

# 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

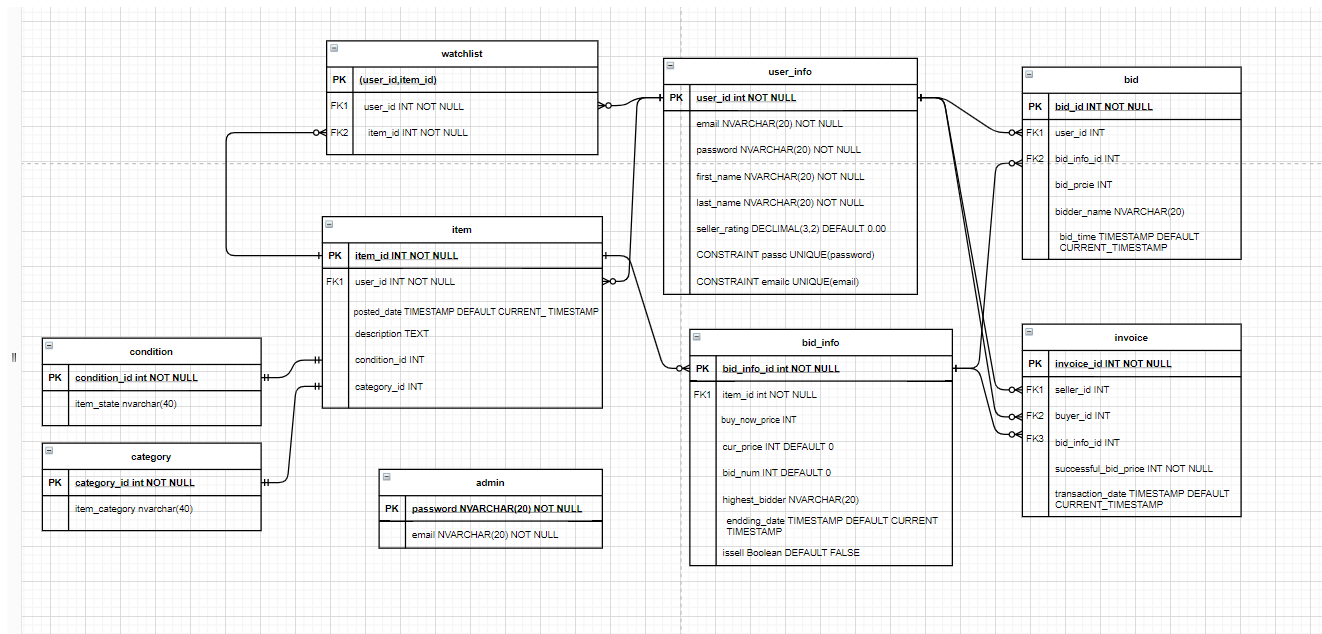
---

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



## 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 decomposition for 1NF

```
16 CREATE TABLE user_info
17 (
18     user_id INT NOT NULL AUTO_INCREMENT,
19     password NVARCHAR(20) NOT NULL,
20     first_name NVARCHAR(20) NOT NULL,
21     last_name NVARCHAR(20) NOT NULL,
22     email NVARCHAR(320) NOT NULL,
23     seller_rating DECIMAL(3,2) DEFAULT 0.00,
24     CONSTRAINT passc UNIQUE (password),
25     CONSTRAINT emailc UNIQUE (email),
26     PRIMARY KEY (user_id)
27 );
28
```

## item

- Because the item has one to many relationships with the user\_info, the foreign key is set to user\_id
- Posted date 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)

```
29 CREATE TABLE item
30 (
31     item_id INT NOT NULL AUTO_INCREMENT,
32     user_id INT,
33     posted_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
34     description TEXT,
35     condition_id INT,
36     category_id INT,
37     PRIMARY KEY(item_id),
38     FOREIGN KEY (user_id) REFERENCES user_info(user_id)
39 );
40
```

## condition,category

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

```
53 CREATE TABLE condition
54 (
55     condition_id INT NOT NULL AUTO_INCREMENT,
56     item_state NVARCHAR(40),
57     PRIMARY KEY(condition_id)
58 );
59
60 CREATE TABLE category
61 (
62     category_id INT,
63     item_category NVARCHAR(40),
64     PRIMARY KEY(category_id)
65 );
66
```

## watchlist

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

```
41 CREATE TABLE watchlist
42 (
43     user_id INT,
44     item_id INT,
45     PRIMARY KEY (user_id,item_id),
46     FOREIGN KEY (user_id) REFERENCES user_info(user_id),
47     FOREIGN KEY (item_id) REFERENCES item(item_id)
48     ON DELETE CASCADE
49 );
50
51
```

## 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.

```
67 CREATE TABLE bid_info
68 (
69     item_id INT,
70     bid_info_id INT NOT NULL AUTO_INCREMENT,
71     buy_now_price INT,
72     cur_price INT DEFAULT 0,
73     bid_num INT DEFAULT 0,
74     highest_bidder NVARCHAR(20),
75     ending_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
76     issell Boolean DEFAULT false,
77     PRIMARY KEY (bid_info_id),
78     FOREIGN KEY (item_id) REFERENCES item(item_id)
79     ON DELETE CASCADE
80 );
81
```

## 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

```
82 CREATE TABLE bid
83 (
84     bid_id INT NOT NULL AUTO_INCREMENT,
85     user_id INT,
86     bidder_name NVARCHAR(20),
87     bid_info_id INT,
88     bid_price INT,
89     bid_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
90     PRIMARY KEY (bid_id),
91     FOREIGN KEY (user_id) REFERENCES user_info(user_id),
92     FOREIGN KEY (bid_info_id) REFERENCES bid_info(bid_info_id)
93 );
94
```

## 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

```
96 CREATE TABLE invoice
97 (
98     invoice_id INT NOT NULL AUTO_INCREMENT,
99     bid_info_id INT,
100     seller_id INT,
101     buyer_id INT,
102     successful_bid_price INT NOT NULL,
103     transaction_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
104     PRIMARY KEY (invoice_id),
105     FOREIGN KEY (bid_info_id) REFERENCES bid_info(bid_info_id),
106     FOREIGN KEY (seller_id) REFERENCES user_info(user_id),
107     FOREIGN KEY (buyer_id) REFERENCES user_info(user_id)
108 );
109
```

## admin

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

```
9 CREATE TABLE admin
10 (
11     email NVARCHAR(320) NOT NULL,
12     password NVARCHAR(20) NOT NULL,
13     PRIMARY KEY (password)
14 );
15
```

# DDL Trigger,Event,Procedure Description

## EVENT

- The bid\_scheuler event calls a procedure every minute to determine if the auction is over

```
161 #SET GLOBAL event_scheduler = ON;
162
163 DROP EVENT IF EXISTS bid_scheduler;
164 DELIMITER $$
165
166 CREATE
167     event bid_scheduler ON SCHEDULE EVERY 1 MINUTE STARTS '2019-01-01 00:00:00'
168 DO
169     BEGIN
170         CALL bid_end();
171     END $$
172
173 DELIMITER ;
```

## PROCEDURE

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

```
150
151 CREATE PROCEDURE bid_end()
152 BEGIN
153     UPDATE bid_info
154     SET issell = true
155     where ending_date <= CURRENT_TIMESTAMP
156     AND issell = false;
157 END $$
158
```

## 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

```
115 CREATE TRIGGER `bid_update` AFTER INSERT ON `bid` FOR EACH ROW
116 BEGIN
117     UPDATE bid_info
118     SET bid_num = bid_num + 1
119     WHERE bid_info_id = NEW.bid_info_id;
120
121     UPDATE bid_info
122     SET cur_price = NEW.bid_price,
123     highest_bidder = NEW.bidder_name
124     WHERE bid_info_id = NEW.bid_info_id
125     AND NEW.bid_price > cur_price;
126
127     UPDATE bid_info
128     SET ending_date = CURRENT_TIMESTAMP
129     WHERE bid_info_id = NEW.bid_info_id
130     AND buy_now_price = NEW.bid_price;
131
132     CALL bid_end();
133 END$$
```

### ▼ item\_sold

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

```
137 CREATE TRIGGER `item_sold` AFTER UPDATE ON `bid_info` FOR EACH ROW
138 BEGIN
139     DECLARE s_id,b_id INT;
140     DECLARE ise Boolean;
141
142     SELECT issell into ise from bid_info
143     where OLD.issell = false AND NEW.issell = true
144     AND bid_info_id = NEW.bid_info_id;
145
146     IF ise = true AND NEW.bid_num <> 0 THEN
147
148     SELECT user_id into s_id from item
149     inner join bid_info using(item_id)
150     where bid_info_id = NEW.bid_info_id;
151
152     SELECT user_id into b_id from bid_info
153     inner join bid_info using(bid_info_id)
154     where NEW.cur_price = bid_price
155     AND bid_info_id = NEW.bid_info_id;
156
157     INSERT INTO
158     invoice(seller_id,buyer_id,successful_bid_price,bid_info_id)
159     VALUES(s_id,b_id,NEW.cur_price,NEW.bid_info_id);
160
161     END IF;
162 END$$
```

# DML statements for queries

## OVERVIEW

- 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 transform

```
1 # search by category
2 SELECT bid_info_id,description,bid_num,cur_price,highest_bidder,
3 posted_date,ending_date,buy_now_price
4 FROM item INNER JOIN bid_info USING(item_id)
5 WHERE category_id = ? AND user_id <> ? AND ending_date > current_timestamp();
6
7 # search by keyword
8 • SELECT bid_info_id,description,bid_num,cur_price,highest_bidder,
9 posted_date,ending_date,buy_now_price
10 FROM item INNER JOIN bid_info USING(item_id)
11 WHERE description like '%?' AND user_id <> ? AND ending_date > current_timestamp();
12
13 # search by seller
14 • SELECT bid_info_id,description,bid_num,cur_price,highest_bidder,
15 posted_date,ending_date,buy_now_price
16 FROM item INNER JOIN bid_info USING(item_id)
17 INNER JOIN user_info USING(user_id)
18 WHERE first_name = ? AND user_id <> ? AND ending_date > current_timestamp();
19
20 # search by date
21 • SELECT bid_info_id,description,bid_num,cur_price,highest_bidder,
22 posted_date,ending_date,buy_now_price
23 FROM item INNER JOIN bid_info USING(item_id)
24 WHERE posted_date like ? AND user_id <> ? AND ending_date > current_timestamp();
25
```

## 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.

```
26 # check user item (sold,unsold)
27 • SELECT bid_info_id,description,bid_num,cur_price,highest_bidder,
28 posted_date,ending_date,buy_now_price
29 FROM item INNER JOIN bid_info using(item_id)
30 WHERE issell = ? AND user_id = ?;
31
32 # check user bid (sold,unsold)
33 • SELECT description,max(bid_price),cur_price,ending_date
34 FROM bid as B INNER JOIN
35 (SELECT * from item INNER JOIN bid_info USING(item_id)) as S USING(bid_info_id)
36 WHERE B.user_id = ? AND issell = ?
37 GROUP BY bid_info_id;
38
39 # check user account (sell or buy item)
40 • SELECT description,successful_bid_price,transaction_date
41 FROM invoice INNER JOIN bid_info USING(bid_info_id)
42 INNER JOIN item USING(item_id)
43 WHERE transaction_date like '-%?-%'
44 AND (buyer | seller)_id = ?
45 ORDER BY transaction_date;
46
```

## 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.

```
47 # admin check summary by category
48 • SELECT sum(successful_bid_price),count(*),item_category
49 FROM bid_info INNER JOIN invoice USING(bid_info_id)
50 INNER JOIN item USING(item_id) INNER JOIN category USING(category_id)
51 GROUP BY category_id;
52
53 # admin check top3 seller
54 • SELECT sum(successful_bid_price) as S,count(*),email,first_name
55 FROM invoice INNER JOIN user_info ON seller_id = user_id
56 GROUP BY user_id
57 ORDER BY S DESC LIMIT 3;
58
59 # admin check top3 item
60 • SELECT description,highest_bidder,bid_num,successful_bid_price
61 FROM item INNER JOIN bid_info USING(item_id)
62 INNER JOIN invoice USING(bid_info_id)
63 ORDER BY successful_bid_price DESC LIMIT 3;
64
65 # admin check montly profit
66 • SELECT seller_id,successful_bid_price,transaction_date
67 FROM invoice ORDER BY seller_id;
68
```

# 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@swj1:~/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@swj1:~/TERMPROJ$ make
javac *.java -d .
2018314788@swj1:~/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@swj1:~/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

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

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

## Signup menu

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

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

## Login menu

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
```

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



## 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 >---- email : deuslovelt@gmail3
----(1) Sell item---- password : 1q2w3e4r
----(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 received, 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]
----(1) Sell item---- password
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]
----< Sell item >
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):
```

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
```

```
-----< 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
```

## 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
```

# FILE architecture

