**Tasks for Scheduling**

Tasks in Snowflake are used to schedule and automate SQL-based operations. They can run on a **time schedule** using cron expressions or on **dependencies** between tasks (task chaining). A common real-world use case is setting up a pipeline where a **Stream captures new data**, and a **Task schedules incremental loads** into reporting tables every hour. its Executes SQL, Stored procedures can be combine with streams can be scheduled. We can configure with the CRON(crontab guru) Expressions as well.

CRON Job having 6 Stars \*

-- <1> - (0-59) minute

-- <2> - (0-23) hour

-- <3> - (1-31, L) day of month

-- <4> - (1-12, JAN,FEB,MAR..DEC) month

-- <5> - (0-6,SUN,MON,TUE..SAT, L) day of week

-- <6> - (America/Los\_Angeles, America/New\_York, 'America/Chicago') timezone

-- EmptyValue is \*

– CREATE TASK, SHOW TASK, EXECUTE TASK

-- create a task, show task, execute tasks

-- Let's create table --

create or replace table raw\_applicant\_staging

( id number autoincrement start = 111111 increment = 111111,

first\_name varchar,

last\_name varchar,

sex varchar,

ethinicity varchar,

ssn varchar,

street\_address varchar,

education\_level varchar,

years\_of\_experience number,

job\_id number

);

-- CREATE A TASK WITHOUT SCHEDULE WITH WAREHOUSE -- USER MANAGED TASK

create or replace task tsk\_add\_applicant

warehouse = compute\_wh -- optional, we are creating a task which uses the user defined compute power. Hence gave the warehouse

as

insert into raw\_applicant\_staging

(first\_name, last\_name, sex, ethinicity, ssn, street\_address, education\_level,years\_of\_experience,job\_id) values

('James', 'Schwartz', 'M', 'American', '342-76-9087','5676 Washington Street','High School', 5,10) ;

-- check task

show tasks ;

-- since there is no schedule we need to execute task

execute task tsk\_add\_applicant ; -- can only execute standalone task or a root task

-- select

select \* from raw\_applicant\_staging ;

--- CREATE SCHEDULED TASK WITHOUT WAREHOUSE -- A SERVERLESS TASK

delete from raw\_applicant\_staging ;

create or replace task tsk\_add\_applicant\_evry\_min\_server\_less

--warehouse = compute\_wh -- No warehouse

schedule = '1 MINUTE'

as

insert into

raw\_applicant\_staging

(first\_name, last\_name, sex, ethinicity, ssn, street\_address, education\_level,years\_of\_experience,job\_id) values ('James', 'Schwartz', 'M', 'American','342-76-9087','5676 Washington Street','High School', 5,10) ;

-- show tasks

show tasks ;

-- resume and suspend

alter task tsk\_add\_applicant\_evry\_min\_server\_less resume ;

-- check the data

select \* from raw\_applicant\_staging ;

--suspend it, once we see the data

alter task tsk\_add\_applicant\_evry\_min\_server\_less suspend ;

-- CREATE SCHEDULED TASK WITH WAREHOUSE--

delete from raw\_applicant\_staging ;

create or replace task tsk\_add\_applicant\_evry\_min\_usr\_mnged

warehouse = compute\_wh --

schedule = '1 MINUTE'

as insert into

raw\_applicant\_staging (first\_name, last\_name, sex, ethinicity, ssn, street\_address, education\_level,years\_of\_experience,job\_id)

values ('James', 'Schwartz', 'M', 'American','342-76-9087','5676 Washington Street','High School', 5,10) ;

-- show tasks

show tasks ;

-- resume and suspend

alter task tsk\_add\_applicant\_evry\_min\_usr\_mnged resume ;

-- check the data

select \* from raw\_applicant\_staging ;

-- if we wait for 2 more minutes we will see more rows inserted

--suspend it, once we see the data

alter task tsk\_add\_applicant\_evry\_min\_usr\_mnged suspend ;

-- cleanup

show tables ;

drop table if exists RAW\_APPLICANT\_STAGING ;

show tasks ;

drop task if exists TSK\_ADD\_APPLICANT ;

drop task if exists TSK\_ADD\_APPLICANT\_EVRY\_MIN;

drop task if exists TSK\_ADD\_APPLICANT\_EVRY\_MIN\_SERVER\_LESS;

drop task if exists TSK\_ADD\_APPLICANT\_EVRY\_MIN\_USR\_MNGED ;

--------------------------------------------------------------------

--CREATE TASK TO CALL STORED PROCEDURE

-- Let's create table --

create or replace table raw\_applicant\_staging

( id number autoincrement start = 111111 increment = 111111,

first\_name varchar,

last\_name varchar,

sex varchar,

ethinicity varchar,

ssn varchar,

street\_address varchar,

education\_level varchar,

years\_of\_experience number,

job\_id number

);

-- Task can have only one statement(select or storedproc or streams)

-- check the data

select \* from raw\_applicant\_staging ;

-- create a procedure using Snowflake Scripting code

create or replace procedure prc\_add\_applicant()

returns varchar

language sql

as

$$

begin

insert into raw\_applicant\_staging

(first\_name, last\_name, sex, ethinicity, ssn, street\_address, education\_level, years\_of\_experience, job\_id)

values ('James', 'Schwartz', 'M', 'American',' 342-76-9087', '5676 Washington Street','High School', 5, 10) ;

return 'Record Inserted' ;

end;

$$

;

-- test the procedure

select \* from raw\_applicant\_staging ;

call prc\_add\_applicant() ;

select \* from raw\_applicant\_staging ; -- procedure works

-- clean up the table

delete from raw\_applicant\_staging ;

-- create task to call a procedure

create or replace task tsk\_add\_applicant\_call\_prc

warehouse = compute\_wh

schedule = '1 MINUTE'

as

call prc\_add\_applicant() ;

-- show

show tasks ;

-- start the task and check

alter task tsk\_add\_applicant\_call\_prc resume ;

-- check data

select \* from raw\_applicant\_staging ;

-- suspend the task post validations

alter task tsk\_add\_applicant\_call\_prc suspend ;

-- cleanup

drop task if exists tsk\_add\_applicant\_call\_prc ;

--CREATE TASK BASED ON CONDITION (USING WHEN)

--- CREATE TASK USING WHEN --

-- WHEN <Boolean Expression>

-- Only function allowed in WHEN is : system$stream\_has\_data('Stream\_name') -- that we will see when we combine the streams and tasks

-- Let's understand the True and False date conditions

-- clean up the table

delete from raw\_applicant\_staging ;

select getdate() ; -- today's date and time

select dateadd(day,-1,getdate()) ; -- yesterday

select dateadd(day,1,getdate()) ; -- tomorrow

-- lets create a task using when

create or replace task tsk\_add\_applicant\_evry\_min\_cond\_fnc -- [SYSTEM$GET\_PREDECESSOR\_RETURN\_VALUE, SYSTEM$STREAM\_HAS\_DATA]

warehouse = compute\_wh

schedule = '1 MINUTE'

when getdate() between dateadd(day,1,getdate()) and dateadd(day,2,getdate()) -- A true condition

as

insert into raw\_applicant\_staging (first\_name, last\_name, sex, ethinicity, ssn, street\_address, education\_level,years\_of\_experience,job\_id)

values ('James', 'Schwartz', 'M', 'American','342-76-9087','5676 Washington Street','High School', 5,10) ;

-- Invalid expression for task condition expression. Expecting one of the following: [SYSTEM$STREAM\_HAS\_DATA]

-- It can not accept any other function

-- create task using conditions false and true and not using any function

create or replace task tsk\_add\_applicant\_evry\_min\_cond\_false

warehouse = compute\_wh -- optional, we are cereating a task which uses the user defined compute power. Hence gave the warehouse

schedule = '120 MINUTE'

when false -- A False condition

as insert into raw\_applicant\_staging (first\_name, last\_name, sex, ethinicity, ssn, street\_address, education\_level,years\_of\_experience,job\_id)

values ('James', 'Schwartz', 'M', 'American','342-76-9087','5676 Washington Street','High School', 5,10) ;

-- create task with a true condition

create or replace task tsk\_add\_applicant\_evry\_min\_cond\_true

warehouse = compute\_wh -- optional, we are cereating a task which uses the user defined compute power. Hence gave the warehouse

schedule = '1 MINUTE'

when true -- A true condition

as insert into raw\_applicant\_staging (first\_name, last\_name, sex, ethinicity, ssn, street\_address, education\_level,years\_of\_experience,job\_id)

values ('Jessica', 'Escobar', 'F', 'Hispanic','456-93-5629','3234 WateringCan Drive','Undergrad', 4,10) ;

show tasks ;

select \* from raw\_applicant\_staging ;

alter task tsk\_add\_applicant\_evry\_min\_cond\_false resume ;

alter task tsk\_add\_applicant\_evry\_min\_cond\_true resume ;

select \* from raw\_applicant\_staging ;

alter task tsk\_add\_applicant\_evry\_min\_cond\_false suspend ;

alter task tsk\_add\_applicant\_evry\_min\_cond\_true suspend ;

-- cleanup

drop task if exists tsk\_add\_applicant\_evry\_min\_cond\_false ;

drop task if exists tsk\_add\_applicant\_evry\_min\_cond\_true ;

— TROUBLESHOOTING THE TASKS –

-- Now we know that Jessica is inserted but James is not getting inserted..

-- How can we check, troubleshoot the tasks...

-- Use TASK\_HISTORY()

-- get specific task

select \* from table(information\_schema.task\_history()) where lower(name) in ('tsk\_add\_applicant\_evry\_min\_cond\_false') order by completed\_time desc;

-- get tasks in last one hour

select \* from table(information\_schema.task\_history()) where completed\_time between dateadd(hour, -1, getdate()) and getdate() order by completed\_time desc;

-- can add the conditions of query one and two to get the specific task details via time

select \* from table(information\_schema.task\_history()) where completed\_time between dateadd(hour, -2, getdate()) and getdate()

and lower(name) in ('tsk\_add\_applicant\_evry\_min\_cond\_true')

order by completed\_time desc;

-- show

show tasks ; -- all should be in suspended mode

-- cleanup

drop task if exists TSK\_ADD\_APPLICANT\_EVRY\_MIN\_COND\_FALSE ;

drop task if exists TSK\_ADD\_APPLICANT\_EVRY\_MIN\_COND\_TRUE ;

------------------------------------------------------------

--- MORE on SCHEDULING TASK -- USING CRON Expression

--- SCHEDULING TASKS USING CRON --

show parameters like '%timezone%' ;

select \* from raw\_applicant\_staging ;

create or replace task tsk\_add\_applicant\_cron

warehouse = compute\_wh

schedule = 'USING CRON 10 22 5 SEP \* Asia/Calcutta'

as insert into raw\_applicant\_staging (first\_name, last\_name, sex, ethinicity, ssn, street\_address, education\_level,years\_of\_experience,job\_id)

values ('James', 'Schwartz', 'M', 'American','342-76-9087','5676 Washington Street','High School', 5,10) ;

show tasks;

alter task tsk\_add\_applicant\_cron resume;

-- <1> - (0-59) minute

-- <2> - (0-23) hour

-- <3> - (1-31, L) day of month

-- <4> - (1-12, JAN,FEB,MAR..DEC) month

-- <5> - (0-6,SUN,MON,TUE..SAT, L) day of week

-- <6> - (America/Los\_Angeles, America/New\_York, 'America/Chicago') timezone

-- EmptyValue is \*

-- some examples

-- Runs at 4.50 AM every day as per LA time zone

schedule = USING CRON 50 4 \* \* \* America/Los\_Angeles

-- Runs at 4.50 AM on 2nd day of each month as per LA time zone

schedule = USING CRON 50 4 2 \* \* America/Los\_Angeles

-- Runs at 4.50 AM on 2nd day of Oct as per LA time zone

schedule = USING CRON 50 4 2 10 \* America/Los\_Angeles

-- Runs at 4.50 AM in Oct every monday as per LA time zone

schedule = USING CRON 50 4 \* 10 1 America/Los\_Angeles

-- Runs every hour from 4:00 AM to 8:00 PM on last day of each month

schedule = USING CRON 0 4-20 L \* \* America/Los\_Angeles

-- Runs every hour from 4:00 AM to 8:00 PM on last day of each week i.e SAT

schedule = USING CRON 0 4-20 \* \* L America/Los\_Angeles