

Creating Stored Procedures and Using Parameters



Mike McQuillan

ARCHITECT, DEVELOPER, DATA SPECIALIST

@mcqtech www.mcqtech.com



Creating Stored Procedures and Using Parameters

**Creating a Simple
Stored Procedure**

Adding Parameters

**Optional and
Output Parameters**

**Returning Data and System
Options in Stored Procedures**



The First Stored Procedure



Grace

```
CREATE PROCEDURE  
dbo.SelectAllContacts  
AS  
BEGIN;  
SELECT * FROM dbo.Contacts;  
END;
```



Insert Contact Procedure Requirements



Insert contact



Return details



Return ID



No duplicates



```
public int InsertContact(string name)
{
    string sql = $"INSERT INTO dbo.Contacts (Name) VALUES
    ('{name}')";

    return ExecuteSql(sql);
}
```

Embedding SQL in Program Code

A bad idea!

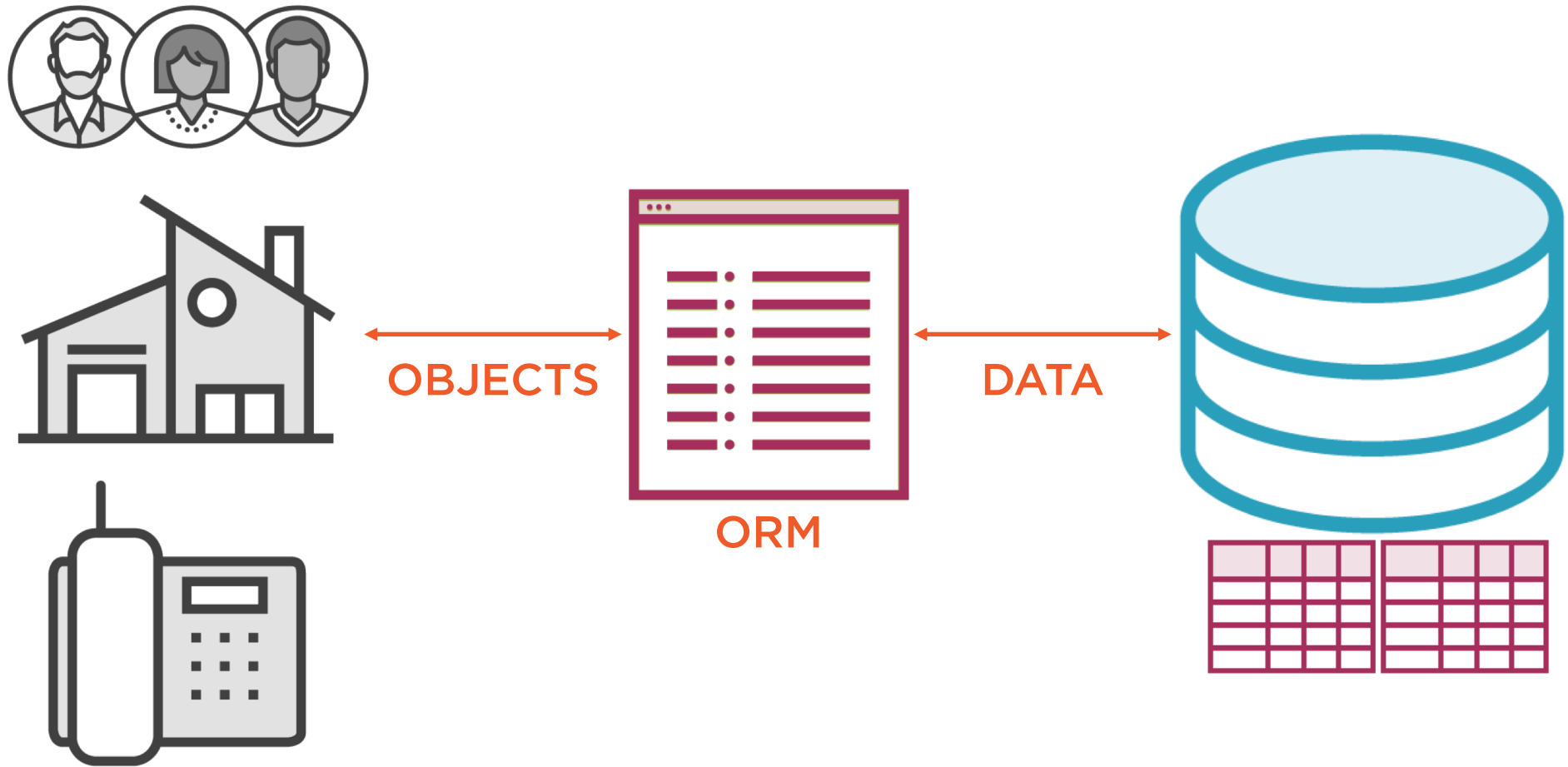


Embedded SQL Limitations

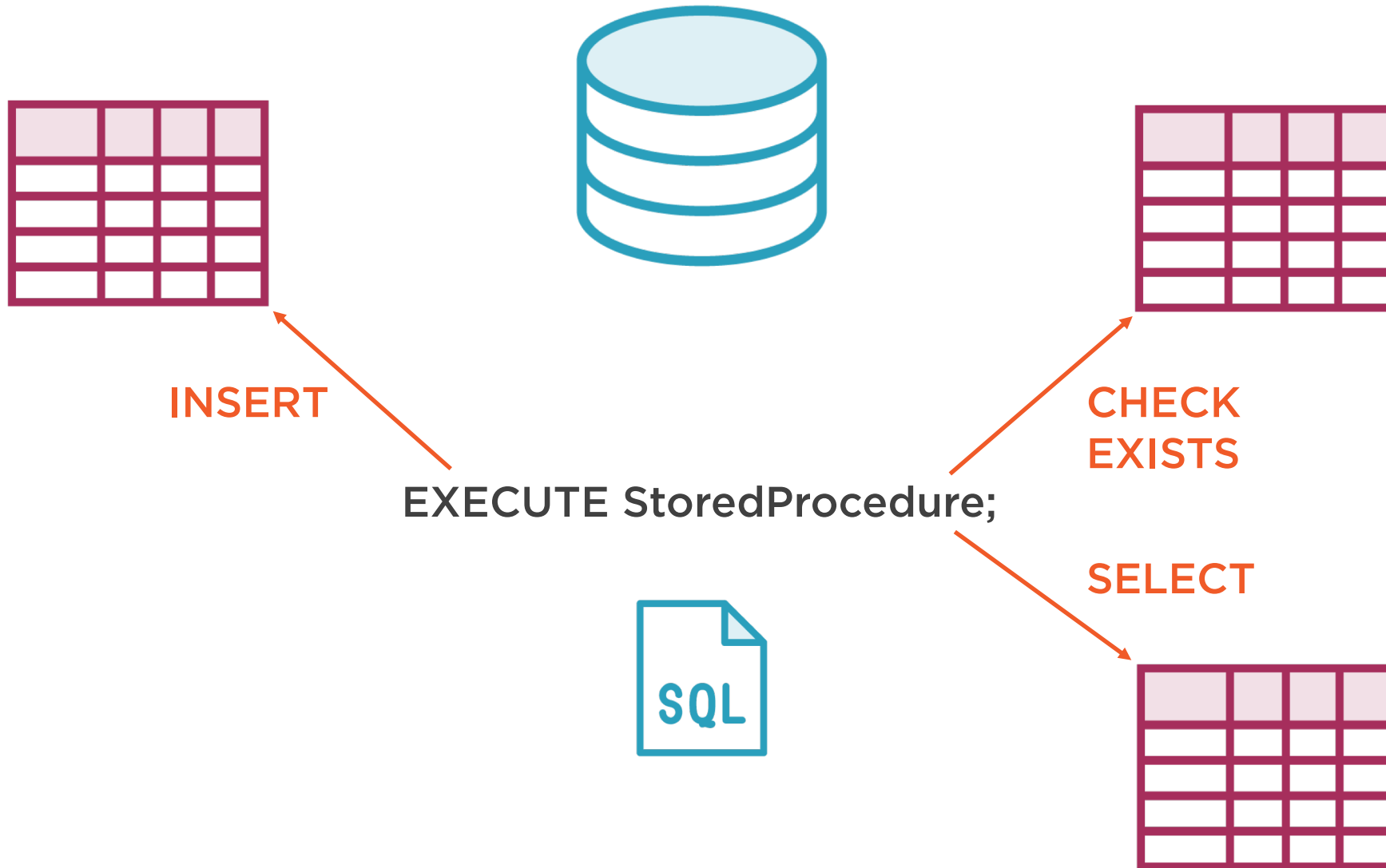


```
public int InsertContact  
(string name)  
DateTime dateOfBirth,  
bool allowContact)
```

Object-relational Mapping



The Stored Procedure



Parameters



Strings



Numbers



Dates, XML



Custom types



Insert Contact Parameters



First Name
Last Name



Date of Birth



Allow Phone
Contact

@ParameterName



```
CREATE PROCEDURE <name of procedure>  
(Parameters (optional))  
AS  
BEGIN;  
  
Statements here...  
  
END;
```

T-SQL Create Procedure Statement

Used to create new stored procedures



Optional Parameters

**@ParameterName =
default**



Insert Contact Procedure Requirements



Insert contact



Return details



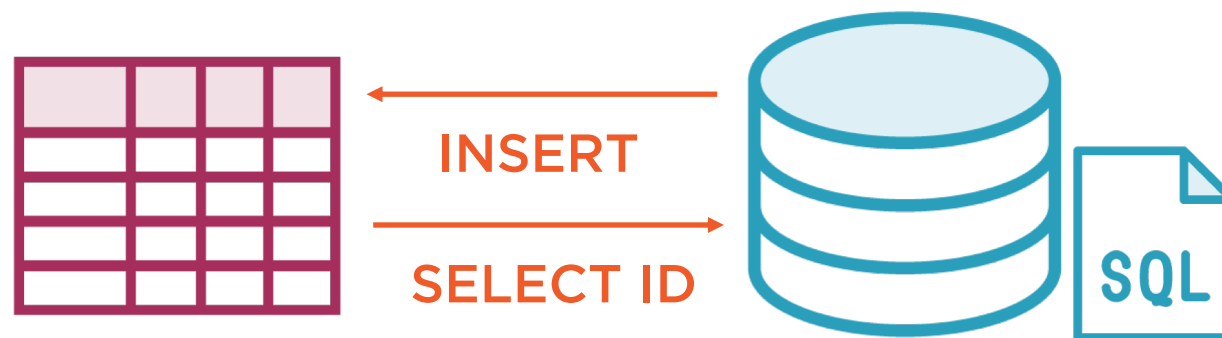
Return ID



No duplicates



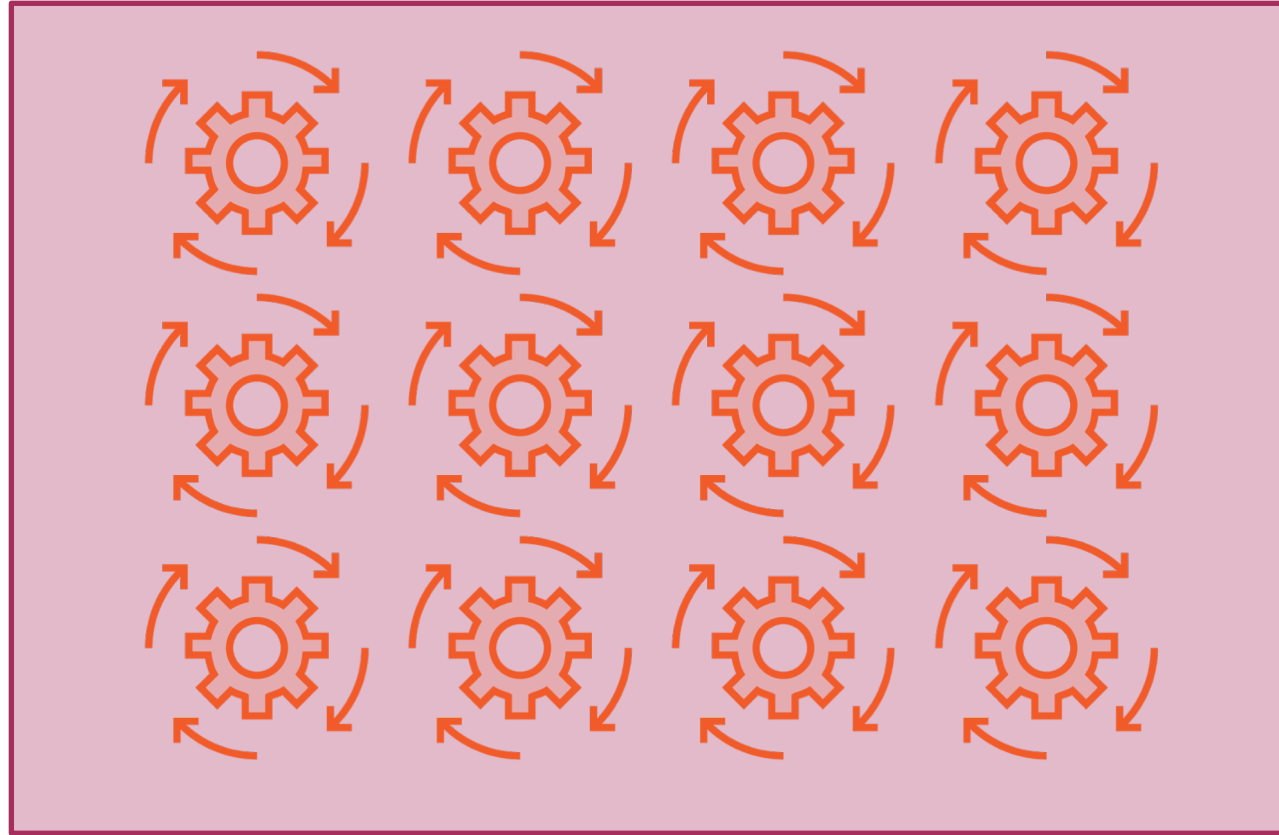
Returning Record ID



Stored Procedure Size



Stored Procedure Operations



Obtaining Contact ID



Grace



Identity column

Identity Columns

				1
				2
				3
				4

ContactId INT IDENTITY(1,1)

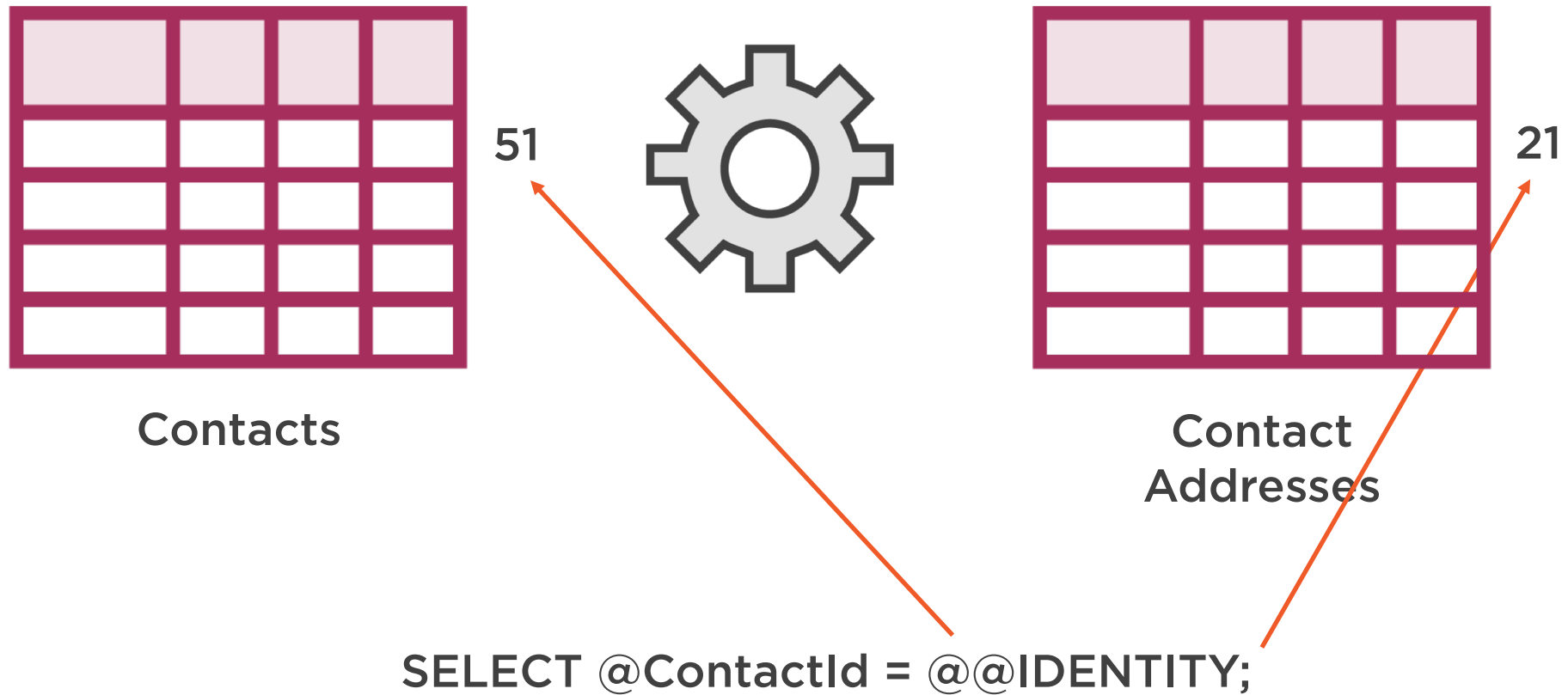


SQL Server System Functions

<https://docs.microsoft.com/en-us/sql/t-sql/functions/system-functions-transact-sql?view=sql-server-2017>



Identity Processing



@@IDENTITY

<https://docs.microsoft.com/en-us/sql/t-sql/functions/identity-transact-sql?view=sql-server-2017>



```
DECLARE @ContactIdOut INT;  
  
EXEC dbo.SelectContactId  
@ContactId = @ContactIdOut OUTPUT;  
  
SELECT @ContactIdOut;
```

Output Parameters

Assign output to holding variable



SET Options

<https://docs.microsoft.com/en-us/sql/t-sql/statements/set-statements-transact-sql?view=sql-server-2017>



Less Network Traffic

EXECUTE 

RESULT

METADATA



Control Flow



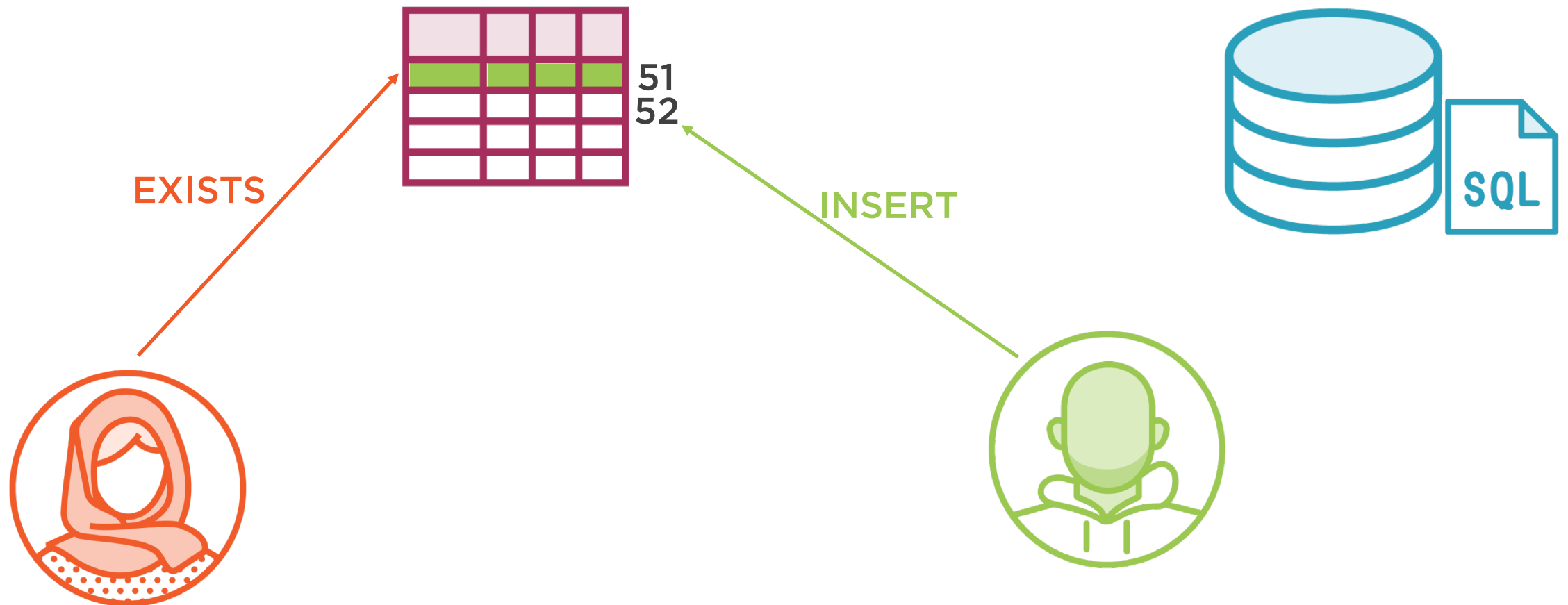
IF statement



WHILE loop



Contact Exists Business Logic



Summary



Implementation options

Parameter-less stored procedures

Mandatory and optional parameters

Retrieving identity values

Nesting stored procedures

