

PROGRAMLAMA LABORATUVARI 2

3. PROJE

Bilgisayar Mühendisliği Bölümü Kocaeli Üniversitesi

Ali Murat Ekmekçi *
200202114

Melih Turaneri †
200202075

MAYIS 2022

1 Özet

Bu doküman Programlama Laboratuvarı 2 dersi 3. Projesi için çözümümüzü açıklamaya yönelik oluşturulmuştur. Dökümanda projenin tanımı, çözüme yönelik yapılan araştırmalar, kullanılan yöntemler, proje hazırlanırken kullanılan geliştirme ortamı ve kod bilgisi gibi programın oluşumunu açıklayan başlıklara yer verilmiştir. Doküman sonunda projemizi hazırlarken kullandığımız kaynaklar bulunmaktadır.

2 Proje Tanımı

2.1 Proje Tanımı

Veri tabanı verilerin organize bir şekilde tutulup kolay bir şekilde ulaşılmasını, yönetilmesini ve güncellenmesini sağlayan diğer bir değış ile sanal ortamda verilerinizin toplandığı birimdir. Veri tabanlarında genellikle

veri sorgulama ve yazma işlemi için (Yapılandırılmış Sorgu Dili) kullanılır.

Veri işlemenin özellikle büyük şirketler için bu kadar önemli olduğu dönemde, veri tabanları sistemin kalbini oluşturmaktadır. Veri tabanları genellikle DBMS (Veri Tabanı Yönetim Sistemi) ile veritabanını tanımlamak, yaratmak, kullanmak, değıştirmek ve veritabanı sistemleri ile ilgili gereksinimleri karşılamak amacıyla birlikte kullanılır.

Veritabanı türlerinden olan ilişkisel veri tabanı, verilerin birbiriyle ilişkili noktalarını baz alan ve bunlara erişim sağlayarak işlem yapan bir veritabanı türüdür. İlişkisel veritabanı teknolojisi, yapılandırılmış bilgilere en verimli ve esnek şekilde erişme olanağını sağlar.

Projede bizden istenen, tasarladığımız veritabanı bankanın müşterilere, çalışanlara, hesaplara ve işlemlere ilişkin bilgileri organize bir şekilde işlemesine yardımcı olacaktır. Bu şekilde bankanın ve müşterilerin ihtiyacı olan bilgilere daha kolay ulaşabil-

*alimuratekmekci@gmail.com

†turaneri45@gmail.com

mesi sağlanmış olacaktır. Ayrıca, veritabanı kullanarak müşteri istekleri ve bankanın ihtiyaçları doğrultusunda raporlar hazırlanabilecektir. Bizden problemlerin çözümü için bir veritabanı tasarımı yapmamız istenmektedir. Projede oluşturulacak veritabanının ER diyagramının oluşturulması ve arayüzde gösterilmesi gerekmektedir. Diyagram üzerinden gerçekleştirilen normalizasyon işlemlerinin adım adım gösterilmelidir.

Projede veritabanı oluşturmak için MySQL, istekleri sağlamak için ise JAVA programlama dilleri kullanılmıştır.

2.2 İsterler

Banka içerisinde müşteri, temsilci ve banka müdürü olmak üzere 3 adet rol bulunmaktadır. Müşteriler ve çalışanlar için gerekli tanımlayıcı bilgiler (Ad Soyad, Telefon, TC No, Adres, E-posta) veri tabanında saklanmalıdır. Bir müşterinin birden fazla hesabı bulunabilir. Hesaplar sistem içerisinde kayıtlı bulunan herhangi bir para birimi cinsinden açılabilir (TL varsayılan olarak gelmelidir). Hesaplar arası para transferinde gerekli durumlarda kur dönüşümü otomatik olarak yapılmalıdır. Rollerin gerçekleştirdiği eylemler aşağıda belirtilmiştir. Tüm bu eylemlerin tasarlanan bir arayüz üzerinden görsel bir şekilde gösterilmesi gerekmektedir.

Müşteriler;

- Hesaplarından para çekebilirler ve yatırabilirler.
- Yeni hesap açma ve var olan bir hesabı silme talebinde bulunabilirler.
- Birbirleri arasında para transferi yapabilirler

- Bilgilerini güncelleyebilirler. (Adres, Telefon vs.)
- Bankaya para transferi yapabilirler. (Kredi borcu ödeme)
- Bankadan kredi talep edebilirler. Kredi sadece TL cinsinden talep edilebilmektedir.

Müşteri Temsilcileri;

- Her müşterinin bir temsilcisi vardır.
- Müşteri ekleme, silme ve düzenleme yapılabilir (silme ve düzenleme işlemleri sadece kendi müşterileri için geçerlidir).
- Müşteri bilgilerini güncelleyebilirler. (Adres, Telefon vs.)
- İlgilendikleri müşterilerin genel durumlarını (gelir, gider ve toplam bakiye) görüntüleyebilmektedir.
- Müşterilerden gelen hesap açma, silme ve kredi taleplerini görüntüleme ve onaylama sorumluluğu temsilcilere aittir.
- İlgilendikleri müşterilerin işlemlerini (para çekme, yatırma ve transfer) görüntüleyebilmektedir.

Banka Müdürü;

- Bankanın genel durumunu (gelir, gider, kar ve toplam bakiye) görüntüleyebilmektedir.
- Yeni para birimi (Dolar, Euro, Sterling vs.) ekleyebilir ve kur değerlerini güncelleyebilir.

- Çalışanların maaş ücretlerini belirleyebileceklerdir.
- Kredi ve gecikme faiz oranını belirler.
- Müşteri ekleyebilir. Sisteme yeni bir müşteri eklenmesi durumunda en az müşteriye sahip olan temsilciye atanır.
- Sistemi bir ay ilerletebilir.
- Bankada gerçekleşen tüm işlemleri (para çekme, yatırma ve transfer) görüntüleyebilmektedir.
- Listelenen işlemlerin aynı anda başlatılması durumunda deadlock oluşup, oluşmadığının analizinin yapabilmektedir.

3 Araştırmalar

Projemize başlamadan önce, MySQL'in ilişkisel veritabanı oluşturabilmemiz için en iyi ortam olduğuna karar verdik. Ardından ilişkisel veritabanları hakkında bilgi topladık. İlişkisel veritabanı, birbiriyle ilişkili veri noktalarını depolayan ve bunlara erişim sağlayan bir veritabanı türüdür. İlişkisel veritabanları, veriyi tablolarla temsil etmenin sezgisel, kolay anlaşılır bir yolu olan ilişkisel modeli temel alır. İlişkisel veritabanında, tablodaki her satır, anahtar adı verilen benzersiz kimliğe sahip bir kayıttır. Tablonun sütunlarında verilerin öznitelikleri bulunur ve her kayıt genellikle her öznitelik için bir değer taşır. Bu da veri noktaları arasında ilişki kurmayı kolaylaştırır.

3.1 İlişkisel veritabanı yönetim sisteminin faydaları

Kolay ama güçlü bir yapıdaki ilişkisel model, her tür ve büyüklükteki kurumlar tarafından çok çeşitli bilgi ihtiyaçları için kullanılmaktadır. İlişkisel veritabanları, stokları takip etmek, e-ticaret işlemlerini gerçekleştirmek, çok büyük miktarlarda misyon açısından kritik müşteri bilgisini yönetmek ve çok daha fazlasını yapmak için kullanılıyor. Veri noktalarının birbiriyle ilgili olduğu; güvenli, kurallara dayalı ve tutarlı bir şekilde yönetilmesi gereken tüm bilgi ihtiyaçları için ilişkisel veritabanı kullanmayı düşünebilirsiniz.

İlişkisel veritabanları 1970'lerden bu yana kullanılıyor. İlişkisel model, avantajları sayesinde günümüzde de veritabanları için en yaygın kabul gören model olmayı sürdürüyor.

Bu bilgileri edindikten sonra projemize başlamaya hazır hale geldik ve başladık.

4 Yöntem

Bu kısımda projemizi oluştururken izlediğimiz yöntemleri açıklayacağız.

Biz projemizi yaparken ilişkisel veritabanı modeli kullanarak ilgili tabloları MySQL ile oluşturup JAVA ile connection kurup ilgili sorgular ile istekleri sağlamayı hedefledik. Müşteriler için belirtilen istekleri kaba taslak şu şekilde sağladık. Bir müşteri kendi hesabına para ekleyebilir ve hesabından para çekebilir, yeni hesap açma ve var olan hesabını silmek için müşteri temsilcisine talepte bulunabilir ,hesapları arasında para transferi yapabilir, kredi

borcu ödeyerekte bankaya para transferi yapabilir, sisteme kayıt olurken vermiş olduğu bilgilerini güncelleyebilir, bankadan kredi talebinde bulunabilir ancak krediyi sadece Türk Lirası cinsinden talep edebilir. Son olarakta son bir ay içerisinde yapmış olduğu bütün hareketleri ve işlemleri özet olarak görüntüleyebilir. Bir müşteri temsilcisinin kendine ait bir müşterisi vardır ve her müşterinin sadece bir temsilcisi vardır. Sistem ikinci bir temsilciye izin vermemektedir. Temsilciler müşteri ekleyebilir, silebilir ve kendi müiterilerinin bilgilerini güncelleyebilirler, kendi müşterilerinin genel durumlarını (gelir, gider ve toplam bakiye) görüntüleyebilirler, müşterilerden gelen hesap açma, silme ve kredi taleplerini görüntüleme ve onaylama sorumlulukları vardır. Son olarak ilgilendikleri müşterilerin işlemlerini (para çekme, yatırma ve transfer) görüntüleyebilmektedirler. Banka müdürü ise, Bankanın genel durumunu (gelir, gider, kar ve toplam bakiye) görüntüleyebilir, Yeni para birimi (USD, EUR, GBP vs.) ekleyebilir ve kur değerlerini güncelleyebilir, banka çalışanı olan temsilcilerin maaşlarını belirler, krediler için faiz ve gecikme faizi oranını belirleyebilir, sisteme müşteri ekleyebilir ve son olarak görüntülemek istediği kadar işlemi aylık özet olarak görüntüleyebilir.

Müşteriler, müşteri temsilcileri ve banka müdürü bunları yaparken ilk olarak JAVA, SQL tabloları ile bağlantı kurar ve kurmuş olduğumuz sorgu cümleleri ile yapılmak istenen işlemi sağlar. İlgili işlem eğer başarılı ise SQL tabloları güncellenir. Ancak istenen işlem başarısız olursa, örneğin bir temsilci temsilcisi olmadığı bir müşteriye erişmek isterse sistem buna izin vermez ve temsilciye uyarı vererek girilen bilginin değiştirilme-

sini ister. Bu tarz uyarı mesajları projemizin karmaşaya müsait her isterine entegre edilmiştir. Böylece kargaşanın önüne geçilirken bir yandan da isterlerin tam sağlanması hedeflenmiştir.

5 Geliştirme Ortamı

Veritabanı oluşturmak için MySQL programı kullandık. Diğer işlemler için JAVA kullandık.

6 Kod Bilgisi

6.1 Akış Diyagramı-ER Diyagramı

Kısım ektedir. [1](Akış Diyagramı) Kısım ektedir. [2](ER Diyagramı)

7 Sonuç

Arayüz tasarımı hariç projenin gerektirdiği tüm isterleri sağladık.

8 Kaynakça

1. <https://www.oracle.com/tr/database/what-is-a-relational-database/>
2. <https://tr.wikipedia.org/wiki/>
3. <https://stackoverflow.com/questions/7598623/how-do-i-set-the-classpath-in-netbeans>
4. <https://mertcangokgoz.com/temel-mysql-komutlari/>

5. <https://www.yazilimkodlama.com/java/java-mysql-veritabani-baglantisi-select-insert-update-delete/>
6. <https://www.lifeacode.com/sql-dersleri/sql-tablo-islemleri-alter-table.html>
7. <https://stackoverflow.com/questions/10896151/java-sql-sqlexception-parameter-index-out-of-range-1-number-of-parameters-wh>
8. <https://www.tutorialspoint.com/jdbc/jdbc-select-records.htm>
9. <https://www.web-gelistirme-sc.com/tr/java/next-ve-nextline-yontemleri-arasindaki-scanner-sinifindaki-fark-nedir/1044541919/>
10. <https://app.diagrams.net/> -Akış di-yagramı oluşturmak için.