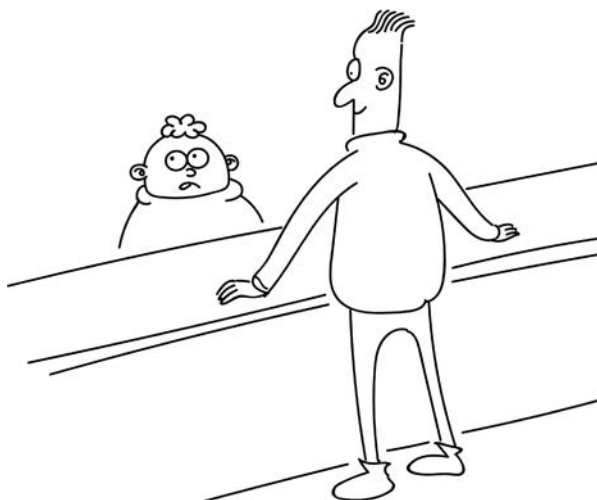




索引也是数据库中的对象,它可以提高查询数据的速度,但会影响插入和删除数据的效率。表的索引与书籍目录类似,在索引中保存了表中记录位置的相关信息,在运行查询时,首先在索引中查询,而不是直接在表中查询,查询到相关记录之后立即跳转到表中存储记录的位置。这与看书时先在目录中查找相关主题,找到后再到书中查找具体内容,在原理上是一样的。

本章介绍索引管理。

5.1 创建索引

创建索引的语法格式如下。

```
CREATE [UNIQUE] INDEX index_name ON table(field)
```

其中 UNIQUE 用于声明创建唯一索引,INDEX index_name 是要创建的索引名,table 是要创建的索引所在的表,field 是创建的索引所在的字段。

创建索引的示例代码如下。



```
-- 在 EMP 表的 EMPNO 字段上创建索引
CREATE INDEX emp_no_index ON EMP(EMPNO);
```

上述代码执行后会在 EMP 表的 EMPNO 字段上创建索引 emp_no_index，使用 MySQL Workbench 工具创建索引如图 5-1 所示。

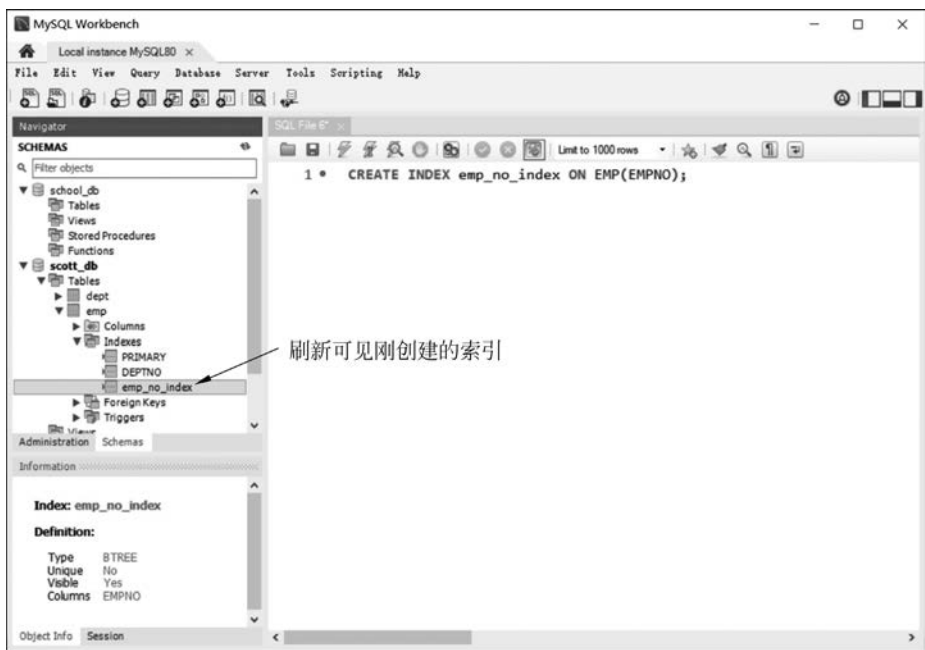


图 5-1 创建索引

由图 5-1 可见，除了刚创建的 emp_no_index 索引外，还有 PRIMARY 和 DEPTNO 两个索引，这两个索引并没有显示地创建，这是因为它们是在创建主键 (PRIMARY) 和外键 (DEPTNO) 的过程中伴随创建的。

💡提示 在创建了索引之后，每当 SQL 语句的 WHERE 子句中引用了索引中的字段，都会大大提高查询速度。

下面比较如下两条查询语句。

```
SELECT * FROM EMP; ①
SELECT * FROM EMP WHERE EMPNO IS NOT NULL; ②
```

上述两条查询语句虽然查询结果相同，但是在表中数据量比较大的情况下，第②条查询语句将优于第①条查询语句。

5.1.1 创建多字段组合索引

为了充分发挥索引的作用，还可以创建多个字段的组合索引，示例代码如下。

```
CREATE INDEX emp_ENAME_JOB_index ON EMP(ENAME,JOB);
```

上述代码执行后会在 EMP 表的 ENAME 和 JOB 字段上创建索引 emp_ENAME_JOB_index,使用 MySQL Workbench 工具创建多字段组合索引如图 5-2 所示。

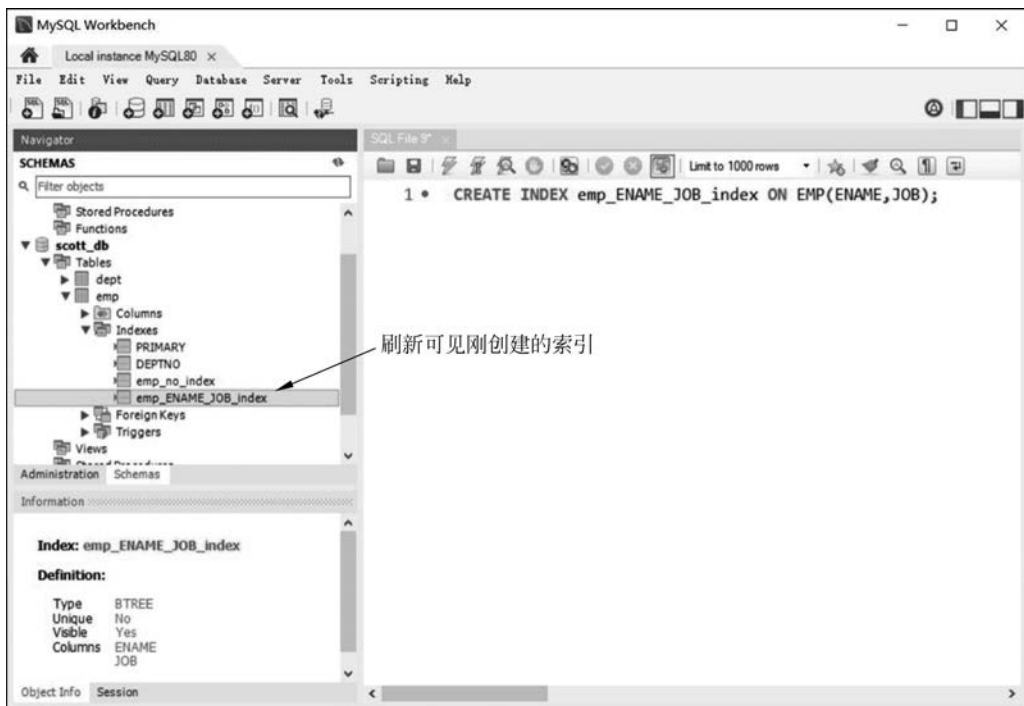


图 5-2 创建多字段组合索引

5.1.2 创建唯一索引

在创建索引时还可以添加 UNIQUE 子句,以创建唯一索引。创建唯一索引的字段,在创建索引的同时添加了 UNIQUE 约束,从而保证数据不会重复。

创建唯一索引示例代码如下。

```
CREATE UNIQUE INDEX emp_no_index2 ON EMP(ENAME);
```

上述代码执行后会在 EMP 表的 ENAME 字段上创建唯一索引 emp_no_index2,使用 MySQL Workbench 创建唯一索引如图 5-3 所示。

为了测试唯一索引的 UNIQUE 约束,可以插入与 ENAME 字段相同的数据,测试代码如下。

```
INSERT INTO EMP (EMPNO,ENAME,JOB) VALUES (8888,'刘备','总经理');
INSERT INTO EMP (EMPNO,ENAME,JOB) VALUES (8889,'刘备','大老板');
```

插入与 ENAME 字段相同的数据,如果使用 MySQL Workbench 工具测试,则第一条数据可以插入,而第二条数据不能插入,如图 5-4 所示。

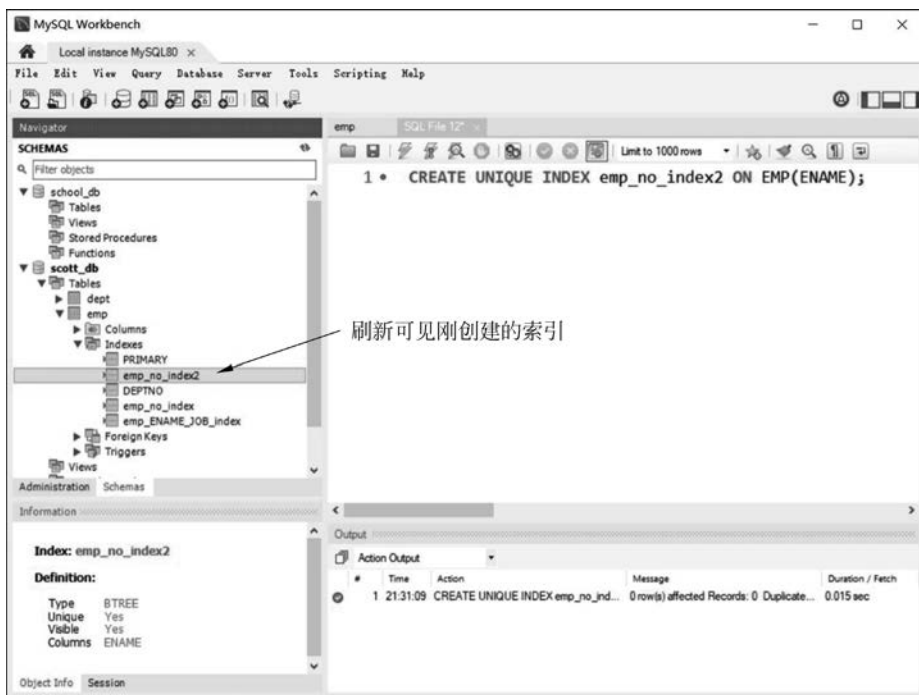


图 5-3 创建唯一索引

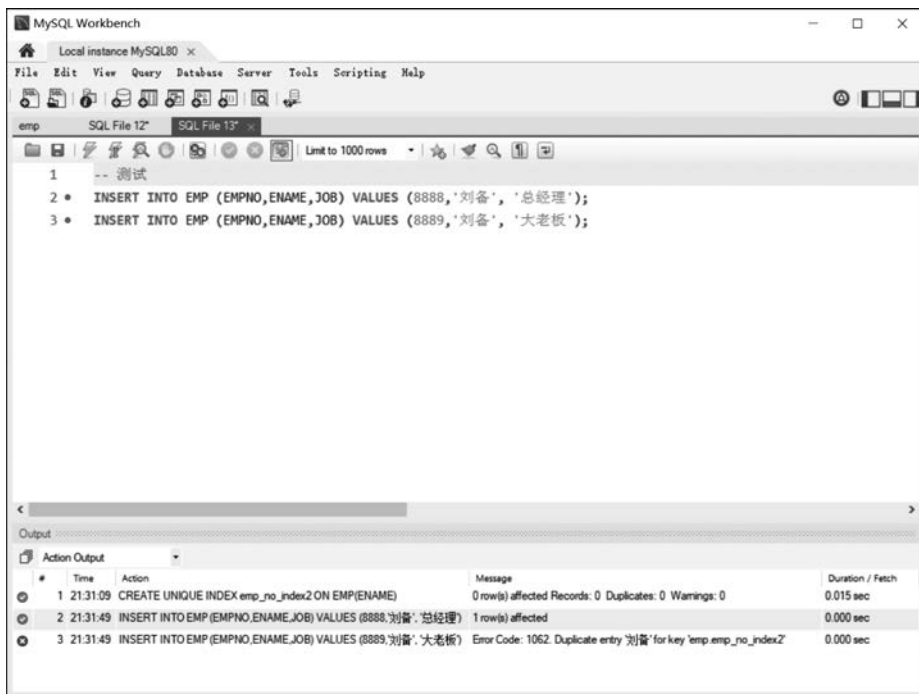


图 5-4 测试唯一索引



微课视频

5.2 删除索引

既然可以创建索引,当然也可以删除索引,删除索引的语法格式如下。

```
DROP INDEX index_name ON table
```

删除索引示例代码如下。

```
DROP INDEX emp_no_index2 ON EMP;
```

上述代码会删除 EMP 表中的 emp_no_index2 索引,如果使用 MySQL Workbench 工具测试,则可见 emp_no_index2 索引已经被删除,如图 5-5 所示。

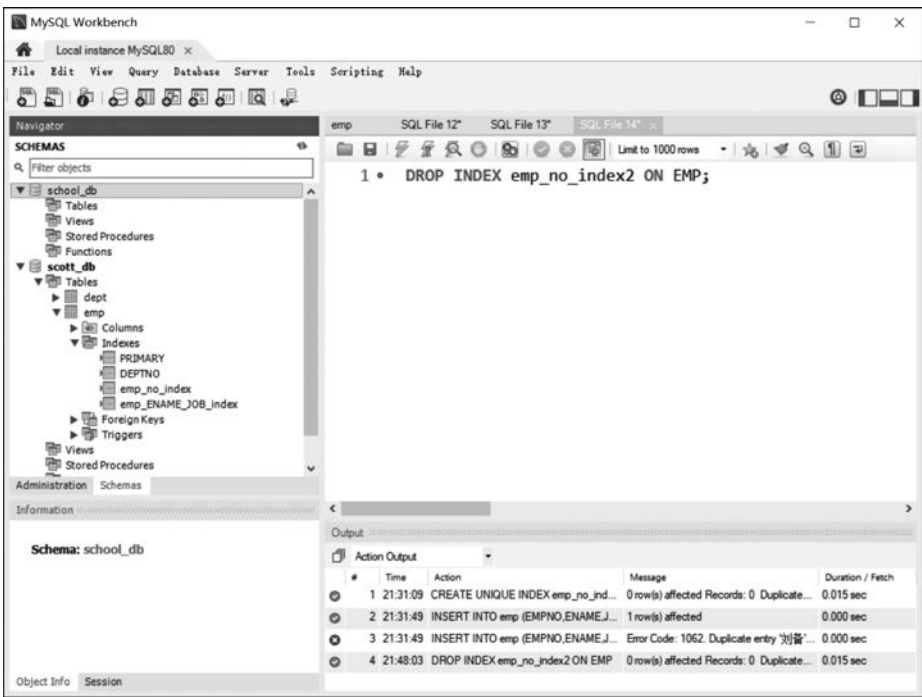


图 5-5 删除索引

5.3 使用索引的最佳实践

虽然从语法上看创建索引很简单,但是使用好索引并不是一件容易的事情。

在数据库创建索引存在着很大的误区,很多人盲目地认为对一个表创建越多的索引,就越可以提高数据库性能,但事实并非如此。索引一方面可以提高查询速度,但另一方面会降低插入和删除数据的速度。下面总结使用索引的一些最佳实践。

在哪些字段上创建索引,应该遵守如下原则。



微课视频

- (1) 大量值：如果在存储大量值的字段上创建索引,索引会更好地发挥作用。
- (2) 在查询中经常使用：在查询的 WHERE 子句中使用的字段上创建索引,能提高查询速度。
- (3) 在表连接操作中经常使用：在表连接时,如果在连接字段上创建索引,也可以提高查询速度。有关表连接操作将在第 10 章介绍。

 **提示** 事实上很难为制作索引整理出一个一般性的规则,因为在不同数据库中查询优化程序可能有很大的区别。可以提供的最好的建议是:如果在某一字段上查询速度缓慢,那么可以创建索引,如果索引使得性能提高,就保留索引,否则就删除索引。

5.4 动手练一练

1. 简答题

请简述创建索引的意义。

2. 选择题

下列哪些语句可以创建索引? ()

- A. DROP VIEW
- B. CREATE VIEW
- C. CREATE INDEX
- D. CREATE TABLE

3. 判断题

- (1) 索引可以提高更新数据的速度。 ()
- (2) 索引可以提高查询速度。 ()
- (3) 在存储大量值的字段上创建索引,索引会更好地发挥作用。 ()
- (4) 在查询的 WHERE 子句中使用的字段上创建索引,能提高查询速度。 ()