GlobalTech Project – SQL Queries

Tool Used – PostgreSQL

Chirag Sharma

Data Analyst

Query | Query History

```sql
--- Find the total revenue per product category for each year, sorted by current year first.
select extract(year from date) as years, p.category, round(sum(s.sales_amount)/10000000,1) || 'M' as Sales
from salesdata as s
 join dimproduct as p
ON s.productkey = p.productkey
group by 1,2
order by 1 desc
```

Data Output | Messages | Notifications

SQL

	years numeric	category text	sales text
1	2025	Beauty & Personal Care	19.4M
2	2025	Books & Media	20.5M
3	2025	Clothing	21.2M
4	2025	Electronics	20.9M
5	2025	Home Appliances	19.1M
6	2025	Sports & Outdoors	20.8M
7	2024	Beauty & Personal Care	28.1M
8	2024	Books & Media	29.8M

Query | Query History

```sql
--- Identify the top 5 products contributing the most to overall sales revenue in the last 12 months.

select p.product_name, sum(s.sales_amount) as Sales
from salesdata as s
join dimproduct as p
ON s.productkey = p.productkey
where s.date >= Current_Date - interval '12 months'
group by 1
order by 2 desc
limit 5

```

Data Output | Messages | Notifications

	product_name text	sales numeric
1	Smart Beam 806	6984563
2	Ultra Go 352	6894529
3	Prime One 369	6866363
4	Max Wave 690	6816128
5	Power Plus 344	6565537

Query | Query History

```sql
--- Calculate the year-over-year sales growth percentage for each product category.

select years, category, sales, trim(case when growth is null then '' else growth end) as growth from
(select *, round(((sales / py) -1)*100, 0) || '%' as Growth from
(select *, lag(sales)over(partition by years) as PY from
(select extract(year from s.date) as years, p.category, round(sum(s.sales_amount)/10000000,1) as Sales
from salesdata as s
join dimproduct as p
ON s.productkey = p.productkey
group by 1, 2
order by 1 desc) as i) as u)
```

Data Output | Messages | Notifications

SQL

	years numeric	category text	sales numeric	growth text
1	2023	Sports & Outdoors	29.1	
2	2023	Home Appliances	29.8	2%
3	2023	Electronics	29.3	-2%
4	2023	Beauty & Personal Care	27.2	-7%
5	2023	Clothing	29.5	8%
6	2023	Books & Media	27.8	-6%
7	2024	Electronics	31.8	
8	2024	Beauty & Personal Care	28.1	-12%

Query | Query History

```sql
--- Find the average selling price per product and compare it to the average unit cost —
-- highlight products with a profit margin below 60%.

select product_name from
(select *, round(((avgprice - avgcost)/avgprice)*100,0) as profitmargin from
(select p.product_name, round(avg(unitprice),1) as avgprice, round(avg(unitcost),1) as avgCost
from salesdata as s
join dimproduct as p
ON s.productkey = p.productkey
group by 1) as i)
where profitmargin <= 60
```

Data Output | Messages

	product_name text
1	Air Flex 692
2	Air Plus 288
3	Air Plus 733
4	Eco Beam 794
5	Eco Edge 195
6	Eco Fit 957
7	Eco Fit 988
8	Eco Touch 236

Query | Query History

```sql
--- Calculate the average discount percentage given per product category and rank categories by discount dependency.

select *, rank()over(order by avgdiscount desc) as ranks from
(select p.category, round(avg(s.discountamount),1) as avgdiscount
from salesdata as s
join dimproduct as p
ON s.productkey = p.productkey
group by 1) as i

```

Data Output | Messages | Notifications

	category text	avgdiscount numeric	ranks bigint
1	Books & Media	56.4	1
2	Home Appliances	55.5	2
3	Sports & Outdoors	55.4	3
4	Clothing	54.3	4
5	Beauty & Personal Care	54.0	5
6	Electronics	52.2	6

Query | Query History

```sql
--- Identify products where stock type is 'High' but sales quantity is below the average in the last quarter.

select distinct p.product_name from salesdata as s
join dimproduct as p
ON s.productkey = p.productkey
where p.stocktypename = 'High' and s.salesquantity <= (select avg(salesquantity) from salesdata)
AND s.date >= (
        SELECT MAX(date) - INTERVAL '3 months'
        FROM salesdata
      )
```

Data Output | Messages

	product_name text
1	Air Beam 324
2	Air Beam 694
3	Air Beam 713
4	Air Beam 776
5	Air Beam 799
6	Air Beam 910
7	Air Beam 989
8	Air Edge 158

Query | Query History

```sql
--- Find the slow-moving product subcategory (lowest % sales quantity) in the last year and their current stock status.
select p.subcategory, p.stocktypename, round(sum(s.salesquantity)/
(select sum(salesquantity) from salesdata)*100,1) || '%' as Quantitygrowth
from salesdata as s
join dimproduct as p
ON s.productkey = p.productkey
where extract(year from s.date) = extract(year from current_date) - 1
group by 1,2
order by 3
limit 3
```

Data Output | Messages | Notifications

	subcategory text	stocktypename text	quantitygrowth text
1	Laptops	High	0.3%
2	Fragrances	High	0.4%
3	Footwear	High	0.4%

Query | Query History

```sql
--- Calculate the total profit per manufacturer and rank them by profitability.


select *, rank()over(order by profit desc) as ranks from
(select *, (sales - cost) || ' M' as Profit from
(select p.manufacturer, round(sum(s.totalcost)/10000000, 1) as Cost , round(sum(s.sales_amount)/10000000, 1) as Sales
from salesdata as s
join dimproduct as p
ON s.productkey = p.productkey
group by 1) as p
order by 4 desc)
```

Data Output | Messages | Notifications

	manufacturer text	cost numeric	sales numeric	profit text	ranks bigint
1	HomePro	20.0	56.6	36.6 M	1
2	TechNova	20.5	56.5	36.0 M	2
3	StyleSphere	19.3	54.2	34.9 M	3
4	EcoWare	18.3	52.2	33.9 M	4
5	CoolBreeze	17.8	49.7	31.9 M	5
6	GadgetX	17.9	49.5	31.6 M	6
7	GlowNest	16.1	46.1	30.0 M	7
8	UrbanVibe	14.9	42.0	27.1 M	8

Query | Query History

```sql
--- Identify loss-making transactions (where total cost > total revenue) and their frequency per product.

select p.product_name, count(*) as Frequency
from salesdata as s
JOIN dimproduct as p
ON s.productkey = p.productkey
where s.totalcost > s.sales_amount
group by 1
order by 2 desc
limit 5
```

Data Output | Messages | Notifications

	product_name text	frequency bigint
1	Prime Go 162	49
2	Max Beam 938	47
3	Eco Fit 331	47
4	Power Flex 811	47
5	Eco X 502	46

Query | Query History

```sql
--- Determine the product category with the highest average profit per transaction over the past two years.

select p.category, round(avg(s.sales_amount - s.totalcost),1) as Profit
from salesdata as s
join dimproduct as p
ON p.productkey = s.productkey
where extract(year from s.date) >= extract(year from current_date) - 2
group by 1
```

Data Output | Messages | Notifications

	category text	profit numeric
1	Beauty & Personal Care	18178.9
2	Books & Media	18199.7
3	Clothing	17772.3
4	Electronics	17833.5
5	Home Appliances	17828.1
6	Sports & Outdoors	18136.1

Query | Query History

```sql
--- Rank sales channels (channelKey) by total sales revenue for the last 6 months.

select *, rank()over(order by revenue_in_Millions desc) from
(select channelkey, round(sum(sales_amount)/10000000,1) as Revenue_in_Millions from salesdata
where date <= current_date - interval '6 month'
group by 1)as p
```

Data Output | Messages | Notifications

	channelkey bigint	revenue_in_millions numeric	rank bigint
1	3	170.5	1
2	2	122.7	2
3	4	85.0	3
4	1	9.7	4

Query | Query History

```sql
--- Find stores with the best category that shows the highest sales contribution

select storekey, category, sales, contribution || '%' as contribution from
(select *, row_number()over(partition by storekey order by Contribution desc) from
(select *, round((sales/nofilter) * 100,0) as Contribution from
(select *, sum(sales)over(partition by storekey) as nofilter from
(select s.storekey, p.category, round(sum(sales_amount)/10000000,1) as Sales from salesdata as s
join dimproduct as p
ON p.productkey = s.productkey
group by 1,2) as i))as p)
where row_number = 1
```

Data Output | Messages | Notifications

	storekey bigint	category text	sales numeric	contribution text
1	200	Electronics	34.7	17%
2	215	Books & Media	1.0	21%
3	234	Electronics	0.6	22%
4	240	Electronics	0.8	20%
5	306	Home Appliances	27.5	18%
6	309	Sports & Outdoors	18.9	18%

Query | Query History

```sql
--- Identify the fastest-growing product by sales quantity in each category over the last 3 months.

select category, product_name, quantity from
(select *, rank()over(partition by category order by quantity desc) as ranks from
(select p.category, p.product_name, sum(s.salesquantity) as Quantity from salesdata as s
join dimproduct as p
ON s.productkey = p.productkey
where s.date >= current_date - interval '3 months'
group by 1,2) as i)
where ranks = 1
```

Data Output | Messages | Notifications

	category text	product_name text	quantity numeric
1	Beauty & Personal Care	Lite Wave 70	11199
2	Books & Media	Pro One 501	13154
3	Clothing	Max Beam 759	11762
4	Electronics	Power Flex 326	14657
5	Home Appliances	Ultra Flex 595	13038
6	Sports & Outdoors	Power Beam 251	12494