Avery Peiffer

aep65

Assignment 7

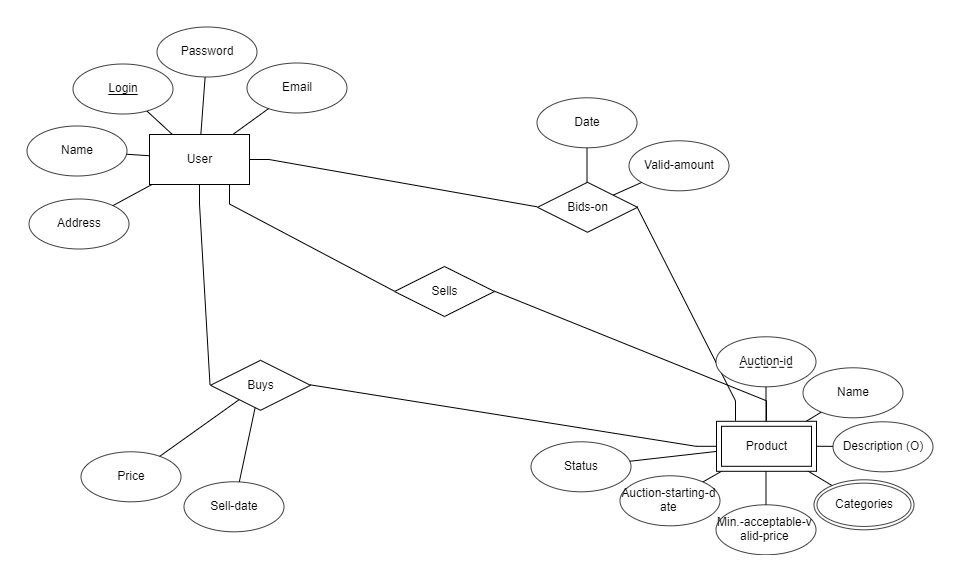
Assumptions:

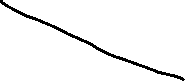


- Users cannot bid the same amount multiple times for an item (i.e., every bid for an item will be a different amount)

- Only one user can sell an item with a given id

- Users can make a profile without selling anything

1. 



User is an entity that contains all of the user’s information (login, password, name, email, address) with their login being the key. User is a strong entity since it can exist independently of the other entities. Product is an entity that contains all of the information about the product being sold. It is a weak entity since it cannot exist without a user to sell it. Auction-id is the key for product, but it does not make up the entire key for a product, so it is underlined with a dotted line. The other information for a product includes name, description (which is optional), status, auction starting date, the minimum acceptable valid price, and the categories (which is a multi-varied attribute and thus is a double oval).

There are three relationships between users and products. First, products are sold by users. The products entity must fully participate in this relation because a product cannot exist without a user to sell it (hence the double line which I had to draw in myself). The user entity partially participates in this relation because a user can exist without selling a product, as stated in the assumptions. One user can sell many products, but one product (speaking in terms of its product-id) cannot be sold by multiple users, which explains the 1-M cardinality.

Second, products are bid on by users. Both users and products partially participate in this relation since users can exist without making bids and products can exist without being bid on. A user can bid on many products, and many users can bid on the same product, which explains the M-N cardinality for the relation.

Third, users can buy products when they meet the price of the seller. I chose to have this be a separate relation from selling because they have different cardinality; semantically, it would also be more useful to keep this data together than to have to find it in the bid data. Both users and products partially participate in this relation since users can exist without buying anything and products can go unsold. Since only one user can buy a given product, but a user can buy multiple products, this explains the 1-M cardinality for this relation.

2.

USER (login, password, name, email, address)

PK (login)

CHECK (password IS NOT NULL)

CHECK (name IS NOT NULL)

CHECK (email IS NOT NULL)

CHECK (address IS NOT NULL)

PRODUCT (auction-id, seller-id, name, description, starting-date, status, min-acceptable-price)

PK (auction-id)

FK (seller-id) 🡪 USER (login)

CHECK (name IS NOT NULL)

CHECK (starting-date IS NOT NULL)

CHECK (status=’under auction’ OR status=’sold’ OR status=’withdrawn’)

CHECK (min-acceptable-price > 0)

BIDS (buyer-id, auction-id, valid-amount, date)

PK (buyer-id, auction-id, valid-amount)

FK (buyer-id) 🡪 USER (login)

FK (auction-id) 🡪 PRODUCT (auction-id)

CHECK (date IS NOT NULL)

SALES (buyer-id, auction-id, sell-date, price)

PK (auction-id)

FK (buyer-id) 🡪 USER (login)

FK (auction-id) 🡪 PRODUCT (auction-id)

CHECK (sell-date IS NOT NULL)

CHECK (price IS NOT NULL)

CATEGORIES (category-name, subcategory-name)

PK (category-name, subcategory-name)

PRODUCT\_CATEGORIES (auction-id, category-name, subcategory-name)

PK (auction-id, category-name, subcategory-name)

FK (auction-id) 🡪 PRODUCT (auction-id)

FK (category-name, subcategory-name) 🡪 CATEGORIES (category-name, subcategory-name)