Optimising Snowflake exercises

List of all assignments for Optimizing Snowflake together at one place. Please find all DDL and all other needed queries for the exercises in following file on GitHub: `/exercises.sql`

02 Virtual Warehouse Configuration

This exercise is focused on effective configuration of virtual warehouses. What parameters should be set and how. We are going to create a single virtual warehouse with following specs:

Create your first virtual warehouse called `COMPUTE_WH`, with following parameters:

- Extra small size
- Enabled auto resume
- Enabled auto suspend
- Suspend after 1 minute
- It will be suspended after creation

Create a multi cluster warehouse called `MULTI_COMPUTE_WH` with following parameters:

- Small size
- Enabled auto resume
- Enable auto suspend
- Suspend after 1 minutes
- It will be suspended after creation
- Min number of clusters: 1
- Max number of clusters: 3
- Economy scaling policy

Next we are going to modify the `STATEMENT_TIMEOUT_IN_SECONDS` parameter. The default value (2 days) is too high and can lead to unintentional cost spikes. First we need to find out what is currently configured:

- On warehouse level
- On session level
- On account level

We will modify the value only on the warehouse level. This is the lowest level and Snowflake will always use this value also in case it would be configured on session and account level. Modify the `STATEMENT_TIMEOUT_IN_SECONDS` on `COMPUTE_WH` to 1 hour (3600 seconds).

As a last step in effective warehouse configuration we are going to create a resource monitor which can suspend our warehouse automatically in case of reaching the defined limit. Once we have the resource monitor in place, we will assign it to our `COMPUTE_WH`.

Create resource monitor `compute_quota` with following parameters:

- Credit quota = 50 credits
- Monthly frequency
- Starts immediately
- It will notify when reaching 75% of the quota
- It will suspend the warehouse when reaching 98% of the quota
- It will suspend the warehouse immediately when reaching 100% of the quota

Assign this resource monitor to our `COMPUTE_WH`.

You can find all the SQL statements on GitHub in `/exercise.sql` file.

07 Table scans and micro partition pruning

We are going to have a look on queries from slides to find out how micro partition pruning works for some queries and how we can improve it. We will also have a look leveraging micro partition statistics in our queries. Last part of this exercises is dedicated to JOIN filters and how Snowflake optimizer can automatically push down some filters to base tables based on join conditions. This is also something what you should be looking for in query plans and if JOIN filters are not in place think how it can be involved into the query to support partition pruning.

Please find all the queries used in this exercise on GitHub in `/exersice.sql` file.

08 Table clustering

During this exercise we are going to work with system functions related to clustering. We will try to check how good is defined clustering and whether we can improve it by changing the clustering key.

We will also try the natural clustering and its impact on query performance.

Please find all the queries used in this exercise on **GitHub** in `/exersice.sql` file.

10 How to improve queries

During this exercise we will focus on trying the explained concepts on real queries. We are going to try optimize the WHERE clause. We will have a look how could be improved the MERGE statements and how CTEs works in relation to query performance.

Please find all the queries used in this exercise on **GitHub** in `/exersice.sql` file.