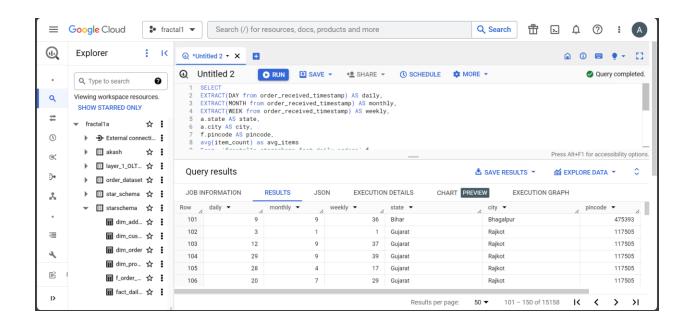
BigQuery SQL Analysis

By: Arpit Kushwaha

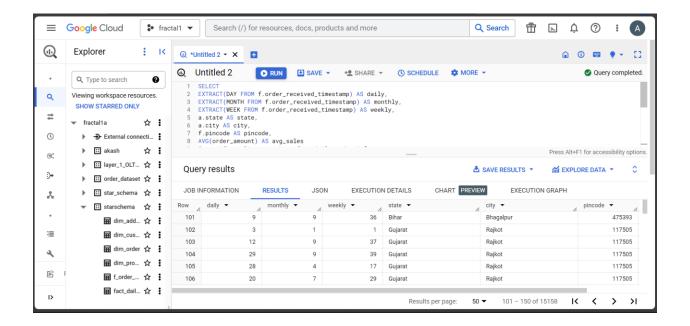
Average number of items per order - daily, monthly, weekly, state, city, pincode:

```
1. SELECT
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
```



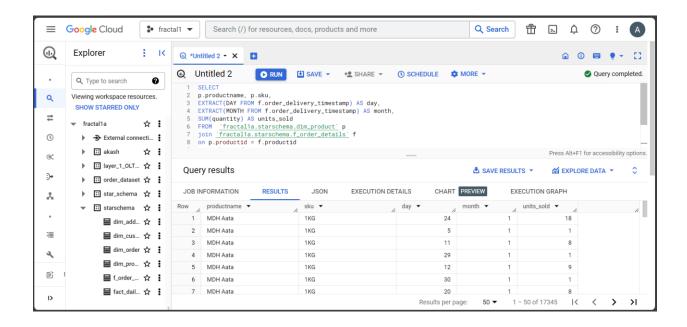
Average amount of sales per order - daily, monthly, weekly, state, city, pincode

```
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
```



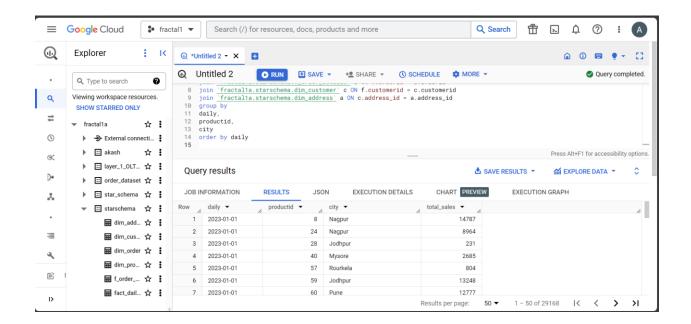
Total number of units sold per day of a product SKU and its monthly trend

```
    SELECT
    p.productname, p.sku,
    EXTRACT(DAY FROM f.order_delivery_timestamp) AS day,
    EXTRACT(MONTH FROM f.order_delivery_timestamp) AS month,
    SUM(quantity) AS units_sold
    FROM `fractal1a.starschema.dim_product` p
    join `fractal1a.starschema.f_order_details` f
    on p.productid = f.productid
    group by productname,
    sku,
    day,
    month
```



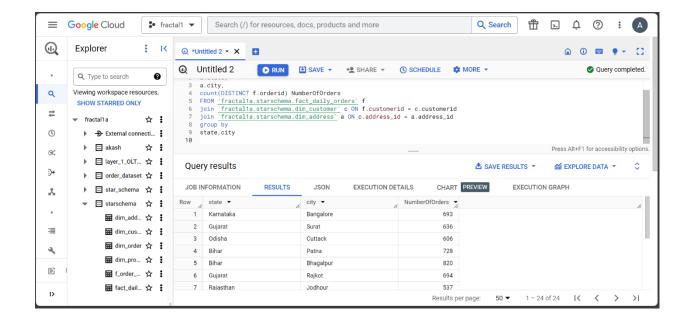
Total Order Amount on daily basis, also to be able to split by product and geography

```
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
```



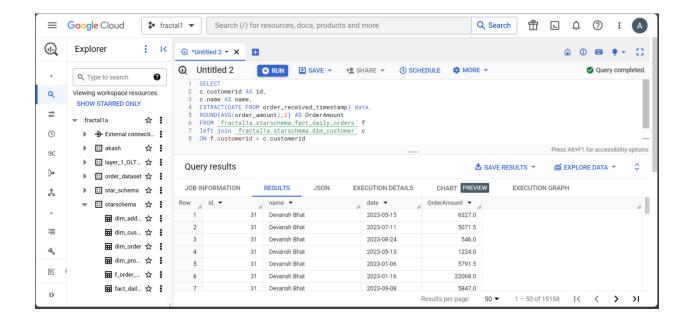
Distribution of orders according to area (state, city, pincode etc)

```
    SELECT
    a.state,
    a.city,
    count(DISTINCT f.orderid) NumberOfOrders
    FROM `fractal1a.starschema.fact_daily_orders` f
    join `fractal1a.starschema.dim_customer` c ON f.customerid = c.customerid
    join `fractal1a.starschema.dim_address` a ON c.address_id = a.address_id
    group by
    state,city
```



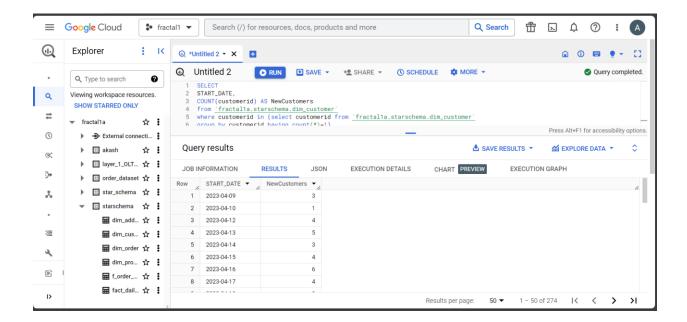
Average order amount per customer on daily basis

```
    SELECT
    c.customerid AS id,
    c.name AS name,
    EXTRACT(DATE FROM order_received_timestamp) date,
    ROUND(AVG(order_amount),2) AS OrderAmount
    FROM `fractal1a.starschema.fact_daily_orders` f
    left join `fractal1a.starschema.dim_customer` c
    ON f.customerid = c.customerid
    group by
    id, date, name
```



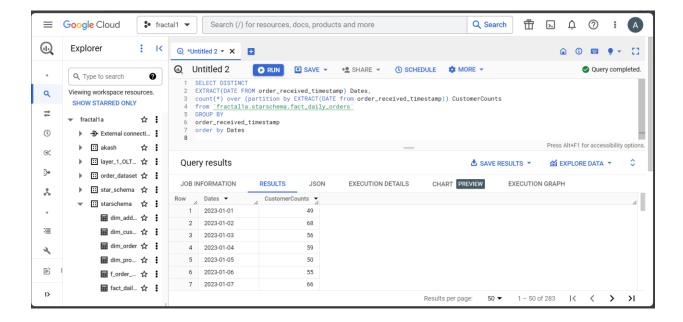
New Customers on daily basis

- 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



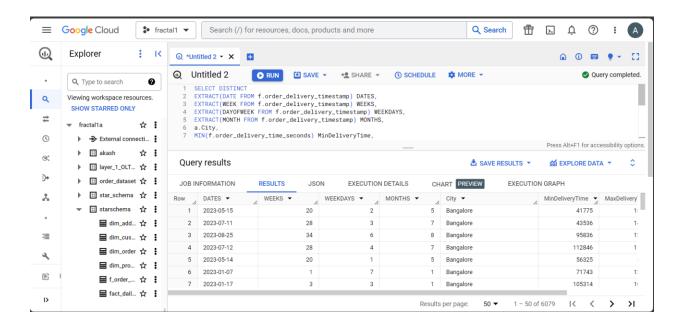
Total count of customers everyday

- 1. SELECT DISTINCT
- EXTRACT(DATE FROM order_received_timestamp) Dates,
- 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



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
EXTRACT(DATE FROM f.order_delivery_timestamp) DATES,
EXTRACT(WEEK FROM f.order_delivery_timestamp) WEEKS,
4. EXTRACT(DAYOFWEEK FROM f.order_delivery_timestamp) WEEKDAYS,
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,
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
```



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

- 1. SELECT DISTINCT EXTRACT(DATE FROM f.order_delivery_timestamp) DATES, EXTRACT(WEEK FROM f.order_delivery_timestamp) WEEKS, EXTRACT(DAYOFWEEK FROM f.order_delivery_timestamp) WEEKDAYS, 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

