CS306

Database Systems Course

Term Project

PHASE - I

Mahir Akbaş ---------- 29492
Bora Urasoğlu ---------- 27775
Ufuk Ulaş Tokat ---------- 28914
Aksel Dindisyan ---------- 32109
Hüseyin Berke Fırat ------ 29011

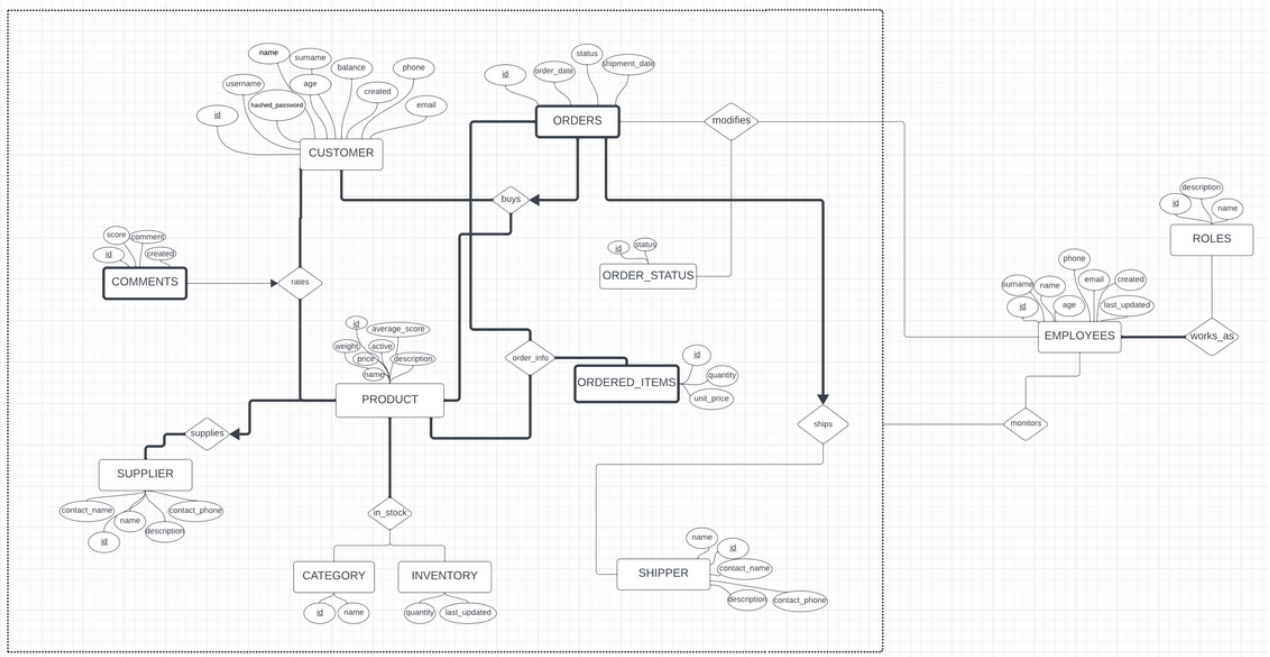
1. Project Description

Project Title: E - Commerce Webpage Project

Introduction: In this project, as a group, we are planning to demonstrate an exemplary e ecommerce web application that is complete with simple frontend and a reliable backend that will satisfy CRUD operations. We intend to add features such as user login/logout & register, product display, landing page, checkout page, orders page

Entity Relations Explanation: As for the ER model, Our system will have a product entity to identify the goods that are being sold. Category and Inventory entities will follow the detailed data about product entity and all products will have a supplier whose information will be stored in supplier entity. We will have another entity that will store customer information including its auth data. When customers buy products they will create orders that will be stored in the orders entity. Ordered items within single order will be stored in the ordered_items entity with the aim of accommodating the details of the order. Moreover, all orders will have an assigned shipper who will transport the products from warehouse to customer. Shipper information will be stored in shipper entity. In addition, customers will be able to rate and comment on the product which they buy. These comments will be stored in a separate commentsentity. Apart from that we will have an entity called employees which will be able to alter the status of orders. They will also monitor all process. Status codes of the orders are stored in the order statusentity. Similar to that, employe's role codes are also stored in separate roles entity with their respective details.

2. ER Model



3. Relational Model

```
/*
E - Commerce Store Database
 creator
*/

DROP DATABASE IF EXISTS ecommerce;
CREATE DATABASE ecommerce;

USE ecommerce;

-- ------------------------------------------------------------- --

CREATE TABLE roles (
    id TINYINT PRIMARY KEY AUTO_INCREMENT,
    `name` VARCHAR(50) NOT NULL,
    description TEXT
);

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

next page ->

```
CREATE TABLE customer (

 id INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50) NOT NULL,
    hashed_password VARCHAR(255) NOT NULL,
 `name` VARCHAR(50) NOT NULL,
 surname VARCHAR(50) NOT NULL,
 age INT NOT NULL,
    phone VARCHAR(30) NOT NULL,
 email VARCHAR(150),
    balance DECIMAL(10,2) NOT NULL DEFAULT 0 CHECK (balance > -1),
    created DATETIME NOT NULL DEFAULT current_timestamp(),
    last_updated DATETIME DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP(),

 CONSTRAINT ageCheck CHECK (age > 0)
) AUTO_INCREMENT = 1;

-- ------------------------------------------------------------- --

CREATE TABLE category (

 id INT PRIMARY KEY AUTO_INCREMENT,
    `name` VARCHAR(40) NOT NULL UNIQUE
) AUTO_INCREMENT=1;

-- ------------------------------------------------------------- --

CREATE TABLE supplier (

 id TINYINT PRIMARY KEY AUTO_INCREMENT,
    `name` VARCHAR(70) NOT NULL,
    contact_name VARCHAR(50) NOT NULL,
    contact_phone VARCHAR(30) NOT NULL,
    `description` TEXT
) AUTO_INCREMENT=1;

-- ------------------------------------------------------------- --

CREATE TABLE shipper (

 id TINYINT PRIMARY KEY AUTO_INCREMENT,
 `name` VARCHAR(70) NOT NULL,
 contact_name VARCHAR(50) NOT NULL,
 contact_phone VARCHAR(30) NOT NULL,
 `description` TEXT
) AUTO_INCREMENT=1;

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

```
CREATE TABLE product (

    id INT NOT NULL PRIMARY KEY auto_increment,
    `name` VARCHAR(100) NOT NULL,
    category_id INT NOT NULL,
    price DECIMAL(10,2) NOT NULL,
    weight DOUBLE NOT NULL,
    supplier_id TINYINT NOT NULL,
    `active` BOOL NOT NULL,
    `description` TEXT,
    average_points  TINYINT,
    created DATETIME NOT NULL DEFAULT current_timestamp(),
    last_updated DATETIME DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP(),

    CONSTRAINT priceCheck CHECK (price > 0),
    CONSTRAINT weightCheck CHECK (weight > 0),
       CONSTRAINT  FK_category_id  FOREIGN  KEY  (category_id)  REFERENCES
category(id) ON DELETE NO ACTION ON UPDATE CASCADE,
       CONSTRAINT  FK_supplier_id  FOREIGN  KEY  (supplier_id)  REFERENCES
supplier(id) ON DELETE NO ACTION ON UPDATE NO ACTION
) AUTO_INCREMENT=1;

-- ------------------------------------------------------------- --

CREATE TABLE inventory (

    product_id INT NOT NULL,
    quantity INT NOT NULL,
    last_updated DATETIME DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP(),

        CONSTRAINT  FK_product_id  FOREIGN  KEY  (product_id)  REFERENCES
product(id),
    CONSTRAINT quantityCheck CHECK (quantity > -1)
);

 -- ------------------------------------------------------------- -

 CREATE TABLE order_status (   -- (0,'NA') 1,'Received') (2,'Processed')
(3,'Shipped') (4,'Delivered')

    status_id TINYINT PRIMARY KEY,
    `name` VARCHAR(30) NOT NULL
);

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

```
CREATE TABLE orders

    id INT PRIMARY KEY AUTO_INCREMENT,
    customer_id INT,
    order_date DATETIME NOT NULL DEFAULT current_timestamp(),
    shipment_date DATETIME,
    shipper_id TINYINT NOT NULL,
      `status` TINYINT NOT NULL DEFAULT 0,   -- 0 -> order NA, 1 -> processing, 2 -> on the way, 3 -> delivered, 4 -> canceled (MUST BE CONVERTED TO ENUM TYPE LATER ON)
    last_updated DATETIME DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP(),

       CONSTRAINT FK_customer_id FOREIGN KEY (customer_id) REFERENCES customer(id) ON DELETE CASCADE ON UPDATE CASCADE,
        CONSTRAINT FK_shipper_id FOREIGN KEY (shipper_id) REFERENCES shipper(id) ON DELETE NO ACTION,
          CONSTRAINT FK_status FOREIGN KEY (`status`) REFERENCES order_status(status_id) ON DELETE NO ACTION ON UPDATE CASCADE
) AUTO_INCREMENT = 1;

-- ------------------------------------------------------------- --

CREATE TABLE employees (

 id INT PRIMARY KEY AUTO_INCREMENT,
    `role` TINYINT NOT NULL,
 `name` VARCHAR(50) NOT NULL,
 surname VARCHAR(50) NOT NULL,
 age INT NOT NULL,
    phone VARCHAR(30) NOT NULL,
 email VARCHAR(150),
 created DATETIME NOT NULL DEFAULT current_timestamp(),
    last_updated DATETIME DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP(),

 CONSTRAINT ageCheck2 CHECK (age > 18),
     CONSTRAINT FK_role FOREIGN KEY (`role`) REFERENCES roles(id) ON DELETE NO ACTION ON UPDATE CASCADE
) AUTO_INCREMENT = 1;

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

```
CREATE TABLE ordered_items (

 id INT PRIMARY KEY AUTO_INCREMENT,
    order_id INT,
    product_id INT,
    quantity INT CHECK (quantity > -1),
    unit_price DECIMAL(10,2)

 -- PRIMARY KEY(order_id, id),
    CONSTRAINT FK_order_id2 FOREIGN KEY (order_id) REFERENCES orders(id)
ON DELETE CASCADE ON UPDATE CASCADE,
   CONSTRAINT   FK_product_id2   FOREIGN   KEY   (product_id)   REFERENCES
product(id) ON DELETE CASCADE ON UPDATE CASCADE
) AUTO_INCREMENT = 1;

-- ------------------------------------------------------------------ --

CREATE TABLE comments (
 comment_id INT PRIMARY KEY AUTO_INCREMENT,
    product_id INT NOT NULL,
    customer_id INT NOT NULL,
    `comment` LONGTEXT,
    `point` INT CHECK ( 0 < point <= 10),
    created DATETIME NOT NULL DEFAULT current_timestamp(),

      CONSTRAINT  FK_customer_id2  FOREIGN  KEY  (customer_id)  REFERENCES
customer(id) ON DELETE CASCADE ON UPDATE CASCADE,
       CONSTRAINT  FK_product_id3  FOREIGN  KEY  (product_id)  REFERENCES
product(id) ON DELETE CASCADE ON UPDATE CASCADE
) AUTO_INCREMENT = 1;

-- ------------------------------------------------------------------ --

DELIMITER //
CREATE TRIGGER after_comment_insert
AFTER INSERT ON comments FOR EACH ROW
BEGIN
    UPDATE product SET average_points = (
        SELECT AVG(`point`)
        FROM comments
        WHERE product_id = NEW.product_id
    )
    WHERE id = NEW.product_id;
END //
DELIMITER ;
```