ASSIGNMENT – 2

NAME: P.RAHUL REDDY

REG NO: 192324214

COURSE: DATABASE MANAGEMENT SYSTEM

COURSE CODE: CSA0593

SCENERIO:

Create a database for managing items, bids, users, and auctions.

– Model tables for items, bids, users, and auctions.

- Write stored procedures for placing and retracting bids.

- Implement triggers to update bid statuses and auction outcomes. - Write SQL queries to analyze bid history and item popularity.

Here's a more detailed breakdown of the database schema, including data types and constraints:

**1. Items Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| **item\_id** | **INT** | **PRIMARY KEY, AUTO\_INCREMENT** | **Unique identifier for the item** |
| **item\_name** | **VARCHAR(100)** | **NOT NULL** | **Name of the item** |
| **description** | **TEXT** |  | **Detailed description of the item** |
| **starting\_price** | **DECIMAL(10,2)** | **NOT NULL** | **Starting price of the auction** |
| **reserve\_price** | **DECIMAL(10,2)** |  | **Minimum acceptable price for the item** |
| **end\_time** | **DATETIME** | **NOT NULL** | **End time of the auction** |
| **seller\_id** | **INT** | **FOREIGN KEY REFERENCES users(user\_id)** | **ID of the seller** |

**2. Bids Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| **bid\_id** | **INT** | **PRIMARY KEY, AUTO\_INCREMENT** | **Unique identifier for the bid** |
| **item\_id** | **INT** | **FOREIGN KEY REFERENCES items(item\_id)** | **ID of the item being bid on** |
| **user\_id** | **INT** | **FOREIGN KEY REFERENCES users(user\_id)** | **ID of the bidder** |
| **bid\_amount** | **DECIMAL(10,2)** | **NOT NULL** | **Amount of the bid** |
| **bid\_time** | **DATETIME** | **NOT NULL** | **Time the bid was placed** |
| **status** | **ENUM('active', 'retracted', 'outbid')** | **NOT NULL** | **Status of the bid** |

**3. Users Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| **user\_id** | **INT** | **PRIMARY KEY, AUTO\_INCREMENT** | **Unique identifier for the user** |
| **username** | **VARCHAR(50)** | **NOT NULL, UNIQUE** | **Username of the user** |
| **password** | **VARCHAR(255)** | **NOT NULL** | **Hashed password of the user** |
| **email** | **VARCHAR(100)** | **NOT NULL, UNIQUE** | **Email address of the user** |

**4. Auctions Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| **auction\_id** | **INT** | **PRIMARY KEY, AUTO\_INCREMENT** | **Unique identifier for the auction** |
| **item\_id** | **INT** | **FOREIGN KEY REFERENCES items(item\_id)** | **ID of the item being auctioned** |
| **highest\_bid** | **DECIMAL(10,2)** |  | **Current highest bid for the item** |
| **highest\_bidder** | **INT** | **FOREIGN KEY REFERENCES users(user\_id)** | **ID of the user with the highest bid** |
| **status** | **ENUM('active', 'closed')** | **NOT NULL** | **Status of the auction** |

Additional Considerations

* Indexes: Consider adding indexes to frequently queried columns like item\_id, user\_id, and end\_time to improve query performance.
* Security: Implement strong password hashing and salting to protect user credentials.
* Data Validation: Validate input data to prevent invalid entries and security vulnerabilities.
* Error Handling: Handle errors gracefully, such as invalid bids or auction closures.
* Scalability: Consider database optimization techniques for large-scale auction systems.

By following these guidelines and tailoring the schema to your specific needs, you can create a robust and efficient database for your auction system.