

Steven Evans
SRE220000
BUAN 6335
Snowflake Lab 7

7.1.2 Warehouse meeting history query 1

DatabasesWorksheets

Search objects

ADMINBADGER_DBBEETLE_DBBISON_DBBLUEJAY_DBBOA_DBCAMEL_DBCATFISH_DBCAT_DBCHEETAH_DBCHIPMUNK_DBCOBRA_DBCOYOTE_DBCRICKET_DBDOG_DBDOLPHIN_DBDRAGON_DBEAGLE_DBFALCON_DBFERRET_DBFINCH_DBFLAMINGO_DB

SNOWFLAKE.ACCOUNT_USAGESettings

63646566676869707172737475767778798081828384858687

```
-- Credits used (all time = past year)
SELECT WAREHOUSE_NAME
      ,SUM(CREDITS_USED_COMPUTE) AS CREDITS_USED_COMPUTE_SUM
FROM ACCOUNT_USAGE.WAREHOUSE_METERING_HISTORY
GROUP BY 1
ORDER BY 2 DESC;

-- Credits used (past N days/weeks/months)
SELECT WAREHOUSE_NAME,
      SUM(CREDITS_USED_COMPUTE) AS CREDITS_USED_COMPUTE_SUM
FROM ACCOUNT_USAGE.WAREHOUSE_METERING_HISTORY
WHERE START_TIME >= DATEADD(DAY, -7, CURRENT_TIMESTAMP()) // Past 7 days
GROUP BY 1
ORDER BY 2 DESC;
```

ResultsChart

	WAREHOUSE_NAME	CREDITS_USED_COMPUTE_SUM
1	BOA_WH	6.468333341
2	TIGER_WH	5.851388890
3	GECKO_WH	5.505000012
4	GROUNDHOG_WH	4.631111119
5	COYOTE_WH	4.627222230

Query Details

Query duration225ms

Rows39

Query ID01b8c55b-0004-44ea-b...

Show more

Ask Copilot

Query 2

Databases

Worksheets

Search objects

ADMIN

BADGER_DB

BEETLE_DB

BISON_DB

BLUEJAY_DB

BOA_DB

CAMEL_DB

CATFISH_DB

CAT_DB

CHEETAH_DB

CHIPMUNK_DB

COBRA_DB

COYOTE_DB

CRICKET_DB

DOG_DB

DOLPHIN_DB

DRAGON_DB

EAGLE_DB

FALCON_DB

FERRET_DB

FINCH_DB

FLAMINGO_DB

SQL

TRAINING_ROLE

CHIPMUNK_WH (X-Small)

Share

SNOWFLAKE.ACCOUNT_USAGE

Settings

```

63 CREATE WAREHOUSE IF NOT EXISTS SNOWFLAKE.CHIPMUNK_WH;
64 USE WAREHOUSE CHIPMUNK_WH;
65
66 USE SCHEMA SNOWFLAKE.ACCOUNT_USAGE;
67
68
69 -- 7.1.2 Examine credit usage by warehouse.
70 -- Monitoring credit consumption for specific objects is a classic use
71 -- of data in the ACCOUNT_USAGE schema. Here is an example of two
72 -- queries that use the WAREHOUSE_METERING_HISTORY view:
73
74 -- Credits used (all time = past year)
75 SELECT WAREHOUSE_NAME
76       ,SUM(CREDITS_USED_COMPUTE) AS CREDITS_USED_COMPUTE_SUM
77   FROM ACCOUNT_USAGE.WAREHOUSE_METERING_HISTORY
78  GROUP BY 1
79  ORDER BY 2 DESC;
80
81 -- Credits used (past N days/weeks/months)
82 SELECT WAREHOUSE_NAME,
83       SUM(CREDITS_USED_COMPUTE) AS CREDITS_USED_COMPUTE_SUM
84   FROM ACCOUNT_USAGE.WAREHOUSE_METERING_HISTORY
85  WHERE START_TIME >= DATEADD(DAY, -7, CURRENT_TIMESTAMP()) // Past 7 days
86  GROUP BY 1
87  ORDER BY 2 DESC;

```

Results

Chart

WAREHOUSE_NAME

CREDITS_USED_COMPUTE_SUM

Query Details

FALCON_DB

6.285000007

Query duration

3.912777783

Rows

3.373611117

Query ID

3.196666670

Show more

2.904166672

Ask Copilot

Details

Definition

Source

Local

7.1.3 Determine warehouses without resource monitor

DatabasesWorksheets

Search objects

ADMINBADGER_DBBEETLE_DBBISON_DBBLUEJAY_DBBOA_DBCAMEL_DBCATFISH_DBCAT_DBCHEETAH_DBCHIPMUNK_DBCOBRA_DBCOYOTE_DBCRICKET_DBDOG_DBDOLPHIN_DBDRAGON_DBEAGLE_DBFALCON_DBFERRET_DBFINCH_DBFLAMINGO_DB

SNOWFLAKE.ACCOUNT_USAGESettings

115-- or date range. This can help prevent certain warehouses from

116-- unintentionally consuming more credits than typically expected.

117

118

119SHOW WAREHOUSES;

120

121SELECT "name" AS WAREHOUSE_NAME

122, "size" AS WAREHOUSE_SIZE

123FROM TABLE(RESULT_SCAN(LAST_QUERY_ID()))

124WHERE "resource_monitor" = 'null';

125

126

127-- You shouldn't find any warehouses without resource monitors because

128-- all of them have at least one. So, your result should be 0. However,

129-- if you have not put a resource monitor on your own warehouse

130-- (CHIPMUNK_WH), then that warehouse may appear in the results.

131

132-- 7.2.0 Billing Metrics

133-- Billing metrics is all about analyzing what you've been billed in the

134-- past so you can determine if there is a way to lower costs in the

135-- future.

136

137-- 7.2.1 Most expensive queries from the last 30 days

138-- The query below analyzes queries that are potentially too expensive

ResultsChart

	WAREHOUSE_NAME	WAREHOUSE_SIZE
1	SYSTEM\$STREAMLIT_NOTEBOOK_WH	X-Small

Query Details

Query duration290ms

Rows1

Query ID01b8e5c-0004-4560-

Show more

Ask Copilot

7.2.0 Evaluating expensive queries

Databases

Worksheets

Search objects

ADMIN

BADGER_DB

BEETLE_DB

BISON_DB

BLUEJAY_DB

BOA_DB

CAMEL_DB

CATFISH_DB

CAT_DB

CHEETAH_DB

CHIPMUNK_DB

COBRA_DB

COYOTE_DB

CRICKET_DB

DOG_DB

DOLPHIN_DB

DRAGON_DB

EAGLE_DB

FALCON_DB

FERRET_DB

FINCH_DB

FLAMINGO_DB

SNOWFLAKE.ACCOUNT_USAGE

Settings

115

-- or date range. This can help prevent certain warehouses from

116

-- unintentionally consuming more credits than typically expected.

117

118

119

SHOW WAREHOUSES;

120

121

SELECT "name" AS WAREHOUSE_NAME

122

,"size" AS WAREHOUSE_SIZE

123

FROM TABLE(RESULT_SCAN(LAST_QUERY_ID()))

124

WHERE "resource_monitor" = 'null';

125

126

127

-- You shouldn't find any warehouses without resource monitors because

128

-- all of them have at least one. So, your result should be 0. However,

129

-- if you have not put a resource monitor on your own warehouse

130

-- (CHIPMUNK_WH), then that warehouse may appear in the results.

131

132

-- 7.2.0 Billing Metrics

133

-- Billing metrics is all about analyzing what you've been billed in the

134

-- past so you can determine if there is a way to lower costs in the

135

-- future.

136

137

-- 7.2.1 Most expensive queries from the last 30 days

138

-- The query below analyzes queries that are potentially too expensive

Results

Chart

WAREHOUSE_NAME

WAREHOUSE_SIZE

1

SYSTEM\$STREAMLIT_NOTEBOOK_WH

X-Small

Query Details

...

Query duration

290ms

Rows

1

Query ID

01b8e5c-0004-4560-

Show more

Ask Copilot

7.2.2 Queries with the most spillage to remote storage

DatabasesWorksheets

Search objects

ADMIN

BADGER_DB

BEETLE_DB

BISON_DB

BLUEJAY_DB

BOA_DB

CAMEL_DB

CATFISH_DB

CAT_DB

CHEETAH_DB

CHIPMUNK_DB

COBRA_DB

COYOTE_DB

CRICKET_DB

DOG_DB

DOLPHIN_DB

DRAGON_DB

EAGLE_DB

FALCON_DB

FERRET_DB

FINCH_DB

FLAMINGO_DB

SQL

TRAINING_ROLECHIPMUNK_WH (X-Small)Share

SNOWFLAKE.ACCOUNT_USAGESettings

...

198

--

199

--

200

--

201

--

202

--

203

--

204

--

205

--

206

--

207

--

208

--

209

--

210

--

211

--

212

--

This query gives you the chance to evaluate expensive queries and take some action. For example, you could look at the query profile, contact the user who executed the query, or take action to optimize these queries.

Below is a list of the columns in this secure view, SNOWFLAKE.ACCOUNT_USAGE.QUERY_HISTORY.

7.2.2 Top 10 Queries With The Most Spillage to Remote Storage

Another way to evaluate the cost of queries is to see if they are spilling to remote storage. The query below allows you to do that.

SELECT query_id, substr(query_text, 1, 50) partial_query_text, user_name, warehouse_name, warehouse_size, BYTES_SPILLED_TO_REMOTE_STORAGE, start_time, end_time, total_elapsed_time/1000 total_elapsed_time FROM snowflake.account_usage.query_history

ResultsChart

QUERY_ID	PARTIAL_QUERY_TEXT	USER_NAME	WAREHO
Query produced no results			

Query Details

Query duration2.2s

Rows0

Query ID01b8cc5e-0004-44ca-0...

Show more

Ask Copilot