-- What is the average order value per customer?

select t1.customer\_id, concat(t1.first\_name, ' ', t1.last\_name) as customer\_name,

count(distinct t3.order\_id) as number\_of\_order, avg(t3.quantity\*t3.list\_price - t3.discount) as average\_order\_value

from customers t1

join orders t2 on t1.customer\_id = t2.customer\_id

join order\_items t3 on t2.order\_id = t3.order\_id

group by 1,2;

-- How many products are sold per category?

select t1.category\_id, t3.product\_id, t2.product\_name, sum(t3.quantity) as number\_of\_quantity\_sold from categories t1

join products t2 on t1.Category\_id = t2.category\_id

join order\_items t3 on t2.product\_id = t3.product\_id

group by 1,2,3;

-- What percentage of total orders come from each store?

select t1.store\_id, t1.store\_name,

count(distinct t2.order\_id) as Number\_of\_orders, ((t3.list\_price\*t3.quantity)-t3.discount) as sales\_amount,

round((count(distinct t2.order\_id)/(select count(\*) from order\_items)),2) as Total\_order\_percentage

from stores t1

join orders t2 on t1.store\_id = t2.store\_id

join order\_items t3 on t2.order\_id = t3.order\_id

group by 1,2;

-- What is the most comman order status?

select order\_status, count(order\_status) as Order\_status\_frequency

from orders

group by 1

order by Order\_status\_frequency desc;

-- How do product sales vary across different stores

select t1.store\_id, t1.store\_name,

t4.product\_name, sum(t3.quantity) as 'Total\_quantity\_sold',

sum((t3.list\_price\*t3.quantity)-t3.discount) as 'Sales'

from stores t1

join orders t2 on t1.store\_id = t2.store\_id

join order\_items t3 on t2. order\_id = t3.order\_id

join products t4 on t4.product\_id = t3.product\_id

group by 1,2,3;

-- What is the trend in monthly sales over the past year

select

monthname(order\_date) as 'Month',

year(order\_date) as 'year',

sum((list\_price\*quantity) - discount) as 'Total\_sales'

from orders t1

join order\_items t2 on t1.order\_id = t2.order\_id

where t1.order\_date >= date\_sub('2018-12-28',interval 12 month)

group by 1

order by 1;

-- What is the correlation between discount percentage and total sales volume?

select ((discount/list\_price)\*100) as 'Discount\_percentage',

sum(quantity) as 'Total\_sales\_volume'

from order\_items

group by 1

order by 1 desc;

-- How many repeat customers are there and how frequently do they place orders?

select t1.customer\_id, concat(t1.first\_name,' ', t1.last\_name) as 'Customer\_name',

count(t1.customer\_id) as 'Repeat\_customers', count(t3.order\_id) as 'Repeat\_orders'

from customers t1

join orders t2 on t1.customer\_id = t2.customer\_id

join order\_items t3 on t2.order\_id = t3.order\_id

group by 1,2;

-- Which products have a seasonal trend in sales?

select t1.product\_id,

t1.product\_name,

concat(monthname(t3.order\_date), ' ',year(t3.order\_date)),

count(t2.order\_id) as 'Number of orders placed'

from products t1

join order\_items t2 on t1.product\_id = t2.product\_id

join orders t3 on t3.order\_id = t2.order\_id

where order\_date >= subdate('2018-12-28', interval 24 month)

group by 1, 2, 3

order by 1;

-- How does customer location affect purchasing behavior?

select

distinct t1.customer\_id,

concat(t1.city, ' ',t1.state) as "Customer's Address",

t1.state,

concat(t3.city, ' ',t3.state) as "store's Address",

t3.state,

count(t2.order\_id) as "Number of order placed",

sum(t4.quantity) as "Total\_Quantity",

sum(t4.List\_price) as "Total\_printed\_cost\_per\_item",

(sum((t4.list\_price\*t4.quantity) - t4.discount)) as "Total\_sales"

from customers t1

join orders t2 on t1.customer\_id = t2.customer\_id

join stores t3 on t3.store\_id = t2.store\_id

join order\_items t4 on t2.order\_id = t4.order\_id

group by 1,2,3,4,5;

-- Perform an RFM(Recency, Frequency, Monetary) analysis

With RFM as(

select t1.customer\_id, concat(t1.first\_name, ' ', t1.last\_name) as "Customer\_name",

datediff(curdate(), max(t2.order\_date)) as "Recency", count(t3.order\_id) as "Frequency",

sum((t3.list\_price\*t3.quantity) - t3.discount) as "Monetary"

from customers t1

join orders t2 on t1.customer\_id = t2.customer\_id

join order\_items t3 on t2.order\_id = t3.order\_id

group by 1,2

)

select customer\_id, customer\_name , Recency as "Recency (No. of days since last purchase)",

Frequency as "Frequency (No. of total orders placed)",

Monetary as "Monetary (Total amount Spent)"

from RFM

-- Write a query to display all columns from the stores table.

desc stores;

-- Retrieve the names and emails of all customers from the customers table.

select (first\_name,' ',last\_name) as "customer\_full\_name", email as "customer's email"

from customers;

-- Count the total number of products available in the products table.

select product\_id, product\_name from products

group by 1;

-- List all distinct order statuses from the orders table.

select order\_status from orders;

-- Find the number of employees (staff) working in the company.

select count(distinct staff\_id) from staffs;

-- Display all store details where the state is 'California'.

select \* from stores

where state = 'CA';

-- Retrieve all product details where the list price is greater than 100.

select \* from products t1

join order\_items t2 on t1.product\_id = t2.product\_id

where t2.list\_price > 100;

-- Write a query to display the store names and their corresponding zip codes.

select store\_name, zip\_code from stores;

-- Retrieve all records from the stocks table where quantity is greater than 50.

select \* from stocks

where quantity > 50;

-- Find all customers whose first name starts with the letter ‘A’.

select concat(first\_name,' ', last\_name) as "Customer's\_name"

from customers

where first\_name like "A%"

-- Retrieve the product name and brand name for all products by joining products and brands.

select t1.product\_name, t2.brand\_name from products t1

join brands t2 on t1.brand\_id = t2.brand\_id;

-- List all orders along with the customer’s first name and last name.

select distinct t2.order\_id, concat(t3.first\_name, ' ', t3.last\_name) as "Customer's name"

from order\_items t1

join orders t2 on t1.order\_id = t2.order\_id

join customers t3 on t2.customer\_id = t3.customer\_id;

-- Find the total quantity of all products available in stock across all stores.

select t1.store\_id, t2.store\_name, t3.product\_name, sum(quantity) AS "Total\_stock\_quantity"

from stocks t1

join stores t2 on t1.store\_id = t2.store\_id

join products t3 on t1.product\_id = t3.product\_id

group by 1,2,3;

-- Write a query to display all products that belong to the "Electronics" category (assume the category name is stored in categories).

select t1.category\_id, t1.category\_name, t2.product\_name,

t2.list\_price from categories t1

join products t2 on t1.Category\_id = t2.category\_id

where category\_name like 'E%'

ORDER BY t2.list\_price DESC;

-- Get the total number of orders placed by each customer (customer\_id, first\_name, total\_orders).

select

t1.customer\_id,

concat(t1.first\_name,' ',t1.last\_name) as "Customer's name",

sum(t2.order\_id) as "Number of Order Placed"

from customers t1

join orders t2 on t1.customer\_id = t2.customer\_id

group by 1,2;

-- Find the total revenue generated from each order (order\_id, total\_price)

select order\_id, sum(list\_price \* quantity) as "Total\_printed\_price",

sum((list\_price \* quantity) - discount) from order\_items

group by 1;

-- Retrieve all products that have never been ordered.

select t1.product\_id, t1.product\_name from products t1

left join order\_items t2 on t1.product\_id = t2.product\_id

where t2.product\_id is NULL;

-- Display the details of orders that have not been shipped yet (shipped\_date is NULL).

select order\_id, shipped\_date from orders

where shipped\_date is NULL;

-- Find the total number of products available in each store.

select t1.store\_id, t1.store\_name, sum(t2.quantity) as "Total\_quantity" from stores t1

join stocks t2 on t1.store\_id = t2.store\_id

group by 1, 2;

-- Write a query to display the details of the highest-priced product in each category.

select t1.category\_id,

t1.category\_name,

t2.product\_name,

max(t3.list\_price) as "highest-priced product"

from categories t1

join products t2 on t1.Category\_id = t2.category\_id

join order\_items t3 on t2.product\_id = t3.product\_id

group by 1,2;

-- Retrive the top 5 customers who have placed the most orders.

select t1. customer\_id, concat(t1.first\_name, ' ', t1.last\_name) as "Customer's\_name",

sum(t3.quantity) as "Number of order placed"

from customers t1

join orders t2 on t1.customer\_id = t2.customer\_id

join order\_items t3 on t2.order\_id = t3.order\_id

group by 1,2

limit 5;

-- Find the store that has the highest total stock quantity.

with highest\_stocks as (

select t1.store\_id, t1.store\_name, sum(t2.quantity) as highest\_total\_stock\_quantity from stores t1

join stocks t2 on t1.store\_id = t2.store\_id

group by 1,2)

select store\_id, store\_name, highest\_total\_stock\_quantity from highest\_stocks

order by highest\_total\_stock\_quantity desc

limit 1;

-- Write a query to calculate the total revenue generated by each store.

select t1.store\_id, t1.store\_name,

sum((t3.list\_price\*t3.quantity) - t3.discount) as "Total Revenue Generated"

from stores t1

join orders t2 on t1.store\_id = t2.store\_id

join order\_items t3 on t2.order\_id = t3.order\_id

group by 1,2;

-- Identify which product has been order the most and the number of times it has been ordered.

with most\_ordered\_product as

(

select t1.product\_id, t1.product\_name,

count(t2.product\_id) as "Number\_of\_times",

sum(t2.quantity) as "Most\_ordered\_Product"

from products t1

join order\_items t2 on t1.product\_id = t2.product\_id

group by 1,2)

select product\_id, product\_name, Number\_of\_times, Most\_ordered\_Product

from most\_ordered\_product;

-- Find all products that have been ordered more than 10 times in a single order.

with most\_ordered\_product as

(

select t1.product\_id, t1.product\_name,

count(t2.order\_id) as "Number\_of\_times"

from products t1

join order\_items t2 on t1.product\_id = t2.product\_id

group by 1,2)

select product\_id, product\_name, Number\_of\_times

from most\_ordered\_product

where Number\_of\_times>10

order by Number\_of\_times ASC;

-- create a stored procedure that accepts a store\_id and returns all products available in that store.

Create procedure store\_retrival(IN store\_id varchar(100))

begin

select t1.product\_id, product\_name from stocks t1

join products t2 on t1.product\_id = t2.product\_id

where store\_id = store\_id

end;

-- An sql query to retrive all store names and their corresponding zip codes

select store\_name, zip\_code from stores;

-- The total number of customers in the database.

select count(customer\_id) from customers;

-- The list of unique product categories available

select distinct category\_id, category\_name from categories;

-- Retrive all orders placed by a specific customer using their customer ID

select \* from orders

where customer\_id = 1;

-- or

select distinct customer\_id, count(t2.order\_id) from orders t1

left join order\_items t2 on t1.order\_id = t2.order\_id

group by 1;

-- List all products with a price higher than $500

select product\_id, product\_name, list\_price from products

where list\_price > 500;

-- Retrive the total number of products available in stock for each store

select distinct t1.store\_id, t2.product\_id, t2.quantity from stores t1

left join stocks t2 on t1.store\_id = t2.store\_id;

-- Find the total revenue generated from each product in the order\_item table

select product\_id, quantity, list\_price,

(quantity\*list\_price) as Total\_price\_before\_discount,

discount, ((quantity \* list\_price \* discount)/100) as Total\_discount\_in\_amount,

sum(((quantity\*list\_price) - ((list\_price \* discount)/100))) as Total\_revenue

from order\_items

group by 1

order by 1;

-- Get a list of customers who have placed more than 5 orders

select t1.customer\_id, concat(t1. first\_name, ' ',t1.last\_name) as Customer\_name , count(t3.order\_id) as Number\_of\_customers from customers t1

left join orders t2 on t1.customer\_id = t2.customer\_id

left join order\_items t3 on t2.order\_id = t3.order\_id

group by 1

having count(t3.order\_id) > 5;

-- Display all orders that have not yet been shipped

select order\_id, shipped\_date from orders

where shipped\_date is null;

-- Retrieve the total aales per store along with store name

select t1.store\_id, t1.store\_name, sum((t3.quantity\*t3.list\_price)-t3.discount) Total\_sales

from stores t1

left join orders t2 on t1.store\_id = t2.store\_id

left join order\_items t3 on t2.order\_id = t3.order\_id

group by 1,2;

-- Identify the top 5 best-selling products based on quantity sold

select t1.product\_id, t1.product\_name, sum(t2.quantity) as total\_quantity from products t1

left join order\_items t2 on t1.product\_id = t2.product\_id

group by 1,2

order by total\_quantity desc limit 5;

-- Find the average discount applied to all products in each category

select t1.product\_id, t1.product\_name, avg(t2.discount) as Average\_discount from products t1

left join order\_items t2 on t1.product\_id = t2.product\_id

group by 1,2;

-- Write a query to determine which store has the highest inventory value.

select t1.store\_id, t1.store\_name, sum(t3.quantity\*t3.list\_price) as Inventory\_value from stores t1

join orders t2 on t1.store\_id = t2.store\_id

join order\_items t3 on t2.order\_id = t3.order\_id

group by 1,2

order by Inventory\_value desc limit 1;

-- Retrieve the top 3 customers who have spent the most in total purcchase

select t1.customer\_id, concat(t1.first\_name,' ',t1.last\_name) as Customer\_name,

count(t3.order\_id) as Total\_orders\_placed, sum(t3.quantity\*t3.list\_price) as total\_purchase

from customers t1

join orders t2 on t1.customer\_id = t2.customer\_id

join order\_items t3 on t2.order\_id = t3.order\_id

group by 1,2

order by total\_purchase desc

limit 3;

-- Find the most frequently ordered product in the last 6 months

select t1.product\_id, t1.product\_name, t3.order\_date, sum(t2.quantity) as Quantity from products t1

join order\_items t2 on t1.product\_id = t2.product\_id

join orders t3 on t2.order\_id = t3.order\_id

where t3.order\_date >= date\_sub('2018-12-28', interval 6 month)

group by 1,2,3

order by t2.quantity desc;