|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Description** |
| UserID (PK) | Integer | Primary Key - Unique User ID |
| Username | Varchar(50) | Username for login and display |
| Password | Varchar(100) | Securely hashed password |
| Email | Varchar(100) | User's email address |
| First\_Name | Varchar(50) | User's first name |
| Last\_Name | Varchar(50) | User's last name |
| Phone\_Number | Varchar(15) | User's phone number |
| Address | Varchar(255) | User's address |
| Registration\_Date | Date | Date of user registration |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**This table stores information about the users**

**Products Table:**

**This table stores information about the products that users want to list or find.**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Description** |
| ProductID (PK) | Integer | Primary Key - Unique Product ID |
| UserID (FK) | Integer | Foreign Key - ID of the user who listed the product |
| Category | Varchar(50) | Category of the product (e.g., Furniture, Electronics, Clothing, etc.) |
| Title | Varchar(100) | Title or name of the product |
| Description | Text | Detailed description of the product |
| Price | Decimal(10, 2) | Price of the product |
| Location | Varchar(100) | Location where the product is available or to be delivered |
| Status | Enum('Available', 'Sold', 'Reserved', 'Expired') | Status of the product listing |
| Listing\_Date | Date | Date when the product was listed |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Transactions Table:**

**This table tracks transactions between users, such as purchases or reservations of products.**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Description** |
| TransactionID (PK) | Integer | Primary Key - Unique Transaction ID |
| BuyerUserID (FK) | Integer | Foreign Key - ID of the user who is buying or reserving |
| SellerUserID (FK) | Integer | Foreign Key - ID of the user who is selling the produc |
| ProductID (FK) | Integer | Foreign Key - ID of the product being transacted |
| Transaction\_Type | Enum('Purchase', 'Reservation') | Type of transaction |
| Transaction\_Date | Date | Date of the transaction |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Images Table (for storing product images):**

**This table stores references to images associated with product listings.**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Description** |
| ImageID (PK) | Integer | Primary Key - Unique Image ID |
| ProductID (FK) | Integer | Foreign Key - ID of the associated product |
| Image\_Path | Varchar(255) | Path to the image file on the server |
|  |  |  |
|  |  |  |

**Orders Table:**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Description** |
| OrderID (PK) | Integer | Primary Key - Unique Order ID |
| BuyerID (FK) | Integer | Foreign Key - ID of the buyer (references Users table) |
| Date\_Placed | DateTime | Date and time when the order was placed |
| Total\_Amount | Decimal(10, 2) | Total amount for the order |
| Shipping\_Address | Varchar(255) | Shipping address for the order |
| Billing\_Address | Varchar(255) | Billing address for the order (if different from shipping address) |
| Payment\_Method | Varchar(50) | Payment method used for the order |
| Payment\_Status | Enum('Pending', 'Completed', 'Cancelled', 'Refunded') | Payment status of the order |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**OrderItems table:**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Description** |
| OrderItemID (PK) | Integer | Primary Key - Unique OrderItem ID |
| OrderID (FK) | Integer | Foreign Key - ID of the parent order (references Orders table) |
| ProductID (FK) | Integer | Foreign Key - ID of the product being ordered (references Products table) |
| Quantity | Integer | Quantity of the product ordered |
| Price | Decimal(10, 2) | Price of the product per unit |
|  |  |  |
|  |  |  |
|  |  |  |

**Categories Table:**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data Type** | **Description** |
| CategoryID (PK) | Integer | Primary Key - Unique Category ID |
| Name | Varchar(50) | Name of the category |
| Description | Text | Description of the category |
|  |  |  |
|  |  |  |

**relationships** that could be defined between the tables for this e-commerce platform:

One-to-Many:

Users to Products - One user (seller) can have many product listings.

Users to Orders - One user (buyer) can have many orders.

Orders to OrderItems - One order can have many order line items.

Many-to-Many:

Products to Orders - One product can be part of many orders, and one order can contain many products. This is modeled using the OrderItems bridge table.

Products to Categories - One product can belong to multiple categories, and one category can contain multiple products. A separate ProductCategories table can define these relationships.

Users to Addresses - One user can have multiple shipping/billing addresses, and one address can be associated with multiple users. A separate UserAddresses table can define these.

Many-to-One

Description: Each order item is associated with one product, but a product can be part of multiple order items (orders).

**Index**

Users Table:

create an index on the "Username" and "Email" fields for fast user lookups and login operations.

If user searches or filtering by "Registration\_Date" is common, consider indexing that field.

Products Table:

Create an index on "Category" if you frequently filter or sort products by category.

Consider indexing the "Status" field if you often need to filter products by their availability status.

If you search or filter by "Listing\_Date," an index on this field can help.

OrderItems Table:

Create indexes on "OrderID" and "ProductID" since these are commonly used in queries to retrieve order details.

Depending on your query patterns, we might also consider indexing "OrderItemID" for direct access to specific order items.

Orders Table:

create an index on the "BuyerID" field since it's used to associate orders with users.

If frequently filter or sort orders by "Date\_Placed," consider indexing this field.

Index "Payment\_Status" if often query orders based on their payment status.

Categories Table:

If you commonly query products by category, consider indexing the "Name" field.

ProductCategories Table (if using Many-to-Many):

Create indexes on both "ProductID" and "CategoryID" since these fields are used to establish