WHITE-PAPER

Datenzugriffe für Berichtsanwender

Steuerung der Berechtigungen

Autor: Felix Krul, Senior BI Architect
Stand: Juni 2013

Inhaltsverzeichnis

1 Berichtswerkzeuge und Benutzerberechtigungen

1.1 Berichtswerkzeuge

In jedem Data Warehouse-System gibt es eine Präsentationsschicht, über die die Endanwender, also die Berichtsempfänger (und -ersteller), auf die Daten zugreifen.

Im Gegensatz zum DWH-Entwicklerteam, das meistens vergleichsweise klein ist, kann der Kreis der Endanwender mehrere hundert oder sogar tausend Personen umfassen, die größtenteils aus den Fachbereichen stammen und meistens keine Entwickler sind. Ein Zugriff auf die Daten mittels selbstgeschriebener SQLs verbietet sich daher, nicht zuletzt auch deswegen, weil die Anforderungen an die Darstellung und Weiterverarbeitung der Daten weit über eine unformatierte Ergebnistabelle hinausgehen.

Aus diesem Grund werden praktisch immer Endanwenderwerkzeuge im Berichtswesen eingesetzt.

1.2 Berechtigungen in den Berichtswerkzeugen

Auch aus Sicht der Administration, Qualitätssicherung und Zugriffsteuerung bietet der Einsatz von Berichtswerkzeugen Vorteile:

- Die Abfragen auf ein DWH können teilweise recht komplex sein, daher ist eine Aufbereitung der Datenstrukturen mit Hilfe einer semantischen Schicht eine wichtige Maßnahme zur Sicherstellung korrekter Abfragen.
- Die meisten Endanwender sind reine Empfänger von Standardberichten. Diese können entsprechend erstellt, qualitätsgesichert und automatisiert verteilt werden.
- Die meisten Anwender erhalten nur Zugriffe auf definierte Data Marts, nur sehr wenige Anwender erhalten globalen Zugriff auf sämtliche Datenbereiche. Auf Datenbankebene müsste dies über getrennte Datenbankzugriffe realisiert werden.
- Selbst wenn ein Zugriff auf einen Data Mart erlaubt ist, bedeutet dies noch nicht, dass alle zugehörigen Sichten und Berichte zugängig sein müssen.
- Schließlich ist meistens[1] auch der Zugriff auf die Dateninhalte eingeschränkt: Häufig dürfen Endanwender nur die Daten sehen, die sie selbst erstellt haben oder sie sehen nur die Daten aus einem bestimmten Teilbereich der Organisationsstruktur.

Generell lassen sich drei Hauptgruppen von Berechtigungen unterscheiden:

1.: Funktionale Rechte:

Diese Rechte steuern, welche Werkzeuge und darin welche Funktionalitäten ein Endanwender freigeschaltet bekommt. Unterschieden werden beispielsweise Administratoren, Designer der semantischen Schicht, Berichtsersteller und Berichtsempfänger.

2.: Objektrechte:

Diese Rechte steuern den Zugriff auf die erlaubten Objekte, also die semantischen Schichten und die Berichte. In den meisten Fällen sind diese Zugriffe projektspezifisch und werden über entsprechende Rollen vergeben.

3.: Datenrechte

Wie oben erwähnt ist meistens auch der Zugriff auf die Daten selbst eingeschränkt. Da dies einer Einschränkung auf der Ebene der Zeilen der Datenbanktabellen entspricht, wird dies auch häufig als *Row Level Security* bezeichnet.

Die Row Level Security ist der Inhalt dieses Dokuments: Ziel ist es, die Steuerung dieser Rechte mit Hilfe von neuen Tabellen (den Securitytabellen) vorzuführen und die Flexibilität des Verfahrens zu zeigen.

[1] Interessanterweise unterscheiden sich hier deutschsprachige und angelsächsischen Länder: In ersteren ist dies die Regel, in letzteren wird diese Art der Einschränkungen häufig nicht für notwendig erachtet!

1.3 Berichtswesen ohne Werkzeug?

Im vorigen Abschnitt 1.2 wurden die Einschränkungen auf Ebene des Berichtswerkzeugs beschrieben. Die Frage ist, ob ein Berichtswesen ohne Werkzeug, d.h. über einen direkten SQL-Zugriff auf die Datenbank überhaupt sinnvoll ist und wenn ja, ob er auch ähnlich abgesichert werden kann.

Grundsätzlich lassen sich auch in Datenbanken Berechtigungen vergeben: Ein Datenbankuser kann nur auf bestimmte Tabellen (oder sogar eingeschränkte Views) berechtigt werden, auch die Einschränkung auf reine Leserechte ist möglich.

Andererseits kann die Erlaubnis, SQLs auszuführen nicht weggenommen werden, da ansonsten kein Berichtswesen möglich wäre.

Auch die Einrichtung von einzelnen Datenbank-Usern mit verschiedenen Rechten für mehrere tausend Endanwender lässt sich kaum bewerkstelligen, zumal die meisten Endanwender reine Berichtsempfänger sind.

Ein Direktzugriff auf die Datenbank ist somit sinnvoll nur für die Mitglieder des Entwicklungsteams zu vergeben (diese haben ohnehin alle Datenzugriffe), alle Endanwender sollten im Berichtswerkzeug arbeiten.

1.4 Im Dokument verwendetes Werkzeug

Alle Beispiele (Semantische Schichten, Berichte) in diesem Dokument wurden mit der Software SAP BusinessObjects BI 4.0 erstellt.

Es sei aber darauf hingewiesen, dass die vorgestellten Funktionalitäten heute aber in allen gängigen Berichtswerkzeugen Standard sind.

2 Die Beispieldatenbank

Zu diesem Dokument gehört eine Beispieldatenbank, an der sämtliche Beispiele erläutert werden. Diese Datenbank liegt im Excel -Format vor (Datei Beispieldatenbank.xls) und kann in jede gängige relationale Datenbank übernommen werden.

Die Daten sind frei erfunden und lediglich für die Demonstration von Beispielauswertungen in diesem Dokument gedacht.

Im Folgenden wird das grundlegende Datenmodell vorgestellt. Weitere Tabellen oder Strukturerweiterungen werden später bei Bedarf hinzugefügt, z.B. die verschiedenen Securitytabellen oder wenn die Historisierung besprochen wird (siehe Abschnitt 10).

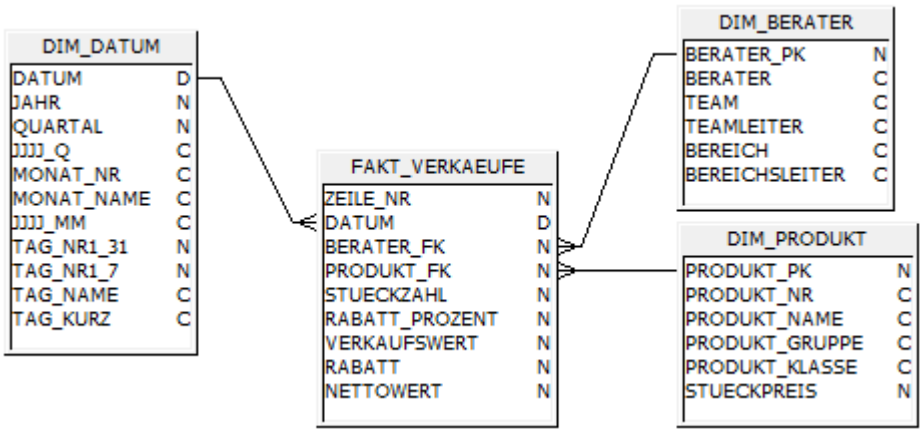
2.1 Grundlegendes Datenmodell des Data Marts

Wir gehen zunächst von einem stark vereinfachten Data Mart aus, der aus drei nicht-historisierten Dimensionen DIM_DATUM, DIM_BERATER und DIM_PRODUKT, sowie einer Faktentabelle FAKT_VERKAEUFE besteht. Die Daten simulieren Verkaufsvorgänge, bei denen die Berater an bestimmten Tage jeweils ein Produkt in einer bestimmten Stückzahl verkauft haben und dabei einen Rabatt gewähren konnten. Die weiteren Kennzahlen sind Verkaufswert = Stückzahl *Stückpreis, Rabatt = Verkaufswert*Rabatt Prozent und Nettowert = Verkaufswert – Rabatt.

In den Dimensionen DIM_BERATER und DIM_PRODUKT wurde jeweils ein künstlicher Primärschlüssel (BERATER_PK /PRODUKT_PK) eingefügt (in DIM_DATUM ist dies das Datum selbst), in der Faktentabelle werden diese als Fremdschlüssel entsprechend referenziert (Felder DATUM/BERATER_FK/PRODUKT_FK.

Da keine Referenzen fehlen, können ausschließlich Inner Joins verwendet werden.

Daraus ergibt sich zunächst das folgende Datenmodell:



2.2 Die Tabellen im Detail

2.2.1 DIM_DATUM

In der Datumsdimension sind alle Datumswerte vom 1.1.2010 bis 31.12.2015 mit typischen Hierarchieelementen wie Jahr, Quartal, Monat,... enthalten. Das Feld TAG_NR1_31 enthält die Tagesnummer im Monat, das Feld TAG_NR1_7 die Wochentagnummer, also 1 = Montag... 7 = Sonntag.

Die folgende Tabelle zeigt einige exemplarische Einträge:

DATUM	JAHR	QUARTAL	JJJJ_Q	MONAT_NR	MONAT_NAME	JJJJ_MM	TAG_NR1_31	TAG_NR1_7	TAG_NAME	TAG_KURZ
28.12.2012	2012	4	2012_4	12	Dezember	2012_12	28	5	Freitag	Fr

29.12.2012	2012	4	2012_4	12	Dezember	2012_12	29	6	Samstag	Sa
30.12.2012	2012	4	2012_4	12	Dezember	2012_12	30	7	Sonntag	So
31.12.2012	2012	4	2012_4	12	Dezember	2012_12	31	1	Montag	Mo
01.01.2013	2013	1	2013_1	01	Januar	2013_01	1	2	Dienstag	Di
02.01.2013	2013	1	2013_1	01	Januar	2013_01	2	3	Mittwoch	Mi
03.01.2013	2013	1	2013_1	01	Januar	2013_01	3	4	Donnerstag	Do

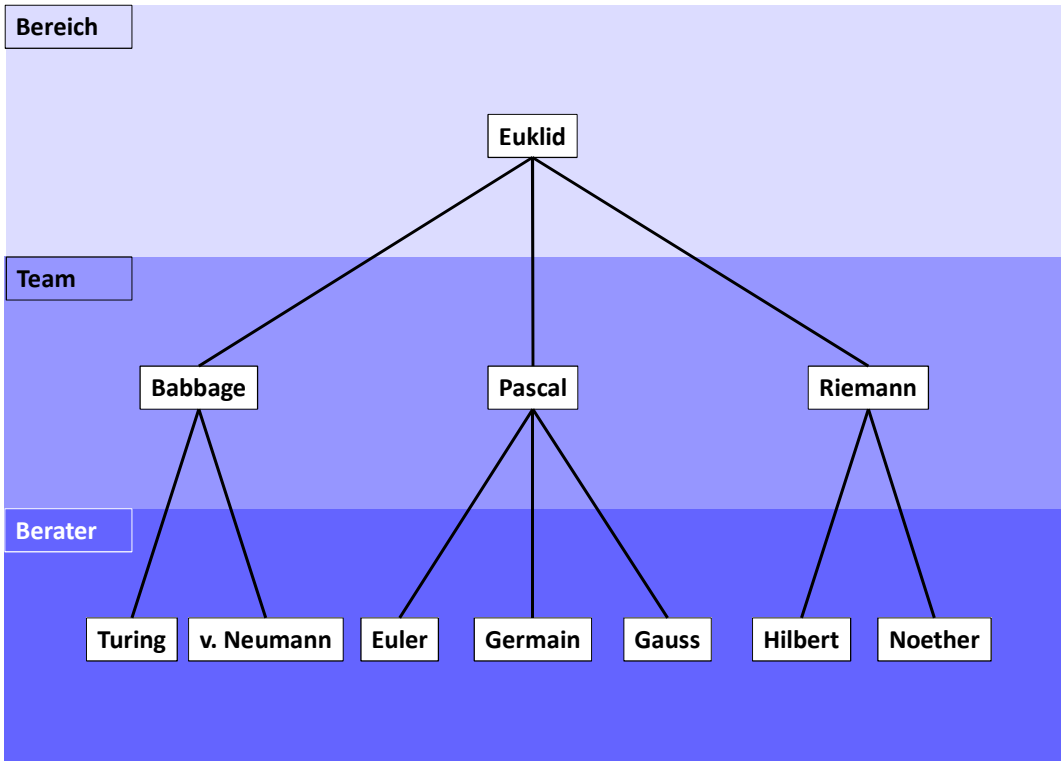
2.2.2 *DIM_BERATER*

In dieser Dimension sind die Berater mit ihrer Organisationshierarchie (Bereich -> Team -> Berater) eingetragen. Als Beispiele wurden berühmte Mathematiker und Physiker gewählt:

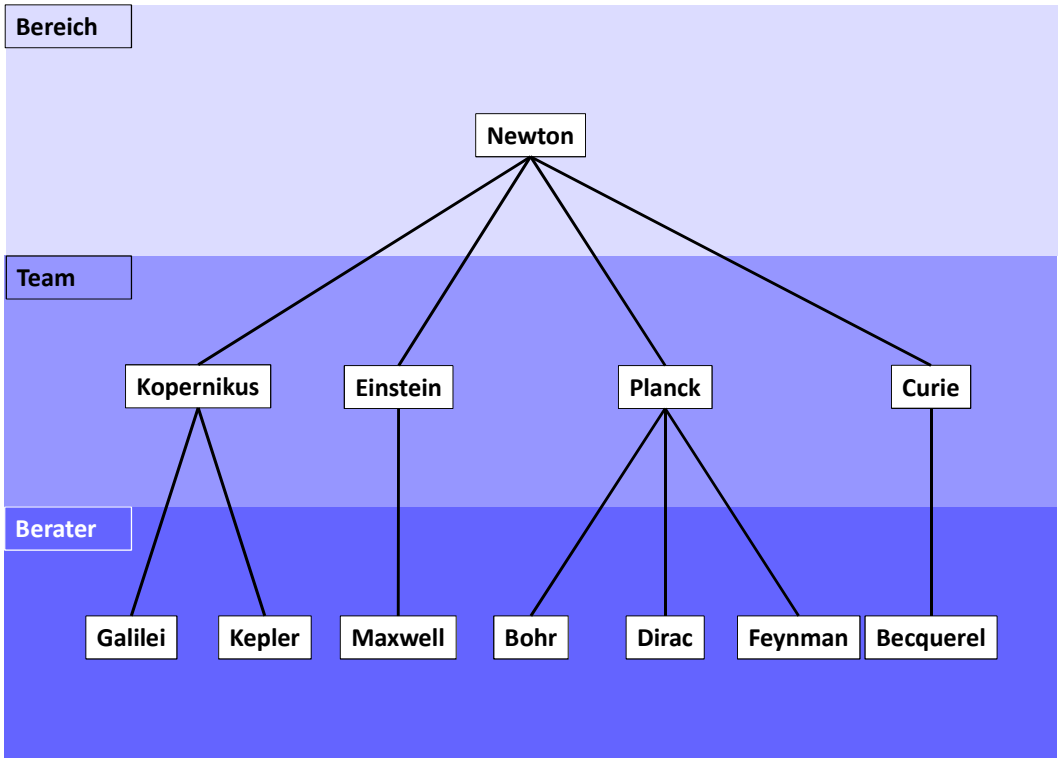
BERATER_PK	BERATER	TEAM	TEAMLEITER	BEREICH	BEREICHSLEITER
1	Newton	Leitung Physiker	Newton	Physiker	Newton
2	Kopernikus	Klassische Physiker	Kopernikus	Physiker	Newton
3	Kepler	Klassische Physiker	Kopernikus	Physiker	Newton
4	Galilei	Klassische Physiker	Kopernikus	Physiker	Newton
5	Einstein	Elektrodynamiker	Einstein	Physiker	Newton
6	Maxwell	Elektrodynamiker	Einstein	Physiker	Newton
7	Planck	Quantenphysiker	Planck	Physiker	Newton
8	Bohr	Quantenphysiker	Planck	Physiker	Newton
9	Dirac	Quantenphysiker	Planck	Physiker	Newton
10	Feynman	Quantenphysiker	Planck	Physiker	Newton
11	Curie	Radioaktivitätsforscher	Curie	Physiker	Newton
12	Bequerel	Radioaktivitätsforscher	Curie	Physiker	Newton
13	Euklid	Leitung Mathematiker	Euklid	Mathematiker	Euklid
14	Pascal	Mathematiker Frühe Neuzeit	Pascal	Mathematiker	Euklid
15	Euler	Mathematiker Frühe Neuzeit	Pascal	Mathematiker	Euklid
16	Germain	Mathematiker Frühe Neuzeit	Pascal	Mathematiker	Euklid
17	Gauss	Mathematiker Frühe Neuzeit	Pascal	Mathematiker	Euklid
18	Riemann	Mathematiker Neuzeit	Riemann	Mathematiker	Euklid
19	Noether	Mathematiker Neuzeit	Riemann	Mathematiker	Euklid
20	Hilbert	Mathematiker Neuzeit	Riemann	Mathematiker	Euklid
21	Babbage	Informatiker	Babbage	Mathematiker	Euklid
22	v. Neumann	Informatiker	Babbage	Mathematiker	Euklid
23	Turing	Informatiker	Babbage	Mathematiker	Euklid

Die folgenden zwei Grafiken zeigen die vollständige Beraterhierarchie:

Bereich Euklid:



Bereich Newton:



2.2.3 DIM_PRODUKT

In dieser Dimension stehen die Produkte mit einer eindeutigen Produkt_Nr, ihrem Namen, einer zweistufigen Hierarchie (Produktgruppe und -klasse), sowie dem Stückpreis.

In dem Geschäft werden Möbel, Elektrogeräte und einige Lebensmittel verkauft, hier ist die vollständige Tabelle:

PRODUKT_PK	PRODUKT_NR	PRODUKT_NAME	PRODUKT_GRUPPE	PRODUKT_KLASSE	STUECKPREIS
1	M_W_01	Stuhl	Wohnzimmer	Möbel	124
2	M_W_02	Regal	Wohnzimmer	Möbel	160
3	M_W_03	Sofa	Wohnzimmer	Möbel	825
4	M_S_01	Bett	Schlafzimmer	Möbel	1500
5	M_S_02	Schrank	Schlafzimmer	Möbel	1800
6	M_S_03	Kommode	Schlafzimmer	Möbel	910
7	M_K_01	Küchenschrank	Küche	Möbel	800
8	M_K_02	Küchentisch	Küche	Möbel	179
9	E_G_01	Herd	Großgeräte	Elektrogeräte	355
10	E_G_02	Kühlschrank	Großgeräte	Elektrogeräte	260
11	E_G_03	Geschirrspüler	Großgeräte	Elektrogeräte	280
12	E_K_01	Staubsauger	Kleingeräte	Elektrogeräte	180
13	E_S_01	Fernseher	Stereo_HiFi	Elektrogeräte	420
14	E_S_02	Radio	Stereo_HiFi	Elektrogeräte	110
15	E_S_03	DVD-Spieler	Stereo_HiFi	Elektrogeräte	180
16	L_S_01	Schokolade	Süßwaren	Lebensmittel	0,99
17	L_S_02	Bonbons	Süßwaren	Lebensmittel	1,99
18	L_S_03	Kekse	Süßwaren	Lebensmittel	2,49
19	S_O_01	1 kg Bananen	Obst	Lebensmittel	1,99
20	S_O_02	1 kg Äpfel	Obst	Lebensmittel	2,29

2.2.4 FAKT_VERKAEUFE

In dieser Tabelle werden die Verkaufsvorgänge gespeichert: Ein Berater verkauft an einem bestimmten Datum ein Produkt (siehe oben). Zur Zuordnung wurden die künstlichen Schlüssel verwendet. Hier sind einige Faktensätze (insgesamt über 2000 Sätze):

ZEILE_NR	DATUM	BERATER_FK	PRODUKT_FK	STUECK-ZAHL	RABATT_PROZENT	VERKAUFS-WERT	RABATT	NETTOWERT
1	01.01.2010	20	5	6	11	10800,00	1188,00	9612,00
8	06.01.2010	23	15	21	8	3780,00	302,40	3477,60
27	24.01.2010	22	1	29	3	3596,00	107,88	3488,12
64	07.03.2010	21	1	8	0	992,00	0,00	992,00
125	07.05.2010	15	6	23	14	20930,00	2930,20	17999,80
216	10.08.2010	16	12	20	4	3600,00	144,00	3456,00
343	12.12.2010	13	4	29	10	43500,00	4350,00	39150,00
512	04.06.2011	3	15	9	25	1620,00	405,00	1215,00
729	09.01.2012	4	8	40	12	7160,00	859,20	6300,80

1000	20.10.2012	19	15	27	25	4860,00	1215,00	3645,00
1331	27.09.2013	20	18	12	0	29,88	0,00	29,88
1728	14.11.2014	4	15	9	25	1620,00	405,00	1215,00

2.3 Verwendete SQL-Syntax

Im Dokument werden viele Beispiele mit den zugehörigen SQL-Abfragen erläutert.

Um die Lesbarkeit zu erhöhen, wurde die Syntax

SELECT <Abfrageobjekte>

FROM <Tabellenliste>

WHERE <Joinbedingungen> AND <sonstige Bedingungen>

anstelle der oft üblichen ANSII 92-Syntax (Joinbedingungen im FROM) gewählt. Da im Dokument keine Outer Joins vorkommen, sind beide Formen äquivalent, daher ist auch eine Übersetzung jederzeit möglich.

3 Einführung: Dateneinschränkungen

3.1 Grundidee

Wenn die Endanwender einen Standardbericht aufrufen und aktualisieren oder auch eine Ad-Hoc-Abfrage ausführen, so haben sie meistens nicht Zugriff auf alle Daten, sondern nur auf ihre eigenen Sätze.

Im obigen Beispiel hätten alle Berater Zugriff auf die Sätze, die zu ihren eigenen Verkäufen gehören, nicht aber auf alle anderen.

Teamleiter hätten dann Zugriff auf alle Sätze, die innerhalb ihres eigenen Teams erstellt werden, Bereichsleiter auf alle Sätze innerhalb des eigenen Bereichs und zentrale Controller oder Vorstände auf alle Datensätze.

Dabei sind meistens die Faktensätze kritisch: Meistens (nicht immer!) ist der Zugriff auf die Stammdaten noch uneingeschränkt erlaubt (jeder darf alle Produkte in ihrer Hierarchie sehen), aber sobald der Zugriff auf Kennzahlen erfolgt, muss die Einschränkung erfolgen.

Die Einschränkung erfolgt nicht auf die Faktentabelle direkt:

- Es wäre nicht sinnvoll, eine Bedingung auf die Satznummer (etwa: Berater Galilei sieht die Sätze 3, 19, 25, 41, 47, ...) ist zum einen sehr umständlich, zum kontraproduktiv: Es werden ja ständig neue Sätze erzeugt. Genau genommen soll die Einschränkung auf die sichtbaren Sätze sogar ein *Resultat* der Dateneinschränkung sein.
- Eine Bedingung auf den (künstlichen!) Fremdschlüssel wäre ebenfalls nicht praktikabel: diese sind nicht sprechend und daher in der Administration nur schwer zu gebrauchen. Außerdem werden wir später auch auf historisierte Dimensionen eingehen, und hier würde jeder neue historisierte Eintrag in der Dimension eine Anpassung der Einschränkung erfordern, was nicht praktikabel ist.
- Grundsätzlich könnte auch das sprechende Einschränkungsattribut (etwa der Name des Beraters) als zusätzliches Feld in die Faktentabelle aufgenommen werden, dies würde aber einerseits die Faktentabelle unnötig vergrößern, andererseits werden wir später sehen, dass die (unten beschriebene) Einschränkung auf die Dimensionstabelle sehr viel flexibler in der Handhabung und auch für Erweiterungen ist.

Wir gehen daher im Folgenden davon aus, dass eine einschränkende Bedingung auf die Dimensionstabelle gesetzt wird, meistens die interne Organisationshierarchie[2].

3.2 Verwendung des Berichtswerkzeugs für die Einschränkungen

Die meisten Berichtswerkzeuge bieten eine Standardfunktionalität zur Einschränkung von Benutzern an, in etwa wie folgt:

- Benutzer „Galilei“ meldet sich an
- wenn Benutzer „Galilei“ in der Abfrage auf die semantische Schicht „Datamart Vertrieb“ das gesicherte Objekt „Umsatz“ verwendet[3], so wird automatisch die Bedingung
  `DIM_BERATER.BERATER = 'Galilei'`
  ins SQL eingefügt.
- Diese Bedingung kann nicht vom Anwender editiert werden, so dass die entsprechende Einschränkung nicht umgangen werden kann.
- Zusätzliche Einschränkungen dürfen selbstverständlich weiterhin vorgenommen werden, die Verknüpfung der Bedingungen erfolgt aber immer über ein logisches AND, so dass auch hier keine Abschwächung der Sicherheitsbedingung erfolgt.

---

[2] Es kann auch andere Fälle geben, wie z.B. ein Produktmanager, der organisationsübergreifend alle Verkäufe für ein bestimmtes Produkt oder eine bestimmte Produktgruppe sehen darf; Abschnitt 8 geht darauf ein.

[3] eine Alternative könnte auch sein: Wenn die gesicherte Tabelle „FAKT_VERKAEUFE“ angesprochen wird.

Es versteht sich, dass für alle Anwender das Recht, das SQL manuell zu bearbeiten, deaktiviert sein muss, ebenso darf kein Anwender einen direkten Zugriff auf die Datenbank (ohne semantische Schicht) erhalten, da ansonsten das SQL entsprechend editiert werden könnte.

Auch für Benutzer auf Team- oder Bereichsebene lässt sich die Bedingung eingeben: Beispielsweise darf der Teamleiter Planck neben seinen eigenen Daten die seiner Mitarbeiter Bohr, Dirac und Feynman sehen, für alle ist der Teamleiter „Planck“, so dass die Bedingung wie folgt lautet:

```
DIM_BERATER.TEAMLEITER = 'Planck'
```

Für den Bereichsleiter Newton wäre die folgende Bedingung einzugeben:

```
DIM_BERATER.BEREICHSLEITER = 'Newton'
```

Diese grundlegende Funktionalität ist durchaus ausreichend, solange die Anforderungen an die Einschränkungen überschaubar sind, also so lange nur wenige solche Fälle zu pflegen sind und keine größeren Komplexitäten in der Sichtbarkeit auftreten.

Der Vorteil ist auch, dass dieselbe Regel meistens auch auf mehrere Anwender angewandt werden kann, beispielsweise auf alle Bereichscontroller, alle Vorstände o.ä.

Schon für mehr als 20 verschiedene Regeln kann die Pflege aber sehr aufwändig, unübersichtlich und fehleranfällig sein. Insbesondere bei häufigen Änderungen ist die Nachvollziehbarkeit stark eingeschränkt.

In diesen Fällen sollte daher nach anderen Lösungen gesucht werden. Der Inhalt dieses Dokuments ist es, den Einsatz von sogenannten Securitytabellen zur Pflege solcher Einschränkungen vorzustellen und zu zeigen, wie das Konzept auch auf komplexe Anforderungen erweitert werden kann.

4 Zeileneinschränkungen mit Securitytabellen: Ein einführendes Beispiel

4.1 Definition der Securitytabelle

In einem ersten Schritt definieren wir eine neue Tabelle, die jedem Anwender alle Berater, deren Daten er sehen darf, zuordnet. Diese Tabelle hat also genau zwei Spalten: Den Anwender und die erlaubten Berater. In der Beispieldatenbank ist sie unter dem Namen SECURITY_EINFACH zu finden.

Jeder Berater erhält daher sich selbst zugeordnet, im folgenden Beispiel die beiden Berater Kepler und Galilei:

ANWENDER	BERATER
Kepler	Kepler
Galilei	Galilei

Ein Teamleiter sieht sich selbst und alle Berater im Team, hier als Beispiel der Teamleiter Kopernikus, der auch die Berater Kepler und Galilei sieht:

ANWENDER	BERATER
Kopernikus	Kopernikus
Kopernikus	Kepler
Kopernikus	Galilei

Ein Bereichsleiter sieht sich selbst, alle Teamleiter im Bereich und alle Berater in den Teams, hier für den Bereichsleiter Newton:

ANWENDER	BERATER
Newton	Newton
Newton	Kopernikus
Newton	Kepler
Newton	Galilei
Newton	Einstein
Newton	Maxwell
Newton	Planck
Newton	Bohr
Newton	Dirac
Newton	Feynman
Newton	Curie
Newton	Bequerel

Schließlich können auch Anwender definiert werden, die alle Daten sehen dürfen, wie zum Beispiel der Anwender „Vorstand“. Diesem werden alle Berater zugewiesen:

ANWENDER	BERATER
Vorstand	Newton
Vorstand	Kopernikus
Vorstand	Kepler
Vorstand	Galilei
Vorstand	Einstein
Vorstand	Maxwell

Vorstand	Planck
Vorstand	Bohr
Vorstand	Dirac
Vorstand	Feynman
Vorstand	Curie
Vorstand	Bequerel
Vorstand	Euklid
Vorstand	Pascal
Vorstand	Euler
Vorstand	Germain
Vorstand	Gauss
Vorstand	Riemann
Vorstand	Noether
Vorstand	Hilbert
Vorstand	Babbage
Vorstand	v. Neumann
Vorstand	Turing

!!! Achtung !!!

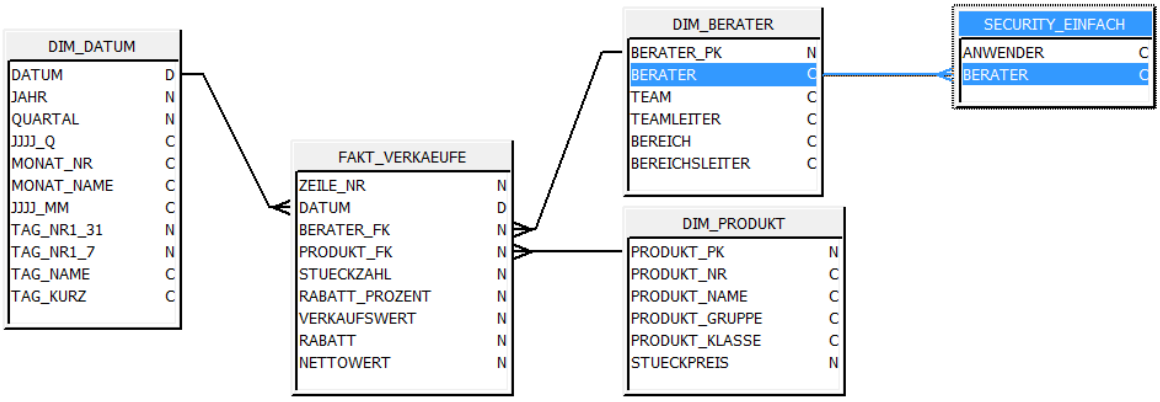
Es muss dringend darauf geachtet werden, dass keine doppelten Einträge in die Tabelle erfolgen, dass also die Zuordnung Anwender zu Berater eindeutig ist.

Andernfalls kommt es bei Abfrage der Faktentabelle, also bei der Ermittlung der Kennzahlen später zu Mehrfachzählungen.

Dies ist auch später im Fall von Knotenzuweisungen zu berücksichtigen, Abschnitt 5.2 geht darauf genauer ein.

4.2 Integration der Securitytabelle in die Tabellenstruktur

Wir erweitern nun die Tabellenstruktur, indem die Tabelle SECURITY_EINFACH mit der Beraterdimension DIM_BERATER über das Feld BERATER verknüpfen. Dabei tritt allerdings ein Problem auf: In SECURITY_EINFACH ist das Feld BERATER kein Primärschlüssel: derselbe Berater kann verschiedenen Anwendern zugewiesen werden. Zunächst ist also die Kardinalität des Joins zwischen DIM_BERATER und SECURITY_EINFACH 1:n (wir gehen aktuell von einer nicht-historisierter Beraterdimension aus, daher ist dort der Berater auch ein alternativer Primärschlüssel zu BERATER_PK).

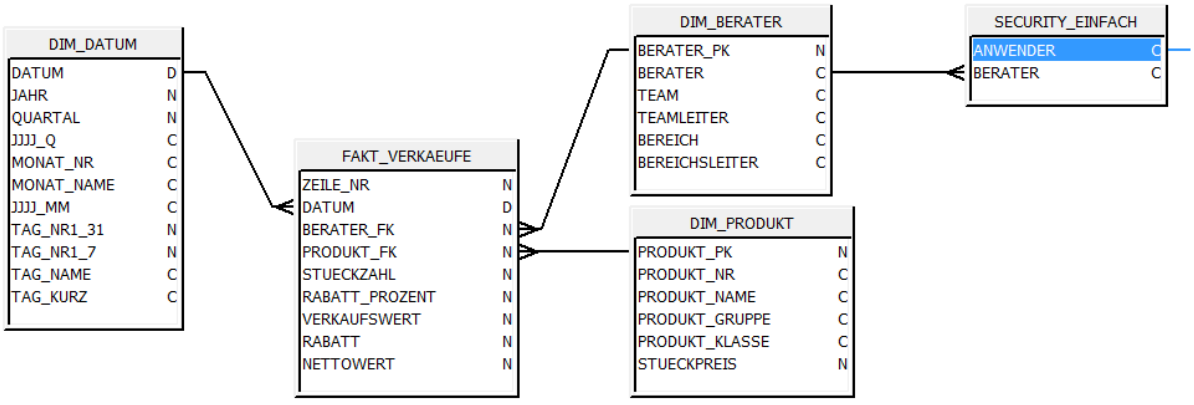


An dieser Stelle erfolgt die Einschränkung der Securitytabelle auf den aktuell angemeldeten Anwender. Damit erfolgt zum einen indirekt die Einschränkung aller anderen Daten auf die für den Anwender freigeschalteten Daten (dies ist das eigentliche Ziel!), zum anderen wird dadurch automatisch die oben erwähnte 1:n-Beziehung zwischen DIM_BERATER und SECURITY_EINFACH zur 1:1-Beziehung: Jeder Anwender erhält die für ihn erlaubten Berater genau einmal zugewiesen, damit ist der Berater für jeden Anwender wieder eindeutig[4].

Diese Einschränkung muss eine Zwangsbedingung sein, d.h. die Anwender dürfen sie nicht umgehen können, sie darf auch nicht optional sein. Wir symbolisieren diese Zwangsbedingung als Self Join an der Securitytabelle, mit folgender Bedingung:

SECURITY_EINFACH.ANWENDER = <aktueller Anwender>

<aktueller Anwender> steht hierbei für den aktuell im System angemeldeten Anwender, der als Parameter ans SQL übergeben werden muss[5].



4.3 *Beispiele für Abfrageergebnisse*

Als Beispielbericht erzeugen wir eine Abfrage auf die Beraterhierarchie, das Jahr und den Verkaufswert als Kennzahl, wobei wir zunächst die Securitytabelle noch nicht verwenden.

Wir erhalten das folgende Ergebnis (als Kreuztabelle aufbereitet):

			JAHR					
BEREICH	**TEAM**	**BERATER**	2010	2011	2012	2013	2014	2015
Euklid	Euklid	Euklid	130.323,00	60.920,00	128.919,89	137.049,77	86.499,09	88.603,40
	Pascal	Euler	49.190,84	63.470,03	79.303,63	82.324,10	156.895,63	106.244,09
		Gauss	194.708,00	178.760,00	130.260,74	127.583,14	97.731,38	52.093,05
		Germain	150.018,00	154.438,00	126.726,84	132.266,72	70.730,32	83.871,13
		Pascal	104.901,00	80.626,00	81.340,00	56.934,82	68.612,72	60.343,21
	Babbage	Babbage	239.987,00	249.631,89	173.792,78	152.825,79	94.115,87	143.013,61
		Turing	274.933,50	307.992,16	321.314,45	370.395,32	409.488,44	337.241,15
		v. Neumann	265.953,59	228.850,58	251.021,99	242.620,08	293.210,11	290.625,19

[4] Eine andere Betrachtungsweise ist: In SECURITY_EINFACH ist die *Kombination* der beiden Spalten ANWENDER und BERATER Primärschlüssel, daher ist für einen Anwender der Berater eindeutig.

[5] Die konkrete Ausführung ist werkzeugspezifisch. Als Beispiel: In SAP BusinessObjects kann seit Version XI 3.1 ein Bedingungsobjekt als verpflichtend definiert werden, es wird dann automatisch angewandt und kann auch nicht entfernt werden. Die Bedingung lautet dann SECURITY_EINFACH.ANWENDER=@variable('BOUSER')

	Riemann	Hilbert	105.249,02	114.704,73	146.108,23	146.035,82	162.171,01	206.808,89
		Noether	108.639,00	170.890,00	143.294,00	149.938,00	162.996,95	155.743,76
		Riemann	92.254,04	57.117,53	114.706,65	127.387,62	110.659,17	127.929,06

Newton	Newton	Newton	875,74	13.138,51	11.014,36	14.388,12	19.927,52	47.098,72
	Kopernikus	Galilei	127.322,93	135.801,10	64.013,86	52.467,05	49.518,79	59.713,46
		Kepler	36.470,00	45.890,00	171.390,00	170.842,00	195.593,00	181.883,00
		Kopernikus	137.247,00	116.952,00	59.616,37	40.558,40	10.900,21	12.686,42
	Einstein	Maxwell	114.619,00	115.937,00	87.782,00	41.324,41	38.885,92	31.350,69
		Einstein	23.168,49	10.089,11	25.698,73	55.095,40	57.687,38	85.034,61
	Planck	Bohr	114.173,06	114.436,51	94.613,04	96.883,05	63.040,75	87.394,17
		Dirac	87.736,25	26.702,50	35.555,47	38.310,63	58.437,28	35.721,95
		Feynman	88.192,00	141.152,00	141.566,00	133.730,00	124.917,35	100.403,67
		Planck	49.110,00	81.633,00	57.715,00	114.963,00	104.595,00	112.624,43
	Curie	Bequerel	4.455,28	9.467,57	31.055,18	50.339,32	124.118,97	141.961,86
		Curie	147.291,00	142.175,00	122.304,82	70.656,88	41.396,24	25.104,97

Als nächstes aktivieren wir die Security und führen denselben Bericht als Anwender Galilei aus. Dieser darf nur seine eigenen Daten sehen, weshalb er folgendes Ergebnis erhält:

			JAHR					
BEREICH	**TEAM**	**BERATER**	2010	2011	2012	2013	2014	2015
Newton	Kopernikus	Galilei	127.322,93	135.801,10	64.013,86	52.467,05	49.518,79	59.713,46

Analog sieht der Anwender Kepler nur seine eigenen Daten:

			JAHR					
BEREICH	**TEAM**	**BERATER**	2010	2011	2012	2013	2014	2015
Newton	Kopernikus	Kepler	36.470,00	45.890,00	171.390,00	170.842,00	195.593,00	181.883,00

Wenn nun ihr Teamleiter Kopernikus den Bericht ausführt, so sieht er die Daten der Berater Kepler, Galilei und Kopernikus:

			JAHR					
BEREICH	**TEAM**	**BERATER**	2010	2011	2012	2013	2014	2015
Newton	Kopernikus	Galilei	127.322,93	135.801,10	64.013,86	52.467,05	49.518,79	59.713,46
		Kepler	36.470,00	45.890,00	171.390,00	170.842,00	195.593,00	181.883,00
		Kopernikus	137.247,00	116.952,00	59.616,37	40.558,40	10.900,21	12.686,42

Der Bereichsleiter Newton sieht alle Daten seines Bereichs:

			JAHR					
BEREICH	**TEAM**	**BERATER**	2010	2011	2012	2013	2014	2015
Newton	Newton	Newton	875,74	13.138,51	11.014,36	14.388,12	19.927,52	47.098,72
	Kopernikus	Galilei	127.322,93	135.801,10	64.013,86	52.467,05	49.518,79	59.713,46
		Kepler	36.470,00	45.890,00	171.390,00	170.842,00	195.593,00	181.883,00
		Kopernikus	137.247,00	116.952,00	59.616,37	40.558,40	10.900,21	12.686,42
	Einstein	Maxwell	114.619,00	115.937,00	87.782,00	41.324,41	38.885,92	31.350,69
		Einstein	23.168,49	10.089,11	25.698,73	55.095,40	57.687,38	85.034,61
	Planck	Bohr	114.173,06	114.436,51	94.613,04	96.883,05	63.040,75	87.394,17
		Dirac	87.736,25	26.702,50	35.555,47	38.310,63	58.437,28	35.721,95
		Feynman	88.192,00	141.152,00	141.566,00	133.730,00	124.917,35	100.403,67
		Planck	49.110,00	81.633,00	57.715,00	114.963,00	104.595,00	112.624,43
	Curie	Bequerel	4.455,28	9.467,57	31.055,18	50.339,32	124.118,97	141.961,86
		Curie	147.291,00	142.175,00	122.304,82	70.656,88	41.396,24	25.104,97

Dem Anwender Vorstand wurden alle Rechte zugewiesen, daher sieht er den Bericht wie im originalen Zustand ohne Einschränkungen.

4.4 Ergänzungen, z.B. für Vertretungsregelungen

Grundsätzlich haben wir schon den Fall bearbeitet, dass einem Anwender mehrere Berater zugewiesen werden, im obigen Fall für Team- und Bereichsleiter.

Es könnte aber auch sonst der Fall auftreten, dass einem Anwender mehrere Berater zugewiesen werden, beispielsweise, um Vertretungsregelungen abzubilden. Als Beispiel betrachten wir die drei Berater im Team Pascal, also Euler, Germain und Gauss. Als Vertretungsregelung soll Euler Germain, Germain Gauss und Gauss Euler vertreten und daher die entsprechenden Daten sehen.

Dazu muss die Securitytabelle für die drei Berater die folgenden Einträge enthalten (die zusätzlichen Einträge sind grün markiert):

ANWENDER	**BERATER**
Euler	Euler
Euler	Germain
Germain	Germain
Germain	Gauss
Gauss	Gauss
Gauss	Euler

Wenn nun beispielsweise der Anwender Euler den obigen Bericht ausführt, so sieht er folgendes Ergebnis:

			JAHR					
BEREICH	**TEAM**	**BERATER**	2010	2011	2012	2013	2014	2015
Euklid	Pascal	Euler	49.190,84	63.470,03	79.303,63	82.324,10	156.895,63	106.244,09
		Germain	150.018,00	154.438,00	126.726,84	132.266,72	70.730,32	83.871,13

Als weiteres Beispiel nimmt der Teamleiter Riemann eine Auszeit, während der der Teamleiter Pascal seine Teamleitung kommissarisch übernimmt, daher muss Pascal die folgenden Einträge in der Securitytabelle erhalten (zusätzliche Einträge des Teams Riemann sind wieder grün markiert):

ANWENDER	**BERATER**
Pascal	Pascal
Pascal	Euler
Pascal	Germain
Pascal	Gauss
Pascal	Riemann
Pascal	Hilbert
Pascal	Noether

Damit erhält Pascal die folgenden Ergebnisse angezeigt:

			JAHR					
BEREICH	**TEAM**	**BERATER**	2010	2011	2012	2013	2014	2015
Euklid	Pascal	Euler	49.190,84	63.470,03	79.303,63	82.324,10	156.895,63	106.244,09
		Gauss	194.708,00	178.760,00	130.260,74	127.583,14	97.731,38	52.093,05
		Germain	150.018,00	154.438,00	126.726,84	132.266,72	70.730,32	83.871,13
		Pascal	104.901,00	80.626,00	81.340,00	56.934,82	68.612,72	60.343,21
	Riemann	Hilbert	105.249,02	114.704,73	146.108,23	146.035,82	162.171,01	206.808,89
		Noether	108.639,00	170.890,00	143.294,00	149.938,00	162.996,95	155.743,76
		Riemann	92.254,04	57.117,53	114.706,65	127.387,62	110.659,17	127.929,06

4.5 Anforderungen an das Berichtswerkzeug; semantische Schichten

Die im Abschnitt 4.2 beschriebenen Erweiterungen der Tabellenstruktur sind besonders einfach in die semantische Schicht eines Berichtswerkzeugs zu integrieren:

- Die erzwungene Verwendung der Securitytabelle kann bei Verwendung einer semantischen Schicht einfach gesteuert werden.
- Der Benutzer wird vom direkten Zugriff auf die Datenbank entkoppelt, so dass auch keine Möglichkeit der Manipulation der zugewiesenen Rechte existiert.

Daraus resultieren aber auch einige Anforderungen an das Berichtswerkzeug:

- Es muss eine Benutzerverwaltung erlauben (ansonsten gibt es keinen aktuell angemeldeten Anwender).
- Dieser aktuelle Anwender muss als Parameter an das SQL übergeben werden können.

- Es muss eine Möglichkeit zur Definition von Zwangsbedingungen geben.
- Den Anwendern muss das Recht, das auszuführende SQL manuell zu ändern oder anzupassen, weggenommen werden: Andernfalls könnte die Sicherheitseinschränkung einfach entfernt werden.
- Aus demselben Grund dürfen die Anwender auch keinen direkten SQL-Zugriff auf die Datenbank erhalten.

5 Einschränkung auf Knoten; Zuweisung von „Alle“-Rechten

5.1 Erweiterung der Securitytabelle auf Knoten

5.1.1 Einfaches Grundbeispiel: Jeder Anwender erhält den eigenen Knoten

Das in Abschnitt 4 vorgestellte Verfahren ist grundsätzlich sehr einfach in der Anwendung und Pflege. Es hat aber den Nachteil, dass nur Zuweisungen auf der untersten Ebene (Berater) erfolgen können, was für Anwender mit sehr weitreichenden Rechten (Bereichsleiter, Vorstände) sehr mühsam werden kann, insbesondere, weil sich die Bereichshierarchie ändern kann und jede solche Änderung entsprechende Auswirkungen in der Securitytabelle hat.

Es wäre viel praktischer und übersichtlicher, wenn Einschränkungen auf allen Ebenen (Knoten) der Beraterhierarchie erfolgen könnten. Genau dies wird im Folgenden beschrieben.

Wir gehen wieder von der in Abschnitt 2.2.2 beschriebenen Beraterdimension aus.

Wir wollen nun dem Teamleiter Planck nicht mehr die vier Berater Planck, Bohr, Dirac und Feynman einzeln zuweisen, sondern stattdessen einfach den Knoten (Team) Planck. Ebenso soll der Bereichsleiter Euklid ebenfalls den Knoten Euklid zugewiesen bekommen. Zum Vergleich: In der obigen Securitytabelle würden diese Einträge dazu führen, dass diese beiden Anwender genau die Sätze sehen würden, die sie selber erzeugt haben, also nicht alle aus ihrem Team bzw. Bereich.

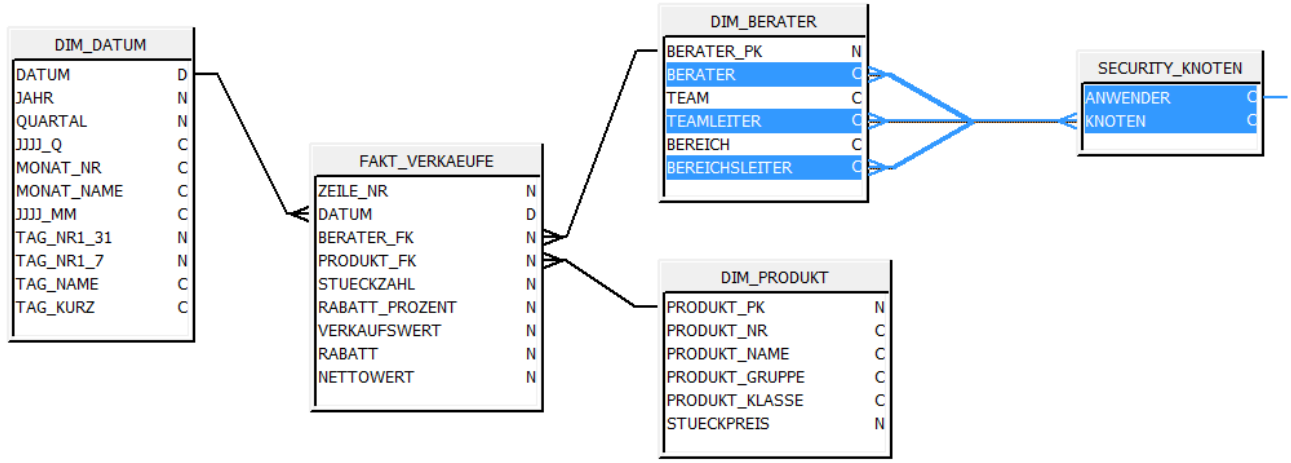
Die Securitytabelle sieht also in diesem Zustand so aus, dass jedem der Anwender der eigene Knoten zugewiesen bekommt:

ANWENDER	KNOTEN
Newton	Newton
Kopernikus	Kopernikus
Kepler	Kepler
Galilei	Galilei
Einstein	Einstein
Maxwell	Maxwell
Planck	Planck
Bohr	Bohr
Dirac	Dirac
Feynman	Feynman
Curie	Curie
Bequerel	Bequerel
Euklid	Euklid
Pascal	Pascal
Euler	Euler
Germain	Germain
Gauss	Gauss
Riemann	Riemann
Noether	Noether
Hilbert	Hilbert
Babbage	Babbage
v. Neumann	v. Neumann
Turing	Turing

Offenkundig kann die Verknüpfung der Tabelle nun nicht mehr allein über das Feld BERATER der Organisationshierarchie erfolgen, da der Knoten ein beliebiger sein kann.

Im konkreten Fall der obigen dreistufigen Hierarchie kann der Knoten ein BERATER, ein TEAMLEITER oder ein BEREICHSLEITER in DIM_BERATER sein, was im SQL drei Bedingungen entspricht. Da es ausreichend ist, dass eine davon erfüllt ist, um die Berechtigung für die entsprechenden Sätze zugewiesen zu bekommen, werden die drei Bedingungen mit einem logischen OR verknüpft.

Die neue Datenbankstruktur sieht nun (unter Verwendung der neuen Securitytabelle SECURITY_KNOTEN) also wie folgt aus:



mit der folgenden Joinbedingung:

```
DIM_BERATER.BERATER=SECURITY_KNOTEN.KNOTEN
OR
DIM_BERATER.TEAMLEITER=SECURITY_KNOTEN.KNOTEN
OR
DIM_BERATER.BEREICHSLEITER=SECURITY_KNOTEN.KNOTEN
```

Dieser Join hat die Kardinalität n:m, da in der Securitytabelle grundsätzlich immer noch verschiedene Anwender denselben Knoten zugewiesen bekommen können (dies wird weiter unten auch wieder geschehen) und weil derselbe Knoten nun auch auf mehrere Sätze in der Dimension verweisen kann: Beispielsweise bezieht sich die Bedingung Bereich = Newton auf insgesamt 12 Einträge.

Zusätzlich wird wieder die Securityeinschränkung, also die Einschränkung auf den aktuell angemeldeten Anwender, eingefügt (oben wieder als Self-Join auf SECURITY_KNOTEN dargestellt):

```
SECURITY_KNOTEN.ANWENDER=<aktueller Anwender>
```

Da auch hier wieder die Eindeutigkeit der Zuordnung Anwender – Knoten notwendige Bedingung ist (siehe Abschnitt 4.1 Ende), ist mit dieser Bedingung der Knoten wieder eindeutig und somit die gesamte Kardinalität zwischen DIM_BERATER und SECURITY_KNOTEN wieder n:1, wodurch Mehrfachzählungen wieder verhindert werden.

Der vorherige Absatz scheint momentan noch etwas übertrieben, da bisher jedem Anwender exakt sein eigener Knoten zugewiesen wurde, dies wird sich aber im Folgenden ändern.

5.1.2 Detaillierte Betrachtung der Verknüpfung

Bevor wir weitere Knoten zuweisen, erfolgt hier zunächst noch eine detaillierte Betrachtung zur Frage, warum das zuvor geschilderte Verfahren so korrekt ist und nicht zu Mehrfachzählungen führt.

Wir betrachten dazu wieder die drei Anwender Galilei (Berater), Kopernikus (Teamleiter) und Newton (Bereichsleiter), die sich anmelden und Berichte ausführen.

Der Berater Galilei hat in der Securitytabelle den Eintrag

ANWENDER	KNOTEN
Galilei	Galilei

Dies ist also die wirksame Zeile, wenn Galilei sich anmeldet.

In der Dimensionstabelle gibt es genau einen Satz, der in einer der drei Spalten BERATER, TEAMLEITER und BEREICHSLEITER den Eintrag „Galilei" enthält, nämlich den des Beraters Galilei:

BERATER_PK	BERATER	TEAM	TEAMLEITER	BEREICH	BEREICHSLEITER
4	**Galilei**	Klassische Physiker	Kopernikus	Physiker	Newton

Betrachten wir nun nochmals die Joinbedingung:

```
DIM_BERATER.BERATER=SECURITY_KNOTEN.KNOTEN
OR
DIM_BERATER.TEAMLEITER=SECURITY_KNOTEN.KNOTEN
OR
DIM_BERATER.BEREICHSLEITER=SECURITY_KNOTEN.KNOTEN
```

Die erste Teilbedingung ist für den obigen Dimensionseintrag (Berater Galilei) erfüllt. Die zwei weiteren Bedingungen sind nicht erfüllt. Da sie aber über ein OR verknüpft sind, verhindern sie nicht die Anwendung der ersten.

Als Resultat erhält der Anwender Galilei, wie gewünscht, seinen eigenen Dimensionseintrag zugewiesen und damit auch die zugehörigen Faktensätze.

Der Teamleiter Kopernikus hat in der Securitytabelle den Eintrag

ANWENDER	KNOTEN
Kopernikus	Kopernikus

In der Dimensionstabelle gibt es drei Einträge, in denen der Eintrag Kopernikus vorkommt:

BERATER_PK	BERATER	TEAM	TEAMLEITER	BEREICH	BEREICHSLEITER
2	**Kopernikus**	Klassische Physiker	**Kopernikus**	Physiker	Newton
3	Kepler	Klassische Physiker	**Kopernikus**	Physiker	Newton
4	Galilei	Klassische Physiker	**Kopernikus**	Physiker	Newton

Für die Berater Kepler und Galilei ist die mittlere der drei Bedingungen im Join, also

```
DIM_BERATER.TEAMLEITER=SECURITY_KNOTEN.KNOTEN
```

erfüllt, daher darf Kopernikus diese zwei Sätze sehen. Die beiden anderen Bedingungen sind nicht erfüllt, spielen aber wegen der OR-Verknüpfung wieder keine Rolle.

Im ersten Satz, also dem des Beraters Kopernikus, sind nun die ersten zwei der Bedingungen erfüllt (die dritte ist wieder irrelevant):

```
DIM_BERATER.BERATER=SECURITY_KNOTEN.KNOTEN
OR
DIM_BERATER.TEAMLEITER=SECURITY_KNOTEN.KNOTEN
```

Führt dies nicht zu einer Mehrfachzuweisung?

Die Antwort darauf lautet (glücklicherweise!): Nein! Die Joinbedingung führt nicht dazu, dass für jede erfüllte Bedingung ein neuer Ergebnissatz generiert wird, vielmehr wird die Bedingung als Ganzes geprüft und bei Erfüllung ein Ergebnissatz erzeugt.

Da die drei Bedingungen mit OR verknüpft sind, genügt es, wenn mindestens eine davon erfüllt ist, es dürfen aber auch (wie in diesem Fall) mehr sein.

Mit derselben Argumentation erhält der Bereichsleiter Newton die korrekten Ergebnisse seines Bereichs angezeigt.

In der Securitytabelle lautet der Eintrag

ANWENDER	KNOTEN
Newton	Newton

In der Dimensionstabelle gibt es dazu 12 Einträge:

BERATER_PK	BERATER	TEAM	TEAMLEITER	BEREICH	BEREICHSLEITER
1	**Newton**	Leitung Physiker	**Newton**	Physiker	**Newton**
2	Kopernikus	Klassische Physiker	Kopernikus	Physiker	**Newton**
3	Kepler	Klassische Physiker	Kopernikus	Physiker	**Newton**
4	Galilei	Klassische Physiker	Kopernikus	Physiker	**Newton**
5	Einstein	Elektrodynamiker	Einstein	Physiker	**Newton**
6	Maxwell	Elektrodynamiker	Einstein	Physiker	**Newton**
7	Planck	Quantenphysiker	Planck	Physiker	**Newton**
8	Bohr	Quantenphysiker	Planck	Physiker	**Newton**
9	Dirac	Quantenphysiker	Planck	Physiker	**Newton**
10	Feynman	Quantenphysiker	Planck	Physiker	**Newton**
11	Curie	Radioaktivitätsforscher	Curie	Physiker	**Newton**
12	Bequerel	Radioaktivitätsforscher	Curie	Physiker	**Newton**

Für die Berater Kopernikus bis Bequerel ist die dritte Teilbedingung erfüllt, also

```
DIM_BERATER.BEREICHSLEITER=SECURITY_KNOTEN.KNOTEN
```

Für den Berater Newton sind sogar alle drei Bedingungen erfüllt, was aber wieder nicht zu Mehrfachzählungen führt.

5.1.3 Ergänzung weiterer Einträge

Bisher wurde jedem Anwender genau sein eigener Knoten zugewiesen. Dies ist häufig aber nicht ausreichend:

- Ein Anwender kann auch Zugriff auf weitere Informationen benötigen, beispielsweise aufgrund einer Vertretungsregelung wie in Abschnitt 4.4 beschrieben; siehe dazu Abschnitt 5.1.3.1
- Im Berichtswesen kann es auch Anwender geben, die Datenzugriffe benötigen, ohne dass sie selber Daten erfassen. Ebenso haben manche Mitarbeiter evtl. keine Analyseberechtigungen, auch wenn sie Daten erfassen. Siehe dazu Abschnitt 5.1.3.2
- Schließlich kann es auch Anwender mit Gesamtrechten geben, wie die Gesamtvertriebsleitung, das zentrale Controlling, der Vorstand usw. Siehe dazu Abschnitt 5.1.3.3

Wir gehen im Weiteren davon aus, dass entsprechende Anwender im Berichtswesen bei Bedarf eingerichtet wurden, wir also nur noch die notwendigen Datenberechtigungen zuweisen müssen.

Die oben vorgestellte Securitytabelle bietet eine sehr einfache Möglichkeit, diese zusätzliche Anforderung zu erfüllen, indem weitere Einträge ergänzt werden.

5.1.3.1 Zusätzliche Einträge für bestehende Anwender

Grundsätzlich kann jedem Anwender jeder beliebige Knoten zugewiesen werden. Der Berater Galilei kann also als Vertretung von Kepler eingesetzt werden, indem der entsprechende Eintrag eingefügt wird (sehr ähnlich wie in Abschnitt 4.4):

ANWENDER	**KNOTEN**
Galilei	Galilei
Galilei	**Kepler**

Im Beispiel aus Abschnitt 4.4 wurde Pascal als Urlaubsvertretung des Teamleiters Riemann eingesetzt, daher benötigt Pascal die folgenden Einträge:

ANWENDER	**BERATER**
Pascal	Pascal
Pascal	**Riemann**

Es kann auch der Fall vorkommen, dass der Teamleiter Einstein zusätzlich die Daten von Bohr sehen darf[6]. Damit benötigt Einstein die folgenden Einträge:

ANWENDER	**BERATER**
Einstein	Einstein
Einstein	**Bohr**

Dies zeigt, wie oben erwähnt, dass Anwendern aus allen Ebenen beliebige Knoten zugeordnet werden können (Einstein ist Teamleiter, Bohr Berater). Genauso könnte der Bereichsleiter Newton auch noch das Team Pascal und die Beraterin Noether zugewiesen bekommen:

ANWENDER	**KNOTEN**
Newton	Newton
Newton	**Pascal**
Newton	**Noether**

Alle bisherigen Ergänzungen hatten eine Gemeinsamkeit: Die zusätzlichen Einträge waren immer aus anderen Hierarchiezweigen, so dass es zu keinen Überlappungen gekommen ist.

Was passiert aber in der folgenden Situation?

Berater Feynman wird als Stellvertreter von Planck benannt und bekommt dessen Knoten zugewiesen. Damit hat Feynman die folgenden Einträge:

ANWENDER	**KNOTEN**
Feynman	Feynman
Feynman	**Planck**

[6] In den Organigrammen großer Firmen können „dotted lines" vorkommen, in denen eine Mitbetreuung bzw. eine zusätzliche Berichtshierarchie dargestellt wird. Dies könnte eine konkrete Anwendung sein.

Nun ist Feynman aber bereits Mitglied des Teams Planck, wodurch eine Überlappung entsteht: Der Dimensionssatz zum Berater Feynman wird zweimal zugewiesen. Der Umgang mit Überlappungen wird in Abschnitt 5.2 ausführlich behandelt.

Eine andere Frage ist, wie die Knotenebene bei der Einschränkung definiert werden kann, wie in folgendem Beispiel:

Der Berater Galilei übernimmt während des Urlaubs die Vertretung für den Teamleiter Kopernikus, *aber nur für dessen eigene Verträge*. Galilei darf also nicht das gesamte Team Kopernikus sehen, sondern nur die des Beraters Kopernikus. Der Eintrag

ANWENDER	KNOTEN
Galilei	Kopernikus

würde aber das gesamte Team als Berechtigung zuweisen. Dieser Fall wird in Abschnitt 5.3 behandelt.

5.1.3.2 Zusätzliche Anwender/Entfernen von Berechtigungen

Wir hatten bereits in Abschnitt 4.1 einen zusätzlichen Anwender „Vorstand" in die Securitytabelle eingetragen, der eigentlich nicht in der Beraterhierarchie vorkommt, aber dennoch Rechte auf das Berichtswesen hat.

Grundsätzlich lassen sich solche Anwender mit beliebigen Rechten ausstatten, das Prinzip ist das selbe wie oben: Der Anwender wird mit den für ihn freigegebenen Knoten in die Securitytabelle eingetragen. Er muss aber auch als Anwender im Berichtswerkzeug angelegt und mit den entsprechenden Zugriffen ausgestattet sein (dies gilt aber generell für alle Anwender).

Als erstes Beispiel definieren wir einen neuen Anwender Cantor, der auch die Daten des Beraters Gauss sehen darf, so dass er den folgenden Eintrag in der Securitytabelle benötigt:

ANWENDER	KNOTEN
Cantor	Gauss

Der neue Anwender Rutherford darf die Daten des Teams Planck und die des Beraters Kepler sehen:

ANWENDER	KNOTEN
Rutherford	Planck
Rutherford	Kepler

Und schließlich darf der neue Mitarbeiter Leibnitz alle Daten des Bereichs Newton, sowie das Team Pascal sehen:

ANWENDER	KNOTEN
Leibnitz	Newton
Leibnitz	Pascal

Das einfache Grundprinzip ist also weiterhin, dass dem Anwender alle Knoten, die er sehen darf, zugewiesen werden. Solange es zu keinen Überlappungen kommt, ist dieses Verfahren auch korrekt.

Sollen einem Anwender die Rechte entzogen werden, ohne dass er als Anwender des Berichtswerkzeugs gelöscht werden soll, so werden seine Einträge in der Securitytabelle einfach gelöscht, dadurch sind sofort alle Datenrechte entfernt.

5.1.3.3 Anwender mit Gesamtrechten

Es gibt Anwender, die Zugriffe auf alle Daten eines Systems benötigen, etwa Vorstände, zentrale Controller, Gesamtvertriebsleiter usw..

Eine Möglichkeit wäre, diese Anwender in der Securitytabelle mit allen Knoten auf oberster Ebene einzutragen, also im obigen Beispiel etwa so (im Beispiel ist „Vorstand" ein Anwender aus dem Vorstand und „Archimedes" ein zentraler Controller):

ANWENDER	KNOTEN
Vorstand	Newton
Vorstand	Euklid
Archimedes	Newton
Archimedes	Euklid

Dies ist ein korrektes Verfahren, es ist allerdings ein wenig umständlich:

- Bei Umstrukturierungen müssen die Einträge nachbearbeitet werden
- Insbesondere bei sehr großen Organisationsstrukturen können viele Einträge nötig sein (nicht nur zwei, wie im Beispiel).

Es wäre sehr viel einfacher, für diese Anwender ein „ALLE"-Recht zuweisen zu können.

Die meisten Dimensionstabellen enthalten aber keinen eigenen obersten Alle-Knoten, auf den diese Bedingung angewandt werden kann (das wäre natürlich ein gangbarer Weg!).

Es gibt aber eine andere Möglichkeit, dies zu realisieren.

Wir gehen zunächst von folgenden Einträgen in der Securitytabelle aus:

ANWENDER	KNOTEN
Vorstand	ALLE
Archimedes	ALLE

Für alle Anwender mit diesem Recht sollte effektiv die Einschränkung verschwinden, anders gesagt sollte die Joinbedingung zwischen SECURITY_KNOTEN und DIM_BERATER äquivalent zu 1=1 (also immer für alle Dimensionseinträge erfüllt) sein.

Eine Bedingung, die dieser Anforderung entspricht, ist

```
SECURITY_KNOTEN.KNOTEN = 'ALLE'
```

Für alle anderen Anwender ist die Bedingung nicht erfüllt, dafür aber eine oder mehrere der anderen in der Joinbedingung. Für die Anwender mit „ALLE"-Recht wiederum ist nur die neue Bedingung erfüllt, alle anderen Teilbedingungen des Joins sind nicht erfüllt.

Damit muss die neue Bedingung in den Join mit aufgenommen werden und mit OR verknüpft werden. Die Gesamtbedingung lautet damit wie folgt[7]:

```
DIM_BERATER.BERATER=SECURITY_KNOTEN.KNOTEN
OR
DIM_BERATER.TEAMLEITER=SECURITY_KNOTEN.KNOTEN
OR
DIM_BERATER.BEREICHSLEITER=SECURITY_KNOTEN.KNOTEN
OR
SECURITY_KNOTEN.KNOTEN = 'ALLE'
```

[7] Ein Hinweis zur Abfrageperformance: Für manche Datenbanken könnte es sinnvoll sein, die ALLE-Bedingung an den Anfang des Ausdrucks zu setzen, weil sie zuerst geprüft wird. Sollte der Securityeintrag ALLE sein, so erkennt die Datenbank die Bedingung 1=1 und muss nicht mehr weiter prüfen. Die Reihenfolge der Prüfung und die Effektivität dieser 1=1-Bedingung ist aber vom Datenbanksystem abhängig.

5.1.4 Betrachtungen zur Abfrageperformance

An dieser Stelle muss die Frage gestellt werden, wie sich dieses Security-Design auf die Abfrageperformance auswirkt. Die Antwort ist nicht einfach und auch eventuell datenbankspezifisch.

Zunächst sind folgende positive Effekte für die Abfragegeschwindigkeit aufzuführen:

- In jeder Abfrage erfolgt die Einschränkung der jeweiligen Securitytabelle auf den aktuell angemeldeten Anwender. Damit wird immer nur ein (wahrscheinlich kleiner) Teil der Securitytabelle in die Abfrage mit einbezogen.
- Auch die Organisationshierarchie selber ist meistens eine eher kleine Dimensionstabelle im Kontext des gesamten Data Warehouses/Data Marts. Zusätzlich wird sie über die Security eingeschränkt.

Es gibt aber auch einige negative Effekte:

- Eine Joinbedingung, die verschiedene Bedingungen mit OR verknüpft, erlaubt für gewöhnlich keinen Indexzugriff auf die davon betroffenen Felder[8]. Das wäre bei einer relativ kleinen Tabelle wie der Organisationshierarchie wieder nicht sehr problematisch, aber:
- Der Optimierungsprozess und der daraus resultierende Ausführungsplan der Abfrage kann von dieser Joinbedingung erheblich gestört werden, woraus sehr langsame Abfragen resultieren können.
- Insbesondere bei Anwendern mit sehr weitreichenden Rechten (ALLE, Bereichsleiter,...) führt die Einschränkung zu einer sehr großen Untermenge der Dimensionstabelle, was den Indexzugriff auf die Datenbank wieder verhindern kann.

Andererseits ist die Securitybedingung aus fachlicher Sicht unverzichtbar, zudem steht es den Anwendern jederzeit frei, zusätzliche Bedingungen einzufügen, die die Performance wieder verbessern.

Dennoch bleibt die Optimierung der Abfrageperformance eine wichtige Thematik, insbesondere wenn, wie im weiteren Verlauf des Dokuments beschrieben, komplexe Zusatzlogiken angewandt werden. In Abschnitt 12 wird das Thema Abfrageperformance nochmals ausführlich betrachtet.

5.2 Überlappungen und die Vermeidung von Mehrfachzählungen

5.2.1 Erläuterung der Problematik

In Abschnitt 5.1.3.1 wurde als Beispiel dem Anwender Feynman außer seinem eigenen Knoten als Vertretungsregel auch der seines Teamleiters Planck zugeiwesen:

ANWENDER	**KNOTEN**
Feynman	Feynman
Feynman	**Planck**

Feynman ist aber bereits Mitglied des Teams Planck, wodurch ein Überlappung entsteht: Der Berater Feynman wird zweimal zugewiesen. **Dies sollte nicht mit der Begründung, die OR-Verknüpfung im Join führt nicht zur Mehrfachzählungen, verwechselt werden:**

Die Joinbedingung prüft für jeden Referenzsatz aus der Securitytabelle, welche Sätze aus der Dimension zuzuordnen sind, wenn nämlich mindestens eine der Teilbedingungen des Joins erfüllt wird. Solange es keine überlappenden Zuweisungen in der Securitytabelle gibt, wird damit (je Anwender natürlich) jeder Dimensionssatz entweder gar nicht (keine Berechtigung) oder genau einmal (Berechtigung gegeben) zugewiesen, was korrekt ist.

Die überlappenden Einträge führen nun aber zu einer Mehrfachzählung, was an einem Beispiel erläutert werden soll.

[8] Dies ist sehr stark von der verwendeten Datenbank abhängig!

Der oben erwähnte Bericht enthält (noch ohne Security) für das Team Planck folgende Informationen:

			JAHR					
BEREICH	**TEAM**	**BERATER**	2010	2011	2012	2013	2014	2015
Newton	Planck	Bohr	114.173,06	114.436,51	94.613,04	96.883,05	63.040,75	87.394,17
		Dirac	87.736,25	26.702,50	35.555,47	38.310,63	58.437,28	35.721,95
		Feynman	88.192,00	141.152,00	141.566,00	133.730,00	124.917,35	100.403,67
		Planck	49.110,00	81.633,00	57.715,00	114.963,00	104.595,00	112.624,43

Wir ergänzen nun den Knoten aus der Securitytabelle in das Ergebnis, wobei die Einschränkung auf den Anwender Feynman erfolgt:

				JAHR					
BEREICH	**TEAM**	**BERATER**	**KNOTEN**	2010	2011	2012	2013	2014	2015
Newton	Planck	Bohr	Planck	114.173,06	114.436,51	94.613,04	96.883,05	63.040,75	87.394,17
		Dirac	Planck	87.736,25	26.702,50	35.555,47	38.310,63	58.437,28	35.721,95
		Feynman	**Feynman**	**88.192,00**	**141.152,00**	**141.566,00**	**133.730,00**	**124.917,35**	**100.403,67**
		Feynman	**Planck**	**88.192,00**	**141.152,00**	**141.566,00**	**133.730,00**	**124.917,35**	**100.403,67**
		Planck	Planck	49.110,00	81.633,00	57.715,00	114.963,00	104.595,00	112.624,43

Die beiden fettgedruckten Zeilen zeigen dasselbe Ergebnis, das aber den beiden (Security-) Knoten Feynman und Planck zugewiesen wird.

Dies geschieht aber nicht, weil der Knoten in die Ergebnistabelle ergänzt wurde, sondern weil die Securitytabelle in der Abfrage enthalten ist (genauer: weil der Join zwischen DIM_BERATER und SECURITY_KNOTEN zu der Mehrfachzuordnung geführt hat). Dies ist aber laut Anforderung immer der Fall!

Wenn nun also der Knoten nicht in der Tabelle angezeigt wird, werden die Ergebnisse für Feynman aufsummiert (gruppiert) und deshalb verdoppelt:

			JAHR					
BEREICH	**TEAM**	**BERATER**	2010	2011	2012	2013	2014	2015
Newton	Planck	Bohr	114.173,06	114.436,51	94.613,04	96.883,05	63.040,75	87.394,17
		Dirac	87.736,25	26.702,50	35.555,47	38.310,63	58.437,28	35.721,95
		Feynman	**176.384,00**	**282.304,00**	**283.132,00**	**267.460,00**	**249.834,70**	**200.807,34**
		Planck	49.110,00	81.633,00	57.715,00	114.963,00	104.595,00	112.624,43

Dies ist übrigens immer der Fall, wenn eine Faktentabelle mit einer anderen Tabelle über eine effektive n:m-Beziehung verknüpft wird.

5.2.2 Lösungsansatz 1: Nur Einträge ohne Überlappungen in der Securitytabelle

Dieser Ansatz klingt zunächst fast zu einfach, er kann aber tatsächlich zielführend sein!

Die grundlegende Frage ist: Warum benötigt ein Anwender einen untergeordneten Knoten, wenn er bereits den übergeordneten sehen darf?

Daraus leitet sich die Verarbeitungslogik ab, dass jedem Anwender für jeden Hierarchiepfad jeweils der höchste komplett sichtbare Knoten zugewiesen wird, so dass Überlappungen vermieden werden. Dies ist

insbesondere dann eine wirksame Regel, wenn die Securitytabelle in einem automatisierten Prozess innerhalb des ETLs, also nicht durch manuell Pflege, erzeugt wird.

5.2.3 Lösungsansatz 2: Überlappungen eindeutig auflösen

5.2.3.1 Herstellung der Eindeutigkeit

In manchen Fällen erfolgt die Pflege der Securitytabelle manuell, außerdem werden wir im weiteren Verlauf des Dokuments (insbesondere in Abschnitt 7) sehen, dass auch andere Anforderungen notwendigerweise zu überlappenden Einträgen führen können, deshalb wird nun ein anderer Ansatz vorgestellt, wie die Überlappungen eindeutig aufgelöst werden können.
Dazu gehen wir zu dem in den Abschnitt 5.1.3.1 und 5.2.1 beschriebenen Beispiel des Anwender Feynman, dem außer seinem eigenen Knoten als Vertretungsregel auch der seines Teamleiters Planck zugeiwesen wurde:

ANWENDER	**KNOTEN**
Feynman	Feynman
Feynman	**Planck**

Damit wird sein eigener Eintrag in DIM_BERATER doppelt referenziert, was zu Mehrfachzählungen führt. Dies ist aber nur ein Sonderfall: Generell kann jeder Satz mehrfach referenziert werden und damit zu Fehlern führen[9].

Für die korrekte Ermittlung der Kennzahlen ist die „ungestörte" 1:n-Beziehung zwischen DIM_BERATER und der Faktentabelle wichtig, die fehlerhafte Kennzahlermittlung erfolgt, weil die mehrfache Referenz der Security de facto zu einer n:m-Beziehung führt.

Wenn die Mehrfachzählung also vermieden werden soll, so muss die Eindeutigkeit des Primärschlüssels in der Dimensionstabelle gewahrt werden. Der Primärschlüssel der Dimension ist der BERATER_PK, er ist im aktuellen Zustand (bevor die Dimension historisiert wird) noch gleichwertig mit dem Feld BERATER.

Die einfachste Methode, die Eindeutigkeit herzustellen, ist ein SELECT DISTINCT auszuführen, etwa mit folgender Abfrage (hier ist der Anwender auf „Feynman" gesetzt):

```
SELECT DISTINCT
  SECURITY_KNOTEN.ANWENDER,
  DIM_BERATER.BERATER_PK,
  DIM_BERATER.BERATER
FROM
  SECURITY_KNOTEN,
  DIM_BERATER
WHERE
  ( DIM_BERATER.BERATER=SECURITY_KNOTEN.KNOTEN
OR
DIM_BERATER.TEAMLEITER=SECURITY_KNOTEN.KNOTEN
OR
DIM_BERATER.BEREICHSLEITER=SECURITY_KNOTEN.KNOTEN
OR
SECURITY_KNOTEN.KNOTEN = 'ALLE')
  AND
  SECURITY_KNOTEN.ANWENDER  =  'Feynman'
```

[9] Im einfachsten Fall durch einen Fehler in der Pflege der Securitytabelle: Wenn derselbe Eintrag mehrfach geschrieben wird!

Das Ergebnis ist die folgende Tabelle:

ANWENDER	BERATER_PK	BERATER
Feynman	7	Planck
Feynman	8	Bohr
Feynman	9	Dirac
Feynman	10	Feynman

Die Spalte BERATER wurde der Übersichtlichkeit halber ergänzt, eigentlich wäre nur der BERATER_PK notwendig, aber wie oben erwähnt ist in diesem Zustand der Dimensionstabelle auch der Berater ein Primärschlüssel.

ACHTUNG:

Die Eindeutigkeit wird nur erreicht, weil die Spalte KNOTEN aus der Securitytabelle nicht in der Abfrage enthalten ist: Diese führt ja gerade zur Mehrfachzählung, die durch das DISTINCT entfernt wird. Wird die Spalte aber eingefügt, so unterscheiden sich die beiden Sätze für den BERATER_PK 10 (Feynman) wieder, weshalb das DISTINCT in diesem Fall nicht zu einer Verdichtung führt. Im Beispiel: Das SQL

```
SELECT DISTINCT
  SECURITY_KNOTEN.ANWENDER,
  SECURITY_KNOTEN.KNOTEN,
  DIM_BERATER.BERATER_PK,
  DIM_BERATER.BERATER
FROM
  SECURITY_KNOTEN,
  DIM_BERATER
WHERE
  ( DIM_BERATER.BERATER=SECURITY_KNOTEN.KNOTEN
OR
DIM_BERATER.TEAMLEITER=SECURITY_KNOTEN.KNOTEN
OR
DIM_BERATER.BEREICHSLEITER=SECURITY_KNOTEN.KNOTEN
OR
SECURITY_KNOTEN.KNOTEN = 'ALLE')
  AND
  SECURITY_KNOTEN.ANWENDER  =  'Feynman'
```

führt zu folgendem Ergebnis:

ANWENDER	KNOTEN	BERATER_PK	BERATER
Feynman	Planck	7	Planck
Feynman	Planck	8	Bohr
Feynman	Planck	9	Dirac
Feynman	**Planck**	**10**	**Feynman**
Feynman	**Feynman**	**10**	**Feynman**

5.2.3.2 *Aufbau eines Strukturelements; Integration in die Tabellenstruktur*

Wir gehen im Folgenden von der korrekten Auflösung aus.

Die Ergebnistabelle von oben sah wie folgt aus:

ANWENDER	BERATER_PK	BERATER
Feynman	7	Planck
Feynman	8	Bohr
Feynman	9	Dirac
Feynman	10	Feynman

Abgesehen von der Spalte BERATER_PK erinnert sie stark an die Grundversion der Security, wie sie in Abschnitt 4 (genauer: 4.1) vorgestellt wurde! Das ist natürlich auch kein Zufall, weil die eindeutige Zuordnung des Anwenders auf die für ihn freigestellten Berater genau das gewünschte Ergebnis ist, das auch die Mehrfachzählungen auflöst.

Es gibt allerdings drei sehr wichtige Unterschiede:

- Wir werden künftig das Feld BERATER wieder entfernen und nur noch den Primärschlüssel BERATER_FK beibehalten.
- In der Abfrage ist bereits die Einschränkung auf den aktuell angemeldeten Anwender erfolgt.
- Die Tabelle in Abschnitt 4.1 wurde vollständig auf Berater-Ebene gepflegt. Die hier vorgestellte Struktur wird aber aus den viel kompakteren Knoteneinschränkungen *abgeleitet*.

Die Frage ist zunächst, wie eine solche Abfrage in die Tabellenstruktur integriert werden kann.

Ein erster Ansatz wäre eine Datenbankview. Diese hat aber den Nachteil, dass der aktuell im Abfragewerkzeug angemeldete Anwender nicht einfach als Parameter an die View übergeben werden kann (außer jeder Anwender hat auch einen zugehörigen Datenbankuser, was eher unüblich ist).
Daher wählen wir fürs erste die Lösung einer abgeleiteten Tabelle[10]: Als Definition verwenden wir das folgende SQL

```
/*******************************************************/
/* Anfang der abgeleiteten Tabelle SECURITY_AUFGELOEST */
/*******************************************************/
SELECT DISTINCT
  SECURITY_KNOTEN.ANWENDER,
  DIM_BERATER.BERATER_PK
FROM
  SECURITY_KNOTEN,
  DIM_BERATER
WHERE
  ( DIM_BERATER.BERATER=SECURITY_KNOTEN.KNOTEN
OR
DIM_BERATER.TEAMLEITER=SECURITY_KNOTEN.KNOTEN
OR
DIM_BERATER.BEREICHSLEITER=SECURITY_KNOTEN.KNOTEN
OR
SECURITY_KNOTEN.KNOTEN = 'ALLE')
  AND
  SECURITY_KNOTEN.ANWENDER  =  <aktueller Anwender>
/*******************************************************/
/* Ende der abgeleiteten Tabelle SECURITY_AUFGELOEST */
/*******************************************************/
```

[10] Auf Englisch auch als „derived table“ oder als „inline view“ bezeichnet

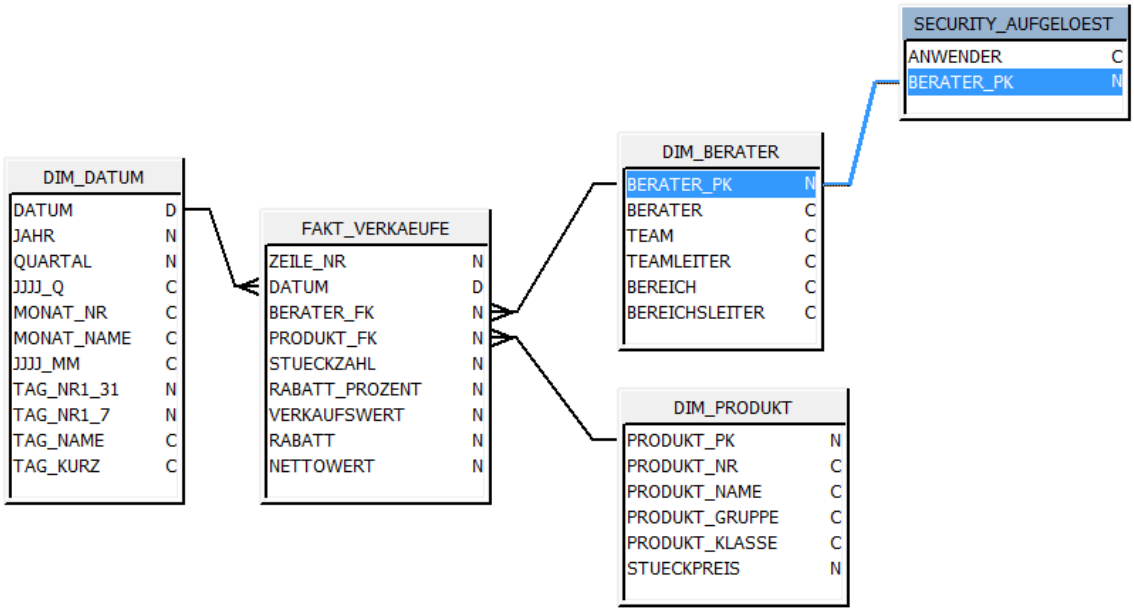
Zu diesem SQL folgende Hinweise:

- Es hat sich bewährt, in das Definitions-SQL von abgeleiteten Tabellen Anfangs- und Endkommentare einzufügen: dadurch wird die Lesbarkeit des finalen SQLs stark verbessert.
- In der Definition sind nun nur noch die Spalten ANWENDER und BERATER_PK enthalten. Genau genommen ist nicht einmal der Anwender wirklich notwendig, da dieser ja auf den aktuellen Anwender eingeschränkt ist. Die einzige relevante Spalte ist also der Primärschlüssel, der noch die Einträge enthält, die der aktuelle Anwender sehen darf!

Nun ist die Frage, wie diese abgeleitete Tabelle SECURITY_AUFGELOEST in die Tabellenstruktur eingebunden werden kann.

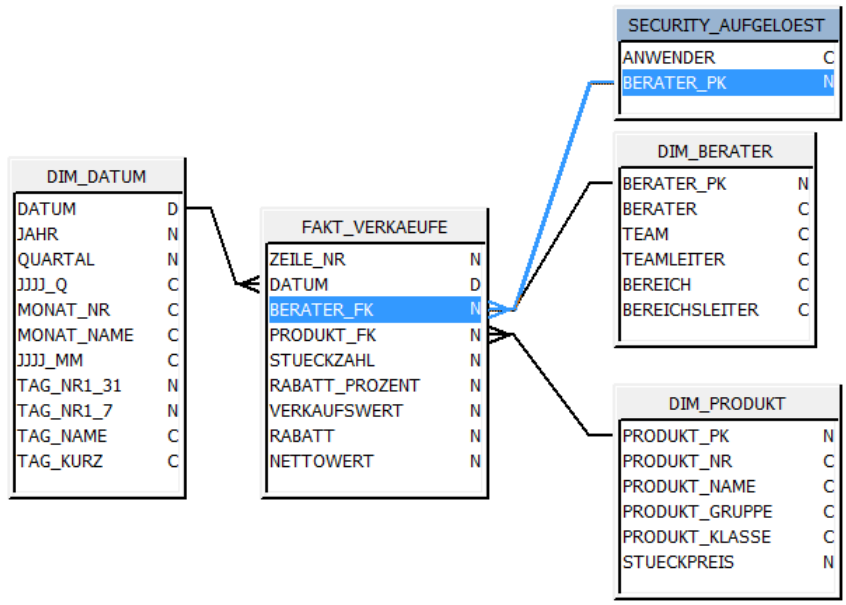
Hierfür gibt es verschiedene Möglichkeiten.

1.: Verknüpfung mit DIM_BERATER



Eine Einschränkung auf den aktuellen Anwender muss hier nicht mehr eingefügt werden, da sie ja bereits im Definitions-SQL der abgeleiteten Tabelle enthalten ist.

2.: Verknüpfung mit der Faktentabelle



Es ist vielleicht etwas überraschend, aber die aufgelöste Security kann nun auch direkt mit der Faktentabelle verknüpft werden:

- in der SQL-Definition wird nicht mehr das Feld BERATER, sondern der Primärschlüssel BERATER_PK verwendet. Da dieser als Fremdschlüssel in der Faktentabelle referenziert wird, ist die direkte Verknüpfung möglich.
- Die abgeleitete Tabelle enthält jetzt genau die für den aktuellen Anwender zugelassenen Schlüssel; damit erfolgt die notwendige Einschränkung.
- Die Dimensionstabelle muss nur noch mit abgefragt werden, wenn die entsprechenden Informationen im Bericht auch tatsächlich benötigt werden.

Welche der beiden Verknüpfungen gewählt wird, hängt von den Berichtsanforderungen ab: Wird fast in jedem Bericht die Organisationshierarchie mit abgefragt, so wird eher die erste Lösung gewählt, ansonsten eher die zweite. Es gibt aber noch eine andere Auflösung:

5.2.4 Vollständige Integration der Dimensionsdaten in die Auflösung

Im Auflösungs-SQL werden die Securitytabelle und DIM_BERATER miteinander abgefragt. Im späteren Berichts-SQL wird dann in den meisten Fällen die Dimension erneut eingebunden, weil die Information benötigt wird.

Eine Alternative dazu ist, alle Attribute der Dimension in die Auflösung mit aufzunehmen und diese später für die Abfragen zu verwenden. Damit ist die zusätzliche neue Aufnahme der Dimension im finalen SQL nicht mehr nötig.

Ein weiterer Vorteil wäre, dass die Beraterdimension nun „personalisiert" ist, also für jeden Anwender nur noch die erlaubten Sätze enthält. Damit sind auch alle benötigten Wertelisten automatisch personalisiert, d.h. es werden nur noch Werte angezeigt, die der jeweilige Anwender sehen darf.
Das zugehörige SQL für die Gesamtauflösung (wieder eine abgeleitete Tabelle) sieht nun wie folgt aus:

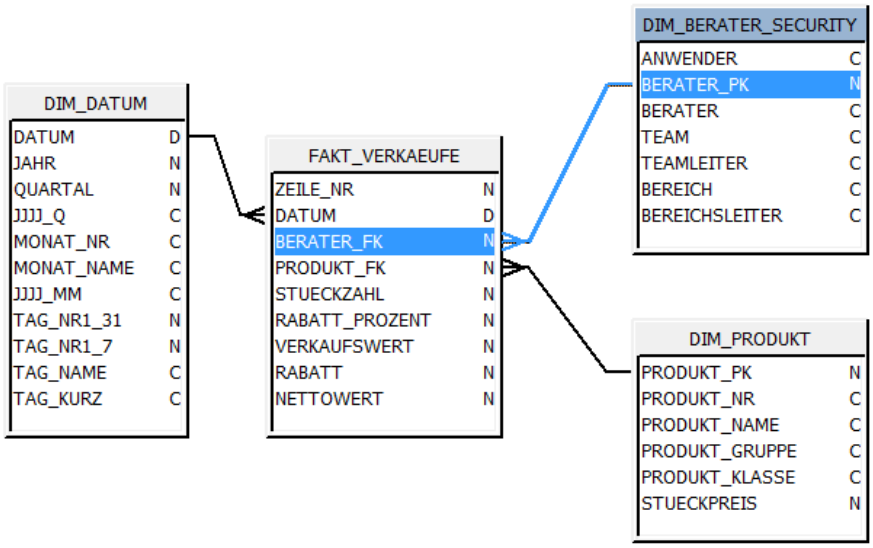
```
/*******************************************************/
/* Anfang der abgeleiteten Tabelle DIM_BERATER_SECURITY */
/*******************************************************/
SELECT DISTINCT
  SECURITY_KNOTEN.ANWENDER,
  DIM_BERATER.BERATER_PK,
  DIM_BERATER.BERATER,
  DIM_BERATER.TEAM,
  DIM_BERATER.TEAMLEITER,
  DIM_BERATER.BEREICH,
  DIM_BERATER.BEREICHSLEITER
FROM
  SECURITY_KNOTEN,
  DIM_BERATER
WHERE
  ( DIM_BERATER.BERATER=SECURITY_KNOTEN.KNOTEN
OR
DIM_BERATER.TEAMLEITER=SECURITY_KNOTEN.KNOTEN
OR
DIM_BERATER.BEREICHSLEITER=SECURITY_KNOTEN.KNOTEN
OR
SECURITY_KNOTEN.KNOTEN = 'ALLE'
  )
  AND  ( SECURITY_KNOTEN.ANWENDER=<aktueller Anwender>  )
/*******************************************************/
/* Ende der abgeleiteten Tabelle DIM_BERATER_SECURITY */
/*******************************************************/
```

Für den Anwender Feynman sieht die Ergebnistabelle wie folgt aus:

ANWENDER	BERATER_FK	BERATER	TEAM	TEAMLEITER	BEREICH	BEREICHSLEITER
Feynman	7	Planck	Quantenphysiker	Planck	Physiker	Newton
Feynman	8	Bohr	Quantenphysiker	Planck	Physiker	Newton
Feynman	9	Dirac	Quantenphysiker	Planck	Physiker	Newton
Feynman	10	Feynman	Quantenphysiker	Planck	Physiker	Newton

Der Grund dafür, dass diese Erweiterung korrekte Ergebnisse liefert, ist, dass der Primärschlüssel der Dimensionstabelle (BERATE_PK) bereits im SQL enthalten war. Alle weiteren Attribute derselben Tabelle führen damit nicht zu einer weiteren Satzaufspaltung, sondern nur zu einer Erweiterung der Informationen.

Diese abgeleitete Tabelle ersetzt nun die bisherige Dimension und die Securitytabelle, so dass die Struktur wie folgt aussieht:



5.2.5 *Gruppierung statt SELECT DISTINCT*

In den bisherigen abgeleiteten Tabellen wurde immer SELECT DISTINCT verwendet, um die Mehrfachzählungen aufzulösen und den Primärschlüssel der Dimension eindeutig dem Anwender zuzuweisen.

In SQL gibt es eine weitere Möglichkeit dafür: Die GROUP BY-Funktionalität. Diese kann aber nur im Zusammenhang mit einer Aggregatfunktion (Sum, count, Avg,...) verwendet werden.

Dafür können wir aber noch nicht die Faktentabelle verwenden:

Das Ziel ist nicht, in der abgeleiteten Tabelle bereits die korrekten Kennzahlen zu ermitteln, vielmehr soll zunächst die Dimension mit der Hierarchie eindeutig aufgelöst werden, um dann den korrekten Zugriff auf die Faktentabelle zu ermöglichen.

Würde in der abgeleiteten Tabelle bereits die (meist sehr großen!) Faktentabelle verwendet werden, so würde dies zu sehr langsamen Abfragen führen, ohne dass die ermittelten Kennzahlen verwendet werden könnten.

Aus der Dimension kann aber zunächst scheinbar keine Kennzahl ermittelt werden, die Frage ist also, wie dies umgangen werden kann.

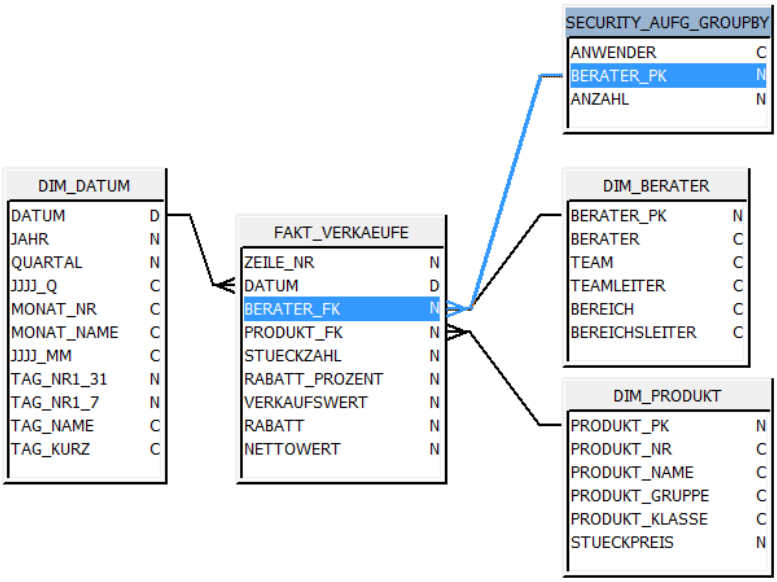
Ein sehr einfacher (und wirksamer) Trick ist, eine Kennzahl wie count(*) zu verwenden, was die beteiligten Sätze zählt und auch bereits mit den bereits im SQL vorhandenen Tabellen ausgeführt werden kann, etwa wie folgt (wir verwenden wieder die rudimentäre Abfrage mit Anwender und BERATER_PK, aber ohne weitere Attribute, die aber problemlos ergänzt werden können):

```
/*********************************************************/
/* Anfang der abgeleiteten Tabelle SECURITY_AUFG_GROUPBY */
/*********************************************************/

SELECT
  SECURITY_KNOTEN.ANWENDER,
  DIM_BERATER.BERATER_PK,
count(*) as ANZAHL
FROM
  SECURITY_KNOTEN,
  DIM_BERATER
WHERE
  ( DIM_BERATER.BERATER=SECURITY_KNOTEN.KNOTEN
OR
DIM_BERATER.TEAMLEITER=SECURITY_KNOTEN.KNOTEN
OR
DIM_BERATER.BEREICHSLEITER=SECURITY_KNOTEN.KNOTEN
OR
SECURITY_KNOTEN.KNOTEN = 'ALLE')
  AND
  SECURITY_KNOTEN.ANWENDER  =  <aktueller Anwender>
GROUP BY
  SECURITY_KNOTEN.ANWENDER,
  DIM_BERATER.BERATER_PK

/*********************************************************/
/* Ende der abgeleiteten Tabelle SECURITY_AUFG_GROUPBY */
/*********************************************************/
```

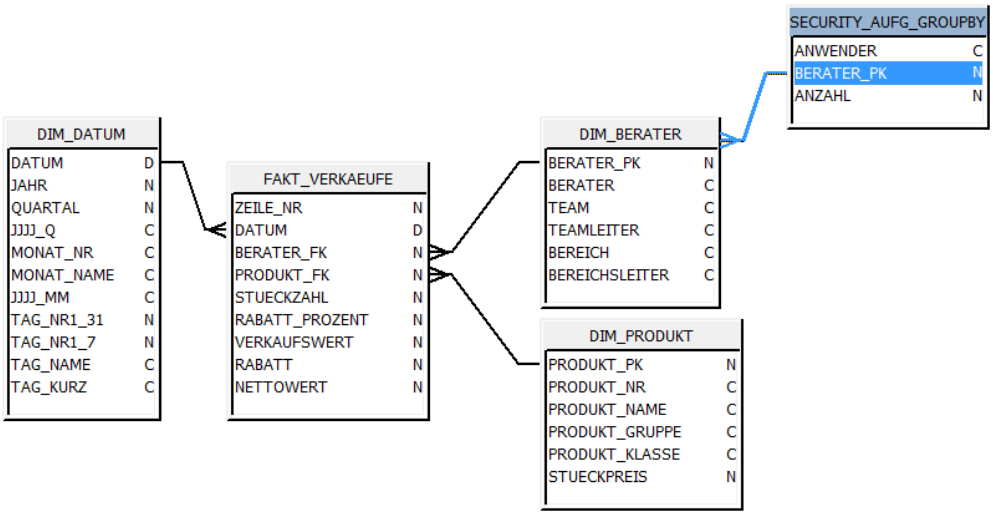
Für den Anwender Feynman ergibt sich folgendes Ergebnis:

ANWENDER	BERATER_PK	ANZAHL
Feynman	7	1
Feynman	8	1
Feynman	9	1
Feynman	10	2

Die Spalte ANZAHL zeigt sogar an, wie viele Sätze erzeugt würden, wenn keine Gruppierung stattfände, d.h. den Grad der Mehrfachzählung. Obgleich dies durchaus interessant sein kann, ist es für die praktische Umsetzung ohne inhaltliche Bedeutung, wir verknüpfen die neue abgeleitete Tabelle analog zur obigen:



oder alternativ



Die beiden Verfahren (DISTINCT/GROUP BY) sind im Hinblick auf die Erzeugung der Eindeutigkeit äquivalent, das GROUP BY-Verfahren wurde insbesondere im Hinblick auf Abschnitt 7, wo es zur einzigen Umsetzungsmöglichkeit werden wird, vorgestellt.

5.2.6 Betrachtungen zur Abfrageperformance

Wie schon in Abschnitt 5.1.4 erwähnt, können die Zuweisungen von Knoten statt Beratern zur Verschlechterung der Abfrageperformance führen. Dies gilt auch für die aufgelösten Strukturen, die zuvor vorgestellt wurden.

Wie schon erwähnt, werden wir in Abschnitt 12 genauer darauf eingehen, wie die Strukturen in dieser Hinsicht optimiert werden können.

Hierzu ist aber anzumerken, dass die *Logik* der Auflösung davon nicht betroffen ist, also korrekt bleibt. Die Performanceoptimierung geschieht ggf. durch Materialisierung der abgeleiteten Tabellen, was aber die Inhalte nicht ändert.

5.3 Berechtigung auf definierte Hierarchieebenen

5.3.1 Beschreibung der Lösung

In DIM_BERATER tauchen die Namen der Teamleiter sowohl im Feld TEAMLEITER, als auch im Feld BERATER (in ihrem eigenen Satz) auf, ebenso tauchen die Namen der Bereichsleiter in den Feldern BEREICHSLEITER, TEAMLEITER und BERATER (die letzten zwei im eigenen Beratersatz) auf. Am Ende von Abschnitt 5.1.3.1 wurde der folgende Fall geschildert:

Der Berater Galilei übernimmt während des Urlaubs die Vertretung für den Teamleiter Kopernikus, *aber nur für dessen eigene Verträge*. Galilei darf also nicht das gesamte Team Kopernikus sehen, sondern nur die des Beraters Kopernikus. Der Eintrag

ANWENDER	KNOTEN
Galilei	Kopernikus

würde aber das gesamte Team als Berechtigung zuweisen.

Mit der aktuellen Definition der Securitytabelle lässt sich dies nicht auflösen, weil die Knotenebene eine zusätzliche, noch nicht vorhandene Information ist.

Wir erweitern daher zunächst die Securitytabelle um eine weitere Spalte HIERARCHIEEBENE und erzeugen eine neue Tabelle mit dem Namen SECURITY_KNOTEN_HIER.

Angenommen, wir wollen im oberen Beispiel dem Anwender Galilei seine eigenen Daten und die des Beraters (!) Kopernikus zuweisen, so wären die folgenden beiden Einträge nötig:

ANWENDER	KNOTEN	HIERARCHIEEBENE
Galilei	Galilei	Berater
Galilei	Kopernikus	Berater

Soll hingegen der Teamleiter Kopernikus zusätzlich zu seinem Team auch noch die Daten des Beraters Planck und die des Bereichs Euklid sehen dürfen, so benötigt er die folgenden Einträge:

ANWENDER	KNOTEN	HIERARCHIEEBENE
Kopernikus	Kopernikus	Team
Kopernikus	Planck	Berater
Kopernikus	Euklid	Bereich

Es sei angemerkt, dass nun für alle Sätze, also auch die „eigenen" Sätze jedes Anwenders, die Hierarchieebene eingetragen werden muss, wie oben in den Sätzen Galilei/ Galilei/**Berater** und Kopernikus/ Kopernikus/**Team.**

Die Information muss nun im Join zwischen der Securitytabelle und DIM_BERATER ergänzt werden.

Betrachten wir zunächst die Einträge für Berater:

Diese sollen berücksichtigt werden, wenn in SECURITY_KNOTEN_HIER.HIERARCHIEEBENE der Wert „Berater" steht. Als Gesamtbedingung ergibt dies

```
(SECURITY_KNOTEN_HIER.KNOTEN = DIM_BERATER.BERATER
AND
SECURITY_KNOTEN_HIER.HIERARCHIEEBENE = 'Berater')
```

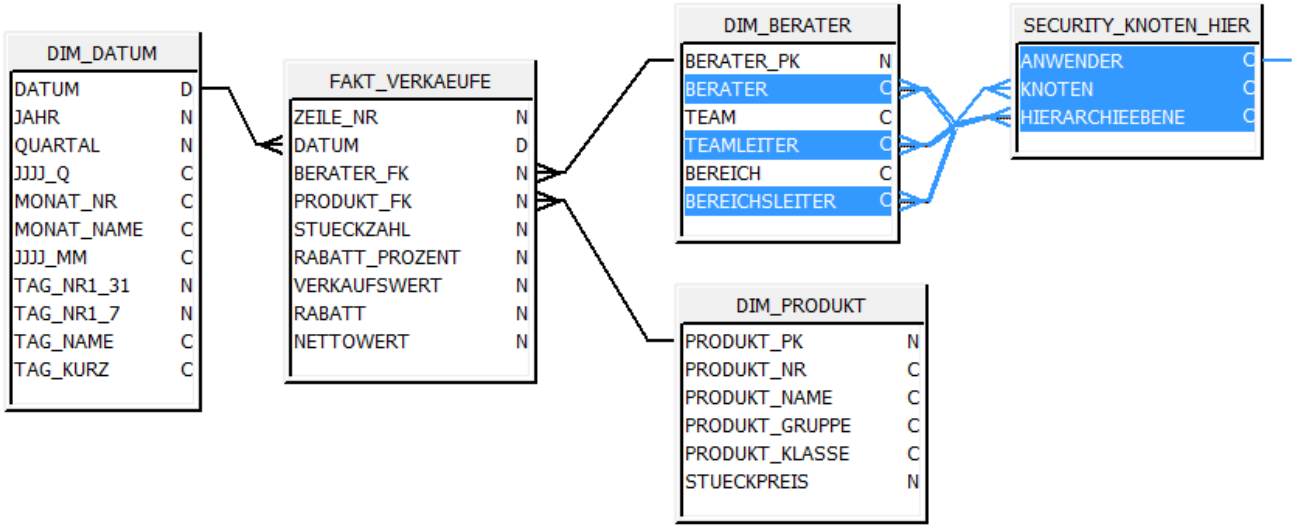
Analoge Bedingungen ergeben sich für die Teams und die Bereiche. Diese drei Einzelbedingungen müssen wieder durch OR verknüpft werden.

Schließlich muss als vierte Bedingung wieder die Bedingung für Anwender mit dem Recht „ALLE“ ergänzt werden, woraus sich folgende Joinbedingung ergibt:

```
(
(SECURITY_KNOTEN_HIER.KNOTEN = DIM_BERATER.BERATER
AND
SECURITY_KNOTEN_HIER.HIERARCHIEEBENE = 'Berater')
OR
(SECURITY_KNOTEN_HIER.KNOTEN = DIM_BERATER.TEAMLEITER
AND
SECURITY_KNOTEN_HIER.HIERARCHIEEBENE = 'Team')
OR
(SECURITY_KNOTEN_HIER.KNOTEN = DIM_BERATER.BEREICHSLEITER
AND
SECURITY_KNOTEN_HIER.HIERARCHIEEBENE = 'Bereich')
OR
(SECURITY_KNOTEN_HIER.KNOTEN = 'ALLE')
)
```

Schließlich muss die Einschränkung auf den aktuellen Anwender wieder ergänzt werden.

Die finale Tabellenstruktur sieht wie folgt aus:



Sie ersetzt die Struktur aus Abschnitt 5.1.1. Es sei aber angemerkt, dass dies nur notwendig ist, wenn aus fachlicher Sicht die Übergabe der Hierarchieebene benötigt wird, ansonsten ist die in 5.1.1. beschriebene, sehr viel einfachere Struktur zu bevorzugen!

5.3.2 Auflösung von Überlappungen

Ein Missverständnis könnte sein, dass durch die erweiterte Information bei der Rechtezuweisung automatisch auch die Mehrfachzählungen durch Überlappungen (siehe vorheriger Abschnitt 5.2) vermieden werden, dies ist aber nicht der Fall, was an einem einfachen Beispiel zu zeigen ist:

Wir gehen wieder wie oben davon aus, dass dem Anwender Feynman seine eigenen Daten und die des Bereichs (also nicht nur des Anwenders!) Planck, zu dem er selber gehört, zugewiesen werden. In der Securitytabelle sind dafür folgende Einträge notwendig:

ANWENDER	KNOTEN	HIERARCHIEEBENE
Feynman	Feynman	Berater
Feynman	Planck	Team

Dies führt zu genau derselben Überlappung wie in Abschnitt 5.2, sie ist sozusagen nur präziser definiert.

Wie kann diese Überlappung aufgelöst werden?

Interessanterweise ist das Verfahren genau dasselbe, wie in Abschnitt 5.2 beschrieben, lediglich die Joinbedingungen müssen entsprechend angepasst werden!

Beispielsweise wurde in Abschnitt 0 die grundlegende abgeleitete Tabelle vorgestellt; in der neuen Version sieht diese wie folgt aus (die neue Joinbedingung ist fett gedruckt):

```
/*******************************************************/
/* Anfang der abgeleiteten Tabelle SECURITY_AUFGELOEST */
/*******************************************************/
SELECT DISTINCT
  SECURITY_KNOTEN.ANWENDER,
  DIM_BERATER.BERATER_PK
FROM
  SECURITY_KNOTEN,
  DIM_BERATER
WHERE
  (
(SECURITY_KNOTEN_HIER.KNOTEN = DIM_BERATER.BERATER
AND
SECURITY_KNOTEN_HIER.HIERARCHIEEBENE = 'Berater')
OR
(SECURITY_KNOTEN_HIER.KNOTEN = DIM_BERATER.TEAMLEITER
AND
SECURITY_KNOTEN_HIER.HIERARCHIEEBENE = 'Team')
OR
(SECURITY_KNOTEN_HIER.KNOTEN = DIM_BERATER.BEREICHSLEITER
AND
SECURITY_KNOTEN_HIER.HIERARCHIEEBENE = 'Bereich')
OR
(SECURITY_KNOTEN_HIER.KNOTEN = 'ALLE')
)
  AND
  SECURITY_KNOTEN.ANWENDER  =  <aktueller Anwender>
/*******************************************************/
/* Ende der abgeleiteten Tabelle SECURITY_AUFGELOEST */
/*******************************************************/
```

Alle anderen in Abschnitt 5.2 beschriebenen Schritte, also die Integration in das Datenmodell, die Ergänzung von Dimensionsdaten oder die Verwendung von GROUP BY statt SELECT DISTINCT, sind identisch.

5.4 Schwierige und pathologische Fälle

Nicht alle Strukturen in einem DWH sind einfach zu verwenden, geschweige denn mit Dateneinschränkungen zu belegen. Wir gehen jetzt auf einige davon ein:

5.4.1 Nicht aufgelöste Parent/Child-Beziehungen

Parent/Child-Tabellen werden in vielen ERP-Systemen verwendet, um Hierarchien abzubilden, dafür gibt es mehrere gute Gründe, die mit der Dynamik von Hierarchien zusammenhängen:

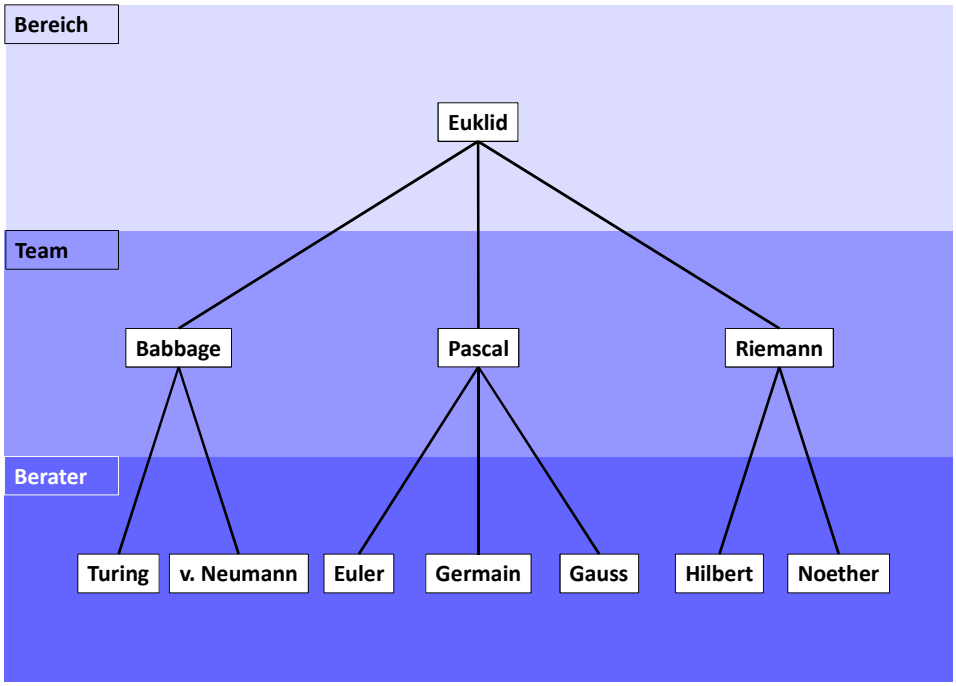
- Wird eine neue Hierarchieebene eingeführt, so werden einfach die entsprechenden Einträge gepflegt.
- Nicht alle Hierarchiepfade müssen gleich lang sein, es werden jeweils die benötigten Pfade erzeugt.

Im DWH-Umfeld sind Parent/Child-Tabellen aber oft nicht gerne gesehen (obwohl manche Werkzeuge sich auf die performante Auflösung solcher Strukturen auch im Berichtswesen spezialisiert haben!):

- Für das Berichtswesen muss immer zunächst eine Auflösung erfolgen. Da Parent/Child-Beziehungen aber rekursiv sind, benötigt dies entweder eine entsprechende Logik in Form von Programmen oder komplexe View-Definitionen.
- Die präzise Zuordnung von fachlichen Bedeutungen für die Hierarchieebenen kann sich gerade aufgrund der Dynamik schwierig gestalten.
- Die korrekte Historisierung von SCD2-Dimensionen kann sehr komplex werden.[11]

Aus diesem Grund werden Parent/Child-Tabellen meistens schon im ETL-Prozess vollständig zu statischen Hierarchien aufgelöst.

Warum aber ist der Umgang mit Parent/Child-Tabellen für die Sicherheitseinschränkungen schwierig? Betrachten wir dazu einen Teil unserer Hierarchie, nämlich den Bereich Euklid:



Als Parent/Child-Tabelle (ab sofort BERATER_PCH genannt) würde dies wie folgt abgebildet werden:

PARENT	CHILD
-	Euklid
Euklid	Babbage
Euklid	Pascal
Euklid	Riemann

[11] Darauf können wir hier nicht näher eingehen, es hat aber damit zu tun, dass die verschiedenen Hierarchieebenen nicht synchron geändert werden müssen und daher eine korrekte zeitliche Zuordnung der verschiedenen Ebenen sehr komplex werden kann.

Babbage	Turing
Babbage	v. Neumann
Pascal	Euler
Pascal	Germain
Pascal	Gauss
Riemann	Hilbert
Riemann	Noether

Wenn eine Securitytabelle mit Zuweisung auf der Ebene Berater (siehe Abschnitt 4) verwendet wird, so ist dies in diesem Fall tatsächlich einfach möglich! Die Spalte BERATER_PCH.CHILD enthält **alle** Mitglieder der Hierarchie und damit alle Berater, und dies eindeutig. Die Spalte ist somit der (vollständige) Primärschlüssel der Tabelle und kann daher mit der Spalte BERATER der Securitytabelle verknüpft werden.

Schwieriger wird die Verwendung einer Securitytabelle mit Knotenzuweisungen.

Für Zuweisungen auf Ebene Berater ist dies noch einfach: Diese können nur im Feld CHILD auftauchen, was eine direkte Verknüpfung mit der Securitytabelle über dieses Feld erlaubt.

Nun weisen wir dem Teamleiter Riemann seinen Knoten zu.

ANWENDER	**KNOTEN**
Riemann	Riemann

In der Tabelle BERATER_PCH sind damit folgende Sätze relevant:

PARENT	**CHILD**
Euklid	Riemann
Riemann	Hilbert
Riemann	Noether

Hier zeigt sich die erste Problematik: Es sind alle Sätze zu berücksichtigen, in denen Riemann als Parent oder Child vorkommt, was sich aber noch einfach über die folgende Joinbedingung lösen lässt:

```
SECURITY_KNOTEN.KNOTEN = BERATER_PCH.PARENT
OR
SECURITY_KNOTEN.KNOTEN = BERATER_PCH.CHILD
```

Nun werden dem Bereichsleiter Euklid seine Rechte zugewiesen:

ANWENDER	**KNOTEN**
Euklid	Euklid

In BERATER_PCH taucht der Eintrag Euklid aber nur in den folgenden Sätzen auf:

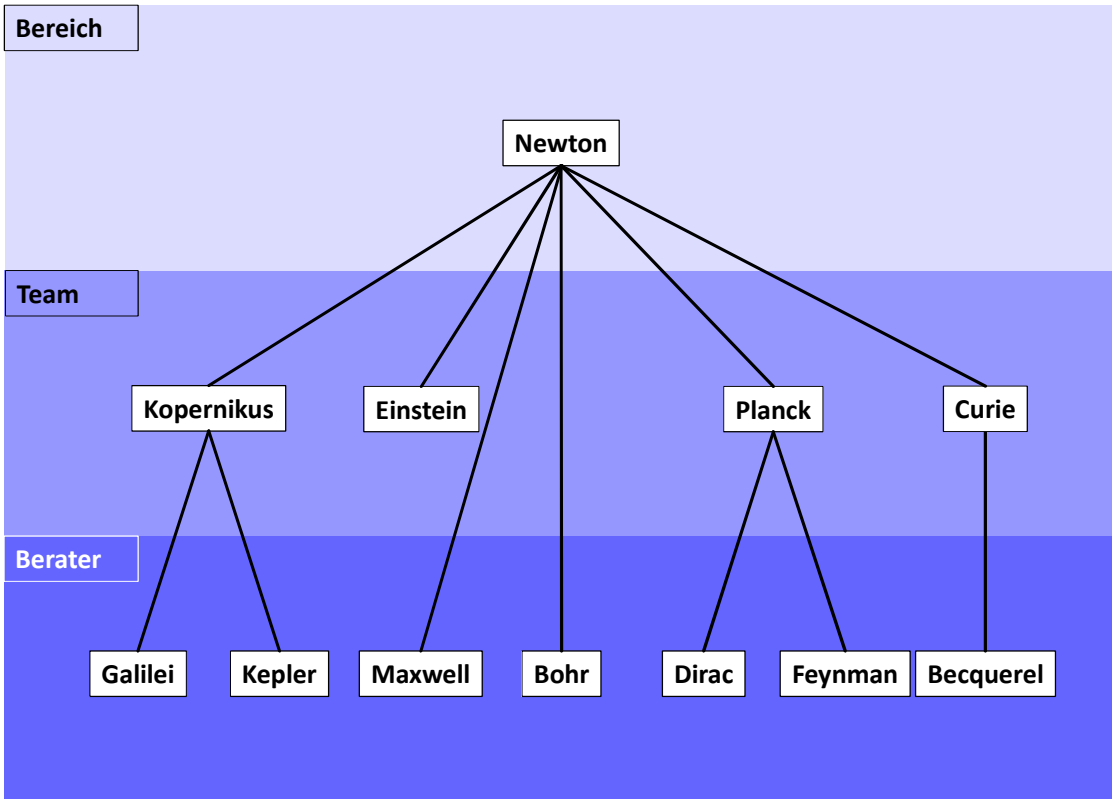
PARENT	**CHILD**
-	Euklid
Euklid	Babbage
Euklid	Pascal
Euklid	Riemann

Um somit alle zum Bereich zugehörigen Berater zu erfassen, muss die Hierarchie zunächst aufgelöst werden! Dasselbe gilt natürlich auch für mehr als dreistufige Hierarchien in den höheren Ebenen.

5.4.2 *Hierarchien mit Lücken und unterschiedlichen Pfadlängen*

Nicht alle Dimensionstabellen haben vollständig ausbalancierte Hierarchien, in denen also alle Einträge dieselbe Hierarchie-Pfadlänge haben und bei denen alle Zwischenebenen gefüllt sind. Es sei darauf hingewiesen, dass das genaue Design der Dimensionstabelle nicht immer beeinflusst werden kann. Hier soll die Frage erörtert werden, wie sich solche Designs auf die Sicherheitseinschränkungen auswirken.

Wir gehen davon aus, dass die Hierarchie vollständig aufgelöst ist[12]. In einem neuen Beispiel gestalten wir die Physikerhierarchie wie folgt um:



Der Teamleiter Einstein ist zwar noch Teamleiter, hat aber keine Mitarbeiter mehr. Die Berater Maxwell und Bohr sind nun als Berater direkt Newton unterstellt.

In der Dimensionstabelle werden die Informationen zu Team- und Bereichsleitern nicht mehr nach unten wiederholt, also steht beispielsweise Newton nur noch als Bereichsleiter, aber nicht mehr als Teamleiter oder Berater eingetragen.

Der künstliche Schlüssel als Zeilenindikator und Primärschlüssel bleibt erhalten.

Damit erhalten wir die folgenden Einträge in der Dimension:

BERATER_PK	BERATER	TEAM	TEAMLEITER	BEREICH	BEREICHSLEITER
1				Physiker	Newton
2		Klassische Physiker	Kopernikus	Physiker	Newton
3	Kepler	Klassische Physiker	Kopernikus	Physiker	Newton
4	Galilei	Klassische Physiker	Kopernikus	Physiker	Newton
5		Elektrodynamiker	Einstein	Physiker	Newton

[12] also nicht als Parent/Child-Tabelle vorliegt

6	Maxwell			Physiker	Newton
7		Quantenphysiker	Planck	Physiker	Newton
8	Bohr			Physiker	Newton
9	Dirac	Quantenphysiker	Planck	Physiker	Newton
10	Feynman	Quantenphysiker	Planck	Physiker	Newton
11		Radioaktivitätsforscher	Curie	Physiker	Newton
12	Bequerel	Radioaktivitätsforscher	Curie	Physiker	Newton

Betrachten wir zunächst wieder die Securitytabelle aus Abschnitt 4.1, also Anwender + Berater.

Diese kann hier nicht mehr sinnvoll verwendet werden, weil die Teamleiter und Bereichsleiter nicht mehr als Berater auftauchen, daher ist diese Referenz nicht mehr vollständig.

Zwei mögliche Modifikationen könnten dies korrigieren:

- Die Zuweisung erfolgt nicht auf den Berater, sondern auf den Primärschlüssel BERATER_PK. Dies ist allerdings schwer zu realisieren, weil dieser Schlüssel ein künstlicher Schlüssel im DWH ist und deshalb keine direkte fachliche Bedeutung hat.
- In der Dimension kann eine weitere Spalte als reine Referenz ergänzt werden, die für jede Zeile die unterste besetzte Hierarchieebene wiederholt. Diese ist allerdings gleichwertig mit der Befüllung des Beraternamens für jeden Satz mit genau dieser Information! Dadurch kann natürlich auch wieder die Einschränkung auf den Berater erfolgen.

Als zweite Alternative verwenden wir wieder die Securitytabelle mit Knoteneinschränkungen wie in Abschnitt 5.1. Interessanterweise lässt sich diese unverändert verwenden, da ja in mindestens einem Attribut des Dimensionssatzes die notwendige Zuordnung (Team, Bereich) erfolgt, anders ließe sich die Hierarchie auch nicht im Bericht verwenden!

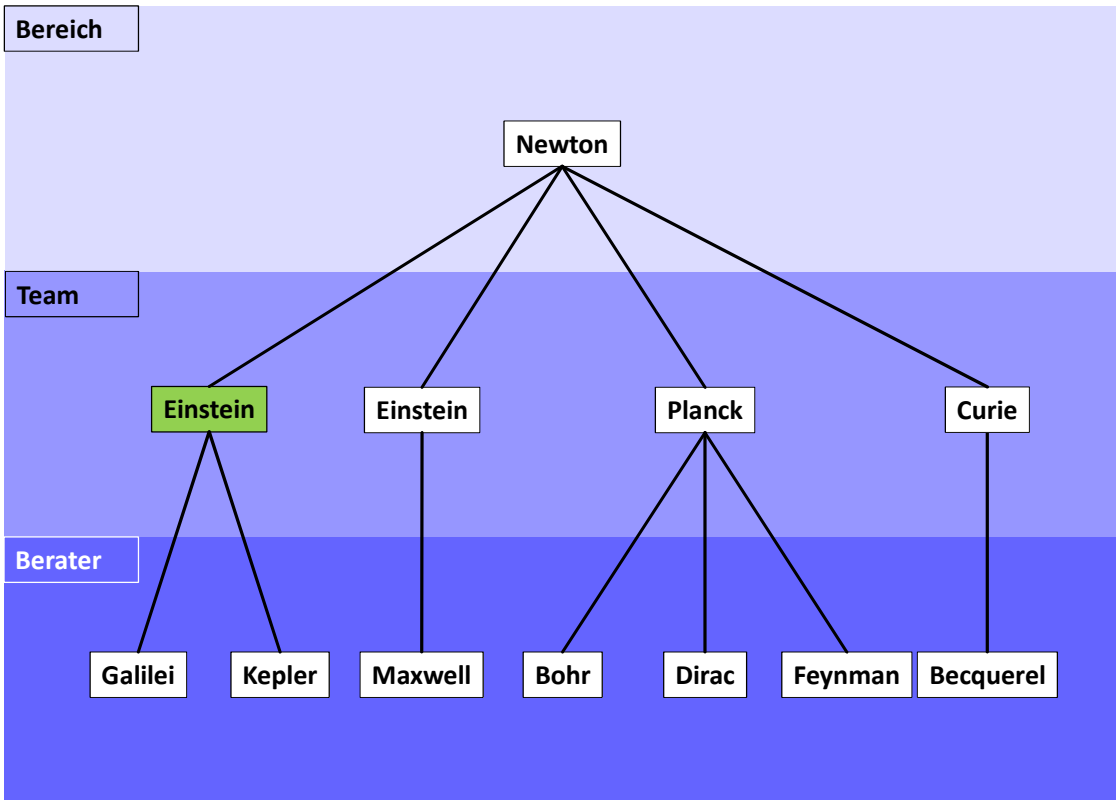
Die in Abschnitt 5.3 beschriebene Einschränkung auf definierte Hierarchieebenen kann wieder nur korrekte Ergebnisse erzielen, wenn die entsprechende Information auf der benötigten Ebene gepflegt ist. Eine Einschränkung auf den Berater (also nicht den Bereichsleiter!) Newton könnte so also wieder nicht erfolgen.

Generell lassen sich alle geschilderten Problematiken beseitigen, wenn die Hierarchien komplett aufgefüllt werden, wie oben beschrieben. Dies ist also die eigentliche Empfehlung für das Securitydesign, zumal dies auch im Berichtswesen häufig besser zu verwenden ist.

5.4.3 Gleichnamige Knoten in unterschiedlichen Hierarchiezweigen

Ein sehr pathologischer Fall ist die Verwendung derselben Knotennamen in unterschiedlichen Hierarchiezweigen.

Betrachten wir dazu wieder die ursprüngliche Beraterhierarchie der Physiker. Es hat sich aber eine Änderung ergeben: Kopernikus ist in Ruhestand gegangen, Einstein hat dieses Team, aber als getrenntes Team übernommen:



Der Anwender Einstein wird in der Security mit Knoten sich selber zugewiesen und sieht beide Teams, was korrekt ist.

Wird nun aber Kepler als Vertretung für das ursprüngliche Team Kopernikus (im Bild grün markiert) eingesetzt, so ist dies in der Security über den Teamleiter Einstein nicht möglich: Darüber würde auch das andere Team sichtbar werden.

Eine Lösung für das Problem wäre hier, über ein für das Team eindeutiges Attribut einzuschränken, beispielsweise den Teamnamen (Klassische Physiker/Elektrodynamiker).

Ist eine solche eindeutige Zuordnung nicht möglich, so lässt sich dieser Wiederspruch nicht auflösen!

6 Berechtigungen auf Rollen statt Anwender

Bisher wurden alle Einschränkungen in der Securitytabelle auf die Namen der entsprechenden Berater, Team- und Bereichsleiter vorgenommen. Dies ist aber nicht notwendigerweise immer zielführend:

- Team- und Bereichsleitungen können wechseln, auch ohne dass damit eine Umstrukturierung verbunden sein muss. In diesem Fall müssten alle Berechtigungen, die sich auf den entsprechenden Namen beziehen, auf den neuen geändert werden, obwohl sich inhaltlich keine Änderung ergeben hat.
- Häufig sind Betreuungen nicht an Personen, sondern an Rollen gebunden. Ein Beispiel aus der Bankenwirtschaft: Ein Kunde wird in einer Filiale betreut. Wenn nun der bisherige Berater, der den Kunden betreut, in ein anderes Land umzieht, wird der Kunde weiter von derselben Filiale betreut. Die Berechtigung hat sich also nicht geändert, lediglich die Personen, die die Berechtigungen brauchen.
- Dieselben Berechtigungen können für verschiedene Personen gültig sein, etwa wie zuvor bei der Vertragsbetreuung: Ale Mitarbeiter einer Filiale benötigen Zugriff auf die in dieser Filiale betreuten Verträge.

Es gibt durchaus personenbezogene Security (wie oben beschrieben), etwa Provisionen an Abschlüssen o.ä., aber eben nicht immer.

Solche rollenspezifische Berechtigungen werden über zwei Securitytabellen definiert:

- In der ersten werden die Rollen ihren Berechtigungen zugewiesen.
- In der zweiten erfolgt die Zuordnung der Anwender zu den Rollen.

Zuvor wird die Dimension um weitere Felder ergänzt, die eine Codierung der Stelle auf den drei Ebenen enthält[13], diese Attribute verwenden wir später für die Einschränkungen:

BERATER_PK	BERATER_STELLE	BERATER	TEAM	TEAM_STELLE	TEAM-LEITER	BEREICH_STELLE	BEREICH	BEREICHS-LEITER
1	P	Newton	Leitung Physiker	P	Newton	P	Physiker	Newton
2	P_1	Kopernikus	Klassische Physiker	P_1	Kopernikus	P	Physiker	Newton
3	P_1_1	Kepler	Klassische Physiker	P_1	Kopernikus	P	Physiker	Newton
4	P_1_2	Galilei	Klassische Physiker	P_1	Kopernikus	P	Physiker	Newton
5	P_2	Einstein	Elektro-dynamiker	P_2	Einstein	P	Physiker	Newton
6	P_2_1	Maxwell	Elektro-dynamiker	P_2	Einstein	P	Physiker	Newton
7	P_3	Planck	Quanten-physiker	P_3	Planck	P	Physiker	Newton
8	P_3_1	Bohr	Quanten-physiker	P_3	Planck	P	Physiker	Newton
9	P_3_2	Dirac	Quanten-physiker	P_3	Planck	P	Physiker	Newton
10	P_3_3	Feynman	Quanten-physiker	P_3	Planck	P	Physiker	Newton
11	P_4	Curie	Radioaktivitäts-forscher	P_4	Curie	P	Physiker	Newton

[13] Genau genommen hätten für die Teams und Bereiche auch die beschreibenden Felder genommen werden können; die Verwendung von langen VARCHAR-Feldern für die Einschränkungen ist aber grundsätzlich nicht optimal. Die Ergänzung der Berater-Stelle war aber in jedem Fall nötig, weil bisher nur der Beratername vorhanden war.

12	P_4_1	Bequerel	Radioaktivitäts-forscher	P_4	Curie	P	Physiker	Newton
13	M	Euklid	Leitung Mathematiker	M	Euklid	M	Mathematiker	Euklid
14	M_1	Pascal	Mathematiker Frühe Neuzeit	M_1	Pascal	M	Mathematiker	Euklid
15	M_1_1	Euler	Mathematiker Frühe Neuzeit	M_1	Pascal	M	Mathematiker	Euklid
16	M_1_2	Germain	Mathematiker Frühe Neuzeit	M_1	Pascal	M	Mathematiker	Euklid
17	M_1_3	Gauss	Mathematiker Frühe Neuzeit	M_1	Pascal	M	Mathematiker	Euklid
18	M_2	Riemann	Mathematiker Neuzeit	M_2	Riemann	M	Mathematiker	Euklid
19	M_2_1	Noether	Mathematiker Neuzeit	M_2	Riemann	M	Mathematiker	Euklid
20	M_2_2	Hilbert	Mathematiker Neuzeit	M_2	Riemann	M	Mathematiker	Euklid
21	M_3	Babbage	Informatiker	M_3	Babbage	M	Mathematiker	Euklid
22	M_3_1	v. Neumann	Informatiker	M_3	Babbage	M	Mathematiker	Euklid
23	M_3_2	Turing	Informatiker	M_3	Babbage	M	Mathematiker	Euklid

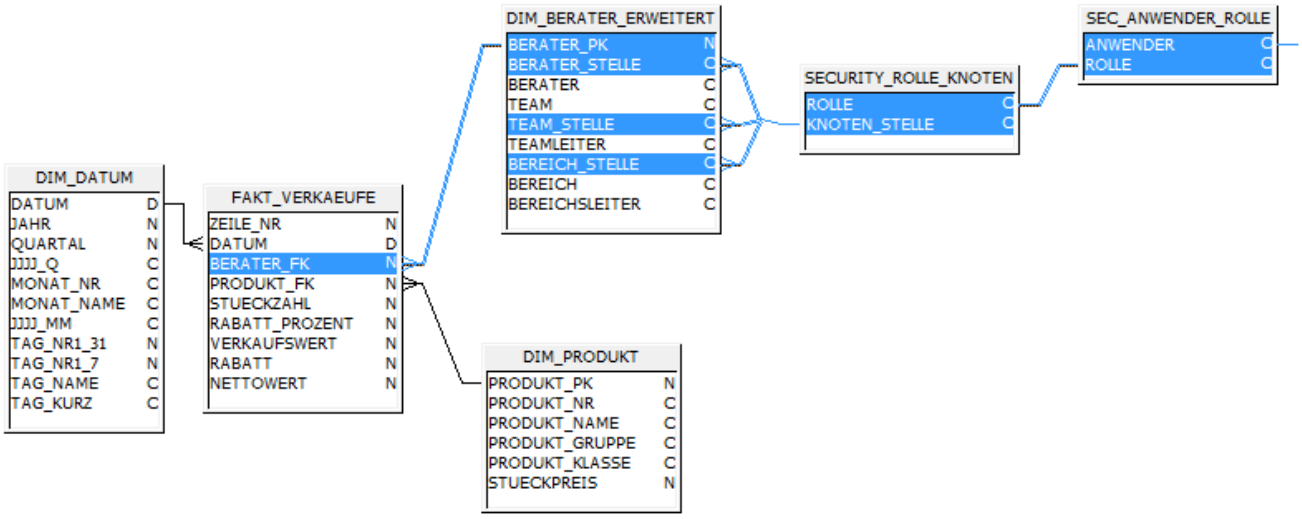
Die erste Securitytabelle enthält die Rollen mit ihren Berechtigungen. Wir verwenden eine Securitytabelle mit Knotenzuweisung (also analog zu Abschnitt 5.1), wir können also alle Hierarchieebenen für Berechtigungen verwenden.
Im einfachsten Fall sind die Rollen nach den Stellen bezeichnet und werden sich selber zugeordnet, zusätzlich tragen wir die Rolle Gesamt mit der Berechtigung „ALLE“ ein:

ROLLE	**KNOTEN_STELLE**
P	P
P_1	P_1
P_1_1	P_1_1
Gesamt	ALLE
...	...

In einer zweiten Tabelle werden die Anwender den Rollen zugewiesen:

ANWENDER	**ROLLE**
Newton	P
Kopernikus	P_1
Kepler	P_1_1
Vorstand	Gesamt
...	...

Wir erhalten folgende Tabellenstruktur:



Die Joinbedingung zwischen der erweiterten Beraterdimension und der ersten Securitytabelle lautet jetzt wie folgt:

```
SECURITY_ROLLE_KNOTEN.KNOTEN_STELLE = DIM_BERATER_ERWEITERT.BERATER_STELLE
OR
SECURITY_ROLLE_KNOTEN.KNOTEN_STELLE = DIM_BERATER_ERWEITERT.TEAM_STELLE
OR
SECURITY_ROLLE_KNOTEN.KNOTEN_STELLE = DIM_BERATER_ERWEITERT.BEREICH_STELLE
OR
SECURITY_ROLLE_KNOTEN.KNOTEN_STELLE = 'ALLE'
```

Weiterhin ist die zweite Securitytabelle auf den aktuellen Anwender eingeschränkt:

```
SEC_ANWENDER_ROLLE.ANWENDER=<aktueller Anwender>
```

Im bisherigen Zustand wirkt diese Struktur noch übertrieben, weil die erste Securitytabelle bis auf die neue Rolle Gesamt nur eine 1:1-Umschlüsselung (wobei sogar die Namen identisch bleiben!) bewirkt und daher zunächst keinen Mehrwert bringt.

Dies ändert sich, sobald die Rollen komplexer definiert sind und dann auch mehreren Anwendern zugewiesen werden.

Als Beispiel sollen alle Berater auch die Daten der Kollegen im eigenen Team sehen dürfen, aber nicht die des Teamleiters, analog sollen die Teamleiter die Daten der anderen Teams sehen dürfen, aber nicht die des Bereichsleiters.

Für die Einschränkung der Teammitarbeiter kann nicht das Team verwendet werden, da dieses auch den Teamleiter beinhaltet. Für das Team Kopernikus mit dem Stellencode P1 können wir daher folgende Rolle BER_P1 (Berater P1) definieren:

ROLLE	KNOTEN_STELLE
MA_P1	P_1_1
MA_P1	P_1_2

Diese Rolle kann nun den beiden Anwendern Kepler und Galilei zugewiesen werden:

ANWENDER	ROLLE
Kepler	MA_P1
Galilei	MA_P1

Für die Teamleiter der beiden Bereiche werden folgende Rollen definiert:

ROLLE	KNOTEN_STELLE
TL_P	P_1
TL_P	P_2
TL_P	P_3
TL_P	P_4
TL_M	M_1
TL_M	M_2
TL_M	M_3

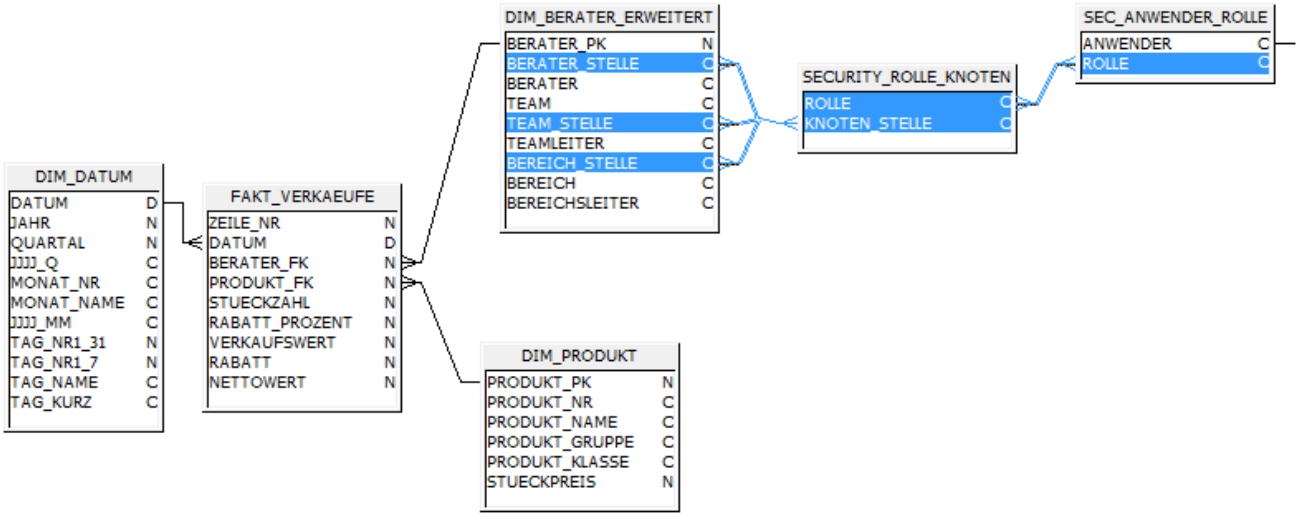
Die Zuweisung der Anwender zu den Rollen ist die folgende:

ANWENDER	ROLLE
Kopernikus	TL_P
Einstein	TL_P
Planck	TL_P
Curie	TL_P
Pascal	TL_M
Riemann	TL_M
Babbage	TL_M

Außerdem können auch wieder weitere Anwender definiert werden, die sogar mehrere Rollen erhalten können:

ANWENDER	ROLLE
Darwin	MA_P1
Darwin	TL_M

Die Joinkardinalitäten ändern sich nun beide zu n:m: Jeder Knoten kann in mehreren Rollen vorkommen, jede Rolle aus mehreren Knoten bestehen. Ebenso kann dieselbe Rolle mehreren Anwendern zugewiesen werden und ein Anwender mehrere Rollen haben:



ACHTUNG!

Die Gefahr von Überlappungen ist hier größer als zuvor, da sie teilweise in den Rollendefinitionen „versteckt" ist. Es ist manchmal nicht mehr unmittelbar sichtbar, dass einem Anwender über seine verschiedenen Rollen bestimmte Dimensionseinträge mehrfach zugewiesen wurden.

Die Empfehlung ist also, wie in Abschnitt 5.2.3 die Überlappungen eindeutig nach ANWENDER/ BERATER_PK aufzulösen.

7 Steuerung der Sichtbarkeit von Detailinformationen

7.1 Was ist damit gemeint?

In den bisherigen Erläuterungen war immer die Annahme, dass ein Anwender, der eine bestimmte Information sehen darf, diese vollständig sehen darf. Hat also beispielsweise der Teamleiter Planck Zugriff auf die Daten der Berater in seinem Team, so darf er auch alle Details der von ihnen abgeschlossenen Verkäufe sehen.

Dies muss aber nicht notwendigerweise der Fall sein: Beispielsweise könnte der Teamleiter zwar die Gesamtsummen je Mitarbeiter sehen dürfen, nicht aber die Details der Einzelsätze.

Auch der Zugriff auf bestimmte Kennzahlen könnte reglementiert sein: Der Verkaufswert darf gesehen werden, die Rabattinformationen und die Nettowerte aber nicht.

Andererseits könnten die Berater Zugriff auf die Gesamtsummen ihres Teams und Bereichs bekommen, um sich im Vergleich zum Rest des Teams sehen zu können, sie dürfen aber nur die eigenen Daten im Detail sehen.

Schließlich könnten die Bereichsleiter die Summen je Mitarbeiter im eigenen Bereich sehen dürfen, die anderen Bereiche aber nur in Summe je Bereich, so dass der eigene Bereich im Vergleich zu den anderen gesehen werden kann.

Dies kann nicht allein durch Zuweisung des Knotens abgebildet werden, wir müssen die Securitytabelle um das sichtbare Level erweitern.

Wir werden im weiteren Verlauf wieder von der Securitytabelle mit direkter Knotenzuweisung aus, also nicht nur auf Beraterebene und auch nicht über Rollen, wie in Abschnitt 6 beschrieben. Wir gehen auch wieder von der nicht-erweiterten Beraterdimension aus.

7.2 Fachliche Anforderungen für das Beispiel

Als erstes definieren wir die fachlichen Anforderungen für das Beispiel.

- Wenn ein Berater sich anmeldet, so darf er seine eigenen Daten in allen Details sehen (also auch alle Faktensätze). Er darf den Rest des Teams in Summe sehen, ebenso den Rest des Bereichs als Summe. Die einzige erlaubte Kennzahl ist in diesem Fall aber der VERKAUFSWERT, alle anderen Kennzahlen müssen leer bleiben.
- Ein Teamleiter darf die Daten des eigenen Teams bis auf den Berater genau inklusive aller Faktendetails sehen: Er hat ja die Aufgabe, das Team aktiv zu steuern. Von den anderen Teams in seinem Bereich darf er die Summen des Verkaufswerts je Team sehen, aber keine Information bis auf den Berater.
- Ein Bereichsleiter darf alle Daten des eigenen Bereichs bis auf den Berater genau sehen, ebenso den Verkaufswert und den Nettowert, aber nicht die anderen Faktendetails. Für die anderen Bereiche darf er die Summe des Verkaufswerts sehen.
- Ein Vorstand hat dieselben Rechte wie der Bereichsleiter, aber für alle Bereiche der Firma.
- Ein zentraler Controller darf alle Daten aller Berater bis ins Detail sehen.

Wir können insgesamt vier folgenden Stufen unterscheiden, wobei die Stufen von 1 bis 4 durchnummeriert wurden. Es wäre auch andersherum möglich (die Relevanz wird später sichtbar), es sollten aber besser keine Buchstaben verwendet werden.

Die vier Stufen sind[14]:

Stufe	was ist sichtbar	was ist nicht sichtbar
1	alle Details, auch alle Faktendetails	-
2	- alle Details der Beraterdimension bis hinunter auf den Berater - alle Dimensionsinformationen aus den anderen Dimensionstabellen - der Verkaufswert und der Nettowert aus den Fakten	- alle anderen Fakteninformationen (Satznummer, Stückzahl, Rabatt-Prozent und Rabatt-Wert)
3	- alle Details der Beraterdimension bis hinunter auf das Team - alle Dimensionsinformationen aus den anderen Dimensionstabellen - der Verkaufswert aus den Fakten	- der Berater aus der Beraterdimension - alle anderen Fakteninformationen (Satznummer, Stückzahl, Rabatt-Prozent und Rabatt-Wert, Nettowert
4	- die Bereichsinformationen aus der Beraterdimension - alle Dimensionsinformationen aus den anderen Dimensionstabellen - der Verkaufswert aus den Fakten	- Berater und Teaminformationen aus der Beraterdimension - alle anderen Fakteninformationen (Satznummer, Rabatte, Nettowert

7.3 Erweiterung der Securitytabelle

Ausgangspunkt ist die Securitytabelle mit Knotenzuweisungen.

Wir beginnen mit den Zuweisungen für den Anwender **Galilei**, der **Berater** im Team Kopernikus ist: Er benötigt seinen eigenen Knoten (Berater Galilei), ebenso das Team Kopernikus und den Bereich Physiker. Diese drei Knoten haben aber unterschiedliche Detailrechte: Den eigenen Knoten darf er mit allen Details, also mit Stufe 1 sehen, das Team Kopernikus in Stufe 3 und den Bereich Newton in Stufe 4.

Die Securitytabelle wird um die Stufe erweitert und erhält somit die folgenden Einträge:

ANWENDER	KNOTEN	STUFE
Galilei	Galilei	1
Galilei	Kopernikus	3
Galilei	Newton	4

Nach den obigen Anforderungen benötigt der **Teamleiter Kopernikus** im eigenen Team ebenfalls alle Detailrechte (1) und für den Rest des Bereichs Newton die Rechte, die Teams zu sehen, also 3:

ANWENDER	KNOTEN	STUFE
Kopernikus	Kopernikus	1
Kopernikus	Newton	3

[14] Dies ist selbstverständlich nur ein Beispiel. Eine andere Möglichkeit wäre, ab Stufe 2 kein Datum, sondern nur noch die monatliche Information, ebenso nicht mehr die Produkt-, sondern nur die Produktgruppe und -klasse zuzuweisen. Die genauen fachlichen Anforderungen müssen vor dem konkreten Design genau definiert werden.

Der **Bereichsleiter Newton** darf in seinem eigenen Bereich alle Berater mit den eingeschränkten Fakteninformationen sehen, also in Stufe 2. Die anderen Bereiche darf er bis auf die Bereichssummen genau sehen, also mit Stufe 4 (bitte beachten: es gibt nur einen anderen Bereich):

ANWENDER	KNOTEN	STUFE
Newton	Newton	2
Newton	Euklid	4

Da Newton alle anderen Bereiche der Firma sehen darf, kann er alternativ auch die folgenden Einträge erhalten:

ANWENDER	KNOTEN	STUFE
Newton	Newton	2
Newton	**ALLE**	4

Der **Vorstand** darf alle Daten der Firma bis auf den Berater genau mit eingeschränkten Faktenrechten sehen, also mit Stufe 2:

ANWENDER	KNOTEN	STUFE
Vorstand	ALLE	2

Der **Controller** darf alle Daten der Firma in allen Details, also in Stufe 1, sehen:

ANWENDER	KNOTEN	STUFE
Controller	ALLE	1

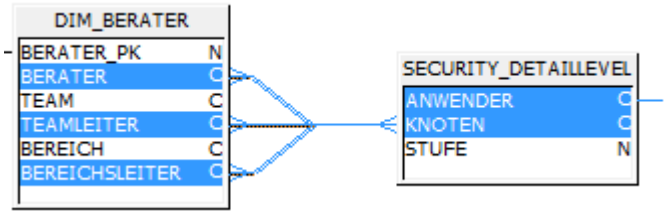
Die Frage ist nun aber: Wie wird diese Securitytabelle mit den anderen Tabellen verknüpft? Hierbei ist zunächst zu berücksichtigen, dass die zugewiesenen Knoten starke Überlappungen aufweisen, die aufzulösen sind.

7.4 Auflösung der Überlappungen und Zuweisung der Maximalrechte

In Abschnitt 5.2 wurden verschiedene Verfahren gezeigt, wie die Überlappungen aufgelöst werden können.

In Abschnitt 5.2.3.1 wurde dabei genauer das Verfahren über SELECT DISTINCT erläutert, dieses hilft uns hier aber nur begrenzt weiter: Wir können zwar die Eindeutige Zuordnung ANWENDER– BERATER_PK herstellen, *aber nicht gemeinsam mit der Stufe*: Diese führt zur Aufspaltung und damit zur mehrfachen Zuordnung.

Wir erzeugen zunächst die folgende Tabellenstruktur:



Das Auflösungs-SQL sieht wie folgt aus[15]:

```
SELECT DISTINCT
  SECURITY_DETAILLEVEL.ANWENDER,
  DIM_BERATER.BERATER_PK,
  DIM_BERATER.BERATER,
  SECURITY_DETAILLEVEL.STUFE
FROM
  SECURITY_DETAILLEVEL,
  DIM_BERATER
WHERE
  ( SECURITY_DETAILLEVEL.KNOTEN = DIM_BERATER.BERATER
OR
SECURITY_DETAILLEVEL.KNOTEN = DIM_BERATER.TEAMLEITER
OR
SECURITY_DETAILLEVEL.KNOTEN = DIM_BERATER.BEREICHSLEITER
OR
SECURITY_DETAILLEVEL.KNOTEN = 'ALLE'  )
  AND  ( SECURITY_DETAILLEVEL.ANWENDER = <aktueller Anwender>  )
```

Es ergibt für den Anwender Galilei mit den obigen Einträgen das folgende Ergebnis:

ANWENDER	BERATER_PK	BERATER	STUFE
Galilei	1	Newton	4
Galilei	2	Kopernikus	3
Galilei	2	Kopernikus	4
Galilei	3	Kepler	3
Galilei	3	Kepler	4
Galilei	4	Galilei	1
Galilei	4	Galilei	3
Galilei	4	Galilei	4
Galilei	5	Einstein	4
Galilei	6	Maxwell	4
Galilei	7	Planck	4
Galilei	8	Bohr	4
Galilei	9	Dirac	4
Galilei	10	Feynman	4
Galilei	11	Curie	4
Galilei	12	Bequerel	4

[15] Der BERATER wurde auch wieder der Übersichtlichkeit halber eingefügt, so dass die Mehrfachzuordnung verständlicher wird

Die farbig markierten Zeilen zeigen die mehrfachen Zuordnungen an.

Um dies aufzulösen, können wir die Tatsache verwenden, dass die verschiedenen Stufen aufeinander aufbauen: Wer Stufe 1 hat, benötigt nicht mehr Stufe 2,3 oder 4, weil deren Rechte bereits enthalten sind, analog für die höheren Stufen.

Wir greifen daher auf das Verfahren aus Abschnitt 5.2.5 zurück und ersetzen das SELECT DISTINCT durch eine Gruppierung. Als Aggregat bilden wir das Minimum der Stufe, weil alle anderen nicht interessieren, und dies löst gleichzeitig etwaige Mehrfachzuordnungen mit derselben Stufe, weil die Rückgabe wieder genau ein Satz je Kombination ist. Wir verwenden daher das folgende Auflösungs-SQL:

```
SELECT
  SECURITY_DETAILLEVEL.ANWENDER,
  DIM_BERATER.BERATER_PK,
  DIM_BERATER.BERATER,
  min(SECURITY_DETAILLEVEL.STUFE) as MINSTUFE
FROM
  SECURITY_DETAILLEVEL,
  DIM_BERATER
WHERE
  ( SECURITY_DETAILLEVEL.KNOTEN = DIM_BERATER.BERATER
OR
SECURITY_DETAILLEVEL.KNOTEN = DIM_BERATER.TEAMLEITER
OR
SECURITY_DETAILLEVEL.KNOTEN = DIM_BERATER.BEREICHSLEITER
OR
SECURITY_DETAILLEVEL.KNOTEN = 'ALLE'  )
  AND  ( SECURITY_DETAILLEVEL.ANWENDER = <aktueller Anwender>)
GROUP BY
  SECURITY_DETAILLEVEL.ANWENDER,
  DIM_BERATER.BERATER_PK,
  DIM_BERATER.BERATER
```

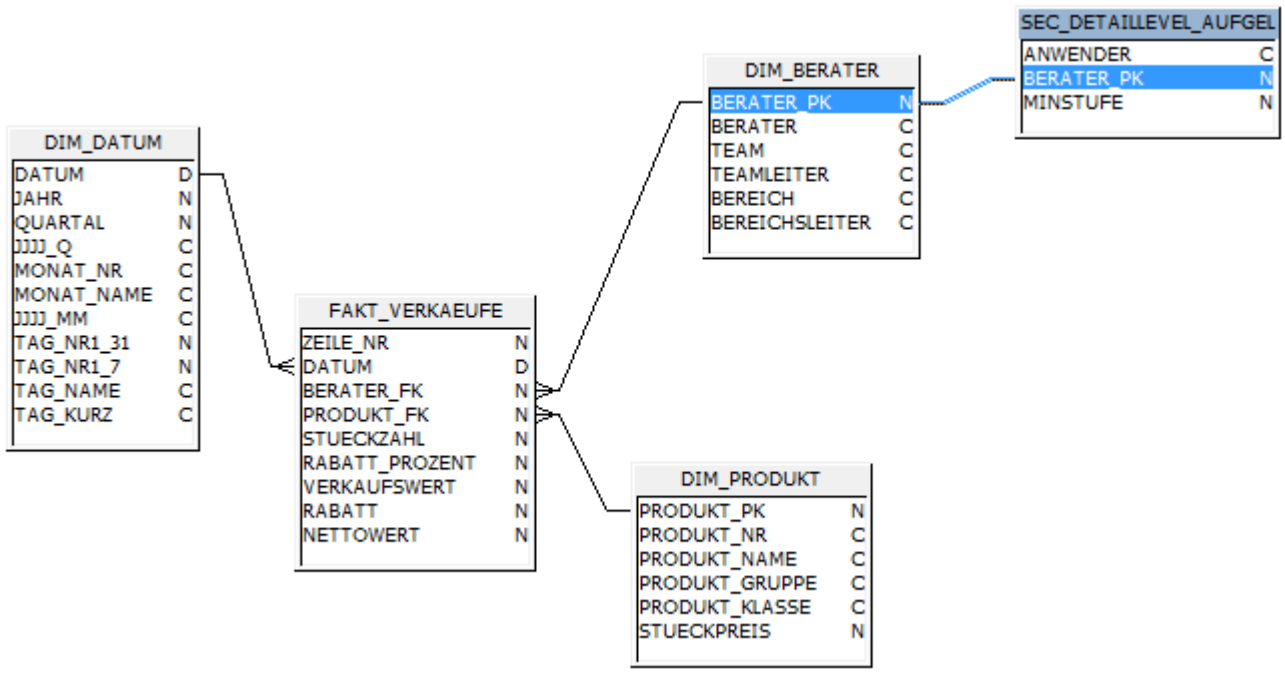
Für den Berater Galilei ergibt sich damit folgende Ergebnistabelle:

ANWENDER	BERATER_PK	BERATER	MINSTUFE
Galilei	1	Newton	4
Galilei	2	Kopernikus	3
Galilei	3	Kepler	3
Galilei	4	Galilei	1
Galilei	5	Einstein	4
Galilei	6	Maxwell	4
Galilei	7	Planck	4
Galilei	8	Bohr	4
Galilei	9	Dirac	4
Galilei	10	Feynman	4
Galilei	11	Curie	4
Galilei	12	Bequerel	4

Diese Abfrage ergibt tatsächlich für alle Anwender das korrekte Ergebnis, weshalb wir diese Struktur als ersten Schritt als abgeleitete Tabelle definieren (allerdings ohne das Feld BERATER) und in die Struktur integrieren.

Wir benötigen die ursprüngliche Securitytabelle nicht mehr, sie ist implizit in der Auflösung enthalten und auch schon auf den aktuell angemeldeten Anwender eingeschränkt.

Damit ergibt sich die folgende Tabellenstruktur:



Zusammenfassend: Die neue aufgelöste Securitytabelle ist eine Kombination der grundsätzlichen Zugriffsrechte (also der Row Level Security: welcher Anwender darf welche Datensätze sehen) mit den Detailrechten (welche Detailinformationen dürfen für jeden Satz gesehen werden).

Im nächsten Schritt werden wir diese Detailsichtbarkeit realisieren.

7.5 Objektdefinitionen mit Detailrechten; Kombination mit Kennzahlen

7.5.1 Objektdefinitionen der Beraterdimension

Betrachten wir zunächst die Spalten BEREICHSLEITER und BEREICH aus der Tabelle DIM_BERATER. Deren Inhalt darf von allen Anwendern, die für den Zugriff auf einen Dimensionseintrag berechtigt sind, gesehen werden.

Dies gilt bereits nicht mehr für die dem Team zugeordneten Spalten TEAM und TEAMLEITER: Für die Stufen 1, 2 und 3 ist sie sichtbar, für 4 aber nicht.

Analog ist die BERATER-Information nur sichtbar, wenn die Stufe 1 oder 2 ist, für 3 und 4 nicht.

Wir kombinieren zunächst die obige aufgelöste Security mit der Dimensionsinformation (mit allen Attributen!):

```
SELECT
  SEC_DETAILLEVEL_AUFGEL.ANWENDER,
  SEC_DETAILLEVEL_AUFGEL.MINSTUFE,
  DIM_BERATER.BERATER_PK,
  DIM_BERATER.BERATER,
  DIM_BERATER.TEAM,
  DIM_BERATER.TEAMLEITER,
  DIM_BERATER.BEREICH,
  DIM_BERATER.BEREICHSLEITER
FROM
  (
  SELECT
  SECURITY_DETAILLEVEL.ANWENDER,
  DIM_BERATER.BERATER_PK,
  min(SECURITY_DETAILLEVEL.STUFE) as MINSTUFE
FROM
  SECURITY_DETAILLEVEL,
  DIM_BERATER
WHERE
  ( SECURITY_DETAILLEVEL.KNOTEN = DIM_BERATER.BERATER
OR
SECURITY_DETAILLEVEL.KNOTEN = DIM_BERATER.TEAMLEITER
OR
SECURITY_DETAILLEVEL.KNOTEN = DIM_BERATER.BEREICHSLEITER
OR
SECURITY_DETAILLEVEL.KNOTEN = 'ALLE'  )
  AND  ( SECURITY_DETAILLEVEL.ANWENDER = <aktueller Anwender>  )
GROUP BY
  SECURITY_DETAILLEVEL.ANWENDER,
  DIM_BERATER.BERATER_PK
  )  SEC_DETAILLEVEL_AUFGEL,
  DIM_BERATER
WHERE
  ( DIM_BERATER.BERATER_PK=SEC_DETAILLEVEL_AUFGEL.BERATER_PK  )
```

Für den Anwender Galilei erhalten wir folgende Ergebnisse:

AN-WENDER	MIN-STUFE	BERATER _PK	BERATER	TEAM	TEAM-LEITER	BEREICH	BEREICHS-LEITER
Galilei	1	4	Galilei	Klassische Physiker	Kopernikus	Physiker	Newton
Galilei	3	2	Kopernikus	Klassische Physiker	Kopernikus	Physiker	Newton
Galilei	3	3	Kepler	Klassische Physiker	Kopernikus	Physiker	Newton
Galilei	4	1	Newton	Leitung Physiker	Newton	Physiker	Newton
Galilei	4	5	Einstein	Elektrodynamiker	Einstein	Physiker	Newton
Galilei	4	6	Maxwell	Elektrodynamiker	Einstein	Physiker	Newton
Galilei	4	7	Planck	Quantenphysiker	Planck	Physiker	Newton
Galilei	4	8	Bohr	Quantenphysiker	Planck	Physiker	Newton
Galilei	4	9	Dirac	Quantenphysiker	Planck	Physiker	Newton

Galilei	4	10	Feynman	Quantenphysiker	Planck	Physiker	Newton
Galilei	4	11	Curie	Radioaktivitätsforscher	Curie	Physiker	Newton
Galilei	4	12	Bequerel	Radioaktivitätsforscher	Curie	Physiker	Newton

Die oben genannten Regeln erfordern nun, dass für die MINSTUFE=4 alle Berater-und Teaminformationen ausgeblendet werden, für 3 alle Beraterinformationen. Wir benötigen allerdings weiterhin (siehe oben) den BERATER_PK als Referenz für die Fakten. Die gewünschte Ansicht sieht also wie folgt aus:

AN-WENDER	MIN-STUFE	BERATER_PK	BERATER	TEAM	TEAM-LEITER	BEREICH	BEREICHS-LEITER
Galilei	1	4	Galilei	Klassische Physiker	Kopernikus	Physiker	Newton
Galilei	3	2		Klassische Physiker	Kopernikus	Physiker	Newton
Galilei	3	3		Klassische Physiker	Kopernikus	Physiker	Newton
Galilei	4	1				Physiker	Newton
Galilei	4	5				Physiker	Newton
Galilei	4	6				Physiker	Newton
Galilei	4	7				Physiker	Newton
Galilei	4	8				Physiker	Newton
Galilei	4	9				Physiker	Newton
Galilei	4	10				Physiker	Newton
Galilei	4	11				Physiker	Newton
Galilei	4	12				Physiker	Newton

Dies ist am einfachsten mit einer Wertzuweisung über ein CASE-Statement zu erreichen: In Abhängigkeit vom Feld MIN-STUFE wird die entsprechende Information angezeigt, oder nicht.

Der Berater wird nur für die Stufen 1 und 2 angezeigt:

Berater:

```
CASE WHEN  SEC_DETAILLEVEL_AUFGEL.MINSTUFE IN (1, 2) then DIM_BERATER.BERATER
end
```

Das Team wird für die Stufen 1, 2 und 3 angezeigt, für 4 bleibt es leer:

Team:

```
CASE WHEN  SEC_DETAILLEVEL_AUFGEL.MINSTUFE IN (1, 2, 3) then DIM_BERATER.TEAM
end
```

Teamleiter:

```
CASE WHEN  SEC_DETAILLEVEL_AUFGEL.MINSTUFE IN (1, 2, 3) then
DIM_BERATER.TEAMLEITER end
```

Die Bereichsinformation ist immer sichtbar, wenn der Satz als solches gesehen werden darf, daher muss hier keine Änderung erfolgen.

Damit lautet das finale aufgelöste Statement wie folgt (Änderungen sind wieder fett markiert):

```
SELECT
  SEC_DETAILLEVEL_AUFGEL.ANWENDER,
  SEC_DETAILLEVEL_AUFGEL.MINSTUFE,
  DIM_BERATER.BERATER_PK,
CASE WHEN  SEC_DETAILLEVEL_AUFGEL.MINSTUFE IN (1, 2) then DIM_BERATER.BERATER
end as BERATER,
CASE WHEN  SEC_DETAILLEVEL_AUFGEL.MINSTUFE IN (1, 2, 3) then DIM_BERATER.TEAM
end as TEAM,
CASE WHEN  SEC_DETAILLEVEL_AUFGEL.MINSTUFE IN (1, 2, 3) then
DIM_BERATER.TEAMLEITER end as TEAMLEITER,
  DIM_BERATER.BEREICH,
  DIM_BERATER.BEREICHSLEITER
FROM
  (
  SELECT
  SECURITY_DETAILLEVEL.ANWENDER,
  DIM_BERATER.BERATER_PK,
  min(SECURITY_DETAILLEVEL.STUFE) as MINSTUFE
FROM
  SECURITY_DETAILLEVEL,
  DIM_BERATER
WHERE
  ( SECURITY_DETAILLEVEL.KNOTEN = DIM_BERATER.BERATER
OR
SECURITY_DETAILLEVEL.KNOTEN = DIM_BERATER.TEAMLEITER
OR
SECURITY_DETAILLEVEL.KNOTEN = DIM_BERATER.BEREICHSLEITER
OR
SECURITY_DETAILLEVEL.KNOTEN = 'ALLE'  )
  AND  ( SECURITY_DETAILLEVEL.ANWENDER = <aktueller Anwender>  )
GROUP BY
  SECURITY_DETAILLEVEL.ANWENDER,
  DIM_BERATER.BERATER_PK
  )  SEC_DETAILLEVEL_AUFGEL,
  DIM_BERATER
WHERE
  ( DIM_BERATER.BERATER_PK=SEC_DETAILLEVEL_AUFGEL.BERATER_PK  )
```

7.5.2 Abfrageergebnisse: Kombination mit Kennzahlen

Dieses Konstrukt wird verständlich, sobald die Dimensionsinformationen mit Kennzahlen kombiniert werden und dabei eine Gruppierung nach diesen Objekten erfolgt. In einer solchen Abfrage wird der BERATER_PK zwar im Join verwendet, er ist aber für die Anwender nicht mehr sichtbar.

Der oben verwendete Standardbericht sieht für den Berater Galilei ohne Verwendung der Detailstufen, also bei voller Sichtbarkeit aller Details, wie folgt aus:

			JAHR					
BEREICHS-LEITER	**TEAM-LEITER**	**BERATER**	2010	2011	2012	2013	2014	2015
Newton	Curie	Bequerel	4.455,28	9.467,57	31.055,18	50.339,32	124.118,97	141.961,86
		Curie	147.291,00	142.175,00	122.304,82	70.656,88	41.396,24	25.104,97

	Einstein	Einstein	23.168,49	10.089,11	25.698,73	55.095,40	57.687,38	85.034,61
		Maxwell	114.619,00	115.937,00	87.782,00	41.324,41	38.885,92	31.350,69
	Kopernikus	Galilei	127.322,93	135.801,10	64.013,86	52.467,05	49.518,79	59.713,46
		Kepler	36.470,00	45.890,00	171.390,00	170.842,00	195.593,00	181.883,00
		Kopernikus	137.247,00	116.952,00	59.616,37	40.558,40	10.900,21	12.686,42
	Newton	Newton	875,74	13.138,51	11.014,36	14.388,12	19.927,52	47.098,72
	Planck	Bohr	114.173,06	114.436,51	94.613,04	96.883,05	63.040,75	87.394,17
		Dirac	87.736,25	26.702,50	35.555,47	38.310,63	58.437,28	35.721,95
		Feynman	88.192,00	141.152,00	141.566,00	133.730,00	124.917,35	100.403,67
		Planck	49.110,00	81.633,00	57.715,00	114.963,00	104.595,00	112.624,43

Werden aber die neuen Definitionen verwendet, so erhalten wir folgendes Ergebnis:

			JAHR					
BEREICHS-LEITER	**TEAM-LEITER**	**BERATER**	2010	2011	2012	2013	2014	2015
Newton	Kopernikus	Galilei	127.322,93	135.801,10	64.013,86	52.467,05	49.518,79	59.713,46
			173.717,00	162.842,00	231.006,37	211.400,40	206.493,21	194.569,42
			629.620,82	654.731,20	607.304,60	615.690,81	633.006,41	666.695,07

Das Ergebnis für den Teamleiter Kopernikus soll sein eigenes Team im Detail (bis auf den Berater) zeigen, sowie alle anderen Teams im Bereich als Teamsumme, und das ist auch der Fall:

			JAHR					
BEREICHS-LEITER	**TEAM-LEITER**	**BERATER**	2010	2011	2012	2013	2014	2015
Newton	Curie		151.746,28	151.642,57	153.360,00	120.996,20	165.515,21	167.066,83
	Einstein		137.787,49	126.026,11	113.480,73	96.419,81	96.573,30	116.385,30
	Kopernikus	Galilei	127.322,93	135.801,10	64.013,86	52.467,05	49.518,79	59.713,46
		Kepler	36.470,00	45.890,00	171.390,00	170.842,00	195.593,00	181.883,00
		Kopernikus	137.247,00	116.952,00	59.616,37	40.558,40	10.900,21	12.686,42
	Newton		875,74	13.138,51	11.014,36	14.388,12	19.927,52	47.098,72
	Planck		339.211,31	363.924,01	329.449,51	383.886,68	350.990,38	336.144,22

Der Bereichsleiter Newton darf alle Berater seines Bereichs sehen, sowie den anderen Bereich (Euklid) in Summe:

			JAHR					
BEREICHS-LEITER	**TEAM-LEITER**	**BERATER**	2010	2011	2012	2013	2014	2015
Euklid			1.716.156,99	1.667.400,92	1.696.789,20	1.725.361,18	1.713.110,69	1.652.516,54
Newton	Curie	Bequerel	4.455,28	9.467,57	31.055,18	50.339,32	124.118,97	141.961,86
		Curie	147.291,00	142.175,00	122.304,82	70.656,88	41.396,24	25.104,97
	Einstein	Einstein	23.168,49	10.089,11	25.698,73	55.095,40	57.687,38	85.034,61
		Maxwell	114.619,00	115.937,00	87.782,00	41.324,41	38.885,92	31.350,69
	Kopernikus	Galilei	127.322,93	135.801,10	64.013,86	52.467,05	49.518,79	59.713,46
		Kepler	36.470,00	45.890,00	171.390,00	170.842,00	195.593,00	181.883,00
		Kopernikus	137.247,00	116.952,00	59.616,37	40.558,40	10.900,21	12.686,42
	Newton	Newton	875,74	13.138,51	11.014,36	14.388,12	19.927,52	47.098,72
	Planck	Bohr	114.173,06	114.436,51	94.613,04	96.883,05	63.040,75	87.394,17
		Dirac	87.736,25	26.702,50	35.555,47	38.310,63	58.437,28	35.721,95
		Feynman	88.192,00	141.152,00	141.566,00	133.730,00	124.917,35	100.403,67
		Planck	49.110,00	81.633,00	57.715,00	114.963,00	104.595,00	112.624,43

7.5.3 Detailrechte auf die Faktentabelle

Die Anforderungen in Abschnitt 7.2 waren auch für die Informationen in der Faktentabelle unterschiedlich:

- Der Verkaufswert ist immer sichtbar, sofern Berechtigungen auf den Satz vorhanden sind.
- Der Nettowert ist für die Stufen 1 und 2 sichtbar.
- alle anderen Fakteninformationen sind nur für Stufe 1 sichtbar.

Wir definieren daher analog zu Abschnitt 7.5.1 Abfrageobjekte, die die Detailsichtbarkeit der Fakteninformation steuern. Da der Verkaufswert immer sichtbar ist, muss diese Definition nicht geändert werden.

Die anderen Definitionen lauten (bitte beachten: Zeile_Nr und Rabatt Prozent sind keine Kennzahlen, sondern Dimensionale Attribute aus der Faktentabelle, daher verwenden sie keine Summe!):

Nettowert:

```
sum(
CASE WHEN  SEC_DETAILLEVEL_AUFGEL.MINSTUFE IN (1, 2) then
FAKT_VERKAEUFE.NETTOWERT end
)
```

Zeile_Nr:

```
CASE WHEN  SEC_DETAILLEVEL_AUFGEL.MINSTUFE = 1 then FAKT_VERKAEUFE.ZEILE_NR
end
```

Stückzahl:

```
sum(
CASE WHEN  SEC_DETAILLEVEL_AUFGEL.MINSTUFE = 1 then FAKT_VERKAEUFE.
FAKT_VERKAEUFE.STUECKZAHL end
)
```

Rabatt Prozent:

```
CASE WHEN  SEC_DETAILLEVEL_AUFGEL.MINSTUFE = 1 then FAKT_VERKAEUFE.
FAKT_VERKAEUFE.RABATT_PROZENT end
```

Rabatt:

```
sum(
CASE WHEN  SEC_DETAILLEVEL_AUFGEL.MINSTUFE = 1 then
FAKT_VERKAEUFE.FAKT_VERKAEUFE.RABATT end
)
```

Um die Auswirkungen zu sehen, betrachten wir zunächst als Vergleich das Ergebnis eines Berichts, für den keinerlei Security angewandt wurde: Wir betrachten für alle Berater in der Hierarchie die Ergebnisse für das Jahr 2013, und zwar die drei Kennzahlen Verkaufswert, Nettowert und Rabatt:

BEREICHS-LEITER	TEAM-LEITER	BERATER	RABATT	NETTOWERT	VERKAUFS-WERT
Euklid	**Babbage**	Babbage	20.068,99	132.756,8	152.825,79
		Turing	47.036,05	323.359,27	370.395,32
		v. Neumann	17.383,38	225.236,7	242.620,08
	Babbage				**765.841,19**
	Euklid	Euklid	3.054,57	133.995,2	137.049,77
	Euklid				**137.049,77**
	Pascal	Euler	1.550,88	80.773,22	82.324,1
		Gauss	7.637,8	119.945,34	127.583,14
		Germain	19.361,87	112.904,85	132.266,72
		Pascal	5.613,84	51.320,98	56.934,82
	Pascal				**399.108,78**
	Riemann	Hilbert	7.245,63	138.790,19	146.035,82
		Noether	18.227,13	131.710,87	149.938
		Riemann	20.866,5	106.521,12	127.387,62
	Riemann				**423.361,44**
Euklid					**1.725.361,18**

Newton	**Curie**	Bequerel	8.435,33	41.903,99	50.339,32
		Curie	13.377,16	57.279,72	70.656,88
	Curie				**120.996,2**
	Einstein	Einstein	11.264,5	43.830,9	55.095,4
		Maxwell	5.884,62	35.439,79	41.324,41
	Einstein				**96.419,81**
	Kopernikus	Galilei	537,39	51.929,66	52.467,05
		Kepler	15.848,4	154.993,6	170.842
		Kopernikus	5.369,2	35.189,2	40.558,4
	Kopernikus				**263.867,45**

	Newton	Newton	724,63	13.663,49	14.388,12
	Newton				**14.388,12**
	Planck	Bohr	9.923,46	86.959,59	96.883,05
		Dirac	1.572,15	36.738,48	38.310,63
		Feynman	9.450,1	124.279,9	133.730
		Planck	15.248,93	99.714,07	114.963
	Planck				**383.886,68**
Newton					**879.558,26**

Nun verwenden wir die eingeschränkten Objektdefinitionen und betrachten die Ergebnisse für verschiedene Anwender. Um zu zeigen, welche Auswirkung die jeweilige höchste Berechtigung hat, wird sie als zusätzliche Spalte (Minstufe[16]) ergänzt.

Als erstes Betrachten wir den Berater Kepler: Er darf seine eigenen Daten im vollen Detail (Stufe 1) sehen, damit auch alle drei Kennzahlen, zusätzlich sein eigenes Team Kopernikus als Summe (Stufe 3) und den Bereich Newton in Summe (Stufe 4). Damit sieht er den Bericht wie folgt:

BEREICHS-LEITER	TEAM-LEITER	BERATER	MINSTUFE	RABATT	NETTOWERT	VERKAUFS-WERT
Newton	**Kopernikus**	Kepler	1	15.848,4	154.993,6	170.842
			3			93.025,45
	Kopernikus					**263.867,45**
			4			615.690,81
						615.690,81
Newton						**879.558,26**

Betrachten wir nun seinen Teamkollegen, den Berater Galilei, der dieselben Team-und Bereichsrechte hat, aber seinen eigenen Beraterknoten im Detail sehen darf:

BEREICHS-LEITER	TEAM-LEITER	BERATER	MINSTUFE	RABATT	NETTOWERT	VERKAUFS-WERT
Newton	**Kopernikus**	Galilei	1	537,39	51.929,66	52.467,05
			3			211.400,4
	Kopernikus					**263.867,45**
			4			615.690,81
						615.690,81
Newton						**879.558,26**

Hier ist bereits ein weiterer Vorteil des Verfahrens sichtbar: Wurde in den obigen Beispielen eine Team- oder Bereichssumme gebildet, so wurde lediglich das Ergebnis des Beraters wiederholt, da dies die einzigen zugewiesenen Datenrechte waren. Im Gegensatz dazu können hier nun korrekte Summen für nicht-eingeschränkte Kennzahlen wie den Verkaufswert auf höheren Stufen sinnvoll ermittelt werden!

[16] Bitte nicht verwirren lassen: Die kleinste Stufe (1) hat die größten Rechte, daher ist die minimale Stufe mit den maximalen Rechten verbunden!

Wir betrachten nun die Ergebnisse des Teamleiters Kopernikus:

BEREICHS-LEITER	TEAM-LEITER	BERATER	MINSTUFE	RABATT	NETTOWERT	VERKAUFS-WERT
Newton	**Curie**		3			120.996,2
	Curie					**120.996,2**
	Einstein		3			96.419,81
	Einstein					**96.419,81**
	Kopernikus	Galilei	1	537,39	51.929,66	52.467,05
		Kepler	1	15.848,4	154.993,6	170.842
		Kopernikus	1	5.369,2	35.189,2	40.558,4
	Kopernikus					**263.867,45**
	Newton		3			14.388,12
	Newton					**14.388,12**
	Planck		3			383.886,68
	Planck					**383.886,68**
Newton						**879.558,26**

Er sieht von seinem eigenen Team alle Details und Kennzahlen, von den anderen Teams im Bereich aber nur die Teamsummen und auch nur für den Verkaufswert.

Nun die Ergebnisse für den Bereichsleiter Newton, der für en eigenen Bereich Stufe 2 (bis auf den Berater, aber nicht alle Kennzahlen) und für alle anderen Bereiche die Stufe 4 (Bereichssumme des Verkaufswerts) sehen darf:

BEREICHS-LEITER	TEAM-LEITER	BERATER	MINSTUFE	RABATT	NETTOWERT	VERKAUFS-WERT
Euklid			4			1.725.361,18
						1.725.361,18
Euklid						**1.725.361,18**

Newton	**Curie**	Bequerel	2		41.903,99	50.339,32
		Curie	2		57.279,72	70.656,88
	Curie					**120.996,2**
	Einstein	Einstein	2		43.830,9	55.095,4
		Maxwell	2		35.439,79	41.324,41
	Einstein					**96.419,81**
	Kopernikus	Galilei	2		51.929,66	52.467,05
		Kepler	2		154.993,6	170.842
		Kopernikus	2		35.189,2	40.558,4
	Kopernikus					**263.867,45**
	Newton	Newton	2		13.663,49	14.388,12

	Newton					**14.388,12**
	Planck	Bohr	2		86.959,59	96.883,05
		Dirac	2		36.738,48	38.310,63
		Feynman	2		124.279,9	133.730
		Planck	2		99.714,07	114.963
	Planck					**383.886,68**
Newton						**879.558,26**

Der Anwender „Vorstand“ hat für alle Daten des Unternehmens die Stufe 2:

BEREICHS-LEITER	TEAM-LEITER	BERATER	MINSTUFE	RABATT	NETTOWERT	VERKAUFS-WERT
Euklid	**Babbage**	Babbage	2		132.756,8	152.825,79
		Turing	2		323.359,27	370.395,32
		v. Neumann	2		225.236,7	242.620,08
	Babbage					**765.841,19**
	Euklid	Euklid	2		133.995,2	137.049,77
	Euklid					**137.049,77**
	Pascal	Euler	2		80.773,22	82.324,1
		Gauss	2		119.945,34	127.583,14
		Germain	2		112.904,85	132.266,72
		Pascal	2		51.320,98	56.934,82
	Pascal					**399.108,78**
	Riemann	Hilbert	2		138.790,19	146.035,82
		Noether	2		131.710,87	149.938
		Riemann	2		106.521,12	127.387,62
	Riemann					**423.361,44**
Euklid						**1.725.361,18**
Newton	**Curie**	Bequerel	2		41.903,99	50.339,32
		Curie	2		57.279,72	70.656,88
	Curie					**120.996,2**
	Einstein	Einstein	2		43.830,9	55.095,4
		Maxwell	2		35.439,79	41.324,41
	Einstein					**96.419,81**
	Kopernikus	Galilei	2		51.929,66	52.467,05
		Kepler	2		154.993,6	170.842
		Kopernikus	2		35.189,2	40.558,4
	Kopernikus					**263.867,45**
	Newton	Newton	2		13.663,49	14.388,12

	Newton					**14.388,12**
	Planck	Bohr	2		86.959,59	96.883,05
		Dirac	2		36.738,48	38.310,63
		Feynman	2		124.279,9	133.730
		Planck	2		99.714,07	114.963
	Planck					**383.886,68**
Newton						**879.558,26**

Und schließlich die Sicht für den Anwender „Controller", der für alle Unternehmensdaten die Stufe 1 zugewiesen bekommen hat:

BEREICHS-LEITER	TEAM-LEITER	BERATER	MINSTUFE	RABATT	NETTOWERT	VERKAUFS-WERT
Euklid	**Babbage**	Babbage	1	20.068,99	132.756,8	152.825,79
		Turing	1	47.036,05	323.359,27	370.395,32
		v. Neumann	1	17.383,38	225.236,7	242.620,08
	Babbage					**765.841,19**
	Euklid	Euklid	1	3.054,57	133.995,2	137.049,77
	Euklid					**137.049,77**
	Pascal	Euler	1	1.550,88	80.773,22	82.324,1
		Gauss	1	7.637,8	119.945,34	127.583,14
		Germain	1	19.361,87	112.904,85	132.266,72
		Pascal	1	5.613,84	51.320,98	56.934,82
	Pascal					**399.108,78**
	Riemann	Hilbert	1	7.245,63	138.790,19	146.035,82
		Noether	1	18.227,13	131.710,87	149.938
		Riemann	1	20.866,5	106.521,12	127.387,62
	Riemann					**423.361,44**
Euklid						**1.725.361,18**
Newton	**Curie**	Bequerel	1	8.435,33	41.903,99	50.339,32
		Curie	1	13.377,16	57.279,72	70.656,88
	Curie					**120.996,2**
	Einstein	Einstein	1	11.264,5	43.830,9	55.095,4
		Maxwell	1	5.884,62	35.439,79	41.324,41
	Einstein					**96.419,81**
	Kopernikus	Galilei	1	537,39	51.929,66	52.467,05
		Kepler	1	15.848,4	154.993,6	170.842
		Kopernikus	1	5.369,2	35.189,2	40.558,4
	Kopernikus					**263.867,45**

	Newton	Newton	1	724,63	13.663,49	14.388,12
	Newton					**14.388,12**
	Planck	Bohr	1	9.923,46	86.959,59	96.883,05
		Dirac	1	1.572,15	36.738,48	38.310,63
		Feynman	1	9.450,1	124.279,9	133.730
		Planck	1	15.248,93	99.714,07	114.963
	Planck					**383.886,68**
Newton						**879.558,26**

7.6 Verwendung des Verfahrens zur Anonymisierung?

Das zuvor beschriebene Verfahren führt dazu, dass bestimmte Detailinformationen nur für berechtigte Personen sichtbar werden. Der Gedanke könnte daher nahe liegen, es auch für die Anonymisierung von Daten zu verwenden.

In gewissem Umfang ist dies sicher möglich, nämlich dann, wenn es um eine personalisierte Anonymisierung geht, genau das ist da das Ergebnis des Verfahrens.

Anonymisierung ist aber meistens in einem viel weiteren Kontext zu sehen: Dem von rechtlichen Vorschriften wie dem Datenschutz, die eine physische Anonymisierung der Daten erfordern, d.h. der Änderung der Datensätze in der Datenbank. Sehr häufig wird auch ein Verfahren angewandt, in dem die Inhalte randomisiert überschrieben werden: Bestehende Informationen (z.B. Kundennamen) werden mit zufälligen Werten überschrieben, und zwar auch wieder auf Datenbankebene.

Beides kann mit dem obigen Verfahren nicht erreicht werden: Erstens ändert es die Daten nur für die Ansicht ab und überschreibt die Daten nicht in der Datenbank (weder mit Leer-, noch mit Zufallswerten), zweitens erfordern rechtlich akzeptierte Randomisierungsverfahren z.T. ausgeklügelte Algorithmen.

7.7 Betrachtungen zur Abfrageperformance

Wie schon in den Abschnitten 5.1.4 und 5.2.6 soll hier kurz die Abfrageperformance betrachtet werden.

Die oben beschriebenen Auflösungen der Securitytabelle, verbunden über mehrere abgeleitete Tabellen und späteren Case-Statements in den Objektdefinitionen können zu massiven Problemen in den Abfragen führen. Beispielsweise können solche Gesamtabfragen von den Datenbanken nur noch schwer optimiert werden, was zu Full Table Scans u.ä. führen kann.

Gerade bei sehr großen Datenmengen ist die einzige sinnvolle Option hier eine Materialisierung der aufgelösten Strukturen, bei der also die fertig aufbereiteten Dimensionsinformationen für alle Benutzer gespeichert und dann bei Bedarf aufgerufen werden. Kennzahlen aus den Fakten würden so nur selten fertig vorberechnet werden, sie sind aber in Bezug auf die Indexzugriffe (um diese geht es hier hauptsächlich) auch nicht so kritisch.

Auch hier sei wieder auf Abschnitt 12 verwiesen, wo wir dies näher beleuchten.

8 Einschränkungen auf mehrere Dimensionstabellen

8.1 Einführendes Beispiel

Bisher haben wir nur Einschränkungen auf die Organisationshierarchie vorgenommen, grundsätzlich lassen sich aber auf alle dimensionalen Attribute Einschränkungen vornehmen. Als klassisches Beispiel ließe sich ein Produktmanager denken, der alle Daten bestimmten Produkten sehen darf, aber über alle Berater hinweg.

Mit Bezug auf die in Abschnitt 0 vorgestellte Produktdimension definieren wir beispielhaft die folgenden Produktmanager:

- einen Generalmanager für die Produktklasse Möbel: GM_Möbel

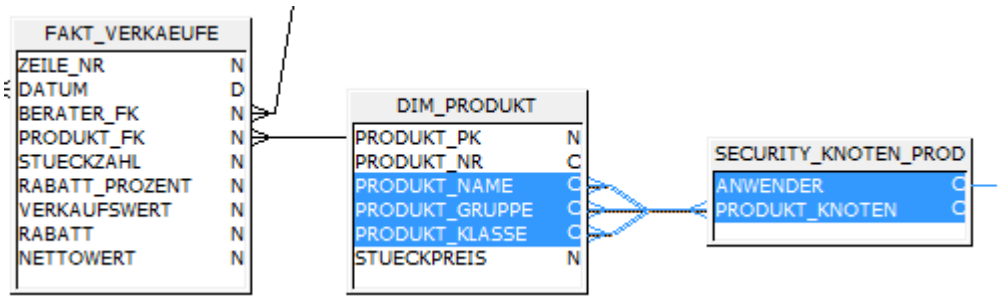
- einen Gruppenmanager GRP_Küche für die Produktgruppen Küche und Großgeräte (übergreifend über zwei Produktklassen: In der Gruppe „Großgeräte“ sind Küchengeräte zusammengefasst!)

- einen Produktmanager für Staubsauger und Fernseher: Er heißt Maier

Wenn wir dies mit den zuvor beschriebenen Lösungen vergleichen, benötigen wir eine neue Securitytabelle auf Knotenebene für die Produkte (vgl. Abschnitt 5.1), und diese benötigt folgende Einträge:

ANWENDER	PRODUKT_KNOTEN
GM_Möbel	Möbel
GRP_Küche	Küche
GRP_Küche	Großgeräte
Maier	Staubsauger
Maier	Fernseher

Die oben beschriebenen Verfahren (Auflösung von Mehrfachzählungen, Detailrechtzuweisungen usw.) können hier alle analog angewandt werden. Wir gehen zur einfacheren Erläuterung von der nicht-überlappenden Zuweisung von Rechten (also ohne Mehrfachzählungen) aus und erhalten folgende neue Struktur:



Hierbei lautet die Joinbedingung zwischen der Securitytabelle und der Produktdimension

```
SECURITY_KNOTEN_PROD.PRODUKT_KNOTEN=DIM_PRODUKT.PRODUKT_NAME
OR
SECURITY_KNOTEN_PROD.PRODUKT_KNOTEN=DIM_PRODUKT.PRODUKT_GRUPPE
OR
SECURITY_KNOTEN_PROD.PRODUKT_KNOTEN=DIM_PRODUKT.PRODUKT_KLASSE
OR
SECURITY_KNOTEN_PROD.PRODUKT_KNOTEN= 'ALLE'
```

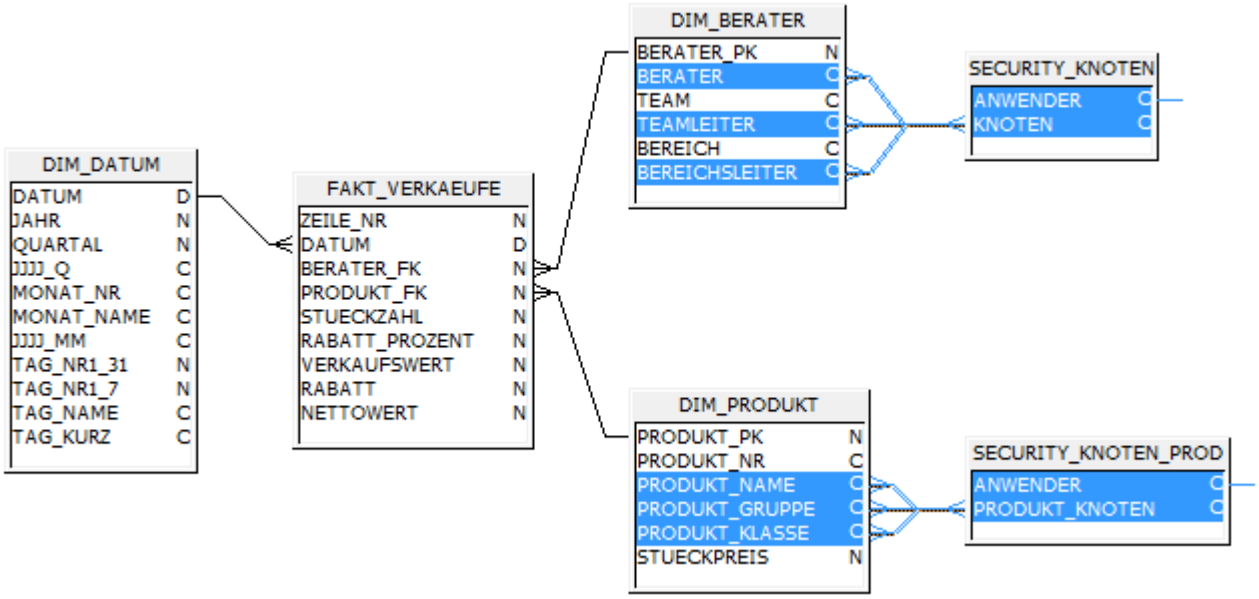
Auf die Securitytabelle erfolgt die übliche Zwangsbedingung

```
SECURITY_KNOTEN_PROD.ANWENDER=<aktueller Anwender>
```

Bis hier ist noch nichts neues passiert, die Ergebnisse für die oben beschriebenen Anwender wären auch genau die erwarteten, wenn es nur diese eine Securitytabelle in der Tabellenstruktur gäbe (d.h. jeder Anwender sähe genau die erlaubten Produktdaten), gemeinsam mit den zuvor beschriebenen Beratereinschränkungen[17] wären aber im aktuellen Zustand die Ergebnisse für alle Anwender leer! Das gilt auch für alle oben definierten Anwender.

Was ist passiert?

Der Grund ist, dass beide Securityeinschränkungen gleichzeitig angewandt werden. Da jeder der bisher definierten Anwender nur in jeweils einer der Securitytabellen steht, führt die jeweils andere Einschränkung zu einer leeren Ergebnismenge.



Offenkundig ist dies so nicht korrekt.

8.2 Lösungsansätze

8.2.1 Lösung 1: die Securitytabellen werden anwenderspezifisch zugewiesen

Falls das Berichtswerkzeug dies zulässt, könnten zwei Anwendergruppen definiert werden: Eine für die Produkt- und eine für die Beratersecurity. In die entsprechenden Gruppen werden die jeweiligen Anwender eingetragen.

Nun wird für jede Gruppe definiert, welche der Securitybedingungen und damit welche der Securitytabellen aktiv wird: Für die Produktmanager wird die Produktsecurity aktiv, für die Berater die Beratersecurity.

Das ist aber nur möglich, solange nicht die Steuerung der Detailsichtbarkeit, wie in Abschnitt 7 erläutert, verwendet wird, da in den finalen CASE-Statements der Objektdefinitionen die Securitytabellen intrinsisch enthalten sind und daher nicht „ausgeschaltet“ werden können.

Außerdem lassen sich komplexere Fälle so nur schwer abbilden.

[17] Egal, welche der obigen Versionen verwendet wird!

8.2.2 Lösung 2: Ergänzung der jeweils anderen Securitytabelle um „ALLE“-Einträge

In Abschnitt 5.1.3.3 wurde beschrieben, wir Anwender mit ALLE-Rechten versehen werden können. Gemeinsam mit der Erweiterung der Joinbedingung führt dies für Anwender mit diesem Recht zu einer Bedingung, die äquivalent zu 1=1 ist und somit die Securitytabelle de facto ignoriert. Genau dies soll für jeden Anwender der einen Securitytabelle in der jeweils anderen der Fall sein.

Damit ist die Lösung die, dass jeder Anwender, der in der einen Securitytabelle mit einem Recht ausgestattet wird, in der anderen den Knoten „ALLE“ zugewiesen bekommt.

Als Beispiel betrachten wir die beiden Securitytabellen[18] für die Anwender Galilei, Kopernikus, Newton, GM_Möbel, GRP_Küche, Maier und Vorstand:

SECURITY_KNOTEN_PROD	
ANWENDER	**PRODUKT_KNOTEN**
GM_Möbel	Möbel
GRP_Küche	Küche
GRP_Küche	Großgeräte
Maier	Staubsauger
Maier	Fernseher
Newton	ALLE
Kopernikus	ALLE
Galilei	ALLE
Vorstand	ALLE

SECURITY_KNOTEN	
ANWENDER	**KNOTEN**
Newton	Newton
Kopernikus	Kopernikus
Galilei	Galilei
GM_Möbel	ALLE
GRP_Küche	ALLE
Maier	ALLE
Vorstand	ALLE

8.3 Kombinierte Rechtezuweisungen

8.3.1 Einfache Kombinationen

In manchen Fällen sind Einschränkungen auf beide Securitytabellen notwendig. Ein typisches Beispiel: Ein Anwender ist Produktmanager für eine bestimmte Produktgruppe und eine bestimmte geografische Region.

In unserem Beispiel: Ein Anwender ist Produktmanager für die Möbel, aber nur für das Team Riemann innerhalb des Bereichs Euklid.

Damit sind folgende Einträge in den Securitytabellen notwendig (der neue Anwender heißt „PM_kombiniert“:

SECURITY_KNOTEN_PROD	
ANWENDER	**PRODUKT_KNOTEN**
PM_kombiniert	Möbel

SECURITY_KNOTEN	
ANWENDER	**KNOTEN**
PM_kombiniert	Riemann

Da die beiden Einschränkungen mit AND verknüpft werden, sind die Ergebnisse genau die gewünschten.

[18] Wir verwenden hier die einfache Knotensecurity aus Abschnitt 5, aber grundsätzlich lassen sich alle obigen Varianten dafür verwenden.

8.3.2 Mehrfache Zuweisungen

Wir erweitern die Rechte für den Anwender PM_kombiniert: Er darf nun die Produkt-Knoten Möbel (Produktklasse), Stereo_HiFi (Produktgruppe) sowie die Produkte Kühlschrank und Schokolade sehen; diese Daten darf er für den Bereich Newton, das Team Riemann und die Berater Pascal und Turing sehen. Damit benötigt er die folgenden Einträge:

SECURITY_KNOTEN_PROD	
ANWENDER	**PRODUKT_KNOTEN**
PM_kombiniert	Möbel
PM_kombiniert	Stereo_HiFi
PM_kombiniert	Kühlschrank
PM_kombiniert	Schokolade

SECURITY_KNOTEN	
ANWENDER	**KNOTEN**
PM_kombiniert	Newton
PM_kombiniert	Riemann
PM_kombiniert	Pascal
PM_kombiniert	Turing

Bis hier konnten alle Rechtezuweisungen abgedeckt werden, weil die Schnittmengenbildung zwischen den beiden Securitytabellen auf die jeweiligen Gesamtrechte angewandt wurde, etwa wie in der folgenden Grafik schematisch dargestellt: Der Anwender ist auf die Produkte P_1, P_5 und P_6 und auf die Berater B_B und B_C eingeschränkt, er darf daher die Daten der sechs Schnittpunkte sehen.

		BERATER					
		B_A	B_B	B_C	B_D	B_E	
PRODUKT	P_1		X	X			
	P_2						
	P_3						
	P_4						
	P_5		X	X			
	P_6		X	X			

Wir werden nun einen Fall sehen, der mit den bestehenden Strukturen nicht mehr abgebildet werden kann.

8.3.3 Zuweisungen von expliziten Kombinationen

Angenommen, ein Anwender soll Produktmanager für Großgeräte im Team Riemann und für Schlafzimmermöbel für die Berater Bohr und Dirac sein.

Die folgende Zuweisung sieht zunächst danach aus:

SECURITY_KNOTEN_PROD	
ANWENDER	**PRODUKT_KNOTEN**
PM_explizit	Großgeräte
PM_explizit	Schlafzimmermöbel

SECURITY_KNOTEN	
ANWENDER	**KNOTEN**
PM_explizit	Riemann
PM_explizit	Bohr
PM_explizit	Dirac

Leider ist dies nicht korrekt: Damit dürfte der Anwender auch die Großgeräte für Bohr und Dirac und die Schlafzimmermöbel für das Team Riemann sehen, was aber nicht erlaubt sein soll.

Es gibt keine direkte Möglichkeit, diesen Fall mit den bestehenden Strukturen abzubilden, vielmehr muss zunächst eine kombinierte Securitytabelle definiert werden:

ANWENDER	PRODUKT_KNOTEN	BERATER_KNOTEN
PM_explizit	Großgeräte	Riemann
PM_explizit	Schlafzimmermöbel	Bohr
PM_explizit	Schlafzimmermöbel	Dirac

Dummerweise müsste diese Tabelle nun mit beiden Dimensionstabellen verknüpft werden, so dass die Kombinationen aufgelöst werden, was so wieder nicht möglich ist.

Die einzige korrekte Möglichkeit wäre es, die Einschränkung auf die Kombinationen der Produkt- und Beraterschlüssel in der Faktentabelle selbst vorzunehmen (da nur dort diese Kombinationen (oder auch „Tupel“) auftreten! Dort wiederum tauchen nur die Fremdschlüssel in Form von Integerschlüsseln auf (oder „bestenfalls“ die Berater und Produkte auf unterster Ebene).

Die notwendigen Schritte wären daher:

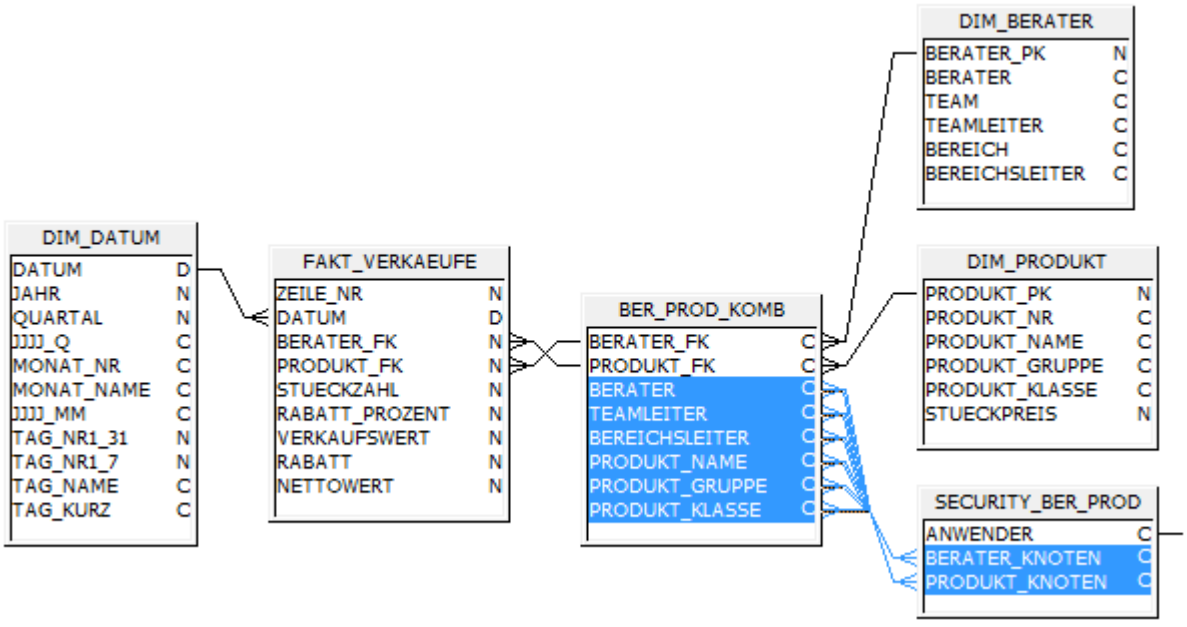
- Ermitteln aller Schlüsselkombinationen in der Faktentabelle in einer neuen Tabelle
- Ergänzung dieser Tabelle um alle einzuschränkenden Attribute aus den Dimensionstabelle
- Verknüpfung dieser Tabelle mit der kombinierten Securitytabelle

Eine solche zusätzliche Tabelle müsste im ETL erzeugt werden und wäre zudem wahrscheinlich sehr groß, weil ja die beiden Dimensionen bisher nicht grundlos getrennt waren! Im schlimmsten Fall, in dem jeder Berater jedes Produkt verkaufen kann, führt dies zu einem kartesischen Produkt der beiden Dimensionen! Aus Performancegesichtspunkten ist dies höchst problematisch.

Diese Verknüpfung hat ihre eigenen Herausforderungen:

Die Bedingungen auf Produkt- und Beraterebene müssen gleichzeitig, also mit AND verknüpft sein, allerdings kann jede der beiden Bedingungen auf jeden Knoten erfolgen, weshalb diese zwei Grundbedingungen aus den bekannten vier mit OR verknüpften Einzelbedingungen bestehen!

Strukturell sieht dies wie folgt aus:



Die Joinbedingung zwischen der Kombinations- und der Securitytabelle lautet:

```
(SECURITY_BER_PROD.BERATER_KNOTEN=BER_PROD_KOMB.BERATER
OR
SECURITY_BER_PROD.BERATER_KNOTEN=BER_PROD_KOMB.TEAMLEITER
OR
SECURITY_BER_PROD.BERATER_KNOTEN=BER_PROD_KOMB.BEREICHSLEITER
OR
SECURITY_BER_PROD.BERATER_KNOTEN= 'ALLE')

AND

(SECURITY_BER_PROD.PRODUKT_KNOTEN=BER_PROD_KOMB.PRODUKT_NAME
OR
SECURITY_BER_PROD.PRODUKT_KNOTEN=BER_PROD_KOMB.PRODUKT_GRUPPE
OR
SECURITY_BER_PROD.PRODUKT_KNOTEN=BER_PROD_KOMB.PRODUKT_KLASSE
OR
SECURITY_BER_PROD.PRODUKT_KNOTEN='ALLE')
```

Dass dies keine empfehlenswerte Struktur ist, muss wahrscheinlich nicht mehr extra erwähnt werden.

9 Mehrere Faktentabellen im selben Schema

In vielen Data Warehouse-Strukturen kann es mehrere Faktentabellen geben, auch innerhalb desselben Data Marts. Damit stellt sich die Frage, wie in diesem Fall Einschränkungen vorzunehmen sind.

Dabei ist zu unterscheiden, ob die Einschränkungen auf die verschiedenen Fakten prinzipiell identisch sind oder nicht.

Wir gehen im Folgenden davon aus, dass die verschiedenen Fakten („Sterne“) über die semantische Schicht für die Abfragen getrennt werden können, wie dies etwa in BusinessObjects-Universen über Kontexte erfolgt.

9.1 Identische Einschränkungen auf verschiedene Fakten

Im einfacheren Fall sind die Einschränkungen identisch. Dann werden beide Fakten über analoge Pfade mit der Securitytabelle verknüpft (z.B. über die Beraterdimension, wie oben) und die Securitybedingung angewandt.

9.2 Unterschiedliche Einschränkungen

Wenn die Security auf die verschiedenen Fakten nicht identisch ist, so müssen ggf. mehrere Securitytabellen mit den jeweils notwendigen Berechtigungen befüllt und mit den jeweiligen Fakten verknüpft werden.

Das grundsätzliche Verfahren ist aber identisch mit dem oben beschriebenen und wird daher nicht mehr näher erläutert.

10 Historisierte Dimensionen mit Einschränkungen

10.1 Grundsätzliche Fragestellung

In vielen Data Warehouse-Systemen wird zwischen historisierten und nicht-historisierten Dimensionen[19] unterschieden.

In einer nicht historisierten SCD1-Dimension wird immer nur die aktuelle Version eines Dimensionssatzes vorgehalten, in einer voll-historisierten SCD2-Dimension wird die gesamte Historie vorgehalten, wobei der Satz um einen weiteren künstlichen Primärschlüssel und einen Gültigkeitszeitraum ergänzt wird. In der Faktentabelle wird dieser künstliche Schlüssel als Fremdschlüssel und damit als Referenz für den Join zur Dimension eingefügt.

Eine solche Historisierung bringt viele weitere mögliche Anforderungen an die Einschränkungen, die wir hier nicht in allen Varianten besprechen können, von denen aber einige exemplarisch genannt und behandelt werden sollen:

- Die Security soll zur aktuellen Zuordnung der Beraterstelle in der Beraterdimension erfolgen, siehe Abschnitt 10.2
- Die Security soll zur historischen Zuordnung des Beraternamens zu den Fakten erfolgen, siehe Abschnitt 10.3
- Die Security soll die gesamte Historie eines Faktensatzes der aktuellen Zuordnung des Faktensatzes zum Berater zuordnen, siehe Abschnitt 10.4.

10.2 Security auf die aktuellen Zuordnungen der Beraterstelle

Dies ist die SCD1-Sicht der Security: Die aktuellen Zuordnungen bestimmen die Sichtbarkeit. Im folgenden Beispiel erfolgt die Referenz auf die Stelle und nicht der Beraternamen.

Ein konkretes Anwendungsbeispiel ist: Die Beraterorganisation ändert sich im Lauf der Zeit (Stellen werden neu-und umbesetzt, Team- und Bereichsleitungen ändern sich usw.), wobei aber die Stellen als solche erhalten bleiben.

Um eine bessere Vergleichbarkeit zu erreichen, erfolgt die Zuordnung aller Faktensätze nach den aktuellen Strukturen der Stellen in der Organisationshierarchie.

Konkret erlaubt dies, bei Gebietsneustrukturierungen, bei denen Stellen verschoben werden, den echten Erfolg oder Misserfolg der Maßnahme zu ermitteln, weil alle Altsätze auf die neue Struktur projiziert werden.

In diesem Sinne müsste ein neuer Stelleninhaber, Teamleiter oder Bereichsleiter auch alle entsprechenden Altsätze sehen dürfen, um eine Vergleichbarkeit der Ergebnisse zu erhalten.

Dies ist genau über die bisher beschriebenen Verfahren zu bewerkstelligen, da diese alle nicht-historische Dimensionen verwendet haben. Wir müssen allerdings darauf achten, dass die Zuordnung auf Stellenebene erfolgt, also mit einer Dimension wie in Abschnitt 6 beschrieben:

BERATER_ PK	BERATER_ STELLE	BERATER	TEAM	TEAM_ STELLE	TEAM-LEITER	BEREICH_ STELLE	BEREICH	BEREICHS-LEITER
1	P	Newton	Leitung Physiker	P	Newton	P	Physiker	Newton
2	P_1	Kopernikus	Klassische Physiker	P_1	Kopernikus	P	Physiker	Newton
3	P_1_1	Kepler	Klassische Physiker	P_1	Kopernikus	P	Physiker	Newton
4	P_1_2	Galilei	Klassische Physiker	P_1	Kopernikus	P	Physiker	Newton
…	…	…	…	…	…	…	…	…

[19] nach Kimball auch als Slowly Changing Dimension Typ 1 und 2 oder kurz SCD1 und SCD2 genannt.

Würden beispielsweise der Bereichsleiter Newton und der Berater Kepler die Stellen tauschen, so wäre das Resultat danach das folgende:

BERATER_ PK	BERATER_ STELLE	BERATER	TEAM	TEAM_ STELLE	TEAM-LEITER	BEREICH _ STELLE	BEREICH	BEREICHS-LEITER
1	P	Kepler	Leitung Physiker	P	Kepler	P	Physiker	Kepler
2	P_1	Kopernikus	Klassische Physiker	P_1	Kopernikus	P	Physiker	Kepler
3	P_1_1	Newton	Klassische Physiker	P_1	Kopernikus	P	Physiker	Kepler
4	P_1_2	Galilei	Klassische Physiker	P_1	Kopernikus	P	Physiker	Kepler
…	…	…	…	…	…	…	…	…

Kepler könnte nun über entsprechende Zuordnungen in der Securitytabelle den gesamten Bereich sehen und Newton nur noch die Daten der Stelle P_1_1, und zwar jeweils die gesamte Historie der zugehörigen Fakten. Die Einträge in die Securitytabelle können in diesem Fall sowohl auf Namens-. als auch auf Stellenebene erfolgen, weil diese in einer SCD1-Dimension gleichwertig sind.

Wichtig ist hier aber, dass die historischen Fakten, die von Kepler auf der Stelle P_1_1 und von Newton auf der Stelle P erzeugt wurden, nun vollständig den neuen Stelleninhabern zugeordnet werden, weil die Faktenzuordnung nicht über den Namen, sondern über die Stelle erfolgt ist.

10.3 Security auf die historische Zuordnung des Beraternamens

Diese Sicht erfordert eine SCD2-historisierte Dimension und wird beispielsweise gerne für die Sicht auf Provisionen verwendet.

Als Beispiel gehen wir davon aus, dass jeder Berater für einen abgeschlossenen Vertrag oder Verkauf eine Provision erhält. Diese Provision ist nicht abhängig von der aktuellen hierarchischen Zuordnung, sondern von der historischen: Wenn beispielsweise Galilei das Team wechselt und somit seine Stelle neu besetzt wird, so bleiben die bisherigen Provisionen dem bisherigen Berater Galilei (inklusive der damaligen Organisationsstruktur) zugeordnet und dürfen auch nicht für den neuen Berater sichtbar sein. Umgekehrt darf Galilei auf der neuen Stelle nur die neu von ihm erzeugten Sätze mit Provision sehen, aber nicht die bisherigen Sätze der Stelle.

Daher darf die Einschränkung nicht auf die Stelle erfolgen, sie muss hier auf den Berater erfolgen.

Wir gehen davon aus, dass die Berater-Informationen in der SCD2-historisierten Dimension gepflegt werden.

Betrachten wir zunächst einige Sätze in der Dimension (um Verwechslungen zu vermeiden, verwenden wir neue Daten; BERATER_SIK[20] ist der historische künstliche Primärschlüssel der Tabelle):

BERATER_SIK	GUE_VON	GUE_BIS	STELLE	BERATER	STADT	TEAMLEITER	BEREICHSLEITER
1	1.1.2010	31.12.2011	B111	Galilei	Padua	Kopernikus	Newton
2	1.1.2010	31.12.2012	B222	Kepler	Berlin	Kopernikus	Newton
3	1.1.2010	31.05.2013	B333	Brahe	Prag	Kopernikus	Newton
4	1.1.2010	31.05.2013	D444	Leibnitz	Berlin	Kant	Sokrates
…	…	…	…	…		…	…
78	1.1.2012	5.1.2012	B111	Galilei	Rom	Kopernikus	Newton
…	…	…	…	…		…	…

[20] SIK steht für Surrogate Integer Key, also künstlicher Integer-Schlüssel

81	6.1.2012	31.05.2013	B111	Galilei	Padua	Kopernikus	Newton
...	...	...	...	...		...	...
97	1.1.2013	31.05.2013	B222	Kepler	Prag	Kopernikus	Newton
...	...	...	...	...		...	...
103	1.6.2013	31.12.9999	B111	Leibnitz	Berlin	Galilei	Newton
104	1.6.2013	31.12.9999	B222	Laplace	Paris	Galilei	Newton
105	1.6.2013	30.6.2013	B333	Brahe	Prag	Galilei	Newton
106	1.6.2013	31.12.9999	C555	Kepler	Prag	Kopernikus	Euklid
107	1.6.2013	31.12.9999	D444	Plato	Berlin	Kant	Sokrates
...	...	...	...	...		...	...
314	1.7.2013	31.12.9999	B333	Bruno	Rom	Galilei	Newton

- Die ersten vier Sätze sind die initial angelegten der vier Beraterstellen B111, B222, B333 und D444.
- Am 1.1.2012 ist Galilei umgezogen, hat aber die Stelle unverändert behalten, am 6.1.2012 erneut (Sätze 78 und 81).
- Am 1.1.2013 ist Kepler umgezogen, ohne die Stelle sonst zu ändern (Satz 97).
- Am 1.6.2013 gab es eine massive Umstrukturierung:
    - Galilei hat Kopernikus als Teamleiter ersetzt (Sätze 103-105).
    - Leibnitz hat die Stelle von Galilei übernommen (Satz 103).
    - Laplace hat die Stelle von Kepler übernommen (Satz 104).
    - Kepler ist mit Kopernikus als Teamleiter in den Bereich Euklid gewechselt (Satz 106).
    - Plato hat die Stelle von Leibnitz übernommen (Satz 107).
- Am 1.7.2013 übernimmt Bruno die Stelle von Brahe (Satz 314).

Betrachten wir dazu die Faktentabelle, in der Versicherungsverträge und ihre Änderungen mit den jeweiligen Prämienänderungen und den für die Änderung zuständigen Beratern eingetragen werden:

SATZNR	VERTRAGSNR	DATUM	BERATER_SIK	BERATER_STELLE	PRAEMIE
1	V0001	03.01.2010	1	B111	316,42
2	V0002	04.01.2010	3	B333	278,73
3	V0004	07.07.2010	2	B222	949,18
4	V0002	08.07.2010	1	B111	441,1
5	V0005	01.12.2010	4	D444	1200,41
6	V0005	01.03.2011	4	D444	825,6
7	V0006	07.04.2011	3	B333	974,56
8	V0007	10.05.2011	4	D444	1100,85
9	V0004	09.08.2011	2	B222	481,22
10	V0001	13.10.2011	1	B111	1017,45
11	V0006	01.01.2012	3	B333	414,15
12	V0001	04.01.2013	78	B111	577,94
13	V0008	31.03.2012	2	B222	365,62
14	V0003	28.08.2012	81	B111	497,6
15	V0005	30.11.2012	4	D444	641,86
16	V0009	09.02.2013	3	B333	872,76
17	V0008	22.04.2013	97	B222	1269,59

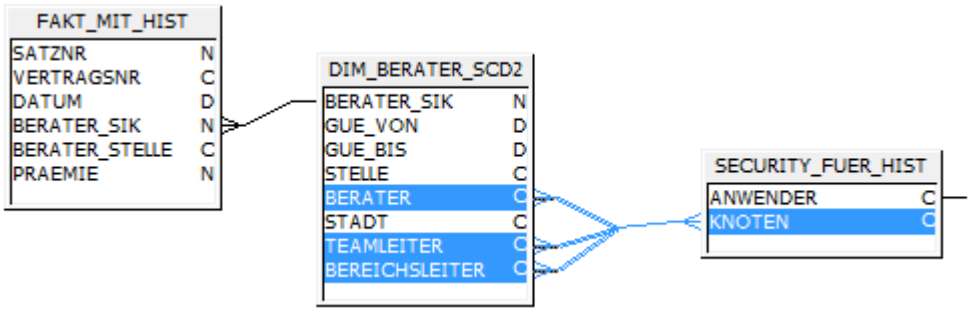
18	V0002	01.05.2013	81	B111	928,43
19	V0010	22.06.2013	106	C555	1610,22
20	V0004	22.06.2013	105	B333	773,82
21	V0008	22.06.2013	104	B222	1380,71
22	V0011	24.06.2013	106	C555	669,85
23	V0001	01.07.2013	103	B111	623,83
24	V0012	07.07.2013	314	B333	815,33
25	V0005	09.07.2013	107	D444	278,8

Als Security verwenden wir diejenige mit Knotenzuweisung, bei der wir aktuell jedem Berater als Anwender seinen eigenen Knoten zuweisen:

ANWENDER	KNOTEN
Galilei	Galilei
Kepler	Kepler
Brahe	Brahe
Leibnitz	Leibnitz
Laplace	Laplace
Plato	Plato
Bruno	Bruno
Kopernikus	Kopernikus
Kant	Kant
Newton	Newton
Euklid	Euklid
Sokrates	Sokrates

Interessanterweise verwenden wir hier keine Historie, vielmehr wird das zeitunabhängige Attribut (Knotenname) verwendet, um alle zeitabhängigen Sätze zuzuordnen.

Die Verknüpfung ist die übliche:



mit der Joinbedingung

```
SECURITY_FUER_HIST.KNOTEN=DIM_BERATER_SCD2.BERATER
OR
SECURITY_FUER_HIST.KNOTEN=DIM_BERATER_SCD2.TEAMLEITER
OR
SECURITY_FUER_HIST.KNOTEN=DIM_BERATER_SCD2.BEREICHSLEITER
OR
SECURITY_FUER_HIST.KNOTEN='ALLE'
```

Als Vergleich erzeugen wir zunächst wieder einen Bericht ohne Einschränkung:

BEREICHS-LEITER	TEAM-LEITER	STADT	BERATER	STELLE	SATZNR	VERTRAGS-NR	DATUM	PRAEMIE
Euklid	Kopernikus	Prag	Kepler	C555	19	V0010	22.06.13	1.610,22
Euklid	Kopernikus	Prag	Kepler	C555	22	V0011	24.06.13	669,85
Newton	Galilei	Berlin	Leibnitz	B111	23	V0001	01.07.13	623,83
Newton	Galilei	Paris	Laplace	B222	21	V0008	22.06.13	1.380,71
Newton	Galilei	Prag	Brahe	B333	20	V0004	22.06.13	773,82
Newton	Galilei	Rom	Bruno	B333	24	V0012	07.07.13	815,33
Newton	Kopernikus	Berlin	Kepler	B222	3	V0004	07.07.10	949,18
Newton	Kopernikus	Berlin	Kepler	B222	9	V0004	09.08.11	481,22
Newton	Kopernikus	Berlin	Kepler	B222	13	V0008	31.03.12	365,62
Newton	Kopernikus	Padua	Galilei	B111	1	V0001	03.01.10	316,42
Newton	Kopernikus	Padua	Galilei	B111	4	V0002	08.07.10	441,1
Newton	Kopernikus	Padua	Galilei	B111	10	V0001	13.10.11	1.017,45
Newton	Kopernikus	Padua	Galilei	B111	14	V0003	28.08.12	497,6
Newton	Kopernikus	Padua	Galilei	B111	18	V0002	01.05.13	928,43
Newton	Kopernikus	Prag	Brahe	B333	2	V0002	04.01.10	278,73
Newton	Kopernikus	Prag	Brahe	B333	7	V0006	07.04.11	974,56
Newton	Kopernikus	Prag	Brahe	B333	11	V0006	01.01.12	414,15
Newton	Kopernikus	Prag	Brahe	B333	16	V0009	09.02.13	872,76
Newton	Kopernikus	Prag	Kepler	B222	17	V0008	22.04.13	1.269,59
Newton	Kopernikus	Rom	Galilei	B111	12	V0001	04.01.13	577,94
Sokrates	Kant	Berlin	Leibnitz	D444	5	V0005	01.12.10	1.200,41
Sokrates	Kant	Berlin	Leibnitz	D444	6	V0005	01.03.11	825,6
Sokrates	Kant	Berlin	Leibnitz	D444	8	V0007	10.05.11	1.100,85
Sokrates	Kant	Berlin	Leibnitz	D444	15	V0005	30.11.12	641,86
Sokrates	Kant	Berlin	Plato	D444	25	V0005	09.07.13	278,8

Hieran wird schon deutlich, wie gleich die Security wirken wird, die Regel lautet: Wenn in einer der Spalten der eingeschränkte Knoten vorkommt, so ist der Satz sichtbar. Das ist sehr ähnlich zum oben beschriebenen Verhalten und das ist auch kein Zufall: Das Ziel war nicht, eine neue Art von Security einzuführen, sondern die Regeln für die Sichtbarkeit der Sätze anzupassen.

Als Beispiel die Ansicht für den Anwender Kepler:

BEREICHS-LEITER	TEAM-LEITER	STADT	BERATER	STELLE	SATZNR	VERTRAGS-NR	DATUM	PRAEMIE
Euklid	Kopernikus	Prag	Kepler	C555	19	V0010	22.06.13	1.610,22
Euklid	Kopernikus	Prag	Kepler	C555	22	V0011	24.06.13	669,85
Newton	Kopernikus	Berlin	Kepler	B222	3	V0004	07.07.10	949,18

Newton	Kopernikus	Berlin	Kepler	B222	9	V0004	09.08.11	481,22
Newton	Kopernikus	Berlin	Kepler	B222	13	V0008	31.03.12	365,62
Newton	Kopernikus	Prag	Kepler	B222	17	V0008	22.04.13	1.269,59

Im Gegensatz dazu ergibt eine Einschränkung auf die Stelle B222 (die ursprüngliche Stelle von Kepler) folgende Resultate:

BEREICHS-LEITER	TEAM-LEITER	STADT	BERATER	STELLE	SATZNR	VERTRAGS-NR	DATUM	PRAEMIE
Newton	Galilei	Paris	Laplace	B222	21	V0008	22.06.13	1.380,71
Newton	Kopernikus	Berlin	Kepler	B222	3	V0004	07.07.10	949,18
Newton	Kopernikus	Berlin	Kepler	B222	9	V0004	09.08.11	481,22
Newton	Kopernikus	Berlin	Kepler	B222	13	V0008	31.03.12	365,62
Newton	Kopernikus	Prag	Kepler	B222	17	V0008	22.04.13	1.269,59

Dies entspricht der nicht-historischen Ansicht, die in Abschnitt 10.2 erläutert wurde!

Es geht hier nicht darum, die eine oder andere Sicht für korrekt zu erklären, vielmehr muss die benötigte Version an den Anforderungen festgemacht werden. Es kann sogar sein, dass beide Versionen parallel einzusetzen sind, was beispielsweise über zwei getrennt zu behandelnde Aliasse der Faktentabelle mit jeweils eigener Security realisiert werden kann.

10.4 Zuordnung der Faktenhistorie zur aktuellen Betreuung

Wenn wir das Beispiel aus dem vorigen Abschnitt betrachten, sehen wir, dass sich die betreuende Stelle der Verträge V0002 und V0004 im Lauf der Zeit geändert hat:

BEREICHS-LEITER	TEAM-LEITER	STADT	BERATER	STELLE	SATZNR	VERTRAGS-NR	DATUM	PRAEMIE
Newton	Kopernikus	Padua	Galilei	B111	4	V0002	08.07.10	441,1
Newton	Kopernikus	Padua	Galilei	**B111**	18	V0002	01.05.13	928,43
Newton	Kopernikus	Prag	Brahe	**B333**	2	V0002	04.01.10	278,73
Newton	Galilei	Prag	Brahe	**B333**	20	V0004	22.06.13	773,82
Newton	Kopernikus	Berlin	Kepler	**B222**	3	V0004	07.07.10	949,18
Newton	Kopernikus	Berlin	Kepler	B222	9	V0004	09.08.11	481,22

Aus Betreuungssicht sollte aber der Berater, der einen Vertrag aktuell betreut, auch die gesamte Vertragshistorie sehen können, weil ansonsten eine effektive Kundenberatung unmöglich ist.

Am obigen Beispiel wird aber schon sichtbar, dass weder eine Einschränkung auf Beraterknoten, noch eine auf die Stelle diesen gewünschten Effekt hat.

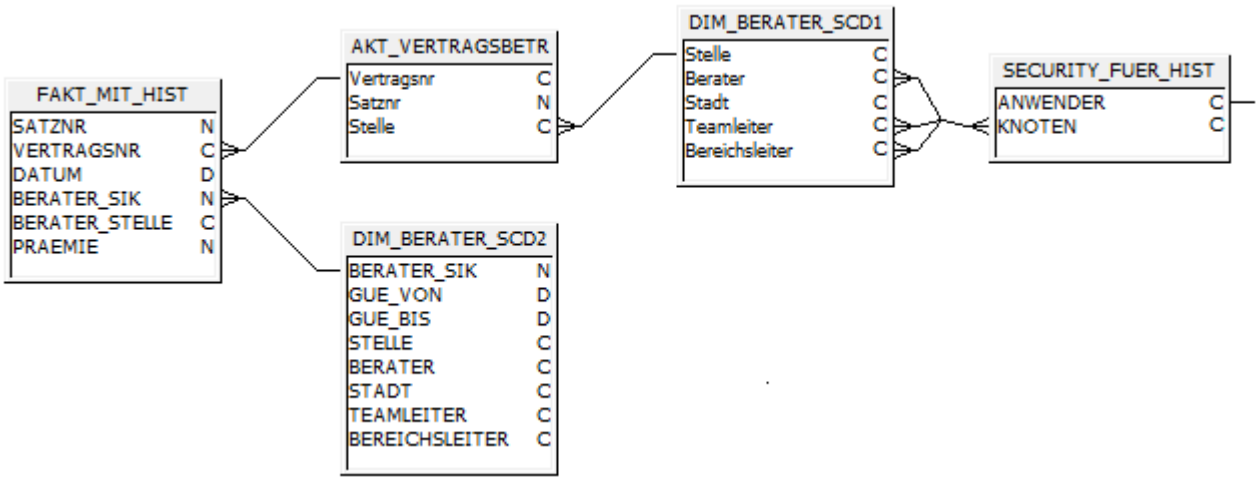
Dazu muss zunächst in einem Zwischenschritt für jeden Vertrag die aktuell gültige betreuende Stelle ermittelt werden, wozu in diesem Beispiel das letzte Datum des Vertrags oder die höchste Satznummer als Referenz der letzten Änderung herangezogen werden können:

VERTRAGSNR	SATZNR	STELLE
V0001	23	B111
V0002	18	B111
V0003	14	B111
V0004	20	B333
V0005	25	D444
V0006	11	B333
V0007	8	D444
V0008	21	B222
V0009	16	B333
V0010	19	C555
V0011	22	C555
V0012	24	B333

Diese Tabelle soll benutzt werden, um die aktuelle Betreuungssituation wiederzugeben (sogar unabhängig von der Security!), daher muss sie mit der nicht historischen SCD1-Dimension verknüpft werden, die folgende Einträge hat:

STELLE	BERATER	STADT	TEAMLEITER	BEREICHSLEITER
B111	Leibnitz	Berlin	Galilei	Newton
B222	Laplace	Paris	Galilei	Newton
C555	Kepler	Prag	Kopernikus	Euklid
D444	Plato	Berlin	Kant	Sokrates
B333	Bruno	Rom	Galilei	Newton

Die Gesamtstruktur ist dann die folgende:



Auf diese Weise kann sogar die historische Betreuungssituation des Vertrags aus aktueller Sicht dargestellt werden, hier zunächst wieder ohne Security (sortiert nach Vertragsnummer und darin nach Satznummer):

SCD1-ATTRIBUTE – AKUELLER BEZUG			SCD2-ATTRIBUTE – HISTORIE			FAKTENINFORMATION			
BEREICHS-LEITER	**TEAM-LEITER**	**BERATER**	**BEREICHS-LEITER**	**TEAM-LEITER**	**BERATER**	**VERTRAGS-NR**	**SATZ-NR**	**DATUM**	**PRAEMIE**
Newton	Galilei	Leibnitz	Newton	Kopernikus	Galilei	V0001	1	03.01.2010	316,42
Newton	Galilei	Leibnitz	Newton	Kopernikus	Galilei	V0001	10	13.10.2011	1.017,45
Newton	Galilei	Leibnitz	Newton	Kopernikus	Galilei	V0001	12	04.01.2013	577,94
Newton	Galilei	Leibnitz	Newton	Galilei	Leibnitz	V0001	23	01.07.2013	623,83
Newton	Galilei	Leibnitz	Newton	Kopernikus	Brahe	V0002	2	04.01.2010	278,73
Newton	Galilei	Leibnitz	Newton	Kopernikus	Galilei	V0002	4	08.07.2010	441,1
Newton	Galilei	Leibnitz	Newton	Kopernikus	Galilei	V0002	18	01.05.2013	928,43
Newton	Galilei	Leibnitz	Newton	Kopernikus	Galilei	V0003	14	28.08.2012	497,6
Newton	Galilei	Bruno	Newton	Kopernikus	Kepler	V0004	3	07.07.2010	949,18
Newton	Galilei	Bruno	Newton	Kopernikus	Kepler	V0004	9	09.08.2011	481,22
Newton	Galilei	Bruno	Newton	Galilei	Brahe	V0004	20	22.06.2013	773,82
Sokrates	Kant	Plato	Sokrates	Kant	Leibnitz	V0005	5	01.12.2010	1.200,41
Sokrates	Kant	Plato	Sokrates	Kant	Leibnitz	V0005	6	01.03.2011	825,6
Sokrates	Kant	Plato	Sokrates	Kant	Leibnitz	V0005	15	30.11.2012	641,86
Sokrates	Kant	Plato	Sokrates	Kant	Plato	V0005	25	09.07.2013	278,8
Newton	Galilei	Bruno	Newton	Kopernikus	Brahe	V0006	7	07.04.2011	974,56
Newton	Galilei	Bruno	Newton	Kopernikus	Brahe	V0006	11	01.01.2012	414,15
Sokrates	Kant	Plato	Sokrates	Kant	Leibnitz	V0007	8	10.05.2011	1.100,85
Newton	Galilei	Laplace	Newton	Kopernikus	Kepler	V0008	13	31.03.2012	365,62
Newton	Galilei	Laplace	Newton	Kopernikus	Kepler	V0008	17	22.04.2013	1.269,59
Newton	Galilei	Laplace	Newton	Galilei	Laplace	V0008	21	22.06.2013	1.380,71
Newton	Galilei	Bruno	Newton	Kopernikus	Brahe	V0009	16	09.02.2013	872,76
Euklid	Kopernikus	Kepler	Euklid	Kopernikus	Kepler	V0010	19	22.06.2013	1.610,22
Euklid	Kopernikus	Kepler	Euklid	Kopernikus	Kepler	V0011	22	24.06.2013	669,85
Newton	Galilei	Bruno	Newton	Galilei	Bruno	V0012	24	07.07.2013	815,33

Die Einschränkung erfolgt nach den SCD1-Attributen. Der Anwender Leibnitz würde also folgende Sätze sehen:

SCD1 – AKTUELLER BEZUG			SCD2 - HISTORIE			FAKTENINFORMATION			
BEREICHS-LEITER	**TEAM-LEITER**	**BERATER**	**BEREICHS-LEITER**	**TEAM-LEITER**	**BERATER**	**VERTRAGS-NR**	**SATZ-NR**	**DATUM**	**PRAEMIE**
Newton	Galilei	Leibnitz	Newton	Kopernikus	Galilei	V0001	1	03.01.2010	316,42
Newton	Galilei	Leibnitz	Newton	Kopernikus	Galilei	V0001	10	13.10.2011	1.017,45
Newton	Galilei	Leibnitz	Newton	Kopernikus	Galilei	V0001	12	04.01.2013	577,94
Newton	Galilei	Leibnitz	Newton	Galilei	Leibnitz	V0001	23	01.07.2013	623,83

Newton	Galilei	Leibnitz	Newton	Kopernikus	Brahe	V0002	2	04.01.2010	278,73
Newton	Galilei	Leibnitz	Newton	Kopernikus	Galilei	V0002	4	08.07.2010	441,1
Newton	Galilei	Leibnitz	Newton	Kopernikus	Galilei	V0002	18	01.05.2013	928,43
Newton	Galilei	Leibnitz	Newton	Kopernikus	Galilei	V0003	14	28.08.2012	497,6

Die Verträge V0002 und V0003 zeigen eine Besonderheit: Leibnitz hat für diese Verträge noch keine eigenen Änderungen vorgenommen, da er aber von Galilei die Stelle übernommen hat, sieht er alle der Stelle zugeordneten Verträge und damit auch diese beiden. Dafür sieht er nicht mehr die Verträge V0005 und V0007, obwohl er an ihnen historisch beteiligt war: Deren Betreuung ist auf einen andern Berater übergegangen.

11 Automatisierte Ermittlung der Securitytabelle

In den obigen Beispielen waren die Securitytabellen sehr eng mit der Organisationshierarchie verknüpft und konnten fast vollständig (bis auf Sonderanwender wie den Vorstand) aus dieser abgeleitet werden. Sollte dies möglich sein, so sollte dies im Rahmen des ETL-Prozesses auch geschehen.

Hier kommt auch das in Abschnitt 6 vorgestellte Rollenkonzept zum Tragen: Es kann beispielsweise vorkommen, dass die Rollen- nicht aber die endgültigen Anwenderberechtigungen aus der Organisationshierarchie hergeleitet werden können.

Die Zuordnung zu den Rollen ist aber evtl. aus einer anderen zentralen Quelle wie LDAP oder AD[21] verfügbar. Sollte dies zugänglich sein, so kann es ebenfalls im ETL-Prozess verwendet werden.

Automatisierte Prozesse sind vorzuziehen, weil sie Fehler in der manuellen Pflege verringern.

[21] LDAP = Lightweight Directory Access Protocol; AD = Active Directory; beide können zur unternehmensweiten Anwenderauthentifizierung und ggf, für Single-Sign-On (SSO) verwendet werden.

12 Optimierung der Abfrageperformance

Im Dokument wurde oft auf die Notwendigkeit der Verbesserung der Abfrageperformance hingewiesen. Die obigen Securityeinschränkungen können insbesondere dann zu Problemen führen, wenn Knoteneinschränkungen verwendet werden, weil dann mehrere Bedingungen mit OR verknüpft werden; im obigen Beispiel waren es drei für die Hierarchiestufen und eine für die ALLE-Einschränkung, es können aber je nach einzuschränkender Tabelle und Anzahl Hierarchiestufen auch sehr viel mehr sein.

Selbst wenn immer eine Einschränkung auf den angemeldeten Anwender erfolgt, kann der Indexzugriff dadurch unmöglich werden.

In den Abschnitten 5.2.3, 5.2.4 und 5.2.5, sowie später in 7.4 wurde die Auflösung der Security beschrieben, um Mehrfachzählungen zu vermeiden, aber auch, um die maximal erlaubten Details zu ermitteln. Dort ist dies aber mit abgeleiteten Tabellen, also über Abfragelogik geschehen, was aus den genannten Gründen schlecht sein kann.

Ein alternativer Weg ist, diese Abfrage zu materialisieren, also fertig aufgelöst in physische Tabellen zu schreiben. Allerdings kann dies nicht mehr zum Abfragezeitpunkt geschehen, daher muss die Auflösung für alle Anwender geschehen.

Nun darf nicht vergessen werden, dass dennoch immer eine Einschränkung auf genau einen Anwender erfolgt. Daher sollte ein Index auf die Spalte Anwender gelegt werden, ggf. kann sogar danach partitioniert werden, so dass genau eine Partition in jeder Abfrage verwendet wird. Die letzte Option ist insbesondere für sehr große Organisationen, die jedem Anwender mehrere hundert Sätze zuweisen und zu insgesamt mehreren Millionen aufgelösten Securityzeilen führen können, sehr interessant.

Es bleibt trotzdem zu bemerken, dass insbesondere für Anwender mit einer großen Zahl von zugewiesenen Knoten (Bereichsleiter, Vorstände,...) die Einschränkung sehr ineffektiv werden kann, da nur eine sehr schwache Auswahl erfolgt.

Daher eine weitere Empfehlung: Die Auflösung sollte bereits alle relevanten Dimensionsattribute enthalten. Auf diese Weise muss kein Join von der aufgelösten Security auf die Dimension erfolgen, diese ist bereits vollständig vorhanden. Insbesondere, wenn die Detailsecurity verwendet wird (Abschnitt 7), sollten die CASE-Statements fertig aufgelöst werden, so dass diese Berechnungen nicht mehr zum Abfragezeitpunkt geschehen müssen, sondern für alle Anwender fertig aufgelöst vorliegen. In diesem Fall ist die Partitionierung besonders sinnvoll.

Diese vorherige Auflösung sollte aber auf keinen Fall für die Fakteninformationen mit Detailsecurity (siehe Abschnitt 7.5.3) verwendet werden! Die Berechnungen auf die bereits selektierten Faktensätze sind auch nicht so kritisch wie die auf die Dimensionssätze.

Haben Sie Fragen zu unserem White-Paper? Unser Autor steht Ihnen gerne zur Verfügung!

Zu Business Intelligence gibt es viel zu sagen und zu schreiben.
Weitere White-Paper finden Sie unter:

www.areto-consulting.de

Autor: Felix Krul
Tel.: +49 221 66 95 75-0
felix.krul@areto-consulting.de