

# Apple Sales and Warranty Analysis Project

## Introduction

This project analyzes Apple product sales and warranty claims using SQL. It provides insights into store performance, product sales trends, and warranty-related metrics across different regions and time periods.

---

## Database Tables Used

- **category**
- **products**
- **stores**
- **sales**
- **warranty**

## Performance Optimization

EXPLAIN ANALYZE

SELECT \* FROM sales

WHERE product\_id='P-44';

CREATE INDEX sales\_product\_id ON sales(product\_id);

CREATE INDEX sales\_store\_id ON sales(store\_id);

CREATE INDEX sales\_sale\_date ON sales(sale\_date);

---

## Queries and Solutions

### 1. Find the number of stores in each country

SELECT country, COUNT(store\_id) AS Total\_orders

FROM stores

GROUP BY country

ORDER BY 2 DESC;

### 2. Total number of units sold by each store

SELECT

```
    st.store_id,  
    st.store_name,  
    SUM(quantity) AS Total_units_sold  
FROM sales AS s  
JOIN stores AS st ON st.store_id = s.store_id  
GROUP BY 1, 2  
ORDER BY 3 DESC;
```

### **3. Sales in December 2023**

```
SELECT COUNT(sale_id) AS Total_sale  
FROM sales  
WHERE TO_CHAR(Sale_date, 'MM-YYYY') = '12-2023';
```

### **4. Stores with no warranty claims**

```
SELECT COUNT(*)  
FROM stores  
WHERE store_id NOT IN (  
    SELECT DISTINCT store_id  
    FROM sales AS s  
    RIGHT JOIN warranty AS w ON s.sale_id = w.sale_id  
);
```

### **5. Percentage of "Warranty Void" claims**

```
SELECT  
    COUNT(claim_id) * 100.0 / (SELECT COUNT(*) FROM warranty) AS  
warranty_void_percentage  
FROM warranty  
WHERE repair_status='Warranty Void';
```

### **6. Store with highest total units sold in the last year**

```
SELECT  
    s.store_id,
```

```
    st.store_name,  
    SUM(s.quantity)  
FROM sales AS s  
JOIN stores AS st ON s.store_id = st.store_id  
WHERE sale_date >= (CURRENT_DATE - INTERVAL '1 year')  
GROUP BY s.store_id, st.store_name  
ORDER BY SUM(quantity) DESC  
LIMIT 1;
```

### **7. Unique products sold in the last year**

```
SELECT COUNT(DISTINCT product_id)  
FROM sales  
WHERE sale_date >= (CURRENT_DATE - INTERVAL '1 year');
```

### **8. Average product price in each category**

```
SELECT  
    p.category_id,  
    c.category_name,  
    AVG(p.price) AS AVERAGE_PRICE  
FROM products AS p  
JOIN category AS c ON p.category_id = c.category_id  
GROUP BY 1, 2  
ORDER BY 3 DESC;
```

### **9. Warranty claims filed in 2020**

```
SELECT COUNT(*) AS warranty_claim_2020  
FROM warranty  
WHERE EXTRACT(YEAR FROM claim_date) = 2020;
```

### **10. Best-selling day for each store**

```
SELECT *  
FROM (
```

```

SELECT
    store_id,
    TO_CHAR(sale_date, 'Day') AS Day_Name,
    SUM(quantity) AS total_unit_sold,
    RANK() OVER (PARTITION BY store_id ORDER BY SUM(quantity) DESC) AS Rank
FROM sales
GROUP BY 1, 2
) AS t1
WHERE Rank = 1;

```

### **11. Least selling product per country per year**

```

WITH product_rank AS (
    SELECT
        st.country,
        p.product_name,
        SUM(s.quantity) AS total_qty_sold,
        RANK() OVER (PARTITION BY st.country ORDER BY SUM(s.quantity)) AS rank
    FROM sales AS s
    JOIN stores AS st ON s.store_id = st.store_id
    JOIN products AS p ON p.product_id = s.product_id
    GROUP BY 1, 2
)
SELECT * FROM product_rank WHERE rank = 1;

```

### **12. Warranty claims filed within 180 days of sale**

```

SELECT COUNT(*)
FROM warranty AS w
LEFT JOIN sales AS s ON s.sale_id = w.sale_id
WHERE (w.claim_date - sale_date <= 180);

```

### **13. Warranty claims for products launched in last two years**

```

SELECT
    p.product_name,
    COUNT(w.claim_id) AS no_claim
FROM warranty AS w
RIGHT JOIN sales AS s ON s.sale_id = w.sale_id
JOIN products AS p ON p.product_id = s.product_id
WHERE p.launch_date >= CURRENT_DATE - INTERVAL '2 years'
GROUP BY 1
HAVING COUNT(w.claim_id) > 0;

```

#### **14. Months in the last 3 years where USA sales exceeded 5000 units**

```

SELECT
    TO_CHAR(sale_date, 'MM-YYYY') AS month,
    SUM(s.quantity) AS Total_Unit_Sold
FROM sales AS s
JOIN stores AS st ON s.store_id = st.store_id
WHERE st.country = 'USA' AND s.sale_date >= CURRENT_DATE - INTERVAL '3 years'
GROUP BY 1
HAVING SUM(s.quantity) > 5000;

```

#### **15. Product category with most warranty claims in last 2 years**

```

SELECT
    c.category_name,
    COUNT(claim_id) AS total_claims
FROM warranty AS w
LEFT JOIN sales AS s ON w.sale_id = s.sale_id
JOIN products AS p ON p.product_id = s.product_id
JOIN category AS c ON c.category_id = p.category_id
WHERE w.claim_date >= CURRENT_DATE - INTERVAL '2 years'
GROUP BY 1;

```

## 16. Probability of warranty claims per 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
) sub
ORDER BY 4 DESC;
```

## 20. Monthly running total of sales for each store over the past four years

```
WITH monthly_sales AS (
    SELECT
        store_id,
        EXTRACT(YEAR FROM sale_date) AS year,
        EXTRACT(MONTH FROM sale_date) AS month,
        SUM(p.price * s.quantity) AS total_revenue
    FROM sales AS s
    JOIN products AS p ON s.product_id = p.product_id
    GROUP 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;
```

---

This document provides a structured overview of the Apple Sales and Warranty Analysis Project, including SQL queries for analysis. 🚀