数据库面试题 1

1. 在一个查询中，使用哪一个关键字能够除去重复列值？  
答案：使用distinct关键字

2. 什么是快照？它的作用是什么？  
答案：快照Snapshot是一个文件系统在特定时间里的镜像，对于在线实时数据备份非常有用。快照对于拥有不能停止的应用或具有常打开文件的文件系统的备份非常重要。对于只能提供一个非常短的备份时间而言，快照能保证系统的完整性。

3. 解释存储过程和触发器  
答案：  
存储过程是一组Transact-SQL语句，在一次编译后可以执行多次。因为不必重新编译Transact-SQL语句，所以执行存储过程可以提高性能。  
触发器是一种特殊类型的存储过程，不由用户直接调用。创建触发器时会对其进行定义，以便在对特定表或列作特定类型的数据修改时执行。

4. SQL Server是否支持行级锁，有什么好处？  
答案：支持动态行级锁定  
SQL Server 2000动态地将查询所引用的每一个表的锁定粒度调整到合适的级别。当查询所引用的少数几行分散在一个大型表中时，优化数据并行访问的最佳办法是使用粒度锁，如行锁。但是，如果查询引用的是一个表中的大多数行或所有行，优化数据并行访问的最佳办法可以是锁定整个表，以尽量减少锁定开销并尽快完成查询。  
SQL Serve 2000通过为每个查询中的每个表选择适当的锁定级别，在总体上优化了数据并发访问。对于一个查询，如果只引用一个大型表中的几行，则数据库引擎可以使用行级锁定；如果引用一个大型表的几页中的多行，则使用页级锁定；如果引用一个小型表中的所有行，则使用表级锁定。

5. 数据库日志干什么用，数据库日志满的时候再查询数据库时会出现什么情况。  
答案：每个数据库都有事务日志，用以记录所有事务和每个事务对数据库所做的修改。

6. 存储过程和函数的区别？  
答案：存储过程是用户定义的一系列SQL语句的集合，涉及特定表或其它对象的任务，用户可以调用存储过程，而函数通常是数据库已定义的方法，它接收参数并返回某种类型的值并且不涉及特定用户表

7. 事务是什么？  
答案：事务是作为一个逻辑单元执行的一系列操作，一个逻辑工作单元必须有四个属性，称为 ACID（原子性、一致性、隔离性和持久性）属性，只有这样才能成为一个事务：  
(1) 原子性  
事务必须是原子工作单元；对于其数据修改，要么全都执行，要么全都不执行。  
(2) 一致性  
事务在完成时，必须使所有的数据都保持一致状态。在相关数据库中，所有规则都必须应用于事务的修改，以保持所有数据的完整性。事务结束时，所有的内部数据结构（如 B 树索引或双向链表）都必须是正确的。  
(3) 隔离性  
由并发事务所作的修改必须与任何其它并发事务所作的修改隔离。事务查看数据时数据所处的状态，要么是另一并发事务修改它之前的状态，要么是另一事务修改它之后的状态，事务不会查看中间状态的数据。这称为可串行性，因为它能够重新装载起始数据，并且重播一系列事务，以使数据结束时的状态与原始事务执行的状态相同。  
(4) 持久性  
事务完成之后，它对于系统的影响是永久性的。该修改即使出现系统故障也将一直保持。

8. 游标的作用？如何知道游标已经到了最后？  
答案：游标用于定位结果集的行，通过判断全局变量@@FETCH\_STATUS可以判断是否到了最后，通常此变量不等于0表示出错或到了最后。

9. 触发器分为事前触发和事后触发，这两种触发有何区别？语句级触发和行级触发有何区别？  
答案：  
事前触发器运行于触发事件发生之前，而事后触发器运行于触发事件发生之后。通常事前触发器可以获取事件之前和新的字段值。  
语句级触发器可以在语句执行前或后执行，而行级触发在触发器所影响的每一行触发一次。

10. 解决死锁的方法有哪些？  
答案：  
解决手段一：SQL Server自动检测和消除死锁  
解决手段二：设置死锁优先级  
解决手段三：设置锁定超时  
解决手段四：使用更新锁避免死锁

11. 提高数据库运行效率的办法有哪些？  
答案：在给定的系统硬件和系统软件条件下，提高数据库系统的运行效率的办法是：  
(1) 在数据库物理设计时，降低范式，增加冗余, 少用触发器, 多用存储过程。  
(2) 当计算非常复杂、而且记录条数非常巨大时(例如一千万条)，复杂计算要先在数据库外面，以文件系统方式用C++语言计算处理完成之后，最后才入库追加到表中去。这是电信计费系统设计的经验。  
(3) 发现某个表的记录太多，例如超过一千万条，则要对该表进行水平分割。水平分割的做法是，以该表主键PK的某个值为界线，将该表的记录水平分割为两个表。若发现某个表的字段太多，例如超过八十个，则垂直分割该表，将原来的一个表分解为两个表。  
(4) 对数据库管理系统DBMS进行系统优化，即优化各种系统参数，如缓冲区个数。  
(5) 在使用面向数据的SQL语言进行程序设计时，尽量采取优化算法。总之，要提高数据库的运行效率，必须从数据库系统级优化、数据库设计级优化、程序实现级优化，这三个层次上同时下功夫。

12. 通俗地理解三个范式  
答案：通俗地理解三个范式，对于数据库设计大有好处。在数据库设计中，为了更好地应用三个范式，就必须通俗地理解三个范式(通俗地理解是够用的理解，并不是最科学最准确的理解)：  
第一范式：1NF是对属性的原子性约束，要求属性具有原子性，不可再分解； 第二范式：2NF是对记录的惟一性约束，要求记录有惟一标识，即实体的惟  
一性；  
第三范式：3NF是对字段冗余性的约束，即任何字段不能由其他字段派生出来，它要求字段没有冗余。没有冗余的数据库设计可以做到。但是，没有冗余的数据库未必是最好的数据库，有时为了提高运行效率，就必须降低范式标准，适当保留冗余数据。具体做法是：在概念数据模型设计时遵守第三范式，降低范式标准的工作放到物理数据模型设计时考虑。降低范式就是增加字段，允许冗余。

13. 简述存储过程的优缺点  
优点：  
1. 更快的执行速度：存储过程只在创造时进行编译，以后每次执行存储过程都不需再重新编译，而一般SQL语句每执行一次就编译一次,所以使用存储过程可提高数据库执行速度；  
2. 与事务的结合，提供更好的解决方案：当对数据库进行复杂操作时(如对多个表进行Update、Insert、Query和Delete时），可将此复杂操作用存储过程封装起来与数据库提供的事务处理结合一起使用；  
3. 支持代码重用：存储过程可以重复使用,可减少数据库开发人员的工作量；4. 安全性高：可设定只有某此用户才具有对指定存储过程的使用权。

缺点：  
1. 如果更改范围大到需要对输入存储过程的参数进行更改，或者要更改由其返回的数据，则您仍需要更新程序集中的代码以添加参数、更新 GetValue() 调用，等等，这时候估计比较繁琐了。  
2. 可移植性差由于存储过程将应用程序绑定到 SQL Server，因此使用存储过程封装业务逻辑将限制应用程序的可移植性。如果应用程序的可移植性在您的环境中非常重要，则将业务逻辑封装在不特定于 RDBMS 的中间层中可能是一个更佳的选择。

14. 主键和唯一索引有什么区别？  
答案：  
相同点:它们都属于实体完整性约束。  
不同点:  
(1) 唯一性约束所在的列允许空值，但是主键约束所在的列不允许空值。  
(2) 可以把唯一性约束放在一个或者多个列上，这些列或列的组合必须有唯一的。但是，唯一性约束所在的列并不是表的主键列。  
(3) 唯一性约束强制在指定的列上创建一个唯一性索引。在默认情况下，创建唯一性的非聚簇索引，但是，也可以指定所创建的索引是聚簇索引。  
(4) 建立主键的目的是让外键来引用。  
(5) 一个表最多只有一个主键，但可以有很多唯一键。

15. 简述索引存取的方法的作用和建立索引的原则  
作用：加快查询速度。  
原则：  
(1) 如果某属性或属性组经常出现在查询条件中，考虑为该属性或属性组建立索引；  
(2) 如果某个属性常作为最大值和最小值等聚集函数的参数，考虑为该属性建立索引；  
(3) 如果某属性经常出现在连接操作的连接条件中，考虑为该属性或属性组建立索引；

16. 简述数据库的设计过程  
数据库设计分为五个阶段：  
需求分析：主要是准确收集用户信息需求和处理需求，并对收集的结果进行整理和分析，形成需求说明。  
概念结构设计：对用户需求进行综合、归纳、抽象，形成一个与与具体的DBMS无关概念模型（一般为ER模型）。  
逻辑结构设计：将概念结构设计的概念模型转化为某个特定的DBMS所支持的数据模型，建立数据库逻辑模式，并对其进行优化，同时为各种用户和应用设计外模式。  
物理结构设计：为设计好的逻辑模型选择物理结构，包括存储结构和存取方法等，建立数据库物理模式。  
实施和维护：实施就是使用DLL语言建立数据库模式，将实际数据载入数据库，建立真正的数据库。维护阶段是对运行中的数据库进行评价、调整和修改。

17. 什么是内存泄漏？  
答案：一般我们所说的内存泄漏指的是堆内存的泄漏。堆内存是程序从堆中为其分配的，大小任意的，使用完后要显示释放内存。当应用程序用关键字new等创建对象时，就从堆中为它分配一块内存，使用完后程序调用free或者delete释放该内存，否则就说该内存就不能被使用，我们就说该内存被泄漏了。

18. 什么是基本表？什么是视图？  
答案：基本表是本身独立存在的表，在 SQL 中一个关系就对应一个表。  
视图是从一个或几个基本表导出的表。视图本身不独立存储在数据库中，是一个虚表

19. 试述视图的优点  
(1) 视图能够简化用户的操作  
(2) 视图使用户能以多种角度看待同一数据；  
(3) 视图为数据库提供了一定程度的逻辑独立性；  
(4) 视图能够对机密数据提供安全保护。

20. 所有的视图是否都可以更新？为什么？  
答案：不是。  
视图是不实际存储数据的虚表，因此对视图的更新，最终要转换为对基本表的更新。  
因为有些视图的更新不能惟一有意义地转换成对相应基本表的更新，所以，并不是所有的视图都是可更新的。

21. 哪类视图是可以更新的？哪类视图是不可更新的？各举一例说明。  
答案：基本表的行列子集视图一般是可更新的。若视图的属性来自集函数、表达式，则该视图肯定是不可以更新的。

22. 维护数据库的完整性、一致性、你喜欢用触发器还是自写业务逻辑？为什么?  
答案：尽可能用约束（包括CHECK、主键、唯一键、外键、非空字段）实现，这种方式的效率最好；其次用触发器，这种方式可以保证无论何种业务系统访问数据库都能维持数据库的完整性、一致性；最后再考虑用自写业务逻辑实现，但这种方式效率最低、编程最复杂，当为下下之策。

23. 在为视图创建索引前，视图本身必须满足哪些条件？  
(1) 视图以及视图中引用的所有表都必须在同一数据库中，并具有同一个所有者  
(2) 索引视图无需包含要供优化器使用的查询中引用的所有表。  
(3) 必须先为视图创建唯一群集索引，然后才可以创建其它索引。  
(4) 创建基表、视图和索引以及修改基表和视图中的数据时，必须正确设置某些 SET 选项（在本文档的后文中讨论）。另外，如果这些 SET 选项正确，查询优化器将不考虑索引视图。  
(5) 视图必须使用架构绑定创建，视图中引用的任何用户定义的函数必须使用 SCHEMABINDING 选项创建。  
(6) 另外，还要求有一定的磁盘空间来存放由索引视图定义的数据。

24. 什么是SQL Server的确定性函数和不确定性函数？  
答案：只要使用特定的输入值集并且数据库具有相同的状态，不管何时调用，始终都能范围相同结果的函数叫确定性函数。几十访问的数据库的状态不变，每次书用特定的输入值都可能范围不同结果的函数叫非确定性函数。