

05.01.2025

# ONLINE EĞİTİM OTOMASYONU

## Proje Ekibi:

- 220260023 - Abdulkadir Karataş
- 220260065 - Ömer Erginoğuz
- 220260037 - Ahmet Melik Yıldırım

## Proje Bilgileri:

- Grup No: Grup 13
- Teslim Tarihi: 5 Ocak 2025
- Ders Adı: Veri Tabanı Sistemleri

## Gereksinim Analizi

### Öğrenci Gereksinimi

Gereksinim	Açıklama
Kursları Görüntüleme	Öğrenci, tüm kursları görüntüleyebilir.
Kurs Kaydı Yapma	Öğrenci, ilgilendiği kurslara kayıt olabilir.
Satın Alma İşlemleri	Öğrenci, kurs satın alma işlemlerini gerçekleştirebilir.
İçeriklere Erişim	Öğrenci, satın aldığı veya kayıt olduğu kursların içeriklerine erişebilir.
Sertifika Görüntüleme	Öğrenci, tamamladığı kurslardan aldığı sertifikaları görüntüleyebilir.
Değerlendirme Yapma	Öğrenci, katıldığı kursları değerlendirebilir ve yorum yapabilir.

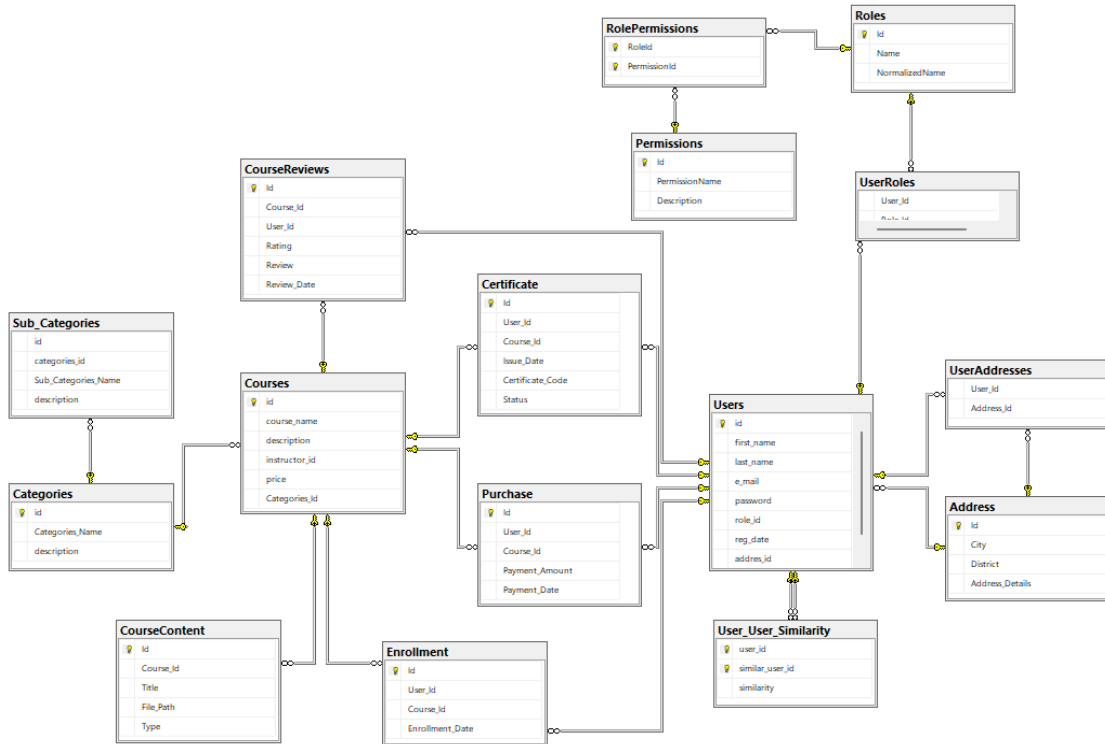
### Eğitmen Gereksinimi

Gereksinim	Açıklama
Kurs Ekleyebilme	Eğitmen, sistemde yeni kurslar oluşturabilir.
İçerik Yükleyebilme	Eğitmen, kurs içeriklerini sisteme yükleyebilir.
Kategori ve Alt Kategori Düzenleme	Eğitmen, kursların kategorilerini ve alt kategorilerini düzenleyebilir.
Kullanıcı Yönetimi	Eğitmen, kullanıcı bilgilerini görüntüleyebilir ve yönetebilir.
Sertifika Görüntüleme	Eğitmen, tamamlanan kurslara sertifika tanımlayabilir.

## Yönetici Gereksinimi

Gereksinim	Açıklama
Kullanıcı Yönetimi	Yönetici sistemdeki tüm kullanıcıları görüntüleyebilir, düzenleyebilir ve silebilir.
Kurs Yönetimi	Yönetici, tüm kursları görüntüleyebilir, düzenleyebilir ve silebilir.
Satın Alma Geçmişi Görüntüleme	Yönetici, kullanıcıların satın alma geçmişlerini görüntüleyebilir.
Raporlama ve İstatistik	Yönetici, kurs istatistiklerini ve raporlarını görebilir.
İçerik ve Kategori Yönetimi	Yönetici, kategoriler ve alt kategorilerde düzenlemeler yapabilir.
Yorum Yönetimi	Yönetici, kurslara yapılan yorumları yönetebilir.

## E-R Diyagramı ve İlişkiler



## Tablolar Arası İlişkiler

Tablo 1	Tablo 2	Arasındaki İlişki
Users	UserAddresses	1:N
UserAddresses	Address	N:1
Users	User-User-Similarity	1:N
Users	Roles	N:M
Users	UserRoles	1:N
Roles	Permissions	N:M
RolePermissions	Roles	N:1
Users	Courses	1:N
Courses	Enrollments	1:N
Courses	Purchases	1:N
Courses	Certificates	1:N
Categories	SubCategories	1:N
SubCategories	Courses	1:N

## Veri Tabanı Sorguları

### 1) Kullanıcı adres bazlı dağılımı sorgusu

```
SELECT
    a.City AS Şehir,
    a.District AS İlçe,
    COUNT(ua.User_Id) AS Kullanıcı_Sayısı
FROM Address a
JOIN UserAddresses ua ON a.Id = ua.Address_Id
GROUP BY
    a.District, a.City
ORDER BY
    Kullanıcı_Sayısı DESC;
```

### 2) Kurslarda verilen sertifika durumları sorgusu

```
SELECT
    c.course_name AS Kurs_Adı,
    COUNT(DISTINCT cert.Id) AS Verilen_Sertifika_Sayısı,
    CASE
        WHEN COUNT(DISTINCT cert.Id) > 0 THEN 'Verildi'
        ELSE 'Verilmedi'
    END AS Sertifika_Durumu
FROM Courses c
```

```

LEFT JOIN Certificate cert ON c.id = cert.Course_Id
GROUP BY
    c.id, c.course_name
ORDER BY
    Sertifika_Durumu ASC, Verilen_Sertifika_Sayısı DESC;

```

### 3) En yüksek derecelendirilen kurslar

```

SELECT
    c.course_name AS Kurs_Adı,
    AVG(cr.Rating) AS Ortalama_Puan,
    COUNT(cr.Id) AS Toplam_Yorum_Sayısı
FROM Courses c
LEFT JOIN CourseReviews cr ON cr.Course_Id = c.id
GROUP BY
    c.id, c.course_name
HAVING
    COUNT(cr.Id) >= 3
ORDER BY
    Ortalama_Puan DESC, Toplam_Yorum_Sayısı DESC;

```

### 4) Kullanıcıların kurs tamamlama oranları ve tamamlanan kurs sayıları

```

SELECT
    CONCAT(u.first_name, ' ', u.last_name) AS Ad_Soyad,
    COUNT(e.Course_Id) AS Toplam_Kayıt,
    COUNT(CASE WHEN cert.Status = 'Verildi' THEN e.User_Id END) AS Tamamlanan_Kurs,
    ROUND(
        (CAST(COUNT(CASE WHEN cert.Status = 'Verildi' THEN e.Course_Id END) AS FLOAT)
        * 100.0) /
        NULLIF(COUNT(e.Course_Id), 0), 2
    ) AS Tamamlama_Oranı
FROM Users u
LEFT JOIN Enrollment e ON u.id = e.User_Id
LEFT JOIN Certificate cert ON cert.Course_Id = e.Course_Id AND cert.User_Id =
e.User_Id
GROUP BY
    u.first_name, u.last_name
ORDER BY
    Tamamlama_Oranı DESC, Tamamlanan_Kurs DESC;

```

## Veri Tabanı Stored Procedure İşlemleri

### 1) Kullanıcılar arası benzerlik hesabı

```

CREATE PROCEDURE CalculateUserSimilarity
AS
BEGIN

    DELETE FROM user_user_similarity;

    INSERT INTO user_user_similarity (user_id, similar_user_id, similarity)
    SELECT
        u1.user_id,
        u2.user_id,
        AVG(u1.Rating * u2.Rating) AS similarity
    FROM CourseReviews u1
    JOIN CourseReviews u2 ON u1.Course_Id = u2.Course_Id
    WHERE u1.user_id != u2.user_id
    GROUP BY u1.user_id, u2.user_id
    HAVING NOT EXISTS (

```

```

        SELECT 1
        FROM user_user_similarity u
        WHERE (u.user_id = u1.user_id AND u.similar_user_id = u2.user_id)
              OR (u.user_id = u2.user_id AND u.similar_user_id = u1.user_id)
    );

    PRINT 'User similarities calculated and stored.';
END;

```

## 2) Collaborative Filtering işlemini yapan Prosedür

```

CREATE PROCEDURE Collaborative_Filtering
    @UserId INT
AS
BEGIN
    SELECT TOP 10 c.Course_Id, AVG(c.Rating) AS average_rating
    FROM CourseReviews c
    JOIN(
        SELECT TOP 5 similar_user_id, similarity
        FROM user_user_similarity
        WHERE user_id = @UserId
        ORDER BY similarity DESC
    ) s ON c.User_Id = s.similar_user_id
    WHERE c.User_Id != @UserId
    GROUP BY c.Course_Id
    ORDER BY average_rating DESC;
END;

```

## 3) Gelişmiş arama işlemi

```

CREATE PROCEDURE SearchCourses
    @CourseName NVARCHAR(100) = NULL,
    @CategoryName NVARCHAR(100) = NULL,
    @MinPrice DECIMAL(10, 2) = NULL,
    @MaxPrice DECIMAL(10, 2) = NULL
AS
BEGIN
    DECLARE @SQL NVARCHAR(MAX);
    DECLARE @TransactionState INT;
    DECLARE @ResultMessage NVARCHAR(200);

    BEGIN TRANSACTION;

    SET @SQL = 'SELECT * FROM SearchView WHERE 1=1';

    IF @CourseName IS NOT NULL
        SET @SQL = @SQL + ' AND course_name LIKE '%' + @CourseName + '%''';

    IF @CategoryName IS NOT NULL
        SET @SQL = @SQL + ' AND category_name LIKE '%' + @CategoryName + '%''';

    IF @MinPrice IS NOT NULL
        SET @SQL = @SQL + ' AND price >= ' + CAST(@MinPrice AS NVARCHAR);

    IF @MaxPrice IS NOT NULL
        SET @SQL = @SQL + ' AND price <= ' + CAST(@MaxPrice AS NVARCHAR);

    BEGIN TRY
        EXEC sp_executesql @SQL;

        SET @TransactionState = 1;
    END TRY
    BEGIN CATCH
        ROLLBACK;
    END CATCH
END;

```

```

        SET @ResultMessage = 'Sorgu başarılı, işlem tamamlandı.';
    END TRY
    BEGIN CATCH
        SET @TransactionState = 0;
        SET @ResultMessage = 'Hata oluştu: ' + ERROR_MESSAGE();
        ROLLBACK TRANSACTION;
    END CATCH

    IF @TransactionState = 1
    BEGIN
        COMMIT TRANSACTION;
    END

    SELECT @ResultMessage AS TransactionResult;

END;

```

## Veri Tabanı Trigger işlemi

### 1) Kurs geri ödeme işlemi

```

CREATE TRIGGER trg_courseRefund
ON Enrollment
INSTEAD OF DELETE
AS
BEGIN
    BEGIN TRANSACTION;

    DECLARE @CourseStartDate SMALLDATETIME;
    DECLARE @User_Id INT;
    DECLARE @Course_Id INT;

    SELECT @User_Id = User_Id, @Course_Id = Course_Id FROM DELETED;

    SELECT @CourseStartDate = Enrollment_Date
    FROM Enrollment
    WHERE User_Id = @User_Id AND Course_Id = @Course_Id;

    IF ABS(DATEDIFF(DAY, @CourseStartDate, GETDATE())) <= 14
    BEGIN
        DELETE FROM Enrollment WHERE User_Id = @User_Id AND Course_Id = @Course_Id;

        IF @@ROWCOUNT > 0
        BEGIN
            PRINT 'İade başarılı.';
            COMMIT TRANSACTION;
        END
        ELSE
        BEGIN
            PRINT 'İade başarısız.';
            ROLLBACK TRANSACTION;
        END
    END
    ELSE
    BEGIN
        PRINT '14 gün geçtiği için iade yapılamaz.';
        ROLLBACK TRANSACTION;
    END

```

```
END  
END;
```

## Veri Tabanı Tabloları

```
CREATE TABLE Users(  
    id INT IDENTITY(1,1) PRIMARY KEY,  
    first_name VARCHAR(255) NOT NULL,  
    last_name VARCHAR(255) NOT NULL,  
    e_mail VARCHAR(255) NOT NULL UNIQUE,  
    password VARCHAR(255) NOT NULL,  
    role_id INT NOT NULL,  
    reg_date SMALLDATETIME DEFAULT GETDATE(),  
    address_id INT NULL,  
    FOREIGN KEY(role_id) REFERENCES Roles(Id),  
    FOREIGN KEY(address_id) REFERENCES Address(Id)  
);  
  
CREATE TABLE Categories(  
    id INT IDENTITY(1,1) PRIMARY KEY,  
    Categories_Name NVARCHAR(255) NOT NULL,  
    description NVARCHAR(MAX) NULL  
);  
  
CREATE TABLE Sub_Categories(  
    id INT IDENTITY(1,1) PRIMARY KEY,  
    categories_id INT NOT NULL,  
    Sub_Categories_Name NVARCHAR(255) NOT NULL,  
    description NVARCHAR(MAX) NULL,  
    FOREIGN KEY(categories_id) REFERENCES Categories(id)  
);  
  
CREATE TABLE Courses(  
    id INT IDENTITY(1,1) PRIMARY KEY,  
    course_name NVARCHAR(255) NOT NULL,  
    description NVARCHAR(MAX) NULL,  
    instructor_id INT NOT NULL,  
    price DECIMAL(10,2) NOT NULL,  
    Categories_Id INT NOT NULL,  
    FOREIGN KEY(Categories_Id) REFERENCES Categories(id)  
);  
  
CREATE TABLE Roles (  
    Id INT IDENTITY(1,1) PRIMARY KEY NOT NULL,  
    Name VARCHAR(255) NOT NULL,  
    NormalizedName VARCHAR(255) NOT NULL  
);  
  
CREATE TABLE UserRoles (  
    User_Id INT NOT NULL,  
    Role_Id INT NOT NULL,  
    FOREIGN KEY (User_Id) REFERENCES Users (id),  
    FOREIGN KEY (Role_Id) REFERENCES Roles (Id)  
);  
  
CREATE TABLE Purchase (  
    Id INT IDENTITY(1,1) PRIMARY KEY NOT NULL,  
    User_Id INT NOT NULL,
```



```

    Course_Id INT NOT NULL,
    Payment_Amount DECIMAL NOT NULL,
    Payment_Date SMALLDATETIME DEFAULT GETDATE(),
    FOREIGN KEY (User_Id) REFERENCES Users (id),
    FOREIGN KEY (Course_Id) REFERENCES Courses (id)
);

CREATE TABLE Certificate(
    Id INT IDENTITY(1,1) PRIMARY KEY NOT NULL,
    User_Id INT NOT NULL,
    Course_Id INT NOT NULL,
    Issue_Date SMALLDATETIME NOT NULL,
    Certificate_Code VARCHAR(50) NOT NULL UNIQUE,
    Status VARCHAR(50) NOT NULL,
    FOREIGN KEY (User_Id) REFERENCES Users (id),
    FOREIGN KEY (Course_Id) REFERENCES Courses (id)
);

CREATE TABLE Address (
    Id INT IDENTITY(1,1) PRIMARY KEY NOT NULL,
    City VARCHAR(255) NOT NULL,
    District VARCHAR(255) NOT NULL,
    Address_Details VARCHAR(255) NOT NULL
);

CREATE TABLE UserAddresses (
    User_Id INT NOT NULL,
    Address_Id INT NOT NULL,
    FOREIGN KEY (User_Id) REFERENCES Users (id),
    FOREIGN KEY (Address_Id) REFERENCES Address (Id)
);

CREATE TABLE CourseContent (
    Id INT IDENTITY(1,1) PRIMARY KEY NOT NULL,
    Course_Id INT NOT NULL,
    Title VARCHAR(255) NOT NULL,
    File_Path VARCHAR(255) NOT NULL,
    Type VARCHAR(255) NOT NULL,
    FOREIGN KEY (Course_Id) REFERENCES Courses (id)
);

CREATE TABLE CourseReviews (
    Id INT IDENTITY(1,1) PRIMARY KEY NOT NULL,
    Course_Id INT NOT NULL,
    User_Id INT NOT NULL,
    Rating DECIMAL(3, 2) NULL,
    Review TEXT NULL,
    Review_Date SMALLDATETIME DEFAULT GETDATE(),
    FOREIGN KEY (User_Id) REFERENCES Users (id),
    FOREIGN KEY (Course_Id) REFERENCES Courses (id)
);

CREATE TABLE Enrollment (
    Id INT IDENTITY(1,1) PRIMARY KEY NOT NULL,
    User_Id INT NOT NULL,
    Course_Id INT NOT NULL,
    Enrollment_Date SMALLDATETIME DEFAULT GETDATE(),
    FOREIGN KEY (User_Id) REFERENCES Users (id),
    FOREIGN KEY (Course_Id) REFERENCES Courses (id)
);

CREATE TABLE User_User_Similarity (

```

```

        user_id INT,
        similar_user_id INT,
        similarity DECIMAL(5, 2),
        PRIMARY KEY (user_id, similar_user_id),
        FOREIGN KEY (user_id) REFERENCES Users(id) ON DELETE CASCADE,
        FOREIGN KEY (similar_user_id) REFERENCES Users(id) ON DELETE NO ACTION
    );

CREATE TABLE Permissions(
    Id INT PRIMARY KEY,
    PermissionName NVARCHAR(100) NOT NULL,
    Description NVARCHAR(255)
);

CREATE TABLE RolePermissions (
    RoleId INT,
    PermissionId INT,
    PRIMARY KEY (RoleId, PermissionId),
    FOREIGN KEY (RoleId) REFERENCES Roles(Id),
    FOREIGN KEY (PermissionId) REFERENCES Permissions(Id)
);

-- CourseSearch View

CREATE VIEW SearchView AS
SELECT
    c.id AS Course_Id,
    c.course_name,
    cat.Categories_Name AS Category_Name,
    c.price
FROM Courses c
JOIN Categories cat ON c.Categories_Id = cat.id;

```

## Veri Tabanı Verileri

```

INSERT INTO Users (id,first_name, last_name, e_mail, password, role_id, reg_date,
addres_id)
VALUES
(4,'Emre', 'Çelik', 'emre.celik@example.com', 'hashed_password_4', 1, '2023-09-20
12:05:15', 4),
(5,'Fatma', 'Aydın', 'fatma.aydin@example.com', 'hashed_password_5', 2, '2023-10-15
09:15:25', 5),
(6,'Seda', 'Özdemir', 'seda.ozdemir@example.com', 'hashed_password_6', 3, '2023-11-01
13:25:30', 6),
(7,'Ali', 'Koç', 'ali.koc@example.com', 'hashed_password_7', 1, '2023-06-18 07:55:45',
7),
(8,'Zeynep', 'Demir', 'zeynep.demir@example.com', 'hashed_password_8', 2, '2023-07-28
15:35:50', 8),
(9,'Can', 'Bakan', 'can.bakan@example.com', 'hashed_password_9', 3, '2023-08-12
11:15:30', 9),
(10,'Büşra', 'Gül', 'busra.gul@example.com', 'hashed_password_10', 1, '2023-09-22
16:10:40', 10),
(11,'Mehmet', 'Yalçın', 'mehmet.yalcin@example.com', 'hashed_password_11', 2, '2023-
06-25 18:30:25', 11),
(12,'Nisa', 'Kuş', 'nisa.kus@example.com', 'hashed_password_12', 3, '2023-07-15
14:00:10', 12),

```

```
(13,'Ömer', 'Sarı', 'omer.sari@example.com', 'hashed_password_13', 1, '2023-08-02
10:45:20', 13),
(14,'Ebru', 'Öztürk', 'ebru.ozturk@example.com', 'hashed_password_14', 2, '2023-09-10
17:05:00', 14),
(15,'Baran', 'Kara', 'baran.kara@example.com', 'hashed_password_15', 3, '2023-10-18
19:15:55', 15),
(16,'Aylin', 'Karaca', 'aylin.karaca@example.com', 'hashed_password_16', 1, '2023-11-
05 13:30:40', 16),
(17,'Okan', 'Bayram', 'okan.bayram@example.com', 'hashed_password_17', 2, '2023-06-08
16:25:10', 17),
(18,'Leyla', 'Yüce', 'leyla.yuce@example.com', 'hashed_password_18', 3, '2023-07-18
12:35:30', 18),
(19,'Gökhan', 'Tunc', 'gokhan.tunc@example.com', 'hashed_password_19', 1, '2023-08-22
14:40:50', 19),
(20,'Murat', 'Başaran', 'murat.basaran@example.com', 'hashed_password_20', 2, '2023-
09-03 09:25:05', 20);
```

```
INSERT INTO Courses (id,course_name, description, instructor_id)
VALUES
```

```
(1,'Kişisel Gelişim', 'Kişisel gelişim ve motivasyon üzerine bir kurs.', 1),
(2,'Yazılım Geliştirme', 'Modern yazılım geliştirme teknikleri ve araçları.', 2),
(3,'Veritabanı Yönetimi', 'Veritabanı tasarımı ve yönetimi hakkında eğitim.', 3),
(4,'İleri Seviye Python', 'Python programlamaya derinlemesine bir bakış.', 1),
(5,'Web Tasarımı', 'Web sayfaları oluşturma ve tasarlama eğitimleri.', 2),
(6,'Makine Öğrenmesi', 'Makine öğrenmesi algoritmalarının uygulanması.', 3);
```

```
INSERT INTO CourseReviews(Course_Id, User_Id, Rating, Review, Review_Date)
VALUES
```

```
(3, 6, 4.25, 'Kurs içeriği oldukça faydalı ancak biraz daha pratik yapabilirdim.',
'2023-10-01 14:00:00'),
(4, 7, 5.00, 'Harika bir eğitim, tüm konuları ayrıntılı bir şekilde ele alıyor.',
'2023-10-05 09:30:00'),
(5, 8, 4.00, 'İçerik genel olarak iyi fakat bazı konularda daha fazla açıklama
olabilirdi.', '2023-10-10 16:00:00'),
(2, 9, 3.50, 'Kurs başlangıç seviyesindeki kullanıcılar için uygun ancak ileri düzeyde
eksikler var.', '2023-10-12 11:45:00'),
(1, 10, 4.75, 'Çok iyi bir kurs, içerik ve anlatım mükemmel, sadece bazı bölümlerde
daha fazla örnek olabilirdi.', '2023-10-15 18:20:00'),
(3, 11, 3.00, 'Kurs eksik ve bazı bölümler yeterince detaylı değildi.', '2023-10-18
13:10:00'),
(4, 12, 5.00, 'Çok kapsamlı ve derinlemesine bir eğitim, her seviyeye hitap ediyor.',
'2023-10-22 15:25:00'),
(5, 13, 4.50, 'İçerik çok öğretici ancak hız biraz fazla, daha yavaş
ilerleyebilirdi.', '2023-10-25 14:40:00'),
(2, 14, 4.25, 'Güzel bir kurs ama ileri seviyeler için yeterli değil.', '2023-10-27
10:00:00'),
(1, 15, 5.00, 'Mükemmel bir kurs, gerçekten çok faydalı.', '2023-10-30 12:30:00'),
(3, 16, 4.00, 'İçerik iyi ama bazı bölümler daha fazla örnekle desteklenebilirdi.',
'2023-11-02 17:00:00'),
(4, 17, 4.75, 'Eğitim harika ama birkaç teknik detay eksikti.', '2023-11-05
09:20:00'),
(5, 18, 3.50, 'Kurs bazı bölümlerde eksik, daha fazla ayrıntı beklerdim.', '2023-11-07
14:50:00'),
(2, 19, 4.00, 'İyi bir kurs ancak bazı konular daha derinlemesine işlenebilirdi.',
'2023-11-10 16:30:00'),
(1, 20, 5.00, 'Çok kapsamlı ve faydalı bir kurs, herkese tavsiye ederim.', '2023-11-13
13:00:00');
```

```
INSERT INTO CourseReviews (Course_Id, User_Id, Rating, Review, Review_Date)
VALUES
```

```
(1, 1, 4.5, 'Çok faydalı bir kurs, kendimi geliştirdim.', '2023-06-20 14:10:10'),
(2, 2, 5.0, 'Gerçekten çok öğretici ve kapsamlı bir kurs.', '2023-07-14 11:20:25'),
(3, 3, 3.8, 'İçeriği iyi ama bazı kısımlar eksikti.', '2023-08-30 13:40:40'),
(1, 4, 4.0, 'Kurs genel olarak iyi ama bazı konular daha ayrıntılı olabilirdi.',
'2023-09-05 10:30:15'),
(2, 5, 5.0, 'Mükemmel bir kurs, her seviyeye hitap ediyor.', '2023-09-20 12:15:45');
```

```
INSERT INTO CourseContent (Course_Id, Title, File_Path, Type)
```

```
VALUES
```

```
(1, 'Kişisel Gelişim 101', '/files/personal_development_intro.pdf', 'PDF'),
(2, 'Yazılım Geliştirme Temelleri', '/files/software_development_basics.mp4',
'Video'),
(3, 'Veritabanı Tasarımı', '/files/db_design_basics.pdf', 'PDF'),
(4, 'Python Temelleri', '/files/advanced_python_basics.mp4', 'Video'),
(5, 'HTML ve CSS Başlangıç', '/files/web_design_intro.pdf', 'PDF'),
(6, 'Makine Öğrenmesi 101', '/files/machine_learning_intro.mp4', 'Video');
```

```
INSERT INTO Certificate (User_Id, Course_Id, Issue_Date, Certificate_Code, Status)
```

```
VALUES
```

```
(1, 1, '2023-07-01 10:00:00', 'CD20230701', 'Verildi'),
(2, 2, '2023-08-01 11:00:00', 'SD20230801', 'Verildi'),
(3, 3, '2023-09-10 12:00:00', 'DB20230910', 'Verildi'),
(4, 4, '2023-09-15 14:00:00', 'PY20230915', 'Verildi'),
(5, 5, '2023-09-20 09:00:00', 'WD20230920', 'Verildi'),
(6, 1, '2023-10-01 10:00:00', 'CD20231001', 'Verildi'),
(7, 2, '2023-10-05 11:00:00', 'SD20231005', 'Verilmedi'),
(18, 3, '2023-10-10 12:00:00', 'DB20231010', 'Verildi'),
(9, 4, '2023-10-15 14:00:00', 'PY20231015', 'Verildi'),
(10, 5, '2023-10-20 09:00:00', 'WD20231020', 'Verilmedi'),
(11, 1, '2023-11-01 10:00:00', 'CD20231101', 'Verildi'),
(12, 2, '2023-11-05 11:00:00', 'SD20231105', 'Verilmedi'),
(13, 3, '2023-11-10 12:00:00', 'DB20231110', 'Verildi'),
(14, 4, '2023-11-15 14:00:00', 'PY20231115', 'Verildi'),
(15, 5, '2023-11-20 09:00:00', 'WD20231120', 'Verildi'),
(16, 1, '2023-12-01 10:00:00', 'CD20231201', 'Verilmedi'),
(17, 2, '2023-12-05 11:00:00', 'SD20231205', 'Verildi'),
(18, 3, '2023-12-10 12:00:00', 'DB20231210', 'Verilmedi'),
(19, 4, '2023-12-15 14:00:00', 'PY20231215', 'Verildi'),
(20, 5, '2023-12-20 09:00:00', 'WD20231220', 'Verildi');
```

```
INSERT INTO Categories (id,Categories_Name, description)
```

```
VALUES
```

```
(1,'Kişisel Gelişim', 'Kişisel gelişim ve motivasyon üzerine eğitimler'),
(2,'Yazılım', 'Yazılım geliştirme ve programlama eğitimleri'),
(3,'Veritabanı', 'Veritabanı yönetimi ve SQL dersleri'),
(4,'Makine Öğrenmesi', 'Makine öğrenmesi ve veri bilimi kursları'),
(5,'Web Tasarımı', 'Web tasarımı ve geliştirme kursları');
```

```
INSERT INTO Sub_Categories (id,categories_id, Sub_Categories_Name, description)
```

```
VALUES
```

```
(1,1, 'Kişisel Gelişim Teknikleri', 'Kişisel gelişim alanında temel teknikler'),
(2,2, 'Web Yazılımı', 'Web uygulamaları geliştirme eğitimleri'),
(3,3, 'SQL Veri Tabanı Yönetimi', 'SQL kullanımı ve veri tabanı yönetimi eğitimleri'),
(4,4, 'Derin Öğrenme', 'Derin öğrenme tekniklerine giriş'),
(5,5, 'UI/UX Tasarımı', 'Kullanıcı arayüzü ve deneyimi tasarımı');
```

```
INSERT INTO Address (City, District, Address_Details)
```

```
VALUES
```

```
('İstanbul', 'Kadıköy', 'Fenerbahçe Mahallesi, No:45'),
('Ankara', 'Çankaya', 'Kocatepe Mahallesi, No:12'),
('İzmir', 'Konak', 'Alsancak Mahallesi, No:7'),
('Bursa', 'Nilüfer', 'İhsaniye Mahallesi, No:32'),
```

```
('Adana', 'Seyhan', 'Yüreğir Mahallesi, No:18');
```

```
INSERT INTO UserAddresses (User_Id, Address_Id)
VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5);
```

```
INSERT INTO UserAddresses(User_Id, Address_Id)
VALUES
(1, 1),
(1, 5),
(2, 2),
(2, 10),
(3, 3),
(3, 15),
(3, 20),
(4, 4),
(5, 6),
(5, 7),
(6, 8),
(6, 12),
(7, 9),
(7, 11),
(8, 13),
(8, 18),
(9, 14),
(10, 16),
(10, 19),
(11, 17),
(12, 20),
(12, 2),
(13, 3),
(13, 8),
(14, 9),
(14, 10),
(15, 5),
(15, 12),
(16, 6),
(16, 13),
(17, 4),
(18, 7),
(18, 15),
(19, 11),
(20, 14),
(20, 19);
```

```
INSERT INTO Enrollment (User_Id, Course_Id, Enrollment_Date)
VALUES
(1, 1, '2023-06-20 10:00:00'),
(2, 2, '2023-07-10 11:00:00'),
(3, 3, '2023-08-15 12:00:00'),
(4, 4, '2023-09-05 13:00:00'),
(5, 5, '2023-09-20 14:00:00');
```

```
INSERT INTO Purchase (User_Id, Course_Id, Payment_Amount, Payment_Date)
VALUES
(1, 1, 199.99, '2023-06-20 09:00:00'),
(2, 2, 249.99, '2023-07-15 10:30:00'),
```

```
(3, 3, 179.99, '2023-08-22 12:00:00'),
(4, 4, 299.99, '2023-09-10 14:15:00'),
(5, 5, 229.99, '2023-09-25 16:00:00');
```

```
INSERT INTO UserRoles (User_Id, Role_Id)
VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 1),
(5, 2),
(6, 3),
(7, 1),
(8, 2),
(9, 3),
(10, 1),
(11, 2),
(12, 3),
(13, 1),
(14, 2),
(15, 3),
(16, 1),
(17, 2),
(18, 3),
(19, 1),
(20, 2);
```

```
INSERT INTO Address (City, District, Address_Details)
VALUES
('Adana', 'Seyhan', 'Cemalpaşa Mahallesi No:15'),
('Mersin', 'Akdeniz', 'Atatürk Bulvarı No:35'),
('Gaziantep', 'Şahinbey', 'Müslüm Güven Caddesi No:50'),
('Eskişehir', 'Tepebaşı', 'Cumhuriyet Mahallesi No:10'),
('Trabzon', 'Ortahisar', 'Süleyman Fethi Caddesi No:27'),
('Samsun', 'Atakum', 'Deniz Mahallesi No:11'),
('Denizli', 'Merkezefendi', 'Kazım Karabekir Caddesi No:72'),
('Bolu', 'Merkez', 'İzzet Baysal Caddesi No:18'),
('Kocaeli', 'İzmit', 'İzmit Yolu No:45'),
('Çorum', 'Merkez', 'Şehitler Caddesi No:9'),
('Aydın', 'Efeler', 'Adnan Menderes Caddesi No:21'),
('Manisa', 'Şehzadeler', 'Sultan Caddesi No:3'),
('Diyarbakır', 'Sur', 'Ali Emiri Caddesi No:27'),
('Kırşehir', 'Merkez', 'Kocatepe Caddesi No:40'),
('Sakarya', 'Adapazarı', 'Cumhuriyet Caddesi No:75'),
('Rize', 'Merkez', 'Piriçelebi Caddesi No:55'),
('Aksaray', 'Merkez', 'Atatürk Bulvarı No:89'),
('Nevşehir', 'Merkez', 'Gülbahar Caddesi No:13'),
('Kastamonu', 'Merkez', 'Zafer Caddesi No:26'),
('Zonguldak', 'Merkez', 'Çaycuma Yolu No:65');
```

```
INSERT INTO Enrollment (User_Id, Course_Id, Enrollment_Date)
VALUES
(1, 1, '2023-06-20 10:00:00'),
(2, 2, '2023-07-10 11:00:00'),
(3, 3, '2023-08-15 12:00:00'),
(4, 4, '2023-09-05 13:00:00'),
(5, 5, '2023-09-20 14:00:00'),
(6, 6, '2023-10-05 15:00:00'),
(7, 1, '2023-06-25 09:30:00'),
(8, 2, '2023-07-15 11:10:00'),
(9, 3, '2023-08-25 16:20:00'),
(10, 4, '2023-09-10 13:45:00'),
```

```
(11, 5, '2023-09-18 14:25:00'),  
(12, 6, '2023-10-10 12:00:00'),  
(13, 1, '2023-06-30 08:50:00'),  
(14, 2, '2023-07-20 11:40:00'),  
(15, 3, '2023-08-28 09:10:00'),  
(16, 4, '2023-09-12 15:30:00'),  
(17, 5, '2023-09-22 14:05:00'),  
(18, 6, '2023-10-15 10:25:00'),  
(19, 1, '2023-07-01 10:15:00'),  
(20, 2, '2023-07-25 13:50:00');
```

```
INSERT INTO Permissions (Id, PermissionName, Description)  
VALUES  
(1,'ViewCourses', 'Kursları görüntüleme izni'),  
(2,'EditCourses', 'Kursları düzenleme izni'),  
(3,'DeleteCourses', 'Kursları silme izni');
```

```
INSERT INTO RolePermissions (RoleId, PermissionId)  
VALUES  
(1, 1),  
(2, 1),  
(2, 2),  
(3, 1),  
(3, 2),  
(3, 3);
```