第 21 章　SQL 优化

在这些年的工作中，笔者发现大多数性能问题都与 SQL 语句有关；在参与过的一些项目中，笔者也发现大多数开发人员不太关心程序中执行的 SQL 语句。正因为这样，业界涌现出很多优秀的 SQL 语句审核平台，协助开发人员和 DBA 的工作。本章能够帮助大家加深对 MySQL 中索引和 Join 算法的理解，可以结合本书中的执行计划和案例章节来阅读本章内容。

21.1　SQL 优化基础概念

说到 SQL 优化，大家首先想到的就是创建索引，但创建索引需要了解相关基础概念。

1．索引

我们知道，MySQL 中的索引通常采用 B-Tree 结构，那么首先就要清楚 B-Tree 和 B+Tree 结构的区别，如图 21-1 所示。

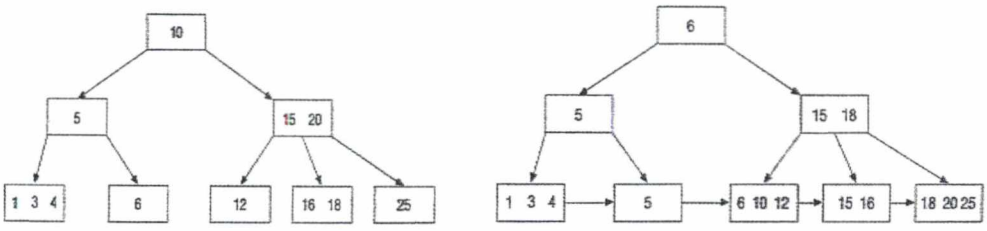

图 21-1

在 InnoDB 中索引是 B+Tree 结构的，在该结构中叶子节点包含了非叶子节点的所有数据，并且叶子节点之间会通过指针连接。

之所以采用 B+Tree 结构，是因为数据库中有>、<、between … and 这类范围查询语句，直接扫描叶子节点即可。

2．聚集索引（主键索引）

InnoDB 的所有的表都是索引组织表，主键与数据存放在一起。InnoDB 选择聚集索引遵循以下原则：

- 在创建表时，如果指定了主键，则将其作为聚集索引。
- 如果没有指定主键，则选择第一个 NOT NULL 的唯一索引作为聚集索引。
- 如果没有唯一索引，则内部会生成一个 6 字节的 rowid 作为主键。