1. 数据库查询语句怎样写效率更高

1、程序中，

保证在实现功能的基础上，尽量减少对数据库的访问次数；在查询时，不要过多地使用通配符；不要在应用中使用数据库游标

1. 避免使用不兼容的数据类型
2. 尽量避免在WHERE子句中对字段进行函数或表达式操作，这将导致引擎 放弃使用索引而进行全表扫描。
3. 充分利用连接条件，在某种情况下，两个表之间可能不只一个的连接条件
4. 使用视图加速查询
5. 能够用BETWEEN的就不要用IN；DISTINCT的就不用GROUP BY
6. 使用sql语句实现图的某一顶点可达的该图的其他顶点的查找

3.数据库完整性措施；

完整性包括 域完整性、实体完整性、参照完整性、用户定义的完整性

SQL Server 提供了一些工具来帮助用户实现数据完整性，其中最主要的是：规则（Rule）、缺省值（Default）、约束（Constraint） 和触发器（Trigger）。

1. 数据的一致性

数据库一致性（Database Consistency）是指事务执行的结果必须是使数据库从一个一致性状态变到另一个一致性状态。

（一）传统的事务控制法

在分布式数据库系统中,“事务”是一系列不可分割的操作序列, 将数据库从一个一致性状态转变到另一个一致性状态, 由于全局事务与局部事务存在并发执行, 可能会造成数据副本不一致。传统的事务控制法通过分布式两段锁协议(2PL 协议) 来保证全局事务与局部事务执行的可串行性, 即可保证事务的一致性调度, 以及通过分布式两段提交协议(2PC 协议) 来同步更新各副本数据 。这对数据操作涉及的记录不多、事务保持时间不长的分布式数据库系统有效。而在一些数据量大、用户对数据的操作范围大的情况下, 事务保持时间长, 若采用2PL 协议, 则会严重地影响事务并发程度, 不能满足实际需要, 同时, 2PC 协议或3PC 协议的方法在网上通信量很大, 而由于网络速度有限, 因此, 会使用户陷入长时间的不可忍受的等待状态, 或遇到频繁的事务失败, 重新启动事务太多, 造成应用程序运行效率低下。

（二）扩展事务控制法

在分布式环境下, 2PC 协议是实现事务原子性的一个主要原则, 然而大多数原有的数据库系统并不支持这一协议, 扩展事务控制法则不依赖于局部数据库系统是否支持2PC 协议, 而是对全局子事务和局部事务进行扩展, 并为各个局部数据库引进了锁表和日志表, 它在局部数据库和控制表之上执行扩展事务, 它们负责维护全局可串行性和实际事务的恢复。通过扩展事务来检测锁, 以避免冲突发生和进行恢复 。该方法在一定的基础上维护了局部事务的自治性, 但它增加了事务的负担而降低了事务本身的执行效率。

（三）复制控制法

分为同步复制控制法和异步复制控制法

利用同步复制使得各数据副本保持紧一致性, 即任何时刻不同站点的副本的值总是一致的。这样, 全局事务管理系统要明确数据的各个副本所在的站点, 当事务对某一数据副本进行了更新时, 全局事务管理系统必须向其他站点发送强制用户修改数据副本的消息以保持数据各副本一致。但是对于出现故障的站点, 则必须等待其恢复以后再进行复制, 因此数据并非严格一致。由于对数据的更新操作频繁, 从而向系统发送的复制消息也频繁, 系统通信量大, 降低了系统处理速度。而且对于很多数据库系统来说, 它并不要求严格的数据一致, 如空间数据库一般允许访问时间相对滞后的数据, 采用同步复制控制并不实用。

利用异步复制使得各数据副本保持松一致性, 即数据的更新与其各副本所做的修改复制有一段时间间隔, 在某一时刻不同站点的副本的值可能是不一致的, 数据暂时不同步。它主要解决两个问题,即如何捕捉到更新数据和什么时候复制。一般采用基于事务日志方法和基于触发器方法捕捉系统更新操作, 大多数数据库系统一般采用事务日志来捕捉数据更新, 如Sybase , Informix , Microsoft SQLServer 等。但由于事务日志格式可能不相同, 设计通用的数据库日志读取程序相当困难, 因而一些系统采用触发器方法。对于触发器法, 一旦被跟踪的对象被更新, 会触发相关事件发生, 记录数据源与数据更新的内容。对于被触发器所捕捉的更新数据, 由复制服务器对各副本进行更新, 复制服务器需要保证被复制的数据是最后的更新结果。

（四）消息队列法

消息队列是进程间通信的一种机制, 两个或多个进程间通过访问共用的系统消息队列来交换信息, 这里将消息队列的概念扩展到位于分布式环境下的不同站点间的进程间通信[5 ] , 由消息管理机构在Internet 上实现可靠的消息传送, 其中涉及的消息队列有发送队列、接收队列、应答队列和管理队列。

应用程序在本地数据库上完成数据更新后, 将更新信息以及本地应答队列和管理队列的地址存放到消息中, 发送到远端接收队列中, 消息交给消息管理机构后首先进入本地发送队列等待发送, 如一切正常, 消息将送到应用程序指定的远端接收队列中, 同时, 一个后台处理程序一直在监视着自己的接收队列, 一旦有消息到达, 它将读消息, 并对本地数据库实施消息中所描述的更新操作, 如果更新成功, 则处理结束, 否则依应答队列的地址信息发送出错消息, 另一个后台处理程序一直在监视应答队列, 根据收到的错误信息, 它将在本地数据库中试图撤消相应的更新操作。

该方法提供的是数据库间异步更新, 而不是同步更新, 这样可能读到的不是最新数据, 所以它不适合要求实时数据同步的分布式数据库系统。

1. 关系数据库的定义

在一个给定的应用领域中，所有实体及实体之间联系的集合构成一个关系数据库。

6.如果一个库非常大，以至于必须对它进行“分割”（或者是别的词，我记不清了），你会怎么做？

分布式数据库的基本思想是将原来集中式数据库中的数据分散存储到多个通过网络连接的数据存储节点上，以获取更大的存储容量和更高的并发访问量。