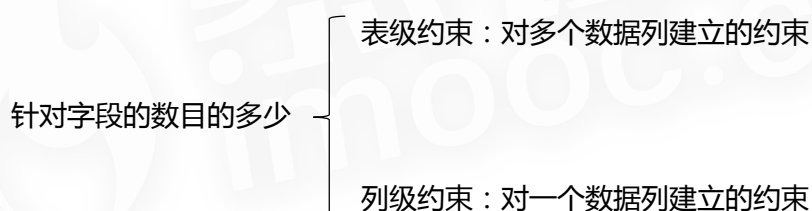


MySQL 之外键约束

约束概述

➤ 创建约束的目的就是保证数据的完整性和一致性。

➤ 约束分类：



按照功能来划分：NOT NULL(非空约束)

PRIMARY KEY(主键约束)

UNIQUE KEY(唯一约束)

DEFAULT(默认约束)

FOREIGN KEY(外键约束)

这里我们主要就外键约束进行详细讲解。

创建外键约束的要求

创建外键约束的目的是保持数据一致性，完整性；以及实现一对一或一对多关系。

创建外键约束的要求有以下几点：

1) 父表和子表必须使用相同的存储引擎，而且禁止使用临时表。

📖 具有外键列的表称为子表；子表所参照的表称为父表。

2) 数据表的存储引擎只能为 InnoDB。

3) 外键列和参照列必须具有相似的数据类型。其中数字的长度或是否有符号位必须相同；

而字符的长度则可以不同。

📖 加 FOREIGN KEY 关键字的列称为外键列；外键列所参照的列称为参照列。

4) 外键列和参照列必须创建索引。如果外键列不存在索引的话,MySQL 将自动创建索引。

如果参照列不存在索引的话,MySQL 不会自动创建索引。

📖 MySQL 会为主键自动创建索引。

存储引擎

概念：

存储引擎也叫表类型，指的是数据表的存储机制、索引方案等配套相关功能。

MySQL 中的数据用各种不同的技术存储在文件（或者内存）中。这些技术中的每一种技术都使用不同的存储机制、索引技巧、锁定水平并且最终提供广泛的不同的功能和能力。

通过选择不同的技术，你能够获得额外的速度或者功能，从而改善你的应用的整体功能。

MySQL5.5 以后默认使用 **InnoDB** 存储引擎 其中 InnoDB 和 BDB 提供事务安全表，其它存储引擎都是非事务安全表。

现在我们就最常用的两种存储引擎 MyISAM 与 InnoDB 进行比较：

	InnoDB	MyISAM
是否支持外键	支持	不支持
每个表所含文件	每个表只有一个 table.frm 文件， 所有的 Innodb 引擎的表的数据将 存放在 ibdata*里面	每个表会生成 3 个文件： table.MYI（索引文件）、 table.MYD（数据文件）、 table.frm（表结构文件）
支持锁的类型	支持行级锁，优势在于更新和删除	支持表级锁，优势在于插入和检索
是否支持事务	支持	不支持

创建外键约束

- 1 创建一个地址管理表 `address` 作为父表，然后查看数据表的创建信息，确认存储引擎是否为 InnoDB：

```
MySQL 5.5 Command Line Client

mysql> CREATE TABLE address(
  -> id SMALLINT UNSIGNED PRIMARY KEY,
  -> name VARCHAR(20) NOT NULL
  -> );
Query OK, 0 rows affected (0.06 sec)
```

SHOW CREATE TABLE address;

```
MySQL 5.5 Command Line Client

-> ;
ERROR 1005 (HY000): Can't create table 'users.teacher' (errno: 150)
mysql> SHOW CREATE TABLE address;
+-----+-----+
| Table | Create Table |
+-----+-----+
| address | CREATE TABLE 'address' (
  'id' smallint(5) unsigned NOT NULL,
  'name' varchar(20) NOT NULL,
  PRIMARY KEY ('id')
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+-----+

1 row in set (0.00 sec)

mysql>
```

默认引擎为InnoDB

由此我们保证了创建数据表的时候使用的是默认引擎 InnoDB。

- 2 使用外键约束创建 `teacher` 表作为子表。

```
MySQL 5.5 Command Line Client

mysql> CREATE TABLE address<
-> id SMALLINT UNSIGNED PRIMARY KEY,
-> name VARCHAR(20) NOT NULL
-> >;
Query OK, 0 rows affected (0.06 sec)

mysql> CREATE TABLE teacher<
-> t_id TINYINT UNSIGNED PRIMARY KEY,
-> t_name VARCHAR(20) NOT NULL,
-> t_add VARCHAR(20) NOT NULL,
-> FOREIGN KEY (t_add) REFERENCES address (name) ON UPDATE CASCADE
-> >;
ERROR 1005 (HY000): Can't create table 'users.teacher' (errno: 150)
```

保证了数据类型相似

我们保证了数据类型相似,也知道存储引擎都为 InnoDB,但是创建数据时有错误信息,这时我们就需要注意是否满足创建外键要求的第四点,外键列和参照列必须创建索引。

先查看已建表 address 的索引情况:

SHOW INDEXES FROM address;

```
MySQL 5.5 Command Line Client

-----+
1 row in set (0.00 sec)

mysql> SHOW INDEXES FROM address\G;
***** 1. row *****
      Table: address
    Non_unique: 0
      Key_name: PRIMARY
    Seq in index: 1
    Column_name: id
      Collation: A
    Cardinality: 0
      Sub_part: NULL
        Packed: NULL
          Null:
    Index_type: BTREE
      Comment:
    Index_comment:
1 row in set (0.00 sec)

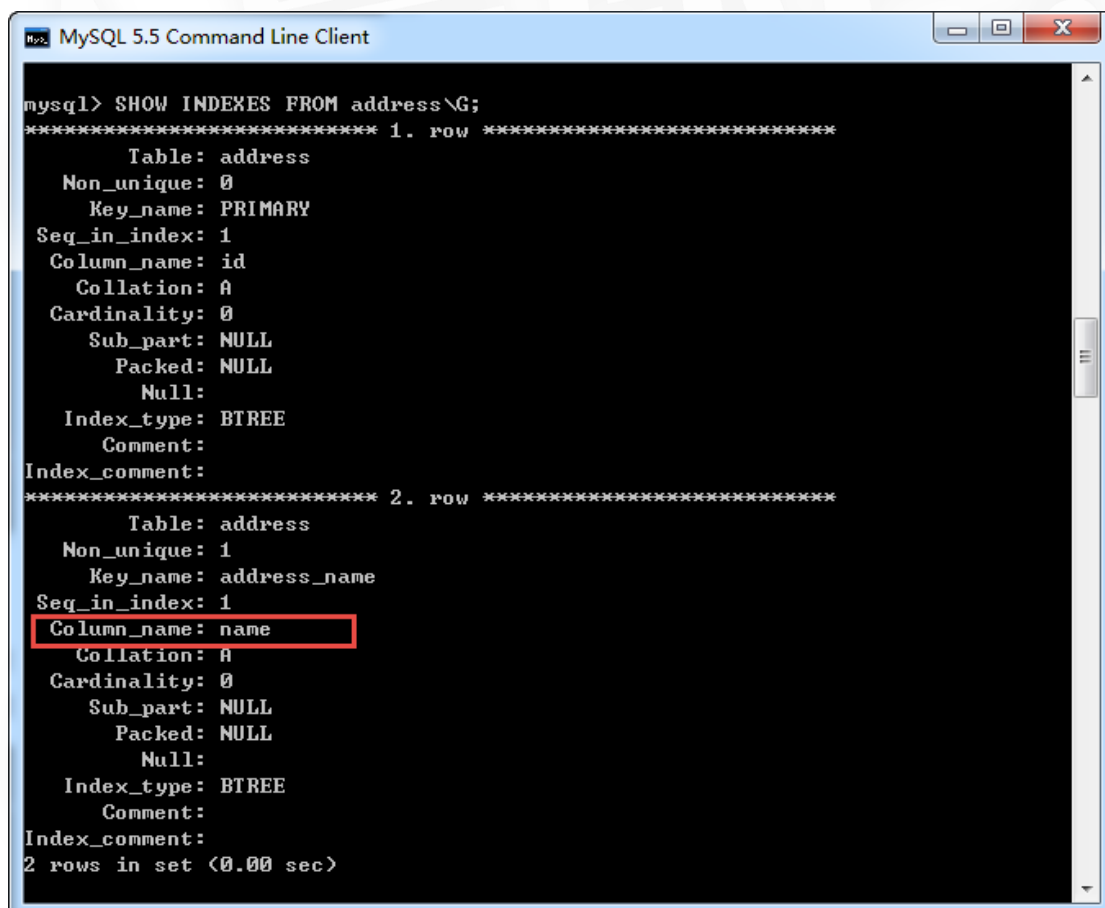
ERROR:
No query specified

mysql>
```

我们可以看出 teacher 表参照的列 name 并没有索引，现在我们为 name 列创建索引

```
mysql> CREATE INDEX address_name ON address(name);  
Query OK, 0 rows affected (0.09 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

这时我们再查看 address 表的索引情况：此时参照列 name 已经创建索引



```
mysql> SHOW INDEXES FROM address\G;  
***** 1. row *****  
Table: address  
Non_unique: 0  
Key_name: PRIMARY  
Seq_in_index: 1  
Column_name: id  
Collation: A  
Cardinality: 0  
Sub_part: NULL  
Packed: NULL  
Null:  
Index_type: BTREE  
Comment:  
Index_comment:  
***** 2. row *****  
Table: address  
Non_unique: 1  
Key_name: address_name  
Seq_in_index: 1  
Column_name: name  
Collation: A  
Cardinality: 0  
Sub_part: NULL  
Packed: NULL  
Null:  
Index_type: BTREE  
Comment:  
Index_comment:  
2 rows in set (0.00 sec)
```

- 3 这时我们在重新创建表 teacher ,可以发现 teacher 表创建成功 并且成功创建了外键。



```
MySQL 5.5 Command Line Client
No query specified

mysql> CREATE TABLE teacher(
-> t_id TINYINT UNSIGNED PRIMARY KEY,
-> t_name VARCHAR(20) NOT NULL,
-> t_add VARCHAR(20) NOT NULL,
-> FOREIGN KEY (t_add) REFERENCES address (name) ON UPDATE CASCADE
-> );
Query OK, 0 rows affected (0.10 sec)

mysql> SHOW CREATE TABLE teacher;
+-----+
| Table | Create Table
+-----+
| teacher | CREATE TABLE `teacher` (
  `t_id` tinyint(3) unsigned NOT NULL,
  `t_name` varchar(20) NOT NULL,
  `t_add` varchar(20) NOT NULL,
  PRIMARY KEY (`t_id`),
  KEY `t_add` (`t_add`),
  CONSTRAINT `teacher_ibfk_1` FOREIGN KEY (`t_add`) REFERENCES `address` (`name`)
  ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 ;
+-----+
1 row in set (0.00 sec)

mysql>
```

建表成功

成功创建外键

我们还可以查看下 MySQL 是否为外键列即 t_add 自动创建了索引。

```
MySQL 5.5 Command Line Client

mysql> SHOW INDEXES FROM teacher\G;
***** 1. row *****
      Table: teacher
    Non_unique: 0
      Key_name: PRIMARY
    Seq_in_index: 1
      Column_name: t_id
      Collation: A
    Cardinality: 0
      Sub_part: NULL
        Packed: NULL
          Null:
    Index_type: BTREE
      Comment:
    Index_comment:
***** 2. row *****
      Table: teacher
    Non_unique: 1
      Key_name: t_add
    Seq_in_index: 1
      Column_name: t_add
      Collation: A
    Cardinality: 0
      Sub_part: NULL
        Packed: NULL
          Null:
    Index_type: BTREE
      Comment:
    Index_comment:
```

通过这个例子来帮助大家理解创建外键的第四点要求：外键列和参照列必须创建索引。

如果外键不存在索引的话，MySQL 将自动创建索引。如果参照列不存在索引的话，MySQL 不会自动创建索引。