CSE 601 Data Mining and Bioinformatics

Project 1: Data Warehouse Design

Submitted By
Mayur Tale
UB IT Name: mayurvin
Person # 50169256

Steps Followed Implementing the Data Warehouse/OLAP:

1) Identified the nature of queries beforehand –queries given in the handout were considered for reference, keeping in mind possible variations of each query
2) Identified potential changes to data – use of materialized views for query processing
3) Data Collection – we didn't have to do much in this step as data was already provided to us and we did not have to get data from external sources like websites, scraping etc.
4) Data Cleaning –
    a. Assign appropriate data types to columns – String to varchar, number, date, etc.
    b. Turning "null" string values to NULL database values
5) Model the data intelligently to correlate data with kind of queries expected – tried our best, better knowledge of data would have made things easier for us to come up with better model(s). Given model takes into consideration only the queries given in project 1 handout as the extensive transactions run on the DW.

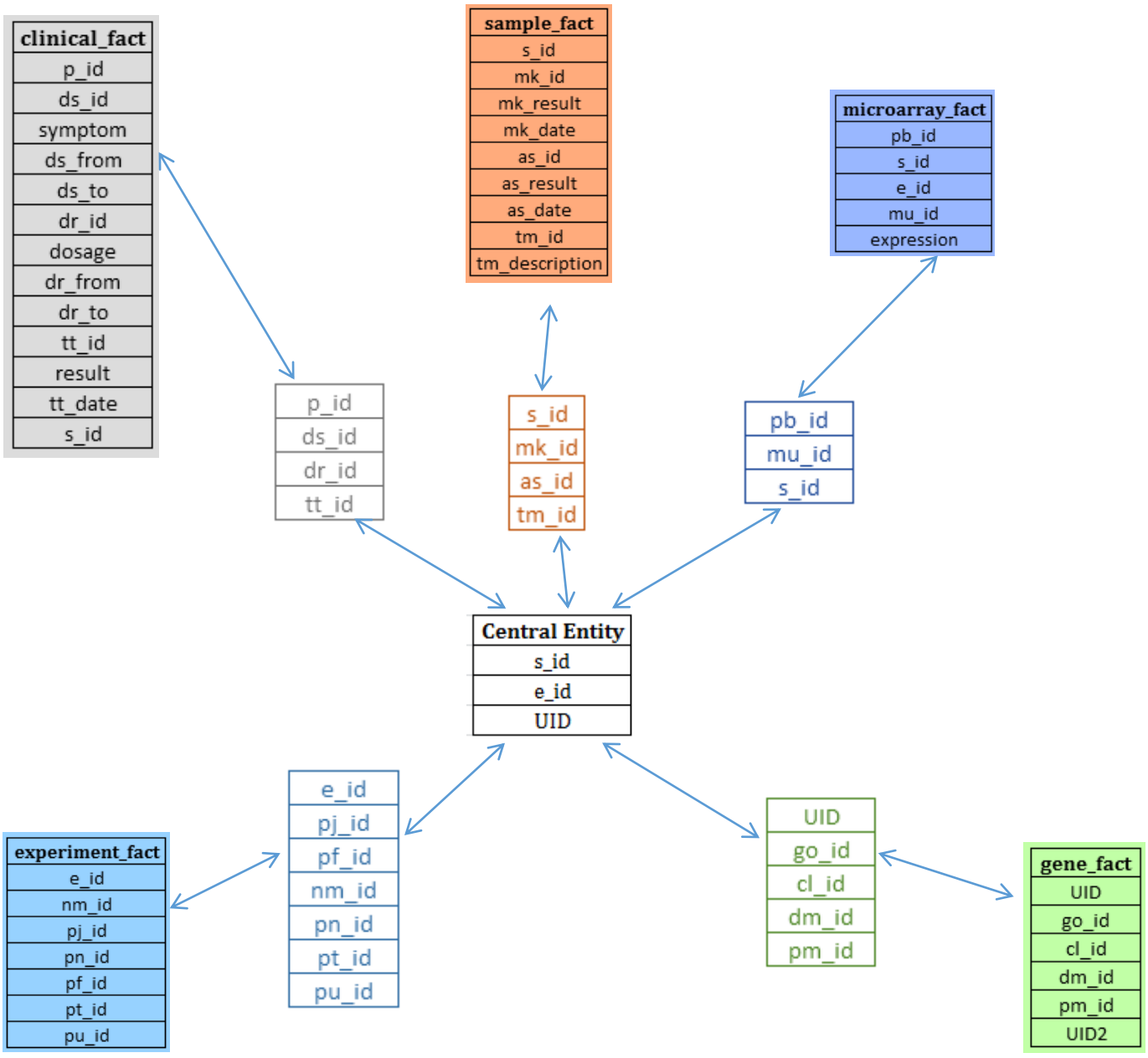
PART I: Proposed Schema Design

A central entity comprising of fact and dimension keys from the fact tables in order to maximize reachability from one relation to another and simplify complex queries. It does add certain extent of redundancy to the model but the speed ups achieved for queries with joins among *distant* tables outweighs that drawback.

The central entity is created to optimize situations where we have to involve join between multiple relations. We selected minimum number of fact and dimension keys to link all the fact tables with each other. Although by doing so we introduce certain redundancy in the model but still this would bring down the query processing time significantly and also reduce the number of joins.

Time Complexity of the model:

Typical OLAP operations consists of Roll Up, Drill down, slice and dice and other statistical operations like t-test and f-test/ANOVA. For instance, consider a cube and we are performing a drill down on particular record on the cube. Since the drill down operation will extend till the extreme base value of the record, we can consider having a time complexity of O(n3). Same situation may be applied to Roll Up which can be traversed in O(n3).


clinical_fact
p_id
ds_id
symptom
ds_from
ds_to
dr_id
dosage
dr_from
dr_to
tt_id
result
tt_date
s_id
sample_fact
s_id
mk_id
mk_result
mk_date
as_id
as_result
as_date
tm_id
tm_description
microarray_fact
pb_id
s_id
e_id
mu_id
expression
p_id
ds_id
dr_id
tt_id
s_id
mk_id
as_id
tm_id
pb_id
mu_id
s_id
Central Entity
s_id
e_id
UID
e_id
pj_id
pf_id
nm_id
pn_id
pt_id
pu_id
UID
go_id
cl_id
dm_id
pm_id
experiment_fact
e_id
nm_id
pj_id
pn_id
pf_id
pt_id
pu_id
gene_fact
UID
go_id
cl_id
dm_id
pm_id
UID2

PART II: SAMPLE QUERIES IN PROJECT HANDOUT

-------2.1

```
SELECT COUNT(DISTINCT PT.P_ID)
FROM MAYUR.DISEASE DS,
   MAYUR.PATIENT PT ,
   MAYUR.CLINICAL_FACT CLF
WHERE CLF.P_ID=PT.P_ID
   AND CLF.DS_ID=DS.DS_ID
   AND DS.DESCRIPTION='tumor';
```

	COUNT(DISTINCTPT.P_ID)
1	53

```
SELECT COUNT(DISTINCT PT.P_ID)
FROM MAYUR.DISEASE DS,
   MAYUR.PATIENT PT,
   MAYUR.CLINICAL_FACT CLF
WHERE CLF.P_ID=PT.P_ID
   AND CLF.DS_ID=DS.DS_ID
   AND DS.type='leukemia';
```

	COUNT(DISTINCTPT.P_ID)
1	27

```
SELECT COUNT(DISTINCT PT.P_ID)
FROM MAYUR.DISEASE DS,
   MAYUR.PATIENT PT,
   MAYUR.CLINICAL_FACT CLF
WHERE CLF.P_ID=PT.P_ID
   AND CLF.DS_ID=DS.DS_ID
   AND DS.name='ALL';
```

	COUNT(DISTINCTPT.P_ID)
1	13

--------2.2

```
SELECT DR.TYPE
FROM MAYUR.DRUG DR,
   MAYUR.PATIENT PT ,
   MAYUR.DISEASE DS,
   MAYUR.CLINICAL_FACT CLF
WHERE CLF.P_ID=PT.P_ID
  AND CLF.DR_ID=DR.DR_ID
  AND CLF.DS_ID = DS.DS_ID
  AND DS.DESCRIPTION='tumor';
```

OUTPUT : 53 Rows of Drug Types

--------Central View----------

```
CREATE VIEW MAYUR.CENTRAL_VIEW AS SELECT TF.DS_ID,TF.P_ID,CLF.S_ID,TF.NAME
FROM MAYUR.CLINICAL_FACT CLF,
   (SELECT CLF.P_ID,DS.DS_ID,DS.NAME
    FROM MAYUR.DISEASE DS, MAYUR.CLINICAL_FACT CLF, MAYUR.PATIENT PT
    WHERE DS.DS_ID=CLF.DS_ID AND PT.P_ID = CLF.P_ID) TF ----AND DS.NAME='ALL'
WHERE TF.P_ID=CLF.P_ID
   AND CLF.S_ID IS NOT NULL ORDER BY 1;
DROP VIEW MAYUR.CENTRAL_VIEW;
SELECT * FROM MAYUR.CENTRAL_VIEW ;
```

---------2.3

```
SELECT (MF.EXP) as EXPRESSION
FROM MAYUR.CENTRAL_VIEW CV,
   MAYUR.MICROARRAY_FACT MF,
   MAYUR.PROBE PB,
   MAYUR.GENE_FACT GF
WHERE CV.S_ID=MF.S_ID
```

```
  AND CV.NAME='ALL'
  AND GF.CL_ID = '00002'
  AND MF.MU_ID='001'
  AND MF.PB_ID= PB.PB_ID
  AND PB.U_ID = GF.U_ID;
```

OUTPUT: 325 Rows of Expression Values

---------2.4

```
CREATE TABLE MAYUR.T_TEST
(
  EXPRESSION NUMBER,
  DISEASE VARCHAR2(20 BYTE),
  SAMPLE_ID NUMBER,
  PROBE_ID NUMBER
)
logging
pctfree 10
initrans 1
storage
(
  initial 65536
  next 1048576
  minextents 1
  maxextents unlimited
  buffer_pool default
)
nocompress
noparallel;
```

-------INSERTING ALL

```
INSERT INTO MAYUR.T_TEST
SELECT  DISTINCT MF.EXP as EXPRESSION, 'ALL', CV.S_ID ,MF.PB_ID
FROM MAYUR.CENTRAL_VIEW CV,
   MAYUR.MICROARRAY_FACT MF,
   MAYUR.GENE_FACT GF,
   MAYUR.PROBE PB
WHERE CV.S_ID=MF.S_ID
  AND MF.PB_ID = PB.PB_ID
  AND PB.U_ID=GF.U_ID
  AND (CV.NAME ='ALL')
  AND GF.GO_ID = 12502
GROUP BY MF.EXP,'ALL',CV.S_ID,MF.PB_ID;
```

----INSERTING NOT ALL

```
INSERT INTO MAYUR.T_TEST
(SELECT DISTINCT MF.EXP as EXPRESSION, 'NOTALL', CV.S_ID ,MF.PB_ID
FROM MAYUR.CENTRAL_VIEW CV,
        MAYUR.MICROARRAY_FACT MF,
        MAYUR.GENE_FACT GF,
        MAYUR.PROBE PB
WHERE CV.S_ID=MF.S_ID
        AND MF.PB_ID = PB.PB_ID
        AND PB.U_ID=GF.U_ID
        AND (CV.NAME !='ALL')
        AND GF.GO_ID = 12502
GROUP BY MF.EXP,'NOTALL',CV.S_ID,MF.PB_ID);
```

------T STATISTICS

```
SELECT STATS_T_TEST_INDEP(DISEASE, EXPRESSION, 'STATISTIC', 'ALL') t_observed,
        STATS_T_TEST_INDEP(DISEASE, EXPRESSION) two_sided_p_value
from MAYUR.T_TEST;
```

	T_OBSERVED	TWO_SIDED_P_VALUE
1	-1.0071267766783914876490350094840463542 8	0.31406569872666135

----------2.5

```
CREATE TABLE MAYUR.F_TEST
(
  EXPRESSION NUMBER,
  DISEASE VARCHAR2(20 BYTE),
  SAMPLE_ID NUMBER,
  PROBE_ID NUMBER
)
logging
pctfree 10
initrans 1
storage
(
  initial 65536
  next 1048576
  minextents 1
  maxextents unlimited
  buffer_pool default
)
nocompress
noparallel;
```

----------ALL

```
INSERT INTO MAYUR.F_TEST
SELECT DISTINCT MF.EXP EXPRESSION, 'ALL', CV.P_ID, MF.PB_ID
FROM MAYUR.CENTRAL_VIEW CV,
   MAYUR.MICROARRAY_FACT MF,
   MAYUR.GENE_FACT GF,
   MAYUR.PROBE PB
WHERE CV.S_ID=MF.S_ID
   AND MF.PB_ID = PB.PB_ID
   AND PB.U_ID=GF.U_ID
   AND(CV.NAME ='ALL' )
   AND GF.GO_ID = 7154
GROUP BY MF.EXP,'ALL',CV.P_ID,MF.PB_ID;
```

----------AML

```
INSERT INTO MAYUR.F_TEST
(SELECT DISTINCT MF.EXP EXPRESSION, 'AML', CV.P_ID, MF.PB_ID
FROM MAYUR.CENTRAL_VIEW CV,
   MAYUR.MICROARRAY_FACT MF,
   MAYUR.GENE_FACT GF,
   MAYUR.PROBE PB
WHERE CV.S_ID=MF.S_ID
   AND MF.PB_ID = PB.PB_ID
   AND PB.U_ID=GF.U_ID
   AND CV.NAME ='AML'
   AND GF.GO_ID = 7154
GROUP BY MF.EXP,'AML',CV.P_ID,MF.PB_ID);
```

----------Colon Tumor

```
INSERT INTO MAYUR.F_TEST
(SELECT DISTINCT MF.EXP EXPRESSION, 'Colon_tumor', CV.P_ID, MF.PB_ID
FROM MAYUR.CENTRAL_VIEW CV,
   MAYUR.MICROARRAY_FACT MF,
   MAYUR.GENE_FACT GF,
   MAYUR.PROBE PB
WHERE CV.S_ID=MF.S_ID
   AND MF.PB_ID = PB.PB_ID
   AND PB.U_ID=GF.U_ID
   AND CV.NAME = 'Colon tumor'
   AND GF.GO_ID = 7154
GROUP BY MF.EXP,'Colon_Tumor',CV.P_ID,MF.PB_ID);
```

----------Breast tumor

```
INSERT INTO MAYUR.F_TEST
(SELECT DISTINCT MF.EXP EXPRESSION, 'Breast_tumor', CV.P_ID, MF.PB_ID
FROM MAYUR.CENTRAL_VIEW CV, MAYUR.MICROARRAY_FACT MF, MAYUR.GENE_FACT GF,
MAYUR.PROBE PB
WHERE CV.S_ID=MF.S_ID
        AND MF.PB_ID = PB.PB_ID
        AND PB.U_ID=GF.U_ID
        AND CV.NAME = 'Breast tumor'
        AND GF.GO_ID = 7154
GROUP BY MF.EXP,'Breast_tumor',CV.P_ID,MF.PB_ID);
```

---------F STATISTICS (ANOVA)

```
SELECT STATS_ONE_WAY_ANOVA(DISEASE, EXPRESSION,'F_RATIO') F_RATIO,
                STATS_ONE_WAY_ANOVA(DISEASE,EXPRESSION,'SIG') P_VALUE
FROM MAYUR.F_TEST;
```

	F_RATIO	P_VALUE
1	3.13891213104594346634474467915700186434	0.024681500506114642

----------2.6

----------Correlation between ALL and ALL

```
CREATE VIEW MAYUR.CORR_ALL AS

(SELECT DISTINCT MF.EXP EXPRESSION, CV.P_ID, MF.PB_ID

FROM MAYUR.CENTRAL_VIEW CV, MAYUR.MICROARRAY_FACT MF, MAYUR.GENE_FACT GF,
MAYUR.PROBE PB

WHERE CV.S_ID=MF.S_ID

  AND MF.PB_ID = PB.PB_ID

  AND PB.U_ID=GF.U_ID

  AND(CV.NAME ='ALL' )

  AND GF.GO_ID = 7154

GROUP BY MF.EXP,CV.P_ID,MF.PB_ID);
```

----------Correlation between ALL and ALL

```
SELECT AVG(CORR(P1.EXPRESSION,P2.EXPRESSION)) CORRELATION

from (SELECT P_ID,PB_ID, EXPRESSION FROM MAYUR.CORR_ALL) P1,

          (SELECT P_ID, PB_ID, EXPRESSION FROM MAYUR.CORR_ALL) P2

WHERE P1.PB_ID = P2.PB_ID

GROUP BY P1.P_ID, P2.P_ID;
```

	CORRELATION
1	0.20942555153994063637252023619221109424 9

----------Correlation between ALL and AML

```
CREATE VIEW MAYUR.CORR_AML AS

(SELECT DISTINCT MF.EXP EXPRESSION, CV.P_ID, MF.PB_ID

FROM MAYUR.CENTRAL_VIEW CV, MAYUR.MICROARRAY_FACT MF, MAYUR.GENE_FACT GF,
MAYUR.PROBE PB

WHERE CV.S_ID=MF.S_ID

  AND MF.PB_ID = PB.PB_ID

  AND PB.U_ID=GF.U_ID
```

```
  AND(CV.NAME ='AML' )
  AND GF.GO_ID = 7154
GROUP BY MF.EXP,CV.P_ID,MF.PB_ID);

SELECT AVG(CORR(P1.EXPRESSION,P2.EXPRESSION)) CORRELATION
from (SELECT P_ID,PB_ID, EXPRESSION FROM MAYUR.CORR_ALL) P1,
          (SELECT P_ID, PB_ID, EXPRESSION FROM MAYUR.CORR_AML) P2
WHERE P1.PB_ID = P2.PB_ID
GROUP BY P1.P_ID, P2.P_ID;
```

	CORRELATION
1	-0.00347560083193060489202510130615539551858 2

PART III: KNOWLEDGE DISCOVERY

----------3.1

```
CREATE VIEW MAYUR.GROUP_A AS
SELECT P_ID , S_ID
FROM CENTRAL_VIEW CV
WHERE CV.NAME = 'ALL';

CREATE VIEW MAYUR.GROUP_B AS
SELECT P_ID, S_ID
FROM CENTRAL_VIEW CV
WHERE CV.NAME <> 'ALL';

create table MAYUR.INFORMATIVE_GENES
(
  DISEASE VARCHAR2(20 BYTE),
  P_ID NUMBER,
  GENE_UID NUMBER,
```

```
  EXPRESSION NUMBER
)
logging
pctfree 10
initrans 1
storage
(
  initial 65536
  next 1048576
  minextents 1
  maxextents unlimited
  buffer_pool default
)
nocompress
noparallel;

INSERT INTO MAYUR.INFORMATIVE_GENES
(SELECT 'ALL', GRP_A.P_ID, GF.U_ID,MF.EXP EXPRESSION
FROM MAYUR.GROUP_A GRP_A,
   MAYUR.MICROARRAY_FACT MF, MAYUR.GENE_FACT GF,MAYUR.PROBE PB
WHERE MF.S_ID=GRP_A.S_ID
   AND MF.PB_ID=PB.PB_ID
   AND PB.U_ID= GF.U_ID
GROUP BY 'ALL',GF.U_ID,GRP_A.P_ID,MF.EXP);

INSERT INTO MAYUR.INFORMATIVE_GENES
(SELECT 'NOTALL', GRP_B.P_ID, GF.U_ID,MF.EXP EXPRESSION
FROM MAYUR.GROUP_B GRP_B,
     MAYUR.MICROARRAY_FACT MF, MAYUR.GENE_FACT GF,MAYUR.PROBE PB
```

```
WHERE MF.S_ID=GRP_B.S_ID
   AND MF.PB_ID=PB.PB_ID
   AND PB.U_ID= GF.U_ID
GROUP BY 'NOTALL',GF.U_ID,GRP_B.P_ID,MF.EXP);

CREATE VIEW MAYUR.INFO_GENE AS
(SELECT GENE_UID
FROM (SELECT GENE_UID,
        STATS_T_TEST_INDEP(DISEASE, EXPRESSION, 'STATISTIC', 'ALL') T_OBSERVED,
        STATS_T_TEST_INDEP(DISEASE, EXPRESSION) TWO_SIDED_P_VALUE
     FROM MAYUR.INFORMATIVE_GENES
     GROUP BY (GENE_UID)
     ORDER BY GENE_UID, T_OBSERVED) INFO
WHERE INFO.TWO_SIDED_P_VALUE < 0.01);
SELECT * FROM MAYUR.INFO_GENE;
```

OUTPUT: 38 Rows of Informative Genes

----------3.2

```
create table MAYUR.P_NEW
(
  GENE_UID NUMBER,
  TEST1 NUMBER,
  TEST2 NUMBER,
  TEST3 NUMBER,
  TEST4 NUMBER,
  TEST5 NUMBER
  --P_ID NUMBER
)
logging
pctfree 10
```

```
initrans 1
storage
(
  initial 65536
  next 1048576
  minextents 1
  maxextents unlimited
  buffer_pool default
)
nocompress
noparallel;

INSERT INTO MAYUR.P_NEW
SELECT U_ID, TEST1, TEST2, TEST3, TEST4, TEST5
FROM MAYUR.TEST_SAMPLES
WHERE U_ID IN (SELECT GENE_UID FROM MAYUR.INFO_GENE);

CREATE VIEW MAYUR.EXP_GRPA AS
SELECT GR.P_ID, GF.U_ID,MF.EXP EXPRESSION
FROM MAYUR.GROUP_A GR,MAYUR.MICROARRAY_FACT MF, MAYUR.GENE_FACT GF,MAYUR.PROBE PB
WHERE MF.S_ID=GR.S_ID AND MF.PB_ID=PB.PB_ID AND PB.U_ID= GF.U_ID
GROUP BY GF.U_ID,GR.P_ID,MF.EXP;
SELECT COUNT(*) FROM MAYUR.EXP_GRPA;

CREATE VIEW MAYUR.EXP_GRPB AS
SELECT GF.U_ID,GR.P_ID,MF.EXP EXPRESSION
FROM MAYUR.GROUP_B GR,MAYUR.MICROARRAY_FACT MF, MAYUR.GENE_FACT GF,MAYUR.PROBE PB
WHERE MF.S_ID=GR.S_ID AND MF.PB_ID=PB.PB_ID AND PB.U_ID= GF.U_ID
GROUP BY GF.U_ID,GR.P_ID,MF.EXP;
```

```
SELECT COUNT(*) FROM MAYUR.EXP_GRPB;

CREATE VIEW MAYUR.INFO_GENE_ALL AS
(SELECT IG.P_ID, IG.U_ID,IG.EXPRESSION
FROM MAYUR.EXP_GRPA IG
WHERE IG.U_ID IN
   (SELECT GENE_UID FROM MAYUR.INFO_GENE));
SELECT COUNT(*)  FROM MAYUR.INFO_GENE_ALL;

CREATE VIEW MAYUR.INFO_GENE_NOT_ALL AS
(SELECT IG.P_ID, IG.U_ID,IG.EXPRESSION
FROM MAYUR.EXP_GRPB IG
WHERE IG.U_ID IN
   (SELECT GENE_UID FROM MAYUR.INFO_GENE));
SELECT COUNT(*) FROM MAYUR.INFO_GENE_NOT_ALL;

DROP TABLE MAYUR.CORR_FOR_TTEST;
create table MAYUR.CORR_FOR_TTEST
(
  DISEASE VARCHAR2(30 BYTE),
  CORR_VALUE NUMBER
)
logging
pctfree 10
initrans 1
storage
(
  initial 65536
  next 1048576
```

```
  minextents 1
  maxextents unlimited
  buffer_pool default
)
nocompress
noparallel;

INSERT INTO MAYUR.CORR_FOR_TTEST
(select 'ALL', CORR(P1.EXPRESSION,P2.TEST5) CORRA
from (select U_ID, EXPRESSION, P_ID from MAYUR.INFO_GENE_ALL) P1,
  (select GENE_UID, TEST5 from MAYUR.P_NEW) P2
where P1.U_ID = P2.GENE_UID
group by P1.P_ID);

INSERT INTO MAYUR.CORR_FOR_TTEST
(select 'NOTALL', CORR(P1.EXPRESSION,P2.TEST5) CORRA
from (select U_ID, EXPRESSION, P_ID from MAYUR.INFO_GENE_NOT_ALL) P1,
  (select GENE_UID, TEST5 from MAYUR.P_NEW) P2
where P1.U_ID = P2.GENE_UID
group by P1.P_ID);

SELECT
    STATS_T_TEST_INDEP(DISEASE, CORR_VALUE, 'STATISTIC', 'ALL') t_observed,
    STATS_T_TEST_INDEP(DISEASE, CORR_VALUE) two_sided_p_value
FROM MAYUR.CORR_FOR_TTEST;
```

Knowledge Discovery Results

TEST PATIENT 1

	T_OBSERVED	TWO_SIDED_P_VALUE
1	18.475170049327341980950792530461052892 8	0.0000000000000000000000030193975776295 77

P_Value < 0.01 ➔ This new patient **HAVE "ALL"**

TEST PATIENT 2

	T_OBSERVED	TWO_SIDED_P_VALUE
1	6.508247454372413015315790274224603309 55	0.000000032547484661996683

P_Value < 0.01 ➔ This new patient **HAVE "ALL"**

TEST PATIENT 3

	T_OBSERVED	TWO_SIDED_P_VALUE
1	-0.289239688465966788804028684666253292 5469	0.77357051847177338

P_Value > 0.01 ➔ This new patient **DOES NOT HAVE "ALL"**

TEST PATIENT 4

	T_OBSERVED	TWO_SIDED_P_VALUE
1	19.160580316731861573477757968463765064 44	0.0000000000000000000000000059265057752089657

P_Value < 0.01 ➔ This new patient **HAVE "ALL"**

TEST PATIENT 5

	T_OBSERVED	TWO_SIDED_P_VALUE
1	-3.030954964549196329687543333937156855 34	0.0038238123330604787

P_Value < 0.01 ➔ This new patient **HAVE "ALL"**