

# Solving Analytical Queries On RedShift Cluster

~Arun Tyagi

## Screenshot of S3:

Amazon S3 > artyagi-s3-bucket-etl-project

### artyagi-s3-bucket-etl-project [Info](#)

**Objects** | Properties | Permissions | Metrics | Management | Access Points

**Objects (5)**

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

[Refresh](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#)

[Upload](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	<a href="#">DIM_ATM/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">DIM_CARD/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">DIM_DATE/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">DIM_LOC/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">FACT_ATM_TRANS/</a>	Folder	-	-	-

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

```
SELECT a.atm_id, a.atm_manufacturer, b.atm_location, count(d.trans_id) as transaction_count,
count(d.atm_status)
from etl.atm a, etl.loc b, etl.FACT_ATM_TRANS d
where d.atm_status = 'Inactive'
and d.atm_prim_id = a.atm_prim_id
and b.location_id = d.location_id
group by a.atm_id, a.atm_manufacturer, b.atm_location
order by transaction_count desc
limit 10
```

The screenshot shows a SQL query editor interface. At the top, there's a status bar indicating 'Status: Connected' with a green checkmark. Below it, a 'Change connection' button is visible. The main area displays the SQL query from the previous block. At the bottom, there are four buttons: 'Run' (orange), 'Save', 'Schedule', and 'Clear'. To the right of these buttons is a 'Send feedback' link with a speech bubble icon.

Rows returned (10)					Export ▼
<input type="text" value="Search rows"/>					< 1 > ⚙
atm_id ▼	atm_manufacturer ▼	atm_location ▼	transaction_count ▼	count ▼	
16	NCR	Skive	44043	44043	
12	NCR	ÅfEøesterÅfÅ¥ Duus	33982	33982	
2	NCR	Vejgaard	33725	33725	
88	NCR	Storcenter indg. A	32183	32183	
30	NCR	NykÅfÅ, bing Mors	30883	30883	
52	NCR	FarsÅfÅ,	27361	27361	
50	NCR	Aarhus	23416	23416	
29	NCR	Skelagervej 15	20773	20773	
81	NCR	Spær ÅfÅ, bmand TornhÅfÅ, j	20148	20148	
102	NCR	Aalborg Storcenter Afd	18297	18297	

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

```
select c.weather_main, c.total_transaction_count,  
NVL(d.inactive_count::int,0) as total_inactive_count,  
round(100.0000*total_inactive_count/c.total_transaction_count,4) as inactive_count_percent  
from  
(select a.weather_main, count(a.trans_id) as total_transaction_count from  
etl.FACT_ATM_TRANS a where a.weather_main !=' ' group by a.weather_main)  
c left outer join  
(select b.weather_main, count(b.atm_status) as inactive_count from  
etl.FACT_ATM_TRANS b where b.atm_status='Inactive' and b.weather_main !=' ' group  
by b.weather_main) d  
on c.weather_main=d.weather_main  
group by c.weather_main,c.total_transaction_count, total_inactive_count  
order by inactive_count_percent desc;
```

```
10  
11 select c.weather_main, c.total_transaction_count,  
12 NVL(d.inactive_count::int,0) as total_inactive_count,  
13 round(100.0000*total_inactive_count/c.total_transaction_count,4) as inactive_count_percent  
14 from  
15 (select a.weather_main, count(a.trans_id) as total_transaction_count from  
16 etl.FACT_ATM_TRANS a where a.weather_main !=' ' group by a.weather_main)  
17 c left outer join  
18 (select b.weather_main, count(b.atm_status) as inactive_count from  
19 etl.FACT_ATM_TRANS b where b.atm_status='Inactive' and b.weather_main !=' ' group  
20 by b.weather_main) d  
21 on c.weather_main=d.weather_main  
22 group by c.weather_main,c.total_transaction_count, total_inactive_count  
23 order by inactive_count_percent desc;  
24
```

weather_main	total_transaction_count	total_inactive_count	inactive_count_percent
Snow	23405	4813	20.5640
Fog	18174	3729	20.5183
Clouds	1181901	194027	16.4165
Rain	545135	86017	15.7790
Clear	543949	85531	15.7241
Mist	82801	12864	15.5360
Thunderstorm	2549	361	14.1624
Drizzle	62530	8670	13.8653
TORNADO	38	1	2.6316
Haze	3	0	0.0000

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

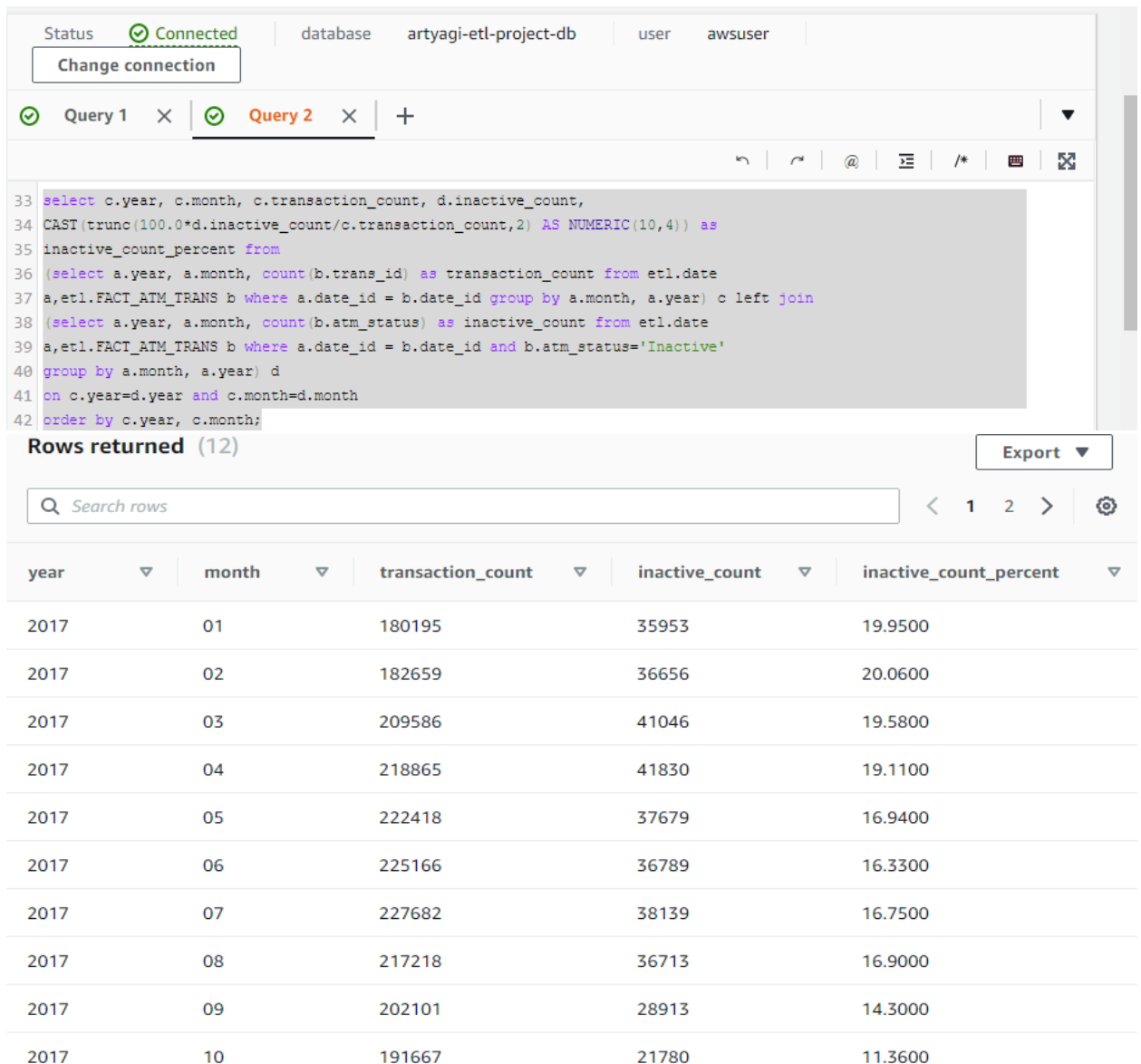
```
SELECT a.atm_id, a.atm_manufacturer, b.atm_location, count(d.trans_id) as
transaction_count
from etl.atm a, etl.loc b, etl.FACT_ATM_TRANS d
where d.atm_prim_id = a.atm_prim_id
and b.location_id = d.location_id
group by a.atm_id, a.atm_manufacturer, b.atm_location
order by transaction_count desc
limit 10
```

```
25 SELECT a.atm_id, a.atm_manufacturer, b.atm_location, count(d.trans_id) as transaction_count
26 from etl.atm a, etl.loc b, etl.FACT_ATM_TRANS d
27 where d.atm_prim_id = a.atm_prim_id
28 and b.location_id = d.location_id
29 group by a.atm_id, a.atm_manufacturer, b.atm_location
30 order by transaction_count desc
31 limit 10
32
33
```

atm_id	atm_manufacturer	atm_location	transaction_count
39	NCR	Svenstrup	55380
20	NCR	Bispensgade	54211
10	NCR	NÃfÃ, rresundby	53794
24	NCR	Hobro	53378
45	NCR	Abildgaard	53198
16	NCR	Skive	44043
40	Diebold Nixdorf	Frederikshavn	43767
1	NCR	NÃfÃ, stved	42787
41	Diebold Nixdorf	Skagen	42732
48	Diebold Nixdorf	BrÃfÃ, nderslev	42493

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

```
select c.year, c.month, c.transaction_count, d.inactive_count,  
CAST(trunc(100.0*d.inactive_count/c.transaction_count,2) AS NUMERIC(10,4)) as  
inactive_count_percent from  
(select a.year, a.month, count(b.trans_id) as transaction_count from etl.date  
a,etl.FACT_ATM_TRANS b where a.date_id = b.date_id group by a.month, a.year) c left join  
(select a.year, a.month, count(b.atm_status) as inactive_count from etl.date  
a,etl.FACT_ATM_TRANS b where a.date_id = b.date_id and b.atm_status='Inactive'  
group by a.month, a.year) d  
on c.year=d.year and c.month=d.month  
order by c.year, c.month;
```



The screenshot shows a database query tool interface. At the top, it indicates a 'Connected' status with a green checkmark, the database name 'artyagi-etl-project-db', and the user 'awsuser'. Below this is a 'Change connection' button. The main area displays a SQL query in a text editor, with line numbers 33 through 42. The query is a complex join between two subqueries to calculate the percentage of inactive ATM transactions per month. Below the query editor, it shows 'Rows returned (12)'. At the bottom, there is a table with 5 columns: 'year', 'month', 'transaction\_count', 'inactive\_count', and 'inactive\_count\_percent'. The table contains 12 rows of data for the year 2017, months 01 through 10.

year	month	transaction_count	inactive_count	inactive_count_percent
2017	01	180195	35953	19.9500
2017	02	182659	36656	20.0600
2017	03	209586	41046	19.5800
2017	04	218865	41830	19.1100
2017	05	222418	37679	16.9400
2017	06	225166	36789	16.3300
2017	07	227682	38139	16.7500
2017	08	217218	36713	16.9000
2017	09	202101	28913	14.3000
2017	10	191667	21780	11.3600

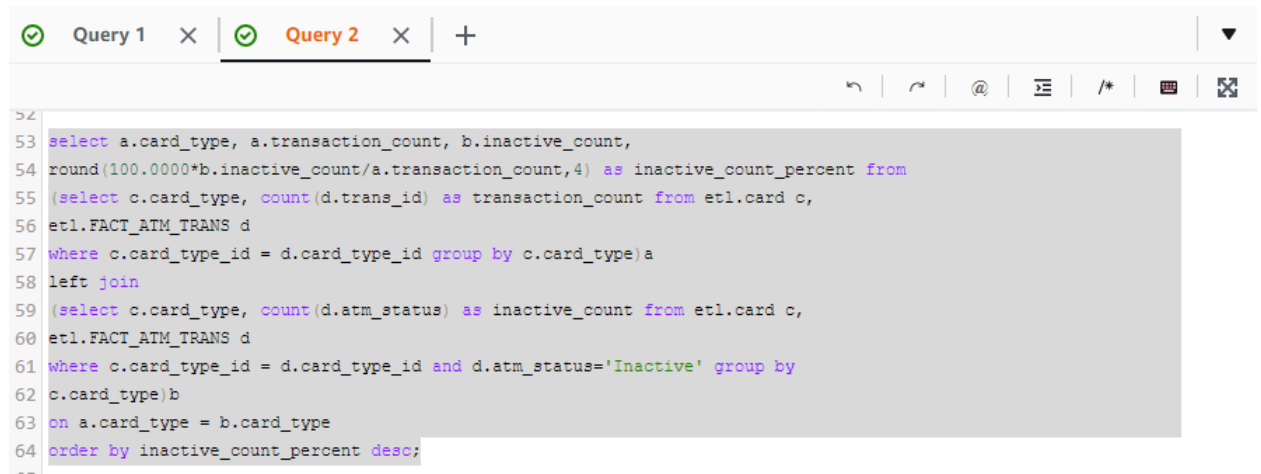
## 5. Top 10 ATMs with the highest total withdrawn amount throughout the year

```
SELECT a.atm_id, a.atm_manufacturer, b.atm_location, sum(d.transaction_amount) as
total_transaction_amount
from etl.atm a, etl.loc b, etl.FACT_ATM_TRANS d
where d.atm_prim_id = a.atm_prim_id
and b.location_id = d.location_id
group by a.atm_id, a.atm_manufacturer, b.atm_location
order by total_transaction_amount desc
limit 10
```

Status	Connected	database	artyagi-etl-project-db	user	awsuser
Change connection					
Query 1	Query 2				
<pre>41 on c.year=d.year and c.month=d.month 42 order by c.year, c.month; 43 44 SELECT a.atm_id, a.atm_manufacturer, b.atm_location, sum(d.transaction_amount) as 45 total_transaction_amount 46 from etl.atm a, etl.loc b, etl.FACT_ATM_TRANS d 47 where d.atm_prim_id = a.atm_prim_id 48 and b.location_id = d.location_id 49 group by a.atm_id, a.atm_manufacturer, b.atm_location 50 order by total_transaction_amount desc 51 limit 10</pre>					
Rows returned (10)					
Search rows					
atm_id	atm_manufacturer	atm_location	total_transaction_amount		
39	NCR	Svenstrup	277097637		
20	NCR	Bispensgade	271008803		
24	NCR	Hobro	268289882		
10	NCR	NÃfÃ,rresundby	267379103		
45	NCR	Abildgaard	265639616		
16	NCR	Skive	220677013		
40	Diebold Nixdorf	Frederikshavn	219812287		
41	Diebold Nixdorf	Skagen	214127315		
1	NCR	NÃfÃ'stved	213721117		
48	Diebold Nixdorf	BrÃfÃ, nderslev	212883099		

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

```
select a.card_type, a.transaction_count, b.inactive_count,
round(100.0000*b.inactive_count/a.transaction_count,4) as inactive_count_percent from
(select c.card_type, count(d.trans_id) as transaction_count from etl.card c,
etl.FACT_ATM_TRANS d where c.card_type_id = d.card_type_id group by c.card_type)a left join
(select c.card_type, count(d.atm_status) as inactive_count from etl.card c,
etl.FACT_ATM_TRANS d where c.card_type_id = d.card_type_id and d.atm_status='Inactive'
group by c.card_type)b on a.card_type = b.card_type order by inactive_count_percent desc;
```

The screenshot shows a SQL query editor with two tabs: 'Query 1' and 'Query 2'. The 'Query 2' tab is active, displaying the SQL query. The query is a complex join between two subqueries. The first subquery calculates the transaction count for each card type, and the second subquery calculates the inactive count for each card type. The results are joined on the card\_type and ordered by the inactive count percentage in descending order. The query is as follows:

```
53 select a.card_type, a.transaction_count, b.inactive_count,
54 round(100.0000*b.inactive_count/a.transaction_count,4) as inactive_count_percent from
55 (select c.card_type, count(d.trans_id) as transaction_count from etl.card c,
56 etl.FACT_ATM_TRANS d
57 where c.card_type_id = d.card_type_id group by c.card_type)a
58 left join
59 (select c.card_type, count(d.atm_status) as inactive_count from etl.card c,
60 etl.FACT_ATM_TRANS d
61 where c.card_type_id = d.card_type_id and d.atm_status='Inactive' group by
62 c.card_type)b
63 on a.card_type = b.card_type
64 order by inactive_count_percent desc;
```

card_type	transaction_count	inactive_count	inactive_count_percent
Mastercard - on-us	458226	86000	18.7680
VISA	170828	30713	17.9789
Dankort - on-us	143813	24680	17.1612
CIRRUS	17362	2953	17.0084
HÃfÃ'vekort - on-us	62487	10331	16.5330
Dankort	28581	4557	15.9442
MasterCard	400507	63482	15.8504
Visa Dankort - on-us	748805	112972	15.0870
HÃfÃ'vekort	8459	1208	14.2806
Visa Dankort	427840	60547	14.1518

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**

```
SELECT a.atm_id, a.atm_manufacturer, b.atm_location,  
CASE c.weekday  
WHEN 'Monday'  
THEN '0'  
WHEN 'Tuesday'  
THEN '0'  
WHEN 'Wednesday'  
THEN '0'  
WHEN 'Thursday'  
THEN '0'  
WHEN 'Friday'  
THEN '0'  
ELSE '1'  
END AS weekend_flag,  
count(d.trans_id) as total_transaction_count  
from etl.atm a, etl.loc b, etl.date c, etl.FACT_ATM_TRANS d  
where d.atm_prim_id = a.atm_prim_id  
and b.location_id = d.location_id  
and c.date_id = d.date_id  
group by a.atm_id, a.atm_manufacturer, b.atm_location, weekend_flag  
order by a.atm_id asc, weekend_flag asc  
limit 10
```

```
SELECT a.atm_id, a.atm_manufacturer, b.atm_location,  
CASE c.weekday  
WHEN 'Monday'  
THEN '0'  
WHEN 'Tuesday'  
THEN '0'  
WHEN 'Wednesday'  
THEN '0'  
WHEN 'Thursday'  
THEN '0'  
WHEN 'Friday'  
THEN '0'  
ELSE '1'  
END AS weekend_flag,
```



```

THEN '0'
ELSE '1'
END AS weekend_flag,
count(d.trans_id) as total_transaction_count
from etl.atm a, etl.loc b, etl.date c, etl.FACT_ATM_TRANS d
where d.atm_prim_id = a.atm_prim_id
and b.location_id = d.location_id
and c.date_id = d.date_id
group by a.atm_id, a.atm_manufacturer, b.atm_location, weekend_flag
order by a.atm_id asc, weekend_flag asc
limit 10

```

atm_id ▾	atm_manufacturer ▾	atm_location ▾	weekend_flag ▾	total_transaction_count ▾
1	NCR	NÃfÃstved	0	32711
1	NCR	NÃfÃstved	1	10076
10	NCR	NÃfÃ, rresundby	0	41667
10	NCR	NÃfÃ, rresundby	1	12127
100	NCR	Intern Skive	0	17812
100	NCR	Intern Skive	1	1
101	NCR	Bryggen Vejle	0	11693
101	NCR	Bryggen Vejle	1	3247
102	NCR	Aalborg Storcenter Afd	0	14556
102	NCR	Aalborg Storcenter Afd	1	3741

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

```
SELECT atm_id,  
atm_manufacturer,  
atm_location,  
weekday,  
total_transaction_count  
FROM (  
select atm_id,  
atm_manufacturer,  
atm_location,  
weekday,  
total_transaction_count,  
max(total_transaction_count) over (partition by atm_id) as max_version  
from (SELECT a.atm_id, a.atm_manufacturer, b.atm_location, c.weekday,  
count(d.trans_id) as total_transaction_count  
from etl.atm a, etl.loc b, etl.date c, etl.FACT_ATM_TRANS d  
where d.atm_prim_id = a.atm_prim_id  
and b.location_id = d.location_id  
and b.atm_location = 'Vejgaard'  
and c.date_id = d.date_id  
group by a.atm_id, a.atm_manufacturer, b.atm_location, c.weekday) c  
) t  
where total_transaction_count = max_version;
```

```
89 SELECT atm_id,  
90 atm_manufacturer,  
91 atm_location,  
92 weekday,  
93 total_transaction_count  
94 FROM (  
95 select atm_id,  
96 atm_manufacturer,  
97 atm_location,  
98 weekday,  
99 total_transaction_count,  
100 max(total_transaction_count) over (partition by atm_id) as max_version  
101 from (SELECT a.atm_id, a.atm_manufacturer, b.atm_location, c.weekday,  
102 count(d.trans_id) as total_transaction_count  
103 from etl.atm a, etl.loc b, etl.date c, etl.FACT_ATM_TRANS d
```

```

99 total_transaction_count,
100 max(total_transaction_count) over (partition by atm_id) as max_version
101 from (SELECT a.atm_id, a.atm_manufacturer, b.atm_location, c.weekday,
102 count(d.trans_id) as total_transaction_count
103 from etl.atm a, etl.loc b, etl.date c, etl.FACT_ATM_TRANS d
104 where d.atm_prim_id = a.atm_prim_id
105 and b.location_id = d.location_id
106 and b.atm_location = 'Vejgaard'
107 and c.date_id = d.date_id
108 group by a.atm_id, a.atm_manufacturer, b.atm_location, c.weekday) c
109 ) t
110 where total_transaction_count = max_version;

```

Rows returned (2)

Export ▼

Q Search rows

< 1 > ⚙

atm_id ▼	atm_manufacturer ▼	atm_location ▼	weekday ▼	total_transaction_count ▼
2	NCR	Vejgaard	Friday	6290
103	Diebold Nixdorf	Vejgaard	Friday	4757