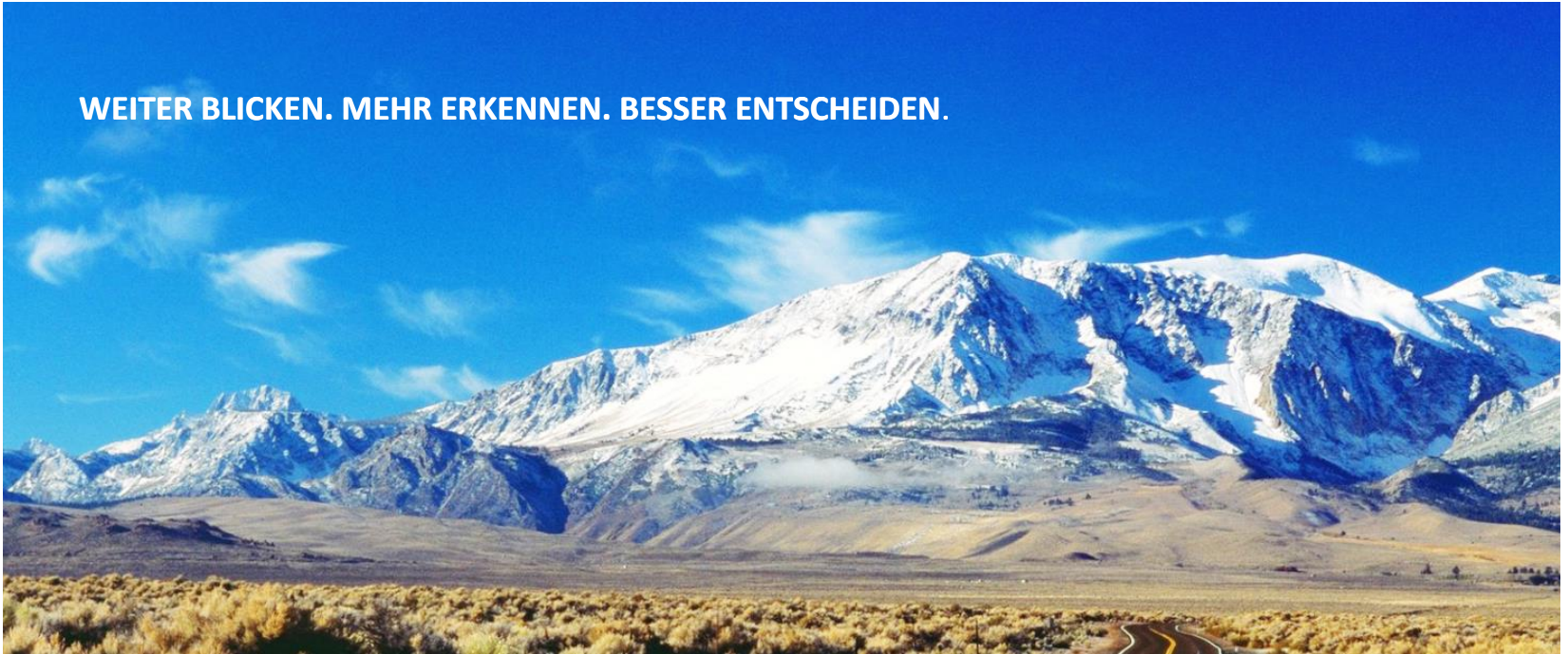



## Das SSIS Paket das sich selbst optimiert

**WEITER BLICKEN. MEHR ERKENNEN. BESSER ENTSCHIEDEN.**



- › SSIS User?
- › Schon mal SSIS Performance Probleme gehabt?
- › Was habt Ihr bisher mit Biml gemacht?

- › Ben Weissman, Solisyon, Nürnberg
- ›  @bweissman
- › [b.weissman@solisyon.de](mailto:b.weissman@solisyon.de)
- › SQL Server seit Version 6.5
- › Zu blöd für C#



Certified Data Vault Modeler

Data Science



Big Data



**Microsoft**  
**CERTIFIED**

Solutions Expert

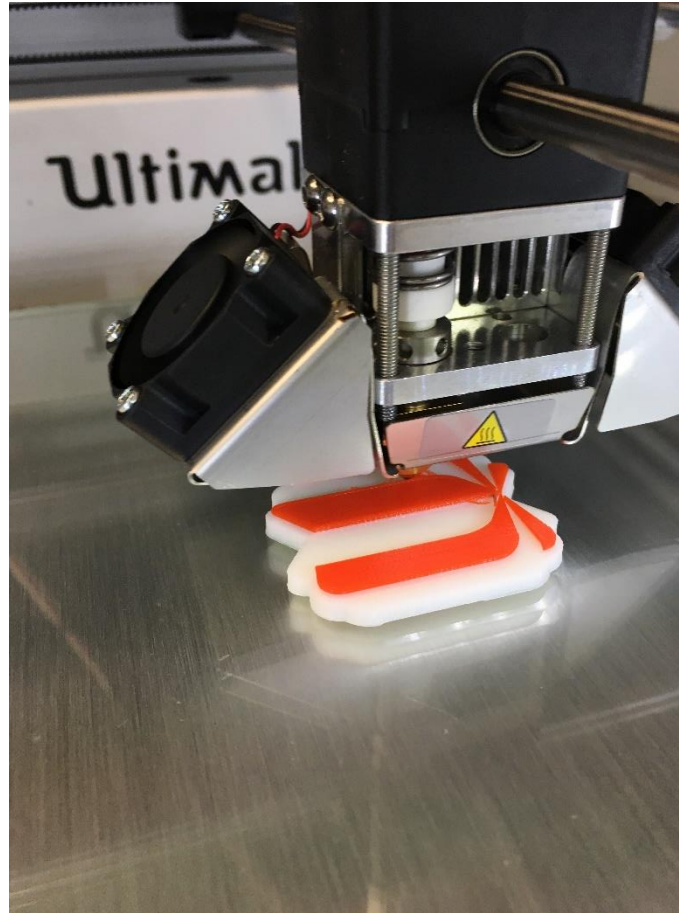
Business Intelligence

**Microsoft**  
**CERTIFIED**

Solutions Expert

Data Platform

# Kleines Intro in Biml



- › Biml ist eine Markup Sprache – also: XML
- › Erfunden/Entwickelt von Varigence
- › So sieht Biml aus:

```
<Biml xmlns="http://schemas.varigence.com/biml.xsd">  
  <Packages>  
    <Package Name="HelloBiml"/>  
  </Packages>  
</Biml>
```
- › Wird erst in Verbindung mit BimlScript/APIs mächtig
- › Kann neben SSIS auch für T-SQL und SSAS genutzt werden
- › Verschiedenste Frontends

 **BimlOnline**

 **BimlExpress**

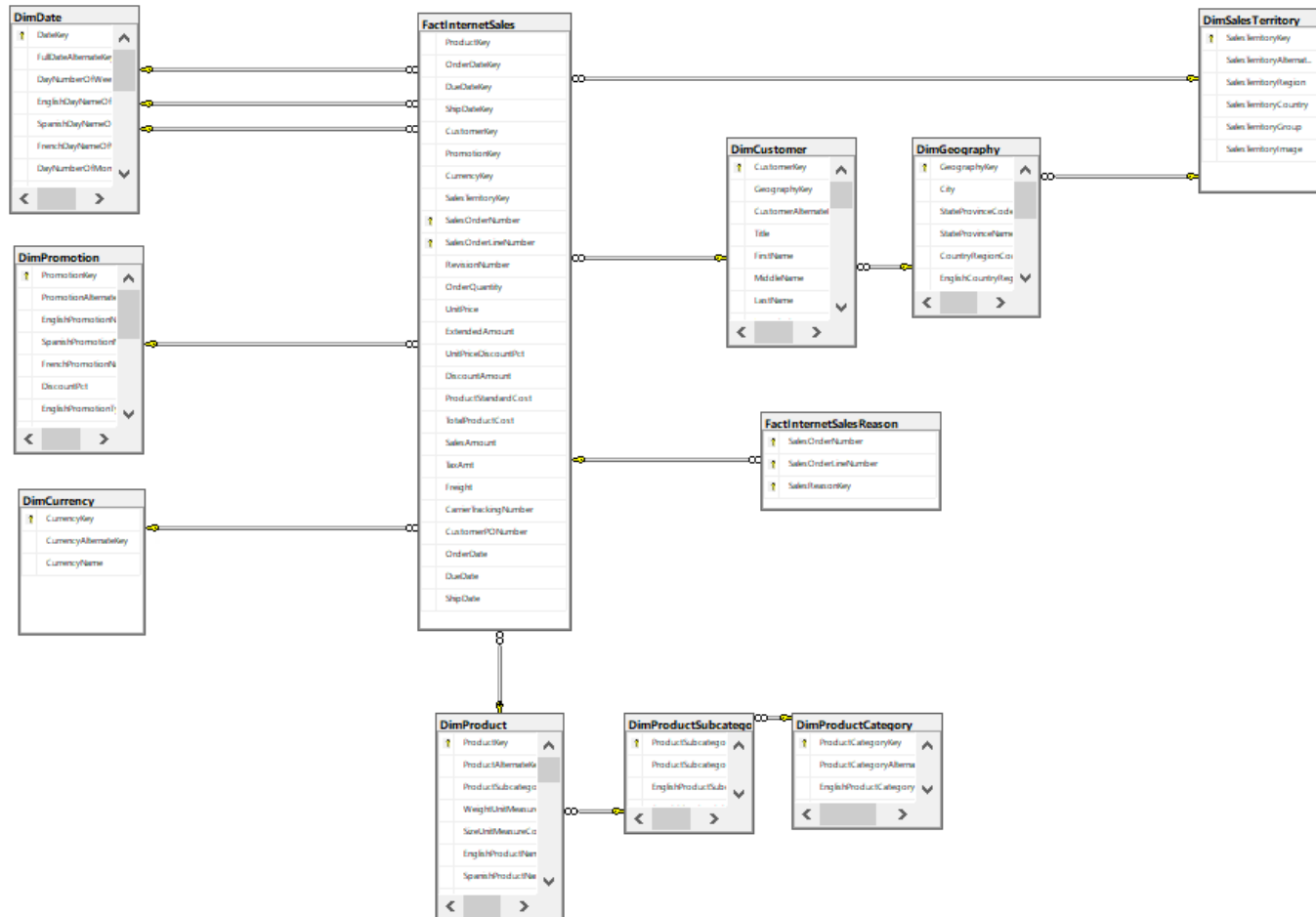
 **BimlStudio**

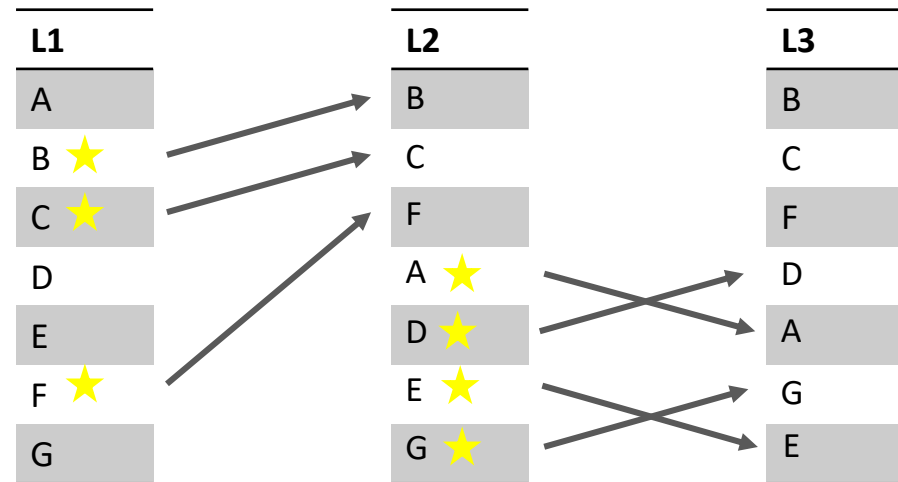
- › Kompatibilität zwischen SQL Versionen wird belanglos
- › Gesteigerte Produktivität durch Zeitersparnis bei “dummen” Datenbeladungen – Suche nach Mustern!
- › Je besser ein Task standardisierbar ist (z.B. Beladung einer Staging Umgebung), desto höher der Zeitgewinn
- › Biml ist der “3D Drucker für SSIS”



- › Topology Sort
- › Pattern evaluation
- › Perfect concurrency

- › Meine Tabellen sind durch Fremdschlüssen voneinander abhängig
- › Es können regelmässig Tabellen hinzukommen oder entfallen
- › Daher muss regelmässig die Reihenfolge der Dataflows in SSIS geprüft werden
- › Schritt 1: Alle Tabellen ohne Fremdschlüssel
- › Schritt 2: nach und nach die weiteren





- › Durchlaufen aller Tabellen
- › Markieren als „besucht“
- › Durchlaufen aller Tabellen die von der aktuellen Tabelle abhängen die noch nicht besucht wurden, diese hinten anhängen

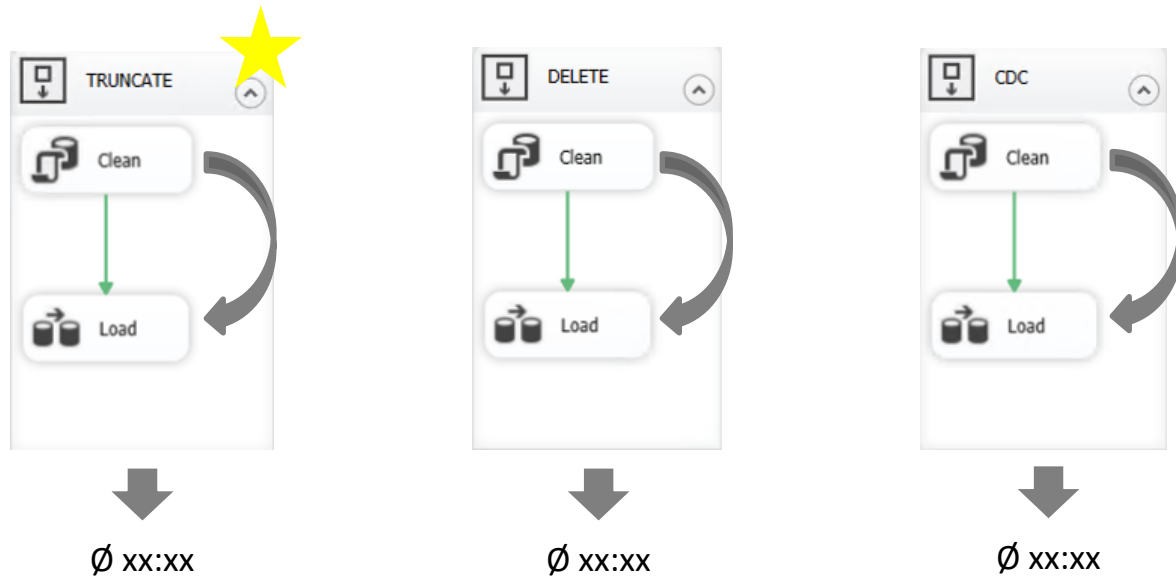
# Demo 1: Linear

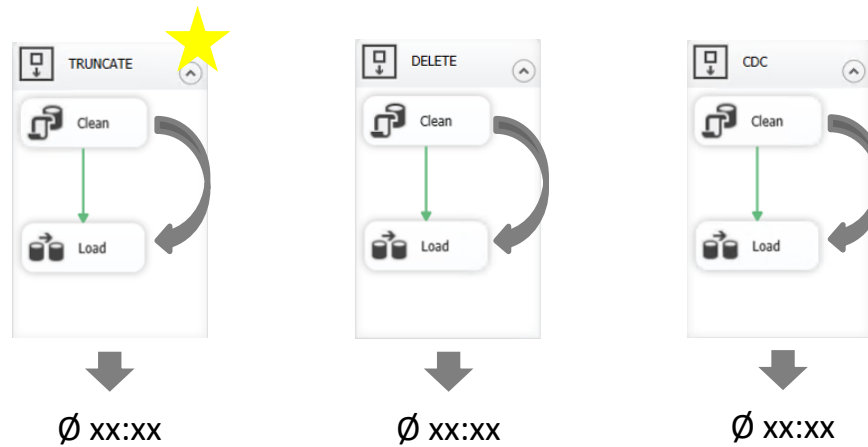
- › Paralleles laden/Durchlaufen aller Tabellen ohne Abhängigkeiten, markieren als geladen
- › Durchlaufen aller Tabellen die von der aktuellen Tabelle direkt abhängen die noch nicht besucht wurden, diese hinten anhängen, markieren als geladen
- › Schleife die alle verbleibenden durchläuft

# Demo 2: Parallel

- › Ich kann Tabellen mit verschiedenen Patterns laden (DELETE, TRUNCATE, CDC...)
- › Jedes dieser Pattern hat eine unterschiedliche Performance/Laufzeit
- › Je nachdem wie sich meine Daten verändern ändert sich diese Performance ggf.
- › Das “richtige” Pattern kann somit im Laufe der Zeit wechseln
- › Ich möchte somit regelmässig das schnellste Pattern identifizieren und dann ab diesem Zeitpunkt nutzen





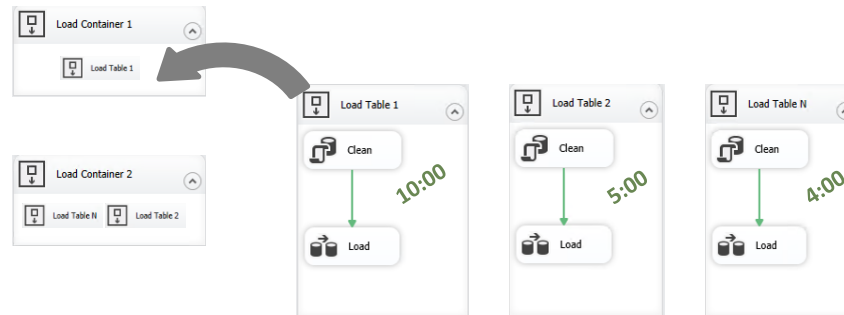


- › Jedes Pattern n mal durchlaufen
- › Durchschnittliche Zeit nehmen
- › „Bestes“ Pattern markieren

# Demo!

- › Mein Quell- oder Zielsystem lässt maximal  $n$  Verbindungen gleichzeitig (sinnvoll) zu
- › Ich möchte meine Dataflows so schnell wie möglich ALLE abgeschlossen haben
- › Somit will ich “kleine” Dataflows zusammenfassen und große unabhängig davon parallel laufen lassen





- › Festlegen Anzahl Container / „Anlegen“ Container
- › Durchlaufen aller Tabellen nach Laufzeit absteigend
- › „Ablegen“ in Container mit kleinster Größe
- › => Alle Container sind relativ gleich groß

# Demo!

 @bweissman

[b.weissman@solisyon.de](mailto:b.weissman@solisyon.de)



# Fragen?



Vielen Dank!

[www.solisyon.de](http://www.solisyon.de)

