

**Hinweise:**

Ein Einführungsvideo zur Arbeit mit dem SQL Server Management Studio finden Sie im OPAL-Kurs unter: *Praktikum => Lehrvideos => 01 - Demonstration SQL Server Management Studio*

Die für das Praktikum erforderlichen Softwarekomponenten sind im Labor Z136 B installiert und können für die Lösung der Aufgabenstellungen verwendet werden.

Für die Arbeit von zu Hause befolgen Sie bitte die im Wiki des OPAL-Kurses beschriebene Verfahrensweise. Diese finden Sie im OPAL-Kurs Business Intelligence (I440):

Kurzlink: <https://t1p.de/OpalBI>

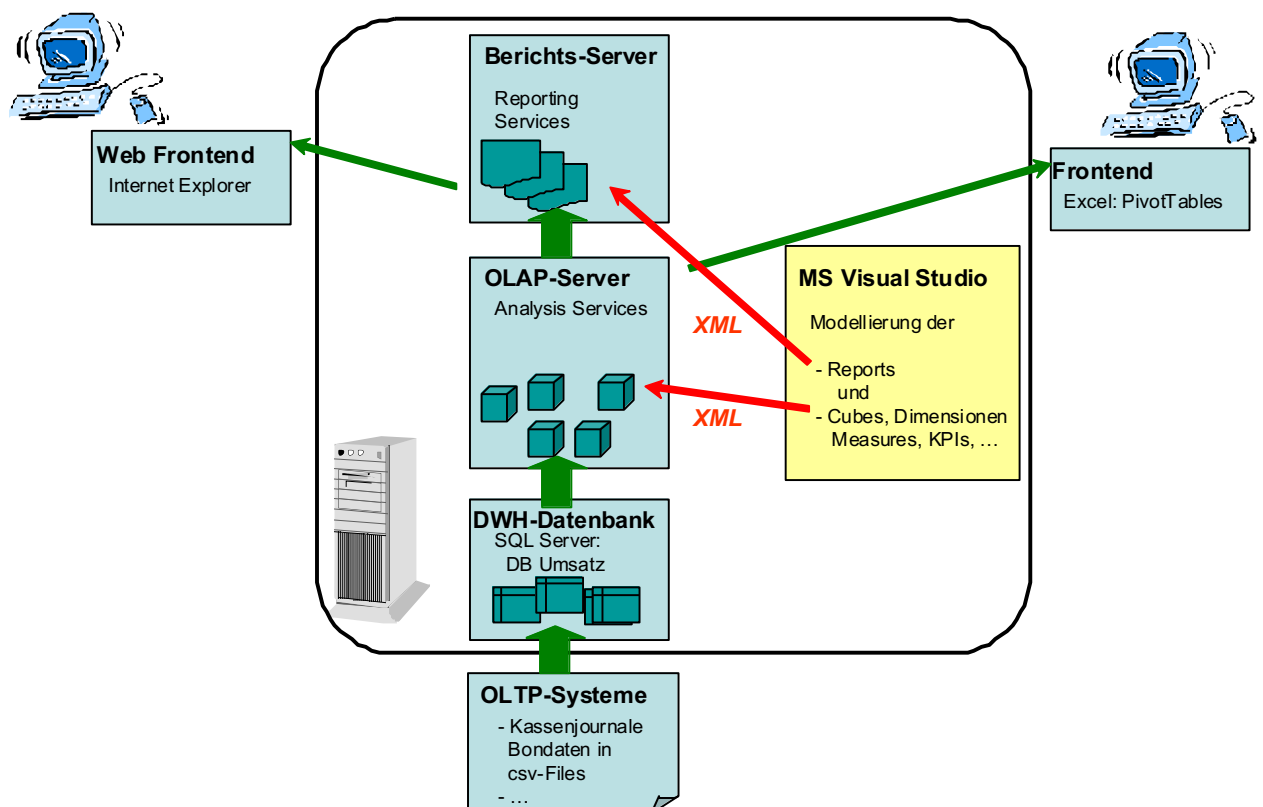
unter: *Wiki => Arbeiten von zu Hause*

## Aufgabensammlung

### zum Praktikum

### Business Intelligence

### Sommersemester 2023



## Aufgabenkomplex A: Data Warehouse in MS Analysis Services

### A.1. Erstellung der Datenbank für das Data Warehouse

Im Praktikum sollen die Daten für das Data Warehouse auf einem MS SQL Server gehalten werden. Dazu ist eine Datenbank zu entwerfen, welche sowohl Dimensionen nach dem Star Schemas als auch Dimensionen nach dem Snowflake Schema enthält. Die Daten für die Fakttable und die Dimensionstabellen liegen teilweise aufbereitet in einzelnen Quelldateien vor.

**Dokumentieren Sie alle Befehle in einer Skriptdatei im SQL Server Management Studio!**

A1.1. Loggen Sie sich auf dem Datenbankserver 141.56.2.45 (MS SQL Server) ein und gehen Sie zu Ihrer Datenbank aus der Lehrveranstaltung Datenbanksysteme II.

A1.2. Erstellen Sie folgende Tabellen in der Datenbank entsprechend der angegebenen Tabellendefinition.

Tabellenname	Feldname	Datentyp	Fremdschlüssel auf
Geografie	<b>Land_ID</b>	VARCHAR (2)	
	Bundesland	VARCHAR (25)	
	Region	VARCHAR (25)	
	Staat	VARCHAR (25)	
MitarbeiterShop	<b>Mitarbeiter_ID</b>	VARCHAR (2)	
	Name	VARCHAR (25)	
Produktkategorie	<b>Kategorie_ID</b>	VARCHAR (2)	
	Kategorie	VARCHAR (25)	
	Kategorie_Manager	VARCHAR (25)	
Produktsubkategorie	<b>Subkategorie_ID</b>	VARCHAR (3)	Produktkategorie. Kategorie_ID
	Subkategorie	VARCHAR (25)	
	Subkategorie_Manager	VARCHAR (25)	
	Kategorie_ID	VARCHAR (2)	
Produkt	<b>Produkt_ID</b>	VARCHAR (4)	Produktsubkategorie. Subkategorie_ID
	Markenname	VARCHAR (25)	
	Produktname	VARCHAR (40)	
	Preis	MONEY	
	Subkategorie_ID	VARCHAR (3)	
Zeit	<b>Mon_ID</b>	VARCHAR (6)	
	Monatsname	VARCHAR (20)	
	Q_ID	VARCHAR (6)	
	Quartal	VARCHAR (10)	
	Jahr	VARCHAR (4)	
Umsatzdaten	<b>Mon_ID</b>	VARCHAR (6)	Zeit.Mon_ID Geografie.Land_ID Produkt.Produkt_ID MitarbeiterShop. Mitarbeiter_ID
	<b>Land_ID</b>	VARCHAR (2)	
	<b>Produkt_ID</b>	VARCHAR (4)	
	<b>Mitarbeiter_ID</b>	VARCHAR (2)	
	Umsatzbetrag	MONEY	
	Umsatzmenge	INTEGER	

- A1.3. Die Inhalte der einzelnen Dimensionstabellen stehen als xls-Dateien im OPAL-Kurs *Business Intelligence (I440)*:

Kurzlink: <https://t1p.de/OpalBI>

unter *Praktikum* => *Praktikumsunterlagen* => *DWH*  
zur Verfügung:

<b>Tabelle</b>	<b>xls-Datei</b>
Geografie	Geografie.xls
MitarbeiterShop	Mitarbeiter.xls
Produkt	Produkt.xls
Produktsubkategorie	Produktsubkategorie.xls
Produktkategorie	Produktkategorie.xls

Kopieren Sie die xls-Dateien in ein lokales Verzeichnis auf Ihrem Computer. Erzeugen Sie jeweils in der xls-Datei in der ersten Spalte hinter den Daten einen SQL INSERT-Befehl für die Tabelle.

Hinweis: Verwenden Sie den Excel-Operator & oder die Excel-Funktion *=Verketten(...)* zum Erzeugen des INSERT-Befehls und die Funktion *Wechseln(...)* zum Anpassen des Dezimaltrennzeichens. Informieren Sie sich in der Excel-Hilfe über deren Wirkungsweise.

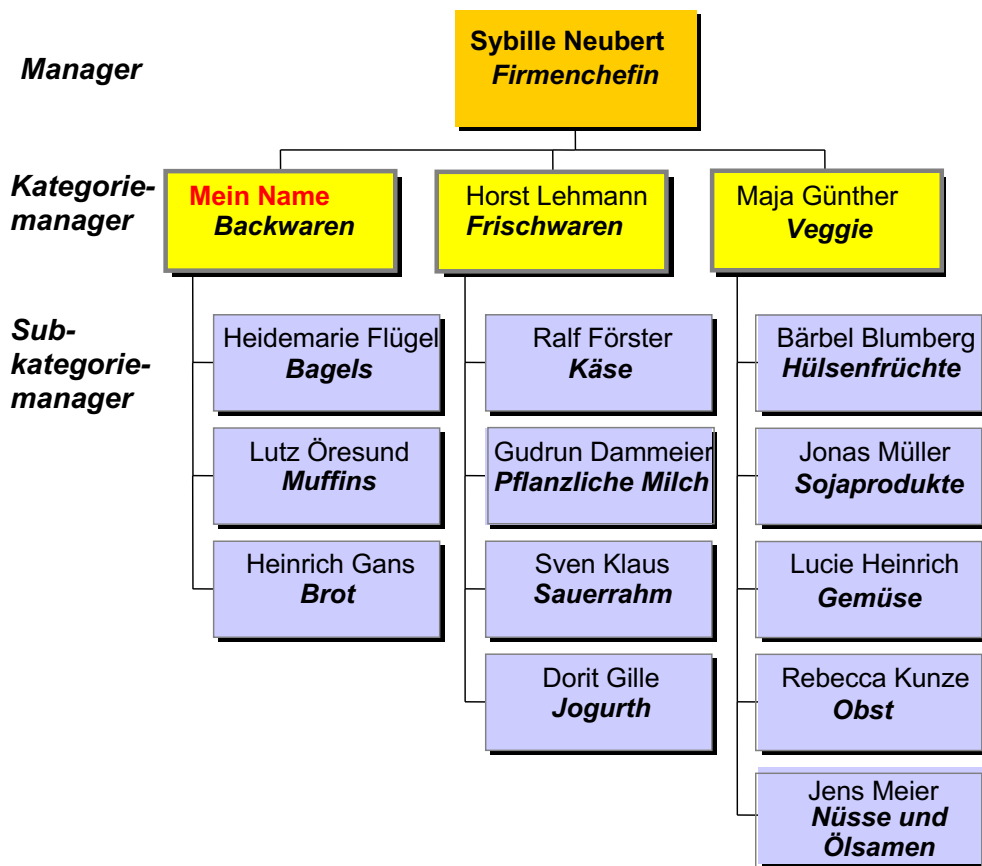
Kopieren Sie die in Excel erzeugten INSERT-Befehle (Spalte mit den Befehlen) in ihre im SQL Server Management Studio geöffnete Skriptdatei und führen Sie die INSERT-Befehle aus.

- A1.4. Fügen Sie in die Tabelle *Zeit* folgende Datensätze ein.

<b>Mon_ID</b>	<b>Monatsname</b>	<b>Q_ID</b>	<b>Quartal</b>	<b>Jahr</b>
202101	Januar	202101	1. Quartal	2021
202102	Februar	202101	1. Quartal	2021
202103	März	202101	1. Quartal	2021
202104	April	202102	2. Quartal	2021
202105	Mai	202102	2. Quartal	2021
202106	Juni	202102	2. Quartal	2021
202107	Juli	202103	3. Quartal	2021
202108	August	202103	3. Quartal	2021
202109	September	202103	3. Quartal	2021
202110	Oktober	202104	4. Quartal	2021
202111	November	202104	4. Quartal	2021
202112	Dezember	202104	4. Quartal	2021
202201	Januar	202201	1. Quartal	2022
202202	Februar	202201	1. Quartal	2022
202203	März	202201	1. Quartal	2022
202204	April	202202	2. Quartal	2022
202205	Mai	202202	2. Quartal	2022
202206	Juni	202202	2. Quartal	2022

Mon_ID	Monatsname	Q_ID	Quartal	Jahr
202207	Juli	202203	3. Quartal	2022
202208	August	202203	3. Quartal	2022
202209	September	202203	3. Quartal	2022
202210	Oktober	202204	4. Quartal	2022
202211	November	202204	4. Quartal	2022
202212	Dezember	202204	4. Quartal	2022

- A1.5. Die organisatorische Struktur des Unternehmens ist hierarchisch aufgebaut. Nachfolgend ist die Hierarchie der Mitarbeiter dargestellt. Es sind die *Mitarbeiter\_ID* und die Verantwortlichkeiten der Mitarbeiter für die einzelnen Produktkategorien bzw. -subkategorien ausgewiesen.



Tragen Sie sich als Kategoriemanager für die Kategorie Backwaren ein.

Ändern Sie dazu den Namen des Mitarbeiters mit der *Mitarbeiter\_ID* = '02' in Tabelle *Mitarbeitershop* in Ihren Namen.

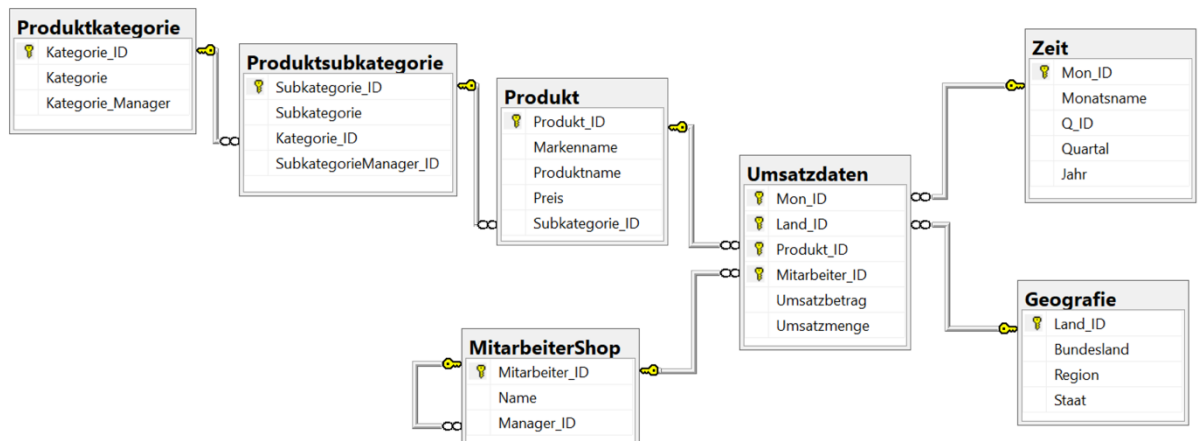
Ändern Sie weiterhin den Kategoriemanager mit der *Kategorie\_ID* = '01' in Tabelle *Produktkategorie* in Ihren Namen.

Die Mitarbeiterhierarchie soll über eine Parent-Child-Beziehung in der Tabelle *MitarbeiterShop* realisiert werden. Fügen Sie dazu in die Tabelle *MitarbeiterShop* eine neue Spalte *Manager\_ID* ein und ergänzen Sie die Datensätze, um die dargestellte Hierarchie abzubilden.

Stellen Sie im Datenbankschema sicher, dass nur eine gültige *Mitarbeiter\_ID* als *Manager\_ID* vergeben werden darf (Fremdschlüssel definieren).

Hinweis: Der Datensatz für die Firmenchefin „Sybille Neubert“ verweist auf sich selbst (d.h. *Manager\_ID* = '01').

- A1.6. Erstellen Sie im SQL Server Managementstudio in Ihrer Datenbank ein neues Datenbankdiagramm mit Namen *DWH-Schema*, das die folgenden Tabellen enthält.



- A1.7. Die OLTP-Daten der Belegerfassung (Kassenbons) befinden sich in den 8 csv-Dateien im für die acht Quartale des Betrachtungszeitraums. Je Quartal existiert eine Datei mit dem Dateinamen *Belege\_YYYY\_Q.csv*. Mit den in diesen Dateien enthaltenen Daten soll die Faktentabelle *Umsatzdaten* des Data Warehouse gefüllt werden.

Die acht csv- Dateien befinden sich OPAL-Kurs Business Intelligence (I440):

Kurzlink: <https://t1p.de/OpalBI>

unter *Praktikum* => *Praktikumsunterlagen* => *DWH*.

Kopieren Sie die Dateien *Belege\_YYYY\_Q.csv* in ein lokales Verzeichnis auf Ihrem Computer und analysieren Sie deren Aufbau und die Anzahl der darin enthaltenen Datensätze.

- A1.8. Um die Daten aus den csv-Dateien in die Faktentabelle *Umsatzdaten* übernehmen zu können, muss eine Transformation der Daten durchgeführt werden. Die Transformation soll für jedes der 8 Quartale mit Hilfe der Tabellen *BelegeTMP* und *UmsatzdatenTMP* erfolgen. Die Tabelle *BelegeTMP* dient der Übernahme der Daten aus der jeweiligen csv-Datei in die Datenbank. In der Tabelle *UmsatzdatenTMP* werden die in die Data Warehouse-Struktur konvertierten Daten abgelegt.

Führen Sie dazu folgende Schritte aus und beachten Sie die unten auf der nächsten Seite angegebenen Hinweise:

- Erstellen Sie in Ihrer Datenbank eine Tabelle *BelegeTMP*, mit den entsprechenden Spalten, die die Daten aus den csv-Dateien aufnehmen können. Achten Sie auf die richtige Reihenfolge der Spalten und die Wahl sinnvoller Datentypen. Erstellen Sie weiterhin eine Tabelle *UmsatzdatenTMP*, deren Struktur der Faktentabelle *Umsatzdaten* entspricht.

- b. Die Daten aus den csv-Dateien sollen in die Faktentabelle Umsatzdaten übernommen werden. Dem entsprechend ist folgender Ablauf durchzuführen:
1. Import der Daten aus der csv-Datei in die Tabelle *BelegeTMP*.
  2. Übernahme der Daten aus der Tabelle *BelegTMP* in die Tabelle *UmsatzdatenTMP* und Transformation der Daten
  3. Weitere Anpassung der Daten in der Tabelle *UmsatzdatenTMP* (Ändern der *Mitarbeiter\_ID* in der Tabelle *UmsatzdatenTMP* entsprechend der Zuständigkeit von Mitarbeitern für die Produktsubkategorien als Sortimentsverantwortliche):  
  
Hinweise: Erweitern Sie dazu die Tabelle *Produktsubkategorie* um eine Spalte zur Ablage der *Mitarbeiter\_ID* des Mitarbeiters, der für diese Subkategorie der Sortimentsverantwortliche ist und erstellen Sie anschließend eine korrelierte Unterabfrage zum Setzen des *Mitarbeiter\_ID* in der Tabelle *UmsatzdatenTMP*.
  4. Übernahme der Daten aus der Tabelle *UmsatzdatenTMP* in die Faktentabelle *Umsatzdaten*.

Hinweise:

- Verwenden Sie zur Datenübernahme aus der csv-Datei (Punkt 1.) den Befehl: BULK INSERT ... mit den Optionen FIRSTROW, FIELDTERMINATOR und ROWTERMINATOR .

```
BULK INSERT
[database_name. [ schema_name ]. | schema_name .]
[ table_name | view_name ]
FROM 'data_file'
[ WITH (
[ [ , ] FIELDTERMINATOR = 'field_terminator' ]
[ [ , ] FIRSTROW =first_row ]
[ [ , ] ROWTERMINATOR = 'row_terminator' ]
)]
```

- Geben Sie im BULK INSERT als data file '**D:\BI\Belege\_YYYY\_Q.csv**' an. Dies ist ein Verzeichnis auf dem Datenbankserver, in dem die 8 Quelldateien ebenfalls stehen. Verwenden Sie **nicht** ihr lokales Verzeichnis. Vom lokalen Verzeichnis ist kein Import möglich.
- Nutzen Sie für die Datenübernahme zwischen den Tabellen (Punkte 2. und 4.) den SQL-Befehl: INSERT INTO ... SELECT ... FROM ...
- Die ersten beiden Stellen der Spalte *Fil\_ID* in den csv-Dateien der Belege bilden die *Land\_ID* für die Tabelle *Umsatzdaten*. (Punkt 2.)
- Verwenden Sie zur Berechnung der *Mon\_ID* aus dem Feld *Datum* die SQL-Funktionen DATEPART.

- A1.9. Überprüfen Sie, ob die Faktentabelle *Umsatzdaten* nach Abschluss des Datenbankimports für alle 8 Quartale nun **164.128** Datensätze enthält, der Umsatzbetrag **9.125.929,37 €** beträgt und der **Preis** der Produkte **plausible Werte** enthält.

**Hinweise:**

Eine Einführungsvideo zur Arbeit mit dem Visual Studio finden Sie im OPAL-Kurs

unter: *Praktikum => Lehrvideos => 02 - Demonstration Visual Studio Projekt*

Die für das Praktikum erforderlichen Softwarekomponenten sind im Labor Z136 B installiert und können für die Lösung der Aufgabenstellungen verwendet werden.

Für die Arbeit von zu Hause befolgen Sie bitte die im Wiki des OPAL-Kurses beschriebene Verfahrensweise über den Terminalserver der HTW Dresden.

Diese finden Sie im OPAL-Kurs Business Intelligence (I440):

Kurzlink: <https://t1p.de/OpalBI>

unter: *Wiki => Arbeiten von zu Hause => Möglichkeit – Login am Terminalserver*

**A2. Erstellung der Cubes im OLAP Server**

Für die Aufbereitung und Anreicherung der in der Data Warehouse Datenbank gehaltenen Daten und die Bereitstellung für Front-End-Anwendungen sollen als OLAP Server die Analysis Services zum Einsatz kommen.

**Anlegen von Datenquelle und Datenquellsicht**

A2.1. Starten Sie das Visual Studio 2021 und legen Sie ein neues BI-Projekt mit dem Namen *Umsatz\_Sxxxxx* an (Menüpunkt *Datei - Neu - Projekt*).

- Vorlage: Business Intelligence
- Typ: Analysis Services-Projekt für mehrdimensionale und Data Mining-Projekte
- Projektname: Umsatz\_Sxxxxx  
*Ersetzen Sie dabei Sxxxxx durch Ihre Bibliotheksnummer*
- Ort: Desktop (oder Nutzerverzeichnis)
- Name der Projektmappe: Umsatz\_Sxxxxx

Hinweis: Bevor Sie sich vom Terminalserver abmelden, sichern Sie bitte das Visual Studio Projekt in Ihr Homeverzeichnis (samba).

A2.2. Legen Sie eine neue Datenquelle an (Projektmappenexplorer: *Datenquellen – Kontextmenü [rechte Maustaste] - Neue Datenquelle*). Button *Neu*

- Anbieter: SQL Server Nativ Client 11.0
- Servername: 141.56.2.45
- Windows-Authentifizierung verwenden: markieren
- Datenbankname: Wählen Sie den Namen Ihrer Datenbank auf dem SQL-Server. Der Name beginnt mit *iw18* oder *gm19* Gefolgt von Ihrer Bibliotheksnummer *sxxxxx*

Hinweis: Mit dem Button *Verbindung testen* können Sie überprüfen, ob die Verbindung zur Datenquelle funktioniert.

Button: *OK*

Button: *Weiter*


- Benutzername: Ihr Anmeldename an der SMB Domain
- Kennwort Ihr Passwort

Button: *Weiter*

- Datenquellenname: UmsatzSource

Button: *Fertigstellen*

A2.3. Legen Sie eine neue Datenquellensicht an (Projektmappenexplorer: *Datenquellensichten* – *Kontextmenü [rechte Maustaste] - Neue Datenquellensicht*). Button *Weiter*

- Eingeschlossene Objekte: Geografie  
(übernehmen mit  ) Produkt  
Produktkategorie  
Produktsubkategorie  
MitarbeiterShop  
Zeit  
Umsatzdaten

Button: *Weiter*

- Name: UmsatzView

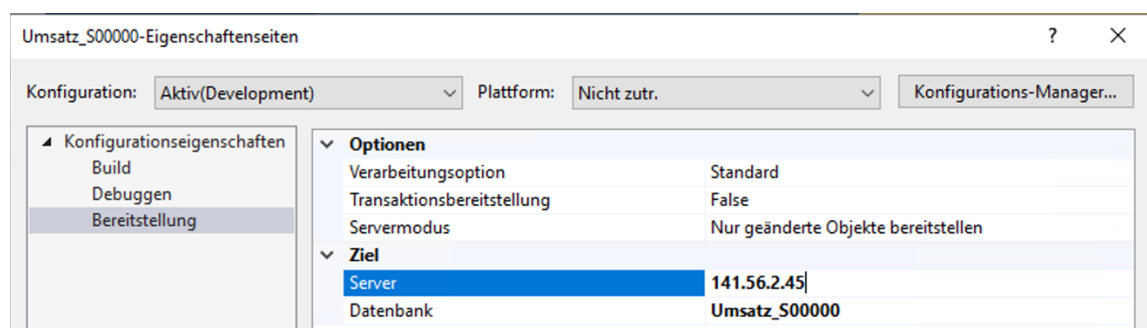
Button: *Fertigstellen*

### Eigenschaften des Projektes anpassen

A2.4. Der OLAP-Server befindet sich ebenfalls auf dem Server mit der IP-Adresse **141.56.2.45**. Damit das Projekt auf diesen Server bereitgestellt wird, muss diese IP-Adresse in den Eigenschaften des Projekts hinterlegt werden.

Wählen Sie dafür im Projektmappenexplorer den Projektnamen **Umsatz\_Sxxxx** und Öffnen Sie über das *Kontextmenü [rechte Maustaste]* die *Eigenschaften*.

Geben Sie unter *Bereitstellung* => *Server* die IP-Adresse an:





## Anlegen der Dimensionen

### ➤ Dimension Zeit

A2.5. Legen Sie eine neue Dimension an (Projektmappenexplorer: *Dimensionen – Kontextmenü [rechte Maustaste] - Neue Dimension*).

- Vorhandene Tabelle verwenden: markieren
- Vorlage: (keine)

Button *Weiter*

- Datenquellensicht: UmsatzView
- Haupttabelle: Zeit
- Schlüsselspalten: Mon\_ID
- Namensspalte: Monatsname

2 x Button *Weiter*

- Name: Zeit

Button *Fertigstellen*

A2.6. Ändern Sie den Namen des Attributs *Mon\_ID* in *Monat*.

A2.7. Stellen Sie das Projekt auf den Analysis Services bereit (Projektmappenexplorer: *Umsatz\_Sxxxxx – Kontextmenü [rechte Maustaste] - Bereitstellen*).

**(Ggf. muss in der *Identitätswechselinformation* der Datenquelle *UmsatzSource.ds* die Option *Bestimmter Windows-Benutzer* gewählt werden und dort Benutzername und Kennwort der Windowsanmeldung hinterlegt werden. - Projektmappenexplorer: *UmsatzSource.ds – Kontextmenü [rechte Maustaste] - Öffnen*).**

A2.8. Verarbeiten Sie das Projekt in den Analysis Services (Projektmappenexplorer: *Umsatzdaten – Kontextmenü [rechte Maustaste] - Verarbeiten*).


Hinweis: Die Schritte A2.7 und A2.8 müssen nach jeder Änderung Metadaten des Projektes durchgeführt werden, bevor diese Änderungen im Browser (oder im Front End) sichtbar werden.

A2.9. Zeigen Sie nach Durchführung der Änderungen die Dimensionsdaten im Browser an.

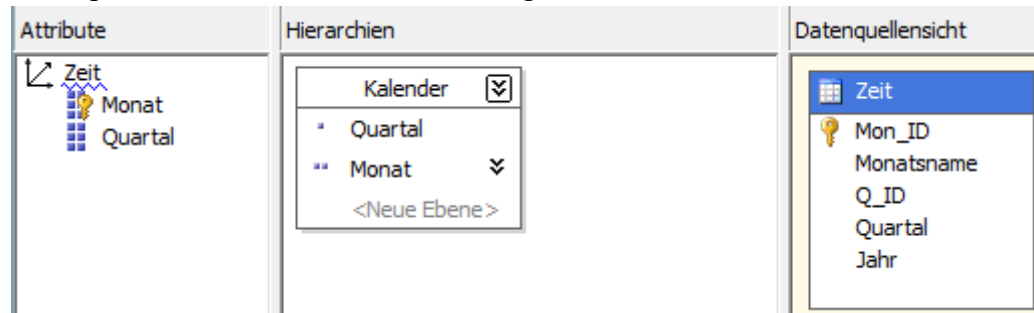
A2.10. Passen Sie die Eigenschaften des Attributes *Monat* der Dimension *Zeit* (im Blatt Dimensionsstruktur) wie folgt an, verarbeiten Sie das Projekt und betrachten Sie anschließend die sich damit ergebende Dimensionsstruktur im Browser:

Attribut	KeyColumns	NameColumn	OrderBy
Monat	Zeit.Mon_ID	Zeit.Monatsname	Key

Hinweis: Die Eigenschaft *KeyColumns* enthält für jedes Attribut in der Dimension den Namen der Spalte mit dem identifizierenden Schlüssel der Elemente. Die Eigenschaft *NameColumn* gibt die Beschriftung an, die für dieses Attribut verwendet wird. Mit der Eigenschaft *OrderBy* kann angegeben werden, nach welcher Spalte die Elemente bei der Anzeige sortiert werden.

Hinweis: Damit nach einer Verarbeitung im Browser die aktuellen Daten angezeigt werden, müssen über den Button *Verbindung wiederherstellen*  die aktuellen Daten vom Server in die Browseransicht geladen werden.

- A2.11. Fügen Sie in der Dimensionsstruktur ein neues Attribut *Quartal* ein. Ziehen Sie dazu das Feld *Quartal* mit der Maus aus der Tabelle *Zeit* (rechts: Datenquellensicht) zu den Attributen der Dimension *Zeit* (links: Attribute). Definieren Sie anschließend per Drag and Drop die Hierarchie *Kalender* wie folgt.



- A2.12. Passen Sie die Eigenschaften des Attributes *Quartal* der Dimension *Zeit* (im Blatt Dimensionsstruktur) wie folgt an, verarbeiten Sie das Projekt und betrachten Sie anschließend die sich damit ergebende Dimensionsstruktur im Browser:

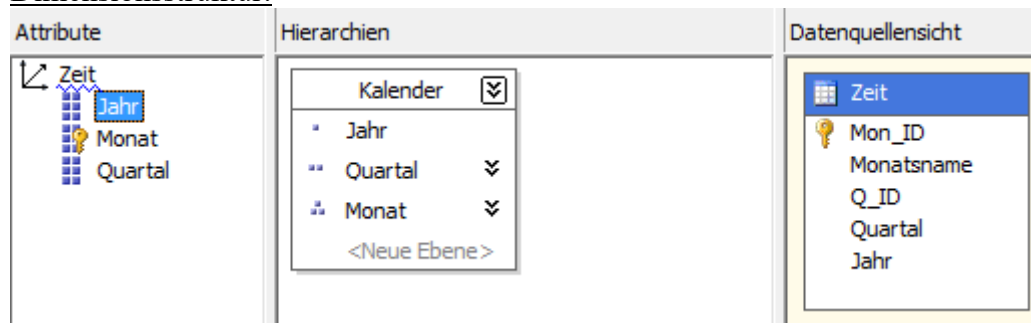
Attribut	KeyColumns	NameColumn	OrderBy
Quartal	Zeit.Q_ID	Zeit.Quartal	Key

- A2.13. Fügen Sie in der Dimensionsstruktur ein neues Attribut *Jahr* ein. Ziehen Sie dazu das Feld *Jahr* mit der Maus aus der Tabelle *Zeit* (rechts: Datenquellensicht) zu den Attributen der Dimension *Zeit* (links: Attribute). Passen Sie anschließend die Attributbeziehungen und die Hierarchie *Kalender* wie folgt an.

Attributbeziehungen:



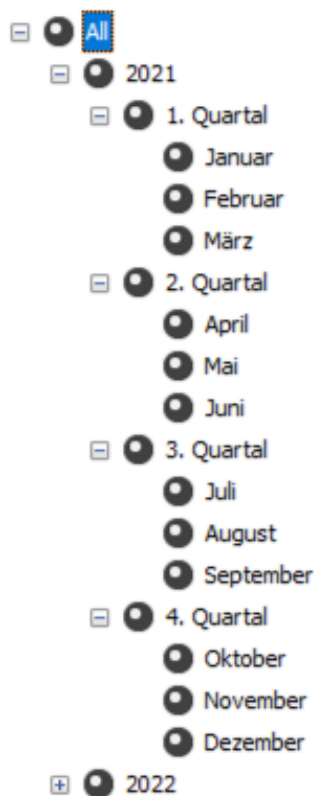
Dimensionsstruktur:



- A2.14. Passen Sie die Eigenschaften des Attributes *Jahr* der Dimension *Zeit* (im Blatt Dimensionsstruktur) wie folgt an, verarbeiten Sie das Projekt und betrachten Sie anschließend die sich damit ergebende Dimensionsstruktur im Browser:

Attribut	KeyColumns	NameColumn	OrderBy
Jahr	Zeit.Jahr	-	Key

Die Dimension *Zeit* mit der Hierarchie *Kalender* sollte nun wie folgt korrekt dargestellt werden.



### ➤ Dimension Geografie

A2.15. Legen Sie eine neue Dimension an (Projektmappenexplorer: *Dimensionen – Kontextmenü* [rechte Maustaste] - *Neue Dimension*).

- Vorhandene Tabelle verwenden: markieren
- Vorlage: (keine)

Button *Weiter*

- Datenquellensicht: UmsatzView
- Haupttabelle: Geografie
- Schlüsselspalten: Land\_ID
- Namensspalte: Bundesland

2 x Button *Weiter*

- Name: Geografie

Button *Fertigstellen*

A2.16. Ändern Sie den Namen des Attributs *Land\_ID* in *Bundesland*, verarbeiten Sie das Projekt und betrachten Sie anschließend die Dimensionsstruktur im Browser.

A2.17. Erweitern Sie die Dimension *Geografie* schrittweise erst um das Attribut *Region* und anschließend um das Attribut *Staat*.

Definieren Sie dazu eine Hierarchie *Unternehmen*. Passen Sie die Attributbeziehungen, und die Eigenschaften *KeyColumns* und *NameColumn* der Attribute entsprechend an.

Ergänzen Sie in der Datenquellensicht *UmsatzView* in der Tabelle *Geografie* eine Spalte *Staat\_Region* mit berechneten Werten.

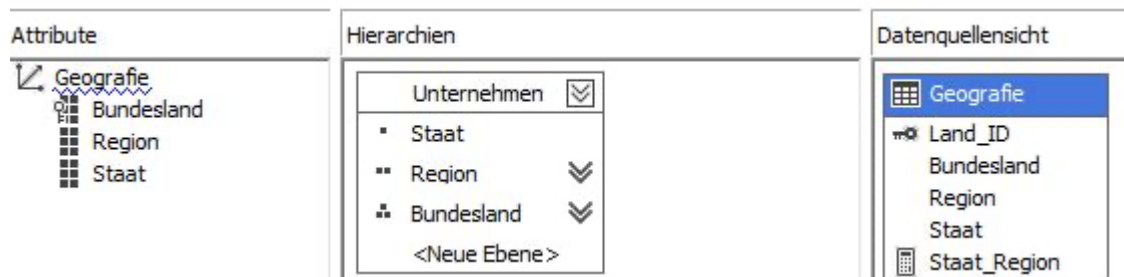
Vorgehen beim Einfügen der Spalte mit berechneten Werten in der Datenquellensicht:

Öffnen Sie die Datenquellensicht *UmsatzView*. Fügen Sie in die Tabelle *Geografie* eine weitere Spalte mit berechneten Werten ein an (Markieren der Tabelle: *Geografie* – Kontextmenü [rechte Maustaste] – *Neue benannte Berechnung*).

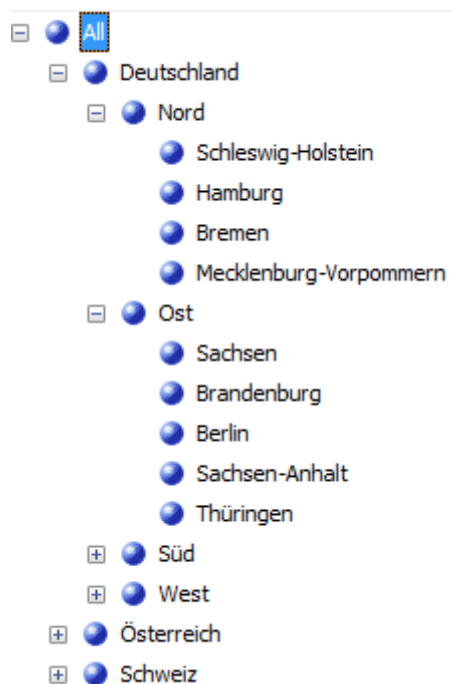
- Spaltenname: Staat\_Region
- Ausdruck: Staat + ' ' + Region

Hinweis: Mittels *Daten durchsuchen* (Markieren der Tabelle: *Geographie* – Kontextmenü [rechte Maustaste] – *Daten durchsuchen*) kann der Inhalt der Tabelle inklusive der in der Datenquellensicht definierten berechneten Felder angezeigt werden.

Verwenden Sie die Spalte *Staat\_Region* in der Definition der Dimension *Geografie*, um eine sinnvolle Datenstruktur für die Dimension und deren Hierarchie zu erzeugen. Beachten Sie insbesondere, dass die Eigenschaften *KeyColumns* und *NameColumn* der Attribute sowie die Attributbeziehungen entsprechend definiert sind.



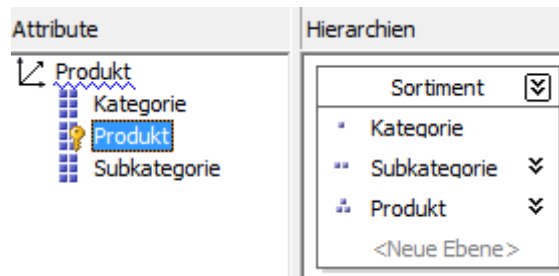
Die Dimension *Geografie* mit der Hierarchie *Unternehmen* sollte nun wie folgt korrekt dargestellt werden.



➤ Dimension Produkt

A2.18. Legen Sie eine neue Dimension *Produkt* an, welche auf den Tabellen *Produkt*, *Produktsubkategorie* und *Produktkategorie* der Datenquellsicht *UmsatzView* basiert.

A2.19. Passen Sie die Attributbeziehungen und die Eigenschaften *KeyColumns* und *NameColumn* der Attribute den Anforderungen an und erstellen Sie per Drag and Drop eine Hierarchie *Sortiment*.



Zeigen Sie anschließend die Dimensionsdaten im Browser an.

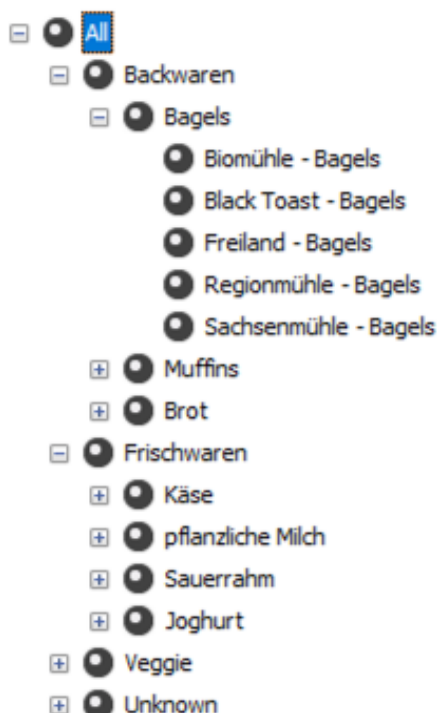
Zeigen Sie dabei insbesondere alle Einträge unter *Backwaren* => *Bagels* an.

Welches Problem ergibt sich beim Anzeigen der *Produktnamen* in der Hierarchie *Sortiment*? Vergleichen Sie dazu den Aufbau der Dimension mit dem Inhalt der Tabelle *Produkt* in der Datenbank *Umsatz* auf dem relationalen Datenbankserver.

A2.20. Passen Sie die Dimension so an, dass eindeutige Produktnamen als Kombination aus Markenname und Produktname angezeigt werden.

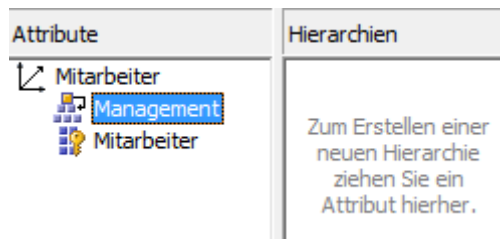
Hinweis: Nutzen Sie die Eigenschaften *KeyColumn* und *NameColumn* und definieren Sie ggf. eine neue berechnete Spalte in der View.

Die Dimension *Produkt* mit der Hierarchie *Sortiment* sollte nun wie folgt korrekt dargestellt werden.



### Dimension Mitarbeiter

- A2.21. Legen Sie eine neue Dimension *Mitarbeiter* an, welche auf der Tabelle *MitarbeiterShop* der Datenquellensicht *UmsatzView* basiert.
- A2.22. Ändern Sie die Bezeichnung des Attributs *Mitarbeiter\_ID* in *Mitarbeiter*.
- A2.23. Ändern Sie die Bezeichnung des Attributs *Manager\_ID* in *Management*.
- A2.24. Aufgrund der Parent-Child-Beziehung in der Tabelle *MitarbeiterShop* wurde vom System direkt in den Attributen bereits eine Hierarchie generiert. (*Management* - siehe folgende Abbildung). Erstellen Sie **keine** weitere Hierarchie für diese Dimension.



Verarbeiten Sie die Dimension *Mitarbeiter* und stellen Sie die Hierarchie *Management* im Browser dar.

### Anlegen des Cubes Umsatz

- A2.25. Legen Sie einen neuen Cube an (Projektmappenexplorer: *Cubes – Kontextmenü [rechte Maustaste] - Neuer Cube*).

- Vorhandene Tabellen verwenden: markieren
- Vorlage: (keine)

Button *Weiter*

- Datenquellensicht: UmsatzView
- Measuregruppentabellen: Umsatzdaten

Button *Weiter*

- Measure: alle markieren

Button *Weiter*

- Dimension: alle markieren

Button *Weiter*

- Cubename: Umsatz

Button *Fertigstellen*

- A2.26. Ändern Sie die Eigenschaft *FormatString* der Measures *Umsatzbetrag* in *#,##0.00 €*

- A2.27. Ändern Sie die Eigenschaft *FormatString* der Measures *Umsatzmenge* in *#,##0*

- A2.28. Verarbeiten Sie den Cube *Umsatz* (Button )

- A2.29. Zeigen Sie den Cubeinhalt des Cubes *Umsatz* im Browser an.

### A3. Visualisierung des Cubes mittels Pivot Tables

#### Verbindung zu den Analysis Services aufbauen

A3.1. Starten Sie Excel und speichern Sie die Arbeitsmappe unter dem Namen *UmsatzanalyseSxxxx.xlsx*.

A3.2. Erzeugen Sie eine neue Verbindung zum Analysis Server Menüpunkt: *Daten => Daten abrufen => Aus Datenbank => Aus Analysis Services*

- Servername: 141.56.2.45
- Anmeldeinformation: *Windows-Zugriffsrechte verwenden* markieren

Button *Weiter*

- Datenbank: Umsatz\_Sxxxxxx  
***Ersetzen Sie dabei Sxxxxxx durch Ihre Bibliotheksnummer***

- Mit einem bestimmten Cube oder einer bestimmten Tabelle verbinden: markieren

- Name: Umsatz

Button *Weiter*

Button *Fertigstellen*

Button *OK*

Hinweis: Die so angelegte Verbindung zum Cube *Umsatz* kann über den Menüpunkt *Daten – Vorhandene Verbindungen* zur Erstellung weiterer PivotTables, z.B. in weiteren Tabellenblättern genutzt werden.

#### Datenauswertung

##### Umsatzbeträge im Bundesland

A3.3. Testen Sie die OLAP-Operationen Drill down/Roll-up, Slice, Pivotierung an der erzeugten PivotTable.

A3.4. Zeigen Sie im Zeilenbereich der PivotTable die Hierarchie *Unternehmen* der Dimension *Geografie*, im Spaltenbereich die Hierarchie *Sortiment* der Produktdimension und als Wert den *Umsatzbetrag* an. Erweitern Sie die Anzeige für die Region *Ost* bis zur Ebene *Bundesland*. Nach welchem Kriterium erfolgt die Sortierung bei der Anzeige der Bundesländer?

A3.5. Ändern Sie die Eigenschaften in der Dimension *Geografie* (im Visual Studie Projekt!) so, dass die Sortierung alphabetisch nach dem Namen der Bundesländer erfolgt. Lassen Sie sich die Änderung in Excel anzeigen.

A3.6. Ändern Sie den Namen des Tabellenblattes mit der PivotTable in *Umsatz\_im\_Bundesland*.

Umsatzbeträge je Produktsubkategorie

- A3.7. Wechseln Sie zum nächsten Tabellenblatt im selben xlsx-Dokument und benennen Sie dieses Tabellenblatt um in *Umsatz\_je\_Produkt*.
- A3.8. Erzeugen Sie darin ebenfalls eine PivotTable, die den Cube *Umsatz* auswertet (siehe Hinweis unter Pkt. 3.2).
- A3.9. Zeigen Sie im Spaltenbereich der PivotTable die Hierarchie *Sortiment* und im Zeilenbereich die Hierarchie *Kalender* an.
- A3.10. Ändern Sie die Anzeige in der PivotTable, damit Trendaussagen bezüglich der Umsatzentwicklung für die einzelnen Produktsubkategorien über die Zeit (auf der Basis von Monatswerten) gewonnen werden können.
- A3.11. Erstellen Sie dazu mittels PivotChart der PivotTable-Tools ein auf der PivotTable basierendes Diagramm vom Typ *Linie*.
- A3.12. Geben Sie die ermittelten Trends qualitativ in der nachfolgenden Tabelle an.

Produktsubkategorie		Trend
Backwaren	Bagels	
	Brot	
	Muffins	
Veggie	Hülsenfrüchte	
	Sojaprodukte	
	Gemüse	
	Obst	
	Nüsse und Ölsamen	



Frischwaren	Joghurt	
	Käse	
	pflanzliche Milch	
	Sauerrahm	

#### Umsatzbeträge je Produktsubkategorie in Sachsen

- A3.13. Wechseln Sie zum nächsten Tabellenblatt im selben xlsx-Dokument und benennen Sie dieses Tabellenblatt um in *Umsatz\_je\_Produkt\_in\_Sachsen*.
- A3.14. Erzeugen Sie darin ebenfalls eine PivotTable, die den Cube *Umsatz* auswertet (siehe Hinweis unter Pkt. 3.2).
- A3.15. Zeigen Sie in der PivotTable, ebenfalls Tendaussagen bezüglich der Umsatzentwicklung für die einzelnen Produktsubkategorien über die Zeit (auf der Basis von Monatswerten), jedoch nur für das Bundesland *Sachsen*, an. Nutzen Sie dafür den Berichtsfiler.
- A3.16. Erstellen Sie dazu ein auf der PivotTable basierendes Diagramm vom Typ *Linie*.

#### Quartal mit der maximalen Umsatzmenge für die von Ihnen betreute Produktkategorie je Staat

- A3.17. Wechseln Sie zum nächsten Tabellenblatt im selben xlsx-Dokument und benennen Sie dieses Tabellenblatt um in *Umsatzmaximum*.
- A3.18. Erzeugen Sie darin ebenfalls eine PivotTable, die den Cube *Umsatz* auswertet (siehe Hinweis unter Pkt. 3.2).
- A3.19. Ermitteln Sie für die einzelnen *Staaten* das *Quartal* mit der maximalen *Umsatzmenge* für die von Ihnen betreute *Produktkategorie*.

Staat	Quartal	Umsatzmenge
Deutschland		
Österreich		
Schweiz		

## A4. Anpassung von Metadaten

### Anzeige der Dimension Mitarbeiter (Parent Child Dimension)

- A4.1. Zeigen Sie die *Umsatzbeträge* bezüglich der Hierarchie *Management* der Dimension *Mitarbeiter* des Cubes *Umsatz* in einer neuen Excel PivotTable an. Analysieren Sie dabei die Werte in den einzelnen Ebenen, der Ebene *All (Gesamtergebnis)* und *Sybille Neubert* sowie die Überschriften für die Ebenen der Dimensionshierarchie.
- A4.2. Blenden Sie in der Dimensionsstruktur der Dimension *Mitarbeiter* die Ebene *All (Gesamtergebnis)* aus (Eigenschaft: *IsAggregatable*).

Attribut	KeyColumns	NameColumn	OrderBy	IsAggregatable
Management	Mitarbeiter. Manager_ID	-	Key	False ( <i>gibt keinen Eintrag „All“</i> )

Attribut	MemberWithDataCaption	MemberWithData
Management	-	NonLeafDataHidden ( <i>keine Anzeige der Werte von Nichtblattknoten</i> )
Mitarbeiter	-	NonLeafDataVisible

- A4.3. Ändern Sie angezeigten Elemente in der Dimension und die Bezeichnungen der Ebenen in der Dimension *Mitarbeiter* mit Hilfe der Eigenschaft *NamingTemplate*:

Eigenschaft *NamingTemplate* des Attributs *Management*:

Ebene	Name
1	(Alle)
2	Manager
3	Kategoriemanager
4	Subkategoriemanager
*	

Zeigen Sie nach Durchführung der Änderungen die Dimensionsdaten im Browser des Cubes *Umsatz* im Visual Studio und in einem neu anzulegenden Tabellenblatt *Management* in der Excel Arbeitsmappe *Umsatzanalyse.xlsx*.


### Ändern der Anzeigereihenfolge in der Dimension Produkt

- A4.4. Ändern Sie die Einstellungen im Analysis Server (Visual Studio Projekt!) so, dass in der PivotTable *Umsatzanalyse.xlsx* die *Kategorien* und die *Subkategorien* der Produkte in alphabetischer Reihenfolge angezeigt werden.
- A4.5. Aktualisieren Sie die jeweils die Anzeige der PivotTables. Hat sich die Sortierung in den PivotTables und in den Diagrammen geändert?
- A4.6. Speichern Sie die Excel Arbeitsmappen.

## A5. Erweiterung des Cubes Umsatz um berechnete Kennzahlen (Berechnungen)

Basierend auf den direkt in der Faktttabelle des Cubes *Umsatz* enthaltenen Kennzahlen *Umsatzbetrag* und *Umsatzmenge* sollen berechnete Kennzahlen (Berechnungen) im Cube erstellt werden.


### Im Umsatzbetrag (Bruttoumsatz) enthaltene Umsatzsteuer

A5.1. Erstellen Sie im Cube *Umsatz* eine neue Berechnung (Registerkarte *Berechnungen*: Button  )

- Name: [Umsatzsteuer]
- Übergeordnete Hierarchie: MEASURES
- Ausdruck:  $[Measures].[Umsatzbetrag] / 1.19 * 0.19$
- Formatzeichenfolge: “#,##0.00 €“
- Sichtbar: TRUE
- Verhalten für nichtleere Elemente:

A5.2. Verarbeiten Sie den Cube *Umsatz* (Button  ) und zeigen Sie den Inhalt der Berechnung *Umsatzsteuer* im Browser an.


### Durchschnittspreis

A5.3. Erstellen Sie im Cube *Umsatz* eine neue Berechnung (Registerkarte *Berechnungen*: Button  )

- Name: [Durchschnittspreis]
- Übergeordnete Hierarchie: MEASURES
- Ausdruck:  $[Measures].[Umsatzbetrag] / [Measures].[Umsatzmenge]$
- Formatzeichenfolge: “#,##0.00 €“
- Sichtbar: TRUE
- Verhalten für nichtleere Elemente: Umsatzmenge

A5.4. Verarbeiten Sie den Cube *Umsatz* (Button  ) und zeigen Sie den Inhalt der Berechnung *Durchschnittspreis* im Browser an.

### Kumulierter Umsatz im aktuellen Jahr

A5.5. Erstellen Sie im Cube *Umsatz* eine neue Berechnung (Registerkarte *Berechnungen*: Button  )

- Name: [kumulierter Umsatz]
- Übergeordnete Hierarchie: MEASURES
- Ausdruck:
 

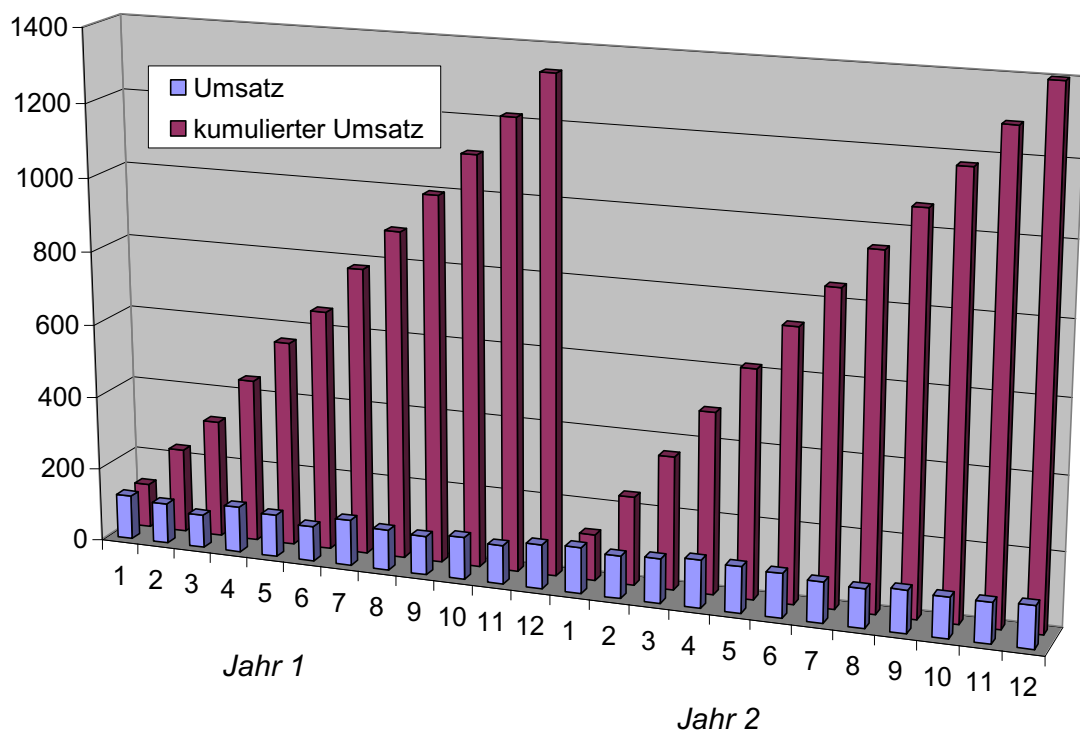
```
Aggregate (
    PeriodsToDate ([Zeit].[Kalender].[Jahr],
    [Zeit].[Kalender].CurrentMember),
    [Measures].[Umsatzbetrag])
```
- Formatzeichenfolge: “#,#.00 €“
- Sichtbar: TRUE
- Verhalten für nichtleere Elemente:

Hinweis: Informieren Sie sich in der Hilfe zu den Funktionen *Aggregate*, *PeriodsToDate* und *CurrentMember*.


A5.6. Verarbeiten Sie den Cube *Umsatz* (Button  ) und zeigen Sie den Inhalt der Berechnung *kumulierter Umsatz* im Browser an.

A5.7. Stellen Sie den Verlauf des kumulierten Umsatzes für die drei Länder über die einzelnen Monate der beiden Jahre auf Basis einer PivotTable in einem neuen Tabellenblatt *kumulierte\_Umsatz* im Exceldokument *Umsatzanalyse.xlsx* dar.

A5.8. Erstellen Sie, basierend auf der PivotTable, ein Diagramm zum Veranschaulichen des Verlaufs des kumulierten Umsatzes für die drei Länder in Anlehnung an die folgende Abbildung.



**Monatsnummer des Jahres**

A5.9. Erstellen Sie im Cube *Umsatz* eine neue Berechnung (Registerkarte *Berechnungen*: Button  )

- Name: [Monnr]
- Übergeordnete Hierarchie: MEASURES
- Ausdruck:

Hinweis: Verwenden Sie im Ausdruck zwei ineinander geschachtelte Case-Anweisungen und die Funktion *CurrentMember.Level* und *CurrentMember.Name*.

- Formatzeichenfolge:
- Sichtbar: TRUE
- Verhalten für nichtleere Elemente:

A5.10. Verarbeiten Sie den Cube *Umsatz* (Button  ) und zeigen Sie den Inhalt der Berechnung *Monnr* im Browser an.

**Jahresprognose des Umsatzes im aktuellen Jahr**

A5.11. Erstellen Sie im Cube *Umsatz* eine neue Berechnung.

- Name: [Umsatzprognose]
- Übergeordnete Hierarchie: MEASURES
- Nehmen Sie die weiteren Einstellungen selbständig vor.

Hinweis: Verwenden Sie zur Bildung des Ausdrucks die bisher berechneten Kennzahlen und wählen Sie eine geeignete Formatierung.

A5.12. Zeigen Sie den Inhalt der Berechnung *Umsatzprognose* im Browser an.

**Vorjahressumme des Umsatzes**

A5.13. Erstellen Sie im Cube *Umsatz* eine neue Berechnung.

- Name: [Vorjahressumme Umsatz]
- Übergeordnete Hierarchie: MEASURES
- Nehmen Sie die weiteren Einstellungen selbständig vor.

Hinweis: Verwenden Sie im Ausdruck eine Case-Anweisung und die Funktion *CurrentMember.Level*, *Parent.Parent.PrevMember*, *Parent.PrevMember* und *PrevMember* und wählen Sie eine geeignet Formatierung

A5.14. Zeigen Sie den Inhalt der Berechnung *Vorjahressumme Umsatz* im Browser an.

**Verhältnis von Umsatzprognose zu Vorjahressumme Umsatz**

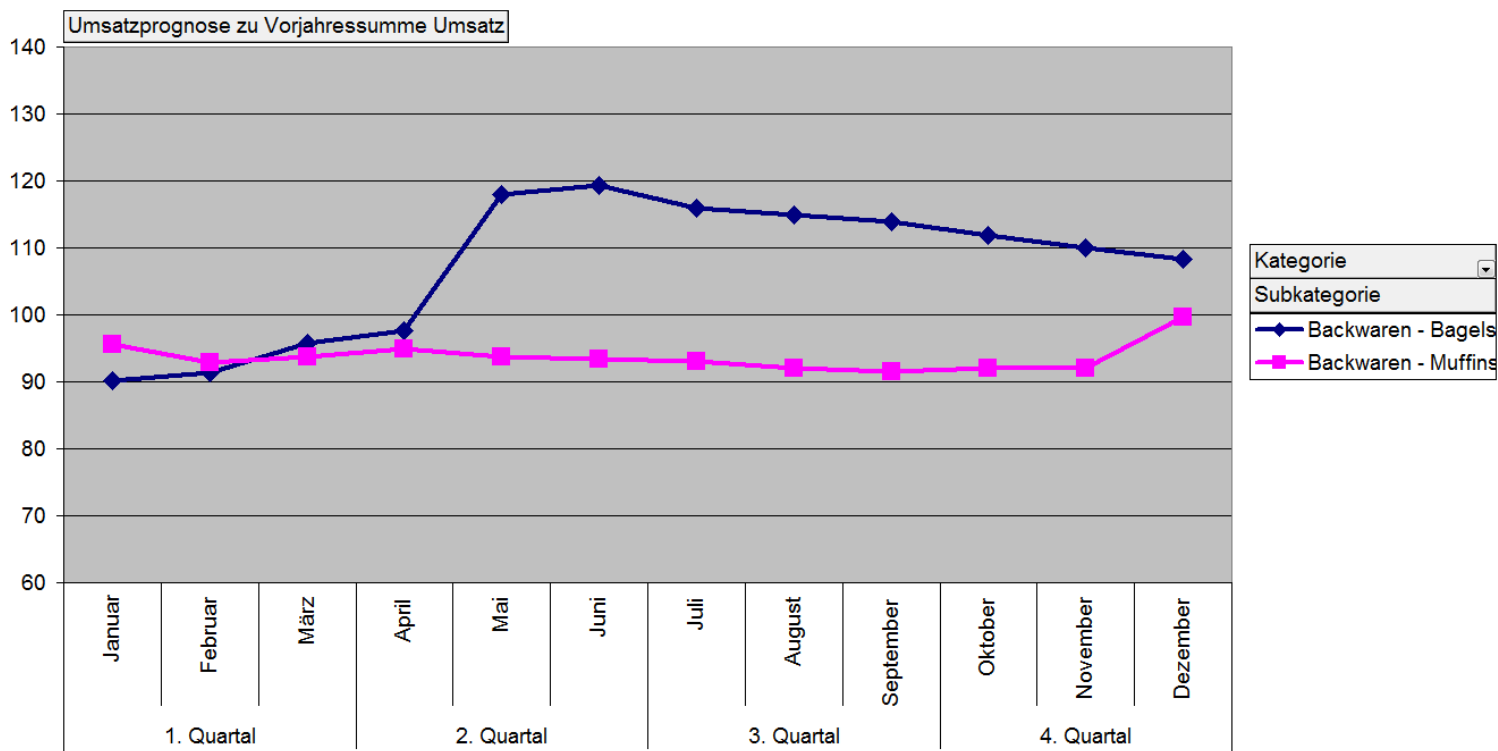
A5.15. Erstellen Sie im Cube *Umsatz* eine neue Berechnung.

- Name: [Umsatzprognose zu Vorjahressumme Umsatz]
- Übergeordnete Hierarchie: MEASURES
- Nehmen Sie die weiteren Einstellungen selbständig vor.

Hinweis: Verwenden Sie im Ausdruck die bisher berechneten Kennzahlen und die Funktion *IIF*, um eine Division durch 0 zu vermeiden und wählen Sie eine geeignete Formatierung

A5.16. Zeigen Sie den Inhalt der Berechnung *Umsatzprognose zu Vorjahressumme Umsatz* im Browser an.

A5.17. Erstellen Sie eine PivotTable mit Diagramm zur Darstellung des Verhältnisses von *Umsatzprognose zu Vorjahressumme Umsatz* für die Subkategorien *Bagels* und *Muffins* im Jahr 2018. Der Inhalt des Diagramms sollte folgender Abbildung entsprechen. Diskutieren Sie die Grafik aus betriebswirtschaftlicher Sicht.



## A6. Erweiterung des Cubes Umsatz um Plandaten

Die Plandaten der Umsätze sollen ebenfalls in das Data Warehouse übernommen werden. Die Werte des Umsatzplanes stehen je Monat (Mon\_ID), Produkt (Produkt\_ID) und Bundesland (Land\_ID) in der als Datei Plandaten.csv im Verzeichnis *I:/prakt/toll/BI* zur Verfügung.

### Erweiterung der DWH-Datenbank Umsatz

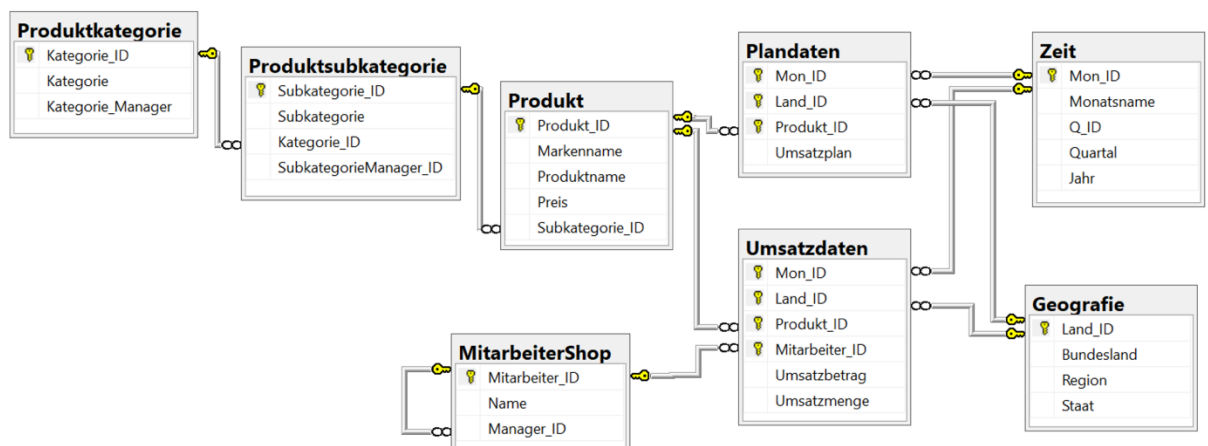
- A6.1. Erstellen Sie eine weitere Fakttable *Plandaten*, die die Daten aus der Datei Plandaten.csv beinhalten soll. Definieren Sie dabei die relevanten Fremdschlüssel auf die entsprechenden Dimensionstabellen.

Die Datei Plandaten.csv finden Sie im OPAL-Kurs *Business Intelligence (I440)*:

Kurzlink: <https://t1p.de/OpalBI>

unter *Praktikum* => *Praktikumsunterlagen/DWH*.

- A6.2. Importieren Sie die Daten aus der Datei Plandaten.csv in die Tabelle *Plandaten* mittels SQL-Befehl BULK INSERT. Geben Sie im BULK INSERT als data file '*D:\BI\Plandaten.csv*' an. Dies ist ein Verzeichnis auf dem Datenbankserver, in dem die zu importierende Datei ebenfalls stehen. Verwenden Sie **nicht** ihr lokales Verzeichnis. Vom lokalen Verzeichnis ist kein Import möglich
- A6.3. Ergänzen Sie das Datenbankdiagramm *DWH-Schema* durch Hinzufügen der Tabelle *Plandaten* wie nachfolgend dargestellt.



### Erweiterung der Datenquellensicht und des Cubes Umsatz (im Visual Studio Projekt)

- A6.4. Ergänzen Sie die Tabelle *Plandaten* in der Datenquellensicht *UmsatView*.
- A6.5. Ergänzen Sie in der Cubestruktur des Cube *Umsatz* die Tabelle *Plandaten* und fügen Sie, basierend auf dieser neuen Fakttable eine neue Measergruppe *Plandaten* im Cube ein.

A6.6. Geben Sie in der nachfolgenden Tabelle die Dimensionen an, nach denen die Kennzahlen in den Measuregruppen *Umsatzdaten* und *Plandaten* ausgewertet werden können?

Hinweis: Betrachten Sie dazu die Dimensionsverwendung der beiden Measuregruppen im Cube *Umsatz* und die Struktur der Fakttabellen, auf denen die Measuregruppen basieren.

Measuregruppe		
Dimensionen		

A6.7. Zeigen Sie die Werte der Kennzahlen *Umsatzbetrag* und *Umsatzplan* für die Kombination verschiedener Dimensionen im Browser an.

#### **Erstellung einer berechneten Kennzahl *Ist zu Plan***

A6.8. Erstellen Sie im Cube *Umsatz* eine neue Berechnung, die *Umsatzbetrag/Umsatzplan* ermitteln soll.

- Name: [Ist zu Plan]
- Übergeordnete Hierarchie: MEASURES
- Nehmen Sie die weiteren Einstellungen selbständig vor.

Hinweis: Verwenden Sie im Ausdruck die bisher berechneten Kennzahlen und die Funktion *IIF*, um eine Division durch 0 zu vermeiden und wählen Sie eine geeignete Formatierung

A6.9. Zeigen Sie den Inhalt der Berechnung *Ist zu Plan* im Browser an.

A6.10. Legen Sie in Ihrem Excel-Dokument *Umsatzanalyse.xlsx* ein neues Tabellenblatt an und benennen Sie das Tabellenblatt um in *Planerfüllung*.

A6.11. Erstellen Sie im Tabellenblatt *Planerfüllung* eine PivotTable, die Erfüllung der Planumsätze durch die Istumsätze in den einzelnen Quartalen für die einzelnen Produktsubkategorien ausweist.

Geben Sie beispielhaft nachfolgend Produktsubkategorien und Quartale an, in denen die vorgegebenen Planwert nicht erfüllt werden.

_____	_____
_____	_____



## A7. Erstellung von Abfragen auf dem relationalen Datenbankschema

Verbinden Sie sich im SQL Server Management Studio mit dem SQL Server 141.56.2.45 und erstellen Sie Abfragen in SQL zur Lösung der folgenden Aufgaben. Dokumentieren Sie die SQL-Befehle in einem Skript.

Hinweis: Verwenden Sie zur Lösung der folgenden Aufgaben an den erforderlichen Stellen unkorrelierte Unterabfragen bzw. die OLAP-Erweiterungen von GROUP BY nach dem SQL-Standard.

- A7.1. Zeigen Sie die Umsatzbeträge je Monat, Mitarbeiter, Produkt und Bundesland für die Bundesländer Sachsen und Thüringen an. (Star-Join)
- A7.2. Zeigen Sie Summen von Umsatzbetrag und Umsatzmenge der Bundesländer Sachsen und Thüringen an. Geben Sie dabei auch die Namen der Bundesländer aus.
- A7.3. Erweitern Sie die Abfrage A 7.2, so dass Umsatzbetrag und Umsatzmenge der Bundesländer Sachsen und Thüringen für die einzelnen Quartale (Anzeige der Q\_ID) ausgegeben werden.
- A7.4. Geben Sie die Umsatzbeträge je Staat und Produktsubkategorie aus.
- A7.5. Erweitern Sie die Abfrage A 7.4, dass auch die Teilsummen ausgegeben werden.
- A7.6. Erweitern Sie die Abfrage A 7.5, dass auch die Jahre mit einbezogen werden.
- A7.7. Geben Sie die Umsatzbeträge je Staat, Region und Bundesland inclusive der Teilsummen entsprechend der sich daraus ergebenden Hierarchie aus.
- A7.8. Geben Sie die Umsatzbeträge je Kategorie, Subkategorie und Produktname der Produkte inclusive der Teilsummen entsprechend der sich daraus ergebenden Hierarchie aus.
- A7.9. Modifizieren Sie die Abfrage A 7.8, so dass an statt des Produktnamen die Umsatzbeträge nun für die Kombination aus Markennamen und Produktnamen ausgegeben werden.




## A8. Erstellung von Abfragen auf den Cube Umsatz mit MDX


Verbinden Sie sich im SQL Server Management Studio mit den Analysis Services 141.56.2.45 und erstellen Sie Abfragen auf ihrem Cube *Umsatz* in MDX zur Lösung der folgenden Aufgaben. Dokumentieren Sie die MDX-Befehle in einem Skript.

- A8.1. Zeigen Sie Umsatzbetrag und Umsatzmenge der Bundesländer *Sachsen* und *Thüringen* an.
- A8.2. Erweitern Sie die Abfrage A8.1, so dass nur Umsatzbetrag und Umsatzmenge des Jahres 2022 angezeigt werden.
- A8.3. Zeigen Sie die Umsatzbeträge für alle Produktkategorien in den Bundesländer *Sachsen* und *Thüringen* für das Jahr 2022 an.
- A8.4. Zeigen Sie die Umsatzbeträge für alle Subkategorien der Produktkategorie *Backwaren* in den Bundesländer *Sachsen* und *Thüringen* für das Jahr 2022 an.

- A8.5. Zeigen Sie die Umsatzbeträge für alle Produktkategorien und alle Staaten für das Jahr 2022 an.
- A8.6. Zeigen Sie die Umsatzbeträge und Umsatzmengen für alle Produktkategorien und alle Staaten für das Jahr 2022 an.
- A8.7. Zeigen Sie den Umsatzbetrag für alle Produktkategorien im Jahr, im Quartal und im Monat an.
- A8.8. Ändern Sie die Abfrage von A8.7 so ab, dass nun Umsatzbetrag und Umsatzmenge für alle Produktkategorien und die Quartale und Monate des Jahres 2022 angezeigt werden.

## A9. Erweiterung des Cubes Umsatz um Key Performance Indicators

- A9.1. Erstellen Sie im Cube *Umsatz* einen neue KPI (Registerkarte *KPIs*: Button  )
- Name: einfacher KPI
  - Zugeordnete Measuregruppe: <Alle>
  - Wertsausdruck:  $[Measures].[Umsatzbetrag]$
  - Zielausdruck:  $[Measures].[Umsatzbetrag] * 1.2$
  - Statusindikator: Verkehrsampel
  - Statusausdruck: -1
  - Trendindikator: Standardpfeil
  - Trendausdruck: -0.6
- A9.2. Verarbeiten Sie den Cube *Umsatz* (Button  ) und betrachten Sie den KPI in der *Browseransicht* der Registerkarte KPI (Button  ).
- A9.3. Erstellen Sie in Ihrem Excel–
- A9.4. Dokument *Umsatzanalyse.xlsx* ein neues Tabellenblatt *KPI\_Produkt* mit einer PivotTable, die *Wert*, *Ziel*, *Status* und *Trend* des eben angelegten KPI für die einzelnen Quartale und Produktkategorien darstellt.
- A9.5. Ändern Sie den Wert der Statusausdrucks auf **0** und den Trendausdruck auf **0.4**. Stellen Sie das Projekt bereit und betrachten Sie die Änderung des KPI in der *Browseransicht* und in der PivotTable in Excel.
- A9.6. Ändern Sie den Wert des Statusausdrucks auf **1** und den Trendausdruck auf **1**. Stellen Sie das Projekt bereit und betrachten Sie die Änderung des KPI in der *Browseransicht* und in der PivotTable in Excel.

A9.7. Erstellen Sie im Cube *Umsatz* einen neue KPI (Registerkarte *KPIs*: Button  )

- Name: Umsatzentwicklung
- Zugeordnete Measuregruppe: <Alle>
- Wertsausdruck: [Measures].[Umsatzbetrag]
- Zielausdruck: [Measures].[Umsatzplan]
- Statusindikator: Verkehrsampel
- Statusausdruck:
 

WENN	Umsatzbetrag > 1.1*Umsatzplan	DANN	1
WENN	Umsatzbetrag > Umsatzplan	DANN	0
SONST			-1

Hinweis: Verwenden Sie zur Bildung des Ausdrucks eine Case-Anweisung.

- Trendindikator: Standardpfeil
- Trendausdruck:
 

WENN	Umsatzbetrag > 1,1*Umsatzbetrag vorheriger Zeitraum	DANN	1
WENN	Umsatzbetrag > 1,02*Umsatzbetrag vorheriger Zeitraum	DANN	0,4
WENN	Umsatzbetrag > 0,98*Umsatzbetrag vorheriger Zeitraum	DANN	0
WENN	Umsatzbetrag > 0,9*Umsatzbetrag vorheriger Zeitraum	DANN	-0,6
SONST		DANN	-1

Hinweise: - Mit „Umsatzbetrag vorheriger Zeitraum“ ist der Umsatz im vorherigen Zeitraum der gleichen Ebene gemeint.

- Verwenden Sie zur Bildung des Ausdrucks eine Case-Anweisung.

A9.8. Legen Sie in Ihrem Excel – Dokument *Umsatzanalyse.xlsx* ein neues Tabellenblatt an und benennen Sie das Tabellenblatt um in *KPI Umsatzentwicklung*.

A9.9. Erstellen Sie im Tabellenblatt *KPI\_Umsatzentwicklung* eine PivotTable, die den KPI *Umsatzentwicklung* in den einzelnen Quartalen für die einzelnen Produktsubkategorien ausweist.

Geben Sie nachfolgend die Produktsubkategorien an, bei denen es eine negative Entwicklung gibt.


A9.10. Analysieren Sie in einem weiteren Tabellenblatt, ob die KPI-Werte regionale Unterschiede aufweisen.

---

---

---

---

---

## Aufgabenkomplex B: Data Mining mit Python

### Hinweise:

Die für das Praktikum erforderlichen Softwarekomponenten sind im Labor Z136 B installiert und können für die Lösung der Aufgabenstellungen verwendet werden.

#### Arbeiten von zu Hause - Variante 1: lokale Installation:

Für die Arbeit mit einer lokalen Installation befolgen Sie bitte die im Wiki des OPAL-Kurses beschriebene Verfahrensweise für das Arbeiten von zu Hause.

Diese finden Sie im OPAL-Kurs Business Intelligence (I440):

Kurzlink: <https://t1p.de/OpalBI>

unter: Wiki => Arbeiten von zu Hause => Möglichkeit – Nutzung eines (lokal installierten) Visual Studio Code

#### Arbeiten von zu Hause - Variante 2: Terminalserver der Fakultät Informatik/Mathematik:

Alternativ können Sie das auf dem Terminalserver der Fakultät Informatik/Mathematik installierte Visual Studio Code nutzen. Für diese Variante befolgen Sie bitte die im Wiki des OPAL-Kurses beschriebene Verfahrensweise über den Terminalserver der HTW Dresden.

Diese finden Sie im OPAL-Kurs Business Intelligence (I440):

Kurzlink: <https://t1p.de/OpalBI>

unter: Wiki => Arbeiten von zu Hause => Möglichkeit – Login am Terminalserver

## B1. Einführendes Beispiel zur Erstellung von Entscheidungsbäumen

B1.1. Kopieren Sie die Datei *PythonBI.zip* aus

*Praktikum => Praktikumsunterlagen=> Python*

des OPAL-Kurses *Business Intelligence (I440)* (Kurzlink: <https://t1p.de/OpalBI>) in ein lokales Verzeichnis.

B1.2. Entpacken Sie die Datei *PythonBI.zip* in Ihrem lokalen Verzeichnis.

B1.3. Starten Sie *Visual Studio Code* und öffnen Sie darin das Verzeichnis *PythonBI*.

B1.4. Wechseln Sie im Visual Studio Code zur Datei des Jupyter Notebooks *EBaumDemoBonitaet.ipynb*.

B1.5. Führen Sie schrittweise die im Notebook darin enthaltenen Python-Befehle aus und analysieren Sie die jeweiligen Ausgaben.

B1.6. Geben Sie für die Methode *Tree.generateTree(...)* die Bedeutung der folgenden Parameter an:

*minElements*: \_\_\_\_\_

*maxLevel*: \_\_\_\_\_

B1.7. Geben Sie die Güte des Modells für Baumerstellung mit den Parameter *minElements=1* und *maxLevel=99* durch Angabe der prozentualen Anteile *richtig positiven*, *falsch positiven*, *falsch negativen* und *richtig negativen* Klassifikationen in nachfolgender Tabelle an.

		Realität (reale Ergebnisse in der Testdatenmenge)	
		<i>positiv</i>	<i>negativ</i>
Ergebnis der Modellanwendung auf die Testdatenmenge	<i>positiv</i>		
	<i>negativ</i>		

B1.8. Geben Sie die Fehlerrate des Modells aus B1.7 an.

Fehlerrate: \_\_\_\_\_ %

B1.9. Geben Sie die Güte des Modells für Baumerstellung mit den Parameter *minElements=5* und *maxLevel=99* durch Angabe der prozentualen Anteile *richtig positiven*, *falsch positiven*, *falsch negativen* und *richtig negativen* Klassifikationen in nachfolgender Tabelle an.

		Realität (reale Ergebnisse in der Testdatenmenge)	
		<i>positiv</i>	<i>negativ</i>
Ergebnis der Modellanwendung auf die Testdatenmenge	<i>positiv</i>		
	<i>negativ</i>		

B1.10. Geben Sie die Fehlerrate des Modells aus B1.9 an.

Fehlerrate: \_\_\_\_\_ %

## B2. Selbständig zu modellierendes Beispiel zur Erstellung von Entscheidungsbäumen

- B2.1. Schauen Sie sich den Inhalt der Datei *PythonBI/data/Konsumkredit.csv* mit einem Editor ihrer Wahl an. Nehmen Sie dabei **keine** Änderungen am Inhalt der Datei vor!

Die Daten besitzen folgende Struktur:

### Zielvariable:

#### kreditwuerdig:

J = „ja“

N = „nein“

### Predictorvariablen:

#### Geschlecht:

W = „weiblich“

M = „männlich“

#### Familienstand:

L = „ledig“

V = „verheiratet“

W = „verwitwet“

#### Kinder:

0 = „keine Kinder“

1 = „ein Kind“

2 = „zwei Kinder“

3 = „drei Kinder“

#### Beruf:

ungelernt

angelernt

unbekannt

Facharbeiter

Buerokraft

Manager

Lehrer

#### Wohneigentum:

J = „ja“

N = „nein“

#### Ersparnisse:

J = „ja“

N = „nein“

#### Einkommen: numerisch

- B2.2. Erstellen Sie in Anlehnung an das einführende Beispiel aus Aufgabe B1 ein Jupyter Notebook, das einen Entscheidungsbaum in den folgend beschriebenen Teilschritten.

#### Hinweis:

Zur Lösung der Aufgabenstellung können Sie das Jupyter Notebook *EBaumDemoBonitaet.ipynb* aus Aufgabe B1 im gleichen Verzeichnis (*.../PythonBI/*) kopieren und dort umbenennen. Öffnen Sie anschließend im Visual Studio Code das Verzeichnis *PythonBI*, um auch die neue Datei zu bearbeiten.

- B2.3. Führen Sie folgende Schritte zur Datenanalyse und Vorverarbeitung durch:

- Darstellung der Wertverteilung der Predictorvariablen *Beruf* und *Familienstand* mit der Methode *Data.schowPieCharts(...)*



- Einfügen einer weiteren Predictorvariable *Einkommensklasse*, die abhängig vom *Einkommen* folgende Klassen besitzt:

Einkommen:

- bis 4000
- 4001 bis 6000
- 6001 bis 8000
- größer 8000

- Darstellung der Wertverteilung der Predictorvariable *Einkommensklasse* mit der Methode `Data.showPieCharts(...)`
- Aufteilung der Daten in LernDatenmenge: 70 % und Testdatenmenge 30 %

B2.4. Erstellung Sie einen Entscheidungsbaum unter Verwendung der default-Werte von *minElements* und *maxLevel*. Geben Sie die Güte des erzeugten Modells durch Angabe der prozentualen Anteile an *richtig positiven*, *falsch positiven*, *falsch negativen* und *richtig negativen* Klassifikationen in nachfolgender Tabelle an.

		Realität (reale Ergebnisse in der Testdatenmenge)	
		<i>positiv</i>	<i>negativ</i>
Ergebnis der Modellanwendung auf die Testdatenmenge	<i>positiv</i>		
	<i>negativ</i>		

B2.5. Geben Sie die Fehlerrate des Modells aus B2.4. an.

Fehlerrate: \_\_\_\_\_ %

B2.6. Erstellung Sie einen Entscheidungsbaum unter Verwendung von *minElements=10*. Geben Sie die Güte des erzeugten Modells durch Angabe der prozentualen Anteile an *richtig positiven*, *falsch positiven*, *falsch negativen* und *richtig negativen* Klassifikationen in nachfolgender Tabelle an.

		Realität (reale Ergebnisse in der Testdatenmenge)	
		<i>positiv</i>	<i>negativ</i>
Ergebnis der Modellanwendung auf die Testdatenmenge	<i>positiv</i>		
	<i>negativ</i>		

B2.7. Geben Sie die Fehlerrate des Modells aus B2.6 an.

Fehlerrate: \_\_\_\_\_ %

B2.8. Erstellung Sie einen Entscheidungsbaum unter Verwendung von  $maxLevel=3$ . Geben Sie die Güte des erzeugten Modells durch Angabe der prozentualen Anteile an *richtig positiven*, *falsch positiven*, *falsch negativen* und *richtig negativen* Klassifikationen in nachfolgender Tabelle an.

		Realität (reale Ergebnisse in der Testdatenmenge)	
		<i>positiv</i>	<i>negativ</i>
Ergebnis der Modellanwendung auf die Testdatenmenge	<i>positiv</i>		
	<i>negativ</i>		

B2.9. Geben Sie die Fehlerrate des Modells aus B2.8 an.

Fehlerrate: \_\_\_\_\_ %

**B3. Einführendes Beispiel zur Klassifikation mittels Neuronalem Netz - XOR**

- B3.1. Starten Sie *Visual Studio Code* und öffnen Sie darin das Verzeichnis *PythonBI*.
- B3.2. Wechseln Sie im Visual Studio Code zur Datei des Jupyter Notebooks *NnDemoXOR.ipynb*.
- B3.3. Führen Sie schrittweise die im Notebook darin enthaltenen Python-Befehle aus und analysieren Sie die jeweiligen Ausgaben.
- B3.4. Geben Sie Größe der durch den Split erzeugten Datenmengen an:

Anzahl Datensätze in den Trainingsdaten: \_\_\_\_\_

Anzahl Datensätze in den Validierungsdaten: \_\_\_\_\_

Anzahl Datensätze in den Testdaten: \_\_\_\_\_

- B3.5. Geben Sie für den Konstruktor der Klasse *Dense* die Bedeutung der folgenden Parameter an:

neurons: \_\_\_\_\_

activation: \_\_\_\_\_

- B3.6. Geben Sie für den Konstruktor der Klassen *SGD* und *SGDMomentum* die Bedeutung des folgenden Parameters an:

lr: \_\_\_\_\_

- B3.7. Geben Sie für die Methode *Trainer.fit(...)* die Bedeutung der folgenden Parameter an:

epochs: \_\_\_\_\_

batch\_size: \_\_\_\_\_

eval\_every: \_\_\_\_\_

- B3.8. Geben Sie die Güte des Netzes mit einer verdeckten Schicht durch Angabe der prozentualen Anteile an richtig positiven, falsch positiven, falsch negativen und richtig negativen Klassifikationen in nachfolgender Tabelle an.

		Realität (reale Ergebnisse in der Testdatenmenge)	
		<i>positiv</i>	<i>negativ</i>
Ergebnis der Modellanwendung auf die Testdatenmenge	<i>positiv</i>		
	<i>negativ</i>		

B3.9. Geben Sie die Fehlerrate des Modells aus B3.8 an.

Fehlerrate: \_\_\_\_\_ %

B3.10. Geben Sie die Güte des Netzes mit zwei verdeckten Schichten durch Angabe der prozentualen Anteile an richtig positiven, falsch positiven, falsch negativen und richtig negativen Klassifikationen in nachfolgender Tabelle an.

		Realität (reale Ergebnisse in der Testdatenmenge)	
		<i>positiv</i>	<i>negativ</i>
Ergebnis der Modellanwendung auf die Testdatenmenge	<i>positiv</i>		
	<i>negativ</i>		

B3.11. Geben Sie die Fehlerrate des Modells aus B3.10 an.

Fehlerrate: \_\_\_\_\_ %

## B4. Selbständig zu modellierendes Beispiel zur Klassifikation mittels Neuronalem Netz

Auf Basis der mit der Aufgabe B2 vorliegenden Daten aus der Datei *Konsumkredit.csv* soll eine Klassifikation mittels Neuronaler Netze erstellt und deren Modellgüte mit der Güte der in Aufgabe B2 erstellten Modelle verglichen werden.

- B4.1. Erstellen Sie im Visual Studio Code im Verzeichnis *PythonBI* ein neues Jupyter Notebook *NnKonsumkredit.ipynb*.
- B4.2. Erstellen Sie im Jupyter Notebook *NnKonsumkredit.ipynb* unter Verwendung der Daten aus der Datei *Konsumkredit.csv* ein Neuronales Netz in den Teilschritten:
- Datenanalyse
  - Vorverarbeitung: Normierung der Daten (siehe nachfolgende Anmerkungen!)
  - Vorverarbeitung: Aufteilen in Lern-, Validierungs- und Testdatenmenge (65 %, 15 % und 20 %)
  - Erstellung des Neuronalen Netzes unter Verwendung der Spalten mit den normierten Werten (Vorschlag: 20 Neuronen in einem Hiddenlayer)
  - Bewertung der Modellgüte

### Anmerkungen zur Normierung:

Für die Variablen sind im `Pandas.DataFrame` weitere Spalte zu erzeugen, in denen die Variablen mit normierten Werten zwischen 0 und 1 abgebildet werden. Die Bezeichnung der neuen Spalten mit den normierten Werten sollen mit *N\_* beginnen.

In Abhängigkeit der jeweiligen Datenkategorie sind Transformationen durchzuführen:

- **binäre Werte**

z.B. *kreditwuertig* mit den Werten J und N

=> eine neue Spalte *N\_kreditwuertig* mit den Werten 0 und 1

Beispielskript in Python:

```
column = 'kreditwuertig'
df[f'N_{column}'] = np.where(df[column] == 'J', 1, 0)
```

- **numerische Werte**

z.B. *Kinder* mit den Werten 0, 1, 2, 3

=> eine neue Spalte *N\_Kinder* mit den Werten im Bereich zwischen 0 und 1

Beispielskript in Python:

```
column = 'Kinder'
df[f'N_{column}'] = (df[column] - df[column].min()) \
    / (df[column].max() - df[column].min())
```

- **nominale Werte**

z.B. *Familienstand* mit den Werten L, V, W

=> je Wert eine neue Spalte in der 1 steht, wenn der Wert zutrifft

Beispielskript in R:

```
column = 'Familienstand'
for value in df[column].unique():
    df[f'N_{column}{value}'] = np.where(df[column] == value, 1, 0)
```

Nach Abschluss der Normierung soll das Pandas.DataFrame folgende Struktur besitzen:

- Geschlecht
- Familienstand
- Kinder
- Beruf
- Wohneigentum
- Einkommen
- Ersparnisse
- Kreditwuerdig
- N\_kreditwuerdig
- N\_Geschlecht
- N\_FamilienstandV
- N\_FamilienstandW
- N\_Familienstand
- N\_Kinder
- N\_Berufungelernt
- N\_Berufangelernt
- N\_Berufunbekannt
- N\_BerufFacharbeiter
- N\_BerufBuerokraft
- N\_BerufManager
- N\_BerufLehrer
- N\_Wohneigentum
- N\_Einkommen
- N\_Ersparnisse

- B4.3. Geben Sie die Güte des erzeugten Modells durch Angabe der prozentualen Anteile an *richtig positiven*, *falsch positiven*, *falsch negativen* und *richtig negativen* Klassifikationen in nachfolgender Tabelle an.

		Realität (reale Ergebnisse in der Testdatenmenge)	
		<i>positiv</i>	<i>negativ</i>
Ergebnis der Modellanwendung auf die Testdatenmenge	<i>positiv</i>		
	<i>negativ</i>		

- B4.4. Geben Sie die Fehlerrate des Modells aus B4.2 an.

Fehlerrate: \_\_\_\_\_ %

- B4.5. Versuchen Sie, die Fehlerrate des Neuronalen Netzes so weit wie möglich zu verringern. Variieren Sie dafür die Anzahl an Hiddenschichten und die darin enthaltene Anzahl an Neuronen, die Hyperparameter des Modells und den verwendeten Optimierer. Dokumentieren Sie Ihre Ergebnisse in folgender Tabelle.

Anzahl Hiddenlayer / Anzahl Neuronen je Layer	epochs	batch_ size	Lern- rate	Opti- mierer	Fehlerrate [%]

Anzahl Hiddenlayer / Anzahl Neuronen je Layer	epochs	batch_ size	Lern- rate	Opti- mierer	Fehlerrate [%]

B4.6. Vergleichen Sie die Fehlerraten der in B2 und B4 erstellten Modelle. Welches Modell besitzt die geringste Fehlerrate?

Modell	Fehlerrate [%]
bester Entscheidungsbaum aus B2	
bestes Neuronales Netz aus B4	