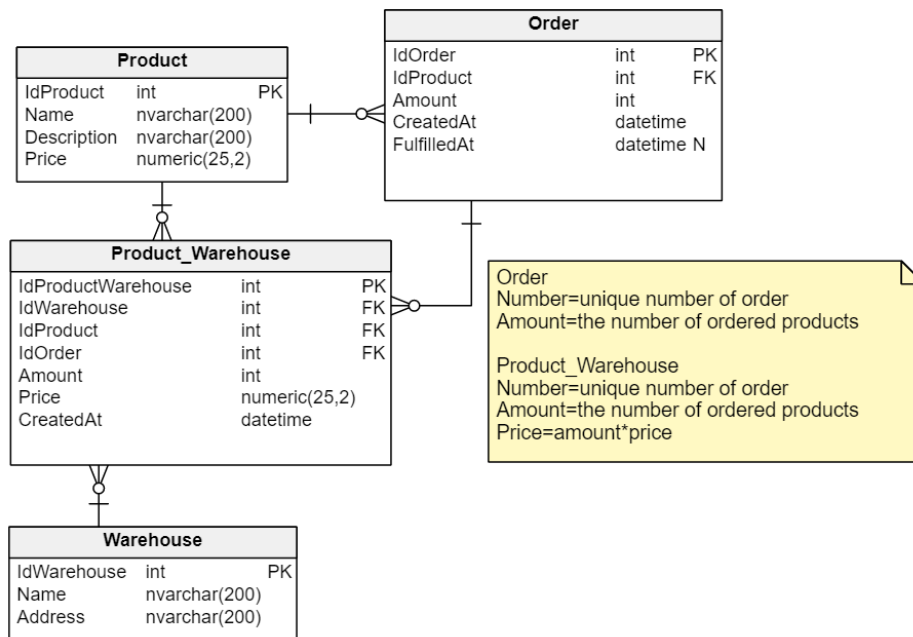


## Ćwiczenia 5

W trakcie niniejszych ćwiczeń ponownie skorzystamy z klas `SqlConnection` i `SqlCommand`. Tym razem logika związana z interakcją z naszą bazą danych będzie jednak nieco bardziej złożona. Tworzymy aplikację dla firmy zajmującej się zarządzaniem stanem magazynu i produktami, które się w nim znajdują. Baza, którą wykorzystujemy zaprezentowana jest poniżej. Ponadto w pliku **create.sql** znajdziecie skrypt, który tworzy tabele i wypełnia je danymi.



1. Stwórz nową aplikację typu REST API
2. Dodaj kontroler o nazwie **WarehousesController**
3. Wewnątrz kontrolera dodaj końcówkę, które będzie odpowiadać na następujące żądanie:
  - a. Kończówka odpowiada na żądanie **HTTP POST** na adres **/api/warehouses**
  - b. Kończówka otrzymuje dane następującej postaci:

```
{  
  "IdProduct": 1,  
  "IdWarehouse": 2,  
  "Amount": 20,  
  "CreatedAt": "2012-04-23T18:25:43.511Z"  
}
```

- c. Wszystkie pola są wymagane. Amount musi być większe niż 0.
- d. Końcówka powinna zrealizować następujący scenariusz działania.

<b>Nazwa</b>	Rejestracja produktu w hurtowni
	<b>Scenariusz główny</b>
1.	Sprawdzamy czy produkt o podanym id istnieje. Następnie sprawdzamy czy hurtownia o podanym id istnieje. Ponadto upewniamy się, że wartość Amount jest większa od 0.
2.	Produkt możemy dodać do hurtowni tylko jeśli w tabeli Order istnieje zlecenie zakupu produktu. Sprawdzamy zatem czy w tabeli Order istnieje rekord z: IdProduct i Amount zgodnym z naszym żądaniem. CreatedAt zamówienia powinno być mniejsze niż CreatedAt pochodzące z naszego żądania (zamówienie/order powinno pojawić się w bazie danych wcześniej niż nasze żądanie).
3.	Sprawdzamy czy przypadkiem to zlecenie nie zostało już zrealizowane. Sprawdzamy czy w tabeli Product_Warehouse nie ma już wiersza z danym IdOrder.
4.	Aktualizujemy kolumnę FullfilledAt zlecenia w wierszu oznaczającym zlecenie zgodnie z aktualną datą i czasem. (UPDATE)
5.	Wstawiamy rekord do tabeli Product_Warehouse. Kolumna Price powinna zawierać pomnożoną cenę pojedynczego produktu z wartością Amount z naszego żądania. Ponadto wstawiamy wartość CreatedAt zgodną z aktualnym czasem. (INSERT)
6.	Jako wynik operacji zwracamy wartość klucza głównego wygenerowanego dla rekordu wstawionego do tabeli Product_Warehouse.
	<b>Scenariusz alternatywny</b>
1a	Produkt/Hurtownia o danym id nie istnieje. Zwracamy błąd 404 z odpowiednią wiadomością.
2a	Nie ma odpowiedniego zlecenia. Zwracamy błąd 404 z odpowiednią wiadomością.
3a	Zlecenie zostało już zrealizowane. Zwracamy odpowiedni kod błędu z wiadomością.

4. Następnie dodaj drugi kontroler **Warehouses2Controller** z końcówką odpowiadającą na żądania wysłane na adres **HTTP POST /api/warehouses2**
  - a. Końcówka realizuje tą samą logikę, ale w tym wypadku uruchamiana jest procedura składowana (załączona w pliku **proc.sql**).
5. Pamiętaj o wstrzykiwaniu zależności, odpowiednich nazwach, kodach zwrotnych http
6. Spróbuj wykorzystać metody wykorzystujące programowanie asynchroniczne i `async/await`.