第七章 MySQL高级特性

#### 一．分区表

1. 实现方式

对底层表的封装（包括索引）。

2. 分区的应用场景与优点

* 表非常大以至于无法全部都放在内存中，或者只在表的最后部分有热点数据，其它均是历史数据。
* 分区表的数据更容易维护。
* 分区表的数据可以存放在不同的物理设备上，从而高效的利用多个硬件设备。
* 分区表可以避免某些特殊的瓶颈，例如InnoDB的单个索引的互斥访问、ext3文件系统的inode锁竞争等。
* 可以备份和恢复独立的分区。

3. 分区的限制

* 一个表最多只能有1024个分区。
* 分区表达式必须是整数或者返回整数的表达式。
* 分区必须包含所有索引列。
* 分区表无法使用外键约束。

4. 对分区表的操作逻辑

无论是增删改查，都是首先打开并锁住所有的底层表，然后再判断要对哪个底层表进行操作。

5. 分区表的类型

（1）RANGE

（2）LIST

（3）HASH

（4）KEY

6. 分区的问题

* NULL会使得分区过滤无效

第一个分区存放分区条件为null的值和无效值

* 分区列和索引列不匹配
* 选择分区的成本可能很高
* 打开并锁住所有底层的成本可能很高
* 维护分区的成本可能很高

7. 查询优化

* 在分区表中，查询需要在where条件中带入分区列。
* select \* from sales\_by\_day where YEAR(day) = 2010不会过滤分区，即使表达式是分区函数。

优化：

select \* from sales\_by\_day where day between ‘2010-01-01’ and ‘2010-12-31’

类似于索引中的函数列会使得索引失效。

#### 二．分布式（XA）事务

1. 分布式事务

将存储引擎级别的ACID扩展到数据库层面，甚至是多个数据库之间。

2. 两阶段提交

（1） 第一阶段

XA事务需要有一个事务协调器来保证所有的事务参与者都完成了准备工作。

（2） 第二阶段

事务协调器收到所有的参与者都准备好的消息，就会告诉事务可以提交了。

注： MySQL在XA事务过程中扮演参与者而不是协调者。

3. 内部XA事务

由于MySQL的插件式架构导致需要在内部使用XA事务。

主要用于跨存储引擎的提交。

缺点：巨大的性能问题。

4. 外部XA事务

性能比内部XA事务消耗更大。

XA事务是一种数据库之间同步数据的方法。尽量避免使用，可以用消息队列代替事务的自动分发，还可以使用MySQL本身的复制机制。

#### 三．查询缓存

1. MySQL查询缓存保存查询返回的完整结果。当查询命中该缓存，MySQL会立刻返回结果，跳过了解析、优化和执行阶段。

2. 查询缓存会影响MySQL的扩展性，所以尽量关闭，使用其他的替代方案。