

## 1. BigQuery scripting syntax

**BEGIN dan END:** Scripting syntax digunakan untuk menjalankan lebih dari satu query, dalam hal ini terdapat dua query utama yaitu terkait pembuatan temporary table data penjualan bulanan per produk dan query untuk mendapatkan data penjualan produk tertinggi tiap bulannya.

## 2. Query untuk membuat temporary table terkait data penjualan bulanan per produk

a. **CREATE TEMP TABLE report\_monthly\_orders\_product\_agg AS:** Query ini digunakan untuk membuat tabel yang sifatnya sementara karena hanya ada saat sesi query ini dieksekusi. Tabel sementara ini diberi nama "report\_monthly\_orders\_product\_agg".

b. CTE (Common Table Expression) untuk tabel "monthly\_sales"

**WITH monthly\_sales AS (...):** Dengan syntax WITH ... AS (...) berarti kita menggunakan CTE yang dalam hal ini berfungsi untuk membuat tabel sementara bernama "monthly\_sales".

**SELECT EXTRACT(YEAR FROM o.created\_at) AS year, EXTRACT(MONTH FROM o.created\_at) AS month, oi.product\_id, p.name AS product\_name, SUM(oi.sale\_price) AS total\_sales:** Dalam syntax Select ini terdapat lima hal yang diambil, berikut penjelasannya:

- Query untuk mengekstrak tahun dari "o.created\_at" (kolom "created\_at" pada tabel "orders") kemudian kolom tersebut diberi alias yaitu "year".
- Query untuk mengekstrak bulan dari "o.created\_at" (kolom "created\_at" pada tabel "orders") kemudian kolom tersebut diberi alias yaitu "month".
- Query untuk mengambil kolom "oi.product\_id" (kolom "product\_id" pada tabel "order\_items").
- Query untuk mengambil kolom "oi.product\_id" (kolom "product\_name" pada tabel "order\_items") kemudian kolom tersebut diberi alias yaitu "product\_name".
- Query untuk menjumlahkan "oi.sale\_price" (kolom "sale\_price" pada tabel "order\_items") kemudian kolom tersebut diberi alias yaitu "total\_sales".

**FROM `bigquery-public-data.thelook\_ecommerce.orders` o:** Query yang mengakses data dari tabel "orders" dalam dataset "thelook\_ecommerce" bagian dari "bigquery-public-data". Data yang diambil ini diberi alias "o".

**JOIN `bigquery-public-data.thelook\_ecommerce.order\_items` oi ON o.order\_id = oi.order\_id JOIN `bigquery-public-data.thelook\_ecommerce.products` p ON oi.product\_id = p.id:** Query untuk menggabungkan 3 tabel ("orders", "order\_items", dan "products") yang ada dalam dataset "thelook\_ecommerce" bagian dari "bigquery-public-data" dengan pemberian alias yaitu data dari tabel "orders" dijuluki "o", data tabel "order\_items" dijuluki "oi", dan data tabel "products" dijuluki "p". Penggabungan tiga tabel ini didasarkan pada foreign key yang relevan (kolom penghubung atau yang sama dengan kolom di tabel lain) yaitu untuk tabel "orders" dengan "order\_items" berdasarkan kolom "order\_id" dan "order\_items" dengan "products" berdasarkan kolom "product\_id".

**GROUP BY year, month, oi.product\_id, p.name):** Query untuk mengelompokkan data berdasarkan kolom "year", "month", "oi.product\_id" (kolom "product\_id" dari tabel "order\_items"), "p.name" (kolom "name" dari "tabel product").

c. CTE tabel ranked\_sales

, **ranked\_sales AS (...)**: diawali dengan tanda koma yang sebelumnya ada syntax WITH yang berarti query ini juga merupakan CTE. CTE ini digunakan untuk membuat tabel sementara bernama "ranked\_sales".

**SELECT year, month, product\_id, product\_name, total\_sales, RANK() OVER (PARTITION BY year, month ORDER BY total\_sales DESC) AS rank**: dalam query ini terdapat enam hal yang diambil sebagai kolom yaitu "year", "month", "product\_id", "product\_name", "total\_sales", dan untuk kolom yang keenam berikut penjelasan lebih rincinya: RANK() digunakan untuk memberikan peringkat pada produk berdasarkan penjualan tertinggi setiap bulan, OVER digunakan untuk menentukan bagaimana proses ranking dilakukan. PARTITION BY mengelompokkan data berdasarkan kolom "year" dan "month", kemudian ORDER BY "total\_sales" berarti syntax untuk mengelompokkan data berdasarkan kolom "total\_sales" dan syntax DESC berarti perintah untuk mengurutkan dari tertinggi ke yang terendah. Jadi kesimpulannya untuk kolom keenam yang diambil adalah ranking total penjualan per bulan tiap tahunnya dari yang penjualannya tertinggi ke yang terendah yang diuluki dengan nama kolom "rank".

**FROM monthly\_sales**): Query yang mengambil data dari "tabel monthly\_sales" yang dibuat di CTE sebelumnya.

3. Query untuk mendapatkan data produk yang penjualannya tertinggi setiap bulan

**SELECT year, month, product\_id, product\_name, total\_sales**: Query untuk memilih kolom "year", "month", "product\_id", "product\_name", dan "total\_sales".

**FROM ranked\_sales**: Query yang mengambil data dari tabel "ranked\_sales".

**WHERE rank = 1**: Query yang menyaring data dengan ketentuan "rank" nya harus 1 yaitu berarti hanya untuk produk dengan penjualan tertinggi.

**ORDER BY year, month**; : Hasil akhirnya diurutkan berdasarkan kolom "year" dan "month" dari terlama ke yang terbaru karena secara default diurutkan secara ASC.