FINAL PROJECT REPORT

RENTAL RESTYLE

DATABASE MANAGEMENT- MSIS 2603

Guided By:

Dr.Shailesh Aggarwal

Submitted By:

Prachi Sharma|Prithviraj Naidu | Sarah Fernandez | Vikita Nayak

Contents

1. Introduction 3
2. Objective 3
3. Scope 4
4. Users of the system 4
5. UML Diagram 5
6. Physical Schema – Database Dictionary 6
7. Queries 10
8. Insertion 16
9. Deletion 16
10. Updation 16
11. Views 17
12. Index 18
13. Triggers 18
14. Business Metrics 19
15. Project Summary 23

1. Introduction

The Rental Restyle is a platform which features different types of dresses and outfits, catering to different occasions and multiple cultural groups for rent for about 8 days. The service enables the customers to rent the outfits according to their needs. We curate designs from the whiz designers, in accordance to the latest trends. Our aim is to deliver high street design, at pocket friendly prices, just for your convenience. The customer can browse the collection based on latest trends, based on their sizes, the kind of occasion they are trying to find the perfect outfit for and also a specific designer.
Doorstep delivery makes these outfits accessible at the customer's convenience. We try to make sure that our dresses come back to us in proper condition because at, The Rental Restyle we firmly believe in utmost customer satisfaction and so the hygiene & condition of the outfits is our top priority.

2. Objective

• Shorter Time Length
o It brings high street fashion online, and hence it makes the process of renting outfits hassle-free with the advent of technology

• Reduced Cost
o It will reduce the cost of buying actual outfits for the customer as they can rent out outfits which they want to wear just one or two times

• Flexibility
o The system aims at adding flexibility to the users as well as the designers. The users can access a myriad of outfits based on their dates, size and function across multiple designer labels

• Provide high quality services affordable
o The Rental Restyle's primary target market is young fashion conscious college goes who run on a tight budget
o The Rental Restyle aims to make high street fashion affordable to people who can't afford fancy designer labels.

• Diversity
o The Rental Restyle brings in a taste of all cultures and multi-national diversity, it's aim being making a traditional Indian outfit accessible to people of all various diverse cultures

• Creation and Expansion of market share
o It enables the designers to expand their market share by catering to a different market segment

3. Scope

For the purpose of this project, Rental Restyle we will only target customers in the state of California. Application is in its nascent stage were we are only targeting bay area customers, eventually we will expand to across state lines. Database has data of past one year from 1 January 2017 till 1 May 2018.

4. Users of the system

1. **Customer**: performs search and rents clothes from application.

- Sign up for application
- Add item to the cart
- Checkout
- Payment
- Place the order
- Request for return
- Feedback
- Contact customer representative

2. **Designer**: Adds new clothes to the application

- Sign up for the application
- Upload their designs
- Update outfit details

3.**Delivery**: Takes care of inventory details and delivery of rented clothes.

- Sign up for the application
- Check notifications for orders
- Update the delivery status

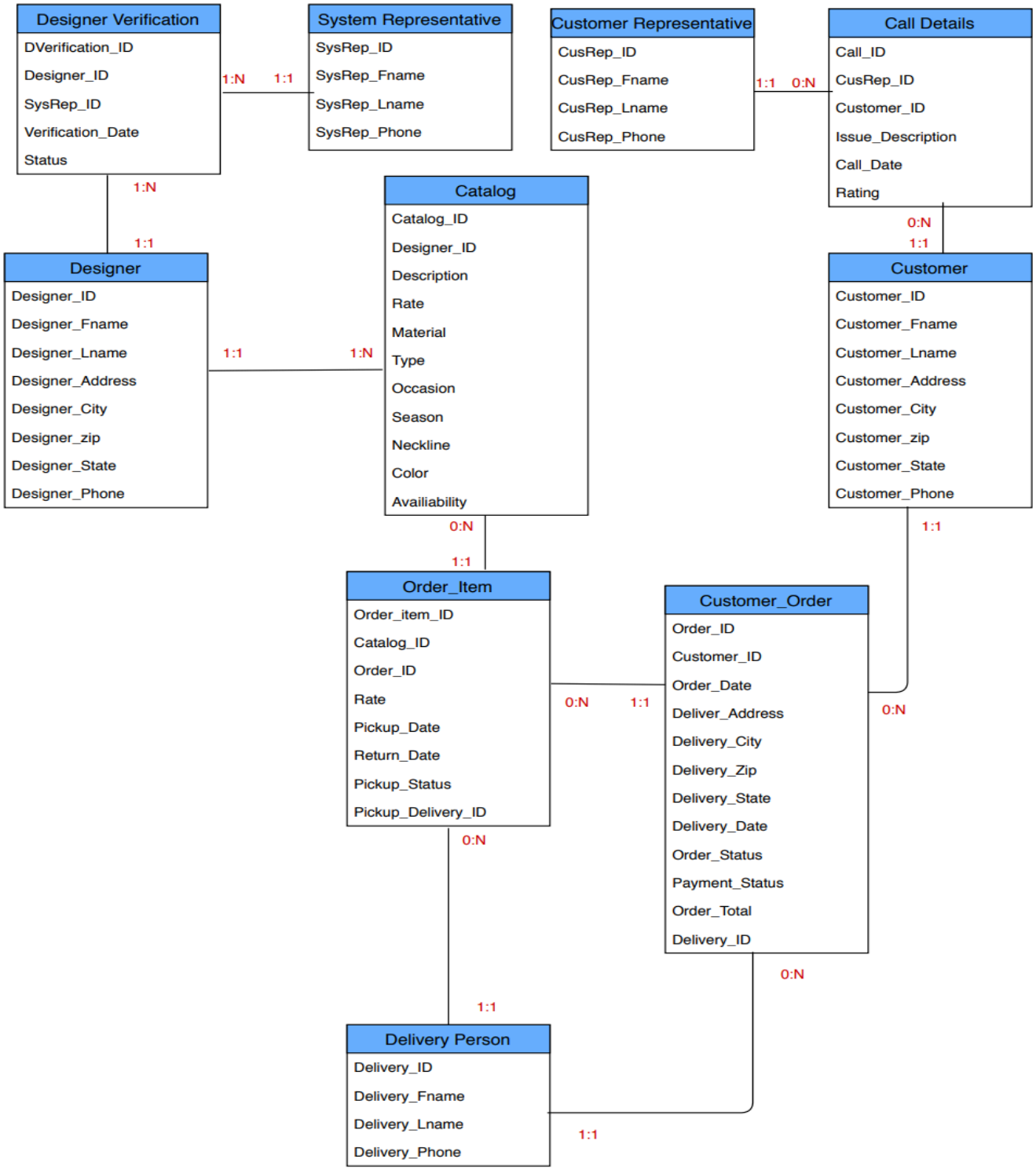
4.**System Representative**: Interacts with customer and provides assistance.

- Verify the designer
- Verify the payment
- Check Order status

5.**Customer Representative**: Verifies the designer, customer details and payment information.

- Interact with customers
- Provide appropriate assistance

5. UML Diagram



6. Physical Schema - Database Dictionary

1. DESIGNER VERIFICATION

Structure:

Column Name	Data Type	Nullability
DVERIFICATION_ID (PRIMARY KEY)	VARCHAR(10)	NOT NULL
DESIGNER_ID (FOREIGN KEY)	VARCHAR(10)	NOT NULL
SYSREP_ID (FOREIGN KEY)	VARCHAR(10)	NOT NULL
VERIFICATION_DATE	DATE	CAN BE NULL
STATUS	VARCHAR(10)	NOT NULL

2. SYSTEM REPRESENTATIVE

Structure:

Column Name	Data Type	Nullability
SYSREP_ID (PRIMARY KEY)	VARCHAR(10)	NOT NULL
SYSREP_FNAME	VARCHAR(20)	NOT NULL
SYSREP_LNAME	VARCHAR(20)	NOT NULL
SYSREP_PHONE	VARCHAR(20)	NOT NULL

3. CUSTOMER REPRESENTATIVE

Structure:

Column Name	Data Type	Nullability
CUSREP_ID (PRIMARY KEY)	VARCHAR(10)	NOT NULL
CUSREP_FNAME	VARCHAR(20)	NOT NULL
CUSREP_LNAME	VARCHAR(20)	NOT NULL
CUSREP_PHONE	VARCHAR(20)	NOT NULL

4. **CALL DETAILS**

Structure:

Column Name	Data Type	Nullability
CALL_ID (PRIMARY KEY)	VARCHAR(10)	NOT NULL
CUSREP_ID (FOREIGN KEY)	VARCHAR(10)	NOT NULL
CUSTOMER_ID (FOREIGN KEY)	VARCHAR(10)	NOT NULL
ISSUE_DESCRIPTION	VARCHAR(255)	NOT NULL
RATING	FLOAT	CAN BE NULL
CALL_DATE	DATETIME	NOT NULL

5. **DESIGNER**

Structure:

Column Name	Data Type	Nullability
DESIGNER_ID (PRIMARY KEY)	VARCHAR(10)	NOT NULL
DESIGNER_FNAME	VARCHAR(20)	NOT NULL
DESIGNER_LNAME	VARCHAR(20)	NOT NULL
DESIGNER_ADDRESS	VARCHAR(255)	NOT NULL
DESIGNER_CITY	VARCHAR(20)	NOT NULL
DESIGNER_ZIP	INT	NOT NULL
DESIGNER_STATE	VARCHAR(20)	NOT NULL
DESIGNER_PHONE	VARCHAR(20)	NOT NULL

6. **CATALOG**

Structure:

Column Name	Data Type	Nullability
CATALOG_ID (PRIMARY_KEY)	VARCHAR(10)	NOT NULL
DESIGNER_ID (FOREIGN KEY)	VARCHAR(10)	NOT NULL

DESCRIPTION	VARCHAR(255)	NOT NULL
RATE	FLOAT	NOT NULL
MATERIAL	VARCHAR(20)	CAN BE NULL
TYPE	VARCHAR(20)	CAN BE NULL
OCCASION	VARCHAR(20)	CAN BE NULL
SEASON	VARCHAR(20)	CAN BE NULL
NECKLINE	VARCHAR(20)	CAN BE NULL
COLOR	VARCHAR(20)	CAN BE NULL
AVAILABILITY	VARCHAR(10)	NOT NULL

7. **CUSTOMER**

Structure:

Column Name	**Data Type**	**Nullability**
CUSTOMER_ID (PRIMARY KEY)	VARCHAR(10)	NOT NULL
CUSTOMER_FNAME	VARCHAR(20)	NOT NULL
CUSTOMER_LNAME	VARCHAR(20)	NOT NULL
CUSTOMER_ADDRESS	VARCHAR(255)	NOT NULL
CUSTOMER_CITY	VARCHAR(20)	NOT NULL
CUSTOMER_ZIP	INT	NOT NULL
CUSTOMER_STATE	VARCHAR(20)	NOT NULL
CUSTOMER_PHONE	VARCHAR(20)	NOT NULL

8. **ORDER_ITEM**

Structure:

Column Name	**Data Type**	**Nullability**
ORDER_ITEM_ID (PRIMARY KEY)	VARCHAR(10)	NOT NULL
CATALOG_ID (FOREIGN KEY)	VARCHAR(10)	NOT NULL

ORDER_ID (FOREIGN KEY)	VARCHAR(10)	NOT NULL
RATE	FLOAT	NOT NULL
QUANTITY_ORDERED	INT	NOT NULL
ORDER_TOTAL	FLOAT	NOT NULL
PICKUP_DATE	DATETIME	NOT NULL
RETURN_DATE	DATETIME	NOT NULL
PICKUP_STATUS	VARCHAR(10)	NOT NULL
PICKUP_DELIVERY_ID (FOREIGN KEY)	VARCHAR(10)	NOT NULL

9. **CUSTOMER_ORDER**

Structure:

Column Name	**Data Type**	**Nullability**
ORDER_ID (PRIMARY KEY)	VARCHAR(10)	NOT NULL
CUSTOMER_ID (FOREIGN KEY)	VARCHAR(10)	NOT NULL
ORDER_DATE	DATE	NOT NULL
DELIVERY_ADDRESS	VARCHAR(255)	NOT NULL
DELIVERY_CITY	VARCHAR(255)	NOT NULL
DELIVERY_ZIP	VARCHAR(255)	NOT NULL
DELIVERY_STATE	VARCHAR(255)	NOT NULL
DELIVERY_DATE	DATE	NOT NULL
ORDER_STATUS	VARCHAR(10)	CAN BE NULL
ORDER_TOTAL	INT	NOT NULL
PAYMENT_STATUS	VARCHAR (10)	NOT NULL
DELIVERY_ID (FOREIGN KEY)	VARCHAR(10)	NOT NULL

10. DELIVERY PERSON

Structure:

Column Name	Data Type	Nullability
DELIVERY_ID (PRIMARY KEY)	VARCHAR(10)	NOT NULL
DELIVERY_FNAME	VARCHAR(20)	NOT NULL
DELIVERY_LNAME	VARCHAR(20)	NOT NULL
DELIVERY_PHONE	VARCHAR(20)	NOT NULL

7. Queries

1) Order placed according to month.

```
SELECT COUNT(Order_ID) as 'NUMBER OF ORDERS PLACED',
MONTHNAME (Order_Date) as 'MONTH'
FROM CUSTOMER_ORDER
GROUP BY  EXTRACT(MONTH FROM Order_Date)
ORDER BY  EXTRACT(MONTH FROM Order_Date);
```

NUMBER OF ORDERS PLACED	MONTH
5	January
4	February
5	March
4	April
6	May
5	June
2	July
3	August
3	September
3	October
2	November
8	December

2) Total Sales revenue generated every month.

```
SELECT SUM(Order_Total) as 'TOTAL REVENUE', MONTHNAME
(Order_Date) as 'MONTH'
FROM CUSTOMER_ORDER
GROUP BY EXTRACT(MONTH FROM Order_Date)
ORDER BY EXTRACT(MONTH FROM Order_Date);
```

TOTAL REVENUE	MONTH
299	January
267	February
310	March
267	April
466	May
342	June
137	July
215	August
165	September
185	October
154	November
586	December

3) List all the order details of designer Xavier Rice.

```
SELECT CO.Order_ID,C.Catalog_ID,C.Description,CO.Order_Date,
O.Return_Datetime
FROM DESIGNER D, CATALOG C,ORDER_ITEM O,CUSTOMER_ORDER CO
WHERE D.Designer_ID = C.Designer_ID AND C.Catalog_ID = O.Catalog_Id
AND CO.Order_ID = O.Order_Id AND
D.Designer_Fname = 'Xavier' AND D.Designer_Lname = 'Rice';
```

Order_ID	Catalog_ID	Order_Date	Return_Datetime
ordr02	ct10	2017-01-10 00:00:00	2017-01-23 00:00:00
ordr37	ct10	2017-12-21 00:00:00	2018-01-03 00:00:00
ordr42	ct10	2018-02-07 00:00:00	2018-02-21 00:00:00
ordr46	ct10	2018-04-01 00:00:00	2018-04-14 00:00:00
ordr17	ct13	2017-06-17 00:00:00	2017-06-30 00:00:00
ordr20	ct13	2017-07-14 00:00:00	2017-07-27 00:00:00
ordr27	ct13	2017-10-15 00:00:00	2017-10-29 00:00:00
ordr40	ct13	2018-01-02 00:00:00	2018-01-06 00:00:00
ordr44	ct13	2017-03-17 00:00:00	2018-03-27 00:00:00
ordr04	ct14	2017-02-03 00:00:00	2017-02-16 00:00:00
ordr16	ct14	2017-06-06 00:00:00	2017-06-19 00:00:00
ordr10	ct15	2017-05-11 00:00:00	2017-05-24 00:00:00
ordr23	ct15	2017-08-31 00:00:00	2017-09-14 00:00:00

4) Top Designer

```
SELECT Designer_Fname AS 'DESIGNER'

FROM DESIGNER

WHERE Designer_ID IN (SELECT Designer_ID FROM CATALOG WHERE
Catalog_ID

IN( SELECT Catalog_ID FROM ((SELECT Catalog_ID,
COUNT(OI1.Catalog_ID)

FROM order_item OI1 GROUP BY OI1.Catalog_ID
```

```
HAVING COUNT(OI1.Catalog_ID)
=
(SELECT MAX(COUNTED) FROM ( SELECT OI.Catalog_ID
AS'DESIGNER',COUNT(OI.Catalog_ID) AS 'COUNTED' FROM Designer D,
Catalog C, order_item OI WHERE D.Designer_ID = C.Designer_ID AND
OI.Catalog_Id = C.Catalog_ID
GROUP BY OI.Catalog_ID) AS R1)) C)));
```

Designer_Fname	Designer_Lname
Courtney	Shaw

5) Number of verifications done by each system representative.

```
SELECT SR.SysRep_ID, SR.SysRep_Fname,COUNT(SR.SysRep_ID) AS '# Of
Verification'
FROM SYSTEM_REPRESENTATIVE SR, DESIGNER_VERIFICATION DV
WHERE SR.SysRep_ID = DV.SysRep_ID
GROUP BY SR.SysRep_ID, SR.SysRep_Fname;
```

SysRep_ID	SysRep_Fname	# Of Verification
SR100	Tyrone	1
SR101	Driscoll	1
SR102	Garrison	2
SR103	Nelle	1

6) how many orders the delivery person has picked up and delivered

```
SELECT Delivery_Fname AS 'DELIVERY PERSON', COUNT(Pickup_Datetime)
AS 'ORDERS PICKED',
COUNT(Return_Datetime) AS 'ORDERS RETURNED'
FROM DELIVERY_PERSON, ORDER_ITEM
WHERE Pickup_Delivery_Id = Delivery_Id
GROUP BY Delivery_Fname
```

DELIVERY PERSON	ORDERS PICKED	ORDERS RETURNED
Cairo	7	7
Dustin	6	6
Fuller	6	5
Jessica	3	3
Lane	4	4
Linda	7	7
Samuel	5	5
Sara	5	5
Vladimir	5	5

7) **Top cities and zip codes that have ordered the most.**

```
SELECT Delivery_City as 'CITY',Delivery_Zip as 'ZIPCODE',
COUNT(Order_Id) as 'NO OF ORDERS'
FROM CUSTOMER_ORDER
GROUP BY Delivery_City,Delivery_Zip
ORDER BY 'NO OF ORDERS' DESC
LIMIT 3;
```

City	Zip	Total ord...
San Jose	50170	1
Pleasanton	99713	2
hayward	61600	3
San Fransico	90720	3
oakland	58579	3
Mountain view	61600	4
San Jose	71867	8
Pleasanton	82282	11
San Jose	30177	15

8) **what was the rating and issues received by month**

```
SELECT CD.Issue_Description as 'ISSUE',
CD.Rating,MONTHNAME(cd.Call_Date) as 'MONTH'
FROM CALL_DETAILS CD
GROUP BY EXTRACT(MONTH FROM (CD.Call_Date))
ORDER BY EXTRACT(MONTH FROM (CD.Call_Date));
```

ISSUE	RATING	MONTH
Package Enquiry	1	January
Size is small	5	February
Products & Policy Enquiry	1	March
Payment Assistance	3	April
Size is small	5	May
Products & Policy Enquiry	1	June
Package Enquiry	2	July
Package Enquiry	5	August
Size issue	2	September
Size is small	5	October
Payment Assistance	3	November
Payment Assistance	4	December

9) **Average rating of each of the customer representatives**

```
SELECT Cusrep_Fname as 'CUSTOMER REPRESENTATIVE', AVG(Rating)
as 'AVERAGE RATING'
FROM CALL_DETAILS C, CUSTOMER_REPRESENTATIVE CR
WHERE C.Cusrep_Id = CR.Cusrep_Id
GROUP BY CR.Cusrep_Id
```

CUSTOMER REPRESENTATIVE	AVERAGE RATING
Renee	3.2000
Veronica	2.6667
Beverly	2.7500
Declan	2.6667
Hiroko	3.7500
Baker	2.8000
Kylee	4.3333
Aidan	2.0000
Fritz	2.5000
Fleur	2.0000

10) Customers who billing address is different from delivery address

```
SELECT C.Customer_Fname ,C.Customer_Lname,C.Customer_Address as
'BILLING ADDRESS',Customer_City as 'BILLING CITY',Customer_Zip as
'BILLING ZIP',Delivery_Address as 'DELIVERYADDRESS',Delivery_City as
'DELIVERY CITY',Delivery_Zip as 'DELIVERY ZIP'
FROM CUSTOMER C,CUSTOMER_ORDER CO
WHERE C.Customer_Id = CO.Customer_Id AND C.Customer_City <>
CO.Delivery_City AND C.Customer_Zip <> CO.Delivery_Zip AND
C.Customer_Address <> CO.Delivery_Address;
```

customer_Fname	customer_Lname	billing address	billing city	billing zip	delivery address	delivery city	delivery zip
Gary	Crosby	P.O. Box 710, 2592 Non Ave	Pleasanton	82282	652-114 Interdum. St	San Jose	71867
Raven	Love	5987 Eu, Av.	hayward	61600	P.O. Box 871, 3652 L	oakland	58579

11) Customer who has placed maximum orders.

```
SELECT  CU.Customer_Fname as 'CUSTOMER NAME', COUNT(CO.Order_Id)
as 'TOTAL ORDER'
FROM CUSTOMER_ORDER CO , Customer CU
WHERE CU.Customer_ID = CO.Customer_Id
GROUP BY CO.Customer_id
HAVING COUNT(CO.Order_Id)
= (SELECT MAX(C.Order_Count)
FROM
  (SELECT COUNT(CO1.Order_Id) as 'ORDER_COUNT', CO1.Order_Id
FROM CUSTOMER_ORDER CO1
GROUP BY CO1.Customer_id) C);
```

Customer Name	Total Order
Asher	9

12) Most ordered cloth item details.

```
SELECT *
FROM CATALOG
WHERE catalog_id IN (
SELECT C.catalog_id
FROM(
SELECT Catalog_ID , COUNT(Catalog_id)
FROM ORDER_ITEM
GROUP BY catalog_id
HAVING COUNT(Catalog_id) = ( SELECT MAX(R.counted)
FROM (SELECT oi.catalog_id, COUNT(oi.catalog_id) as 'COUNTED'
FROM ORDER_ITEM oi
GROUP BY oi.catalog_id) R)) C);
```

Catalog_ID	Designer_ID	Description	Rate	Material	Type	Occasion	Season	Neckline	Color	Availibility
ct12	ds941	Backless High Low Cami Dress	75	Rayon	Slip	Night Out	Summer	Spaghetti Strap	Yellow	available

13) List of all the call details who's rating is < 3

```
SELECT CD.CusRep_ID, CR.CusRep_Fname,CR.CusRep_Lname,
CD.Issue_Description, CD.Rating, C.Customer_ID, C.Customer_Fname,
C.Customer_Lname
FROM CALL_DETAILS CD, CUSTOMER_REPRESENTATIVE CR, Customer C
WHERE CD.CusRep_ID = CR.CusRep_ID AND CD.Customer_ID =
C.Customer_ID AND CD.Rating < 3;
```

CusRep_ID	CusRep_Fname	CusRep_Lname	Issue_Description	Rating	Customer_ID	Customer_Fname	Customer_Lname
CR100	Renee	Neal	Size issue	2	cs101	Philip	Potts
CR100	Renee	Neal	Package Enquiry	1	cs102	Diana	Riley
CR101	Veronica	Harrell	Payment Assistance	1	cs106	Abigail	Preston
CR102	Beverly	Mejia	Products & Policy Enquiry	1	cs109	Arden	Howell
CR102	Beverly	Mejia	Size issue	1	cs107	Frances	Byer
CR103	Declan	Bernard	Package Enquiry	2	cs103	Norman	Walter
CR103	Declan	Bernard	Products & Policy Enquiry	1	cs105	Emma	Higgins
CR104	Hiroko	Rocha	Tracking package delivery	2	cs103	Norman	Walter
CR105	Baker	Howe	Package Enquiry	1	cs109	Arden	Howell
CR105	Baker	Howe	Payment Assistance	2	cs104	Gary	Crosby
CR107	Aidan	Boyer	Products & Policy Enquiry	1	cs104	Gary	Crosby
CR107	Aidan	Boyer	Products & Policy Enquiry	2	cs106	Abigail	Preston
CR108	Fritz	Lopez	Size issue	2	cs105	Emma	Higgins
CR109	Fleur	Scott	Payment Assistance	1	cs102	Diana	Riley

8. INSERTION:

```
INSERT INTO
`Catalog`(Catalog_ID,Designer_ID,Description,Rate,Material,Type,Occasion,Season,Neckline,Color,Availibility) VALUES('ct12','ds941','Backless High Low Cami Dress','75','Rayon','Slip','Night Out','Summer', 'Spaghetti Strap','Yellow','available');
```

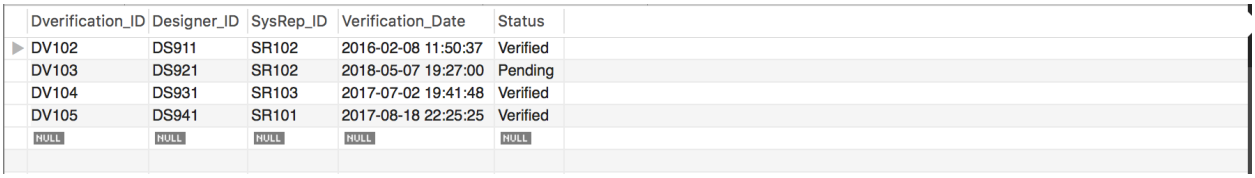
After insertion:

Catalog_ID	Designer_ID	Description	Rate	Material	Type	Occasion	Season	Neckline	Color	Availibility
ct12	ds941	Backless High Low Cami Dress	75	Rayon	Slip	Night Out	Summer	Spaghetti Strap	Yellow	available
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

9. DELETION:

```
DELETE FROM  DESIGNER_VERIFICATION WHERE Dverification_ID = `DV101`;
```

After deletion:

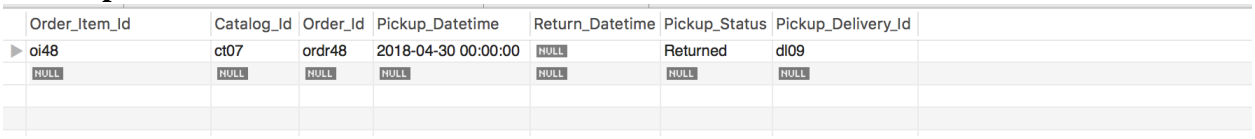


Dverification_ID	Designer_ID	SysRep_ID	Verification_Date	Status
DV102	DS911	SR102	2016-02-08 11:50:37	Verified
DV103	DS921	SR102	2018-05-07 19:27:00	Pending
DV104	DS931	SR103	2017-07-02 19:41:48	Verified
DV105	DS941	SR101	2017-08-18 22:25:25	Verified
NULL	NULL	NULL	NULL	NULL

10.UPDATION:

```
UPDATE ORDER_ITEM SET Pickup_Status='Returned' WHERE
Order_Item_Id='oi48';
```

After updation:



Order_Item_Id	Catalog_Id	Order_Id	Pickup_Datetime	Return_Datetime	Pickup_Status	Pickup_Delivery_Id
oi48	ct07	ordr48	2018-04-30 00:00:00	NULL	Returned	dl09
NULL	NULL	NULL	NULL	NULL	NULL	NULL

11. VIEWS

```
CREATE VIEW `DELIVERY INSIGHTS` as
SELECT
DP.Delivery_ID,DP.Delivery_Fname,DP.Delivery_Lname,C.Customer_Fname,C.Customer_ID,C.Customer_Lname,CT.Catalog_Id,CT.Description,
D.Designer_ID,D.Designer_Fname,D.Designer_Lname
FROM Delivery_Person DP,Customer_Order CO,Customer C,Order_Item OI,Catalog CT,Designer D
WHERE DP.Delivery_ID = CO.Delivery_ID AND CO.Customer_ID = C.Customer_ID
AND OI.Order_ID = CO.Order_ID AND CT.Designer_ID = D.Designer_ID;
```

After creating view:

Details of delivery person activities whos name is Dustin and delivery id is dl02.

```
SELECT * FROM `DELIVERY INSIGHTS` WHERE Delivery_ID = 'dl02';
```

Delivery_ID	Delivery_Fname	Delivery_Lname	Customer_Fname	Customer_ID	Customer_Lname	Catalog_Id	Description	Designer_ID
dl02	Dustin	Flowers	Norman	cs103	Walter	ct01	Tribal print Bardot dress	ds901
dl02	Dustin	Flowers	Norman	cs103	Walter	ct02	Plunging Neck Floral Dress	ds901
dl02	Dustin	Flowers	Norman	cs103	Walter	ct03	Flower Print Ruffle Dip Hem Cami Dress	ds901
dl02	Dustin	Flowers	Norman	cs103	Walter	ct04	Thick Strap Lace Up Low Back Layered Dress	ds901
dl02	Dustin	Flowers	Norman	cs103	Walter	ct05	Split side embroidered applique dress	ds901
dl02	Dustin	Flowers	Diana	cs102	Riley	ct01	Tribal print Bardot dress	ds901
dl02	Dustin	Flowers	Diana	cs102	Riley	ct02	Plunging Neck Floral Dress	ds901
dl02	Dustin	Flowers	Diana	cs102	Riley	ct03	Flower Print Ruffle Dip Hem Cami Dress	ds901
dl02	Dustin	Flowers	Diana	cs102	Riley	ct04	Thick Strap Lace Up Low Back Layered Dress	ds901
dl02	Dustin	Flowers	Diana	cs102	Riley	ct05	Split side embroidered applique dress	ds901
dl02	Dustin	Flowers	Norman	cs103	Walter	ct01	Tribal print Bardot dress	ds901
dl02	Dustin	Flowers	Norman	cs103	Walter	ct02	Plunging Neck Floral Dress	ds901
dl02	Dustin	Flowers	Norman	cs103	Walter	ct03	Flower Print Ruffle Dip Hem Cami Dress	ds901
dl02	Dustin	Flowers	Norman	cs103	Walter	ct04	Thick Strap Lace Up Low Back Layered Dress	ds901
dl02	Dustin	Flowers	Norman	cs103	Walter	ct05	Split side embroidered applique dress	ds901
dl02	Dustin	Flowers	Asher	cs100	Dyer	ct01	Tribal print Bardot dress	ds901
dl02	Dustin	Flowers	Asher	cs100	Dyer	ct02	Plunging Neck Floral Dress	ds901

DELIVERY INSIGHTS 50

Action Output

	Time	Action	Response	Duration / F
81	21:25:06	select * from order_item LIMIT 0, 5...	50 row(s) returned	0.00027 sec
82	21:25:15	SELECT * FROM `DELIVERY INSIGH...	60 row(s) returned	0.00064 sec

```
CREATE VIEW `CUSTOMER FEEDBACK INSIGHTS 12` as
SELECT CD.CusRep_ID, CR.CusRep_Fname,CR.CusRep_Lname,
CD.Issue_Description, CD.Rating, C.Customer_ID, C.Customer_Fname,
C.Customer_Lname
FROM CALL_DETAILS CD, CUSTOMER_REPRESENTATIVE CR, Customer C
WHERE CD.CusRep_ID = CR.CusRep_ID AND CD.Customer_ID = C.Customer_ID;
```

After creating view:

Details of all the calls that have rating greater than 3

```
SELECT * FROM `CUSTOMER FEEDBACK INSIGHTS 12` WHERE Rating > 3;
```

CusRep_ID	CusRep_Fname	CusRep_Lname	Issue_Description	Rating	Customer_ID	Customer_Fname	Customer_Lname
CR100	Renee	Neal	Size issue	5	cs103	Norman	Walter
CR100	Renee	Neal	Order confirmation assistance	4	cs108	Raven	Love
CR100	Renee	Neal	Damaged Product	4	cs109	Arden	Howell
CR101	Veronica	Harrell	Package Enquiry	4	cs108	Raven	Love
CR102	Beverly	Mejia	Size is small	5	cs101	Philip	Potts
CR102	Beverly	Mejia	Products & Policy Enquiry	4	cs107	Frances	Byer
CR103	Declan	Bernard	Followup on new package delivery	5	cs101	Philip	Potts
CR104	Hiroko	Rocha	Size is small	5	cs106	Abigail	Preston
CR104	Hiroko	Rocha	Size is small	5	cs102	Diana	Riley
CR105	Baker	Howe	Package Enquiry	5	cs104	Gary	Crosby
CR106	Kylee	Barry	Payment Assistance	4	cs107	Frances	Byer
CR106	Kylee	Barry	Payment Assistance	5	cs105	Emma	Higgins
CR106	Kylee	Barry	Products & Policy Enquiry	4	cs102	Diana	Riley

12. INDEX

```
CREATE UNIQUE INDEX on CUSTOMER_ORDER (Order_ID)

CREATE UNIQUE INDEX on CUSTOMER (Customer_ID)
```

13. TRIGGERS

```
CREATE TABLE `Catalog_audit` (
  id INT AUTO_INCREMENT PRIMARY KEY,
  Rate INT NOT NULL,
  changedat DATETIME DEFAULT NULL,
  action VARCHAR(50) DEFAULT NULL);


DELIMITER $$
CREATE TRIGGER before_Catalog_update
  BEFORE UPDATE ON Catalog
```

```
   FOR EACH ROW
BEGIN
   INSERT INTO catalog_audit
   SET action = 'update',
    Rate= OLD.Rate,
     changedat = NOW();
END$$
DELIMITER ;

UPDATE CATALOG SET Rate =85 WHERE Catalog_ID ='ct12';
```

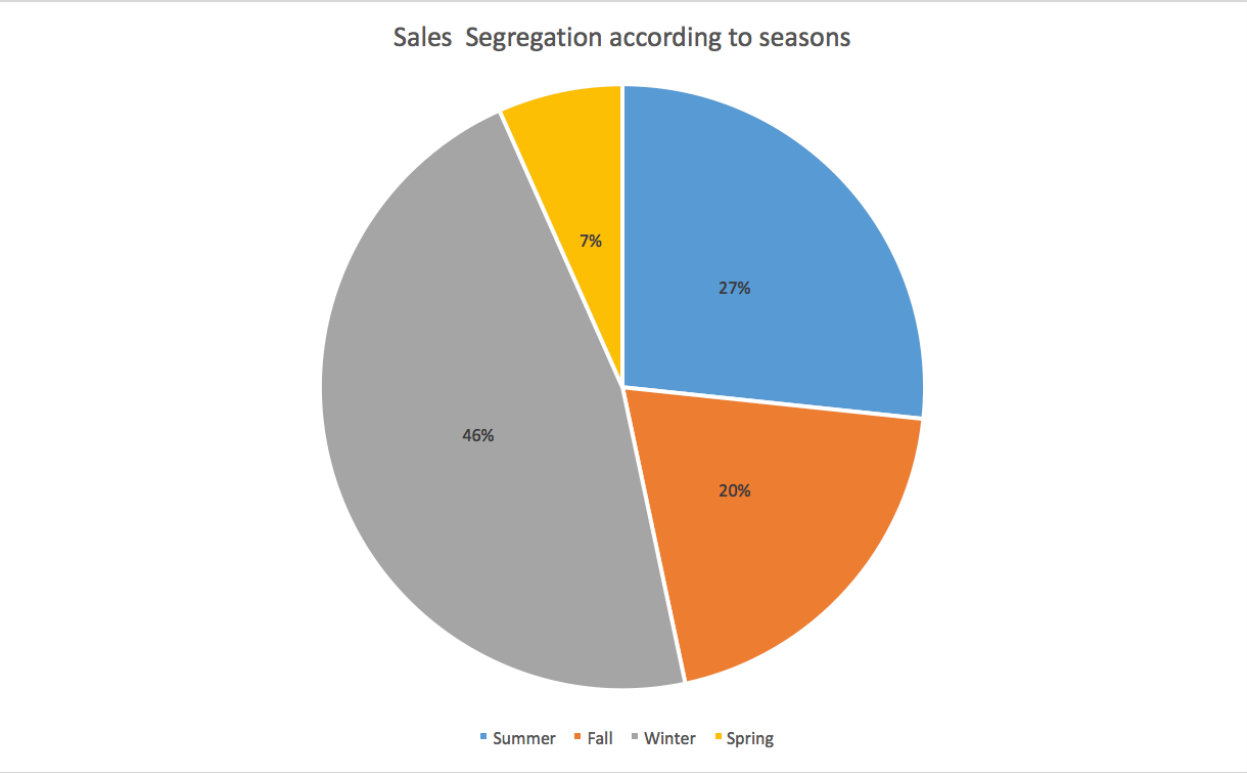
After Updation:

```
SELECT * FROM catalog_audit;
```

id	Catalog_id	Rate	changedat	action
1	ct12	85	2018-06-01 23:17:16	update

14. BUSINESS METRICS

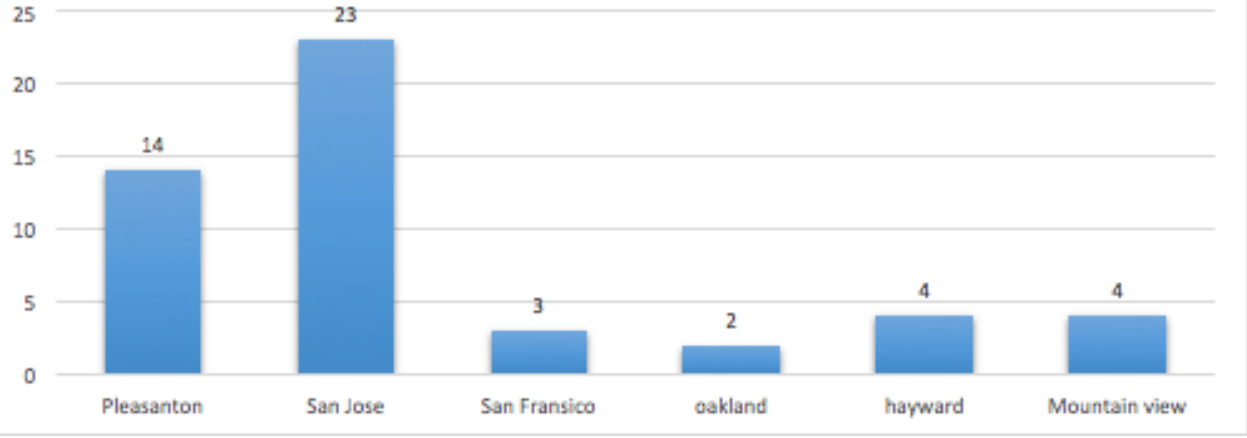
1)Sales Segre mentation according to seasons



2)Orders placed by customer according to city.

```
SELECT COUNT(CO.Order_ID) as 'TOTAL ORDERS', C.Customer_City
FROM CUSTOMER C ,CUSTOMER_ORDER CO
WHERE C.Customer_Id = CO.Customer_ID
GROUP BY C.Customer_City;
```

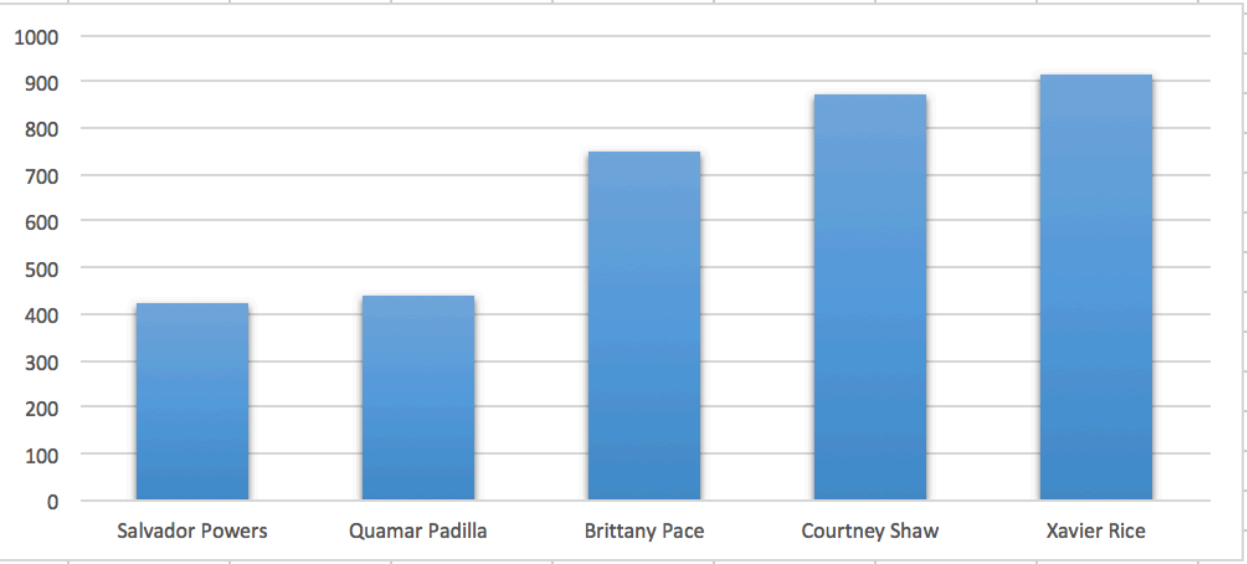
Total orders	Customer_City
4	hayward
4	Mountain view
2	oakland
14	Pleasanton
3	San Fransico
23	San Jose



3)Sales generated by designers.

```
SELECT SUM(CO.Order_Total) as 'Total Sales', d.Designer_Fname,
d.Designer_Lname
FROM CUSTOMER_ORDER CO,DESIGNER D,ORDER_ITEM OI , CATALOG CT
WHERE CO.Order_Id = OI.Order_Id And CT.Catalog_Id = OI.Catalog_Id And
D.Designer_Id = CT.Designer_Id
GROUP BY D.Designer_Id;
```

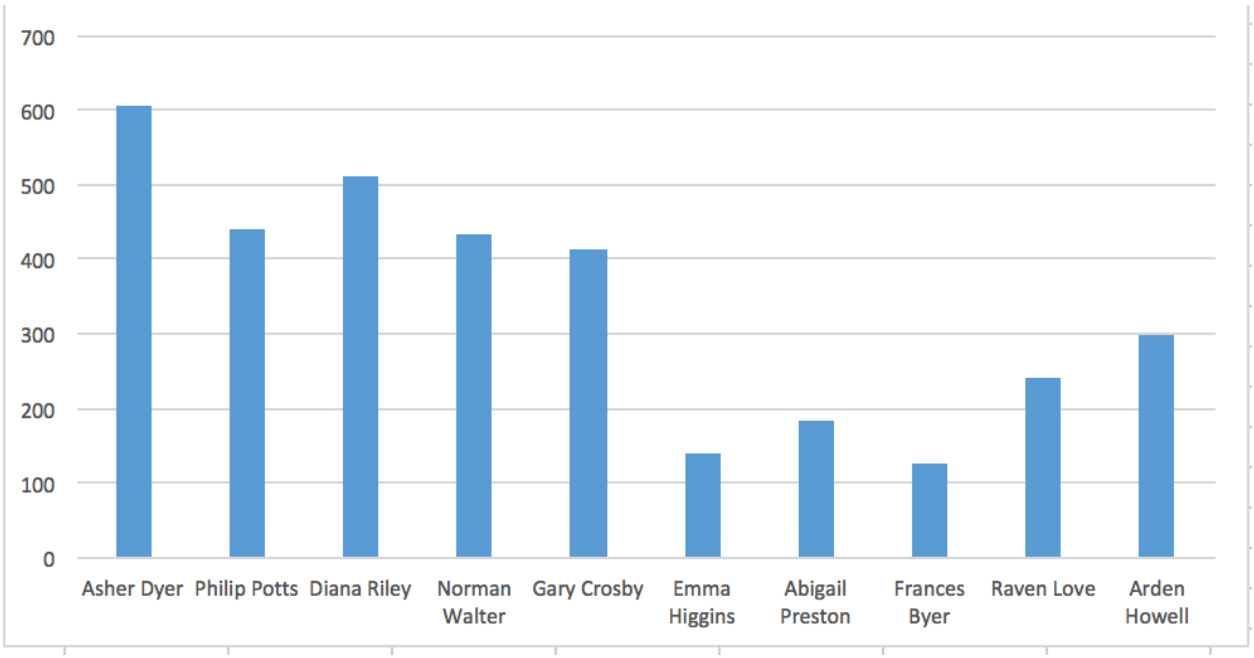
Total Sales	Designer_Fname	Designer_Lname
748	Brittany	Pace
422	Salvador	Powers
440	Quamar	Padilla
913	Xavier	Rice
870	Courtney	Shaw



4) Top ten customer Revenue generated

```
SELECT SUM(CO.Order_Total) as 'Total Sales', D.Customer_Fname,
D.Customer_Lname
FROM CUSTOMER_ORDER CO,CUSTOMER D,ORDER_ITEM OI , CATALOG CT
WHERE CO.Order_Id = OI.Order_Id AND CT.Catalog_Id = OI.Catalog_Id AND
D.Customer_Id = CO.Customer_Id
GROUP BY D.Customer_Id;
```

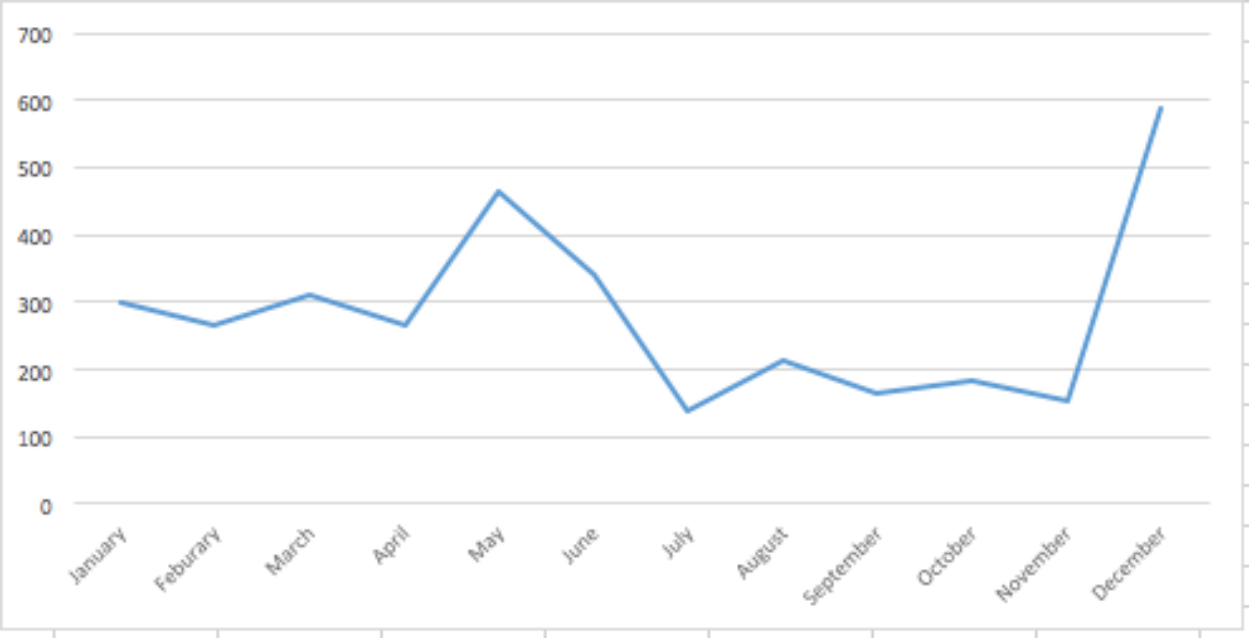
Total Sales	Customer_Fname	Customer_Lname
606	Asher	Dyer
439	Philip	Potts
511	Diana	Riley
434	Norman	Walter
414	Gary	Crosby
140	Emma	Higgins
185	Abigail	Preston
127	Frances	Byer
240	Raven	Love
297	Arden	Howell



5)Trend of ordering according to month.

```
SELECT SUM(Order_Total) as 'TOTAL REVENUE', MONTHNAME (Order_Date) as
'MONTH'
FROM CUSTOMER_ORDER
GROUP BY  EXTRACT(MONTH FROM Order_Date)
ORDER BY  EXTRACT(MONTH FROM Order_Date);
```

TOTAL REVENUE	MONTH
299	January
267	February
310	March
267	April
466	May
342	June
137	July
215	August
165	September
185	October
154	November
586	December



15. Project Summary:

The database we implemented can be used to automate an entire business process. Each cog in the wheel of Rental Restyle business segment can be automated using this database. We looked at the actor(s) and their interaction with the system which resulted in formation of data.

Post completion of the project we learned a few valuable lessons.

We learned that:

1. Data has a value when it is easy to analyze upon.
2. Raw data has to be shaped into a database in such a way that it is free for analysis.
3. How to look at a business process from a Data perspective, where it is kept, when and how it is fetched.
4. How to design UML and Use-cases.

Throughout the span of this project, we encountered a few problems. They were:

1. Limiting the scope of the project: For instance, the team decided not to include a inventory in the system as functionality.
2. Designing use-cases for the system: Identifying the granularity of each item that was available for rent, whether we had to consider each item individually.

3. Business Metrics: Deciding as to which business metrics were important and to include in the project.

We resolved these problems through:

1. The guidance of professor

2. Frequent team meets and brainstorming sessions.

If we were to do this project all over again, we would brainstorm more on the utility of the project with an emphasis on analytical processing of the data and using an output-oriented streamlining of operations.