

第 5 章



索引管理

索引也是数据库中的一种对象,它可以提高查询数据的速度,但在插入和删除数据时,则会降低效率,在索引中保存了表中记录位置的相关信息,表的索引可以与图书目录类似,在运行查询时,首先在索引中查询,而不是在表本身中查找,之后立即跳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,然后到书中找具体内容的原理是一样的。

本章介绍索引管理。



微课视频

5.1 创建索引

创建索引的语法结构如下:

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

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

创建索引的示例代码如下:

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

上述代码执行后,会在员工表的 EMPNO 字段上创建索引 emp_no_index,使用 MySQL Workbench 创建索引如图 5-1 所示。

从图 5-1 中可见除了刚创建的 emp_no_index 索引外,还有 PRIMARY 和 DEPTNO 两个索引,这些索引并没有显式地创建,它们是在创建了主键 (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 子句引用索引中的字段时,会大大提高查询速度。

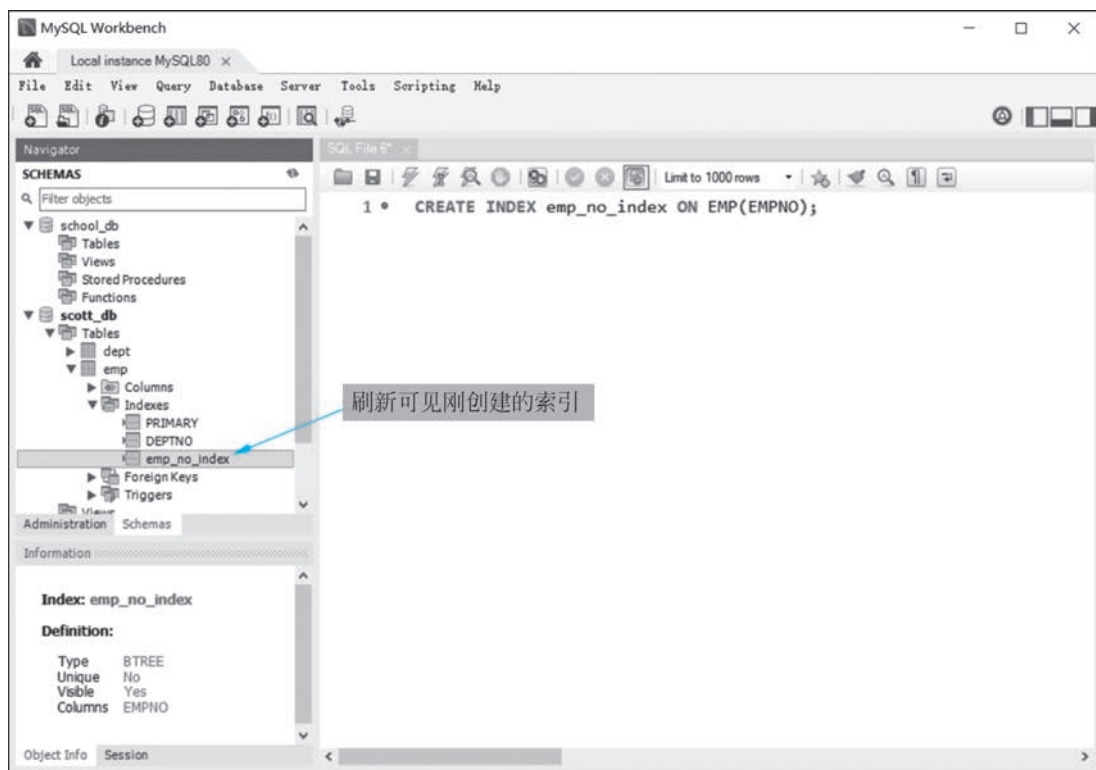


图 5-1 创建索引

两个查询的比较如下：

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,但是如果表中数据量比较大的情况下,第②行查询语句优于第①行的查询语句。

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员工表的 ENAME 和 JOB 字段上创建索引 emp_ENAME_JOB_index,使用 MySQL Workbench 创建组合索引如图 5-2 所示。

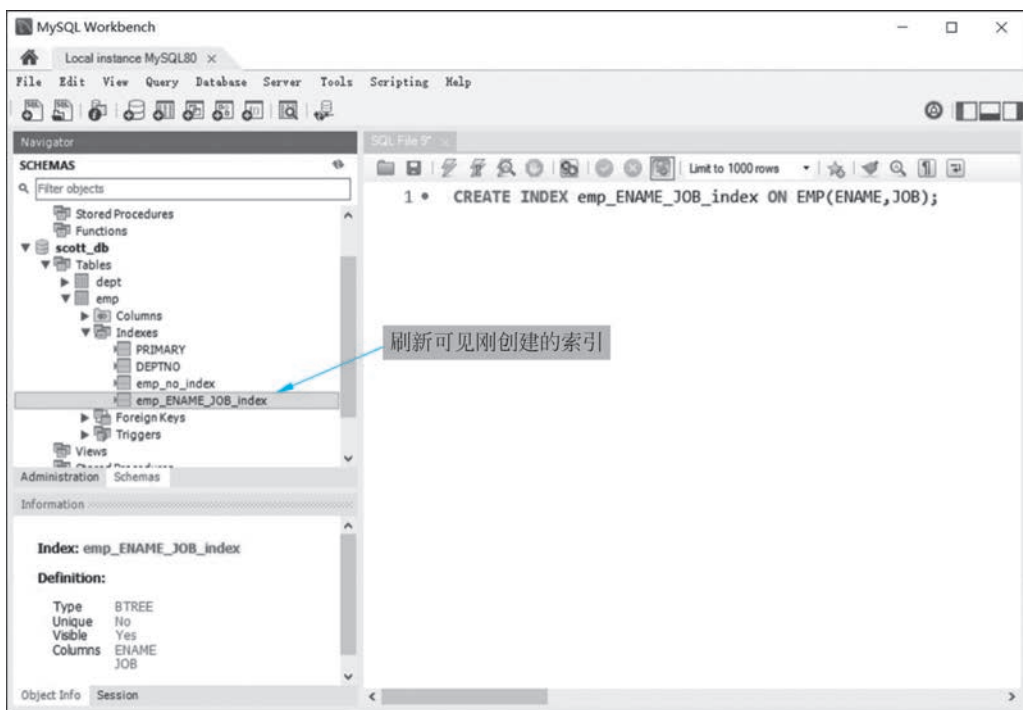


图 5-2 创建组合索引

5.1.2 创建唯一索引

在创建索引时还可以添加 UNIQUE 子句创建**唯一索引**,那么什么是唯一索引呢? 唯一索引是指在创建索引的同时,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员工表的 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 ENAME 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');
```

插入的 ENAME 数据相同,如果使用 MySQL Workbench 测试,结果如图 5-4 所示,可见第 1 条数据可以插入,而第 2 条数据不能插入。

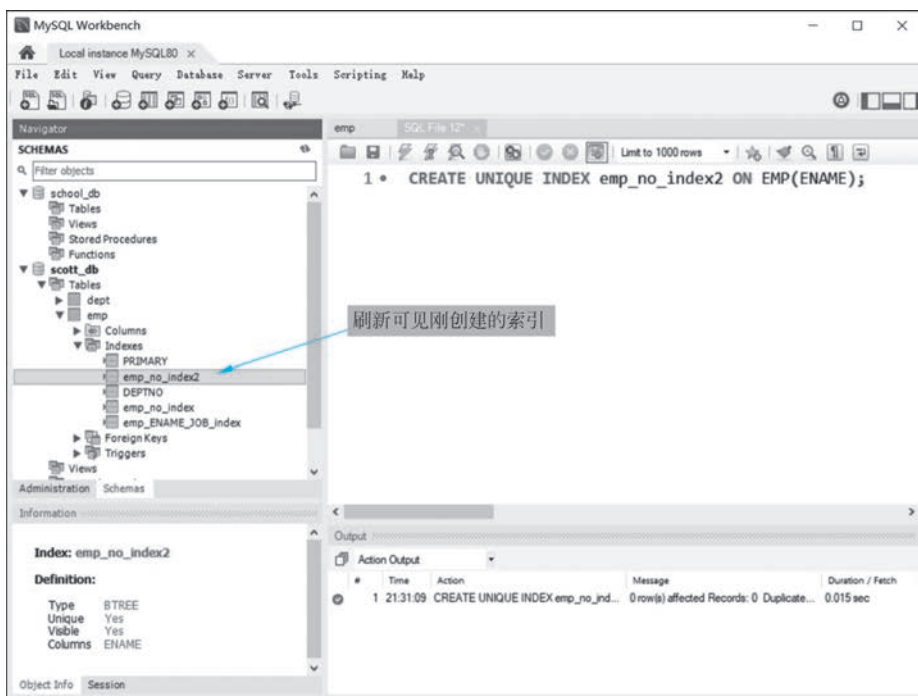


图 5-3 创建唯一索引

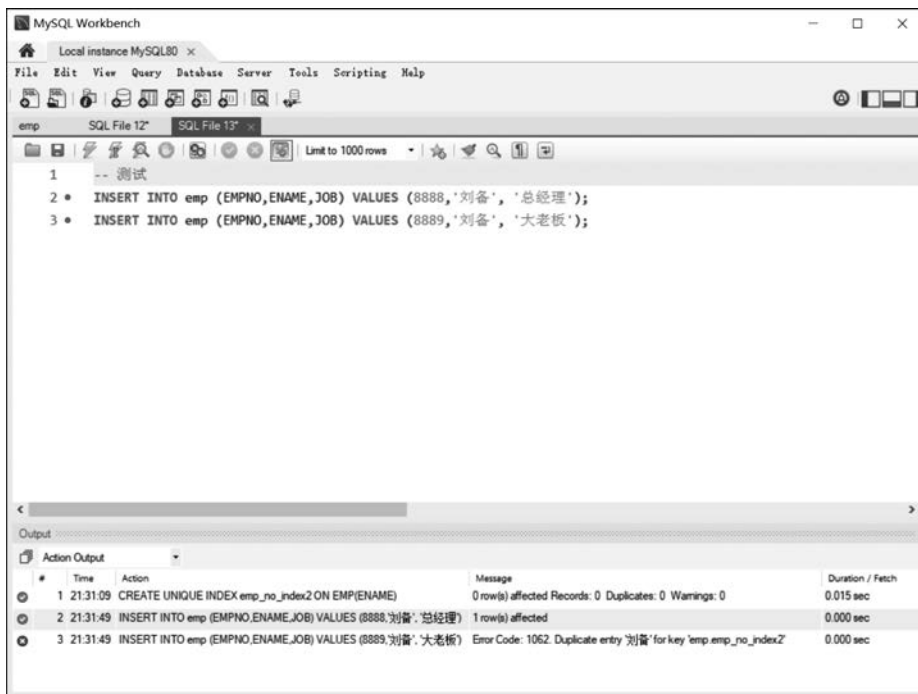


图 5-4 测试唯一索引结果



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,删除索引的语法结构如下:

```
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 测试,结果如图 5-5 所示,可见 emp_no_index2 索引已经被删除。

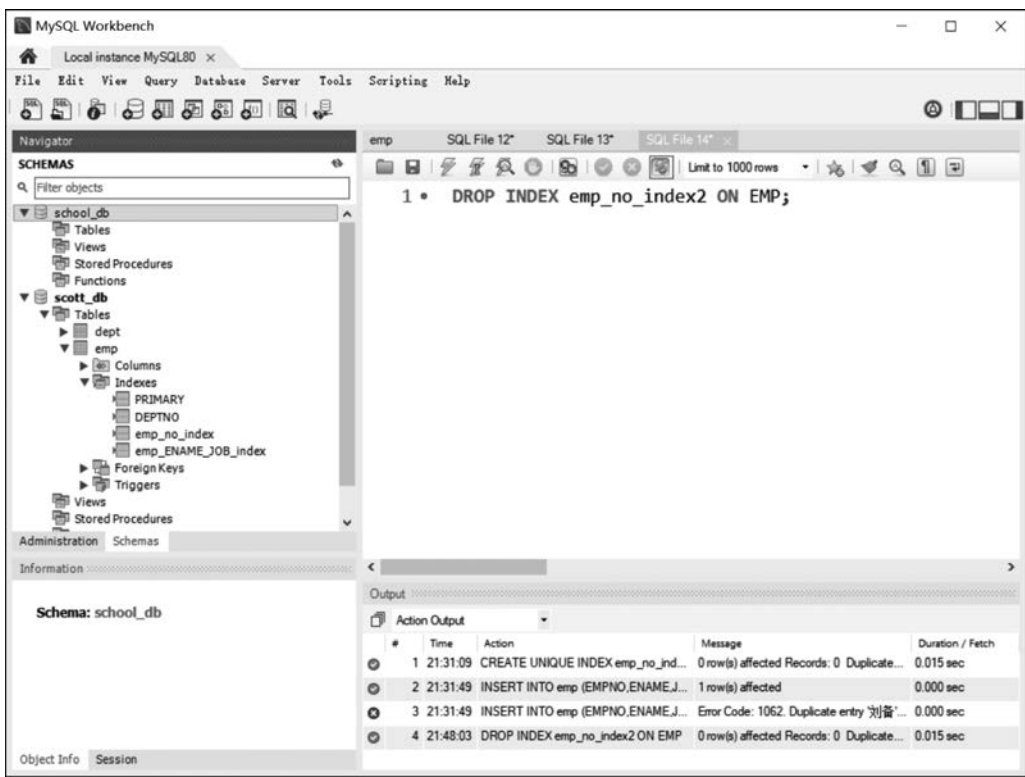


图 5-5 删除索引



微课视频

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,但是如何使用好索引并不是一件容易的事情。

在数据库中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,就可以提高数据库性能,但是并非如此。索引一方面可以提高查询速度;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。

在如下字段上创建索引,应该遵守的原则为:

- (1) 大量值字段,如果在存储大量值的字段上创建索引,索引会很好地发挥作用。
- (2) 在查询中经常使用字段,在查询 WHERE 子句中使用的字段上创建索引,它能提高查询速度。
- (3) 在表连接操作中经常使用字段,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 事实上很难为创建索引归纳出一般性的规则,因为在各个数据库中查询优化程序有很大区别。好的建议:如果在某一字段上查询性能速度缓慢,则可以创建索引;如果创建索引使得性能提高,则保留它,否则将其删除。

5.4 本章小结

本章重点介绍使用 SQL 语言创建索引,然后介绍删除索引,最后介绍使用索引的最佳实践。

5.5 同步练习

一、简述题

简述创建索引的意义。

二、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 |

三、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,索引会很好地发挥作用。()
4. 在查询 WHERE 子句中使用的字段上创建索引,它能提高查询速度。()