EIU
TRƯỜNG ĐẠI HỌC QUỐC TẾ MIỀN ĐÔNG
EASTERN INTERNATIONAL UNIVERSITY

Review Content
Quarter**:** /Academic year**:**
School of Computing and Information Technology.

Course code: **CSE 301**
Course name: **Introduction to Databases** Duration: **90 minutes**

Instructions:

- *Non-programming calculator: Yes ☑ | No ☐*
- *This paper consists of 2 sections. This paper carries 100 marks*

  Section 1 (40 marks) includes 3 essay questions.

  Section 2 (60 marks) includes 2 practices on computers.

Section 1: Essay questions: The student answers right below the question.

Given an e-commerce system database schema below:

Customers(CustomerID, Name, Email, Phone)

Description: This table contains information about customers including their unique identifier (CustomerID), customer name (Name) and their contact information such as email and phone number.

Products(ProductID, Name, CategoryID, Price, Stock_Quantity)

Description: This table contains information about the products on the ecommerce system including their unique identifier (ProductID), product name(Name), category identifier (CategoryID), price and their stock quantity (Stock_Quantity).

Orders(OrderID, CustomerID, OrderDate, Shipping_Address)

Description: This table contains information about the order including their unique identifier (OrderID), customer identifier(CustomerID), date of order (Orderdate) and the shipping address (Shipping_Address).

OrderItems(OrderItemID, OrderID, ProductID, Quantity)

Description: This table contains information about the order item for each product ordered in the order including their unique identifier (OrderItemID), order identifier(OrderID), product identifier(ProductID), quantity of order product(Quantity).

Categories(CategoryID, Name, Description)

Description: This table contains information about product category including their unique identifier (CategoryID), category name (Name) and their category description (Description).

Question 1(10 marks): Find primary key and foreign key of ecommerce system database schema

Table Name	Primary Key	Foreign Key
1. **Customers**		
2. **Products**		
3. **Orders**		
4. **OrderItems**		
5. **Category**		

Question 2 (20 marks) Relational Algebra

a) Find the name of products that have price greater than 100000.

.....

.....

b) Find the name and stock quantity of products that their category is “Electronics”.

.....

.....

c) Find the name of the customer and total payment of their order.

.....

.....

.....

.....

d) Find the name of product and their category name that has highest selling in quantity.

.....

.....

.....

.....

Section 2: Practice on computer

Given a University Library Management System database schema as below:

Problem Statement:
The library of a university needs a database management system to efficiently handle its operations and meet the following requirements:

1. **Books Management:**
   - Each book has a unique identifier (BookID), title, author, genre, publication year, and the total number of copies available in the library.
   - Each book belongs to a specific category (e.g., textbook, reference book, research material).
2. **Readers Management:**
   - The library caters to **students**, **faculty members**, and **staff** as its readers. Each reader has a unique identifier (ReaderID), full name, date of birth, university ID, department, and contact details.
   - Readers are classified into types (e.g., "Student," "Faculty," "Staff") with different borrowing limits and durations.
3. **Borrowing/Returning Transactions:**
   - Each borrowing transaction records the **book borrowed**, the **reader borrowing the book**, the **staff member processing the transaction**, and details such as borrow date, due date, and return date.
   - The system must track the status of the transaction (e.g., "Returned," "Overdue").
4. **University Staff Management:**
   - Library staff handle operations such as issuing and receiving books. Each staff member has a unique identifier (StaffID), name, role, and work schedule.
5. **Reservation System:**
   - Readers can reserve books that are currently unavailable.
   - Each reservation records details such as reservation ID, book reserved, reader reserving the book, the reservation date, and its status (e.g., "Pending," "Completed").
   - Reservations are processed on a first-come, first-served basis, ensuring fairness in book allocation.

Question 1 (20 marks) Using **draw.io** to design an ER - Diagram and Identify Entities with attributes and the relationships between entities in the above schema management. (Export image from draw.io then insert on file).

Question 2 (40 marks) Practice on My SQL Workbench

A. Constraint

1. Restricts transaction statuses to predefined values as follows: *Returned, Not Returned, Overdue*
2. Ensures reservation dates are not in the future.

B. SQL Query

3. Retrieve all book reservations that are currently in the status 'Pending' and were made within the last 7 days.
4. Find any borrowed book exceeds the maximum loan duration described as below:
    - Students can borrow up to 3 books at a time for a maximum of 14 days.
    - Faculty can borrow up to 5 books at a time for a maximum of 30 days.
    - Staff can borrow up to 2 books at a time for a maximum of 7 days.
5. Retrieve Total Borrowed Books by Reader.
6. Update the status of all book reservations to 'Completed' if the current status is *'Pending*' and the reservation date is more than 7 days old. This ensures that only reservations that are overdue for completion are updated, while those with a recent reservation date or already completed status remain unchanged.
7. Create a procedure to check if a specific book is available for borrowing. The procedure takes the book's ID as input and verifies whether the number of available copies (*QuantityAvailable*) is greater than zero. It returns a status (*Allowed or Not Allowed*) and a message explaining the result, such as whether borrowing is permitted or if the book is currently unavailable. This simple procedure focuses solely on book availability and provides clear output for decision-making.

------ *End* ------