



## Effektivitätssatelliten

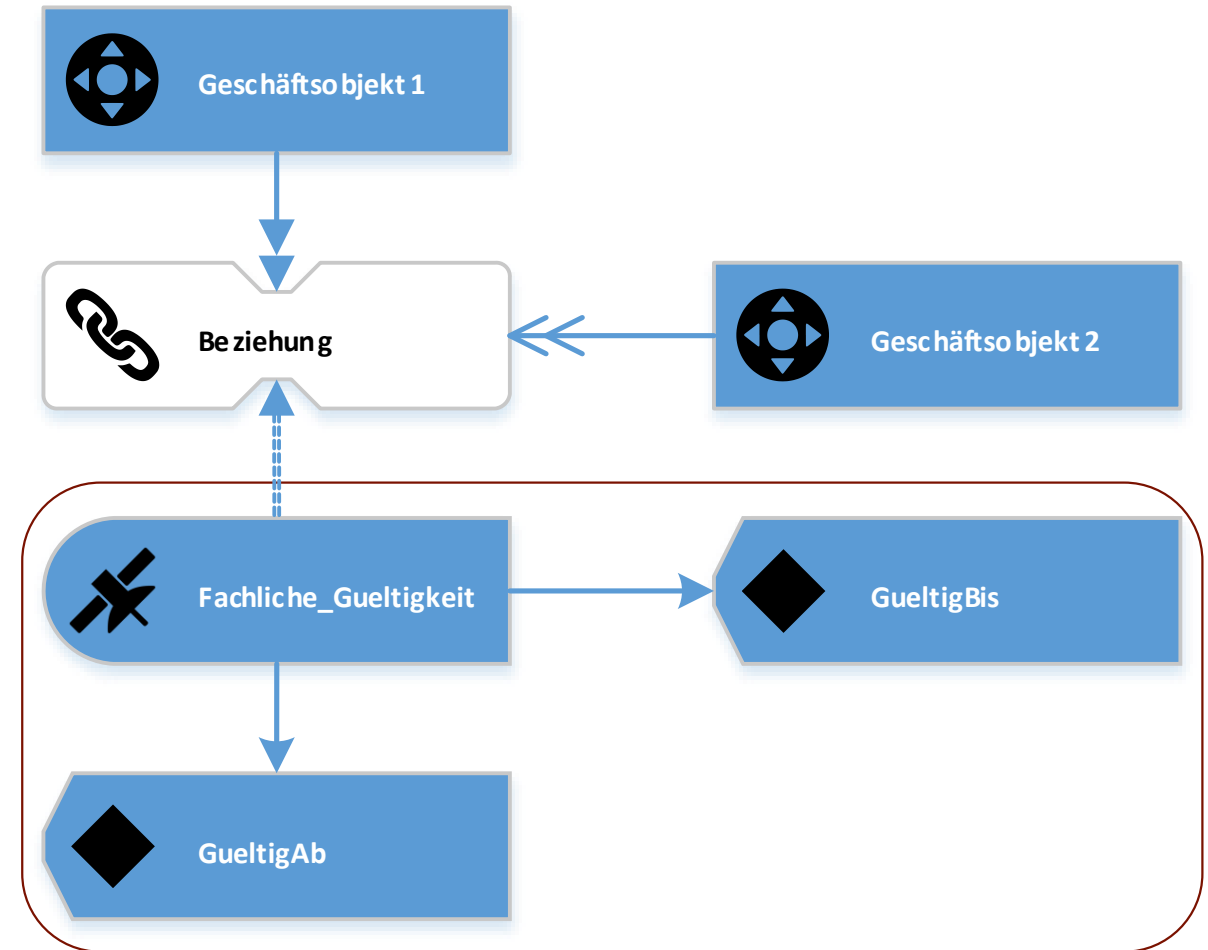
Beladung von Effektivitätssatelliten  
in  
Data Vault Umgebungen

# Content

- Sinn und Zweck
  - Definition
  - Variante: „Driving-Key LSAT“
- Beladungsmuster
  - Anforderungen
  - Testfälle
- Praktisches Beispiel
- Beispielimplementierung
- Erforderliche Metadaten
- Credits

# Definition

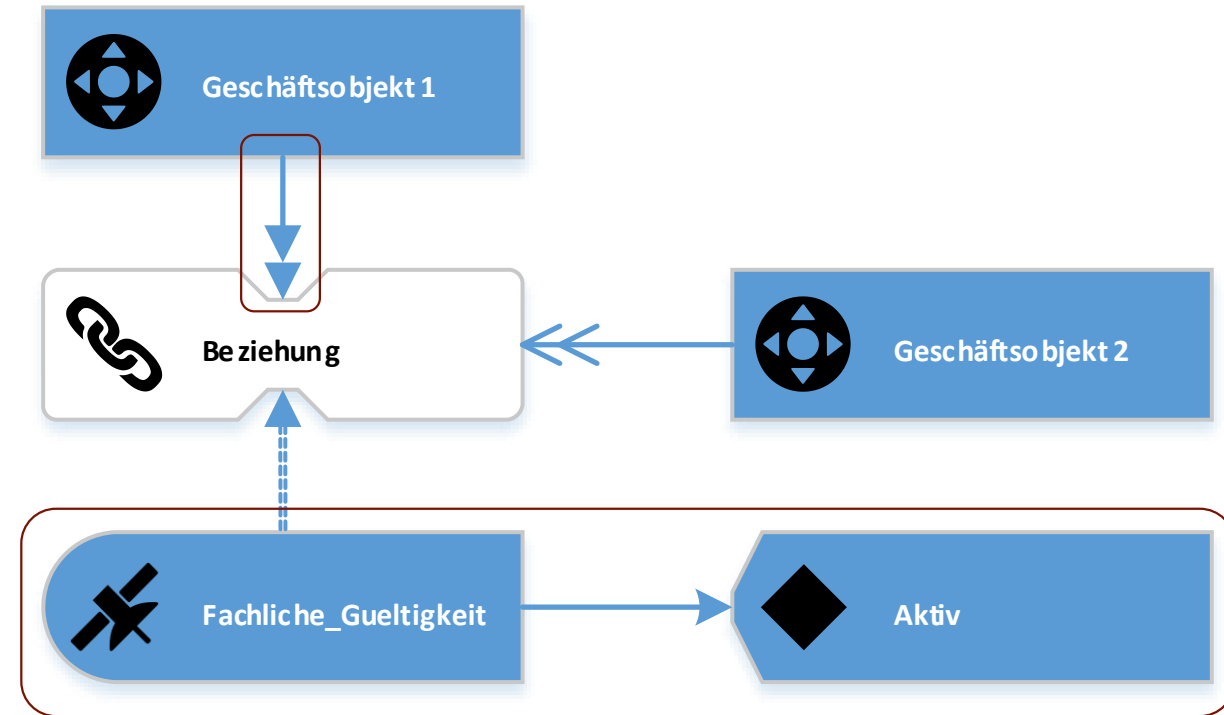
- Erfassen der **fachlichen Gültigkeit** von Beziehungen oder Geschäftsobjekten
- Gültigkeitsdaten (mind. Gültigkeitsbeginn) müssen von der Quelltablelle bereitgestellt werden
- Der Satellit enthält jederzeit einen aktiven Eintrag pro Beziehung im Link
- Das End-Dating im Satelliten basiert auf einem Driving-Key
- Der Verzicht führt zu Informationsverlust!



Eigene Darstellung (in Anlehnung an Linstedt & Olschimke, 2016, S. 145)

# Variante: „Driving-Key LSAT“

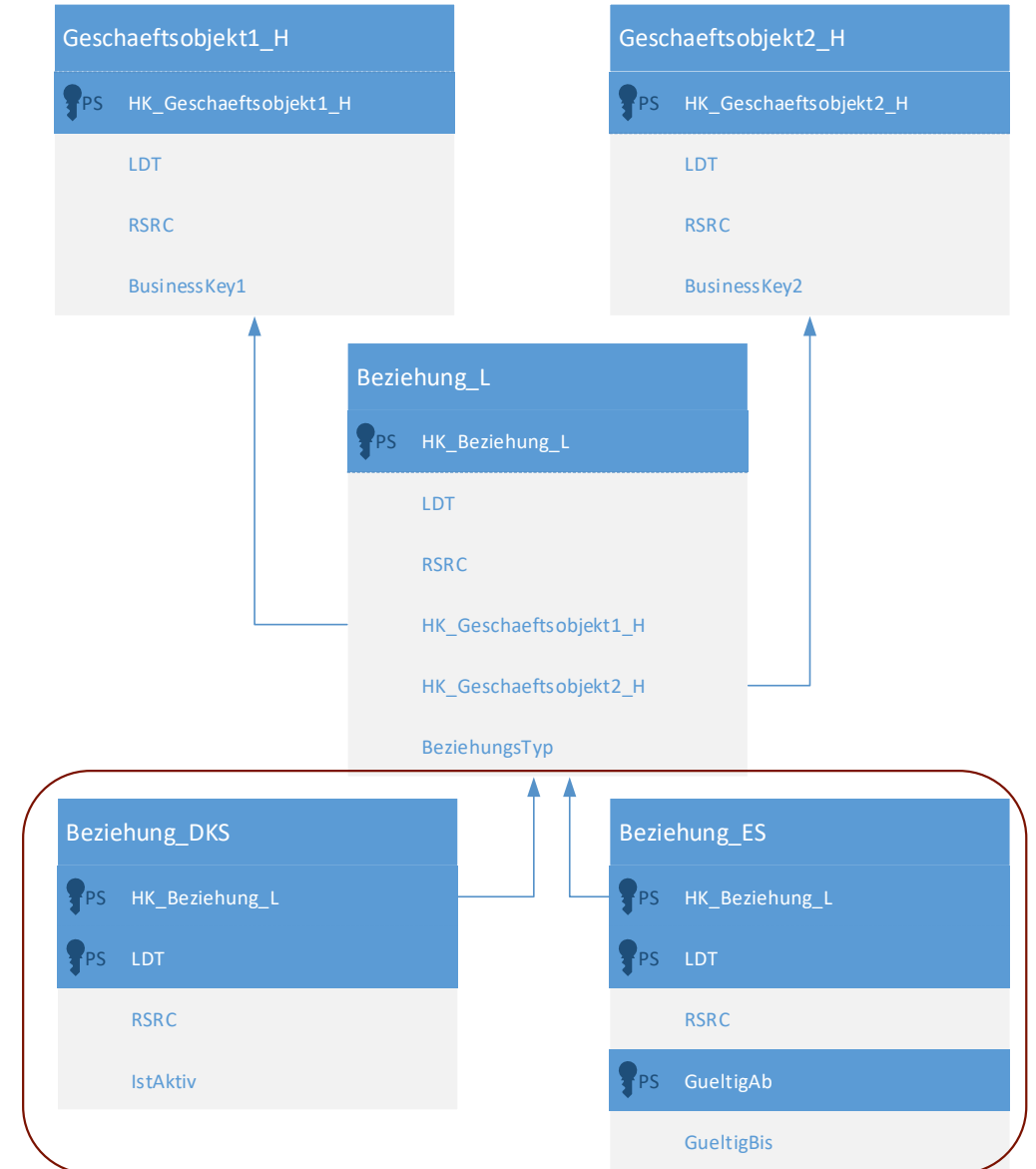
- Erfassen der fachlichen Gültigkeit von Beziehungen (oder Geschäftsobjekten) zum Zeitpunkt der Beladung
- Kommen zum Einsatz wenn die Quelltable keine fachlichen Gültigkeiten bereitstellt
- Der Satellit enthält jederzeit einen aktiven Eintrag pro Beziehung im Link
- Ein (mehrteiliger) Driving-Key muss identifiziert werden



Eigene Darstellung (in Anlehnung an Linstedt & Olschimke, 2016, S. 145)

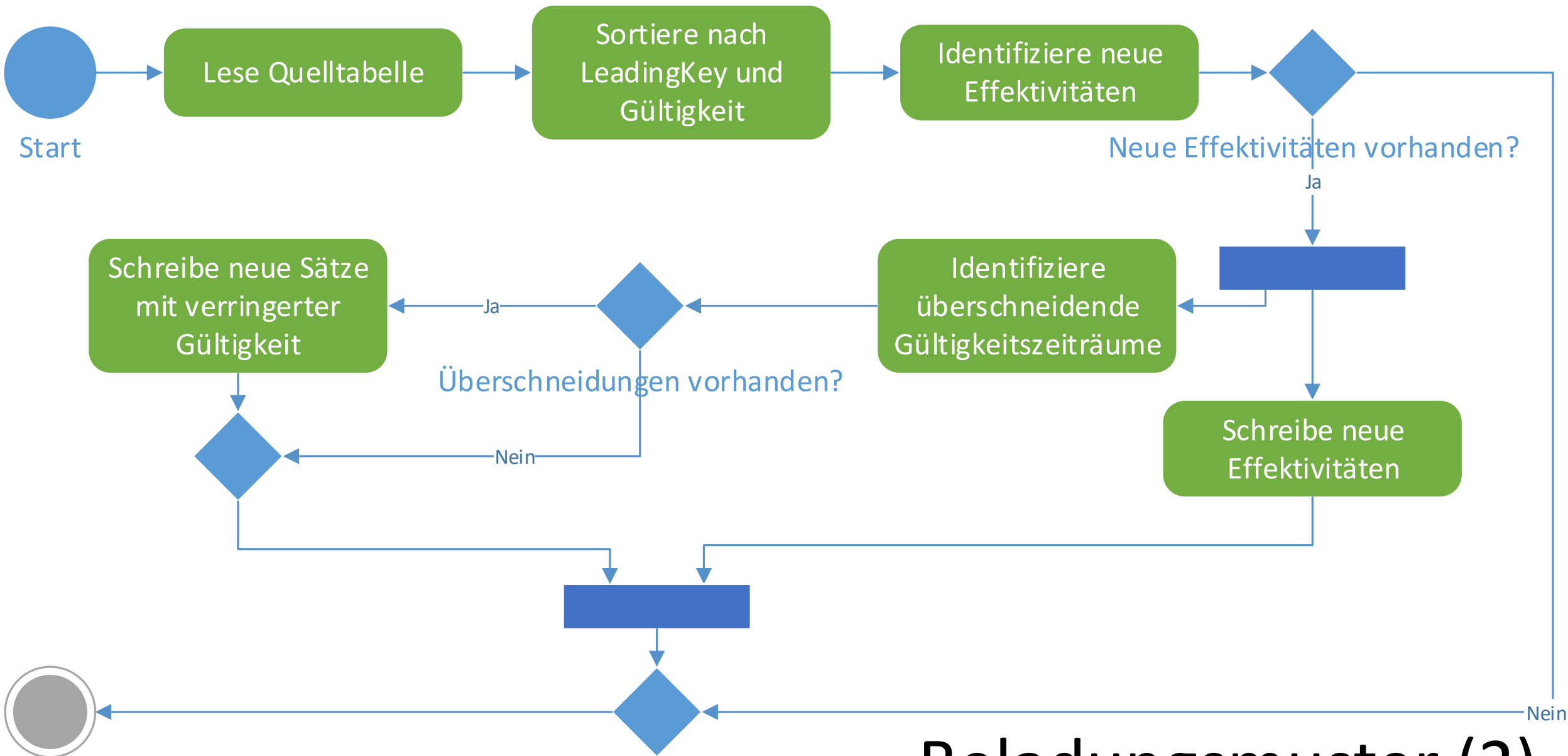
# Aufbau der Tabellen

- Keine Kontextinformationen in den Satelliten
- Aufbau nach Standardmuster
  - Hashdiff bietet keinen Mehrwert
- Driving-Key LSAT:
  - Ein Boolean-Attribut ob gültig zum Zeitpunkt der Beladung
- Effektivitätssatelliten:
  - Multi-Aktiv
  - Zwei Datetime-Attribute, die den Gültigkeitszeitraum (von – bis) definieren.



# Beladungsmuster (1): Anforderungen

- Beladungsstatements sollen generierbar sein.
- Insert-Only Statements
- Beladung in einem einzigen Schritt
- Überlappende Gültigkeitszeiträume in der Quelle sollen aufgelöst werden
- Beladung soll sowohl mit vollständigen als auch Inkrementen der Quelldaten funktionieren
- Physikalische Löschung von Beziehungen im Quellsystem sollen erkannt werden.
  - Dies ist bei einer inkrementellen Beladung allerdings nicht möglich!
- Möglichst performante Ausführung



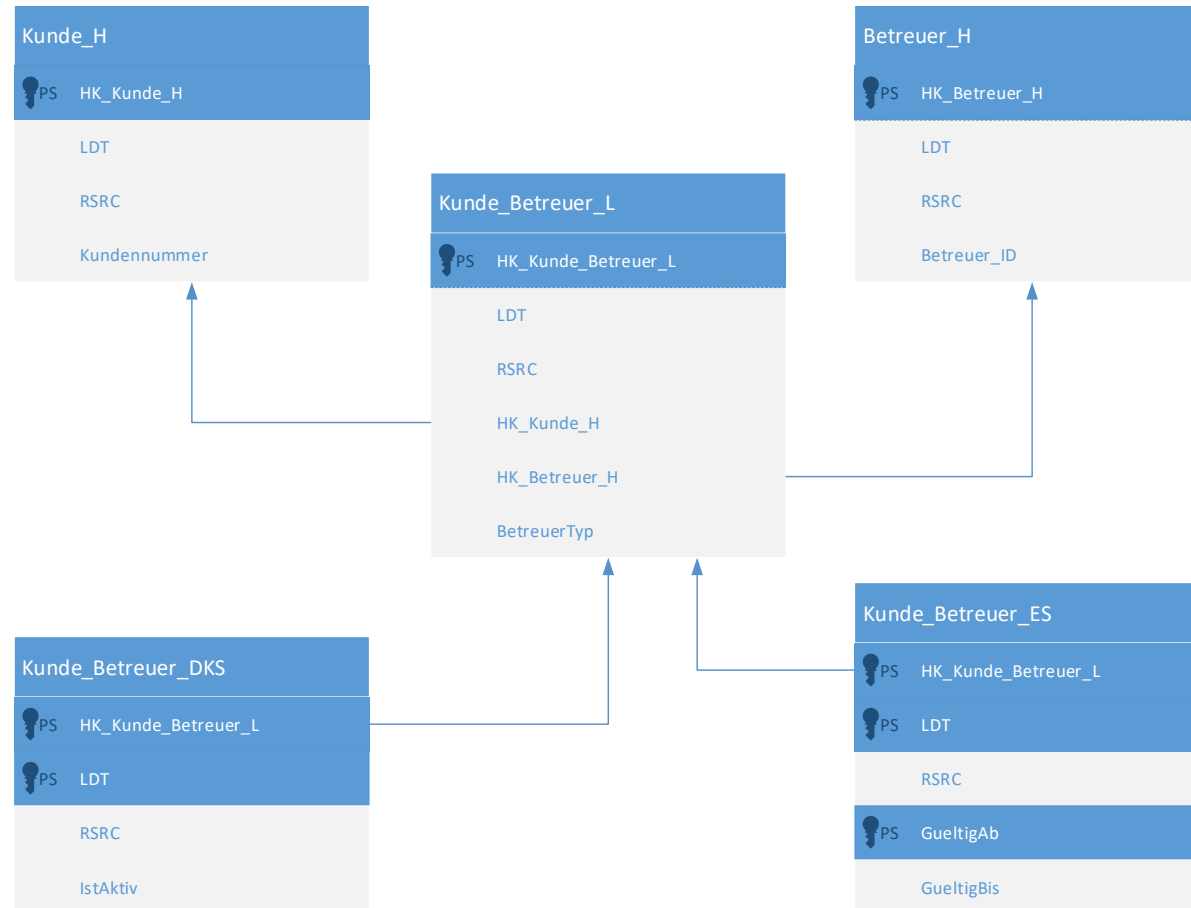
# Beladungsmuster (2)

# Beladungsmuster (3): Testfälle

- Gültigkeitsende der Beziehung 1A in Quelle = „05.03.2017“ und Gültigkeitsbeginn der Beziehung 1B in Quelle = „04.03.2017“
- Wechsel der Beziehung von 1A zu 1B und dann zurück zu 1A
  - Bei inkrementeller Beladung besonders fehleranfällig!
- Beziehung 1A mit Gültigkeitsbeginn „10.03.2017“ kommt in Quelle doppelt vor
  - mit und ohne unterschiedlichem Gültigkeitsende
- Gültigkeitsbeginn einer Beziehung liegt nach deren Gültigkeitsende
- Gültigkeitsbeginn in Quelle = „16.03.2017 12:30:00“ und Gültigkeitsende in Quelle = „16.03.2017“ (Ohne Zeitstempel)



# Praktisches Beispiel (1): Datenmodell



# Praktisches Beispiel (2)

## Stage

LDT	Kunde	Betreuer	Betreuer Typ	Von	Bis
1	1	A	1	05.01.	20.01.
1	2	B	1	01.01.	

## Link

HK_L	LDT	HK_H1	HK_H2	Typ
1;A;1;	1	1;	A;	1
2;B;1;	1	2;	B;	1

## Effektivitätssatellit

Hash	LDT	Von	Bis	LEDT
1;A;1;	1	05.01.	20.01.	
2;B;1;	1	01.01.	9999	

# Beispielimplementierung: Beladung (1)

```
WITH srcdata1 AS
(SELECT a.hk_kunde_betreuer_l AS linkkey
      ,a.hk_kunde_h AS leadingkey
      ,a.hk_betreuer_h AS deptkey
      ,a.von AS gueltigab
      ,least(coalesce(MIN(a.von) over(PARTITION BY a.hk_kunde_h ORDER BY a.von DESC rows BETWEEN
unbounded preceding AND 1 preceding) - INTERVAL '0.001' SECOND, DATE '9999-12-31')
      ,coalesce(a.bis, DATE '9999-12-31')) AS gueltigbis
FROM stage.v_kunde_betreuer a),
srcdata AS
(SELECT a.linkkey
      ,a.leadingkey
      ,a.deptkey
      ,a.gueltigab
      ,a.gueltigbis
FROM srcdata1 a
WHERE NOT EXISTS (SELECT NULL
                  FROM kunde_betreuer_bes y
                  WHERE y.hk_kunde_betreuer_l = a.linkkey
                        AND y.gueltigab = a.gueltigab
                        AND (y.gueltigbis = a.gueltigbis
                        OR a.gueltigbis = DATE '9999-12-31')
                        AND y.ledt = DATE '9999-12-31'))),
```

# Beispielimplementierung: Beladung (2)

```
effdata AS
(SELECT hextoraw(l.hk_kunde_betreuer_l) AS linkkey
, hextoraw(l.hk_kunde_h) AS leadingkey
, hextoraw(l.hk_betreuer_h) AS deptkey
, es.gueltigab
, es.gueltigbis
FROM v_kunde_betreuer_bes es
JOIN kunde_betreuer_l_l
ON l.hk_kunde_betreuer_l = es.hk_kunde_betreuer_l
WHERE es.ledt = DATE '9999-12-31'
AND es.gueltigab < es.gueltigbis),
newgueltigbis AS
(SELECT e.linkkey
, e.leadingkey
, e.deptkey
, e.gueltigab
, MIN(s.gueltigab) - INTERVAL '0.001' SECOND AS gueltigbis
FROM effdata e
JOIN srcdata s
ON s.leadingkey = e.leadingkey
WHERE NOT (s.deptkey = e.deptkey AND s.gueltigab = e.gueltigab)
AND s.gueltigab < e.gueltigbis
GROUP BY e.linkkey
, e.leadingkey
, e.deptkey
, e.gueltigab),
```

# Beispielimplementierung: Beladung (3)

newrows AS

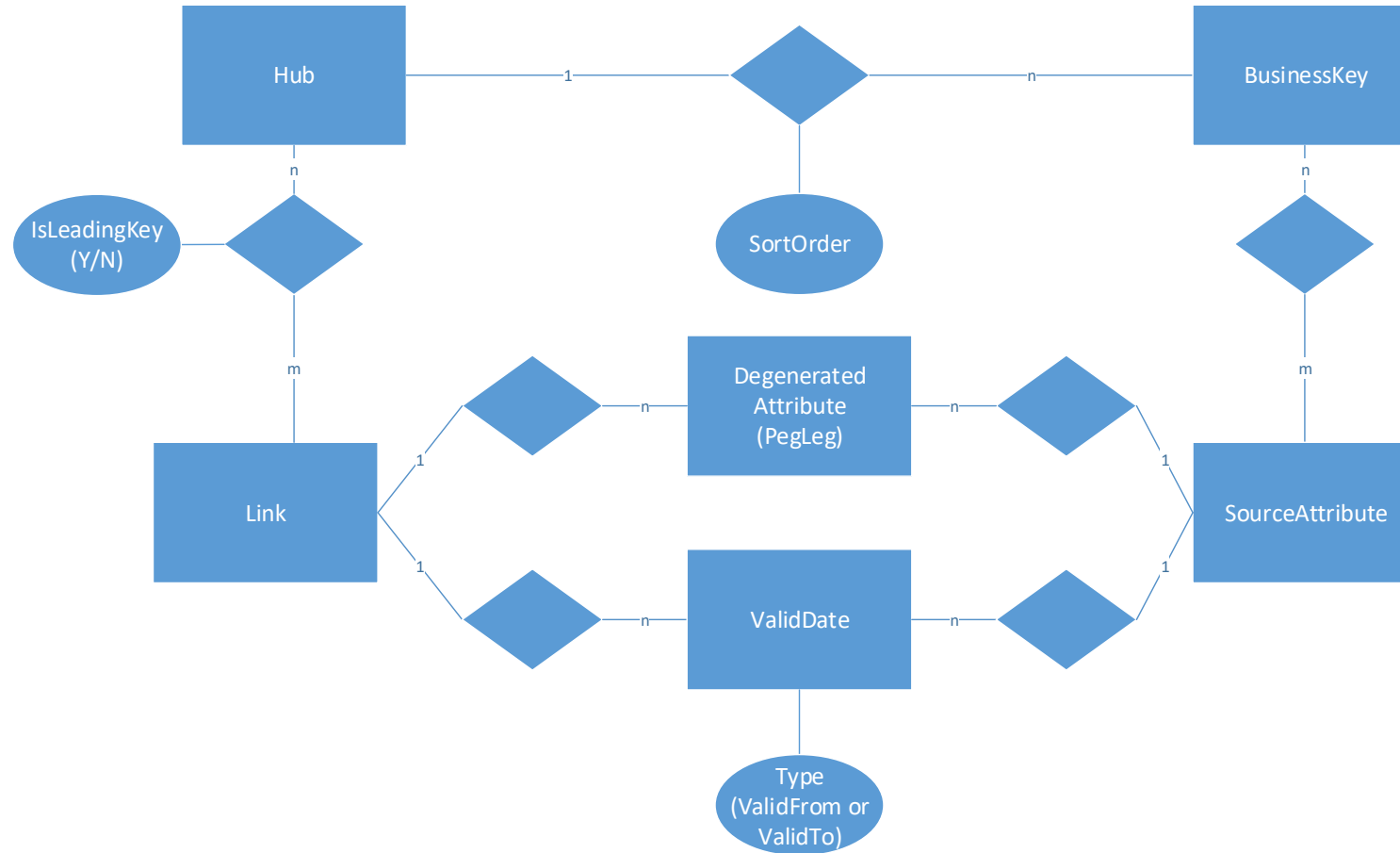
(SELECT \* FROM srcdata UNION ALL SELECT \* FROM  
newgueltigbis)

```
SELECT linkkey AS hk_kunde_betreuer_1  
      ,SYSDATE AS ldt  
      ,gueltigab  
      ,gueltigbis  
FROM newrows a
```

# Beispielimplementierung: Abfrage

```
SELECT kh.kundennummer
       ,bh.betreuer_id
       ,kbl.betreuertyp
FROM kunde_h kh
JOIN kunde_betreuer_L kbl
  ON kbl.hk_kunde_h = kh.hk_Kunde_h
JOIN v_kunde_betreuer_es kbes
  ON kbes.hk_kunde_betreuer_1 = kbl.hk_kunde_betreuer_1
 AND kbes.ledt = DATE '9999-12-31'
JOIN betreuer_h bh
  ON bh.hk_betreuer_h = kbl.hk_betreuer_h
WHERE SYSDATE BETWEEN kbes.gueutigab AND kbes.gueutigbis;
```

# Erforderliche Metadaten



# Credits

- Urheber des Konzeptes (Effektivitäts-)Satellit:
  - Dan Linstedt
  - Michael Olschimke
- Urheber des Konzeptes „Driving-Key LSAT“:
  - Roelant Vos
- Aufbau und Verbesserung der Beladungsmuster
  - Sheida Nobakht
  - Dr. Jörn Peter



Bildquelle: [www.scalefree.com](http://www.scalefree.com)



© Peer Maans Carlson





