

## 创建索引

创建索引主要有三种方式：

1. 创建表的时候直接创建索引；
2. 已存在的表上创建索引；
3. 修改表结构的方式添加索引。

### 6.2.1 新建表时创建索引

通常，新建表的时候就能为表创建索引。语法如下：

```
CREATE TABLE 表名 (  
    字段名 1 数据类型 [完整性约束条件],  
    字段名 2 数据类型 [完整性约束条件],  
    .....,  
    [UNIQUE | FULLTEXT | SPATIAL ] INDEX | KEY  
    [索引名] (字段名 1 [(长度)] [ASC | DESC])  
);
```

说明：

UNIQUE:可选。表示索引为唯一性索引。

FULLTEXT:可选。表示索引为全文索引。

SPATIAL:可选。表示索引为空间索引。

INDEX 和 KEY:用于指定字段为索引，两者选择其中之一就可以了，作用是一样的。

索引名:可选。给创建的索引取一个新名称。

字段名 1:指定索引对应的字段的名称，该字段必须是前面定义好的字段。

长度:可选。指索引的长度，必须是字符串类型才可以使用。

ASC:可选。表示升序排列。

DESC:可选。表示降序排列。

**【例 6-1】创建普通索引:**为 Lib 数据库创建 school\_bk 表，表结构参照 school 表，并为 schoolName 创建普通索引。在 MySQL 的命令行格式中，输入以下命令：

```
drop table if exists school_bk;  
CREATE TABLE school_bk (  
    schoolNo char(4) NOT NULL PRIMARY KEY,  
    schoolName varchar(20) NOT NULL,  
    index id_schoolName(schoolName)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
show create table school_bk;
```

**【例 6-2】创建唯一性索引:**为 Lib 数据库创建 school\_bk 表，表结构参照 school 表，并为 schoolName 创建唯一性索引。在 MySQL 的命令行格式中，输入以下命令：

```
drop table if exists school_bk;  
CREATE TABLE school_bk (  
    schoolNo char(4) NOT NULL PRIMARY KEY,  
    schoolName varchar(20) NOT NULL,
```

```

    UNIQUE index id_schoolName(schoolName)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
show create table school_bk;

```

**【例 6-3】创建组合索引:**为 Lib 数据库创建 book\_bk 表,表结构参照 book 表,并为 book 表中 (bookName, author, publishName) 创建组合索引。在 MySQL 的命令行格式中, 输入以下命令:

```

drop table if exists book_bk;
CREATE TABLE book_bk (
    bookNo char(4) NOT NULL PRIMARY KEY,
    classNo char(4) NOT NULL,
    bookName varchar(40) NOT NULL,
    author varchar(10) DEFAULT NULL,
    publishName varchar(20) DEFAULT NULL,
    publishDate datetime DEFAULT NULL,
    introduction varchar(100) DEFAULT NULL,
    onShelf char(2) DEFAULT '是',
    price float DEFAULT NULL,
    number int(11) NOT NULL DEFAULT '0',
    index id_book(bookName,author,publishName)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
show create table book_bk;

```

1. 请使用以下语句确认组合索引的使用效果

```
explain select * from book_bk where publishName=' aaa' and author=' bbb' ;
```

```
mysql> explain select * from book_bk where publishName=' aaa' and author=' bbb' ;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	book_bk	NULL	ALL	NULL	NULL	NULL	NULL	1	100.00	Using where

1 row in set, 1 warning (0.00 sec)

注: 从运行结果上看到, 上述查询没有使用到 id\_book 的索引, 仅仅使用了条件查询。

2. 请使用以下语句确认组合索引的使用效果

```
explain select * from book_bk where bookName=' aaa' and author=' bbb' ;
```

```
mysql> explain select * from book_bk where bookName=' aaa' and author=' bbb' ;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	book_bk	NULL	ref	id_book	id_book	155	const,const	1	100.00	NULL

1 row in set, 1 warning (0.00 sec)

注: 从运行结果上看到, 上述查询正确使用了 id\_book 的索引, 可见只有在查询条件中使用了创建索引时的第一个字段, 索引才会被使用。

3. 请使用以下语句确认组合索引的使用效果

```
explain select * from book_bk where bookName=' aaa' and publishName=' bbb' ;
```

```
mysql> explain select * from book_bk where bookName='aaa' and publishName='bbb';
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	book_bk	NULL	ref	id_book	id_book	122	const	1	100.00	Using index condition

```
1 row in set, 1 warning (0.00 sec)
```

注：从运行结果上看到，上述查询也使用了 id\_book 的索引，但只用到了查询条件中的第一个字段，而查询条件中的第二个字段是使用了索引下推功能，可见查询条件中字段的先后顺序十分关键，直接决定了 sql 在查询中是否能使用到该索引。。

**索引下推 (index condition pushdown )** 简称 ICP，在 Mysql5.6 的版本以上推出，用于优化查询。

在不使用 ICP 的情况下，在使用非主键索引(又叫普通索引或者二级索引)进行查询时，存储引擎通过索引检索到数据，然后返回给 MySQL 服务器，服务器然后判断数据是否符合条件。

在使用 ICP 的情况下，如果存在某些被索引的列的判断条件时，MySQL 服务器将这一部分判断条件传递给存储引擎，然后由存储引擎通过判断索引是否符合 MySQL 服务器传递的条件，只有当索引符合条件时才会将数据检索出来返回给 MySQL 服务器。

索引条件下推优化可以减少存储引擎查询基础表的次数，也可以减少 MySQL 服务器从存储引擎接收数据的次数。

#### 4. 请使用以下语句确认组合索引的使用效果

```
explain select * from book_bk where publishName='aaa' and bookName='ccc' and author='bbb';
```

```
mysql> explain select * from book_bk where publishName='aaa' and author='bbb' and bookName='ccc';
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	book_bk	NULL	ref	id_book	id_book	218	const,const,const	1	100.00	NULL

```
1 row in set, 1 warning (0.00 sec)
```

注：从运行结果上看到，上述查询完整的使用了 id\_book 的索引。

**【例 6-4】创建全文索引:**为 Lib 数据库创建 book\_bk 表，表结构参照 book 表，并为 introduction 创建全文索引。在 MySQL 的命令行格式中，输入以下命令：

```
drop table if exists book_bk;
CREATE TABLE book_bk (
    bookNo char(4) NOT NULL PRIMARY KEY,
    classNo char(4) NOT NULL,
    bookName varchar(40) NOT NULL,
    author varchar(10) DEFAULT NULL,
    publishName varchar(20) DEFAULT NULL,
    publishDate datetime DEFAULT NULL,
    introduction varchar(100) DEFAULT NULL,
    onShelf char(2) DEFAULT '是',
    price float DEFAULT NULL,
    number int(11) NOT NULL DEFAULT '0',
    fulltext index id_introduction(introduction)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
show create table book_bk;
```

**【例 6-5】创建短索引：**为 Lib 数据库创建 book\_bk 表，表结构参照 book 表，并为 introduction 创建长度为 10 的短索引。在 MySQL 的命令行格式中，输入以下命令：

```
drop table if exists book_bk;
CREATE TABLE book_bk (
    bookNo char(4) NOT NULL PRIMARY KEY,
    classNo char(4) NOT NULL,
    bookName varchar(40) NOT NULL,
    author varchar(10) DEFAULT NULL,
    publishName varchar(20) DEFAULT NULL,
    publishDate datetime DEFAULT NULL,
    introduction varchar(100) DEFAULT NULL,
    onShelf char(2) DEFAULT '是',
    price float DEFAULT NULL,
    number int(11) NOT NULL DEFAULT '0',
    index id_introduction(introduction(10))
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
show create table book_bk;
```