INST0001 - Database Systems 2024-25

Individual Work

Student ID: 24163454
Module Tutor's Name: Karen Stepanyan, Foteini Valeonti
Word Count: 1095 words

The **Sustainable Development Goal 2 (Zero Hunger)** aims to "end hunger, achieve food security and improved nutrition, and promote sustainable agriculture" by 2030 [1]. This report focuses on monitoring two critical indicators: **SDG 2.1.1 (Prevalence of Undernourishment)** and **SDG 2.1.2 (Prevalence of Moderate or Severe Food Insecurity)**. These indicators measure the percentage of the population whose food consumption is insufficient to meet dietary energy requirements and those facing difficulties accessing food of sufficient quality and quantity, respectively [2].

The core design principle was to build a modular, scalable schema for FareShare where food security metrics, population statistics, and food distribution data are stored in separate but interconnected tables to effectively monitor SDG 2 indicators across **regions** and **time periods**. This approach aligns with recommendations by the UNECE, which emphasize that SDG monitoring systems require **temporal flexibility** and **regional granularity** [3].

A key architectural decision was to make **Region an entity**, not just an attribute. This choice was based on its central role in comparative analysis: each Region has its own demographic profile, food insecurity rates, and food aid distribution patterns. Treating Region as an entity enables rich one-to-many relationships with other tables (e.g., one Region can have multiple years of food security data or multiple distribution branches). The PopulationStats entity captures demographic data such as total population in the region. By linking this entity to the Food Security Metrics table, which records yearly data on undernourishment and food insecurity levels, the database enables the calculation of relative food insecurity and undernourished population percentages, enhancing comparability across regions and time periods. To analyze the relationship between food distribution efforts and food insecurity, the Food Distributed entity tracks the amount of food supplied to different regions each year. This table is linked to the Branch entity, which represents distribution centers responsible for redistributing food. The connection between food distribution, population statistics, and food security metrics allows for in-depth analysis of whether food aid efforts effectively reduce undernourishment and food insecurity over time. By maintaining separate but well-integrated tables, the database ensures data normalization while enabling complex queries that measure food security trends, assess the efficiency of food redistribution programs, and identify high-priority regions for intervention.

A crucial consideration in database design was ensuring **time-series tracking** to analyze trends in food security indicators over time. By including a Year attribute in key tables, queries can assess historical trends, year-over-year changes, and long-term progress toward SDG 2 targets [4]. This allows policymakers and stakeholders to determine whether interventions are effective or if certain regions require urgent action.

In terms of **optimization**, data types were carefully chosen to balance storage efficiency and computational performance—for instance, FLOAT or DECIMAL data types were used for food insecurity percentages to maintain precision without unnecessary memory consumption. An additional implementation step was ensuring **data consistency and accuracy**. To achieve this, data validation mechanisms were incorporated to prevent inconsistencies, such as ensuring that Food Insecure Population >= Undernourished Population (Undernourished individuals fall within the broader category of Food Insecurity), or Meals Provided >= Total Beneficiaries (More than one meal can be provided to a beneficiary).

Another crucial implementation aspect was the ability to calculate **dynamic indicators** — such as Food Insecurity percentages — on the fly. Rather than storing precomputed values, the database is designed to compute these metrics in real time through SQL queries that join the FoodSecurityMetrics table with PopulationStats. This means new indicators, such as child malnutrition, can be added as columns in FoodSecurityMetrics or as a separate linked table related to children data, maintaining modularity without structural changes.

The first key query calculates the food insecurity percentage per region and year, providing a comparative assessment of food security across different geographical areas, allowing identification of regions where food insecurity is most severe.

```
SELECT R.REGION_NAME,
       P.YEAR, ROUND((FSM.FOOD_INSECURE_POPULATION / P.TOTAL_POPULATION) * 100, 1) AS FOOD_INSECURE_PERCENTAGE
FROM FOOD_SECURITY_METRICS FSM
JOIN POPULATIONSTATS P ON FSM.REGION_ID = P.REGION_ID
AND FSM.YEAR = P.YEAR
JOIN REGION R ON P.REGION_ID = R.REGION_ID
ORDER BY R.REGION_NAME, P.YEAR;

```

REGION_NAME	YEAR	FOOD_INSECURE_PERCENTAGE
Birmingham	2021	24.0
Birmingham	2022	22.0
Birmingham	2023	20.0
Birmingham	2024	18.0
London	2021	12.5
London	2022	12.0
London	2023	11.0
London	2024	10.0
Plymouth	2021	19.0
Plymouth	2022	17.0
Plymouth	2023	15.0
Plymouth	2024	13.0
Sussex	2021	15.0
Sussex	2022	14.0
Sussex	2023	13.0
Sussex	2024	11.5

The following query investigates the correlation between food distribution and food security, assessing the extent to which increased food supply impacts the reduction of hunger.

```
SELECT R.REGION_NAME,
       FD.YEAR,
       SUM(FD.TONNES_OF_FOOD) AS TONNES_OF_FOOD_DISTRIBUTED,
       (SELECT FSM.FOOD_INSECURE_POPULATION
            FROM FOOD_SECURITY_METRICS FSM
            WHERE FSM.REGION_ID = R.REGION_ID
            AND FSM.YEAR = FD.YEAR) AS FOOD_INSECURE_POPULATION
FROM FOOD_DISTRIBUTED FD
JOIN BRANCH B ON FD.BRANCH_ID = B.BRANCH_ID
JOIN REGION R ON B.REGION_ID = R.REGION_ID
GROUP BY R.REGION_NAME, FD.YEAR
ORDER BY R.REGION_NAME, FD.YEAR;
```

REGION_NAME	YEAR	TONNES_OF_FOOD_DISTRIBUTED	FOOD_INSECURE_POPULATION
Birmingham	2021	852.80	481920
Birmingham	2022	969.00	450000
Birmingham	2023	1117.60	421200
Birmingham	2024	1306.40	392400
London	2021	2905.64	1050000
London	2022	3140.85	1020600
London	2023	3431.80	950620
London	2024	3802.10	880000
Plymouth	2021	100.04	41800
Plymouth	2022	119.00	38352
Plymouth	2023	140.80	34920
Plymouth	2024	170.20	31265
Sussex	2021	618.20	240000
Sussex	2022	689.35	227500
Sussex	2023	775.20	214630
Sussex	2024	881.60	193545

Another query identifies the top three regions with the highest undernourished populations, helping prioritize hunger hotspots for effective resource allocation.

```
SELECT R.REGION_NAME,
       FSM.YEAR,
       FSM.UNDERNOURISHED_POPULATION
FROM FOOD_SECURITY_METRICS FSM
JOIN REGION R ON FSM.REGION_ID = R.REGION_ID
WHERE FSM.YEAR = 2021
ORDER BY FSM.UNDERNOURISHED_POPULATION DESC
LIMIT 3;
```

REGION_NAME	YEAR	UNDERNOURISHED_POPULATION
London	2021	210000
Birmingham	2021	100400
Sussex	2021	48000

Furthermore, tracking total food distributed per region over time reveals trends in food supply efforts, showing whether redistribution is increasing, stagnating, or declining in specific areas.

```
SELECT R.REGION_NAME,
       FD.YEAR,
       SUM(FD.TONNES_OF_FOOD) AS TONNES_OF_FOOD_DISTRIBUTED
FROM FOOD_DISTRIBUTED FD
JOIN BRANCH B ON FD.BRANCH_ID = B.BRANCH_ID
JOIN REGION R ON B.REGION_ID = R.REGION_ID
GROUP BY R.REGION_NAME, FD.YEAR
ORDER BY R.REGION_NAME, FD.YEAR;
```

REGION_NAME	YEAR	TONNES_OF_FOOD_DISTRIBUTED
Birmingham	2021	852.80
Birmingham	2022	969.00
Birmingham	2023	1117.60
Birmingham	2024	1306.40
London	2021	2905.64
London	2022	3140.85
London	2023	3431.80
London	2024	3802.10
Plymouth	2021	100.04
Plymouth	2022	119.00
Plymouth	2023	140.80
Plymouth	2024	170.20
Sussex	2021	618.20
Sussex	2022	689.35
Sussex	2023	775.20
Sussex	2024	881.60

Before this project, I had a foundational understanding of relational databases, normalization, and SQL querying, but applying these principles to a real-world scenario significantly deepened my expertise. I initially designed direct relationships between all tables, creating a web of dependencies that complicated queries and violated normalization principles. This failure taught me to step back and reconsider the fundamental purpose of the database—monitoring progress toward ending hunger across regions and time — and restructure around the Region entity as the central reference point. I realised an effective database design begins with understanding the domain problem thoroughly before diving into technical implementation.

One of the key realizations was structuring food security metrics separately from population statistics, as this separation was essential for maintaining data integrity and avoiding duplication. However, this design choice came with the challenge of ensuring efficient joins to ensure that queries remained performant when handling large datasets. Furthermore, the importance of temporal tracking became evident during database implementation. It was essential to structure the database so that all records were tied to a specific year, making it possible to track trends and perform detailed historical comparisons.

One significant challenge faced was applying real-world constraints on our synthetic data like Food Received >= Food Distributed. Initially, I attempted to implement this constraint using a CHECK constraint with a subquery, only to realize that MySQL does not support subqueries within CHECK constraints. To work around this, I explored using trigger functions, but this approach was also unsuccessful due to insufficient administrative privileges. This experience has fundamentally changed my perspective on ensuring data quality and highlighted the importance of handling cross-table checks at the application level, rather than solely relying on database constraints.

Working with real-world data and addressing the challenges it presented also involved emotional engagement. The weight of dealing with hunger data made me more aware of the human impact behind the numbers. I realized that technical work can sometimes distance us from the real-world implications it represents. To maintain this awareness, I deliberately included a "Total Beneficiaries" field to keep the human side of the data visible. This shift in perspective transformed my view of the project, from a technical task to a tool for influencing change. Moving forward, I will approach database design with a greater emphasis on understanding the broader context in which the data will be used, ensuring that future work is not only efficient but also impactful.

References:

[1] United Nations. Goal 2 | End hunger, achieve food security and improved nutrition and promote sustainable agriculture. SDGs. 2024. Available at: https://sdgs.un.org/goals/goal2.

[2] SDGIndicators. 2.1.1 Prevalence of undernourishment. Food and Agriculture Organization of the United Nations. Available at: https://www.fao.org/sustainable-development-goals-data-portal/data/indicators/2.1.1-prevalence-of-undernourishment/en.

[3] United Nations Economic Commission for Europe. Measuring the progress of the Sustainable Development Goals in the UNECE region. 2021. Available at: https://unece.org/sites/default/files/2021-04/2012761_E_web.pdf.

[4] Joschka Herteux, Christoph Raeth, Martini G, Baha A, Kyriacos Koupparis, Ilaria Lauzana, et al. Forecasting trends in food security with real time data. Communications Earth & Environment. 2024 Oct 21. Available at: https://www.nature.com/articles/s43247-024-01698-9

Appendix:

Food Insecurity Percentage per Region and Year

```
SELECT R.REGION_NAME,
       P.YEAR, ROUND( (FSM.FOOD_INSECURE_POPULATION /
       P.TOTAL_POPULATION) * 100, 1) AS FOOD_INSECURE_PERCENTAGE
FROM FOOD_SECURITY_METRICS FSM
JOIN POPULATIONSTATS P ON FSM.REGION_ID = P.REGION_ID
AND FSM.YEAR = P.YEAR
JOIN REGION R ON P.REGION_ID = R.REGION_ID
ORDER BY R.REGION_NAME, P.YEAR;
```

Correlating Food Distribution with Food Insecure Population:

```
SELECT R.REGION_NAME,
        FD.YEAR, SUM(FD.TONNES_OF_FOOD) AS TONNES_OF_FOOD_DISTRIBUTED,
(SELECT FSM.FOOD_INSECURE_POPULATION
   FROM FOOD_SECURITY_METRICS FSM
   WHERE FSM.REGION_ID = R.REGION_ID
   AND FSM.YEAR = FD.YEAR) AS FOOD_INSECURE_POPULATION
FROM FOOD_DISTRIBUTED FD
JOIN BRANCH B ON FD.BRANCH_ID = B.BRANCH_ID
JOIN REGION R ON B.REGION_ID = R.REGION_ID
GROUP BY R.REGION_NAME, FD.YEAR
ORDER BY R.REGION_NAME, FD.YEAR;
```

Top 3 Regions with the most undernourished population in a particular year:

```
SELECT R.REGION_NAME,
          FSM.YEAR,
          FSM.UNDERNOURISHED_POPULATION
FROM FOOD_SECURITY_METRICS FSM
JOIN REGION R ON FSM.REGION_ID = R.REGION_ID
WHERE FSM.YEAR = 2021
ORDER BY FSM.UNDERNOURISHED_POPULATION DESC
LIMIT 3;
```

Tracking Total Food Distributed Per Region by Year:

```
SELECT R.REGION_NAME,
          FD.YEAR,
          SUM(FD.TONNES_OF_FOOD) AS TONNES_OF_FOOD_DISTRIBUTED
FROM FOOD_DISTRIBUTED FD
JOIN BRANCH B ON FD.BRANCH_ID = B.BRANCH_ID
JOIN REGION R ON B.REGION_ID = R.REGION_ID
GROUP BY R.REGION_NAME, FD.YEAR
ORDER BY R.REGION_NAME, FD.YEAR
```

Acknowledgment:

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