SCALING FEATURE GENERATION FROM PROTOTYPING TO PRODUCTION AT REWE

REWE Systems GmbH | March 2019 | Benjamin Greve

AGENDA

1 / Introduction

2 / Example Project: Predicting Brand Market Fit

3 / Feature Generation in Prototyping – The Problems

4 / Moving From Prototyping to Production – Lessons Learned

INTRODUCTION / ME

Benjamin Greve

Data Scientist at REWE Systems

- Mathematician
- Working in data science projects for 5+ years (2 years at REWE)
- Likes hiking, climbing, dancing, cooking (especially the eating part)

INTRODUCTION / REWE GROUP

HANDEL DEUTSCHLAND

REWE
REWE.de
REWE TO GO
nahkauf
PENNY

HANDEL INTERNATIONAL

BILLA
MERKUR
ADEG
BIPA
PENNY MARKET

FACHMARKT NATIONAL

toom
Respekt, wer's selber macht.

TOURISTIK

DER Touristik
DER Business Travel
DERPART
DERTOUR
JAHN REISEN
ITS
MEIER'S WELTREISEN
ADAC Reisen
lti hotels
Club Calimera
KUONI
helvetic tours
ITS coop TRAVEL
ITS BILLA REISEN
apollo
EXIM holding

SONSTIGE

REWE digital
ZooRoyal
FÜRSIE wir handeln.
REWE SYSTEMS
EHA Energie geht neue Wege
commercetools
KÖLNER WEINKELLER
WEINFREUNDE REWE

Source: https://www.rewe-group-geschaeftsbericht.de, 14.03.2019

INTRODUCTION / REWE GROUP

REWE Group facts (2017)

57.8
Billion Euro

TOTAL REVENUE 2017

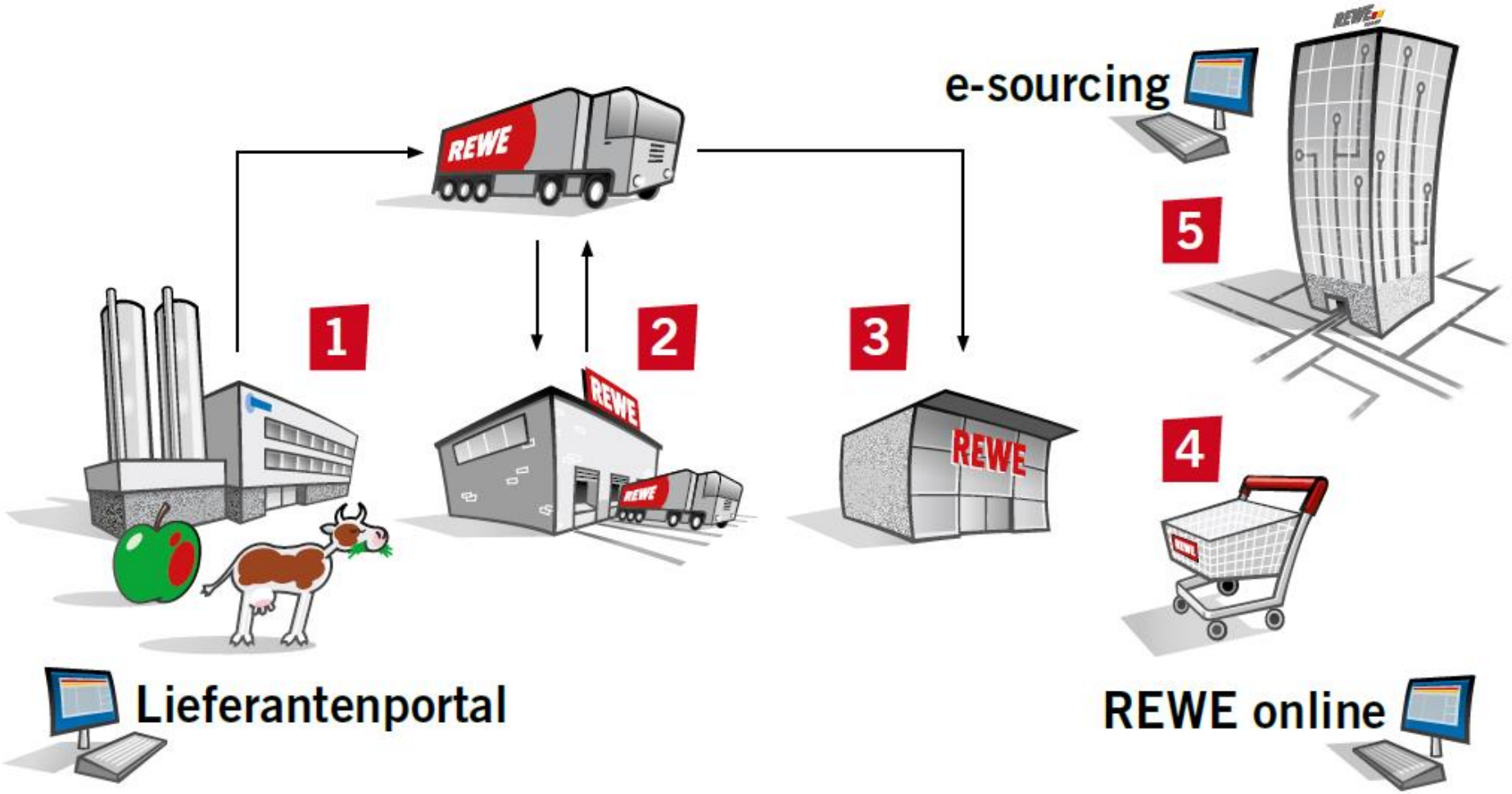
345,000

EMPLOYEES 2017

15,300

MARKETS 2017

INTRODUCTION / REWE SYSTEMS



INTRODUCTION / REWE SYSTEMS

1,200
IT specialists

30,000
cash registers

>1 Bil.
data sets
every day

REWE SYSTEMS

30
locations

7,500
markets

200,000
users

EXAMPLE PROJECT: PREDICTING BRAND MARKET FIT

Not an actual project – just for illustration purposes

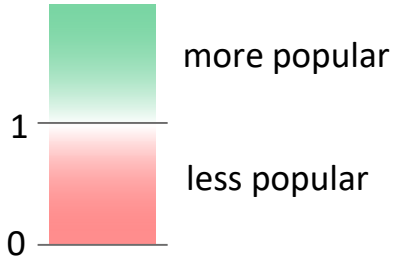
We want to predict how well a brand will be received in a particular market to assist category managers in selecting the right brands for each market.

BRAND / CATEGORY:

A set of products that form a logical group. Can contain between one and a few thousand products.

Technical definition

TARGET VARIABLE y:
Brand popularity in current market compared to average across all REWE markets

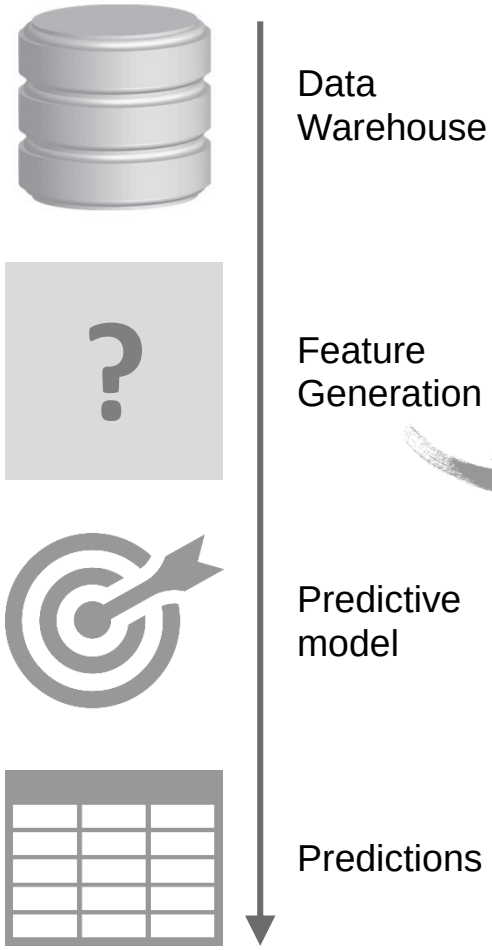


FEATURES x_1, x_2, …

- Popularity of wider category in market
- Number of competing articles
- Location of market, incl. demographical information
- …

FEATURE GENERATION IS CENTRAL PART IN THE DATA SCIENCE WORKFLOW

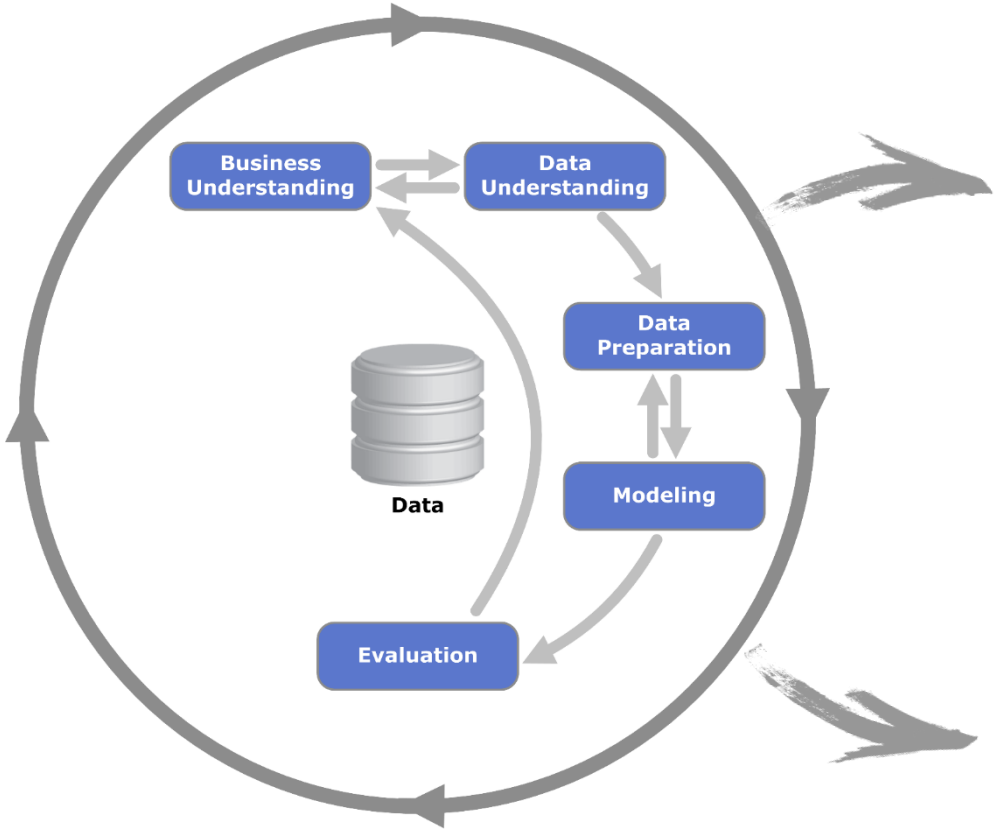
CRISP-DM: Cross-industry standard process for data mining

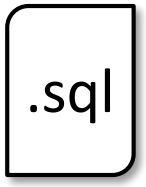


„Everything you do with the data before you apply a predictive model“

https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining#/media/File:CRISP-DM_Process_Diagram.png

A TYPICAL RESULT OF PROTOTYPING IS A VERY LONG SQL SCRIPT





OUTPUT #1: Very long SQL script

- 2000+ lines of SQL
- 40+ intermediate tables
- interdependencies
- inconsistent naming, chaotic code style
- not optimized for performance
- not robust

→ not production-ready or scalable

OUTPUT#2: Very long R or Python script

- Different topic for another talk

Prototyping performed by Data Scientists

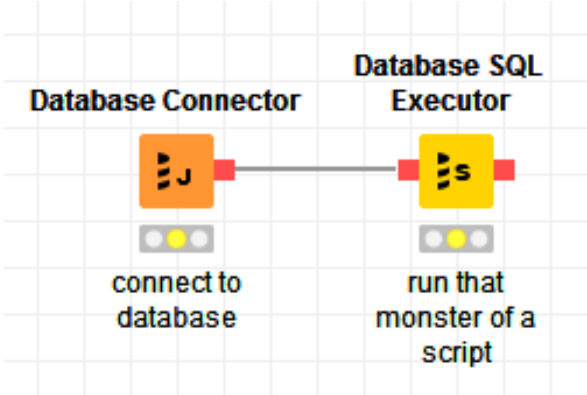
A TYPICAL RESULT OF PROTOTYPING IS A VERY LONG SQL SCRIPT

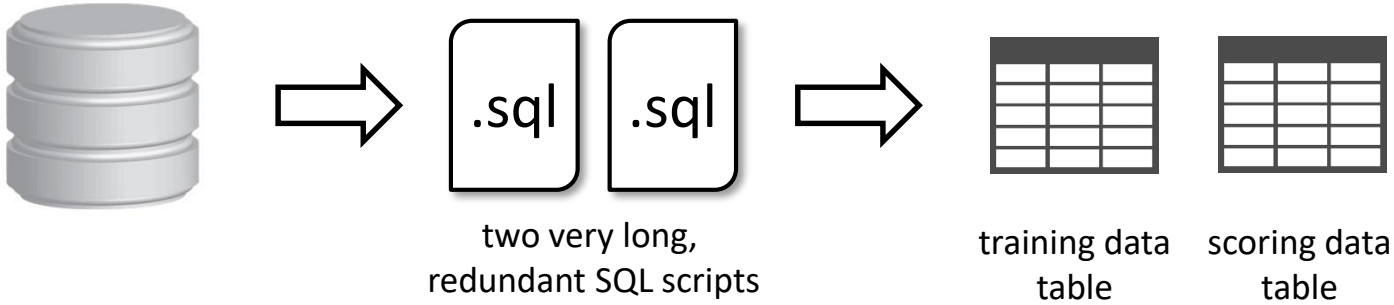
<u>OUTPUT #1: Very long SQL script</u>

- 2000+ lines of SQL
- 40+ intermediate tables
- interdependencies
- inconsistent naming, chaotic code style
- not optimized for performance
- not robust

→ not production-ready or scalable

AVOID LONG, MONOLITHIC SQL SCRIPTS

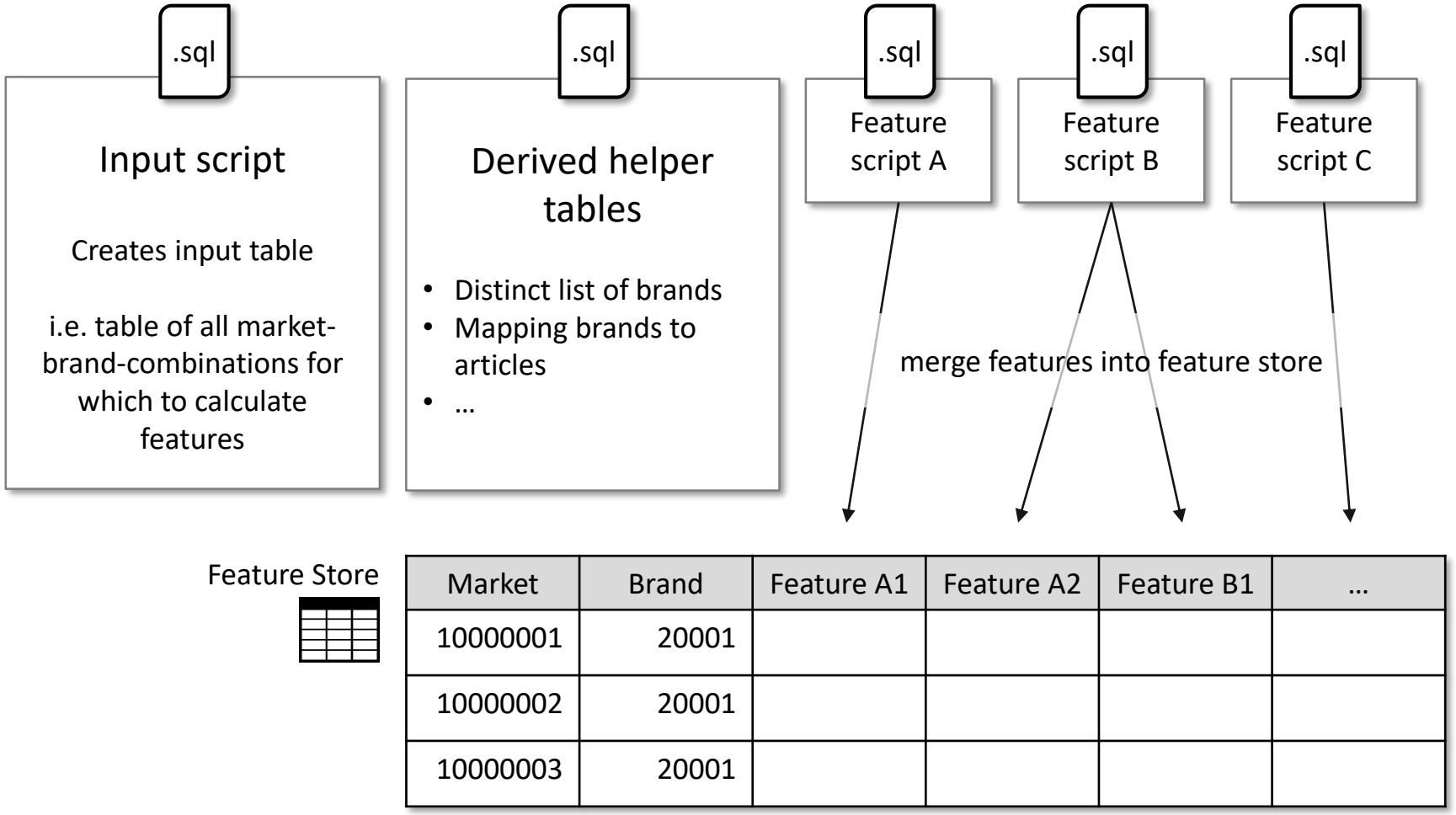




Weaknesses:

- Duplicate/redundant SQL for every feature
- Parameters hidden within the scripts
- Adding new features is hard due to interdependencies

USE A MODULAR FEATURE GENERATION INSTEAD



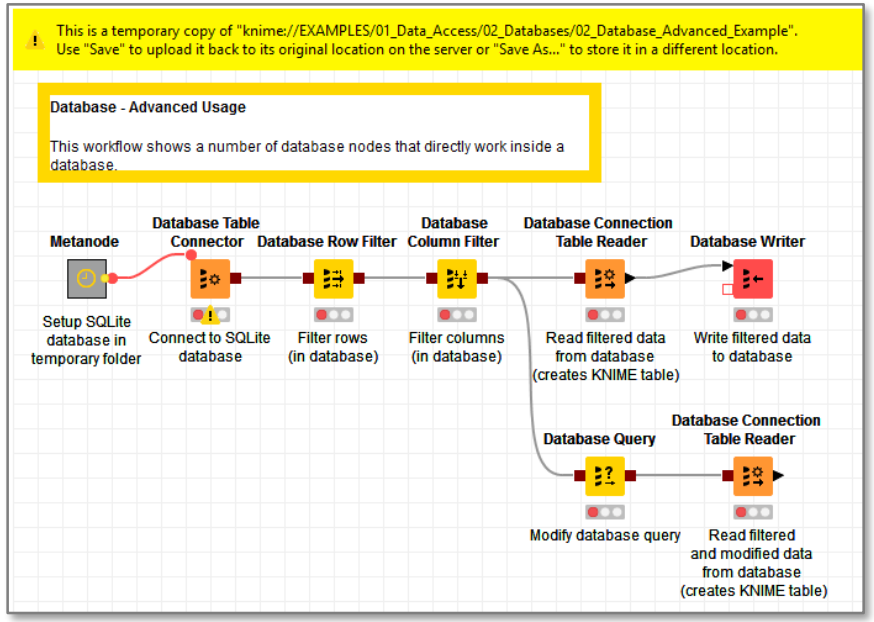
Strengths:

- ✓ Highly modular and scalable
- ✓ Parameters are centrally defined
- ✓ Feature scripts are mostly independent, making it easy to add new features

DECOMPOSING COMPLEX SQL WITHIN KNIME LEADS TO COMPLEX WORKFLOWS

- ✓ Code
- ✓ Powerful version control (releases)
- ✓ Analysts fluent in SQL

- Harder to explain/show

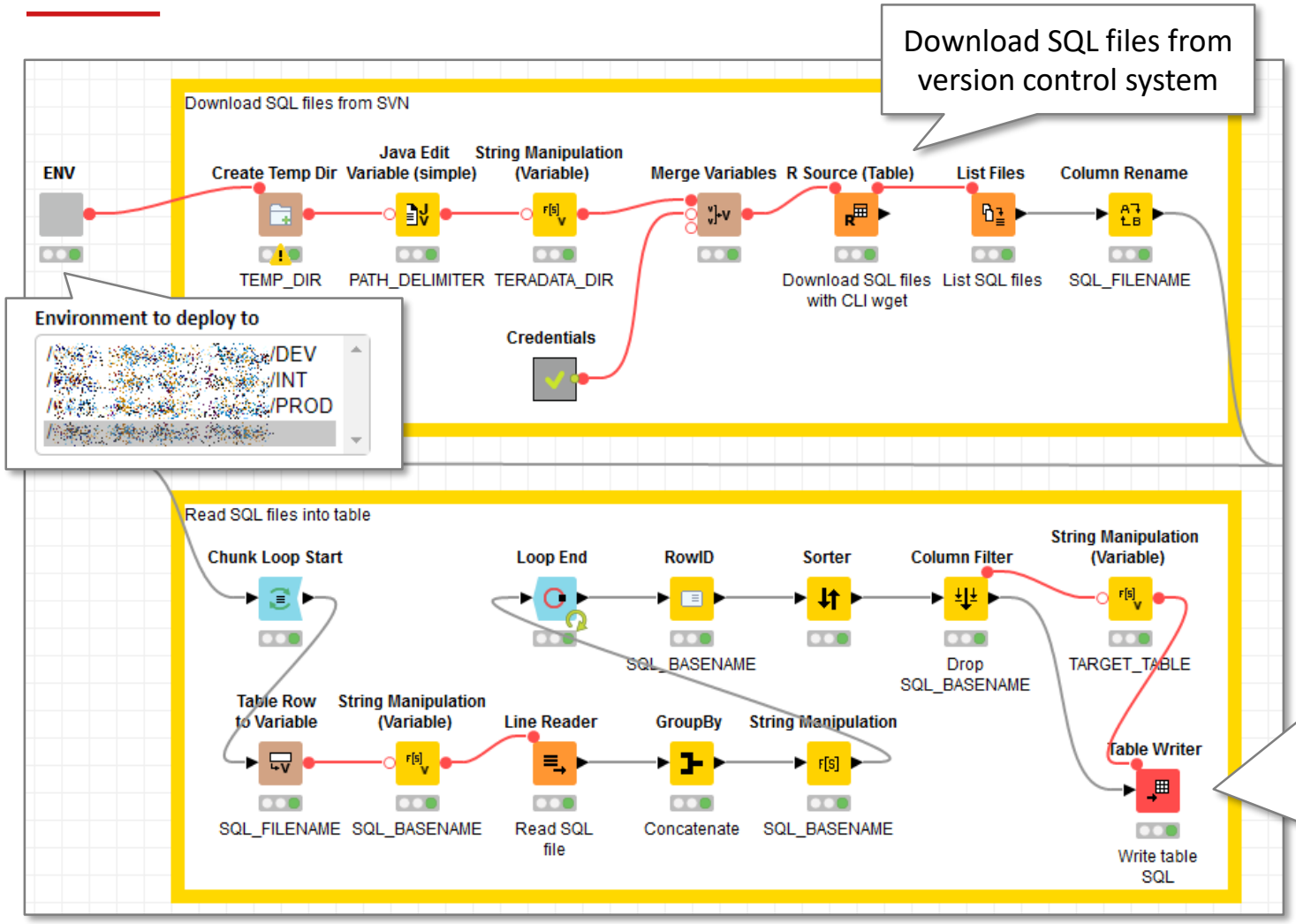


Example workflow from KNIME Examples server

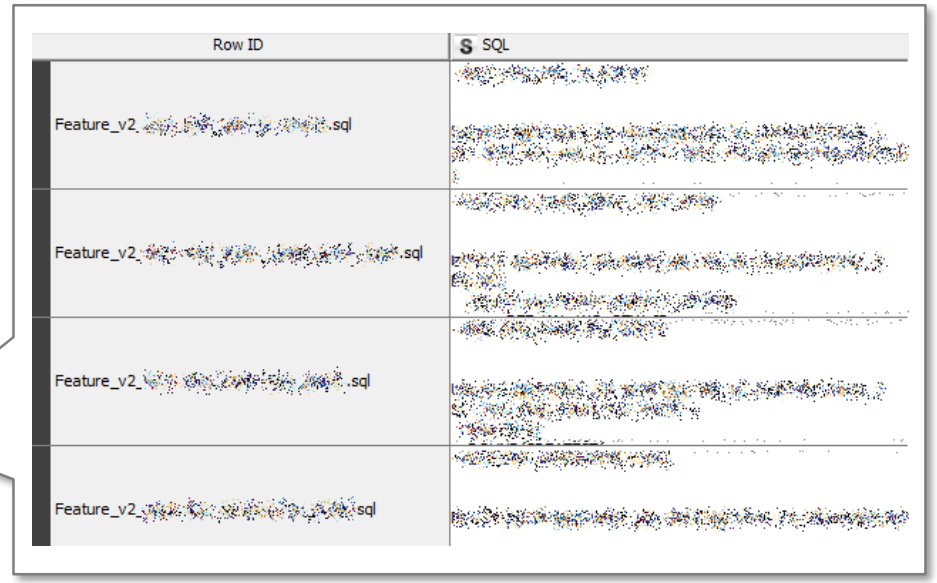
- ✓ Visual, KNIME native

- Gets too complex as SQL logic grows
- Hard to translate into standard data warehouse ETLs

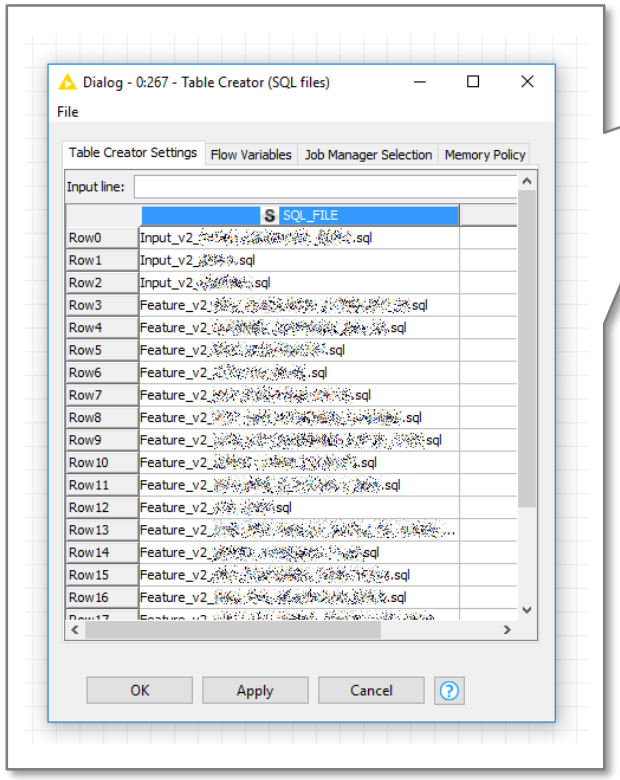
A DEPLOYMENT WORKFLOW PULLS SQL SCRIPTS FROM VERSION CONTROL TO KNIME SERVER

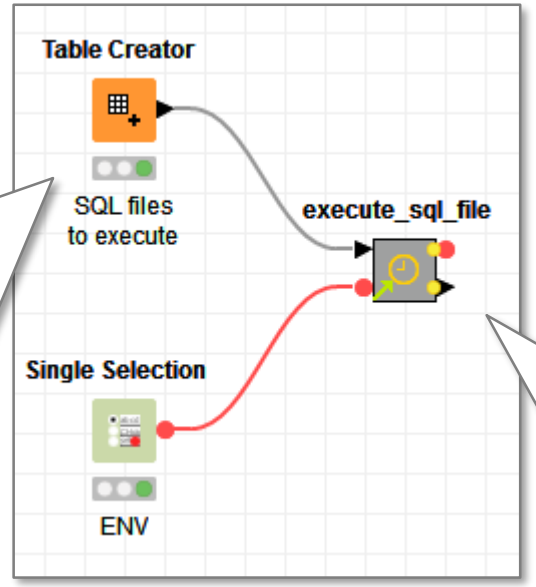


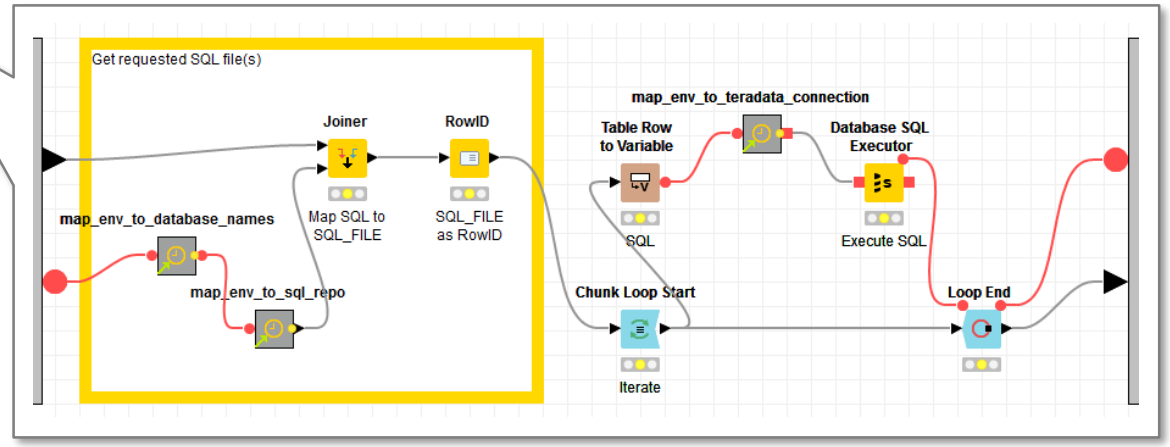
A fully automated deployment workflow copies snapshot or release versions from the version control system to the KNIME Server



IT'S VERY SIMPLE TO EXECUTE THE DEPLOYED SQL FILES

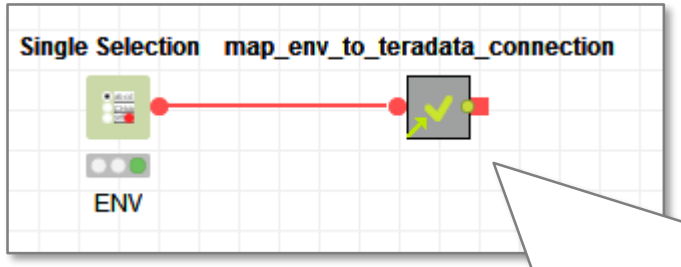






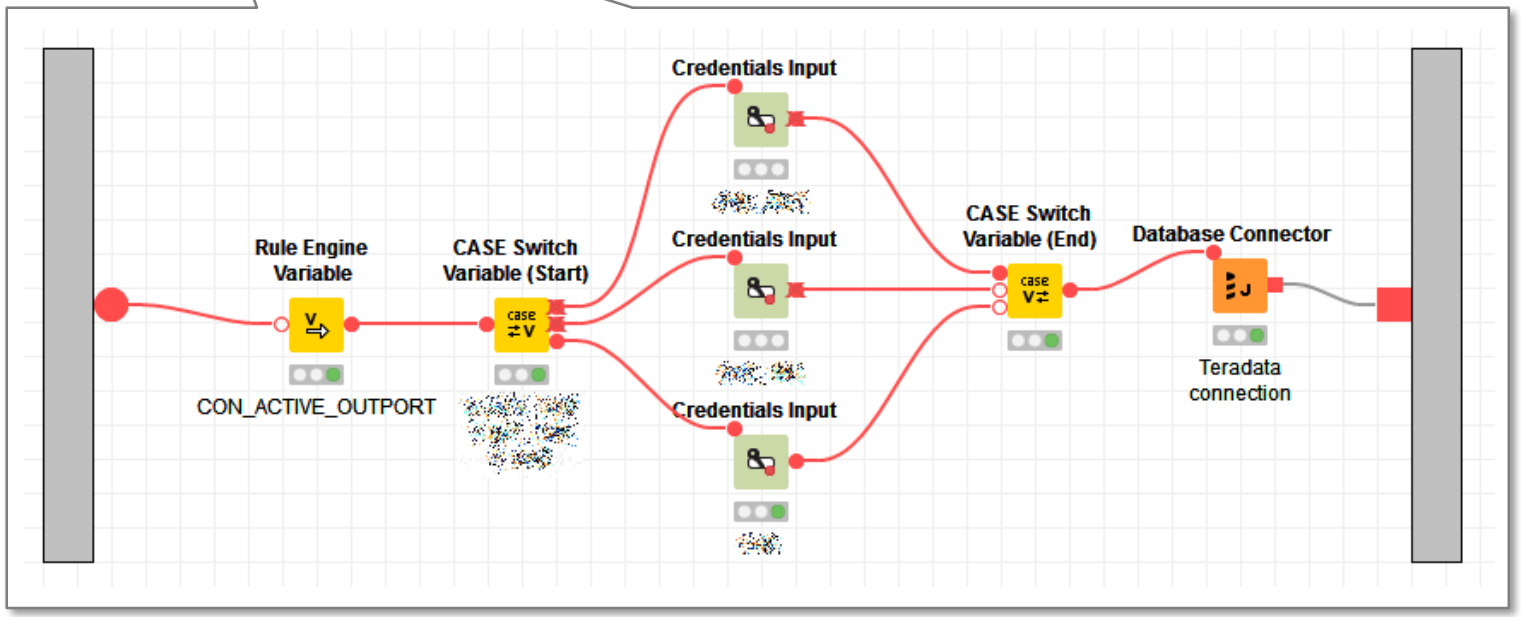
- To perform Feature Generation, execute a set of SQL scripts in a given order
- Just write down the name of the files to be executed in the correct order and run the workflow

HELPFUL METANODES EXAMPLE: GET DATABASE CONNECTION



Useful metanodes help to implement a staging concept, e.g.:

Select staging environment → obtain connection to the corresponding database



NEXT STEPS

Currently:

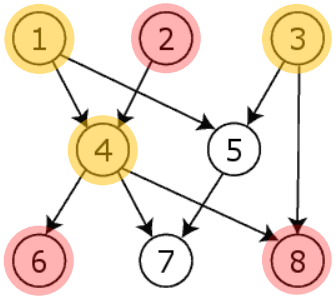
	SQL_FILE
Row0	Input_v2_[illegible].sql
Row1	Input_v2_[illegible].sql
Row2	Input_v2_[illegible].sql
Row3	Feature_v2_[illegible].sql
Row4	Feature_v2_[illegible].sql
Row5	Feature_v2_[illegible].sql
Row6	Feature_v2_[illegible].sql

- specify scripts manually
- order manually to take dependencies into account

Next level:

	SQL_FILE
Row0	Feature_13
Row1	Feature_5
Row2	Feature_7
Row3	Feature_36
Row4	Feature_2
Row5	Feature_9
Row6	Feature_10

- specify features in arbitrary order



Feature dependency graph

- Automatic dependency resolution
- Already built into KNIME!

THANKS

Thanks for your attention