

BLM210 PROGRAMLAMA LABORATUVARI II

PROJE 3

PROJE TESLİM TARİHİ: 13.05.2022

BANKA YÖNETİM SİSTEMİ

Projenin Amacı:

Bu projede bir bankanın yönetim sistemi için bir veritabanı tasarlamamız ve bu veritabanının üzerinde gerekli işlemleri gerçekleyen bir uygulama geliştirmemiz amaçlanmaktadır.

Programlama Dili:

Proje C++/Java/Python/C# dillerinden biri kullanılarak form projesi olarak gerçekleştirilebilir. Veri tabanı yönetim sistemi için MySQL, PostgreSQL veya MsSQL kullanılabilir.

Açıklama:

Tasarlanan 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 ulaşabilmesi sağlanmış olacaktır. Ayrıca, veritabanı kullanarak müşteri istekleri ve bankanın ihtiyaçları doğrultusunda raporlar hazırlanabilecektir.

Bir veritabanı tasarımının ilk aşamasında sistemin ihtiyaçlarının belirlenmesi ve depolanacak bilgi türlerinin tanımlanması için Varlık-İlişki (ER) diyagramı oluşturulmalıdır. ER diyagramı sistem içerisinde var olabilecek varlıkların ve aralarındaki ilişkilerin görsel olarak ifade edilmesi için kullanılır. Geliştirme sırasında, ER diyagramı gereksinimlerin daha açık ve özlü bir şekilde haritalanmasına yardımcı olmaktadır.

Sizlerden problemlerin çözümü için bir veritabanı tasarımı yapmanız istenmektedir. Tablo sayısı tüm tablolar en az 3NF normalizasyon formuna uyacak şekilde size bırakılmıştır.

Projede oluşturulacak veritabanının ER diyagramının oluşturulması ve arayüzde gösterilmesi gerekmektedir. Diyagram üzerinden gerçekleştirilen normalizasyon işlemlerinin adım adım gösterilmelidir.

ER diyagramı: <https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/>

Normalization : <https://www.guru99.com/database-normalization.html>

Problem Tanımı:

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.

Sistemdeki Roller:

- Müşteriler;
 - o Hesaplarından para çekebilirler ve yatırabilirler.
 - o Yeni hesap açma ve var olan bir hesabı silme talebinde bulunabilirler.
 - Bakiyesi “0” olmayan bir hesap silinemez.
 - o Birbirleri arasında para transferi yapabilirler.
 - Farklı para birimlerine sahip hesaplar arası transferler sırasında gönderilen miktar hedef para birimine otomatik olarak çevrilmelidir.
 - o Bilgilerini güncelleyebilirler. (Adres, Telefon vs.)
 - o Bankaya para transferi yapabilirler. (Kredi borcu ödeme)
 - o Bankadan kredi talep edebilirler.
 - Kredi sadece TL cinsinden talep edilebilmektedir.
 - Bankanın kredi talebini onaylaması durumunda istenilen vade oranınca (faiz ve anapara toplamı) bölünerek aylara borç olarak yansıtılır.
 - Aylık özet görüntülemeye kredi borcu ödemeleri için ödenen faiz ve anapara ayrı ayrı görüntülenmelidir.
 - Müşterinin aylık borcunun tamamını ödememesi durumunda kalan borç ek faiz hesaplanarak bir sonraki aya devreder.
 - Faiz ve gecikme faiz oranı banka müdürü tarafından belirlenir.
 - Aylık borç ve kalan borç ayrı ayrı görüntülenmeli. (müşteri isterse tüm borcunu tek seferde ödeyebilir)
 - Erken ödeme durumlarında gelecek aylar için faiz alınmayacaktır.
 - o Aylık özetlerini görüntüleyebilirler. (Geçerli ay içerisinde yaptığı para gönderme, çekme, kredi borcu ödeme gibi işlemlerin özeti)
- Banka müdürü;
 - o Bankanın genel durumunu (gelir, gider, kar ve toplam bakiye) görüntüleyebilmektedir.
 - o Yeni para birimi (Dolar, Euro, Sterling vs.) ekleyebilir ve kur değerlerini güncelleyebilir.
 - o Çalışanların maaş ücretlerini belirleyebilecektir.
 - Tek bir çalışan türü vardır (müşteri temsilcisi). Hepsinin maaş miktarı aynıdır.
 - o Kredi ve gecikme faiz oranını belirler.
 - o Müşteri ekleyebilir.
 - Sisteme yeni bir müşteri eklenmesi durumunda en az müşteriye sahip olan temsilciye atanır.
 - o Sistemi bir ay ilerletebilir.
 - İsterlerin test edilebilmesi için sizlerden uygulama tarihini bir ay ileriye öteleyebilmeniz istenmektedir.
 - Bu ilerletme işlemi sonucunda -> maaşların ödenmesi, gelir-gider durumlarının güncellenmesi ve müşterilerin bir sonraki aya ait borçlarının kendilerine yansıtılması gerekmektedir.

- o Bankada gerçekleşen tüm işlemleri (para çekme, yatırma ve transfer) görüntüleyebilmektedir.
 - İşlemleri listelerken “son X adet işlemi listele” şeklinde bir seçenek sunulmalıdır.
 - Örnek olarak “son 5 işlemi listele” sorgusunun çıktısı Tablo I’de gösterilmiştir.
- o Listelenen işlemlerin aynı anda başlatılması durumunda deadlock oluşup, oluşmadığının analizinin yapabilmektedir. Deadlock analizi ayrı bölümde açıklanacaktır.

İşlem No	Kaynak	Hedef	İşlem	Tutar	Kaynak Bakiye	Hedef Bakiye	Tarih
1	Hesap1	Hesap2	Para Gönderme	100	900	1100	11.04.22
2	Hesap1	Banka	Borç Ödeme	150	750	null	13.04.22
3	Kullanıcı1	Hesap1	Para Yatırma	200	null	950	15.04.22
4	Hesap2	Hesap1	Para Gönderme	200	900	1150	11.05.22
5	Hesap2	Kullanıcı2	Para Çekme	500	400	null	17.05.22

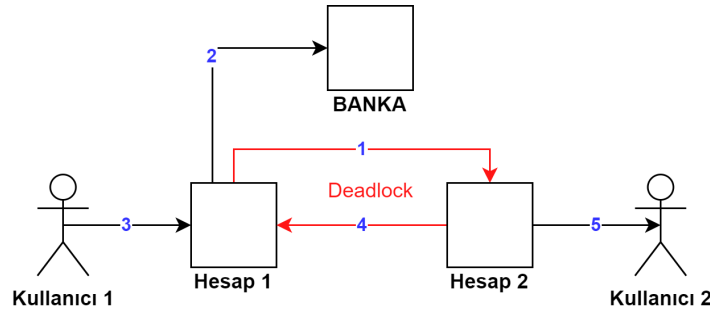
Tablo 1. İşlem Tablosu

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

Deadlock Analizi:

Para gönderimi sırasında hedefin işlem yapması engellenmektedir. Bu nedenle para almakta olan bir hesap para gönderimi yapamamaktadır. Deadlock analizi için tüm işlemlerin aynı anda çalışmaya başladığı ve paralel şekilde çalıştığı kabul edilecektir.

Tablo I için deadlock analizi yapmak istersek :



Şekilde görüldüğü üzere aynı anda başlamış olan 1 ve 4 işlemleri birbirlerini kilitlemektedir. Sistem çıktı olarak deadlock sayısını ve deadlock olan olan işlemler döndürecektir.

Örneğin Tablo I için :

Deadlock sayısı : 1

İşlemler : (1-4) şeklinde bir çıktının arayüzde görüntülenmesi gerekmektedir.

Birden fazla deadlock durumunda çıktı aşağıdaki şekildeki gibi gösterilmelidir.

Deadlock sayısı : 3

İşlemler : (1-3-4) , (2-7) , (8-9)

İsterler ve Puanlama

- **Veritabanı Tasarımı (30p)**
 - Gerekli tüm bilgilerin ilişkisel bir veritabanında saklanması (15p)
 - Tüm tabloların 3NF formunda normalize edilmesi (15p)
- **Arayüz Tasarımı (20p)**
 - Tabloların arayüz üzerinden görüntülenmesi (15p)
 - Tablolar üzerinde ekleme, silme ve çıkarma işlemlerinin yapılabilmesi (5p)
- **Rollerin Gerçeklenmesi (30p)**
- **Deadlock Analizi (10p)**
- **Rapor (10p) (Rapor hazırlamayan gruplar sunuma alınmayacaktır)**
- **Proje değerlendirmeleri sunum performansına göre yapılacak olup tüm isterler bütüncül olarak ele alınacaktır. Yukarıdaki verilen puan gösterimleri tekil olarak yapılan bir isterden tam puan alınmasını garanti etmemektedir dolayısıyla tüm isterlerin karşılanması önem taşımaktadır.**

Notlar:

- Rollerin kullanılması için arayüzde 3 adet panel (müşteri, temsilci, banka müdürü) tasarlanmalıdır.
- Gerekli olan tüm bilgiler ilişkisel bir veritabanında saklanmalıdır (MSSQL, MySQL, PostgreSQL).
- Veritabanı en az 3NF optimizasyona sahip olacak şekilde tasarlanmalıdır.
- Gerçekleştirilen tüm işlemler bir GUI üzerinden görüntülenebilmelidir.
 - Java için -> swing, javafx vs.
 - C++ için -> Qt, GTK
 - Python için -> Tkinter, PyQt

- o C# -> Wpf , Winforms gibi kütüphaneler kullanılabilir.
- ER tasarımının raporda bulunması önemlidir.
 - o Diyagram raporda bulunmalıdır ancak sunum sırasında bilgisayardan da görsel olarak (png, pdf vs.) gösterilmelidir.

Ödev Teslimi

- Proje en fazla iki kişilik gruplar halinde yapılacaktır. Her öğretim sadece kendi içerisinde grup olabilecektir.
- Rapor iee formatında (önceki yıllarda verilen formatta) 4 sayfa, akış diyagramı veya yalancı kod içeren, özet, giriş, yöntem, deneysel sonuçlar, sonuç ve kaynakça bölümünden oluşmalıdır.
- Dersin takibi projenin teslimi dahil edestek.kocaeli.edu.tr sistemi üzerinden yapılacaktır. edestek.kocaeli.edu.tr sitesinde belirtilen tarihten sonra getirilen projeler kabul edilmeyecektir.
- Proje ile ilgili sorular edestek2.kocaeli.edu.tr sitesindeki forum üzerinden Arş.Gör. Abdurrahman GÜN veya Arş.Gör. Tuğrul Hakan GENÇTÜRK'e sorulabilir.
- Demo tarihleri daha sonra duyurulacaktır.
- Demo sırasında algoritma, geliştirdiğiniz kodun çeşitli kısımlarının ne amaçla yazıldığı ve geliştirme ortamı hakkında sorular sorulabilir.
- Kullandığınız herhangi bir satır kodu açıklamanız istenebilir.
- Göndermiş olduğunuz kodlar benzerlik testinden geçirilecektir. Benzerlik çıkması halinde aşağıdaki puanlama uygulanacaktır.
 - Gruplar arası Benzer çıkma (kod parçası alma) -projeden -(eksi)50 puan
 - Para karşılığı başka birine yaptırma projeden -(eksi)100 puan
 - İnternette kodu (kod parçası) alma projeden -(eksi)50 puan