**Submitted to:**

**Rezwan Ahmed**

assistant professor

Department of computer Science

American International University – bangladesh

**Submitted by:**

1. *Riyad , MD Ahsan Ferdous*

*ID:**15-29804-2*

1. Ahmed , Md ShaonID: 15-30027-2
2. Roy , Kamal KrishnaID: 15-29788-2
3. Saied , Ahmed FoyezID : 16-31496-1

**Section: G**

E-COmmerce website

Database Model

**Table of contents**

**Part - A**

1. Business and system summary ...................................................................................1
2. Business overview
3. Overview .............................................................................................2
4. Project Summary .................................................................................2
5. Technical summary of super shop management system database
6. List of Entities of the model .................................................................3
7. List of attributes of the model .............................................................3
8. List of constraints according to table....................................................3
9. SQL statement to create sample table and insert sample values ........4
10. Identifying objective & describing the database working
11. Users services ......................................................................................9
12. Administrator services .........................................................................9
13. Entity – Relationship diagram (ER Diagram) ...............................................................10
14. Normalization .............................................................................................................11
15. Relationship diagram ..................................................................................................12
16. Screenshots of the table with only data type of the attributes....................................13
17. Screenshot of the tables with sample data records ....................................................15
18. Demonstrating some database use scenarios and SQL queries...................................18
19. View creation of the table
20. Complex view ......................................................................................21
21. Simple view .........................................................................................22

**Part - B**

Summarizing learning experience from database project ..............................................24

* **Part - A**

**Business and system summary**

This ‘Super shop management database system’ will help a shop itself better and also help showing better service can a shop provide using this database system. It helps us to understand how a customer, salesman and product information is stored in a super shop information system, what types of products are taken by customer, how the system manages the relation with customer and salesman, how a product’s information are enlisted in the shop. How commissions are given to salesman are also provided by this system. The system also provides the total amount of shopping by a customer with goods information. Salesman are provided to assist customer to buy goods. Salesman can access into the system of super shop management by creating data into this system. It’s also provides financial records specific to a single component, or groups of components.

**Business overview**

**Overview:** Super shop management is primarily about specifying the size and placement of stocked goods. Super shop management is required at different locations within a facility or within multiple locations of a supply network to protect the regular and planned course of production against the random disturbance of running out of materials or goods.

The way super shop works is not very complicated. Shop has to be the capability to keep information about customer and employee. Shop, therefore, has the most important component and hence they are customer, employee, products and shop itself. Shop has a big collection of inventory. The inventory can contain different type of products.

But the most important thing about Shop is that the system can get much more complicated if it is not organized and thus, therefore, it is necessary to create a management system that can organize the data and help the shop to be organized and work properly.

**Project Summary:** Super shop management system is not really much complicated but the problem is having huge number of customer and there is so many new customer coming into the shop every day or week. Every customer has a separate identity. New employee may come to the shop and some employee may leave the shop.

The shop has so many products. The products have to be identified as product type. Salesman got commission by selling a fixed amount.

In every month there is a necessity to take summarizing of the sell to identify the profit and to keep up the information.

**Technical summary of super shop management system database**

The database system that prepared in this project can be accessed via a software program or a web-server based system using internet access or as an embedded data storage system. The database system and the project was made using Oracle SQL Express 11g, Microsoft Word. All the tables of entity and attributes were made using Oracle SQL Express 11g.The database system is very simple in design and to implement. This system requires very low system resources and this system will work in almost all configurations.

**List of Entities of the model:**

* Admin
* Seller
* User
* Product
* Supply\_Order
* Order
* Promo
* Cart
* Category

**List of attributes of the model:**

* **Admin’s Attributes** :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Attributes** | A\_id | A\_password | A\_address | A\_email | A\_mobile | U\_status | A\_type |
| **Data Type** | number(5) | varchar2(15) | varchar2(30) | varchar2(30) | number(11) | varchar2(30) | varchar2(30) |

* **Seller’s Attributes:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Attributes** | S\_id | S\_password | S\_address | S\_email | S\_mobile | U\_status | S\_type |
| **Data Type** | number(5) | varchar2(30) | varchar2(30) | varchar2(30) | number(11) | varchar2(30) | varchar2(30) |

* **G\_user’s Attributes:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Attributes** | Gu\_id | Gu\_password | Gu\_address | Gu\_email | Gu\_mobile | u\_status | Gu\_type |
| **Data Type** | number(5) | varchar2(15) | varchar2(30) | varchar2(30) | number(11) | varchar2(30) | varchar2(30) |

* **Product’s attributes :**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Attributes** | product\_id | product\_name | product\_price | product\_avlble | product\_sell\_price | product\_orginal \_price | Category\_id |
| **Data Type** | number(5) | varchar2(30) | varchar2(30) | varchar2(30) | number(5) | number(5) | number(5) |

* **Supply order attributes :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | Supply\_id | Supply\_date | Supply\_status | S\_id |
| **Data Type** | number(5) | number(10) | varchar2(30) | number(5) |

* **Cart’s attributes (updated):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | cart\_id | cart\_status | user\_id | Gu\_type | Order\_id |
| **Data Type** | number(5) | varchar2(30) | number(5) | varchar2(30) | number(5) |

* **Order\_t attributes :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | order\_id | order\_date | payment\_method | payment\_status | return\_id |
| **Data Type** | number(5) | number(20) | varchar2(30) | varchar2(30) | number(5) |

* **Categories attributes :**

|  |  |  |
| --- | --- | --- |
| **Attributes** | Category\_id | Category\_name |
| **Data Type** | number(5) | varchar2(30) |

* **Promo\_use attributes :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | Promo\_id | user\_id | user\_type | user\_use\_date | user\_use\_account |
| **Data Type** | number(5) | number(5) | varchar2(30) | number(22) | number(20) |

**List of constraints according to table:**

|  |  |  |
| --- | --- | --- |
| **Table Name** | **Constraint Name** | **Constraint Attribute** |
| G\_user | G\_user\_pk | g\_u\_id |
| Seller | Seller\_pk | S\_id |
| Admin | Admin\_pk | A\_id |
| Orders | Orders\_pk | Order\_ID |
| Products | Products\_pk | p\_ID |
| Supply\_order | supply\_order\_pk | supply\_id |
| cart | cart\_pk | cart\_id |
| Category | category\_pk | category\_id |
| promo | promo\_pk | promo\_id |

**SQL statement to create sample table and insert sample values:**

create table Admin

(

a\_id number(5),

a\_password varchar2(15),

a\_address varchar2(30) not null,

a\_email varchar2(30) unique,

a\_mobile number(11) not null,

u\_status varchar2(30) not null,

a\_type varchar2(30),

CONSTRAINT Admin\_pk PRIMARY KEY (a\_id),

CONSTRAINT Admin\_unique unique (a\_type)

);

create table Seller

(

s\_id number(5),

s\_password varchar2(30),

s\_address varchar2(30),

s\_email varchar2(30) unique,

s\_mobile number(11) unique,

u\_status varchar2(30) not null,

s\_type varchar2(30),

constraint Seller\_pk primary key(s\_id),

constraint Seller\_unique unique(s\_type)

);

create table G\_user

(

g\_u\_id number(5) primary key,

g\_u\_password varchar2(15),

g\_u\_address varchar2(30) not null,

g\_u\_email varchar2(30) unique,

g\_u\_mobile number(11) not null,

u\_status varchar2(30) not null,

g\_u\_type varchar2(30) unique

);

create table Promo

(

promo\_id number(5),

promo\_desc varchar2(30),

Promo\_expiry varchar2(30) not null,

promo\_percentage number(20),

promo\_status varchar2(20),

promo\_limit varchar2(30),

promo\_use\_count number(10),

a\_id number(5),

CONSTRAINT Promo\_pk PRIMARY KEY (promo\_id),

CONSTRAINT a\_idi\_fk FOREIGN KEY (a\_id) REFERENCES Admin(a\_id)

);

create table Categories

(

category\_id number(5),

category\_name varchar2(30),

CONSTRAINT Categories\_pk PRIMARY KEY (category\_id)

);

create table Products

(

product\_id number(5),

product\_name varchar2(30) ,

product\_price varchar2(30) unique,

product\_avlble varchar2(30) not null,

product\_sell\_price number(5) unique,

product\_original\_price number(5) unique,

category\_id number(5),

CONSTRAINT Products\_pk PRIMARY KEY (product\_id),

CONSTRAINT categoryid10\_fk FOREIGN KEY (category\_id) REFERENCES Categories(category\_id)

);

create table Order\_t

(

order\_id number(5),

order\_date number(20),

payment\_method varchar2(30),

payment\_status varchar2(30),

return\_id number(5),

CONSTRAINT Order\_t\_pk PRIMARY KEY (order\_id),

CONSTRAINT returnid\_fk FOREIGN KEY (return\_id) REFERENCES Return\_t(return\_id)

)

create table Cart

(

cart\_id number(5),

cart\_status varchar2(30) unique,

user\_id number(5),

g\_u\_type varchar2(30),

order\_id number(5),

CONSTRAINT Cart\_pk PRIMARY KEY (cart\_id),

CONSTRAINT userid1\_fk FOREIGN KEY (user\_id) REFERENCES Promo\_use(user\_id),

CONSTRAINT gutype1\_fk FOREIGN KEY ( g\_u\_type) REFERENCES G\_user( g\_u\_type),

CONSTRAINT orderid1\_fk FOREIGN KEY (order\_id) REFERENCES Order\_t(order\_id)

);

create table Supply\_order

(

supply\_id number(5),

supply\_date number(10) not null,

supply\_status varchar2(30),

s\_id number(5),

CONSTRAINT Supply\_order\_pk PRIMARY KEY (supply\_id),

CONSTRAINT sid3\_fk FOREIGN KEY (s\_id) REFERENCES Seller(s\_id)

);

create table review

(

review\_id number(5),

review\_text varchar2(30),

review\_status varchar2(30),

review\_date number(10) not null,

product\_id number(5),

CONSTRAINT review\_pk PRIMARY KEY (review\_id),

CONSTRAINT productid4\_fk FOREIGN KEY (product\_id) REFERENCES Products(product\_id)

);

insert into Admin values('2','apple1','sirajgonj','nishi@gmail.com','01111','status','admin1')

insert into Seller values('1','orange','tangail','jannat@gmail.com','019437438','none','user')

insert into G\_user values('1','orange2','bogra1','afrina1@gmail.com','019432438','none','user')

insert into G\_user values('2','orange3','bogra4','afrina13@gmail.com','0194432438','none','generaluser')

insert into Promo values('1','pri','date over','50','per','limited','34','1')

insert into Promo values('2','pro','date over','58','pre','limited','39','2')

insert into Categories values('1','first')

insert into Categories values('2','last')

insert into Products values('1','shoap','100','available','80','70','1')

insert into Products values('2','shampoo','200','available','100','90','2')

insert into p\_include\_wishlist values('1','1','1')

insert into p\_include\_wishlist values('2','2','10')

insert into Order\_t values('1',to\_date('09-09-18','dd-mm-yyyy'),'cash','pending','1')

insert into Order\_t values('2',to\_date('09-10-18','dd-mm-yyyy'),'cash','pending','2')

insert into Cart values('1','user','1','user','1')

insert into Cart values('2','user1','2','generaluser','2')

insert into Supply\_order values('1',to\_date('09-06-18','dd-mm-yyyy'),'user','1')

insert into Supply\_order values('2',to\_date('09-05-18','dd-mm-yyyy'),'user1','2')

insert into review values('1','post','user',to\_date('09-05-18','dd-mm-yyyy'),'1')

insert into review values('2','post','user1',to\_date('02-05-18','dd-mm-yyyy'),'2')

Thus how the database system was created by following above pattern. The data types are shown in the table charts and all entities and attributes names as well. This database is following joining function of DBMS between the tables and the tables are linked with each other with the foreign keys. By these foreign keys these tables are connected with each other. In those tables any suitable and eligible type of data and especially appropriate data can be stored or modify.

**Identifying objective & describing the database working**

Our objective is to make a website for a company to start an E-Commerce business. The website will contain varities of products like electronics, garments , groceries etc. This website will have 3 types of users,

1. Consumer/Regular User/Visitor
2. Seller
3. Admin

In the website, all users can

1. registration
2. login
3. view and update their profile information.

Aditionally the general users will be able to do the following task:

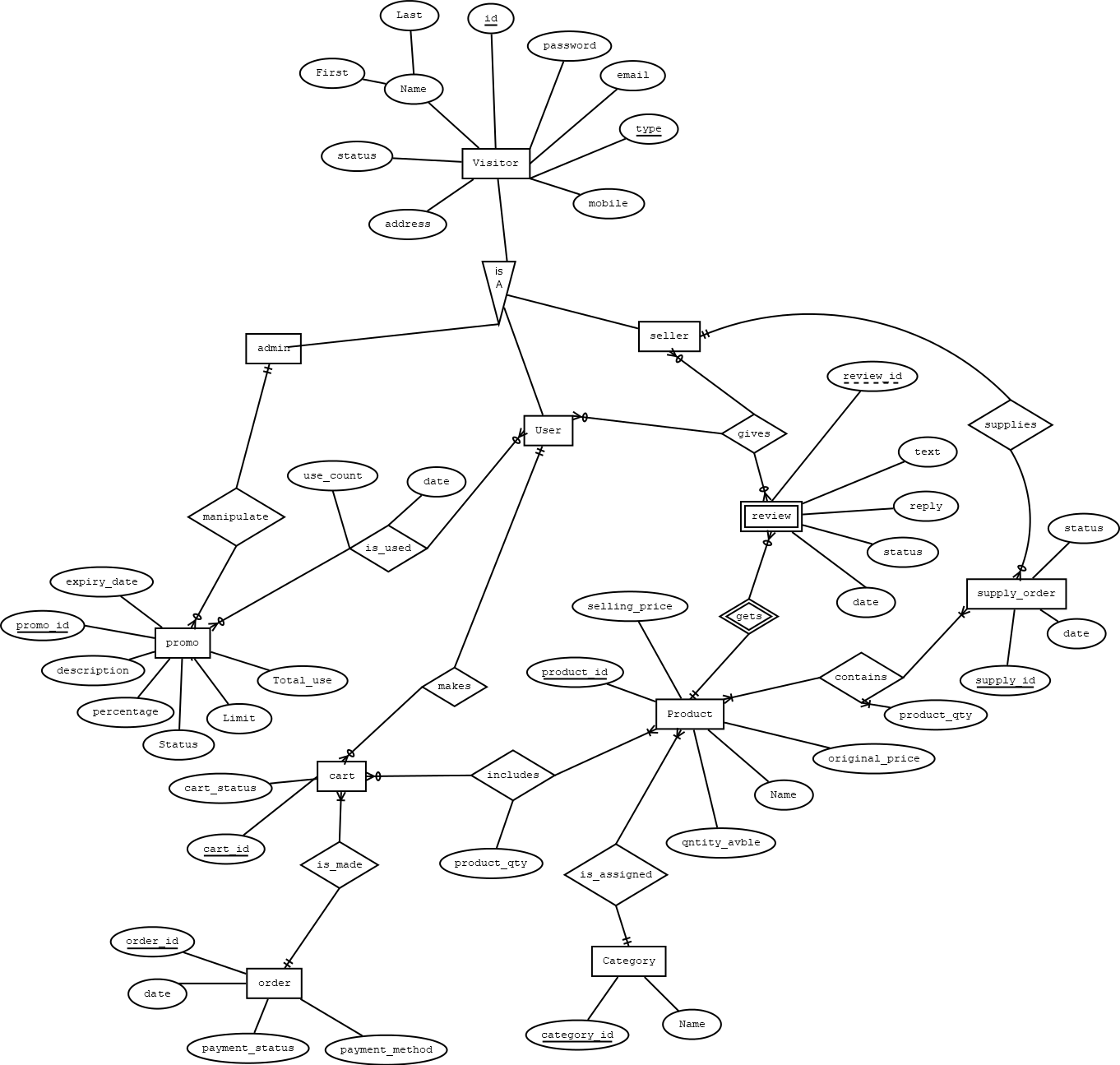
1. User can visit product from different product categories.
2. Choose Products to save into Wishlish and Cart
3. In future or instantly, they can buy it from the cart or wishlist.
4. User can pay immediately using different payment method
5. Instead of paying instantly they can choose to pay cash on delivery
6. After making an order , user can show order list and check current status of an order.
7. If there is any problem in product, user can request for return.
8. User can contact customer service using a message box.

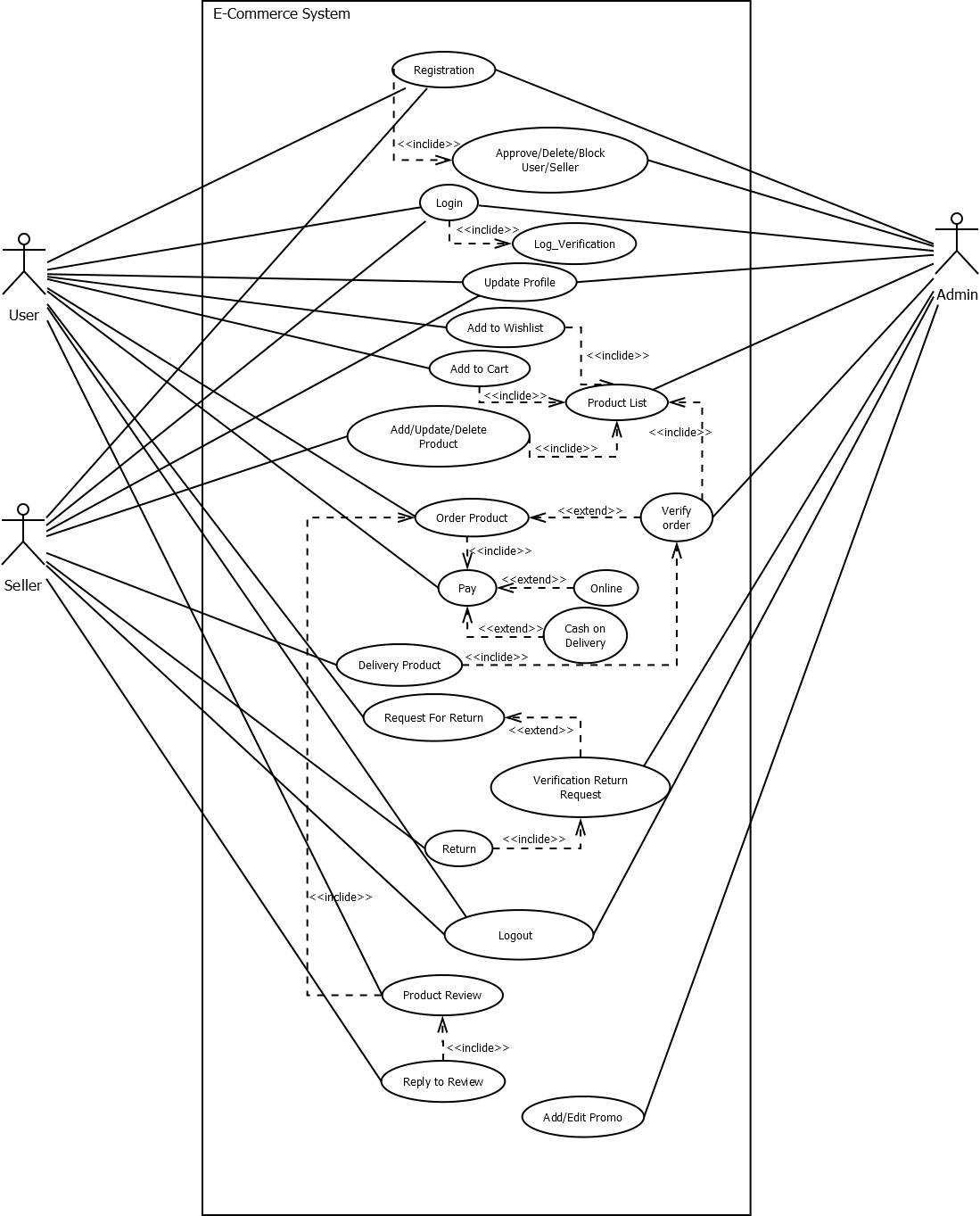
Admin will have the following unique functionality:

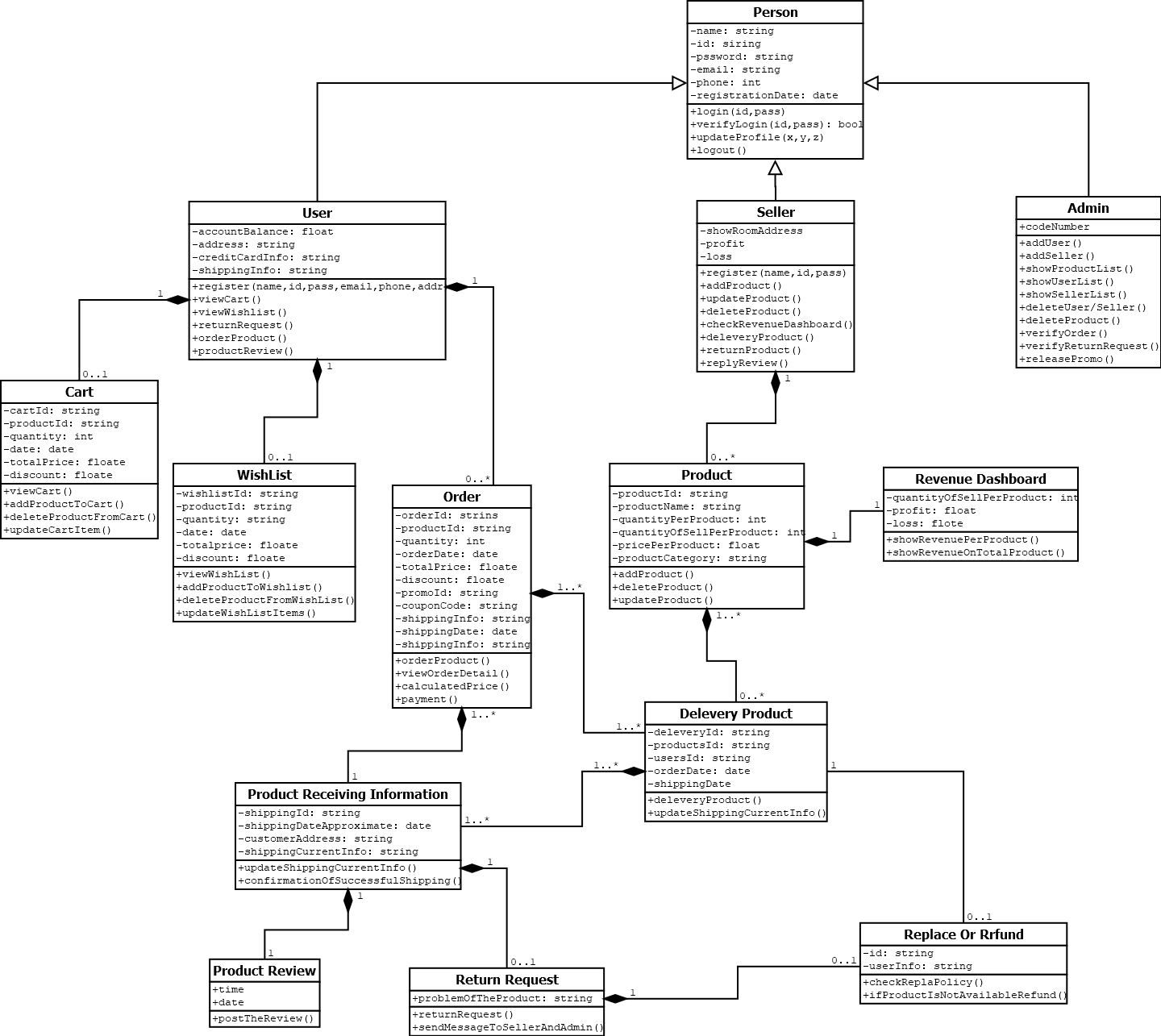
1. Can create, delete/block, aprrove user/seller/admin
2. Manipulate product (approve/reject/update)
3. Add/Update promocode
4. Reply to the query from other user
5. Can manipulate the order info/status
6. Can approve/reject/remove the review of product

Seller can enjoy the following additonal features:

1. Can request to add a product
2. Update the stock of a product
3. Can aprrove a product request from admin/user
4. Send message to the admin
5. Check revenue dashboard
6. Check product order status
7. Can reply to review or answer questions

**Entity – Relationship diagram (ER Diagram)**

**Use Case Diagram**

**Class Diagram**

**Normalization (upto 3NF)**

1. Admin:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| a\_id | a\_password | a\_address | a\_email | a\_mobile | u\_status | u\_type |

1. Seller:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| s\_id | s\_password | s\_address | s\_email | s\_mobile | u\_status | u\_type |

1. G\_user:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| g\_u\_id | g\_u\_password | g\_u\_address | g\_u\_email | g\_u\_mobile | u\_status | u\_type |

1. Multivalued attribute for admin:

Seller\_name

|  |  |  |  |
| --- | --- | --- | --- |
| s\_id | u\_type | first\_name | last\_name |

1. Multivalued attribute for seller:

Admin\_name

|  |  |  |  |
| --- | --- | --- | --- |
| a\_id | u\_type | first\_name | last\_name |

1. Multivalued attribute for g\_user:

G\_user\_name

|  |  |  |  |
| --- | --- | --- | --- |
| g\_u\_id | u\_type | first\_name | last\_name |

1. Promo:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| promo\_id | promo\_desc | Promo\_expiry | promo\_percentage | promo\_status |
| promo\_limit | promot\_use\_count | a\_id |

1. Promo\_use:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| promo\_id | user\_id | user\_type | user\_use\_date | user\_use\_count |

1. P\_include\_cart:

|  |  |  |
| --- | --- | --- |
| cart\_id | product\_id | product\_qntity |

1. Cart: ***[Updated]***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cart\_id | cart\_status | user\_id | g\_u\_type | order\_id |

1. Order\_t: ***[Updated]***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| order\_id | order\_date | payment\_method | payment\_status | return\_id |

1. Products: ***[Updated]***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| product\_id | product\_name | product\_price | product\_avlble | product\_sell\_price | product\_original\_price |
| category\_id |

1. Categories:

|  |  |
| --- | --- |
| category\_id | category\_name |

1. Supply\_order:

|  |  |  |  |
| --- | --- | --- | --- |
| supply\_id | supply\_date | supply\_status | s\_id |

1. Supply\_contains:

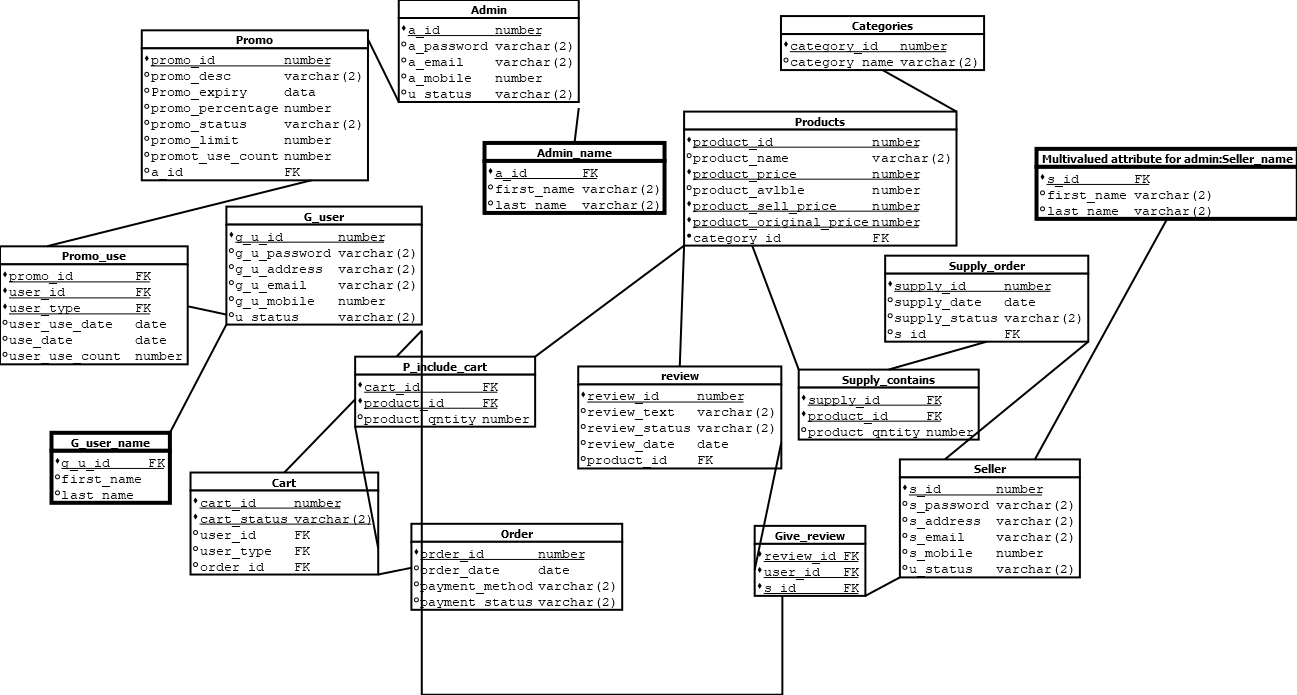
|  |  |  |
| --- | --- | --- |
| supply\_id | product\_id | product\_qntity |

1. review: ***[Updated]***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| review\_id | review\_text | review\_status | review\_date | product\_id |

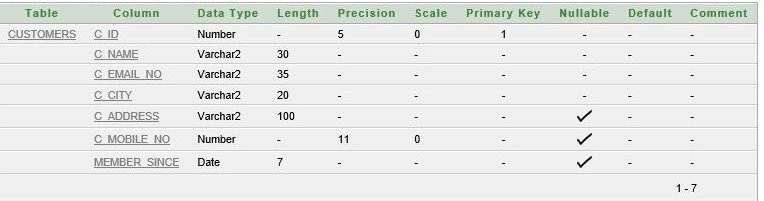
1. Give\_review:

|  |  |  |
| --- | --- | --- |
| review\_id | user\_id | s\_id |

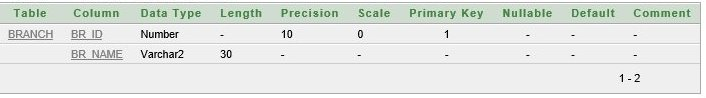
**Relationship diagram (Schema Diagram)**

**Screenshots of the table with only data type of the attributes**

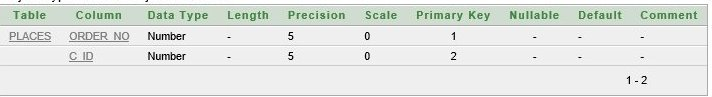
**Customers:**



**Branch:**



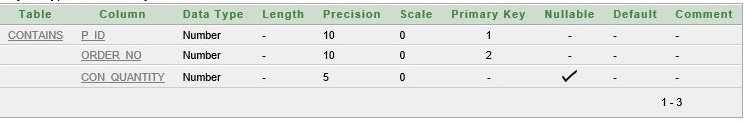
**Places:**



**Orders:**



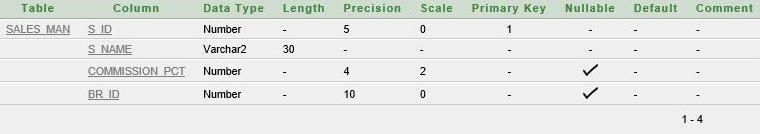
**Contains:**



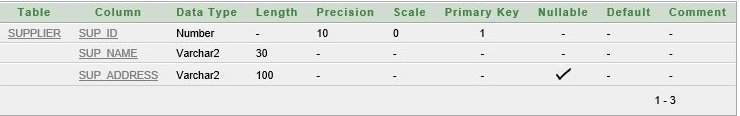
**Products:**



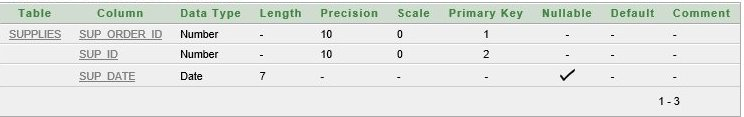
**Salesman:**



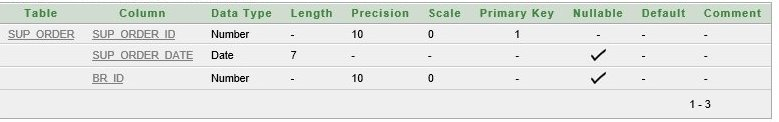
**Supplier:**



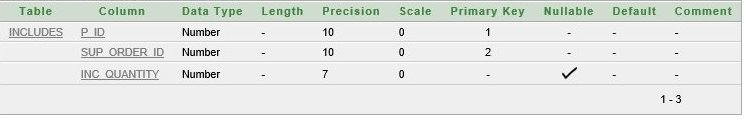
**Supplies:**



**Sup\_order:**

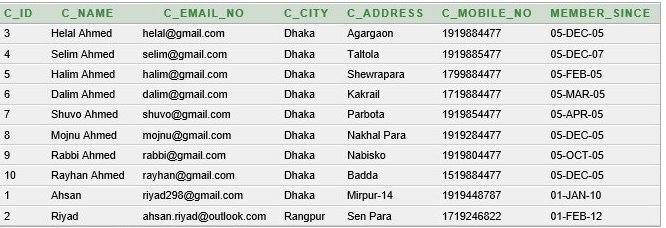


**Includes:**

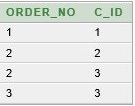


**Screenshot of the tables with sample data records**

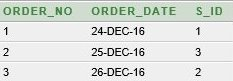
**Customers:**



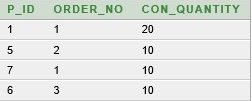
**Places:**



**Orders:**



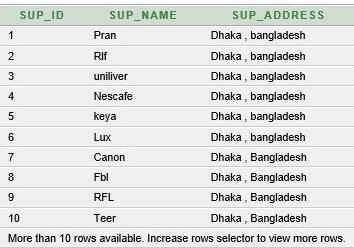
**Contains:**



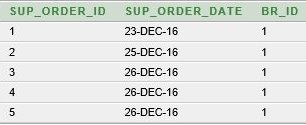
**Products:**



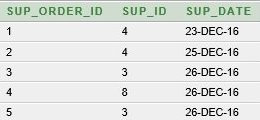
**Supplier:**



**Sup\_order:**



**Supplies:**



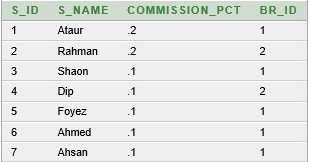
**Includes:**



**Branch:**



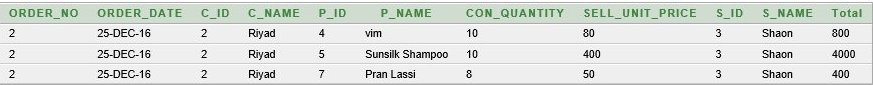
**Sales\_man:**



**Demonstrating some database use scenarios and SQL queries**

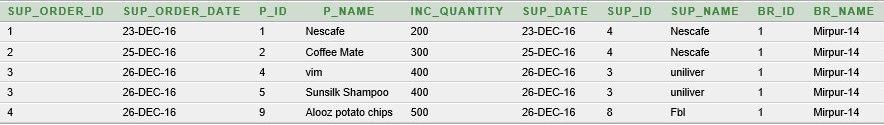
**1.** Generate a bill for order number 2

*select o.order\_no , o.order\_date , c.c\_id , c.c\_name , p.p\_id , p.p\_name , con.con\_quantity , p.sell\_unit\_price , s.s\_id , s.s\_name , con.con\_quantity\*p.sell\_unit\_price as "Total" from orders o , contains con , sales\_man s , customers c , products p , places pl where o.order\_no = pl.order\_no and pl.c\_id = c.c\_id and o.order\_no = con.order\_no and o.s\_id = s.s\_id and con.p\_id = p.p\_id and o.order\_no = 2*



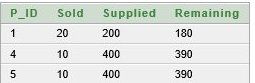
2. Display total supply of the shop

*select supo.sup\_order\_id , supo.sup\_order\_date , inc.p\_id , p.p\_name , inc.inc\_quantity , supplies.sup\_date , supl.sup\_id , supl.sup\_name , br.br\_id , br.br\_name from products p , includes inc , branch br , sup\_order supo , supplies , supplier supl where p.p\_id = inc.p\_id and inc.sup\_order\_id = supo.sup\_order\_id and supo.sup\_order\_id = supplies.sup\_order\_id and supo.br\_id = br.br\_id and supplies.sup\_id = supl.sup\_id order by supo.sup\_order\_id asc*



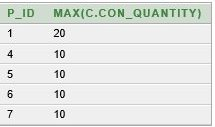
3. Display product supplied , sold and remaing number by Product id

*select c.p\_id , sum(c.con\_quantity) as "Sold" , sum(i.inc\_quantity) as "Supplied" , (sum(i.inc\_quantity)-sum(c.con\_quantity)) as "Remaining" from contains c , includes i , products p where c.p\_id = i.p\_id and i.p\_id = p.p\_id group by c.p\_id order by c.p\_id*



4. Display Maximum selling product by group by Product id

*select c.p\_id , max(c.con\_quantity) from products p , orders o , contains c where o.order\_no = c.order\_no and p.p\_id = c.p\_id group by c.p\_id*



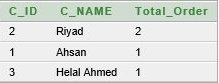
5. Display the sales\_id who sold has the highest selling history

*select o.s\_id, count(o.order\_no) from orders o group by o.s\_id order by count(o.order\_no) desc*



6. Display the most frequent customer id

*select p.c\_id , c.c\_name , count(p.order\_no) as "Total\_order" from places p , customers c where c.c\_id = p.c\_id group by p.c\_id , c.c\_name order by count(p.order\_no) desc*



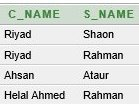
7. Display the mobile number of the customer who id is 2

*select c\_id , c\_name , c\_mobile\_no from customers where c\_id = 2*



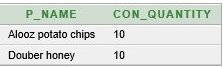
8 . Find the customer name and his sales man name

*select distinct(c.c\_name) , s.s\_name from customers c , sales\_man s , places p , orders o where c.c\_id = p.c\_id and p.order\_no = o.order\_no and s.s\_id = o.s\_id*



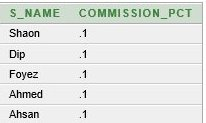
9. Find out goods and quantity which was sold on ‘26-Dec-2016’

*select p.p\_name , con.con\_quantity from products p , contains con , orders o where p.p\_id = con.p\_id and o.order\_no = con.order\_no and o.order\_date = '26/dec/16'*



10 . Find out which salesman get .1% commisson;

*select s\_name , commission\_pct from sales\_man where commission\_pct = .1*



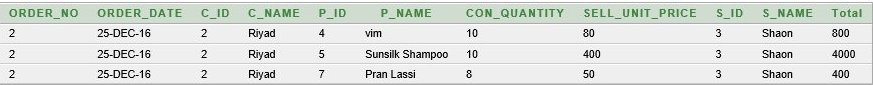
**View creation of the table**

**Complex view:**

1. **SQL Query:** create a view for generating bill

*Create or replace view bill as (select o.order\_no , o.order\_date , c.c\_id , c.c\_name , p.p\_id , p.p\_name , con.con\_quantity , p.sell\_unit\_price , s.s\_id , s.s\_name , con.con\_quantity\*p.sell\_unit\_price as "Total" from orders o , contains con , sales\_man s , customers c , products p , places pl where o.order\_no = pl.order\_no and pl.c\_id = c.c\_id and o.order\_no = con.order\_no and o.s\_id = s.s\_id and con.p\_id = p.p\_id and o.order\_no =*

**Output of the view:** select \* from bill;

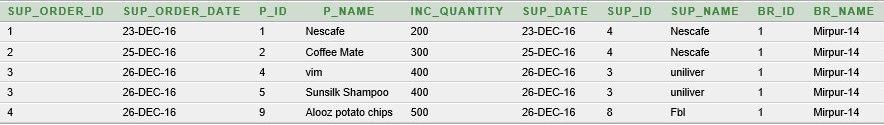


1. **SQL Query:** create view supply\_order so that it shows supply orders

*Create or replace view supply\_oder as (select supo.sup\_order\_id , supo.sup\_order\_date , inc.p\_id , p.p\_name , inc.inc\_quantity , supplies.sup\_date , supl.sup\_id , supl.sup\_name , br.br\_id , br.br\_name from products p , includes inc , branch br , sup\_order supo , supplies , supplier supl where p.p\_id = inc.p\_id and inc.sup\_order\_id = supo.sup\_order\_id and supo.sup\_order\_id = supplies.sup\_order\_id and supo.br\_id = br.br\_id and supplies.sup\_id = supl.sup\_id order by supo.sup\_order\_id asc*

*)*

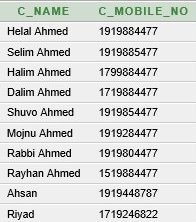
**Output of the view:** select \* from supply\_order;



**Simple view:**

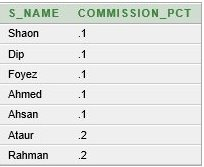
1. **SQL Query:** *create or replace view phone\_numbe as (select c\_name , c\_mobile\_no from customers )*

**Output of the view:** select \* from phone\_number;



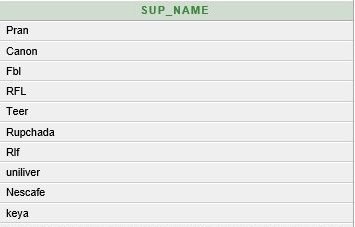
1. **SQL Query:** *create or replace view sman\_commission as (select s\_name , commission\_pct from sales\_man)*

**Output of the view:** select \* from sman\_commission;



1. **Sql Query :** create or replace view supplier\_name as (select sup\_name from supplier)

*Select \* from supplier\_name*



* **Bonus**

**Summarize learning experience from database project**

We have learned many lessons and hade valuable experience doing this project on database system. Though we had several problems doing this project but the achievements were more precious and efficient.

* This project would not be possible without the help of ORACLE SQL APPLICATION EXPRESS software.
* Companies or various institutes use databases because they can store a large number of records, ease of use when locating information, it is easy to add new data and to edit or delete old data, there is ease of storage Data can be imported into other applications
* The whole database model and specially the tables were created very carefully and the joining or linking among the table were done very carefully and also followed DBMS rules
* The constraints such as primary keys or foreign keys of the tables were assigned properly for linking among tables
* By fulfilling these objectives and system policies this project would be a proper and organized database system to run the system.
* The number of levels that the software is handling can be made unlimited in future from the current status of handling up to N levels as currently laid down by the software.
* Efficiency can be further enhanced and boosted up to a great extent by normalizing  and de-normalizing the database tables used in the project as well as taking the kind of the alternative set of data structures and advanced calculation algorithms available.
* We can in future generalize the application from its current customized status wherein other vendors developing and working on similar applications can utilize this system and make changes to it according to their client’s needs.
* Our main outcome would be faster processing of information as compared to the current system with high accuracy and reliability.
* Automatic and error free report generation as per the specified format with ease.
* With a fully automated solution, lesser staff, better space utilization and peaceful work environment, the institute is bound to experience high turnover.
* A future application of this system lies in the fact that the proposed system would remain relevant in the future.

In case there be any additions or deletion of the services, addition or deletion of any individual in any type of modification in future can be implemented easily. The data collected by the system will be useful for some other purposes also.

END\*\*