REPORT

보고서 작성 서약서

1. 나는 타학생의 보고서를 복사(Copy)하지 않았습니다.

2. 나는 타학생의 보고서를 인터넷에서 다운로드 하여 대체하지 않았습니다.

3. 나는 타인에게 보고서 제출 전에 보고서를 보여주지 않았습니다.

4. 보고서 제출 기한을 준수하였습니다.

나는 보고서 작성시 위법 행위를 하지 않고,
성.균.인으로서 나의 명예를 지킬 것을 약속합니다.

과　　목 : Introduction to Database
담당교수: 남범석
학　　과 : 반도체시스템공학과
학　　번 : 2018314788
이　　름: 오해성
제 출 일 : 2020.12.02

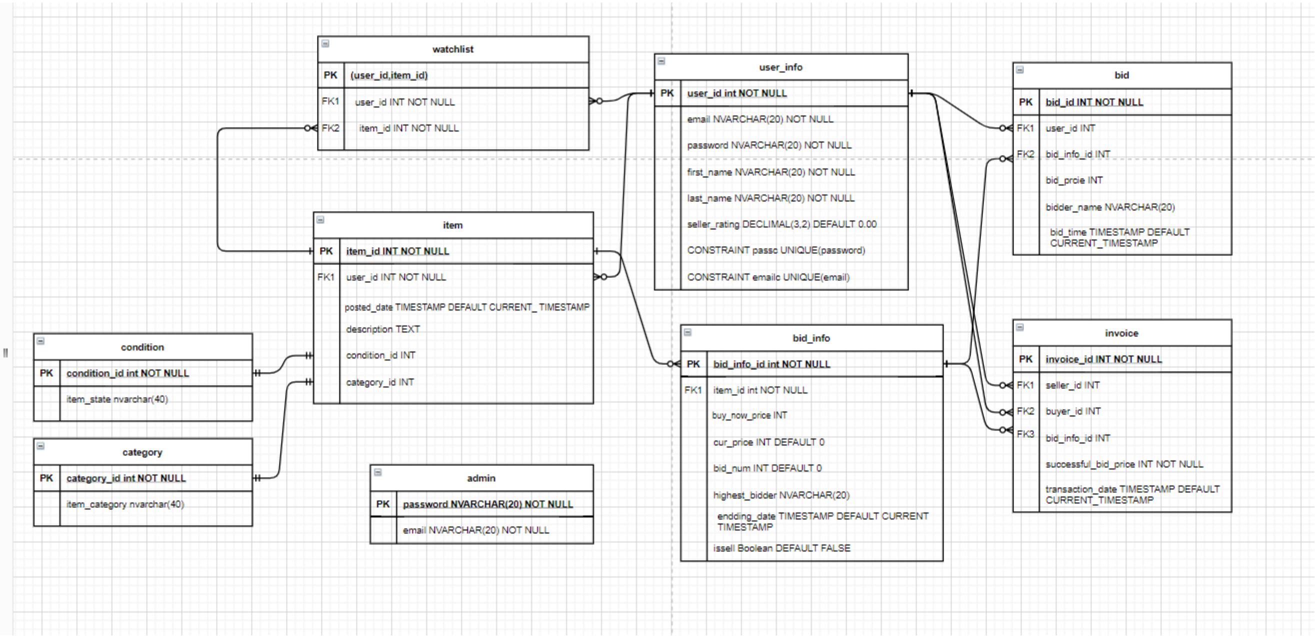
Introduction

This document covers the database design of the auction system given as a term project and the development of the user interface program.

The order of the documents is as follows.

1. Database schema and description with figure
2. DDL Trigger,Event,Procedure Description
3. DML states for queues
4. User manual

Database schema and description


watchlist
PK
(user_id,item_id)
FK1
user_id INT NOT NULL
FK2
item_id INT NOT NULL
user_info
PK
user_id int NOT NULL
email NVARCHAR(20) NOT NULL
password NVARCHAR(20) NOT NULL
first_name NVARCHAR(20) NOT NULL
last_name NVARCHAR(20) NOT NULL
seller_rating DECLIMAL(3,2) DEFAULT 0.00
CONSTRAINT passc UNIQUE(password)
CONSTRAINT emailc UNIQUE(email)
bid
PK
bid_id INT NOT NULL
FK1
user_id INT
FK2
bid_info_id INT
bid_prcie INT
bidder_name NVARCHAR(20)
bid_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP
item
PK
item_id INT NOT NULL
FK1
user_id INT NOT NULL
posted_date TIMESTAMP DEFAULT CURRENT_ TIMESTAMP
description TEXT
condition_id INT
category_id INT
condition
PK
condition_id int NOT NULL
item_state nvarchar(40)
category
PK
category_id int NOT NULL
item_category nvarchar(40)
admin
PK
password NVARCHAR(20) NOT NULL
email NVARCHAR(20) NOT NULL
bid_info
PK
bid_info_id int NOT NULL
FK1
item_id int NOT NULL
buy_now_price INT
cur_price INT DEFAULT 0
bid_num INT DEFAULT 0
highest_bidder NVARCHAR(20)
endding_date TIMESTAMP DEFAULT CURRENT TIMESTAMP
issell Boolean DEFAULT FALSE
invoice
PK
invoice_id INT NOT NULL
FK1
seller_id INT
FK2
buyer_id INT
FK3
bid_info_id INT
successful_bid_price INT NOT NULL
transaction_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP

overview

By default, most tables use the Surrogate key as their primary key

because data is likely to overlap,

assuming that the auction system has a large number of users.

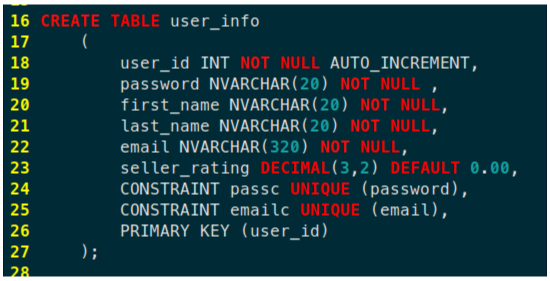
Normal form of databases is 2NF

Bider_name on the Bid table failed to satisfy 3NF

because it was dependent on both bid_id and user_id.

user_info

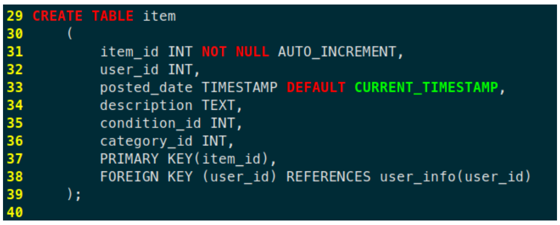
- Due to the characteristics of log-in, email and password are unique
- first_name,last_name decompositon for 1NF



```
CREATE TABLE user_info
    (
        user_id INT NOT NULL AUTO_INCREMENT,
        password NVARCHAR(20) NOT NULL ,
        first_name NVARCHAR(20) NOT NULL,
        last_name NVARCHAR(20) NOT NULL,
        email NVARCHAR(320) NOT NULL,
        seller_rating DECIMAL(3,2) DEFAULT 0.00,
        CONSTRAINT passc UNIQUE (password),
        CONSTRAINT emailc UNIQUE (email),
        PRIMARY KEY (user_id)
    );
```

item

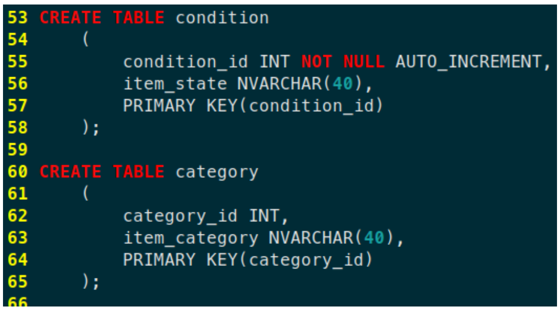
- Because the item has one to many relationships with the user_info, the foregin key is set to user_id
- Posted dated automatically set by CURRENT_TIMESTAMP()
- Instead of adding condition and category directly, make a table and connect it through condition_id and category_id values (one-to-one)



```
CREATE TABLE item
    (
        item_id INT NOT NULL AUTO_INCREMENT,
        user_id INT,
        posted_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
        description TEXT,
        condition_id INT,
        category_id INT,
        PRIMARY KEY(item_id),
        FOREIGN KEY (user_id) REFERENCES user_info(user_id)
    );

```

condition,category

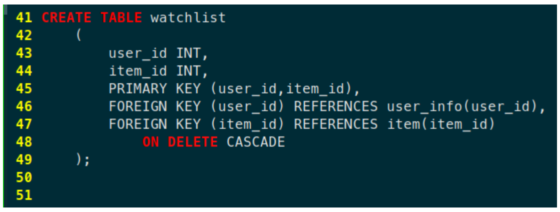
- this tables have state of item
- value of this table is predetermined by admin



```
CREATE TABLE condition
    (
        condition_id INT NOT NULL AUTO_INCREMENT,
        item_state NVARCHAR(40),
        PRIMARY KEY(condition_id)
    );

CREATE TABLE category
    (
        category_id INT,
        item_category NVARCHAR(40),
        PRIMARY KEY(category_id)
    );
```

watchlist

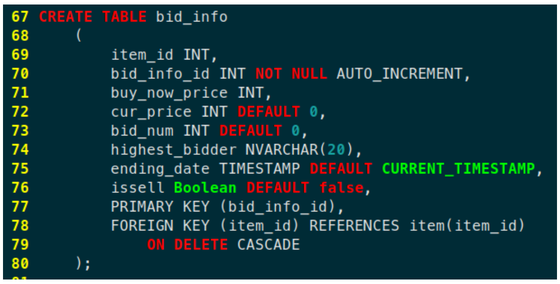
- Since items and users have many-to-many relationships, create this tables to connect to one-to-many relationships.



```
CREATE TABLE watchlist
    (
        user_id INT,
        item_id INT,
        PRIMARY KEY (user_id,item_id),
        FOREIGN KEY (user_id) REFERENCES user_info(user_id),
        FOREIGN KEY (item_id) REFERENCES item(item_id)
            ON DELETE CASCADE
    );


```

bid_info

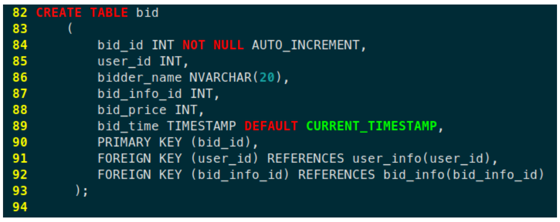
- bid_info table has auction information for one item
- When a user bids for an item, the value of bid_num,highest_bidder,cur_price,issell,ending_date is updated according to the conditions by trigger.
- Every minute, mysql event calls a procedure that terminates the auction if the ending_date value is earlier than the current time.
- If the issell value is updated to true, add a row to the invoice table.



```
CREATE TABLE bid_info
    (
        item_id INT,
        bid_info_id INT NOT NULL AUTO_INCREMENT,
        buy_now_price INT,
        cur_price INT DEFAULT 0,
        bid_num INT DEFAULT 0,
        highest_bidder NVARCHAR(20),
        ending_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
        issell Boolean DEFAULT false,
        PRIMARY KEY (bid_info_id),
        FOREIGN KEY (item_id) REFERENCES item(item_id)
            ON DELETE CASCADE
    );
```

bid

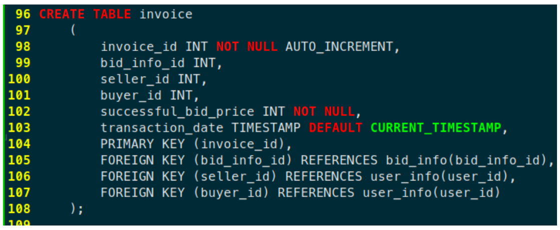
- If the user bids, add the information to this table.
- It would be better to delete the bidder_name column and obtain the bidder name by user_id.
- Because bid is many-to-one relationship with user and bid_info,
  foreign key is set both in user_id and bid_info_id



```sql
CREATE TABLE bid
    (
        bid_id INT NOT NULL AUTO_INCREMENT,
        user_id INT,
        bidder_name NVARCHAR(20),
        bid_info_id INT,
        bid_price INT,
        bid_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
        PRIMARY KEY (bid_id),
        FOREIGN KEY (user_id) REFERENCES user_info(user_id),
        FOREIGN KEY (bid_info_id) REFERENCES bid_info(bid_info_id)
     );

```

invoice

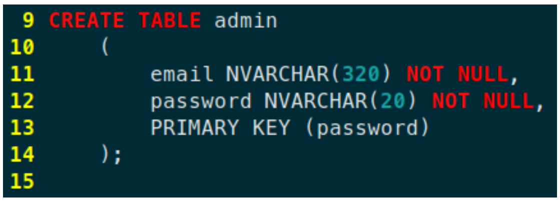
- if the item is sold, the invoice table will contain the id and information of the buyer and seller.
- transaction_date determines the exact time of termination of the auction



```sql
CREATE TABLE invoice
    (
        invoice_id INT NOT NULL AUTO_INCREMENT,
        bid_info_id INT,
        seller_id INT,
        buyer_id INT,
        successful_bid_price INT NOT NULL,
        transaction_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
        PRIMARY KEY (invoice_id),
        FOREIGN KEY (bid_info_id) REFERENCES bid_info(bid_info_id),
        FOREIGN KEY (seller_id) REFERENCES user_info(user_id),
        FOREIGN KEY (buyer_id) REFERENCES user_info(user_id)
    );
```

admin

- It contains information on the administrator's account
  that has been predetermined for access to the admin menu.

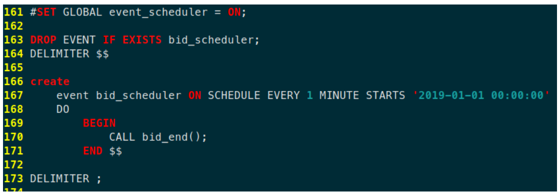


```sql
CREATE TABLE admin
    (
        email NVARCHAR(320) NOT NULL,
        password NVARCHAR(20) NOT NULL,
        PRIMARY KEY (password)
    );

```

DDL Trigger,Event,Procedure Description

EVENT

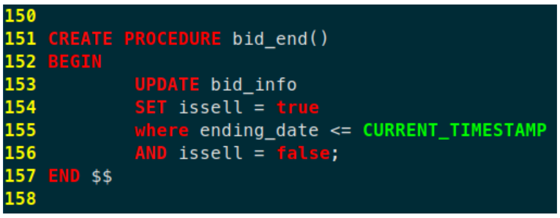
- The bid_scheuler event calls a procedure every minute to determine if the auction is over



```
#SET GLOBAL event_scheduler = ON;

DROP EVENT IF EXISTS bid_scheduler;
DELIMITER $$

create
    event bid_scheduler ON SCHEDULE EVERY 1 MINUTE STARTS '2019-01-01 00:00:00'
    DO
        BEGIN
            CALL bid_end();
        END $$

DELIMITER ;
```

PROCEDURE

- The procedure updates the issell for bid_info whose end date is past.
- After that item _sold trigger occurs.

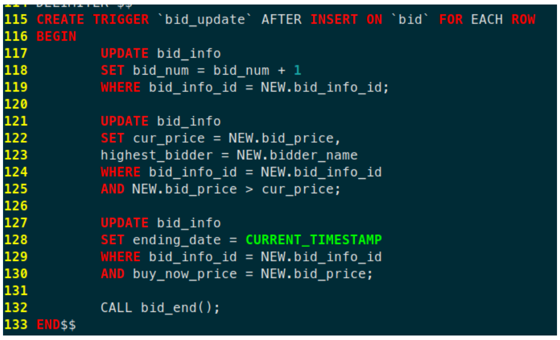


```

CREATE PROCEDURE bid_end()
BEGIN
        UPDATE bid_info
        SET issell = true
        where ending_date <= CURRENT_TIMESTAMP
        AND issell = false;
END $$

```

TRIGGER

▼ bid_update

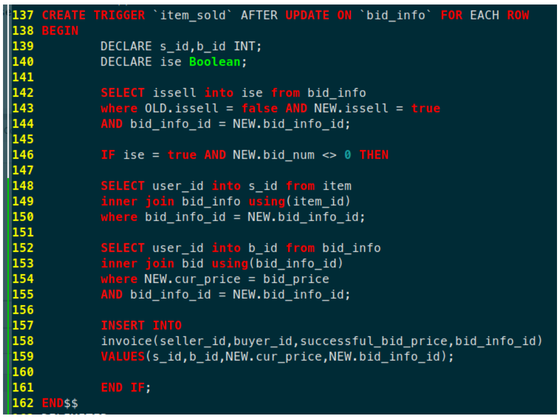
- The bid_update trigger updates the information in the bid_info table after the bid is added.
- If the bid price is the same as the current bid price change the bid_ending value to
  the current time and call the bid_end procedure



```
CREATE TRIGGER `bid_update` AFTER INSERT ON `bid` FOR EACH ROW
BEGIN
        UPDATE bid_info
        SET bid_num = bid_num + 1
        WHERE bid_info_id = NEW.bid_info_id;

        UPDATE bid_info
        SET cur_price = NEW.bid_price,
        highest_bidder = NEW.bidder_name
        WHERE bid_info_id = NEW.bid_info_id
        AND NEW.bid_price > cur_price;

        UPDATE bid_info
        SET ending_date = CURRENT_TIMESTAMP
        WHERE bid_info_id = NEW.bid_info_id
        AND buy_now_price = NEW.bid_price;

        CALL bid_end();
END$$
```

▼ item_sold

- If the issell value is updated from false to true them add row to the invoice.
- This implementaion has a large Internal overhead so needs to be modified



```
CREATE TRIGGER `item_sold` AFTER UPDATE ON `bid_info` FOR EACH ROW
BEGIN
        DECLARE s_id,b_id INT;
        DECLARE ise Boolean;

        SELECT issell into ise from bid_info
        where OLD.issell = false AND NEW.issell = true
        AND bid_info_id = NEW.bid_info_id;

        IF ise = true AND NEW.bid_num <> 0 THEN

        SELECT user_id into s_id from item
        inner join bid_info using(item_id)
        where bid_info_id = NEW.bid_info_id;

        SELECT user_id into b_id from bid_info
        inner join bid using(bid_info_id)
        where NEW.cur_price = bid_price
        AND bid_info_id = NEW.bid_info_id;

        INSERT INTO
        invoice(seller_id,buyer_id,successful_bid_price,bid_info_id)
        VALUES(s_id,b_id,NEW.cur_price,NEW.bid_info_id);

        END IF;
END$$
```

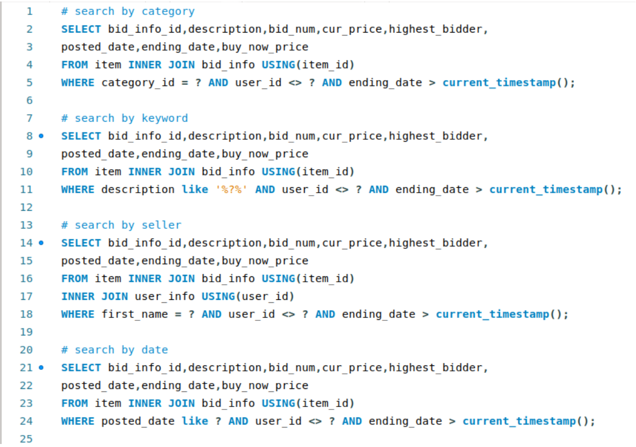
DML statements for queries

OVERVEIW

- The parts marked with question marks are the values determined at program runtime.
- The result value of the query is analyzed by search_engine class to output useful result to the user.

DML for search

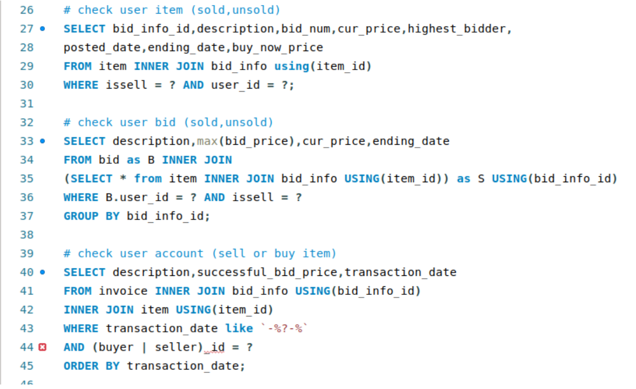
- The results of the search do not include the products of the users and expired products.
- user_input(choice) insert into first question mark with proper transfrom



```sql
# search by category
SELECT bid_info_id,description,bid_num,cur_price,highest_bidder,
posted_date,ending_date,buy_now_price
FROM item INNER JOIN bid_info USING(item_id)
WHERE category_id = ? AND user_id <> ? AND ending_date > current_timestamp();

# search by keyword
SELECT bid_info_id,description,bid_num,cur_price,highest_bidder,
posted_date,ending_date,buy_now_price
FROM item INNER JOIN bid_info USING(item_id)
WHERE description like '%?%' AND user_id <> ? AND ending_date > current_timestamp();

# search by seller
SELECT bid_info_id,description,bid_num,cur_price,highest_bidder,
posted_date,ending_date,buy_now_price
FROM item INNER JOIN bid_info USING(item_id)
INNER JOIN user_info USING(user_id)
WHERE first_name = ? AND user_id <> ? AND ending_date > current_timestamp();

# search by date
SELECT bid_info_id,description,bid_num,cur_price,highest_bidder,
posted_date,ending_date,buy_now_price
FROM item INNER JOIN bid_info USING(item_id)
WHERE posted_date like ? AND user_id <> ? AND ending_date > current_timestamp();
```

DML for user check

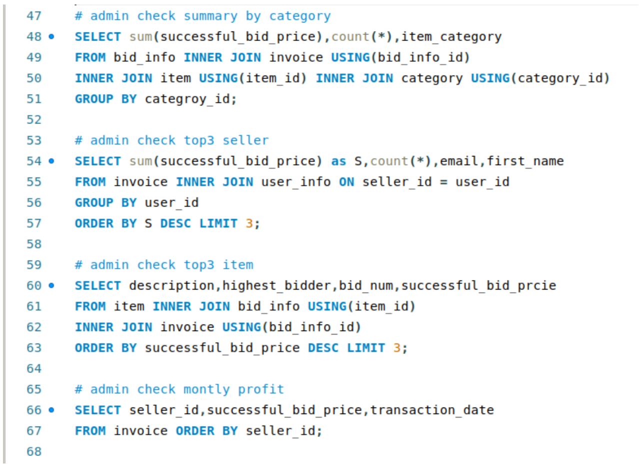
- The first query obtains the result by separating the products posted by the user into those sold and those not sold.
- The second query joins the bid_info, item, and bid to know the highest price and current status of the bid by the user.
- The third query joins invoice, bid_info,item to obtain information about the products sold and purchased by the user.



```sql
# check user item (sold,unsold)
SELECT bid_info_id,description,bid_num,cur_price,highest_bidder,
posted_date,ending_date,buy_now_price
FROM item INNER JOIN bid_info using(item_id)
WHERE issell = ? AND user_id = ?;

# check user bid (sold,unsold)
SELECT description,max(bid_price),cur_price,ending_date
FROM bid as B INNER JOIN
(SELECT * from item INNER JOIN bid_info USING(item_id)) as S USING(bid_info_id)
WHERE B.user_id = ? AND issell = ?
GROUP BY bid_info_id;

# check user account (sell or buy item)
SELECT description,successful_bid_price,transaction_date
FROM invoice INNER JOIN bid_info USING(bid_info_id)
INNER JOIN item USING(item_id)
WHERE transaction_date like `-%?-%`
AND (buyer | seller)_id = ?
ORDER BY transaction_date;
```

DML for admin

- The first query group by category id to know the sales volume and amount by category.
- The second query joins invoice and user_id to get information from the top three sellers.
- The third query sorted by winning bid price to know the top three items.
- The fourth query sorted by seller_id to calculate the discount rate for each user.

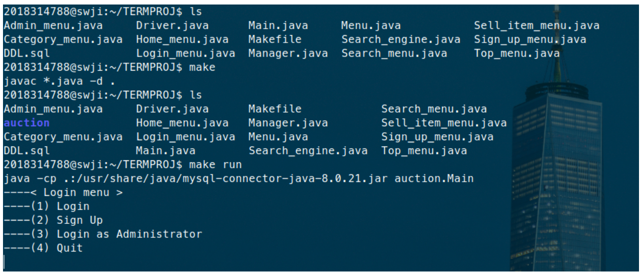


```sql
# admin check summary by category
SELECT sum(successful_bid_price),count(*),item_category
FROM bid_info INNER JOIN invoice USING(bid_info_id)
INNER JOIN item USING(item_id) INNER JOIN category USING(category_id)
GROUP BY categroy_id;

# admin check top3 seller
SELECT sum(successful_bid_price) as S,count(*),email,first_name
FROM invoice INNER JOIN user_info ON seller_id = user_id
GROUP BY user_id
ORDER BY S DESC LIMIT 3;

# admin check top3 item
SELECT description,highest_bidder,bid_num,successful_bid_prcie
FROM item INNER JOIN bid_info USING(item_id)
INNER JOIN invoice USING(bid_info_id)
ORDER BY successful_bid_price DESC LIMIT 3;

# admin check montly profit
SELECT seller_id,successful_bid_price,transaction_date
FROM invoice ORDER BY seller_id;
```

USER MANUAL

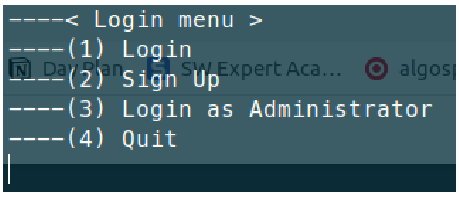
💡 Since Makefile is included with Java file, the procedure for running the program is as follows

$ make // compile java file
$ make run // run java class
$ make clean // remove class directory



```
2018314788@swji:~/TERMPROJ$ ls
Admin_menu.java     Driver.java      Main.java     Menu.java           Sell_item_menu.java
Category_menu.java  Home_menu.java   Makefile      Search_engine.java  Sign_up_menu.java
DDL.sql             Login_menu.java  Manager.java  Search_menu.java    Top_menu.java
2018314788@swji:~/TERMPROJ$ make
javac *.java -d .
2018314788@swji:~/TERMPROJ$ ls
Admin_menu.java     Driver.java      Makefile            Search_menu.java
auction             Home_menu.java   Manager.java        Sell_item_menu.java
Category_menu.java  Login_menu.java  Menu.java           Sign_up_menu.java
DDL.sql             Main.java        Search_engine.java  Top_menu.java
2018314788@swji:~/TERMPROJ$ make run
java -cp .:/usr/share/java/mysql-connector-java-8.0.21.jar auction.Main
----< Login menu >
----(1) Login
----(2) Sign Up
----(3) Login as Administrator
----(4) Quit
```

First menu

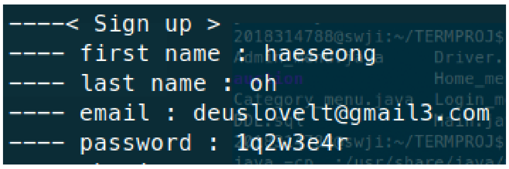


```
----< Login menu >
----(1) Login
----(2) Sign Up
----(3) Login as Administrator
----(4) Quit
```

Users can enter the next menu by entering numbers 1 through 4.

If another string or integer value is received wait for the correct input without entering the next menu

Signup menu



```
----< Sign up >
---- first name : haeseong
---- last name : oh
---- email : deuslovelt@gmail3.com
---- password : 1q2w3e4r
```

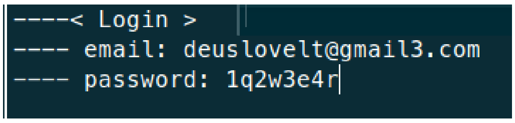
Enter the user's name and email password.

If there is a duplicate email or password in the database

the message is printed and sign up menu show again.

after successful enter of information enter the login menu

Login menu



```
----< Login >
---- email: deuslovelt@gmail3.com
---- password: 1q2w3e4r
```

Enter the user's email and password. If the input is incorrect, repeat the login menu again.

Login as Administrator menu is same with next menu but

check the email and password in admin database.

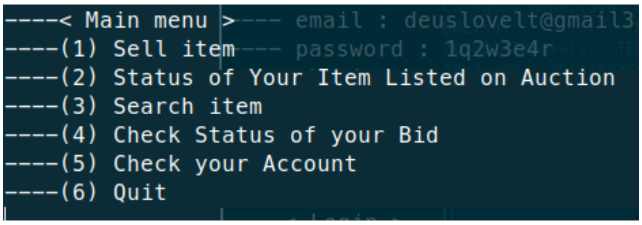
The pre-added admin account information is as follows.

⚙ email : ohsolution
pw : 2018314788

Main menu

After logging in, the user enters the main menu with five functions.

If you enter a value other than a number from 1 to 6, program continue to receive the value



```
----< Main menu >
----(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
```

Sell item menu

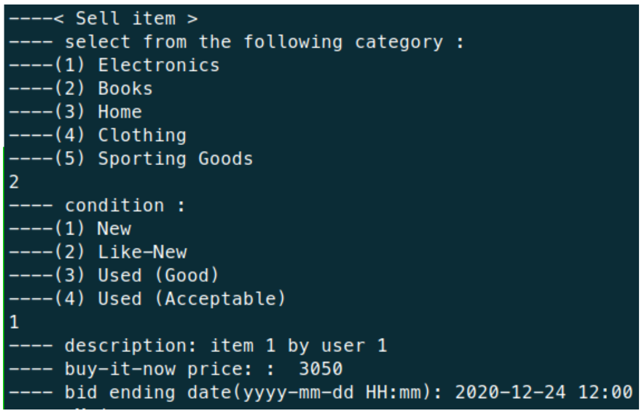
In the main menu, you can press 1 to enter the sell item menu.

After entering the category and status of the item by number,

you have to enter the description as a single string and

enter the buy-it-now price and bid_ending date according to the format

After successful input is recevied, program registers the item in the database.

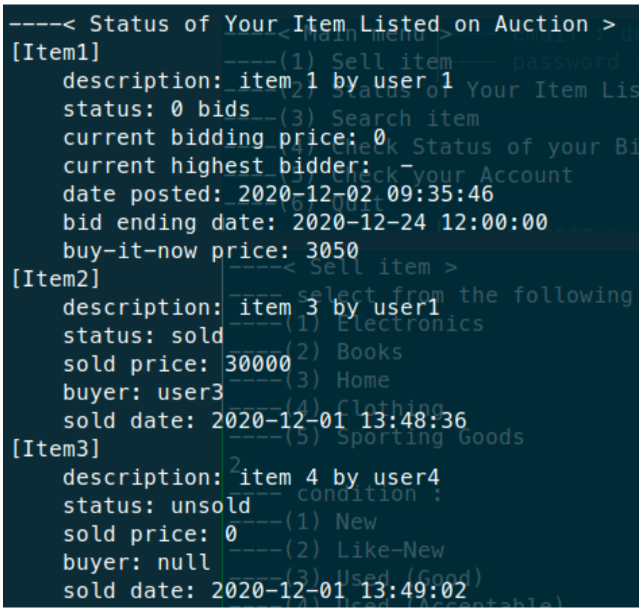


```
----< Sell item >
---- select from the following category :
----(1) Electronics
----(2) Books
----(3) Home
----(4) Clothing
----(5) Sporting Goods
2
---- condition :
----(1) New
----(2) Like-New
----(3) Used (Good)
----(4) Used (Acceptable)
1
---- description: item 1 by user 1
---- buy-it-now price: :  3050
---- bid ending date(yyyy-mm-dd HH:mm): 2020-12-24 12:00
```

Status of user item

In the main menu, you can press 2 to check your item status

The description and status of the item appear as a result.

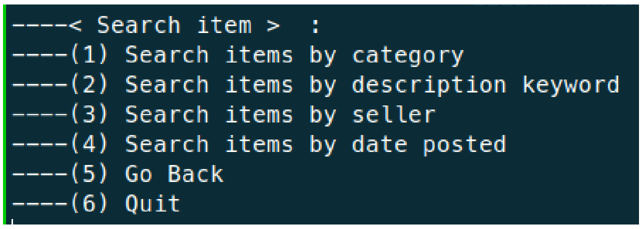


```
----< Status of Your Item Listed on Auction >
[Item1]
    description: item 1 by user 1
    status: 0 bids
    current bidding price: 0
    current highest bidder:  -
    date posted: 2020-12-02 09:35:46
    bid ending date: 2020-12-24 12:00:00
    buy-it-now price: 3050
[Item2]
    description: item 3 by user1
    status: sold
    sold price: 30000
    buyer: user3
    sold date: 2020-12-01 13:48:36
[Item3]
    description: item 4 by user4
    status: unsold
    sold price: 0
    buyer: null
    sold date: 2020-12-01 13:49:02
```

Search item menu

In the main menu, you can press 3 to search item on sale posted by other users

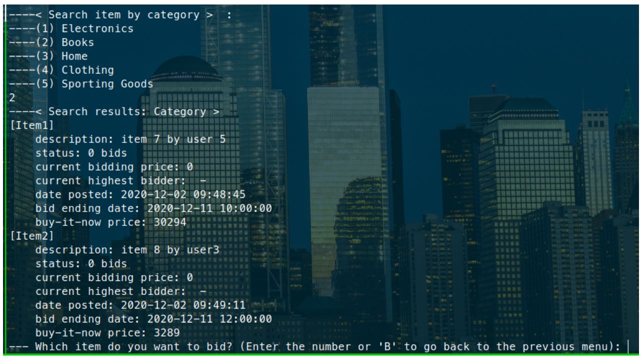
to get result you have to chose which basis you take



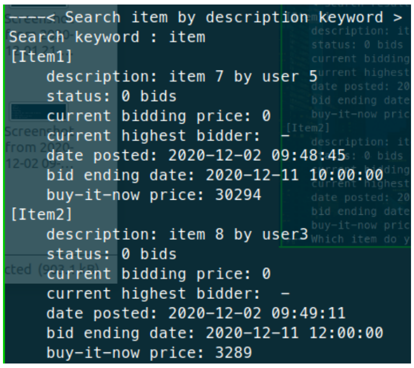
```
----< Search item >  :
----(1) Search items by category
----(2) Search items by description keyword
----(3) Search items by seller
----(4) Search items by date posted
----(5) Go Back
----(6) Quit
```

Search by basis

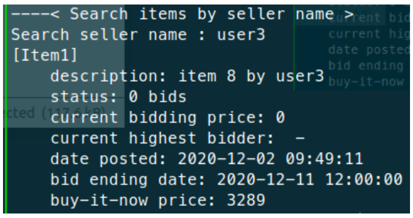
1. category search : take one category to search
2. keyword search : Search all items that contain the keyword entered by the user in item description
3. seller search : Search all items that seller has same first name with user input
4. date search : Search for items such that user input is same with posted date



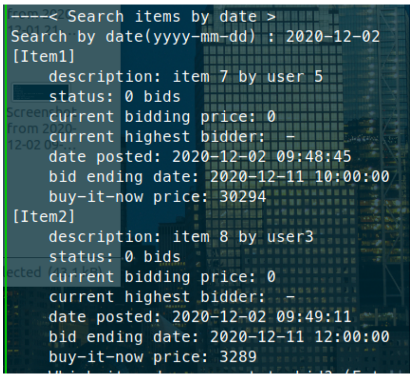
```
----< Search item by category >  :
----(1) Electronics
----(2) Books
----(3) Home
----(4) Clothing
----(5) Sporting Goods
2
----< Search results: Category >
[Item1]
    description: item 7 by user 5
    status: 0 bids
    current bidding price: 0
    current highest bidder:  -
    date posted: 2020-12-02 09:48:45
    bid ending date: 2020-12-11 10:00:00
    buy-it-now price: 30294
[Item2]
    description: item 8 by user3
    status: 0 bids
    current bidding price: 0
    current highest bidder:  -
    date posted: 2020-12-02 09:49:11
    bid ending date: 2020-12-11 12:00:00
    buy-it-now price: 3289
--- Which item do you want to bid? (Enter the number or 'B' to go back to the previous menu):
```



```
----< Search item by description keyword >
Search keyword : item
[Item1]
    description: item 7 by user 5
    status: 0 bids
    current bidding price: 0
    current highest bidder:  -
    date posted: 2020-12-02 09:48:45
    bid ending date: 2020-12-11 10:00:00
    buy-it-now price: 30294
[Item2]
    description: item 8 by user3
    status: 0 bids
    current bidding price: 0
    current highest bidder:  -
    date posted: 2020-12-02 09:49:11
    bid ending date: 2020-12-11 12:00:00
    buy-it-now price: 3289
```



```
----< Search items by seller name >
Search seller name : user3
[Item1]
    description: item 8 by user3
    status: 0 bids
    current bidding price: 0
    current highest bidder:  -
    date posted: 2020-12-02 09:49:11
    bid ending date: 2020-12-11 12:00:00
    buy-it-now price: 3289
```

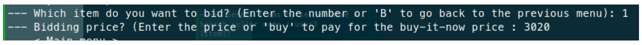


```
----< Search items by date >
Search by date(yyyy-mm-dd) : 2020-12-02
[Item1]
    description: item 7 by user 5
    status: 0 bids
    current bidding price: 0
    current highest bidder:  -
    date posted: 2020-12-02 09:48:45
    bid ending date: 2020-12-11 10:00:00
    buy-it-now price: 30294
[Item2]
    description: item 8 by user3
    status: 0 bids
    current bidding price: 0
    current highest bidder:  -
    date posted: 2020-12-02 09:49:11
    bid ending date: 2020-12-11 12:00:00
    buy-it-now price: 3289
```

After search user can bid for item

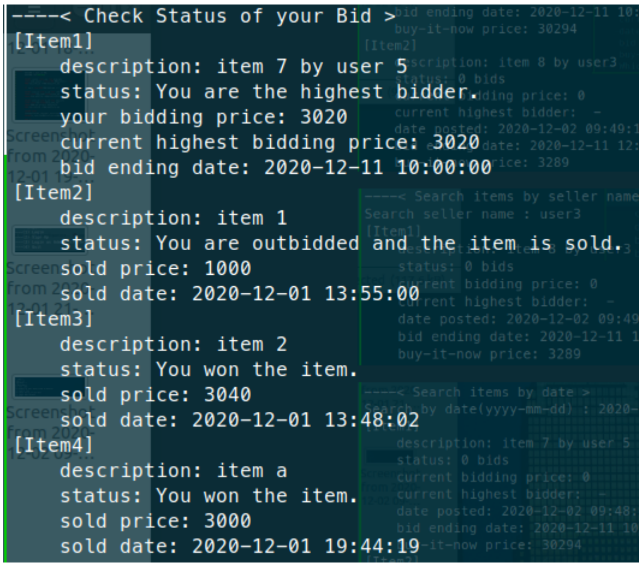
Bidding price only accept when range is

(cur_price,buy-it-now price]



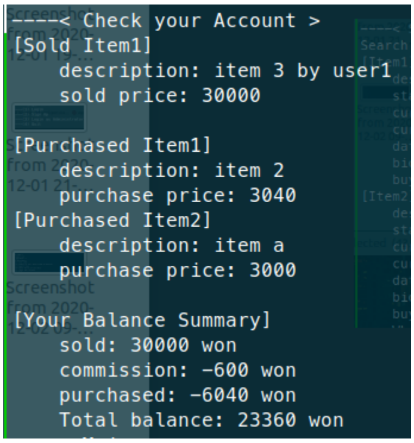
```
--- Which item do you want to bid? (Enter the number or 'B' to go back to the previous menu): 1
--- Bidding price? (Enter the price or 'buy' to pay for the buy-it-now price : 3020
```

Check status user bid



```
----< Check Status of your Bid >
[Item1]
    description: item 7 by user 5
    status: You are the highest bidder.
    your bidding price: 3020
    current highest bidding price: 3020
    bid ending date: 2020-12-11 10:00:00
[Item2]
    description: item 1
    status: You are outbidded and the item is sold.
    sold price: 1000
    sold date: 2020-12-01 13:55:00
[Item3]
    description: item 2
    status: You won the item.
    sold price: 3040
    sold date: 2020-12-01 13:48:02
[Item4]
    description: item a
    status: You won the item.
    sold price: 3000
    sold date: 2020-12-01 19:44:19
```

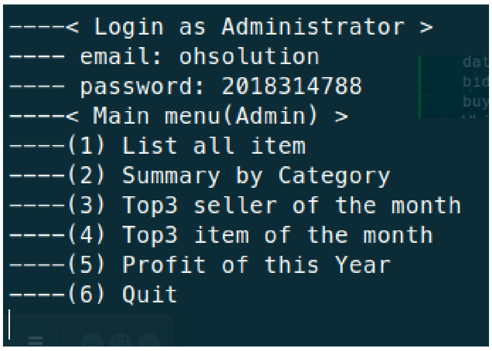
Check user account



```
----< Check your Account >
[Sold Item1]
    description: item 3 by user1
    sold price: 30000

[Purchased Item1]
    description: item 2
    purchase price: 3040
[Purchased Item2]
    description: item a
    purchase price: 3000

[Your Balance Summary]
    sold: 30000 won
    commission: -600 won
    purchased: -6040 won
    Total balance: 23360 won
```

Admin menu

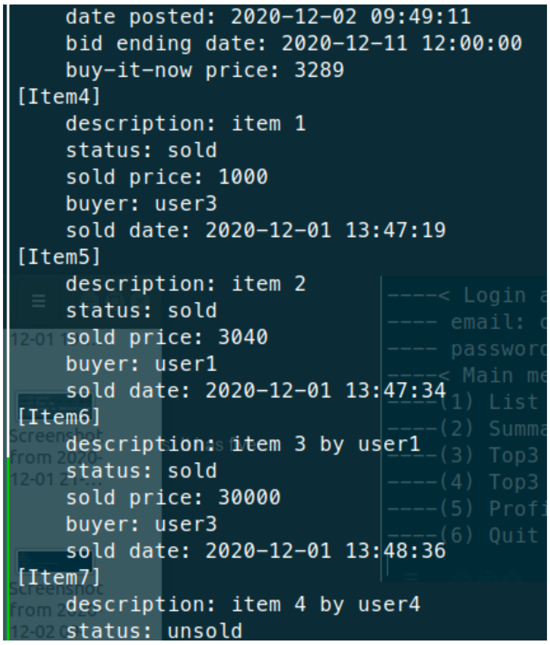
In the first menu, press 3 to log in to the administrator account, Admin main menu appears. It has five functions.



```
----< Login as Administrator >
---- email: ohsolution
---- password: 2018314788
----< Main menu(Admin) >
----(1) List all item
----(2) Summary by Category
----(3) Top3 seller of the month
----(4) Top3 item of the month
----(5) Profit of this Year
----(6) Quit
```

List all item

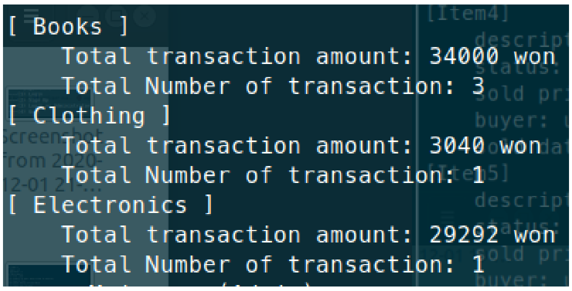
Outputs the information and status of all items registered in the database.



```
    date posted: 2020-12-02 09:49:11
    bid ending date: 2020-12-11 12:00:00
    buy-it-now price: 3289
[Item4]
    description: item 1
    status: sold
    sold price: 1000
    buyer: user3
    sold date: 2020-12-01 13:47:19
[Item5]
    description: item 2
    status: sold
    sold price: 3040
    buyer: user1
    sold date: 2020-12-01 13:47:34
[Item6]
    description: item 3 by user1
    status: sold
    sold price: 30000
    buyer: user3
    sold date: 2020-12-01 13:48:36
[Item7]
    description: item 4 by user4
    status: unsold
```

Summary by category

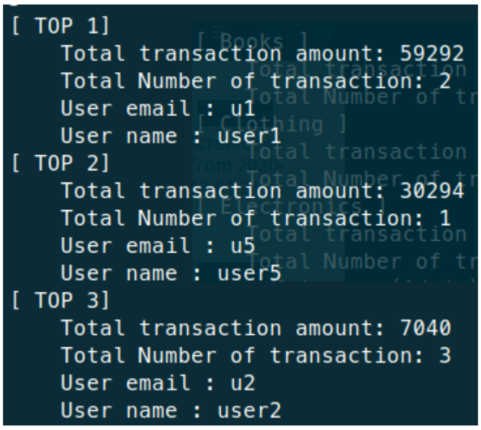
The total volume and amount of transactions are printed by classifying them into categories.



```
[ Books ]
    Total transaction amount: 34000 won
    Total Number of transaction: 3
[ Clothing ]
    Total transaction amount: 3040 won
    Total Number of transaction: 1
[ Electronics ]
    Total transaction amount: 29292 won
    Total Number of transaction: 1
```

Top 3 seller

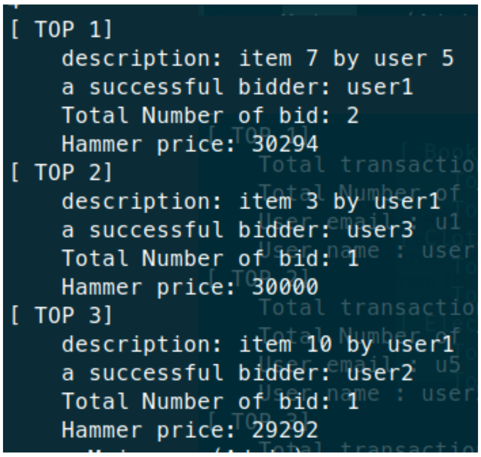
Outputs user information for the top three sales figures.



```
[ TOP 1]
    Total transaction amount: 59292
    Total Number of transaction: 2
    User email : u1
    User name : user1
[ TOP 2]
    Total transaction amount: 30294
    Total Number of transaction: 1
    User email : u5
    User name : user5
[ TOP 3]
    Total transaction amount: 7040
    Total Number of transaction: 3
    User email : u2
    User name : user2
```

Top 3 item

Outputs information on the top 3 items of transaction amount

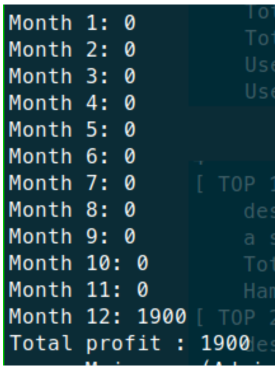


```
[ TOP 1]
    description: item 7 by user 5
    a successful bidder: user1
    Total Number of bid: 2
    Hammer price: 30294
[ TOP 2]
    description: item 3 by user1
    a successful bidder: user3
    Total Number of bid: 1
    Hammer price: 30000
[ TOP 3]
    description: item 10 by user1
    a successful bidder: user2
    Total Number of bid: 1
    Hammer price: 29292
```

Profit of this year

The company's profit generated from the commission

in the transaction is organized monthly and the total is printed out.



```
Month 1: 0
Month 2: 0
Month 3: 0
Month 4: 0
Month 5: 0
Month 6: 0
Month 7: 0
Month 8: 0
Month 9: 0
Month 10: 0
Month 11: 0
Month 12: 1900
Total profit : 1900
```