



Bilgisayar Programcılığı

BYP154 Veritabanı Yönetim Sistemleri
Proje Raporu

Mobil Tarife ve Kampanya Yönetim Sistemi

Öğrenciler :

ID	İsim & Soyisim
240757022	Ahmet Rasim BEYHAN
240757039	Samet YILMAZ

TABLE OF CONTENTS

1	INTRODUCTION	4
1.1	PROJECT DESCRIPTION.....	4
2	ENTITY RELATIONAL MODEL	5
2.1	ENHANCED ER DIAGRAM.....	5
2.2	RELATIONAL SCHEMA & MAPPING	6
3	NORMALIZATION.....	10
3.1	FUNCTIONAL DEPENDENCIES.....	10
3.2	UNNORMALISED FORM	10
3.3	FIRST NORMAL FORM.....	11
3.4	SECOND NORMAL FORM	11
3.5	THIRD NORMAL FORM.....	11
4	APPLICATION(for teams of three members).....	8

LIST OF FIGURES

Figure 1: EER diagram of the Company Employee Administrative Database	5
Figure 2: Relational schema of the Company Employee Administrative Database with arrows indicating referential integrity.....	6

1 INTRODUCTION

1.1 PROJECT DESCRIPTION

Mobil Operatör Tarife ve Kampanya Yönetim Sistemi (MOTCMS), mobil hizmet planlarını, promosyonları, müşterileri, abonelikleri, kullanımı, faturalandırmayı, ödemeleri, envanteri ve sistem ayarlarını tanımlamak ve yönetmek için gereken tüm idari verileri depolar. Aşağıdaki temel veriler gereksinim analizi sırasında tanımlanmıştır ve veritabanında temsil edilecektir:

Tarifeler: Her tarifenin benzersiz bir kimliği, adı, aylık ücreti, ses dakikası hakkı, veri hakkı (MB/GB), SMS kotası ve geçerlilik süresi (gün) vardır.

Kampanyalar: Her kampanyanın benzersiz bir kimliği, adı, açıklaması, başlangıç ve bitiş tarihleri, indirim oranı, bağlantılı tarifeleri ve uygunluğu yöneten iş kuralları vardır.

Abonelikler: Hangi müşterinin hangi tarifeye (ve varsa kampanyaya) abone olduğunu, abonelik başlangıç ve bitiş tarihleri dahil olmak üzere izler.

Kullanım Kayıtları: Kullanılan ses dakikalarının, tüketilen verilerin ve gönderilen SMS'lerin abonelik başına günlük kayıtları.

Faturalandırma ve Ödemeler: Abonelik dönemi başına (ay/yıl) oluşturulan faturalar, toplam tutar, düzenleme tarihi, vade tarihi, ödeme durumu; ayrıca ödeme işlemleri (tutar, tarih, yöntem).

Yönetimsel Modüller: Kullanıcı ve rol yönetimi, izinler, sistem ayarları (anahtar-değer çiftleri), denetim günlükleri (EKLE/GÜNCELLE/SİL), bildirimler, personel yorumlarıyla destek biletleri, ürün ve cihaz envanteri, siparişler, stok hareketleri ve sistem hata takibi.

2 ENTITY RELATIONAL MODEL

2.1 ENHANCED ER DIAGRAM

(Below is an example, you should create your own EER Diagram related to your project.)

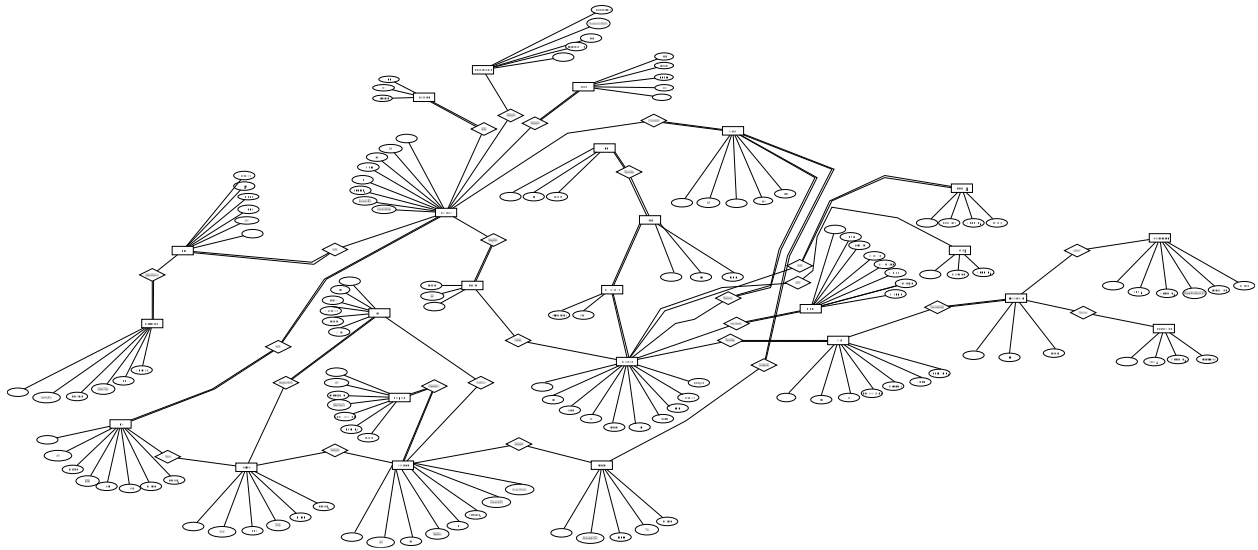


Figure 1: EER diagram of the Company Employee Administrative Database

2.2 RELATIONAL SCHEMA & MAPPING

(Below is an example, you should create your own Relational Schema related to your project. Do not forget to show your PKs (underlined> and FKs (dotted).)

(Show the intermediate steps !!!

- Mapping a regular entity
- Mapping a composite attribute
- Mapping a multivalued attribute
- Mapping Weak Entities
- Mapping Binary Relationships
- Mapping Unary Relationships

)

Telefon: ID (PK), AboneID (FK), Numara, Tip, Durum

SimKart: ID (PK), Kod, Barkod, PUK, Durum, AboneID (FK)

Cihaz: ID (PK), Ad, SeriNo, IMEI, Marka, Model, Durum, AboneID (FK)

Varlik: ID (PK), Ad, Barkod, SeriNo, Tip, Durum, KayitTarihi

StokHareket: ID (PK), VarlikID (FK), IslemTarihi, IslemTipi, Miktar, KayitTarihi

AdNetwork: ID (PK), Ad, SektorID (FK), WebAdresi, Telefon, Eposta

ApiAyarlari: ID (PK), Ad, Baslik, URL, KullaniciID (FK), Durum, SonGuncellemeTarihi

Konum: ID (PK), Ad, Il, Ilce, Ulke

Sektorler: ID (PK), Ad, Aciklama

Kurumlar: ID (PK), Ad, Adres, Telefon, Fax, SektorID (FK), KonumID (FK), KurumKodu

Aboneler: ID (PK), TCNo, Ad, Soyad, DogumTarihi, Eposta, Telefon, Adres, KayitTarihi, KurumID (FK)

Abonelik: ID (PK), AboneID (FK), TarifeID (FK), KampanyaID (FK), BaslangicTarihi, BitisTarihi, KayitTarihi

Tarife: ID (PK), Ad, Fiyat, Sure, Aciklama, KayitTarihi

Kampanya: ID (PK), Ad, BaslangicTarihi, BitisTarihi, SonKullanımTarihi, IndirimliFiyat, Aciklama

IliskiliAyar: ID (PK), Ad, Tip, Deger, KullaniciID (FK), OlusturmaTarihi

Yetkili: ID (PK), Ad, Unvan, Telefon, Eposta

Pozisyon: ID (PK), Ad, Aciklama

Kullanici: ID (PK), Ad, Soyad, Email, Telefon, Sifre, TCNo, PozisyonID (FK), KayitTarihi, AktifMi

Rol: ID (PK), Ad, Aciklama

Yetki: ID (PK), Ad, Aciklama

KullaniciRol: KullaniciID (PK, FK), RolID (PK, FK)

RolYetki: RolID (PK, FK), YetkiID (PK, FK)

BildirimTuru: ID (PK), Ad, Aciklama

Bildirim: ID (PK), Ad, Detay, GecikmeSuresi, OlusturmaTarihi, AktifMi, BildirimTuruID (FK)

BildirimAtama: ID (PK), BildirimID (FK), AtananKullaniciID (FK), AtayanKullaniciID (FK),
AtanmaTarihi, Durum

BildirimOkunma: ID (PK), BildirimID (FK), KullaniciID (FK), OkunmaTarihi

GirisLog: ID (PK), KullaniciID (FK), GirisTarihi, IPAdres

CikisLog: ID (PK), KullaniciID (FK), CikisTarihi

AuditTrail: ID (PK), TabloAd, Islemturu, DegisenAlan, EskiDeger, YeniDeger, KullaniciID (FK),
DegistirmeTarihi

SistemAyarlar: ID (PK), ParametreAdi, Deger, OlusturmaTarihi, GuncellemeTarihi

Takvim: ID (PK), Baslik, BaslangicTarihi, BitisTarihi, KullaniciID (FK), Oncelik, Aciklama

TeleServices: ID (PK), ServisAdi, Baslangic, Bitis, KullaniciID (FK), Tutar

SMS: ID (PK), GonderenKullaniciID (FK), AliciNumara, MesajIcerigi, GonderimTarihi, Adet, Tutar

Fax: ID (PK), GonderenKullaniciID (FK), AliciNumara, GonderimTarihi, Adet, Tutar

Cagiri: ID (PK), ArayanTelefonID (FK), ArananTelefonID (FK), Baslangic, Bitis, Sure, Ucret

BYP154 VERİTABANI YÖNETİM SİSTEMLERİ

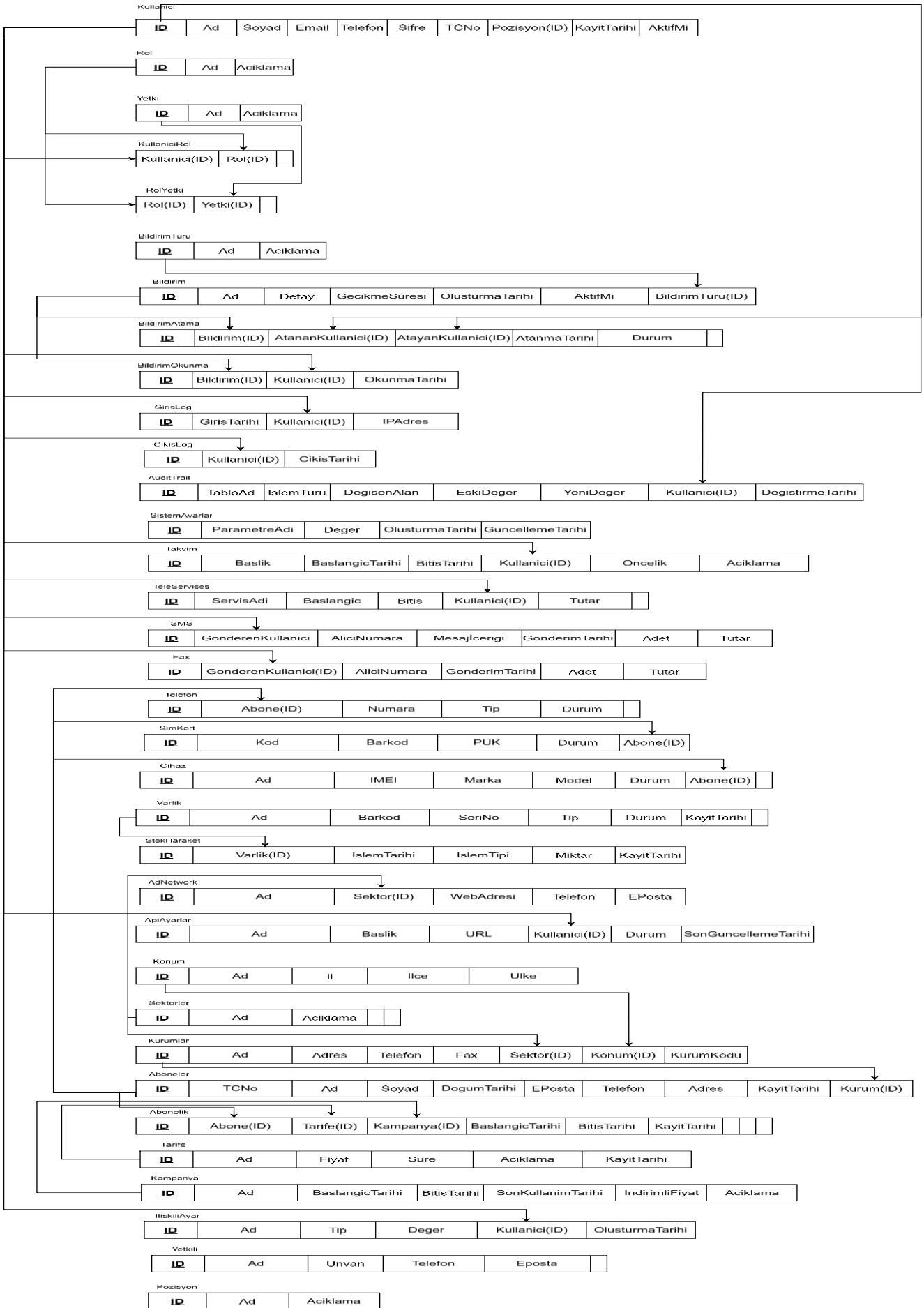


Figure 2: Relational schema of the Company Employee Administrative Database with arrows indicating referential integrity

3 NORMALIZATION

(Apply the normalization process to the model.)

3.1 FUNCTIONAL DEPENDENCIES

(Below is an example showing the functional dependencies, you should do your own project.)

(THE FOLLOWING IS THE SET OF FUNCTIONAL DEPENDENCIES (FD'S) OF MOBİL SİSTEM.)

F = {

FD1: KULLANICIID → AD, SOYAD, EMAIL, TELEFON, SIFRE, TCNO, POZISYONID, KAYITTARIHI, AKTIFMI

FD2: ROLID → AD(ROL), ACIKLAMA(ROL)

FD3: YETKIID → AD(YETKI), ACIKLAMA(YETKI)

FD4: TELEFONID → ABONEID, NUMARA, TIP, DURUM

FD5: SIMKARTID → KOD, BARKOD, PUK, DURUM, ABONEID

FD6: CIHAZID → AD(CIHAZ), SERINO, IMEI, MARKA, MODEL, DURUM, ABONEID

FD7: ABONEID → TCNO, AD(ABONE), SOYAD, DOGUMTARIHI, EPOSTA, TELEFON, ADRES, KAYITTARIHI, KURUMID

FD8: TARIFEID → AD(TARIFE), FIYAT, SURE, ACIKLAMA, KAYITTARIHI

FD9: KAMPANYAID → AD(KAMPANYA), BASLANGICTARIHI, BITISTARIHI, SONKULLANIMTARIHI, INDIRIMLIFİYAT, ACIKLAMA

FD10: ABONELIKID → ABONEID, TARIFEID, KAMPANYAID, BASLANGICTARIHI, BITISTARIHI, KAYITTARIHI

FD11: BILDIRIMID → AD(BILDIRIM), DETAY, GECIKMESURESİ, OLUSTURMATARIHI, AKTIFMI, BILDIRIMTURUID

FD12: TAKVIMID → BASLIK, BASLANGICTARIHI, BITISTARIHI, KULLANICIID, ONCELIK, ACIKLAMA

FD13: CAGRIID → ARAYANTELEFONID, ARANANTELEFONID, BASLANGIC, BITIS, SURE, UCRET

}

3.2 UNNORMALISED FORM

Veriler tekrarlı gruplar içeriyor.

Örn: Abone tablosunda aynı kişiye ait çoklu telefonlar, cihazlar ve SIM kartlar tek satırda virgülle listelenebilir.

Bir satırda:

AboneID | Ad | Telefon1, Telefon2, ... | Cihaz1, Cihaz2, ... | SimKart1, SimKart2, ...

Bu yapı 1NF'ye uygun değil.

3.3 FIRST NORMAL FORM

Tekrarlı gruplar ayrıldı, her alan atomik hale getirildi.

Telefon, Cihaz, SimKart gibi çoklu değerler ayrı tablolara ayrıldı ve AboneID ile ilişkilendirildi.

Artık her satır sadece tek telefon, tek cihaz veya tek SIM kartı gösterir.

Tüm tablolar birincil anahtar (PK) içerecek şekilde tasarlandı.

3.4 SECOND NORMAL FORM

Tüm tablolar tam anahtara bağlı hale getirildi.

Örn: Abonelik tablosu AbonelikID'ye bağlı, kısmi bağımlılık kalmadı.

KullaniciRol ve RolYetki gibi ara tablolar, bileşik anahtar (KullaniciID, RolID) yapısına getirildi.

Yani, kısmi bağımlılık ortadan kaldırıldı.

3.5 THIRD NORMAL FORM

Transitif bağımlılıklar kaldırıldı.

Örn: Kullanici tablosunda sadece PozisyonID tutulur, pozisyonun adı ve açıklaması Pozisyon tablosunda kalır.

Kullanici → PozisyonID → PozisyonAd zinciri kesildi, böylece 3NF sağlandı.

Tüm tablolar artık:

PK → diğer alanlar şeklinde net bağımlılık içeriyor,

başka tabloya bağlı alanları ise sadece FK üzerinden bağlı.

4 APPLICATION

This step applies to projects with three members. Teams of three are expected to develop an application using the database they created after completing the previous steps. For example, a web application.