**SQL QUERY**

**Analyzing E-commerce Business Performance**

**Author: Awalsyah Rinanto Putra**

**Data Analyst/Data Scientist**

**Part I**

**Create Table and Set Primary and Foreign Key Query**

1. **Create Table Query**

* **Customers Dataset Table**

CREATE TABLE IF NOT EXISTS customers\_dataset (

customer\_id VARCHAR,

"customer\_unique\_id" VARCHAR,

"customer\_zip\_code\_prefix" VARCHAR,

"customer\_city" VARCHAR,

"customer\_state" VARCHAR

)

* **Sellers Dataset Table**

CREATE TABLE IF NOT EXISTS sellers\_dataset (

seller\_id VARCHAR,

seller\_zip\_code\_prefix VARCHAR,

seller\_city VARCHAR,

seller\_state VARCHAR

)

* **Orders Items Dataset Table**

CREATE TABLE IF NOT EXISTS order\_items\_dataset (

order\_id varchar,

"order\_item\_id" int,

"product\_id" varchar,

"seller\_id" varchar,

"shipping\_limit\_date" timestamp,

"price" float,

"freight\_value" float

)

* **Order Reviews Dataset Table**

CREATE TABLE IF NOT EXISTS order\_reviews\_dataset (

review\_id varchar,

"order\_id" varchar,

"review\_score" varchar,

"review\_comment\_title" varchar,

"review\_comment\_message" varchar,

"review\_creation\_date" timestamp,

"review\_answer\_timestamp" timestamp

)

**Product Dataset Table**

CREATE TABLE IF NOT EXISTS product\_dataset (

"" int,

product\_id varchar,

product\_category\_name varchar,

product\_name\_lenght float,

product\_description\_lenght float,

product\_photos\_qty float,

product\_weight\_g float,

product\_length\_cm float,

product\_height\_cm float,

product\_width\_cm float

)

* **Orders Dataset Table**

CREATE TABLE IF NOT EXISTS orders\_dataset (

order\_id varchar,

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

)

* **Geolocation Dataset Table**

CREATE TABLE IF NOT EXISTS geolocation\_dataset (

geolocation\_zip\_code\_prefix varchar,

geolocation\_lat real,

geolocation\_lng real,

geolocation\_city varchar,

geolocation\_state varchar

)

* **Orders Payments\_Dataset Table**

CREATE TABLE IF NOT EXISTS order\_payments\_dataset (

order\_id VARCHAR,

payment\_sequential int,

payment\_type VARCHAR,

payment\_installments float,

payment\_value float

)

1. **Set Primary Key Query**

alter table orders\_dataset

add primary key (order\_id);

alter table seller\_dataset

add primary key (seller\_id);

alter table product\_dataset

add primary key (product\_id);

alter table geolocation\_dataset

add primary key (geolocation\_zip\_code\_prefix);

1. **Foreign Key Query**

alter table order\_items\_dataset

add foreign key (order\_id) references orders\_dataset;

alter table order\_items\_dataset

add foreign key (seller\_id) references sellers\_dataset;

alter table order\_items\_dataset

add foreign key (product\_id) references product\_dataset;

alter table order\_reviews\_dataset

add foreign key (order\_id) references orders\_dataset;

alter table customers\_dataset

add foreign key (customer\_zip\_code\_prefix) references geolocation\_dataset;

alter table orders\_dataset

add foreign key (customer\_id) references customers\_dataset;

**Part II**

**SQL Query: Analyzing Annual Customer Growth Rate**

1. **Create ‘Monthly Active User Table’**

SELECT

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

count (DISTINCT customers\_dataset.customer\_id) / count (distinct date\_part('month', order\_purchase\_timestamp)) as MAU

from customers\_dataset

join orders\_dataset

ON orders\_dataset.customer\_id = customers\_dataset.customer\_id

group by 1

order by 1

* **Save ‘Monthly Active User Table’**

create table monthly\_active\_user\_table as (

SELECT

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

count (DISTINCT customers\_dataset.customer\_id) / count (distinct date\_part('month', order\_purchase\_timestamp)) as monthly\_active\_user

from customers\_dataset

join orders\_dataset

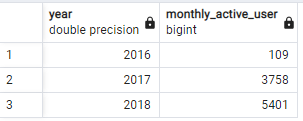
ON orders\_dataset.customer\_id = customers\_dataset.customer\_id

group by 1

order by 1

)

* **Monthly Active User Table**



1. **Create Annual New Customer Table**

with first\_year\_order as (

SELECT

customer\_unique\_id,

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

FROM customers\_dataset

join orders\_dataset

ON orders\_dataset.customer\_id = customers\_dataset.customer\_id

group by 1

)

select

year,

count (distinct customer\_unique\_id) as new\_customer

from first\_year\_order

group by 1

order by 1

* **‘Save Annual New Customer Table’**

**c**reate table annual\_new\_customer\_table as (

with first\_year\_order as (

SELECT

customer\_unique\_id,

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

FROM customers\_dataset

join orders\_dataset

ON orders\_dataset.customer\_id = customers\_dataset.customer\_id

group by 1

)

select

year,

count (distinct customer\_unique\_id) as new\_customer

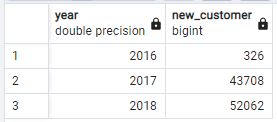
from first\_year\_order

group by 1

order by 1

)

* **‘Annual\_New\_Customer Table’ View**



1. **Create ‘Annual Customer Repeat Order’ Table**

with annual\_order\_table as (

SELECT

customer\_unique\_id,

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

count (order\_purchase\_timestamp) as total\_order

from customers\_dataset

join orders\_dataset

ON orders\_dataset.customer\_id = customers\_dataset.customer\_id

group by 1, 2

having count (order\_purchase\_timestamp) > 1

)

select

year,

count (total\_order) as total\_repeat\_order

from annual\_order\_table

group by 1

order by 1

* **Save ‘Annual\_Customer Repeat\_Order\_Table’**

create table annual\_repeat\_order as (

with annual\_order\_table as (

SELECT

customer\_unique\_id,

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

count (order\_purchase\_timestamp) as total\_order

from customers\_dataset

join orders\_dataset

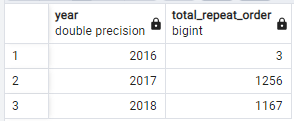
ON orders\_dataset.customer\_id = customers\_dataset.customer\_id

group by 1, 2

having count (order\_purchase\_timestamp) > 1

)

* **‘Annual\_Customer Repeat\_Order Table’ View**



1. **Create ‘Annual Total Order’ Table**

with total\_all\_order as (

SELECT

customer\_unique\_id,

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

count (order\_purchase\_timestamp) as total\_order

from customers\_dataset

join orders\_dataset

ON orders\_dataset.customer\_id = customers\_dataset.customer\_id

group by 1, 2

)

select

year,

count (total\_order) as total\_repeat\_order

from total\_all\_order

group by 1

order by 1

* **Save ‘Annual\_Total Order\_Table’**

create table annual\_total\_order\_table as (

with total\_all\_order as (

SELECT

customer\_unique\_id,

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

count (order\_purchase\_timestamp) as total\_order

from customers\_dataset

join orders\_dataset

ON orders\_dataset.customer\_id = customers\_dataset.customer\_id

group by 1, 2

)

select

year,

count (total\_order) as total\_order

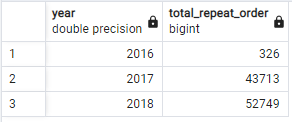
from annual\_total\_order\_table

group by 1

order by 1

)

* **‘Annual\_Total\_Order\_Table’ View**



1. **Joint Annual New Customer, Annual Customer Repeat Order, Annual Total Order and Monthly Active User Tables**

select

mau.year,

mau.monthly\_active\_user,

nct.new\_customer,

aro.customer\_repeat\_order,

atr.total\_order

from monthlyactiveuser\_table as mau

join annual\_new\_customer\_table as nct

on mau.year = nct.year

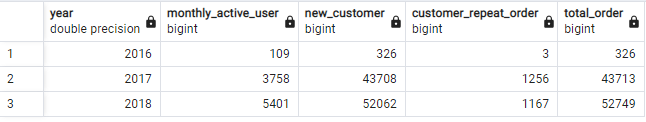
join annual\_repeat\_order as aro

on mau.year = aro.YEAR

join annual\_total\_order\_table as atr

on mau.year = atr.year

* **Final Table View**



**Part III**

**SQL Query: Analyzing Product Category Quality Analysis**

1. **Analyzing Annual Revenue**

CREATE TABLE annual\_revenue\_table as

select

year,

Revenue,

order\_status

FROM (Select

sum(oi.price+oi.freight\_value) as Revenue,

DATE\_PART('YEAR',o.order\_purchase\_timestamp) as year,

o.order\_status

FROM orders\_dataset AS o

JOIN order\_items\_dataset AS oi

ON o.order\_id = oi.order\_id

GROUP BY 2, 3) as rvn

WHERE order\_status in ('delivered')

GROUP BY 1,2,3

ORDER BY 1

1. **Analyzing Annual Cancelled Order**

CREATE TABLE annual\_cancelled\_order\_table AS

SELECT

DATE\_PART('YEAR',order\_purchase\_timestamp) as year,

count(order\_status) as cancelled

FROM orders\_dataset

WHERE order\_status in ('canceled')

GROUP BY 1

ORDER BY year;

1. **Analyzing Top Product Category Generates The Most Revenue**

CREATE TABLE top\_category\_table AS

SELECT

year,

rank,

product\_category\_name

FROM (

SELECT

product\_category\_name,

year,

pendapatan,

rank() over(partition by year ORDER BY pendapatan DESC)

FROM(

SELECT

product\_category\_name,

DATE\_PART('YEAR',o.order\_purchase\_timestamp) as year,

sum(oi.price+oi.freight\_value) as pendapatan

FROM order\_items\_dataset as oi

JOIN product\_dataset as p on oi.product\_id = p.product\_id

JOIN orders\_dataset as o on oi.order\_id = o.order\_id

GROUP BY 1,2

)as ass

GROUP BY 1,2,3) as ast

WHERE rank=1

GROUP BY 1,2,3

ORDER BY 1;

1. **Analyzing Top Cancelled Product Category**

CREATE TABLE annual\_cancelled\_order\_table AS

SELECT

DATE\_PART('YEAR',order\_purchase\_timestamp) as year,

count(order\_status) as cancelled

FROM orders\_dataset

WHERE order\_status in ('canceled')

GROUP BY 1

ORDER BY year;

1. **Table Configuration**

*Note: Because some columns have the same name, we need to rename the columns name to avoid error.*

alter table top\_category\_cancel\_table

rename column product\_category\_name to cancel\_product\_category

alter table top\_category\_table

rename column product\_category\_name to top\_product\_category

1. **Table Joints**

create table if not exists product\_performance\_table as (

SELECT

ar.year,

ar.revenue,

tc.product\_category\_name,

tcc.product\_category,

aco.cancelled

FROM annual\_revenue\_table as ar

JOIN top\_category\_table as tc on ar.year=tc.year

JOIN top\_category\_cancel\_table as tcc on ar.year=tcc.year

JOIN annual\_cancelled\_order\_table as aco on ar.year=aco.year

)

**Part IV**

**SQL QUERY: PAYMENT TYPE USAGE ANALYSIS**

1. **Create All-Time Top Payment Type Usage**

SELECT

op.payment\_type,

count(o.order\_id) as total\_cust

FROM orders\_dataset as o

JOIN order\_payments\_dataset as op on o.order\_id = op.order\_id

GROUP BY 1

ORDER BY 2 desc

1. **Create ‘Annual Payment Type Usage’ Table**

SELECT

payment\_type,

count(CASE WHEN (DATE\_PART('YEAR',o.order\_purchase\_timestamp)) ='2016' then 1 ELSE Null

END) AS "2016",

count(CASE WHEN (DATE\_PART('YEAR',o.order\_purchase\_timestamp))='2017' then 2 ELSE Null

END) AS "2017",

count(CASE WHEN (DATE\_PART('YEAR',o.order\_purchase\_timestamp))='2018' then 3 ELSE Null

END) AS "2018"

FROM orders\_dataset as o

JOIN order\_payments\_dataset as op on o.order\_id = op.order\_id

GROUP BY 1

ORDER BY payment\_type ASC;