



**T.C
FIRAT ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**FESTİVAL VE BİLET SATIŞ OTOMASYONU
PROJESİ**

RAPOR

YUNUS EMRE BAYCAN 220260303

HASAN AKYÜZ 220260047

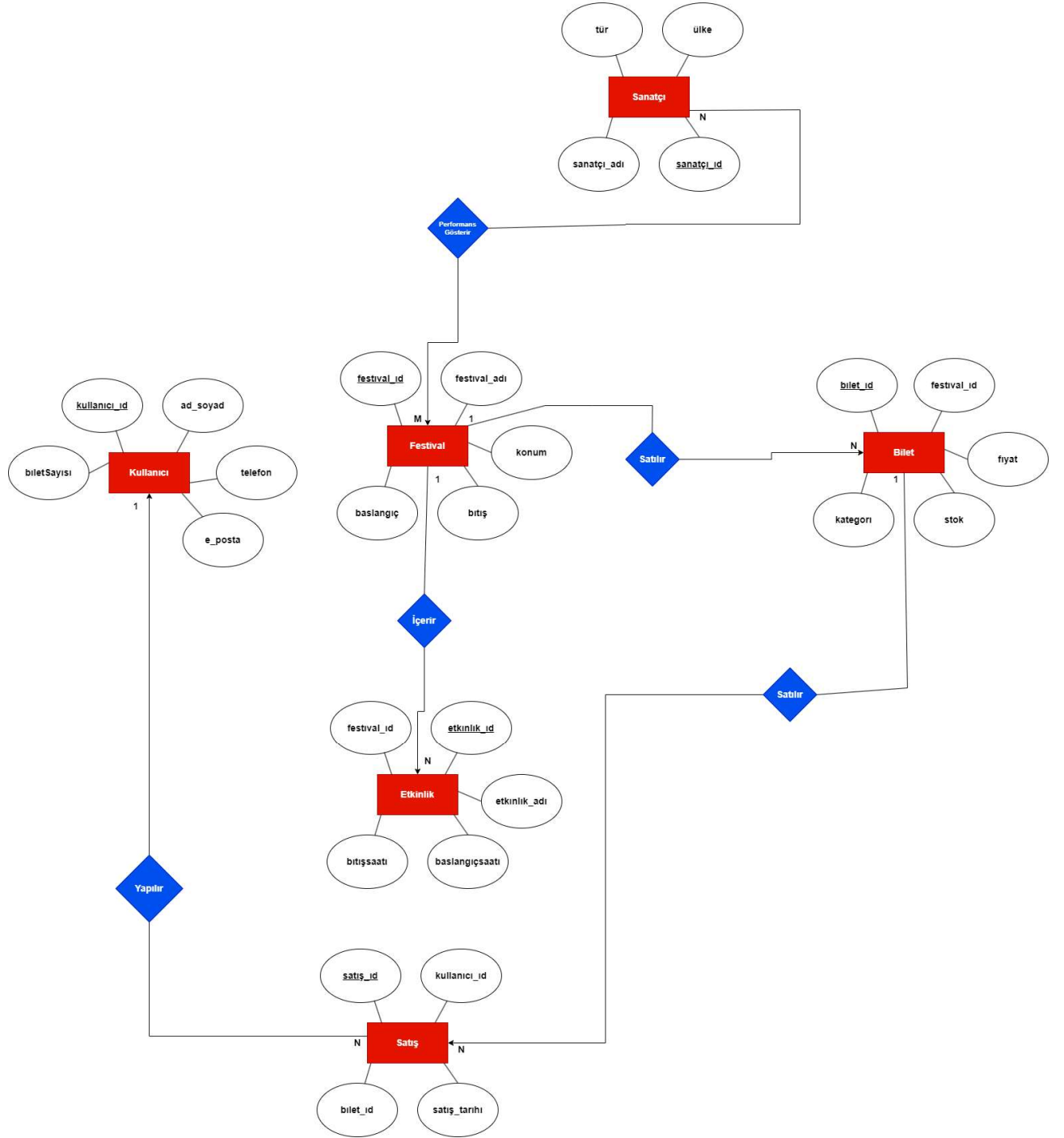
GÖRKEM TÜRK 220260033

1. Proje Konusu

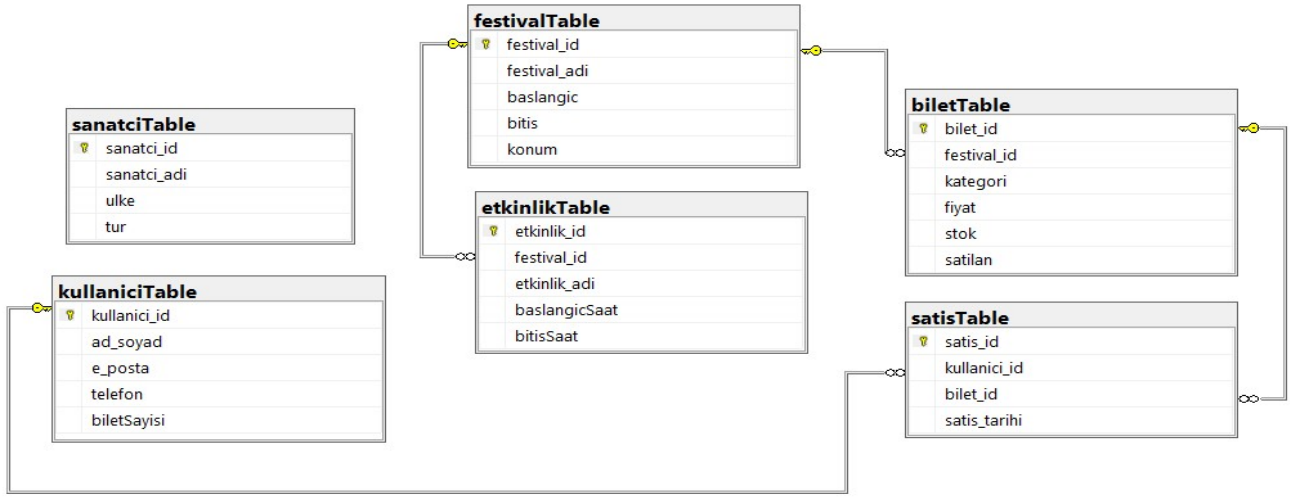
Bu projede bir **Festival Ve Bilet Satış Otomasyonu** geliştirilecektir. Bu sistem, sanatçıların, kullanıcıların, festivallerin, etkinliklerin, biletlerin ve satışların yönetilmesini sağlayacaktır. Sistem, kullanıcıların festival ve etkinlik bilgilerini görüntülemesini, bilet satın almasını ve sanatçılar hakkında bilgi edinmesini sağlarken, yöneticilerin de festival ve etkinlikleri düzenleyip izlemelerine olanak tanır.

2. E-R Diyagramı-Veri Tabanı Şeması-İlişkisel Veri Modeli

Projenin E-R diyagramı:



Projenin veri tabanı şeması:



Projenin ilişkisel veri modeli:

- **Kullanıcı**(kullanıcı_id,biletSayısı,ad_soyad,telefon,e_posta)
- **Satış**(satış_id,kullanıcı_id,bilet_id,satış_tarihi)
- **Etkinlik**(etkinlik_id,festival_id,etkinlik_adi,başlangıçsaatı,bitişsaatı)
- **Festival**(festival_id,festival_adi,konum,bitiş,başlangıç)
- **Sanatçı**(sanatçı_id,sanatçı_adi,tür,ülke)
- **Bilet**(bilet_id,festival_id,fiyat,stok,kategori)

3. Varlıkların Nitelikleri, İlişkileri ve Kısıtlamaları

Kullanıcı Tablosu

- **Gereksinimler:** Bir kullanıcı profil yönetimini sağlayabilir, bilet satın alma işlemlerini ve kendi geçmiş kayıt takibini yapabilir.
- **Kısıtlamalar:**
 - **Kullanıcı ID:** Her kullanıcı bir kimlik numarasına sahip olmalı (Primary Key).
 - **Ad-Soyadı:** Kullanıcı adı ve soyadı boş olamaz
 - **Telefon:** Telefon numarası benzersiz olmalı ve boş olamaz.
 - **E-posta:** E-posta adresi, benzersiz olmalı ve boş olamaz.
 - **Bilet Sayısı:** Bilet sayısı negatif değer alamaz, pozitif bir tam sayı değeri olmalı ve boş olamaz.

Festival Tablosu

- **Gereksinimler:** Festivale ekleme, silme, güncelleme, detayları görüntüleme ve festivalin programını oluşturulması yapılabilir.
- **Kısıtlamalar:**
 - **Festival ID:** Festival bir kimlik numarasına sahip olmalı (Primary Key).
 - **Festival Adı:** Festival adı boş olamaz
 - **Konum:** Konum boş olamaz.
 - **Başlangıç:** Bir festivalin başlangıç tarihi ve boş olamaz.
 - **Bitiş:** Festivalin bitiş tarihi, başlangıç tarihinden sonra olmalıdır.

Etkinlik Tablosu

- **Gereksinimler:** Etkinliklerin düzenlenmesi, hangi festivalde ve nerede olması gerektiği, etkinliklerin zamanlarının programlanması yapılabilir.
- **Kısıtlamalar:**
 - Etkinlik ID: Etkinlik bir kimlik numarasına sahip olmalı (Primary Key).
 - Festival ID: Festival bir kimlik numarasına sahip olmalı (Foreign Key). **Festival** tablosuna referans.
 - Etkinlik Adı: Etkinlik adı boş olamaz.
 - Başlangıç Saati: Etkinliğin başlangıç saati festivalin tarihleri arasında olmalı ve boş olamaz.
 - Bitiş Saati: Etkinliğin bitiş saati festivalin tarihleri arasında olmalıdır.

Bilet Tablosu

- **Gereksinimler:** Biletlerin kategorileştirilmesi ve fiyat hesaplamaları, stok takibi ve satış işlemleri yapılabilir.
- **Kısıtlamalar:**
 - Bilet ID: Bilet bir kimlik numarasına sahip olmalı (Primary Key).
 - Festival ID: Festival bir kimlik numarasına sahip olmalı (Foreign Key). **Festival** tablosuna referans.
 - Fiyat: Fiyat pozitif bir tam sayı olmalıdır ve boş olamaz.
 - Stok: Stok negatif bir tam sayı olamaz.
 - Kategori: Biletlerin kategorileri.

Satış Tablosu

- **Gereksinimler:** Satış personellerinin kayıt ve profillerinin güncellenmesi, bilet satın alan kullanıcıları bilgilerin tutulmasını yapabilir.
- **Kısıtlamalar:**
 - Satış ID: Satış bir kimlik numarasına sahip olmalı (Primary Key).
 - Kullanıcı ID: Her kullanıcı bir kimlik numarasına sahip olmalı (Foreign Key). **Kullanıcı** tablosuna referans.
 - Bilet ID: Bilet bir kimlik numarasına sahip olmalı (Foreign Key). **Bilet** tablosuna referans.
 - Satış Tarihi: Biletin satış tarihi boş olamaz.

Sanatçı Tablosu

- **Gereksinimler:** Her sanatçını kayıt edilmesi, profil yönetimi, sanatçıların ilgili festivallerle eşleştirilmesi ve performanslarının programlanması yapılabilir.
- **Kısıtlamalar:**
 - Sanatçı ID: Sanatçı bir kimlik numarasına sahip olmalı (Primary Key).
 - Sanatçı Adı: Sanatçı adı boş olamaz.
 - Tür: Her sanatçı bir türe sahip olmalı ve boş olamaz.
 - Ülke: Her sanatçı bir ülkeye bağlı olmalı ve boş olamaz.

4. Temel İlişkiler

1. Sanatçı-Festival İlişkisi

- n:m (Çoktan- Çok) ilişki.
- Bir sanatçı birden fazla festivalde performans verebilir.
- Bir festivalde birden fazla sanatçı performans verebilir.

2. Festival-Etkinlik İlişkisi

- 1:n (Birden- Çok) ilişki.
- Bir festival birden fazla etkinlik içerebilir.
- Her etkinlik sadece bir festivale ait olabilir.

3. Festival-Bilet İlişkisi

- 1:n (Birden- Çok) ilişki.
- Bir festival için birden fazla bilet türü olabilir.
- Her bilet sadece bir festivale ait olabilir.

4. Kullanıcı-Satış İlişkisi

- 1:n (Birden- Çok) ilişki.
- Bir kullanıcı birden fazla bilet satın alabilir.
- Her satış işlemi bir kullanıcıya aittir.

5. Bilet-Satış İlişkisi

- 1:n (Birden- Çok) ilişki.
- Bir satışta birden fazla bilet satılabilir.
- Her bilet sadece bir satışta satılır.

5. Otomasyon Roller ve Gereksinimleri

Projede farklı roller için farklı gereksinimler olacaktır. Proje de kullanıcı rolleri ve her bir rol için yetki ve gereksinimler aşağıdaki şekilde belirlenmiştir:

1. Kullanıcı Rolü:

- Festival programını görüntüleyebilir.
- Bilet satın alabilir.
- Satın aldığı biletleri görüntüleyebilir ve yönetebilir.
- Festival ve etkinlikler hakkında yorumlar yapabilir.
- Kendi profil bilgilerini güncelleyebilir.
- Destek talebi oluşturabilir.
- Geçmiş işlemlerini ve bilet geçmişini görüntüleyebilir.

2. Festival Personeli Rolü:

- Etkinlik giriş-çıkış kontrolü yapabilir.
- Bilet doğrulama işlemlerini gerçekleştirebilir.
- Ziyaretçilere yönlendirme ve destek sağlayabilir.
- Etkinlik alanındaki sorunları raporlayabilir.
- Günlük etkinlik programını görüntüleyebilir.
- Acil durum protokollerini yönetebilir.

3. Etkinlik Koordinatörü Rolü:

- Etkinlik programını düzenleyebilir.
- Sanatçı ve personel koordinasyonunu sağlayabilir.
- Etkinlik alanı düzenini yönetebilir.
- Teknik gereksinimleri takip edebilir.
- Etkinlik raporları oluşturabilir.
- Anlık sorunlara müdahale edebilir.

4. Bilet Satış Sorumlusu Rolü:

- Bilet satışlarını yönetebilir.
- Bilet kategorilerini düzenleyebilir.
- İade ve değişim işlemlerini gerçekleştirebilir.
- Satış raporları oluşturabilir.
- Promosyon kodları tanımlayabilir.
- Müşteri şikayetlerini yönetebilir.

5. Festival Yöneticisi Rolü:

- Tüm festival sürecini yönetebilir.
- Personel atama ve yetkilendirme yapabilir.
- Festival bütçesini yönetebilir.
- Sanatçı anlaşmalarını yönetebilir.
- Sponsorluk işlemlerini yürütebilir.
- Tüm raporlara erişebilir ve analiz edebilir.
- Sistem ayarlarını düzenleyebilir.
- Festival politikalarını belirleyebilir.
- Kriz yönetimi yapabilir.

6. Sanatçı/Performans Ekibi Rolü:

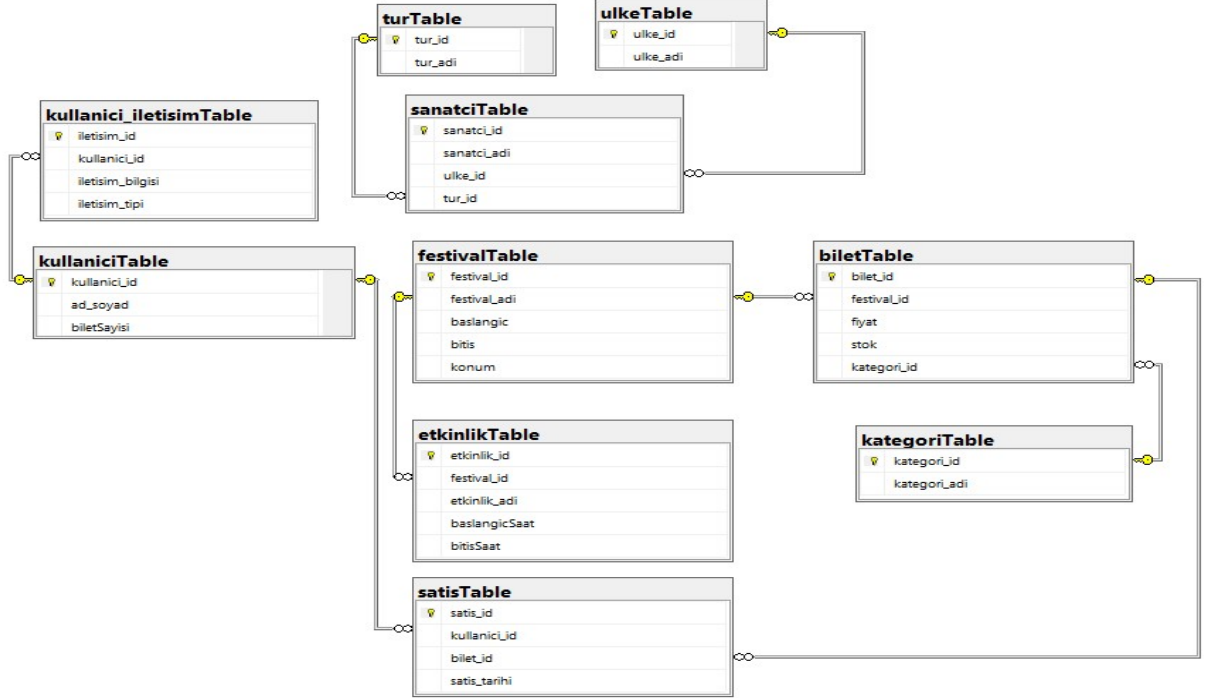
- Kendi performans programını görüntüleyebilir.
- Teknik gereksinimlerini bildirebilir.
- Prova programını yönetebilir.
- Sözleşme detaylarını görüntüleyebilir.
- Ekip listesini güncelleyebilir.

7. Teknik Ekip Rolü:

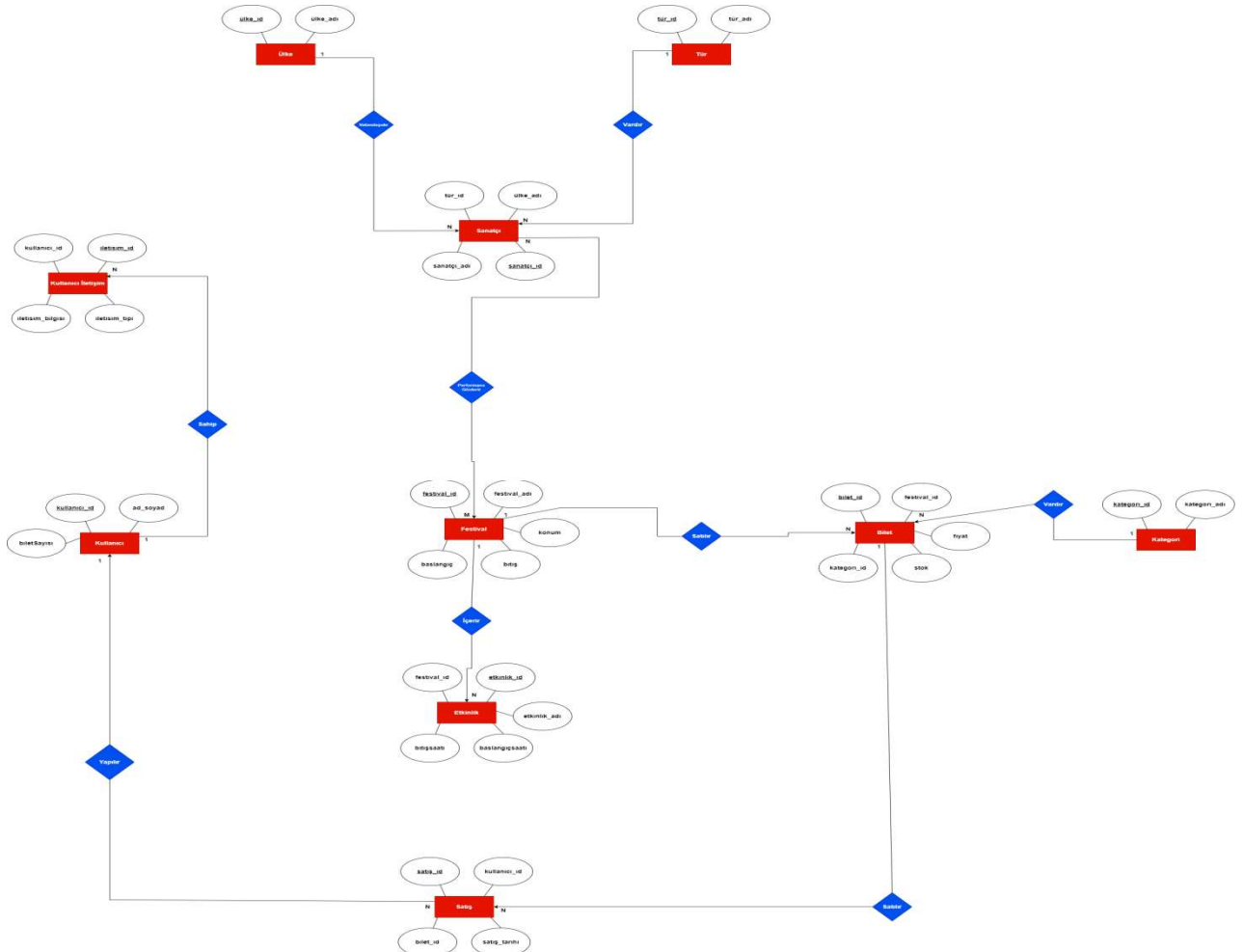
- Teknik ekipman kontrolü yapabilir.
- Ses ve ışık sistemlerini yönetebilir.
- Arıza bildirimlerini takip edebilir.
- Bakım programını düzenleyebilir.
- Teknik raporlar oluşturabilir.

6. Normalizasyonlu E-R Diyagramı-Güncellenmiş Veri Tabanı Şeması-Güncellenmiş İlişkisel Veri Modeli

Projenin güncellenmiş veri tabanı şeması:



Projenin normalize edilmiş E-R diyagramı:



Projenin güncellenmiş ilişkisel veri modeli:

- **Kullanıcı**(kullanıcı_id,biletSayısı,ad_soyad)
- **Satış**(satış_id,kullanıcı_id,bilet_id,satış_tarihi)
- **Etkinlik**(etkinlik_id,festival_id,etkinlik_adı,başlangıçsaatı,bitişsaatı)
- **Festival**(festival_id,festival_adı,konum,bitiş,başlangıç)
- **Sanatçı**(sanatçı_id,sanatçı_adı,tür_id,ülke_id)
- **Bilet**(bilet_id,festival_id,fiyat,stok,kategori_id)
- **Kullanıcı İletişim**(iletisim_id,kullanıcı_id,iletisim_bilgisi,iletisim_tipi)
- **Tür**(tür_id,tür_adı)
- **Ülke**(ülke_id,ülke_adı)
- **Kategori**(kategori_id,kategori_adı)

7. Güncellenmiş Varlıkların Nitelikleri, İlişkileri ve Kısıtlamaları

Kullanıcı Tablosu

- **Gereksinimler:** Bir kullanıcı profil yönetimini sağlayabilir, bilet satın alma işlemlerini ve kendi geçmiş kayıt takibini yapabilir.
- **Kısıtlamalar:**
 - **Kullanıcı ID:** Her kullanıcı bir kimlik numarasına sahip olmalı (Primary Key).
 - **Ad-Soyadı:** Kullanıcı adı ve soyadı boş olamaz
 - **Bilet Sayısı:** Bilet sayısı negatif değer alamaz, pozitif bir tam sayı değeri olmalı ve boş olamaz.

Festival Tablosu

- **Gereksinimler:** Festivale ekleme, silme, güncelleme, detayları görüntüleme ve festivalin programını oluşturulması yapılabilir.
- **Kısıtlamalar:**
 - **Festival ID:** Festival bir kimlik numarasına sahip olmalı (Primary Key).
 - **Festival Adı:** Festival adı boş olamaz
 - **Konum:** Konum boş olamaz.
 - **Başlangıç:** Bir festivalin başlangıç tarihi ve boş olamaz.
 - **Bitiş:** Festivalin bitiş tarihi, başlangıç tarihinden sonra olmalıdır.

Etkinlik Tablosu

- **Gereksinimler:** Etkinliklerin düzenlenmesi, hangi festivalde ve nerede olması gerektiği, etkinliklerin zamanlarının programlanması yapılabilir.
- **Kısıtlamalar:**
 - **Etkinlik ID:** Etkinlik bir kimlik numarasına sahip olmalı (Primary Key).
 - **Festival ID:** Festival bir kimlik numarasına sahip olmalı (Foreign Key). **Festival** tablosuna referans.
 - **Etkinlik Adı:** Etkinlik adı boş olamaz.
 - **Başlangıç Saati:** Etkinliğin başlangıç saati festivalin tarihleri arasında olmalı ve boş olamaz.
 - **Bitiş Saati:** Etkinliğin bitiş saati festivalin tarihleri arasında olmalıdır.

Bilet Tablosu

- **Gereksinimler:** Biletlerin fiyat hesaplamaları, stok takibi ve satış işlemleri yapılabilir.

- **Kısıtlamalar:**
 - Bilet ID: Bilet bir kimlik numarasına sahip olmalı (Primary Key).
 - Festival ID: Festival bir kimlik numarasına sahip olmalı (Foreign Key). **Festival** tablosuna referans.
 - Fiyat: Fiyat pozitif bir tam sayı olmalıdır ve boş olamaz.
 - Stok: Stok negatif bir tam sayı olamaz.
 - Kategori ID: Kategori bir kimlik numarasına sahip olmalı (Foreign Key). **Kategori** tablosuna referans.

Satış Tablosu

- **Gereksinimler:** Satış personellerinin kayıt ve profillerinin güncellenmesi, bilet satın alan kullanıcıları bilgilerin tutulmasını yapabilir.
- **Kısıtlamalar:**
 - Satış ID: Satış bir kimlik numarasına sahip olmalı (Primary Key).
 - Kullanıcı ID: Her kullanıcı bir kimlik numarasına sahip olmalı (Foreign Key). **Kullanıcı** tablosuna referans.
 - Bilet ID: Bilet bir kimlik numarasına sahip olmalı (Foreign Key). **Bilet** tablosuna referans.
 - Satış Tarihi: Biletin satış tarihi boş olamaz.

Sanatçı Tablosu

- **Gereksinimler:** Her sanatçının profil yönetimi, sanatçıların ilgili festivallerle eşleştirilmesi ve performanslarının programlanması yapılabilir.
- **Kısıtlamalar:**
 - Sanatçı ID: Sanatçı bir kimlik numarasına sahip olmalı (Primary Key).
 - Sanatçı Adı: Sanatçı adı boş olamaz
 - Tür ID: Tür bir kimlik numarasına sahip olmalı (Foreign Key). **Tür** tablosuna referans
 - Ülke ID: Ülke bir kimlik numarasına sahip olmalı (Foreign Key). **Ülke** tablosuna referans

Kullanıcı İletişim Tablosu

- **Gereksinimler:** Bir kullanıcı iletişim bilgilerinin güncellemesini ve ilgili olan iletişim türünün seçimini yapabilir.
- **Kısıtlamalar:**
 - İletişim ID: İletişim bir kimlik numarasına sahip olmalı (Primary Key).
 - Kullanıcı ID: Kullanıcı bir kimlik numarasına sahip olmalı (Foreign Key). **Kullanıcı** tablosuna referans
 - İletişim Bilgisi: İletişim bilgisi boş olamaz.
 - İletişim Tipi: İletişim tipi pozitif bir tam sayı ya da sözcüksel bir yapıda olabilir ve boş olamaz.

Ülke Tablosu

- **Gereksinimler:** Sanatçılara ait ülke bilgileri elde edilebilir.
- **Kısıtlamalar:**
 - Ülke ID: Ülke bir kimlik numarasına sahip olmalı (Primary Key).
 - Ülke Adı: Ülke adı boş olamaz.

Tür Tablosu

- **Gereksinimler:** Sanatçıların hangi türe performans gösterdikleri belirlenebilir ve etkinlik ve festival bu duruma göre programlanabilir.
- **Kısıtlamalar:**
 - Tür ID: Tür bir kimlik numarasına sahip olmalı (Primary Key).
 - Tür Adı: Tür adı boş olamaz.

Kategori Tablosu

- **Gereksinimler:** Biletler ilgili yerlere kategorize edilebilir ve kullanıcının istediği durumda rahatça seçim yapmasını sağlar.
- **Kısıtlamalar:**
 - Kategori ID: Kategori bir kimlik numarasına sahip olmalı (Primary Key).
 - Kategori Adı: Kategori adı boş olamaz.

8. Güncellenmiş Temel İlişkiler

1. Sanatçı-Festival İlişkisi

- n:m (Çoktan- Çok) ilişki.
- Bir sanatçı birden fazla festivalde performans verebilir.
- Bir festivalde birden fazla sanatçı performans verebilir.

2. Festival-Etkinlik İlişkisi

- 1:n (Birden- Çok) ilişki.
- Bir festival birden fazla etkinlik içerebilir.
- Her etkinlik sadece bir festivale ait olabilir.

3. Festival-Bilet İlişkisi

- 1:n (Birden- Çok) ilişki.
- Bir festival için birden fazla bilet türü olabilir.
- Her bilet sadece bir festivale ait olabilir.

4. Kullanıcı-Satış İlişkisi

- 1:n (Birden- Çok) ilişki.
- Bir kullanıcı birden fazla bilet satın alabilir.
- Her satış işlemi bir kullanıcıya aittir.

5. Bilet-Satış İlişkisi

- 1:n (Birden- Çok) ilişki.
- Bir satışta birden fazla bilet satılabilir.
- Her bilet sadece bir satışta satılır.

6. Bilet-Kategori İlişkisi

- 1:n (Birden- Çok) ilişki.
- Bir kategoride birden fazla bilet olabilir.
- Her bilet sadece bir kategoride bulunur.

7. Sanatçı-Ülke İlişkisi

- 1:n (Birden- Çok) ilişki.
- Bir ülkede birden fazla sanatçı olabilir.
- Her sanatçı sadece bir ülkenin vatandaşıdır.

8. Sanatçı-Tür İlişkisi

- 1:n (Birden- Çok) ilişki.
- Bir türde birden fazla sanatçı olabilir.
- Her sanatçı sadece bir türe sahiptir.

9. Kullanıcı-Kullanıcı İletişim İlişkisi

- 1:n (Birden- Çok) ilişki.
- Bir kullanıcı birden fazla iletişime sahip olabilir.
- Her iletişim sadece bir kullanıcıya aittir.

9. SQL Komutaları Ve Örnekler

Tablo Oluşturma

```
satisTable = satis_id, kullanıcı_id, bilet_id, satis_tarihi
biletTable = bilet_id, festival_id, fiyat, stok, kategori_id
kullanıcıTable = kullanıcı_id, ad_soyad, biletSayisi
festivalTable = festival_id, festival_adi, baslangic, bitis, konum
turTable = tur_id, tur_adi
ulkeTable = ulke_id, ulke_adi
iletisimTable = iletisim_id, kullanıcı_id, iletisim_bilgisi, iletisim_tipi
etkinlikTable = etkinlik_id, festival_id, etkinlik_adi, baslangisSaat, bitisSaat
kategoriTable = kategori_id, kategori_adi
sanatciTable = sanatci_id, sanatci_adi, ulke_id, tur_id
```

-- Kullanıcı Tablosu

```
CREATE TABLE kullanıcıTable (
    kullanıcı_id INT PRIMARY KEY,
    ad_soyad VARCHAR(50),
    biletSayisi INT DEFAULT 0
);
```

-- Ülke Tablosu

```
CREATE TABLE ulkeTable (
    ulke_id INT PRIMARY KEY,
    ulke_adi VARCHAR(50)
);
```

-- Tür Tablosu

```
CREATE TABLE turTable (
    tur_id INT PRIMARY KEY,
    tur_adi VARCHAR(50)
);
```

-- Festival Tablosu

```
CREATE TABLE festivalTable (  
    festival_id INT PRIMARY KEY,  
    festival_adi VARCHAR(100),  
    baslangic DATE,  
    bitis DATE,  
    konum VARCHAR(15)  
);
```

-- Kullanıcı İletişim Tablosu

```
CREATE TABLE kullanıcı_iletisimTable (  
    iletisim_id INT PRIMARY KEY,  
    kullanıcı_id INT,  
    iletisim_bilgisi VARCHAR(250),  
    iletisim_tipi VARCHAR(50),  
    FOREIGN KEY (kullanıcı_id) REFERENCES kullanıcıTable(kullanıcı_id)  
);
```

-- Kategori Tablosu

```
CREATE TABLE kategoriTable (  
    kategori_id INT PRIMARY KEY,  
    kategori_adi VARCHAR(100)  
);
```

-- Sanatçı Tablosu

```
CREATE TABLE sanatciTable (  
    sanatci_id INT PRIMARY KEY,  
    sanatci_adi VARCHAR(50),  
    ulke_id INT,  
    tur_id INT,  
    FOREIGN KEY (ulke_id) REFERENCES ulkeTable(ulke_id),  
    FOREIGN KEY (tur_id) REFERENCES turTable(tur_id)  
);
```

-- Etkinlik Tablosu

```
CREATE TABLE etkinlikTable (  
    etkinlik_id INT PRIMARY KEY,  
    festival_id INT,  
    etkinlik_adi VARCHAR(50),  
    baslangisSaat TIME,  
    bitisSaat TIME),  
    FOREIGN KEY (festival_id) REFERENCES festivalTable(festival_id)  
);
```

-- Bilet Tablosu

```
CREATE TABLE biletTable (  
    bilet_id INT PRIMARY KEY,  
    festival_id INT,  
    fiyat INT,  
    kategori_id INT,  
    FOREIGN KEY (festival_id) REFERENCES festivalTable(festival_id),  
    FOREIGN KEY (kategori_id) REFERENCES kategoriTable(kategori_id)  
);
```

-- Satış Tablosu

```
CREATE TABLE satisTable (  
    satis_id INT PRIMARY KEY,  
    kullanıcı_id INT,  
    bilet_id INT,  
    satis_tarihi DATE,  
    FOREIGN KEY (kullanıcı_id) REFERENCES kullanıcıTable(kullanıcı_id),  
    FOREIGN KEY (bilet_id) REFERENCES biletTable(bilet_id)  
);
```

Trigger-Transaction-Stored Procedure

--SATIŞ BİLGİSİ GETİRME STORE PROCEDURE

```
create proc sp_satisBilgisi  
as  
begin  
SELECT  
    s.satis_id,  
    s.satis_tarihi,  
    k.ad_soyad,  
    k.biletSayisi,  
    b.fiyat,  
    b.stok,  
    b.kategori_id,  
    f.festival_adi,  
    f.baslangic,  
    f.bitis,  
    f.konum  
FROM  
    satisTable s  
JOIN  
    biletTable b ON s.bilet_id = b.bilet_id  
JOIN  
    kullanıcıTable k ON s.kullanıcı_id = k.kullanıcı_id  
JOIN  
    festivalTable f ON b.festival_id = f.festival_id;  
end
```

--ÇALIŞTIRMA KOMUTU

```
exec sp_satisBilgisi
```

-- KİŞİ BİLET SAYISI ARTTIRMA TRİGGER

```
create trigger biletSayisi  
on satisTable  
after insert  
as  
begin  
    declare @kullanıcıid int  
    select @kullanıcıid=kullanıcı_id from inserted  
    update kullanıcıTable set biletSayisi = biletSayisi +1 where kullanıcı_id = @kullanıcıid  
  
end
```

--STOKTAN BİLET AZALTMA TRİGGER

-- STOK TAKİP

```
create trigger stokAzaltma
on satisTable
after insert
as
begin
    declare @biletid int
    select @biletid=bilet_id from inserted
    update biletTable set stok = stok -1 where bilet_id = @biletid

end
```

--BİLET SAYISI AZALTMA TRİGGER

```
CREATE TRIGGER biletSayisiAzalt
ON satisTable
AFTER DELETE
AS
BEGIN
    -- Silinen kayıttaki kullanıcı_id'yi al
    DECLARE @kullaniciid INT;

    SELECT @kullaniciid = kullanıcı_id FROM deleted;

    -- Kullanıcının bilet sayısını 1 azalt
    UPDATE kullanıcıTable
    SET biletSayisi = biletSayisi - 1
    WHERE kullanıcı_id = @kullaniciid;
END;
```

--BİLET SAYISI ARTTIRMA

```
create TRIGGER biletArttırma
ON satisTable
AFTER DELETE
AS
BEGIN
    DECLARE @biletid INT;

    SELECT @biletid = bilet_id FROM deleted;

    UPDATE biletTable
    SET stok = stok + 1
    WHERE bilet_id = @biletid;
END;
```

--SATIŞ PROCEDURE VE TRANSACTION DAHİL SATIŞ EKLEME

```
CREATE PROCEDURE sp_satisEkle
    @kullanici_id INT,
    @bilet_id INT,
    @tarih DATETIME
AS
BEGIN
    BEGIN TRANSACTION;
```

```

-- Stok kontrolü

IF (SELECT stok FROM biletTable WHERE bilet_id = @bilet_id) <= 0
BEGIN
    ROLLBACK;
    SELECT 'Bilet Stok Tükendi' AS Mesaj;
    RETURN;
END

-- Satış işlemi
INSERT INTO satisTable (kullanici_id, bilet_id, satis_tarihi)
VALUES (@kullanici_id, @bilet_id, @tarih);

COMMIT;
SELECT 'Satış Başarılı' AS Mesaj;
END;

-----

--ÇALIŞTIRMA ÖRNEK KOD

exec sp_satisEkle
    @kullanici_id = 101,
    @bilet_id = 20003,
    @tarih = '2025-01-04';

```

Tablo Örnek Veri Ekleme

-- Kullanıcı Tablosu Örnek Veriler

```

INSERT INTO kullanıcıTable (kullanici_id, ad_soyad) VALUES
(1, 'Ali Veli'),
(2, 'Ayşe Yılmaz'),
(3, 'Ahmet Kaya');

```

-- Ülke Tablosu Örnek Veriler

```

INSERT INTO ulkeTable (ulke_id, ulke_adi) VALUES
(1, 'Türkiye'),
(2, 'Birleşik Krallık'),
(3, 'ABD');

```

-- Tür Tablosu Örnek Veriler

```

INSERT INTO turTable (tur_id, tur_adi) VALUES
(1, 'Rock'),
(2, 'Pop'),
(3, 'Jazz');

```

-- Festival Tablosu Örnek Veriler

```

INSERT INTO festivalTable (festival_adi, baslangic, bitis, konum) VALUES
('Rock Fest', '10:00', '22:00', 'Ankara'),
('Pop Fest', '12:00', '20:00', 'İstanbul'),
('Caz Fest', '14:00', '23:00', 'İzmir');

```

-- Kullanıcı İletişim Tablosu Örnek Veriler

```
INSERT INTO kullanıcı_iletisimTable (iletisim_id, kullanıcı_id, iletisim_bilgisi, iletisim_tipi) VALUES
(1, 101, 'ali@gmail.com', 'Email'),
(2, 102, '555-1234', 'Telefon'),
(3, 103, '@ahmetkaya', 'Twitter');
```

-- Kategori Tablosu Örnek Veriler

```
INSERT INTO kategoriTable (kategori_id, kategori_adi) VALUES
(1, 'VIP'),
(2, 'Standart'),
(3, 'Öğrenci');
```

-- Sanatçı Tablosu Örnek Veriler

```
INSERT INTO sanatciTable (sanatci_id, sanatci_adi, ulke_id, tur_id) VALUES
(1, 'Metallica', 1, 1),
(2, 'Tarkan', 1, 2),
(3, 'Miles Davis', 2, 3);
```

-- Etkinlik Tablosu Örnek Veriler

```
INSERT INTO etkinlikTable (festival_id, etkinlik_adi, baslangisSaat, bitisSaat) VALUES
(1, 'Metallica Konseri', '10:30', '12:00'),
(2, 'Tarkan Sahne', '13:00', '15:00'),
(3, 'Miles Davis Performansı', '16:00', '18:00');
```

-- Bilet Tablosu Örnek Veriler

```
INSERT INTO biletTable (festival_id, fiyat, kategori_id) VALUES
(1, 300, 1),
(2, 200, 2),
(3, 150, 3);
```

-- Satış Tablosu Örnek Veriler

```
INSERT INTO satisTable (kullanıcı_id, bilet_id, satis_tarihi) VALUES
(1, 1, '2025-01-01'),
(2, 2, '2025-01-02'),
(1, 3, '2025-01-03');
```

--SATIŞ VERİSİ SİLME

```
DELETE FROM satisTable WHERE satis_id = 4
```


Tablolara Örnek Fotoğraflar

1-

```
exec sp_satisEkle @kullanici_id = 104, @bilet_id = 10003, @tarih = '2025-05-01';

select * from biletTable
select * from kullaniciTable
```

82 %

Results Messages

Mesaj

1	Bilet Stok Tükendi
---	--------------------

	bilet_id	festival_id	fiyat	stok	kategori_id
1	10003	1	1500	0	1
2	20003	1	750	987	2
3	30003	1	500	998	3
4	50003	2	1500	1000	1
5	60003	2	750	1000	2
6	70003	2	500	1000	3
7	80003	3	1500	1000	1
8	90003	3	750	1000	2
9	100003	3	500	1000	3
10	110003	4	1500	1000	1
11	120003	4	750	1000	2

	kullanici_id	ad_soyad	biletSayisi
1	101	Ahmet Yıldırım	6
2	102	Ahmet Yılmaz	1
3	103	Onur Çetin	1
4	104	Derya Demir	1
5	105	Kaan Aslan	0
6	106	Merve Yılmaz	0
7	107	Burak Özdemir	2
8	108	Seda Akdoğan	1
9	109	Serhat Kara	0
10	110	Ayhan Çetin	0
11	111	Cemal Aydın	0

2-

```
exec sp_satisEkle @kullanici_id = 104, @bilet_id = 10003, @tarih = '2025-05-01';

select * from biletTable
select * from kullaniciTable
```

82 %

Results Messages

Mesaj

1	Satış Başarılı
---	----------------

	bilet_id	festival_id	fiyat	stok	kategori_id
1	10003	1	1500	1	1
2	20003	1	750	987	2
3	30003	1	500	998	3
4	50003	2	1500	1000	1
5	60003	2	750	1000	2
6	70003	2	500	1000	3
7	80003	3	1500	1000	1
8	90003	3	750	1000	2
9	100003	3	500	1000	3
10	110003	4	1500	1000	1
11	120003	4	750	1000	2

SATIŞ
TABLOSUNA
VERİ EKLEME

	kullanici_id	ad_soyad	biletSayisi
1	101	Ahmet Yıldırım	6
2	102	Ahmet Yılmaz	1
3	103	Onur Çetin	1
4	104	Derya Demir	0
5	105	Kaan Aslan	0
6	106	Merve Yılmaz	0
7	107	Burak Özdemir	2
8	108	Seda Akdoğan	1
9	109	Serhat Kara	0
10	110	Ayhan Çetin	0
11	111	Cemal Aydın	0

3-

```
exec sp_satisEkle @kullanici_id = 104, @bilet_id = 10003, @tarih = '2025-05-01';
```

```
select * from biletTable
```

```
select * from kullanıcıTable
```

82 %

Results Messages

Mesaj	
1	Satış Başarılı

	bilet_id	festival_id	fiyat	stok	kategori_id
1	10003	1	1500	0	1
2	20003	1	750	987	2
3	30003	1	500	998	3
4	50003	2	1500	1000	1
5	60003	2	750	1000	2
6	70003	2	500	1000	3
7	80003	3	1500	1000	1
8	90003	3	750	1000	2
9	100003	3	500	1000	3
10	110003	4	1500	1000	1
11	120003	4	750	1000	2

	kullanici_id	ad_soyad	biletSayisi
1	101	Ahmet Yıldırım	6
2	102	Ahmet Yılmaz	1
3	103	Onur Çetin	1
4	104	Derya Demir	1
5	105	Kaan Aslan	0
6	106	Merve Yılmaz	0
7	107	Burak Özdemir	2
8	108	Seda Akdoğan	1
9	109	Serhat Kara	0
10	110	Ayhan Çetin	0
11	111	Cemal Aydın	0

4-

```
delete from satisTable where satis_id = 29
```

```
select * from biletTable
select * from kullanıcıTable
select * from satisTable
```

82 %

Results Messages

	bilet_id	festival_id	fiyat	stok	kategori_id
1	10003	1	1500	0	1
2	20003	1	750	988	2
3	30003	1	500	998	3
4	50003	2	1500	1000	1
5	60003	2	750	1000	2
6	70003	2	500	1000	3
7	80003	3	1500	1000	1
8	90003	3	750	1000	2
9	100003	2	500	1000	2

	kullanici_id	ad_soyad	biletSayisi
1	101	Ahmet Yıldırım	5
2	102	Ahmet Yılmaz	1
3	103	Onur Çetin	1
4	104	Derya Demir	1
5	105	Kaan Aslan	0
6	106	Merve Yılmaz	0
7	107	Burak Özdemir	2
8	108	Seda Akdoğan	1
9	109	Serhat Kara	0

SATIŞ İPTALİ

	satis_id	kullanici_id	bilet_id	satis_tarihi
15	20	101	20003	2025-01-04
16	21	101	20003	2025-01-04
17	22	101	20003	2025-01-04
18	24	101	20003	2025-01-04
19	25	107	10003	2025-05-01
23	29	104	10003	2025-05-01
20	20	107	10003	2025-05-01
21	28	108	10003	2025-05-01
22	29	104	10003	2025-05-01

5-

```
delete from satisTable where satis_id = 29  
  
select * from biletTable  
select * from kullanıcıTable  
select * from satisTable
```

82 %

Results Messages

	bilet_id	festival_id	fiyat	stok	kategori_id
1	10003	1	1500	1	1
2	20003	1	750	988	2
3	30003	1	500	998	3
4	50003	2	1500	1000	1
5	60003	2	750	1000	2
6	70003	2	500	1000	3
7	80003	3	1500	1000	1
8	90003	3	750	1000	2
9	100003	2	500	1000	2

	kullanici_id	ad_soyad	biletSayisi
1	101	Ahmet Yıldırım	5
2	102	Ahmet Yılmaz	1
3	103	Onur Çetin	1
4	104	Derya Demir	0
5	105	Kaan Aslan	0
6	106	Merve Yılmaz	0
7	107	Burak Özdemir	2
8	108	Seda Akdoğan	1
9	109	Sedat Kara	0

	satis_id	kullanici_id	bilet_id	satis_tarihi
14	19	1412	1800...	2025-01-04
15	20	101	20003	2025-01-04
16	21	101	20003	2025-01-04
17	22	101	20003	2025-01-04
18	24	101	20003	2025-01-04
19	25	107	10003	2025-05-01
20	26	107	10003	2025-05-01
21	28	108	10003	2025-05-01