Step 5 of Your PDA

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

For a brief intro to mySQL scripting, check out this Script Programming Reference (http://www.ntu.edu.sg/home/ehchua/programming/sql/MySQL_Intermediate.html)

(10 pts.) Make sure you have foreign key constraints in your database. If you do not, add them, and show the create table statement that defines these foreign key constraints. Create an update that violates the foreign key constraint. List the update command and then show the output of running this update.

```
mysql> update hotels set company_id = 8 where hotel_id = 1;
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails
(`my_hotel_2`.`hotels`, CONSTRAINT `hotels_ibfk_1` FOREIGN KEY (`company_id`) REFERENCES
`companies` (`company_id`))
mysql>
```

(15 pts.) Write a mySQL procedure or function (your choice) and script that calls this procedure/function. This procedure/function should be logical for your database and should include one or more inputs and an output. After calling the procedure you should print out the results of the output. If you procedure/function makes multiple database state changes document the changes with before/after select commands showing a (small) set of data that is changed. Include prose (English) that describes what the procedure/function does.

```
mysql> DELIMITER ;
mysql> DELIMITER //
mysql> CREATE PROCEDURE get_someoneid_salary
    -> (IN id INT)
    -> BEGIN
    ->   SELECT salary FROM staffs
    ->   WHERE staff_id = id;
    -> END //
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;


mysql> call get_someoneid_salary(500);
+--------+
| salary |
+--------+
|  40000 |
+--------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql>
```

(10 pts.) Create a logging trigger (and table, if necessary) and demonstrate how it works with your database.

```
mysql> CREATE TABLE logger
    -> (logger_id         INT UNSIGNED AUTO_INCREMENT PRIMARY KEY
    -> ,logger_event      VARCHAR(50)
    -> ,logger_table      VARCHAR(50)
    -> ,logger_instring   VARCHAR(100)) ENGINE=MyISAM;
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER //
mysql> CREATE TRIGGER staffs_insert
    -> BEFORE INSERT ON staffs
    -> FOR EACH ROW
    -> BEGIN
    ->
    ->   INSERT INTO logger
    ->   VALUES(null
    ->         ,'insert'
    ->         ,'staffs'
    ->         ,new.name);
    -> END;
    -> //
```

```
mysql> DELIMITER //
mysql> CREATE TRIGGER staffs_insert
    -> BEFORE INSERT ON staffs
    -> FOR EACH ROW
    -> BEGIN
    ->
    ->   INSERT INTO logger
    ->   VALUES(null
    ->         ,'insert'
    ->         ,'staffs'
    ->         ,new.name);
    -> END;
    -> //
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;

mysql> INSERT INTO staffs VALUES (502, 'test2', 26, 'Male', 20000, 2) ;
Query OK, 1 row affected (0.01 sec)

mysql> select * from logger;
+-----------+--------------+--------------+-----------------+
| logger_id | logger_event | logger_table | logger_instring |
+-----------+--------------+--------------+-----------------+
|         1 | insert       | staffs       | test2           |
|         2 | insert       | staffs       | test2           |
+-----------+--------------+--------------+-----------------+
2 rows in set (0.00 sec)

mysql>
```

(Extra Credit 15 pts.) Create an index and show how it speeds up two types of queries: 1) a selection on a single relation; and 2) a selection that involves a join. Note, to get times that will be measurably different you will likely need to populate your databases with a larger data set.

Create index:

```
mysql> create index myindex2 on employees(first_name);
Query OK, 0 rows affected (0.95 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> create index myindex on salaries(salary);
Query OK, 0 rows affected (9.19 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> show index from employees;
+-----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
| Table     | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment |
+-----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
| employees |          0 | PRIMARY  |            1 | emp_no      | A         |      299556 |     NULL | NULL   |      | BTREE      |         |               |
| employees |          1 | myindex2 |            1 | first_name  | A         |        1266 |     NULL | NULL   |      | BTREE      |         |               |
+-----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
2 rows in set (0.00 sec)

mysql> show index from salaries;
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
| Table    | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
| salaries |          0 | PRIMARY  |            1 | emp_no      | A         |      299037 |     NULL | NULL   |      | BTREE      |         |               |
| salaries |          0 | PRIMARY  |            2 | from_date   | A         |     2838426 |     NULL | NULL   |      | BTREE      |         |               |
| salaries |          1 | myindex  |            1 | salary      | A         |      109703 |     NULL | NULL   |      | BTREE      |         |               |
+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
3 rows in set (0.00 sec)
```

This is search with index on a single relation:

```
select salary from salaries where salary < 60000;

|  59999 |
|  59999 |
|  59999 |
|  59999 |
|  59999 |
|  59999 |
|  59999 |
|  59999 |
|  59999 |
|  59999 |
|  59999 |
|  59999 |
|  59999 |
|  59999 |
|  59999 |
|  59999 |
|  59999 |
|  59999 |
+--------+
1348032 rows in set (0.85 sec)

mysql>
```

This is search without index on a single relation:

```
select salary from salaries where salary < 60000;
|  43427 |
|  43002 |
|  46543 |
|  49510 |
|  52868 |
|  58058 |
|  59307 |
|  49597 |
|  50783 |
|  51569 |
|  53001 |
|  55380 |
|  59420 |
|  40000 |
|  42140 |
|  42357 |
|  45702 |
|  46206 |
|  47429 |
|  49971 |
|  51182 |
|  55003 |
+--------+
1348032 rows in set (1.58 sec)

mysql>
```

This is search without index that involves a join

```
mysql> select first_name, salary from salaries s, employees e where s.emp_no = e.emp_no and salary <60000 and first_name like 'l%';


| Leon       |  43167 |
| Leon       |  45702 |
| Leon       |  47281 |
| Leon       |  49103 |
| Leon       |  51414 |
| Lobel      |  40000 |
| Lobel      |  40528 |
| Lobel      |  44778 |
| Lobel      |  47746 |
| Lobel      |  51681 |
| Lobel      |  55848 |
| Lobel      |  56748 |
| Lobel      |  57278 |
+------------+--------+
39836 rows in set (0.20 sec)

mysql>
```

This is search with index that involves a join:

mysql> select first_name, salary from salaries s, employees e where s.emp_no = e.emp_no and salary <60000 and first_name like 'l%';

```
| Lunjin     |  48100 |
| Lunjin     |  52390 |
| Lunjin     |  54116 |
| Lunjin     |  54445 |
| Lunjin     |  50764 |
| Lunjin     |  52286 |
| Lunjin     |  42593 |
| Lunjin     |  45356 |
| Lunjin     |  48228 |
| Lunjin     |  51883 |
| Lunjin     |  56095 |
| Lunjin     |  55710 |
| Lunjin     |  50071 |
| Lunjin     |  50742 |
| Lunjin     |  54358 |
| Lunjin     |  58404 |
| Lunjin     |  58351 |
+------------+--------+
39836 rows in set (0.11 sec)

mysql>
```