Parallel and distributed systems

SQL assignment

Martí Municoy, Jorge Pardillos

Q0. Can you describe the series of steps to open a database for querying?

1. Open mysql: *mysql -u root -p* (and type your password).
2. Type *show databases;* in order to explore the available databases.
3. Type *use A;* in order to start querying the database "A".

Q1. What is the purpose of this query?
SELECT * from Sources;

It shows the all the entries from the table called sources.

Q2. Get 5 GenBank ids and corresponding descriptions.

SELECT gbId, description FROM Descriptions LIMIT 5;

Q3. What is the purpose of this query?
SELECT count(*) from LocusLinks;

It gives the number of entries in the table LocusLinks;

Q4. How many different Affy ids are in the expression data?

SELECT count(*) FROM Data;

Q5. What is the expression level of Affy id U95-32123_at in experiment number 1?

SELECT level FROM Data WHERE affyId="U95-32123_at"AND expt=1;

Q6. Find all the gene descriptions, along with their GenBank ids containing the word “Human”?

```
SELECT * FROM Descriptions WHERE Descriptions LIKE “%Human%”;
```

Q7. What Gene Ontology descriptions (and corresponding accession) contain the phrase “protein kinase”? Answer should be provided in ascending order of accessions.

```
SELECT * FROM GO_Descr WHERE description LIKE “%protein kinase%” ORDER BY goAcc ASC;
```

Q8. Which AffyId of table Data correspond to sequences in Targets table with the phrase “kinase” in their description? Use the following command:
LOAD DATA INFILE ‘file.tsv’ INTO TABLE Targets;
To add a new entry in Descriptions with the string “kinase” and the gbId= “M18228”.
Now repeat the query again

```
SELECT Data.affyId
FROM Data, Targets, Descriptions
WHERE Data.affyId=Targets.affyId AND Targets.gbId=Descriptions.gbId AND Descriptions.description
LIKE “%kinase%”;
```

This query gives 0 results.

We introduce one extra entry to the table descriptions: *load data infile ‘new.tsv’ into table Descriptions;*. The file is in the directory /var/lib/mysql/experiments/.

Repeating the same query again is still showing 0 results. There is an affyId corresponding to the gbId that we have introduced (M18228), but that affyId is not in the Data table.

Q9. Get two affyId, uId and descriptions in LocusDescr in reverse alphabetical order of descriptions

```
SELECT DISTINCT Data.affyId, UniSeqs.uId, UniDescr.description
FROM Data, UniSeqs, UniDescr, Targets
WHERE UniDescr.uId=UniSeqs.uId AND UniSeqs.gbId=Targets.gbId AND Targets.affyId =Data.affyId
ORDER BY UniDescr.description DESC
LIMIT 2;
```

Q10. How would you find the average expression level of each experiment in Data?

```
SELECT exptId , AVG(level) FROM Data GROUP BY exptId;
```

Q11. What is the average expression level of each array probe (affyId) across all experiments?

```
SELECT affyId, AVG(level) FROM Data GROUP BY affyId;
```

Q12. What is the purpose of the following query?

```
SELECT Data.affyId, Data.level, Data.exptId, DataCopy.affyId,
DataCopy.level, DataCopy.exptId
FROM Data, Data DataCopy
WHERE Data.level > 10 * DataCopy.level
AND Data.affyId=DataCopy.affyId
AND Data.affyId LIKE “AFFX%” LIMIT 10;
```

From the table called “Data” it takes the entries with an affy-name beginning with the string “AFFX” and selects all of them for which the level number is 10 times bigger for an experiment than for some other experiments of the same affy-name.

Q13. Write a query to provide three different descriptions for all gbId in table Targets

There are four tables containing descriptions. Therefore there are 4 possible combinations and 4 possible outputs, depending on which three tables are chosen (Descriptions, LocusDescr, GO_Descr or UniDescr).

One option will be:

```
SELECT Targets.gbId, Descriptions.description, LocusDescr.description, UniDescr.description
FROM Targets, Descriptions, LocusDescr, UniDescr, LocusLinks, UniSeqs
WHERE Targets.gbId=Descriptions.gbId AND Targets.gbId=LocusLinks.gbId
AND LocusLinks.linkId=LocusDescr.linkId AND Targets.gbId=UniSeqs.gbId
AND UniSeqs.uId=UniDescr.uId;
```

Q14. Write a query to provide all gene ontology (GO_descr) descriptions related with all species in table Species sorted alphabetically and providing the first five results. Export the query to a tab-separated-file with the command:
SELECT * FROM TABLE INTO OUTFILE (‘data.out’);

```
SELECT * FROM (
SELECT MY.species, GO_Descr.description FROM GO_Descr, (
SELECT LocusDescr.linkId AS linkId, LocusDescr.species FROM LocusDescr
UNION ALL
SELECT LocusLinks.linkId, Targets.species FROM LocusLinks, Targets
WHERE Targets.gbId=LocusLinks.gbId
) AS MY, Ontologies WHERE Ontologies.linkId=MY.linkId AND GO_Descr.goAcc=Ontologies.goAcc
ORDER BY species ASC LIMIT 5
) AS final INTO OUTFILE ’/tmp/14.out’;
```