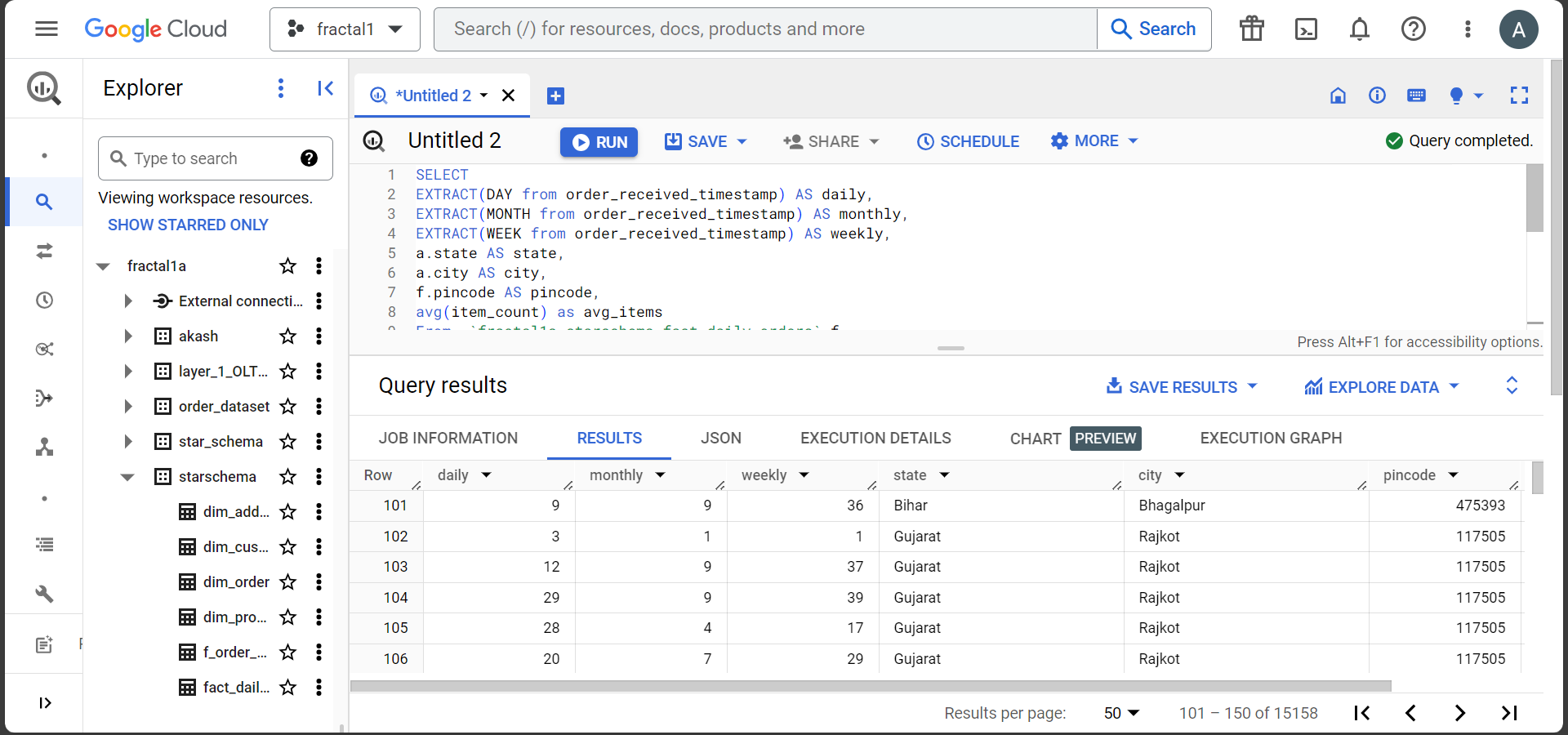
**BigQuery SQL Analysis**

**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



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

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated