CSE601 Project 1: Data Warehouse/OLAP System

Due: Oct. 10, 2015 (report and demo must be given)

In this project, you are asked to implement a clinical and genomic data warehouse based on your schema design using the Oracle system. A good data warehouse should satisfy the following requirements: 1) support regular and statistical OLAP operations; 2) be robust to potential changes in the future; and 3) support knowledge discovery.

The original data will be provided in the plain text files under the directory

/projects/azhang/cse601/Data\_For\_Project1/. A detailed description of the file format is attached at the end. The information related to Oracle system can be found at:

<https://wiki.cse.buffalo.edu/services/content/oracle>

<https://wiki.cse.buffalo.edu/services/content/how-create-oracle-table>

<https://wiki.cse.buffalo.edu/services/content/how-drop-oracle-table>

<https://wiki.cse.buffalo.edu/services/content/how-use-jdbc-oracle>

Part I:

You are required to implement your data warehouse schema in the Oracle system. Then populate your data warehouse with the provided data sets.

Do the join of all the data from the table based on the schema. This gives the easy access of data for the following queries.

Part II:

Your data warehouse is supposed to support the regular OLAP operations (e.g., roll-up, drill down, slice, dice and pivot), as well as some statistical operations (e.g., t-test, ANOVA, and correlation). In the following are some typical queries by users. You may use either SQL, PL/SQL, or external programs (e.g. in Java) to answer the queries. Notice that you should retrieve the data from the Oracle system instead of the original plain text files. Report your approach and the results returned by your data warehouse.

* List the number of patients who had “tumor” (disease description), “leukemia” (disease type) and “ALL” (disease name), separately.

Three Queries

Tumour = 53

Leukemia : 27

ALL : 13

PATIENTS TABLE AND DISEASE TABLE

* List the types of drugs which have been applied to patients with “tumor”.

HOW TO MERGE DRUG WITH PATIENT AND DISEASE

* For each sample of patients with “ALL”, list the mRNA values (expression) of probes in cluster id “00002” for each experiment with measure unit id = “001”. (**Note:** measure unit id corresponds to mu\_id in microarray\_fact.txt, cluster id corresponds to cl\_id in gene\_fact.txt, mRNA expression value corresponds to exp in microarray\_fact.txt, UID in probe.txt is a foreign key referring to gene\_fact.txt)
* For probes belonging to GO with id = “0012502”, calculate the t statistics of the expression values between patients with “ALL” and patients without “ALL”. (**Note:** Assume the expression values of patients in both groups have equal variance, use the t test for unequal sample size, equal variance)
* For probes belonging to GO with id=“0007154”, calculate the F statistics of the expression values among patients with “ALL”, “AML”, “colon tumor” and “breast tumor”. (**Note:** Assume the variances of expression values of all four patient groups are equal.)
* For probes belonging to GO with id=“0007154”, calculate the average correlation (individual correlation then average) of the expression values between two patients with “ALL”, and calculate the average correlation of the expression values between one “ALL” patient and one “AML” patient. (**Note:** For each patient, there is a list of gene expression values belonging to GO with id=“0007154”. Suppose you get “ALL” patients and “AML” patient. For the average correlation of the expression values between two patients with “ALL”, you need first calculate Person Correlations then calculate the average value. For the average correlation of the expression values between one “ALL” patient and one “AML” patient, you need first calculate Person Correlations then calculate the average value.)

Part III:

Use your data warehouse and the OLAP operations to support knowledge discovery. (**Note:** Please read the README.txt in the data file folder carefully)

1. Given a specific disease, find the informative genes.

For example, suppose we are interested in the cancer “ALL”.

1. Find all the patients with “ALL” (group A), while the other patients serve as the control (group B).
2. For each gene, calculate the t-statistics for the expression values between group A and group B.
3. If the p-value of the t-test is smaller than 0.01, this gene is regarded as an “informative” gene.

2. Use informative genes to classify a new patient (five test cases in test\_samples.txt are given in the data).

For example, given a new patient PN, we want to predict whether he/she has “ALL”.

1. Find the informative genes w.r.t. “ALL”.
2. Find all the patients with “ALL” (group A).
3. For each patient PA in group A, calculate the correlation rA of the expression values of the informative genes between PN and PA.
4. Patients without “ALL” serve as the control (group B).
5. For each patient PB in group B, calculate the correlation rB of the expression values of the informative genes between PN and PB.
6. Apply t-test on rA and rB, if the p-value is smaller than 0.01, the patient is classified as “ALL”.

**Appendix: Descriptions of Data File Format**

The data file with respect to each entity will start with a row describing the fields of the entity. Then each following row in the file corresponds to one instance of the entity.

1. Clinical data space

Entities: patient, disease, drug, test and sample

Fact table: clinical\_fact

File: patient.txt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| p\_id | ssn | name | gender | DOB |

File: disease.txt

|  |  |  |  |
| --- | --- | --- | --- |
| ds\_id | name | type | description |

File: drug.txt

|  |  |  |  |
| --- | --- | --- | --- |
| dr\_id | name | type | description |

File: test.txt

|  |  |  |  |
| --- | --- | --- | --- |
| tt\_id | name | type | setting |

File: clinical\_fact.txt

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| p\_id | ds\_id | sympton | ds\_from | ds\_to | dr\_id | dosage | dr\_from | dr\_to | tt\_id | result | tt\_date | s\_id |

2. Sample data space

Entities: sample, marker, assay, term

Fact table: sample\_fact

File: sample.txt

|  |  |  |  |
| --- | --- | --- | --- |
| s\_id | source | amount | sp\_date |

File: marker

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| mk\_id | name | type | locus | description |

File: assay.txt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| as\_id | name | type | setting | description |

File: term.txt

|  |  |  |  |
| --- | --- | --- | --- |
| tm\_id | name | type | setting |

File: sample\_fact.txt

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| s\_id | mk\_id | mk\_result | mk\_date | as\_id | as\_result | as\_date | tm\_id | tm\_description |

3. Microarray and proteomic data space

Entities: probe, measureUnit

Fact table: microarray\_fact

File: probe.txt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| pb\_id | UID | name | description | isQC |

File: measureUnit.txt

|  |  |  |  |
| --- | --- | --- | --- |
| mu\_id | name | type | description |

File: microarray\_fact.txt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| s\_id | e\_id | pb\_id | mu\_id | expression |

4. Gene data space

Entites: gene, go, cluster, domain, promoter

Fact table: gene\_fact

File: gene.txt

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UID | seqType | accession | version | seqDataset | speciesID | status |

File: go.txt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| go\_id | accession | type | name | definition |

File: cluster.txt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| cl\_id | num | pattern | tool | tSetting | description |

File: domain.txt

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| dm\_id | type | db | accession | title | length | description |

File: promoter.txt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| pm\_id | type | sequence | length | description |

File: gene\_fact.txt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| UID | go\_id | cl\_id | dm\_id | pm\_id | UID2 |

5. Experiment data space

Entities: experiment, project, platform, norm, person, protocal, publication

Fact table: experiment\_fact

File: experiment.txt

|  |  |  |
| --- | --- | --- |
| e\_id | name | type |

File: project.txt

|  |  |  |  |
| --- | --- | --- | --- |
| pj\_id | name | investigator | description |

File: platform.txt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| pf\_id | hardware | software | settings | description |

File: norm.txt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| nm\_id | type | software | parameters | description |

File: person.txt

|  |  |  |  |
| --- | --- | --- | --- |
| pn\_id | name | labName | contact |

File: protocal.txt

|  |  |  |  |
| --- | --- | --- | --- |
| pt\_id | name | text | createdBy |

File: publication.txt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| pu\_id | pub\_med\_id | title | authors | abstract | pubDate |

File: experiment\_fact.txt

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| e\_id | nm\_id | pj\_id | pn\_id | pf\_id | pt\_id | pu\_id |