SINTAX PEMBUATAN TABEL

```
----- product_dataset
CREATE TABLE product_dataset (
        product_id varchar PRIMARY KEY not null,
        product_category_name varchar ,
        product_name_lenght float8,
        product_description_length float8,
        product_photos_qty float8,
        product_weight_g float8,
        product_length_cm float8 ,
        product_height_cm float8 ,
        product_width_cm float8
);

-----customers_dataset
CREATE TABLE customers_dataset (
        customer_id varchar PRIMARY KEY not null,
        customer_unique_id varchar ,
        customer_zip_code_prefix int ,
        customer_city VARCHAR ,
        customer_state VARCHAR ,

);

-----seller_dataset
CREATE TABLE customers_dataset (
        seller_id varchar PRIMARY KEY not null,
        seller_zip_code_prefix int ,
        seller_city VARCHAR ,
        seller_state VARCHAR ,
);

-----geolocation_dataset
CREATE TABLE geolocation_dataset (
        geolocation_zip_code_prefix VARCHAR,
        geolocation_lat NUMERIC ,
        geolocation_lng NUMERIC,
        geolocation_city VARCHAR,
        geolocation_state VARCHAR
);
```

```
----orders_dataset
CREATE TABLE orders_dataset (
        order_id varchar PRIMARY KEY not null,
        customer_id varchar,
        order_status varchar,
        order_purchase_timestamp TIMESTAMP,
        order_approved_at TIMESTAMP,
        order_delivered_carrier_date TIMESTAMP,
        order_delivered_customer_date TIMESTAMP,
        order_estimated_delivery_date TIMESTAMP
);

----order_items_dataset
CREATE TABLE order_items_dataset (
        order_id varchar ,
        order_item_id int ,
        product_id varchar,
        seller_id varchar,
        shipping_limit_date TIMESTAMP,
        price numeric,
        freight_value numeric
);

-----order_payments_dataset
CREATE TABLE order_payments_dataset (
        order_id varchar ,
        payment_sequential int ,
        payment_type varchar ,
        paymeny_installments int ,
        payment_value float8
);

-----order_review_dataset
CREATE TABLE order_review_dataset (
        review_id varchar,
        order_id varchar,
        review_score int,
        review_comment_title VARCHAR,
        review_comment_message VARCHAR,
        review_creation_date TIMESTAMP,
        review_answer_timestamp timestamp
);
```

SINTAX QUERY TUGAS 2

- - - - Rata-rata Monthly Active User (MAU) per tahun

```
select year, avg(mau) as average_mau
from (
select
        date_part('year', o.order_purchase_timestamp) as years,
        date_part('month', o.order_purchase_timestamp) as months,
        count(distinct c.customer_unique_id) as mau
    from orders_dataset o
    join customer_dataset c on o.customer_id = c.customer_id
    group by 1,2
) temp
group by 1
```

- - - - Total customer baru per tahun

```
Select
        date_part('year', first_purchase) as years,
        count(1) as newcust
from (
        select
        cd.customer_unique_id,
        min(od.order_purchase_timestamp) as first_purchase
        from orders_dataset od
        join customer_dataset cd on cd.customer_id = od.customer_id
        group by 1
) unik
group by 1
order by 1
```

- - - - Jumlah customer yang melakukan repeat order per tahun

```
Select
year,
count(repeat_order) as repeat
from (
        select date_part('year', od.order_purchase_timestamp) as year,
        cd.customer_unique_id,
        count(1) as repeat_order
        from orders_dataset od
        join customer_dataset cd on od.customer_id = cd.customer_id
        group by 1,2
        Having count(1) > 1
) ok
group by 1
order by 1
```

- - - - Rata-rata frekuensi order untuk setiap tahun

```
select
	year,
	round(avg(orderperyears),3) as avg_orderperyears
from (
	select date_part('year', od.order_purchase_timestamp) as year,
	cd.customer_unique_id,
	count(1) as orderperyears
	from orders_dataset od
	join customer_dataset cd on od.customer_id = cd.customer_id
	group by 1,2

) opy
group by 1
```

- - - - Gabungan Seluruh Tabel

```
with
calc_mau as (
select year, avg(mau) as average_mau
from (
select
        date_part('year', o.order_purchase_timestamp) as years,
        date_part('month', o.order_purchase_timestamp) as months,
        count(distinct c.customer_unique_id) as mau
    from orders_dataset o
    join customer_dataset c on o.customer_id = c.customer_id
    group by 1,2
) temp
group by 1
),
calc_newcust as (
Select
    date_part('year', first_purchase) as years,
    count(1) as newcust
from (
    select
    cd.customer_unique_id,
    min(od.order_purchase_timestamp) as first_purchase
    from orders_dataset od
    join customer_dataset cd on cd.customer_id = od.customer_id
    group by 1
```

```
) unik
group by 1
order by 1
),
calc_repeatorder as (
Select
year,
count(repeat_order) as repeat
from (
        select date_part('year', od.order_purchase_timestamp) as year,
        cd.customer_unique_id,
        count(1) as repeat_order
        from orders_dataset od
        join customer_dataset cd on od.customer_id = cd.customer_id
        group by 1,2
        Having count(1) > 1
) ok
group by 1
order by 1
),
calc_frekorder as(
select
        year,
        round(avg(orderperyears),3) as avg_orderperyears
from (
        select date_part('year', od.order_purchase_timestamp) as year,
```

```
        cd.customer_unique_id,
        count(1) as orderperyears
        from orders_dataset od
        join customer_dataset cd on od.customer_id = cd.customer_id
        group by 1,2
) opy
group by 1
)
select distinct
        mau.year,
        mau.avg_mau,
        newc.newcust,
        rep.old_cust,
        fk.avg_od
from calc_mau mau
join calc_newcust newc on mau.year = mau.year
join calc_repeatorder rep on rep.year = mau.year
join calc_frekorder fk on fk.year = mau.year
```

Task 3

– - - - Revenue per Tahun

```
create table total_revenue_per_year as
```

```
select
    date_part('year', od.order_purchase_timestamp) as years,
    sum (rpo) as revenue
from (
    select
        order_id,
    (price + freight_value) as rpo
    from order_items_dataset
    group by 1,2
) subq
join orders_dataset od on subq.order_id = od.order_id
where od.order_status = 'delivered'
group by 1
order by 1
```

- - - - Jumlah Cancel Order Per Tahun

```
create table total_cancel_per_year as
select
```

```
    date_part('year', order_purchase_timestamp) as years,
    count (1) as num_canceled_orders
from orders_dataset
where order_status = 'canceled'
group by 1
order by 1
```

- - - - Top Kategori Yang Menghasilkan Revenue Terbesar Per Tahun

```
create table top_product_category_by_revenue_per_year as
select
```

```
    years,
    product_category_name,
    revenue
from (
    select
        date_part('year', od.order_purchase_timestamp) as years,
        p.product_category_name,
        sum (oid.price + oid.freight_value) as revenue,
        rank()over(partition by date_part('year', od.order_purchase_timestamp)
        order by sum(oid.price + oid.freight_value) desc) as rk
    from order_items_dataset oid
    join orders_dataset od on od.order_id = oid.order_id
    join product_dataset p on p.product_id = oid.product_id
    where od.order_status = 'delivered'
    group by 1,2) sq
where rk = 1
order by 1
```

- - - - Kategori Yang Mengalami Cancel Order Terbanyak Per Tahun

```
create table most_canceled_product_category_per_year as
select
```

```
    year,
    product_category_name,
    num_canceled
from (
    select
        date_part('year', od.order_purchase_timestamp) as year,
        p.product_category_name,
        count (1) as num_canceled,
        rank()over(partition by date_part('year', od.order_purchase_timestamp)
                order by count(1) desc) as rk
    from order_items_dataset oid
    join orders_dataset od on od.order_id = oid.order_id
    join product_dataset p on p.product_id = oid.product_id
    where od.order_status = 'canceled'
    group by 1,2) sq
where rk = 1
order by 1
```

- - - - gabungan tabel

```
select
    a.years,
```

```
    a.product_category_name as top_product_category_by_revenue,
    a.revenue as category_revenue,
    b.revenue as year_total_revenue,
    c.product_category_name as most_canceled_product_category,
    c.num_canceled as category_num_canceled,
    d.num_canceled_orders as year_total_num_canceled
from top_product_category_by_revenue_per_year a
join total_revenue_per_year b on a.years = b.years
join most_canceled_product_category_per_year c on a.years = c.year
join total_cancel_per_year d on d.year = a.years
```

Tugas 4

- - - - jumlah penggunaan masing-masing tipe pembayaran untuk setiap tahun

```
with
tmp as(
select
    date_part('year', od.order_purchase_timestamp) as years,
    opd.payment_type,
    count(1) as num_used
from order_payments_dataset opd
join orders_dataset od on od.order_id = opd.order_id
group by 1,2
)


     select
        payment_type,
        sum(case when years = '2016' then num_used else 0 end) as year_2016,
        sum(case when years = '2017' then num_used else 0 end) as year_2017,
        sum(case when years = '2018' then num_used else 0 end) as year_2018
    from tmp
    group by 1
```