

## **Model attribute explanation**

### **In the Store Model, attributes are:**

Name – name is a char datatype with a maximum of 60 characters because name can be any combination of made-up characters into words, and it's simply not reasonable to have a store name that is longer than 60 characters, even in foreign languages.

Location – I used char field like the name attribute, but with a maximum of 200 characters. This is because a store's location or address can be much longer than its name.

Contact\_email – This attribute I used an email field with a maximum length of 100 characters. I used the email field because it validates email input format using a predefined regex that checks the string for "@" and "." characters.

### **In the Tub model, attributes are:**

Flavour – Just like the name attribute I used char field but with maximum of 30 characters. This is because it's unreasonable to have a flavour name that exceeds 30 characters.

Size – To accommodate various tub sizes, I used a float field, as well as an input validator that checks whether the tub size is less than 0 or greater than 50 litres. This is because it is impossible of having a negative tub size or a tub size larger than 50 litres for an ice cream store.

isVegan – I used a Boolean field because there should only be two possible outcomes when determining whether an ice cream tub is vegan or not. I also set the default value to FALSE to reduce customer complaints in the event that the manager forgot to set the value. In this case, TRUE means the ice cream is vegan, while FALSE means it is not.

isGlutenFree – I used a Boolean field, just like the isVegan attribute. TRUE indicates that the ice cream is gluten-free; FALSE indicates that it is not gluten-free.

Store – I used ForeignKey to account for the ice cream tub's many-to-one relationship with the store, where each ice cream tub can only be assigned to one store and each store can have many ice cream tubs. I also use "on delete=models.CASCADE" options, so that if a specific store is deleted, all related ice cream tubs are also deleted from the database.

## **CRUD operations analysis (create, read, update and delete):**

### **Store managers:**

In order to run a business, store managers must be able to control everything within their designated store. In this case, the store manager should be able to READ their store information so that they are aware of the details of their store, but they should not be able to modify store information for security reasons, particularly for stores that are not under their management. This improves the store's separation of duties, confidentiality, and integrity.

The store manager should have full CRUD access to the ice cream tubs information in their store. This allows them to read, create, delete, or update an ice cream tub as needed.

E.g. when the store manager discovers that the customers in their area prefer chocolate over vanilla, he or she removes the vanilla tub and replaces it with more chocolate tubs of varying sizes.

### **Head offices:**

The head office should have full CRUD access to modify store information. That way they can read store information to see how they are performing, as well as relocate or update store information when necessary.

E.g. Head office discovers that a specific ice cream store is significantly underperforming in comparison to others, so they decide to close or delete that store and relocate to somewhere else. Thus, they need all CRUD access to the store.

Head office, on the other hand, should only have read-only access to each store's ice cream tubs. This is because we don't want head office interfering with each store's day-to-day operations, instead, the managers should handle that. As a result, separation of duties improves.

E.g. the store manager can request the head office for a new tub of ice cream flavour. When the manager receives the new ice cream tubs, he or she can add them to their store. The head office should then be able to see/read about this specific ice cream tubs being added to the designated store.

### **website visitors:**

Website visitors should only be able to read basic store information (name, address, contact) and what ice cream flavours are available for purchase. They should not be allowed to modify any data for security reasons.

## **Benefits of using Django:**

### **Ease of development 1:**

Django provides a lot of built-in “common” functionality that is useful for website developing. The authentication page and admin panels for this ice cream application are available in admin.py, so I don't need to create them from scratch. Instead, I just need to code the necessary admin functions, and Django admin will take care of admin authentication as well as creating and deleting models and saving model data to a SQL database.

### **Ease of development 2:**

Django also supports template inheritance, which allows you to create a base HTML that is shared by all of its child HTML. This eliminating the need to write the same HTML block repeatedly. In this case, I have made homePageBase.html the base HTML, from which any child HTML can inherit and simply put their own variations in the block content section. This allowed me to build this ice cream web application much more quickly and easily.