CSE601 Project 1: Data Warehouse/OLAP System

Aidong Zhang

Team

Prasanna Pai – 50132731

Jesal Janani – 50132391

Chandana Satya Prakash – 50134196

Part I -----------

Data Warehouse Design

Data warehousing design lifecycle consists of four different process: conceptual modeling, logical design, data warehouse construction and application development. We as a team, have followed each process in the development of the data warehouse and were inspired with BioStar data model which helped us in efficient development of data warehouse schema.

BioStar schema is a slight modification of star schema having measure table as an interface join between the fact table and the dimension table. Also BioStar schema provides extensibility and flexibility in design of the data schema thereby helping the user with performing the OLAP operations and other statistical processes. We in this process, divided the schema in four different clusters: clinical data space, microarray data space, gene data space and experiment data space. We defined measure tables between the patient table and other dimension tables in the clinical data space for joining and filtering out the raw data. Joining with the help of measure tables helps us in data cleansing. Same process has been applied to the microarray data space, gene data space and experiment data space for quicker retrieval of data and data purification.

One of the main requirements in database design is to avoid data redundancy and BioStar helps in overcoming this issue by classifying the tables in Fact and Dimension table.

Advantage of using BioStar model:

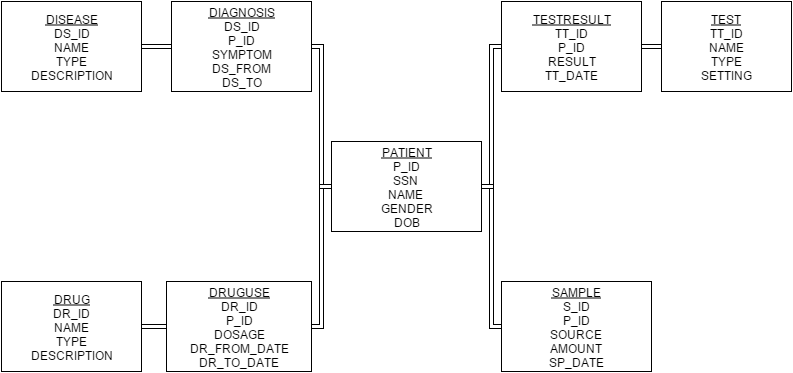
1. the many-to-many relationships between the central fact entity and dimensions are handled using the m-tables
2. the BioStar model allows an m-table to have non-measure attributes that include single- or bi-temporal support for a measure and is used for clinical data analysis
3. BioStar model can also handle incomplete data and can thus be of great use in dealing with biomedical studies

Time Complexity for BioStar schema model:

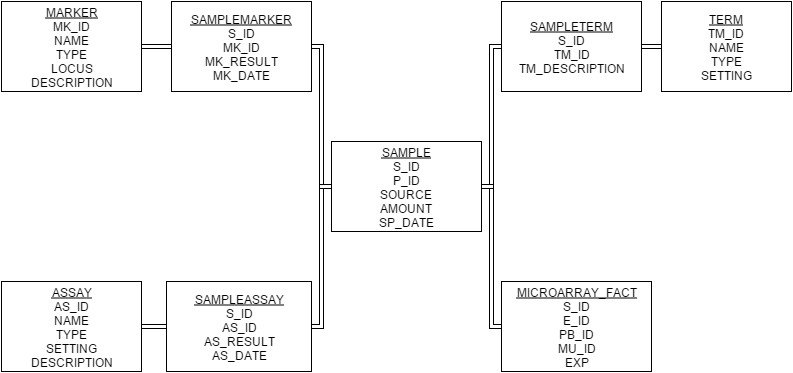
Typical OLAP operations in BioStar schema consists of roll-up, drill down, slice and dice and other statistical operations like t-test and p-select. For instance, consider a cube and we are performing a drill down on a particular record on the cube. Since the drill down operation will go the most base value of the record, we can consider it having a time complexity of O(n^3). Same situation applies for roll up operation which can be traversed in O(n^3).

Schema diagrams –

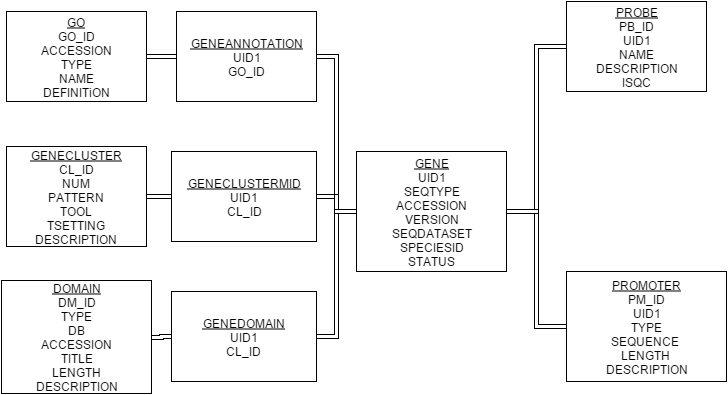
CLINICAL DATA SPACE -------



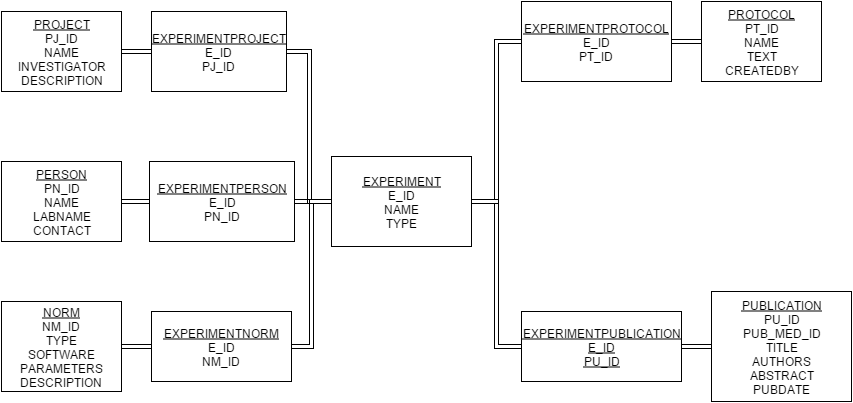
SAMPLE DATA SPACE --------------



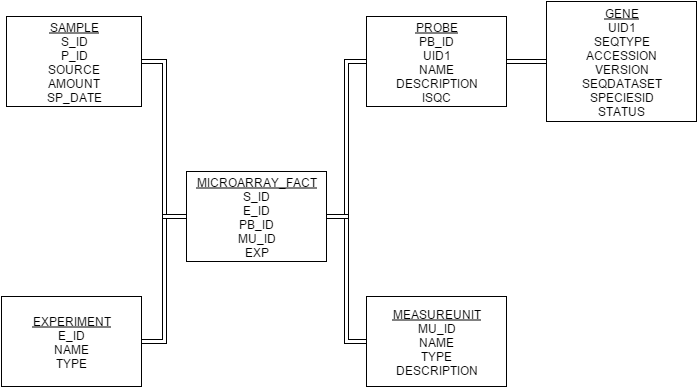
GENE DATA SPACE ----------------------



EXPERIMENT DATA SPACE -----------------



MICROARRAY DATA SPACE --------------------



Part II ----------------

Queries –

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.

Number of patients who had ‘tumor’ (disease description) -

Query: select count(P\_ID) TUMOR\_COUNT

from DIAGNOSIS1

where DS\_ID IN

(select DS\_ID from DISEASE where DESCRIPTION='tumor');

Output: 53

Number of patients who had ‘leukemia’ (disease type) –

Query: select count(P\_ID) LEUKEMIA\_COUNT

from DIAGNOSIS1

where DS\_ID IN

(select DS\_ID from DISEASE where TYPE='leukemia');

Output: 27

Number of patients with ‘ALL’ (disease name) –

Query: select count(P\_ID) ALL\_COUNT

from DIAGNOSIS1

where DS\_ID IN

(select DS\_ID from DISEASE where NAME='ALL');

Output: 13

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

Query: select distinct TYPE DRUG\_TYPES

from DRUG1

where DR\_ID IN

(select DR\_ID from DRUGUSE1 where P\_ID IN

(select P\_ID from DIAGNOSIS1 where DS\_ID IN

(select DS\_ID from DISEASE1 where DESCRIPTION='tumor')

)

) order by TYPE;

Output: 20 drug types listed to have been applied to patients with ‘tumor’ from Drug Type 001 … Drug Type 020

* 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)

Query: SELECT count(EXP) FROM MICROARRAY\_FACT1 WHERE MU\_ID = 1 AND PB\_ID IN

(SELECT PB\_ID FROM PROBE1 WHERE UID1 IN (SELECT UID1 FROM GENECLUSTERMID1 WHERE CL\_ID = 2)) AND S\_ID IN (SELECT S\_ID FROM SAMPLE1 WHERE P\_ID IN (SELECT P\_ID FROM DIAGNOSIS1 WHERE DS\_ID IN (SELECT DS\_ID FROM DISEASE1 WHERE NAME = 'ALL')))

Output: 325 mRNA expression values listed

* 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)

Query:

create table TTESTTEMP1

(

DS\_TYPE VARCHAR2(20 BYTE),

EXPR NUMBER,

G\_UID NUMBER

)

logging

tablespace CSE601

pctfree 10

initrans 1

storage

(

initial 65536

next 1048576

minextents 1

maxextents unlimited

buffer\_pool default

)

nocompress

noparallel;

-- ALL

insert into TTESTTEMP1 (

select 'ALL', EXPRESSIONS.EXP, PROBES.UID1

from (select PB\_ID, UID1 from PROBE1 where UID1 IN (select UID1 from GENEANNOTATION1 where GO\_ID = 0012502)) PROBES,

(select EXP, PB\_ID from MICROARRAY\_FACT1 where S\_ID IN

(select S\_ID from SAMPLE1 where P\_ID IN

(select P\_ID from DIAGNOSIS1 where DS\_ID IN

(select DS\_ID from DISEASE1 where NAME = 'ALL')

)

)

) EXPRESSIONS

where PROBES.PB\_ID = EXPRESSIONS.PB\_ID);

-- NOT ALL

insert into TTESTTEMP1 (

select 'NOTALL', EXPRESSIONS.EXP, PROBES.UID1

from (select PB\_ID, UID1 from PROBE1 where UID1 IN (select UID1 from GENEANNOTATION1 where GO\_ID = 0012502)) PROBES,

(select EXP, PB\_ID from MICROARRAY\_FACT1 where S\_ID IN

(select S\_ID from SAMPLE1 where P\_ID IN

(select P\_ID from DIAGNOSIS1 where DS\_ID IN

(select DS\_ID from DISEASE1 where NAME != 'ALL')

)

)

) EXPRESSIONS

where PROBES.PB\_ID = EXPRESSIONS.PB\_ID);

-- T statistics

select AVG(DECODE(DS\_TYPE, 'ALL', EXPR, null)) ALL\_AVERAGE,

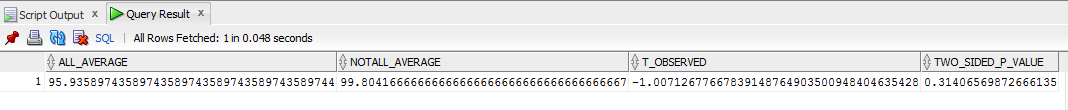
AVG(DECODE(DS\_TYPE, 'NOTALL', EXPR, null)) NOTALL\_AVERAGE,

STATS\_T\_TEST\_INDEP(DS\_TYPE, EXPR, 'STATISTIC', 'ALL') t\_observed,

STATS\_T\_TEST\_INDEP(DS\_TYPE, EXPR) two\_sided\_p\_value

from TTESTTEMP1;

Output:



* 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.)

Query:

create table ANOVA\_TEMP

(

DIS\_NAME VARCHAR2(20 BYTE),

EXP VARCHAR2(20 BYTE)

)

logging

tablespace CSE601

pctfree 10

initrans 1

storage

(

initial 65536

next 1048576

minextents 1

maxextents unlimited

buffer\_pool default

)

nocompress

noparallel;

-- ALL

insert into ANOVA\_TEMP

select 'ALL', EXP

from MICROARRAY\_FACT1

where PB\_ID in

(select PB\_ID from PROBE1 where UID1 in

(select UID1 from GENEANNOTATION1 where GO\_ID = 7154)

)

and S\_ID in

(select S\_ID from SAMPLE1 where P\_ID in

(select P\_ID from DIAGNOSIS1 where DS\_ID in

(select DS\_ID from DISEASE1 where NAME = 'ALL')

)

);

-- AML

insert into ANOVA\_TEMP

select 'AML', EXP

from MICROARRAY\_FACT1

where PB\_ID in

(select PB\_ID from PROBE1 where UID1 in

(select UID1 from GENEANNOTATION1 where GO\_ID = 7154)

)

and S\_ID in

(select S\_ID from SAMPLE1 where P\_ID in

(select P\_ID from DIAGNOSIS1 where DS\_ID in

(select DS\_ID from DISEASE1 where NAME = 'AML')

)

);

-- Colon Tumor

insert into ANOVA\_TEMP

select 'Colon tumor', EXP

from MICROARRAY\_FACT1

where PB\_ID in

(select PB\_ID from PROBE1 where UID1 in

(select UID1 from GENEANNOTATION1 where GO\_ID = 7154)

)

and S\_ID in

(select S\_ID from SAMPLE1 where P\_ID in

(select P\_ID from DIAGNOSIS1 where DS\_ID in

(select DS\_ID from DISEASE1 where NAME = 'Colon tumor')

)

);

-- Breast Tumor

insert into ANOVA\_TEMP

select 'Breast tumor', EXP

from MICROARRAY\_FACT1

where PB\_ID in

(select PB\_ID from PROBE1 where UID1 in

(select UID1 from GENEANNOTATION1 where GO\_ID = 7154)

)

and S\_ID in

(select S\_ID from SAMPLE1 where P\_ID in

(select P\_ID from DIAGNOSIS1 where DS\_ID in

(select DS\_ID from DISEASE1 where NAME = 'Breast tumor')

)

);

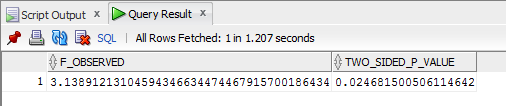
-- F statistics

select STATS\_ONE\_WAY\_ANOVA(DIS\_NAME,EXP,'F\_RATIO') f\_observed,

STATS\_ONE\_WAY\_ANOVA(DIS\_NAME,EXP,'SIG') two\_sided\_p\_value

from ANOVA\_TEMP;

Output:



* For probes belonging to GO with id=“0007154”, calculate the average correlation 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.)

Query:

create table CORRELATION\_TEMP1

(

P\_ID NUMBER,

PB\_ID NUMBER,

EXP NUMBER

)

logging

tablespace CSE601

pctfree 10

initrans 1

storage

(

initial 65536

next 1048576

minextents 1

maxextents unlimited

buffer\_pool default

)

nocompress

noparallel;

create table CORRELATION\_TEMP2

(

P\_ID NUMBER,

PB\_ID NUMBER,

EXP NUMBER

)

logging

tablespace CSE601

pctfree 10

initrans 1

storage

(

initial 65536

next 1048576

minextents 1

maxextents unlimited

buffer\_pool default

)

nocompress

noparallel;

-- ALL

insert into CORRELATION\_TEMP1

select SAMPLE1.P\_ID, MICROARRAY.PB\_ID, MICROARRAY.EXP

from (select S\_ID, PB\_ID, EXP from MICROARRAY\_FACT1 where PB\_ID in

(select PB\_ID from PROBE1 where UID1 in (select UID1 from GENEANNOTATION1 where GO\_ID = 7154))

) MICROARRAY,

(select S\_ID, P\_ID from SAMPLE1 where P\_ID in

(select P\_ID from DIAGNOSIS1 where DS\_ID in

(select DS\_ID from DISEASE1 where NAME = 'ALL')

)

) SAMPLE1

where SAMPLE1.S\_ID = MICROARRAY.S\_ID;

-- AML

insert into CORRELATION\_TEMP2

select SAMPLE1.P\_ID, MICROARRAY.PB\_ID, MICROARRAY.EXP

from (select S\_ID, PB\_ID, EXP from MICROARRAY\_FACT1 where PB\_ID in

(select PB\_ID from PROBE1 where UID1 in (select UID1 from GENEANNOTATION1 where GO\_ID = 7154))

) MICROARRAY,

(select S\_ID, P\_ID from SAMPLE1 where P\_ID in

(select P\_ID from DIAGNOSIS1 where DS\_ID in

(select DS\_ID from DISEASE1 where NAME = 'AML')

)

) SAMPLE1

where SAMPLE1.S\_ID = MICROARRAY.S\_ID;

-- Correlation between 'ALL' and 'ALL'

select AVG(CORR(P1.EXP,P2.EXP)) CORRELATION

from (select P\_ID,PB\_ID, EXP from CORRELATION\_TEMP1) P1,

(select P\_ID, PB\_ID, EXP from CORRELATION\_TEMP1) P2

where P1.PB\_ID = P2.PB\_ID

group by P1.P\_ID, P2.P\_ID;

-- Correlation between 'ALL' and 'AML'

select AVG(CORR(P1.EXP,P2.EXP)) CORRELATION

from (select P\_ID,PB\_ID, EXP from TRIAL6) P1,

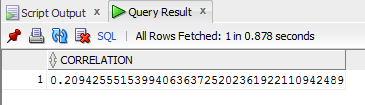
(select P\_ID, PB\_ID, EXP from TRIAL7) P2

where P1.PB\_ID = P2.PB\_ID

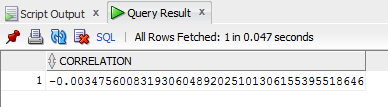
group by P1.P\_ID, P2.P\_ID;

Output:

Average correlation between ‘ALL’ patients



Average correlation between ‘ALL’ and ‘AML’



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”.

------ 3.1

create table INFO\_TEMP

(

DS\_TYPE VARCHAR2(20 BYTE),

EXPR NUMBER,

G\_UID NUMBER

)

logging

tablespace CSE601

pctfree 10

initrans 1

storage

(

initial 65536

next 1048576

minextents 1

maxextents unlimited

buffer\_pool default

)

nocompress

noparallel;

create table INFORMATIVE\_GENES1

(

GENE\_UID NUMBER

)

logging

tablespace CSE601

pctfree 10

initrans 1

storage

(

initial 65536

next 1048576

minextents 1

maxextents unlimited

buffer\_pool default

)

nocompress

noparallel;

-- ALL

insert into INFO\_TEMP (

select 'ALL', EXPRESSIONS.EXP, PROBES.UID1

from (select PB\_ID, UID1 from PROBE1) PROBES,

(select EXP, PB\_ID from MICROARRAY\_FACT1 where S\_ID in

(select S\_ID from SAMPLE1 where P\_ID in

(select P\_ID from DIAGNOSIS1 where DS\_ID in

(select DS\_ID from DISEASE1 where NAME = 'ALL')

)

)

) EXPRESSIONS

WHERE PROBES.PB\_ID = EXPRESSIONS.PB\_ID);

-- NOTALL

insert into INFO\_TEMP (

select 'NOTALL', EXPRESSIONS.EXP, PROBES.UID1

from (select PB\_ID, UID1 from PROBE1) PROBES,

(select EXP, PB\_ID from MICROARRAY\_FACT1 where S\_ID in

(select S\_ID from SAMPLE1 where P\_ID in

(select P\_ID from DIAGNOSIS1 where DS\_ID in

(select DS\_ID from DISEASE1 where NAME != 'ALL')

)

)

) EXPRESSIONS

WHERE PROBES.PB\_ID = EXPRESSIONS.PB\_ID);

-- Informative Genes

insert into INFORMATIVE\_GENES1

select GENE\_UID

from (select G\_UID GENE\_UID,

AVG(DECODE(DS\_TYPE, 'ALL', EXPR, null)) ALL\_AVERAGE,

AVG(DECODE(DS\_TYPE, 'NOTALL', EXPR, null)) NOTALL\_AVERAGE,

STATS\_T\_TEST\_INDEP(DS\_TYPE, EXPR, 'STATISTIC', 'ALL') t\_observed,

STATS\_T\_TEST\_INDEP(DS\_TYPE, EXPR) two\_sided\_p\_value

FROM INFO\_TEMP

GROUP BY ROLLUP (G\_UID)

ORDER BY G\_UID, t\_observed) INF

WHERE INF.two\_sided\_p\_value < 0.01;

Output: 38

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”.

Query:

------ 3.2

insert into PART32NEW

select G\_UID, EXP, P\_ID

from NEW\_PATIENT

where P\_ID = 'np1';

insert into PART32ALL

select PROBE1.UID1, EXP, P\_ID

from PROBE1,

(select P\_ID, ALLSAMPLES.S\_ID, PB\_ID, EXP

from MICROARRAY\_FACT1,

(select P\_ID, S\_ID from SAMPLE1 where P\_ID in

(select P\_ID from DIAGNOSIS1 where DS\_ID IN (select DS\_ID from DISEASE1 where NAME = 'ALL'))) ALLSAMPLES

where MICROARRAY\_FACT1.S\_ID = ALLSAMPLES.S\_ID) TMP

where PROBE1.PB\_ID = TMP.PB\_ID;

insert into PART32NOTALL

select PROBE1.UID1, EXP, P\_ID

from PROBE1,

(select P\_ID, ALLSAMPLES.S\_ID, PB\_ID, EXP

from MICROARRAY\_FACT1,

(select P\_ID, S\_ID from SAMPLE1 where P\_ID in

(select P\_ID from DIAGNOSIS1 where DS\_ID IN (select DS\_ID from DISEASE1 where NAME != 'ALL'))) ALLSAMPLES

where MICROARRAY\_FACT1.S\_ID = ALLSAMPLES.S\_ID) TMP

where PROBE1.PB\_ID = TMP.PB\_ID;

-- Ra

insert into PART32\_TTEST

select 'ALL', CORR(P1.EXP,P2.EXP) CORRA from

(select GENE\_UID, EXP, P\_ID from PART32ALL where GENE\_UID NOT IN (select GENE\_UID from INFORMATIVE\_GENES)) P1,

(select GENE\_UID, EXP, P\_ID from PART32NEW where GENE\_UID NOT IN (select GENE\_UID from INFORMATIVE\_GENES)) P2

where P1.GENE\_UID = P2.GENE\_UID

group by P1.P\_ID, P2.P\_ID;

-- Rb

insert into PART32\_TTEST

select 'NOTALL', CORR(P1.EXP,P2.EXP) CORRB from

(select GENE\_UID, EXP, P\_ID from PART32NOTALL where GENE\_UID NOT IN (select GENE\_UID from INFORMATIVE\_GENES)) P1,

(select GENE\_UID, EXP, P\_ID from PART32NEW where GENE\_UID NOT IN (select GENE\_UID from INFORMATIVE\_GENES)) P2

where P1.GENE\_UID = P2.GENE\_UID

group by P1.P\_ID, P2.P\_ID;

-- T statistics between Ra and Rb

SELECT AVG(DECODE(R\_TYPE, 'ALL', R\_VALUE, null)) ALL\_AVERAGE,

AVG(DECODE(R\_TYPE, 'NOTALL', R\_VALUE, null)) NOTALL\_AVERAGE,

STATS\_T\_TEST\_INDEP(R\_TYPE, R\_VALUE, 'STATISTIC', 'ALL') t\_observed,

STATS\_T\_TEST\_INDEP(R\_TYPE, R\_VALUE) two\_sided\_p\_value

FROM PART32\_TTEST;

Output: t-observed : 0.66825

p-value : 0.50699

Since that value is greater than 0.01, Test1 patient is not classified as ‘ALL’