

E-Ticaret Veritabanı – Teknik Rapor

1) Amaç ve Kapsam

Bu proje, bir **e-ticaret senaryosu** için ilişkisel bir veritabanı tasarlamayı, tabloları örnek verilerle doldurmayı ve tablolar arasındaki ilişkileri çeşitli **SELECT** ve **JOIN** sorguları ile doğrulamayı amaçlamaktadır.

Projede ayrıca tablo yapıları örnek verilerle test edildi ve ilişkiler **SELECT** ile **JOIN** sorguları kullanılarak doğrulandı.

2) Veritabanı Şeması ve Tablolar

Veritabanında altı temel tablo bulunuyor. Her tablo, farklı türde verileri saklamak üzere tasarlandı:

- **Customers:** Müşteri bilgilerini (ad, soyad, e-posta, şehir) içerir ve customer_id alanı birincil anahtar (**PK**) olarak tanımlanmıştır.
- **Orders:** Sipariş bilgilerini saklar. Her sipariş, customer_id alanı üzerinden **Customers** tablosuna bağlanarak **foreign key (FK)** ilişkisinde tutulur.
- **Products:** Ürün adlarını ve fiyat bilgilerini içerir. Ayrıca category_id üzerinden **Categories** tablosuna bağlanır.
- **Categories:** Ürün kategorilerini tanımlar ve category_id birincil anahtar (**PK**) olarak kullanılır.
- **Payments:** Her siparişe ait ödeme bilgilerini içerir. order_id alanı **Orders** tablosuna bağlanır ve **1:1** ilişki sağlamak için **UNIQUE** kısıtlaması eklenmiştir.
- **OrderItems:** Siparişlerdeki ürün detaylarını saklayan ara tablodur. **Orders** ve **Products** tabloları arasında **M:N** ilişkiyi kurar.

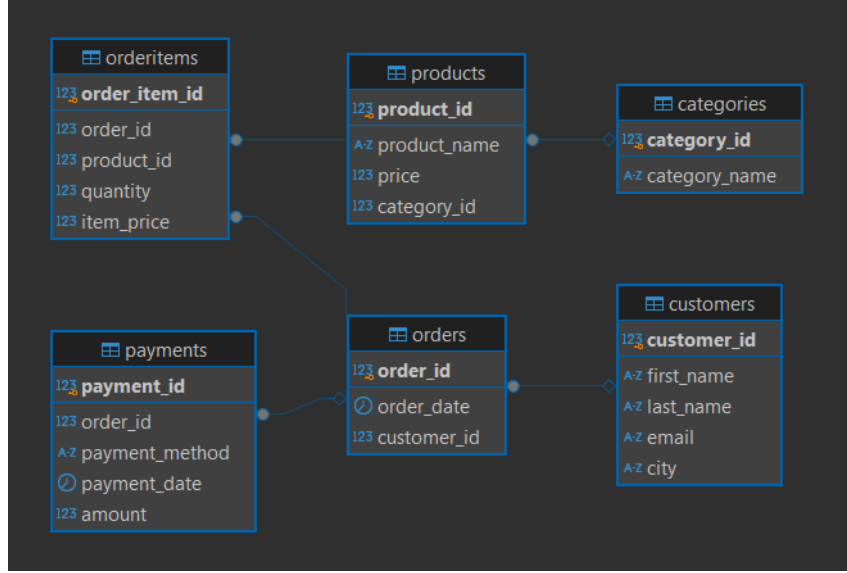
Not: Şema, PK ve FK tanımlamalarıyla birlikte tablolar arasındaki **1:N** ve **1:1** ilişkileri açık bir şekilde yansıtmaktadır.

3) Tablolar Arasındaki İlişkiler

Kurulan veritabanı, tablolar arasında şu ilişkileri içeriyor:

- **Customers (1) → (N) Orders**
Bir müşteri birden fazla sipariş verebilir.
- **Orders (1) → (1) Payments**
Her siparişin yalnızca bir ödemesi vardır.
- **Categories (1) → (N) Products**
Her kategori birden fazla ürünü kapsar.
- **Orders (1) → (N) OrderItems** ve **Products (1) → (N) OrderItems**
Siparişler ve ürünler, **OrderItems** tablosu üzerinden **M:N** ilişkiyle bağlanır.

Bu yapı, proje için hazırlanan **ER diyagramında** görsel olarak açıkça gösterilmektedir.



4) Örnek Veriler

Tabloların doğru çalışıp çalışmadığını görmek için her tabloya **INSERT** komutlarıyla örnek veri eklenmiştir.

Ürünler, müşteriler, siparişler, ödemeler ve sipariş kalemleri için veri girilerek sistemin uçtan uca test edilmesi sağlandı.

5) Doğrulama Sorguları

Veritabanının doğru çalıştığını test etmek için çeşitli sorgular hazırlandı:

- **Temel kontroller:**
SELECT * sorguları ile tabloların içerikleri kontrol edildi.
- **JOIN testleri:**
 - **Customer ↔ Orders:** Müşteri ve sipariş bilgileri birleştirilerek müşteri bazında siparişler listelendi.
 - **Products ↔ Categories:** Ürün ve kategori bilgileri eşleştirilerek hangi ürünün hangi kategoriye ait olduğu gösterildi.
 - **Orders ↔ Payments:** Sipariş ve ödeme bilgileri birleştirilerek ödeme detayları doğrulandı.

- Ayrıca **OrderItems** tablosu üzerinden **Orders** ve **Products** tablolarını birleştirerek **çoktan çoğa** ilişkinin doğru çalıştığı test edilmiştir.

6) Çalıştırma Adımları

1. Yeni bir veritabanı veya şema oluşturun (isteğe bağlı).
2. ecommerce_project.sql dosyasını **pgAdmin** veya **DBeaver** üzerinden çalıştırın.
Kod sırası tablo oluşturma, veri ekleme ve doğrulama sorgularını kapsar.
3. ER diyagramını veritabanı aracından otomatik oluşturup tablolar arası bağlantıları görsel olarak kontrol edebilirsiniz.

7) Değerlendirme Kriterlerine Uyum

- **Veritabanı tasarımı:** PK ve FK'ler doğru tanımlanarak tablolar arasındaki ilişkiler kurulmuştur.
- **Veri türleri:** VARCHAR, DATE ve DECIMAL seçimleri senaryoya uygun şekilde belirlenmiştir.
- **Örnek veri girişi:** Her tabloya örnek veriler eklenmiş, sistem uçtan uca test edilmiştir.
- **Sorgular:** Hazırlanan SELECT ve JOIN sorguları ile tablolar arası ilişkiler başarıyla doğrulanmıştır.

8) İyileştirme Önerileri

Proje, temel gereksinimleri karşılamakla birlikte aşağıdaki geliştirmeler eklenebilir:

- **Veri bütünlüğü:**
email alanına **UNIQUE NOT NULL** kısıtı eklenebilir.
- **Kontroller:**
price, amount ve item_price gibi sayısal alanlara **CHECK (>=0)** kısıtları tanımlanabilir.
- **Veri silme senaryoları:**
Yabancı anahtarlara **ON DELETE CASCADE** eklenerek bağlı verilerin otomatik temizlenmesi sağlanabilir.
- **Performans optimizasyonu:**
Sık kullanılan FK alanlarına (ör. orders.customer_id, orderitems.order_id) **index** eklenebilir.