

Principle of DBMS Fall 2023 COP 5725

Midterm Exam

Part I (50 points)

Design a simple database for a Car Rental Company. The main purpose of the system is to keep track of which car is rented by which customer.

There are **six entities** in this database (DO NOT add any new entities to this database):

- **Store**
 - The company has many stores. Each store has their own stock of cars for rent.
- **Car**
 - A car belongs to a single store.
- **Customer**
 - A customer needs to register with the company only once. After that he/she can rent the car from any store.
 - The database should keep a record of when and where (in which store) the customer was registered.
- **TimeSlot**
 - To make it simple, let's assume:
 - A car can only be rented on a daily basis. For example, a customer can only rent a car for a whole day on 7/20/2023.
 - The **Date** column in the Timeslot table is stored as an integer like this: 20230720.
- **RentalTransaction**
 - This is the core table that keeps track of which car is rented by which customer for which timeslot.
- **RentalLog**
 - This table keeps a permanent record of which customer rented which car at what timeslot.
 - The logs must remain valid even after the related records have been updated or deleted. For example, after a customer update his/her phone number, the table should still be able to know the customer's phone number at time of the rental.

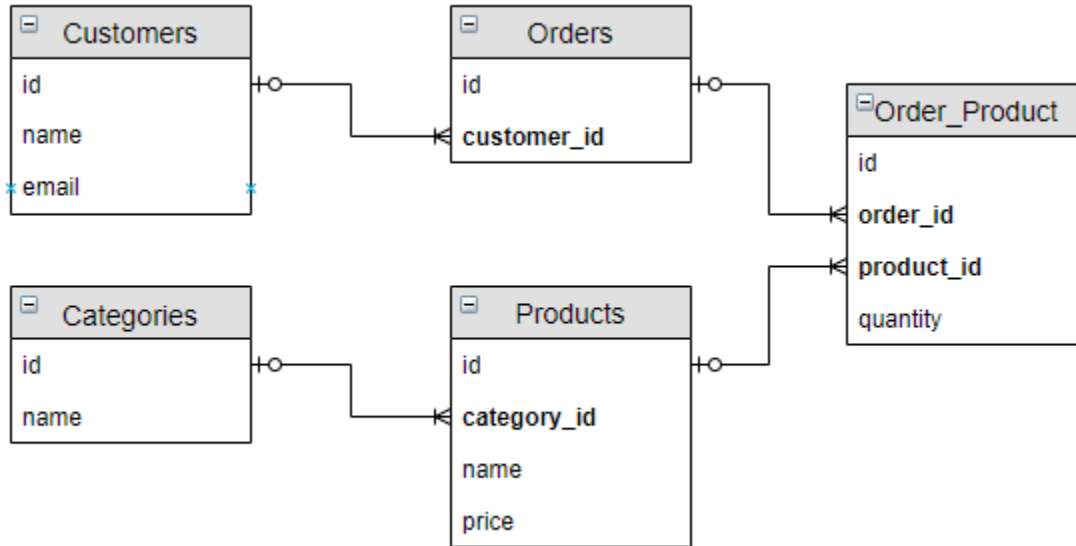
1. [25 points] Design a conceptual schema. The schema should **only contain the six entities** listed above.

2. [25 points] Convert the schema you designed in (1) to a relational schema.

Notes: When drawing the diagram, please follow the notation style in Course Project 1 Solution.

Part II (50 points, each problem is worth 5 points)

Write SQL queries for the following database:



1. Display all customers whose email **is null** or **contains keyword** 'fiu'
2. Display the **number** of orders placed by all customers whose name **starts with** 'J'.
3. Insert a new record (name = 'John', email = 'abc@fiu.edu') into the **Customers** table.
Assume Customers.id will be automatically assigned by the database.
4. Create the Order_Product table as follows:
 - **id** is the primary key and its data type is **serial**
 - **quantity** is an integer which **doesn't allow null** and has a **default value of 0**
 - **order_id** and **product_id** are foreign keys and will be **set to null on delete**.

5. Display the average price of all the products. For example, there are only three products in the database with a price of 7, 8 and 9. Then the average price would be 8.

6. Display the calculated total amount for Order 103 (Orders.id = 103). How to calculate the total amount: for example, if an order includes 1 iPhone and 3 iPads, then the total for that order would be $1 * \text{iPhone_price} + 3 * \text{iPad_price}$.

7. Display all customers with the number of orders they placed. Note that customers with no purchase also need to be listed. An example output:

customer_name	number_of_orders
John	7
Anna	0
Mia	3

8. Display all orders with their calculated total amount. For example, an order purchased 1 iPhone and 3 iPads, then the total for that order would be $1 * \text{iPhone_price} + 3 * \text{iPad_price}$. An example output:

order_id	total_amount
1	7.5
2	8.5
3	10

9. Display the product with the most **total sales amount** (sale amount = price*quantity, **total sales amount** = sum of sale amount in all the orders). For example, if iPhone12 (price = 10) was sold in two orders for amount of 3 and 1 respectively, then the total sales amount = $10 * 3 + 10 * 1 = 40$. If there is a tie, display all of them. An example output (there is a tie in this example):

product_name	total_sales_amount
iPhone 12	30000
iPhone 13	30000

10. Display all categories and the most expensive product in them. If multiple products in the same category tie as the most expensive product in that category, display all of them (see iPhone 12 and iPhone 13 in the example below). Note that categories with no products also need to be listed, with null product columns. An example output:

category_name	most_expensive_product_name	price
Books	The Wealth of Nations	99.48
Phones	iPhone 12	1199.00
Phones	iPhone 13	1199.00
Clothes	<i>NULL</i>	<i>NULL</i>