1.MySQL在创建数据表的时候创建索引

在MySQL中创建表的时候，可以直接创建索引。基本的语法格式如下：

```
CREATE TABLE 表名字段名 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])
);
```

- 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，两者选择其中之一即可以了，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：指定索引所对应的字段的名称，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。

MySQL创建普通索引

创建一个普通索引时，不需要加任何UNIQUE、FULLTEXT或者SPATIAL参数。

实例：创建一个名为index1的数据表，在表内的id字段上建立一个普通索引。

1. 创建普通索引的SQL代码如下：

```
CREATE TABLE index1(id INT,
        name VARCHAR(20),
        sex BOOLEAN,

INDEX(id)

);
```

在DOS提示符窗口中查看MySQL创建普通索引的操作效果。如下图所示：

从上图中可以看出，运行结果显示普通索引创建成功。

2. 使用SHOW CREATE TABLE语句查看表的结构。如下图所示：

从上图中可以看出，在id字段上已经建立了一个名为id的普通索引。

语句：

```
KEY `id` (`id`)
```

括号内的id是字段名称，括号外是指外面的id是索引名称。

3. 使用EXPLAIN语句查看索引是否被使用。SQL代码如下：

```
EXPLAIN SELECT * FROM index1 where id=1 \G
```

在DOS提示符窗口中查看使用EXPLAIN语句查看索引是否被使用的操作效果。如下图所示：

上图中的结果显示，possible_keys和key的值都为id，说明id索引已经存在，并且查询时已经使用了索引。

MySQL创建唯一性索引

如果使用UNIQUE参数进行约束，则可以创建唯一性索引。

实例：创建一个名为index2的数据表，在表内的id字段上建立一个唯一性索引。并且设置id字段以升序的形式进排列。

1. 创建一个唯一性索引的SQL代码如下：

```
CREATE TABLE index2(id INT UNIQUE,
        name VARCHAR(20),

UNIQUE INDEX index2_id(id ASC)

);
```

index2_id理为唯一性索引的的一个新名字。

在DOS提示符窗口中查看MySQL创建唯一性索引的操作效果。如下图所示：

从上图中可以看出，运行结果显示创建成功。

2. 使用SHOW CREATE TABLE语句查看表的结构，SQL代码如下：

```
SHOW CREATE TABLE index2 \G
```

在DOS提示符窗口中查看使用SHOW CREATE TABLE语句查看表的结构的效果，如下图所示：

从上图中可以看出，在id字段上建立了名为id和index2_id的两个唯一性索引，这样做，可以提高数据的查询速度。

如果在创建index2表时，id字段没有进行唯一性结束，如下所示：

```
CREATE TABLE index2(id INT,
        name VARCHAR(20),

UNIQUE INDEX index2_id(id ASC)

);
```

则也可以在id字段上成功创建名为index2_id的唯一性索引。但是，这样可能达不到提高查询速度的目的。

MySQL创建全文索引

全文索引使用FULLTEXT参数，并且只能在CHAR、VARCHAR或TEXT类型的字段上创建。

全文索引只用于全文搜索。

现在，MyISAM引擎引擎和InnoDB引擎引擎都支持全文索引。

实例：创建一个名为index3的数据表，在表中的info字段上建立名为index3_info的全文索引。

1. 创建全文索引的SQL代码如下：

```
CREATE TABLE index3(id INT,
        info VARCHAR(20),

FULLTEXT INDEX index3_info(info)

)ENGINE=MyISAM;
```

如果设置ENGINE=InnoDB，则可以在InnoDB存储引擎上创建全文索引。

在DOS提示符窗口中查看MySQL创建全文索引的操作效果。如下图所示：

从上图中可以看出，代码的执行结果显示创建成功。

2. 使用SHOW CREATE TABLE语句查看index3数据表的结构。如下图所示：

从上图中可以看出，在info字段上已经建立了一个名为index3_info的全文索引。

注意

我使用的是MySQL 5.6.19版本，已经可以在InnoDB存储引擎中创建全文索引了。

全文索引非常适合于大型数据集，对于小的数据集，它的用处可能比较小。

MySQL创建单列索引

单列索引是在数据表的单个字段上创建的索引。一个表中可以创建多个单列索引。唯一性索引和普通索引等都为单列索引。

实例：创建一个名为index4的数据表，在表中的subject字段上建立名为index4_st的单列索引。

1. 创建单列索引的SQL代码如下：

```
CREATE TABLE index4(id INT,
        subject VARCHAR(30),

INDEX index4_st(subject(10))

);
```

在DOS提示符窗口中查看MySQL创建单列索引的操作效果。如下图所示：

从上图中可以看出，代码执行的结果显示创建成功。

2. 使用SHOW CREATE TABLE语句查看index4数据表的结构。如下图所示：

从上图中可以看出，在subject字段上已经建立了一个名为index4_st的单列索引。

注意：subject字段长度为30，而index4_st这里的索引长度只有10。这样做是为了提高查询速度。对于字符型的数据，可以不用查询全部保存，而只查询它前面的若干字符信息。

MySQL创建多列索引

创建多列索引是在表的多个字段上创建一个索引。

实例：创建一个名为index5的数据表，在表中的name和sex字段上建立名为index5_ns的多列索引。

1. 创建多列索引的SQL代码如下：

```
CREATE TABLE index5(id INT,
        name VARCHAR(20),
        sex CHAR(4),

INDEX index5_ns(name,sex)

);
```

在DOS提示符窗口中查看MySQL创建多列索引的操作效果。如下图所示：

从上图中可以看出，代码的执行结果显示index5_ns索引创建成功。

2. 使用SHOW CREATE TABLE语句查看index5数据表的结构。如下图所示：

从上图中可以看出，name和sex字段上已经建立了一个名为index5_ns的多列索引。

3. 多列索引中，只有查询条件中使用了这些字段中第一个字段时，索引才会被使用。

先在index5数据表中添加一些数据之后，然后使用EXPLAIN语句可以查看索引的使用情况。如果只是使用name字段作为查询条件进行查询，如下图所示：

从上图中可以看出，possible_keys和key的值都是index5_ns，Extra（额外信息）显示正在使用索引，这说明使用name字段进行索引时，索引index5_ns已经被使用。

4. 如果只使用sex字段作为查询条件进行查询，如下图所示：

从上图中可以看出，possible_keys和key的值都是NULL，Extra（额外信息）显示正在使用where条件查询，而未使用索引。

提示

使用多列索引时一定要特别注意，只有使用了索引中的第一个字段时才会触发索引，如果没有使用索引中的第一个字段，那么这个多列索引就不会起作用。因此，在优化查询速度时，可以考虑优化多列索引。

MySQL创建空间索引

使用SPATIAL参数能够创建空间索引。创建空间索引时，表的存储引擎必须是MyISAM类型，而且，索引字段必须有非空约束。

实例：创建一个名为index6的数据表，在表中的space字段上建立名为index6_sp的空间索引。

1. 创建空间索引的SQL代码如下：

```
CREATE TABLE index6(id INT,
        space GEOMETRY NOT NULL,

SPATIAL INDEX index6_sp(space)

)ENGINE=MyISAM;
```

在DOS提示符窗口中查看MySQL创建空间索引的操作效果。如下图所示：

从上图可以看出，代码执行结果显示空间索引创建成功。

2. 使用SHOW CREATE TABLE语句可看index6数据表的结构。如下图所示：

从上图中可以看出，在space字段上已经建立了一个名为index6_sp的空间索引。

注意，space字段要非空的，而且数据类型是GEOMETRY类型，这个类型是空间数据类型。

空间数据类型包括GEOMETRY、POINT、LINESTRING和POLYGON共四种。

2.添加索引

```
1.添加PRIMARY KEY（主键索引）
mysql>ALTER TABLE `table_name` ADD PRIMARY KEY (`column`)
2.添加UNIQUE(唯一索引)
mysql>ALTER TABLE `table_name` ADD UNIQUE (
`column`
)
3.添加INDEX(普通索引)
mysql>ALTER TABLE `table_name` ADD INDEX index_name (`column`)
4.添加FULLTEXT(全文索引)
mysql>ALTER TABLE `table_name` ADD FULLTEXT (`column`)
5.添加多列索引
mysql>ALTER TABLE `table_name` ADD INDEX index_name (`column1`, `column2`, `column3`)
```