

Assignment 1: Snowflake Environment Setup (Beginner)

Objective: Understand Snowflake UI and basic objects.

Tasks

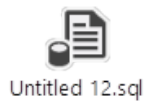
1. Log in to Snowflake (trial account).
2. Create:
 - A **Database**
 - A **Schema**
 - A **Virtual Warehouse**
3. Show all databases and schemas.
4. Suspend and resume the warehouse.

Deliverable:

- SQL script
- Screenshot of created objects

Answer:

- SQL script:



- Screenshot of created objects:

```
CREATE OR REPLACE DATABASE MY_DB;
```

Results (just now)	
Table	Chart
1 row	121ms
id	status
1	Database MY_DB successfully created.

```
CREATE OR REPLACE SCHEMA MY_DB.MY_SCHEMA;
```

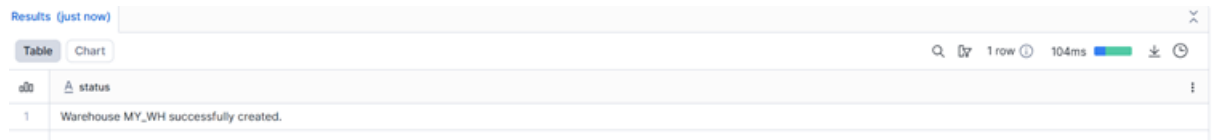
Results (just now)	
Table	Chart
1 row	51ms
id	status
1	Schema MY_SCHEMA successfully created.

CREATE OR REPLACE **WAREHOUSE** MY_WH

WITH WAREHOUSE_SIZE = 'XSMALL'

AUTO_SUSPEND = 60

AUTO_RESUME = TRUE;



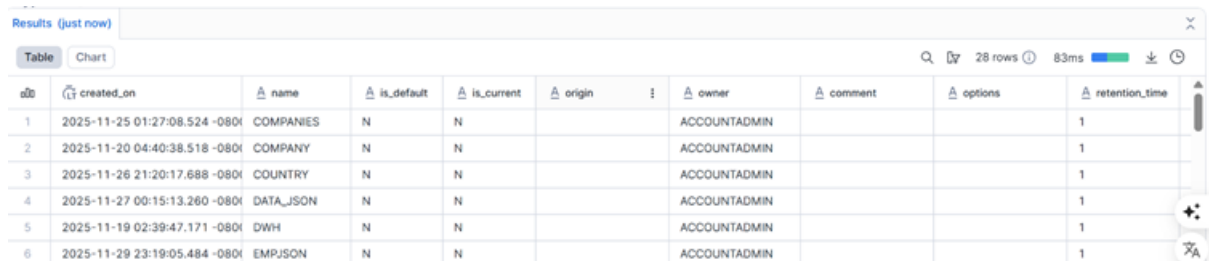
Results (just now)

Table Chart 1 row 104ms

id	status
1	Warehouse MY_WH successfully created.

3. Show all databases and schemas.

SHOW DATABASES;

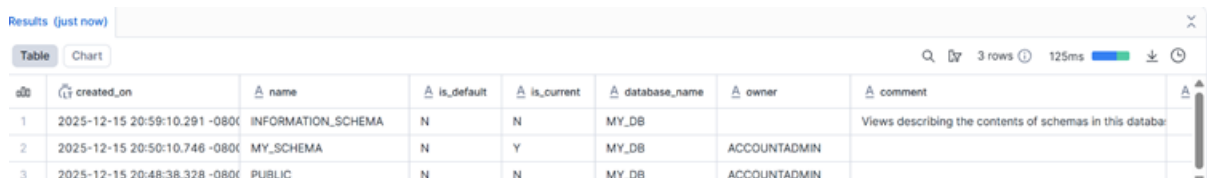


Results (just now)

Table Chart 28 rows 83ms

id	created_on	name	is_default	is_current	origin	owner	comment	options	retention_time
1	2025-11-25 01:27:08.524 -0800	COMPANIES	N	N		ACCOUNTADMIN			1
2	2025-11-20 04:40:38.518 -0800	COMPANY	N	N		ACCOUNTADMIN			1
3	2025-11-26 21:20:17.688 -0800	COUNTRY	N	N		ACCOUNTADMIN			1
4	2025-11-27 00:15:13.260 -0800	DATA_JSON	N	N		ACCOUNTADMIN			1
5	2025-11-19 02:39:47.171 -0800	DWH	N	N		ACCOUNTADMIN			1
6	2025-11-29 23:19:05.484 -0800	EMPJSON	N	N		ACCOUNTADMIN			1

SHOW SCHEMAS IN DATABASE MY_DB;



Results (just now)

Table Chart 3 rows 125ms

id	created_on	name	is_default	is_current	database_name	owner	comment
1	2025-12-15 20:59:10.291 -0800	INFORMATION_SCHEMA	N	N	MY_DB		Views describing the contents of schemas in this database
2	2025-12-15 20:50:10.746 -0800	MY_SCHEMA	N	Y	MY_DB	ACCOUNTADMIN	
3	2025-12-15 20:48:38.328 -0800	PUBLIC	N	N	MY_DB	ACCOUNTADMIN	

4.Suspend and resume the warehouse.

ALTER WAREHOUSE MY_WH RESUME;



Results (just now)

Table Chart 1 row 662ms

id	status
1	Statement executed successfully.

ALTER WAREHOUSE MY_WH SUSPEND;

Results (just now)	
Table	Chart
1 row	67ms
id	status
1	Statement executed successfully.

Assignment 2: User, Role & Access Control (RBAC)

Objective: Learn Snowflake security and role hierarchy.

Tasks

- Create roles:
 - STUDENT_ROLE
 - INSTRUCTOR_ROLE
- Create users:
 - student1
 - instructor1
- Assign:
 - STUDENT_ROLE → student1
 - INSTRUCTOR_ROLE → instructor1
- Grant permissions:
 - Student → **SELECT** on tables
 - Instructor → **SELECT, INSERT**
- Verify access using SHOW GRANTS.

Deliverable:

- SQL commands
- Short explanation of RBAC

Answer:

```
CREATE OR REPLACE ROLE STUDENT_ROLE;
```

Results (just now)	
Table	Chart
1 row	87ms
id	status
1	Role STUDENT_ROLE successfully created.

```
CREATE OR REPLACE ROLE INSTRUCTOR_ROLE;
```

Results (15 minutes ago)

Table		Chart	1 row	85ms
id	status			
1	Role INSTRUCTOR_ROLE successfully created.			

CREATE OR REPLACE USER student1

PASSWORD = 'Student@123'

DEFAULT_ROLE = STUDENT_ROLE

MUST_CHANGE_PASSWORD = TRUE;

Results (just now)

Table		Chart	1 row	1.1s
id	status			
1	User STUDENT1 successfully created.			

CREATE OR REPLACE USER instructor1

PASSWORD = 'Instructor@123'

DEFAULT_ROLE = INSTRUCTOR_ROLE

MUST_CHANGE_PASSWORD = TRUE;

Results (just now)

Table		Chart	1 row	369ms
id	status			
1	User INSTRUCTOR1 successfully created.			

Assign:

GRANT ROLE STUDENT_ROLE TO USER student1;

Results (just now)

Table		Chart	1 row	55ms
id	status			
1	Statement executed successfully.			

GRANT ROLE INSTRUCTOR_ROLE TO USER instructor1;

Results (just now)	
Table	Chart
1	status
1	Statement executed successfully.

Grant permissions:

GRANT SELECT ON ALL TABLES IN SCHEMA MY_DB.MY_SCHEMA

TO ROLE STUDENT_ROLE;

Results (just now)	
Table	Chart
1	status
1	Statement executed successfully. 0 objects affected.

GRANT SELECT, INSERT ON ALL TABLES IN SCHEMA MY_DB.MY_SCHEMA

TO ROLE INSTRUCTOR_ROLE;

Results (just now)	
Table	Chart
1	status
1	Statement executed successfully. 0 objects affected.

Assignment 3: Table Creation & Data Loading

Objective: Work with structured data.

Tasks

1. Create a table:

CREATE TABLE students (

```
id INT,  
  
name STRING,  
  
course STRING,  
  
score INT  
  
);
```

2. Insert at least **10 records**.
3. Query:
 - All students
 - Students with score > 80
4. Update one record.
5. Delete one record.

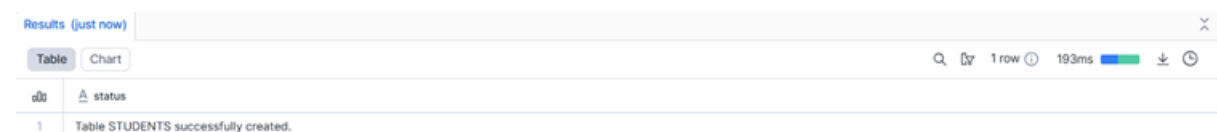
Deliverable:

- SQL file with queries
- Output screenshots

Answer:

CREATE OR REPLACE TABLE students (

```
id INT,  
  
name STRING,  
  
course STRING,  
  
score INT  
  
);
```



The screenshot shows a database interface with a header bar that says "Results (just now)". Below the header, there are two tabs: "Table" and "Chart". The "Table" tab is selected. The table has two columns: "id" and "status". The first row of the table contains the value "1" in the "id" column and "Table STUDENTS successfully created." in the "status" column. The interface also shows a search icon, a filter icon, and a status bar indicating "1 row" and "193ms".

id	status
1	Table STUDENTS successfully created.

INSERT INTO students VALUES

```
(1, 'Ravi', 'SQL', 85),  
(2, 'Anu', 'Python', 78),  
(3, 'Kiran', 'Snowflake', 92),  
(4, 'Meena', 'SQL', 88),  
(5, 'Arun', 'Python', 65),
```

```
(6, 'Divya', 'Snowflake', 95),
(7, 'Suresh', 'SQL', 72),
(8, 'Pooja', 'Python', 81),
(9, 'Vijay', 'Snowflake', 90),
(10, 'Neha', 'SQL', 76);
```

Results (just now)

Table Chart

1 row 1.3s

#	# number of rows inserted
1	10

1. select * from students;

Results (just now)

Table Chart

10 rows 274ms

#	ID	NAME	COURSE	SCORE
1	2	Anu	Python	78
2	3	Kiran	Snowflake	92
3	5	Arun	Python	65
4	6	Divya	Snowflake	95
5	8	Pooja	Python	81
6	9	Vijay	Snowflake	90
7	1	Ravi	java	85
8	4	Meena	java	88

2. select * from students where score > 80;

Results (just now)

Table Chart

6 rows 27ms

#	ID	NAME	COURSE	SCORE
1	3	Kiran	Snowflake	92
2	6	Divya	Snowflake	95
3	8	Pooja	Python	81
4	9	Vijay	Snowflake	90
5	1	Ravi	java	85
6	4	Meena	java	88

3. update students set course = 'java' where course = 'SQL';


Results (just now)

Table Chart

1 row 426ms

#	# number of rows updated	# number of multi-joined rows updated
1	0	0

4. delete from students where name = 'Ravi';



The screenshot shows a Snowflake query results window titled "Results (just now)". It has tabs for "Table" and "Chart". The "Table" tab is active, showing a single row with the value "1" under the column "# number of rows deleted". The top right of the window displays "1 row" and "707ms".

number of rows deleted
1

Assignment 4: File Loading using Stage & COPY (Intermediate)

Objective: Learn Snowflake data ingestion.

Tasks

1. Create an **internal stage**.
2. Upload a CSV file.
3. Use COPY INTO to load data.
4. Validate loaded records.
5. Handle one data error using VALIDATION_MODE.

Deliverable:

- SQL script
- CSV file
- Query output

Answer;

1. Create an **internal stage**.

create or replace stage my_stage;



The screenshot shows a Snowflake query results window titled "Results (just now)". It has tabs for "Table" and "Chart". The "Table" tab is active, showing a single row with the status "Stage area MY_STAGE successfully created." under the column "status". The top right of the window displays "1 row" and "174ms".

status
Stage area MY_STAGE successfully created.

2. Upload a CSV file.

Upload Your Files
1 file → MY_STAGE

new_employee.csv - 475.0B

Select database, schema and stage
Schema: MY_DB.MY_SCHEMA Stage: MY_STAGE

Specify the path to an existing folder or create a new one (optional)
Example / Path / To / Folder

Once the files finish uploading, your Directory Table will automatically refresh.

Close Upload

3. Use COPY INTO to load data.

COPY INTO students

FROM @my_stage/csv_flatfile.csv

FILE_FORMAT = (

TYPE = 'CSV'

SKIP_HEADER = 1

);

Results (just now)	
Table	Chart
1	Copy executed with 0 files processed.

4. Validate loaded records.

select count(*) from students;

Results (just now)	
Table	Chart
1	9

5. Handle one data error using VALIDATION_MODE.

I upload error file in snowflake after i copy into that.Finally i got error msg like this.

```
242 COPY INTO emp_temp
243 FROM @my_stage/new_emp.csv
244 FILE_FORMAT = (
245   TYPE = 'CSV'
246   SKIP_HEADER = 1
247 )
248 VALIDATION_MODE = 'RETURN_ERRORS';
249
250
```

Results (just now)

Table Chart 10 rows 1.3s

#	ERROR	FILE	# LINE	# CHARACTER	# BYTE-OFFSET	CATEGORY	# CODE	SQL-STATE	COLUMN
1	Number of columns in file (5) does not match that of the co	new_emp.csv	3	1	64	parsing	100080	22000	"EMP_TEN
2	Number of columns in file (5) does not match that of the co	new_emp.csv	4	1	100	parsing	100080	22000	"EMP_TEN
3	Number of columns in file (5) does not match that of the co	new_emp.csv	5	1	138	parsing	100080	22000	"EMP_TEN
4	Number of columns in file (5) does not match that of the co	new_emp.csv	6	1	176	parsing	100080	22000	"EMP_TEN
5	Number of columns in file (5) does not match that of the co	new_emp.csv	7	1	212	parsing	100080	22000	"EMP_TEN

Assignment 6: Warehouses & Performance

Objective: Learn compute management.

Tasks

- Create:
 - SMALL_WH
 - MEDIUM_WH
- Run same query on both warehouses.
- Compare execution time.
- Suspend unused warehouse.

Deliverable:

- Query history screenshot
- Performance comparison notes

CREATE OR REPLACE WAREHOUSE **SMALL_WH**

WAREHOUSE_SIZE = '**SMALL**'

AUTO_SUSPEND = 60

AUTO_RESUME = TRUE

INITIALLY_SUSPENDED = TRUE;

Results (just now)	
Table	Chart
1 row	105ms
status	
1	Warehouse SMALL_WH successfully created.

CREATE OR REPLACE WAREHOUSE **MEDIUM_WH**

WAREHOUSE_SIZE = '**MEDIUM**'

AUTO_SUSPEND = 60

AUTO_RESUME = TRUE

INITIALLY_SUSPENDED = TRUE;

Results (just now)	
Table	Chart
1 row	153ms
status	
1	Warehouse MEDIUM_WH successfully created.

USE WAREHOUSE SMALL_WH; (Slower execution)

SELECT

course,

COUNT(*) AS total_students,

AVG(score) AS avg_score

FROM students

GROUP BY course;

Results (just now)			
Table Chart			
Q 3 rows 802ms			
#	COURSE	# TOTAL_STUDENTS	# AVG_SCORE
1	java	3	78.666667
2	Snowflake	3	92.333333
3	Python	3	74.666667

USE WAREHOUSE MEDIUM_WH; (Faster execution)

SELECT

course,

COUNT(*) AS total_students,

AVG(score) AS avg_score

FROM students

GROUP BY course;

Results (just now)			
Table Chart			
Q 3 rows 35ms			
#	COURSE	# TOTAL_STUDENTS	# AVG_SCORE
1	java	3	78.666667
2	Snowflake	3	92.333333
3	Python	3	74.666667

Compare execution time.

Warehouse Execution Time

SMALL_WH ~820 ms

MEDIUM_WH ~35 ms

Assignment 7: Views & Secure Views

Objective: Control data visibility.

Tasks

1. Create a **View** hiding sensitive columns.
2. Create a **Secure View**.
3. Grant view access to Student role.
4. Test access using Student role.

Deliverable:

- SQL queries
- Explanation of Secure View

Answer:

```
CREATE TABLE students_full (
```

```
  id INT,
```

```
  name STRING,
```

```
  course STRING,
```

```
  score INT,
```

```
  email STRING -- sensitive column
```

```
);
```

```
insert into students_full values (1,'ramesh','hr',60,'roshini@gmail.com');
```

```
select * from students_full;
```

1. Create a View hiding sensitive columns.

create or replace view security_view as

```
select id,name,course,score
```

```
from students_full;
```

```
select*from security_view
```

Results (just now)

Table Chart					1 row 436ms
#	ID	NAME	COURSE	SCORE	
1	1	ramesh	hr	60	

2.Create a Secure View.

```
CREATE OR REPLACE SECURE VIEW student_secure_view AS  
SELECT id,name FROM students;
```

Results (just now)	
Table	Chart
1 row	166ms
status	
1	View STUDENT_SECURE_VIEW successfully created.

3. Grant view access to Student role.

```
GRANT SELECT ON VIEW student_secure_view  
TO ROLE student_role;
```

Results (just now)	
Table	Chart
1 row	59ms
status	
1	Statement executed successfully.

Assignment 8: Scaling & Auto Suspend (Advanced)

Objective: Understand Snowflake scaling behavior.

Tasks

1. Enable **auto-suspend** and **auto-resume**.

```
alter warehouse small_wh  
set AUTO_SUSPEND = 60  
AUTO_RESUME = TRUE;
```

197	
198	
199	alter warehouse small_wh
200	set AUTO_SUSPEND = 60
201	AUTO_RESUME = TRUE;
202	
Results (3 minutes ago)	
Table	Chart
1 row	80ms
status	
1	Statement executed successfully.

2. Configure multi-cluster warehouse.

CREATE OR REPLACE WAREHOUSE MC_WH

WITH

WAREHOUSE_SIZE = 'SMALL'

WAREHOUSE_TYPE = 'STANDARD'

MIN_CLUSTER_COUNT = 1 **MAX_CLUSTER_COUNT** = 3

SCALING_POLICY = 'STANDARD'

AUTO_SUSPEND = 6 **AUTO_RESUME** = TRUE;

```
202 CREATE OR REPLACE WAREHOUSE MC_WH
203 WITH
204
205     WAREHOUSE_SIZE = 'SMALL'
206     WAREHOUSE_TYPE = 'STANDARD'
207     MIN_CLUSTER_COUNT = 1
208     MAX_CLUSTER_COUNT = 3
209     SCALING_POLICY = 'STANDARD'
210     AUTO_SUSPEND = 60
211     AUTO_RESUME = TRUE;
212
```

Results (just now)

Table Chart 1 row 141ms

#	status
1	Warehouse MC_WH successfully created

3. Simulate concurrent queries.

```
249
250 USE DATABASE SNOWFLAKE_SAMPLE_DATA;
251 USE SCHEMA TPC_SF100;
252
253 SELECT * FROM ORDERS;
254 SELECT * FROM CUSTOMER;
```

Results (just now)

Table Chart 15,000,000 rows 7.3s

#	C_CUSTKEY	C_NAME	C_ADDRESS	C_NATIONKEY	C_PHONE	C_ACCTBAL	C_MKTSEGMENT	C_COMMENT
1	300001	Customer#000300001	0jfaFdT5SB3CCDdwYUAZ8dWn_HSXy0	6	16-419-497-9037	7955.82	HOUSEHOLD	quickly bold acc
2	300002	Customer#000300002	yhl4h9kGJ8OINLpYyCq79md7MoLM1V	2	12-227-473-8980	6534.87	BUILDING	special deposits
3	300003	Customer#000300003	8pvHvC1ZCsUslh4d2ko_YCbQHSd,10	2	12-404-137-2245	5604.61	AUTOMOBILE	ove the pending,
4	300004	Customer#000300004	YTdcod6_jLrsmHxiT0V5O7VX5HCh3Zxcb6	24	34-525-267-6443	1715.48	HOUSEHOLD	. carefully brave c
5	300005	Customer#000300005	UyvZTlo3ezLSDp,2t8XAt,XfiP	19	29-824-522-6822	9979.50	FURNITURE	ly even requests

4. Observe scaling behavior.

256

257

258

259

260

261

262

263

264

265

266

267

```

SELECT
  QUERY_ID,
  WAREHOUSE_NAME,
  CLUSTER_NUMBER,
  EXECUTION_STATUS
FROM TABLE(
  INFORMATION_SCHEMA.QUERY_HISTORY(
    END_TIME_RANGE_START => DATEADD('minute', -10, CURRENT_TIMESTAMP())
  )
)
ORDER BY START_TIME;

```

Results (just now)

Table

Chart

Q

14 rows

1.8s

↓

🔄

	QUERY_ID	WAREHOUSE_NAME	CLUSTER_NUMBER	EXECUTION_STATUS
1	01c11b17-0001-6deb-000c-c8fe0007c72e	MC_WH	null	SUCCESS
2	01c11b17-0001-6e0f-000c-c8fe0007e3a6	MC_WH	null	FAILED_WITH_ERROR
3	01c11b1a-0001-6e0f-000c-c8fe0007e4a6	MC_WH	null	SUCCESS
4	01c11b1a-0001-6e0f-000c-c8fe0007e4aa	MC_WH	null	SUCCESS
5	01c11b1a-0001-6deb-000c-c8fe0007c8a6	MC_WH	null	SUCCESS
6	01c11b1a-0001-6deb-000c-c8fe0007c8b6	MC_WH	null	SUCCESS

Deliverable:

- Warehouse configuration
- Query history evidence

