

Proje Adı	Eğitim Kurumu Kayıt Sistemi
Proje Konusu	Yeni öğrenci başvuru ve kayıt süreci
Veritabanı Adı	SAT242516014

Veritabanında bulunan kullanıcı tanımlı tabloların (table) oluşturma SQL ifadelerini yazınız

(AspNet... ile başlayan tablolar yazılmayacak)

```

CREATE TABLE [dbo].[Student]
(
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [Ad] [nvarchar](50) NULL,
    [Soyad] [nvarchar](50) NULL,
    [Email] [nvarchar](100) NULL,
    [Numara] [nvarchar](20) NULL,
    [DogumTarihi] [date] NULL,
    [Tc] [char](11) NULL,
    CONSTRAINT [PK_Student] PRIMARY KEY CLUSTERED ([Id] ASC)
)
GO
CREATE TABLE [dbo].[Ders]
(
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [Ad] [nvarchar](100) NULL, -- Dersin Adı (örn: Matematik 101)
    [Kod] [nvarchar](20) NULL, -- Ders Kodu (örn: MAT101)
    [Kredi] [int] NULL, -- Dersin kredisi
    CONSTRAINT [PK_Ders] PRIMARY KEY CLUSTERED ([Id] ASC)
)
GO
CREATE TABLE [dbo].[OgrenciDersKayit]
(
    [OgrenciId] [int] NOT NULL, -- Öğrenci tablosundan ID
    [DersId] [int] NOT NULL, -- Ders tablosundan ID
    [KayitTarihi] [date] DEFAULT GETDATE(),
    [Donem] [nvarchar](20) NULL, -- (Örn: 2025-Güz)
    CONSTRAINT [PK_OgrenciDersKayit] PRIMARY KEY CLUSTERED
    (
        [OgrenciId] ASC,
        [DersId] ASC
    ),
    CONSTRAINT [FK_OgrenciDersKayit_Student] FOREIGN KEY ([OgrenciId])
        REFERENCES [dbo].[Student]([Id])
        ON DELETE CASCADE, -- Öğrenci silinirse, ders kayıtları da silinir

    -- Ders tablosu ile ilişki (Foreign Key)
    CONSTRAINT [FK_OgrenciDersKayit_Ders] FOREIGN KEY ([DersId])
        REFERENCES [dbo].[Ders]([Id])
        ON DELETE CASCADE -- Ders silinirse, o derse ait kayıtlar da silinir
)
GO

```

Veritabanında bulunan kullanıcı tanımlı görünümlerin (view) oluşturma SQL ifadelerini yazınız.

```
CREATE VIEW [dbo].[View_Student]
AS
SELECT *
FROM Student
GO
CREATE VIEW [dbo].[View_Ders]
AS
SELECT Id, Kod, Ad, Kredi FROM dbo.Ders
GO
CREATE VIEW [dbo].[View_OgrenciDersKayit]
AS
SELECT
    odk.OgrenciId,
    s.Ad + ' ' + s.Soyad AS OgrenciAdSoyad,
    odk.DersId,
    d.Ad AS DersAd,
    d.Kod AS DersKod,
    odk.KayitTarihi,
    odk.Donem
FROM
    [dbo].[OgrenciDersKayit] odk
INNER JOIN
    [dbo].[Student] s ON odk.OgrenciId = s.Id
INNER JOIN
    [dbo].[Ders] d ON odk.DersId = d.Id
GO
```

Veritabanında bulunan kullanıcı tanımlı prosedürlerin (procedure) oluşturma SQL ifadelerini yazınız.

```
CREATE PROCEDURE [dbo].[sp_Student_Add_Update_Remove]
    @operation VARCHAR(10),
    @jsonvalues NVARCHAR(MAX)
AS
BEGIN
    SELECT *
    INTO #temp
    FROM OPENJSON(@jsonvalues)
    WITH
    (
        Id INT,
        Ad NVARCHAR(50),
        Soyad NVARCHAR(50),
        Department VARCHAR(50), -- Bu sütun Student tablosunda yok, dikkate
alınmayacak.
        Email NVARCHAR(100),
        Numara NVARCHAR(20),
        DoğumTarihi DATE,
        Tc CHAR(11)
    )

    DECLARE @rowcount INT = NULL

    IF @operation = 'add'
    BEGIN
        INSERT Student (Ad, Soyad, Email, Numara, DoğumTarihi, Tc)
        SELECT Ad, Soyad, Email, Numara, DoğumTarihi, Tc
        FROM #temp

        SET @rowcount = @@ROWCOUNT
    END

    IF @operation = 'update'
    BEGIN
        UPDATE s
        SET s.Ad = t.Ad,
            s.Soyad = t.Soyad,
            s.Email = t.Email,
            s.Numara = t.Numara,
            s.DoğumTarihi = t.DoğumTarihi,
            s.Tc = t.Tc
        FROM #temp t JOIN Student s ON t.Id = s.Id

        SET @rowcount = @@ROWCOUNT
    END

    IF @operation = 'remove'
    BEGIN
        DELETE s
        FROM #temp t JOIN Student s ON t.Id = s.Id

        SET @rowcount = @@ROWCOUNT
    END

    SELECT @operation [Key], IIF(ISNULL(@rowcount, 0) > 0, 1, 0) [Value]
END
GO
CREATE PROCEDURE [dbo].[sp_Students]
    @pagination Type_Dictionary_String_String READONLY,
    @where Type_Dictionary_String_String READONLY
AS
BEGIN
    ;
    WITH cte AS
```

```
(
    SELECT *
    FROM View_Student
),
cte_TRC AS (
    SELECT COUNT(*) TotalRecordCount
    FROM cte
)
SELECT *
FROM cte, cte_TRC
END
GO
CREATE PROCEDURE [dbo].[sp_Ders_Listele]
    @pagination dbo.Type_Dictionary_String_String READONLY,
    @where dbo.Type_Dictionary_String_String READONLY
AS
BEGIN
    ;
    WITH cte AS (
        SELECT * FROM dbo.View_Ders
    ),
    cte_TRC AS (
        SELECT COUNT(*) AS TotalRecordCount FROM cte
    )
    SELECT * FROM cte, cte_TRC;
END
GO
CREATE PROCEDURE [dbo].[sp_Ders_Add_Update_Remove]
    @operation VARCHAR(10),
    @jsonvalues NVARCHAR(MAX)
AS
BEGIN
    SELECT *
    INTO #temp
    FROM OPENJSON(@jsonvalues)
    WITH (
        Id INT,
        Kod NVARCHAR(20),
        Ad NVARCHAR(100),
        Kredi INT
    );

    DECLARE @rowcount INT = NULL;

    IF @operation = 'add'
    BEGIN
        INSERT INTO dbo.Ders (Kod, Ad, Kredi)
        SELECT Kod, Ad, Kredi FROM #temp;

        SET @rowcount = @@ROWCOUNT;
    END

    IF @operation = 'update'
    BEGIN
        UPDATE d
        SET
            d.Kod = t.Kod,
            d.Ad = t.Ad,
            d.Kredi = t.Kredi
        FROM
            #temp t
        JOIN
            dbo.Ders d ON t.Id = d.Id;
```

```
        SET @rowcount = @@ROWCOUNT;
    END

    IF @operation = 'remove'
    BEGIN
        DELETE d
        FROM
            #temp t
        JOIN
            dbo.Ders d ON t.Id = d.Id;

        SET @rowcount = @@ROWCOUNT;
    END

    -- *** DÜZELTME: Buradaki 'N' harfi silindi ***

    SELECT @operation AS [Key], IIF(ISNULL(@rowcount, 0) > 0, 1, 0) AS [Value];
END
GO
CREATE PROCEDURE [dbo].[sp_OgrenciDersKayit_Listele]
    @pagination dbo.Type_Dictionary_String_String READONLY,
    @where dbo.Type_Dictionary_String_String READONLY
AS
BEGIN
    ;
    WITH cte AS (
        SELECT * FROM dbo.View_OgrenciDersKayit
    ),
    cte_TRC AS (
        SELECT COUNT(*) AS TotalRecordCount FROM cte
    )
    SELECT * FROM cte, cte_TRC;
END
GO
CREATE PROCEDURE [dbo].[sp_OgrenciDersKayit_Add_Update_Remove]
    @operation VARCHAR(10),
    @jsonvalues NVARCHAR(MAX)
AS
BEGIN
    SELECT *
    INTO #temp
    FROM OPENJSON(@jsonvalues)
    WITH (
        OgrenciId INT,
        DersId INT,
        KayitTarihi DATE,
        Donem NVARCHAR(20)
    );

    DECLARE @rowcount INT = NULL;

    IF @operation = 'add'
    BEGIN
        INSERT INTO dbo.OgrenciDersKayit (OgrenciId, DersId, KayitTarihi, Donem)
        SELECT OgrenciId, DersId, KayitTarihi, Donem
        FROM #temp;

        SET @rowcount = @@ROWCOUNT;
    END

    IF @operation = 'update'
    BEGIN
        UPDATE odk
        SET
```

```

        odk.KayitTarihi = t.KayitTarihi,
        odk.Donem = t.Donem
    FROM
        #temp t
    JOIN
        dbo.OgrenciDersKayit odk ON t.OgrenciId = odk.OgrenciId AND t.DersId =
odk.DersId;

    SET @rowcount = @@ROWCOUNT;
END

IF @operation = 'remove'
BEGIN
    DELETE odk
    FROM
        #temp t
    JOIN
        dbo.OgrenciDersKayit odk ON t.OgrenciId = odk.OgrenciId AND t.DersId =
odk.DersId;

    SET @rowcount = @@ROWCOUNT;
END

SELECT @operation AS [Key], IIF(ISNULL(@rowcount, 0) > 0, 1, 0) AS [Value];
END
GO
CREATE PROCEDURE [dbo].[sp_Student_Lookup]
AS
BEGIN
    SELECT Id, Ad + ' ' + Soyad AS AdSoyad FROM dbo.Student ORDER BY Ad;
END
GO
CREATE PROCEDURE [dbo].[sp_Ders_Lookup]
AS
BEGIN
    SELECT Id, '[' + Kod + ']' + Ad AS Ad FROM dbo.Ders ORDER BY Kod;
END
GO

```

Veritabanında bulunan kullanıcı tanımlı veri tiplerinin (type) oluşturma SQL ifadelerini yazınız.

```

CREATE TYPE [dbo].[Type_Dictionary_String_String] AS TABLE
(
    [Key] [nvarchar](100) NULL,
    [Value] [nvarchar](max) NULL
)
GO

```

Veritabanında bulunan indislerin (index) oluşturma SQL ifadelerini yazınız.

```

CREATE UNIQUE NONCLUSTERED INDEX [IX_Student_Email] ON [dbo].[Student]
(
    [Email] ASC
)
WHERE [Email] IS NOT NULL;
GO
CREATE UNIQUE NONCLUSTERED INDEX [IX_Student_Numara] ON [dbo].[Student]
(
    [Numara] ASC
)
WHERE [Numara] IS NOT NULL;
GO
CREATE UNIQUE NONCLUSTERED INDEX

```

Veritabanı diyagramı oluşturunuz ve resim olarak ekleyiniz

(AspNet... ile başlayan tabloları diyagramda göstermeyiniz. Tablolar arası ilişkileri diyagramda gösteriniz)

