

# Vollständigkeit von Migrationsprogrammen messen und visualisieren



Philipp Salvisberg  
Senior Principal Consultant



BASEL ■ BERN ■ BRUGG ■ DÜSSELDORF ■ FRANKFURT A.M. ■ FREIBURG I.BR. ■ GENÈVE  
HAMBURG ■ KOPENHAGEN ■ LAUSANNE ■ MÜNCHEN ■ STUTTGART ■ WIEN ■ ZÜRICH

**trivadis**  
makes IT easier. ■ ■ ■

# ■ Philipp Salvisberg

- Trivadianer seit April 2000
  - Senior Principal Consultant, Partner
  - Mitglied des Verwaltungsrats
  - [philipp.salvisberg@trivadis.com](mailto:philipp.salvisberg@trivadis.com)
  - [www.salvis.com/blog](http://www.salvis.com/blog)
  - [@phsalvisberg](https://twitter.com/phsalvisberg)
- Datenbanknahe Softwareentwicklung mit Oracle Database Server
- Über 20 Jahre Erfahrung mit Oracle Produkten
- Autor von SQL Developer Extensions PL/SQL Cop und PL/SQL Unwrapper



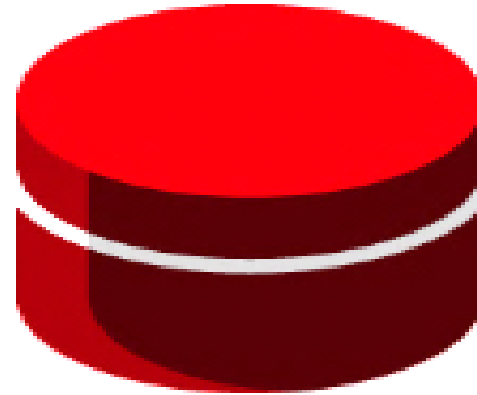
# ■ Agenda

1. Projekt
2. Motivation
3. Messen
4. Visualisieren
5. Kernaussagen

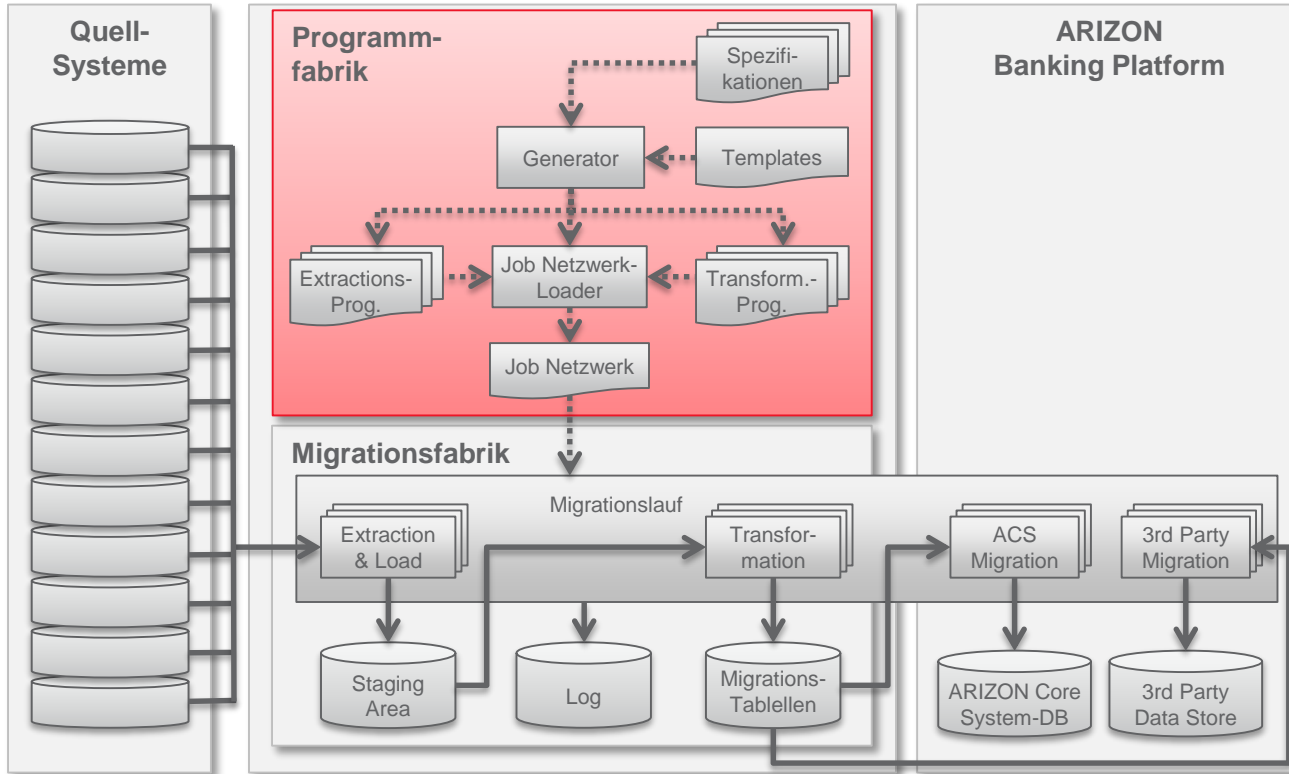
# Projekt

# ■ Migration auf Avaloq Banking System

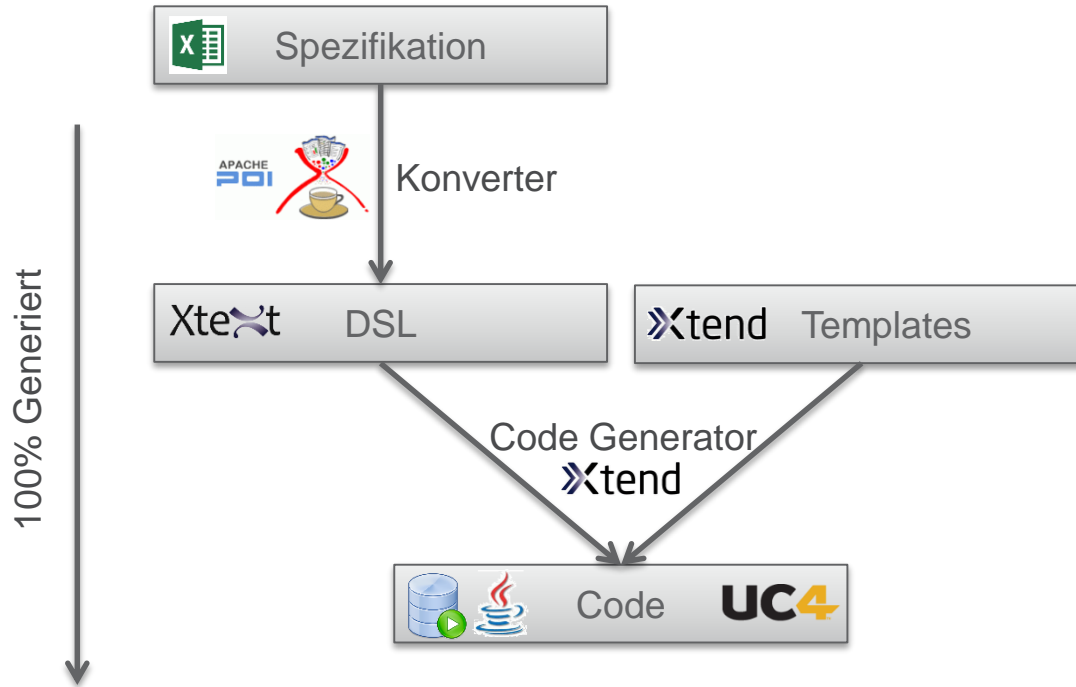
- Über 300 dezentrale “DIALBA 2000” Installationen
- Individuelle Konfigurationen
- Zentrales Avaloq Banking System
- Bank = Business Unit
- Einheitliche Konfiguration
- Einführung ab 2017



# Kontext



# ■ Programmfabrik



Abstraktions-Layer 1

Abstraktions-Layer 2

Code-Layer

# Beispiel

DATEI	START	EINFÜGEN	SEITENLAYOUT	FORMELN	DATEN	ÜBERPRÜFEN	ANS
D92							
1	A	B			Mapping name	natural	
2					Source Alias	natural_person	
3							
4					Transform		
5	pers	O_DOC_ID			1		
6	pers	O_OBJ_ID			2		
7	pers	O_MIGRT_RUN_SEQ_NR			3		
8	pers	I_MIGRT_SOURCE			4	default()	
9	pers	I_MIGRT_KEY			5	"zreg.man_migrt"	
10	pers	I_MIGRT_KEY_SEQ_NR			6	get_seq_nr('p	
11	pers	I_OBJ_ID			7		
12	pers	I_OBJ_KEY			8		
13	pers	I_OBJ_KEY_ID			9		
14	pers	I_EFF_DATE			10	decision_table	
15	pers	I_OPEN_DATE			11	decision_table(open_date, i_datum = zreg.edatum)	decision_table(open_date, i_datum = zreg.edatum)
16	pers	I_CLOSE_DATE			12		
17	pers	I_SORT_ALPHA			13		
18	pers	I_SORT_NR			14		
19	pers	I_BU_ID			15		
20	pers	I_BIRTH_DATE			16	zreg.gdatum	
21	pers	I_CEASE_DATE			17		
22	pers	I_DEATH_DATE			18	zreg.sterbedatum	
23	pers	I_ESTABLISH_DATE			19	zreg.gdatum	
24	pers	I_COUNTRY_NAT_ID			20	get_obj_id(rmsCountry_iso_3, translate(zreg.nation, rch_country))	

```
Java - test/rsc-gen-idi/person.spec : Eclipse Platform
File Edit Navigate Search Project Run Window Help
Quick Access Resource Java

group person depends on addr

--map natural group person
migration call (migrtgen_person, get_seq_nr('person.natural'), "5100", "6100")
data (pers, natural_person.v) {
  i_migrt_source = default()
  i_migrt_key = "zreg.man_mandantnr || '-' || zreg.zrnr"
  i_migrt_key_seq_nr = get_seq_nr('person.natural')
  i_eff_date = decision_table(open_date, i_datum = zreg.edatum)
  i_open_date = decision_table(open_date, i_datum = zreg.edatum)
  i_birth_date = zreg.gdatum
  i_death_date = zreg.sterbedatum
  i_country_nat_id = get_obj_id(rmsCountry_iso_3, translate(zreg.nation, rch_country))
  i_dom_addr_key = "zreg.man_mandantnr || '-' || zreg.zrnr"
  i_dom_addr_key_id = get_obj_key_id(andsaddr_migr)
  i_gender_id = zreg.geschlecht
  i_lang_id = get_code_id(code_wup_lang_intl_id = translate(zreg.sprache, rch_lang))
  i_marital_status_id = get_code_id(code_marital_status_intl_id = translate(zreg.zivilstand, rch_marital_status))
  i_profession = zreg.beruf
  i_nof_emp_id = zreg.anzahlmacd

--map corporate group person
migration call (migrtgen_person, get_seq_nr('person.corporate'), "5100", "6100")
data (pers, corporate_person.v) {
  i_migrt_source = default()
  i_migrt_key = "zreg.man_mandantnr || '-' || zreg.zrnr"
  i_migrt_key_seq_nr = get_seq_nr('person.corporate')
  i_eff_date = decision_table(open_date, i_datum = zreg.edatum)
  i_open_date = decision_table(open_date, i_datum = zreg.edatum)
  i_establish_date = zreg.gdatum
  i_dom_addr_key = "zreg.man_mandantnr || '-' || zreg.zrnr"
  i_dom_addr_key_id = get_obj_key_id(andsaddr_migr)
  i_lang_id = get_code_id(code_wup_lang_intl_id = translate(zreg.sprache, rch_lang))
  i_person_type_id = get_number(def_obj_sub_type_person_legal)
  i_nof_emp_id = zreg.anzahlmacd

--map assoc group person
migration call (migrtgen_person, get_seq_nr('person.assoc'), "5100", "6100")
data (pers, assoc_person.v) {
  i_migrt_source = default()
  i_migrt_key = "zreg.man_mandantnr || '-' || zreg.zrnr"
  i_migrt_key_seq_nr = get_seq_nr('person.assoc')
```

```
Oracle SQL Developer: Package Body KAN_MIGRT_PERSON Body@esvan015 - k
File Edit View Navigate Run Source Team Tools Window Help
Code Profiles | Details | References | Errors | Dependencies | Grants
Find
SELECT 'NATURAL' AS i_migrt_source,
zreg.man_mandantnr || '-' || zreg.zrnr AS i_migrt_key,
an_migrt_key_seq.get_migrt_key_seq_nr(i_group_name => 'person.natural', i_tenant => i_tenant)
CASE
WHEN (zreg.edatum <= an_constant_helper.get_date('session#', 'session#today'))
THEN
zreg.edatum
ELSE
an_constant_helper.get_date(i_package_name => 'session#', i_constant_name => 'session#')
END AS i_eff_date,
CASE
WHEN (zreg.edatum <= an_constant_helper.get_date('session#', 'session#today'))
THEN
zreg.edatum
ELSE
an_constant_helper.get_date(i_package_name => 'session#', i_constant_name => 'session#')
END AS i_open_date,
zreg.gdatum AS i_birth_date,
zreg.sterbedatum AS i_death_date,
an_lookup_helper.get_obj_id(i_intl_id => 'rsaCountry_iso_3', i_key_val => DECODE(zreg.
, 1, 'CHE'
, 16, 'ARF'
, 20, 'ZAP'
, 28, 'NAM'
, 32, 'ALB'
, 36, 'DZA'
, 44, 'DEU'
, 60, 'AND'
, 62, 'AGO'
, 64, 'ATX'
, 65, 'ATA'
, 66, null
, 70, 'ANT'
, 74, 'SAU'
, 82, 'LSP'
```





# Motivation

# ■ Definition of Done – oder was heisst “fertig” genau?



# ■ GTA versus Programmfabrik



# Messen

# ■ Auszug „Migration“

```
INSERT INTO k.migrt_addr addr (... , i_salut_id, ...)
SELECT ...
  migr.rm$lookup_helper#.get_code_id(
    i_tab => 'code_salut',
    i_fld => 'intl_id',
    i_val => DECODE(zreg.anredc
      , 1000, null
      , 1001, 'rm$mr'
      , 1002, 'rm$mrs'
      , 1003, 'rm$mrms'
      , 1004, 'rm$mrsmr'
      , 1005, 'rm$mrmr', ...
      , 1010, 'rm$company'
    ) AS i_salut_id, ...
FROM own_d2kz.d2k_ba_zreg zreg ...
```

2. Target-Column

1. Source-Column

2. Target-Column

3. Target-Domain-Values  
(gültige Werte)

# ■ Ansatz

- Migrationscode wird vollständig anhand der Spezifikationen generiert
- Verwendungen werden auch anhand der Spezifikationen identifiziert und entsprechende Insert-Statements in Staging-Tabellen generiert
  - Source-Columns (mit Hilfe statischer Code-Analyse, Trivadis PL/SQL Parser)
  - Target-Columns
  - Target-Domain-Values
- Laden des Modells anhand Staging-Tabellen, Avaloq Parametrierung und der Oracle Datenstrukturen des Quell- und Ziel-Systems
- Model des Fertigstellungsgrad der Migrations-Programme gehört zum Installationsumfang

# 1. Source Columns (aus der primären Quelle "Dialba")

RM\$SD_SYSTEMS	
P *	SYSTEM_ID INTEGER
U *	SYSTEM_NAME VARCHAR2(30 CHAR)
RM\$SD_SYS_PK (SYSTEM_ID) RM\$SD_SYS_UK (SYSTEM_NAME)	

RM\$SD_TABLES	
P *	TABLE_ID INTEGER
UF *	SYSTEM_ID INTEGER
U *	TABLE_NAME VARCHAR2(30)
RM\$SD_TAB_PK (TABLE_ID) RM\$SD_TAB_UK (SYSTEM_ID, TABLE_NAME) RM\$SD_TAB_SYS_FK (SYSTEM_ID)	

RM\$SD_COLUMNS	
P *	COLUMN_ID INTEGER
UF *	TABLE_ID INTEGER
U *	COLUMN_NAME VARCHAR2(30 CHAR)
RM\$SD_COL_PK (COLUMN_ID) RM\$SD_COL_UK (TABLE_ID, COLUMN_NAME) RM\$SD_COL_TAB_FK (TABLE_ID)	

RM\$SD_SOURCE_COLUMNS	
P *	SOURCE_COLUMN_ID INTEGER
UF *	COLUMN_ID INTEGER
F *	STATE_ID INTEGER
RM\$SD_SOURCE_COLUMNS_PK (SOURCE_COLUMN_ID) RM\$SD_SOURCE_COLUMNS_UK (COLUMN_ID) RM\$SD_SOURCE_COLS_COLS_FK (COLUMN_ID) RM\$SD_SOURCE_COLS_STATES_FK (STATE_ID)	

use static code analysis of transformation queries

state_name	aggr_order_seq	done
missing	0	0
registered	1	0
open – analysis/design	2	0
open – migration analysis	3	0
open – migration factory	4	0
open – DQQM	5	0
open – parametrization	6	0
irrelevant	7	1
completed	8	1

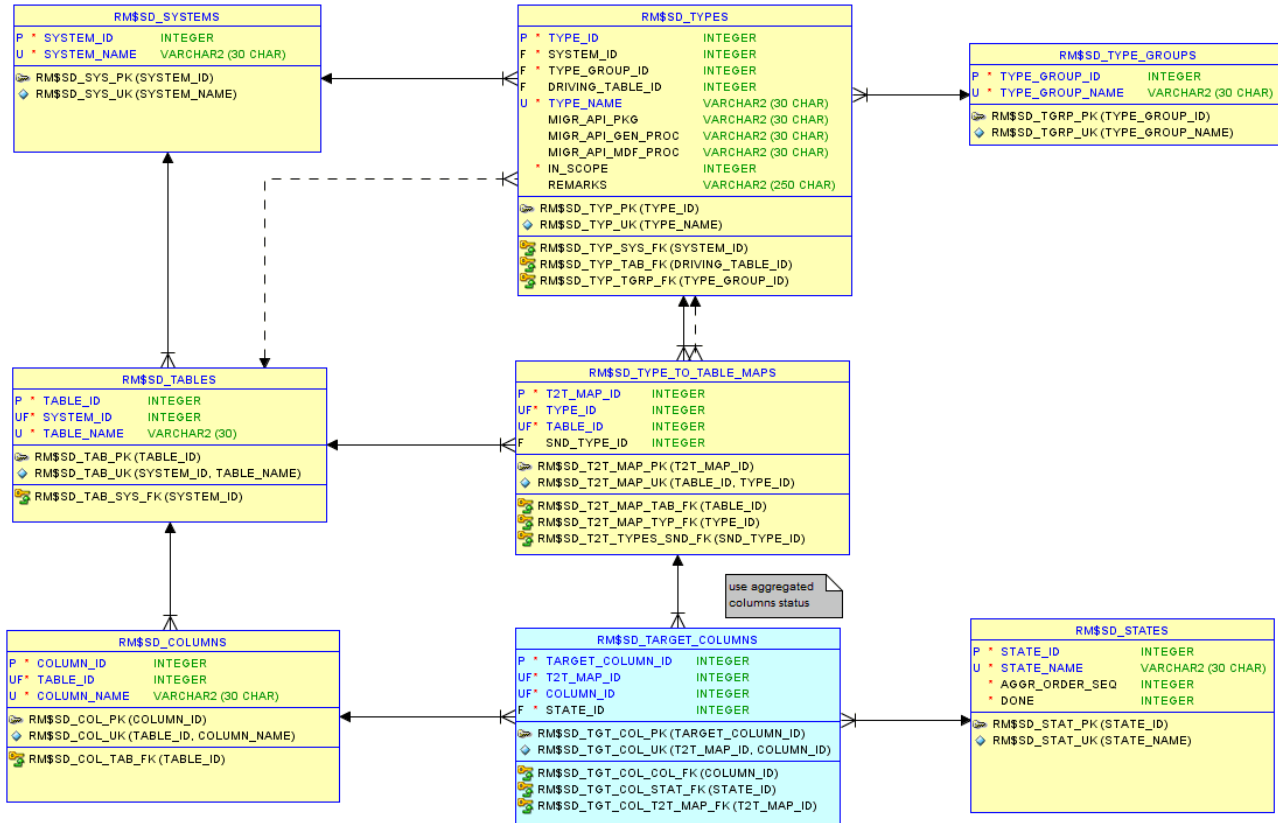
RM\$SD_STATES	
P *	STATE_ID INTEGER
U *	STATE_NAME VARCHAR2(30 CHAR)
	AGGR_ORDER_SEQ INTEGER
	DONE INTEGER
RM\$SD_STAT_PK (STATE_ID) RM\$SD_STAT_UK (STATE_NAME)	

# ■ 1. Source Columns – Beispiel

column_name	table_name	done	state_name	percent
anredec	d2k_ba_zreg	1	completed	100%
versandart	d2k_ba_zreg	0	missing	0%



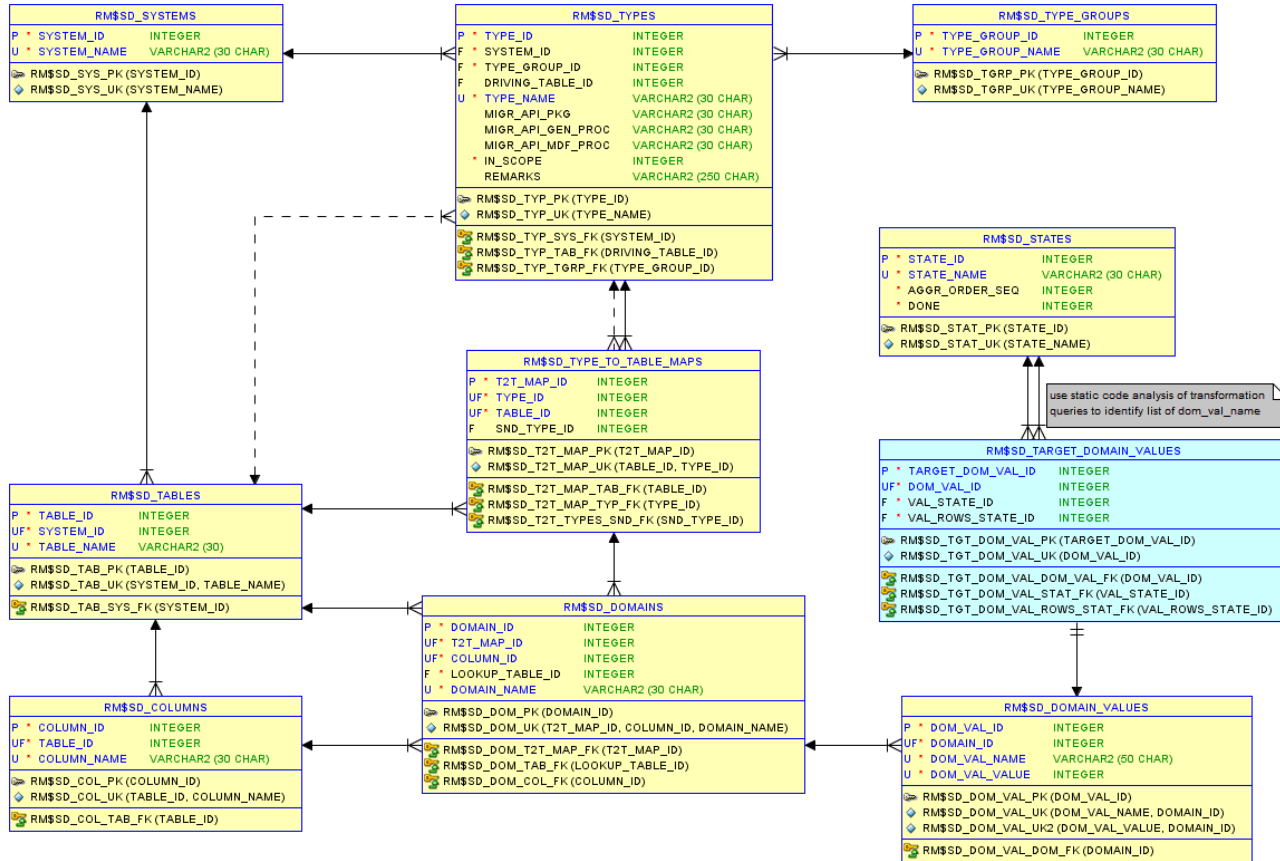
## 2. Target Columns (Avaloq)



## ■ 2. Target Columns – Beispiel

column_name	table_name	Type_name	done	state_name	percent
i_addr_role_id	migrt_addr	addr	0	open – analysis/design	50%
i_floor_nr	migrt_addr	addr	0	missing	0%
i_salut_id	migrt_addr	addr	1	completed	100%
i_salut_letter_id	migrt_addr	addr	1	irrelevant	100%

# 3. Target Domain Values (gültige Werte)



### 3. Target Domain Values – Beispiel

dom_val_name	domain_name	column_name	table_name	type_name	val_done	val_stat_name	val_rows_done	val_rows_stat_name	percent
cardctr	addr_type	i_addr_type_id	migrt_addr	addr	0	missing	0	missing	0%
country	addr_type	i_addr_type_id	migrt_addr	addr	0	missing	0	missing	0%
electrical	addr_type	i_addr_type_id	migrt_addr	addr	0	open – analysis/design	1	completed	75%
match	addr_type	i_addr_type_id	migrt_addr	addr	0	missing	0	missing	0%
postal	addr_type	i_addr_type_id	migrt_addr	addr	0	open – analysis/design	1	completed	75%
realty	addr_type	i_addr_type_id	migrt_addr	addr	0	missing	0	missing	0%
m	addr_type	i_obj_class_id	migrt_class	addr	1	completed	1	completed	100%

# Visualisieren

# ■ Google TreeMap Beispiel



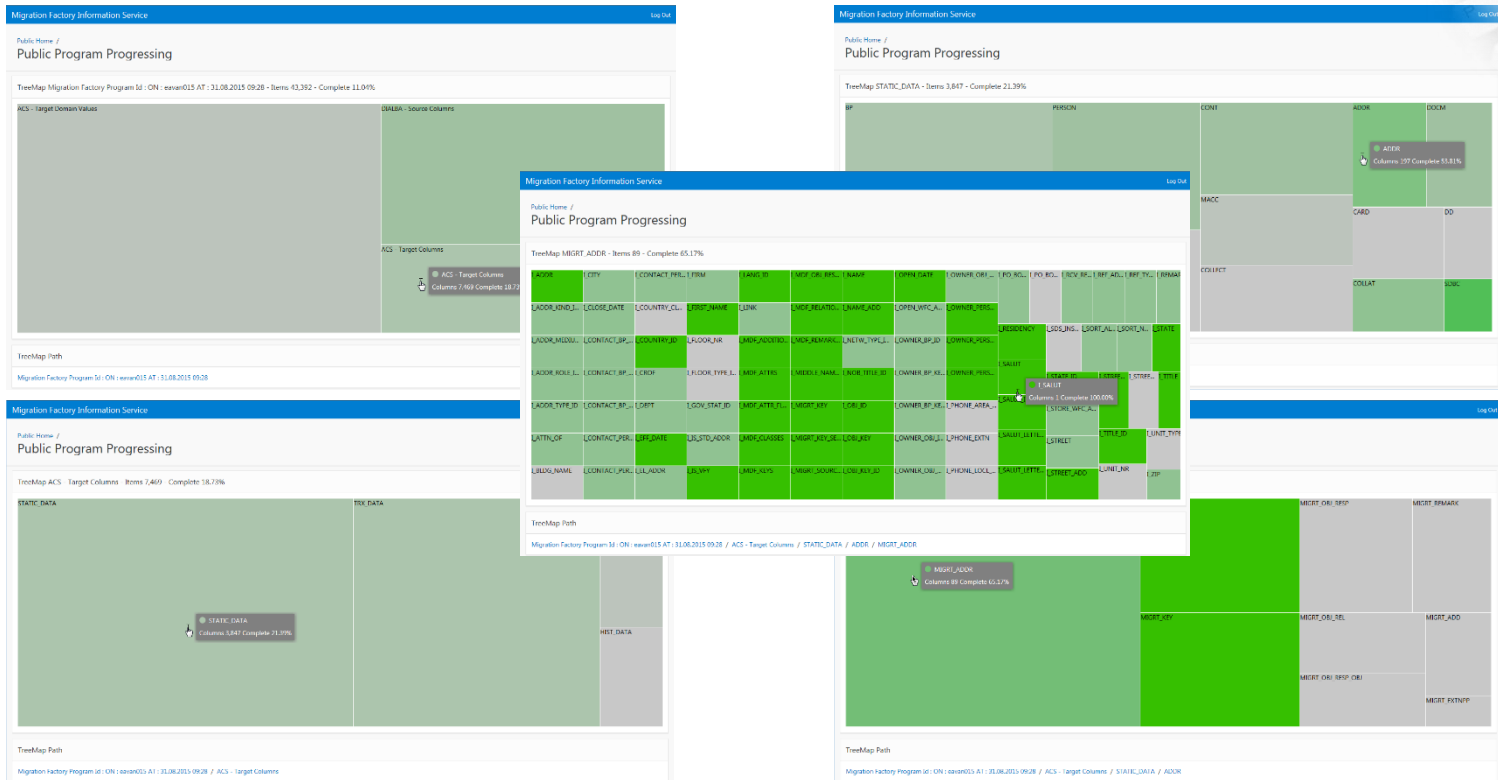
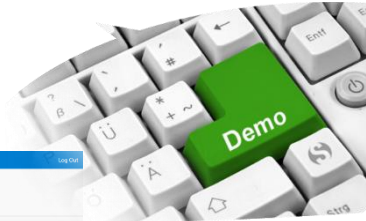
- Erste Dimension
  - Bestimmt die Grösse der Fläche
  - Beispiel: Umsatz
- Zweite Dimension
  - Bestimmt die Farbe
  - Verlauf von "Rot" bis "Grün"
  - Scala ist oben Rechts angegeben
  - Beispiel: Wachstum im Vergleich zur Vorperiode
- Daten als Hierarchie (rekursive Struktur)

# ■ Target Columns als Google Tree Map



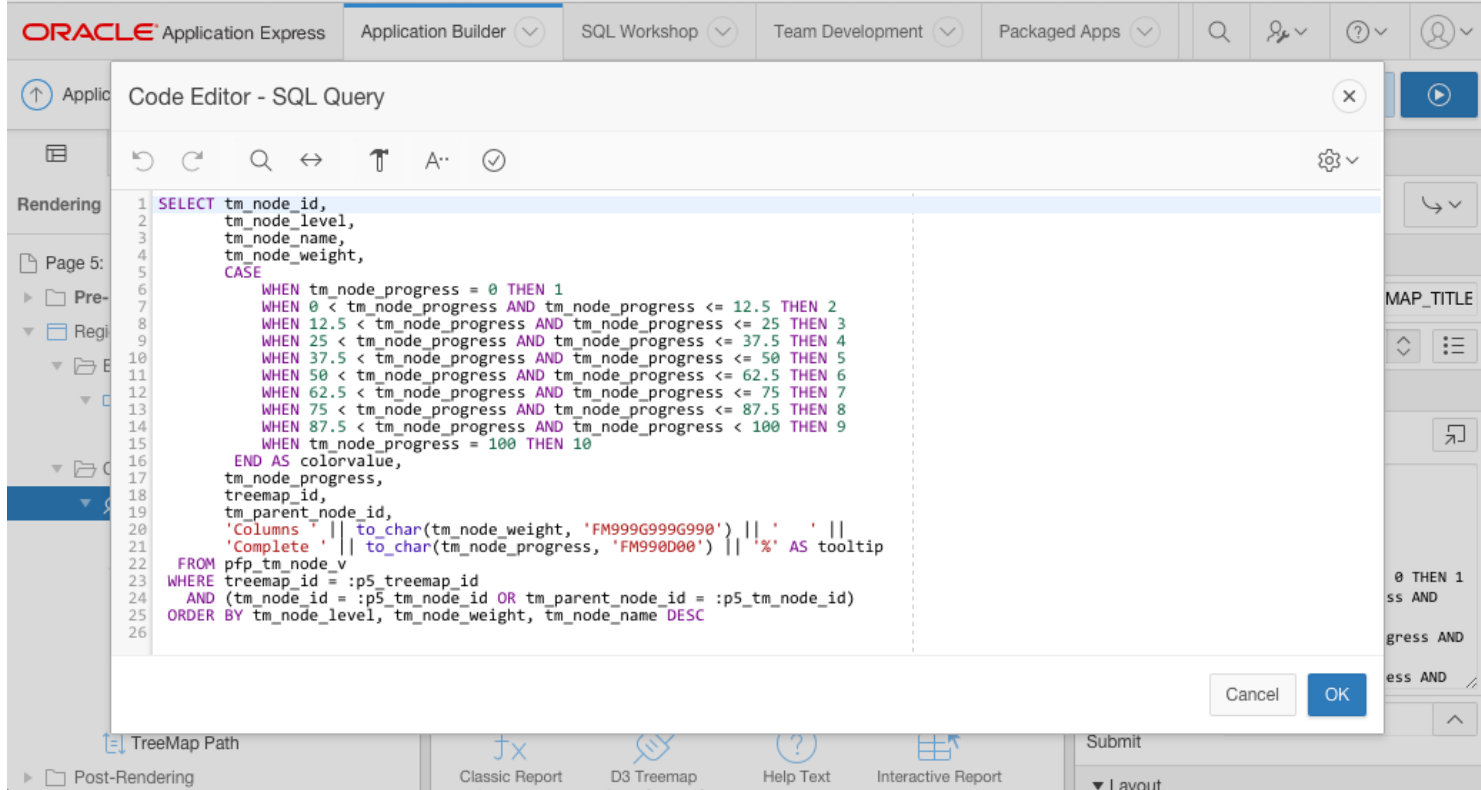
- Erste Dimension
  - Anzahl Columns
  - Bestimmt die Grösse der Fläche
- Zweite Dimension
  - Anzahl "erledigte" Columns
  - Bestimmt die Farbe der Fläche
  - Verlauf von "Grau" bis "Grün"

# Target Columns als APEX Tree Map (d3.js)





# SQL Query für TreeMap



The screenshot shows the Oracle Application Express interface. The main window is titled "Code Editor - SQL Query" and contains the following SQL query:

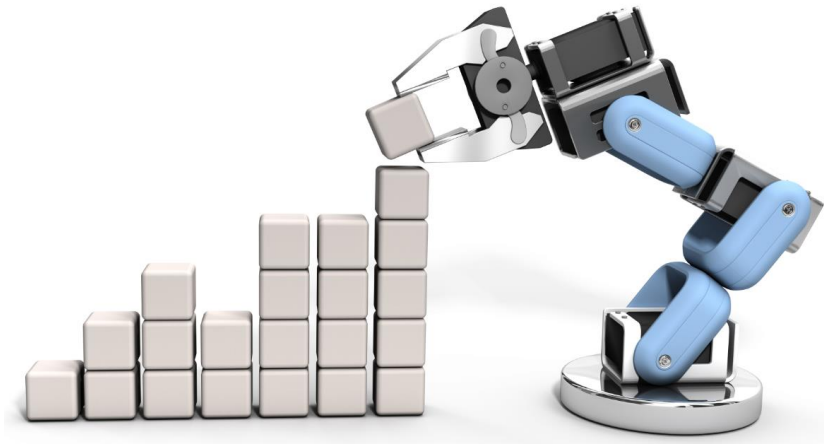
```
1 SELECT tm_node_id,
2        tm_node_level,
3        tm_node_name,
4        tm_node_weight,
5        CASE
6          WHEN tm_node_progress = 0 THEN 1
7          WHEN 0 < tm_node_progress AND tm_node_progress <= 12.5 THEN 2
8          WHEN 12.5 < tm_node_progress AND tm_node_progress <= 25 THEN 3
9          WHEN 25 < tm_node_progress AND tm_node_progress <= 37.5 THEN 4
10         WHEN 37.5 < tm_node_progress AND tm_node_progress <= 50 THEN 5
11         WHEN 50 < tm_node_progress AND tm_node_progress <= 62.5 THEN 6
12         WHEN 62.5 < tm_node_progress AND tm_node_progress <= 75 THEN 7
13         WHEN 75 < tm_node_progress AND tm_node_progress <= 87.5 THEN 8
14         WHEN 87.5 < tm_node_progress AND tm_node_progress < 100 THEN 9
15         WHEN tm_node_progress = 100 THEN 10
16       END AS colorvalue,
17        tm_node_progress,
18        treemap_id,
19        tm_parent_node_id,
20        'Columns ' || to_char(tm_node_weight, 'FM999G999G990') || ' ' ||
21        'Complete ' || to_char(tm_node_progress, 'FM990D00') || '% ' AS tooltip
22 FROM pfp_tm_node_v
23 WHERE treemap_id = :p5_treemap_id
24        AND (tm_node_id = :p5_tm_node_id OR tm_parent_node_id = :p5_tm_node_id)
25 ORDER BY tm_node_level, tm_node_weight, tm_node_name DESC
26
```

The interface includes a top navigation bar with "ORACLE Application Express", "Application Builder", "SQL Workshop", "Team Development", and "Packaged Apps". The left sidebar shows a tree view with "Page 5:" and "Pre-Rendering". The bottom toolbar contains "TreeMap Path", "Classic Report", "D3 Treemap", "Help Text", "Interactive Report", "Submit", and "Layout".

# Kernaussagen

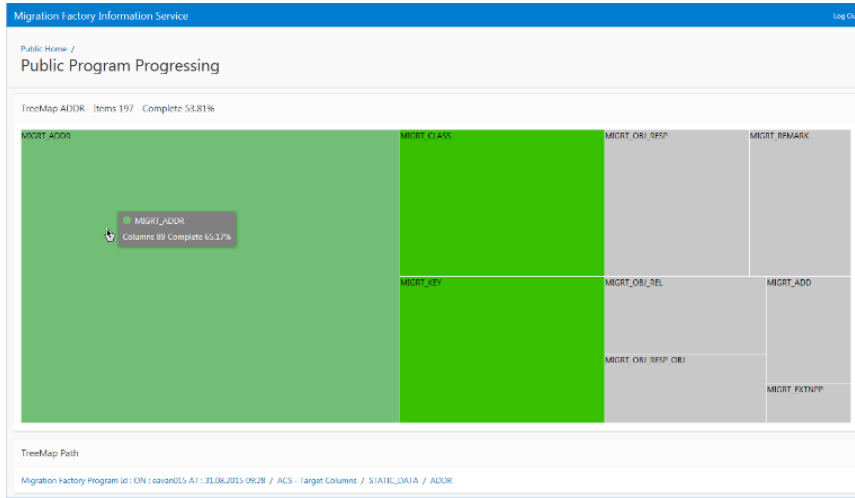
# ■ Vollautomatisierte Messungen

- Basis für laufend aktuelle Statistiken zum Programm-Fortschritt
- Einfacher durch "Model Driven Architecture"
- Trivadis PL/SQL Parser für statische Analyse von SQL Statements



# ■ Tree-Maps

- Visualisieren die Vollständigkeit der Migrationsprogramme gut
- Stellen grossen Datenmengen übersichtlich dar
- Ersetzen keine Detail-Reports (z.B. Differenz zwischen zwei Ständen)



# ■ Informations- und Planungsinstrument

- Beantwortet Fragen im Zusammenhang mit Nachvollzug und Data Lineage
- Ersetzt keine Tests zur Vollständigkeit und Korrektheit der Datenübernahme:
  - Datensätze (Business Objekte)
  - Columns generischer Tabellen in Quellsystemen



# Fragen und Antworten...

Philipp Salvisberg  
Senior Principal Consultant

Tel. +41 58 459 52 31  
philipp.salvisberg@trivadis.com

