

FLUTTER

SQLite CRUD

24. hafta • Yerel Veri Saklama

36 hafta, 72 bağımsız çalışan Flutter projesi, gerçek uygulama ekranları, uygulamalı dersler ve öğrenci geri bildiriminden sonra incelenebilen çözüm kaynakları.

Bu kitap güncel ders kaynaklarından otomatik üretilmiştir; özet ya da jenerik şablon değildir.

Öğrenme Haritası

Hafta	Ders	Projeler
24	SQLite CRUD	Telefon Rehberi • Cep Harçlığı Defteri

24. Hafta — SQLite CRUD

Bu hafta ne yapacağız?

Önceki hafta tablo oluşturup kayıt ekledik ve listeledik. Bu hafta bir verinin bütün yaşam döngüsünü tamamlayacağız: oluşturma (CREATE), okuma (READ), güncelleme (UPDATE) ve silme (DELETE). Telefon Rehberi ve Cep Harçlığı Defteri projelerinde bütün işlemler gerçek SQLite satırları üzerinde uygulanacaktır.

Kazanımlar

- CRUD işlemlerini kullanıcı arayüzüyle ilişkilendirme
- Satırı birincil anahtar üzerinden güvenli biçimde güncelleme ve silme
- Düzenleme formuna mevcut değerleri aktarma
- Her yazma işleminden sonra listeyi veritabanından yeniden sorgulama
- REAL, TEXT, kısıt ve otomatik kimlik alanlarını kullanma
- CRUD akışının tamamını bellek içi SQLite testiyle doğrulama

Çözümlü uygulama — SQLite Telefon Rehberi

Tablo

```
CREATE TABLE kisiler(  
  id INTEGER PRIMARY KEY AUTOINCREMENT,  
  ad TEXT NOT NULL,  
  telefon TEXT NOT NULL UNIQUE  
)
```

Telefonun UNIQUE olması aynı numaranın yanlışlıkla birden çok kişiye eklenmesini engeller.

CREATE — Yeni kişi ekleme

```
Future<int> ekle(Kisi kisi) async {  
  return (await db).insert('kisiler', kisi.toMap());  
}
```

READ — Kişileri okuma

```
Future<List<Kisi>> listele() async {  
  final maps = await (await db).query(  
    'kisiler',  
    orderBy: 'ad',  
  );  
  return maps.map(Kisi.fromMap).toList();  
}
```

UPDATE — Telefonu güncelleme

```
Future<int> guncelle(Kisi kisi) async {  
  return (await db).update(  
    'kisiler',  
    kisi.toMap(),  
    where: 'id = ?',  
    whereArgs: [kisi.id],  
  );  
}
```

where yazılmazsa tablodaki bütün satırlar güncellenebilir. Bu nedenle düzenlenecek satırın id değeri her zaman koşula gönderilir.

DELETE — Kişiyi silme

```
Future<int> sil(int id) async {  
  return (await db).delete(  
    'kisiler',  
    where: 'id = ?',  
    whereArgs: [id],  
  );  
}
```

Düzenleme penceresi

TextField.initialValue, mevcut telefon numarasını kullanıcıya gösterir. Güncelle düğmesi yeni değeri ana sayfaya döndürür; SQL işlemi tamamlandıktan sonra liste tekrar sorgulanır.

Gerçek uygulama ekranı

Test Ece Yılmaz'ı eklemiş, telefonunu UPDATE ile değiştirmiş, ikinci kişiyi ekleyip DELETE etmiş ve veritabanında yalnız güncellenmiş Ece kaydının kaldığını doğrulamıştır.

SQLite Telefon Rehberi

Ekle

1 kişi

CRUD etkin

E

Ece Yılmaz
0555 999 88 77



SQLite Telefon Rehberi gerçek Flutter ekranı

Özgün uygulama — SQLite Cep Harçlığım

Hareket tablosu

```
CREATE TABLE hareketler(  
  id INTEGER PRIMARY KEY AUTOINCREMENT,  
  aciklama TEXT NOT NULL,  
  tutar REAL NOT NULL CHECK(tutar != 0)  
)
```

Gelir pozitif, gider negatif tutarla saklanır. Böylece bakiye doğrudan toplamla bulunabilir:

```
final bakiye = hareketler.fold<double>(  
  0,  
  (toplam, hareket) => toplam + hareket.tutar,  
)
```

Tutarı güncelleme

```
final yeniTutar = hareket.tutar > 0
    ? hareket.tutar + 25
    : hareket.tutar - 25;

await depo.guncelle(Hareket(
  id: hareket.id,
  aciklama: hareket.aciklama,
  tutar: yeniTutar,
));
```

Son kayıt önce gelecek şekilde okuma

```
database.query(
  'hareketler',
  orderBy: 'id DESC',
)
```

Test 100 TL gider eklemiş, güncellemeyle 125 TL gider yapmış, 300 TL gelir eklemiş ve gideri silmiştir. Son bakiye ile SQL sonucu 300 TL olarak doğrulanmıştır.

SQLite Cep Harçlıđım

Güncel bakiye
₺300.00

Açıklama

Tutar

Gelir

Ekle

+

Haftalık harçlık
Gelir

++₺300.00



SQLite Cep Harçlıđı gerçek Flutter ekranı

CRUD işlem sırası

1. Kullanıcı formu doldurur veya bir satır seçer.
2. Girdi arayüzde doğrulanır.
3. Depolama sınıfı parametrelili SQL işlemini yürütür.
4. Etkilenen satır sayısı veya kimlik alınır.
5. Liste SQLite'tan yeniden sorgulanır.
6. setState ile yalnız güncel sonuç ekrana verilir.

Uygulama görevleri

Kolay — Yapılacaklar veritabanı

Görev ekleme, tamamlandı bilgisini güncelleme, listeleme ve silme işlemlerini uygulayın.

Orta — Kütüphane kitapları

Kitap adı, yazar ve ödünç alan alanlarını saklayın. Ödünç verme/iade işlemini UPDATE, kitap kaldırmayı DELETE ile yapın.

İleri — Kantin stok yönetimi

Ürün, fiyat ve stok alanları oluşturun. Satışta stok azaltın, fiyat düzenleyin, ürünü silin ve toplam stok değerini SQL sonuçlarından hesaplayın. Negatif stoğu veritabanı kısıtıyla engelleyin.

Güvenlik ve doğruluk kontrolü

- Güncelleme ve silme sorgusunda WHERE id = ? bulunmalı.
- Kullanıcı verisi SQL metnine birleştirilmemeli; whereArgs kullanılmalı.
- Yazma tamamlanmadan başarı mesajı gösterilmemeli.
- Silme gibi geri alınması zor işlemlerde üretim sürümüne onay penceresi eklenmeli.
- UNIQUE ve CHECK hataları kullanıcıya anlaşılır biçimde çevrilmeli.
- Bütün CRUD işlemleri ayrı testlerle veya tek uçtan uca akışla doğrulanmalı.

Geri bildirim ve çözüm erişimi

Öğrenci CRUD uygulamasını gönderdikten sonra İpucunu göster doğru WHERE koşulları ve veri akışı hakkında yön verir. Örnek çözümü göster gerçek model, depo ve arayüz kodunu; indirme bağlantısı ise tüm SQLite bağımlılıklarıyla çalışan projeyi sunar.

Kaynak projeler

- Konu uygulaması: 06 - Yerel Veri Saklama/Uygulamalar/Uygulama-24-Telefon Rehberi
- Özgün uygulama: 06 - Yerel Veri Saklama/Uygulamalar/Ozgun-24-Cep Harçl■■■■ Defteri