

Table-valued Parameters and Refactoring



Mike McQuillan

ARCHITECT, DEVELOPER, DATA SPECIALIST

@mcqtech www.mcqtech.com



Table-valued Parameters and Refactoring

**Refactoring a
Stored Procedure**

Table-valued Type

**Calling a Table-valued Type
Stored Procedure**

**(Bad) Alternatives to
Table-valued Types**



Problems



Grace



Insert Notes Procedure



String Split Function

Hello,Goodbye



Hello **Goodbye**



Using a Cursor



Declare cursor



Open cursor



Try to obtain the first record



Check if record found



Process record if found



Obtain next record



Check if record found



Repeat until all records read



Close cursor



Deallocate cursor

```
CREATE TYPE <name of type>  
FROM <base type>;
```

```
CREATE TYPE <name of type>  
AS TABLE  
(  
    Column definitions here...  
);
```

T-SQL Create Type Statement

Use to create aliases for primitive types, or to create custom data types



```
DROP TYPE IF EXISTS <name of type>;
```

T-SQL Drop Type Statement

Use to drop types

A type can only be dropped if no other objects depend on it



User-defined Table Types

```
DECLARE TYPE TableType  
AS TABLE  
(  
    Col1 INT,  
    Col2 VARCHAR(20)  
);
```



```
DECLARE @Table TABLEType;  
(  
    Col1 INT,  
    Col2 VARCHAR(20)  
);
```

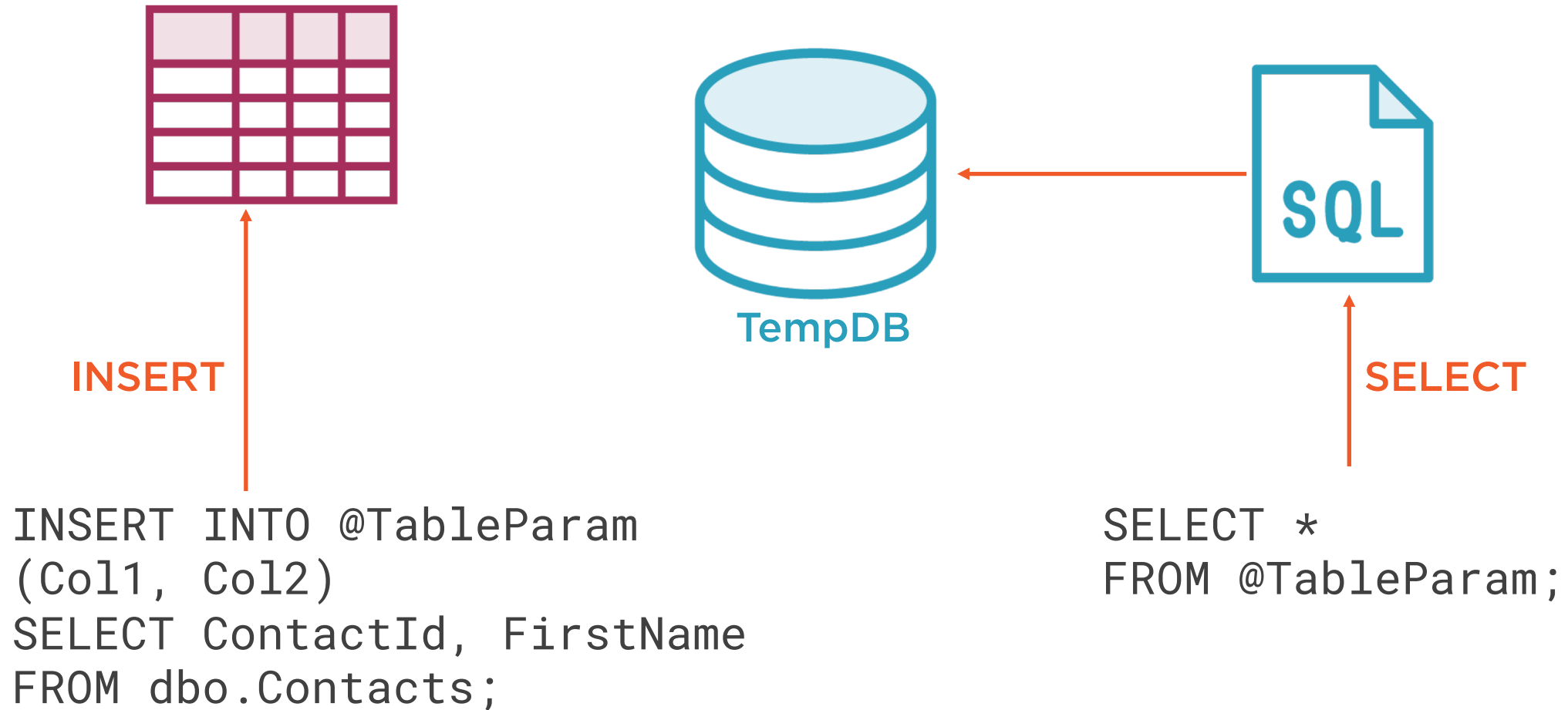


Table-valued Parameters

**[https://docs.microsoft.com/en-us/
sql/relational-databases/tables/
use-table-valued-parameters-
database-engine?
view=sql-server-2017](https://docs.microsoft.com/en-us/sql/relational-databases/tables/use-table-valued-parameters-database-engine?view=sql-server-2017)**



How TVP Data Are Stored



Summary



Passing multiple records - badly

Don't use cursors

While loops

Table-valued Parameters in stored procedures