[0306] RSD File:

```
<Database Name="Northwind">
    <Schema Name="dbo">
        <CustomTable Name="myCustomers">
            <Columns>
                <Column Name="CustomerID"
                SqlType="nvarchar" Length="10"/>
                <Column Name="CustomerName"
                SqlType="nvarchar" Length="20"/>
                <Column Name="Address"
                SqlType="nvarchar" Length="20" />
                <Column Name="City"
                SqlType="nvarchar" Length="20" />
                <Column Name="State"
                SqlType="nvarchar" Length="20" />
                <Column Name="Comments"
                SqlType="nvarchar" Length="max" />
            </Columns>
            <QueryCommand>
                <InlineCommand>
                    <Body>Select * From Customers Where
                    state='WA'</Body>
                </InlineCommand>
            </QueryCommand>
            <InsertCommand>
                <CommandReference
                Name="sp_CustomerInsert"/>
            </InsertCommand>
            <UpdateCommand>
                <InlineCommand>
                    <Body>UPDATE Customers SET
                    CustomerName=@CustomerName,
Address=@Address, City=@City, State=@State,
Comments=@Comments WHERE
CustomerID=@CustomerID</Body>
                </InlineCommand>
            </UpdateCommand>
            <DeleteCommand>
                <CommandReference Name="sp_CustomerDelete"/>
            </DeleteCommand>
            <CustomKey Name="mycust_pk">
                <ColumnRef Name="CustomerID"/>
            </CustomKey>
        </CustomTable>
    </CustomTables>
</Database>
```

[0307] ExistingSP

[0308] This example is the same as the InlineCommand example, but with YTDSales on the Customer. The difference here is that the Stored Procedures are not written as part of the application mapping is developed for. The following features are therefore introduced: explicit binding of parameters; partial binding of parameters; SPs return resultsets—Insert returns the CustomerID that can be propagated back to the target. This result set also makes use of partial binding.

[0309] RSD File:

```
    <Database Name="Northwind">
        <Schema Name="dbo">
            <CustomTable Name="myCustomers">
                <Columns>
                    <Column Name="CompanyName"
                    SqlType="nvarchar" Length="20"/>
                    <Column Name="CustomerName"
                    SqlType="nvarchar" Length="20"/>
                    <Column Name="CustomerID"
                    SqlType="nvarchar" Length="10"/>
```

-continued

```
                    <Column Name="CustomerYTD"
                    SqlType="decimal"/>
                </Columns>
                <QueryCommand>
                    <InlineCommand>
                        <Body>
                        Select CustomerId, CompanyName,
                        CustomerName, YTDSales AS
CustomerYTD from udf_AllCustomers( )
                        </Body>
                    </InlineCommand>
                </QueryCommand>
                <InsertCommand>
                    <CommandReference
                    Name="sp_CustomerInsert"/>
                    <ParameterBinding Parameter="cid"
                    Column="CustomerID" />
                    <ParameterBinding Parameter="cname"
                    Column="CustomerName" />
                    <ColumnBinding Column="NewCustomerID"
CustomColumn="CustomerID"/>
                </InsertCommand>
                <UpdateCommand>
                    <ComandReference
                    Name="sp_CustomerUpdate"/>
                    <ParameterBinding
                    Parameter="OriginalCustomerIDToUpdate"
CustomColumn="CustomerID" Version="Original"/>
                </UpdateCommand>
                <DeleteCommand>
                    <CommandReference
                    Name="sp_CustomerDelete"/>
                </DeleteCommand>
                <CustomKey Name="mycust_pk">
                    <ColumnRef Name="CustomerID"/>
                </CustomKey>
            </CustomTable>
        </Schema>
    </Database>
```

[0310] In this example, the QueryCommand takes advantage of default Column Binding by aliasing the column returned from the udf_AllCustomers ( ) procedure to match the definition in the Columns section of the CustomTable

[0311] The InsertCommand uses an explicit binding (assuming that the CustomerYTD is not bound, since the InsertCommand's parameters have been declared in the SP definition).

[0312] The UpdateCommand uses a partial binding, to bind the parameter for the original key value in the update, but allows the current value to default to the appropriately-named parameter (CustomerId) that is assumed to be declared in the SP definition. The rest of the fields to be updated are also default bound.

[0313] Serialization

[0314] Custom Tables will be serialized inline with the RSD file under the <CustomTables> element which resides underneath the <Schema> element. As noted above, Custom Tables belong to the RSD-extensions namespace, providing a means for tools which generate/refresh RSD documents from the database to know explicitly which elements they should ignore (as long as those elements are preserved "as-is" by the tool). When generating/refreshing the RSD from a database, Custom Table elements may be invalidated based on metadata changes within the server.

[0315] Referring now to **FIG. 7**, there is illustrated a block diagram of a computer operable to execute the disclosed