

CS 1803

Individual Homework 9 – SQL Practice

Due: Friday, April 8th, before 6 PM

Out of 100 points

Files to submit: 1. HW9.TXT

This is an **INDIVIDUAL** assignment!

Students may only collaborate with fellow students currently taking CS 1803, the TA's and the lecturer. Collaboration means talking through problems, assisting with debugging, explaining a concept, etc. You should not show another student your SQL statements, exchange SQL statements or write SQL statements for others.

For Help:

- TA Helpdesk – Schedule posted on class website.
- Email TA's or use T-Square Forums

Notes:

- **Don't forget to include the required comments and collaboration statement (as outlined on the course syllabus).**
 - **Do not wait until the last minute** to do this assignment in case you run into problems.
-

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 people Table:

```
CREATE TABLE people
  (id INTEGER PRIMARY KEY AUTO_INCREMENT UNIQUE NOT NULL,
  name TEXT NOT NULL,
  email TEXT,
  phone TEXT)
```

Example:

id	name	email	phone
1	"Jay Summet"	"summetj@gatech.edu"	"111-222-3333"
2	"Sally Smith"	"ssmith@gatech.edu"	"111-222-3333"

Warm-up Problems:

1. (3 points) Add a new person, with your name and any email address you choose (do NOT specify a phone number) into the table (Allow the DB to generate the ID!).
2. (3 points) Return (only) the name and email address from the people table where the phone number is exactly "404-111-2222".
3. (3 points) Return (only) the name and email address from the people table where the phone number starts with "404".
4. (3 points) Return all columns of all records where the phone number field contains "007" in any location.
5. (3 points) Change the email of any record with a name of "Jay Summet" to "summetj" (without a @gatech.edu at the end).
6. (3 points) Remove all records from the people table where the email address ends with "uga.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

Basic SQL Questions:

7. (5 points) Return the customer names and dates (only) of any orders that earned a profit above \$25 dollars.

8. (5 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!*

9. (5 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. Note that if you are doing this homework early, your yesterday may be earlier than the day before the homework is due. This indicates that you are responsible, and are likely to be able to finish the homework without problems.)

Advanced SQL Questions:

10. (8 points) Return the total amount of money received for all orders (Sum of all payments).

11. (8 points) Return the average profit for all orders.

12. (10 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)!

13. (10 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).

14. (10 points) The Cut Rate Web Development (Inc) company hired by your boss did not properly fill in the profitPercent column. As created, the database has no profit percent information. (Unless another student has completed problem 15 before you!) To remove all profit percent information (restoring that column to the default setting) you may issue the following SQL command: **UPDATE orders SET profitPercent = NULL**

For each order, return the date, customer, and a calculated percent profit. You may not use the percentProfit field in the database. Instead, you must calculate the percent profit using only the payment and the profit columns.

15. (10 points) After realizing that the Cut Rate Web (inc) company he hired to handle the order entry website failed to fill in this column, your boss has asked you to calculate and fill in the correct data using SQL! Write an SQL command that will both calculate the profit percent (using the profit and payment columns) AND insert the correct data into the database!

After you figure this out, please help out any students after you by removing all profit percent information (restoring that column to the default setting) in the database by issuing the following SQL command: **UPDATE orders SET profitPercent = NULL**

Grading

You will earn points as follows for each statement that works correctly according to the specifications in each problem description.

Questions 1-6 (3 points each)	18 points
Questions 7-9 (5 points each)	15 points
Questions 10-11 (8 points each)	16 points
Questions 12-15 (10 points each)	40 points
Properly Submitting a plain text file:	11 points
Total Possible:	100 points

Miscellaneous Penalties:

Upper/Lower case issue with a table/column name:	-2 points
SQL produces correct answer, but with extra columns:	-3 points
SQL produces correct answer, sorted incorrectly:	-2 points
Submitted SQL corrupts the database:	-25 points
Lack of a collaboration statement:	-100 points