

Capital University of Science & Technology (CUST)

**NAME:** MUHAMMAD ASIF

**REG NO:** BSE233078

**GROUP MEMBERS:**

1. RANA HAMMAD MUSHTAQ (BSE233041)
2. MUHAMMAD KHAN (BSE233019)
3. MUHAMMAD ABDULLAH KHALID-(BSE233051)

**DATE:** 12-26-2024

**SECTION:** 2

**COURSE:** INTRO TO DATABASE SYSTEM

**TOPIC:** ONLINE BOOKSTORE DATABASE

**SUMBITTIED TO:** MAAM HINA RASHID

**GITHUB LINK :** https://github.com/masif078

**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 |

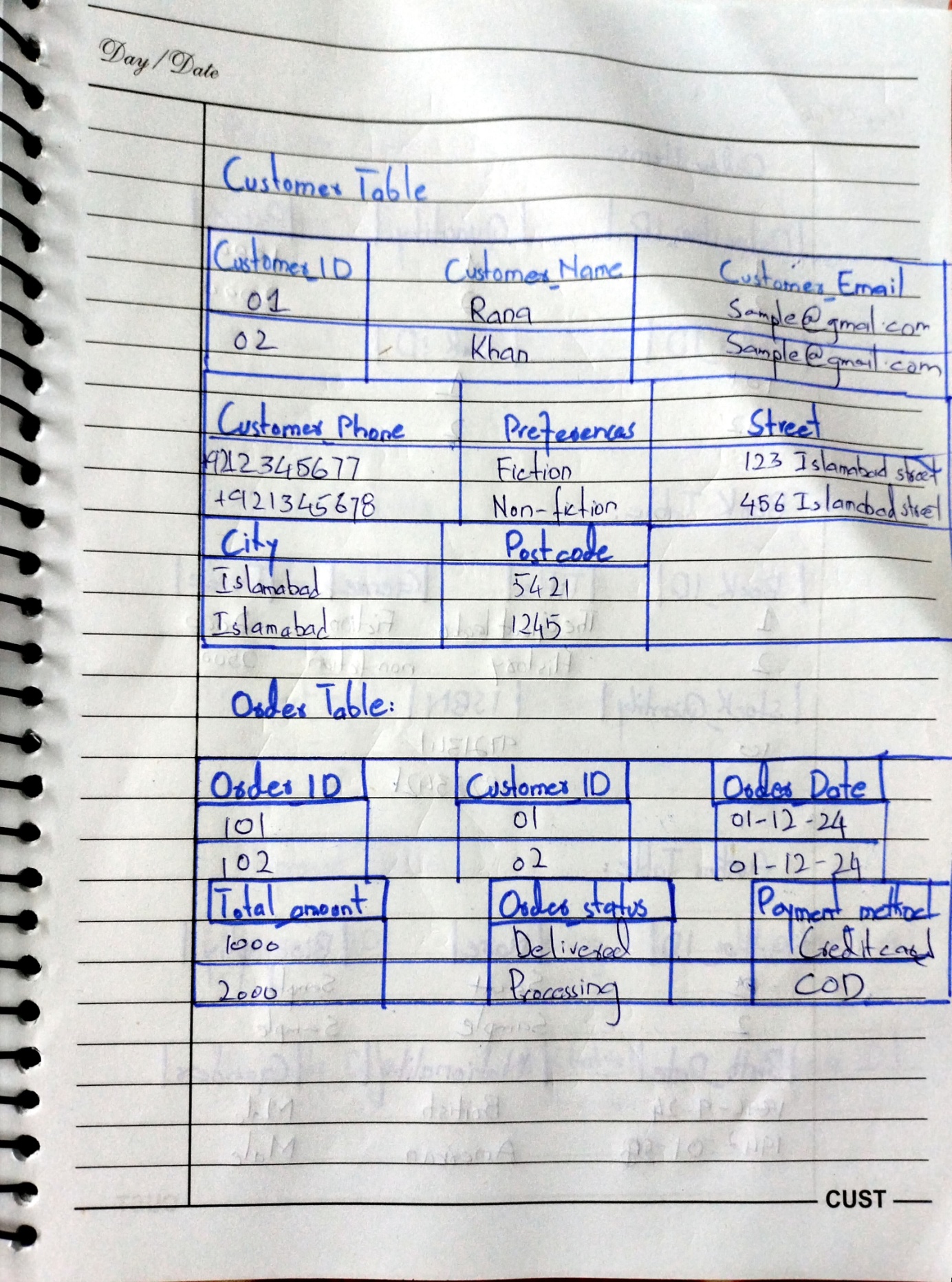
**Warehouse Table**

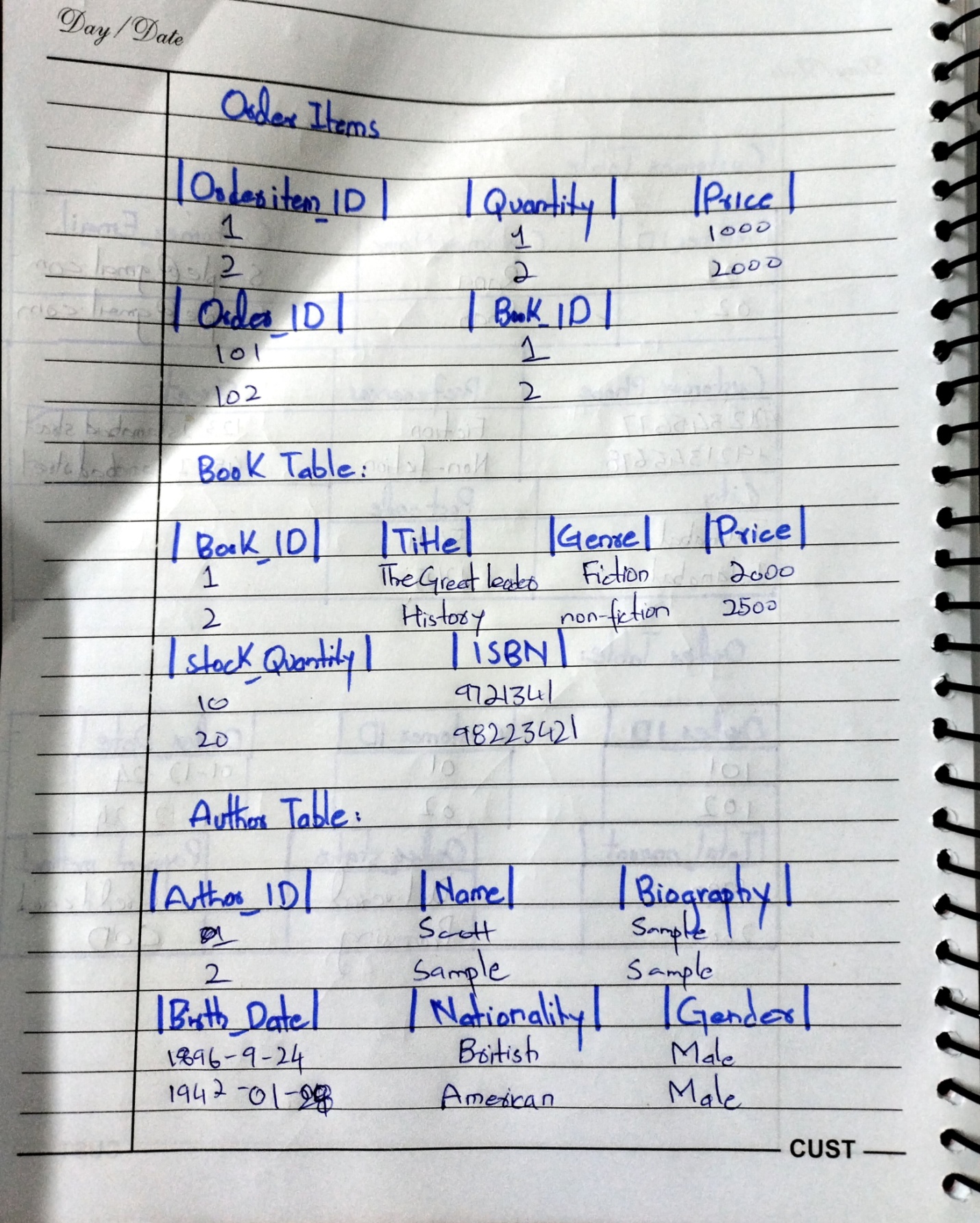
|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Constraints |
| Warehouse\_ID | INT | PRIMARY KEY |
| Location | VARCHAR(100) | NOT NULL |
| Manager | VARCHAR(100) |  |
| Contact\_Info | VARCHAR(100) |  |
| Capacity | INT | NOT NULL |

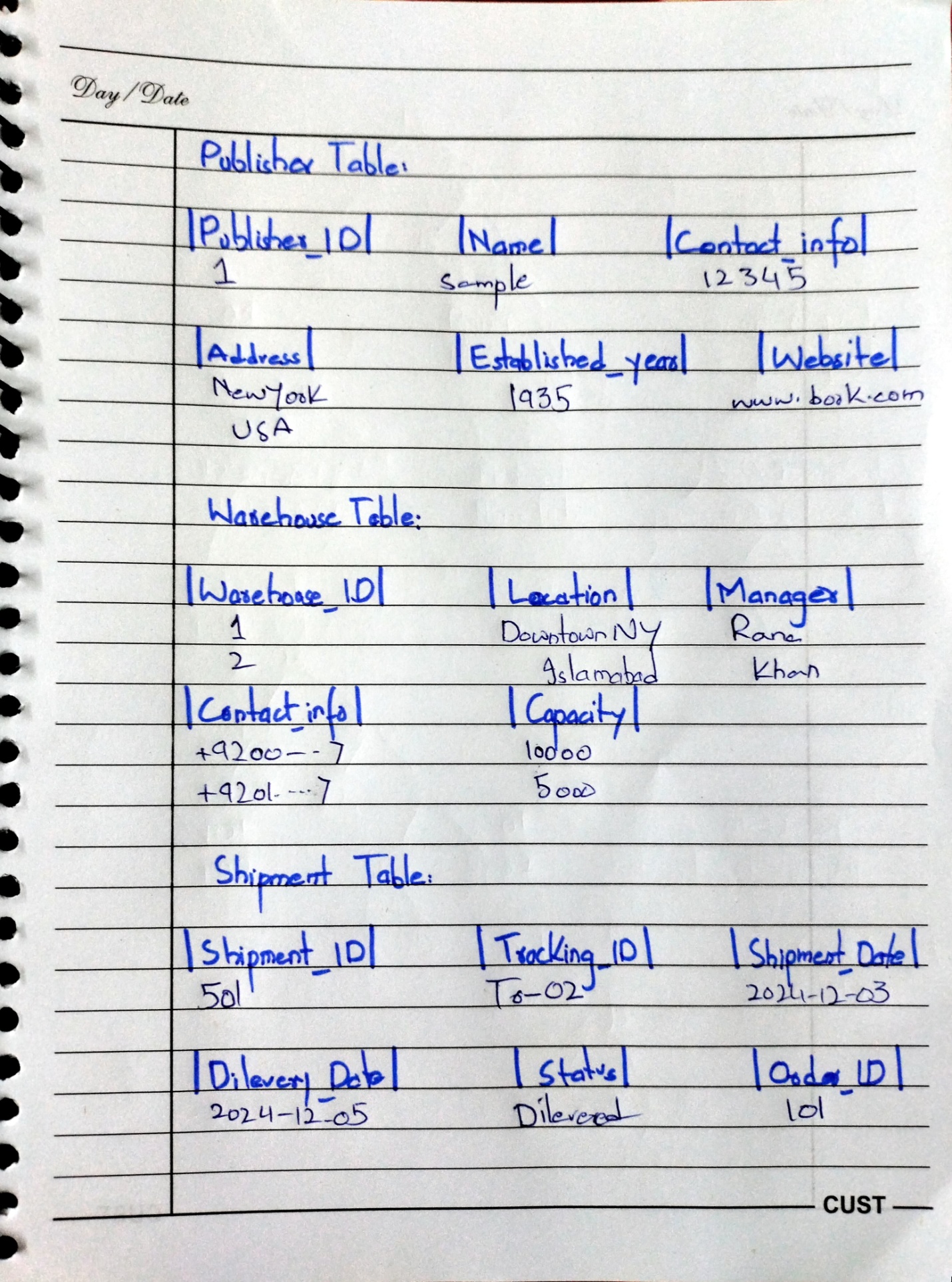
### ****4. Establishing Relationships****

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.**







### 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.**