

Online Alışveriş Platformu — Veri Tabanı Tasarımı

1. Giriş

Bu projenin amacı, bir online alışveriş platformu için ilişkisel bir veritabanı tasarlamak ve temel işlevlerini desteklemektir.

Proje kapsamında; müşterilerin kayıt, sipariş ve ödeme süreçleri, ürünlerin kategori ve stok yönetimi, satıcıların ürün yönetimi ve raporlama işlemleri dikkate alınmıştır.

Veritabanı, platformun performanslı, tutarlı ve ölçeklenebilir bir şekilde çalışmasını sağlamak üzere tasarlanmıştır.

Bu rapor, veritabanı tablolarını, ilişkileri, veri tiplerini ve örnek veri ekleme adımlarını açıklamakta, ayrıca raporlama sorguları ve karşılaşılan problemleri detaylandırmaktadır.

2. Tasarım

Veritabanı tasarımı, platformun iş süreçlerine uygun olarak gerçekleştirilmiştir. Tasarımın temel amacı; veri tekrarını azaltmak, veri tutarlılığını sağlamak ve sorgu performansını artırmaktır.

2.1 Tablolar

- Müşteriler (Musteriler): Kullanıcı bilgilerini tutar (Ad, Soyad, Email vb.).
- Siparişler (Siparisler): Her siparişin bilgilerini saklar (Toplam Tutar, Ödeme Türü vb.).
- Sipariş Detayları (SiparisDetaylari): Siparişlerdeki ürünleri ve adetlerini içerir.
- Ürünler (Urunler): Platformdaki ürün bilgilerini ve stok durumunu tutar.
- Kategoriler (Kategoriler): Ürünlerin kategorilerini tanımlar.
- Satıcılar (Saticilar): Ürünleri platforma ekleyen satıcı bilgilerini içerir.

2.2 Kritik İlişkiler

- Bir müşteri birden fazla sipariş verebilir (1:N).
- Bir sipariş birden fazla üründen oluşabilir (1:N).

- Her ürün bir kategoriye ve bir satıcıya aittir (1:1 veya N:1).
- İlişkilerde ON DELETE CASCADE ve ON UPDATE CASCADE tercih edilmiştir, böylece kayıt silme veya güncelleme işlemleri ilişkili tablolarda otomatik olarak yansır.

3. Uygulama

Veritabanı PostgreSQL üzerinde tasarlanmış olup, MySQL'de de benzer şekilde uygulanabilir.

3.1 Veri Tipleri

- Kimlik alanları: INT
- Metin alanları: NVARCHAR
- Tarih alanları: DATE / DATETIME
- Fiyat ve tutar: DECIMAL(10,2)

3.2 Örnek Veri Ekleme

-- Müşteri ekleme

```
INSERT INTO Musteriler (Ad, Soyad, Email)  
VALUES ('Ahmet', 'Yılmaz', 'ahmet@example.com');
```

-- Sipariş ekleme

```
INSERT INTO Siparisler ( ToplamTutar, OdemeTuru)  
VALUES ( 500.00, 'Kredi Kartı');
```

4. Raporlama Sorguları

- Aylara göre sipariş sayısı: Dönemsel satış analizi.
- Kategori bazlı toplam satışlar: Hangi kategorilerin daha çok gelir getirdiğinin analizi.
- Hiç satılmamış ürünler: Depoda bekleyen ürünlerin tespiti.
- Hiç sipariş vermemiş müşteriler: Aktif olmayan kullanıcıların belirlenmesi.

- En çok satılan ürünler: Trend ürünlerin tespiti.
- En yüksek ciroya sahip satıcılar: En çok kazandıran satıcıların analizi.

5. Karşılaşılan Problemler

- Aynı e-posta ile birden fazla müşteri kaydı yapılması → Unique constraint hatası.
- Stok güncellemelerinde hatalı adet girilmesi → Negatif stok tutarsızlığı.
- Çok sayıda ürün ve sipariş sorgusunda performans düşüklüğü → indeks ihtiyacı ortaya çıktı.
- DELETE işlemlerinde foreign key hataları → sipariş silindiğinde detay kayıtlarının da silinmesi gerekti.