

你能向我简要叙述一下SQL Server 2000中使用的一些数据库对象吗?

你希望听到的答案包括这样一些对象:表格、视图、用户定义的函数,以及存储过程;如果他们还能够提到像触发器这样的对象就更好了。如果应聘者不能回答这个基本的问题,那么这不是一个好兆头。

NULL是什么意思?

NULL(空)这个值是数据库世界里一个非常难缠的东西,所以有不少应聘者会在这个问题上跌跟头您也不要觉得意外。

NULL这个值表示UNKNOWN(未知):它不表示""(空字符串)。假设您的SQL Server数据库里有ANSI_NULLS,当然在默认情况下会有,对NULL这个值的任何比较都会生产一个NULL值。您不能把任何值与一个UNKNOWN值进行比较,并在逻辑上希望获得一个答案。您必须使用IS NULL操作符。

什么是索引?SQL Server 2000里有什么类型的索引?

任何有经验的数据库开发人员都应该能够很轻易地回答这个问题。一些经验不太多的开发人员能够回答这个问题,但是有些地方会说不清楚。

简单地说,索引是一个数据结构,用来快速访问数据库表格或者视图里的数据。在SQL Server里,它们有两种形式:聚集索引和非聚集索引。聚集索引在索引的叶级保存数据。这意味着不论聚集索引里有表格的哪个(或哪些)字段,这些字段都会按顺序被保存在表格。由于存在这种排序,所以每个表格只会有一个聚集索引。非聚集索引在索引的叶级有一个行标识符。这个行标识符是一个指向磁盘上数据的指针。它允许每个表格有多个非聚集索引。

什么是主键?什么是外键?

主键是表格里的(一个或多个)字段,只用来定义表格里的行;主键里的值总是唯一的。外键是一个用来建立两个表格之间关系的约束。这种关系一般都涉及一个表格里的的主键字段与另外一个表格(尽管可能是同一个表格)里的一系列相连的字段。那么这些相连的字段就是外键。

什么是触发器?SQL Server 2000有什么不同类型的触发器?

让未来的数据库开发人员知道可用的触发器类型以及如何实现它们是非常有益的

触发器是一种专用类型的存储过程,它被捆绑到SQL Server 2000的表格或者视图上。在SQL Server 2000里,有INSTEAD-OF和AFTER两种触发器。INSTEAD-OF触发器是替代数据操控语言(Data Manipulation Language, DML)语句对表格执行语句的存储过程。例如,如果我有一个用于TableA的INSTEAD-OF-UPDATE触发器,同时对这个表格执行一个更新语句,那么INSTEAD-OF-UPDATE触发器里的代码会执行,而不是我执行的更新语句则不会执行操作。

AFTER触发器要在DML语句在数据库里使用之后才执行。这些类型的触发器对于监视发生在数据库表格里的数据变化十分好用。

您如何确定一个带有名为Fld1字段的TableB表格里只具有Fld1字段里的那些值,而这些值同时在名为TableA的表格的Fld1字段里?

这个与关系相关的问题有两个可能的答案。第一个答案(而且是您希望听到的答案)是使用外键限制。外键限制用来维护引用的完整性。它被用来确保表格里的字段只保存有已经在不同的(或者相同的)表格里的另一个字段里定义了的值。这个字段就是候选键(通常是另外一个表格的主键)。

另外一种答案是触发器。触发器可以被用来保证以另外一种方式实现与限制相同的作用,但是它非常难设置与维护,而且性能一般都很糟糕。由于这个原因,微软建议开发人员使用外键限制而不是触发器来维护引用的完整性。

对一个投入使用的在线事务处理表格有过多索引需要有什么样的性能考虑?

你正在寻找进行与数据操控有关的应聘人员。对一个表格的索引越多,数据库引擎用来更新、插入或者删除数据所需要的时间就越多,因为在数据操控发生的时候索引也必须要维护。

你可以用什么来确保表格里的字段只接受特定范围里的值?

这个问题可以用多种方式来回答，但是只有一个答案是“好”答案。您希望听到的回答是Check限制，它在数据库表格里被定义，用来限制输入该列的值。

触发器也可以被用来限制数据库表格里字段能够接受的值，但是这种办法要求触发器在表格里被定义，这可能会在某些情况下影响到性能。因此，微软建议使用Check限制而不是其他方式来限制域的完整性。

返回参数和output参数之间的区别是什么？如果应聘者能够正确地回答这个问题，那么他的机会就非常大了，因为这表明他们具有使用存储过程的经验。

返回参数总是由存储过程返回，它用来表示存储过程是成功还是失败。返回参数总是INT数据类型。OUTPUT参数明确要求由开发人员来指定，它可以返回其他类型的数据，例如字符型和数值型的值。(可以用作输出参数的数据类型是有一些限制的。)您可以在一个存储过程里使用多个OUTPUT参数，而您只能使用一个返回参数。

什么是相关子查询？如何使用这些查询？

经验更加丰富的开发人员将能够准确地描述这种类型的查询。

相关子查询是一种包含子查询的特殊类型的查询。查询里包含的子查询会真正请求外部查询的值，从而形成一个类似于循环的状况。

返回参数和output参数之间的区别是什么？
建表，增，删，改，查，排序，分组，升序，降序，表连接（左右），有工作经验那就要求高啦，数据库优化，数据结构，存储过程，游标，sql脚本

1. 从哈希表，二叉树和链表中取元素需要多少时间？如果你有数百万记录呢？

哈希表需要 $O(1)$ 时间，二叉树需要 $O(\log N)$ (N 是树中节点数)，链表需要 $O(N)$ (N 是链表中节点数)。如果数据结构工作正常（比如哈希表没有或只有相对少量冲突，二叉树是平衡的），数百万记录并不影响效率。如果工作不正常，那么效率会随着记录数上升而下降。

2. 覆盖(Overriding)和重载(Overloading)的区别是什么？ ([detailed answer](#))

覆盖在运行时决定，重载是在编译时决定。并且覆盖和重载的机制不同，例如在Java中，重载方法的签名必须不同于原先方法的，但对于覆盖签名必须相同。

3. fork一个进程和生成一个线程有什么区别？

当你fork一个进程时，新的进程将执行和父进程相同的代码，只是在不同的内存空间中。但当你已在有进程中生成一个线程时，它会生成一个新的代码执行路线，但共享同一个内存空间。

4. 什么是临界区？ ([answer](#))

临界区是一段代码，十分重要，在多线程中同一时间只能被一个线程执行。可以用信号量或互斥量来保护临界区。在Java中你可以用synchronized关键字或ReentrantLock来保护临界区。

5. 值类型和引用类型有什么区别？ ([answer](#))

值类型是更加优化的类型，总是不可变的(immutable)，例如Java原始的int、long、double和float。引用类型指向一个对象，可能是可变的，也可能是不变的。你也可以说值类型指向一个值，引用类型指向一个对象。

6. 什么是在进程中的堆和栈？ ([detailed answer](#))

在同一个进程中，有两块不同的内存区域。以Java来说，栈用来存储原始值和指向对象的引用类型，但对象本身总是在堆中被创建。堆和栈的一个重要区别是，堆内存被所有线程共享，但每个线程有自己的栈。

7. 什么是版本控制？ ([answer](#))

版本控制是用来存储代码和管理代码库版本的软件，例如SVN、CVS、Git、Perforce和ClearCase。它们在对代码、审查代码和从之前的稳定版本构造时十分高效。所有的专业开发都使用某种版本控制工具，否则你无法有效的管理代码，尤其是如果有20个开发者在同一个代码库上工作的时候。版本控制工具在保持代码库一致性和处理代码冲突上扮演着十分重要的角色。

8. 什么是强类型程序设计语言？ ([answer](#))

在强类型语言中，编译器确保类型的正确性，例如你无法在String类型中存放数字，反之亦然。Java是强类型语言，因此存在各种数据类型（如int、float、String、char、boolean等）。你只能将兼容的值存入相应的类型中。与此相反，弱类型语言不要求在编译时进行类型检查，它们根据上下文处理值。Python和Perl是两个常见的弱类型程序设计语言的例子，你可以将数字组成的字符串保存在数字类型中。

9. 可否描述一下有效(valid)的XML和格式正确(well-formed)的XML的区别？

格式正确的XML有根元素，所有标签都是正确关闭的，属性是正确定义的，它们的值正确地加上了引号。另一方面，有效的XML可以根据一个XSD文件或模式(schema)进行验证。所以一个XML可能是格式正确但不有效的（因为包含不被模式允许的标签）。

10. DOM和SAX语法分析器有什么区别？([detailed answer](#))

DOM语法分析器是驻留内存的，将整个XML文件装载到内存中，并创建一个DOM树进行语法分析。SAX语法分析器是一个基于事件的语法分析器，所以它根据收到的事件（如开始标签、结束标签、属性开始和属性结束）来对XML文档进行语法分析。根据他们的分析方法，DOM语法分析器并不适用于大的XML文件，因为它会占用大量的内存空间，你的进程可能会耗尽内存。应该用SAX分析大的文件。对于小的文件，DOM往往比SAX快很多。

11. 线程和进程的关系是什么？([detailed answer](#))

一个进程可以有多个线程，但一个线程总是属于唯一的进程。两个进程不能共享内存空间，除非它们有意通过共享内存进行进程间通信。但是同一进程的两个线程总是共享相同的内存。

12. 不可变(immutable)类是什么意思？([detailed answer](#))

一个类，如果在创建之后它的状态就不能被改变，那么他就是不可变的。例如Java中的String。一旦你创建了一个String，例如“Java”，你就不能再改变它的内容。任何对这个字符串的改变（例如转换到大写、与另一个String连接）将创建一个新的对象。不可变的对象在并程序设计中很有用，因为它们可以在进程间被共享，不需要担心同步。事实上，整个函数式程序设计的模型都是基于不可变对象构建的。

13. 你为何要创建模拟(mock)对象？([answer](#))

模拟对象在测试软件中一个独立的单元时很有用，事实上，存根(stub)和模拟都是创建自动化单元测试的有力工具。假设你在写一个显示货币兑换率的程序，但没有一个可以连通的URL，现在如果想测试你的代码，可以用模拟对象。在Java的世界中，有很多框架可以为你生成强大的模拟对象，例如Mockito和PowerMock。

14. 什么是SQL注入？

SQL注入是一种安全漏洞，它使得入侵者可以从系统中窃取数据。任何从用户那里得到输入并不加验证地创建SQL查询的系统都可能被SQL注入攻击。在这样的系统中，入侵者可以输入SQL代码，而不是数据，来获取额外的数据。有很多敏感信息（如用户id、密码和个人信息）被人利用这种漏洞获取的实例。在Java中，你可以用Prepared语句来避免SQL注入。

15. 在SQL中，内连接(inner join)和左连接(left join)有什么区别？([answer](#))

在SQL中，主要有两种连接类型，内连接和外连接。外连接包括右外连接和左外连接。内连接和左连接的主要区别是，内连接中两个表都匹配的记录才被选中，左连接中两个表都匹配的记录被选中，外加左表的所有记录都被选中。要留意包含“所有”的查询，它们往往要求左连接，例如写一个SQL查询来找所有的部门和它们的雇员人数。如果你用内连接处理这个查询，你会漏掉没有人工作的空部门。

16. MVC中的V代表什么，意味着什么？([answe](#))

在MVC模式中，V是视图(View)。视图是用户看到的东西，比如网页。这是一个非常重要的web应用开发设计模式，它基于关注点分离原则，目的是不同模块可以独立修改，不影响其他模块。在Java的世界中，有很多提供MVC模式的开源框架，例如Struts 2和Spring MVC。顺便说一下，M代表模

型(Model), C代表控制器(Controller)。模型是实际的业务对象, 例如用户、雇员、订单, 控制器用来将请求分发给正确的处理单元。

17. 类和对象的区别是什么? ([detailed answer](#))

类是用来创建对象的设计图。一个类包括代码和行为, 一个对象包括状态和行为。要创建一个对象, 你必须创建一个表达对象结构的类。类还被用来在内存中映射对象, 在Java中, JVM替你完成这项工作。

18. 什么是疏耦合(loose-coupling)?

疏耦合是一种值得追求的软件特性, 它使得对软件一个部分的修改不会影响到其他的部分。例如, 在一个疏耦合的软件中, 对UI布局的改变不应该影响后端的类结构。

19. 组合(composition), 聚合(aggregation)和关联(association)的区别是什么? ([detailed answer](#))

关联的意思是两个对象是相互联系的。组合是关联的一种形式, 即一个对象由多个对象组成, 但是它们必须共存, 例如人体由各种器官组合而成, 独立的器官不能生存, 它们必须在身体内发挥作用。聚合也是关联的一种形式, 表示对象的集合, 例如城市是居民的聚合。

20. 接口和抽象类有什么区别? ([detailed answer](#))

这是所有程序员面试最经典的问题。接口是最纯粹的抽象形式, 没有任何具体的东西。抽象类是一些抽象和具体事物的组合体。这个区别在不同语言中可能会不同, 例如在Java中你可以扩展(extend)多个接口, 但只能继承一个抽象类。更详细的讨论见于详细答案。

21. 什么是单元测试? ([answer](#))

单元测试是测试独立单元 (而不是整个应用程序) 功能性的一种方法。在不同语言中, 有很多工具可以做单元测试。例如在Java中, 你可以用JUnit或TestNG来写单元测试。单元测试经常在构建时自动运行, 或者在一个持续的环境 (例如Jenkins) 中运行。

22. 你能否描述三种不同的在应用程序发布前对其进行测试的方式?

单元测试, 集成测试, 冒烟测试。单元测试用来测试独立的单元是否依照预期工作, 集成测试用来测试已被测试过的独立单元能否共同工作, 冒烟测试用来测试软件最常用的功能是否正常工作, 例如在一个飞机订票网站中, 你应该能订票, 取消或更改航班。

23. 迭代和递归有什么区别? ([detailed answer](#))

迭代通过循环来重复执行同一步骤, 递归通过调用函数自身来做重复性工作。递归经常是复杂问题 (例如汉诺塔、反转链表或反转字符串) 的清晰简洁的解决方案。递归的一个缺陷是深度, 由于递归在栈中存储中间结果, 你只能进行一定深度的递归, 在那之后你的程序会因为StackOverflowError而崩溃。这就是在产品代码中优先使用迭代而不是递归的原因。

24. &和&&运算符的区别是什么? ([detailed answer](#))

&是位运算符, &&是逻辑运算符。&和&&的一个区别是位运算符 (&) 可以被用于整型和布尔类型, 逻辑运算符 (&&) 只能被用于布尔类型变量。当你写a & b时, 两个整型数的每一位都会进行与运算。当你写a && b时, 第二个参数可能会也可能不会被执行, 这也是它被称为短路运算符的原因, 至少在Java中是这样的。我很喜欢这个问题, 经常对初级开发者和毕业生问这个问题。

25. 1 XOR 1的结果是什么?

答案是0, 因为XOR在两个操作数 (按位) 不同时返回1, 相同时返回0。例如0 XOR 0仍然是零, 但0 XOR 1和1 XOR 0的结果是1。

26. 如何得到一个整型数的最后一位? ([answer](#))

用取模运算符, 数字 % 10返回数字的最后一位。例如2345 % 10会返回5, 567 % 10会返回7。类似的, 除运算符可以用来去掉数字的最后一位, 例如2345 / 10的结果是234, 567 / 10的结果是56。这是值得了解的一个重要技巧, 可以用来解决类似回文数、反转数的问题。

27. 什么是测试驱动开发?

测试驱动是一种常见的开发方法，在这种方法中，测试代码在功能代码之前编写。测试决定了程序的结构。测试驱动的纯粹主义者在写应用写测试之前，不会写一行的应用代码。这能大幅度地提高代码质量，经常被认为是巨星级开发者的品质。

28. 里氏替换原则(Liskov substitution principle, LSP)是什么? ([answer](#))

里氏替换原则是鲍勃大叔称作SOLID的五条设计原则中的一条。里氏替换原则规定，所有的子类都能作为父类的代理(proxy)工作。例如，如果一个方法需要父类对象作为输入，那么如果你提供一个子类对象，它也应该正常工作。任何不能替代父类的类都违反了里氏替换原则。这实际上是一个难以答出的问题，如果你答出了，那么就会给面试官留下好的印象。

29. 什么是开闭(Open closed)设计原则? ([answer](#))

开闭原则是SOLID中另一个重要的原则，它规定一个系统对扩展是开放的，但对修改是封闭的。意思是说，如果一个新的功能要被加入一个稳定的系统，那么你不需碰已被测试过的现有代码，新的功能可以通过只添加新类来实现。

30. 二叉树和二叉查找树的区别是什么?

二叉查找树是有序的二叉树，所有节点（例如根节点）的左子树节点的值都小于或等于该节点的值，右子树节点的值都大于或等于该节点的值。它是一个重要的数据结构，可以用来表示有序的数据。

31. 你能否给出一个递归算法的实际例子? ([example](#))

递归算法能适用在很多地方，例如与二叉树和链表相关的算法。几个与递归算法的例子包括反转字符串，计算斐波那契数列。其他的例子包括反转链表、树遍历以及快速排序。

31. 算法的时间复杂度是什么?

时间复杂度表示的是运行时间对输入量的比率。他表示一个算法处理一定量的输入需要多长时间。它是一个估计值，但足够表示输入量从十增长到一千万时，你的算法会有什么样的表现。

33. 链表和数组有哪些重要区别? ([detailed answer](#))

链表和数组都是程序设计世界中重要的数据结构。它们间最明显的区别是，数组将元素存放在连续的地址中，链表将数据存放在内存中任意的地方。这使得链表有巨大的扩展自己的灵活性，因为内存总是分散的。这种情况总是可能的：你无法创建一个数组来存放一百万个整数，但可以用链表来存放，因为空间是存在的，只是不连续。其他不同都是源于这项事实。例如，在数组中，如果你知道下标，可以用 $O(1)$ 的时间得到一个元素，但在链表中要花 $O(n)$ 的时间。更多不同参见详细答案。

33. 在哈希表中处理冲突的方法都有哪些?

线性探测(linear probing)，二次哈希(double hashing)和链接(chaining)。在线性探测中，如果桶已经被占据了，那么函数会线性地检查下一个桶，直到找到一个空位。在链接中，多个元素可以存储在同一个桶中。

34. 正则表达式是什么意思? ([answer](#))

正则表达式是在文本型数据上进行模式匹配的方法。它是一种搜索的强有力方法，例如搜索长字符串中的某些字符，例如搜索一本书中是否含有某个单词。所有主流程序设计语言都支持正则表达式，但是Perl正则表达式的能力是著名的。Java的java.util.regex包也支持类似Perl的正则表达式。你可以用正则表达式检查email地址是否有效，电话号码是否有效，邮政编码是否有效，甚至社会保险号(SSN)是否有效。正则表达式最简答的例子之一是检查字符串是不是一个数。

35. 什么是无状态(stateless)系统?

无状态系统是不维护内部状态的系统。这种系统在任何时刻，对相同的输入都会给出相同的输出。编写优化一个无状态系统总是比较简单的，所以如果可能，你总是应该优先编写无状态系统。

36. 写一个SQL查询，在雇员表中查找第二高的工资。 ([solution](#))

这是SQL面试的经典题目之一，尽管已经很老了，还是很有趣，并且可以追问很多问题来测试候选人的知识深度。你可以用相关或不相关的子查询来查找第二高工资。如果你在用SQL Server或

MySQL，你也可以用类似TOP和LIMIT之类的关键字，前提是面试官允许。查找第二高工资的最简答方法是：

这个查询首先查找最高工资，然后将它从列表中排除，再查找最高工资。很明显，第二次返回的是第二高工资。

37. 可否描述一下什么是关联的和不关联的子查询？(answer)

在关联的子查询中，内层查询依赖于外层查询，对外层查询的每一行执行。非关联的子查询不依赖于外层查询，可以独立执行。因此，前者慢，后者快。顺便说一下，关联的子查询有一些很棒的应用，其中包括在雇员表中查找第N高的工资，这在上一道SQL问题中也有提到。

39. 不用算术运算符，如何判定一个数是否是二的幂？(solution)

当你听到不能用算术运算符的限制时，应该立刻假定这是一道关于位运算的题。如果没有这条限制，那么你可以轻松地用取模和除运算符检查一个数是不是二的幂。用位运算符，有一个很巧妙的方法来完成。你可以用下面这段代码来检查一个数是不是二的幂

```
Java
public static boolean powerOfTwo(int x) {
    return (x & (x - 1)) == 0;
}
```

$x \& (x-1)$ 是一个很棒的技巧，可以将最右边的为1的位设为0。我是从《高效程序的奥秘》这本书中学到的。

40. 如何在UNIX上找到一个正在运行的Java进程？(command)

你可以组合使用ps和grep命令来查找UNIX机器上的任何进程。假设你的Java进程有名字，或者有任何可以用来匹配的文字，那么使用这个命令。

```
ps -ef | grep "myJavaApp"
```

ps -e将列出所有的进程（所有用户的进程，不只是你的），ps -f将显示所有细节，包括PID。如果你想要深入调查或用kill命令杀死这个进程，你会需要PID。

41. 如何在UNIX中寻找大的文件，例如1GB以上的文件？(command)

你可以轻松地用find命令寻找大的文件，因为它提供依据大小寻找文件的选项。如果你的文件系统满了，你的Java进程因为没有空间而崩溃，那么就使用这个命令。这个命令可以列出所有大于1GB的文件。你可以很容易地改变大小，例如寻找所有100MB以上的文件，就用+100M。

```
find . -type f -size +1G -print
```

42. shell脚本是什么？

shell脚本是包含程序元素（例如if和for循环）的一组shell命令，它可以自动做一些重复的任务。例如，你可以写一个shell脚本来每天清理日志文件，为记录历史备份数据，以及其他家务活、版本发布、监视等等。