（一）什么是存储过程？有哪些优缺点？

存储过程是一些预编译的SQL语句。

更加直白的理解：存储过程可以说是一个记录集，它是由一些T-SQL语句组成的代码块，这些T-SQL语句代码像一个方法一样实现一些功能（对单表或多表的增删改查），然后再给这个代码块取一个名字，在用到这个功能的时候调用他就行了。

* 存储过程是一个预编译的代码块，执行效率比较高
* 一个存储过程替代大量T\_SQL语句 ，可以降低网络通信量，提高通信速率
* 可以一定程度上确保数据安全

如果你对存储过程不熟悉，建议阅读：[存储过程详解-博客园](http://link.zhihu.com/?target=http%3A//www.cnblogs.com/knowledgesea/archive/2013/01/02/2841588.html)

（二）索引是什么？有什么作用以及优缺点？

索引是对数据库表中一或多个列的值进行排序的结构，是帮助MySQL高效获取数据的数据结构

你也可以这样理解：索引就是加快检索表中数据的方法。数据库的索引类似于书籍的索引。在书籍中，索引允许用户不必翻阅完整个书就能迅速地找到所需要的信息。在数据库中，索引也允许数据库程序迅速地找到表中的数据，而不必扫描整个数据库。

**MySQL**数据库几个基本的索引类型：普通索引、唯一索引、主键索引、全文索引

* 索引加快数据库的检索速度
* 索引降低了插入、删除、修改等维护任务的速度
* 唯一索引可以确保每一行数据的唯一性
* 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
* 索引需要占物理和数据空间

如果你对索引还不太熟悉，建议阅读：[漫谈数据库索引](http://link.zhihu.com/?target=http%3A//kb.cnblogs.com/page/45712/" \t "_blank)

（三）什么是事务？

事务（Transaction）是并发控制的基本单位。所谓的事务，它是一个操作序列，这些操作要么都执行，要么都不执行，它是一个不可分割的工作单位。事务是数据库维护数据一致性的单位，在每个事务结束时，都能保持数据一致性。

如果你对索引还不太熟悉，建议阅读：[数据库事务 - Mr. David 专栏](http://link.zhihu.com/?target=http%3A//blog.csdn.net/zdwzzu2006/article/details/5947062" \t "_blank)

（四）数据库的乐观锁和悲观锁是什么？

数据库管理系统（DBMS）中的并发控制的任务是确保在多个事务同时存取数据库中同一数据时不破坏事务的隔离性和统一性以及数据库的统一性。

乐观并发控制(乐观锁)和悲观并发控制（悲观锁）是并发控制主要采用的技术手段。

* 悲观锁：假定会发生并发冲突，屏蔽一切可能违反数据完整性的操作
* 乐观锁：假设不会发生并发冲突，只在提交操作时检查是否违反数据完整性。

如果你对两种锁还不太熟悉，建议阅读：[深入理解乐观锁与悲观锁](http://link.zhihu.com/?target=http%3A//www.open-open.com/lib/view/open1452046967245.html)

（五） 使用索引查询一定能提高查询的性能吗？为什么

通常,通过索引查询数据比全表扫描要快.但是我们也必须注意到它的代价.

索引需要空间来存储,也需要定期维护, 每当有记录在表中增减或索引列被修改时,索引本身也会被修改. 这意味着每条记录的INSERT,DELETE,UPDATE将为此多付出4,5 次的磁盘I/O. 因为索引需要额外的存储空间和处理,那些不必要的索引反而会使查询反应时间变慢.使用索引查询不一定能提高查询性能,索引范围查询(INDEX RANGE SCAN)适用于两种情况:

* 基于一个范围的检索,一般查询返回结果集小于表中记录数的30%
* 基于非唯一性索引的检索

（六）简单说一说**drop、delete**与**truncate**的区别

SQL中的**drop、delete、truncate**都表示删除，但是三者有一些差别

* **delete和truncate**只删除表的数据不删除表的结构
* 速度,一般来说:**drop> truncate >delete**
* **delete**语句是dml,这个操作会放到**rollback segement**中,事务提交之后才生效;  
  如果有相应的**trigger**,执行的时候将被触发. **truncate,drop**是ddl, 操作立即生效,原数据不放到r**ollback segment**中,不能回滚. 操作不触发**trigger**.

如果你对这三者的用法还不太熟悉，建议阅读：[drop、truncate和delete的区别](http://link.zhihu.com/?target=http%3A//blog.csdn.net/ws0513/article/details/49980547)

（七）**drop、delete**与**truncate**分别在什么场景之下使用？

* 不再需要一张表的时候，用**drop**
* 想删除部分数据行时候，用**delete**，并且带上where子句
* 保留表而删除所有数据的时候用**truncate**

（八） 超键、候选键、主键、外键分别是什么？

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。一个属性可以为作为一个超键，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。

（九）什么是视图？以及视图的使用场景有哪些？

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，试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。

* 只暴露部分字段给访问者，所以就建一个虚表，就是视图。
* 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

（十）说一说三个范式。

第一范式（1NF）：数据库表中的字段都是单一属性的，不可再分。这个单一属性由基本类型构成，包括整型、实数、字符型、逻辑型、日期型等。

第二范式（2NF）：数据库表中不存在非关键字段对任一候选关键字段的部分函数依赖（部分函数依赖指的是存在组合关键字中的某些字段决定非关键字段的情况），也即所有非关键字段都完全依赖于任意一组候选关键字。

第三范式（3NF）：在第二范式的基础上，数据表中如果不存在非关键字段对任一候选关键字段的传递函数依赖则符合第三范式。所谓传递函数依赖，指的是如 果存在"A → B → C"的决定关系，则C传递函数依赖于A。因此，满足第三范式的数据库表应该不存在如下依赖关系： 关键字段 → 非关键字段 x → 非关键字段y

所谓的关系型数据库，是建立在关系模型基础上的数据库，借助于集合代数等数学概念和方法来处理数据库中的数据。

本章将向您讲解 SQLite 编程人员所使用的简单却有用的命令。这些命令被称为 SQLite 的点命令，这些命令的不同之处在于它们不以分号 **;** 结束。

有个重要的点值得注意，SQLite 是**不区分大小写**的，但也有一些命令是大小写敏感的，比如 **GLOB** 和 **glob** 在 SQLite 的语句中有不同的含义。

在 SQLite 中，值的数据类型与值本身是相关的，而不是与它的容器相关。