3.11 数据库锁

数据库一般分为俩类：悲观锁，乐观锁。

悲观锁，从数据开始更改时就将数据锁住，知道更改完成才释放。

乐观锁，直到修改完成准备提交所做的修改到数据库的时候才会将数据锁住。完成更改后释放。

悲观锁一般就是我们通常说的数据库锁机制。

《==========================================================================================》

3.12 分区分表

1.什么是mysql分表和分区

分表：表面意思就是，把一张表分成N多个小表

分区：分区就是把一张表的数据分成N多个区块，这些区块可以在同一个磁盘上，也可以在不同的磁盘上

《==========================================================================================》

2.mysql分表和分区有什么区别呢

实现方式上

Mysql的分表是真正的分表，一张表分成很多表后，每一个小表都是完整的一张表。

分区不一样，一张大表进行分区后，他还是一张表，不会变成第二张表，但是他的存储区块变多了。

《==========================================================================================》

3.提高性能上

分表后，单表的并发能力提高了，磁盘I/O性能也提高了。

《==========================================================================================》

4.分区类型

Range分区、list分区、hash分区、key分区。

《==========================================================================================》

3.13 应该使用哪一种方式来实施数据库分库分表，这要颈 看数据库中数据量的瓶颈 所在，并综合项目的业务类型进行考虑。

如果数据库是因为表太多而造成海量数据，并且项目的各项业务逻辑划分清晰、低耦合，那么规则简单明了、人员实施的垂直切分必是首选。

而如果数据库中的表并不多，但单表的数据量很大、或数据热度很高，这种情况之下就应该选择水平切分。

单库多表：

随着用户数量的增加，user表的数据量会越来越大，当数据达到一定程度的时候对user表的查询会渐渐的变慢，从而影响整个DB的性能。

多库多表：

随着数据量增加也许单台DB的存储空间不够，随着查询量的增加单台数据库服务器已经没办法支持。

《==========================================================================================》

3.14 mysql 读写分离

在实际的应用中，绝大部分情况都是读远大于写。Mysql提供了读写分离的机制，所有的写操作都必须对应到Master，读操作可以在Master和Slave和机器上进行，Slave与Master的结构完全一样，一个Master可以有多个Slave。

《==========================================================================================》

3.15 MySQL SQL 语句优化

1.对查询进行优化，应尽量避免全表扫描。

2.应尽量避免在where子句中使用!=或<>操作符。

3.尽量避免在where子句中对字段进行null值判断。

4.应尽量避免在where子句中使用or来连接条件

5.in和not in也要慎用。

《==========================================================================================》

3.16 数据库优化方案整理

1.优化分片键

（1）不要使用分区，分区性能很低，难以维护

（2）不要使用主从数据库，读写分离有了缓存就再也不需要分主从了，毕竟延迟问题无法解决。

2.优化索引

（1）索引列务必重复度低

（2）使用索引就不能用OR查询，否则索引不起作用

（3）使用索引，like模糊查询不能以%开头

优化缓存

分离冷热数据，对于大内存访问频率低的数据适当分离

提高缓存命中率，命名问题