

Database Systems

Project Deliverables 2

System Implementation

LUCIEN GEORGE - 260571775
OSSAMA AHMED - 260549558
ZEYAD SALEH - 260556530
ALIAA MOHARRAM - 260572705



MCGILL UNIVERSITY

Submitted at COMP 421 course

February 29, 2016

COMP 421 PROJECT DELIVERABLES 2

Project: G16 Database design project 2016

Document Version Number: 1.3

Date: February 29th 2016

Authors: Lucien George 260571775
Ossama Ahmed 260549558
Zeyad Saleh 260556530
Aliaa Moharram 260572705

1 TABLE OF CONTENTS

1	Table of Contents	2
2	Tasks II, III and IV: Creating and populating Tables.....	6
2.1	users	7
2.1.1	Creation Command.....	7
2.1.2	Description.....	7
2.1.3	Insertion Command	7
2.1.4	Populated Table	7
2.2	auction	8
2.2.1	Creation Command.....	8
2.2.2	Description.....	8
2.2.3	Insertion Command	8
2.2.4	Populated Table	8
2.3	bid	9
2.3.1	Creation Command.....	9
2.3.2	Description.....	9
2.3.3	Insertion Command	9
2.3.4	Populated Table	9
2.4	item.....	10
2.4.1	Creation Command.....	10
2.4.2	Description.....	10
2.4.3	Insertion Command	10
2.4.4	Populated Table	10
2.5	approves (Relation)	11
2.5.1	Creation Command.....	11
2.5.2	Description.....	11
2.5.3	Insertion Command	11
2.5.4	Populated Table	11
2.6	reviews.....	12
2.6.1	Creation Command.....	12

2.6.2	Description.....	12
2.6.3	Insertion Command	12
2.6.4	Populated Table.....	12
2.7	clothing	13
2.7.1	Command.....	13
2.7.2	Description.....	13
2.7.3	Insertion Command	13
2.7.4	Populated Table.....	13
2.8	books	14
2.8.1	Command.....	14
2.8.2	Description.....	14
2.8.3	Insertion Command	14
2.8.4	Populated Table	14
3	Task V: Queries.....	15
3.1	Get the history of items sold by a specific user	16
3.1.1	Discussion.....	16
3.1.2	Command.....	16
3.1.3	Breakdown and results.....	16
3.2	Select the categories that a buyer bids on the most	17
3.2.1	Discussion.....	17
3.2.2	Command.....	17
3.2.3	Breakdown and results.....	17
3.3	get all bids in a certain auction.....	19
3.3.1	Discussion.....	19
3.3.2	Command.....	19
3.3.3	Breakdown and results.....	19
3.4	Get the average bid price on a certain auction	20
3.4.1	Discussion.....	20
3.4.2	Command.....	20
3.4.3	Breakdown and results.....	20
3.5	Get all bids offered by a certain buyer in all auctions:	21
3.5.1	Discussion.....	21
3.5.2	Command.....	21
3.5.3	Breakdown and results.....	21
3.6	Return bids on all auctions ordered by descending price and ascending aid.....	22
3.6.1	Discussion.....	22
3.6.2	Command.....	22
3.6.3	Breakdown and results.....	22
3.7	Get the average bid done by a specific user on a certain category.....	23
3.7.1	Discussion.....	23
3.7.2	Command.....	23
3.7.3	Breakdown and results.....	23

3.8 Get the best bid price on all the user's items that they are selling along with the item's information.....	24
3.8.1 Discussion.....	24
3.8.2 Command.....	24
3.8.3 Breakdown and results.....	24
3.9 Get the history of the items bought by a specific user	25
3.9.1 Discussion.....	25
3.9.2 Command.....	25
3.9.3 Breakdown and results.....	25
4 Task VI: Interesting data modification commands	26
4.1 approving a bid and inserting the buyer and seller info into approves	26
4.1.1 Discussion.....	26
4.1.2 Command.....	26
4.1.3 Breakdown and results.....	26
4.2 Delete auctions after 6 months elapsing	27
4.2.1 Discussion.....	27
4.2.2 Command.....	27
4.2.3 Breakdown and results.....	27
4.3 Modify the ratings for all the users based on the reviews table	28
4.3.1 Discussion.....	28
4.3.2 Command.....	28
4.3.3 Breakdown and results.....	28
4.4 Change the status of all bids on items which already has an approved bid	29
4.4.1 Discussion.....	29
4.4.2 Command.....	29
4.4.3 Breakdown and results.....	29
4.5 Inserting a buyer review given the itemid of the item bought.....	30
4.5.1 Discussion.....	30
4.5.2 Command.....	30
4.5.3 Breakdown and results.....	30
5 Task VII: Create 2 views on top of the database schema	31
5.1 Limited access to user information	31
5.1.1 Command.....	31
5.1.2 The view	31
5.1.3 Updating the view	31
5.2 All Participants in all auctions	32
5.2.1 Command.....	32
5.2.2 The view	32
5.2.3 Updating the view	32
6 Task VIII: Check constraints	33
6.1 Reviews checks	33
6.1.1 Reviewee different than reviewer	33

6.1.2	The rating must not be less than 1 or larger than 5.....	33
6.2	Auction.....	33
6.3	Approves.....	33
7	Extra work: an android app that integrates some of the queries we discussed	34

2 TASKS II, III AND IV: CREATING AND POPULATING TABLES

The following are the list of all tables (7 entity sets + 1 relation + 2 views) that we have in our database.

cs421=> select * from cs421g16.				
cs421g16.approves	cs421g16.auctionparticipants	cs421g16.books	cs421g16.item	cs421g16.securedusers
cs421g16.auction	cs421g16.bid	cs421g16.clothing	cs421g16.reviews	cs421g16.users

users is the entity storing all relevant user/client information

auction is where an item is showcased for bidding and sale

bid is the entity set that allows a buyer to place a certain bid in a certain auction

item is the object being sold

approves is the relation that links a seller to his approved bidder

reviews is the feedback of a certain

clothing is a subcategory of item with its own attributes

books is a subcategory of item with its own attributes as well

In the next section we will be discussing these relations, their creation command, description, insertion queries as well as their population.

2.1 USERS

2.1.1 Creation Command

```
create table users(email varchar(100) primary key, fname varchar(100),lname varchar(100),dateofbirth date,gender varchar(50),university varchar(200),address varchar(300),rating integer,picture bytea,phonenumer varchar(20));
```

2.1.2 Description

```
|cs421=> \d users
      Table "cs421g16.users"
    Column |          Type          | Modifiers
-----+-----+-----+
email   | character varying(100) | not null
fname   | character varying(100) |
lname   | character varying(100) |
dateofbirth | date |
gender  | character varying(50) |
university | character varying(200) |
address  | character varying(300) |
rating   | integer |
picture  | bytea |
phonenumer | character varying(20) |

Indexes:
  "users_pkey" PRIMARY KEY, btree (email)
Check constraints:
  "users_rating_check" CHECK (rating > 0 AND rating < 6)
Referenced by:
  TABLE "approves" CONSTRAINT "approves_seller_email_fkey" FOREIGN KEY (seller_email) REFERENCES users(email)
  TABLE "approves" CONSTRAINT "buyer_foreign" FOREIGN KEY (buyer_email) REFERENCES users(email)
  TABLE "auction" CONSTRAINT "email_foreign" FOREIGN KEY (email) REFERENCES users(email)
  TABLE "bid" CONSTRAINT "foreign_email" FOREIGN KEY (email) REFERENCES users(email)
```

2.1.3 Insertion Command

```
insert into users values('lucien.george@mail.mcgill.ca','Lucien','George','09-16-1995','Male','McGill University','3450 Drummond',5,null,'+1(514) 430-8272');
```

```
insert into users values('josephakl95@gmail.com', 'Joseph', 'Akl', '04-25-1995', 'Male', 'McGill University', '3460 Peel', 4, Null, '+1(514) 445 7873');
```

2.1.4 Populated Table

```
|cs421=> select * from users;
      email | fname | lname | dateofbirth | gender | university | address | rating | picture | phonenumer
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
aliamoharram@hotmail.com | Aliaa | Moharram | 1995-01-08 | Female | McGill University | 3410 Peel | 5 | | +1(514) 243-6927
lucien.george@mail.mcgill.ca | Lucien | George | 1995-09-16 | Male | McGill University | 3450 Drummond | 5 | | +1(514) 430-8272
ossoama.samir.ahmed@gmail.com | Ossama Samir | Ahmed | 1992-10-21 | Male | McGill University | Samuel Home | 1 | | +1(514) 836-2217
bettina.kemme@mcgill.ca | Bettina | Kemme | 2016-01-01 | Female | McGill University | Prof home | 5 | | Not Available
zeyadnazemsaleh@gmail.com | Zeyad | Saleh | 1994-05-07 | Male | McGill University | Zeyad Home | 5 | | +1(514) 585-6769
josephakl95@gmail.com | Joseph | Akl | 1995-04-25 | Male | McGill University | 3460 Peel | 4 | | +1(514) 445 7873
(6 rows)
```

2.2 AUCTION

2.2.1 Creation Command

```
CREATE SEQUENCE auction_id_seq MINVALUE 1;
```

```
create table auction(aid integer primary key,initialprice money,starttime time SET DEFAULT current_time,date date SET DEFAULT current_date,status varchar(50) ,email varchar(200),itemid integer,foreign key (email) references users(email), foreign key (itemid) references item(itemid));
```

```
ALTER TABLE auction ALTER aid SET DEFAULT nextval('auction_id_seq')
```

```
ALTER SEQUENCE auction_id_seq OWNED BY auction.aid
```

2.2.2 Description

```
cs421=> \d auction
          Type      Table "cs421g16.auction"
 Column |          Modifiers
-----+-----+
 aid   | integer      not null default nextval('auction_id_seq'::regclass)
 initialprice | money
 starttime | time without time zone default ('now'::text)::time with time zone
 date   | date        default ('now'::text)::date
 status  | character varying(50)
 email   | character varying(200)
 itemid  | integer
Indexes:
 "auction_pkey" PRIMARY KEY, btree (aid)
 "unique_itemid" UNIQUE CONSTRAINT, btree (itemid)
Foreign-key constraints:
 "email_foreign" FOREIGN KEY (email) REFERENCES users(email)
 "itemid_foreign" FOREIGN KEY (itemid) REFERENCES item(itemid)
Referenced by:
 TABLE "bid" CONSTRAINT "aid_foreign" FOREIGN KEY (aid) REFERENCES auction(aid)
```

2.2.3 Insertion Command

```
insert into auction values(3,260,'12:00pm','02-28-2016','Active','aliaamoharram@hotmail.com',4);
```

2.2.4 Populated Table

```
[cs421=> select * from auction;
      aid | initialprice |      starttime    |      date      | status |           email            |      itemid
-----+-----+-----+-----+-----+-----+-----+
      3 |     $260.00 | 12:00:00       | 2016-02-28 | Active | aliaamoharram@hotmail.com |      4
      4 |   $15,250.00 | 10:00:00       | 2016-02-28 | Active | lucien.george@mail.mcgill.ca |      3
      2 |      $90.00 | 11:30:00       | 2015-12-20 | Active | ossama.samir.ahmed@gmail.com |      2
      5 |     $280.00 | 14:27:21.554515 | 2015-09-28 | Active | zeyadnazemsaleh@gmail.com |      5
      6 |     $280.00 | 14:30:17.520242 | 2015-07-28 | Active | zeyadnazemsaleh@gmail.com |      6
      1 |   $1,148.00 | 09:00:00       | 2016-02-28 | Active | zeyadnazemsaleh@gmail.com |      1
(6 rows)
```

2.3 BID

2.3.1 Creation Command

```
CREATE SEQUENCE bid_id_seq MINVALUE 1;
```

```
create table bid(bidid integer primary key,price money,status varchar(50),time time SET DEFAULT current_time,date date SET DEFAULT current_date,email varchar(200),aid integer,foreign key (aid) references auction(aid),foreignkey (email) references users(email));
```

```
ALTER TABLE bid ALTER bidid SET DEFAULT nextval('bid_id_seq')
```

```
ALTER SEQUENCE bid_id_seq OWNED BY bid.bidid
```

2.3.2 Description

Table "cs421g16.bid"		
Column	Type	Modifiers
bidid	integer	not null default nextval('bid_id_seq'::regclass)
price	money	
status	character varying(50)	
time	time without time zone	default ('now'::text)::time with time zone
date	date	default ('now'::text)::date
email	character varying(200)	
aid	integer	
Indexes:		
"bids_pkey" PRIMARY KEY, btree (bidid)		
Foreign-key constraints:		
"aid_foreign" FOREIGN KEY (aid) REFERENCES auction(aid)		
"foreign_email" FOREIGN KEY (email) REFERENCES users(email)		

2.3.3 Insertion Command

```
insert into bid values(7,120,'Active',current_time,current_date,'aliaamoharram@hotmail.com',2);  
insert into bid values(8,130,'Active',current_time,current_date,'josephakl95@gmail.com',2);
```

2.3.4 Populated Table

Table "bid"								
bidid	price	status	time	date	date	email	aid	
2	\$280.00	Active	12:03:00	2016-02-23	zeyadnazemsa@...com	zeyadnazemsa@...com	3	
3	\$300.00	Active	12:06:00	2016-02-23	lucien.george@mail.mcgill.ca	lucien.george@mail.mcgill.ca	3	
4	\$15,500.00	Approved	10:11:00	2016-02-23	zeyadnazemsa@...com	zeyadnazemsa@...com	4	
10	\$8,000.00	Active	09:29:15.711991	2016-02-29	aliaamoharram@hotmail.com	aliaamoharram@hotmail.com	6	
11	\$11,000.00	Active	09:29:30.59055	2016-02-29	aliaamoharram@hotmail.com	aliaamoharram@hotmail.com	7	
1	\$100.00	Approved	11:32:00	2016-02-23	lucien.george@mail.mcgill.ca	lucien.george@mail.mcgill.ca	2	
9	\$1,250.00	Approved	08:42:54.160533	2016-02-29	aliaamoharram@hotmail.com	aliaamoharram@hotmail.com	1	
7	\$120.00	Cancelled	08:36:23.87167	2016-02-29	aliaamoharram@hotmail.com	aliaamoharram@hotmail.com	2	
8	\$130.00	Cancelled	08:36:40.402029	2016-02-29	josephakl95@gmail.com	josephakl95@gmail.com	2	
6	\$1,200.00	Cancelled	09:02:00	2016-02-23	osssama.samir.ahmed@gmail.com	osssama.samir.ahmed@gmail.com	1	
5	\$15,250.00	Cancelled	10:04:00	2016-02-23	aliaamoharram@hotmail.com	aliaamoharram@hotmail.com	4	

2.4 ITEM

2.4.1 Creation Command

```
CREATE SEQUENCE item_id_seq MINVALUE 1;
```

```
create table item (itemid integer primary key,category varchar(100),description  
varchar(300),deliveryoption varchar(300),picture bytea,name varchar(200));
```

```
ALTER TABLE item ALTER itemid SET DEFAULT nextval('item_id_seq')
```

```
ALTER SEQUENCE item_id_seq OWNED BY item.itemid
```

2.4.2 Description

Table "cs421g16.item"			
Column	Type	Modifiers	
itemid	integer	not null default nextval('item_id_seq'::regclass)	
category	character varying(100)		
description	character varying(300)		
deliveryoption	character varying(300)		
picture	bytea		
name	character varying(200)		
Indexes:			
"items_pkey" PRIMARY KEY, btree (itemid)			
Referenced by:			
TABLE "books" CONSTRAINT "books_book_id_fkey" FOREIGN KEY (item_id) REFERENCES item(itemid)			
TABLE "clothing" CONSTRAINT "clothing_clothing_id_fkey" FOREIGN KEY (item_id) REFERENCES item(itemid)			
TABLE "auction" CONSTRAINT "itemid_foreign" FOREIGN KEY (itemid) REFERENCES item(itemid)			

2.4.3 Insertion Command

```
insert into item values (3,'car','Audi A3 2002 with 56000km on the odometer','pick-up',null,'Audi A3');
```

2.4.4 Populated Table

itemid	category	description	deliveryoption	picture	name
1	Furniture	I -shaped couch.Very generous seating.Max width: 280cm, Min depth: 95cm,Max depth: 168cm	pick-up		L-shaped couch
2	Furniture	rectangle dining table.Length: 110cm, Width: 74cm	pick-up or delivery		dining table
3	Car	Audi A3 2002 with 56000km on the odometer	pick-up		Audi A3
4	Sports	Soccer ball from world cup 2014	delivery		Soccer ball
5	Ticket	Kygo concert. Date: 03-04-2016,Time: 9:00pm. Standing	pick-up		Kygo ticket
6	Book	Software Engineering Textbook	pick-up		Software Engineering
7	Clothing	Eshtiret neck	delivery		T-shirt
9	Electronics	Old Macbook pro 2005 with 52000km on the odometer	pick-up		Old Macbook
8	Car	Toyota Corolla 2008 with 100000km on the odometer	pick-up		Toyota Corolla
16	Electronics	I just bought a new MacBook pro, so I'm selling my old one.	pick-up or delivery		MacBook Pro 13 inch (2013)
17	Electronics	Selling my old ipod touch, 8gb. I lost the charger but it uses the same kind as the old iphones. Perfect for the gym!	pick-up or delivery		Ipod Touch 8GB
18	Music	Any drummers on here? I'm moving away and selling my baby! Professional level stuff in excellent condition.	pick-up		Drum stuff
19	Furniture	Tear along seam, easily fixed by tossing blanket on couch. Doesn't take up much space, easily folds out into bed.	pick-up		sofa bed
20	Furniture	Hi ! I have a good condition IKEA Magiken coffee table for sale. This model is rare and no longer offered by IKEA.	pick-up or delivery		IKEA coffee table
21	Furniture	Ikea Koll queen size mattress. Used for only one year. Original price is \$300	pick-up or delivery		good quality queen size mattress
22	Furniture	\$15 for both the sofa and chair. \$25 for each chair. Old	pick-up or delivery		sofa and computer chair
23	Furniture	Comfy living room couch perfect for a small room, approximately 130cm. Available for pick up	delivery		Comfy couch
24	Furniture	dresser with 3 drawers	pick-up		dresser
25	Furniture	Lovely full length mirror, quite heavy and will require car pick	pick-up		Mirror
26	Furniture	pick up in park ex	pick-up		round table with 4 chairs
27	Household	Selling it because I got an electric kettle!	pick-up or delivery		Stainless stovetop kettle
28	Electronics	I just bought a new MacBook pro, so I'm selling my old one.	pick-up or delivery		MacBook Pro 13 inch (2014)
14	Sports	Never worn, too tight on my hands. I have size 14 feet, so I'm sure they will fit someone with size 12/13 feet and under	pick-up		XL biking/sporting gloves

2.5 APPROVES (RELATION)

2.5.1 Creation Command

```
create table approves(date date,time time,seller_email varchar(200),buyer_email varchar(200),bidid integer,foreign key(seller_email) references users(email),foreign key (buyer_email references users(email));
```

2.5.2 Description

```
[cs421=> \d approves
      Table "cs421g16.approves"
    Column |          Type          | Modifiers
-----+-----+-----+
  date   |        date           |
  time   | time without time zone |
seller_email | character varying(200) |
buyer_email  | character varying(200) |
  bidid  | integer             |
Check constraints:
  "seller_buyer" CHECK (seller_email::text <> buyer_email::text)
Foreign-key constraints:
  "approves_seller_email_fkey" FOREIGN KEY (seller_email) REFERENCES users(email)
  "buyer_foreign" FOREIGN KEY (buyer_email) REFERENCES users(email)
```

2.5.3 Insertion Command

Since This table is a relation, the insertion into the table is going to be a command linking the seller to the buyer on a certain bid, in a certain date and time. The population of this relation will be discussed later in the report.

2.5.4 Populated Table

```
[cs421=> select * from approves;
  date | time | seller_email | buyer_email | bidid
-----+-----+-----+-----+-----+
(0 rows)
```

2.6 REVIEWS

2.6.1 Creation Command

```
create table reviews (reviewer_email varchar(200),reviewee_email varchar(200),description varchar(300),rating integer);
```

2.6.2 Description

```
[cs421=> \d reviews
      Table "cs421g16.reviews"
 Column | Type | Modifiers
-----+-----+
 reviewer_email | character varying(200) |
 reviewee_email | character varying(200) |
 description | character varying(300) |
 rating | integer |
Check constraints:
 "different" CHECK (reviewer_email::text <> reviewee_email::text)
 "reviews_rating_check" CHECK (rating > 0 AND rating < 6)
```

2.6.3 Insertion Command

```
insert into reviews values('lucien.george@mail.mcgill.ca','ossama.samir.ahmed@gmail.com','Ossama was honest about the condition of the item he was selling and the transaction was pretty simple. He was a little bit late though to our meeting.',4);
```

2.6.4 Populated Table

```
[cs421=> select * from reviews;
   reviewer_email | reviewee_email | description | rating
-----+-----+-----+-----+
lucien.george@mail.mcgill.ca | ossama.samir.ahmed@gmail.com | Ossama was honest about the condition of the item he was selling and the transaction was pretty simple. He was a little bit late though to our meeting. | 4
allamoharrame@hotmail.com | zeyadazemsahe@gmail.com | Zeyad described perfectly the condition of the item and made the transaction process really easy | 5
(2 rows)
```

2.7 CLOTHING

2.7.1 Command

```
create table clothing(item_id integer primary key,size varchar(20),condition varchar(200),brand varchar(200),color varchar(200),foreign key(item_id) references item(itemid));
```

2.7.2 Description

```
[cs421=> \d clothing;
                                         Table "cs421g16.clothing"
   Column      |      Type      |                         Modifiers
-----+-----+-----+
item_id    | integer      | not null default nextval('item_id_seq'::regclass)
size       | character varying(20) |
condition  | character varying(200) |
brand      | character varying(200) |
color      | character varying(200) |
Indexes:
    "clothing_pkey" PRIMARY KEY, btree (item_id)
Foreign-key constraints:
    "clothing_coloring_id_fkey" FOREIGN KEY (item_id) REFERENCES item(itemid)
```

2.7.3 Insertion Command

```
insert into clothing values (7,'M','Brand new','Abercrombie & Fitch','Grey');
```

2.7.4 Populated Table

```
[cs421=> select * from clothing;
   item_id | size | condition |         brand         | color
-----+-----+-----+-----+-----+
        7 | M   | Brand new | Abercrombie & Fitch | Grey
(1 row)
```

2.8 BOOKS

2.8.1 Command

```
create table books(item_id integer primary key,edition integer,author varchar(100),title  
varchar(200),literature varchar(200),foreign key (item_id) references item(itemid));
```

2.8.2 Description

```
|cs421=> \d books;  
                                         Table "cs421g16.books"  
   Column      |      Type      |                         Modifiers  
-----+-----+-----+  
item_id | integer | not null default nextval('item_id_seq'::regclass)  
edition | integer |  
author  | character varying(100) |  
title   | character varying(200) |  
literature | character varying(200) |  
Indexes:  
  "books_pkey" PRIMARY KEY, btree (item_id)  
Foreign-key constraints:  
  "books_book_id_fkey" FOREIGN KEY (item_id) REFERENCES item(itemid)
```

2.8.3 Insertion Command

```
insert into books values(6,10,'Ian Sommerville','Software Engineering','Engineering','Software engineering  
book for ECSE 321 class bearly opened it still in great condition');
```

2.8.4 Populated Table

```
|cs421=> select * from books;  
 item_id | edition |    author     |      title      | literature  
-----+-----+-----+-----+-----+  
       6 |      10 | Ian Sommerville | Software Engineering | Engineering  
(1 row)
```

3 TASK V: QUERIES

For this task we have discussed some of the most important queries that can be implemented in this project. The following is a list of all the suggested queries from the least to the most interesting, followed by the implementation and a brief discussion of the 5 most crucial ones.

- get items that are available for auctioning from the database
- get the name of the seller who initiated an auction
- get the auctions initiated by a specific seller
- the most active time for bidding on clothing.
- get the period that most students buy textbooks in.
- for a given user the type of item selling/buying most.
- get the average bid price done by a specific user on a specific category.
- get the history of item bought by a specific user.
- get the best bid price on all the user's items that they are selling along with the item's information
- get bids on a certain item if you are the seller of this item
- get the bids offered by a specific buyer in any auction
- get the average bid price on a certain auction
- Return bids on a certain auction ordered by highest to lowest price
- get the history of item sold by a specific user.

We have decided to implement the following queries:

3.1 GET THE HISTORY OF ITEMS SOLD BY A SPECIFIC USER

3.1.1 Discussion

This query is done over joining 3 tables (users, auction and item). Whereas the same query can be done over joining only the auction and item tables(because the seller's email is included in auction), the joining of the 3 tables allows for a wider scope where you can search for the history of items sold by a specific user using his first name, last name or any other attributes in user.

3.1.2 Command

```
select name, itemid, email from
item natural join (select itemid, email from users natural join auction) as temp
where email = 'zeyadzhazemsaleh@gmail.com';
```

3.1.3 Breakdown and results

```
select * from users natural join auction
```

```
cs421=> (select * from users natural join auction);
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| name | itemid | email | fname | lname | dateofbirth | gender | university | address | rating | picture | phonenumer | aid | initialprice | starttime | date | status | itemid |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| aliaamoharram@hotmail.com | 1 | Aliaa | Moharram | 1995-01-08 | Female | McGill University | 3410 Peel | 5 | 1 | +1(514) 243-6927 | 3 | $260.00 | 12:00:00 | 2016-02-28 | Active | 4
| lucien.george@mail.mcgill.ca | 2 | Lucien | George | 1995-09-16 | Male | McGill University | 3450 Drummond | 5 | 1 | +1(514) 430-8272 | 4 | $15,250.00 | 10:00:00 | 2016-02-28 | Active | 3
| ossama.samir.ahmed@gmail.com | 3 | Ossama | Samir | Ahmed | 1992-10-21 | Male | McGill University | Samuel Home | 1 | 1 | +1(514) 836-2217 | 2 | $90.00 | 11:30:00 | 2015-12-20 | Active | 2
| zeyadzhazemsaleh@gmail.com | 4 | Zeyad | Saleh | 1994-05-07 | Male | McGill University | Zeyad Home | 5 | 1 | +1(514) 585-6769 | 1 | $1,148.00 | 09:00:00 | 2016-02-28 | Active | 1
| zeyadzhazemsaleh@gmail.com | 5 | Zeyad | Saleh | 1994-05-07 | Male | McGill University | Zeyad Home | 5 | 1 | +1(514) 585-6769 | 5 | $280.00 | 14:27:21.554515 | 2015-09-28 | Active | 5
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
(5 rows)
```

```
select itemid, email from users natural join auction;
```

```
[cs421=> (select itemid, email from users natural join auction);
+-----+-----+
| itemid | email
+-----+-----+
| 4 | aliaamoharram@hotmail.com
| 3 | lucien.george@mail.mcgill.ca
| 2 | ossama.samir.ahmed@gmail.com
| 1 | zeyadzhazemsaleh@gmail.com
| 5 | zeyadzhazemsaleh@gmail.com
+-----+
(5 rows)
```

```
select name, itemid, email from
```

```
item natural join (select itemid, email from users natural join auction) as temp;
```

```
cs421=> select name, itemid, email from
item natural join
(select itemid, email from users natural join auction) as temp;
+-----+-----+-----+
| name | itemid | email
+-----+-----+-----+
| Soccer ball | 4 | aliaamoharram@hotmail.com
| Audi A3 | 3 | lucien.george@mail.mcgill.ca
| dining table | 2 | ossama.samir.ahmed@gmail.com
| Kygo ticket | 5 | zeyadzhazemsaleh@gmail.com
| L-shaped couch | 1 | zeyadzhazemsaleh@gmail.com
+-----+
(5 rows)
```

```
select name, itemid, email from
```

```
item natural join (select itemid, email from users natural join auction) as temp
```

```
where email = 'zeyadzhazemsaleh@gmail.com';
```

```
cs421=> select name, itemid, email from
item natural join
(select itemid, email from users natural join auction) as temp
where email = 'zeyadzhazemsaleh@gmail.com';
+-----+-----+-----+
| name | itemid | email
+-----+-----+-----+
| Kygo ticket | 5 | zeyadzhazemsaleh@gmail.com
| L-shaped couch | 1 | zeyadzhazemsaleh@gmail.com
+-----+
(2 rows)
```

3.2 SELECT THE CATEGORIES THAT A BUYER BIDS ON THE MOST

3.2.1 Discussion

In most of the online trading applications, an interface is dedicated to suggesting products to the user based on their previous purchases or their search history. This query extracts a list of categories of items that a user has submitted bids to, to create such a customized user interface.

3.2.2 Command

```
select item.category from
(select bid.aid, bid.email, auction.itemid from bid inner join auction on bid.aid = auction.aid)x
inner join item on x.itemid = item.itemid
where email = 'aliaamoharram@hotmail.com' group by item.category order by count(*) desc;
```

3.2.3 Breakdown and results

```
cs421=> select * from bid inner join auction on bid.aid = auction.aid;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| bidid | price | status | time | date | email | aid | bid | initialprice | starttime | date | status | email | itemid |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
6 | $1,200.00 | Active | 09:02:00 | 2016-02-23 | ossama.samir.ahmed@gmail.com | 1 | 1 | $1,148.00 | 09:00:00 | 2016-02-28 | Active | zeyadnazemsoleh@gmail.com | 1
2 | $280.00 | Active | 12:03:00 | 2016-02-23 | zeyadnazemsoleh@gmail.com | 1 | 3 | $260.00 | 12:00:00 | 2016-02-28 | Active | aliaamoharram@hotmail.com | 4
3 | $300.00 | Active | 12:06:00 | 2016-02-23 | lucien.george@mail.mcgill.ca | 3 | 3 | $260.00 | 12:00:00 | 2016-02-28 | Active | aliaamoharram@hotmail.com | 4
1 | $100.00 | Active | 11:32:00 | 2016-02-23 | lucien.george@mail.mcgill.ca | 2 | 2 | $90.00 | 11:30:00 | 2016-12-28 | Active | ossama.samir.ahmed@gmail.com | 2
4 | $15,250.00 | Expired | 2016-02-23 | aliaamoharram@hotmail.com | 4 | 4 | $15,250.00 | 10:00:00 | 2016-02-28 | Active | ossama.samir.ahmed@gmail.com | 3
5 | $1,250.00 | Cancelled | 10:04:00 | 2016-02-23 | aliaamoharram@hotmail.com | 4 | 4 | $1,250.00 | 10:00:00 | 2016-02-28 | Active | lucien.george@mail.mcgill.ca | 3
7 | $120.00 | Active | 08:36:23.87167 | 2016-02-29 | aliaamoharram@hotmail.com | 2 | 2 | $90.00 | 11:30:00 | 2015-12-20 | Active | ossama.samir.ahmed@gmail.com | 2
8 | $130.00 | Active | 08:36:40.402029 | 2016-02-29 | josephak195@gmail.com | 2 | 2 | $90.00 | 11:30:00 | 2015-12-20 | Active | ossama.samir.ahmed@gmail.com | 2
9 | $1,250.00 | Active | 08:42:54.160823 | 2016-02-29 | aliaamoharram@hotmail.com | 1 | 1 | $1,148.00 | 09:00:00 | 2016-02-28 | Active | zeyadnazemsoleh@gmail.com | 1
10 | $8,000.00 | Active | 09:29:15.711991 | 2016-02-29 | aliaamoharram@hotmail.com | 6 | 6 | $7,000.00 | 12:00:00 | 2016-02-28 | Active | lucien.george@mail.mcgill.ca | 8
11 | $11,000.00 | Active | 09:29:38.59655 | 2016-02-29 | aliaamoharram@hotmail.com | 7 | 7 | $10,000.00 | 12:00:00 | 2016-02-28 | Active | lucien.george@mail.mcgill.ca | 9
(11 rows)

cs421=> select bid.aid, bid.email, auction.itemid from bid inner join auction on bid.aid = auction.aid;
+-----+-----+-----+
| aid | email | itemid |
+-----+-----+-----+
1 | ossama.samir.ahmed@gmail.com | 1
3 | zeyadnazemsoleh@gmail.com | 4
2 | lucien.george@mail.mcgill.ca | 2
4 | zeyadnazemsoleh@gmail.com | 3
4 | aliaamoharram@hotmail.com | 3
2 | aliaamoharram@hotmail.com | 2
2 | josephak195@gmail.com | 2
1 | aliaamoharram@hotmail.com | 1
6 | aliaamoharram@hotmail.com | 8
7 | aliaamoharram@hotmail.com | 9
(11 rows)

cs421=> select * from
cs421-> (select bid.aid, bid.email, auction.itemid from bid inner join auction on bid.aid = auction.aid)x
cs421-> inner join item on x.itemid = item.itemid;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| aid | email | itemid | itemid | itemid | category | description | price | deliveryoption | picture | name |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 | aliaamoharram@hotmail.com | 1 | 1 | 1 | Furniture | L-shaped couch.Very generous seating.Max width: 280cm, Min depth: 95cm,Max depth: 168cm | $1,148.00 | I pick-up | | L-shaped couch
1 | ossama.samir.ahmed@gmail.com | 1 | 1 | 1 | Furniture | L-shaped couch.Very generous seating.Max width: 280cm, Min depth: 95cm,Max depth: 168cm | $1,148.00 | I pick-up | | L-shaped couch
2 | josephak195@gmail.com | 2 | 1 | 2 | Furniture | rectangular dining table.Length: 110cm, Width: 74cm | $84.00 | I pick-up or delivery | | dining table
2 | aliaamoharram@hotmail.com | 2 | 2 | 2 | Furniture | rectangular dining table.Length: 110cm, Width: 74cm | $84.00 | I pick-up or delivery | | dining table
2 | lucien.george@mail.mcgill.ca | 2 | 2 | 2 | Furniture | rectangular dining table.Length: 110cm, Width: 74cm | $84.00 | I pick-up or delivery | | dining table
4 | zeyadnazemsoleh@gmail.com | 3 | 3 | 1 | Car | Audi A3 2002 with 56000km on the odometer | $15,000.00 | I pick-up | | Audi A3
4 | aliaamoharram@hotmail.com | 3 | 3 | 1 | Car | Audi A3 2002 with 56000km on the odometer | $15,000.00 | I pick-up | | Audi A3
4 | zeyadnazemsoleh@gmail.com | 3 | 3 | 1 | Car | Soccer ball from world cup 2014 | $250.00 | I delivery | | Soccer ball
3 | lucien.george@mail.mcgill.ca | 4 | 4 | 1 | Sports | Soccer ball from world cup 2014 | $250.00 | I delivery | | Soccer ball
3 | zeyadnazemsoleh@gmail.com | 4 | 4 | 1 | Sports | Chevy Mustang 2005 with 830000km on the odometer | $10,000.00 | I pick-up | | Chevrolet Mustang
6 | aliaamoharram@hotmail.com | 9 | 9 | 1 | Car | Toyota Corolla 2008 with 100000km on the odometer | $7,000.00 | I pick-up | | Toyota Corolla
6 | aliaamoharram@hotmail.com | 8 | 8 | 1 | Car | Toyota Corolla 2008 with 100000km on the odometer | $7,000.00 | I pick-up | | Toyota Corolla
(11 rows)

cs421=> select x.email, item.category from
cs421-> (select bid.aid, bid.email, auction.itemid from bid inner join auction on bid.aid = auction.aid)x
cs421-> inner join item on x.itemid = item.itemid;
+-----+-----+
| email | category |
+-----+-----+
aliaamoharram@hotmail.com | Furniture
ossama.samir.ahmed@gmail.com | Furniture
josephak195@gmail.com | Furniture
aliaamoharram@hotmail.com | Furniture
lucien.george@mail.mcgill.ca | Furniture
aliaamoharram@hotmail.com | Car
zeyadnazemsoleh@gmail.com | Car
lucien.george@mail.mcgill.ca | Sports
zeyadnazemsoleh@gmail.com | Sports
aliaamoharram@hotmail.com | Car
aliaamoharram@hotmail.com | Car
(11 rows)
```

```

cs421=> select x.email, item.category from
  (select bid.aid, bid.email, auction.itemid from bid inner join auction on bid.aid = auction.aid)x
 [inner join item on x.itemid = item.itemid where email = 'aliaamoharram@hotmail.com';
      email          | category
-----+-----
aliaamoharram@hotmail.com | Furniture
aliaamoharram@hotmail.com | Furniture
aliaamoharram@hotmail.com | Car
aliaamoharram@hotmail.com | Car
aliaamoharram@hotmail.com | Car
(5 rows)

cs421=> select item.category, count(*) from
  (select bid.aid, bid.email, auction.itemid from bid inner join auction on bid.aid = auction.aid)x
 [cs421-> inner join item on x.itemid = item.itemid
[cs421-> where email = 'aliaamoharram@hotmail.com' group by item.category;
      category | count
-----+-----
Furniture |    2
Car        |    3
(2 rows)

cs421=> select item.category from
  (select bid.aid, bid.email, auction.itemid from bid inner join auction on bid.aid = auction.aid)x
 [cs421-> inner join item on x.itemid = item.itemid
[cs421-> where email = 'aliaamoharram@hotmail.com' group by item.category order by count(*) desc;
      category
-----
Car
Furniture
(2 rows)

```

3.3 GET ALL BIDS IN A CERTAIN AUCTION

3.3.1 Discussion

This is one of the most important queries as it connects 2 tables auction and bids

3.3.2 Command

```
select * from (select email, aid from auction where aid = 2) as temp1 inner join bid on bid.aid = temp1.aid;
```

3.3.3 Breakdown and results

Select email, aid from auction where aid=4; This command returns the required auction

```
[cs421=> select email, aid from users natural join (select * from auction where aid = 4) as temp1;
      email          | aid
-----+-----
  lucien.george@mail.mcgill.ca | 4
(1 row)
```

```
select * from (select email, aid from auction where aid = 4) as temp1 inner join bid on bid.aid = temp1.aid
```

```
[cs421=> select * from (select email, aid from auction where aid = 4) as temp1 inner join bid on bid.aid = temp1.aid;
      email          | aid | bidid | price   | status    | time     | date     | email          | aid
-----+-----+-----+-----+-----+-----+-----+-----+
  lucien.george@mail.mcgill.ca | 4 | 5 | $15,250.00 | Canceled | 10:04:00 | 2016-02-23 | aliaamoharram@hotmail.com | 4
  lucien.george@mail.mcgill.ca | 4 | 4 | $15,500.00 | Approved  | 10:11:00 | 2016-02-23 | zeyadnazemsaleh@gmail.com | 4
(2 rows)
```

3.4 GET THE AVERAGE BID PRICE ON A CERTAIN AUCTION

3.4.1 Discussion

This query is interesting in the sense where it uses the grouping and aggregate functions to return useful statistical information in a certain auction

3.4.2 Command

```
select aid, avg(price::numeric), max(price) from bid  
group by aid having aid = 3;
```

3.4.3 Breakdown and results

Select * from bid;

```
[cs421=> select * from bid;  
+-----+-----+-----+-----+-----+-----+-----+  
| bidid | price | status | time | date | email | aid |  
+-----+-----+-----+-----+-----+-----+-----+  
| 6 | $1,200.00 | Active | 09:02:00 | 2016-02-23 | ossama.samir.ahmed@gmail.com | 1 |  
| 2 | $280.00 | Active | 12:03:00 | 2016-02-23 | zeyadnazemsaleh@gmail.com | 3 |  
| 3 | $300.00 | Active | 12:06:00 | 2016-02-23 | aliaamoharram@hotmail.com | 3 |  
| 5 | $15,250.00 | Canceled | 10:04:00 | 2016-02-23 | aliaamoharram@hotmail.com | 4 |  
| 1 | $100.00 | Active | 11:32:00 | 2016-02-23 | ossama.samir.ahmed@gmail.com | 2 |  
| 4 | $15,500.00 | Approved | 10:11:00 | 2016-02-23 | zeyadnazemsaleh@gmail.com | 4 |  
(6 rows)
```

select aid, avg(price::numeric), max(price) from bid group by aid;

```
[cs421=> select aid, avg(price::numeric), max(price) from bid  
[cs421-> group by aid;  
+-----+-----+  
| aid | avg | max |  
+-----+-----+  
| 4 | 15375.0000000000000000 | $15,500.00 |  
| 1 | 1200.0000000000000000 | $1,200.00 |  
| 2 | 100.0000000000000000 | $100.00 |  
| 3 | 290.0000000000000000 | $300.00 |  
(4 rows)
```

select aid, avg(price::numeric), max(price) from bid
group by aid having aid = 3;

```
cs421=> select aid, avg(price::numeric), max(price) from bid  
[group by aid having aid = 3;  
+-----+-----+  
| aid | avg | max |  
+-----+-----+  
| 3 | 290.0000000000000000 | $300.00 |  
(1 row)
```

3.5 GET ALL BIDS OFFERED BY A CERTAIN BUYER IN ALL AUCTIONS:

3.5.1 Discussion

This is an easy but essential query, that allows a user to see all the bids he is offering in all auctions

3.5.2 Command

```
select * from bid  
where bid.email = 'aliaamoharram@hotmail.com';
```

3.5.3 Breakdown and results

Select * from bid;

```
[cs421=> select * from bid;  
+-----+-----+-----+-----+-----+-----+-----+  
| bidid | price | status | time | date | email | aid |  
+-----+-----+-----+-----+-----+-----+-----+  
| 6 | $1,200.00 | Active | 09:02:00 | 2016-02-23 | ossama.samir.ahmed@gmail.com | 1 |  
| 2 | $280.00 | Active | 12:03:00 | 2016-02-23 | zeyadnazemsaeh@gmail.com | 3 |  
| 3 | $300.00 | Active | 12:06:00 | 2016-02-23 | aliaamoharram@hotmail.com | 3 |  
| 5 | $15,250.00 | Canceled | 10:04:00 | 2016-02-23 | aliaamoharram@hotmail.com | 4 |  
| 1 | $100.00 | Active | 11:32:00 | 2016-02-23 | ossama.samir.ahmed@gmail.com | 2 |  
| 4 | $15,500.00 | Approved | 10:11:00 | 2016-02-23 | zeyadnazemsaeh@gmail.com | 4 |  
(6 rows)
```

```
select * from bid where bid.email = 'aliaamoharram@hotmail.com';
```

```
cs421=> select * from bid  
cs421-> where bid.email = 'aliaamoharram@hotmail.com';  
+-----+-----+-----+-----+-----+-----+-----+  
| bidid | price | status | time | date | email | aid |  
+-----+-----+-----+-----+-----+-----+-----+  
| 3 | $300.00 | Active | 12:06:00 | 2016-02-23 | aliaamoharram@hotmail.com | 3 |  
| 5 | $15,250.00 | Canceled | 10:04:00 | 2016-02-23 | aliaamoharram@hotmail.com | 4 |  
(2 rows)
```

3.6 RETURN BIDS ON ALL AUCTIONS ORDERED BY DESCENDING PRICE AND ASCENDING AID

3.6.1 Discussion

This is an important query as it allows for the sellers to see how people are valuing their items

3.6.2 Command

```
select * from bid order by aid, price desc;
```

3.6.3 Breakdown and results

Select * from bid;

```
[cs421=> select * from bid;
+-----+
| bidid | price      | status    | time       | date       | email           | aid |
+-----+
| 6     | $1,200.00  | Active   | 09:02:00  | 2016-02-23 | ossama.samir.ahmed@gmail.com | 1  |
| 2     | $280.00    | Active   | 12:03:00  | 2016-02-23 | zeyadnazemsaleh@gmail.com   | 3  |
| 3     | $300.00    | Active   | 12:06:00  | 2016-02-23 | aliaamoharram@hotmail.com | 3  |
| 5     | $15,250.00 | Canceled | 10:04:00  | 2016-02-23 | aliaamoharram@hotmail.com | 4  |
| 1     | $100.00    | Active   | 11:32:00  | 2016-02-23 | ossama.samir.ahmed@gmail.com | 2  |
| 4     | $15,500.00 | Approved  | 10:11:00  | 2016-02-23 | zeyadnazemsaleh@gmail.com   | 4  |
+-----+
(6 rows)
```

select * from bid order by aid, price desc;

```
cs421=> select * from bid order by aid, price desc;
+-----+
| bidid | price      | status    | time       | date       | email           | aid |
+-----+
| 6     | $1,200.00  | Active   | 09:02:00  | 2016-02-23 | ossama.samir.ahmed@gmail.com | 1  |
| 1     | $100.00    | Active   | 11:32:00  | 2016-02-23 | ossama.samir.ahmed@gmail.com | 2  |
| 3     | $300.00    | Active   | 12:06:00  | 2016-02-23 | aliaamoharram@hotmail.com | 3  |
| 2     | $280.00    | Active   | 12:03:00  | 2016-02-23 | zeyadnazemsaleh@gmail.com   | 3  |
| 4     | $15,500.00 | Approved  | 10:11:00  | 2016-02-23 | zeyadnazemsaleh@gmail.com   | 4  |
| 5     | $15,250.00 | Canceled | 10:04:00  | 2016-02-23 | aliaamoharram@hotmail.com | 4  |
+-----+
(6 rows)
```

Notice that the column aid is sorted in ascending order and price in descending order for each aid.

3.7 GET THE AVERAGE BID DONE BY A SPECIFIC USER ON A CERTAIN CATEGORY

3.7.1 Discussion

This is an important query as it allows the system to recommend to the user specific items that they are probably going to bid on; given that the item's price is within the average of the user's history of bidding on this item's category.

3.7.2 Command

```
select avg(temp2.price::numeric::float8) from (select * from (select aid,price from bid where email='aliaamoharram@hotmail.com') as temp1 join auction on temp1.aid=Auction.aid) as temp2 join item on temp2.itemid = item.itemid and item.category='Furniture';
```

3.7.3 Breakdown and results

The following is a breakdown of the results.

price	email	category
\$1,250.00	aliaamoharram@hotmail.com	Furniture
\$1,200.00	osama.samir.ahmed@gmail.com	Furniture
\$130.00	josephak195@gmail.com	Furniture
\$120.00	aliaamoharram@hotmail.com	Furniture
\$100.00	lucien.george@mail.mcgill.ca	Furniture
\$280.00	zeyadhamzaeh@gmail.com	Sports
\$300.00	lucien.george@mail.mcgill.ca	Sports
\$15,250.00	aliaamoharram@hotmail.com	Car
\$15,500.00	zeyadhamzaeh@gmail.com	Car
\$8,000.00	aliaamoharram@hotmail.com	Car
\$11,000.00	aliaamoharram@hotmail.com	Car

Result of the query:

```
cs421=> select avg(temp2.price::numeric::float8) from (select * from (select aid,price from bid where email='aliaamoharram@hotmail.com') as temp1 join auction on temp1.aid=Auction.aid) as temp2 join item on temp2.itemid = item.itemid and item.category='Furniture';
avg
_____
685
(1 row)
```

3.8 GET THE BEST BID PRICE ON ALL THE USER'S ITEMS THAT THEY ARE SELLING ALONG WITH THE ITEM'S INFORMATION

3.8.1 Discussion

This is an important query as it allows the user to check all the best bid prices on all the items that they are selling along with the item's information.

3.8.2 Command

```
select * from (select aid, MAX(price)
from bid
where (aid IN (select aid from auction where email = 'zeyadzhazemsaleh@gmail.com')) GROUP BY aid ) as
temp join item on item.itemid=(select itemid from auction where auction.aid=temp.aid);
```

3.8.3 Breakdown and results

The following is a breakdown of the results.

```
cs421=> select price, temp1.aID from (select bid.aid,price from bid join auction on bid.aid=auction.aid and auction.email='zeyadzhazemsaleh@gmail.com') as temp
1
;
+-----+-----+
| price | aid |
+-----+-----+
| $1,200.00 | 1
| $1,250.00 | 1
(2 rows)
```

The result:

aid	max name	itemid	category	description	deliveryoption	picture
1	\$1,250.00 -shaped couch	1	Furniture	L-shaped couch.Very generous seating.Max width: 280cm, Min depth: 95cm,Max depth: 168cm	pick-up	L

(1 row)

3.9 GET THE HISTORY OF THE ITEMS BOUGHT BY A SPECIFIC USER

3.9.1 Discussion

This is an important query as it allows the user to check the history of the items that they bought.

3.9.2 Command

```
SELECT * from (SELECT itemid from bid join auction on bid.email='zeyadnazemsaleh@gmail.com' and auction.aid=bid.aid and bid.status='Approved') AS temp1 join item ON temp1.itemid=item.itemid
```

3.9.3 Breakdown and results

The following is a breakdown of the results.

bidid	price	status	time	date	email	aid
2	\$280.00	Active	12:03:00	2016-02-23	zeyadnazemsaleh@gmail.com	3
3	\$300.00	Active	12:06:00	2016-02-23	lucien.george@mail.mcgill.ca	3
4	\$15,500.00	Approved	10:11:00	2016-02-23	zeyadnazemsaleh@gmail.com	4
5	\$15,250.00	Cancelled	10:04:00	2016-02-23	aliamoharam@hotmail.com	4
10	\$8,000.00	Active	09:29:15.711991	2016-02-29	aliamoharam@hotmail.com	6
11	\$11,000.00	Active	09:29:30.59855	2016-02-29	aliamoharam@hotmail.com	7
7	\$120.00	Cancelled	08:36:23.87157	2016-02-29	aliamoharam@hotmail.com	2
8	\$130.00	Cancelled	08:36:49.402029	2016-02-29	josephak195@gmail.com	2
1	\$100.00	Approved	11:32:00	2016-02-23	lucien.george@mail.mcgill.ca	2
6	\$1,200.00	Cancelled	09:02:00	2016-02-23	osama.samir.ahmed@gmail.com	1
9	\$1,250.00	Approved	08:42:54.160533	2016-02-29	aliamoharam@hotmail.com	1

The result:

cs421=> SELECT * from (SELECT itemid from bid join auction on bid.email='zeyadnazemsaleh@gmail.com' and auction.aid=bid.aid and bid.status='Approved') AS temp1 join item ON temp1.itemid=item.itemid
;
itemid itemid category description deliveryoption picture name
(1 row)

4 TASK VI: INTERESTING DATA MODIFICATION COMMANDS

4.1 APPROVING A BID AND INSERTING THE BUYER AND SELLER INFO INTO APPROVES

4.1.1 Discussion

This is an interesting data modification command as it inserts values of 2 different tables into the relationship that links them. It also uses features such as storing the time and date of the approval. This query need to be followed by another query that sets the status of the bid to approved

4.1.2 Command

```
insert into approves (date, time, seller_email, buyer_email, bidid) values (current_date, current_time,
(select auction.email from bid inner join auction on bid.aid = auction.aid
where bidid = 4),
(select bid.email from bid inner join auction on bid.aid = auction.aid where bidid = 4),
(select bid.bidid from bid inner join auction on bid.aid = auction.aid where bidid = 4));
```

4.1.3 Breakdown and results

Select * from approves;

```
[cs421=> select * from approves;
  date | time | seller_email | buyer_email | bidid
-----+-----+-----+-----+
(0 rows)
```

select auction.email,bid.email,bid.bidid from bid inner join auction on bid.aid = auction.aid where bidid=4;

```
[cs421=> select auction.email, bid.email, bid.bidid from bid inner join auction on bid.aid = auction.aid where bidid = 4;
      email      |      email      | bidid
-----+-----+-----+
  lucien.george@mail.mcgill.ca | zeyadnazemsahe@gmail.com |     4
(1 row)
```

```
insert into approves (date, time, seller_email, buyer_email, bidid) values (current_date, current_time,
(select auction.email from bid inner join auction on bid.aid = auction.aid where bidid = 4),
(select bid.email from bid inner join auction on bid.aid = auction.aid where bidid = 4),
(select bid.bidid from bid inner join auction on bid.aid = auction.aid where bidid = 4));
```

```
[cs421=> insert into approves (date, time, seller_email, buyer_email, bidid) values (current_date, current_time, (select auction.email from bid inner join auction on bid.aid = auction.aid
where bidid = 4),
(select bid.email from bid inner join auction on bid.aid = auction.aid where bidid = 4),
(select bid.bidid from bid inner join auction on bid.aid = auction.aid where bidid = 4));
INSERT 0 1
```

select * from approves;

```
[cs421=> select * from approves;
      date      |      time      |      seller_email      |      buyer_email      | bidid
-----+-----+-----+-----+
  2016-02-28 | 20:49:17.031547 | lucien.george@mail.mcgill.ca | zeyadnazemsahe@gmail.com |     4
(1 row)
```

update bid SET status = 'Approved' where bidid = 4;

```
[cs421=> update bid SET status = 'Approved' where bidid = 4;
UPDATE 1
```

[cs421=> select * from bid;

```
      bidid      |      price      |      status      |      time      |      date      |      email      |      aid
-----+-----+-----+-----+-----+-----+
  6 | $1,200.00 | Active | 09:02:00 | 2016-02-23 | ossama.samir.ahmed@gmail.com | 1
  2 | $280.00 | Active | 12:03:00 | 2016-02-23 | zeyadnazemsahe@gmail.com | 3
  3 | $300.00 | Active | 12:06:00 | 2016-02-23 | aliaamoharram@hotmail.com | 3
  1 | $100.00 | Active | 11:32:00 | 2016-02-23 | ossama.samir.ahmed@gmail.com | 2
  5 | $15,250.00 | Cancelled | 10:04:00 | 2016-02-23 | aliaamoharram@hotmail.com | 4
  4 | $15,500.00 | Approved | 10:11:00 | 2016-02-23 | zeyadnazemsahe@gmail.com | 4
(6 rows)
```

4.2 DELETE AUCTIONS AFTER 6 MONTHS ELAPSING

4.2.1 Discussion

This was a very debatable operation. The reason for this is because deleting auctions after 6 months being elapsed would result in the loss of history of items sold or bought by certain users. However, we don't want our database to be full of items that have no important need anymore. Independently from the usage, we have decided to implement the query as it exhibits interesting casting features and date operations.

4.2.2 Command

```
delete from auction where current_date > date + interval '6 month';
```

4.2.3 Breakdown and results

The entry highlighted in red corresponds to an auction that has been active for a duration greater than 6 months. The following is a breakdown of the removal of that auction from the system following the constraint mentioned in the requirements.

```
[cs421=> select * from auction;
  aid | initialprice | starttime | date | status | email | itemid
-----+-----+-----+-----+-----+-----+-----+
  3 | $260.00 | 12:00:00 | 2016-02-28 | Active | aliaamoharram@hotmail.com | 4
  4 | $15,250.00 | 10:00:00 | 2016-02-28 | Active | lucien.george@mail.mcgill.ca | 3
  2 | $90.00 | 11:30:00 | 2015-12-20 | Active | ossama.samir.ahmed@gmail.com | 2
  5 | $280.00 | 14:27:21.554515 | 2015-09-28 | Active | zeyadhadzemsaleh@gmail.com | 5
  6 | $280.00 | 14:30:17.520242 | 2015-07-28 | Active | zeyadhadzemsaleh@gmail.com | 6
  1 | $1,148.00 | 09:00:00 | 2016-02-28 | Active | zeyadhadzemsaleh@gmail.com | 1
(6 rows)

[cs421=> select * from auction where current_date > date;
  aid | initialprice | starttime | date | status | email | itemid
-----+-----+-----+-----+-----+-----+-----+
  2 | $90.00 | 11:30:00 | 2015-12-20 | Active | ossama.samir.ahmed@gmail.com | 2
  5 | $280.00 | 14:27:21.554515 | 2015-09-28 | Active | zeyadhadzemsaleh@gmail.com | 5
  6 | $280.00 | 14:30:17.520242 | 2015-07-28 | Active | zeyadhadzemsaleh@gmail.com | 6
(3 rows)

[cs421=> select * from auction where current_date > date + interval '6 month';
  aid | initialprice | starttime | date | status | email | itemid
-----+-----+-----+-----+-----+-----+-----+
  6 | $280.00 | 14:30:17.520242 | 2015-07-28 | Active | zeyadhadzemsaleh@gmail.com | 6
(1 row)

[cs421=> delete from auction where current_date > date + interval '6 month';
DELETE 1

[cs421=> select * from auction;
  aid | initialprice | starttime | date | status | email | itemid
-----+-----+-----+-----+-----+-----+-----+
  3 | $260.00 | 12:00:00 | 2016-02-28 | Active | aliaamoharram@hotmail.com | 4
  4 | $15,250.00 | 10:00:00 | 2016-02-28 | Active | lucien.george@mail.mcgill.ca | 3
  2 | $90.00 | 11:30:00 | 2015-12-20 | Active | ossama.samir.ahmed@gmail.com | 2
  5 | $280.00 | 14:27:21.554515 | 2015-09-28 | Active | zeyadhadzemsaleh@gmail.com | 5
  1 | $1,148.00 | 09:00:00 | 2016-02-28 | Active | zeyadhadzemsaleh@gmail.com | 1
(5 rows)
```

4.3 MODIFY THE RATINGS FOR ALL THE USERS BASED ON THE REVIEWS TABLE

4.3.1 Discussion

Calculating the user's rating is a crucial part of our system as it affects the other users decision in accepting the bid from this specific user since the bids are anonymous in our system. Modifying and calculating the user's rating can be done through the following two options, either after a new review is inserted in the reviews table which can be resolved and implemented using a trigger function, or by running a modification query on all the users ratings. We decided to implement the second option as it shows more interesting and complex sql features.

4.3.2 Command

UPDATE users

```
SET rating = (SELECT AVG(reviews.rating) FROM reviews WHERE reviews.reviewee_email= users.email)
;
```

4.3.3 Breakdown and results

The following is a breakdown of the calculation of all the users ratings given that we modified the ratings of all users to have the value of 1 to illustrate the modification command.

email	fname	lname	dateofbirth	gender	university	address	rating	picture	phonenumber
aliaamoharram@hotmail.com	Aliaa	Moharram	1995-01-08	Female	McGill University	3410 Peel	1		+1(514) 243-6927
lucien.george@mail.mcgill.ca	Lucien	George	1995-09-16	Male	McGill University	3450 Drummond	1		+1(514) 430-8272
bettina.kemme@mcgill.ca	Bettina	Kemme	2016-01-01	Female	McGill University	Prof home	1		Not Available
zeyadnazemsaleh@gmail.com	Zeyad	Saleh	1994-05-07	Male	McGill University	Zeyad Home	1		+1(514) 585-6769
josephak95@gmail.com	Joseph	AkL	1995-04-25	Male	McGill University	3460 Peel	1		+1(514) 445 7873
osama.samir.ahmed@gmail.com	Ossama Samir	Ahmed	1992-10-21	Male	McGill University	Samuel Home	1		+1(514) 836-2217

The result:

email	fname	lname	dateofbirth	gender	university	address	rating	picture	phonenumber
aliaamoharram@hotmail.com	Aliaa	Moharram	1995-01-08	Female	McGill University	3410 Peel	1		+1(514) 243-6927
lucien.george@mail.mcgill.ca	Lucien	George	1995-09-16	Male	McGill University	3450 Drummond	5		+1(514) 430-8272
bettina.kemme@mcgill.ca	Bettina	Kemme	2016-01-01	Female	McGill University	Prof home	Not Available		
zeyadnazemsaleh@gmail.com	Zeyad	Saleh	1994-05-07	Male	McGill University	Zeyad Home	5		+1(514) 585-6769
josephak95@gmail.com	Joseph	AkL	1995-04-25	Male	McGill University	3460 Peel	5		+1(514) 445 7873
osama.samir.ahmed@gmail.com	Ossama Samir	Ahmed	1992-10-21	Male	McGill University	Samuel Home	4		+1(514) 836-2217

Given that the reviews table was the following:

reviewer_email	reviewee_email	rating	description
lucien.george@mail.mcgill.ca	osama.samir.ahmed@gmail.com	4	Ossama was honest about the condition of the item he was selling and the transaction was pretty simple. He was a little bit late though to our meeting.
aliaamoharram@hotmail.com	zeyadnazemsaleh@gmail.com	5	Zeyad described perfectly the condition of the item and made the transaction process really easy
zeyadnazemsaleh@gmail.com	lucien.george@mail.mcgill.ca	5	Zeyad was on time

4.4 CHANGE THE STATUS OF ALL BIDS ON ITEMS WHICH ALREADY HAS AN APPROVED BID

4.4.1 Discussion

After the seller approves a bid on an item they are selling, all the other bids on this item should have a status of “Cancelled”. We decided to implement a modification query to go through all the bids in the bid table and change the bid status of a bid to cancelled if there is another bid which has the same auction id and a status of “Approved”.

4.4.2 Command

UPDATE bid

set status= ‘Cancelled’

```
WHERE aid IN (select auction.aID from auction join bid on (auction.aID= bid.aID) AND (bid.status= 'Approved')) and status<>'Approved'  
;
```

4.4.3 Breakdown and results

The following is a breakdown of the modification of all the bids status. As we can see in the first picture, there are three bids that got approved.

bidid	price	status	time	date	email	aid
2	\$200.00	Active	12:03:00	2016-02-23	zeyadnazemsaleh@gmail.com	3
3	\$300.00	Active	12:06:00	2016-02-23	lucien.george@mail.mcgill.ca	3
4	\$15,500.00	Approved	10:11:00	2016-02-23	zeyadnazemsaleh@gmail.com	4
10	\$8,000.00	Active	09:29:15.711991	2016-02-29	aliamoharram@hotmail.com	6
11	\$11,000.00	Active	09:29:30.59055	2016-02-29	aliamoharram@hotmail.com	7
1	\$100.00	Approved	11:32:00	2016-02-23	lucien.george@mail.mcgill.ca	2
9	\$1,250.00	Approved	08:42:54.160533	2016-02-29	aliamoharram@hotmail.com	1
7	\$120.00	Active	08:36:23.87167	2016-02-29	aliamoharram@hotmail.com	2
8	\$130.00	Active	08:36:40.402029	2016-02-29	josephak195@gmail.com	2
6	\$1,200.00	Active	09:02:00	2016-02-23	osama.samir.ahmed@gmail.com	1
5	\$15,250.00	Active	10:04:00	2016-02-23	aliamoharram@hotmail.com	4

The result:

cs421=> UPDATE bid set status= ‘Cancelled’ WHERE aid IN (select auction.aID from auction join bid on (auction.aID= bid.aID) AND (bid.status= 'Approved')) and status<>'Approved' ; UPDATE 4	cs421=> select * from bid; bidid price status time date email aid -----+-----+-----+-----+-----+-----+----- 2 \$200.00 Active 12:03:00 2016-02-23 zeyadnazemsaleh@gmail.com 3 3 \$300.00 Active 12:06:00 2016-02-23 lucien.george@mail.mcgill.ca 3 4 \$15,500.00 Approved 10:11:00 2016-02-23 zeyadnazemsaleh@gmail.com 4 10 \$8,000.00 Active 09:29:15.711991 2016-02-29 aliamoharram@hotmail.com 6 11 \$11,000.00 Active 09:29:30.59055 2016-02-29 aliamoharram@hotmail.com 7 1 \$100.00 Approved 11:32:00 2016-02-23 lucien.george@mail.mcgill.ca 2 9 \$1,250.00 Approved 08:42:54.160533 2016-02-29 aliamoharram@hotmail.com 1 7 \$120.00 Cancelled 08:36:23.87167 2016-02-29 aliamoharram@hotmail.com 2 8 \$130.00 Cancelled 08:36:40.402029 2016-02-29 josephak195@gmail.com 2 6 \$1,200.00 Cancelled 09:02:00 2016-02-23 osama.samir.ahmed@gmail.com 1 5 \$15,250.00 Cancelled 10:04:00 2016-02-23 aliamoharram@hotmail.com 4 (11 rows)
---	--

4.5 INSERTING A BUYER REVIEW GIVEN THE ITEMID OF THE ITEM BOUGHT

4.5.1 Discussion

This is an interesting data modification command as it inserts values of 2 different tables into the relationship that links them. After a bid gets approved, and the transaction is completed, the buyer and seller are asked to review each other; given the item id. We decided to implement the data modification command for a buyer review. The seller review insertion command should be similar with few changes.

4.5.2 Command

```
INSERT into reviews(reviewer_email, reviewee_email, rating, description) VALUES ((select bid.email from bid where bid.aid=(select auction.aid from auction where auction.itemid=3 ) and bid.status='Approved'),(select auction.email from auction where auction.itemid=3), 5, 'Lucien was on time');
```

4.5.3 Breakdown and results

The following is a breakdown of the insertion of the buyer review.

reviewer_email	reviewee_email	rating	description
lucien.george@mail.mcgill.ca	ossama.samir.ahmed@gmail.com	4	Ossama was honest about the condition of the item he was selling and the transaction was pretty simple. He was a little bit late though to our meeting.
aliaamoharram@hotmail.com	zeyadahazemsaleh@gmail.com	5	Zeyad described perfectly the condition of the item and made the transaction process really easy

The result of the modification command:

reviewer_email	reviewee_email	rating	description
lucien.george@mail.mcgill.ca	ossama.samir.ahmed@gmail.com	4	Ossama was honest about the condition of the item he was selling and the transaction was pretty simple. He was a little bit late though to our meeting.
aliaamoharram@hotmail.com	zeyadahazemsaleh@gmail.com	5	Zeyad described perfectly the condition of the item and made the transaction process really easy
zeyadahazemsaleh@gmail.com	lucien.george@mail.mcgill.ca	5	Lucien was on time

5 TASK VII: CREATE 2 VIEWS ON TOP OF THE DATABASE SCHEMA

5.1 LIMITED ACCESS TO USER INFORMATION

Sometimes it is preferable to hide some information from application developers, for security and privacy reasons. We have agreed that the email should be secured, and that some other information is irrelevant to show in the entity set users. This is why we create a view, called securedusers, to protect and hide some irrelevant information in the application domain.

5.1.1 Command

```
create view securedUsers
(fname, lname, gender, rating, phonenum)
as select users.fname, users.lname, users.gender,
users.rating, users.phonenum
from users;
```

5.1.2 The view

```
|cs421=> select * from users;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|email | fname | lname | dateofbirth | gender | university | address | rating | picture | phonenum
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|alihamoharram@hotmail.com | Aliaa | Moharram | 1995-01-08 | Female | McGill University | 3410 Peel | 5 | | +1(514) 243-6927
|lucien.george@mail.mcgill.ca | Lucien | George | 1995-09-16 | Male | McGill University | 3450 Drummond | 5 | | +1(514) 430-8272
|ossoama.samir.ahmed@gmail.com | Ossama Samir | Ahmed | 1992-10-21 | Male | McGill University | Samuel Home | 1 | | +1(514) 836-2217
|bettina.kemme@mcgill.ca | Bettina | Kemme | 2016-01-01 | Female | McGill University | Prof home | 5 | | Not Available
|zeyad.hazensaleh@gmail.com | Zeyad | Saleh | 1994-05-07 | Male | McGill University | Zeyad Home | 5 | | +1(514) 585-6769
|josephak195@gmail.com | Joseph | Akl | 1995-04-25 | Male | McGill University | 3460 Peel | 4 | | +1(514) 445 7873
(6 rows)

|cs421=> select * from securedusers ;
+-----+-----+-----+-----+-----+
|fname | lname | gender | rating | phonenum
+-----+-----+-----+-----+
|Aliaa | Moharram | Female | 5 | +1(514) 243-6927
|Lucien | George | Male | 5 | +1(514) 430-8272
|Ossama Samir | Ahmed | Male | 1 | +1(514) 836-2217
|Bettina | Kemme | Female | 5 | Not Available
|Zeyad | Saleh | Male | 5 | +1(514) 585-6769
|Joseph | Akl | Male | 4 | +1(514) 445 7873
(6 rows)
```

5.1.3 Updating the view

```
|cs421=> update securedusers SET rating = '3' where lname = 'Ahmed';
UPDATE 1
|cs421=> select * from securedusers ;
+-----+-----+-----+-----+-----+
|fname | lname | gender | rating | phonenum
+-----+-----+-----+-----+
|Aliaa | Moharram | Female | 5 | +1(514) 243-6927
|Lucien | George | Male | 5 | +1(514) 430-8272
|Bettina | Kemme | Female | 5 | Not Available
|Zeyad | Saleh | Male | 5 | +1(514) 585-6769
|Joseph | Akl | Male | 4 | +1(514) 445 7873
|Ossama Samir | Ahmed | Male | 3 | +1(514) 836-2217
(6 rows)

|cs421=> select * from users;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|email | fname | lname | dateofbirth | gender | university | address | rating | picture | phonenum
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|alihamoharram@hotmail.com | Aliaa | Moharram | 1995-01-08 | Female | McGill University | 3410 Peel | 5 | | +1(514) 243-6927
|lucien.george@mail.mcgill.ca | Lucien | George | 1995-09-16 | Male | McGill University | 3450 Drummond | 5 | | +1(514) 430-8272
|bettina.kemme@mcgill.ca | Bettina | Kemme | 2016-01-01 | Female | McGill University | Prof home | 5 | | Not Available
|zeyad.hazensaleh@gmail.com | Zeyad | Saleh | 1994-05-07 | Male | McGill University | Zeyad Home | 5 | | +1(514) 585-6769
|josephak195@gmail.com | Joseph | Akl | 1995-04-25 | Male | McGill University | 3460 Peel | 4 | | +1(514) 445 7873
|ossoama.samir.ahmed@gmail.com | Ossama Samir | Ahmed | 1992-10-21 | Male | McGill University | Samuel Home | 3 | | +1(514) 836-2217
(6 rows)
```

The rating was updated in the view as well as in the base relation as it is easy to trace back which tuple we are referring to, since the view corresponds to only one table.

5.2 ALL PARTICIPANTS IN ALL AUCTIONS

A useful view was a view showing all active users in the system. This was done by creating the view auctionparticipants

5.2.1 Command

```
create view auctionParticipants
(bidderid, sellerid, bidid, aid, bidstatus, auctionstatus)
as select bid.email, auction.email, bidid, auction.aid, bid.status, auction.status
from bid inner join auction on bid.aid = auction.aid;
```

5.2.2 The view

```
[cs421=> select * from auction order by aid;
+-----+-----+-----+-----+-----+-----+
| aid | initialprice | starttime | date   | status  |           email        | itemid |
+-----+-----+-----+-----+-----+-----+
| 1  | $1,148.00    | 09:00:00  | 2016-02-28 | Active  | zeyadnazemsaleh@gmail.com | 1
| 2  | $90.00       | 11:30:00  | 2015-12-20  | Active  | ossama.samir.ahmed@gmail.com | 2
| 3  | $260.00      | 12:00:00  | 2016-02-28  | Active  | aliaamoharram@hotmail.com | 4
| 4  | $15,250.00   | 10:00:00  | 2016-02-28  | Active  | lucien.george@mail.mcgill.ca | 3
| 5  | $280.00      | 14:27:21.554515 | 2015-09-28  | Active  | zeyadnazemsaleh@gmail.com | 5
(5 rows)

[cs421=> select * from bid order by aid;
+-----+-----+-----+-----+-----+-----+
| bidid | price   | status | time   | date   |           email        | aid   |
+-----+-----+-----+-----+-----+-----+
| 6  | $1,200.00 | Active | 09:02:00 | 2016-02-23 | ossama.samir.ahmed@gmail.com | 1
| 1  | $100.00   | Active | 11:32:00 | 2016-02-23 | lucien.george@mail.mcgill.ca | 2
| 2  | $280.00   | Active | 12:03:00 | 2016-02-23 | zeyadnazemsaleh@gmail.com | 3
| 3  | $300.00   | Active | 12:06:00 | 2016-02-23 | lucien.george@mail.mcgill.ca | 3
| 5  | $15,250.00 | Cancelled | 10:04:00 | 2016-02-23 | aliaamoharram@hotmail.com | 4
| 4  | $15,500.00 | Approved  | 10:11:00 | 2016-02-23 | zeyadnazemsaleh@gmail.com | 4
(6 rows)

[cs421=> select * from auctionparticipants order by aid;
+-----+-----+-----+-----+-----+-----+
| bidderid |           sellerid | bidid | aid | bidstatus | auctionstatus |
+-----+-----+-----+-----+-----+-----+
| ossama.samir.ahmed@gmail.com | zeyadnazemsaleh@gmail.com | 1 | 1 | Active | Active
| lucien.george@mail.mcgill.ca | ossama.samir.ahmed@gmail.com | 1 | 2 | Active | Active
| zeyadnazemsaleh@gmail.com | aliaamoharram@hotmail.com | 2 | 3 | Active | Active
| lucien.george@mail.mcgill.ca | aliaamoharram@hotmail.com | 3 | 3 | Active | Active
| aliaamoharram@hotmail.com | lucien.george@mail.mcgill.ca | 5 | 4 | Cancelled | Active
| zeyadnazemsaleh@gmail.com | lucien.george@mail.mcgill.ca | 4 | 4 | Approved | Active
(6 rows)
```

5.2.3 Updating the view

```
[cs421=> update auctionparticipants set bidstatus = 'Approved' where bidid = 1;
ERROR:  cannot update view "auctionparticipants"
DETAIL:  Views that do not select from a single table or view are not automatically updatable.
HINT:  To enable updating the view, provide an INSTEAD OF UPDATE trigger or an unconditional ON UPDATE DO INSTEAD rule.
```

None of the tables got updated. The error was self explanatory, a rule needs to be set that specifies what to do in case of an update on a view relating more than 2 tables.

6 TASK VIII: CHECK CONSTRAINTS

6.1 REVIEWS CHECKS

The reviews section needs multiple checks to satisfy the integrity constraint.

Table "cs421g16.reviews"		
Column	Type	Modifiers
reviewer_email	character varying(200)	
reviewee_email	character varying(200)	
description	character varying(300)	
rating	integer	

Check constraints:

```
"different" CHECK (reviewer_email::text <> reviewee_email::text)
"reviews_rating_check" CHECK (rating > 0 AND rating < 6)
```

6.1.1 Reviewee different than reviewer

```
alter table reviews add constraint different check(reviewer_email::text <> reviewee_email::text);
cs421=> Insert into reviews(reviewer_email,reviewee_email) values('ossama.ahmed@mail.mcgill.ca','ossama.ahmed@mail.mcgill.ca')
cs421=> ;
ERROR: new row for relation "reviews" violates check constraint "different"
DETAIL: Failing row contains (ossama.ahmed@mail.mcgill.ca, ossama.ahmed@mail.mcgill.ca, null, null).
```

6.1.2 The rating must be less than 1 or larger than 5

```
alter table reviews add constraint reviews_rating_check check (rating >0 and rating <6);
cs421=> Insert into reviews(reviewer_email,reviewee_email,rating) values('ossama.ahmed@mail.mcgill.ca','zeyad.saleh@mail.mcgill.ca', 7)
;
ERROR: new row for relation "reviews" violates check constraint "reviews_rating_check"
DETAIL: Failing row contains (ossama.ahmed@mail.mcgill.ca, zeyad.saleh@mail.mcgill.ca, null, 7).
cs421=> Insert into reviews(reviewer_email,reviewee_email,rating) values('ossama.ahmed@mail.mcgill.ca','zeyad.saleh@mail.mcgill.ca', -1)
;
ERROR: new row for relation "reviews" violates check constraint "reviews_rating_check"
DETAIL: Failing row contains (ossama.ahmed@mail.mcgill.ca, zeyad.saleh@mail.mcgill.ca, null, -1).
cs421=> ■
```

6.2 AUCTION

An auction corresponds to only 1 item, thus we have to add the following check constraint:

```
alter table auction add constraint unique_itemid check unique(itemid);
```

```
cs421=> insert into auction(aid, itemid) values(30,1)
cs421=> ;
ERROR: duplicate key value violates unique constraint "unique_itemid"
DETAIL: Key (itemid)=(1) already exists.
```

6.3 APPROVES

Need to check that the seller did not buy his own goods. This is done using the following command:

```
alter table approves seller_buyer check (seller_email::text <> buyer_email::text);
cs421=> Insert into approves(seller_email,buyer_email) values('ossama.ahmed@mail.mcgill.ca','ossama.ahmed@mail.mcgill.ca')
;
ERROR: new row for relation "approves" violates check constraint "seller_buyer"
DETAIL: Failing row contains (null, null, ossama.ahmed@mail.mcgill.ca, ossama.ahmed@mail.mcgill.ca, null).
```

7 EXTRA WORK: AN ANDROID APP THAT INTEGRATES SOME OF THE QUERIES WE DISCUSSED

Offers on my Listing (iOS)

Custom Search

Home

Make a Bid

My Listings

CONFIRMED LISTINGS

- Baguette \$200 Contact Buyer Sam Ahmed

ACTIVE LISTINGS

(SLIDE LISTINGS FOR OPTIONS)

- Timberland Winter Boo Asking offer: \$0 Best offer: \$0 3
- Motorola Fast Charger Asking offer: \$0 Best offer: \$0 3
- Used box of Apple Juic Asking offer: \$2 Best offer: \$0 3

New Listing

Free For Sale

Item name

Item Description max 240 characters

Item type

Add an image

Add Listing

Create a new Listing (iOS)

The above screenshots were taken from the app that we developed to integrate some of the sql queries and modification commands that we discussed earlier. The database used for the development of this app was the NOSQL third party provider Firebase.io.

Code is available at: <https://github.com/ossamaAhmed/McHacks-FFS>