



# Power BI Optimierung

DataSyn GmbH



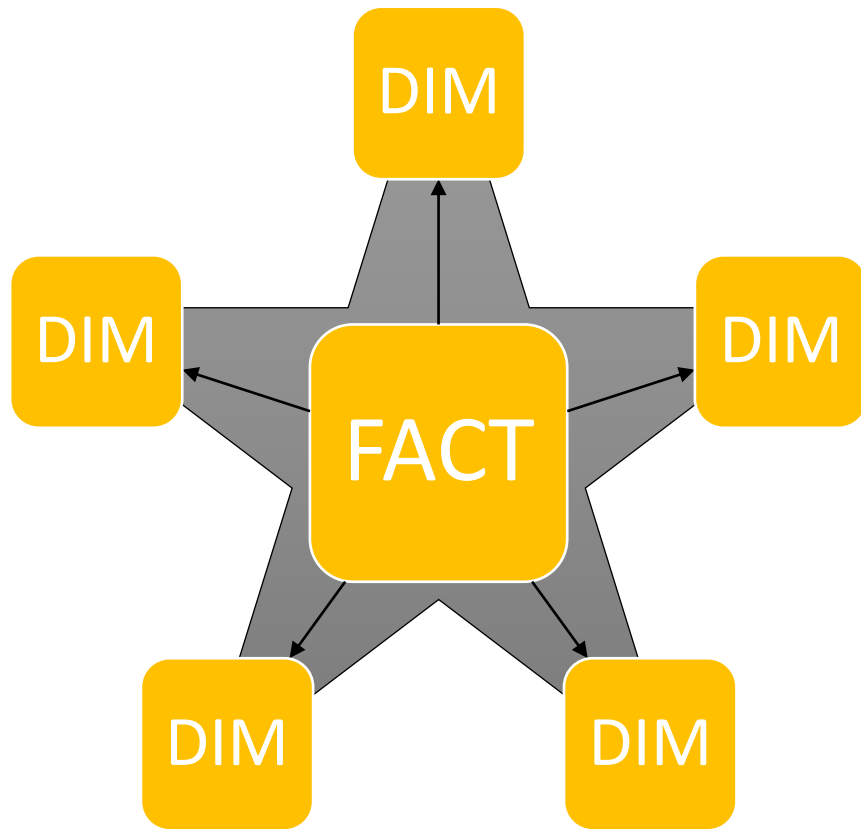


# Data Model

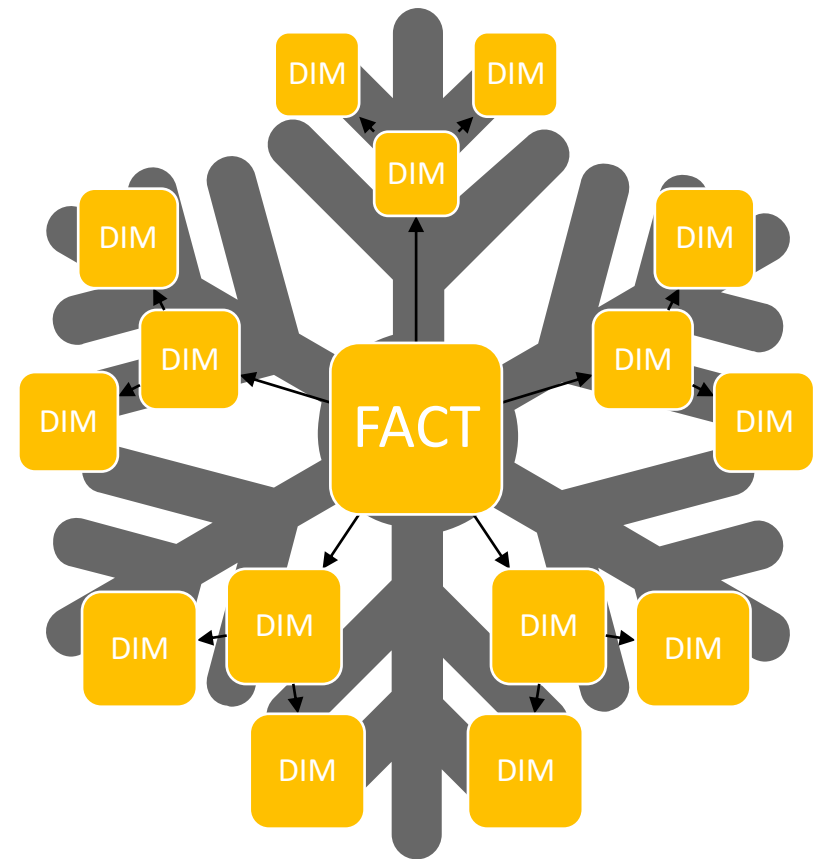
- Star Schema!
- Große Tabellen aufteilen (normalisieren)
- Bi-direktionales Filtern & m:n Beziehungen vermeiden
- Keep it simple!

# Data Model: Star Schema vs Snowflake

Star Schema



vs Snowflake Schema



# Power Query

- Anbindung an richtige Datenquelle
  - Data Warehousing
  - Direct Query?
  - SQL native Query
- Daten bereinigen
- Schritte komprimieren
- PBI kann problemlos Millionen von Zeilen verarbeiten, aber die Transformationen kosten Leistung!
- Das Erstellen von aggregierten Tabellen wird die Leistung erhöhen
- = Table.Buffer(TableName): Speichert Daten zwischen und kann Query beschleunigen
- Deaktivieren von "Laden aktivieren", um zu verhindern, dass irrelevante Tabellen in das Datenmodell geladen werden und RAM verbrauchen

# Power Query: SQL native query

## Classic M Query

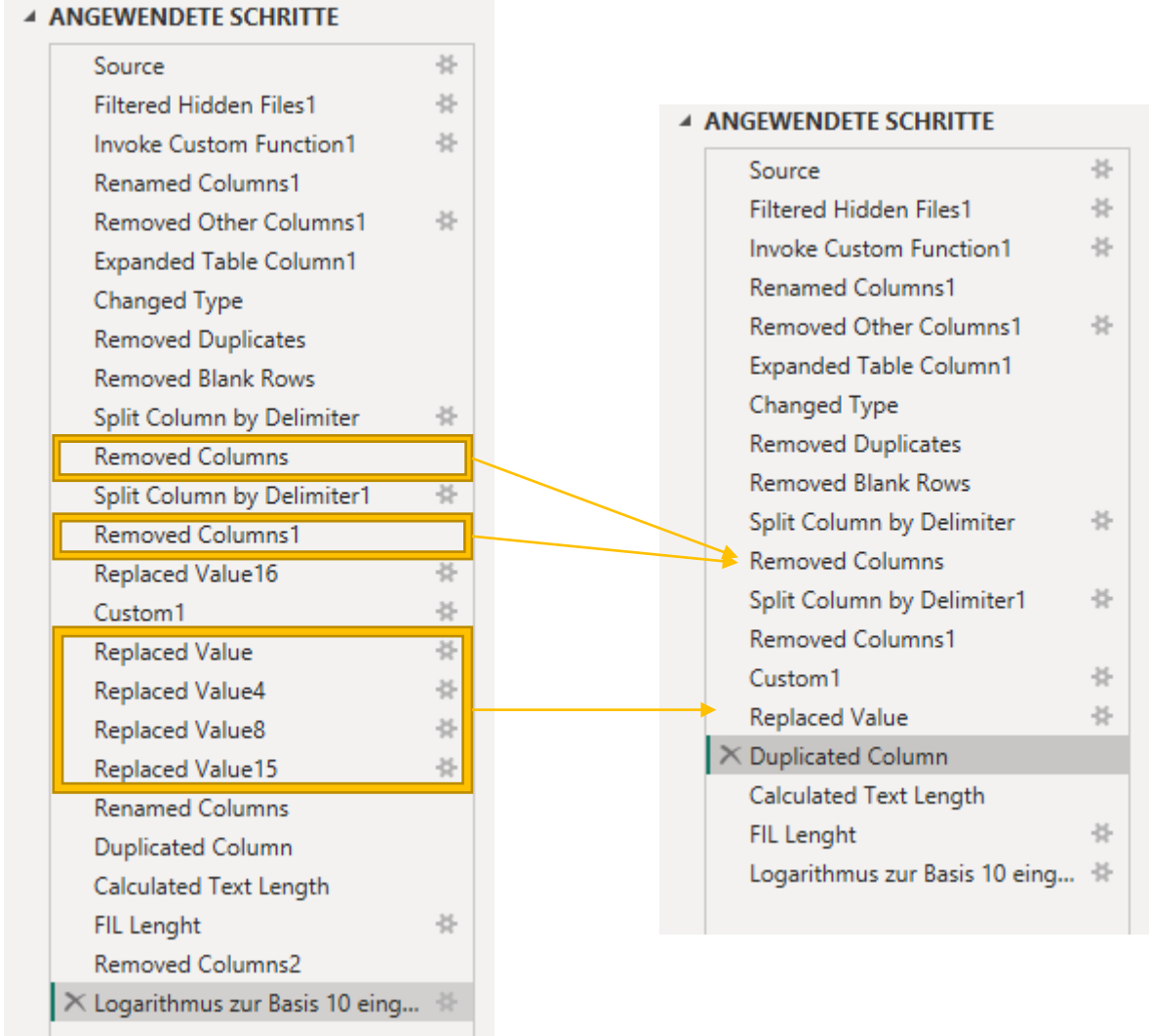
```
let
    Source = Sql.Databases("Demo"),
    ContosoRetailDW = Source{[Name="ContosoRetailDW"]}[Data],
    DataModeling_Customer =
    ContosoRetailDW{[Schema="DataModeling",Item="Customer"]}[Data]
in
    DataModeling_Customer
```



## Native Query

```
let
    Source = Sql.Database("ContosoRetailDW.database.windows.net",
    "Demo", [Query = "SELECT * FROM DataModel.Customer WHERE City
    IS Tokyo"])
```

# Power Query: Abfrage Komprimierung



# DAX

- Komplexe Measures & Aggregationen vermeiden
- Großes Modell mit einfachem DAX-Code besser als kleineres Modell mit komplexem DAX-Code!



# Report

- Seiten klein halten
- Wenige aussagekräftige Visuals nutzen
- Wenige Slicer einbauen
- Unnötige Interaktionen zwischen Visuals vermeiden
- Keine Redundanzen

