

DOKUMEN QUERY ANALYZING ECOMMERCE BUSINESS PERFORMANCE WITH SQL

Disusun Oleh :

Ammar Khaq Baasir

Linkedin : <https://www.linkedin.com/in/ammар-baasir-973873234/>

Lihat & Download File Format SQL :

[https://drive.google.com/file/d/1bFhxOQ5sOdBvrUtu7mMB230jRxNdZ5ds/
view?usp=sharing](https://drive.google.com/file/d/1bFhxOQ5sOdBvrUtu7mMB230jRxNdZ5ds/view?usp=sharing)

DAFTAR ISI

Task 1 :

Data Preprocessing

Task 2 :

Annual Customer Activity Growth Analysis

Task 3 :

Annual Product Category Quality Analysis

Task 4 :

Analysis of Annual Payment Type Usage

TASK 1

Part 1 :

Membuat sebuah database baru beserta tabel – tabelnya dan menentukan jenis tipe data yang digunakan pada setiap kolomnya.

- Query Pembuatan Database :

```
CREATE DATABASE Mini_project1;
```

- Query Pembuatan Tabel, Mendefinisikan Tipe Data, dan Memberikan Dokumentasi berupa keterangan setiap kolom pada setiap tabel.

```
CREATE TABLE IF NOT EXISTS public.customers_dataset
(
    customer_id character varying(36) COLLATE pg_catalog."default" NOT NULL,
    customer_unique_id character varying(36) COLLATE pg_catalog."default" NOT NULL,
    customer_zip_code_prefix numeric(6, 0) NOT NULL,
    customer_city character varying(50) COLLATE pg_catalog."default" NOT NULL,
    customer_state character varying(5) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT customers_dataset_pkey PRIMARY KEY (customer_id),
    CONSTRAINT customers_dataset_customer_id_customer_id1_key UNIQUE (customer_id)
    INCLUDE(customer_id)
);
```

```
COMMENT ON TABLE public.customers_dataset
    IS 'tabel yang menyimpan data dari customers :
- customer_id : id dari customers,
- customer_unique_id : unique id dari customers,
- customer_zip_code_prefix : kode pos tempat tinggal dari customers,
- customer_city : nama kota dari customers,
- customer_state : nama state (kode negara bagian) tempat tinggal dari customers: ';
```

```
CREATE TABLE IF NOT EXISTS public.geolocations_dataset
(
    geolocation_zip_code_prefix numeric(6, 0) NOT NULL,
    geolocation_lat numeric(20, 0) NOT NULL,
    "geolocation_lng" numeric(20, 0) NOT NULL,
    geolocation_city character varying(50) COLLATE pg_catalog."default" NOT NULL,
    geolocation_state character varying(6) COLLATE pg_catalog."default" NOT NULL
);
```

```
COMMENT ON TABLE public.geolocations_dataset
    IS 'tabel yang menyimpan data geolocations :
- geolocation_zip_code_prefix : menunjukkan kode pos dari lokasi geografis,
- geolocation_lat : menunjukkan posisi lintang(latitude) dari lokasi geografis,
- geolocation_lng : menunjukkan posisi bujur(longitude) dari lokasi geografis,
```

- geolocation_city : menunjukkan nama kota dari lokasi geografis,
- geolocation_state : menunjukkan nama (kode nama) dari state (negara bagian)';

```
CREATE TABLE IF NOT EXISTS public.order_items_dataset
```

```
(
  order_id character varying(36) COLLATE pg_catalog."default" NOT NULL,
  order_item_id numeric(3, 0) NOT NULL,
  product_id character varying(36) COLLATE pg_catalog."default" NOT NULL,
  seller_id character varying(36) COLLATE pg_catalog."default" NOT NULL,
  shipping_limit_date timestamp with time zone NOT NULL,
  price double precision NOT NULL,
  freight_value double precision NOT NULL
);
```

```
COMMENT ON TABLE public.order_items_dataset
```

```
IS 'tabel ini menyimpan data order atau transaksi dari customers :
```

- order_id : menunjukkan id order,
- order_item_id : menunjukkan banyaknya item yang di order,
- product_id : menunjukkan id product,
- seller_id : menunjukkan id sellers,
- shipping_limit_date : menunjukkan batas waktu pengiriman product (pesanan) dimana batasan waktu yang diberikan kepada sellers yang harus menyerahkan product pada ekspedisi pengiriman,
- price : menunjukkan harga dari product,
- freight_value : menunjukkan biaya pengiriman (ongkos kirim)';

```
CREATE TABLE IF NOT EXISTS public.orders_dataset
```

```
(
  order_id character varying(36) COLLATE pg_catalog."default" NOT NULL,
  customer_id character varying(36) COLLATE pg_catalog."default" NOT NULL,
  order_status character varying(12) COLLATE pg_catalog."default" NOT NULL,
  order_purchase_timestamp timestamp with time zone NOT NULL,
  order_approved_at timestamp with time zone,
  order_delivered_carrier_date timestamp with time zone,
  order_delivered_customer_date timestamp with time zone,
  order_estimated_delivery_date timestamp with time zone,
  CONSTRAINT orders_dataset_pkey PRIMARY KEY (order_id),
  CONSTRAINT orders_dataset_order_id_order_id1_key UNIQUE (order_id)
  INCLUDE(order_id)
);
```

```
COMMENT ON TABLE public.orders_dataset
```

```
IS 'tabel ini berisikan data order dari customers :
```

- order_id : menunjukkan id order,
- customer_id : menunjukkan id customer,
- order_status : menunjukkan status dari order,
- order_purchase_timestamp : menunjukkan waktu pertama kali customer melakukan order,
- order_approved_at : menunjukkan waktu pesanan dari customer disetujui oleh sistem,
- order_delivered_carrier_date : menunjukkan waktu dimana pesanan customer diterima ekspedisi pengiriman.
- order_delivered_customer_date : menunjukkan waktu dimana pesanan telah diterima customer (pihak ekspedisi pengiriman menyerahkan kepada customer).

order_estimated_delivery_date : menunjukkan waktu dimana estimasi awal pesanan yang diterima customer (lama waktu dari customer memesan hingga customer menerima pesanannya)';

```
CREATE TABLE IF NOT EXISTS public.payments_dataset
```

```
(
  order_id character varying(36) COLLATE pg_catalog."default" NOT NULL,
  payment_sequential numeric(3, 0) NOT NULL,
  payment_type character varying(15) COLLATE pg_catalog."default" NOT NULL,
  payment_installments numeric(2, 0) NOT NULL,
  payment_value numeric(5, 0) NOT NULL
);
```

```
COMMENT ON TABLE public.payments_dataset
```

IS 'tabel ini berisikan data payments dari customers :

- order_id : menunjukkan id order,
- payment_sequential : menunjukkan jumlah atau banyaknya pembayaran yang dilakukan dalam satu pesanan,
- payment_type : menunjukkan jenis tipe pembayaran yang digunakan,
- payment_installments : menunjukkan banyaknya (berapa kali) pembayaran yang dilakukan ini dapat berisi banyaknya jumlah angsuran,
- payment_value : menunjukkan jumlah besaran pembayaran yang dilakukan';

```
CREATE TABLE IF NOT EXISTS public.products_dataset
```

```
(
  product_id character varying(36) COLLATE pg_catalog."default" NOT NULL,
  product_category_name character varying(50) COLLATE pg_catalog."default",
  product_name_lenght double precision,
  product_description_lenght double precision,
  product_photos_qty double precision,
  product_weight_g double precision,
  product_length_cm double precision,
  product_height_cm double precision,
  product_width_cm double precision,
  CONSTRAINT products_dataset_pkey PRIMARY KEY (product_id),
  CONSTRAINT products_dataset_product_id_product_id1_key UNIQUE (product_id)
  INCLUDE(product_id)
);
```

```
COMMENT ON TABLE public.products_dataset
```

IS 'tabel ini menyimpan data dari jenis - jenis product :

- product_id : menunjukkan id product
- product_category_name : menunjukkan nama kategori dari product,
- product_name_lenght : menunjukkan panjang nama product,
- product_description_lenght : menunjukkan panjang deskripsi dari product,
- product_photos_qty : menunjukkan gambar quantity dari product
- product_weight_g : menunjukkan ukuran berat product berdasarkan gram
- product_length_cm : menunjukkan ukuran panjang product berdasarkan cm
- product_height_cm : menunjukkan ukuran tinggi product berdasarkan cm
- product_width_cm : menunjukkan ukuran lebar product berdasarkan cm';

```
CREATE TABLE IF NOT EXISTS public.reviews_dataset
```

```
(
  review_id character varying(36) COLLATE pg_catalog."default" NOT NULL,
```

```

order_id character varying(36) COLLATE pg_catalog."default" NOT NULL,
review_score numeric(1, 0) NOT NULL,
review_comment_title character varying(50) COLLATE pg_catalog."default",
review_comment_message character varying(350) COLLATE pg_catalog."default",
review_creation_date timestamp with time zone NOT NULL,
review_answer_timestamp timestamp with time zone NOT NULL
);

COMMENT ON TABLE public.reviews_dataset
    IS 'tabel ini menyimpan data reviews dari customers :
- review_id : menunjukkan id review,
- order_id : menunjukkan id order,
- review_score : menunjukkan nilai review dari customer,
- review_comment_title : menunjukkan judul review dari customer
- review_comment_message : menunjukkan isi review dari customer
- review_creation_date : menunjukkan waktu review yang dibuat atau diberikan oleh customer
- review_answer_timestamp : menunjukkan waktu dari jawaban (feedback) yang diberikan atas review dari customer';

CREATE TABLE IF NOT EXISTS public.sellers_dataset
(
    seller_id character varying(36) COLLATE pg_catalog."default" NOT NULL,
    seller_zip_code_prefix numeric(6, 0) NOT NULL,
    seller_city character varying(50) COLLATE pg_catalog."default" NOT NULL,
    seller_state character varying(5) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT sellers_dataset_pkey PRIMARY KEY (seller_id),
    CONSTRAINT sellers_dataset_seller_id_seller_id1_key UNIQUE (seller_id)
        INCLUDE(seller_id)
);

COMMENT ON TABLE public.sellers_dataset
    IS 'tabel yang menyimpan data dari sellers :
- seller_id : menunjukkan id unique dari sellers,
- seller_zip_code_prefix : menunjukkan kode pos tempat tinggal sellers,
- seller_city : menunjukkan nama kota tempat tinggal sellers,
- seller_state : menunjukkan nama state (kode negara bagian) tempat tinggal sellers';

ALTER TABLE IF EXISTS public.order_items_dataset
    ADD CONSTRAINT order_items_dataset_order_id_fkey FOREIGN KEY (order_id)
        REFERENCES public.orders_dataset (order_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID;

ALTER TABLE IF EXISTS public.order_items_dataset
    ADD CONSTRAINT order_items_dataset_product_id_fkey FOREIGN KEY (product_id)
        REFERENCES public.products_dataset (product_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID;

ALTER TABLE IF EXISTS public.order_items_dataset
    ADD CONSTRAINT order_items_dataset_seller_id_fkey FOREIGN KEY (seller_id)

```

```
REFERENCES public.sellers_dataset (seller_id) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
NOT VALID;
```

```
ALTER TABLE IF EXISTS public.orders_dataset
ADD CONSTRAINT orders_dataset_customer_id_fkey FOREIGN KEY (customer_id)
REFERENCES public.customers_dataset (customer_id) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
NOT VALID;
```

```
ALTER TABLE IF EXISTS public.payments_dataset
ADD CONSTRAINT payments_dataset_order_id_fkey FOREIGN KEY (order_id)
REFERENCES public.orders_dataset (order_id) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
NOT VALID;
```

END;

Pada **database** Mini_project1 ini terdiri atas **8** tabel yang telah didefinisikan pada query diatas. **8** tabel yang dibuat menyimpan informasi yang berbeda - beda, yakni sebagai berikut :

- **customers_dataset :**

tabel ini menyimpan informasi data - data dari customers yang melakukan pembelian pada ecommerce. Data - data customers ini berupa Alamat, customers id, city, dll.

- **geolocations_dataset :**

tabel yang menyimpan informasi data geolocation dari customers maupun sellers. Data geolocation berupa garis lintang (latitude, longitude).

- **order_items_dataset :**

tabel yang menyimpan informasi mengenai data product yang dibeli oleh customers. Data yang tersimpan pada tabel ini berupa data harga product, ongkos kirim, dll.

- **orders_dataset :**

tabel yang menyimpan informasi mengenai informasi pesanan dari customers. Data yang tersimpan pada tabel ini berupa data waktu customers orders, status orders, hingga waktu pesanan customers tiba (diterima).

- **payments_dataset :**

tabel ini menyimpan informasi mengenai tipe - tipe pembayaran yang digunakan oleh customers saat melakukan pembelian.

- **products_dataset :**

tabel ini menyimpan informasi mengenai detail product yang dibeli oleh customers. Ini berupa nama product, category product, dll.

- **reviews_dataset :**

tabel ini menyimpan informasi mengenai review yang diberikan oleh customer setelah mereka membeli product dari ecommerce.

- **sellers_dataset :**

tabel ini menyimpan detail informasi mengenai seller yang menjual productnya pada ecommerce.

TASK 2

Pada Task 2 ini, kami ingin mencari informasi mengenai Annual Customer Activity Growth Analysis. Ini merupakan informasi mengenai perkembangan jumlah customer pada layanan ecommerce setiap tahunnya. Analisis ini membuat beberapa poin yakni :

- Rata – Rata jumlah customers aktif bulanan (Monthly Active User) untuk setiap tahunnya.
- Total Customers baru per tahun.
- Jumlah customers yang melakukan repeat order per tahun.
- Rata – rata frequency order untuk setiap tahun.

Pada task ini juga, didefinisikan sebuah CTE (tabel sementara) yang digunakan sebagai tabel utama dalam melakukan analisis ecommerce, tabel ini akan selalu digunakan pada setiap tasknya. Oleh karena itu penggunaan CTE sangat diperlukan, berikut alasan penggunaan CTE :

1. Kode Mudah Dibaca.
2. Kemudahan Dalam Pengelompokan Data.
3. Penggunaan Kode Berulang – Ulang.
4. Optimasi Kinerja

Berikut ini merupakan tabel CTE (master_table) yang akan digunakan pada task ini dan task berikutnya :

```
WITH master_table AS(  
    SELECT  
        database_customers.customer_id,  
        database_customers.customer_unique_id,  
        database_order.order_id,  
        database_order.order_status,  
        database_order.order_purchase_timestamp AS tanggal_order,  
        database_order_item.order_item_id AS jumlah_order,  
        database_order_item.price,  
        database_order_item.freight_value,  
        database_product.product_id,  
        database_product.product_category_name  
    FROM  
        public.customers_dataset AS database_customers  
    JOIN  
        public.orders_dataset AS database_order  
    ON  
        database_order.customer_id = database_customers.customer_id  
    JOIN  
        public.order_items_dataset AS database_order_item  
    ON  
        database_order_item.order_id = database_order.order_id  
    JOIN
```



```

        public.products_dataset AS database_product
    ON
        database_product.product_id = database_order_item.product_id
),

```

Part 1 :

Part ini akan mencari nilai dari rata – rata customers aktif bulanan untuk setiap tahunnya, informasi ini digunakan untuk melihat seberapa baik dan banyak user yang aktif pada setiap bulannya. Semakin banyak customers aktif maka ini menandakan bahwa Perusahaan memiliki prospek yang baik terhadap pendapatan Perusahaan, dimana Perusahaan memiliki kolam atau pelanggan yang berpeluang melakukan transaksi pada layanan ecommerce kami.

Berikut merupakan query untuk mencari rata – rata customers aktif bulanan pada setiap tahunnya (MAU) :

```

monthly_active_user AS(
    SELECT
        EXTRACT(YEAR FROM tanggal_order) AS tahun,
        COUNT(DISTINCT customer_id) AS monthly_active_user
    FROM
        master_table
    GROUP BY
        1
),

```

Jumlah Monthly Active User merujuk pada kolom customer_id, kolom ini menyimpan informasi id dari customer tetapi tidak bersifat unik. Maka dari itu digunakan sebuah fungsi DISTINCT untuk mendapatkan customer_id yang bersifat unik.

Part 2 :

Pada part 2, kami ingin mengetahui seberapa banyak user baru yang join pada layanan ecommerce kami, informasi ini akan digunakan untuk mengetahui seberapa besar progress perkembangan Perusahaan dalam memperluas pasar dengan cara mengukur banyaknya customer yang join pada setiap tahunnya. Semakin banyak customer yang melakukan join pada setiap tahunnya, maka semakin baik kondisi perusahaan dalam memperluas atau menjangkau pelanggan – pelanggan dan ini menandakan perusahaan memiliki pertumbuhan yang positif pada setiap tahunnya.

Berikut merupakan query untuk mencari informasi mengenai total customer baru pada setiap tahunnya :

```

new_customers AS(
    SELECT
        EXTRACT(YEAR FROM tanggal_order) AS tahun,
        COUNT(DISTINCT(customer_unique_id)) AS new_customers
    FROM
        master_table
    GROUP BY
        1
),

```

Jumlah new customer merujuk pada kolom customer_unique_id, kolom ini menyimpan informasi id dari customer yang bersifat unik. Pada part ini tetap menggunakan teknik CTE untuk membuat sebuah tabel karena CTE ini bersifat sementara atau tidak perlu dilakukan pembuatan tabel permanen.

Part 3 :

Part 3 merupakan part yang akan mencari informasi mengenai total customer yang melakukan repeat order pada setiap tahunnya. Informasi repeat order ini merupakan parameter yang digunakan untuk mengukur seberapa baik layanan dari ecommerce kami terhadap customer – customer. Semakin banyak total customer yang melakukan repeat order pertahunnya, maka ini menandakan semakin banyak pula customer yang puas dan senang terhadap layanan dari ecommerce kami. Selain itu, repeat order menandakan banyak customer tersebut percaya atau yakin terhadap Perusahaan kami yang dibuktikan bahwa customer tersebut melakukan beberapa order tanpa merasa cemas barang yang mereka pesan hilang atau tidak sesuai.

Berikut merupakan query untuk mencari jumlah customer yang melakukan repeat order per tahunnya pada layanan ecommerce kami :

```
repeat_order AS(
  SELECT
    EXTRACT(YEAR FROM tanggal_order) AS tahun,
    COUNT(DISTINCT customer_id) - COUNT(DISTINCT customer_unique_id) AS
total_repeat_order
  FROM
    master_table
  GROUP BY
    1
),
```

Total repeat order didapatkan dari perhitungan antara kolom customer_id (id customer) dikurangi dengan dengan kolom customer_unique_id.

Pada part ini tetap menggunakan teknik CTE untuk membuat sebuah tabel karena CTE ini bersifat sementara atau tidak perlu dilakukan pembuatan tabel permanen.

Part 4 :

Pada part ini, akan dicari sebuah informasi mengenai rata – rata frequency order pertahunnya. Informasi ini digunakan untuk mengetahui apakah terjadi penurunan rata – rata penjualan pada setiap tahunnya.

Berikut merupakan query untuk mencari informasi mengenai frequency order pertahunnya :

```
frequency_order AS(
  SELECT
    EXTRACT(YEAR FROM tanggal_order) AS tahun,
    ROUND(AVG(jumlah_order), 3) AS average_freq_order
  FROM
    master_table
  GROUP BY
    1
```

)

Pada part ini tetap menggunakan teknik CTE untuk membuat sebuah tabel karena CTE ini bersifat sementara atau tidak perlu dilakukan pembuatan tabel permanen.

Part 5 :

Pada part ini, kami menggabungkan setiap part dari part 1 hingga part 4, ini dilakukan untuk mendapatkan informasi berupa satu buah tabel akhir yang menyimpan informasi dari part 1 hingga part 4. Di part ini kami akan mendapatkan informasi atau tabel utama berupa Annual Customer Activity Growth Analysis..

Berikut query untuk mendapatkan informasi mengenai Annual Customer Activity Growth Analysis :

```
SELECT
    mau.tahun,
    mau.monthly_active_user,
    nc.new_customers,
    ro.total_repeat_order,
    fo.average_freq_order
FROM
    monthly_active_user mau
JOIN
    new_customers nc ON mau.tahun = nc.tahun
JOIN
    repeat_order ro ON nc.tahun = ro.tahun
JOIN
    frequency_order fo ON ro.tahun = fo.tahun
```

Pada part ini, untuk mendapatkan tabel utama Annual Customer Activity Growth Analysis. Maka harus dilakukan join dari part 1 hingga part 4.

TASK 3

Part 1 :

Pada bagian ini kita akan menggunakan tabel utama berupa Tabel CTE (master_table) yang sama dengan task sebelumnya. Tabel CTE ini digunakan sebagai rujukan awal dalam analisis **Annual Product Category Quality Analysis**.

Untuk part pertama akan dilakukan analisis berupa total revenue yang didapatkan oleh Perusahaan setiap tahunnya. Revenue merujuk pada harga barang (price) + harga ongkir barang (freight_value).

Berikut merupakan query total revenue perusahaan per tahun :

```
revenue_table AS(  
  SELECT  
    EXTRACT(YEAR FROM tanggal_order) AS tahun,  
    ROUND(SUM(CAST((price + freight_value) AS numeric)), 2) AS total_revenue_year  
  FROM  
    master_table  
  WHERE  
    order_status = 'delivered'  
  GROUP BY  
    1  
)
```

Query yang dibuat menggunakan teknik CTE, dimana teknik ini digunakan untuk memudahkan proses penggunaan berulang – ulang sebuah query. Selain itu menggunakan CTE ini akan menghemat ruang penyimpanan database dari Perusahaan karena tabel CTE ini bersifat sementara atau tidak perlu dilakukan pembuatan tabel permanen.

Pada kasus ini terdapat klausa berupa order_status = delivered, karena kami beropini bahwa revenue suatu Perusahaan didapatkan ketika suatu produk tersebut berhasil terjual sepenuhnya tanpa ada return ataupun canceled, sebaliknya apabila terdapat canceled maka ini merupakan sebuah kerugian bagi perusahaan.

Part 2 :

Pada part ini, kami akan menghitung berapa banyak total barang yang mengalami canceled order pada setiap tahunnya. Perhitungan ini digunakan untuk mengetahui berapa banyak barang yang menyebabkan kerugian bagi perusahaan dan dapat menjadi acuan estimasi banyaknya customers yang mengalami experience yang kurang baik ketika melakukan pembelian pada Perusahaan kami.

Berikut merupakan query untuk menghitung total barang yang mengalami canceled order per tahun :

```
canceled_table AS(
```

```

SELECT
    EXTRACT(YEAR FROM tanggal_order) AS tahun,
    SUM(jumlah_order) AS total_canceled_year
FROM
    master_table
WHERE
    order_status = 'canceled'
GROUP BY
    1
ORDER BY
    1 DESC
),

```

Total canceled didapatkan dari kolom jumlah_order, dimana kolom ini berisi total produk yang dibeli oleh customer dalam 1 kali transaksi. Pada kasus ini terdapat klausa berupa order_status = canceled, klausa ini digunakan karena klausa ini menunjukkan bahwa product tersebut mengalami canceled order oleh customer.

Pada part ini tetap menggunakan teknik CTE untuk membuat sebuah tabel karena CTE ini bersifat sementara atau tidak perlu dilakukan pembuatan tabel permanen.

Part 3 :

Pada part ini, dilakukan sebuah analisis terhadap top category product yang memberikan revenue paling tinggi untuk perusahaan. Analisis ini dilakukan guna mendapatkan informasi tentang category product apa yang sangat banyak memberikan pendapatan perusahaan, category product tersebut akan dijadikan sebuah product unggulan bagi Perusahaan kedepannya dan product tersebut akan dijadikan sebagai product senjata utama untuk mendapatkan pendapatan yang lebih tinggi pada tahun – tahun berikutnya.

Berikut merupakan query untuk menemukan top product category yang memberikan revenue tertinggi bagi Perusahaan setiap tahunnya :

```

product_top_revenue AS(
    SELECT
        EXTRACT(YEAR FROM tanggal_order) AS tahun,
        product_category_name AS product_category_revenue,
        ROUND(SUM(CAST((price + freight_value) AS numeric)), 2) AS total_revenue_product,
        RANK() OVER(PARTITION BY
            EXTRACT(YEAR FROM tanggal_order)
            ORDER BY ROUND(SUM(CAST((price + freight_value) AS numeric)), 2) DESC)
AS top_rank
    FROM
        master_table
    WHERE
        order_status = 'delivered' AND product_category_name IS NOT NULL
    GROUP BY
        1, 2
    ORDER BY
        1, 4 ASC
),

```

Total revenue yang digunakan pada analisis ini merujuk pada harga product (price) + harga ongkir product (freight_value). Setelah didapatkan total revenue setiap product category, maka akan diambil product category yang memiliki urutan teratas (rank 1) pada setiap tahunnya. Pada kasus ini terdapat klausa berupa order_status = delivered, karena kami beropini bahwa revenue didapatkan dari hasil keberhasilan transaksi customer atau hingga customer mendapatkan barang yang mereka pesan. Selain itu, terdapat klausa IS NOT NULL ini digunakan untuk mengembalikan product category name yang terdapat value (nama) pada setiap barisnya atau kami menolak untuk mendapatkan revenue Perusahaan tetapi tidak diketahui product category namanya.

Pada part ini tetap menggunakan teknik CTE untuk membuat sebuah tabel karena CTE ini bersifat sementara atau tidak perlu dilakukan pembuatan tabel permanen.

Part 4 :

Pada part ini, kami melakukan analisis mengenai top product category name yang paling banyak mengalami canceled order pada setiap tahunnya. Analisis ini dilakukan untuk mengetahui product apa saja yang menjadi penyebab utama perusahaan mengalami kerugian pada setiap tahunnya. Informasi category product yang didapatkan akan dijadikan sebagai bahan evaluasi untuk tahun – tahun berikutnya agar product tersebut tidak selalu mengalami canceled order paling banyak.

Berikut merupakan query untuk mencari top category product name yang mengalami canceled paling banyak setiap tahunnya :

```
product_top_canceled AS(
  SELECT
    EXTRACT(YEAR FROM tanggal_order) AS tahun,
    product_category_name AS product_category_canceled,
    SUM(jumlah_order) AS total_canceled_product,
    RANK() OVER(PARTITION BY EXTRACT(YEAR FROM tanggal_order)
                ORDER BY SUM(jumlah_order) DESC) AS top_rank
  FROM
    master_table
  WHERE
    order_status = 'canceled'
  GROUP BY
    1, 2
  ORDER BY
    1, 4
)
```

Total canceled product didapatkan dari hasil perhitungan kolom jumlah order, kolom ini berisikan total produk yang dibeli oleh customer pada saat 1 kali transaksi. Setelah didapatkan total canceled setiap product category, maka akan diambil product category name yang memiliki urutan teratas (rank 1) pada setiap tahunnya. Pada part ini menggunakan klausa order_status = canceled karena klausa ini menunjukkan bahwa product tersebut mengalami canceled order oleh customer

Pada part ini tetap menggunakan teknik CTE untuk membuat sebuah tabel karena CTE ini bersifat sementara atau tidak perlu dilakukan pembuatan tabel permanen.

Part 5 :

Pada part ini, kami menggabungkan setiap part dari part 1 hingga part 4, ini dilakukan untuk mendapatkan informasi berupa satu buah tabel akhir yang menyimpan informasi dari part 1 hingga part 4. Di part ini kami akan mendapatkan informasi atau tabel utama berupa Annual Product Category Quality Analysis.

Berikut ini merupakan query untuk mendapatkan tabel Annual Category Product Quality Analysis pertahunnya :

```
SELECT
    rt.tahun,
    ptr.product_category_revenue,
    ptr.total_revenue_product,
    rt.total_revenue_year,
    ptc.product_category_canceled,
    ptc.total_canceled_product,
    ct.total_canceled_year
FROM
    revenue_table rt
JOIN
    canceled_table ct ON rt.tahun = ct.tahun
JOIN
    product_top_revenue ptr ON ct.tahun = ptr.tahun
JOIN
    product_top_canceled ptc ON ptr.tahun = ptc.tahun
WHERE
    ptr.top_rank = 1 AND ptc.top_rank = 1
ORDER BY
    1
```

Pada part ini, untuk mendapatkan tabel utama Annual Category Product Quality Analysis. Maka harus dilakukan join dari part 1 hingga part 4.

TASK 4

Part 1 :

Pada part ini, kami melakukan sebuah analisis tentang type pembayaran yang sering digunakan oleh customer pada saat melakukan transaksi. Analisis ini digunakan untuk mendapatkan informasi tentang payment type usage, informasi ini dapat digunakan untuk memberikan peningkatan penjualan pada perusahaan dimana kami dapat memberikan promo kepada customer apabila melakukan transaksi dengan metode tertentu.

Berikut ini merupakan query untuk mendapatkan informasi mengenai Analysis of Annual Payment Type Usage per tahun :

```
WITH payments_fav AS(  
  SELECT  
    EXTRACT(YEAR FROM od.order_purchase_timestamp) AS tahun,  
    pd.payment_type,  
    COUNT(2) AS total_used  
  FROM  
    payments_dataset pd  
  JOIN  
    orders_dataset od ON pd.order_id = od.order_id  
  GROUP BY  
    1, 2  
  ORDER BY  
    1, 3 DESC  
)
```

Query diatas merupakan query untuk pembuatan tabel sementara (CTE), tabel ini digunakan untuk rujukan analisis payment type usage. Pada query diatas dilakukan join antara dua tabel yakni tabel orders_dataset untuk mengambil nilai pada kolom order_purchase_timestamp (informasi tanggal transaksi) dan tabel payments_dataset untuk mengambil nilai payment_type (tipe pembayaran yang digunakan customer).

Pada part ini tetap menggunakan teknik CTE untuk membuat sebuah tabel karena CTE ini bersifat sementara atau tidak perlu dilakukan pembuatan tabel permanen.

Part 2 :

Pada part 2, kami melakukan query untuk mendapatkan informasi berupa sebuah tabel pivot yang berisi nilai tipe pembayaran yang digunakan customer pada setiap tahunnya.

Berikut merupakan query untuk mendapatkan tabel Annual Payment Type Usage pertahun :

```
SELECT  
  payment_type,  
  SUM(CASE WHEN tahun = 2016 THEN total_used ELSE 0 END) AS tahun_2016,
```



```
SUM(CASE WHEN tahun = 2017 THEN total_used ELSE 0 END) AS tahun_2017,  
SUM(CASE WHEN tahun = 2018 THEN total_used ELSE 0 END) AS tahun_2018  
FROM  
    payments_fav  
GROUP BY  
    1
```