MSIS 2603 - DBMS
Fall 2016
Project Report

Team - TechWizards

Malya Jain (W1279111)
Mishita Agarwal (W1264507)
Sayari Ghosh (W1248420)

Table of Contents

1. BUSINESS APPLICATION DESCRIPTION

Sell-It - Ecommerce Platform to sell used goods

Information technology and the Internet have had a dramatic effect on business operations. Companies with substantial buying power are racing to invest widely in e-commerce applications.

Sell-It is an emerging classified ecommerce platform, which provides an opportunity to the local communities with vibrant online marketplaces. It connects sellers and buyers on a single platform and enable them to sell, buy or exchange used goods with the help of few clicks, thereby reducing an overhead of physical interactions. The start-up became operational from September 2015 and the revenues are increasing at a good rate.

Sell-It showcases a wide range of products, including electronics, books, furniture, and vehicles. The motivation behind this business idea is to make avail the used items at smart prices to the customers and reduce the painstaking procedure of selling the items for the seller, on a single platform. Front-end interface involves the buyers and sellers and back-end involves the Admin and Business users. Admin manages the user's account and performs a background check for all the products and sellers before posting an advertisement and makes sure that no duplicate items are posted.

2. USER TYPES OR ENTITIES

SELL-It has the following users or entities:

1. **Buyer** - Buyer logs into the website, searches for desired product using the search criteria, looks at different ads and selects an ad which satisfies his/her requirements for buying an item.
2. **Seller** - Seller posts ads on the website of items that he/she wants to sell online on this website.
3. **Admin** - Admin regulates the online boarding of buyers and sellers, authorizes the advertisement before posting on website and checks for any duplication of items posted by the seller.
4. **Business User** - Business generates analytics for the company and collects payments and transfers them to the seller.

3. USER CASES

a. BUYER:

1. Find buyers who have not bought any listing greater than $1000
2. How many buyers bid on a listing before the bid actually got selected?
3. The buyer Yaroslav Zuev wants to search for available electronics (tablets) from the listings available on Sell-It, sorted on the basis of seller listing
4. Sell-It decides to give 20% discount coupon to the buyer who made the maximum and second maximum purchase on the website till date. Thomas Satchel pulls out a list of buyers with their expenditure on the website
5. The business wants to target the buyers who have not made any purchase till now so that the advertisements can be sent to them via email. How many inactive buyers are there?
6. Calculate the highest bids for all the unique listing IDs
7. The buyer details for the shipping just got updated for LISTING_ID 4004.
8. Which carriers have never shipped to a buyer with the zipcode 97852
9. Which buyers have bought from sellers who are from the same region?
10. Find all the Buyers or those Buyer id's which have bought in 2015 but not bought at all in 2016
11. Find the status of shipping of the item purchased by Oscar A. Drake where tracking number is 1379573'
12. Find the number of Buyers per region.

b. SELLER:

1. What is the second highest BID FOR Listing id 4005?
2. What is the percent increase in the premium sellers on Sell-It from Quarter 1 2016 to Q3 2016?
3. How many premium and normal sellers are there on Sell-It till date?
4. The seller Steven L. Bratton wants to check the estimated selling price for the mobile phone that the wants to sell. Calculate the price.
5. Seller Sara Cowell (Seller_ID = 2005) decides to change the membership from normal to premium, with payment made by credit card.
6. Sellers want to update their membership whose accounts have expired
7. Which sellers have renewed their accounts more than 1 time
8. How many listings have the maximum success rate criteria via social media
9. Find the number of premium sellers sorted by their payment mode (CC/Paypal)
10. What is the maximum average rating per region?
11. Find Carrier(s) with the most number of packages shipped

c. ADMIN:

1. Which users are both buyers and sellers on Sell-It?
2. Admin runs a query to check the buyers who have made purchases more than once
3. Admin wants to check the sellers who have sold listings more than once on Sell-It
4. Which listing successful transactions have been received by buyer?
5. Admin wants to check the listings which are still open on Sell-It to track them to closure
6. What is the maximum amount paid by the shipments per carrier? What is the name of that carrier?
7. Admin wants to check the maximum amount on which the mobile phone was sold on the website ever
8. Admin Noel Glover wants to check those Buyers whose listings are shipped via different or same carrier in the Zipcode 55402
9. Admin Mitchell Archam wants to see the full listing of the transactions and details of the items sold
10. When admin runs the query with the third parties for the shipment status, he notices that transaction_id 1671549 AND 525845 have been delivered. He decides to change the status to delivered and get corresponding transaction_id, listing_id, seller_id and buyer_id
11. Pull out the shipping details of all the shipments

d. BUSINESS USER:

1. Find all the Buyers who have bought from category Books but not bought from category called Furniture.
2. Find region with Buyers who have bought maximum number of times.
3. Which region has sellers who have sold maximum number of times
4. Calculate the total payments received via the Paypal platform
5. Find the most popular category which is listed by the Seller
6. Business user Robert Pattinson want to do the comparative analysis of the Social media Platform 'Facebook' to record its progress in the past quarter wise
7. Business user Thomas Sachtel want to do the comparative analysis of the Social media Platform 'Facebook' to record its progress in the past quarter wise
8. Robert Pattinson is the founder and CEO of Sell-It and wants to check which employees have been in the organization for more than 1 year and eligible for promotion

4. LIST OF TABLES

1. EMPLOYEE
2. SELLER
3. BUYER
4. PURCHASE
5. LISTING
6. CATEGORY
7. SUBCATEGORY
8. BID
9. RATING
10. PAYMENT
11. SHIPPING
12. ADVERTISEMENT

5. DATABASE DICTIONARY - PHYSICAL SCHEMA

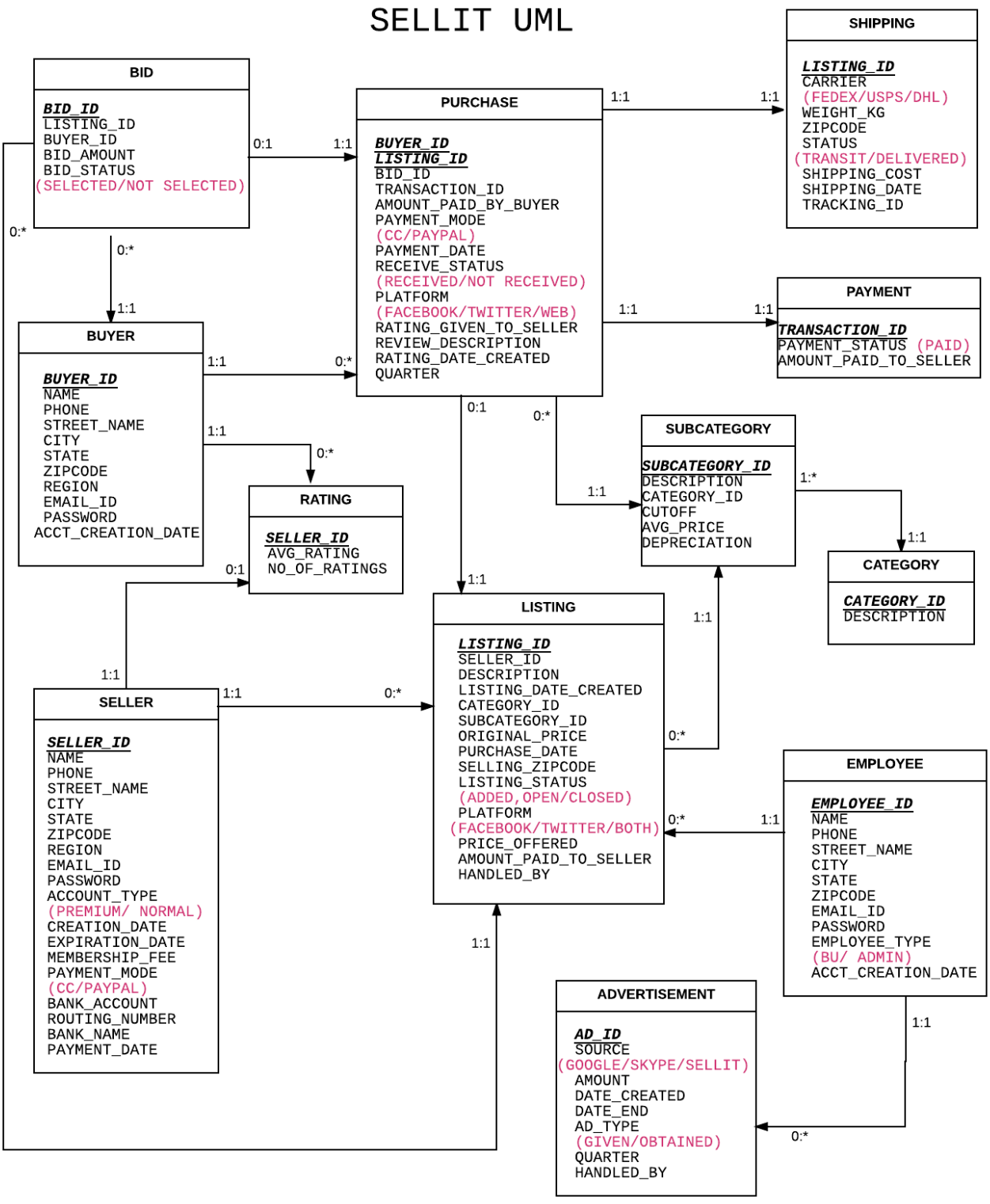
EMPLOYEE	
EMPLOYEE_ID	**INT NOT NULL AUTO_INCREMENT PRIMARY KEY**
NAME	**VARCHAR(50) NOT NULL**
PHONE	**VARCHAR(20) NOT NULL**
STREET_NAME	**VARCHAR(50) NOT NULL**
CITY	**VARCHAR(20) NOT NULL**
STATE	**VARCHAR(20) NOT NULL**
ZIPCODE	**CHAR(5) NOT NULL**
EMAIL_ID	**VARCHAR(50) NOT NULL**
PASSWORD	**VARCHAR(20) NOT NULL**
EMPLOYEE_TYPE	**VARCHAR(10) NOT NULL CAN BE (BU/ ADMIN)**
ACCT_CREATION_DATE	**DATETIME NOT NULL**
SELLER	
SELLER_ID	**INT NOT NULL AUTO_INCREMENT PRIMARY KEY**
NAME	**VARCHAR(50) NOT NULL**
PHONE	**VARCHAR(20) NOT NULL**
STREET_NAME	**VARCHAR(50) NOT NULL**
CITY	**VARCHAR(20) NOT NULL**
STATE	**VARCHAR(20) NOT NULL**
ZIPCODE	**CHAR(5) NOT NULL**
REGION	**VARCHAR(12) NOT NULL**
EMAIL_ID	**VARCHAR(50) NOT NULL**
PASSWORD	**VARCHAR(50) NOT NULL**
ACCOUNT_TYPE	**VARCHAR(10) NOT NULL CAN BE (PREMIUM, NORMAL)**
CREATION_DATE	**DATETIME NOT NULL**

EXPIRATION_DATE	DATETIME NOT NULL
MEMBERSHIP_FEE	INT NOT NULL
PAYMENT_MODE	VARCHAR(10) CAN BE (CC,PAYPAL)
BANK_ACCOUNT	INT NOT NULL
ROUTING_NUMBER	INT NOT NULL
BANK_NAME	VARCHAR(20) NOT NULL
PAYMENT_DATE	DATETIME
BUYER	
BUYER_ID	INT NOT NULL AUTO_INCREMENT PRIMARY KEY
NAME	VARCHAR(50) NOT NULL
PHONE	VARCHAR(20) NOT NULL
STREET_NAME	VARCHAR(50) NOT NULL
CITY	VARCHAR(20) NOT NULL
STATE	VARCHAR(20) NOT NULL
ZIPCODE	CHAR(5) NOT NULL
REGION	VARCHAR(12) NOT NULL
EMAIL_ID	VARCHAR(50) NOT NULL
PASSWORD	VARCHAR(50) NOT NULL
ACCT_CREATION_DATE	DATETIME NOT NULL
LISTING	
LISTING_ID	INT NOT NULL AUTO_INCREMENT PRIMARY KEY
SELLER_ID	INT NOT NULL FOREIGN KEY REFERENCES SELLER(SELLER_ID)
DESCRIPTION	VARCHAR (150) NOT NULL
CATEGORY_ID	VARCHAR (20) NOT NULL FOREIGN KEY REFERENCES CATEGORY(CATEGORY_ID)
SUBCATEGORY_ID	VARCHAR (20) NOT NULL FOREIGN KEY REFERENCES SUBCATEGORY(SUBCATEGORY_ID)
ORIGINAL_PRICE	DECIMAL (10,2) NOT NULL
PURCHASE_DATE	DATETIME NOT NULL
SELLING_ZIPCODE	CHAR(5) NOT NULL
LISTING_STATUS	VARCHAR(10) NOT NULL CAN BE(ADDED, OPEN, CLOSE)
PLATFORM	VARCHAR(10) NOT NULL CAN BE (FACEBOOK, TWITTER, BOTH)
PRICE_OFFERED	DECIMAL(10,2) NOT NULL
AMOUNT_PAID_TO_SELLER	DECIMAL(10,2) NOT NULL
HANDLED_BY	VARCHAR(10) NOT NULL FOREIGN KEY REFERENCES EMPLOYEE(EMPLOYEE_ID)
LISTING_DATE_CREATED	DATETIME NOT NULL
PURCHASE	
BUYER_ID	INT NOT NULL FOREIGN KEY REFERENCES BUYER(BUYER_ID) PRIMARY KEY

LISTING_ID	INT NOT NULL FOREIGN KEY REFERENCES SELLER_LISTING(LISTING_ID) PRIMARY KEY
PAYMENT_STATUS	VARCHAR(10) NOT NULL CAN BE (PAID, NOT PAID)
TRANSACTION_ID	INT NOT NULL FOREIGN KEY REFERENCES PAYMENT(TRANSACTION_ID)
AMOUNT_PAID_BY_BUYER	DECIMAL(10,2) NOT NULL
PAYMENT_MODE	VARCHAR(10) CAN BE(CC,PAYPAL)
PAYMENT_DATE	DATETIME NOT NULL
RECEIVE_STATUS	VARCHAR(20) NOT NULL CAN BE (RECEIVED, NOT RECEIVED)
PLATFORM(	VARCHAR(10) CAN BE (FACEBOOK, TWITTER, WEB)
RATING_GIVEN_TO_SELLER	INT NOT NULL
REVIEW_DESCRIPTION	VARCHAR (150)
RATING_DATE_CREATED	DATETIME
BID_ID	INT NOT NULL FOREIGN KEY REFERENCES BID(BID_ID)
QUARTER	VARCHAR(20) NOT NULL
CATEGORY	
CATEGORY_ID	INT NOT NULL AUTO_INCREMENT PRIMARY KEY
DESCRIPTION	VARCHAR (50) NOT NULL
SUBCATEGORY	
SUBCATEGORY_ID	INT NOT NULL AUTO_INCREMENT PRIMARY KEY
DESCRIPTION	VARCHAR (50) NOT NULL
CATEGORY_ID	INT NOT NULL FOREIGN KEY REFERENCES CATEGORY(CATEGORY_ID)
CUTOFF_%	INT NOT NULL
AVG_PRICE	INT NOT NULL
DEPRECIATION	INT NOT NULL
RATING	
SELLER_ID	INT NOT NULL FOREIGN KEY REFERENCES SELLER(SELLER_ID) PRIMARY KEY
AVG_RATING	INT NOT NULL
NO_OF_RATINGS	INT NOT NULL
BID	
BID_ID	INT NOT NULL FOREIGN KEY REFERENCES SELLER_LISTING(LISTING_ID) PRIMARY KEY
LISTING_ID	INT NOT NULL FOREIGN KEY REFERENCES SELLER_LISTING(LISTING_ID)
BUYER_ID	INT NOT NULL FOREIGN KEY REFERENCES SELLER_LISTING(LISTING_ID)
BID_AMOUNT	DECIMAL(10,2)

BID_STATUS	**VARCHAR(20) NOT NULL CAN BE(SELECTED, NOT SELECTED)**
PAYMENT	
TRANSACTION_ID	**INT NOT NULL AUTO_INCREMENT PRIMARY KEY**
PAYMENT_STATUS	**VARCHAR(20) NOT NULL CAN BE (PAID, NOT PAID)**
AMOUNT_PAID_TO_SELLER	**DECIMAL(10,2) NOT NULL**
ADVERTISEMENT	
AD_ID	**INT NOT NULL AUTO_INCREMENT PRIMARY KEY**
SOURCE	**VARCHAR(20) NOT NULL CAN BE (GOOGLE, SKYPE,SELLIT)**
AMOUNT	**DECIMAL(10,2) NOT NULL**
DATE_CREATED	**DATETIME NOT NULL**
DATE_END	**DATETIME NOT NULL**
AD_TYPE	**VARCHAR(20) NOT NULL CAN BE (GIVEN, OBTAINED)**
QUARTER	**VARCHAR(20) NOT NULL**
HANDLED_BY	**INT NOT NULL FOREIGN KEY REFERENCES EMPLOYEE(EMPLOYEE_ID)**
SHIPPING	
LISTING_ID	**INT NOT NULL AUTO_INCREMENT PRIMARY KEY**
CARRIER	**VARCHAR(20) NOT NULL CAN BE (FEDEX, USPS, DHL)**
WEIGHT_KG	**DECIMAL(5,2) NOT NULL**
ZIPCODE	**CHAR(5) NOT NULL**
STATUS	**VARCHAR(20) NOT NULL CAN BE(TRANSIT, DELIVERED)**
SHIPPING_COST	**DECIMAL(10,2) NOT NULL**
SHIPPING_DATE	**DATETIME NOT NULL**
TRACKING_ID	**INT NOT NULL**

6. UML

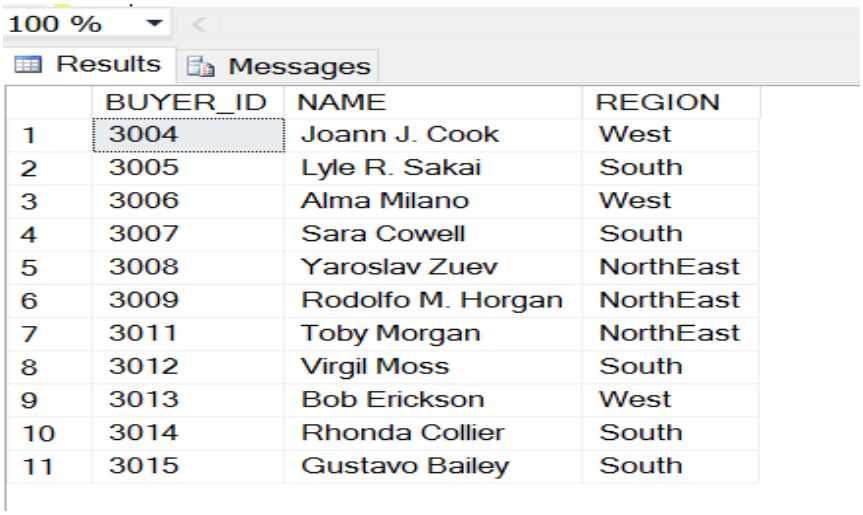

SELLIT UML
BID
BID_ID
LISTING_ID
BUYER_ID
BID_AMOUNT
BID_STATUS
(SELECTED/NOT SELECTED)
PURCHASE
BUYER_ID
LISTING_ID
BID_ID
TRANSACTION_ID
AMOUNT_PAID_BY_BUYER
PAYMENT_MODE
(CC/PAYPAL)
PAYMENT_DATE
RECEIVE_STATUS
(RECEIVED/NOT RECEIVED)
PLATFORM
(FACEBOOK/TWITTER/WEB)
RATING_GIVEN_TO_SELLER
REVIEW_DESCRIPTION
RATING_DATE_CREATED
QUARTER
SHIPPING
LISTING_ID
CARRIER
(FEDEX/USPS/DHL)
WEIGHT_KG
ZIPCODE
STATUS
(TRANSIT/DELIVERED)
SHIPPING_COST
SHIPPING_DATE
TRACKING_ID
PAYMENT
TRANSACTION_ID
PAYMENT_STATUS (PAID)
AMOUNT_PAID_TO_SELLER
BUYER
BUYER_ID
NAME
PHONE
STREET_NAME
CITY
STATE
ZIPCODE
REGION
EMAIL_ID
PASSWORD
ACCT_CREATION_DATE
RATING
SELLER_ID
AVG_RATING
NO_OF_RATINGS
SUBCATEGORY
SUBCATEGORY_ID
DESCRIPTION
CATEGORY_ID
CUTOFF
AVG_PRICE
DEPRECIATION
CATEGORY
CATEGORY_ID
DESCRIPTION
LISTING
LISTING_ID
SELLER_ID
DESCRIPTION
LISTING_DATE_CREATED
CATEGORY_ID
SUBCATEGORY_ID
ORIGINAL_PRICE
PURCHASE_DATE
SELLING_ZIPCODE
LISTING_STATUS
(ADDED,OPEN/CLOSED)
PLATFORM
(FACEBOOK/TWITTER/BOTH)
PRICE_OFFERED
AMOUNT_PAID_TO_SELLER
HANDLED_BY
SELLER
SELLER_ID
NAME
PHONE
STREET_NAME
CITY
STATE
ZIPCODE
REGION
EMAIL_ID
PASSWORD
ACCOUNT_TYPE
(PREMIUM/ NORMAL)
CREATION_DATE
EXPIRATION_DATE
MEMBERSHIP_FEE
PAYMENT_MODE
(CC/PAYPAL)
BANK_ACCOUNT
ROUTING_NUMBER
BANK_NAME
PAYMENT_DATE
EMPLOYEE
EMPLOYEE_ID
NAME
PHONE
STREET_NAME
CITY
STATE
ZIPCODE
EMAIL_ID
PASSWORD
EMPLOYEE_TYPE
(BU/ ADMIN)
ACCT_CREATION_DATE
ADVERTISEMENT
AD_ID
SOURCE
(GOOGLE/SKYPE/SELLIT)
AMOUNT
DATE_CREATED
DATE_END
AD_TYPE
(GIVEN/OBTAINED)
QUARTER
HANDLED_BY

7. USE CASES - QUERIES PER USER TYPE

Buyer

1) Find buyers who have not bought any listing greater than $1000

```
SELECT BUYER_ID,NAME,REGION  FROM BUYER B
WHERE B.BUYER_ID NOT IN
(SELECT DISTINCT B.BUYER_ID
FROM PURCHASE P JOIN SHIPPING S
ON P.LISTING_ID = S.LISTING_ID
JOIN BUYER B ON B.BUYER_ID = P.BUYER_ID
WHERE (P.AMOUNT_PAID_BY_BUYER-S.SHIPPING_COST) > 1000)
```



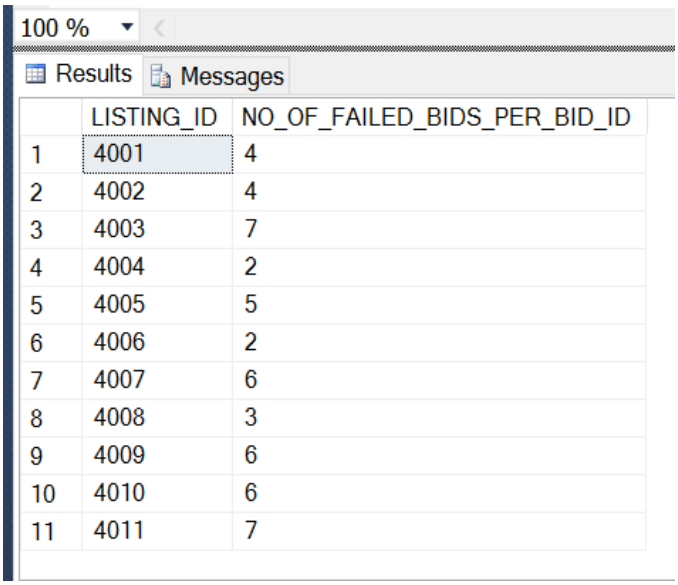
100 %

Results | Messages

	BUYER_ID	NAME	REGION
1	3004	Joann J. Cook	West
2	3005	Lyle R. Sakai	South
3	3006	Alma Milano	West
4	3007	Sara Cowell	South
5	3008	Yaroslav Zuev	NorthEast
6	3009	Rodolfo M. Horgan	NorthEast
7	3011	Toby Morgan	NorthEast
8	3012	Virgil Moss	South
9	3013	Bob Erickson	West
10	3014	Rhonda Collier	South
11	3015	Gustavo Bailey	South

2) How many buyers bid on a listing before the bid actually got selected?

```
SELECT LISTING_ID,COUNT(*) AS 'NO_OF_FAILED_BIDS_PER_BID_ID'
FROM BID B
WHERE BID_STATUS = 'NOT SELECTED'
GROUP BY B.LISTING_ID
```



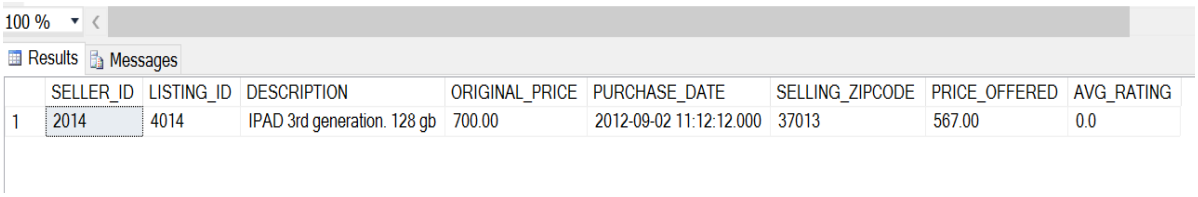
100 %

Results | Messages

	LISTING_ID	NO_OF_FAILED_BIDS_PER_BID_ID
1	4001	4
2	4002	4
3	4003	7
4	4004	2
5	4005	5
6	4006	2
7	4007	6
8	4008	3
9	4009	6
10	4010	6
11	4011	7

3) **The buyer Yaroslav Zuev wants to search for available electronics (tablets) from the listings available on Sell-It, sorted on the basis of seller listing**

```
SELECT S.SELLER_ID,L.LISTING_ID, L.DESCRIPTION, L.ORIGINAL_PRICE, L.PURCHASE_DATE,
L.SELLING_ZIPCODE, L.PRICE_OFFERED, R.AVG_RATING

FROM SELLER S JOIN RATING R

ON S.SELLER_ID = R.SELLER_ID

JOIN LISTING L ON S.SELLER_ID = L.SELLER_ID

JOIN SUBCATEGORY SC ON L.SUBCATEGORY_ID = SC.SUBCATEGORY_ID

WHERE L.LISTING_STATUS = 'OPEN'

AND SC.DESCRIPTION LIKE 'TABLET'

ORDER BY R.AVG_RATING DESC
```



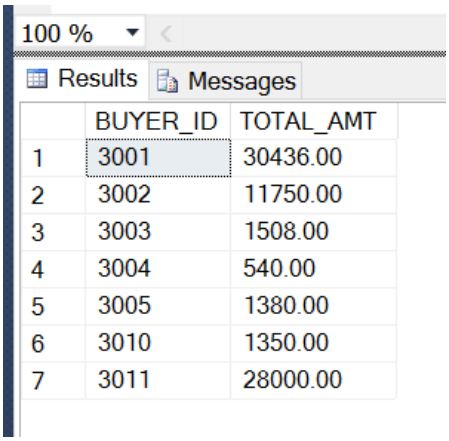
100 %

Results | Messages

	SELLER_ID	LISTING_ID	DESCRIPTION	ORIGINAL_PRICE	PURCHASE_DATE	SELLING_ZIPCODE	PRICE_OFFERED	AVG_RATING
1	2014	4014	IPAD 3rd generation. 128 gb	700.00	2012-09-02 11:12:12.000	37013	567.00	0.0

4) **Sell-It decides to give 20% discount coupon to the buyer who made the maximum and second maximum purchase on the website till date. Thomas Satchel pulls out a list of buyers with their expenditure on the website**

```
SELECT P.BUYER_ID, SUM(P.AMOUNT_PAID_BY_BUYER) AS 'TOTAL_AMT'

FROM PURCHASE P JOIN LISTING L

ON P.LISTING_ID = L.LISTING_ID

WHERE L.LISTING_STATUS = 'CLOSED'

GROUP BY P.BUYER_ID
```



100 %

Results | Messages

	BUYER_ID	TOTAL_AMT
1	3001	30436.00
2	3002	11750.00
3	3003	1508.00
4	3004	540.00
5	3005	1380.00
6	3010	1350.00
7	3011	28000.00

5) **The business wants to target the buyers who have not made any purchase till now so that the advertisements can be sent to them via email. How many inactive buyers are there?**

```
SELECT COUNT(B.BUYER_ID) AS 'INACTIVE_BUYERS'
FROM BUYER B
WHERE B.BUYER_ID NOT IN
(SELECT P1.BUYER_ID
FROM PURCHASE P1
WHERE P1.TRANSACTION_ID IN
(SELECT DISTINCT P.TRANSACTION_ID
FROM PAYMENT P
WHERE P.PAYMENT_STATUS = 'PAID'))
```

Results | Messages

	INACTIVE_BUYERS
1	8

6) **Calculate the highest bids for all the unique listing IDs**

```
SELECT L.LISTING_ID, MAX(B.BID_AMOUNT)
AS 'MAX_BID'
FROM BID B LEFT OUTER JOIN LISTING L ON B.LISTING_ID = L.LISTING_ID
GROUP BY L.LISTING_ID
```

Results Messages

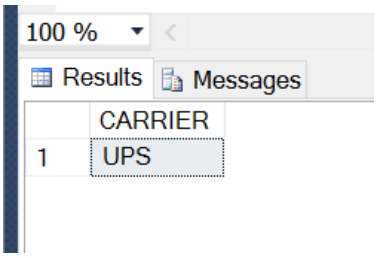
	LISTING_ID	MAX_BID
1	4001	419.00
2	4002	1408.00
3	4003	27000.00
4	4004	540.00
5	4005	1700.00
6	4006	140.00
7	4007	1080.00
8	4008	850.00
9	4009	450.00
10	4010	10500.00
11	4011	25000.00

7) The buyer details for the shipping just got updated for LISTING_ID 4004.

```
UPDATE SHIPPING

SET CARRIER = 'FEDEX', WEIGHT_KG = 1.5, ZIPCODE = 80220,

STATUS = 'DELIVERED', SHIPPING_COST = 100, SHIPPING_DATE =

'2016-03-31 13:21:53', TRACKING_ID = 54729

WHERE LISTING_ID = 4004
```

8) Which carriers have never shipped to the zipcode 97852

```
SELECT DISTINCT S.CARRIER

FROM SHIPPING S

WHERE CARRIER NOT IN

(SELECT S1.CARRIER

FROM SHIPPING S1

WHERE S1.ZIPCODE = 97852)
```



9) Which buyers have bought from sellers who are from the same region?

```
SELECT DISTINCT B.BUYER_ID, B.NAME
```

```
FROM BUYER B JOIN PURCHASE P ON B.BUYER_ID = P.BUYER_ID JOIN
LISTING L ON P.LISTING_ID = L.LISTING_ID JOIN SELLER S ON
S.SELLER_ID = L.SELLER_ID
WHERE S.REGION = B.REGION
```

Results | Messages

	BUYER_ID	NAME
1	3001	Oscar A. Drake
2	3002	Angie T. Kettle
3	3003	Elizabeth N. Posey
4	3004	Joann J. Cook
5	3005	Lyle R. Sakai
6	3010	Victor Ribeiro Dias

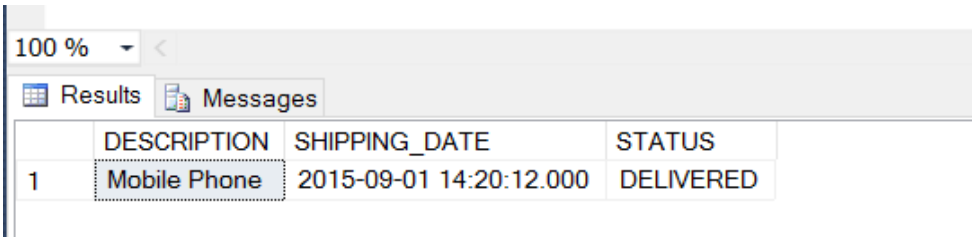
10) Find all the Buyers or those Buyer id's which have bought in 2015 but not bought at all in 2016

```
SELECT P.BUYER_ID, P.LISTING_ID, P.AMOUNT_PAID_BY_BUYER
FROM PURCHASE P
WHERE DATEDIFF(YEAR,'2015-12-31',P.PAYMENT_DATE) <1
AND P.BUYER_ID NOT IN
(SELECT DISTINCT P1.BUYER_ID
FROM PURCHASE P1
WHERE DATEDIFF(YEAR,PAYMENT_DATE,'2016-12-31') < 1)
```

	BUYER_ID	LISTING_ID	AMOUNT_PAID_BY_BUYER
1	3003	4002	1508.00

11) Find the status of shipping of the item purchased by Oscar A. Drake where tracking number is '1379573'

```
SELECT SU.DESCRIPTION, S.SHIPPING_DATE, S.STATUS
FROM SHIPPING S, LISTING L, PURCHASE P, SUBCATEGORY SU
WHERE S.LISTING_ID = P.LISTING_ID
AND P.LISTING_ID = L.LISTING_ID
AND L.SUBCATEGORY_ID = SU.SUBCATEGORY_ID
AND S.TRACKING_ID = '1379573'
```



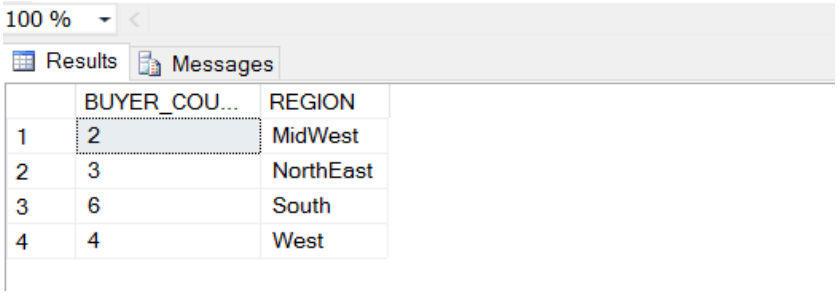
100 %

Results | Messages

	DESCRIPTION	SHIPPING_DATE	STATUS
1	Mobile Phone	2015-09-01 14:20:12.000	DELIVERED

11) Find the number of Buyers per region.

```
SELECT COUNT(B.BUYER_ID) AS 'BUYER_COUNT', B.REGION

FROM BUYER B

GROUP BY REGION
```



100 %

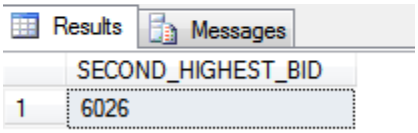
Results | Messages

	BUYER_COU...	REGION
1	2	MidWest
2	3	NorthEast
3	6	South
4	4	West

Seller

12) What is the second highest BID FOR Listing id 4005?

```
SELECT MAX(B.BID_ID) AS 'SECOND_HIGHEST_BID'

FROM BID B

WHERE B.LISTING_ID = 4005

AND B.BID_STATUS = 'NOT SELECTED'
```



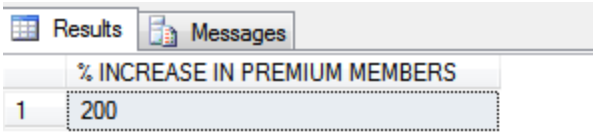
Results | Messages

	SECOND_HIGHEST_BID
1	6026

13) What is the percent increase in the premium sellers on Sell-It from Quarter 1 2016 to Q3 2016?

```
SELECT ((T2.PREMIUM_COUNT2-T1.PREMIUM_COUNT1)*100/T1.PREMIUM_COUNT1)

AS '% INCREASE IN PREMIUM MEMBERS'

FROM

(SELECT COUNT(*) AS 'PREMIUM_COUNT1'

FROM SELLER S1

WHERE S1.ACCOUNT_TYPE = 'PREMIUM'
```

```
AND S1.CREATION_DATE > = '2015-09-01'
AND S1.CREATION_DATE <='2015-12-31')T1,
(SELECT COUNT(*) AS 'PREMIUM_COUNT2'
FROM SELLER S2
WHERE S2.ACCOUNT_TYPE = 'PREMIUM'
AND S2.CREATION_DATE > = '2016-01-01'
AND S2.CREATION_DATE <='2016-12-31') T2
```

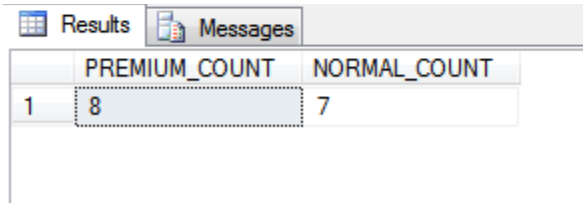


Results | Messages

	% INCREASE IN PREMIUM MEMBERS
1	200

14) How many premium and normal sellers are there on Sell-It till date?

```
SELECT T1.PREMIUM_COUNT, T2.NORMAL_COUNT
FROM
(SELECT COUNT(*) AS 'PREMIUM_COUNT'
FROM SELLER
WHERE SELLER.ACCOUNT_TYPE = 'PREMIUM') T1,
(SELECT COUNT(*) 'NORMAL_COUNT'
FROM SELLER
WHERE SELLER.ACCOUNT_TYPE = 'NORMAL') T2
```



Results | Messages

	PREMIUM_COUNT	NORMAL_COUNT
1	8	7

15) The seller Steven L. Bratton wants to check the estimated selling price for the mobile phone that he wants to sell. Calculate the price.

```
SELECT S.SUBCATEGORY_ID,S.DESCRIPTION,S.CATEGORY_ID,S.ESTIMATED_SELLING_PRICE
FROM ESTIMATED_SELLING_PRICE S
WHERE S.DESCRIPTION LIKE 'MOBILE%'
```

100 %

Results | Messages

	SUBCATEGORY_ID	DESCRIPTION	CATEGORY_ID	ESTIMATED_SELLING_PRICE
1	8001	Mobile Phone	7001	470

16) **Seller Sara Cowell (Seller_ID = 2005) decides to change the membership from normal to premium, with payment made by credit card.**

Before update:

```
SELECT S.SELLER_ID, S.NAME, S.ACCOUNT_TYPE, S.MEMBERSHIP_FEE, S.PAYMENT_MODE

FROM SELLER S

WHERE S.SELLER_ID = 2004
```

	SELLER_ID	NAME	ACCOUNT_TYPE	MEMBERSHIP_FEE	PAYMENT_MODE
1	2004	Sara Cowell	Normal	0	NULL

Update:

```
UPDATE SELLER

SET TYPE = 'PREMIUM', MEMBERSHIP_FEE = 200,

EXPIRATION_DATE = '2017-11-22 14:23:54', PAYMENT_MODE = 'CC', PAYMENT_DATE = '2016-11-22 14:23:54'

WHERE SELLER_ID = 2004
```

Messages

```
(1 row(s) affected)
```

After update:

```
SELECT S.SELLER_ID, S.NAME, S.ACCOUNT_TYPE, S.MEMBERSHIP_FEE, S.PAYMENT_MODE

FROM SELLER S

WHERE S.SELLER_ID = 2004
```

Results Messages

	SELLER_ID	NAME	ACCOUNT_TYPE	MEMBERSHIP_FEE	PAYMENT_MODE
1	2004	Sara Cowell	PREMIUM	200	CC

17) **Sellers want to update their membership whose accounts have expired**

```
UPDATE SELLER

SET MEMBERSHIP_FEE= (MEMBERSHIP_FEE +200EXPIRATION_DATE) = DATEADD(YEAR, 1,EXPIRATION_DATE)

FROM SELLER S

WHERE SELLER_ID IN

(SELECT S1.SELLER_ID
```

```
FROM SELLER S1

WHERE S1.EXPIRATION_DATE<GETDATE())
```

Messages

```
(2 row(s) affected)
```

18) Which sellers have renewed their accounts more than 1 time

```
SELECT SELLER_ID, S.NAME

FROM SELLER S

WHERE DATEDIFF(YEAR, S.CREATION_DATE, S.EXPIRATION_DATE) >  1

AND S.ACCOUNT_TYPE = 'Premium'
```

	SELLER_ID	NAME
1	2001	Garrett D. Green
2	2003	Bruce Jhonson

19) How many listings have the maximum success rate criteria via social media

```
SELECT L.LISTING_ID, L.PLATFORM AS 'LISTING_PLATFORM', P.PLATFORM AS
'PURCHASE_PLATFORM'

FROM LISTING L JOIN PURCHASE P

ON L.LISTING_ID = P.LISTING_ID

WHERE L.PLATFORM = P.PLATFORM

OR L.PLATFORM = 'BOTH'

AND P.PLATFORM IN ('FACEBOOK','TWITTER')
```

100 %

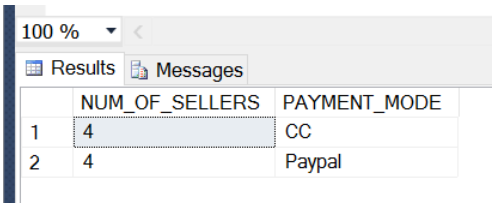
Results | Messages

	LISTING_ID	LISTING_PLATFORM	PURCHASE_PLATFORM
1	4001	BOTH	FACEBOOK
2	4003	BOTH	TWITTER
3	4010	TWITTER	TWITTER
4	4008	FACEBOOK	FACEBOOK
5	4006	FACEBOOK	FACEBOOK
6	4007	FACEBOOK	FACEBOOK
7	4015	FACEBOOK	FACEBOOK

19.) Find the number of premium sellers sorted by their payment mode(CC/Paypal)

```
SELECT COUNT(SELLER_ID) AS 'NUM_OF_SELLERS', S.PAYMENT_MODE
```

```
FROM SELLER S

WHERE S.ACCOUNT_TYPE = 'Premium'

GROUP BY S.PAYMENT_MODE
```

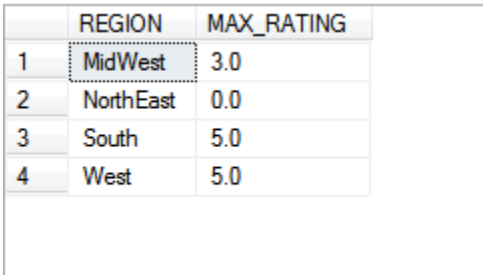


100 % | Results | Messages

	NUM_OF_SELLERS	PAYMENT_MODE
1	4	CC
2	4	Paypal

20) What is the maximum average rating per region?

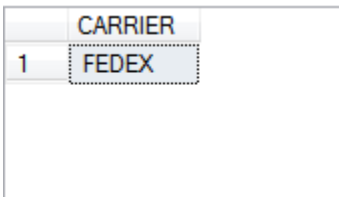
```
SELECT S.REGION, MAX(R.AVG_RATING) AS 'MAX_RATING'

FROM SELLER S JOIN RATING R

ON S.SELLER_ID = R.SELLER_ID

GROUP BY S.REGION
```



	REGION	MAX_RATING
1	MidWest	3.0
2	NorthEast	0.0
3	South	5.0
4	West	5.0

21) Find Carrier(s) with the most number of packages shipped

```
SELECT S.CARRIER FROM SHIPPING S

GROUP BY S.CARRIER

HAVING COUNT(*) >= ALL

(SELECT COUNT(*)

FROM SHIPPING S1

GROUP BY S1.CARRIER)
```

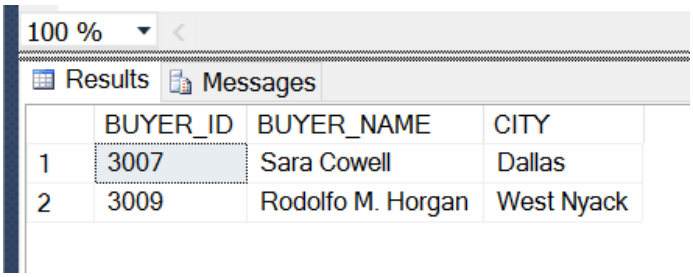


	CARRIER
1	FEDEX

Admin

22) Which users are both buyers and sellers on Sell-It?

```
SELECT B.BUYER_ID, B.NAME AS BUYER_NAME, B.CITY

FROM BUYER B JOIN SELLER S

ON B.EMAIL_ID = S.EMAIL_ID
```



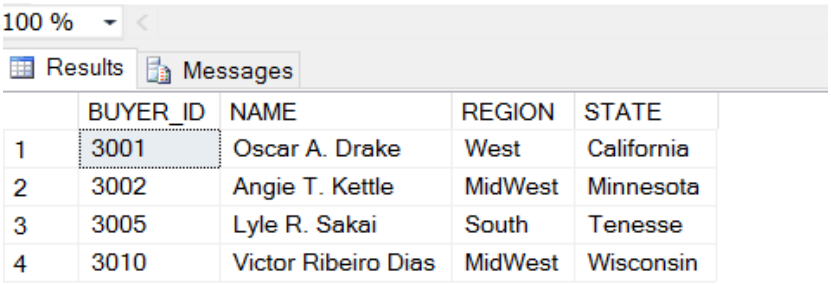
100 %

Results | Messages

	BUYER_ID	BUYER_NAME	CITY
1	3007	Sara Cowell	Dallas
2	3009	Rodolfo M. Horgan	West Nyack

23) Admin runs a query to check the buyers who have made purchases more than once

```
SELECT B.BUYER_ID, B.NAME, B.REGION, B.STATE

FROM BUYER B

WHERE B.BUYER_ID IN

(SELECT P.BUYER_ID

FROM PURCHASE P

GROUP BY P.BUYER_ID

HAVING COUNT(P.LISTING_ID)>1)
```



100 %

Results | Messages

	BUYER_ID	NAME	REGION	STATE
1	3001	Oscar A. Drake	West	California
2	3002	Angie T. Kettle	MidWest	Minnesota
3	3005	Lyle R. Sakai	South	Tenesse
4	3010	Victor Ribeiro Dias	MidWest	Wisconsin

24) Admin wants to check the sellers who have sold listings more than once on Sell-It

```
SELECT S.SELLER_ID, S.NAME, S.REGION, S.STATE

FROM SELLER S

WHERE S.SELLER_ID IN

(SELECT L.SELLER_ID

FROM PURCHASE P LEFT JOIN LISTING L ON P.LISTING_ID = L.LISTING_ID

GROUP BY L.SELLER_ID
```

HAVING COUNT(P.TRANSACTION_ID)>1)

	SELLER_ID	NAME	REGION	STATE
1	2001	Garrett D. Green	West	California
2	2008	Cindy C. Ross	South	Texas
3	2010	Leon M. Vick	Midwest	Ohio

25) Which listing successful transactions have been received by the buyer?

SELECT P.TRANSACTION_ID, B.BUYER_ID, B.NAME, B.EMAIL_ID, P.LISTING_ID

FROM PURCHASE P LEFT JOIN BUYER B

ON P.BUYER_ID = B.BUYER_ID

JOIN LISTING L ON L.LISTING_ID = P.LISTING_ID

WHERE P.RECEIVE_STATUS = 'RECEIVED'

Results Messages

	TRANSACTION_ID	BUYER_ID	NAME	LISTING_ID
1	5001	3001	Oscar A. Drake	4001
2	5003	3001	Oscar A. Drake	4003
3	5005	3002	Angie T. Kettle	4005
4	5010	3002	Angie T. Kettle	4010
5	5002	3003	Elizabeth N. Posey	4002
6	5004	3004	Joann J. Cook	4004
7	5008	3005	Lyle R. Sakai	4008
8	5009	3005	Lyle R. Sakai	4009
9	5006	3010	Victor Ribeiro Dias	4006
10	5007	3010	Victor Ribeiro Dias	4007
11	5011	3011	Toby Morgan	4015

26) Admin wants to check the listings which are still open on Sell-It to track them to closure

SELECT L.LISTING_ID, L.LISTING_DATE_CREATED

FROM LISTING L

WHERE L.LISTING_STATUS = 'OPEN'

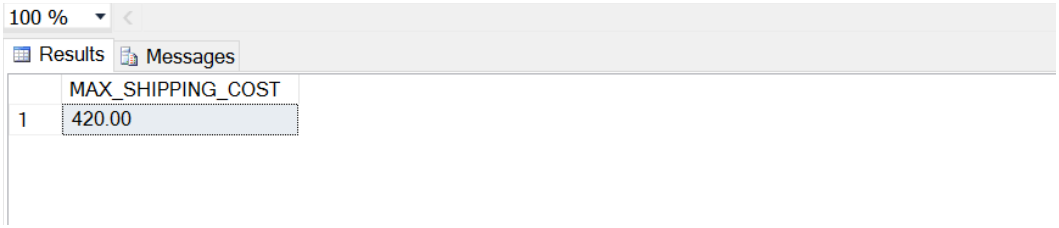
100 %

Results Messages

	LISTING_ID	LISTING_DATE_CREATED
1	4011	2016-11-01 18:53:21.000
2	4012	2016-11-01 23:53:21.000
3	4013	2016-11-06 18:53:21.000
4	4014	2016-11-17 15:32:57.000

27) What is the maximum amount paid by the shipments per carrier?

```
SELECT MAX(T1.CUMULATIVE_SHIPPING_COST) AS MAX_SHIPPING_COST
FROM
(SELECT S.CARRIER, SUM(S.SHIPPING_COST) AS 'CUMULATIVE_SHIPPING_COST'
FROM SHIPPING S
WHERE S.STATUS = 'DELIVERED'
GROUP BY S.CARRIER)T1
```

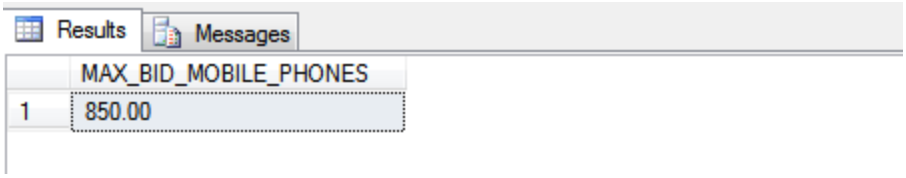


28) What is the name of that carrier?

```
SELECT S.CARRIER
FROM SHIPPING S
WHERE S.STATUS = 'DELIVERED'
GROUP BY S.CARRIER
HAVING SUM(S.SHIPPING_COST) IN
(SELECT MAX(T1.CUMULATIVE_SHIPPING_COST) AS MAX_SHIPPING_COST
FROM
(SELECT S.CARRIER, SUM(S.SHIPPING_COST) AS 'CUMULATIVE_SHIPPING_COST'
FROM SHIPPING S
WHERE S.STATUS = 'DELIVERED'
GROUP BY S.CARRIER)T1)
```

29) Admin wants to check the maximum amount on which the mobile phone was sold on the website ever

```
SELECT MAX(B.BID_AMOUNT) AS 'MAX_BID_MOBILE_PHONES'
FROM LISTING L JOIN BID B ON L.LISTING_ID = B.LISTING_ID
WHERE L.SUBCATEGORY_ID = 8001
```



Results | Messages

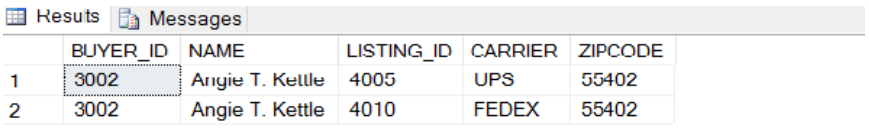
	MAX_BID_MOBILE_PHONES
1	850.00

30) Admin Noel Glover wants to check those Buyers whose listings are shipped via different or same carrier in the Zipcode 55402

SELECT B.BUYER_ID, B.NAME, S.LISTING_ID, S.CARRIER, S.ZIPCODE

FROM BUYER B JOIN SHIPPING S ON B.ZIPCODE = S.ZIPCODE

AND S.ZIPCODE = 55402



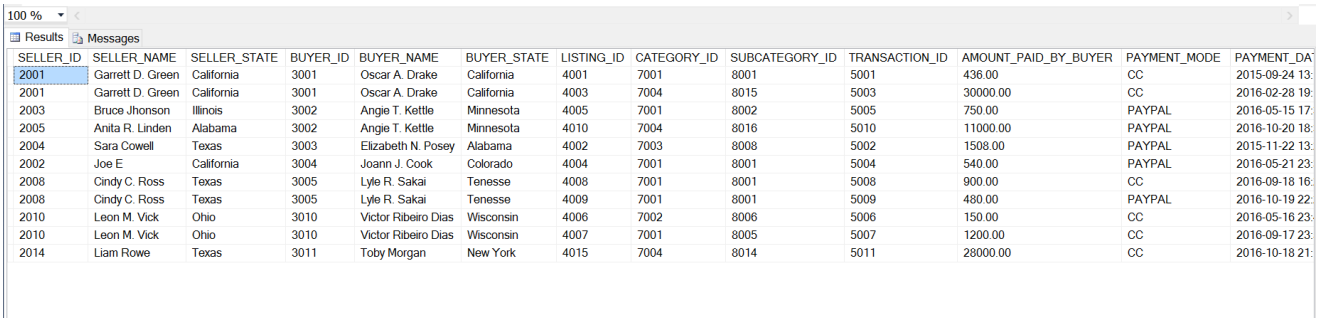
Results | Messages

	BUYER_ID	NAME	LISTING_ID	CARRIER	ZIPCODE
1	3002	Angie T. Kettle	4005	UPS	55402
2	3002	Angie T. Kettle	4010	FEDEX	55402

31) Admin Mitchell Archam wants to see the full listing of the transactions and details of the items sold

SELECT *

FROM CORRESPONDING_BUYER_AND_SELLER



100 % | Results | Messages

SELLER_ID	SELLER_NAME	SELLER_STATE	BUYER_ID	BUYER_NAME	BUYER_STATE	LISTING_ID	CATEGORY_ID	SUBCATEGORY_ID	TRANSACTION_ID	AMOUNT_PAID_BY_BUYER	PAYMENT_MODE	PAYMENT_DA
2001	Garrett D. Green	California	3001	Oscar A. Drake	California	4001	7001	8001	5001	436.00	CC	2015-09-24 13:
2001	Garrett D. Green	California	3001	Oscar A. Drake	California	4003	7004	8015	5003	30000.00	CC	2016-02-28 19:
2003	Bruce Jhonson	Illinois	3002	Angie T. Kettle	Minnesota	4005	7001	8002	5005	750.00	PAYPAL	2016-05-15 17:
2005	Anita R. Linden	Alabama	3002	Angie T. Kettle	Minnesota	4010	7004	8016	5010	11000.00	PAYPAL	2016-10-20 18:
2004	Sara Cowell	Texas	3003	Elizabeth N. Posey	Alabama	4002	7003	8008	5002	1508.00	PAYPAL	2015-11-22 13:
2002	Joe E	California	3004	Joann J. Cook	Colorado	4004	7001	8001	5004	540.00	PAYPAL	2016-05-21 23:
2008	Cindy C. Ross	Texas	3005	Lyle R. Sakai	Tenesse	4008	7001	8001	5008	900.00	CC	2016-09-18 16:
2008	Cindy C. Ross	Texas	3005	Lyle R. Sakai	Tenesse	4009	7001	8001	5009	480.00	PAYPAL	2016-10-19 22:
2010	Leon M. Vick	Ohio	3010	Victor Ribeiro Dias	Wisconsin	4006	7002	8006	5006	150.00	CC	2016-05-16 23:
2010	Leon M. Vick	Ohio	3010	Victor Ribeiro Dias	Wisconsin	4007	7001	8005	5007	1200.00	CC	2016-09-17 23:
2014	Liam Rowe	Texas	3011	Toby Morgan	New York	4015	7004	8014	5011	28000.00	CC	2016-10-18 21:

32) When admin runs the query with the third parties for the shipment status, he notices that transaction_id 1671549 AND 525845 have been delivered. He decides to change the status to delivered and get corresponding transaction_id, listing_id, seller_id and buyer_id

SELECT P.TRANSACTION_ID, P.LISTING_ID, L.SELLER_ID, P.BUYER_ID, S.STATUS

FROM SHIPPING S LEFT JOIN PURCHASE P

ON S.LISTING_ID = P.LISTING_ID

LEFT JOIN LISTING L ON L.LISTING_ID = P.LISTING_ID

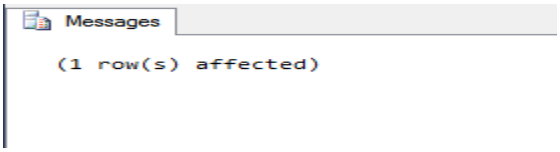
WHERE S.TRACKING_ID = 1671549

	TRANSACTION_ID	LISTING_ID	SELLER_ID	BUYER_ID	STATUS
1	5002	4002	2004	3003	TRANSIT

UPDATE SHIPPING

SET STATUS = ‘DELIVERED’

WHERE TRACKING_ID = 1671549



SELECT P.TRANSACTION_ID, P.LISTING_ID, L.SELLER_ID, P.BUYER_ID, S.STATUS

FROM SHIPPING S LEFT JOIN PURCHASE P

ON S.LISTING_ID = P.LISTING_ID

LEFT JOIN LISTING L ON L.LISTING_ID = P.LISTING_ID

WHERE S.TRACKING_ID = 1671549

	TRANSACTION_ID	LISTING_ID	SELLER_ID	BUYER_ID	STATUS
1	5002	4002	2004	3003	DELIVERED

SELECT P.TRANSACTION_ID, P.LISTING_ID, L.SELLER_ID, P.BUYER_ID, S.STATUS

FROM SHIPPING S LEFT JOIN PURCHASE P ON S.LISTING_ID = P.LISTING_ID LEFT JOIN LISTING L ON L.LISTING_ID = P.LISTING_ID

WHERE S.TRACKING_ID = 5258453

	TRANSACTION_ID	LISTING_ID	SELLER_ID	BUYER_ID	STATUS
1	5010	4010	2005	3002	TRANSIT

UPDATE SHIPPING

SET STATUS = ‘DELIVERED’

WHERE TRACKING_ID = 5258453

SELECT P.TRANSACTION_ID, P.LISTING_ID, L.SELLER_ID, P.BUYER_ID, S.STATUS

FROM SHIPPING S LEFT JOIN PURCHASE P ON S.LISTING_ID = P.LISTING_ID LEFT JOIN LISTING L ON L.LISTING_ID = P.LISTING_ID

WHERE S.TRACKING_ID = 5258453

	TRANSACTION_ID	LISTING_ID	SELLER_ID	BUYER_ID	STATUS
1	5010	4010	2005	3002	DELIVERED

33) Pull out the shipping details of all the shipments

SELECT * FROM SHIPPING_DETAILS

100 %

Results | Messages

	CARRIER	BUYER_ID	BUYER_ZIPCODE	BUYER_REGION	SHIPPING_DATE	SHIPPING_COST	SELLER_ID	SELLING_ZIPCODE	SELLER_REGION
1	FEDEX	3001	97852	West	2015-09-01 14:20:12.000	15.00	2001	91733	West
2	DHL	3001	97852	West	2016-03-08 22:50:32.000	300.00	2001	91733	West
3	UPS	3002	55402	MidWest	2016-05-11 04:04:21.000	50.00	2003	60601	MidWest
4	FEDEX	3002	55402	MidWest	2016-10-15 17:34:17.000	500.00	2005	36201	South
5	UPS	3003	35203	South	2016-01-20 23:44:44.000	100.00	2004	94086	South
6	FEDEX	3005	37210	South	2016-09-23 09:27:25.000	50.00	2008	76024	South
7	FEDEX	3005	37210	South	2016-10-13 21:22:22.000	30.00	2008	76024	South
8	FEDEX	3010	54725	MidWest	2016-05-12 01:30:04.000	10.00	2010	52175	Midwest
9	DHL	3010	54725	MidWest	2016-09-20 12:00:33.000	120.00	2010	52175	Midwest

Business User

34) Find all the Buyers who have bought from category Books but not bought from category called Furniture.

```
SELECT B.BUYER_ID

FROM BUYER B JOIN PURCHASE P ON B.BUYER_ID=P.BUYER_ID

JOIN LISTING L

ON L.LISTING_ID=P.LISTING_ID

JOIN CATEGORY C

ON C.CATEGORY_ID = L.CATEGORY_ID

WHERE C.DESCRIPTION LIKE 'BOOKS%'

AND B.BUYER_ID NOT IN

(SELECT B1.BUYER_ID

FROM BUYER B1 INNER JOIN PURCHASE P1

ON B1.BUYER_ID=P1.BUYER_ID

INNER JOIN LISTING L1

ON L1.LISTING_ID=P1.LISTING_ID

INNER JOIN CATEGORY C1

ON C1.CATEGORY_ID = L1.CATEGORY_ID

WHERE C.DESCRIPTION LIKE 'FURNITURE%')
```

35) Find region who have Buyers who have bought maximum number of times.

```
SELECT B.REGION,COUNT(B.BUYER_ID) AS 'MAX NO BUYERS'
FROM BUYER B INNER JOIN PURCHASE P ON B.BUYER_ID = P.BUYER_ID
GROUP BY B.REGION
HAVING COUNT(B.BUYER_ID) IN
(SELECT MAX (T1.NO_OF_BUYERS)
FROM
(SELECT B.REGION, COUNT(B.BUYER_ID) AS 'NO_OF_BUYERS'
FROM BUYER B INNER JOIN PURCHASE P ON B.BUYER_ID = P.BUYER_ID
GROUP BY B.REGION)T1)
```

36) Which region has sellers who have sold maximum number of times

```
SELECT S.REGION
FROM LISTING L JOIN PURCHASE P
ON L.LISTING_ID = P.LISTING_ID
JOIN SELLER S
ON L.SELLER_ID = S.SELLER_ID
GROUP BY S.REGION
HAVING COUNT(S.SELLER_ID) IN
(SELECT MAX (T1.NO_OF_SELLERS)
FROM
(SELECT S.REGION, COUNT(S.SELLER_ID) AS 'NO_OF_SELLERS'
FROM LISTING L JOIN PURCHASE P
ON L.LISTING_ID = P.LISTING_ID
JOIN SELLER S
ON L.SELLER_ID = S.SELLER_ID
```

GROUP BY S.REGION) T1)

37) Calculate the total payments received via the Paypal platform

SELECT SUM(T1.FROM_MEMBERSHIP+T2.FROM_LISTING) AS 'TOTAL_AMOUNT_THROUGH_PAYPAL'

FROM

(SELECT SUM(S.MEMBERSHIP_FEE) AS 'FROM_MEMBERSHIP'

FROM SELLER S

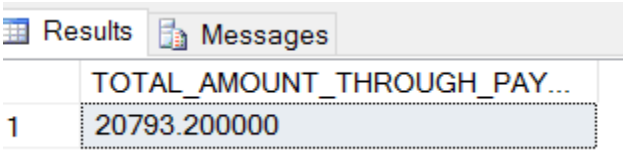
WHERE ACCOUNT_TYPE = 'PREMIUM' AND PAYMENT_MODE='PAYPAL') T1,

(SELECT SUM(B.BID_AMOUNT*S.CUTOFF/100) AS 'FROM_LISTING'

FROM LISTING L, PURCHASE P, BID B, SUBCATEGORY S

WHERE L.LISTING_ID=P.LISTING_ID

AND P.BID_ID=B.BID_ID AND S.SUBCATEGORY_ID=L.SUBCATEGORY_ID) T2



38) Find the most popular category which is listed by the Seller

SELECT C.CATEGORY_ID,C.DESCRIPTION

FROM CATEGORY C JOIN LISTING L ON C.CATEGORY_ID=L.CATEGORY_ID

GROUP BY C.CATEGORY_ID,C.DESCRIPTION

HAVING COUNT(L.LISTING_ID) IN

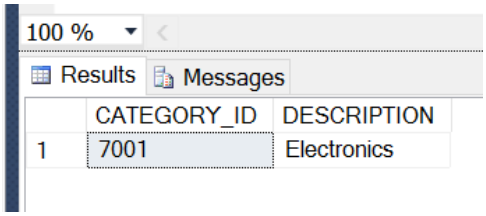
(SELECT MAX(T1.NO_OF_LISTING)

FROM

(SELECT COUNT (L1.LISTING_ID) AS 'NO_OF_LISTING'

FROM LISTING L1

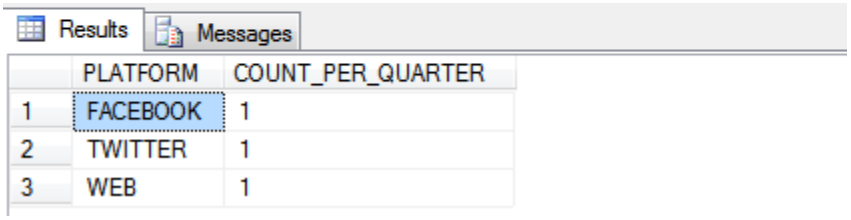
GROUP BY L1.CATEGORY_ID)T1)



	CATEGORY_ID	DESCRIPTION
1	7001	Electronics

39) Business user Robert Pattinson want to do the comparative analysis of the Social media Platform 'Facebook' to record its progress for 2016 Q4

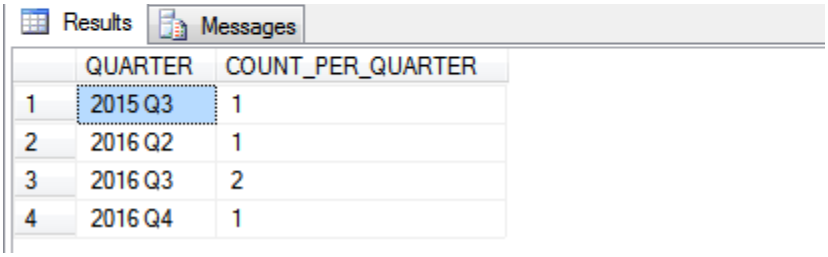
```
SELECT PLATFORM,COUNT(*) AS 'COUNT_PER_QUARTER'

FROM PURCHASE

WHERE QUARTER ='2016 Q4'

GROUP BY PLATFORM
```



	PLATFORM	COUNT_PER_QUARTER
1	FACEBOOK	1
2	TWITTER	1
3	WEB	1

40) Business user Thomas Sachtel want to do the comparative analysis of the Social media Platform 'Facebook' to record its progress in the past quarter wise

```
SELECT QUARTER,  COUNT(*) AS 'COUNT_PER_QUARTER'

FROM PURCHASE

WHERE PLATFORM ='FACEBOOK'

GROUP BY QUARTER
```



	QUARTER	COUNT_PER_QUARTER
1	2015 Q3	1
2	2016 Q2	1
3	2016 Q3	2
4	2016 Q4	1

41) Robert Pattinson is the founder and CEO of Sell-It and wants to check which employees have been in the organization for more than 1 year and eligible for promotion

```
SELECT E.EMPLOYEE_ID, E.NAME, E.EMAIL_ID, E.EMPLOYEE_TYPE

FROM EMPLOYEE E

WHERE DATEDIFF(YEAR,E.ACCT_CREATION_DATE,GETDATE()) =1
```

100 %

Results | Messages

	EMPLOYEE_ID	NAME	EMAIL_ID	EMPLOYEE_TYPE
1	1001	Robert Pattinson	rpattinson@sellit.com	BU
2	1002	Thomas Satchel	tsatchel@sellit.com	BU
3	1003	Amy Schumer	aschumer@sellit.com	Admin

8. INDEXES

1) CREATE UNIQUE INDEX i_SELLER_ID on SELLER(SELLER_ID)

100 %

Messages

```
Command(s) completed successfully.
```

2) CREATE UNIQUE INDEX i_BUYER_ID on BUYER(BUYER_ID)

100 %

Messages

```
Command(s) completed successfully.
```

3) CREATE UNIQUE INDEX i_LISTING_ID on LISTING(LISTING_ID)

100 %

Messages

```
Command(s) completed successfully.
```

4) CREATE UNIQUE INDEX i_BID_ID on BID(BID_ID)

100 %

Messages

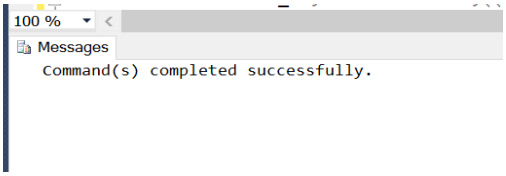
```
Command(s) completed successfully.
```

9. VIEWS

1) View for creating estimated Selling Price for all the listings sold in the past.

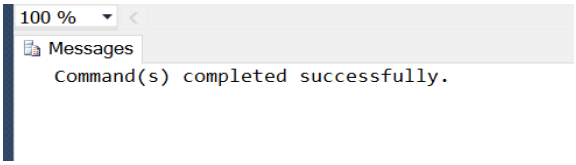
```
CREATE VIEW ESTIMATED_SELLING_PRICE AS

SELECT S.SUBCATEGORY_ID,  S.DESCRIPTION,S.CATEGORY_ID,

((S.AVG_PRICE*(100-S.DEPRECIATION))/100) AS 'ESTIMATED_SELLING_PRICE'
```

```
FROM SUBCATEGORY S JOIN CATEGORY C
ON S.CATEGORY_ID = C.CATEGORY_ID
WHERE S.AVG_PRICE>0
```



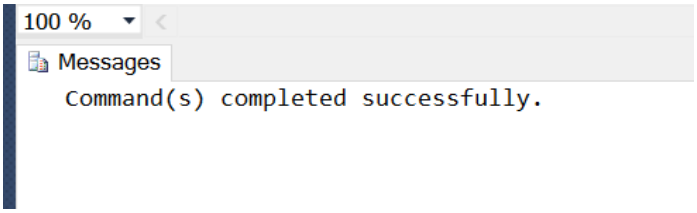
2) View for getting all the buyers corresponding to the sellers from whom they have bought the listing

```
CREATE VIEW CORRESPONDING_BUYER_AND_SELLER AS
SELECT L.SELLER_ID,S.NAME AS SELLER_NAME,S.STATE AS SELLER_STATE,P.BUYER_ID,B.NAME AS BUYER_NAME,B.STATE AS BUYER_STATE,
L.LISTING_ID,L.CATEGORY_ID,SUBCATEGORY_ID,
P.TRANSACTION_ID,AMOUNT_PAID_BY_BUYER,P.PAYMENT_MODE,P.PAYMENT_DATE
FROM LISTING L
JOIN PURCHASE P
ON L.LISTING_ID = P.LISTING_ID
JOIN BUYER B
ON B.BUYER_ID = P.BUYER_ID
JOIN SELLER S
ON S.SELLER_ID = L.SELLER_ID
```



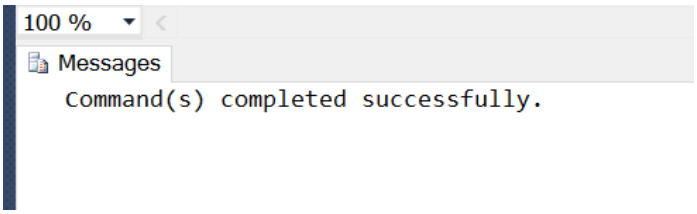
3) View for calculating the average of bids 'NOT' selected for the listing and obtaining the Standard Deviation against the bids actually selected.

```
CREATE VIEW AVG_BIDS_NOT_SELECTED AS
 SELECT B.LISTING_ID, AVG(B.BID_AMOUNT) AS 'AVG_BIDS_NOT_SELECTED'
 FROM BID B
 WHERE BID_STATUS = 'NOT SELECTED'
 GROUP BY B.LISTING_ID
```



4) View for obtaining the shipping details of all the successful purchases

```
CREATE VIEW SHIPPING_DETAILS AS

SELECT L.LISTING_ID,S.CARRIER,P.BUYER_ID,S.ZIPCODE AS BUYER_ZIPCODE,B.REGION AS
BUYER_REGION,S.SHIPPING_DATE,SHIPPING_COST,S1.SELLER_ID,L.SELLING_ZIPCODE,S1.REGION
AS SELLER_REGION

FROM SHIPPING S JOIN LISTING L

ON S.LISTING_ID = L.LISTING_ID

JOIN PURCHASE P

ON L.LISTING_ID = P.LISTING_ID

JOIN SELLER S1 ON L.SELLER_ID = S1.SELLER_ID

JOIN BUYER B ON P.BUYER_ID = B.BUYER_ID

WHERE CARRIER IS NOT NULL
```



10. BUSINESS MATRICES

1) Obtain the revenue generated by listings

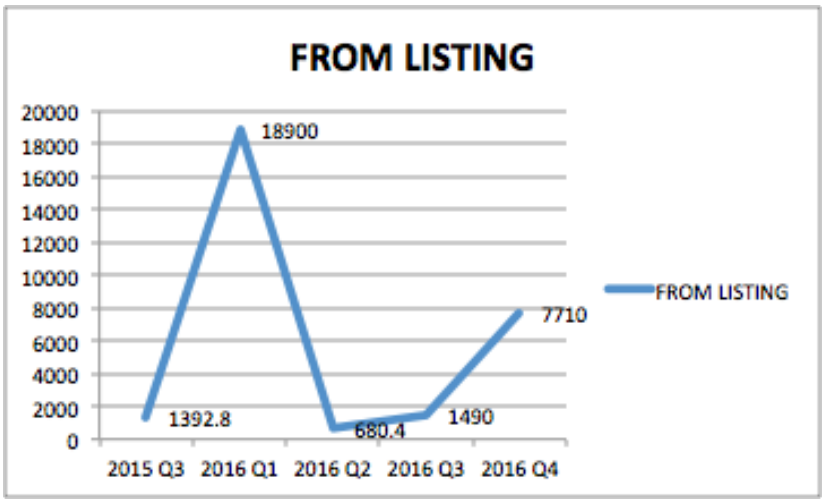
```
SELECT L.LISTING_ID,L.SUBCATEGORY_ID,B.BID_AMOUNT, P.PAYMENT_DATE,
P.AMOUNT_PAID_BY_BUYER,L.AMOUNT_PAID_TO_SELLER,S.CUTOFF,S1.SHIPPING_COST

FROM LISTING L JOIN PURCHASE P

ON L.LISTING_ID = P.LISTING_ID

JOIN SUBCATEGORY S

ON L.SUBCATEGORY_ID = S.SUBCATEGORY_ID

JOIN BID B

ON P.BID_ID = B.BID_ID

JOIN SHIPPING S1

ON S1.LISTING_ID = L.LISTING_ID
```

AND SHIPPING_COST IS NOT NULL

100 %

Results | Messages

	LISTING_ID	SUBCATEGORY_ID	BID_AMOUNT	PAYMENT_DATE	AMOUNT_PAID_BY_BUYER	AMOUNT_PAID_TO_SELLER	CUTOFF	SHIPPING_COST
1	4001	8001	418.00	2015-09-24 13:12:21.000	436.00	351.80	20	15.00
2	4003	8015	27000.00	2016-02-28 19:13:54.000	30000.00	19200.00	30	300.00
3	4005	8002	750.00	2016-05-15 17:32:53.000	750.00	610.00	20	50.00
4	4010	8016	10500.00	2016-10-20 18:32:21.000	11000.00	7850.00	30	500.00
5	4002	8008	1408.00	2015-11-22 13:23:24.000	1508.00	1156.00	25	100.00
6	4008	8001	850.00	2016-09-18 16:21:21.000	900.00	730.00	20	50.00
7	4009	8001	450.00	2016-10-19 22:21:41.000	480.00	390.00	20	30.00
8	4006	8006	140.00	2016-05-16 23:43:12.000	150.00	130.40	14	10.00
9	4007	8005	1080.00	2016-09-17 23:52:11.000	1200.00	930.00	25	120.00



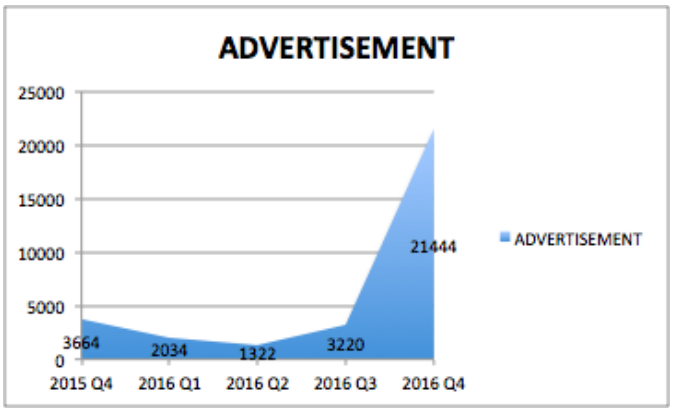
2) Calculate the total advertising revenue per quarter.

```
SELECT T1.QUARTER,(T1.TOTAL_AMT_OBTAINED - T2.TOTAL_AMT_GIVEN) AS
'REVENUE_GENERATED_BY_AD'

FROM

(SELECT SUM(A.AMOUNT) AS 'TOTAL_AMT_OBTAINED', A.QUARTER

FROM ADVERTISEMENT A

WHERE A.AD_TYPE = 'OBTAINED'

GROUP BY A.QUARTER)T1 LEFT OUTER JOIN

(SELECT SUM(A.AMOUNT) AS 'TOTAL_AMT_GIVEN', A.QUARTER

FROM ADVERTISEMENT A

WHERE A.AD_TYPE = 'GIVEN'

GROUP BY A.QUARTER) T2

ON T1.QUARTER = T2.QUARTER
```

100 %

Results | Messages

	QUARTER	REVENUE_GENERATED_BY_AD
1	2015 Q4	3664.00
2	2016 Q1	2034.00
3	2016 Q2	1322.00
4	2016 Q3	3220.00
5	2016 Q4	21444.00



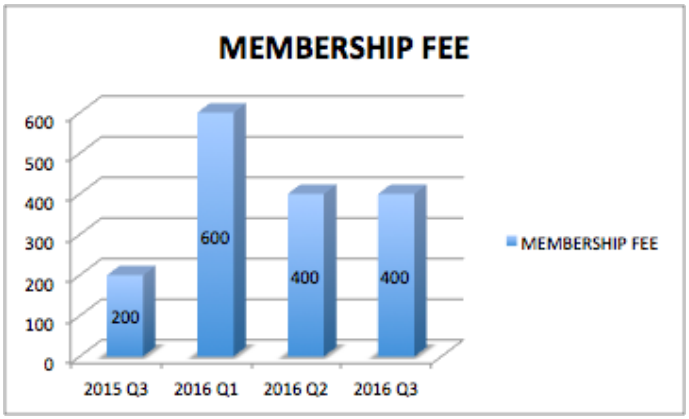
3) Calculate the total Membership revenue per Quarter.

```
SELECT SELLER_ID, MEMBERSHIP_FEE, PAYMENT_DATE

FROM SELLER

WHERE ACCOUNT_TYPE = 'PREMIUM'
```

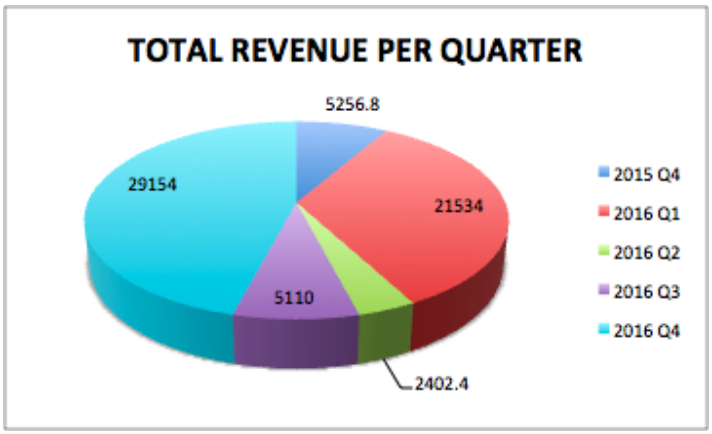
100 %

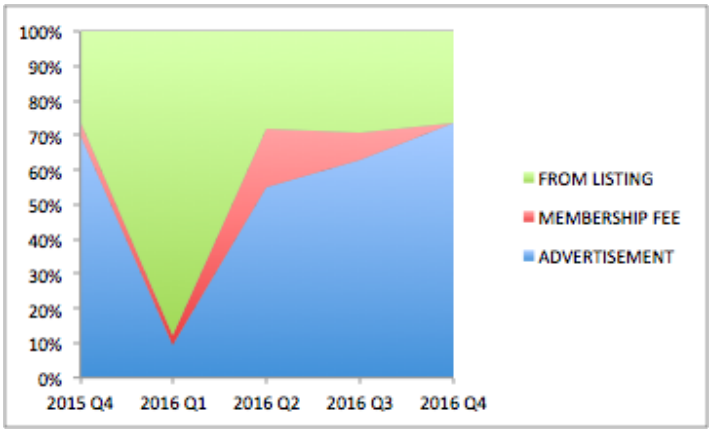
Results | Messages

	SELLER_ID	MEMBERSHIP_FEE	PAYMENT_DATE
1	2001	200	2015-10-24 07:00:00.000
2	2003	200	2016-01-26 08:15:20.000
3	2005	200	2016-02-12 23:15:20.000
4	2006	200	2016-02-15 08:15:20.000
5	2009	200	2016-04-25 21:25:25.000
6	2010	200	2016-04-27 21:25:25.000
7	2011	200	2016-07-27 21:25:25.000
8	2013	200	2016-09-27 21:25:25.000



COMBINED REVENUE:





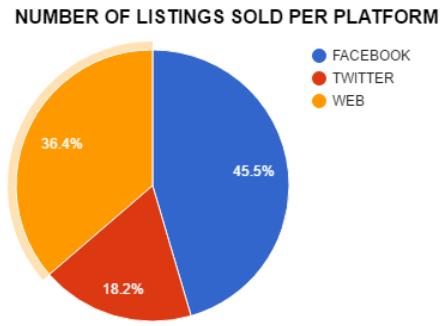
4) Find the number of Listings sold per platform

```
SELECT COUNT(LISTING_ID) AS 'NUM_OF_LISTINGS' , P.PLATFORM
FROM PURCHASE P
WHERE P.PAYMENT_STATUS = 'PAID'
```

GROUP BY P.PLATFORM

Results | Messages

	NUM_OF_LISTIN...	PLATFORM
1	5	FACEBOOK
2	2	TWITTER
3	4	WEB



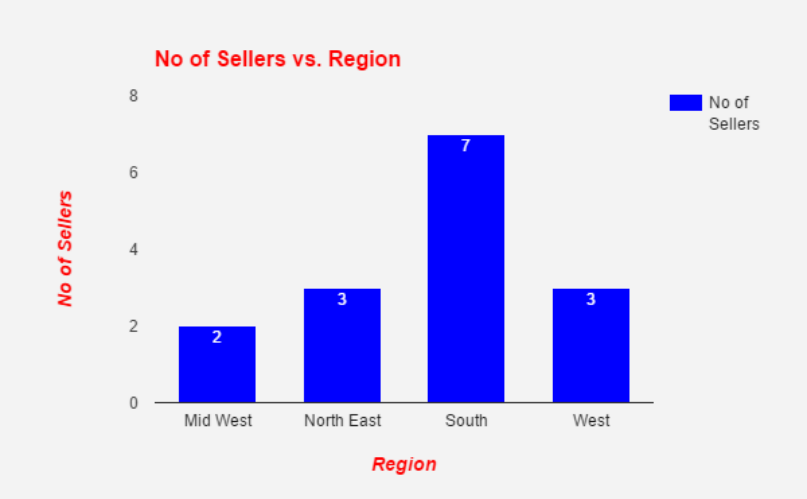
5) Find the number of Sellers sorted by Region

SELECT COUNT(SELLER_ID) AS 'Num_OF_SELLERS_PER_REGION', S.REGION

FROM SELLER SGROUP BY S.REGION

Results | Messages

	Num_OF_SELLERS_PER_REGION	REGION
1	2	MidWest
2	3	NorthEast
3	7	South
4	3	West



6) Find the number of Buyers sorted by Region.

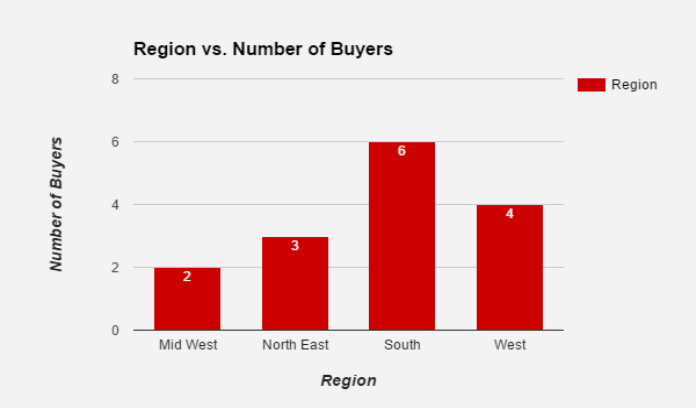
SELECT COUNT(BUYER_ID) AS 'Num_OF_BUYERS_PER_REGION', B.REGION

FROM BUYER B

GROUP BY B.REGION

Results | Messages

	Num_OF_BUYERS_PER_REGI...	REGION
1	2	MidWest
2	3	NorthEast
3	6	South
4	4	West



7) Rachel Green wants to find out the standard deviation between mean Bid amounts and the Bid amount actually selected

SELECT T2.LISTING_ID,(T2.BID_AMOUNT-A.AVG_BIDS_NOT_SELECTED) AS 'STANDARD_DEVIATION'

FROM

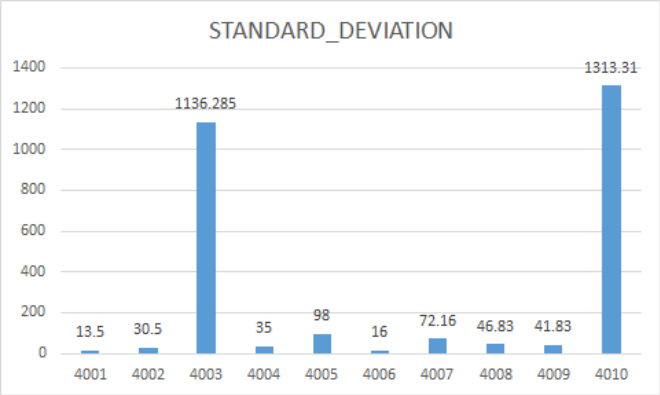
(SELECT LISTING_ID,BID_AMOUNT

```
FROM BID
WHERE BID_STATUS = 'SELECTED')T2
JOIN AVG_BIDS_NOT_SELECTED A
ON A.LISTING_ID = T2.LISTING_ID
```

100 %

Results | Messages

	LISTING_ID	STANDARD_DEVIATION
1	4001	13.500000
2	4002	30.500000
3	4003	1136.285715
4	4004	35.000000
5	4005	98.000000
6	4006	16.000000
7	4007	72.166667
8	4008	46.333334
9	4009	41.833334
10	4010	1313.333334



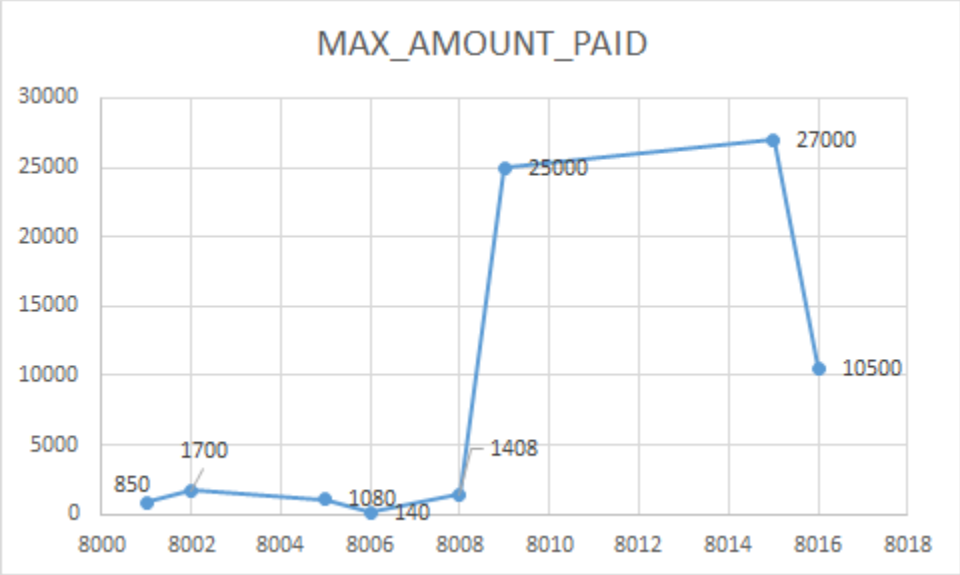
8) What is the maximum price per Subcategory on which it is sold?

```
SELECT S.SUBCATEGORY_ID,MAX(BID_AMOUNT) AS 'MAX_AMOUNT_PAID'
FROM SUBCATEGORY S JOIN LISTING L
ON L.SUBCATEGORY_ID = S.SUBCATEGORY_ID
JOIN BID B
ON L.LISTING_ID = B.LISTING_ID
GROUP BY S.SUBCATEGORY_ID
```

100 %

Results | Messages

	SUBCATEGORY_ID	MAX_AMOUNT_PAID
1	8001	850.00
2	8002	1700.00
3	8005	1080.00
4	8006	140.00
5	8008	1408.00
6	8009	25000.00
7	8015	27000.00
8	8016	10500.00



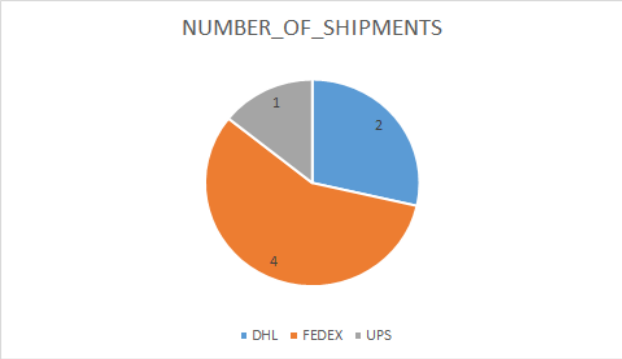
9) Calculate the successful number of Shipments per carrier

```
SELECT S.CARRIER, COUNT(S.LISTING_ID) AS 'NUMBER_OF_SHIPMENTS'
FROM SHIPPING S
WHERE S.STATUS = 'DELIVERED'
GROUP BY S.CARRIER
```

100 %

Results | Messages

	CARRIER	NUMBER_OF_SHIPMENTS
1	DHL	2
2	FEDEX	4
3	UPS	1

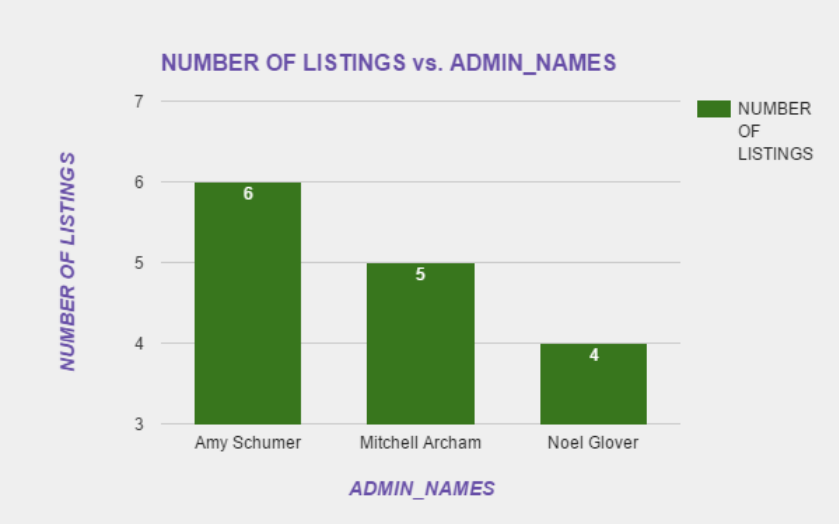


10) Which admin has handled the maximum number of Listings so that he/she can be given the best performer award?

```
SELECT E.NAME, T1.HANDLED_BY, MAX(T1.NO_OF_LISTINGS_HANDLED) AS 'MAX_LISTINGS'

FROM

(SELECT L.HANDLED_BY,COUNT(LISTING_ID) AS 'NO_OF_LISTINGS_HANDLED'

FROM LISTING L

GROUP BY HANDLED_BY) T1 JOIN EMPLOYEE E

ON E.EMPLOYEE_ID = T1. HANDLED_BY

GROUP BY T1.HANDLED_BY,E.NAME
```

Results | Messages

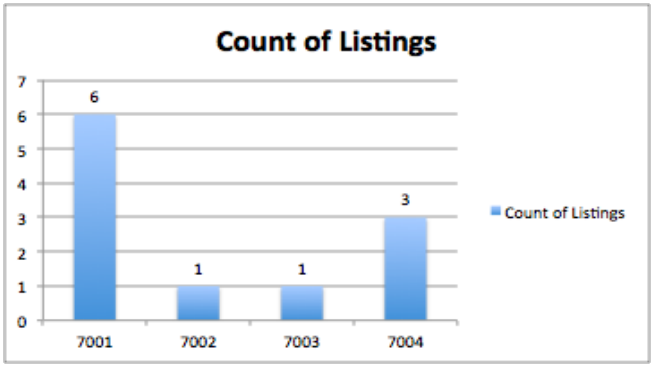
	NAME	HANDLED_BY	MAX_LISTINGS
1	Amy Schumer	1003	6
2	Mitchell Archam	1004	5
3	Noel Glover	1006	4



11) Find the total Listings purchased on the website per Category

```
SELECT L.CATEGORY_ID, COUNT(L.CATEGORY_ID) AS 'COUNT_MAX'

FROM PURCHASE P INNER JOIN LISTING L ON P.LISTING_ID = L.LISTING_ID

GROUP BY L.CATEGORY_ID
```

	CATEGORY_ID	COUNT_MAX
1	7001	6
2	7002	1
3	7003	1
4	7004	3



11.PROJECT SUMMARY

1) Summarize the experience with the project

It was a very good learning experience doing this project. We got the opportunity to learn the nuances of database management systems. This didn't only involve the technical expertise but also inculcated the skills to work in a team. So starting from forming a team, choosing the e-commerce application, UML approval etc, all the activities were very interesting and full of learning. We have been interacting with e-commerce websites like Amazon, eBay, Wal-Mart etc. in our day to day lives, but for the first time got to understand the operations of these applications. We applied our database technical skills for example SQL queries, joins, implementing temporal data, concepts, triggers and views which are required for the making of back-end of these e-commerce applications.

2) What was the hardest part of the project

This project was full of challenges, as a team of 3 members we did everything together starting from choosing the application and implementing the concept but the most challenging part of the project was finalizing the UML part of the application. Since e-commerce application has a broad scope, it involves several segments of front-end and back-end side of the application. Finalizing the entities, and to de-scope the application for appropriate relationships, cardinalities and most important attributes and keys was challenging. Another challenge was to finalize the test data. As to execute the queries, we need to insert the data logically, it was tedious to think ahead and insert the correct data across tables, which can satisfy all the constraints.

E-commerce application does not only involve users but also involves monetization. So, creating the business metrics was tough. It required us to write long sub queries which involved lots of joins, then

views and triggers. Lastly, the application contains past and present customer records, so timestamp played a vital role in designing application and in fetching the data from tables. It is challenging to manage and fetch data around datetime.

3) **What problems did you run against this project**

While designing the UML we faced many difficulties for mapping relations, cardinalities and to decide attributes maintaining all the constraints and data integrity.

Insertion of data was very challenging and tedious. E-commerce application involves several number of users like Buyers, Sellers in the front-end and the employees at the back-end of the project. As, the business application involves bidding, seller reviews and ratings which can be updated every day. It is difficult to insert the correct data across the attributes which can satisfy all the primary key constraints, managing the null values and most importantly avoiding the redundancy.

4) **How did you solve the problems?**

Team collaboration and teamwork was the key to all the problems.

It was very challenging to decide the entities in UML. We were doing brainstorming to select the final modules and entities out of various combinations of Relations and attributes from which data can be derived in best possible way. First of all, we made the UML on paper and whiteboard to easily visualize the mapping between various entities.

Evaluation by the professor after every phase helped us a lot to think in the perspective of maintaining the data integrity and consistency and to avoid update, deletion anomalies and data redundancy. So, this approach not only helped us to finalize the UML but also gave us an idea to create the test data. After this, we first thought about the functions of all the users, the type of queries that can be formed for the role of customers, employees etc. The overall process took ample of time as we had to manage not only the users but also the monetization part of the project.

5) **If you were to do this project again, what methodology would you use?**

If we do this project again, then first of all we will come up with appropriate UML in less time and with accuracy. We will enumerate all the possible interactions between different users and evaluate every case in such a manner that helps to create our test data and execution of queries smoothly, without having to go back and change even the slightest part of our UML. In this way, we will be able to avoid all kinds of anomalies and data redundancy.

For creating the test data also, we will start with creating only 2 or 3 records in all the tables by solving all possible errors and once we get the idea of how to proceed with creating data, we will create the rest of test data

6) **Suggestions how to refine this project for the next class**

Designing of front end along with back end of the application, this will give us a complete hands on experience to make a project. It would be easy to explain the functionality of the project with the help of user interfaces.