

Práctica ORM

Hoy tenemos la oportunidad de relacionarnos un poco más con los ORMs. Como sabemos, estos nos ayudan a traducir objetos o entidades a filas en una base de datos.

Sobre la Práctica

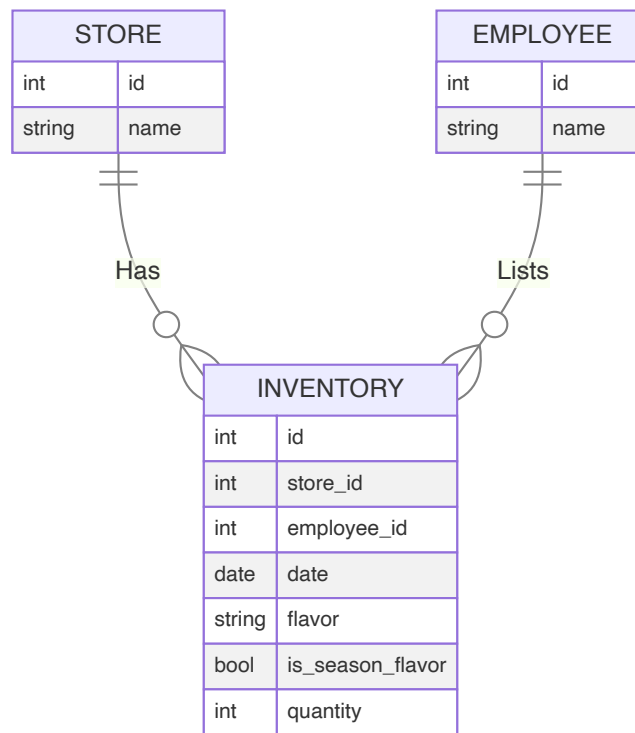
Se les comparte el conjunto de datos `frozono.csv` el cual es el inventario de la cadena de paletas de helado Frozono 🍦. Tal conjunto de datos tiene la siguiente estructura:

Columna	Descripción	Ejemplo
Store	Nombre de la tienda	Store A
Date	Fecha del levantamiento	2023-09-14
Flavor	Sabor del helado	Chocolate
Is Season Flavor	Si es sabor de temporada	No
Quantity	Cantidad de paletas disponibles	20
Listed By	El empleado que registró el inventario	Emily

Se busca que cada equipo desarrolle una REST API que administre el conjunto de datos utilizando un ORM y una base de datos PostgreSQL. La tecnología a escoger por parte de cada equipo (a excepción de la base de datos) queda a su discreción, por lo que pueden considerar el lenguaje y ORM que les resulte más fácil para trabajar.

Sobre las Entidades

El conjunto de datos provisto es en evidencia un único archivo, sin embargo esto ha de ser normalizado, por lo que se les pide que representen el conjunto de datos de la siguiente manera.



Sobre la REST API

Junto con `frozono.csv` es provisto `frozono-open-api.json`. Este archivo es una [OpenAPI Specification \(OAS\)](#), esto quiere decir que pueden utilizar aplicaciones como Postman o Insomnia para generar clientes REST que consuman rutas o *endpoints*. Tales rutas son las esperadas a desarrollar por ustedes en su aplicación, a su vez les permitirá verificar el correcto funcionamiento de su aplicación. Finalmente, se les comparte la ruta `/inventory/upload` que es la ruta en la que tendrán que enviar el archivo CSV y cargar datos.

Sobre la Entrega

Se les solicita entregar los siguientes elementos:

1. Una REST API que utilice un ORM para manejar las entidades ilustradas arriba utilizando clases.
2. Scripts de migraciones de base de datos para generar las entidades a partir de las clases.
3. Una REST API que exponga las rutas listadas en `frozono-open-api.json`.
 1. Es indispensable que la ruta `/inventory/upload` registre los datos compartidos en `frozono.csv`.
4. Las instrucciones para la apropiada corrida de la app, y la carga de datos. Ejemplo:

```
npm run migrate
npm run start
```

2. **Bonus: Habilitar la app mediante Docker y/o Docker Compose.**

5. Todo esto en un repo de Git.