

BigQuery SQL Analysis

By: Arpit Kushwaha

Average number of items per order - daily, monthly, weekly, state, city, pincode:

```
1. SELECT
2. EXTRACT(DAY from order_received_timestamp) AS daily,
3. EXTRACT(MONTH from order_received_timestamp) AS monthly,
4. EXTRACT(WEEK from order_received_timestamp) AS weekly,
5. a.state AS state,
6. a.city AS city,
7. f.pincode AS pincode,
8. avg(item_count) as avg_items
9. From `fractal1a.starschema.fact_daily_orders` f
10. left join `fractal1a.starschema.dim_customer` c
11. on f.customerid = c.customerid
12. left join `fractal1a.starschema.dim_address` a
13. on c.address_id = a.address_id
14. group by
15. daily,
16. monthly,
17. weekly,
18. state,
19. city,
20. pincode
```

The screenshot displays the Google Cloud BigQuery interface. On the left, the 'Explorer' pane shows a project named 'fractal1a' with a 'starschema' dataset containing tables like 'dim_add...', 'dim_cus...', 'dim_order', 'dim_pro...', 'f_order...', and 'fact_dail...'. The main editor shows a SQL query titled 'Untitled 2' with the following code:

```
1 SELECT
2 EXTRACT(DAY from order_received_timestamp) AS daily,
3 EXTRACT(MONTH from order_received_timestamp) AS monthly,
4 EXTRACT(WEEK from order_received_timestamp) AS weekly,
5 a.state AS state,
6 a.city AS city,
7 f.pincode AS pincode,
8 avg(item_count) as avg_items
9 From `fractal1a.starschema.fact_daily_orders` f
10 left join `fractal1a.starschema.dim_customer` c
11 on f.customerid = c.customerid
12 left join `fractal1a.starschema.dim_address` a
13 on c.address_id = a.address_id
14 group by
15 daily,
16 monthly,
17 weekly,
18 state,
19 city,
20 pincode
```

The query has been executed successfully, as indicated by the 'Query completed.' status. Below the query editor, the 'Query results' section is visible, showing a table with columns: Row, daily, monthly, weekly, state, city, and pincode. The results are as follows:

Row	daily	monthly	weekly	state	city	pincode
101	9	9	36	Bihar	Bhagalpur	475393
102	3	1	1	Gujarat	Rajkot	117505
103	12	9	37	Gujarat	Rajkot	117505
104	29	9	39	Gujarat	Rajkot	117505
105	28	4	17	Gujarat	Rajkot	117505
106	20	7	29	Gujarat	Rajkot	117505

At the bottom, the 'Results per page' dropdown is set to 50, and the total number of results is 101 out of 15158.

Average amount of sales per order - daily, monthly, weekly, state, city, pincode

```
1. SELECT
2. EXTRACT(DAY FROM f.order_received_timestamp) AS daily,
3. EXTRACT(MONTH FROM f.order_received_timestamp) AS monthly,
4. EXTRACT(WEEK FROM f.order_received_timestamp) AS weekly,
5. a.state AS state,
6. a.city AS city,
7. f.pincode AS pincode,
8. AVG(order_amount) AS avg_sales
9. From `fractal1a.starschema.fact_daily_orders` f
10. left join `fractal1a.starschema.dim_customer` c
11. ON f.customerid = c.customerid
12. left join `fractal1a.starschema.dim_address` a
13. ON c.address_id = a.address_id
14. group by
15. daily,
16. monthly,
17. weekly,
18. state,
19. city,
20. pincode
```

The screenshot shows the Google Cloud BigQuery console interface. At the top, there's a search bar and navigation icons. The left sidebar shows the 'Explorer' view with a list of workspace resources under the project 'fractal1a', including 'External connections', 'akash', 'layer_1_OLT...', 'order_dataset', 'star_schema', 'starschema', 'dim_add...', 'dim_cus...', 'dim_order', 'dim_pro...', 'f_order...', and 'fact_dail...'. The main area displays a SQL query in a text editor, titled 'Untitled 2'. The query is the same one as shown in the previous block. Below the query editor, the 'Query results' section is visible, showing a table with 6 columns: 'daily', 'monthly', 'weekly', 'state', 'city', and 'pincode'. The table contains 6 rows of data. The 'Preview' tab is selected, showing the first 5 rows of the result set. The bottom status bar indicates 'Results per page: 50' and '101 - 150 of 15158'.

Query results

Row	daily	monthly	weekly	state	city	pincode
101	9	9	36	Bihar	Bhagalpur	475393
102	3	1	1	Gujarat	Rajkot	117505
103	12	9	37	Gujarat	Rajkot	117505
104	29	9	39	Gujarat	Rajkot	117505
105	28	4	17	Gujarat	Rajkot	117505
106	20	7	29	Gujarat	Rajkot	117505

Total number of units sold per day of a product SKU and its monthly trend

```
1. SELECT
2. p.productname, p.sku,
3. EXTRACT(DAY FROM f.order_delivery_timestamp) AS day,
4. EXTRACT(MONTH FROM f.order_delivery_timestamp) AS month,
5. SUM(quantity) AS units_sold
6. FROM `fractal1a.starschema.dim_product` p
7. join `fractal1a.starschema.f_order_details` f
8. on p.productid = f.productid
9. group by productname,
10. sku,
11. day,
12. month
```

The screenshot shows the Google Cloud BigQuery console interface. The top navigation bar includes the Google Cloud logo, a project selector set to 'fractal1', and a search bar. The left sidebar contains the 'Explorer' panel with a search bar and a list of workspace resources under 'fractal1a', including 'External connect...', 'akash', 'layer_1_OLT...', 'order_dataset', 'star_schema', and 'starschema'. The main panel displays a SQL query in a text editor titled 'Untitled 2'. The query is a SELECT statement that joins 'dim_product' and 'f_order_details' tables to calculate the total units sold per day and month for each product SKU. Below the query editor, the 'Query results' section is visible, showing a table with 7 rows of data. The table has columns for 'productname', 'sku', 'day', 'month', and 'units_sold'. The data shows that for the product 'MDH Aata' (SKU '1KG'), the units sold vary by day and month. The bottom of the interface shows the 'Results per page' set to 50, with a total of 17345 results.

```
1 SELECT
2 p.productname, p.sku,
3 EXTRACT(DAY FROM f.order_delivery_timestamp) AS day,
4 EXTRACT(MONTH FROM f.order_delivery_timestamp) AS month,
5 SUM(quantity) AS units_sold
6 FROM `fractal1a.starschema.dim_product` p
7 join `fractal1a.starschema.f_order_details` f
8 on p.productid = f.productid
```

Query results

Row	productname	sku	day	month	units_sold
1	MDH Aata	1KG	24	1	18
2	MDH Aata	1KG	5	1	1
3	MDH Aata	1KG	11	1	8
4	MDH Aata	1KG	29	1	1
5	MDH Aata	1KG	12	1	9
6	MDH Aata	1KG	30	1	1
7	MDH Aata	1KG	20	1	8

Results per page: 50 1 - 50 of 17345

Total Order Amount on daily basis, also to be able to split by product and geography

```
1. SELECT
2. EXTRACT(DATE FROM f.order_received_timestamp) AS daily,
3. o.productid,
4. a.city,
5. sum(order_amount) total_sales
6. FROM `fractal1a.starschema.fact_daily_orders` f
7. join `fractal1a.starschema.f_order_details` o ON f.orderid = o.orderid
8. join `fractal1a.starschema.dim_customer` c ON f.customerid = c.customerid
9. join `fractal1a.starschema.dim_address` a ON c.address_id = a.address_id
10. group by
11. daily,
12. productid,
13. city
14. order by daily
```

The screenshot shows the Google Cloud BigQuery console interface. On the left is the Explorer pane showing the project 'fractal1a' and its schemas. The main editor displays a SQL query titled 'Untitled 2'. Below the query editor, the 'Query results' section is visible, showing a table with 7 rows of data. The table has columns for 'daily' (date), 'productid', 'city', and 'total_sales'. The results are sorted by 'daily'.

Query results table:

Row	daily	productid	city	total_sales
1	2023-01-01	8	Nagpur	14787
2	2023-01-01	24	Nagpur	8964
3	2023-01-01	28	Jodhpur	231
4	2023-01-01	40	Mysore	2685
5	2023-01-01	57	Rourkela	804
6	2023-01-01	59	Jodhpur	13248
7	2023-01-01	60	Pune	12777

Distribution of orders according to area (state, city, pincode etc)

```
1. SELECT
2. a.state,
3. a.city,
4. count(DISTINCT f.orderid) NumberOfOrders
5. FROM `fractal1a.starschema.fact_daily_orders` f
6. join `fractal1a.starschema.dim_customer` c ON f.customerid = c.customerid
7. join `fractal1a.starschema.dim_address` a ON c.address_id = a.address_id
8. group by
9. state,city
```

The screenshot shows the Google Cloud BigQuery console interface. On the left, the 'Explorer' pane displays a project named 'fractal1' with a 'starschema' dataset containing tables like 'fact_daily_orders', 'dim_customer', and 'dim_address'. The main editor shows a SQL query titled 'Untitled 2' that counts distinct orders by state and city. Below the query editor, the 'Query results' section is active, displaying a table with 7 rows of data. The table has columns for 'state', 'city', and 'NumberOfOrders'. The results show data for Karnataka (Bangalore), Gujarat (Surat), Odisha (Cuttack), Bihar (Patna), Bihar (Bhagalpur), Gujarat (Rajkot), and Rajasthan (Jodhpur).

Row	state	city	NumberOfOrders
1	Karnataka	Bangalore	693
2	Gujarat	Surat	636
3	Odisha	Cuttack	606
4	Bihar	Patna	728
5	Bihar	Bhagalpur	820
6	Gujarat	Rajkot	694
7	Rajasthan	Jodhpur	537

Average order amount per customer on daily basis

```
1. SELECT
2. c.customerid AS id,
3. c.name AS name,
4. EXTRACT(DATE FROM order_received_timestamp) date,
5. ROUND(AVG(order_amount),2) AS OrderAmount
6. FROM `fractal1a.starschema.fact_daily_orders` f
7. left join `fractal1a.starschema.dim_customer` c
8. ON f.customerid = c.customerid
9. group by
10.id, date, name
```

The screenshot displays the Google Cloud BigQuery interface. On the left, the 'Explorer' pane shows the project 'fractal1a' with various datasets and schemas. The main editor shows a SQL query titled 'Untitled 2'. The query is a SELECT statement that calculates the average order amount per customer on a daily basis. The query has been executed successfully, as indicated by the 'Query completed.' status. Below the query, the 'Query results' section shows a table with 7 rows of data. The table has columns for Row, id, name, date, and OrderAmount. The results show that customer 31 (Devansh Bhat) has 7 orders with a total amount of 5847.0.

Query results

Row	id	name	date	OrderAmount
1	31	Devansh Bhat	2023-05-15	6327.0
2	31	Devansh Bhat	2023-07-11	5071.5
3	31	Devansh Bhat	2023-08-24	546.0
4	31	Devansh Bhat	2023-05-13	1224.0
5	31	Devansh Bhat	2023-01-06	5791.5
6	31	Devansh Bhat	2023-01-16	22068.0
7	31	Devansh Bhat	2023-09-08	5847.0

Results per page: 50 1 - 50 of 15158

New Customers on daily basis

1. `SELECT`
2. `START_DATE,`
3. `COUNT(customerid) AS NewCustomers`
4. `from `fractal1a.starschema.dim_customer``
5. `where customerid in (select customerid from`
``fractal1a.starschema.dim_customer``
6. `group by customerid having count(*)=1)`
7. `group by start_date`

The screenshot displays the Google Cloud BigQuery console. The top navigation bar includes the Google Cloud logo, a project selector set to 'fractal1', and a search bar. The left sidebar shows the 'Explorer' view with a tree of workspace resources under the 'fractal1a' project, including datasets like 'akash', 'layer_1_OLT...', 'order_dataset', 'star_schema', and 'starschema'. The main panel shows a query editor with a SQL query titled 'Untitled 2'. The query is as follows:

```
1 SELECT
2 START_DATE,
3 COUNT(customerid) AS NewCustomers
4 from `fractal1a.starschema.dim_customer`
5 where customerid in (select customerid from `fractal1a.starschema.dim_customer`
6 group by customerid having count(*)=1)
```

Below the query editor, the 'Query results' section is visible, showing a table with two columns: 'START_DATE' and 'NewCustomers'. The table contains 8 rows of data. The bottom of the interface shows pagination information: 'Results per page: 50' and '1 - 50 of 274'.

Row	START_DATE	NewCustomers
1	2023-04-09	3
2	2023-04-10	1
3	2023-04-12	4
4	2023-04-13	5
5	2023-04-14	3
6	2023-04-15	4
7	2023-04-16	6
8	2023-04-17	4

Total count of customers everyday

1. `SELECT DISTINCT`
2. `EXTRACT(DATE FROM order_received_timestamp) Dates,`
3. `count(*) over (partition by EXTRACT(DATE from order_received_timestamp)) CustomerCounts`
4. `from `fractal1a.starschema.fact_daily_orders``
5. `GROUP BY`
6. `order_received_timestamp`
7. `order by Dates`

The screenshot shows the Google Cloud BigQuery console interface. The top navigation bar includes the Google Cloud logo, a project selector set to 'fractal1', and a search bar. The left sidebar contains the 'Explorer' panel with a search bar and a list of workspace resources under 'fractal1a', including 'External connecti...', 'akash', 'layer_1_OLT...', 'order_dataset', 'star_schema', and 'starschema'. The main panel displays a query in a text editor titled 'Untitled 2'. The query is as follows:

```
1 SELECT DISTINCT
2 EXTRACT(DATE FROM order_received_timestamp) Dates,
3 count(*) over (partition by EXTRACT(DATE from order_received_timestamp)) CustomerCounts
4 from `fractal1a.starschema.fact_daily_orders`
5 GROUP BY
6 order_received_timestamp
7 order by Dates
8
```

Below the query editor, the 'Query results' section is visible, showing a table with 7 rows and 2 columns: 'Dates' and 'CustomerCounts'. The table is displayed in the 'RESULTS' tab, with other tabs like 'JOB INFORMATION', 'JSON', 'EXECUTION DETAILS', 'CHART', 'PREVIEW', and 'EXECUTION GRAPH' also available. The status bar at the bottom indicates 'Results per page: 50' and '1 - 50 of 283'.

Row	Dates	CustomerCounts
1	2023-01-01	49
2	2023-01-02	68
3	2023-01-03	56
4	2023-01-04	59
5	2023-01-05	50
6	2023-01-06	55
7	2023-01-07	66

Average time to delivery order. Min and Max time. To be able to slice and dice on hour, weekday, weekend, daily, monthly, geography,

```

1. SELECT DISTINCT
2. EXTRACT(DATE FROM f.order_delivery_timestamp) DATES,
3. EXTRACT(WEEK FROM f.order_delivery_timestamp) WEEKS,
4. EXTRACT(DAYOFWEEK FROM f.order_delivery_timestamp) WEEKDAYS,
5. EXTRACT(MONTH FROM f.order_delivery_timestamp) MONTHS,
6. a.City,
7. MIN(f.order_delivery_time_seconds) MinDeliveryTime,
8. MAX(f.order_delivery_time_seconds) MaxDeliveryTime,
9. AVG(f.order_delivery_time_seconds) AvgDeliveryTime,
10. FROM `fractal1a.starschema.fact_daily_orders` f
11. join `fractal1a.starschema.dim_customer` c on f.customerid = c.customerid
12. join `fractal1a.starschema.dim_address` a on c.address_id = a.address_id
13. group by
14. DATES,
15. WEEKS,
16. WEEKDAYS,
17. MONTHS,
18. City

```

The screenshot shows the Google Cloud BigQuery console interface. On the left is the Explorer pane showing the project 'fractal1a' and its schema 'starschema'. The main area displays a SQL query titled 'Untitled 2' with the following text:

```

1 SELECT DISTINCT
2 EXTRACT(DATE FROM f.order_delivery_timestamp) DATES,
3 EXTRACT(WEEK FROM f.order_delivery_timestamp) WEEKS,
4 EXTRACT(DAYOFWEEK FROM f.order_delivery_timestamp) WEEKDAYS,
5 EXTRACT(MONTH FROM f.order_delivery_timestamp) MONTHS,
6 a.City,
7 MIN(f.order_delivery_time_seconds) MinDeliveryTime,

```

Below the query editor, the 'Query results' section is visible, showing a table with 7 rows of data. The table has columns for DATES, WEEKS, WEEKDAYS, MONTHS, City, MinDeliveryTime, and MaxDeliveryTime. The data is filtered to show only rows for Bangalore.

Row	DATES	WEEKS	WEEKDAYS	MONTHS	City	MinDeliveryTime	MaxDeliveryTime
1	2023-05-15	20	2	5	Bangalore	41775	1:
2	2023-07-11	28	3	7	Bangalore	43536	1:
3	2023-08-25	34	6	8	Bangalore	95836	1:
4	2023-07-12	28	4	7	Bangalore	112846	1
5	2023-05-14	20	1	5	Bangalore	56325	.
6	2023-01-07	1	7	1	Bangalore	71743	1:
7	2023-01-17	3	3	1	Bangalore	105314	1:

At the bottom of the results section, it indicates 'Results per page: 50' and '1 - 50 of 6079'.

Total orders : to be able to slice and dice on hour, weekday, weekend, daily, monthly, geography

```

1. SELECT DISTINCT
2. EXTRACT(DATE FROM f.order_delivery_timestamp) DATES,
3. EXTRACT(WEEK FROM f.order_delivery_timestamp) WEEKS,
4. EXTRACT(DAYOFWEEK FROM f.order_delivery_timestamp) WEEKDAYS,
5. EXTRACT(MONTH FROM f.order_delivery_timestamp) MONTHS,
6. a.City,
7. COUNT(orderid) NumberOfOrders
8. FROM `fractal1a.starschema.fact_daily_orders` f
9. join `fractal1a.starschema.dim_customer` c on f.customerid = c.customerid
10. join `fractal1a.starschema.dim_address` a on c.address_id = a.address_id
11. group by
12. DATES,
13. WEEKS,
14. WEEKDAYS,
15. MONTHS,
16. City

```

The screenshot shows the Google Cloud BigQuery console interface. On the left is the Explorer pane showing the project 'fractal1a' and its schema 'starschema'. The main editor displays a SQL query titled 'Untitled 2'. The query is a SELECT statement with DISTINCT, extracting DATE, WEEK, DAYOFWEEK, and MONTH from the 'order_delivery_timestamp' column, along with 'City' and 'COUNT(orderid) NumberOfOrders'. The query is joined with 'dim_customer' and 'dim_address' tables. The query has been executed successfully, as indicated by the 'Query completed' status. Below the query editor, the 'Query results' section is visible, showing a table with columns: DATES, WEEKS, WEEKDAYS, MONTHS, City, and NumberOfOrders. The table contains 7 rows of data, all from Bangalore.

Row	DATES	WEEKS	WEEKDAYS	MONTHS	City	NumberOfOrders
1	2023-05-15	20	2	5	Bangalore	4
2	2023-07-11	28	3	7	Bangalore	3
3	2023-08-25	34	6	8	Bangalore	3
4	2023-07-12	28	4	7	Bangalore	1
5	2023-05-14	20	1	5	Bangalore	1
6	2023-01-07	1	7	1	Bangalore	6
7	2023-01-17	3	3	1	Bangalore	1

Results per page: 50 | 1 - 50 of 6079