
1、超键、候选键、主键、外键

超键：在关系中能唯一标识元组的属性集称为关系模式的超键。一个属性可以作为超键，多个属性组合在一起也可以作为一个超键。超键包含候选键和主键。

候选键：是最小超键，即没有冗余元素的超键。

主键：数据库表中对储存数据对象予以唯一和完整标识的数据列或属性的组合。一个数据列只能有一个主键，且主键的取值不能缺失，即不能为空值（Null）。

外键：在一个表中存在的另一个表的主键称此表的外键。

2、什么是事务？什么是锁？

事务：就是被绑定在一起作为一个逻辑工作单元的 SQL 语句分组，如果任何一个语句操作失败那么整个操作就被失败，以后操作就会回滚到操作前状态，或者是上有个节点。为了确保要么执行，要么不执行，就可以使用事务。要将有组语句作为事务考虑，就需要通过 ACID 测试，即原子性，一致性，隔离性和持久性。

锁：在所有的 DBMS 中，锁是实现事务的关键，锁可以保证事务的完整性和并发性。与现实生活中锁一样，它可以使某些数据的拥有者，在某段时间内不能使用某些数据或数据结构。当然锁还分级别的。

3、数据库事务的四个特性及含义

原子性：整个事务中的所有操作，要么全部完成，要么全部不完成，不可能停滞在中间某个环节。事务在执行过程中发生错误，会被回滚（Rollback）到事务开始前的状态，就像这个事务从来没有执行过一样。

一致性：在事务开始之前和事务结束以后，数据库的完整性约束没有被破坏。

隔离性：隔离状态执行事务，使它们好像是系统在给定时间内执行的唯一操作。如果有两个事务，运行在相同的时间内，执行相同的功能，事务的隔离性将确保每一事务在系统中认为只有该事务在使用系统。这种属性有时称为串行化，为了防止事务操作间的混淆，必须串行化或序列化请求，使得在同一时间仅有一个请求用于同一数据。

持久性：在事务完成以后，该事务对数据库所作的更改便持久的保存在数据库之中，并不会被回滚。

4、什么是视图？

视图是一种虚拟的表，具有和物理表相同的功能。可以对视图进行增，改，查，操作，视图通常是有一个表或者多个表的行或列的子集。对视图的修改不影响基本表。它使得我们获取数据更容易，相比多表查询。

如下两种场景一般会使用到视图：

（1）不希望访问者获取整个表的信息，只暴露部分字段给访问者，所以就建一个虚表，就是视图。

（2）查询的数据来源于不同的表，而查询者希望以统一的方式查询，这样也可以建立一个视图，把多个表查询结果联合起来，查询者只需要直接从视图中获取数据，不必考虑数据来源于不同表所带来的差异。

注：这个视图是在数据库中创建的 而不是用代码创建的。

5、触发器的作用？

触发器是一种特殊的存储过程，主要是通过事件来触发而被执行的。它可以强化约束，来维护数据的完整性和一致性，可以跟踪数据库内的操作从而不允许未经许可的更新和变化。可以级联运算。如，某表上的触发器上包含对另一个表的数据操作，而该操作又会导致该表触发器被触发。

6、维护数据库的完整性和一致性，你喜欢用触发器还是自写业务逻辑？为什么？

尽可能使用约束，如 check, 主键，外键，非空字段等来约束，这样做效率最高，也最方便。其次是使用触发器，这种方法可以保证，无论什么业务系统访问数据库都可以保证数据的完整性和一致性。最后考虑的是自写业务逻辑，但这样做麻烦，编程复杂，效率低下。

7、索引的作用？和它的优点缺点是什么？

数据库索引，是数据库管理系统中一个排序的数据结构，以协助快速查询、更新数据库表中数据。索引的实现通常使用 B 树及其变种 B+ 树。

在数据之外，数据库系统还维护着满足特定查找算法的数据结构，这些数据结构以某种方式引用（指向）数据，这样就可以在这些数据结构上实现高级查找算法。这种数据结构，就是索引。

为表设置索引要付出代价的：一是增加了数据库的存储空间，二是在插入和修改数据时要花费较多的时间(因为索引也要随之变动)。

创建索引可以大大提高系统的性能（优点）：

第一，通过创建唯一性索引，可以保证数据库表中每一行数据的唯一性。

第二，可以大大加快数据的检索速度，这也是创建索引的最主要的原因。

第三，可以加速表和表之间的连接，特别是在实现数据的参考完整性方面特别有意义。

第四，在使用分组和排序子句进行数据检索时，同样可以显著减少查询中分组和排序的时间。

第五，通过使用索引，可以在查询的过程中，使用优化隐藏器，提高系统的性能。

也许会有人要问：增加索引有如此多的优点，为什么不对表中的每一个列创建一个索引呢？

增加索引也有许多不利的方面：

第一，创建索引和维护索引要耗费时间，这种时间随着数据量的增加而增加。

第二，索引需要占物理空间，除了数据表占数据空间之外，每一个索引还要占一定的物理空间，如果要建立聚簇索引，那么需要的空间就会更大。

第三，当对表中的数据进行增加、删除和修改的时候，索引也要动态的维护，这样就降低了数据的维护速度。

索引是建立在数据库表中的某些列的上面。在创建索引的时候，应该考虑在哪些列上可以创建索引，在哪些列上不能创建索引。

一般来说，应该在哪些列上创建索引：

- （1）在经常需要搜索的列上，可以加快搜索的速度；
- （2）在作为主键的列上，强制该列的唯一性和组织表中数据的排列结构；
- （3）在经常用在连接的列上，这些列主要是一些外键，可以加快连接的速度；
- （4）在经常需要根据范围进行搜索的列上创建索引，因为索引已经排序，其指定的范围是连续的；

(5) 在经常需要排序的列上创建索引，因为索引已经排序，这样查询可以利用索引的排序，加快排序查询时间；

(6) 在经常使用在 WHERE 子句中的列上面创建索引，加快条件的判断速度。

同样，对于有些列不应该创建索引：

第一，对于那些在查询中很少使用或者参考的列不应该创建索引。这是因为，既然这些列很少使用到，因此有索引或者无索引，并不能提高查询速度。相反，由于增加了索引，反而降低了系统的维护速度和增大了空间需求。

第二，对于那些只有很少数据值的列也不应该增加索引。这是因为，由于这些列的取值很少，例如人事表的性别列，在查询的结果中，结果集的数据行占了表中数据行的很大比例，即需要在表中搜索的数据行的比例很大。增加索引，并不能明显加快检索速度。

第三，对于那些定义为 text, image 和 bit 数据类型的列不应该增加索引。这是因为，这些列的数据量要么相当大，要么取值很少。

第四，当修改性能远远大于检索性能时，不应该创建索引。这是因为，修改性能和检索性能是互相矛盾的。当增加索引时，会提高检索性能，但是会降低修改性能。当减少索引时，会提高修改性能，降低检索性能。因此，当修改性能远远大于检索性能时，不应该创建索引。

8、drop,delete 与 truncate 的区别

- drop 直接删掉表。
- truncate 删除表中数据，再插入时自增长 id 又从 1 开始。
- delete 删除表中数据，可以加 where 字句。

(1) DELETE 语句执行删除的过程是每次从表中删除一行，并且同时将该行的删除操作作为事务记录在日志中保存以便进行回滚操作。TRUNCATE TABLE 则一次性地从表中删除所有的数据并不把单独的删除操作记录记入日志保存，删除行是不能恢复的。并且在删除的过程中不会激活与表有关的删除触发器。执行速度快。

(2) 表和索引所占空间。当表被 TRUNCATE 后，这个表和索引所占用的空间会恢复到初始大小，而 DELETE 操作不会减少表或索引所占用的空间。drop 语句将表所占用的空间全释放掉。

(3) 一般而言，drop > truncate > delete

(4) 应用范围。TRUNCATE 只能对 TABLE；DELETE 可以是 table 和 view

(5) TRUNCATE 和 DELETE 只删除数据，而 DROP 则删除整个表（结构和数据）。

(6) truncate 与不带 where 的 delete : 只删除数据, 而不删除表的结构 (定义) drop 语句将删除表的结构被依赖的约束 (constrain), 触发器 (trigger) 索引 (index); 依赖于该表的存储过程/函数将被保留, 但其状态会变为: invalid。

(7) delete 语句为 DML (data maintain Language), 这个操作会被放到 rollback segment 中, 事务提交后才生效。如果有相应的 trigger, 执行的时候将被触发。

(8) truncate、drop 是 DLL (data define language), 操作立即生效, 原数据不放到 rollback segment 中, 不能回滚。

(9) 在没有备份情况下, 谨慎使用 drop 与 truncate。要删除部分数据行采用 delete 且注意结合 where 来约束影响范围。回滚段要足够大。要删除表用 drop; 若想保留表而将表中数据删除, 如果于事务无关, 用 truncate 即可实现。如果和事务有关, 或老师想触发 trigger, 还是用 delete。

(10) Truncate table 表名 速度快, 而且效率高, 因为: truncate table 在功能上与不带 WHERE 子句的 DELETE 语句相同: 二者均删除表中的全部行。但 TRUNCATE TABLE 比 DELETE 速度快, 且使用的系统和事务日志资源少。DELETE 语句每次删除一行, 并在事务日志中为所删除的每行记录一项。TRUNCATE TABLE 通过释放存储表数据所用的数据页来删除数据, 并且只在事务日志中记录页的释放。

(11) TRUNCATE TABLE 删除表中的所有行, 但表结构及其列、约束、索引等保持不变。新行标识所用的计数值重置为该列的种子。如果想保留标识计数值, 请改用 DELETE。如果要删除表定义及其数据, 请使用 DROP TABLE 语句。

(12) 对于由 FOREIGN KEY 约束引用的表, 不能使用 TRUNCATE TABLE, 而应使用不带 WHERE 子句的 DELETE 语句。由于 TRUNCATE TABLE 不记录在日志中, 所以它不能激活触发器。