1. 建立组合索引（bcd）

共8条数据

20210825212209

select \* from user where b>1 ：不走索引，避免了多次回表扫描

select \* from user where b>6 ：走索引，避免了多次回表扫描

覆盖索引

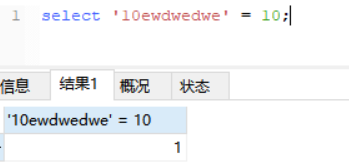
select b from user where b>1 ：走索引，因为不需要回表

select b from user ：走组合索引，因为组合索引的叶子节点存储的内容比较少，磁盘IO相对于全表扫描少的多，检索速度快

1. mysql类型转换

实际类型和插入的类型不匹配，字符和数字会转换

字符->数字:字符最前面的数字



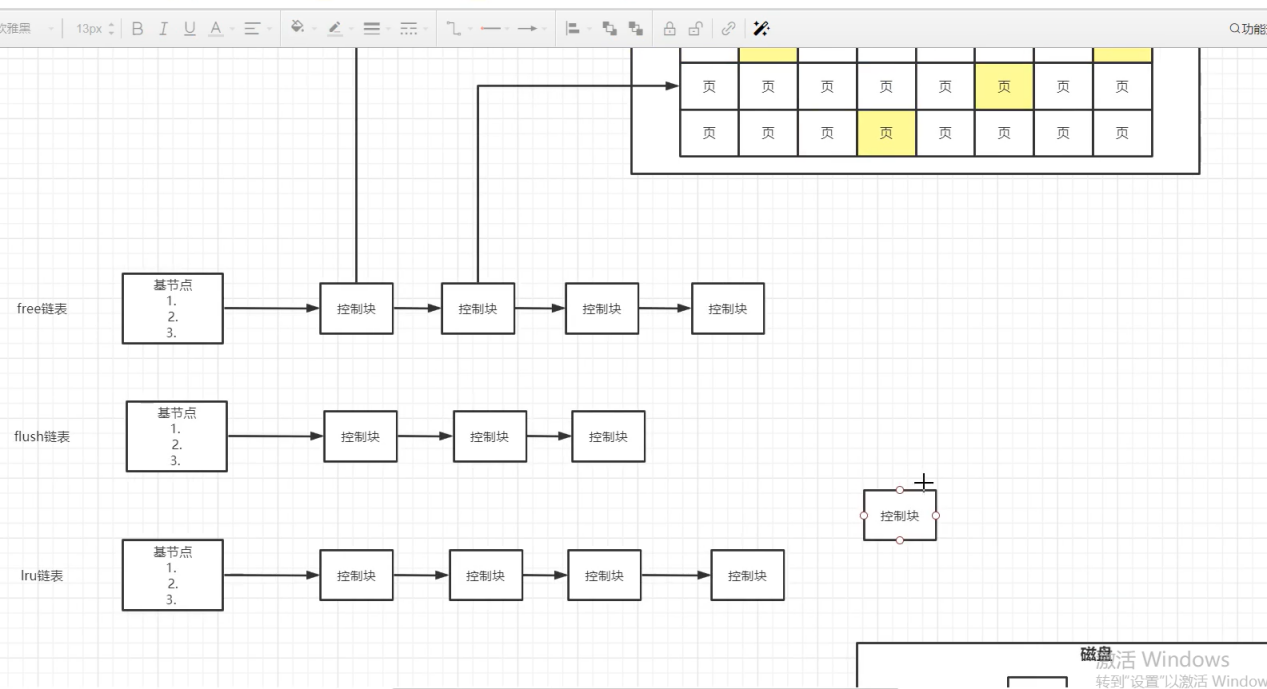
1. 执行流程

Buffer pool（128M）

空闲区域管理：free链表

脏页（update）：flush链表，IO线程会定时刷盘

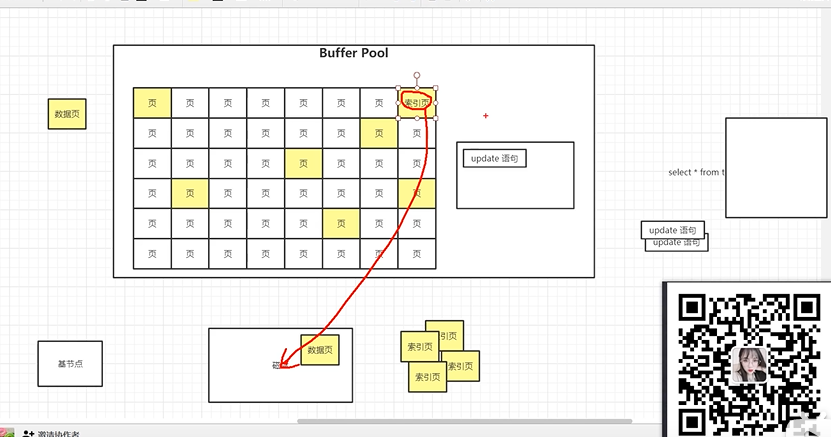
Bufferpool满了淘汰机制：LRU链表



（1）从磁盘中读取某一页到内存中

Update：更新数据页、索引页

索引页修改：后续select的时候整合索引页和chang buffer



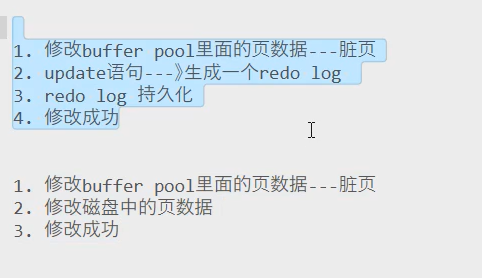
查询：如果可以使用索引，优化器会选择对应的索引进行查询，然后从磁盘读取N次索引页，通过索引页定位到数据页（回表、覆盖索引），然后从磁盘中读取数据页到bufffer pool中。更新内存数据，记录log；IO线程会定时将Buffer pool的数据持久化到磁盘中。

1. Redo log（用于本机数据恢复）

记录的是物理位置的修改

（1）为什么有redo log？

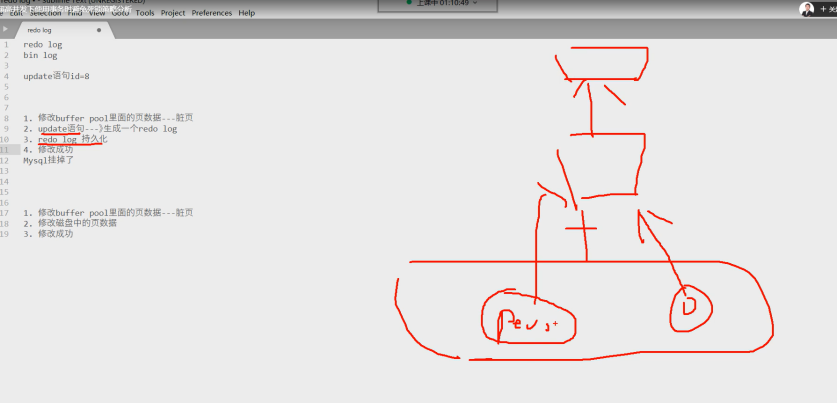
Update更新数据的时候，先修改内存的数据，如果还未来得及持久化（IO线程未来会将数据持久化到磁盘），mysql宕机了，就会出现数据丢失的情况（只改了内存数据，没改磁盘数据）



第二种方式肯定比第一种慢，因为写到磁盘需要进行磁盘寻道操作。

1. mysql重启数据恢复

会整合redo log和磁盘数据给用户返回正确数据

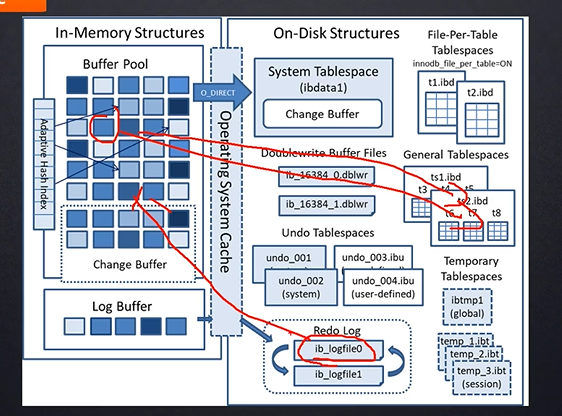


1. redo log file满了，触发数据刷盘（Checkpoint）

redo log file满了，查看当前buffer pool中有log file对应的数据页，如果有则将数据页持久化到磁盘，此时log file文件才可以进行写入。

因此，如果log file比较小，就会经常持久化，性能就下降。如果log file过大，重启恢复数据，也会变慢。

调优点：log file



1. redo log何时持久化

Commit提交事务的时候，因为如果rollback就完全没必要持久化。



1. Binlog（用于主从复制）

Sql语句

1. 有binlog为什么还要有redolog？

Binlog是Mysql的Server层，redolog是存储引擎Innodb的。

Binlog比较慢，Innodb自己设计redolog

Redolog相对于binlog（记录sql语句）恢复数据更快，因为redolog记录物理位置的修改，恢复很快；而binlog记录的是sql语句，恢复的时候得执行sql语句。

但是，redolog不能用于主从复制

1. Undo log

反向sql语句

用于rollback回滚，把buffer pool的数据还原。