Login

select \* from e\_commerce.users;

select \* from e\_commerce.users where email='theresa12@gmail.com' and password='admin123';

Admin assign roles to employee:

Combo box = select \* from e\_commerce.role;

Assign role\_id using employee\_id

update e\_commerce.employee set role\_id\_pk\_fk = 10003 where emp\_id=1;

User Management

It will be normal Add/Edit/Modify in admin section for Admin Users.

New User Signup/Add admin

insert into e\_commerce.employee (first\_name,last\_name,loging\_username,login\_password) values ('harish','kumar','harish','harish123');

Admin edit data and update

update e\_commerce.employee set first\_name='ganesh', last\_name='acharaya', loging\_username='ganesh' ,login\_password='ganesh123' where emp\_id=3;

**Configuration Management**

Admin can define constants which can be used for shopping cart. This values need to define in admin so that admin can change those any point of time without changing any code. For ex: admin\_email, notification\_email. Use same CRUD package.

Admin - Banner Management Refer Banner table and use CRUD to create ADD / EDIT / DELETE operation.

**Admin** - Product Management Admin can manage the product from admin site.

**Basic Product details will be saved in product table.**

insert into e\_commerce.product (product\_name,price,quantity,weight) values ('Tusha Rice',55.00,1,'1 kg')

Product images will be saved in product\_images table.

insert into e\_commerce.product\_image(productImage) values(LOAD\_FILE('C:\code\e-commerce\ecommerce\public\images\semi.png'))

Product categories will be saved in product\_categories table.

insert into product\_category (product\_id,category\_id) values(2,1);

Product parameters will be saved in product\_attributes\_assoc table.

insert into product\_attributes\_assoc (product\_id\_fk, brand, manufacture\_date,expire\_date) values(1,'Tanddul','2020/3/20','2022/3/20');

Refer : product, product\_categories, product\_images, product\_attributes\_assoc

**Display Product**

select product.product\_id, product.product\_name,product.price,product.quantity, product\_attributes\_assoc.brand, product\_attributes\_assoc.expire\_date from product

join product\_attributes\_assoc

on product.product\_id = product\_attributes\_assoc.product\_id\_fk;

**Category Table**

insert into e\_commerce.category (category\_name) values ('bread');

Admin - Coupon Management Admin will be able to generate the **coupon cod**e which will get applied on grand total – shipping charges. This alphanumeric code admin will generate from admin which can be redeemed in frontend.

insert into coupon(product\_id,amount,discount) values (2,1000,10);

**Payment Method** – There will be two payment methods COD – Cash on Delivery Paypal IPN – Refer <https://developer.paypal.com/docs/classic/ipn/gs_IPN/>

insert into payment(payment\_id , payment\_method) values(1,Cash On Delivery);

insert into payment(payment\_id , payment\_method) values(2,'Paytm');

**Customer Order - three tables enter once**

insert into customer( customer\_id,customer\_name) values (1,'roopa');

insert into customer\_details(customer\_id\_fk,first\_name,last\_name,user\_name,password,phone) values(1,'roopa','iyer','roopa','roopa123',8876543234);

insert into customer\_address(address\_id,customer\_id\_fk,address1,address2,street,city,state,pincode) values (1,1,'#334','#334','kt street','mysore','karnataka',570021);

**Admin :** Order Management Admin should be able to see the order listing and the details of each order. It should contain the following information:

1. Customer details with Address
2. 2. Ordered products
3. 3. Shipping Method
4. 4. Payment gateway details.
5. **Customer details with Address**

insert into user\_order values(1,1);

select \* from customer\_address where customer\_id\_fk=(select customer\_id from user\_order where order\_id=1);

1. **Ordered products**

**insert into order\_details (order\_details\_id,order\_id,product\_id,coupon\_id,quantity,payment\_id) values (1,1,1,1,1,1);**

**update order\_details set shippingaddress=(select address1 from customer\_address where customer\_id\_fk=1)**

**where order\_id =1;**

**select product\_id ,quantity from order\_details where order\_id=1;**

After order has placed stock has to be reduced in quantity column of product

sold= select product\_id ,quantity from order\_details where order\_id=1;

stock=select quantity from order\_details;

select \* from order\_details where order\_id=(select order\_id from user\_order where customer\_id =1);

update product set quantity=(sold-stock)

Admin : Reports Admin will be able to grab below reports from the admin section.

1. Sales Report
2. Customer Registered
3. Coupons used