COMP3013 Coursework: Design Report

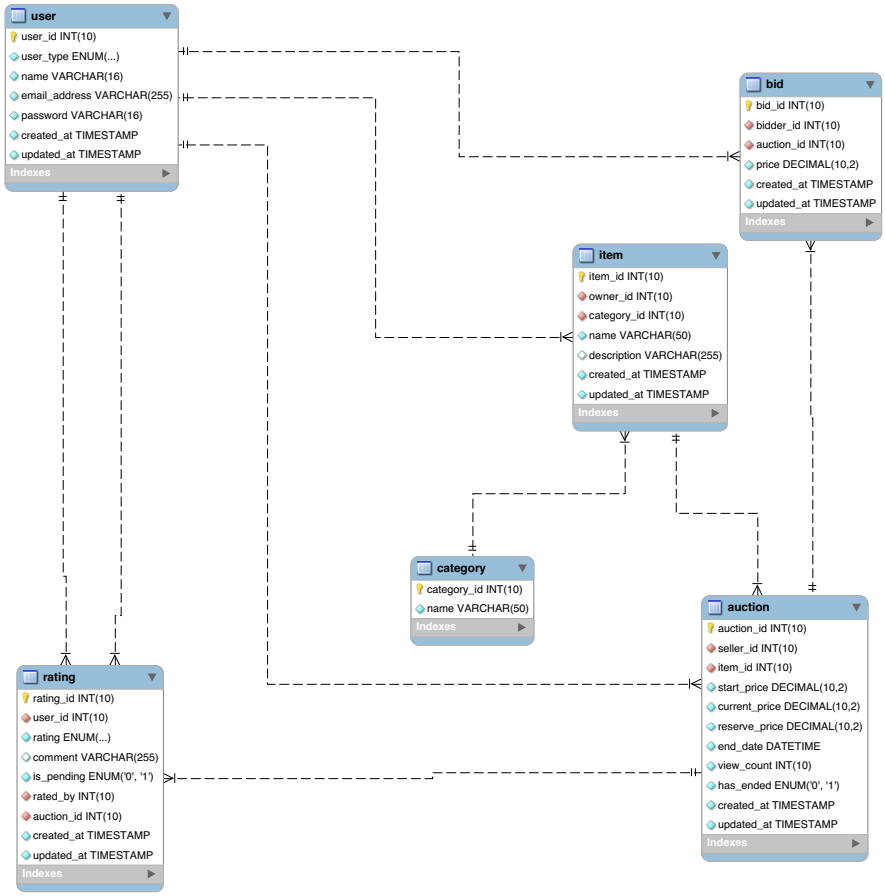
Matthew Lapointe, Takeaki Yamasaki, Zhengyi Yang

18th March 2016

1 YouTube Video

`https://youtu.be/aTJTMciGE98`

2 Entity Relationship Diagram



3 Database Schema

This schema implements the ER diagram shown above. `user` and `category` are strong entities, while `item`, `auction`, `bid`, and `rating` are weak entities, with appropriate foreign key attributes. `user` to `item`, `user` to `auction`, `user` to `bid`, `user` to `rating`, and `auction` to `bid` are all one to many relationships.

user (user_id, user_type, name, email_address, password, created_at, updated_at)

item (item_id, owner_id **references** user(user_id), category_id **references** category(category_id), name, description, created_at, updated_at)

auction (auction_id, seller_id **references** user(user_id), item_id **references** item(item_id), start_price, current_price, reserve_price, end_date, view_count, has_ended, created_at, updated_at)

bid (bid_id, bidder_id **references** user(user_id), auction_id **references** auction(auction_id), price, created_at, updated_at)

category (category_id, name)

rating (rating_id, user_id **references** user(user_id), rating, comment, is_pending, rated_by **references** user(user_id), auction_id **references** auction(auction_id), created_at, updated_at)

4 Normalisation Analysis

The schema has been implemented in SQL, thus is necessary does not contain repeating groups. The database is in First Normal Form.

There are no tables containing composite keys, so it must be the case that each non-primary key attribute is fully functionally dependent on its primary key. The database, then, is in Second Normal Form.

Examining the non-primary key attributes of each table, it is clear that no transitive dependencies exist. The database is in Third Normal Form.

5 Database Queries

5.1 Registering and Logging In

```
SELECT *
FROM USER
WHERE email_address=$email_address
```

```
  AND password=$password
  AND user_type=$user_type;
```

This query retrieves entries in the `user` table which match the email address, password, and user type (seller/buyer) selected. This allows for authentication of a user attempting to log in.

```
INSERT INTO user(name,email_address,password,user_type,created_at)
VALUES($name,$email_address, $password, $user_type, NULL);
```

This query inserts a new user into the database.

5.2 Creating Auctions

```
INSERT INTO item(owner_id, category_id, name, description, created_at)
VALUES($_SESSION['user_id'], $item_category, $item_name, $item_desc, NULL);
```

This query inserts a new item into the `item` table.

```
INSERT INTO auction(seller_id, item_id, start_price, current_price, reserve_price,
                    end_date, has_ended, created_at)
VALUES($_SESSION['user_id'], $item['item_id'], $start_price, $start_price,
       $reserve_price, $end_date, '0', NULL);
```

This query inserts creates a new entry in the `auction` table.

5.3 Browsing Auctions

```
SELECT i.item_id,
       i.name,
       i.description
FROM item AS i
LEFT JOIN auction AS a ON i.item_id=a.item_id
WHERE i.category_id=$_GET['category']
  AND a.has_ended='0'
ORDER BY $sort;
```

This query retrieves all items on auction in the selected category. It arranges items by the selected criterion.

5.4 Handling Bids

```
SELECT *
FROM bid
WHERE auction_id=$auction['auction_id']
ORDER BY price DESC;
```

This query retrieves all the bids placed in a given auction.

```
INSERT INTO bid(bidder_id, auction_id, price, created_at)
VALUES($_SESSION['user_id'], $auction['auction_id'], $bid_price, NULL);
```

This query inserts a new bid into the `bid` table.

```
SELECT a.auction_id,
       i.name AS item_name,
       i.item_id,
       u.user_id AS seller_id,
       u.name AS seller_name,
       u.email_address AS seller_address,
       a.reserve_price
FROM auction AS a
LEFT JOIN item AS i ON i.item_id = a.item_id
LEFT JOIN USER AS u ON u.user_id = a.seller_id
WHERE end_date <= now()
  AND has_ended='0';
```

This query is run periodically, retrieving auctions which have ended. The item is then awarded to the user who placed the highest bid, if that bid exceeds the reserve price.

5.5 Watching Auctions

```
SELECT b.price,
       u.user_id,
       u.name,
       u.email_address
FROM bid AS b
LEFT JOIN USER AS u ON b.bidder_id=u.user_id
WHERE b.auction_id=$auction['auction_id']
ORDER BY b.price DESC LIMIT 1;
```

When a new bid is placed, this query retrieves the previously highest bid, if it exists. An email is then sent to that user notifying them that they have been outbid.

5.6 Seller Reports

```
SELECT a.end_date,
       a.current_price,
       a.reserve_price,
       a.auction_id,
       a.view_count,
       i.name
FROM auction AS a
LEFT JOIN item AS i ON a.item_id = i.item_id
```

```
WHERE a.seller_id =$seller['user_id']
  AND a.end_date > now();
```

This query retrieves auctions for a given seller which have not yet ended. The information is then sent to the user in an update email.

5.7 Ratings

```
SELECT *
FROM rating
WHERE user_id=$_GET['USER']
  AND is_pending='0';
```

This query retrieves from the `rating` table all ratings for a given user, which are displayed on their user page.

```
SELECT round(avg(CASE rating WHEN '5' THEN 5 WHEN '4' THEN 4
                             WHEN '3' THEN 3 WHEN '2' THEN 2
                             WHEN '1' THEN 1 ELSE NULL END),1) avg_rating
FROM rating
WHERE user_id=$_GET['USER']
  AND is_pending='0';
```

This query computes the average rating for a given user, which is displayed on their user page.

5.8 Recommendations

```
SELECT *
FROM auction
WHERE auction.auction_id
IN (

SELECT bid.auction_id
FROM bid
WHERE bid.bidder_id
IN (

SELECT bid.bidder_id
FROM bid
WHERE bid.bidder_id <> $user_id
AND bid.auction_id
IN (

SELECT bid.auction_id
FROM bid
WHERE bid.bidder_id = $user_id
```

```
GROUP BY bid.auction_id
)
GROUP BY bid.bidder_id
)
GROUP BY bid.auction_id
)
AND auction.has_ended =  '0'
LIMIT 10;
```

This query provides recommendations for user by retrieving the sorts of auctions other people, who have also bid on the sorts of auctions the user have previously bid on, are currently bidding on.