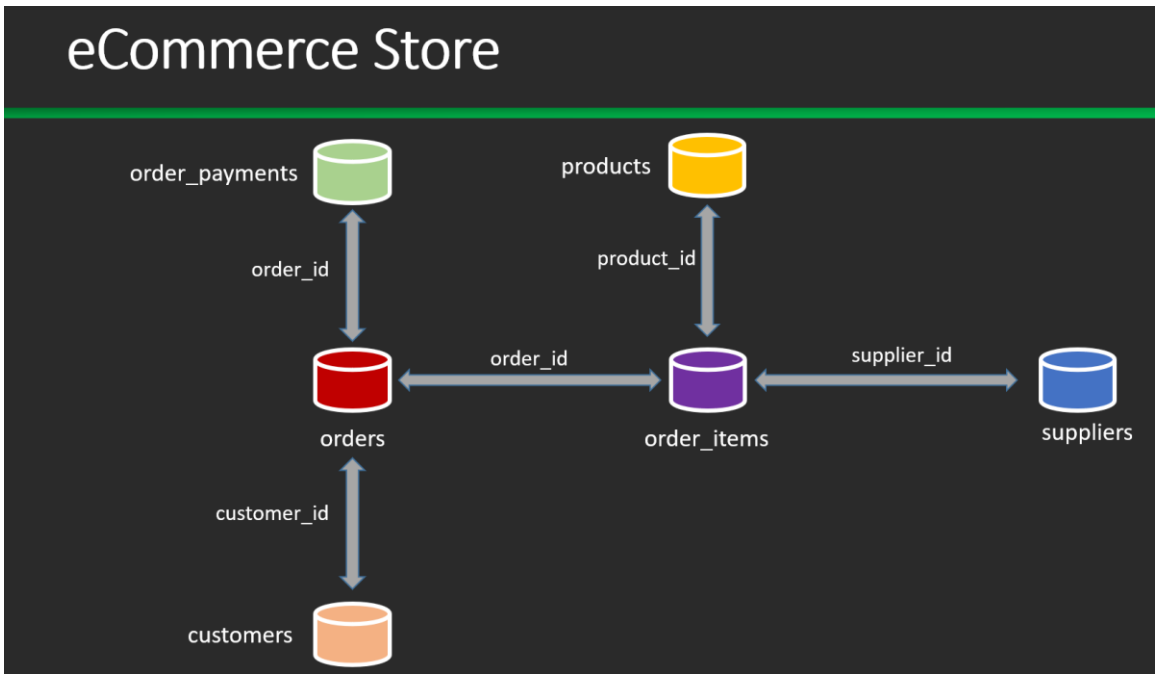
Level 2- Final Project

Introduction

Welcome to our final project. In this project, we will use the eCommerce store datasets we already uploaded during the training. The following diagram will be helpful in quickly identifying the connections between the tables. To check your answers, please use the second file with the project solution.

Good luck!



Part 1 - Combining Data from Multiple Tables

Exercise #1

The list of all books (book title, year of publication) published between 1997 and 2005 and ordered in ascending by the year of publication (Tip – there are two separate tables for books: Book1 table is for books published until 1999 and the book2 table is for books published after 1999).

Answer:

Exercise #2

The list of publishers that published books in 1990 and 2004 (only publishers that published books in both years). (Tip – use the intersect operator).

Answer:

Exercise #3

Answer:

Exercise #4

List all order items (order id, order item id, price) with the product information (product id, product category name), with a price higher than 3000, order by the order id and then order item id.

Answer:

Exercise #5

List all orders (order id, order status) that order status is “shipped” with their order items (order item id, price) with the product information (product id, product category name) with a price between 50 and 60, order by the order id and then order item id. (tip – you will need more than one join).

Answer:

Exercise #6

List of all products (product id, product category name) from the category “art” with their order items (order id, order item id, price) whether the product was ordered or not.

Answer:

Part 2 - Subqueries

Exercise #7

All products from the “art” or “perfumery” category (product id, product category name, product price) that the product price is less than the average price of all products.

Answer:

Exercise #8

List all the names of customers that have orders with a “delivered” status with the products category “art” and price is more than 120 (Tip – use three sub-queries, break the problem into small steps, and check each sub-query before adding the outer query, start with the products table).

Answer:

Part 3 - Conditional Logic (CASE)

Exercise #9

List all orders reviews (order id, review score) and transform the review score (scale 1 to 5) into a wording scale ('Very Dissatisfied', 'Dissatisfied', 'Neutral', 'Satisfied', 'Very Satisfied')

Answer:

Exercise #10

List all products (product id, product category, product price) and add a new column that will classify each product into a specific price category:

- Price<=50 → "Low Price"
- 50<Price<100 → "Medium Price"
- Price>100 → "High Price"

Answer:

Exercise #11

Based on the classification in the previous answer, summarize the number of products per each price category.

Answer:

Part 4 - Window Functions

Exercise #12

List of products (product id, product price, product category) and for each product, the average product price of the product category. Remove rows with NULL value in product price, and order the result based on product price descending.

Answer:

Exercise #13

List the five heaviest products (product id, product category, product weight) from each product category.

Answer:

Part 5 - Simplify Queries (Views, CTEs)

Exercise #14

Create a view called “customers_shipped_vw” for all customers (customer id, customer name, customer city, order status) with orders that are in “shipping” status.

Answer:

Exercise #15

Create a CTE (“with”) that will create a sequential number based on the product price for each product category (product id, product category, product price, row num) and remove rows with NULL value in the product price.

Use the CTE to query for the three most expensive products from each category.

Answer: