***Note: This document was updated on 3/28/22 to address permissions issue for links in problem 1.***

**Homework 5: Advanced SQL (Total Points: 100)**

Due: April 8, 11:59PM ET

**Problem 1: Create a DB and load data (Points: 25)**

1.1. First create the database and load the sample data we are providing. To make your job easier, we have provided all the SQL statements needed to accomplish this in the files [create\_objects.sql](https://e59-spring-2022-hw5-sql-files.s3.amazonaws.com/create_objects.sql) and [load\_data.sql](https://e59-spring-2022-hw5-sql-files.s3.amazonaws.com/load_data.sql). Download these files using the preceding Canvas links, and upload them to your EC2 instance to a location/folder where you have read, write, and execute privileges. In section, we will cover in full detail how to transfer the files and change folder/file permissions so that the scripts are executable.

Start mysql from the directory where you placed the files and execute the following commands at the mysql prompt in this order:

source create\_objects.sql;

source load\_data.sql;

After each one of these commands you should see multiple messages like this (the number of rows affected may be 0 or 1):

Query OK, 1 row affected (0.01 sec)

***Note:*** *These scripts were sourced from* [*https://www.sqlservertutorial.net/sql-server-sample-database/*](https://www.sqlservertutorial.net/sql-server-sample-database/) *and adapted for MySQL.*

No need to share any screenshots for this portion; we will know you were successful if you are able to complete the next set of tasks. (5 points)

1.2. Execute the following commands:

DESCRIBE TABLE orders; (5 points)  
DESCRIBE TABLE order\_items; (5 points)  
DESCRIBE TABLE staffs; (5 points)  
SELECT COUNT(\*) FROM customers; (5 points)

and provide screenshots of the results:

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

**Problem 2: Retrieve order data by created date (Points: 25)**

Create a SELECT statement that lists the first\_name, last\_name and email of any customer that has placed an order between March 1, 2018 and March 10, 2018 **inclusive**.

Sort the results by first\_name and last\_name in ascending order.

2.1. Provide the SQL statement used. (15 points)

|  |
| --- |
| SELECT  first\_name, last\_name, email  FROM  customers  JOIN  orders ON orders.customer\_id = customers.customer\_id  WHERE  order\_date >= '2018-03-01'  AND order\_date <= '2018-03-10'  ORDER BY first\_name , last\_name ASC; |

2.2. Provide a screenshot showing the last 10 rows of the results obtained by the SQL statement shown above. Make sure it includes the line that indicates how many total rows were returned. (10 points)

|  |
| --- |
|  |

**Problem 3: Retrieve order data by creation date and location (Points: 25)**

Obtain the list of orders placed in California between March 1, 2018 and March 10, 2018 inclusive, and the orders placed in New York between March 11, 2018 and March 15, 2018 inclusive. List the first\_name, last\_name and email of the user, along with the id and date of the order and the store name and state.

3.1. Provide the SQL statement used. (15 points)

|  |
| --- |
| SELECT  customers.first\_name,  customers.last\_name,  customers.email,  order\_id,  order\_date,  store\_name,  t.state  FROM  (SELECT  orders.\*, stores.store\_name, stores.state  FROM  orders  JOIN stores ON stores.store\_id = orders.store\_id  WHERE  stores.state = 'CA'  AND orders.order\_date BETWEEN '2018-03-01' AND '2018-03-10' UNION SELECT  orders.\*, stores.store\_name, stores.state  FROM  orders  JOIN stores ON stores.store\_id = orders.store\_id  WHERE  stores.state = 'NY'  AND orders.order\_date BETWEEN '2018-03-11' AND '2018-03-15') t  JOIN  customers ON t.customer\_id = customers.customer\_id; |

3.2. Provide a screenshot showing the full result set. Make sure it includes the line that indicates how many rows were returned. (10 points)

|  |
| --- |
|  |

**Problem 4: Prepare complete contact list for Marketing Team (Points: 15)**

In preparation for an upcoming grand opening event for a new BikeStores location, the marketing team has tasked you with putting together a SINGLE list of all customers and employees. The list should contain the following: id, person’s full name, last\_name, first\_name, email, phone number, and a flag designating if the person is a customer (flag = ‘Y’ or ‘N’). Additionally, the list should be ordered in ascending order by person last\_name. (Hint: Ensure that you only include active employees)

The report headers should be labeled as such:

* person\_id
* person\_full\_name
* person\_last\_name
* person\_first\_name
* person\_email
* person\_phone\_no
* person\_is\_customer

4.1 Provide the SQL statement used. (10 points)

|  |
| --- |
| SELECT  staff\_id AS person\_id,  CONCAT(first\_name, ' ', last\_name) AS person\_full\_name,  last\_name AS person\_last\_name,  first\_name AS person\_first\_name,  email AS person\_email,  phone AS person\_phone\_no,  'N' AS person\_is\_customer  FROM  staffs  WHERE  active = 1  UNION SELECT  customer\_id AS person\_id,  CONCAT(first\_name, ' ', last\_name) AS person\_full\_name,  last\_name AS person\_last\_name,  first\_name AS person\_first\_name,  email AS person\_email,  phone AS person\_phone\_no,  'Y' AS person\_is\_customer  FROM  customers  ORDER BY person\_last\_name; |

4.2 Provide a screenshot of the header row with column names and the last 10 rows from the results set. Ensure the line indicating the “rows in set” is visible. (5 points)

|  |
| --- |
|  |

**Problem 5: Manipulating query output that contains NULL values (Points: 10)**

This problem involves the use of a MySQL function to substitute a string for NULL values in the query result set (Hint: Greg covered this as part of lecture). Query the customers table for the first 5 records (customer\_id 1-5) – use an ORDER BY statement to properly display records. Include the following fields in your result set: customer\_id, first\_name, last\_name, phone, and email. For the phone field, if a customer does not have a phone number (e.g. NULL), instead of NULL in the result set, you should display ‘not on file’.

5.1 Provide the SQL statement used. (5 points)

|  |
| --- |
| SELECT  customer\_id,  first\_name,  last\_name,  IFNULL(phone, 'not on file') AS phone,  email  FROM  customers  ORDER BY customer\_id  LIMIT 5; |

5.2 Provide a screenshot of the entire results set. (5 points)

|  |
| --- |
|  |

**Bonus – Problem 6: Retrieve a staff report (Points: 10)**

Create a SQL statement that shows the first\_name, last\_name, store\_id, email, phone of every manager as well as the number of active staff members reporting to them (tip: A manager for a store reports to Fabiola Jackson). Note: If you wish to include Fabiola Jackon in the results set, you are free to do so, but we will not deduct any points if you do not.  
  
6.1. Provide the SQL statement used. (5 points)

|  |
| --- |
| SELECT  s1.first\_name,  s1.last\_name,  s1.store\_id,  s1.email,  s1.phone,  COUNT(s2.staff\_id) AS num\_active\_staffs  FROM  staffs s1  JOIN  staffs s2 ON s1.staff\_id = s2.manager\_id  GROUP BY s1.first\_name , s1.last\_name , s1.store\_id , s1.email , s1.phone; |

6.2. Provide a screenshot showing the full result set. Make sure it includes the line that indicates how many rows were returned. (5 points)

|  |
| --- |
|  |