Generating Query Pattern:

/*Q1. For one Application, get the list of the warehouse from the APP Mapping table*/

Collect all warehouse name for your application

/*Q2. Load the last 10days of data into a local table for one warehouse*/

```
create or replace  table demo_db.public.query_trend DATA_RETENTION_TIME_IN_DAYS =0 as
select to_date(start_time)as start_dt,
total_elapsed_time,CLUSTER_NUMBER,PERCENTAGE_SCANNED_FROM_CACHE,QUERY_TYPE,QUERY_L
OAD_PERCENT
From

"SNOWFLAKE_METADATA"."CUSTOM_VIEWS.<FunctionShortname>_<AppShortname>_QUERY_HISTOR
Y_V
where to_date(start_time)>=current_date-10 and to_date(start_time)<current_date
and warehouse_name=<Warehosue Name>
and warehouse_size is not null ;
```

/*Q3. Below query results date wise count of queries running between
"Running in less than 1sec",
"Running between 1sec and 1min",
"Running between 1min and 3min",
"Running between 3min and 10min",
"Running greater than 10min".*/

```
select start_dt,
        SUM(CASE WHEN  total_elapsed_time_sec<=1 THEN 1 ELSE 0 END) AS "Running in less than
1sec",
        SUM(CASE WHEN total_elapsed_time_sec >1 and total_elapsed_time_sec<=60 THEN 1 ELSE 0
END) AS "Running between 1sec and 1min",
        SUM(CASE WHEN total_elapsed_time_sec >60 and total_elapsed_time_sec<=180 THEN 1 ELSE 0
END) AS "Running between 1min and 3min",
        SUM(CASE WHEN total_elapsed_time_sec >180 and total_elapsed_time_sec<=600 THEN 1 ELSE
0 END) AS "Running between 3min and 10min",
        SUM(CASE WHEN total_elapsed_time_sec >600 THEN 1 ELSE 0 END) AS "Running greater than
10min",
        count(1) as tot_query_cnt
FROM (select start_dt,total_elapsed_time/1000 as total_elapsed_time_sec from
demo_db.public.query_trend)a
GROUP BY 1 ORDER BY 1 ASC;
```

/*Q4. Below query results date wise % of queries running between
"Running in less than 1sec",
"Running between 1sec and 1min",
"Running between 1min and 3min",
"Running between 3min and 10min",
"Running greater than 10min". */

```
select start_dt,
        count(1) as "Total query count",
        (SUM(CASE WHEN  total_elapsed_time_sec<=1 THEN 1 ELSE 0 END)/"Total query count")*100
AS "Running in less than 1sec",
        (SUM(CASE WHEN total_elapsed_time_sec >1 and total_elapsed_time_sec<=60 THEN 1 ELSE 0
END)/"Total query count")*100 AS "Running between 1sec and 1min",
        (SUM(CASE WHEN total_elapsed_time_sec >60 and total_elapsed_time_sec<=180 THEN 1 ELSE
0 END)/"Total query count")*100 AS "Running between 1min and 3min",
        (SUM(CASE WHEN total_elapsed_time_sec >180 and total_elapsed_time_sec<=600 THEN 1 ELSE
0 END)/"Total query count")*100 AS "Running between 3min and 10min",
        (SUM(CASE WHEN total_elapsed_time_sec >600 THEN 1 ELSE 0 END)/"Total query count")*100
AS "Running greater than 10min"
   FROM (select start_dt,total_elapsed_time/1000 as total_elapsed_time_sec from
demo_db.public.query_trend)a
   GROUP BY 1 ORDER BY 1 ASC;
```

/*Q5. Below query results date wise number of queries utilizing warehouse cache
"0% Warehouse Cache",
"Less than 10% Warehouse Cache",
"Between 10% and 30% Warehouse Cache",
"Between 30% and 50% Warehouse Cache",
"Between 50% and 70% Warehouse Cache",
"More than 70% Warehouse Cache". */

```
select start_dt,
        SUM(CASE WHEN PERCENTAGE_SCANNED_FROM_CACHE=0 THEN 1 ELSE 0 END) AS "0%
Warehosue Cache",
        SUM(CASE WHEN PERCENTAGE_SCANNED_FROM_CACHE>0 AND
PERCENTAGE_SCANNED_FROM_CACHE<10 THEN 1 ELSE 0 END) AS "Less than 10% Warehosue Cache",
        SUM(CASE WHEN PERCENTAGE_SCANNED_FROM_CACHE>=10 AND
PERCENTAGE_SCANNED_FROM_CACHE<30 THEN 1 ELSE 0 END) AS "Between 10% and 30% Warehosue
Cache",
        SUM(CASE WHEN PERCENTAGE_SCANNED_FROM_CACHE>=30 AND
PERCENTAGE_SCANNED_FROM_CACHE<50 THEN 1 ELSE 0 END) AS "Between 30% and 50% Warehosue
Cache",
        SUM(CASE WHEN PERCENTAGE_SCANNED_FROM_CACHE>=50 AND
PERCENTAGE_SCANNED_FROM_CACHE<70 THEN 1 ELSE 0 END) AS "Between 50% and 70% Warehosue
Cache",
```

```
        SUM(CASE WHEN PERCENTAGE_SCANNED_FROM_CACHE>=70 THEN 1 ELSE 0 END) AS "More
than 70% Warehosue Cache"
FROM  demo_db.public.query_trend
GROUP BY 1 ORDER BY 1
```

/* Q6. For each QUERY_TYPE below query results date wise average runtime and query count. */

```
SELECT start_dt,QUERY_TYPE,AVG(total_elapsed_time)/1000 as total_elapsed_in_Sec,COUNT(1) FROM
demo_db.public.query_trend
GROUP BY 1,2 ORDER BY 1,2
```

/* Q7. For each QUERY_TYPE below query results, average runtime, and query count. */

```
SELECT QUERY_TYPE,AVG(total_elapsed_time)/1000,COUNT(1) FROM
demo_db.public.query_trend
GROUP BY 1 ORDER BY 1
```

/* Q8. Below query results date wise average load on the warehouse. */

```
SELECT start_dt,AVG(QUERY_LOAD_PERCENT)
FROM  demo_db.public.query_trend
GROUP BY 1 ORDER BY 1
```

/* Q9. The below query results in a date wise number of queries running on each cluster. */

```
SELECT start_dt,CLUSTER_NUMBER,COUNT(1)
FROM  demo_db.public.query_trend
GROUP BY 1,2 ORDER BY 1,2;
```

Repeat process (2 to 9) for all the warehouse.

Identifying long running /recurring queries:

/*Q1. For one Application, get the list of the warehouse from the APP Mapping table*/

Collect all warehouse name for your application

/*Q2. Load the last 15days of data into a local table for one warehouse*/

```
CREATE OR REPLACE table demo_db.public.WH_LONG_QUERY1 DATA_RETENTION_TIME_IN_DAYS =0
AS
select warehouse_name, warehouse_size, database_name, schema_name, query_id, query_text,
QUERY_TYPE, user_name, total_elapsed_time/(60000) as total_elapsed_time_min ,
start_time, end_time, ROW_NUMBER() over (PARTITION BY warehouse_name, warehouse_size,
query_text ORDER BY start_time DESC ) row_nr
from
"SNOWFLAKE_METADATA"."CUSTOM_VIEWS.<FunctionShortName>_<AppShortName>_QUERY_HISTO
RY_V
where to_date(start_time)>= current_date-15 and to_date(start_time)<current_date and
warehouse_size is not null and warehouse_name ='<Warehouse name>'
order by total_elapsed_time desc;


create or replace table  demo_db.public.Query_WH_15_days DATA_RETENTION_TIME_IN_DAYS =0 as
select a.warehouse_name, a.warehouse_size, a.query_text, a.cnt, a.AVG_TIME as Avg_Time_Min,
b.query_id As Last_Query_Id, start_time as Last_Start_Tm, end_time As Last_End_Tm
from
(
SELECT warehouse_name,warehouse_size,query_text,COUNT(1) cnt,AVG(total_elapsed_time_min)
AVG_TIME
FROM  demo_db.public.WH_LONG_QUERY1
GROUP BY warehouse_name,warehouse_size,query_text
)a
inner join
(select distinct warehouse_name,warehouse_size,query_text,query_id,start_time,end_time
from demo_db.public.WH_LONG_QUERY1 where row_nr=1 )b
on (a.warehouse_name=b.warehouse_name and a.query_text=b.query_text and
coalesce(a.warehouse_size,'')=coalesce(b.warehouse_size,''))
--where AVG_TIME>5
order by AVG_TIME desc, cnt desc
```

/*Q3. Top 50 queries by number of runs*/

```
select warehouse_name, warehouse_size, query_text, cnt, Avg_Time_Min, Last_Query_Id,
Last_Start_Tm, Last_End_Tm
from demo_db.public.Query_WH_15_days
order by cnt desc, Avg_Time_Min desc limit 50
```

/*Q4. Top 50 queries by duration*/

```
select warehouse_name, warehouse_size, query_text, cnt, Avg_Time_Min, Last_Query_Id,
Last_Start_Tm, Last_End_Tm
from demo_db.public.Query_WH_15_days
order by Avg_Time_Min desc, cnt desc
limit 50
```

Repeat process (2 to 4) for all the warehouse.