

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:

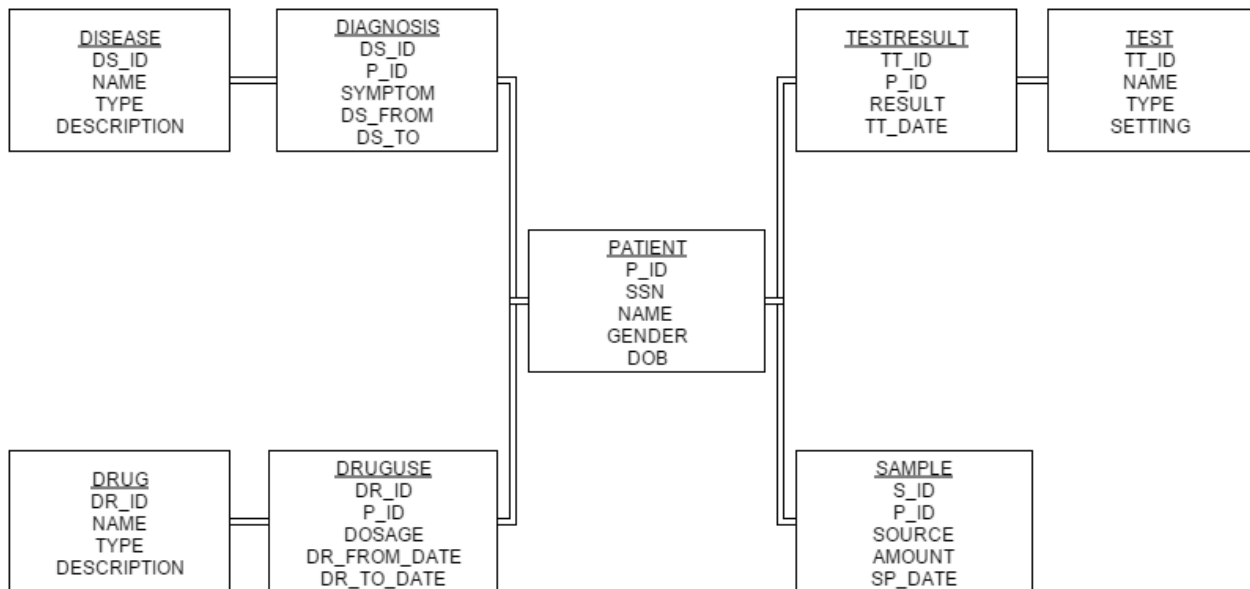
1. the many-to-many relationships between the central fact entity and dimensions are handled using the m-tables
2. the BioStar model allows an m-table to have non-measure attributes that include single- or bi-temporal support for a measure and is used for clinical data analysis
3. BioStar model can also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

Time Complexity for BioStar schema model:

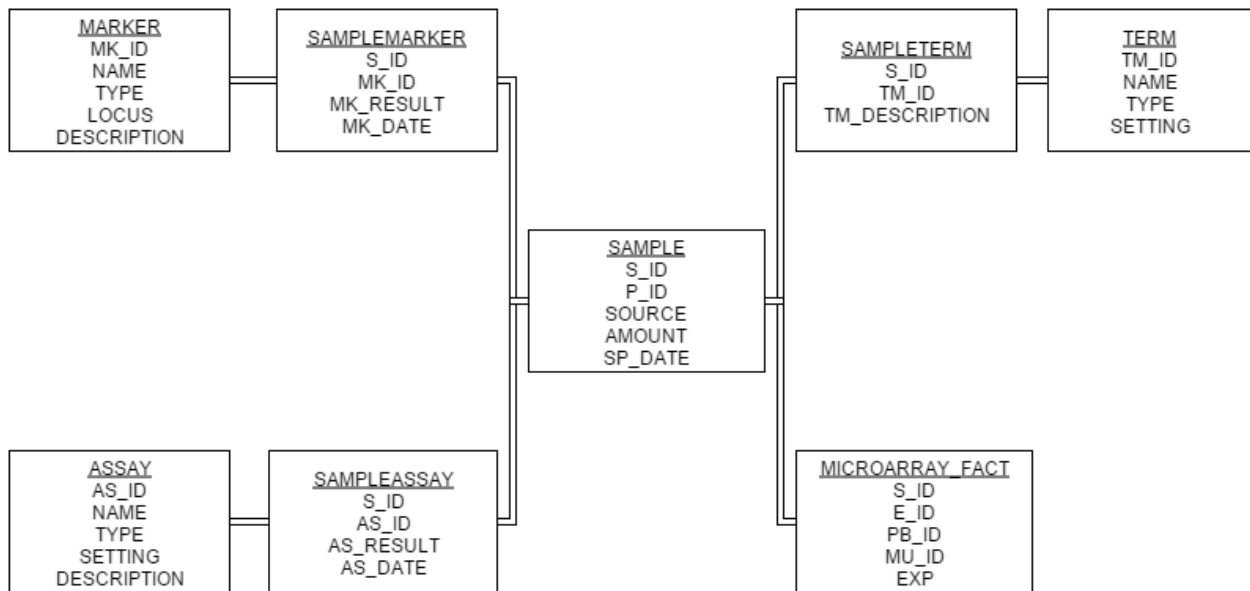
Typical OLAP operations in BioStar schema consists of roll-up, drill down, slice and dice and other statistical operations like t-test and p-select. For instance, consider a cube and we are performing a drill down on a particular record on the cube. Since the drill down operation will go the most base 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 –

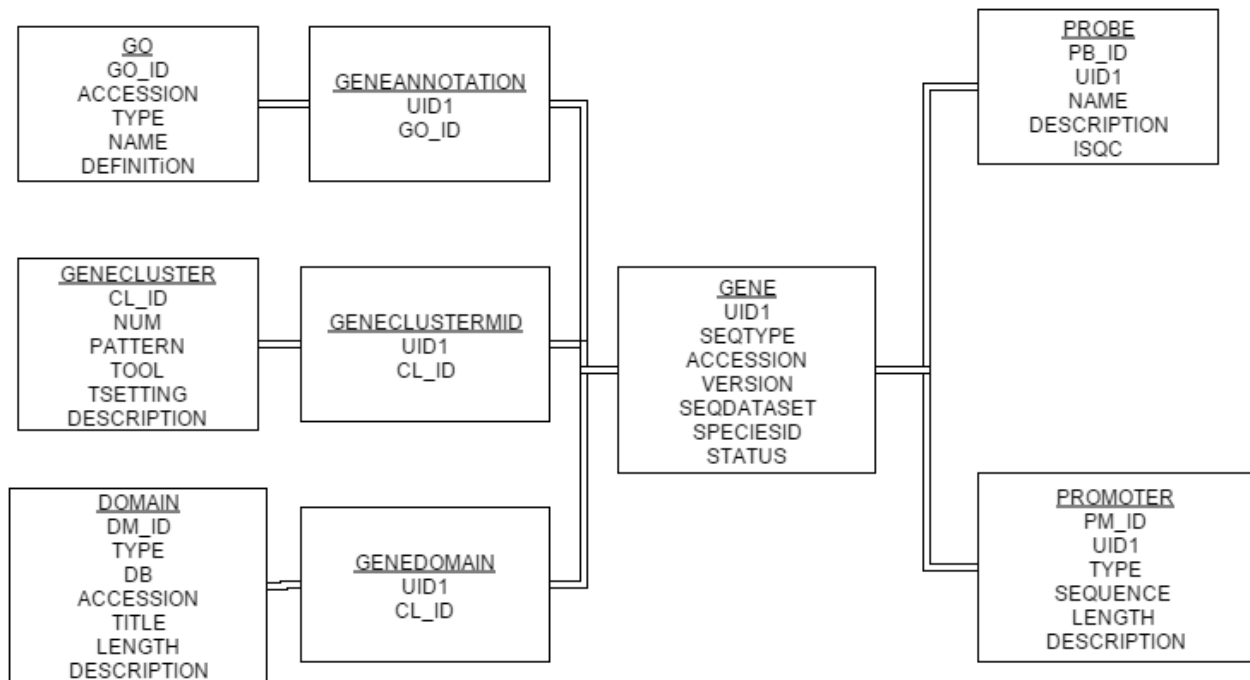
CLINICAL DATA SPACE -----



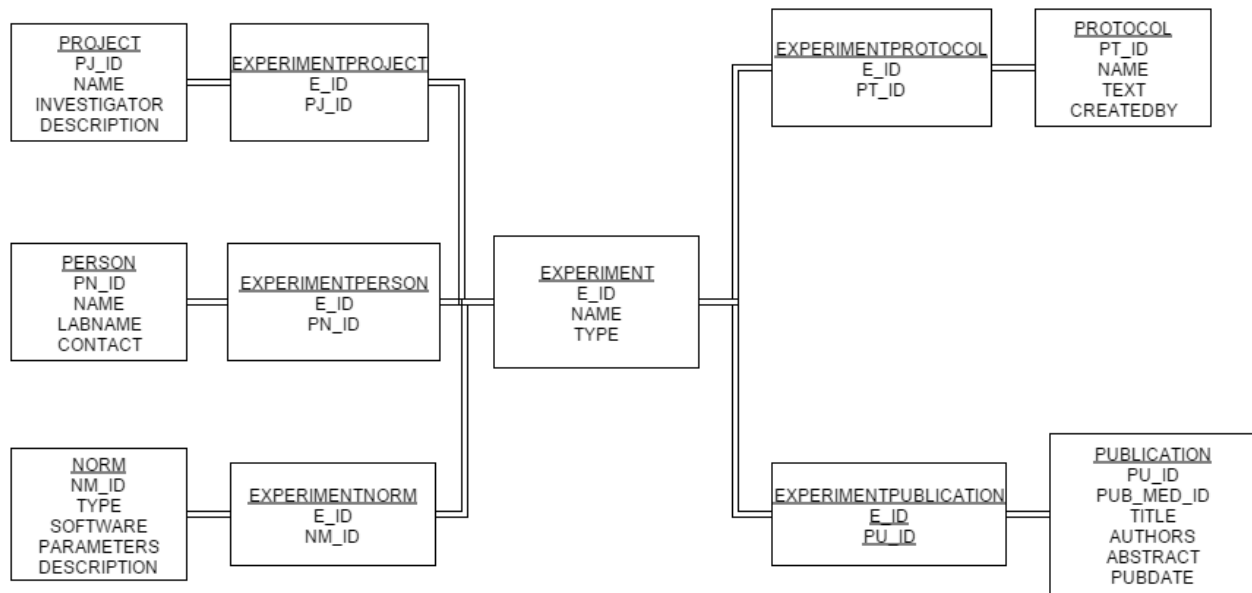
SAMPLE DATA SPACE -----



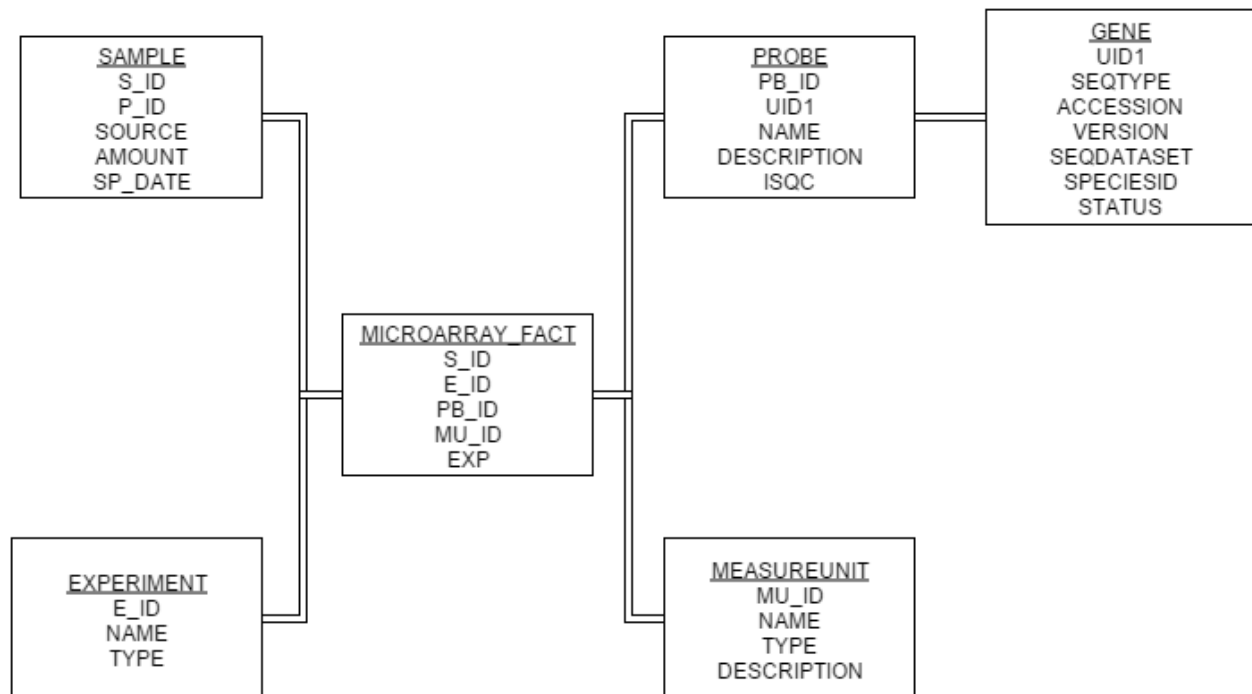
GENE DATA SPACE -----



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);

```

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.)


```

-- 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

```

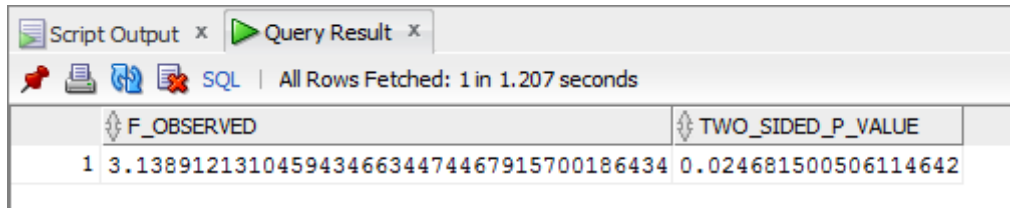
```

(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:



	F_OBSERVED	TWO_SIDED_P_VALUE
1	3.13891213104594346634474467915700186434	0.024681500506114642

- 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 N_1 "ALL" patients and N_2 "AML" patient. For the average correlation of the expression values between two patients with "ALL", you need first calculate $N_1 \times (N_1 - 1)/2$ 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 $N_1 \times N_2$ 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;

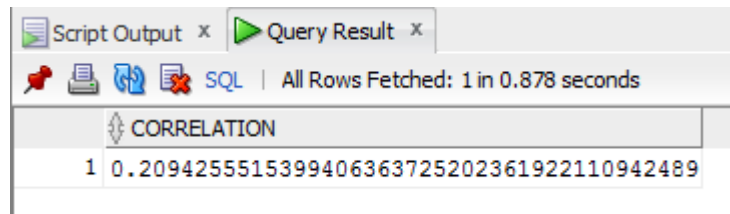
-- Correlation between 'ALL' and 'ALL'
select AVG(CORR(P1.EXP,P2.EXP)) CORRELATION
from (select P_ID,PB_ID, EXP from CORRELATION_TEMP1) P1,
     (select P_ID, PB_ID, EXP from CORRELATION_TEMP1) P2
where P1.PB_ID = P2.PB_ID
group by P1.P_ID, P2.P_ID;

-- 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:

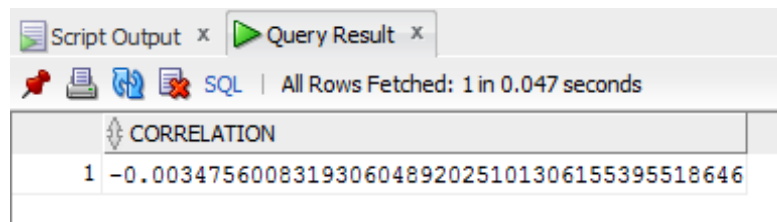
Average correlation between 'ALL' patients



The screenshot shows a SQL query result window with a single row of data. The column is labeled 'CORRELATION' and the value is 0.2094255515399406363725202361922110942489.

	CORRELATION
1	0.2094255515399406363725202361922110942489

Average correlation between 'ALL' and 'AML'



The screenshot shows a SQL query result window with a single row of data. The column is labeled 'CORRELATION' and the value is -0.003475600831930604892025101306155395518646.

	CORRELATION
1	-0.003475600831930604892025101306155395518646

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 P_N , 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'