Öğrenci Adı Soyadı : Yağmur Topçu

Öğrenci Numarası: 121521031

T.C. MARMARA ÜNİVERSİTESİ FEN FAKÜLTESİ İSTATİSTİK BÖLÜMÜ

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

Sude YILMAZ 121520053 - Merve ŞEREF 121522824 - Yağmur TOPÇU 121521031

Biz SQL Server Management Studio üzerinden veri tabanı oluşturup verilerimizi tablolayıp gerekli sorguları yaptık.

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

1. **Gerçekleştirdiğiniz projenin amacını detaylı bir şekilde açıklayınız. (10 p)**

Müzik mağazalarının ürün, müşteri ve satış işlemlerini etkili bir şekilde yönetmelerine olanak sağlayan kapsamlı bir veri tabanı uygulaması geliştirmektir. Bu uygulama, müzik mağazasına ait sanatçı, albüm, şarkı, müşteri ve fatura gibi bilgilerin saklanmasını sağlamaktadır. Aynı zamanda kullanıcıların yeni albümler veya şarkılar eklemesine, müşterilerin bilgilerini güncellemesine ve satış işlemlerini takip etmesine imkân tanımaktadır.

Çalışma, kullanıcıların şarkıları türlerine göre sınıflandırmasına, satış raporlarını görüntülemesine ve müşteri alışkanlıklarını analiz etmesine olanak sağlamaktadır. Böylece, yöneticiler satış stratejilerini belirleyebilir, en çok tercih edilen müzik türlerini analiz edebilir ve genel performans hakkında bilgi sahibi olabilir.

Bu sayede, müzik mağazasıyla ilgili tüm işlemler düzenli bir veri yapısıyla yönetilirken, verimlilik artırılmış ve karar alma süreçleri kolaylaştırılmıştır.

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

**Tablolar**

a. Customers: CustomerID, FirstName, LastName, Company, Address, City, State, Country, PostalCode, Phone, Fax, Email, SupportRepID

b. Employees: EmployeeID, FirstName, LastName, Title, ReportsTo, BirthDate, HireDate, Address, City, State, Country, PostalCode, Phone, Fax, Email

c. Invoices: InvoiceID, CustomerID, InvoiceDate, BillingAddress, BillingCity, BillingState, BillingCountry, BillingPostalCode, Total

d. InvoiceLines: InvoiceLineID, InvoiceID, TrackID, UnitPrice, Quantity

e. Tracks: TrackID, Name, AlbumID, MediaTypeID, GenreID, Composer, Milliseconds, Bytes, UnitPrice

f. Albums: AlbumID, Title, ArtistID

g. Artists: ArtistID, Name

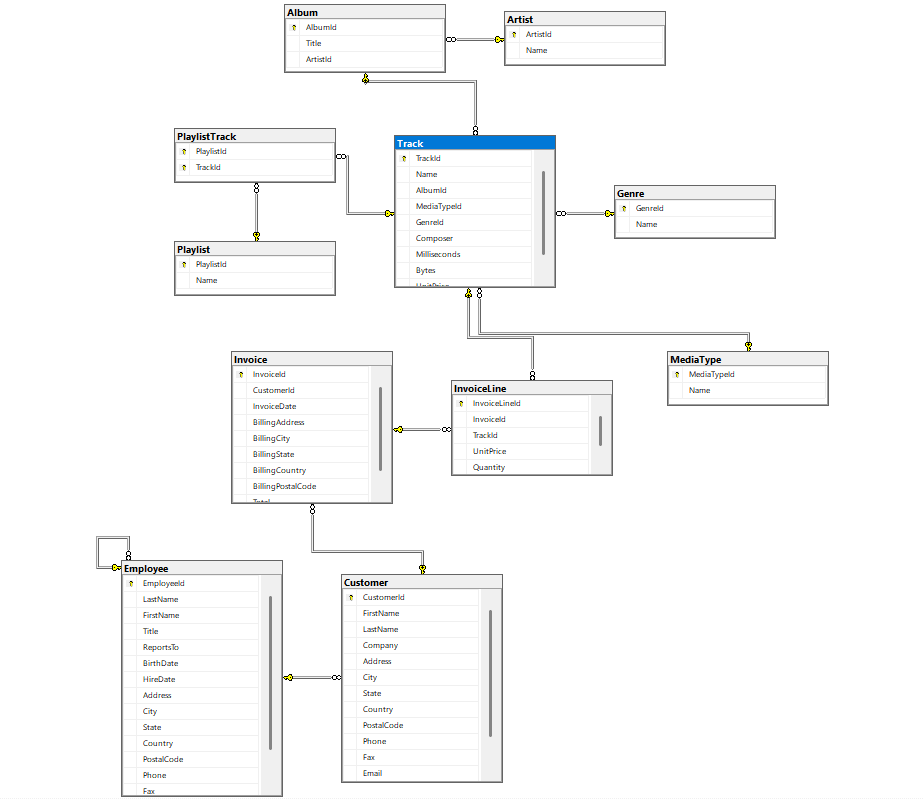
h. Genres: GenreID, Name

i. MediaTypes: MediaTypeID, Name

j. Playlists: PlaylistID, Name

k. PlaylistTrack: PlaylistID, TrackID

**Veri tabanı Normalizasyon aşamalarını, 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)**



Veri tabanımız 1NF, 2NF ve 3NF kurallarını sağlamaktadır. Veri tabanımız normalizasyona uygundur.

**İlişkiler (Relationships)**

Customers → Invoices (Müşteriler → Faturalar)

Her müşteri birden fazla fatura oluşturabilir.

İlişki türü: 1:N (bire-çok)

Invoices → InvoiceLines (Faturalar → Fatura Satırları)

Her fatura, birden fazla fatura satırı içerebilir.

İlişki türü: 1:N

InvoiceLines → Tracks (Fatura Satırları → Müzik Parçaları)

Her fatura satırı, bir müzik parçasına aittir.

İlişki türü: N:1 (çok-bire)

Tracks → Albums (Müzik Parçaları → Albümler)

Her müzik parçası bir albüme aittir.

İlişki türü: N:1

Albums → Artists (Albümler → Sanatçılar)

Her albüm bir sanatçıya aittir.

İlişki türü: N:1

Tracks → MediaTypes (Müzik Parçaları → Medya Türleri)

Her parça bir medya türüne sahiptir.

İlişki türü: N:1

Tracks → Genres (Müzik Parçaları → Türler)

Her parça bir müzik türüne aittir.

İlişki türü: N:1

Playlists → PlaylistTrack (Çalma Listeleri → Çalma Listesi-Parça)

Her çalma listesi birden fazla parçaya sahip olabilir.

İlişki türü: 1:N

PlaylistTrack → Tracks (Çalma Listesi-Parça → Müzik Parçaları)

Her satır, bir parçayı temsil eder.

İlişki türü: N:1

Employees → Customers (Çalışanlar → Müşteriler)

Her müşteri bir çalışan tarafından desteklenir.

İlişki türü: 1:N

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

CREATE TABLE [dbo].[Album]

(

[AlbumId] INT NOT NULL,

[Title] NVARCHAR(160) NOT NULL,

[ArtistId] INT NOT NULL,

CONSTRAINT [PK\_Album] PRIMARY KEY CLUSTERED ([AlbumId])

);

CREATE TABLE [dbo].[Customer]

(

[CustomerId] INT NOT NULL,

[FirstName] NVARCHAR(40) NOT NULL,

[LastName] NVARCHAR(20) NOT NULL,

[Company] NVARCHAR(80),

[Address] NVARCHAR(70),

[City] NVARCHAR(40),

[State] NVARCHAR(40),

[Country] NVARCHAR(40),

[PostalCode] NVARCHAR(10),

[Phone] NVARCHAR(24),

[Fax] NVARCHAR(24),

[Email] NVARCHAR(60) NOT NULL,

[SupportRepId] INT,

CONSTRAINT [PK\_Customer] PRIMARY KEY CLUSTERED ([CustomerId])

);

1. **5 adet DML (update, insert, delete) içeren kodları yazılmalıdır. (10 p)**

Select ile ArtistID değeri değiştirme:

DECLARE @YeniArtistId INT;

SET @YeniArtistId = 5;

SELECT \* FROM Artist WHERE ArtistId = @YeniArtistId;

Insert ile yeni müşteri ekleme:

INSERT INTO Customer (CustomerId,FirstName, LastName, Company, Address, City, State,Country,PostalCode,Phone,Fax,Email, SupportRepId)

VALUES(60,'Sude','Yılmaz','AppleInc.','Kadikoy','İstanbul',NULL, 'Turkey',34744,+905344562578,02164563553 ,'sude\_\_yilmaz@hotmail.com', 3);

Update ile e-mail adresini değiştirme:

UPDATE Customer

SET Email = 'sude.yilmaz@icloud.com'

WHERE CustomerId = 60;

Select ile Seçilen müşterinin istenilen bilgilerini çekme:

SELECT FirstName, LastName, Email, City

FROM Customer

WHERE CustomerId = 60;

Delete ile eklenen satırı silme:

DELETE FROM Customer

WHERE CustomerId = 60;

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. (20 p)**

metin, sayı, numara, yazı tipi, yazılım içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil 1 5 ten fazla şarkı içeren albüm isimleri ve şarkı sayısı

**metin, ekran görüntüsü, yazılım, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**

Şekil Sanatçı Adı ve Albüm Adını veriyor.

metin, ekran görüntüsü, ekran, görüntüleme, sayı, numara içeren bir resim

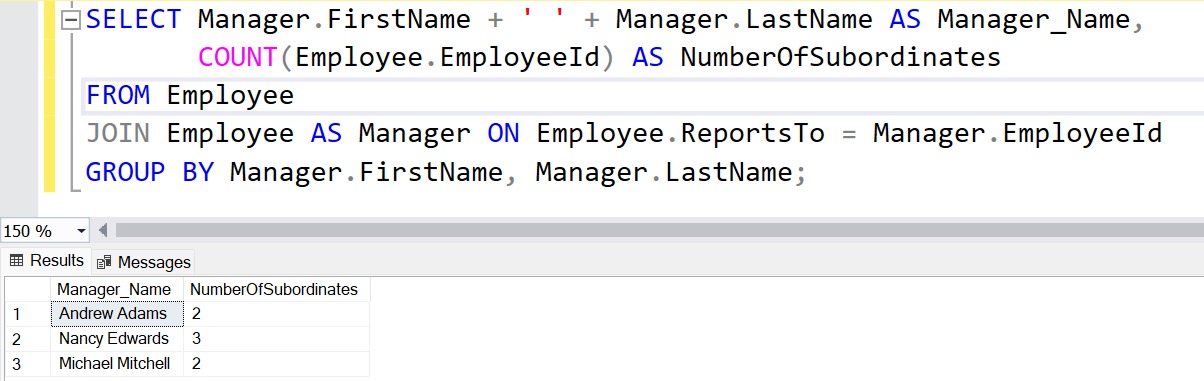
Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil Çoktan aza doğru şarkı sayısı ve sanatçı adı

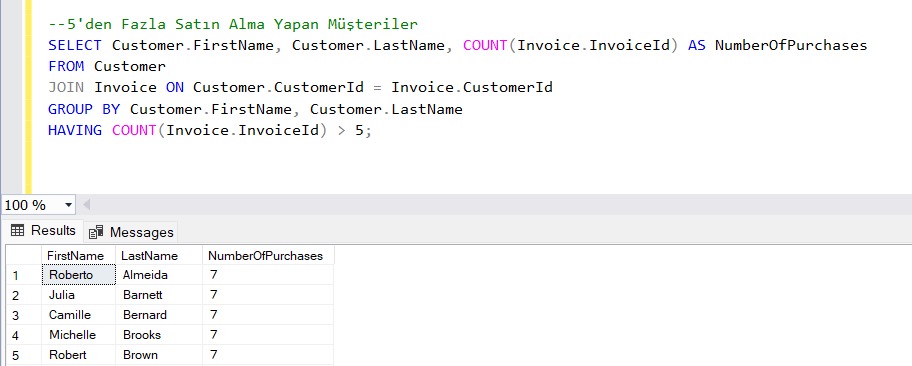
metin, ekran görüntüsü, yazılım, web sayfası içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

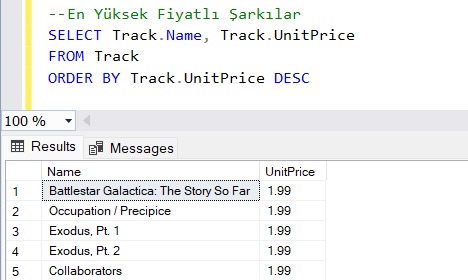
Şekil Ortalama albümdeki şarkı süreleri



Şekil Her yöneticinin sorumlu olduğu çalışan sayısı



Şekil 5'ten fazla satın alma yapan müşteri adı ve soyadı

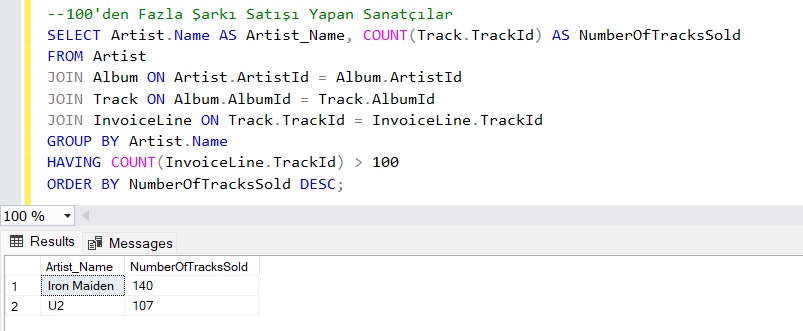


Şekil En yüksek fiyatlı şarkılar

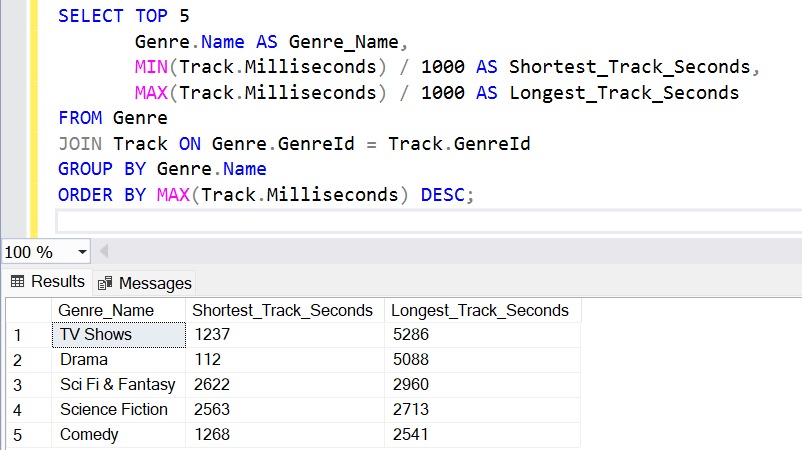
metin, ekran görüntüsü, yazılım, sayı, numara içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil 8 En çok satış yapılan ülkeler(çoktan aza)



Şekil 9 100’den fazla şarkı satışı yapan sanatçılar ve satılan şarkı sayısı

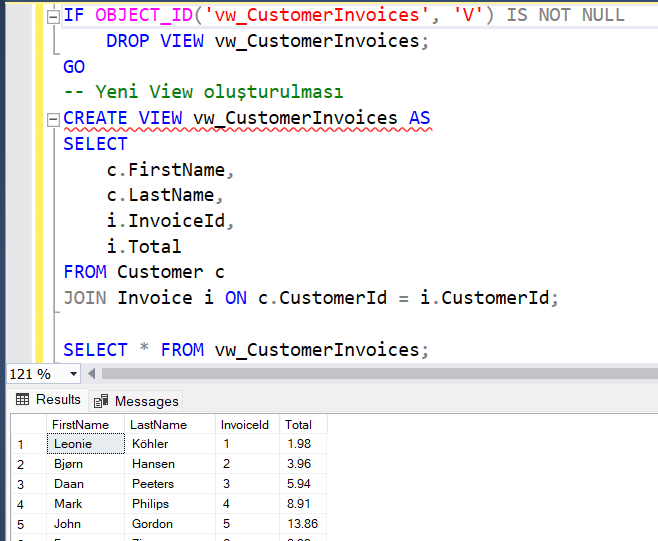


Şekil 10 En Kısa ve En Uzun Şarkılara Sahip İlk 5 Müzik Türü

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

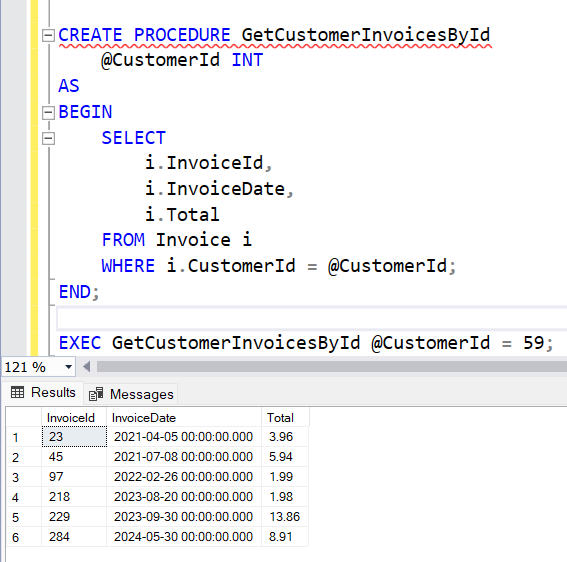
View, bir veya daha fazla tablodan alınan verilerin sanal bir tablosudur. Fiziksel olarak veri içermez, sadece bir SQL sorgusunun sonucu gibi davranır. Kullanıcının karmaşık sorgular yazmak zorunda kalmadan veriye erişmesini sağlar.

View;



Şekil 11 View ile kolayca fatura sahibi ve tutarını çekme

Saklı Yordam;



Şekil 12 Seçilen müşterinin fatura tarihi ve fiyatını çekme (Saklı Yordam)