Programming Assignment 2, CMSC724, Spring 3012

Instructor: Amol Deshpande

The assignment is to be done by yourself. Due date: Friday March 8.

This assignment has two parts: (1) write a simple client program in java that accesses the database using embedded SQL; (2) understand the ANALYZE capabilities provided by PostgreSQL. We will use the PostgreSQL database system for this assignment.

Submission: a single hardcopy containing everything would be preferred, but you can also send me a single email with the Java program, and any pdf or text files as separate attachments (please do not submit a zip file).

1 PART 1 [2 pts]

One of more prominent ways to use a database system is using an external client, using APIs such as ODBC and JDBC. This allows you to run queries against the database and access the results from within say a Java program.

Here are some useful links:
`http://en.wikipedia.org/wiki/Java_Database_Connectivity`
`http://www.mkyong.com/java/how-do-connect-to-postgresql-with-jdbc-driver-java/`
`http://jdbc.postgresql.org/index.html`

The last link has detailed examples in the "documentation" section.

Your Task: Write a Java program to do the same thing as Question 4 from the SQL assignment:

```
4. For 2004 Olympics, generate a list - (birthyear, num-players, num-gold-medals) -
containing the years in which the atheletes were born, the number of players
born in each of those years who won at least one gold medal, and the number of gold
medals won by the players born in that year.
```

... with the constraint that you can only issue a simple query to the database that does the join between the three relations. Specifically, you should only issue the following query to the database, rest of the processing should be done in the Java program:

```
select p.birthdate, p.player_id
from results r, events e, players p
where p.player_id = r.player_id and e.event_id = r.event_id
      and e.olympic_id = 'ATH2004'and r.medal = 'GOLD'
```

You should submit the Java program, and the answer itself should be submitted along with the rest of the assignment.

2 PART 2 [0.5+0.5+0.5+0.5+1=3 pts]

Like most other database systems, PostgreSQL allows you to analyze query execution plans, and tune the execution. The main PostgreSQL commands are EXPLAIN and EXPLAIN ANALYZE. The first one simply prints out the plan, but the second one also runs the query and compares the estimated values with actual values observed during execution.

An important accompanying command is "VACUUM ANALYZE"; this recalculates all the statistics used by PostgreSQL optimizer.

The assignment here is to answer 5 questions using these tools. The questions are present in the accompanying "assignment-analyze.sql" file. Most of the questions use a modified TPC-H Benchmark database, more details are available at: `http://www.tpc.org/tpch`. I am reproducing the schema of the dataset on the next page.

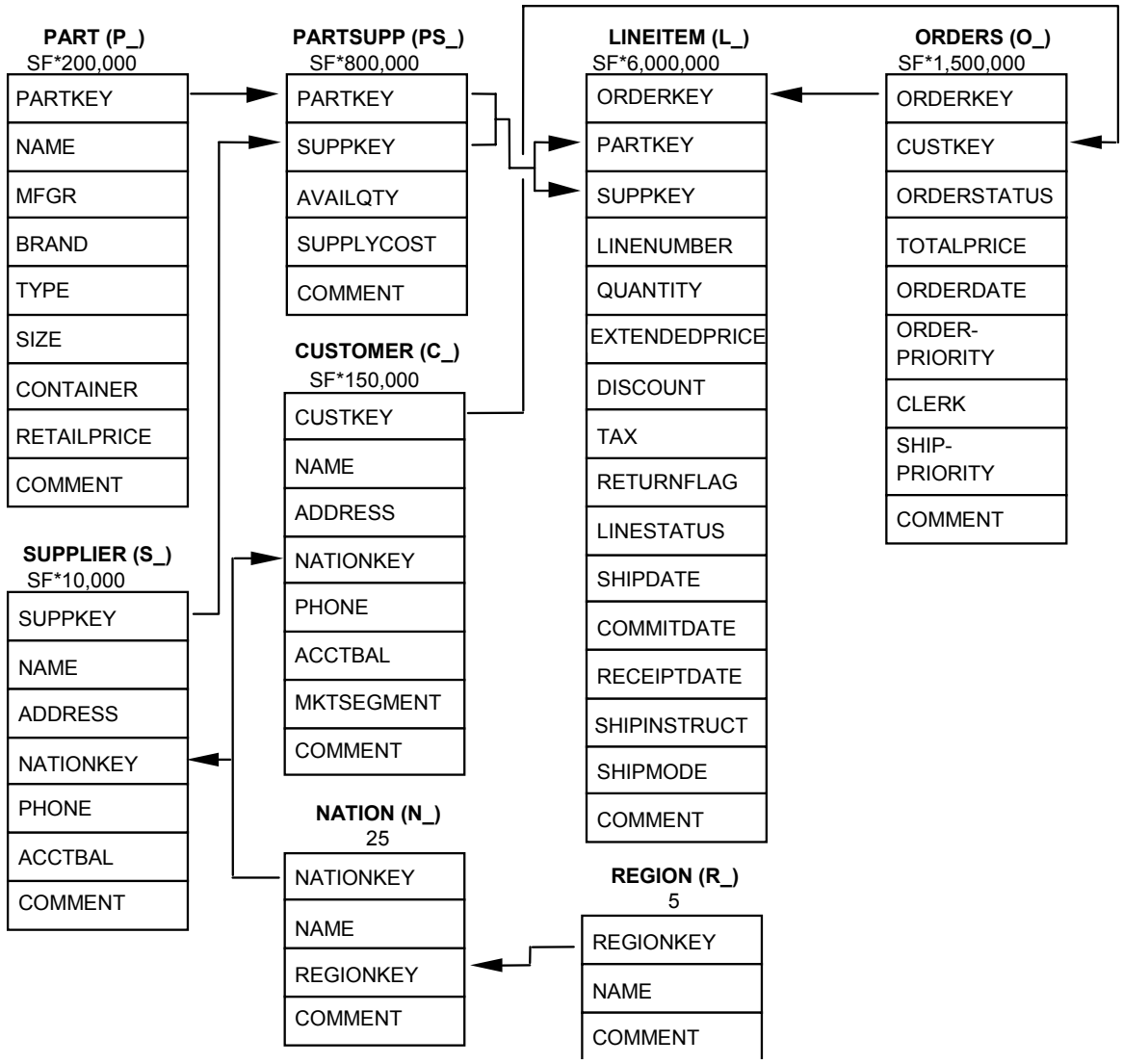

Figure 1: TPC-H Schema

Here is the file ”assignment-analyze.sql” verbatim.

```
---------------------------------
--- Olympics Data Set
---------------------------------
-- 1. Draw the query plan for the following query. Very briefly: how well did the
-- estimates match up the actual values?

select r1.event_id, r1.player_id as gold_player_id, r1.result as gold_result,
       r2.result as silver_result, r2.result - r1.result as diff
from IndividualMedals r1, IndividualMedals r2, events e
where r1.event_id = r2.event_id and r1.medal_obtained = 'GOLD' and r2.medal_obtained = 'SILVER'
      and e.event_id = r1.event_id and e.result_noted_in = 'seconds';

-- 2.  Draw the query plan for the following query. Clearly annotate the nodes in the
-- query plan that correspond to the tables "c1", "c2", and "temp".
with temp as (
    select c1.country_id, cast(c2.num_players as float)/c1.num_players as ratio
    from (select country_id, count(player_id) as num_players from players group by country_id) c1,
         (select country_id, count(player_id) as num_players from players where substr(name, 1, 1) in
    where c1.country_id = c2.country_id
)
select c.name
from temp t, countries c
where ratio = (select max(ratio) from temp) and t.country_id = c.country_id;


---------------------------------
--- TPC-H Data Set
--- Download the tpch-load.sql file and the tpch.zip files. The first file
--- ("\i tpch-load.sql") will load the entire dataset.
---------------------------------

-- 3. PostgreSQL does not do a good job with the second and third query in the
-- following set, taking more time for (2) than for (1). Explain.
explain analyze select * from lineitem, orders where o_orderkey = l_orderkey;
explain analyze select * from lineitem, orders
        where extract(year from o_orderdate) = 1996 and o_orderkey = l_orderkey;
explain analyze select * from lineitem, orders
        where extract(year from o_orderdate) = 1997 and o_orderkey = l_orderkey;

-- 4. Inspite of a very selective join condition (returning a total of 1844
-- rows), the following query does not execute efficiently. Why?  How may
-- you fix it? There is a very easy fix which PostgreSQL suprising does not
-- automatically employ.
select count(*) from orders, lineitem where l_shipdate - o_orderdate = 1
        and extract(year from o_orderdate) = 1992;

-- 5. For the following query, explain in some detail why the final result size
-- is underestimated in one case, and overestimated in the other.
explain analyze select * from supplier,customer
        where s_nationkey=c_nationkey and s_acctbal < 5000;
explain analyze select * from supplier,customer
        where s_nationkey=c_nationkey and s_acctbal > 5000;
```