Maintain a Database Project and CD using SQL Server Data Tools in practical terms

Kamil Nowiński

Principal Microsoft Consultant

altius

SQL Player
Play with data & have fun!

Kamil Nowiński

Microsoft Data Platform **MVP**
Speaker, blogger, data enthusiast
Principal Microsoft Consultant at Altius (www.altiusdata.com)
Almost 20 yrs experience as DEV/BI/(DBA)
Member of the Data Community PL
Project member of „SCD Merge Wizard"
Founder of blog SQLPlayer (www.SQLplayer.net)

SQL Server Certificates:
MCITP, MCP, MCTS, MCSA, MCSE Data Platform,
MCSE Data Management & Analytics
Moreover: Bicycle, Running, Digital photography
@NowinskiK, @SQLPlayer

Blog

SQL Player
Play with data & have fun!

www.SQLPlayer.net

- Technical posts
- Various skill level
- Cheet sheets
- Recommended books
- Many useful other links
- Interviews (Podcast)
- YouTube Channel

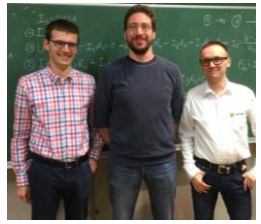
Scan me

“Ask SQL Family” #podcast



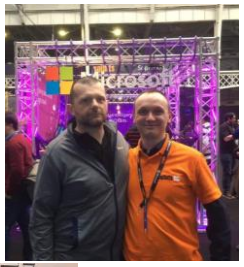


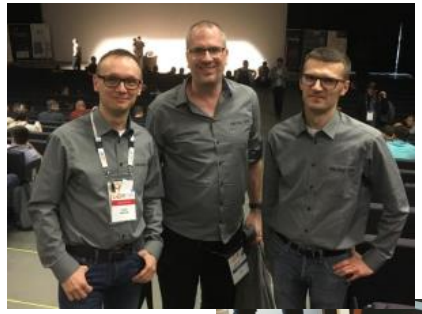


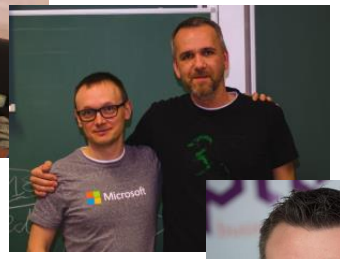


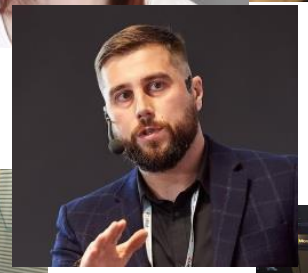
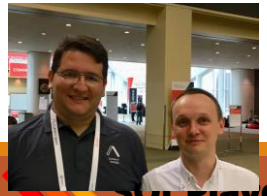






Scan me

YouTube

www.SQLPlayer.net/YouTube

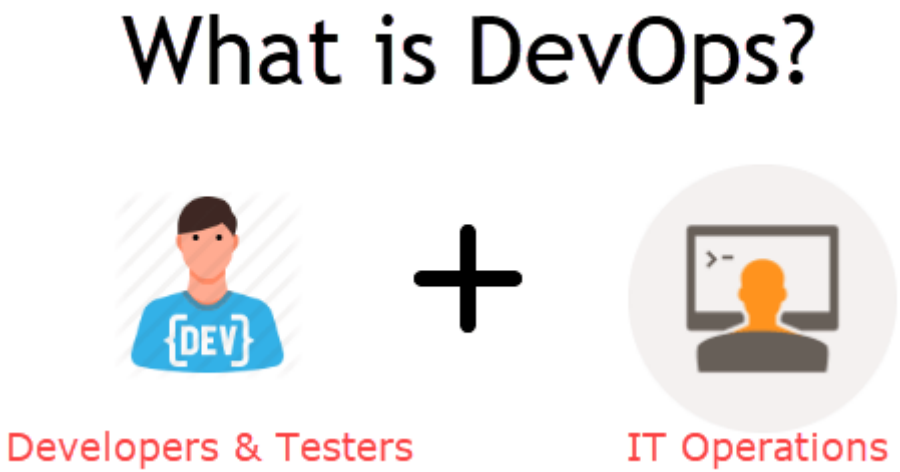
Slides available

github.com
/NowinskiK/CommunityEvents

DevOps – definition (Wikipedia)

DevOps (a clipped compound of "**development**" and "**operations**") is a software engineering **culture and practice** that aims at unifying software development (Dev) and software operation (Ops). The main characteristic of the DevOps movement is to strongly advocate **automation and monitoring** at **all steps** of software construction, from integration, testing, releasing to deployment and infrastructure management.

DevOps aims at shorter development cycles, increased deployment frequency, and more dependable releases, in close alignment with business objectives.



Redgate – State of database DevOps 2020

Yes

36% Adopted across some projects

18% Adopted across all projects

14% In Proof of Concept or experimentation

Not yet

18% Planning to adopt within two years

13% No plans to adopt within two years

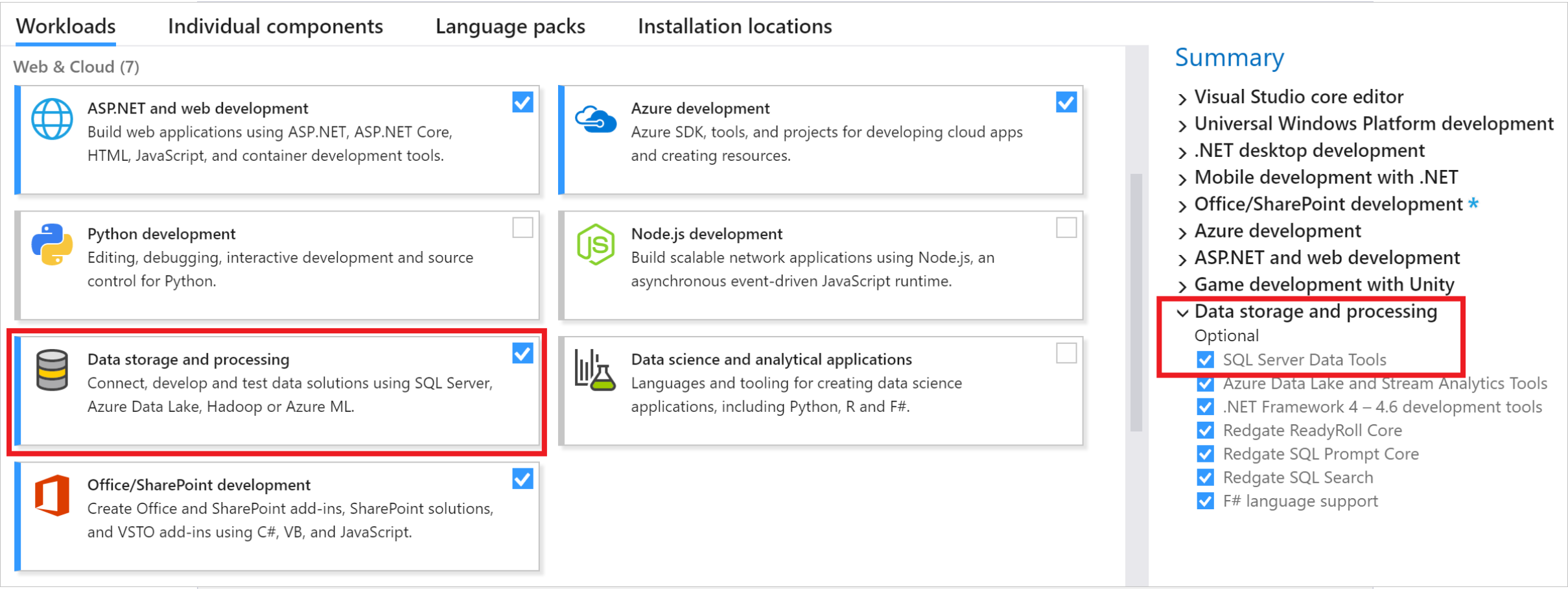
DevOps – how to start?

- Database in source control
- Build CI process
- Deploy (CD) empty database
- Then add the rests:
  - data,
  - security,
  - Unit tests
  - Integration-test (if required)

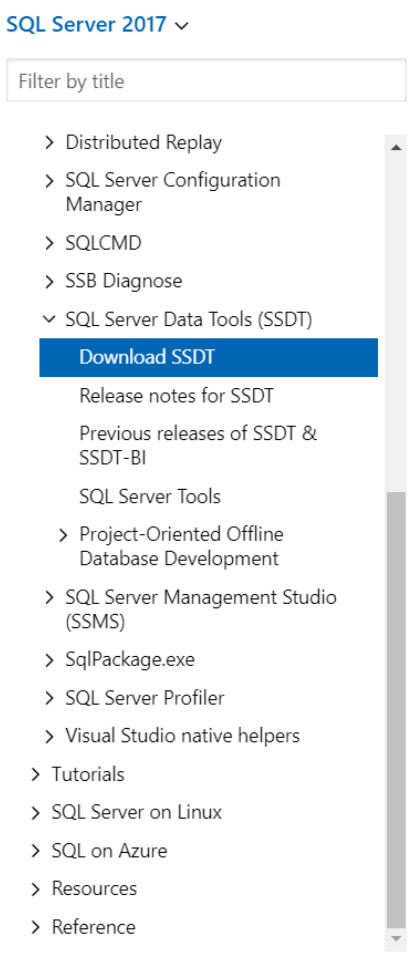
What is the SSDT?

- SQL Server Data Tools
- Now, it’s a part of Visual Studio
- Free!
- Database project, including:
  - Schema, Stored Procedures, Functions,
  - Tables, Views, Security, CLR
  - ... and much more!

SSDT: How to install?



SSDT: How to download, install?



Changes in SSDT for Visual Studio 2019

With Visual Studio 2019, the required functionality to enable Analysis Services, Integration Services, and Reporting Services projects has moved into the respective Visual Studio extensions. The core SSDT functionality to create Database Projects has remained integral to Visual Studio (you need to select the Data storage and processing workload during install). There is no more standalone SSDT installation required.

If you already have a license to Visual Studio 2019:

- For SQL Database Projects, install the Data storage and Processing workload for Visual Studio
- For Analysis Services, Integration Services or Reporting Services projects, install the appropriate extension(s) from the marketplace

If you don't already have a license to Visual Studio 2019:

- Install Visual Studio 2019 Community
- Install the Analysis Services, Integration Services or Reporting Services as appropriate

Changes in SSDT for Visual Studio 2017

Starting with Visual Studio 2017, the functionality of creating Database Projects has been integrated into the Visual Studio installation. There is no need to install the SSDT standalone installer for the core SSDT experience. To create Integration Services/Analysis Services/Reporting Services projects you still need the SSDT standalone installer.

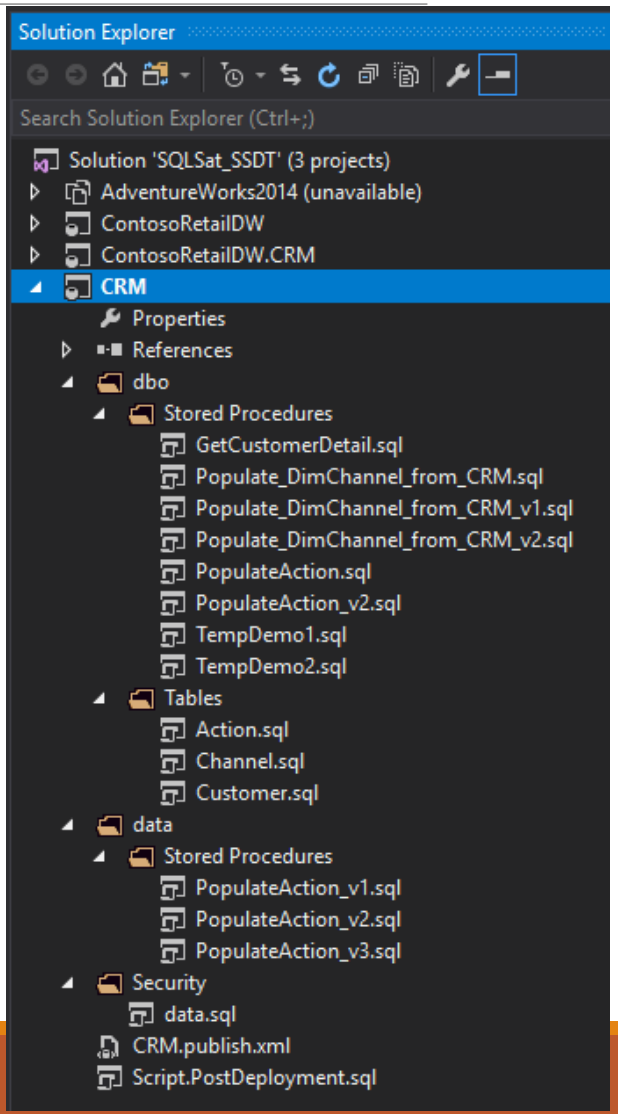
- For Database Projects, install the Data Storage and Processing workload for Visual Studio
- For Analysis Services, Integration Services or Reporting Services projects, download and install SQL Server Data Tools

DACPAC, BACPAC

- DACPAC = Data Tier AppliCation PACkage
  - Doesn’t contain DATA
  - Contains SCHEMA Only
- BACPAC = BACkup PACage
  - Contains SCHEMA
  - Contains DATA (BCP native format)
- ZIP format

How to start with database project?

- Install SSDT
- Create new db project and import:
  - From script
  - Directly from SQL server
  - Data-Tier Application (DACPAC)



DEMO

- Create empty database project
- Import database from server
- Create references & variables
- Resolve first issues
- Resolve circular references

(Well) known issues

(Well) known issues

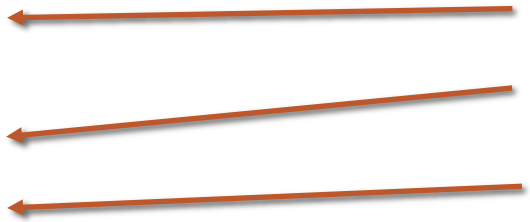
- Same database references
- Circular references/dependencies
- Unvisible temp tables
- Disable/Enable trigger across databases
- How to manage of data?
- Can I deploy SQL Jobs?

Database dependencies & references

CRM_Audit

Tables:

- Customer_Audit
- CustomerAddress_Audit
- CustomerEmail_Audit
- TransactionLog



CRM

Tables:

- Customer
- InvoiceHeader
- InvoiceLines
- CustomerAddress
- CustomerEmail

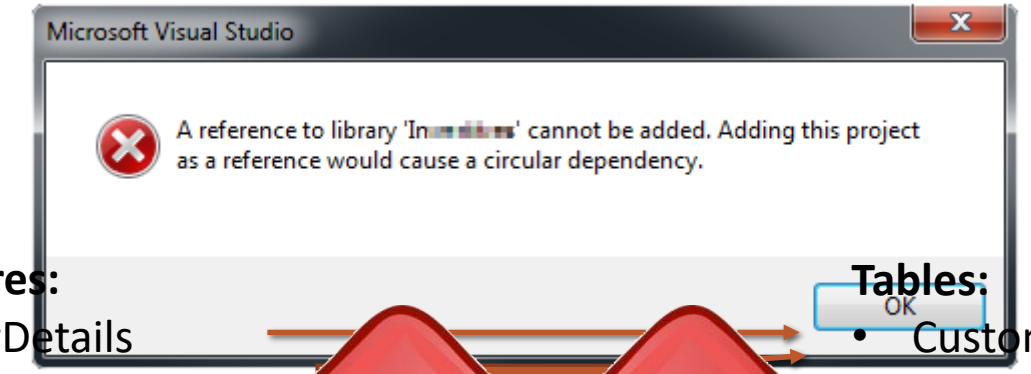
Triggers:

- Tr_Customer
- Tr_CustomerEmail

Stored Procedures:

- AddLogItem

Circular dependencies



CRM_Interface

Stored Procedures:
- GetCustomerDetails
- MergeCustomers
- AddLogItem
- GetCustomerInvoice

Tables:
- Customer_Audit
- CustomerAddress_Audit
- CustomerEmail_Audit
- TransactionLog

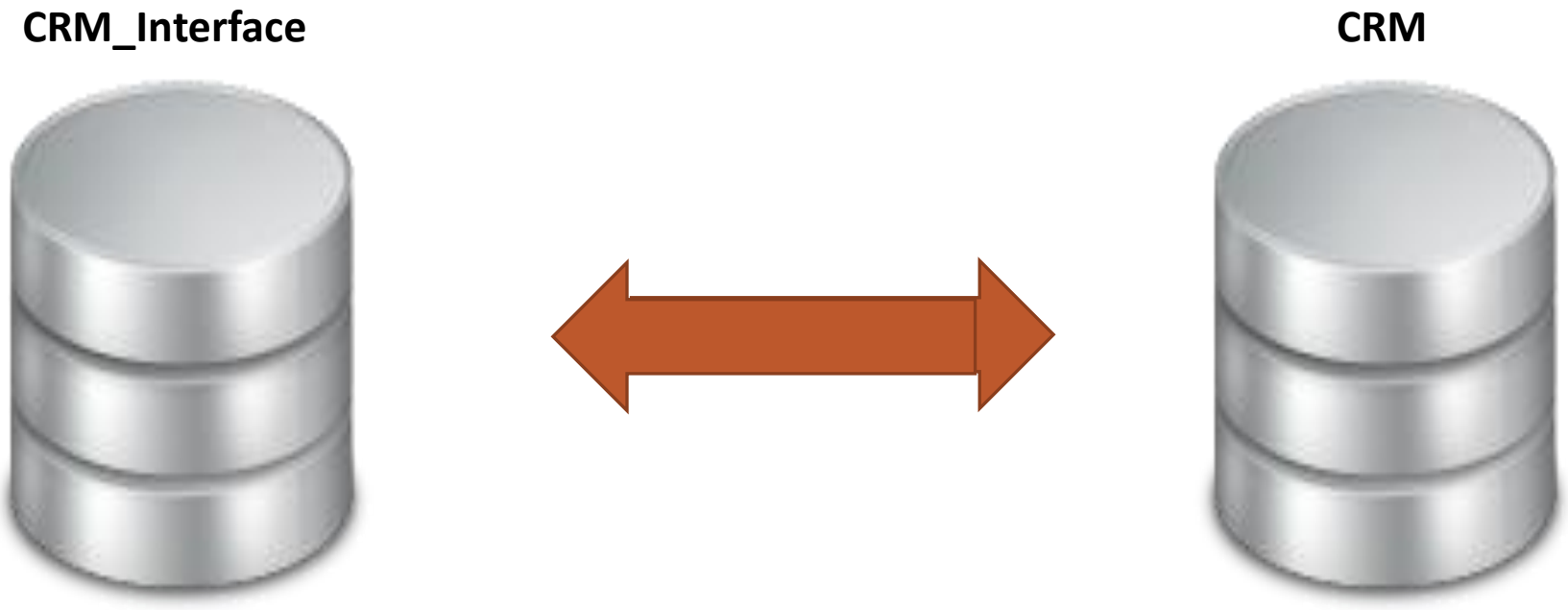
Tables:
- Customer
- InvoiceHeader
- InvoiceLines
- CustomerAddress
- CustomerEmail

Triggers:
- Tr_Customer
- Tr_CustomerEmail

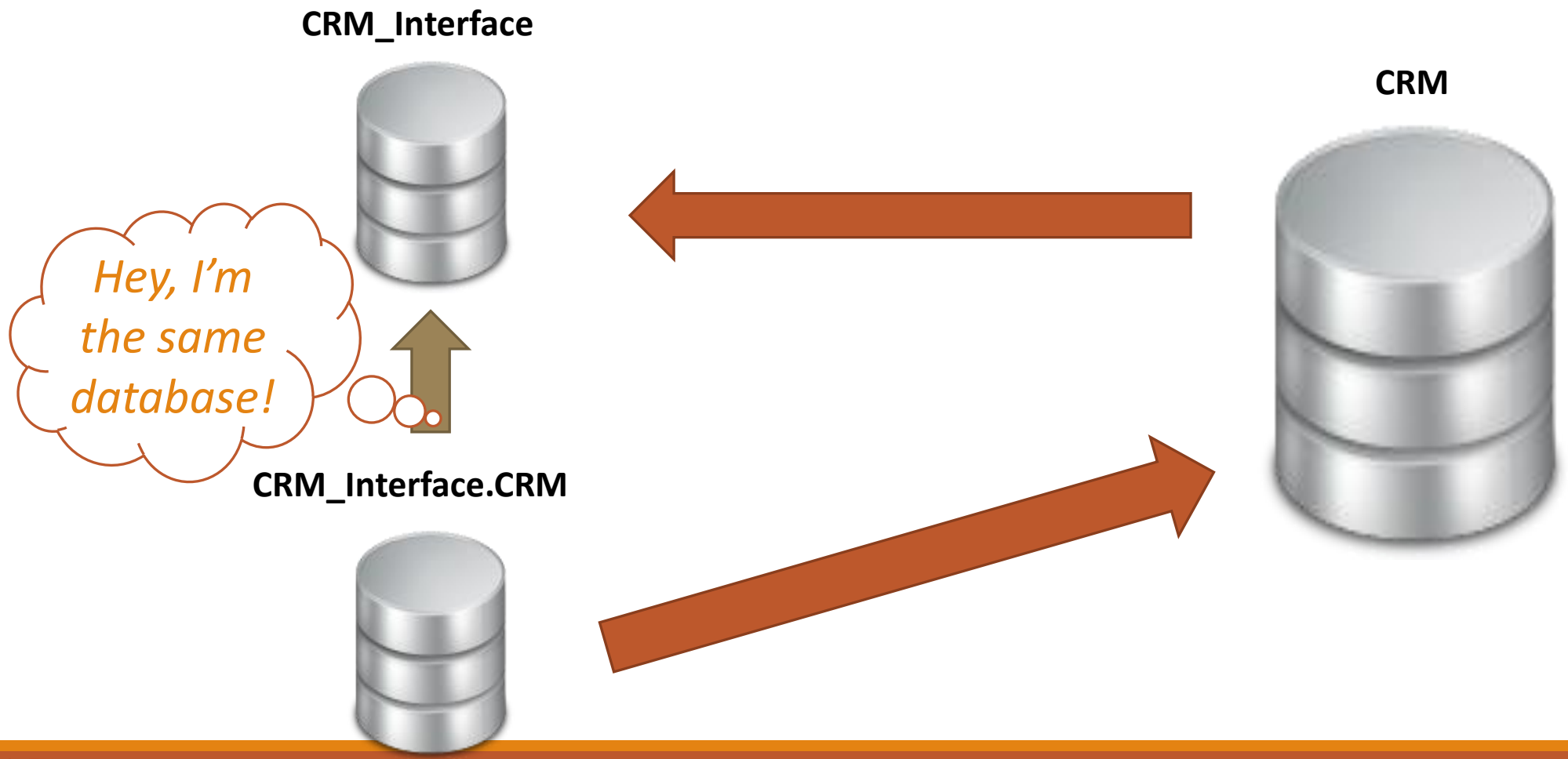
Stored Procedures:
- AddLogItem

CRM

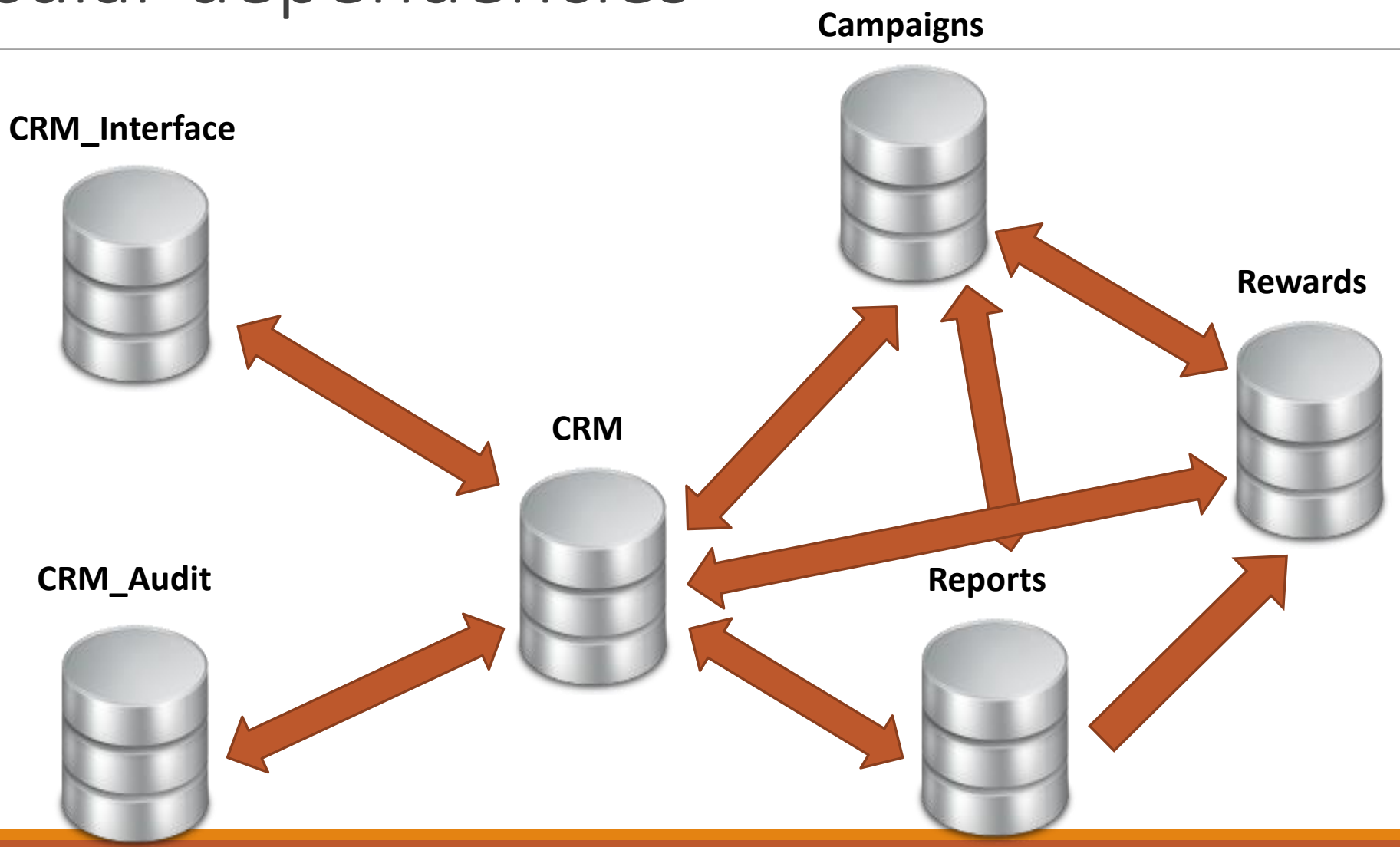
Circular dependencies



Circular dependencies



Circular dependencies



Circular dependencies

Know issue: temp table

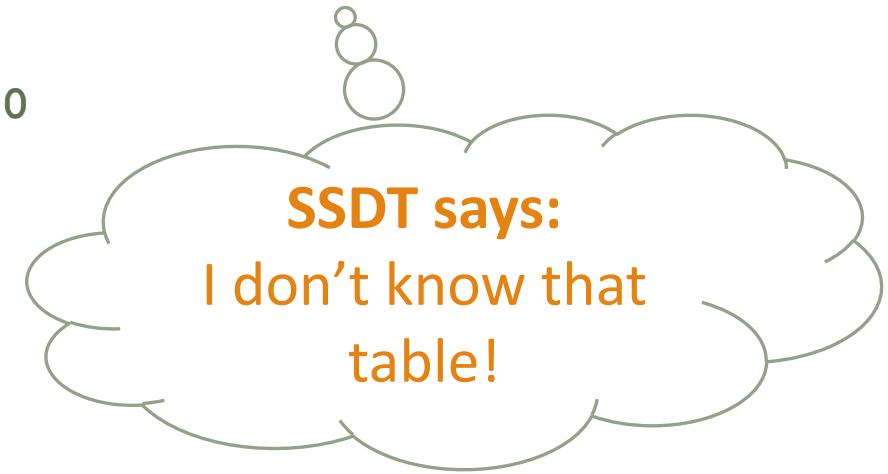
```
CREATE PROCEDURE [dbo].[TempDemo1]
AS


CREATE TABLE #TempTable (Id INT);
EXEC TempDemo2;
```

```
CREATE PROCEDURE [dbo].[TempDemo2]
AS

SELECT * FROM #TempTable;

RETURN 0
```



Know issue: temp table

```
CREATE PROCEDURE [dbo].[TempDemo1]
AS



CREATE TABLE #TempTable (Id INT);
EXEC TempDemo2;
```

```
CREATE PROCEDURE [dbo].[TempDemo2]
AS

IF 0=1
  CREATE TABLE #TempTable (ID int);

SELECT * FROM #TempTable;

RETURN 0
```

Know issue: Disable/Enable trigger #1

- Disable/Enable trigger from other database

```
ALTER TABLE [$(ContosoRetailDW)].dbo.DimChannel
    DISABLE TRIGGER [Trigger_DimChannel];
```

	Code	Description
⊗		SQL71502: Procedure: [dbo].[Populate_DimChannel_from_CRM] has an unresolved reference to object [dbo].[Trigger_DimChannel].

Know issue: Disable/Enable trigger #2

- Alternative #1:
  - Use dynamic SQL (not recommended)

```
--Alternative solution in SSDT:
EXEC sp_executeSQL N'ALTER TABLE [$(ContosoRetailDW)].dbo.DimChannel
        DISABLE TRIGGER [Trigger_DimChannel];
        '
```

Know issue: Disable/Enable trigger #3

- Alternative #2:
  - Create SP in the second database (locally for the trigger)
  - Call that SP from ’Remote’ database

```
CREATE PROCEDURE [dbo].[Toggle_Trigger_DimChannel]
    @enable BIT
AS
    IF @enable = 1
        ENABLE TRIGGER dbo.[Trigger_DimChannel] ON dbo.DimChannel;
    ELSE
        DISABLE TRIGGER dbo.[Trigger_DimChannel] ON dbo.DimChannel;

RETURN 0
```

How to include **data** in a database project?

- SSDT has no built-in solution for including data
- Use Post-Deployment script to populate table
- Wrap the scripts into stored procedures
- Make sure the order of referenced tables

How to include **data** in a database project?

- Scenario #1: initial values only
  - For the very first time (run) only
  - Target table is empty
  - INSERT
  - Example script

How to include **data** in a database project?

- Scenario #2: User has NO access to data
  - Full MERGE statement
  - Include DELETE clause
  - Example script

How to include **data** in a database project?

- Scenario #3: users CAN add values from app
  - MERGE statement
  - Exclude DELETE clause
  - Example script

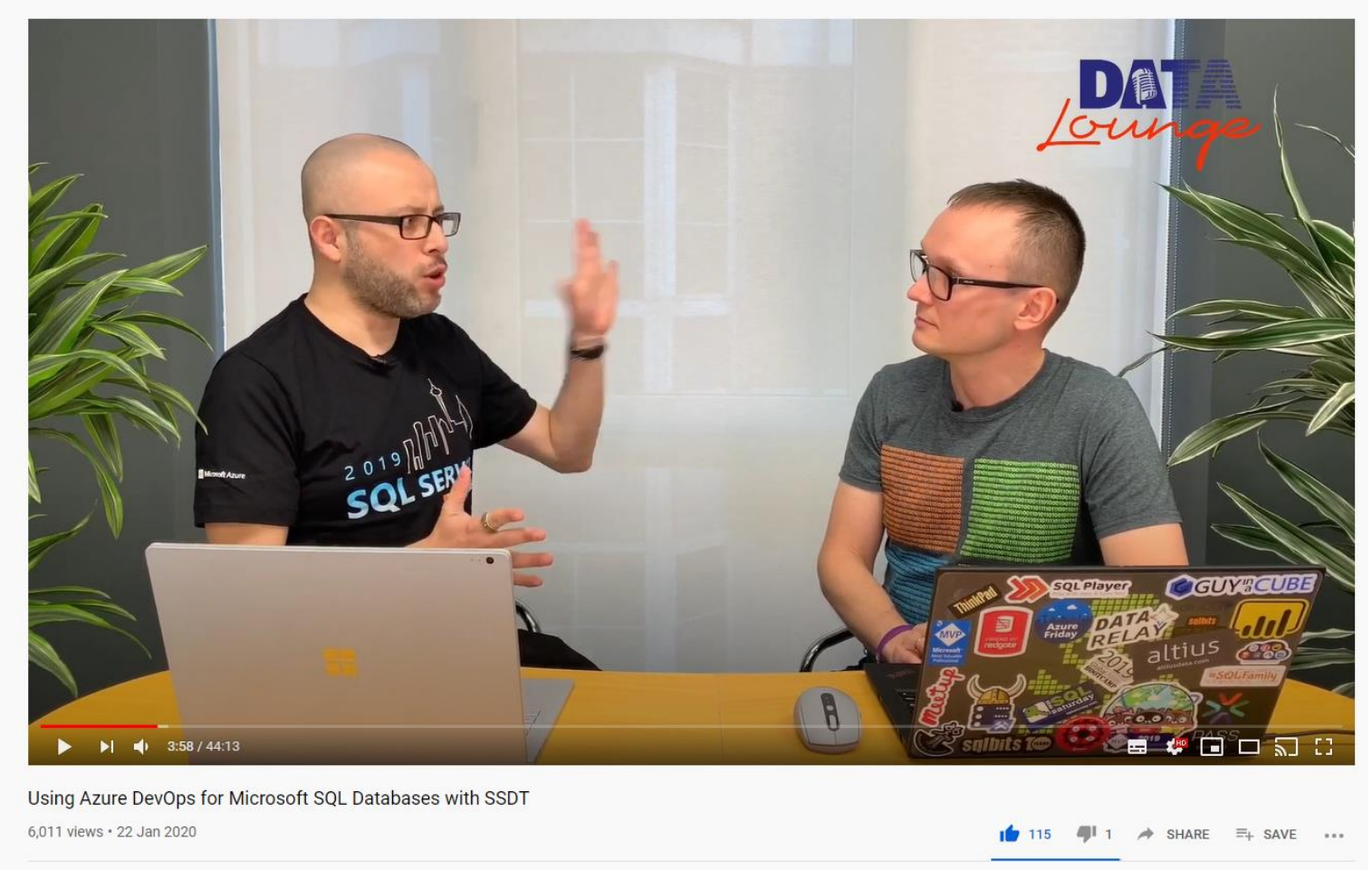
DEMO

- Create publish profile
- Deploy manually
- Data script in Stored Procedure
- Post-Deployment script
- How to change data & deploy it

SQL Jobs in database project

- Facts:
  - DACPAC is a database level project
  - SQLJobs are on server-level
- How to cope with:
  - Add separate database project
  - Only T-SQL scripts
  - Pre/Post deployment script to include above
  - Use PowerShell and SALT module from Sabin.io

CI/CD with Azure DevOps & SSDT



Resources: https://sqlplayer.net/ssdt

SQL Server Data Tools (SSDT)

All the following posts are in SSDT series on the blog:

- New DB project with SSDT and import existing database
- How to create DACPAC file?
- #TSQL2sDay – How SQLPackage.exe can spoil your deployment
- Tokenization of database project in SSDT
- SSDT (SqlPackage) thread crashes with error when publishing
- Script and deploy the data for database from SSDT project
- 🎬 Deployment of Microsoft SQL database with Azure DevOps (video)

QR Code – link to this page

Microsoft resources:

- 🛠 SQL Server Data Tools (MSDN)
- 🛠 SQLPackage.exe (MSDN)
- 🛠 Data-tier Application Framework (DACFx)
- Microsoft SQL Server Data Tools Team Blog
- GIT – branch organization

Other resources:

Questions?

Thank you!

kamil@nowinski.net

@NowinskiK @SQLPlayer

SQLPlayer.net

https://github.com/NowinskiK/CommunityEvents

Kamil Nowinski
Microsoft Data Platform MVP
MCSE Data Platform & MCSE Data Management and Analytics