

CS 2316 – Recitation Assignment

SQL Practice 2

Instructions:

In this assignment, you will **NOT write any python** code. (Unless you want to...but don't turn it in!) Instead, you will turn in SQL statements in a text file (use Notepad, or some other text editing program). *Be sure to place your name, email address, and collaboration statement at the top of your text file.* Each of the problems below will be solved with a single SQL statement (Although the statements may extend to multiple lines as the problems get harder.) You will be provided with a description of the data in tables, as well as a question to answer about that data. Your answer to each question will be the SQL statement that finds or calculates the answer to the question or task. Number your answers to match the question numbers in this document.

Note that you are not submitting factual answers to the questions in this homework. Instead, you are submitting SQL statements that cause the DB engine to GENERATE the answers to the questions asked in this homework.

To receive credit, your SQL statement must execute properly on the class database (Which runs MySQL). Note that we encourage you to test your statements before submission by running them interactively on the myPHPAdmin web interface to the class DB located at:

<http://academic-mysql.cc.gatech.edu>

The orders Table:

```
CREATE TABLE orders
  (cid INTEGER AUTO_INCREMENT UNIQUE NOT NULL,
  customer TEXT NOT NULL,
  date DATE,
  payment DOUBLE,
  profit DOUBLE,
  profitPercent DOUBLE,
  numitems INTEGER)
```

An example:

| cid | customer | date | payment | profit | profitPercent | numitems |
|-----|----------|----------|---------|--------|---------------|----------|
| 1 | "Summet" | 11/11/13 | 245.32 | 34.2 | NULL | 5 |
| 2 | "Sole" | 10/05/12 | 1243.33 | 240.8 | NULL | 40 |
| 3 | "Barr" | 02/30/12 | 804.32 | 182.3 | NULL | 5 |
| 4 | "Garcia" | 04/21/10 | 32.85 | 8.32 | NULL | 1 |

SQL Questions:

1. (2 points) Return the customer names and dates (only) of any orders that earned a profit above \$25 dollars.
2. (2 points) Return the customer names (only) who have placed at least one order where the payment was over \$500. Make sure *not to duplicate* any customer names! *Display the customer names alphabetically!*
3. (2 points) Change all orders made yesterday so that their profit field is set to zero. (Encode yesterday as a literal date in your SQL statement.)

Advanced SQL Questions:

4. (4 points) Return the total amount of money received for all orders (Sum of all payments).
5. (4 points) Return the average profit for all orders.
6. (5 points) Return all columns for orders which had greater than average profit. NOTE: Your SQL statement may NOT include a literal number. You must calculate the average profit using an embedded SQL statement! Sort the output in *reverse numerical order* based upon the profit (i.e. the most profitable order(s) will be first)!
7. (5 points) Run a report that returns how much profit was made on each particular day. Order the results by day (descending, so the most recent day is at the top). Your report should have the date and sum profit (only).