Complete SQL Analysis Report

A1

Calculates total units sold per product to identify top-selling SKUs.

```sql
SELECT productid, SUM(inventoryquantity) As Total_Unit_Sold
FROM sales
GROUP BY  productid
ORDER BY Total_Unit_Sold DESC;
```

A2

Analyzes monthly and yearly sales volume by category for demand trends.

```sql
SELECT s.sales_month, s.sales_year ,p.productcategory, SUM(s.inventoryquantity) As sales_volume
FROM sales s
JOIN product p  ON p.productid = s.productid
GROUP BY s.sales_year , s.sales_month,p.productcategory
ORDER BY sales_volume DESC;
```

A3

Shows sales volume by category for a specific month and year.

```
SELECT p.productcategory, SUM(s.inventoryquantity) As sales_volume
FROM sales s
JOIN product p  ON p.productid = s.productid
WHERE s.sales_year= '2021' AND s.sales_month='11'
GROUP BY p.productcategory
ORDER BY sales_volume DESC;
```

A4

Compares inflation rate and sales volume over time.

```
SELECT s.sales_month, s.sales_year, AVG(f.inflationrate) AS Avg_inflation,
SUM (s.inventoryquantity) AS sales_volume
FROM sales s
JOIN factors f ON f.salesdate= s.salesdate
GROUP BY sales_year, sales_month;
```

A5

Summarizes inflation vs. sales volume over the last 12 months.

```
SELECT s.sales_month, s.sales_year, AVG(f.inflationrate) AS Avg_inflation,
SUM (s.inventoryquantity) AS sales_volume
FROM sales s
JOIN factors f ON f.salesdate= s.salesdate
GROUP BY sales_year, sales_month;
```

A6

Average sales without promotions—baseline demand.

```
SELECT p.productcategory, ROUND(AVG(s.inventoryquantity)) AS Avg_sales_without_promotion,
p.promotions
FROM sales s
JOIN product p ON p.productid = s.productid
WHERE p.promotions = 'No'
GROUP BY p.productcategory, p.promotions

UNION ALL
```

A7

Average sales with promotions—promotion-driven demand.

```
SELECT p.productcategory, ROUND(AVG(s.inventoryquantity)) AS Avg_sales_without_promotion,
p.promotions
FROM sales s
JOIN product p ON p.productid = s.productid
WHERE p.promotions = 'Yes'
GROUP BY p.productcategory, p.promotions;
```

A8

Calculates overall category-level average sales.

```
SELECT p.productcategory, ROUND(AVG(s.inventoryquantity)) AS Avg_sales
FROM sales s
JOIN product p ON p.productid = s.productid
GROUP BY p.productcategory;
```

A9

Yearly GDP vs yearly total sales comparison.

```sql
SELECT s.sales_year, SUM(f.GDP) AS Total_GDP, SUM(s.inventoryquantity) AS Total_sales
FROM sales s
JOIN factors f ON f.salesdate = s.salesdate
GROUP BY s.sales_year
ORDER BY total_sales DESC;
```

A10

Top 10 products by total units sold.

```sql
SELECT productid, SUM(inventoryquantity) AS Units_Sold
FROM sales
GROUP BY productid
ORDER BY units_sold DESC
LIMIT 10;
```

A11

Aggregates GDP and total sales by year.

```sql
SELECT s.sales_year, SUM(f.GDP) AS Total_GDP, SUM(s.inventoryquantity) AS Total_sales
FROM sales s
JOIN factors f ON f.salesdate = s.salesdate
GROUP BY s.sales_year
ORDER BY total_sales DESC;
```

A12

Top 10 selling products — alternative screenshot.

```sql
SELECT productid, SUM(inventoryquantity) AS Units_Sold
FROM sales
GROUP BY productid
ORDER BY units_sold DESC
LIMIT 10;
```

A13

Seasonal factor analysis by category.

```
SELECT p.productcategory,ROUND (AVG(f.seasonalfactor),4) AS average_seasonal_factor,
SUM(s.inventoryquantity) AS Total_Sales
FROM sales s
JOIN product p ON p.productid = s.productid
JOIN factors f ON f.salesdate = s.salesdate
GROUP BY productcategory
ORDER BY average_seasonal_factor;
```

A14

Average sales and promotion count per category.

```
SELECT p.productcategory, AVG(s.inventoryquantity) AS avg_sales_quantity,
COUNT(CASE WHEN p.promotions = 'Yes' THEN 1 END ) AS promotion_count
FROM  sales s
JOIN product p on s.productid = p.productid
GROUP BY p.productcategory
ORDER BY avg_sales_quantity;
```

A15

Duplicate of Total Units Sold analysis.

```
SELECT productid, SUM(inventoryquantity) As Total_Unit_Sold
FROM sales
GROUP BY  productid
ORDER BY Total_Unit_Sold DESC;
```

A17

Sales volume by month/year and a month-specific snapshot.

```
SELECT s.sales_month, s.sales_year ,p.productcategory, SUM(s.inventoryquantity) As sales_volume
FROM sales s
JOIN product p  ON p.productid = s.productid
GROUP BY s.sales_year , s.sales_month,p.productcategory
ORDER BY sales_volume DESC;
SELECT p.productcategory, SUM(s.inventoryquantity) As sales_volume
FROM sales s
JOIN product p  ON p.productid = s.productid
WHERE s.sales_year= '2021' AND s.sales_month='11'
GROUP BY p.productcategory
ORDER BY sales_volume DESC;
```

A18

Inflation vs. sales grouped by month and year.

```
SELECT s.sales_month, s.sales_year, AVG(f.inflationrate) AS Avg_inflation,
SUM (s.inventoryquantity) AS sales_volume
FROM sales s
JOIN factors f ON f.salesdate= s.salesdate
GROUP BY sales_year, sales_month;
```

A19

Inflation and sales trends over the last 12 months.

```
SELECT s.sales_month, s.sales_year, AVG(f.inflationrate) AS Avg_inflation,
SUM (s.inventoryquantity) AS Total_sales_volume
FROM sales s
JOIN Factors f ON f.salesdate=s.salesdate
WHERE s.salesdate >= (current_date - Interval '1 year' )
GROUP BY sales_year,sales_month
ORDER BY sales_year,sales_month
```

A20

Baseline sales without promotion.

```
SELECT p.productcategory, ROUND(AVG(s.inventoryquantity)) AS Avg_sales_without_promotion,
p.promotions
FROM sales s
JOIN product p ON p.productid = s.productid
WHERE p.promotions = 'No'
GROUP BY p.productcategory, p.promotions
```

A21

Duplicate baseline sales without promotion.

```
SELECT p.productcategory, ROUND(AVG(s.inventoryquantity)) AS Avg_sales_without_promotion,
p.promotions
FROM sales s
JOIN product p ON p.productid = s.productid
WHERE p.promotions = 'No'
GROUP BY p.productcategory, p.promotions
```

A22

Average sales under promotions.

```
SELECT p.productcategory, ROUND(AVG(s.inventoryquantity)) AS Avg_sales_without_promotion,
p.promotions
FROM sales s
JOIN product p ON p.productid = s.productid
WHERE p.promotions = 'Yes'
GROUP BY p.productcategory, p.promotions;
```

A24

Category-level average sales summary.

```
SELECT p.productcategory, ROUND(AVG(s.inventoryquantity)) AS Avg_sales
FROM sales s
JOIN product p ON p.productid = s.productid
GROUP BY p.productcategory;
```

A26

Yearly GDP and sales summary (duplicate screenshot).

```
SELECT s.sales_year, SUM(f.GDP) AS Total_GDP, SUM(s.inventoryquantity) AS Total_sales
FROM sales s
JOIN factors f ON f.salesdate = s.salesdate
GROUP BY s.sales_year
ORDER BY total_sales DESC;
```

A28

Top 10 units sold per product (duplicate screenshot).

```
SELECT s.sales_year, SUM(f.GDP) AS Total_GDP, SUM(s.inventoryquantity) AS Total_sales
FROM sales s
JOIN factors f ON f.salesdate = s.salesdate
GROUP BY s.sales_year
ORDER BY total_sales DESC;
```

A30

Seasonality factor vs total sales per category.

```sql
SELECT productid, SUM(inventoryquantity) AS Units_Sold
FROM sales
GROUP BY productid
ORDER BY units_sold DESC
LIMIT 10;
```

A32

Average sales + promotion counts per category (duplicate screenshot).

```sql
SELECT p.productcategory,ROUND (AVG(f.seasonalfactor),4) AS average_seasonal_factor,
SUM(s.inventoryquantity) AS Total_Sales
FROM sales s
JOIN product p ON p.productid = s.productid
JOIN factors f ON f.salesdate = s.salesdate
GROUP BY productcategory
ORDER BY average_seasonal_factor;
```

A35

Category-level averages and promo influence summary.

```
SELECT p.productcategory, AVG(s.inventoryquantity) AS avg_sales_quantity,
COUNT(CASE WHEN p.promotions = 'Yes' THEN 1 END ) AS promotion_count
FROM  sales s
JOIN product p on s.productid = p.productid
GROUP BY p.productcategory
ORDER BY avg_sales_quantity;
```