7

DATA ANALYTICS JOURNEY

Building a Product-Wise Sales Report Using SQL

FAISAL SHAHZAD
DATA ANALYST

1

Building a Product-Wise Sales Report (FY2021) Using SQL

In this task, I built an end-to-end **product-level sales report** using **MySQL**, covering:

- Sold quantity
- Product & variant details
- Gross price per item
- Total gross revenue

All calculated accurately using **fiscal year–based logic.**

2 Business Requirement

Business Requirement

The goal was to generate a **product-wise sales report** for a specific fiscal year that includes:

1. Date / Month
2. Product Name
3. Variant
4. Sold Quantity
5. Gross Price per Item
6. Gross Price Total

Purpose:

To enable easy analysis and reporting in Excel.

3 Key Challenge & Solution

Challenge: Fiscal Year vs Calendar Year

Sales data is stored using **calendar dates,** but business reporting follows a **fiscal year (Apr–Mar)**.

Using calendar years directly would result in:

- Incorrect fiscal grouping
- Wrong price mapping
- Inaccurate reporting

Solution

To solve this, I created a reusable SQL function that converts any calendar date into its correct fiscal year.

This function is then used consistently across joins and filters.

4

Fiscal Year Function Code

-- BLOCK 1: Get raw sales data WITH january_sales AS (SELECT product_id, SUM(quantity

```sql
CREATE FUNCTION get_fiscal_year (calendar_date DATE)
RETURNS INTEGER
DETERMINISTIC
BEGIN
  -- Declare a variable to store the fiscal year
  DECLARE fiscal_year INT;

  -- Add 4 months to the given calendar date
  -- This shifts Jan-Mar dates into the correct fiscal year
  SET fiscal_year = YEAR(DATE_ADD(calendar_date, INTERVAL 4 MONTH));

  -- Return the calculated fiscal year
  RETURN fiscal_year;
END;
```

Logic:

Calendar Date → +4 Months → Extract Year → Fiscal Year

5 Why This Works & Data Model

Why This Approach Works

By shifting the calendar date by 4 months, all sales records are correctly aligned with the fiscal year (Apr–Mar).

This ensures that sales data and pricing data always match the same fiscal year.

Data Used

- fact_sales_monthly → sales date & sold quantity
- fact_gross_price → product pricing by fiscal year
- dim_product → product name & variant details

All tables are joined using product_code and the calculated fiscal year.

6

Final SQL Query

-- BLOCK 1: Get raw sales data WITH january_sales AS (SELECT product_id, SUM(quantity

```sql
-- Product-wise sales report with fiscal year-aligned pricing
SELECT
  s.date,
  p.product AS product_name,
  s.sold_quantity,
  p.variant,
  g.gross_price AS gross_price,
  (g.gross_price * s.sold_quantity) AS gross_price_total
FROM fact_sales_monthly s

-- Join gross price using product code and derived fiscal year
JOIN fact_gross_price g
  ON s.product_code = g.product_code
 AND g.fiscal_year = get_fiscal_year(s.date)

-- Join product dimension to get product and variant details
JOIN dim_product p
  ON s.product_code = p.product_code

-- Filter for a specific customer and fiscal year
WHERE s.customer_code = 90002002
  AND get_fiscal_year(s.date) = 2021

-- Show latest dates first
ORDER BY s.date DESC;
```

7 Final Output & Excel Report

Final Report Output

The final SQL query produces a product-wise sales report with:

- Date / Month
- Product Name & Variant
- Sold Quantity
- Gross Price per Item
- Gross Price Total

This report is then exported to Excel for further analysis and reporting.

📎 The **Excel file** is attached below this post.

FAISAL SHAHZAD

Thanks!

Follow me for another tips

DATA ANALYSTS