Assignment 1

CS 348 Fall 2014

Background

The goal of this assignment is to develop your skills in writing queries to access data within a relational database. The assignment covers two different query languages: Relational Algebra and SQL. For the schema and sample data we will use a small instance of a TPC-H database, which simulates a business scenario involving a purchase-order/invoicing system for a generic parts inventory control system.

You are highly encouraged to complete Assignment 0 before starting this assignment and using your own computing resources to test and debug your queries with different datasets. Having your own database server will allow you to modify the data and test your queries with different database instances to better check the correctness of results.

General Instructions

1. The TPC-H database schema contains the following eight relations: PART, SUPPLIER, PARTSUPP, CUSTOMER, NATION, REGION, LINEITEM, ORDERS. For a more detailed description of the schema (including all attribute names and integrity constraints), please consult the SQL DDL and data model available from the course webpage.
2. This assignment is to be performed individually. Sharing either RA/SQL statements or query results with other students will be considered cheating.
3. The best way to start this assignment is to get your own environment ready (Assignment 0) and then play around with a few queries to get a feel for how SQL queries are structured, before even getting to answering questions.
4. Remember that a query that returns correct results on one instance of a database does not mean it will return correct results for all instances of the same database. The sample data provided (or that you create) to test queries may not capture all boundary conditions.
5. The best advice for handling this assignment is that you get started early.

What to Submit

You must submit **both** (i) hardcopy solutions to **ALL** questions in this assignment including SQL queries and (ii) electronically submit solutions of Part B of this assignment using the undergraduate CS assignment submission system. **Failing to submit either means an automatic "zero" on this assignment.**

You must use the coversheet provided with this assignment to submit the solutions.

You MUST deposit solutions, with the completed coversheet attached to the front of your assignment, in the appropriately labeled box on the 4th floor of MC before the deadline.

For Part B, your hardcopy solution should include the SQL statement neatly formatted. As is good coding practice you should assume that your query will be read by a human.

YOU MUST FORMAT THE PRINTED QUERIES NEATLY. Marks may be deducted for unreadable and/or badly formatted queries.

For each query, you must submit a separate file named "Q#.sql" that contains *only* a single SQL query. You may not create additional tables or views for use by these queries.

Queries (and thus each file) must end with a semicolon (;)

There are no marks for efficiency but none of the queries should take more than 30 seconds to execute on most systems/datasets.

Make sure to comment complex portions of your SQL queries in the written solutions.

Do not submit queries that crash, are syntactically incorrect or fetch a un-necessarily large number of rows to show your “attempt.” Submitting a query that results in an error message will get ZERO marks automatically and the marker will not read your written solutions for an explanation.

Electronic submissions have to be done using the **`submit`** system in the undergrad environment (type **`man submit`** for help). You have used this system in previous undergrad courses as well.

Specific Instructions for Part B:

1. You have two ways in which you can test your queries
    i. Use your own resources (database server or pre-setup virtual machine) - See Assignment 0 for details. The virtual machine provided comes with a 100MB randomized dataset for TPC-H.
    ii. Use the pre-setup 10MB TPC-H database db2 database hosted at: https://www.student.cs.uwaterloo.ca/~cs338/webdb/cs338_mysql.php?db=tpch. Note that with this option, you will not be able to modify the database, add/remove rows etc. and so comprehensively testing your queries will not be possible, see point 2 below. Thus option (i) is highly recommended.
2. Your queries must work over *all* possible database instances that conform to the schema, not just the sample data provided (i.e., your query shouldn't "hardcode" assumptions about the database instance).
3. For SQL, the output for all queries should be sorted by the attributes as indicated in the query, see example below.

Example Query 0

List the name and phone number of supplier number 1.

Example Solution (Q0.sql)

```
SELECT s_name, s_phone
FROM SUPPLIER
WHERE s_suppkey = 1
ORDER BY s_name, s_phone
;
```

Part A: Relational Algebra [25 marks]

Write Queries one through four listed below in relational algebra. Mark division is as follows: 1, 2, 3 [5 Marks] and 4 [10 Marks].

Part B: SQL [75 marks]

Write ALL queries listed below in SQL.
Mark Division:
1, 2, 3, 4, 5 [5 Marks]
6, 7, 8, 9, 10 [10 Marks]

For SQL part, sort the query results using the "ORDER BY" clause based on the attribute names specified (refer to the above example). You are allowed rename the column names in the data to names you are asked to generate.

List of Queries

Query 1

A customer wants to know what parts are available between the sizes of 43 through 47 (including 43 and 47) of brand 'Brand#42'. Write a query to retrieve all such parts. The output should have a single column p_name, and is ordered by p_name.

Query 2

The collection team in our Canadian office wants to know the custkeys, names and phone numbers of all local customers that owe us money so that they can harass these customers over the phone. Write a query that will retrieve all custkeys, names, and phone numbers of customers in the nation of 'CANADA' that have an account balance strictly less than zero. The output should have the column names: c_custkey, c_name, c_phone and is sorted in this order.

Query 3

Management wants to improve client relations, with both customers and suppliers, who trust us the most. They want a list of all names and phone numbers belonging to either customers that have an account balance strictly greater than 9975 or suppliers that have an account balance strictly greater than 9500. Write a single statement to retrieve these required names and phone numbers. The output should have the column names: name, phone and and is sorted in this order.

Query 4

Management wants to know which countries are well represented by our network of suppliers. Write a statement to list the name of each nation that contains at least two different suppliers.

Now let us suppose that instead of at least two different suppliers, management wants to know the names of each nation that has at least seven different suppliers. In your hard copy solutions explain (describe in less than a paragraph of text, no need to give actual queries) what would be the best way change your original answer, in both RA and SQL, to accommodate for this change in specifications. The output should consist of the column name: n_name and is ordered by the same.

Query 5

For each region and for each nation located in that region, list the region name and nation name. For regions not containing a nation, the output should contain a single tuple with the region name and a NULL nation name. The output should consist of the column names: region, nation and is ordered by these columns.

Query 6

Management is concerned about the sheer number of customers who open an account using our website, but never buy anything. They want to know how many such customers exist in our database. Retrieve a single tuple with a single attribute (named ‘Total’) which will represent the total number of customers in our database that have never placed an order.

Query 7

Management wants to ensure that our best customer in every country is taken care of by specially trained customer representatives. For each nation list the nation name, customer name and the largest account balance of the customer located in that nation. For this question you can assume that every nation contains at least one customer that has at least one order.

In your physical submission explain whether your solution will work when the assumption that every nation contains at least one customer that has at least one order is taken away. The output consists of column names: nation_name, c_name, acct_balance and is ordered by the columns in that order.

Query 8

For each nation, list the nation key, nation name, the number of orders placed by customers living in that nation, and the cumulative price for all of those orders cast as a DECIMAL(13,2). The result should have the column names : n_key, n_name, num_orders, cum_price and is ordered by descending cumulative order price (cum_price).

Query 9

The management team wants a list of nations whose sales have been underperforming. Print out the list of nation names along with the total number of orders placed by

customers from those nations but only for nations that have below average number of orders compared to all nations.

Note: You may not hard code the average number of orders per nation in the query! Your query must find the average number of orders per nation and then compare it to the totals for each nation to create a list of nations underperforming the average sales figure.

Hint: You should first write a query that finds out the average number of sales per nation. Then wrap it in a larger query to get the final answer.

Your output should have columns : nation_name, total_orders and is ordered by nation_name, total_orders.

Query 10

For each customer market segment, list the:

1. market segment
2. cumulative (total) order price for orders of status 'F' made by customers in that market segment
3. cumulative (total) order price for orders of status 'O' made by customers in that market segment
4. cumulative (total) order price for orders of status 'P' made by customers in that market segment

The totals should be cast as DECIMAL(13,2) and the output should consist of column names: c_mktsegment, total_F, total_O, and total_P, respectively. Order the result by c_mktsegment.

Hint: Try to use the divide and conquer approach and think of dividing the task into smaller tasks. Then use the results as intermediate tables to build a larger query! Note that this is a difficult question and there are many ways of handling this task. You are free to read up on external resources and utilize SQL (MySQL) commands/syntax not taught in the course. Part of this exercise is for you to discover the most suitable approach and method available in MySQL to generate the report requested.