**INSY – 661**

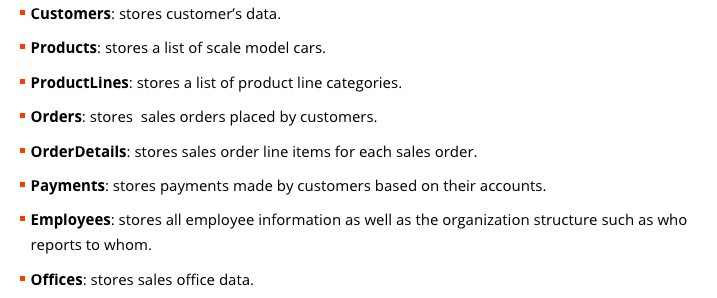
**Assignment 2: SQL**

Please refer to the ERD shown below and write SQL queries to answer the questions that follow.

A sample database for this ERD is available at: <https://www.db-fiddle.com/f/eFE3c17RDHaQ1gzqDrqwYJ/0>

**Diagram

Description automatically generated**

****

**DDL**

**A screenshot of a cell phone

Description automatically generated**

**A close up of a logo

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

**A close up of a logo

Description automatically generated**

**NOTE: Your answer may differ as there are many different ways to do answer the same question. Moreover, in some cases, the question may be ambiguous such that it is interpreted differently. In such case, one should write down their assumption.**

Q1. Write an SQL Command to display the order number, order date, product code, quantity ordered, total payment for this product (unit price \* number of units) customer number, and customer name for order number 10101.

Sample report (note: values may differ if your data base has different data)

Table

Description automatically generated

Q2. Write an SQL Command to show the total amount to be paid by customer, product variety (i.e., total number of unique product id in that order), maximum unit price, and minimum unit price for order number 10101.

Sample report (note: values may differ if your data base has different data)

**Application

Description automatically generated with medium confidence**

Q3. Write an SQL Command that will find any customers (customer ID, customer Name, city, and Phone) from “Madrid” who have not placed orders, results should be ordered by customer ID (Descending).

Q4. Write an SQL query to find the top 10 popular products (i.e., total quantity ordered/sold). Display the product ID, product name, and total quantities ordered by customers.

Hint: Start with producing a list of all the products and the total quantity each product has been ordered.

Q5. Write an SQL query to list order number, product Code, quantity ordered, and avg quantity ordered for that product (across all orders) for those customer orders where the Order quantity for a product in that order is smaller than the average order quantity for that product (Across all orders). Results should be ordered by quantity ordered (Ascending)

Sample report (note: values may differ if your data base has different data)

**Table

Description automatically generated**

NOTE:What do we mean by “for that product”?

Each order has multiple products in it. We need to compare each product in an order with the average order quantity for that product (Across all orders) and display the **order number, product code**,**and quantity ordered if the condition is met (i.e., quantity ordered for a product is smaller than the average order quantity for that product)**

Graphical user interface

Description automatically generated

**Graphical user interface, text, application

Description automatically generated**

Q6. Write an SQL query to list customer name, customer number, total amount payable and credit limit for all the customers who have an amount payable greater than their credit limit.

Hint:

1. amount that had to be paid (i.e., toBePAid) can be calculated by adding quantityOrdered\*priceEach for each product in each order for a customer.
2. Totalpaid = sum (amount) from payments for that customer
3. Amount payable = toBePAid – Totalpaid
4. creditLimit is in the customer table.

Sample report (note: values may differ if your data base has different data)

Table

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Q7. Add a “creditworthiness” attribute to the customers column with default value “good”

For all the customers in the result of Q7, change their creditworthiness to “bad”

Hint: This query mostly uses the last query, Q6.

Graphical user interface, text, application, email

Description automatically generated

Q8. Write an SQL query to find the middle class in employee (who reports to someone also being reported to others). Display the name, job title, the number of employees working under him/her, and employee ID of employee working above him/her (i.e., his/her manager’s id)

Graphical user interface, text, application, email

Description automatically generated

Q9: Let’s say the company plans to cut one product line, according to the total sales (i.e., line which has lowest sales). Write an SQL query to find which one should be cut? (Show the name of product line and total sales)

A screenshot of a computer

Description automatically generated with medium confidence

Q10: *Background information: Recommender system aimed at suggesting relevant items to users. It is widely used in many areas like eCommerce, social media, or music Platform. One popular method to build recommender system is “collaborative filtering”. The core idea of collaborating filtering is recommended items that a user might like on the basis of reactions by similar users.*

Write an SQL query to find the most similar user (and number of same products) for customer 124 according to historical orders. In other words, for customer 124, we need to find which customer has most matched purchased items. Please note that in the process of calculating similarity, products should be distinct (one customer may purchase same product multiple times or multiple quantities of a product).

For example:

For customer 124, he/she purchased item: A, B, C.

|  |  |  |
| --- | --- | --- |
| Customer Number | Purchased | Same Purchased (Distinct) |
| 101 | A,C,D,E,A,C,F,G | 2 (A and C) |
| 102 | A,B,F,G,H,B | 2 (A and B) |
| 103 | A,B,C,T,Y,U,I,O | 3 (A, B and C) |

In this example, the most similar customer with customer 124 is customer 103 and number of same products is 3.

Graphical user interface, text, application, email

Description automatically generated

|  |
| --- |
| Optional questions (not to be submitted as part of this assignment):  Do you find any potential issues to calculate the similarities between users with this method? List and explain these issues.  Can you come with a better way to calculate the similarity between users? Please try to write a SQL query for your ideas.  **NOTE: After the due date for this HW, you can share your ideas with the class on MS teams under homework channel.** |