CS 3380 Lab Assignment 6

1 Directions and Submission Instructions

This assignment must be completed by **Sunday, October 4th at 11:59 PM**. You must upload your PHP code to Blackboard. The uploaded file should be named `lab6.php`. Your code must also be hosted on your Azure VM. Your lab 6 submission should be reachable and functional through the following URL:

`http://cs3380-YOURPAWPRINT.cloudapp.net/lab6/lab6.php`

If that URL does not work you will lose points. When submitting your lab6.php file via Blackboard, please include your URL in the submission comments. (This greatly helps the TAs during grading.) Late submissions, either for the files or the URL, will not be accepted.

You **should not** submit the `data.sql` file.

2 Goals

- Creating SQL views
- Writing queries with sub-queries
- Using built-in SQL functions
- Writing queries with the SQL set operators

3 Tasks

3.1 Download

Begin by downloading a SQL file needed for this assignment by executing the following commands in your terminal:

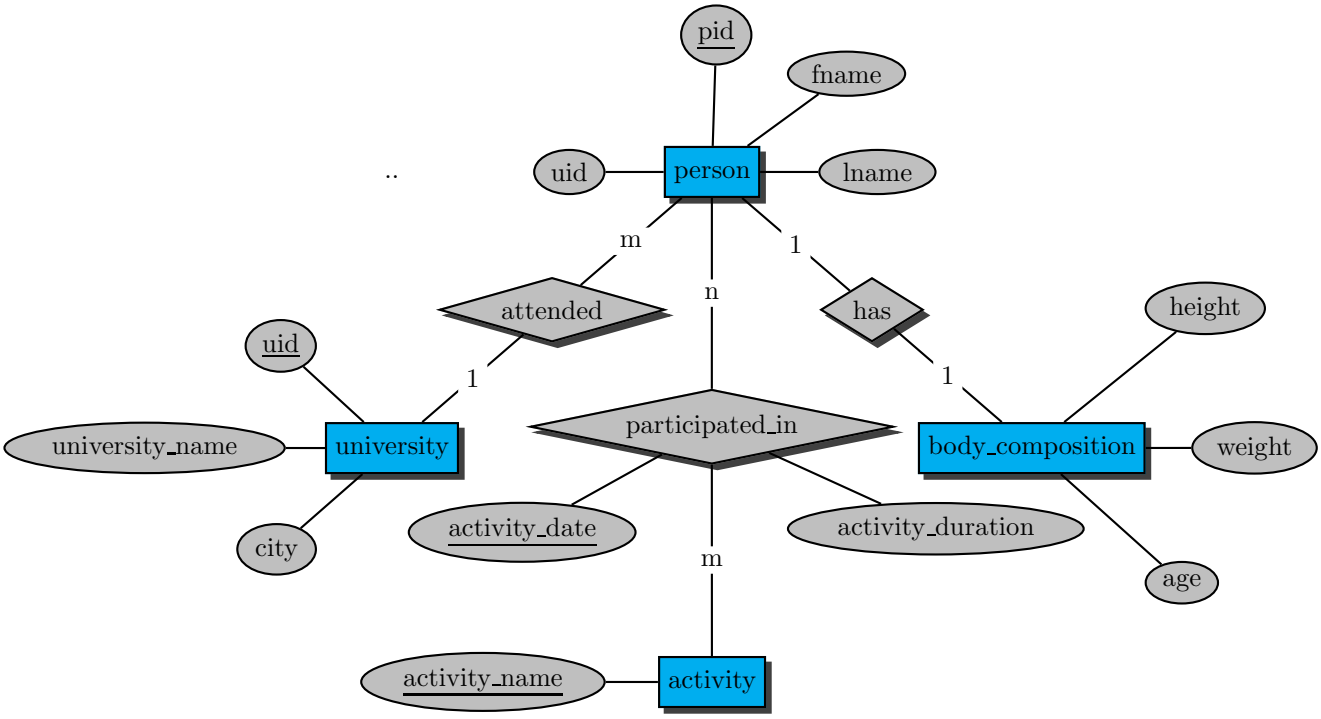
```
sudo mkdir /var/www/html/lab6
cd /var/www/html/lab6
sudo wget http://cs3380-mapp86.cloudapp.net/lab6/data.sql
```

Note that you might not be able to copy-paste the above commands. You may need to type them manually into your terminal.

Next, run data.sql file with the `mysql` command to import the dataset.

3.2 Inspect the Data

The `data.sql` file will create five tables within your database. This dataset stores (fictional) information about students that took part in a study on body composition and activity from a variety of universities. An ERD for this dataset follows.



3.3 Implementation

You will be responsible for creating a PHP script that allows a user to execute one of the following 8 queries by selecting from a drop down box and clicking a button. The code should connect to the database and execute the appropriate query based on what is selected in the drop down box.

Be smart about how you code this. It should not take a large amount of PHP code to complete this assignment. For example, my code for this assignment is around 125 lines. You could write a PHP function that accepts an string containing a SQL statement, execute it and write out the HTML table. Your PHP page must show the number of records returned and a table of query results for the following 8 queries.

Hint: If you haven't caught on already, you should recycle your PHP code from lab 5.

3.3.1 Create a View: Part 1

Create a view that shows the person's id (pid), first name (fname) and last name (lname) for all people who have a body weight above 140. This view should be named "weight" (without the quotes). You must use an `INNER JOIN` in the views query. Your PHP page should then query the view (i.e. `SELECT * FROM weight`). *(8 rows)*

3.3.2 Create a View: Part 2

Create a view that returns the first name (fname), last name (lname) and BMI for people with a weight above 150. This view should be named "BMI". You must use an `INNER JOIN` and you must reference the "weight" view created in 3.3.1. BMI is calculated as

$$703 \cdot \frac{weight}{height^2} \tag{1}$$

In this view, round the BMI value to the nearest whole number. For example, a person with a height of 71 inches and weight of 145 lbs would have a BMI of 20.2 which would be rounded to 20. Use an SQL function

to achieve this rounded result. Your PHP page should then query the view (i.e. `SELECT * FROM bmi`). *(~~12~~ 8 rows)*

3.3.3 Using `EXISTS`

Write a query that shows returns the name and city of the university that has no people in database that are associated with it. Your query **must** use `EXISTS` to achieve. *(2 rows)*

3.3.4 Using `IN`

Write a query that returns only the uid value for all universities in the city Columbia. Then use that query with an `IN` sub-query expression to retrieve the first and last names for all people that go to school in Columbia. *(4 rows)*

3.3.5 Using `NOT IN`

Write a query that returns all of the activities with records in the `participated_in` table. Then use that query with a `NOT IN` sub-query expression to retrieve the activities that are not played by any player in the database. *(2 rows)*

3.3.6 Using `UNION`

Write a query that returns the pid of all people listed in `participated_in` that participate in 'running'. Then modify your query to use `UNION` to return all people who run or play racquetball. You **must** use the `UNION` operator to accomplish this. You **cannot** use `OR`. *(5 Rows)*

3.3.7 Using `INTERSECTS`

Write a query that returns the first and last name of all people listed in `body_composition` table who are older than 30 years old. Then modify your query to use `INTERSECTS` to return all people who are older than 30 and are taller than 65 inches. You **must** use the `INTERSECTS` operator to accomplish this. You **cannot** use `AND`. *(3 rows)*

3.3.8 Using `ORDER BY`

Write a query that returns peoples first and last names weight, height, and age. Records should be ordered first by height in descending (Z-to-A order), then by weight in ascending order, and finally by the person's last name in ascending order. *(12 rows)*