

Chapter 17: Tasks & Transactions in Snowflake

Tasks & Transactions in Snowflake

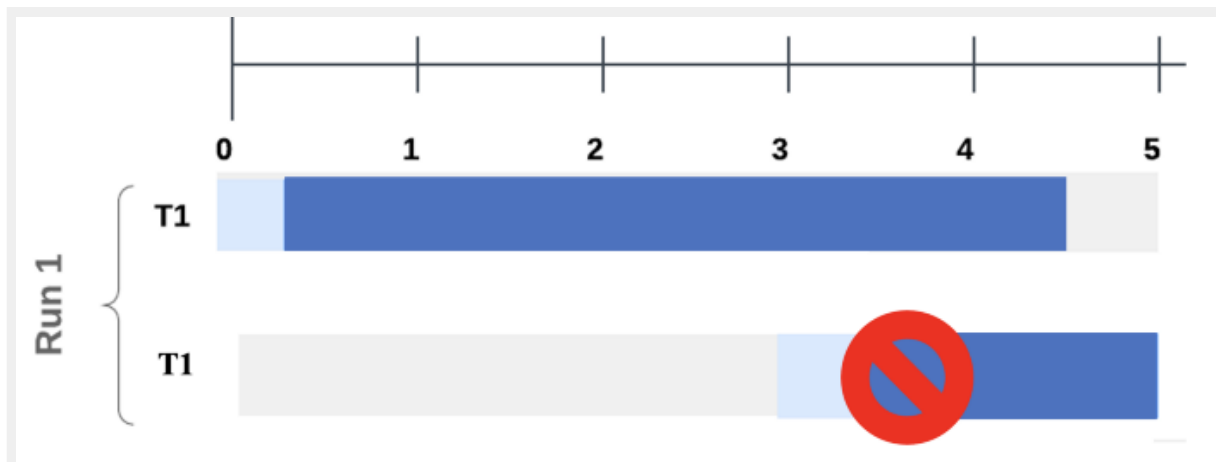
We are coming to the end of the Snowflake SnowPro Core course, but we can't forget about Tasks and Transactions in Snowflake, which will allow us to extend the functionality of our favourite Cloud Data Warehouse. Let's get to it!

1. [Introduction to tasks](#)
2. [Tree of tasks](#)
3. [Task History](#)
4. [Transactions](#)
5. [Typical Exam Questions](#)

INTRODUCTION TO TASKS

Snowflake tasks are schedulable scripts that are run inside your Snowflake environment. No event source can trigger a task; instead, a task runs on a schedule, which can be defined when creating a task, and they run with the privileges of the task owner. A task can execute a single SQL statement, including a call to a Stored Procedure. This last part is really important, and it usually appears in the Snowflake SnowPro Core Exam.

Snowflake ensures only one instance of a task with a schedule is executed at a given time. If a task is still running when the next scheduled execution time occurs, then that scheduled time is skipped. Imagine that we have a task that it's executed every 5 minutes. If the first execution hasn't finished by the time that the second one should start, the second one will be skipped. Tasks also have a maximum duration of 60 minutes by default. If they haven't finished by then, they will be automatically terminated.



Execution of task T1 skipped.

You can also specify a CRON expression and time zone for periodically running a scheduled task. When you first create a task, it will be suspended by default. You can activate it with the following command:

```
ALTER TASK mytask RESUME;
```

TREE OF TASKS

Users can define a simple tree-like structure of tasks that starts with a root task and is linked together by task dependencies. The Root task will be the only one to which the scheduler is set. The children's tasks only run after the parent task finishes.

Considerations:

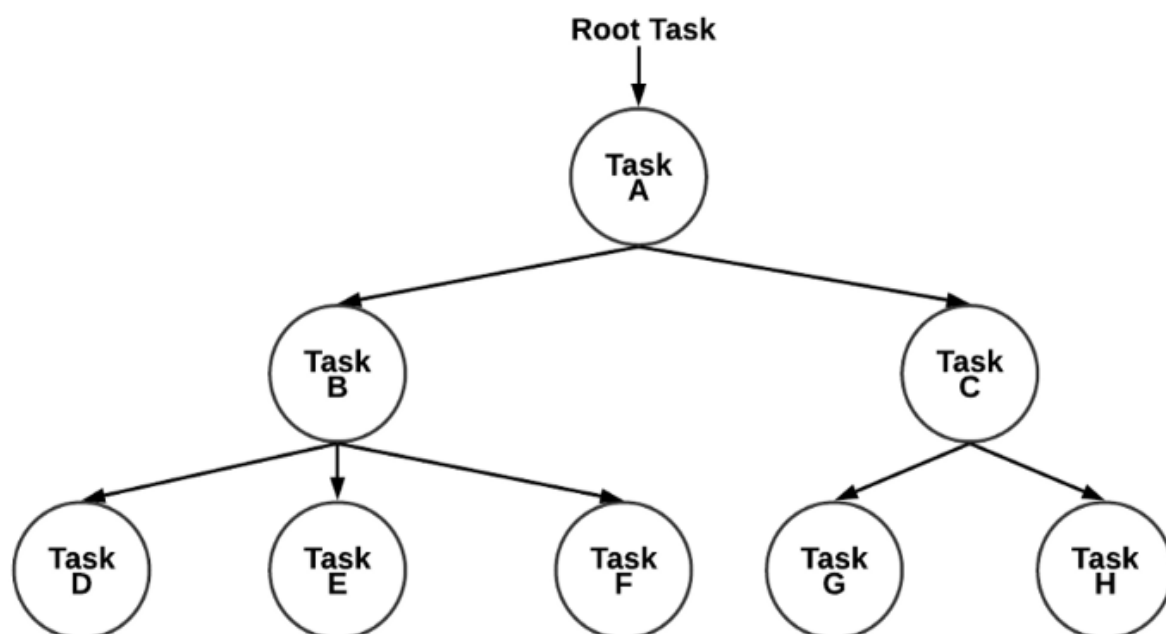
- *Each task can have a maximum of 100 children.*
- *A tree of tasks can have a maximum of 1000 tasks, including the root one.*
- *A child task can only have a parent task.*
- *You set the scheduler in the root task.*

We can create child tasks with the following commands:

```
CREATE TASK <newTask> AFTER <rootTask>;
```

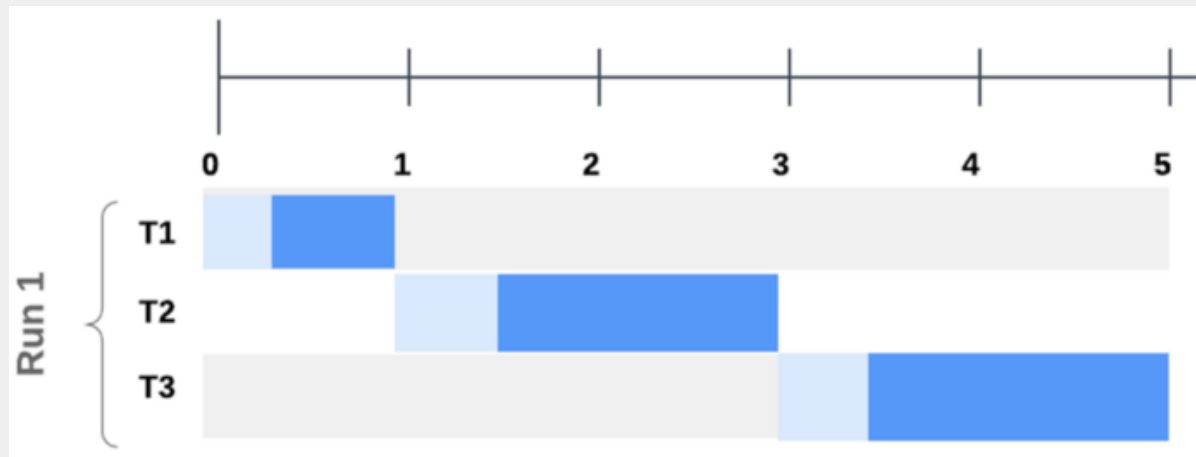
```
ALTER TASK <newTask> ADD AFTER <rootTask>;
```

In the following example of a tree of tasks, "task A" is executed, and when it finishes, tasks "B" and "C" are launched at the same time. When "task B" finishes, Snowflake will launch tasks "D", "E", and "F" at the same time. When "task C" finishes, the tasks "G" and "H", the same thing will happen.



Tree of tasks example (via docs.snowflake.com)

In this example of a tree of 3 tasks, you can see that there is only one task running at a time. The root task can have a queuing time in case there is already something else in the warehouse. Child tasks will always have a queuing time.



Three tasks running on Snowflake.

A typical question in the exam is what would happen if we eliminated the parent task. In this case, there are two options:

- Child task becomes a standalone task.
- Child task becomes a root task.

And what if the owner role of a task is deleted? Task Ownership is reassigned to the role that dropped this role. This is also a typical exam question.

TASK HISTORY

You can use this Snowflake function to query the history of task usage within a specified date range, and you can find it in the information schema. You need one of these privileges to see the task history:

- *AccountAdmin role.*
- *You are the owner of a task.*
- *You have the global `MONITOR_EXECUTION` privilege.*

```
SELECT * FROM table(information_schema.task_history()) ORDER
BY scheduled_time;
```

TRANSACTIONS

A transaction is a sequence of SQL statements that are committed or rolled back as a unit. All statements in the transaction are either applied (i.e., committed) or undone (i.e., rolled back) together. For this reason, transactions are ACID (Atomicity, Consistency, Isolation & Durability) Compliant. It provides a safe environment in which to operate on your data. Once a transaction is complete, its data is consistent.

Things to know about transactions:

- *Snowflake takes 4 hours to abort it if we do not abort it with the `SYSTEM$ABORT_TRANSACTION` function if the session is disconnected for whatever reason and the transaction remains in a detached state.*
- *Each transaction has independent scope.*
- *Snowflake does not support Nested Transactions, although it supports Nested Procedure Calls.*

TYPICAL EXAM QUESTIONS

1. What object will you use to schedule a merge statement in Snowflake so that it runs every hour?

1. Task
2. Stream
3. Pipe
4. Table

Solution: 1

2. Select the true statements about Snowflake tasks:

1. A task can execute a single SQL Statement
2. A task can execute multiple SQL Statements
3. A task can execute a call to a Stored Procedure
4. A task can execute a function

Solution: 1, 3

3. A task is still being executed before the next scheduled task. What is going to happen with the new scheduled task?

1. Snowflake will abort it
2. Snowflake will skip it
3. Snowflake will wait for the previous task to complete

Solution: 2

4. Which roles or privileges are required to view TASK_HISTORY?

- 1. AccountAdmin*
- 2. MONITOR EXECUTION privilege*
- 3. SysAdmin*
- 4. SecurityAdmin*
- 5. Task owner (OWNERSHIP privilege)*

Solution: 1, 2, 5

5. What will happen to the child task if you remove its predecessor?

- 1. The child task is removed from the system*
- 2. The child task may become the root task*
- 3. The child task may become a standalone task*

Solution: 2, 3

6. A task went into a loop. How long will the task run before Snowflake finishes it?

- 1. 15 minutes*

2. 30 minutes
3. 60 minutes
4. 4 hours

Solution: 3. Tasks have a maximum duration of 60 minutes by default. If they haven't finished by then, they will be automatically terminated.

7. The owner of a task (the one who has the OWNERSHIP privilege) is deleted. What will happen to the task?

1. The task will belong to the role that dropped the owner's role
2. The task is deleted
3. The task is suspended

Solution: 1

8. If a transaction disconnects and goes into a detached state, which cannot be committed or rolled back, how long will Snowflake take to abort the transaction?

1. 15 minutes
2. 60 minutes
3. 4 hours
4. 12 hours

Solution: 3. If the transaction is left open, Snowflake typically rolls back the transaction after being idle for four hours. You can abort a running transaction with the system function `SYSTEM$ABORT_TRANSACTION`.

9. Does Snowflake support Nested Transactions?

1. True

2. False

Solution: 2

Thanks for Reading!