THE LOOK E-COMMERCE QUERY

--- **create table order_items** ---

```
create table order_items
(
id int8 primary key,
order_id int8,
user_id int8,
product_id int8,
inventory_item_id int8,
status varchar(255),
created_at timestamptz,
shipped_at timestamptz,
delivered_at timestamptz,
returned_at timestamptz,
sale_price  numeric
)
```

--- **import data into order_items** ---

```
copy order_items (id, order_id, user_id, product_id, inventory_item_id, status, created_at, shipped_at,
                  delivered_at, returned_at, sale_price)
from 'D:\SKD\Data Analyst\3. SQL\4. Project\7. The Look E-commerce\
      The Look E-commerce Data\order_items.csv'
delimiter ','
csv header
```

--- **size and details of order_items** ---

```
select * from order_items
select count(*) as rows from order_items
select count(*) as columns from information_schema.columns where table_name = 'order_items'
select column_name, data_type from information_schema.columns where table_name = 'order_items'
```

--- **create table orders** ---

```
create table orders
(
order_id int8 primary key,
user_id int8,
status varchar(255),
gender varchar(255),
created_at timestamptz,
returned_at timestamptz,
shipped_at timestamptz,
delivered_at timestamptz,
num_of_item int8
)
```

--- **import data into orders** ---

```
copy orders (order_id, user_id, status, gender, created_at, returned_at, shipped_at, delivered_at,
                num_of_item)
from 'D:\SKD\Data Analyst\3. SQL\4. Project\7. The Look E-commerce\
        The Look E-commerce Data\orders.csv'
delimiter ','
csv header
```

--- **size and details of orders** ---

```
select * from orders
select count(*) as rows from orders
select count(*) as columns from information_schema.columns where table_name = 'orders'
select column_name, data_type from information_schema.columns where table_name = 'orders'
```

--- **create table products** ---

```
create table products
(
id int8 primary key,
cost numeric,
category varchar(255),
name varchar(255),
brand varchar(255),
retail_price numeric,
department varchar(255),
sku varchar(255),
distribution_center_id int8
)
```

--- **import data into products** ---

```
copy products (id, cost, category, name, brand, retail_price, department, sku, distribution_center_id)
from 'D:\SKD\Data Analyst\3. SQL\4. Project\7. The Look E-commerce\The Look E-commerce Data\products.csv'
delimiter ','
csv header
```

--- **size and details of products** ---

```
select * from products
select count(*) as rows from products
select count(*) as columns from information_schema.columns where table_name = 'products'
select column_name, data_type from information_schema.columns where table_name = 'products'
```

--- **create table users** ---

```
create table users
(
id int8 primary key,
first_name varchar(255),
last_name varchar(255),
email varchar(255),
age int8,
gender varchar(255),
state varchar(255),
street_address varchar(255),
postal_code varchar(255),
city varchar(255),
country varchar(255),
latitude numeric,
longitude numeric,
traffic_source varchar(255),
created_at timestamptz
)
```

--- **import data into users** ---

```
copy users (id, first_name, last_name, email, age, gender, state, street_address, postal_code, city, country,
                latitude, longitude, traffic_source, created_at)
from 'D:\SKD\Data Analyst\3. SQL\4. Project\7. The Look E-commerce\The Look E-commerce Data\users.csv'
delimiter ','
csv header
```

--- **size and details of users** ---

```
select * from users
select count(*) as rows from users
select count(*) as columns from information_schema.columns where table_name = 'users'
select column_name, data_type from information_schema.columns where table_name = 'users'
```

--- **Answering the Business Questions of the_look** ---

```
-- Q1. How much are we selling monthly? Is it high or low compared to last month?
select
        extract (month from order_items.created_at) as month,
        to_char (order_items.created_at, 'month') as month_name,
        round(sum(order_items.sale_price * orders.num_of_item), 2) as revenue,
        count(distinct order_items.order_id) as order_count,
        count(distinct order_items.user_id) as customers_purchased
from order_items
inner join orders
on order_items.order_id = orders.order_id
where order_items.status not in ('Cancelled', 'Returned')
group by month, month_name
order by revenue desc

-- Q2. Who are our customers? Which country do we have major customers coming from?
--      Which Gender and Age group brought in the most profit?

-- a. customers by country
with customers
as(
        select
                distinct order_items.user_id,
                sum (case when users.gender = 'M' then 1 else null end) as male,
                sum (case when users.gender = 'F' then 1 else null end) as female,
                users.country as country
        from order_items
        inner join users
        on order_items.user_id = users.id
        where order_items.status not in ('Cancelled', 'Returned')
        group by user_id, country
        )
select
        country,
        count(distinct user_id) as customers_count,
        count(female) as female,
        count(male) as male
from customers
group by country
order by customers_count desc

-- b. customers by gender
select
        orders.gender,
        round(sum(order_items.sale_price * orders.num_of_item), 2) as revenue,
        sum (orders.num_of_item) as quantity
from order_items
inner join orders
on order_items.user_id = orders.user_id
```

```
where order_items.status not in ('Cancelled', 'Returns')
group by gender
order by revenue desc

-- c. customers by age group
select
        case
                when users.age < 12 then 'kids'
                when users.age between 12 and 20 then 'teenagers'
                when users.age between 20 and 30 then 'young adults'
                when users.age between 30 and 50 then 'adults'
                when users.age > 50 then 'elderly'
                end as age_group,
        count (distinct order_items.user_id) as total_customer
from order_items
inner join users
on order_items.user_id = users.id
group by age_group
order by total_customer desc

-- Q3. What brands and product categories are we selling more and the least?
--        What are we making money on?

-- a. brand sales
select
        products.brand as brand,
        round(sum(sale_price * num_of_item), 2) as revenue,
        sum(num_of_item) as quantity
from order_items
inner join orders
on order_items.order_id = orders.order_id
inner join products
on order_items.product_id = products.id
where order_items.status not in ('Cancelled', 'Returned')
group by brand
order by revenue desc

-- b. product category sales
select
        category as product_category,
        round(sum(sale_price * num_of_item), 2) as revenue,
        sum(num_of_item) as quantity
from order_items
inner join orders
on order_items.order_id = orders.order_id
inner join products
on order_items.product_id = products.id
where order_items.status not in ('Cancelled', 'Returned')
group by category
order by revenue desc
```

```sql
-- Q4. What are the most cancelled and returned brands and product categories?

-- a. brand cancellation and return
select
        products.brand as brand,
        sum (case when order_items.status = 'Cancelled' then 1 else null end) as cancelled,
        sum (case when order_items.status = 'Returned' then 1 else null end) as returned
from order_items
inner join products
on order_items.product_id = products.id
group by brand
order by cancelled desc
-- order by returned desc

-- b. product category cancellation ansd return
select
        products.category as category,
        sum (case when order_items.status = 'Cancelled' then 1 else null end) as cancelled,
        sum (case when order_items.status = 'Returned' then 1 else null end) as returned
from order_items
inner join products
on order_items.product_id = products.id
group by category
order by cancelled desc
-- order by returned desc

-- Q5. What marketing channel are we doing well on?
select
        users.traffic_source as traffic_source,
        count(distinct order_items.user_id) as total_customers
from order_items
inner join users
on order_items.user_id = users.id
where order_items.status not in ('cancelled', 'Returned')
group by traffic_source
order by total_customers desc

-- Q6. We will provide promotions during Chinese New Year celebrations for female customers in China via
--         email.
select
        id,
        email
from users
where
        gender = 'F'
        and country = 'China'
order by id
```

```
-- Q7. Provide a list of 10 customer IDs and emails with the largest total overall purchase. We will give a
--         discount for Campaign 3.3!
select
        users.id as customer_id,
        users.email as email,
        round(sum(order_items.sale_price * orders.num_of_item), 2) as total_purchase
from order_items
inner join orders
on order_items.order_id = orders.order_id
inner join users
on orders.user_id = users.id
group by customer_id, email
order by total_purchase desc
limit 10

-- Q8. Create a query to get frequencies, average order value, and the total number of unique users
--         where status is completed grouped by month (Skillset: Intermediate SQL)
select
        to_char (created_at,  'yyyy-mm') as month_year,
        round ((count(distinct order_id)/count(distinct order_id)), 2) as frequencies,
        round ((sum (sale_price)/count(distinct order_id)), 2) as average_order_value,
        count (distinct user_id) as total_unique_users
from order_items
where status = 'Complete'
group by month_year
order by month_year desc
```