Data Storage and Retrieval
Final Exam
Fall 2016

D. Gillman

December 6, 2016

Due 8pm, Thursday, December 8, 2016

The exam is in two parts.
Part I is intended as a 90-minute exam.
Part II involves programming and is open-ended.

Work by yourself.
Show your work and explain your reasoning.

Part I (60 points)

This part is closed-book, with lifelines. If you need to look something up, please do so and note what you looked up.

1.
   a. What are the defining characteristics of a primary key in a relational database table?

   b. Consider the hubway trips table below.
      i. What is the primary key? What is a foreign key?

      ii. Say that a table is *normalized* if the value of any one column cannot be determined by the value of another column that is not the primary key. Show that this table is not normalized.

      iii. Revise the schema so that there are two tables that are normalized in the above sense, and identify their primary and foreign keys.

```
trip_id                 bigint,
status                  text,
duration                integer,
start_date              timestamp,
strt_statn              integer,
end_date                timestamp,
end_statn               integer,
bike_nr                 text,
user_id                 bigint,
user_subscription_type  text,
user_zip_code           text,
user_birth_date         integer,
user_gender             text
```

2. Consider the authors table in the fanfiction database:

```
  Column  |       Type        | Modifiers
----------+-------------------+-----------
 author   | character varying |
 id       | integer           | not null
 country  | character varying |
 joindate | date              |
Indexes:
    "authors_pkey" PRIMARY KEY, btree (id)
    "authors_author_key" UNIQUE CONSTRAINT, btree (author)
    "author_cty_idx" btree (country)
    "author_jd_idx" btree (joindate)
```

and these two queries:

```
fanfiction=> explain select * from authors where author like '%L';
                          QUERY PLAN
---------------------------------------------------------------
 Seq Scan on authors  (cost=0.00..5264.04 rows=1997 width=27)
   Filter: ((author)::text ~~ '%L'::text)
(2 rows)

fanfiction=> explain select * from authors where author like 'L%';
                                    QUERY PLAN
-------------------------------------------------------------------------------------
 Bitmap Heap Scan on authors  (cost=593.75..3510.28 rows=9984 width=27)
   Filter: ((author)::text ~~ 'L%'::text)
   ->  Bitmap Index Scan on authors_author_key  (cost=0.00..591.25 rows=9883 width=0)
         Index Cond: (((author)::text >= 'L'::text) AND ((author)::text < 'M'::text))
```

a. What does the number `9883` represent and how is it derived by the postgres query planner?

b. What does `591.25` represent and how is it derived? (You don't need to remember the precise factors in the formula but please describe the index structure and its role in the derivation.)

c. In the second query plan, describe what the bitmap heap scan and filter do.

d. Explain why the query planner did not choose to use an index for the first query.

3.

a. Match the query to the result, without running the queries. Give your reasoning. (If you run the queries as a lifeline you must still give your reasoning.)

I.
```
select * from movies where title % 'anatomy of a murder' limit 5;
```

II.
```
select * from movies where title @@ 'of a murder' limit 5;
```

III.
```
select * from movies
where cube_distance(genre,'(0, 0, 0, 5, 5, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0)')<2;
```

IV.
```
select * from movies where metaphone(title, 3) = metaphone('Anatomy of a Murder',3);
```

V.
```
select * from movies where title ilike '%a murder' limit 5;
```

A.

```
 movie_id |        title         |                          genre
----------+----------------------+-------------------------------------------------------------
       21 | Anatomy of a Murder  | (0, 0, 0, 5, 5, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0)
      156 | Dial M For Murder    | (0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0)
      270 | Murder, She Said     | (0, 0, 0, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
      271 | Murder at the Gallop | (0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
      275 | Murder Most Foul     | (0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
(5 rows)
```

B.

```
 movie_id |        title        |                          genre
----------+---------------------+-------------------------------------------------------------
       21 | Anatomy of a Murder | (0, 0, 0, 5, 5, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0)
(1 row)
```

C.

```
 movie_id |          title           |                          genre
----------+--------------------------+-----------------------------------------------------------
       21 | Anatomy of a Murder      | (0, 0, 0, 5, 5, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0)
      866 | And Then There Were None | (0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
     2575 | And Soon the Darkness    | (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 5, 0)
(3 rows)
```

D.

```
 movie_id |        title        |                         genre
----------+---------------------+-----------------------------------------------------------
       21 | Anatomy of a Murder | (0, 0, 0, 5, 5, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0)
      121 | Miller's Crossing   | (0, 0, 0, 5, 5, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0)
      144 | Wild at Heart       | (0, 0, 0, 5, 5, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0)
     1368 | Legal Eagles        | (0, 0, 0, 5, 5, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0)
(4 rows)
```

E.

```
 movie_id |        title        |                         genre
----------+---------------------+-----------------------------------------------------------
       21 | Anatomy of a Murder | (0, 0, 0, 5, 5, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0)
      276 | Murder Ahoy!        | (0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
      515 | Murder My Sweet     | (0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
      553 | A Perfect Murder    | (0, 0, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0)
     1217 | Murder at 1600      | (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0)
(5 rows)
```

b. Consider the table

```
  Table "public.movies_actors"
   Column  |  Type   | Modifiers
 ----------+---------+-----------
  movie_id | integer | not null
  actor_id | integer | not null
 Indexes:
     "movies_actors_movie_id_actor_id_key" UNIQUE CONSTRAINT, btree (movie_id, actor_id)
     "movies_actors_actor_id" btree (actor_id)
     "movies_actors_movie_id" btree (movie_id)
 Foreign-key constraints:
     "movies_actors_actor_id_fkey" FOREIGN KEY (actor_id) REFERENCES actors(actor_id)
     "movies_actors_movie_id_fkey" FOREIGN KEY (movie_id) REFERENCES movies(movie_id)
```

Give one query that uses an outer join to test whether the first foreign key constraint is violated, and give another query that does not use an outer join to test whether the second foreign key constraint is violated.

4. Create HBase schemas for the following data.
    a. A social media site has users with id, name, sex, and age attributes; the users have **friendships** of different types (acquaintance, family, colleague, etc.) with other users. Give the rowkey, column families and columns for a single table to contain this information.

    b. A college database contains the following students with id, name, sex, and age attributes; courses with id, title, introduction, and teacher id attributes, and course enrollments of different types (for-credit, audit) with student id, course id attributes. Show how to create two tables to contain this information.

5. MongoDB
    a. Identify three documents, one cursor, and one collection in this shell interaction:

```
> var disp = db.phones.find( )
> disp.limit(1)
{
       "_id" : 58005550000,
       "components" : {
              "country" : 5,
              "area" : 800,
              "prefix" : 555,
              "number" : 5550000
       },
       "display" : "+5 800-5550000"
}
> disp.count()
200001
> var disp = db.phones.find({ "components.number" : {$gt : 5550012} }, { "_id" :
0, "display" : 1} )
> disp.limit(1)
{ "display" : "+4 800-5550013" }
> disp.count()
```

    b. I didn't show the output of the last command. What will it be, assuming the `phone` numbers are consecutive?

    c. Write a command to change the value of "country" for 800-5550000 to include both a country code and a country name. Recall that modifying an existing field requires the `$set` modifier: `update(condition, {$set : contents})`.

Part II (40 points)

Do two of the following three problems.

1. The `country` collection contains country name and country code. Write a procedure for merging `country` and `phone` data into one collection.

2. I generated the hbase 'wiki' table on compscio1 from the file

/usr/share/databases/SevenDatabases/code/hbase/wikipedia-pages-part13.xml.bz2.

Familiarize yourself with the table, checking the articles in the table against the articles on Wikipedia. Some of the pages are articles, and some are not: files, templates, internal Wikipedia notes, etc.

Write a python program to read the table and create a new table 'wikistats_yourname' with column families 'link_histogram' and 'page_histogram'. Create one histogram of column:value pairs for each column family:

i. Number of links per article
ii. Number of pages of each type: article, template, etc.

Start by copying the following sample script and hbase module directory to your own working directory:

/usr/share/databases/SevenDatabases/code/hbase/thrift/sample_client.py
/usr/share/databases/SevenDatabases/code/hbase/thrift/hbase

Use these Cloudera blogs for reference:

http://blog.cloudera.com/blog/2013/09/how-to-use-the-hbase-thrift-interface-part-1/
http://blog.cloudera.com/blog/2013/12/how-to-use-the-hbase-thrift-interface-part-2-insertinggetting-rows/

3. Download the Yelp Dataset Challenge data from this folder or from the website. Read the sections called **The Challenge** and **Notes on the Dataset**.

   Yelp wants your help solving the following problem: add a feature to the Yelp website listing the businesses of one type nearest to a given business of another type -- for example, the doctors nearest to a given restaurant -- in order of increasing distance.

   Solve as much of the problem as you can, and sketch how your part of the solution would be combined with other parts to make a complete solution.