A) Snowflakes Time Travel Functionality:

1. write statements to Create table customer with Customerid and Customer name columns with suitable data types and data retention of 50 days and then later use alter table command to change data retention in days to 20.

create table Customer (Custid varchar,Cusname varchar)

data\_retention\_time\_in\_days=50

alter table Customer set data\_retention\_time\_in\_days=20

show tables

2. Insert Couple of customer records in above table and after 2 minutes insert another 2 records into customer table . Then execute query which selects historical data from table as of 3 minutes. How many records will be there in table?

insert into Customer ( Custid,Cusname) values (10,'tulasiram')

insert into Customer ( Custid,Cusname) values (20,'tulasi')

insert into Customer ( Custid,Cusname) values (30,'ram')

Gap time 3 mins

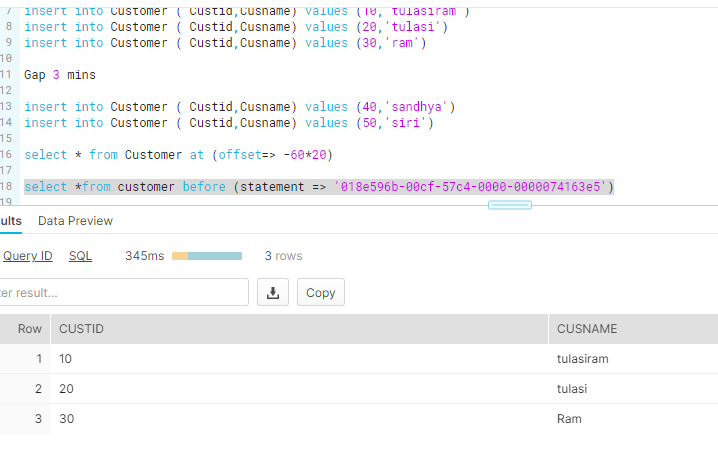
insert into Customer ( Custid,Cusname) values (40,'sandhya')

insert into Customer ( Custid,Cusname) values (50,'siri')

select \* from Customer at (offset=> -60\*20)

3. Write query that selects historical data from a customer table up to, but not including any changes made by the first insert statement of 2nd question above.

select \*from customer before (statement => '018e596b-00cf-57c4-0000-0000074163e5')



B) Data Cloning:

1. Write statements to Create database by name mydb , schema by name myschema and create table by name department with departmentnumber , departmentname and location with suitable datatypes and insert couple of records.

After 1 hour drop table department and then Write statement that creates a clone of a schema by name myschema\_clone and all its objects as they existed 1 hour before the current time. Does myschema\_clone have department table?

create database mydb

create schema myschema

create table department (Deptno varchar,Deptname varchar, location varchar)

insert into department (deptno, deptname,location) values (1,'IT','hyd'),(2,'CSC','Bng'),(3,'EEE','BZA')

select \* from department

drop table department

create schema myschema\_clone clone myschema at(offset => -3600);

use schema myschema\_clone

select \* from department

show tables

2. Create database mydb\_clone that will create a clone of mydb database and all its objects as they existed prior to the completion of the insert statement in first question above. How many records are there in department table?

create database mydb\_clone clone mydb before(statement => '018e6077-00e3-8059-0000-000007418785'); 🡺 Insert statement query id

use database mydb\_clone;

show tables;

select \* from department

C) DataSharing:

1. Create 2 snowflake accounts. One will be data provider and another data consumer

In data provider account create database Productdb , Productschema and create table products with productid, productname and price with suitable datatypes and insert couple of records

Create share by name productshare in dataprovider account and share product table with consumer account and then loginto consumer account query produts table.

Pl provide statements along with snapshot of query results as part of solution.

Provider Account

----------------------

create database Productdb

create schema Productschema

create table products(prodid varchar, prodname varchar, price varchar)

insert into products(prodid,prodname,price) values (12,'Electronics',1200),(13,'Food',1500)

select \* from products

use role accountadmin

create share productshare;

grant usage on database Productdb to share productshare;

grant usage on schema productdb.productschema to share productshare

grant select on table Productdb.Productschema.products to share productshare

show grants to share productshare;

alter share productshare add accounts=cw33055;

Consumer

* ---------------

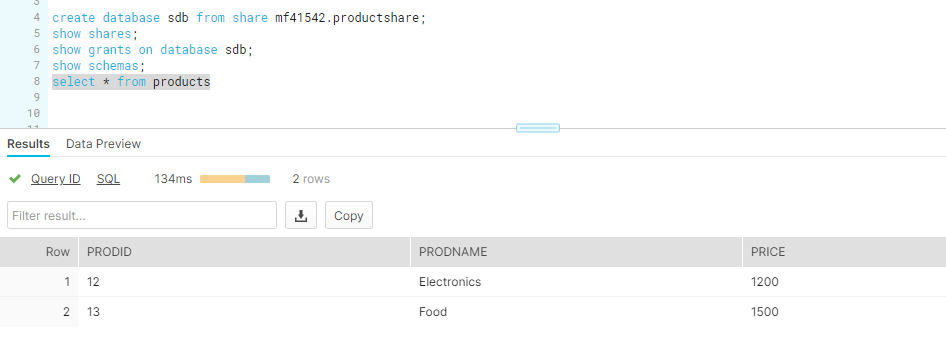
create database sdb from share mf41542.productshare;

show shares;

show grants on database sdb;

show schemas;

select \* from products



2. Secure Views in Snowflake

Create secure view by name vw\_product in provider account which will restrict productid field from products table in question 1 above and share that view with consumer account and when view is queried from consumer account it should display productname and price data only.

Provider

-----------

create or replace secure view vw\_product as select prodname,price from products

select \* from vw\_product ;

grant select on view Productdb.Productschema.vw\_product to share productshare

Consumer

-------------

select \* from vw\_product;

