**A) Snowflakes Time Travel Functionality:**

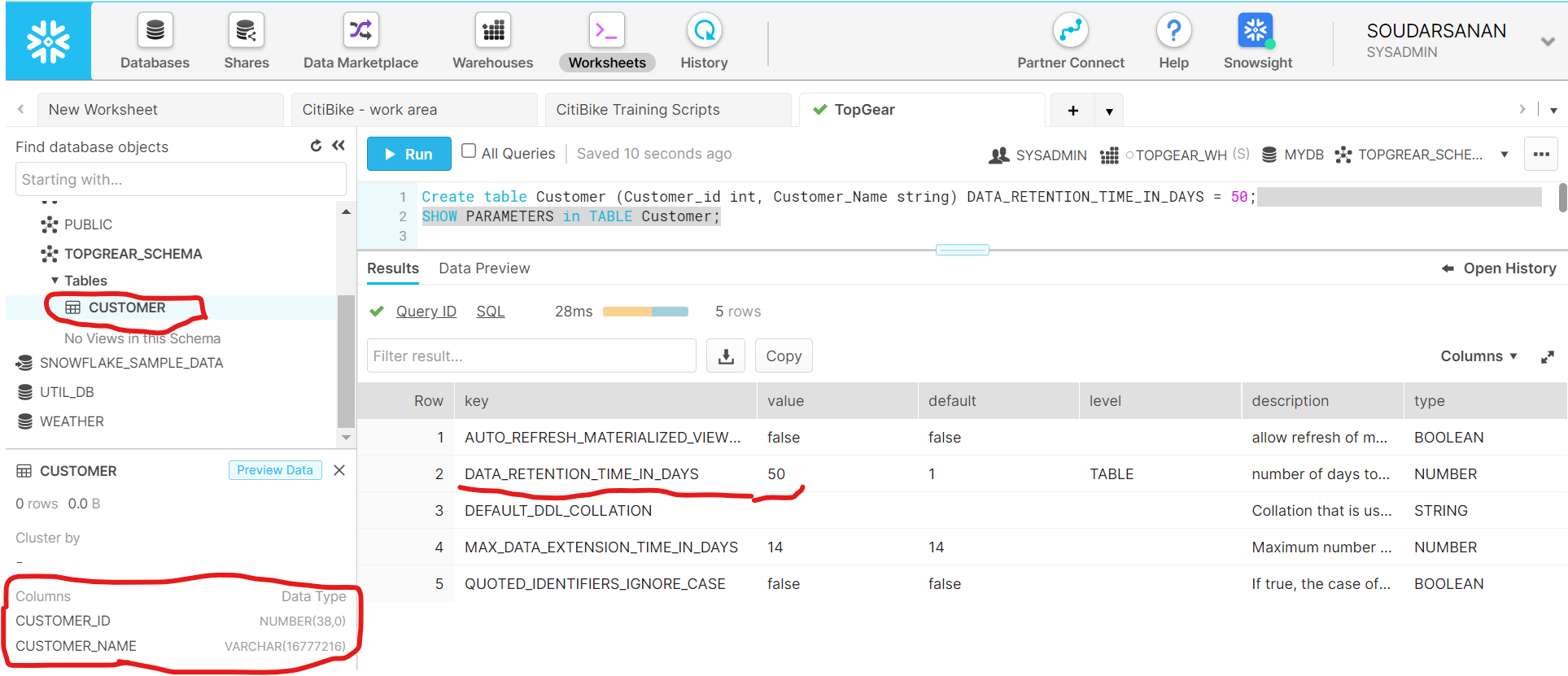
1. write statements to Create table customer with Customerid and Customer name columns with suitable data types and data retention of 50 days and then later use alter table command to change data retention in days to 20

**Queries**

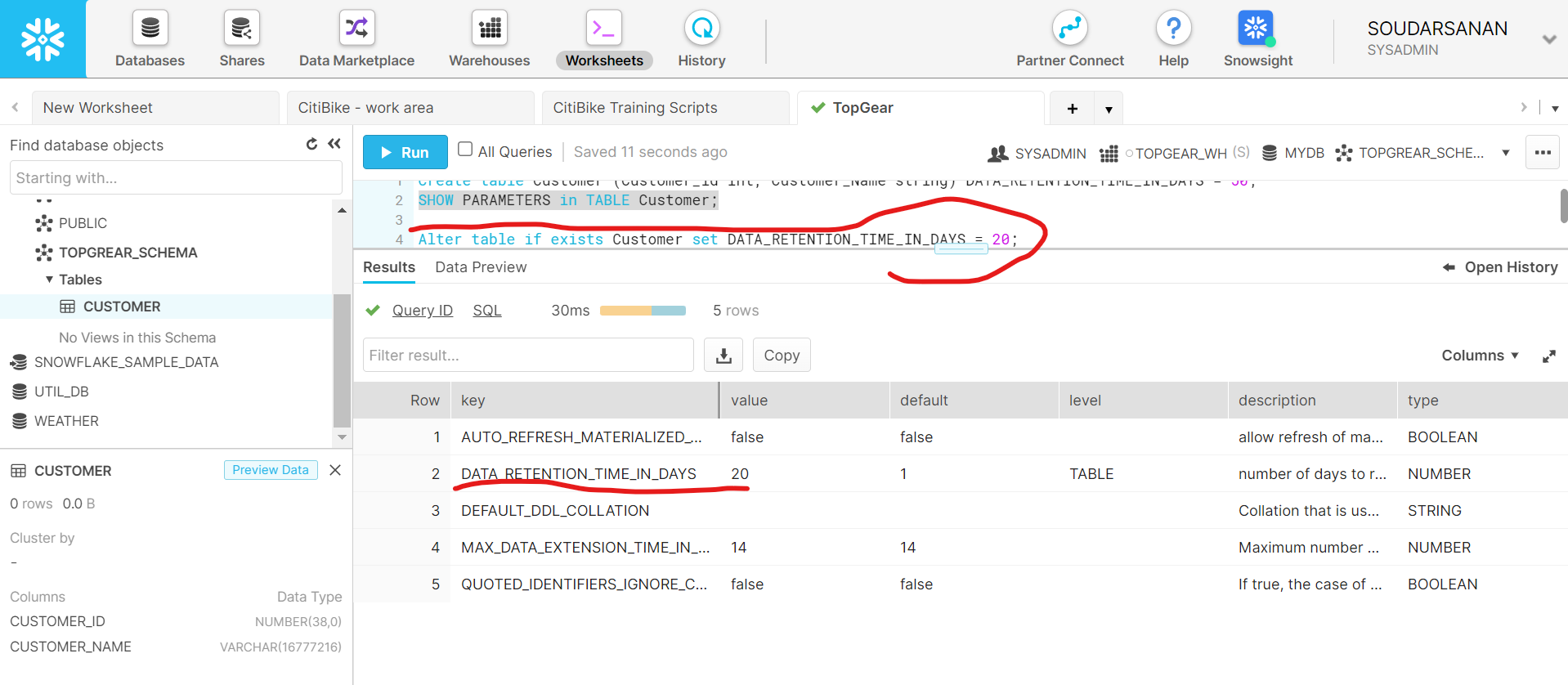
Create table Customer (Customer\_id int, Customer\_Name string) DATA\_RETENTION\_TIME\_IN\_DAYS = 50;

SHOW PARAMETERS in TABLE Customer;

**Screenshot 1** -> Table creation



Screenshot 2 -> Alter table – data retention for 20 days

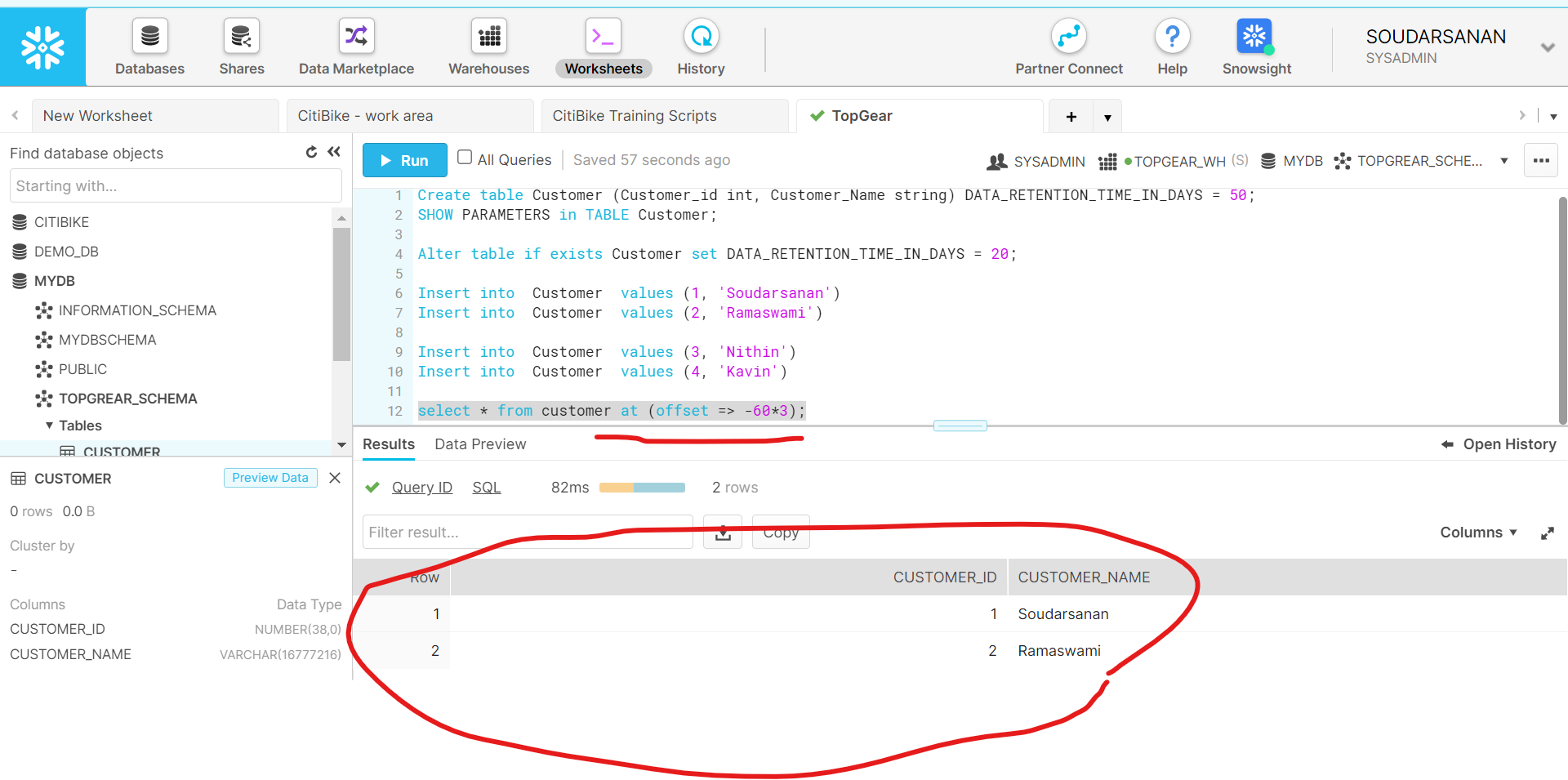


1. Insert Couple of customer records in above table and after 2 minutes insert another 2 records into customer table. Then execute query which selects historical data from table as of 3 minutes. How many records will be there in table?

**Queries**

Select \* from customer at (offset => -60\*3);

Snapshot 1 🡪 selects historical data from table as of 3 minutes 🡪 two records will be displayed as shown below

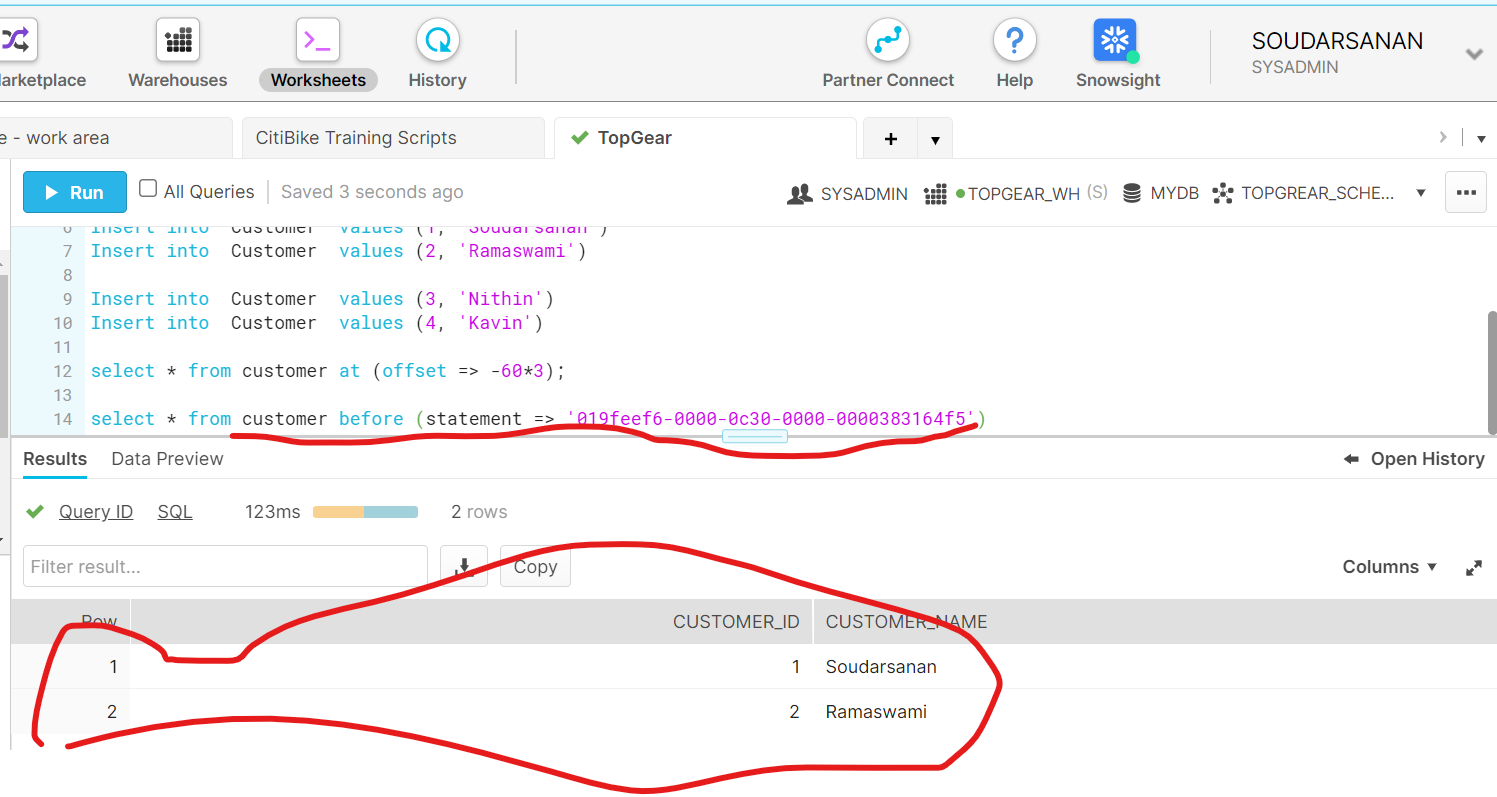


1. Write query that selects historical data from a customer table up to, but not including any changes made by the first insert statement of 2nd question above.

**Queries**

Select \* from customer before (statement => '019feef6-0000-0c30-0000-0000383164f5')

Snapshot 1: As below



**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?

**Queries**

Create database if not exists mydb;

Create schema if not exists mydb.myschema;

Create table department(departmentnumber int, departmentname string, location string);

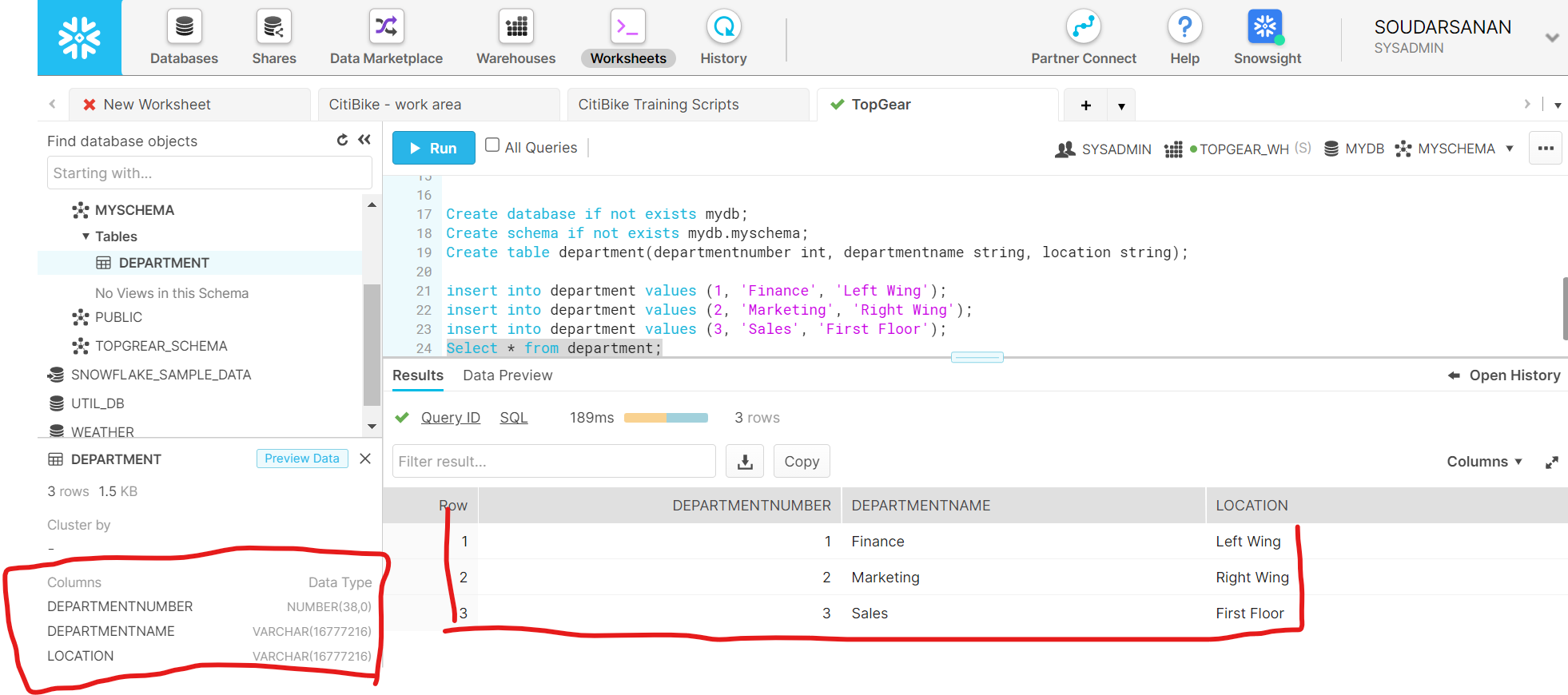
insert into department values (1, 'Finance', 'Left Wing');

insert into department values (2, 'Marketing', 'Right Wing');

insert into department values (3, 'Sales', 'First Floor');

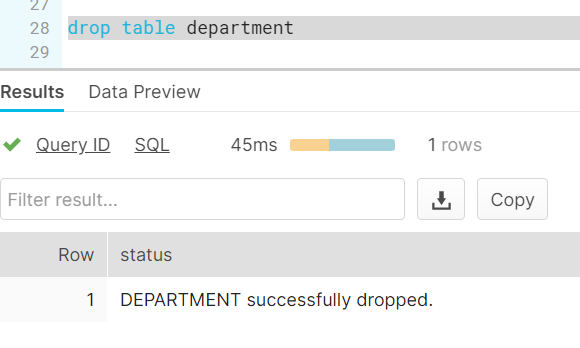
Select \* from department;

Snapshot1: Db/Schema/table creation and data insert



Snapshot2: table department dropped from mydb

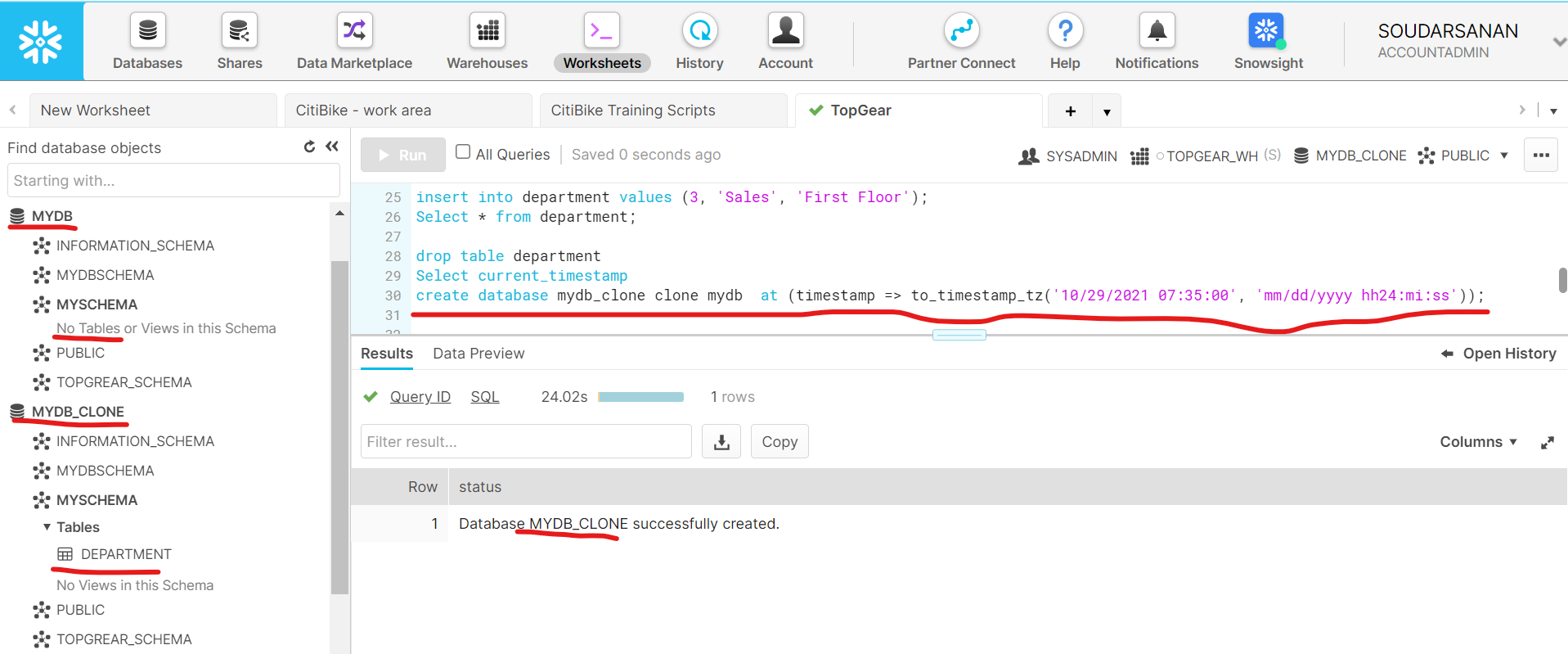
SQL: Drop table Department;



**Queries**

Create database mydb\_clone clone mydb at (timestamp => to\_timestamp\_tz('10/29/2021 07:35:00', 'mm/dd/yyyy hh24:mi:ss'));

Snapshot 3: mydb\_clone created with dropped table 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.

**Queries**

Create database Productdb;

Create schema ProductSchema;

Create table products (productid int, productname string, price decimal);

Insert into products values (1, 'Pen', 5);

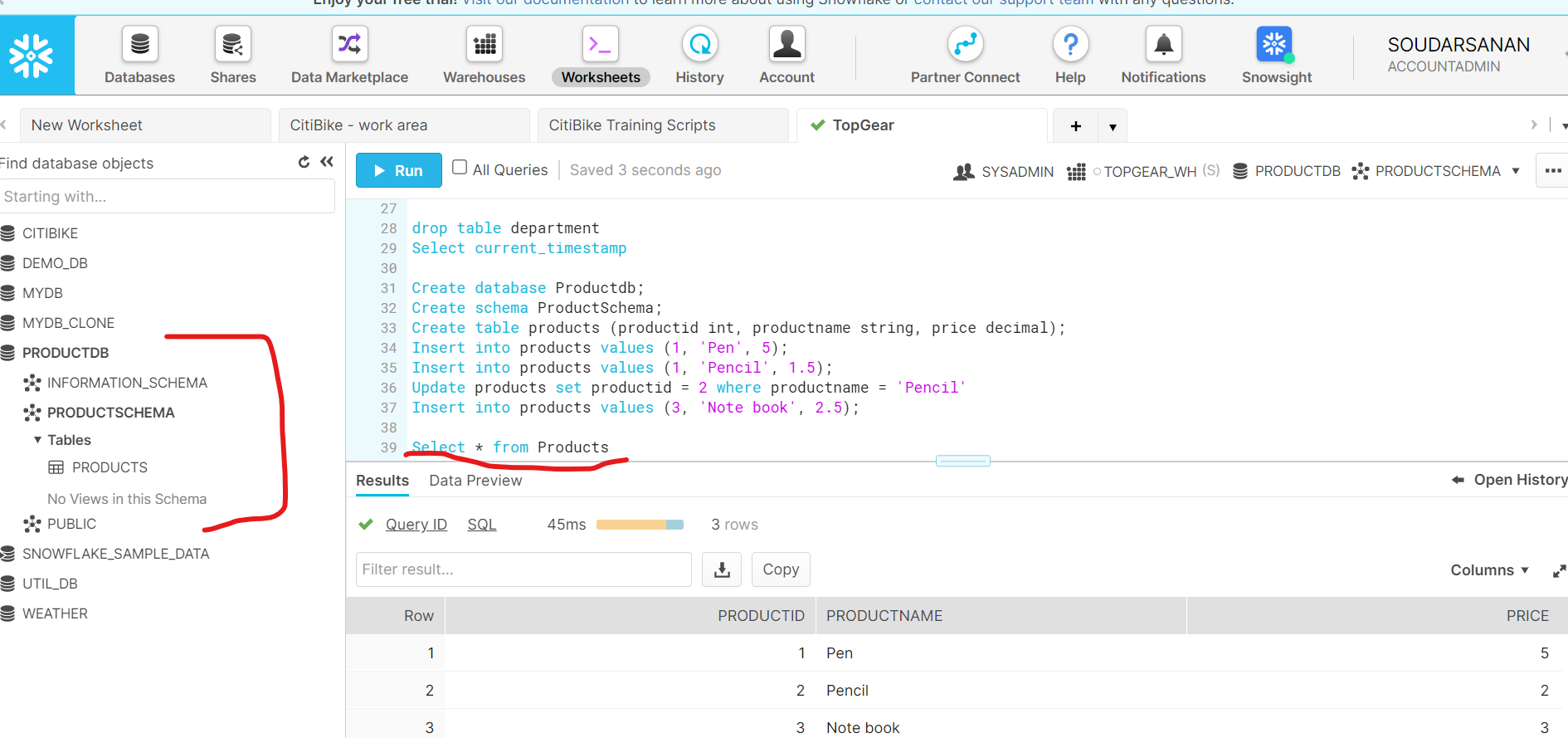
Insert into products values (1, 'Pencil', 1.5);

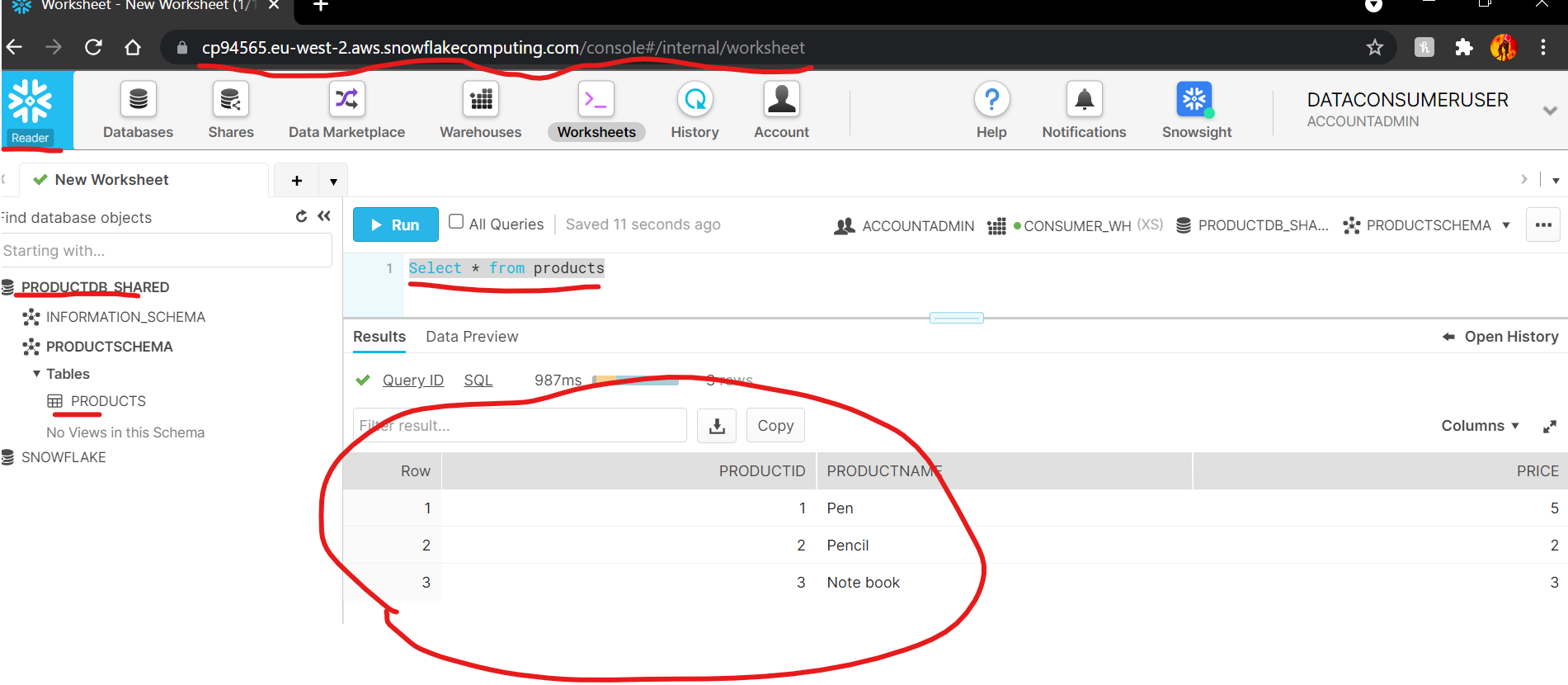
Update products set productid = 2 where productname = 'Pencil'

Insert into products values (3, 'Note book', 2.5);

Select \* from Products

**Snapshot1**: Provider 🡪 db, table & data





1. 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.

**Snapshot1**: Reader account not able to see Productid

