


Dimensionale Modellierung in Microsoft Fabric Warehouse: Dimensionstabellen

Artikel • 25.06.2024

Gilt für:  SQL-Analyseendpunkt und Warehouse in Microsoft Fabric

Hinweis

Dieser Artikel ist Teil der [Dimensionsmodellierungsreihe](#) von Artikeln. Diese Serie konzentriert sich auf Anleitungen und bewährte Entwurfspraktiken im Zusammenhang mit der Dimensionale Modellierung in Microsoft Fabric Warehouse.

Dieser Artikel enthält Anleitungen und bewährte Methoden zum Entwerfen von **Dimensionstabellen** in einem dimensionalen Modell. Er bietet eine praktische Anleitung für das [Warehouse in Microsoft Fabric](#), das viele T-SQL-Funktionen wie die Erstellung von Tabellen und die Verwaltung von Daten in Tabellen unterstützt. Sie haben also die vollständige Kontrolle über die Erstellung Ihrer Dimensionsmodelltabellen und das Laden dieser mit Daten.

Hinweis

In diesem Artikel bezieht sich der Begriff *Data Warehouse* auf ein Enterprise Data Warehouse, das umfassende Integration kritischer Daten in der gesamten Organisation bereitstellt. Im Gegensatz dazu bezieht sich der eigenständige Begriff *Warehouse* auf ein Fabric Warehouse, bei dem es sich um eine Software-as-a-Service (SaaS)-relationale Datenbank handelt, die Sie zum Implementieren eines Data Warehouse verwenden können. Aus Gründen der Klarheit wird in diesem Artikel letzteres als *Fabric Warehouse* erwähnt.

Tipp

Wenn Sie noch keine Erfahrung mit der dimensionalen Modellierung haben, ist diese Artikelserie Ihr erster Schritt. Sie soll keine vollständige Diskussion über die dimensionale Modellierung von Designs bieten. Weitere Informationen finden Sie direkt in weit verbreiteten Veröffentlichungen wie *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling* (3. Ausgabe, 2013) von Ralph Kimball und anderen.

In einem dimensionalen Modell beschreibt eine Dimensionstabelle eine Entität, die für Ihre Geschäfts- und Analyseanforderungen relevant ist. Im Allgemeinen stellen Dimensionstabellen die *Elemente* dar, die Sie modellieren. Es kann sich um Produkte, Personen, Orte oder ein anderes Konzept handeln, einschließlich Datum und Zeit. Um Dimensionstabellen leicht zu identifizieren, wird ihren Namen normalerweise ein `d_` oder `Dim_` vorangestellt.

Struktur der Dimensionstabelle

Um die Struktur einer Dimensionstabelle zu beschreiben, betrachten Sie das folgende Beispiel einer Verkäufertabelle namens `d_Salesperson`. In diesem Beispiel werden gute Entwurfspraktiken angewandt. Jede der Spaltengruppen wird in den folgenden Abschnitten beschrieben.

SQL

```
CREATE TABLE d_Salesperson
(
    --Surrogate key
    Salesperson_SK INT NOT NULL,

    --Natural key(s)
    EmployeeID VARCHAR(20) NOT NULL,

    --Dimension attributes
    FirstName VARCHAR(20) NOT NULL,
    <...>

    --Foreign key(s) to other dimensions
    SalesRegion_FK INT NOT NULL,
    <...>

    --Historical tracking attributes (SCD type 2)
    RecChangeDate_FK INT NOT NULL,
    RecValidFromKey INT NOT NULL,
    RecValidToKey INT NOT NULL,
    RecReason VARCHAR(15) NOT NULL,
    RecIsCurrent BIT NOT NULL,

    --Audit attributes
    AuditMissing BIT NOT NULL,
    AuditIsInferred BIT NOT NULL,
    AuditCreatedDate DATE NOT NULL,
    AuditCreatedBy VARCHAR(15) NOT NULL,
    AuditLastModifiedDate DATE NOT NULL,
    AuditLastModifiedBy VARCHAR(15) NOT NULL
);
```

Ersatzschlüssel

Die Beispieldimensionstabelle verfügt über einen *Ersatzschlüssel*, der `Salesperson_SK` benannt wird. Ein Ersatzschlüssel ist ein einspaltiger eindeutiger Bezeichner, der in der Dimensionstabelle generiert und gespeichert wird. Es handelt sich um eine **Primärschlüsselspalte**, die verwendet wird, um sich auf andere Tabellen im dimensionalen Modell zu beziehen.

Mit Ersatzschlüsseln wird versucht, das Data Warehouse von Änderungen in den Quelldaten zu isolieren. Sie bieten auch viele weitere Vorteile, wie z. B. die Möglichkeit:

- mehrere Datenquellen zu konsolidieren (Vermeidung des Zusammentreffens von doppelten Identifikatoren).
- mehrspaltige natürliche Schlüssel zu einem effizienteren, einspaltigen Schlüssel zu konsolidieren.
- den Dimensionsverlauf mit einem sich **langsam ändernden Maß (SCD) Typ 2** zu verfolgen.
- die Breite der Faktentabelle zur Speicheroptimierung zu begrenzen (durch Auswahl des kleinstmöglichen Integer-Datentyps).

Eine Ersatzschlüsselspalte ist eine empfohlene Methode, auch wenn ein natürlicher Schlüssel (weiter beschrieben) als akzeptabler Kandidat erscheint. Sie sollten auch vermeiden, den Schlüsselwerten Bedeutung zu geben (mit Ausnahme von Datums- und Uhrzeitdimensionsschlüsseln, wie später beschrieben).

Natürliche Schlüssel

Die Beispieldimensionstabelle verfügt auch über einen *natürlichen Schlüssel*, der `EmployeeID` benannt wird. Ein natürlicher Schlüssel ist der im Quellsystem gespeicherte Schlüssel. Er ermöglicht die Zuordnung der Dimensionsdaten zu ihrem Quellsystem, was in der Regel durch einen ETL-Prozess (Extrahieren, Laden und Transformieren) zum Laden der Dimensionstabelle erfolgt. Manchmal wird ein natürlicher Schlüssel als *Geschäftsschlüssel* bezeichnet und seine Werte können für Geschäftsbutzer sinnvoll sein.

Manchmal verfügen Dimensionen nicht über einen natürlichen Schlüssel. Dies kann der Fall für Ihre **Datumsdimension** oder Nachschlagedimensionen oder wenn Sie Dimensionsdaten durch Normalisierung eines Flatfiles erzeugen.

Dimensionsattribute

Eine Beispieldimensionstabelle verfügt auch über *Dimensionsattribute*, z. B. die `FirstName` -

Spalte. Dimensionsattribute stellen Kontext zu den numerischen Daten bereit, die in verknüpften Faktentabellen gespeichert sind. Es handelt sich in der Regel um Textspalten, die in Analyseabfragen zum Filtern und Gruppieren (Slice and Dice) verwendet werden, aber nicht selbst aggregiert werden sollen. Einige Dimensionstabellen enthalten nur wenige Attribute, während andere viele Attribute enthalten (so viele, wie nötig sind, um die Abfrageanforderungen des Dimensionsmodells zu erfüllen).

Tipp

Eine gute Möglichkeit, um zu bestimmen, welche Dimensionen und Attribute Sie benötigen, besteht darin, die richtigen Personen zu finden und die richtigen Fragen zu stellen. Achten Sie insbesondere auf die Erwähnung des Wortes *nach*. Wenn beispielsweise jemand sagt, dass er Verkäufe *nach* Vertriebsmitarbeiter, *nach* Monat und *nach* Produktkategorie analysieren muss, teilt er Ihnen mit, dass er Dimensionen mit diesen Attributen benötigt.

Wenn Sie ein [Direct Lake-Semantikmodell](#) erstellen möchten, sollten Sie alle möglichen Spalten einschließen, die zum Filtern und Gruppieren als Dimensionsattribute erforderlich sind. Das liegt daran, dass die Semantikmodelle von Direct Lake keine berechneten Spalten unterstützen.

Fremdschlüssel

Die Beispieldimensionstabelle verfügt auch über einen *Fremdschlüssel*, der `SalesRegion_FK` benannt wird. Andere Dimensionstabellen können auf einen Fremdschlüssel verweisen und ihre Anwesenheit in einer Dimensionstabelle ist ein Sonderfall. Es weist darauf hin, dass die Tabelle mit einer anderen Dimensionstabelle verknüpft ist, was bedeutet, dass sie Teil einer [Snowflake-Dimension](#) sein kann oder mit einer [Outtrigger-Dimension](#) verknüpft ist.

Fabric Warehouse [unterstützt Fremdschlüsseleinschränkungen](#), kann jedoch nicht erzwungen werden. Daher ist es wichtig, dass Ihre ETL-Prozesstests für die Integrität zwischen verwandten Tabellen beim Laden von Daten durchgeführt werden.

Es ist dennoch ratsam, Fremdschlüssel zu erstellen. Ein guter Grund zum Erstellen nicht erzwungener Fremdschlüssel besteht darin, Modellierungstools wie Power BI Desktop zu ermöglichen, Beziehungen zwischen Tabellen im semantischen Modell automatisch zu erkennen und zu erstellen.

Verlaufsverfolgungsattribute

Die Beispieldimensionstabelle verfügt auch über verschiedene *Verlaufsverfolgungsattribute*. Verlaufsverfolgungsattribute sind optional, je nachdem, ob Sie bestimmte Änderungen im Quellsystem nachverfolgen möchten. Sie ermöglichen das Speichern von Werten, um die primäre Rolle eines Data Warehouses zu unterstützen, was die Vergangenheit genau beschreiben soll. Diese Attribute speichern insbesondere den historischen Kontext, wenn der ETL-Prozess neue oder geänderte Daten in die Dimension lädt.

Weitere Informationen finden Sie weiter unten in diesem Artikel unter [Historische Änderungen verwalten](#).

Überwachungsattribute

Die Beispieldimensionstabelle verfügt auch über verschiedene *Überwachungsattribute*. Überwachungsattribute sind optional, aber empfohlen. Sie ermöglichen es Ihnen, nachzuverfolgen, wann und wie Dimensionsdatensätze erstellt oder geändert wurden, und sie können Diagnose- oder Problembehandlungsinformationen enthalten, die während ETL-Prozessen erhoben wurden. Sie möchten zum Beispiel nachverfolgen, wer (oder welcher Prozess) wann eine Zeile aktualisiert hat. Überwachungsattribute können auch bei der Diagnose eines schwierigen Problems helfen, etwa wenn ein ETL-Prozess unerwartet abbricht. Sie können auch Dimensionselemente als Fehler oder [abgeleitete Elemente](#) kennzeichnen.

Größe der Dimensionstabelle

Häufig sind die nützlichsten und vielseitigsten Dimensionen in einem dimensional Modell große, breite Dimensionen. Sie sind *groß* in Bezug auf Zeilen (über Millionen) und breit in Bezug auf die Anzahl der Dimensionsattribute (potenziell Hunderte). Die Größe ist nicht so wichtig (obwohl Sie für die kleinstmögliche Größe entwerfen und optimieren sollten). Wichtig ist, dass die Dimension die erforderliche Filterung, Gruppierung und genaue historische Analyse von Faktendaten unterstützt.

Große Dimensionen können aus mehreren Quellsystemen stammen. In diesem Fall muss die Dimensionsverarbeitung die Daten kombinieren, zusammenführen, deduplizieren und standardisieren sowie Ersatzschlüssel zuweisen.

Im Vergleich dazu sind einige Dimensionen winzig. Sie können Nachschlagetabellen darstellen, die nur mehrere Datensätze und Attribute enthalten. Häufig speichern diese kleinen Dimensionen Kategoriewerte im Zusammenhang mit Transaktionen in Faktentabellen und sie werden als Dimensionen mit Ersatzschlüsseln implementiert, um sich auf die Faktendatensätze zu beziehen.

💡 Tipp

Wenn Sie viele kleine Dimensionen haben, sollten Sie sie in eine [Junk-Dimension](#) konsolidieren.

Designkonzepte für Dimensionen

In diesem Abschnitt werden verschiedene Designkonzepte für Dimensionen beschrieben.

Denormalisierung vs. Normalisierung

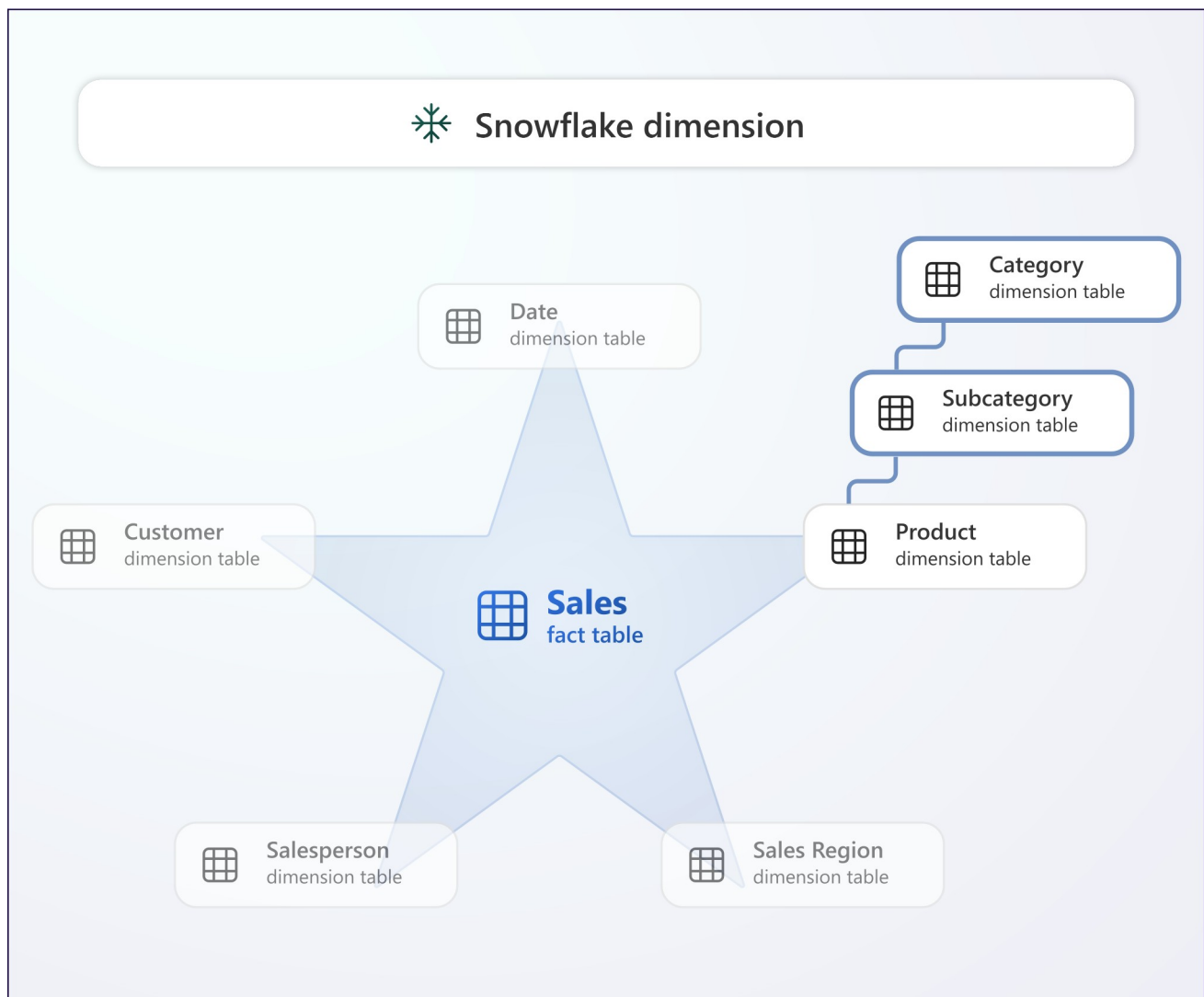
Es ist fast immer der Fall, dass Dimensionstabellen *denormalisiert* werden sollten. Während die *Normalisierung* der Begriff ist, der Daten beschreibt, die so gespeichert werden, dass sich wiederholende Daten reduziert werden, ist Denormalisierung der Begriff, der definiert, wo vorberechnete redundante Daten existieren. Redundante Daten entstehen in der Regel durch die Speicherung von Hierarchien (siehe unten), d. h. Hierarchien werden vereinfacht. Eine Produktdimension könnte beispielsweise Unterkategorie (und die zugehörigen Attribute) und Kategorie (und die zugehörigen Attribute) speichern.

Da Dimensionen im Allgemeinen klein sind (im Vergleich zu Faktentabellen), werden die Kosten für die Speicherung redundanter Daten fast immer durch die verbesserte Abfrageleistung und Benutzerfreundlichkeit aufgewogen.

Schneeflockendimensionen

Eine Ausnahme bei der Denormalisierung besteht darin, eine *Snowflake-Dimension* zu entwerfen. Eine Snowflake-Dimension wird normalisiert und speichert die Dimensionsdaten in mehreren verwandten Tabellen.

Das folgende Diagramm zeigt eine Snowflake-Dimension, die drei verknüpfte Dimensionstabellen umfasst: Product, Subcategory und Category.



Erwägen Sie die Implementierung einer Snowflake-Dimension in folgenden Fällen:

- Die Dimension ist extrem groß und die Speicherkosten überwiegen den Bedarf an hoher Abfrageleistung. (Es ist jedoch regelmäßig zu überprüfen, ob dies immer noch der Fall ist.)
- Sie brauchen Schlüssel, um die Dimension mit höherrangigen Fakten zu verbinden. Die Faktentabelle für den Verkauf speichert beispielsweise Zeilen auf Produktebene, die Faktentabelle für das Verkaufsziel speichert jedoch Zeilen auf Unterkategorieebene.
- Sie müssen **historische Änderungen** auf höheren Granularitätsebenen nachverfolgen.

ⓘ Hinweis

Denken Sie daran, dass eine Hierarchie in einem Power BI-Semantikmodell nur auf Spalten aus einer einzelnen Semantikmodelltabelle basieren kann. Daher sollte eine Snowflake-Dimension ein denormalisiertes Ergebnis liefern, indem eine Ansicht verwendet wird, die die Snowflake-Tabellen miteinander verbindet.

Hierarchien

Häufig erzeugen Dimensionsspalten *Hierarchien*. Hierarchien ermöglichen das Untersuchen von Daten auf unterschiedlichen Zusammenfassungsebenen. Die anfängliche Ansicht einer Matrixanzeige kann z. B. jährliche Umsätze anzeigen und der Berichtsanwender kann [Detailinformationen anzeigen](#), um vierteljährliche und monatliche Umsätze anzuzeigen.

Es gibt drei Möglichkeiten zum Speichern einer Hierarchie in einer Dimension. Verwenden Sie Folgendes:

- Spalten aus einer einzelnen, denormalisierten Dimension.
- Eine Snowflake-Dimension, die mehrere verwandte Tabellen umfasst.
- Eine Beziehung zwischen übergeordneten und untergeordneten Elementen (selbstreferenzierend) in einer Dimension.

Hierarchien können ausgeglichen oder unausgeglichen sein. Es ist auch wichtig zu verstehen, dass einige Hierarchien unregelmäßig sind.

Ausgeglichene Hierarchien

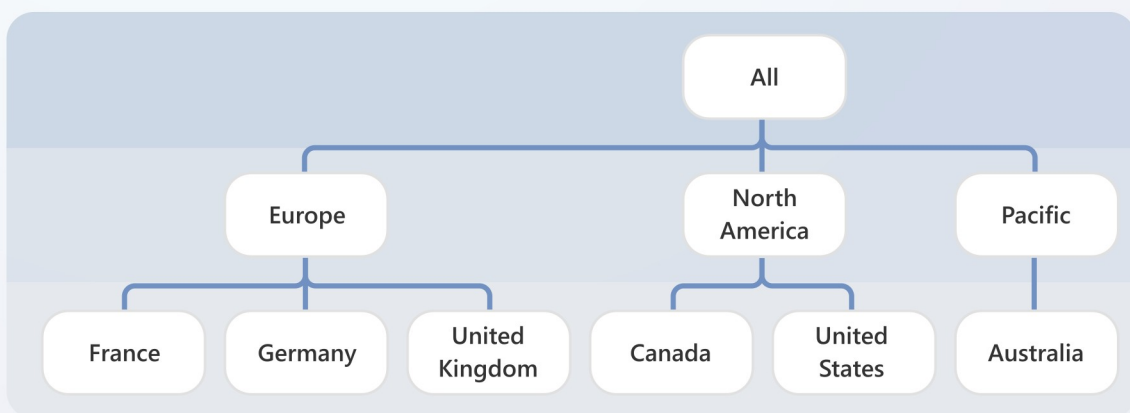
Ausgeglichene Hierarchien sind die am häufigsten verwendeten Hierarchietypen. Eine ausgeglichene Hierarchie weist die gleiche Anzahl von Ebenen auf. Ein gängiges Beispiel für eine ausgeglichene Hierarchie ist eine Kalenderhierarchie in einer Datumsdimension, die Ebenen für Jahr, Quartal, Monat und Datum umfasst.

Das folgende Diagramm zeigt eine ausgeglichene Hierarchie der Vertriebsregionen. Sie umfasst zwei Ebenen, die Vertriebsregionengruppe und die Vertriebsregion.



Balanced hierarchy

SalesRegion_SK	Group	SalesRegion
1	Europe	France
2	Europe	Germany
3	Europe	United Kingdom
4	North America	Canada
5	North America	United States
6	Pacific	Australia



Ebenen einer ausgeglichenen Hierarchie basieren entweder auf Spalten aus einer einzelnen, denormalisierten Dimension oder aus Tabellen, die eine Snowflake-Dimension bilden. Wenn sie auf einer einzelnen, denormalisierten Dimension basieren, enthalten die Spalten, die die höheren Ebenen darstellen, redundante Daten.

Bei ausgeglichenen Hierarchien beziehen sich Fakten immer auf eine einzelne Hierarchieebene, die in der Regel die niedrigste Ebene ist. Auf diese Weise können die Fakten bis zur höchsten Ebene der Hierarchie aggregiert (aufgerollt) werden. Fakten können sich auf jede Ebene beziehen, die durch das Aggregationsintervall der Faktentabelle bestimmt wird. So kann beispielsweise die Umsatzfaktentabelle auf Datumsebene gespeichert werden, während die Umsatzzielfaktentabelle auf Quartalsebene gespeichert werden kann.

Unausgeglichene Hierarchien

Unausgeglichene Hierarchien sind die am häufigsten verwendeten Hierarchietypen. Eine unausgeglichene Hierarchie weist Ebenen auf, die auf einer Beziehung zwischen

übergeordneten und untergeordneten Elementen basieren. Aus diesem Grund wird die Anzahl der Ebenen in einer unausgeglichene Hierarchie durch die Dimensionszeilen und nicht durch bestimmte Spalten der Dimensionstabelle bestimmt.

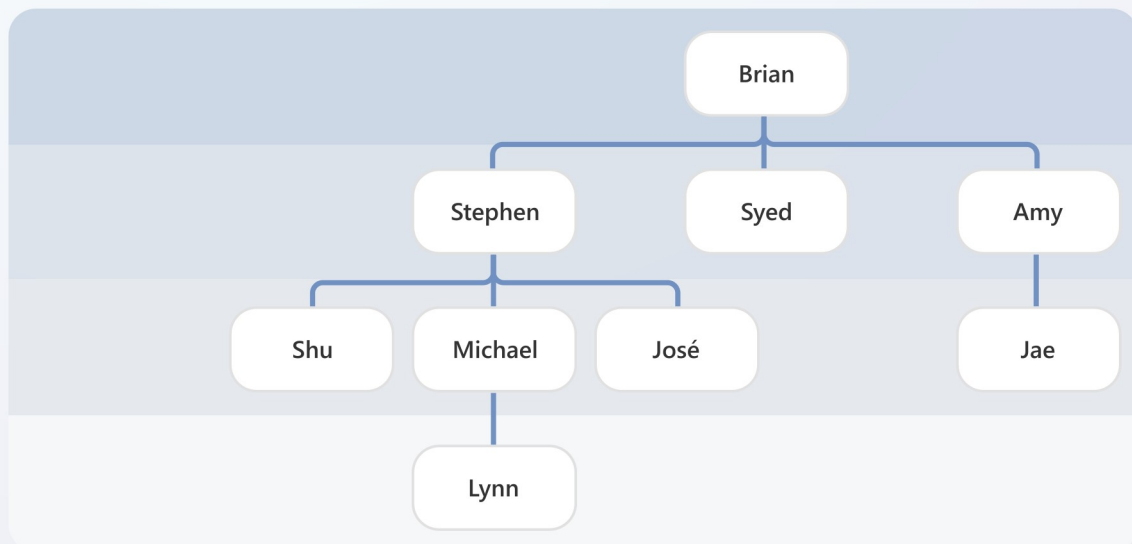
Ein gängiges Beispiel für eine unausgeglichene Hierarchie ist eine Mitarbeiterhierarchie, in der jede Zeile in einer Mitarbeiterdimension sich auf eine Berichts-Manager-Zeile *in derselben Tabelle* bezieht. In diesem Fall kann jeder Mitarbeiter ein Manager*in sein, der Mitarbeiter meldet. Natürlich haben einige Verzweigungen der Hierarchie mehr Ebenen als andere.

Das folgende Diagramm zeigt eine unausgeglichene Hierarchie. Es besteht aus vier Ebenen und jedes Mitglied in der Hierarchie ist ein Vertriebsmitarbeiter. Beachten Sie, dass Vertriebsmitarbeiter in der Hierarchie eine unterschiedliche Anzahl von Vorgängern haben, je nachdem, wem sie unterstellt sind.



Unbalanced hierarchy

Salesperson_SK	ReportsTo_SK	Salesperson
277	277	Brian Welcker
290	277	Amy Alberts
272	277	Stephen Jiang
294	277	Syed Abbas
287	272	Shu Ito
281	272	Michael Blythe
288	272	José Saraiva
291	290	Jae Pak
299	281	Lynn Tsoflias



Weitere gängige Beispiele für unausgeglichene Hierarchien sind Stücklisten, Unternehmensbesitzmodelle und Finanzbuchhaltung.

Bei unausgebalancierten Hierarchien beziehen sich Fakten immer auf das Aggregationsintervall der Dimension. Beispielsweise beziehen sich Verkaufsdaten auf unterschiedliche Vertriebsmitarbeiter, die unterschiedliche Berichtsstrukturen haben. Die Dimensionstabelle hätte einen Ersatzschlüssel (mit dem Namen `Salesperson_SK`) und eine Fremdschlüsselspalte `ReportsTo_Salesperson_FK`, die auf die Primärschlüsselspalte verweist. Jeder Vertriebsmitarbeiter, der niemanden zu verwalten hat, befindet sich nicht unbedingt auf der untersten Ebene eines Zweiges der Hierarchie. Wenn sie sich nicht auf der untersten Ebene befinden, kann ein Vertriebsmitarbeiter Produkte verkaufen und über Vertriebsmitarbeiter

verfügen, die ebenfalls Produkte verkaufen. Der Rollup der Faktendaten muss also den einzelnen Vertriebsmitarbeiter und alle seine Nachfolgerelemente berücksichtigen.

Die Abfrage von Hierarchien zwischen übergeordneten und untergeordneten Elementen kann komplex und langsam sein, insbesondere bei großen Dimensionen. Während das Quellsystem Beziehungen als übergeordnete und untergeordnete Beziehungen speichern kann, empfehlen wir Ihnen, die Hierarchie zu *naturalisieren*. In diesem Fall bedeutet naturalisieren, dass die Hierarchieebenen in der Dimension als Spalten umgewandelt und gespeichert werden.

Tipp

Wenn Sie sich entscheiden, die Hierarchie nicht zu naturalisieren, können Sie dennoch eine Hierarchie basierend auf einer Beziehung zwischen übergeordneten und untergeordneten Elementen in einem Power BI-Semantikmodell erstellen. Dieser Ansatz wird jedoch für große Dimensionen nicht empfohlen. Weitere Informationen finden Sie unter [in DAX](#).

Unregelmäßige Hierarchien

Manchmal ist eine Hierarchie *unregelmäßig*, da das übergeordnete Element eines Elements in der Hierarchie auf einer Ebene vorhanden ist, die nicht unmittelbar darüber liegt. In diesen Fällen wiederholen fehlende Ebenenwerte den Wert des übergeordneten Elements.

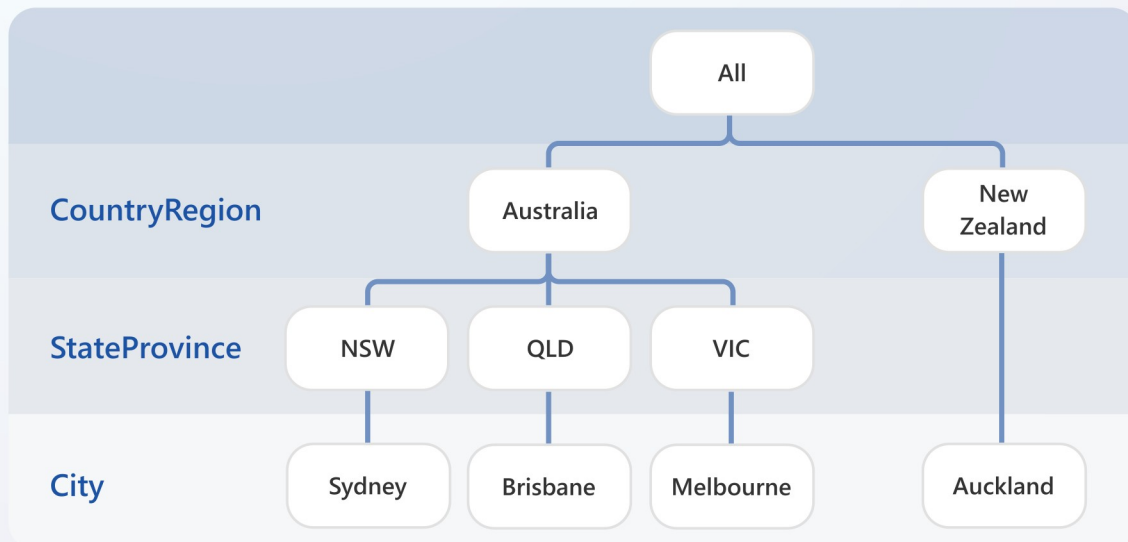
Betrachten Sie ein Beispiel für eine ausgeglichene Geografiehierarchie. Eine unregelmäßige Hierarchie ist vorhanden, wenn ein Land/eine Region keine Bundesländer oder Kantone hat. Neuseeland hat beispielsweise weder Bundesländer noch Kantone. Wenn Sie also die Zeile Neuseeland einfügen, sollten Sie auch den Wert für Land/Region in der `StateProvince`-Spalte speichern.

Das folgende Diagramm zeigt eine unregelmäßige Hierarchie geografischer Regionen.



Ragged hierarchy

Geography_SK	CountryRegion	StateProvince	City
1	Australia	NSW	Sydney
2	Australia	QLD	Brisbane
3	Australia	VIC	Melbourne
4	New Zealand	New Zealand	Auckland



Verwalten von historischen Änderungen

Bei Bedarf können historische Änderungen durch die Implementierung einer *langsam veränderliche Dimension* (SCD) verwaltet werden. Ein SCD behält den historischen Kontext bei, wenn neue oder geänderte Daten geladen werden.

Im Folgenden werden die häufigsten SCD-Typen aufgeführt.

- **Typ 1:** Überschreiben Sie das vorhandene Dimensionselement.
- **Typ 2:** Einfügen eines neuen zeitbasierten *versionsbasierten* Dimensionselements.
- **Typ 3:** Nachverfolgen des eingeschränkten Verlaufs mit Attributen.

Es ist möglich, dass eine Dimension sowohl SCD-Typ 1- als auch SCD-Typ 2-Änderungen unterstützen kann.

SCD-Typ 3 wird nicht häufig verwendet, was zum Teil daran liegt, dass er in einem

semantischen Modell schwer zu verwenden ist. Überlegen Sie sorgfältig, ob ein SCD-Typ 2-Ansatz besser geeignet wäre.

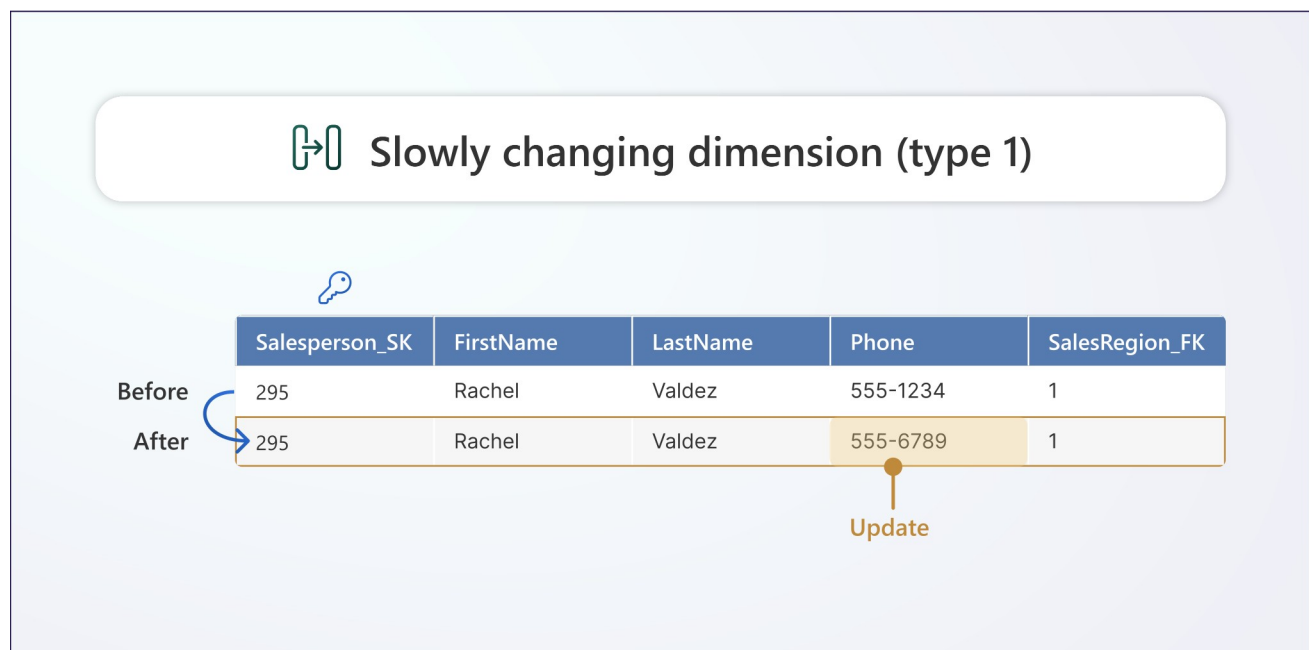
💡 Tipp

Wenn Sie eine *schnell veränderliche Dimension* antizipieren, bei der es sich um eine Dimension handelt, die sich häufig ändert, sollten Sie stattdessen dieses Attribut der **Faktentabelle** hinzufügen. Wenn das Attribut numerisch ist, wie der Produktpreis, können Sie es als Measure in die Faktentabelle aufnehmen. Wenn es sich bei dem Attribut um einen Textwert handelt, können Sie eine Dimension erstellen, die auf allen Textwerten basiert und ihren Dimensionsschlüssel zur Faktentabelle hinzufügen.

SCD Typ 1

SCD-Typ 1-Änderungen überschreiben die vorhandene Dimensionszeile, da keine Änderungen nachverfolgt werden müssen. Dieser SCD-Typ kann auch verwendet werden, um Fehler zu beheben. Es ist ein gängiger SCD-Typ und sollte für die meisten sich ändernden Attribute verwendet werden, z. B. Kundenname, E-Mail-Adresse und andere.

Das folgende Diagramm zeigt den Vorher- und Nachher-Zustand eines Dimensionselements eines Vertriebsmitarbeiters, dessen Telefonnummer sich geändert hat.



Dieser SCD-Typ behält keine historische Perspektive bei, da die vorhandene Zeile aktualisiert wird. Das bedeutet, dass Änderungen des SCD-Typs 1 zu verschiedenen übergeordneten Aggregationen führen können. Wenn beispielsweise ein Vertriebsmitarbeiter einer anderen Vertriebsregion zugewiesen ist, würde eine SCD-Typ 1-Änderung die Dimensionszeile

überschreiben. Das Rollup der historischen Verkaufsergebnisse des Vertriebsmitarbeiters auf die Region würde dann zu einem anderen Ergebnis führen, da es nun die neue aktuelle Verkaufsregion verwendet. Es ist so, als ob dieser Vertriebsmitarbeiter immer der neuen Vertriebsregion zugewiesen wurde.

SCD Typ 2

SCD-Typ 2-Änderungen führen zu neuen Zeilen, die eine zeitbasierte Version eines Dimensionselements darstellen. Es gibt immer eine aktuelle Versionszeile, die den Zustand des Dimensionselements im Quellsystem widerspiegelt. [Historische Nachverfolgungsattribute](#) in den Speicherwerten der Dimensionstabellen, die das Identifizieren der aktuellen Version (aktuelles Kennzeichen ist `TRUE`) und den Gültigkeitszeitraum ermöglichen. Ein Ersatzschlüssel ist erforderlich, da beim Speichern mehrerer Versionen doppelte natürliche Schlüssel vorhanden sind.

Es ist eine gängige Art von SCD, sollte jedoch für Attribute reserviert werden, die historische Perspektive beibehalten müssen.

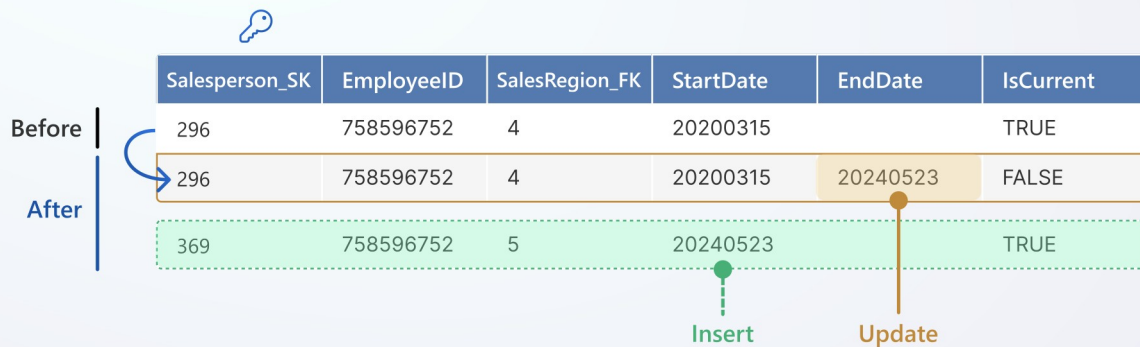
Wenn beispielsweise ein Vertriebsmitarbeiter einer anderen Vertriebsregion zugewiesen ist, umfasst eine SCD-Typ-2-Änderung einen Aktualisierungsvorgang und einen Einfügevorgang.

1. Der Aktualisierungsvorgang überschreibt die aktuelle Version, um die historischen Nachverfolgungsattribute zu setzen. Insbesondere wird die Spalte für das Gültigkeitsende auf das ETL-Verarbeitungsdatum (oder einen geeigneten Zeitstempel im Quellsystem) und das aktuelle Kennzeichen auf `FALSE` gesetzt.
2. Der Einfügevorgang fügt eine neue, aktuelle Version hinzu, wobei die Spalte mit dem Gültigkeitsbeginn auf den Wert der Spalte mit dem Gültigkeitsende (der zur Aktualisierung der vorherigen Version verwendet wurde) und das aktuelle Kennzeichen auf `TRUE` gesetzt wird.

Es ist wichtig zu verstehen, dass die Granularität der verwandten Faktentabellen nicht auf der Ebene des Vertriebsmitarbeiters, sondern auf der Ebene der *Vertriebsmitarbeiter-Version* liegt. Der Rollup ihrer historischen Verkaufsergebnisse auf die Region wird korrekte Ergebnisse liefern, aber es wird zwei (oder mehr) Versionen von Vertriebsmitarbeitern geben, die analysiert werden müssen.

Das folgende Diagramm zeigt den Vorher- und Nachher-Zustand eines Dimensionselements eines Vertriebsmitarbeiters, dessen Vertriebsgebiet sich geändert hat. Da die Organisation die Leistung der Vertriebsmitarbeiter nach der Region analysieren möchte, der sie zugewiesen sind, löst sie eine SCD-Typ-2-Änderung aus.

Slowly changing dimension (type 2)



💡 Tipp

Wenn eine Dimensionstabelle SCD-Typ 2-Änderungen unterstützt, sollten Sie ein Bezeichnungsattribut einschließen, das das Element und die Version beschreibt. Ein Beispiel: Die Vertriebsmitarbeiterin Lynn Tsoflias von Adventure Works wechselt vom australischen Verkaufsgebiet in das britische Verkaufsgebiet. Das Bezeichnungsattribut für die erste Version könnte „Lynn Tsoflias (Australien)“ lauten und das Bezeichnungsattribut für die neue, aktuelle Version könnte „Lynn Tsoflias (Vereinigtes Königreich)“ lauten. Wenn hilfreich, können Sie auch die Gültigkeitsdaten in die Bezeichnung einschließen.

Sie sollten die Notwendigkeit der historischen Genauigkeit im Vergleich zu Benutzerfreundlichkeit und Effizienz ausgleichen. Versuchen Sie, zu viele SCD-Typ-2-Änderungen in einer Dimensionstabelle zu vermeiden, da dies zu einer überwältigenden Anzahl von Versionen führen kann, die für Analysten schwer zu verstehen sind.

Außerdem könnten zu viele Versionen darauf hindeuten, dass ein geändertes Attribut möglicherweise besser in der Faktentabelle gespeichert werden kann. In Erweiterung des früheren Beispiels könnte bei häufigen Änderungen der Verkaufsregion die Verkaufsregion als Dimensionsschlüssel in der Faktentabelle gespeichert werden, anstatt eine SCD Typ 2 zu implementieren.

Berücksichtigen Sie die folgenden SCD-Typ 2-Verlaufsnachverfolgungsattribute.

SQL


```
CREATE TABLE d_Salesperson
(
    <...>

    --Historical tracking attributes (SCD type 2)
    RecChangeDate_FK INT NOT NULL,
    RecValidFromKey INT NOT NULL,
    RecValidToKey INT NOT NULL,
    RecReason VARCHAR(15) NOT NULL,
    RecIsCurrent BIT NOT NULL,

    <...>
);
```

Hier sind die Zwecke der Attribute für die historische Verfolgung.

- In der `RecChangeDate_FK` -Spalte wird das Datum gespeichert, an dem die Änderung wirksam wurde. Sie können abfragen, wann Änderungen vorgenommen wurden.
- In den Spalten `RecValidFromKey` und `RecValidToKey` werden die Gültigkeitsdaten für die Zeile gespeichert. Sie können das früheste in der Datumsdimension gefundene Datum für `RecValidFromKey` speichern, um die ursprüngliche Version darzustellen und `01/01/9999` für die `RecValidToKey` der aktuellen Versionen speichern.
- Die Spalte `RecReason` ist optional. Sie ermöglicht das Dokumentieren des Grundes, warum die Version eingefügt wurde. Sie könnte codieren, welche Attribute geändert wurden, oder es könnte sich um einen Code aus dem Quellsystem handeln, der einen bestimmten geschäftlichen Grund angibt.
- Die `RecIsCurrent` -Spalte ermöglicht nur das Abrufen aktueller Versionen. Sie wird verwendet, wenn der ETL-Prozess beim Laden von Faktentabellen nach Dimensionsschlüsseln sucht.

ⓘ Hinweis

Einige Quellsysteme speichern keine historischen Änderungen, daher ist es wichtig, dass die Dimension regelmäßig verarbeitet wird, um Änderungen zu erkennen und neue Versionen zu implementieren. Auf diese Weise können Sie Änderungen kurz nach ihrem Auftreten erkennen und die Gültigkeitsdaten sind genau.

SCD Typ 3

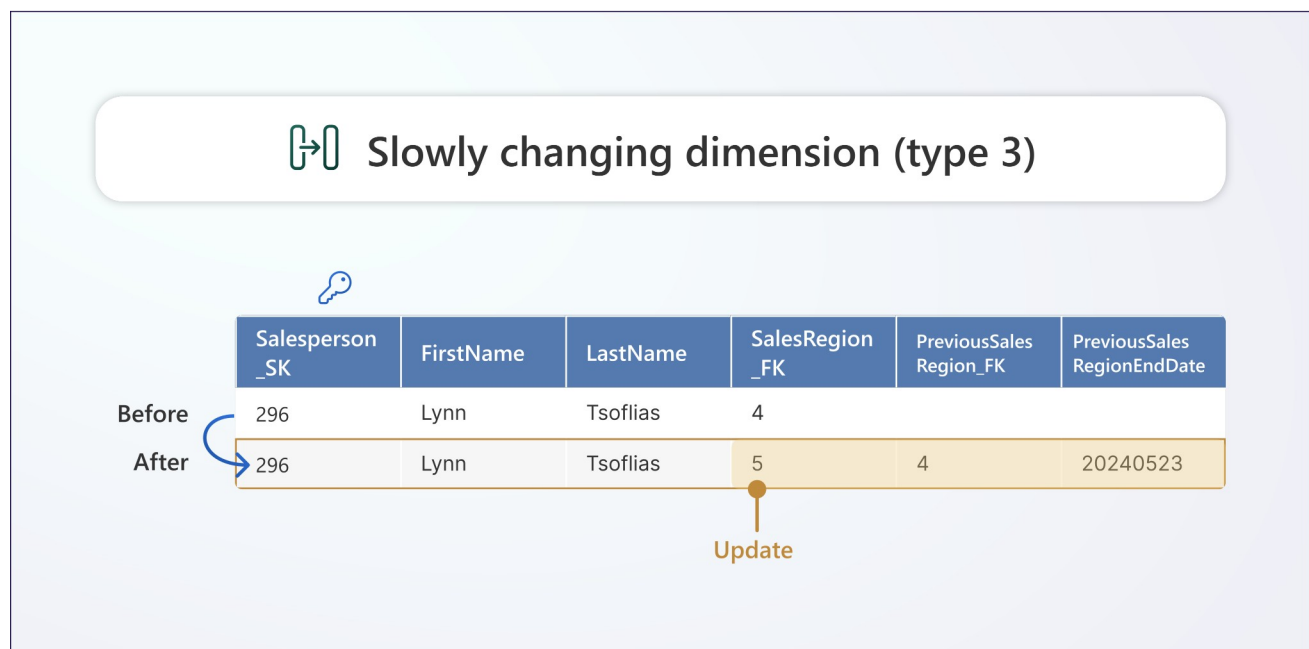
SCD Typ 3-Änderungen verfolgen einen begrenzten Verlauf mit Attributen. Dieser Ansatz kann nützlich sein, wenn die letzte Änderung oder eine Reihe der neuesten Änderungen

erfasst werden muss.

Dieser SCD-Typ behält eine *begrenzte* historische Perspektive bei. Er kann nützlich sein, wenn nur die anfänglichen und aktuellen Werte gespeichert werden sollen. In diesem Fall wären keine zwischenzeitlichen Änderungen erforderlich.

Wenn beispielsweise ein Vertriebsmitarbeiter einer anderen Vertriebsregion zugewiesen ist, überschreibt ein SCD-Typ 3 die Dimensionszeile. Eine Spalte, die speziell das vorherige Verkaufsgebiet speichert, wird als vorheriges Verkaufsgebiet festgelegt, und das neue Verkaufsgebiet wird als aktuelles Verkaufsgebiet festgelegt.

Das folgende Diagramm zeigt den Vorher- und Nachher-Zustand eines Dimensionselements eines Vertriebsmitarbeiters, dessen Vertriebsgebiet sich geändert hat. Da die Organisation eine frühere Vertriebsgebietszuordnung ermitteln möchte, löst sie eine Änderung vom SCD-Typ 3 aus.



Spezielle Dimensionselemente

Sie können Zeilen in eine Dimension einfügen, die fehlende, unbekannte, N/A- oder Fehlerzustände darstellt. Sie können z. B. die folgenden Ersatzschlüsselwerte verwenden.

 **Tabelle erweitern**

Schlüsselwert	Kostenträger
0	Fehlt (im Quellsystem nicht verfügbar)
-1	Unbekannt (Nachschlagefehler beim Laden einer Faktentabelle)

Schlüsselwert	Kostenträger
-2	N/V (nicht zutreffend)
-3	Fehler

Kalender und Zeit

Fast ohne Ausnahme speichern Faktentabellen Measures zu bestimmten Zeitpunkten. Zur Unterstützung der Analyse nach Datum (und möglicherweise Zeit) müssen Kalenderdimensionen (Datum und Zeit) vorhanden sein.

Es ist ungewöhnlich, dass ein Quellsystem Kalenderdimensionsdaten aufweisen würde, daher muss es im Data Warehouse generiert werden. Normalerweise werden sie einmal generiert und wenn es sich um eine Kalenderdimension handelt, wird sie bei Bedarf um zukünftige Daten erweitert.

Datumsdimension

Die Datumsdimension (oder Kalenderdimension) ist die am häufigsten verwendete Dimension für die Analyse. Sie speichert eine Zeile pro Datum und unterstützt die allgemeine Anforderung, nach bestimmten Datumsperioden wie Jahren, Quartalen oder Monaten zu filtern oder zu gruppieren.

Wichtig

Eine Datumsdimension sollte kein Aggregationsintervall enthalten, das sich auf die Tageszeit erstreckt. Wenn eine Tageszeitanalyse erforderlich ist, sollten Sie sowohl eine Datumsdimension als auch eine [Zeitdimension](#) haben (siehe nächste Beschreibung). Faktentabellen, in denen Fakten zur Tageszeit gespeichert werden, sollten zwei Fremdschlüssel aufweisen, einen für jede dieser Dimensionen.

Der natürliche Schlüssel der Datumsdimension sollte den Datentyp **Datum** verwenden. Der Ersatzschlüssel sollte das Datum mithilfe des Formats `YYYYMMDD` und des **int**-Datentyps speichern. Diese akzeptierte Praxis sollte die einzige Ausnahme (neben der Zeitdimension) sein, wenn der Wert des Ersatzschlüssels Bedeutung hat und menschlich lesbar ist. Die Speicherung von `YYYYMMDD` als Datentyp **int** ist nicht nur effizient und numerisch sortiert, sondern entspricht auch dem eindeutigen Datumsformat der International Standards Organization (ISO) 8601.

Im Folgenden finden Sie einige allgemeine Attribute, die in eine Datumsdimension

eingeschlossen werden sollen.

- `Year`, `Quarter`, `Month`, `Day`
- `QuarterNumberInYear`, `MonthNumberInYear` – die möglicherweise zum Sortieren von Beschriftungen erforderlich sind.
- `FiscalYear`, `FiscalQuarter` – einige Firmenbuchhaltungspläne beginnen Mitte des Jahres, sodass sich der Anfang/Ende des Kalenderjahres und des Geschäftsjahres unterscheiden.
- `FiscalQuarterNumberInYear`, `FiscalMonthNumberInYear` – die möglicherweise zum Sortieren von Beschriftungen erforderlich sind.
- `WeekOfYear` – Es gibt mehrere Möglichkeiten, die Woche des Jahres zu bezeichnen, einschließlich eines ISO-Standards mit 52 oder 53 Wochen.
- `IsHoliday`, `HolidayText` – Wenn Ihre Organisation in mehreren Regionen tätig ist, sollten Sie mehrere Gruppen von Feiertagslisten verwalten, die jede Geografie als separate Dimension oder in mehreren Attributen in der Datumsdimension beobachtet. Durch das Hinzufügen eines `HolidayText`-Attributs können Feiertage für die Berichterstellung identifiziert werden.
- `IsWeekday` – In einigen Regionen ist die Standardarbeitswoche nicht Montag bis Freitag. Die Arbeitswoche ist beispielsweise Sonntag bis Donnerstag in vielen Regionen des Nahen Ostens, während andere Regionen eine Arbeitswoche mit vier oder sechs Tagen verwenden.
- `LastDayOfMonth`
- `RelativeYearOffset`, `RelativeQuarterOffset`, `RelativeMonthOffset`, `RelativeDayOffset` – die zur Unterstützung der relativen Datumsfilterung (z. B. Vormonat) erforderlich sein könnte. Aktuelle Zeiträume verwenden einen Offset von Null (0); Frühere Zeiträume speichern Offsets von -1, -2, -3...; Zukünftige Zeiträume speichern Offsets von 1, 2, 3...

Wie bei jeder Dimension ist es wichtig, dass sie Attribute enthält, die die bekannten Filter-, Gruppierungs- und Hierarchieranforderungen unterstützen. Möglicherweise gibt es auch Attribute, die Übersetzungen von Bezeichnungen in andere Sprachen speichern.

Wenn die Dimension verwendet wird, um sich auf übergeordnete Fakten zu beziehen, kann die Faktentabelle das erste Datum der Datumsperiode verwenden. Beispielsweise würde eine Umsatzziel-Faktentabelle, in der vierteljährliche Vertriebsmitarbeiterziele gespeichert werden, das erste Datum des Quartals in der Datumsdimension speichern. Ein alternativer Ansatz besteht darin, Spalten in der Datumsdimension zu erstellen. Beispielsweise könnte ein Quartalsschlüssel den Quartalsschlüssel mithilfe des Formats `yyyyq` und des Datentyps **smallint** speichern.

Die Dimension sollte mit dem bekannten Datumsbereich gefüllt werden, der von allen

Faktentabellen verwendet wird. Sie sollte auch zukünftige Datumsangaben enthalten, an denen das Data Warehouse Fakten zu Zielen, Budgets oder Prognosen speichert. Wie bei anderen Dimensionen können Sie Zeilen einschließen, die fehlende, unbekannte, N/A- oder Fehlersituationen darstellen.

Tipp

Durchsuchen Sie das Internet nach dem „Datumsdimensionsgenerator“, um Skripts und Tabellenkalkulationen zu finden, die Datumsdaten generieren.

Normalerweise sollte der ETL-Prozess zu Beginn des nächsten Jahres die Datumsdimensionszeilen auf eine bestimmte Anzahl von Jahren erweitern. Wenn die Dimension relative Offset-Attribute enthält, muss der ETL-Prozess täglich ausgeführt werden, um Offset-Attributwerte basierend auf dem aktuellen Datum (heute) zu aktualisieren.

Zeitdimension

Manchmal müssen Fakten zu einem Zeitpunkt (z. B. Tageszeit) gespeichert werden. In diesem Fall erstellen Sie eine Zeitdimension (oder Uhrendimension). Es könnte ein Aggregationsintervall von Minuten ($24 \times 60 = 1440$ Zeilen) oder sogar Sekunden ($24 \times 60 \times 60 = 86400$ Zeilen) haben. Andere mögliche Aggregationsintervalle sind halbe Stunde oder Stunde.

Der natürliche Schlüssel der Zeitdimension sollte den Datentyp **Datum** verwenden. Der Ersatzschlüssel könnte ein geeignetes Format verwenden und Werte speichern, die eine Bedeutung haben und für den Menschen lesbar sind, z. B. durch Verwendung des HHMM- oder HHMMSS -Formats.

Im Folgenden finden Sie einige allgemeine Attribute, die in eine Zeitdimension eingeschlossen werden sollen.

- Hour, HalfHour, QuarterHour, Minute
- Zeitperiodenbeschriftungen (Morgen, Nachmittag, Abend, Nacht)
- Arbeitsschichtnamen
- Peak- oder Off-Peak-Flags

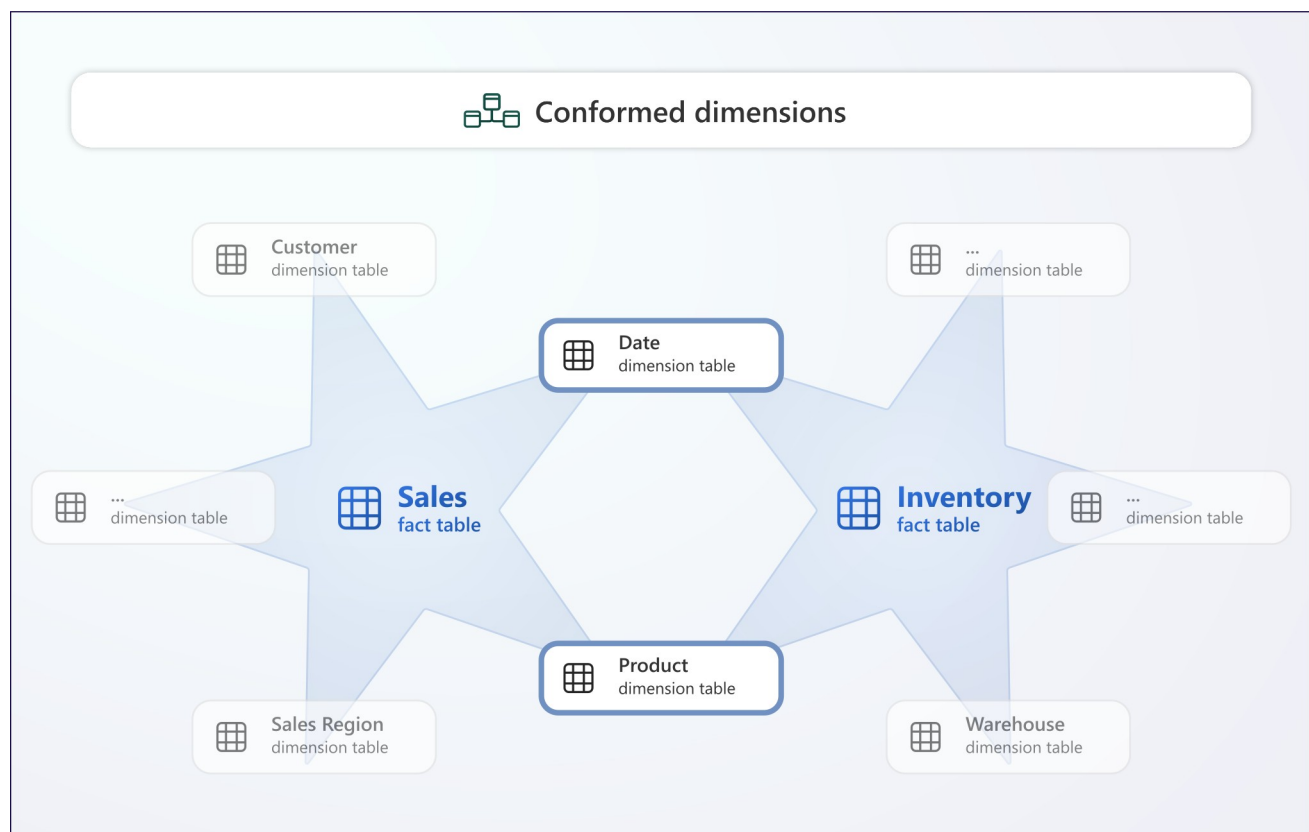
Konforme Dimensionen

Einige Dimensionen sind möglicherweise *konforme Dimensionen*. Konforme Dimensionen

beziehen sich auf viele Faktentabellen und werden daher von mehreren Sternen in einem Dimensionsmodell gemeinsam genutzt. Sie liefern Konsistenz und können Ihnen helfen, die laufende Entwicklung und Wartung zu reduzieren.

Es ist beispielsweise typisch, dass Faktentabellen mindestens einen Datumsdimensionsschlüssel speichern (da die Aktivität fast immer nach Datum und/oder Zeit aufgezeichnet wird). Aus diesem Grund ist eine Datumsdimension eine gemeinsame konforme Dimension. Daher sollten Sie sicherstellen, dass Ihre Datumsdimension Attribute enthält, die für die Analyse aller Faktentabellen relevant sind.

Das folgende Diagramm zeigt die `sales`-Faktentabelle und die `Inventory`-Faktentabelle. Jede Faktentabelle bezieht sich auf die `date`-Dimension und `product`-Dimension, die konforme Dimensionen sind.



Als weiteres Beispiel könnten Mitarbeiter und Benutzer die gleiche Gruppe von Personen sein. In diesem Fall kann es sinnvoll sein, die Attribute jeder Entität zu kombinieren, um eine konforme Dimension zu erzeugen.

Dimensionen mit unterschiedlichen Rollen

Wenn in einer Faktentabelle mehrmals auf eine Dimension verwiesen wird, wird sie als *Dimension mit unterschiedlichen Rollen* bezeichnet.

Wenn eine Verkaufsfaktentabelle beispielsweise über die Dimensionsschlüssel Bestelldatum,

Versanddatum und Lieferdatum verfügt, kann die Datumsdimension auf drei Arten in Beziehung gesetzt werden. Jeder Weg steht für eine bestimmte *Rolle*, aber es gibt nur eine physische Datumsdimension.

Das folgende Diagramm zeigt eine `Flight`-Faktentabelle. Die `Airport`-Dimension ist eine Dimension mit unterschiedlichen Rollen, da sie zweimal mit der Faktentabelle als `Departure Airport`-Dimension und `Arrival Airport`-Dimension verknüpft ist.



Junkdimensionen

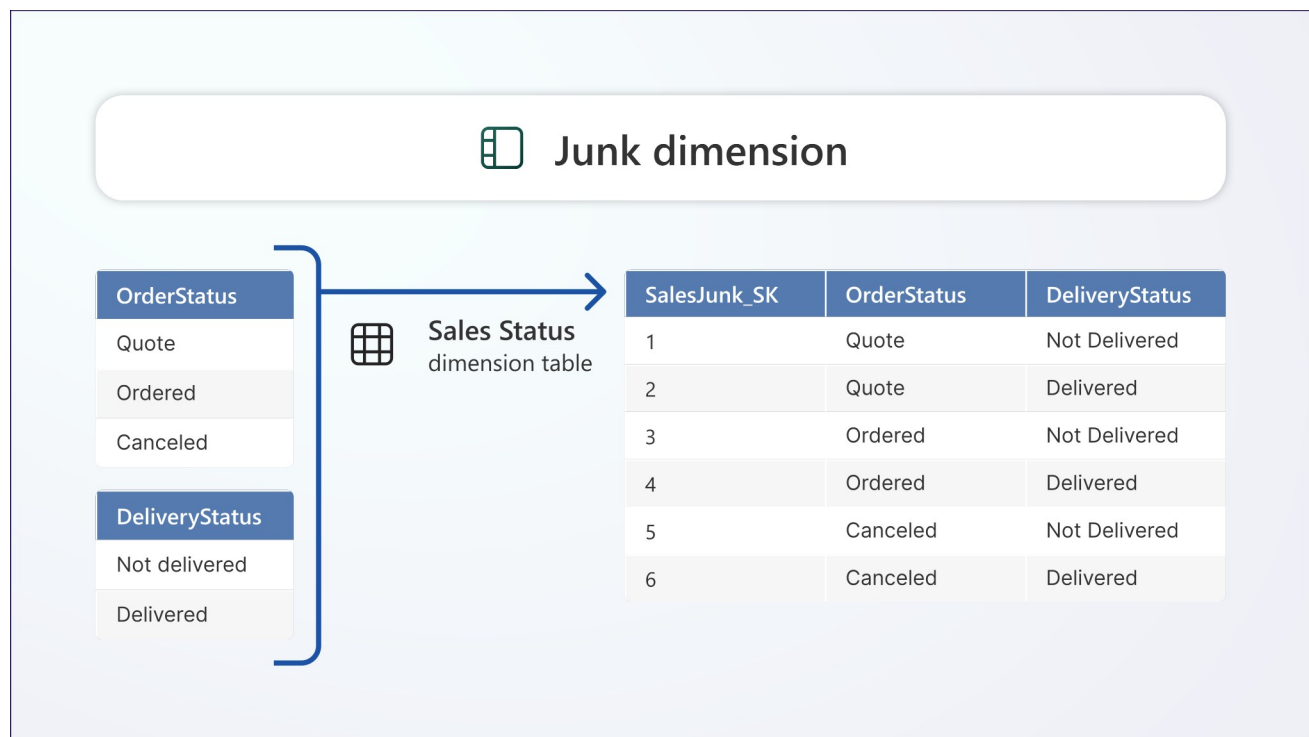
Eine *Junkdimension* ist nützlich, wenn viele unabhängige Dimensionen vorliegen, insbesondere wenn sie nur wenige Attribute (vielleicht eines) umfassen und wenn diese Attribute eine geringe Kardinalität (wenige Werte) haben. Das Ziel einer Junk-Dimension ist es, viele kleine Dimensionen in einer einzigen Dimension zu konsolidieren. Dieser Entwurfsansatz kann die Anzahl der Dimensionen verringern und die Anzahl der Faktentabellenschlüssel und somit die Größe der Faktentabellenspeicher verringern. Sie

helfen auch, den Clutter des [Datenausschnitts](#) zu reduzieren, da sie weniger Tabellen für Benutzer darstellen.

Eine Junkdimensionstabelle speichert normalerweise das kartesische Produkt aller Dimensionsattributwerte mit einem Ersatzschlüsselattribut.

Gute Kandidaten umfassen Kennzeichnungen und Indikatoren, Auftragsstatus und demografische Daten des Kunden (Geschlecht, Altersgruppe und andere).

Das folgende Diagramm zeigt eine Junk-Dimension mit dem Namen `Sales Status`, die Auftragsstatuswerte und Lieferstatuswerte kombiniert.



Degenerierte Dimensionen

Eine *degenerierte Dimension* kann auftreten, wenn die Dimension das gleiche Aggregationsintervall aufweist wie die zugehörigen Fakten. Ein gängiges Beispiel für eine degenerierte Dimension ist eine Verkaufsauftragsnummerndimension, die sich auf eine Verkaufsfaktentabelle bezieht. In der Regel ist die Rechnungsnummer ein einzelnes, nicht hierarchisches Attribut in der Faktentabelle. Daher ist es eine akzeptierte Methode, diese Daten nicht zu kopieren, um eine separate Dimensionstabelle zu erstellen.

Das folgende Diagramm zeigt eine `Sales Order`-Dimension, die eine degenerierte Dimension ist, die auf der `SalesOrderNumber`-Spalte in einer Verkaufsfaktentabelle basiert. Diese Dimension ist als Ansicht implementiert, die die eindeutigen Verkaufsauftragsnummernwerte abrufen.

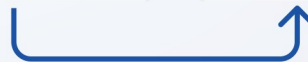


Degenerate dimension



Sales Order
dimension table (view)

...	SalesQty	Product_FK	SalesOrderNumber	SalesOrderNumber
...	20	100268	10239013	10239013
...	10	101522	10239013	10239014
...	4	103455	10239014	10239015
...



💡 Tipp

Es ist möglich, eine Ansicht in einem Fabric Warehouse zu erstellen, die die degenerierte Dimension als Dimension für Abfragezwecke darstellt.

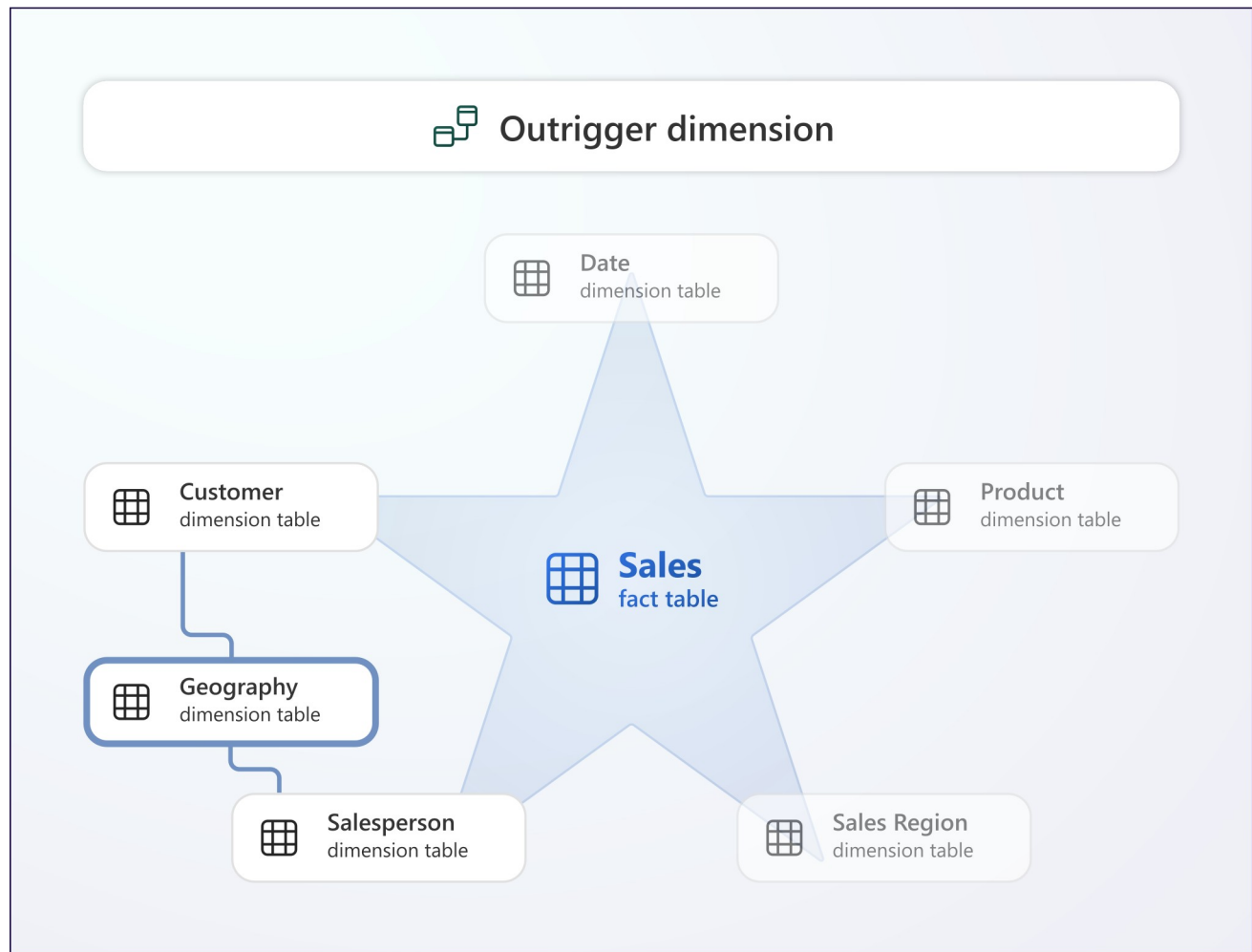
Aus Sicht der Semantikmodellierung von Power BI kann eine degenerierte Dimension mithilfe von [Power Query](#) als separate Tabelle erstellt werden. Auf diese Weise entspricht das semantische Modell der bewährten Praxis, dass Felder, die zum Filtern oder Gruppieren verwendet werden, aus Dimensionstabellen stammen und Felder, die zur Zusammenfassung von Fakten verwendet werden, aus Faktentabellen stammen.

Outrigger-Dimensionen

Wenn sich eine Dimensionstabelle auf andere Dimensionstabellen bezieht, wird sie als *Outrigger-Dimension* bezeichnet. Eine Outrigger-Dimension kann dazu beitragen, Definitionen im dimensional Modell zu erfüllen und wiederzuverwenden.

Sie können z. B. eine geografische Dimension erstellen, in der geografische Standorte für jede Postleitzahl gespeichert werden. Diese Dimension könnte dann von Ihrer Kundendimension *und* Vertriebsmitarbeiterdimension referenziert werden, wodurch der Ersatzschlüssel der geografischen Dimension gespeichert wird. Auf diese Weise können Kunden und Vertriebsmitarbeiter dann mithilfe konsistenter geografischer Standorte analysiert werden.

Das folgende Diagramm zeigt eine Geography-Dimension, die eine Outrigger-Dimension ist. Sie bezieht sich nicht direkt auf die Sales-Faktentabelle. Stattdessen besteht ein indirekter Zusammenhang über die Dimension Customer und die Dimension Salesperson.



Berücksichtigen Sie, dass die Datumsdimension als Outrigger-Dimension verwendet werden kann, wenn andere Dimensionstabellenattribute Datumsangaben speichern. Beispielsweise könnte das Geburtsdatum in einer Kundendimension mithilfe des Ersatzschlüssels der Datumsdimensionstabelle gespeichert werden.

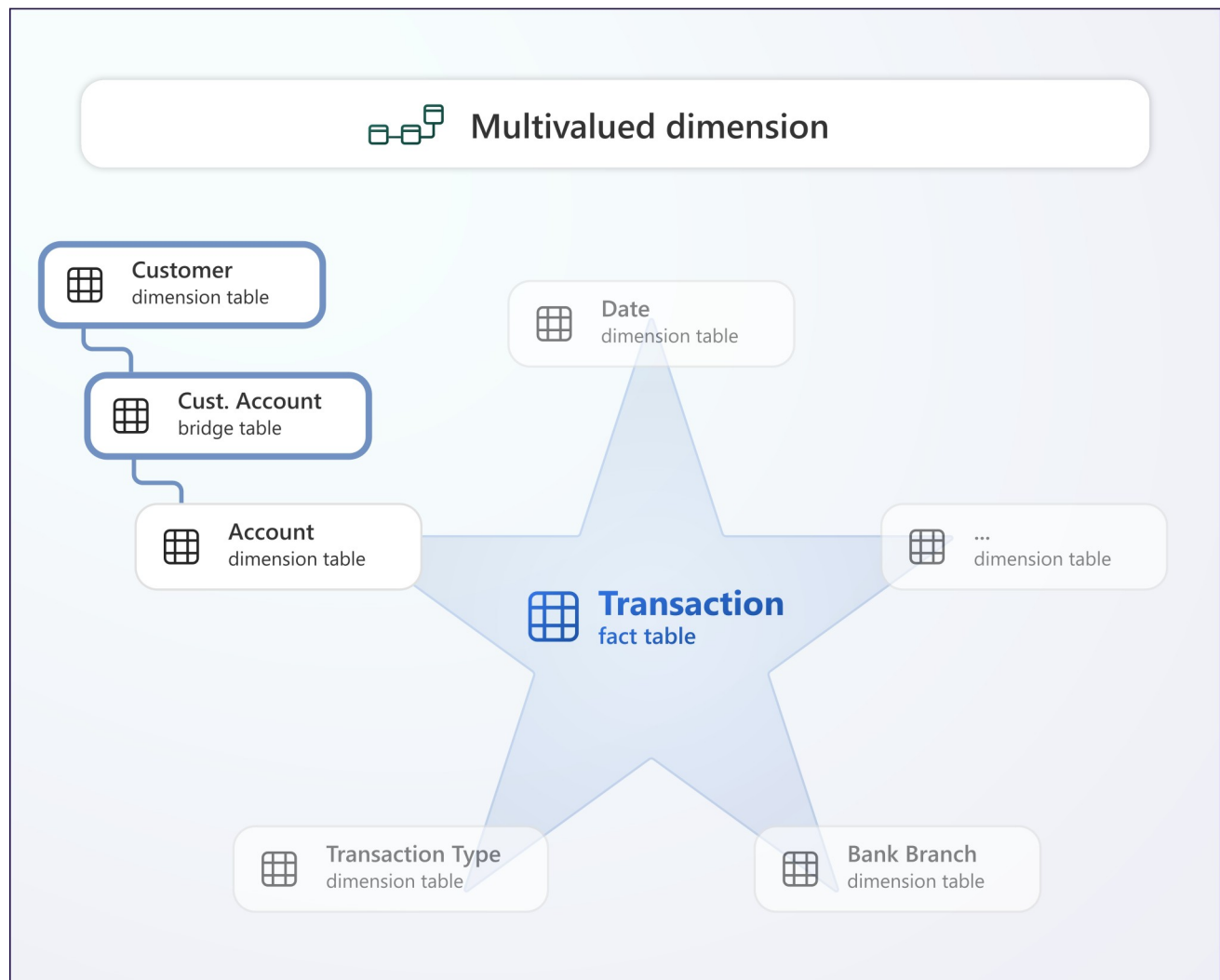
Mehrwertige Dimensionen

Wenn ein Dimensionsattribut mehrere Werte speichern muss, müssen Sie eine *mehrwertige Dimension* entwerfen. Sie implementieren eine mehrwertige Dimension, indem Sie eine *Brückentabelle* erstellen (manchmal auch als *Verknüpfungstabelle* bezeichnet). Eine Brückentabelle speichert eine m:n-Beziehung zwischen Entitäten.

Angenommen, es gibt eine Vertriebsperson-Dimension und jeder Vertriebsmitarbeiter wird einer oder mehreren Vertriebsregionen zugewiesen. In diesem Fall ist es sinnvoll, eine Vertriebsregionsdimension zu erstellen. Diese Dimension speichert jede Vertriebsregion nur einmal. Eine separate Tabelle, die als Brückentabelle bezeichnet wird, speichert eine Zeile für

jeden Vertriebsmitarbeiter und jede Vertriebsregionsbeziehung. Physisch gibt es eine 1:n-Beziehung von der Vertriebsmitarbeiter-Dimension zur Brückentabelle und eine weitere 1:n-Beziehung aus der Vertriebsregion-Dimension zur Brückentabelle. Es besteht eine m:n-Beziehung zwischen Vertriebsmitarbeitern und Vertriebsregionen.

Im folgenden Diagramm bezieht sich die Account -Dimensionstabelle auf die Transaction - Faktentabelle. Da Kunden mehrere Konten haben können und Konten mehrere Kunden haben können, ist die customer -Dimensionstabelle über die Customer Account - Brückentabelle verknüpft.



Zugehöriger Inhalt

Im nächsten Artikel dieser Reihe erfahren Sie mehr über Anleitungen und bewährte Entwurfsverfahren für [Faktentabellen](#).