Name: Madison Miatke

Deliverable 3

Accessing My Application: I placed my web application and database on the NJIT servers so that it could be accessed through the internet. To access my Deliverable 3 application go the following link: https://web.njit.edu/~mdm56/cs331/deliverable3/index.html

Goals:

The Project's Goal: The goal of this part of the project was to create a MySQL database and a query application that would use SQL code to request the information from the database I created. By doing this I would gain a better understanding of the full cycle of designing and building a database.

My Goal: I wanted to take the deliverable three a step further and use my web development skills to build out an entire interface by hand that would utilize these queries and a connection to the database to allow user interaction with the requirements described.

Challenges:

There were a few challenges I faced during the creation of my application.

1. Creating the database took a few attempts because I realized the order you create foreign keys and tables is important. Once I realized that I made a plan of the order I needed to create tables.
2. Filling the database manually was pretty tedious and took a while because I needed to create instances of the queries you requested. For example there needed to be photographers that only took portrait pictures.
3. There were queries that were worded ambiguous, one of the worst being number 7. It could be interpreted two ways. (List models who modeled in all photos taken by photographer Y)
    a. You could read this is all the photos the models were in were taken by the photographer.
    b. Or you could read it as all photos the photographer took the models were in

    These are different because I wondered if a model could be in other photographer's pictures or can the photographer take pictures of other models.
4. Filling fake information was tedious but became easier when I discovered random names, address, and birthday generators.

Revisions To Deliverable 2:

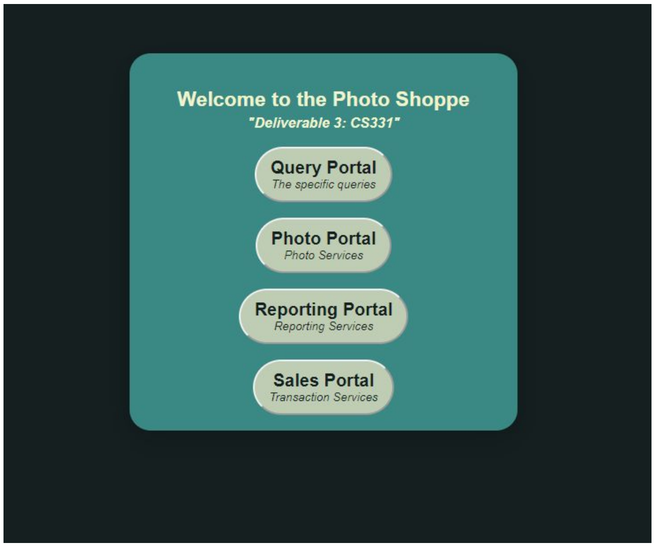
I used the solution to deliverable 2 given to the class with some minor changes. To allow for easier query creation, I added attributes to eliminate the need for multivalued Foriegn Keys and Primary Keys.

1. In the MODEL table I made a multi-attribute constraint that made MName MBDate unique and then added a new primary key MID so that attribute could be referenced with a foriegn key in MODELS.
2. I did something identical to Photographer adding a new primary key PID and making a multi-attribute constraint that requires PName and PBDate to be unique.
3. Lastly I did the identical adjustment to LOCATION, by adding the primary key LOCID that would be referenced in Landscape rather than LocationPlace and LocationCountry.

The Project:

Description/Execution: I decided I would build my web application with PHP, HTML, and a little CSS and JavaScript while utilizing NJIT's phpMyAdmin to build the MySQL database.

I created a plan to make a four part interface:



Query Portal: A webpage that could execute the 14 specific queries and show entire tables.

Photo Portal: A webpage that allowed adding a new photo, deleting an existing photo, changing photo information, and querying photo information

Reporting Portal: A web page that could generate sales reports including some found in the query portal and some that I would think of on my own.

Sales Portal: A webpage that could process a sale, view a sale (given transactionID), and find sales (given different features e.g. date, customer name etc.).

Each web page would allow for the user to interact with the inputs to the queries I was required to write for the deliverable

SQL Tables:

Abstract:

Create Table	CREATE TABLE IF NOT EXISTS `Abstract` (
	  `PhotoID` int(11) NOT NULL,
	  `Comment` varchar(255) NOT NULL
	);

Add Constraints To Table	ALTER TABLE `Abstract`
	 ADD PRIMARY KEY (`PhotoID`);
	
	ALTER TABLE `Abstract`
	 ADD CONSTRAINT `Abstract_ibfk_1` FOREIGN KEY (`PhotoID’) REFERENCES `Photo` (`PhotoID`) ON DELETE CASCADE ON UPDATE CASCADE;

Populate Table	INSERT INTO `Abstract` (`PhotoID`, `Comment`) VALUES
	(3, 'An orange.'),
	[…],
	(23, 'A cup of coffee.');

Customer:

Create Table	CREATE TABLE IF NOT EXISTS `Customer` (
	  `LoginName` varchar(255) NOT NULL,
	  `Password` varchar(255) NOT NULL,
	  `CName` varchar(255) NOT NULL,
	  `CType` varchar(255) NOT NULL,
	  `BillingAddress` varchar(255) NOT NULL,
	  `Str1` varchar(255) NOT NULL,
	  `Str2` varchar(255) NOT NULL,
	  `City` varchar(255) NOT NULL,
	  `State` varchar(2) NOT NULL,
	  `Zip` varchar(5) NOT NULL
	);

Add Constraints To Table	ALTER TABLE `Customer` ADD PRIMARY KEY (`LoginName`);

Populate Table	INSERT INTO `Customer` (`LoginName`, `Password`, `CName`, `CType`, `BillingAddress`, `Str1`, `Str2`, `City`, `State`, `Zip`) VALUES
	('aswift31', 'as121', 'Alysha Swift', 'individual', '29 Carpenter St.\r\nSumter, SC 29150', '29 Carpenter St.', '', 'Sumter', 'SC', '29150'),
	[...],
	('bhunt23', 'bh816', 'Boyd Hunt', 'individual', '60 Sierra St.\r\nMuskogee, OK 74403', '60 Sierra St.', '', 'Muskogee', 'OK', '74403');

Influences:

Create Table	CREATE TABLE IF NOT EXISTS `Influences` (
	  `EPID` int(11) NOT NULL,
	  `RPID` int(11) NOT NULL
	);

Add Constraints To Table	ALTER TABLE `Influences` ADD PRIMARY KEY (`EPID`,`RPID`), ADD KEY `Influences_ibfk_2` (`RPID`); ALTER TABLE `Influences` ADD CONSTRAINT `Influences_ibfk_1` FOREIGN KEY (`EPID`) REFERENCES `Photographer` (`PID`) ON DELETE CASCADE ON UPDATE CASCADE, ADD CONSTRAINT `Influences_ibfk_2` FOREIGN KEY (`RPID`) REFERENCES `Photographer` (`PID`) ON DELETE CASCADE ON UPDATE CASCADE;

Populate Table	INSERT INTO `Influences` (`EPID`, `RPID`) VALUES
	(6, 1),
	[...],
	(3, 2);

Landscape:

Create Table	CREATE TABLE IF NOT EXISTS `Landscape` (
	  `PhotoID` int(11) NOT NULL,
	  `LOCID` int(11) NOT NULL
	);

Add Constraints To Table	ALTER TABLE `Landscape` ADD PRIMARY KEY (`PhotoID`), ADD KEY `LOCID` (`LOCID`); ALTER TABLE `Landscape` ADD CONSTRAINT `Landscape_ibfk_1` FOREIGN KEY (`PhotoID`) REFERENCES `Photo` (`PhotoID`) ON DELETE CASCADE ON UPDATE CASCADE, ADD CONSTRAINT `Landscape_ibfk_2` FOREIGN KEY (`LOCID`) REFERENCES `Location` (`LOCID`) ON DELETE CASCADE ON UPDATE CASCADE;

Populate Table	INSERT INTO `Landscape` (`PhotoID`, `LOCID`) VALUES
	(2, 1),
	[...],
	(16, 1);

Location:

Create Table	CREATE TABLE IF NOT EXISTS `Location` (
	  `LOCID` int(11) NOT NULL,
	  `Place` varchar(255) NOT NULL,
	  `Country` varchar(255) NOT NULL,
	  `Description` varchar(255) NOT NULL
	);

Add Constraints To Table	ALTER TABLE `Location` ADD PRIMARY KEY (`LOCID`), ADD UNIQUE KEY `Place_2` (`Place`,`Country`);
Populate Table	INSERT INTO `Location` (`LOCID`, `Place`, `Country`, `Description`) VALUES (1, 'Chicago', 'United States', 'The Windy City'),

	[...], (2, 'New York City', 'United States', 'Big apple');

Model:

Create Table	CREATE TABLE IF NOT EXISTS `Model` (
	  `MID` int(11) NOT NULL,
	  `MName` varchar(255) NOT NULL,
	  `MBDate` date NOT NULL,
	  `MBio` varchar(255) NOT NULL,
	  `MSex` varchar(255) NOT NULL
	);

Add Constraints To Table	ALTER TABLE `Model` ADD PRIMARY KEY (`MID`), ADD UNIQUE KEY `MName_2` (`MName`,`MBDate`);

Populate Table	INSERT INTO `Model` (`MID`, `MName`, `MBDate`, `MBio`, `MSex`) VALUES
	(1, 'Safiyyah Rigby', '1997-07-29', 'Specializes in solo portraits.', 'Female'),
	[...],
	(2, 'Rafferty Garrison', '1992-05-29', '', 'Male');

Models:

Create Table	CREATE TABLE IF NOT EXISTS `Models` (
	  `PhotoID` int(11) NOT NULL,
	  `MID` int(11) NOT NULL,
	  `Agency` varchar(255) NOT NULL
	);

Add Constraints To Table	ALTER TABLE `Models` ADD PRIMARY KEY (`PhotoID`,`MID`), ADD KEY `Models_ibfk_2` (`MID`); ALTER TABLE `Models` ADD CONSTRAINT `Models_ibfk_1` FOREIGN KEY (`PhotoID`) REFERENCES `Portrait` (`PhotoID`) ON DELETE CASCADE ON UPDATE CASCADE, ADD CONSTRAINT `Models_ibfk_2` FOREIGN KEY (`MID`) REFERENCES `Model` (`MID`) ON DELETE CASCADE ON UPDATE CASCADE;

Populate Table	INSERT INTO `Models` (`PhotoID`, `MID`, `Agency`) VALUES
	(1, 1, 'Model''s For Cheap'),
	[...],
	(1, 10, 'Model''s For Cheap');

Photo:

Create Table	CREATE TABLE IF NOT EXISTS `Photo` (
	  `PhotoID` int(11) NOT NULL,
	  `Speed` decimal(10,2) NOT NULL,
	  `Film` varchar(255) NOT NULL,
	  `F-Stop` decimal(10,2) NOT NULL,
	  `Color/B&W` varchar(255) NOT NULL,
	  `Resolution` varchar(255) NOT NULL,
	  `Price` decimal(10,2) NOT NULL,
	  `Date` date DEFAULT NULL,
	  `TransID` int(11) DEFAULT NULL,
	  `PID` int(11) NOT NULL
	);

Add Constraints To Table	ALTER TABLE `Photo` ADD PRIMARY KEY (`PhotoID`), ADD KEY `PID` (`PID`), ADD KEY `TransID` (`TransID`); ALTER TABLE `Photo` ADD CONSTRAINT `Photo_ibfk_1` FOREIGN KEY (`PID`) REFERENCES `Photographer` (`PID`) ON DELETE CASCADE ON UPDATE CASCADE, ADD CONSTRAINT `Photo_ibfk_2` FOREIGN KEY (`TransID`) REFERENCES `Transaction` (`TransID`) ON DELETE SET NULL ON UPDATE SET NULL;

Populate Table	INSERT INTO `Photo` (`PhotoID`, `Speed`, `Film`, `F-Stop`, `Color/B&W`, `Resolution`, `Price`, `Date`, `TransID`, `PID`) VALUES
	(1, 0.25, 'CineStill', 3.00, 'Color', '1025x768', 20.00, '2020-03-13', 1, 1),
	[...]
	(2, 0.10, 'Fomapan', 4.00, 'B&W', '800x600', 30.00, '2020-03-14', 1, 2);

Photographer:

Create Table	CREATE TABLE IF NOT EXISTS `Photographer` (
	  `PID` int(11) NOT NULL,
	  `PName` varchar(255) NOT NULL,
	  `PBDate` date NOT NULL,
	  `PBio` varchar(255) NOT NULL,
	  `PAddress` varchar(255) NOT NULL,
	  `Color/B&W` varchar(255) NOT NULL,
	  `PNationality` varchar(255) NOT NULL
	);

Add Constraints To Table	ALTER TABLE `Photographer` ADD PRIMARY KEY (`PID`), ADD UNIQUE KEY `PName` (`PName`,`PBDate`);

Populate Table	INSERT INTO `Photographer` (`PID`, `PName`, `PBDate`, `PBio`, `PAddress`, `Color/B&W`, `PNationality`) VALUES
	(1, 'Adam Smith', '1992-12-12', 'A photographer from Newport, VA.', '8C East Dogwood Drive\r\nNewport News, VA 23601', 'Color', 'American'),
	[...],
	(2, 'Lauren Ericson', '1993-06-06', 'A photographer from South Haven, Missouri.', '222 Catherine Rd.\r\nSouthaven, MS 38671', 'B&W', 'American');

Portrait:

Create Table	CREATE TABLE IF NOT EXISTS `Portrait` (
	  `PhotoID` int(11) NOT NULL,
	  `Head` varchar(255) NOT NULL
	);

Add Constraints To Table	ALTER TABLE `Portrait`
	 ADD PRIMARY KEY (`PhotoID`);
	ALTER TABLE `Portrait`
	ADD CONSTRAINT `Portrait_ibfk_1` FOREIGN KEY (`PhotoID`) REFERENCES `Photo` (`PhotoID`) ON DELETE CASCADE ON UPDATE CASCADE;

Populate Table	INSERT INTO `Portrait` (`PhotoID`, `Head`) VALUES
	(1, 'Y'),
	[...],
	(4, 'Y');

Transaction:

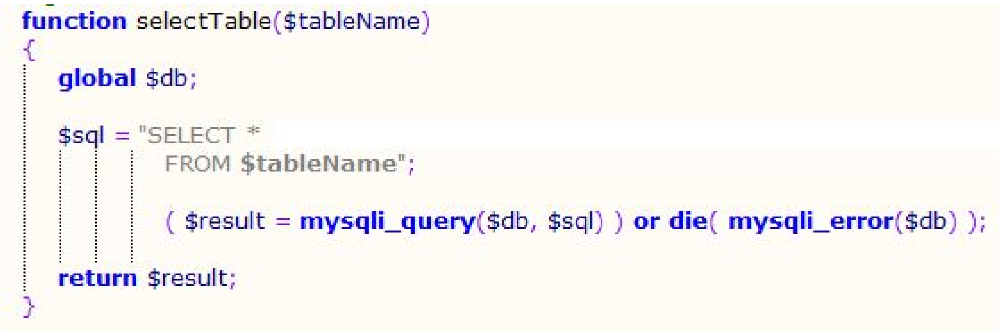
Create Table	CREATE TABLE IF NOT EXISTS `Transaction` (
	  `TransID` int(11) NOT NULL,
	  `TDate` date NOT NULL,
	  `CardNo` varchar(255) NOT NULL,
	  `CardType` varchar(255) NOT NULL,
	  `CardExpDate` varchar(10) NOT NULL,
	  `TotalAmount` decimal(10,2) NOT NULL,
	  `LoginName` varchar(255) NOT NULL
	);

Add Constraints To Table	ALTER TABLE `Transaction`
	 ADD PRIMARY KEY (`TransID`), ADD KEY `LoginName` (`LoginName`);
	ALTER TABLE `Transaction`
	ADD CONSTRAINT `Transaction_ibfk_1` FOREIGN KEY (`LoginName`) REFERENCES `Customer` (`LoginName`) ON DELETE CASCADE ON UPDATE CASCADE;

Populate Table	INSERT INTO `Transaction` (`TransID`, `TDate`, `CardNo`, `CardType`, `CardExpDate`, `TotalAmount`, `LoginName`) VALUES
	(1, '2020-04-02', '378506309523884', 'American Express', '2025-10', 100.00, 'cphelps21'),
	[...],
	(2, '2020-04-02', '370218812263903', 'American Express', '2021-02', 21.00, 'llees12');

SQL from Query Portal: Note that the queries are SQL but may include a php variable that contains the input from the web application.

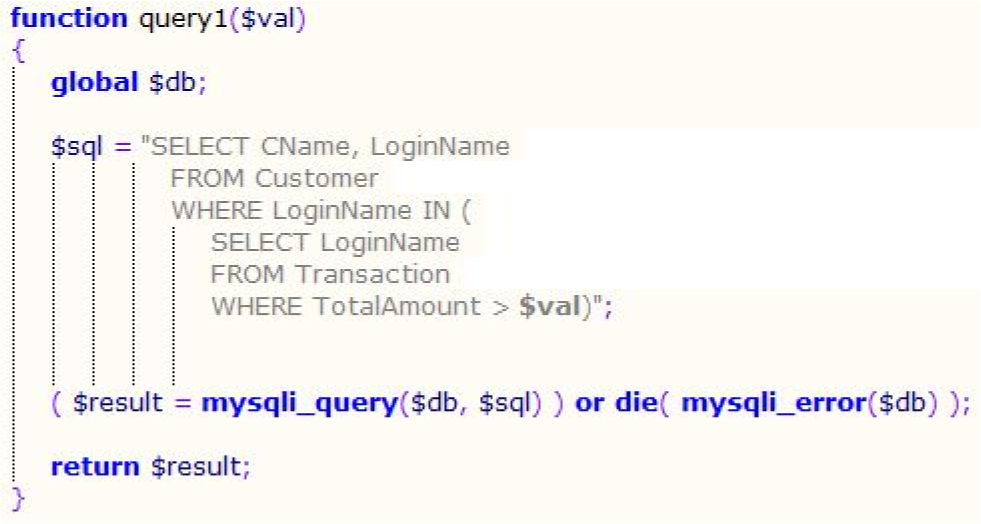
View Tables:



```
function selectTable($tableName)
{
    global $db;

    $sql = "SELECT *
            FROM $tableName";

            ( $result = mysqli_query($db, $sql) ) or die( mysqli_error($db) );

    return $result;
}
```

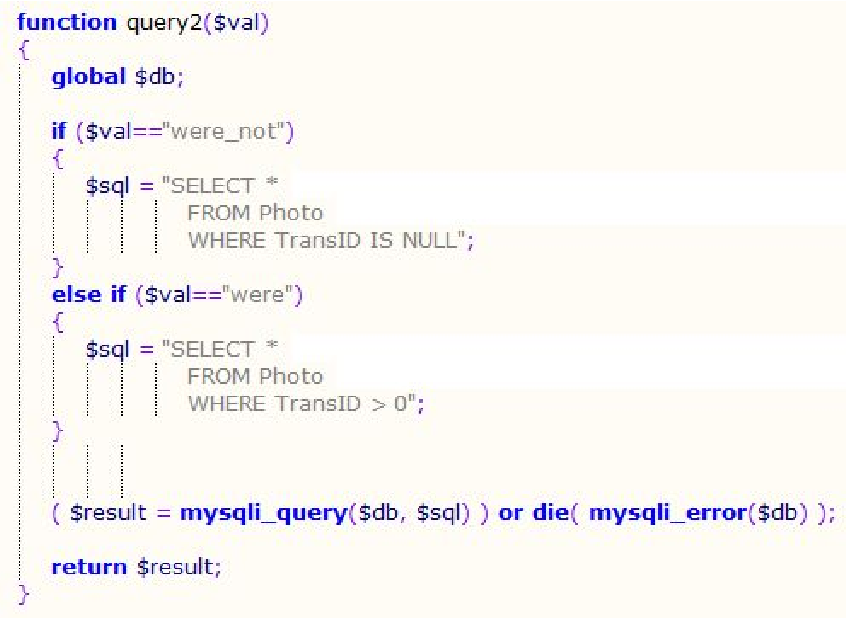
Specific Queries:

1. List customers who spent more than 100$ for the photos.



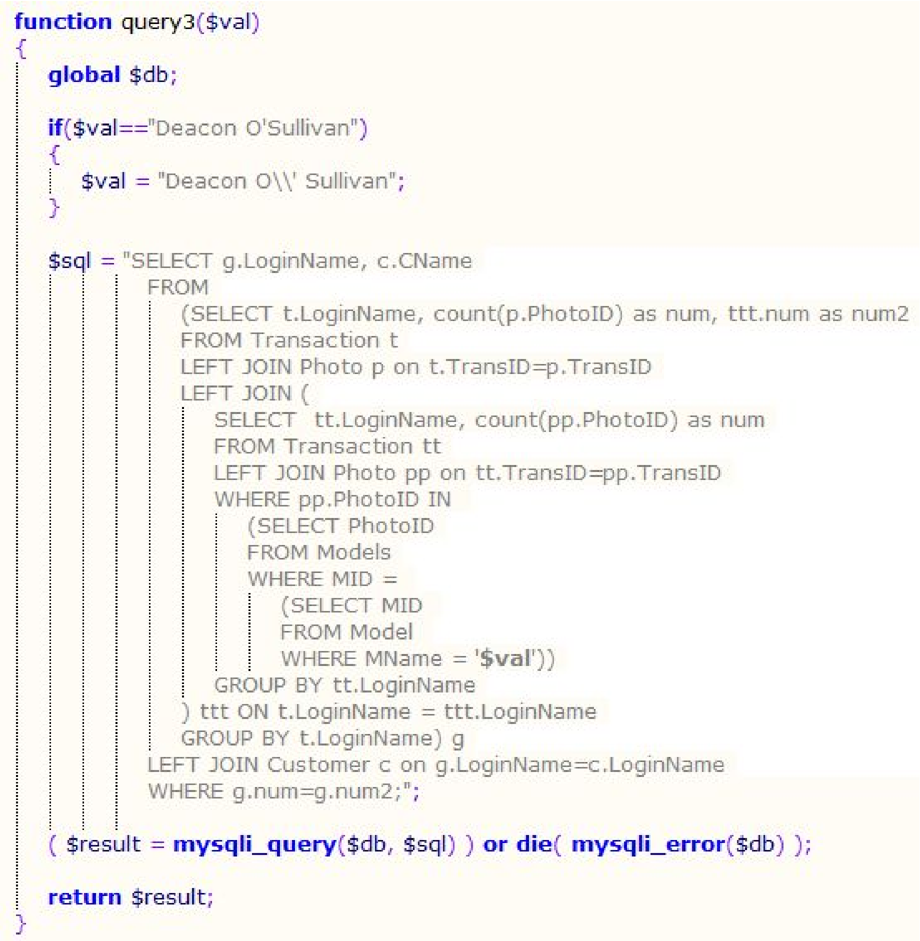
```
function query1($val)
{
    global $db;

    $sql = "SELECT CName, LoginName
            FROM Customer
            WHERE LoginName IN (
                SELECT LoginName
                FROM Transaction
                WHERE TotalAmount > $val)";

    ( $result = mysqli_query($db, $sql) ) or die( mysqli_error($db) );

    return $result;
}
```

2. List photos which were not bought.



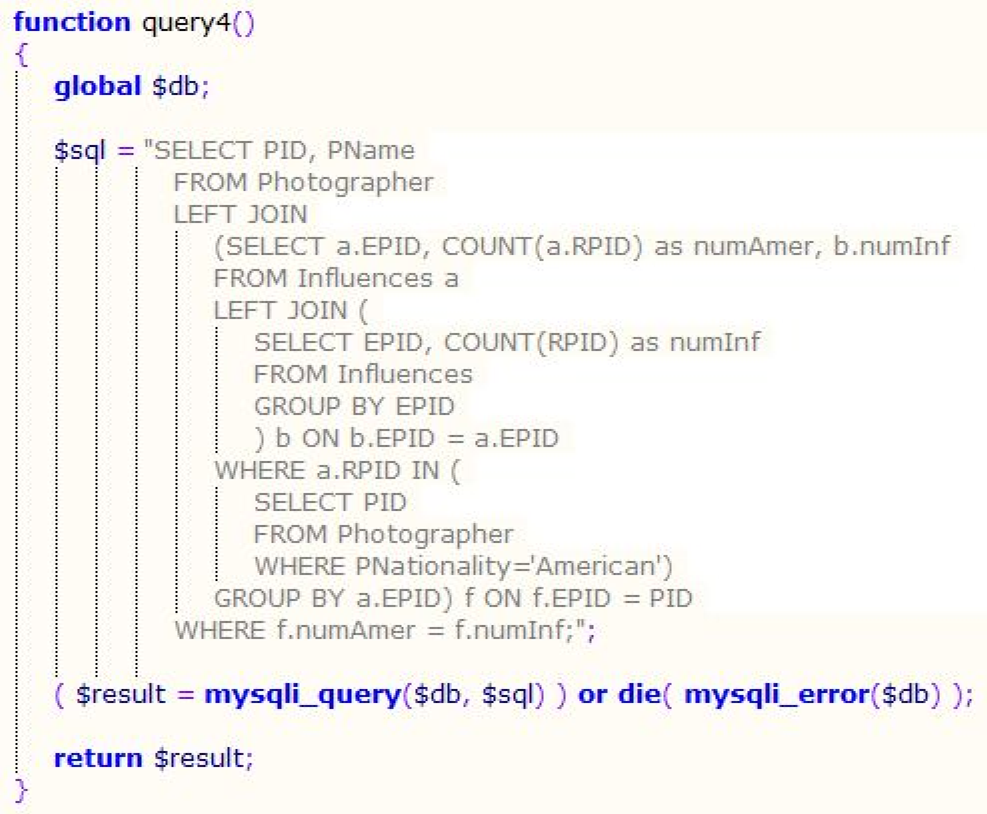
```
function query2($val)
{
    global $db;

    if ($val=="were_not")
    {
        $sql = "SELECT *
                FROM Photo
                WHERE TransID IS NULL";
    }
    else if ($val=="were")
    {
        $sql = "SELECT *
                FROM Photo
                WHERE TransID > 0";
    }

    ( $result = mysqli_query($db, $sql) ) or die( mysqli_error($db) );

    return $result;
}
```

3. List customers who bought all photos (portraits) in which a model X modeled.



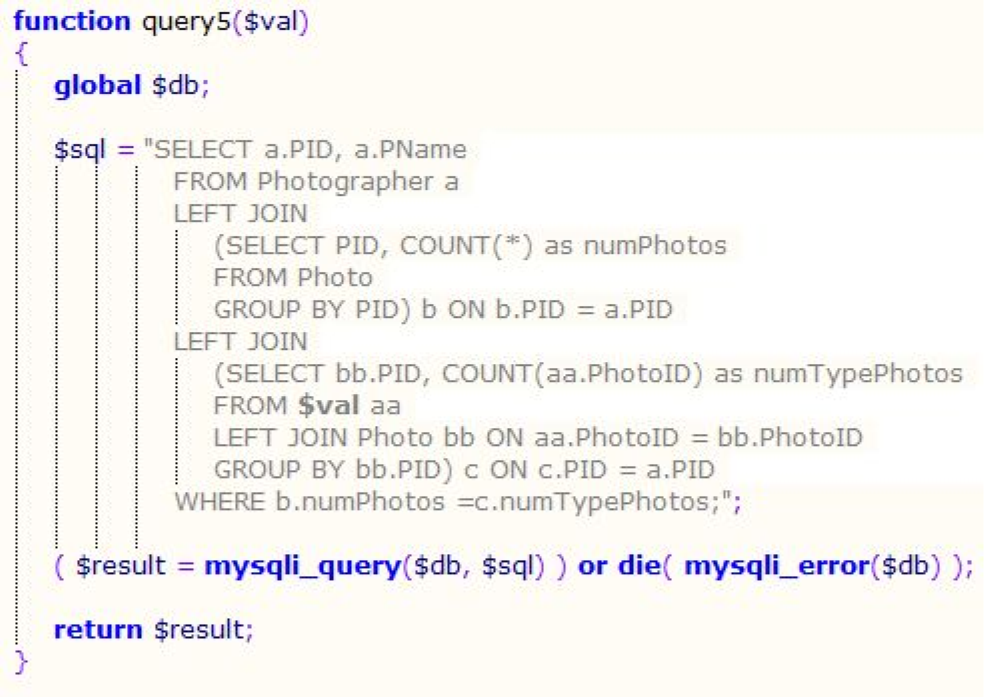
```
function query3($val)
{
    global $db;

    if($val=="Deacon O'Sullivan")
    {
        $val = "Deacon O\\' Sullivan";
    }

    $sql = "SELECT g.LoginName, c.CName
            FROM
                (SELECT t.LoginName, count(p.PhotoID) as num, ttt.num as num2
                FROM Transaction t
                LEFT JOIN Photo p on t.TransID=p.TransID
                LEFT JOIN (
                    SELECT  tt.LoginName, count(pp.PhotoID) as num
                    FROM Transaction tt
                    LEFT JOIN Photo pp on tt.TransID=pp.TransID
                    WHERE pp.PhotoID IN
                        (SELECT PhotoID
                        FROM Models
                        WHERE MID =
                            (SELECT MID
                            FROM Model
                            WHERE MName = '$val'))
                    GROUP BY tt.LoginName
                ) ttt ON t.LoginName = ttt.LoginName
                GROUP BY t.LoginName) g
            LEFT JOIN Customer c on g.LoginName=c.LoginName
            WHERE g.num=g.num2;";

    ( $result = mysqli_query($db, $sql) ) or die( mysqli_error($db) );

    return $result;
}
```

4. List photographers who influenced exclusively photographers who are US citizens.



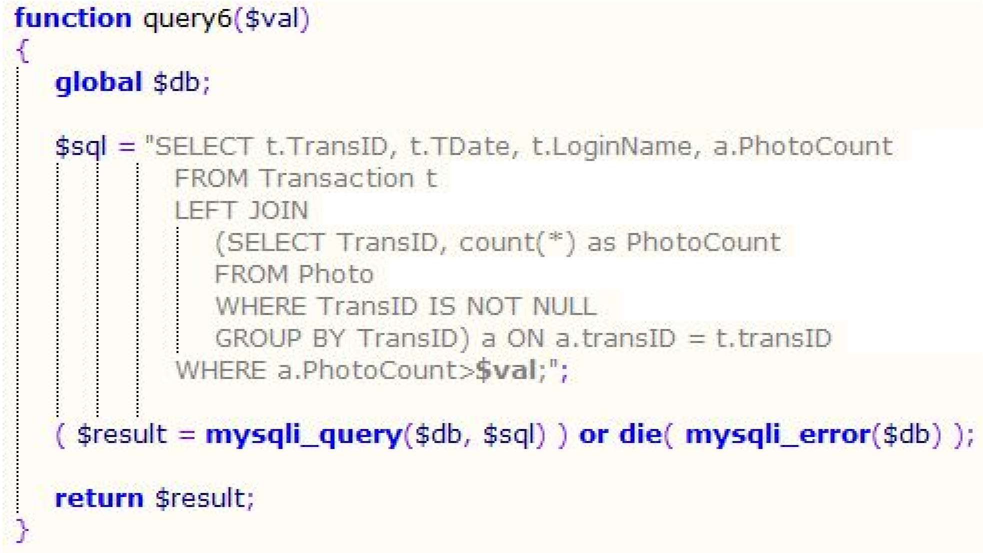
```
function query4()
{
    global $db;

    $sql = "SELECT PID, PName
            FROM Photographer
            LEFT JOIN
                (SELECT a.EPID, COUNT(a.RPID) as numAmer, b.numInf
                FROM Influences a
                LEFT JOIN (
                    SELECT EPID, COUNT(RPID) as numInf
                    FROM Influences
                    GROUP BY EPID
                    ) b ON b.EPID = a.EPID
                WHERE a.RPID IN (
                    SELECT PID
                    FROM Photographer
                    WHERE PNationality='American')
                GROUP BY a.EPID) f ON f.EPID = PID
            WHERE f.numAmer = f.numInf;";

    ( $result = mysqli_query($db, $sql) ) or die( mysqli_error($db) );

    return $result;
}
```

5. List photographers which took only portrait photos.



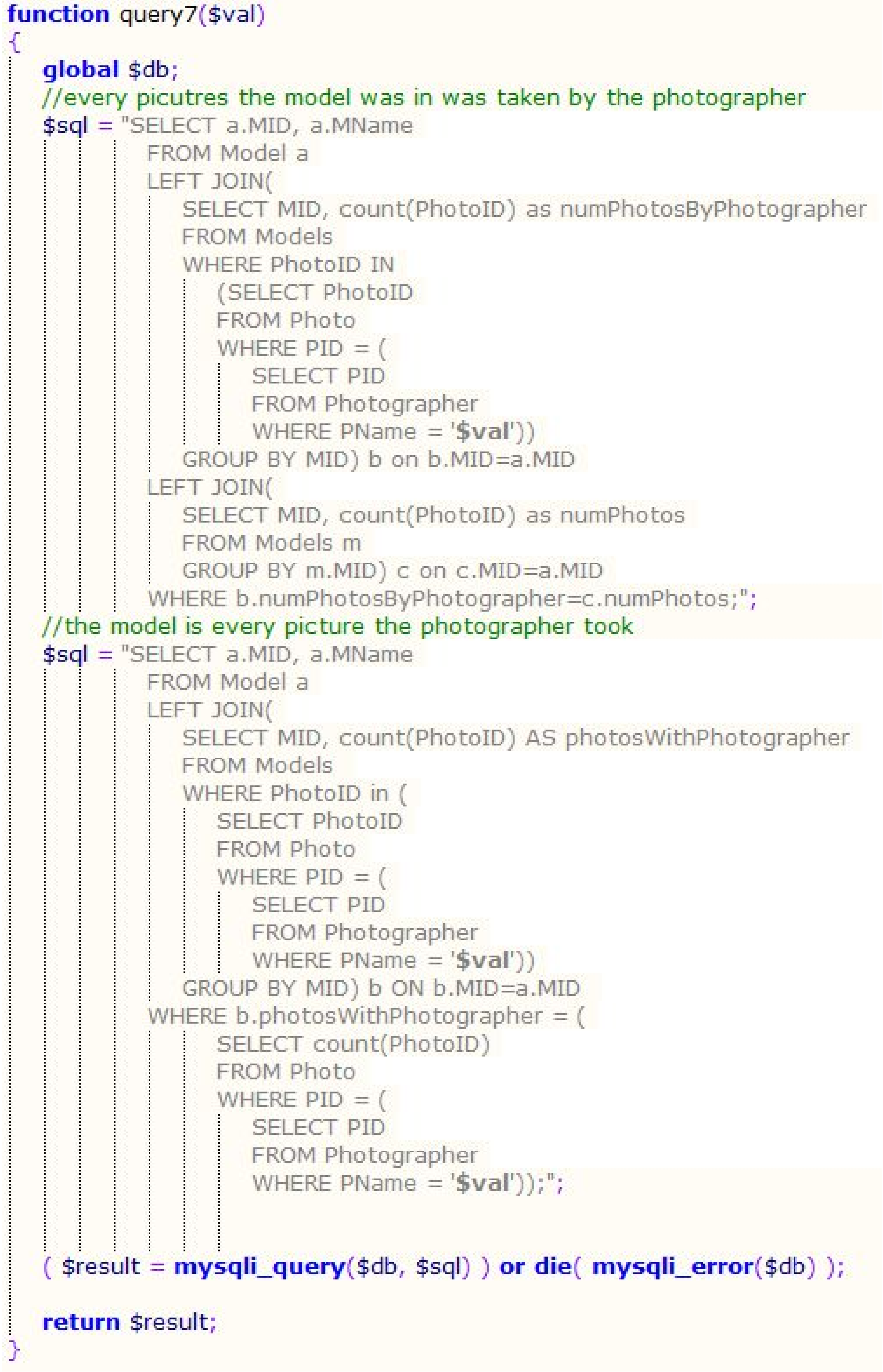
```
function query5($val)
{
    global $db;

    $sql = "SELECT a.PID, a.PName
            FROM Photographer a
            LEFT JOIN
                (SELECT PID, COUNT(*) as numPhotos
                FROM Photo
                GROUP BY PID) b ON b.PID = a.PID
            LEFT JOIN
                (SELECT bb.PID, COUNT(aa.PhotoID) as numTypePhotos
                FROM $val aa
                LEFT JOIN Photo bb ON aa.PhotoID = bb.PhotoID
                GROUP BY bb.PID) c ON c.PID = a.PID
            WHERE b.numPhotos =c.numTypePhotos;";

    ( $result = mysqli_query($db, $sql) ) or die( mysqli_error($db) );

    return $result;
}
```

6. List transactions (transID) which contain more than 3 photos.



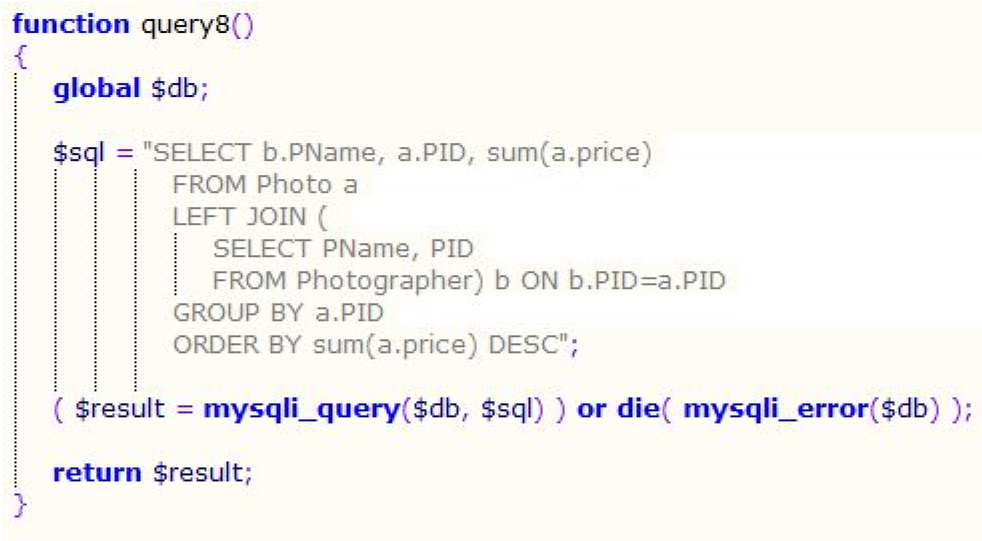
```
function query6($val)
{
   global $db;

   $sql = "SELECT t.TransID, t.TDate, t.LoginName, a.PhotoCount
            FROM Transaction t
            LEFT JOIN
               (SELECT TransID, count(*) as PhotoCount
               FROM Photo
               WHERE TransID IS NOT NULL
               GROUP BY TransID) a ON a.transID = t.transID
            WHERE a.PhotoCount>$val;";

   ( $result = mysqli_query($db, $sql) ) or die( mysqli_error($db) );

   return $result;
}
```

7. List models who modeled in all photos taken by photographer Y. (I wrote both interpretations)



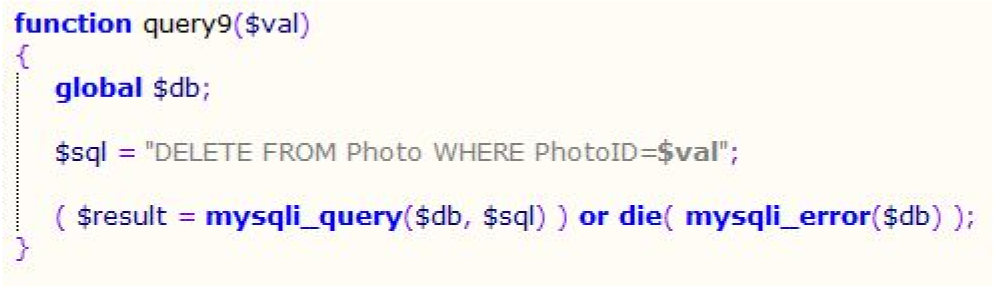
```
function query7($val)
{
    global $db;
    //every picutres the model was in was taken by the photographer
    $sql = "SELECT a.MID, a.MName
                FROM Model a
                LEFT JOIN(
                    SELECT MID, count(PhotoID) as numPhotosByPhotographer
                    FROM Models
                    WHERE PhotoID IN
                        (SELECT PhotoID
                        FROM Photo
                        WHERE PID = (
                            SELECT PID
                            FROM Photographer
                            WHERE PName = '$val'))
                    GROUP BY MID) b on b.MID=a.MID
                LEFT JOIN(
                    SELECT MID, count(PhotoID) as numPhotos
                    FROM Models m
                    GROUP BY m.MID) c on c.MID=a.MID
                WHERE b.numPhotosByPhotographer=c.numPhotos;";
    //the model is every picture the photographer took
    $sql = "SELECT a.MID, a.MName
                FROM Model a
                LEFT JOIN(
                    SELECT MID, count(PhotoID) AS photosWithPhotographer
                    FROM Models
                    WHERE PhotoID in (
                        SELECT PhotoID
                        FROM Photo
                        WHERE PID = (
                            SELECT PID
                            FROM Photographer
                            WHERE PName = '$val'))
                    GROUP BY MID) b ON b.MID=a.MID
                WHERE b.photosWithPhotographer = (
                        SELECT count(PhotoID)
                        FROM Photo
                        WHERE PID = (
                            SELECT PID
                            FROM Photographer
                            WHERE PName = '$val'));";


    ( $result = mysqli_query($db, $sql) ) or die( mysqli_error($db) );

    return $result;
}
```

8. Rank the photographers by the total cost (sum of prices) of the photos they took.



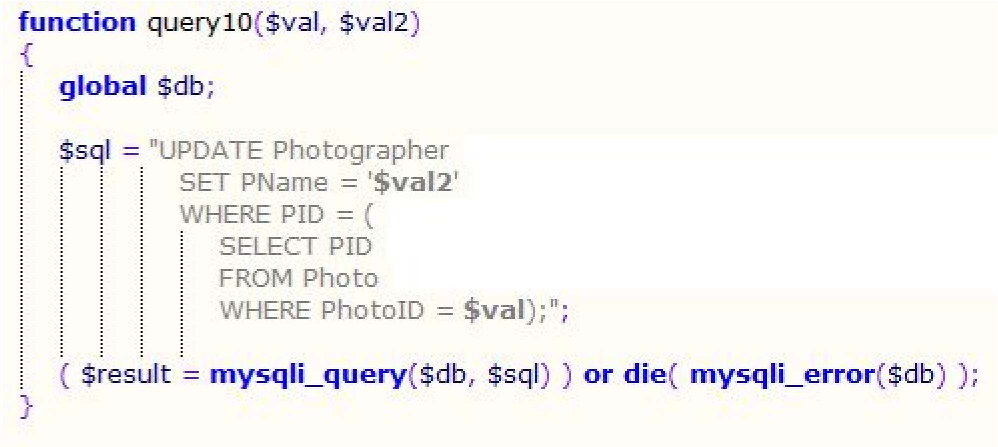
```
function query8()
{
    global $db;

    $sql = "SELECT b.PName, a.PID, sum(a.price)
            FROM Photo a
            LEFT JOIN (
                SELECT PName, PID
                FROM Photographer) b ON b.PID=a.PID
            GROUP BY a.PID
            ORDER BY sum(a.price) DESC";

    ( $result = mysqli_query($db, $sql) ) or die( mysqli_error($db) );

    return $result;
}
```

9. Delete from relation Photo the photo with photoID=X.



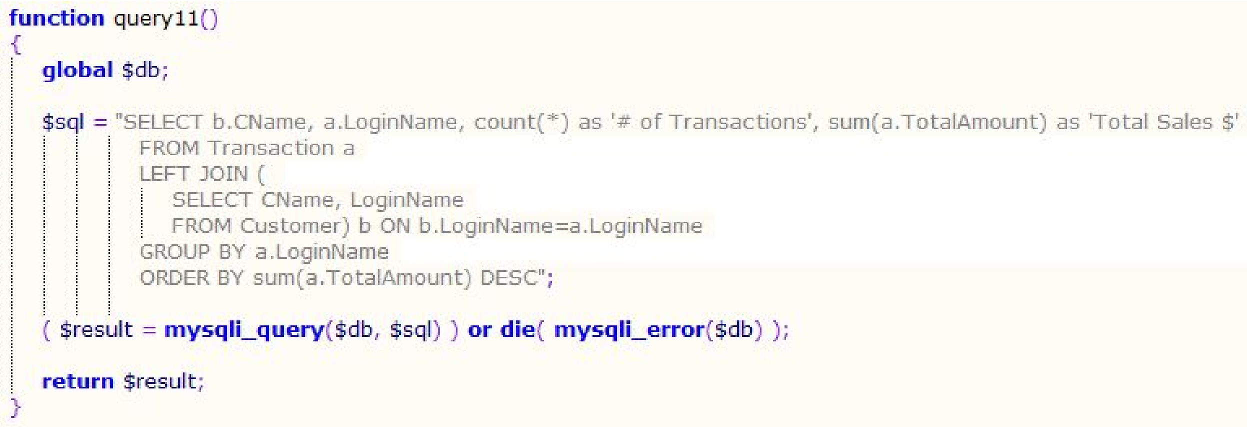
```
function query9($val)
{
    global $db;

    $sql = "DELETE FROM Photo WHERE PhotoID=$val";

    ( $result = mysqli_query($db, $sql) ) or die( mysqli_error($db) );
}
```

10. Update the photographer name of the photo with photoID=X to Y.



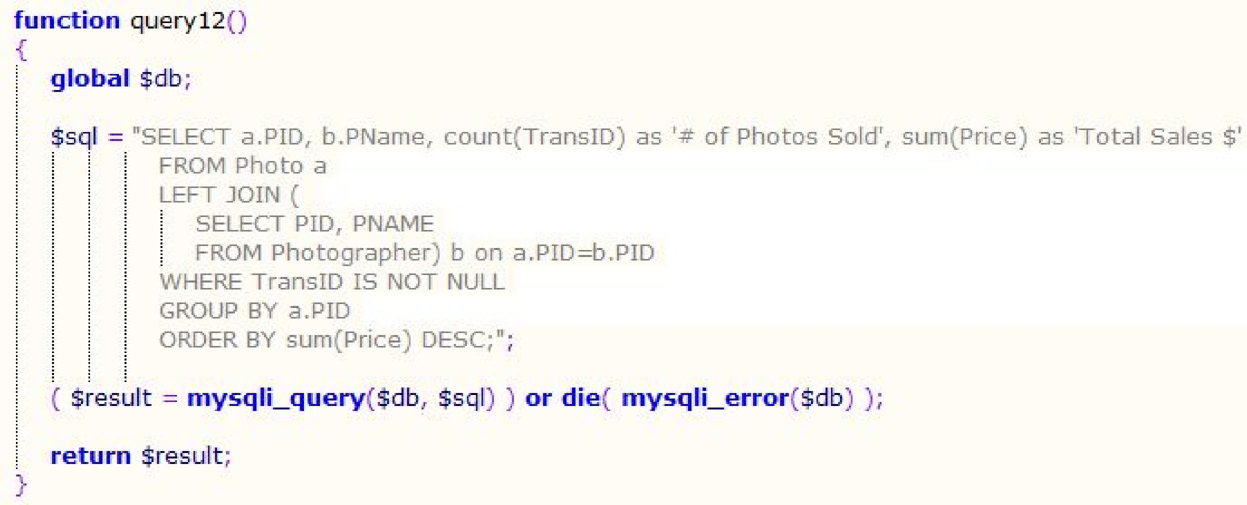
```
function query10($val, $val2)
{
    global $db;

    $sql = "UPDATE Photographer
            SET PName = '$val2'
            WHERE PID = (
                SELECT PID
                FROM Photo
                WHERE PhotoID = $val);";

    ( $result = mysqli_query($db, $sql) ) or die( mysqli_error($db) );
}
```

11. Compute total sales per customer



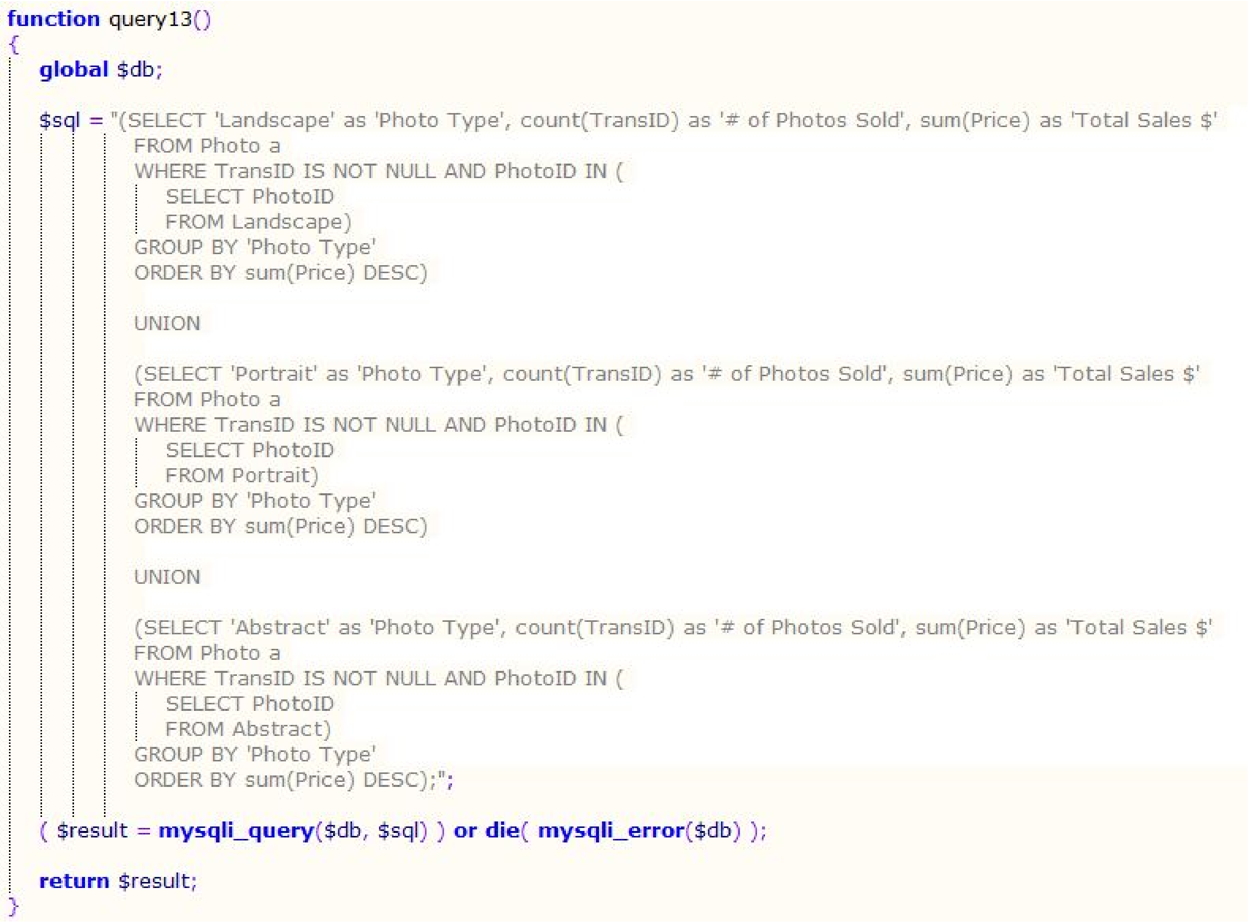
```
function query11()
{
    global $db;

    $sql = "SELECT b.CName, a.LoginName, count(*) as '# of Transactions', sum(a.TotalAmount) as 'Total Sales $'
            FROM Transaction a
            LEFT JOIN (
                SELECT CName, LoginName
                FROM Customer) b ON b.LoginName=a.LoginName
            GROUP BY a.LoginName
            ORDER BY sum(a.TotalAmount) DESC";

    ( $result = mysqli_query($db, $sql) ) or die( mysqli_error($db) );

    return $result;
}
```

12. Compute total sales per photographer sorted by photographer



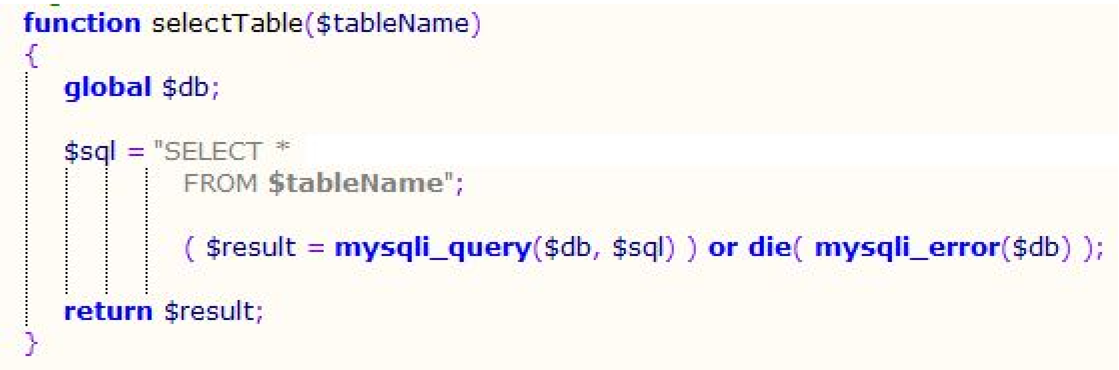
```
function query12()
{
    global $db;

    $sql = "SELECT a.PID, b.PName, count(TransID) as '# of Photos Sold', sum(Price) as 'Total Sales $'
            FROM Photo a
            LEFT JOIN (
                SELECT PID, PNAME
                FROM Photographer) b on a.PID=b.PID
            WHERE TransID IS NOT NULL
            GROUP BY a.PID
            ORDER BY sum(Price) DESC;";

    ( $result = mysqli_query($db, $sql) ) or die( mysqli_error($db) );

    return $result;
}
```

13. Compute total sales by photo type (portrait, landscape etc.)



```
function query13()
{
    global $db;

    $sql = "(SELECT 'Landscape' as 'Photo Type', count(TransID) as '# of Photos Sold', sum(Price) as 'Total Sales $'
            FROM Photo a
            WHERE TransID IS NOT NULL AND PhotoID IN (
                SELECT PhotoID
                FROM Landscape)
            GROUP BY 'Photo Type'
            ORDER BY sum(Price) DESC)

            UNION

            (SELECT 'Portrait' as 'Photo Type', count(TransID) as '# of Photos Sold', sum(Price) as 'Total Sales $'
            FROM Photo a
            WHERE TransID IS NOT NULL AND PhotoID IN (
                SELECT PhotoID
                FROM Portrait)
            GROUP BY 'Photo Type'
            ORDER BY sum(Price) DESC)

            UNION

            (SELECT 'Abstract' as 'Photo Type', count(TransID) as '# of Photos Sold', sum(Price) as 'Total Sales $'
            FROM Photo a
            WHERE TransID IS NOT NULL AND PhotoID IN (
                SELECT PhotoID
                FROM Abstract)
            GROUP BY 'Photo Type'
            ORDER BY sum(Price) DESC);";

    ( $result = mysqli_query($db, $sql) ) or die( mysqli_error($db) );

    return $result;
}
```

14. Compute top n dates (in a total sales per date list)

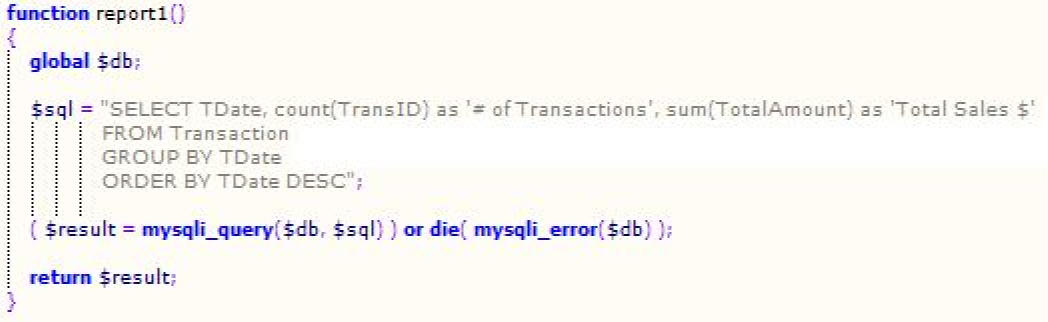


```
function selectTable($tableName)
{
    global $db;

    $sql = "SELECT *
            FROM $tableName";

            ( $result = mysqli_query($db, $sql) ) or die( mysqli_error($db) );

    return $result;
}
```

To see the result of any of these queries execute them at:
https://web.njit.edu/~mdm56/cs331/deliverable3/QueryPortal.php?func=query

SQL From Reporting Portal

Report 1. Sales Per Date



```
function report1()
{
  global $db;

  $sql = "SELECT TDate, count(TransID) as '# of Transactions', sum(TotalAmount) as 'Total Sales $'
        FROM Transaction
        GROUP BY TDate
        ORDER BY TDate DESC";

  ( $result = mysqli_query($db, $sql) ) or die( mysqli_error($db) );

  return $result;
}
```

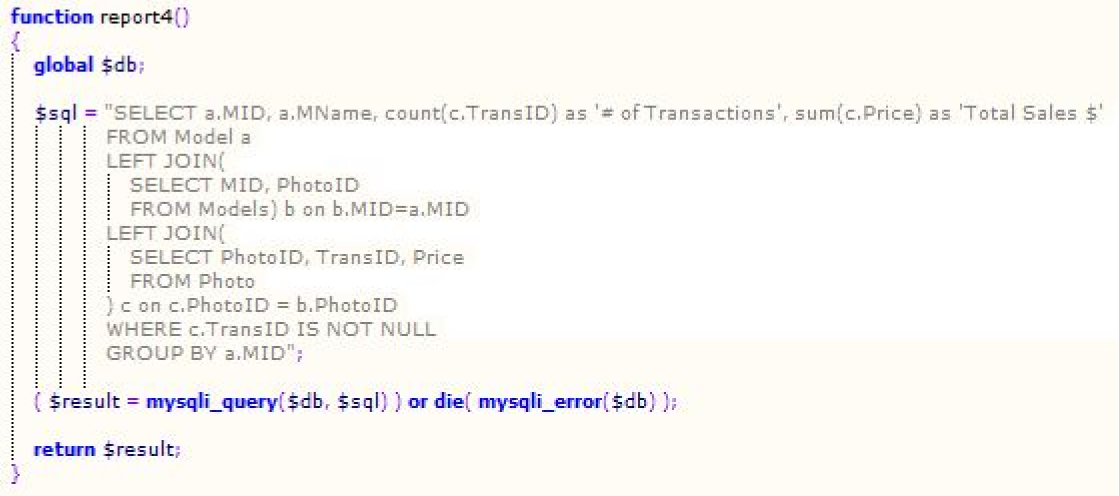
Report 2. Sales Per Customer

Same as query11();

Report 3. Sales Per Photographer

Same as query12();

Report 4. Sales Per Model

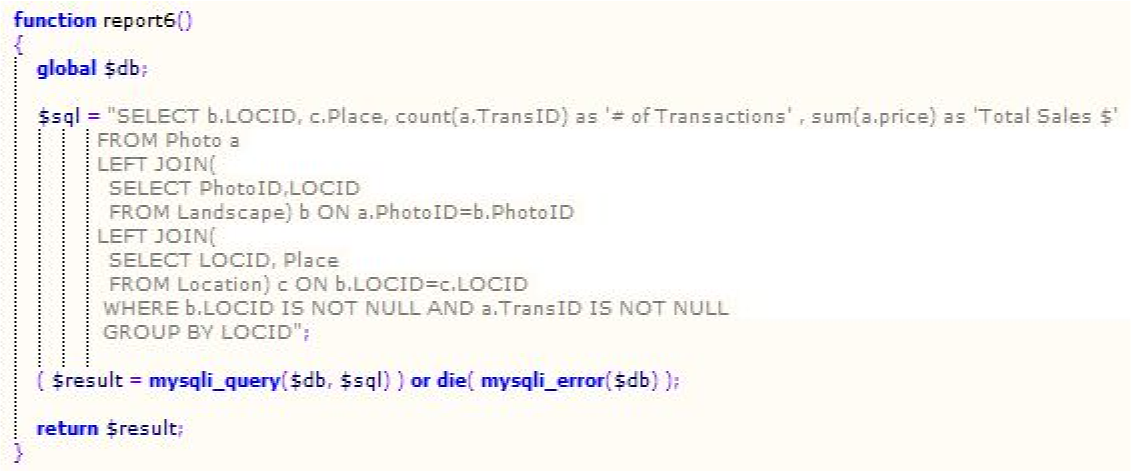


```
function report4()
{
  global $db;

  $sql = "SELECT a.MID, a.MName, count(c.TransID) as '# of Transactions', sum(c.Price) as 'Total Sales $'
        FROM Model a
        LEFT JOIN(
          SELECT MID, PhotoID
          FROM Models) b on b.MID=a.MID
        LEFT JOIN(
          SELECT PhotoID, TransID, Price
          FROM Photo
        ) c on c.PhotoID = b.PhotoID
        WHERE c.TransID IS NOT NULL
        GROUP BY a.MID";

  ( $result = mysqli_query($db, $sql) ) or die( mysqli_error($db) );

  return $result;
}
```

Report 5. Sales Per Photo Type

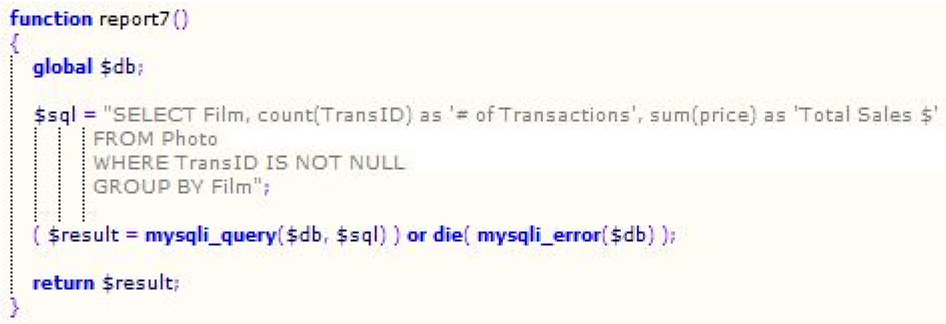
Same as query13();

Report 6. Sales Per Location



```
function report6()
{
  global $db;

  $sql = "SELECT b.LOCID, c.Place, count(a.TransID) as '# of Transactions' , sum(a.price) as 'Total Sales $'
        FROM Photo a
        LEFT JOIN(
          SELECT PhotoID,LOCID
          FROM Landscape) b ON a.PhotoID=b.PhotoID
        LEFT JOIN(
          SELECT LOCID, Place
          FROM Location) c ON b.LOCID=c.LOCID
         WHERE b.LOCID IS NOT NULL AND a.TransID IS NOT NULL
         GROUP BY LOCID";

  ( $result = mysqli_query($db, $sql) ) or die( mysqli_error($db) );

  return $result;
}
```

Report 7. Sales Per Film Type

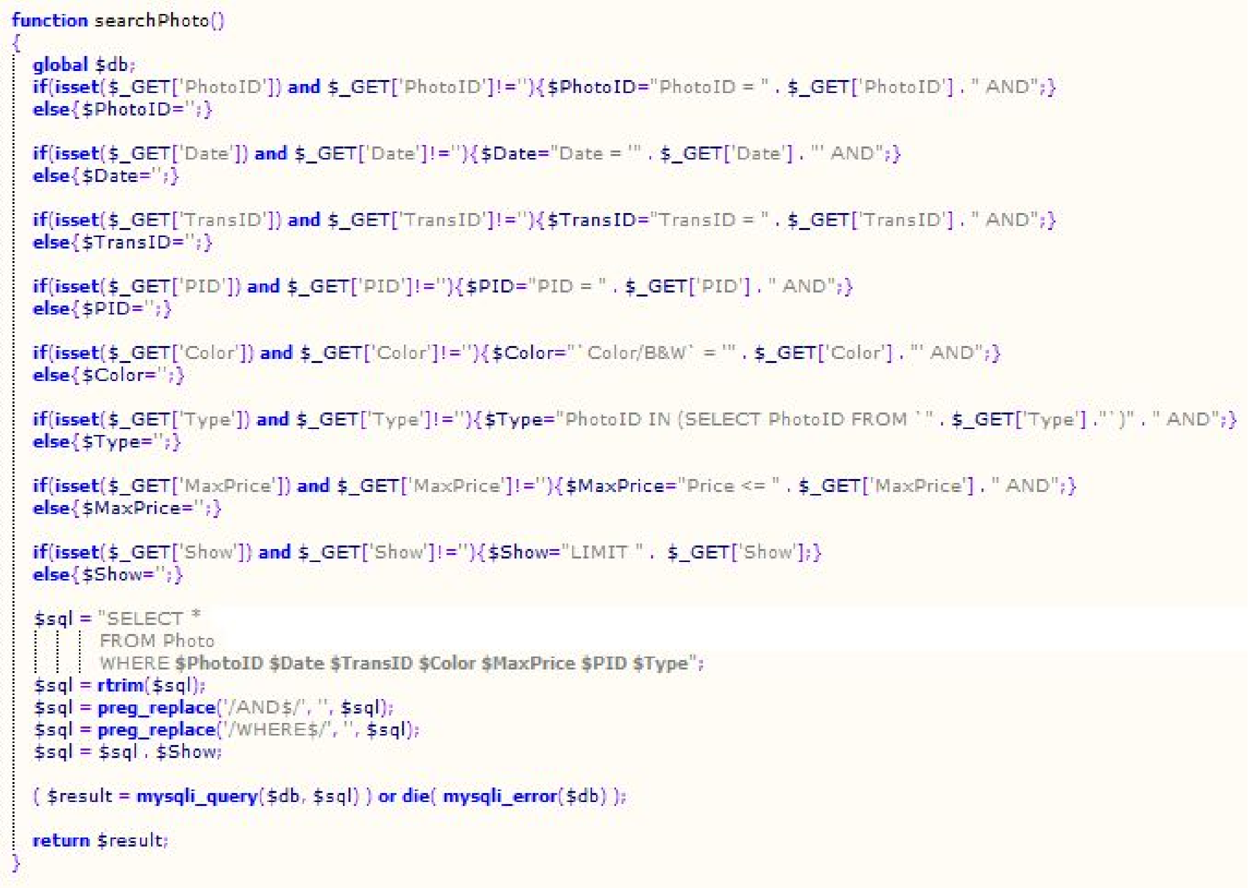


```
function report7()
{
  global $db;

  $sql = "SELECT Film, count(TransID) as '# of Transactions', sum(price) as 'Total Sales $'
        FROM Photo
        WHERE TransID IS NOT NULL
        GROUP BY Film";

  ( $result = mysqli_query($db, $sql) ) or die( mysqli_error($db) );

  return $result;
}
```

To see the result of any of these queries execute them at:
https://web.njit.edu/~mdm56/cs331/deliverable3/ReportingPortal.php

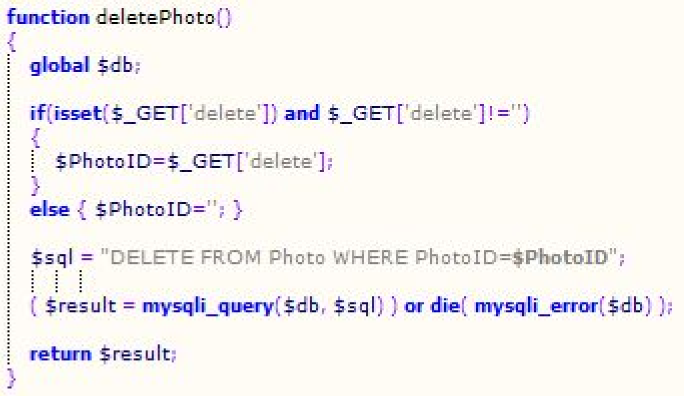
SQL From Photo Portal

Task 1. Search Photo Database



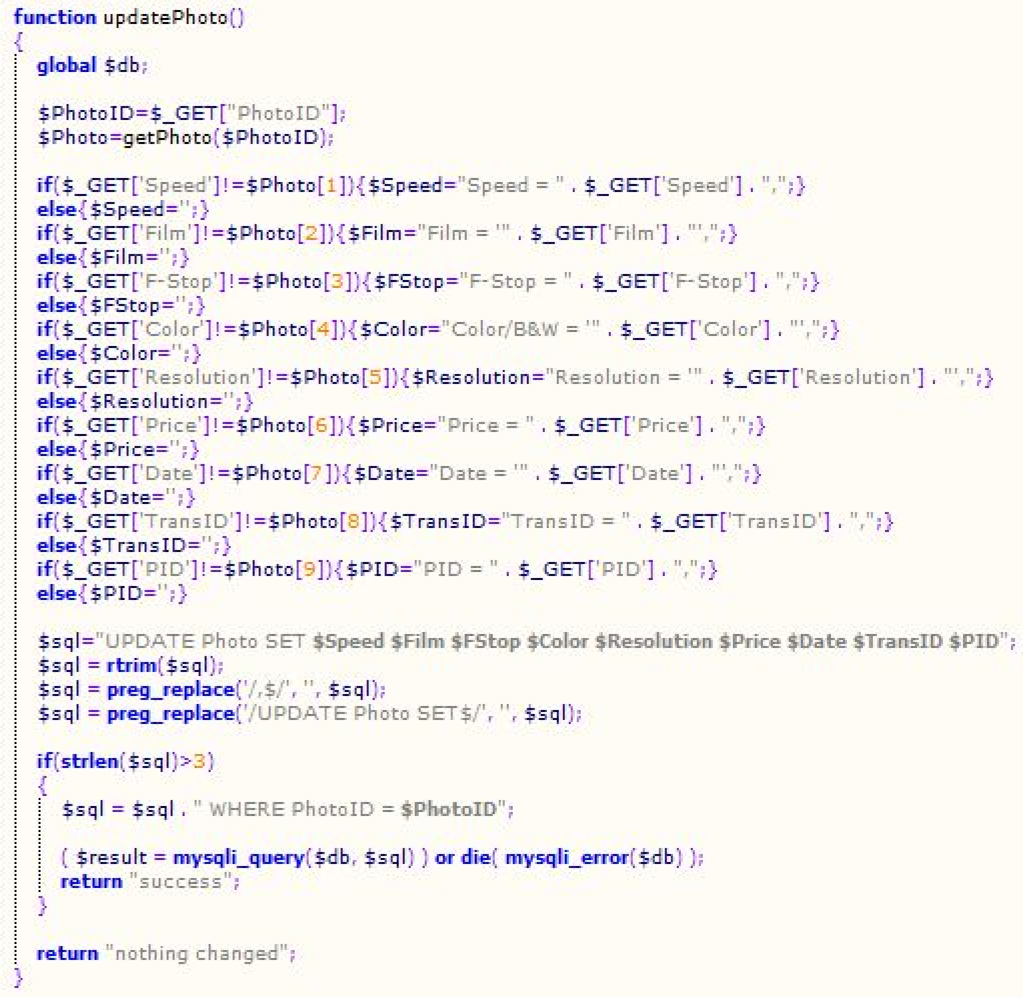
```
function searchPhoto()
{
  global $db;
  if(isset($_GET['PhotoID']) and $_GET['PhotoID']!=''){$PhotoID="PhotoID = " . $_GET['PhotoID'] . " AND";}
  else{$PhotoID='';}

  if(isset($_GET['Date']) and $_GET['Date']!=''){$Date="Date = '" . $_GET['Date'] . "' AND";}
  else{$Date='';}

  if(isset($_GET['TransID']) and $_GET['TransID']!=''){$TransID="TransID = " . $_GET['TransID'] . " AND";}
  else{$TransID='';}

  if(isset($_GET['PID']) and $_GET['PID']!=''){$PID="PID = " . $_GET['PID'] . " AND";}
  else{$PID='';}

  if(isset($_GET['Color']) and $_GET['Color']!=''){$Color="`Color/B&W` = '" . $_GET['Color'] . "' AND";}
  else{$Color='';}

  if(isset($_GET['Type']) and $_GET['Type']!=''){$Type="PhotoID IN (SELECT PhotoID FROM `" . $_GET['Type'] ."`)" . " AND";}
  else{$Type='';}

  if(isset($_GET['MaxPrice']) and $_GET['MaxPrice']!=''){$MaxPrice="Price <= " . $_GET['MaxPrice'] . " AND";}
  else{$MaxPrice='';}

  if(isset($_GET['Show']) and $_GET['Show']!=''){$Show="LIMIT " .  $_GET['Show'];}
  else{$Show='';}

  $sql = "SELECT *
          FROM Photo
          WHERE $PhotoID $Date $TransID $Color $MaxPrice $PID $Type";
  $sql = rtrim($sql);
  $sql = preg_replace('/AND$/', '', $sql);
  $sql = preg_replace('/WHERE$/', '', $sql);
  $sql = $sql . $Show;

  ( $result = mysqli_query($db, $sql) ) or die( mysqli_error($db) );

  return $result;
}
```

Task 2. Delete Photo From Photo Database



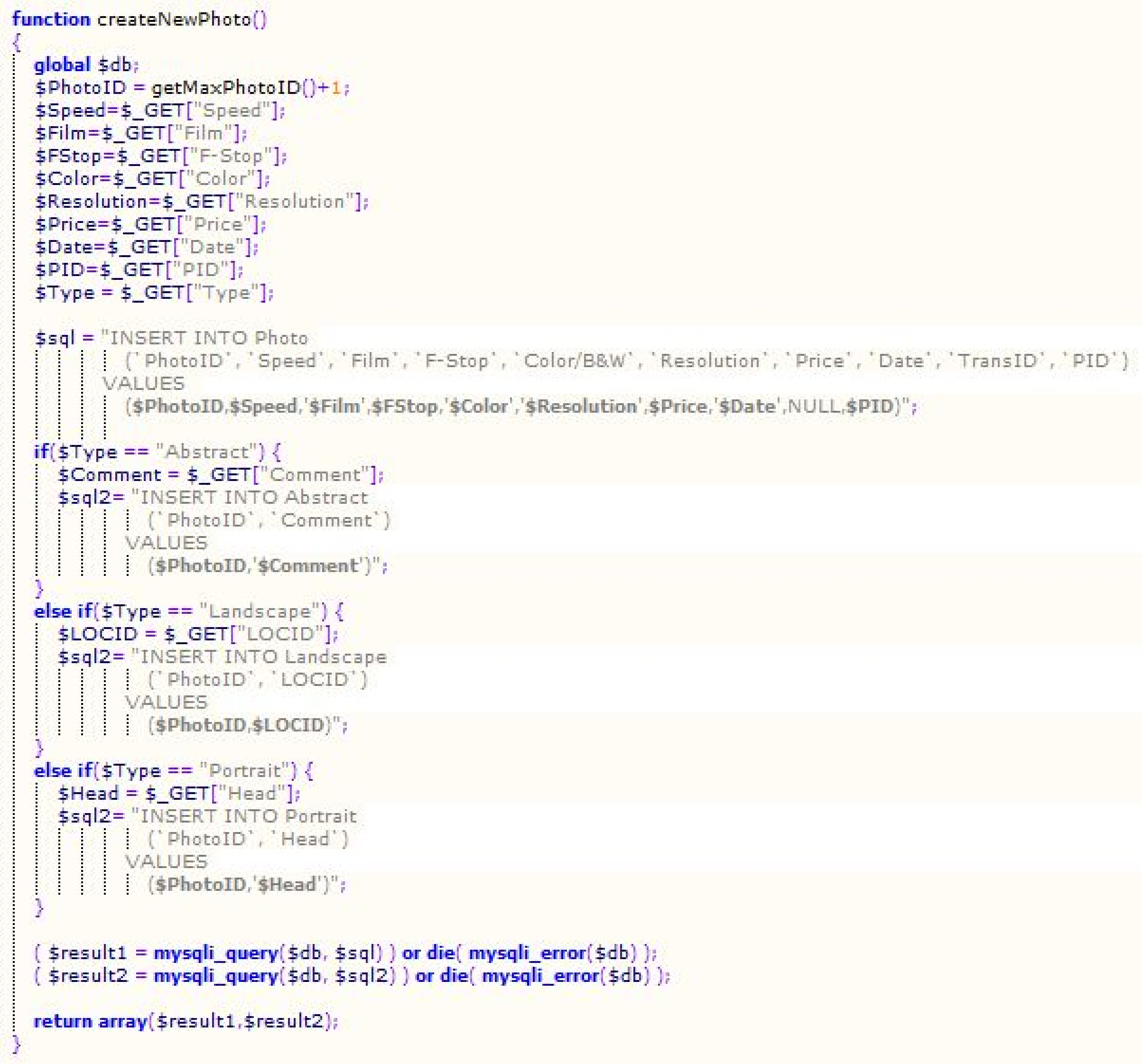
```
function deletePhoto()
{
  global $db;

  if(isset($_GET['delete']) and $_GET['delete']!='')
  {
    $PhotoID=$_GET['delete'];
  }
  else { $PhotoID=''; }

  $sql = "DELETE FROM Photo WHERE PhotoID=$PhotoID";

  ( $result = mysqli_query($db, $sql) ) or die( mysqli_error($db) );

  return $result;
}
```

Task 3. Update Photo in Photo Database



```
function updatePhoto()
{
  global $db;

  $PhotoID=$_GET["PhotoID"];
  $Photo=getPhoto($PhotoID);

  if($_GET['Speed']!=$Photo[1]){$Speed="Speed = " . $_GET['Speed'] . ",";}
  else{$Speed='';}
  if($_GET['Film']!=$Photo[2]){$Film="Film = '" . $_GET['Film'] . "',";}
  else{$Film='';}
  if($_GET['F-Stop']!=$Photo[3]){$FStop="F-Stop = " . $_GET['F-Stop'] . ",";}
  else{$FStop='';}
  if($_GET['Color']!=$Photo[4]){$Color="Color/B&W = '" . $_GET['Color'] . "',";}
  else{$Color='';}
  if($_GET['Resolution']!=$Photo[5]){$Resolution="Resolution = '" . $_GET['Resolution'] . "',";}
  else{$Resolution='';}
  if($_GET['Price']!=$Photo[6]){$Price="Price = " . $_GET['Price'] . ",";}
  else{$Price='';}
  if($_GET['Date']!=$Photo[7]){$Date="Date = '" . $_GET['Date'] . "',";}
  else{$Date='';}
  if($_GET['TransID']!=$Photo[8]){$TransID="TransID = " . $_GET['TransID'] . ",";}
  else{$TransID='';}
  if($_GET['PID']!=$Photo[9]){$PID="PID = " . $_GET['PID'] . ",";}
  else{$PID='';}

  $sql="UPDATE Photo SET $Speed $Film $FStop $Color $Resolution $Price $Date $TransID $PID";
  $sql = rtrim($sql);
  $sql = preg_replace('/,$/', '', $sql);
  $sql = preg_replace('/UPDATE Photo SET$/', '', $sql);

  if(strlen($sql)>3)
  {
    $sql = $sql . " WHERE PhotoID = $PhotoID";

    ( $result = mysqli_query($db, $sql) ) or die( mysqli_error($db) );
    return "success";
  }

  return "nothing changed";
}
```

Task 4. Add New Photo to Photo Database

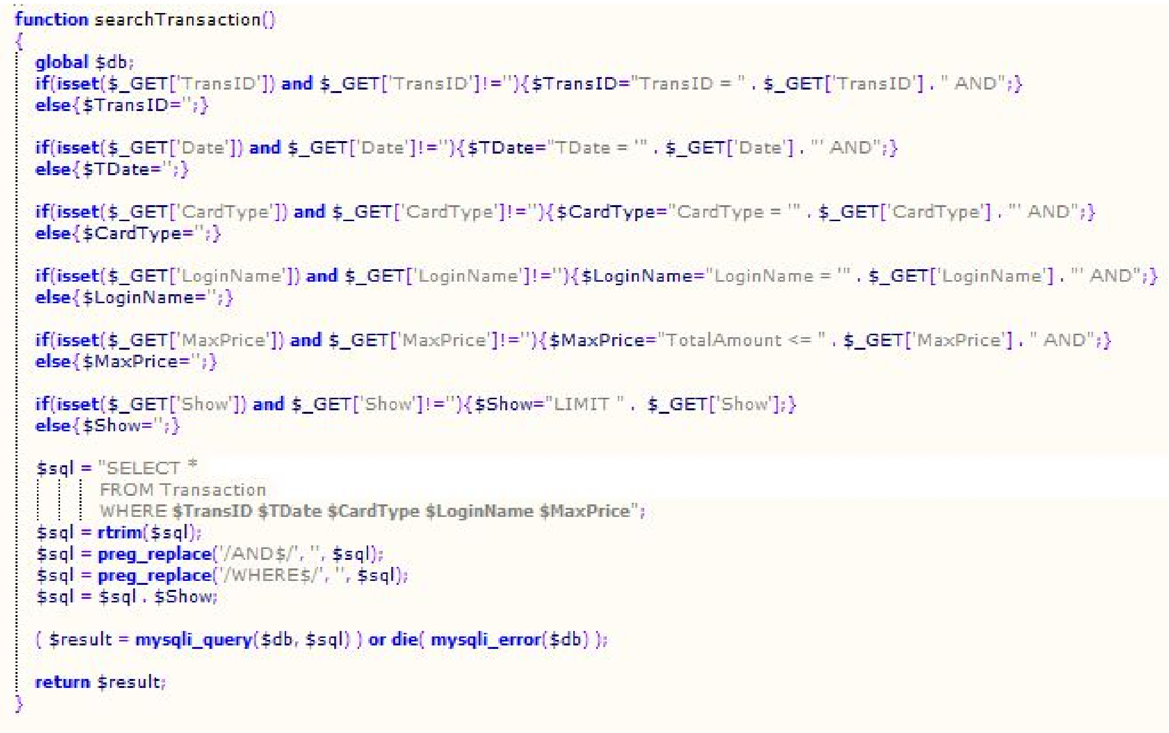


```
function createNewPhoto()
{
    global $db;
    $PhotoID = getMaxPhotoID()+1;
    $Speed=$_GET["Speed"];
    $Film=$_GET["Film"];
    $FStop=$_GET["F-Stop"];
    $Color=$_GET["Color"];
    $Resolution=$_GET["Resolution"];
    $Price=$_GET["Price"];
    $Date=$_GET["Date"];
    $PID=$_GET["PID"];
    $Type = $_GET["Type"];

    $sql = "INSERT INTO Photo
                (`PhotoID`, `Speed`, `Film`, `F-Stop`, `Color/B&W`, `Resolution`, `Price`, `Date`, `TransID`, `PID`)
            VALUES
                ($PhotoID,$Speed,'$Film',$FStop,'$Color','$Resolution',$Price,'$Date',NULL,$PID)";

    if($Type == "Abstract") {
        $Comment = $_GET["Comment"];
        $sql2= "INSERT INTO Abstract
                    (`PhotoID`, `Comment`)
                VALUES
                    ($PhotoID,'$Comment')";
    }
    else if($Type == "Landscape") {
        $LOCID = $_GET["LOCID"];
        $sql2= "INSERT INTO Landscape
                    (`PhotoID`, `LOCID`)
                VALUES
                    ($PhotoID,$LOCID)";
    }
    else if($Type == "Portrait") {
        $Head = $_GET["Head"];
        $sql2= "INSERT INTO Portrait
                    (`PhotoID`, `Head`)
                VALUES
                    ($PhotoID,'$Head')";
    }

    ( $result1 = mysqli_query($db, $sql) ) or die( mysqli_error($db) );
    ( $result2 = mysqli_query($db, $sql2) ) or die( mysqli_error($db) );

    return array($result1,$result2);
}
```

To see the result of any of these queries execute them at: https://web.njit.edu/~mdm56/cs331/deliverable3/PhotoPortal.php

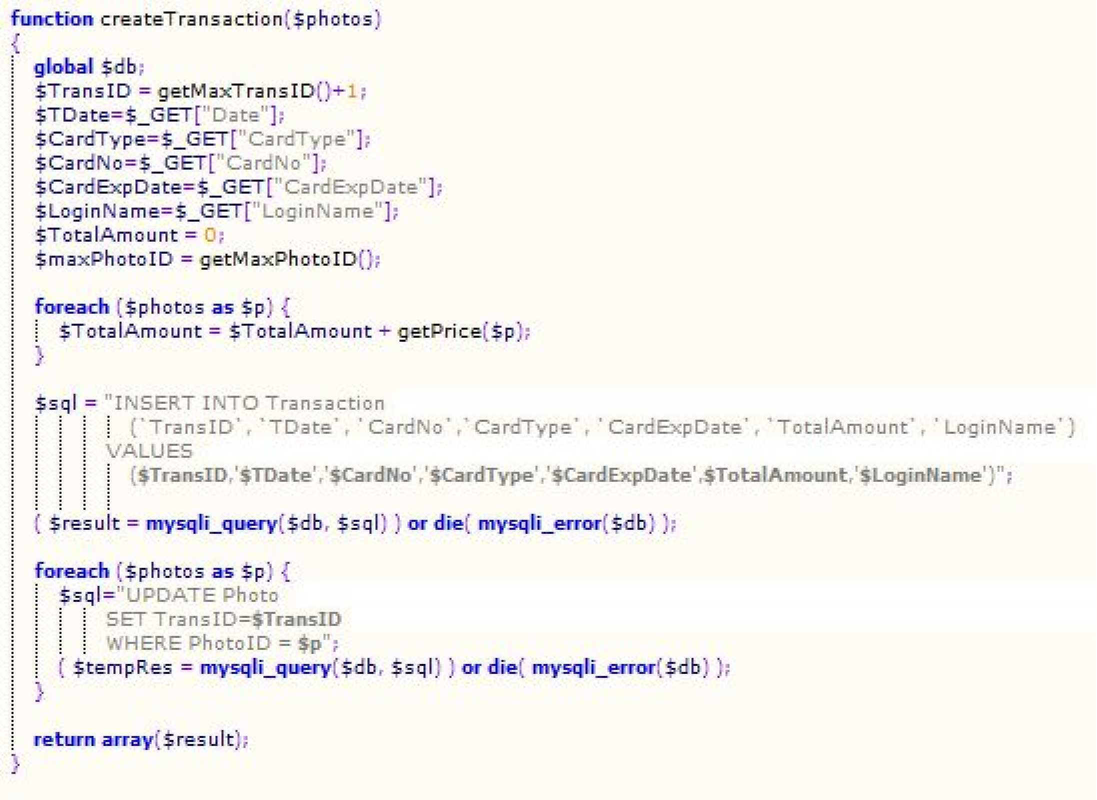
SQL From Sales Portal

Task 1. Search Transactions



```
function searchTransaction()
{
  global $db;
  if(isset($_GET['TransID']) and $_GET['TransID']!=''){$TransID="TransID = " . $_GET['TransID'] . " AND";}
  else{$TransID='';}

  if(isset($_GET['Date']) and $_GET['Date']!=''){$TDate="TDate = '" . $_GET['Date'] . "' AND";}
  else{$TDate='';}

  if(isset($_GET['CardType']) and $_GET['CardType']!=''){$CardType="CardType = '" . $_GET['CardType'] . "' AND";}
  else{$CardType='';}

  if(isset($_GET['LoginName']) and $_GET['LoginName']!=''){$LoginName="LoginName = '" . $_GET['LoginName'] . "' AND";}
  else{$LoginName='';}

  if(isset($_GET['MaxPrice']) and $_GET['MaxPrice']!=''){$MaxPrice="TotalAmount <= " . $_GET['MaxPrice'] . " AND";}
  else{$MaxPrice='';}

  if(isset($_GET['Show']) and $_GET['Show']!=''){$Show="LIMIT " .  $_GET['Show'];}
  else{$Show='';}

  $sql = "SELECT *
        FROM Transaction
        WHERE $TransID $TDate $CardType $LoginName $MaxPrice";
  $sql = rtrim($sql);
  $sql = preg_replace('/AND$/', '', $sql);
  $sql = preg_replace('/WHERE$/', '', $sql);
  $sql = $sql . $Show;

  ( $result = mysqli_query($db, $sql) ) or die( mysqli_error($db) );

  return $result;
}
```

Task 2. Process New Transactions



```
function createTransaction($photos)
{
  global $db;
  $TransID = getMaxTransID()+1;
  $TDate=$_GET["Date"];
  $CardType=$_GET["CardType"];
  $CardNo=$_GET["CardNo"];
  $CardExpDate=$_GET["CardExpDate"];
  $LoginName=$_GET["LoginName"];
  $TotalAmount = 0;
  $maxPhotoID = getMaxPhotoID();

  foreach ($photos as $p) {
    $TotalAmount = $TotalAmount + getPrice($p);
  }

  $sql = "INSERT INTO Transaction
          (`TransID`, `TDate`, `CardNo`,`CardType`, `CardExpDate`, `TotalAmount`, `LoginName`)
        VALUES
          ($TransID,'$TDate','$CardNo','$CardType','$CardExpDate',$TotalAmount,'$LoginName')";

  ( $result = mysqli_query($db, $sql) ) or die( mysqli_error($db) );

  foreach ($photos as $p) {
    $sql="UPDATE Photo
        SET TransID=$TransID
        WHERE PhotoID = $p";
    ( $tempRes = mysqli_query($db, $sql) ) or die( mysqli_error($db) );
  }

  return array($result);
}
```

To see the result of any of these queries execute them at: **https://web.njit.edu/~mdm56/cs331/deliverable3/SalesPortal.php**

There are other instances of sql used in other functions for other webpages. To see those please look at the *MyFunctions.txt.* That file contains all the sql used in the entire project including the functions you see above.