

Power BI Optimization

DataSyn GmbH



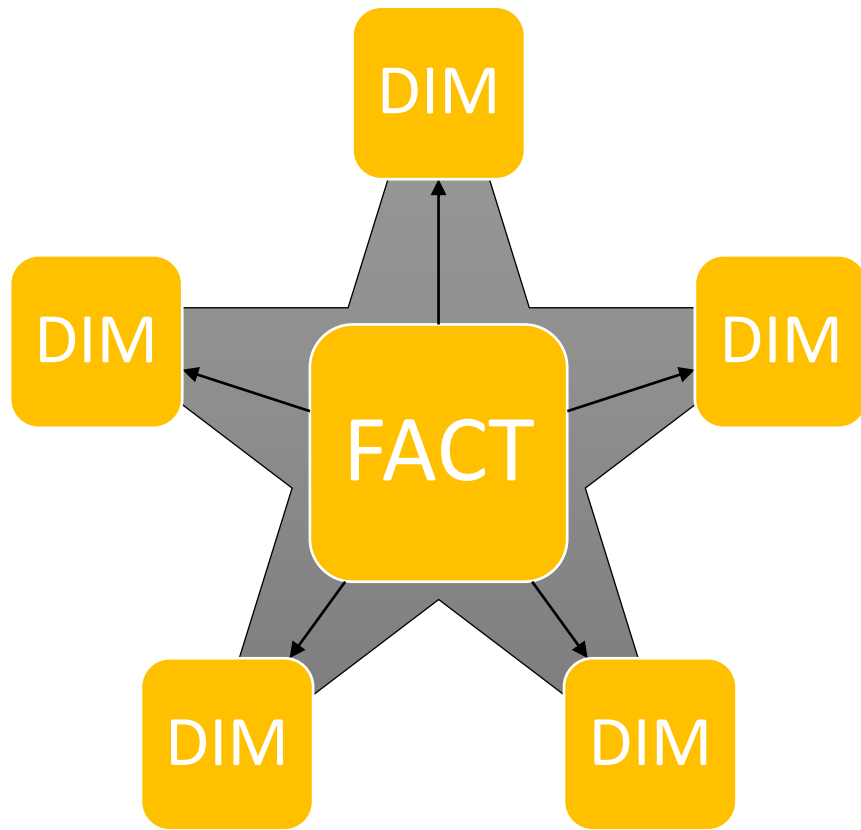


Data Model

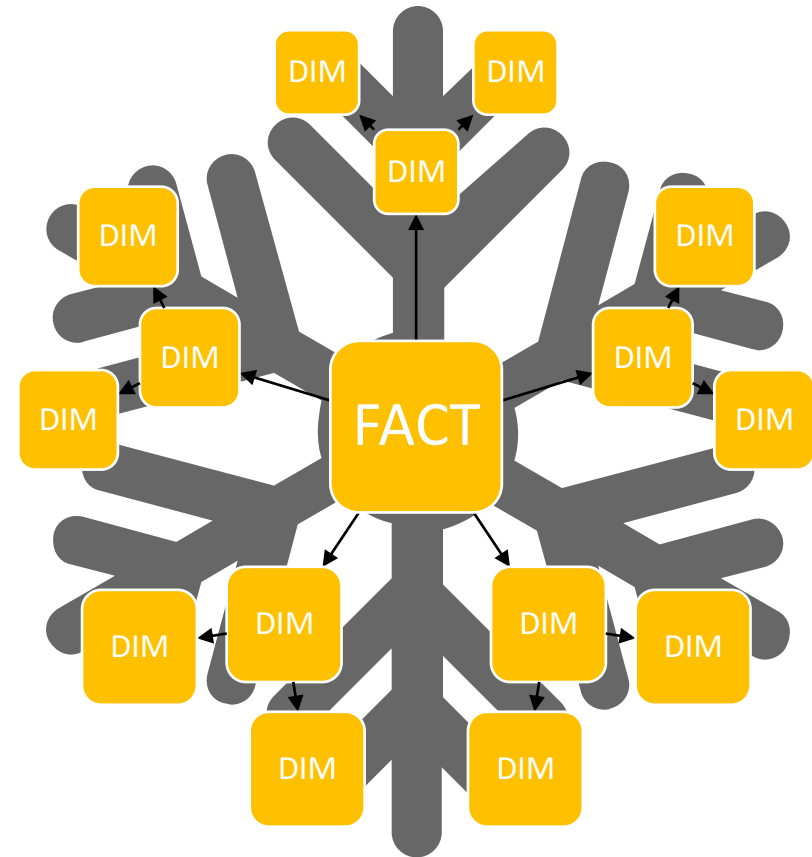
- Star Schema!
- Split large tables (normalize)
- Bidirectional filtering and avoidance of m:n relationships
- Keep it simple!

Data Model: Star Schema vs Snowflake

Star Schema



vs Snowflake Schema





Power Query

- Use the right data source
 - Data Warehousing
 - Direct Query?
 - SQL native Query
- Clean data
- Compress steps
- PBI can easily handle millions of rows, but the transformations cost performance!
- Creating aggregated tables will increase performance
- = Table.Buffer(TableName): saves a table in memory and can accelerate the query
- Disable „Enable load“ to prevent unwanted tables to be loaded into the data model and consume memory

Power Query: SQL native query

Classic M Query

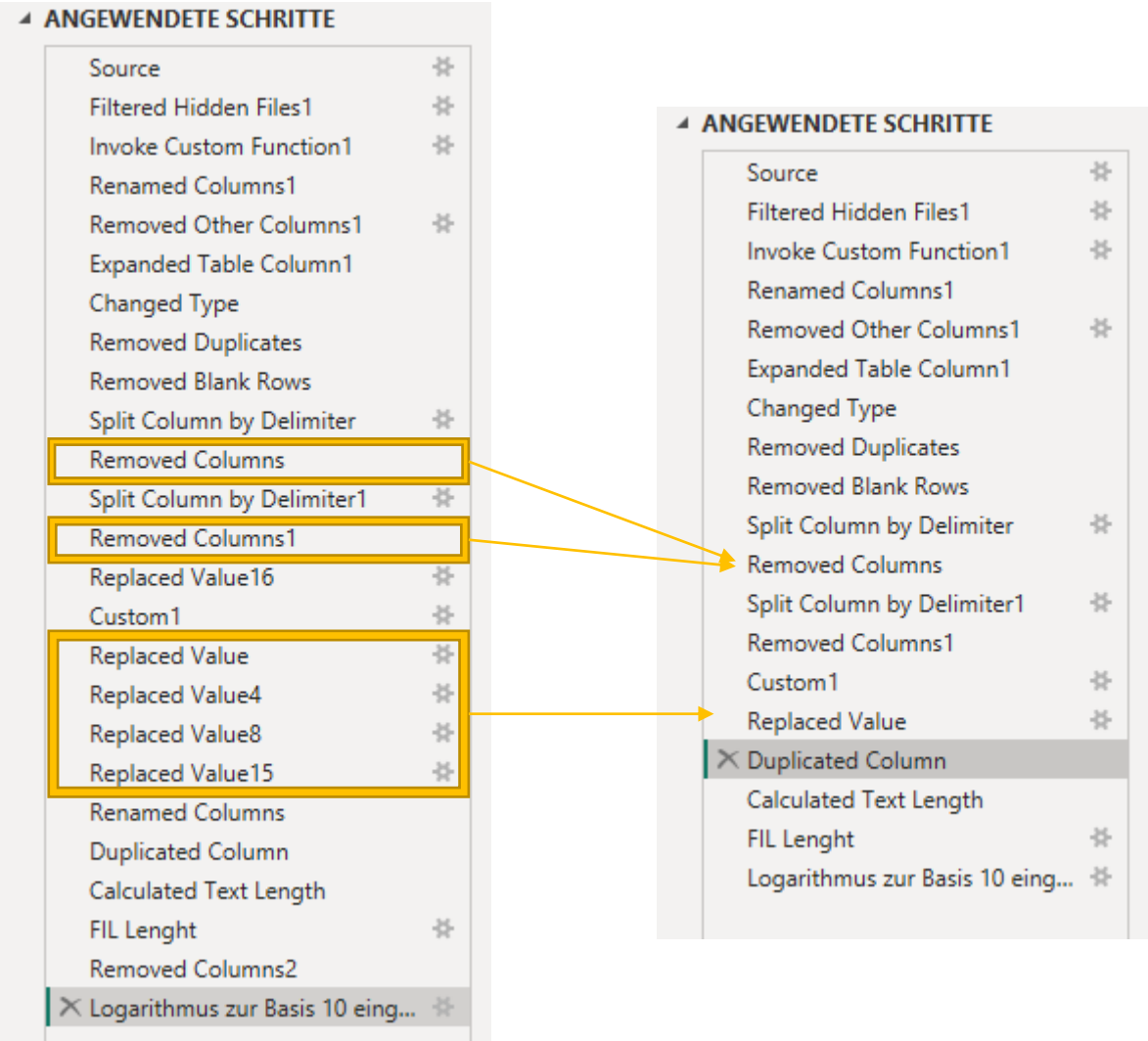
```
let
    Source = Sql.Databases("Demo"),
    ContosoRetailDW = Source{[Name="ContosoRetailDW"]}[Data],
    DataModeling_Customer =
    ContosoRetailDW{[Schema="DataModeling",Item="Customer"]}[Data]
in
    DataModeling_Customer
```



Native Query

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

Power Query: query compression



DAX

- Avoid complex measures & aggregations
- Large model with simple DAX code better than smaller model with complex DAX code!



Report

- Keep pages small
- Use few meaningful visuals
- Limit the number of slicers
- Avoid unnecessary interactions between visuals
- No redundancies

