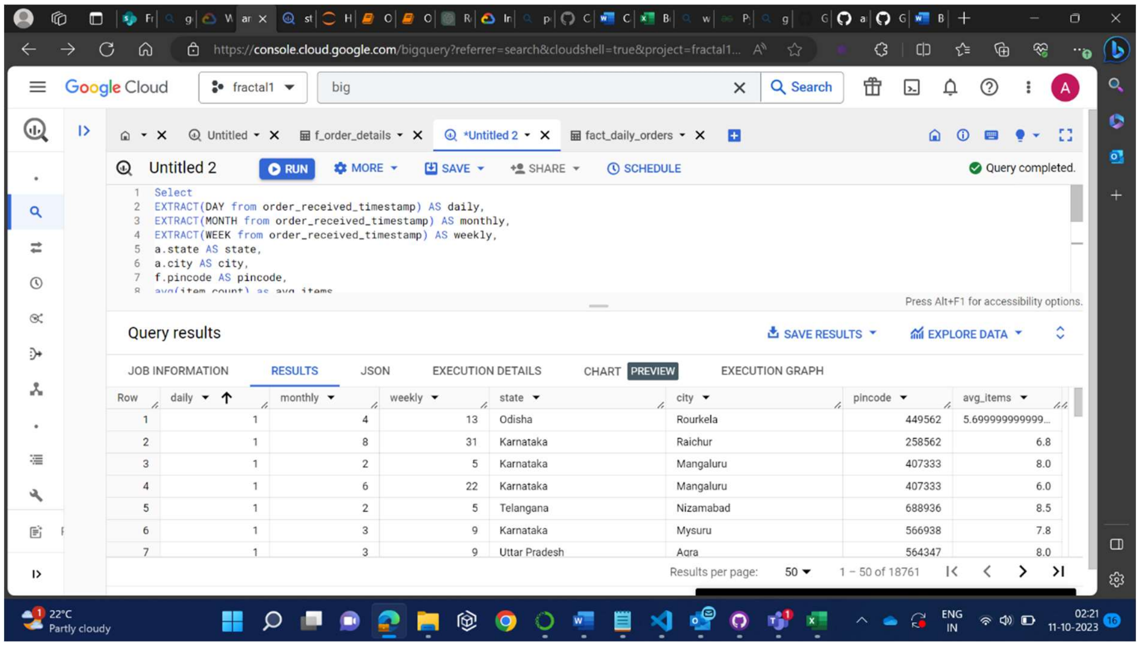
Capstone Project Big Query Analysis

Q.1. Average number of items per order - daily, monthly, weekly, state, city, pincode.

```
Select
EXTRACT(DAY from order_received_timestamp) AS daily,
EXTRACT(MONTH from order_received_timestamp) AS monthly,
EXTRACT(WEEK from order_received_timestamp) AS weekly,
a.state AS state,
a.city AS city,
f.pincode AS pincode,
avg(item_count) as avg_items

From  `fractal1a.star_schema.fact_daily_orders` f
left join `fractal1a.star_schema.dim_customer` c
  on f.customerid = c.customerid
left join `fractal1a.star_schema.dim_address` a
  on c.address_id = a.address_id
group by
daily,
monthly,
weekly,
state,
city,
pincode
```



Untitled 2 — Query completed.

```
Select
EXTRACT(DAY from order_received_timestamp) AS daily,
EXTRACT(MONTH from order_received_timestamp) AS monthly,
EXTRACT(WEEK from order_received_timestamp) AS weekly,
a.state AS state,
a.city AS city,
f.pincode AS pincode,
avg(item_count) as avg_items
```

Query results

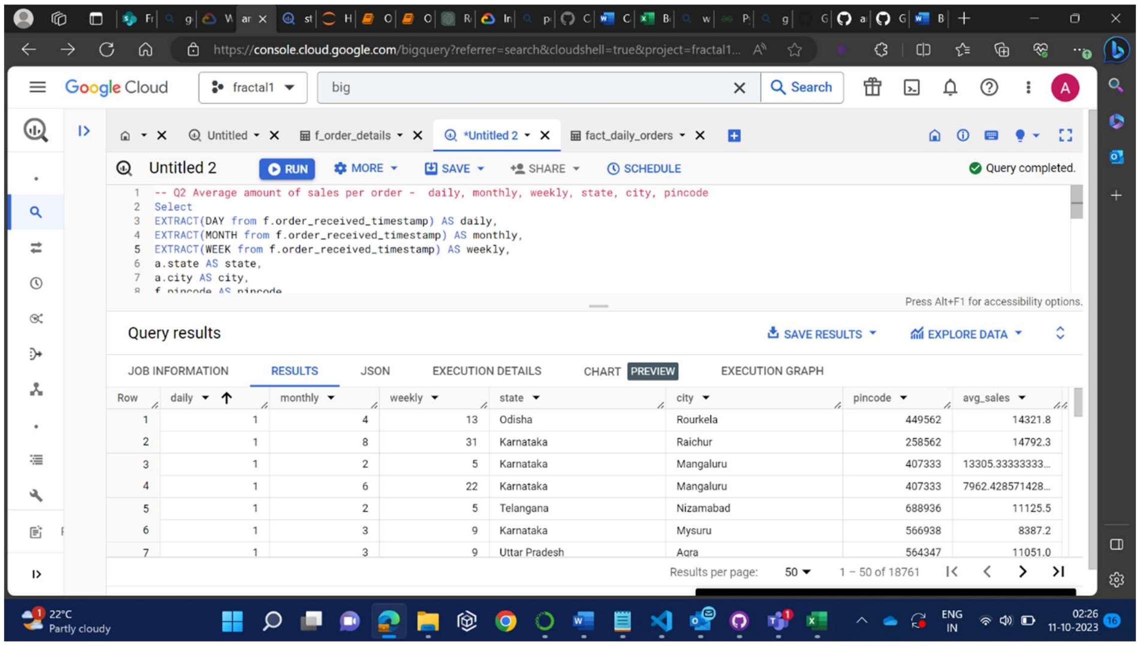
Row	daily	monthly	weekly	state	city	pincode	avg_items
1	1	4	13	Odisha	Rourkela	449562	5.699999999999...
2	1	8	31	Karnataka	Raichur	258562	6.8
3	1	2	5	Karnataka	Mangaluru	407333	8.0
4	1	6	22	Karnataka	Mangaluru	407333	6.0
5	1	2	5	Telangana	Nizamabad	688936	8.5
6	1	3	9	Karnataka	Mysuru	566938	7.8
7	1	3	9	Uttar Pradesh	Agra	564347	8.0

Results per page: 50 — 1 – 50 of 18761

Q.2. Average amount of sales per order - daily, monthly, weekly, state, city, pincode.

```
Select
EXTRACT(DAY from f.order_received_timestamp) AS daily,
```

```sql
EXTRACT(MONTH from f.order_received_timestamp) AS monthly,
EXTRACT(WEEK from f.order_received_timestamp) AS weekly,
a.state AS state,
a.city AS city,
f.pincode AS pincode,
avg(order_amount) as avg_sales


From  `fractal1a.star_schema.fact_daily_orders` f
left join `fractal1a.star_schema.dim_customer` c
  on f.customerid = c.customerid
left join `fractal1a.star_schema.dim_address` a
  on c.address_id = a.address_id
group by
daily,
monthly,
weekly,
sTATE,
city,
pincode
```



Untitled 2

```sql
-- Q2 Average amount of sales per order -  daily, monthly, weekly, state, city, pincode
Select
EXTRACT(DAY from f.order_received_timestamp) AS daily,
EXTRACT(MONTH from f.order_received_timestamp) AS monthly,
EXTRACT(WEEK from f.order_received_timestamp) AS weekly,
a.state AS state,
a.city AS city,
f.pincode AS pincode
```

Query completed.

Query results

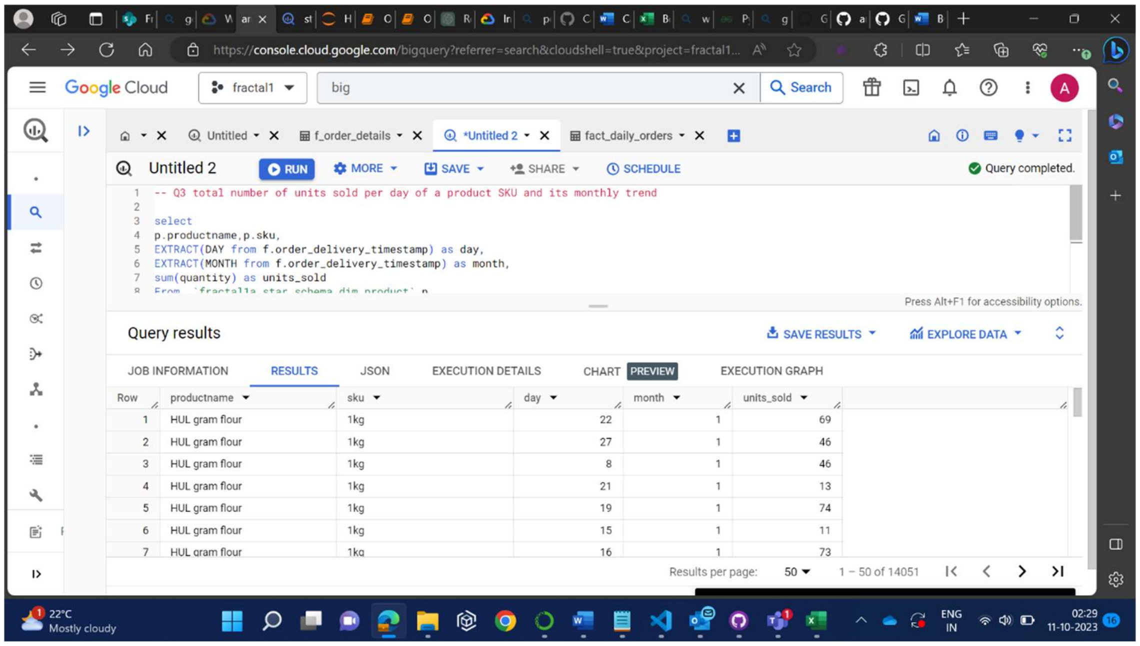
Row	daily	monthly	weekly	state	city	pincode	avg_sales
1	1	4	13	Odisha	Rourkela	449562	14321.8
2	1	8	31	Karnataka	Raichur	258562	14792.3
3	1	2	5	Karnataka	Mangaluru	407333	13305.33333333...
4	1	6	22	Karnataka	Mangaluru	407333	7962.428571428...
5	1	2	5	Telangana	Nizamabad	688936	11125.5
6	1	3	9	Karnataka	Mysuru	566938	8387.2
7	1	3	9	Uttar Pradesh	Agra	564347	11051.0

Results per page: 50   1 – 50 of 18761

Q.3. Total number of units sold per day of a product SKU and its monthly trend.

```sql
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  `fractalb.star_schema.dim_product` p
```

```
join `fractalb.star_schema.f_order_details` f

on p.productid = f.productid

group by productname,

sku,

day,

month
```



```
-- Q3 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.star_schema.dim_product` p
```

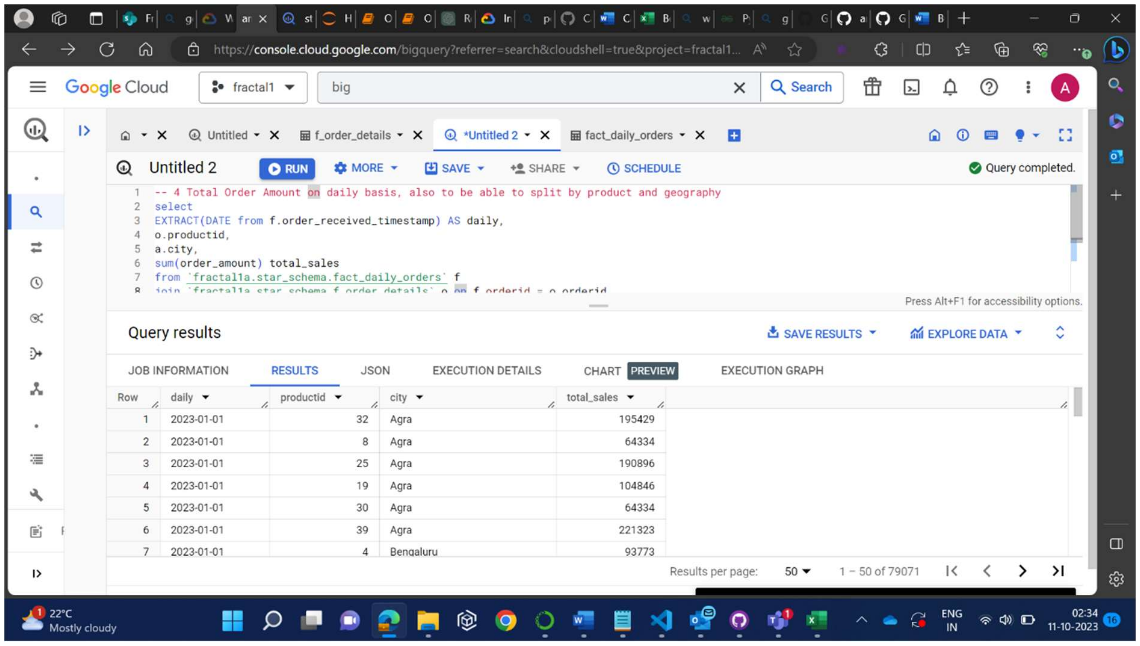
Query results

Row	productname	sku	day	month	units_sold
1	HUL gram flour	1kg	22	1	69
2	HUL gram flour	1kg	27	1	46
3	HUL gram flour	1kg	8	1	46
4	HUL gram flour	1kg	21	1	13
5	HUL gram flour	1kg	19	1	74
6	HUL gram flour	1kg	15	1	11
7	HUL gram flour	1kg	16	1	73

Results per page: 50 1 – 50 of 14051

Q.4. Total Order Amount on daily basis, also to be able to split by product and geography.

```
select
EXTRACT(DATE from f.order_received_timestamp) AS daily,
o.productid,
a.city,
sum(order_amount) total_sales
from `fractal1a.star_schema.fact_daily_orders` f
join `fractal1a.star_schema.f_order_details` o on f.orderid = o.orderid
join `fractal1a.star_schema.dim_customer` c on f.customerid = c.customerid
join `fractal1a.star_schema.dim_address` a on c.address_id = a.address_id
group by
daily, productid, city
order by daily
```



```
-- 4 Total Order Amount on daily basis, also to be able to split by product and geography
select
EXTRACT(DATE from f.order_received_timestamp) AS daily,
o.productid,
a.city,
sum(order_amount) total_sales
from `fractal1a.star_schema.fact_daily_orders` f
join `fractal1a.star_schema.f_order_details` o on f.orderid = o.orderid
```

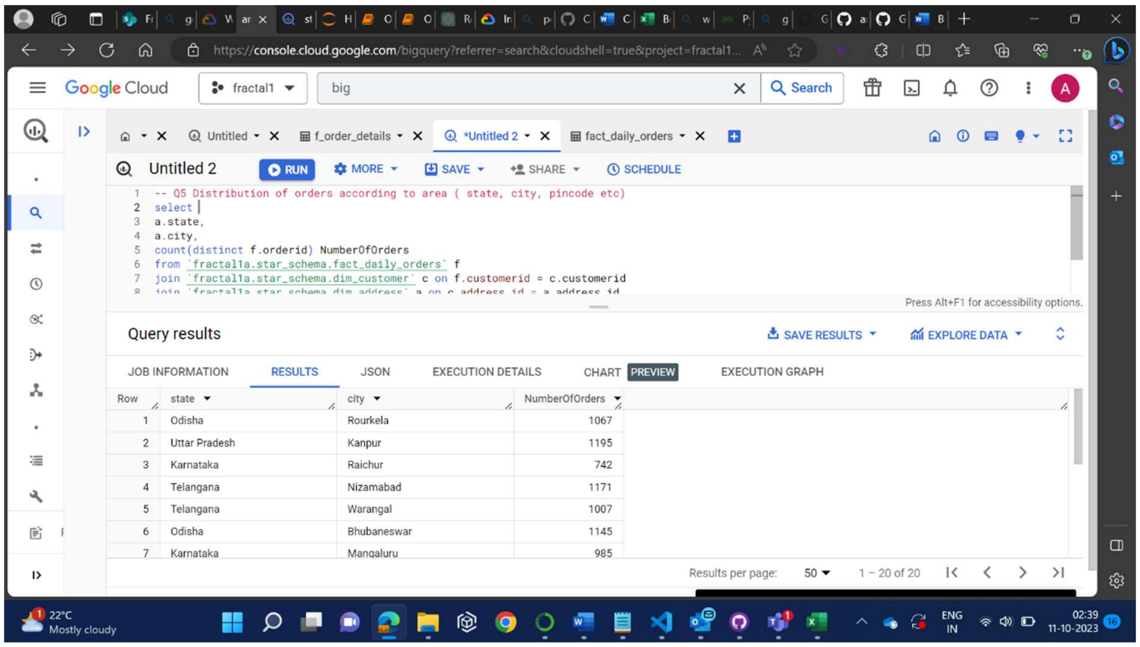
Query results

Row	daily	productid	city	total_sales
1	2023-01-01	32	Agra	195429
2	2023-01-01	8	Agra	64334
3	2023-01-01	25	Agra	190896
4	2023-01-01	19	Agra	104846
5	2023-01-01	30	Agra	64334
6	2023-01-01	39	Agra	221323
7	2023-01-01	4	Bengaluru	93773

Results per page: 50 1 – 50 of 79071

Q.5. Distribution of orders according to area (state, city, pincode etc)

```sql
select
a.state,
a.city,
count(distinct f.orderid) NumberOfOrders
from `fractal1a.star_schema.fact_daily_orders` f
join `fractal1a.star_schema.dim_customer` c on f.customerid = c.customerid
join `fractal1a.star_schema.dim_address` a on c.address_id = a.address_id
group by
state,city
```



```
-- Q5 Distribution of orders according to area ( state, city, pincode etc)
select
a.state,
a.city,
count(distinct f.orderid) NumberOfOrders
from `fractal1a.star_schema.fact_daily_orders` f
join `fractal1a.star_schema.dim_customer` c on f.customerid = c.customerid
join `fractal1a.star_schema.dim_address` a on c.address_id = a.address_id
```

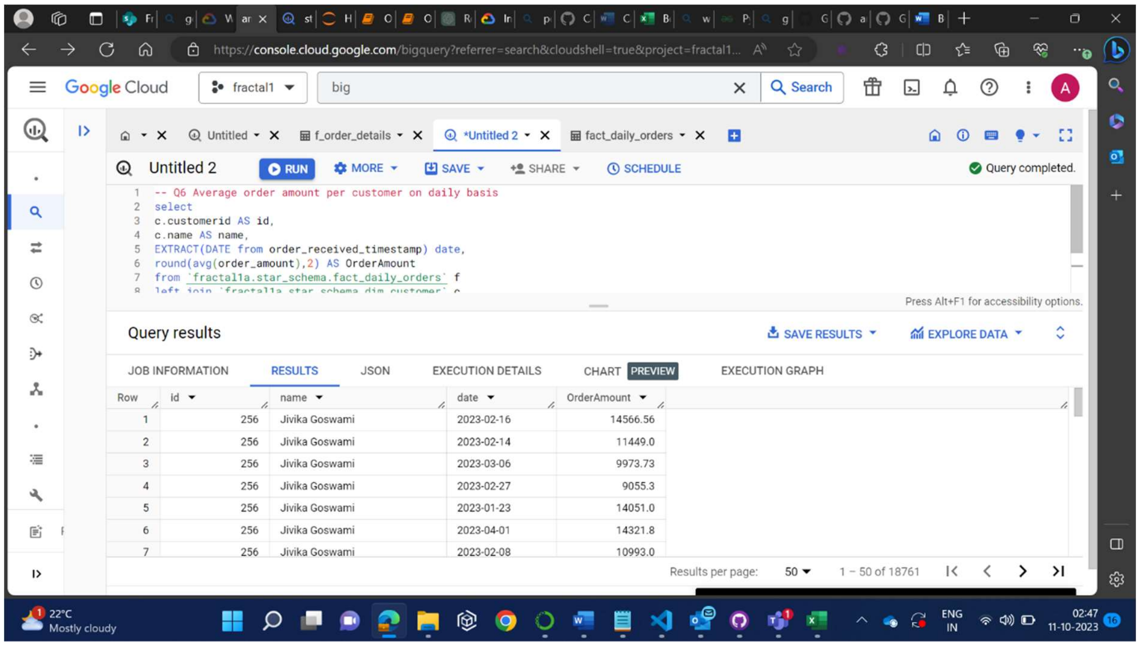
Query results

Row	state	city	NumberOfOrders
1	Odisha	Rourkela	1067
2	Uttar Pradesh	Kanpur	1195
3	Karnataka	Raichur	742
4	Telangana	Nizamabad	1171
5	Telangana	Warangal	1007
6	Odisha	Bhubaneswar	1145
7	Karnataka	Mangaluru	985

Results per page: 50 1 – 20 of 20

Q.6. 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.star_schema.fact_daily_orders` f
left join `fractal1a.star_schema.dim_customer` c
on f.customerid = c.customerid
group by
id, date, name
```



```
-- Q6 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.star_schema.fact_daily_orders` f
left join `fractal1a.star_schema.dim_customer` c
```

Query completed.

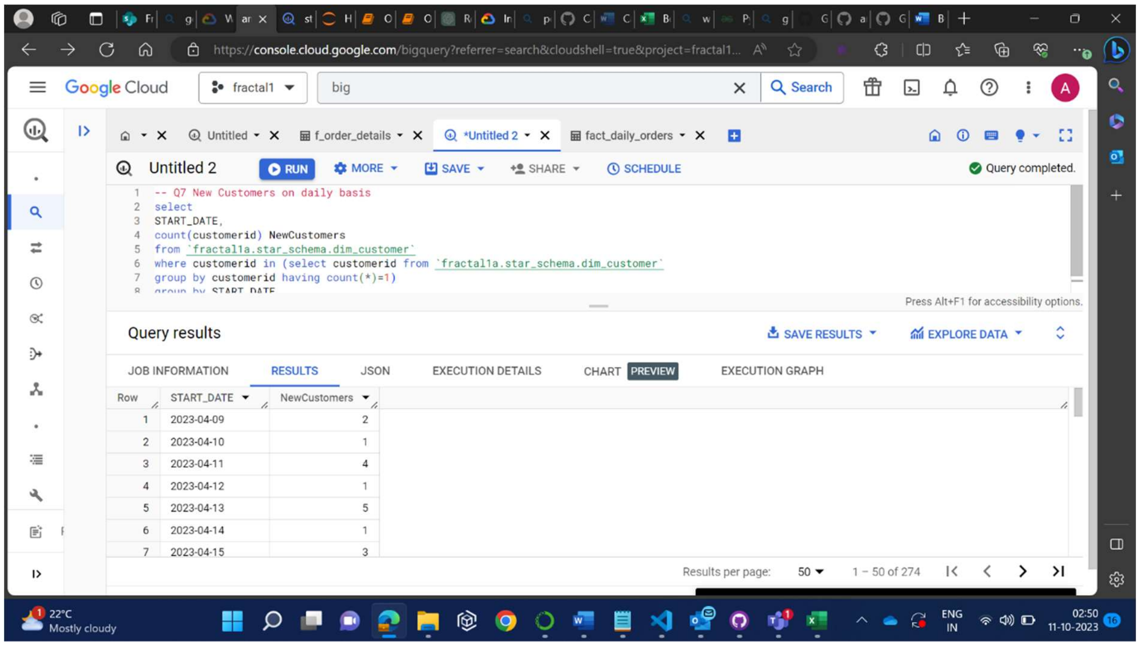
Query results

Row	id	name	date	OrderAmount
1	256	Jivika Goswami	2023-02-16	14566.56
2	256	Jivika Goswami	2023-02-14	11449.0
3	256	Jivika Goswami	2023-03-06	9973.73
4	256	Jivika Goswami	2023-02-27	9055.3
5	256	Jivika Goswami	2023-01-23	14051.0
6	256	Jivika Goswami	2023-04-01	14321.8
7	256	Jivika Goswami	2023-02-08	10993.0

Results per page: 50 1 – 50 of 18761

Q.7. New Customers on daily basis.

```
select
START_DATE,
count(customerid) NewCustomers
from `fractal1a.star_schema.dim_customer`
where customerid in (select customerid from `fractal1a.star_schema.dim_customer`
group by customerid having count(*)=1)
group by START_DATE
```



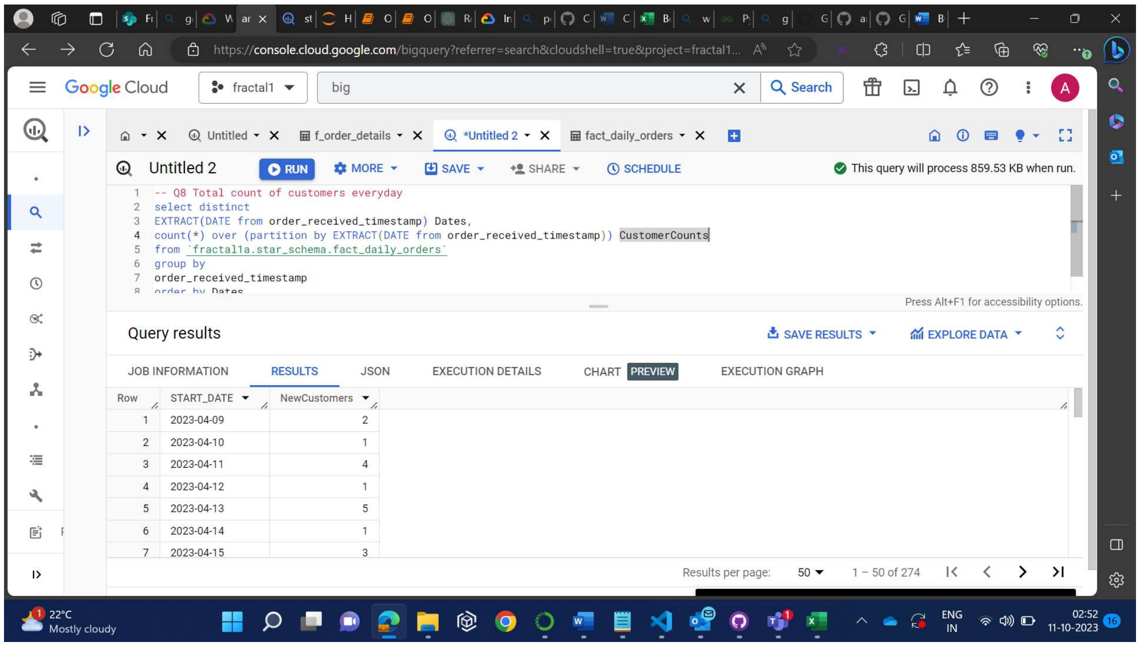
```sql
-- Q7 New Customers on daily basis
select
START_DATE,
count(customerid) NewCustomers
from `fractal1a.star_schema.dim_customer`
where customerid in (select customerid from `fractal1a.star_schema.dim_customer`
group by customerid having count(*)=1)
group by START_DATE
```

Query results

Row	START_DATE	NewCustomers
1	2023-04-09	2
2	2023-04-10	1
3	2023-04-11	4
4	2023-04-12	1
5	2023-04-13	5
6	2023-04-14	1
7	2023-04-15	3

Q.8. Total count of customers everyday.

```sql
select distinct
EXTRACT(DATE from order_received_timestamp) Dates,
count(*) over (partition by EXTRACT(DATE from order_received_timestamp))
CustomerCounts
from `fractal1a.star_schema.fact_daily_orders`
group by
order_received_timestamp
order by Dates
```



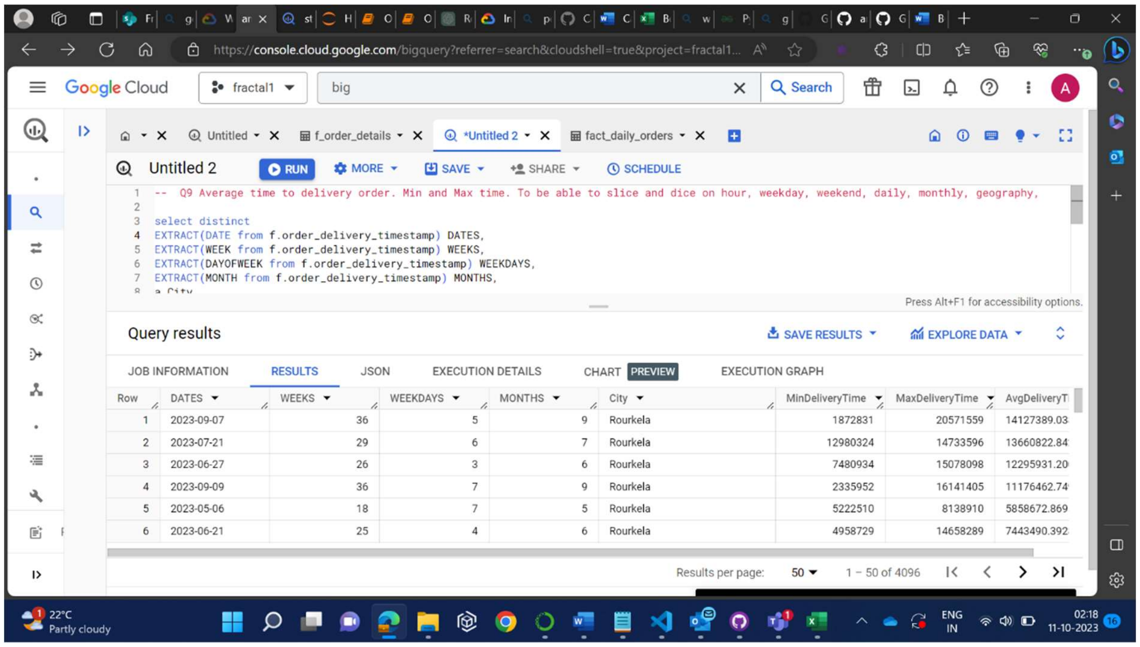
```sql
-- Q8 Total count of customers everyday
select distinct
EXTRACT(DATE from order_received_timestamp) Dates,
count(*) over (partition by EXTRACT(DATE from order_received_timestamp)) CustomerCounts
from `fractal1a.star_schema.fact_daily_orders`
group by
order_received_timestamp
order by Dates
```

Query results

Row	START_DATE	NewCustomers
1	2023-04-09	2
2	2023-04-10	1
3	2023-04-11	4
4	2023-04-12	1
5	2023-04-13	5
6	2023-04-14	1
7	2023-04-15	3

Q.9 Average time to delivery order. Min and Max time. To be able to slice and dice on hour, weekday, weekend, daily, monthly, geography.

```sql
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,
a.City,
Min(f.order_delivery_time_seconds) MinDeliveryTime,
max(f.order_delivery_time_seconds) MaxDeliveryTime,
avg(f.order_delivery_time_seconds) AvgDeliveryTime,
from `fractal1a.star_schema.fact_daily_orders` f
join `fractal1a.star_schema.dim_customer` c on f.customerid = c.customerid
join `fractal1a.star_schema.dim_address` a on c.address_id = a.address_id
group by
DATES,
WEEKS,
WEEKDAYS,
MONTHS,
City
```



```sql
--  Q9 Average time to delivery order. Min and Max time. To be able to slice and dice on hour, weekday, weekend, daily, monthly, geography,

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,
```

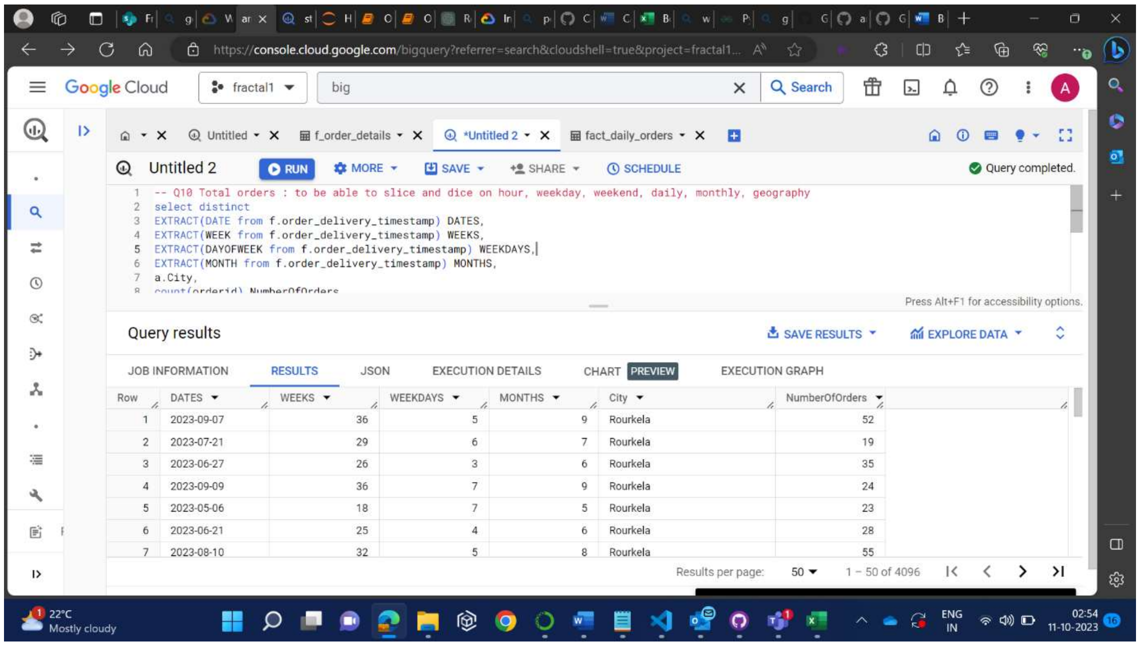
Query results

Row	DATES	WEEKS	WEEKDAYS	MONTHS	City	MinDeliveryTime	MaxDeliveryTime	AvgDeliveryT
1	2023-09-07	36	5	9	Rourkela	1872831	20571559	14127389.03
2	2023-07-21	29	6	7	Rourkela	12980324	14733596	13660822.84
3	2023-06-27	26	3	6	Rourkela	7480934	15078098	12295931.20
4	2023-09-09	36	7	9	Rourkela	2335952	16141405	11176462.74
5	2023-05-06	18	7	5	Rourkela	5222510	8138910	5858672.869
6	2023-06-21	25	4	6	Rourkela	4958729	14658289	7443490.392

Results per page: 50 1 – 50 of 4096

Q.10. Total orders : to be able to slice and dice on hour, weekday, weekend, daily, monthly, geography.

```sql
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,
a.City,
count(orderid) NumberOfOrders
from `fractalb.star_schema.fact_daily_orders` f
join `fractalb.star_schema.dim_customer` c on f.customerid = c.customerid
join `fractalb.star_schema.dim_address` a on c.address_id = a.address_id
group by
DATES,WEEKS,WEEKDAYS,MONTHS,City
```



```
-- Q10 Total orders : to be able to slice and dice on hour, weekday, weekend, daily, monthly, geography
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,
a.City,
count(orderid) NumberOfOrders
```

Query completed.

Query results

Row	DATES	WEEKS	WEEKDAYS	MONTHS	City	NumberOfOrders
1	2023-09-07	36	5	9	Rourkela	52
2	2023-07-21	29	6	7	Rourkela	19
3	2023-06-27	26	3	6	Rourkela	35
4	2023-09-09	36	7	9	Rourkela	24
5	2023-05-06	18	7	5	Rourkela	23
6	2023-06-21	25	4	6	Rourkela	28
7	2023-08-10	32	5	8	Rourkela	55

Results per page: 50   1 – 50 of 4096