

BA1-Praktikum WS2022/23
Add-in für Abwesenheitsanalyse
in Dynamics 365 Business Central
Meilenstein 3

Vorname	Nachname	Matrikelnummer
Ismail	Ouabi	

Den Meilenstein 1 haben Sie allein oder in Zweierteams vorbereitet. Im restlichen Praktikum bilden wir Teams mit **minimal 3 und maximal 4 Studierenden**. Das bedeutet, dass Sie sich im neuen Team aus den unterschiedlichen Ansätzen, die die Teammitglieder einbringen, auf ein einziges gemeinsames Konzept für die Beschreibung der fachlichen Domäne, dem Design der Fakten- und Dimensionstabellen sowie den Befüllungsprozessen einigen müssen und hier in diesem Dokument darstellen, bevor Sie sich mit den Aufgaben zum Meilenstein 2 auseinandersetzen. Sollten bei der Besprechung des Meilenstein 1 Verbesserungsmöglichkeiten deutlich geworden sein, empfehlen wir dringend dies auch umzusetzen. Im Meilenstein 2 wird die Ausarbeitung zum MS1 nicht erneut bewertet. Allerdings ist eine gute Bearbeitung des MS1 Voraussetzung, um auch die folgenden Aufgaben des MS2 und MS3 gut lösen zu können.

Abgabe des Meilensteins 2 **zusammen mit der finalen Fassung der Dokumentation zum Meilenstein 1** per Upload in das entsprechende ILIAS-Übungselement.

Meilenstein 1 (konsolidiert, und ggf nach Fehlerbereinigung)

Kontext: Anhand einer Fallstudie werden die Entwicklungsgrundlagen des ERP-Systems „Microsoft Dynamics 365 Business Central“ praktisch auf eine Aufgabenstellung „Business Intelligence“-Kontext angewendet. Ziel des Praktikums ist der Kompetenzaufbau zur Erfassung einer exemplarischen fachlichen Aufgabe und deren Umsetzung durch Erweiterung einer exemplarischen Standard-ERP-Software.

Hinweis: Bearbeiten Sie die Aufgabe selbständig einzeln oder im Zweierteam. Zusammenarbeit mit anderen Teams ist nicht erlaubt. Erstellen Sie eine Dokumentation. Textstellen, die Sie aus anderen Quellen verwendet haben, müssen gemäß wissenschaftlicher Standards gekennzeichnet werden. Drücken Sie Ihre Aussagen mit eigenen Worten aus, um zu vermeiden, dass bei einer Plagiatskontrolle Auffälligkeiten ausgewiesen werden.

SZENARIO UND AUSGANGSSITUATION

Die Geschäftsleitung des Kleinunternehmens Cronus AG hat den Eindruck, dass die Arbeitsmoral der Mitarbeiter in letzter Zeit gesunken sei. Sie ist sich nicht sicher, ob dies angesichts der angespannten wirtschaftlichen Lage nur ihr subjektiver Eindruck sei, oder ob sich dies wirklich objektivieren lasse. In einem Seminar für Führungskräfte hat sie gehört, dass sich eine verschlechternde Motivation der Mitarbeiter oft in einer negativen Entwicklung des Krankenstandes und insbesondere der Entwicklung des Krankenstandes an Montagen niederschlägt. Ob diese Hypothese wirklich belastbar ist, sei dahingestellt. Aber immerhin entscheidet sich das Management dazu, ein flexibles Reporting-System

für die Analyse des Krankenstandes aufbauen zu wollen. Aus diesen Überlegungen heraus übergibt die Geschäftsleitung an Sie als IT-Berater/in den Auftrag zu prüfen, wie regelmäßige und flexible Analysen der Abwesenheiten durchgeführt werden können. Ihre Aufgabe im Meilenstein 1 besteht nun darin, ein Konzept zu entwickeln, in dem deutlich wird, dass Sie die fachliche Aufgabe verstehen, und das einen sinnvollen Lösungsansatz für eine „Analyse-freundliche“ Datenstruktur beinhaltet.

Fachliches Szenario

Die Cronus AG hat derzeit acht Mitarbeiter, die jeweils einer Abteilung wie Produktion, Verkauf und Verwaltung zugeordnet sind. Eine Ausnahme bildet das Top-Management, was keiner Abteilung zugerechnet wird. Das Unternehmen verwendet Business Central von Microsoft als ERP System. Die Cronus AG hat sich bewusst für eines der marktführenden ERP-Systeme für kleine und mittelständische

Unternehmen entschieden. Es basiert auf gut etablierter Technologie, nämlich „.NET“ und einer MS SQL-Datenbank.

Abwesenheiten von Mitarbeitern werden durch eine Abwesenheitsmeldung an die Verwaltung mitgeteilt. Die wichtigsten Meldungen dieser Art sind Krankmeldungen und genehmigte Urlaubsanträge. Eine Abwesenheit wird als Datensatz in Business Central erfasst, wenn die Abwesenheit beendet ist.

No.†	Last Name	First Name	Department Code
BK	Kohler	Bernhard	
DH	Hembrock	Detlef	VERKAUF
GP	Peters	Gebhard	PROD
HS	Simon	Hanno	VERW
HH	Hoffmann	Isma	PROD

Abbildung 1: Snapshot aus der Mitarbeitertabelle

Dies ist ein etwas unübliches Verfahren, was aber hilft, die Anzahl der Eingaben im System zu reduzieren. Damit ist klar, dass die Daten von Business Central gegenwärtig nicht zum Monitoring der aktuellen Abwesenheiten geeignet sind. Dies ist aber auch derzeit aufgrund des kleinen Mitarbeiterstamms auch nicht nötig. Aus den erfassten Abwesenheitsmeldungen gehen unter anderem der betroffene Mitarbeiter, der Beginn und das Ende, der Abwesenheitsgrund und die Zahl der davon betroffenen Arbeitstage hervor.

Employee No.	From Date	To Date	Cause of Absence Code	Description	Quantity	Unit Code
LG	01.06.2020	05.06.2020	URLAUB	Urlaub	5	TAG
MH	01.06.2020	05.06.2020	URLAUB	Urlaub	5	TAG
GP	05.06.2020		KRANK	Krank	8	STU

Abbildung 2: Snapshot aus der Abwesenheitsregistrierung

Abwesenheiten im Unternehmen sind grundsätzlich ein- oder mehrtägig. Untertägige Abwesenheiten, zB ein halber Tag, werden nicht über eine Abwesenheitsmeldung erfasst, sondern im Rahmen eines Arbeitszeitkontos der Zweitwirtschaft behandelt und sind hier nicht relevant. Beim Eintragen der Abwesenheitsnachweises in das ERP-System wird die Zahl der betroffenen Arbeitstage händisch anhand eines Basiskalenders berechnet, aus dem die arbeitsfreien Kalendertage hervorgehen.

Anforderung der Unternehmensleitung

Sie sollen der Unternehmensberatung ein Konzept vorschlagen, wie eine flexible Analyseplattform auf Basis dieser Daten innerhalb von Business Central geschaffen werden kann. Es ist also ausdrücklich nicht das Ziel, dass Sie eine Analyse der Abwesenheiten vornehmen, sondern das Ziel ist, dass Sie eine

Analyseplattform entwickeln, in der die Unternehmensleitung entsprechende Analysen regelmäßig und selbständig vornehmen kann. Es ist auch nicht das Ziel, ein Monitoring-System zu schaffen, also wie viele Mitarbeiter sind aktuell abwesend, sondern es sollen vergangene Zeiträume untersucht und dadurch beispielsweise Trends ausgewertet werden können.

Lösungsgrobentwurf

Analysieren Sie das Szenario und entwickeln daraus ein Konzept:

1. Ein Domänenmodells, welches gut verständlich ist und sich auf die wesentlichen Aspekte Ihres Vorhabens konzentriert.
2. Welche Beobachtungsgrößen (Messgrößen) zu Ereignissen und Prozessen sind in der Domäne aus den Daten ableitbar, die für die Analyse relevant erscheinen?
3. Entwickeln Sie Vorschläge für passende Kennzahlen und Aufrissgrößen für die Analyse!
4. Untersuchen Sie mindestens zwei Faktentabellen unterschiedlichen Typs, die als Basis der Berichtsaufgaben dienen können (oder ggf auch mehrere, wenn Sie das als sinnvoll ansehen).
5. Legen Sie in einem Excel-Spreadsheet exemplarische Abwesenheitsdaten an und zeigen exemplarisch, wie aus diesen Daten die beiden Faktentabellen gefüllt werden können.
6. Diskutieren Sie die Vor- und Nachteile der beiden Faktentabellen im Hinblick auf die konkrete Aufgabenstellung.
7. Entwerfen Sie sinnvolle Dimensionsstrukturen für die Faktentabellen!
8. Entwerfen Sie die Befüllung der Faktentabellen per Pseudocode!

Hinweis:

- Als Hilfestellung steht Ihnen ein Spreadsheet-File zur Verfügung, indem anhand von Bikesharing-Daten die Umsetzung und das Füllen von Faktentabellen demonstriert wird. Als Demo für exemplarische Berichte wird hier die Excel-Pivot-Funktionalität verwendet. Per Drag&Drop können damit bereits interaktiv diverse Berichte generiert werden (s. Anhang).

1 MEILENSTEIN 1

1.1 ZU UNTERSUCHENDE FACHLICHE DOMÄNE

1.1.1 Domänenmodell

1.1.2 Beobachtungsgrößen der Domäne

1.1.3 Kennzahlen und Aufrissgrößen für die Analyse!

1.2 DESIGN FAKTENTABELLEN

1.2.1 Proof-of-Concept anhand eines Spreadsheets

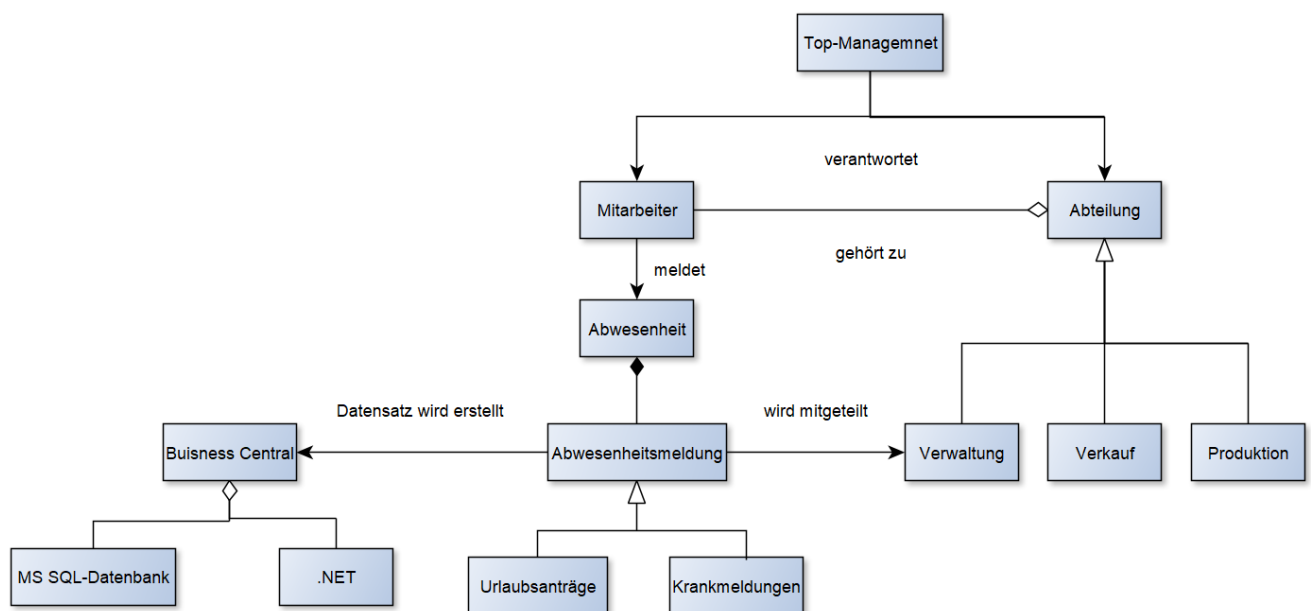
1.2.2 Analyse Faktentabellen für Anwendung

1.3 DIMENSIONSSTRUKTUREN FÜR GEWÄHLTE FAKTENTABELLEN

1.4 DESIGN ZUR BEFÜLLUNG DER GEWÄHLTEN FAKTENTABELLEN

1.1 ZU UNTERSUCHENDE FACHLICHE DOMÄNE

1.1.1 Domänenmodell



1.1.2 Beobachtungsgrößen

Messgrößen für Ereignisse:

- Der Mitarbeiter reicht eine Krankenmeldung an Verwaltung ein
- Der Mitarbeiter stellt einen Urlaubsantrag an Verwaltung.

Messgrößen für Prozesse:

- Der Beginn und das Ende der der Abwesenheitsgrund, Abwesenheitsdauer
- und Anzahl der davon betroffenen Arbeitstage.

1.1.3 Kennzahlen und Aufrissgrößen für die Analyse

Absolut:

- Anzahl der Abwesenheitstagen
 - Aufrissgröße: in Monaten
- Anzahl der Sollarbeitstage
 - Aufrissgröße: an Wochentagen

Relativ:

- Krankenstand (Krankenquote)
 - Aufrissgröße: an Wochentagen
- Krankenstand
 - Aufrissgröße: in Monaten

1.2 DESIGN FAKTENTABELLE

1.2.1 Proof-of-Concept anhand eines Spreadsheets

- **PSFT: ermittelt die folgenden Fakten:**
 - Die Anzahl der die Sollarbeitstage
 - Anzahl der Abwesenheitstage
- **ASFT: ermittelt die folgenden Fakten:**
 - Abwesenheiten an Montagen
 - Abwesenheiten an Nicht Montagen
 - Dauer der Abwesenheit in Tagen

1.2.2 Bewertung der Faktentabellen für Anwendung Accumalting Snaphot Fact Table:

ASFT

Sinnvoll, um die Zeitdauer des Abwesenheitsprozess zu messen und Unterscheidungen nach Kategorien zu erstellen. Vor allem die zwei Zeitstempel von Eingang und Ausgang der Meldung sowie die durchgehend gleiche Datenstruktur vereinfacht die Implementation dieser Faktentabelle

PSFT

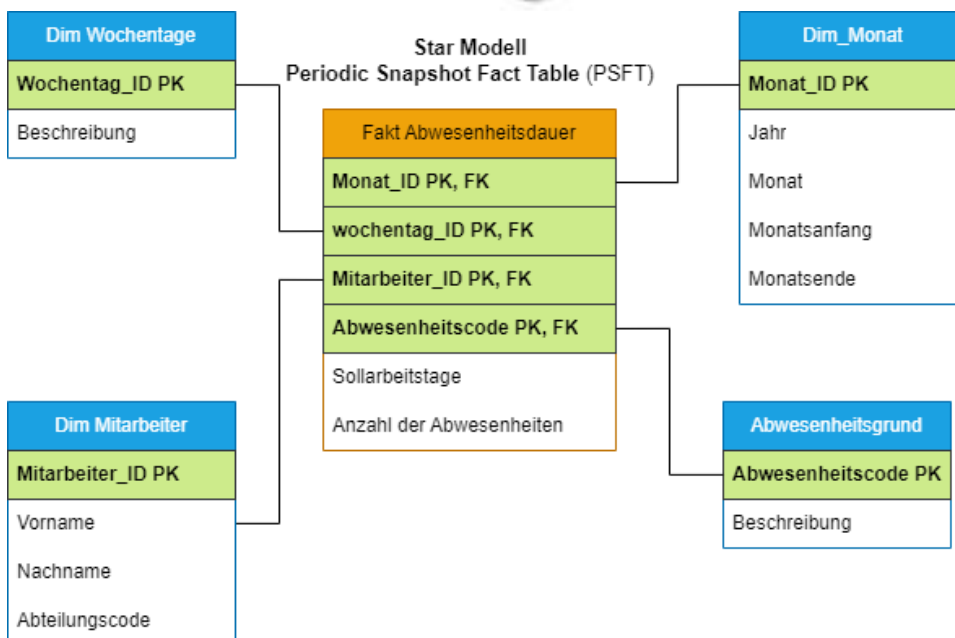
Vorteil:

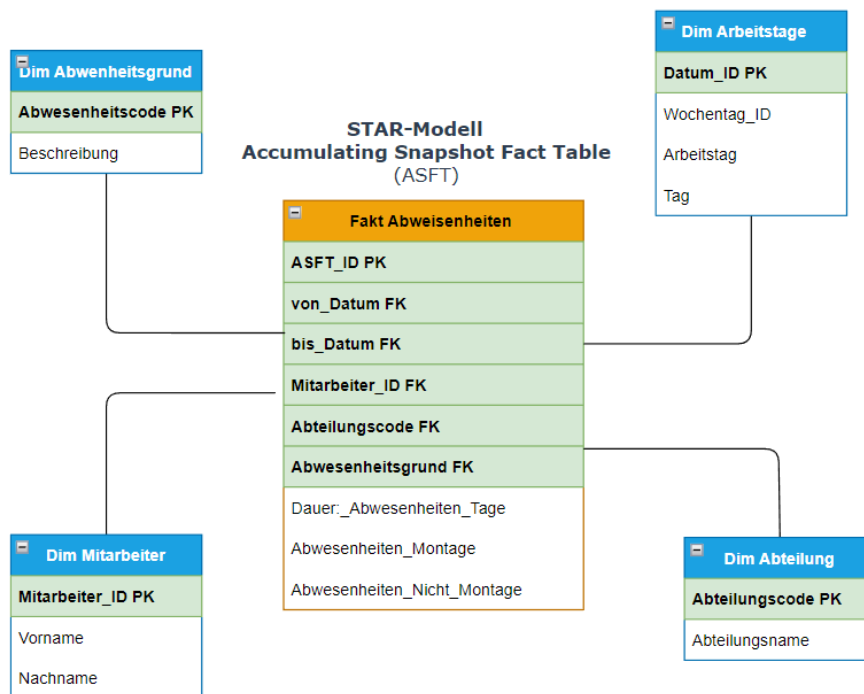
- PSFT ist sehr gut geeignet für die Erstellung von Berichten, da sie eine Vielzahl von Kennzahlen enthält, sowohl relative wie **Krankenstand** als auch absolute wie **Mitarbeiter-Abwesenheitstage**. Diese Kennzahlen können einfach schnell aus der Tabelle abgeleitet und berechnet werden und bieten eine einfache Analysemöglichkeit.

Nachteil:

- Der Zusammengesetzte Schlüssel, der aus den IDs für **Wochentag**, **Mitarbeiter** und **Monat** besteht, kann den Zugriff auf Daten verlangsamen und die Abfragegeschwindigkeit verringern, da er eine komplexere Struktur hat als ein einfacher Primärschlüssel.
- Einzelne Ereignisse sind in der PSFT nicht mehr identifizierbar, da sie aggregiert und zu einem bestimmten Zeitpunkt dargestellt werden.

1.3 DESIGN FAKTENTABELLE DIMENSIONSTRUKTUREN FÜR GEWÄHLTE FAKTENTABELLEN





1.4 DESIGN ZUR BEFÜLLUNG DER GEWÄHLTEN FAKTENTABELLEN

Pseudocode für PSFT

//Push

Schleife über alle Datensätze rec_mo von **Dim_Monat**

Schleife über alle Datensätze rec_w von **Dim_Wochentag**

Schleife über alle Datensätze rec_mi Von **Dim_Mitarbeiter**

Schleife über alle Datensätze rec_mi Von **Dim_Abteilung**

IF Dim_Mitarbeiter.abteilungscode = Dim_Abteilung.Abteilungscode **then**

PSFT-abteilungscode <- rec_mi.Abteilungscode

End IF

Schleife über alle Datensätze rec_ab Von **Dim_Abwesenheitsgrund**

PSFT.Monat_ID <- rec_mo.Monat_ID

PSFT.Wochentag_ID <- rec_w.Wochentag_ID

PSFT.Mitarbeiter_ID <- rec_mi.Mitarbeiter_ID

PSFT.Abwesenheitgrund <- rec.ab.Abwesenheitgrund

Erstelle einen neuen Datensatz mit PSFT.Monat_ID

PSFT.Wochentag_ID, PSFT.Mitarbeiter_ID,

PSFT.Abteilungscode, PSFT.Abwesenheitgrund

Endschleife //Dim_Abwesenheitsgrund

Endschleife // Dim_Abteilung

Endschleife //Dim_Mitarbeiter

Endschleife //Dim_Wochentag

Endschleife //Dim_Monat

//Pull

Schleife über alle Datensätze rec von PSFT

Rec_Monat <- lese Datensatz aus der Tabelle **Dim_Monat** mit der Bedingung Monat_ID = rec.Monat_ID

Monatsanfang <- **rec_Monat**.Monatsanfang

Monatsende <- **rec_Monat**.Monatsende

T_tmp <- filtere Tabelle **Arbeitstag_tab** mit der Bedingung Datum_ID zwischen Monatsanfang und Monatsende UND Wochentag_ID = rec.Wochentag_ID UND Arbeitstag = wahr

Rec.**Sollarbeitstag** <- Count...des Feldes Datum_ID von Tabelle **T_tmp**

Rohdaten_temp <- Filtere Tabelle Rohdaten mit Bedingung Rohdaten.Mitarbeiter_ID = rec.Mitarbeiter_ID und rehdaten.Abwesenheitsgrund = rec.Abwesenheitsgrund // hier werden alle Einträge für jeden Mitarbeiter und ihren Abwesenheitsgrund gefiltert

Temp_count <- 0

Schleife über alle Datensätze **rec_roh** von **Rohdaten_temp**

Anzahl <- 0

Falls rec_roh.ersterMonat = rec.Monat_ID // hier betrachten wir den ersten Monat von Awesenheit

Falls rec_roh.bis_Datum <= Monatsende // falls die Abwesenheitdauer nicht den aktuellen Monat überschreitet z.B von Monat .7 bis Monat .7

T_tmp <- filtere Tabelle **Arbeitstag_tab** mit der Bedingung Datum_ID zwischen rec_roh.von_Datum und rec_roh.**bis_Datum** UND Wochentag_ID = rec.Wochentag_ID UND Arbeitstag = wahr

Anzahl <- Count...des Feldes Monat_ID der Tabelle **T_tmp**

Temp_count <-Temp_count + Anzahl

Andernfalls //rec_roh.bis_Datum > Monatsende falls die Abwesenheitdauer den aktuellen Monat überschreitet z.B von Monat .7 bis Monat .8

T_tmp <- filtere Tabelle **Arbeitstag_tab** mit der Bedingung Datum_ID zwischen rec_roh.von_Datum und **Monatsende** UND Wochentag_ID = rec.Wochentag_ID UND Arbeitstag = wahr

Anzahl <- Count...des Felders Monat_ID der Tabelle T_tmp

Temp_count <-Temp_count + Anzahl

Endefalls

Falls rec_roh.zweiterMonat = rec.Monat_ID

T_tmp <- filtere Tabelle **Arbeitstag_tab** mit der Bedingung Datum_ID zwischen **Monatsanfang** und rec_roh.bis_Datum UND Wochetag_ID = rec.Wochentag_ID UND Arbeitstag = wahr

Anzahl <- count...des Felders Monat_ID der Tabelle T_tmp

Temp_count <-Temp_count + Anzahl

EndeFalls

EndeSchleife

Rec.**Anzahl_der_Abwesenheitstage**<- Temp_count

Endeschleife

Pseudocode für ASFT

Erstelle Tabelle ASFT_Abwesenheit mit Spalten ASFT_ID, von_Datum, bis_Datum, Mitarbeitercode, Abteilungscode, Abwesenheitsgrund, Dauer_Abwesenheit_Montage, Abwesenheiten_Montage, Abwesenheiten_NichtMontag

ID := 0

// PK und FK befüllen

Schleife über alle Datensätze rec von Tabelle Rohdaten

ASFT_ID := ID +1

Von_Datum := rec.von_Datum

Bis_Datum := rec.bis_Datum

Mitarbeitercode := rec.Mitarbeiter_ID

Abteilungscode := rec.Abteilungscode

Abwesenheitsgrund := rec.Abwesenheitsgrund

Füge Datensatz in die Tabelle ASFT_Abwesenheiten

Ende Schleife

// Fakten befüllen

Schleife über alle Datensätze **rec** von Tabelle ASFT_Abwesenheit

Arbeitstage_tab_tmp ← Filtere Tabelle Arbeitstage_tab im Intervall von Datum_ID = **rec.von_Datum** bis Datum_ID = **rec.bis_Datum** und der Bedingung Arbeitstag = WAHR

ArbeitstageAbwesend:= Zähle Datensätze von Tabelle **Arbeitstage_tab_tmp**

Rec.Dauer_Abwesenheiten_Tage := ArbeitstageAbwesend

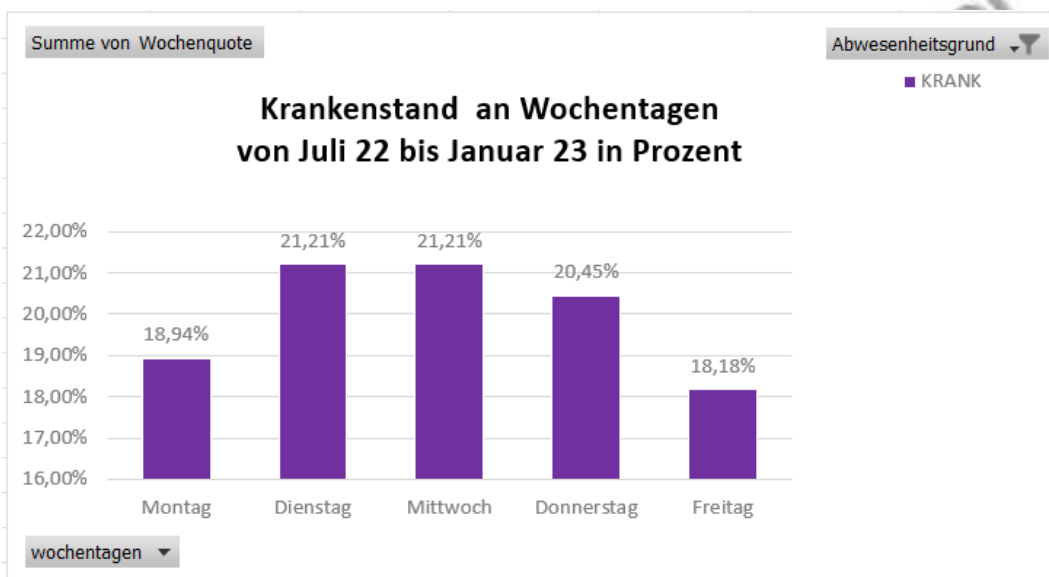
Arbeitstage_tab_tmp ← Filtere Tabelle Arbeitstage_tab im Intervall von Datum_ID = **rec.von_Datum** bis Datum_ID = **rec.bis_Datum** und der Bedingung Arbeitstag = WAHR und Wochentag_ID = 1 // 1 = Montag

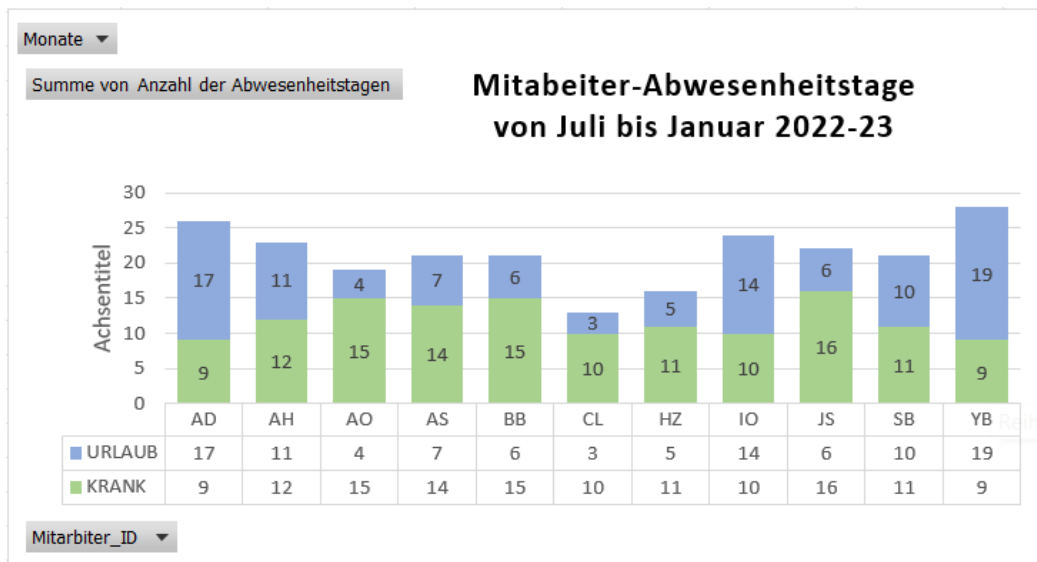
Rec.Abwesenheiten_Montage := Zähle Datensätze von Tabelle **Arbeitstage_tab_tmp**

Rec.Abwesenheiten_Nicht_Montage := **rec.Dauer_Abwesenheiten_Tage** – **rec.Abwesenheiten_Montage**

Ende Schleife**ANHANG****Exemplarisches Excel-Pivot Diagramm für die Aufgabenstellung**

Formel:Wochenquote='Anzahl der Abwesenheitstagen'/ 132 // 132 ist die Gesamttabwesenheitstage als krank





Formel: Krankheitsquote = 'Anzahl der Abwesenheitstagen' / Sollarbeitstage



MEILENSTEIN 2

Für den zweiten Meilenstein soll ein Konzept entwickelt werden, wie die Analyseplattform, für die Sie bereits in der ersten Phase Kennzahlen und zwei STAR-Schemata entworfen haben, im ERP-System Business Central umgesetzt werden kann. Für den Meilenstein 2 brauchen Sie nichts in Business Central zu implementieren. Wir arbeiten im Meilenstein 2 im Wesentlichen konzeptionell. Trotzdem ist es an einigen Stellen nötig, sich das Video „Mitarbeiter und Mitarbeiterabwesenheiten anlegen“ der Selbst-Lerneinheit anzuschauen, obwohl die Schritte innerhalb BC erst für den Meilenstein 3 zu realisieren sind. Sie werden zB für den Meilenstein 3 alle **Teammitglieder als Mitarbeiter** in Business Central anlegen und diese jeweils Abteilungen zuordnen. Für den Meilenstein 2 müssen Sie wissen, welche Abteilungen zur Verfügung stehen (man siehe dazu das o.g. Video).

Folgende Schritte sind für den Meilenstein 2 erforderlich:

1. Legen Sie fest, wie die **Mitglieder Ihres Teams als Mitarbeiter** in Business Central den Abteilungen zugeordnet werden. Mindestens zwei Mitarbeiter sollten zu ein und derselben Abteilung gehören.
2. Legen Sie fest, welche **Arbeitstage** für die Cronus AG relevant sein sollen. Dies ist für die konkrete Durchführung von Tests und die Verifizierung, dass die Kennzahlen wie Krankenstand korrekt ermittelt werden, nötig. Wir benötigen dafür einen Kalender als Berechnungsgrundlage, aus dem die Arbeitstage hervorgehen. Konkret heißt das: welche Feiertage sind gültig. Obligatorisch ist, dass an regulären Wochentagen (Mo-Fr) gearbeitet wird. Eine Angabe wie NRW-Feiertagskalender wäre zB ausreichend. Sie werden später für den Meilenstein 3 diese Daten in einen „leeren“ Kalender des ERP-Systems anlegen müssen oder einen bestehenden Kalender entsprechend modifizieren (s. Video).
3. Planen Sie einen Testcase, in dem Sie für jeden von Ihnen angelegten Mitarbeiter mindestens zwei Abwesenheitsdatensätze für November+Dezember 2021 + Januar 2022 vorsehen. Vorkommen sollten
 - i. Mindestens zwei eintägige Abwesenheiten an Montagen
 - ii. Mindestens eine mehrtägige Abwesenheit über ein Wochenende hinweg
 - iii. Mindestens eine mehrtägige Abwesenheit über den Jahreswechsel hinweg
 - iv. Mindestens eine mehrtägige Abwesenheit über einen Feiertag hinweg
- b. Legen Sie diese Abwesenheiten als Datensätze in dem Excel-Sheet an, das Sie für den Meilenstein 1 bereits verwendet haben. Ergänzen Sie im selben Excelsheet, wie diese Abwesenheitsdaten mit dem Kontext der neu angelegten Mitarbeiter und den zwei Monatsfenster sich in den Fakten- bzw Dimensionstabellen widerspiegeln.
- c. Berechnen Sie mit Hilfe von Excel, die von Ihnen geplanten Kennzahlen für die Abwesenheitsanalyse für den Testfall Krankenstand in einer vorgegebenen Abteilung mit Aufriss nach Monaten exemplarisch aus den Fakten- und Dimensionstabellen!
- d. Planen Sie einen weiteren Testcase, der dazu dient zu testen, ob das Programm **fehlerhafte Daten** erkennt. Exemplarisch soll folgende Konstellation erkannt und ausgewiesen werden:
 - i. Zwei mehrtägige Abwesenheiten, die sich überschneiden, sodass ein Mitarbeiter mindestens an einem Arbeitstag zB gleichzeitig als abwesend wegen Krankheit und wegen Urlaub registriert ist.

4. Entwerfen Sie die für Ihre STAR-Modelle benötigten Tabellen, die Sie später in Business Central anlegen werden. Dazu gehören insbesondere die Angabe aller Felder sowie die Spezifikation von **Primärschlüsseln** und **Fremdschlüsselbeziehungen** zB

Tabelle 1: Dim_Dep XYZ

Field	Type	Spec.	Description
<u>DepID</u>	Text	PK	Department <u>identifier</u>
<u>Description</u>	Text		Name of department
...			

Tabelle 2: Facttable PSFT XYZ

Field	Type	Spec.	Description
<u>DepID</u>	Text	PK und FK zu <u>Dim_Dep.No</u>	Department <u>identifier</u>
<u>DayId</u>	Datum	PK und FK zu <u>Dim_Zeit.Day</u>	<u>Calendar day Identifier</u>
<u>NumAbsDays</u>	Zahl		Number of Lost Working Days by Absence of an employee
...			

Folgende für die Bearbeitung der Aufgabe nötige oder hilfreiche Datenstrukturen stehen Ihnen im Business Central zur Verfügung:

- **5200 Employee**
- **5206 Cause of Absence**
- **5207 Employee Absence**
- **Date** (Systemtabelle, Hinweise dazu im Anhang, bzw in der Lerneinheit)

Schauen Sie sich die Tabellenbeschreibungen zu 5200, 5207 und Date an (per Visual Studio Code) und ermitteln Sie, welche Felder Sie für die Generierung Ihrer Tabellen voraussichtlich benötigen werden. Beschreiben Sie nur die für das Praktikum relevanten Felder in übersichtlicher Form (zB in der nachfolgenden exemplarischen Struktur):

Tabelle 3: 5200 Employee

Field	Type	Spec.	Description
<u>No.</u>	Code 10	PK	<u>Employee Id: Mitarbeiternummer</u>
...			
...			
...			

5. Visualisieren Sie zusätzlich als Überblick die Datenflüsse zwischen den Tabellen (ein exemplarisches Modell ist im Anhang dargestellt).
6. Beschreiben Sie alle Befüllungsprozesse der Dimensionstabellen als Pseudocode. Im Meilenstein 1 haben Sie bereits die Befüllung der Faktentabellen beschrieben. Kontrollieren Sie nochmals auch diesen Programmmentwurf.

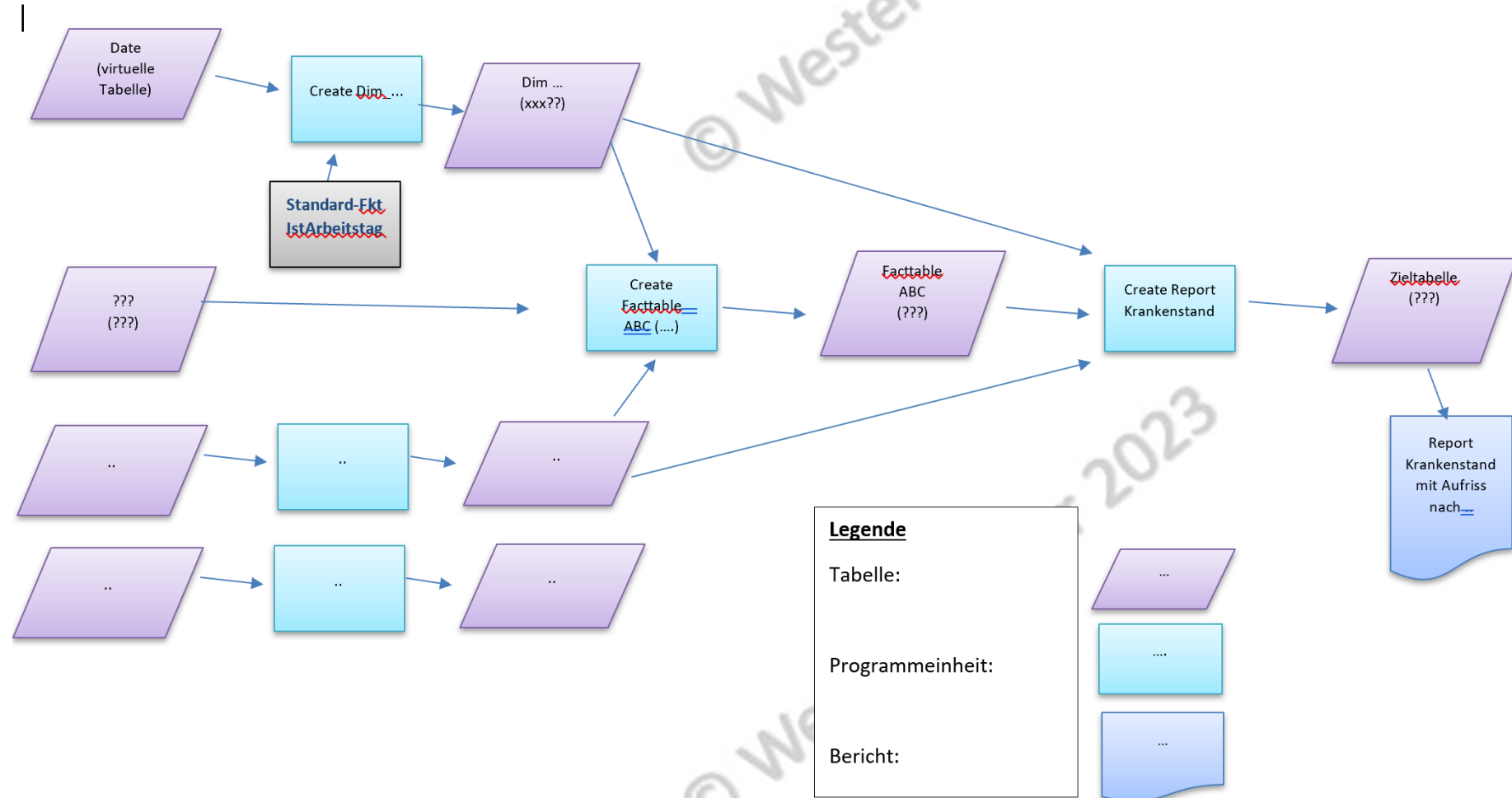
7. Beschreiben Sie per Pseudocode exemplarisch, wie aus den STAR-Tabellen der **Krankenstand an Montagen in einer vorgegebenen Abteilung** für einen **Aufriss nach Monaten für das Jahr 2021** berechnet und in eine Zieltabelle geschrieben werden kann (exemplarisch s. nachfolgende Tabelle 4: Demozieltabelle).

Tabelle 4: Demozieltabelle

Field	Type	Spec.	Description
<u>DepID</u>	Code 10	PK	<u>DepId: Department identifier</u>
<u>Abteilungsname</u>	Text		<u>zB Produktion</u>
<u>MonthId</u>	Code10	PK	<u>zB 2021-01</u>
<u>Monatsname</u>	Text		<u>zB 2021 Januar</u>
<u>ArbeitstageMontag</u>	Integer		<u>zB 4</u>
<u>KrankheitstageMontage</u>	Integer		<u>Zb 1</u>
<u>KrankenstandMontage</u>	Decimal		<u>zB 0.25</u>

Sie können auch eine andere Zieltabelle wählen. Im Rahmen des Proof-of-Concepts soll der Unternehmensleistung exemplarisch gezeigt werden, dass für sie interessante Berichte generiert werden können. Ansonsten ist zu erläutern, dass auch andere Berichte mit geringem Aufwand aus dem geplanten STAR-Modell erzeugt werden können.

Sie können bei Ihrem Konzeptentwurf die Existenz einer Standardfunktion IstArbeitstag(Date) voraussetzen und verwenden, die für einen beliebigen Kalendertag „Date“ ermittelt, ob dieser im relevanten Kalender als Arbeitstag spezifiziert ist. Eine Auseinandersetzung mit den Tabellen im Business Central Standard, die die arbeitsfreien Tage verwalten, ist nicht nötig.

ANHANG**Exemplarische Visualisierung der Datenflüsse zwischen den Tabellen**

Virtuelle Tabelle Date

Die virtuelle Datumstabelle DATE (ID 2000000007) in Business Central bietet Zugriff auf Tage, Wochen, Monate, Quartale und Jahre.

Field	Description
Period Type	0,1,2,... entsprechen (Tage, Wochen, Monate, Quartale oder Jahre).
Period Start	Das Datum des ersten Tages in der Periode
Period End	Das Datum des letzten Tages in der Periode
Period No.	Die Nummer des Zeitraums, zB bei Wochen die Kalenderwoche, bei Monaten die Monatsnummer
Period Name	zB bei Tage der Wochentag

Der zusammengesetzte Primärschlüssel von Date ist "Period Type", "Period Start".

++ 1.4 2-

2.1 FESTLEGUNG MITARBEITER

IO	Ismail	Ouabi	VERKAUF
AS	Alexander	Schmidt	VERWALTUNG
CL	Christian	Ludwig	VERKAUF
JS	Julian	Szudrowicz	PRODUKTION

2.2 FESTLEGUNG KALENDER

Siehe Excel Arbeitsblatt „NRW-Feiertage 22 und 23“ und Arbeistage_tab

2.3 GEPLANTE TESTCASES

AS	VERW	24.11.2022	27.11.2022	Urlaub
AS	VERW	27.11.2022	09.12.2022	Krank
AD	VERKAUF	28.11.2022	02.12.2022	Urlaub
IO	VERKAUF	28.11.2022	02.12.2022	Urlaub
SB	PROD	29.11.2022	01.12.2022	Krank
HZ	PROD	30.11.2022	03.12.2022	Krank
YB	VERW	01.12.2022	04.12.2022	Urlaub
YB	VERW	01.12.2022	05.12.2022	Urlaub
CL	VERKAUF	05.12.2022	07.12.2022	Urlaub
AD	VERKAUF	06.12.2022	20.12.2022	Urlaub
BB	VERKAUF	13.12.2022	13.12.2022	Krank
JS	PROD	14.12.2022	05.01.2023	Krank
IO	VERKAUF	19.12.2022	24.12.2022	Urlaub
AO	PROD	20.12.2022	05.01.2023	Krank
AS	VERW	22.12.2022	29.12.2022	Urlaub
BB	VERKAUF	23.12.2022	05.01.2023	Krank
IO	VERKAUF	25.12.2022	01.01.2023	Urlaub
AS	VERW	28.12.2022	02.01.2023	Krank

(AUSCHNITT AUS ROHDATEN IN EXCEL)

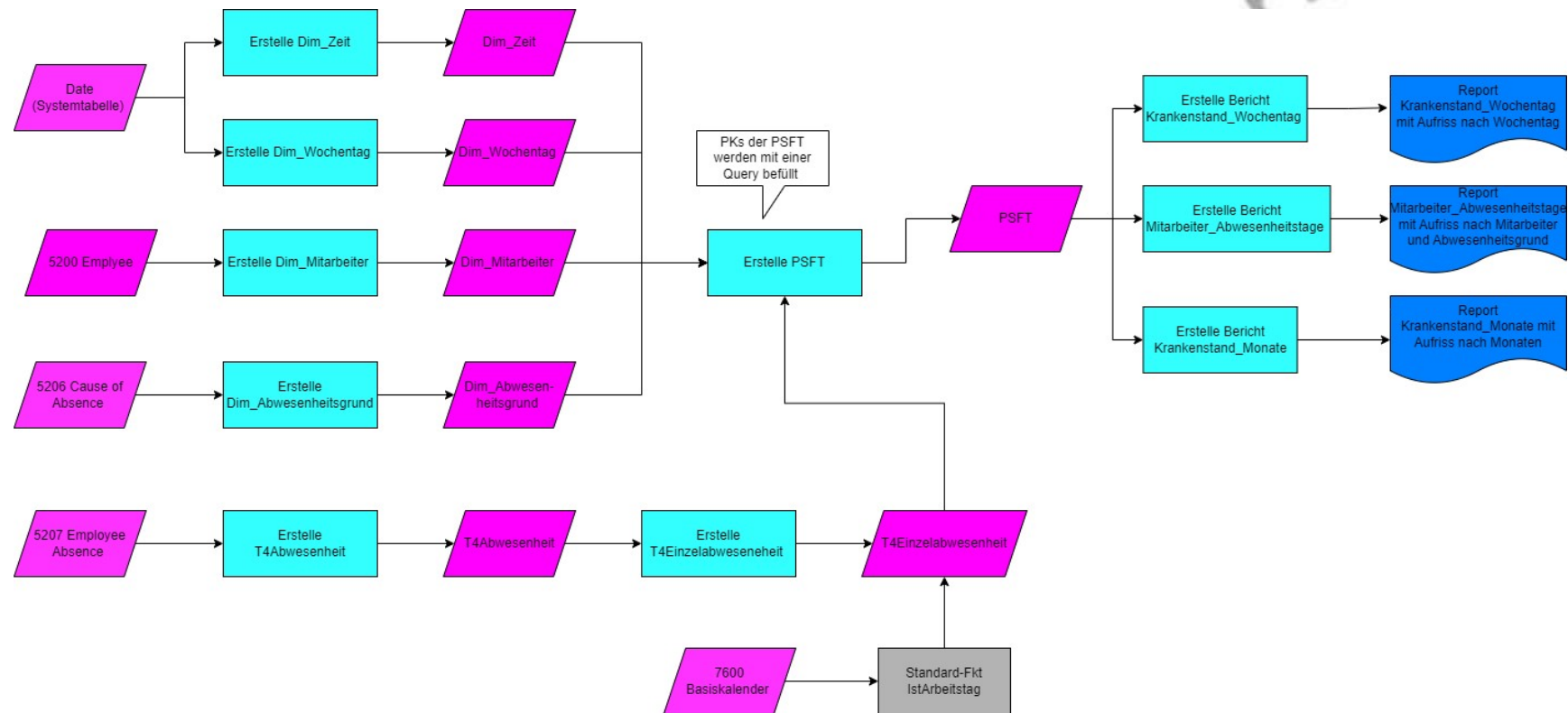
c. Siehe den Bericht in Excel „Krankenstand in Monaten“

d. die rot markierten Zeilen weisen Überscheidungen aus (wir haben mit VBA gearbeitet)

2.4 TABELLE FÜR DIE GEGEBENEN STARMODELLE

Siehe Excel Arbeitsblatt "Tabellen" in unserm Excelldokument

2.5 VISUALISIERUNG DER DATENFLÜSSE



2.6 PSEUDOCODE DER DIMENSIONSTABELLEN

Dim Mitarbeiter

Erstelle Tabelle Dim_Mitarbeiter mit den Spalten Mitarbeitercode, Vorname, Nachname, Abteilungscode

Schleife über alle Datensätze rec von Employee

Füge in Dim_Mitarbeiter

(

```
Mitarbeitercode := rec.Mitarbeitercode,  
Vorname := rec.Vorname,  
Nachname := rec.Nachname,  
Abteilungscode = rec.Arbeitsvertragscode  
)  
Ende Schleife
```

Dim Abwesenheitsgrund

Erstelle Tabelle Dim_Abwesenheitsgrund mit den Spalten Abwesenheitscode,
Schleife über alle Datensätze rec von Cause of Absence
Füge in Dim_Abwesenheitsgrund
(
 Abwesenheitscode := rec.Abwesenheitsgrund
)
Ende Schleife

Dim Zeit

Erstelle Tabelle Dim_Monat mit den Spalten Monat_Jahr_ID, Jahr, Monat, Monatsanfang, Monatsende
Date_tmp <-- Filtere Tabelle Date mit Bedingung Period Type = Monat
Schleife über alle Datensätze rec von Tabelle Date_temp
Füge in Dim_Monat
(

```
Monat_Jahr_ID = Format rec.Startdatum, ' <month,2><Year4> ' )  
Jahr = Date2DMY( rec.Startdatum,3)  
Monatsanfang = rec.PeriodStart  
Monatsende = rec.PeriodEnd  
)  
Ende Schleife
```

Dim_Wochentag

Erstelle Tabelle Dim_Wochentag rec mit den Spalten Wochentag_ID, Beschreibung

Date_temp ← filtere Tabelle Date mit der Bedingung Peridentyp = Tag

Schleife von counter = 1 bis 5

temp_counter ← lese ersten Datensatz von Tabelle Date_temp mit der Bedingung Periodennr = counter

rec.Wochentag_Id ← counter

rec.Beschreibung ← Date_temp.Periodenname

Erstelle einen Datensatz mit rec.Wochentag_Id, rec.Beschreibung

Ende Schleife

Dim_Arbeitstage

Erstelle Tabelle Dim_Arbeitstage mit Spalten Datum_ID, Wochentag_ID, Arbeitstag, Tag

Schleife über alle Datensätze rec Tabelle Basiskalender (7600)

Füge Datensatz in Dim_Arbeitstage

```
(  
    Datum_ID := rec.Datum  
    Arbeitstag = IstArbeitstag(rec.Datum) (Standardfunktion)  
    Tag := rec.Tag  
    Wochentag_ID := Date2DWY(rec.Datum,1)  
)  
Ende Schleife
```

2.7 PSEUDOCODE DES BERICHTS

//Push von DIMs nach Zieltabelle

Erstelle ein Tabelle ZielTabelle mit Spalten DepID, MonatId, Abteilungsname, Monatsname, ArbeitstageMontage, KrankenstageMontage, KrankenstandMontage

Schleife über alle Datensätze rec_ab Dim_Abteilung

Schleife über aller Datensätze rec_mo Dim_Monat

ziel_tab.Dep_id <- rec_ab.Dep_id

ziel_tab.Monat_id <- rec_mo.Monat_Id

ziel_tab.Dep_Abteilungsname <- rec_ab. Abteilungsname

ziel_tab.Monatsname <- rec_mo.Monat

erstelle einen Datensatz mit ziel_tab.Dep_id, ziel_tab.Monat_id, ziel_tab.Dep_Abteilungsname, ziel_tab.Monatsname

Endschleife

Endschleife

//PULL aus PSFT

Schleife über alle Datensätze rec von Zieltabelle

PSFT_temp <- filtere PSFT mit der Bedingung Monat_Id = rec.Monat_Id,

Abteilungscode = rec.Dep_Id, Wochentag_id = 1

`Record` <- Lese ersten Datensatz aus der Tabelle `PSFT_Temp`

`Rec.ArbeistageMontag` <- `record`.Sollarbeitstage

`Rec.KrankheitstageMontage` <- count...des Feldes `Anzahl_der_Abwesenheitstage`

`Rec.KrankenstandMonateg` <- `Rec.KrankheitstageMontage` / `Rec.ArbeistageMontag`

Endschleife

3. ÄNDERUNGEN VON MS2 ZU MS3

- Dimension "Abteilung" entfernt, weil das später zu Problemen im Kreuzprodukt in der PSFT geführt hätte
- ASFT wird nicht mehr betrachtet
- Neue Tabellen für Abwesenheitsregistrierung (T4Abwesenheit und T4EinzelAbwesenheit), um einerseits unsere Abwesenheitsdaten von den Bestehenden zu unterscheiden und um die Abwesenheiten in einzelenen Tage unterbrechen zu können

3.1 STAMMDATEN EINFLEGEN

1. Pflegen Sie die nötigten Stammdaten in Businnes Central ein

M0030	Julian	Szudrowicz	—	JSZUDROW...	N
M0040	Ismail	Ouabi	—	IOUABI	N
M0050	Alexander	Schmidt	—	ASCHMIDT	N
M0060	Christian	Ludwig	—	CLUDWIG	N

Wiederholungsmus... ↑	Datum ↑	Tag ↑	Beschreibung	Frei
-	02.01.2023	Montag		<input type="checkbox"/>
Jährlich	03.10.2022		Tag der Deutschen Einheit	<input checked="" type="checkbox"/>
Jährlich	01.11.2022		Allerheiligen	<input checked="" type="checkbox"/>
Wöchentlich		Samstag		<input checked="" type="checkbox"/>
Wöchentlich		Sonntag		<input checked="" type="checkbox"/>

(Benutzerdefinierte Kalenderänderungen 6702)

3.2 IMPLEMENTIEREN SIE DAS VON IHNEN ENTWORFENE DESIGN

Erklärungstext zur Vorgehensweise:

Wir haben zuerst neue Tabellen angelegt für die Abwesenheitsregistrierung. Die Tabelle "T4Abwesenheit" wird mit den Daten aus "Employee Absence" befüllt, aber mit einem Filter für unsere Mitarbeiter. Die Tabelle "T4EinzelAbwesenheit" wird aus "T4Abwesenheit" befüllt mit der Bedingung, dass die Abwesenheiten runtergebrochen werden auf einzelne Tage. Dabei entsteht ein zusammengesetzter PK aus der "Entry.No" und dem "Datum". Für jede Dimension wird eine Tabelle angelegt und eine Page vom Typ "List". In den jeweiligen Lists haben wir Actions angelegt, die die Dimensionstabellen befüllen. Mithilfe einer Query, die die Dimensionen mit Hilfe eines Crossjoins zusammenführt, und einer Codeunit werden die PKs der PSFT befüllt. Anschließend haben wir noch eine Page "T4Zieltabelle" angelegt, in der die PSFT als ListPart eingebettet wird und wo diverse Actions vorhanden sind (Berichte erstellen, Fakten befüllen usw.).

(siehe Anhang)

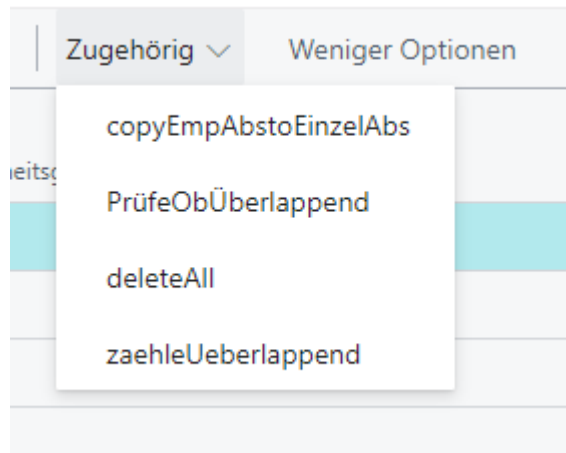
3.3 TESTCASES EINFÜHREN

employeeNo	von_Datum	bis_datum	Abwesenheitsgrund	Menge	EntryNo ↑
M0040	07.10.2022	09.10.2022	KRANK	2	673
M0040	13.10.2022	15.10.2022	KRANK	4	674
M0040	18.11.2022	28.11.2022	KRANK	5	675
M0040	28.11.2022	02.12.2022	URLAUB	4	676
M0040	19.12.2022	24.12.2022	URLAUB	7	677
M0040	25.12.2022	01.01.2023	URLAUB	5	678
M0030	03.10.2022	07.10.2022	KRANK	4	679
M0030	27.10.2022	02.11.2022	KRANK	4	680
M0030	14.11.2022	21.11.2022	URLAUB	6	681
M0030	01.12.2022	05.12.2022	URLAUB	3	682
M0030	24.12.2022	03.01.2023	KRANK	6	683
M0050	24.11.2022	27.11.2022	URLAUB	2	684
M0050	29.11.2022	09.12.2022	KRANK	9	685
M0050	22.12.2022	29.12.2022	URLAUB	5	686
M0050	27.12.2022	02.01.2023	KRANK	5	687
M0060	07.11.2022	10.11.2022	KRANK	4	688
M0060	01.12.2022	04.12.2022	URLAUB	2	689
M0060	06.12.2022	07.12.2022	URLAUB	2	690
M0060	25.12.2022	05.01.2023	KRANK	8	691

T4EinzelAbwesenheit: Alle ▾ | Suchen + Neu Löschen Liste bearbeiten | Weitere Optionen

employeeNo	Datum ↑	Abwesenheitsgrund	Menge	istÜberlappend	entryNo ↑
M0030	04.10.2022	KRANK	1	<input type="checkbox"/>	679
M0030	05.10.2022	KRANK	1	<input type="checkbox"/>	679
M0030	06.10.2022	KRANK	1	<input type="checkbox"/>	679
M0040	07.10.2022	KRANK	1	<input type="checkbox"/>	673
M0030	07.10.2022	KRANK	1	<input type="checkbox"/>	679
M0040	13.10.2022	KRANK	1	<input type="checkbox"/>	674
M0040	14.10.2022	KRANK	1	<input type="checkbox"/>	674
M0030	27.10.2022	KRANK	1	<input type="checkbox"/>	680
M0030	28.10.2022	KRANK	1	<input type="checkbox"/>	680
M0030	31.10.2022	KRANK	1	<input type="checkbox"/>	680
M0030	02.11.2022	KRANK	1	<input type="checkbox"/>	680
M0060	07.11.2022	KRANK	1	<input type="checkbox"/>	688
M0060	08.11.2022	KRANK	1	<input type="checkbox"/>	688
M0060	09.11.2022	KRANK	1	<input type="checkbox"/>	688
M0060	10.11.2022	KRANK	1	<input type="checkbox"/>	688
M0030	14.11.2022	URLAUB	1	<input type="checkbox"/>	681
M0030	15.11.2022	URLAUB	1	<input type="checkbox"/>	681
M0030	16.11.2022	URLAUB	1	<input type="checkbox"/>	681
M0030	17.11.2022	URLAUB	1	<input type="checkbox"/>	681

Das Feld "IstÜberlappend" zeigt an, ob es eine Abwesenheit doppelt gibt --> "EmployeeNo" und "Datum" stimmt überein



Die Action “PrüfeObÜberlappend” prüft dann doppelte Abwesenheiten und setzt “IstÜberlappend” auf true

M0040	28.11.2022	KRANK	1	✓	675
M0040	28.11.2022	URLAUB	1	✓	676
M0050	27.12.2022	URLAUB	1	✓	686
M0050	27.12.2022	KRANK	1	✓	687
M0060	27.12.2022	KRANK	1	✓	691
M0060	27.12.2022	URLAUB	1	✓	692
M0050	28.12.2022	URLAUB	1	✓	686
M0050	28.12.2022	KRANK	1	✓	687
M0060	28.12.2022	KRANK	1	✓	691
M0060	28.12.2022	URLAUB	1	✓	692
M0050	29.12.2022	URLAUB	1	✓	686
M0050	29.12.2022	KRANK	1	✓	687
M0060	29.12.2022	KRANK	1	✓	691
M0060	29.12.2022	URLAUB	1	✓	692
M0060	30.12.2022	KRANK	1	✓	691
M0060	30.12.2022	URLAUB	1	✓	692
M0060	02.01.2023	KRANK	1	✓	691
M0060	02.01.2023	URLAUB	1	✓	692
M0060	03.01.2023	KRANK	1	✓	691

Bei der Berechnung wird dann wie folgt mit diesen Datensätzen umgegangen --> Es wird nach dem Abwesenheitsgrund geschaut, weil wenn doppelte Abwesenheiten unterschiedliche Gründe haben, wird nur der Datensatz mit "Krank" betrachtet. Das liegt daran, dass ein geplanter Urlaub sich verlegt, wenn man in dieser Zeit erkrankt.

4 DOKUMENTATION

Unsere Zieltabelle

[PSFTBefüllen](#)

Berichte

[⚡ PKs_befüllen](#) [⚡ Fakten_befüllen](#) [⚡ Delete_Fakten](#) [⚡ GesamtAbwesenheitstage](#) [⚡ deleteALL](#)

General

Abteilung ALLE ▾

Abwesenheitsgrund KRANK oder URLAUB ▾

Mitarbeiter ALLE ▾

Zieltabelle

T4PSFT ▾

Zeit_ID ↑	Wochentag_ID ↑	Mitarbeiter_ID ↑	Abwesenheitsg... ↑	Anzahl der Abwesenheita... ↑	Sollarbeitstage	Abwesenheitsquote	
→ 2022/10	⋮	1	M0030	KRANK	1	4	0,25
2022/10		1	M0030	URLAUB	0	4	0,00
2022/10		1	M0040	KRANK	0	4	0,00
2022/10		1	M0040	URLAUB	0	4	0,00

Mitarbeiter Details

General

Nr. M0030
Vorname Julian
Nachname Szudrowicz
Abteilungscode PROD

PSFTBefüllen


[Berichte](#)[⚡ wochentage_bericht](#) [⚡ Monate_bericht](#) [⚡ Mitarbeiter_Abwesenheitstage](#)

General

Berechnung der Sollarbeitstage von allen Mitarbeitern + Anzahl Gesamtabwesenheiten:

154	012023	5	M0030	KRANK	4	0
155	012023	5	M0030	URLAUB	4	0
156	012023	5	M0040	KRANK	4	0
157	012023	5	M0040	URLAUB	4	0
158	012023	5	M0050	KRANK	4	0
159	012023	5	M0050	URLAUB	4	0
160	012023	5	M0060	KRANK	4	0
161	012023	5	M0060	URLAUB	4	0
162	Ergebnis				672	80
163						
164						

Ausschnitt aus Excel



Anzahl der GesamtAbwesenheitstage: 80

Sollarbeitstage: 672

OK

Ausschnitt aus Business Central

Hier erkennt man, dass die Berechnung im Anhang (Code) abgelegten Code richtig sein muss, da sowohl in der Excel-Tabelle als auch in Business Central die gleichen Werte ausgegeben werden.

Krankenstand an Wochentagen für alle Mitarbeiter:

2				
3	Zeilenbeschriftungen ▾	Summe von Anzahl der Abwesenheitstagen	Summe von wochentagequotee	Summe von Sollarbeitstage
4	Montag	12	9,38%	128
5	Dienstag	15	11,03%	136
6	Mittwoch	15	11,03%	136
7	Donnerstag	19	13,97%	136
8	Freitag	19	13,97%	136
9	Gesamtergebnis	80	11,90%	672
10				

Ausschnitt aus Excel

RANK

i Anzahl der Abwesenheitstage an Wochentagen

Anzahl Abwesenheiten und Krankenstand

für Mitarbeiter: ALLE

Abwesenheitsgrund: KRANK oder URLAUB

Montag--> 12 --> 9,38%

Dienstag--> 15 --> 11,03%

Mittwoch--> 15 --> 11,03%

Donnerstag--> 19 --> 13,98%

Freitag--> 19 --> 13,98%

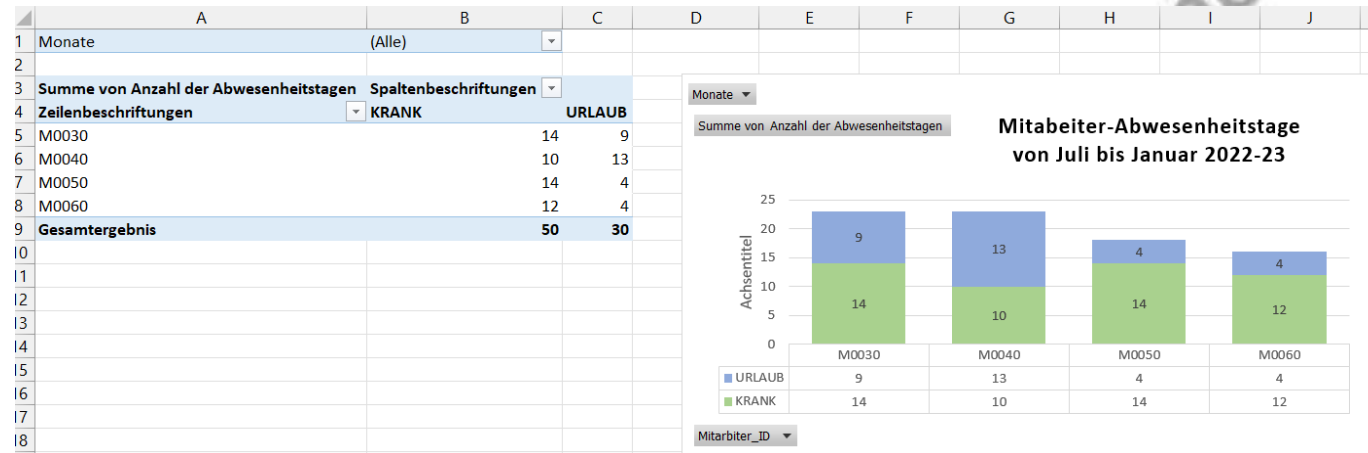
OK

Wochentag_ID ↑	Mitarbeiter	Abwesenheitsgrund	Anzahl Abwesenheiten	Wochentagequote	Sollarbeitstage	Quot
1	M003				4	
1	M003				4	
1	M004				4	
1	M004				4	
1	M0050	KRANK	0		4	
1	M0050	URLAUB	0		4	
1	M0060	KRANK	0		4	

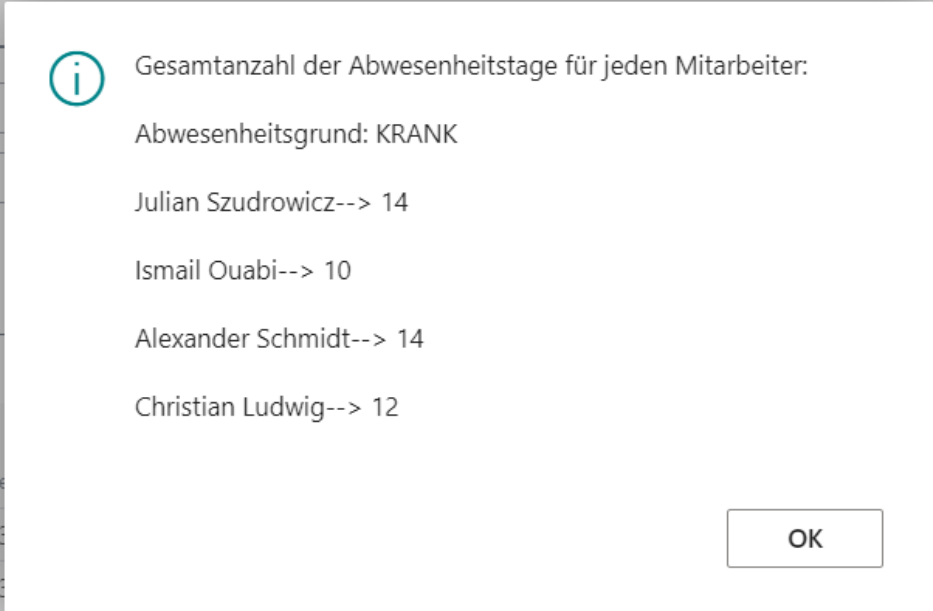
Ausschnitt aus Business Central

Hier erkennt man, dass die Berechnung im Anhang (Code) abgelegten Code richtig sein muss, da sowohl in der Excel-Tabelle als auch in Business Central die gleichen Werte ausgegeben werden.

Anzahl Abwesenheiten pro Mitarbeiter:



Auschnitt aus Excel



The screenshot shows a Business Central interface with a modal dialog box. The dialog box has a title bar with an information icon and the text 'Gesamtanzahl der Abwesenheitstage für jeden Mitarbeiter:'. Below the title, it lists the absence reason 'Abwesenheitsgrund: KRANK' and then displays the total absence days for five employees: Julian Szudrowicz (14), Ismail Ouabi (10), Alexander Schmidt (14), and Christian Ludwig (12). An 'OK' button is at the bottom right of the dialog. In the background, a table is visible with columns for employee ID, absence reason, and absence days. The table contains data for employees M0040 and M0050, showing absence days for 'KRANK' and 'URLAUB'.

Mitarbeiter	Abwesenheitsgrund	Abwesenheitstage
M0040	KRANK	0
M0040	URLAUB	0
M0050	KRANK	0


Ausschnitt aus Business Central

Hier erkennt man, dass die Berechnung im Anhang (Code) abgelegten Code richtig sein muss, da sowohl in der Excel-Tabelle als auch in Business Central die gleichen Werte ausgegeben werden.

Anzahl Abwesenheiten und Krankenstand pro Monat:

1				
2	Jahre	(Alle)		
3				
4	Zeilenbeschriftungen	Summe von Anzahl der Abwesenheitstagen	Summe von Krankheitsquote	Summe von Sollarbeitstage
5	Okt	10	6,25%	160
6	Nov	24	14,29%	168
7	Dez	39	23,21%	168
8	Jan	7	3,98%	176
9	Gesamtergebnis	80	11,90%	672

Ausschnitt aus Excel

 Anzahl der Abwesenheitstage in Monaten
Anzahl Abwesenheiten und Krankenstand
für Mitarbeiter: ALLE
Abwesenheitsgrund: KRANK oder URLAUB
OKTOBER--> 10 --> 6,25%
NOVEMBER--> 24 --> 14,29%
DEZEMBER--> 39 --> 23,22%
JANUAR--> 7 --> 3,98%

OK

Hier erkennt man, dass die Berechnung im Anhang (Code) abgelegten Code richtig sein muss, da sowohl in der Excel-Tabelle als auch in Business Central die gleichen Werte ausgegeben werden.

ANHANG (CODE)

Struktur:

Abschnitte 1-4:

Neue Tabellen für die Abwesenheiten

Abschnitte 5-12:

Dimensionstabellen

Abschnitte 13+14:

Querys und Codeunits

Abschnitte 15+16:

PSFT

Abschnitt 17:

Zieltabelle (Berichte)

Page	actions
DIM_Mitarbeiter	Pull_daten
DIM_Zeit	Pull_daten
DIM_Wochentag	Pull_daten
DIM_Abwesenheitsgrund	Pull_daten
Abwesenheit	SetEmployeeFilter deleteAll
Einzelabwesenheit	copyEmpAbstoEinzelAbs PrüfeObÜberlappend zaehleUeberlappend deleteAll
PSFT	Gruppe 1: PKs_befüllen Fakten_befüllen Delete_Fakten GesamtAbwesenheitstage deleteALL Gruppe2: wochentage_bericht Monate_bericht Mitarbeiter_Abwesenheitstage

Neue Tabellen für die Abwesenheiten:

Abschnitt 1)

T4Abwesenheiten.Table.al

```
table 123456764 T4Abwesenheit
{
    Caption = 'T3Abwesenheit';
    DataClassification = ToBeClassified;
```



```
fields
{
    field(1; employeeNo; Code[20])
    {
        Caption = 'employeeNo';
        DataClassification = ToBeClassified;
    }
    field(2; von_Datum; Date)
    {
        Caption = 'von_Datum';
        DataClassification = ToBeClassified;
    }
    field(3; bis_datum; Date)
    {
        Caption = 'bis_datum';
        DataClassification = ToBeClassified;
    }
    field(4; Abwesenheitsgrund; Code[10])
    {
        Caption = 'Abwesenheitsgrund';
        DataClassification = ToBeClassified;
    }
    field(5; Menge; Integer)
    {
        Caption = 'Menge';
        DataClassification = ToBeClassified;
    }
    field(7; EntryNo; Integer)
    {
        Caption = 'EntryNo';
```

```
        DataClassification = ToBeClassified;
    }
    field(6; istUeberlappend; Boolean)
    {
        Caption = 'istUeberlappend';
        DataClassification = ToBeClassified;
    }
}
keys
{
    key(PK; EntryNo)
    {
        Clustered = true;
    }
}
}
```

Abschnitt 2)

AT4Abwesenheiten.Page.al

```
page 123456766 T4Abwesenheit
{
    ApplicationArea = All;
    Caption = 'T4Abwesenheit';
    PageType = List;
    SourceTable = T4Abwesenheit;

    layout
    {
        area(content)
    }
}
```

```
repeater(General)
{
    field(employeeNo; Rec.employeeNo)
    {
        ToolTip = 'Specifies the value of the employeeNo field.';
    }
    field(von_Datum; Rec.von_Datum)
    {
        ToolTip = 'Specifies the value of the von_Datum field.';
    }
    field(bis_datum; Rec.bis_datum)
    {
        ToolTip = 'Specifies the value of the bis_datum field.';
    }
    field(Abwesenheitsgrund; Rec.Abwesenheitsgrund)
    {
        ToolTip = 'Specifies the value of the Abwesenheitsgrund field.';
    }
    field(Menge; Rec.Menge)
    {
        ToolTip = 'Specifies the value of the menge field.';
    }
    // field(istUeberlappend; Rec.istUeberlappend)
    // {
    //     ToolTip = 'Specifies the value of the istUeberlappend field.';
    // }
    field(EntryNo; Rec.EntryNo)
    {
        ToolTip = 'Specifies the value of the EntryNo field.';
    }
}
```

```
    }  
  }  
  actions  
  {  
    area(Navigation)  
    {  
      action(SetEmployeeFilter)  
      {  
        ApplicationArea = All;  
        Caption = 'filtere Gruppe 4';  
        Image = Filter;  
  
        trigger onaction()  
        begin  
          rec5207.Setfilter("Employee No.", 'M0030|M0040|M0050|M0060');  
          if rec5207.FindFirst() then repeat rec.employeeNo:=rec5207."Employee No.";  
            rec.von_Datum:=rec5207."From Date";  
            rec.bis_datum:=rec5207."To Date";  
            rec.Abwesenheitsgrund:=rec5207."Cause of Absence Code";  
            rec.Menge:=rec5207.Quantity;  
            rec.EntryNo:=rec5207."Entry No.";  
            rec.Insert();  
          until rec5207.Next() = 0;  
        end;  
      }  
      action(deleteAll)  
      {  
        ApplicationArea = All;  
  
        trigger OnAction()  
        begin
```

```
        if rec.FindFirst()then Rec.DeleteAll();
    end;
}
action(zaehleDatensaetze)
{
    ApplicationArea = All;

    trigger OnAction()
    begin
        counter:=0;
        if rec.FindFirst()then repeat counter:=counter + 1;
            until Rec.Next() = 0;
        Message('Anzahl Datensätze :' + Format(counter));
    end;
}
}
}
var rec5207: Record "Employee Absence";
cu7600: Codeunit 7600;
dat: date;
rec7602: Record 7602;
counter: Integer;
}
```

Abschnitt 3)

T4EinzelAbwesenheiten.Table.al

```
table 123456765 T4EinzelAbwesenheit
{
    Caption = 'T4Abwesenheitstage';
    DataClassification = ToBeClassified;
```

```
fields
{
    field(1; employeeNo; Code[5])
    {
        Caption = 'employeeNo';
        DataClassification = ToBeClassified;
    }
    field(2; Datum; Date)
    {
        Caption = 'Datum';
        DataClassification = ToBeClassified;
    }
    field(3; Abwesenheitsgrund; Code[6])
    {
        Caption = 'Abwesenheitsgrund';
        DataClassification = ToBeClassified;
    }
    field(4; EntryNo; Integer)
    {
        Caption = 'entryNo';
        DataClassification = ToBeClassified;
    }
    field(5; istUeberlappend; Boolean)
    {
        Caption = 'istÜberlappend';
        DataClassification = ToBeClassified;
    }
    field(6; Menge; Integer)
    {
        Caption = 'Menge';
```

2023

```
        DataClassification = ToBeClassified;
    }
}
keys
{
    key(PK; Datum, EntryNo)
    {
        Clustered = true;
    }
}
}
```

Abschnitt 4)

T4EinzelAbwesenheiten.Page.al

```
page 123456768 T4EinzelAbwesenheit
{
    ApplicationArea = All;
    Caption = 'T4EinzelAbwesenheit';
    PageType = List;
    SourceTable = T4EinzelAbwesenheit;

    layout
    {
        area(content)
        {
            repeater(General)
            {
                field(employeeNo; Rec.employeeNo)
                {
                    Tooltip = 'Specifies the value of the employeeNo field.';
                }
            }
        }
    }
}
```



```
}
field(Datum; Rec.Datum)
{
    Tooltip = 'Specifies the value of the von_Datum field.';
}
field(Abwesenheitsgrund; Rec.Abwesenheitsgrund)
{
    Tooltip = 'Specifies the value of the Abwesenheitsgrund field.';
}
field(Menge; Rec.Menge)
{
    Tooltip = 'Specifies the value of the menge field.';
}
field(istUeberlappend; Rec.istUeberlappend)
{
    Tooltip = 'Specifies the value of the istUeberlappend field.';
}
field(EntryNo; Rec.EntryNo)
{
    Tooltip = 'Specifies the value of the EntryNo field.';
}
}
}
}
actions
{
    area(Navigation)
    {
        action(copyEmpAbstoEinzelAbs)
        {
            ApplicationArea = All;
        }
    }
}
```



```
        trigger OnAction()
        begin
            if recAbwesenheit.FindFirst() then repeat if rec7602.FindFirst() then for dat:=recAbwesenheit.von_Datum to
recAbwesenheit.bis_datum do // 08.01.2022
            if not cu7600.IsNonworkingDay(dat, rec7602) then // falls arbeitstage ist
            if not Rec.get(dat, recAbwesenheit.EntryNo) then begin
                Rec.EmployeeNo:=recAbwesenheit.employeeNo;
                Rec.EntryNo:=recAbwesenheit.EntryNo;
                Rec.Datum:=dat;
                Rec.Abwesenheitsgrund:=recAbwesenheit.Abwesenheitsgrund;
                Rec.menge:=1;
                Rec.Insert();
            end
            else
                Message('die Abwesenheitstage existieren bereits');
            until recAbwesenheit.Next() = 0;
        end;
    }
    action(PrüfeObÜberlappend)
    {
        ApplicationArea = All;

        trigger OnAction()
        begin
            if rec.FindFirst() then repeat if rec2.FindFirst() then repeat if (rec.Datum = rec2.Datum) and (Rec.employeeNo =
rec2.employeeNo) and (rec.EntryNo <> rec2.EntryNo) then Rec.istUeberlappend:=true;
                Rec.Modify();
            until rec2.Next() = 0;
        until Rec.Next() = 0;
    end;
end;
```

```
}
action(deleteAll)
{
    ApplicationArea = All;

    trigger OnAction()
    begin
        if rec.FindFirst() then Rec.DeleteAll();
    end;
}
action(zaehleUeberlappend)
{
    ApplicationArea = ALL;

    trigger OnAction()
    begin
        if Rec.FindFirst() then repeat if rec.istUeberlappend = true then counter:=counter + 1;
            until Rec.Next() = 0;
        message('Anzahl der ueberlappende Datensätze ist: ' + format(counter));
    end;
}
}

var rec7602: Record 7602;
recAbwesenheit: Record T4Abwesenheit;
rec2: record T4EinzelAbwesenheit;
cu7600: Codeunit 7600;
dat: date;
counter: integer;
}
```

Dimensionstabellen:

Abschnitt 5)

T4Dim_Mitarbeiter.Table.al

```
table 123456760 T4Dim_Mitarbeiter
{
    Caption = 'T4Dim_Mitarbeiter';
    DataClassification = ToBeClassified;

    fields
    {
        field(1; EmployeeNr; Code[10])
        {
            Caption = 'Nr';
            DataClassification = ToBeClassified;
            NotBlank = true; // not null
        }
        field(2; Vorname; Text[15])
        {
            Caption = 'Vorname';
            DataClassification = ToBeClassified;
        }
        field(3; Nachname; Text[30])
        {
            Caption = 'Nachname';
            DataClassification = ToBeClassified;
        }
        field(4; Abteilungscode; Code[30])
        {
            Caption = 'Abteilungscode';
            DataClassification = ToBeClassified;
            TableRelation = "Employment Contract"; // foreign key
        }
    }
}
```

```
    }  
  }  
  keys  
  {  
    key(key1; EmployeeNr)  
    {  
      Clustered = true;  
    }  
  }  
}
```

Abschnitt 6)

T4Dim_Mitarbeiter.Page.al

```
page 123456760 T4Dim_Mitarbeiter  
{  
  ApplicationArea = All;  
  Caption = 'T4Dim_Mitarbeiter';  
  PageType = List;  
  SourceTable = T4Dim_Mitarbeiter;  
  
  layout  
  {  
    area(content)  
    {  
      repeater(General)  
      {  
        field(Nr; Rec.EmployeeNr)  
        {  
          Tooltip = 'Specifies the value of the Nr field.';  
        }  
      }  
    }  
  }  
}
```

```
        field(Vorname; Rec.Vorname)
        {
            Tooltip = 'Specifies the value of the Vorname field.';
        }
        field(Nachname; Rec.Nachname)
        {
            Tooltip = 'Specifies the value of the Nachname field.';
        }
        field(Abteilungscode; Rec.Abteilungscode)
        {
            Caption = 'Abteilungscode';
            Tooltip = 'Specifies the value of the Abteilungscode field.';
        }
    }
}
actions
{
    area(Navigation)
    {
        action(Pull_Data)
        {
            trigger onAction()
            begin
                recEmployee.Setfilter("No.", 'M0030|M0040|M0050|M0060');
                if recEmployee.FindFirst() then repeat rec.EmployeeNr:=recEmployee."No.";
                    rec.Vorname:=recEmployee."First Name";
                    rec.Nachname:=recEmployee."Last Name";
                    rec.Abteilungscode:=recEmployee."Emplymt. Contract Code";
                    rec.insert();
                until recEmployee.Next() = 0;
            end
        }
    }
}
```

```
        end;  
    }  
}  
var recEmployee: record Employee;  
}
```

Abschnitt 7)

T4_Dim_Abwesenheitsgrund.Table.al

```
table 123456763 T4Dim_Abwesenheitsgrund  
{  
    Caption = 'T4Dim_Abwesenheitsgrund';  
    DataClassification = ToBeClassified;  
  
    fields  
    {  
        field(1; "Abwesenheitscode"; Code[6])  
        {  
            Caption = 'Code';  
            DataClassification = ToBeClassified;  
        }  
        field(2; Beschreibung; Text[6])  
        {  
            Caption = 'Beschreibung';  
            DataClassification = ToBeClassified;  
        }  
    }  
    keys  
    {  
        key(PK; Abwesenheitscode)
```

```
    {  
        Clustered = true;  
    }  
}  
}
```

Abschnitt 8)

T4_Dim_Abwesenheitsgrund.Page.al

```
page 123456764 T4Dim_Abwesenheitsgrund  
{  
    ApplicationArea = All;  
    Caption = 'T4Dim_Abwesenheitsgrund';  
    PageType = List;  
    SourceTable = T4Dim_Abwesenheitsgrund;  
  
    layout  
    {  
        area(content)  
        {  
            repeater(General)  
            {  
                field("Code"; Rec.Abwesenheitscode)  
                {  
                    Tooltip = 'Specifies the value of the Code field.';  
                }  
                field(Beschreibung; Rec.Beschreibung)  
                {  
                    Tooltip = 'Specifies the value of the Beschreibung field.';  
                }  
            }  
        }  
    }  
}
```

```
    }  
  }  
  actions  
  {  
    area(Navigation)  
    {  
      action(Pull_Data)  
      {  
        ApplicationArea = ALL;  
  
        trigger OnAction()  
        begin  
          rec5206.Setfilter("Code", 'KRANK|URLAUB');  
          if rec5206.FindSet() then repeat Rec.Abwesenheitscode:=rec5206."Code";  
            Rec.Beschreibung:=rec5206.Description;  
            rec.insert();  
          until rec5206.next = 0;  
        end;  
      }  
    }  
  }  
  }  
  var rec5206: Record "Cause of Absence";  
}
```

Abschnitt 9)

T4_Dim_Zeit.Table.al

```
table 123456761 T4Dim_Zeit
```

```
{  
  Caption = 'T4Dim_Zeit';  
  DataClassification = ToBeClassified;  
}
```



```
fields
{
    field(1; MonatJahr_ID; Code[7])
    {
        Caption = 'MonatJahr_ID';
        DataClassification = ToBeClassified;
    }
    field(2; Jahr; Integer)
    {
        Caption = 'Jahr';
        DataClassification = ToBeClassified;
    }
    field(3; Monatname; Code[10])
    {
        Caption = 'Monatname';
        DataClassification = ToBeClassified;
    }
    field(4; Monatsanfang; Date)
    {
        Caption = 'Monatsanfang';
        DataClassification = ToBeClassified;
    }
    field(5; Monatsende; Date)
    {
        Caption = 'Monatsende';
        DataClassification = ToBeClassified;
    }
    field(6; Quartal; integer)
    {
        Caption = 'Quartal';
```

```
        DataClassification = ToBeClassified;
    }
}
keys
{
    key(PK; MonatJahr_ID)
    {
        Clustered = true;
    }
}
```

Abschnitt 10)

T4_Dim_Zeit.Page.al

```
page 123456761 T4Dim_Zeit
{
    ApplicationArea = All;
    Caption = 'T4Dim_Zeit';
    PageType = List;
    SourceTable = T4Dim_Zeit;

    layout
    {
        area(content)
        {
            repeater(General)
            {
                field(MonatJahr_ID; Rec.MonatJahr_ID)
                {
                    Tooltip = 'Specifies the value of the MonatJahr_ID field.';
                }
            }
        }
    }
}
```

```
}  
field(Jahr; Rec.Jahr)  
{  
    Tooltip = 'Specifies the value of the Jahr field.';  
}  
field(Monatname; Rec.Monatname)  
{  
    Tooltip = 'Specifies the value of the Monatname field.';  
}  
field(Monatsanfang; Rec.Monatsanfang)  
{  
    Tooltip = 'Specifies the value of the Monatsanfang field.';  
}  
field(Monatsende; Rec.Monatsende)  
{  
    Tooltip = 'Specifies the value of the Monatsende field.';  
}  
field(Quartal; Rec.Quartal)  
{  
    Tooltip = 'Specifies the value of the Quartal field.';  
}  
}  
}  
actions  
{  
    area(Navigation)  
    {  
        action(PULL_Data)  
        {  
            ApplicationArea = ALL;  
        }  
    }  
}
```

```
trigger OnAction()
begin
    Direction:='>'; // aufrunden // https://learn.microsoft.com/en-us/dynamics365/business-central/dev-
itpro/developer/methods-auto/system/system-round-method
    Precision:=1; //ohne Nachkommerzahl
    MonthText:=recZeit."Period Type"::Month;
    recZeit.SetRange("Period Type", monthtext);
    recZeit.SetRange("Period Start", 20221001D, 20230101D); // von 01.10.2022 bis 01.01.2023
    if recZeit.FindFirst()then repeat rec.MonatJahr_ID:=FORMAT(recZeit."Period Start", 7, '<Year4>/<month,2>');
        rec.Jahr:=Date2DMY(recZeit."Period Start", 3);
        rec.Monatname:=recZeit."Period Name";
        Rec.Monatsanfang:=recZeit."Period Start";
        Rec.Monatsende:=recZeit."Period End";
        rec.Quartal:=Round((Date2DMY(recZeit."Period Start", 2) / 3), Precision, Direction);
        rec.Insert();
    until recZeit.Next() = 0;
end;
}
action(deleteALL)
{
    ApplicationArea = ALL;

    trigger OnAction()
    begin
        if rec.FindFirst()then rec.DeleteAll();
    end;
}
}
}
var recZeit: Record "date";
```

```
monthtext: option;  
Direction: text;  
Precision: Decimal;  
}
```

Abschnitt 11)

T4_Dim_Wochentag.Table.al

```
table 123456762 T4Dim_Wochentag  
{  
    Caption = 'T4Dim_Wochentag';  
    DataClassification = ToBeClassified;  
  
    fields  
    {  
        field(1; Wochentag_ID; Integer)  
        {  
            Caption = 'Wochentag_ID';  
            DataClassification = ToBeClassified;  
        }  
        field(2; Beschreibung; Text[10])  
        {  
            Caption = 'Beschreibung';  
            DataClassification = ToBeClassified;  
        }  
    }  
    keys  
    {  
        key(PK; Wochentag_ID)  
        {  
            Clustered = true;  
        }  
    }  
}
```

```
}  
}  
}
```

Abschnitt 12)

T4_Dim_Wochentag.Page.al

```
page 123456763 T4Dim_Wochentag  
{  
    ApplicationArea = All;  
    Caption = 'T4Dim_Wochentag';  
    PageType = List;  
    SourceTable = T4Dim_Wochentag;  
  
    layout  
    {  
        area(content)  
        {  
            repeater(General)  
            {  
                field(Wochentag_ID; Rec.Wochentag_ID)  
                {  
                    Tooltip = 'Specifies the value of the Wochentag_ID field.';  
                }  
                field(Beschreibung; Rec.Beschreibung)  
                {  
                    Tooltip = 'Specifies the value of the Beschreibung field.';  
                }  
            }  
        }  
    }  
}  
actions
```

```
{
  area(Navigation)
  {
    action(Pull_Data)
    {
      ApplicationArea = ALL;

      trigger OnAction()
      begin
        dayOption:=recDate."Period Type"::"date";
        recDate.setrange("Period Type", dayOption);
        for n:=1 to 5 do begin
          Rec.Wochentag_ID:=n;
          recDate.setrange("Period No.", n);
          if recDate.FindSet()then Rec.Beschreibung:=recDate."Period Name";
          rec.insert();
        end end;
      end
    }
  }
}
var recDate: Record "date";
dayOption: Option;
n: integer;
}
```

Querys und Codeunits

Abschnitt 13)

T4g_Kreuzprodukt.Query.al

```
query 123456760 T4g_Kreuzprodukt
{
```

```
Caption = 'T4G_PSFT';
QueryType = Normal;

elements
{
  dataitem(DataItem1;
    T4Dim_Zeit)
  {
    SqlJoinType = crossJoin;

    column(MonatJahr_ID;
      MonatJahr_ID)
    {
    }
    dataitem(DataItem2;
      T4Dim_Wochentag)
    {
      SqlJoinType = crossJoin;

      column(Wochentag_ID;
        Wochentag_ID)
      {
      }
      dataitem(DataItem3;
        T4Dim_Mitarbeiter)
      {
        SqlJoinType = crossJoin;

        column(EmployeeNr;
          EmployeeNr)
        {

```



```
}  
dataitem(DataItem4;  
T4Dim_Abwesenheitsgrund)  
{  
SqlJoinType = crossJoin;  
  
column(Abwesenheitscode;  
Abwesenheitscode)  
{  
}  
} //DataItem4  
} //DataItem3  
} //DataItem2  
} // DataItem1  
} // elements  
}
```

Abschnitt 14)

T4CuFuerQuery.Codeunit.al

```
codeunit 123456760 T4CuFuerQuery  
{  
    trigger OnRun()  
    begin  
        if GKreuzProdukt.Open() then begin  
            while GKreuzProdukt.Read() do begin  
                PSFT.Zeit_ID:=GKreuzProdukt.MonatJahr_ID;  
                PSFT.Wochentag_ID:=GKreuzProdukt.Wochentag_ID;  
                PSFT.Mitarbeiter_ID:=GKreuzProdukt.EmployeeNr;  
                PSFT.Abwesenheitsgrund:=GKreuzProdukt.Abwesenheitscode;  
                if dimMitarbeiter.Get(GKreuzProdukt.EmployeeNr) then PSFT.Abteilungscode:=dimMitarbeiter.Abteilungscode;  
            end  
        end  
    end  
}
```

```
        PSFT.Insert();  
    end;  
    GKreuzProdukt.Close();  
end;  
end;  
var GKreuzProdukt: Query T4g_Kreuzprodukt;  
PSFT: Record T4PSFT;  
dimMitarbeiter: Record T4Dim_Mitarbeiter;  
}
```

PSFT

Abschnitt 15)

T4PSFT.Table.al

```
table 123456766 T4PSFT  
{  
    Caption = 'PSFT';  
    DataClassification = ToBeClassified;  
  
    fields  
    {  
        field(1; Zeit_ID; Code[7])  
        {  
            Caption = 'Zeit_ID';  
            DataClassification = ToBeClassified;  
        }  
        field(2; Wochentag_ID; Integer)  
        {  
            Caption = 'Wochentag_ID';  
            DataClassification = ToBeClassified;  
        }  
    }  
}
```

```
field(3; Mitarbeiter_ID; Code[5])
{
    Caption = 'Mitarbeiter_ID';
    DataClassification = ToBeClassified;
}
field(4; Abwesenheitsgrund; Code[6])
{
    Caption = 'Abwesenheitsgrund';
    DataClassification = ToBeClassified;
}
field(5; Sollarbeitstage; Integer)
{
    Caption = 'Sollarbeitstage';
    DataClassification = ToBeClassified;
}
field(6; Anzahl_Abwesenheitstage; Integer)
{
    Caption = 'Anzahl der Abwesenheistage';
    DataClassification = ToBeClassified;
}
field(7; Quote; Decimal)
{
    Caption = 'Quote';
    DataClassification = ToBeClassified;
}
field(8; Abteilungscode; code[8])
{
    Caption = 'Abteilungscode';
    DataClassification = ToBeClassified;
}
}
```

```
keys
{
    key(PK; Zeit_ID, Wochentag_ID, Mitarbeiter_ID, Abwesenheitsgrund)
    {
        Clustered = true;
    }
}
```

Abschnitt 16)

T4PSFT.Page.al

```
page 1234567 T4PSFT
{
    ApplicationArea = All;
    Caption = 'T4PSFT';
    PageType = ListPart;
    SourceTable = T4PSFT;

    layout
    {
        area(content)
        {
            repeater(General)
            {
                field(Zeit_ID; Rec.Zeit_ID)
                {
                    Tooltip = 'Specifies the value of the Zeit_ID field.';
                }
                field(Wochentag_ID; Rec.Wochentag_ID)
                {
```

```
        Tooltip = 'Specifies the value of the Wochentag_ID field.';
    }
    field(Mitarbeiter_ID; Rec.Mitarbeiter_ID)
    {
        Tooltip = 'Specifies the value of the Mitarbeiter_ID field.';
    }
    field(Abwesenheitsgrund; Rec.Abwesenheitsgrund)
    {
        Tooltip = 'Specifies the value of the Abwesenheitsgrund field.';
    }
    field(Anzahl_Abwesenheistage; Rec.Anzahl_Abwesenheitstage)
    {
        Tooltip = 'Specifies the value of the Anzahl_Abwesenheistage field.';
    }
    field(Sollarbeitstage; Rec.Sollarbeitstage)
    {
        Tooltip = 'Specifies the value of the Sollarbeitstage field.';
    }
    field(Quote; Rec.quote)
    {
        Tooltip = 'quote';
    }
    field(Abteilungscode; Rec.Abteilungscode)
    {
    }
}
}
}
```

Zieltabelle (Berichte)

Abschnitt 17)

T4Zieltabelle.Page.al

```
page 123456765 T4Zieltabelle
{
    ApplicationArea = All;
    Caption = 'Zieltabelle';
    PageType = Card;
    SourceTable = T4PSFT;
    PromotedActionCategories = 'New,Processs,Report,Manage,New Document,PSFTBefüllen,Berichte,DateOperations,TimeKepping,
AbsenceOperations';

    layout
    {
        area(content)
        {
            group(General)
            {
                field(v_abteilung; v_abteilung)
                {
                    Caption = 'Abteilung';
                }
                field(v_Mitarbeiter; v_Mitarbeiter)
                {
                    Caption = 'Mitarbeiter';
                }
                field(v_Abwesenheitsgrund; v_Abwesenheitsgrund)
                {
                    Caption = 'Abwesenheitsgrund';
                }
            }
        }
    }
}
```

```
group(Zieltabelle)
{
    part(T4PSFT; T4PSFT)
    {
    }
}
area(FactBoxes)
{
    part("Mitarbeiter Details"; T4Mitarbeiter_Details)
    {
        ApplicationArea = All;
        Provider = T4PSFT;
        SubPageLink = "No.="field(Mitarbeiter_ID);
    }
}
actions
{
    area(Navigation)
    {
        group(füllePSFT)
        {
            action(PKs_befüllen)
            {
                ApplicationArea = All;
                ToolTip = 'PKS aus Dimisionen befüllen';
                Promoted = true;
                PromotedCategory = Category6;
                PromotedOnly = true;
                PromotedIsBig = true;
            }
        }
    }
}
```

```
trigger OnAction()
begin
    cu.Run();
end;
}
action(Fakten_befüllen)
{
    ApplicationArea = All;
    ToolTip = 'Faktspalten befüllen';
    Promoted = true;
    PromotedCategory = Category6;
    PromotedOnly = true;
    PromotedIsBig = true;

    trigger OnAction()
    begin
        if psft.FindFirst()then repeat recDIMWochentag.get(PSFT.Wochentag_ID);
            recDIM_Zeit.get(psft.Zeit_ID);
            if rec7602.FindFirst()then for dat:=recDIM_Zeit.Monatsanfang to recDIM_Zeit.Monatsende do begin
                if not cu7600.IsNonworkingDay(dat, rec7602) and (Format(dat, 0, '<Weekday Text>') =
recDIMWochentag.Beschreibung)then begin
                    counter:=counter + 1; //zählt die Sollarbeitstage
                    recEinzelabwesenheit.SetRange(Datum, dat);
                    recEinzelabwesenheit.SetRange(employeeNo, psft.Mitarbeiter_ID);
                    recEinzelabwesenheit.SetRange(Abwesenheitsgrund, PSFT.Abwesenheitsgrund);
                    if recEinzelabwesenheit.FindFirst()then begin
                        if recEinzelabwesenheit.istUeberlappend = false then
Anzahl_Abwesenheit:=Anzahl_Abwesenheit + 1
                        else if(recEinzelabwesenheit.istUeberlappend = true) and (PSFT.Abwesenheitsgrund =
'KRANK')then Anzahl_Abwesenheit:=Anzahl_Abwesenheit + 1;
```



```
                end;
            end;
        end;
        PSFT.Sollarbeitstage:=counter;
        PSFT.Anzahl_Abwesenheitstage:=Anzahl_Abwesenheit;
        PSFT.Quote:=PSFT.Anzahl_Abwesenheitstage / PSFT.Sollarbeitstage;
        recEinzelabwesenheit.Reset();
        counter:=0;
        Anzahl_Abwesenheit:=0;
        PSFT.Modify();
    until psft.Next() = 0;
end;
}
action>Delete_Fakten
{
    ApplicationArea = All;
    ToolTip = 'Fakten löschen';
    Promoted = true;
    PromotedCategory = Category6;
    PromotedOnly = true;
    PromotedIsBig = true;

    trigger OnAction()
    begin
        if psft.FindFirst() then repeat PSFT.Anzahl_Abwesenheitstage:=0;
            PSFT.Sollarbeitstage:=0;
            PSFT.Quote:=0;
            PSFT.Modify();
        until PSFT.Next = 0;
    end;
}
```

```
    action(GesamtAbwesenheitstage)
    {
        ApplicationArea = All;
        ToolTip = 'Gesamtanzahl der Abwesenheitstage zählen';
        Promoted = true;
        PromotedCategory = Category6;
        PromotedOnly = true;
        PromotedIsBig = true;

        trigger OnAction()
        begin
            if PSFT.CalcSums(Anzahl_Abwesenheitstage) and PSFT.CalcSums(Sollarbeitstage) then Message('Anzahl der
GesamtAbwesenheitstage: ' + Format(PSFT.Anzahl_Abwesenheitstage) + '\\\' + 'Sollarbeitstage: ' + Format(psft.Sollarbeitstage));
            end;
        }
    action(deleteALL)
    {
        ApplicationArea = All;
        ToolTip = 'Tabelle löschen';
        Promoted = true;
        PromotedCategory = Category6;
        PromotedOnly = true;
        PromotedIsBig = true;

        trigger OnAction()
        begin
            if psft.FindFirst() then PSFT.DeleteAll();
            end;
        }
    }
}
group(Berichte)
```

```
{
  action(wochentage_bericht)
  {
    ApplicationArea = All;
    ToolTip = 'Bericht für Wochentage';
    Promoted = true;
    PromotedCategory = Category7;
    PromotedOnly = true;
    PromotedIsBig = true;

    trigger OnAction()
    begin
      Direction:='>';
      Precision:=0.01;
      if recDIMWochentag.FindFirst()then repeat PSFT.SetRange(Wochentag_ID, recDIMWochentag.Wochentag_ID);
        if Format(v_abteilung) <> 'ALLE' then PSFT.SetRange(Abteilungscode, Format(v_abteilung))
        else if Format(v_Mitarbeiter) <> 'ALLE' THEN PSFT.SetRange(Mitarbeiter_ID, Format(v_Mitarbeiter));
        if Format(v_Abwesenheitsgrund) <> 'KRANK oder URLAUB' THEN PSFT.SetRange(Abwesenheitsgrund,
Format(v_Abwesenheitsgrund));
        if PSFT.CalcSums(Anzahl_Abwesenheitstage) and PSFT.CalcSums(Sollarbeitstage)then begin
          if PSFT.Anzahl_Abwesenheitstage <> 0 then v_quote:=Round(((PSFT.Anzahl_Abwesenheitstage /
PSFT.Sollarbeitstage) * 100), Precision, Direction);
          MyList.add(recDIMWochentag.Beschreibung + '--> ' + Format(PSFT.Anzahl_Abwesenheitstage) +
' --> ' + Format(v_quote) + '%');
        end;
        PSFT.Reset();
        until recDIMWochentag.Next() = 0;
        Output:='Anzahl der Abwesenheitstage an Wochentagen\\ Absolut und Relativ Kennzahl\\ für Mitarbeiter: ' +
format(v_Mitarbeiter) + ' \\Abwesenheitsgrund: ' + Format(v_Abwesenheitsgrund) + '\\';
        foreach e in MyList do output+=e + '\\';
        Message(Output);
      end;
    end;
  end;
}
```

```
        MyList.RemoveRange(1, 5);
    end;
}
action(Monate_bericht)
{
    ApplicationArea = All;
    ToolTip = 'Bericht für Wochentage';
    Promoted = true;
    PromotedCategory = Category7;
    PromotedOnly = true;
    PromotedIsBig = true;

    trigger OnAction()
    begin
        Direction:='>';
        Precision:=0.01;
        if recDIM_Zeit.FindFirst()then repeat PSFT.SetRange(Zeit_ID, recDIM_Zeit.MonatJahr_ID);
            if Format(v_abteilung) <> 'ALLE' then PSFT.SetRange(Abteilungscode, Format(v_abteilung))
            else if Format(v_Mitarbeiter) <> 'ALLE' THEN PSFT.SetRange(Mitarbeiter_ID, Format(v_Mitarbeiter));
            if Format(v_Abwesenheitsgrund) <> 'KRANK oder URLAUB' THEN PSFT.SetRange(Abwesenheitsgrund,
Format(v_Abwesenheitsgrund));
                if PSFT.CalcSums(Anzahl_Abwesenheitstage) and PSFT.CalcSums(Sollarbeitstage)then begin
                    if PSFT.Anzahl_Abwesenheitstage <> 0 then v_quote:=Round(((PSFT.Anzahl_Abwesenheitstage /
PSFT.Sollarbeitstage) * 100), Precision, Direction);
                    MyList.add(recDIM_Zeit.Monatname + '--> ' + Format(PSFT.Anzahl_Abwesenheitstage) + '--> ' +
Format(v_quote) + '%');
                end;
                PSFT.Reset();
                until recDIM_Zeit.Next() = 0;
                Output:='Anzahl der Abwesenheitstage in Monaten \\ für Mitarbeiter: ' + format(v_Mitarbeiter) + '
\\Abwesenheitsgrund: ' + Format(v_Abwesenheitsgrund) + '\\';
```

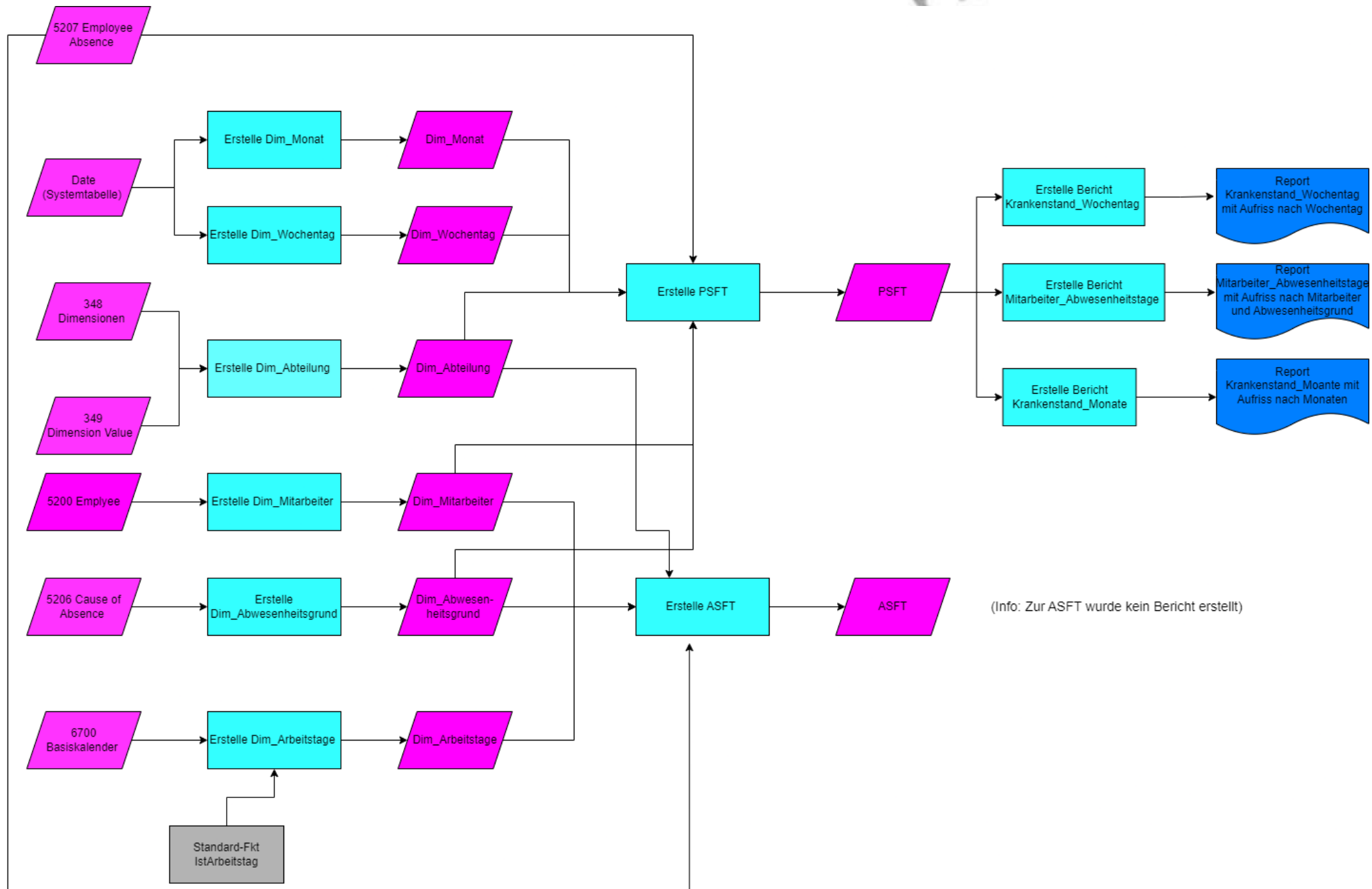
```
        foreach e in MyList do output+=e + '\\';
        Message(Output);
        MyList.RemoveRange(1, 4);
    end;
}
action(Mitarbeiter_Abwesenheitstage)
{
    ApplicationArea = All;
    ToolTip = 'Bericht Mitarbeiter-Abwesenheitstage';
    Promoted = true;
    PromotedCategory = Category7;
    PromotedOnly = true;
    PromotedIsBig = true;

    trigger OnAction()
    begin
        if recDim_MIT.FindFirst() then repeat PSFT.SetRange(Mitarbeiter_ID, recDim_MIT.EmployeeNr);
            if Format(v_Abwesenheitsgrund) <> 'KRANK oder URLAUB' THEN PSFT.SetRange(Abwesenheitsgrund,
Format(v_Abwesenheitsgrund));
            if PSFT.CalcSums(Anzahl_Abwesenheitstage) then MyList.add(recDim_MIT.Vorname + ' ' +
recDim_MIT.Nachname + '--> ' + Format(PSFT.Anzahl_Abwesenheitstage));
            PSFT.Reset();
            until recDim_MIT.Next() = 0;
            Output:='Gesamtanzahl der Abwesenheitstage für jeden Mitarbeiter: ' + ' \\Abwesenheitsgrund: ' +
Format(v_Abwesenheitsgrund) + '\\';
            foreach e in MyList do output+=e + '\\';
            Message(Output);
            MyList.RemoveRange(1, 4);
        end;
    }
}
```

```
    }  
}  
var cu7600: Codeunit 7600;  
dat: date;  
rec7602: Record 7602;  
cu: Codeunit T4CuFuerQuery;  
PSFT: Record T4PSFT;  
recDim_MIT: Record T4Dim_Mitarbeiter;  
recDIM_Zeit: Record T4Dim_Zeit;  
recDIMWochentag: Record T4Dim_Wochentag;  
recDIMAbwesenheitsgrund: Record T4Dim_Abwesenheitsgrund;  
counter: integer;  
Anzahl_Abwesenheit: Integer;  
recEinzelabwesenheit: Record T4EinzelAbwesenheit;  
v_abteilung: option ALLE, PROD, VERKAUF, VERW, ;  
v_Mitarbeiter: Option "ALLE", "M0030", "M0040", "M0050", "M0060";  
v_Abwesenheitsgrund: Option "KRANK oder URLAUB", "KRANK", "URLAUB";  
MyList: List of[Text];  
msgText: Text[30];  
e: text;  
Output: text;  
v_quote: decimal;  
DecimalToRound: Decimal;  
Direction: Text;  
Precision: Decimal;  
}
```

Anhang (alte Sachen):

2023



© Westenberger 2023

© Westenberger 2023