
System Requirements Specification Index

For

Snowflake web
server log analysis
using staging &
custom functions

Version 2.0

Problem Statement : snowflake webserver log analysis

Description : Use relevant methods operations to perform specified activities which are given in the instructions.

TechSolutions Inc. is a rapidly growing tech company that offers a suite of web-based applications to its global customer base. As the user base expanded, the volume of web server logs increased, making it challenging to monitor and analyze server performance and user activities.

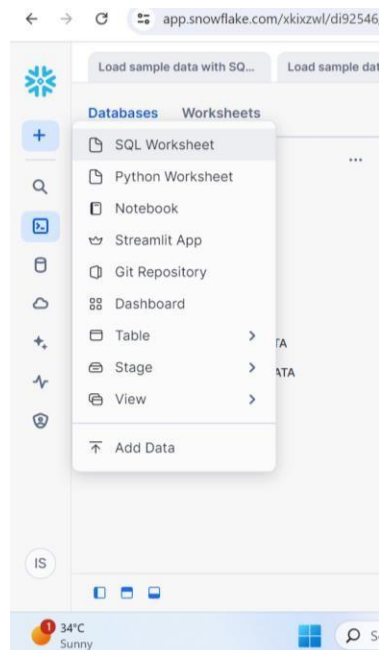
Objective

TechSolutions Inc. aimed to implement a scalable solution to store, manage, and analyze web server log data to gain insights into user behavior, identify and troubleshoot errors, and optimize server performance.

Implementation

TechSolutions Inc. chose Snowflake for its scalable data warehousing capabilities and ease of handling time-series data. The following steps were taken:

- Steps to login in the snowflake account
- Use the credential given to you through your host



- Click on the sql worksheet to open the query editor

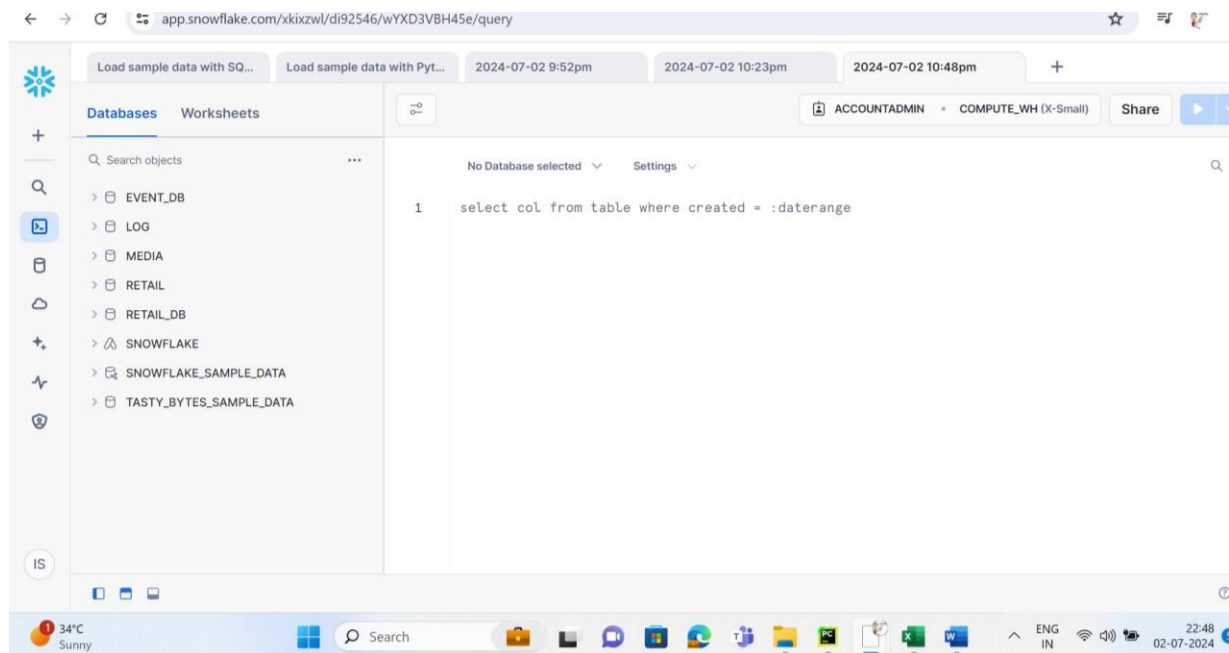
Sample data log

Timestamp	Log Level	Message	User ID	Session ID
2024-06-30 08:00:00	INFO	User logged in	alice123	session001
2024-06-30 08:05:00	ERROR	Failed to load page	alice123	session001
2024-06-30 08:10:00	INFO	User logged out	alice123	session001
2024-07-01 09:00:00	INFO	User logged in	bob_smith	session002
2024-07-01 09:15:00	WARN	Slow response time	bob_smith	session002
2024-07-01 09:20:00	ERROR	Page not found	bob_smith	session002
2024-07-02 10:00:00	INFO	User logged in	charlie_007	session003
2024-07-02 10:30:00	INFO	Page loaded successfully	charlie_007	session003
2024-07-02 11:00:00	ERROR	Database connection failed	charlie_007	session003
2024-07-02 11:15:00	INFO	User logged out	charlie_007	session003
2024-07-02 12:00:00	INFO	User logged in	diana_king	session004
2024-07-02 12:10:00	INFO	Page loaded successfully	diana_king	session004
2024-07-02 12:20:00	WARN	High memory usage	diana_king	session004
2024-07-02 12:30:00	ERROR	Service unavailable	diana_king	session004
2024-07-02 12:45:00	INFO	User logged out	diana_king	session004
2024-07-02 13:00:00	INFO	User logged in	eve_nash	session005
2024-07-02 13:05:00	INFO	Page loaded successfully	eve_nash	session005
2024-07-02 13:15:00	INFO	User logged out	eve_nash	session005
2024-07-03 14:00:00	INFO	User logged in	franklin_92	session006
2024-07-03 14:05:00	ERROR	Login failed	franklin_92	session006
2024-07-03 14:15:00	INFO	User logged out	franklin_92	session006
2024-07-04 15:00:00	INFO	User logged in	george_t	session007
2024-07-04 15:10:00	INFO	Page loaded successfully	george_t	session007
2024-07-04 15:20:00	WARN	Unexpected input	george_t	session007
2024-07-04 15:30:00	ERROR	Session timeout	george_t	session007

2024-07-04 15:45:00	INFO	User logged out	george_t	session007
2024-07-05 16:00:00	INFO	User logged in	hannah_b	session008
2024-07-05 16:05:00	INFO	Page loaded successfully	hannah_b	session008
2024-07-05 16:15:00	ERROR	Payment failed	hannah_b	session008
2024-07-05 16:30:00	INFO	User logged out	hannah_b	session008

Task A STAGING IN SNOWFLAKE

1. Create a custom user stage with the database
2. Install vs code and add snowflake extension connect to the snowflake account from vscode
3. Create a new table to store the table
4. Load the stage data in the table
5. Query the weblog table to view data have been imported successfully



- Click on the database icon on the left side in the console

Databases

+ Database

9 Databases

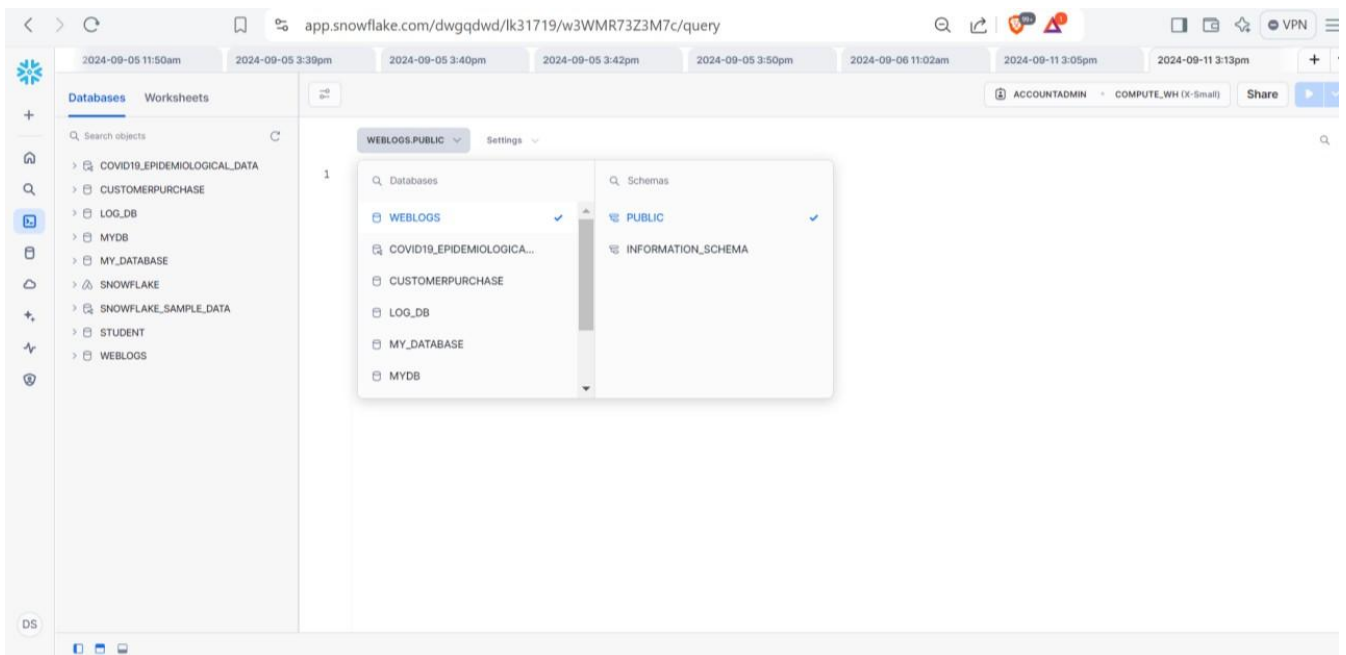
Q Search

Source All



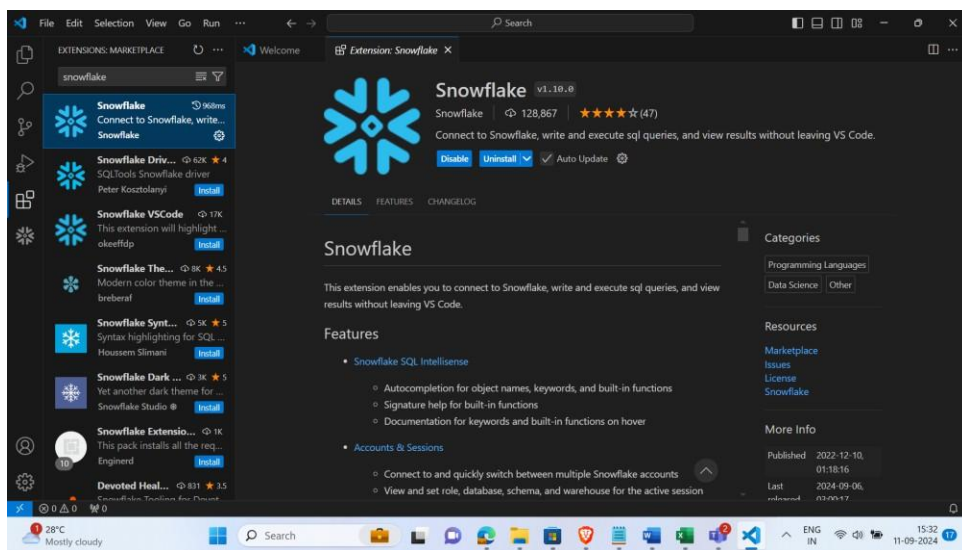
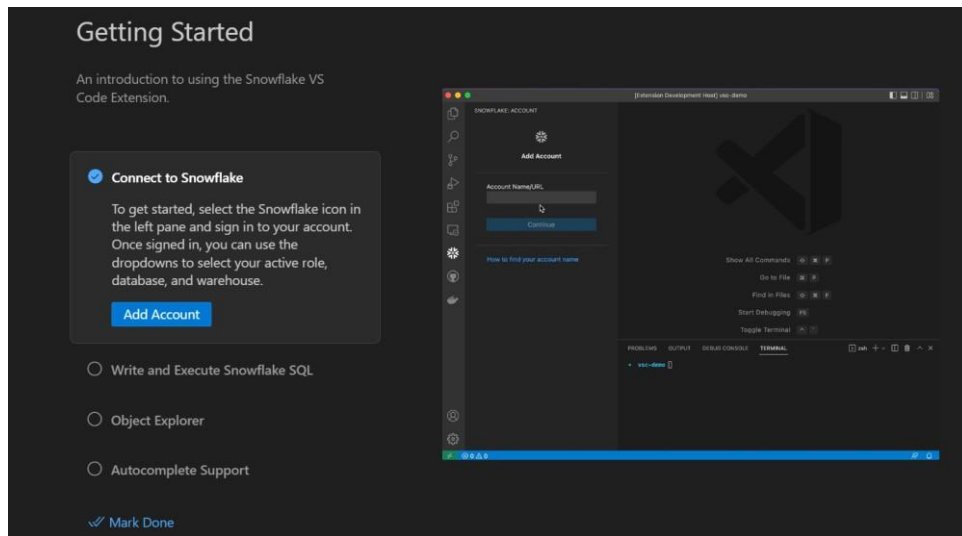
NAME ↑	SOURCE	OWNER	CREATED	
COVID19_EPIDEMIOLOGICAL_DATA	Share	ACCOUNTADMIN	1 week a...	...
CUSTOMERPURCHASE	Local	ACCOUNTADMIN	1 week a...	...
LOG_DB	Local	ACCOUNTADMIN	1 week a...	...
MYDB	Local	ACCOUNTADMIN	1 week a...	...
MY_DATABASE	Local	ACCOUNTADMIN	6 days a...	...
SNOWFLAKE	Share	—	2 weeks...	...
SNOWFLAKE_SAMPLE_DATA	Share	ACCOUNTADMIN	2 weeks...	...
STUDENT	Local	ACCOUNTADMIN	5 days a...	...
WEBLOGS	Local	ACCOUNTADMIN	4 minute...	...

- Create a database give it name weblogs
- Now click on Create - >Sql worksheet
- Select the database like show in the image we have created weblogs
- Select the same database name and schema name .

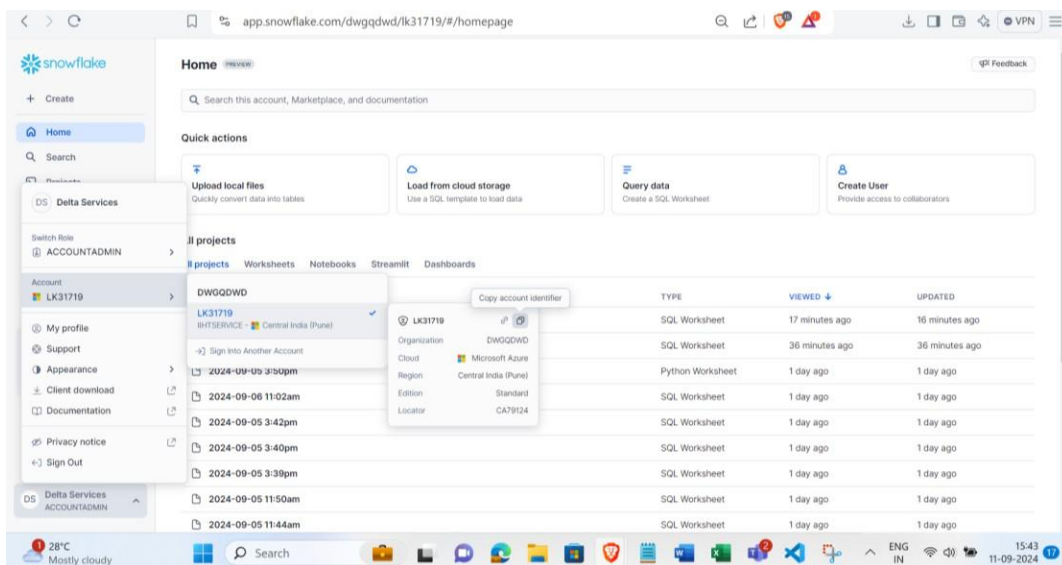


The screenshot displays the Snowflake web interface in a browser. The address bar shows the URL: `app.snowflake.com/dwgqdw/1k31719/w3WMR73Z3M7c#query`. The interface includes a sidebar with a search bar and a list of databases: `COVID19_EPIDEMIOLOGICAL_DATA`, `CUSTOMERPURCHASE`, `LOG_DB`, `MYDB`, `MY_DATABASE`, `SNOWFLAKE`, `SNOWFLAKE_SAMPLE_DATA`, `STUDENT`, and `WEBLOGS`. The main workspace shows a SQL query: `CREATE OR REPLACE STAGE user_stage;`. Below the query, the 'Results' tab is active, displaying a table with one row: `1 Stage area USER_STAGE successfully created.`. To the right of the results, a 'Query Details' panel shows: `Query duration: 152ms`, `Rows: 1`, and `Query ID: 01b664f0-0001-31ab-00...`. The bottom of the image shows a Windows taskbar with the date `11-09-2024` and time `15:22`.

- Open VS code go to extensions search for snowflake and install .



- Click on the install button
- You need to get the account ID from the snowflake console



ACCOUNT

Connection Name

ca79124.central-india.azure

Auth Method

Username/password

Username

iihtservice

Password

.....

Sign in

QUERY HISTORY

- Authenticate to the snowflake account
- Create the staging environment you can use the same structure from the screenshot you must get the similar output .

```

PS C:\Users\Vignesh> snowsql -a ca79124.central-india.azure -u iihtservice -q "
>> USE DATABASE weblogs;
>> USE SCHEMA public;
>> PUT 'file:///D:/important/project documents/snowflake/advanced/weblogs.csv' @user_stage;"
>>
Password:
* SnowSQL * v1.3.0
Type SQL statements or !help

```

```

1 Row(s) produced. Time Elapsed: 0.087s
+-----+-----+-----+-----+-----+-----+-----+
| source | target | source_size | target_size | source_compression | target_compression | status | message |
+-----+-----+-----+-----+-----+-----+-----+
| weblogs.csv | weblogs.csv.gz | 1910 | 496 | NONE | GZIP | UPLOADED |
+-----+-----+-----+-----+-----+-----+-----+
1 Row(s) produced. Time Elapsed: 2.032s
Goodbye!

```

- Creating the table for data to be imported to the table

SNOWFLAKE

ACCOUNT

WEBLOGS

PUBLIC

COMPUTE_WH

OBJECT EXPLORER

Search

MY_DATABASE

SNOWFLAKE_SAMPLE_DATA

STUDENT

WEBLOGS

QUERY HISTORY

CREATE OR REPLACE TABLE ...

CREATE OR REPLACE TABLE user J...

CREATE OR REPLACE TABLE user J...

Execute

```

1 CREATE OR REPLACE TABLE weblogs (
2   Timestamp TIMESTAMP,
3   Log_Level STRING,
4   Message STRING,
5   User STRING,
6   Session STRING
7 );
8

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SNOWFLAKE SNOWFLAKE

status

1 Table WEBLOGS successfully created.

Query Details

01b6f51f-0001-31ab-0005-8ca60002ad1a

- Create the userstage using the following query

Sql code CREATE OR REPLACE STAGE user_stage;

- Create the table by selecting the weblogs database .

The screenshot shows the Snowflake IDE interface. The top pane displays a SQL query: `COPY INTO weblogs`, `FROM @user_stage/weblogs.csv`, and `FILE_FORMAT = (TYPE = 'CSV' FIELD_OPTIONALLY_ENCLOSED_BY = '' SKIP_HEADER = 1);`. The bottom pane shows the results of the query in a table format.

file	status	rows_parsed	rows_loaded
user_stage/weblogs.csv	LOADED	30	30

Query Details:

- Query ID: 01b6f53d-0001-31aa-0005-8ca60002c0e2
- Query Duration: 601ms
- Rows: 1
- End time: Wed Sep 11 2024
- Scanned: 496.0B (100%)
- Role: ACCOUNTADMIN
- DB/Schema: WEBLOGS.PUBLIC
- Warehouse: COMPUTE_WH

Query the database

- SELECT * FROM weblogs;

The screenshot shows the Snowflake IDE interface. The top pane displays a SQL query: `select * from weblogs;`. The bottom pane shows the results of the query in a table format.

	TIMESTAMP	LOG_LEVEL	MESSAGE	USER
1	30-06-2024 08:00	INFO	User logged in	alice123
2	30-06-2024 08:05	ERROR	Failed to load page	alice123
3	30-06-2024 08:10	INFO	User logged out	alice123
4	01-07-2024 09:00	INFO	User logged in	bob_smith
5	01-07-2024 09:15	WARN	Slow response time	bob_smith
6	01-07-2024 09:20	ERROR	Page not found	bob_smith
7	02-07-2024 10:00	INFO	User logged in	charlie_007

Query Details:

- Query ID: 01b6f53e-0001-31aa-0005-8ca60002c0ee
- Query Duration: 147ms
- Rows: 30
- End time: Wed Sep 11 2024
- Scanned: 3.0KB (100%)
- Role: ACCOUNTADMIN
- DB/Schema: WEBLOGS.PUBLIC
- Warehouse: COMPUTE_WH

- Once you have imported the table check for respective rows and columns are imported correctly.

Task B Perform operation with the table

- 1 Retrieve all logs for a specific user? "hannah_b"
- 2 Count the number of errors in the logs
- 3 Retrieve logs within a specific time range
- 4 Group logs by log level
- 5 Find the most recent log entry for each user

Task C Perform UDF function stored procedure

1. Connect to Snowpark with database created.
2. Create a categorize_log_level UDF function "Critical," "Warning," or "Normal" and print the results
3. Create and register the filter_and_categorize_logs stored procedure
4. Test the Stored Procedure
- 5 .Check the stored procedure for the user "hannah_b"
- 6 Execute and Validate the stored procedure for hannah_b_categorized_logs .

Note please create the **categorize_log_level** function with the same name

To Connect to Snowpark you can use any editor (template to connect to snowflake)

from snowflake.snowpark import Session

```
session = Session.builder.configs({
    "account": "<your_account>",
    "user": "<your_username>",
    "password": "<your_password>",
    "role": "<your_role>",
    "warehouse": "<your_warehouse>",
    "database": "<your_database>",
    "schema": "<your_schema>"
}).create()
```

Execution Steps to Follow:

1. Open the snowflake console
2. Import the dataset from the document
3. Perform all the query respective to the question provided
4. Take screenshots of the query execution
5. Upload the code to the Github

-----X-----

