MYSQL 查询优化实验报告

陈一鸣解子傲

June 30, 2015

Contents

1 概述

1.1 主要目的

通过对不同情况下查询语句的执行分析，巩固和加深对查询和查询优化相关理论知识的理解，提高优化数据库系统的实践能力。

1.2 优化手段

主要用到的优化手段与工具如下，参考资料也一并列出

1.2.1 Explain

- 实验 6：数据库查询优化与范式
- MySQL 优化—工欲善其事，必先利其器之 EXPLAIN

1.2.2 建立索引

- 课本内容

1.2.3 优化语句

- 课本内容

1.2.4 慢查询日志

- MySQL 优化—工欲善其事，必先利其器（2）慢查询部分

1.2.5 Percona Toolkit

- MySQL 优化—工欲善其事，必先利其器（2）Percona Toolkit 部分
- PERCONA TOOLKIT DOCUMENTATION

1.2.6 Docker

- Docker
- Docker MySQL repository
- http://dockerpool.com/static/books/docker_practice/appendix_repo/mysql.html
- http://blog.flux7.com/blogs/docker/docker-tutorial-series-part-1-an-introduction

1.3 优化思路

由于本次实验的数据量较小，各种查询（嵌套查询除外）几乎都能在 0.1 秒以下的时间运行完毕，而这个时间很大程度还受到其他因素的影响（如：机器当前负载等）。而这么短的时间也无法进行慢查询日志的记录，Percona Toolkit 可以做到精确到纳秒，但仍要基于慢日志查询。故本次优化不以执行 SQL 语句的时间为标准，而是以 Explain 语句结果中的 rows 指标为标准，rows 越小，认为效率越高。主要的优化手段也只用到了 **Explain**, **Index** 和 **语句的优化**。

1.3.1 优化过程

1. Explain 原查询
2. 结合 Explain 结果进行优化
3. Explain 优化后的查询

2 实验环境

本次实验为了统一两人的环境，将实验环境搭建在了实验室内网的一台 PC 上，不仅如此，还将 MySQL 搭建在了 Docker 虚拟机中。每次实验前用 Python 脚本将数据库重置一遍，严格控制变量。（另：此处能用 DockerFile 进行初始化，但因初始化的同时要进行数据的初始化，故选择 Python 脚本）

以下是实验环境的具体信息

2.1 PC 信息

- System Information
  - Manufacturer: Dell Inc.
  - Product Name: Inspiron 620
- OS

  Ubuntu 14.04.2 LTS
- CPU

  Intel(R) Core(TM) i5-2320 CPU @ 3.00GHz
- Memory

  4 GB 1333 MHz
- Hard Drive

  1 TB 7200 rpm

2.2 Docker

2.2.1 Docker 简介

Docker 是一个开源的应用容器引擎，让开发者可以打包他们的应用以及依赖包到一个可移植的容器中，然后发布到任何流行的 Linux 机器上，也可以实现虚拟化。容器是完全使用沙箱机制，相互之间不会有任何接口（类似 iPhone 的 app）。几乎没有性能开销, 可以很容易地在机器和数据中心中运行。最重要的是，他们不依赖于任何语言、框架包括系统。[1]

2.2.2 Docker Mysql

以下是用 Docker 建立本次实验环境的语句（仅供参考）：

- Pull

```
docker pull dl.dockerpool.com:5000/mysql:latest
docker tag dl.dockerpool.com:5000/mysql:latest mysql:latest
```

- Run

```
docker run --name test-mysql -e MYSQL_ROOT_PASSWORD=123123 -p 3306:3306 -d mysql
```

- Shell

```
docker exec -it test-mysql bash
```

2.2.3 Docker info

本次 Docker 的运行环境

```
docker info
```

```
Storage Driver: aufs
 Root Dir: /var/lib/docker/aufs
 Backing Filesystem: extfs
 Dirs: 29
 Dirperm1 Supported: true
Execution Driver: native-0.2
Kernel Version: 3.16.0-38-generic
Operating System: Ubuntu 14.04.2 LTS
CPUs: 4
Total Memory: 3.844 GiB
```

[1] http://baike.baidu.com/view/11854949.htm

2.2.4 使用 Docker 的好处

1. 多个 MySQL 实例并行
   可以同时在一台机上运行多个 Docker 实例（即多个 MySQL 实例），两个人可同时进行实验，且互不影响。
2. 方便重置
   只用执行几段代码即可重新搭建出一个新的实验环境，结合 Python 脚本做到任意操作数据库都能轻松回到初始状态。
3. 与本机隔离
   由于跑在虚拟机中，故不必担心影响到服务器本身的环境。

2.3 Python 脚本

详见本项目 Github 主页[2]

2.3.1 reset.py

调用 `init.sql` 中的 sql 语句将数据库重置。

2.3.2 update.py

利用数据库中已有的表信息，从 csv 文件中提取出相应的数据载入到数据库中。（因此只需要完成表的设计即能自动录入，即便表结构变化也不会受到影响）

3 实验内容

3.1 垂直分割

3.1.1 数据含义分析

结合 csv 文件中的数据名和数据值得含义如下：

- 标签分析

  shop_id 编号
  name 店名
  alias 又名
  province 省份
  city 城市
  city_pinyin 城市拼音
  city_id 城市编号
  area 城区
  big_cate 大类
  big_cate_id 大类编号
  small_cate 小类
  small_cate_id 小类编号
  address 地址
  business_area 商区
  phone 电话
  hours 开业时间
  avg_price 平均价格
  stars 星级
  photos 照片
  description 描述
  tags 标签
  map_type 地图类型
  original_latitude 原始纬度
  original_longitude 原始经度
  google_longitude 谷歌经度
  google_latitude 谷歌纬度
  navigation 导航
  traffic 交通
  atmosphere 气氛
  characteristics 特色
  payment 付款
  product_rating 产品评价
  environment_rating 环境评价
  service_rating 服务评价
  all_remarks 所有评价数

[2] https://github.com/dsdshcym/Database-Query-Optimization

very_good_remarks「非常好」评价数
good_remarks「好」评价数
common_remarks「一般」评价数
bad_remarks「差」评价数
very_bad_remarks「非常差」评价数
recommended_dishes 推荐菜品
nearby_shops 附近商铺
is_chains 是连锁店
group 团购
card 优惠券

3.1.2 分割

分析后分为 14 个表，如下（主键用下划线标出）

- 基本信息 (basic)

  shop_id 编号
  name 店名
  alias 又名
  address 地址
  phone 电话
  hours 开业时间
  avg_price 平均价格
  payment 付款
  is_chains 是连锁店

- 城市相关
  - 所在地 (shop_id_area)

    shop_id 编号
    area 城区
    business_area 商区
  - 店-城市 (shop_id_city_id)

    shop_id 编号
    city_id 城市编号
  - 城市 (city_id_city)

    city_id 城市编号
    city 城市
  - 城市、城市拼音 (city_id_city_pinyin)

    city_id 城市编号
    city_pinyin 城市拼音
  - 城市、省份 (city_id_province)

    city_id 城市编号
    province 省份

- 分类
  - 编号-小类 (shop_id_small_cate_id)

    shop_id 编号
    small_cate 小类
  - 小类 (small_cate_id_small_cate)

    small_cate_id 小类编号
    small_cate 小类
  - 编号-大类 (shop_id_big_cate_id)

    shop_id 编号
    small_cate 大类
  - 大类 (big_cate_id_big_cate)

    big_cate_id 大类编号
    big_cate 大类

- 地图类型 (map_info)

  shop_id 编号
  map_type 地图类型
  original_latitude 原始纬度
  original_longitude 原始经度
  google_longitude 谷歌经度
  google_latitude 谷歌纬度
  traffic 交通

- 大众相关 (dazhong)

  shop_id 编号
  navigation 网站导航
  recommended_dishes 推荐菜品
  characteristics 特色
  stars 星级
  photos 照片
  description 描述
  tags 标签
  atmosphere 气氛
  nearby_shops 附近商铺

- 评价 (remark)

  shop_id 编号
  product_rating 产品评价
  environment_rating 环境评价
  service_rating 服务评价
  all_remarks 所有评价数
  very_good_remarks 「非常好」评价数
  good_remarks 「好」评价数
  common_remarks 「一般」评价数
  bad_remarks 「差」评价数
  very_bad_remarks 「非常差」评价数

- 优惠 (discount)

  shop_id 编号
  group 团购
  card 优惠券

3.1.3 建表语句

见本项目 Github 页面[3]
其中所有 Char, Float 变量均为定长，保证效率[4]

3.2 查询优化

3.2.1 单表操作

- 查询表中的所有字段

1. 查询 basic 表

(a) 原查询
实际应用中会出现查询所有店铺信息的情况，比如查询 basic 表中的所有字段的结果

```
explain select *
from basic;
```

```
+----+-------------+-------+------+---------------+------+---------+------+------+-------+
| id | select_type | table | type | possible_keys | key  | key_len | ref  | rows | Extra |
+----+-------------+-------+------+---------------+------+---------+------+------+-------+
| 1  | SIMPLE      | basic | ALL  | NULL          | NULL | NULL    | NULL | 963  | NULL  |
+----+-------------+-------+------+---------------+------+---------+------+------+-------+
1 rows in set (0.05 sec)
```

可见 rows 为 963

(b) 无法优化
查询所有字段需遍历表中所有元组，无法再进一步优化了

2. 查询所有小类

(a) 原查询

```
explain select *
from small_cate_id_small_cate;
```

```
+----+-------------+--------------------------+------+---------------+------+---------+------+------+-------+
| id | select_type | table                    | type | possible_keys | key  | key_len | ref  | rows | Extra |
+----+-------------+--------------------------+------+---------------+------+---------+------+------+-------+
|  1 | SIMPLE      | small_cate_id_small_cate | ALL  | NULL          | NULL | NULL    | NULL |   37 | NULL  |
+----+-------------+--------------------------+------+---------------+------+---------+------+------+-------+
1 row in set (0.03 sec)
```

(b) 无法优化
查询所有字段需遍历表中所有元组，无法再进一步优化了

- 查询表中的指定字段

1. 查询所有店铺的名称

(a) 原查询

```
explain select name
from basic;
```

[3] https://github.com/dsdshcym/Database-Query-Optimization/blob/master/init.sql
[4] http://coolshell.cn/articles/1846.html 第 15 条

```
+----+-------------+-------+------+---------------+------+---------+------+------+-------+
| id | select_type | table | type | possible_keys | key  | key_len | ref  | rows | Extra |
+----+-------------+-------+------+---------------+------+---------+------+------+-------+
| 1  | SIMPLE      | basic | ALL  | NULL          | NULL | NULL    | NULL | 963  | NULL  |
+----+-------------+-------+------+---------------+------+---------+------+------+-------+
1 rows in set (0.05 sec)
```

可见 rows 为 963

(b) 无法优化
该查询需遍历表中所有元组，无法再进一步优化了

2. 查询优惠表中的所有团购信息

(a) 原查询

```
explain select group_info
from discount;
```

```
+----+-------------+----------+------+---------------+------+---------+------+------+-------+
| id | select_type | table    | type | possible_keys | key  | key_len | ref  | rows | Extra |
+----+-------------+----------+------+---------------+------+---------+------+------+-------+
|  1 | SIMPLE      | discount | ALL  | NULL          | NULL | NULL    | NULL | 1000 | NULL  |
+----+-------------+----------+------+---------------+------+---------+------+------+-------+
1 row in set (0.00 sec)
```

(b) 无法优化
该查询需遍历表中所有元组，无法再进一步优化了

- 查询表中没有重复的字段（distinct）的使用

1. 查询不重名的所有店铺名称

(a) 原查询

```
explain select distinct name
from basic;
```

```
+----+-------------+-------+------+---------------+------+---------+------+------+-----------------+
| id | select_type | table | type | possible_keys | key  | key_len | ref  | rows | Extra           |
+----+-------------+-------+------+---------------+------+---------+------+------+-----------------+
| 1  | SIMPLE      | basic | ALL  | NULL          | NULL | NULL    | NULL | 963  | Using temporary |
+----+-------------+-------+------+---------------+------+---------+------+------+-----------------+
1 rows in set (0.05 sec)
```

可见 rows 为 963

(b) 无法优化
该查询需遍历表中所有元组，无法再进一步优化了

2. 查询所有地图类型

(a) 原查询

```
explain select distinct map_type
from map_info;
```

```
+----+-------------+----------+------+---------------+------+---------+------+------+-----------------+
| id | select_type | table    | type | possible_keys | key  | key_len | ref  | rows | Extra           |
+----+-------------+----------+------+---------------+------+---------+------+------+-----------------+
|  1 | SIMPLE      | map_info | ALL  | NULL          | NULL | NULL    | NULL | 1000 | Using temporary |
+----+-------------+----------+------+---------------+------+---------+------+------+-----------------+
1 row in set (0.00 sec)
```

(b) 添加索引

```
create index index_map_type on map_info(map_type);
```

再次查询

```
+----+-------------+----------+-------+---------------+----------------+---------+------+------+--------------------------+
| id | select_type | table    | type  | possible_keys | key            | key_len | ref  | rows | Extra                    |
+----+-------------+----------+-------+---------------+----------------+---------+------+------+--------------------------+
|  1 | SIMPLE      | map_info | range | index_map_type | index_map_type | 1       | NULL |    5 | Using index for group-by |
+----+-------------+----------+-------+---------------+----------------+---------+------+------+--------------------------+
1 row in set (0.02 sec)
```

rows 变为 5 ，效率显著提高

- 条件查询各表主键的字段（单值查询或范围查询）

1. 用一个精确的店编号去查找店铺信息

(a) 原查询
比如查找 basic 中 shop_id=10328540 的店铺信息

```
explain select *
from basic
where shop_id=10328540;
```

```
+----+-------------+-------+-------+---------------+---------+---------+-------+------+-------+
| id | select_type | table | type  | possible_keys | key     | key_len | ref   | rows | Extra |
+----+-------------+-------+-------+---------------+---------+---------+-------+------+-------+
| 1  | SIMPLE      | basic | const | PRIMARY       | PRIMARY | 4       | const | 1    | NULL  |
+----+-------------+-------+-------+---------------+---------+---------+-------+------+-------+
1 rows in set (0.05 sec)
```

可见 rows 已经为 1

(b) 无需优化
因为表创建时已经自带以主键为关键值的索引，无需优化

2. 用店编号范围去查找一部分店铺信息

(a) 原查询
比如查找 basic 中 shop_id>10328540 and shop_id<10329940 的店铺信息

```
explain select *
from basic
where shop_id>10328540 and shop_id<10329940;
```

```
+----+-------------+-------+-------+---------------+---------+---------+------+------+-------------+
| id | select_type | table | type  | possible_keys | key     | key_len | ref  | rows | Extra       |
+----+-------------+-------+-------+---------------+---------+---------+------+------+-------------+
| 1  | SIMPLE      | basic | range | PRIMARY       | PRIMARY | 4       | NULL | 36   | Using where |
+----+-------------+-------+-------+---------------+---------+---------+------+------+-------------+
1 rows in set (0.01 sec)
```

可见 rows 已经为 36

(b) 无需优化
因为表创建时已经自带以主键为关键值的索引，无需优化

- 条件查询各表中普通字段（单值查询或范围查询）

1. 用店名来查找店铺信息

(a) 原查询比如查找 base 中 name 为「林师傅」的结果

```
explain select *
from basic
where name='林师傅';
```

```
+----+-------------+-------+------+---------------+------+---------+------+------+-------------+
| id | select_type | table | type | possible_keys | key  | key_len | ref  | rows | Extra       |
+----+-------------+-------+------+---------------+------+---------+------+------+-------------+
| 1  | SIMPLE      | basic | ALL  | NULL          | NULL | NULL    | NULL | 963  | Using where |
+----+-------------+-------+------+---------------+------+---------+------+------+-------------+
1 rows in set (0.03 sec)
```

可见 rows 为 963

(b) 对 name 进行索引

```
create index index_name on basic(name);
```

再次查找

```
+----+-------------+-------+------+---------------+------------+---------+-------+------+-----------------------+
| id | select_type | table | type | possible_keys | key        | key_len | ref   | rows | Extra                 |
+----+-------------+-------+------+---------------+------------+---------+-------+------+-----------------------+
| 1  | SIMPLE      | basic | ref  | index_name    | index_name | 150     | const | 1    | Using index condition |
+----+-------------+-------+------+---------------+------------+---------+-------+------+-----------------------+
1 rows in set (0.04 sec)
```

可见 rows 已经变为 1，是优化前的 0.1%

2. 查找人均消费在某一范围内的的店铺名称

(a) 原查询
比如查找 base 中 avg_price<50 的结果

```
explain select name
from basic
where avg_price<50;
```

```
+----+-------------+-------+------+---------------+------+---------+------+------+-------------+
| id | select_type | table | type | possible_keys | key  | key_len | ref  | rows | Extra       |
+----+-------------+-------+------+---------------+------+---------+------+------+-------------+
| 1  | SIMPLE      | basic | ALL  | NULL          | NULL | NULL    | NULL | 963  | Using where |
+----+-------------+-------+------+---------------+------+---------+------+------+-------------+
1 rows in set (0.03 sec)
```

可见 rows 为 963

(b) 对 avg_price,name 进行索引

```
create index index_name_price on basic(avg_price,name);
```

再次查找

```
+----+-------------+-------+-------+-----------------+-----------------+---------+------+------+--------------------------+
| id | select_type | table | type  | possible_keys   | key             | key_len | ref  | rows | Extra                    |
+----+-------------+-------+-------+-----------------+-----------------+---------+------+------+--------------------------+
| 1  | SIMPLE      | basic | range | index_name_price | index_name_price | 2       | NULL | 617  | Using where; Using index |
+----+-------------+-------+-------+-----------------+-----------------+---------+------+------+--------------------------+
1 rows in set (0.05 sec)
```

可见 rows 已经变为 617, 是优化前的 64%

- 一个表中多个字段条件查询（单值查询或范围查询）

1. 多条件查询 remark 表中的 shop_id

(a) 原查询

```
explain select shop_id
from remark
where product_rating > 8.5 and environment_rating > 8.5;
```

```
+----+-------------+--------+------+---------------+------+---------+------+------+-------------+
| id | select_type | table  | type | possible_keys | key  | key_len | ref  | rows | Extra       |
+----+-------------+--------+------+---------------+------+---------+------+------+-------------+
|  1 | SIMPLE      | remark | ALL  | NULL          | NULL | NULL    | NULL | 1000 | Using where |
+----+-------------+--------+------+---------------+------+---------+------+------+-------------+
1 row in set (0.00 sec)
```

可见 rows 为 1000

(b) 添加索引

```
create index index_product_rating on remark(product_rating);
create index index_environment_rating on remark(environment_rating);
```

(c) 优化后

```
explain select shop_id
from remark
where product_rating > 8.5 and environment_rating > 8.5;
```

```
+----+-------------+--------+-------+-----------------------------------------+-------------------------+---------+------+------+-----------------------------------------+
| id | select_type | table  | type  | possible_keys                           | key                     | key_len | ref  | rows | Extra                                   |
+----+-------------+--------+-------+-----------------------------------------+-------------------------+---------+------+------+-----------------------------------------+
|  1 | SIMPLE      | remark | range | index_product_rating,index_environment_rating | index_product_rating | 5       | NULL |   52 | Using index condition; Using where |
+----+-------------+--------+-------+-----------------------------------------+-------------------------+---------+------+------+-----------------------------------------+
1 row in set (0.00 sec)
```

优化后 rows 变为 52，效率显著提高

2. 通过 name 和 alias 查询 phone

(a) 原查询

```
explain select phone from basic
where name = '巫山烤全鱼' and alias = '重庆鸡公煲';
```

```
+----+-------------+-------+------+---------------+------+---------+------+------+-------------+
| id | select_type | table | type | possible_keys | key  | key_len | ref  | rows | Extra       |
+----+-------------+-------+------+---------------+------+---------+------+------+-------------+
|  1 | SIMPLE      | basic | ALL  | NULL          | NULL | NULL    | NULL |  963 | Using where |
+----+-------------+-------+------+---------------+------+---------+------+------+-------------+
1 row in set (0.00 sec)
```

可见 rows 为 963，有优化空间

(b) 添加索引

```
create index index_name on basic(name);
```

```
+----+-------------+-------+------+---------------+------------+---------+-------+------+-----------------------------------+
| id | select_type | table | type | possible_keys | key        | key_len | ref   | rows | Extra                             |
+----+-------------+-------+------+---------------+------------+---------+-------+------+-----------------------------------+
|  1 | SIMPLE      | basic | ref  | index_name    | index_name | 150     | const |    3 | Using index condition; Using where |
+----+-------------+-------+------+---------------+------------+---------+-------+------+-----------------------------------+
1 row in set (0.00 sec)
```

rows 降为 3

```
create index index_alias on basic(alias);
```

```
+----+-------------+-------+------+-----------------------+-------------+---------+------+------+-----------------------------------+
| id | select_type | table | type | possible_keys         | key         | key_len | ref  | rows | Extra                             |
+----+-------------+-------+------+-----------------------+-------------+---------+------+------+-----------------------------------+
|  1 | SIMPLE      | basic | ref  | index_name,index_alias | index_alias | 120     | const |    1 | Using index condition; Using where |
+----+-------------+-------+------+-----------------------+-------------+---------+------+------+-----------------------------------+
1 row in set (0.07 sec)
```

rows 降为 1 ，效率显著提高

(c) 更改查询语句顺序 + 索引

由于 alias 上大多为空值，重复率较低，可先查询 alias 再查询 name

```
create index index_alias on basic(alias);

explain select phone from basic
where alias = '重庆鸡公煲' and name = '巫山烤全鱼';
```

```
+----+-------------+-------+------+--------------+-------------+---------+------+------+-----------------------------------+
| id | select_type | table | type | possible_keys | key         | key_len | ref  | rows | Extra                             |
+----+-------------+-------+------+--------------+-------------+---------+------+------+-----------------------------------+
|  1 | SIMPLE      | basic | ref  | index_alias   | index_alias | 120     | const |    1 | Using index condition; Using where |
+----+-------------+-------+------+--------------+-------------+---------+------+------+-----------------------------------+
1 row in set (0.00 sec)
```

rows 降为 1 ，效率显著提高

3. 用店名和人均消费值来查找店铺信息

(a) 原查询

```
explain select *
from basic
where name='林师傅' and avg_price=12;
```

```
+----+-------------+-------+------+---------------+------+---------+------+------+-------------+
| id | select_type | table | type | possible_keys | key  | key_len | ref  | rows | Extra       |
+----+-------------+-------+------+---------------+------+---------+------+------+-------------+
| 1  | SIMPLE      | basic | ALL  | NULL          | NULL | NULL    | NULL | 963  | Using where |
+----+-------------+-------+------+---------------+------+---------+------+------+-------------+
1 rows in set (0.05 sec)
```

可见 rows 为 963

(b) 添加索引

```
create index index_name on basic(avg_price,name);
```

(c) 优化后

```
+----+-------------+-------+------+---------------+------------+---------+-------------+------+-----------------------+
| id | select_type | table | type | possible_keys | key        | key_len | ref         | rows | Extra                 |
+----+-------------+-------+------+---------------+------------+---------+-------------+------+-----------------------+
| 1  | SIMPLE      | basic | ref  | index_name    | index_name | 152     | const,const | 1    | Using index condition |
+----+-------------+-------+------+---------------+------------+---------+-------------+------+-----------------------+
1 rows in set (0.04 sec)
```

可见 rows 已经变为 1 , 是优化前的 0.1%

4. 查找特定店名但人均消费在一定范围的店铺名称

(a) 原查询

```
explain select name
from basic
where name='林师傅' and avg_price<50;
```

```
+----+-------------+-------+------+---------------+------+---------+------+------+-------------+
| id | select_type | table | type | possible_keys | key  | key_len | ref  | rows | Extra       |
+----+-------------+-------+------+---------------+------+---------+------+------+-------------+
| 1  | SIMPLE      | basic | ALL  | NULL          | NULL | NULL    | NULL | 1000 | Using where |
+----+-------------+-------+------+---------------+------+---------+------+------+-------------+
1 rows in set (0.04 sec)
```

可见 rows 为 1000

(b) 添加索引

```
create index index_name_price on basic(avg_price,name);
```

(c) 优化后

```
+----+-------------+-------+-------+-----------------+-----------------+---------+------+------+-------------------------+
| id | select_type | table | type  | possible_keys   | key             | key_len | ref  | rows | Extra                   |
+----+-------------+-------+-------+-----------------+-----------------+---------+------+------+-------------------------+
| 1  | SIMPLE      | basic | range | index_name_price | index_name_price | 2       | NULL | 617  | Using where; Using index |
+----+-------------+-------+-------+-----------------+-----------------+---------+------+------+-------------------------+
```

可见 rows 已经变为 617 ，是优化前的 61.7%

- 用”in”进行条件查询

1. 用几个 small_cate_id 来查询 shop_id

(a) 原查询

```
explain select shop_id
from shop_id_small_cate_id
where small_cate_id in ('g101', 'g103', 'g105', 'g107');
```

```
+----+-------------+----------------------+-------+---------------+---------------+---------+------+------+--------------------------+
| id | select_type | table                | type  | possible_keys | key           | key_len | ref  | rows | Extra                    |
+----+-------------+----------------------+-------+---------------+---------------+---------+------+------+--------------------------+
|  1 | SIMPLE      | shop_id_small_cate_id | range | small_cate_id | small_cate_id | 18      | NULL |   76 | Using where; Using index |
+----+-------------+----------------------+-------+---------------+---------------+---------+------+------+--------------------------+
1 row in set (0.01 sec)
```

可见 rows 为 76

(b) 无需优化

因为 small_cate_id 已经是 small_cate_id_small_cate 表的主键，已有索引，故无需优化。

2. 查找符合某几个店名的店铺

(a) 原查询

```
explain select *
from basic
where name in ('真好味','阿姨布丁','青春学堂');
```

```
+----+-------------+-------+------+---------------+------+---------+------+------+-------------+
| id | select_type | table | type | possible_keys | key  | key_len | ref  | rows | Extra       |
+----+-------------+-------+------+---------------+------+---------+------+------+-------------+
| 1  | SIMPLE      | basic | ALL  | NULL          | NULL | NULL    | NULL | 963  | Using where |
+----+-------------+-------+------+---------------+------+---------+------+------+-------------+
1 rows in set (0.05 sec)
```

可见 rows 为 963

(b) 添加索引

```
create index index_name on basic(name);
```

(c) 优化后

```
+----+-------------+-------+-------+---------------+------------+---------+------+------+-----------------------+
| id | select_type | table | type  | possible_keys | key        | key_len | ref  | rows | Extra                 |
+----+-------------+-------+-------+---------------+------------+---------+------+------+-----------------------+
| 1  | SIMPLE      | basic | range | index_name    | index_name | 150     | NULL | 3    | Using index condition |
+----+-------------+-------+-------+---------------+------------+---------+------+------+-----------------------+
1 rows in set (0.04 sec)
```

可见 rows 已经变为 3 , 是优化前的 0.3%

- 一个表中 group by、order by、having 联合条件查询

1. 按平均价格来查询

(a) 原查询

```
explain select * from basic
where avg_price < 20
order by avg_price;
```

```
+----+-------------+-------+------+---------------+------+---------+------+------+----------------------------+
| id | select_type | table | type | possible_keys | key  | key_len | ref  | rows | Extra                      |
+----+-------------+-------+------+---------------+------+---------+------+------+----------------------------+
|  1 | SIMPLE      | basic | ALL  | NULL          | NULL | NULL    | NULL |  963 | Using where; Using filesort |
+----+-------------+-------+------+---------------+------+---------+------+------+----------------------------+
1 row in set (0.00 sec)
```

可见 rows 为 963

(b) 利用索引进行优化

```
create index index_price on basic(avg_price);
```

(c) 优化后

```
explain select * from basic
where avg_price < 20
order by avg_price;
```

```
+----+-------------+-------+-------+---------------+-------------+---------+------+------+-----------------------+
| id | select_type | table | type  | possible_keys | key         | key_len | ref  | rows | Extra                 |
+----+-------------+-------+-------+---------------+-------------+---------+------+------+-----------------------+
|  1 | SIMPLE      | basic | range | index_price   | index_price | 2       | NULL |  235 | Using index condition |
+----+-------------+-------+-------+---------------+-------------+---------+------+------+-----------------------+
1 row in set (0.04 sec)
```

2. remark 表分组排序筛选查找

(a) 原查询

将 remark 按环境评分 environment_rating 分组, 筛选出食品评分 product_rating > 7 的餐馆, 并且按平均评价数量 all_remarks 排序

```
explain select shop_id,environment_rating,product_rating,all_remarks
from remark
group by environment_rating
having product_rating>7
order by all_remarks;
```

```
+----+-------------+--------+------+---------------+------+---------+------+------+--------------------------------+
| id | select_type | table  | type | possible_keys | key  | key_len | ref  | rows | Extra                          |
+----+-------------+--------+------+---------------+------+---------+------+------+--------------------------------+
| 1  | SIMPLE      | remark | ALL  | NULL          | NULL | NULL    | NULL | 1000 | Using temporary; Using filesort |
+----+-------------+--------+------+---------------+------+---------+------+------+--------------------------------+
1 rows in set (0.03 sec)
```

可见 rows 为 1000

(b) 添加索引

```
create index index_environment_rating,product_rating,all_remarks on remark(environment_rating,product_rating,all_remarks);
```

(c) 优化后

```
+----+-------------+--------+-------+---------------+---------+---------+------+------+-----------------------------------------------------------------+
| id | select_type | table  | type  | possible_keys | key     | key_len | ref  | rows | Extra                                                           |
+----+-------------+--------+-------+---------------+---------+---------+------+------+-----------------------------------------------------------------+
| 1  | SIMPLE      | remark | range | index_1       | index_1 | 5       | NULL | 84   | Using index for group-by; Using temporary; Using filesort |
+----+-------------+--------+-------+---------------+---------+---------+------+------+-----------------------------------------------------------------+
1 rows in set (0.06 sec)
```

可见 rows 已经变为 84 , 是优化前的 0.84%

3.2.2 复合查询

- 多表联合查询

1. 查询某 city 的所有 shop_id

(a) 原查询

```
explain select shop_id
from shop_id_city_id as SC, city_id_city as C
where SC.city_id = C.city_id and C.city = '上海';
```

```
+----+-------------+-------+------+---------------+---------+---------+----------------+------+-------------+
| id | select_type | table | type | possible_keys | key     | key_len | ref            | rows | Extra       |
+----+-------------+-------+------+---------------+---------+---------+----------------+------+-------------+
|  1 | SIMPLE      | C     | ALL  | PRIMARY       | NULL    | NULL    | NULL           |   59 | Using where |
|  1 | SIMPLE      | SC    | ref  | city_id       | city_id | 2       | test.C.city_id |    1 | Using index |
+----+-------------+-------+------+---------------+---------+---------+----------------+------+-------------+
2 rows in set (0.01 sec)
```

可见 rows 为 59 * 1 ，因 city_id 为主键，故在 SC 表上无需优化

(b) 添加索引

```
create index index_city on city_id_city(city);
```

(c) 优化后

```
explain select shop_id
from shop_id_city_id as SC, city_id_city as C
where SC.city_id = C.city_id and C.city = '上海';
```

```
+----+-------------+-------+------+-------------------+------------+---------+----------------+------+--------------------------+
| id | select_type | table | type | possible_keys     | key        | key_len | ref            | rows | Extra                    |
+----+-------------+-------+------+-------------------+------------+---------+----------------+------+--------------------------+
|  1 | SIMPLE      | C     | ref  | PRIMARY,index_city | index_city | 45      | const          |    1 | Using where; Using index |
|  1 | SIMPLE      | SC    | ref  | city_id           | city_id    | 2       | test.C.city_id |    1 | Using index              |
+----+-------------+-------+------+-------------------+------------+---------+----------------+------+--------------------------+
2 rows in set (0.01 sec)
```

可见 rows 变为 1 * 1 ，效率显著提高

2. 查询某小类下的所有商铺名

(a) 原查询

```
explain select B.shop_id, B.name
from basic as B, shop_id_small_cate_id as SS
where B.shop_id = SS.shop_id and SS.small_cate_id = 'g101';
```

```
+----+-------------+-------+-------+----------------------+--------------+---------+-----------------+------+--------------------------+
| id | select_type | table | type  | possible_keys        | key          | key_len | ref             | rows | Extra                    |
+----+-------------+-------+-------+----------------------+--------------+---------+-----------------+------+--------------------------+
|  1 | SIMPLE      | SS    | ref   | PRIMARY,small_cate_id | small_cate_id | 18      | const           |   32 | Using where; Using index |
|  1 | SIMPLE      | B     | eq_ref | PRIMARY              | PRIMARY      | 4       | test.SS.shop_id |    1 | NULL                     |
+----+-------------+-------+-------+----------------------+--------------+---------+-----------------+------+--------------------------+
2 rows in set (0.01 sec)
```

rows 为 32 * 1

(b) 无需优化

因为 small_cate_id 和 shop_id 都是主键，已有索引，故无需优化

3. 寻找平均价格低于 20 元且食物评分高于 8 的店铺

(a) 原查询

```
explain select basic.name,basic.avg_price,remark.product_rating
from basic,remark
where basic.shop_id=remark.shop_id and basic.avg_price<20 and remark.product_rating>8;
```

```
+----+-------------+--------+--------+---------------+---------+---------+-------------------+------+-------------+
| id | select_type | table  | type   | possible_keys | key     | key_len | ref               | rows | Extra       |
+----+-------------+--------+--------+---------------+---------+---------+-------------------+------+-------------+
| 1  | SIMPLE      | basic  | ALL    | PRIMARY       | NULL    | NULL    | NULL              | 1000 | Using where |
| 1  | SIMPLE      | remark | eq_ref | PRIMARY       | PRIMARY | 4       | test.basic.shop_id | 1    | Using where |
+----+-------------+--------+--------+---------------+---------+---------+-------------------+------+-------------+
2 rows in set (0.06 sec)
```

可见 rows 分别为 1000, 1

(b) 添加索引

```
create index index_avg_price on basic(avg_price);
```

(c) 优化后

```
+----+-------------+-------+--------+-----------------------+-----------------+---------+-----------------+------+-----------------------+
| id | select_type | table | type   | possible_keys         | key             | key_len | ref             | rows | Extra                 |
+----+-------------+-------+--------+-----------------------+-----------------+---------+-----------------+------+-----------------------+
| 1  | SIMPLE      | basic | range  | PRIMARY,index_avg_price | index_avg_price | 2       | NULL            | 235  | Using index condition |
| 1  | SIMPLE      | remark | eq_ref | PRIMARY              | PRIMARY         | 4       | test.basic.shop_id | 1    | Using where           |
+----+-------------+-------+--------+-----------------------+-----------------+---------+-----------------+------+-----------------------+
2 rows in set (0.05 sec)
```

可见 rows 已经变为 235, 1 ，是优化前的 23.5%

- join 查询

1. 利用拼音查询城市名

(a) 原查询

```
explain select city
from city_id_city as C
inner join city_id_city_pinyin as CP
where C.city_id = CP.city_id and CP.city_pinyin = 'shanghai';
```

```
+----+-------------+-------+--------+---------------+---------+---------+---------------+------+-------------+
| id | select_type | table | type   | possible_keys | key     | key_len | ref           | rows | Extra       |
+----+-------------+-------+--------+---------------+---------+---------+---------------+------+-------------+
|  1 | SIMPLE      | C     | ALL    | PRIMARY       | NULL    | NULL    | NULL          |   59 | NULL        |
|  1 | SIMPLE      | CP    | eq_ref | PRIMARY       | PRIMARY | 2       | test.C.city_id |    1 | Using where |
+----+-------------+-------+--------+---------------+---------+---------+---------------+------+-------------+
2 rows in set (0.01 sec)
```

可见 rows 为 59 * 1 ，因 city_id 为主键，故在 C 表上无需优化

(b) 添加索引

```
create index index_city_pinyin on city_id_city_pinyin(city_pinyin);
```

(c) 优化后

```
explain select city
from city_id_city as C
inner join city_id_city_pinyin as CP
where C.city_id = CP.city_id and CP.city_pinyin = 'shanghai';
```

```
+----+-------------+-------+--------+-----------------------+------------------+---------+-----------------+------+--------------------------+
| id | select_type | table | type   | possible_keys         | key              | key_len | ref             | rows | Extra                    |
+----+-------------+-------+--------+-----------------------+------------------+---------+-----------------+------+--------------------------+
|  1 | SIMPLE      | CP    | ref    | PRIMARY,index_city_pinyin | index_city_pinyin | 45      | const           |    1 | Using where; Using index |
|  1 | SIMPLE      | C     | eq_ref | PRIMARY               | PRIMARY          | 2       | test.CP.city_id |    1 | NULL                     |
+----+-------------+-------+--------+-----------------------+------------------+---------+-----------------+------+--------------------------+
2 rows in set (0.01 sec)
```

rows 变为 1 * 1 ，效率显著提高

2. 获取所有商铺的图片

(a) 原查询

```
explain select B.shop_id, D.photos
from basic as B
inner join dazhong as D
where B.shop_id = D.shop_id;
```

```
+----+-------------+-------+--------+---------------+---------+---------+-----------------+------+-------------+
| id | select_type | table | type   | possible_keys | key     | key_len | ref             | rows | Extra       |
+----+-------------+-------+--------+---------------+---------+---------+-----------------+------+-------------+
|  1 | SIMPLE      | D     | ALL    | PRIMARY       | NULL    | NULL    | NULL            |  939 | NULL        |
|  1 | SIMPLE      | B     | eq_ref | PRIMARY       | PRIMARY | 4       | test.D.shop_id |    1 | Using index |
+----+-------------+-------+--------+---------------+---------+---------+-----------------+------+-------------+
2 rows in set (0.43 sec)
```

可见 rows 为 939 * 1

(b) 无法优化

因为每个商铺的图片地址均不相同，故读取所有地址需要遍历全表，无法优化

- 存在量词（exists）查询

1. 查询产品评价高于 9 的 shop_id, name

(a) 原查询

```
explain select B.shop_id, B.name
from basic as B
where exists
(select *
from remark as R
where R.shop_id = B.shop_id and R.product_rating > 9);
```

```
+----+--------------------+-------+--------+---------------+---------+---------+-----------------+------+-------------+
| id | select_type        | table | type   | possible_keys | key     | key_len | ref             | rows | Extra       |
+----+--------------------+-------+--------+---------------+---------+---------+-----------------+------+-------------+
|  1 | PRIMARY            | B     | ALL    | NULL          | NULL    | NULL    | NULL            |  963 | Using where |
|  2 | DEPENDENT SUBQUERY | R     | eq_ref | PRIMARY       | PRIMARY | 4       | test.B.shop_id |    1 | Using where |
+----+--------------------+-------+--------+---------------+---------+---------+-----------------+------+-------------+
2 rows in set (0.01 sec)
```

可见 rows 为 963 * 1

(b) 添加索引

因 shop_id 已有索引，故不再添加

```
create index index_product_rating on remark(product_rating);
```

但添加索引后结果不变，考虑是 exists 的问题

(c) 改为 join 查询

```
explain select B.shop_id, B.name
from basic as B, remark as R
where R.shop_id = B.shop_id and R.product_rating > 9;
```

```
+----+-------------+-------+--------+---------------------------+--------------------+---------+---------------+------+--------------------------+
| id | select_type | table | type   | possible_keys             | key                | key_len | ref           | rows | Extra                    |
+----+-------------+-------+--------+---------------------------+--------------------+---------+---------------+------+--------------------------+
|  1 | SIMPLE      | R     | range  | PRIMARY,index_product_rating | index_product_rating | 5       | NULL          |    5 | Using where; Using index |
|  1 | SIMPLE      | B     | eq_ref | PRIMARY                   | PRIMARY            | 4       | test.R.shop_id |    1 | NULL                     |
+----+-------------+-------+--------+---------------------------+--------------------+---------+---------------+------+--------------------------+
2 rows in set (0.05 sec)
```

经过添加索引和改为 join 查询后，rows 变为 5 * 1 ，效率显著提高

- 嵌套子查询（select … from (select …)）

1. 查找 stars 大于 4.0 的 shop_id

(a) 原查询

```
select shop_id
from (select * from dazhong where stars > 4.0) as D;
```

```
+----+-------------+------------+------+---------------+------+---------+------+------+-------------+
| id | select_type | table      | type | possible_keys | key  | key_len | ref  | rows | Extra       |
+----+-------------+------------+------+---------------+------+---------+------+------+-------------+
|  1 | PRIMARY     | <derived2> | ALL  | NULL          | NULL | NULL    | NULL |  958 | NULL        |
|  2 | DERIVED     | dazhong    | ALL  | NULL          | NULL | NULL    | NULL |  958 | Using where |
+----+-------------+------------+------+---------------+------+---------+------+------+-------------+
2 rows in set (0.03 sec)
```

可见 rows 为 958 * 958 ，效率极低

(b) 添加索引

```
create index index_stars on dazhong(stars);
```

添加索引后

```
explain select shop_id
from (select * from dazhong where stars > 4.0) as D;
```

```
+----+-------------+------------+-------+---------------+-------------+---------+------+------+-----------------------+
| id | select_type | table      | type  | possible_keys | key         | key_len | ref  | rows | Extra                 |
+----+-------------+------------+-------+---------------+-------------+---------+------+------+-----------------------+
|  1 | PRIMARY     | <derived2> | ALL   | NULL          | NULL        | NULL    | NULL |   86 | NULL                  |
|  2 | DERIVED     | dazhong    | range | index_stars   | index_stars | 5       | NULL |   86 | Using index condition |
+----+-------------+------------+-------+---------------+-------------+---------+------+------+-----------------------+
2 rows in set (0.02 sec)
```

rows 降为 86 * 86 ，效率在一定程度上提高

(c) 改为非嵌套查询

```
explain select shop_id
from dazhong where stars > 4.0;
```

```
+----+-------------+---------+-------+---------------+-------------+---------+------+------+--------------------------+
| id | select_type | table   | type  | possible_keys | key         | key_len | ref  | rows | Extra                    |
+----+-------------+---------+-------+---------------+-------------+---------+------+------+--------------------------+
|  1 | SIMPLE      | dazhong | range | index_stars   | index_stars | 5       | NULL |   86 | Using where; Using index |
+----+-------------+---------+-------+---------------+-------------+---------+------+------+--------------------------+
1 row in set (0.01 sec)
```

经过添加索引、改为非嵌套查询优化后，rows 降为 86 ，效率显著提高

2. 查找所有位于杨浦区的店铺 id 和名称与平均价格

(a) 原查询

```
explain select shop_id,name,avg_price
from basic
where shop_id in(select shop_id
from shop_id_area
where area='杨浦区');
```

```
+----+-------------+--------------+--------+---------------+---------+---------+--------------------------+------+-------------+
| id | select_type | table        | type   | possible_keys | key     | key_len | ref                      | rows | Extra       |
+----+-------------+--------------+--------+---------------+---------+---------+--------------------------+------+-------------+
| 1  | SIMPLE      | shop_id_area | ALL    | PRIMARY       | NULL    | NULL    | NULL                     | 1000 | Using where |
| 1  | SIMPLE      | basic        | eq_ref | PRIMARY       | PRIMARY | 4       | test.shop_id_area.shop_id | 1    | NULL        |
+----+-------------+--------------+--------+---------------+---------+---------+--------------------------+------+-------------+
2 rows in set (0.04 sec)
```

可见 rows 分别为 1000,1

(b) 添加索引

对 shop_id_area 的 area 进行索引

```
create index index_shop_id_area on shop_id_area(area);
```

(c) 优化后

```
+----+-------------+--------------+--------+---------------------------+--------------------+---------+---------------------------+------+--------------------------+
| id | select_type | table        | type   | possible_keys             | key                | key_len | ref                       | rows | Extra                    |
+----+-------------+--------------+--------+---------------------------+--------------------+---------+---------------------------+------+--------------------------+
| 1  | SIMPLE      | shop_id_area | ref    | PRIMARY,index_shop_id_area | index_shop_id_area | 120     | const                     | 19   | Using where; Using index |
| 1  | SIMPLE      | basic        | eq_ref | PRIMARY                   | PRIMARY            | 4       | test.shop_id_area.shop_id | 1    | NULL                     |
+----+-------------+--------------+--------+---------------------------+--------------------+---------+---------------------------+------+--------------------------+
2 rows in set (0.03 sec)
```

可见 rows 已经变为 19, 1 ，是优化前的 1.9%

3.2.3 其他查询

- 向表中插入记录

1. 直接向 city_id_city 中插入新元组

(a) 原查询

```
explain insert into city_id_city(city_id, city)
values (1001, 'test');
```

```
+----+-------------+-------+------+---------------+------+---------+------+------+----------------+
| id | select_type | table | type | possible_keys | key  | key_len | ref  | rows | Extra          |
+----+-------------+-------+------+---------------+------+---------+------+------+----------------+
|  1 | SIMPLE      | NULL  | NULL | NULL          | NULL | NULL    | NULL | NULL | No tables used |
+----+-------------+-------+------+---------------+------+---------+------+------+----------------+
1 row in set (0.00 sec)
```

直接插入无需优化

- 删除记录

1. 直接将 city_id_city 中的一项删除

(a) 原查询

```
explain delete
from city_id_city
where city = 'test';
```

```
+----+-------------+--------------+------+---------------+------+---------+------+------+-------------+
| id | select_type | table        | type | possible_keys | key  | key_len | ref  | rows | Extra       |
+----+-------------+--------------+------+---------------+------+---------+------+------+-------------+
|  1 | SIMPLE      | city_id_city | ALL  | NULL          | NULL | NULL    | NULL |   60 | Using where |
+----+-------------+--------------+------+---------------+------+---------+------+------+-------------+
1 row in set (0.01 sec)
```

rows 为 60

(b) 添加索引

```
create index index_city on city_id_city(city);
```

添加索引后

```
explain delete
from city_id_city
where city = 'test';
```

```
+----+-------------+--------------+-------+---------------+------------+---------+-------+------+-------------+
| id | select_type | table        | type  | possible_keys | key        | key_len | ref   | rows | Extra       |
+----+-------------+--------------+-------+---------------+------------+---------+-------+------+-------------+
|  1 | SIMPLE      | city_id_city | range | index_city    | index_city | 45      | const |    1 | Using where |
+----+-------------+--------------+-------+---------------+------------+---------+-------+------+-------------+
1 row in set (0.01 sec)
```

rows 变为 1 ，效率显著提高

(c) 使用主键

使用主键来进行删除操作

```
explain delete
from city_id_city
where city_id = 1001 and city = 'test';
```

```
+----+-------------+--------------+-------+---------------+---------+---------+-------+------+-------------+
| id | select_type | table        | type  | possible_keys | key     | key_len | ref   | rows | Extra       |
+----+-------------+--------------+-------+---------------+---------+---------+-------+------+-------------+
|  1 | SIMPLE      | city_id_city | range | PRIMARY       | PRIMARY | 2       | const |    1 | Using where |
+----+-------------+--------------+-------+---------------+---------+---------+-------+------+-------------+
1 row in set (0.03 sec)
```

rows 也为 1 ，效率显著提高

- 聚集函数

1. 统计某城市中有多少商铺

(a) 原查询

```
explain select count(city_id)
from shop_id_city_id
where city_id = 1;
```

```
+----+-------------+-----------------+------+---------------+---------+---------+-------+------+-------------+
| id | select_type | table           | type | possible_keys | key     | key_len | ref   | rows | Extra       |
+----+-------------+-----------------+------+---------------+---------+---------+-------+------+-------------+
|  1 | SIMPLE      | shop_id_city_id | ref  | city_id       | city_id | 2       | const |  257 | Using index |
+----+-------------+-----------------+------+---------------+---------+---------+-------+------+-------------+
1 row in set (0.00 sec)
```

rows 为 257 ，但已经是利用了主键索引了，且 count 函数是需要遍历所有 257 个结果的，故已经无法再优化了

(b) 实际应用

在实际应用 count 语句时，使用 `count(*)` 将比 count(具体键值) 快。因为 `count(*)` 利用的是主键索引[5]。

- 其他查询

1. Like 语句

(a) 查找所有星巴克的分店

i. 原查询

```
explain select shop_id,name
from basic
where name like '星巴克%';
```

```
+----+-------------+-------+------+---------------+------+---------+------+------+-------------+
| id | select_type | table | type | possible_keys | key  | key_len | ref  | rows | Extra       |
+----+-------------+-------+------+---------------+------+---------+------+------+-------------+
| 1  | SIMPLE      | basic | ALL  | NULL          | NULL | NULL    | NULL | 963  | Using where |
+----+-------------+-------+------+---------------+------+---------+------+------+-------------+
1 rows in set (0.03 sec)
```

可见 rows 为 963

ii. 添加索引

```
create index index_name on basic(name);
```

iii. 优化后

```
+----+-------------+-------+-------+---------------+------------+---------+------+------+--------------------------+
| id | select_type | table | type  | possible_keys | key        | key_len | ref  | rows | Extra                    |
+----+-------------+-------+-------+---------------+------------+---------+------+------+--------------------------+
| 1  | SIMPLE      | basic | range | index_name    | index_name | 150     | NULL | 6    | Using where; Using index |
+----+-------------+-------+-------+---------------+------------+---------+------+------+--------------------------+
1 rows in set (0.03 sec)
```

可见 rows 已经变为 6 , 是优化前的 0.6%

iv. 优化查询语句

将 Like 查询转化为范围查询效率更高（「兌」是「克」在 utf-8 编码下的后一位）

```
select shop_id, name
from basic
where name>'星巴克' and name<'星巴兌';
```

[5] http://blog.itpub.net/22664653/viewspace-774679/