

用户登录系统

记录用户登录信息的一个系统，我们简化业务后只留下一张表。

关系型数据库的设计

```
mysql> select * from login;
```

| user_id | name | login_times | last_login_time |
|---------|----------------|-------------|---------------------|
| 1 | ken thompson | 5 | 2011-01-01 00:00:00 |
| 2 | dennis ritchie | 1 | 2011-02-01 00:00:00 |
| 3 | Joe Armstrong | 5 | 2011-03-01 00:00:00 |

user_id 表的主键，name 表示用户名，login_times 表示该用户的登录次数，每次用户登录后，login_times 会自增，而 last_login_time 更新为当前时间。

REDIS 的设计

关系型数据转化为 KV 数据库，方法如下：

Key 名: “表名:主键名:主键的值:列名”

Value: 列值

一般使用冒号做分割符，这是不成文的规矩。

以上的关系数据转化成 kv 数据后记录如下：

```
Set login:user_id:1:login_times 5
```

```
Set login:user_id:2:login_times 1
```

```
Set login:user_id: 3:login_times 5
```

```
Set login:user_id:1:last_login_time 2011-1-1
```

```
Set login:user_id:2:last_login_time 2011-2-1
```

```
Set login:user_id:3:last_login_time 2011-3-1
```

```
set login:user_id:1:name "ken thompson"
```

```
set login:user_id:2:name "dennis ritchie"
```

```
set login:user_id:3:name "Joe Armstrong"
```

这样在已知主键的情况下，通过 `get`、`set` 就可以获得或者修改用户的登录次数和最后登录时间和姓名

例如：想知道 `user_id` 为 1 的登录累计次数多少

```
get login:user_id:1:login_times
```

问：如果想查询上次登录时间为 2011 年 3 月 1 日的用户名是什么该怎么办？

答：进行冗余存储

再存储如下信息

```
sadd "login:name:ken thompson:user_id" 1
```

```
sadd "login:name:dennis ritchie:user_id" 2
```

```
sadd "login:name:Joe Armstrong:user_id" 3
```

```
sadd "login:login_times:5:user_id" 1
```

```
sadd "login:login_times:1:user_id" 2
```

```
sadd "login:login_times:5:user_id" 3
```

```
sadd "login:last_login_time:2011-1-1:user_id" 1
```

```
sadd "login:last_login_time:2011-2-1:user_id" 2
```

```
sadd "login:last_login_time:2011-3-1:user_id" 3
```

问：如果想查询上次登录时间为 2011 年 3 月 1 日的用户名是什么该怎么办？

先输入命令

```
smembers "login:last_login_time:2011-3-1:user_id"
```

得到返回值 3

再根据得到的结果使用

`get login:user_id:3:name`

即可得到结果

问：为什么主键作为 key 名用 set，而非主键作为 key 名用 sadd？

答：主键是唯一的，而非主键可能不唯一，例如 last_login_time 中有两个 5 对应不同的 user_id，如果用 set 的话，会导致新值覆盖旧值，并且使用集合更方便查询

问：如果想查询累计登陆次数为 5，且上次登录时间为 2011-3-1 的用户名

答：累计登陆次数为 5 的 user_id 存储的 key 名：

`login:login_times:5:user_id`

上次登录时间为 2011-3-1 的 user_id 存储的 key 名：

`login:last_login_time:2011-3-1:user_id`

以上两个集合做 sinter 或 sinterstore，可以得到结果 3

最后使用 `get login:user_id:3:name`

总结:

关系型数据转化为 KV 数据库, 方法如下:

先以主键作为 key 名依次存储其他字段的值

```
set "表名:主键名:主键的值:列名" "列的值"
```

再以除主键以外的字段作为 key 名, 存储主键的值

```
sadd "表名:列名:列值:主键名" "主键的值"
```

查找方式:

1、 如果由主键查找其他列的值, 直接 `get "表名:主键名:主键的值:列名"`

2、 如果由非主键的列 A 查找非主键列 B 的值
先 `smembers "表名:列 A:列 A 的值:主键名"`
根据结果再 `get "表名:主键名:主键的值:列 B"`

3、 如果是包含“与、或”的条件查询, 如列 A 与列 B 的与条件,
来查询列 C 的值则
先 `sinter/sunion "表名:列 A:列 A 的值:主键名" "表名:列 B:列 B 的值:主键名"`
再根据结果 `get "表名:主键名:主键的值:列"`

4、有如下关系型表 login，按要求回答以下问题，并写出必要的 Redis 命令：

| user_id | name | login_times | last_login_date |
|---------|--------|-------------|-----------------|
| 1 | ken | 5 | 2011-01-01 |
| 2 | dennis | 5 | 2011-03-01 |

(1) 将 login 表以 Redis 键值形式进行存储

答：

```
set login:user_id:1:login_times 5
```

```
set login:user_id:2:login_times 5
```

```
set login:user_id:1:last_login_time 2011-01-01
```

```
set login:user_id:2:last_login_time 2011-03-01
```

```
set login:user_id:1:name ken
```

```
set login:user_id:2:name dennis
```

```
sadd "login:name:ken:user_id" 1
```

```
sadd "login:name:dennis:user_id" 2
```

```
sadd "login:login_times:5:user_id" 1
```

```
sadd "login:login_times:5:user_id" 2
```

```
sadd "login:last_login_time:2011-01-01:user_id" 1
```

```
sadd "login:last_login_time:2011-03-01:user_id" 2
```

(2) 如何查找 user_id 为 1 的累计登陆次数？

```
get login:user_id:1:login_times
```

(3) 如何查找上次登录日期为 2011 年 3 月 1 日的用户姓名？

先输入命令

```
members "login:last_login_time:2011-03-01:user_id"
```

得到 2

再 `get login:user_id:2:name`

(4) 如何查找累计登录次数为 5 次，且上次登录时间为 2011-3-1 的用户名？

输入命令：

```
sinter login:login_times:5:user_id login:last_login_time:2011-03-01:user_id
```

得到 2

再 `get login:user_id:2:name`