Solving analytical queries on Redshift Cluster

Here, you have to write the query used for solving the question and the screenshots of the table which is outputted after the query is run on the AWS Redshift Query editor UI.

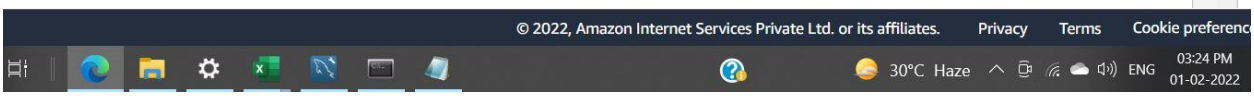
1. **Top 10 ATMs where most transactions are in the 'inactive' state**

```
-- 1


SELECT A.atm_number,
       A.atm_manufacturer,
       L.location,
       IC.inactive_count
FROM atm_etl.dim_location L,
     atm_etl.dim_atm A,


  (SELECT F.atm_id,
          count(F.atm_id) AS inactive_count
   FROM atm_etl.fact_atm_trans F
   WHERE F.atm_status = 'Inactive'
   GROUP BY F.atm_id) IC
WHERE IC.atm_id = A.atm_id
  AND A.atm_location_id = L.location_id
ORDER BY IC.inactive_count DESC
LIMIT 10;
```

atm_number	atm_manufacturer	location	inactive_count
atm_16	NCR	Skive	44043
atm_12	NCR	Aalborg Hallen	33982
atm_2	NCR	Skipperen	33725
atm_88	NCR	Aalborg Storcenter indg. D	32183
atm_30	NCR	NykÃƒÂ¸bing Mors	30883
atm_52	NCR	FarsÃƒÂ¸	27361
atm_50	NCR	Aarhus	23416
atm_29	NCR	Skelagervej 15	20773
atm_81	NCR	Spar KÃƒÂ¸bmand TornhÃƒÂ¸j	20148
atm_102	NCR	Storcenter indg. A	18297

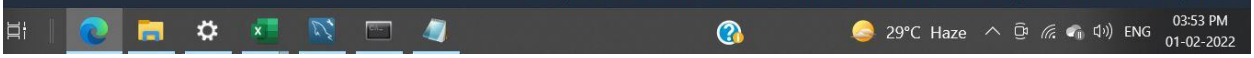


2. **Number of ATM failures corresponding to the different weather conditions recorded at the time of the transactions**

```
--2


SELECT SQ.weather_main,
       count(SQ.status) AS total,
       sum(SQ.status) AS inactive,
       100.0* sum(SQ.status)/count(SQ.status) AS percent_inactive
FROM
  (SELECT F.weather_main,
          F.atm_status,
          CASE
              WHEN F.atm_status = 'Active' THEN 0
              ELSE 1
          END AS status
   FROM atm_etl.fact_atm_trans F) SQ
GROUP BY SQ.weather_main
ORDER BY percent_inactive DESC;
```

weather_main	total	inactive	percent_inactive
Snow	23405	4813	20.563982055116428
Fog	18174	3729	20.518322878837900
	8087	1645	20.341288487696302
Clouds	1181901	194027	16.416518811643276
Rain	545135	86017	15.779027213442541
Clear	543949	85531	15.724084427032681
Mist	82801	12864	15.536044250673301
Thunderstorm	2549	361	14.162416633974107
Drizzle	62530	8670	13.865344634575403
TORNADO	38	1	2.631578947368421

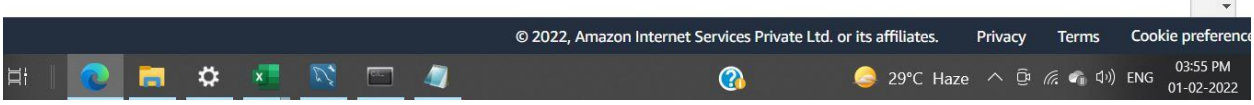


3. **Top 10 ATMs with the most number of transactions throughout the year**

```sql
--3


SELECT A.atm_number,
       A.atm_manufacturer,
       L.location,
       count(A.atm_number) AS total_trans
FROM atm_etl.fact_atm_trans F
LEFT JOIN atm_etl.dim_atm A ON A.atm_id = F.atm_id
LEFT JOIN atm_etl.dim_location L ON L.location_id = A.atm_location_id
GROUP BY A.atm_number,
         A.atm_manufacturer,
         L.location
ORDER BY total_trans DESC
LIMIT 10;
```

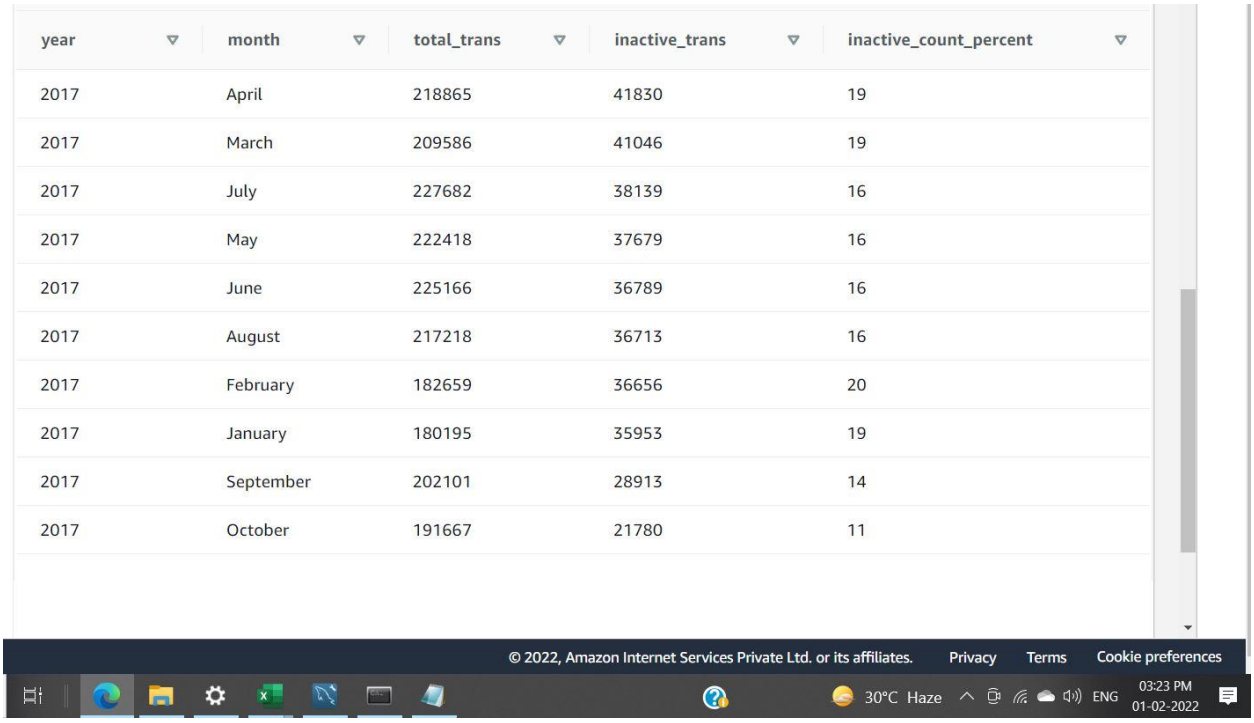
atm_number	atm_manufacturer	location	total_trans
atm_39	NCR	Svenstrup	55380
atm_20	NCR	HjÃƒÂ¸rring	54211
atm_10	NCR	NÃƒÂ¦rresundby	53794
atm_24	NCR	Hobro	53378
atm_45	NCR	Frederikshavn	53198
atm_16	NCR	Skive	44043
atm_40	Diebold Nixdorf	Abildgaard	43767
atm_1	NCR	NÃƒÂ¦stved	42787
atm_41	Diebold Nixdorf	Skagen	42732
atm_48	Diebold Nixdorf	Intern BrÃƒÂ¸nderslev	42493



4. **Number of overall ATM transactions going inactive per month for each month**

```
--4


SELECT YEAR,
       MONTH,
       count(status) AS total_trans,
       sum(status) AS inactive_trans,
       100 * sum(status)/count(status) AS inactive_count_percent
FROM
  (SELECT D.year,
          D.month,
          CASE
              WHEN F.atm_status = 'Active' THEN 0
              ELSE 1
          END AS status
   FROM atm_etl.fact_atm_trans F
   INNER JOIN atm_etl.dim_date D ON F.date_id = D.date_id) SQ
GROUP BY YEAR,
         MONTH
ORDER BY inactive_trans DESC;
```



year	month	total_trans	inactive_trans	inactive_count_percent
2017	April	218865	41830	19
2017	March	209586	41046	19
2017	July	227682	38139	16
2017	May	222418	37679	16
2017	June	225166	36789	16
2017	August	217218	36713	16
2017	February	182659	36656	20
2017	January	180195	35953	19
2017	September	202101	28913	14
2017	October	191667	21780	11

5. **Top 10 ATMs with the highest total withdrawn amount throughout the year**

```sql
--5


SELECT A.atm_number,
       A.atm_manufacturer,
       L.location,
       sum(F.transaction_amount) AS total_withdrawn
FROM atm_etl.fact_atm_trans F
LEFT JOIN atm_etl.dim_atm A ON A.atm_id = F.atm_id
LEFT JOIN atm_etl.dim_location L ON L.location_id = A.atm_location_id
WHERE F.service = 'Withdrawal'
  AND F.atm_status = 'Active'
GROUP BY A.atm_number,
         A.atm_manufacturer,
         L.location
ORDER BY total_withdrawn DESC
LIMIT 10;
```

atm_number	atm_manufacturer	location	total_withdrawn
atm_39	NCR	Svenstrup	277097637
atm_20	NCR	HjÃƒÂ¸rring	271008803
atm_24	NCR	Hobro	268289882
atm_10	NCR	NÃƒÂ¸rresundby	267379103
atm_45	NCR	Frederikshavn	265639616
atm_40	Diebold Nixdorf	Abildgaard	219812287
atm_41	Diebold Nixdorf	Skagen	214127315
atm_1	NCR	NÃƒÂ¦stved	213721117
atm_48	Diebold Nixdorf	Intern BrÃƒÂ¸nderslev	212883099
atm_13	NCR	SÃƒÂ¦by	205905693

6. **Number of failed ATM transactions across various card types**

```
--6


SELECT SQ.card_type,
       count(SQ.status) AS total,
       sum(SQ.status) AS inactive,
       100.0 * sum(SQ.status)/count(SQ.status) AS percent_inactive
FROM
  (SELECT CT.card_type,
          CASE
              WHEN F.atm_status = 'Active' THEN 0
              ELSE 1
          END AS status
   FROM atm_etl.fact_atm_trans F
   INNER JOIN atm_etl.dim_card_type CT ON CT.card_type_id = F.card_typ
e_id) SQ
GROUP BY card_type
ORDER BY percent_inactive DESC;
```

card_type	total	inactive	percent_inactive
Mastercard - on-us	458226	86000	18.768031495375644
VISA	170828	30713	17.978902755988479
Dankort - on-us	143813	24680	17.161174580879336
CIRRUS	17362	2953	17.008409169450524
HÃƒÂ¦vekort - on-us	62487	10331	16.533038872085393
Dankort	28581	4557	15.944158706833210
MasterCard	400507	63482	15.850409605824617
Visa Dankort - on-us	748805	112972	15.086971908574328
HÃƒÂ¦vekort	8459	1208	14.280647830712850
Visa Dankort	427840	60547	14.151785714285714

7. **Number of transactions happening on an ATM on weekdays and on weekends throughout the year. Order this by the ATM_number, ATM_manufacturer, location, weekend_flag and then total_transaction_count**

<Query>

<Screenshot of the resultant table>

8. **Most active day in each ATMs from location "Vejgaard"**

<Query>

<Screenshot of the resultant table>