

IE332 Assignment #X

I certify that the submitted work does not violate any academic misconduct rules, and that it is solely my own work. By listing my name I acknowledge that any misconduct will result in appropriate consequences. Moreover, **I have read and understood the assignment instructions.**

``As a Boilermaker pursuing academic excellence, I pledge to be honest and true in all that I do. Accountable together - we are Purdue."

FULL name, as indicated in Blackboard (no PUIDs!):

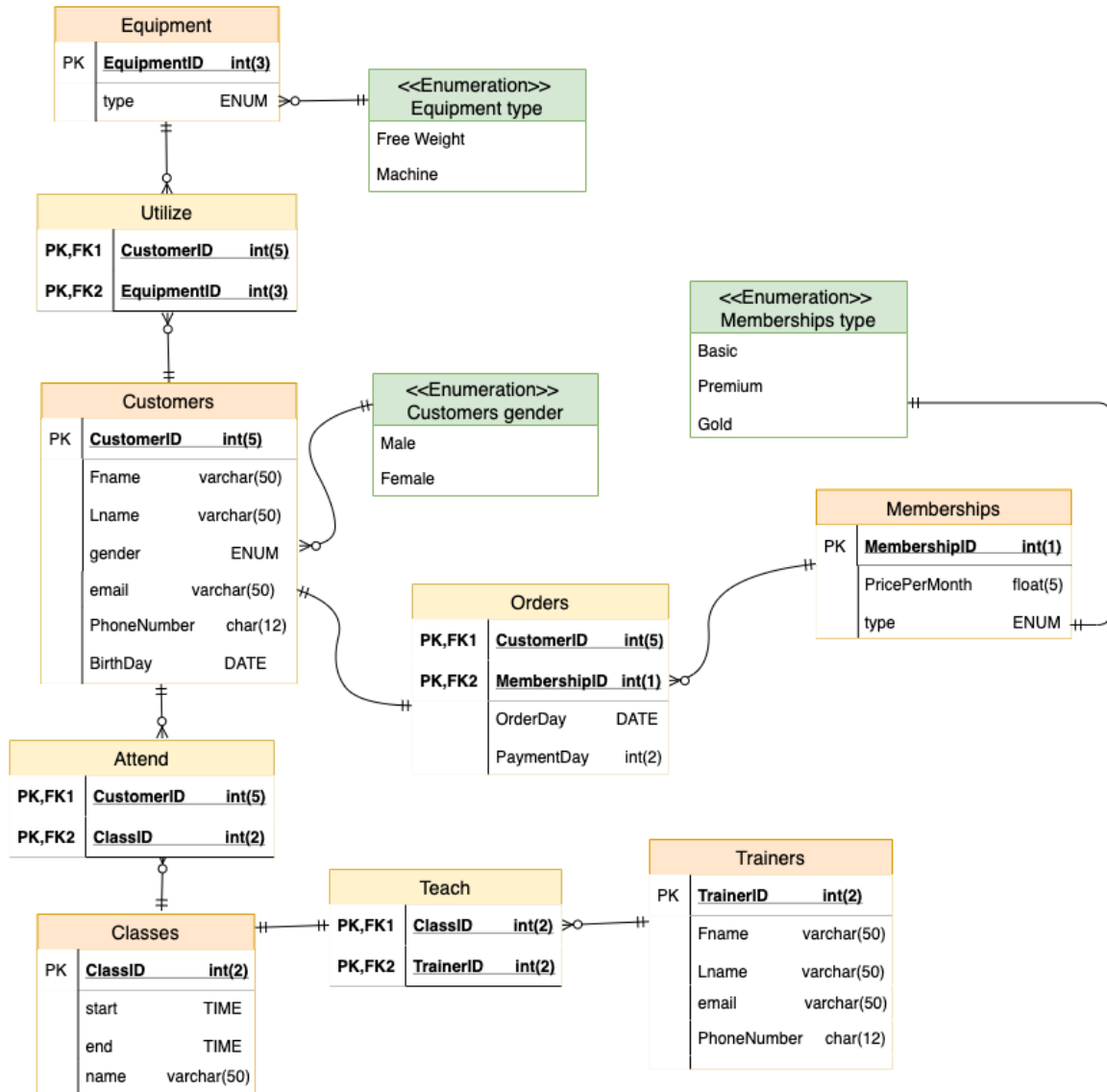
1. Daniel Moraes Pinheiro Reinaux

Date: 03/28/2020

WEBPAGE URL : <https://web.ics.purdue.edu/~dreinaux/quiz.html>

1. DATABASE

(a) ERD



(b)

```
CREATE TABLE Equipment(  
    EquipmentID int(3) UNIQUE NOT NULL AUTO_INCREMENT,  
    type ENUM('Machine', 'Free Weight'),  
    PRIMARY KEY (EquipmentID)  
);
```

```
CREATE TABLE Customers(  
    CustomerID int(5) UNIQUE NOT NULL AUTO_INCREMENT,  
    FName varchar(50),  
    Lname varchar(50),  
    gender ENUM('Male', 'Female'),
```

```
email varchar(50),
PhoneNumber char(12),
BirthDay DATE,
PRIMARY KEY (CustomerID)
);
```

```
CREATE TABLE Memberships(
MembershipID int(1) UNIQUE NOT NULL AUTO_INCREMENT,
PricePerMonth float(5),
type ENUM('Basic', 'Premium', 'Gold'),
PRIMARY KEY (MembershipID)
);
```

```
CREATE TABLE Classes(
ClassID int(2) UNIQUE NOT NULL AUTO_INCREMENT,
name varchar(50),
start TIME,
ends TIME,
PRIMARY KEY (ClassID)
);
```

```
CREATE TABLE Trainers(
TrainerID int(2) UNIQUE NOT NULL AUTO_INCREMENT,
Fname varchar(50),
Lname varchar(50),
email varchar(50),
PhoneNumber int(10),
PRIMARY KEY (TrainerID)
);
```

```
CREATE TABLE Utilize(
CustomerID int(5) UNIQUE NOT NULL,
EquipmentID int(3) UNIQUE NOT NULL,
PRIMARY KEY (CustomerID, EquipmentID),
FOREIGN KEY (CustomerID) REFERENCES Customers (CustomerID),
FOREIGN KEY (EquipmentID) REFERENCES Equipment (EquipmentID)
);
```

```
CREATE TABLE Orders(
CustomerID int(5) NOT NULL,
MembershipID int(1) NOT NULL,
OrderDay DATE,
PaymentDay int(2),
PRIMARY KEY (CustomerID, MembershipID),
FOREIGN KEY (CustomerID) REFERENCES Customers (CustomerID),
FOREIGN KEY (MembershipID) REFERENCES Memberships (MembershipID)
);
```

```
CREATE TABLE Attend(
CustomerID int(5) NOT NULL,
ClassID int(2) NOT NULL,
PRIMARY KEY (CustomerID, ClassID),
FOREIGN KEY (CustomerID) REFERENCES Customers (CustomerID),
```

```

FOREIGN KEY (ClassID) REFERENCES Classes (ClassID)
);

CREATE TABLE Teach(
  ClassID int(2) NOT NULL,
  TrainerID int(2) NOT NULL,
  PRIMARY KEY (ClassID, TrainerID),
  FOREIGN KEY (ClassID) REFERENCES Classes (ClassID), ON DELETE CASCADE
  FOREIGN KEY (TrainerID) REFERENCES Trainers (TrainerID) ON DELETE CASCADE
);

```

2. WEBSITE + SQL

(a)

```

<!DOCTYPE html>
<html>
<head>
<title>332 QUIZ</title>
<script src="quiz.js"></script>
</head>
<body>

<h1 style="text-align:center;">Let's get fit!</h1>

<div style="text-align:center;">
<h2 style="text-align:center;">System structure</h2>
<p style="text-align:center;" title="System Structure">As I am very passionate about fitness, I decided
to choose a gym as the system for my project design. Consider a simple commercial gym oriented
towards various classes. There are your usual      customers of the gym, who pay for a monthly
gym membership of their choosing, among 3 different ones. Although the gym is focused on classes, it
also has all the equipment necessary for customers to just go and use those instead of attending any
classes if they wish to do so. The customer could also use both services. Lastly, the classes are taught by
Instructors/Trainers of the gym.
</p>
</div>

<ul>
<li>Equipment have a unique equipment ID and a type which can be free weights or machines.</li>
<li>Customers have a unique defining customer ID, a first name, a last name, a gender, an email, a
phone number and a birth date.</li>
<li>The different 3 different gym memberships are defined by a unique ID, a type which can be either
Basic, Premium, or Gold and a price per month.</li>
<li>The classes are defined by a unique class ID, a name, and a start and end times (of the day).</li>
<li>Lastly, the gym instructors have a trainer ID, a first name, a last name, an email and a phone
number.</li>
</ul>

<div style="text-align:center;">

</div>

```

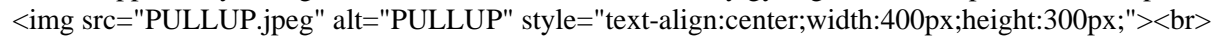
(b)

THE 3 LIFTING PILLARS:

- [Pull Up](#PU)
- [Bench Press](#BP)
- [Squat](#S)

Pull-Up

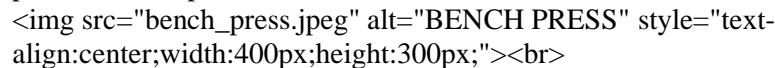
The pull up is, for me, the most effective and reliable back exercise. It is my preferred back exercise because it is the most effective lat-builder, while also requiring strong muscular responses from many secondary back muscles and also the biceps. It is a true test of a person's back strength and part of overall upper body strength. It is a badass exercises that every gym-goer must be proficient at.



- Standing below the pull-up bar, grab the bar with both hands equally distances from each other, at around shoulder width.
- Retract your scapula and push your chest outwards.
- Pull yourself up towards the bar, until reaching the bar with your chin.
- Slowly, lower yourself back down to starting position.

Bench Press

As most people would agree, the bench press is the kind of the upper body lifts. It is for sure the single best chest exercise you can do. It allows you to overload the chest more than any other exercise, while being great at enabling progressive overload due to the use of the barbell. The bench press works primarily your pectoral muscles but secondarily it also works the shoulders and triceps to quite a high extent. It is a staple of any upper body muscle building routine and true test of a person's pectoral strength and part of overall upper body strength. It is a badass exercises that every gym-goer must be proficient at.

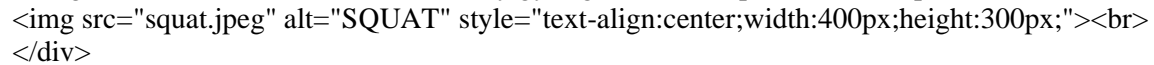


- lay down on the bench, feet flat on the ground and with the barbell hovering over on top of your eyes.
- Retract your scapula and push your chest outwards.
- With your arms wider than your shoulder width, grab the barbell with both hands, and lower it towards your chest slowly.
- Explosively push the bar straight up until your arms are vertical, focusing on contracting your chest.

Squat

Finally, the squat is considered by many the king of all exercises. It is the definite winner for all of lower body muscles. It is for sure the single best legs exercise you can do. Similarly to the bench press, It allows you to overload the legs more than almost any other exercise, while being great at enabling

progressive overload due to the use of the barbell. The squat works primarily your quadriceps, but is also great for hamstrings, gluteus and core muscles for stabilization. It is definitely a staple of any lower body muscle building routine and true test of a person's lower body strength and part of overall body strength. It is a badass exercise that every gym-goer must be proficient at.

A placeholder for an image with alt="SQUAT" and style="text-align:center;width:400px;height:300px;"

<hr>

(c)

HTML WEBFORM:

<h3 style="text-align:center">Populating the database with the form(s) below:</h3>

<div style="text-align:center">

<form name="quizform" action="quiz.php" method="post" onsubmit="return checkMyForm()">

<fieldset>

<legend>Equipment:</legend>

<label for="type">Equipment type:</label>

<input type="radio" id="type" name="type" value="Machine">

<label for="type">Machine</label>

<input type="radio" id="type" name="type" value="Free Weight">

<label for="type">Free Weight</label>

<input type="submit" name="sb" value="submit1">

</fieldset>

<fieldset>

<legend>Customers:</legend>

<label for="cu_Fname">First Name:</label>

<input type="text" id="cu_Fname" name="cu_Fname">

<label for="cu_Lname">Last Name:</label>

<input type="text" id="cu_Lname" name="cu_Lname">

<label for="gender">Gender:</label>

<input type="text" id="gender" name="gender">

```
<label for="cu_email">Email Address:</label>
<input type="email" id="cu_email" name="cu_email">
<label for="cu_phone">Phone Number:</label>
<input type="tel" id="cu_phone" name="cu_phone" placeholder="123-456-7891" pattern="[0-9]{3}-[0-9]{3}-[0-9]{4}">
<label for="bd">Birth Date:</label>
<input type="date" id="bd" name="bd"><br>
<input type="submit" name="sb" value="submit2">

</fieldset>

<fieldset>
<legend><b>Memberships:</b></legend>
<label for="price">Price/mo:</label>
<input type="number" step="0.01" placeholder="1.23" id="price" name="price">
<label for="m_type">Membership type:</label>
<input type="radio" id="m_type" name="m_type">
<label for="m_type">Basic</label>
<input type="radio" id="m_type" name="m_type">
<label for="m_type">Premium</label>
<input type="radio" id="m_type" name="m_type">
<label for="m_type">Gold</label><br>
<input type="submit" name="sb" value="submit3">

</fieldset>

<fieldset>
<legend><b>Classes:</b></legend>
<label for="cl_name">Name:</label>
<input type="text" id="cl_name" name="cl_name">
<label for="start">Start time:</label>
<input type="time" id="start" name="start">
<label for="end">Ending time:</label>
<input type="time" id="end" name="end"><br>
<input type="submit" name="sb" value="submit4">

</fieldset>

<fieldset>
<legend><b>Trainers:</b></legend>
<label for="t_Fname">First Name:</label>
<input type="text" id="t_Fname" name="t_Fname">
<label for="t_Lname">Last Name:</label>
<input type="text" id="t_Lname" name="t_Lname">
<label for="t_email">Email Address:</label>
<input type="email" id="t_email" name="t_email">
<label for="t_phone">Phone Number:</label>
<input type="tel" id="t_phone" placeholder="123-456-7891" pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}"
name="t_phone"><br>
<input type="submit" name="sb" value="submit5">
</fieldset>
<input type="reset">
</form>
```

</div>

Populating the database with the form(s) below:

Equipment:

Equipment type: ☐ Machine ☐ Free Weight

Customers:

First Name: Last Name: Gender: ☐ Male ☐ Female Email Address: Phone Number: Birth Date:

Memberships:

Price/mo: Membership type: ☐ Basic ☐ Premium ☐ Gold

Classes:

Name: Start time: Ending time:

Trainers:

First Name: Last Name: Email Address: Phone Number:

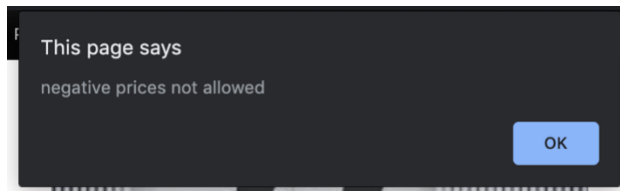
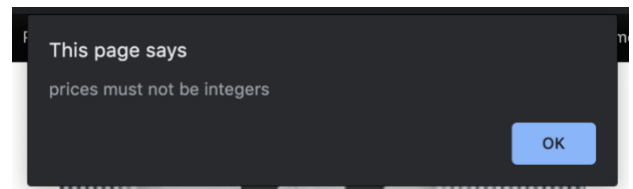
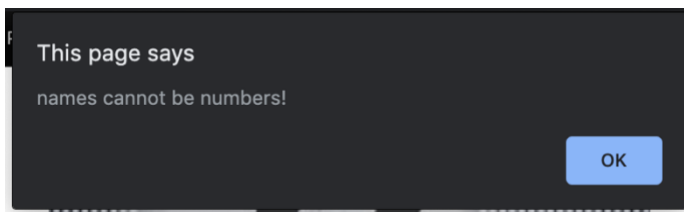
JAVASCRIPT VALIDATION CODE

```
function checkMyForm() {
  var a = document.forms["quizform"]["cu_Fname"].value;
  var aa = document.forms["quizform"]["cu_Lname"].value;

  var b = document.forms["quizform"]["price"].value;
  var c = document.forms["quizform"]["t_Fname"].value;
  var cc = document.forms["quizform"]["t_Lname"].value;

  var d = document.forms["quizform"]["cl_name"].value;

  if (!isNaN(a) | !isNaN(aa)){
    window.alert("names cannot be numbers!");
    return false;
  }
  if (!isNaN(c) | !isNaN(cc)){
    window.alert("names cannot be numbers!");
    return false;
  }
  if (!isNaN(d)){
    window.alert("names cannot be numbers!");
    return false;
  }
  if (b < 0) {
    window.alert("negative prices not allowed");
    return false;
  } else if (b % 1 == 0){
    window.alert("prices must not be integers");
    return false;
  }
}
```

(d)

```
<?php
    $servername = "mydb.itap.purdue.edu";
    $username = "dreinaux";
    $password = "edmath16";
    $dbname = "dreinaux";
    // Create connection
    $conn = mysqli_connect($servername, $username, $password , $dbname);
    // Check connection
    if (!$conn) {
        die("Connection failed: " . mysqli_connect_error());
    }
    if(isset($_SERVER['HTTP_X_FORWARDED_FOR']) &&
$_SERVER['HTTP_X_FORWARDED_FOR'] != "") {
    $ip_address = $_SERVER['HTTP_X_FORWARDED_FOR'];
    } else {
    $ip_address = $_SERVER['REMOTE_ADDR'];
    }

    echo "Data submitted by IP address: $ip_address";

    if ($_POST['sb'] == 'submit1'){
        $type = $_POST['type'];
        $sql = "INSERT INTO Equipment (type) VALUES ('$type')";
        $result = mysqli_query($conn, $sql);
        if ($result){
            echo "New equipment record successfully created! Please go back to the previous page to add
another record.";
        }
        else{
```

```

        echo "Error: " . $sql . "<br>" . mysqli_error($conn);
    }
}

if ($_POST['sb'] == 'submit2'){
    $fname = $_POST['cu_fname'];
    $lname = $_POST['cu_lname'];
    $gender = $_POST['gender'];
    $email = $_POST['cu_email'];
    $phone = $_POST['cu_phone'];
    $bd = $_POST['bd'];
    $sql = "INSERT INTO Customers (Fname, Lname, gender, email, PhoneNumber, BirthDay)
VALUES ('$fname', '$lname', '$gender', '$email', '$phone', '$bd')";
    $result = mysqli_query($conn, $sql);
    if ($result){
        echo "New customer record successfully created! Please go back to the previous page to add another
record.";
    }
    else{
        echo "Error: " . $sql . "<br>" . mysqli_error($conn);
    }
}

if ($_POST['sb'] == 'submit3'){
    $price = $_POST['price'];
    $type = $_POST['m_type'];
    $sql = "INSERT INTO Memberships (PricePerMonth, type) VALUES ('$price', '$type')";
    $result = mysqli_query($conn, $sql);
    if ($result){
        echo "New membership record successfully created! Please go back to the previous page to add
another record.";
    }
    else{
        echo "Error: " . $sql . "<br>" . mysqli_error($conn);
    }
}

if ($_POST['sb'] == 'submit4'){
    $name = $_POST['cl_name'];
    $start = $_POST['start'];
    $end = $_POST['end'];
    $sql = "INSERT INTO Classes (name, start, ends) VALUES ('$name', '$start', '$end')";
    $result = mysqli_query($conn, $sql);
    if ($result){
        echo "New class record successfully created! Please go back to the previous page to add another
record.";
    }
    else{
        echo "Error: " . $sql . "<br>" . mysqli_error($conn);
    }
}

if ($_POST['sb'] == 'submit5'){

```

```

    $fname = $_POST['t_Fname'];
    $lname = $_POST['t_Lname'];
    $email = $_POST['t_email'];
    $phone = $_POST['t_phone'];
    $sql = "INSERT INTO Trainers (Fname, Lname, email, PhoneNumber) VALUES ('$fname', '$lname',
    '$email', '$phone')";
    $result = mysqli_query($conn, $sql);
    if ($result){
        echo "New trainer record successfully created! Please go back to the previous page to add another
        record.";
    }
    else{
        echo "Error: " . $sql . "<br>" . mysqli_error($conn);
    }
}

mysqli_close($conn);
?>

```

(e)

Data submitted by IP address: 192.168.13.1

(f)

20+ DATABASE ENTRIES ADDED SUCCESSFULLY THROUGH THE WEB FORM/PHP.

NOTE: The Memberships table by definition (of my design) only contains 3 different memberships, thus it is only a 3 row table. All the other tables have been inserted with 20+ rows.

CUSTOMERS TABLE:

Showing rows 0 - 39 (40 total, Query took 0.0007 sec)

SELECT * FROM Customers

Profiling [Inline] [Edit]

Show: Start row: 0 Number of rows: 30 Headers every 100 rows

Sort by key: None

	CustomerID	Fname	Lname	gender	email	PhoneNumber	BirthDay
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	Daniel	Smith	Male	danismith@gmail.com	345-729-3839	1996-02-02
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	Daniel	Ricardo	Male	danirichi@gmail.com	234-893-9184	1995-01-25
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	Daniel	Morales	Male	danimorales@gmail.com	383-481-4810	1990-02-25
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	Rachal	Oliveira	Female	racheloliveira@gmail.com	844-824-8194	1985-02-15
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	Brandon	Marlon	Male	brandonmarlon@gmail.com	844-824-8191	1983-02-02
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	9	Peter	Langula	Male	peterlangula@gmail.com	844-824-8191	1991-04-25
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	10	Trevis	Harrison	Male	trevissharrison@gmail.com	412-523-4124	1989-03-20
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	11	Anna	Smith	Female	annasmith@gmail.com	412-523-4150	1989-03-20
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	12	Harrison	Ford	Male	harrisonford@gmail.com	412-523-9481	1950-06-20
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	13	Travis	Mozart	Male	travismozi@gmail.com	412-523-8491	1979-06-20
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	14	Isabella	Alcantra	Female	isabellaalca@gmail.com	412-523-8194	1995-10-10
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	15	Ruth	Videla	Female	ruthvidela@gmail.com	456-523-8194	1998-11-10

3. R + SQL

(a)

```
> library(rjson)
> fromJSON(readLines("http://api.hostip.info/get_json.php", warn=F))$ip
[1] "177.68.13.243"
```

(b)

Query 1: What are the customer ID's of those customers who are enrolled in a membership of type 'Gold'?:

```
sql_b1 <- "SELECT Customers.Fname AS FirstName, Customers.Lname AS LastName,
Orders.PaymentDay AS PaymentDay FROM Customers, Orders WHERE
Customers.CustomerID=Orders.CustomerID AND Customers.gender='Female'"
```

Query 2: What are the first and last names of the customers who attend boxing class?

```
sqlb2 <- "SELECT Fname, Lname FROM Customers, Attend, Classes WHERE
Customers.CustomerID=Attend.CustomerID AND Attend.ClassID=Class.ClassID AND
Class.name='Boxing'"
```

(c)

Query 1: Display the Class ID and name of all the classes that are being attended by at least 5 customers, and the number of customers attending it. Order results by descending class attendance.

```
sqlc1 <- "SELECT ClassID, name, COUNT(a.ClassID) AS NumberCustomers FROM Classes
AS cl, Attend AS a WHERE a.ClassID=cl.ClassID GROUP BY a.ClassID HAVING
NumberCustomers >= 5 ORDER BY NumberCustomers DESC"
```

Query 2: Display the number of male customers enrolled in each of the three membership types.

```
sqlc2 <- "SELECT Memberships.type AS type, COUNT(Memberships.type) AS Num,
Customers.gender AS gender FROM Customers, Orders, Memberships WHERE
Customers.CustomerID=Orders.CustomerID AND
Orders.MembershipID=Memberships.MembershipID GROUP BY type HAVING
Customers.gender='Male'"
```

(d)

Query 1: Display each memberships type, price per month, order day and the payment day (day of month).

```
sqld1 <- "SELECT Memberships.type, Memberships.PricePerMonth, Orders.OrderDay,
Orders.PaymentDay FROM Memberships INNER JOIN Orders ON
Memberships.MembershipID=Orders.MembershipID"
```

Query 2: Display each customers First name, Last name, gender and the date at which they purchased their membership.

```
sqld2 <- "SELECT Customers.Fname, Customers.Lname, Customers.gender, Orders.OrderDay
FROM Customers INNER JOIN Orders ON Customers.CustomerID=Orders.CustomerID"
```

(e)

Query 1: Display the first and last names of all the customers whose age is larger than the average age of all the customers.

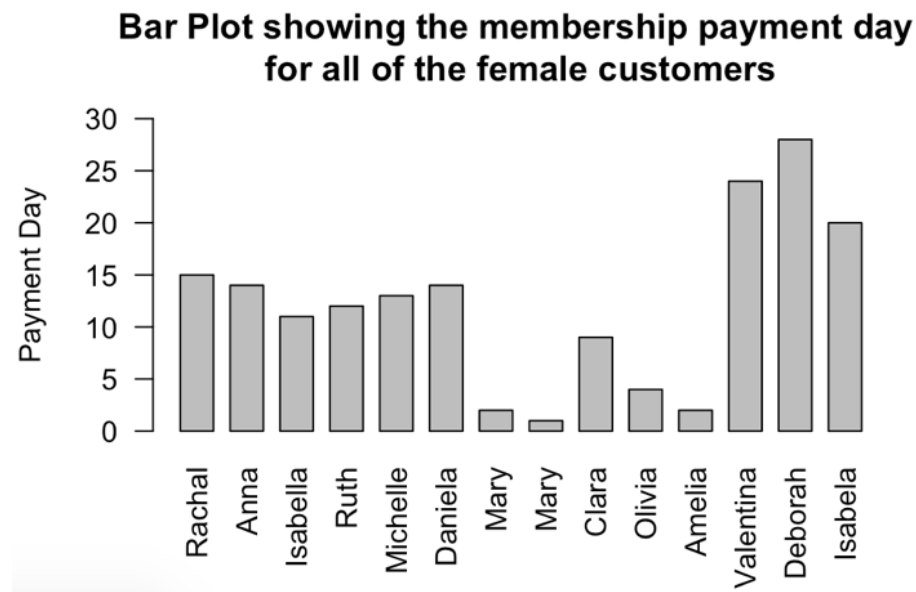
```
sql_e1 <- "SELECT Fname, Lname, DATEDIFF('2020-03-28', Customers.BirthDay)/365.25
AS Age FROM Customers WHERE DATEDIFF('2020-03-28', Customers.BirthDay)/365.25 >
(SELECT AVG(DATEDIFF('2020-03-28', Customers.BirthDay)/365.25) AS age FROM
Customers)"
```

Query 2: Display the first names, last names, gender and Payment Day of the customers who have a due Payment Day larger than the payment day of customer whose customer ID = 25

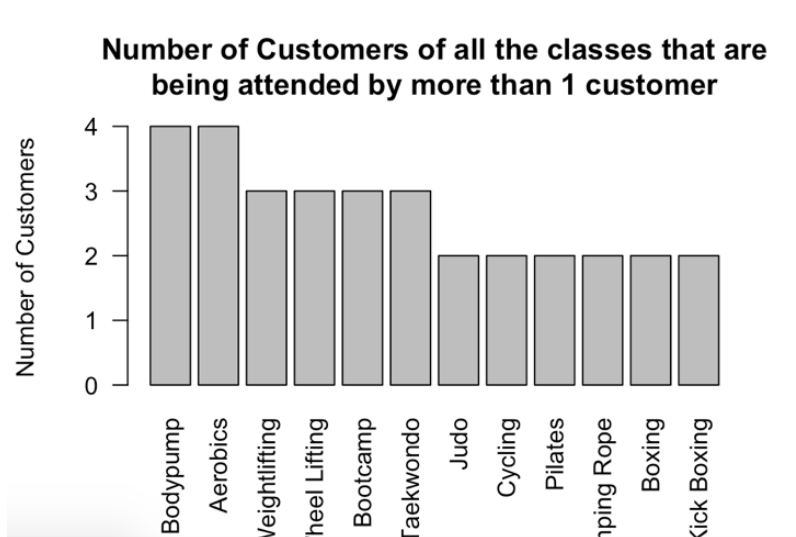
```
sql_e2 <- "SELECT Fname, Lname, gender, Orders.PaymentDay AS PaymentDay FROM
Orders, Customers WHERE Customers.CustomerID=Orders.CustomerID AND
Orders.PaymentDay > (SELECT Orders.PaymentDay FROM Orders WHERE
Orders.CustomerID=00025)"
```

(f)

Plot for b (Query 1): `plot_b <- barplot(height=data_b1$PaymentDay, names.arg=data_b1$FirstName, las=2, space=0.5, main="Bar Plot showing the membership payment day for all of the female customers", ylab="Payment Day", ylim=c(0,30))`

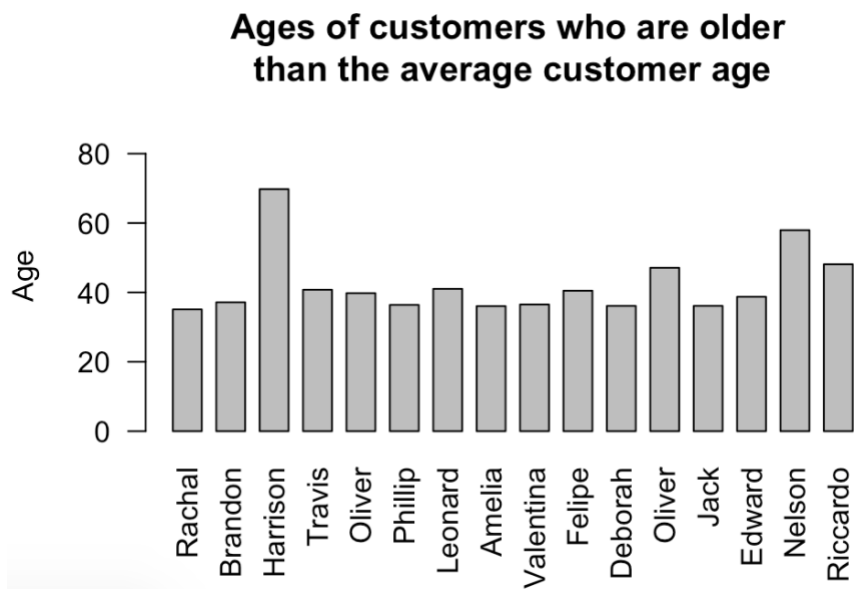


Plot for c (Query 1): `plot_c <- barplot(height=data_c1$NumberCustomers, names.arg=data_c1$ClassName, las=2, main="Number of Customers of all the classes that are being attended by more than 1 customer", ylab="Number of Customers")`



Plot for d (Query):

Plot for e (Query 1): `plot_e <- barplot(height=data_e1$Age, names.arg=data_e1$Fname, las=2, space=0.5, ylab="Age", main="Ages of customers who are older than the average customer age", ylim=c(0,90))`



REFERENCES

<https://www.w3schools.com/default.asp>

<https://www.geeksforgeeks.org/form-validation-using-html-javascript/>