

Assignment # 3

This assignment requires that you can connect to DB2 from python. If you are using an AWS instance, make sure that you use the public instance that I have prepared: databasetuning-db2v9.7 (ami-4b4ea222)

IMPORTANT: The first step when you have gone through all the license and DB2 setup screens (as usual) is to enter the following command as root to complement the Python 2.5.1 installation on your instance:

```
# /usr/lib/python/site-packages/easy_install multiprocessing
```

If you are using your own local instance, make sure you follow these instructions:

<http://tiny.cc/EIVer> . If you have a problem installing `ibm_db`, do not hesitate to contact me, chances that I ran into your problem already (IBM has already updated their documentation, they might have to do it again).

Part 1: Query Tuning

To complete this part, you will need to use the explain facilities. See documentation at the following URL: <http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.admin.doc/doc/r0005736.htm>, <http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.admin.doc/doc/r0008441.htm>

Consider the following query (query.sql on the class web site) whose objective is to find the social security number and name of the employees who have a higher attendance rate (hundreds1) or a lower sickness rate (hundreds2) than their neighbors (the neighbors are the employees whose home is located in a 10km radius from their home – the location of an employee home is defined by lat and long expressed in decimal microdegree – 1 microdegree is equivalent to 10 cm):

```
select distinct e1.ssn, e1.name from employees e1, employees e2 where e1.hundreds1 > e2.hundreds1 or e1.hundreds2 < e2.hundreds2 and sqrt(bigint(e2.lat - e1.lat)*bigint(e2.lat - e1.lat) + bigint(e2.long - e1.long)*bigint(e2.long - e1.long)) < 10000
```

Note that you should get a feel for the cardinality and cost of this query before you actually execute it (should you decide to do so).

How can you make this query perform faster? You should describe (a) the baseline performance of the query as such on your system (e.g., the cost from `db2explain` as long as the statistics are up to date), (b) the potential performance problems with this query (or the potential for performance improvements), (c) actions that you can take to make the query faster, and (d) results of experiments quantifying the impact of the actions you defined in (c).

Part 2: Tuning Experimentation

Consider the row compression feature in DB2 v9 – see documentation at the following URL:

<http://www.ibm.com/developerworks/data/library/techarticle/dm-0605ahuja/index.html>

A. What are the potential benefits and drawbacks of using row compression?

B. Define and conduct experiments that illustrate the trade-offs between these benefits and drawbacks. For the experiments, you should reuse the `reads.py` and `writes.py` scripts from Assignment 2. You should (a) define the experiments, (b) present the results on your system (do not forget to define the system you are studying – version of DB2, OS, Hardware).

Part 3: You SHOULD keep a logbook of the problems you face in Part 1 and Part 2.