MyNetDiary Food Dataset Specification v2.1

This document specifies exact format for MyNetDiary dataset. General information can be found at http://www.mynetdiary.com/db

Overview

There are several tables storing information for each food. The tables are normalized for easy export and loading into other databases. For each food there will be one record in the **Food** table, **multiple FoodWeight** records with available food serving information and **multiple FoodNutrientValue** records provided by food vendor.
You may also want to review MyNetDiary_Food_Samples.xls containing variety of sample values for all fields specified below.

1. Food set

Food header information

Field name	**Type**	**Description**
foodId	int	primary key, food identifier
foodDesc	varchar(255)	Food description, includes additional words from foodGroupDesc
boost	double	>1, popularity rank based on usage in MyNetDiary community
imageId	int	An identifier of one of several hundred food icons used by MyNetDiary to create food lists. The data extract includes an archive with food icons. Every icon's filename contains imageId.
mainUPC	varchar(14)	UPC code for packaged food
foodGroupId	mediumint	refers to FoodGroup.foodId

2. FoodWeight set

Food serving information

Field name	**Type**	**Description**
foodId	int	primary key, refers to Food.foodId
sortOrder	smallint	primary key, serving identifier within the food
msreDesc	varchar(255)	a description of food portion, usually, the amount consumable within a single meal

amount	double	how many portions defined by msreDesc are usually served or displayed at food label
gmWgt	double	How many grams are in serving. Is gmWgt is null, then food vendor published "gramless" food label with some msreDesc and nutrient values defined without specifying grams. *Gramless food has 1 serving with gmWgt null*.

Note on Foods without FoodWeights

Several hundred Foods in MyNetDiary dataset do not have FoodWeight rows at all. FoodNutrientValue table contains nutrient values per 100g for such foods. MyNetDiary application code adds FoodWeights on the fly with the following values:

msreDesc ='grams', amount =100, gmWgt=100, sortOrder=0

msreDesc ='oz', amount =1, gmWgt=28.3495, sortOrder=1

This logic is specific to our application serving English and Metric users based on their preferences. We did not add these rows to DB because your application may have different logic (e.g. you do not need 100g weight suggestion if you serve US only). We suggest that you add FoodWeight rows appropriate for your users in the same way as we do in our applications.

Note on Foods with volume servings

Some of the foods may have one or more volume servings in the DB:

- ml
- fl oz
- teaspoon
- tablespoon
- cup

While loading food from DB, MyNetDiary application detects existing volume servings and adds remaining servings from the list above. The nutrients in added servings are recalculated with simple proportion: e.g. if 237ml of food contains 130 calories, then a tablespoon (14.8ml) contains about 8 calories.

We did not add volume servings to the DB because fl oz and cups are US-specific, while European user may prefer ml. Your application can implement the same logic displaying complete set of volume servings.

3. FoodNutrientValue set

Nutrient content of food

Field name	**Type**	**Description**
foodId	int	primary key, refers to Food.foodId

nutrNo	smallint	primary key, refers to Nutrient.nutrNo
nutrVal	double	When food is gramless , then nutrVal contains absolute nutrient value in food's single serving, otherwise, for food with gram servings (or no servings at all), nutrVal contains density of nutrient per 100g of food. Please notice that some foods do not have any servings defined in FoodWeight table.

4. FoodGroup set

Food tree for hierarchical display and processing of food information. Includes USDA groups and additional vendor-specific groups found under Brands group.

Field name	**Type**	**Description**
foodGroupId	mediumint	primary key, root record has foodGrooupId=0
foodGroupDesc	varchar(255)	Food group description
masterGroupId	mediumint	Refers to foodGrooupId for parent group, is null only for root record.

5. Nutrient set

Note: Nutrient definitions are inherited from USDA data set. See the USDA site for details.

Field name	**Type**	**Description**
nutrNo	smallint	primary key
nutrDesc	varchar(100)	Nutrient name
units	varchar(20)	Units of measure

nutrNo	**nutrDesc**	**units**

208	Calories	kcal
204	Total Fat	g
605	Trans Fat	g
606	Saturated Fat	g
646	Polyunsaturated Fat	g
645	Monounsaturated Fat	g
601	Cholesterol	mg
307	Sodium	mg
205	Total Carbohydrates	g
291	Dietary Fiber	g
269	Sugars	g
203	Protein	g
318	Vitamin A	iu
401	Vitamin C	mg
301	Calcium(Ca)	mg
303	Iron(Fe)	mg
415	Vitamin B-6	mg
418	Vitamin B-12	mcg
324	Vitamin D	iu
323	Vitamin E	mg
430	Vitamin K	mcg
304	Magnesium(Mg)	mg
305	Phosphorus(P)	mg
306	Potassium(K)	mg
309	Zinc(Zn)	mg
312	Copper(Cu)	mg
315	Manganese(Mn)	mg
317	Selenium(Se)	mcg
404	Thiamin	mg
405	Riboflavin	mg
406	Niacin	mg
410	Pantothenic Acid	mg
417	Total Folate	mcg
221	Alcohol Ethyl	g
262	Caffeine	mg
209	Starch	g

255	Water	g
210	Sucrose	g
211	Glucose(Dextrose)	g
212	Fructose	g
213	Lactose	g
214	Maltose	g
321	Carotene Beta	mcg
322	Carotene Alpha	mcg
700	Sugar Alcohols	g

5. Data Types

Type name	**Min value**	**Max value**	**Notes**
int	-2147483648	2147483647	Integer number
mediumint	-8388608	8388607	Medium integer number
smallint	-32768	32767	Small integer number
double	4.9e-324	1.7e+308	up to 10-digits after decimal point
varchar(N)	not applicable	not applicable	variable length string, where N is max length

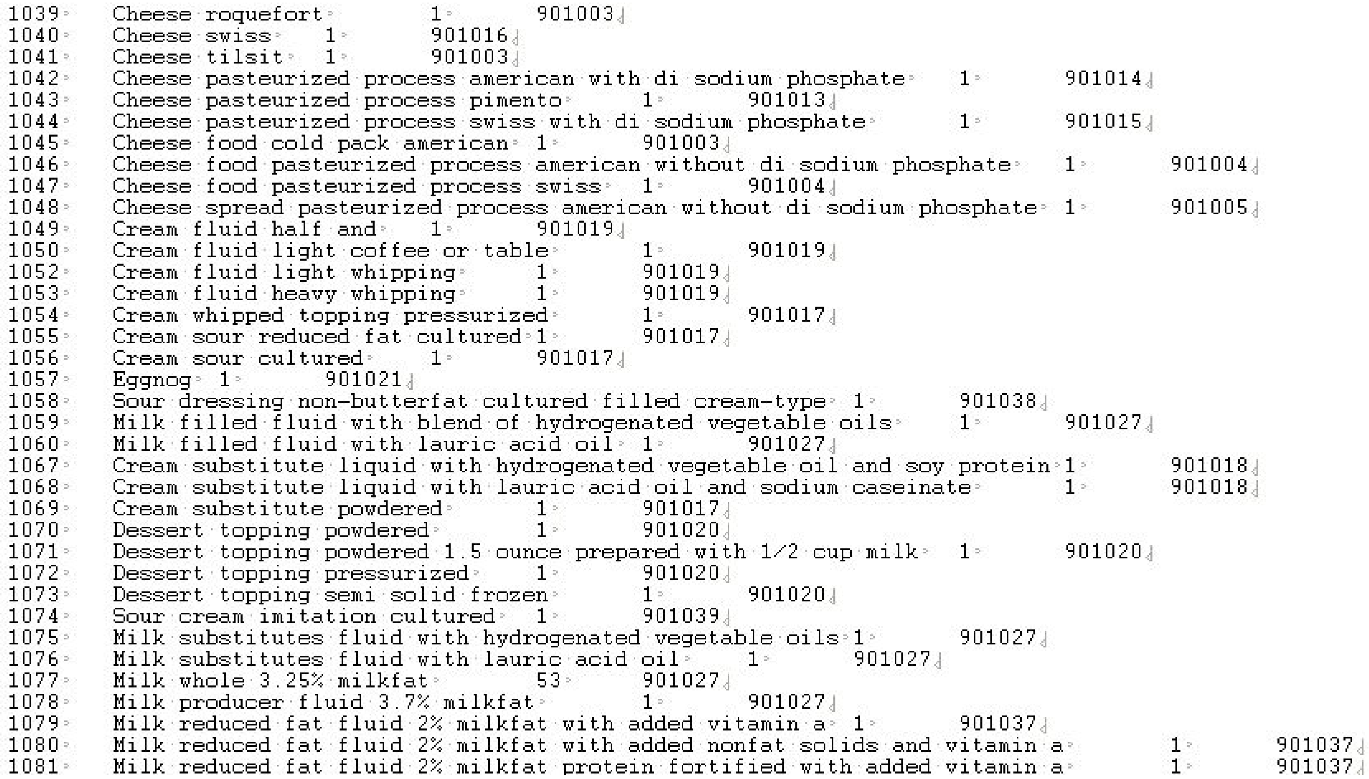
6. Flat file format

1. Every dataset is represented by a separate .txt file.
2. Every line in a file represents a single element from the dataset.
3. Fields in the line are delimited with tab character.
4. Null value is encoded as \N string (2 characters: slash and uppercase N).

Below is a fragment of Food set with

→ symbol representing tab character and

↵ symbol representing new line delimiter:



```
1039    Cheese roquefort        1       901003
1040    Cheese swiss    1       901016
1041    Cheese tilsit   1       901003
1042    Cheese pasteurized process american with di sodium phosphate    1       901014
1043    Cheese pasteurized process pimento      1       901013
1044    Cheese pasteurized process swiss with di sodium phosphate       1       901015
1045    Cheese food cold pack american  1       901003
1046    Cheese food pasteurized process american without di sodium phosphate    1       901004
1047    Cheese food pasteurized process swiss   1       901004
1048    Cheese spread pasteurized process american without di sodium phosphate  1       901005
1049    Cream fluid half and    1       901019
1050    Cream fluid light coffee or table       1       901019
1052    Cream fluid light whipping      1       901019
1053    Cream fluid heavy whipping      1       901019
1054    Cream whipped topping pressurized       1       901017
1055    Cream sour reduced fat cultured 1       901017
1056    Cream sour cultured     1       901017
1057    Eggnog  1       901021
1058    Sour dressing non-butterfat cultured filled cream-type  1       901038
1059    Milk filled fluid with blend of hydrogenated vegetable oils     1       901027
1060    Milk filled fluid with lauric acid oil  1       901027
1067    Cream substitute liquid with hydrogenated vegetable oil and soy protein 1       901018
1068    Cream substitute liquid with lauric acid oil and sodium caseinate       1       901018
1069    Cream substitute powdered       1       901017
1070    Dessert topping powdered        1       901020
1071    Dessert topping powdered 1.5 ounce prepared with 1/2 cup milk   1       901020
1072    Dessert topping pressurized     1       901020
1073    Dessert topping semi solid frozen       1       901020
1074    Sour cream imitation cultured   1       901039
1075    Milk substitutes fluid with hydrogenated vegetable oils 1       901027
1076    Milk substitutes fluid with lauric acid oil     1       901027
1077    Milk whole 3.25% milkfat        53      901027
1078    Milk producer fluid 3.7% milkfat        1       901027
1079    Milk reduced fat fluid 2% milkfat with added vitamin a  1       901037
1080    Milk reduced fat fluid 2% milkfat with added nonfat solids and vitamin a        1       901037
1081    Milk reduced fat fluid 2% milkfat protein fortified with added vitamin a        1       901037
```

7. Updates

MyNetDiary dataset changes over time to reflect changes in food industry: vendors buy and sell food brands, companies change name while growing the business, vendors introduce new product flavors. While improving the dataset, MyNetDiary research team marks obsolete foods as retired at their discretion. Sometimes, MyNetDiary team will retire complete set of foods for some dynamic vendor like Subway or McDonald's and create new food definitions from scratch. When food is retired, apps using MyNetDiary database will find new food label by the same keywords. The retired food stays in the database for reporting purposes.

Dataset update files contain all current foods. Some foods may be added, other may be modified, and majority of foods unchanged since the last extract.
In addition to current set, the update includes retiredFoodId.txt file containing all food identifiers retired since the beginning of MyNetDiary data research. Some of these identifiers may be not found in your dataset, which should be tolerated by you update process.

Here is an example of update process for MySQL database. Please notice that **update process does not delete anything, it adds, updates or retires foods:**

1. Make sure your database has no connections serving end users.

2. Load the data using mysql utility:

```
SET foreign_key_checks = 0;

load data local infile 'Food.txt' replace into table Food;
load data local infile 'FoodGroup.txt' replace into table FoodGroup;
load data local infile 'FoodNutrientValue.txt' replace into table FoodNutrientValue;
load data local infile 'FoodWeight.txt' replace into table FoodWeight;
load data local infile 'Nutrient.txt' replace into table Nutrient;

create temporary table retiredFood(id int not null primary key);
load data local infile 'retiredFoodId.txt' into table retiredFood;
update Food set retired=true where foodId in (select id from retiredFood);

SET foreign_key_checks = 1;
```

3. Verify food count, it should be the same in db and in file:

    mysql:

    ```
    select count(*) from food where userId=0 and retired=0 and contributed=false;
    ```

    file system:

    ```
    cat Food.txt |wc -l
    ```