```
use role developer;
use database debadatta_mohanty_db;
use warehouse training_wh;
create transient schema TRAINING_SCHEMA_CLONE;
use schema TRAINING_SCHEMA_CLONE;

--Active Bytes will show the storage
--snowflake.account_usage.table_storage_metrics
--information_schema.table_storage_metrics

-- Cloning Tables
-- Create a sample table
CREATE OR REPLACE TABLE employees
 (emp_id number,
 first_name varchar,
 last_name varchar
 );
-- Populate the table with some seed records.
Insert into employees
values(100,'John','Smith'),
 (200,'Sam','White'),
 (300,'Bob','Jones'),
 (400,'Linda','Carter');
-- Show the content of the table employees in the demo_db database and the public schema
select * from employees;

-- Create a clone of the table
CREATE OR REPLACE TABLE employees_clone
  CLONE employees;
```

-- Show the content of the clone which should be the same as the original table that it was cloned from.

```
select * from employees_clone;
```

```
select * from employees;
```

-- Add one more record to the clone table

```
insert into employees_clone values(500,'Mike','Jones');
```

```
insert into employees values(600,'Deb','Mohanty');
```

-- Verify the content of the clone table to show the original set of records and the additional new record that we just added

```
select * from employees_clone;
```

-- Verify the content of the original employees table. It should be the original content without the record we added to the clone.

```
select * from employees;
```

-- Yet another SQL to show all the records in the employees table but not in the employees clone table. The result should be just the new employee record of "Mike Jones" that was added to the employees_clone table.

```
select * from employees_clone

minus

select * from employees;
```

-- Yet another SQL to show all the records in the employees table but not in the employees clone table. The result should be just the new employee record of "Mike Jones" that was added to the employees_clone table.

```
select * from employees

minus
```

```sql
select * from employees_clone;


-- Add one new record to the original employee table to show all the records in the employees table but not in the employees clone table.  The result should be just the new employee record of "George Brown" that was added to the employees table.

insert into employees

values(600,'George','Brown');


select * from employees

minus

select * from employees_clone;


-- Delete emp_id 100 from the employees table.

delete from employees where emp_id = 100;


-- Check the content of the employees table.  The employee with Emp_id = 100 is no longer there

select * from employees;


-- Check the content of the employees_clone table.  The employee with Emp_id  = 100 is still there.

select * from employees_clone;




--  Cloning Databases


use role sysadmin;


-- Create a database clone (demo_db_clone) from the original database demo_db

CREATE or replace DATABASE demo_db_clone CLONE demo_db;
```

```
-- Point to the demo_db_clone database

use database demo_db_clone;


-- Drop the table employees from the demo_db_clone database

drop table demo_db_clone.public.employees;


-- Drop the table employees_clone from the demo_db_clone database

drop table demo_db_clone.public.employees_clone;


-- Showing tables in the demo_db_clone would show neither the employees table nor the
employees_clone table.  They are gone.

show tables;


-- Point to the original demo_db database

use demo_db;


-- Showing the tables in the original demo_db database will show all the tables in that database
including the employees and employees_clone tables

show tables;


-- Cloning Schema

-- Create a cloned schema from the original public schema

CREATE or replace SCHEMA "DEMO_DB".public_clone CLONE
"DEMO_DB"."PUBLIC";"DEMO_DB"."PUBLIC_CLONE"


CREATE OR REPLACE DATABASE DEMO_DB_CLONE CLONE DEMO_DB;


-- Point to the public schema

use schema public;
```

-- Show tables in the original public schema. That should return all the tables in the public schema including a table called employees;

```
show tables;
```

-- Point to the public_clone schema

```
use schema public_clone;
```

-- Show tables in the public_clone schema

```
show tables;
```

-- Drop the table employees_clone in the public_clone schema;

```
drop table public_clone.employees_clone;
```

-- Show tables in the public_clone schema. The result of that command should not have in it a table called employees_clone;

```
show tables;
```

-- Point to the original public schema

```
use schema public;
```

-- The result of that command should include in it the table called employees_clone since it was dropped from the clone schema and not the original public schema.

```
show tables;
```