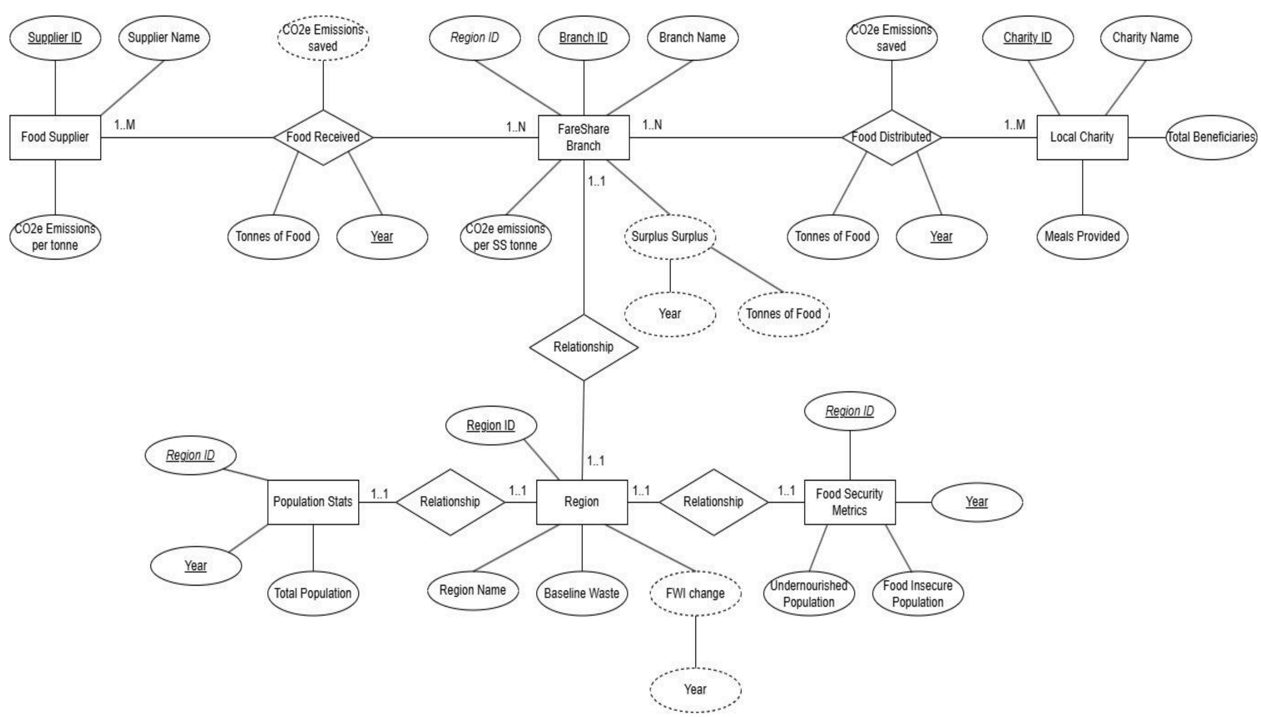
INST0001 Database Systems 2024-25
Group A15 Group Work
Student numbers: 24166100, 24163454, 22086535
Word Count: 1645

Introduction:

FareShare is a charitable organization focused on redistributing good-to-eat surplus food to local charities that feed people in need [1]. This food would otherwise go to waste, which has significant environmental implications. Our database aims to track FareShare's contributions to three Sustainable Development Goals (SDGs)[2]:

- **SDG 2: Zero Hunger**
  - **Indicator 2.1.1:** Prevalence of undernourishment
  - **Indicator 2.1.2:** Prevalence of moderate or severe food insecurity
- **SDG 12: Responsible Consumption and Production**
  - **Indicator 12.3.1:** Food waste index (FWI) reduction
- **SDG 13: Climate Action**
  - **Indicator 13.2.2:** Total Greenhouse Gas (GHG) emissions per year

ER Diagram:



Overview of ER diagram:

Where appropriate, entities have a unique identifier attribute and readable name in text. Here are justifications for non-self-explanatory attributes:

FareShare Branch: Represents the operational branches responsible for surplus food transport.

- **Region ID**: Links branches and geographical regions.
- **CO2E per Surplus Tonne**: Emissions released per tonne of surplus surplus food, (food that is received but failed to be distributed).

Food Supplier: Stores information about suppliers providing surplus food.

- **CO2 Emissions per Tonne**: CO2 emissions per tonne of food that the food supplier would normally dispose of without intervention from FareShare.

Local Charity: Tracks charities that receive redistributed food and distribute it to beneficiaries.

- **Total Beneficiaries**: Number of people served, relevant to SDG12.
- **Meals Provided**: Number of meals distributed.

Region: Defines the geographical areas where FareShare operates.

- **Baseline Waste**: Total food waste in a region for a baseline year for FWI.
- **FWI Change (derived)**: Monitors trends in food waste reduction for SDG12

Population Stats: Captures demographic data necessary for understanding food security challenges across regions.

- **Region ID:** Links population data to specific regions.
- **Year:** Allows for tracking changes in population over time.
- **Total Population:** Enables evaluating the proportions of the population affected by food insecurity.

Food Security Metrics: Tracks key hunger-related indicators essential for evaluating progress toward food security goals.

- **Region ID**: Ensures food security data is linked to specific regions for localized analysis.
- **Year**: Enables time-series analysis of food security trends.
- **Undernourished Population**: Quantifies individuals experiencing chronic hunger with consumption insufficient to meet minimum dietary energy requirements.
- **Food Insecure Population**: Quantifies individuals with experiences of uncertainty in access to sufficient food or inadequate food quality.

The relationships between entities are structured to reflect the practical usage of the database. The Food Received relationship between Food Supplier and FareShare Branch is a many-to-many (N:M) relationship, where each supplier can provide surplus food to multiple branches, and each branch can receive food from multiple suppliers. This relationship includes attributes such as Tonnes of Food and Year, enabling the calculation of emissions saved and redistribution efficiency. The Food Distributed relationship between FareShare Branch and Local Charity is also N:M. This relationship tracks redistributed food quantities, CO_2 emissions saved, and meals provided. Additionally, Region-based relationships integrate demographic data with food security metrics, facilitating a structured analysis of FareShare's impact across geographical areas. The Region and FareShare Branch entities maintain a 1:1 relationship, as each branch operates within a designated region, while regional statistics entities, including Population Stats and Food Security Metrics, also share a 1:1 relationship with the Region entity, ensuring complete coverage for demographic and food security analysis. Food Received and Food Distributed are modelled with complete participation cardinalities to represent that every branch receives and supplies food and that food suppliers and local charities would only be stored in the database if they were actively involved.

Derived attributes implemented in individual reports.

Overview of Schema Design:

- FOOD SUPPLIER (<u>Supplier_ID</u>, Supplier_Name, CO2_Emissions_per_Tonne)
  - Constraints:
    - Supplier_Name must be NOT NULL

- FARESHARE BRANCH (<u>Branch_ID</u>, *Region_ID*, Branch_Name, CO2E_per_SS_Tonne)
  - Foreign Keys:
    - Region_ID → References REGION(Region_ID)
  - Constraints:
    - Region_ID must be NOT NULL
    - Branch_Name must be NOT NULL

- LOCAL CHARITY (<u>Charity_ID</u>, Charity_Name, Total_Beneficiaries, Meals_Provided)
  - Constraints:
    - Charity_Name must be NOT NULL

- FOOD RECEIVED (<u>*Supplier_ID*</u>, <u>*Branch_ID*</u>, <u>Year</u>, Tonnes_of_Food, CO2E_Emissions_Saved)
  - Foreign Keys:
    - Supplier_ID → References FOOD SUPPLIER(Supplier_ID)
    - Branch_ID → References FARESHARE BRANCH(Branch_ID)

- FOOD DISTRIBUTED (<u>*Branch_ID*</u>, <u>*Charity_ID*</u>, <u>Year</u>, Tonnes_of_Food, CO2E_Emissions_Saved)
  - Foreign Keys:
    - Branch_ID → References FARESHARE BRANCH(Branch_ID)
    - Charity_ID → References LOCAL CHARITY(Charity_ID)

- REGION (<u>Region_ID</u>, Region_Name, Baseline_Waste)
  - Constraints:
    - Region_Name must be NOT NULL

- POPULATION STATS (<u>*Region_ID*</u>, <u>Year</u>, Total_Population)
  - Foreign Keys:
    - Region_ID → References REGION(Region_ID)

- FOOD SECURITY METRICS (<u>*Region_ID*</u>, <u>Year</u>, Undernourished_Population, Food_Insecure_Population)
  - Foreign Keys:
    - Region_ID → References REGION(Region_ID)

Each entity is uniquely identifiable through primary keys, such as Supplier_ID for Food Supplier, preventing duplicate records and ensuring entity integrity. Composite keys in Food Received and Food Distributed ((Supplier_ID, Branch_ID, Year) and (Branch_ID, Charity_ID, Year)) ensure accurate time-dependent tracking of food flow.

Referential integrity is enforced using foreign keys to link entities. Region_ID in FareShare Branch links each branch to its respective region, while Supplier_ID and Branch_ID in Food Received ensure that food transactions involve only registered suppliers and branches.

Constraints are applied to maintain data consistency and prevent errors. NOT NULL constraints on critical attributes such as Supplier Name ensure that essential information is always recorded. Foreign keys with NOT NULL constraints guarantee that all branches, food suppliers, and charities must be associated with valid regions and organizations, reflecting real-world dependencies.

Normalisation:

We ensured our database is normalized by ensuring the following:

First Normal Form (1NF):

A table is in 1NF if:

1. It contains only atomic (indivisible) values—no lists or sets of values in a single field.
2. There are no repeating groups or multivalued attributes.

In our schema:

- The Local Charity table has attributes Charity_ID, Charity_Name, Total_Beneficiaries, and Meals_Provided, all of which are atomic. For instance, Total_Beneficiaries stores a single numeric value per row rather than a list of beneficiaries.
- The Food Received table stores Tonnes_of_Food in separate rows for each (Supplier_ID, Branch_ID, Year) combination instead of using a single row to store multiple food quantities received over multiple years.

Second Normal Form (2NF):

A table is in 2NF if:

1. It is already in 1NF.
2. Non-key attribute depends on entire primary key, not just part of it (No partial dependencies).

To ensure 2NF, we identified and resolved partial dependencies:

- The Food Received table has a composite primary key (Supplier_ID, Branch_ID, Year). The attribute Tonnes_of_Food depends on the full composite key, not just a part of it. If Year were removed, Tonnes_of_Food would no longer be uniquely identified.

- The Food Security Metrics table separates time-variant data (e.g., Undernourished_Population, Food_Insecure_Population) from time-invariant data (e.g., Region_Name in the REGION table), ensuring that each non-key attribute fully depends on the primary key (Region_ID, Year).

Third Normal Form (3NF):

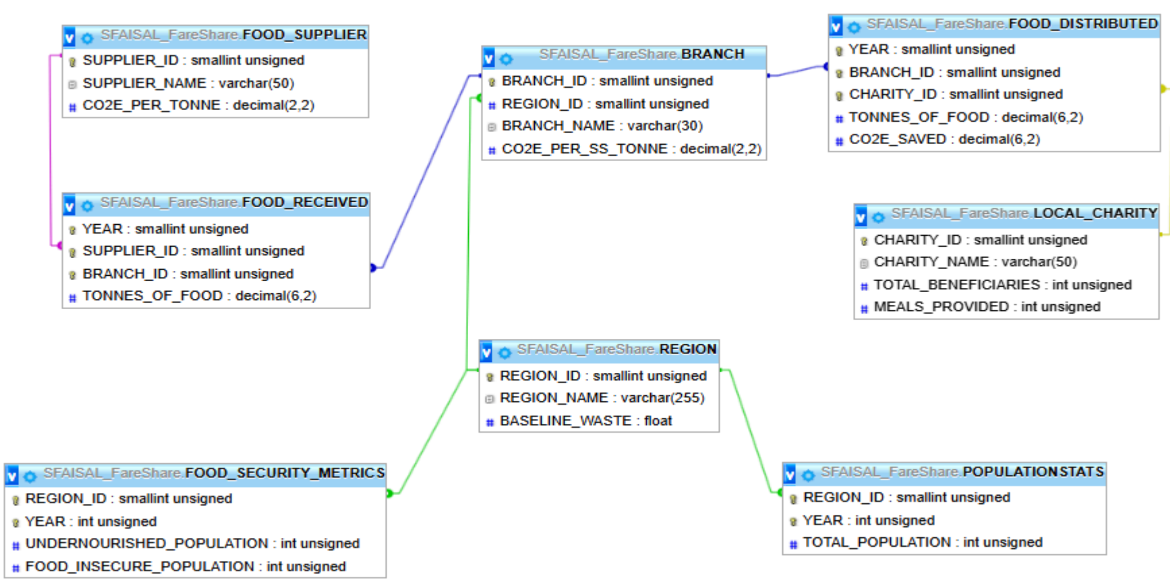
A table is in 3NF if:

1. It is in 2NF.
2. Non-key attributes do not depend on other non-key attributes (no transitive dependencies).

To ensure 3NF, we identified and removed transitive dependencies:

- In the Food Security Metrics table, Undernourished_Population and Food_Insecure_Population depend only on the primary key (Region_ID, Year) and not on any other non-key attribute. If these values were influenced by Baseline_Waste from the Region table, it would create a transitive dependency. Instead, Baseline_Waste is stored separately in the Region table, preventing this issue.
- CO2E_Saved in the Food Distributed table is a derived attribute that can be calculated dynamically (e.g., using queries based on Tonnes_of_Food and CO2E_per_SS_Tonne). Instead of storing redundant values, we calculate it as needed, eliminating transitive dependencies and reducing data redundancy.

MySQL implementation:



The creation of entity tables was done in a structured sequence to ensure foreign key dependencies were correctly instantiated. For example, the Region table had to be created before Branch since Branch references Region_ID. Similarly, Food_Supplier and Local_Charity were created before defining relationships in Food_Received and Food_Distributed.

Below are typical examples of SQL commands used for various DDL tasks in the implementation of our database:

Entity table creation:

```
CREATE TABLE REGION
(
    REGION_ID SMALLINT UNSIGNED,
    REGION_NAME VARCHAR(255) NOT NULL,
    BASELINE_WASTE FLOAT
    COMMENT 'BASELINE FOOD WASTE IN TONNES',
    PRIMARY KEY (REGION_ID)
);

CREATE TABLE BRANCH
(
    BRANCH_ID SMALLINT UNSIGNED,
    REGION_ID SMALLINT UNSIGNED NOT NULL,
    BRANCH_NAME VARCHAR(30) NOT NULL,
    CO2E_PER_SS_TONNE DECIMAL(2,2),
    PRIMARY KEY (BRANCH_ID),
    FOREIGN KEY (REGION_ID) REFERENCES REGION(REGION_ID)
);
```

Relationship table creation:

```
CREATE TABLE FOOD_RECEIVED
(
    YEAR SMALLINT UNSIGNED,
    SUPPLIER_ID SMALLINT UNSIGNED,
    BRANCH_ID SMALLINT UNSIGNED,
    TONNES_OF_FOOD DECIMAL(5,2) NOT NULL,
    PRIMARY KEY (BRANCH_ID, SUPPLIER_ID, YEAR),
    FOREIGN KEY (BRANCH_ID) REFERENCES BRANCH(BRANCH_ID)
    ON DELETE RESTRICT ON UPDATE CASCADE,
    FOREIGN KEY (SUPPLIER_ID) REFERENCES FOOD_SUPPLIER(SUPPLIER_ID)
    ON DELETE CASCADE ON UPDATE CASCADE
);
```

The selection of appropriate data types was key to ensuring accuracy and efficiency.

- Numeric attributes such as TONNES_OF_FOOD and CO2E_SAVED were defined using DECIMAL, ensuring precise calculations without rounding errors.

- Primary keys such as SUPPLIER_ID, BRANCH_ID, and REGION_ID were set as SMALLINT UNSIGNED, conserving storage and restricting domain to positive integers.
- Text-based attributes like SUPPLIER_NAME and REGION_NAME used VARCHAR, with lengths chosen to balance storage efficiency.

Foreign key constraints play a critical role in maintaining data consistency. If a primary key changes, all related records update automatically (ON UPDATE CASCADE). If a record is deleted, its linked records are also deleted to keep the database clean (ON DELETE CASCADE).

We applied logical constraints to ensure data integrity like Undernourished_Population <= Food_Insecure_Population or Meals_Provided >= Total_Beneficiaries.

```
ALTER TABLE LOCAL_CHARITY
ADD CONSTRAINT CHK_MEALS_BENEFICIARIES CHECK (MEALS_PROVIDED >= TOTAL_BENEFICIARIES);

ALTER TABLE FOOD_SECURITY_METRICS
ADD CONSTRAINT CHK_POPULATIONS_CONSISTENT CHECK (FOOD_INSECURE_POPULATION >= UNDERNOURISHED_POPULATION);
```

Below are the example SQL statements for inserting data in entities.

```
INSERT INTO REGION (REGION_ID, REGION_NAME, BASELINE_WASTE) VALUES
(10, 'London', 1460000), (20, 'Sussex', 164250),
(30, 'Plymouth', 21900), (40, 'Birmingham', 200750);

INSERT INTO BRANCH (BRANCH_ID, REGION_ID, BRANCH_NAME, CO2E_PER_SS_TONNE) VALUES
(1, 10, 'FareShare London', 0.05), (2, 20, 'FareShare Sussex', 0.04),
(3, 30, 'FareShare Plymouth', 0.03), (4, 40, 'FareShare Birmingham', 0.06);
```

Synthetic Data Generation:

The synthetic data was generated with realism and variability in mind, drawing on real-world references [1][3][4][5] to ensure accuracy. The data was modeled to reflect plausible trends and estimates, such as food received and redistributed increasing annually in line with FareShare's growth. To add realism, slight noise was introduced by introducing random fluctuations.

Causality and interconnectivity were also key factors in the data generation process. For example, as more food is redistributed, it leads to a reduction in food insecurity and CO2 emissions. Similarly, food received was always greater than or equal to food distributed, maintaining a logical flow of food.

Part of the data was synthesized with the assistance of ChatGPT-4o, which helped in ensuring the logical interconnections and consistency across various data points.

Critical Reflection: Limitations and future work:

A major difficulty was in enforcing certain logical constraints, such as ensuring that "Food Distributed" is always less than or equal to "Food Received". MySQL's limitations, particularly the inability to enforce subqueries within check constraints, meant that we had to explore alternative methods like triggers. However, due to admin permissions, this approach was also not possible. This experience revealed how database constraints are not always as straightforward as they might initially seem, and how external factors, like database permissions and system configuration, can influence design decisions.

The integration of real-world trends and variability in synthetic data generation was another key challenge. While it was crucial to model food distribution data in a way that reflects the realities of organizations like FareShare, ensuring that this data remained plausible while introducing randomness to account for real-world fluctuations required careful balancing.

Future improvements could focus on capturing seasonal variations in food supply and demand, which directly impact distribution patterns. The constraints discussed earlier could also be enforced at the application level, offering more flexibility in managing business logic and ensuring data integrity. Additionally, while 'emissions per tonne' values for food suppliers and branches are currently treated as constants for simplicity, a more dynamic approach could involve restructuring the database to account for changes in waste management by the organisations. This would require adding new tables linked to Food Supplier and Branch entities with a “Year” prime attribute, enabling more sophisticated tracking of emissions over time.

References:

[1] FareShare website (https://fareshare.org.uk/ accessed on 23/02/2025)

[2] UN The 17 Goals (https://sdgs.un.org/goals accessed on 23/02/2025)

[3] Food Waste in the UK (https://researchbriefings.files.parliament.uk/documents/CBP-7552/CBP-7552.pdf accessed on 25/02/2025)

[4] J.A. Moult, S.R. Allan, C.N. Hewitt, M. Berners-Lee, “Greenhouse gas emissions of food waste disposal options for UK retailers” (https://www.sciencedirect.com/science/article/pii/S0306919217309168#s0015 accessed on 25/02/2025)

[5] R. Hannah, “You want to reduce the carbon footprint of your food? Focus on what you eat, not whether your food is local” ('https://ourworldindata.org/food-choice-vs-eating-local accessed on 12/03/2025)

Acknowledgment:

We would like to acknowledge the use of ChatGPT-4o, for assisting us in structuring the report. The model helped make the text more concise and aligned with the required word limit (since we were going quite above it) while ensuring clarity and coherence throughout.

The data for Food Security Metrics and Food Received was generated by ChatGPT-4o through properly thought-of and realistic prompts (with proper context provided):

> generate synthetic data for Food Security Metrics and Food Distributed keeping in mind that it should show improvement in SDG indicators i am tracking and the improvement should be non-linear (variable):
> SDG 2: Zero Hunger
> Indicator 2.1.1: Prevalence of undernourishment
> Indicator 2.1.2: Prevalence of moderate or severe food insecurity
> In particular:
> - increased food distribution should be correlated with decreased food unnourished and food insecure population
> - food distribution should increase every year, and food insecurity/food undernourished population should decrease every year