

FIRAT ÜNİVERSİTESİ
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü

BMÜ329 VERİ TABANI SİSTEMLERİ
DÖNEM PROJESİ RAPORU

Mobil Oyun Veri Tabanı

Grup No	13
Takım Üyesi 1	230260081 - Bilal Tekin
Takım Üyesi 2	220260089 - Yağmur Gür
Takım Üyesi 3	230260007 - Leyla Erdoğan
Ders Sorumlusu	Selen GÜRBÜZ
Teslim Tarihi	27.12.2025

İÇİNDEKİLER

- 1. GİRİŞ VE PROJE TANIMI**
 - 1.1 Projenin Amacı ve Kapsamı
 - 1.2 Hedef Kullanıcılar
- 2. PROJE GEREKSİNİMLERİ**
 - 2.1 Fonksiyonel Gereksinimler
 - 2.2 Fonksiyonel Olmayan Gereksinimler
 - 2.3 İş Kuralları ve Kısıtlamalar
- 3. VERİ TABANI TASARIMI – VARLIK-İLİŞKİ (E-R) MODELİ**
 - 3.1 E-R Diyagramı
- 4. İLİŞKİSEL ŞEMALAR**
 - 4.1 E-R Modelinden İlişkisel Şemalara Dönüşüm
- 5. NORMALİZASYON**
 - 5.1 Normalizasyon Süreci
 - 5.2 Son Normalize Edilmiş Şema
- 6. SQL SERVER VERİ TABANI ŞEMASI**
 - 6.1 Tablo Oluşturma Komutları
 - 6.2 Veri Tabanı Diyagramı
- 7. ÖRNEK VERİLER**
 - 7.1 Veri Ekleme Komutları (INSERT)
- 8. SQL KOMUTLARI VE SCRIPT DOSYALARI**
 - 8.1 Gereksinim Bazlı SQL Komutları
- 9. SAKLI YORDAM VE TETİKLEYİCİ**
 - 9.1 Saklı Yordamlar
 - 9.1.1 Amaç ve İş Operasyonu
 - 9.1.2 Saklı Yordam Kodları
 - 9.1.3 Test Senaryoları ve Sonuçları
 - 9.2 Tetikleyiciler
 - 9.2.1 Amaç ve Tetikleme Koşulu
 - 9.2.2 Tetikleyici Kodları
 - 9.2.3 Test Senaryoları ve Sonuçları
- 10. TRANSACTION YÖNETİMİ**
 - 10.1 Transaction Senaryosu
 - 10.2 Transaction Kodu

10.3 Başarılı Senaryo Testi

10.4 Hata Senaryosu ve ROLLBACK Testi

11. TAKIM ÇALIŞMASI VE GÖREV DAĞILIMI

11.1 Takım Üyeleri ve Rollerİ

11.2 Gerçekleştirilen İşler ve Sorumluluk Matrisi

12. SONUÇ VE DEĞERLENDİRME

1. GİRİŞ VE PROJE TANIMI

1.1 Projenin Amacı ve Kapsamı

Bu projenin amacı, seviye tabanlı bir mobil oyunun kullanıcı etkileşimlerini ve oyun içi verilerini düzenli ve güvenli bir şekilde yönetebilecek bir veri tabanı yönetim sistemi tasarlamaktır. Sistem; **kullanıcı kayıtlarının tutulması, seviyelerin tanımlanması, kullanıcıların elde ettiği skorların saklanması, can (life) yönetimi, oyun içi güçlendirmelerin (power-up) kullanımı ve satın alma işlemlerinin kayıt altına alınması** gibi temel ihtiyaçları karşılamayı hedeflemektedir.

Proje, mobil oyunlarda karşılaşılan **veri tekrarının önlenmesi, veri bütünlüğünün sağlanması ve ölçülebilir bir yapı oluşturulması** problemlerine çözüm sunmaktadır. Bu sayede çok sayıda kullanıcının aynı anda oyunu oynadığı durumlarda bile veriler düzenli ve tutarlı bir biçimde yönetilebilmektedir.

1.2 Hedef Kullanıcılar

Bu veri tabanı sistemi temel olarak aşağıdaki kullanıcı gruplarını hedef almaktadır:

- **Oyuncular:**

Oyunu aktif olarak oynayan kullanıcılardır. Oyuncular; sisteme kayıt olabilir, oyun seviyelerini oynayarak skor elde edebilir, can (life) kullanabilir, oyun içi güçlendirmelerden faydalananabilir ve bu güçlendirmeleri satın alabilir. Oyunculara ait tüm oyun verileri (skorlar, seviye ilerlemeleri, can durumu ve güçlendirme bilgileri) sistem tarafından düzenli bir şekilde saklanır.

- **Oyun Geliştirici:**

Oyunun veri yapısını yöneten ve oyun içeriğini düzenleyen yetkili kullanıcı grubudur. Oyun geliştiriciler; seviyeleri tanımlayabilir, hedef skor ve maksimum hamle sayısı gibi seviye parametrelerini belirleyebilir, oyun içi güçlendirmeleri sisteme ekleyebilir ve kullanıcı davranışlarını analiz etmek amacıyla skor ve oyun verilerini inceleyebilir. Bu kullanıcı grubu, sistemin sürdürülebilir ve dengeli bir şekilde çalışmasından sorumludur.

2. PROJE GEREKSİNİMLERİ

2.1 Fonksiyonel Gereksinimler

ID	Gereksinim Adı	Açıklama
1	Kullanıcı Kayıt	Oyuncular sisteme kullanıcı adı, e-posta ve şifre bilgileri ile kayıt olabilmelidir.
2	Kullanıcı Girişи	Oyuncular sisteme kayıtlı e-posta ve şifre bilgileri ile giriş yapabilmelidir.
3	Seviye Tanımlama	Oyun geliştirici tarafından seviyeler; seviye numarası, hedef skor ve maksimum hamle bilgileri ile tanımlanabilmelidir.
4	Skor Kaydı	Oyuncuların oynadıkları seviyelerde elde ettikleri skorlar ve kullanılan hamle sayısı sistemde saklanabilmelidir.
5	Seviye–Kullanıcı İlişkisi	Her skor kaydı ilgili kullanıcı ve seviye ile ilişkilendirilmelidir.
6	Can Yönetimi	Oyuncuların mevcut can sayısı takip edilmeli ve can kazanım zamanı sisteme tutulabilmelidir.
7	Güçlendirme Tanımlama	Oyun içi güçlendirmeler ad ve açıklama bilgileri ile sisteme eklenebilmelidir.
8	Güçlendirme Satın Alma	Oyuncular oyun içi güçlendirmeleri belirli miktarlarda satın alabilmelidir.
9	Kullanıcı Güçlendirme Envanteri	Oyuncuların sahip olduğu güçlendirmeler ve miktarları sistemde güncel olarak tutulabilmelidir.
10	Satın Alma Kayıtları	Geçerleştirilen tüm satın alma işlemleri tarih ve kullanıcı bilgileri ile kaydedilmelidir.

2.2 Fonksiyonel Olmayan Gereksinimler

ID	Gereksinim Adı	Açıklama
1	Güvenlik	Kullanıcı şifreleri güvenli bir şekilde saklanmalı ve yetkisiz erişimler engellenmelidir.

2	Veri Bütünlüğü	Tablolar arasında referans bütünlüğü (foreign key) kısıtlamaları uygulanmalıdır.
3	Performans	Sistem, aynı anda çok sayıda kullanıcının veri işlemlerini gecikme olmadan gerçekleştirebilmelidir.
4	Ölçeklenebilirlik	Kullanıcı ve seviye sayısının artması durumunda sistem performans kaybı yaşamadan çalışabilmelidir.
5	Yedekleme	Veri kaybını önlemek amacıyla veritabanı düzenli olarak yedeklenebilmelidir.
6	Bakım Kolaylığı	Veri tabanı yapısı, ilerde yeni tablolar ve özellikler eklenmesine uygun şekilde tasarlanmalıdır.

2.3 İş Kuralları, Kısıtlamalar

Bu bölümde, mobil oyun veri tabanı sisteminin doğru ve tutarlı bir şekilde çalışabilmesi için belirlenen iş kuralları ve kısıtlamalar açıklanmıştır:

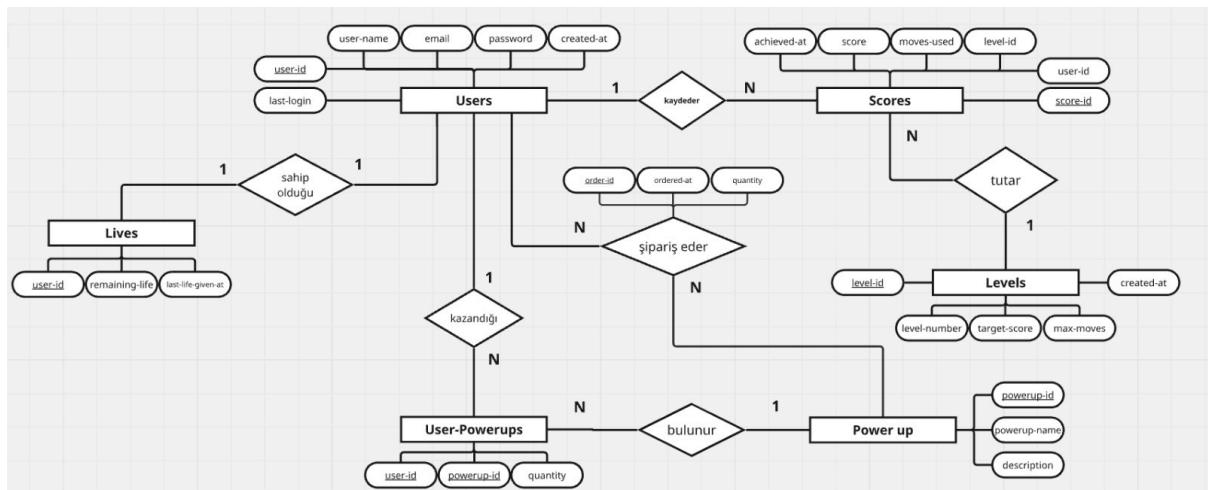
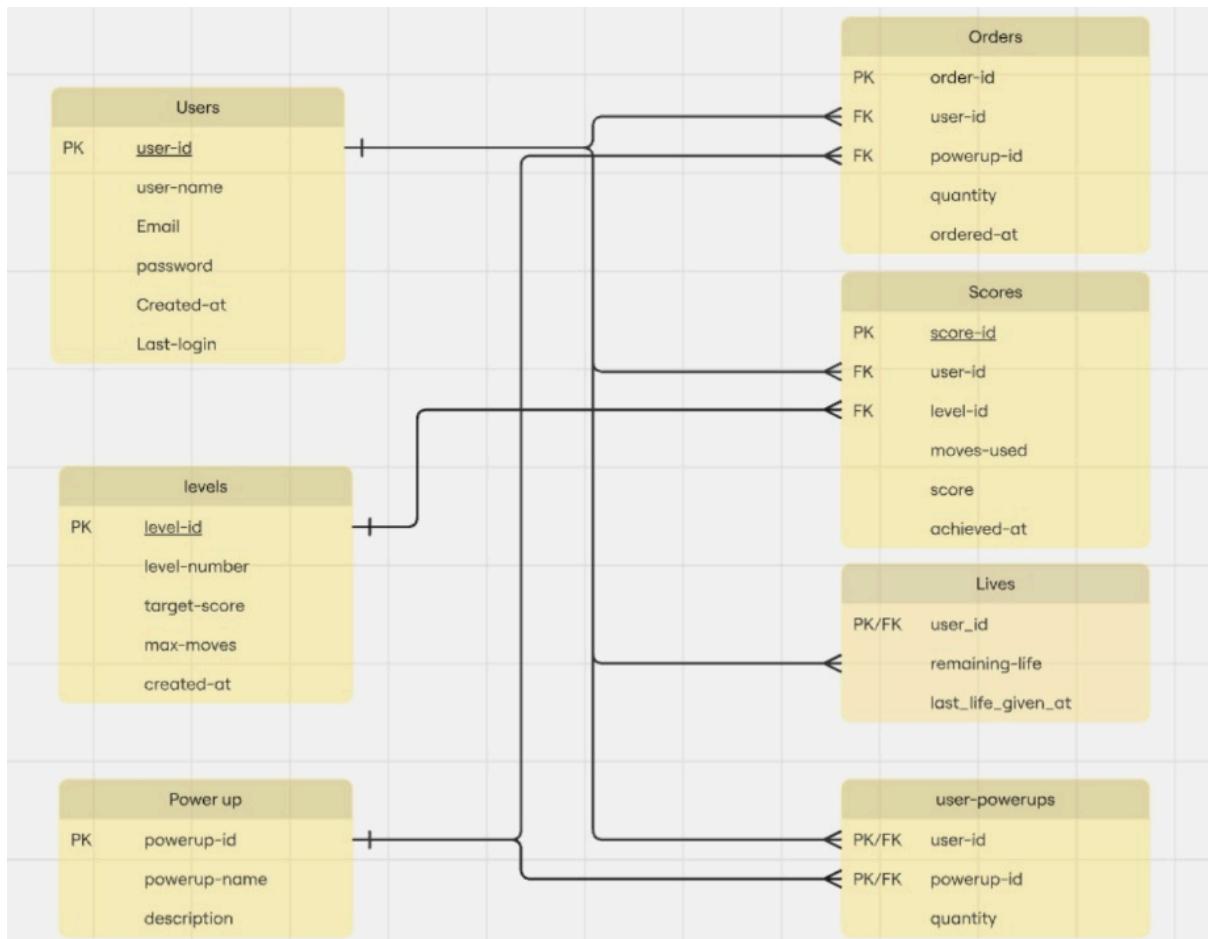
- 1-) Her kullanıcı, sistemde **benzersiz bir kullanıcı kimliği (user_id)** ile tanımlanmalıdır. Aynı e-posta adresi ile birden fazla kullanıcı hesabı oluşturulamaz.
- 2-) Her seviye, sistemde **benzersiz bir seviye numarası (level_id)** ile tanımlanmalı ve aynı seviye numarası birden fazla kez tanımlanmamalıdır.
- 3-) Bir kullanıcı, aynı seviyeyi birden fazla kez oynayabilir; ancak her oyun sonucu **ayrı bir skor kaydı** olarak sistemde tutulmalıdır.
- 4-) Skor kayıtları, mutlaka **geçerli bir kullanıcı (user_id)** ve **geçerli bir seviye (level_id)** ile ilişkilendirilmelidir. Kullanıcı veya seviye silinmeden önce ilgili skor kayıtları dikkate alınmalıdır.
- 5-) Kullanıcıların sahip olabileceği **can (life) sayısı sıfırın altına düşemez**. Can kazanımı ve kullanımı sistem tarafından kontrol edilmelidir.

6-) Kullanıcıların sahip olduğu güçlendirme miktarı **negatif olamaz**. Güçlendirme kullanıldığında miktar azaltılmalı, satın alındığında artırılmalıdır.

7-) Her satın alma işlemi, **geçerli bir kullanıcı ve geçerli bir güçlendirme ile ilişkilendirilmelidir**. Satın alma miktarı sıfırdan büyük olmalıdır.

3. VERİ TABANI TASARIMI - VARLIK-İLİŞKİ (E-R) MODELİ

3.1 E-R Diyagramı



NOT: Şemalar ilk rapordan sonra BCNF'ye uyması için güncellenmiştir.

4. İLİŞKİSEL ŞEMALAR

4.1 E-R'dan İlişkisel Şemalarına Dönüşüm

users (`user_id`, `user_name`, `email`, `password`, `created_at`, `last_login`)

levels (`level_id`, `level_number`, `target_score`, `max_moves`, `created_at`)

powerup (`powerup_id`, `powerup_name`, `description`)

lives (`life_id`, `user_id`, `remaining_life`, `last_life_given_at`)

- *Yabancı Anahtar:* `user_id`, **users** tablosuna referans verir.

scores (`score_id`, `user_id`, `level_id`, `moves_used`, `score`, `achieved_at`)

- *Yabancı Anahtar:* `user_id`, **users** tablosuna referans verir.
- *Yabancı Anahtar:* `level_id`, **levels** tablosuna referans verir.

orders (`order_id`, `user_id`, `powerup_id`, `quantity`, `ordered_at`)

- *Yabancı Anahtar:* `user_id`, **users** tablosuna referans verir.
- *Yabancı Anahtar:* `powerup_id`, **powerup** tablosuna referans verir.

user_powerups (`id`, `user_id`, `powerup_id`, `quantity`)

- *Yabancı Anahtar:* `user_id`, **users** tablosuna referans verir.
- *Yabancı Anahtar:* `powerup_id`, **powerup** tablosuna referans verir.

5. NORMALİZASYON

5.1 Normalizasyon Süreci

A) BCNF (Boyce–Codd Normal Form) Kontrolü

Yapılan analiz sonucunda:

- **Users, Levels, PowerUp, Orders, Scores** tablolarında belirleyiciler süper anahtar olduğundan BCNF sağlanmaktadır.
- **Lives** tablosunda $\text{user_id} \rightarrow \text{remaining_life}$, $\text{last_life_given_at}$ bağımlılığı bulunduğu ve user_id süper anahtar olmadığı için BCNF ihlali tespit edilmiştir.
- **User_Powerups** tablosunda $(\text{user_id}, \text{powerup_id}) \rightarrow \text{quantity}$ bağımlılığı bulunmasına rağmen yapay bir id alanı kullanıldığı için BCNF ihlali oluşmuştur.

B) BCNF İhlallerinin Giderilmesi

Tespit edilen BCNF ihlallerini gidermek amacıyla aşağıdaki düzenlemeler yapılmıştır:

- **Lives** tablosunda:
 - life_id kaldırılmış,
 - user_id birincil anahtar olarak tanımlanmıştır.
- **User_Powerups** tablosunda:
 - Yapay id alanı kaldırılmış,
 - $(\text{user_id}, \text{powerup_id})$ bileşik anahtar olarak belirlenmiştir.

Bu düzenlemeler sonucunda ilgili tablolar BCNF seviyesine getirilmiştir.

5.2 Son Normalize Edilmiş Şema

Normalizasyon süreci sonunda elde edilen son ilişkisel şema aşağıdaki gibidir:

Users (`user_id`, `user_name`, `email`, `password`, `created_at`, `last_login`)

Levels (`level_id`, `level_number`, `target_score`, `max_moves`, `created_at`)

Powerup (`powerup_id`, `powerup_name`, `description`)

Lives (`user_id`, `remaining_life`, `last_life_given_at`)

Scores (`score_id`, `user_id`, `level_id`, `moves_used`, `score`, `achieved_at`)

Orders (`order_id`, `user_id`, `powerup_id`, `quantity`, `ordered_at`)

User_Powerups (`user_id`, `powerup_id`, `quantity`)

6. SQL SERVER VERİ TABANI ŞEMASI

6.1 Tablo Oluşturma Komutları

```
create database OyunProje;
```

```
go;
```

```
use OyunProje;
```

```
create table users(
    user_id int identity(1,1) primary key,
    user_name varchar(30) not null,
    email varchar(50) not null,
    password varchar(50) not null,
    created_at datetime default getdate(),
    last_login datetime
);
```

```
go;
```

```
use OyunProje;
```

```
create table powerup(
    powerup_id int identity(1,1) primary key,
    powerup_name varchar(50) not null,
    description varchar(250) not null
);
```

```
go;
```

```
use OyunProje;
```

```
create table Levels(
    level_id int identity(1,1) primary key,
    level_number int not null,
```

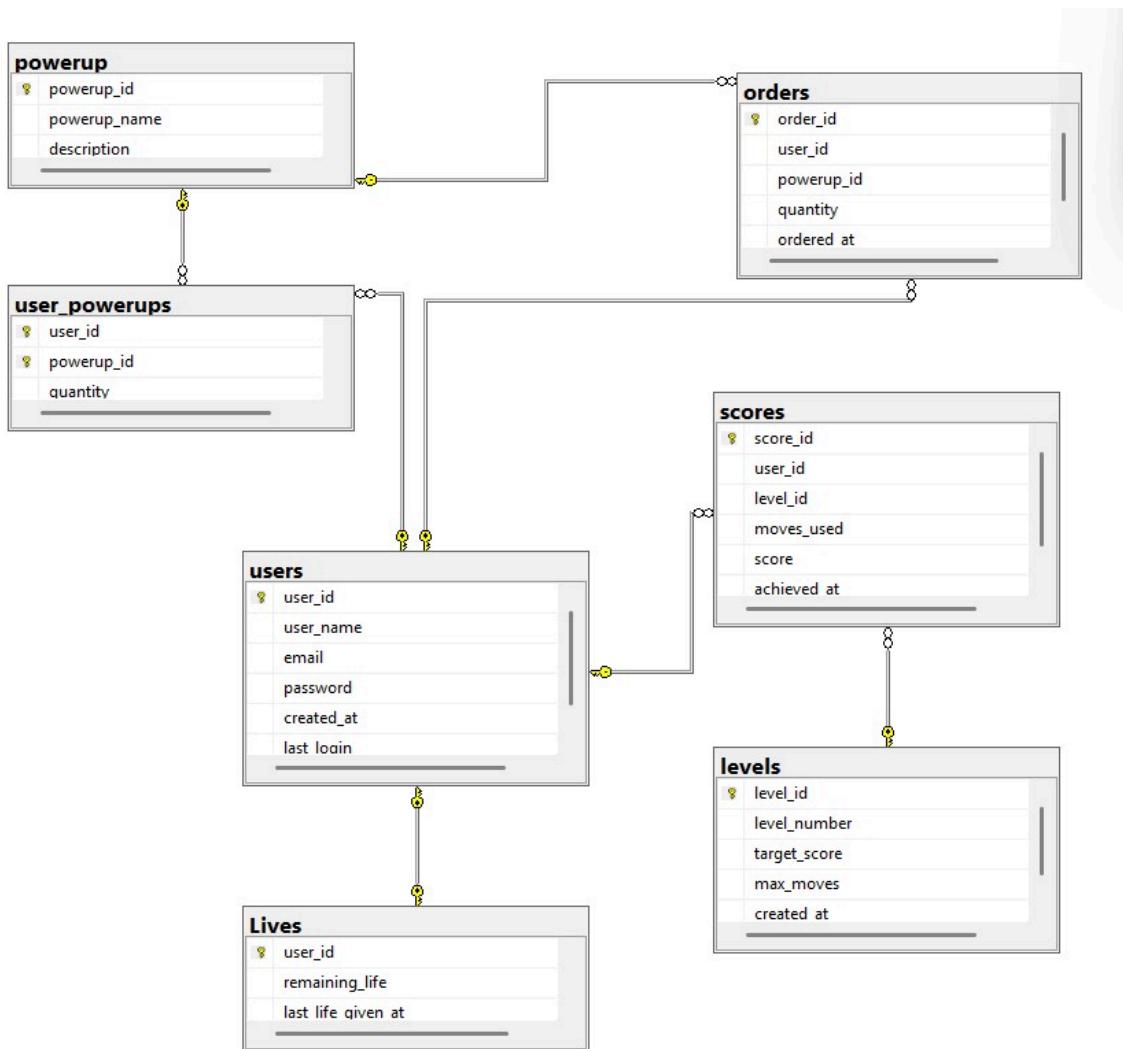
```
target_score int not null,  
max_moves int not null,  
created_at datetime default getdate()  
);  
  
use OyunProje;  
create table Lives(  
user_id int primary key,  
remaining_life int default 5,  
last_life_given_at datetime not null,  
foreign key (user_id) references users(user_id) on delete cascade on update  
cascade  
);  
  
go;  
  
use OyunProje;  
create table scores(  
score_id int identity(1,1) primary key,  
user_id int not null,  
level_id int not null,  
moves_used int not null,  
score int not null,  
achieved_at datetime default getdate(),  
foreign key (user_id) references users(user_id) on delete cascade on update  
cascade,  
foreign key (level_id) references Levels(level_id) on delete cascade on update  
cascade  
);  
  
go;  
  
use OyunProje;
```

```
create table orders(
    order_id int identity(1,1) primary key,
    user_id int not null,
    powerup_id int not null,
    quantity int not null,
    ordered_at datetime default getdate(),
    foreign key (user_id) references users(user_id) on delete cascade on update
    cascade,
    foreign key (powerup_id) references powerup(powerup_id) on delete cascade on
    update cascade
);
```

```
go;
```

```
use OyunProje;
create table user_powerups(
    user_id int not null,
    powerup_id int not null,
    quantity int not null,
    primary key(user_id,powerup_id),
    foreign key (user_id) references users(user_id) on delete cascade on update
    cascade,
    foreign key (powerup_id) references powerup(powerup_id) on delete cascade on
    update cascade
);
go;
```

6.2 Veri Tabanı Diyagramı



7. ÖRNEK VERİLER

7.1 Veri Ekleme Komutları (INSERT)

```
use OyunProje;

insert into user(user_name,email,password)
values('Leyla','leyla@gmail.com','leyla123');

insert into user(user_name,email,password)
values('yağmur','yağmur@gmail.com','yağmur123');

insert into user(user_name,email,password)
values('bilal','bilal@gmail.com','bilal123');

insert into user(user_name,email,password)
values('ahmet','ahmet@gmail.com','ahmet123456');

insert into user(user_name,email,password)
values('berna','berna@gmail.com','berna123456');

insert into user(user_name,email,password)
values('ceren','ceren@gmail.com','ceren123456');

insert into Lives(user_id,last_life_given_at) values (1,getdate());
insert into Lives(user_id,last_life_given_at) values (2,getdate());
insert into Lives(user_id,last_life_given_at) values (3,getdate());
insert into Lives(user_id,last_life_given_at) values (4,getdate());
insert into Lives(user_id,last_life_given_at) values (5,getdate());
insert into Lives(user_id,last_life_given_at) values (6,getdate());

insert into powerup(powerup_name,description) values('marşmelov','birleştirildiği
karenin satır ve sütununu temizler. 4 aynı renk şekerin eşleştirilmesiyle elde edilir');

insert into powerup(powerup_name,description) values('akide şekeri','birleştirildiği
karenin etrafındaki 8 kareyi temizler. L ve T şeklinde aynı renk şekerin
eşleştirilmesiyle elde edilir.');

insert into powerup(powerup_name,description) values('nane şekeri','birleştirildiği
karedeki şekerle aynı renkli tüm şekerleri patlatır. Aynı satır veya sütundaki 5 aynı
renk şekerin birleştirilmesiyle elde edilir.');

insert into powerup(powerup_name,description) values('lolipop','birleştirildiği şekerle
aynı renk tüm şekerleri lolipopun rengine çevirir. 7 aynı renk şekerin eşleştirilmesiyle
elde edilir, elde edildiği şekerin rengini alır.');

insert into powerup(powerup_name,description) values('biskuvi','üzerine çift
tıklandığında bütün alanı yok eder. Sadece ödül olarak ya da satın alımla elde
edilebilir');
```

```
insert into levels(level_number,target_score,max_moves) values(1,1000,50);
insert into levels(level_number,target_score,max_moves) values(2,2000,45);
insert into levels(level_number,target_score,max_moves) values(3,3000,40);
insert into levels(level_number,target_score,max_moves) values(4,4000,30);
insert into levels(level_number,target_score,max_moves) values(5,5000,20);
insert into user_powerups(user_id,powerup_id,quantity) values(1,1,3);
insert into user_powerups(user_id,powerup_id,quantity) values(1,2,0);
insert into user_powerups(user_id,powerup_id,quantity) values(1,3,0);
insert into user_powerups(user_id,powerup_id,quantity) values(1,4,0);
insert into user_powerups(user_id,powerup_id,quantity) values(1,5,0);
insert into user_powerups(user_id,powerup_id,quantity) values(2,1,3);
insert into user_powerups(user_id,powerup_id,quantity) values(2,2,0);
insert into user_powerups(user_id,powerup_id,quantity) values(2,3,0);
insert into user_powerups(user_id,powerup_id,quantity) values(2,4,0);
insert into user_powerups(user_id,powerup_id,quantity) values(2,5,0);
insert into user_powerups(user_id,powerup_id,quantity) values(3,1,3);
insert into user_powerups(user_id,powerup_id,quantity) values(3,2,0);
insert into user_powerups(user_id,powerup_id,quantity) values(3,3,0);
insert into user_powerups(user_id,powerup_id,quantity) values(3,4,0);
insert into user_powerups(user_id,powerup_id,quantity) values(3,5,0);
insert into user_powerups(user_id,powerup_id,quantity) values(4,1,3);
insert into user_powerups(user_id,powerup_id,quantity) values(4,2,0);
insert into user_powerups(user_id,powerup_id,quantity) values(4,3,0);
insert into user_powerups(user_id,powerup_id,quantity) values(4,4,0);
insert into user_powerups(user_id,powerup_id,quantity) values(5,5,0);
insert into user_powerups(user_id,powerup_id,quantity) values(5,1,3);
insert into user_powerups(user_id,powerup_id,quantity) values(5,2,0);
insert into user_powerups(user_id,powerup_id,quantity) values(5,3,0);
insert into user_powerups(user_id,powerup_id,quantity) values(5,4,0);
insert into user_powerups(user_id,powerup_id,quantity) values(5,5,0);
insert into user_powerups(user_id,powerup_id,quantity) values(6,1,3);
insert into user_powerups(user_id,powerup_id,quantity) values(6,2,0);
```

```
insert into user_powerups(user_id,powerup_id,quantity) values(6,3,0);
insert into user_powerups(user_id,powerup_id,quantity) values(6,4,0);
insert into user_powerups(user_id,powerup_id,quantity) values(6,5,0);
insert into orders(user_id,powerup_id,quantity) values(1,1,3);
insert into orders(user_id,powerup_id,quantity) values(2,1,3);
insert into orders(user_id,powerup_id,quantity) values(3,1,3);
insert into orders(user_id,powerup_id,quantity) values(4,1,3);
insert into orders(user_id,powerup_id,quantity) values(5,1,3);
insert into orders(user_id,powerup_id,quantity) values(6,1,3);
insert into scores(user_id,level_id,moves_used,score) values(1,1,20,1500);
insert into scores(user_id,level_id,moves_used,score) values(2,1,21,1400);
insert into scores(user_id,level_id,moves_used,score) values(3,1,22,1300);
insert into scores(user_id,level_id,moves_used,score) values(4,1,23,1200);
insert into scores(user_id,level_id,moves_used,score) values(5,1,24,1100);
insert into scores(user_id,level_id,moves_used,score) values(6,1,25,1600);
insert into scores(user_id,level_id,moves_used,score) values(2,2,21,2400);
insert into scores(user_id,level_id,moves_used,score) values(3,2,22,2300);
insert into scores(user_id,level_id,moves_used,score) values(4,2,23,2200);
insert into scores(user_id,level_id,moves_used,score) values(5,2,24,2100);
insert into scores(user_id,level_id,moves_used,score) values(6,2,25,2600);
insert into scores(user_id,level_id,moves_used,score) values(3,3,22,3300);
insert into scores(user_id,level_id,moves_used,score) values(4,3,23,3200);
insert into scores(user_id,level_id,moves_used,score) values(5,3,24,3100);
insert into scores(user_id,level_id,moves_used,score) values(6,3,25,3600);
insert into scores(user_id,level_id,moves_used,score) values(4,4,23,4200);
insert into scores(user_id,level_id,moves_used,score) values(5,4,24,4100);
insert into scores(user_id,level_id,moves_used,score) values(6,4,25,4600);
insert into scores(user_id,level_id,moves_used,score) values(5,5,24,5100);
insert into scores(user_id,level_id,moves_used,score) values(6,5,25,5600);
```

8. SQL KOMUTLARI VE SCRIPT DOSYALARI

8.1 Gereksinim Bazlı SQL Komutları ve Script Dosyaları

SQL Komutu:

```
create database OyunProje;
```

```
go;
```

```
//Sistemi kullanan oyunculara ait kullanıcı adı, e-posta, şifre ve giriş bilgilerini tutan tablodur.
```

```
use OyunProje;
create table users(
    user_id int identity(1,1) primary key,
    user_name varchar(30) not null,
    email varchar(50) not null,
    password varchar(50) not null,
    created_at datetime default getdate(),
    last_login datetime
```

```
);
```

```
go;
```

```
//Oyunda yer alan güçlendirme öğelerinin ad ve açıklama bilgilerini saklayan tablodur.
```

```
use OyunProje;
create table powerup(
    powerup_id int identity(1,1) primary key,
    powerup_name varchar(50) not null,
    description varchar(250) not null
);
```

```
go;
```

```
//Oyundaki seviyelere ait seviye numarası, hedef skor, maksimum hamle sayısı ve oluşturulma tarihini tutan tablodur.
```

```
use OyunProje;
create table Levels(
    level_id int identity(1,1) primary key,
    level_number int not null,
```

```
target_score int not null,  
max_moves int not null,  
created_at datetime default getdate()  
);  
  
//Her kullanıcının sahip olduğu mevcut can sayısını ve son can kazanım zamanını tutan tablodur.  
use OyunProje;  
create table Lives(  
user_id int primary key,  
remaining_life int default 5,  
last_life_given_at datetime not null,  
foreign key (user_id) references users(user_id) on delete cascade on update cascade  
);  
  
go;  
  
//Kullanıcıların oynadıkları seviyelerde elde ettikleri skorları, kullanılan hamle sayısını ve skor tarihini saklayan tablodur.  
use OyunProje;  
create table scores(  
score_id int identity(1,1) primary key,  
user_id int not null,  
level_id int not null,  
moves_used int not null,  
score int not null,  
achieved_at datetime default getdate(),  
foreign key (user_id) references users(user_id) on delete cascade on update cascade,  
foreign key (level_id) references Levels(level_id) on delete cascade on update cascade  
);  
  
go;  
  
//Kullanıcıların oyun içi güçlendirme satın alma işlemlerini miktar ve tarih bilgileriyle birlikte tutan tablodur.  
use OyunProje;  
create table orders(  
order_id int identity(1,1) primary key,  
user_id int not null,  
powerup_id int not null,
```

```
quantity int not null,  
ordered_at datetime default getdate(),  
foreign key (user_id) references users(user_id) on delete cascade on update cascade,  
foreign key (powerup_id) references powerup(powerup_id) on delete cascade on update cascade  
);  
  
go;  
  
//Kullanıcıların sahip oldukları güçlendirmeleri ve bu güçlendirmelerin miktarlarını gösteren envanter  
tablosudur.  
use OyunProje;  
create table user_powerups(  
user_id int not null,  
powerup_id int not null,  
quantity int not null,  
  
//Kullanıcı ile güçlendirme arasındaki çoktan çoha ilişkisi temsil eder ve bileşik birincil anahtar ile  
tekrar kayıtları engellerken, foreign key ve cascade kuralları sayesinde veri bütünlüğünü sağlar.  
primary key(user_id,powerup_id),  
foreign key (user_id) references users(user_id) on delete cascade on update cascade,  
foreign key (powerup_id) references powerup(powerup_id) on delete cascade on update cascade  
);  
  
go;
```

9. SAKLI YORDAM VE TETİKLEYİCİ

9.1 Saklı Yordam Adı (Stored Procedure)

9.1.1 Amaç ve İş Operasyonu

Bu saklı yordamlar, mobil oyun sisteminde iş operasyonları olan **can yenileme sürecinin otomatik yönetilmesi ve kullanıcıların oyun içi satın alma geçmişinin doğru şekilde hesaplanması** işlemlerini gerçekleştirmek amacıyla geliştirilmiştir. Can yenileme işlemi zaman kısıtına bağlı olarak sistem tarafından kontrol edilirken, satın alma sayımı işlemi kullanıcı ve güçlendirme bazında tutarlı ve güvenilir veri elde edilmesini sağlamaktadır.

9.1.2 Saklı Yordam Kodu

A)Life Refill Prosedür Kodu

Bu saklı yordam, can sayısı 5'in altında olan kullanıcının, son can kazanımından itibaren 15 dakika geçmişse can sayısını 1 artırır ve can kazanım zamanını günceller.

Kod:

```
create procedure life_refill
as
begin
set nocount on;
update Lives
set remaining_life=remaining_life+1,
last_life_given_at=getdate()
where remaining_life<5
and datediff(minute,last_life_given_at,getdate())>=15;
end;

go;
```

B) User Order Count Prosedür Kodu

Bu saklı yordam, belirtilen kullanıcı ve güçlendirme için yapılan tüm satın alma işlemlerinin toplam miktarını hesaplayarak verilen değişkene atar.

Kod:

```
create procedure user_order_count
@user_id int,
@powerup_id int,
@count int
as
begin
select @count=isnull(sum(quantity),0)
from orders
where user_id=@user_id and powerup_id=@powerup_id;
end;

go;
```

9.1.3 Test Senaryoları ve Sonuçları

A-SONUÇ

	user_id	remaining_life	last_life_given_at
1	1	0	2025-12-25 16:56:24.403
2	2	1	2025-12-25 17:08:11.937

SQLQuery1.sq...L\bilal (52)*

```
1 exec life_refill;
```

100 % ① 0 ↑ ↓

İletiler

Komutlar başarıyla tamamlandı.

Tamamlanma süresi:2025-12-25T17:43:47.8978977+03:00

SQLQuery1.sq...L\bilal (52)*

```
1 use OyunProje;
2 select * from Lives
```

100 % ① 0 ↑ ↓

Sonuçlar İletiler

	user_id	remaining_life	last_life_given_at
1	1	1	2025-12-25 17:43:47.897
2	2	2	2025-12-25 17:43:47.897

B-SONUÇ

SQLQuery1.sq...L\bilal (63)*

```
1 select * from orders
```

100 % ✓ Sorun bulunamadı

Sonuçlar İletiler

	order_id	user_id	powerup_id	quantity	ordered_at
1	0	1	1	3	2025-12-26 12:00:09.700
2	1	1	2	3	2025-12-26 12:00:18.000
3	2	1	3	3	2025-12-26 12:00:20.410
4	3	1	4	3	2025-12-26 12:00:23.070
5	4	1	5	3	2025-12-26 12:00:25.160
6	5	2	5	3	2025-12-26 12:00:35.503
7	6	2	4	2	2025-12-26 12:00:41.343
8	7	2	3	5	2025-12-26 12:00:45.713
9	8	2	2	7	2025-12-26 12:00:49.983
10	9	2	1	12	2025-12-26 12:00:53.903
11	10	3	1	15	2025-12-26 12:01:04.463
12	11	3	1	12	2025-12-26 12:01:07.583
13	12	3	2	23	2025-12-26 12:01:14.613
14	13	3	3	1	2025-12-26 12:01:19.310
15	14	3	4	2	2025-12-26 12:01:26.523
16	15	3	5	5	2025-12-26 12:01:33.153

SQLQuery1.sq...L\bilal (63)*

```
1 declare @totalorder int;
2 EXEC user_order_count @user_id =3 ,@powerup_id = 1, @count =@totalorder output;
3 SELECT @totalorder AS 'Kategorideki Ürün Sayısı'
```

100 % ✘ 1 ⚠ 0 ↑ ↓

Sonuçlar İletiler

	Kategorideki Ürün Sayısı
1	27

9.2 Tetikleyici (Trigger)

9.2.1 Amaç ve Tetikleme Koşulu

Tetikleyicilerin amacı, **insert (ekleme)** işlemlerinden sonra ilgili yardımcı kayıtları otomatik olarak oluşturmak ve veri bütünlüğünü sağlamak; yeni kullanıcı veya güçlendirme eklendiğinde sistem gerekli başlangıç verilerini kendiliğinden üretir.

9.2.2 Tetikleyici Kodu

A)User Life Log Trigger

Açıklama: Bu tetikleyici, sisteme yeni bir kullanıcı eklendiğinde kullanıcıya otomatik olarak başlangıç can bilgisi oluşturulmasını sağlar.

Kod:

```
create trigger user_life_log
on users
after insert
as
begin
insert into Lives(user_id,last_life_given_at)
select user_id,getdate()
from inserted
end;

go;
```

B)TRG_AfterPowerupInsert Trigger

Açıklama: Bu tetikleyici, sisteme yeni bir güçlendirme eklendiğinde tüm mevcut kullanıcılar için bu güçlendirmeye ait envanter kaydını sıfır miktar ile otomatik olarak oluşturur.

Kod:

```
create trigger TRG_AfterPowerupInsert
onpowerup
after insert
as
begin

set nocount on;
insert into user_powerups (user_id, powerup_id, quantity)
select
u.user_id,i.powerup_id,0
from users u
```

```
cross join inserted i;
```

```
end;
```

```
go;
```

C)TRG_AfterUserInsert Trigger

Açıklama: Bu tetikleyici, sisteme yeni bir kullanıcı eklendiğinde tüm mevcut güçlendirmeler için kullanıcıya ait envanter kayıtlarını otomatik olarak oluşturur.

Kod:

```
create trigger TRG_AfterUserInsert
onusers
after insert
as
begin
set nocount on;
insert into user_powerups (user_id, powerup_id, quantity)
select
i.user_id,p.powerup_id,0
from inserted i
cross join powerup p;
end;
go;
```

9.2.3 Test Senaryoları ve Sonuçları

A-SONUÇ

SQLQuery1.sq...L\bilal (70)* ✎ X

```
1 | use OyunProje;
2 | select * from users
```

100 % ✓ Sorun bulunamadı

Sonuçlar İletiler

user_id	user_name	email	password	created_at	last_login
---------	-----------	-------	----------	------------	------------

SQLQuery1.sq...L\bilal (70)* ✎ X

```
1 | use OyunProje;
2 | select * from Lives|
```

100 % ✓ Sorun bulunamadı

Sonuçlar İletiler

user_id	remaining_life	last_life_given_at
---------	----------------	--------------------

SQLQuery1.sq...L\bilal (70)* ✎ X

```
1 | use OyunProje;
2 | insert into users(user_name,email,password) values ('leyla','leyla@gmail.com','leyla123')
```

100 % ✓ Sorun bulunamadı

İletiler

(1 satır etkilendi)
(1 satır etkilendi)

Tamamlanma süresi:2025-12-25T16:56:24.4111642+03:00

SQLQuery1.sq...L\bilal (70)* ✎ X

```
1 | \v use OyunProje;
2 | | select * from users
```

100 % ✓ Sorun bulunamadı

■ Sonuçlar ■ İletiler

	user_id	user_name	email	password	created_at	last_login
1	1	leyla	leyla@gmail.com	leyla123	2025-12-25 16:56:24.400	NULL

SQLQuery1.sq...L\bilal (70)* ✎ X

```
1 | \v use OyunProje;
2 | | select * from Lives
```

100 % ✓ Sorun bulunamadı

■ Sonuçlar ■ İletiler

	user_id	remaining_life	last_life_given_at
1	1	5	2025-12-25 16:56:24.403

B-SONUÇ

SQLQuery1.sq...L\bilal (70)* ↗ X

```
1 | \v use OyunProje;
2 | | select * from users
```

100 % ✓ Sorun bulunamadı

Sonuçlar İletiler

	user_id	user_name	email	password	created_at	last_login
1	1	leyla	leyla@gmail.com	leyla123	2025-12-25 16:56:24.400	NULL

SQLQuery1.sq...L\bilal (70)* ↗ X

```
1 | \v use OyunProje;
2 | | select * from user_powerups
```

100 % ✓ Sorun bulunamadı

Sonuçlar İletiler

	user_id	powerup_id	quantity
--	---------	------------	----------

SQLQuery1.sq...L\bilal (70)* ↗ X

```
1 | \v use OyunProje;
2 | | select * from powerup
```

100 % ✓ Sorun bulunamadı

Sonuçlar İletiler

	powerup_id	powerup_name	description
--	------------	--------------	-------------

```
SQLQuery1.sq...L\bilal (70)* ↗ X
1 | \ use OyunProje;
2 | \ insert into powerup(powerup_name,description)
3 | | values('marşmelov','birleştirildiği karenin satır ve sütununu temizler. 4 aynı renk şekerin eşleştirilmesiyle elde edilir.')
100 % ▾ Sorun bulunamadı ⏪ Sat: 3 Krkt:
İletiler
(1 satır etkilendi)
Tamamlanma süresi:2025-12-25T17:03:39.6264285+03:00
```

```
SQLQuery1.sq...L\bilal (70)* ↗ X
1 | \ use OyunProje;
2 | \ select * from users
100 % ▾ ✖ 1 ⚠ 0 ↑ ↓
Sonuçlar İletiler


|   | user_id | user_name | email           | password | created_at              | last_login |
|---|---------|-----------|-----------------|----------|-------------------------|------------|
| 1 | 1       | leyla     | leyla@gmail.com | leyla123 | 2025-12-25 16:56:24.400 | NULL       |


```

```
SQLQuery1.sq...L\bilal (70)* ↗ X
1 | \ use OyunProje;
2 | \ select * from user_powerups
100 % ▾ ✖ 1 ⚠ 0 ↑ ↓
Sonuçlar İletiler


|   | user_id | powerup_id | quantity |
|---|---------|------------|----------|
| 1 | 1       | 1          | 0        |


```

SQLQuery1.sq...L\bilal (70)* ↻ ✎

```

1 | \v use OyunProje;
2 | | select * from powerup

```

100 % ▾ ➊ 0 ↑ ↓

☰ Sonuçlar 📂 İletiler

	powerup_id	powerup_name	description
1	1	marşmelov	birleştirildiği karenin satır ve sütununu temiz...

C-SONUÇ

SQLQuery1.sq...L\bilal (70)* ↻ ✎

```

1 | \v use OyunProje;
2 | | select * from users

```

100 % ▾ ✅ Sorun bulunamadı

☰ Sonuçlar 📂 İletiler

	user_id	user_name	email	password	created_at	last_login
1	1	leyla	leyla@gmail.com	leyla123	2025-12-25 16:56:24.400	NULL

SQLQuery1.sq...L\bilal (70)* ↻ ✎

```

1 | \v use OyunProje;
2 | | select * from user_powerups

```

100 % ▾ ✅ Sorun bulunamadı

☰ Sonuçlar 📂 İletiler

	user_id	powerup_id	quantity
--	---------	------------	----------

SQLQuery1.sq...L\bilal (70)* ✎ X

```
1 | \v use OyunProje;
2 | | select * from powerup
```

100 % ✓ Sorun bulunamadı

Sonuçlar İletiler

powerup_id	powerup_name	description
------------	--------------	-------------

SQLQuery1.sq...L\bilal (70)* ✎ X

```
1 | \v use OyunProje;
2 | | insert into powerup(powerup_name,description)
3 | | values('marşmelov','birleştirildiği karenin satır ve sütununu temizler. 4 aynı renk şekerin eşleştirilmesiyle elde edilir.')
```

100 % ✓ Sorun bulunamadı Sat: 3 Krkt:

İletiler

(1 satır etkilendi)

Tamamlanma süresi: 2025-12-25 17:03:39.6264285+03:00

SQLQuery1.sq...L\bilal (70)* ✎ X

```
1 | \v use OyunProje;
2 | | select * from users
```

100 % ✎ 1 ▲ 0 ↑ ↓

Sonuçlar İletiler

	user_id	user_name	email	password	created_at	last_login
1	1	leyla	leyla@gmail.com	leyla123	2025-12-25 16:56:24.400	NULL

SQLQuery1.sq...L\bilal (70)* ✎ X

```
1 | \v use OyunProje;
2 | | select * from user_powerups
```

100 % ✎ 1 ⚠ 0 ↑ ↓

Sonuçlar İletiler

	user_id	powerup_id	quantity
1	1	1	0

SQLQuery1.sq...L\bilal (70)* ✎ X

```
1 | \v use OyunProje;
2 | | select * from powerup
```

100 % ✎ 1 ⚠ 0 ↑ ↓

Sonuçlar İletiler

	powerup_id	powerup_name	description
1	1	marşmelov	birleştirildiği karenin satır ve sütununu temiz...

10. TRANSACTION YÖNETİMİ

10.1 Transaction Senaryosu Adı: Oyun İçi Güçlendirme Satın Alma İşlemi

Bu transaction senaryosu, bir kullanıcının oyun içi bir güçlendirmeyi satın alması sürecini **bütüncül ve tutarlı** bir şekilde gerçekleştirmek amacıyla tasarlanmıştır. Satın alma işlemi sırasında, güçlendirmeye ait **sipariş kaydının oluşturulması** ve kullanıcının **güçlendirme envanterindeki miktarın güncellenmesi** işlemleri birlikte yürütülmektedir. İşlemlerden herhangi birinin başarısız olması durumunda transaction geri alınarak veri tutarsızlığı oluşması engellenmekte ve sistemin tutarlılığı korunmaktadır.

10.2 Transaction Kodu

```
create procedure order_take
@user_id int,
@powerup_id int,
@quantity int as
begin
begin transaction
begin try
insert into orders(user_id,powerup_id,quantity)
values (@user_id,@powerup_id,@quantity);
update user_powerups
set quantity=quantity+@quantity
where user_id=@user_id and powerup_id=@powerup_id;
if(@@ROWCOUNT=0)
begin
THROW 50001, 'Hata: The specified user or powerup could not be found.', 1;
end
commit transaction;
print 'order completed successfully';
end try
begin catch
rollback transaction;
print 'there is an error:'+Error_message()
```

```
end catch  
end;0https://docs.google.com/document/d/1sv8Mo-skzVQkCjmsArGWtw6x4S-oDQo  
z/edit
```

10.3 Başarılı Senaryo Testi

SQLQuery1.sq...L\bilal (66)*

```
1 | select * from user_powerups
```

100 % ✓ Sorun bulunamadı

Sonuçlar İletiler

	user_id	powerup_id	quantity
1	1	1	3
2	1	2	0
3	1	3	0
4	1	4	0
5	1	5	0
6	2	1	0
7	2	2	0
8	2	3	0
9	2	4	0
10	2	5	0
11	3	1	0
12	3	2	0
13	3	3	0
14	3	4	0
15	3	5	0

SQLQuery1.sq...L\bilal (66)*

```
1 | select * from orders
```

100 % ✓ Sorun bulunamadı

Sonuçlar İletiler

	order_id	user_id	powerup_id	quantity	ordered_at
1	20	1	1	3	2025-12-27 19:47:27.823

SQLQuery1.sq...L\bilal (66)* ✎ X

```
1 | exec order_take @user_id=2,@powerup_id=1,@quantity=5;
```

100 % ✓ Sorun bulunamadı

İletiler

```
(1 satır etkilendi)

(1 satır etkilendi)
order completed successfully

Tamamlanma süresi:2025-12-27T19:54:22.6871127+03:00
```

SQLQuery1.sq...L\bilal (66)* ✎ X

```
1 | select * from user_powerups
```

100 % ✓ Sorun bulunamadı

İletiler

Sonuçlar

	user_id	powerup_id	quantity
1	1	1	3
2	1	2	0
3	1	3	0
4	1	4	0
5	1	5	0
6	2	1	5
7	2	2	0
8	2	3	0
9	2	4	0
10	2	5	0
11	3	1	0
12	3	2	0
13	3	3	0
14	3	4	0
15	3	5	0

SQLQuery1.sq...L\bilal (66)* ✎ X

```
1 | select * from orders
```

100 % ✓ Sorun bulunamadı

Sonuçlar İletiler

	order_id	user_id	powerup_id	quantity	ordered_at
1	20	1	1	3	2025-12-27 19:47:27.823
2	22	2	1	5	2025-12-27 19:54:22.680

10.4 Hata Senaryosu ve ROLLBACK Testi

SQLQuery1.sq...L\bilal (80)* ✎ X

```
1 | use OyunProje;
2 | select * from user_powerups|
```

100 % ✓ Sorun bulunamadı

Sonuçlar İletiler

	user_id	powerup_id	quantity
1	1	1	3
2	1	2	0
3	1	3	0
4	1	4	0
5	1	5	0
6	2	1	0
7	2	2	0
8	2	3	0
9	2	4	0
10	2	5	0
11	3	1	0
12	3	2	0
13	3	3	0
14	3	4	0
15	3	5	0

SQLQuery1.sql...L\bilal (80)* ✎ X

```
1 | \v use OyunProje;
2 | | select * from orders|
```

100 % ✅ Sorun bulunamadı

Sonuçlar İletiler

	order_id	user_id	powerup_id	quantity	ordered_at
1	16	1	1	10	2025-12-26 14:07:14.990

SQLQuery1.sql...L\bilal (80)* ✎ X

```
1 | \v use OyunProje;
2 | | exec order_take @user_id=10,@powerup_id=3,@quantity=4;
```

100 % ✕ 1 ⚠ 0 ↑ ↓

İletiler

```
(1 satır etkilendi)

(0 satır etkilendi)
there is an error:Hata: The specified user or powerup could not be found.

Tamamlanma süresi:2025-12-26T14:31:05.0916288+03:00
```

SQLQuery1.sq...L\bilal (80)* ✎ X

```
1 | 1 use OyunProje;
2 | 2 select * from user_powerups
```

100 % ▾ 1 0 ↑ ↓

■ Sonuçlar ॥ İletiler

	user_id	powerup_id	quantity
1	1	1	3
2	1	2	0
3	1	3	0
4	1	4	0
5	1	5	0
6	2	1	0
7	2	2	0
8	2	3	0
9	2	4	0
10	2	5	0
11	3	1	0
12	3	2	0
13	3	3	0
14	3	4	0
15	3	5	0

SQLQuery1.sq...L\bilal (80)* ✎ X

```
1 | 1 use OyunProje;
2 | 2 select * from orders
```

100 % ▾ ✓ Sorun bulunamadı

■ Sonuçlar ॥ İletiler

	order_id	user_id	powerup_id	quantity	ordered_at
1	16	1	1	10	2025-12-26 14:07:14.990

11. TAKIM ÇALIŞMASI VE GÖREV DAĞILIMI

11.1 Takım Üyeleri ve Rollerİ

Takım Üyesi	Öğrenci No	Temel Rol
Bilal Tekin	230260081	Proje Lideri / Veri Tabanı Geliştiricisi
Yağmur Gür	220260089	Veri Tabanı Tasarımcısı
Leyla Erdoğan	230260007	Dokümantasyon / Kalite Kontrol

11.2 Gerçekleştirilen İşler ve Sorumluluk Matrisi

No	Yapılan İş / Özellik	Talimat Veren	Geçekleştiren	Kontrol Eden	Durum
1	Proje gereksinimlerinin belirlenmesi	Yağmur Gür	Leyla Erdoğan	Yağmur Gür	✓
2	E-R diyagramının oluşturulması	Leyla Erdoğan	Yağmur Gür	Leyla Erdoğan	✓
3	İlişkisel şemalara dönüştürme	Yağmur Gür	Leyla Erdoğan	Yağmur Gür	✓
4	Normalizasyon (3NF/BCNF)	Bilal Tekin	Yağmur Gür	Bilal Tekin	✓
5	SQL Server'da şema oluşturma	Leyla Erdoğan	Bilal Tekin	Yağmur Gür	✓
6	Birincil/Yabancı anahtar tanımları	Bilal Tekin	Yağmur Gür	Bilal Tekin	✓
7	Nitelik kısıtlamalarının eklenmesi	Leyla Erdoğan	Yağmur Gür	Bilal Tekin	✓
8	Örnek verilerin eklenmesi	Bilal Tekin	Yağmur Gür	Bilal Tekin	✓
9	SQL sorgu script dosyalarının hazırlanması	Yağmur Gür	Bilal Tekin	Yağmur Gür	✓
10	Saklı yordamın (Stored Procedure) yazılması	Leyla Erdoğan	Bilal Tekin	Leyla Erdoğan	✓
11	Tetikleyicinin (Trigger) yazılması	Yağmur Gür	Bilal Tekin	Yağmur Gür	✓
12	Transaction yönetimi kodunun yazılması	Yağmur Gür	Bilal Tekin	Yağmur Gür	✓

13	COMMIT/ROLLBACK test senaryoları	Yağmur Gür	Bilal Tekin	Leyla Erdoğan	✓
14	Rapor hazırlama ve dokümantasyon	Bilal Tekin & Yağmur Gür	Leyla Erdoğan	Bilal Tekin & Yağmur Gür	✓
15	Son kontrol ve teslim	Bilal Tekin & Yağmur Gür	Leyla Erdoğan	Bilal Tekin & Yağmur Gür	✓

12. SONUÇ VE DEĞERLENDİRME

Bu proje kapsamında, seviye tabanlı bir mobil oyunun ihtiyaçlarını karşılayabilecek kapsamlı ve tutarlı bir **veri tabanı yönetim sistemi** tasarlanmıştır. Proje sürecinde, gerçek hayatta sıkça karşılaşılan kullanıcı yönetimi, skor takibi, oyun içi kaynak yönetimi ve satın alma süreçleri ele alınarak bu gereksinimlere uygun bir ilişkisel veri tabanı modeli oluşturulmuştur.

En önemli kazanımlardan biri; teorik olarak öğrenilen **veri tabanı tasarıımı, normalizasyon, primary key–foreign key ilişkileri, tetikleyiciler, saklı yordamlar ve transaction kavramlarının** gerçekçi bir senaryo üzerinden uygulanması olmuştur. Özellikle veri bütünlüğünün sağlanması ve iş kurallarının veritabanı seviyesinde kontrol edilmesi, sistemin güvenilirliği açısından önemli bir deneyim sağlamıştır.

Süreçte karşılaşılan bazı zorluklar; çoktan çoğu ilişkilerin modellenmesi ve can ile güçlendirme gibi dinamik verilerin yönetilmesi olmuştur. Bu zorluklar, normalizasyon kurallarına uygun tasarım, ara tabloların kullanılması ve tetikleyici ile saklı yordamların sisteme entegre edilmesi sayesinde çözülmüştür. Sonuç olarak ise geliştirilen veri tabanı sistemi; tutarlı, genişletilebilir ve yönetilebilir bir yapıdadır.