

数据库优化总结

1、数据库连接池 C3PO DBCP DRUID Proxool

2、sql优化

3、数据库表设计

4、主从同步，读写分离

数据库优化我们从以下几个方面处理：

(1) 使用数据库连接，初始化在连接池中创建一些连接，避免了每次连接数据库都创建和销毁连接，造成资源的浪费。

(2) 缓存：避免与数据库的交互，减轻了服务器的压力，redis或者oscache或者memcache

(3) 表结构的设计

- 三范式

- 第一范式(确保每列保持原子性)

第一范式是最基本的范式。如果数据库表中的**所有字段值都是不可分解的原子值**，就说明该数据库表满足了第一范式。

- 第二范式(确保表中的每列都和主键相关)

第二范式在第一范式的基础之上更进一层。第二范式需要确保数据库表中的每一列都和主键相关，而不能只与主键的某一部分相关（主要针对联合主键而言）。**也就是说在一个数据库表中，一个表中只能保存一种数据，不可以把多种数据保存在同一张数据库表中。**

- 第三范式(确保每列都和主键列直接相关,而不是间接相关)

第三范式需要确保数据表中的**每一列数据都和主键直接相关，而不能间接相关。**

- 索引

经常查询，不经常修改的数据上面创建索引，但是后期要维护好

(4) sql优化

1. sql的执行顺序，执行顺序，from 表名， where条件为从后到前执行，后面的字段最大的过滤数据，再执行group by分组，然后执行having条件，再执行order by条件，然后执行select *，例如：`select * from t_user where 性别 = "女" and province = "山东"`

2. 尽量避免联查，因为联查，查询数据会比单表查询，多更多数据，例如：用户表和角色表

3. 适度冗余。例如：当用户下订单时，需要给用户发手机短信，这时候需要用户的手机号码，如果使用订单表 and 用户表联查，性能会降低，所以在订单表中创建的时候多创建一个用户ID和用户手机号的两个字段，

4. 范围字段适度。例如：id字段长度（36），name字段长度（100），pwd字段长度（50）

5. 适度的创建索引。例如：数据一般在5~6万条的时候创建索引，`CREATE [UNIQUE|CLUSTERED] INDEX INDEX_NAME ON TABLE_NAME (PROPERTY_NAME)`

其中UNIQUE和CLUSTERED为可选项，分别是建立**唯一索引**和聚簇索引，具体解释为：

UNIQUE:表示此索引的每一个索引值只对应唯一的数据。

CLUSTERED:表示要建立的索引时聚簇索引，即索引项的顺序与表中记录的物理顺序一致的索引组织。

6. 避免表的字段过多（50个字段以内）
7. 根据查询和更新的比例，采用读写分离
8. 对数据表采用横切割，纵切割。例如：产品表（总表），分割成多个表（电子类产品表，衣服产品表）

（5）主从同步，读写分离，双击热备