

# “Was interessiert mich mein Geschwätz von gestern ... “

---

Versionierung historischer Daten im DWH  
mittels Kenntnisständen

## **IKB im Überblick**

---

- Finanzierungspartner für Mittelstand seit mehr als 90 Jahren
- ca. 700 Mitarbeiter
- Aktionär: Lone Star 100 %
- Bilanzsumme: 16,1 Mrd. €

## **Bank für den Mittelstand**

---

- Einzig bundesweit agierende Bank, die ausschließlich Firmenkunden begleitet
- Starke Kreditexpertise
- Langjährige Erfahrung im Fördermittelkreditgeschäft
- Kapitalmarkt- und Beratungsdienstleistungen
- Zugang zu mehr als 4.000 Fokuskunden

## Technische Basis

---

- Oracle 12c Rel. 1 Enterprise Edition
- Partitioning Option; In Memory Option; Diagnostic- / Tuning-Pack
- Virtual Private Database; Error Logging
- Fujitsu-Siemens Primenergy BX920S2; x86, 64 Bit, 16 CPU, 512 GB RAM; Suse Enterprise Linux 11.2

## Struktur

---

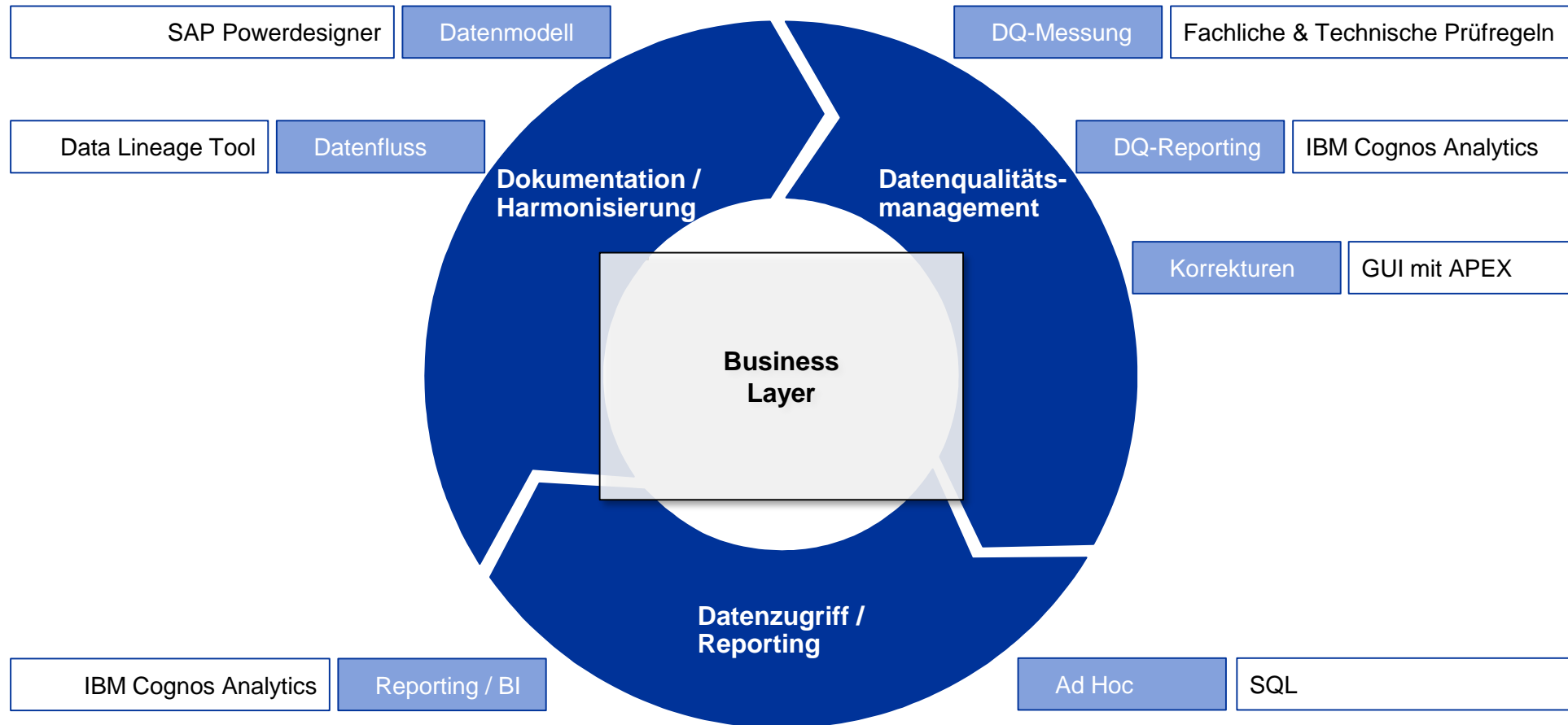
- Klassischer Aufbau mit Staging-Area, Core-Layer, Business-Layer
- Verschiedene Auslieferungs-Layer
- Service-Layer für Applikationen und aktive Datamarts

## ETL / Applikationen

---

- Laden aus Vorsystemen über eigenentwickeltes Framework (PL/SQL) per DB-Link, Filetransfer, Webservice
- Beladungsrythmus: overnight / near time
- Datenvolumen: ca. 6 TB
- Geschäftslogik: PL/SQL
- Online-Anwendungen: ApEX, Java
- Reporting/BI: Oracle EPM / Essbase / IBM Cognos Analytics

# Business Layer – Eigenschaften und Funktionalitäten



## Anforderungen FinRep

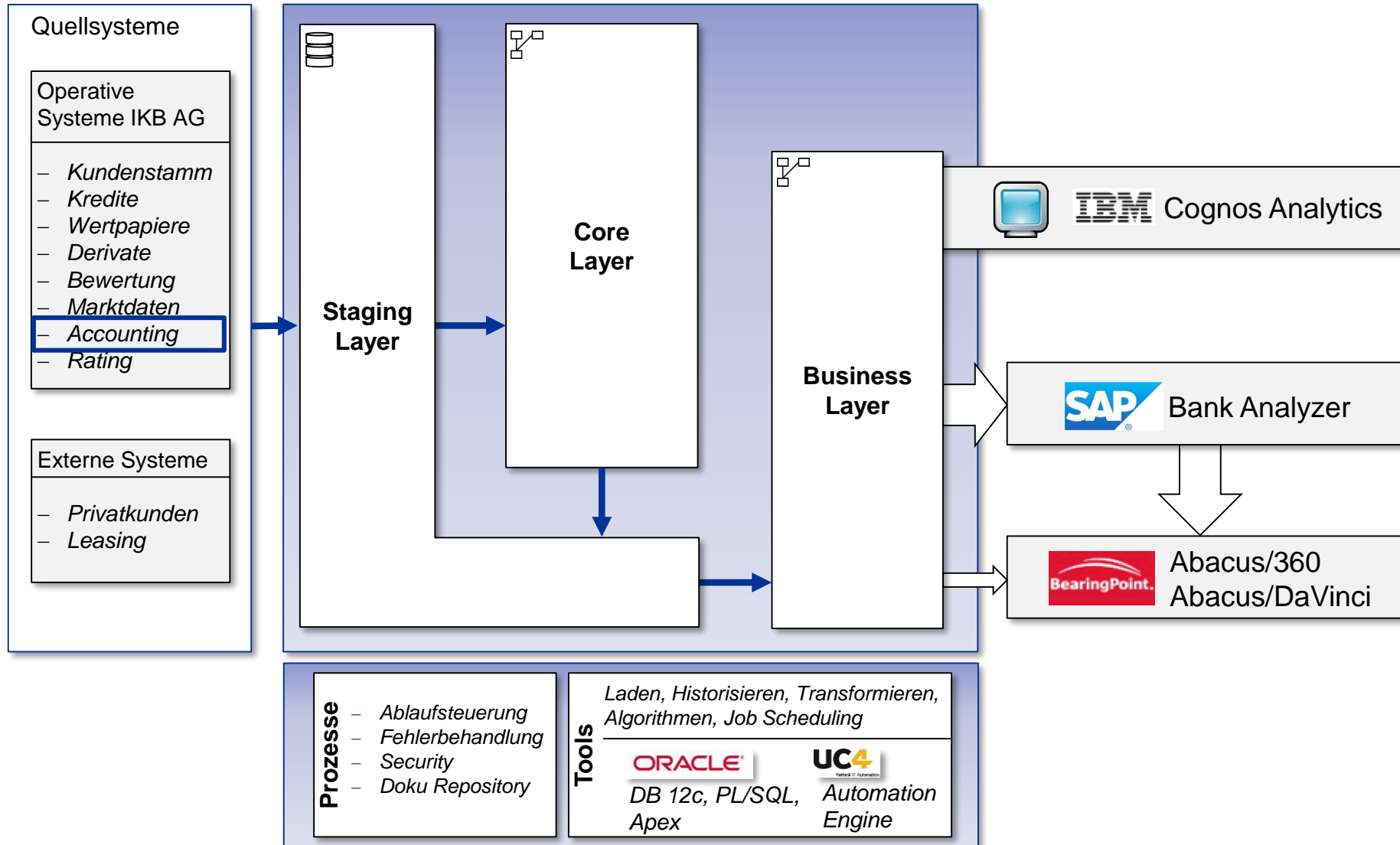
- Die EZB etabliert ein EU-weites, vergleichbares Finanzmeldewesen, basierend auf den FinRep Templates
- Für die FinRep Meldung müssen eine Vielzahl von Templates befüllt werden. Die Anforderungen aus den Templates erfordern Aufrisse von Bilanz- und GuV- Posten nach verschiedenen Merkmalen (Kontrahenten, Non- Performing, Forbearance, Sitzland, Branchen etc.) auf Einzelgeschäftsebene

Formular		F04.09 Non-trading debt instruments measured at a cost-based method				
Meldeeinheit		01-xx-xx-xx				
Stichtag						
Cluster						
Export Datum						
	Wenn sich die nationalen GAAP auf die BAD stützen	Nicht wertgeminderte Vermögenswerte	Wertgeminderte Vermögenswerte [Bruttobuchwert]	Einzelwertberichtigungen aufgrund von Ausfallrisiken	Allgemeine Wertberichtigungen aufgrund von Ausfallrisiken und Bankrisiken mit CRR art 4(95)	Buchwert
		010	020	030	040	Annex V Part 2.39
010	Schuldverschreibungen					
	020 Zentralbanken					
	030 Staatssektor					
	040 Kreditinstitute					
	050 Sonstige Finanzunternehmen					
	060 Nichtfinanzielle Unternehmen					
070	Darlehen und Kredite					
	080 Zentralbanken					
	090 Staatssektor					
	100 Kreditinstitute					
	110 Sonstige Finanzunternehmen					
	120 Nichtfinanzielle Unternehmen					
	130 Haushalte					
	140 NICHT ZU HANDEL SZWECKEN GEHALTENE, NACH EINER KOSTENMETHODE BEWERTETE SCHULDITITEL					

## Rolle des DWH

- Zusammenführen von Daten aus dem Accountingsystem mit Einzelgeschäftsdaten der bestandsführenden Systeme
- Rückwirkende Buchungen für einen vergangenen Stichtag berücksichtigen

# Struktur des IKB-DWH / Einbettung in IKB-Systemlandschaft



# Abbildung von Kenntnisständen im Datenmodell

## Tabellen wurden „kenntnisstandsfähig“ gemacht mit doppelter Historisierung

- Fachliche Gültigkeit
  - VF\_DAT („valid from“) + VT\_DAT („valid to“) für Gültigkeitszeitraum oder
  - STICHTAG für tägliche Datenablage
- Technische Gültigkeit (sekundengenau) mit VF\_KENNTNIS\_DAT + VT\_KENNTNIS\_DAT
  - VF\_KENNTNIS\_DAT entspricht Erstellungsdatum eines Datensatzes
  - VT\_KENNTNIS\_DAT entspricht Änderungsdatum eines Datensatzes

◇ SAPIDFS_ID	◇ VF_DAT	◇ VT_DAT	◇ VF_KENNTNIS_DAT	◇ VT_KENNTNIS_DAT	◇ SACHKONTO_NR	◇ DEBITOR_KTO_NR_EG
574930	31.03.2018 00:00:00	31.12.4000 00:00:00	16.04.2018 12:41:25	31.12.4000 00:00:00	6820200009	4104012015
674928	31.03.2018 00:00:00	31.12.4000 00:00:00	16.04.2018 12:41:25	31.12.4000 00:00:00	7400000122	2001018077
674929	31.03.2018 00:00:00	31.12.4000 00:00:00	16.04.2018 12:41:25	31.12.4000 00:00:00	7400000122	2001018371
674930	31.03.2018 00:00:00	31.12.4000 00:00:00	16.04.2018 12:41:25	31.12.4000 00:00:00	1140000076	3001007916
674931	31.03.2018 00:00:00	31.12.4000 00:00:00	16.04.2018 12:41:25	31.12.4000 00:00:00	7400000122	2001018170

## Mögliche Abfrage

```
SELECT *
  FROM sap_fi_salden s
 WHERE DATE'2018-12-31' BETWEEN s.vf_dat AND s.vt_dat
        AND SYSDATE BETWEEN s.vf_kenntnis_dat AND s.vt_kenntnis_dat;
```

# Ausblenden alter Kenntnisstände mittels VPD

## Problem

Viele Tabellen wurden „kenntnisstands-fähig“ gemacht

→ an diversen Programmstellen müsste Bedingung ergänzt werden:

```
SYSDATE BETWEEN vf_kennntnis_dat AND vt_kennntnis_dat
```

## Lösung

- Einsatz von VPD zur automatischen Einschränkung auf den aktuellen Kenntnisstand
- Achtung bei Materialized Views!

OBJECT_OWNER	OBJECT_NAME	POLICY_GROUP	POLICY_NAME	PF_OWNER	PACKAGE	FUNCTION	SEL	INS	UPD	DEL
DWHDTL	SAP_FI_SALDEN	SYS_DEFAULT	SAP_FI_SALDEN_KT	DWHTEC	PK_DWHO_VPD	FK_GET_VPD_FILTER_KENNTNIS	YES	YES	YES	YES

```
FUNCTION fk_get_vpd_filter_kennntnis (i_vcTabOwner IN VARCHAR2, i_vcTab IN VARCHAR2)
RETURN VARCHAR2
IS
  c_vcDatFormat CONSTANT VARCHAR2(16) := 'YYYYMMDDHH24MISS';
  d_Datum DATE := fk_get_erkennntnisdatum;
BEGIN
  IF USER = 'CGN_BL' THEN
    RETURN NULL;
  ELSIF d_Datum = c_dUnendlich THEN
    RETURN 'TO_DATE(''31.12.4000'', 'DD.MM.YYYY') = '||i_vcTab||'.vt_kennntnis_dat';
  ELSIF d_Datum IS NULL THEN
    RETURN NULL;
  ELSE
    RETURN 'TO_DATE(''||TO_CHAR(d_Datum, c_vcDatFormat)||'', '||c_vcDatFormat||') BETWEEN '
      ||i_vcTab||'.vf_kennntnis_dat AND '||i_vcTab||'.vt_kennntnis_dat';
  END IF;
END fk_get_vpd_filter_kennntnis;
```

Erkenntnisdatum wird beim Initialisieren auf "unendlich" gesetzt, kann aber gezielt pro Session geändert werden



## Problem

Im Reporting sollen alle Kenntnisstände verfügbar sein

## Lösung

- Für den User CGN\_BL, mit dem Cognos auf die DB zugreift, keine VPD-Policy
- Protokollierung verschiedener Kenntnisstände eines Stichtags (nicht je Tabelle)

ID	STICHTAG	KENNTNIS_NR	KENNTNIS_AKTUELL_KZ	KENNTNIS_DAT	KOMMENTAR
8162	31.03.2018 00:00:00	1	N	01.08.2018 00:00:00	Initial
8451	31.03.2018 00:00:00	2	N	15.10.2018 08:10:57	rückwirkender Lauf (z. B. bei FI-Salden)
11455	31.03.2018 00:00:00	3	N	14.01.2019 08:53:27	Initial Befuellung von FX_UMRECHNUNGSKURS
12017	31.03.2018 00:00:00	4	J	21.01.2019 14:43:31	Korrektur der Kennzahl DIABKUMB, nachdem

- Anlage von Views zur Integration der Kenntnisstände in Reporting-Entitäten

```
CREATE OR REPLACE VIEW vw_eg_merkmale AS
SELECT bl.*
      , dk.id kenntnis_id
      , dk.kenntnis_dat
      , dk.kenntnis_nr
FROM dgm_kenntnisstaende dk
JOIN eg_merkmale bl
  ON bl.stichtag = dk.stichtag
  AND dk.kenntnis_dat BETWEEN bl.vf_kenntnis_dat AND bl.vt_kenntnis_dat
```

- Join zwischen Entitäten immer auch über KENNTNIS\_ID
- Vorabauswahl eines Kenntnisstands vor eigentlichem Reporting

## Gründe für Entstehung neuer Kenntnisstände

- Erstmalige Verarbeitung eines Stichtags im Nachtbatch
- Manuelle Korrekturen über eine APEX-Maske

EG\_MERKMALE

Zurück

Go Actions Edit Änderungen abbrechen Änderungen speichern Freigeben (FB) Freigeben (DQM)

NICHT EDITIERBAR

<input checked="" type="checkbox"/>	Stichtag	EGKEY ID	EG IDENT	Herkunft	Kommentar	Verzug Tage	Verzug ab Dat	Valuation Unit
<input checked="" type="checkbox"/>	31.12.2018	749	FGK_3000558566_12...	IKBPRIVAT	-	-	-	-

1 rows selected

Prüfungen

**Simulation Methoden**  
Keine wartenden Änderungen in der Korrekturschicht!  
Simulation wird nicht gestartet!

**ZTV Prüfungen**  
Prüfung zum Stichtag 20.03.2019  
keine Auffälligkeit!  
Prüfung zum Ultimo 31.12.2018  
keine Auffälligkeit!

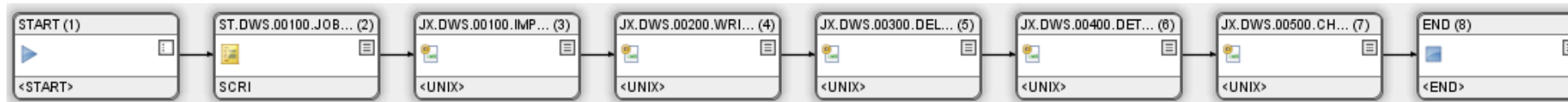
- Manuelle Änderungen auf der Datenbank
  - Nur im Notfall
  - Einfügen des Kenntnisstands in DQM\_KENNTNISSTAENDE könnte vergessen werden
- Rückwirkende Beladung einer oder mehrerer Tabellen (automatisiert oder durch manuellen Jobplan-Aufruf)

## Verarbeitungsschritte bei jeder Aktualisierung über Metadaten-gestütztes Verfahren

1. Speicherung des Verarbeitungszeitpunkts (sekundengenau)
2. Erstellung einer neuen Stage-Tabelle (z.B. DWHSTG.SAP\_FI\_SALDEN\_1582000)
3. Einfügen des neuen Datenstands in die neue Tabelle inkl. Fehler-Logging mit Zeitstempel aus (1)

```
LOG ERRORS INTO ERR$_M_SAP_FI_SALDEN (  
  TO_CHAR(pk_dwh0_dwsstd.sf_g_dlauf_dat  
  | pk_dwh0_dwsstd.sf_format_ora_err_tag))  
REJECT LIMIT UNLIMITED
```





4. Deltabildung im Vergleich zu vorherigem Datenstand in eigener Stage-Tabelle (DWHSTG.SAP\_FI\_SALDEN\_DLT)
  - Bisher: Verwendung der Stage-Tabelle des Vortags
  - Neue Option: Verwendung des Datenstands in der Zieltabelle
5. Übernahme der Änderungen in die Zieltabelle inkl. Fehler-Logging mit Zeitstempel aus (1)
6. Prüfung auf Existenz von Fehler-Datensätzen mit Zeitstempel aus (1)
  - Info-E-Mail
  - bedingter Prozessabbruch



# Delta-Ermittlung bei rückwirkendem Laden für Gültigkeitsspanne

## Herausforderungen bei der Delta-Ermittlung

Gültigkeitszeiträume mit VF\_DAT + VT\_DAT müssen aufgeteilt werden

SAPIDFS_ID	VF_DAT	VT_DAT	VF_KENNTNIS_DAT	VT_KENNTNIS_DAT	FACHL_ATTRIBUT	Vorgang
316734	17.10.2018	31.12.4000	18.10.2018 01:42:25	31.12.4000 00:00:00	123	
						Terminierung des Kenntnisstands
316734	17.10.2018	31.12.4000	18.10.2018 01:42:25	05.01.2019 01:40:37	123	
						Einfügen neuer Kenntnisstände
316734	17.10.2018	30.12.2018	05.01.2019 01:40:38	31.12.4000 00:00:00	123	Zeitraum davor
316734	31.12.2018	31.12.2018	05.01.2019 01:40:38	31.12.4000 00:00:00	456	Relevanter Stichtag
316734	01.01.2019	31.12.4000	05.01.2019 01:40:38	31.12.4000 00:00:00	123	Zeitraum danach

# Delta-Ermittlung bei rückwirkendem Laden für einzelnen Stichtag

## Herausforderungen bei der Delta-Ermittlung

Bei Vergleich mit der Zieltabelle darf nur der zum fachlichen Stichtag gültige Stand verwendet werden

- ➔ Kennzeichnung von Vergleichsspalten (z.B. STICHTAG) oder
- ➔ Anwendung von Filtern auf der Zieltabelle

```
update eg_acc_kennzahlen
  set egkey_abstaffelung_id = 1
  where egkey_id = 201483
  and stichtag = TRUNC(sysdate-1);

commit;

select *
  from eg_acc_kennzahlen_dlt
 order by 1,2,3,4,5,6,7;
```

Abfrageergebnis x

SQL | Alle Zeilen abgerufen:6 in 0 Sekunden

EGKEY_ID	KF_ID_EUR	SAPIDFS_ID	EGKEY_ABSTAFFELUNG_ID	KF_...	LADE_DAT	AKTION_KZ
1	201483 ABZIB	493403	1 (null)		01.03.2019 14:09:57	D
2	201483 ABZIB	493403	(null)	(null)	01.03.2019 14:09:57	I
3	201483 BUWB	4349	1 (null)		01.03.2019 14:09:57	D
4	201483 BUWB	4349	(null)	(null)	01.03.2019 14:09:57	I
5	201483 ZIEB	415450	1 (null)		01.03.2019 14:09:57	D
6	201483 ZIEB	415450	(null)	(null)	01.03.2019 14:09:57	I

# Vergleich von Kenntnisständen / Regressionstests (Aufruf)

- Regressionstests sind jetzt sehr einfach und nachträglich möglich
- Identifizierung geänderter Spalten mit eigenem Framework

```
BEGIN
pk_dwh0_vpd.pr_set_erkennnisdatum(NULL);
pk_lib.pr_diffs(
  ivcTabelle      => 'tcd_temp',
  ivcSqlAlt       => q'[SELECT * FROM eg_acc_kennzahlen
                      WHERE stichtag = DATE'2019-02-27'
                      AND '28.02.2019 14:09:56' BETWEEN vf_kenntnis_dat
                                                                AND vt_kenntnis_dat]',
  ivcSqlNeu       => q'[SELECT * FROM eg_acc_kennzahlen
                      WHERE stichtag = DATE'2019-02-27'
                      AND '28.02.2019 14:09:57' BETWEEN vf_kenntnis_dat
                                                                AND vt_kenntnis_dat]',
  ivcStaticCols   => 'stichtag, egkey_id, kf_id_eur, sapidfs_id',
  ivcExcludeCols  => 'anl_dat, anl_user, aend_dat, aend_user, prog_name'
                  || ', vf_kenntnis_dat, vt_kenntnis_dat, aktion_kz'
);
pk_dwh0_vpd.pr_set_erkennnisdatum();
END;
```

## Ergebnisse

```
SELECT *  
FROM tcd_temp;
```

tausgabe x Abfrageergebnis x

SQL | Alle Zeilen abgerufen: 3 in 0,003 Sekunden

STICHTAG	EGKEY_ID	KF_ID_EUR	SAPIDFS_ID	SPALTE	WERT_ALT	WERT_NEU
27.02.2019 00:00:00	201483	ABZIB	493403	EGKEY_ABSTAFFELUNG_ID 1		[leer]
27.02.2019 00:00:00	201483	BUWB	4349	EGKEY_ABSTAFFELUNG_ID 1		[leer]
27.02.2019 00:00:00	201483	ZIEB	415450	EGKEY_ABSTAFFELUNG_ID 1		[leer]

## Abfrage für weitergehende Analyse (z.B. Korrelation zu anderen Spalten)

```
WITH alt AS (SELECT "STICHTAG","EGKEY_ID","KF_ID_EUR","SAPIDFS_ID","EGKEY_ABSTAFFELUNG_ID", ...
             FROM (SELECT * FROM eg_acc_kennzahlen
                   WHERE stichtag = DATE'2019-02-27'
                   AND '28.02.2019 14:09:56' BETWEEN vf_kenntnis_dat
                                     AND vt_kenntnis_dat))
, neu AS (SELECT "STICHTAG","EGKEY_ID","KF_ID_EUR","SAPIDFS_ID","EGKEY_ABSTAFFELUNG_ID", ...
            FROM (SELECT * FROM eg_acc_kennzahlen
                  WHERE stichtag = DATE'2019-02-27'
                  AND '28.02.2019 14:09:57' BETWEEN vf_kenntnis_dat
                                     AND vt_kenntnis_dat))
, alles AS (SELECT 'ALT' quelle$, alt.* FROM alt
             UNION ALL SELECT 'NEU' quelle$, neu.* FROM neu)
SELECT t.*
       , a."QUELLE$", a."EGKEY_ABSTAFFELUNG_ID", a."KF_ABSTAFFELUNG_ID", ...
FROM tcd_temp t
LEFT JOIN alles a ON (1 = 1
                    AND t.EGKEY_ID = a.EGKEY_ID
                    AND t.KF_ID_EUR = a.KF_ID_EUR
                    AND t.STICHTAG = a.STICHTAG
                    AND t.SAPIDFS_ID = a.SAPIDFS_ID
                    AND DECODE(wert_neu,NULL,'ALT','NEU') = a.quelle$)
ORDER BY t.EGKEY_ID, t.KF_ID_EUR, t.STICHTAG, t.SAPIDFS_ID;
```



## Statistik

---

/\*

Gesamtanzahl: 3

Anzahl Schlüssel: 3

Anzahl alte Schlüssel: 0

Anzahl neue Schlüssel: 0

Max. Anzahl Spalten: 1

Avg. Anzahl Spalten: 1

Median Anzahl Spalten: 1

\*/

## Zusammenfassung

---

- Über die Statistik ist sofort ersichtlich, ob es Änderungen gab
- Änderungen sind sehr schnell abfragbar dank separater Tabelle
- Betroffene Spalten sind sehr schnell identifizierbar
- Korrelation zu anderen Spalten schnell ermittelbar
- Klassifizierung und somit Legitimation von Änderungen einfach möglich  
`WHERE spalte = 'EGKEY_ABSTAFFELUNG_ID' AND wert_alt < wert_neu * 1.5`
- Prozess ist flexibel einsetzbar
  - ... wurde z.B. im Rahmen der OWB-Ablösung durch PL/SQL genutzt
  - ... wird auch vom Fachbereich über eine APEX-Maske zu Reconciliation-Zwecken genutzt

## Vorteile

- Absicherung vor Datenverlust, da nichts gelöscht wird und nur VT\_DAT / VT\_KENNTNIS\_DAT aktualisiert werden
- Gesteigerte Nachvollziehbarkeit
- Erstmaliges und rückwirkendes Laden mit den gleichen Prozessabläufen
- Abnehmerspezifisches Reporting möglich mit Auswahl von Kenntnisstand
- Sehr einfache Regressionstests
- Einheitliche Dateiformate zum Test durch den Fachbereich
- Gesteigerte Entwicklungs- und Testgeschwindigkeit



## Nachteile

- Höherer Speicherplatzverbrauch durch Speicherung von Kenntnisständen, die häufig nicht mehr benötigt werden
- Performanceeinbußen durch
  - Overhead durch ständige Deltabildung
  - Zusätzliche Joins beim Reporting
  - Anwendung von VPD



## Ulrich Suhre

IKB Deutsche Industriebank AG  
Wilhelm-Bötzkens-Straße 1  
40474 Düsseldorf

Telefon +49 211 8221-4195  
E-Mail ulrich.suhre@ikb.de

## Timo Bergenthal

IKB Deutsche Industriebank AG  
Wilhelm-Bötzkens-Straße 1  
40474 Düsseldorf

Telefon +49 211 8221-4712  
E-Mail timo.bergenthal@ikb.de

Copyright:

März 2019

Herausgeber: IKB Deutsche Industriebank AG, Wilhelm-Bötzkens-Straße 1, 40474 Düsseldorf

Rechtsform: Aktiengesellschaft

Sitz: Düsseldorf

Handelsregister: Amtsgericht Düsseldorf, HR B 1130

Vorsitzender des Aufsichtsrats: Dr. Karl-Gerhard Eick

Vorsitzender des Vorstands: Dr. Michael H. Wiedmann

Vorstand: Claus Momburg, Dr. Jörg Oliveri del Castillo-Schulz, Dirk Volz