

ADO.NET 实现主从表更新的一种方法

Realization to Update on Master - Slave Table in ADO.NET

陆锡聪 (韩山师范学院数学与信息技术学院 广东潮州 521041)

摘要:本文详细介绍了在 ADO.NET 数据访问机制下实现主从表更新的一种方案,并提供 C#语言对本方案的实现过程,本方案具有现实意义。

关键词:ADO.NET 主从表 .NET

1 ADO.NET 数据访问原理

ADO.NET 是全新的面向对象模型,适应于分布式及 Internet 等大型应用程序环境,是专为 .NET 平台设计的数据访问机制。它具有支持 OLE DB、离线存取、XML 数据交换格式的特点。ADO.NET 能有效地从数据操作中将数据访问分解为多个可以单独使用或一前一后使用的不连续组件。数据集(DataSet)和 .NET 数据提供程序是 ADO.NET 的两个核心组件,其中后者包含 Connection、Command 和 DataAdapter 等组件,ADO.NET 体系结构如图 1 所示。

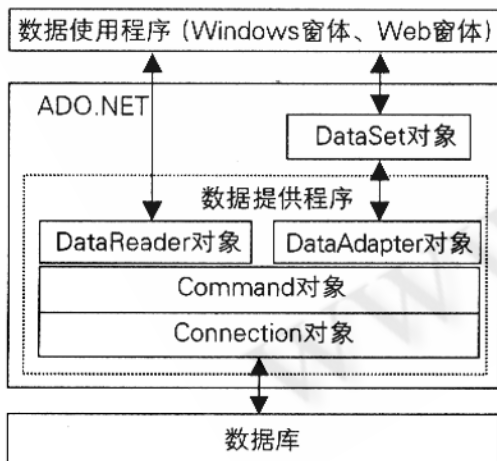


图 1 ADO.NET 体系结构

DataSet 使 ADO.NET 实现独立于任何数据源的数据访问,我们可以把不同数据源的数据读取到 DataSet 中,也可以把各种操作(插入、删除、更改)后的数据更新到对应的数据源。DataSet 中包含 DataTable 对象,

表示数据表的集合;DataTable 对象又包含 DataRow 对象,表示数据行的集合,DataRow 能够保留新增、更改、删除等状态(DataRowState 属性),也能够通过当前版本和初始版本,以标识对行中存储的值的更改。

Connection 对象提供与数据源的连接,Command 对象能够访问用于返回数据、修改数据、运行存储过程以及发送或检索参数信息的数据库命令。DataAdapter 提供连接 DataSet 对象和数据源的桥梁,它使用 Command 对象在数据源中执行 SQL 命令,以便将数据加载到 DataSet 中,并使对 DataSet 中数据的更改与数据源保持一致。

2 主从表更新的问题

主从表结构在数据应用开发中具有普遍性和代表性。一般地主表存储共用数据,从表存储明细数据。根据关系型数据库规范化设计原理,在一般应用中,数据库设计通常要满足第三范式(3NF),这样我们在设计应用程序时,必须要解决主从表结构(一对多表)的实现问题,特别地,当具有严格的约束关系时,主从表数据更新时经常由于不满足约束条件而发生错误。

例如,要求主从表满足关系数据库完整性规则:

(1) 实体完整性,要求主表中主键不准为空。

(2) 参照完整性,要求子表中外键的值必须与主表中的主键对应。

大多数主从表均具有上述完整性规则。在 ADO.NET 中,如果 DataSet 存储主从表数据,并且直接调用 DataAdapter 中的 Update 方法时,有可能不满足完整

性规则而返回错误,原因就在于 Update 方法不会遵守“数据库完整性规则”把数据更新到数据库中。

3 主从表更新的方案及实现

本主从表的更新方案为以下 3 点规则,要求按规则顺序执行。

(1) 在 DataSet 更新之前按每一张表和表的行状态分别由 DataAdapter 提交,即主表与从表分别提交,每表再根据“DataRowState”属性确定的三种状态分组提交。

(2) 在(1)基础上,先更新状态为“Added”(新增)和“Modified”(修改)的数据行,再更新“Deleted”(删除)的数据行。

(3) 在(1)和(2)基础上,如果数据行状态为“Added”(新增)和“Modified”(修改),则先更新主表,后更新子表;如为“Deleted”(删除),则先更新子表,后更新主表。

不失一般性,这里假定 dsDataSet 为存储待更新的数据集,其中包含主表 dtMaster 和从表 dtSlave,参照上述方案,以 Visual C#语言实现 dsDataSet 更新的步骤分解如下。

(1) 记录分组,即根据 dsDataSet 中各表的 DataRowState 将各个表记录进行分组并存储在不同的数据集(dsAdded, dsModified, dsDeleted)中,代码如下。

```
DataSet dsAdded = dsDataSet. GetChanges ( DataRowState. Added );
```

```
DataSet dsModified = dsDataSet. GetChanges ( DataRowState. Modified );
```

```
DataSet dsDeleted = dsDataSet. GetChanges ( DataRowState. Deleted );
```

(2) 更新主表中 DataRowState 为“Added”和“Modified”的记录。

```
SqlDataAdapter. Update ( dsAdded, " dtMaster" );
SqlDataAdapter. Update ( dsModified, " dtMaster" );
```

这里假定数据源为 Microsoft SQL Server, SqlDataAdapter 为已经实例化的 DataAdapter 对象,包含用于

填充数据集 dsDataSet 和更新 SQL Server 数据库的一组数据命令和一个数据库连接,以下 SqlDataAdapter 与此相同。

(3) 更新从表中 DataRowState 为“Added”和“Modified”的记录。

```
SqlDataAdapter. Update ( dsAdded, " dtSlave" );
```

```
SqlDataAdapter. Update ( dsModified, " dtSlave" );
```

(4) 更新从表中 DataRowState 为“Deleted”的记录。

```
SqlDataAdapter. Update ( dsDeleted, " dtSlave" );
```

(5) 更新主表中 DataRowState 为“Deleted”的记录。

```
SqlDataAdapter. Update ( dsDeleted, " dtMaster" );
```

笔者把本方案分解为 5 个实现步骤,比较完美地实现了主从表的数据更新。

4 结束语

主从表在数据库应用开发中具有一定的普遍性,如订单,成品入库单,发货单,材料入库单,领料单等,均具有主表(共用数据)与从表(明细数据)结构,应用本方案可以比较好地解决了数据更新问题,在软件开发中具有现实意义。

参考文献

- 1 王丽等,基于.NET的通用数据访问层的设计与实现,哈尔滨师范大学自然科学学报[J],2005,21:51~54.
- 2 孔延香等,ADO.NET数据库访问技术研究,现代电子技术[J],2005,16:51~53.
- 3 李用江,分布式应用程序中数据访问技术的剖析,计算机工程与设计[J],2005,7:1882~1885.
- 4 杨晓能,多层数据库系统开发中主从表结构的实现,山西科技[J],2004,2:43~45.
- 5 Paul Dickinson, Fabio Claudio Ferracchiati 著,谭建波、李立新译,ADO.NET高级编程[M],北京:清华大学出版社,2003.