Pembuatan ERD

```
alter table customers_dataset add primary key (customer_id);
alter table orders_dataset add foreign key (customer_id) references customers_dataset;

alter table orders_dataset add primary key (order_id);
alter table reviews_dataset add foreign key (order_id) references orders_dataset;
alter table payments_dataset add foreign key (order_id) references orders_dataset;
alter table order_items_dataset add foreign key (order_id) references orders_dataset;

alter table products_dataset add primary key (product_id);
alter table order_items_dataset add foreign key (product_id) references products_dataset;

alter table sellers_dataset add primary key (seller_id);
alter table order_items_dataset add foreign key (seller_id) references sellers_dataset;

alter table geolocations_dataset add primary key (zip_code_prefix);
alter table customers_dataset add foreign key (zip_code_prefix) references geolocations_dataset;
alter table sellers_dataset add foreign key (zip_code_prefix) references geolocations_dataset;
```

Menghilangkan nilai duplikat pada geolocations_dataset

```
with cte1 as (
   delete
   from geolocations_dataset
   returning *
), cte2 as (
   select
      row_number() over(partition by zip_code_prefix order by zip_code_prefix desc) as rn,
      *
   from cte1
)
insert into geolocations_dataset
select zip_code_prefix, geolocation_lat, geolocation_lng, geolocation_city, geolocation_state
from cte2
where rn = 1
```

Rata-rata Monthly Active User (MAU) per tahun

```
select
year,
round(avg(mau)) as average_mau
from (
            select
            date_part('year', od.order_purchase_timestamp) as year,
            date_part('month', od.order_purchase_timestamp) as month,
            count (DISTINCT cd.customer_unique_id) as mau
            from customers_dataset as cd
            join orders_dataset as od
            on cd.customer_id = od.customer_id
            group by year, month
            ) as subq
group by year
```

Total customer baru per tahun

```
select
subq.year,
count(subq.customer_unique_id) as total_customer_baru
from(
        select
        cd.customer_unique_id,
        min(date_part('year', od.order_purchase_timestamp)) as year
        from orders_dataset as od
        join customers_dataset as cd
        on od.customer_id = cd.customer_id
        group by cd.customer_id
        ) as subq
group by subq.year
order by subq.year
```

Jumlah customer yang melakukan repeat order per tahun

```
select
subq.year,
count(subq.jumlah_order) as total_customers_repeat_order
from(
        select
        cd.customer_unique_id,
        date_part('year', od.order_purchase_timestamp) as year,
        count(od.order_id) as jumlah_order
        from orders_dataset as od
        join customers_dataset as cd
        on od.customer_id = cd.customer_id
        group by cd.customer_unique_id, year
        having count(od.order_id) > 1
        ) as subq
group by year
order by year
```

Rata-rata frekuensi order untuk setiap tahun

```
select
subq.year,
round(avg(subq.jumlah_order)) as ratarata_frekuensi_order
from(
        select
        date_part('year', od.order_purchase_timestamp) as year,
        cd.customer_unique_id,
        count(od.order_id) as jumlah_order
        from orders_dataset as od
        join customers_dataset as cd
        on od.customer_id = cd.customer_id
        group by year, cd.customer_unique_id
        ) as subq
group by subq.year
order by subq.year
```

Penggabungan semua tabel

```
select
        mau.year,
        mau.average_mau,
        cb.total_customer_baru,
        ro.total_customers_repeat_order,
        fo.ratarata_frekuensi_order
from temp_mau as mau
join temp_customerbaru as cb on mau.year = cb.year
join temp_repeatorder as ro on ro.year = mau.year
join temp_frekuensiorder as fo on fo.year = mau.year
```

*Note: Penggabungan semua table dilakukan setelah membuat temporary table dan dalam satu query yang sama

Revenue per tahun

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

Jumlah cancel order per tahun

```
SELECT
date_part('year', order_purchase_timestamp) as year,
count(order_status) as cancel_order
from orders_dataset
where order_status = 'canceled'
group by year
```

Top kategori yang menghasilkan revenue terbesar per tahun

```
select
subq.year,
subq.product_category_name,
subq.jumlah_revenue
from(
        select
        date_part('year', od.order_purchase_timestamp) as year,
        pd.product_category_name,
        sum(oid.price + oid.freight_value) as jumlah_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 as oid
        join orders_dataset as od
        on oid.order_id = od.order_id
        join products_dataset as pd
        on oid.product_id = pd.product_id
        where od.order_status ='delivered'
        group by year, pd.product_category_name
        order by year
        ) as subq
--where subq.year = '2016' --Hapus garis untuk filter tahun 2016
--where subq.year = '2017' --Hapus garis untuk filter tahun 2017
--where subq.year = '2018' --Hapus garis untuk filter tahun 2018
order by subq.jumlah_revenue desc
limit 5
```

```
-- Query Perbandingan Top 1 per tahun
select
subq.year,
subq.product_category_name,
subq.jumlah_revenue
from(
        select
        date_part('year', od.order_purchase_timestamp) as year,
        pd.product_category_name,
        sum(oid.price + oid.freight_value) as jumlah_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 as oid
        join orders_dataset as od
        on oid.order_id = od.order_id
        join products_dataset as pd
        on oid.product_id = pd.product_id
        where od.order_status ='delivered'
        group by year, pd.product_category_name
        order by year
        ) as subq
where rk = 1
```

Kategori yang mengalami cancel order terbanyak per tahun

```
select
subq.year,
subq.product_category_name,
subq.cancel_order
from(
        select
        date_part('year', od.order_purchase_timestamp) as year,
        pd.product_category_name,
        count(1) as cancel_order,
        rank()
        over(partition by date_part('year', od.order_purchase_timestamp)
        order by count(1) desc) as rk
        from order_items_dataset as oid
        join orders_dataset as od
        on oid.order_id = od.order_id
        join products_dataset as pd
        on oid.product_id = pd.product_id
        where od.order_status ='canceled'
        group by year, pd.product_category_name
        order by year
        ) as subq
where rk = 1
```

Penggabungan semua tabel

```
select
        r.year,
        rt.product_category_name as top_produk_revenue,
        rt.jumlah_revenue as category_revenue,
        r.revenue as total_revenue,
        cot.product_category_name as top_cancel_produk,
        cot.cancel_order,
        co.cancel_order as total_cancel_order
from revenue as r
join revenue_terbesar as rt on r.year = rt.year
join cancel_order as co on r.year = co.year
join cancel_order_terbanyak as cot on r.year = cot.year
```

*Note: Penggabungan semua table dilakukan setelah membuat table

Jumlah Penggunaan Masing-Masing Tipe Pembayaran Untuk Setiap Tahun

```
--Mengecek Nilai Yang Kosong atau Null
select payment_type
from payments_dataset
where payment_type is null

--Jumlah Penggunaan Masing-Masing Tipe Pembayaran Berdasarkan tahun
select
pd.payment_type,
date_part('year', od.order_purchase_timestamp) as year,
count(pd.order_id) as jumlah_penggunaan
from payments_dataset as pd
join orders_dataset as od
on pd.order_id = od.order_id
group by pd.payment_type, year
order by pd.payment_type, year

--Jumlah Penggunaan Masing-Masing Tipe Pembayaran
select
payment_type,
count(order_id) as jumlah_penggunaan
from payments_dataset
group by payment_type
order by jumlah_penggunaan desc
```

Penggabungan Tabel Bentuk Pivot Tabel

```
--Membuat Temporary tabel
create temp table tmp as (
select
	date_part('year', od.order_purchase_timestamp) as year,
	pd.payment_type,
	count(pd.order_id) as jumlah_penggunaan
from payments_dataset as pd
join orders_dataset as od
on pd.order_id = od.order_id
group by year, pd.payment_type
),

--Penggabungan Tabel Bentuk Pivot Tabel
select *,
case when tahun_2016 = 0 then NULL
else round((tahun_2017 - tahun_2016) / tahun_2016, 2)
end as kenaikan_persen_2016_2017,
case when tahun_2017 = 0 then NULL
else round((tahun_2018 - tahun_2017) / tahun_2017, 2)
end as kenaikan_persen_2017_2018
from (
	select
	payment_type,
	sum(case when year = '2016' then jumlah_penggunaan else 0 end) as tahun_2016,
	sum(case when year = '2017' then jumlah_penggunaan else 0 end) as tahun_2017,
	sum(case when year = '2018' then jumlah_penggunaan else 0 end) as tahun_2018
	from tmp
	group by payment_type
	) as subq
order by 5 desc
```