

Database for Shopping Mall

Hyunki Lee

Georgia State University

hlee152@student.gsu

1. Introduction

These days, a lot of people prefer to buy items on online. The trend is not for certain years. Thus, the shopping mall go along the current trends such as they sell not only certain type of items but also treating various sort of items. The manager who manages the shopping mall need to contact with diverse suppliers because suppliers will be different depend on sort of products. Moreover, taking care of customers will be more important since the number of online shopping users are increasing. This database will help to manage the shopping mall because it provides not only supplier information but also questions from customers information as well.

2. Topic & Scope

The database focuses on management online shopping mall from manager's view. This database will help to administer the shopping mall efficiently. The target audiences are shopping malls that treat various products from various suppliers. Thus, this database may not suitable for small size or treating limited category of products.

3. Implementation

a. Mysql

This database has nine tables. Each table has their own primary key which is id. There are five foreign keys. The details of each table follow with screen shots.

1. admin table

```

CREATE TABLE `mall`.`admin` (
  `admin_id` VARCHAR(10) NOT NULL,
  `admin_password` VARCHAR(45) NULL,
  PRIMARY KEY (`admin_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

INSERT INTO `mall`.`admin`
(`admin_id`,
`admin_password`)
VALUES
("admin", "admin");

```

The admin table is for log-in purpose since the database is for managing shopping mall.

2. supplier table

```

CREATE TABLE `mall`.`supplier` (
  `supplier_id` VARCHAR(10) NOT NULL,
  `company` VARCHAR(45) NULL,
  `contact_manager` VARCHAR(255) NULL,
  `contact_number` VARCHAR(255) NULL,
  PRIMARY KEY (`supplier_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

INSERT INTO `mall`.`supplier`
(`supplier_id`,
`company`,
`contact_manager`,
`contact_number`)
VALUES
("s1", "Dust", "Peter", "404-321-8763"),
("s2", "appa", "William", "404-123-4567"),
("s3", "Mcdo", "Jenny", "770-345-5678"),
("s4", "Fam", "Janet", "321-735-1870"),
("s5", "Leg", "James", "521-321-1237"),
("s6", "Wind", "Alex", "432-759-4679"),
("s7", "Big", "George", "404-902-7315"),
("s8", "Small", "June", "905-346-9853"),
("s9", "Rain", "Tran", "665-096-1593"),
("s10", "Real", "Lucas", "654-987-3210");

```

The supplier table contain information of supplier_id, company, supplier company's manager name, supplier company's phone number.

3. product table

```

CREATE TABLE `mall`.`product` (
  `product_id` VARCHAR(10) NOT NULL,
  `product_name` VARCHAR(255) NULL,
  `product_category` VARCHAR(255) NULL,
  `product_price` FLOAT NULL,
  `product_quantity` INT NULL,
  `supplier_id` VARCHAR(10) NULL,
  PRIMARY KEY (`product_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

ALTER TABLE `mall`.`product`
ADD CONSTRAINT `product_FK1`
  FOREIGN KEY (`supplier_id`)
    REFERENCES `mall`.`supplier` (`supplier_id`);

INSERT INTO `mall`.`product`
(`product_id`,
`product_name`,
`product_category`,
`product_price`,
`product_quantity`,
`supplier_id`)
VALUES
('p1','envelopement','life','40',20,'s1'),
('p10','hoodie','cloth','40',5,'s10'),
('p2','chocolatebox','food','50',10,'s2'),
('p3','caramelcandy','food','50',20,'s3'),
('p4','tshirt','cloth','40',10,'s4'),
('p5','dryer','beauty','50',20,'s5'),
('p6','shoe','cloth','40',10,'s6'),
('p7','kitchenware','life','50',20,'s7'),
('p8','snackbar','food','40',10,'s8'),
('p9','cream','beauty','50',10,'s9');

```

The product table contains product's id, name, category, price and quantity. There is one foreign key which is supplier_id. The key is reference supplier table.

4. order_detail table

```

CREATE TABLE `mall`.`order_detail` (
  `order_id` VARCHAR(10) NOT NULL,
  `product_id` VARCHAR(10) NULL,
  `order_quantity` INT NULL,
  PRIMARY KEY (`order_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

INSERT INTO `mall`.`order_detail`
(`order_id`,
`product_id`,
`order_quantity`)
VALUES
("o1", "p1", 3),
("o2", "p2", 5),
("o3", "p3", 10),
("o4", "p4", 6),
("o5", "p5", 1),
("o6", "p6", 3),
("o7", "p7", 3),
("o8", "p8", 3),
("o9", "p9", 3),
("o10", "p10", 1);

ALTER TABLE `mall`.`order_detail`
ADD CONSTRAINT `order_detail_FK1`
FOREIGN KEY (`product_id`)
REFERENCES `mall`.`product` (`product_id`);

```

order id, product id and order quantity are. in the order_detail table. The foreign key is product_id which is in the product table.

5. buyer table

```

CREATE TABLE `mall`.`buyer` (
  `buyer_id` VARCHAR(10) NOT NULL,
  `buyer_phone` VARCHAR(255) NULL,
  `email` VARCHAR(255) NULL,
  `buyer_name` VARCHAR(255) NULL,
  `buyer_password` VARCHAR(255) NULL,
  `buyer_address` VARCHAR(255) NULL,
  `buyer_city` VARCHAR(255) NULL,
  `buyer_state` VARCHAR(255) NULL,
  `buyer_country` VARCHAR(255) NULL,
  `buyer_zip` VARCHAR(10) NULL,
  PRIMARY KEY (`buyer_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

INSERT INTO `mall`.`buyer`
(`buyer_id`, `buyer_phone`, `email`, `buyer_name`, `buyer_password`, `buyer_address`, `buyer_city`, `buyer_state`, `buyer_country`, `buyer_zip`)
VALUES
("1", "770-567-890", "john@gmail.com", "John", "abcde", "3020 street", "Atlanta", "GA", "US", "30333"),
("2", "660-123-890", "jane@gmail.com", "Jane", "abcdeasdf", "1231 street", "Los Angeles", "CA", "US", "12579"),
("3", "102-293-5153", "kim@gmail.com", "Kim", "asl23asd", "1342 In St.", "Irvine", "CA", "US", "29786"),
("4", "769-844-1019", "matthew@gmail.com", "Matthew", "da34asdbv", "1231 Saint Clair", "Atlanta", "GA", "US", "33897"),
("5", "517-704-7914", "dominic@gmail.com", "Dominic", "abc12asda", "12312 street Dr", "Augusta", "GA", "US", "46579"),
("6", "457-678-952", "Lynn@gmail.com", "Lynn", "nfg1231a", "512 main", "Columbus", "GA", "US", "30792"),
("7", "975-684-930", "Dustin@gmail.com", "Dustin", "zpej12", "road 1231", "Macon", "GA", "US", "79630"),
("8", "975-652-642", "kelly@gmail.com", "Kelly", "asbqweizxc", "Wall street", "Athens", "GA", "US", "36987"),
("9", "125-789-654", "Davie@gmail.com", "Davie", "vzxqwee52", "80 Dr.", "Atlanta", "GA", "US", "36925"),
("10", "123-456-789", "lee@gmail.com", "Lee", "asdfg", "123 street", "Atlanta", "GA", "US", "12345");

```

This is the buyer table. There are buyer id, phone number, email, name, password, address, city, state, country, and zip code.

6. shipment table

```

CREATE TABLE `mall`.`shipment` (
  `ship_id` VARCHAR(10) NOT NULL,
  `ship_phone` VARCHAR(255) NULL,
  `ship_company` VARCHAR(255) NULL,
  PRIMARY KEY (`ship_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

INSERT INTO `mall`.`shipment`
(`ship_id`,
`ship_phone`,
`ship_company`)
VALUES
("sh1", "951-654-357", "Fedex"),
("sh2", "951-654-357", "Fedex"),
("sh3", "682-493-0019", "UPS"),
("sh4", "682-493-0019", "UPS"),
("sh5", "300-600-500", "USPS"),
("sh6", "300-600-500", "USPS"),
("sh7", "682-493-0019", "UPS"),
("sh8", "682-493-0019", "UPS"),
("sh9", "951-654-357", "Fedex"),
("sh10", "951-654-357", "Fedex");

```


Ship id, phone number, and company is in the shipment table. The primary key is the ship_id.

7. payment table

```
CREATE TABLE `mall`.`payment` (  
  `payment_id` VARCHAR(10) NOT NULL,  
  `card_holder` VARCHAR(255) NULL,  
  `card_type` VARCHAR(255) NULL,  
  `card_number` FLOAT NULL,  
  `card_exp_month` INT NULL,  
  `card_exp_year` INT NULL,  
  `billing_address` VARCHAR(255) NULL,  
  `billing_city` VARCHAR(255) NULL,  
  `billing_state` VARCHAR(255) NULL,  
  `billing_country` VARCHAR(255) NULL,  
  `billing_zip` VARCHAR(255) NULL,  
  `order_id` VARCHAR(255) NULL,  
  PRIMARY KEY (`payment_id`));  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;  
  
INSERT INTO `mall`.`payment`  
(`payment_id`,  
`card_holder`,  
`card_type`,  
`card_number`,  
`card_exp_month`,  
`card_exp_year`,  
`billing_address`,  
`billing_city`,  
`billing_state`,  
`billing_country`,  
`billing_zip`,  
`order_id`)  
VALUES  
(  
  ("pa1", "John", "VISA", "3578489563", "03", "25", "3020 street", "Atlanta", "GA", "US", "30333", "o1"),  
  ("pa2", "Jane", "VISA", "3574896532", "04", "29", "1231 street", "Los Angeles", "CA", "US", "12579", "o2"),  
  ("pa3", "Kim", "VISA", "4987561230", "06", "22", "1342 In St.", "Irvine", "GA", "US", "29786", "o3"),  
  ("pa4", "Matthew", "VISA", "9751489630", "11", "23", "1231 Saint Clair", "Atlanta", "GA", "US", "33897", "o4"),  
  ("pa5", "Dominic", "VISA", "3574963015", "12", "25", "12312 street Dr", "Augusta", "GA", "US", "46579", "o5"),  
  ("pa6", "Lynn", "VISA", "8500126400", "1", "22", "512 main", "Columbus", "GA", "US", "30792", "o6"),  
  ("pa7", "Dustin", "VISA", "6699880045", "02", "24", "road 1231", "Macon", "GA", "US", "79630", "o7"),  
  ("pa8", "Kelly", "VISA", "6655980231", "06", "24", "Wall street", "Athens", "GA", "US", "36987", "o8"),  
  ("pa9", "Davie", "VISA", "1000549875", "07", "23", "80 Dr.", "Atlanta", "GA", "US", "36925", "o9"),  
  ("pa10", "Lee", "VISA", "1234567891", "03", "22", "123 street", "Atlanta", "GA", "US", "12345", "o10");
```

The payment table has general payment information. payment_id is the primary key. Card information that needs to pay cost with billing address is the main columns. The type zip code is varchar because some country provide zip code not only numbers but hyphen between numbers as well.

8. orders table

```

CREATE TABLE `mall`.`orders` (
  `order_id` VARCHAR(10) NOT NULL,
  `buyer_id` VARCHAR(10) NOT NULL,
  `total_price` FLOAT NULL,
  `total_quantity` INT NULL,
  `payment_id` VARCHAR(10) NULL,
  `payment_date` DATE NULL,
  `order_date` DATE NULL,
  `cancel` INT NULL,
  `paid` INT NULL,
  `fulfill` INT NULL,
  `ship_date` DATE NULL,
  `ship_id` VARCHAR(10) NULL,
  PRIMARY KEY (`order_id`));
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

INSERT INTO `mall`.`orders`
(`order_id`,
`buyer_id`,
`total_price`,
`total_quantity`,
`payment_id`,
`payment_date`,
`order_date`,
`cancel`,
`paid`,
`fulfill`,
`ship_date`,
`ship_id`)
VALUES
("o1", "1", 120, 3, "pa1", "2021-03-11", "2021-03-11", 0, 1, 1, "2021-03-14", "sh1"),
("o2", "2", 250, 5, "pa2", "2021-02-11", "2021-02-11", 0, 1, 1, "2021-02-14", "sh2"),
("o3", "3", 500, 10, "pa3", "2021-01-11", "2021-01-11", 0, 1, 1, "2021-01-14", "sh3"),
("o4", "4", 240, 6, "pa4", "2021-02-20", "2021-02-20", 0, 1, 1, "2021-02-24", "sh4"),
("o5", "5", 40, 1, "pa5", "2021-01-19", "2021-01-19", 0, 1, 1, "2021-01-24", "sh5"),
("o6", "6", 120, 3, "pa6", "2021-03-20", "2021-03-20", 0, 1, 1, "2021-03-30", "sh6"),
("o7", "7", 150, 3, "pa7", "2021-01-14", "2021-01-14", 0, 1, 1, "2021-01-20", "sh7"),
("o8", "8", 120, 3, "pa8", "2021-02-22", "2021-02-22", 0, 1, 1, "2021-02-24", "sh8"),
("o9", "9", 150, 3, "pa9", "2021-03-06", "2021-03-06", 0, 1, 1, "2021-03-14", "sh9"),
("o10", "10", 40, 1, "pa10", "2021-04-19", "2021-04-19", 0, 1, 1, "2021-04-21", "sh10");

ALTER TABLE `mall`.`orders`
ADD CONSTRAINT `order_FK1`
  FOREIGN KEY (`buyer_id`)
    REFERENCES `mall`.`buyer` (`buyer_id`),

ADD CONSTRAINT `order_FK2`
  FOREIGN KEY (`payment_id`)
    REFERENCES `mall`.`payment` (`payment_id`),

ADD CONSTRAINT `order_FK3`
  FOREIGN KEY (`ship_id`)
    REFERENCES `mall`.`shipment` (`ship_id`),

ADD CONSTRAINT `order_FK4`
  FOREIGN KEY (`order_id`)
    REFERENCES `mall`.`order_detail` (`order_id`);

```

This is the orders table. The word ('order') is keyword in Mysql, thus the name (orders) is used for this table. The order table has general information with the status of order such as cancel, paid, fulfil, ship date. There are three foreign keys that buyer_id, payment_id, ship_id, and order_id. Each foreign key goes to each table (buyer_id -> buyer, payment_id -> payment, ship_id -> shipment, order_id -> order_detail). The type of columns (cancel, paid, fulfill) is INT since '0' means not yet, '1' means yes. Payment date and shipment date use DATE data type.

9. qna table

```
CREATE TABLE `mall`.`qna` (
  `qna_id` VARCHAR(10) NOT NULL,
  `qna_title` VARCHAR(255) NULL,
  `buyer_id` VARCHAR(10) NULL,
  `qna_date` DATE NULL,
  `qna_context` VARCHAR(2048) NULL,
  PRIMARY KEY (`qna_id`)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

ALTER TABLE `mall`.`qna`
ADD CONSTRAINT `qna_FK1`
  FOREIGN KEY (`buyer_id`)
    REFERENCES `mall`.`buyer` (`buyer_id`);

INSERT INTO `mall`.`qna`
(`qna_id`,
`qna_title`,
`buyer_id`,
`qna_date`,
`qna_context`)
VALUES
("q1", "Question for the product that I bought", "1", "2021-03-14", "When I can get my items?"),
("q2", "Question for the product that I bought", "2", "2021-02-14", "Can you explain more about the items?"),
("q3", "Question for the product that I bought", "3", "2021-01-14", "Can I get the item as soon as possible?"),
("q4", "Question for the product that I bought", "4", "2021-02-24", "It is too expensive."),
("q5", "Question for the product that I bought", "5", "2021-01-24", "When I can get my items?"),
("q6", "Question for the product that I bought", "6", "2021-03-30", "Can I refund?"),
("q7", "Question for the product that I bought", "7", "2021-01-20", "Can I cancel my order?"),
("q8", "Question for the product that I bought", "8", "2021-02-23", "When I can get my items?"),
("q9", "Question for the product that I bought", "9", "2021-03-10", "When I can get my items?"),
("q10", "Question for the product that I bought", "10", "2021-04-20", "Is the last item?");
```

The qna table has qna id, title, buyer id, qna date, and context. The foreign key is buyer_id thus we can easily find a person who writes the context.

b. PHP

The condb.php is the connection between PHP and Mysql.

```
<?php
$con=mysqli_connect("localhost","name","password","database name") or die(mysqli_error($con));
?>
```

Most pages except index, log-out pages, the begging of the php contraction is session_start() with require 'condb.php'


```
<?php
    session_start();
    require 'condb.php';
?>
```

The header.php includes session information, thus the web page show certain menus depending on conditions.

```
<?php
if(isset($_SESSION['email'])){
?>
<li><a href="products.php"> Product List</a></li>
<li><a href="order_list.php"> Order List</a></li>
<li><a href="buyer_list.php">Buyer List</a></li>
<li><a href="supplier_list.php">Supplier List</a></li>
<li><a href="qna.php">QnA List</a></li>
<li><a href="logout.php">Logout</a></li>
<?php
}else{
?>
<li><a href="login.php">Login</a></li>
<?php
}
?>
```

The login_submit.php, the php compare user input with data that in the database. In this case, the user input must match the admin information in Mysql. If the data is not in the database, pop up alert otherwise, go to index.php.

```
<?php
require 'condb.php';
session_start();
$email=mysqli_real_escape_string($con,$_POST['email']);

$password=mysqli_real_escape_string($con,$_POST['password']);

$user_authentication_query="select admin_id from admin where admin_id='$email' and admin_password='$password'";
$user_authentication_result=mysqli_query($con,$user_authentication_query) or die(mysqli_error($con));
$rows_fetched=mysqli_num_rows($user_authentication_result);
if($rows_fetched==0){
?>
<script>
    window.alert("Wrong ID or password");
</script>
<meta http-equiv="refresh" content="1; url=login.php" />
<?php
}else{
$row=mysqli_fetch_array($user_authentication_result);
$_SESSION['email']=$email;
$_SESSION['id']=$row['id'];
header('location: index.php');
}
?>
```

Buyer.php represent information from buyer table.

```

<?php
$product_check_query="select * from buyer";
$product_result=mysqli_query($con,$product_check_query);
while( $pd_row = mysqli_fetch_array( $product_result ) ) {
    echo '<tr><td>' . $pd_row[ 'buyer_id' ] . '</td><td>' . $pd_row[ 'buyer_name' ] . '</td><td>' . $pd_row[ 'buyer_phone' ] .
    '</td><td>' . $pd_row[ 'email' ] .
    '</td><td>' . $pd_row[ 'buyer_address' ] . '</td><td>' . $pd_row[ 'buyer_city' ]
    . '</td><td>' . $pd_row[ 'buyer_state' ] . '</td><td>' . $pd_row[ 'buyer_country' ] . '</td><td>' . $pd_row[ 'buyer_zip' ]
    . '</td></tr>';
}
?>

```

The order_list.php show certain data from orders table, and the information is ordered by order date. Furthermore, there is a sum of price and sum or quantities.

```

<?php
$order_check_query="select * from orders order by order_date ASC";
$order_result=mysqli_query($con,$order_check_query);
while( $order_row = mysqli_fetch_array( $order_result ) ) {
    echo '<tr><td>' . $order_row[ 'order_id' ] . '</td><td>' . $order_row[ 'buyer_id' ] . '</td><td>' . $order_row[ 'total_price' ] .
    '</td><td>' . $order_row[ 'total_quantity' ] .
    '</td><td>' . $order_row[ 'order_date' ] . '</td><td>' . $order_row[ 'paid' ] . '</td><td>' . $order_row[ 'fulfill' ] .
    '</td><td>' . $order_row[ 'ship_date' ] .
    '</td></tr>';
}
?>

```

```

<?php

$total_query="select SUM(total_price) from orders";
$total_result=mysqli_query($con,$total_query);
while ( $total = mysqli_fetch_array( $total_result ) ) {
    echo "<h3>". "Total price : " . $total[ 'SUM(total_price)' ] . "</h3>";
}

?>

```

```

<?php

$total_query="select SUM(total_quantity) from orders";
$total_result=mysqli_query($con,$total_query);
while ( $total = mysqli_fetch_array( $total_result ) ) {
    echo "<h3>". "Total Quantity(Sales) : " . $total[ 'SUM(total_quantity)' ] . "</h3>";
}

?>

```

In the product.php, the information is from product table and ordered by product name. There is a total product quantity sorted by product category.

```

<?php
$product_check_query="select * from product order by product_name";
$product_result=mysqli_query($con,$product_check_query);
while( $pd_row = mysqli_fetch_array( $product_result ) ) {
    echo '<tr><td>' . $pd_row[ 'product_id' ] . '</td><td>' . $pd_row[ 'product_name' ] . '</td><td>' . $pd_row[ 'product_category' ] .
    '</td><td>' . $pd_row[ 'product_price' ] .
    '</td><td>' . $pd_row[ 'product_quantity' ] . '</td><td>' . $pd_row[ 'supplier_id' ] . '</td></tr>';
}
?>

```

```
<?php

$total_query="select product_category, SUM(product_quantity) from product group by product_category";
$total_result=mysqli_query($con,$total_query);
while ($total = mysqli_fetch_array( $total_result)){
    echo "<h3>". "Total ". $total['product_category']. " = " . $total['SUM(product_quantity)'] . "</h3>";
}

?>
```

In the `supplier_list.php`, the two table which are supplier and product use and order by product category. In this page, you can check the lowest stock item's supplier id, how much left (product quantity), and who you need to contact (manager name).

```
<?php

$product_check_query="select * from supplier,product where supplier.supplier_id = product.supplier_id order by product.product_category";
$product_result=mysqli_query($con,$product_check_query);
while( $pd_row = mysqli_fetch_array( $product_result ) ) {
    echo "<tr><td>" . $pd_row[ 'supplier_id' ] . "</td><td>" . $pd_row[ 'company' ] . "</td><td>" . $pd_row[ 'contact_manager' ] .
    "</td><td>" . $pd_row[ 'contact_number' ] .
    "</td><td>" . $pd_row[ 'product_name' ] . "</td><td>" . $pd_row[ 'product_category' ] . "</td></tr>";
}

?>
```

```
<?php

$total_query="select * from supplier,product where supplier.supplier_id = product.supplier_id order by product.product_quantity asc limit 1";
$total_result=mysqli_query($con,$total_query);
while ($total = mysqli_fetch_array( $total_result)){
    echo "<h3>". "The lowest stock item is provided by " . $total['supplier_id'] . "</h3>";
    echo "<br>";
    echo "<h3>". "The quantity is " . $total['product_quantity'] . "</h3>";
    echo "<br>";
    echo "<h3>". "Need to contact to " . $total['contact_manager'] . "</h3>";
}

?>
```

In the `qna.php`, five tables are combined such as product, qna, buyer, orders, order_detail. The data is ordered by qna date, also you can check the what product has the number of questions.

```
<?php

$product_check_query="select * from qna,buyer,orders,order_detail,product
where qna.buyer_id = buyer.buyer_id and buyer.buyer_id = orders.buyer_id and orders.order_id = order_detail.order_id and
order_detail.product_id = product.product_id order by qna.qna_date asc ";
$product_result=mysqli_query($con,$product_check_query);
while( $pd_row = mysqli_fetch_array( $product_result ) ) {
    echo "<tr><td>" . $pd_row[ 'qna_id' ] . "</td><td>" . $pd_row[ 'buyer_id' ] . "</td><td>" . $pd_row[ 'qna_title' ] .
    "</td><td>" . $pd_row[ 'qna_date' ] .
    "</td><td>" . $pd_row[ 'qna_context' ] . "</td><td>" . $pd_row[ 'product_name' ] . "</td></tr>";
}

?>
```

```

<?php
$total_query="select product.product_name, SUM(product.product_name) from qna,buyer,orders,order_detail,product
where qna.buyer_id = buyer.buyer_id and buyer.buyer_id = orders.buyer_id and orders.order_id = order_detail.order_id
and order_detail.product_id = product.product_id group by product.product_name";
$total_result=mysqli_query($con,$total_query);
while ($total = mysqli_fetch_array( $total_result)){
    $total['SUM(product.product_name)'] = $total['SUM(product.product_name)']+1;
    echo "<h3>". "The number of Question for ".$total['product_name'], " is ".$total['SUM(product.product_name)']. "</h3>";
    echo "<br>";
}
?>

```

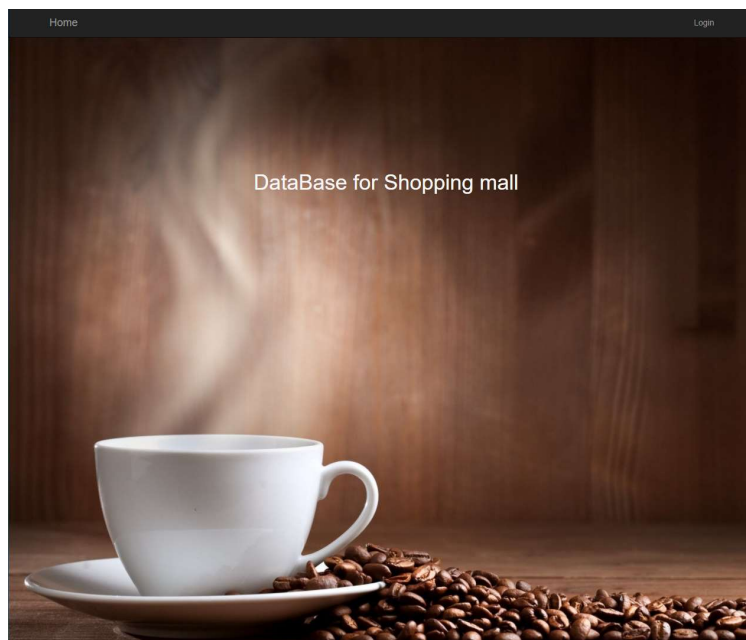
When you logout, the session is unset and destroyed.

```

<?php
    session_start();
    session_unset();
    session_destroy();
?>

```

This is some screen shots of web pages.



Home

Product List

Order List

Buyer List

Supplier List

QnA List

Logout

Supplier List

Supplier id	Company name	Company manager	Contact number	Product name	Product category
s5	Leg	James	521-321-1237	dryer	beauty
s9	Rain	Tran	665-096-1593	cream	beauty
s10	Real	Lucas	654-987-3210	hoodie	cloth
s4	Fam	Janet	321-735-1670	tshirt	cloth
s6	Wind	Alex	432-759-4679	shoe	cloth
s2	appa	William	404-123-4567	chocolatebox	food
s3	Mcdoo	Jenny	770-345-5678	caramelcandy	food
s8	Small	June	905-346-9853	snackbar	food
s1	Dust	Peter	404-321-6763	envelopment	life
s7	Big	George	404-902-7315	kitchenware	life

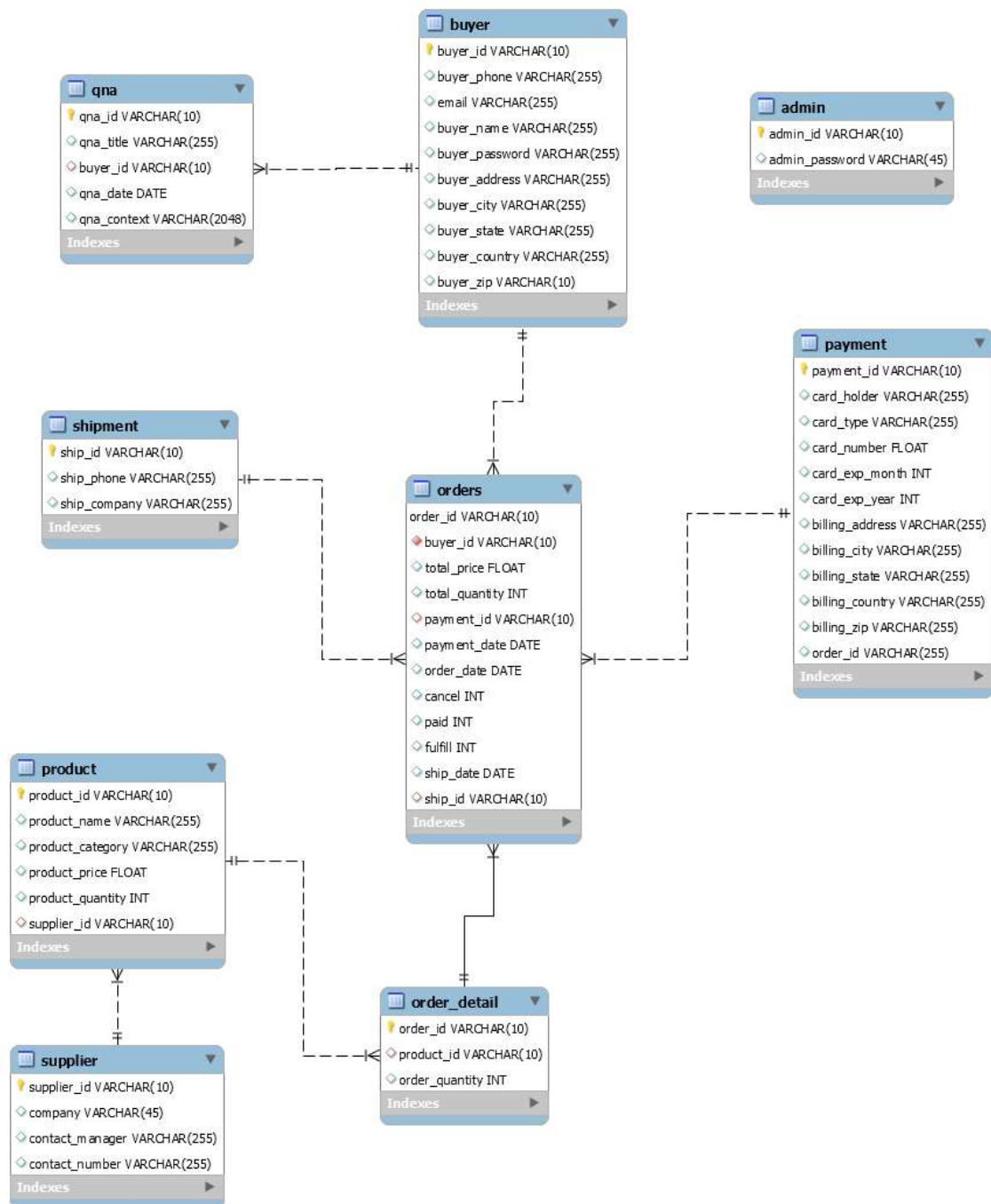
The lowest stock item is provided by s10

The quantity is 5

Need to contact to Lucas

4. ER diagram

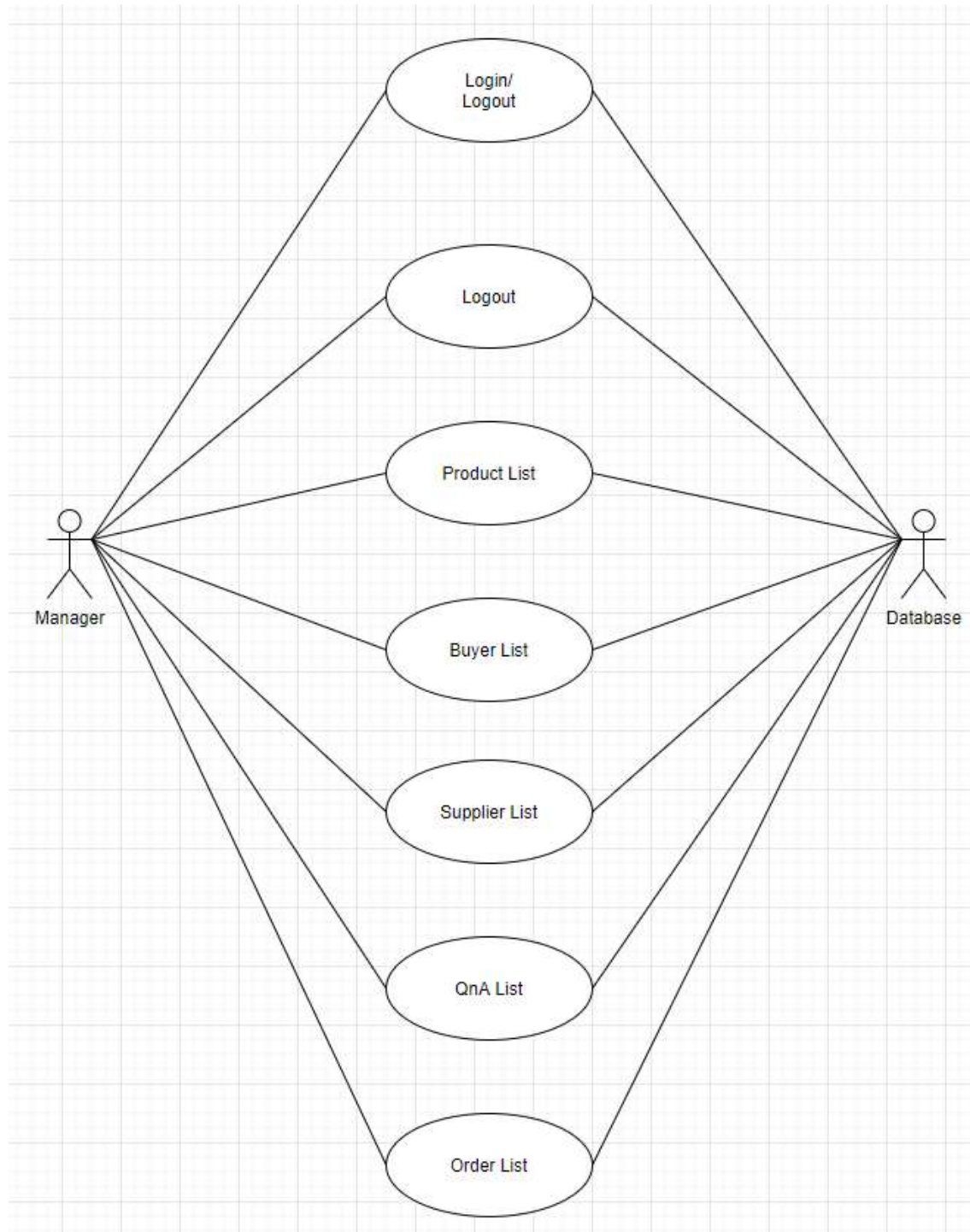
This is the ER diagram for the database.



The admin table is not related to any tables. Orders are connected with most tables. Qna table is just need the buyer's information thus only connects with buyer table. We need to retrieve orders, order_detail, product table to reach supplier table. The data type of each table's id is varchar since the id is combining numbers and letters.

5. Use case diagram

This is the use case diagram.



6. Conclusion

The most shopping mall database are focusing on order process or payment process however this database concentrates on managing products, suppliers and customer feedback. The schema's structure is that each table are related, thus the data can be managed efficiently. Some important information such as lowest stock item or which product has most questions is directly show in the web page. The manager is easy to administer the shopping mall. This database will be suitable where needs to manage and analyze between products, suppliers, customers.

7. References

a. *Bootswatch: Free themes for Bootstrap*. (2021). Bootswatch.

<https://bootswatch.com/>

b. *PHP: MySQL Database*. (2021). W3school.

https://www.w3schools.com/php/php_mysql_intro.asp