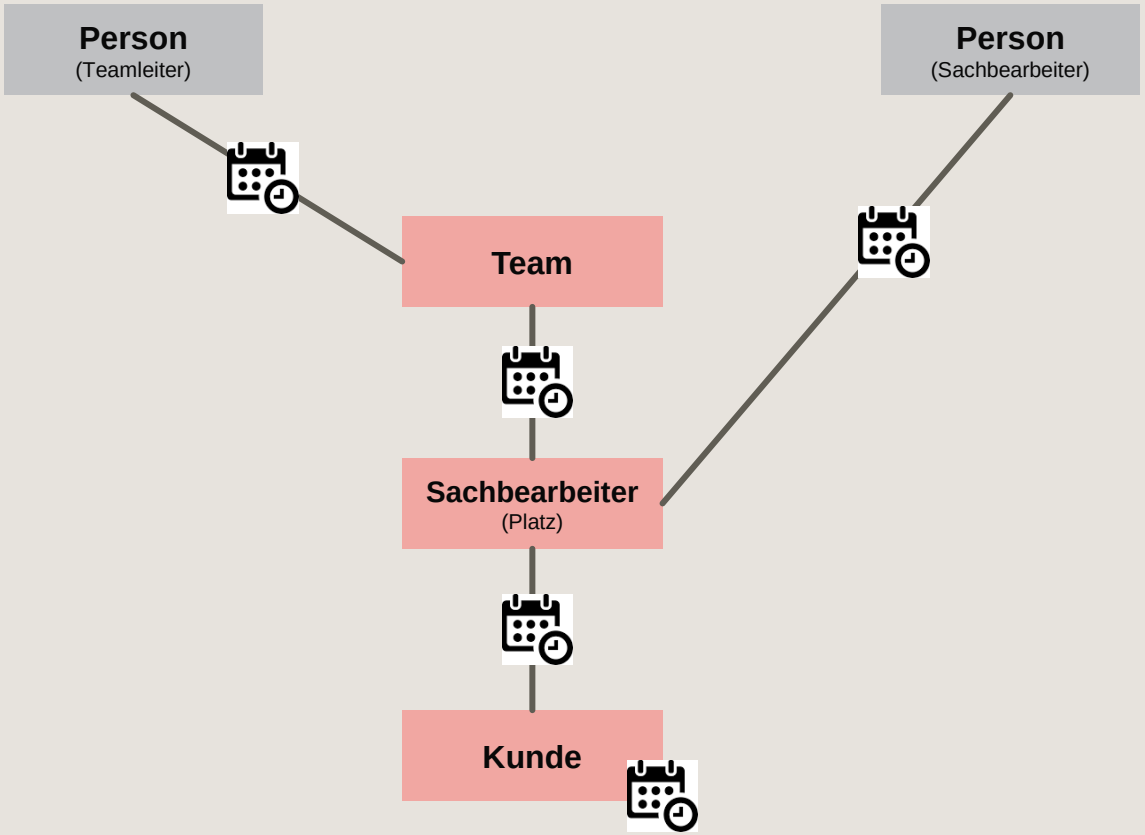
SQL-basierte SCD2-Versionierung hierarchischer Strukturen

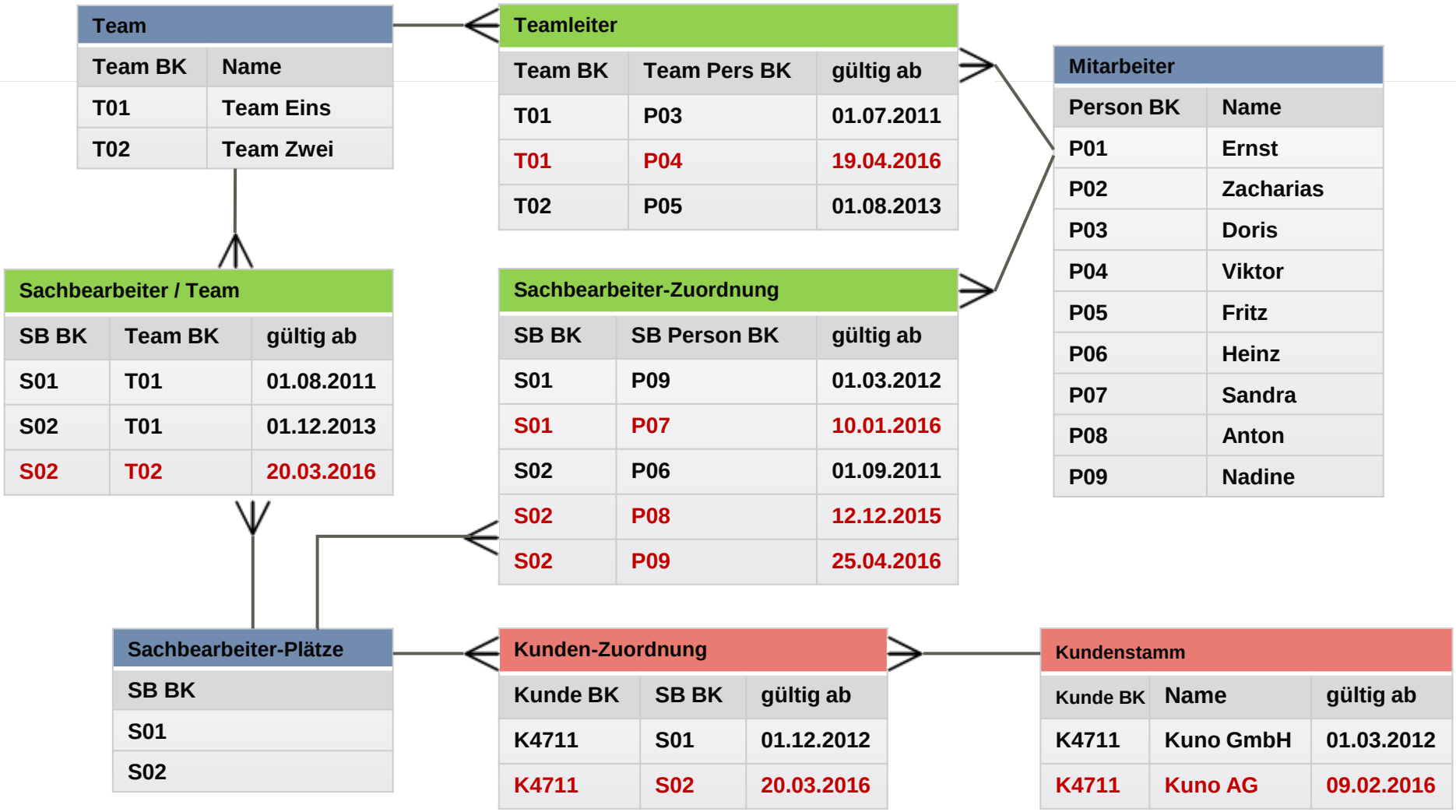
Meik Truschkowski
nobilia-Werke J. Stickling GmbH & Co. KG | Verl
Projektleiter Business Intelligence und Data Warehousing

Problem: Historisierung (Versionierung)



nobilia

Änderungen in den Quelldaten



Team

Team BK	Name
T01	Team Eins
T02	Team Zwei

Teamleiter

Team BK	Team Pers BK	gültig ab
T01	P03	01.07.2011
T01	P04	19.04.2016
T02	P05	01.08.2013

Mitarbeiter

Person BK	Name
P01	Ernst
P02	Zacharias
P03	Doris
P04	Viktor
P05	Fritz
P06	Heinz
P07	Sandra
P08	Anton
P09	Nadine

Sachbearbeiter / Team

SB BK	Team BK	gültig ab
S01	T01	01.08.2011
S02	T01	01.12.2013
S02	T02	20.03.2016

Sachbearbeiter-Zuordnung

SB BK	SB Person BK	gültig ab
S01	P09	01.03.2012
S01	P07	10.01.2016
S02	P06	01.09.2011
S02	P08	12.12.2015
S02	P09	25.04.2016

Sachbearbeiter-Plätze

SB BK
S01
S02

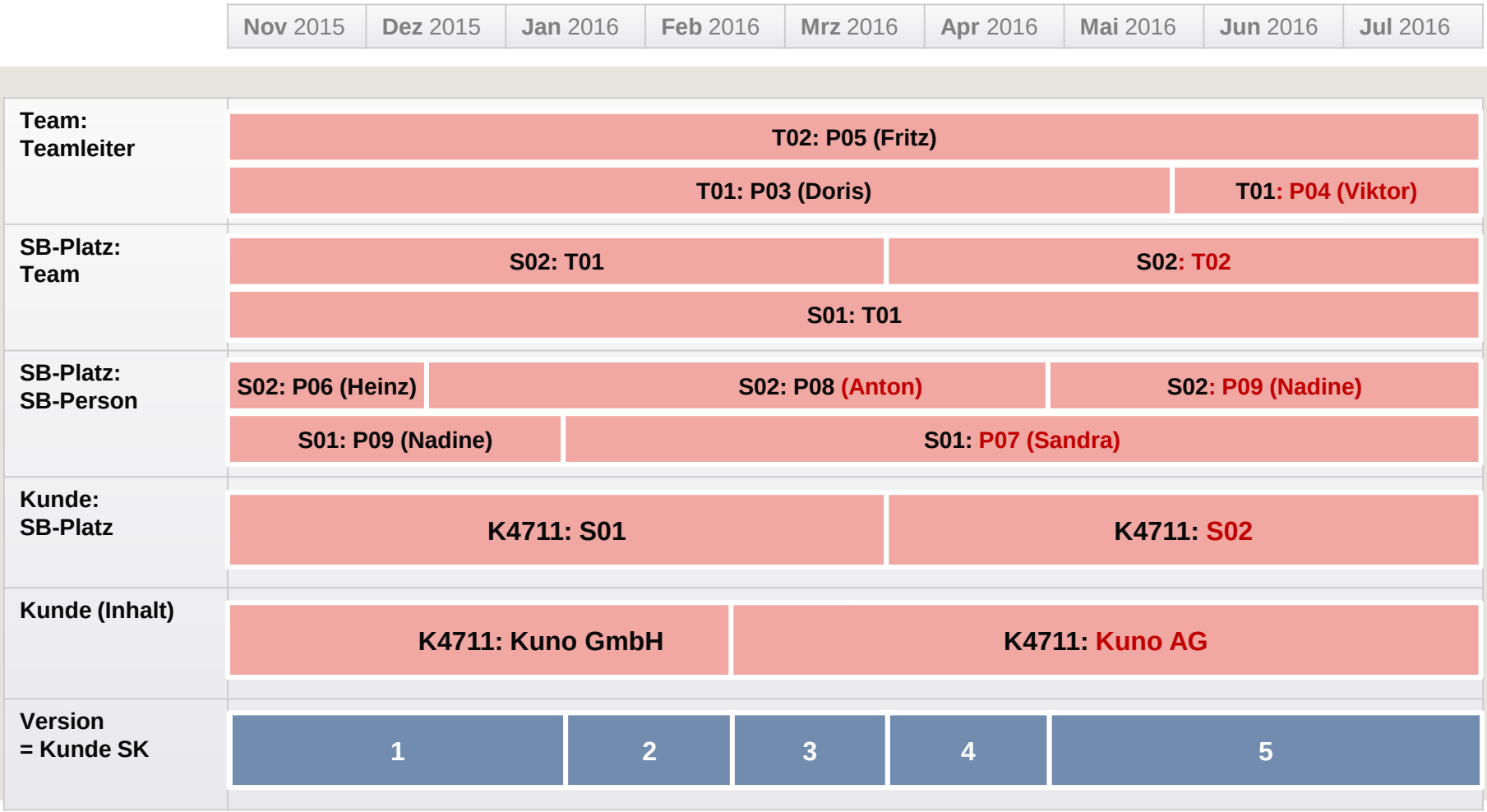
Kunden-Zuordnung

Kunde BK	SB BK	gültig ab
K4711	S01	01.12.2012
K4711	S02	20.03.2016

Kundenstamm

Kunde BK	Name	gültig ab
K4711	Kuno GmbH	01.03.2012
K4711	Kuno AG	09.02.2016

Zeitleiste



nobilia

Ergebnismenge

kleinstmögliche Zeitscheiben

KUNDE SK	KUNDE BK	KUNDE_NAME	SB BK	SB PERSON BK	TEAM BK	TEAM PERSON BK	GÜLTIG AB	GÜLTIG_BIS
1	K4711	Kuno GmbH	S01	P09	T01	T03	01.01.2000	09.02.2016
2	K4711	Kuno GmbH	S01	P07	T01	T03	10.01.2016	08.02.2016
3	K4711	Kuno AG	S01	P07	T01	T03	09.02.2016	19.03.2016
4	K4711	Kuno AG	S02	P08	T02	P05	20.03.2016	24.04.2016
5	K4711	Kuno AG	S02	P09	T02	P05	25.04.2016	31.12.9999

KIMBALL!

…nun zur Lösung in SQL…

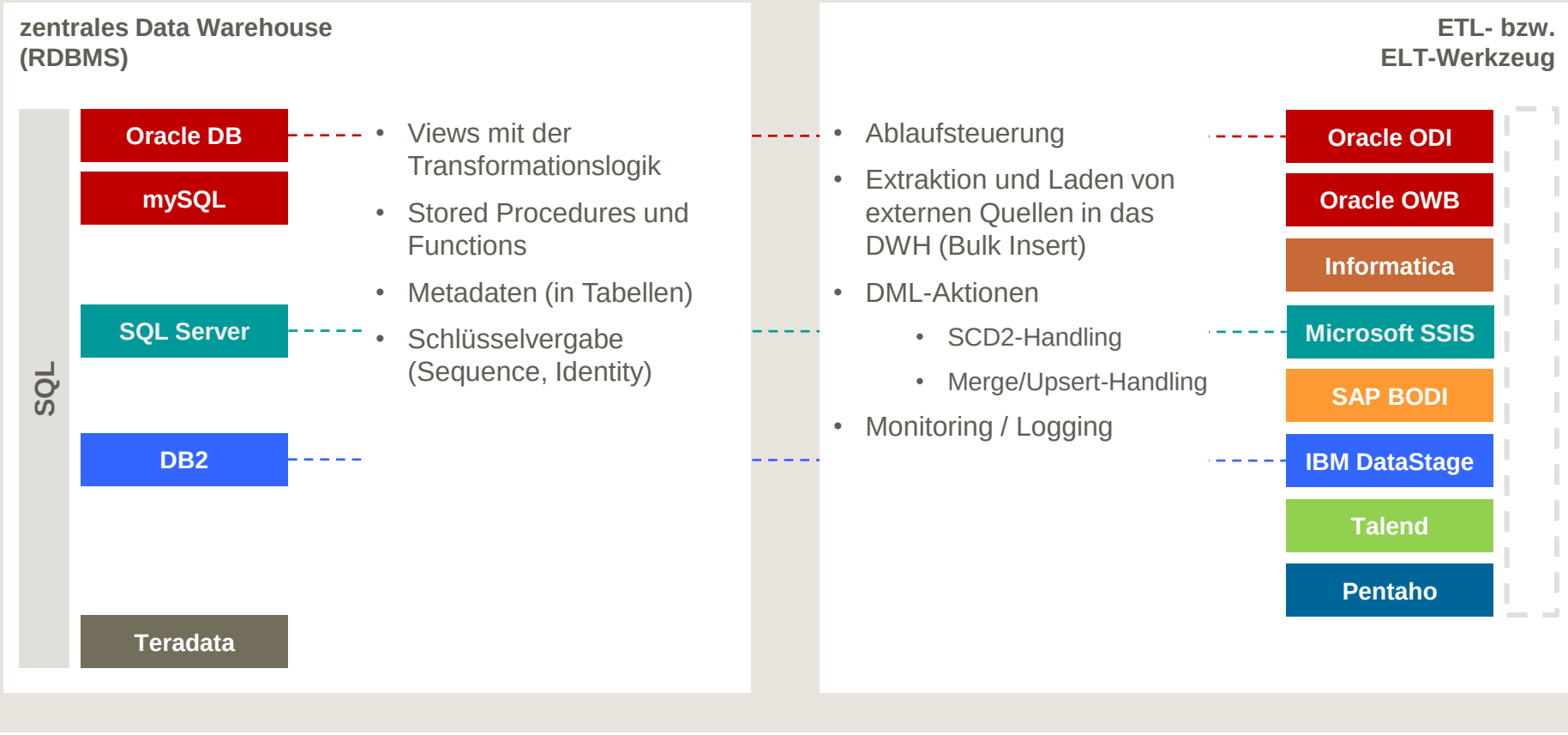
Moment mal – in SQL???

Als Skript?

Es gibt doch moderne ETL-Tools!?

Hybrides ~~ETL~~ ELT

Was heißt hier eigentlich Plattform-unabhängig?



nobilia

Argumente für und gegen hybrides ELT

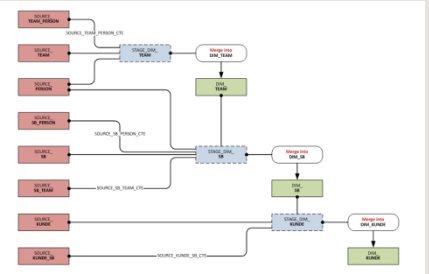
Warum das so toll ist… oder auch nicht…

+

- SQL-Code allgemein verständlich
- Rapid Prototyping
- ANSI-Standard
- Einflussnahme auf die Performance
- Stored Procedures

-

- SQL-Kenntnisse notwendig
- kein zeilenbasiertes ETL
- „Magic Numbers“
- Dokumentation (Data Lineage) manuell



nobilia

Zurück zur Lösung

Voraussetzungen

Gültigkeitsdatum:

- beide Datumsspalten für Zuordnung zu Fakten notwendig
- einheitliches Start-Datum der ersten Versionen (01.01.2000)
- einheitliches End-Datum der gültigen Version (31.12.9999)
- keine Lücken
- keine Überlappungen

In der Demo werden die Gültigkeitsregeln im SQL-Skript sichergestellt!

SQL-Dialekt:

- Common Table Expressions (CTE)
- Greatest() und Least()-Funktionen
- Case When Then Else bzw. Iif oder ähnliches Konstrukt

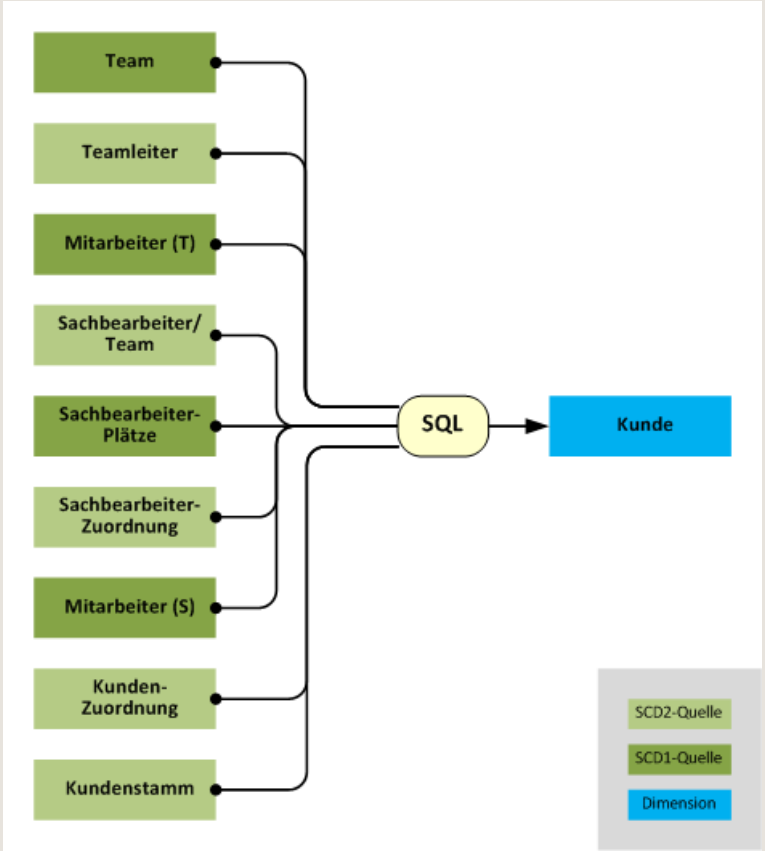
weiter mit der Lösung

Variante 1

Eine einzige Ergebnis-Tabelle

- Ein Transformationsschritt (ein SQL-Statement)
- Referenzierung nur auf Version des Kunden möglich
- keine Referenzierung auf Version von SB-Platz oder Team möglich

Gute Lösung, wenn keine Hierarchien zwischen den Dimensionen bestehen!

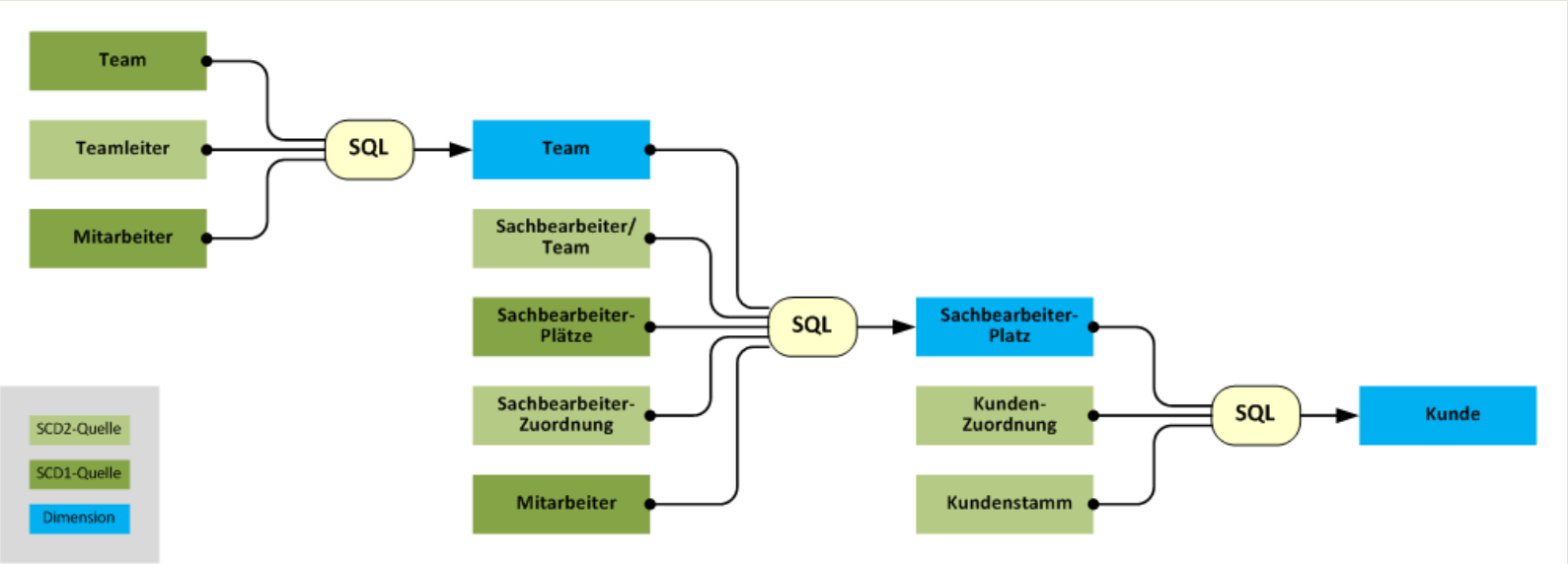


nobilia

weiter mit der Lösung

Variante 2

- Drei Transformationsschritte (drei SQL-Statements)
- jeweils einzelne versionierte Tabellen für Team und SB-Platz
- Referenzierung in der Kunden-Tabelle auf deren SKs
- Star-Schema-Tabelle als View

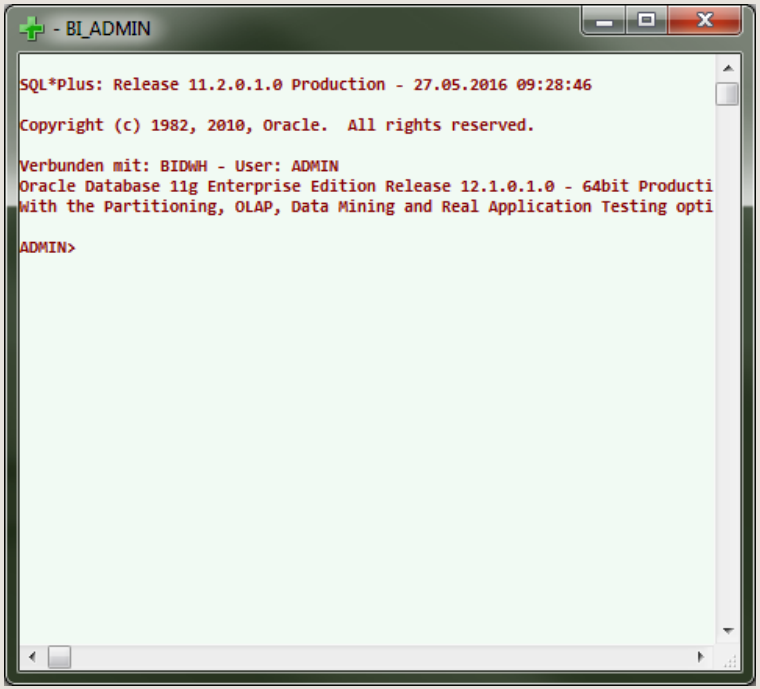


nobilia

Demonstration der Lösung

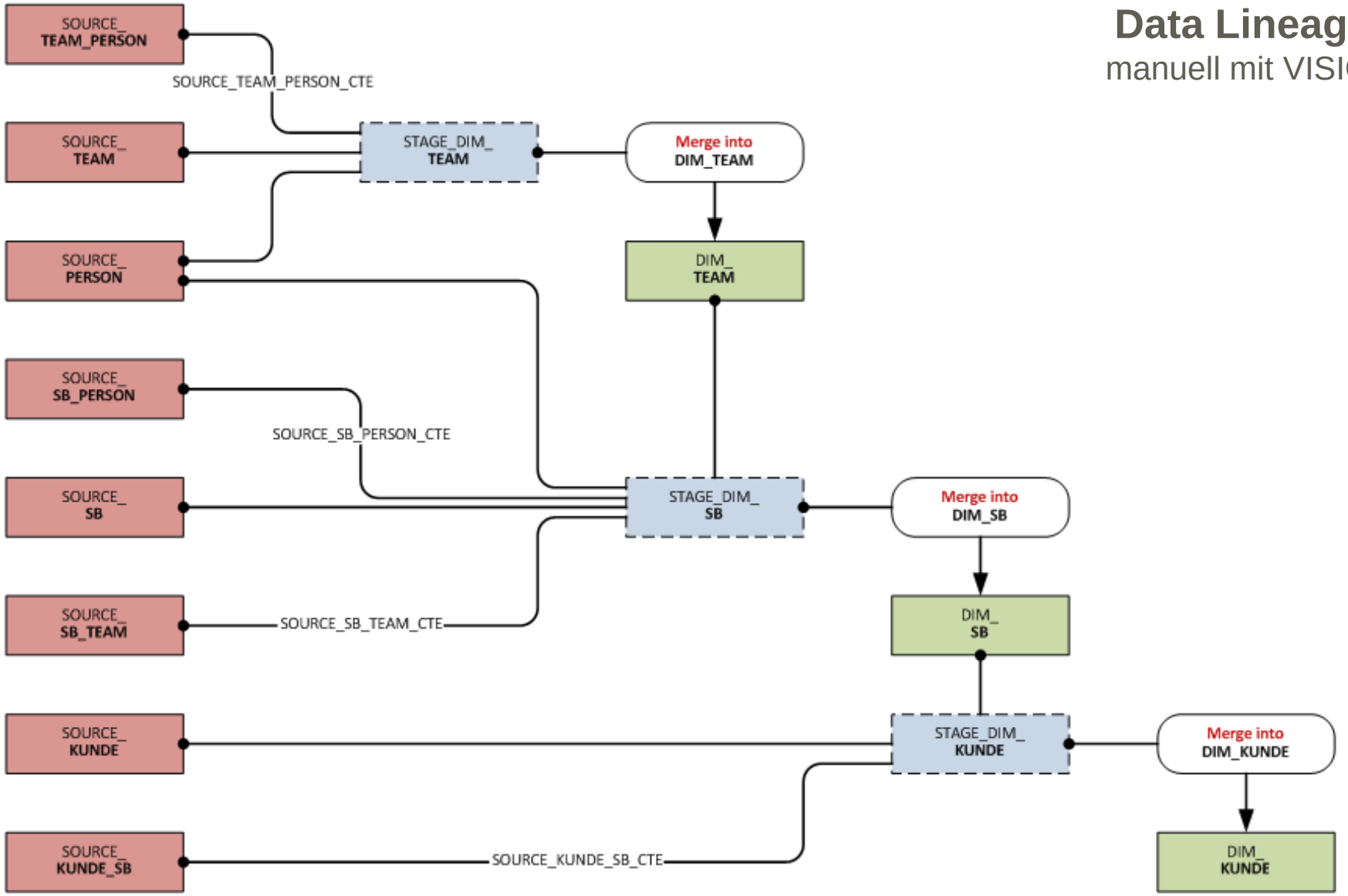
in Variante 2

- reines SQL (Oracle)
- kein ETL/ELT-Tool im Beispiel
- alle Tabellen/Views im Schema „TEST“
- Demo im SQL*Plus



Die gezeigte Lösung funktioniert übrigens auch mit Parent-Child-Hierarchien. Dabei muss allerdings rekursives SQL (mit verschachtelten CTEs) angewendet werden!

Data Lineage
manuell mit VISIO



nobilia

Probleme und Lösungen?

Wie bekommt man seine Versionen sauber?

Wenn…?	…dann!
Komponente ohne Versionen	eine Version mit dem ersten Gültig-ab- und finalen Gültig-bis-Datum erzeugen
Lücken oder Überlappungen	eine der Datumsspalten als „führend“ bestimmen, danach die fehlende Spalte in SQL errechnen
nur eine der beiden Datumsspalten vorhanden	fehlende Spalte in SQL errechnen (Sub-Select)
erste Parent-Version ist erst später gültig als erste Child-Version	einheitliches erstes Gültig-ab- und finales Gültig-bis-Datum verwenden
„Late arriving Facts“	bei der jeweils letzten Version immer das finale Gültig-bis-Datum verwenden
Gültigkeiten ändern sich nachträglich und rückwirkend (vor allem bei Zuordnungen) 	a) bitemporale Datenhaltung in Dimensionen und Fakten (noch nicht ausprobiert) b) Dimension neu aufbauen, Fakten neu schlüsseln (Informationsverlust)

…noch Antworten?