COMP 3421 Database Organization & Mgmt I

Professor Andrew Hannum

Assignment 3

Cheng Zhang

2019/5/14

Step 3 of Your PDA (Personal Database Application)

Note1: this assignment is a slight modification of material developed by the Stanford Database Group

Note2: Remember to back up your work!

In this phase of your PDA you will make sure your relations have keys and foreign keys, create a substantial amount of data for you databases, and load your database with this data using the mySQL "load" command. First, make sure that your relations have keys and that the relations created from ER Model relationships use foreign keys to specify the keys of the relations you are referencing.

To create the data, you have multiple options:

1. Write a program in any programming language you like that creates large files of records in a format acceptable to the load command, then load the data into your PDA relations.
   * If you are using real data for your PDA, your program will need to transform the data into files of records conforming to your PDA schema.
   * If you are generating fake data, it is up to you to make sure your data conforms to database requirements (read below)
2. Use a data mocking tool to fabricate data: you should generate either random or nonrandom (e.g., sequential) records conforming to your schema. It is both fine and expected for your data values--strings especially--to be meaningless gibberish. The point of generating large amounts of data is so that you can experiment with a database of realistic size, rather than the small "toy" databases often used in classes. The data you generate and load should be on the order of:

At least one relations with tens-of-thousands of tuples  
At least two additional relations with thousands of tuples

If the semantics of your application includes relations that are expected to be relatively small (e.g., schools within a university), it is fine to use some small relations, but please ensure that you also have relations of the sizes prescribed above as well. When fabricating data via any method, there are two important points to keep in mind:

\*Make sure you do not generate duplicate values for attributes that serve a keys.\*

Your PDA almost certainly includes relations that are expected to join with each other. For example, you may have a Student relation with attribute courseNo that's expected to join with attribute number in relation Course. In generating data, be sure to generate values that actually do join--otherwise all of your interesting queries will have empty results! One way to guarantee join-ability is to generate the values in one relation, then use the generated values in one relation to select joining values for the other relation. For example, you could generate course numbers first (either sequentially or randomly), then use these numbers to fill in the courseNo values in the student relation.

Turn in the following:

1. Any code you write for generating or transforming data

2. A sample (small, not the entire file) datafile. You can contact the TA if you are having trouble generating data.

3. The output of:

1) "show tables";

2) a "describe table\_name" for each table "table\_name"; and

3) a count(\*) on each of your tables; and

4) the output of a select command on each table that returns less than 10 tuples. The output would look something the sample output below. Please upload a pdf file to Canvas.

Use the "load" command to load your data files (if they are CSV, or similar). Here is an example of my load commands used to load the relations:

load data local infile '/Users/leut/data\_sailors' into table sailors  
fields terminated by ','  
lines terminated by '\n'  
;

load data local infile '/Users/leut/data\_boats' into table boats  
fields terminated by ','  
lines terminated by '\n'  
;

load data local infile '/Users/leut/data\_reserve' into table reserve  
fields terminated by ','  
lines terminated by '\n'  
;

**Answer:**

mysql> show tables;

+----------------------+

| Tables\_in\_my\_hotel\_2 |

+----------------------+

| bills |

| companies |

| customers |

| hotels |

| rooms |

| staffs |

+----------------------+

6 rows in set (0.00 sec)

mysql> select count(\*) from bills;

+----------+

| count(\*) |

+----------+

| 1500 |

+----------+

1 row in set (0.00 sec)

mysql> select count(\*) from companies;

+----------+

| count(\*) |

+----------+

| 5 |

+----------+

1 row in set (0.00 sec)

mysql> select count(\*) from customers;

+----------+

| count(\*) |

+----------+

| 1499 |

+----------+

1 row in set (0.00 sec)

mysql> select count(\*) from hotels;

+----------+

| count(\*) |

+----------+

| 12 |

+----------+

1 row in set (0.00 sec)

mysql> select count(\*) from rooms;

+----------+

| count(\*) |

+----------+

| 10500 |

+----------+

1 row in set (0.00 sec)

mysql> select count(\*) from staffs;

+----------+

| count(\*) |

+----------+

| 500 |

+----------+

1 row in set (0.00 sec)

mysql> describe bills;

+----------+---------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+----------+---------+------+-----+---------+-------+

| bill\_id | int(11) | NO | PRI | NULL | |

| room\_id | int(11) | NO | MUL | NULL | |

| staff\_id | int(11) | NO | MUL | NULL | |

| price | int(11) | YES | | NULL | |

+----------+---------+------+-----+---------+-------+

4 rows in set (0.00 sec)

mysql> describe companies;

+------------+--------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+------------+--------------+------+-----+---------+-------+

| company\_id | int(11) | NO | PRI | NULL | |

| name | varchar(255) | YES | | NULL | |

+------------+--------------+------+-----+---------+-------+

2 rows in set (0.01 sec)

mysql> describe customers;

+-------------+--------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------------+--------------+------+-----+---------+-------+

| customer\_id | int(11) | NO | PRI | NULL | |

| name | varchar(255) | YES | | NULL | |

| age | int(11) | NO | | NULL | |

| gender | varchar(255) | YES | | NULL | |

| hotel\_id | int(11) | NO | MUL | NULL | |

| room\_id | int(11) | NO | MUL | NULL | |

| bill\_id | int(11) | NO | MUL | NULL | |

+-------------+--------------+------+-----+---------+-------+

7 rows in set (0.00 sec)

mysql> describe hotels;

+------------+--------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+------------+--------------+------+-----+---------+-------+

| hotel\_id | int(11) | NO | PRI | NULL | |

| name | varchar(255) | YES | | NULL | |

| company\_id | int(11) | YES | MUL | NULL | |

+------------+--------------+------+-----+---------+-------+

3 rows in set (0.00 sec)

mysql> describe rooms;

+----------+---------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+----------+---------+------+-----+---------+-------+

| room\_id | int(11) | NO | PRI | NULL | |

| hotel\_id | int(11) | NO | MUL | NULL | |

| type | int(11) | YES | | NULL | |

| status | int(11) | YES | | NULL | |

+----------+---------+------+-----+---------+-------+

4 rows in set (0.00 sec)

mysql> describe staffs;

+----------+--------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+----------+--------------+------+-----+---------+-------+

| staff\_id | int(11) | NO | PRI | NULL | |

| name | varchar(255) | YES | | NULL | |

| age | int(11) | YES | | NULL | |

| gender | varchar(255) | YES | | NULL | |

| salary | int(11) | YES | | NULL | |

| hotel\_id | int(11) | NO | MUL | NULL | |

+----------+--------------+------+-----+---------+-------+

6 rows in set (0.00 sec)

mysql> select \* from bills where price < 200 ;

+---------+---------+----------+-------+

| bill\_id | room\_id | staff\_id | price |

+---------+---------+----------+-------+

| 231 | 197 | 364 | 161 |

| 642 | 5088 | 94 | 130 |

| 909 | 544 | 442 | 91 |

| 1278 | 10497 | 426 | 94 |

| 1333 | 3865 | 19 | 154 |

+---------+---------+----------+-------+

5 rows in set (0.00 sec)

mysql> select \* from companies where company\_id < 3 ;

+------------+---------+

| company\_id | name |

+------------+---------+

| 0 | Gabcube |

| 1 | Tekfly |

| 2 | Abata |

+------------+---------+

3 rows in set (0.00 sec)

mysql> select \* from customers where age = 6 ;

+-------------+--------------------+-----+--------+----------+---------+---------+

| customer\_id | name | age | gender | hotel\_id | room\_id | bill\_id |

+-------------+--------------------+-----+--------+----------+---------+---------+

| 6 | Darlleen Verissimo | 6 | Female | 9 | 468 | 285 |

| 58 | Fredek Fawkes | 6 | Male | 1 | 327 | 375 |

| 112 | Griselda Duerden | 6 | Female | 4 | 367 | 117 |

| 259 | Meris Iskowitz | 6 | Female | 7 | 33 | 320 |

| 311 | Ross Aishford | 6 | Male | 9 | 224 | 382 |

| 401 | Sydney Linskill | 6 | Female | 6 | 347 | 303 |

| 487 | Celestia Attle | 6 | Female | 1 | 420 | 258 |

| 742 | Rene Gosker | 6 | Male | 4 | 4438 | 1341 |

| 1076 | Oren Kelby | 6 | Male | 5 | 1843 | 939 |

| 1110 | Erick Blesdill | 6 | Male | 9 | 6260 | 1292 |

+-------------+--------------------+-----+--------+----------+---------+---------+

10 rows in set (0.00 sec)

mysql> select \* from hotels where company\_id < 2 ;

+----------+----------+------------+

| hotel\_id | name | company\_id |

+----------+----------+------------+

| 5 | Katz | 0 |

| 4 | Brainbox | 1 |

+----------+----------+------------+

2 rows in set (0.00 sec)

mysql> select \* from rooms where room\_id < 2 ;

+---------+----------+------+--------+

| room\_id | hotel\_id | type | status |

+---------+----------+------+--------+

| 0 | 1 | 5 | 0 |

| 1 | 11 | 1 | 0 |

+---------+----------+------+--------+

2 rows in set (0.00 sec)

mysql> select \* from staffs where age < 19 ;

+----------+--------------+------+--------+--------+----------+

| staff\_id | name | age | gender | salary | hotel\_id |

+----------+--------------+------+--------+--------+----------+

| 75 | Thoughtworks | 18 | Male | 24461 | 7 |

| 93 | Shufflester | 18 | Male | 42103 | 11 |

| 139 | Kimia | 18 | Female | 23004 | 3 |

| 230 | Oodoo | 18 | Male | 45818 | 3 |

| 266 | Oozz | 18 | Male | 38966 | 0 |

| 332 | Zava | 18 | Male | 19036 | 10 |

| 395 | JumpXS | 18 | Male | 34465 | 7 |

| 462 | Avaveo | 18 | Male | 49092 | 1 |

| 473 | Tavu | 18 | Male | 42065 | 11 |

+----------+--------------+------+--------+--------+----------+

9 rows in set (0.00 sec)

mysql>