**Data Preparation**

1. Create New Dataset

Pertama kita buat terlebih dahulu database ecommerce pada query tool

create database ecommerce;

Selanjutnya kita buat 9 table baru yang akan berada di database ecommerce

create table customer (

customer\_id VARCHAR(250),

customer\_unique\_id VARCHAR(250),

customer\_zip\_code\_prefix INTEGER,

customer\_city VARCHAR(50),

customer\_state VARCHAR(50)

);

create table geolocation (

geo\_zip\_code\_prefix INTEGER,

geo\_lat NUMERIC,

geo\_lng NUMERIC,

geo\_city VARCHAR(250),

geo\_state VARCHAR(250)

);

create table order\_items (

order\_id VARCHAR(250),

order\_item\_id INTEGER,

product\_id VARCHAR(250),

seller\_id VARCHAR(250),

shipping\_limit\_date TIMESTAMP,

price FLOAT,

freight\_value FLOAT

);

CREATE TABLE order\_payments (

order\_id VARCHAR(250),

payment\_sequential INTEGER,

payment\_type VARCHAR(250),

payment\_installments INTEGER,

payment\_value FLOAT

);

CREATE TABLE order\_reviews (

review\_id VARCHAR(250),

order\_id VARCHAR(250),

review\_score INTEGER,

review\_comment\_title VARCHAR(250),

review\_comment\_message TEXT,

review\_creation\_date TIMESTAMP,

review\_answer\_timestamp TIMESTAMP

);

CREATE TABLE orders (

order\_id VARCHAR(250),

customer\_id VARCHAR(250),

order\_status VARCHAR(250),

order\_purchase\_timestamp TIMESTAMP,

order\_approved\_at TIMESTAMP,

order\_delivered\_carrier\_date TIMESTAMP,

order\_delivered\_customer\_date TIMESTAMP,

order\_estimated\_delivery\_date TIMESTAMP

);

CREATE TABLE product (

product\_id VARCHAR(250),

product\_category\_name VARCHAR(250),

product\_name\_lenght NUMERIC,

product\_description\_lenght NUMERIC,

product\_photos\_qty NUMERIC,

product\_weight\_g NUMERIC,

product\_length\_cm NUMERIC,

product\_height\_cm NUMERIC,

product\_width\_cm NUMERIC

);

CREATE TABLE sellers (

seller\_id VARCHAR(250),

seller\_zip\_code\_prefix INTEGER,

seller\_city VARCHAR(250),

seller\_state VARCHAR(250)

);

1. Importing Data CSV to Dataset

Selanjutnya kita akan mengimpor data CSV ke masing-masing 9 Table yang sudah dibuat

COPY customer (

customer\_id,

customer\_unique\_id,

customer\_zip\_code\_prefix,

customer\_city,

customer\_state

)

FROM ‘C:\Users\Azam Fathurrahman\Documents\! Projects\Project SQL\1 Analyzing eCommerce Business Performance with SQL\Dataset\customers\_dataset.csv’

DELIMITER ','

CSV HEADER;

COPY geolocation (

geo\_zip\_code\_prefix,

geo\_lat,

geo\_lng,

geo\_city,

geo\_state

)

FROM "C:\Users\Azam Fathurrahman\Documents\! Projects\Project SQL\1 Analyzing eCommerce Business Performance with SQL\Dataset\geolocation\_dataset.csv"

DELIMITER ','

CSV HEADER;

COPY order\_items (

order\_id,

order\_item\_id,

product\_id,

seller\_id,

shipping\_limit\_date,

price,

freight\_value

)

FROM "C:\Users\Azam Fathurrahman\Documents\! Projects\Project SQL\1 Analyzing eCommerce Business Performance with SQL\Dataset\order\_items\_dataset.csv"

DELIMITER ','

CSV HEADER;

COPY order\_payments (

order\_id,

payment\_sequential,

payment\_type,

payment\_installments,

payment\_value

)

FROM "C:\Users\Azam Fathurrahman\Documents\! Projects\Project SQL\1 Analyzing eCommerce Business Performance with SQL\Dataset\order\_payments\_dataset.csv"

DELIMITER ','

CSV HEADER;

COPY order\_reviews (

review\_id,

order\_id,

review\_score,

review\_comment\_title,

review\_comment\_message,

review\_creation\_date,

review\_answer\_timestamp

)

FROM "C:\Users\Azam Fathurrahman\Documents\! Projects\Project SQL\1 Analyzing eCommerce Business Performance with SQL\Dataset\order\_reviews\_dataset.csv"

DELIMITER ','

CSV HEADER;

COPY orders (

order\_id,

customer\_id,

order\_status,

order\_purchase\_timestamp,

order\_approved\_at,

order\_delivered\_carrier\_date,

order\_delivered\_customer\_date,

order\_estimated\_delivery\_date

)

FROM "C:\Users\Azam Fathurrahman\Documents\! Projects\Project SQL\1 Analyzing eCommerce Business Performance with SQL\Dataset\orders\_dataset.csv"

DELIMITER ','

CSV HEADER;

COPY product (

product\_id,

product\_category\_name,

product\_name\_lenght,

product\_description\_lenght,

product\_photos\_qty,

product\_weight\_g,

product\_length\_cm,

product\_height\_cm,

product\_width\_cm

)

FROM "C:\Users\Azam Fathurrahman\Documents\! Projects\Project SQL\1 Analyzing eCommerce Business Performance with SQL\Dataset\product\_dataset.csv"

DELIMITER ','

CSV HEADER;

COPY sellers (

seller\_id,

seller\_zip\_code\_prefix,

seller\_city,

seller\_state

)

FROM "C:\Users\Azam Fathurrahman\Documents\! Projects\Project SQL\1 Analyzing eCommerce Business Performance with SQL\Dataset\sellers\_dataset.csv"

DELIMITER ','

CSV HEADER;

1. Create Entity Relationship Diagram

Primary Key

alter table product add constraint pk\_product primary key (product\_id);

alter table customer add constraint pk\_customer primary key (customer\_id);

alter table geolocation add constraint pk\_geo primary key (zip\_code\_prefix);

alter table orders add constraint pk\_orders primary key (order\_id);

alter table sellers add constraint pk\_sellers primary key (seller\_id);

Foreign Key

alter table order\_items add foreign key (product\_id) references product;

alter table customer add foreign key (zip\_code\_prefix) references geolocation;

alter table orders add foreign key (customer\_id) references customer;

alter table order\_items add foreign key (order\_id) references orders;

alter table order\_items add foreign key (seller\_id) references sellers;

alter table sellers add foreign key (zip\_code\_prefix) references geolocation;

alter table order\_payments add foreign key (order\_id) references orders;

alter table order\_items add foreign key (product\_id) references product;

alter table order\_reviews add foreign key (order\_id) references orders;

**Annual Customer Activity Growth Analysis**

1. Average Monthly Active User

select

order\_year,

round(avg(MAU),2) as average\_mau

from

(select

date\_part('year', o.order\_purchase\_timestamp) as order\_year,

date\_part('month', o.order\_purchase\_timestamp) as order\_monnth,

count(distinct customer\_unique\_id) as MAU

from orders\_dataset o

join customer\_dataset c on o.customer\_id = c.customer\_id

group by 1,2) subq

group by 1

1. New Customer

select

date\_part('year', order\_year) as order\_year,

count(1) as new\_customer

from

(select

c.customer\_id,

min(o.order\_purchase\_timestamp) as order\_year

from orders\_dataset o

join customer\_dataset c on o.customer\_id = c.customer\_id

group by 1) subq

group by 1

1. Repeat Customer

select

order\_year,

count(distinct customer\_unique\_id) as repeating\_customers

from

(select

date\_part('year', o.order\_purchase\_timestamp) as order\_year,

c.customer\_unique\_id,

count(1) as purchase\_frequency

from orders\_dataset o

join customer\_dataset c on o.customer\_id = c.customer\_id

group by 1,2

having count(1)>1) subq

group by 1

1. Average Order Frecuency

select

order\_year,

round(avg(purchase\_frequency),3) as avg\_purchase\_frequency

from

(select

date\_part('year', o.order\_purchase\_timestamp) as order\_year,

c.customer\_unique\_id,

count(1) as purchase\_frequency

from orders\_dataset o

join customer\_dataset c on o.customer\_id = c.customer\_id

group by 1,2) subq

group by 1

1. Menggabungkan keempat metrik

with

MAU as(

select

order\_year,

round(avg(MAU),2) as average\_mau

from

(select

date\_part('year', o.order\_purchase\_timestamp) as order\_year,

date\_part('month', o.order\_purchase\_timestamp) as order\_monnth,

count(distinct customer\_unique\_id) as MAU

from orders\_dataset o

join customer\_dataset c on o.customer\_id = c.customer\_id

group by 1,2) subq

group by 1),

new\_customer as(

select

date\_part('year', order\_year) as order\_year,

count(1) as new\_customer

from

(select

c.customer\_id,

min(o.order\_purchase\_timestamp) as order\_year

from orders\_dataset o

join customer\_dataset c on o.customer\_id = c.customer\_id

group by 1) subq

group by 1),

repeat\_customer as(

select

order\_year,

count(distinct customer\_unique\_id) as repeating\_customers

from

(select

date\_part('year', o.order\_purchase\_timestamp) as order\_year,

c.customer\_unique\_id,

count(1) as purchase\_frequency

from orders\_dataset o

join customer\_dataset c on o.customer\_id = c.customer\_id

group by 1,2

having count(1)>1) subq

group by 1),

avg\_purchase\_freq as(

select

order\_year,

round(avg(purchase\_frequency),3) as avg\_purchase\_frequency

from

(select

date\_part('year', o.order\_purchase\_timestamp) as order\_year,

c.customer\_unique\_id,

count(1) as purchase\_frequency

from orders\_dataset o

join customer\_dataset c on o.customer\_id = c.customer\_id

group by 1,2) subq

group by 1)

select

mau.order\_year,

mau.average\_mau,

nc.new\_customer,

rc.repeating\_customers,

apf.avg\_purchase\_frequency

from MAU mau

join new\_customer nc on mau.order\_year = nc.order\_year

join repeat\_customer rc on rc.order\_year = mau.order\_year

join avg\_purchase\_freq apf on apf.order\_year = mau.order\_year

**Annual Product Category Quality Analysis**

1. Revenue per tahun

create table revenue\_per\_year as

select

date\_part('year', order\_purchase\_timestamp) as year,

sum(revenue\_per\_order) as revenue

from

(select

order\_id,

sum(price+freight\_value) as revenue\_per\_order

from order\_items\_dataset

group by 1) subq

join orders\_dataset o on subq.order\_id = o.order\_id

where o.order\_status = 'delivered'

group by 1

1. Jumlah Cancel Order per tahun

create table cancel\_per\_year as

select

date\_part('year', order\_purchase\_timestamp) as year,

count(1) as num\_canceled\_booking

from orders\_dataset

where order\_status = 'canceled'

group by 1

1. Kategori yang menghasilkan revenue terbesar per tahun

create table top\_product\_revenue\_per\_year as

select

year,

product\_category\_name,

revenue

from

(select

date\_part('year', o.order\_purchase\_timestamp) as year,

p.product\_category\_name,

sum(price+freight\_value) as revenue,

rank() over(partition by date\_part('year', o.order\_purchase\_timestamp)

order by sum(price+freight\_value) desc)

from order\_items\_dataset oi

join product\_dataset p on oi.product\_id = p.product\_id

join orders\_dataset o on o.order\_id = oi.order\_id

where o.order\_status = 'delivered'

group by 1,2) subq

where rank = 1

1. Kategori yang mengalami cancel order terbesar per tahun

create table top\_cancel\_product\_per\_year as

select

year,

product\_category\_name,

num\_canceled

from

(select

date\_part('year', o.order\_purchase\_timestamp) as year,

p.product\_category\_name,

count(2) as num\_canceled,

rank() over(partition by date\_part('year', o.order\_purchase\_timestamp)

order by count(1) desc)

from order\_items\_dataset oi

join product\_dataset p on oi.product\_id = p.product\_id

join orders\_dataset o on o.order\_id = oi.order\_id

where o.order\_status = 'canceled'

group by 1,2) subq

where rank = 1

1. Menggabungkan keempat tabel

select

cpy.year,

cpy.num\_canceled\_booking,

rpy.revenue,

tcp.product\_category\_name top\_product\_by\_canceled,

tcp.num\_canceled,

tpr.product\_category\_name top\_product\_by\_revenue,

tpr.revenue

from cancel\_per\_year cpy

join revenue\_per\_year rpy on cpy.year = rpy.year

join top\_cancel\_product\_per\_year tcp on tcp.year = cpy.year

join top\_product\_revenue\_per\_year tpr on tpr.year = cpy.year

**Analysis of Annual Payment Type Usage**

1. Total Jumlah Penggunaan Setiap Tipe Pembayaran

select

payment\_type,

count(1) as num\_payment

from order\_payments\_dataset

group by 1

order by 2 desc

1. Jumlah Penggunaan Setiap Tipe Pembayaran

select

payment\_type,

sum(case when year = '2016' then num\_used else 0 end) as year\_2016,

sum(case when year = '2017' then num\_used else 0 end) as year\_2017,

sum(case when year = '2018' then num\_used else 0 end) as year\_2018

from

(select

date\_part('year', o.order\_purchase\_timestamp) as year,

op.payment\_type,

count(2) as num\_used

from order\_payments\_dataset op

join orders\_dataset o on o.order\_id = op.order\_id

group by 1, 2) subq

group by 1