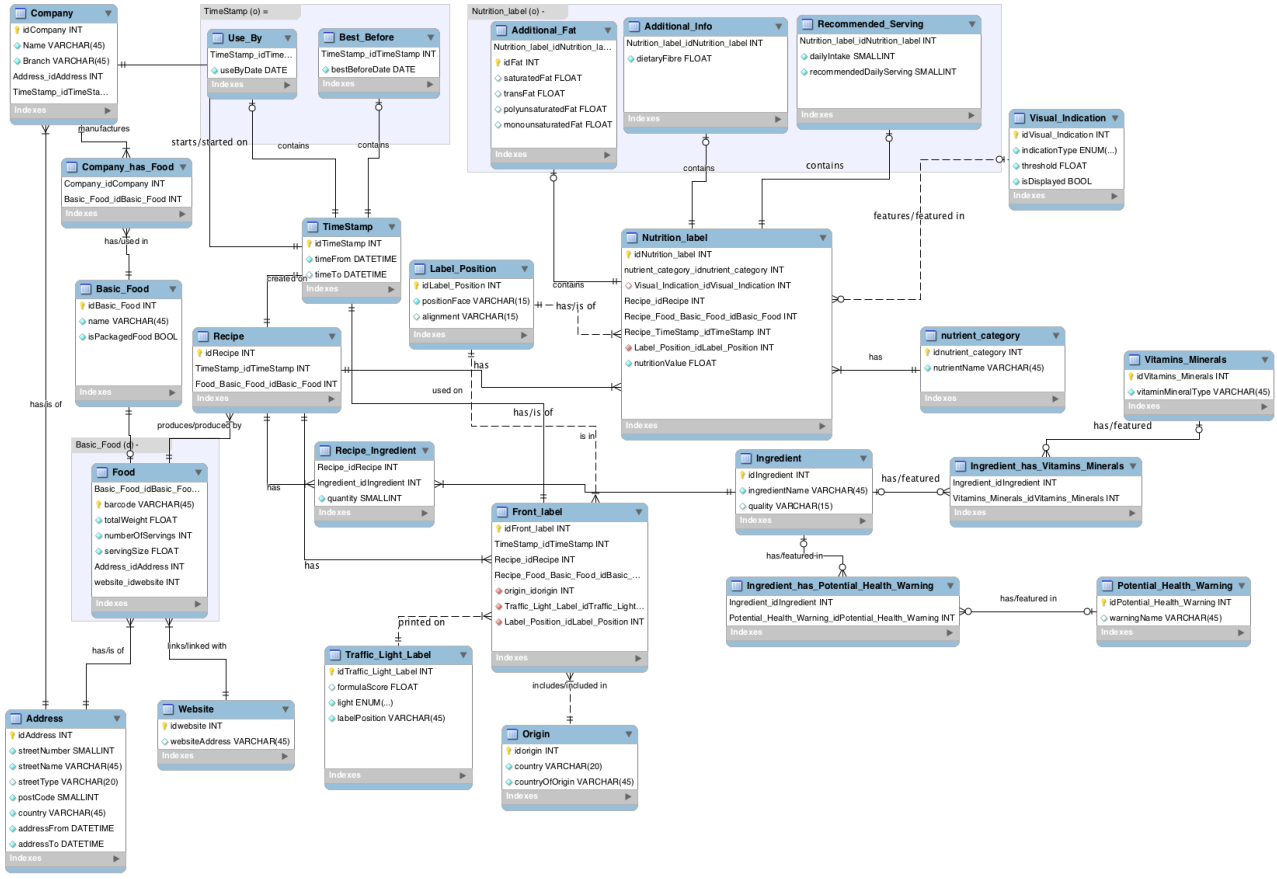
Emily Ha 747184 Database
Systems

Emily Ha 747184 [Database Systems Assignment One]



As S.T.A.R Manufacturers are a multinational food company, the assumption was made that it would have different branches of the company in different countries. Hence, address was factored out as a table. It is assumed that each company would produce different food products with a different recipe containing different quantities of ingredients. This was a direct result of resolving the many to many relationship between food and ingredient. An attribute 'quality' is included in ingredients to accommodate in cases of using cheaper substitutes.

As basic packaged food didn't require labels in Batmania, a sub-type 'Food' was created which extends that 'Basic Food', to have relationships with labels and recipes. Food would also contain details which wasn't specified a particular position in packaging. Instead of normalising a one-to-one relationship with an entity for each detail, I decided to keep all the information stored in the food entity. This would eliminate the need to create an entity for total weight and barcode which would've used unique values to be stored in one entry anyway. Consequently, websites, address of manufacture which would affect multiple products are abstracted as entities.

A timestamp entity was abstracted to record historical data for recipes, to record when the company started, and the time labels were printed and used. It was made the super-type for 'Use-By' and 'Best Before' to inherit from to fulfil the need for either or both to be included.

Emily Ha 747184 Database Systems

To display the full traffic light label, you would have to refer to the traffic light label entity and derive the quantity per 100 grams for each nutrient from the nutrients label. As long as serving size was stored for the food product, we can calculate the wanted nutrients. The visual indicators ('low','medium','high') are abstracted as a look-up table that has an optional one to many relationship(due to certain nutrients requiring an indicator) with the nutrients table. This was done because the Batmanian government changes the thresholds occasionally. Hence, any update changed in the visual indicator entity will only need to be changed once and the results will cascade where it is used. The traffic light entity was kept denormalised to keep the enums for the light indicators. The formula was expected not to change, hence the enums won't experience any change and by keeping them as enums, it would reduce the need for an extra join with a look-up table and speed up queries.

The nutrition label entity is the core of the labelling as it included many foreign primary keys in order to be unique, increase query performance(querying for values that appeared on a label with a specified timeframe) and assumed to only store the values for quantities per serving. Quantities per 100 grams would have to be derived. It was also the super-type that could be extended to include optional information for additional fats, dietary fibre, and recommended serving. Everything that needed to be displayed in the Nutrition Information Panel could be done by doing a single join with 'Food' and the values could be calculated. It was denormalised (many repeated rows would be stored for the same 'recipe id', 'food id', and 'time id') to keep the flexibility in deriving nutrient values. Label positions were also assumed to be stored in case packaging changes over time.