# M2 Data Science Professional Practise – Data Science Project

## Introduction

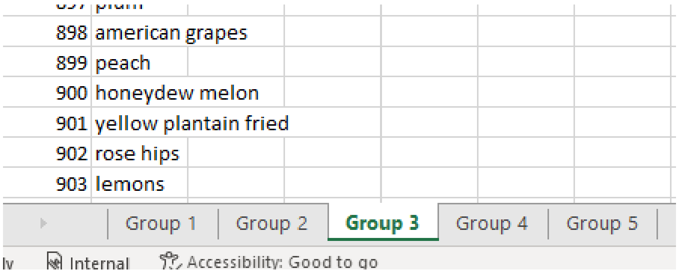
This project explored the nutritional value of foods using a structured data science approach to develop a custom healthiness scoring framework. A publicly available dataset of nutritional information per 100g was used, which included values for macronutrients (such as protein, fat, and carbohydrates) and micronutrients (such as iron, vitamin C, and potassium). The dataset was cleaned and stored in SQL Server, enriched with external lookup data via Excel, and visualised in an interactive Power BI dashboard.

The project aimed to calculate a composite health score for each food item based on evidence from public health frameworks. This included guidance from the World Health Organization (WHO, 2003) and the UK Food Standards Agency's traffic light labelling system (FSA, 2020). Additional calculations were introduced to identify WHO compliance levels and categorise foods as red, amber, or green in terms of nutritional risk. This project demonstrates practical data engineering skills, analytical thinking, and ethical awareness in designing a scoring model that can inform consumer choices in a meaningful way.

## Data Infrastructure & Tools

Three primary tools were used to support this project: SQL Server, Excel, and Power BI. SQL Server was selected for its ability to handle structured data transformations and aggregation efficiently. The dataset was imported from CSV and stored in a dedicated schema. Data type issues, such as fields being interpreted as text rather than numbers, were addressed using SQL casting functions such as TRY_CAST() and ISNULL() (Microsoft Docs, 2023). These were essential for enabling reliable numeric calculations during the scoring and validation phases.

Excel was used to support the addition of lookup values for food names and group classifications. The original dataset used numerical reference codes, so several Excel sheets containing mappings from these references to actual food names and groupings were consolidated. These tables were appended and imported into Power BI, where relationships were built to connect them to the main SQL dataset.



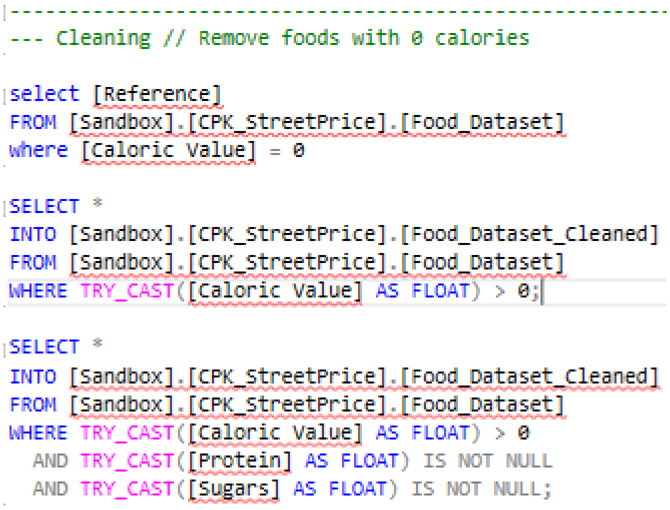
| | |
|---|---|
| 897 | plum |
| 898 | american grapes |
| 899 | peach |
| 900 | honeydew melon |
| 901 | yellow plantain fried |
| 902 | rose hips |
| 903 | lemons |

Power BI was used for data modelling and visualisation due to its compatibility with SQL Server and its intuitive dashboard interface. The tool allowed for the development of a clear and interactive reporting environment, with slicers, custom measures, and tooltips to improve insight generation. Its ability to dynamically link visuals with filters and enable drill-down capability was key to making the health score and classification system understandable to non-technical users.

Overall, the chosen infrastructure provided a scalable, modular workflow that could be updated or replicated across other food datasets or public health models.

## Data Engineering

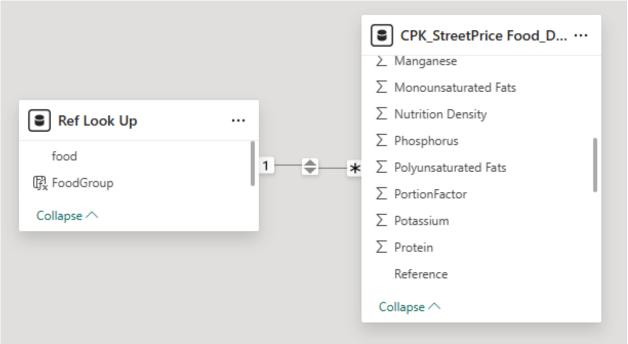
Data engineering began with importing the raw CSV file into SQL Server. Challenges encountered included truncated text fields, incorrect encodings, and improper field types — particularly where numeric values were stored as VARCHAR. These issues were resolved using ALTER TABLE scripts and TRY_CAST() to convert fields like fat, sodium, protein, and sugar to FLOAT types (Microsoft Docs, 2023).



```
------------------------------------------------------------
--- Cleaning // Remove foods with 0 calories

select [Reference]
FROM [Sandbox].[CPK_StreetPrice].[Food_Dataset]
where [Caloric Value] = 0

SELECT *
INTO [Sandbox].[CPK_StreetPrice].[Food_Dataset_Cleaned]
FROM [Sandbox].[CPK_StreetPrice].[Food_Dataset]
WHERE TRY_CAST([Caloric Value] AS FLOAT) > 0;

SELECT *
INTO [Sandbox].[CPK_StreetPrice].[Food_Dataset_Cleaned]
FROM [Sandbox].[CPK_StreetPrice].[Food_Dataset]
WHERE TRY_CAST([Caloric Value] AS FLOAT) > 0
  AND TRY_CAST([Protein] AS FLOAT) IS NOT NULL
  AND TRY_CAST([Sugars] AS FLOAT) IS NOT NULL;
```

Records with invalid or misleading data were filtered out, including rows where caloric value was zero or missing. Such entries, which might represent trace ingredients or non-consumable items, could distort health score calculations and were excluded from the analytical model.

A central feature of the data transformation process was the creation of a custom HealthinessScore. This calculated field was developed using weighted values of positive nutrients (e.g. dietary fibre, protein, unsaturated fats, vitamin C, B6, magnesium, potassium) and negative nutrients (e.g. saturated fats, sugars, sodium). These were normalised by caloric content and scaled using a PortionFactor variable to account for likely consumption amounts. The formula was designed to follow principles from nutrition research which support the benefits of whole foods rich in fibre and protein while discouraging excessive sugar, fat, and sodium intake (Harvard T.H. Chan School of Public Health, 2021).

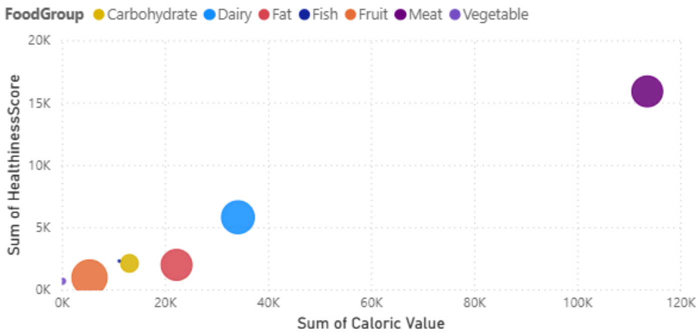


Lookup values for food names and categories were integrated using Excel sheets. These were imported into Power BI and joined using a one-to-many relationship based on the Reference column. Each transformation step was validated using SQL queries such as AVG(), TOP(), and comparisons between row counts before and after cleaning.

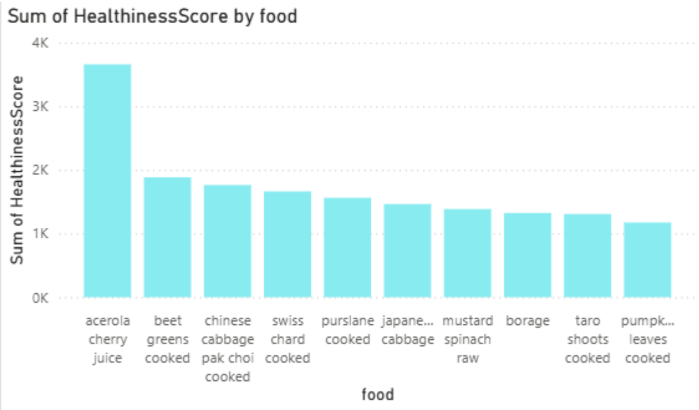
This phase of the project ensured that data was accurate, meaningful, and ready for scoring, classification, and visualisation.

## Data Visualisation & Dashboards

The cleaned and enriched dataset was connected to Power BI, where a visual dashboard was developed to explore the calculated health metrics. Visual design principles were applied to promote clarity, accessibility, and interactivity.

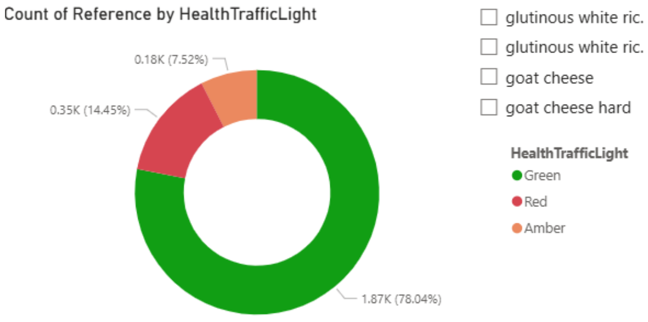


The dashboard included a scatter plot showing the relationship between calories and HealthinessScore, with bubble size reflecting sugar content and

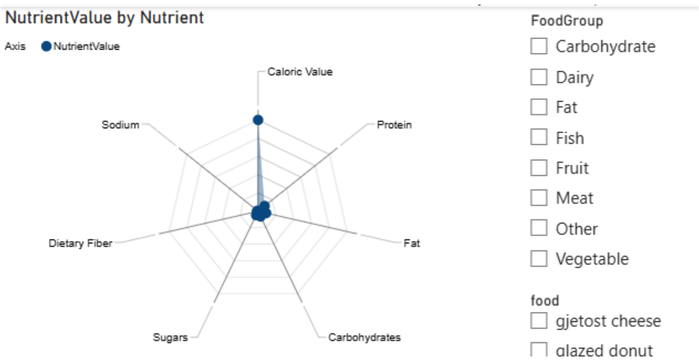


colour indicating food group. This enabled users to quickly identify high-calorie but healthy foods (e.g. nuts, seeds) and contrast them with high-sugar or high-sodium items. A bar chart highlighted the top 10 healthiest foods, with filters in place to remove items with implausibly low-calorie values, improving insight accuracy.

A donut chart displayed the distribution of foods by traffic light classification (green, amber, red), using thresholds based on FSA guidelines (FSA, 2020). A radar chart was also included, allowing users to view the nutrient breakdown of a single food item across selected categories such as protein, fat, fibre, and sodium.



Slicers were introduced for FoodGroup, calorie thresholds, and HealthTrafficLight, allowing users to customise their exploration experience. Tooltips were enriched with nutrient data and scoring labels to aid interpretation.



These visuals collectively transformed the scoring logic into an accessible decision-support tool, enhancing user understanding of how various foods compare nutritionally.

## Data Analytics

The core analytical task in this project was designing a score to represent the healthiness of a food item based on its nutrient composition. Inspired by approaches used in public health frameworks (WHO, 2003; FSA, 2020), the scoring model combined empirical guidelines with statistical aggregation.

The HealthinessScore formula was based on weighted averages of nutrients. Positive weights were applied to protein, fibre, unsaturated fats, and micronutrients such as potassium and vitamin C, while negative weights were applied to saturated fats, sodium, and sugar. The score was then normalised per 100 kcal to avoid penalising nutrient-dense, high-calorie foods that are typically eaten in small quantities. The PortionFactor column was used to scale scores for foods like condiments, which may be healthy in small quantities but otherwise skew results.

A WHO compliance metric was also created. This score indicated whether each food met key nutrient thresholds defined by the WHO's Europe Region model — for example, whether it had less than 10g of sugar and less than 500mg of sodium per 100g (WHO, 2003). This was used as a secondary benchmark to compare against the custom score.

A traffic light classification system was added using UK FSA guidance. Each nutrient was assigned a red, amber, or green flag based on defined thresholds, and foods were categorised accordingly. This enabled the dashboard to provide a colour-coded health risk indicator that was simple to interpret and familiar to UK audiences (FSA, 2020).

Ethically, care was taken to avoid misrepresenting nutritional risk. By filtering out foods with low calories and assigning weightings to account for consumption frequency, the model avoided giving misleadingly high scores to uncommon or non-food items. No personal or sensitive data was processed, and all datasets were open access.

**Referencing and Academic Support**

The scoring approach and food classification logic were based on nutritional guidance from several key public health sources. The World Health Organization (2003) provided the nutrient profile model used to establish pass/fail thresholds for key nutrients. The UK Food Standards Agency's (2020) traffic light labelling system offered a benchmark for colour-coded classification, while The Harvard T.H. Chan School of Public Health (2021) was used to support the inclusion of specific micronutrients such as potassium and magnesium in the health score model.

Technical references from Microsoft Learn (2023) supported SQL functions used to transform and clean the data, particularly the use of TRY_CAST, ISNULL, and the SQL Import Wizard. Visualisation guidance was informed by best practice principles in data storytelling and interactivity.

**Recommendations and Future Improvements**

While the current model provides a strong foundation for analysing nutritional health, several future improvements could enhance its precision and applicability. Introducing more realistic portion size estimates using published dietary data or food packaging information would better account for typical consumption. Expanding the classification system using natural language processing to auto-categorise foods based on their names could streamline food group assignment. A machine learning model could also be developed to predict health scores based on nutrient profiles and compare them against actual public health recommendations. Additionally, incorporating pricing or cost-per-portion data could support more practical comparisons for health-conscious budgeting. These enhancements would further strengthen the utility and relevance of the dashboard for end users.

**References**

World Health Organization (WHO), 2003. *Diet, Nutrition and the Prevention of Chronic Diseases.* WHO Technical Report Series, No. 916. Geneva: WHO.

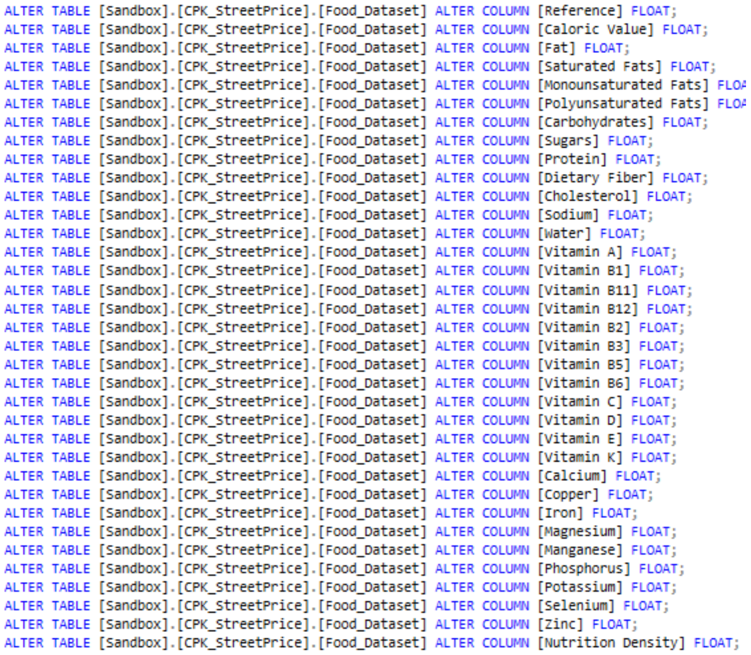
UK Food Standards Agency, 2020. *Front of Pack Nutrition Labelling Guidance.* [online] Available at: <https://www.food.gov.uk> [Accessed 11 March 2025].

Harvard T.H. Chan School of Public Health, 2021. *The Nutrition Source – Carbohydrates, Fats and Proteins.* [online] Available at: <https://www.hsph.harvard.edu/nutritionsource/> [Accessed 14 March 2025].

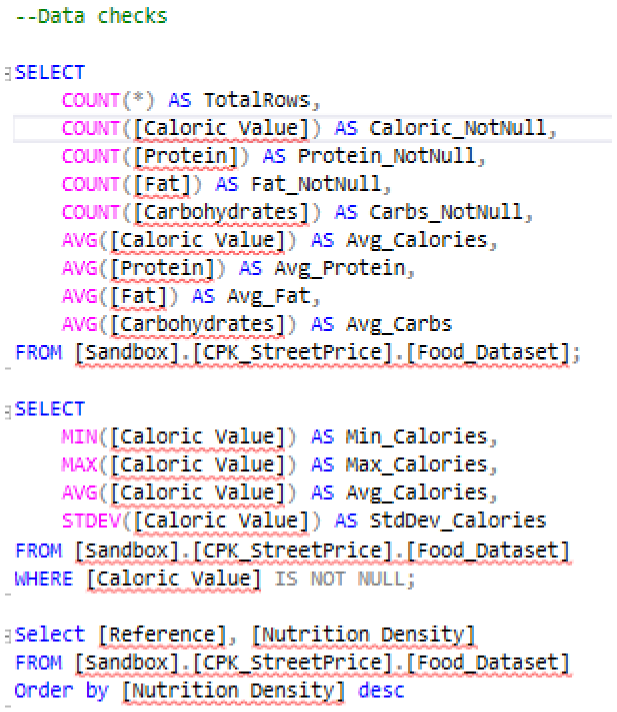
Microsoft, 2023. *CAST and CONVERT (Transact-SQL).* [online] Microsoft Learn. Available at: <https://learn.microsoft.com/en-us/sql/t-sql/functions/cast-and-convert-transact-sql> [Accessed 14 March 2025].

Few, S., 2009. *Now You See It: Simple Visualization Techniques for Quantitative Analysis.* Oakland, CA: Analytics Press.

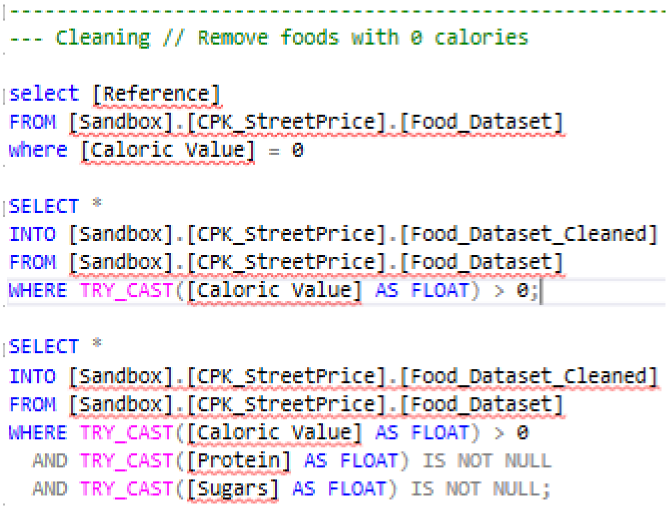
## Other/Enlarged SQL Scripts



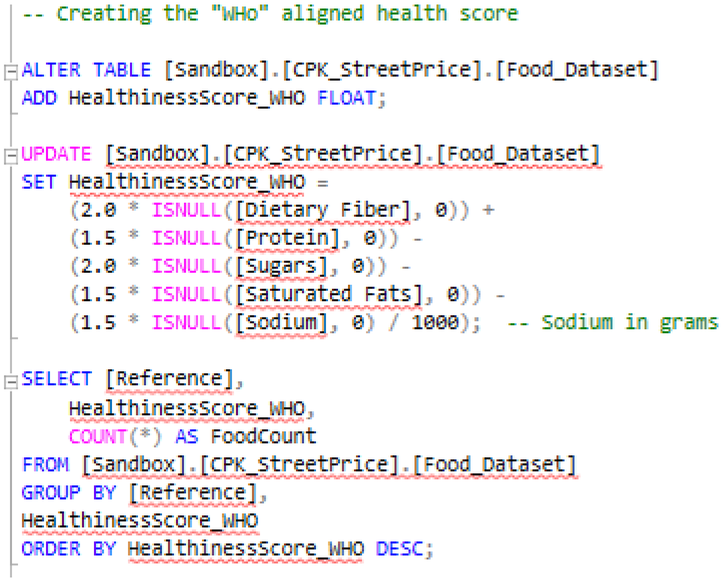
```sql
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Reference] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Caloric Value] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Fat] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Saturated Fats] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Monounsaturated Fats] FLOA
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Polyunsaturated Fats] FLOA
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Carbohydrates] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Sugars] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Protein] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Dietary Fiber] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Cholesterol] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Sodium] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Water] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Vitamin A] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Vitamin B1] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Vitamin B11] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Vitamin B12] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Vitamin B2] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Vitamin B3] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Vitamin B5] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Vitamin B6] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Vitamin C] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Vitamin D] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Vitamin E] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Vitamin K] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Calcium] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Copper] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Iron] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Magnesium] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Manganese] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Phosphorus] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Potassium] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Selenium] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Zinc] FLOAT;
ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset] ALTER COLUMN [Nutrition Density] FLOAT;
```



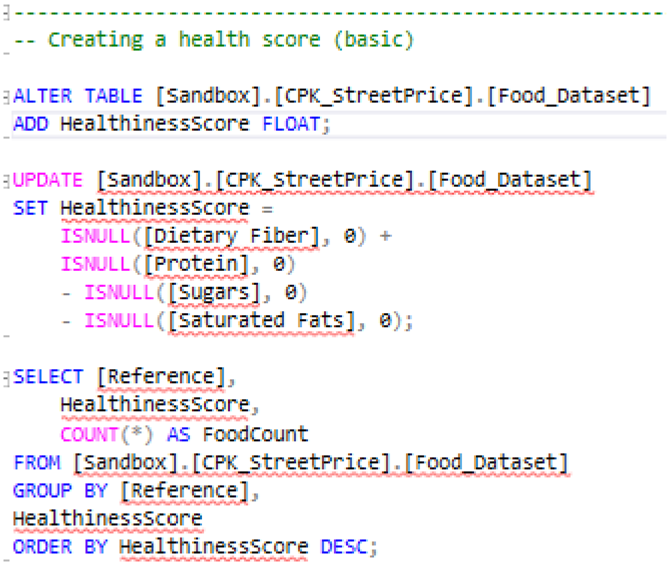
```sql
--Data checks

SELECT
    COUNT(*) AS TotalRows,
    COUNT([Caloric Value]) AS Caloric_NotNull,
    COUNT([Protein]) AS Protein_NotNull,
    COUNT([Fat]) AS Fat_NotNull,
    COUNT([Carbohydrates]) AS Carbs_NotNull,
    AVG([Caloric Value]) AS Avg_Calories,
    AVG([Protein]) AS Avg_Protein,
    AVG([Fat]) AS Avg_Fat,
    AVG([Carbohydrates]) AS Avg_Carbs
FROM [Sandbox].[CPK_StreetPrice].[Food_Dataset];

SELECT
    MIN([Caloric Value]) AS Min_Calories,
    MAX([Caloric Value]) AS Max_Calories,
    AVG([Caloric Value]) AS Avg_Calories,
    STDEV([Caloric Value]) AS StdDev_Calories
FROM [Sandbox].[CPK_StreetPrice].[Food_Dataset]
WHERE [Caloric Value] IS NOT NULL;

Select [Reference], [Nutrition Density]
FROM [Sandbox].[CPK_StreetPrice].[Food_Dataset]
Order by [Nutrition Density] desc
```



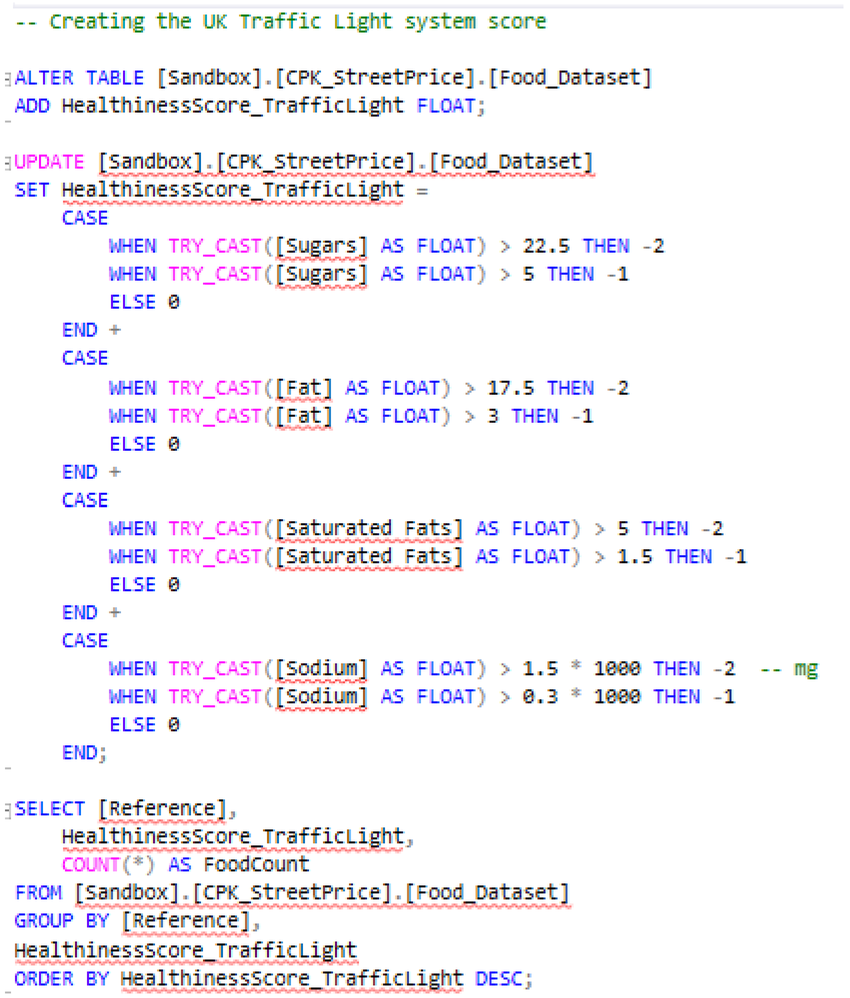
```sql
------------------------------------------------------------
--- Cleaning // Remove foods with 0 calories

select [Reference]
FROM [Sandbox].[CPK_StreetPrice].[Food_Dataset]
where [Caloric Value] = 0

SELECT *
INTO [Sandbox].[CPK_StreetPrice].[Food_Dataset_Cleaned]
FROM [Sandbox].[CPK_StreetPrice].[Food_Dataset]
WHERE TRY_CAST([Caloric Value] AS FLOAT) > 0;

SELECT *
INTO [Sandbox].[CPK_StreetPrice].[Food_Dataset_Cleaned]
FROM [Sandbox].[CPK_StreetPrice].[Food_Dataset]
WHERE TRY_CAST([Caloric Value] AS FLOAT) > 0
  AND TRY_CAST([Protein] AS FLOAT) IS NOT NULL
  AND TRY_CAST([Sugars] AS FLOAT) IS NOT NULL;
```



```
-- Creating the "WHo" aligned health score

ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset]
ADD HealthinessScore_WHO FLOAT;

UPDATE [Sandbox].[CPK_StreetPrice].[Food_Dataset]
SET HealthinessScore_WHO =
    (2.0 * ISNULL([Dietary Fiber], 0)) +
    (1.5 * ISNULL([Protein], 0)) -
    (2.0 * ISNULL([Sugars], 0)) -
    (1.5 * ISNULL([Saturated Fats], 0)) -
    (1.5 * ISNULL([Sodium], 0) / 1000);   -- Sodium in grams

SELECT [Reference],
    HealthinessScore_WHO,
    COUNT(*) AS FoodCount
FROM [Sandbox].[CPK_StreetPrice].[Food_Dataset]
GROUP BY [Reference],
HealthinessScore_WHO
ORDER BY HealthinessScore_WHO DESC;
```



```
-------------------------------------------------------
-- Creating a health score (basic)

ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset]
ADD HealthinessScore FLOAT;

UPDATE [Sandbox].[CPK_StreetPrice].[Food_Dataset]
SET HealthinessScore =
    ISNULL([Dietary Fiber], 0) +
    ISNULL([Protein], 0)
    - ISNULL([Sugars], 0)
    - ISNULL([Saturated Fats], 0);

SELECT [Reference],
    HealthinessScore,
    COUNT(*) AS FoodCount
FROM [Sandbox].[CPK_StreetPrice].[Food_Dataset]
GROUP BY [Reference],
HealthinessScore
ORDER BY HealthinessScore DESC;
```



```
-- Creating the UK Traffic Light system score

ALTER TABLE [Sandbox].[CPK_StreetPrice].[Food_Dataset]
ADD HealthinessScore_TrafficLight FLOAT;

UPDATE [Sandbox].[CPK_StreetPrice].[Food_Dataset]
SET HealthinessScore_TrafficLight =
    CASE
        WHEN TRY_CAST([Sugars] AS FLOAT) > 22.5 THEN -2
        WHEN TRY_CAST([Sugars] AS FLOAT) > 5 THEN -1
        ELSE 0
    END +
    CASE
        WHEN TRY_CAST([Fat] AS FLOAT) > 17.5 THEN -2
        WHEN TRY_CAST([Fat] AS FLOAT) > 3 THEN -1
        ELSE 0
    END +
    CASE
        WHEN TRY_CAST([Saturated Fats] AS FLOAT) > 5 THEN -2
        WHEN TRY_CAST([Saturated Fats] AS FLOAT) > 1.5 THEN -1
        ELSE 0
    END +
    CASE
        WHEN TRY_CAST([Sodium] AS FLOAT) > 1.5 * 1000 THEN -2  -- mg
        WHEN TRY_CAST([Sodium] AS FLOAT) > 0.3 * 1000 THEN -1
        ELSE 0
    END;

SELECT [Reference],
    HealthinessScore_TrafficLight,
    COUNT(*) AS FoodCount
FROM [Sandbox].[CPK_StreetPrice].[Food_Dataset]
GROUP BY [Reference],
HealthinessScore_TrafficLight
ORDER BY HealthinessScore_TrafficLight DESC;
```