

MySQL数据库引擎

1.常用引擎:

MyISAM和InnoDB

2.二者区别:

- MyISAM是非事务安全型的，而InnoDB是事务安全型的。
- MyISAM锁的粒度是表级，而InnoDB支持行级锁定。
- MyISAM支持全文索引，而InnoDB不支持全文索引
- MyISAM表是保存成文件形式的，在跨平台的数据转移中使用MyISAM存储会省去不少的麻烦。
- InnoDB表比MyISAM表更安全，可以保证数据不丢失的情况下，切换非事务表到事务表

3.应用场景

- MyISAM 管理非事务表，它提供高速存储和检索，以及全文搜索能力。如果应用中需要执行大量的SELECT查询，那么MyISAM是更好的选择。
- InnoDB用于事务处理应用程序，具有众多特性，包括ACID事务支持。如果应用中需要执行大量的INSERT或UPDATE操作，则应该使用InnoDB,这样可以提高多用户并发操作的性能。

4.存储引擎索引实现

- MyISAM索引实现：MyISAM索引使用了B+Tree作为索引结构，叶子结点的data域存放的是数据记录的地址。MyISAM中索引检索的算法为首先按照B+Tree搜索算法搜索索引，如果指定的Key存在，则取出其data域的值，然后以data域的值作为地址，读取相应数据记录。主索引和辅助索引的存储结构没有任何区别。
- InnoDB索引实现：虽然InnoDB也使用B+Tree作为索引结构，但具体实现方式却与MyISAM截然不同。第一个重大区别是InnoDB的数据文

件本身就是索引文件。从上文知道，MyISAM索引文件和数据文件是分离的，索引文件仅保存数据记录的地址。而在InnoDB中，表数据文件本身就是按B+Tree组织的一个索引结构，这棵树的叶节点data域保存了完整的数据记录。这种索引叫做聚集索引。因为InnoDB的数据文件本身要按主键聚集，所以InnoDB要求表必须有主键（唯一且非空）（MyISAM可以没有），如果没有显式指定，则MySQL系统会自动选择一个可以唯一标识数据。第二个与MyISAM索引的不同是InnoDB的辅助索引data域存储相应记录主键的值而不是地址。换句话说，InnoDB的所有辅助索引都引用主键作为data域。

数据库范式

码：设 K 为某表中的一个属性或属性组，若除 K 之外的所有属性都完全函数依赖于 K（这个“完全”不要漏了），那么我们称 K 为候选码。在实际中我们通常可以理解为：假如当 K 确定的情况下，该表除 K 之外的所有属性的值也就随之确定，那么 K 就是候选码，简称为码。

主属性：包含在任何一个候选码中的属性为主属性。

- 1NF:属性不可分，如“电话号码”这个属性可以继续被分割为“办公电话”、“手机号码”等属性，在第一范式的语义下不应该被作为单独的一列出现。
- 2NF:符合1NF，非主属性完全依赖于码。即一个数据库表中只能保存一种数据，不能把多种数据保存到同一张数据表中，主要针对联合主键来说。（多对多）
- 3NF:符合2NF,非主属性不传递依赖于码。即每一列数据和主键直接相关，而不能间接相关。（一对多）
- BCNF:符合3NF,不存在任何属性完全依赖于非码的任何一组属性。BCNF一定是3NF,3NF不一定是BCNF

满足范式要求的数据库设计是结构清晰的，同时可避免数据冗余和操作异常。这并意味着不符合范式要求的设计一定是错误的，在数据库表中存在1：1或1：N关系这种较特殊的情况下，合并导致的不符合范式要求反而是合理的。

