UNIVERSITY COLLEGE LONDON

Design Report of Database for Online Auction System

COMPGC06 Database Systems

GROUP 29

Tuo Fang 15052014

Boyuan Zhang 14112299

Jingting Li 15028396

Youtube Link: https://youtu.be/80NL-gPCJ34

CATALOGUE

1. Database Design

1.1.Conceptual Design

The online auction system should have the following capabilities. Users can register and create accounts; users have roles of seller or buyer; seller can create auctions and buyer can bid for items; both buyer and seller have access to get the report of the progress of the auction; buyer and seller can rate the auction and have visible ratings aggregated from their feedbacks on participations in auctions.

Considering the requirements of the online auction system, the database implementation should contains at least following entities: user (seller and buyer), item, bid, auction, rating, status, category. Figure 1 shows the conceptual model of the database.

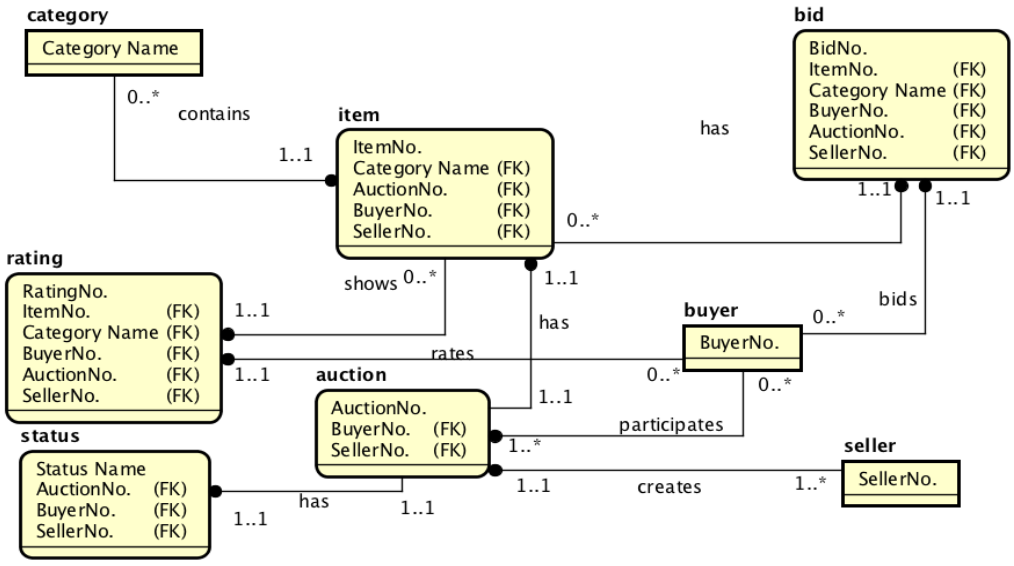

figure 1. conceptual model

1.2. Logical Design (Entity Relationship Diagram)

Given the conceptual model of the database, we add attributes other than primary key and foreign key to every entity in order to build the Entity Relationship Diagram (ER Diagram) as demonstrated in figure 2.

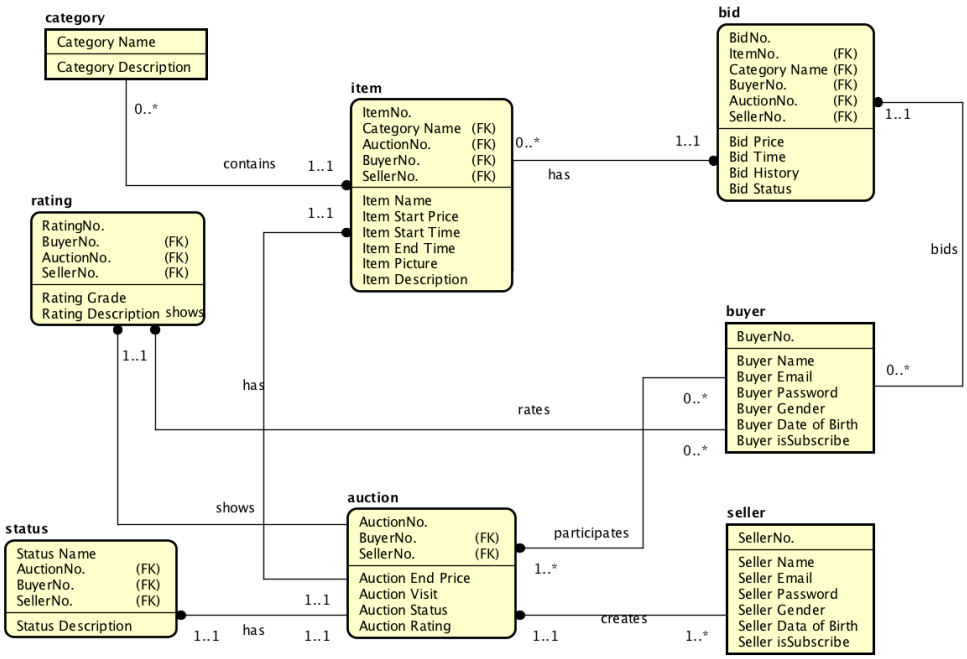

figure 2 Entity Relationship Diagram

1.3. Physical Design (Database Schema)

The database schema is implemented according to ER diagram. Noted that buyer and seller entity, which have almost the same attributes, are merged into one entity user to achieve simplicity. Figure 3 displays the physical model of the database system.

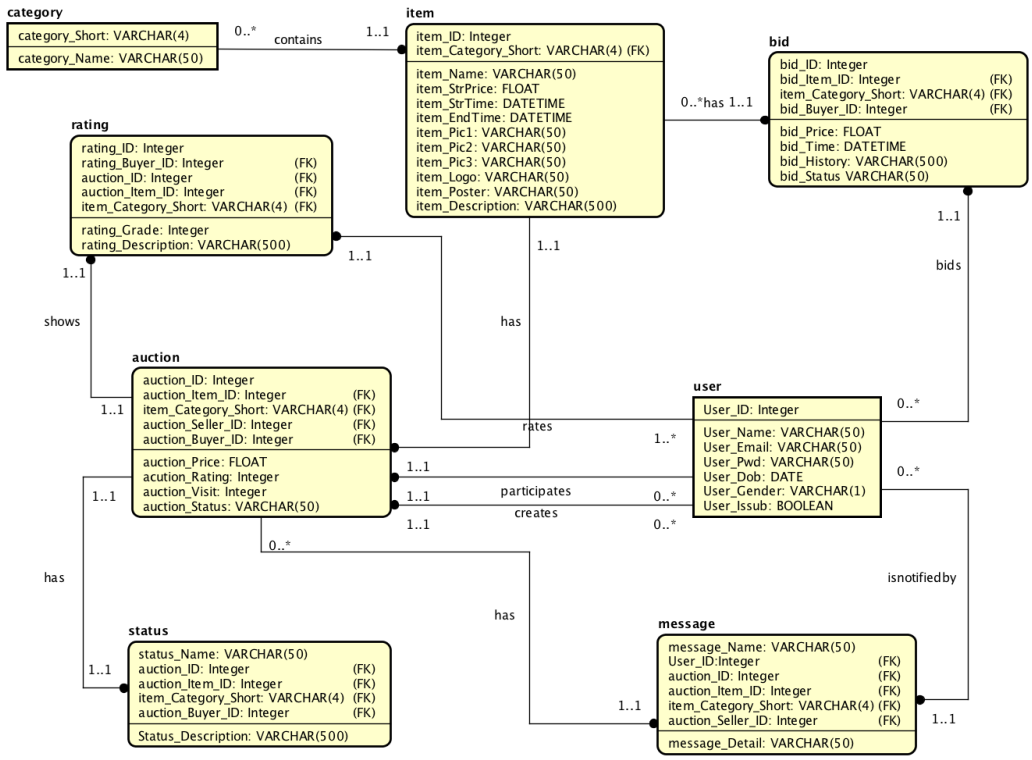

figure 3 Physical Model Diagram

Ultimately, 8 tables were created to support dynamic web content, as described in the following. Each of these tables will be explained in detail.

Table of Users: *tbl_user*
This table store the details of users that are registered in the database. The information is obtained during the user registration process. Particularly the passwords stored in this table are hashed. The role of user (seller, buyer or both) is identified in specific situation, e.g. if a user bids for an item, the role of the user at this circumstances is buyer. The primary key is user_id.

user_id	user_name	user_email	user_pwd	user_dob	user_gender	user_issub

---

Table of items: *tbl_item*
The item table stores the data of the items that is current in auction. Including category, name, description, image, price, time and so on. In addition, the item_poster column is used to indicate if the item is shown in the index page. The primary key is item_id. And foreign key is item_category_short which indicates the belonging relationship with entity category.

item_id	item_name	item_strprice	item_strtime	item_endtime	item_pic1	item_pic2	item_pic3	item_logo	item_poster	item_seller_id	item_category_short	item_description

---

Table of bids: *tbl_bid*
The bid table stores every bid buyer has performed. The table stores information of the certain bid including bid price, bid time, buyer id and so on. The primary key is bid_id. And foreign keys are bid_buyer_id and bid_item_id. Noted that attribute item_category_short is removed for normalisation, which will be discussed in detail in next chapter.

bid_id	bid_price	bid_time	bid_item_id	bid_buyer_id	bid_history	bid_status

---

Table of auctions: *tbl_auction*
The auction table stores each created auction. The information is obtained during the auction creation process. The primary key is auction_id. And foreign keys are auction_item_id, auction_buyer_id and auction_seller_id. The auction_buyer_id column only stores the buyer with highest bid and other buyers information can be obtained in bid_history column in tbl_bid.

auction_id	auction_price	auction_rating	auction_visit	auction_status	auction_buyer_id	auction_seller_id	auction_item_id

---

Table of ratings: *tbl_rating*
The rating table simply stores all the rating records. The information is obtained after every bid. Each rating is provided by a buyer towards the current auction who have participated in the auction. So foreign keys are rating_item_id and rating_buyer_id. It is the same choosing rating_item_id or rating_auction_id as foreign key because item and auction have 1 to 1 relationship.

rating_id	rating_grade	rating_item_id	rating_description	rating_buyer_id

---

Table of categories: *tbl_category*
Filters by categories should automatically generated. This table is used to store each possible category to be used to sort and organise the catalogue in the catalogue page. Primary key is category_short.

category_short	category_name

Table of status: *tbl_status*
Status table stores the status of auctions. The notification can be sent to users when status changes. Primary key is status_name.

status_name	status_description

Table of messages: *tbl_message*
Message table stores every message that should be sent to specific user about an auction he of she has participated or created. Primary key is message_name and foreign keys are message_user_id and message_auction_id.

message_name	message_user_id	message_auction_id	message_detail

2. Normalisation Analysis

2.1. First Normal Form (1NF)

First Normal Form is a relation in which the intersection of each row and column contains only one value, which our database schema well satisfied. So our database design is already at 1NF.

2.2. Second Normal Form (2NF)

Second Normal Form is a relation that is in first normal form and every non-primary-key attribute is fully functionally dependent on primary key.

The resulting 1NF relations are as follows:

category	(category_Short, category_Name)
rating	(rating_ID, rating_Buyer_ID, auction_ID, auction_Item_ID, item_Category_Short, rating_Grade, rating_Description)
auction	(auction_ID, auction_Item_ID, item_Category_Short, auction_Seller_ID, auction_Buyer_ID, auction_Price, auction_Rating, auction_Visit, auction_Status)
item	(item_ID, item_Name, item_Category_Short, item_name, item_StrPrice, item_StrTime, item_EndTime, item_Pic1, item_Pic2, item_Pic3, item_Logo, item_Poster, item_Description)
bid	(bid_ID, bid_item_ID, item_Category_Short, bid_Buyer_ID, bid_Price, bid_Time, bid_History, bid_Status)
user	(user_ID, user_Name, user_Email, user_Pwd, user_Dob, user_Gender, user_Issub)
status	(status_Name, auction_ID, auction_Item_ID, item_Category_Short, auction_Buyer_ID, Status_Description)
message	(message_ID, user_ID, auction_ID, auction_Item_ID, item_Category_Short, auction_Seller_ID, message_Detail)

Underlined attributes are primary keys. Following listed relations are not fully functional dependent:

status_Name →auction_ID, auction_Item_ID, item_Category_Short, auction_Buyer_ID

So we update the relation as follows:

status_Name →status_Description

After the update, it can be seen that every attribute in every relation is fully functionally dependent on the primary key, i.e. once primary key is set, values of all other attributes are settled. Until now the database design is at 2NF.

2.3. Third Normal Form (3NF)

Third Normal Form is a relation that is in First and Second Normal Form and in which no non-candidate-key is transitively dependent on any candidate key. So the main aim to convert from 2NF to 3NF is to remove all the transitive dependency. There are several transitive dependencies in 2NF as follows:

Rating: auction_ID→auction_Item_ID
auction_Item_ID→item_Category_Short
Auction: auction_Item_ID→item_Category_Short
Bid: bid_Item_ID→item_Category_Short
Message: auction_ID→auction_Item_ID
auction_ID→auction_Seller_ID
auction_Item_ID→item_Category_Short

After removing all the transitive dependencies above, we can assure that the database design is at 3NF.

3. Explanation SQL

3.1. Capability 1

Users can register with the system and create accounts.
Users have roles of seller or buyer with different privileges.

INSERT INTO
tbl_user(user_name,user_email,user_pwd,user_dob,user_gender,user_issub)
VALUES('$user_name','$email','$password','$dob','$gender','$issub')
Explanation:
Insert user information into *tbl_user*.

SELECT user_email FROM tbl_user WHERE user_email='$email'
Explanation:

Check duplication between emails in database and that user registers with. If so, pop up a warning window and ask for typing again.

SELECT user_name FROM tbl_user WHERE user_name='$user_name'
Explanation:
Check duplication between usernames in database and that user registers with. If so, pop up a warning window and ask for typing again.

SELECT * FROM tbl_user WHERE user_name = '$loginname'
Explanation:
Login, search the usernames in the database and check the match of password

3.2. Capability 2

Sellers can create auctions for particular items, setting suitable conditions and features of the items including the item description, categorisation, starting price, reserve price and end date.

INSERT INTO
tbl_item(item_name,item_strprice,item_strtime,item_endtime,item_pic1,item_pic2,item_pic3,item_seller_id,item_category_short,item_description)
VALUES('$itemname','$bidamount','$starttime','$endtime','$images[0]','$images[1]','$images[2]','$sellerid','$cateshort','$description')");
Explanation:
Insert item information into *tbl_item*.

INSERT INTO tbl_auction(auction_price, auction_seller_id, auction_item_id)
VALUES ('$bidamount','$sellerid','$itemid')");
Explanation:
Insert new auction into *tbl_auction*.

3.3. Capability 3

Buyers can search the system for particular kinds of item being auctioned and can browse the visually re-arrange listings of items within categories.

SELECT item_name, item_category_short FROM tbl_item WHERE item_name LIKE '$searchname%'
Explanation:
Search for particular kinds of item matching the typed word at the beginning.

SELECT * FROM tbl_item WHERE item_category_short='$category' AND item_endtime > '$now'
Explanation:
Select all the items that are in active auctions in a certain category.

3.4. Capability 4

Buyers can bid for items and see other buyers' bids. The system will manage the auction until the set end time and award the item to the highest bidder. The system should confirm to both the winner and seller of an auction its outcome.

SELECT bid_id FROM tbl_bid WHERE bid_item_id = '$item_id'

UPDATE tbl_bid SET bid_status = 'bid_not_highest' WHERE bid_item_id = '$item_id'
Explanation:
Select and change status of all the bids before current bid to not highest.

SELECT * FROM tbl_bid WHERE bid_item_id = '$item_id' ORDER BY bid_price DESC LIMIT 1
Explanation:
Fetch the highest price of all the bids and determine whether the price of current bid is higher than former highest price.

SELECT * FROM tbl_item WHERE item_id = '$item_id'
Explanation:
Fetch the end time of current auction and determine whether the auction has been expired at present.

SELECT * FROM tbl_bid WHERE bid_item_id = '$item_id' AND bid_buyer_id = '$user_id'
UPDATE tbl_bid
SET bid_price = '$bid_price', bid_time = '$date', bid_history='$bidhistory'
WHERE bid_item_id = '$item_id' AND bid_buyer_id = '$user_id'
Explanation:
Search for the bid user has performed and update its price, time and bid history. Noted that bid history contains price and bid time of all the bids the user has.

INSERT INTO tbl_bid(bid_id, bid_price, bid_time, bid_item_id, bid_buyer_id,bid_history)
VALUES (NULL, '$bid_price', '$date', '$item_id', '$user_id','$bidhis')
Explanation:
Simply insert all the information of the bid if the user bids the item for the first time.

UPDATE tbl_auction
SET auction_price = '$bid_price', auction_buyer_id = '$user_id'
WHERE auction_item_id = '$item_id'
Explanation:
Update all the relevant information including price and buyer of current auction.

SELECT tbl_user.user_name, tbl_bid.bid_price, tbl_bid.bid_time FROM tbl_bid, tbl_user
WHERE bid_item_id = '$item_id' AND user_id = bid_buyer_id ORDER BY tbl_bid.bid_price DESC
Explanation:
Select and display the bid history including bidder, bid price and bid time.

SELECT * FROM tbl_bid
INNER JOIN tbl_item
ON tbl_bid.bid_item_id = tbl_item.item_id
WHERE bid_buyer_id = '$bidderid'
AND bid_status = 'bid_winner'

SELECT * FROM tbl_auction WHERE auction_item_id = '$item_id1'
Explanation:
Select all the expired auctions buyer has participated and won, and then notify the buyer if such auction exists.

SELECT * FROM tbl_auction
INNER JOIN tbl_user
ON tbl_auction.auction_buyer_id = tbl_user.user_id
INNER JOIN tbl_item
ON tbl_auction.auction_item_id = tbl_item.item_id
WHERE auction_seller_id = '$bidderid'
AND auction_status = 'auction_closed'
Explanation:
Select all the expired auctions seller has created, and then notify the seller the final detail of the auction including bid price, buyer, etc. if such auction exists.

3.5. Capability 5

Buyers can watch auctions on items and receive emailed updates on bids on those items including notifications when they are outbid.

SELECT * FROM tbl_bid
INNER JOIN tbl_item
ON tbl_bid.bid_item_id = tbl_item.item_id
WHERE bid_buyer_id = '$bidderid'
AND bid_status = 'bid_not_highest'

SELECT * FROM tbl_auction WHERE auction_item_id = '$item_id'
Explanation:
Select all the bids user has participated and been outbid, and then notify the user if such bid exists.

SELECT * FROM tbl_bid
INNER JOIN tbl_item
ON tbl_bid.bid_item_id = tbl_item.item_id
WHERE bid_buyer_id = '$bidderid'
AND bid_status = 'bid_not_winner'

SELECT * FROM tbl_auction WHERE auction_item_id = '$item_id2'
Explanation:
Search the auction user has participated and not won, and then notify the user if such auction exists.

SELECT * FROM tbl_bid
INNER JOIN tbl_item
ON tbl_bid.bid_item_id = tbl_item.item_id
WHERE bid_buyer_id = '$bidderid'
AND bid_status IN ('bid_highest','bid_not_highest')
Explanation:

Search all the bids buyer has performed in active auction and display all the detail including item name, bid price and bid time.

```
SELECT * FROM tbl_bid
INNER JOIN tbl_item
ON tbl_bid.bid_item_id = tbl_item.item_id
WHERE bid_buyer_id = '$bidderid'
AND bid_status IN ('bid_winner','bid_not_winner','bid_done')
```

Explanation:
Search all the bids buyer has performed in expired auction and display all the detail including item name, bid price and bid time and whether the buyer has won the auction.

3.6. Capability 6

Sellers can receive reports on the progress of the auction through to completion and how much viewing traffic their auction items have had.

```
SELECT * FROM tbl_auction
INNER JOIN tbl_item
ON tbl_auction.auction_item_id = tbl_item.item_id
WHERE auction_seller_id = '$bidderid'
AND auction_status = 'auction_active'
```

```
SELECT * FROM tbl_bid
INNER JOIN tbl_user
ON tbl_bid.bid_buyer_id = tbl_user.user_id
WHERE bid_item_id = '$item_id3'
ORDER BY bid_time DESC
```

Explanation:
Search all the active auctions created by a seller and then notify him or her when some buyer has bid for this item.

```
SELECT * FROM tbl_auction
INNER JOIN tbl_item
ON tbl_auction.auction_item_id = tbl_item.item_id
WHERE auction_seller_id = '$sellerid'
AND auction_status = 'auction_active'
```

```
SELECT * FROM tbl_status WHERE status_name = 'auction_active'
```

Explanation:
Search all the active auctions created by a seller and display detail information including auction highest price, auction highest bidder, auction view traffic, auction current rating, etc.

```
SELECT * FROM tbl_auction
INNER JOIN tbl_item
ON tbl_auction.auction_item_id = tbl_item.item_id
WHERE auction_seller_id = '$sellerid'
AND auction_status <> 'auction_active'
```

SELECT user_name FROM tbl_user WHERE user_id = '$buyerid'
Explanation:
Search all the expired auctions created by a seller and display detail information including auction final price, auction buyer, auction view traffic, auction rating, etc.

3.7. Capability 7

Buyers and sellers have visible ratings aggregated from the feedback on their participation in auctions.

INSERT INTO tbl_rating(rating_id, rating_grade, rating_item_id, rating_description, rating_buyer_id)
VALUES (NULL, '$grade', '$item_id', '$description', '$buyerid')
Explanation:
Insert rating information into database.

SELECT * FROM tbl_rating WHERE rating_item_id = '$item_id'
UPDATE tbl_auction SET auction_rating = '$overall_rating'
WHERE auction_item_id = '$item_id'
Explanation:
Select all the rating belonging to a certain item and calculate the aggregated value to update the overall rating grade for the auction.