**CSE 601 – Project 1**

**Data Warehouse/ OLAP operations**

**Akash Mandole 50168447**

**Bhavani Sundara Raman 50169253**

**Harsha Sudarshan 50169593**

**Data Warehouse Schema:**

In our implementation we have used our Bio Constellation schema to effectively retrieve the data using queries. We have studied Biostar schema and identified its advantages and disadvantages and have proposed Bio Constellation schema to overcome some of the disadvantages. Biostar schema as proposed in the paper, seemed to suit our data set the best. It removes redundancy and identifies the important entity around which all measure and dimensions are surrounded. This model provides flexibility by enabling classification hierarchies for which some or all data can be used. The classifications are planned and dimensions may differ depending on central entities and measures. New data can be added or deleted any time without causing redundancies. New dimensions can also be populated without affecting other tables. However in order to obtain certain data it becomes challenging to combine data from different Biostar schemas. In our model, we have tried to overcome this querying complexity by combining ideas from fact constellation schema and bio star schema.

The basic idea is to explore the correlation across various dimensions by arranging various data spaces into a ring structure. For instance dimension 2 of data space 1 and dimension 3 of data space 2 are correlated. Dimensions can also be shared across multiple data spaces for instance data space 1 and data space 3 share a common dimension. This information can be very useful during data mining and can be very effective against complex queries across data spaces. Following figure illustrates the general schema.

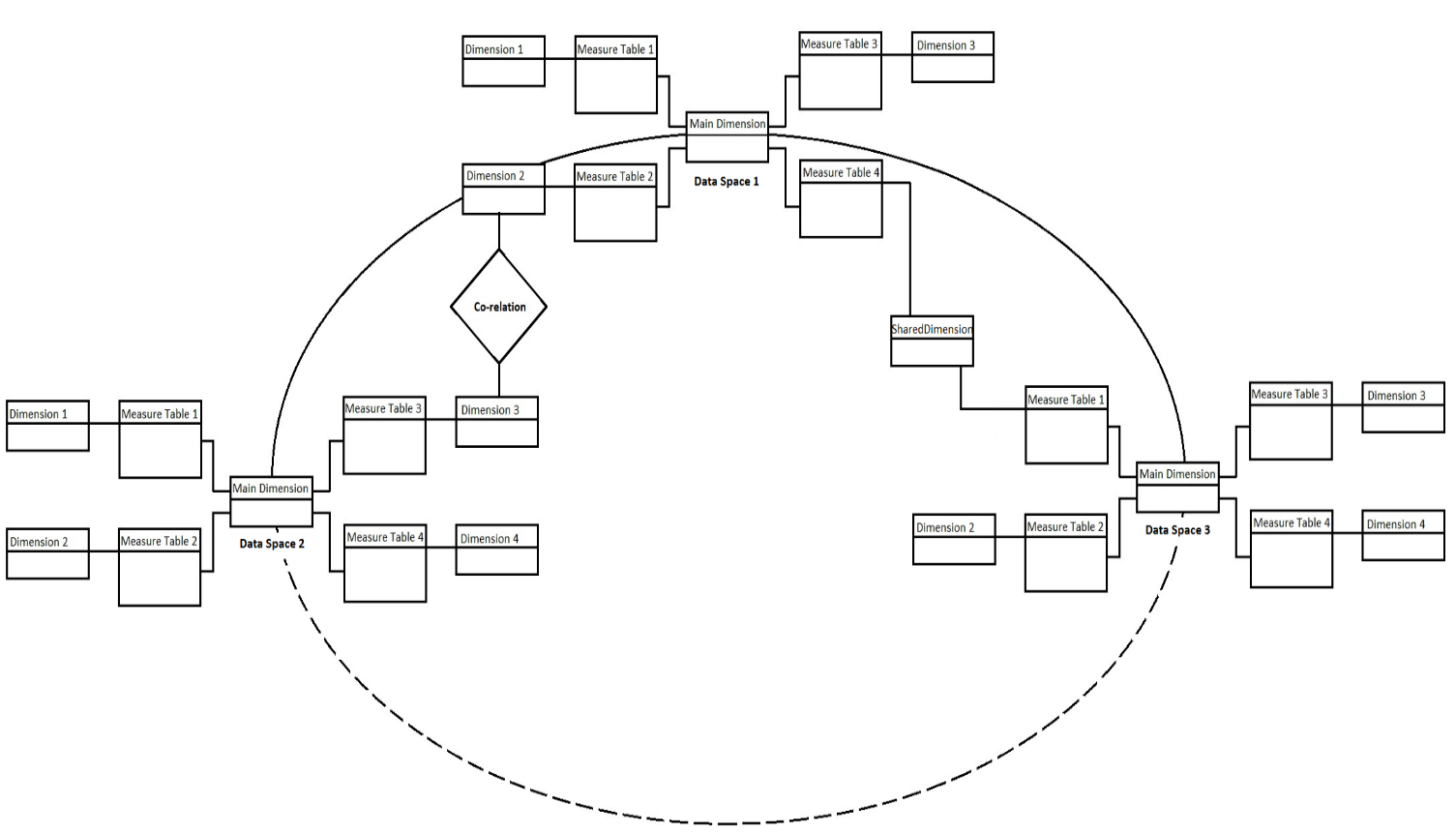
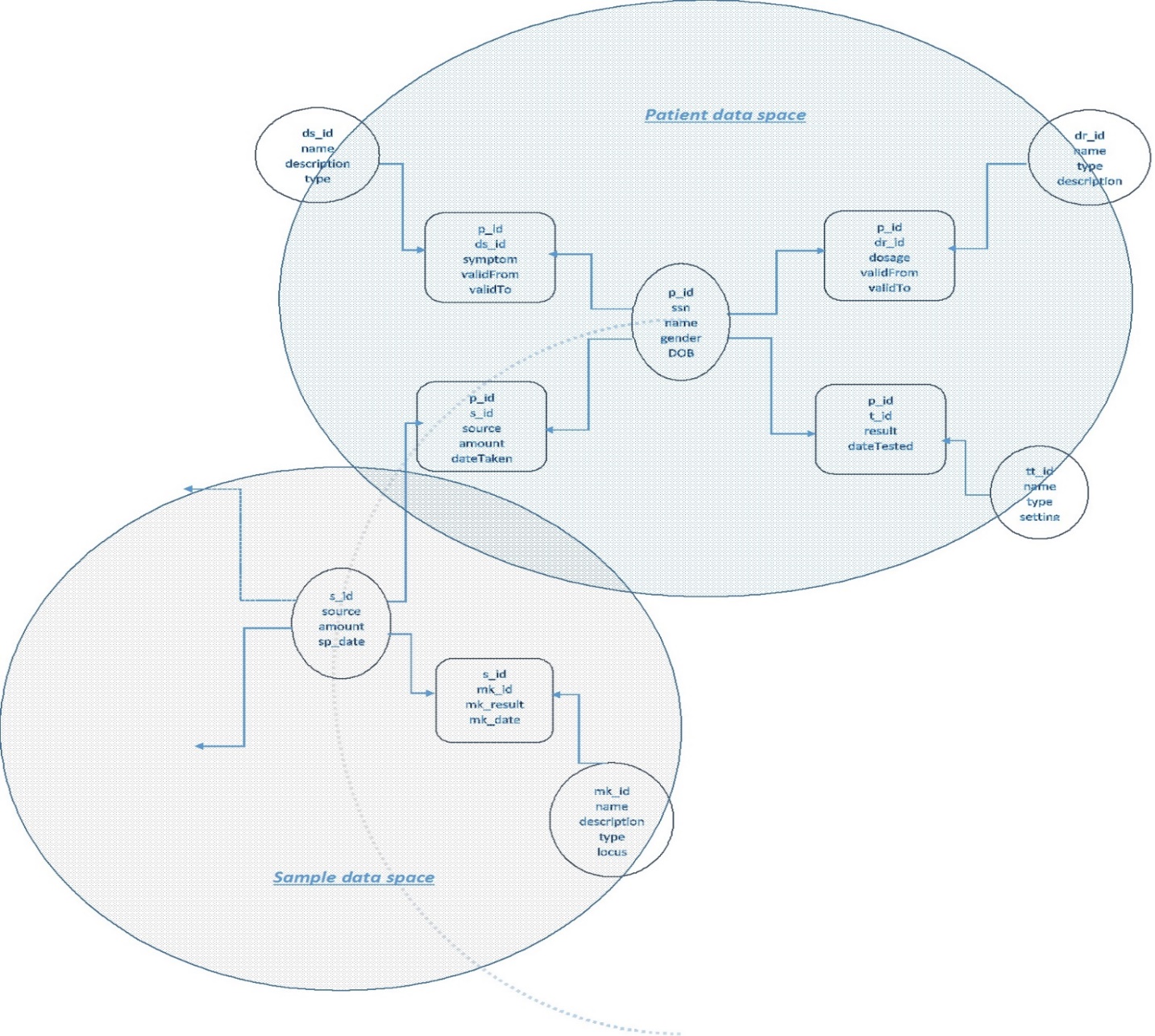
Fig 1: Example of Bio constellation design using part of the given data

Fig 2: Bio constellation schema with patient data space and sample data space



**Implementation Notes**

       We have implemented our own schema Bio constellation which provides more efficiency and ease of use, then the provided star schema. A detailed time complexity for queries are provided in further sections.

       We have used software engineering principle of modular design and reusability by creating stored procedures and functions to execute queries.

       These functions not only provide a lucid understanding of code but also provide a more general query format for Eg. We can call  ONE\_DS\_CORR() function for various parameters like ‘ALL’ or ‘AML’.

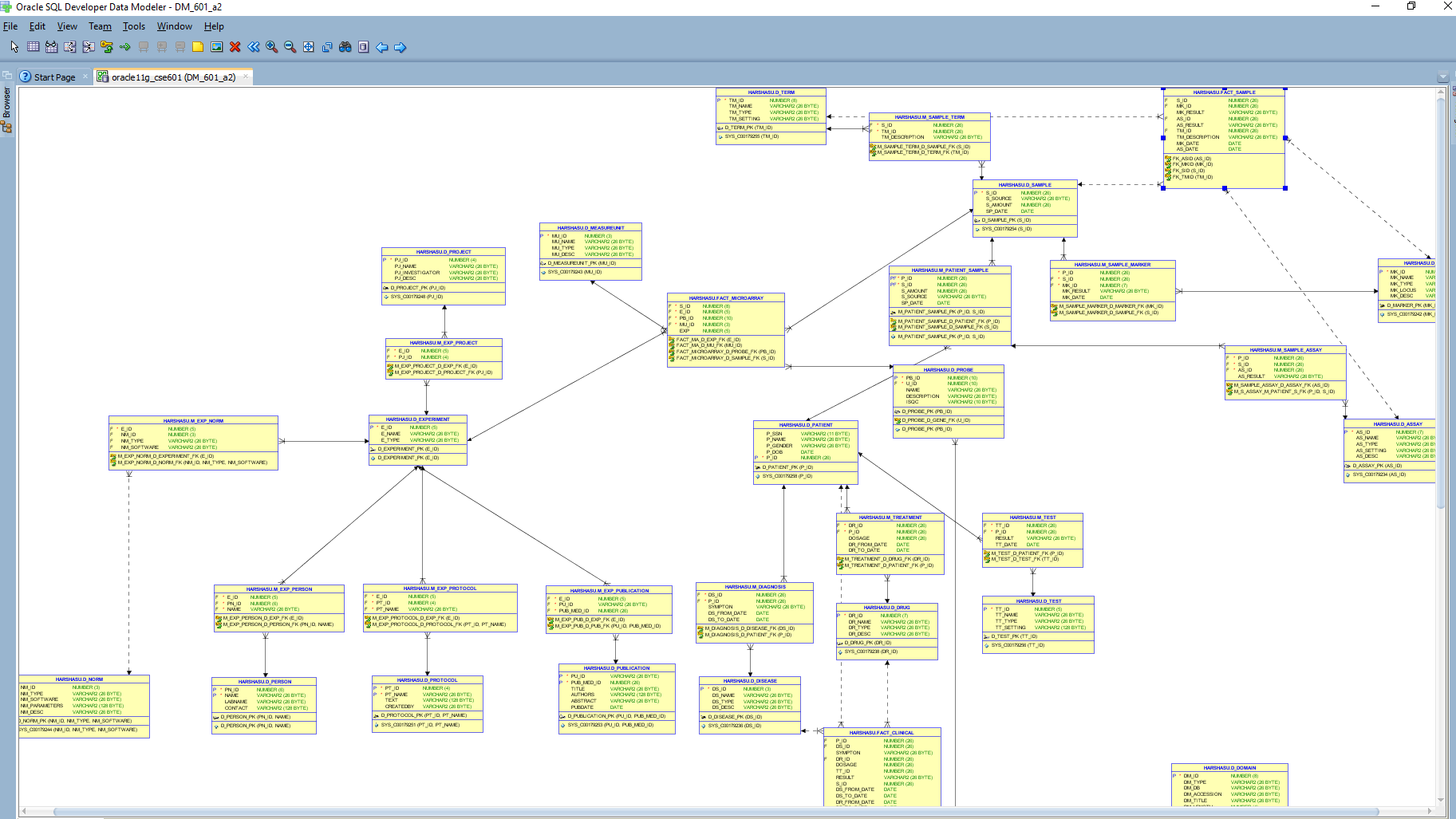
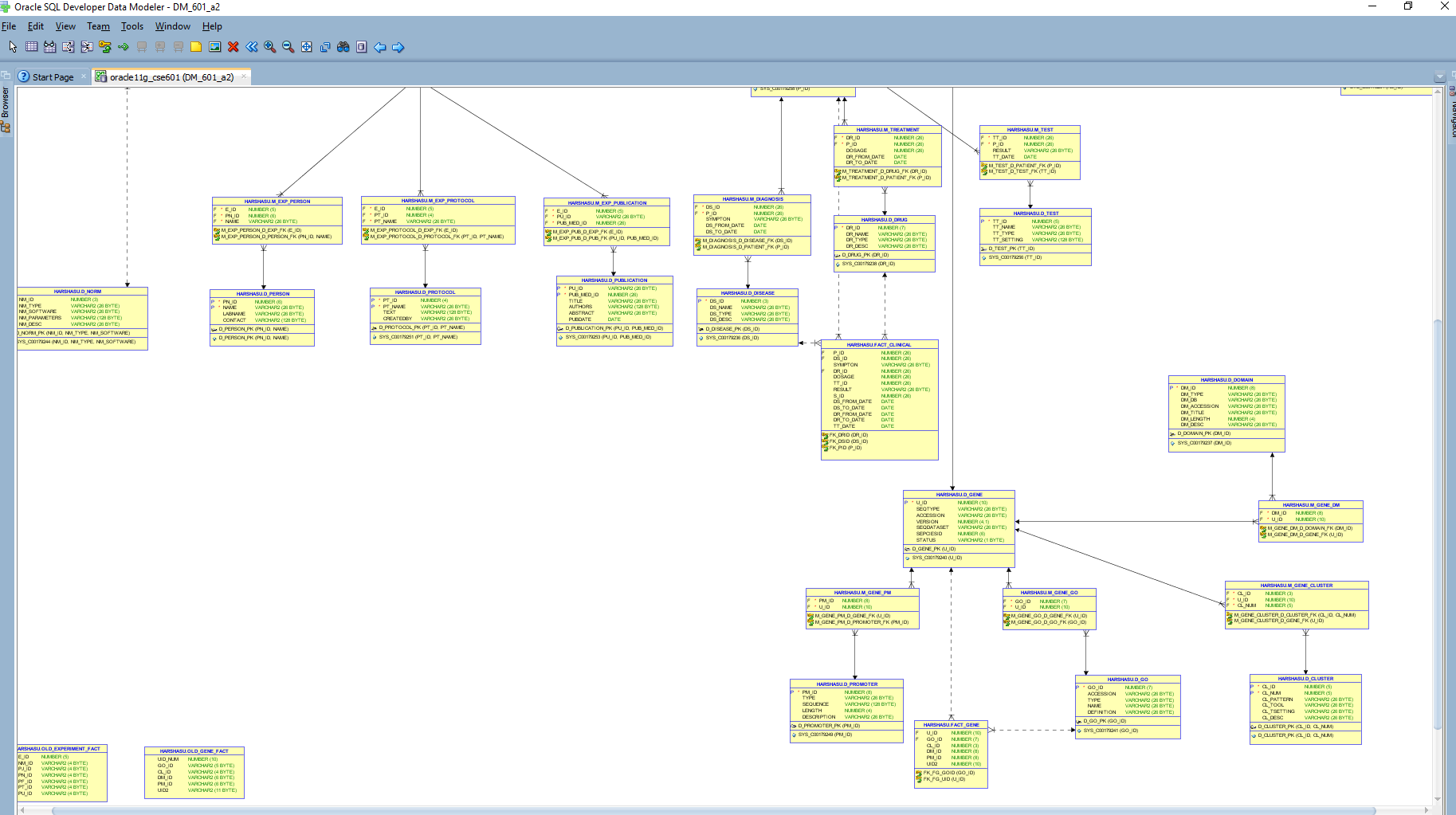
       We have created a simplistic UI design to connect query the database and retrieve results for some queries.

       For Data analytics we have used OLAP operations. For example in the following query we have used ROLLUP operation.

STATS\_T\_TEST\_INDEP(decode(md.sympton,'ALL','ALL','OTHERS'), vpg.EXP) two\_sided\_p\_value from view\_patient\_till\_go vpg, m\_diagnosis md where vpg.P\_ID = md.P\_ID GROUP BY ROLLUP(vpg.U\_ID));

**Importing Data from Excel to Database:**

We imported data data from Excel to our database using SQL Developer. SQL Developer provides with options to import data from external source. We created tables using import function.

**Data Model of the schema design:**

We have used software engineering principles in designing reusable procedures and functions to get complex data analytics.

**Results of sample queries in Project Description Part II:**

In our approach, we have created multiple measure tables and try to use the measure tables as much as possible (instead of fact tables).

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

Solution:

Q1. Since d\_disease had limited records, the index was instead created on m\_diagnosis ( a view on d\_patient and d\_disease)

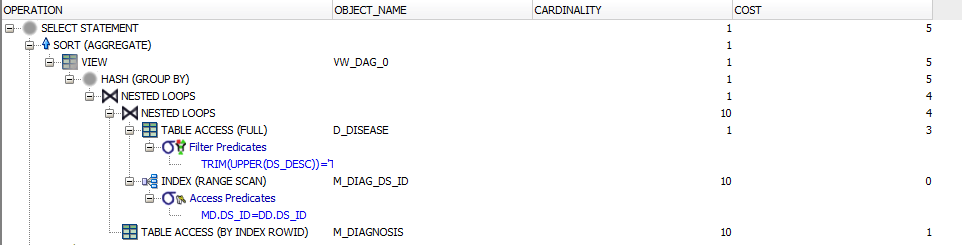
**List patients who had tumor (ds desc)**

select count(distinct p\_id)

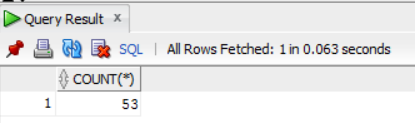
from m\_diagnosis md , d\_disease dd

where md.ds\_id = dd.ds\_id

and trim(upper(ds\_desc)) = 'TUMOR';



RESULT:



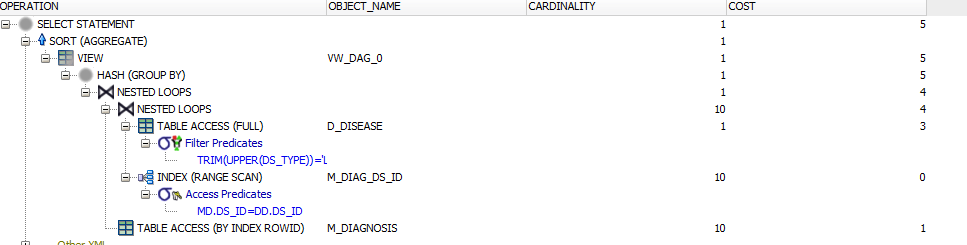
**List patients with ds name ALL**

select count(distinct p\_id)

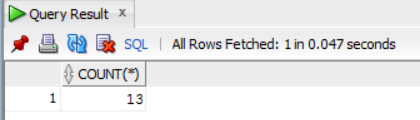
from m\_diagnosis md , d\_disease dd

where md.ds\_id = dd.ds\_id

and trim(upper(ds\_name)) = 'ALL';



RESULT:



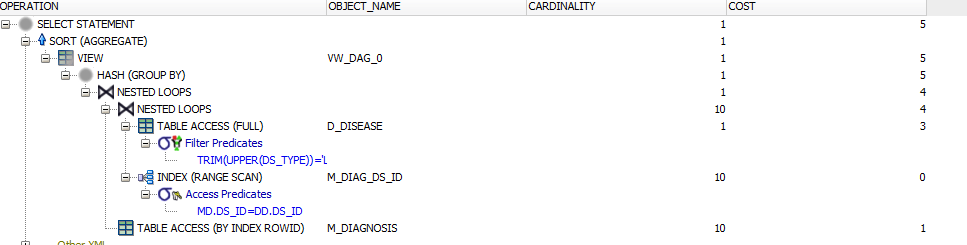
**List patients who had ds type leukemia**

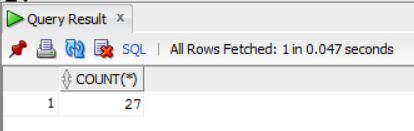
select count(distinct p\_id)

from m\_diagnosis md , d\_disease dd

where md.ds\_id = dd.ds\_id

and trim(upper(ds\_type)) = 'LEUKEMIA';





2) List the types of drugs which have been applied to patients with “ tumor”

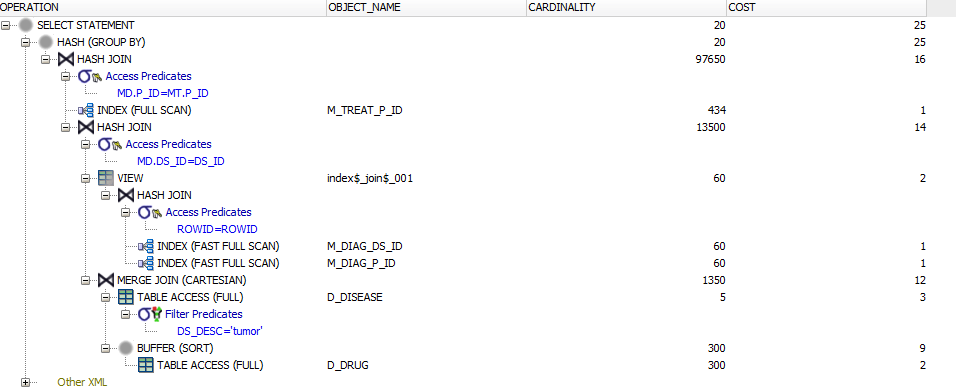
Solution:

Q2.

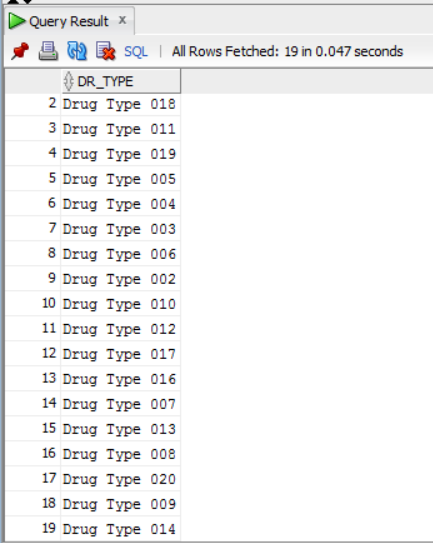
**List the types of drugs for patients with tumor (ds desc)**

select UNIQUE DR\_TYPE from HARSHASU.D\_DRUG where DR\_ID IN (select DR\_ID from HARSHASU.FACT\_CLINICAL where DS\_ID IN (select DS\_ID from HARSHASU.D\_DISEASE where DS\_DESC = 'tumor'));

As can be seen from below, multiple indexes are used when the query is run.



RESULT:



3) 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)

Solution:

select EXP from FACT\_MICROARRAY where MU\_ID =001 and S\_ID in (select S\_ID from FACT\_CLINICAL where S\_ID is not null and P\_ID in (select P\_ID from FACT\_CLINICAL where DS\_ID = (select D\_DISEASE.DS\_ID from D\_DISEASE where DS\_NAME ='ALL'))) and PB\_ID in (select PB\_ID from D\_PROBE where U\_ID in (select U\_ID from FACT\_GENE where CL\_ID =002));

RESULT:



4) 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)

Solution:

Q4. T statistics

select STATS\_T\_TEST\_INDEP(decode(dds.DS\_NAME,'ALL','ALL','OTHERS'), vpg.EXP, 'STATISTIC','ALL') t\_observed,

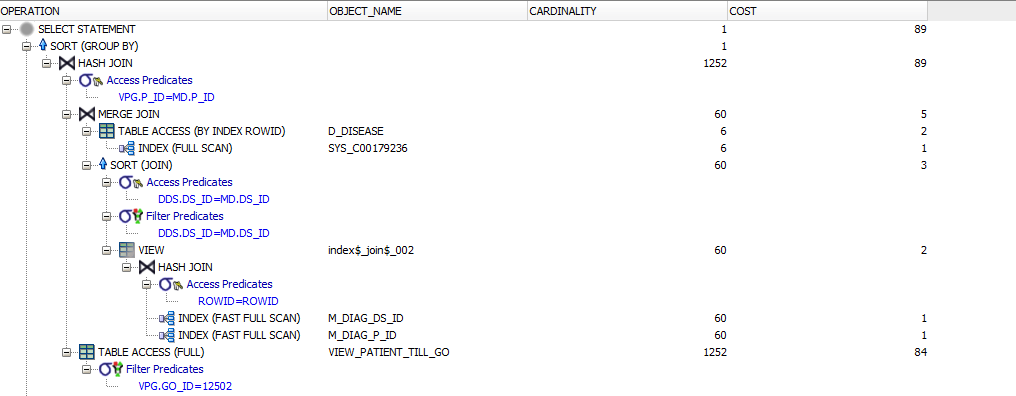
STATS\_T\_TEST\_INDEP(decode(dds.DS\_NAME,'ALL','ALL','OTHERS'), vpg.EXP) two\_sided\_p\_value

from view\_patient\_till\_go vpg, m\_diagnosis md, d\_disease dds

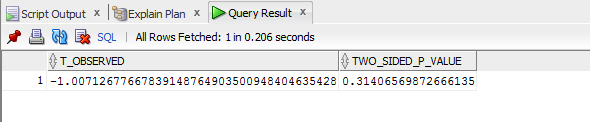
where vpg.GO\_ID=12502

and vpg.P\_ID = md.P\_ID

and dds.ds\_id = md.DS\_ID;



RESULT:



5) 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.)Solution:

select STATS\_ONE\_WAY\_ANOVA(md.SYMPTON, vpg.EXP, 'F\_RATIO') f\_ratio,

STATS\_ONE\_WAY\_ANOVA(md.SYMPTON, vpg.EXP, 'SIG') p\_value

from view\_patient\_till\_go vpg, m\_diagnosis md , d\_disease dds

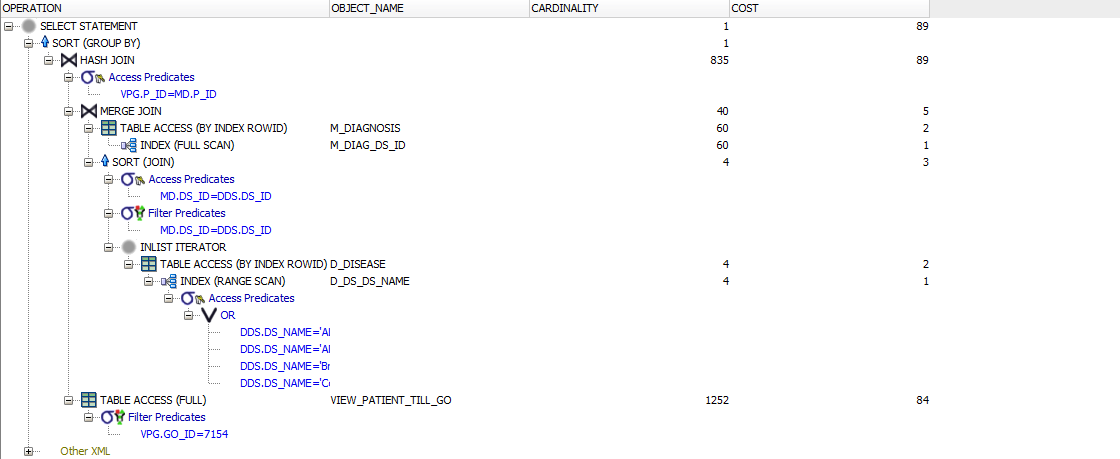
where vpg.GO\_ID=7154

and vpg.P\_ID = md.P\_ID

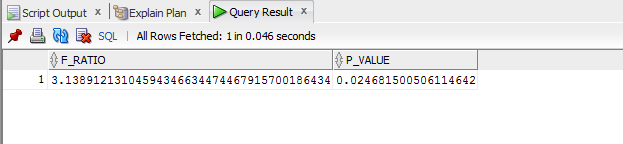
and md.DS\_ID = dds.DS\_ID

and dds.DS\_NAME in ('ALL','AML','Colon tumor','Breast tumor');

As can be observed from explain plan, multiple indexes are picked resulting in reduced cost



RESULT:



6) 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 𝑁1 “ ALL” patients and 𝑁2 “ AML” patient. For the average correlation of the expression values between two patients with “ ALL”, you need first calculate 𝑁1 × (𝑁1 − 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 𝑁1 × 𝑁2 Person Correlations then calculate the average value.)

Solution:

Declare

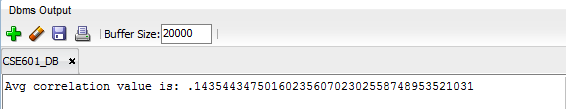
z number:=0;

begin

z:=ONE\_DS\_CORR('7154','ALL');

dbms\_output.put\_line('Avg correlation value is: '||z);

end;



declare

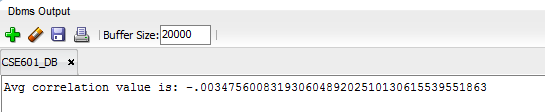
z number:=0;

begin

z:=TWO\_DS\_CORR('7154','ALL','AML');

dbms\_output.put\_line('Avg correlation value is: '||z);

end;



**Results for Project Description Part III**

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.

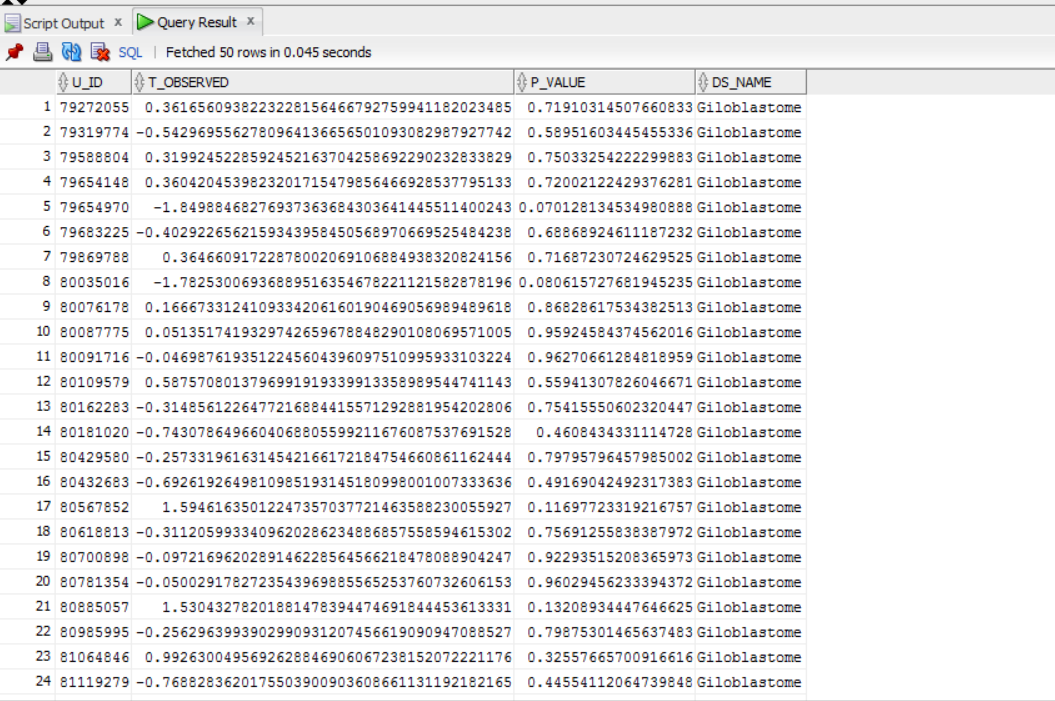
3. If the p-value of the t-test is smaller than 0.01, this gene is regarded as an “informative” gene.

Solution:

Begin

generate\_gene\_sig\_data;

End;

****

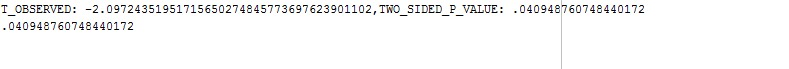
2) Use informative genes to classify a new patient (five test cases in test\_samples.txt are given in the data). For example, given a new patient PN, we want to predict whether he/she has “ALL”. 1) Find the informative genes w.r.t. “ALL”. 2) Find all the patients with “ALL” (group A). 3) For each patient 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”

Solution:

BEGIN

    DBMS\_OUTPUT.PUT\_LINE(new\_patient\_classification('TEST2','AML'));

END;



**User Interface:**

