

1. Membuat Temporary Table

Pertama-tama kita membuat temporary table terlebih dahulu untuk menampung kolom yang telah dimodifikasi sedemikian rupa untuk memenuhi informasi dari produk beserta waktu transaksinya

Langkah 1:

```
CREATE TEMP TABLE report_monthly_orders_product_agg AS
```

Membuat temporary tabel 'report_monthly_orders_product_agg' dengan perintah 'CREATE TEMP TABLE', dimana tabel tersebut terdiri dari beberapa kolom

Langkah 2:

```
SELECT
  p.name AS product_name,
  p.brand,
  p.category,
  FORMAT_TIMESTAMP('%Y-%m', o.created_at) AS month,
  o.status,
  COUNT(o.product_id) AS total_sales,
  SUM(o.sale_price) AS total_revenue
```

Memunculkan kolom yang terdiri dari: product_name, month, status, total_sales, dan total_revenue

product_name didapatkan dari kolom name pada tabel

`bigquery-public-data.thelook_ecommerce.products` yang dilambangkan dengan p

brand didapatkan dari kolom name pada tabel

`bigquery-public-data.thelook_ecommerce.products` yang dilambangkan dengan p

category didapatkan dari kolom name pada tabel

`bigquery-public-data.thelook_ecommerce.products` yang dilambangkan dengan p

month didapatkan dari kolom created_at pada tabel

`bigquery-public-data.thelook_ecommerce.order_items` yang dilambangkan dengan o, kolom month di modifikasi agar hanya memunculkan tahun-bulan saja dengan perintah 'FORMAT_TIMESTAMP'

status didapatkan dari kolom status pada tabel

`bigquery-public-data.thelook_ecommerce.order_items` yang dilambangkan dengan o

total_sales didapatkan dari perhitungan total munculnya product_id pada tabel

`bigquery-public-data.thelook_ecommerce.order_items` yang dilambangkan dengan o

total_revenue didapatkan dari penjumlahan sale_price untuk setiap produk pada tabel `bigquery-public-data.thelook_ecommerce.order_items` yang dilambangkan dengan o

Langkah 3:

```
FROM
|   `bigquery-public-data.thelook_ecommerce.products` p
LEFT JOIN
|   `bigquery-public-data.thelook_ecommerce.order_items` o
ON
|   p.id = o.product_id
```

Kolom diambil dari tabel `bigquery-public-data.thelook_ecommerce.products` yang dilambangkan dengan p dan melakukan irisan (LEFT JOIN) dengan tabel `bigquery-public-data.thelook_ecommerce.order_items` yang dilambangkan dengan o di product_id pada masing-masing tabel

Langkah 4:

```
WHERE
|   o.status = 'Complete'
```

Kueri ini berfungsi agar informasi produk yang ditampilkan hanyalah produk yang memiliki status complete

Langkah 5:

```
GROUP BY
|   product_name,
|   month,
|   o.status,
|   p.brand,
|   p.category
```

Tabel akan dikelompokkan sesuai dengan product_name, month, status, brand dan category

Langkah 6:

HAVING

```
month IS NOT NULL AND total_revenue IS NOT NULL AND p.brand IS NOT NULL;
```

Tabel difilter menggunakan perintah 'HAVING', dimana tabel hanya memunculkan hasil kolom month yang tidak kosong, total_revenue yang tidak kosong dan brand yang tidak kosong

Result

 temporary_table

2. Produk Dengan Penjualan Tertinggi Setiap Bulan

Setelah membuat temporary table, kita akan melakukan kueri untuk mencari produk dengan penjualan tertinggi setiap bulannya, pada kasus ini, penjualan tertinggi memiliki dua kriteria yaitu: total penjualan item tertinggi dan total pendapatan tertinggi

Langkah 1:

SELECT

```
month,  
product_name,  
brand,  
category,  
total_sales,  
ROUND(total_revenue,2) AS total_revenue,  
status
```

Memunculkan kolom month, product_name, brand, category, total_sales, total_revenue, dan status

untuk kolom total_revenue akan di dibulatkan menjadi 2 bilangan dengan perintah 'ROUND'

Langkah 2:

```
FROM  
  (  
    SELECT  
      month,  
      product_name,  
      brand,  
      category,  
      total_sales,  
      total_revenue,  
      status,  
      ROW_NUMBER() OVER (PARTITION BY month ORDER BY total_sales DESC, total_revenue DESC) as row_num  
    FROM  
      report_monthly_orders_product_agg  
  )
```

Kita akan melakukan subquery, dimana subquery ini berfungsi untuk memunculkan informasi produk berdasarkan partisi month yang diurutkan mulai dari total_sales terbesar (jika total_sales beberapa produk sama pada satu bulan, maka akan diurutkan kembali mulai dari total revenue yang terbesar) dan diambil dari tabel 'report_monthly_orders_product_agg'

Langkah 3:

```
WHERE  
  row_num = 1
```

Kueri ini memastikan agar hanya produk dengan penjualan tertinggi setiap bulannya (row_num = 1)

Langkah 4:

```
ORDER BY month ASC;
```

Kolom diurutkan berdasarkan bulan dengan perintah 'ORDER BY' mulai dari yang terkecil 'ASC'

Result

📄 product_per_month