

SNOWFLAKE PROJECT DOCUMENTATION

- **CAMP Batch B1 V3 – I**
- **Group 2**
- **Supervised by Mr. Pritam**

Team Members: -

Yashaswi Singh
Mohit Agarwal
Ritika Das
Stuti Bansal
Anivesh Gupta
Varan Kumar Gupta
Devashish Singh

INDEX

1. Project Description

2. Tasks to be performed

- Creation of database
- Creation of schemas
- Creation of table as per dataset
- Creation of Integration Object
- Creation of external stage for loading the data structure
- Creation of snowpipe for autoingesting of data from S3 bucket
- Creation of stream on the given table
- SCD 2 operation on the consumer table

3. Data Analysis

Project Description: This project is to ingest and analyze a [dataset](#) from kaggle having details related to causes which can lead to heart disease.

Tasks Performed: -

Query: Create database SF_PROJECT;

Description: Created a database name SF_PROJECT

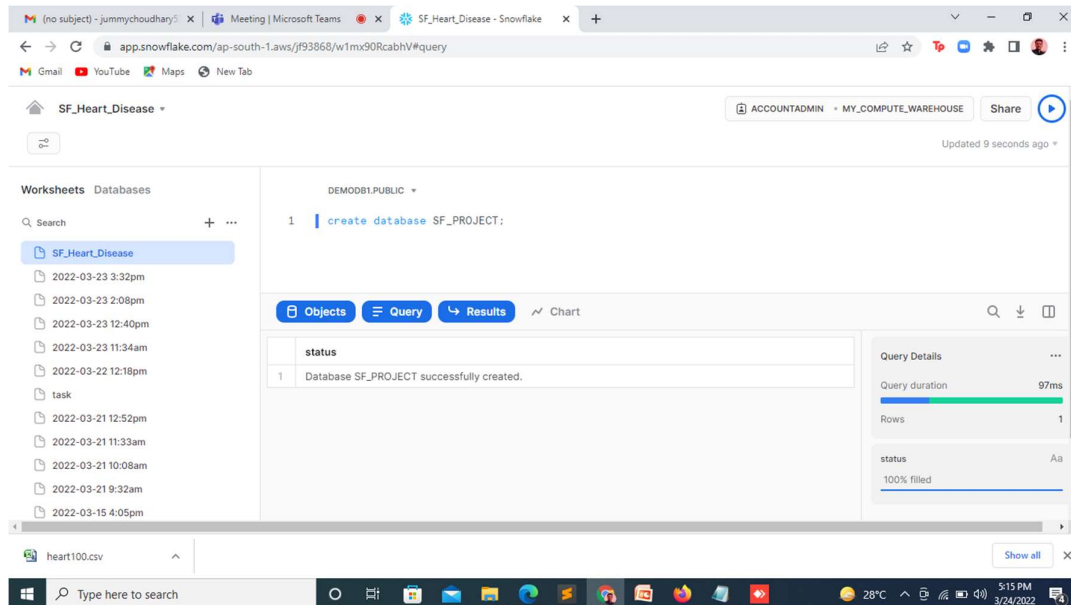


Fig 1

Query: create schema SF_Project.itr_rds;
create schema SF_Project.itr_dm;
create schema SF_Project.itr_rds_landing;

Description: Created 3 schemas namely ITR_RDS, ITR_DM, ITE_RDS_LANDING

**SCHEMA: A database schema defines how data is organized within a relational database.*

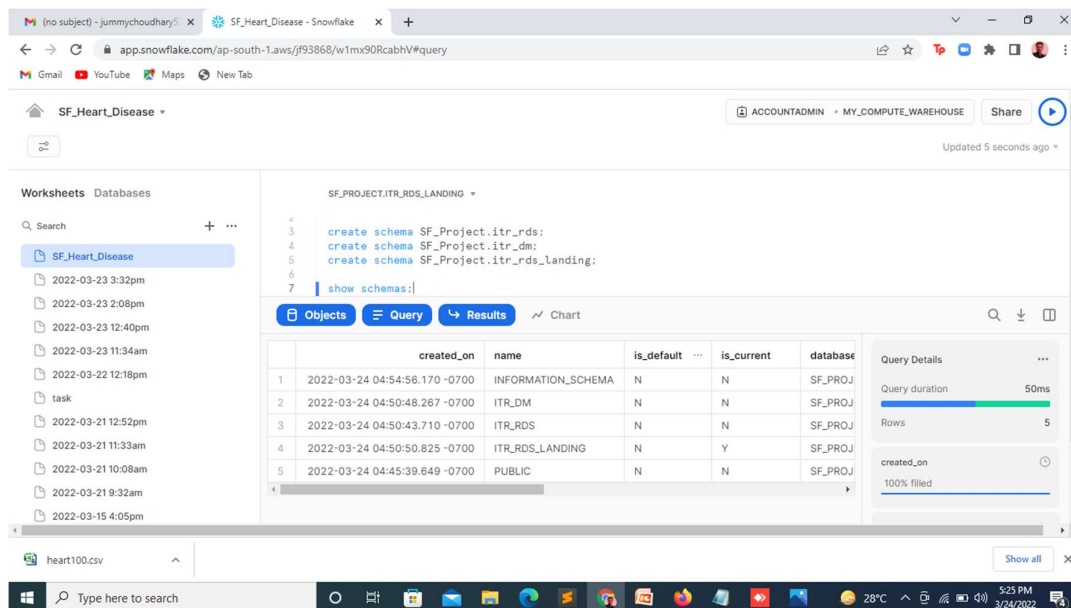


Fig 2

Query: create table heart_disease (
heartdisease varchar,
BMI decimal,
smoking varchar,
AlcoholDrinking varchar,
Stroke varchar,
PhysicalHealth number,
MentalHealth number,
DiffWalking varchar,
Sex varchar,
AgeCategory varchar,
Race varchar,
Diabetic varchar,
PhysicalActivity varchar,
GenHealth varchar,
SleepTime number,
Asthma varchar,
KidneyDisease varchar,
SkinCancer varchar);

Description: Table is created with the desired columns and its specified datatype.

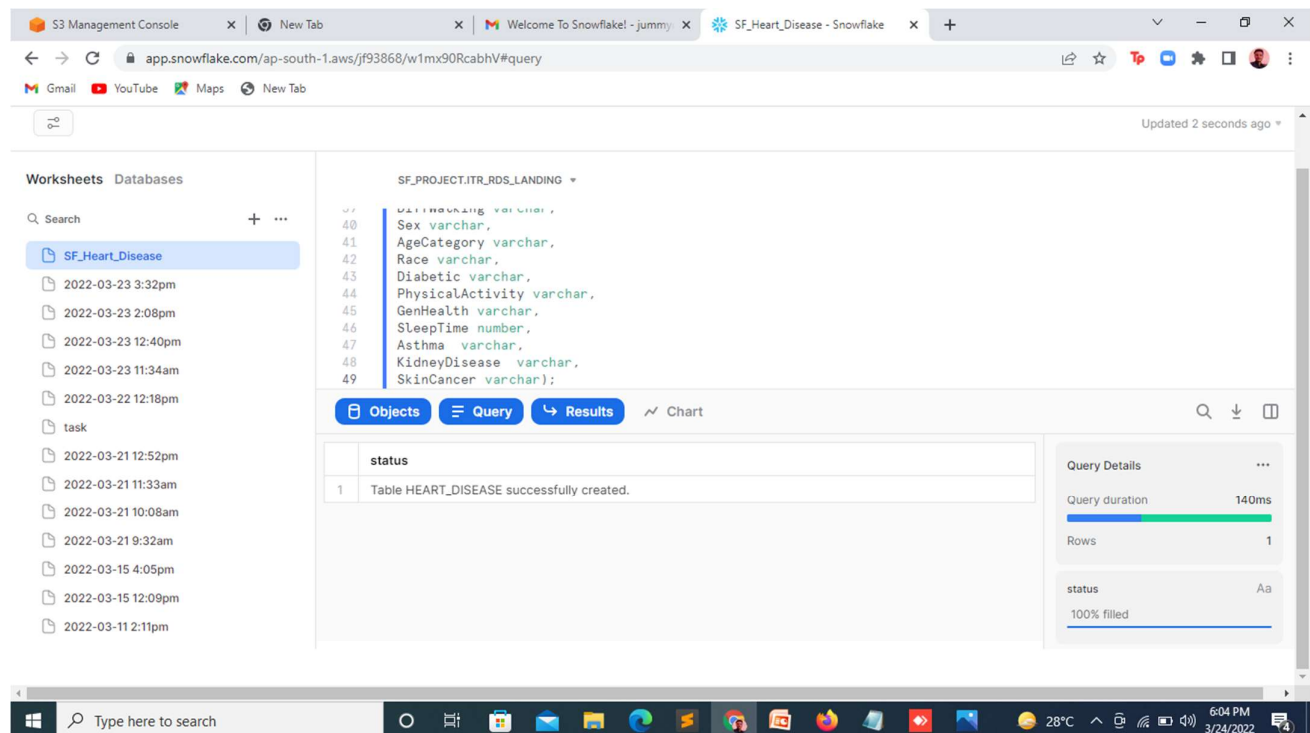


Fig 3

Query: desc table heart_disease;

Description: Showing the details of the table

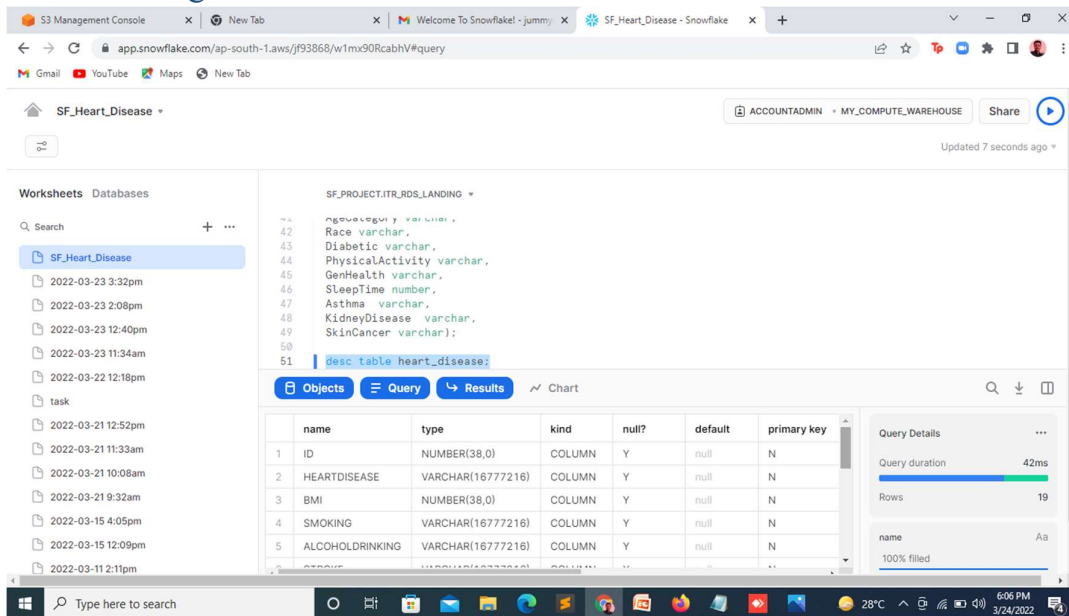


Fig 4

Query: create or replace storage integration s3_int_obj
type = external_stage
storage_provider = s3
enabled = true
storage_aws_role_arn = 'arn:aws:iam::084370864130:role/flatbucket6_policy_role'
storage_allowed_locations = ('s3://flatbucket6/');

Description: Connecting snowflake with our s3 bucket named FLATBUCKET6 in AWS.

**Storage integration: A storage integration is a Snowflake object that stores a generated identity and access management (IAM) entity for your external cloud storage, along with an optional set of allowed or blocked storage locations (Amazon S3, Google Cloud Storage, or Microsoft Azure).*

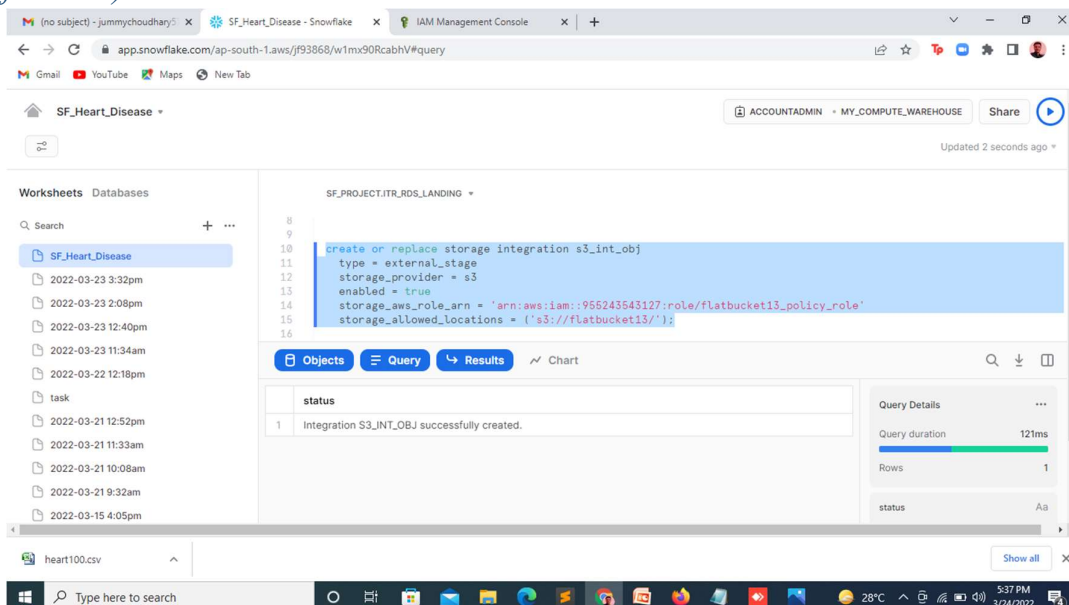


Fig 5

Query: create or replace stage sf_project.itr_rds_landing.my_ext_stage
 storage_integration = s3_int_obj
 url = 's3://flatbucket12'
 file_format = (type = csv field_delimiter=',' skip_header = 1 null_if =
 ('NULL','null') empty_field_as_null = true field_optionally_enclosed_by='');

Description: Here an external stage is created named as MY_EXT_STAGE with storage integration as S3_INT_OBJ. The file is formatted in csv form with delimiter as “,” and with the help of skip_header attribute as 1, the first row i.e. the header of the file will be skipped.
**External stage: Creates an interface between Snowflake and an external cloud storage location.*

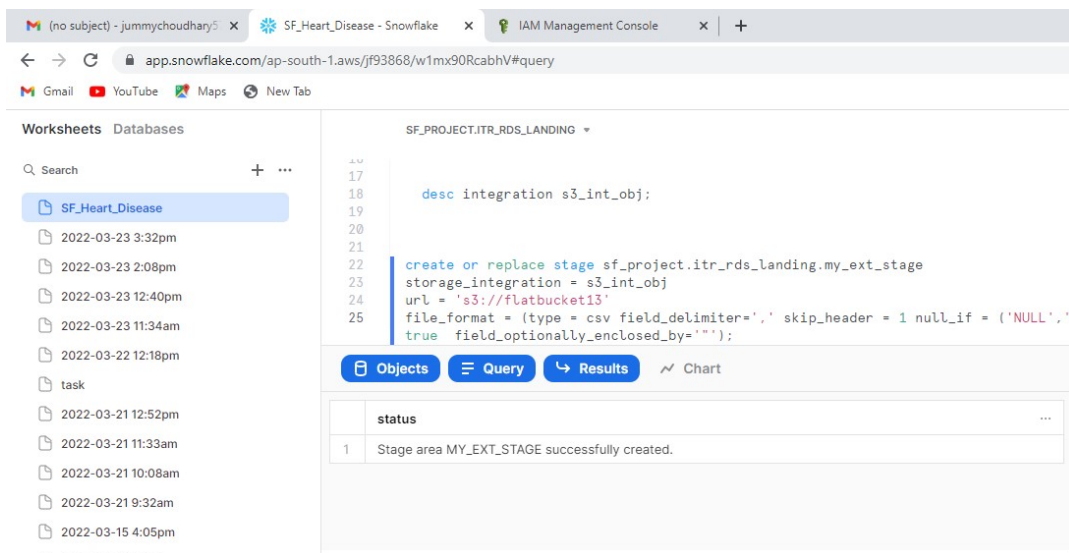


Fig 6

Query: list @sf_project.itr_rds_landing.my_ext_stage;

Description: Listing the details of external stage.

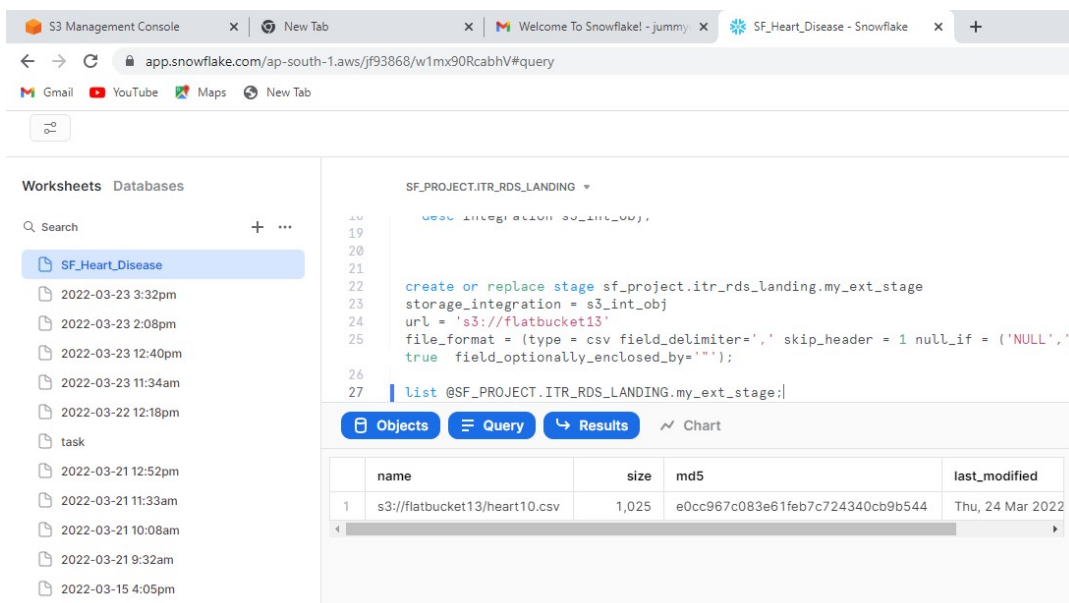


Fig 7.1

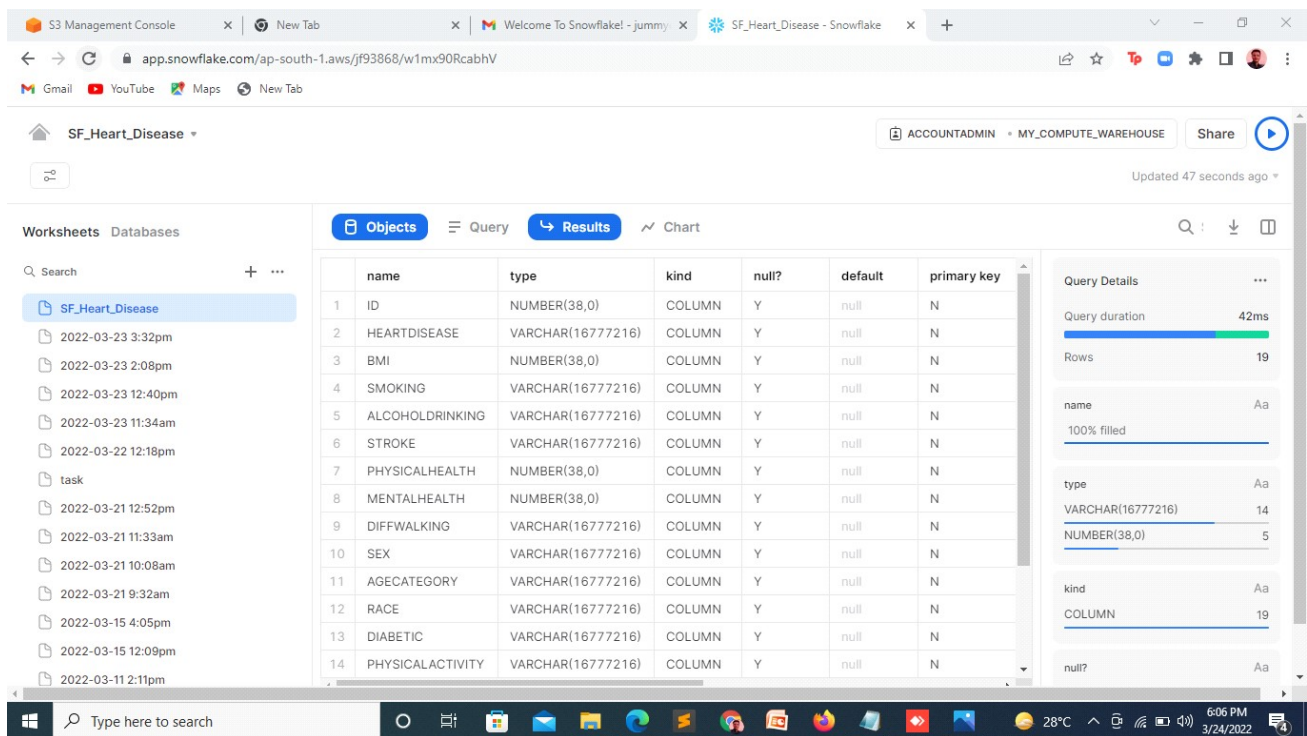


Fig 7.2

Query: create or replace pipe sf_project.itr_rds_landing.sf_snowpipe1
 auto_ingest=true as
 copy into sf_project.itr_rds_landing.heart_disease
 rom @sf_project.itr_rds_landing.my_ext_stage;

Description: Here a pipe is created named as SF_SNOWPIPE1 given auto_ingest as true i.e whenever new data is found in the MY_EXT_STAGE, then it is automatically inserted in the table.

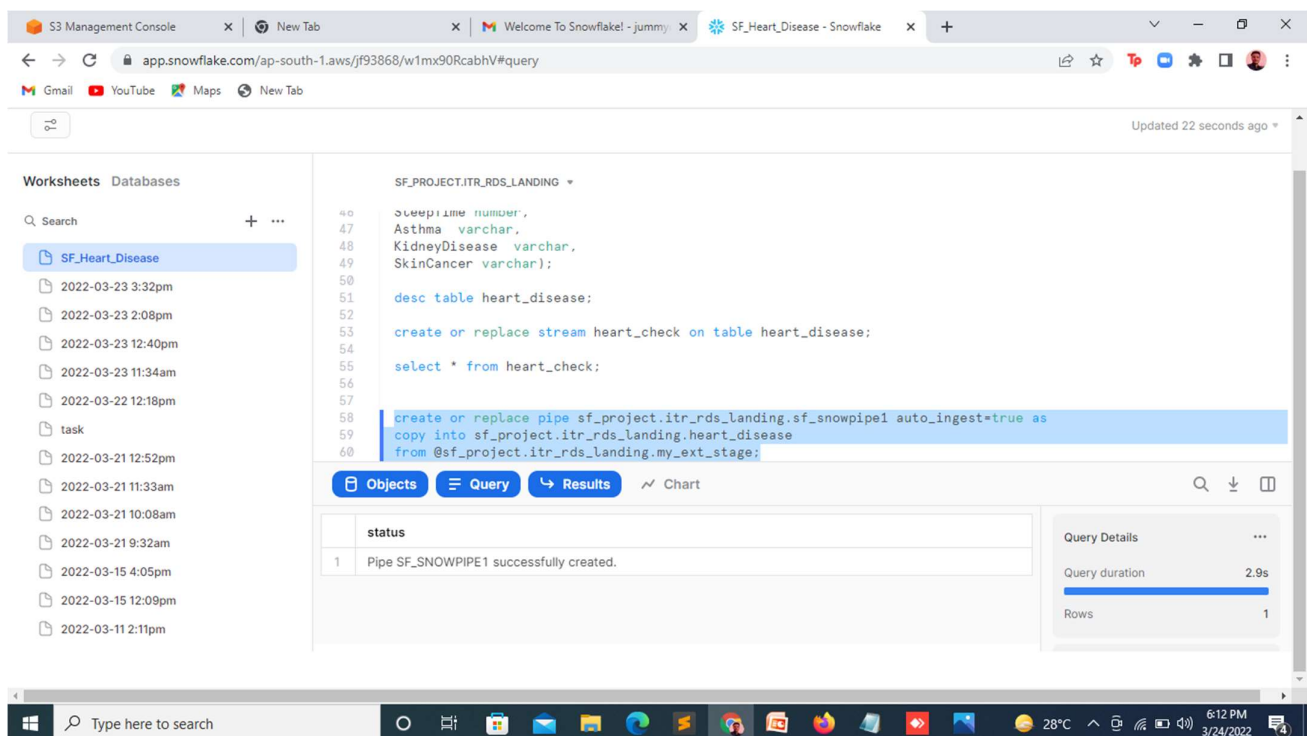


Fig 8

Query: show pipes;

Description: Showing the description of pipes created above.

The screenshot shows the Snowflake Management Console interface. The left sidebar displays a list of worksheets under the 'SF_Heart_Disease' database, with the most recent one selected. The main query editor shows a SQL script that includes creating a stream, a table, and a pipe. The query is executed, and the results are displayed in a table below the editor. The table has columns for 'created_on', 'name', 'database_name', 'schema_name', and 'definition'. The results show one row for the pipe 'SF_SNOWPIPE1' in the 'SF_PROJECT' database and 'ITR_RDS_LANDING' schema. The 'Query Details' panel on the right indicates a query duration of 58ms and 1 row returned.

```
47  <table></table>
48  <table></table>
49  <table></table>
50  <table></table>
51  desc table heart_disease;
52
53  create or replace stream heart_check on table heart_disease;
54
55  select * from heart_check;
56
57
58  create or replace pipe sf_project.itr_rds_landing.sf_snowpipe1 auto_ingest=true as
59  copy into sf_project.itr_rds_landing.heart_disease
60  from @sf_project.itr_rds_landing.my_ext_stage;
61
62
63  show pipes;
```

| | created_on | name | database_name | schema_name | definition |
|---|-------------------------------|--------------|---------------|-----------------|------------|
| 1 | 2022-03-24 05:42:26.048 -0700 | SF_SNOWPIPE1 | SF_PROJECT | ITR_RDS_LANDING | copy i |

Query Details

- Query duration: 58ms
- Rows: 1

Fig. 9

Query: alter pipe sf_project.itr_rds_landing.sf_snowpipe1 refresh;

Description: Tells whether the dataset is sent or not.

The screenshot shows the Snowflake Management Console interface. The left sidebar displays a list of worksheets under the 'SF_Heart_Disease' database, with the most recent one selected. The main query editor shows a SQL script that includes creating a stream, a table, and a pipe, followed by a 'show pipes;' query and an 'alter pipe' query to refresh the pipe. The query is executed, and the results are displayed in a table below the editor. The table has columns for 'File' and 'Status'. The results show one row for the file 'heart10.csv' with a status of 'SENT'. The 'Query Details' panel on the right indicates a query duration of 1.2s and 1 row returned.

```
53  <table></table>
54  create or replace stream heart_check on table heart_disease;
55
56  select * from heart_check;
57
58
59  create or replace pipe sf_project.itr_rds_landing.sf_snowpipe1 auto_ingest=true as
60  copy into sf_project.itr_rds_landing.heart_disease
61  from @sf_project.itr_rds_landing.my_ext_stage;
62
63  show pipes;
64
65
66
67  alter pipe sf_project.itr_rds_landing.sf_snowpipe1 refresh;
```

| | File | Status |
|---|-------------|--------|
| 1 | heart10.csv | SENT |

Query Details

- Query duration: 1.2s
- Rows: 1

Fig. 10

Query: create table heart_disease_tgt (
 id number,
 heartdisease varchar,
 BMI decimal,
 smoking varchar,
 AlcoholDrinking varchar,
 Stroke varchar,
 PhysicalHealth number,
 MentalHealth number,
 DiffWalking varchar,
 Sex varchar,
 AgeCategory varchar,
 Race varchar,
 Diabetic varchar,
 PhysicalActivity varchar,
 GenHealth varchar,
 SleepTime number,
 Asthma varchar,
 KidneyDisease varchar,
 SkinCancer varchar,
 stream_type string default null,
 rec_version number default 0,
 REC_DATE TIMESTAMP_LTZ);

Description: Target table is created with the desired columns and its specified datatype.

The screenshot displays the Snowflake web interface. On the left, the 'Worksheets' panel shows a list of worksheets, with 'SF_Heart_Disease' selected. The main panel shows the query execution results for the query 'SF_PROJECT.ITR_RDS_LANDING'. The query text is visible at the top, and the results are shown in a table format. The table has 8 columns: name, type, kind, null?, default, primary key, and an additional column for the value. The rows show the details for each column created in the table.

| | name | type | kind | null? | default | primary key |
|----|---------------|-------------------|--------|-------|---------|-------------|
| 15 | GENHEALTH | VARCHAR(16777216) | COLUMN | Y | null | N |
| 16 | SLEEPTIME | NUMBER(38,0) | COLUMN | Y | null | N |
| 17 | ASTHMA | VARCHAR(16777216) | COLUMN | Y | null | N |
| 18 | KIDNEYDISEASE | VARCHAR(16777216) | COLUMN | Y | null | N |
| 19 | SKINCANCER | VARCHAR(16777216) | COLUMN | Y | null | N |
| 20 | STREAM_TYPE | VARCHAR(16777216) | COLUMN | Y | null | N |
| 21 | REC_VERSION | NUMBER(38,0) | COLUMN | Y | 0 | N |
| 22 | REC_DATE | TIMESTAMP_LTZ(9) | COLUMN | Y | null | N |

Query Details: Query duration 41ms, Rows 22.

Fig. 11

Query: create or replace stream HEART_CHECK on table HEART_DISEASE;

Description: Created a stream to update and load data from landing table to consumer table.

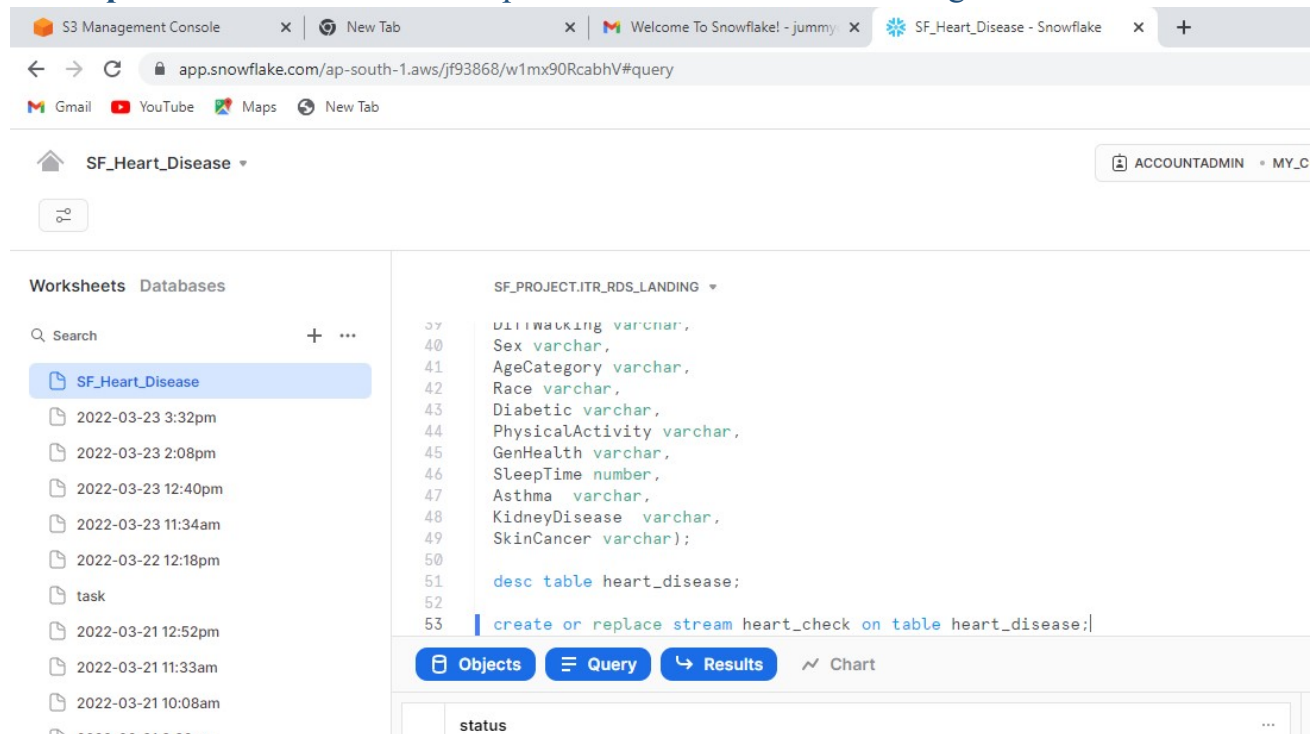


Fig. 12

Query: CREATE TASK heart_task

WAREHOUSE = my_first_warehouse

SCHEDULE = '1 minute'

WHEN

SYSTEM\$STREAM_HAS_DATA('heart_check')

AS

merge into heart_disease_tgt t

using heart_check s

on t.id=s.id and (metadata\$action='DELETE')

when matched and metadata\$isupdate='FALSE'

then update set rec_version=9999,

stream_type='DELETE' when matched

and metadata\$isupdate='TRUE' then update set rec_version=rec_version-1

when not matched then

insert (id, heartdisease, BMI, smoking, Alcohol Drinking, Stroke, PhysicalHealth
, MentalHealth, DiffWalking, Sex, AgeCategory, Race, Diabetic
, PhysicalActivity, GenHealth, SleepTime, Asthma, KidneyDisease, SkinCancer
, stream_type, rec_version, REC_DATE)

values(s.id, s.heartdisease , s.BMI ,smoking ,s.AlcoholDrinking ,s.Stroke
,s.PhysicalHealth ,s.MentalHealth ,s.DiffWalking ,s.Sex , s.AgeCategory ,s.Race
,s.Diabetic ,s.PhysicalActivity,s.GenHealth ,s.SleepTime,s.Asthma,s.KidneyDisease
,s.SkinCancer, metadata\$action,0,CURRENT_TIMESTAMP());

Description: A task is created to automate changes from landing table to consumer table and here we are using type 2 scd.

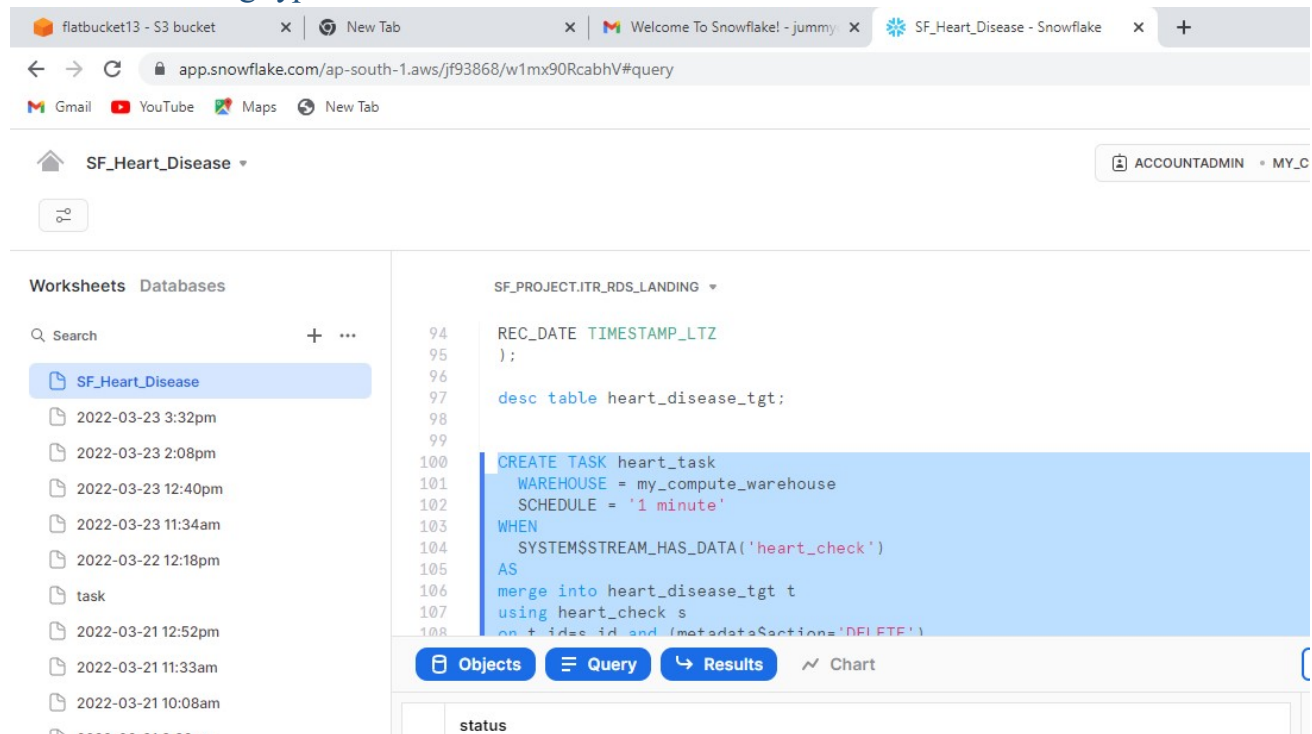


Fig. 13

Query: select * from heart_disease_tgt;

Description: showing the whole target table i.e HEART_DISEASE_TGT at once.

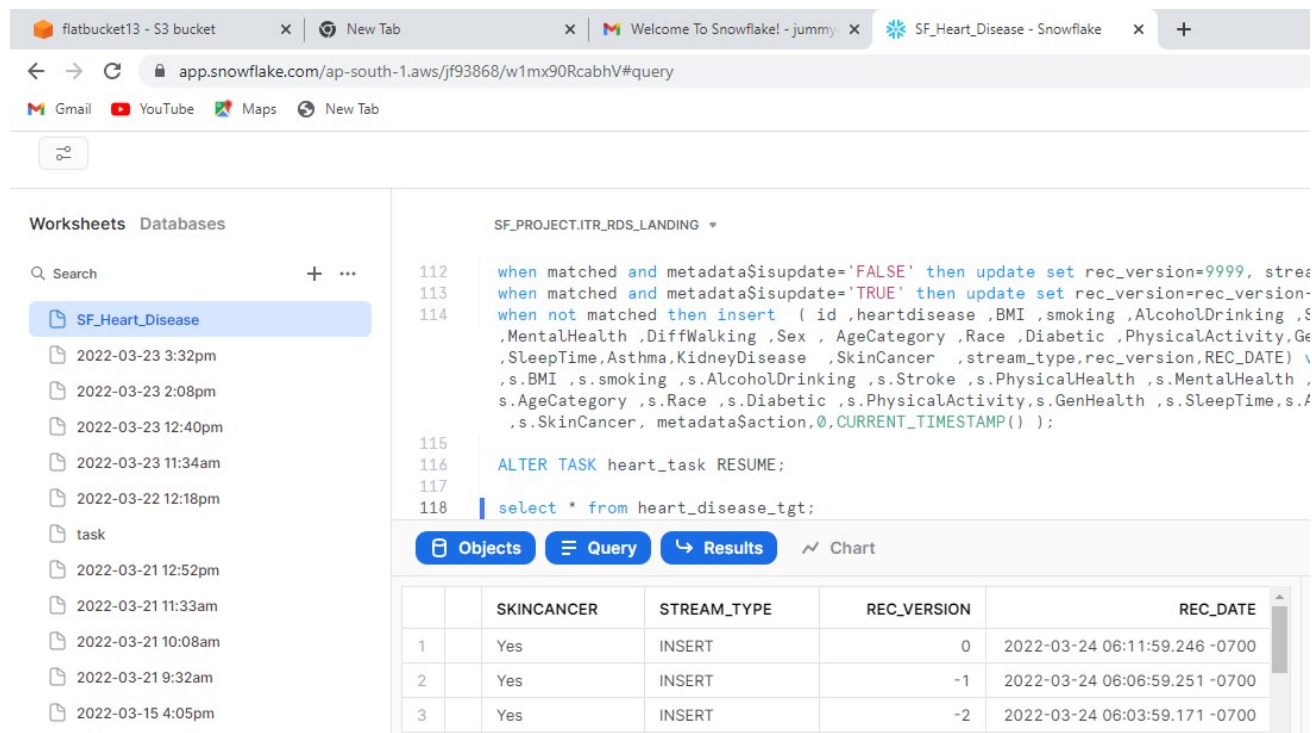


Fig. 14

Data Analysis: -

- Which gender have majority heart attack?

select sex, count(heartdisease) from heart_disease where heartdisease='Yes' group by sex;

The screenshot shows the Snowflake SQL interface. The query editor contains the following SQL code:

```
select * from heart_disease;

select sex, count(heartdisease) from heart_disease where heartdisease='Yes' group by sex;

select physicalhealth, mentalhealth from heart_disease where stroke='Yes';
select avg(bmi), genhealth from heart_disease group by genhealth;
select avg(bmi) from heart_disease where diffwalking='Yes';

select distinct(agecategory) from heart_disease;

select avg(bmi) from heart_disease where skincancer='Yes';
```

The results tab shows a table with two columns: SEX and COUNT(HEARTDISEASE). The data is as follows:

| SEX | COUNT(HEARTDISEASE) |
|--------|---------------------|
| Female | 1 |

Solution 1

- Which combination of physical and mental health causes stroke?

select physicalhealth, mentalhealth from heart_disease where stroke='Yes';

The screenshot shows the Snowflake SQL interface. The query editor contains the following SQL code:

```
select * from heart_disease;

select sex, count(heartdisease) from heart_disease where heartdisease='Yes' group by sex;

select physicalhealth, mentalhealth from heart_disease where stroke='Yes';
select avg(bmi), genhealth from heart_disease group by genhealth;
select avg(bmi) from heart_disease where diffwalking='Yes';

select distinct(agecategory) from heart_disease;

select avg(bmi) from heart_disease where skincancer='Yes';
```

The results tab shows a table with two columns: PHYSICALHEALTH and MENTALHEALTH. The data is as follows:

| PHYSICALHEALTH | MENTALHEALTH |
|----------------|--------------|
| 0 | 0 |

Solution 2

- GenHealth wise average BMI of people?
`select avg(bmi),genhealth from heart_disease group by genhealth;`

The screenshot shows the Snowflake SQL interface. The left sidebar lists worksheets, with 'Heart_Disease_Worksheet' selected. The main area displays a SQL query in the editor:

```
125 select * from heart_disease;
126
127
128 select sex,count(heartdisease) from heart_disease where heartdisease='Yes' group by sex;
129
130 select physicalhealth,mentalhealth from heart_disease where stroke='Yes';
131 select avg(bmi),genhealth from heart_disease group by genhealth;
132 select avg(bmi) from heart_disease where diffwalking='Yes';
133
134 select distinct(agecategory) from heart_disease;
135
136
137
138 select avg(bmi) from heart_disease where skincancer='Yes';
139
140
141
```

Below the editor, the 'Results' tab is active, showing a table with the following data:

| | AVG(BMI) | GENHEALTH |
|---|-----------|-----------|
| 1 | 20.333333 | Very good |

Solution 3

- Average age of people having problem in walking?
`select avg(bmi) from heart_disease where diffwalking='Yes';`

The screenshot shows the Snowflake SQL interface. The left sidebar lists worksheets, with 'Heart_Disease_Worksheet' selected. The main area displays a SQL query in the editor:

```
125 select * from heart_disease;
126
127
128 select sex,count(heartdisease) from heart_disease where heartdisease='Yes' group by sex;
129
130 select physicalhealth,mentalhealth from heart_disease where stroke='Yes';
131 select avg(bmi),genhealth from heart_disease group by genhealth;
132 select avg(bmi) from heart_disease where diffwalking='Yes';
133
134 select distinct(agecategory) from heart_disease;
135
136
137
138 select avg(bmi) from heart_disease where skincancer='Yes';
139
140
141
```

Below the editor, the 'Results' tab is active, showing a table with the following data:

| | AVG(BMI) |
|---|----------|
| 1 | 31.1 |

Solution 4

➤ Average BMI of people having Skin Cancer

Select avg(bmi) from heart_disease where skincancer='Yes';

The screenshot shows the Snowflake SQL interface. The left sidebar displays a list of worksheets, with 'Heart_Disease_Worksheet' selected. The main query editor contains the following SQL code:

```

125 select * from heart_disease;
126
127
128 select sex,count(heartdisease) from heart_disease where heartdisease='Yes' group by sex;
129
130 select physicalhealth,mentalhealth from heart_disease where stroke='Yes';
131 select avg(bmi),genhealth from heart_disease group by genhealth;
132 select avg(bmi) from heart_disease where diffwalking='Yes';
133
134 select distinct(agecategory) from heart_disease;
135
136
137
138 | select avg(bmi) from heart_disease where skincancer='Yes';
139
140
141
  
```

Below the query editor, the 'Results' tab is active, showing a table with one column, 'AVG(BMI)', and one row with the value '2'.

Solution 5

➤ Possibilities/Percentage of people having both Heart Diseases stroke and Skin Cancer

Select count(*) from heart_disease where heartdisease='Yes' and skincancer='Yes';

The screenshot shows the Snowflake SQL interface. The left sidebar displays a list of worksheets, with 'Heart_Disease_Worksheet' selected. The main query editor contains the following SQL code:

```

125 select * from heart_disease;
126
127
128 select sex,count(heartdisease) from heart_disease where heartdisease='Yes' group by sex;
129
130 select physicalhealth,mentalhealth from heart_disease where stroke='Yes';
131 select avg(bmi),genhealth from heart_disease group by genhealth;
132 select avg(bmi) from heart_disease where diffwalking='Yes';
133
134 select distinct(agecategory) from heart_disease;
135
136
137
138
139
140 | select count(*) from heart_disease where heartdisease='Yes' and skincancer='Yes';
141
  
```

Below the query editor, the 'Results' tab is active, showing a table with one column, 'COUNT(*)', and one row with the value '1'.

Solution 6