**E-Commerce Website Final Project**

**CMPT 586**

By: Chris Minda

**Business Plan**

**Business Plan:**

Group Members: Chris Minda (working alone)

-What are your products?

The products for the site are hockey equipment. The types of hockey equipment being sold on the site are gloves, helmets, jerseys, skates, and sticks.

-Who do you expect your client to be?

Customers for the site will be anyone looking to buy hockey equipment. Whether they are the seasoned veteran or the newbie looking to lace up skates for the first time, customers will find products and information about the products they are looking for.

-What access your various users should have to your product – define your four user groups and discuss in high level English what access they should have

Customer: The customer needs to have a unique account, see the products, the prices, customer reviews and be able to choose and order items from the site.

DBA: The DBA is able to create new users whether they be customers or employees. The DBA can also insert new items and categories of items for the website.

Shipping department: Employees see orders that need to be filled, the items that need to be sent out, and to what address they will be shipped to.

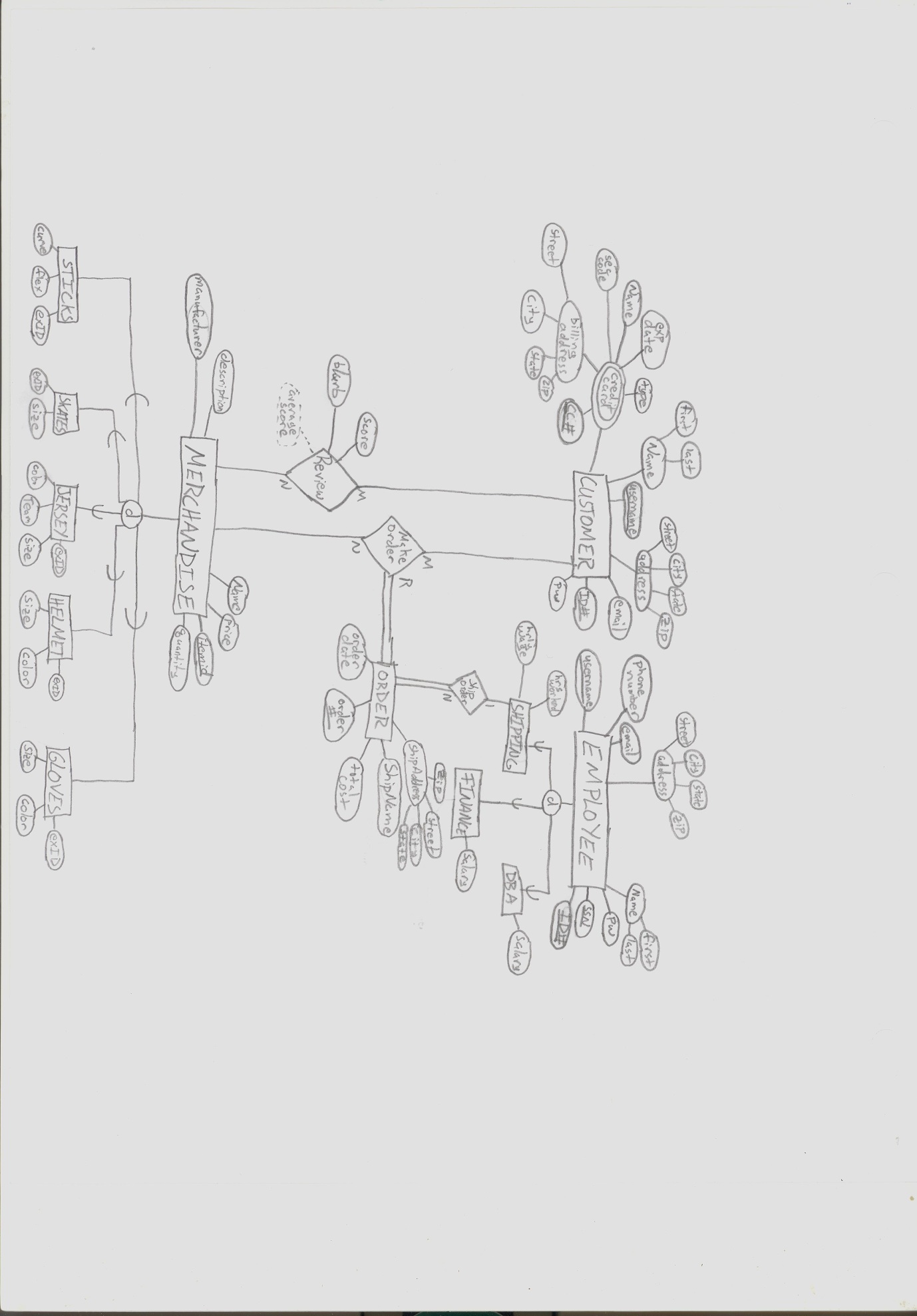
Finance Department: The finance department needs to be able to see the amount of money made and number of items sold so that they can decide on what needs to be restocked and what is working/not working for the business.

**Data Views**

**Website Views:**

* **CUSTOMER**
  + A customer can see the attributes of:
    - Usernames of self and others
    - All personal customer information aside from customerid and password for security issues.
    - Personal ORDERS: the orderid, totalPrice, orderdate, status, who the item is being shipped to and the address to where it is being shipped. All MERCHANDISE ordered by a customer in said ORDERS can be viewed in a table of the items ordered.
    - MERCHANDISE can be viewed and ordered by the CUSTOMER. The CUSTOMER can see any item and specialization whose quantity > 0. This includes the attributes of name, price, description, manufacturer, the category, and any specialization attributes of the MERCHANDISE.
    - MERCHANDISE reviews: the CUSTOMER can see and edit his/her own personal reviews as well as those others have written for MERCHANDISE.
* **DBA**
  + A database administrator can see the attributes of:
    - Can see all users and all of their attributes. This includes any CUSTOMERS and EMPLOYEES.
      * The only attribute that a DBA cannot access is the password and that is because it is encrypted with sha1.
    - The DBA can also insert/update any CUSTOMER or EMPLOYEE account.
    - Can see all items and insert/update all of them.
* **FINANCE**
  + A finance employee can see the attributes of:
    - Can update the quantity of any specialization of MERCHANDISE.
    - Can see all items and insert/update them. The finance department can also create new specializations of an item.
    - Can see all attributes of his/her EMPLOYEE account.
    - Can view and update any EMPLOYEE’s payment amount attribute.
* **SHIPPING**
  + A shipping employee can see the attributes of :
    - Can view all ORDERs and the MERCHANDISE in the ORDER.
    - Can view personal account page that holds all attributes of the EMPLOYEE account.
    - Can set the current status of any ORDER to ‘Shipping’

**Conceptual Data Model**



The Hockey Heaven website is meant to assist **CUSTOMERS** in buying hockey equipment.

* Each **CUSTOMER** has a customerid, username, name, address, password, and email.
* Name is a composite attribute made up of a firstName and lastName.
* Address is a composite attribute made up of a street, city, state, and zip code.
* The customerid number and username is unique to each **CUSTOMER**
* The password associated with **CUSTOMER** makes it possible for the **CUSTOMER** to login his/her account.
* Each customer has a multivalued, composite attribute of credit cards. The credit card attribute has a billing address, security code, expiration date, name, type and credit card number.
  + Billing address is a composed of a street, city, state, and zip code.
  + Credit card number is a unique value to each credit card.
  + The security code is a special code associated with each card to try and make card use more secure

There are a number of different **EMPLOYEES** who work for Hockey Heaven, making it possible to maintain the site, the finances, and ship items to **CUSTOMERS**.

* Each **EMPLOYEE** has an employeeid, username, ssn, name, address, password, email, and phone number.
* Name is a composite attribute made up of a firstName and lastName.
* Address is a composite attribute made up of street, city, state, and zip code.
* The employeeid, ssn, and username are all unique to each **EMPLOYEE**, each having a different purpose.
* The password associated with **EMPLOYEE** makes it possible for the **EMPLOYEE** to login to his/her account.
* Each **EMPLOYEE** can be further subdivided into **SHIPPING**, **FINANCE**, and **DBA**.
* The **DBA** controls and modifies the database.
  + Each **DBA** has a salary attribute.
* The **SHIPPER** checks orders, gathers them together, and sends the orders out.
  + Each **SHIPPER** has an hourly wage, and number of hours worked.
  + Each **SHIPPER** has a relation to **ORDER** where if an order has been shipped it will hold the **SHIPPER**’s unique ID and if it hasn’t it will hold an ID of 0.
* **FINANCE** keeps track of inventory and sales.
  + Each **FINANCE** employee has a salary attribute.

Hockey Heaven’s **MERCHANDISE** sold is a wide array of hockey equipment (gloves, helmets, jerseys, skates and sticks).

* Each piece of **MERCHANDISE** has an itemid, name, price, description, manufacturer, and quantity.
* Each piece of **MERCHANDISE** has a unique itemid.
* Each piece of **MERCHANDISE** can be further specialized to **GLOVES**, **HELMETS**, **JERSEYS**, **SKATES**, and **STICKS**.
* **GLOVES** are a type of **MERCHANDISE**
  + Each **GLOVE** has a **GLOVE** size, color, and exID.
  + The exID distinguishes a **GLOVE** with one color and size from a **GLOVE** with the same itemid but a different size and color.
* **HELMETS** are another type of **MERCHANDISE**.
  + Each **HELMET** has a **HELMET** size, color, and exID.
  + The exID distinguishes a **HELMET** with one color and size from a **HELMET** with the same itemid but a different size and color.
* **JERSEYS** are another type of **MERCHANDISE**.
  + Each **JERSEY** has a color, team, **JERSEY** size, and exID.
  + The exID distinguishes a **JERSEY** with one size, team, and color from a **JERSEY** with the same itemid but a different size, team, and color.
  + Color is whatever color the **JERSEY** is. If it is a team game **JERSEY**, it will be labeled as home, away, or alternate.
  + If the **JERSEY** is a practice jersey, it will be labeled as such in the team attribute, otherwise it will hold the team’s name.
* **SKATES** are another type of **MERCHANDISE**
  + Each **SKATE** has a bootsize and exID.
  + The exID distinguishes a **SKATE** with one size from a **SKATE** with the same itemid but a different size.
* **STICKS** are the final piece of **MERCHANDISE** sold by Hockey Heaven.
  + Each **STICK** has a flex, curve, and exID.
  + The exID distinguishes a **STICK** with one flex and curve from a **STICK** with the same itemid but a different flex and curve.

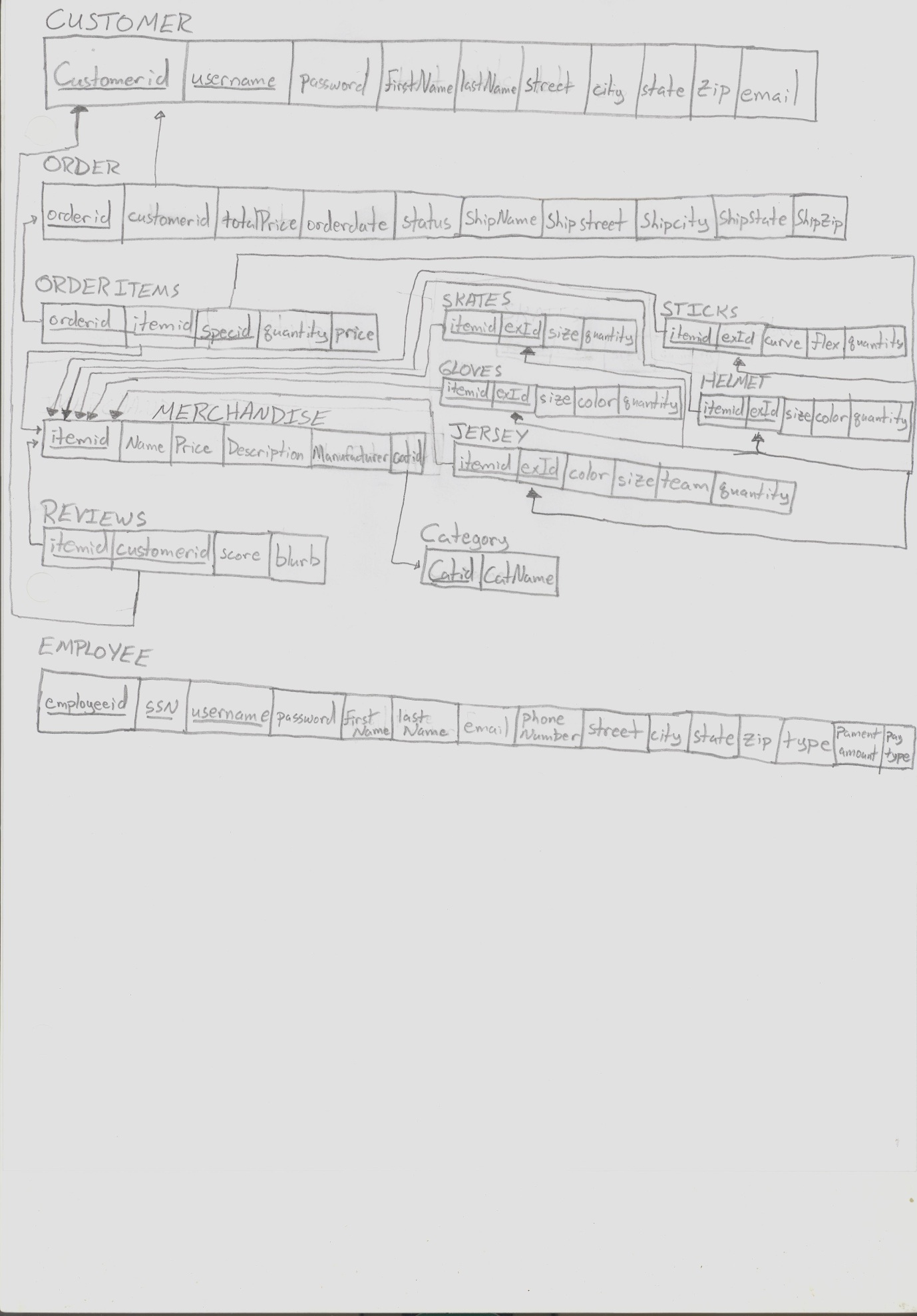
Each **CUSTOMER** can order a number of pieces of **MERCHANDISE**. The orders are organized in the **ORDER** entity.

* Each **ORDER** has a time of **ORDER**, an order number, status, shipName, shipAddress, and total cost of the **ORDER**.
* The order number is unique to each order.
* Each **ORDER** must have a **CUSTOMER** associated with it. This will be done through associating the **CUSTOMER**’s customerid with the order.
* Each **ORDER**’s shipName and shipAddress are to specify where the order is being shipped, in case a customer wants to ship somewhere other than their home address or want to ship an **ORDER** to a friend/family member.
* The status is updated to determine where in the process of processing, gathering, and shipping the **ORDER** is.

**MERCHANDISE** can have **reviews** associated with them that are written by **CUSTOMERS**.

* Each review has a score and blurb.
* Each review will be associated with a piece of **MERCHANDISE** through the itemid.
* Each review will be associated with a **CUSTOMER** through a customerid.
* The review for each item will have a calculated value of their average

**Physical Data Model**



The **CUSTOMER** table holds the attributes of customerid, username, password, firstName, lastName, street, city, state, zip, and email. This is the table that will hold the information for any customers of the Hockey Heaven website.

* The customerid is the key of CUSTOMER.
* The customerid is unique to each CUSTOMER entity and is generated at the time of account creation.
* The username is a unique name used to identify an account. This is mainly used to keep some anonymity for users when they review items.
* The password is a string between 6 and 16 characters. The password is encrypted by the sha1 function.
* The firstName and lastName attributes are the name of the CUSTOMER.
* The street, city, state, and zip are the address of the CUSTOMER. These are used as the default shipping address for any ORDERs that a user makes.

The **CATEGORY** table holds the attributes of catid and catname.

* The catid is the key of CATEGORY.
* The catid is a unique number id that is used in the MERCHANDISE table so items can be set to a specific CATEGORY of item.
* The catname is a unique name for each CATEGORY.

The **MERCHANDISE** table holds the attributes of itemid, name, price, description, manufacturer, and catid.

* The itemid is the key of MERCHANDISE.
* The itemid is a unique number that is generated so that
* The name is the name of the product for sale.
* The price holds a value of what the item will cost.
* The description holds a short description about the item.
* The manufacturer holds the name of the company who made the item.
* The catid holds the id number of the CATEGORY of item that the MERCHANDISE is.

The **GLOVES** table holds the attributes of itemid, exId, size, color, and quantity.

* The itemid and exId are the composite key of GLOVES.
* The itemid holds the unique id of the MERCHANDISE which the GLOVES are.
* The exId holds a unique id for the GLOVES entity to distinguish it from other GLOVES which are the same item but are of a different size/color.
* The size is the size of the GLOVES. The size of a pair of GLOVES is an integer value, usually falling between 13 and 15.
* The color is the color of the GLOVES. The color can hold multiple colors if the GLOVES have more than one color.
* The quantity is the number of items left in inventory of this specific item.

The **HELMETS** table holds the attributes of itemid, exId, size, color, and quantity.

* The itemid and exId are the composite key of HELMETS.
* The itemid holds the unique id of the MERCHANDISE which the HELMETS are.
* The exId holds a unique id for the HELMETS entity to distinguish it from other HELMETS which are the same item but are of a different size/color.
* The size attribute holds a string value that usually is expressed by a letter/letters S, M, L, XL or by a word Small, Medium, Large.
* The color is the color of the HELMETS. The color can hold multiple colors if the HELMETS have more than one color.
* The quantity is the number of items left in inventory of this specific item.

The **JERSEYS** table holds the attributes of itemid, exId, color, size, team, and quantity.

* The itemid and exId are the composite key of JERSEYS.
* The itemid holds the unique id of the MERCHANDISE which the JERSEYS are.
* The exId holds a unique id for the JERSEYS entity to distinguish it from other JERSEYS which are the same item but are of a different size/color/team.
* The size attribute holds a string value that usually is expressed by a letter/letters S, M, L, XL or by a word Small, Medium, Large, or Goalie.
* The color is the color of the JERSEY. The color can hold multiple colors if the JERSEY has more than one color. If a JERSEY is a team game JERSEY it will hold home, away, or alternate instead.
* The team attribute holds the name of the team for which the JERSEY is made. If it is a practice jersey it holds ‘practice’ for the team attribute.
* The quantity is the number of items left in inventory of this specific item.

The **SKATES** table holds the attributes of itemid, exId, size, and quantity.

* The itemid and exId are the composite key of SKATES.
* The itemid holds the unique id of the MERCHANDISE which the SKATES are.
* The exId holds a unique id for the SKATES entity to distinguish it from other SKATES which are the same item but are of a different size.
* The size is the size of the SKATES. The size of a pair of SKATES is an integer value, usually falling in the range of a normal pair of shoes.
* The quantity is the number of items left in inventory of this specific item.

The **STICKS** table holds the attributes of itemid, exId, curve, flex, and quantity.

* The itemid and exId are the composite key of STICKS.
* The itemid holds the unique id of the MERCHANDISE which the STICKS are.
* The exId holds a unique id for the STICKS entity to distinguish it from other STICKS which are the same item but are of a different flex/curve.
* The flex of a STICK is how stiff the stick is. The higher the number the stiffer the stick. Flex for most sticks fall between the values of 75 to 110, so an integer value holds the value for flex.
* The curve is the curve of the blade on the STICK. Most curves are given the name of the professional player’s blade curve which the blade is modeled after.
* The quantity is the number of items left in inventory of this specific item.

The **ORDER** table holds the attributes of orderid, customerid, totalPrice, orderdate, status, shipName, shipStreet, shipCity, shipState, and shipZip.

* The orderid is the key of ORDER.
* The customerid is a foreign key associating a CUSTOMER with the order.
* The orderid is the unique id of an ORDER made by a CUSTOMER.
* The totalPrice is the total amount that was spent on the ORDER.
* The orderdate is the date that the ORDER was made on.
* The status is the current state which the ORDER is in, for example it can be and currently is the only values of “Processing”, “Preparing”, or “Shipping”.
* The shipName, shipStreet, shipCity, shipState, and shipZip is the name and address of where the ORDER is to be shipped. This exists for the reason of if someone wants to ship an ORDER as a gift or to a different location than the address in CUSTOMER. The CUSTOMER names and address are the default values of these attributes.

The **ORDERITEMS** table holds the attributes of orderid, itemid, specid, quantity, and price.

* The the orderid, itemid, and specid are the composite key of ORDERITEMS.
* The orderid is the unique id of the ORDER which this item is associated with.
* The itemid is the unique id of the MERCHANDISE which has been ordered.
* The specid is the unique id of the specialization of the item ordered, so that a CUSTOMER will receive an item that has the correct size, color, etc. when they order MERCHANDISE.
* The quantity the amount of the item that has been ordered.
* Price is the price of the item.

The **REVIEWS** table holds the attributes of itemid, customerid, score, and blurb.

* The itemid and customerid are the composite key of a REVIEW.
* The itemid is the unique id of the MERCHANDISE which the review is associated with.
* The customerid is the unique id of the CUSTOMER which the review is associated with.
* The score is the score given by the customer. It holds a tinyint value between 0 and 5.
* The blurb is a short review for the item written by the customer.

The **EMPLOYEE** table holds the attributes of employeeid, ssn, username, password, firstName, lastName, email, phoneNumber, street, city, state, zip, type, payment type, and payment amount.

* The employeeid is the key of EMPLOYEE.
* The employeeid is the unique id of the EMPLOYEE generated upon creation of the EMPLOYEE.
* The ssn is a unique value holding the social security number of the employee.
* The username is a unique name for users to be used to login to the database.
* The password is the password used to login to the Hockey Heaven website. The password is encrypted with sha1.
* The firstName and lastName are the first and last name of the EMPLOYEE.
* The email is the email that can be used to contact the EMPLOYEE.
* The phoneNumber is the phone number that the EMPLOYEE can be contacted at.
* The street, city, state, and zip make up the address of the EMPLOYEE.
* The type is the type of account the EMPLOYEE has. The different values that this attribute currently holds is ‘admin’, ‘finance’, and ‘shipping’.
* The payment type is how the EMPLOYEEs pay is determined. The two types of values for this attribute are ‘hourly’ and ‘salary’.
* The payment amount is the amount that the EMPLOYEE should be paid. If his/her type of pay is ‘hourly’ then it is what the EMPLOYEE is paid per hour. If his/her type of pay is ‘salary’ then it is what the EMPLOYEE is paid yearly.

**Website Implementation**

**Website Implementation:**

Every page on the Hockey Heaven website is derived from the page class, which is setup to quickly and easily display a page for the site. The content variable associated with the page can be set to different values so that different information can be displayed on a page.

When a user tries to access the Hockey Heaven Website, the user must login to the database with a valid account. If the user does not login, he/she will be sent back to the login page and will be unable to view any of the websites pages.

There is no register option. Only admin’s can create new users in this implementation, so that no one outside of the website’s permission can access this site to order items since it is a dummy site meant to demonstrate implementation of a site that interfaces with a database.

When the user has logged in the $\_SESSION array will be set at keys ‘valid\_user’ with the account name and at ‘AccountType’ with the type of account that is accessing the site. A user will be welcomed with the confirmation that they have logged in.

On the home page a user can see a search bar and navigation bar that will help a user find their way around the site. On the front page, three random items for sale are displayed using the shuffle function on an array of all MERCHANDISE items in the database. A user can click on the items and view the item page for those items displayed on the front page.

The search bar searches all MERCHANDISE for the text that is input if the drop down menu for associated with the search is set to all. Otherwise, the search will be limited to the category specified in the dropdown menu.

At the top right corner of every page is the option to logout and the name of the user logged in. If logout is clicked, the user will be logged out, as well as the $\_SESSION array being unset. If the username is clicked the user will be redirected to an account page where he/she can view all of their personal information. Also, to the right of the page is a link to the cart of items a user intends to purchase. The cart will be discussed later.

The ‘customer’ user will see a navigation bar that will let them go to a list of all items that are any of the category types, ie. GLOVES, HELMETS, JERSEYS, SKATES, and STICKS.  They also have the option to Contact the company through an email form.

The ‘admin’, ‘finance’, and ‘shipping’ users all have the navigation bar of a ‘customer’ as well as navigation bars of their own. The ‘admin’ has buttons that lead him to a page for inserting items, customers, employees, and item categories. The ‘finance’ user has buttons that goes to a financial summary, a table for the companies payroll, and a table of the current stock of merchandise. The ‘shipping’ user has a navigation button to take him/her to a page of open orders.

An item page displays the photo of the item, the name, price, description, average review score  as well as a drop down menu of any of the specializations of the item which is displayed. The specializations of these items are only displayed to customers if they have a quantity greater than zero.

A ‘finance’ and ‘admin’ user can add new items as well as item specializations to the database. A ‘finance’ user also has the option to update the quantity of items that are available.

At the bottom of every item page is also the reviews that any user has written for the item. A ‘customer’ has the option to write a review for the item they are viewing. Upon clicking write a review, the user will be redirected to a form where they will be given a chance to give the item a score as well as write a short review for the item. As long as the form is filled out the review will be stored in the database with the itemid of the item and customerid of the writer of the review.

‘Admin’s also have the ability to create, update, and delete CUSTOMERS and EMPLOYEES from the database.  When generating a new user and the password is entered, it will be encrypted using the sha1 function. Same goes if a user changes his/her password.

Each query that is made is checked to return a result. If the variable $result of a query does not receive any value or it is false then the query did not work as intended, so the function that this query is implemented in will return false. Some queries also check to see that a result is returned with a specific number of rows.   
  
The cart holds the values of the total price, the number of items being purchased, and an array of the items that will be bought. The total price is stored under the key of ‘totalPrice’ and the number of items is stored in the ‘items’ key in the $\_SESSION array. The ‘totalPrice’ and ‘items’ values are calculated anytime there is a new item added to or the quantity is changed for an item in the cart with the calculatePrice and calculateItems functions respectively.  
  
Each item being purchased is stored as an Item object in the ‘cart’ array which is part of the $\_SESSION array. Each Item object stores the itemid, specId, and quantity being purchased. The key which each Item object is stored under in the ‘cart’ array is a combination of the itemid and specid of the MERCHANDISE being purchased. When the Item is stored in the cart array, it is “serialized” so it can be stored across the $\_SESSION array. This is done through the serialize function in php, which “generates a storable representation of a value…It is useful for storing or passing PHP values around without losing their type and structure.” When the Item is accessed it is “unserialized” with the php unserialize function and stored in as an item object so the values can be accessed.  
  
If a CUSTOMER views the ‘cart’ and it is empty they will receive a message saying the cart is empty, otherwise they will receive a table that displays the items that are queued up for purchase. A table row consists of a small picture of the item, the name of the item, the item specialization, the quantity being purchased and the total price for this item. After all items, have been displayed the total quantity of items being purchased and the total price of the order are displayed. At the bottom of the page, the CUSTOMER has the option to continue shopping or proceed to checkout to purchase the items.

**Database Issue Analysis**

**Database Issue Analysis:**  
  
**Redundant Data**:

There is redundant data in the database so that different tables can refer to other tables and access relevant data to a user.

Without any redundant data, a number of tables wouldn’t make sense. For example, REVIEWS wouldn’t refer back to the MERCHANDISE(itemid) or the CUSTOMER(customerid) that has made them. Same goes for CUSTOMERs(customerid), ORDERs(orderid), and the MERCHANDISE(itemid) making it impossible to determine what items are part of an order and who ordered these items.

**Accessibility to Data:**

Most data is reachable through the database. This is made possible through foreign keys referring records back to relevant record for example a CUSTOMER being able to see his/her own ORDERs they have made because each order holds a foreign key of the customerid of the CUSTOMER who placed the order.

The number for each CATEGORY type is not viewable since the name of the category is always displayed in its place.

Passwords aren’t directly accessible for any account types and if they were accessed would be returned as a string that is sha1 encrypted value.

**Database Normal Form:**

Most of the tables in the database are in 3rd Normal Form. Each table has a key or composite key which refers to the record as a whole.

The CUSTOMER table’s key customerid refers to every attribute of the customer and cannot be subdivided into other records without referring directly back to customerid.

The MERCHANDISE table’s key itemid refers to everything about the customer. There is a foreign key of catid which refers to the  CATEGORY which the item is a part of.

GLOVES, HELMETS, JERSEYS, SKATES, and STICKS are all specialized records of MERCHANDISE. Each one of these records holds the foreign key of itemid to refer it back to the MERCHANDISE it is. Every attribute outside of itemid is completely relevant and only refers  to the specialized item.

REVIEWS use a composite key of the itemid and customerid which without either of them the REVIEW would lose its value. This prevents multiple reviews being written by more than one user and making sure the review refers to the correct item.

ORDERs use a key of orderid. All values stored in the table directly relate to an order. ORDERs actually is further divided into ORDERITEMS so that it can store one to a large array of items

The only record that is in 2nd Normal Form is the EMPLOYEE table which has attributes of type referring to account type and payment type. To put this table could have been made to hold the different types of EMPLOYEEs and the type of payment they will receive (ie. All EMPLOYEEs working in ‘shipping’ will be paid hourly while EMPLOYEEs who are ‘admins’ and work for the ‘finance’ department are paid by salary).

**PHP Code**

*accountPage.php*

<?php

session\_start();

if(!isset($\_SESSION['valid\_user']))

{

header("Location: home.php");

exit;

}

require\_once('customerFunc.php');

require\_once('authFunc.php');

require\_once('item.php');

require\_once('page.php');

$page= new Page();

$db= dbConnect();

if(isset($\_SESSION['valid\_user']))

{

if($\_SESSION['AccountType'] == 'customer')

{

$query= "select \* from customer

where username=\"".$\_SESSION['valid\_user']."\"";

}

else

{

$query= "select \* from employees

where username=\"".$\_SESSION['valid\_user']."\"";

}

$result= $db->query($query);

if($result->num\_rows == 0)

{

$page->content= "Unable to access your account. Please try again later.";

}

else

{

$row= $result->fetch\_assoc();

//display information common among employees and customers

$page->content="

<h2>

".ucfirst($\_SESSION['valid\_user'])."'s Account Page:

</h2>

<a href=\"changePWPage.php\">Change Password</a>

<table>

<tr>

<th>

Name:

</th>

<td>

".$row['firstName']." ".$row['lastName']."

</td>

</tr>

<tr>

<th>

Address:

</th>

<td>

".$row['street']." <br />

".$row['city'].", ".$row['st']." ".$row['zip']."<br />

<a href=\"changeaddress.php\">Change Address</a>

</td>

</tr>

<tr>

<th>

Email:

</th>

<td>

".$row['email']."<br />

<a href=\"changeemail.php\">Change Email</a>

</td>

</tr>";

if(isAdmin() || isShipping() || isFinance())

{

$page->content.="<tr>

<th>Salary: </th>

<td>\$".$row['paymentamount']."</td>

</tr>";

}

$page->content.= "<tr>

<th></th>

<td>

<a href=\"changePWPage.php\">Change Password</a>

</td>

</tr></table>";

if($\_SESSION['accountType'] == 'customer')

{

//extra information for customers

$page->content.="<h2><a href=\"ordersPage.php\">View Your Orders</a></h2> <br /> <br />";

$page->content.=getReviews($row['customerid']);

}

}

}

else

{

$page->content= "Sorry you are currently not logged in and cannot view this page.";

}

$page->Display();

?>

*addstock.php*

<?php

session\_start();

require\_once('page.php');

require\_once('itemFunc.php');

$page= new Page();

if(!updateItemStock($\_GET['itemid'], $\_GET['extras'], $\_GET['qty']))

{

$page->content= "Unable to update item stock. Please try again later.";

}

else

{

$page->content= "Stock updated.";

}

$page->Display();

?>

*admin.php*

<?

session\_start();

require\_once("page.php");

require\_once("authFunc.php");

require\_once('item.php');

$page= new Page();

if(isAdmin())

{

$page->content= "Welcome admin";

}

else

{

$page->content= "You are not authorized to enter the administration area.";

}

$page->Display();

?>

*admincategory.php*

<?php

session\_start();

require('page.php');

require('adminFunc.php');

$page= new Page();

$page->content= CategoryForm();

$page->Display();

?>

*admincustomer.php*

<?php

session\_start();

require\_once('adminFunc.php');

require\_once('page.php');

$page= new Page();

$user= getCustomer($\_GET['id']);

$page->content.="<h3>Customer</h3> <table><td><a href= \"viewCustomer.php\">View Customers</a></td></table>". CustomerForm($user)."";

$page->Display();

?>

*adminemployee.php*

<?php

session\_start();

require\_once('adminFunc.php');

require('page.php');

$page= new Page();

$user= getEmployee($\_GET['id']);

$page->content.="<h3>Employee</h3><table><td><a href=\"viewEmployees.php\">Employee List</a></td></table>". EmployeeForm($user)."";

$page->Display();

?>

*adminFunc.php*

<?

require\_once('itemFunc.php');

require\_once("dbFunc.php");

function AddItem($itemid, $name, $manufacturer, $catid, $price, $description)

{

$db= dbConnect();

$query= "select \* from merchandise

where itemid=".$itemid."";

$result= $db->query($query);

if(($result->num\_rows!=0))

{

echo "itemid error";

return false;

}

/\*$query= "select catid from categories

where catname=\"".$catname."\"";

$result= $db->query($query);

if((!result) || ($result->num\_rows!=0))

{

return false;

}

$row= $result->fetch\_assoc();

$catid= $row['catid'];\*/

//insert item

$query="insert into merchandise values

(".$itemid.",

\"".$name."\",

\"".$manufacturer."\",

".$catid.",

".$price.",

\"".$description."\")";

$result= $db->query($query);

if(!$result)

{

return false;

}

else

{

return true;

}

}

function AddCategory($catname)

{

$db= dbConnect();

$query= "select \* from categories

where catname=\"".$catname."\"";

$result= $db->query($query);

if((!$result) || ($result->num\_rows!=0))

{

return false;

}

$query= "insert into categories values

catname=\"".$catname."\"";

}

function insertSpecForm($itemid)

{

$item= getItemDetails($itemid);

$cat= getCategoryName($item['catid']);

$db= dbConnect();

$query= "select \*

from ".strtolower($cat)."";

$result= $db->query($query);

if(!$result)

{

return "Unable to find item at this time.";

}

$row= $result->fetch\_assoc();

$form="

Insert new spec for ".$item['name']."

<table><form action=\"insertSpec.php\">

<input type=\"hidden\" name=\"catname\" value=\"".$cat."\">";

foreach($row as $key=>$value)

{

if($key!= 'itemid')

$form.="<tr><th>".$key.":</th><td><input type=\"text\" name=\"".$key."\"></td></tr>";

else

$form.="<input type=\"hidden\" name=\"itemid\" value=".$itemid.">";

}

$form.="<tr><td colspan=\"2\"><input type=\"submit\" value=\"Insert Spec\"></form></td></tr>";

return $form;

}

function insertSpec($newSpec)

{

$db= dbConnect();

$insertValues="";

$insertKeys="";

$i= 1;

foreach($newSpec as $key=>$value)

{

if($key != 'catname')

{

if($i!= 1)

{

$insertKeys.=", ";

$insertValues.=", ";

}

$insertKeys.= $key;

if(is\_float($value) || is\_int($value))

$insertValues.= $value;

else

$insertValues.="\"".$value."\"";

$i= 2;

}

}

$query= "insert into ".strtolower($newSpec['catname'])." (".$insertKeys.")

values ( ".$insertValues." )";

$result=$db->query($query);

echo mysql\_error();

if(!$result)

{

return false;

}

return true;

}

function AddEmployee($username, $pw, $ssn, $type, $firstname, $lastname, $street, $city, $state, $zip, $email, $phone, $paymentamount)

{

$db= dbConnect();

$query= "select \*

from employees

where ssn= \"".$ssn."\" or username=\"".$username."\"";

$result= $db->query($query);

if((!$result) || $result->num\_rows >0)

{

return false;

}

if($type == 'shipper')

{

$paymentType= "hourly";

}

else

{

$paymentType= 'salary';

}

//insert item

$query="insert into employees set

username=\"".$username."\",

ssn=\"".$ssn."\",

category=\"".$type."\",

firstName=\"".$firstname."\",

lastName=\"".$lastname."\",

street=\"".$street."\",

city=\"".$city."\",

st=\"".$state."\",

zip=\"".$zip."\",

email=\"".$email."\",

phone=\"".$phone."\",

password=\"".sha1($pw)."\",

paymentamount=".$paymentamount.",

paymenttype=\"".$paymentType."\"";

$result= $db->query($query);

if(!$result)

{

return false;

}

else

{

return true;

}

}

function AddCustomer($username, $pw, $firstName, $lastName, $street, $city, $state, $zip, $email)

{

$db= dbConnect();

$query= "select \* from customer

where username=\"".$username."\"";

$result= $db->query($query);

if($result->num\_rows!=0)

{

return false;

}

//insert item

$query="insert into customer set

username=\"".$username."\",

firstName=\"".$firstName."\",

lastName=\"".$lastName."\",

street=\"".$street."\",

city=\"".$city."\",

st=\"".$state."\",

zip=\"".$zip."\",

email=\"".$email."\",

password=\"".sha1($pw)."\"";

$result= $db->query($query);

if(!$result)

{

return false;

}

else

{

return true;

}

}

function UpdateItem($olditemid, $itemid, $name, $manufacturer, $catid, $price, $description)

{

$db= dbConnect();

$query="UPDATE merchandise

set itemid= ".$itemid.",

name= \"".$name."\",

manufacturer= \"".$manufacturer."\",

catid=".$catid.",

price= ".$price.",

description= \"".$description."\"

where itemid= ".$olditemid."";

$result= $db->query($query);

echo mysql\_error();

if(!$result)

{

return false;

}

else

{

return true;

}

}

function UpdateEmployee($employeeid, $street, $city, $state, $zip, $email, $phone, $paymentamount)

{

$db= dbConnect();

$query="update employees set

street=\"".$street."\",

city=\"".$city."\",

st=\"".$state."\",

zip=\"".$zip."\",

email=\"".$email."\",

phone=\"".$phone."\",

paymentamount=".$paymentamount."

where employeeid=".$employeeid."";

$result= $db->query($query);

if(!$result)

{

return false;

}

else

{

return true;

}

}

function UpdateCustomer($customerid, $username, $street, $city, $state, $zip, $email)

{

$db= dbConnect();

$query="update customer set

username=\"".$username."\",

street=\"".$street."\",

city=\"".$city."\",

st=\"".$state."\",

zip=\"".$zip."\",

email=\"".$email."\"

where customerid=".$customerid."";

$result= $db->query($query);

if(!$result)

{

return false;

}

else

{

return true;

}

}

function UpdateCategory($catid, $catname)

{

$db= dbConnect();

$query= "update categories

set catname=\"".$catname."\"

where catid=".$catid."";

$result= $db->query($query);

if(!$result)

{

return false;

}

else

{

return true;

}

}

function DeleteItem($itemid)

{

//deletes item identified by $itemid from the database

$db= dbConnect();

$query= "delete from merchandise

where itemid=".$itemid."";

$result= $db->query($query);

if(!$result)

{

return false;

}

else

{

return true;

}

}

function DeleteItemSpecialization($itemid, $exid, $catname)

{

//finds all of the specializations

$db= dbConnect();

$query= "delete from ".$catname."

where itemid=".$itemid." and exID=\"".$exid."\"";

$result= $db->query($query);

if(!$result)

{

return false;

}

else

{

return true;

}

}

function DeleteUser($id, $accountType)

{

$db= dbConnect();

if($accountType== 'customer')

{

$query= "delete from customer

where customerid=".$id."";

}

else

{

$query= "delete from employees

where employeeid=".$id."";

}

$result= $db->query($query);

if(!$result)

{

return false;

}

else

{

return true;

}

}

function DeleteCategory($catname)

{

$db= dbConnect();

$query= "delete from categories

where catname\"=".$catname."\"";

$result= $db->query($query);

if(!$result)

{

return false;

}

else

{

return true;

}

}

function MerchForm($item= '')

{

$file= $item? 'editMerch.php': 'insertMerch.php';

$itemid= $item?$item['itemid']:'';

$name= $item?$item['name']:'';

$manufacturer= $item?$item['manufacturer']:'';

$cost= $item?$item['price']:'';

$desc= $item?$item['description']:'';

$submission= $item? 'Update': 'Add';

$edit= is\_array($item);

//if passed an existing item, proceed in edit mode

$form= "<form method=\"post\"

action=\"".$file."\">

<table border=\"0\">

<tr>

<td>ItemID:</td>

<td><input type=\"text\" name=\"itemid\"

value=".$itemid."> </td>

</tr>

<tr>

<td>Item Name:</td>

<td><input type=\"text\" name=\"name\"

value=\"".$name."\"> </td>

</tr>

<tr>

<td>Manufacturer:</td>

<td><input type=\"text\" name=\"manufacturer\"

value=\"".$manufacturer."\"> </td>

</tr>

<tr>

<td>Category:</td>

<td><select name=\"catid\">";

$result=getCategories();

$num\_cat= $result->num\_rows;

for($i=0; $i<$num\_cat; $i++)

{

$row= $result->fetch\_assoc();

$form.= "<option value=".$row['catid']."";

$form.=">".$row['catname']."</option>";

}

$form.="</tr>

<tr>

<td>Price:</td>

<td><input type=\"text\" name=\"price\"

value=\"".$cost."\"> </td>

</tr>

<tr>

<td>Description:</td>

<td><textarea rows=\"3\" cols=\"50\" name=\"description\">

".$desc."</textarea>

</tr>

<tr>";

if(!$edit)

{

$form.="<td colspan=\"2\" align=\"center\">";

}

else

{

$form.="<td align=\"center\"><input type=\"hidden\" name=\"olditemid\"

value=".$item['itemid']." />";

}

$form.="<input type=\"submit\" value=\"".$submission." Item\" />

</form></td>";

if($edit)

{

$form.="<td>

<form method=\"post\" action=\"deleteItem.php\">

<input type=\"hidden\" name=\"itemid\"

value=".$item['itemid']." />

<input type=\"submit\" value=\"Delete item\" />

</form></td>";

}

$form.="</td>

</tr>

</table>";

return $form;

}

function CategoryForm($cat= '')

{

//$edit= is\_array($cat);

$form="";

$submitTo=$cat? 'editCat.php':'insertCat.php';

$name= $cat?$cat['catname']:'';

//if passed an existing item, proceed in edit mode

$form.= "<form method=\"post\"

action=\"".$submitTo."\">

<table border=\"0\">

<tr>

<td>Category Name:</td>

<td><input type=\"text\" name=\"catname\"

value=\"".$cat['catname']."\"> </td>

</tr>

<tr>";

if(!$edit)

{

$form.="<td colspan=\"2\" align=\"center\">";

}

else

{

$form.="<td align=\"center\"><input type=\"hidden\" name=\"catid\"

value=\"".$cat['catid']."\" />";

}

$submission= $edit? 'Update': 'Add';

$form.="<input type=\"submit\" value=\"".$submission." Category \"/>

</form></td>";

if($edit)

{

$form.="<td>

<form method=\"post\" action=\"deleteCat.php\">

<input type=\"hidden\" name=\"catid\"

value=\"".$category['catid']."\" />

<input type=\"submit\" value=\"Delete Category\" />

</form></td>";

}

$form.="</td>

</tr>

</table>";

return $form;

}

function CustomerForm($user= '')

{

$edit= is\_array($user);

$file= $edit? 'editCustomer.php': 'insertCustomer.php';

$username= $edit?$user['username']:'';

$firstName= $edit?$user['firstName']:'';

$lastName= $edit?$user['lastName']:'';

$street= $edit?$user['street']:'';

$city= $edit?$user['city']:'';

$state= $edit?$user['st']:'';

$zip= $edit?$user['zip']:'';

$email=$edit?$user['email']:'';

$pass=$edit?$user['password']:'';

//if passed an existing item, proceed in edit mode

$form= "<form method=\"post\"

action=\"".$file."\">

<table border=\"0\">

<tr>

<td>Username:</td>

<td><input type=\"text\" name=\"username\"

value=\"".$username."\"> </td>

</tr>

<tr>

<td>First Name:</td>

<td><input type=\"text\" name=\"firstName\"

value=\"".$firstName."\"> </td>

</tr>

<tr>

<td>Last Name:</td>

<td><input type=\"text\" name=\"lastName\"

value=\"".$lastName."\"> </td>

</tr>

<tr>

<td>Street:</td>

<td><input type=\"text\" name=\"street\"

value=\"".$street."\"> </td>

</tr>

<tr>

<td>City:</td>

<td><input type=\"text\" name=\"city\"

value=\"".$city."\"> </td>

</tr>

<tr>

<td>State:</td>

<td><input type=\"text\" name=\"state\"

value=\"".$state."\"> </td>

</tr>

<tr>

<td>Zip Code:</td>

<td><input type=\"text\" name=\"zip\"

value=\"".$zip."\"> </td>

</tr>

<tr>

<td>E-mail:</td>

<td><input type=\"text\" name=\"email\"

value=\"".$email."\"> </td>

</tr>";

if(!$edit)

{

$form.="<tr>

<td>Password:</td>

<td><input type=\"text\" name=\"pw\"

value=\"\"> </td>

</tr>";

}

if(!$edit)

{

$form.="<td colspan=\"2\" align=\"center\">";

}

else

{

$form.="<td align=\"center\"><input type=\"hidden\" name=\"customerid\"

value=".$user['customerid']." />";

}

$submission= $edit?'Update':'Add';

$form.="<input type=\"submit\" value=\"".$submission." Customer\" />

</form></td>";

if($edit)

{

$form.="<td>

<form method=\"post\" action=\"deleteCustomer.php\">

<input type=\"hidden\" name=\"customerid\"

value=".$user['customerid']." />

<input type=\"submit\" value=\"Delete Customer\" />

</form></td>";

}

$form.="</td>

</tr>

</table>";

return $form;

}

function EmployeeForm($user= '')

{

$edit= is\_array($user);

$file= $edit? 'editEmployee.php': 'insertEmployee.php';

$username= $edit?$user['username']:'';

$cat= $edit?$user['category']:'';

$firstName= $edit?$user['firstName']:'';

$lastName= $edit?$user['lastName']:'';

$ssn= $edit?$user['ssn']:'';

$street= $edit?$user['street']:'';

$city= $edit?$user['city']:'';

$state= $edit?$user['st']:'';

$zip= $edit?$user['zip']:'';

$email=$edit?$user['email']:'';

$phone=$edit?$user['phone']:'';

$paymentamount=$edit?$user['paymentamount']:'';

$pass=$edit?$user['password']:'';

//if passed an existing item, proceed in edit mode

$form= "<form method=\"post\"

action=\"".$file."\">

<table border=\"0\">

<tr>

<td>Username:</td>

<td><input type=\"text\" name=\"username\"

value=\"".$username."\"> </td>

</tr>

<tr>

<td>Account Type:</td>

<td>

<select name=\"category\">

<option value=\"admin\">Administrator</option>

<option value=\"shipping\">Shipping</option>

<option value=\"finance\">Finance<option>

</select>

</td>

</tr>

<tr>

<td>First Name:</td>

<td><input type=\"text\" name=\"firstName\"

value=\"".$firstName."\"> </td>

</tr>

<tr>

<td>Last Name:</td>

<td><input type=\"text\" name=\"lastName\"

value=\"".$lastName."\"> </td>

</tr>

<tr>

<td>SSN:</td>

<td><input type=\"text\" name=\"ssn\"

value=\"".$ssn."\"> </td>

</tr>

<tr>

<td>Street:</td>

<td><input type=\"text\" name=\"street\"

value=\"".$street."\"> </td>

</tr>

<tr>

<td>City:</td>

<td><input type=\"text\" name=\"city\"

value=\"".$city."\"> </td>

</tr>

<tr>

<td>State:</td>

<td><input type=\"text\" name=\"state\"

value=\"".$state."\"> </td>

</tr>

<tr>

<td>Zip Code:</td>

<td><input type=\"text\" name=\"zip\"

value=\"".$zip."\"> </td>

</tr>

<tr>

<td>E-mail:</td>

<td><input type=\"text\" name=\"email\"

value=\"".$email."\"> </td>

</tr>

<tr>

<td>Phone:</td>

<td><input type=\"text\" name=\"phone\"

value=\"".$phone."\"> </td>

</tr>

<tr>

<td>Payment Amount:</td>

<td><input type=\"text\" name=\"paymentamount\"

value=\"".$paymentamount."\"> </td>

</tr>

<tr>

<td>Password:</td>

<td><input type=\"text\" name=\"pw\"

value=\"".$pass."\"> </td>

</tr>";

if(!$edit)

{

$form.="<td colspan=\"2\" align=\"center\">";

}

else

{

$form.="<td align=\"center\"><input type=\"hidden\" name=\"employeeid\"

value=".$user['employeeid']." />";

}

$submission= $edit?'Update':'Add';

$form.="<input type=\"submit\" value=\"".$submission." Employee \"/>

</form></td>";

if($edit)

{

$form.="<td>

<form method=\"post\" action=\"deleteEmployee.php\">

<input type=\"hidden\" name=\"employeeid\"

value=".$user['employeeid']." />

<input type=\"submit\" value=\"Delete Employee\" />

</form></td>";

}

$form.="</td>

</tr>

</table>";

return $form;

}

function getEmployee($id)

{

$db= dbConnect();

$query= "select \* from employees

where employeeid=".$id."";

$result= $db->query($query);

if(!$result)

{

return false;

}

$emp= $result->fetch\_assoc();

return $emp;

}

function getEmployees()

{

$db= dbConnect();

if($\_SESSION['AccountType']=='admin')

{

$destination="\"adminemployee.php\"";

}

else if($\_SESSION['AccountType']=='finance')

{

$destination="\"financeEmployee.php\"";

}

$query= "select \* from employees";

$result= $db->query($query);

if(!$result)

{

return false;

}

$employees= "";

while($emp= $result->fetch\_assoc())

{

$employees.="<table widht=\"40\%\">

<tr><th>Employee ".$emp['employeeid'].": ".$emp['firstName']." ".$emp['lastName']."</th>

<td><form action=".$destination."> <input type=\"hidden\" name=\"id\" value=".$emp['employeeid']."><input type=\"submit\" value=\"Edit Employee\"></form></td></tr>

<tr><th>Username:</th><td>".$emp['username']."</td></tr>

<tr><th>Account Type:</th><td>".$emp['category']."</td></tr>

<tr><th>Address:</th><td>".$emp['street']."<br />

".$emp['city'].", ".$emp['st']." ".$emp['zip']."</td></tr>

<tr><th>Email:</th><td>".$emp['email']."</td></tr>

<tr><th>Phone:</th><td>".$emp['phone']."</td></tr>

<tr><th>".$emp['paymenttype']."</th><td>".$emp['paymentamount']."</td></tr>

</table> <br />";

}

return $employees;

}

function getCustomer($id)

{

$db= dbConnect();

$query= "select \* from customer

where customerid=".$id."";

$result= $db->query($query);

if(!$result)

{

return false;

}

$c= $result->fetch\_assoc();

return $c;

}

function getCustomers()

{

$db= dbConnect();

$query= "select \* from customer";

$result= $db->query($query);

if(!$result)

{

echo "uable to execute query";

return false;

}

$customers= "";

while($c= $result->fetch\_assoc())

{

$customers.="<table widht=\"40\%\">

<tr><th>Customer ".$c['customerid'].": ".$c['firstName']." ".$c['lastName']."</th>

<td><form action=\"admincustomer.php\"> <input type=\"hidden\" name=\"id\" value=".$c['customerid']."><input type=\"submit\" value=\"Edit Customer\"></form></td></tr>

<tr><th>Username:</th><td>".$c['username']."</td></tr>

<tr><th>Address:</th><td>".$c['street']."<br />

".$c['city'].", ".$c['st']." ".$c['zip']."</td></tr>

<tr><th>Email:</th><td>".$c['email']."</td></tr>

</table> <br />";

}

return $customers;

}

?>

*adminitem.php*

<?php

session\_start();

require\_once('page.php');

require\_once('adminFunc.php');

require\_once('itemFunc.php');

$page= new Page();

$id= $\_GET['item'];

$item= getItemDetails($id);

$page->content= MerchForm($item);

$page->Display();

?>

*authFunc.php*

<?php

require\_once('dbFunc.php');

function login($username, $pwd, $accountType)

{

$db= dbConnect();

//check username and password

if($accountType == 'customer')

{

$query= "select \* from customer

where username=\"".$username."\" and

password= \"".sha1($pwd)."\"";

}

else

{

$query= "select \* from employees

where username=\"".$username."\" and

password=\"".sha1($pwd)."\" and

category= \"".$accountType."\"";

}

$result=$db->query($query);

if(!$result)

{

return false;

}

if($result->num\_rows == 0)

{

return false;

}

else

{

return true;

}

}

function isAdmin()

{

if($\_SESSION['AccountType']== 'admin')

{

return true;

}

else

{

return false;

}

}

function isShipping()

{

if($\_SESSION['AccountType'] == 'shipping')

{

return true;

}

else

{

return false;

}

}

function isFinance()

{

if($\_SESSION['AccountType'] == 'finance')

{

return true;

}

else

{

return false;

}

}

function loginform()

{

return "<form action= \"dblogin.php\" method=\"post\">

Username:<br/>

<input type=\"text\" name=\"username\"><br/>

Password:<br/>

<input type=\"password\" name=\"pwd\"><br/>

Account Type:

<select name=\"accountType\">

<option value=\"customer\">Customer</option>

<option value=\"shipping\">Shipping</option>

<option value=\"finance\">Finance</option>

<option value=\"admin\">Administrator</option>

</select>

<input type=\"submit\" value=\"Log In\">

</form>";

}

function changePW($username, $oldPw, $newPw, $accountType)

{

//change the password from old to new

if(login($username, $oldPw, $accountType))

{

$db= dbConnect();

if(!$db)

{

return false;

}

if($accountType == 'customer')

{

$query="update customer

set password = \"".sha1($newPw)."\"

where username= \"".$username."\"

and password=\"".sha1($oldPw)."\"";

$result= $db->query($query);

if(!$result)

{

return false; //not changed

}

else

{

return true; //changed

}

}

else

{

$query="update employees

set password=\"".sha1($newPw)."\"

where username=\"".$username."\"

and password=\"".sha1($oldPw)."\"";

$result= $db->query($query);

if(!$result)

{

return false;

}

else

{

return true;

}

}

}

else

{

echo "unable to login";

return false; //wrong oldPw

}

}

function changePWForm()

{

return "<form action= \"changePW.php\">

<table>

<tr>

<td class=\"graybg\">

New password

</td>

<td class=\"graybg\">

<input type=\"password\" name=\"newPW\">

</td>

</tr>

<tr>

<td class=\"graybg\">

Re-enter new password

</td>

<td class=\"graybg\">

<input type=\"password\" name=\"newPW2\">

</td>

</tr>

<td class=\"graybg\">

Old Password

</td>

<td class=\"graybg\">

<input type=\"password\" name=\"oldPW\">

</td>

</table>

<input type=\"submit\" value=\"Change\">

</form>";

}

?>

*bibliography.php*

<?php

session\_start();

require\_once("page.php");

$page= new Page();

$page->content= "<h3>Citations:</h3>

<b>Inspired by:</b> <br />

• \"Hockey Equipment - Hockey Sticks, Hockey Skates, Ice Hockey Skates, Inline Hockey Skates and More Hockey Gear.\" Hockey Equipment - Hockey Sticks, Hockey Skates, Ice Hockey Skates, Inline Hockey Skates and More Hockey Gear. N.p., n.d. Web 21 Oct. 2012. <www.hockeygiant.com> <br /><br />

• \"Hockey Equipment & Gear | Hockey Sticks & Skates | Ice Hockey Helmets & Gloves | HockeyMonkey.com.\" Hockey Equipment & Gear | Hockey Sticks & Skates | Ice Hockey Helmets & Gloves | HockeyMonkey.com. N.p., n.d. Web. 21 Oct. 2012.<www.hockeymonkey.com> <br /> <br />

• \"Total Hockey Stores | Hockey Sticks, Hockey Skates, Apparel & More.\" Total Hockey Stores | Hockey Sticks, Hockey Skates, Apparel & More. N.p., n.d. Web. 15 Dec. 2012. <www.totalhockey.com > <br /> <br />

<br />

Images from: <br />

<b>All hockey equipment images used from:</b> <br />

• \"Hockey Equipment - Hockey Sticks, Hockey Skates, Ice Hockey Skates, Inline Hockey Skates and More Hockey Gear.\" Hockey Equipment - Hockey Sticks, Hockey Skates, Ice Hockey Skates, Inline Hockey Skates and More Hockey Gear. N.p., n.d. Web 21 Oct. 2012. <www.hockeygiant.com> <br /><br />

<b>Header Image:</b><br />

• Pond\_hockey.jpg. N.d. Photograph. The Fremd High School English Ning. By Russ Anderson. 2012. Web. 2 Dec. 2012. <http://api.ning.com/files/f\*OHYOdIjBy3xykmHY7UNkvqxVh1jEAHUnroneJ\*3byEq\*qI0twXSTIW2YdBFMN3dfvU2yNbmBe7G1uz21yDuXlYsypOosKX/pond\_hockey.jpg>. <br /> <br />

<b>Code Citing:</b><br />

• Welling, Luke, and Laura Thomson. PHP and MySQL Web Development, Fourth Edition. Upper Saddle River [etc.: Addison-Wesley, 2009. Print. <br /><br />

• \"Storing Objects in Php Session.\" Stackoverflow. N.p., 18 Sept. 2009. Web. 5 Dec. 2012. <http://stackoverflow.com/questions/1442177/storing-objects-in-php-session>.<br /><br />

• \"PHP Redirect To Another URL Page Script Redirect Redirect Web Page.\" Frequently Asked Questions About Linux UNIX RSS. N.p., n.d. Web. 29 Nov. 2012. <http://www.cyberciti.biz/faq/php-redirect/>. <br /><br />

• \"PHP Manual.\" PHP:. N.p., Nov.-Dec. 2012. Web. Nov.-Dec. 2012. <http://www.php.net/manual/en/>. <br /><br />

";

$page->Display();

?>

*cart.php*

<?

session\_start();

if(!isset($\_SESSION['valid\_user']))

{

header("Location: home.php");

exit;

}

require\_once("page.php");

require\_once("itemFunc.php");

require\_once("item.php");

require\_once("cartForm.php");

class CartPage extends Page

{

public $newitem;

public $newspec;

public function Display()

{

$this->title= "Hockey Heaven: Your shopping cart";

//<head> tag

echo "<html>\n<head>\n";

$this->DisplayTitle();

$this->DisplayKeywords();

$this->DisplayStyles();

echo "</head>\n<body>\n";

//<body> tag

$this->DisplayHeader();

$this->DisplayCart();

$this->DisplaySearch();

$this->DisplayMenu($this->buttons);

if(isAdmin())

{

$this->DisplayMenu($this->adminbuttons);

}

if(isShipping())

{

$this->DisplayMenu($this->shippingbuttons);

}

if(isFinance())

{

$this->DisplayMenu($this->financebuttons);

}

$this->UpdateCartItems();

$this->DisplayFooter();

echo "</body>\n</html>\n";

}

public function UpdateCartItems()

{

if($this->newitem)

{

if(!isset($\_SESSION['cart']))

{

$\_SESSION['cart']= array();

$\_SESSION['items']= 0;

$\_SESSION['totalPrice']= 0.0;

}

if(isset($\_SESSION['cart'][$this->newitem."".$this->newspec]))

{

$another= new Item();

$another= unserialize($\_SESSION['cart'][$this->newitem."".$this->newspec]);

$another->qty++;

$\_SESSION['cart'][$this->newitem."".$this->newspec]= serialize($another);

unset($another);

}

else

{

$theitem= new Item();

$theitem->itemid= $this->newitem;

$theitem->specid= $this->newspec;

$theitem->qty= 1;

$\_SESSION['cart'][$this->newitem."".$this->newspec]= serialize($theitem);

unset($theitem);

/\*

$\_SESSION['cart'][$this->newitem."".$this->newspec]= new Item();

$\_SESSION['cart'][$this->newitem."".$this->newspec]->itemid=$this->newitem;

$\_SESSION['cart'][$this->newitem."".$this->newspec]->specid=$this->newspec;

$\_SESSION['cart'][$this->newitem."".$this->newspec]->qty= 1;

echo $this->newitem."".$this->newspec;

\*/

}

$\_SESSION['totalPrice']= calculatePrice($\_SESSION['cart']);

$\_SESSION['items']= calculateItems($\_SESSION['cart']);

}

if(isset($\_POST['save']) && isset($\_SESSION['cart']))

{

$item= new Item();

foreach($\_SESSION['cart'] as $itemid=>$value)

{

if($\_POST[$itemid] == '0')

{

unset($\_SESSION['cart'][$itemid]);

}

else

{

$item= unserialize($\_SESSION['cart'][$itemid]);

$item->qty= $\_POST[$itemid];

$\_SESSION['cart'][$itemid]=serialize($item);

}

}

$\_SESSION['totalPrice']= calculatePrice($\_SESSION['cart']);

$\_SESSION['items']= calculateItems($\_SESSION['cart']);

}

if(($\_SESSION['cart']) )

{

$this->content=DisplayCartItems($\_SESSION['cart'], $\_SESSION['totalPrice'], $\_SESSION['items']);

echo $this->content;

}

else

{

echo "<p>There are no items in your cart </p>";

}

?>

<br />

<table>

<tr>

<td>

<form action="home.php">

<input type="submit" value="Continue Shopping" />

</form>

</td>

<td>

<form action="checkout.php">

<input type="submit" value="Checkout" />

</form>

</td>

</tr>

</table>

<?

}

}

$new= $\_GET['new'];

$spec= $\_GET['extras'];

$page= new CartPage();

$page->newitem= $new;

$page->newspec= $spec;

$page->Display();

?>

*cartForm.php*

<?php

require\_once('item.php');

require\_once('customerFunc.php');

require\_once('itemFunc.php');

function DisplayCartItems($cart, $totalPrice, $totalItems, $update= true)

{

$cartForm.="";

$cartForm.= "<table border=\"0\" width=\"100%\" cellspacing=\"0\">

<form action=\"cart.php\" method=\"post\">

<tr><th colspan=\"2\" bgcolor=\"#cccccc\">Item</th>

<th bgcolor=\"#cccccc\">Price</th>

<th bgcolor=\"#cccccc\">Quantity</th>

<th bgcolor=\"#cccccc\">Total</th>

</tr>";

$theItem= new Item();

//display each item as a table row

foreach($cart as $item)

{

$theItem= unserialize($item);

//get the item details

$itemdet= getItemDetails($theItem->itemid);

$itemSpec=getSpecItemDetails($theItem->specid, $theItem->itemid);

$cartForm.="<tr>";

$cartForm.="<td align=\"left\">";

if(file\_exists("Images/".$itemdet['name'].".jpg"))

{

$size= GetImageSize("Images/".$itemdet['name'].".jpg");

if(($size[0] >0) && ($size[1] > 0))

{

$cartForm.="<img src=\"Images/".$itemdet['name'].".jpg\"

style=\"border: 1px solid black\"

width=\"".($size[0]/3)."\"

height=\"".($size[1]/3)."\" />";

}

}

$cartForm.="</td>";

//display the items name price and quantity being bought

$cartForm.="<td align=\"left\">

<a href=\"itemPage.php?itemid=".$theItem->itemid."\"> ".$itemdet['name']." </a><br />".$itemSpec."</td>

<td align=\"center\">\$".number\_format($itemdet['price'], 2)."</td>

<td align=\"center\">";

//allow changes in text boxes

$cartForm.="<input type=\"text\" name=\"".$theItem->itemid."".$theItem->specid."\" value=\"".$theItem->qty."\" size=\"3\">

</td>

<td align=\"center\">\$".number\_format($itemdet['price']\*$theItem->qty,2)."</td>

</tr>\n";

}

//diplay the total row

$cartForm.="<tr>

<th colspan=\"3\" bgcolor=\"#cccccc\">&nbsp;</td>

<th align=\"center\" bgcolor=\"#cccccc\">".$totalItems."</th>

<th align=\"center\" bgcolor=\"#cccccc\">

\$".number\_format($totalPrice,2)."</th>

</tr>";

if($update)

{

$cartForm.="<tr>

<td colspan=\"3\">&nbsp;</td>

<td align=\"center\">

<input type=\"hidden\" name=\"save\" value=\"true\" />

<input type= \"submit\" value=\"Save Changes\" />

</td>

<td>&nbsp;</td>

</tr>";

}

$cartForm.="</form></table>";

return $cartForm;

}

function DisplayCheckoutForm()

{

$customer= customerDetails($\_SESSION['valid\_user']);

$checkoutForm="";

$checkoutForm.= "<br />

<table border=\"0\" width=\"100%\" cellspacing=\"0\">

<form action=\"purchase.php\" method=\"post\">

<tr><th colspan=\"2\" bgcolor=\"#cccccc\">Shipping Address</th></tr>

<tr>

<td>Name</td>

<td><input type=\"text\" name=\"name\" value=\"".$customer['firstName']." ".$customer['lastName']."\" maxlength=\"40\" size=\"40\"/></td>

</tr>

<tr>

<td>Address</td>

<td><input type=\"text\" name=\"address\" value=\"".$customer['street']."\" maxlength=\"40\" size=\"40\"/></td>

</tr>

<tr>

<td>City</td>

<td><input type=\"text\" name=\"city\" value=\"".$customer['city']."\" maxlength=\"20\" size=\"40\"/></td>

</tr>

<tr>

<td>State</td>

<td><input type=\"text\" name=\"state\" value=\"".$customer['st']."\" maxlength=\"20\" size=\"40\"/></td>

</tr>

<tr>

<td>Zip Code</td>

<td><input type=\"text\" name=\"zip\" value=\"".$customer['zip']."\" maxlength=\"10\" size=\"40\"/></td>

</tr>

<tr>

<td>

</td>

<td>

<input type=\"submit\" value=\"Submit Order\">

</td>

</tr>

</form>

</table> <br />";

return $checkoutForm;

}

function DisplayShipping($shipping)

{

$form="<table border=\"0\" width=\"100\%\" cellspacing= \"0\">

<tr><th align=\"left\">Shipping</td>

<td align=\"right\">".number\_format($shipping,2)."</td></tr>

<tr><th bgcolor=\"#cccccc\" align=\"left\"> Total </th>

<th bgcolor=\"#cccccc\" align=\"right\">\$".number\_format($shipping+$\_SESSION['calculatePrice'],2)."</th>

</tr>

</table> <br />";

}

function DisplayCreditCardForm($name, $id= '')

{

$form="<table border=\"0\" width=\"100\%\" cellspacing=\"0\">

<form action=\"process.php\" method=\"post\">

<tr><th colspan=\"5\" bgcolor=\"#cccccc\">Credit Card Details</th></tr>

<tr>

<td>Type</td>

<td>Card Number</td>

<td>Security Code</td>

<td>Name on card</td>

<td colspan=\"2\">Expiration Date<br />

Month/Year</td>

</tr>

<tr>

<td><select name=\"cardType\">

<option value=\"VISA\">VISA</option>

<option value=\"MasterCard\">MasterCard</option>

<option value=\"American Express\">American Express</option>

</select>

</td>

<td><input type=\"text\" name=\"cardNum\" value=\"\" maxlength=\"16\" size=\"40\"></td>

<td><input type=\"text\" name=\"seccode\" value=\"\" maxlength=\"4\" size=\"4\"></td>

<td><input type=\"text\" name=\"cardName\" value = \".$name.\" maxlength=\"40\" size=\"40\"></td>

<td>

<select name=\"month\">

<option value=\"01\">01</option>

<option value=\"02\">02</option>

<option value=\"03\">03</option>

<option value=\"04\">04</option>

<option value=\"05\">05</option>

<option value=\"06\">06</option>

<option value=\"07\">07</option>

<option value=\"08\">08</option>

<option value=\"09\">09</option>

<option value=\"10\">10</option>

<option value=\"11\">11</option>

<option value=\"12\">12</option>

</select>

<select name=\"year\">";

for ($y = date("Y"); $y < date("Y") + 10; $y++)

{

$form.="<option value=\"".$y."\">".$y."</option>";

}

$form.="</select>

</td>

</tr>

<td colspan=\"6\" align=\"center\">

<p><strong>Please press Purchase to confirm your purchase, or Continue Shopping to

add or remove items</strong></p>

<input type=\"hidden\" name=\"id\" value=".$id." >

<input type=\"submit\" value=\"Confirm Purchase\" >

</td>

</tr>

</table>";

return $form;

}

?>

*changePW.php*

<?php

session\_start();

require\_once('authFunc.php');

require\_once('dataValid.php');

require\_once('page.php');

$page= new Page();

if(!filledOut($\_POST))

{

$page->content="You have not filled out the form correctly. <br />

Please try again. <br />".changePWForm()."";

}

else

{

$newPW= $\_GET['newPW'];

$newPW2= $\_GET['newPW2'];

$oldPW= $\_GET['oldPW'];

$user= $\_SESSION['valid\_user'];

$type= $\_SESSION['AccountType'];

if($newPW != $newPW2)

{

$page->content="Passwords entered are not the same. Password has not been changed. <br />

".changePWForm()."";

}

else if((strlen($newPW)>16) || (strlen($newPW)<6))

{

$page->content="New password must be between 6 and 16 characters. Try again. <br />

".changePWForm()."";

}

else

{

if(changePW($user, $oldPW, $newPW, $type))

{

$page->content="Password changed.";

}

else

{

$page->content="Password could not be changed. Please try again later.";

}

}

}

$page->Display();

?>

*changePWPage.php*

<?php

session\_start();

require\_once('authFunc.php');

require\_once('page.php');

$page= new Page();

$page->content="<h2>Change Password</h2> <br />

".changePWForm()."";

$page->Display();

?>

*checkout.php*

<?php

session\_start();

require('dbFunc.php');

require\_once('cartForm.php');

require\_once('item.php');

require('page.php');

$page= new page();

if($\_SESSION['cart'] && (array\_count\_values($\_SESSION['cart'])))

{

$page->content="";

$page->content= DisplayCartItems($\_SESSION['cart'], $\_SESSION['totalPrice'], $\_SESSION['items'], false);

$page->content.=DisplayCheckoutForm();

}

else

{

$page->content="Your shopping cart is empty.";

}

$page->Display();

?>

*contact.php*

<?php

session\_start();

require("page.php");

$contact= new Page();

$contact->content= "<form name=\"myWebForm\"

action=\"mailto:customer\_service@hockeyheaven.com\"

method=\"post\">

<table>

<tr>

<td class=\"graybg\">

Username:

</td>

<td class= \"graybg\">

<input name=\"user\" value=".$\_SESSION['valid\_user'].">

</td>

</tr>

<tr>

<td class=\"graybg\">

Subject:

</td>

<td class=\"graybg\">

<input name=\"subject\">

</td>

</tr>

</table>

Comments<br />

<textarea rows=\"10\" cols=\"75\"

name=\"comments\" wrap=\"physical\"></textarea><br/>

<input type=\"submit\" value=\"Send\"/>

</form>";

$contact->Display();

?>

*customerFunc.php*

<?php

require\_once('dbFunc.php');

function getReviews($customerid)

{

$db= dbConnect();

$query="select \* from reviews

where customerid=".$customerid."";

$result=$db->query($query);

if((!$result) || ($result->num\_rows < 1))

{

return "";

}

$reviews= "<h2>Reviews</h2>

<table>

<tr>

<th>Item</th>

<th>Score</th>

<th>Review</th>

</tr>";

while($row= $result->fetch\_assoc())

{

$query2="select name from merchandise

where itemid=".$row['itemid']."";

$result2= $db->query($query2);

$row2= $result2->fetch\_assoc();

$reviews.="<tr>

<td>

<a href=\"itemPage.php?item=".$row['itemid']."\"> ".$row2['name']."</a>

</td>

<td>".$row['score']."/5 </td>

<td>

<a href=\"reviewPage.php?customerid=".$row['customerid']."&amp&amp itemid=".$row['itemid']."\">View Review</a>

</td>

</tr>";

}

$reviews.="</table>";

return $reviews;

}

function getOrders($name)

{

if(!isset($name))

{

return "You are not logged in. Please login to view this page.";

}

$db= dbConnect();

$query= "select customerid from customer

where username=".$name."";

$result=$db->query($query);

if(!isset($result) || $result->num\_rows == 0)

{

return "Unable to access database at this time. Please try again later.";

}

$row= $result->fetch\_assoc();

$query="select \* from orders

where customerid=".$row['customerid']."";

$result=$db->query($query);

if((!$result) || ($result->num\_rows < 1))

{

return "No current orders found";

}

$orders="<h2>Orders</h2>";

while($row= $result->fetch\_assoc())

{

//setup a table for the order

$orders.="<table>

<tr>

<td>Order made: ".$row['orderdate']."</td>

<td>Status: ".$row['orderstatus']."</td>

</tr>

<tr>

<th>Name</th> <br/>

<th>Specs</th>

<th>Quantity</th>

<th>Cost</th>

</tr>";

$query2="select \* from orderitems

where orderid=".$row['orderid']."";

$result2= $db->query($query2);

while($row2= $result2->fetch\_assoc())

{

$query3="select price from merchandise

where itemid=".$row2['itemid']."";

$result3= $db->query($query3);

$price= $result3->fetch\_assoc();

$orders.="<tr>

<td>

<a href=\"itemPage.php?item=".$row2['itemid']."\"> ".$row2['name']."</a>

</td>

<td>

Function to regulate specs here

</td>

<td>

".$row2['quantity']."

</td>

<td>

".number\_format($price,2)."

</td>

</tr>";

}

$orders.="<tr>

<th colspan=\"2\">Total</th>

<th>".$row['totalcost']."</th>

<th>

</tr>

</table> <br /> <br />";

}

return $orders;

}

function customerDetails($name)

{

$db= dbConnect();

$query= "select \* from customer

where username=\"".$name."\"";

$result=$db->query($query);

$customer=$result->fetch\_assoc();

return $customer;

}

?>

*dataValid.php*

<?php

function filledOut($formVars)

{

//test that each variable has a value

foreach($formVars as $key => $value)

{

if((!isset($key)) || ($value ==''))

{

return false;

}

}

return true;

}

function validEmail($address)

{

if(ereg("^[a-zA-Z0-9\_\.\-]+@[a-zA-Z0-9\_\.\-]+\.[a-zA-Z0-9\_\.\-]+$", $address))

{

return true;

}

else

{

return false;

}

}

?>

*dbFunc.php*

<?php

require\_once('item.php');

function dbConnect()

{

@ $result= new mysqli('localhost', 'mindac\_mindac' , 'saints19' ,'mindac\_HH');

if(mysqli\_connect\_errno())

{

echo "Error: Could not connect to database. Please try again later.";

exit;

}

$result->autocommit(TRUE);

return $result;

}

function dbResultArray($result)

{

$resultarray= array();

for($count=0; $row= $result->fetch\_assoc(); $count++)

{

$resultarray[$count]= $row;

}

return $resultarray;

}

?>

*dblogin.php*

<?

session\_start();

require\_once("page.php");

require\_once("authFunc.php");

$page= new Page();

$username= $\_POST["username"];

$pwd= $\_POST["pwd"];

$accountType= $\_POST["accountType"];

$\_SESSION['valid\_user'];

$\_SESSION['AccountType'];

if(!$username || !$pwd)

{

$page->content= loginform();

}

if(login($username, $pwd, $accountType))

{

$\_SESSION['valid\_user']= $username;

$\_SESSION['AccountType']= $accountType;

$page->content= "Login Successful! Welcome, ".$username.".";

}

else

{

//if not match is found tell user they were not logged in

$page->content= "Invalid username/password combination. Please try again.".loginform()."

<br />".$username." not found.";

}

$page->Display();

?>

*deleteCustomer.php*

<?php

session\_start();

require\_once('dbFunc.php');

require\_once('page.php');

require\_once('adminFunc.php');

$page= new Page();

if(DeleteUser($\_POST['customerid'], 'customer'))

{

$page->content= "Customer removed from database";

}

else

{

$page->content= "Unable to remove customer from database";

}

$page->Display();

?>

*deleteEmployee.php*

<?php

session\_start();

require\_once('dbFunc.php');

require\_once('page.php');

require\_once('adminFunc.php');

$page= new Page();

echo $\_POST['employeeid'];

if(DeleteUser($\_POST['employeeid'], 'employee'))

{

$page->content= "Employee removed from database";

}

else

{

$page->content= "Unable to remove employee from database";

}

$page->Display();

?>

*deleteItem.php*

<?php

session\_start();

require\_once('adminFunc.php');

require\_once('page.php');

$page= new Page();

if(DeleteItem($\_POST['itemid']))

{

$page->content= "Item deleted.";

}

else

{

$page->content.="Item not deleted. Please try again.";

}

$page->Display();

?>

*editCustomer.php*

<?php

session\_start();

require('page.php');

require('dataValid.php');

require('adminFunc.php');

$page= new Page();

$username= $\_POST['username'];

$street= $\_POST['street'];

$city= $\_POST['city'];

$state= $\_POST['state'];

$zip= $\_POST['zip'];

$email= $\_POST['email'];

if(filledOut($\_POST))

{

echo $\_POST['customerid'];

if(UpdateCustomer($\_POST['customerid'], $username, $street, $city, $state, $zip, $email))

{

$page->content= "User:<em> ".stripslashes($username)."</em> updated.";

}

else

{

$page->content= "User:<em> ".stripslashes($username)."</em> could not be updated. Please try again later.";

}

}

else

{

$page->content= "The form was not filled out correctly. Please try again.";

}

$page->Display();

?>

*editEmployee.php*

<?php

session\_start();

require('page.php');

require('dataValid.php');

require('adminFunc.php');

$page= new Page();

$username= $\_POST['username'];

$pw= $\_POST['pw'];

$ssn= $\_POST['ssn'];

$type= $\_POST['category'];

$firstname= $\_POST['firstName'];

$lastname= $\_POST['lastName'];

$street= $\_POST['street'];

$city= $\_POST['city'];

$state= $\_POST['state'];

$zip= $\_POST['zip'];

$email= $\_POST['email'];

$phone= $\_POST['phone'];

$paymentamount= $\_POST['paymentamount'];

if(filledOut($\_POST))

{

echo $\_POST['employeeid'];

if(UpdateEmployee($\_POST['employeeid'],$street, $city, $state, $zip, $email, $phone, $paymentamount))

{

$page->content= "User:<em> ".stripslashes($username)."</em> updated in database.";

}

else

{

$page->content= "User:<em> ".stripslashes($username)."</em> could not be updated.";

}

}

else

{

$page->content= "The form was not filled out correctly. Please try again.";

}

$page->Display();

?>

*editMerch.php*

<?php

session\_start();

require\_once('adminFunc.php');

require\_once('page.php');

$page= new Page();

if(UpdateItem($\_POST['olditemid'], $\_POST['itemid'], $\_POST['name'], $\_POST['manufacturer'], $\_POST['catid'], $\_POST['price'], $\_POST['description']))

{

$page->content= "Item Updated.";

}

else

{

$page->content.="Item not updated. Please try again.";

}

$page->Display();

?>

*finance.php*

<?

session\_start();

require\_once("page.php");

require\_once("authFunc.php");

require\_once('item.php');

$page= new Page();

if(isFinance())

{

$page->content= "Welcome to the Finance homepage";

}

else

{

$page->content= "You are not authorized to enter the finance area.";

}

$page->Display();

?>

*financeEmployee.php*

<?php

session\_start();

require\_once('adminFunc.php');

require\_once('financeFunc.php');

require\_once('dbFunc.php');

require\_once('page.php');

$id= $\_GET['id'];

$page= new Page();

$emp= getEmployee($id);

$page->content=updateSalaryForm($emp);

$page->Display();

?>

*financeFunc.php*

<?php

require\_once('dbFunc.php');

function addToItemQuantity($itemid, $specid, $qty)

{

$db= dbConnect();

$query= "select catid

from merchandise

where itemid=".$itemid."";

$result= $db->query($query);

if(!$result)

{

return false;

}

$row= $result->fetch\_assoc();

$cat= getCategoryName($row['catid']);

$query= "update ".$cat."

set quantity= quantity+".$qty."

where itemid=".$itemid." and exId=\"".$specid."\"";

$result= $db->query($query);

if(!$result)

{

return false;

}

return true;

}

function updateSalaryForm($emp)

{

$form= "<table><form action=\"updateSalary.php\" method=\"post\">

<tr><th>Employee:</th><td>".$emp['firstName']." ".$emp['lastName']."</td></tr>

<tr><th>Payment Amount: <br />

(".$emp['paymenttype'].")</th><td><input type=\"text\" name=\"paymentamount\" value=".$emp['paymentamount']."></td></tr>

<input type=\"hidden\" name=\"empid\" value=".$emp['employeeid'].">

<tr><th></th><td><input type=\"submit\" value=\"Change Pay\"></td></tr>

</form></table>";

return $form;

}

function updateSalary($amount,$id)

{

$db= dbConnect();

$query= "update employees set

paymentamount=".$amount."

where employeeid=".$id."";

$result= $db->query($query);

if(!$result)

{

return false;

}

return true;

}

function getMerchandise()

{

$db= dbConnect();

$query= "select \* from merchandise";

$result= $db->query($query);

if(!$result)

{

return false;

}

$items= "";

while($item= $result->fetch\_assoc())

{

$items.="<table width=\"30\%\">

<tr><th>Item ".$item['itemid'].": ".$item['name']."</th>

<td><form action=\"financeitem.php\"> <input type=\"hidden\" name=\"item\" value=".$item['itemid']."><input type=\"submit\" value=\"Edit Item\"></form></td></tr>

<tr><th>Price:</th><td>".$item['price']."</td></tr>

<tr><th>Manufacturer:</th><td>".$item['manufacturer']."</td></tr>

</table> <br />";

}

return $items;

}

?>

*financeitem.php*

<?php

session\_start();

require\_once('dbFunc.php');

require\_once('page.php');

require\_once('itemFunc.php');

$page= new Page();

if(!$\_GET['item'])

{

$page->content= "Item unspecified. Unable to reveal stock sheet.";

}

else

{

$item= getItemDetails($\_GET['item']);

$page->content= "<h4>Update stock for ".$item['name']."</h4>

<form action=\"addstock.php\">".CreateExtrasForm($\_GET['item'])."

<input type=\"text\" name=\"qty\">

<input type=\"hidden\" name=\"itemid\" value=".$\_GET['item'].">

<input type=\"submit\" value=\"Update Stock\">

</form>";

}

$page->Display();

?>

*gloves.php*

<?

session\_start();

if(!isset($\_SESSION['valid\_user']))

{

header("Location: home.php");

exit;

}

require\_once("dbFunc.php");

require\_once('item.php');

require\_once("page.php");

class GlovePage extends Page

{

public function Display()

{

//<head> tag

echo "<html>\n<head>\n";

$this->DisplayTitle();

$this->DisplayKeywords();

$this->DisplayStyles();

echo "</head>\n<body>\n";

//<body> tag

$this->DisplayHeader();

$this->DisplayCart();

$this->DisplaySearch();

$this->DisplayMenu($this->buttons);

if(isAdmin())

{

$this->DisplayMenu($this->adminbuttons);

}

if(isShipping())

{

$this->DisplayMenu($this->shippingbuttons);

}

if(isFinance())

{

$this->DisplayMenu($this->financebuttons);

}

$this->DisplayGloves();

$this->DisplayFooter();

echo "</body>\n</html>\n";

}

public function DisplayGloves()

{

$db= dbConnect();

//find the catid of gloves

$query= "select catid from categories

where catname= \"gloves\"";

$result= $db->query($query);

if($result->num\_rows > 0)

{

$rowid= $result->fetch\_assoc();

$id= $rowid['catid'];

//find all merchandise with the gloves catid

$query="select \* from merchandise

where catid=".$id."";

$result= $db->query($query);

}

$num\_results= $result->num\_rows;

echo "<p>Number of gloves retrieved: ".$num\_results."</p>";

for($i= 0; $i < $num\_results; $i++)

{

$row= $result->fetch\_assoc();

echo "<p><b>".($i+1)." "." Name: ";

echo "<a style=\"color:#800000\" href=\"itemPage.php?itemid=".$row['itemid']."\" >".$row['name']."</a>";

echo "</b><br />Manufacturer: ";

echo $row['manufacturer'];

echo "<br />ItemID: ";

echo $row['itemid'];

echo "<br /> Price: ";

echo $row['price'];

echo "</p>";

}

$result->free();

$db->close();

}

}

$page= new GlovePage();

$page->Display();

?>

*helmets.php*

<?

session\_start();

if(!isset($\_SESSION['valid\_user']))

{

header("Location: home.php");

exit;

}

require\_once("page.php");

require\_once('item.php');

require\_once("dbFunc.php");

class HelmetPage extends Page

{

public function Display()

{

//<head> tag

echo "<html>\n<head>\n";

$this->DisplayTitle();

$this->DisplayKeywords();

$this->DisplayStyles();

echo "</head>\n<body>\n";

//<body> tag

$this->DisplayHeader();

$this->DisplayCart();

$this->DisplaySearch();

$this->DisplayMenu($this->buttons);

if(isAdmin())

{

$this->DisplayMenu($this->adminbuttons);

}

if(isShipping())

{

$this->DisplayMenu($this->shippingbuttons);

}

if(isFinance())

{

$this->DisplayMenu($this->financebuttons);

}

$this->DisplayHelmets();

$this->DisplayFooter();

echo "</body>\n</html>\n";

}

public function DisplayHelmets()

{

$db= dbConnect();

//find the catid of gloves

$query= "select catid from categories

where catname= \"helmets\"";

$result= $db->query($query);

if($result->num\_rows > 0)

{

$rowid= $result->fetch\_assoc();

$id= $rowid['catid'];

//find all merchandise with the gloves catid

$query="select \* from merchandise

where catid=".$id."";

$result= $db->query($query);

}

$num\_results= $result->num\_rows;

echo "<p>Number of helmets retrieved: ".$num\_results."</p>";

for($i= 0; $i < $num\_results; $i++)

{

$row= $result->fetch\_assoc();

echo "<p><b>".($i+1)." "." Name: ";

echo "<a style=\"color:#800000\" href=\"itemPage.php?itemid=".$row['itemid']."\" >".$row['name']."</a>";

echo "</b><br />Manufacturer: ";

echo $row['manufacturer'];

echo "<br />ItemID: ";

echo $row['itemid'];

echo "<br /> Price: ";

echo $row['price'];

echo "</p>";

}

$result->free();

$db->close();

}

}

$page= new HelmetPage();

$page->Display();

?>

*home.php*

<?php

session\_start();

require\_once("page.php");

require\_once("itemFunc.php");

$homepage= new Page();

$homepage->content= "<p>Welcome to Hockey Heaven.</p>

<br/>";

$homepage->content.="<h4>Check out some of these items</h4>".GetRandomItems()."";

$homepage->Display();

?>

*insertCat.php*

<?php

session\_start();

require\_once('adminFunc.php');

require\_once('page.php');

$page= new Page();

if(AddCategory($\_POST['catname']))

{

$page->content= "Item inserted into Categories.";

}

else

{

$page->content.="Category not inserted. Please try again.";

}

$page->Display();

?>

*insertCustomer.php*

<?php

session\_start();

require('page.php');

require('dataValid.php');

require('adminFunc.php');

$page= new Page();

$username= $\_POST['username'];

$pw= $\_POST['pw'];

$firstname= $\_POST['firstName'];

$lastname= $\_POST['lastName'];

$street= $\_POST['street'];

$city= $\_POST['city'];

$state= $\_POST['state'];

$zip= $\_POST['zip'];

$email= $\_POST['email'];

if(filledOut($\_POST))

{

if(AddCustomer($username, $pw, $firstname, $lastname, $street, $city, $state, $zip, $email))

{

$page->content= "User:<em> ".stripslashes($username)."</em> added to database.";

}

else

{

$page->content= "User:<em> ".stripslashes($username)."</em> could not be added to the database.";

}

}

else

{

$page->content= "The form was not filled out correctly. Please try again.";

}

$page->Display();

?>

*insertEmployee.php*

<?php

session\_start();

require('page.php');

require('dataValid.php');

require('adminFunc.php');

$page= new Page();

$username= $\_POST['username'];

$pw= $\_POST['pw'];

$ssn= $\_POST['ssn'];

$type= $\_POST['category'];

$firstname= $\_POST['firstName'];

$lastname= $\_POST['lastName'];

$street= $\_POST['street'];

$city= $\_POST['city'];

$state= $\_POST['state'];

$zip= $\_POST['zip'];

$email= $\_POST['email'];

$phone= $\_POST['phone'];

$paymentamount= $\_POST['paymentamount'];

if(filledOut($\_POST))

{

if(AddEmployee($username, $pw, $ssn, $type, $firstname, $lastname, $street, $city, $state, $zip, $email, $phone, $paymentamount))

{

$page->content= "User:<em> ".stripslashes($username)."</em> added to database.";

}

else

{

$page->content= "User:<em> ".stripslashes($username)."</em> could not be added to the database.";

}

}

else

{

$page->content= "The form was not filled out correctly. Please try again.";

}

$page->Display();

?>

*insertMerch.php*

<?php

session\_start();

require\_once('adminFunc.php');

require\_once('page.php');

$page= new Page();

if(AddItem($\_POST['itemid'], $\_POST['name'], $\_POST['manufacturer'], $\_POST['catid'], $\_POST['price'], $\_POST['description']))

{

$page->content= "Item inserted into Merchandise.";

}

else

{

$page->content.="Item not inserted. Please try again.";

}

$page->Display();

?>

*insertReview.php*

<?php

session\_start();

require\_once('customerFunc.php');

require\_once('reviewFunc.php');

require\_once('page.php');

$page= new Page();

$customer= customerDetails($\_SESSION['valid\_user']);

if(submitReview($\_POST['itemid'], $customer['customerid'], $\_POST['score'], $\_POST['blurb']))

{

$page->content= "Review submitted.";

}

else

{

$page->content="Unable to submit review. Please try again later.";

}

$page->Display();

?>

*insertSpec.php*

<?php

session\_start();

require\_once('adminFunc.php');

require\_once('page.php');

$page= new Page();

if(isset($\_GET))

{

echo $\_GET['catname'];

if(insertSpec($\_GET))

{

$page->content= "Item specialization created.";

}

else

{

$page->content= "Unable to create item specialization.";

}

}

$page->Display();

?>

*item.php*

<?php

class Item

{

public $itemid;

public $specid;

public $qty;

public function \_\_set($name, $value)

{

$this->$name= $value;

}

}

?>

*itemFunc.php*

<?php

require\_once('dbFunc.php');

require\_once('authFunc.php');

require\_once('item.php');

function calculateShipping()

{

return 5.00;

}

function getCategories()

{

$db= dbConnect();

$query= "select catid, catname from categories";

$result= $db->query($query);

if(!$result)

{

return false;

}

$num\_cats= $result->num\_rows;

if($num\_cats ==0)

{

return false;

}

return $result;

}

function getCategoryName($catid)

{

$db= dbConnect();

$query= "select catname from categories

where catid= ".$catid."";

$result= $db->query($query);

if(!$result)

{

return false;

}

$num\_cats= $result->num\_rows;

if($num\_cats == 0)

{

return false;

}

$row= $result->fetch\_object();

return $row->catname;

}

function getItems($catid)

{

if((!$catid) || ($catid == ''))

{

return false;

}

$db= dbConnect();

$query= "select \* from merchandise

where catid= ".$catid."";

$result= $db->query($query);

if(!$result)

{

return false;

}

$numitems= $result->num\_rows;

if($numitems == 0)

{

return false;

}

$result= db\_result\_to\_array($result);

return $result;

}

function getItemDetails($itemid)

{

if((!$itemid) || ($itemid ==''))

{

return false;

}

$db= dbConnect();

$query= "select \* from merchandise where itemid=".$itemid."";

$result= $db->query($query);

if(!$result)

{

return false;

}

$item= $result->fetch\_assoc();

return $item;

}

function getSpecItemDetails($specid, $itemid)

{

if((!$itemid) || ($itemid =='') || (!$specid) || ($specid==''))

{

return false;

}

$db= dbConnect();

$query= "select catid from merchandise where itemid=".$itemid."";

$result= $db->query($query);

if(!$result)

{

return false;

}

$item= $result->fetch\_assoc();

$cat= getCategoryName($item['catid']);

$query= "select \* from ".strtolower($cat)."

where itemid=".$itemid." and exId=\"".$specid."\"";

$result= $db->query($query);

if(!$result || $result->num\_rows == 0)

{

return false;

}

$row= $result->fetch\_assoc();

$special="";

foreach($row as $key=>$value)

{

if($key != 'exId' && $key != 'quantity' && $key != 'itemid')

$special.= $value."/";

}

return $special;

}

function calculatePrice($cart)

{

$price= 0.0;

if(is\_array($cart))

{

$db= dbConnect();

foreach($cart as $itemid =>$qty)

{

$item= new Item();

$item= unserialize($cart[$itemid]);

$query= "select price from merchandise where itemid=".$item->itemid."";

$result= $db->query($query);

if($result)

{

$items= $result->fetch\_object();

$itemprice= $items->price;

$price+=$itemprice\*$item->qty;

}

}

}

return $price;

}

function calculateItems($cart)

{

$items= 0;

if(is\_array($cart))

{

foreach($cart as $itemid => $qty)

{

$item= new Item();

$item= unserialize($cart[$itemid]);

$items+= $item->qty;

}

}

return $items;

}

function GetRandomItems()

{

$db= dbConnect();

$query= "select \* from merchandise";

$result= $db->query($query);

if(!$result)

{

return "Cannot access database. Please try again later.";

}

$res= array();

$res= dbResultArray($result);

shuffle($res);

$items="<table>

<tr>";

for($count=0; $count < 3; $count++)

{

$items.= "<td><a style=\"color:#800000\" href=\"itemPage.php?itemid=".$res[$count]['itemid']."\" ><img src= \"Images/".$res[$count]['name'].".jpg\" width=\"180\" height=\"180\"/><br />"

.$res[$count]['name']."</a>

<br /> Price: \$".$res[$count]['price']."</td>";

}

$items.="</table>

</tr>";

return $items;

}

//finds the average review score for an item

function getReviewScore($id)

{

$db= dbConnect();

$query= "select avg(score) as avgScore from reviews

where itemid=".$id."";

$result= $db->query($query);

if(!isset($result)|| $result->num\_rows == 0)

{

return "No reviews have been written for this item.";

}

else

{

$row=$result->fetch\_assoc();

return "".$row['avgScore']."/5";

}

}

function updateItemStock($itemid, $specid, $inc)

{

$db= dbConnect();

$item= getItemDetails($itemid);

$cat= getCategoryName($item['catid']);

$query="update ".strtolower($cat)."

set quantity=quantity+".$inc."

where itemid=".$itemid." and exId=\"".$specid."\"";

$result=$db->query($query);

if(!$result)

{

return false;

}

return true;

}

function CreateExtrasForm($id)

{

//need to insert results of query

$row= getItemDetails($id);

$catname= getCategoryName($row['catid']);

$db= dbConnect();

$query= "select \* from ".strtolower($catname)."

where itemid=".$id."";

$result=$db->query($query);

if(!$result)

return false;

$num\_results= $result->num\_rows;

$extrasform= "<select name=\"extras\">";

while($row= $result->fetch\_assoc())

{

if($row['quantity'] > 0 || isFinance() || isAdmin() || isShipping())

{

$extrasform.= "<option value=".$row['exId'].">";

foreach($row as $key=>$value)

{

if($key != 'exId' && $key != 'quantity' && $key != 'itemid' && $value != 'Practice')

$extrasform.= $value." ";

}

$extrasform.="</option>";

}

}

$extrasform.="</select>";

return $extrasform;

}

?>

*itemPage.php*

<?php

session\_start();

if(!isset($\_SESSION['valid\_user']))

{

header("Location: home.php");

exit;

}

require\_once("page.php");

require\_once("dbFunc.php");

require\_once("authFunc.php");

require\_once("item.php");

require\_once("itemFunc.php");

require\_once("reviewFunc.php");

class ItemPage extends Page

{

public $name;

public $manu;

public $price;

public $desc;

public $extrasform;

public function Display()

{

//<head> tag

$this->CreateExtrasForm($\_GET['itemid']);

echo "<html>\n<head>\n";

$this->DisplayTitle();

$this->DisplayKeywords();

$this->DisplayStyles();

echo "</head>\n<body>\n";

//<body> tag

$this->DisplayHeader();

$this->DisplayCart();

$this->DisplaySearch();

$this->DisplayMenu($this->buttons);

if(isAdmin())

{

$this->DisplayMenu($this->adminbuttons);

}

if(isShipping())

{

$this->DisplayMenu($this->shippingbuttons);

}

if(isFinance())

{

$this->DisplayMenu($this->financebuttons);

}

$this->DisplayItemInfo($\_GET['itemid']);

echo $this->content;

$this->DisplayFooter();

echo "</body>\n</html>\n";

}

public function DisplayItemInfo($id)

{

$row= getItemDetails($id);

$this->name= $row['name'];

$this->manu= $row['manufacturer'];

$this->price= $row['price'];

$this->desc= $row['description'];

$this->content= "<table>

<tr>

<td>

<img src= \"Images/".$this->name.".jpg\" width=\"180\" height=\"180\"/>

</td>

<td>

<b>".$this->name."</b><br/>

Made by:".$this->manu."<br/>

Price:\$".$this->price."<br/>

Review Score: ".getReviewScore($id)."

<form action=\"writeReview.php\">

<input type=\"hidden\" name=\"itemid\" value=\"".$id."\">

<input type=\"submit\" value=\"Review this item\">

</form>

</td>

</tr>

<tr>

<td>

</td>

<td>

<form action=\"cart.php\">

<input type=\"hidden\" name=\"new\" value=".$id.">

".$this->extrasform."

<input type=\"submit\" value=\"Add to Cart\">

</form>";

if(isAdmin())

{

$this->content.="<form action=\"adminitem.php\">

<input type=\"hidden\" name=\"item\" value=".$id.">

<input type=\"submit\" value=\"Edit Item\">

</form>";

}

if(isFinance())

{

$this->content.=" <form action=\"financeitem.php\">

<input type=\"hidden\" name=\"item\" value=".$id.">

<input type=\"submit\" value=\"Edit Item\">

</form>";

$this->content.="<form action=\"specItem.php\">

<input type=\"hidden\" name=\"item\" value=".$id.">

<input type=\"submit\" value=\"Add Specs\">

</form>";

}

$this->content.="</td>

</tr>

</table>

<p>

".$this->desc."

</p>

<br />

".getReviews($id)."";

}

public function CreateExtrasForm($id)

{

//need to insert results of query

$row= getItemDetails($id);

$catname= getCategoryName($row['catid']);

$db= dbConnect();

$query= "select \* from ".strtolower($catname)."

where itemid=".$id."";

$result=$db->query($query);

if(!$result)

return false;

$num\_results= $result->num\_rows;

$this->extrasform= "<select name=\"extras\">";

while($row= $result->fetch\_assoc())

{

if($row['quantity'] > 0)

{

$this->extrasform.= "<option value=".$row['exId'].">";

foreach($row as $key=>$value)

{

if($key != 'exId' && $key != 'quantity' && $key != 'itemid' && $value != 'Practice')

$this->extrasform.= $value." ";

}

$this->extrasform.="</option>";

}

}

$this->extrasform.="</select>";

}

}

$page= new ItemPage();

$page->Display();

?>

*jerseys.php*

<?

session\_start();

if(!isset($\_SESSION['valid\_user']))

{

header("Location: home.php");

exit;

}

require\_once("page.php");

require\_once("dbFunc.php");

require\_once('item.php');

class JerseyPage extends Page

{

public function Display()

{

//<head> tag

echo "<html>\n<head>\n";

$this->DisplayTitle();

$this->DisplayKeywords();

$this->DisplayStyles();

echo "</head>\n<body>\n";

//<body> tag

$this->DisplayHeader();

$this->DisplayCart();

$this->DisplaySearch();

$this->DisplayMenu($this->buttons);

if(isAdmin())

{

$this->DisplayMenu($this->adminbuttons);

}

if(isShipping())

{

$this->DisplayMenu($this->shippingbuttons);

}

if(isFinance())

{

$this->DisplayMenu($this->financebuttons);

}

$this->DisplayJerseys();

$this->DisplayFooter();

echo "</body>\n</html>\n";

}

public function DisplayJerseys()

{

$db= dbConnect();

//find the catid of gloves

$query= "select catid from categories

where catname= \"jerseys\"";

$result= $db->query($query);

if($result->num\_rows > 0)

{

$rowid= $result->fetch\_assoc();

$id= $rowid['catid'];

//find all merchandise with the gloves catid

$query="select \* from merchandise

where catid=".$id."";

$result= $db->query($query);

}

$num\_results= $result->num\_rows;

echo "<p>Number of jerseys retrieved: ".$num\_results."</p>";

for($i= 0; $i < $num\_results; $i++)

{

$row= $result->fetch\_assoc();

echo "<p><b>".($i+1)." "." Name: ";

echo "<a style=\"color:#800000\" href=\"itemPage.php?itemid=".$row['itemid']."\" >".$row['name']."</a>";

echo "</b><br />Manufacturer: ";

echo $row['manufacturer'];

echo "<br />ItemID: ";

echo $row['itemid'];

echo "<br /> Price: ";

echo $row['price'];

echo "</p>";

}

$result->free();

$db->close();

}

}

$page= new JerseyPage();

$page->Display();

?>

*login.php*

<?php

require\_once('page.php');

require\_once('authFunc.php');

$page= new Page();

$page->content= "<h3> Login </h3>";

$page->content.= loginform();

$page->Display();

?>

*logout.php*

<?

session\_start();

require("page.php");

$old\_user= $\_SESSION['valid\_user'];

unset($\_SESSION['valid\_user']);

unset($\_SESSION['AccountType']);

session\_destroy();

$page= new Page();

$page->content= "<h1>Log Out </h1> <br />";

if(!empty($old\_user))

{

$page->content.= "Logged out. <br />";

}

else

{

$page->content.= "You were not logged in, and so have not been logged out. <br />";

}

$page->Display();

?>

*orderFunc.php*

<?php

require\_once('dbFunc.php');

require\_once('item.php');

function processCC($cardDetails)

{

return true;

}

function insertOrder($address)

{

extract($address);

$db= dbConnect();

$db->autocommit(FALSE);

//insert customer address

$query="select customerid from customer

where username=\"".$\_SESSION['valid\_user']."\"";

$result= $db->query($query);

if($result->num\_rows>0)

{

$customer= $result->fetch\_assoc();

$customerid= $customer['customerid'];

}

else

{

echo "Unable to find cutomer";

return false;

/\*$query= "insert into customers values

firstName=\"".$firstName."\",

lastName=\"".$lastName."\",

street=\"".$address."\",

city=\"".$city."\",

st=\"".$state."\",

zip=\"".$zip."";

$result= $db->query($query);

if(!$result)

{

return false;

}\*/

}

$date= date("Y-m-d");

//insert general order and shipping info

$query= "insert into orders values

('',".$customerid.", ".$\_SESSION['totalPrice'].", \"".$date."\", \"Processing\",\"".$name."\", \"".$address."\", \"".$city."\", \"".$state."\", \"".$zip."\")";

$result= $db->query($query);

if(!$result)

{

echo "unable to insert order";

return false;

}

$query= "select orderid from orders where

customerid=".$customerid." and

totalcost >(".$\_SESSION['totalPrice']."-.001) and

totalcost <(".$\_SESSION['totalPrice']."+.001) and

orderdate=\"".$date."\" and

shipName=\"".$name."\" and

shipAddress=\"".$address."\" and

shipCity=\"".$city."\" and

shipState=\"".$state."\" and

shipZip=\"".$zip."\"";

$result= $db->query($query);

if($result->num\_rows>0)

{

$order= $result->fetch\_object();

$orderid= $order->orderid;

}

else

{

echo "unable to find orderid";

return false;

}

//insert each item

foreach($\_SESSION['cart'] as $itemid=>$quantity)

{

$i= unserialize($\_SESSION['cart'][$itemid]);

$item= getItemDetails($i->itemid);

$query= "delete from orderitems where

orderid= ".$orderid." and specid=\"".$i->specid."\" and itemid = ".$i->itemid."";

$result= $db->query($query);

$query="insert into orderitems values

(".$orderid.",".$i->itemid.",\"".$i->specid."\",".$item['price'].",".$i->qty.")";

$result= $db->query($query);

if(!$result)

{

echo "unable to insert order item";

return false;

}

$cat= getCategoryName($item['catid']);

echo $cat;

$query= "update ".strtolower($cat)."

set quantity= quantity-".$i->qty."

where itemid=".$i->itemid." and exId=\"".$i->specid."\"";

$result= $db->query($query);

if(!$result)

{

return false;

}

}

$db->commit();

$db->autocommit(TRUE);

return $orderid;

}

function updateOrder($id, $status)

{

if($id== '')

{

return false;

}

$db= dbConnect();

$query= "update orders

set orderstatus=\"".$status."\"

where orderid=".$id."";

$result= $db->query($query);

if(!$result)

{

return false;

}

return true;

}

?>

*orderItems.php*

<?php

session\_start();

require\_once('page.php');

require\_once('dbFunc.php');

require\_once('item.php');

$page= new Page();

$id= $\_GET['orderid'];

$db= dbConnect();

$query= "select \* from orderitems

where orderid=".$id."";

$result= $db->query($query);

$numres= $result->num\_rows;

$page->content= "<h3>Order ".$id." Items</h3>

Number of unique items found: ".$numres."";

$page->content.="<form action=\"shipped.php\" >

<input type=\"hidden\" name=\"orderid\" value=".$id.">

<input type=\"submit\" value=\"Order Shipped\">

</form>";

for($i= 0; $i<$numres; $i++)

{

$row= $result->fetch\_assoc();

$query="select \* from merchandise

where itemid=".$row['itemid']."";

$item= $db->query($query);

$page->content.="<table width= \"25\%\">

<tr><th>Item</th><td>".($i+1)."</td>";

if(!$result)

{

$page->content.= "Item number ".$row['itemid']."not found.";

}

else

{

$itemrow= $item->fetch\_assoc();

$page->content.= "<tr><th>Name</th><td>".$itemrow['name']."</td></tr>

<tr><th>Itemid</th><td>".$row['itemid']."</td></tr>

<tr><th>Specid</th><td>".$row['specid']."</td></tr>

<tr><th>Quantity</th><td>".$row['quantity']."</td></tr>

</table>

<br />";

}

}

$page->Display();

?>

*orders.php*

<?

require\_once("page.php");

require\_once("dbFunc.php");

require\_once('item.php');

class OrdersPage extends Page

{

public function Display()

{

//<head> tag

echo "<html>\n<head>\n";

$this->DisplayTitle();

$this->DisplayKeywords();

$this->DisplayStyles();

echo "</head>\n<body>\n";

//<body> tag

$this->DisplayHeader();

$this->DisplayCart();

$this->DisplaySearch();

$this->DisplayMenu($this->buttons);

$this->DisplayOrders();

$this->DisplayFooter();

echo "</body>\n</html>\n";

}

public function DisplayOrders()

{

$db= dbConnect();

$query= "select \*

from orders";

$result= $db->query($query);

$num\_results= $result->num\_rows;

echo "<p>Number of orders retrieved: ".$num\_results."</p>";

for($i= 0; $i < $num\_results; $i++)

{

$row= $result->fetch\_assoc();

echo "<table>

<tr>

<th>".($i+1)." "." OrderID: </th>

<td><a href=\"orderItems.php?orderid=".$row['orderid']."\">".$row['orderid']."</a></td>

</tr>

<tr>

<th>

Customer:

</th>

<td>

".$row['shipName']."

</td>

</tr>

<tr>

<th>

Order Made:

</th>

<td>

".$row['orderdate']."

</td>

</tr>

<tr>

<th>

Total:

</th>

<td>

".$row['totalcost']."

</td>

</tr>

<tr>

<th>

Order Status:

</th>

<td>

".$row['orderstatus']."

</td>

</tr>

</table>

<br />";

}

$result->free();

$db->close();

}

}

$page= new OrdersPage();

$page->Display();

?>

*ordersPage.php*

<?php

session\_start();

require\_once('customerFunc.php');

require\_once('page.php');

$page= new Page();

$page->content="<h3>".$\_SESSION['valid\_user']."'s Orders:</h3><br />";

$page->content=getOrders($\_SESSION['valid\_user']);

$page->Display();

?>

*page.php*

<?php

require\_once('item.php');

require\_once('authFunc.php');

class Page

{

//the page's attributes

public $content;

public $title= "Hockey Heaven";

public $keywords= "Hockey Heaven, Hockey equipment, sticks,

skates, gloves, helmets, jerseys";

//buttons for the different web pages

public $buttons= array("Home"=> "home.php",

"Gloves"=> "gloves.php",

"Helmets"=> "helmets.php",

"Jerseys"=> "jerseys.php",

"Skates"=> "skates.php",

"Sticks"=> "sticks.php",

"Contact"=>"contact.php");

public $adminbuttons= array("AdminHome"=>"admin.php",

"Customers"=>"admincustomer.php",

"Employees"=>"adminemployee.php",

"Items"=>"adminitem.php",

"Categories"=>"admincategory.php");

public $shippingbuttons= array("Orders"=>"orders.php");

public $financebuttons= array("Financial"=>"finance.php",

"Employee Payroll"=>"viewEmployees.php",

"Stock"=>"stock.php");

//the operations for a page

public function \_\_set($name, $value)

{

$this->$name= $value;

}

//function to display the page

public function Display()

{

//<head> tag

echo "<html>\n<head>\n";

$this->DisplayTitle();

$this->DisplayKeywords();

$this->DisplayStyles();

echo "</head>\n<body>\n";

//<body> tag

$this->DisplayHeader();

$this->DisplayCart();

$this->DisplaySearch();

$this->DisplayMenu($this->buttons);

if(isAdmin())

{

$this->DisplayMenu($this->adminbuttons);

}

if(isShipping())

{

$this->DisplayMenu($this->shippingbuttons);

}

if(isFinance())

{

$this->DisplayMenu($this->financebuttons);

}

if(!isset($\_SESSION['valid\_user']))

{

$this->content= loginform();

}

echo $this->content;

$this->DisplayFooter();

echo "</body>\n</html>\n";

}

//displays the title

public function DisplayTitle()

{

echo "<title>".$this->title."</title>";

}

//places the keywords in the meta tag

public function DisplayKeywords()

{

echo "<meta name=\"keywords\"

content=\"".$this->keywords."\"/>";

}

//sets the styles for the page through css

public function DisplayStyles()

{

?>

<style>

h1 {

color:white; font-size:24pt; text-align:center;

font-family: arial, sans-serif, Verdana

}

.menu{

color:white; font-size:12pt; text-align:center;

font-family: arial, sans-serif, Verdana

}

td{

background:grey

}

td.button

{

background:blue

}

p{

color:black; font-size:12pt; text-align:justify;

font-family:arial, sans-serif,Verdana

}

p.foot{

color:white; font-size:9pt; text-align:center;

font-family:arial, sans-serif, Verdana;

font-weight: bold

}

a:link, a:visited, a:active{

color:white

}

a.title{

color:#0055FF

}

</style>

<?php

}

public function DisplayHeader()

{

?>

<table width="100%" cellpadding="12"

cellspacing="0" border="0">

<tr class="button">

<td align="left" class="button"><a href="home.php"><img src="Images/pond\_hockey.jpg" width="120" height="90"/></a></td>

<td class="button">

<a href="home.php"><h1>Hockey Heaven</h1></a>

</td>

<?

if(isset($\_SESSION['valid\_user']))

{

?>

<td align= "right" class="button"><a href="accountPage.php"><? echo $\_SESSION['valid\_user'] ?></a> | <a href="logout.php">Logout</a>

<? }

else

{

?>

<td align="right" class="button"><a href="login.php">Login</a>

<? }

?>

</td>

</tr>

</table>

<?php

}

public function DisplaySearch()

{

?>

<form name="search" action="searchResults.php" method="post">

Search <select name="merchType">

<option value="all">All</option>

<option value="Gloves">Gloves</option>

<option value="Helmets">Helmets</option>

<option value="Jerseys">Jerseys</option>

<option value="Skates">Skates</option>

<option value="Sticks">Sticks</option>

</select>

<input type="text" name="value">

<input type="submit" value="Go">

</form>

<?

}

public function DisplayCart()

{

?>

<div style="text-align:right;">

<a href="cart.php" id="Shopping Cart" style="color:blue">Cart</a>

</div>

<?

}

public function DisplayMenu($buttons)

{

echo "<table width=\"100%\" bgcolor=\"white\"

cellpadding=\"4\" cellspacing=\"4\">\n";

echo "<tr>\n";

//calculate the button size

$width= 100/count($buttons);

while(list($name, $url)= each($buttons))

{

$this->DisplayButton($width, $name, $url,

!$this->IsURLCurrentPage($url));

}

echo "</tr>\n";

echo "</table>\n";

}

public function IsURLCurrentPage($url)

{

if(strpos($\_SERVER['PHP\_SELF'], $url)==false)

{

return false;

}

else

{

return true;

}

}

public function DisplayButton($width, $name, $url, $active= true)

{

if($active)

{

echo "<td width= \"".$width."%\" class=\"button\">

<a href=\"".$url."\">

<!--<img src=\"s-logo.gif\" alt=\"".$name."\" border=\"0\" /> </a>-->

<a href=\"".$url."\"><span class=\"menu\">".$name."</span></a>

</td>";

}

else

{

echo "<td width=\"".$width."%\">

<!--<img src=\"side-logo.gif\">-->

<span class=\"menu\">".$name."</span>

</td>";

}

}

public function DisplayFooter()

{

?>

<table width="100%" bgcolor="blue" cellpadding="12" border="0">

<tr>

<td class="button">

<p class="foot">&copy; Hockey Heaven</p>

<p class="foot">Database project <a href="bibliography.php" >bibliography</a>.</p>

</td>

</tr>

</table>

<?php

}

}

?>

*process.php*

<?php

session\_start();

require\_once('page.php');

require\_once('item.php');

require\_once('itemFunc.php');

require\_once('cartForm.php');

require\_once('orderFunc.php');

$page= new Page();

$cardType = $\_POST['cardType'];

$cardNumber = $\_POST['cardNum'];

$cardMonth = $\_POST['month'];

$cardYear = $\_POST['year'];

$cardName = $\_POST['cardName'];

$id= $\_POST['id'];

if(($\_SESSION['cart']) && ($cardType) && ($cardNumber) &&

($cardMonth) && ($cardYear) && ($cardName))

{

$page->content=DisplayCartItems($\_SESSION['cart'], $\_SESSION['totalPrice'], $\_SESSION['items'], false);

$page->content.=DisplayShipping(CalculateShipping());

if(processCC($\_POST))

{

if(updateOrder($id, "Processing"))

{

session\_destroy();

$page->content.= "Thank you for shopping with us. Your order has been placed.";

$page->content.="<table><form action=\"home.php\"><input type=\"submit\" value=\"Continue Shopping\"></form></table>";

}

else

{

$page->content.="Unable to process order. Please try again later.";

$page->content.="<table><form action=\"purchase.php</table><input type=\"submit\" value=\"Purchase\"></form></table>";

}

}

else

{

$page->content.="Card could not be processed. Please try again later.";

$page->content.="<table><form action=\"purchase.php</table><input type=\"submit\" value=\"Purchase\"></form></table>";

}

}

else

{

$page->content.= "<p>You did not fill in all the fields, please try again.</p><hr />";

$page->content.="<table><form action=\"purchase.php</table><input type=\"submit\" value=\"Purchase\"></form></table>";

}

$page->Display();

?>

*purchase.php*

<?php

session\_start();

require\_once('page.php');

require\_once('item.php');

require\_once('cartForm.php');

require\_once('itemFunc.php');

require\_once('orderFunc.php');

$page= new Page();

$name= $\_POST['name'];

$address= $\_POST['address'];

$city= $\_POST['city'];

$state= $\_POST['state'];

$zip= $\_POST['zip'];

if (($\_SESSION['cart']) && ($name) && ($address) && ($city) && ($zip))

{

$id= false;

if($id= insertOrder($\_POST))

{

$page->content= DisplayCartItems($\_SESSION['cart'], $\_SESSION['totalPrice'], $\_SESSION['totalItems'], false);

$page->content.= DisplayShipping(calculateShipping());

$page->content.=DisplayCreditCardForm($name, $id);

}

else

{

$page->content= "Unable to store order. Please try again later";

}

}

else

{

$page->content= "You did not properly fill out the form. Please try again.";

}

$page->Display();

?>

*reviewFunc.php*

<?php

require\_once('dbFunc.php');

require\_once('itemFunc.php');

function ReviewForm($id, $rev= '')

{

$edit= is\_array($rev);

$db= dbConnect();

$item= getItemDetails($id);

$form="<h3>Review for ".$item['name']."";

$submitTo=$edit? 'editReview.php':'insertReview.php';

$blurb=$edit?$rev['blurb']:'';

$score= $edit?$rev['score']:'';

//if passed an existing item, proceed in edit mode

$form.= "<form method=\"post\"

action=\"".$submitTo."\">

<table border=\"0\">

<tr>

<td>

<img src=\"Images/".$item['name'].".jpg\" height=\"100\" width=\"100\"/>

</td>

<td>

<b>Score:</b>

<select name=\"score\">

<option value=\"0\">0</option>

<option value=\"1\">1</option>

<option value=\"2\">2</option>

<option value=\"3\">3</option>

<option value=\"4\">4</option>

<option value=\"5\">5</option>

</select>

</td>

</tr>

<tr>

<td><b>Review:</b></td>

<td><textarea rows=\"20\" cols=\"45\" name= \"blurb\" value=\"blurb\">".$blurb."</textarea></td>

</tr>

<tr>";

if(!$edit)

{

$form.="<td colspan=\"2\" align=\"center\">";

}

else

{

$form.="<td align=\"center\"><input type=\"hidden\" name=\"catid\"

value=\"".$cat['catid']."\" />";

}

$submission= $edit? 'Update': 'Submit';

$form.="<input type=\"hidden\" name=\"itemid\" value=\"".$id."\">

<input type=\"submit\" value=\"".$submission." Review \"/>

</form></td>";

if($edit)

{

$form.="<td>

<form method=\"post\" action=\"deleteReview.php\">

<input type=\"hidden\" name=\"itemid\"

value=\"".$id."\" />

<input type=\"submit\" value=\"Delete Review\" />

</form></td>";

}

$form.="</td>

</tr>

</table>";

return $form;

}

function submitReview($item, $user, $score, $blurb)

{

$db= dbConnect();

$query= "insert into reviews

values (".$item.",".$user.",".$score.",\"".$blurb."\")";

$result= $db->query($query);

if($result)

{

return true;

}

else

{

return false;

}

}

function getReviews($item)

{

$db= dbConnect();

$query= "select \*

from reviews

where itemid=".$item."";

$result= $db->query($query);

if(!$result || $result->num\_rows == 0)

{

return "There are no reviews for this item.";

}

$reviews= "";

while($row= $result->fetch\_assoc())

{

$query= "select username from customer

where customerid=".$row['customerid']."";

$result2= $db->query($query);

$customer= $result2->fetch\_assoc();

$reviews.="<table width=\"40\%\">

<tr><th>Review by:</th> <td>".$customer['username']."</td></tr>

<tr><th>Score:</th><td>".$row['score']."/5</td></tr>

<tr><th>Review:</th><td>".$row['blurb']."</td></tr>

</table>";

}

return $reviews;

}

?>

*searchResults.php*

<?php

require("page.php");

require("dbFunc.php");

class ResultsPage extends Page

{

public function Display()

{

//<head> tag

echo "<html>\n<head>\n";

$this->DisplayTitle();

$this->DisplayKeywords();

$this->DisplayStyles();

echo "</head>\n<body>\n";

//<body> tag

$this->DisplayHeader();

$this->DisplayCart();

$this->DisplaySearch();

$this->DisplayMenu($this->buttons);

if(isAdmin())

{

$this->DisplayMenu($this->adminbuttons);

}

if(isShipping())

{

$this->DisplayMenu($this->shippingbuttons);

}

if(isFinance())

{

$this->DisplayMenu($this->financebuttons);

}

echo $this->content;

$this->DisplaySearchResults();

$this->DisplayFooter();

echo "</body>\n</html>\n";

}

public function DisplaySearchResults()

{

//create short variable names

$merchType= $\_POST['merchType'];

$value= $\_POST['value'];

if(!$merchType || !$value)

{

echo "Search could not be completed. You have not entered search details. Please try again.";

echo "<br /> <br /> <br /> ";

return;

}

$db= dbConnect();

if($merchType == "all")

{

$query= "select \* from merchandise where name like '%".$value."%'";

}

else

{

$query= "select catid from categories where catname like '%".$merchType."%'";

$result= $db->query($query);

if($result->num\_rows == 0)

{

return false;

}

$innerRow= $result->fetch\_assoc();

$query="select \* from merchandise where catid=".$innerRow['catid']." and name like '%".$value."%'";

}

$result= $db->query($query);

$num\_results= $result->num\_rows;

echo "<p>Number of items found: ".$num\_results."</p>";

for($i= 0; $i < $num\_results; $i++)

{

$row= $result->fetch\_assoc();

echo "<p><b>".($i+1)." "." Name: ";

echo "<a style=\"color:#800000\" href=\"itemPage.php?itemid=".$row['itemid']."\">".htmlspecialchars($row['name'])."</a>";

echo "</b><br />Manufacturer: ";

echo $row['manufacturer'];

echo "<br />ItemID: ";

echo $row['itemid'];

echo "<br /> Price: ";

echo $row['price'];

echo "</p>";

}

$result->free();

$db->close();

}

}

$page= new ResultsPage();

$page->Display();

?>

*shipped.php*

<?php

session\_start();

require\_once("page.php");

require\_once("dbFunc.php");

require\_once("orderFunc.php");

$page= new Page();

if(updateOrder($\_GET['orderid'], "Shipped"))

{

$page->content= "Order has been updated to \"Shipped\"";

}

else

{

$page->content= "Unable to update order.";

}

$page->Display();

?>

*skates.php*

<?

session\_start();

if(!isset($\_SESSION['valid\_user']))

{

header("Location: home.php");

exit;

}

require\_once("page.php");

require\_once('item.php');

require\_once("dbFunc.php");

class SkatePage extends Page

{

public function Display()

{

//<head> tag

echo "<html>\n<head>\n";

$this->DisplayTitle();

$this->DisplayKeywords();

$this->DisplayStyles();

echo "</head>\n<body>\n";

//<body> tag

$this->DisplayHeader();

$this->DisplayCart();

$this->DisplaySearch();

$this->DisplayMenu($this->buttons);

if(isAdmin())

{

$this->DisplayMenu($this->adminbuttons);

}

if(isShipping())

{

$this->DisplayMenu($this->shippingbuttons);

}

if(isFinance())

{

$this->DisplayMenu($this->financebuttons);

}

$this->DisplaySkates();

$this->DisplayFooter();

echo "</body>\n</html>\n";

}

public function DisplaySkates()

{

$db= dbConnect();

//find the catid of gloves

$query= "select catid from categories

where catname= \"skates\"";

$result= $db->query($query);

if($result->num\_rows > 0)

{

$rowid= $result->fetch\_assoc();

$id= $rowid['catid'];

//find all merchandise with the gloves catid

$query="select \* from merchandise

where catid=".$id."";

$result= $db->query($query);

}

$num\_results= $result->num\_rows;

echo "<p>Number of skates retrieved: ".$num\_results."</p>";

for($i= 0; $i < $num\_results; $i++)

{

$row= $result->fetch\_assoc();

echo "<p><b>".($i+1)." "." Name: ";

echo "<a style=\"color:#800000\" href=\"itemPage.php?itemid=".$row['itemid']."\" >".$row['name']."</a>";

echo "</b><br />Manufacturer: ";

echo $row['manufacturer'];

echo "<br />ItemID: ";

echo $row['itemid'];

echo "<br /> Price: ";

echo $row['price'];

echo "</p>";

}

$result->free();

$db->close();

}

}

$page= new SkatePage();

$page->Display();

?>

*specItem.php*

<?php

session\_start();

require\_once("page.php");

require\_once("dbFunc.php");

require\_once("item.php");

require\_once("itemFunc.php");

require\_once('adminFunc.php');

$page= new Page();

$page->content=insertSpecForm($\_GET['item']);

$page->Display();

?>

*sticks.php*

<?

session\_start();

if(!isset($\_SESSION['valid\_user']))

{

header("Location: home.php");

exit;

}

require\_once("page.php");

require\_once('item.php');

require\_once("dbFunc.php");

class StickPage extends Page

{

public function Display()

{

//<head> tag

echo "<html>\n<head>\n";

$this->DisplayTitle();

$this->DisplayKeywords();

$this->DisplayStyles();

echo "</head>\n<body>\n";

//<body> tag

$this->DisplayHeader();

$this->DisplayCart();

$this->DisplaySearch();

$this->DisplayMenu($this->buttons);

if(isAdmin())

{

$this->DisplayMenu($this->adminbuttons);

}

if(isShipping())

{

$this->DisplayMenu($this->shippingbuttons);

}

if(isFinance())

{

$this->DisplayMenu($this->financebuttons);

}

$this->DisplaySticks();

$this->DisplayFooter();

echo "</body>\n</html>\n";

}

public function DisplaySticks()

{

$db= dbConnect();

//find the catid of gloves

$query= "select catid from categories

where catname= \"sticks\"";

$result= $db->query($query);

if($result->num\_rows > 0)

{

$rowid= $result->fetch\_assoc();

$id= $rowid['catid'];

//find all merchandise with the gloves catid

$query="select \* from merchandise

where catid=".$id."";

$result= $db->query($query);

}

$num\_results= $result->num\_rows;

echo "<p>Number of sticks retrieved: ".$num\_results."</p>";

for($i= 0; $i < $num\_results; $i++)

{

$row= $result->fetch\_assoc();

echo "<p><b>".($i+1)." "." Name: ";

echo "<a style=\"color:#800000\" href=\"itemPage.php?itemid=".$row['itemid']."\" >".$row['name']."</a>";

echo "</b><br />Manufacturer: ";

echo $row['manufacturer'];

echo "<br />ItemID: ";

echo $row['itemid'];

echo "<br /> Price: ";

echo $row['price'];

echo "</p>";

}

$result->free();

$db->close();

}

}

$page= new StickPage();

$page->Display();

?>

*stock.php*

<?php

session\_start();

require\_once("dbFunc.php");

require\_once('page.php');

require\_once('itemFunc.php');

require\_once('financeFunc.php');

$page= new Page();

$page->content= getMerchandise();

$page->Display();

?>

*updateSalary.php*

<?php

session\_start();

require\_once('page.php');

require\_once('dataValid.php');

require\_once('adminFunc.php');

require\_once('financeFunc.php');

$page= new Page();

$empid= $\_POST['empid'];

$amount= $\_POST['paymentamount'];

if(filledOut($\_POST))

{

if(updateSalary($amount, $empid))

{

$page->content= "Salary updated";

}

else

{

$page->content= "Salary not updated";

}

}

else

{

$page->content= "The form was not filled out correctly. Please try again.";

}

$page->Display();

?>

*viewCustomer.php*

<?php

session\_start();

require\_once('dbFunc.php');

require\_once('page.php');

require\_once('adminFunc.php');

$page=new Page();

$page->content= getCustomers();

$page->Display();

?>

*viewEmployees.php*

<?php

session\_start();

require\_once('dbFunc.php');

require\_once('page.php');

require\_once('adminFunc.php');

$page=new Page();

$page->content= getEmployees();

$page->Display();

?>

*writeReview.php*

<?php

session\_start();

require\_once('reviewFunc.php');

require\_once('page.php');

$id= $\_GET['itemid'];

$page= new Page();

$page->content= ReviewForm($id);

$page->Display();

?>

**SQL Code**

-- MySQL dump 10.11

--

-- Host: localhost Database: mindac\_HH

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

-- Server version 5.0.96-community-log

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8 \*/;

/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;

/\*!40103 SET TIME\_ZONE='+00:00' \*/;

/\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;

--

-- Table structure for table `categories`

--

DROP TABLE IF EXISTS `categories`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `categories` (

`catid` int(10) unsigned NOT NULL auto\_increment,

`catname` char(10) NOT NULL,

PRIMARY KEY (`catid`)

) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `categories`

--

LOCK TABLES `categories` WRITE;

/\*!40000 ALTER TABLE `categories` DISABLE KEYS \*/;

INSERT INTO `categories` VALUES (1,'Helmets'),(2,'Gloves'),(3,'Jerseys'),(4,'Skates'),(5,'Sticks');

/\*!40000 ALTER TABLE `categories` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `creditcard`

--

DROP TABLE IF EXISTS `creditcard`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `creditcard` (

`ccnum` char(16) NOT NULL,

`customerid` int(10) unsigned NOT NULL,

`name` char(60) default NULL,

`expdate` char(5) default NULL,

`secnum` char(3) default NULL,

`billstreet` char(30) default NULL,

`billcity` char(30) default NULL,

`billstate` char(2) default NULL,

`zip` char(5) default NULL,

PRIMARY KEY (`ccnum`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `creditcard`

--

LOCK TABLES `creditcard` WRITE;

/\*!40000 ALTER TABLE `creditcard` DISABLE KEYS \*/;

/\*!40000 ALTER TABLE `creditcard` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `customer`

--

DROP TABLE IF EXISTS `customer`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `customer` (

`customerid` int(10) unsigned NOT NULL auto\_increment,

`firstName` char(20) NOT NULL,

`lastName` char(20) NOT NULL,

`street` char(45) NOT NULL,

`city` char(30) NOT NULL,

`st` char(2) default NULL,

`zip` char(5) default NULL,

`email` char(40) default NULL,

`username` char(20) default NULL,

`password` char(40) NOT NULL,

PRIMARY KEY (`customerid`)

) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `customer`

--

LOCK TABLES `customer` WRITE;

/\*!40000 ALTER TABLE `customer` DISABLE KEYS \*/;

INSERT INTO `customer` VALUES (1,'Frank','Smith','654 Franklin Place','Paramus','NJ','07452','fsmith@gmule.com','smithy','fa9edc42d09e8228ef1cebaf571c291bf9f317bd');

/\*!40000 ALTER TABLE `customer` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `employees`

--

DROP TABLE IF EXISTS `employees`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `employees` (

`employeeid` int(10) unsigned NOT NULL auto\_increment,

`username` char(16) NOT NULL,

`ssn` char(9) NOT NULL,

`category` char(10) default NULL,

`firstName` char(20) NOT NULL,

`lastName` char(20) NOT NULL,

`street` char(45) NOT NULL,

`city` char(30) NOT NULL,

`st` char(2) default NULL,

`zip` char(5) default NULL,

`email` char(40) default NULL,

`phone` char(12) default NULL,

`password` char(40) NOT NULL,

`paymenttype` char(10) NOT NULL default 'hourly',

`hoursworked` tinyint(3) unsigned NOT NULL default '0',

`paymentamount` float(8,2) NOT NULL,

PRIMARY KEY (`ssn`),

UNIQUE KEY `username` (`username`),

UNIQUE KEY `employeeid` (`employeeid`)

) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `employees`

--

LOCK TABLES `employees` WRITE;

/\*!40000 ALTER TABLE `employees` DISABLE KEYS \*/;

INSERT INTO `employees` VALUES (3,'batman','iambatman','finance','Bruce','Wayne','4 Wayne Manor Drive','Gotham City','NY','09042','bwayne@wanye.org','212-894-4444','300646b7440f09ef1a8bc0ca8cd5504cf236cd5b','salary',0,85732.57),(4,'deadpool','somepun4u','shipping','Wade','Wilson','23 Marvel Street','Carlstadt','NJ','07045','mercwithamouth@internet.org','111-111-1111','2cacb144c135bc943c89b7d66b2618019018e320','hourly',0,15.75),(2,'cminda','thisafake','admin','Chris','Minda','123 Washington Ave.','Hackensack','NJ','07602','cminda@thatsite.net','474-848-9295','756528af53c11a25f37e639d12e34126460c10da','salary',0,45534.56);

/\*!40000 ALTER TABLE `employees` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `gloves`

--

DROP TABLE IF EXISTS `gloves`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `gloves` (

`exId` char(6) NOT NULL,

`itemid` int(10) NOT NULL,

`color` char(15) NOT NULL,

`glovesize` int(11) NOT NULL,

`quantity` int(11) NOT NULL,

KEY `itemid` (`itemid`),

CONSTRAINT `gloves\_ibfk\_3` FOREIGN KEY (`itemid`) REFERENCES `merchandise` (`itemid`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `gloves`

--

LOCK TABLES `gloves` WRITE;

/\*!40000 ALTER TABLE `gloves` DISABLE KEYS \*/;

INSERT INTO `gloves` VALUES ('bl13',1,'Black',13,14),('blg13',1,'Black/Gold',13,10),('blg15',1,'Black/Gold',15,3),('blo13',3,'Black/Orange',13,12),('blo14',3,'Black/Orange',14,4),('brw14',2,'Black/Red/White',14,10),('brw15',2,'Black/Red/White',15,7),('bw15',2,'Black/White',15,4),('Nav14',1,'Navy',14,5),('Nav15',1,'Navy',15,7),('nw13',2,'Navy/White',13,3),('blo15',3,'Black/Orange',15,6),('brw15',3,'Black/Red/White',15,2),('blo13',4,'Black/Orange',13,3),('blo15',4,'Black/Orange',15,11),('nw14',4,'Navy/White',14,7),('brw13',4,'Black/Red/White',13,9),('nav13',5,'Navy',13,2),('br14',5,'Black/Red',14,7),('rw15',1,'Red/White',15,22);

/\*!40000 ALTER TABLE `gloves` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `helmets`

--

DROP TABLE IF EXISTS `helmets`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `helmets` (

`exId` char(6) NOT NULL,

`itemid` int(10) NOT NULL,

`color` char(20) NOT NULL,

`helmetsize` char(4) NOT NULL,

`quantity` int(11) NOT NULL,

KEY `itemid` (`itemid`),

CONSTRAINT `helmets\_ibfk\_1` FOREIGN KEY (`itemid`) REFERENCES `merchandise` (`itemid`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `helmets`

--

LOCK TABLES `helmets` WRITE;

/\*!40000 ALTER TABLE `helmets` DISABLE KEYS \*/;

INSERT INTO `helmets` VALUES ('bls',6,'Black','S',7),('blm',6,'Black','M',5),('bll',6,'Black','L',12),('bll',6,'White','L',2),('navxl',6,'Navy','XL',3),('bls',7,'Black','S',4),('roys',7,'Royal','S',6),('redl',7,'Red','L',3),('wl',7,'White','L',4),('blxs',8,'Black','XS',5),('bls',8,'Black','S',8),('blm',8,'Black','M',2),('bll',8,'Black','L',4),('wm',8,'White','M',4),('blbls',9,'Black w/ Black Vents','S',4),('blgrm',9,'Black w/ Grey Vents','M',12),('navgrl',9,'Navy w/ Grey Vents','L',4),('blblm',9,'Black w/ Black Vents','M',7),('bls',10,'Black','S',5),('ws',10,'White','S',14),('blm',10,'Black','M',3),('wm',10,'White','M',12),('bll',10,'Black','L',3),('wxl',10,'White','XL',5);

/\*!40000 ALTER TABLE `helmets` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `jerseys`

--

DROP TABLE IF EXISTS `jerseys`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `jerseys` (

`exId` char(6) NOT NULL,

`itemid` int(10) NOT NULL,

`color` char(15) NOT NULL,

`team` char(20) NOT NULL,

`jerseysize` char(10) NOT NULL,

`quantity` int(11) NOT NULL,

KEY `itemid` (`itemid`),

CONSTRAINT `jerseys\_ibfk\_1` FOREIGN KEY (`itemid`) REFERENCES `merchandise` (`itemid`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `jerseys`

--

LOCK TABLES `jerseys` WRITE;

/\*!40000 ALTER TABLE `jerseys` DISABLE KEYS \*/;

INSERT INTO `jerseys` VALUES ('bllxl',11,'Black','Practice','Large/XL',7),('blsm',11,'Black','Practice','Small/Med',5),('dgl',12,'Dark Green','Practice','Large',8),('dgm',12,'Dark Green','Practice','Medium',10),('dgs',12,'Dark Green','Practice','Small',14),('dgxl',12,'Dark Green','Practice','XL',3),('navl',13,'Navy','Practice','Large',3),('navm',13,'Navy','Practice','Medium',11),('navs',13,'Navy','Practice','Small',13),('orl',13,'Orange','Practice','Large',17),('purlxl',11,'Purple','Practice','Large/XL',20),('pursm',11,'Purple','Practice','Small/Med',4),('redm',13,'Red','Practice','Medium',12),('roylxl',11,'Royal','Practice','Large/XL',17),('roysm',11,'Royal','Practice','Small/Med',6),('silm',13,'Silver','Practice','Medium',14),('sils',13,'Silver','Practice','Small',12),('blxl',14,'Black','Practice','XL',17),('whm',14,'White','Practice','Medium',11),('redxxl',14,'Red','Practice','XXL',15),('wxl',14,'White','Practice','XL',9),('navxl',14,'Navy','Practice','XL',10),('blum',15,'Blue','Practice','Medium',14),('blul',15,'Blue','Practice','Large',8),('bluxl',15,'Blue','Practice','XL',7),('bws',16,'Black/White','Practice','Small',14),('bwl',16,'Black/White','Practice','Large',4),('rwm',16,'Red/White','Practice','Medium',12),('rwl',16,'Red/White','Practice','Large',11),('nvwxxl',16,'Navy/White','Practice','XXL',2),('nvwgol',16,'Navy/White','Practice','Goalie',12),('gws',16,'Green/White','Practice','Small',6),('gwgol',16,'Green/White','Practice','Goalie',4);

/\*!40000 ALTER TABLE `jerseys` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `merchandise`

--

DROP TABLE IF EXISTS `merchandise`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `merchandise` (

`itemid` int(11) NOT NULL auto\_increment,

`name` char(80) NOT NULL,

`manufacturer` char(15) NOT NULL,

`catid` int(10) unsigned NOT NULL,

`price` float(5,2) default NULL,

`description` text,

PRIMARY KEY (`itemid`)

) ENGINE=InnoDB AUTO\_INCREMENT=68 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `merchandise`

--

LOCK TABLES `merchandise` WRITE;

/\*!40000 ALTER TABLE `merchandise` DISABLE KEYS \*/;

INSERT INTO `merchandise` VALUES (1,'Bauer Supreme TOTALONE Senior Hockey Gloves','Bauer',2,149.99,'Foams - Dual Density foams featuring MAX-SORB protection <br />\r\n\r\nThumb - Patented by 3-Piece Free Flex Lock Thumb\r\nInserts - Poly\r\n\r\nShell - Lightweight stretch matrix nylon, textured synthetic leather\r\n\r\nFIT/MOBILITY :\r\nOverall Fit - Anatomical\r\nBackhand Pad - Memory foam for personalized fit\r\nFingers - 3 piece index finger\r\nPalm - Nubuck nash with gecko nash overlay\r\nBack Roll - Center segmentation\r\nCuff - Pro multi segmented cuff\r\nLiner - THERMO MAX+'),(2,'CCM U+ 06 Senior Hockey Gloves','CCM',2,69.99,' Thumb: New flex thumb\r\nBackhand: Soft triple decker foam construction\r\nPalm: Reinforced Nash with nylon mesh gussets\r\nCuff: Pro open cuff roll\r\nFingers: Soft triple decker foam construction\r\nLiner: Soft fabric liner with comfort foam\r\nExterior: Hybrid nylon / PU (weekly usage)'),(3,'Easton Stealth 85S Senior Hockey Gloves','Easton',2,124.99,'Enhanced range of motion and mobility with new angled cuff roll design\r\nPalm is Nash base with AX Suede? overlay for durability and excellent feel\r\nTriple-segmented index, middle and ring fingers provide excellent mobility\r\nLightweight shell provides the rich look, while PE inserts over mid-density foams provide protection\r\n'),(4,'Reebok 7k KFS Senior Hockey Gloves','Reebok',2,109.99,'Cuff - Seamless flexible cuff with now binding for increased comfort and flexibility\r\nPalm - Combination stretch and nash palm\r\nProtection - Plastic inserts in backhand protect against slashes\r\nBlocking Pattern - KFS locking pattern allows the glove to follow the natural movement of the hand with HD foam and PE inserts'),(5,'Warrior Remix Senior Hockey Gloves- 2012','Warrior',2,124.99,'4 Bar style design, classic pro look with a refined traditional high-volume fit and feel\r\nLightweight Tufftek outer shell\r\nDual-layered foam with plastic insert protection\r\nDual-layered extra durable Clarino Authentic Pro Palm\r\nV-Tek? micro-mesh liner controls moisture and odor'),(6,'Bauer 9900 Hockey Helmet','Bauer',1,139.99,' FXPP Liner - Located in key impact zones, the Fused Expanded Polypropylene foam liner helps dissipate 20% less energy during impact than the same volume of traditional EPP foam\r\nBiomechanical Shell - Dual-density HDPE shell combines both protective and flexible material, allowing the helmet to self-adjust to the vast head shapes in the game and provide a better, more personalized fit\r\nPORON® XRD? Technology - Exclusive to BAUER Hockey, PORON® XRD? Extreme impact Protection is a super-light, pliable material with the ability to dissipate more than 90% of force on impact. Placed strategically in the helmet, PORON® XRD? offers breathable and flexible protection while consistently managing impact where you head needs it most\r\nSpecs\r\nGameday Protection: FXPP foam protection / Triple density impact management foam / PORON® XRD? / Ergo translucent ear covers / CSA, HECC, CE certified\r\nCustomized Fit: Bio-mechanical shell / Occipital lock 2.0 / Tool-free adjustment / Three distinct shell sizes\r\nComfort: Slow memory comfort foam / Ventilation ports / Molded memory foam temple pads / MICROBAN® antibacterial protection\r\n'),(7,'CCM Vector V10 Hockey Helmet with Cage','CCM',1,149.99,'Brand new ergonomic, aerodynamic shell design\r\nEPP liner for multiple impact protection\r\nHeat activated molded memory foam for comfort\r\nLift and lock tool-free length adjustment\r\nExtensive venting for maximum air flow'),(8,'Easton E600 Hockey Helmet','Easton',1,124.99,'Full coverage EPP foam liner with PC shell and HDPE subshell to provide multi-impact durability\r\nGiro inspired Fit system combined with plush padding for excellent stability and best in class comfort\r\nA lightweight performance helmet that has excellent crush resistant properties'),(9,'Reebok 11k Hockey Helmet with Cage','Reebok',1,204.99,'New Reebok Shell - Next generations aerodynamic shell with ventilation designed for a low-profile fit\r\nEPP Floating Liner - EPP flexible liner with comfort cushions contours to the unique shape of your head\r\nMicrodial II - Permits 360 degree wrap and vertical adjustments to alleviate pressure points and hold helmet firmly in place\r\nFit Clips - Permits easy tool-less length adjustment\r\nSubshells - Using carbon fiber technology that revolutionized the stick market, Reebok designed a lightweight helmet without sacrificing protection'),(10,'Warrior Krown 360 Hockey Helmet','Warrior',1,159.99,'\r\nIntroducing the KROWN 360 ? a unique helmet design focused on paramount fit and layered impact protection for comfort and safety.\r\n\r\nThe Krown 360 uses a unique helmet design focused on paramount fit and layered impact protection for comfort and safety. Its four-piece shell and liner converge to create a true 360 degree fit, with a one-of-a-kind single dial adjustment that conveniently activates the 4-Play Fit System.\r\n\r\n\r\nWarrior Krown Helmet Features159.99\r\nEPP foam for lightweight, high-impact protection\r\nComfort foam pads for fit stability and cushioning\r\nCSA, HECC and CE certified\r\nFour precisely sized shells (S, M, L and XL)\r\nStrategically located Impax Molded Foam for protection against multiple low impacts\r\n360° FIT\r\n\r\nFour-piece shell and liner converge to create a true 360 degree fit\r\nA unique single dial adjustment conveniently activates the 4-Play Fit System\r\n360° PROTECTION\r\n\r\nThree intelligently selected, impact-specific layers of foam protection:\r\nEFP for lightweight, high-impact protection'),(11,'CCM 10200 Hockey Practice Jersey','CCM',3,19.99,'CCM patented lightweight Air-Knit fabric\r\n100% polyester double knit\r\nSingle-layer shoulders\r\nCross-over interlock collar'),(12,'CCM 17000 Center Ice Hockey Jersey- Dallas Stars','CCM',3,79.99,'Patented Air-Knit 100% polyester double knit\r\nReinforced stitching on shoulders and armholes\r\nDouble shoulders\r\nInterlock neck trim\r\nCCM/NHL neck label\r\nExclusive NHL replica practice jersey\r\nChest logos feature a combination of embroidery and applique twill\r\nCCM embroidery patch on back of the neck and front hem\r\nNHL Center Ice embroidered patch on front upper chest\r\n'),(13,'Kamazu 14100 Flexx Lite Hockey Jersey','Kamazu',3,29.99,'\r\nKamazu patented lightweight Air-Knit fabric\r\n100% Polyester double knit\r\nSingle-layer shoulders\r\nCross-over interlock collar'),(14,'Nike Bauer Hockey Goalie Jersey','Bauer',3,24.99,'Content : 100% Polyester Mesh\r\nV-neck bounded collar opening with front and back yoke. NIKE BAUER design trademark screen print across back bottom hem. Front of jersey open for team embellishment\r\nGoalie Cut Jersey'),(15,'RBK 27000 Platinum Performance Hockey Practice Jersey- Toronto Maple Leafs','Reebok',3,79.99,'Mid-weight ULTRAFIL 100% polyester\r\nExclusive replica practice jersey of the NHL\r\nPolyester mesh gusset inserts for improved ventilation\r\nChest crests feature a combination of embroidery and applique twill\r\nReflective shoulder and sleeve braid\r\nInterlock neck trim\r\nReinforced stitching on shoulders and armholes\r\nReebok logo direct embroidered patch on front neckline insert\r\nNHL shield embroidered patch on front neckline insert\r\nReebok/NHL licensed jersey woven neck label\r\n'),(16,'Reebok 20P00 Edge Practice Jersey','Reebok',3,39.99,'\r\nMid weight Ultrafil base with contrasting inserts\r\nMesh inserts\r\n100% polyester\r\nEmbroidered Reebok logos'),(17,'Bauer Nexus 1000','Bauer',4,749.99,'The Nexus skate series is where tradition meets technology, for the player who is looking \r\nfor maximum comfort and explosive acceleration.\r\n\r\nThey have a classic fit profile, which means that you sit deeper in this boot with more \r\nwrap and width than what you would experience in either of the Supreme or Vapor series \r\nskates.\r\n\r\nBauer uses an ultra light weight yet robust Tech Mesh quarter panel in the boot. It is \r\nfully heat moldable to help you achieve an optimal fit with minimal break in time. This \r\nconstruction provides pro-level support and also allows for efficient energy transfer \r\nbetween you and your skate. Beneath the hydrophobic retro tan Clarino® liner, the deep V \r\nheel fit is enhanced with very generous anatomical foam ankle pads that not only provide \r\na great amount of heel lock, but also make the skates very comfortable right out of the \r\nbox. The Nexus 1000 also features Total Edge comfort which helps alleviate abrasion high \r\non the ankle.\r\n\r\nDouble lace bite protection is achieved from the two-piece design of the ultra-thick pro \r\nfelt tongue and the molded metatarsal guard which also helps protect the front of the foot\r\nwhile blocking shots. This is the thickest white felt tongue that Bauer offers, but for \r\nas robust as it is, it still allows for a good amount of forward flex.\r\n\r\nIt has an ultra lightweight vented composite outsole which provides a nice rigid platform\r\n for the boot and reduces torque under the boot while you\'re skating.\r\n\r\nIf you\'re looking for a top-of-the-line classic retro-style skate that is infused with\r\n current skate technologies, look no further than the Nexus 1000 from Bauer.'),(18,'CCM Uplus12','CCM',4,479.99,'-U Foam Pro core construction\r\n-Vector Armor surlyn quarter package\r\n-Moisture wicking Clarino interior liner\r\n-Carbon composite exhaust outsole\r\n-Custom support exhaust insole\r\n-CCM Armor tongue\r\n-CCM E-Pro Holder with ProFormance Lite Stainless Steel Runner\r\n-Skate Weight: One skate, in size 8.0 D weighs 856 grams/30.19 ounces'),(19,'Easton Stealth s12','Easton',4,449.99,'-This skate tends to fit 1/2 Size smaller than your shoe size\r\n-Unmatched composite value with more lateral flex in the upper ankle\r\n-Coil Technology? texalium-glass composite boot form with Dryflow? - light, flexible, protective\r\n-Stealth Padlock ? molded anatomical ankle shape locks down heel for increased power\r\n-Pro-style white felt tongue with flexible lace-bite protection\r\n-Bio-Dri? Microfiber liner provides added grip and classic feel\r\n-InTS Integrated Tendon Structure is comfortable and functional\r\n-Easton Bio-Dri? Composite footbed with removable heel lift\r\n-Injected abrasion guard in high wear areas for improved durability\r\n-Composite toecap fully covered with longer-lasting rubber guard\r\n-Lightweight Parabolic Stainless Steel (LPS2) runner ? fast and flexible'),(20,'Graf Ultra G3S','Graf',4,779.99,'-EPP quarter construction\r\n-Thermolam construction system\r\n-AMC composite lining\r\n-Anatomical ankle padding'),(21,'RBK 20k Pump','Reebok',4,959.99,'The Reebok K skate series was designed to maximize power transmission, while \r\nproviding a customized fit that enhances the skaters stride. The Dynamic Support \r\nSystem, or DSS, is evident throughout the series and refers to the ideal skate \r\nconstruction which optimizes flexibility and stiffness. Reebok uses its proprietary \r\nlast, designed and engineered to provide stride-enhancing fit accuracy.\r\n\r\nThe 20K has a generous fit profile, but is armed with the Pump® technology which allows \r\nair to circulate around the ankles for great heel lock and a customized fit. The 20K is \r\nalso equipped with the Skate Lock feature which allows for variability within two \r\nseparate lacing zones - lower and upper.\r\n\r\nReebok uses its Pro Armour? V construction in the quarter. This is a fused multi-layered \r\npackage reinforced with metal mesh for great durability, long term performance and great \r\nstiffness. The other elements of the quarter are part of the Dynamic Support System. The \r\nflexion zone is an embedded composite zone, the fibers of which are oriented at 45° which \r\nprovides excellent support, but at the same time helps to control forward flex. The spinal\r\nzone is also an embedded composite zone, however, these fibers are oriented at 90° and 0° \r\nwhich is a stronger orientation for locking the heel in place.\r\n\r\nHeel lock is further enhanced through the use of internal memory foams and the Pump®. The \r\nfact that these are heat moldable just adds to the already great amount of customization \r\nthat this skate offers, which of course is going to further reduce break in time. The \r\ntongue is made of pro felt with a lightweight EPE foam core and an external molded lace \r\nbite protector, giving you a good amount of forward flex. Meanwhile, on the opposite end \r\nof the skating stride, the 20K tendon guard has rubber inserts that allow it to flex which\r\nenhances your mobility.\r\n\r\nThe 20K has the ventilated full lightweight carbon composite outsole which helps reduce \r\nweight in a skate and provides excellent rigidity under the boot thus reducing torque \r\nwhile you skate. If you\'re looking for a skate with great customized fit features and \r\ntechnology that helps to maximize power transmission, look no further than the 20K from \r\nReebok.'),(22,'Bauer Vapor APX Griptac','Bauer',5,249.99,'\r\nWith the creation of the VAPOR X:60, BAUER introduced to the hockey world the smartest\r\nstick in the history of the game. The X:60 features revolutionary Intelli-Sense Shot\r\nTechnology, which utilizes a dual kick zone flex profile for a quick release on wrist\r\nand snap shots while producing powerful slap shots and one-timers. \r\n\r\nWith feedback taken from elite level players around the globe, Bauer has developed a new\r\nblade core that offers the soft feel of SUPREME TOTALONE\'s Power Core 3 and combines it\r\nwith the lightweight performance of VAPOR\'s Aero Foam II Core to create the ultimate\r\nbalance of light weight and feel for the puck. As a testament to its uniqueness, the\r\nblade core was awarded a patent by both the U.S. and Canadian governments. \r\n\r\nPackaged together with the Pure Shot Blade Profile for improved accuracy, Monocomp \r\nTechnology for optimized balance and feel, and our exclusive use of light weight \r\nTeXtreme® carbon fiber, the new blade core will help you score more goals from anywhere \r\non the ice... and make you \"feel\" really good about it. \r\n\r\n-Patented dual density blade core\r\n-Intelli-Sense shot technology\r\n-Monocomp Technology\r\n-Pure Shot blade profile\r\n-Lightweight TeXtreme® construction\r\n-Micro Feel II shaft dimension\r\n-Double concave walls\r\n-VAPOR premium dual taper\r\nGRIPTAC grip application\r\nLength - 60in.'),(23,'Easton Stealth RS II Grip','Easton',5,309.99,'If you love to shoot, the Stealth line of sticks offer features and technologies at every\r\nprice point to help enhance shot accuracy. Drawing on the successes of the original RS,\r\nEaston set out to make this version of the RS even better. This RS II is lighter, very\r\nwell-balanced, more responsive and more durable than its predecessor and it looks great\r\ntoo. Easton kept with the tonal black look on the RS II and the entire Stealth line uses\r\nasymmetrical graphics where the forehand graphic is different than the backhand graphic\r\non the stick.\r\n\r\nVibration dampening and improved durability from impacts such as slashes and shots are two\r\nof the benefits of the proprietary textured woven Kevlar® wrap on the outermost layer of\r\nthe shaft. The shaft geometry, which is very popular among Easton\'s pro players, has\r\nsquare corners with slightly concave walls and allows you to get a nice secure grip of\r\nyour stick. The RS II is available in either a matte or Grip finish.\r\n\r\nIt\'s the elliptical profile of the taper that sets the RS II apart from all other sticks.\r\nThis is a finely tuned taper section that provides torsional resistance towards the blade\r\nopening up, which is a contributing factor to shot accuracy. This area also provides the\r\nlow kick point of the stick, which allows for a quick release.\r\n\r\nThe blade was a main area of focus for Easton as they re-engineered it to be more\r\nresponsive and durable. Easton uses a multi-rib construction through its micro-bladder\r\nprocess. Basically, this means that there are 3 structural ribs that run the length of\r\nthe blade with a combination of a bladder and 4 foam core inserts between the ribs. This\r\nprovides a more durable core to wall bond as well as dampening for a great feel.\r\n\r\nGo ahead, take the shot. The Easton Stealth RS II has been engineered to enhance shot\r\naccuracy.'),(24,'Easton Synergy SY50','Easton',5,36.99,'-Aircraft veneer construction\r\n-Aspen core\r\n-5mm corner radius\r\n-Senior (95 Flex) 58\"'),(25,'RBK 20k Sickkick 4 Fullgrip','Reebok',5,299.99,'The Sickick IV series of sticks from Reebok is all about control, accuracy and a quick\r\nrelease. The quick release comes from the low-mid kick point, while control and accuracy\r\ncome from their Dual Matrix II? and AccuBlade? technologies which are also found\r\nthroughout the line. The 20K is lightweight, has great balance and a nice clean look.\r\n\r\nReebok uses its Pure Fiber?, or true one-piece construction of high modulus carbon\r\nfiber for this stick. This type of construction eliminates overlapping material in the\r\nlower shaft, thus reducing weight and improving consistency in energy transfer.\r\nThe senior shaft uses a Traditional geometry with square-corners.\r\n\r\nThe 20K is available with Reebok\'s Dual Grip Technology where the bottom half of the\r\nshaft has a matte finish and allows your hand to move up and down freely and quickly,\r\nwhile the top half of the shaft has a Griptonite finish that helps lock the stick in\r\nplace when you clamp down. If you prefer the full Griptonite finish on the shaft, the\r\nsenior model of the 20K is also available in full grip.\r\n\r\nThe gradual taper of the stick is what gives it the low-mid kick point.\r\n\r\nDual Matrix II? is technology that is used in the shaft and the blade of the stick, but is\r\nclearly visible in the blade. The orientation of the fibers on the forehand are at 0°\r\nand 90°, while the fibers on the backhand are at 45°. This variation causes tension\r\nduring loading that creates a slingshot effect on release. The other technology that\r\nReebok uses is called AccuBlade?. This is a foam core blade that gets stiffer as you go\r\nfrom heel to toe resulting in less torqueing to keep the blade face closed.\r\n\r\nWith technologies that enhance control, accuracy and release time, the Reebok 20K is all\r\nthat, and more...'),(26,'Sherwood Endure 180','Sherwood',5,29.99,'-Compression molded shaft for a consistent flex profile and greater responsivess\r\n-Multi-layer birch lamination Responsive stick\r\n-Ensures great control and superior sliding on all surfaces\r\n-Prevents premature wearing and increases blade stiffness\r\n-ABS construction');

/\*!40000 ALTER TABLE `merchandise` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `orderitems`

--

DROP TABLE IF EXISTS `orderitems`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `orderitems` (

`orderid` int(10) unsigned NOT NULL,

`itemid` int(10) NOT NULL,

`specid` char(6) NOT NULL default '000000',

`itemprice` float(6,2) NOT NULL,

`quantity` tinyint(4) default '1',

PRIMARY KEY (`orderid`,`itemid`),

KEY `itemid` (`itemid`),

CONSTRAINT `orderitems\_ibfk\_1` FOREIGN KEY (`orderid`) REFERENCES `orders` (`orderid`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `orderitems\_ibfk\_2` FOREIGN KEY (`itemid`) REFERENCES `merchandise` (`itemid`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `orderitems`

--

LOCK TABLES `orderitems` WRITE;

/\*!40000 ALTER TABLE `orderitems` DISABLE KEYS \*/;

INSERT INTO `orderitems` VALUES (12,1,'bl13',149.99,1),(12,10,'blm',159.99,1),(12,26,'Rya100',29.99,2),(13,11,'roysm',19.99,1),(13,19,'7d',449.99,1);

/\*!40000 ALTER TABLE `orderitems` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `orders`

--

DROP TABLE IF EXISTS `orders`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `orders` (

`orderid` int(10) unsigned NOT NULL auto\_increment,

`customerid` int(10) unsigned NOT NULL,

`totalcost` float(6,2) default NULL,

`orderdate` date default NULL,

`orderstatus` char(15) default NULL,

`shipName` char(60) default NULL,

`shipAddress` char(80) default NULL,

`shipCity` char(30) default NULL,

`shipState` char(20) default NULL,

`shipZip` char(10) default NULL,

PRIMARY KEY (`orderid`)

) ENGINE=InnoDB AUTO\_INCREMENT=14 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `orders`

--

LOCK TABLES `orders` WRITE;

/\*!40000 ALTER TABLE `orders` DISABLE KEYS \*/;

INSERT INTO `orders` VALUES (12,1,369.96,'2012-12-10','Shipped','Frank Smith','43 Paramus Road','Paramus','NJ','07452'),(13,1,469.98,'2012-12-10','Preparing','Frank Smith','43 Paramus Road','Paramus','NJ','07452');

/\*!40000 ALTER TABLE `orders` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `reviews`

--

DROP TABLE IF EXISTS `reviews`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `reviews` (

`itemid` int(10) NOT NULL,

`customerid` int(10) unsigned NOT NULL,

`score` int(10) unsigned NOT NULL,

`blurb` text,

KEY `itemid` (`itemid`,`customerid`),

KEY `customerid` (`customerid`),

CONSTRAINT `reviews\_ibfk\_2` FOREIGN KEY (`customerid`) REFERENCES `customer` (`customerid`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `reviews\_ibfk\_1` FOREIGN KEY (`itemid`) REFERENCES `merchandise` (`itemid`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `reviews`

--

LOCK TABLES `reviews` WRITE;

/\*!40000 ALTER TABLE `reviews` DISABLE KEYS \*/;

INSERT INTO `reviews` VALUES (1,1,4,'These gloves are great. I love the way they feel. They are extremely durable as well.');

/\*!40000 ALTER TABLE `reviews` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `skates`

--

DROP TABLE IF EXISTS `skates`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `skates` (

`exId` char(6) NOT NULL,

`itemid` int(10) NOT NULL,

`bootsize` tinyint(3) unsigned NOT NULL,

`quantity` int(11) NOT NULL,

KEY `itemid` (`itemid`),

CONSTRAINT `skates\_ibfk\_1` FOREIGN KEY (`itemid`) REFERENCES `merchandise` (`itemid`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `skates`

--

LOCK TABLES `skates` WRITE;

/\*!40000 ALTER TABLE `skates` DISABLE KEYS \*/;

INSERT INTO `skates` VALUES ('13d',17,13,14),('12d',17,12,11),('7d',17,7,4),('8d',17,8,7),('6d',18,6,15),('7d',18,7,6),('9d',18,9,9),('10d',18,10,5),('10d',19,10,17),('11d',19,11,8),('7d',19,7,4),('9d',19,9,7),('7d',20,7,4),('8d',20,8,9),('12d',20,12,15),('11d',20,11,7),('7d',21,7,6),('9d',21,9,9),('11d',21,11,7),('12d',21,12,6);

/\*!40000 ALTER TABLE `skates` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `sticks`

--

DROP TABLE IF EXISTS `sticks`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `sticks` (

`exId` char(6) NOT NULL,

`itemid` int(10) NOT NULL,

`curve` char(10) NOT NULL,

`flex` tinyint(3) unsigned NOT NULL,

`quantity` int(11) NOT NULL,

KEY `itemid` (`itemid`),

CONSTRAINT `sticks\_ibfk\_1` FOREIGN KEY (`itemid`) REFERENCES `merchandise` (`itemid`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `sticks`

--

LOCK TABLES `sticks` WRITE;

/\*!40000 ALTER TABLE `sticks` DISABLE KEYS \*/;

INSERT INTO `sticks` VALUES ('Bac77',22,'Backstrom',77,18),('Bac102',22,'Backstrom',102,7),('Kan95',22,'Kane',95,4),('Ove87',22,'Ovechkin',87,5),('Ove102',22,'Ovechkin',102,8),('Mal95',22,'Malkin',95,15),('Cam75',23,'Cammalleri',75,7),('Cam100',23,'Cammalleri',100,9),('Sha110',23,'Shanahan',110,11),('Sak100',23,'Sakic',100,6),('Char95',24,'Chara',95,7),('Sak95',24,'Sakic',95,8),('Zet95',24,'Zetterberg',95,11),('Par95',24,'Parise',95,15),('Spez75',25,'Spezza',75,9),('Ham100',25,'Hamrlik',100,12),('Cros10',25,'Crosby',100,10),('Cros85',25,'Crosby',85,7),('Cof100',26,'Coffey',100,5),('Sta100',26,'Statsny',100,7),('Bou100',26,'Bouchard',100,2),('Rya100',26,'Ryan',100,1);

/\*!40000 ALTER TABLE `sticks` ENABLE KEYS \*/;

UNLOCK TABLES;

/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;

-- Dump completed on 2012-12-15 16:31:53

**Citations**

**Citations**:

**Inspired by:**

* "Hockey Equipment - Hockey Sticks, Hockey Skates, Ice Hockey Skates, Inline Hockey Skates and More Hockey Gear." *Hockey Equipment - Hockey Sticks, Hockey Skates, Ice Hockey Skates, Inline Hockey Skates and More Hockey Gear*. N.p., n.d. Web 21 Oct. 2012. <www.hockeygiant.com>
* "Hockey Equipment & Gear | Hockey Sticks & Skates | Ice Hockey Helmets & Gloves | HockeyMonkey.com." *Hockey Equipment & Gear | Hockey Sticks & Skates | Ice Hockey Helmets & Gloves | HockeyMonkey.com*. N.p., n.d. Web. 21 Oct. 2012.<www.hockeymonkey.com>
* "Total Hockey Stores | Hockey Sticks, Hockey Skates, Apparel & More." *Total Hockey Stores | Hockey Sticks, Hockey Skates, Apparel & More*. N.p., n.d. Web. 15 Dec. 2012. <www.totalhockey.com >

**Images from:**

All hockey equipment images used from:

* "Hockey Equipment - Hockey Sticks, Hockey Skates, Ice Hockey Skates, Inline Hockey Skates and More Hockey Gear." *Hockey Equipment - Hockey Sticks, Hockey Skates, Ice Hockey Skates, Inline Hockey Skates and More Hockey Gear*. N.p., n.d. Web 21 Oct. 2012. <www.hockeygiant.com>

Header Image:

* *Pond\_hockey.jpg*. N.d. Photograph. *The Fremd High School English Ning*. By Russ Anderson. 2012. Web. 2 Dec. 2012. <http://api.ning.com/files/f\*OHYOdIjBy3xykmHY7UNkvqxVh1jEAHUnroneJ\*3byEq\*qI0twXSTIW2YdBFMN3dfvU2yNbmBe7G1uz21yDuXlYsypOosKX/pond\_hockey.jpg>.

**Code Citing:**

* Welling, Luke, and Laura Thomson. PHP and MySQL Web Development, Fourth Edition. Upper Saddle River [etc.: Addison-Wesley, 2009. Print.
* "Storing Objects in Php Session." Stackoverflow. N.p., 18 Sept. 2009. Web. 5 Dec. 2012. <http://stackoverflow.com/questions/1442177/storing-objects-in-php-session>.
* "PHP Redirect To Another URL â Page Script Redirect â Redirect Web Page." Frequently Asked Questions About Linux UNIX RSS. N.p., n.d. Web. 29 Nov. 2012. <http://www.cyberciti.biz/faq/php-redirect/>.
* "PHP Manual." PHP:. N.p., Nov.-Dec. 2012. Web. Nov.-Dec. 2012. <http://www.php.net/manual/en/>.