Assignment 1

Deadline: Day 02/10/2024 @ 23:59

**[Total Mark for this Assignment is 8]**

***Introduction to Database***

***IT244***

**Instructions:**

* You must submit two separate copies **(one Word file and one PDF file)** using the Assignment Template on Blackboard via the allocated folder. These files **must not be in compressed format**.
* It is your responsibility to check and make sure that you have uploaded both the correct files.
* Zero mark will be given if you try to bypass the SafeAssign (e.g. misspell words, remove spaces between words, hide characters, use different character sets, convert text into image or languages other than English or any kind of manipulation).
* Email submission will not be accepted.
* You are advised to make your work clear and well-presented. This includes filling your information on the cover page.
* You must use this template, failing which will result in zero mark.
* You MUST show all your work, and text must not be converted into an image, unless specified otherwise by the question.
* Late submission will result in ZERO mark.
* The work should be your own, copying from students or other resources will result in ZERO mark.
* Use **Times New Roman** font for all your answers.

Student Details:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Name:** ###  **CRN:** ### |  | **ID:** ### |
|  |  |  |

# Question One

***2 Marks***

*Learning Outcome(s):*

*CLO 1 Explain the concepts and architectures involved in the database development.*

Using your own world, differentiate between database schema and database state with examples.

|  |  |  |
| --- | --- | --- |
| Feature | Database Schema | Database State |
| **Definition** | The structure or blueprint of the database. | The actual data contained within the database. |
| **Nature** | Static and changes infrequently. | Dynamic and changes frequently. |
| **Components** | Defines tables, columns, data types, keys, and constraints. | Contains actual records/entries in the tables. |
| **Example** | **Table: Students** Columns: student\_id, name, age, major | **Table: Students** Records: (101, Ahmed, 20, Computer Science) (102, Alli, 22, Biology) |
| **Purpose** | To outline how data is organized and related. | To represent the current content of the database. |
| **Modification** | Requires database migration or updates to change. | Changes with every insert, update, or delete operation. |
| **Representation** | Defines relationships between tables (e.g., foreign keys). | Represents current relationships through actual data (e.g., student enrollments in courses). |

# Question Two

***2 Marks***

*Learning Outcome(s): CLO3*

Create Entity-Relationship model, Relational model, and write SQL queries

The advent of online shopping has revolutionized the traditional shopping experience, offering convenience to consumers worldwide. However, have you ever pondered the intricate processes that unfold once you confirm your purchase? Beyond the seamless transaction lies a sophisticated infrastructure comprising databases, servers, and applications working in unison.

Provide the relational database schema for an online shopping system where customers place orders. An order may consist of multiple products, with products categorized for easy organization and search. The Customer table identifies the customers placing orders with their personal information, while the Category table specifies the product categories and description. The Product table lists the available items for purchase and their prices, each linked to a category through the Category ID. Orders, recorded in the Order table with date of order and the total amount, and they are associated with customers through the Customer ID. The Order Detail table captures the intricate relationship between orders and products, detailing the products included in an order along with their respective quantities.

**1. Customer Table**

* **Table Name**: Customer
* **Description**: Stores information about customers.

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| CustomerID | INT (Primary Key, Auto Increment) | Unique identifier for each customer |
| FirstName | VARCHAR(50) | Customer's first name |
| LastName | VARCHAR(50) | Customer's last name |
| Email | VARCHAR(100) | Customer's email address |
| Phone | VARCHAR(15) | Customer's phone number |
| Address | VARCHAR(255) | Customer's address |
| City | VARCHAR(50) | City of the customer |
| State | VARCHAR(50) | State of the customer |
| ZipCode | VARCHAR(10) | Zip code of the customer |

**2. Category Table**

* **Table Name**: Category
* **Description**: Stores product categories.

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| CategoryID | INT (Primary Key, Auto Increment) | Unique identifier for each category |
| CategoryName | VARCHAR(100) | Name of the product category |
| Description | TEXT | Description of the category |

**3. Product Table**

* **Table Name**: Product
* **Description**: Stores product information.

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| ProductID | INT (Primary Key, Auto Increment) | Unique identifier for each product |
| ProductName | VARCHAR(100) | Name of the product |
| Price | DECIMAL(10, 2) | Price of the product |
| CategoryID | INT (Foreign Key) | ID of the category (links to Category) |
| StockQuantity | INT | Available quantity of the product |

**4. Order Table**

* **Table Name**: Orders
* **Description**: Stores order information.

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| OrderID | INT (Primary Key, Auto Increment) | Unique identifier for each order |
| CustomerID | INT (Foreign Key) | ID of the customer (links to Customer) |
| OrderDate | DATETIME | Date and time when the order was placed |
| TotalAmount | DECIMAL(10, 2) | Total amount of the order |

**5. Order Detail Table**

* **Table Name**: OrderDetail
* **Description**: Captures the details of products within each order.

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| OrderDetailID | INT (Primary Key, Auto Increment) | Unique identifier for each order detail |
| OrderID | INT (Foreign Key) | ID of the order (links to Orders) |
| ProductID | INT (Foreign Key) | ID of the product (links to Product) |
| Quantity | INT | Number of units of the product ordered |
| PriceAtOrder | DECIMAL(10, 2) | Price of the product at the time of order |

**Relationships**

* The Customer table is linked to the Orders table through CustomerID.
* The Category table is linked to the Product table through CategoryID.
* The Product table is linked to the OrderDetail table through ProductID.
* The Orders table is linked to the OrderDetail table through OrderID.

# Question Three

***2 Marks***

*Learning Outcome(s):*

*CLO3: Create Entity-Relationship model, Relational model, and write SQL queries.*

Compare and contrast the effects of specialization and generalization on the hierarchy of Enhanced Entity-Relationship (EER) diagrams. Give examples to illustrate how generalization and specialization differ.

**Specialization**

**Specialization** is the process of defining a new, more specific entity from a more general entity. It creates a hierarchical structure where the superclass is divided into sub-entities (subclasses), each inheriting attributes and relationships of the superclass but also possessing specific attributes or relationships of their own.

* **Top-down approach**: Specialization starts from a general entity (superclass) and divides it into specialized entities (subclasses) based on some distinguishing characteristic.
* **Effects on hierarchy**: Specialization adds levels of hierarchy by introducing subclasses to represent more detailed entity types. It results in a **narrow and deeper** hierarchy because each specialized subclass is placed below the general superclass.

**Example of Specialization:**

Imagine an entity called **Person**. Specialization could divide this entity into more specific entities, such as **Student** and **Professor**. Both inherit general attributes of Person (like name, date of birth), but each subclass could have additional specific attributes, such as:

* **Student**: studentID, course\_enrolled
* **Professor**: professorID, department

In this case, **Person** is the superclass, and **Student** and **Professor** are subclasses created by specialization.

**Generalization**

**Generalization** is the reverse of specialization. It involves identifying common features among several distinct entities and abstracting them into a single, generalized entity. Multiple entities are generalized into a higher-level entity.

* **Bottom-up approach**: Generalization starts with distinct entities (subclasses) and groups them into a common superclass that contains their shared characteristics.
* **Effects on hierarchy**: Generalization leads to a **broader and shallower** hierarchy, as multiple entities are merged into one general entity. This reduces the number of distinct entities at the lower level.

**Example of Generalization:**

Consider two entities, **Car** and **Motorcycle**. By generalization, they can be combined into a more general entity, **Vehicle**, which inherits shared attributes like make, model, and year, while keeping specific attributes for the subclasses:

* **Car**: number\_of\_doors
* **Motorcycle**: type\_of\_handlebars

In this example, **Vehicle** is the generalized superclass, and **Car** and **Motorcycle** are the subclasses that existed before the generalization.

**Key Differences**

|  |  |  |
| --- | --- | --- |
| Aspect | Specialization | Generalization |
| Process direction | Top-down (from general entity to specific sub-entities) | Bottom-up (from specific entities to a general entity) |
| Hierarchy structure | Leads to a narrow, deeper hierarchy with more specific subclasses | Leads to a broader, shallower hierarchy by combining entities |
| Purpose | To distinguish specific entities from a general entity | To combine similar entities into a common superclass |
| Example | **Person** → **Student**, **Professor** | **Car**, **Motorcycle** → **Vehicle** |
| Focus | On creating more detailed entities with specific characteristics | On simplifying and reducing redundancy in the entity model |

# Question Four

***2 Marks***

*Learning Outcome(s):*

CLO 3

Instructors: Create Entity-Relationship model, Relational model, and write SQL queries

Draw an ER diagram for the following SPL Saudi Post system.

SPL application relies on a company-wide information system. In SPL Saudi Post system, shipped items are the main component of the SPL product tracking information system. Shipped items can be characterized by item number (unique), weight, dimensions, insurance amount, destination, and final delivery date. Shipped items are received into the SPL system at a single retail center. Retail centers are characterized by their type, unique ID, and address. Shipped items make their way to their destination via one or more standard SPL transportation events (flights, truck deliveries). These transportation events are characterized by a unique schedule Number, a type (flight, truck), and a delivery Route.

