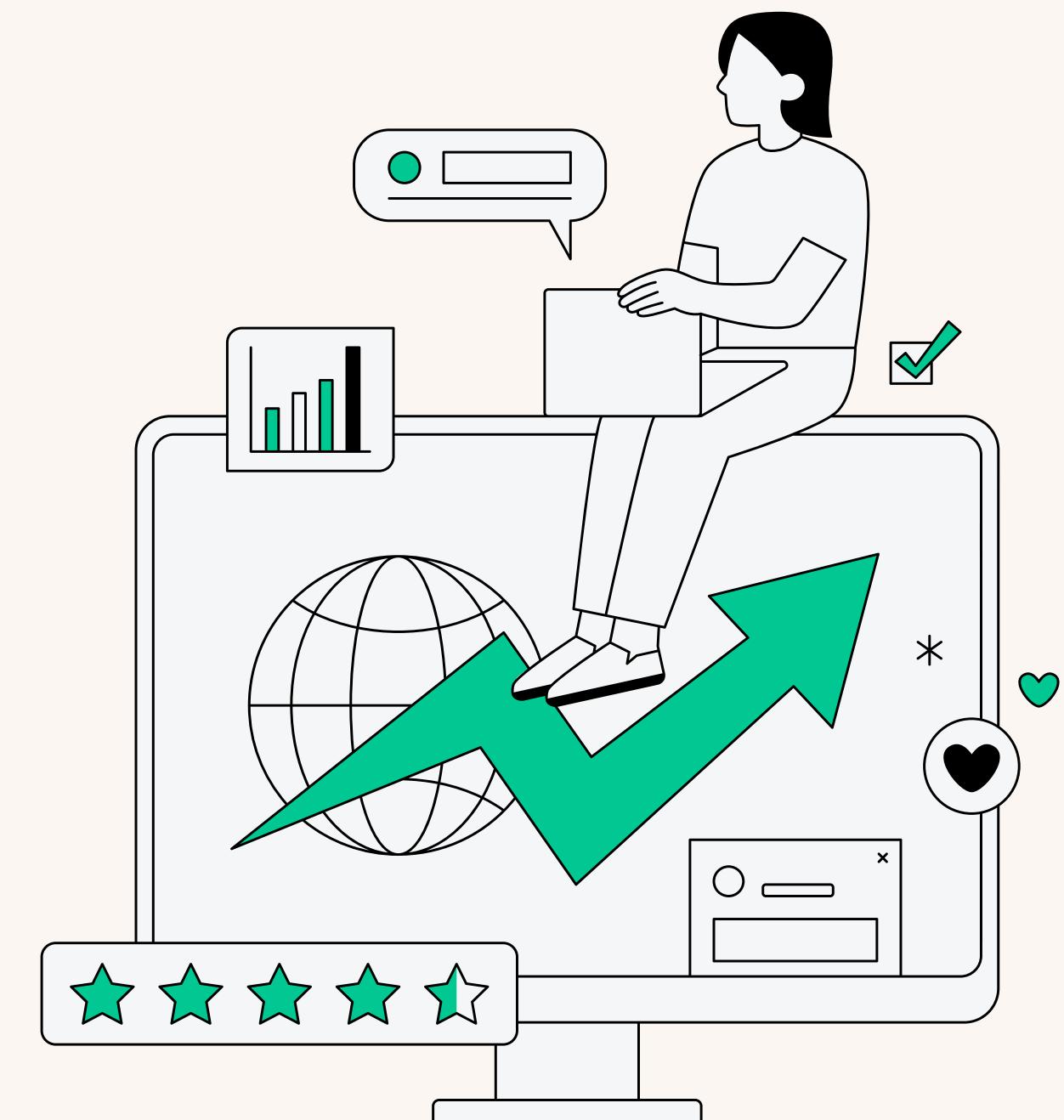


Presented by : Shivang Dubey

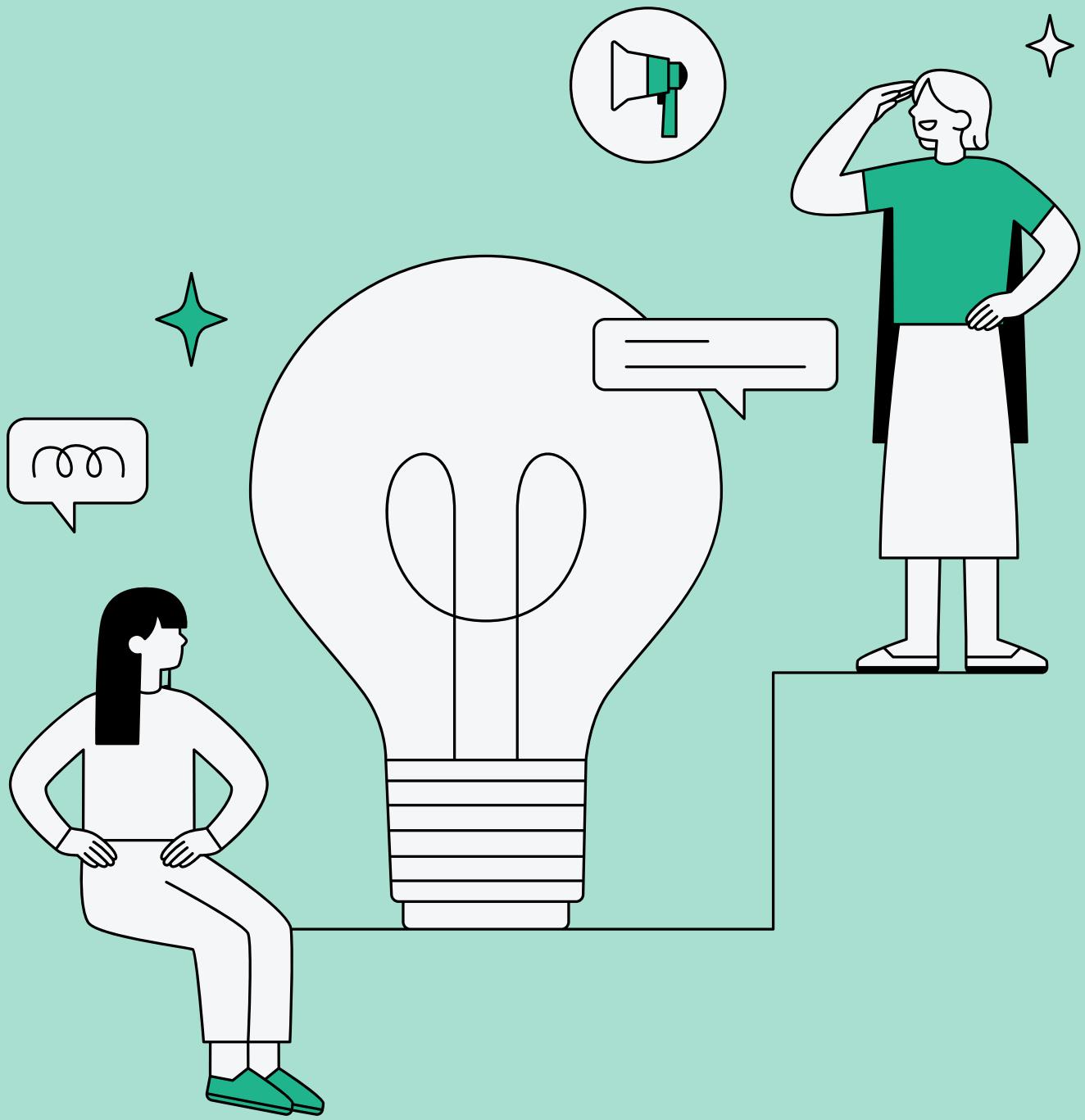
Phone Mart insights

AN ADVANCE SQL PROJECT



Introduction to Phone Mart insights

PhoneMart Insights is a comprehensive data analysis project focused on the sales of mobile phones across multiple stores around the world. The project aims to analyze sales data collected from a global network of PhoneMart stores to provide actionable insights into performance trends, customer preferences, inventory management, and regional sales patterns. By leveraging SQL for querying and analyzing the data, PhoneMart Insights will enable data-driven decision-making to optimize sales strategies and improve customer satisfaction.



OBJECTIVE

01.

Sales Performance Analysis: To identify high-performing stores, track sales over time, and highlight seasonal or regional variations in demand for different phone models.

02.

Customer Insights: To understand customer buying behavior, popular brands, and preferences for specific features or price points.

03.

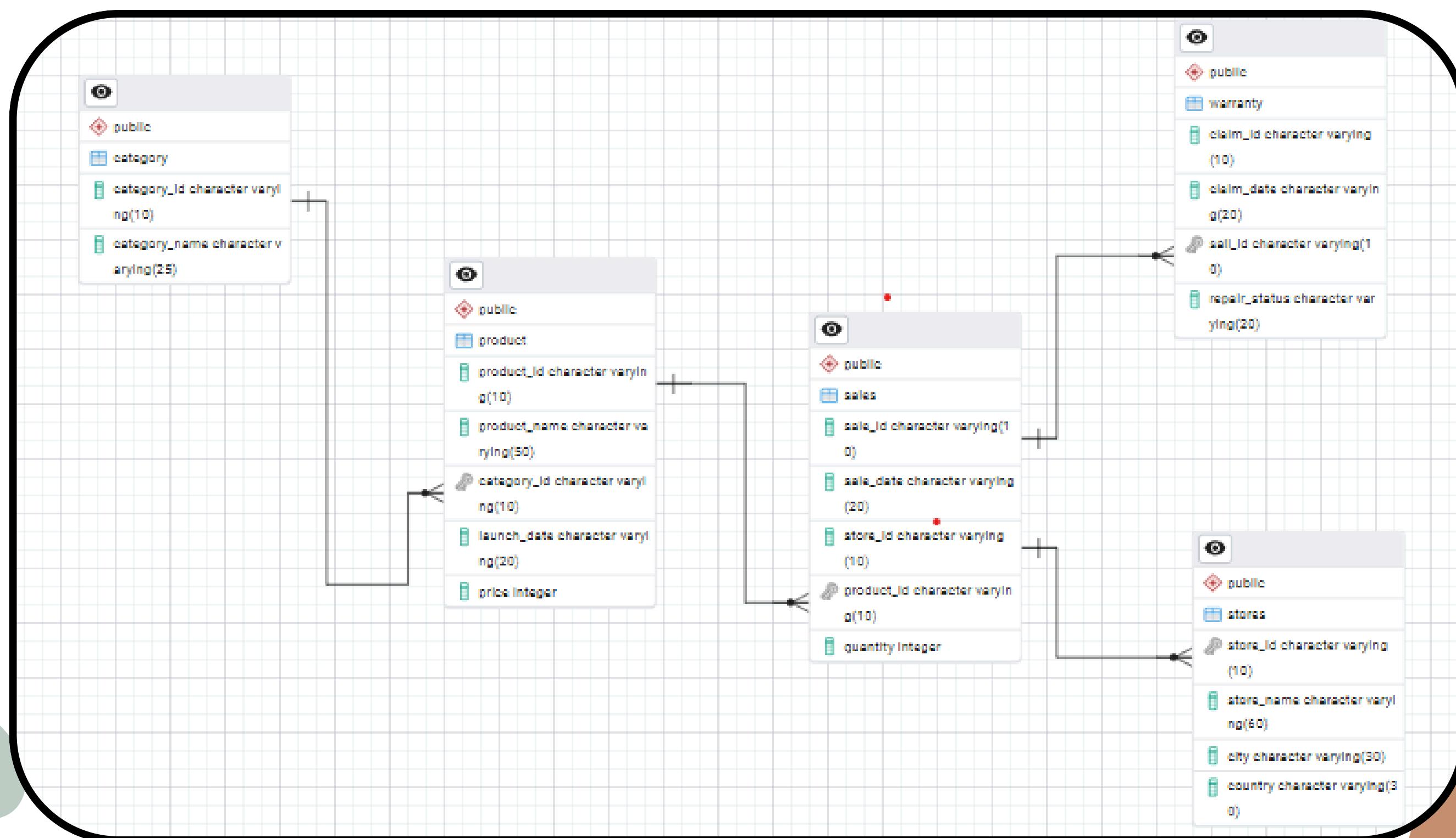
Revenue and Profitability Analysis: To calculate revenue generated by each store and analyze profit margins to identify opportunities for cost savings or promotional efforts.

04.

Revenue and Profitability Analysis: To calculate revenue generated by each store and analyze profit margins to identify opportunities for cost savings or promotional efforts.



ENTITY RELATIONSHIP MODEL FOR PHONE MART INSIGHTS



IDENTIFY THE LEAST SELLING PRODUCT IN EACH COUNTRY FOR EACH YEAR BASED ON TOTAL UNITS SOLD.

```
select
    country ,
    product_name
from
(
    select
        st.country,
        p.product_name ,
        sum(sa.quantity) as total_sales ,
        rank() over(partition by st.country order by sum(sa.quantity)) as ranking
    from sales sa
    join stores st
    on sa.store_id = st.store_id
    join product p
    on sa.product_id = p.product_id
    group by 1,2
)s
where ranking = 1
```

	country character varying (30)	product_name character varying (50)
1	Argentina	MacBook Air M2
2	Argentina	iPhone 12 Pro
3	Australia	iMac 24-inch M1
4	Austria	HomePod mini 2nd Gen
5	Belgium	iPhone 8
6	Brazil	Apple Watch Series 6
7	Canada	Apple TV 4K 2nd Gen
8	China	Apple TV 4K Ultra
9	Denmark	iPad Air 5th Gen
10	France	Magic Keyboard
11	Germany	iPhone 11 Pro
12	Greece	iPhone 13 Pro
13	India	Mac Mini with M1 Chip
14	Indonesia	iPhone 13 Pro

FOR EACH STORE, IDENTIFY THE BEST-SELLING DAY BASED ON HIGHEST QUANTITY SOLD.

```
select * from sales
select sale_date , store_id ,total_sales
from
(
select sale_date ,
store_id ,
sum(quantity) as total_sales ,
row_number() over(partition by store_id order by sum(quantity) desc ) as rnk
from sales
group by 1,2
order by store_id
) s
where rnk = 1|
```

	country character varying (30) 	product_name character varying (50) 
1	Argentina	MacBook Air M2
2	Argentina	iPhone 12 Pro
3	Australia	iMac 24-inch M1
4	Austria	HomePod mini 2nd Gen
5	Belgium	iPhone 8
6	Brazil	Apple Watch Series 6
7	Canada	Apple TV 4K 2nd Gen
8	China	Apple TV 4K Ultra
9	Denmark	iPad Air 5th Gen
10	France	Magic Keyboard
11	Germany	iPhone 11 Pro
12	Greece	iPhone 13 Pro
13	India	Mac Mini with M1 Chip
14	Indonesia	iPhone 13 Pro
15	Italy	MacBook Air M2
16	Japan	Apple TV+ Subscription
17	Japan	Apple TV 4K 2nd Gen

CALCULATE THE CORRELATION BETWEEN PRODUCT PRICE AND WARRANTY CLAIMS FOR PRODUCTS SOLD IN THE LAST FIVE YEARS, SEGMENTED BY PRICE RANGE.

```
SELECT
CASE
    WHEN p.price < 300 THEN 'Less Expenses Product'
    WHEN p.price BETWEEN 300 AND 700 THEN 'Mid Range Product'
    ELSE 'Expensive Product'
END as price_segment ,
COUNT (w.claim_id) as total_Claim
FROM warranty as w
LEFT JOIN
sales as s
ON w.sale_id = s.sale_id
JOIN
product as p
ON p.product_id = s.product_id
WHERE to_date(w.claim_date , 'mm dd ,yyyy') >= CURRENT_DATE - INTERVAL ' 5 year '
GROUP BY 1
```

	price_segment text	total_claim bigint
1	Less Expenses Product	35
2	Mid Range Product	34
3	Expensive Product	45

CALCULATE TOTAL NUMBERS OF UNITS SOLD BY EACH STORE

```
select sa.store_id ,  
st.store_name ,  
sum(sa.quantity)  
from sales sa  
join stores st  
on sa.store_id = st.store_id  
group by 1,2  
order by 3 desc
```

	store_id character varying (10) 	store_name character varying (60) 	sum bigint 
1	ST-12	Apple Queen's Park	175
2	ST-50	Apple Mission Bay	168
3	ST-24	Apple Pacifico Mall	132
4	ST-56	Apple Namdaemun Market	127
5	ST-11	Apple Seville Park	118
6	ST-29	Apple Loop Street	116
7	ST-48	Apple Place d'Italie	114
8	ST-41	Apple Innbruck Square	110
9	ST-9	Apple Darling Harbour	101
10	ST-53	Apple Bahnhofstrasse	96
11	ST-21	Apple Sandton City	94
12	ST-33	Apple Colaba District	93
13	ST-63	Apple MG Road	90
14	ST-13	Apple Italian Square	90
15	ST-27	Apple Tverskaya Street	89
16	ST-61	Apple Yas Island	88
17	ST-18	Apple Kings Road	88

HOW MANY STORES HAVE NEVER FILED A WARRANTY CLAIM

```
select *  
from stores  
where store_id NOT IN  
(  
    select distinct s.store_id  
    from sales as s  
    right join warranty as w  
    on s.sale_id = w.sale_id  
) ;
```

	store_id character varying (10) 	store_name character varying (60) 	city character varying (30) 	country character varying (30) 
1	ST-1	Apple Gateway of New York	New York	USA
2	ST-2	Apple Ridgeview Plaza	Los Angeles	USA
3	ST-6	Apple Tower Bridge	London	UK
4	ST-7	Apple Champs-Elysées	Paris	France
5	ST-10	Apple Alexanderplatz	Berlin	Germany
6	ST-16	Apple Marina Mall	Dubai	UAE
7	ST-19	Apple Gangnam Square	Seoul	South Korea
8	ST-20	Apple Harbour City	Toronto	Canada
9	ST-22	Apple Paulista Avenue	Sao Paulo	Brazil
10	ST-30	Apple Beachfront	Rio de Janeiro	Brazil
11	ST-34	Apple Pavilion Gardens	Kuala Lumpur	Malaysia
12	ST-38	Apple Rue de la Paix	Brussels	Belgium
13	ST-46	Apple Rådhusplassen	Copenhagen	Denmark
14	ST-49	Apple Boulevard Haussmann	Paris	France
15	ST-54	Apple Happy Valley Road	Hong Kong	China
16	ST-55	Apple High Street	Melbourne	Australia
17	ST-57	Apple Leopoldstrasse	Munich	Germany

CALCULATE THE PERCENTAGE OF WARRANTY CLAIMS MARKED AS "WARRANTY VOID "

```
select
round(count(claim_id) / (select count(*) from warranty ) :: numeric * 100 , 2 ) as repair_status_percentage
from warranty
where repair_status = 'Warranty Void'
```

	repair_status_percentage	lock
1	21.85	🔒

HOW MANY WARRANTY CLAIMS WERE FILED IN 2020?

```
select count(claim_id) as total_count  
from warranty  
where extract(year from to_date(claim_date , 'mm dd, yyyy')) = 2020
```

total_count	bigint
1	28

DETERMINE HOW MANY WARRANTY CLAIMS WERE FILED FOR PRODUCTS LAUNCHED IN THE LAST TWO YEARS.

```
select
p.product_name,
COUNT(w.claim_id) as no_claim,
COUNT(s.sale_id)
FROM warranty as w
RIGHT JOIN
sales as s
ON s.sale_id = w.sale_id
JOIN product as p
ON p.product_id = s.product_id
WHERE to_date(p.launch_date , 'mm dd , yyyy' ) >= CURRENT_DATE - INTERVAL '2 years '
GROUP BY 1
HAVING COUNT(w.claim_id) > 0
```

	product_name character varying (50)	no_claim bigint	count bigint
1	HomePod 2nd Gen	1	7

FIND THE AVERAGE PRICE OF PRODUCTS IN EACH CATEGORY.

```
select category_id ,  
round (avg(price) , 2)  as avg_price  
from product  
group by 1
```

	category_id character varying (10)	avg_price numeric
1	CAT-8	188.75
2	CAT-10	150.25
3	CAT-5	440.67
4	CAT-9	249.00
5	CAT-4	766.69
6	CAT-3	726.78
7	CAT-1	1499.00
8	CAT-7	2665.67
9	CAT-6	171.50
10	CAT-2	249.00

DETERMINE THE PERCENTAGE CHANCE OF RECEIVING WARRANTY CLAIMS AFTER EACH PURCHASE FOR EACH COUNTRY

```
SELECT
country,
total_unit_sold,
total_claim,
COALESCE(total_claim :: numeric/total_unit_sold:: numeric * 100, 0 )
as risk
FROM
(SELECT
st.country,
SUM (s.quantity) as total_unit_sold,
COUNT(w.claim_id) as total_claim
FROM sales as s
JOIN stores as st
ON s.store_id = st.store_id
LEFT JOIN
warranty as w
ON w.sale_id = s.sale_id
group by 1) t1
order by 4 desc
```

	country character varying (30)	total_unit_sold bigint	total_claim bigint	risk numeric
1	Greece	89	7	7.86516853932584269700
2	Japan	61	3	4.91803278688524590200
3	Switzerland	143	7	4.89510489510489510500
4	Thailand	87	4	4.59770114942528735600
5	Mexico	22	1	4.545454545454545500
6	Portugal	116	5	4.31034482758620689700
7	Sweden	146	6	4.10958904109589041100
8	Canada	197	8	4.06091370558375634500
9	Russia	99	4	4.04040404040404040400
10	Australia	235	9	3.82978723404255319100
11	Italy	146	5	3.42465753424657534200
12	India	270	9	3.333333333333333333300
13	Singapore	154	5	3.24675324675324675300
14	Argentina	132	4	3.03030303030303030300
15	Germany	118	3	2.54237288135593220300
16	Netherlands	88	2	2.27272727272727272700

CALCULATE HOW MANY WARRANTY CLAIMS WERE FILED WITHIN 180 DAYS OF A PRODUCT SALE.

```
select count(*)  
from warranty w  
join sales s  
on w.sale_id = s.sale_id  
where to_date(w.claim_date,'mm dd ,yyyy') <= to_date(sale_date , 'dd mm ,yyyy') - interval '6 months'
```

	count	
1	29	

LIST THE MONTHS IN THE LAST THREE YEARS AND WITH THE SALES IN THESE IN THE USA.

```
select
substring(sale_date , 4 ,10) as new_date ,
st.country ,
count(quantity)
from sales s
join stores st
on s.store_id = st.store_id
where st.country = 'USA'
    and extract (year from to_date(sale_date , 'dd mm ,yyyy')) >= extract(year from current_date ) - 3
group by 1,2
order by 3 desc
HAVING
```

	new_date text	country character varying (30)	count bigint
1	05-2023	USA	3
2	08-2021	USA	2
3	09-2024	USA	1
4	01-2023	USA	1
5	03-2023	USA	1
6	11-2021	USA	1
7	07-2023	USA	1
8	11-2023	USA	1
9	10-2021	USA	1
10	04-2022	USA	1
11	07-2024	USA	1
12	02-2023	USA	1
13	03-2022	USA	1

IDENTIFY THE PRODUCT CATEGORY WITH THE MOST WARRANTY CLAIMS FILED IN THE LAST TWO YEARS.

```
select * from warranty
select * from sales
select * from product

select
p.category_id ,
count(claim_id) as total_claim
from warranty w
join sales s
on w.sale_id = s.sale_id
join product p
on s.product_id = p.product_id
where to_date(claim_date , 'mm dd , yyyy') >= current_date - interval ' 2 years '
group by 1
order by 2
```

	category_id character varying (10)	total_claim bigint
1	CAT-2	1
2	CAT-8	1
3	CAT-7	2
4	CAT-1	2
5	CAT-5	2
6	CAT-6	3
7	CAT-9	3
8	CAT-10	4
9	CAT-3	5
10	CAT-4	5

IDENTIFY THE STORE WITH THE HIGHEST PERCENTAGE OF "PAID REPAIRED" CLAIMS RELATIVE TO TOTAL CLAIMS FILED.

```
WITH paid_repair
AS
(SELECT
s.store_id,
COUNT(w.claim_id) AS paid_repaired
FROM sales AS s
RIGHT JOIN warranty AS w
ON w.sail_id = s.sale_id
WHERE w.repair_status = 'Paid Repaired'
GROUP BY 1
),
total_repaired
AS
(SELECT
s.store_id,
COUNT(w.claim_id) AS total_repaired
FROM sales AS s
RIGHT JOIN warranty AS w
ON w.sail_id = s.sale_id
GROUP BY 1)
SELECT
tr.store_id,
pr.paid_repaired,
tr.total_repaired,
ROUND(pr.paid_repaired:: numeric/
tr.total_repaired:: numeric * 100 ,2) AS percentage_paid_repaired
FROM paid_repair AS pr
JOIN
total_repaired tr
ON pr.store_id = tr.store_id
join stores AS st
on tr.store_id = st.store_id|
```

	store_id character varying (10)	store_name character varying (60)	paid_repaired bigint	total_repaired bigint	percentage_paid_repaired numeric
1	ST-13	Apple Italian Square	2	3	66.67
2	ST-17	Apple Punta Cana Plaza	1	1	100.00
3	ST-21	Apple Sandton City	1	1	100.00
4	ST-24	Apple Pacifico Mall	1	4	25.00
5	ST-25	Apple Fashion District	1	2	50.00
6	ST-27	Apple Tverskaya Street	1	4	25.00
7	ST-28	Apple Orchard Road	1	4	25.00
8	ST-29	Apple Loop Street	1	3	33.33
9	ST-42	Apple Palacio de Cibeles	1	1	100.00
10	ST-43	Apple Acropolis Center	1	7	14.29
11	ST-52	Apple Avenida da Liberdade	1	3	33.33
12	ST-53	Apple Bahnhofstrasse	2	6	33.33
13	ST-62	Apple Drottninggatan	1	4	25.00
14	ST-64	Apple Fifth Avenue	1	1	100.00
15	ST-66	Apple Kingdom Street	1	1	100.00

WRITE A QUERY TO CALCULATE THE MONTHLY RUNNING TOTAL OF SALES FOR EACH STORE OVER THE PAST FOUR YEARS AND COMPARE TRENDS DURING THIS PERIOD.

```
with monthly_sales
as
(
select
store_id ,
Extract(year from to_date (sale_date,'dd mm,yyyy')) AS year ,
Extract(month from to_date (sale_date,'dd mm ,yyyy')) AS month ,
sum(p.price * s.quantity ) as total_revenue
from sales as s
join
product as p
on s.product_id = p.product_id
group by 1,2,3
order by 1,2,3
)

select
store_id ,
month ,
year ,
total_revenue ,
sum(total_revenue) over (partition by store_id order by year , month ) as running_total
from monthly_sales
```

	store_id character varying (10)	month numeric	year numeric	total_revenue bigint	running_total numeric
1	ST-1	2	2016	3225	3225
2	ST-1	12	2018	2400	5625
3	ST-1	10	2021	2780	8405
4	ST-10	3	2019	24975	24975
5	ST-10	5	2022	21588	46563
6	ST-10	3	2023	6993	53556
7	ST-11	4	2016	1112	1112
8	ST-11	6	2020	18683	19795
9	ST-11	2	2021	2148	21943
10	ST-11	3	2021	35985	57928
11	ST-11	10	2021	6578	64506
12	ST-11	12	2021	3043	67549
13	ST-11	3	2024	5586	73135
14	ST-11	8	2024	1937	75072
15	ST-12	5	2015	12627	12627
16	ST-12	3	2016	930	13557
17	ST-12	8	2017	11980	25537

PRESENTED BY SHIVANG DUBEY

Thank you !

