CSE 6339 Sample Projects

1. Random Sampling from Databases :

For this project, you will use the Employees table. Implement the following sampling algorithms : sampling with replacement, sampling without replacement and reservoir sampling. Augment the employees table with department information (eg : employee's last department) and then perform stratified sampling based on department. You may choose to store the sample as a table in the database or hold it in-memory. The size of the sample table must be configurable.

Given a set of queries, run them on the sample table and estimate the approximate value. Compare the error bounds with the real value. How does the error bound and accuracy change as sample size increases ?

Sample Queries :

1. Count of male employees in department d.
2. Number of employees with age > 60.

2. Ripple Joins

For this project, you will use the Employee's table. Implement a ripple join algorithm that can answer sum/count/avg queries that will involve the tables (employees ,dept_employee) and (employees, salaries). After each join, update the estimated result and the confidence interval in the UI. You need not implement the iterators explicitly. For eg, if you are using Mysql use the idiom "order by rand() limit 1" to get 1 random tuple. Test how the answer improves with different aspect ratios.

Sample Queries:

1. Count of male employees in department d.
2. Average salary of employees in the age group 40-50.

3. Page Rank for Paper Citations

The idea of page rank is heavily inspired from bibliometrics. In this project, you will implement Pagerank for citations using the citations dataset and evaluate its impact. Allow the values of α and E to specified by the user. You can ignore the problem of dangling links and rank sinks. Print the top 100 papers with highest citations. Do they correspond to paper with highest citations ?

Note that this is a medium sized dataset and the adjacency matrix "may not" fully fit in memory. You can use external libraries for sparse matrices like CERN COLT, Boost::matrix_sparse, scipy.sparse etc. You must not use the linear algebra libraries to find the eigen vector and instead implement the power iteration algorithm yourself.

4. DBXplorer

In this project you will be implementing the DBXPlorer with pubcol indexing for the employee database. Use the following tables : employees , titles, departments and dept_emp. You can choose to index only textual columns.

Sample Queries :

1. Show details about employees given last name and department name
2. Find employees with a given department name and title.

5. NRA and TA

In this project you will implement a very simple text based search engine using the text

document datasets . You will construct inverted list for each word which lists all the documents in which it occurs. Store the IDF value of the word in the inverted list and TF value for each entry in the inverted list. Sort each inverted list using TF score of the documents. Additionally, load the dataset into a Lucene database.

Given a multi word search query, fetch the top 10 results using TA and NRA algorithm. The scoring to be used is TF-IDF. Compare your results with that of Lucene. You can either use Lucene API for fetching the search results or use Apache Solr directly.

Dataset Details :

1. Employees table :

Employees schema is a medium sized database used by MySQL team to test regression issues in new releases of MySQL. You can download the latest file from http://launchpad.net/test-db/employees-db-1/1.0.6/+download/employees_db-full-1.0.6.tar.bz2 . When you extract the archive, it will contain employees.sql. Sourcing the file in mysql will create the schema and load the database with records.

2. Citations Dataset :

This dataset contains citation graph details from SIGKDD 2003. You can download the file from http://www.cs.cornell.edu/projects/kddcup/download/hep-ph-citations.tar.gz . Details about the input file format is explained in http://www.cs.cornell.edu/projects/kddcup/datasets.html . Each line of the file contains two entries : paper , paper it cites to.

3. Text document dataset :

This dataset contains a set of 30 short text files. You can download the data from http://www.infosci.cornell.edu/Courses/info4300/2010fa/test/30textfiles.zip . It also contains a list of stop words which you can ignore from indexing.