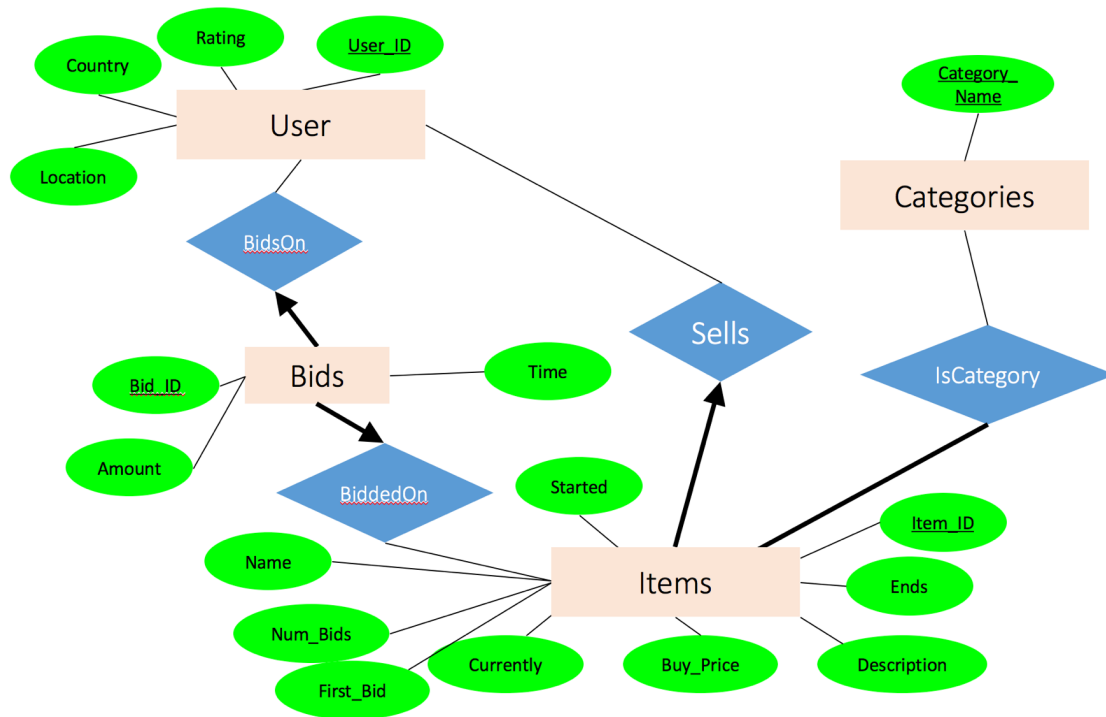


AuctionBase Project: Database and the Web

Part 1: From JSON to SQL

Solutions

ER Diagram



Notes:

- There are many ways to draw this ER diagram, as there are several ways to interpret the data. We did not deduct points for differing diagrams as long as the ER diagrams were designed correctly.
- This representation consolidates all sellers and bidders into one table called Users. This eliminates duplication for users that are both users and sellers. There are other ways to model this point, though.
- Categories and IsCategory can be combined into one table in the schema, as in practice every category we know about is associated with a product. This can be consolidated into the IsCategory table which has the two keys from its associated entities: category_name and item_id.
- The Sells relation can also be consolidated into a foreign key on a Seller_ID attribute in the Items table.
- Finally, the two Bid relationships and one Bid entity can be consolidated into one Bids table. See schema below.

Schema

Users (User_ID, Rating, Location, Country)

Items (Item_ID, Seller_ID, Name, Buy_Price, First_Bid, Currently, Number_of_Bids, Started, Ends, Description)

Categories (Item_ID, Category)

Bids (Item_ID, User_ID, Time, Amount)

Notes:

- As before, there are many ways to design your schema. That said, we *highly* recommend you switch to using this schema for later parts of the project if your design differs significantly. It will make later parts of the project much easier.
- The keys for the Bids table are not necessary at this stage of the project. We will be implementing additional constraints in Project Part II.

Queries

Query 1

Find the number of users in the database.

```
SELECT count(*)  
FROM Users;
```

Result: 13422

Query 2

Find the number of users from New York (i.e., users whose location is the string “New York”).

```
SELECT count(*)  
FROM Users  
WHERE Location = "New York";
```

Result: 80

Query 3

Find the number of auctions belonging to exactly four categories.

```
SELECT count(*)  
FROM (  
    SELECT Item_ID  
    FROM Categories  
    GROUP BY Item_ID  
    HAVING count(Category) = 4  
);
```

Result: 8365

Query 4

Find the ID(s) of auction(s) with the highest current price.

```
SELECT Item_ID
```

```
FROM Items
WHERE Currently = (SELECT max(Currently) FROM Items);
```

Result: 1046871451

Query 5

Find the number of sellers whose rating is higher than 1000.

```
SELECT count(distinct Items.Seller_ID)
FROM Items, Users
WHERE Items.Seller_ID = Users.User_ID and Rating > 1000;
```

Result: 3130

Query 6

Find the number of users who are both sellers and bidders.

```
SELECT count(distinct Items.Seller_ID)
FROM Items, Bids
WHERE Items.Seller_ID = Bids.User_ID;
```

Result: 6717

Query 7

Find the number of categories that include at least one item with a bid of more than \$100.

```
SELECT count(distinct Category)
FROM Categories
WHERE Item_ID in (SELECT Item_ID FROM Bids WHERE Amount > 100);
```

Result: 150