Öğrenci Adı Soyadı :Erol Yeşilyurt

Öğrenci Numarası: 170422058

T.C. MARMARA ÜNİVERSİTESİ TEKNOLOJİ FAKÜLTESİ

2024-2025 EĞİTİM ÖĞRETİM YILI

GÜZ DÖNEMİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

VERİTABANI YÖNETİM SİSTEMLERİ DERSİ

PROJE-FİNAL DÖKÜMANI

|  |
| --- |
| Tüm sorular soru metnini altındaki alanlara cevaplandırılacaktır. Bu belgenin formatını bozmayınız, sadece gerekli alanları doldurunuz. |

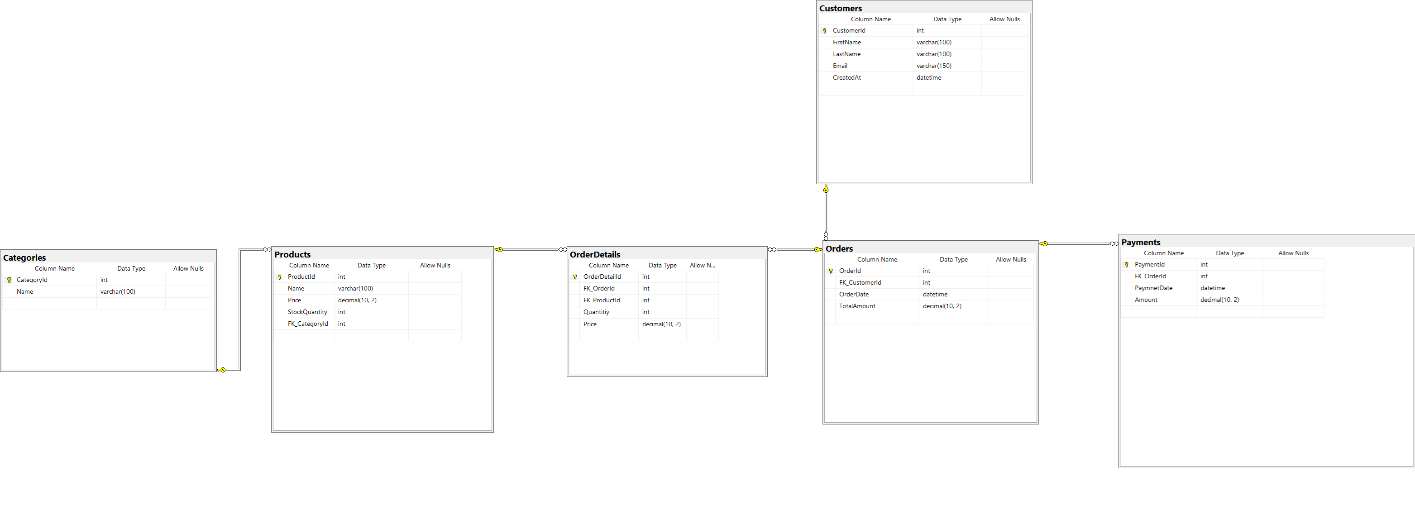
Gerçekleştirdiğiniz veri tabanı projesi için grup arkadaşlarınızın isimlerini yazınız ve projenize ait veri tabanı/diğer yazılım bileşenleri hakkında bilgi veriniz. (5 p)

Proje arkadaşım Erol Yeşilyurt ile birlikte MSSQL ile birlikte .net core web API teknolojisi ile birlikte front end kısmında angular kullanılmıştır. Toplamda 6 tablo ile birlikte ilişkisel tablolar kullanılarak projemiz oluşturulmuştur.

Gerçekleştirdiğiniz veri tabanı projesi için proje dokümanınızı ve dosyalarınızı içeren herkese açık github bağlantılarınızı paylaşınız. (5 p)

Github.com/1yusufdogan

Gerçekleştirdiğiniz projenin amacını detaylı bir şekilde açıklayınız. (10 p)  
  
Gerçekleştirdiğimiz bu proje bir e ticaret sitesinin database tarafında tablolar ve tabloların birbiri ile olan ilişkisi , tabloların taşıması gereken özellikler ve nitelikler , veri tabanı ile birlikte back and kısmının birbiri ile olan bağlantısı ve back end kısmında olması gereken özellikler , aynı zamanda veri tabanı içerisinde bulunan verilerin güvenli ve sağlıklı bir şekilde kullanıcıya ulaşmasını ve kolay kullanabilir bir şekilde müşteri / kullanıcı dostubir projedir. Bu proje, bir e-ticaret platformunun müşteri, ürün, sipariş ve ödeme bilgilerini yönetmek için bir veritabanı tasarlamayı hedeflemektedir. Temel amaç, satış sürecini dijital ortamda düzenlemek ve müşterilere daha iyi bir alışveriş deneyimi sunmaktır. Aynı zamanda ürün envanteri yönetimi, sipariş takibi ve ödeme işlemleri gibi işlevlerin etkin bir şekilde yürütülmesi amaçlanmaktadır.

Tasarladığınız veri tabanı mimarisinde hangi tablo ve ilişkileri kullanıldığınızı açıklayınız. (10 p)  


Veri tabanı ER (Entity Relationship) diagramının bilgisayar ortamında çizilmiş halini paylaşınız. (Ara raporda eksik kısımlar bu raporda giderilmelidir ve ER çizme programlarından faydalanıbilir. Elle çizim, çizip fotoğrafını çekme vb. kabul edilmeyecektir.) (10 p)

Herhangi iki tablonuz için DDL (create) kodları yazılmalıdır. (10 p)

-- Customers tablosu

CREATE TABLE Customers (

CustomerId INT PRIMARY KEY,

FirstName VARCHAR(100),

LastName VARCHAR(100),

Email VARCHAR(100),

CreatedAt DATETIME

);

-- Orders tablosu

CREATE TABLE Orders (

OrderId INT PRIMARY KEY,

CustomerId INT,

OrderDate DATETIME,

TotalAmount DECIMAL(10, 2),

FOREIGN KEY (CustomerId) REFERENCES Customers(CustomerId)

);

5 adet DML (update, insert, delete) içeren kodları yazılmalıdır. (10 p)

-- 1. Insert - Yeni müşteri ekleme

INSERT INTO Customers (CustomerId, FirstName, LastName, Email, CreatedAt)

VALUES (1, 'John', 'Doe', 'john.doe@example.com', GETDATE());

-- 2. Insert - Yeni sipariş ekleme

INSERT INTO Orders (OrderId, CustomerId, OrderDate, TotalAmount)

VALUES (1, 1, GETDATE(), 250.00);

-- 3. Update - Müşteri e-posta güncelleme

UPDATE Customers

SET Email = 'john.d.newemail@example.com'

WHERE CustomerId = 1;

-- 4. Delete - Sipariş silme

DELETE FROM Orders

WHERE OrderId = 1;

-- 5. Update - Sipariş tutarı güncelleme

UPDATE Orders

SET TotalAmount = 300.00

WHERE OrderId = 1;

Projenize ait kendi belirlediğiniz 10 adet SQL sorgusu yazınız, sorguların amacını ve sonuç çıktısını da lütfen ekleyiniz. (Açıklama: Sorgular Select deyimleri ve gruplama fonksiyonlarını HAVING deyimini (min, max, avg, count gibi) ve join deyimlerini (en az iki tablo ile birleştirme sorgusu) içerecek şekilde basitten karmaşığa doğru gitmelidir. Proje sunum anında veri tabanınıza ait sorular SQL ortamında gösterilecek ve açıklanacaktır. Raporunuzda ise sorgular, sorguların cevap ve sonuçlarının ekran görüntüsü olarak paylaşılması beklenmektedir. (10 p) Örnek:

1. Tüm müşterileri listeleme:

SELECT \* FROM Customers;

Amaç: Veritabanındaki tüm müşteri bilgilerini getirir.

2. Toplam sipariş miktarını ve tarihini listeleme:

SELECT OrderId, OrderDate, TotalAmount

FROM Orders;

Amaç: Sipariş bilgilerini ve tutarlarını gösterir.

3. Müşterilerin toplam sipariş sayısını gruplama ile gösterme:  
metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

SELECT CustomerId, COUNT(\*) AS TotalOrders

FROM Orders

GROUP BY CustomerId;

Amaç: Her müşteri için verilen sipariş sayısını bulur.

4. 1000 TL’den büyük siparişleri gösterme:

SELECT OrderId, TotalAmount

FROM Orders

WHERE TotalAmount > 1000;

Amaç: 1000 TL üstü siparişleri listeler.

5. Müşteri ve sipariş bilgilerini birleştirme (Join):

SELECT Customers.FirstName, Customers.LastName, Orders.OrderDate, Orders.TotalAmount

FROM Customers

JOIN Orders ON Customers.CustomerId = Orders.CustomerId;

Amaç: Müşteri adı ile sipariş tarihlerini ve tutarlarını birleştirir.

6. Sipariş sayısı 5’ten fazla olan müşteriler:

SELECT CustomerId, COUNT(\*) AS OrderCount

FROM Orders

GROUP BY CustomerId

HAVING COUNT(\*) > 5;

Amaç: 5’ten fazla sipariş veren müşterileri listeler.

7. Ürün ve kategori bilgilerini listeleme (Join):

SELECT Products.Name AS ProductName, Categories.Name AS CategoryName

FROM Products

JOIN Categories ON Products.CategoryId = Categories.CategoryId;

Amaç: Ürün ve kategorilerinin ilişkili listesini verir.

8. Toplam stok miktarını ve ortalama fiyatı hesaplama:

SELECT SUM(StockQty) AS TotalStock, AVG(Price) AS AveragePrice

FROM Products;

Amaç: Tüm ürünlerin toplam stok miktarını ve ortalama fiyatını gösterir.

9. Ödenen toplam tutarı hesaplama:

SELECT SUM(Amount) AS TotalPayments

FROM Payments;

Amaç: Tüm ödemeler için toplam tutarı hesaplar.

10. En son siparişi veren müşteriyi bulma:

SELECT TOP 1 Customers.FirstName, Customers.LastName, Orders.OrderDate

FROM Customers

JOIN Orders ON Customers.CustomerId = Orders.CustomerId

ORDER BY Orders.OrderDate DESC;

Amaç: En son siparişi veren müşteriyi bulur.

metin, ekran görüntüsü, yazılım, web sayfası içeren bir resim

Açıklama otomatik olarak oluşturuldu

Eğer gerçekleştirmiş iseniz, veri tabanı bağlama ve uygulama geliştirme aşamalarınızı kısaca açıklayarak, kullanıcı ara yüz ekranından bir örnek veriniz. Ve geliştirdiğiniz ara yüzü anlatınız. (10 p)  
  
Ara yüz örneği:

• Login Page: Kullanıcı adı ve şifre alanları bulunan, giriş işlemi sonrası kullanıcıyı ana sayfaya yönlendiren basit bir ekran.

• Başvuru Listesi Sayfası: Giriş yapmış kullanıcının oluşturduğu başvuruların listelendiği bir sayfa.

Eğer veri tabanı bağlama işlemini gerçekleştirmemiş iseniz VTYS sistemlerinde Transaction nedir açıklayınız ve çalışmanızdan bir Transaction örneği veriniz. (10 p)

Transaction, bir grup veri tabanı işleminin bir bütün olarak başarıyla tamamlanmasını sağlayan bir işlem kümesidir. Örneğin, sipariş oluşturma ve ödeme işlemi bir transaction içinde yapılabilir.

BEGIN TRANSACTION;

INSERT INTO Orders (OrderId, CustomerId, OrderDate, TotalAmount)

VALUES (2, 2, GETDATE(), 500.00);

INSERT INTO Payments (PaymentId, OrderId, PaymentDate, Amount)

VALUES (2, 2, GETDATE(), 500.00);

COMMIT;

View nedir açıklayınız ve bir adet view, bir adet saklı yordam (Stored Procedute) ifadesine ait SQL deyimlerinin sorgusunu ve cevabını yazınız.  
(10 p)

View veritabanında sanal teblo gibi davranan veritababnı nesnesidir.Gerektiğinde sorgulardaki sonuçları tablo gibi kullanmamıza yarar.View bu özellikleriyle veri güvenliği ve tekrar kullanılabilirliği sağlar.

View oluşturma örneği:

CREATE VIEW ActiveEmployees AS

SELECT EmployeeID, FirstName, LastName

FROM Employees

WHERE Status = 'Active';

Employees tablosundaki aktif calısanları içeren bir sorgu.Active employees görünümünü sorgularda kullanmak mümkündür.

SELECT \* FROM ActiveEmployees;

Bu sorgu ile onceki sorguda oluşturulan ActiveEmployees tablosunu kullanmamızı sağlar.

Saklı yordam oluşturma:

CREATE PROCEDURE GetActiveEmployees

AS

BEGIN

SELECT EmployeeID, FirstName, LastName

FROM Employees

WHERE Status = 'Active';

END;

Bu sorgu, GetActiveEmployees adında bir saklı yordam yaratır ve çağrıldığında tüm aktif çalışanları getirir.

Saklı yordam çağrısı;

EXEC GetActiveEmployees;

Bu sorgu, oluşturulan GetActiveEmployees saklı yordamını çalıştırır ve sonuç olarak aktif çalışanların bilgilerini alır.