



# Capital University of Science & Technology (CUST)

---

**NAME:** RANA HAMMAD MUSHTAQ

**REG NO:** BSE233041

**GROUP MEMBERS:**

1. MUHAMMAD ABDULLAH KHALID-(BSE233051)
2. MUHAMMAD KHAN (BSE233019)
3. MUHAMMAD ASIF-(BSE233078)

**DATE:** 12-25-2024

**SECTION:** 2

**COURSE:** INTRO TO DATABASE SYSTEM

**TOPIC:** ONLINE BOOKSTORE DATABASE

**SUMBITTIED TO:** MAAM HINA RASHID

**GITHUB LINK :** <https://github.com/RanaHammadMushtaq>

## Entities and Attributes:

1. **Customer:** *Customer\_ID*, name, email, phone number, preferences.
2. **Order:** *Order\_ID*, order date, total amount, order status, payment method.
3. **Review:** *Review\_ID*, rating, comment, customer ID, book ID.
4. **Book:** *Book\_ID*, title, genre, price, stock quantity, ISBN.
5. **Author:** *Author\_ID*, name, biography, birthdate.
6. **Publisher:** *Publisher\_ID*, name, contact info.
7. **Order\_Item:** *Order\_Item\_ID*, quantity, price, order ID, book ID.
8. **Shipment:** *Shipment\_ID*, tracking ID, shipment date, status, order ID.
9. **Warehouse:** *Warehouse\_ID*, location, manager, capacity.

## Relationships:

1. **Customer - Order**
  - Relationship Name: *Places\_Order*
  - Degree: 2 (binary relationship)
  - Cardinality: *One-to-Many* (One customer can place multiple orders, but each order is placed by one customer)
2. **Customer - Review**
  - Relationship Name: *Writes\_Review*
  - Degree: 2
  - Cardinality: *One-to-Many* (One customer can write multiple reviews, but each review is written by one customer)
3. **Order - Order\_Item**
  - Relationship Name: *Has\_Items*
  - Degree: 2

- Cardinality: *One-to-Many* (One order can have multiple items, but each item belongs to one order)

#### 4. **Order\_Item - Book**

- Relationship Name: *Contains\_Book*
- Degree: 2
- Cardinality: *Many-to-One* (One order item represents one book, but each book can appear in multiple order items)

#### 5. **Author - Book**

- Relationship Name: *Writes\_Book*
- Degree: 2
- Cardinality: *One-to-Many* (One author can write multiple books, but each book is written by one author)

#### 6. **Publisher - Book**

- Relationship Name: *Publishes\_Book*
- Degree: 2
- Cardinality: *One-to-Many* (One publisher can publish multiple books, but each book is published by one publisher)

#### 7. **Book - Review**

- Relationship Name: *Has\_Review*
- Degree: 2
- Cardinality: *One-to-Many* (One book can have multiple reviews, but each review is for one specific book)

#### 8. **Order - Shipment**

- Relationship Name: *Has\_Shipment*
- Degree: 2
- Cardinality: *One-to-One* (Each order has one shipment associated with it, and each shipment is linked to one order)

#### 9. **Warehouse - Book**

- Relationship Name: *Stores\_Book*
- Degree: 2

- Cardinality: *One-to-Many* (One warehouse can store multiple books, but each book is associated with a specific warehouse)

## 2. Challenges and Considerations

- **Handling Many-to-Many Relationships:**
  - Many-to-Many relationships (e.g., Book-Author, Book-Warehouse) required introducing junction tables to normalize the database and maintain integrity.
- **Ensuring Data Integrity:**
  - Using foreign key constraints to enforce relationships and avoid orphan records.

## 3.

### CUSTOMER TABLE:

Field Name	Data Type	Constraints
Customer_ID	INT	PRIMARY KEY
Customer_Name	VARCHAR(100)	NOT NULL
Customer_Email	VARCHAR(100)	UNIQUE, NOT NULL
Customer_Phone	VARCHAR(15)	UNIQUE
Preferences	TEXT	
Street	VARCHAR(100)	
City	VARCHAR(50)	
Postcode	VARCHAR(10)	

### Order Table

Field Name	Data Type	Constraints
Order_ID	INT	PRIMARY KEY
Customer_ID	INT	FOREIGN KEY
Order_Date	DATE	NOT NULL
Total_Amount	DECIMAL(10,2)	NOT NULL
Order_Status	VARCHAR(50)	
Payment_Method	VARCHAR(50)	

### Order Items Table

Field Name	Data Type	Constraints
OrderItem_ID	INT	PRIMARY KEY
Quantity	INT	NOT NULL
Price	DECIMAL(10,2)	NOT NULL
Order_ID	INT	FOREIGN KEY
Book_ID	INT	FOREIGN KEY

### Shipment Table

Field Name	Data Type	Constraints
Shipment_ID	INT	PRIMARY KEY
Tracking_ID	VARCHAR(50)	UNIQUE
Shipment_Date	DATE	
Delivery_Date	DATE	
Status	VARCHAR(50)	
Order_ID	INT	FOREIGN KEY

### Book Table

Field Name	Data Type	Constraints
Book_ID	INT	PRIMARY KEY
Title	VARCHAR(200)	NOT NULL
Genre	VARCHAR(50)	
Price	DECIMAL(10,2)	NOT NULL
Stock_Quantity	INT	NOT NULL
ISBN	VARCHAR(20)	UNIQUE

### Author Table

Field Name	Data Type	Constraints
Author_ID	INT	PRIMARY KEY
Name	VARCHAR(100)	NOT NULL
Biography	TEXT	
Birth_Date	DATE	
Nationality	VARCHAR(50)	
Gender	VARCHAR(10)	

### Publisher Table

Field Name	Data Type	Constraints
Publisher_ID	INT	PRIMARY KEY
Name	VARCHAR(100)	NOT NULL
Contact_Info	TEXT	
Street	VARCHAR(100)	
City	VARCHAR(50)	
Postcode	VARCHAR(10)	
Established_Year	INT	

Website	VARCHAR(100)	
---------	--------------	--

### Review Table

Field Name	Data Type	Constraints
Review_ID	INT	PRIMARY KEY
Rating	INT	CHECK (Rating BETWEEN 1 AND 5)
Comment	TEXT	
Date	DATE	
Customer_ID	INT	FOREIGN KEY
Book_ID	INT	FOREIGN KEY

### Warehouse Table

Field Name	Data Type	Constraints
Warehouse_ID	INT	PRIMARY KEY
Location	VARCHAR(100)	NOT NULL
Manager	VARCHAR(100)	
Contact_Info	VARCHAR(100)	
Capacity	INT	NOT NULL

## 4. Establishing Relationships

1. **One-to-Many**: Relationships like **Customer-Order** and **Order-OrderItems** are implemented using foreign keys (e.g., **Customer\_ID** in **Order** table).
2. **Many-to-Many**: Relationships like **Book-Author** require a junction table:
  - **BookAuthor Table**: Attributes: **Book\_ID** (FK), **Author\_ID** (FK).

5.

Day / Date \_\_\_\_\_

**Customer Table**

Customer ID	Customer Name	Customer Email
01	Rana	Sample@gmail.com
02	Khan	Sample@gmail.com

  

Customer Phone	Preferences	Street
912345677	Fiction	123 Islamabad street
+9121345678	Non-fiction	456 Islamabad street

  

City	Post code
Islamabad	5421
Islamabad	1245

**Order Table:**

Order ID	Customer ID	Order Date
101	01	01-12-24
102	02	01-12-24

  

Total amount	Order status	Payment method
1000	Delivered	Credit card
2000	Processing	COD

CUST —



Day/Date

### Order Items

Order item_ID	Quantity	Price
1	1	1000
2	2	2000

Order_ID	Book_ID
101	1
102	2

### Book Table:

Book_ID	Title	Genre	Price
1	The Great books	Fiction	2000
2	History	non-fiction	2500

stock_Quantity	ISBN
10	9721341
20	98223421

### Authors Table:

Author_ID	Name	Biography
1	Scott	Sample
2	Sample	Sample

Birth Date	Nationality	Gender
1896-9-24	British	Male
1942-01-28	American	Male

CUST



Day / Date

### Publisher Table:

Publisher_ID	Name	Contact_info
1	sample	12345
Address	Established_years	Website
New York USA	1935	www.book.com

### Warehouse Table:

Warehouse_ID	Location	Manager
1	Downtown NY	Rana
2	Islamabad	Khan
Contact_info	Capacity	
+9200---7	10000	
+9201---7	5000	

### Shipment Table:

Shipment_ID	Tracking_ID	Shipment_Date
501	T8-02	2024-12-03
Delivery_Date	Status	Order_ID
2024-12-05	Delivered	101

CUST

## Additional Explanation of Design Decisions:

### 1. Entity Identification and Structuring:

- Each entity was carefully identified based on its real-world significance in the book management system. Attributes were chosen to ensure that each entity captures the necessary information without redundancy.
- For example, **Customer** includes fields for both contact information and preferences, supporting both operational needs and marketing analysis.

### 2. Resolving Many-to-Many Relationships:

- Many-to-Many relationships (e.g., between **Book** and **Author**, **Customer** and **Review**, and **Book** and **Warehouse**) were resolved using junction tables. These tables allow for a clean and normalized structure, making the database more maintainable and efficient for complex queries.

### 3. Data Integrity and Validation:

- Primary keys and foreign keys were used rigorously to enforce relationships between entities and prevent orphaned records.
- Constraints such as **NOT NULL**, **UNIQUE**, and **CHECK** (e.g., ensuring **Rating** is between 1 and 5) were added to maintain data accuracy and quality.

### 4. Ease of Querying:

- The relational data model was designed to allow straightforward querying for common business needs:
  - Tracking orders and shipments by customer.
  - Identifying low-stock books from the **Warehouse** table.
  - Retrieving books written by specific authors or published by specific publishers.
  - Analyzing customer reviews and feedback for books.

### 5. Consideration of Real-World Processes:

- Relationships like **Order** to **Shipment** (1:1) and **Order** to **Order Items** (1:N) were designed to mirror real-world workflows. Each order has one associated

shipment but may contain multiple items, reflecting standard e-commerce and logistics processes.

**6. Normalization for Redundancy Reduction:**

- The database was normalized up to at least the third normal form (3NF). For example, instead of storing publisher details redundantly with every book, a separate **Publisher** entity was created with a relationship to **Book**.