



Assignment Code: DA-AG-014

Introduction to SQL and Advanced Functions | Assignment

Instructions:

Carefully read each question before attempting. Use Google Docs, Microsoft Word, or a similar tool to type out each theoretical question along with its answer. For practical questions, use **SQL Workbench** (or your designated SQL tool) to complete the required tasks. Once you have finished, save both the Word/Docs file and SQL File as PDF documents. Please ensure that you do not zip or archive the files before uploading. Submit the PDF files directly to the LMS or as instructed by your teacher. Each question carries 20 marks.

Total Marks: 200

Question 1 : Explain the fundamental differences between DDL, DML, and DQL commands in SQL. Provide one example for each type of command.

Answer :



Question 2 : What is the purpose of SQL constraints? Name and describe three common types of constraints, providing a simple scenario where each would be useful.

Answer :

Question 3 : Explain the difference between `LIMIT` and `OFFSET` clauses in SQL. How would you use them together to retrieve the third page of results, assuming each page has 10 records?

Answer :



Question 4 : What is a Common Table Expression (CTE) in SQL, and what are its main benefits? Provide a simple SQL example demonstrating its usage.

Answer :

Question 5 : Describe the concept of SQL Normalization and its primary goals. Briefly explain the first three normal forms (1NF, 2NF, 3NF).

Answer :



Question 6 : Create a database named **ECommerceDB** and perform the following tasks:

1. Create the following tables with appropriate data types and constraints:
 - Categories
 - CategoryID (INT, PRIMARY KEY)
 - CategoryName (VARCHAR(50), NOT NULL, UNIQUE)
 - Products
 - ProductID (INT, PRIMARY KEY)
 - ProductName (VARCHAR(100), NOT NULL, UNIQUE)
 - CategoryID (INT, FOREIGN KEY → Categories)
 - Price (DECIMAL(10,2), NOT NULL)
 - StockQuantity (INT)
 - Customers
 - CustomerID (INT, PRIMARY KEY)
 - CustomerName (VARCHAR(100), NOT NULL)
 - Email (VARCHAR(100), UNIQUE)
 - JoinDate (DATE)
 - Orders
 - OrderID (INT, PRIMARY KEY)
 - CustomerID (INT, FOREIGN KEY → Customers)
 - OrderDate (DATE, NOT NULL)
 - TotalAmount (DECIMAL(10,2))
2. Insert the following records into each table

- Categories

| CategoryID | Category Name |
|------------|---------------|
| 1 | Electronics |
| 2 | Books |
| 3 | Home Goods |
| 4 | Apparel |



- Products

| ProductID | ProductName | CategoryID | Price | StockQuantity |
|-----------|-----------------------|------------|---------|---------------|
| 101 | Laptop Pro | 1 | 1200.00 | 50 |
| 102 | SQL Handbook | 2 | 45.50 | 200 |
| 103 | Smart Speaker | 1 | 99.99 | 150 |
| 104 | Coffee Maker | 3 | 75.00 | 80 |
| 105 | Novel : The Great SQL | 2 | 25.00 | 120 |
| 106 | Wireless Earbuds | 1 | 150.00 | 100 |
| 107 | Blender X | 3 | 120.00 | 60 |
| 108 | T-Shirt Casual | 4 | 20.00 | 300 |

- Customers

| CustomerID | CustomerName | Email | Joining Date |
|------------|------------------|---------------------|--------------|
| 1 | Alice Wonderland | alice@example.com | 2023-01-10 |
| 2 | Bob the Builder | bob@example.com | 2022-11-25 |
| 3 | Charlie Chaplin | charlie@example.com | 2023-03-01 |
| 4 | Diana Prince | diana@example.com | 2021-04-26 |



- Orders

| OrderID | CustomerID | OrderDate | TotalAmount |
|---------|------------|-----------|-------------|
| | | | |

| | | | |
|------|---|------------|---------|
| 1001 | 1 | 2023-04-26 | 1245.50 |
| 1002 | 2 | 2023-10-12 | 99.99 |
| 1003 | 1 | 2023-07-01 | 145.00 |
| 1004 | 3 | 2023-01-14 | 150.00 |
| 1005 | 2 | 2023-09-24 | 120.00 |
| 1006 | 1 | 2023-06-19 | 20.00 |

Answer :

Question 7 : Generate a report showing `CustomerName`, `Email`, and the `TotalNumberofOrders` for each customer. Include customers who have not placed any orders, in which case their `TotalNumberofOrders` should be 0. Order the results by `CustomerName`.

Answer :



Question 8 : Retrieve Product Information with Category: Write a SQL query to display the **ProductName**, **Price**, **StockQuantity**, and **CategoryName** for all products. Order the results by **CategoryName** and then **ProductName** alphabetically.

Answer :



Question 9 : Write a SQL query that uses a Common Table Expression (CTE) and a Window Function (specifically `ROW_NUMBER()` or `RANK()`) to display the `CategoryName`, `ProductName`, and `Price` for the top 2 most expensive products in each `CategoryName`.

Answer :

Question 10 : You are hired as a data analyst by Sakila Video Rentals, a global movie rental company. The management team is looking to improve decision-making by analyzing existing customer, rental, and inventory data.

Using the Sakila database, answer the following business questions to support key strategic initiatives.

Tasks & Questions:

1. Identify the top 5 customers based on the total amount they've spent. Include customer name, email, and total amount spent.
2. Which **3 movie categories** have the **highest rental counts**? Display the category name and number of times movies from that category were rented.
3. Calculate how many films are available at each store and how many of those have **never been rented**.
4. Show the **total revenue per month** for the year 2023 to analyze business seasonality.
5. Identify customers who have rented **more than 10 times** in the last 6 months.

