DBMS Project

▼ triggers and procedures

▼ Triggers

- **add_user** - adds new user details to the respective table based on cv_flag.
- **update_stock** → when new transaction is inserted into transactions table.(specific to vendor) or stock is updated.

▼ procedures

- **get_expenditure -** gets aggregate of money spent by customer
- **get_budget -** get customer budget
- **show_stock -** shows stock of each vendor separately
- **enter_transaction-** enters new transaction.
- **view_transaction** - view transaction filtered by id
- **update_stock** - updates vendor stock
- **view_profit -** view vendor profit based on cost price and selling price
- **add_product -** adds new product to list so stock can have this product as well
- **insert_new_user** - inserts new user

▼ MySQL commands used

▼ MySQL tables

- login_credentials → enter data in signup page, check data in login page

```
create table login_credentials( id int not null primary key auto_increment, username varchar(30), password varchar(30), firstna
```

```
desc login_credentials//
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| id          | int          | NO   | PRI | NULL    | auto_increment |
| username    | varchar(30)  | YES  |     | NULL    |                |
| password    | varchar(100) | YES  |     | NULL    |                |
| firstname   | varchar(30)  | YES  |     | NULL    |                |
| lastname    | varchar(30)  | YES  |     | NULL    |                |
| phonenumber | int          | YES  |     | NULL    |                |
| cv_flag     | varchar(2)   | YES  |     | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
7 rows in set (0.01 sec)
```

- customer → all customers

```
create table customer(customer_id int not null primary key, budget double not null default 0, name varchar(30), username va
```

```
desc customer//
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| customer_id | int          | NO   | PRI | NULL    |       |
| budget      | double       | NO   |     | 0       |       |
| name        | varchar(30)  | YES  |     | NULL    |       |
| username    | varchar(30)  | NO   |     | NULL    |       |
| password    | varchar(100) | YES  |     | NULL    |       |
| phonenumber | int          | YES  |     | NULL    |       |
| cv_flag     | varchar(2)   | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
7 rows in set (0.10 sec)
```

- vendor

```
create table vendor(vendor_id int not null primary key, name varchar(30), username varchar(30) not null, password varchar(1
```

```
desc vendor//
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| vendor_id   | int          | NO   | PRI | NULL    |       |
| name        | varchar(30)  | YES  |     | NULL    |       |
| username    | varchar(30)  | NO   |     | NULL    |       |
| password    | varchar(100) | YES  |     | NULL    |       |
| phonenumber | int          | YES  |     | NULL    |       |
| cv_flag     | varchar(2)   | NO   |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
6 rows in set (0.07 sec)
```

- product

```
create table product(product_id int(11) not null primary key, type varchar(30), name varchar(30), cost_price double, sellin
```

```
desc product//
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| product_id    | int         | NO   | PRI | NULL    |       |
| type          | varchar(30) | YES  |     | NULL    |       |
| name          | varchar(30) | YES  |     | NULL    |       |
| cost_price    | double      | YES  |     | NULL    |       |
| selling_price | double      | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
5 rows in set (0.01 sec)
```

- stock

```
create table stock(vendor_id int,foreign key(vendor_id) references vendor(vendor_id),product_id int,foreign key(product_id
```

```
desc stock//
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| vendor_id     | int         | YES  | MUL | NULL    |       |
| product_id    | int         | YES  | MUL | NULL    |       |
| quantity      | double      | YES  |     | NULL    |       |
| quantity_unit | varchar(30) | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
```

- transaction

```
create table transaction(transaction_id int(11) not null primary key, customer_id int(11), foreign key(customer_id) referen
t_id), date_time datetime, quantity double, quantity_unit varchar(30))//
```

```
desc transaction//
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| transaction_id | int         | NO   | PRI | NULL    |       |
| customer_id    | int         | YES  | MUL | NULL    |       |
| vendor_id      | int         | YES  | MUL | NULL    |       |
| product_id     | int         | YES  | MUL | NULL    |       |
| date_time      | datetime    | YES  |     | NULL    |       |
| quantity       | double      | YES  |     | NULL    |       |
| quantity_unit  | varchar(30) | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
```

▼ Procedures and triggers

- add_user trigger to fill vendor and customer

```
create trigger add_user after insert on login_credentials for each row begin if (new.cv_flag = '0') then insert into custom
```

- view_transactions

```
create procedure view_transactions(in x integer)
    -> begin
    -> if((select cv_flag from login_credentials where id = x) = '0') then select * from transaction where customer_id = x;
    -> end if;
    -> if((select cv_flag from login_credentials where id = x) = '1') then select * from transaction where vendor_id = x;
    -> end if;
    -> end //
```

- view_profit():

```
create procedure view_profit(in ven_id integer)
-> begin
-> select sum((product.selling_price - product.cost_price) * transaction.quantity) from transaction join product on product
-> end //
```

- trigger update_stock

```
create trigger update_stock after insert on transaction for each row
    -> begin
    -> declare t real;
    -> select quantity into t from stock where vendor_id = new.vendor_id and product_id = new.product_id;
    -> update stock set quantity = t - new.quantity where vendor_id = new.vendor_id and product_id = new.product_id;
    -> end//
```

- update stock procedure:

```
create procedure update_stock(in id integer, in pid integer, in qty real)
    -> begin
    -> declare t real;
    -> if exists(select * from stock where vendor_id = id and product_id = pid) then
    -> select quantity into t from stock where vendor_id = id and product_id = pid;
    -> update stock set quantity = t + qty where vendor_id = id and product_id = pid;
    -> else
    -> insert into stock(vendor_id, product_id, quantity) values(id, pid, qty);
    -> end if;
    -> end //
```

- delete user

```
create procedure delete_user(in x integer)
    -> begin
    -> delete from login_credentials where id = x;
    -> update vendor set cv_flag = 2 where vendor_id = x;
    -> update customer set cv_flag = 2 where customer_id = x;
    -> end //
```

- show vendor stock

```
create procedure show_vendor_stock(in id integer)
    -> begin
    -> select * from stock where vendor_id = id;
    -> end//
```

- add product:

```
create procedure add_product(in id integer, in ty varchar(30), in nm varchar(30), in cp real, in sp real)
    -> begin
    -> insert into product values(id, ty, nm, cp, sp);
    -> end //
```

- insert_transaction():

```
create procedure enter_transaction(in id integer, in cid integer, in vid integer, in pid integer, in qt real, in qu varchar
    -> begin
    -> insert into transaction values(id, cid, vid, pid, now(), qt, qu);
    -> end //
```

- get_expenditure():

```
create procedure get_expenditure(in cust_id integer)
-> begin
-> select sum(transaction.quantity * product.selling_price) from transaction join product on product.product_id = transacti
-> end//
```

- get_budget():

```
create procedure get_budget(in cust_id integer)
-> begin
-> select budget from customer where customer_id = cust_id;
-> end//
```

- show_stock() → shows vendor stock in different tables ***********on customer stock page*******

```
create procedure show_stock()
begin
declare id integer;
declare fin integer default 0;
declare c cursor for select vendor_id from vendor;
declare continue handler for not found set fin= 1;
open c;
ven:loop
fetch c into id;
if(fin = 1) then leave ven;
end if;
select * from stock where vendor_id = id ;
end loop ven;
close c;
end//
```

- insert_new_user signup page

```
create procedure insert_new_user(uname varchar(30), pass varchar(100), fname varchar(30), lname varchar(30), pnum int(1
begin insert into login_credentials values (DEFAULT,uname,pass,fname,lname,pnum,cv_flag); end//
```