

ADIYAMAN ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ / BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

VERİ TABANI YÖNETİM SİSTEMLERİ DERSİ DÖNEM PROJESİ RAPORU

Öğrenci Bilgileri:

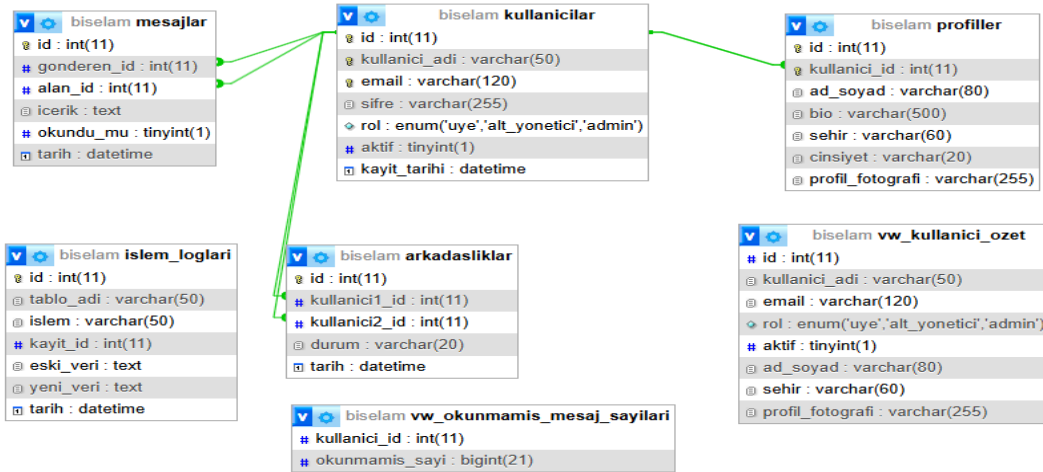
- **Ad Soyad:** Enes Emiral
- **Öğrenci No:** 242104045
- **Proje Konusu:** Bi'Selam - Sosyal Arkadaşlık ve Mesajlaşma Platformu

1. PROJE TANIMI VE KAPSAMI

Bu proje, kullanıcıların profil oluşturabildiği, ilgi alanlarını tanımlayabildiği, birbirlerine arkadaşlık isteği gönderip onaylayabildiği ve anlık mesajlaşabildiği bir sosyal platform veri tabanı mimarisidir. Projenin temel amacı; ilişkisel modelleme, 3. Normal Form (3NF) kuralları ve ileri SQL tekniklerinin (Stored Procedure, View, Transaction ve Trigger) uygulanmasıdır.

2. VERİ TABANI TASARIMI VE İLİŞKİSEL YAPI

Projede veri bütünlüğünü sağlamak adına toplam **7 adet tablo** tasarlanmıştır.



- **kullanicilar:** Temel hesap bilgilerini, şifreleri ve kullanıcı yetki rollerini (uye, admin) barındırır¹.
- **profiller:** Kullanıcı detaylarını saklar ve kullanicilar tablosuna kullanic_iadi üzerinden **1:1** ilişki ile bağlıdır².

- **ilgi_alanlari & kullanıcı_ilgileri:** ilgi alanlarını ayrı bir tabloda tutarak veri tekrarını önleyen **M:N** (Çoka Çok) ilişkisidir³.
- **mesajlar:** Kullanıcılar arası iletişimi ve okunma durumlarını saklar⁴.
- **arkadasliklar:** Kullanıcılar arasındaki sosyal bağlantıları ve istek durumlarını yönetir⁵.
- **islem_loglari:** Sistemdeki tüm INSERT, UPDATE ve DELETE hareketlerini tetikleyiciler aracılığıyla kayıt altında tutan loglama tablosudur⁶.

3. NORMALİZASYON VE KISITLAMALAR (CONSTRAINTS)

Veri tabanı **3. Normal Form (3NF)** kurallarına göre düzenlenmiştir. Örneğin; ilgi alanları doğrudan kullanıcı tablosuna yazılmak yerine, veri tutarlılığını sağlamak için bağımsız bir tabloya (**ilgi_alanlari**) taşınmıştır⁷.

Uygulanan Kısıtlamalar:

- **Primary & Foreign Keys:** Tüm tablolarda ilişkisel bütünlük için anahtar yapısı kurulmuştur⁸.
- **Unique:** Kullanıcı adı ve e-posta alanları tekil (unique) olarak tanımlanmıştır⁹.
- **Check Constraints:** **chk_kendine_istek** (kullanıcının kendisine istek atmasını engeller) ve **chk_icerik_bos_degil** kısıtlamaları eklenmiştir¹⁰.

4. İLERİ SQL UYGULAMALARI

4.1. Stored Procedures (Saklı Yordamlar)

Ödev gereksinimi olan 3 adet prosedür başarıyla tanımlanmıştır¹¹:

1. **sp_mesaj_gonder:** Kullanıcılar arası mesajlaşmayı yönetir. İçerisinde bulunan **SIGNAL SQLSTATE** komutlarıyla, kullanıcının kendine mesaj atması veya boş mesaj gönderilmesi veritabanı seviyesinde engellenerek veri bütünlüğü korunur.
2. **sp_arkadaslik_onayla:** Arkadaşlık tablosundaki durum alanını onaylandı olarak günceller. Sadece isteği alan kullanıcının (**kullanici2_id**) bu işlemi yapabilmesi için mantıksal kontrol içerir.
3. **sp_kullanici_ilgi_ekle:** **ON DUPLICATE KEY UPDATE** yapısını kullanarak, ilgi alanı tabloda yoksa oluşturur, varsa mevcut ID'yi alarak **kullanici_ilgileri** tablosuna kayıt yapar. Bu sayede "Çoka Çok" ilişki yönetilir.

```

1  -- 1. Mesaj Gönderim Kontrolü ve İşlemi
2  CREATE PROCEDURE sp_mesaj_gonder(IN p_gonderen INT, IN p_alan INT, IN p_icerik VARCHAR(1000))
3  BEGIN
4      IF p_gonderen = p_alan THEN
5          SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT='Kendine mesaj gönderemezsin';
6      END IF;
7      INSERT INTO mesajlar(gonderen_id, alan_id, icerik) VALUES(p_gonderen, p_alan, p_icerik);
8  END;
9
10 -- 2. Arkadaşlık İsteği Onaylama
11 CREATE PROCEDURE sp_arkadaslik_onayla(IN p_arkadaslik_id INT, IN p_onaylayan_k_id INT)
12 BEGIN
13     UPDATE arkadasliklar SET durum='onaylandi'
14     WHERE id=p_arkadaslik_id AND kullanıcı2_id=p_onaylayan_k_id;
15 END;

```

4.2. Views (Görünümler)
















Veriye erişimi kolaylaştırmak ve karmaşık JOIN işlemlerini basitleştirmek için 3 adet View oluşturulmuştur¹⁵:

- **vw_kullanici_ozet:** JOIN işlemi ile kullanıcılar ve profiller tablolarını birleştirir. Uygulama tarafında her seferinde uzun JOIN sorguları yazmak yerine bu görünüm üzerinden hızlıca tüm üye detaylarına erişilir. ¹⁶
- **vw_okunmamis_mesaj_sayilari:** GROUP BY ve COUNT fonksiyonlarını kullanarak her kullanıcı için bekleyen mesaj sayısını anlık olarak hesaplar. ¹⁷
- **vw_onayli_arkadasliklar:** Sadece onaylandi durumundaki kayıtları filtreleyerek sistemdeki aktif sosyal ağ bağlantılarını sunar. ¹⁸

```

1  -- Detaylı Kullanıcı Listesi Görünümü
2  CREATE VIEW vw_kullanici_ozet AS
3  SELECT k.id, k.kullanici_adi, k.email, p.ad_soyad, p.sehir
4  FROM kullanicilar k
5  LEFT JOIN profiller p ON p.kullanici_id = k.id;
6
7  -- Okunmamış Mesajların Raporlanması
8  CREATE VIEW vw_okunmamis_mesaj_sayilari AS
9  SELECT alan_id AS kullanıcı_id, COUNT(*) AS okunmamis_sayi
10 FROM mesajlar WHERE okundu_mu = 0 GROUP BY alan_id;

```

				id	kullanici_adi	email	rol	aktif	ad_soyad	sehir	profil_fotografi
<input type="checkbox"/>				1	admin	admin@biselam.com	admin	1	Sistem Yöneticisi	İstanbul	default.jpg
<input type="checkbox"/>				2	rakanazor	rakanazor@biselam.com	uye	1	rakanazor		profil_2_6941f502ae7e1.jpg
<input type="checkbox"/>				3	BiSelam	biselam@biselam.com	admin	1	BiSelam	Adana	unknown.jpeg
<input type="checkbox"/>				4	Muhammet	muhasav@gmail.com	admin	1	Muhammet	Köymaniye	profil_4_69485a0aa926a.jpg
<input type="checkbox"/>				5	Emirhan	emirhansav@gmail.com	admin	1	Emirhan	NULL	unknown.jpeg

4.3. Transaction Yönetimi

Bütünlük: Arkadaşlık isteği oluşturulurken aynı anda ilgi alanı eklenmesi gibi işlemlerde kullanılır. ¹⁹

Hata Yönetimi: Eğer işlem sırasında bir kısıtlama (Örn: UNIQUE e-posta hatası) ihlal edilirse, ROLLBACK komutu ile tüm değişiklikler geri alınarak veritabanının tutarsız bir duruma düşmesi engellenir. ²⁰

```
1  -- Arkadaşlık İsteği ve İlgi Alanı Ekleme İşlemi
2  START TRANSACTION;
3  INSERT INTO arkadasliklar(kullanici1_id, kullanici2_id, durum) VALUES(1, 3, 'bekliyor');
4  CALL sp_kullanici_ilgi_ekle(1, 'Sinema');
5  COMMIT;
6
7  -- Hata Durumunda Geri Alma Örneği
8  START TRANSACTION;
9  UPDATE kullanicilar SET email='hata@test.com' WHERE id=1;
10 ROLLBACK;
```

4.4. Trigger (Tetikleyiciler) ve Loglama

trg_mesajlar_ai/au/ad: mesajlar tablosunda yapılan her INSERT, UPDATE ve DELETE işlemi sonrasında (AFTER) otomatik çalışır. ²²

Loglama: İşlem yapılan kaydın eski ve yeni hali JSON_OBJECT fonksiyonu ile JSON formatına dönüştürülerek islem_loglari tablosuna kaydedilir. Bu, ileri düzey bir loglama yöntemidir ve verinin geçmişe dönük takibini sağlar.

```
1  -- Yeni Mesaj Loglama Tetikleyicisi
2  CREATE TRIGGER trg_mesajlar_ai AFTER INSERT ON mesajlar
3  FOR EACH ROW
4  BEGIN
5  INSERT INTO islem_loglari(tablo_adi, islem, kayit_id, yeni_veri)
6  VALUES('mesajlar', 'INSERT', NEW.id, JSON_OBJECT('icerik', NEW.icerik));
7  END;
```

<div>←T→</div>										
			id	tablo_adi	islem	kayit_id	eski_veri	yeni_veri	tarikh	
<input type="checkbox"/>				1	mesajlar	MESAJ_SILINDI	2	{"id":2,"gonderen_id":5,"alan_id":4,"icerik":"sa",...	NULL	2025-12-21 23:34:27
<input type="checkbox"/>				2	mesajlar	MESAJ_SILINDI	4	{"id":4,"gonderen_id":4,"alan_id":5,"icerik":"sa",...	NULL	2025-12-21 23:37:37

4.5. KRİTİK ANALİZ SORGULARI

Veri tabanı üzerinde ileri seviye SQL teknikleri (JOIN, GROUP BY, HAVING, SUBQUERY) kullanılarak üretilen analiz sorguları aşağıdadır:

- **En Çok Mesaj Atan İlk 5 Kullanıcı:** JOIN ve GROUP BY kullanılarak platformun en aktif kullanıcıları tespit edilmektedir.

- **Şehir Bazlı Kullanıcı Dağılımı:** HAVING kısıtlaması ile sadece sistemde en az 2 kullanıcısı bulunan şehirlerin istatistiği tutulmaktadır.
- **İlgi Alanı Analizi:** JOIN ve COUNT ile hangi ilgi alanının daha popüler olduğu analiz edilmektedir.

```
1  -- 1) En çok mesaj atan ilk 5 kullanıcı
2  SELECT k.kullanici_adi, COUNT(*) AS mesaj_sayisi
3  FROM mesajlar m
4  JOIN kullanicilar k ON k.id = m.gonderen_id
5  GROUP BY k.id
6  ORDER BY mesaj_sayisi DESC
7  LIMIT 5;
8
9  -- 2) Şehirlere göre kullanıcı sayısı (en az 2 kullanıcı olan şehirler)
10 SELECT p.sehir, COUNT(*) AS kisi
11 FROM profiller p
12 WHERE p.sehir IS NOT NULL AND p.sehir <> ''
13 GROUP BY p.sehir
14 HAVING COUNT(*) >= 2
15 ORDER BY kisi DESC;
16
17 -- 3) İlgi alanına göre kullanıcı sayısı (İleri JOIN örneği)
18 SELECT i.ad, COUNT(*) AS kisi
19 FROM ilgi_alanlari i
20 JOIN kullanıcı_ilgileri ki ON ki.ilgi_id = i.id
21 GROUP BY i.id
22 ORDER BY kisi DESC;
```

5. ARAYÜZ VE SUNUM

Proje için PHP programlama dili kullanılarak modern ve işlevsel bir kullanıcı arayüzü tasarlanmıştır. Veri tabanı CRUD (Ekleme, Okuma, Güncelleme, Silme) işlemleri bu arayüz üzerinden test edilmiştir. Proje yönetimi ve sürüm kontrolü için GitHub kullanılmıştır.

Canlı Demo Adresi: biselam.enesemiral.com

GitHub Repository: <https://github.com/emiralcode/Bi-Selam.git>



emiralcodes Update README.md

42ed2a0 · 1 hour ago

5 Commits

admin	Add initial project files for social platform	13 hours ago
uploads	Add initial project files for social platform	13 hours ago
.gitignore	Add initial project files for social platform	13 hours ago
README.md	Update README.md	1 hour ago
anasayfa.php	Add initial project files for social platform	13 hours ago
arkadas_islem.php	Add initial project files for social platform	13 hours ago
arkadaslarim.php	Add initial project files for social platform	13 hours ago
biselim.sql	Add initial project files for social platform	13 hours ago
cikis_yap.php	Add initial project files for social platform	13 hours ago
db_sample.php	Add initial project files for social platform	13 hours ago
footer.php	Add initial project files for social platform	13 hours ago
giris.php	Add initial project files for social platform	13 hours ago
giris_yap.php	Add initial project files for social platform	13 hours ago
header.php	Add initial project files for social platform	13 hours ago
kayit.html	Add initial project files for social platform	13 hours ago
kayit_ol.php	Add initial project files for social platform	13 hours ago
konusma.php	Add initial project files for social platform	13 hours ago
mesaj_gonder_ajax.php	Add initial project files for social platform	13 hours ago
mesajlari_getir_ajax.php	Add initial project files for social platform	13 hours ago
mesajlarim.php	Add initial project files for social platform	13 hours ago
profil_duzenle.php	Add initial project files for social platform	13 hours ago
profil_guncelle.php	Add initial project files for social platform	13 hours ago
uyeler.php	Add initial project files for social platform	13 hours ago