

# DB Kalorienbilanz

**Projekt:** DB Kalorienbilanz

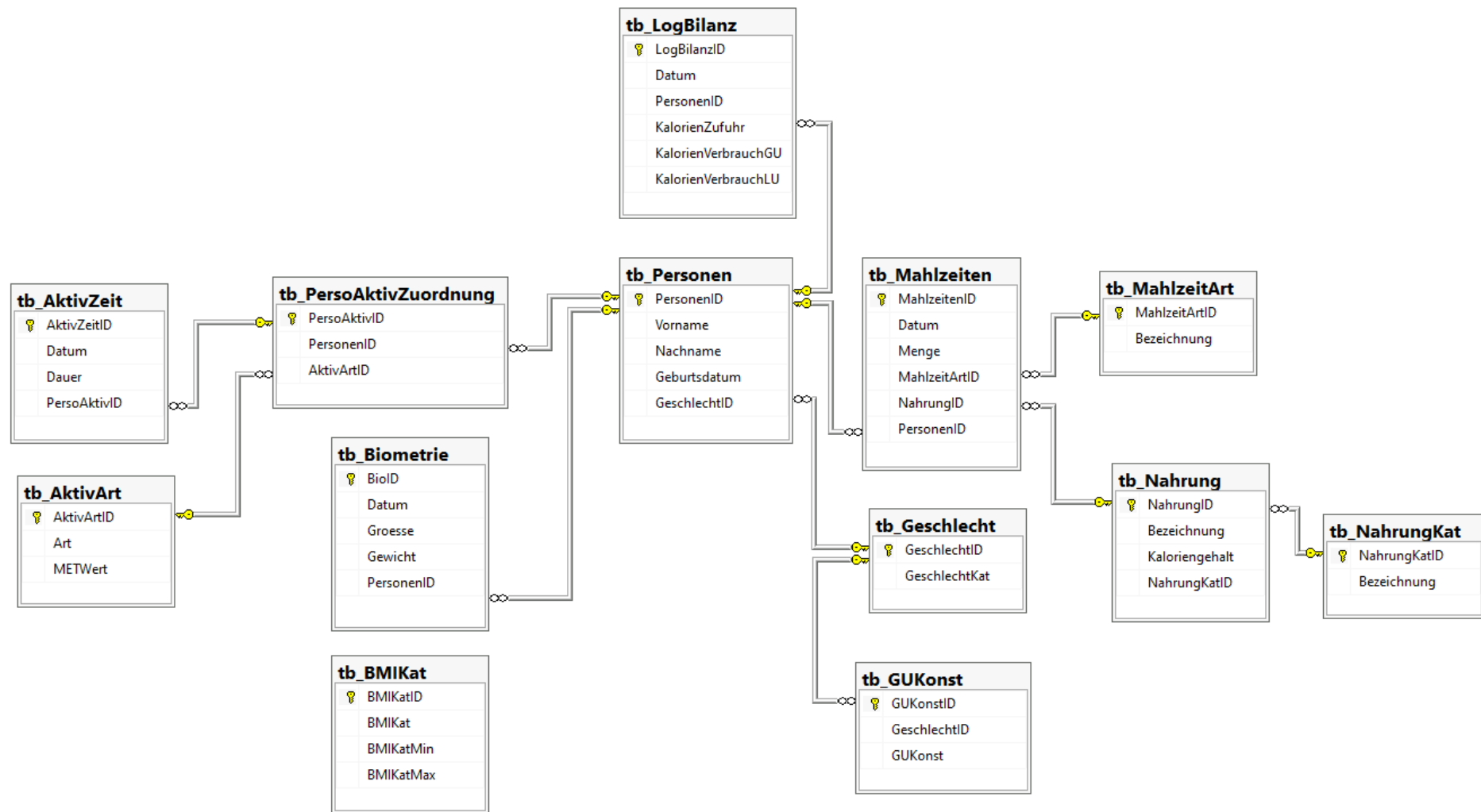
**Projektabgabe:** 14.07.2023

**Teammitglieder:** Valerie Münz, Jan [REDACTED], Marie [REDACTED], Kathy [REDACTED]

## Businesslogik:

Mit der **DB Kalorienbilanz** kann für verschiedene Personen die Kalorienzufuhr und der Kalorienverbrauch im Zeitverlauf erfasst werden.

Die Tabellenstruktur kann dem nachfolgenden Diagramm und der nachfolgenden Tabelle entnommen werden:



| Tabellen und Spalten |                    |         |              |            |   |                                |   |
|----------------------|--------------------|---------|--------------|------------|---|--------------------------------|---|
| Tabelle              | Spalten            | Einheit | Datentyp     | NULL-Werte | Schlüssel/Indizes                           | Standardwert                   | Einschränkungen   |
| tb_AktivArt          | <b>AktivArtID</b>  |         | int          | Nicht NULL | PK_AktivArtID (gruppiert)                   |                                |   |
|                      | Art                |         | nvarchar(50) | Nicht NULL | IX_Art (eindeutig, nicht gruppiert)         |                                |   |
|                      | METWert            |         | float        | Nicht NULL |   |                                | CK_METWert:<br>([METWert]>=(1.5) AND [METWert]<=(20))     |
| tb_AktivZeit         | <b>AktivZeitID</b> |         | int          | Nicht NULL | PK_AktivZeitID (gruppiert)                  |                                |   |
|                      | Datum              |         | date         | Nicht NULL |   | Heutiges Datum:<br>(getdate()) | CK_AktivZeit_Datum:<br>([Datum]<=getdate())               |
|                      | Dauer              | Minuten | int          | Nicht NULL |   |                                | CK_AktivZeit_Dauer:<br>([Dauer]>=(1) AND [Dauer]<=(1440)) |
|                      | PersoAktivID       |         | int          | Nicht NULL | FK_tb_AktivZeit_tb_PersoAktivZuordnung      |                                |   |
| tb_Biometrie         | <b>BioID</b>       |         | int          | Nicht NULL | PK_BioID (gruppiert)                        |                                |   |
|                      | Datum              |         | date         | Nicht NULL | IX_BioDatPerso (eindeutig, nicht gruppiert) | Heutiges Datum:<br>(getdate()) | CK_Biometrie_Datum:<br>([Datum]<=getdate())               |
|                      | Groesse            | cm      | tinyint      | Nicht NULL |   |                                | CK_Biometrie_Groesse:<br>([Groesse]>(0))                  |

|  |                     |    |              |            |   |  |  |
|--|---------------------|----|--------------|------------|---|--|--|
|  | Gewicht             | kg | tinyint      | Nicht NULL |   |  | CK_Biometrie_Gewicht:<br>([Gewicht]>(0)) |
|  | PersonenID          |    | int          | Nicht NULL | FK_tb_Biometrie_tb_Person<br>en<br>IX_BioDatPerso (eindeutig,<br>nicht gruppiert) |  |  |
| tb_BMIKat<br>(Hilfstabelle, bisher nicht<br>genutzt) | <b>BMIKatID</b>     |    | int          | Nicht NULL | PK_BMIKatID (gruppiert)   |  |  |
|  | BMIKat              |    | nvarchar(50) | Nicht NULL | IX_BMIKat (eindeutig, nicht<br>gruppiert)   |  |  |
|  | BMIKatMin           |    | tinyint      | Nicht NULL | IX_BMIKatMin (eindeutig,<br>nicht gruppiert)                                      |  |  |
|  | BMIKatMax           |    | tinyint      | Nicht NULL | IX_BMIKatMax (eindeutig,<br>nicht gruppiert)                                      |  |  |
| tb_Geschlecht  | <b>GeschlechtID</b> |    | int          | Nicht NULL | PK_GeschlechtID (gruppiert)   |  |  |
|  | GeschlechtKat       |    | nvarchar(50) | Nicht NULL | IX_GeschlechtKat (eindeutig,<br>nicht gruppiert)                                  |  |  |
| tb_GUKonst   | <b>GUKonstID</b>    |    | int          | Nicht NULL | PK_GUKonstID (gruppiert)  |  |  |
|  | GeschlechtID        |    | int          | Nicht NULL | IX_GeGUKonst (eindeutig,<br>nicht gruppiert)                                      |  |  |
|  | GUKonst             |    | smallint     | Nicht NULL | IX_GeGUKonst (eindeutig,<br>nicht gruppiert)                                      |  |  |
| tb_LogBilanz   | <b>LogBilanzID</b>  |    | int          | Nicht NULL | PK_LogBilanzID (gruppiert)  |  |  |
|  | Datum               |    | date         | Nicht NULL |   |  |  |

|                |                      |       |              |            |   |                                   |  |
|----------------|----------------------|-------|--------------|------------|---|-----------------------------------|--|
|                | PersonenID           |       | int          | Nicht NULL | FK_tb_LogBilanz_tb_Personen                       |                                   |  |
|                | Kalorienzufuhr       | kcal  | float        | NULL       |   |                                   |  |
|                | KalorienverbrauchGU  | kcal  | float        | NULL       |   |                                   |  |
|                | KalorienverbrauchLU  | kcal  | float        | NULL       |   |                                   |  |
| tb_MahlzeitArt | <b>MahlzeitArtID</b> |       | int          | Nicht NULL | PK_MahlzeitArtID<br>(gruppiert)                   |                                   |  |
|                | Bezeichnung          |       | nvarchar(50) | Nicht NULL | IX_MahlzeitArtBez<br>(eindeutig, nicht gruppiert) |                                   |  |
| tb_Mahlzeiten  | <b>MahlzeitenID</b>  |       | int          | Nicht NULL | PK_MahlzeitenID (gruppiert)                       |                                   |  |
|                | Datum                |       | date         | Nicht NULL |   | Heutiges<br>Datum:<br>(getdate()) | CK_Mahlzeiten_Datum:<br>([Datum]<=getdate()) |
|                | Menge                | g, ml | int          | Nicht NULL |   |                                   | CK_Mahlzeiten_Menge:<br>([Menge]>(0))        |
|                | MahlzeitArtID        |       | int          | Nicht NULL | FK_tb_Mahlzeiten_tb_MahlzeitArt                   |                                   |  |
|                | NahrungID            |       | int          | Nicht NULL | FK_tb_Mahlzeiten_tb_Nahrung                       |                                   |  |
|                | PersonenID           |       | int          | Nicht NULL | FK_tb_Mahlzeiten_tb_Personen                      |                                   |  |

|                        |                     |                            |              |            |  |  |   |
|------------------------|---------------------|----------------------------|--------------|------------|--|--|---|
| tb_Nahrung             | <b>NahrungID</b>    |                            | int          | Nicht NULL | PK_NahrungID (gruppiert)   |  |   |
|                        | Bezeichnung         |                            | nvarchar(50) | Nicht NULL | IX_NahrungBez (eindeutig, nicht gruppiert)   |  |   |
|                        | Kaloriengehalt      | kcal pro 100 g bzw. 100 ml | int          | Nicht NULL |  |  | CK_Nahrung_Kaloriengehalt: ([Kaloriengehalt]>(0)) |
|                        | NahrungKatID        |                            | int          | Nicht NULL | FK_tb_Nahrung_tb_NahrungKat  |  |   |
| tb_NahrungKat          | <b>NahrungKatID</b> |                            | int          | Nicht NULL | PK_NahrungKatID (gruppiert)  |  |   |
|                        | Bezeichnung         |                            | nvarchar(50) | Nicht NULL | IX_NahrungKatBez (eindeutig, nicht gruppiert)  |  |   |
| tb_PersoAktivZuordnung | <b>PersoAktivID</b> |                            | int          | Nicht NULL | PK_PersoAktivID (gruppiert)  |  |   |
|                        | PersonenID          |                            | int          | Nicht NULL | FK_tb_PersoAktivZuordnung_tb_Personen<br>IX_PersoAktivZuordnung (eindeutig, nicht gruppiert) |  |   |
|                        | AktivArtID          |                            | int          | Nicht NULL | FK_tb_PersoAktivZuordnung_tb_AktivArt<br>IX_PersoAktivZuordnung (eindeutig, nicht gruppiert) |  |   |
| tb_Personen            | <b>PersonenID</b>   |                            | int          | Nicht NULL | PK_PersonenID (gruppiert)  |  |   |

|  |              |  |              |            |                                  |  |   |
|--|--------------|--|--------------|------------|----------------------------------|--|---|
|  | Vorname      |  | nvarchar(50) | Nicht NULL |                                  |  |   |
|  | Nachname     |  | nvarchar(50) | Nicht NULL |                                  |  |   |
|  | Geburtsdatum |  | date         | Nicht NULL |                                  |  | CK_Personen_Geburtsdatum:<br>([Geburtsdatum]<getdate()) |
|  | GeschlechtID |  | int          | Nicht NULL | FK_tb_Personen_tb_<br>Geschlecht |  |   |

### **001-DB-Create-Scripts**

- **CREATE DATABASE:** 001-01-CREATE-DB\_Kalorienbilanz.sql
  - This script creates the data base *Kalorienbilanz*.
- **CREATE TABLE:** 001-02-CREATE-tb\_MahlzeitArt.sql
  - This script creates the table *MahlzeitArt* with the columns *MahlzeitArtID* as primary key and *Bezeichnung*.
- **CREATE NONCLUSTERED INDEX:** 001-03-CREATE-IX\_PersoAktivZuordnung.sql
  - This script creates the unique nonclustered index [*IX\_PersoAktivZuordnung*] containing the columns *PersonenID* and *AktivArtID* for the table *PersoAktivZuordnung*.
- **ADD CONSTRAINT FOREIGN KEY:** 001-04-CREATE-FK\_tb\_PersoAktivZuordnung\_tb\_Personen.sql
  - This script defines the column *PersonenID* as a foreign key in the table *PersoAktivZuordnung*.
- **ADD CONSTRAINT CHECK:** 001-05-CK\_Nahrung\_Kaloriengehalt.sql
  - This script adds the constraint check [*CK\_Nahrung\_Kaloriengehalt*] to the table *Nahrung*, which determines that the inserted value in the column *Kaloriengehalt* must be greater than 0.



## 002-Abfragen und View

| Abfragen und Sichten        |  |   |
|-----------------------------|--|---|
| Name                        | Ausgabe  | Skript-Datei                                    |
| View_Personen_weibl         | <p>This view shows all female persons from the table <i>Personen</i>, ordered by their last and first name in ascending order.</p> <p><b>tb_Personen:</b> Nachname (ASC, 1), Vorname (ASC, 2), Geschlecht = 1</p>  | 002-01-Generate_View_Personen_weibl.sql         |
| View_PersoAktivArt          | <p>This view shows all persons from the table <i>Personen</i>, ordered by their last and first name in ascending order, for which an activity was recorded, and the respective activities in ascending order with their corresponding MET values from the table <i>AktivArt</i>.</p> <p><b>tb_Personen:</b> Nachname (ASC, 1), Vorname (ASC, 2) [INNER JOIN]<br/> <b>tb_PersoAktivZuordnung:</b> PersoAktivID [INNER JOIN] <b>tb_AktivArt:</b> Art (ASC, 3), METWert</p> | 002-02-Generate_View_PersoAktivArt.sql          |
| View_PersonAktivZeitDatum   | <p>This view shows the total duration of all activities, which were recorded for the person with <i>PersonenID</i> 2 in the <i>AktivZeit</i> table on 9 July 2023.</p> <p><b>tb_Personen:</b> PersonenID, Nachname, Vorname [INNER JOIN]<br/> <b>tb_PersoAktivZuordnung</b> [INNER JOIN] <b>tb_AktivZeit:</b> Datum</p> <p>GROUP BY: Datum, PersonenID, Nachname, Vorname; SUM: Dauer (SummeZeit); HAVING: Datum = 09.07.2023, PersonenID = 2</p>                        | 002-03-Generate_View_PersonAktivZeitDatum.sql   |
| View_PersonenLogBilanz      | <p>This view shows the calorie intake and expenditure from <i>tb_LogBilanz</i> for all persons.</p> <p><b>tb_Personen:</b> PersonenID (ASC, 1), Nachname, Vorname [LEFT JOIN]<br/> <b>tb_LogBilanz:</b> Datum (DESC, 2), Kalorienzufuhr, KalorienverbrauchGU, KalorienverbrauchLU</p>  | 002-04-Generate_View_PersonenLogBilanz.sql      |
| View_PersonenLogBilanzSumme | <p>This view shows the total calorie intake and expenditure per day from <i>tb_LogBilanz</i> for all persons.</p> <p><b>tb_Personen:</b> PersonenID (ASC, 1), Nachname, Vorname [LEFT JOIN]<br/> <b>tb_LogBilanz:</b> Datum (DESC, 2), Kalorienzufuhr, KalorienverbrauchGU, KalorienverbrauchLU</p> <p>GROUP BY: PersonenID, Nachname, Vorname, Datum; SUM: Kalorienzufuhr, KalorienverbrauchGU, KalorienverbrauchLU</p>   | 002-05-Generate_View_PersonenLogBilanzSumme.sql |

### 003-StoredFunctions

| Funktionen                    |                       |   |  |
|-------------------------------|-----------------------|---|--|
| Name                          | Variablen             | Beschreibung  | Skript-Datei   |
| <b>Tabellenwertfunktionen</b> |                       |   |  |
| tf_PersonAktivZeitDatum       | @Datum<br>@PersonenID | The table valued function <i>[tf_PersonAktivZeitDatum]</i> receives a date and the ID of a person as input and returns the given date, the ID, the last name and the first name of the given person as well as the sum of all activity durations recorded for the given person at the given date. | 003-01-01-CREATE_tf_PersonAktivZeitDatum.sql<br><br>003-01-01-TEST_tf_PersonAktivZeitDatum.sql             |
| tf_GrundumsatzParameter       | @PersonenID           | The table valued function <i>[tf_GrundumsatzParameter]</i> receives the ID of a person as input and returns all parameters which are needed for the computation of the basal metabolic rate of the given person.  | 003-01-02-CREATE_tf_GrundumsatzParameter.sql   |
| tf_LeistungsumsatzParameter   | @PersonenID<br>@Datum | The table valued function <i>[tf_LeistungsumsatzParameter]</i> receives the ID of a person and a date as input and returns the respective values of all parameters which are needed for the computation of the performance metabolic rate of the given person at the given date.                  | 003-01-03-CREATE_tf_LeistungsumsatzParameter.sql   |
| <b>Skalarwertfunktionen</b>   |                       |   |  |
| sf_Zeitstempel                | /                     | The scalar valued function <i>[sf_Zeitstempel]</i> returns the current time stamp in the following format: yyyyMMdd-HHmssfff.   | 003-02-01-CREATE_sf_Zeitstempel.sql  |
| sf_Aktivzeit_Sum_Person_Datum | @Datum<br>@PersonenID | The scalar valued function <i>[sf_Aktivzeit_Sum_Person_Datum]</i> receives the ID of a person and a date as input and returns the sum of all activity durations of the given person at the given date.<br><br><b>tb_Personen [INNER JOIN] tb_PersoAktivZuordnung [INNER JOIN] tb_AktivZeit</b>    | 003-02-02-CREATE_sf_Aktivzeit_Sum_Person_Datum.sql<br><br>003-02-02-TEST_sf_Aktivzeit_Sum_Person_Datum.sql |

|                       |  |   |  |
|-----------------------|--|---|--|
| sf_GetAge             | @GebDat                                    | The scalar valued function <i>[sf_GetAge]</i> receives a date of birth as input and returns the age.  | 003-02-03-CREATE_sf_GetAge.sql             |
| sf_GetGrundumsatz     | @Gewicht<br>@Groesse<br>@Alter<br>@GUKonst | The scalar valued function <i>[sf_GetGrundumsatz]</i> receives the weight, the height, the age and a constant (male: 5, female: -161) of a person as input and returns the basal metabolic rate of this person.<br><br>Formel: Grundumsatz = (10 * Gewicht) + (6.25 * Größe) - (5 * Alter) + Geschlechtskonstante                     | 003-02-04-CREATE_sf_GetGrundumsatz.sql     |
| sf_GetLeistungsumsatz | @METWert<br>@Gewicht<br>@Dauer             | The scalar valued function <i>[sf_GetLeistungsumsatz]</i> receives a MET value of an activity, the weight of a person and the duration of the respective activity as input and returns the performance metabolic rate of the selected person for the given activity.<br><br>Leistungsumsatz = (MET-Wert * 3.5 * Gewicht * Dauer)/200; | 003-02-05-CREATE_sf_GetLeistungsumsatz.sql |
| sf_GetAktivArtID      | @PersoAktivID                              | The scalar valued function <i>[sf_GetAktivArtID]</i> receives a PersoAktivID as input and returns the respective AktivArtID from the PersoAktivZuordnung table.   | 003-02-06-CREATE_sf_GetAktivArtID.sql      |
| sf_GetMETWert         | @AktivArtID                                | The scalar valued function <i>[sf_GetMETWert]</i> receives an AktivArtID as input and returns the respective MET value from the AktivArt table.   | 003-02-07-CREATE_sf_GetMETWert.sql         |
| sf_GetPersonenID      | @PersoAktivID                              | The scalar valued function <i>[sf_GetPersonenID]</i> receives a PersoAktivID as input and returns the respective PersonenID from the PersoAktivZuordnung table.   | 003-02-08-CREATE_sf_GetPersonenID.sql      |
| sf_GetGewicht         | @PersonenID                                | The scalar valued function <i>[sf_GetGewicht]</i> receives a PersonenID as input and returns the weight of the selected person from the Biometrie table.  | 003-02-09-CREATE_sf_GetGewicht.sql         |
| sf_GetGroesse         | @PersonenID                                | The scalar valued function <i>[sf_GetGroesse]</i> receives a PersonenID as input and returns the height of the selected person from the Biometrie table.  | 003-02-10-CREATE_sf_GetGroesse.sql         |

|                      |  |   |   |
|----------------------|--|---|---|
| sf_GetGebDat         | @PersonenID  | The scalar valued function <i>[sf_GetGebDat]</i> receives a PersonenID as input and returns the date of birth of the selected person from the Personen table.   | 003-02-11-CREATE_sf_GetGebDat.sql         |
| sf_GetGeschlechtID   | @PersonenID  | The scalar valued function <i>[sf_GetGeschlechtID]</i> receives a PersonenID as input and returns the GeschlechtID of the selected person from the Personen table.  | 003-02-12-CREATE_sf_GetGeschlechtID.sql   |
| sf_GetGUKonst        | @GeschlechtID                                      | The scalar valued function <i>[sf_GetGUKonst]</i> receives a GeschlechtID as input and returns the GUKonst from the GUKonst table.  | 003-02-13-CREATE_sf_GetGUKonst.sql        |
| sf_GetKaloriengehalt | @NahrungID   | The scalar valued function <i>[sf_GetKaloriengehalt]</i> receives a NahrungID as input and returns the respective calorie value from the GUKonst table.   | 003-02-14-CREATE_sf_GetKaloriengehalt.sql |
| sf_GetKalorienzufuhr | @Menge<br>@Kaloriengehalt                          | The scalar valued function <i>[sf_GetKalorienzufuhr]</i> receives a quantity per 100 g or ml and a calorie value as input and returns the calorie intake.   | 003-02-15-CREATE_sf_GetKalorienzufuhr.sql |
| sf_IstNeuePerson     | @Vorname<br>@Nachname<br>@GebDat,<br>@GeschlechtID | The scalar valued function <i>[sf_IstNeuePerson]</i> receives the first name, the last name, the date of birth and the sex of a person and checks if the combination of the given values already exists. If the combination does not exist it returns 1, otherwise 0. | 003-02-16-CREATE_sf_IstNeuePerson.sql     |
| sf_GetMaxPersonenID  | /  | The scalar valued function <i>[sf_GetMaxPersonenID]</i> returns the highest PersonenID from the Personen table.   | 003-02-17-CREATE_sf_GetMaxPersonenID.sql  |

#### 004-StoredProcedures

| Prozeduren                               |  |  |
|--|--|--|
| Name                                     | Beschreibung   | Skript-Datei   |
| sp_Backup_Kalorienbilanz_mit_Zeitstempel | The procedure <i>[sp_Backup_Kalorienbilanz_mit_Zeitstempel]</i> backs up the data base and automatically generates the file name based on the time stamp.  | 004-01-<br>CREATE_sp_Backup_Kalorienbilanz_mit_Zeitstempel.sql<br><br>004-01-EXEC_sp_Backup_Kalorienbilanz_mit_Zeitstempel.sql |
| sp_AddPersonMitBiometrie                 | The procedure <i>[sp_AddPersonMitBiometrie]</i> checks if the combination of a specific first name, a specific last name, a specific date of birth and a specific sex exists. If the combination already exists it throws an error message. Otherwise it inserts values in the Personen and the Biometrie table. | 004-02-CREATE_sp_AddPersonMitBiometrie.sql<br><br>004-02-TEST_sp_AddPersonMitBiometrie.sql                                     |

#### 005-DML-Trigger

| Trigger                    |              |   |  |  |
|----------------------------|--------------|---|--|--|
| Name                       | Tabelle      | Beschreibung  | Skript-Datei   | Hinweise                               |
| tr_AktivZeit24h_INSERT     | tb_AktivZeit | The trigger <i>[tr_AktivZeit24h_INSERT]</i> prevents a row from being inserted in the AktivZeit table when the total activity duration within one day would exceed 24 hours (1440 minutes). | 005-01-CREATE-tr_AktivZeit24h_INSERT.sql<br><br>005-01-TEST_tr_AktivZeit24h_INSERT.sql |  |
| tr_AktivZeit24h_UPDATE     | tb_AktivZeit | The trigger <i>[tr_AktivZeit24h_UPDATE]</i> prevents a row from being updated in the AktivZeit table when the total activity duration within one day would exceed 24 hours (1440 minutes).  | 005-02-CREATE-tr_AktivZeit24h_UPDATE.sql   |  |
| tr_AktivZeitKaloLog_INSERT | tb_AktivZeit | The trigger <i>[tr_AktivZeitKaloLog_INSERT]</i> is automatically invoked as soon as new values are inserted into the AktivZeit table and logs   | 005-03-CREATE-tr_AktivZeitKaloLog_INSERT.sql   | Der Benutzer darf nicht mehrere Zeilen |

|                            |               |  |  |   |
|----------------------------|---------------|--|--|---|
|                            |               | the respective calory expenditure in the LogBilanz table. If the combination of a specific PersonenID and a specific date occurs for the first time, the basal metabolic rate as well as the performance metabolic rate are logged. Otherwise only the performance metabolic rate is logged. |  | auf einmal hinzufügen.  |
| tr_AktivZeitKaloLog_DELETE | tb_AktivZeit  | The trigger <i>[tr_AktivZeitKaloLog_DELETE]</i> is automatically invoked as soon as values are deleted from the AktivZeit table and logs the respective removal of calory expenditure in the LogBilanz table.  | 005-04-CREATE-tr_AktivZeitKaloLog_DELETE.sql | Der Benutzer darf nicht mit einem Skript mehrere Zeilen auf einmal löschen. |
| tr_MahlzeitKaloLog_INSERT  | tb_Mahlzeiten | The trigger <i>[tr_MahlzeitKaloLog_INSERT]</i> is automatically invoked as soon as new values are inserted into the Mahlzeiten table and logs the respective calory intake in the LogBilanz table.   | 005-05-CREATE-tr_MahlzeitKaloLog_INSERT.sql  |   |
| tr_MahlzeitKaloLog_DELETE  | tb_Mahlzeiten | The trigger <i>[tr_MahlzeitKaloLog_DELETE]</i> is automatically invoked as soon as new values are deleted from the Mahlzeiten table and logs the respective removal of calory intake in the LogBilanz table.   | 005-06-CREATE-tr_MahlzeitKaloLog_DELETE.sql  |   |

## **006-LOGN-USER-ROLE**

- **Login auf Serverebene:** 006-01-CREATE\_LOGIN\_KalorienbilanzUser.sql
  - This script creates a login for the SQL server using SQL authentication (Anmeldename: KalorienbilanzUser, Passwort: 1234). In this case usernames and passwords are stored in encrypted system tables within the SQL server.
- **Login/User auf Datenbankebene:** 006-02-CREATE\_USER\_KalorienbilanzBenutzer.sql
  - This script creates a database user for the DB Kalorienbilanz with specific rights regarding specific database objects. A database user must be created in order for a login (in this case: login via SQL authentication) to have access to a database.

## **007-BackUp**

- **Backup\_Kalorienbilanz:** 007-01-Script\_Autobackup.sql
  - automatisches Backup: stündlich zwischen 7:00 und 16:05 Uhr, im Zeitraum vom 07.07.2023 bis zum 14.07.2023
- **Masterskript:** 007-02-Masterscript.sql
- **Finales Backup:** Kalorienbilanz-20230713-200231030.bak