

# 作业七：MySQL Backup, Recovery & Partition

519021910913 黄喆敏

请你根据上课内容，针对你在E-BookStore项目中的数据库设计，详细回答下列问题：

1. 请你详细描述如何通过全量备份和增量备份来实现系统状态恢复。

**答：**全量备份包含了某个时间点的所有数据，而增量备份包含了上一次备份结束后被修改的文件的数据。因此，我们应将全量备份和增量备份**两者结合**，来实现系统状态恢复。

备份时，每隔一段较长的时间（例如6小时），我们进行一次全量备份；在之后的时间中，每隔一段较短的时间（例如1小时），我们进行一次增量备份，将数据库的变化写入bin-log文件中。由于增量备份的bin-log文件较大，因此进行全量备份后，应删去之前所有的bin-log文件。此外，为了保证备份的安全性，可以将备份放在不同的服务器上。

恢复时，我们先通过最近的一次全量备份，恢复到某个时间点数据库的状态。再使用bin-log文件，根据全量备份后的所有增量备份，实现系统状态恢复。

2. 请你根据MySQL缓存的工作原理，描述预取机制的优点。

**答：**MySQL的默认引擎InnoDB提供了预取机制。它通过**异步读取**缓存池里的多个page，加快这些page的访问速度。InnoDB提供了两种预取机制，分别为**线性预读** (Linear read-ahead) 和**随机预读** (Random read-ahead)。

线性预读为常用的预读方式。对于线性预读，每次连续读一个extent的threshold个page时（threshold默认为56），会触发下一个extent的page的预读。我们可以通过配置 `innodb_read_ahead_threshold` 参数，调整需要读的连续页面的数量，以触发预读操作。

对于随机预读，当同一个extent中的page在缓冲池中发现时，则会将该extent的剩余page读到缓冲池中。随机预读默认为OFF。

预取机制的好处在于，它是**使用后台线程异步完成的**。当我们需要对数据库进行**范围查询**，或者**全表扫描**时，此时InnoDB会触发线性预读机制，每次提前异步读取下一个extent的page，**提高读取的吞吐量**。

在我的E-Bookstore中，有很多情况下需要进行范围搜索。例如**分页显示书籍**，每次获取一页的内容，采用预取机制可以大大加快获取下一页内容的速度；管理员查看所有订单时，需要进行全表扫描。此时采用预取机制，也可加快速度，提高吞吐量。

3. 请你按照你的理解，阐述Partition机制有什么好处？

**答：**Partition机制的好处如下所示：

- 分区文件可以存储在不同的物理磁盘上，因此使用分区可以在一张表中**存储比一个硬盘或者文件系统分区上的数据更多的数据**。
- 对于**无用的过时数据**，我们可以通过删除仅存在那些数据的分区，很容易地删除那些数据。例如我们需要保留三个月的有效数据时，可以直接清除所在的分区；对于**新数据**，我们可以为它们添加新的分区，来存储数据。
- **可以很大程度地优化一些查询的速度**。因为在where中包含分区条件时，我们可以只扫描一个或多个分区，提高查询效率。例如课上所举的例子，我们可以将本科生和研究生分为两个分区。查询本科生或者研究生的信息时，只需要扫描它们所在的分区即可，不需要扫描所有分区，以此提高效率。

- 我们可以将不同的分区在不同的物理磁盘上，这样查询时可以减轻单个磁盘的压力，提高查询吞吐量。

在我的E-Bookstore中，对于order表，可以根据**订单时间**进行分区。用户根据时间查询订单时，只需要在所在的分区查询即可。添加订单时，直接添加至对应的分区；对于book表，已经记录了每个书的分类，例如文学类、科幻类等。因此，我们可以**根据书籍的分类进行分区**，用户在根据书籍类型搜索书籍时，直接搜索对应的分区，可以大大提高搜索的效率。

#### 4. 如果数据文件在一台机器上有足够的存储空间存储，是否还需要进行Partition?

**答：**我认为还是有必要进行分区。因为分区机制的好处不仅仅是可以存储更多的数据，更重要的是，**分区可以提高查询效率，且分区表的数据更容易维护**。以E-Bookstore为例，其数据文件较小，在本机有足够的存储空间存储。但是通过分区，我们可以优化查询效率，提高吞吐量；在统计订单数等涉及到**聚合函数的查询**时，我们可以在不同的分区上并行处理，最终汇总所有分区的结果，加快速度；想要批量删除某一类的书时，我们可以使用清除整个分区的方式，更加安全、简便。

## Reference

[https://dev.mysql.com/doc/refman/8.0/en/innodb-performance-read\\_ahead.html](https://dev.mysql.com/doc/refman/8.0/en/innodb-performance-read_ahead.html)

<https://dev.mysql.com/doc/refman/8.0/en/partitioning.html>