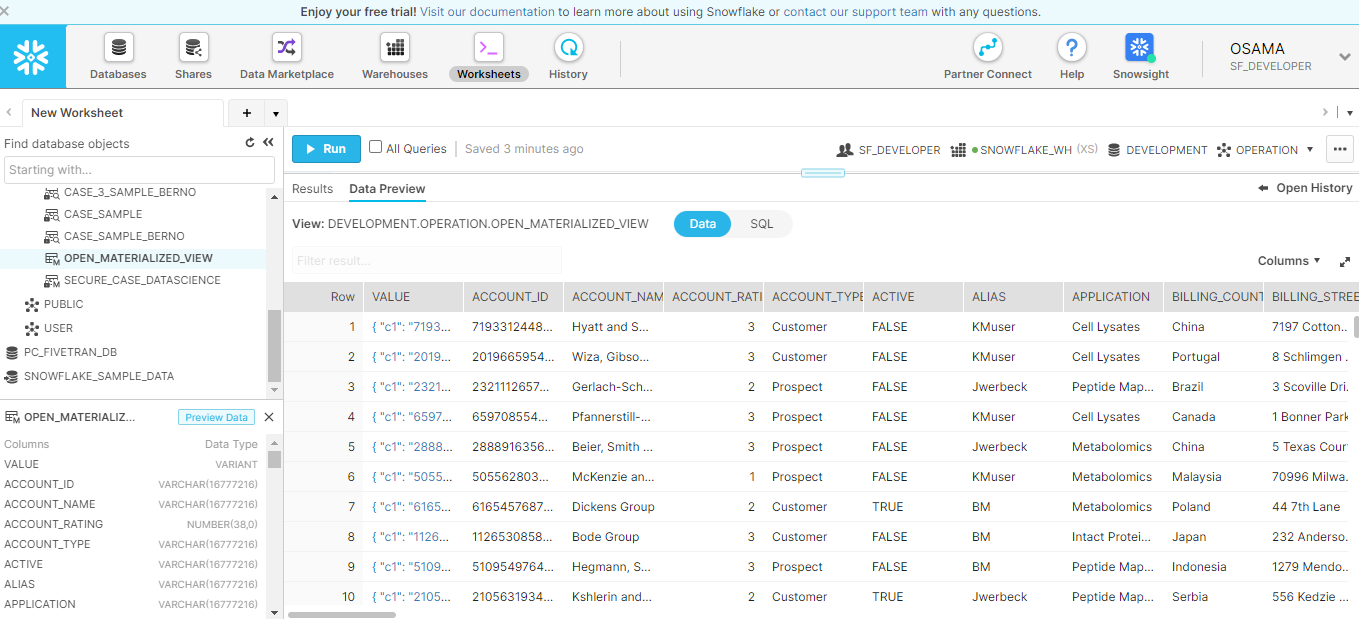
For security or privacy reasons, we wish not to expose the underlying tables or internal structural details for a view. With secure views, the view definition and details are only visible to authorized users



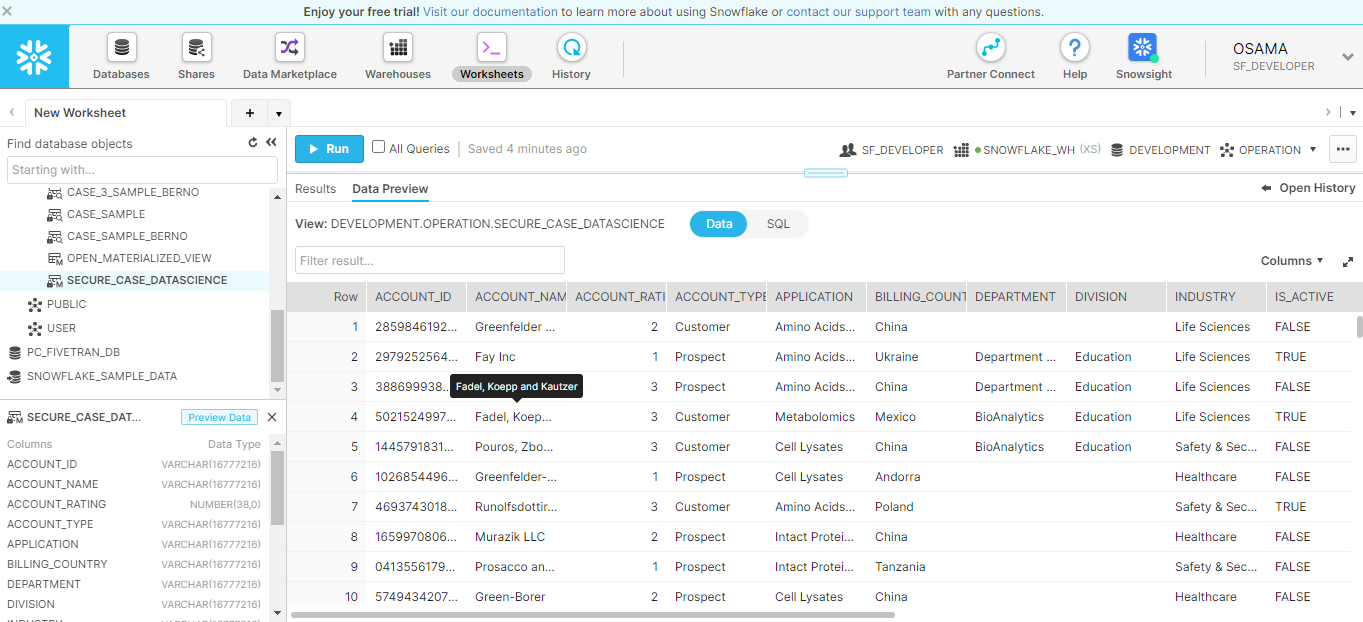
**Materialized views**

A materialized view’s results are stored, almost as though the results were a table. This allows faster access, but requires storage space and active maintenance, both of which incur additional costs.

We have used materialized view for faster access and to provide cost savings

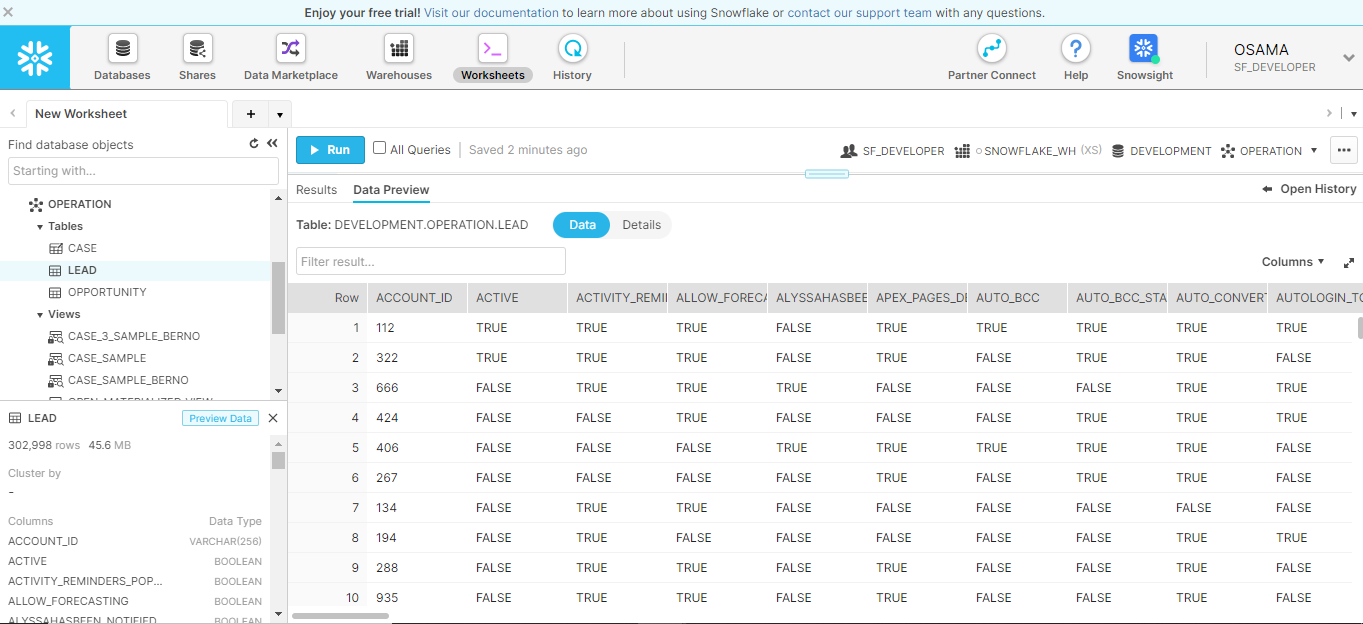


**Secure Materialized views**



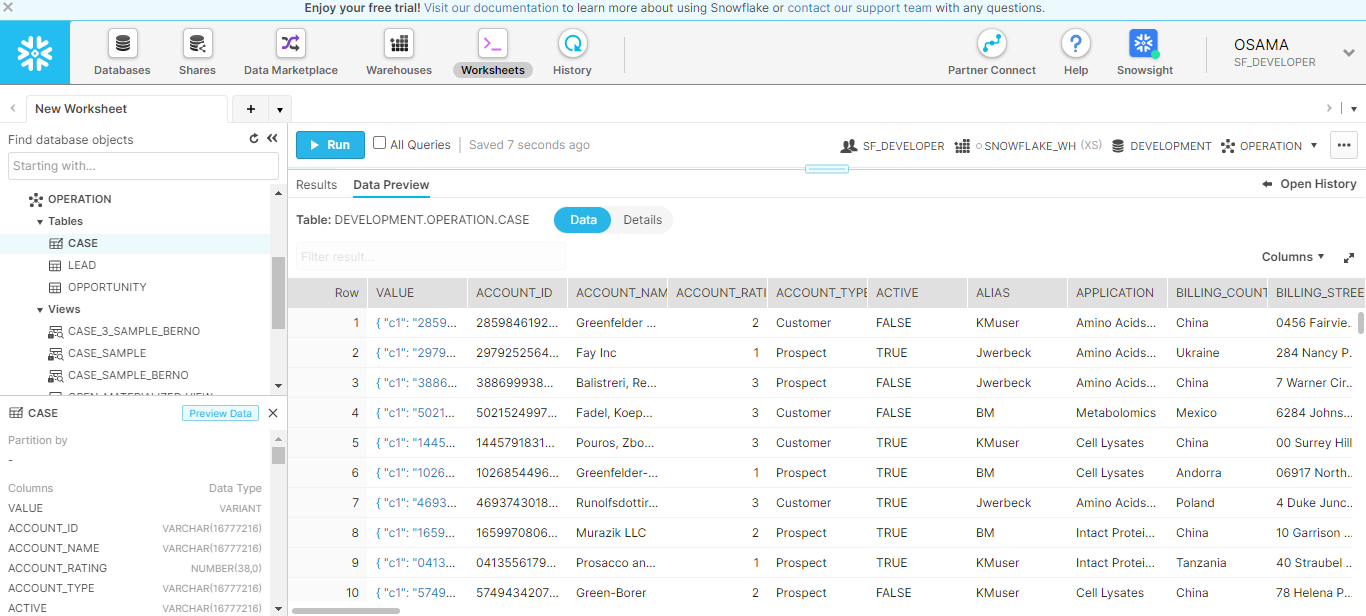
**Tables**

All data in Snowflake is stored in database tables, logically structured as collections of columns and rows.



**External Table**

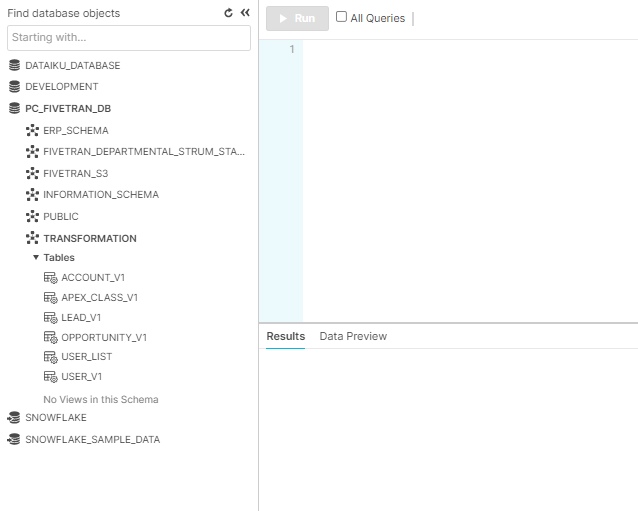
In a typical table, the data is stored in the database; however, in an external table, the data is stored in files in an external stage. External tables store file-level metadata about the data files, such as the filename, a version identifier and related properties. This enables querying data stored in files in an external stage as if it were inside a database.



**Transient Table**

Transient tables are similar to permanent tables with the key difference that they do not have a Fail-safe period.

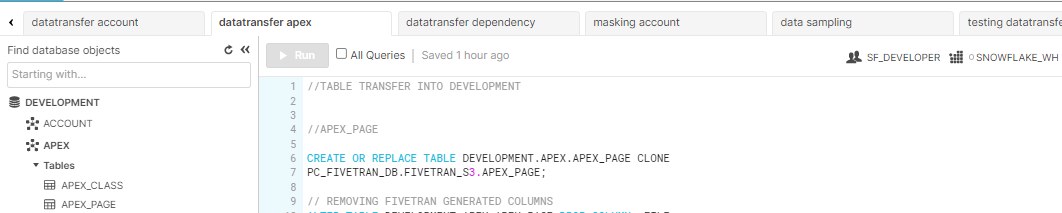
Transient tables are specifically designed for transitory data that needs to be maintained beyond each session enable



**4.6. Zero-copy Cloning**

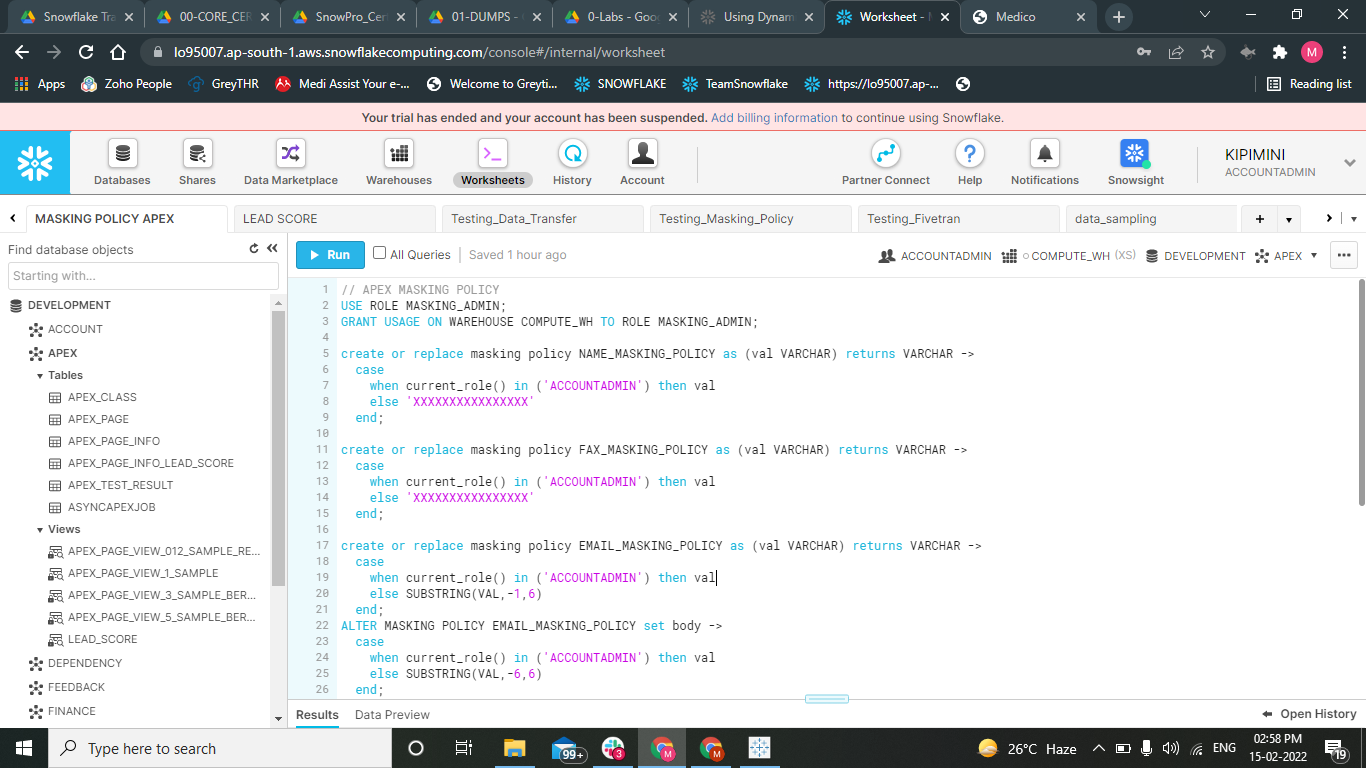
A clone is writable and is independent of its source (i.e. changes made to the source or clone are not reflected in the other object).Parameters that are explicitly set on a source database, schema, or table are retained in any clones created from the source container or child objects.

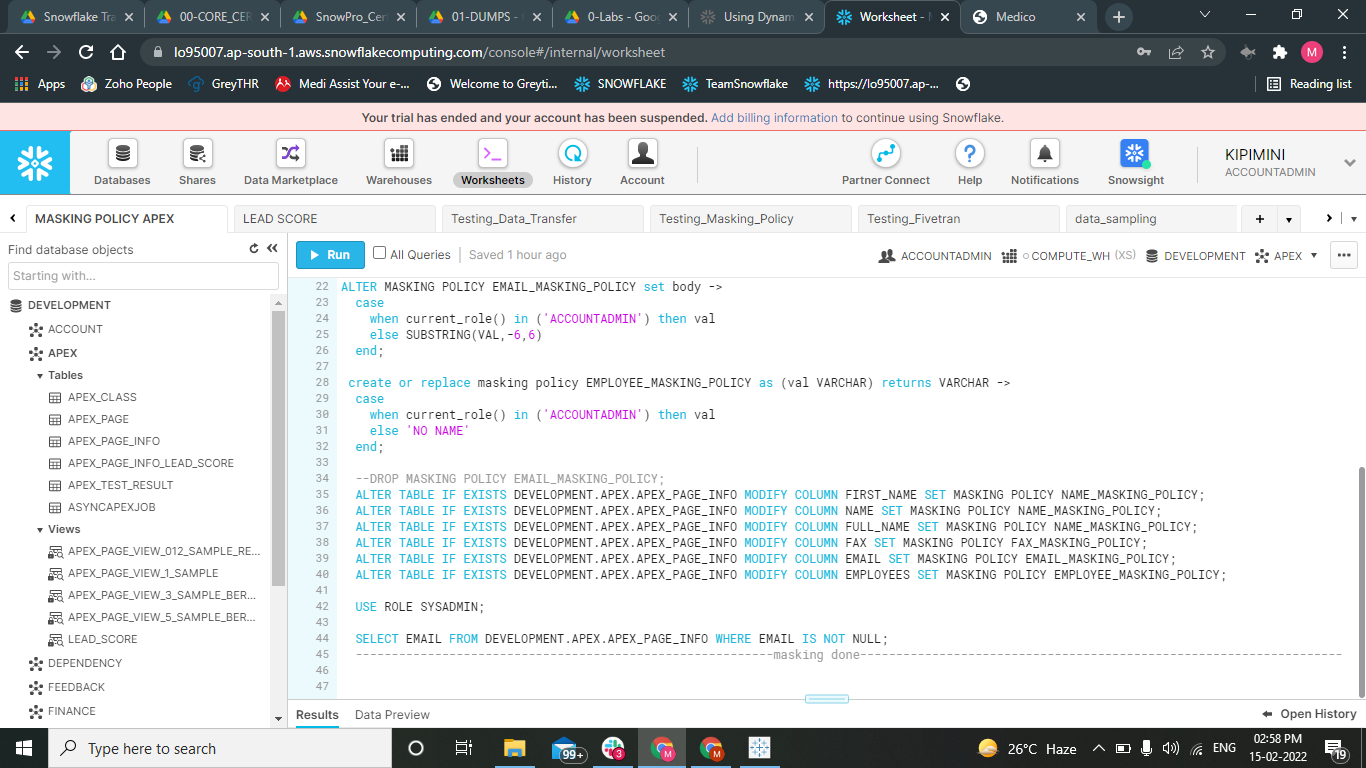
We have used the zero copy cloning feature of snowflake to transfer our data from our data lake PC\_FIVETRAN\_DB to data warehouse DEVELOPMENT.



**4.7. Data Masking**

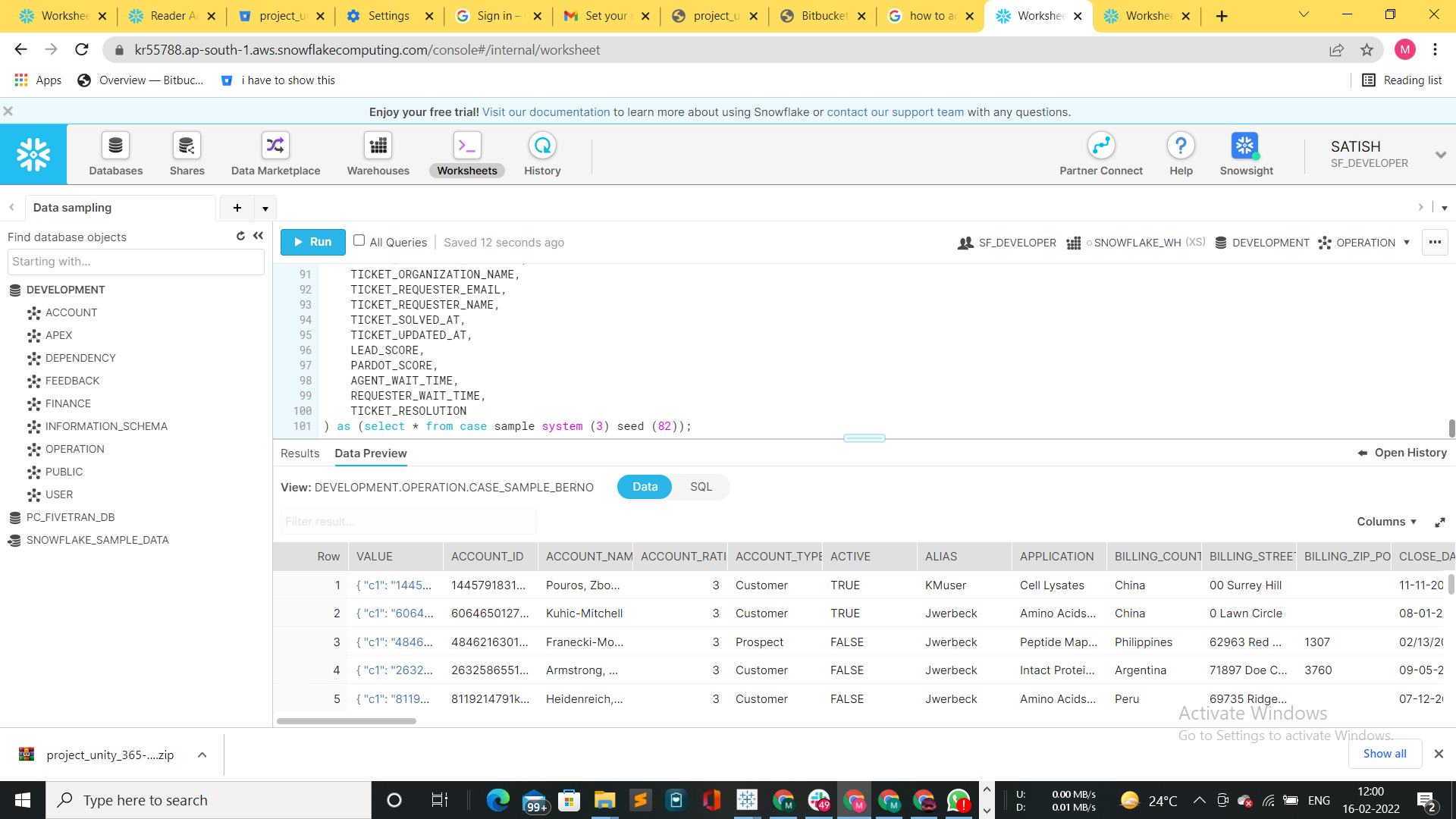
Data masking is a version of your organizational data in a confidential manner. The goal is to protect sensitive data, while providing a functional alternative when real data is not needed. The following are the code snippets of data masking used





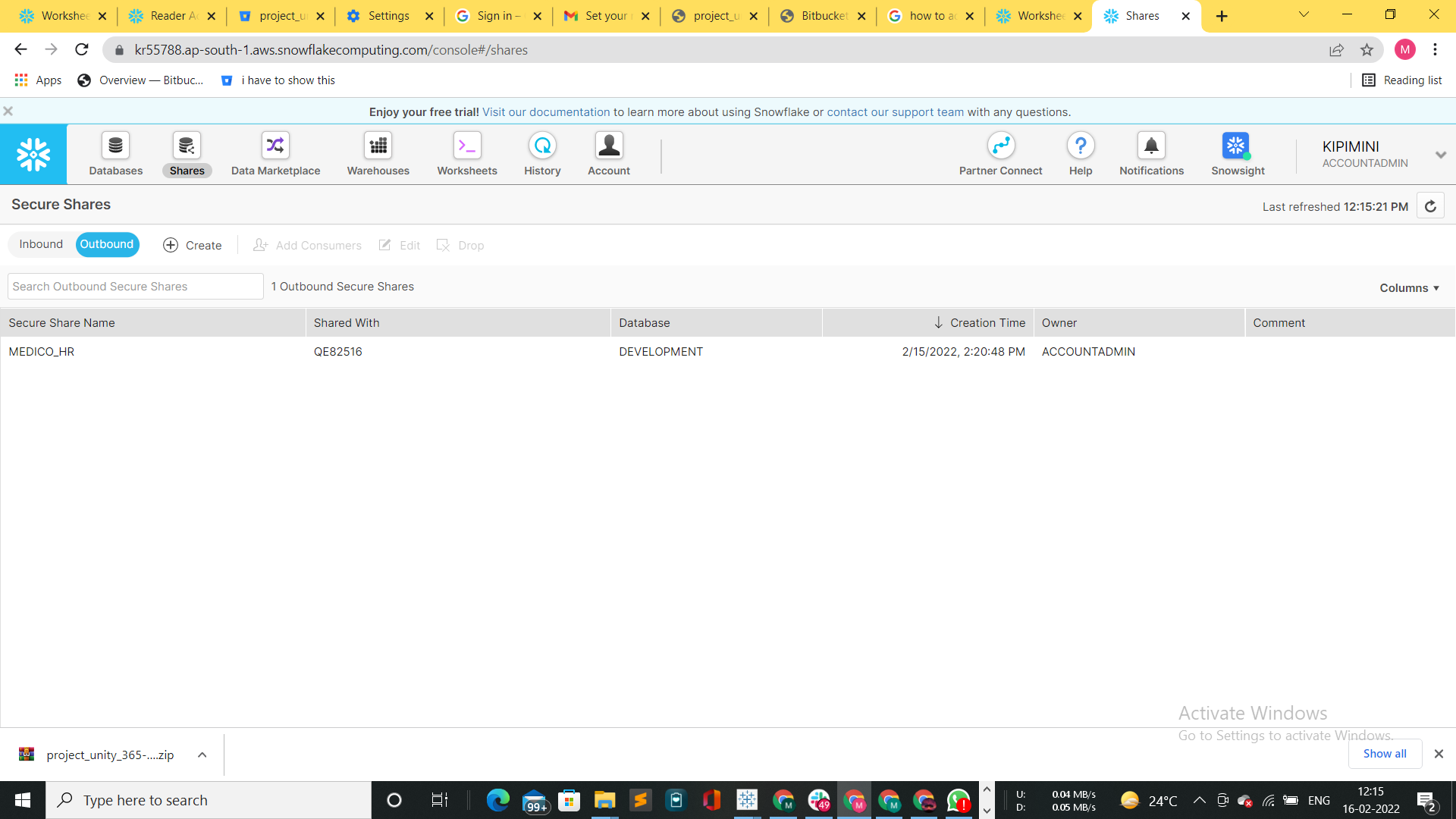
**4.8. Data Sampling**

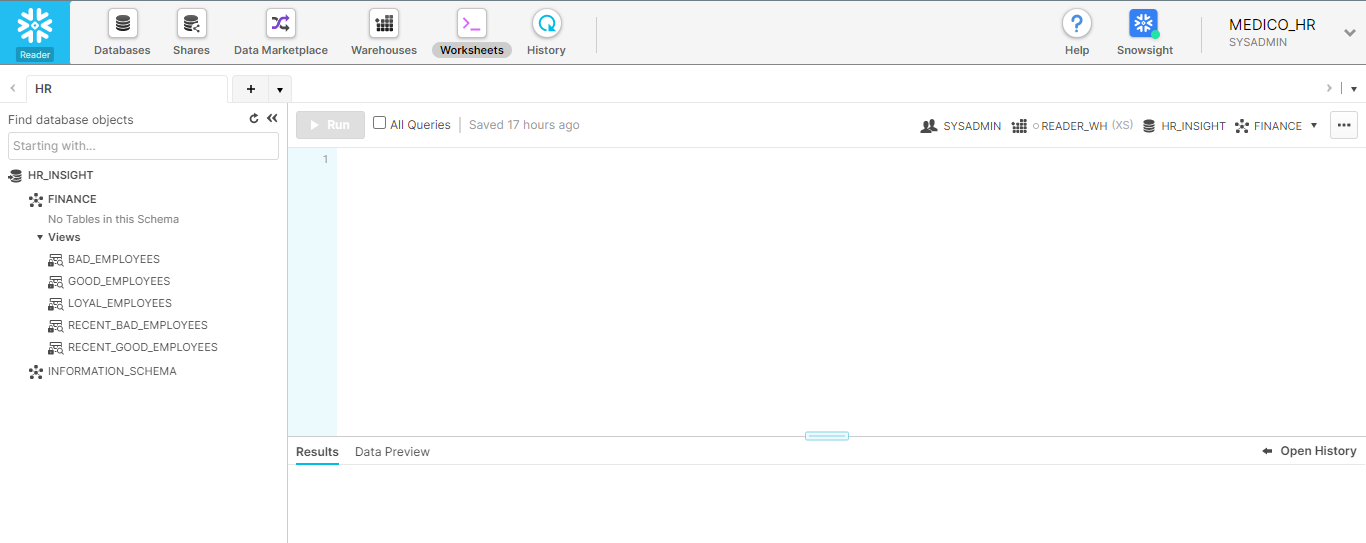
In data sampling, here we are creating three data samples either by randomly selecting rows or randomly selecting blocks of data (based on the micropartition schema adopted by snowflake) and in the sampling method we are using Bernoulli and System method for our case table to check the consistency of the data which will be further used by machine learning models.



**4.9. Secure Data Sharing**

Here we are creating a reader account for our client Medico and giving access to the Medico HR Team. And for that we are creating an outbound share for Medico HR and sharing the secure views i.e. employee details like: BAD\_EMPLOYEES, GOOD\_EMPLOYEES, etc. Which will be further used by their HR team.



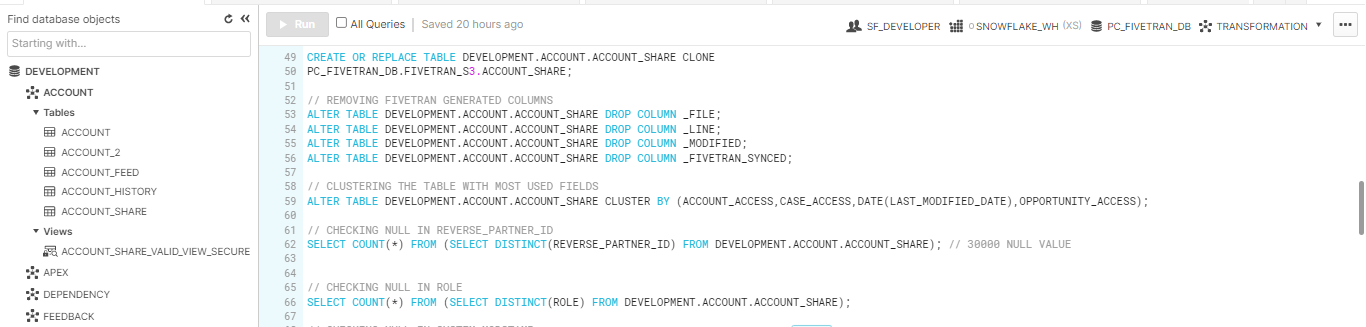


**4.10. Clustering**

Snowflake produces well-clustered data in tables; however, over time, particularly as DML occurs on very large tables the data in some table rows might no longer cluster optimally on desired dimensions.

To improve the clustering of the underlying table micro-partitions, you can always manually sort rows on key table columns and re-insert them into the table; however, performing these tasks could be cumbersome and expensive. Instead, Snowflake supports automating these tasks by designating one or more table columns/expressions as a *clustering key* for the table. A table with a clustering key defined is considered to be *clustered*. You can cluster [materialized views](https://docs.snowflake.com/en/user-guide/views-materialized.html), as well as tables. The rules for clustering tables and materialized views are generally the same.

We have performed clustering on tables like ACCOUNT\_2, ACCOUNT\_FEED, APEX\_PAGE, ZONE, APEX\_PAGE\_INFO, CAMPAIGN, VISUALFORCE ACCESS METRICS etc.



**4.11. Data Unloading**

Snowflake supports bulk export (i.e. unload) of data from a database table into flat, delimited text files.We have unloaded the feedback data to AWS S3 for further analysis of the data.



