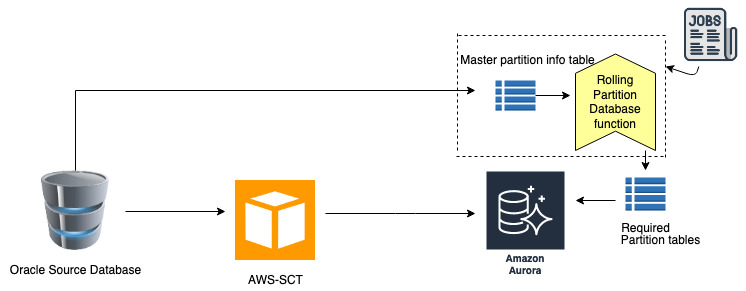
**HANDLING INTERVAL PARTITION IN POSTGRESQL**



**ROLLING PARTITIONS**

**Description**

Interval partition options is not there is PostgreSQL, we are handling it by proactively creating the partitions in prior using with a custom logic, we call it as a rolling partition as it can be scheduled as a job which will take care of partition table creations continuously.

****

Migrate the oracle interval partition tables using aws-sct deploy them in the target aurora PostgreSQL database later on use the rolling partition logic to achieve the functionality.

**Topics covered by the logic.**

* Interval partition table using below conditions

1. MONTH
2. YEAR
3. QUARTER

**Objects developed for the logic**

Create Rolling partition is a standalone function which can be called from anywhere to create the required partition tables, below are the list of database objects developed for achieving the functionality.

* Schema “aws\_utility “

This is the schema which need to be created as all the developed objects will be deployed inside this schema, you can find the script to create this schema in attachment “create\_schema.sql”.

* Master Table “master\_partition\_info”

This is the master table which will hold the information about the partition table’s,

Data need to be populated into this table from source oracle, script to generate the data is there in the attachment

* Type “rolling\_partitions\_array”

This is a type which need to be created in aws\_utility schema which will be a return type of rolling partition function.

* Function “create\_rolling\_partitions”

This is the main function which will create the required partition tables.this function takes three inputs (schema\_name,table\_name & number of partitions required),output of this function is number of partitions created.

* Function “rolling\_partitions”

This function is created, if incase the partitions need to be created for entire schema the we need to pass the schema\_name for this function, it will internally calls create\_rolling\_partitions function and creates all the required partitions,out of this function will be a array of table\_names,number of partitions created.

**Steps to deploy the scripts.**

1. Connect to the database as a super user who have privilege of connecting and creating the database objects
2. Open create\_schema.sql and execute it
3. Open create\_type.sql and execute it.
4. Open master\_partition\_info.sql file & create “master\_partition\_info” table
5. Copy the master\_partition\_info query from the same file execute it in source oracle copy the output of the query as a insert scripts & execute them in the target database, which will populate the required info about the partition tables
6. Open “create\_rolling\_partitions.sql” file & execute it.
7. Open “rolling\_partitions.sql” file & execute it.

**Note:** if needed you can use the sct converted sample partition tables in the attachment,open “sample\_pg\_partition\_tables.sql” & execute it, it will create one scheme called “partitions\_test\_schema” & 4 partition tables

If you are using the sample tables then use the “master\_partition\_info” sample insert scripts as well for testing.

Below are some sample commands

* Running for entire schema

Syntax: select aws\_utility.rolling\_partitions(schema\_name,number of partitions required);

i..e

select aws\_utility.rolling\_partitions('partitions\_test\_schema',12);

* Running for individual tables

Syntax:

select aws\_utility.rolling\_partitions(schema\_name,table\_name,number of partitions required, execute date);

It will consider the default execute date as a current data if not specified

i.e:

select aws\_utility.create\_rolling\_partitions('partitions\_test\_schema','malmrablgmst',12);

select aws\_utility.create\_rolling\_partitions('partitions\_test\_schema','malmrablgmst',12,current\_date-30);

select aws\_utility.create\_rolling\_partitions('partitions\_test\_schema','malmrablgmst',12,current\_date+60);

select aws\_utility.create\_rolling\_partitions('partitions\_test\_schema','test\_partition\_interval\_year',12,current\_date-60);

select aws\_utility.create\_rolling\_partitions('partitions\_test\_schema','test\_partition\_interval\_year',12,current\_date);