Brücken bauen im dimensionalen Modell

Dani Schnider
Trivadis AG
Zürich/Glattbrugg, Schweiz

Schlüsselworte:

Data Warehouse, Data Mart, Star Schema, Dimensionale Modellierung, Bridge Tables, Multi-Valued Dimensions, Rekursive Hierarchien

Einleitung

Bridge Tables werden in der dimensionalen Modellierung verwendet, um Dimensionen mit Mehrfachattributen (Multi-Valued Dimensions) oder rekursive Hierarchien in einer Dimension abzubilden. Diese Erweiterung des Star Schemas ist zwar mächtig, aber auch komplex in der Anwendung. Anhand von konkreten Beispielen wird hier erläutert, wie Bridge Tables modelliert, geladen und abgefragt werden können. Außerdem wird aufgezeigt, warum Bridge Tables nicht in jedem Fall die beste Lösung sind, wo ihre Risiken liegen und wie diese durch geeignete Alternativen vermieden werden können.

Mehrfachattribute im Star Schema

Nehmen wir an, die DOAG möchte Auswertungen über die Anzahl Teilnehmer an den einzelnen Vorträgen an der DOAG-Konferenz machen und erstellt dafür ein Star Schema mit verschiedenen Dimensionen, unter anderem einer Dimension DIM_SESSION, in welcher die verschiedenen Sessions (Vorträge) aufgeführt sind. Die Erstellung eines solchen Star Schemas stellt kein Problem dar, mit Ausnahme eines kleinen, aber aus Modellierungssicht unschönen Details: Es gibt Vorträge mit mehr als einem Referenten.

Wie kann ein solcher Sachverhalt in einem dimensionalen Datenmodell abgebildet werden? Wie immer gibt es mehrere Möglichkeiten. Die Namen der Referenten als komma-separierte Liste in einem Attribut abzuspeichern, ist eine davon. Diese nicht sehr elegante Lösung ist aber schwerfällig für die Abfragen. Andere Varianten sind mehrere Attribute (SPEAKER_1, SPEAKER_2, SPEAKER_3) in der Dimensionstabelle oder eine separate Dimension DIM_SPEAKER, die aus der Faktentabelle mehrfach referenziert wird. Nachteil dieser Lösungen – neben den ebenfalls nicht ganz trivialen Abfragen – ist die Beschränkung auf eine maximale Anzahl von Referenten. Ein pragmatischer Ansatz besteht darin, pro Vortrag einen Hauptreferenten zu definieren und nur diesen in der Dimensionstabelle zu speichern. Diese Lösung ist zwar einfach zu realisieren, führt aber zu fehlenden Informationen bei den Auswertungen.

Multi-Valued Bridge Tables

Eine vollständige und einfache Lösung für die Abbildung von Mehrfachattributen ist in einem klassischen Star Schema mit Dimensions- und Faktentabellen nicht möglich. Um solche Datenbestände abzubilden, kann aber eine weitere Art von Tabellen verwendet werden: die Bridge Table. Wie der Name besagt, bildet eine Bridge Table eine Brücke zwischen zwei Dimensionen oder

zwischen einer Dimensions- und einer Faktentabelle. Diese beiden Möglichkeiten werden anhand unseres Beispiels mit den DOAG-Vorträgen genauer erläutert.

Um das Beispiel zu illustrieren, wurde eine Reihe von Vorträgen aus dem Stream „DWH & BI“ der DOAG-Konferenz 2012 ausgewählt (in Wirklichkeit gibt es natürlich noch viele weitere interessante Vorträge). Die für unser Problem mit den Mehrfachattributen interessanteste Session ist dabei der Vortrag „Oracle Essbase Backup & Recovery“. Warum? Weil der Vortrag von zwei Referenten, Holger Huck und Mircea Bobei, gehalten wird. Um im dimensionalen Datenmodell Sessions mit zwei (oder mehr) Referenten abbilden zu können, wird die Dimension DIM_SESSION durch eine Bridge Table sowie eine zusätzliche Dimensionstabelle erweitert, wie in Abbildung 1 gezeigt.

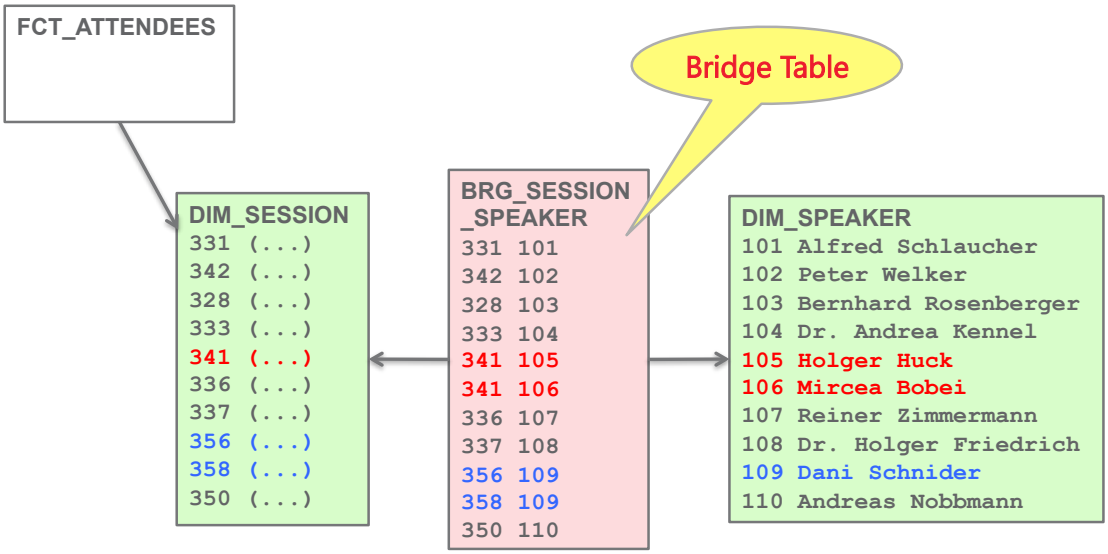

Abb. 1: Beispiel mit Multi-Valued Attribute Bridge Table

Zusätzlich zur Dimensionstabelle DIM_SESSION wird eine weitere Dimensionstabelle DIM_SPEAKER angelegt, in welcher sämtliche Referenten der DOAG-Konferenz (hier nur ein Ausschnitt) abgespeichert wird. Durch die Bridge Table BRG_SESSION_SPEAKER werden die n-zu-n-Beziehungen zwischen DIM_SESSION und DIM_SPEAKER abgebildet, wie wir es aus der relationalen Datenmodellierung kennen. Durch diese sogenannte „Multi-Valued Attribute Bridge Table“ lassen sich sowohl Vorträge mit mehreren Referenten als auch Referenten mit mehreren Vorträgen abbilden.[1]

Eine andere Alternative besteht darin, eine „Multi-Valued Dimension Bridge Table“ zwischen Faktentabelle und Dimensionstabelle zu verwenden. Dazu ändern wir das Datenmodell unseres Beispiels so, dass die Dimensionen DIM_SESSION und DIM_SPEAKER als unabhängige Dimensionen modelliert werden (und somit separat aus der Faktentabelle referenziert werden). Um Vorträge mit mehreren Referenten abbilden zu können, wird zwischen Faktentabelle und Dimensionstabelle DIM_SPEAKER eine Bridge Table gelegt, wie in Abbildung 2 dargestellt.

[1] In einer allgemeineren Form werden solche Bridge Tables auch so modelliert, dass eine weitere n-zu-n-Beziehung zwischen der Bridge Table und der ausgehenden Dimensionstabelle (hier DIM_SESSION) besteht. Details siehe [1], Seite 205 und [2], Seite 210.

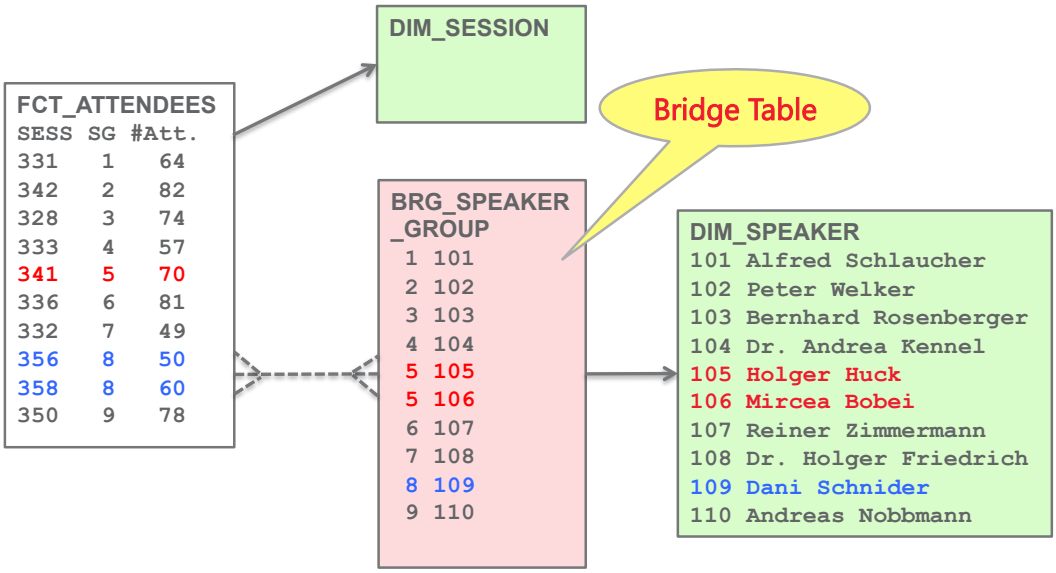

Abb. 2: Beispiel mit Multi-Valued Dimension Bridge Table

Die Einträge in der Faktentabelle FCT_ATTENDEES enthalten die Anzahl der Teilnehmer für die einzelnen Sessions (die hier aufgeführten Zahlen sind frei erfunden und entsprechen nicht der tatsächlichen Teilnehmerzahl). Die Fakten referenzieren jedoch nicht einen einzelnen Referenten in DIM_SPEAKER, sondern eine „Speaker Group“ (Attribut SG), die in der Bridge Table definiert ist. Auf diese Weise ist es ebenfalls möglich, Vorträge mit beliebig vielen Referenten abzubilden.

Zu beachten ist in diesem Beispiel die n-zu-n-Beziehung zwischen Faktentabelle und Bridge Table. Sie verhindert die Definition von Foreign Key Constraints zwischen den Tabellen. Dieses Problem kann aber gelöst werden durch eine zusätzliche Dimensionstabelle (z.B. DIM_SPEAKER_GROUP) mit nur einem Attribut, einem künstlichen Schlüssel, der dann sowohl von der Faktentabelle als auch von der Bridge Table referenziert wird.

Hohe Flexibilität und hohe Komplexität

Der Vorteil von Bridge Tables liegt in der Flexibilität: Die fachlichen Zusammenhänge mit Mehrfachattributen können vollständig abgebildet werden, und es gibt keine Limitierung der Anzahl Werte. Auch ein Vortrag mit zehn oder mehr Referenten könnte in beiden oben erwähnten Datenmodellen abgebildet werden. Ob ein 45-minütiger Vortrag mit zehn Referenten jedoch sinnvoll wäre, ist eine andere Frage, die hier nicht behandelt werden soll.

Flexibilität hat allerdings ihren Preis. Im Falle von Bridge Tables äußert sich dieser durch eine höhere Komplexität, sei es beim Datenmodell (n-zu-n-Beziehung), in der ETL-Logik oder bei den Abfragen auf das Star Schema. Bei den Abfragen müssen spezielle Vorkehrungen getroffen werden, um Mehrfachzählungen zu vermeiden, wie im nächsten Abschnitt beschrieben.

Wo liegt die zusätzliche Komplexität bei den ETL-Prozessen? Neben dem Einfügen oder Ersetzen von Dimensionseinträgen müssen auch die zugehörigen Datensätze in der Bridge Table bewirtschaftet werden. Das kann zum Beispiel heißen, dass nachträglich ein zusätzlicher Referent für einen bereits angemeldeten und ins DWH geladenen Vortrag gemeldet wird. Dies führt zu einem neuen Eintrag in der Bridge Table. Kommt die Absage eines Referenten (z.B. weil er zur Einsicht kommt, dass zehn Referenten zu viel sind), muss die entsprechende Zuordnung aus der Bridge Table gelöscht werden.

Wie bitte? Löschoperationen in einem DWH gibt es normalerweise nicht – aber bei Bridge Tables können sie durchaus zweckmäßig und notwendig sein. Die hier aufgeführten Beispiele gehen von der einfachen Annahme aus, dass keine Historisierung der Dimensionsdaten nötig ist, dass wir es also mit Slowly Changing Dimensions Typ 1 (SCD 1) zu tun haben.

Bei SCD Typ 2 wird es einiges komplexer. So hat das Einfügen einer neuen Version in der Dimensionstabelle auch die Erstellung von neuen Versionen aller zugehörigen Einträge in der Bridge Table zur Folge. Eine versionierte Bridge Table wächst dadurch typischerweise sehr rasch, da für jede Änderung eines Dimensionseintrags sämtliche Gruppenzugehörigkeiten kopiert werden müssen. Bei Änderungen von Gruppenzugehörigkeiten (z.B. nachträgliche An- und Abmeldungen von Referenten), müssen in der Bridge Table neue Versionen erstellt und teilweise bestehende Einträge kopiert werden. Bei Multi-Valued Bridge Tables müssen je nach Art der Änderung auch zusätzliche Versionen in der Dimensionstabelle eingefügt werden. Schließlich muss bei Bridge Tables in Kombination mit SCD 2 beachtet werden, dass bei Abfragen immer eine Einschränkung des Datumintervalls auf die Bridge Table notwendig ist, da sonst mehrere Versionen aus der Dimensionstabelle selektiert werden. Die Einschränkung aufgrund des Joins mit der Faktentabelle, wie sonst bei SCD2-Dimensionen üblich, genügt hier nicht.

Abfragen auf Bridge Tables

Der letzte erwähnte Punkt führt uns zu einer wesentlichen Fehlerquelle im Zusammenhang mit Bridge Tables: Mehrfachzählungen bei Abfragen. Um die Problematik zu erläutern, führen wir ein paar SQL-Abfragen auf das Beispielschema aus Abbildung 2 aus.

Zuerst möchten wir wissen, wie viele Teilnehmer jeder Referent in seinen Vorträgen hat. Die Frage lässt sich mit folgender SQL-Abfrage beantworten:

```
SELECT d.speaker_name
     , SUM(f.num_attendees)
  FROM fct_attendees f
  JOIN brg_speaker_group b ON (b.speaker_group_id = f.speaker_group_id)
  JOIN dim_speaker d ON (d.speaker_id = b.speaker_id)
GROUP BY d.speaker_name
```

Die Query liefert für alle Referenten korrekte Resultate. Dass Holger Huck und Mircea Bobei je 70 Zuhörer haben, liegt daran, dass sie einen gemeinsamen Vortrag präsentieren. Aus Sicht jedes einzelnen Referenten ist die ermittelte Anzahl Teilnehmer korrekt.

Nun möchten wir die Abfrage so ändern, dass die Anzahl der Teilnehmer nicht pro einzelnem Referent, sondern pro Firma, bei der die Referenten angestellt sind, ermittelt wird. Dieser „Drill-Up“ wird üblicherweise so realisiert, dass einfach nach einem anderen Attribut der Dimension – hier nach dem Firmennamen – aggregiert wird:

```
SELECT d.company_name
     , SUM(f.num_attendees)
  FROM fct_attendees f
  JOIN brg_speaker_group b ON (b.speaker_group_id = f.speaker_group_id)
  JOIN dim_speaker d ON (d.speaker_id = b.speaker_id)
GROUP BY d.company_name
```

Doch liefert diese SQL-Abfrage das korrekte Resultat?

In der für das Beispiel willkürlich zusammengestellten Liste von Referenten sind „zufälligerweise“ die Hälfte der Personen Trivadis-Mitarbeiter, wie Abbildung 3 zeigt.

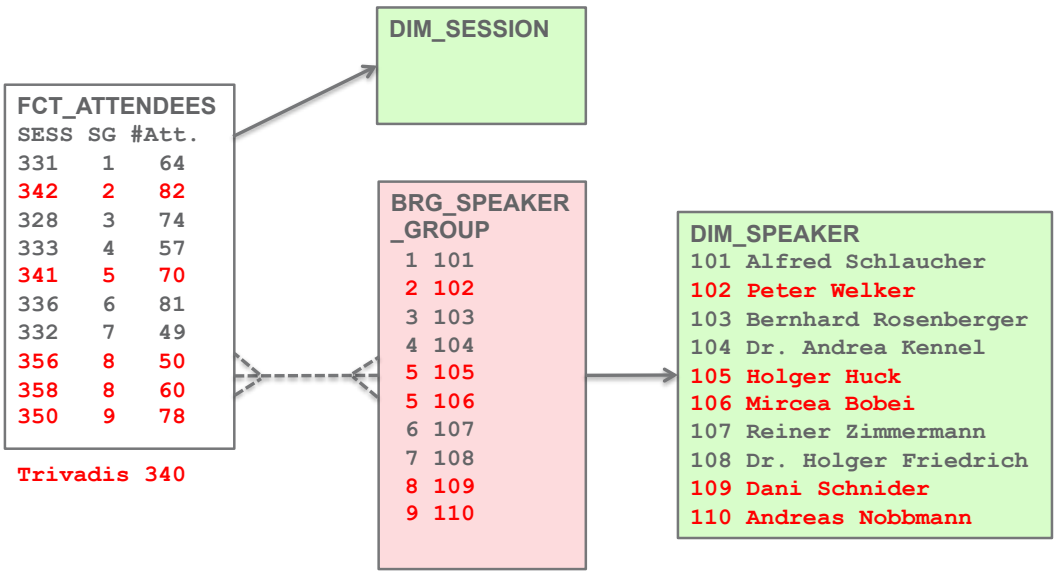

Abb. 3: Anzahl Vortragsteilnehmer der Trivadis-Referenten

Werden die (erfundenen) Teilnehmerzahlen der fünf Trivadis-Vorträge zusammengezählt, ergibt die Summe 340 Teilnehmer. Die SQL-Query gibt jedoch als Resultat die Zahl 410 zurück. Wo liegt der Fehler?

Die Ursache liegt bei der Doppelzählung der 70 Teilnehmer, die gebannt dem Vortrag von Holger Huck und Mircea Bobei folgen. Da dieser Vortrag von zwei Referenten gehalten wird, ergibt die SQL-Query für diesen Vortrag die doppelte Anzahl Teilnehmer – also 70 zu viel. Man stelle sich das Resultat bei einer Präsentation mit zehn Referenten vor...

Zur Vermeidung von Mehrfachzählungen wird in der Bridge Table ein zusätzliches Attribut mit einer Gewichtung eingeführt, wie in Abbildung 4 dargestellt. Vorträge mit einem Referenten erhalten die Gewichtung 100% (bzw. 1.0), bei Vorträgen mit mehreren Referenten wird die Gewichtung prozentual auf die Referenten verteilt – bei zwei Referenten also je 50% (bzw. 0.5).

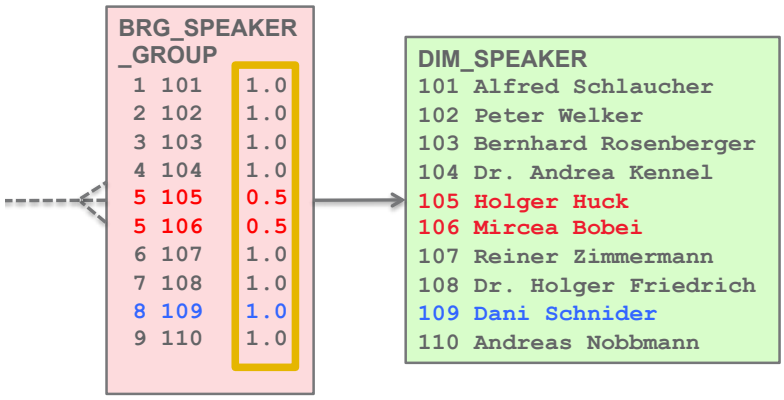

Abb. 4: Gewichtung der Zuordnungen in der Bridge Table

Diese Gewichtung wird für die Korrektur von Mehrfachzählungen bei Abfragen auf übergeordnete Aggregationsstufen (z.B. Referenten einer Firma, eines Landes oder für das Gesamttotal) verwendet:

```
SELECT d.company_name
     , SUM(f.num_attendees * b.allocation_factor)
  FROM fct_attendees f
  JOIN brg_speaker_group b ON (b.speaker_group_id = f.speaker_group_id)
  JOIN dim_speaker d ON (d.speaker_id = b.speaker_id)
GROUP BY d.company_name
```

Aber aufgepasst: Bei Abfragen auf der untersten Stufe (Teilnehmerzahl pro Referent) darf die Gewichtung nicht verwendet werden!

Vereinfachung der Abfragen

Einmal mehr zeigt sich hier das Dilemma zwischen Flexibilität und Komplexität. Für erfahrene Power-User, die unterschiedlichste Auswertungen nach verschiedenen Kriterien durchführen möchten und in der Lage sind, entsprechende Adhoc-Queries zu formulieren, bietet ein Datenmodell mit Bridge Tables zahlreiche Möglichkeiten. Doch die meisten Endanwender – und viele BI-Tools – scheitern an der Komplexität der Abfragen. Hier sind Vereinfachungen gefragt.

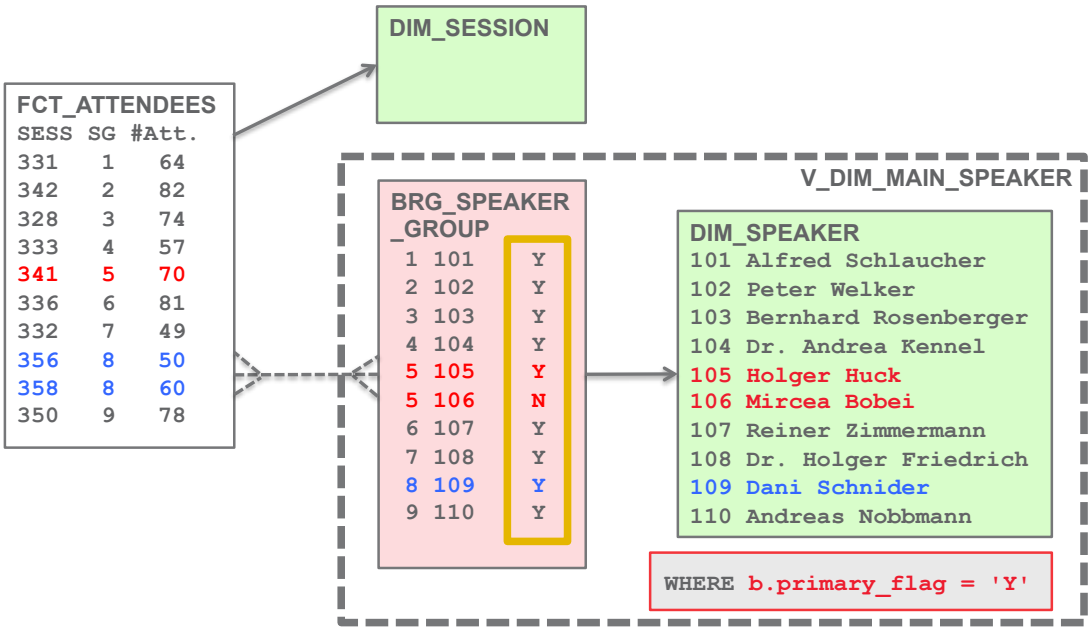

Abb. 5: Vereinfachung durch View über Bridge Table

Eine Möglichkeit zur Vereinfachung besteht darin, die Komplexität der Bridge Table hinter einer View zu verstecken. Dazu wird die Bridge Table um ein zusätzliches Attribut PRIMARY_FLAG ergänzt. Für jede Referentengruppe wird eine Person als Hauptreferent markiert. Die View schränkt nun den Datenbestand so ein, dass pro Vortrag nur der jeweilige Hauptreferent angezeigt wird, wie in Abbildung 5 dargestellt. Die meisten Endanwender arbeiten mit dieser View wie mit einer „normalen“ Dimensionstabelle. Für spezielle Auswertungen, in welchen auch die zusätzlichen Referenten gefragt sind, wird hingegen direkt auf die Bridge Table und die zugehörige Dimensionstabelle zugegriffen.

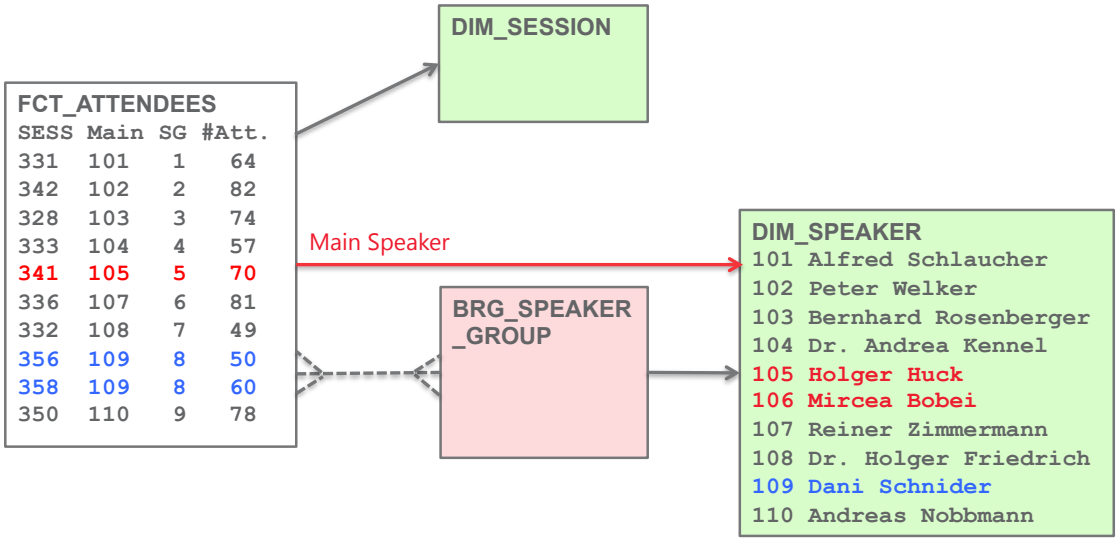

Abb. 6: Vereinfachung durch zusätzliche Beziehung auf Dimensionstabelle

Als weitere Variante kann eine zusätzliche Beziehung zwischen Faktentabelle und Dimensionstabelle definiert werden, die den Hauptreferenten jedes Vortrags identifiziert (vgl. Abbildung 6). Die Standardabfragen der Endanwender verwenden ausschließlich diese Verbindung zur Dimensionstabelle DIM_SPEAKER, während die Bridge Table nur für spezifische Abfragen durch entsprechend geschulte Power-User zur Anwendung kommt.

Rekursive Hierarchien

Wir haben uns nun ausführlich mit einem Einsatzgebiet von Bridge Tables befasst, nämlich der Abbildung von Mehrfachattributen in Dimensionen. Daneben gibt es aber noch einen weiteren typischen Anwendungsbereich: Rekursive Hierarchien, wie sie zum Beispiel in Mitarbeiter-Organigrammen, Organisationseinheiten, Stücklisten oder Kostenstellen zum Einsatz kommen. Eine rekursive Hierarchie besteht aus Dimensionseinträgen, die auf übergeordnete Dimensionseinträge (z.B. den Vorgesetzten eines Mitarbeiters) verweisen.

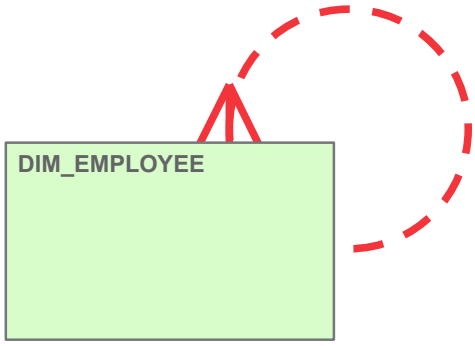

Abb. 7: Dimensionstabelle mit rekursiver Hierarchie

Typisch für solche Hierarchien ist, dass die Anzahl der Hierarchiestufen nicht fix ist. Eine flexible Möglichkeit besteht in der Implementierung mittels Self-Relationship (auch „Schweinsohr“ genannt), also einer Fremdschlüsselbeziehung auf die gleiche Tabelle, wie in Abbildung 7 dargestellt.

In Oracle SQL lassen sich darauf hierarchische Abfragen der folgenden Art ausführen:

```
SELECT emp_id, name, parent_emp_id
  FROM dim_employee
 START WITH name = 'Jones'
 CONNECT BY PRIOR emp_id = parent_emp_id
```

Neben der Einschränkung, dass diese Abfrage Oracle-spezifisch ist, besteht auch der Nachteil, dass solche Abfragen in vielen BI-Tools nicht oder nur mit erheblichem Aufwand realisiert werden können.

Ein häufig gewählter und bewährter Ansatz besteht darin, die rekursive Hierarchie als „flache" Dimensionstabelle zu implementieren und fehlende Hierarchiestufen durch Wiederholung der übergeordneten Einträge zu füllen.[2] In vielen Fällen ist diese Lösung zweckmäßig, hat aber die Eigenschaft, dass die Anzahl der Hierarchiestufen durch das Design der Dimensionstabelle beschränkt wird. Falls diese Einschränkung ein Problem darstellen sollte, lässt sich eine rekursive Hierarchie auch mit einer Bridge Table abbilden.

Hierarchy Bridge Tables

Eine Hierarchy Bridge Table ist eine Tabelle, welche für jede Kombination von Dimensionseinträgen eine Referenz auf den übergeordneten und den untergeordneten Datensatz sowie die Anzahl der Hierarchiestufen dazwischen festhält. Das Beispiel in Abbildung 8 zeigt eine Mitarbeiterdimension, welche die 14 Mitarbeiter der altbekannten EMP-Tabelle aus dem Oracle-Beispielschema SCOTT enthält. Um die gesamte Mitarbeiterhierarchie abzubilden, werden in der zugehörigen Bridge Table 39 Einträge benötigt (nicht alle davon sind hier dargestellt).

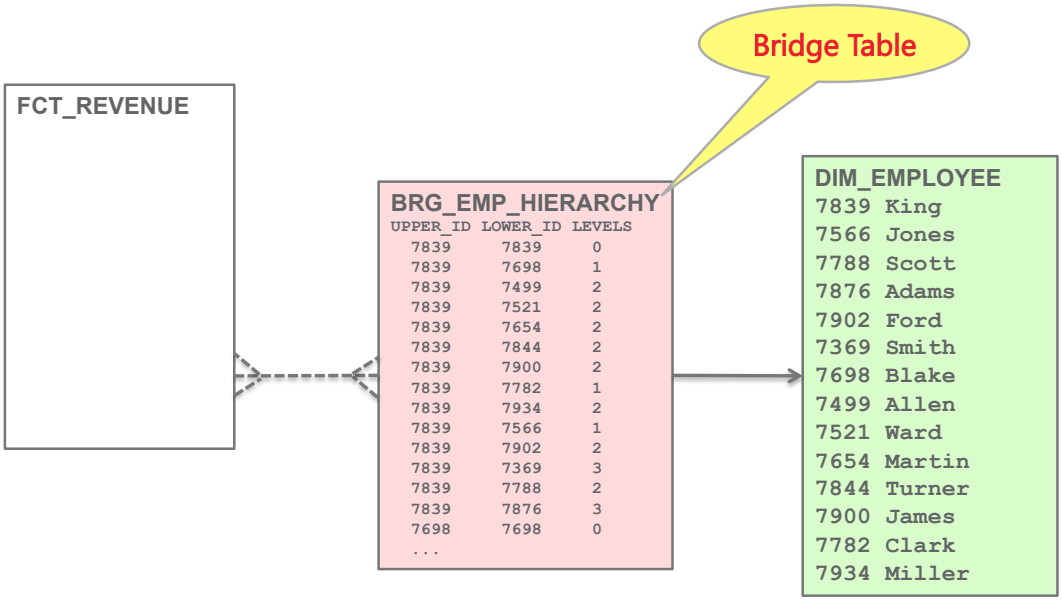

Abb. 8: Beispiel mit Hierarchy Bridge Table

Soll nun zum Beispiel der Umsatz aller Mitarbeiter ermittelt werden, die Mr. Jones unterstellt sind, lässt sich dies mit einer einfachen SQL-Abfrage formulieren:

[2] Eine anschauliches Erklärung dazu ist in [2], Seite 224 – 227 zu finden.

```
SELECT SUM(f.amount)
  FROM fct_revenue f
  JOIN brg_emp_hierarchy b ON (b.lower_id = f.emp_id)
  JOIN dim_employee d ON (d.emp_id = b.upper_id)
 WHERE d.name = 'Jones'
```

Durch Vertauschen der Attribute LOWER_ID und UPPER_ID der Bridge Table lassen sich auch ähnliche Abfragen formulieren, welche die übergeordneten Datensätze aufsummieren (z.B. „Mr. Jones und alle seine Vorgesetzten“).

Wie in Abbildung 8 ersichtlich, gibt es zwischen der Faktentabelle und der Bridge Table wiederum eine n-zu-n-Beziehung. Bei einer Hierarchy Bridge Table kann diese auf einfache Weise eliminiert werden, indem das Datenmodell wie in Abbildung 9 modelliert wird:

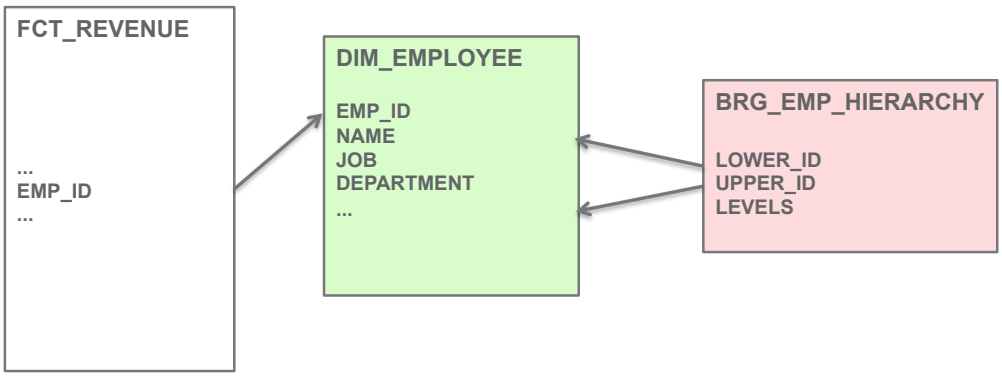

Abb. 9: Eliminierung der n-zu-n-Beziehung einer Hierarchy Bridge Table

Literatur

[1] Ralph Kimball, Margy Ross: The Data Warehouse Toolkit, Second Edition
John Wiley and Sons, Inc., 2002, ISBN 978-0471200246

[2] Christopher Adamson: Star Schema, The Complete Reference
McGraw-Hill Companies, 2010, ISBN 978-0071744324

Kontaktadresse:

Dani Schnider
Trivadis AG
Europa-Strasse 5
CH-8152 Glattbrugg

Telefon: +41(0)44-808 70 20
Fax: +41(0)44-808 70 21
E-Mail dani.schnider@trivadis.com
Internet: www.trivadis.com