**-- Task 1 : Creating roles**

use role securityadmin;

create or replace role admin;

grant role admin to role accountadmin;

create or replace role PII;

grant role PII to role accountadmin;

create or replace role developer;

grant role developer to role admin;

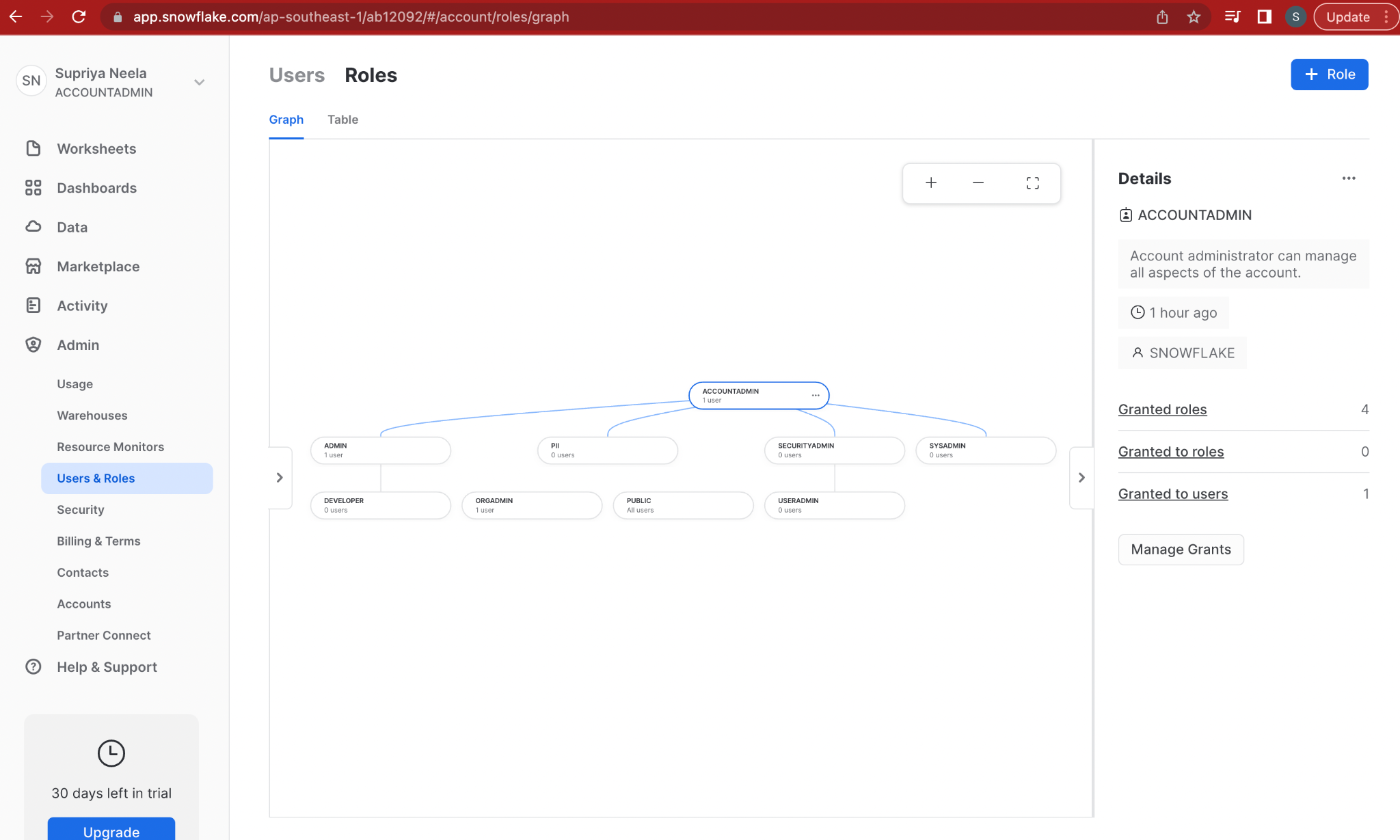
----- Granting permissions to admin role for creating database and schemas

use role securityadmin;

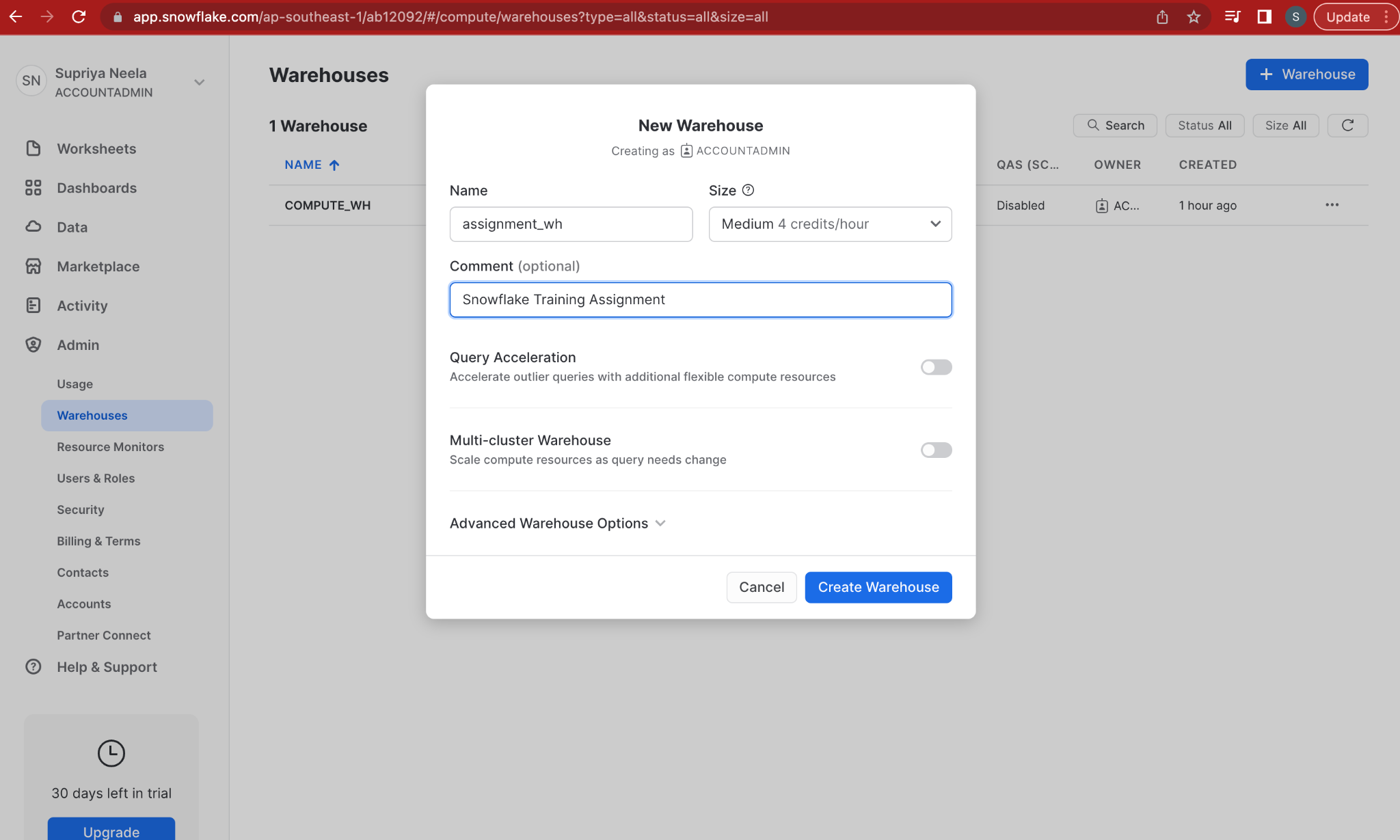
grant role admin to user supriyaneela;

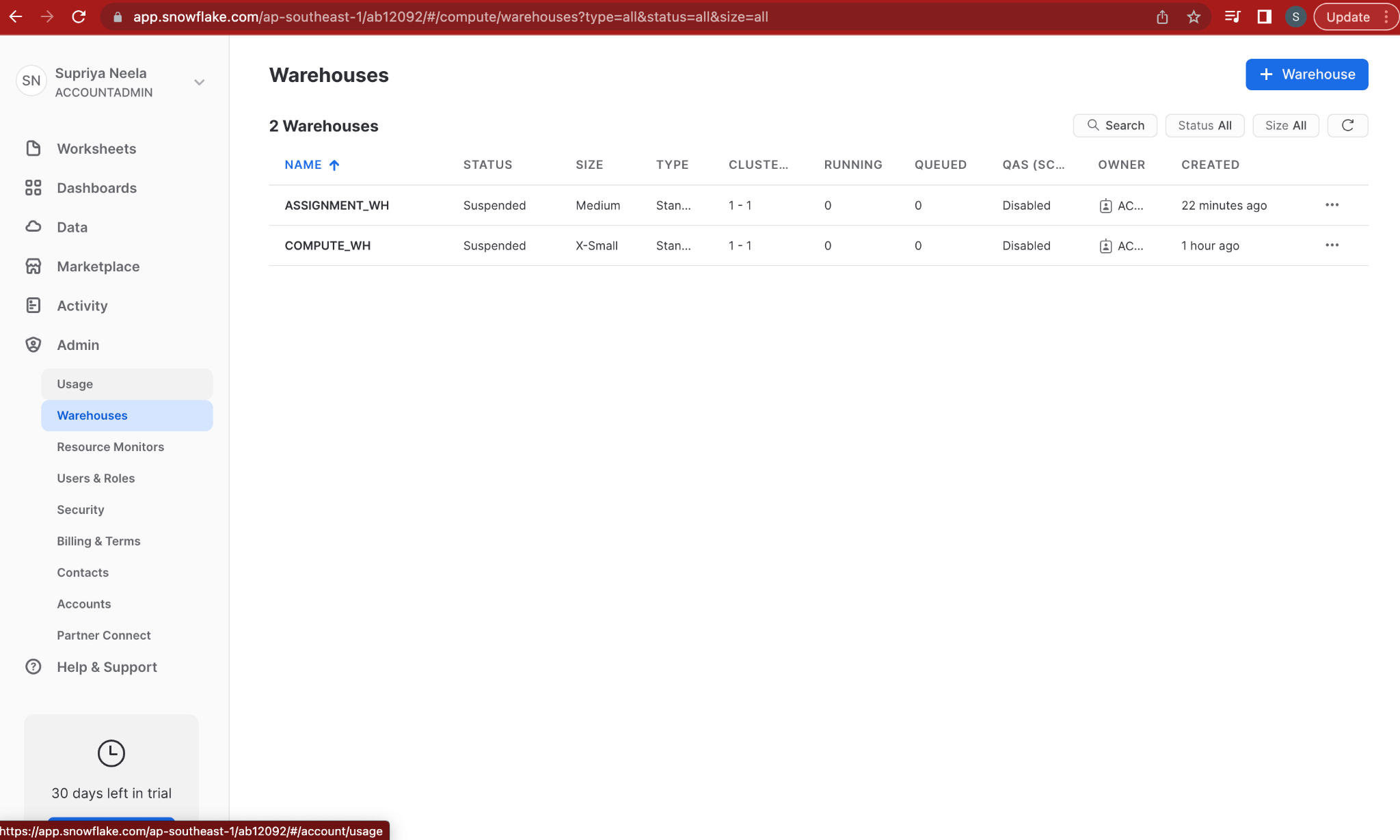
grant usage on warehouse assignment\_wh to role admin;

grant operate on warehouse assignment\_wh to role admin;



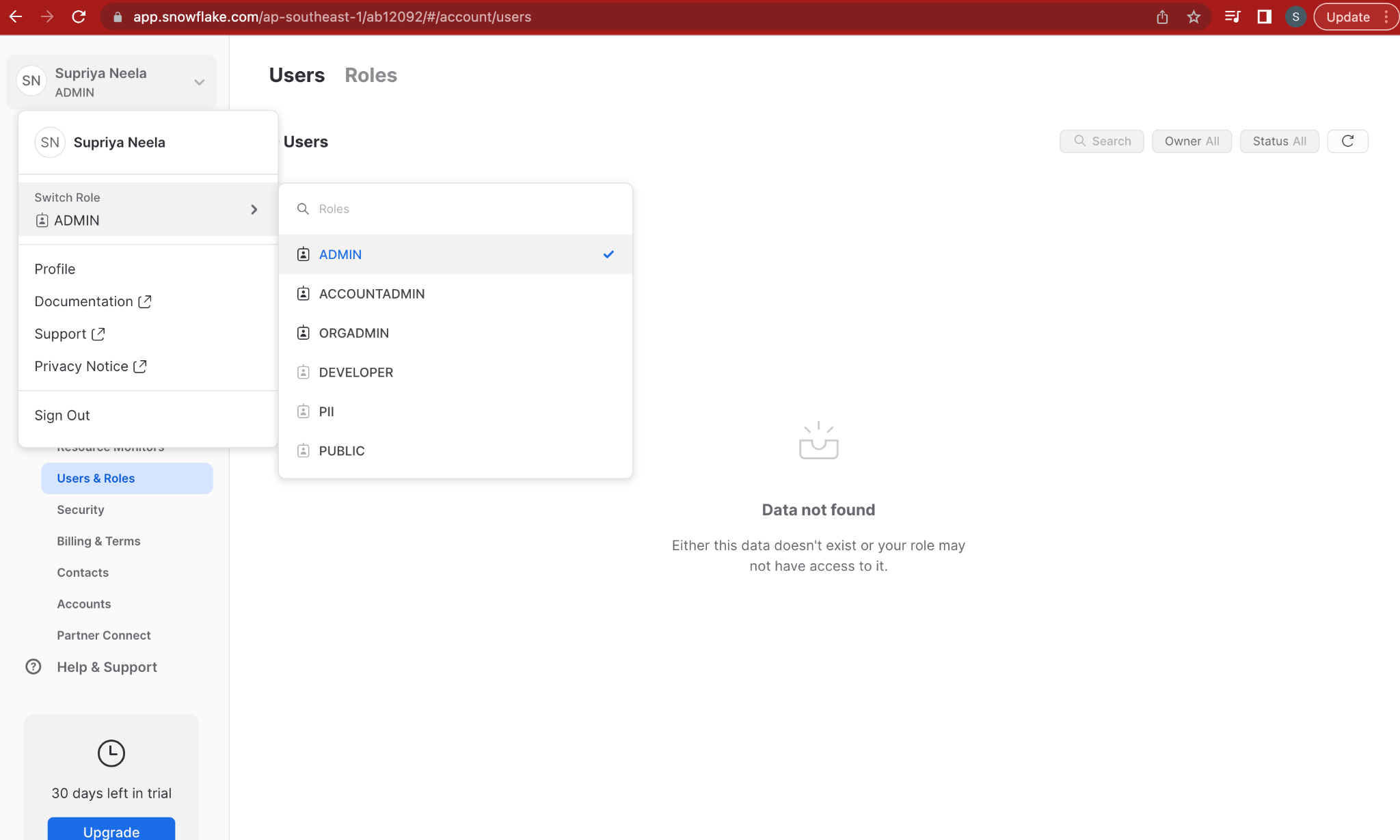
**-- Task 2 :Created warehouse assignment\_wh and granted access**





**-- Task 3 : Switched to admin account**

use role sysadmin;



**-- Task 4 : Create database assignment\_db**

grant create database on account to role admin;

use role admin;

use warehouse assignment\_wh;

create or replace database assignment\_db;

use database assignment\_db;

**-- Task 5 : Create a schema my\_schema**

create or replace schema my\_schema;

use schema my\_schema;

**-- Task 6 : Create sample employee table with audit columns elt\_ts,elt\_by,file\_name**

create or replace table employees(

EMPLOYEE\_ID integer PRIMARY KEY,

FIRST\_NAME string,

LAST\_NAME string,

EMAIL string,

PHONE\_NUMBER string,

HIRE\_DATE string,

JOB\_ID string,

SALARY integer,

MANAGER\_ID integer,

DEPARTMENT\_ID integer,

elt\_ts timestamp,

elt\_by string,

file\_name string

);

**--Task 7 : Create a variant of Employees table**

create or replace table employeesVariant(

EMPLOYEE\_ID integer PRIMARY KEY,

FIRST\_NAME string,

LAST\_NAME string,

EMAIL string,

PHONE\_NUMBER string,

HIRE\_DATE string,

JOB\_ID string,

SALARY integer,

MANAGER\_ID integer,

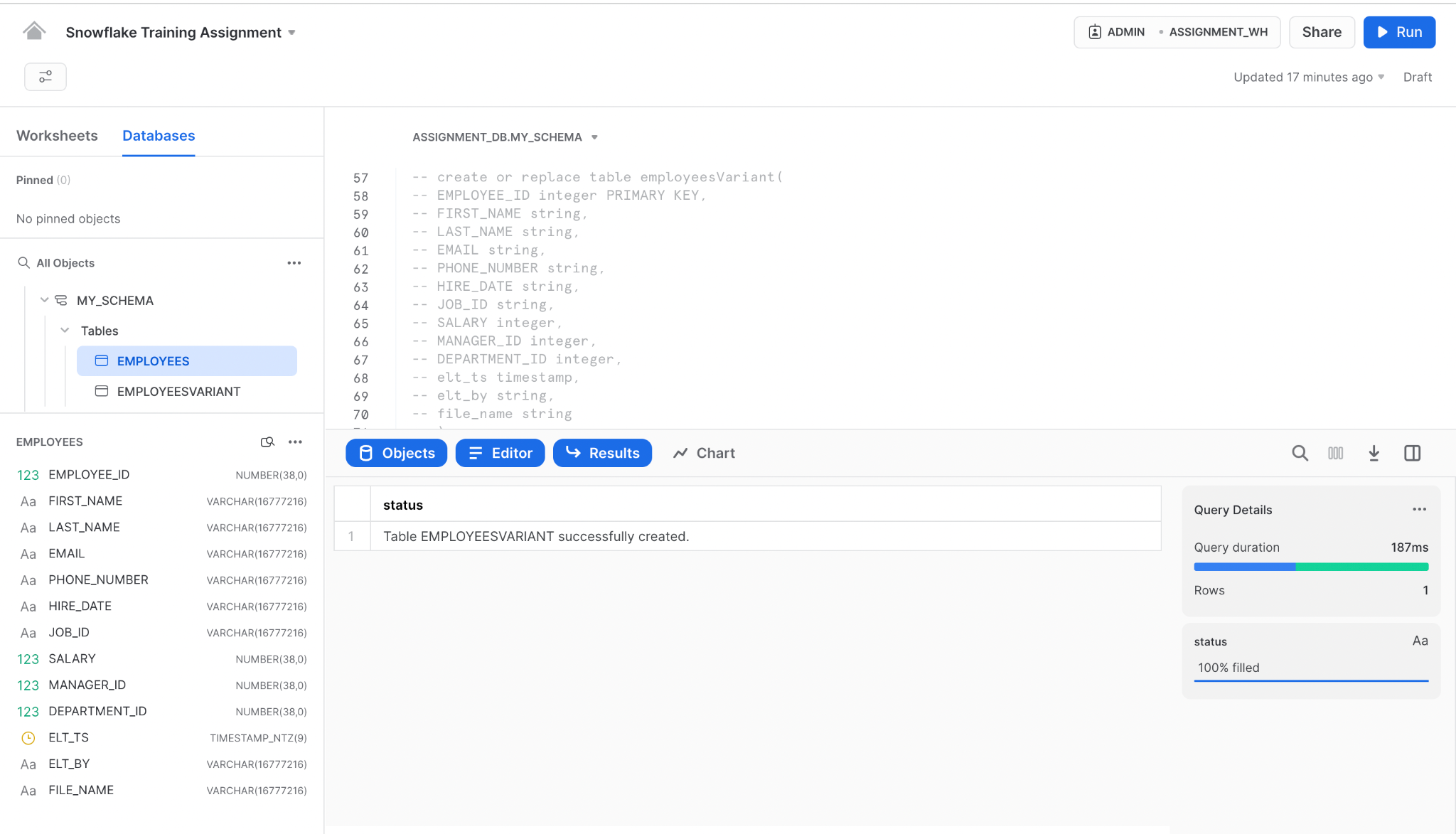
DEPARTMENT\_ID integer,

elt\_ts timestamp,

elt\_by string,

file\_name string

);



**-- Task 8 : Load the file into an external and internal stage**

create or replace file format r\_csv

TYPE = "CSV"

COMPRESSION = "GZIP"

FILE\_EXTENSION= 'csv.gz'

SKIP\_HEADER = 1

ERROR\_ON\_COLUMN\_COUNT\_MISMATCH=FALSE

EMPTY\_FIELD\_AS\_NULL = TRUE;

create or replace stage internal\_stage file\_format='r\_csv';

create or replace stage external\_stage

**-- Loading through internal stage**

**On local terminal execute ->** put file:///Users/saisupriya/Desktop/employee.csv @internal\_stage;

copy into employees from

(SELECT $1,$2,$3,$4,$5,$6,$7,$8,$10,$11,

CURRENT\_TIMESTAMP,

to\_varchar('local'),

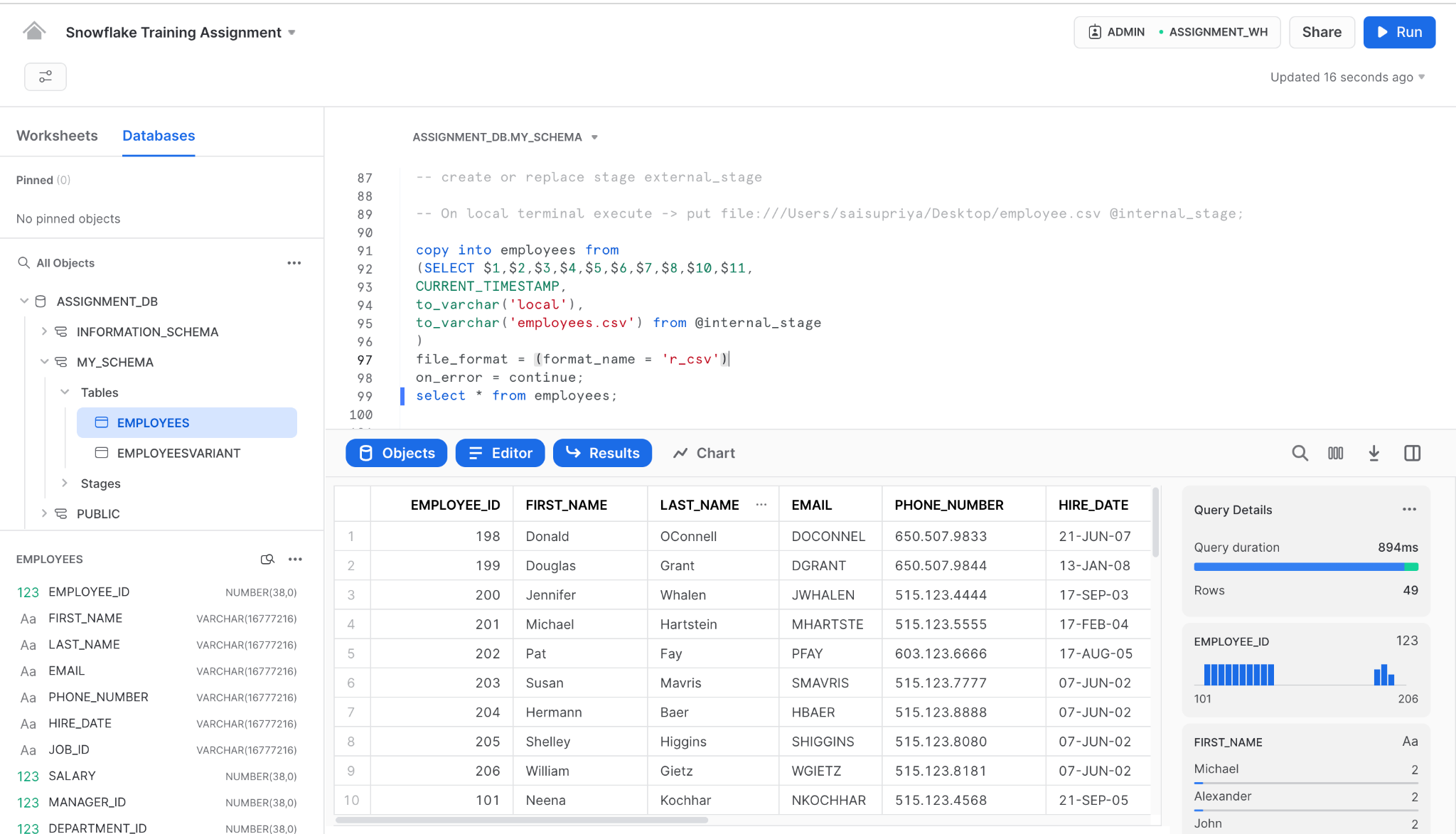
to\_varchar('employees.csv') from @internal\_stage

)

file\_format = (format\_name = 'r\_csv')

on\_error = continue;

select \* from employees;



**-- Loading through external stage**

1.Uploaded employee.csv file to aws S3 Bucket

2.Edited bucket policy

{

"Version": "2012-10-17",

"Id": "Policy1665674245003",

"Statement": [

{

"Sid": "Stmt1665674243203",

"Effect": "Allow",

"Principal": "\*",

"Action": "s3:GetObject",

"Resource": "arn:aws:s3:::snowflake-ass/\*"

}

]

}

create or replace stage external\_stage url='s3://snowflake-ass/';

copy into employeesVariant from

(SELECT $1,$2,$3,$4,$5,$6,$7,$8,$10,$11,CURRENT\_TIMESTAMP,

to\_varchar('AWS S3'),

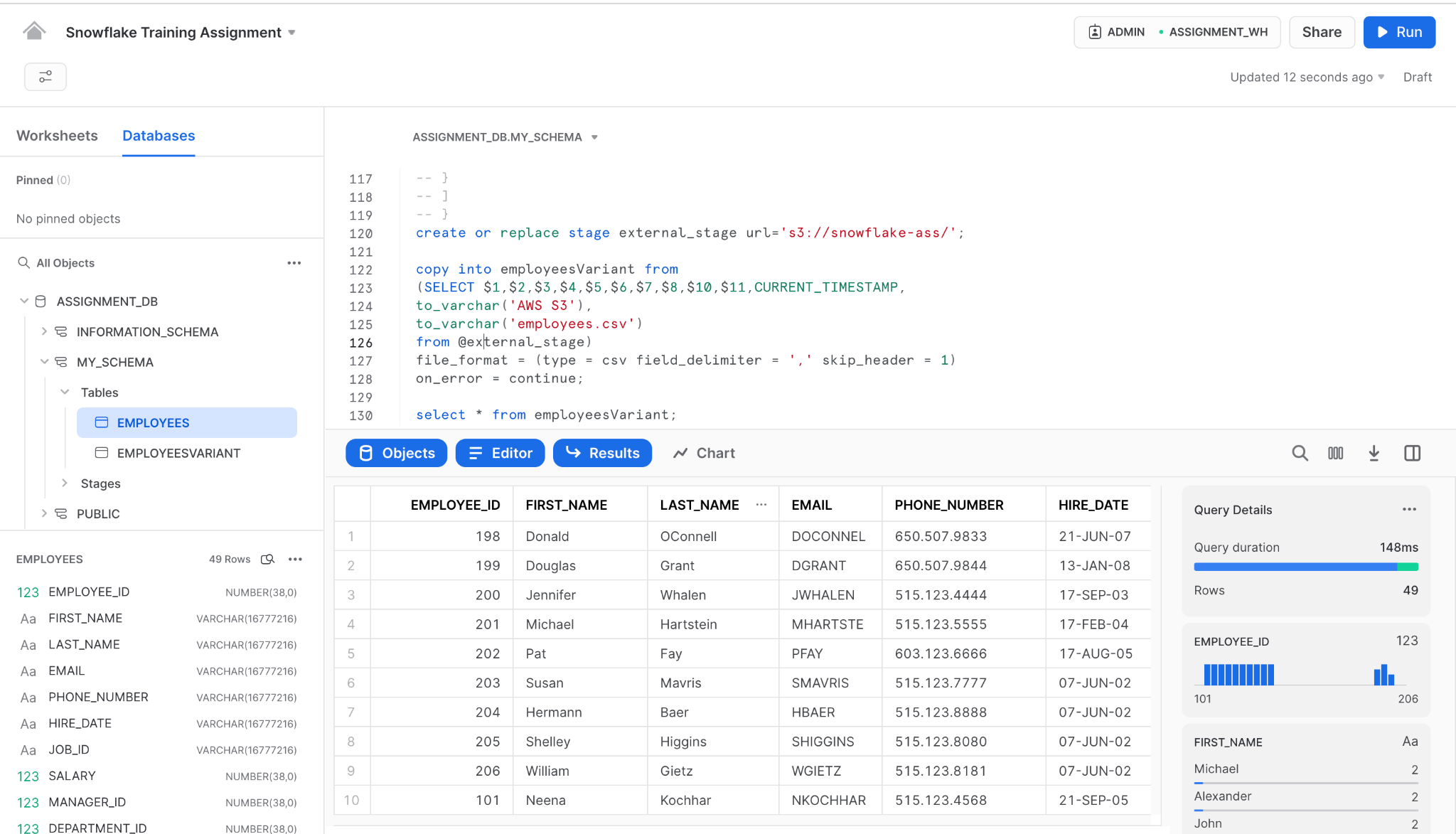
to\_varchar('employees.csv')

from @external\_stage)

file\_format = (type = csv field\_delimiter = ',' skip\_header = 1)

on\_error = continue;

select \* from employeesVariant;

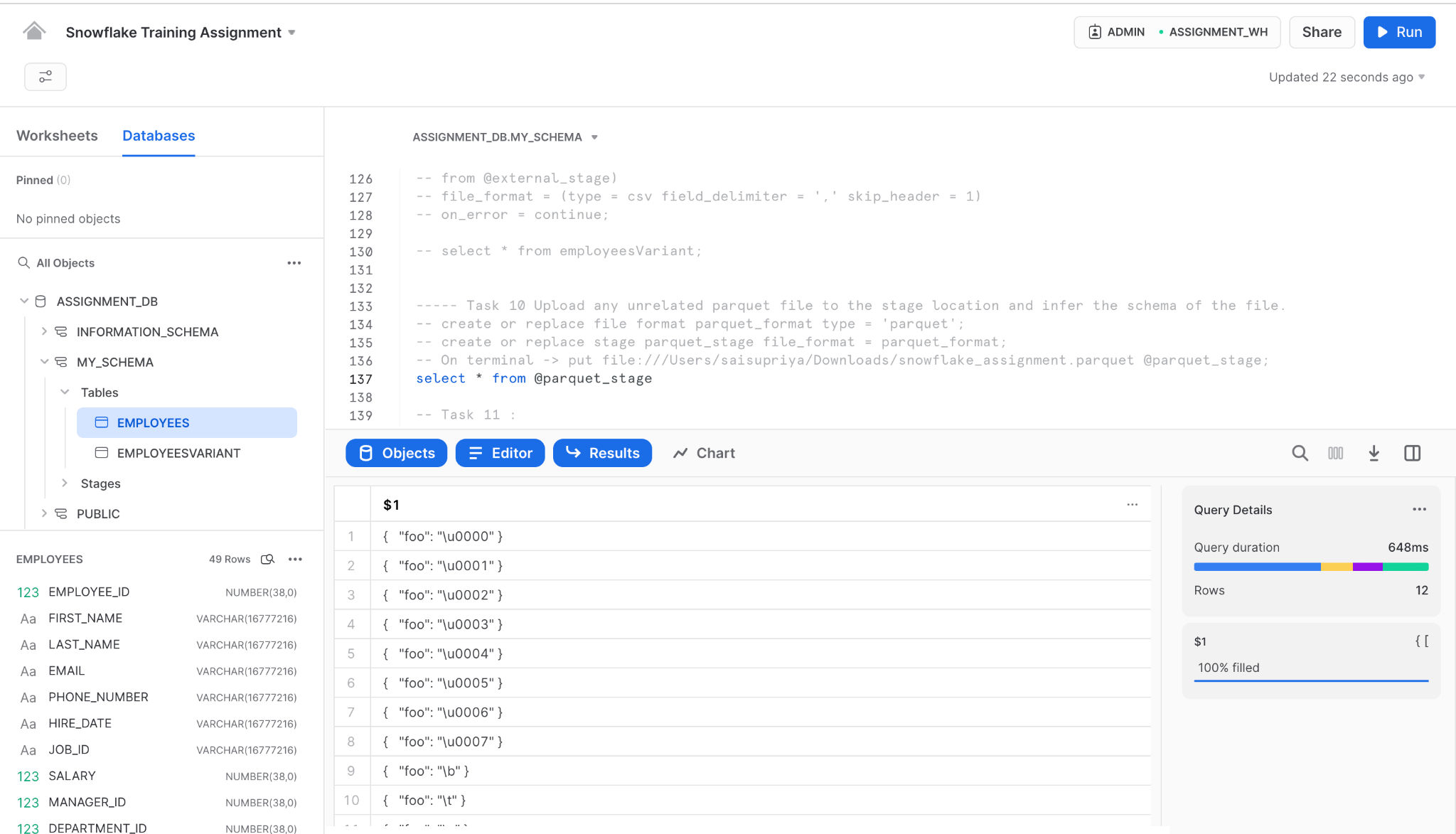


**--- Task 10 Upload any unrelated parquet file to the stage location and infer the schema of the file.**

create or replace file format parquet\_format type = 'parquet';

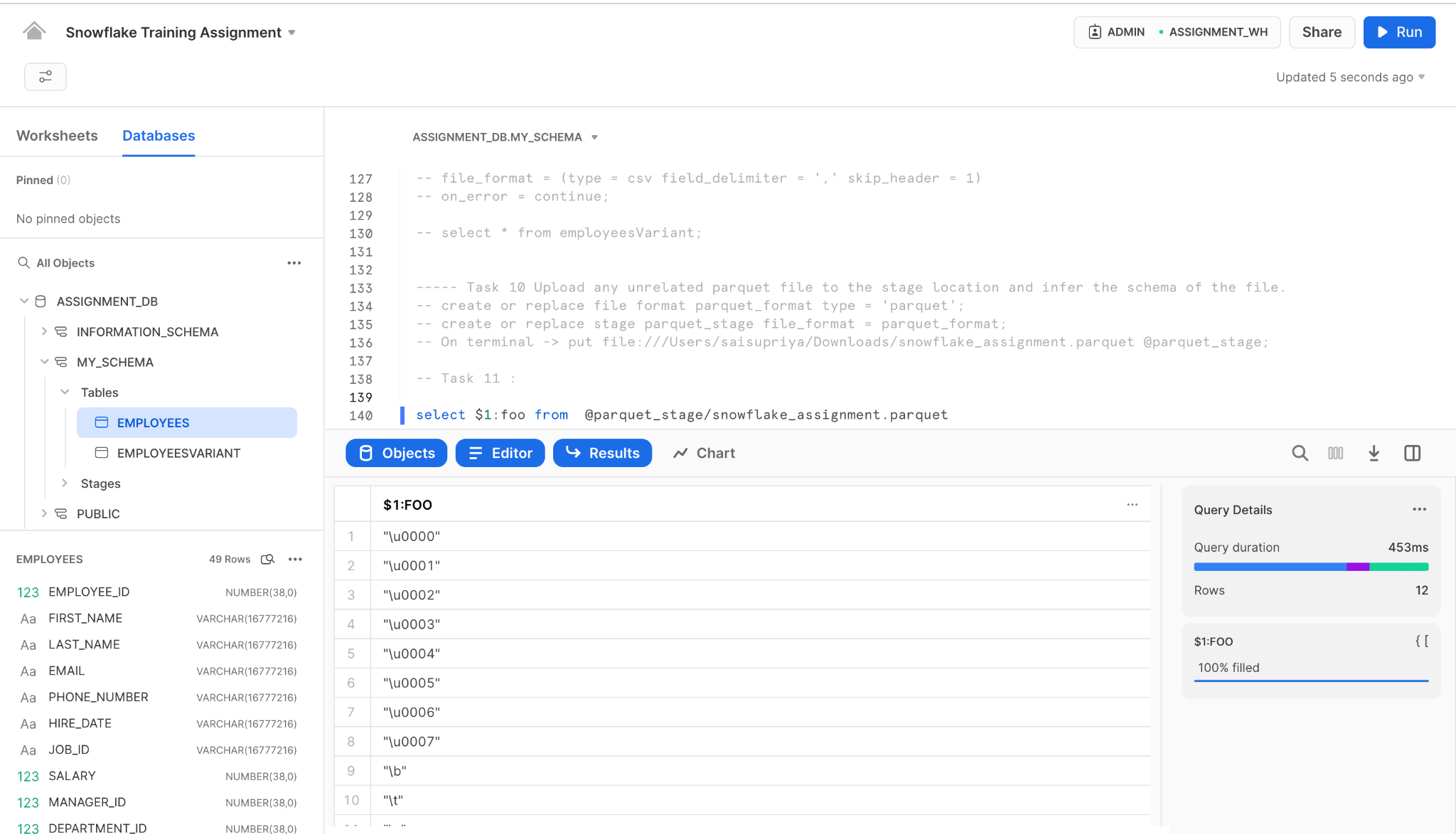
create or replace stage parquet\_stage file\_format = parquet\_format;

**On terminal ->** put file:///Users/saisupriya/Downloads/snowflake\_assignment.parquet @parquet\_stage;



**-- Task 11 : Run a select query on the staged parquet file without loading it to a snowflake table.**

select $1:foo from @parquet\_stage/snowflake\_assignment.parquet



**-- Task 12 : Masking Policy : If the role is PII the value of these columns should be visible.**

**Granting privileges for developer and PII roles**

use role securityadmin;

grant usage on database assignment\_db to role developer;

grant usage on schema assignment\_db.my\_schema to role developer;

grant usage on database assignment\_db to role PII;

grant usage on schema assignment\_db.my\_schema to role PII;

grant select on all tables in schema assignment\_db.my\_schema to role developer;

grant select on all tables in schema assignment\_db.my\_schema to role PII;

grant usage on warehouse assignment\_wh to role PII;

grant operate on warehouse assignment\_wh to role PII;

**Before masking all the details are visible to PII and developer**

use role admin;

use schema my\_schema;

create or replace masking policy PII\_email as (email string) returns string ->

case

when current\_role() in ('PII') then email

else '\*\*\*\*\*\*\*\*\*'

end;

create or replace masking policy PII\_phone\_number as (phone\_number string) returns

string ->

case

when current\_role() in ('PII') then phone\_number

else '\*\*\*\*\*\*\*\*\*'

end;

alter table if exists my\_schema.employees modify column email set masking policy

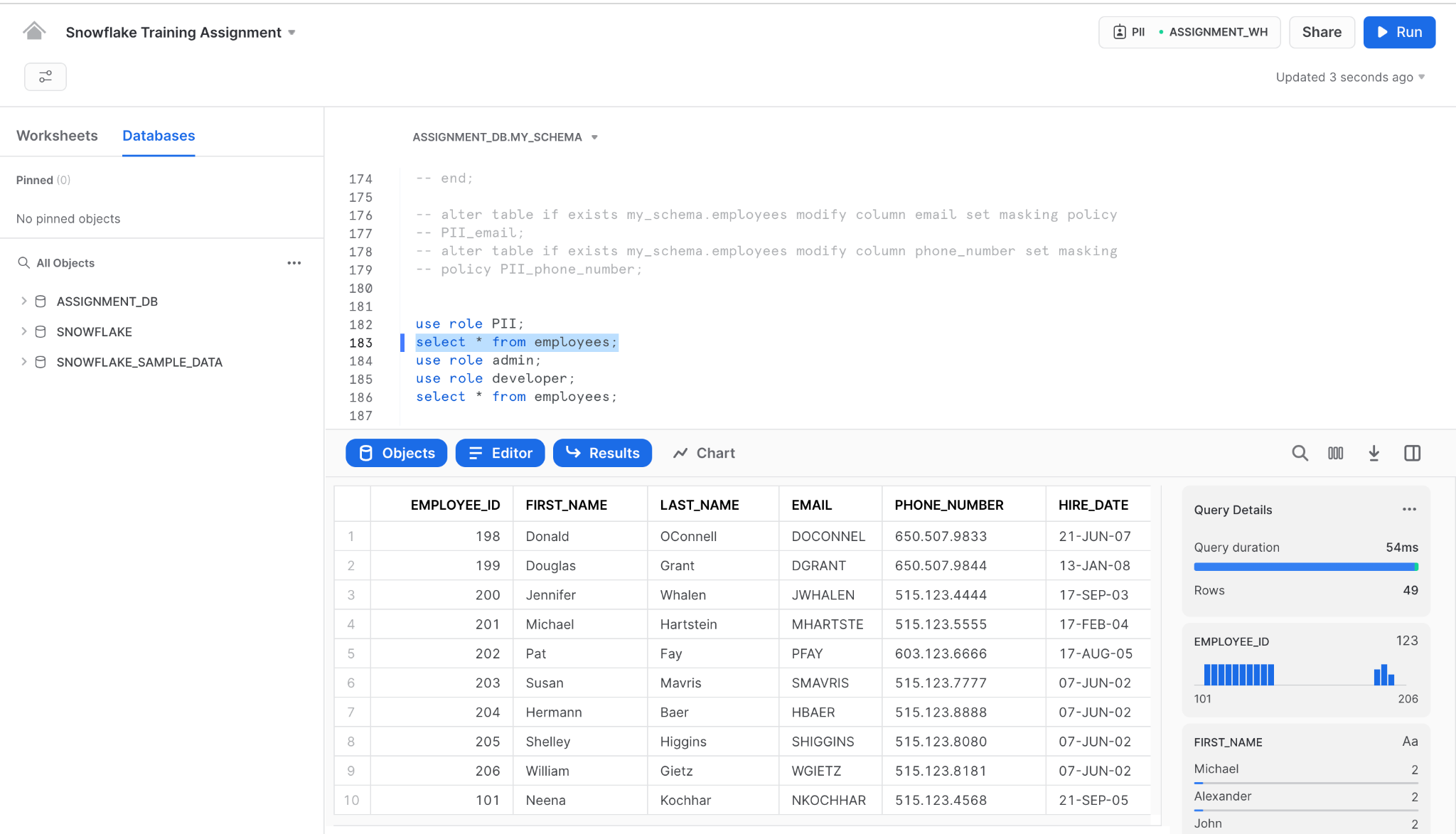
PII\_email;

alter table if exists my\_schema.employees modify column phone\_number set masking

policy PII\_phone\_number;

use role PII;

select \* from employees;



use role admin;

use role developer;

select \* from employees;

