
Introduction to Database

-Term project-

2016310936 우승민

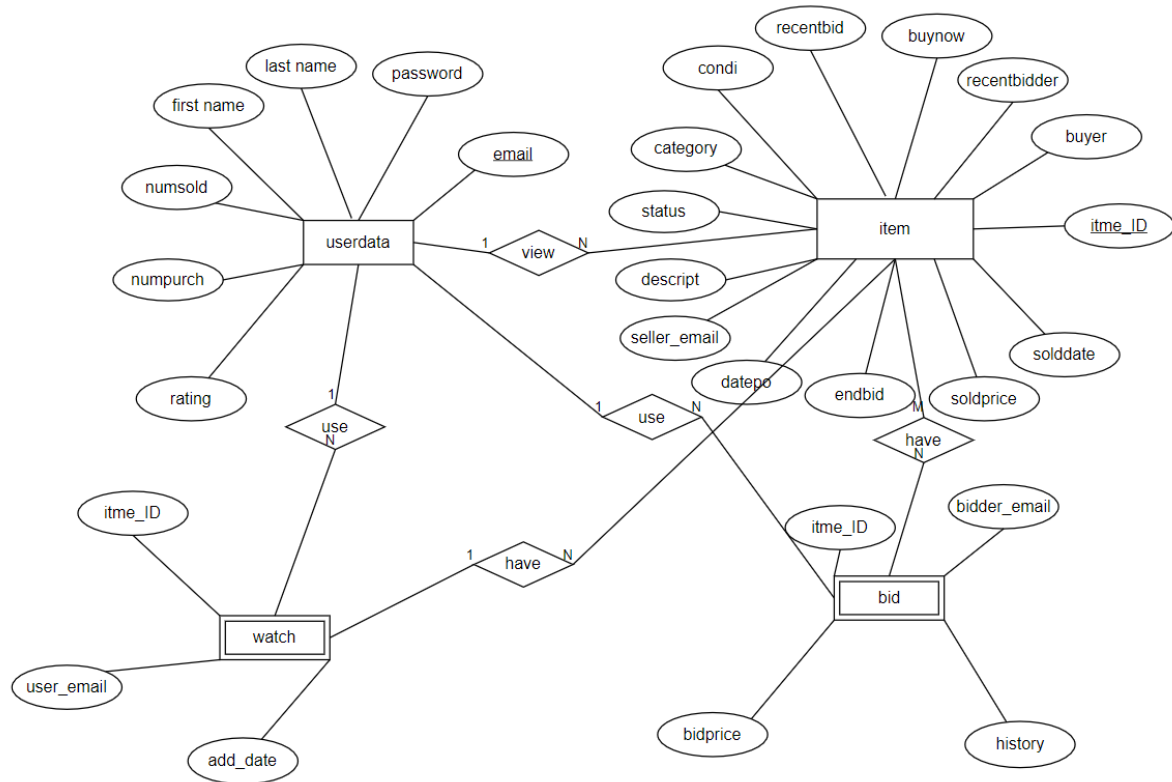
Contents

1. Schema Diagram	4
1.1 Overall E-R diagram	4
1.2 Entities	4
1.3 Relational Schema	6
2. DDL statements	7
2.1 userdata	7
2.2 item	7
2.3 watch	8
2.4 watch	8
2.5 Event	8
2.6 Trigger	9
3. DML statements	10
3.1 Sign up	10
3.2 Login	10
3.3 Sell	11
3.4 Status	11
3.5 Search	12
3.6 Check bid	14
3.7 Check Account	15
4. Administrator mode	16
4.1 Login Administrator	16
4.2 Manage item	17
4.4 Show profit	18
.....	19
5. User manual	20

5.1 Login / Sign up	20
5.2. Sell	21
5.3 Status	21
5.4. Search	22
5.5 Check bid	25
5.6 Check Account.....	25

1. Schema Diagram

1.1 Overall E-R diagram



1.2 Entities

1) userdata

userdata
<u>email</u>
password
first name
last name
rating
numsold
numpurch

Userdata has user's information. When users sign up, then email, password, first name, last name are made. And email is a primary key. Rating, numsold, numpurch are changed when user buys item or sells item.

2) item

item
<u>item_ID</u>
category
seller_email
descript
condi
recentbid
buynow
status
datepo
endbid
soldprice
recentbidder
buyer
solddate

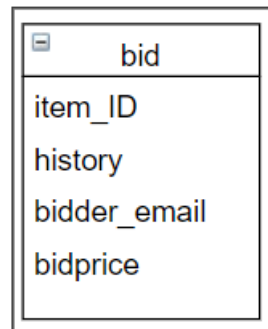
Item has information of item like it's category, condition, and so on. Item_ID is a primary key of item. When user uploads item, then user should input it's category, description, condition, buy-it-now price, end bidding date. So item_ID, category, seller_email, descript, condi, recentbid, buynow, status, datepo, endbid are written automatically when users upload item. And other things like status, buyer and so on are changed when item is sold, or some user bids it.

3) watch

watch
user_email
item_ID
add_date

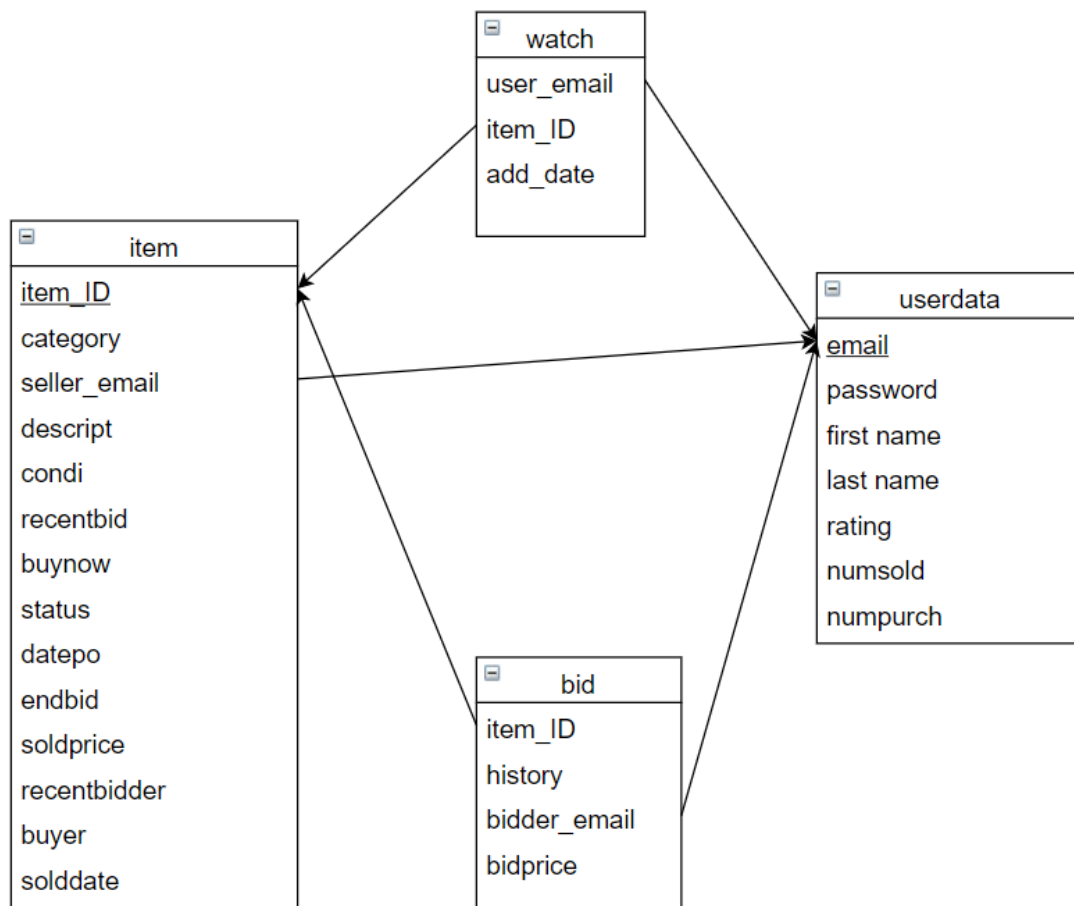
When user searches item by category, description and so on, user watches many items. And all of items which user watch, are uploaded automatically and also add_date which means the time user watches items is uploaded.

4) bid



When user is bidding items, some information is uploaded. Bidder_email means user_email, item_ID means item which user is bidding, bidprice means the price which user sets money to bid item, history means the time when user bid.

1.3 Relational Schema



2. DDL statements

2.1 userdata

```
create table userdata
(
    email          varchar(40) not null,
    password       varchar(40) not null,
    firstname      varchar(20) not null,
    lastname       varchar(20) not null,
    rating         integer,
    numsold        integer,
    numpurch       integer,
    primary key (email)
);
```

Email, password, firstname, lastname should be written so they can't be NULL. And data type is varchar and others data type is integer. Email is primary key.

2.2 item

```
create table item
(
    item_ID        integer not null,
    category       integer not null,
    seller_email    varchar(40) not null,
    descript       varchar(30) not null,
    condi         integer not null,
    recentbid      integer,
    buynow         integer,
    status         varchar(12),
    datepo        datetime,
    endbid         datetime,
    soldprice      integer,
    recentbidder   varchar(40),
    buyer          varchar(40),
    solddate       datetime,
    primary key (item_ID),
    foreign key(seller_email) references userdata(email)
);
```

Datepo, endbid, solddate data type is datetime like "2020-12-03 12:00:00". Category and condition data type is integer because I set like this. And item_ID is primary key. Seller_email is foreign key references by userdata.

2.3 watch

```
create table watch
(
    user_email      varchar(40) not null,
    item_ID         integer not null,
    add_date        datetime,
    foreign key (user_email) references userdata(email),
    foreign key (item_ID) references item(item_ID)
);
```

When user search items and watch list, then watch list is made. User_email, item_ID are foreign key, add_date means the time when user watch items.

2.4 watch

```
create table bid
(
    item_ID         integer not null,
    history         datetime,
    bidder_email    varchar(40) not null,
    bidprice        integer not null,
    foreign key (item_ID) references item(item_ID),
    foreign key (bidder_email) references userdata(email)
);
```

When user bid items, then bid information is inserted. History means the time when user bid items.

2.5 Event

```
create event bidsold on schedule every 1 second on completion preserve enable
do update item set status = 'sold', soldprice = recentbid, buyer = recentbidder, solddate = endbid
where item.endbid < now() and item.status != 'sold' and item.status != '0 bid' and item.status != 'fail'; ||
```

```
create event bidfail on schedule every 1 second on completion preserve enable
do update item set status = 'fail' where item.endbid < now() and item.status = '0 bid'; ||
```

These two events are activated every 1 second. When time is over item's end bidding date, then item's information is changed. If item is bid at least one or more, then item is sold by highest bidder. Else item is not bid by anyone, then item is failed bidding.

2.6 Trigger

```
create trigger sold after update on item for each row
begin if old.status != 'sold' and new.status = 'sold' then
update userdata set numsold = numsold + 1, rating = rating + 1 where email = new.seller_email; end if;
end; ||
```

```
create trigger purch after update on item for each row
begin if old.status != 'sold' and new.status = 'sold' then
update userdata set numpurch = numpurch + 1 where email = new.buyer; end if;
end; ||
```

These two triggers are activated when some user buy items. Users who buy items gets +1 at num of purchasing item. And users who sell items gets +1 at num of sold, rating.

3. DML statements

3.1 Sign up

```
void Signup(){
    char email[40];
    char password[40];
    char first[20];
    char last[20];
    printf("-----< Sign up >\n");
    printf("----- first name:");
    scanf("%s", first);
    printf("----- last name:");
    scanf("%s", last);
    printf("----- email:");
    scanf("%s", email);
    printf("----- password:");
    scanf("%s", password);

    char *si;
    sprintf(si, "insert into userdata VALUES ('%s', '%s', '%s', '%s', '0', '0', '0')", email, password, first, last);
    if(mysql_query(conn, si)){
        printf("error");
        mysql_close(conn);
        exit(1);
    }

    cout << endl;
}
```

When user sign up, then new information is inserted at userdata. Email, password, first name, last name is written by user, and others are set 0.

3.2 Login

```
int query_stat = mysql_query(conn, "select * from userdata");
MYSQL_RES *result = mysql_store_result(conn);

if(result == NULL){
    printf("mysql_store_result: %s\n", mysql_error(conn));
    mysql_close(conn);
    exit(1);
}

int num_fields = mysql_num_fields(result);

MYSQL_ROW row;
char email[40];
char password[40];
int lo = 0;
printf("-----< Login >\n");
printf("----- email:");
scanf("%s", email);
printf("----- password:");
scanf("%s", password);

while((row = mysql_fetch_row(result))){
    if(strcmp(row[0], email)==0){
        if(strcmp(row[1], password)==0){
            lo = 1;
            break;
        }
    }
}
```

When users select login menu, they should enter email and password. After then, database checkes whether their email and password are existed and right.

3.3 Sell

```
char *si = new char[200];
sprintf(si, "insert into item VALUES ('%d', '%d', '%s', '%s', '%d', '%d', '%d', '0 bid', now(), '%s', NULL, NULL, NULL, NULL)", item_ID, cate, ee, des, con, 0, buy, date);

delete des;

if(mysql_query(conn, si)){
    delete si;
    printf("error");
    mysql_close(conn);
    exit(1);
}
```

When users upload item, then new information is uploaded. Item_ID is set by the order in which item is uploaded. Cate means category, ee means user's email, des means description, con means condition, buy means buy-it-now price, date means end bidding date. These data is set by users. So I use sprintf.

3.4 Status

```
printf("----< Status of Your Item Listed on Auction >\n");

int query_stat = mysql_query(conn, "select * from item");

MYSQL_RES *result = mysql_store_result(conn);

if(result == NULL){
    printf("mysql_store_result: %s\n", mysql_error(conn));
    mysql_close(conn);
    exit(1);
}

int num_fields = mysql_num_fields(result);

MYSQL_ROW row;
int count = 1;
while((row = mysql_fetch_row(result))){
    if(strcmp(row[2], ee)==0){
        cout << "[Item " << count++ << "]" << endl;
        cout << "        description: " << row[3] << endl;
        cout << "        status: " << row[7] << endl;
        if(strcmp(row[7], "sold")!=0){
            cout << "        current bidding price: " << row[5] << endl;
            cout << "        current highest bidder: ";
            if(row[11] != NULL) cout << row[11] << endl;
            else cout << "NULL" << endl;
            cout << "        date posted: " << row[8] << endl;
            cout << "        bid ending date: " << row[9] << endl;
        }
    }
}
```

In status function, program bring information from item. And checking is there any items which users upload to sell. And if it is existed, then show list of them.

3.5 Search

```
printf("----< Search results: Category >#n");

int count = 1;
int ct = 0;
int *item_arr = new int[100];
while((row = mysql_fetch_row(result))) {
    if(atoi(row[1]) == ca) {
        if(strcmp(row[7], "sold") != 0 && strcmp(row[7], "fail") != 0 && strcmp(row[2], ee) != 0) {
            cout << "[Item " << count++ << "]" << endl;
            cout << "    description: " << row[3] << endl;
            cout << "    status: " << row[7] << endl;
            item_arr[ct] = atoi(row[0]);
            cout << "    current bidding price: " << row[5] << endl;
            cout << "    current highest bidder: ";
            if(row[11] != NULL) cout << row[11] << endl;
            else cout << "NULL" << endl;
            cout << "    date posted: " << row[8] << endl;
            cout << "    bid ending date: " << row[9] << endl;

            char *watch = new char[200];
            sprintf(watch, "insert into watch VALUES ('%s', '%d', now())", ee, item_arr[ct++]);
            mysql_query(conn, watch);
            delete watch;
        }
    }
}
```

When users search items by category or description, ... Then, they watch item list which item is not sold, not fail, and not uploaded by searching user. And these watching list is inserted at watch table.

```
cout << "---- Bidding price? (Enter the price or 'buy' to pay for the buy-it-now price): ";
char *bid = new char[20];
cin >> bid;

if(strncmp(bid, "buy", 3) == 0) {
    char *watch = new char[200];
    sprintf(watch, "select buynow from item where item_ID = %d", item_arr[buy]);
    mysql_query(conn, watch);
    result = mysql_store_result(conn);
    row = mysql_fetch_row(result);

    int soldprice = atoi(row[0]);

    sprintf(watch, "update item set status = 'sold', soldprice = %d, solddate = now(), buyer = '%s' where item_ID = %d", soldprice, ee, item_arr[buy]);
    mysql_query(conn, watch);
    delete watch;
}
```

When user selects item which users want to buy or bid. Then they are asked Bidding price. When user enter 'buy' then item's information is changed. Status -> sold, soldprice -> user input, solddate -> now time, buyer -> user_email.

```

else{
    int price = atoi(bid);
    char *watch = new char[200];
    sprintf(watch, "select status, recentbid from item where item_ID = %d", item_arr[buy]);
    mysql_query(conn, watch);
    result = mysql_store_result(conn);
    row = mysql_fetch_row(result);
    int i=0;
    char rr[6];
    while(row[0][i] != 'b') i++;
    int j=0;
    for(j=0; j<=i-2; j++)
        rr[j] = row[0][j];
    rr[j] = '\0';
    if(price < atoi(row[1]))
        sprintf(watch, "update item set status = '%d bids' where item_ID = %d", atoi(rr)+1, item_arr[buy]);
    else
        sprintf(watch, "update item set status = '%d bids', recentbid = %d, recentbidder = '%s' where item_ID = %d", atoi(rr)+1, price, ee, item_arr[buy]);
    mysql_query(conn, watch);

    sprintf(watch, "insert into bid VALUES ('%d', now(), '%s', '%d')", item_arr[buy], ee, price);
    mysql_query(conn, watch);
    delete watch;
}

```

Else user enter number then item is not sold but bidded. So item's information is changed not like sold. Status -> n bids. And if user sets highest bidding price, then recentbid, recentbidder is also changed.

And bid table also uploaded.

3.6 Check bid

```
char *check = new char[100];
sprintf(check, "select * from bid where bidder_email = '%s'", ee);
int query_stat = mysql_query(conn, check);

MYSQL_RES *result = mysql_store_result(conn);

if(result == NULL){
    printf("mysql_store_result: %s\n", mysql_error(conn));
    mysql_close(conn);
    exit(1);
}

int num_fields = mysql_num_fields(result);

MYSQL_ROW row;

int ct = 0;

int *item_arr = new int[100];
int *price = new int[100];
while((row = mysql_fetch_row(result))){
    item_arr[ct] = atoi(row[0]);
    price[ct++] = atoi(row[3]);
}
```

First, system check bid table so find all of items which are bid by user.

```
for(int i=0; i<ct; i++){
    sprintf(check, "select * from item where item_ID = %d", item_arr[i]);
    if(mysql_query(conn, check)) cout << "error" << endl;
    result = mysql_store_result(conn);
    row = mysql_fetch_row(result);
    cout << "[item " << i+1 << "]" << endl;
    cout << "        description: " << row[3] << endl;
    cout << "        status: ";
    if(strcmp(row[7], "sold")==0){
        if(strcmp(row[12], ee)==0){
            cout << "You won the item" << endl;
        }
        else
            cout << "You are outbidded and the item is sold." << endl;
        cout << "        sold price: " << row[10] << endl;
        cout << "        sold date: " << row[13] << endl;
    }
    else{
        if(strcmp(row[11], ee)==0){
            cout << "You are the highest bidder." << endl;
        }
        else cout << "You are outbidded." << endl;
        cout << "        your bidding price: " << price[i] << endl;
        cout << "        current highest bidding price: " << row[5] << endl;
        cout << "        bid ending date: " << row[9] << endl;
    }
}
}
```

And then, show user about their bidding item.

3.7 Check Account

```
cout << "----< Check your Account >" << endl;

char *check = new char[100];
sprintf(check, "select descript, soldprice from item where seller_email = '%s' and status = 'sold'", ee);
int query_stat = mysql_query(conn, check);

MYSQL_RES *result = mysql_store_result(conn);

int *sold = new int[100]();
int count = 0;
if(result == NULL){
    cout << "No sold item" << endl;
}
else{
    int num_fields = mysql_num_fields(result);

    MYSQL_ROW row;

    while((row = mysql_fetch_row(result))){
        cout << "[Sold Item " << count+1 << "]" << endl;
        cout << "    description: " << row[0] << endl;
        if(row[1]){
            cout << "    sold price: " << row[1] << endl;
            sold[count++] = atoi(row[1]);
        }
        else{
            cout << "    sold price: NULL " << endl;
            sold[count++] = 0;
        }
    }
}
```

First, system show item list which users uploaded to sell. So finding items which user uploaded and check if they are sold.

```
sprintf(check, "select descript, soldprice from item where buyer = '%s'", ee);
mysql_query(conn, check);
result = mysql_store_result(conn);

int purch = 0;
if(result == NULL){
    cout << "No purchased item" << endl;
}
else{
    int num_fields = mysql_num_fields(result);

    MYSQL_ROW row;

    count = 0;
    while((row = mysql_fetch_row(result))){
        cout << "[Purchased Item " << count+1 << "]" << endl;
        cout << "    description: " << row[0] << endl;
        if(row[1]){
            cout << "    purchase price: " << row[1] << endl;
            purch += atoi(row[1]);
        }
        else{
            cout << "    purchase price: NULL" << endl;
        }
        count++;
    }
}
```

After, system show item list which users buy. So finding items which user buy.

4. Administrator mode

4.1 Login Administrator

```
int LoginAd(){
    int query_stat = mysql_query(conn, "select * from userdata");
    MYSQL_RES *result = mysql_store_result(conn);

    if(result == NULL){
        printf("mysql_store_result: %s\n", mysql_error(conn));
        mysql_close(conn);
        exit(1);
    }

    int num_fields = mysql_num_fields(result);

    MYSQL_ROW row;
    char email[40];
    char password[40];
    int lo = 0;
    printf("----< Login Administrator >\n");
    printf("---- email:");
    scanf("%s", email);
    printf("---- password:");
    scanf("%s", password);

    if(strcmp("admin", email) == 0){
        while((row = mysql_fetch_row(result))){
            if(strcmp(row[0], email) == 0){
                if(strcmp(row[1], password) == 0){
```

For using administrator mode, you should select login administrator. And unlike login, it does not need user's email by reference because administrator email is always 'admin'.

4.2 Manage item

Show item list is same as users searching item. By category, description, seller, date. So I skip this. After then admin can change item information.

```
cout << "---- What do you want?" << endl;
cout << "---- (1) change information" << endl;
cout << "---- (2) Go Back" << endl;
cout << "---- (3) Quit" << endl;
cin >> bid;
if(bid == 1){
    cout << "---- What do you want to change?" << endl;
    cout << "---- (1) category" << endl;
    cout << "---- (2) description" << endl;
    cout << "---- (3) condition" << endl;
    cout << "---- (4) end bidding date" << endl;
    cout << "---- (5) status" << endl;
    int act;
    cin >> act;
    if(act == 1){
        cout << "---- select category" << endl;
        cout << "      (1) Electronics" << endl;
        cout << "      (2) Books" << endl;
        cout << "      (3) Home" << endl;
        cout << "      (4) Clothing" << endl;
        cout << "      (5) Sporting Goods" << endl;
        int category;
        cin >> category;
        char *watch = new char[200];
        sprintf(watch, "update item set category = '%d' where item_ID = '%d'", category, item_arr[buy]);
        mysql_query(conn, watch);
        delete watch;
    }
}
```

Admin can change item information. If admin thinks this item's category is wrong, or end bidding date is too short or long, and so on. Changing other things is not allowed.

4.3 Manage user

```
printf("----< Show users >\n");

int query_stat = mysql_query(conn, "select * from userdata");
MYSQL_RES *result = mysql_store_result(conn);

if(result == NULL){
    printf("mysql_store_result: %s\n", mysql_error(conn));
    mysql_close(conn);
    exit(1);
}

int num_fields = mysql_num_fields(result);
MYSQL_ROW row;

int count = 1;

while((row = mysql_fetch_row(result))){
    cout << "[User " << count++ << "]" << endl;
    cout << "      email: " << row[0] << endl;
    cout << "      password: " << row[1] << endl;
    cout << "      rating: " << row[4] << endl;
    cout << "      numsold: " << row[5] << endl;
    cout << "      numpurchased: " << row[6] << endl;
}

if(count == 1){
```

To manage user, first system show admin user list.

```

char *em = new char[40];
cout << "----- What do you want? (Enter number and user's email) ex) 1 asdf" << endl;
cout << "----- (1) change information" << endl;
cout << "----- (2) Go Back" << endl;
cout << "----- (3) Quit" << endl;
cin >> bid;
if(bid == 1){
    cin >> em;
    cout << "----- Please enter what to be changed and how to be changed except email ex) password 1234" << endl;
    char *change1 = new char[40];
    char *change2 = new char[40];
    cin >> change1 >> change2;
    char *watch = new char[200];
    sprintf(watch, "update userdata set %s = '%s' where email = '%s'", change1, change2, em);
    mysql_query(conn, watch);
    delete watch;
    delete change1;
    delete change2;
}

```

And then, admin selects user to change. And then, enter which information to change and to what.

4.4 Show profit

```

cout << "----- Show each profit of item" << endl;

int query_stat = mysql_query(conn, "select * from item");
MYSQL_RES *result = mysql_store_result(conn);

if(result == NULL){
    printf("mysql_store_result: %s\n", mysql_error(conn));
    mysql_close(conn);
    exit(1);
}

int num_fields = mysql_num_fields(result);
MYSQL_ROW row;

while((row = mysql_fetch_row(result))){
    if(strcmp(row[7], "sold") == 0){
        cout << "item_ID : " << row[0] << endl;
        if(atoi(row[1]) == 1) cout << "category : Electronics" << endl;
        else if(atoi(row[1]) == 2) cout << "category : Books" << endl;
        else if(atoi(row[1]) == 3) cout << "category : Home" << endl;
        else if(atoi(row[1]) == 4) cout << "category : clothing" << endl;
        else cout << "category : Sporting Goods" << endl;
        cout << "seller_email : " << row[2] << endl;
        cout << "description : " << row[3] << endl;
        if(row[10]) cout << "price : " << row[10] << " won" << endl;
    }
}

```

First, system show admin item information which is sold.

```

query_stat = mysql_query(conn, "select seller_email, soldprice from item where status = 'sold' order by seller_email");
;

result = mysql_store_result(conn);

char sold[100][40] = {0};
int ct = 0;
int price[100] = {0};
int total[100] = {0};
int count = 0;
int cnt=0;
while((row = mysql_fetch_row(result))){
    strcpy(sold[count], row[0]);
    if(row[1]) price[count] = atoi(row[1]);
    else price[count] = 0;
    count++;
}

for(int i=0; i<count; i++){
    if(strcmp(sold[i], sold[i+1]) == 0){
        ct++;
        if(ct < 3)
            total[cnt] = price[i] * 0.02;
        else
            total[cnt] = price[i] * 0.01;
    }
    else{

```

And then distinguish them group by user. Because commission is different by num of sold.

5. User manual

5.1 Login / Sign up

```
2016310936@swji:~/db/1$ ./a.out
----< Login menu >
----(1) Login
----(2) Sign Up
----(3) Login as Administrator
----(4) Quit
----Your choice: _
```

To login, enter '1'. To sign up, enter '2'.

```
----Your choice: 2
----< Sign up >
----first name:a
----last name:a
----email:a
----passord:a

----< Login menu >
----(1) Login
----(2) Sign Up
----(3) Login as Administrator
----(4) Quit
----Your choice: _
```

If you select 2, then you should enter 4 things in order. After then system go back login menu.

```
----< Login menu >
----(1) Login
----(2) Sign Up
----(3) Login as Administrator
----(4) Quit
----Your choice: 1
----< Login >
----email:a
----password:a

----< Main menu > : (Enter the number)
----(1) Sell item
----(2) Status of Your Item Listed on Auction
----(3) Search item
----(4) Check Status of your Bid
----(5) Check your Account
----(6) Quit
```

If you want to login, you should enter what you input at sign up. After then system go main menu.

5.2. Sell

```
-----< Main menu > : (Enter the number)
----- (1) Sell item
----- (2) Status of Your Item Listed on Auction
----- (3) Search item
----- (4) Check Status of your Bid
----- (5) Check your Account
----- (6) Quit
1
-----< Sell item >
----- select from the following category : (Enter the number)
----- (1) Electronics
----- (2) Books
----- (3) Home
----- (4) Clothing
----- (5) Sporting Goods
1
----- condition: (Enter the number)
----- (1) New
----- (2) Like-New
----- (3) Used (Good)
----- (4) Used (Acceptable)
1
----- description:1
----- buy-it-now price:123
----- bid ending date:2020-12-21 15:00
```

If you select sell menu, then you should enter some things in order. Category, condition, buy-it-now price should be only number. And bid ending date should be entered "YYYY-MM-DD HH:MM"

After uploading, system go back main menu.

5.3 Status

```
-----< Main menu > : (Enter the number)
----- (1) Sell item
----- (2) Status of Your Item Listed on Auction
----- (3) Search item
----- (4) Check Status of your Bid
----- (5) Check your Account
----- (6) Quit
2
-----< Status of Your Item Listed on Auction >
[Item 1]
description: 1
status: 0 bid
current bidding price: 0
current highest bidder: NULL
date posted: 2020-12-03 17:25:16
bid ending date: 2020-12-21 15:00:00
```

If you select status menu, you can watch your item list.

5.4. Search

```
----(3) Search item
----(4) Check Status of your Bid
----(5) Check your Account
----(6) Quit
3
----< Search item > : (Enter the number)
----(1) Search items by category
----(2) Search items by description keyword
----(3) Search items by seller
----(4) Search items by date posted
----(5) Go Back
----(6) Quit
1
----< Search items by category > : (Enter the number)
----(1) Electronics
----(2) Books
----(3) Home
----(4) Clothing
----(5) Sporting Goods
1
----< Search results: Category >
[Item 1]
    description: 111
    status: 2 bids
    current bidding price: 2
    current highest bidder: zxcv
    date posted: 2020-12-03 14:55:46
    bid ending date: 2020-12-04 12:00:00
---- Which item do you want to bid? (Enter the number or 'B' to go back to the previous menu):
```

If you select search item, then you select searching way. In category, you just enter number.

```
----< Search item > : (Enter the number)
----(1) Search items by category
----(2) Search items by description keyword
----(3) Search items by seller
----(4) Search items by date posted
----(5) Go Back
----(6) Quit
2
----< Search items by description keyword >
---- Search keyword : (Enter the keyword)
1
----< Search results : keyword search >
[Item 1]
    description: 111
    status: 2 bids
    current bidding price: 2
    current highest bidder: zxcv
    date posted: 2020-12-03 14:55:46
    bid ending date: 2020-12-04 12:00:00
[Item 2]
    description: 123
    status: 0 bid
    current bidding price: 0
    current highest bidder: NULL
    date posted: 2020-12-03 14:56:17
    bid ending date: 2020-12-07 15:00:00
---- Which item do you want to bid? (Enter the number or 'B' to go back to the previous menu):
```

In description, system shows item list which description starts by string you enter.

```

----(3) Search items by seller
----(4) Search items by date posted
----(5) Go Back
----(6) Quit
3
----< Search items by seller >
---- Search keyword : (Enter the keyword)
qwer
----< Search results : seller search >
[Item 1]
    description: 111
    status: 2 bids
    current bidding price: 2
    current highest bidder: zxcv
    date posted: 2020-12-03 14:55:46
    bid ending date: 2020-12-04 12:00:00
[Item 2]
    description: 123
    status: 0 bid
    current bidding price: 0
    current highest bidder: NULL
    date posted: 2020-12-03 14:56:17
    bid ending date: 2020-12-07 15:00:00
[Item 3]
    description: 22
    status: 2 bids
    current bidding price: 44
    current highest bidder: asdf
    date posted: 2020-12-03 14:56:49
    bid ending date: 2020-12-22 20:20:00
--- Which item do you want to bid? (Enter the number or 'B' to go back to the previous menu):

```

In seller, you should enter seller's email exactly.

```

----(4) Search items by date posted
----(5) Go Back
----(6) Quit
4
----< Search items by date posted >
---- Search keyword : (Enter the keyword)
2020-12-03
----< Search results : date search >
[Item 1]
    description: 111
    status: 2 bids
    current bidding price: 2
    current highest bidder: zxcv
    date posted: 2020-12-03 14:55:46
    bid ending date: 2020-12-04 12:00:00
[Item 2]
    description: 123
    status: 0 bid
    current bidding price: 0
    current highest bidder: NULL
    date posted: 2020-12-03 14:56:17
    bid ending date: 2020-12-07 15:00:00
[Item 3]
    description: 22
    status: 2 bids
    current bidding price: 44
    current highest bidder: asdf
    date posted: 2020-12-03 14:56:49
    bid ending date: 2020-12-22 20:20:00
--- Which item do you want to bid? (Enter the number or 'B' to go back to the previous menu):

```

In date posted, you should enter like 'YYYY-MM-DD'.

After then you enter item number or 'B' to go back previous menu.

```

2020-12-03
----< Search results : date search >
[Item 1]
  description: 111
  status: 2 bids
  current bidding price: 2
  current highest bidder: zxcv
  date posted: 2020-12-03 14:55:46
  bid ending date: 2020-12-04 12:00:00
[Item 2]
  description: 123
  status: 0 bid
  current bidding price: 0
  current highest bidder: NULL
  date posted: 2020-12-03 14:56:17
  bid ending date: 2020-12-07 15:00:00
[Item 3]
  description: 22
  status: 2 bids
  current bidding price: 44
  current highest bidder: asdf
  date posted: 2020-12-03 14:56:49
  bid ending date: 2020-12-22 20:20:00
--- Which item do you want to bid? (Enter the number or 'B' to go back to the previous menu): 1
--- Bidding price? (Enter the price or 'buy' to pay for the buy-it-now price): 2

```

If you select item, then you select bidding or buying now. If you enter number, bidding is start. Like under image.

```

2020-12-03
----< Search results : date search >
[Item 1]
  description: 111
  status: 3 bids
  current bidding price: 2
  current highest bidder: a
  date posted: 2020-12-03 14:55:46
  bid ending date: 2020-12-04 12:00:00
[Item 2]
  description: 123
  status: 0 bid
  current bidding price: 0
  current highest bidder: NULL
  date posted: 2020-12-03 14:56:17
  bid ending date: 2020-12-07 15:00:00
[Item 3]
  description: 22
  status: 2 bids
  current bidding price: 44
  current highest bidder: asdf
  date posted: 2020-12-03 14:56:49
  bid ending date: 2020-12-22 20:20:00
--- Which item do you want to bid? (Enter the number or 'B' to go back to the previous menu): 2
--- Bidding price? (Enter the price or 'buy' to pay for the buy-it-now price): buy

```

If you enter 'buy' then item is sold by you and list is changed.

```

2020-12-03
----< Search results : date search >
[Item 1]
  description: 111
  status: 3 bids
  current bidding price: 2
  current highest bidder: a
  date posted: 2020-12-03 14:55:46
  bid ending date: 2020-12-04 12:00:00
[Item 2]
  description: 22
  status: 2 bids
  current bidding price: 44
  current highest bidder: asdf
  date posted: 2020-12-03 14:56:49
  bid ending date: 2020-12-22 20:20:00
--- Which item do you want to bid? (Enter the number or 'B' to go back to the previous menu):

```


5.5 Check bid

```
----< Main menu > : (Enter the number)
----(1) Sell item
----(2) Status of Your Item Listed on Auction
----(3) Search item
----(4) Check Status of your Bid
----(5) Check your Account
----(6) Quit
4
[Item 1]
  description: 111
  status: You are the highest bidder.
  your bidding price: 2
  current highest bidding price: 2
  bid ending date: 2020-12-04 12:00:00
```

If you select check bid, then you can watch your bidding list.

5.6 Check Account

```
----< Main menu > : (Enter the number)
----(1) Sell item
----(2) Status of Your Item Listed on Auction
----(3) Search item
----(4) Check Status of your Bid
----(5) Check your Account
----(6) Quit
5
----< Check your Account >

[Purchased Item 1]
  description: 123
  purchase price: 456

[Your Balance Summary]
  sold: 0 won
  commission: 0 won
  purchased: -456 won
Total balance: -456 won
```

If you select check account, you can watch sold item list, and purchased item list. Also you can see your total money.