Auction DB report

This report presents an in-depth analysis of the Auction Database system, focusing on its core entities, attributes, and relationships. The design aims to streamline auction operations by organizing data for users, items, auctions, bids, and transactions in a structured and efficient manner. Key insights include the identification of primary keys, attribute classifications, and cardinality mappings to ensure data integrity and

Prepared by

Osama Haj Hasan

Mohammad Qabaja

Eyas Raddad

Prepared for

Dr. Sherin Hijazi

One percent 1% Auction platform

Introduction

This project aims to design and analyze a robust Auction Database system that facilitates seamless and efficient auction operations. By structuring and organizing critical data components such as users, items, auctions, bids, and transactions, the system aims to enhance functionality and ensure data integrity. The project emphasizes the identification of primary keys, attributes, and relationships to optimize database performance. This foundational framework will serve as a blueprint for implementing advanced auction features, streamlining interactions between users and the system, and supporting reliable and secure transaction processing.

Auction Scenario

Consider the design of the following database system for managing an online auction platform: users can register on the platform as buyers or sellers, each identified by a unique user ID. Sellers can list items for auction, where each item has a unique item ID, a name, description, starting price, and a timestamp of when it was listed. Items are associated with auctions, which have a unique auction ID, a status indicating whether the auction is ongoing, completed, or canceled, as well as start and end times. Buyers can participate by placing bids on active auctions. Each bid is associated with a unique bid ID, the bid amount, and the timestamp of the bid. To ensure transaction integrity, each auction results in a single transaction recorded with a transaction ID, the final price, and the timestamp of the transaction. Users can have a balance on the platform for payments and settlements. Each auction involves multiple bids, but only the highest bid at the end of the auction is selected to finalize the transaction.

Data Base Analysis

Entities and attributes:

- Users (id, username, email, balance, password_hash)
  - Id: simple attribute
  - Username: simple attribute
  - Email: simple attribute
  - Balance: simple attribute
  - Password_hash: simple attribute

- Items (id, description, starting_price, added_at)
  - Id: simple attribute
  - Name: simple attribute
  - Description: simple attribute
  - starting_price: simple attribute
  - added_at: simple attribute

- Auctions (id, start_time, end_time, status)
  - Id: simple attribute
  - start_time: simple attribute
  - end_time: simple attribute
  - status: simple attribute

- Bids (id, amount, bid_time)
  - Id: simple attribute
  - amount: simple attribute
  - bid_time: simple attribute

- Transactions (id, final_price, transaction_time)
  - Id: simple attribute
  - final_price: simple attribute
  - transaction_time: simple attribute

Key attributes:

- PK: Users(id)
- PK: Items(id)
- PK: Auctions(id)
- PK: Bids(id)
- PK: Transactions(id)

Value sets:

- Users:
  - Id: number
  - Username: string
  - Email: string
  - Balance: number

- Password_hash: string

- Items:
  - Id: number
  - Name: string
  - Description: string
  - starting_price: date
  - added_at: date

- Auctions:
  - Id: number
  - start_time: date
  - end_time: date
  - status: string

- Bids:
  - Id: number
  - amount: number
  - bid_time: date

- Transactions:
  - Id: number
  - final_price: number
  - transaction_time: date

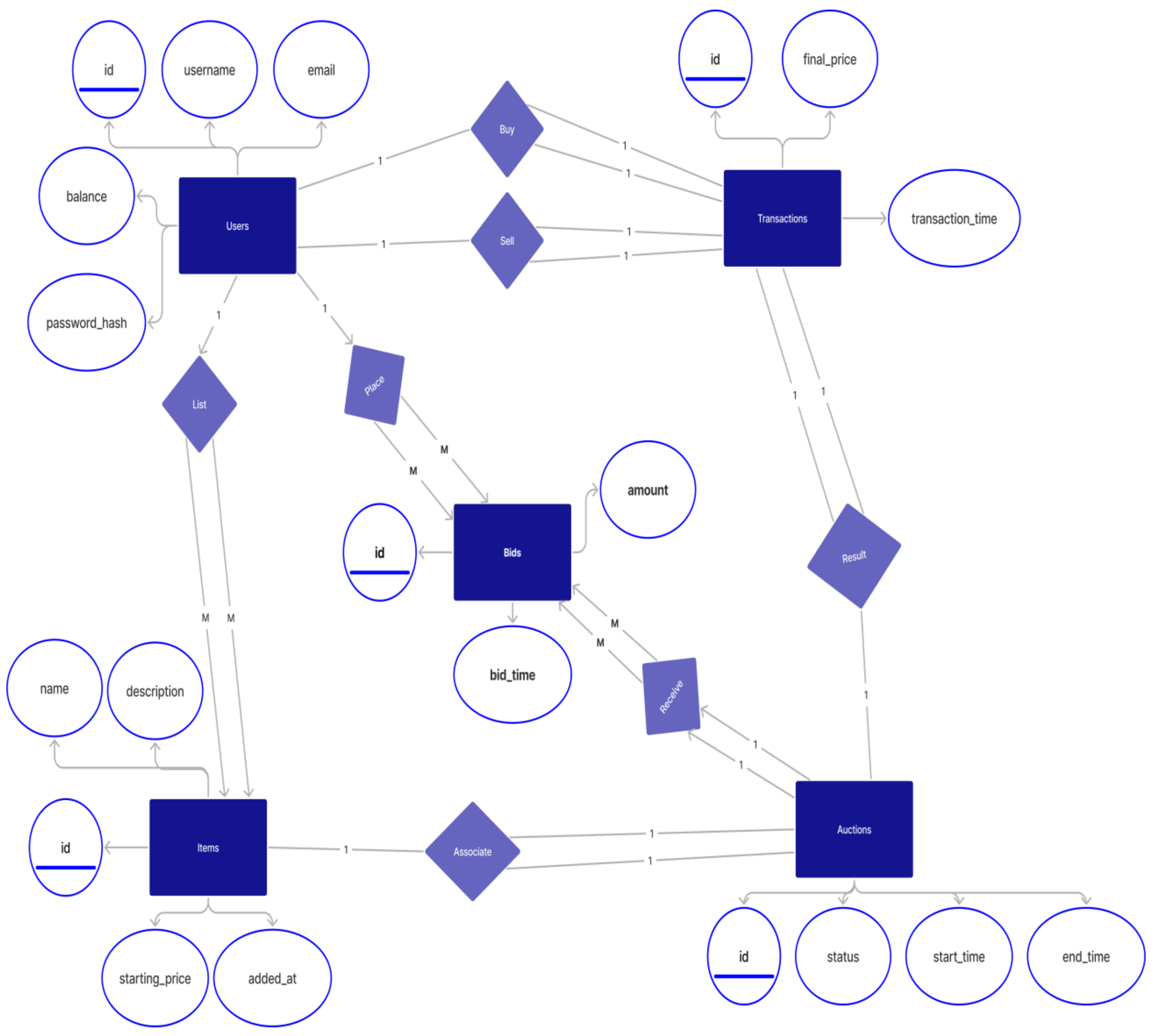
Relationships:

- List: (Users, Items)
- Associate: (Items, Auctions)
- Receive: (Auctions, Bids)
- Place: (Users, Bids)
- Result: (Auctions, Transactions)
- Buy: (Users, Transactions)
- Sell: (Users, Transactions)

Mapping Cardinality:

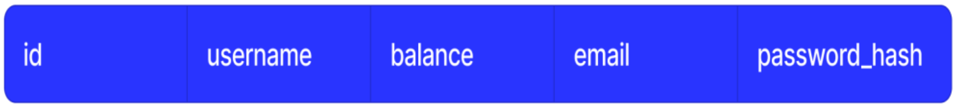
- List: (1, M)
- Associate: (1, 1)
- Receive: (1, M)
- Place: (1, M)
- Result: (1, 1)
- Buy: (1, 1)
- Sell: (1, 1)

Data Base Design



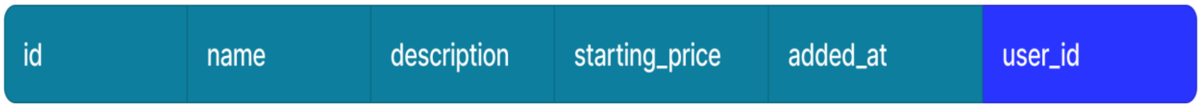
Data Base Translation

1. Users:



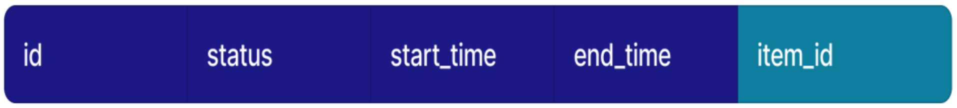
id	username	balance	email	password_hash

2. Items:



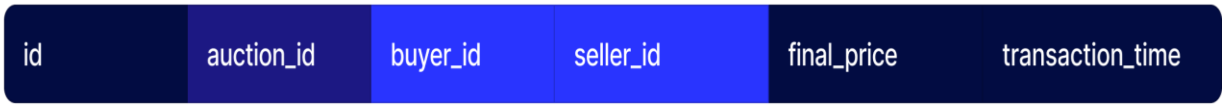
id	name	description	starting_price	added_at	user_id

3. Auctions:



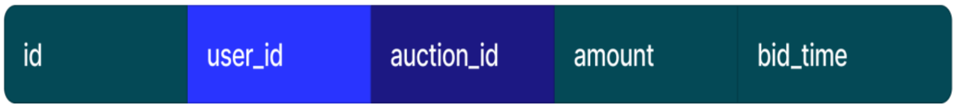
id	status	start_time	end_time	item_id

4. Transactions:



id	auction_id	buyer_id	seller_id	final_price	transaction_time

5. Bids:



id	user_id	auction_id	amount	bid_time

MySQL Code

```
CREATE TABLE Users (
        id INT AUTO_INCREMENT,
        username VARCHAR(25) NOT NULL UNIQUE,
        email VARCHAR(40) NOT NULL UNIQUE,
        password_hash VARCHAR(60) NOT NULL,
        balance DECIMAL(20,2) DEFAULT 0.0,
        PRIMARY KEY (id)
);
ALTER TABLE Users
AUTO_INCREMENT = 11111;

CREATE TABLE Items (
        id INT AUTO_INCREMENT,
        i_name VARCHAR(50) NOT NULL,
        i_description TEXT,
        starting_price DECIMAL(15, 2) NOT NULL,
        added_at      DATETIME      NOT      NULL      DEFAULT
CURRENT_TIMESTAMP,
        user_id INT NOT NULL,
        PRIMARY KEY (id),
CONSTRAINT FOREIGN KEY (user_id) REFERENCES Users (id) ON
DELETE CASCADE

);


CREATE TABLE Auctions (

        id INT AUTO_INCREMENT,

        item_id INT NOT NULL,

        a_status VARCHAR(10) NOT NULL CHECK (a_status IN ('active',
'completed', 'canceled')),

        start_time    DATETIME    NOT    NULL    DEFAULT
CURRENT_TIMESTAMP,

        end_time Date NOT NULL,

        PRIMARY KEY (id),

        CONSTRAINT FOREIGN KEY (item_id) REFERENCES Items
(id) ON DELETE CASCADE

);

ALTER TABLE Auctions

AUTO_INCREMENT = 7000;


CREATE TABLE Transactions (

        id INT AUTO_INCREMENT,

        auction_id INT NOT NULL,

        buyer_id INT NOT NULL,
	seller_id INT NOT NULL,
	final_price DECIMAL(15, 2) NOT NULL,
	transaction_time DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
	PRIMARY KEY (id),
	CONSTRAINT FOREIGN KEY (auction_id) REFERENCES Auctions (id) ON DELETE CASCADE,
	CONSTRAINT FOREIGN KEY (buyer_id) REFERENCES Users (id) ON DELETE CASCADE,
	CONSTRAINT FOREIGN KEY (seller_id) REFERENCES Users (id) ON DELETE CASCADE
);

ALTER TABLE Transactions
AUTO_INCREMENT = 10000;

CREATE TABLE Bids (
	id INT AUTO_INCREMENT,
	user_id INT NOT NULL,
	auction_id INT NOT NULL,
	amount DECIMAL(15, 2) NOT NULL,
	bid_time DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (id),
    CONSTRAINT FOREIGN KEY (auction_id) REFERENCES Auctions (id) ON DELETE CASCADE,
    CONSTRAINT FOREIGN KEY (user_id) REFERENCES Users (id) ON DELETE CASCADE
);
ALTER TABLE Bids
AUTO_INCREMENT = 5000;
```

Basic data insertion

Add User:

```
INSERT INTO Users (username, email, password_hash) VALUES
("admin", "alex19@gmail.com", "1234");
INSERT INTO Users (username, email, password_hash) VALUES
("Joe", "joe99@gmail.com", "12345");
```

Add Item:

```
INSERT INTO Items (i_name, i_description, starting_price, user_id)
VALUES ("Mercedes-Benz 300SL","The Mercedes-Benz 300SL is a
two-seater sports car, first manufactured as a coupe (1954–1957) with its
signature gullwing doors, and subsequently as a roadster (1957–1963).
The 300SL was launched in February 1954 at the International Motor
Sports Show in New York City. In a bid to attract American buyers early,
the company opted to introduce the car in the US first instead of
Europe.", 30000, 11111);
```

Open an Auction:

```
INSERT INTO Auctions (item_id, a_status, end_time) VALUES (1,
'active','2025-01-01 04:00:00');
```

Place Bid:

```
INSERT INTO Bids (user_id, auction_id, amount) VALUES (11112,
7000, 35000);
```

Transaction:

```
INSERT INTO Transactions (auction_id, buyer_id, seller_id,
final_price) VALUES (7000, 11112, 11111, 35000);
```