一种多选项的存储与高效查询的解决方案

文章分类：JavaDemo； 标签：JavaCodeSnippet； 作者：Hackyle；

更新时间：Wed Jul 26 19:11:45 CST 2023

1. **背景**
2. 设计思想
   1. 将多选项转换为数字
      1. 例1
      2. 例2
   2. 查询原理
3. 项目启动
   1. 新增数据演示
   2. 查询数据演示
4. 查询示例
   1. 构造数据
   2. 插入到数据库
   3. 查询选择了"1-编程"这个选项的记录
   4. 查询选择了"2-听音乐唱歌，5-看电影"这些选项的记录

本文主要内容

- 对于多选项的值，如何保存？本文提供了一种非常规的方案。
- 对于记录在数据库中的多选项的值，如何快速查询那些记录是包含了某个（某些）选项？本文使用了“与位运算”解决查询问题。
- **源码地址：https://github.com/HackyleShawe/JavaDemos/tree/master/Examples/multi-options-storage-query-demo**

文章前置知识：SpringBoot、JDBCTemplate、位运算（与运算）、jQuery

内容导览

- 背景
- 设计思想
  - 将多选项转换为数字
    - 例1
    - 例2
  - 查询原理
- 项目启动
  - 新增数据演示
  - 查询数据演示
- 查询示例
  - 构造数据
  - 插入到数据库
  - 查询选择了"1-编程"这个选项的记录
  - 查询选择了"2-听音乐唱歌，5-看电影"这些选项的记录

背景

在项目开发中，如何保存多选项的值呢？例如下图中的职业发展和兴趣爱好

新增数据

Name:
Gender: Man Woman
Address:
CareerOption: 收入无上限 培训与发展 职业价值感 行业稳定性 社交与人脉 塑造个人品牌 团队综合素质 终身学习
InterestOption: 编程 听音乐唱歌 篮球 玩游戏 看电影 享美食 健身 旅游 看书 写作
Submit

- 最容易想到的就是，选择了哪些选项，就把该选项值存储起来。
- 在数据库层面设置一个VARCHAR，例如选择了"收入无上线、培训与发展、职业价值感"，前端就传递"收入无上线、培训与发展、职业价值感"，数据库就保存为"收入无上线、培训与发展、职业价值感"。

可是这样存在一个问题，我要查询那些人选择了其中的某个、某些选项，就很难实现。例如，查询那些人的兴趣爱好是"编程、篮球"，查询兴趣爱好是"看书、写作"的人数有多少。

为了解决这种多选项的高效查询问题，本人设计了一种方法，可以实现**快速、高效地查询**多选项中有某个、某些选项的所有记录。

设计思想

主要思想

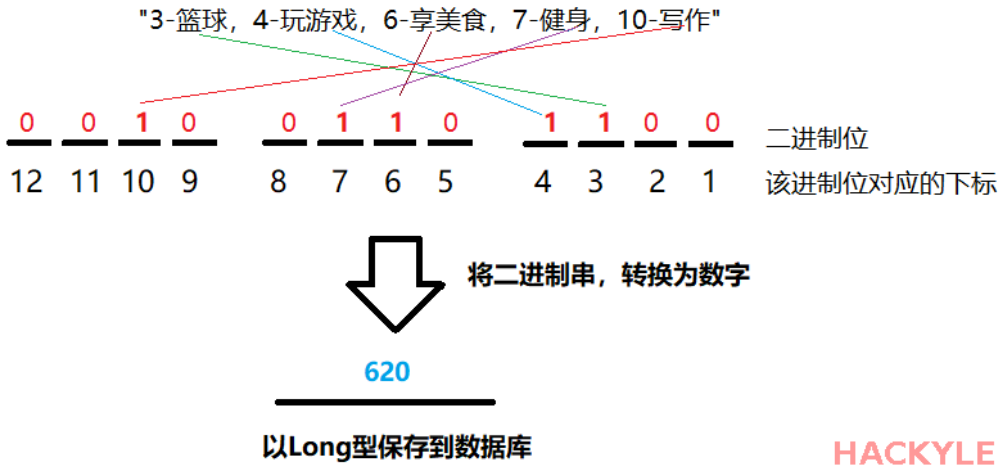
- 将多选项进行**编号**：例如：对职业发展的多选项进行编号为：1-收入无上限，2-培训与发展，3-职业价值感，4-行业稳定性，5-社交与人脉，6-塑造个人品牌，7-团队综合素质，8-终身学习；对兴趣爱好的多选项进行编号为：1-编程，2-听音乐唱歌，3-篮球，4-玩游戏，5-看电影，6-享美食，7-健身，8-旅游，9-看书，10-写作。

- 查询：前端还是只传选项编号；后端将其转换为数字；在数据库层面使用**位运算中的与运算**，匹配包含了参数选项的记录。

将多选项转换为数字

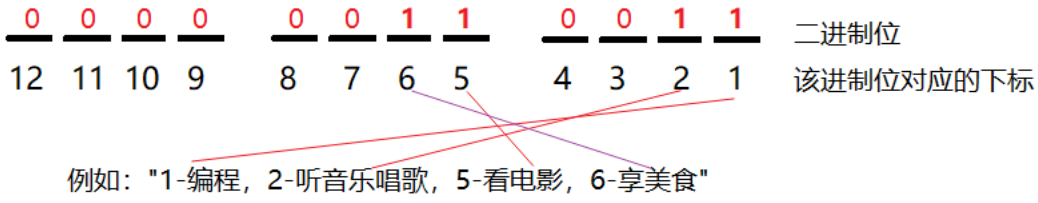
现以兴趣爱好为例，为其多选项定义编号：1-编程，2-听音乐唱歌，3-篮球，4-玩游戏，5-看电影，6-享美食，7-健身，8-旅游，9-看书，10-写作

根据二进制位下标与十进制数的互转：



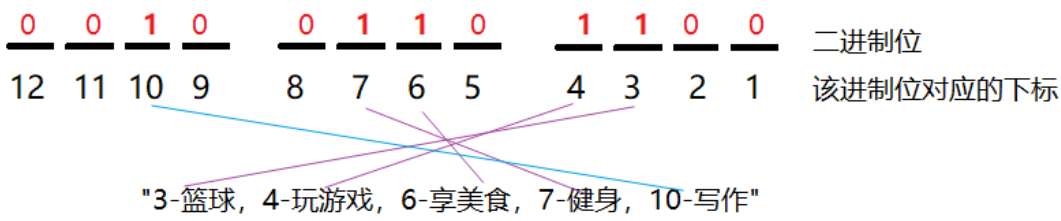
例1

用户A勾选的兴趣爱好为："1-编程，2-听音乐唱歌，5-看电影，6-享美食"
前端传递的串为：1,2,5,6
转换为二进制串：0011 0011
转换规则：在有选项编号出现的位下标的位置上填充1，其他位置填充0
转换为十进制后落库：51



例2

用户B勾选的兴趣爱好为："3-篮球，4-玩游戏，6-享美食，7-健身，10-写作"
前端传递的串为：3,4,6,7,10
转换为二进制串：0010 0110 1100
转换为十进制后落库：620



查询原理

查询的核心思想：与（&）位运算

（包含）

需求：查询所有选有"听音乐唱歌，看电影"的记录

⇩

前端传递来的查询条件："2,5"

转换查询条件为二进制（原理同新增保存时）：0010 0010

转换为十进制：34

⇩

根据**与运算**原理，编写为SQL: SELECT * FROM person WHERE **interests & 34 = 34** AND deleted = 0;

与运算过程：现假定数据库中interests有两条记录

1. 243(10)，0000 1111 0011(2)
2. 908(10)，0011 1000 1100(2)

⇩

查询条件34(10)，0010 0010(2)**与第一条记录匹配原理**：

0000 0001 0010
& 0000 1111 0011

0000 0001 0010 与运算结果仍为查询条件，说明这条记录包含了"2-听音乐唱歌，5-看电影"这些选项

查询条件34(10)，0010 0010(2)**与第二条记录匹配原理：**

0000 0001 0010
& 0011 1000 1100
#---
0000 0000 0000 与运算结果不为查询条件，说明这条记录不包含了"2-听音乐唱歌，5-看电影"这些选项

项目启动

Step 1：在application.yml中修改数据库连接参数

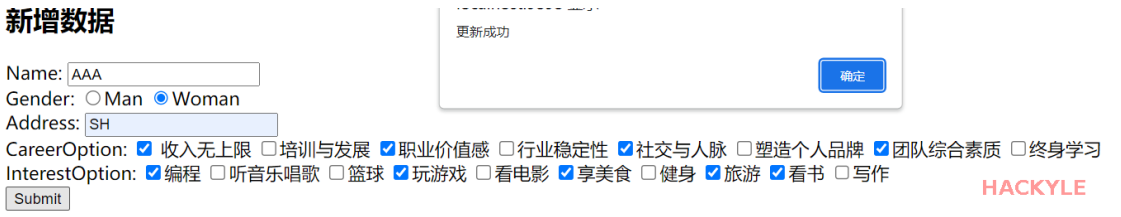
Step 2：执行resources/sql.sql下的SQL文件，初始化数据

Step 3：从启动类App.java启动

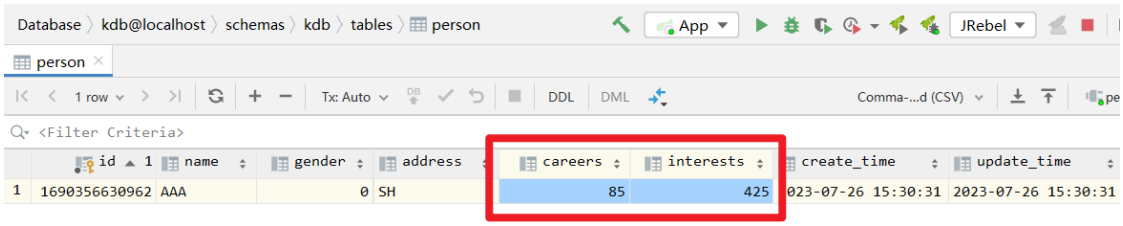
Step 4：启动成功后进入前端页面：http://localhost:9898/person.html

新增数据演示

必要数据，多选项进行勾选：

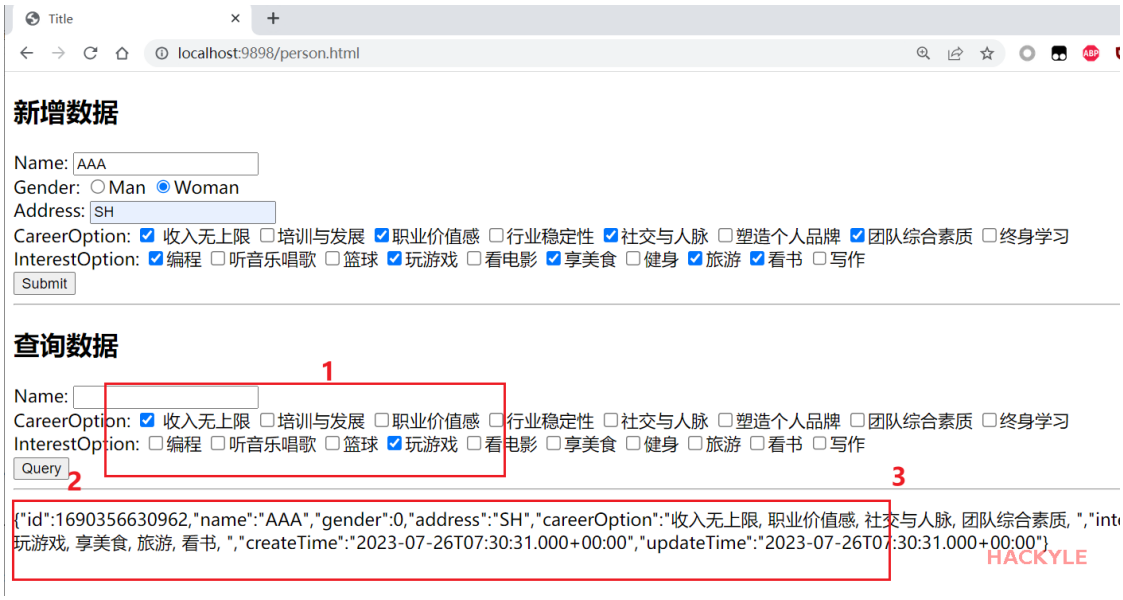


去数据库查看刚刚新增的记录：

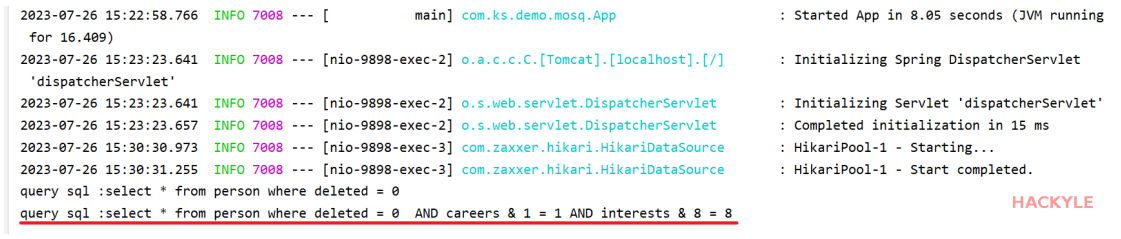


查询数据演示

选择查询条件，点击“Query”进行查询：



查看运行日志，显示执行的SQL：



查询示例

以多选项”兴趣爱好“为例，展示查询的工作原理

构造数据

AA选择了"1-编程，2-听音乐唱歌，5-看电影，6-享美食，7-健身，8-旅游"
前端传递的编号串：1,2,5,6,7,8
转换为二进制串：0000 1111 0011
转换为数字：243

转换为数字：908

CC选择了"2-听音乐唱歌，3-篮球，5-看电影，6-享美食，7-健身，9-看书，10-写作"
前端传递的编号串：2,3,5,6,7,9,10
转换为二进制串：0011 0111 0110
转换为数字：886

DD选择了"1-编程，3-篮球，4-玩游戏，6-享美食，7-健身，8-旅游，10-写作"
前端传递的编号串：1,3,4,6,7,8,10
转换为二进制串：0010 0110 1101
转换为数字：749

EE选择了"2-听音乐唱歌，4-玩游戏，5-看电影，7-健身，8-旅游，10-写作"
前端传递的编号串：2,4,5,7,8,10
转换为二进制串：0010 1101 1010
转换为数字：730

FF选择了"1-编程，2-听音乐唱歌，3-篮球，4-玩游戏，5-看电影，7-健身"
前端传递的编号串：1,2,3,4,5,7
转换为二进制串：0000 0101 1111
转换为数字：95

GG选择了"1-编程，3-篮球，4-玩游戏，6-享美食，8-旅游，10-写作"
前端传递的编号串：1,3,4,6,8,10
转换为二进制串：0010 1010 1101
转换为数字：685

插入到数据库

```sql
DROP TABLE IF EXISTS person;
CREATE TABLE person (
    id BIGINT AUTO_INCREMENT,
    name VARCHAR(50) DEFAULT NULL COMMENT '姓名',
    gender INT DEFAULT NULL COMMENT '性别，0-女，1-男',
    address VARCHAR(128) DEFAULT NULL COMMENT '地址',
    careers BIGINT DEFAULT NULL COMMENT '职业发展多选项',
    -- 兴趣爱好多选项。可选项：1-编程，2-听音乐唱歌，3-篮球，4-玩游戏，5-看电影，6-享美食
    -- 例如，全选："11 1111 1111"，保存为十进制=1023，全不选："00 0000 0000"，保存为十
    -- LONG最大支持64位，最多支持64个多选项的任意选择
    interests BIGINT DEFAULT NULL COMMENT '兴趣爱好多选项',
    create_time DATETIME DEFAULT NULL,
    update_time DATETIME DEFAULT NULL,
    deleted INT DEFAULT 0 COMMENT '是否删除：0-否，1-是',
    PRIMARY KEY(id)
);
```

```sql
-- 将上文中构造的数据，以SQL的形式插入到数据库中，只以多选项"兴趣爱好"为例
INSERT INTO person(name, gender, address, careers, interests, create_time, update_
VALUES ('AA', 1, 'SH CN', 1, 243, '2022-12-12','2023-12-12', 0),
       ('BB', 1, 'SH CN', 1, 908, '2022-12-12','2023-12-12', 0),
       ('CC', 1, 'SH CN', 1, 886, '2022-12-12','2023-12-12', 0),
       ('DD', 1, 'SH CN', 1, 749, '2022-12-12','2023-12-12', 0),
       ('EE', 1, 'SH CN', 1, 730, '2022-12-12','2023-12-12', 0),
       ('FF', 1, 'SH CN', 1, 95, '2022-12-12','2023-12-12', 0),
       ('GG', 1, 'SH CN', 1, 685, '2022-12-12','2023-12-12', 0);
```

查询选择了"1-编程"这个选项的记录

AA： 0000 1111 0011
BB： 0011 1000 1100
CC： 0011 0111 0110
DD： 0010 0110 1101
EE： 0010 1101 1010
FF： 0000 0101 1111
GG： 0010 1010 1101

将"1-编程"进行转换

- 转换为二进制：0000 0000 0001
- 转换为十进制：1

查询原理：将查询条件"0000 0000 0001"与AA~GG的二进制位进行与运算后，仍然为查询条件的记录，则是选择了"1-编程"这个选项的记录

查询过程

- 将查询条件与AA的二进制位进行与运算：

```
   0000 0000 0001
&  0000 1111 0011
#--------------------------
   0000 0000 0001  与运算结果仍为查询条件，说明这条记录包含了"1-编程"这个选项
```

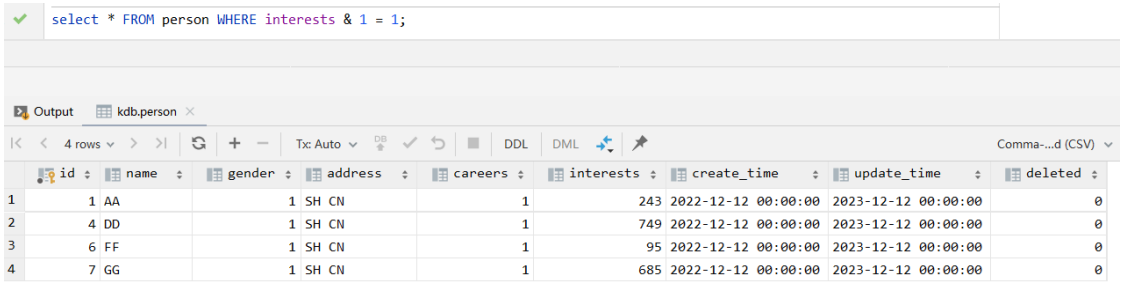
- 将查询条件与BB的二进制位进行与运算：

```
   0000 0000 0001
&  0011 1000 1100
#---------------------------
   0000 0000 0000   与运算结果不为查询条件，说明这条记录不包含了"1-编程"这个选项
```

- 其他记录运算同理
- 最终发现只有AA、DD、FF、GG的运算结果符合条件，这四个记录就是满足"选择了1-编程这个选项的所有记录"

SQL实现

```
select * FROM person WHERE interests & 1 = 1;
```



	id	name	gender	address	careers	interests	create_time	update_time	deleted
1	1	AA	1	SH CN	1	243	2022-12-12 00:00:00	2023-12-12 00:00:00	0
2	4	DD	1	SH CN	1	749	2022-12-12 00:00:00	2023-12-12 00:00:00	0
3	6	FF	1	SH CN	1	95	2022-12-12 00:00:00	2023-12-12 00:00:00	0
4	7	GG	1	SH CN	1	685	2022-12-12 00:00:00	2023-12-12 00:00:00	0

查询选择了"2-听音乐唱歌，5-看电影"这些选项的记录

目标：在多选项中查询选择了"2-听音乐唱歌，5-看电影"这些选项的记录

用户的"兴趣爱好"多选项（二进制形式）

AA： 0000 1111 0011
BB： 0011 1000 1100
CC： 0011 0111 0110
DD： 0010 0110 1101

将"2-听音乐唱歌，5-看电影"进行转换

- 转换为二进制：0000 0001 0010
- 转换为十进制：18

查询原理：将查询条件"0000 0001 0010"与AA~GG的二进制位进行与运算后，仍然为查询条件的记录，则是选择了"2-听音乐唱歌，5-看电影"这个选项的记录

查询过程

- 将查询条件与AA的二进制位进行与运算：

```
  0000 0001 0010
& 0000 1111 0011
#--------------------------
   0000 0001 0010  与运算结果仍为查询条件，说明这条记录包含了"2-听音乐唱歌，5-看电影"这
```

- 将查询条件与BB的二进制位进行与运算：

```
  0000 0001 0010
& 0011 1000 1100
#----------------------------
  0000 0000 0000   与运算结果不为查询条件，说明这条记录不包含了"2-听音乐唱歌，5-看电影"
```

- 其他记录运算同理
- 最终发现只有AA、CC、EE、FF的运算结果符合条件，这四个记录就是满足"选择了2-听音乐唱歌，5-看电影这些选项的所有记录"

SQL实现

```
select * FROM person WHERE interests & 18 = 18;
```

```
select * FROM person WHERE interests & 18 = 18;
```

Output kdb.person

4 rows Tx: Auto DDL DML Comma-...d (CSV

	id	name	gender	address	careers	interests	create_time	update_time
1	1	AA	1	SH CN	1	243	2022-12-12 00:00:00	2023-12-12 00:00:00
2	3	CC	1	SH CN	1	886	2022-12-12 00:00:00	2023-12-12 00:00:00
3	5	EE	1	SH CN	1	730	2022-12-12 00:00:00	2023-12-12 00:00:00
4	6	FF	1	SH CN	1	95	2022-12-12 00:00:00	2023-12-12 00:00:00

File Edit View Format Tools Table Help

Input comment, please

p 0 words tiny

SUBMIT RESET