Term Project: Nutrition Database

By: Shaumbe Granger, Anthony Arnold, Jose Varela

Table of Contents

Description

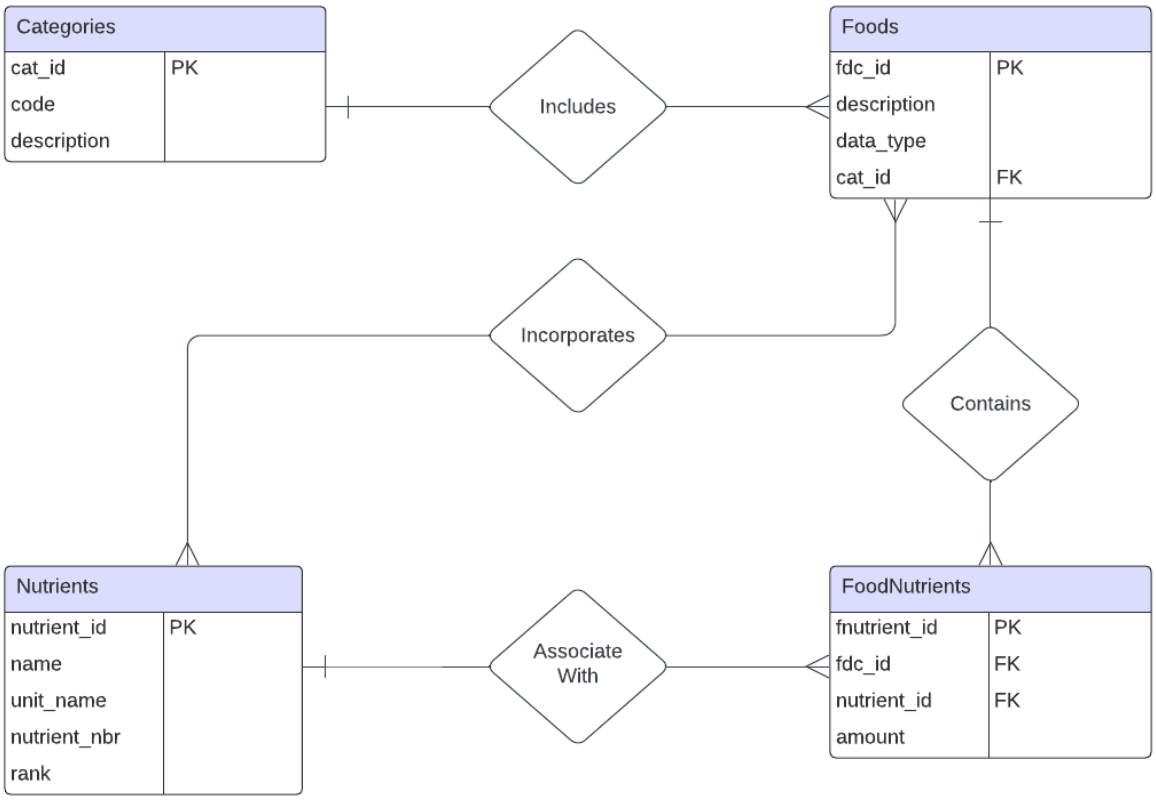
With the expansive selection of food items necessary for survival, it is important to understand the breakdown of nutrients within them. Complexity plagues the nutritional makeup of even the simplest food products, causing avoidance and confusion when diving deeper into dietary facts. In this project, the aim is to bridge the gap between the foods loved by all and the nutrients nested within them. Creating a relational database to hold nutrient and product information would act as a reliable resource to monitor, update, and add relations that easily connect foods with their nutritional facts. Users can query the database to find foods based on various criteria such as fiber content, calorie amount, fat content, iron content, etc. The implementation of a nutritional database would also promote healthier lifestyles, making it a very useful tool to use daily.

Design and Implementation

1. E-R Diagram

Figure 1

E-R Relationships Diagram



Entities:

1. Categories
2. Foods
3. Nutrients
4. FoodNutrients

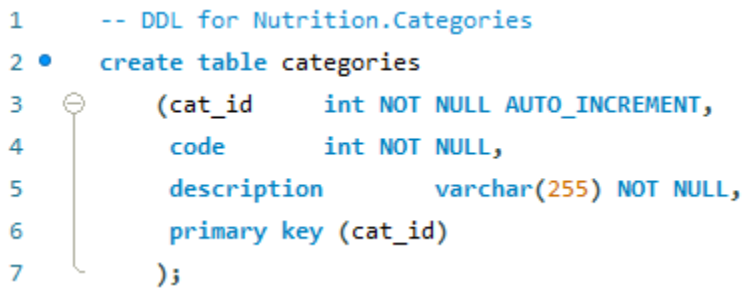
2. Attributes Description

Categories:

- cat_id: Primary key, unique identifier for each category (integer).
- code: Numeric code for the category (integer).
- description: Name or description of the category (string).
- Each attribute is atomic, and there are no repeating groups.
- Non-key attributes (code, description) are fully dependent on the primary key (id).

Figure 2

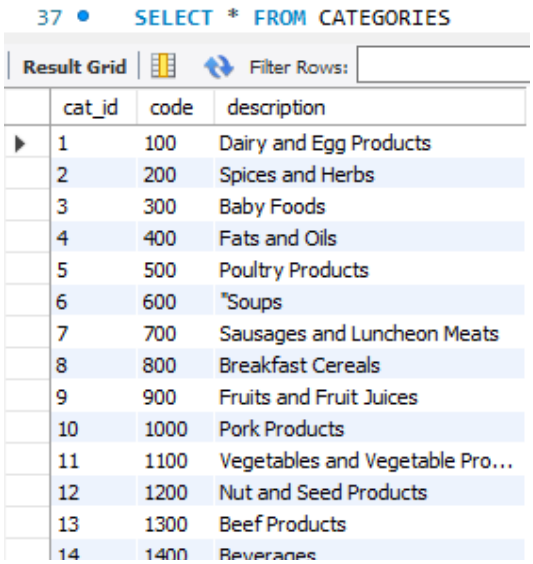
Screen capture showing the sql query used to create the Categories table.



```
1     -- DDL for Nutrition.Categories
2     create table categories
3         (cat_id     int NOT NULL AUTO_INCREMENT,
4          code       int NOT NULL,
5          description        varchar(255) NOT NULL,
6          primary key (cat_id)
7         );
```

Figure 3

Screen capture selecting from the Categories table.



cat_id	code	description
1	100	Dairy and Egg Products
2	200	Spices and Herbs
3	300	Baby Foods
4	400	Fats and Oils
5	500	Poultry Products
6	600	"Soups
7	700	Sausages and Luncheon Meats
8	800	Breakfast Cereals
9	900	Fruits and Fruit Juices
10	1000	Pork Products
11	1100	Vegetables and Vegetable Pro...
12	1200	Nut and Seed Products
13	1300	Beef Products
14	1400	Beverages

Foods:

- fdc_id: Primary key, unique identifier for each food (integer).
- description: Name or description of the food (string).
- data_type: Type of data (string).
- cat_id: Foreign key referencing the id column in the Categories table (integer).
- Each attribute is atomic, and there are no repeating groups.
- Non-key attributes (description, data_type, cat_id) are fully dependent on the primary key (fdc_id).

Figure 4

Screen capture showing the SQL query used to create the Foods table.

```
9     -- DDL for Nutrition.Foods
10    create table foods
11        (fdc_id     int NOT NULL,
12         data_type varchar(50) DEFAULT NULL,
13         `description`  varchar(255) NOT NULL,
14         cat_id int DEFAULT NULL,
15         primary key (fdc_id),
16         foreign key (cat_id) references categories (cat_id)
17        );
```

Figure 5

Screen capture selecting from the Foods table.

```
39    SELECT * FROM Foods
```

Result Grid | Filter Rows: | Edit:

fdc_id	data_type	description	cat_id
321358	foundation_food	Hummus; commercial	16
321359	foundation_food	Milk; reduced fat; fluid; 2% milkf...	1
321360	foundation_food	Tomatoes; grape; raw	11
321505	foundation_food	Salt; table; iodized	2
321611	foundation_food	Beans; snap; green; canned; re...	11
321900	foundation_food	Broccoli; raw	11
322228	foundation_food	Milk; lowfat; fluid; 1% milkfat; wi...	1
322559	foundation_food	Milk; nonfat; fluid; with added vi...	1
322892	foundation_food	Milk; whole; 3.25% milkfat; with ...	1
323121	foundation_food	Frankfurter; beef; unheated	7
323294	foundation_food	Nuts; almonds; dry roasted; wit...	12
323444	foundation_food	Cheese; ricotta; whole milk	1

FoodNutrients:

- fnutrient_id: Primary key, unique identifier for each record (integer).
- fdc_id: Foreign key referencing the fdc_id column in the Foods table (integer).
- nutrient_id: Foreign key referencing the id column in the Nutrients table (integer).
- amount: Amount of the nutrient in the food (decimal).
- Each attribute is atomic, and there are no repeating groups.
- Non-key attributes (fdc_id, nutrient_id, amount) are fully dependent on the primary key (id).

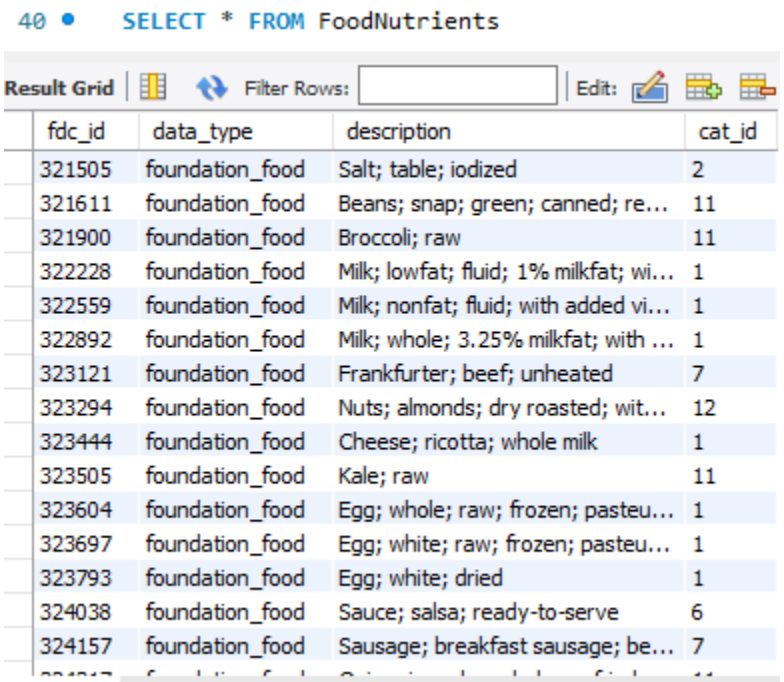
Figure 6

Screen capture showing the SQL query used to create the FoodNutrients table.

```
29    -- DDL for Nutrition.Foodnutrients
30    create table foodnutrients
31        (fnutrient_id int NOT NULL AUTO_INCREMENT,
32         fdc_id         int NOT NULL,
33         nutrient_id        int NOT NULL,
34         amount     decimal(10,2) NOT NULL,
35         primary key (fnutrient_id),
36         foreign key (fdc_id) references foods (fdc_id),
37         foreign key (nutrient_id) references nutrients (nutrient_id)
38        );
```

Figure 7

Screen capture selecting from the FoodNutrients table.



```
40 •    SELECT * FROM FoodNutrients
```

Result Grid | Filter Rows: | Edit:

fdc_id	data_type	description	cat_id
321505	foundation_food	Salt; table; iodized	2
321611	foundation_food	Beans; snap; green; canned; re...	11
321900	foundation_food	Broccoli; raw	11
322228	foundation_food	Milk; lowfat; fluid; 1% milkfat; wi...	1
322559	foundation_food	Milk; nonfat; fluid; with added vi...	1
322892	foundation_food	Milk; whole; 3.25% milkfat; with ...	1
323121	foundation_food	Frankfurter; beef; unheated	7
323294	foundation_food	Nuts; almonds; dry roasted; wit...	12
323444	foundation_food	Cheese; ricotta; whole milk	1
323505	foundation_food	Kale; raw	11
323604	foundation_food	Egg; whole; raw; frozen; pasteu...	1
323697	foundation_food	Egg; white; raw; frozen; pasteu...	1
323793	foundation_food	Egg; white; dried	1
324038	foundation_food	Sauce; salsa; ready-to-serve	6
324157	foundation_food	Sausage; breakfast sausage; be...	7

Nutrients:

- nutrient_id: Primary key, unique identifier for each nutrient (integer).
- name: Name or description of the nutrient (string).
- unit_name: Name of the unit used to measure the nutrient (string).
- nutrient_nbr: Numeric code for the nutrient (integer).
- rank: Rank of the nutrient (decimal).
- Each attribute is atomic, and there are no repeating groups.
- Non-key attributes (name, unit_name, nutrient_nbr, rank) are fully dependent on the primary key (id).

Figure 8

Screen capture showing the SQL query used to create the Nutrients table.

```
19      -- DDL for Nutrition.Nutrients
20      create table nutrients
21          (nutrient_id        int NOT NULL,
22           name           varchar(255) DEFAULT NULL,
23           unit_name      varchar(50) DEFAULT NULL,
24           nutrient_nbr           int DEFAULT NULL,
25           `rank` decimal(10,2) DEFAULT NULL,
26           primary key (nutrient_id)
27          );
```

Figure 9

Screen capture selecting from the Nutrients table.

```
40      SELECT * FROM Nutrients
```

Result Grid | Filter Rows: | Edit: | Export/Import:

nutrient_id	name	unit_name	nutrient_nbr	rank
1001	Solids	G	201	200.00
1002	Nitrogen	G	202	500.00
1003	Protein	G	203	600.00
1004	Total lipid (fat)	G	204	800.00
1005	Carbohydrate; by difference	G	205	1110.00
1006	Fiber; crude (DO NOT USE - Archived)	G	206	999999.00
1007	Ash	G	207	1000.00
1008	Energy	KCAL	208	300.00
1009	Starch	G	209	2200.00
1010	Sucrose	G	210	1600.00
1011	Glucose	G	211	1700.00
1012	Fructose	G	212	1800.00
1013	Lactose	G	213	1900.00
1014	Maltose	G	214	2000.00
1015	Amylose	G	218	999999.00
1016	Amylopectin	G	219	999999.00
1017	Pectin	G	220	999999.00

3. **Primary and Foreign Keys**

Categories

- **Primary key:** cat_id
- **Foreign keys:** N/A

Foods

- **Primary key:** fdc_id
- **Foreign keys:**
    - *Categories:* cat_id

Nutrients

- **Primary key:** nutrient_id
- **Foreign keys:** N/A

FoodNutrients

- **Primary key:** fnutrient_id
- **Foreign keys:**
    - *Foods:* fdc_id
    - *Nutrients:* nutrient_id

4. **Normal Form**

The tables appear to be in at least the Third Normal Form (3NF). Each attribute is atomic, there are no repeating groups, and each non-key attribute is fully dependent on the primary key.

5. **Multi-table Queries**

- **High Fiber Foods**

```
SELECT f.description AS Food, fn.amount AS FiberAmount, n.unit_name AS
Unit
FROM Foods f
JOIN FoodNutrients fn ON f.fdc_id = fn.fdc_id
JOIN Nutrients n ON fn.nutrient_id = n.nutrient_id
WHERE n.name LIKE '%Fiber%' AND fn.amount >= 5;
-- Assuming 5 grams as high fiber threshold
```

- **Low Calorie Foods**

```
SELECT f.description AS Food, fn.amount AS CalorieAmount, n.unit_name AS
Unit
FROM Foods f
JOIN FoodNutrients fn ON f.fdc_id = fn.fdc_id
JOIN Nutrients n ON fn.nutrient_id = n.nutrient_id
WHERE n.name LIKE '%Energy%' AND fn.amount <= 100;
-- Assuming 100 calories as low calorie threshold
```

Figure 10

Screen capture showing the results of the 'High Fiber Foods' query.

```sql
SELECT f.description AS Food, fn.amount AS FiberAmount, n.unit_name AS Unit
FROM Foods f
JOIN FoodNutrients fn ON f.fdc_id = fn.fdc_id
JOIN Nutrients n ON fn.nutrient_id = n.nutrient_id
WHERE n.name LIKE '%Fiber%' AND fn.amount >= 5;
-- Assuming 5 grams as high fiber threshold
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Food	FiberAmount	Unit
Hummus; commercial	5.40	G
Nuts; almonds; dry roasted; with salt added	11.00	G
Seeds; sunflower seed kernels; dry roasted; wit...	10.30	G
Figs; dried; uncooked	9.80	G
Restaurant; Latino; pupusas con frijoles (pupus...	5.80	G
Bread; whole-wheat; commercially prepared	6.00	G
Figs; dried; uncooked	9.80	G
Flour; whole wheat; unenriched	10.60	G
Flour; spelt; whole grain	9.34	G
Flour; potato	5.40	G
Flour; almond	9.27	G
Flour; oat; whole grain	10.50	G
Almond butter; creamy	9.72	G
Flaxseed; ground	23.13	G
Peanut butter; creamy	6.32	G
Sesame butter; creamy	8.37	G
Nuts; pecans; halves; raw	5.79	G
Nuts; walnuts; English; halves; raw	5.21	G

Figure 11

Screen capture showing the results of the 'Low Calorie Foods' query.

```
8 •  SELECT f.description AS Food, fn.amount AS CalorieAmount, n.unit_name AS Unit
9    FROM Foods f
10   JOIN FoodNutrients fn ON f.fdc_id = fn.fdc_id
11   JOIN Nutrients n ON fn.nutrient_id = n.nutrient_id
12   WHERE n.name LIKE '%Energy%' AND fn.amount <= 100;
13   -- Assuming 100 calories as low calorie threshold
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Food	CalorieAmount	Unit
Milk; reduced fat; fluid; 2% milkfat; with added ...	50.00	KCAL
Tomatoes; grape; raw	27.00	KCAL
Beans; snap; green; canned; regular pack; drai...	21.00	KCAL
Broccoli; raw	32.00	KCAL
Milk; lowfat; fluid; 1% milkfat; with added vitami...	43.00	KCAL
Milk; nonfat; fluid; with added vitamin A and vit...	34.00	KCAL
Milk; whole; 3.25% milkfat; with added vitamin D	60.00	KCAL
Kale; raw	35.00	KCAL
Egg; white; raw; frozen; pasteurized	48.00	KCAL
Sauce; salsa; ready-to-serve	29.00	KCAL
Pickles; cucumber; dill or kosher dill	12.00	KCAL
Grapefruit juice; white; canned or bottled; uns...	37.00	KCAL
Peaches; yellow; raw	42.00	KCAL
Kale; frozen; cooked; boiled; drained; without salt	36.00	KCAL
Carrots; frozen; unprepared (Includes foods fo...	37.00	KCAL
Mustard; prepared; yellow	61.00	KCAL
Kiwifruit; green; raw	58.00	KCAL
Melons; cantaloupe; raw	34.00	KCAL

6. Queries using SQL aggregate functions

- **Low Fat Foods**

  SELECT f.description AS Food, AVG(fn.amount) AS AvgFatAmount
  FROM Foods f
  JOIN FoodNutrients fn ON f.fdc_id = fn.fdc_id
  JOIN Nutrients n ON fn.nutrient_id = n.nutrient_id
  WHERE n.name = 'Total lipid (fat)'
  GROUP BY f.description
  HAVING AVG(fn.amount) <= 3;
  -- Assuming 3 grams as low fat threshold

- **High Iron Foods**

  SELECT f.description AS Food, MAX(fn.amount) AS MaxIronAmount
  FROM Foods f
  JOIN FoodNutrients fn ON f.fdc_id = fn.fdc_id

```
JOIN Nutrients n ON fn.nutrient_id = n.nutrient_id
WHERE n.name LIKE '%Iron%Fe%'
GROUP BY f.description
HAVING MAX(fn.amount) >= 5;
-- Assuming 5 milligrams as high iron threshold
```

Figure 11

Screen capture showing the results of the 'Low Fat Foods' query.

```
15 •  SELECT f.description AS Food, AVG(fn.amount) AS AvgFatAmount
16    FROM Foods f
17    JOIN FoodNutrients fn ON f.fdc_id = fn.fdc_id
18    JOIN Nutrients n ON fn.nutrient_id = n.nutrient_id
19    WHERE n.name = 'Total lipid (fat)'
20    GROUP BY f.description
21    HAVING AVG(fn.amount) <= 3;
22    -- Assuming 3 grams as low fat threshold
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Food	AvgFatAmount
Milk; reduced fat; fluid; 2% milkfat; with added ...	1.900000
Tomatoes; grape; raw	0.630000
Beans; snap; green; canned; regular pack; drai...	0.390000
Broccoli; raw	0.340000
Milk; lowfat; fluid; 1% milkfat; with added vitami...	0.950000
Milk; nonfat; fluid; with added vitamin A and vit...	0.080000
Kale; raw	1.490000
Egg; white; raw; frozen; pasteurized	0.160000
Egg; white; dried	0.650000
Sauce; salsa; ready-to-serve	0.190000
Pickles; cucumber; dill or kosher dill	0.430000
Grapefruit juice; white; canned or bottled; uns...	0.700000
Peaches; yellow; raw	0.270000
Kale; frozen; cooked; boiled; drained; without salt	1.210000
Carrots; frozen; unprepared (Includes foods fo...	0.470000

Figure 12

Screen capture showing the results of the 'High Iron Foods' query.

```
24 •  SELECT f.description AS Food, MAX(fn.amount) AS MaxIronAmount
25    FROM Foods f
26    JOIN FoodNutrients fn ON f.fdc_id = fn.fdc_id
27    JOIN Nutrients n ON fn.nutrient_id = n.nutrient_id
28    WHERE n.name LIKE '%Iron%Fe%'
29    GROUP BY f.description
30    HAVING MAX(fn.amount) >= 5;
31    -- Assuming 5 milligrams as high iron threshold
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Food	MaxIronAmount
Seeds; sunflower seed kernels; dry roasted; wit...	5.21
Egg; whole; dried	6.97
Egg; yolk; dried	9.28
Beans; Dry; Dark Red Kidney (0% moisture)	6.58
Beans; Dry; Pink (0% moisture)	5.46
Beans; Dry; Navy (0% moisture)	5.29
Beans; Dry; Light Red Kidney (0% moisture)	6.95
Beans; Dry; Carioca (0% moisture)	5.87
Beans; Dry; Tan (0% moisture)	5.85
Beans; Dry; Pinto (0% moisture)	5.40
Beans; Dry; Medium Red (0% moisture)	6.27
Beans; Dry; Cranberry (0% moisture)	5.26
Beans; Dry; Light Tan (0% moisture)	7.29
Beans; Dry; Black (0% moisture)	5.34
Beans; Dry; Red (0% moisture)	5.05

7. **Query using subquery**

- **Foods with specific nutrients (e.g., Spinach)**

```
SELECT Foods.description AS Food, Nutrients.name AS Nutrient,
FoodNutrients.amount, Nutrients.unit_name
FROM FoodNutrients
JOIN Nutrients ON FoodNutrients.nutrient_id = Nutrients.nutrient_id
JOIN (
        SELECT fdc_id, description
        FROM Foods
        WHERE description LIKE '%Spinach%'
        )
 AS Foods ON FoodNutrients.fdc_id = Foods.fdc_id;
```

Figure 13

Screen capture showing the results of the 'Foods with specific nutrients' query.

```
33 •  SELECT Foods.description AS Food, Nutrients.name AS Nutrient, FoodNutrients.amount, Nutrients.unit_name
34    FROM FoodNutrients
35    JOIN Nutrients ON FoodNutrients.nutrient_id = Nutrients.nutrient_id
36    JOIN (
37                SELECT fdc_id, description
38                FROM Foods
39                WHERE description LIKE '%Spinach%'
40    ) AS Foods ON FoodNutrients.fdc_id = Foods.fdc_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Food	Nutrient	amount	unit_name
Spinach; baby	Iron; Fe	1.26	MG
Spinach; baby	Magnesium; Mg	92.86	MG
Spinach; baby	Phosphorus; P	39.05	MG
Spinach; baby	Potassium; K	581.80	MG
Spinach; baby	Sodium; Na	111.40	MG
Spinach; baby	Zinc; Zn	0.45	MG
Spinach; baby	cis-Lutein/Zeaxanthin	0.00	UG
Spinach; baby	Copper; Cu	0.08	MG
Spinach; baby	Vitamin C; total ascorbic acid	26.52	MG
Spinach; baby	Thiamin	0.08	MG
Spinach; baby	Manganese; Mn	0.49	MG
Spinach; baby	Riboflavin	0.19	MG
Spinach; baby	Selenium; Se	0.00	UG
Spinach; baby	Niacin	0.55	MG
Spinach; baby	Carotene; beta	3397.44	UG
Spinach; baby	Carotene; alpha	0.00	UG

8. Query including GROUP BY and HAVING

- **Grouping Foods by Category with High Fat Content:**

```
SELECT c.description AS Category, COUNT(*) AS FoodCount
FROM Foods f
JOIN Categories c ON f.cat_id = c.cat_id
JOIN FoodNutrients fn ON f.fdc_id = fn.fdc_id
JOIN Nutrients n ON fn.nutrient_id = n.nutrient_id
WHERE n.name = 'Total lipid (fat)' AND fn.amount > 10
-- Assuming 10 grams as high fat threshold
GROUP BY c.description
HAVING COUNT(*) > 5;
-- Assuming at least 5 foods per category
```

Figure 14

Screen capture showing the results of the 'Grouping Foods by Category with High Fat Content' query.

```
42 •  SELECT c.description AS Category, COUNT(*) AS FoodCount
43    FROM Foods f
44    JOIN Categories c ON f.cat_id = c.cat_id
45    JOIN FoodNutrients fn ON f.fdc_id = fn.fdc_id
46    JOIN Nutrients n ON fn.nutrient_id = n.nutrient_id
47    WHERE n.name = 'Total lipid (fat)' AND fn.amount > 10
48    -- Assuming 10 grams as high fat threshold
49    GROUP BY c.description
50    HAVING COUNT(*) > 5;
51    -- Assuming at least 5 foods per category
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Category	FoodCount
Sausages and Luncheon Meats	9
Nut and Seed Products	18
Dairy and Egg Products	29
Beef Products	6