

第一章：mysql存储引擎

1.innodb存储引擎

- 支持事务；行锁设计；支持外键；支持非锁定读。
- 4种事务隔离级别，默认为repeatable，同时使用next-key lock的算法避免幻读。
- 支持多版本并发控制，提高性能。
- 索引支持 btree索引，hash索引，以及全文检索功能。

2.MyISAM存储引擎

- 不支持事务；表锁实现；支持全文检索；存储引擎由数据文件和索引文件组成。

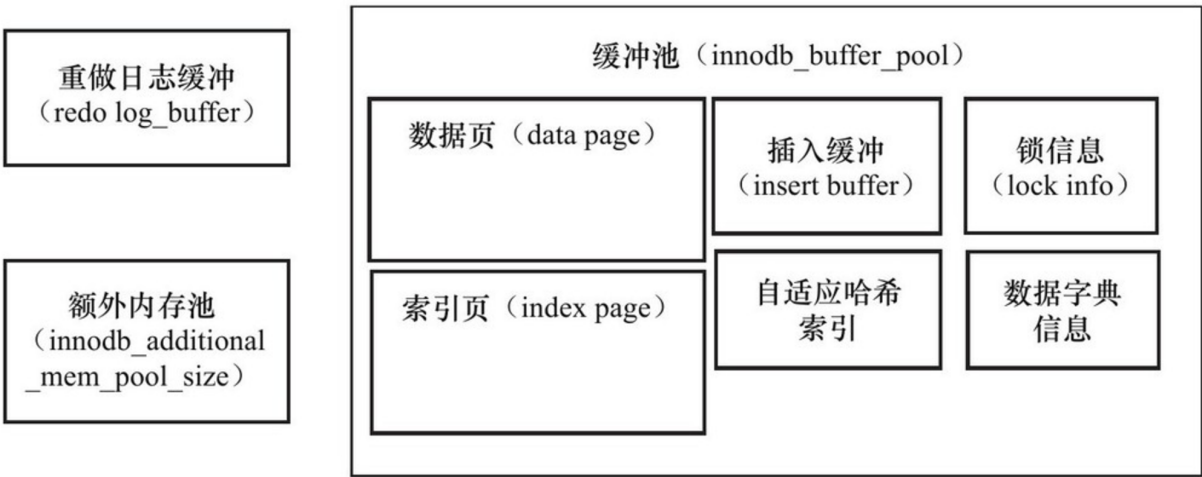
第二章：innodb存储引擎

- 1.概述：MySQL第一个支持完整的acid事物的存储引擎，特点是行级锁，多版本控制，支持外键，提供一致性非锁定读，并且更高效的利用cpu。

2.体系结构：

- 后台线程 操作innodb存储引擎缓冲池 与文件进行更新。

3.内存池结构：



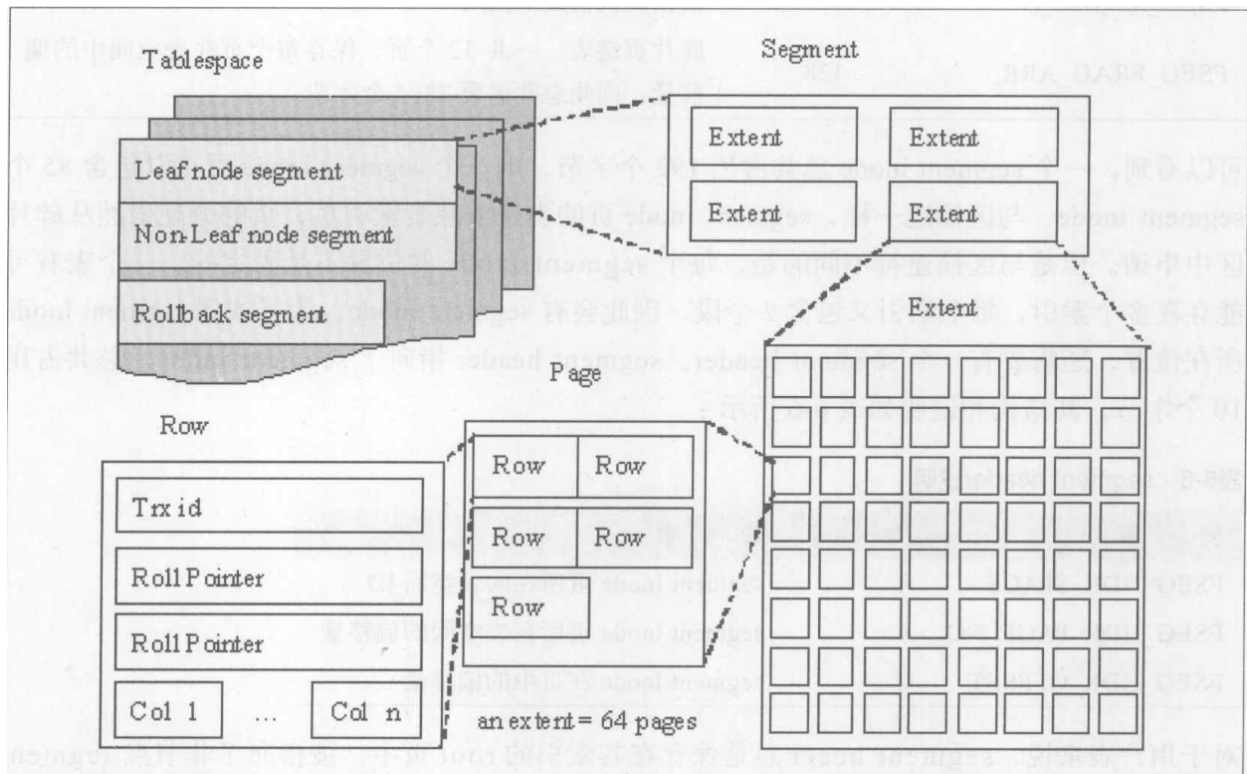
- 4.事务持久性的要求的保障：采用的write ahead log策略.即，在事务提交时，先对重做日志进行更新，再修改页数据.当发生宕机时，可通过重做日志回复数据。

5.innodb的关键特性：

- ①插入缓存（提高性能）：对于非聚集索引的插入或者更新操作并不是马上进行更新到索引页，而是先判断缓冲池中是否有该数据，有则直接更改，没有则先插入到缓冲区，再以一定的频率刷新。
- ②两次写（提高可靠性）；对脏页进行刷新时，先写到doublewriter buffer缓冲中，在刷新到磁盘中。

第四章：表

1.innodb表结构



2.约束

数据完整性：①实体完整性，通过主键或unique约束。②数据完整性，通过触发器完成。③域完整性，通过not null，int 等来约束。

约束和索引的区别：约束是逻辑概念，索引是一种数据结构，索引是实现约束的一种方式。

第五章：索引

btree索引和hash索引，以及全文检索

1.hash索引是自适应的，根据表的使用情况自动生成，不能人为干预。

2.btree索引

2.1 btree索引并不是根据给定键值找到具体的行，而是查到行所在的页，然后把页加载到内存，然后进行查找。

2.2 btree是高扇叶的平衡树，插入和删除都需要大量操作使得其平衡，所以尽量减少该操作。

2.3 聚集索引和辅助索引都是b树结构，但是聚集索引的叶子节点存储的是具体的行信息，而辅助索引存储的是聚集索引的键值。

2.4 聚集索引不是物理上连续的，而是逻辑上连续的；聚集索引对主键的排序查找和范围查找比较快。

2.5 何时使用索引：字段范围广，无重复。

2.6 全文检索：使用倒排索引来实现。他在辅助表中存储了单词和单词所在的位置。

第六章：锁

锁是为了支持对共享资源的并发访问，提供数据的完整性和一致性。

1.lock的对象是事物，用来锁定数据库中的对象，如行，页，表。一般lock锁在commit或rollback之后释放（不同事务级别释放的时间可能不同）

2.锁：行级别有共享锁（S）和排他锁（X），数据库会对页，表上意向锁。（IS,IX）

如果相对行上排他锁，必需对其所在的表，页上意向排他锁（IX）

3.一致性非锁定读：

innodb通过多版本控制的方法，来并行的读取。即，如果当前行在执行update或delete操作，读取操作不会等待，而是去读该行的快照数据，就是历史版本，因为不会有操作会对历史版本进行修改，所以，可以并发读取，不上锁。

事物级别 read committed：会读取最新的一份快照数据。

事务级别 repeatable read：读取事务开始时的行数据版本

4.一致性锁定读：

默认级别repeatable read下，使用一致性非锁定读，但由于一些需求，需要锁定时，可以使用加锁来进行读取，保证数据逻辑的一致性。

语句：①select for update; ②select lock in share model

5.锁的算法

5.1 record lock：单行记录上锁

5.2 cap lock：锁定一个范围，但是不包含改行

5.3 next-key lock：就是cap lock + record lock的组合，锁定一个范围，并且包括该行。

幻读现象：同一事务下，连续两次执行同一查询语句，结果不一致的现象。

解决办法：innodb默认事务隔离级别为 repeatable，使用next-key算法，可以避免幻读现象。而在read committed级别下，行锁算法为 record，所以会出现幻读现象。

6.锁带来的问题

锁实现满足的事务隔离的要求。使得事物并发工作。但是锁会带来以下问题：①脏读 ②不可重复读 ③丢失更新

6.1 脏读：事务隔离级别为 read uncommitted

脏读数据是指：一个事物读取到了，另一个事物修改但是还没有提交的数据，

6.2 不可重复读：事物隔离级别为 read committed

指一个事物读取到了另一个事物修改并且提交的数据，但是在该事物开始前，没有任何事务对该数据进行修改。即在该事务过程中，发生的事情。

6.3 数据更新：一个事物的操作，覆盖掉了另一个事物的操作。

7. 死锁：两个及以上事务在执行中，相互竞争资源锁而造成的相互等待的现象。

解决策略：①获取锁时采用超时获取，如果获取超时，则进行回滚等操作。

②主动检测死锁

8. innodb不存在锁升级的说法，因为其不是根据每个行记录产生锁，而是根据每个事务访问的每个页进行管理，对页进行加锁。

第七章：事务

1.目的：事物会把数据库从一种一致状态转换成另外一种一致性的状态。

2.事务的ACID特性：A原子性 (atomicity) C一致性 (consistency)

I隔离性 (isolation) D持久性 (durability)

解释：①原子性：事务要么都完成，要么都不完成

②一致性：数据从一种一致性状态到另一种一致性状态

③隔离性：事务提交前，其他事务对其不可见（事务级别不同而不同）

④持久性：事务一旦提交，结果就是永久性的。

3.事物的分类：扁平事务，带有保存点的扁平事务，分布式事物.....

4.事务的实现：①隔离性用锁来实现 ②原子性和持久性用redo log 重做日志实现

③一致性由undo log实现

redo log：记录的是修改操作。如 insert into 等sql语句

undo log：记录的是每一行的具体内容，用于回滚，也是对多版本控制的支持。

5.事物的隔离级别：①read uncommitted ②read committed ③repeatable read

④serializable

6.分布式事务必需用serializable