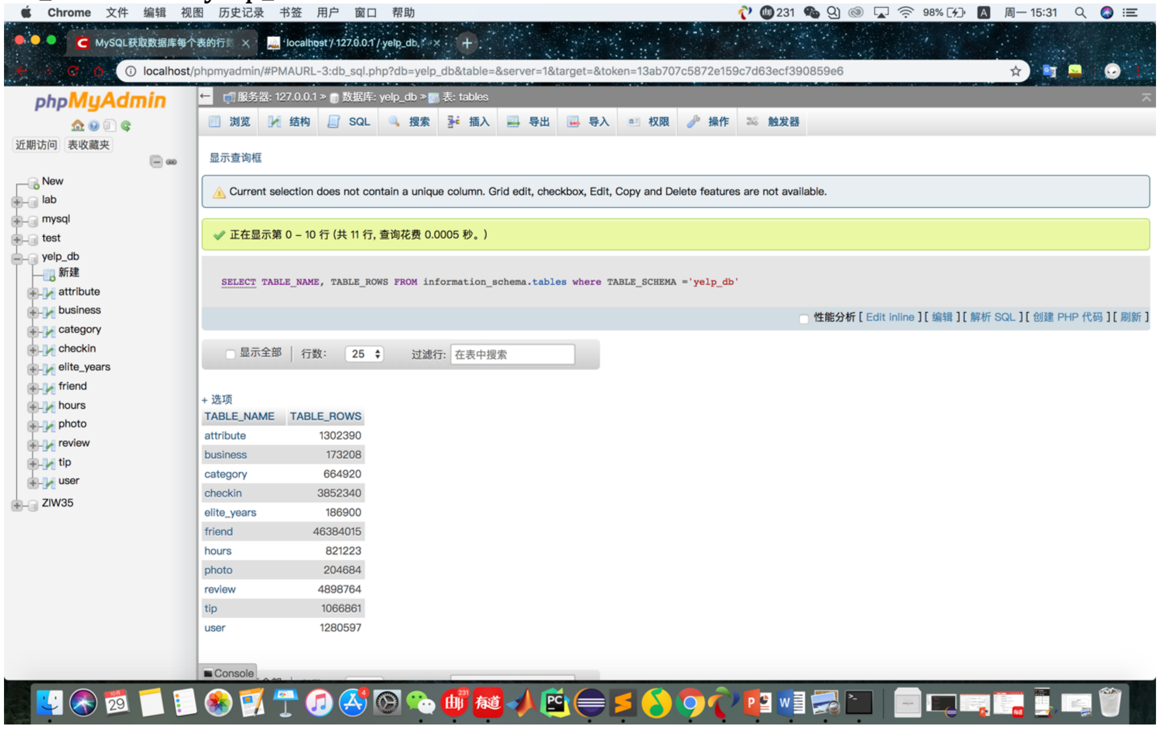
# INFSCI 2710 – Database Management – Fall 2018

## Homework 3 – Play Data!

1. Use the settings in lab session to connect to your MySQL and phpmyadmin
2. For all tasks, use the Yelp database to fulfill the requirements.
3. Attach your **<u>SQL statements + screenshot</u>** below each question and save it as Word/PDF (highly recommended) file
4. Name the Word document (PDF) with your answers as ***YourPittID_infsci2710_homework3.docx (pdf)***. In other words, if your Pitt ID (first part of your Pitt email) is abc123, your submission file should be named ***abc123_infsci2710_homework3.docx (pdf)***
5. Submit your work (SQL file + Document) via CourseWeb.

**Task 1 (10 points)**: Please download the Yelp dataset from: [https://shorturl.at/fhqLX](https://shorturl.at/fhqLX) . You should download a 2.71GB compressed file (yelp_db.tar) from the website. Please import the dataset to your MySQL server and **show the number of rows of each table in your 'yelp_db' database.**

SELECT TABLE_NAME, TABLE_ROWS FROM information_schema.tables where TABLE_SCHEMA ='yelp_db'



phpMyAdmin
服务器: 127.0.0.1 » 数据库: yelp_db » 表: tables
浏览 结构 SQL 搜索 插入 导出 导入 权限 操作 触发器
显示查询框

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

正在显示第 0 – 10 行 (共 11 行, 查询花费 0.0005 秒。)

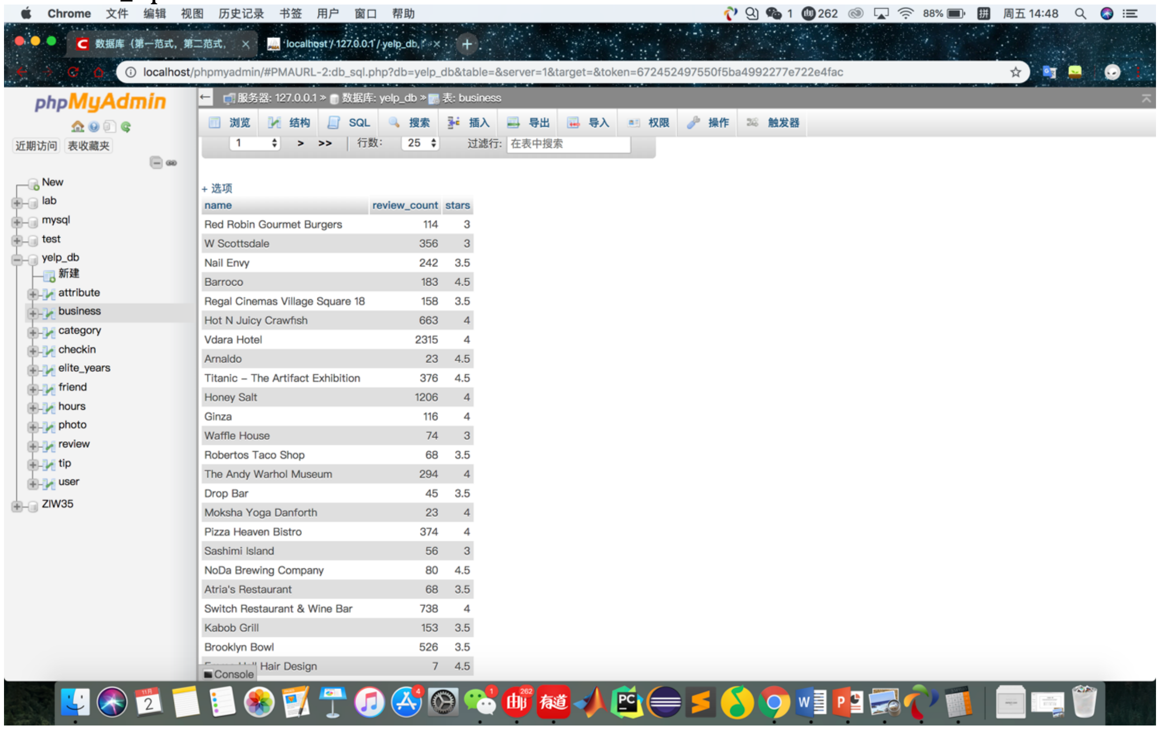
```
SELECT TABLE_NAME, TABLE_ROWS FROM information_schema.tables where TABLE_SCHEMA ='yelp_db'
```

| TABLE_NAME | TABLE_ROWS |
|---|---|
| attribute | 1302390 |
| business | 173208 |
| category | 664920 |
| checkin | 3852340 |
| elite_years | 186900 |
| friend | 46384015 |
| hours | 821223 |
| photo | 204684 |
| review | 4898764 |
| tip | 1066861 |
| user | 1280597 |

**Task 2 (10 points):** Please write a JOIN SQL code which shows three columns: 1) the business name, 2) the business review counts and 3) business stars with the conditions of

a. The businesses have received (at least one) reviews after Jan 1, 2016.
b. The businesses have received reviews from the users who have more than 10 fans.
c. The businesses are still open.

```
SELECT b.name,b.review_count, b.stars
FROM business as b JOIN review as r ON b.id = r.business_id join user as u ON
r.user_id = u.id
WHERE r.date > 2016-01-01
AND fans > 10
AND b.is_open=1
```



localhost/phpmyadmin/#PMAURL-2:db_sql.php?db=yelp_db&table=&server=1&target=&token=672452497550f5ba4992277e722e4fac

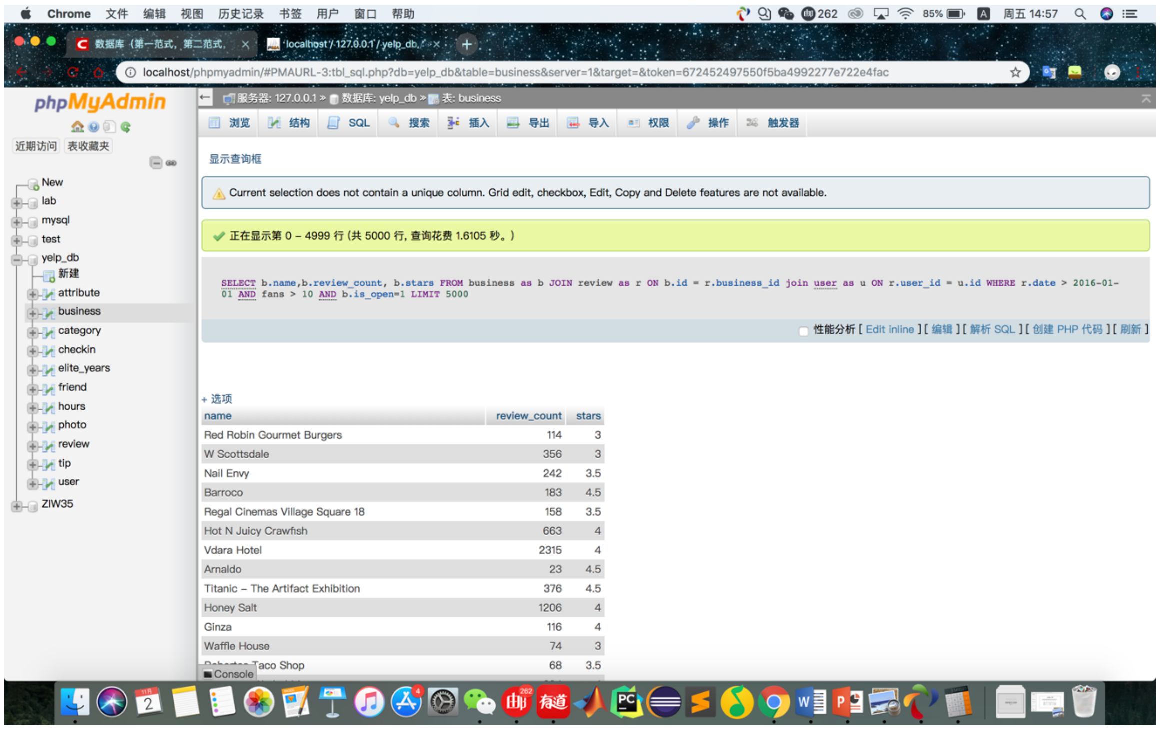
服务器: 127.0.0.1 » 数据库: yelp_db » 表: business

| name | review_count | stars |
|---|---|---|
| Red Robin Gourmet Burgers | 114 | 3 |
| W Scottsdale | 356 | 3 |
| Nail Envy | 242 | 3.5 |
| Barroco | 183 | 4.5 |
| Regal Cinemas Village Square 18 | 158 | 3.5 |
| Hot N Juicy Crawfish | 663 | 4 |
| Vdara Hotel | 2315 | 4 |
| Arnaldo | 23 | 4.5 |
| Titanic – The Artifact Exhibition | 376 | 4.5 |
| Honey Salt | 1206 | 4 |
| Ginza | 116 | 4 |
| Waffle House | 74 | 3 |
| Robertos Taco Shop | 68 | 3.5 |
| The Andy Warhol Museum | 294 | 4 |
| Drop Bar | 45 | 3.5 |
| Moksha Yoga Danforth | 23 | 4 |
| Pizza Heaven Bistro | 374 | 4 |
| Sashimi Island | 56 | 3 |
| NoDa Brewing Company | 80 | 4.5 |
| Atria's Restaurant | 68 | 3.5 |
| Switch Restaurant & Wine Bar | 738 | 4 |
| Kabob Grill | 153 | 3.5 |
| Brooklyn Bowl | 526 | 3.5 |
| [illegible] Hair Design | 7 | 4.5 |

**Task 3 (20 points):** In Task 2, how many return rows do you have in the query? How long does the SQL execution return the result? Please explain the 'execution time' on phpmyadmin (see the figure below) as well as the actual execution time (approximation is fine). Why does the actual execution time not equal to query execution time? Please explain your answer or ideas. [Hint: try to 'limit' the return rows to compare the difference]

```
SELECT b.name,b.review_count, b.stars
FROM business as b JOIN review as r ON b.id = r.business_id join user as u ON
r.user_id = u.id
WHERE r.date > 2016-01-01
AND fans > 10
AND b.is_open=1
```

LIMIT 5000



显示查询框

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

正在显示第 0 – 4999 行 (共 5000 行, 查询花费 1.6105 秒。)

```
SELECT b.name,b.review_count, b.stars FROM business as b JOIN review as r ON b.id = r.business_id join user as u ON r.user_id = u.id WHERE r.date > 2016-01-
01 AND fans > 10 AND b.is_open=1 LIMIT 5000
```

| name | review_count | stars |
|---|---|---|
| Red Robin Gourmet Burgers | 114 | 3 |
| W Scottsdale | 356 | 3 |
| Nail Envy | 242 | 3.5 |
| Barroco | 183 | 4.5 |
| Regal Cinemas Village Square 18 | 158 | 3.5 |
| Hot N Juicy Crawfish | 663 | 4 |
| Vdara Hotel | 2315 | 4 |
| Arnaldo | 23 | 4.5 |
| Titanic – The Artifact Exhibition | 376 | 4.5 |
| Honey Salt | 1206 | 4 |
| Ginza | 116 | 4 |
| Waffle House | 74 | 3 |
| [illegible] Taco Shop | 68 | 3.5 |

In task2, 689575 rows are returned. It returns these rows about 5-6 minutes. When I limit the number of rows to 5000, the execution time is 1.6105 s, but the actual execution time is about11.2800 s.

Exec time is when DbVisualizer requests the JDBC driver to execute the SQL until control is returned back to DbVisualizer. Fetch time is when DbVisualizer requests the JDBC driver to fetch the result set until it is returned to DbVisualizer. The actual execute time is that the result of the exec time plus fetch time, and fetch time is also influenced by the hardware of your computer.

**Task 4 (20 points):** In Task 2, the actual execution time, which takes minutes, is not acceptable in a real-world online system. How can you refine the database schema, so that you can reduce the actual execution time? Please explain your solution(s) and attach the SQLs for altering the database schema. Is your solution violating the rule of nominalization? [Hint: the data will be re-use quite often, you need to avoid the I/O cost.]

Because the table business and user only meet the NF2, I can change the two tables to meet the NF3. First, for table business, because only the “id”, “name”, “review_count”, “is_open”, “stars” should belong to this table, the others can be removed a new table “location”.

Create a new table “location”:

```
CREATE TABLE location(id int AUTO_INCREMENT NOT NULL PRIMARY
KEY,business_id varchar(22) NOT NULL, neighborhood varchar(255), address
varchar(255), city varchar(255), state varchar(255), postal_code varchar(255),
latitude float, longitude float, FOREIGN KEY (business_id) REFERENCES business(id)
ON DELETE CASCADE)
```

Insert the data to table location:

```
INSERT INTO `location`(`business_id`, `neighborhood`, `address`, `city`, `state`, `posta
l_code`, `latitude`, `longitude`) SELECT id, neighborhood, address,city, state, postal_c
ode, latitude, longitude FROM business
```

Delete the columns from table business:

```
ALTER TABLE business DROP neighborhood, DROP address, DROP city, DROP state,
DROP postal_code, DROP latitude, DROP longitude;
```

For table user, only the id, name, review_count, yelp_since, useful, cool, funny, fans, average_stars, should belong to table user, the others should be inserted into a new table compliment.

Create a new table compliment:

```
CREATE TABLE compliment(id int NOT NULL AUTO_INCREMENT PRIMARY KEY,
user_id varchar(22) NOT NULL, compliment_hot int(11), compliment_more int(11),
compliment_profile int(11), compliment_cute int(11), compliment_list int(11),
compliment_note int(11), compliment_plain int(11), compliment_cool int(11),
compliment_funny int(11), compliment_writer int(11), compliment_photos int(11),
FOREIGN KEY(user_id) REFERENCES user(id) ON DELETE CASCADE);
```

Insert the data to the table compliment:

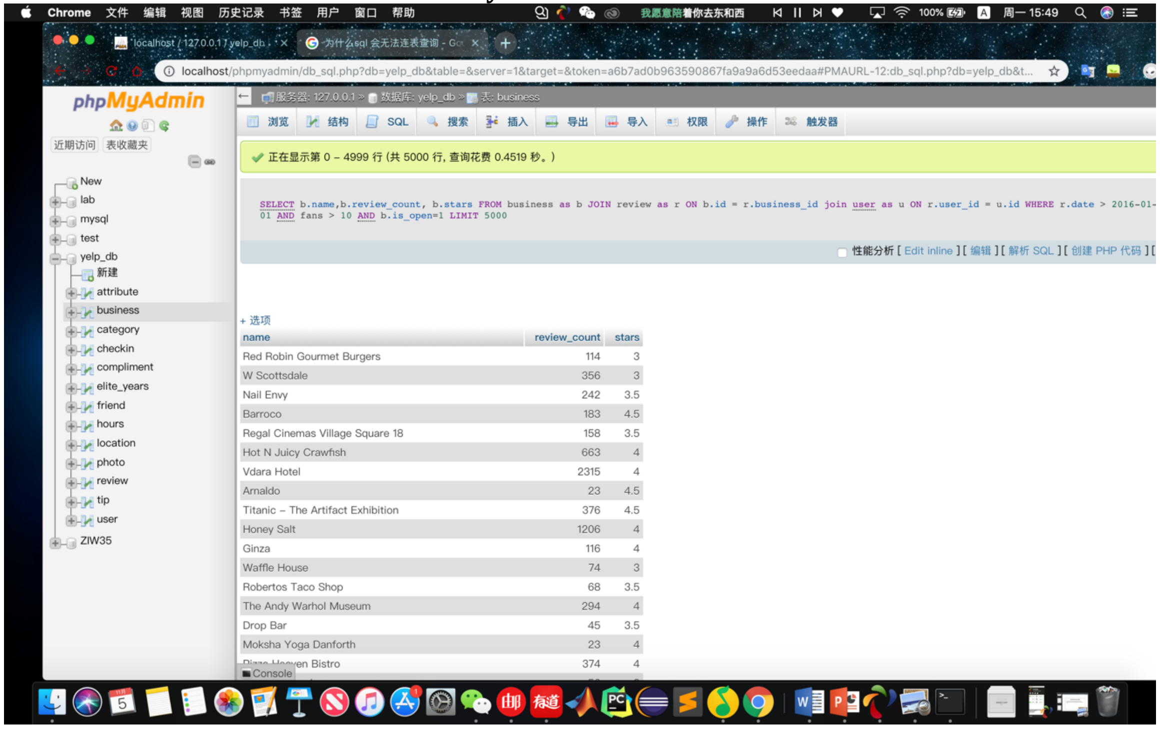
```
INSERT INTO `compliment`(`user_id`, `compliment_hot`, `compliment_more`, `compli
ment_profile`, `compliment_cute`, `compliment_list`, `compliment_note`,`compliment
_plain`, `compliment_cool`, `compliment_funny`, `compliment_writer`, `compliment_p
hotos`) SELECT id,compliment_hot, compliment_more,compliment_profile, complim
ent_cute, compliment_list, compliment_note, compliment_plain, compliment_cool, co
mpliment_funny,compliment_writer,compliment_photos FROM user
```

Delete the columns in table user:

```
ALTER TABLE user DROP COLUMN compliment_hot, DROP compliment_more,
DROP compliment_profile, DROP compliment_cute, DROP compliment_list, DROP
```

compliment_note, DROP compliment_plain, DROP compliment_cool, DROP compliment_funny, DROP compliment_writer, DROP compliment_photos;

Before change the tables' scheme, we need about 90s to execute. After these, we can reduce the execute time and we only need about 30s.



服务器: 127.0.0.1 » 数据库: yelp_db » 表: business

✓ 正在显示第 0 – 4999 行 (共 5000 行, 查询花费 0.4519 秒。)

```
SELECT b.name,b.review_count, b.stars FROM business as b JOIN review as r ON b.id = r.business_id join user as u ON r.user_id = u.id WHERE r.date > 2016-01-
01 AND fans > 10 AND b.is_open=1 LIMIT 5000
```

| name | review_count | stars |
|---|---|---|
| Red Robin Gourmet Burgers | 114 | 3 |
| W Scottsdale | 356 | 3 |
| Nail Envy | 242 | 3.5 |
| Barroco | 183 | 4.5 |
| Regal Cinemas Village Square 18 | 158 | 3.5 |
| Hot N Juicy Crawfish | 663 | 4 |
| Vdara Hotel | 2315 | 4 |
| Arnaldo | 23 | 4.5 |
| Titanic – The Artifact Exhibition | 376 | 4.5 |
| Honey Salt | 1206 | 4 |
| Ginza | 116 | 4 |
| Waffle House | 74 | 3 |
| Robertos Taco Shop | 68 | 3.5 |
| The Andy Warhol Museum | 294 | 4 |
| Drop Bar | 45 | 3.5 |
| Moksha Yoga Danforth | 23 | 4 |
| Pizza Heaven Bistro | 374 | 4 |

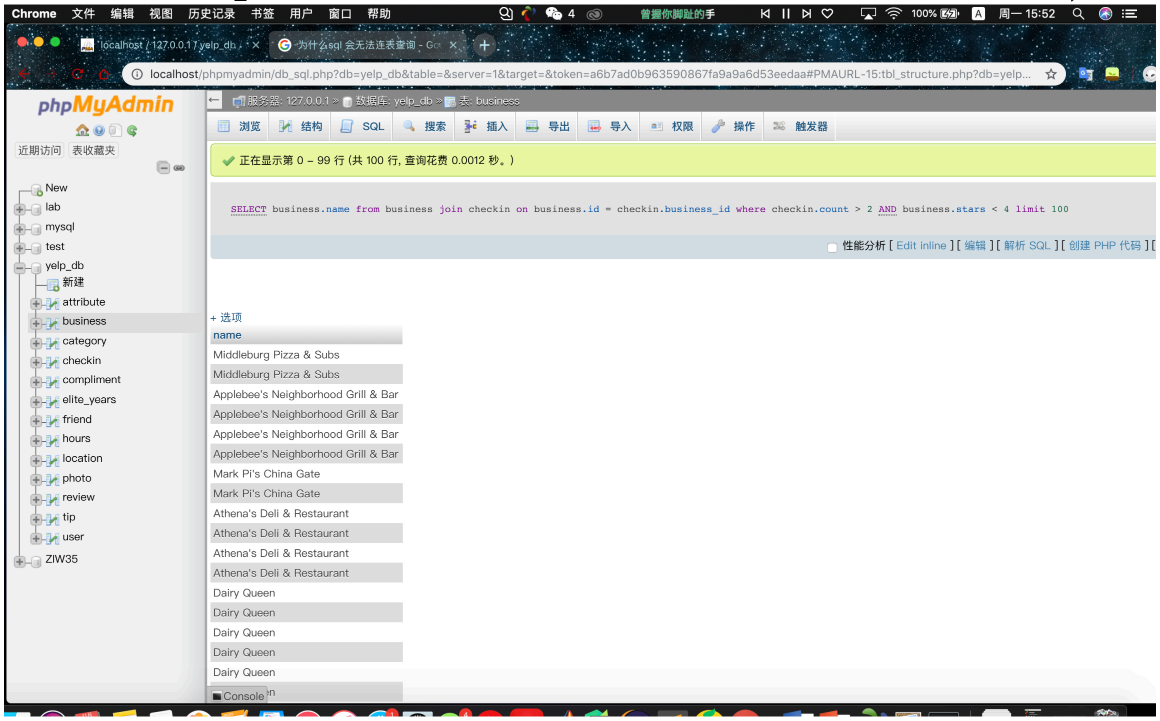
**Task 5 (20 points):** Same as Task 4, if you cannot change the database schema, as a web developer, how would you solve the long SQL query time issue? Please explain your solution(s) and the idea(s) of how to integrate your solution with the Yelp Data. [Hint: you may consider a suitable data structure or cache mechanism]

I think the cache mechanism is a good solution, it will reduce much of the executing time and if the table doesn't exit, the executing process won't go through the invalid table, it also reduce the executing time.

**Task 6 (20 points):** Please visit Yelp.com to find **two** social functions (the functions should be meaningful for delivering some useful information to the users) and propose the SQL statements (at least three tables should be used in each SQL) to retrieve the data. For example, you may propose a social function of 'top restaurant of the day', so you will need to write a SQL to select the businesses that have the most reviews on a specific day. Please explain your answer and discuss the actual query result (number of return rows and execution time, you will also need to show the result with scattershot) using Yelp data.

1.Find the business that the number of counts is over 2 and stars is less than 4.

SELECT business.name from business join checkin on business.id = checkin.business_id where  checkin.count > 2 AND business.stars < 4 limit 100;


phpMyAdmin
服务器: 127.0.0.1 » 数据库: yelp_db » 表: business
浏览 结构 SQL 搜索 插入 导出 导入 权限 操作 触发器
New
lab
mysql
test
yelp_db
新建
attribute
business
category
checkin
compliment
elite_years
friend
hours
location
photo
review
tip
user
ZIW35
性能分析 [ Edit inline ] [ 编辑 ] [ 解析 SQL ] [ 创建 PHP 代码 ] [
+ 选项
Console

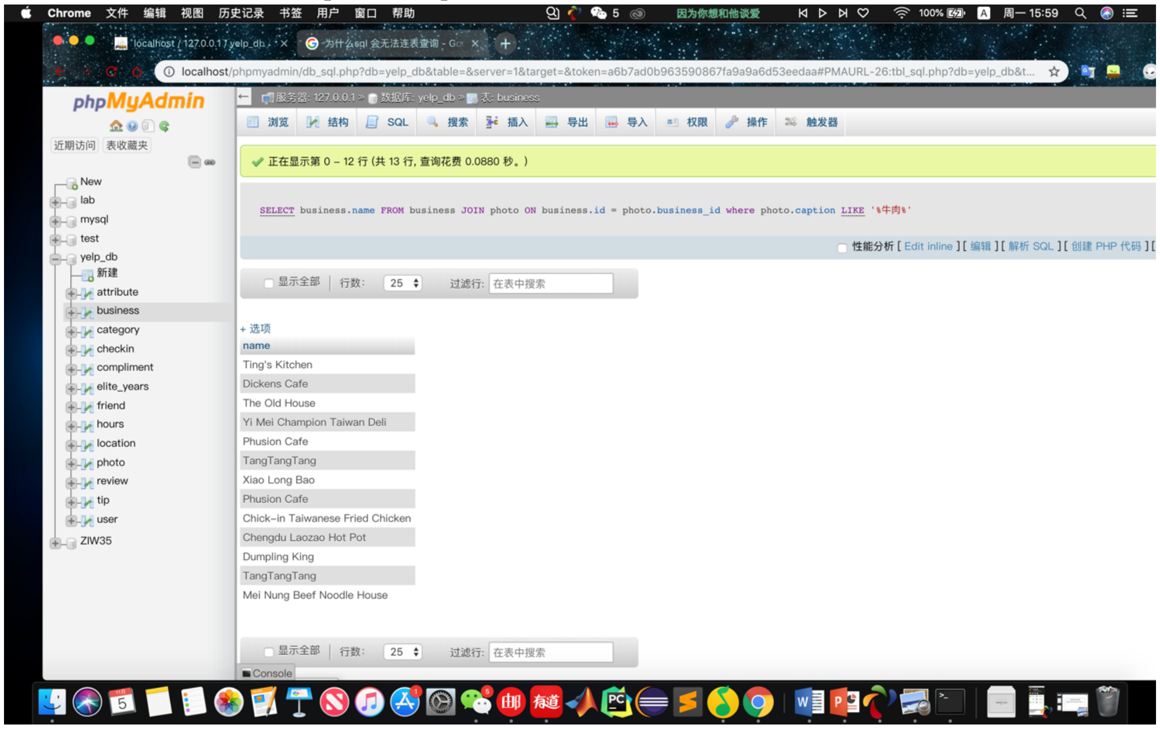
✓ 正在显示第 0 – 99 行 (共 100 行, 查询花费 0.0012 秒。)

```
SELECT business.name from business join checkin on business.id = checkin.business_id where checkin.count > 2 AND business.stars < 4 limit 100
```

| name |
|---|
| Middleburg Pizza & Subs |
| Middleburg Pizza & Subs |
| Applebee's Neighborhood Grill & Bar |
| Applebee's Neighborhood Grill & Bar |
| Applebee's Neighborhood Grill & Bar |
| Applebee's Neighborhood Grill & Bar |
| Mark Pi's China Gate |
| Mark Pi's China Gate |
| Athena's Deli & Restaurant |
| Athena's Deli & Restaurant |
| Athena's Deli & Restaurant |
| Athena's Deli & Restaurant |
| Dairy Queen |
| Dairy Queen |
| Dairy Queen |
| Dairy Queen |
| Dairy Queen |

2. Find the business that caption have the key word “牛肉” :

SELECT business.name FROM business JOIN photo ON business.id = photo.business_id where photo.caption LIKE '%牛肉%';


phpMyAdmin
服务器: 127.0.0.1 » 数据库: yelp_db » 表: business
浏览 结构 SQL 搜索 插入 导出 导入 权限 操作 触发器
New
lab
mysql
test
yelp_db
新建
attribute
business
category
checkin
compliment
elite_years
friend
hours
location
photo
review
tip
user
ZIW35
性能分析 [ Edit inline ] [ 编辑 ] [ 解析 SQL ] [ 创建 PHP 代码 ] [
显示全部 | 行数: 25 过滤行: 在表中搜索
+ 选项
Console

✓ 正在显示第 0 – 12 行 (共 13 行, 查询花费 0.0880 秒。)

```
SELECT business.name FROM business JOIN photo ON business.id = photo.business_id where photo.caption LIKE '%牛肉%'
```

| name |
|---|
| Ting's Kitchen |
| Dickens Cafe |
| The Old House |
| Yi Mei Champion Taiwan Deli |
| Phusion Cafe |
| TangTangTang |
| Xiao Long Bao |
| Phusion Cafe |
| Chick-in Taiwanese Fried Chicken |
| Chengdu Laozao Hot Pot |
| Dumpling King |
| TangTangTang |
| Mei Nung Beef Noodle House |