



SAKARYA ÜNİVERSİTESİ

Veritabanı Yönetim Sistemleri

2023-2024 güz dönemi

Ödev Raporu

YAZAN ALSOLYMAN

yazan.alsolyman@ogr.sakarya.edu.tr

B211210572

AHMED FOUAD FAROUK MOHAMED

fouad.mohamed@ogr.sakarya.edu.tr

B211210575

2023

BİLGİSAYAR MÜHENDİSLİĞİ

PROBLEMİN TANITIMI

Projemiz bir kitap veritabanı tasarımını temsil ediyor. Tablolar, yayınevleri, yazarlar, kitaplar ve bu kitapların türleri, ödülleri, satışları, çevirileri ve değerlendirmeleri gibi bilgileri içeriyor.

İŞ KURALLARI

- Bir yazar çok sayıda kitap yazabilir ve en az bir.
- Bir kitap yalnızca bir yazar tarafından yazılabilir ve en az bir.
- Bir yayınevi çok sayıda kitap yayınlayabilir ve en az bir.
- Bir kitap yalnızca bir yayınevi tarafından yayınlanabilir ve en az bir.
- Bir yazar çok sayıda kitapları olabilir ve en az bir.
- Yazar kitapları yalnızca bir yazar tarafından yazılabilir ve en az bir.
- Bir yazar çok sayıda ödül kazanabilir ya da hiç.
- Yazar ödülleri yalnızca bir yazar tarafından kazanılabilir ve en az bir.
- Adisyonun servisTurNosu, personelID ,tarihi ve durumu vardır.
- Bir kitap farklı türü olabilir (kalıtım) -> (manga , felsefe , macera ve polisiye).
- Bir kitap çok sayıda değerlendirmesi olabilir ve en az bir.
- Bir kitap değerlendirmesi yalnızca bir kitaba ait olabilir.
- Bir kitap çok sayıda çevirileri olabilir ya da hiç.
- Bir kitap çeviri yalnızca bir kitaba ait olabilir.
- Bir kitap çok sayıda satışı olabilir ya da hiç
- Bir kitap satışı yalnızca bir kitaba ait olabilir.
- Bir kitap çok sayıda ödül kazanabilir ya da hiç.
- Bir kitap ödülü yalnızca bir kitaba ait olabilir.
- Bir kitap çok sayıda formatı olabilir ve en az bir.
- Bir format çok kitaba ait olabilir ve en az bir.
- Kitap ile format çok – çok ilişkisi olduğundan formatArac bir tablo oluşturulacak.

İLİŞKİSEL ŞEMA

yayincilar (yayinevi_id:(SERIAL) , yayinevi_adi:(VARCHAR) , kurulusTarihi (DATE))

yazar (yazar_id:(SERIAL) , ad:(VARCHAR) , soyad:(VARCHAR) , ulke:(VARCHAR))

kitap (kitap_id:(SERIAL) , sayfaSayisi:(INTEGER) , yayinTarihi:(DATE) , yazar_id:(INTEGER) , yayinevi_id:(INTEGER))

manga (kitap_id:(INTEGER) , manga_adi:(VARCHAR))

felsefe (kitap_id:(INTEGER) , felsefe_adi:(VARCHAR))

macera (kitap_id:(INTEGER) , macera_adi:(VARCHAR))

polisiye (kitap_id:(INTEGER) , polisiye_adi:(VARCHAR))

yazarOdulleri (yOdulleri_id:(SERIAL) , odul_adi (VARCHAR) , odul_yili:(DATE) , yazar_id:(INTEGER))

yazarKitaplari (yKitaplari_id:(SERIAL) , yazar_id:(INTEGER) , kitap_id:(INTEGER))

kitapSatislari (Satislari_id:(SERIAL) , satisMiktari:(INTEGER) , kitap_id:(INTEGER))

kitapCevirileri (Cevirileri_id:(SERIAL) , dil:(VARCHAR) , kitap_id:(INTEGER))

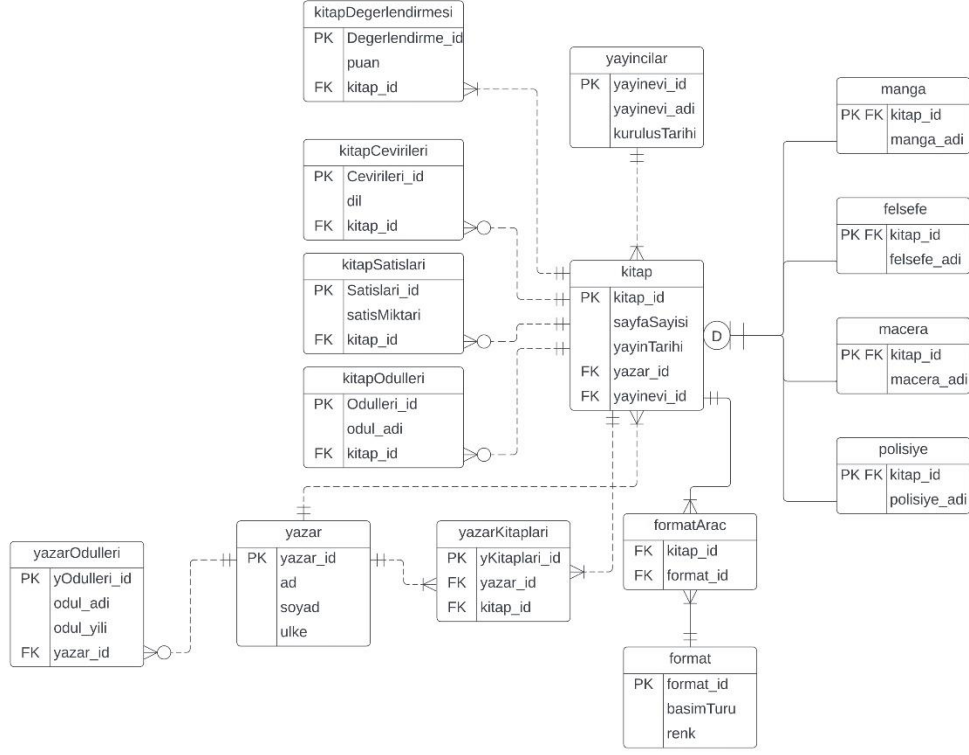
kitapDegerlendirmesi (Degerlendirme_id:(SERIAL) , puan:(INTEGER) , kitap_id:(INTEGER))

kitapOdulleri (Odulleri_id:(SERIAL) , odul_adi:(VARCHAR) , kitap_id:(INTEGER))

formatArac (kitap_id:(INTEGER) , format_id:(SERIAL))

format (format_id:(SERIAL) , basimTuru:(VARCHAR) , renk:(VARCHAR))

VARLIK BAGINTI MODELİ



KODLAR :

```
CREATE DATABASE kitapProjesi;  
CREATE SCHEMA kitap;
```

```
CREATE TABLE "yayincilar" (  
  "yayinevi_id" SERIAL PRIMARY KEY,  
  "yayinevi_adi" VARCHAR(50) NOT NULL,  
  "kurulusTarihi" DATE  
);
```

```
CREATE TABLE "yazar" (  
  "yazar_id" SERIAL PRIMARY KEY,  
  "ad" VARCHAR(50) NOT NULL,  
  "soyad" VARCHAR(50) NOT NULL,  
  "ulke" VARCHAR(50) NOT NULL  
);
```

YEŞİL -> Kalıtım İlişkisi

```
CREATE TABLE kitap.kitap (  
  "kitap_id" SERIAL,  
  "sayfaSayisi" INTEGER,  
  "yayinTarihi" DATE NOT NULL,  
  "yazar_id" INTEGER REFERENCES "yazar"("yazar_id") NOT NULL,  
  "yayinevi_id" INTEGER REFERENCES "yayincilar"("yayinevi_id")  
  NOT NULL,  
  CONSTRAINT "kitapPK" PRIMARY KEY ("kitap_id")  
);
```

```
CREATE TABLE kitap.manga (  
  "manga_id" SERIAL PRIMARY KEY,  
  "manga_adi" VARCHAR(50) NOT NULL,  
  "yayinevi_id" INTEGER REFERENCES "yayincilar"("yayinevi_id")  
  NOT NULL,  
  "yazar_id" INTEGER REFERENCES "yazar"("yazar_id") NOT NULL,  
  "yayinTarihi" DATE NOT NULL,  
  "sayfaSayisi" INTEGER  
);
```

```
"kitap_id" INTEGER,  
"manga_adi" VARCHAR(50) NOT NULL,  
CONSTRAINT "mangaPK" PRIMARY KEY ("kitap_id")  
);
```

```
CREATE TABLE kitap.felsefe (  
"kitap_id" INTEGER,  
"felsefe_adi" VARCHAR(50) NOT NULL,  
CONSTRAINT "felsefePK" PRIMARY KEY ("kitap_id")  
);
```

```
CREATE TABLE kitap.macera (  
"kitap_id" INTEGER,  
"macera_adi" VARCHAR(50) NOT NULL,  
CONSTRAINT "maceraPK" PRIMARY KEY ("kitap_id")  
);
```

```
CREATE TABLE kitap.polisiye (  
"kitap_id" INTEGER,  
"polisiye_adi" VARCHAR(50) NOT NULL,  
CONSTRAINT "polisiyePK" PRIMARY KEY ("kitap_id")  
);
```

```
ALTER TABLE "kitap"."manga"  
ADD CONSTRAINT "mangaFK" FOREIGN KEY ("kitap_id")  
REFERENCES "kitap"."kitap" ("kitap_id")  
ON DELETE CASCADE  
ON UPDATE CASCADE;
```

```
ALTER TABLE "kitap"."felsefe"  
ADD CONSTRAINT "felsefeFK" FOREIGN KEY ("kitap_id")  
REFERENCES "kitap"."kitap" ("kitap_id")
```

ON DELETE CASCADE
ON UPDATE CASCADE;

ALTER TABLE "kitap"."macera"
ADD CONSTRAINT "maceraFK" FOREIGN KEY ("kitap_id")
REFERENCES "kitap"."kitap" ("kitap_id")
ON DELETE CASCADE
ON UPDATE CASCADE;

ALTER TABLE "kitap"."polisiye"
ADD CONSTRAINT "polisiyeFK" FOREIGN KEY ("kitap_id")
REFERENCES "kitap"."kitap" ("kitap_id")
ON DELETE CASCADE
ON UPDATE CASCADE;

CREATE TABLE "yazarOdulleri" (
"yOdulleri_id" SERIAL PRIMARY KEY,
"odul_adi" VARCHAR(50) NOT NULL,
"odul_yili" DATE,
"yazar_id" INTEGER REFERENCES "yazar"("yazar_id")
);

CREATE TABLE "yazarKitaplari" (
"yKitaplari_id" SERIAL PRIMARY KEY,
"yazar_id" INTEGER REFERENCES "yazar"("yazar_id"),
"kitap_id" INTEGER REFERENCES "kitap"."kitap"("kitap_id")
);

CREATE TABLE "kitapSatislari" (
"Satislari_id" SERIAL PRIMARY KEY,
"satisMiktari" INTEGER,
"kitap_id" INTEGER REFERENCES "kitap"."kitap"("kitap_id")

);

```
CREATE TABLE "kitapCevirileri" (  
  "Cevirileri_id" SERIAL PRIMARY KEY,  
  "dil" VARCHAR(50) NOT NULL,  
  "kitap_id" INTEGER REFERENCES "kitap"."kitap"("kitap_id")  
);
```

```
CREATE TABLE "kitapDegerlendirmesi" (  
  "Degerlendirme_id" SERIAL PRIMARY KEY,  
  "puan" INTEGER,  
  "kitap_id" INTEGER REFERENCES "kitap"."kitap"("kitap_id")  
);
```

```
CREATE TABLE "kitapOdulleri" (  
  "Odulleri_id" SERIAL PRIMARY KEY,  
  "odul_adi" VARCHAR(50) NOT NULL,  
  "kitap_id" INTEGER REFERENCES "kitap"."kitap"("kitap_id")  
);
```

```
CREATE TABLE "formatArac" (  
  "kitap_id" INTEGER REFERENCES "kitap"."kitap"("kitap_id"),  
  "format_id" SERIAL,  
  PRIMARY KEY ("kitap_id", "format_id")  
);
```

```
CREATE TABLE "format" (  
  "format_id" SERIAL PRIMARY KEY,  
  "basimTuru" VARCHAR(50),  
  "renk" VARCHAR(50)  
);
```


FUNCTIONS :

1 - kitapEkle :

```
CREATE OR REPLACE FUNCTION kitapEkle(  
    sayfa_sayisi INTEGER,  
    yayin_tarihi DATE,  
    yazar_id INTEGER,  
    yayinevi_id INTEGER  
)  
RETURNS INTEGER AS $$  
DECLARE  
    new_book INTEGER;  
BEGIN  
    INSERT INTO kitap.kitap ("sayfaSayisi", "yayinTarihi",  
"yazar_id", "yayinevi_id")  
    VALUES (sayfa_sayisi, yayin_tarihi, yazar_id, yayinevi_id)  
    RETURNING "kitap_id" INTO new_book;  
  
    RETURN new_book;  
END;  
$$ LANGUAGE plpgsql;
```

```
SELECT kitapEkle(500, '2023-01-01'::DATE, 1, 1);
```

2 - kitapSil :

```
CREATE OR REPLACE FUNCTION kitapSil(book_id INTEGER)
RETURNS VOID AS $$
BEGIN
    DELETE FROM kitap.kitap WHERE "kitap_id" = book_id;
END;
$$ LANGUAGE plpgsql;
```

SELECT kitapSil(1);

3 - kitapGetir :

```
CREATE OR REPLACE FUNCTION kitapGetir(book_id INTEGER)
RETURNS TABLE (
    kitap_id INTEGER,
    sayfa_sayisi INTEGER,
    yayin_tarihi DATE,
    yazar_id INTEGER,
    yayinevi_id INTEGER
) AS $$
BEGIN
    RETURN QUERY SELECT k."kitap_id", k."sayfaSayisi",
k."yayinTarihi", k."yazar_id", k."yayinevi_id"
    FROM kitap.kitap AS k
    WHERE k."kitap_id" = book_id;
END;
$$ LANGUAGE plpgsql;
```

SELECT * FROM kitapGetir(1);

4 - kitapGuncelle :

```
CREATE OR REPLACE FUNCTION kitapGuncelle(  
    p_kitap_id INTEGER,  
    p_sayfa_sayisi INTEGER,  
    p_yayin_tarihi DATE,  
    p_yazar_id INTEGER,  
    p_yayinevi_id INTEGER  
)  
RETURNS VOID AS $$  
BEGIN  
    UPDATE kitap.kitap  
    SET  
        "sayfaSayisi" = p_sayfa_sayisi,  
        "yayinTarihi" = p_yayin_tarihi,  
        "yazar_id" = p_yazar_id,  
        "yayinevi_id" = p_yayinevi_id  
    WHERE "kitap_id" = p_kitap_id;  
END;  
$$ LANGUAGE plpgsql;  
  
SELECT kitapGuncelle( 1,300,'2023-01-01', 1, 1 );
```

TRIGGERS :

1 - defaultyayinTarihi :

```
CREATE OR REPLACE FUNCTION defaultyayinTarihi ()  
RETURNS TRIGGER AS $$  
BEGIN
```

```
  IF NEW."yayinTarihi" IS NULL THEN  
    NEW."yayinTarihi" := CURRENT_DATE;  
  END IF;
```

```
  RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER defaultyayinTarihi_trigger  
BEFORE INSERT ON kitap.kitap  
FOR EACH ROW  
EXECUTE FUNCTION defaultyayinTarihi ();
```

```
INSERT INTO kitap.kitap ("sayfaSayisi", "yazar_id", "yayinevi_id")  
VALUES (300, 1, 1);
```

2 - yeniPuan :

```
CREATE OR REPLACE FUNCTION yeniPuan()  
RETURNS TRIGGER AS $$  
BEGIN  
    IF NEW.puan IS NULL THEN  
        NEW.puan := 10;  
    ELSE  
        IF NEW.puan > 5 THEN  
            NEW.puan := 10;  
        ELSE  
            NEW.puan := 1;  
        END IF;  
    END IF;  
    RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER yeniPuan_trigger  
BEFORE INSERT ON "kitapDegerlendirmesi"  
FOR EACH ROW  
EXECUTE FUNCTION yeniPuan();
```

```
INSERT INTO "kitapDegerlendirmesi" ("puan", "kitap_id")  
VALUES (1, 1);
```

3 - defaultCeviri :

```
CREATE OR REPLACE FUNCTION defaultCeviri()  
RETURNS TRIGGER AS $$  
BEGIN  
    IF NEW.dil IS NULL THEN  
        NEW.dil := 'turkce';  
    END IF;  
    RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER defaultCeviri_trigger  
BEFORE INSERT ON "kitapCevirileri"  
FOR EACH ROW  
EXECUTE FUNCTION defaultCeviri();
```

```
INSERT INTO "kitapCevirileri" ("kitap_id")  
VALUES (1);
```

4 - soyadDegistir :

```
CREATE OR REPLACE FUNCTION soyadDegistir()  
RETURNS TRIGGER AS $$  
BEGIN  
    IF NEW.ad = 'mustafa' THEN  
        NEW.soyad := 'kilic';  
    END IF;  
    RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER soyadDegistir_trigger  
BEFORE INSERT ON "yazar"  
FOR EACH ROW  
EXECUTE FUNCTION soyadDegistir();
```

```
INSERT INTO "yazar" ("ad", "soyad", "ulke")  
VALUES ('mustafa', 'ahmed', 'turkey');
```

UYGULAMA :

Form1

yazar	SAVE
yayincilar	SAVE
kitap	SAVE
manga	SAVE
felsefe	SAVE
macera	SAVE
polisiye	SAVE

ARAMA :

Form1

	yazar_id	ad	soyad	ulke
*				

yazar	SAVE
yayincilar	SAVE
kitap	SAVE
manga	SAVE
felsefe	SAVE
macera	SAVE
polisiye	SAVE

EKLEME :

Form1

	yazar_id	ad	soyad	ulke
▶	1	Yazan	Alsolyman	Suriye
*				

Buttons:

- yazar SAVE
- yayincilar SAVE
- kitap SAVE
- manga SAVE
- felsefe SAVE
- macera SAVE
- polisiye SAVE

Changes saved successfully!

OK

SİLME :

Form1

	yazar_id	ad	soyad	ulke
✍				
*				

Buttons:

- yazar SAVE
- yayincilar SAVE
- kitap SAVE
- manga SAVE
- felsefe SAVE
- macera SAVE
- polisiye SAVE

GÜNCELLEME :

Form1

	yazar_id	ad	soyad	ulke
▶	1	Ahmed	Fouad	Mısır
*				

yazar SAVE

yayincilar SAVE

kitap SAVE

manga SAVE

felsefe SAVE

macera SAVE

polisiye SAVE

Changes saved successfully!

OK

Github Linki -> <https://github.com/4yznx/VTYS-Projesi-B211210572>