# DRONE DELIVERY SYSTEM

## A PROJECT REPORT

*In partial fulfilment for the course*

*Of*

## DRONE DELIVERY SYSTEM

**GROUP: CLUSTERS**

**TEAM MEMBERS:**

1. ***Avinita Shetty***
2. ***Neha Reddy***
3. ***Pragalbha Yadav***
4. ***Priyanka Kadam***
5. ***Vishwa Shah***

# NORTHEASTERN UNIVERSITY

# BOSTON, MASSACHUSSETTS

# APRIL 2015

# Acknowledgement

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are highly indebted to Prof. Chaiyaporn Mutsalklisana for his guidance and constant supervision as well as for providing necessary information regarding the project & also for his support in completing the project.

We would like to express our gratitude towards the members of Northeastern University for their kind co-operation and encouragement which help us in completion of this project.

We would like to express our special gratitude and thanks to our Teaching Assistant Tanmay Ingle and Vaibhav Pattani for giving us his attention, time and valuable suggestions.

Our thanks and appreciations also go to our colleague in developing the project and people who have willingly helped us out with their abilities.

# INDEX

# Introduction

Technology is evolving at a rapid phase, to keep up with the demands and needs of people across the globe. One such technical innovation is the rapid expansion in the usage of Drones. Drones are unmanned aerial vehicles that are used mostly to aid the military and combat services. Drones are faster and less prone to human error, and hence have potential to take over industries where speed and accurate delivery is a key criteria. There are several industries such as healthcare and ecommerce industries that are looking to invest in this technology to improve the services that they provide.

The objective of our project is to build a database for a typical ecommerce website which includes the use drones to deliver their products rather than the traditional methods of delivery.

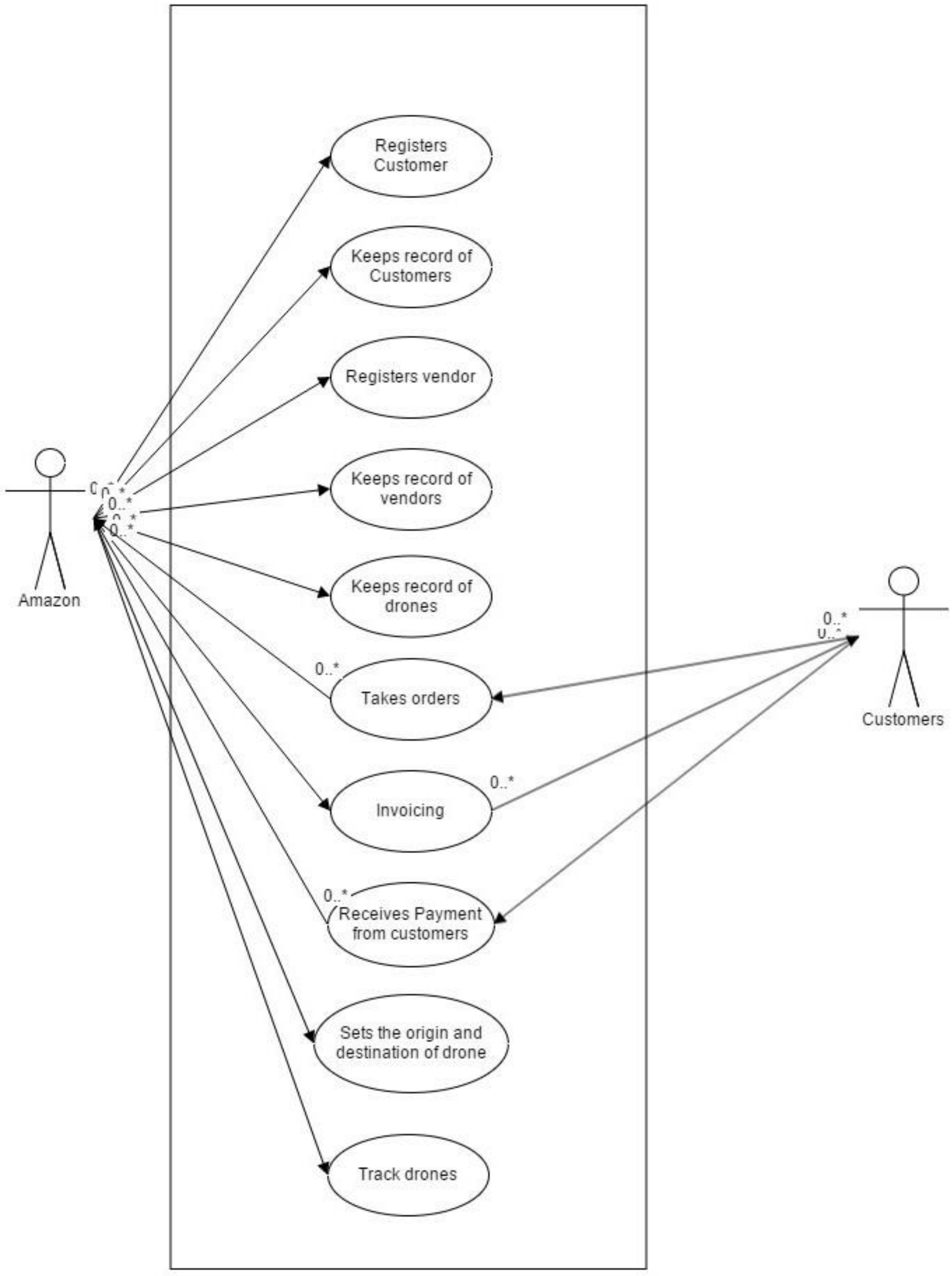
It includes details about customers using the services of this website, along with the drone details and their related information. It also includes data about products and payment information, which shows the jest of the functionality of this website with this kind of delivery service.

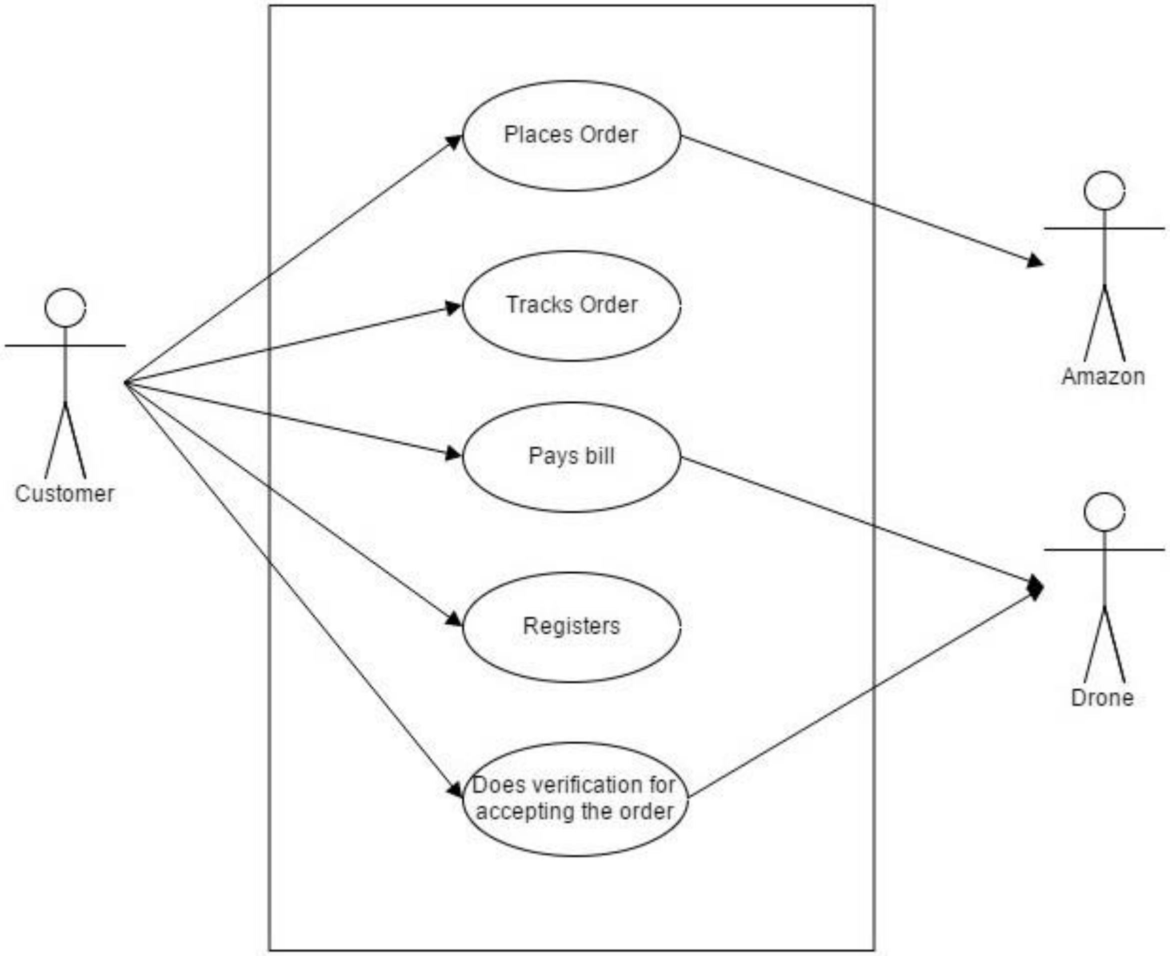
The drones are going to use zip codes to find the exact path that the drone takes from the warehouse address to the customer shipping address. We have added the weather capabilities to the drone as an extra functionality, which might benefit the company in better understanding how the system and overcome certain difficulties.

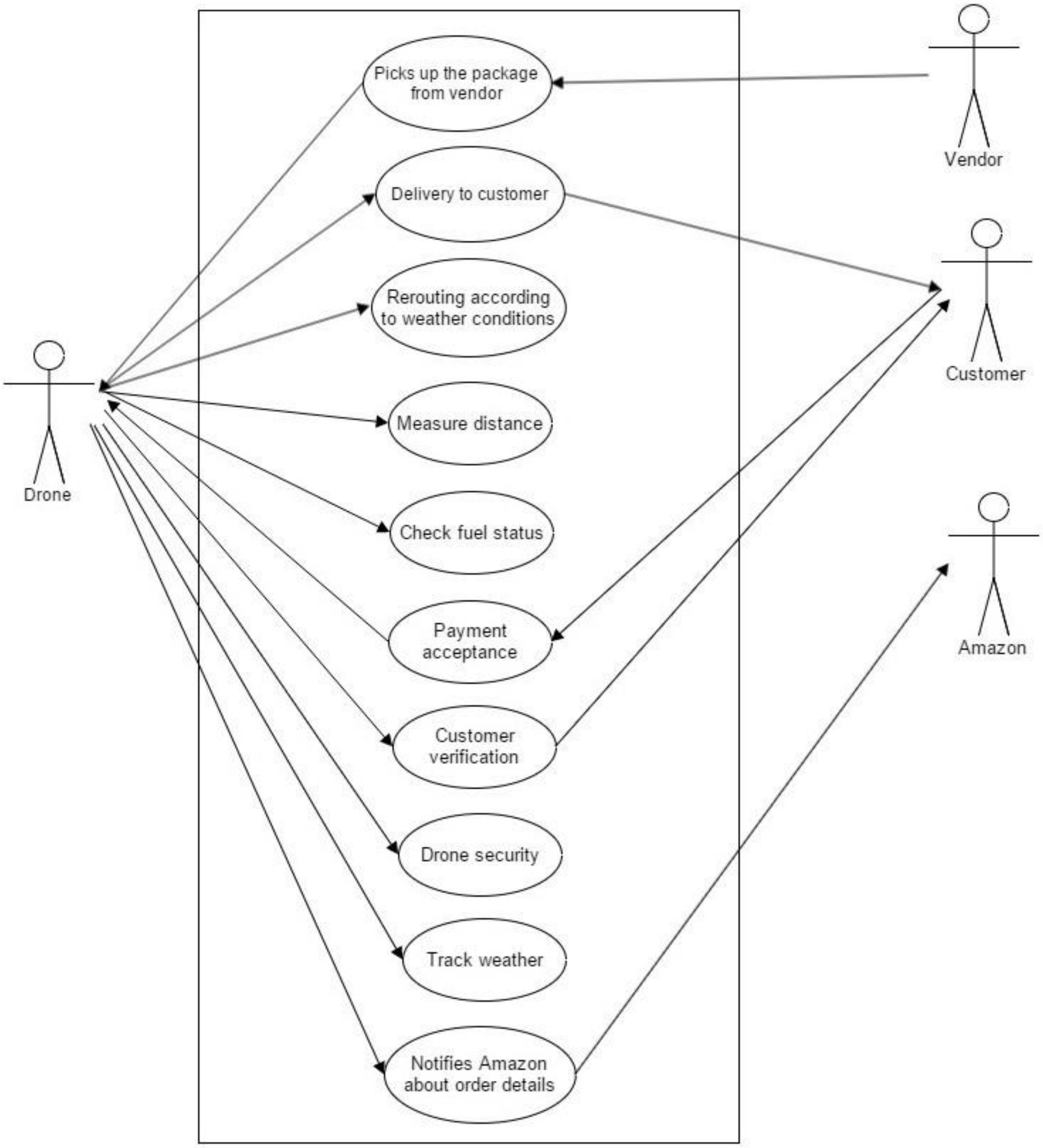
# **Limitations**

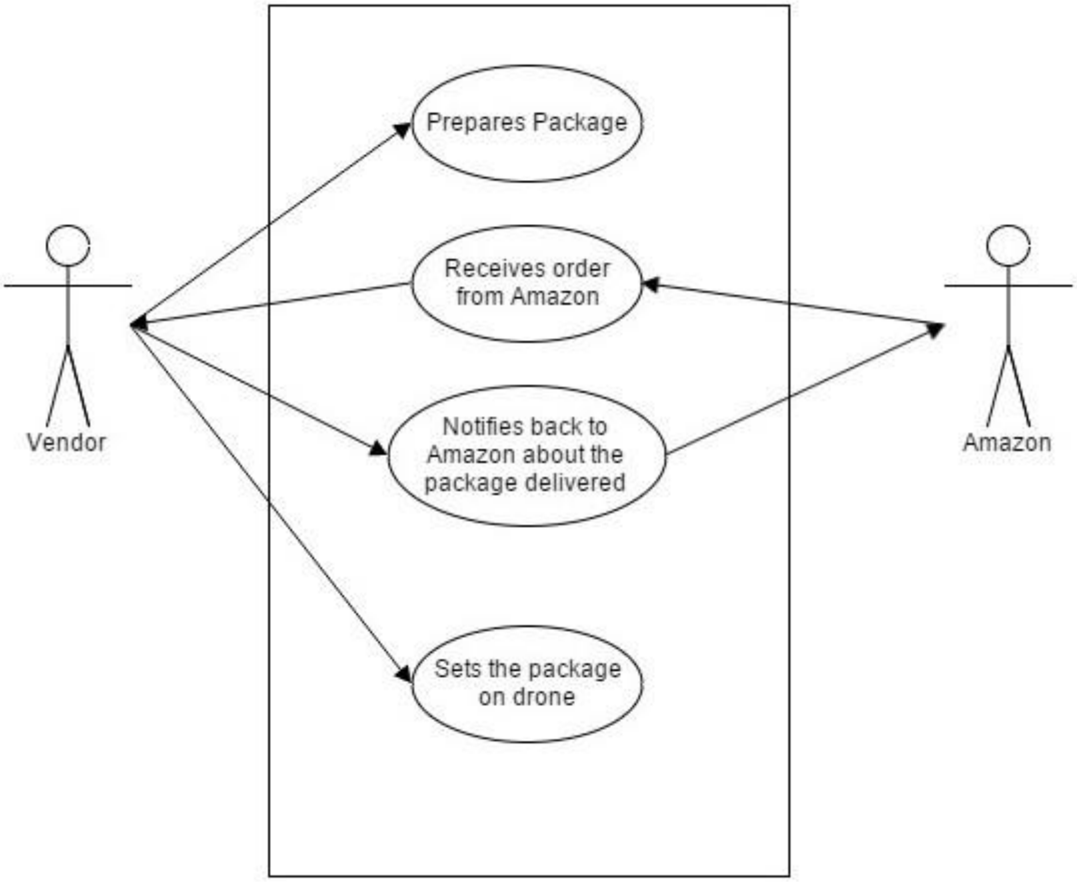
- The e-commerce website is assumed to be amazon.com.
- It has vendors and customers from USA only.
- Warehouses includes vendor warehouses and amazon warehouses.
- The product is delivered by a drone present in that product vendor warehouse.
- Customers can order only one product at a time.
- One drone will carry only one order item at one time.
- No cash on delivery option.
- The customer is assumed to get package directly, i.e drop off at the zip code location.
- No partial transactions are taken into consideration.
- The order item is created only after the payment is completed.
- One category is associated to only one department.

# Use Case Diagrams

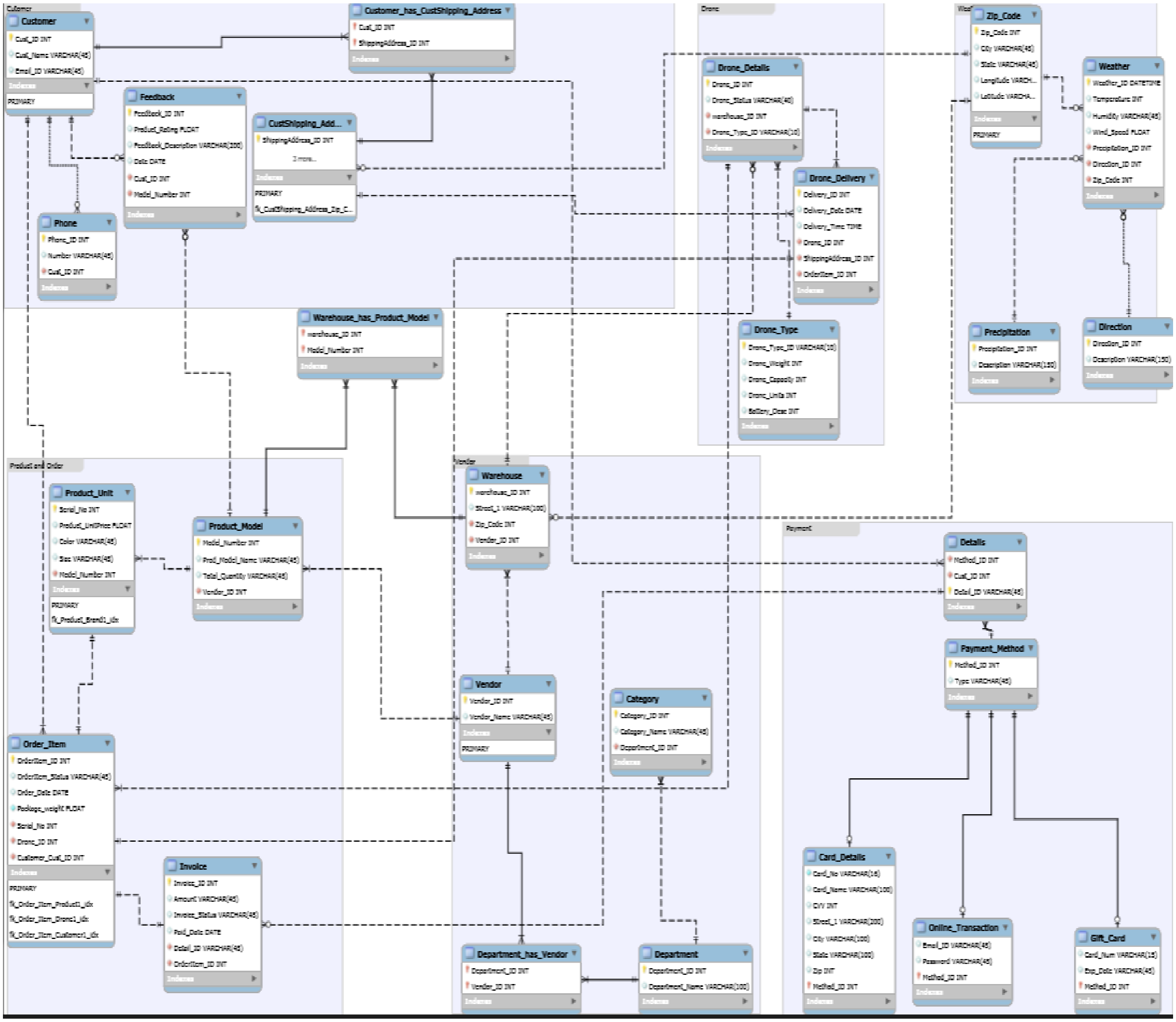



Places Order
Tracks Order
Pays bill
Registers
Does verification for accepting the order
Customer
Amazon
Drone


Picks up the package from vendor
Delivery to customer
Rerouting according to weather conditions
Measure distance
Check fuel status
Payment acceptance
Customer verification
Drone security
Track weather
Notifies Amazon about order details
Drone
Vendor
Customer
Amazon


Prepares Package
Receives order from Amazon
Notifies back to Amazon about the package delivered
Sets the package on drone
Vendor
Amazon

# EER DIAGRAM


Customer
Customer
Cust_ID INT
Cust_Name VARCHAR(45)
Email_ID VARCHAR(45)
Indexes
PRIMARY
Customer_has_CustShipping_Address
Cust_ID INT
ShippingAddress_ID INT
Indexes
Feedback
Feedback_ID INT
Product_Rating FLOAT
Feedback_Description VARCHAR(200)
Date DATE
Cust_ID INT
Model_Number INT
Indexes
CustShipping_Add...
ShippingAddress_ID INT
3 more...
Indexes
PRIMARY
fk_CustShipping_Address_Zip_C...
Phone
Phone_ID INT
Number VARCHAR(45)
Cust_ID INT
Indexes
Drone
Drone_Details
Drone_ID INT
Drone_Status VARCHAR(45)
warehouse_ID INT
Drone_Type_ID VARCHAR(10)
Indexes
Drone_Delivery
Delivery_ID INT
Delivery_Date DATE
Delivery_Time TIME
Drone_ID INT
ShippingAddress_ID INT
OrderItem_ID INT
Indexes
Drone_Type
Drone_Type_ID VARCHAR(10)
Drone_Weight INT
Drone_Capacity INT
Drone_Units INT
Battery_Desc INT
Indexes
Wea
Zip_Code
Zip_Code INT
City VARCHAR(45)
State VARCHAR(45)
Longitude VARCH...
Latitude VARCHA...
Indexes
PRIMARY
Weather
Weather_ID DATETIME
Temperature INT
Humidity VARCHAR(45)
Wind_Speed FLOAT
Precipitation_ID INT
Direction_ID INT
Zip_Code INT
Indexes
Precipitation
Precipitation_ID INT
Description VARCHAR(150)
Indexes
Direction
Direction_ID INT
Description VARCHAR(150)
Indexes
Warehouse_has_Product_Model
warehouse_ID INT
Model_Number INT
Indexes
Product and Order
Product_Unit
Serial_No INT
Product_UnitPrice FLOAT
Color VARCHAR(45)
Size VARCHAR(45)
Model_Number INT
Indexes
PRIMARY
fk_Product_Brand1_idx
Product_Model
Model_Number INT
Prod_Model_Name VARCHAR(45)
Total_Quantity VARCHAR(45)
Vendor_ID INT
Indexes
Vendor
Warehouse
warehouse_ID INT
Street_1 VARCHAR(100)
Zip_Code INT
Vendor_ID INT
Indexes
Vendor
Vendor_ID INT
Vendor_Name VARCHAR(45)
Indexes
PRIMARY
Category
Category_ID INT
Category_Name VARCHAR(45)
Department_ID INT
Indexes
Order_Item
OrderItem_ID INT
OrderItem_Status VARCHAR(45)
Order_Date DATE
Package_weight FLOAT
Serial_No INT
Drone_ID INT
Customer_Cust_ID INT
Indexes
PRIMARY
fk_Order_Item_Product1_idx
fk_Order_Item_Drone1_idx
fk_Order_Item_Customer1_idx
Invoice
Invoice_ID INT
Amount VARCHAR(45)
Invoice_Status VARCHAR(45)
Paid_Date DATE
Detail_ID VARCHAR(45)
OrderItem_ID INT
Indexes
Department_has_Vendor
Department_ID INT
Vendor_ID INT
Indexes
Department
Department_ID INT
Department_Name VARCHAR(100)
Indexes
Payment
Details
Method_ID INT
Cust_ID INT
Detail_ID VARCHAR(45)
Indexes
Payment_Method
Method_ID INT
Type VARCHAR(45)
Indexes
Card_Details
Card_No VARCHAR(16)
Card_Name VARCHAR(100)
CVV INT
Street_1 VARCHAR(200)
City VARCHAR(100)
State VARCHAR(100)
Zip INT
Method_ID INT
Indexes
Online_Transaction
Email_ID VARCHAR(45)
Password VARCHAR(45)
Method_ID INT
Indexes
Gift_Card
Card_Num VARCHAR(15)
Exp_Date VARCHAR(45)
Method_ID INT
Indexes

# Normalization

Database normalization is the process of organizing the fields and tables of a relational database

to minimize redundancy. Normalization usually involves dividing large tables into smaller tables

and defining relationships between them.

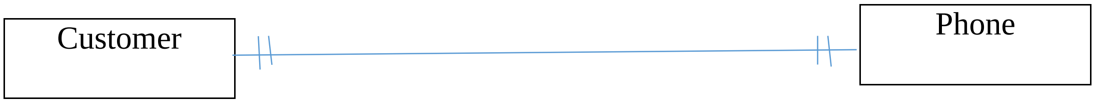
All the tables in the database system are normalized. It involves 1NF, 2NF and 3NF.

Enforcing 1 NF:

A table is in 1NF if it is free from multi-valued rows. In this database system all the tables contain only single values.

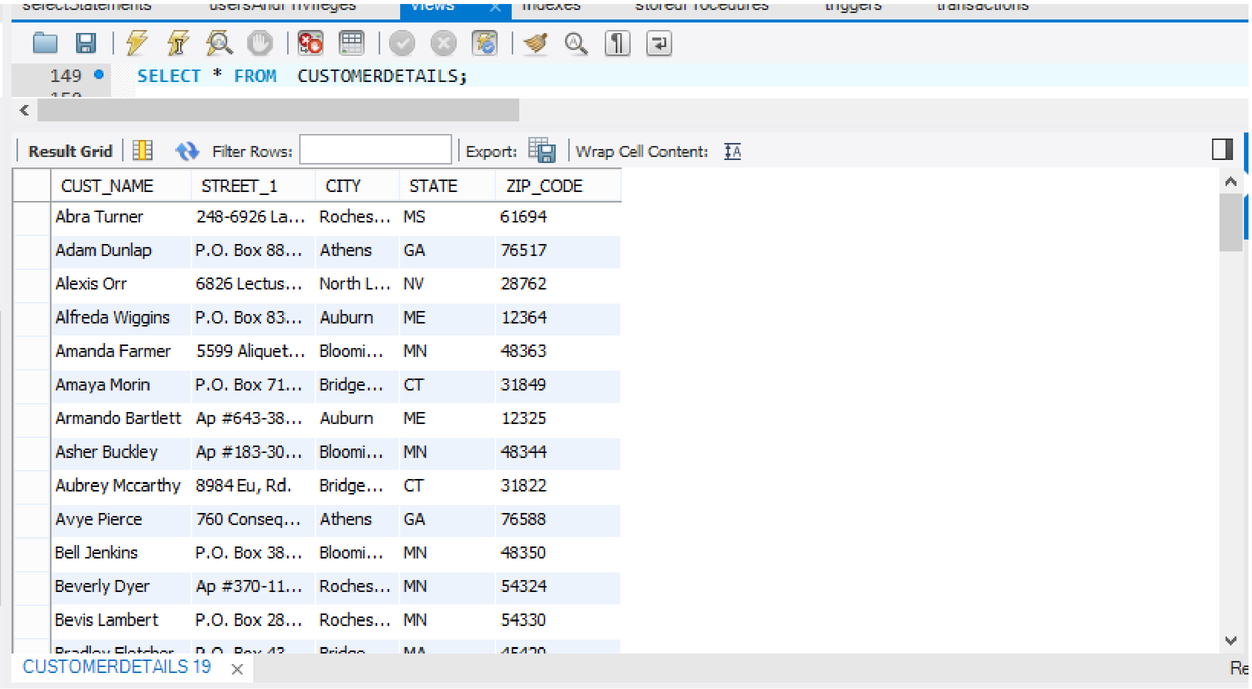
Enforcing 2NF & 3NF:

A table is in 2NF if it is in 1NF and the whole of the composite primary key is used to uniquely identify all the other non-key attributes. Attributes which have partial dependent on one part of the composite key alone are removed and formed as a separate table. A table is in 3NF if it is in 2NF and no non-key field depends on a key which is not a primary key. It overcomes the update and deletion anomalies



# Views

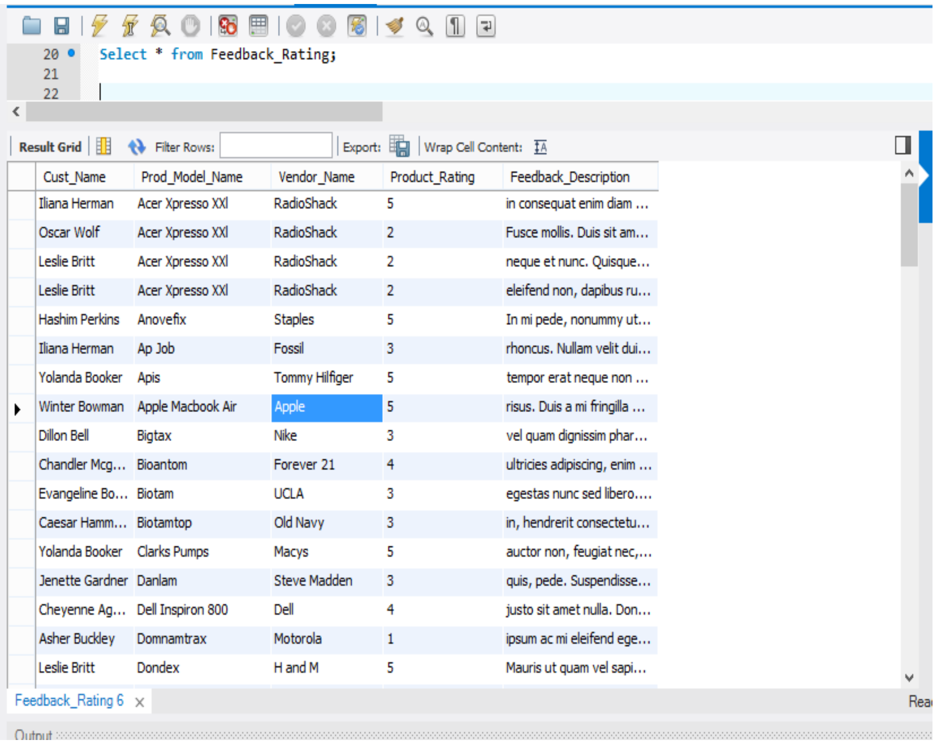
**Customer Details View**: This view will display all the customer details like name and address.



```
SELECT * FROM CUSTOMERDETAILS;
```

| CUST_NAME | STREET_1 | CITY | STATE | ZIP_CODE |
|---|---|---|---|---|
| Abra Turner | 248-6926 La... | Roches... | MS | 61694 |
| Adam Dunlap | P.O. Box 88... | Athens | GA | 76517 |
| Alexis Orr | 6826 Lectus... | North L... | NV | 28762 |
| Alfreda Wiggins | P.O. Box 83... | Auburn | ME | 12364 |
| Amanda Farmer | 5599 Aliquet... | Bloomi... | MN | 48363 |
| Amaya Morin | P.O. Box 71... | Bridge... | CT | 31849 |
| Armando Bartlett | Ap #643-38... | Auburn | ME | 12325 |
| Asher Buckley | Ap #183-30... | Bloomi... | MN | 48344 |
| Aubrey Mccarthy | 8984 Eu, Rd. | Bridge... | CT | 31822 |
| Avye Pierce | 760 Conseq... | Athens | GA | 76588 |
| Bell Jenkins | P.O. Box 38... | Bloomi... | MN | 48350 |
| Beverly Dyer | Ap #370-11... | Roches... | MN | 54324 |
| Bevis Lambert | P.O. Box 28... | Roches... | MN | 54330 |
| Bradley Fletcher | P.O. Box 43 | Bridge... | MA | 45420 |

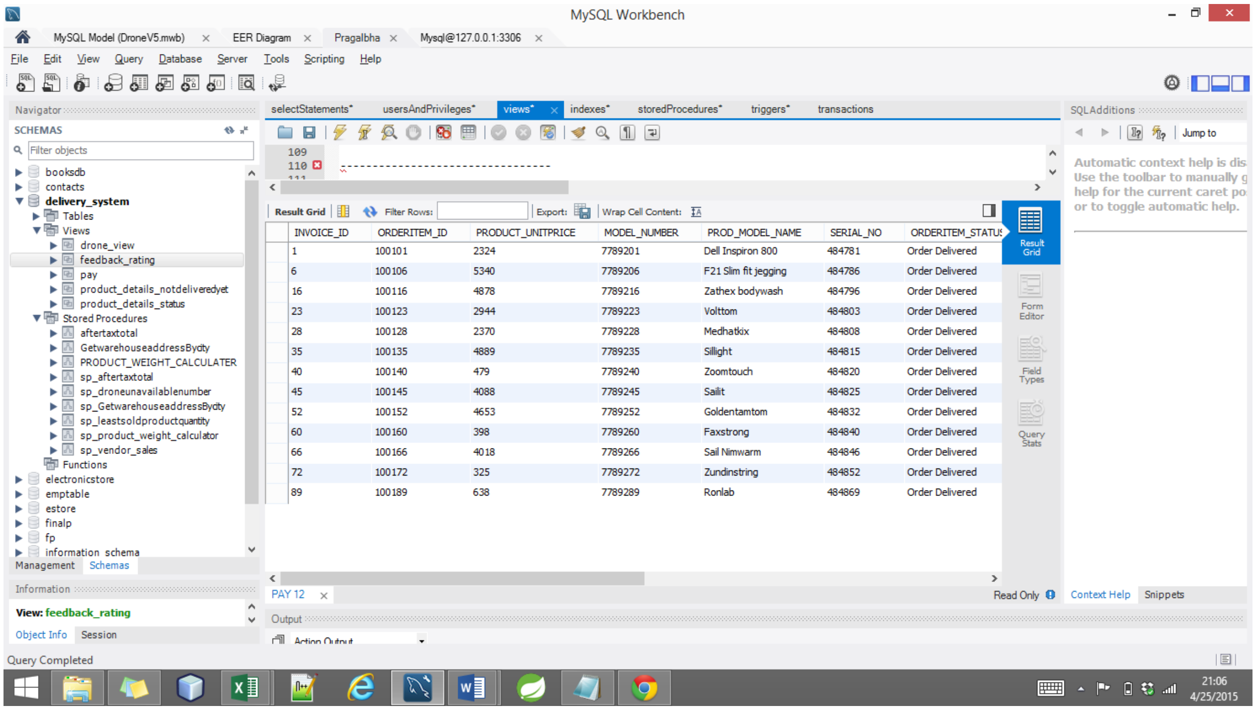
Feedback Rating View: It is a view which will allow the customers and vendors to see the feedback of the various products.



```sql
Select * from Feedback_Rating;
```

| Cust_Name | Prod_Model_Name | Vendor_Name | Product_Rating | Feedback_Description |
|---|---|---|---|---|
| Iliana Herman | Acer Xpresso XXl | RadioShack | 5 | in consequat enim diam ... |
| Oscar Wolf | Acer Xpresso XXl | RadioShack | 2 | Fusce mollis. Duis sit am... |
| Leslie Britt | Acer Xpresso XXl | RadioShack | 2 | neque et nunc. Quisque... |
| Leslie Britt | Acer Xpresso XXl | RadioShack | 2 | eleifend non, dapibus ru... |
| Hashim Perkins | Anovefix | Staples | 5 | In mi pede, nonummy ut... |
| Iliana Herman | Ap Job | Fossil | 3 | rhoncus. Nullam velit dui... |
| Yolanda Booker | Apis | Tommy Hilfiger | 5 | tempor erat neque non ... |
| Winter Bowman | Apple Macbook Air | Apple | 5 | risus. Duis a mi fringilla ... |
| Dillon Bell | Bigtax | Nike | 3 | vel quam dignissim phar... |
| Chandler Mcg... | Bioantom | Forever 21 | 4 | ultricies adipiscing, enim ... |
| Evangeline Bo... | Biotam | UCLA | 3 | egestas nunc sed libero.... |
| Caesar Hamm... | Biotamtop | Old Navy | 3 | in, hendrerit consectetu... |
| Yolanda Booker | Clarks Pumps | Macys | 5 | auctor non, feugiat nec,... |
| Jenette Gardner | Danlam | Steve Madden | 3 | quis, pede. Suspendisse... |
| Cheyenne Ag... | Dell Inspiron 800 | Dell | 4 | justo sit amet nulla. Don... |
| Asher Buckley | Domnamtrax | Motorola | 1 | ipsum ac mi eleifend ege... |
| Leslie Britt | Dondex | H and M | 5 | Mauris ut quam vel sapi... |

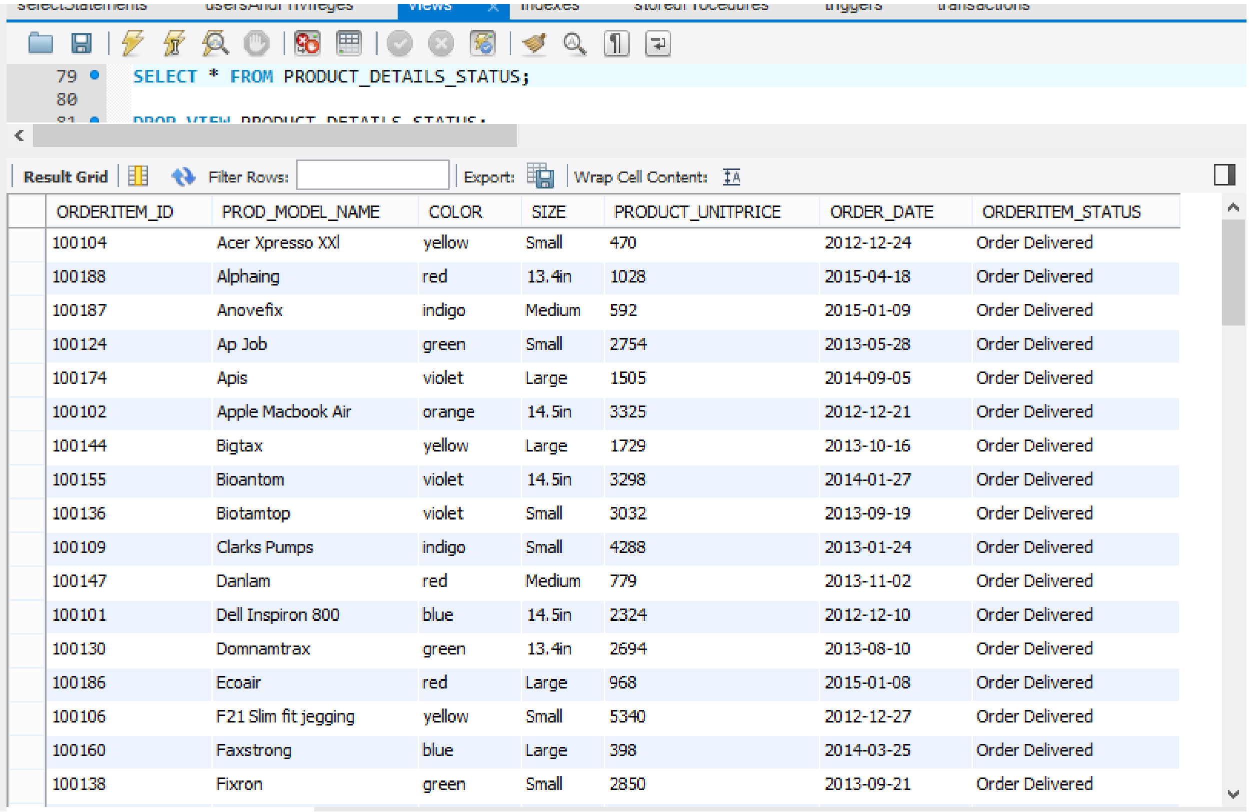
Payment View: This will include the customer details, card details , invoice details and product details.



```
109
110 ✖ --------------------------------
```

| INVOICE_ID | ORDERITEM_ID | PRODUCT_UNITPRICE | MODEL_NUMBER | PROD_MODEL_NAME | SERIAL_NO | ORDERITEM_STATUS |
|---|---|---|---|---|---|---|
| 1 | 100101 | 2324 | 7789201 | Dell Inspiron 800 | 484781 | Order Delivered |
| 6 | 100106 | 5340 | 7789206 | F21 Slim fit jegging | 484786 | Order Delivered |
| 16 | 100116 | 4878 | 7789216 | Zathex bodywash | 484796 | Order Delivered |
| 23 | 100123 | 2944 | 7789223 | Volttom | 484803 | Order Delivered |
| 28 | 100128 | 2370 | 7789228 | Medhatkix | 484808 | Order Delivered |
| 35 | 100135 | 4889 | 7789235 | Sillight | 484815 | Order Delivered |
| 40 | 100140 | 479 | 7789240 | Zoomtouch | 484820 | Order Delivered |
| 45 | 100145 | 4088 | 7789245 | Sailit | 484825 | Order Delivered |
| 52 | 100152 | 4653 | 7789252 | Goldentamtom | 484832 | Order Delivered |
| 60 | 100160 | 398 | 7789260 | Faxstrong | 484840 | Order Delivered |
| 66 | 100166 | 4018 | 7789266 | Sail Nimwarm | 484846 | Order Delivered |
| 72 | 100172 | 325 | 7789272 | Zundinstring | 484852 | Order Delivered |
| 89 | 100189 | 638 | 7789289 | Ronlab | 484869 | Order Delivered |

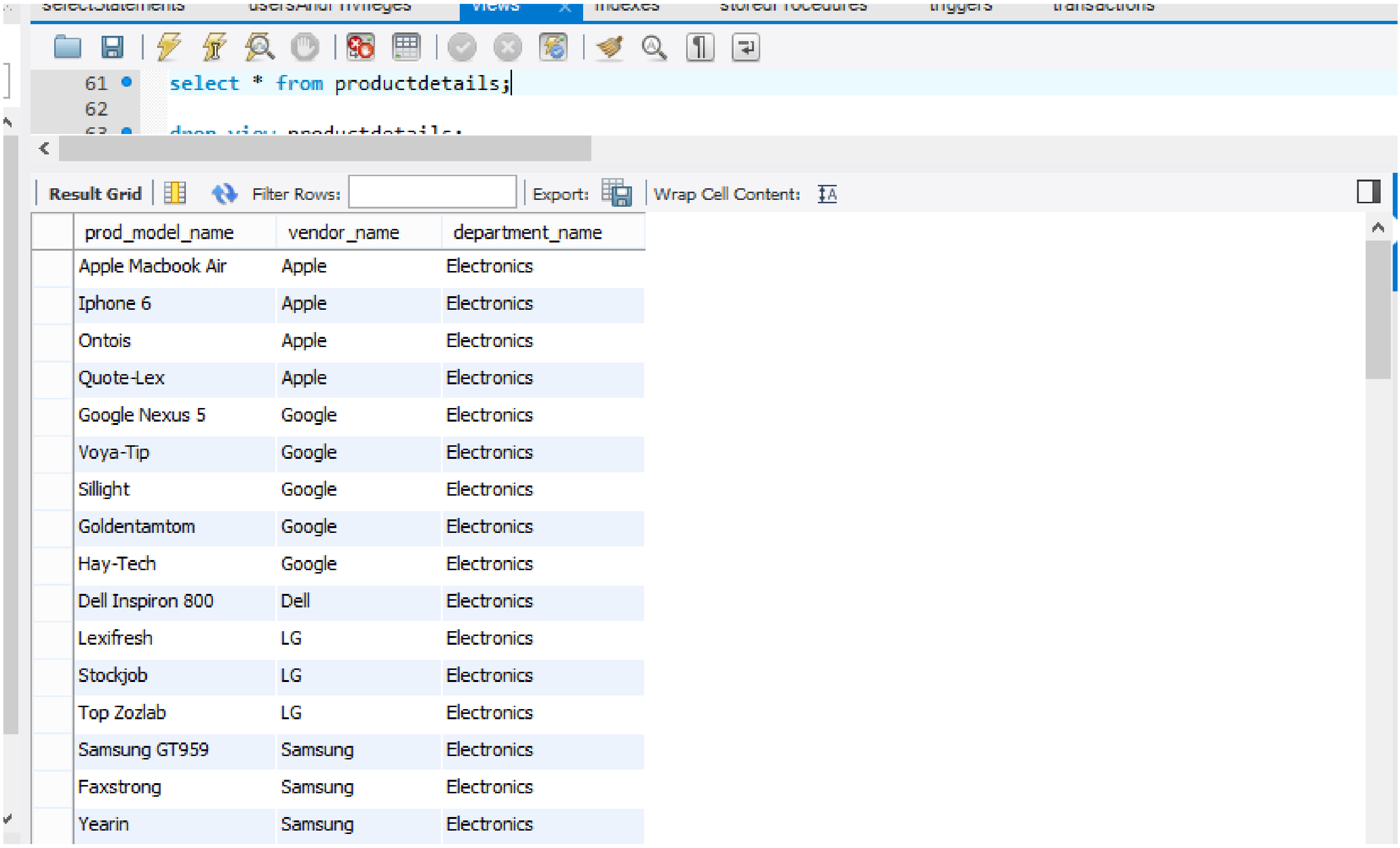
**Product Details Status View**: To check the status of the product.



```
SELECT * FROM PRODUCT_DETAILS_STATUS;

DROP VIEW PRODUCT_DETAILS_STATUS;
```

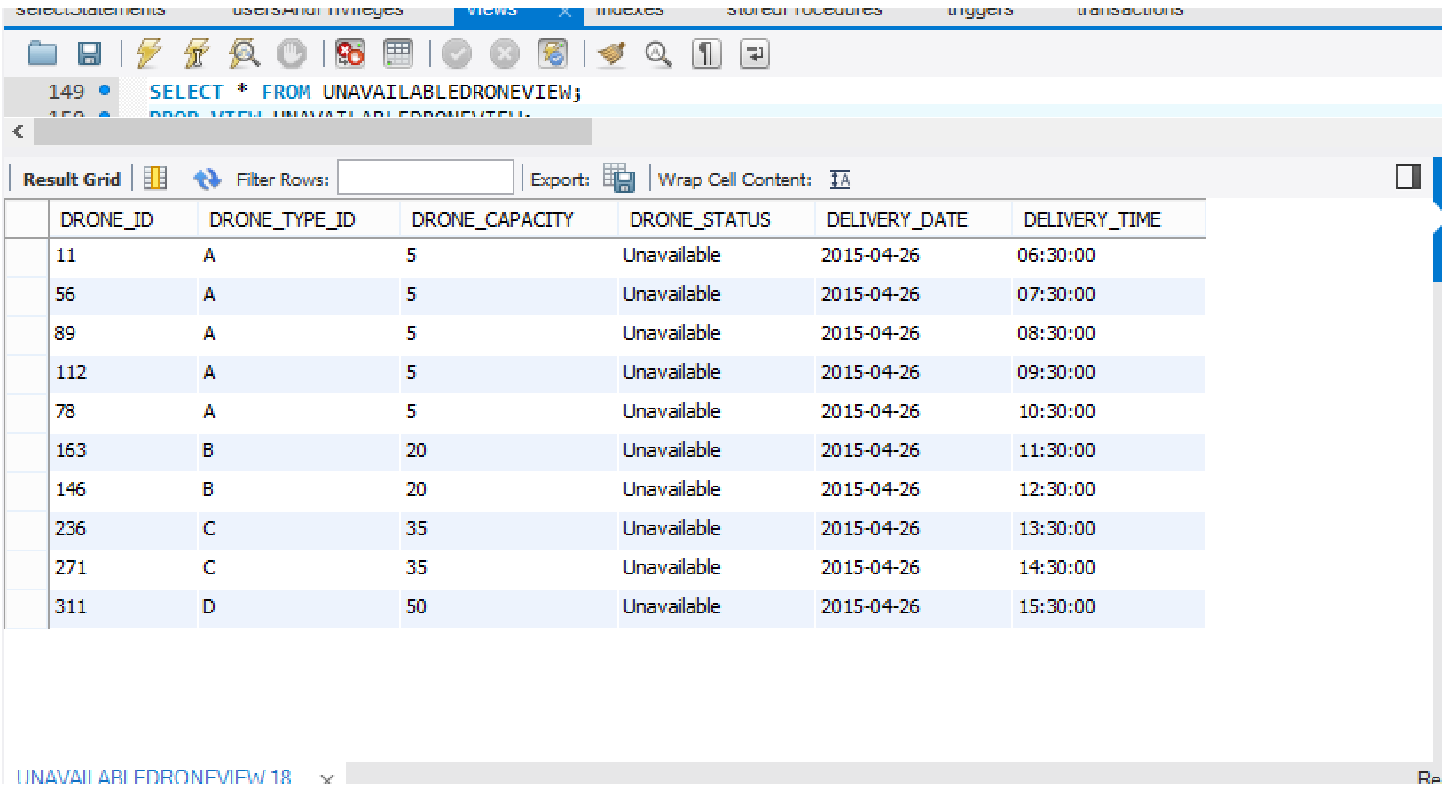
| ORDERITEM_ID | PROD_MODEL_NAME | COLOR | SIZE | PRODUCT_UNITPRICE | ORDER_DATE | ORDERITEM_STATUS |
|---|---|---|---|---|---|---|
| 100104 | Acer Xpresso XXl | yellow | Small | 470 | 2012-12-24 | Order Delivered |
| 100188 | Alphaing | red | 13.4in | 1028 | 2015-04-18 | Order Delivered |
| 100187 | Anovefix | indigo | Medium | 592 | 2015-01-09 | Order Delivered |
| 100124 | Ap Job | green | Small | 2754 | 2013-05-28 | Order Delivered |
| 100174 | Apis | violet | Large | 1505 | 2014-09-05 | Order Delivered |
| 100102 | Apple Macbook Air | orange | 14.5in | 3325 | 2012-12-21 | Order Delivered |
| 100144 | Bigtax | yellow | Large | 1729 | 2013-10-16 | Order Delivered |
| 100155 | Bioantom | violet | 14.5in | 3298 | 2014-01-27 | Order Delivered |
| 100136 | Biotamtop | violet | Small | 3032 | 2013-09-19 | Order Delivered |
| 100109 | Clarks Pumps | indigo | Small | 4288 | 2013-01-24 | Order Delivered |
| 100147 | Danlam | red | Medium | 779 | 2013-11-02 | Order Delivered |
| 100101 | Dell Inspiron 800 | blue | 14.5in | 2324 | 2012-12-10 | Order Delivered |
| 100130 | Domnamtrax | green | 13.4in | 2694 | 2013-08-10 | Order Delivered |
| 100186 | Ecoair | red | Large | 968 | 2015-01-08 | Order Delivered |
| 100106 | F21 Slim fit jegging | yellow | Small | 5340 | 2012-12-27 | Order Delivered |
| 100160 | Faxstrong | blue | Large | 398 | 2014-03-25 | Order Delivered |
| 100138 | Fixron | green | Small | 2850 | 2013-09-21 | Order Delivered |

**Product Details View**: To check the status of the product.



```
select * from productdetails;

drop view productdetails;
```

| prod_model_name | vendor_name | department_name |
|---|---|---|
| Apple Macbook Air | Apple | Electronics |
| Iphone 6 | Apple | Electronics |
| Ontois | Apple | Electronics |
| Quote-Lex | Apple | Electronics |
| Google Nexus 5 | Google | Electronics |
| Voya-Tip | Google | Electronics |
| Sillight | Google | Electronics |
| Goldentamtom | Google | Electronics |
| Hay-Tech | Google | Electronics |
| Dell Inspiron 800 | Dell | Electronics |
| Lexifresh | LG | Electronics |
| Stockjob | LG | Electronics |
| Top Zozlab | LG | Electronics |
| Samsung GT959 | Samsung | Electronics |
| Faxstrong | Samsung | Electronics |
| Yearin | Samsung | Electronics |



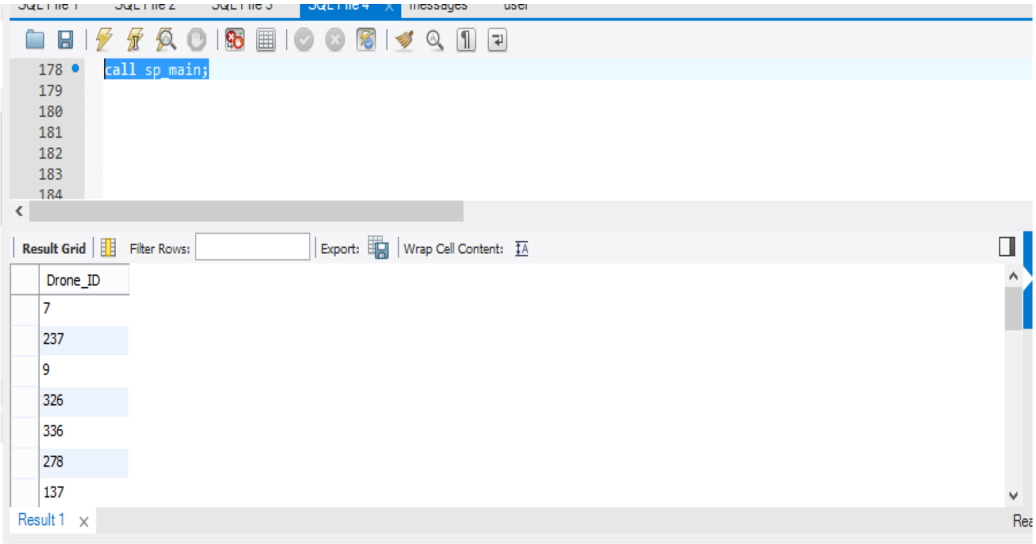
```
SELECT * FROM UNAVAILABLEDRONEVIEW;
DROP VIEW UNAVAILABLEDRONEVIEW;
```

| DRONE_ID | DRONE_TYPE_ID | DRONE_CAPACITY | DRONE_STATUS | DELIVERY_DATE | DELIVERY_TIME |
|---|---|---|---|---|---|
| 11 | A | 5 | Unavailable | 2015-04-26 | 06:30:00 |
| 56 | A | 5 | Unavailable | 2015-04-26 | 07:30:00 |
| 89 | A | 5 | Unavailable | 2015-04-26 | 08:30:00 |
| 112 | A | 5 | Unavailable | 2015-04-26 | 09:30:00 |
| 78 | A | 5 | Unavailable | 2015-04-26 | 10:30:00 |
| 163 | B | 20 | Unavailable | 2015-04-26 | 11:30:00 |
| 146 | B | 20 | Unavailable | 2015-04-26 | 12:30:00 |
| 236 | C | 35 | Unavailable | 2015-04-26 | 13:30:00 |
| 271 | C | 35 | Unavailable | 2015-04-26 | 14:30:00 |
| 311 | D | 50 | Unavailable | 2015-04-26 | 15:30:00 |

**Unavailable Drone View**: This view will display all the drones which are unavailable.

# Stored Procedures

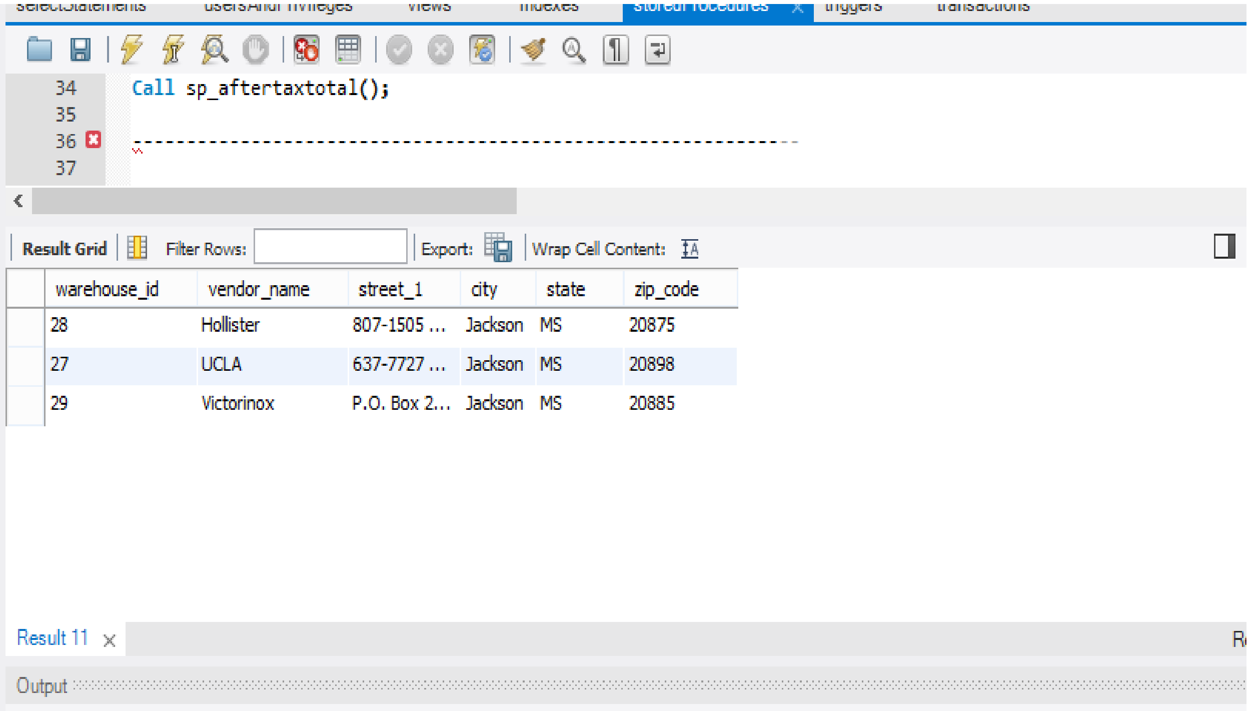
1. Main Procedure: A procedure to assign a Drone_Id whenever an order is placed by a customer. This takes into consideration the package weight and compares it with the drone capacity to find an appropriate match with the drones that are available in the warehouse.



```
call sp_main;
```

| Drone_ID |
|---|
| 7 |
| 237 |
| 9 |
| 326 |
| 336 |
| 278 |
| 137 |

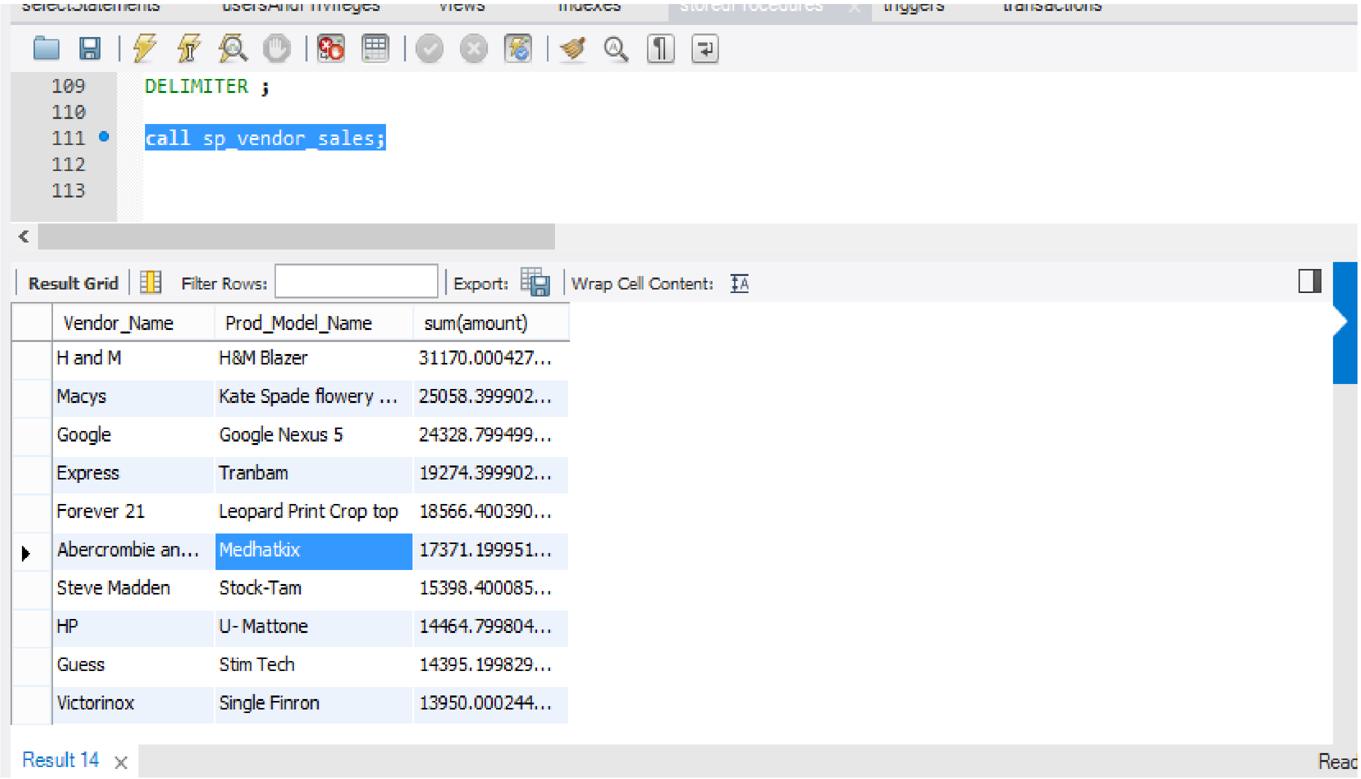
2. AfterTaxTotal: A procedure to get total amount invoice by adding tax to the product unit price.



```
Call sp_aftertaxtotal();

-------------------------------------------------------------------
```

| warehouse_id | vendor_name | street_1 | city | state | zip_code |
|---|---|---|---|---|---|
| 28 | Hollister | 807-1505 ... | Jackson | MS | 20875 |
| 27 | UCLA | 637-7727 ... | Jackson | MS | 20898 |
| 29 | Victorinox | P.O. Box 2... | Jackson | MS | 20885 |

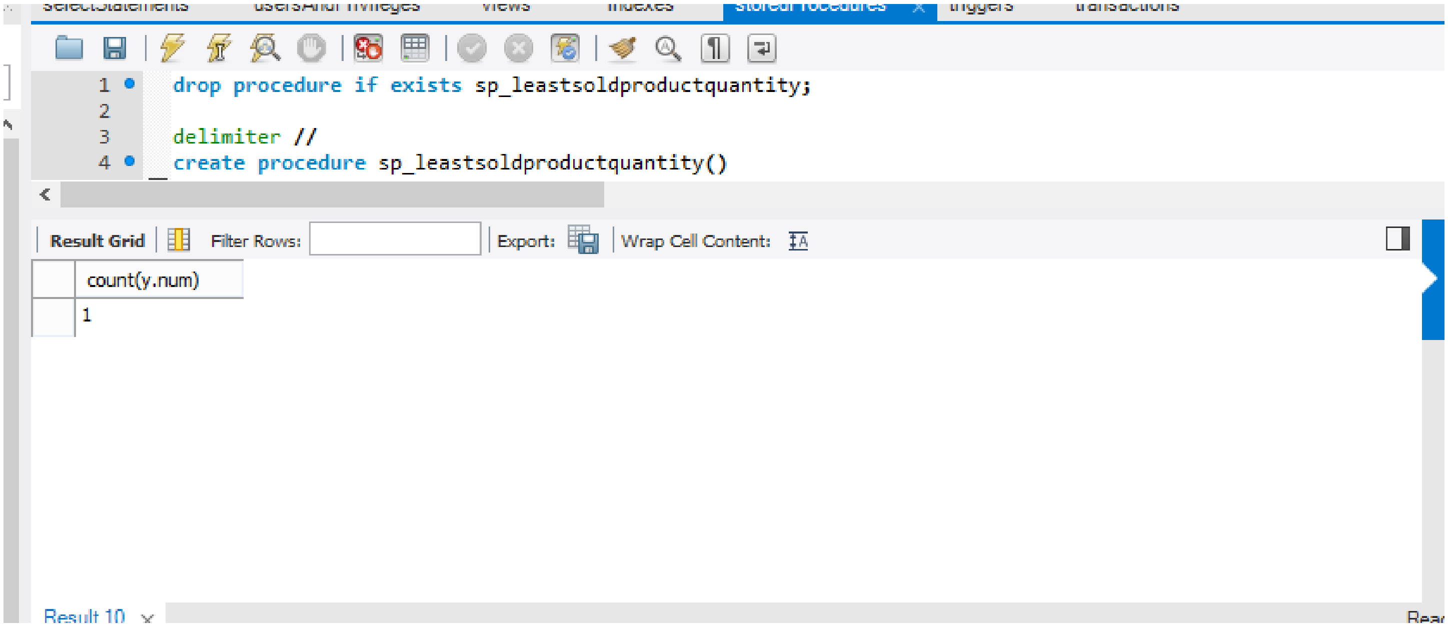
3. Vendor Sales: A procedure to find the top ten vendors with the product model that has the maximum sales.



```
DELIMITER ;

call sp_vendor_sales;


```

| Vendor_Name | Prod_Model_Name | sum(amount) |
|---|---|---|
| H and M | H&M Blazer | 31170.000427... |
| Macys | Kate Spade flowery ... | 25058.399902... |
| Google | Google Nexus 5 | 24328.799499... |
| Express | Tranbam | 19274.399902... |
| Forever 21 | Leopard Print Crop top | 18566.400390... |
| Abercrombie an... | Medhatkix | 17371.199951... |
| Steve Madden | Stock-Tam | 15398.400085... |
| HP | U-Mattone | 14464.799804... |
| Guess | Stim Tech | 14395.199829... |
| Victorinox | Single Finron | 13950.000244... |

4. Least Sold Product Quantity: A procedure to find the least sold product

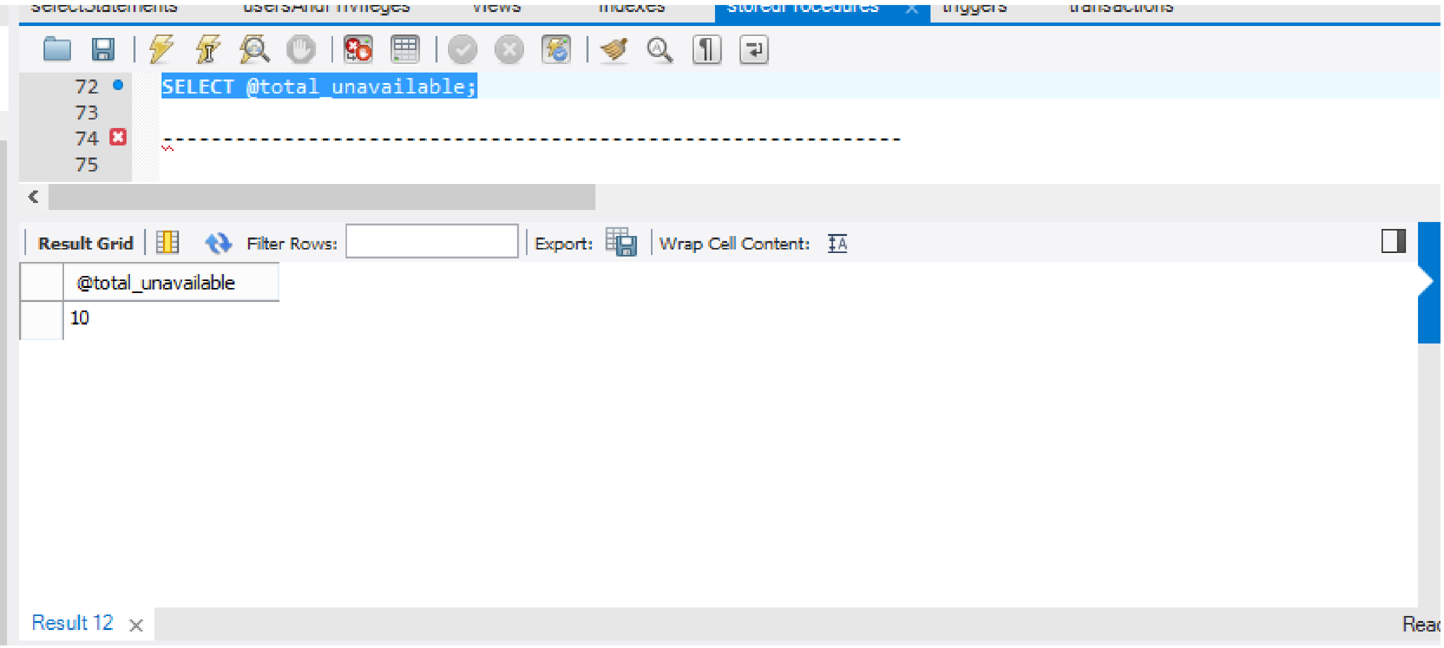


```
drop procedure if exists sp_leastsoldproductquantity;

delimiter //
create procedure sp_leastsoldproductquantity()
```

| count(y.num) |
|---|
| 1 |

# Parametric Procedures

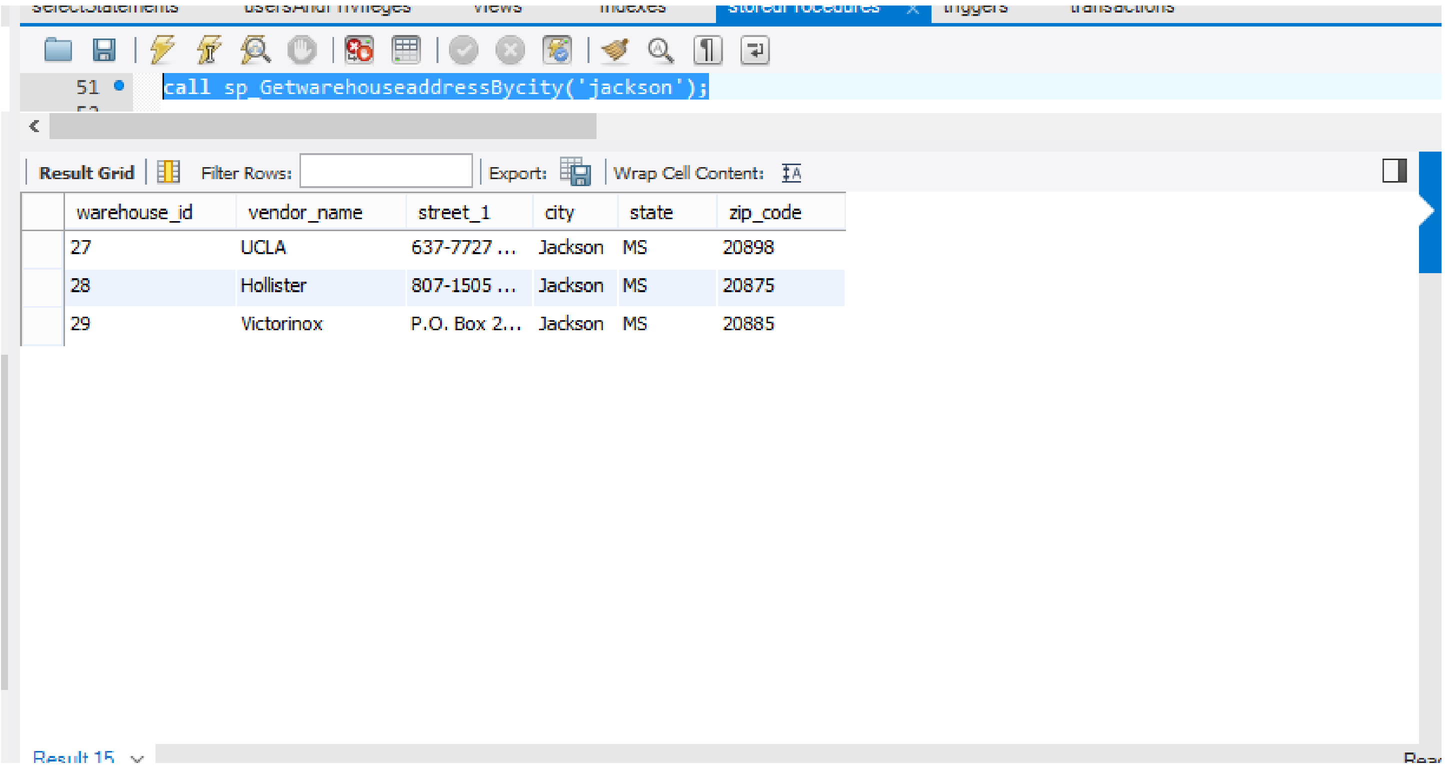
1. Drone Unavailable Number: Parametric OUT procedure to count the number of drones with a particular status.



```
SELECT @total_unavailable;

----------------------------------------------------------------
```

| | @total_unavailable |
|---|---|
| | 10 |

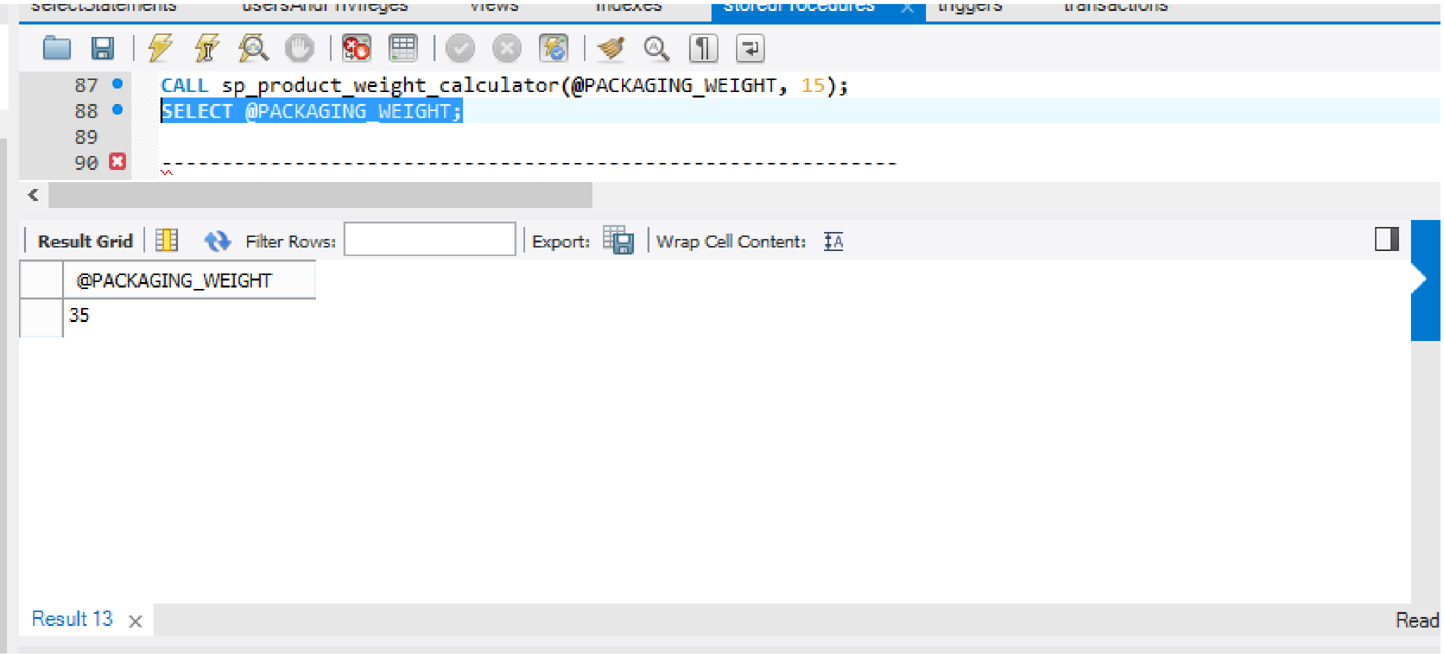
2. Get Warehouse Address by City: Parametric IN procedure to get warehouse address by inputting a particular city.



```
call sp_GetwarehouseaddressBycity('jackson');
```

| warehouse_id | vendor_name | street_1 | city | state | zip_code |
|---|---|---|---|---|---|
| 27 | UCLA | 637-7727 ... | Jackson | MS | 20898 |
| 28 | Hollister | 807-1505 ... | Jackson | MS | 20875 |
| 29 | Victorinox | P.O. Box 2... | Jackson | MS | 20885 |

3. Product Weight Calculator: Parametric INOUT procedure to calculate approximate package weight.



```sql
CALL sp_product_weight_calculator(@PACKAGING_WEIGHT, 15);
SELECT @PACKAGING_WEIGHT;

---------------------------------------------------------
```

```
@PACKAGING_WEIGHT
35
```

# TRIGGERS

1. **A trigger to reduce the quantity of the Product_model when an item is inserted into the order_item table. This reduces the total quantity available by 1.**

```
    Delimiter //
Create trigger reducequantity after insert on order_item for each row
begin
update product_model pm inner join product_unit pu on
pm.Model_number=pu.Model_Number
inner join order_item oi on oi.serial_no= pu.serial_no
set pm.total_quantity= pm.total_quantity-1 ;
end //

drop trigger reducequantity;
```

2. **A trigger to find the average feedback rating in product_model, when a feed feedback is entered in the feedback table.**

```
Delimiter //
Create trigger averagefeedback after insert on feedback for each row
begin
update
product_model pm
inner join feedback fb
on
pm.Model_number=fb.Model_Number
set pm.average_rating= num where
num=(sum(fb.product_rating)/count(fb.product_rating));

end //
```

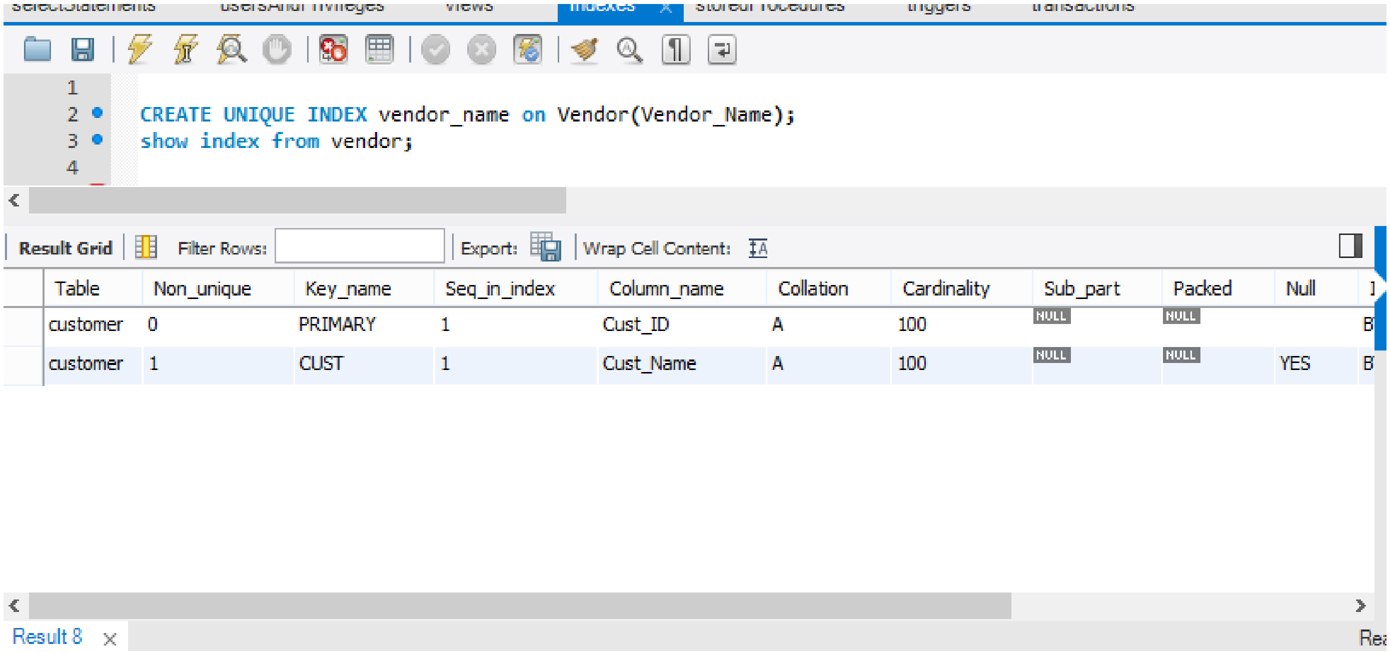
# TRANSACTIONS

- In almost all applications that access SQL databases, multiple users concurrently attempt to view and modify data. The simultaneous operations may result in data that is inconsistent and inaccurate –Or worse, database corruption. Using “Transactions” avoids these problems by isolating each operation.
- Transaction is a set of one or more SQL statements that perform a set of related actions.
- The statements are grouped together and treated as a single unit whose success or failure depends on the successful execution of each statement in the transaction.

```
start transaction;
savepoint savepoint1;
select @OrderItem_ID:=max(OrderItem_ID) from order_item;
set @OrderItem_ID=@OrderItem_ID+1;

insert into
order_item(OrderItem_ID,OrderItem_Status,Order_Date,Package_weight,Serial_No,Drone_ID)
values (@OrderItem_ID,"Order InProcess",now(),3.79,484784,12);

rollback to savepoint savepoint1;

insert into
order_item(OrderItem_ID,OrderItem_Status,Order_Date,Package_weight,Serial_No,Drone_ID)
values(@OrderItem_ID,"Order InProcess",now(),6.78,484786,15);
commit;
```

# INDEXES

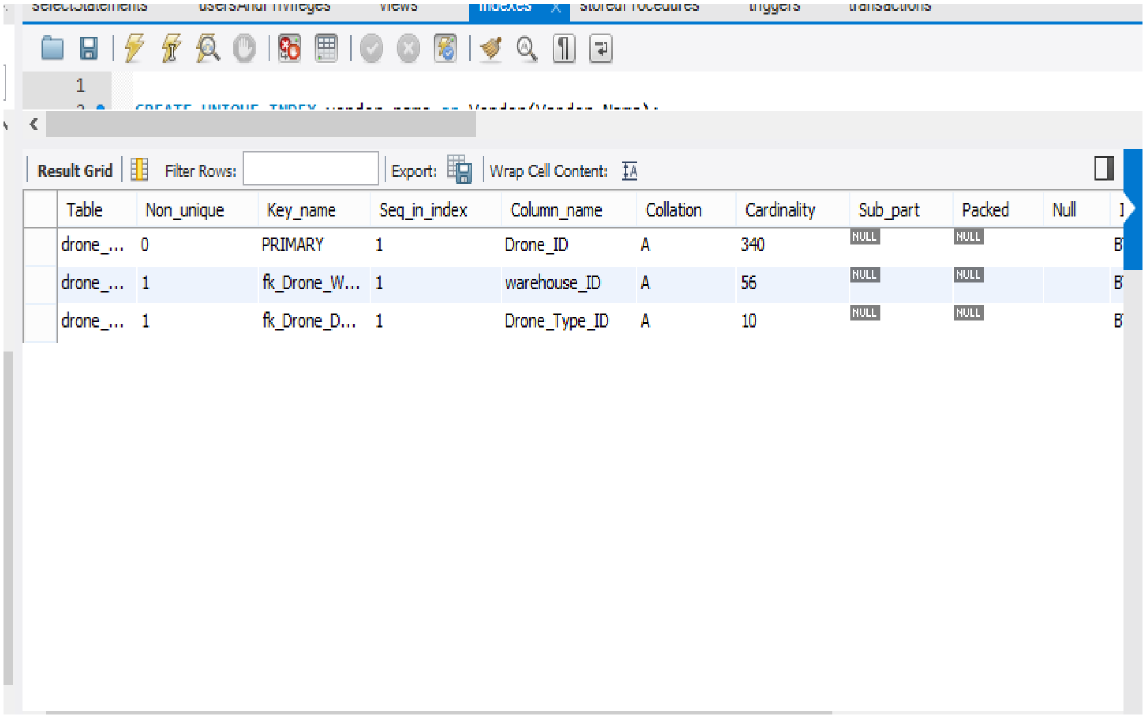
Customer Index: This is an Index on the Customer table using Customer name.



```
CREATE UNIQUE INDEX vendor_name on Vendor(Vendor_Name);
show index from vendor;
```

| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | I |
|---|---|---|---|---|---|---|---|---|---|---|
| customer | 0 | PRIMARY | 1 | Cust_ID | A | 100 | NULL | NULL |  | B |
| customer | 1 | CUST | 1 | Cust_Name | A | 100 | NULL | NULL | YES | B |

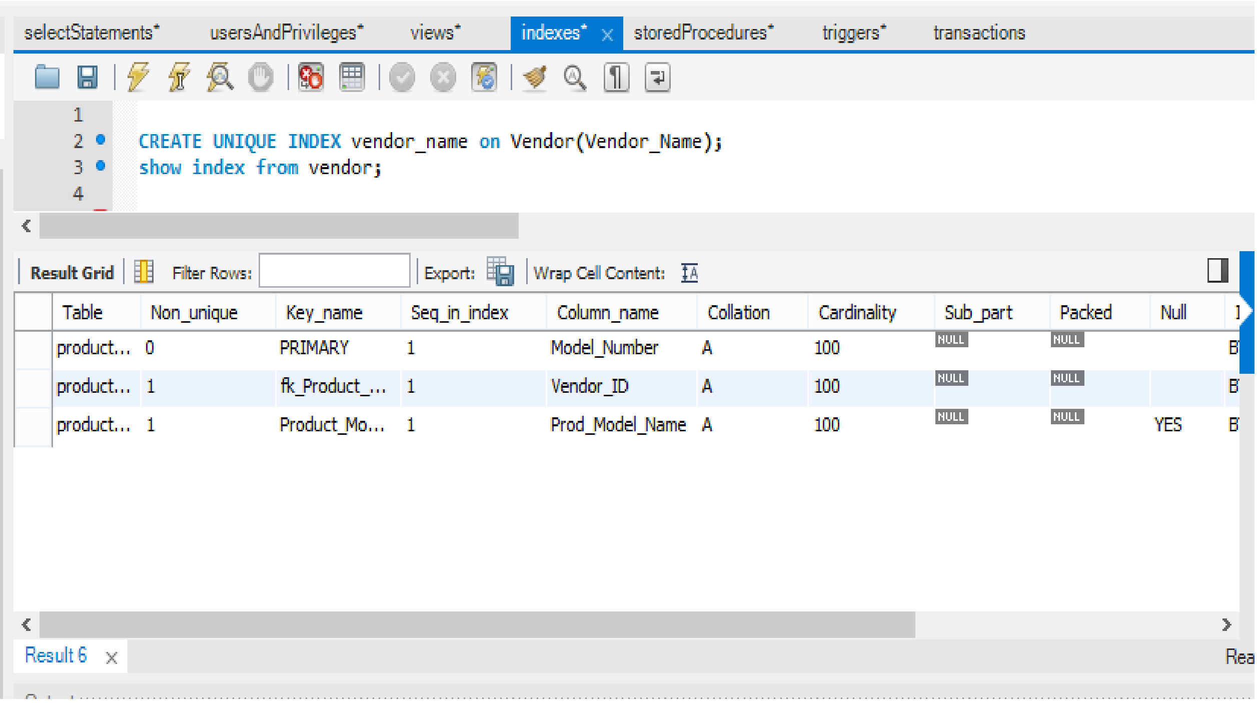
Drone Details: This is an Index on the Drone_Details table using Drone_ID .



```
CREATE UNIQUE INDEX vendor_name on Vendor(Vendor_Name);
```

| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | I |
|---|---|---|---|---|---|---|---|---|---|---|
| drone_... | 0 | PRIMARY | 1 | Drone_ID | A | 340 | NULL | NULL | | B |
| drone_... | 1 | fk_Drone_W... | 1 | warehouse_ID | A | 56 | NULL | NULL | | B |
| drone_... | 1 | fk_Drone_D... | 1 | Drone_Type_ID | A | 10 | NULL | NULL | | B |

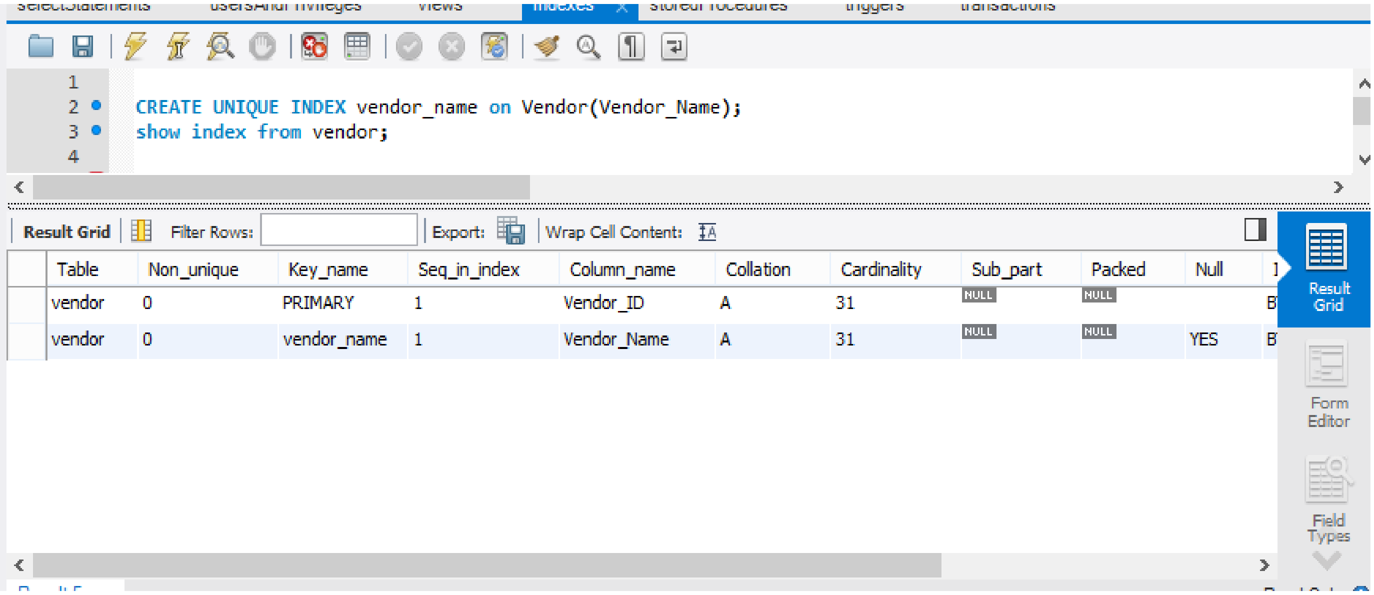
Product Model: This is an Index on the Product table using Product_Model_name.



```

CREATE UNIQUE INDEX vendor_name on Vendor(Vendor_Name);
show index from vendor;

```

| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | I |
|---|---|---|---|---|---|---|---|---|---|---|
| product... | 0 | PRIMARY | 1 | Model_Number | A | 100 | NULL | NULL |  | B |
| product... | 1 | fk_Product_... | 1 | Vendor_ID | A | 100 | NULL | NULL |  | B |
| product... | 1 | Product_Mo... | 1 | Prod_Model_Name | A | 100 | NULL | NULL | YES | B |

Vendor: This is an Index on the Vendor table using Vendor_Name.



```
CREATE UNIQUE INDEX vendor_name on Vendor(Vendor_Name);
show index from vendor;
```

| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | I |
|---|---|---|---|---|---|---|---|---|---|---|
| vendor | 0 | PRIMARY | 1 | Vendor_ID | A | 31 | NULL | NULL |  | B |
| vendor | 0 | vendor_name | 1 | Vendor_Name | A | 31 | NULL | NULL | YES | B |

# USERS AND PRIVILEGES

MySQL is an open source database management software that helps users store, organize and later retrieve data. It has a variety of options to grant specific users nuanced privileges within the tables and databases.

**Users:**

- Ecommerce Admin:
- Drone Manager:
- Vendor:
- Customer:

**Privileges:**

- Create: allows them to create new tables or databases
- Insert: allows them to insert rows into tables
- Select: allows them to use the select command to read through databases
- Update: allow them to update table rows.
- Drop: allows them to delete rows from tables.

# BACKUPS

Our Database implements two kinds of backup

- Full Backup
- Incremental Backup

- Since it is an e-commerce website, we have assumed that there is least traffic during the night.
- We have also assumed that there is lesser traffic during the weeknights rather than weekend nights.
- We assume that full backup takes 6 hours to complete.
- We assume that an incremental backup takes about 3 hours to complete.
- If full backup is scheduled on public holidays (Christmas, Thanksgiving) then it will be completed on the previous day as well as the day after the holiday.

| Day of the week | Type of Backup | Timing |
|---|---|---|
| Sunday | Incremental Backup | 1:30 am-4:30 am |
| Monday | Full Backup | 12 am-6 am |
| Tuesday | Incremental Backup | 1:30 am-4:30 am |
| Wednesday | Incremental Backup | 1:30 am-4:30 am |
| Thursday | Incremental Backup | 1:30 am-4:30 am |
| Friday | Full Backup | 12 am- 6 am |
| Saturday | Incremental Backup | 1:30 am-4:30 am |

# CONCLUSION AND FUTURE SCOPE

Thus we have achieved the main objective of our project to build a database for a typical Ecommerce website which uses Drones to deliver orders placed by Customers, with speed and accuracy. We have completely normalized all our tables in the database. We have performed all the operations on these normalized tables. We have successfully implemented concepts of Triggers, Procedures, Views, Indexes and Back-up and Recovery taught in the class. We have also integrated Business Intelligence concepts.

There are still few functionalities which we can work on in the future:

- We would want to implement an algorithm to find shortest path for the drones and track the drone path during delivery and save this path in the database.
- We will perform more Business Intelligence to make optimum use of the database and analyze this data well for better business.

# REFERENCES

https://dev.mysql.com/

http://www.mysqltutorial.org/

http://stackoverflow.com/

http://www.w3schools.com/

http://generatedata.com/

http://www.wikipedia.com/

*Books:*

MySQL Documentation

Modern Database Management

Beginning SQL

Presentation slides by Prof. Mutsalklisana