Name – Sithalechumi Narayanan
Date – 10/07/2015
BINF 650 – Fall 2015
Homework – 5

1. I want to get the average grade for (John) Goodman and (Michael) Caine. Below are two different attempts at this and they produce two different answers. Indicate which the correct command is. Write a (very) brief description on what the incorrect command is doing.

```
SELECT a.lastname , AVG(m.grade)
FROM movies m, isin i, actors a
WHERE m.mid=i .mid AND i .actor=a.aid
AND a.lastname='Goodman' OR lastname='Caine'
GROUP BY a.lastname;

SELECT a.lastname , AVG(m.grade)
FROM movies m, isin i, actors a
WHERE m.mid=i.mid AND i.actor=a.aid
AND a.lastname IN ('Caine','Goodman')
GROUP BY a.lastname ;
```

Average grade for Michael Caine is 5.6667 and John Goodman is 6.7778
The correct command to get this value is the second one.

In the first command the condition is a.lastname= 'Goodman' or lastname='Caine'. It doesn't look for the lastname either Goodman or Caine, but it looks at the last name being (**Goodman AND m.mid=i.mid AND i.actor=a.aid** ) OR (**lastname='Caine'**). There is no join for Lastname as 'Caine', therefore it does a Cartesian product of 1801872 records for 'Caine', hence the wrong average.

2. Find the movies (mid and title) with both Matthew Broderick and Jean Reno in them. (You do not have to use first names since no other actors have these last names).

```
mysql> SELECT DISTINCT isin.mid, movies.title FROM actors, isin, movies
WHERE actors.aid = isin.actor AND isin.mid = movies.mid AND isin.mid IN
(SELECT a.mid FROM isin a, isin b WHERE a.mid = b.mid AND a.actor =
(SELECT aid FROM actors WHERE lastname = 'Broderick') AND b.actor =
(SELECT aid FROM actors WHERE lastname ='Reno'));
+------+----------+
| mid  | title    |
+------+----------+
|  276 | Godzilla |
+------+----------+
1 row in set (1.97 sec)
```

3. Find all of the movies (mid and title) which have two actors named John. (Note there is an error in the database and John Cleese is listed as being in Life of Brian twice. This will show up as a positive response in your query. Do not worry about it and you do not need to remove this movie from your answer.

```
mysql> SELECT movies.mid, movies.title FROM actors, isin, movies WHERE
movies.mid = isin.mid AND isin.actor = actors.aid AND actors.firstname
= 'John' GROUP BY movies.mid, movies.title HAVING COUNT(*) = 2;
+-----+--------------------------+
| mid | title                    |
+-----+--------------------------+
|  28 | Con Air                  |
|  88 | O Brother, Where Are Thou |
| 263 | Life of Brian            |
| 314 | The Blues Brothers       |
| 347 | Shadows and Fog          |
| 497 | 1941                     |
| 572 | Stripes                  |
+-----+--------------------------+
7 rows in set (0.01 sec)
```

4. (This is the same question as in a previous HW, but now you must use only a single MySQL command). Find the author listed for AE004091 that is published in other journals. The return should be the author's pid and the number of journals where he is published. The return should not include any other author.

```
mysql> SELECT published.aid, count(genome.journal) FROM
published,genome WHERE genome.accession = published.accession AND
published.aid IN (SELECT aid FROM published WHERE accession =
'AE004091') AND genome.accession != 'AE004091';
+------+-----------------------+
| aid  | count(genome.journal) |
+------+-----------------------+
|  418 |                     2 |
+------+-----------------------+
1 row in set (0.19 sec)
```

5. In this database there are only a few genes with 3 splices and they are all from the same genome. Return the organism name for the genome that has genes with 3 splices.

```
mysql> SELECT DISTINCT genome.organism FROM genome, protein, (SELECT
pid FROM splices GROUP BY pid HAVING COUNT(*) = 3) AS s_query WHERE
protein.accession = genome.accession AND s_query.pid = protein.pid;
+-------------------------------+
| organism                      |
+-------------------------------+
| Streptomyces coelicolor A3(2) |
+-------------------------------+
1 row in set (0.07 sec)
```

http://dev.mysql.com/doc/refman/5.6/en/subquery-optimization.html
Based on the explanation given under 8.2.1.18.3 Optimizing Derived Tables (Subqueries) in the FROM Clause.

6. In genome AE004091 there are several genes. Consider the DNA that starts at the beginning of the first gene and ends at the end of the last gene. Within this DNA there are

regions that are used for coding and regions that are between genes (or splices). What is the percentage of this DNA that is used for coding? Consider a small example: The DNA is 100 bases and there are only 2 genes. The first starts at 1 and ends at 40. The second gene starts at 50 and ends at 100. 90 bases are used in coding and so the coding percentage is 90/100 = 0.90.

```
mysql> SELECT start, stop, stop - start + 1 AS 'Total Length',
codingRegion,(codingRegion /(stop - start + 1)) * 100   AS 'Percentage
Of DNA' FROM((SELECT start FROM splices WHERE pid = (SELECT pid FROM
protein WHERE accession = 'AE004091'  LIMIT 1)) AS start), ((SELECT
stop FROM splices WHERE pid = (SELECT pid FROM protein WHERE accession
= 'AE004091' ORDER BY pid DESC LIMIT 1)) AS stop), ((SELECT SUM(stop -
start) AS codingRegion FROM splices WHERE pid IN (SELECT pid FROM
protein WHERE accession = 'AE004091')) AS codingRegion);
+-------+---------+--------------+--------------+-------------------+
| start | stop    | Total Length | codingRegion | Percentage Of DNA |
+-------+---------+--------------+--------------+-------------------+
|   483 | 6264361 |      6263879 |      5593747 |           89.3016 |
+-------+---------+--------------+--------------+-------------------+
1 row in set (0.27 sec)
```