

Proje Adı	Gelir Gider Takip Sistemi
Proje Konusu	Kullanıcının gelir ve giderlerini takip edip rapor oluşturabileceği bir sistem
Veritabanı Adı	SAT242516021

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

(AspNet... ile başlayan tablolar yazılmayacak)

```

CREATE TABLE kullanicı (
    kullanicı_id INT IDENTITY(1,1) PRIMARY KEY,
    ad NVARCHAR(50) NOT NULL,
    email NVARCHAR(100) UNIQUE,
    parola NVARCHAR(100) NOT NULL
);
CREATE TABLE kategori (
    kategori_id INT IDENTITY(1,1) PRIMARY KEY,
    kategori_adi NVARCHAR(50) NOT NULL,
    tur NVARCHAR(10) NOT NULL
);
CREATE TABLE gelir (
    gelir_id INT IDENTITY(1,1) PRIMARY KEY,
    kullanicı_id INT NOT NULL,
    kategori_id INT NOT NULL,
    miktar DECIMAL(10,2) NOT NULL,
    tarih DATE NOT NULL,
    açıklama NVARCHAR(255),
    FOREIGN KEY (kullanicı_id) REFERENCES kullanicı(kullanicı_id),
    FOREIGN KEY (kategori_id) REFERENCES kategori(kategori_id)
);
CREATE TABLE gider (
    gider_id INT IDENTITY(1,1) PRIMARY KEY,
    kullanicı_id INT NOT NULL,
    kategori_id INT NOT NULL,
    miktar DECIMAL(10,2) NOT NULL,
    tarih DATE NOT NULL,
    açıklama NVARCHAR(255),
    FOREIGN KEY (kullanicı_id) REFERENCES kullanicı(kullanicı_id),
    FOREIGN KEY (kategori_id) REFERENCES kategori(kategori_id)
);

```

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

```

CREATE VIEW vw_GelirGiderOzet AS
SELECT
    u.ad AS KullaniciAd,
    k.kategori_adi AS Kategori,
    k.tur AS Tur,
    ISNULL(g.miktar, 0) AS GelirMiktar,
    ISNULL(d.miktar, 0) AS GiderMiktar,
    COALESCE(g.tarih, d.tarih) AS Tarih
FROM kullanicı u
LEFT JOIN gelir g ON u.kullanici_id = g.kullanici_id
LEFT JOIN gider d ON u.kullanici_id = d.kullanici_id

```

```

LEFT JOIN kategori k ON (k.kategori_id = g.kategori_id OR k.kategori_id = d.kategori_id);
GO

CREATE VIEW vw_GelirListesi AS
SELECT
    g.gelir_id,
    u.ad AS KullaniciAd,
    k.kategori_adi AS Kategori,
    g.miktar,
    g.tarih,
    g.aciklama
FROM gelir g
JOIN kullanici u ON g.kullanici_id = u.kullanici_id
JOIN kategori k ON g.kategori_id = k.kategori_id;
GO

CREATE VIEW vw_GiderListesi AS
SELECT
    d.gider_id,
    u.ad AS KullaniciAd,
    k.kategori_adi AS Kategori,
    d.miktar,
    d.tarih,
    d.aciklama
FROM gider d
JOIN kullanici u ON d.kullanici_id = u.kullanici_id
JOIN kategori k ON d.kategori_id = k.kategori_id;
GO

```

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

```

CREATE PROCEDURE sp_GetGelirler
    @KullaniciID INT = NULL,
    @KategoriID INT = NULL,
    @BaslangicTarih DATE = NULL,
    @BitisTarih DATE = NULL,
    @SiralaSutun NVARCHAR(50) = 'tarih',
    @SiralaYon NVARCHAR(4) = 'DESC',
    @Sayfa INT = 1,
    @SayfaBoyutu INT = 5
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @Offset INT = (@Sayfa - 1) * @SayfaBoyutu;

    SELECT
        g.gelir_id,
        u.ad AS KullaniciAd,
        k.kategori_adi AS Kategori,

```

```

g.miktar,
g.tarih,
g.aciklama
FROM gelir g
INNER JOIN kullanici u ON g.kullanici_id = u.kullanici_id
INNER JOIN kategori k ON g.kategori_id = k.kategori_id
WHERE (@KullaniciID IS NULL OR g.kullanici_id = @KullaniciID)
AND (@KategoriID IS NULL OR g.kategori_id = @KategoriID)
AND (@BaslangicTarih IS NULL OR g.tarih BETWEEN @BaslangicTarih AND ISNULL(@BitisTarih, GETDATE()))
ORDER BY
CASE WHEN @SiralaSutun = 'miktar' AND @SiralaYon = 'ASC' THEN g.miktar END ASC,
CASE WHEN @SiralaSutun = 'miktar' AND @SiralaYon = 'DESC' THEN g.miktar END DESC,
CASE WHEN @SiralaSutun = 'tarih' AND @SiralaYon = 'ASC' THEN g.tarih END ASC,
CASE WHEN @SiralaSutun = 'tarih' AND @SiralaYon = 'DESC' THEN g.tarih END DESC
OFFSET @Offset ROWS FETCH NEXT @SayfaBoyutu ROWS ONLY;
END;
GO

```

```

CREATE PROCEDURE sp_GetGiderler
@KullaniciID INT = NULL,
@KategoriID INT = NULL,
@BaslangicTarih DATE = NULL,
@BitisTarih DATE = NULL,
@SiralaSutun NVARCHAR(50) = 'tarih',
@SiralaYon NVARCHAR(4) = 'DESC',
@Sayfa INT = 1,
@SayfaBoyutu INT = 5
AS
BEGIN
SET NOCOUNT ON;

DECLARE @Offset INT = (@Sayfa - 1) * @SayfaBoyutu;

SELECT
d.gider_id,
u.ad AS KullaniciAd,
k.kategori_adi AS Kategori,
d.miktar,
d.tarih,
d.aciklama
FROM gider d
INNER JOIN kullanici u ON d.kullanici_id = u.kullanici_id
INNER JOIN kategori k ON d.kategori_id = k.kategori_id
WHERE (@KullaniciID IS NULL OR d.kullanici_id = @KullaniciID)
AND (@KategoriID IS NULL OR d.kategori_id = @KategoriID)
AND (@BaslangicTarih IS NULL OR d.tarih BETWEEN @BaslangicTarih AND ISNULL(@BitisTarih, GETDATE()))
ORDER BY
CASE WHEN @SiralaSutun = 'miktar' AND @SiralaYon = 'ASC' THEN d.miktar END ASC,
CASE WHEN @SiralaSutun = 'miktar' AND @SiralaYon = 'DESC' THEN d.miktar END DESC,
CASE WHEN @SiralaSutun = 'tarih' AND @SiralaYon = 'ASC' THEN d.tarih END ASC,

```

```

CASE WHEN @SiralaSutun = 'tarih' AND @SiralaYon = 'DESC' THEN d.tarih END DESC
OFFSET @Offset ROWS FETCH NEXT @SayfaBoyutu ROWS ONLY;
END;
GO

```

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

```
CREATE TYPE para_miktari FROM DECIMAL(10,2) NOT NULL;
```

```
GO
```

```
CREATE TYPE kategori_turu FROM NVARCHAR(10) NOT NULL;
```

```
GO
```

```
CREATE TABLE ornek_kayit (
    id INT IDENTITY(1,1) PRIMARY KEY,
    tur kategori_turu,
    miktar para_miktari
);
```

Veritabanında bulunan indislerin (index) oluşturma SQL ifadelerini yazınız.

```
CREATE INDEX idx_kullanici_ad
```

```
ON kullanici(ad);
```

```
GO
```

```
CREATE INDEX idx_kategori_adi
```

```
ON kategori(kategori_adi);
```

```
GO
```

```
CREATE INDEX idx_gelir_tarih
```

```
ON gelir(tarih);
```

```
GO
```

```
CREATE INDEX idx_gider_tarih
```

```
ON gider(tarih);
```

```
GO
```

Veritabanı diyagramı oluşturunuz ve resim olarak ekleyiniz

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

