

## 데이터베이스 프로젝트 1

20161277 김재훈

2025-04-21

이 ER schema 에는 Store, Product, Vendor, Customer, Transaction, Transaction\_Item, Inventory 의 일곱 개 entity 가 있습니다. Store 가 Inventory 를 Maintain 하며 Transaction 을 Host 하고, Customer 가 Transaction 을 Place 하고, Transaction 은 Transaction\_Item 을 Contain 하고, Vendor 가 Product 를 Supply 하고, Product 는 Inventory 에 StoredAs 되어 있으며 Transaction\_Item 에도 IncludedIn 되어 있습니다. 각 entity 는 그 entity 를 대표하는 ID 값이나 그러한 ID 값들의 composite 을 primary key 로 갖습니다.

대부분의 entity 들은 다양한 실제 시나리오를 반영해 partial relationship 을 갖습니다. 예를 들어, Store 가 Inventory 를 갖기 전에 먼저 DB 상에 존재해야 할 수 있습니다. 하지만 total relationship 이 필요한 경우도 있습니다. 모든 Transaction 은 Store 에서 일어나야 한다면, 모든 Inventory 는 해당하는 Store 와 Product 가 있어야 하는 점이 그렇습니다. Transaction\_Item 은 Transaction 이 있어야만 성립하는 weak entity 입니다. 모든 relationship 은 1-to-many 의 형태를 갖습니다. 하나의 Store 가 여러 Transaction 을 가지고, 하나의 Transaction 이 여러 Transaction\_Item 을 갖는 식입니다.

### Sample Queries:

1. Which stores currently carry a product (by UPC, name, or brand), and how much inventory do they have?

Inventory entity 를 이용해 확인할 수 있습니다. UPC 가 주어진다면 Inventory entity 만으로 어느 StoreID 를 가진 store 가 얼마나 해당 product 를 갖고 있는지 CurrentStock 을 통해 알 수 있고, product 의 brand 가 주어진다면 Product entity 를 Inventory entity 와 join 함으로써 알 수 있습니다.

2. Which products have the highest sales volume in each store over the past month?

Transaction, Transaction\_Item, Product, Store entity 를 통해 알 수 있습니다. Transaction 의 DateTime 을 통해 지난 달만 남기고 필터링 하고, Transaction\_Item 과 join 을 해 ProductID 와 StoreID 별 Quantity 를 알 수 있습니다. 여기서 Product entity 를 이용해 Quantity 와 Price 를 곱해 sales volume 을 계산하면 질문의 답을 알 수 있습니다.

3. Which store has generated the highest overall revenue this quarter?

Transaction entity 에서 DateTime 으로 지난 분기만 필터링하고, StoreID 로 group by 를 한 뒤 TotalAmount attribute 를 모두 더해주면 store 별 total revenue 를 알 수 있습니다.

4. Which vendor supplies the most products across the chain, and how many total units have been sold?

Vendor 와 Product entity 를 join 해서 어느 vendor 가 가장 많은 product 를 공급하는지 알 수 있습니다.

Transaction\_Item, Product, Vendor entity 를 join 하고 VendorID 에 대해 group by 를 함으로써 질문의 답을 알 수 있습니다.

5. Which products in each store are below reorder threshold and need restocking?

Inventory entity 에서 CurrentStock 과 ReorderThreshold attribute 를 비교해서 어느 product 가 어느 store 에서 restocking 이 필요한지 알 수 있습니다. 이렇게 각 store 가 다른 ReorderThreshold 를 가질 때에도 대응할 수 있습니다.

6. List the top 3 items that loyalty program customers typically purchase with coffee.

Customer, Transaction, Transaction\_Item, Product entity 를 통해 알 수 있습니다. Transaction\_Item 과 Product 를 join 해 product 의 이름에서 커피를 찾고, 같은 TransactionID 를 공유하는 UPC 들 중 Transaction 에서 CustomerID 를 갖고 있는 것들을 찾으면 됩니다.

7. Among franchise-owned stores, which one offers the widest variety of products, and how does that compare to corporate-owned stores?

Store, Inventory entity 를 join 함으로써 store 의 OwnershipType 에 따른 product 의 다양성을 확인할 수 있습니다.