 对于任何数据库来说，报表应用是其不可缺少的一部分。在SQLServer数据库中，提供了一个帮助管理员设计、创建、管理报表的工具，即报表服务器。通过这个报表服务器可以让报表的创建更加的简单，安全更加有保证。不过这个报表工具能否发挥其应有的作用，最好还是要看报表工具的使用者，是否掌握了相关的使用技巧。笔者这次要跟各位读者讨论的就是，在SQLServer数据库如何正确使用带参数的报表。 一、带报表参数的典型应用。 在一个报表中加入参数，最直接的结果就是可以提高查询语句的重复利用性。如用户可以通过更改参数来调整显示的结果等等。对于这些常规的应用笔者不做过多的阐述。笔者现在要说的是，带参数报表的一些高级应用。 参数报表比较高级的应用就是实现报表钻取。钻取是改变维的层次，变换分析的粒度。它包括向上钻取和向下钻取。向上钻取是在某一维上将低层次的细节数据概括到高层次的汇总数据，或者减少维数；向下钻取是指自动生成汇总行的分析方法。简单的说，现在数据库中有一张销售订单表。根据这张表可以生成一张各个月份的销售统计表。但是，有可能用户在查看这张报表的时候，对某个月份的统计结果有怀疑，为此需要查看这个月份的销售明细。此时如果利用带参数的报表实现钻取功能的话，那么就不需要重新查询或者生成报表。而只需要直接在这张报表上点击月份，系统就会自动打开另外一张报表。这张报表中的内容就是这月份的销售明细。从技术的角度讲，就

是通过参数的传递，将这张报表的时间信息作为另一张报表的查询参数。从而让系统自动根据这个参数来生成相应的数据，从而简化用户的操作。 二、带参数报表要避免注入式攻击。 在使用参数报表的时候，特别需要注意一点就是防止注入式攻击。注入式攻击各位读者或许都了解。可是对于为什么使用参数的报表容易引起注入式攻击，可能大家并不怎么了解。这主要是因为参数如果采用的是string数据类型所造成的。即如果参数采用的是string数据类型，那么就表示用户可以根据需要输入任何类型的字符串。此时如果用户输入了一些注入式攻击的代码当作参数，则就可能会导致注入式攻击。为此如果生成报表时，采用的参数时String数据类型的，就需要特别的注意。为了防止这个注入式攻击，笔者建议如果采用的参数一定要是String数据类型的话，那么最好能够遵循下面的规则。DB2数据库与SQLServer数据库的异同。 首先，在客户端将报表查询语句传递给数据库之前，即将参数复制给Select语句之前，最好进行验证。即要验证输入的参数值中，是否存在一些特殊的符号。这些符号往往跟输入攻击有关。如果存在这些特殊字符的话，则需要向用户提供警告信息，表明存在注入式攻击的可能性。并且，系统可以拒绝接受这个参数。这个避免注入式攻击的方法比较消极。如果这些特殊符号确实是查询参数中包含的内容，那么也无法使用。其次，可以通过值列表的方式来向数据库传递参数。在没有提供值列表的情况下，如果参数是字符类型的，则系统向用户显示的是一个可以使用任何值的文本框。此时数据库管理员可以使用可用值列表的方式来规范化参数的输入，限制其输入一些特殊的字符。也就是说，在定义 String类型的参数报

表时，让系统向用户显示一个下拉的列表框，然后用户通过选择来指定参数。这个操作就跟Excel表格中的下拉列框差不多，用户只能够选择数据库管理员所提供的值，或者说只能够选择某张表中存在的值。由于用户不能够自己输入值，而只能够选择，这就可以有效的避免注入式攻击。不过采用这种方式有一个缺陷，就是如果有效的值太多的话，这个列表就会很长。为此用户在选择参数的时候，就会很麻烦。如当有效值有500个的话，那么就需要在 500个值中选择一个值，显然这有点困难。即使按照参数的名字顺序来排列，选择也是比较麻烦的。大内存SQLServer数据库的加速剂 第三，可以利用列表查询的方式，来避免注入式攻击。即当用户输入一个参数之后，系统会自动从一个列表中查询是否存在这个值。如果存在的话，则将这个参数赋值给查询语句中的变量。如果不存在的话则提醒用户参数可能输入错误。如现在有一张销售订单明细报表。用户可能需要根据订单号码来查询销售订单明细。此时这个订单号码就是一个字符型的参数。当用户输入这个参数的时候，并不是马上传递给数据库，这么做太危险，容易产生注入式攻击。而是前台应用程序也从后台数据库中取得所有的销售订单的订单号码信息。当用户输入参数之后，前台应用程序会把这个用户输入的参数跟自己查询出来的信息先进行对比。如果有匹配的信息，就将这个参数传递给后台数据库。如果没有的话，就向用户报告错误的信息。有些应用程序在设计的时候，还会更进一步。如客户端程序会先从数据库中取得订单号码与对应的订单ID。当用户输入参数之后，会进行比对。如果比对成功的话，那么客户端应用程序会将这个订单号码对应的订单ID作为参数传

递给查询语句。也就是说，从数据库服务器角度来讲，真正的参数是订单ID（整数型数据类型）而不是订单号码（字符串数据类型）。通过这个数据类型转换，从而可以从根本上防止注入式的攻击。 以上三种方式都可以很有效的避免注入式攻击。数据库管理员需要根据实际应用来选择合适的解决方案。如当有效值比较少的时候，如按年份来统计销售订单时，则可以使用列表的形式。当有效值比较多，特别是这个有效值会自动增长的时候，则可以使用列表查询的方式。总之一个基本的原则，对于String参数，一定要进行验证其合法性。否则的话，很容易造成注入式攻击。 三、对于日期型的数据给与特殊的照顾。 日期型的数据是数据库中最容易出现问题的一个数据类型。因为不同语言环境下，如英语与汉语环境下，其采用的日期格式是不同的。如果数据库中定义了某个日期格式，而输入的参数如果不符合这个格式的话，则系统就会认为这条记录不存在，从而在报表中查询不到相关的数据。为此如果在报表中要使用日期型数据参数的话，将会是一件比较麻烦的事情。所以，在应用程序设计时，数据库管理员最好提醒前台应用程序的设计者，能够规范化日期的格式。如可以要求他们，对于日期型的数据作为参数时，用户不能够手工输入日期。因为不同的用户输入习惯不同，如有些人会按年月日的格式输入（有些用户会把8月份写成08，而有些直接写成8），有些人则会按月、日、年的格式进行输入。由于格式不统一，那么数据库就很难按照同一个规则进行转换。为此，对于日期型的数据作为参数时，最好在前台应用程序中能够规范化输入的格式。如以一个统计的格式输入。要做到这一点的话，就可以通过一个日期型的控件来

完成。即用户不能够手工输入日期型的数据。当遇到某个参数时日期型的数据时，当鼠标定位到这个文本框，则系统就会弹出一个类似日历的界面。用户只有通过选择日期来输入日期型的数据，从而规范化用户的输入。另外也可以通过掩码的方式来规范用户输入的格式。即预先规定年月日的输入掩码。用户在输入的时候必须按照这个格式，否则的话，系统不会接受用户的输入。这两种方式都可以实现对日期数据的规范化。 当用户按照同一个格式输入日期数据后，以后的工作就容易处理了。在将参数传递给数据库的时候，可以在查询语句中加入一个日期型数据的强制转换语句。将输入的日期型数据按照系统表中定义的日期型数据进行转换。即如果前台客户端输入的日期型数据格式是日、月、年（只要输入的内容统一即可，没有具体的要求），然后在查询语句中就可以通过数据类型转换工具对数据类型进行转换。如将日、月、年表示的字符型数据类型表示会年、月、日的日期型数据类型。如此的话，就可以保证用户输入的参数是数据库可以识别的。就可以避免因为日期格式不一致或者数据类型不一致而导致报表不能够抓取记录。 编辑特别推荐: 全国计算机等级考试（等考）指定教材 全国计算机等级考试学习视频 全国计算机等级考试网上辅导招生 全国计算机等级考试时间及科目预告 百考试题教育全国计算机等级考试在线测试平台 全国计算机等级考试资料下载 全国计算机等级考试论坛 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com