VERİTABANI TASARIMI

Öğretim Görevlisi A. Berika VAROL MALKOÇOĞLU

BEYKOZ ÜNİVERSİTESİ

İçindekiler

- Veri işleme dili (DDL)
- SQL ifadeleri
- E-R diyagram örnekleri
- Örnek veritabanı şeması

Veri Tanımlama Dili (DDL)

- Veri Tanımlama Dili yani (Data Definition Language – DDL) veritabanı veya tablo oluşturmayı, üzerinde değişiklik yapmayı veya silmeyi sağlayan ifadelerdir.
  - Create
  - Alter
  - Drop
  - Truncate
  - Comment
  - Rename

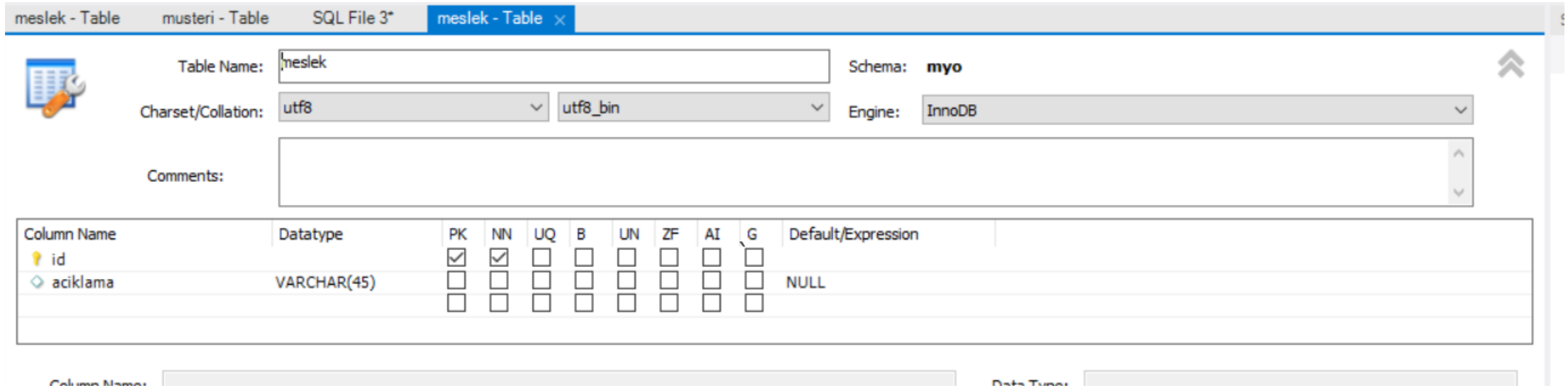
Create

- Veri tabanındaki nesnelerin oluşturulabilmesi için kullanılır.
- Bu komutu kullanabilmek için kullanıcının Sistem Yöneticisi veya Veritabanı kurucusu rollerinden birine sahip olması gerekir.

Syntax:

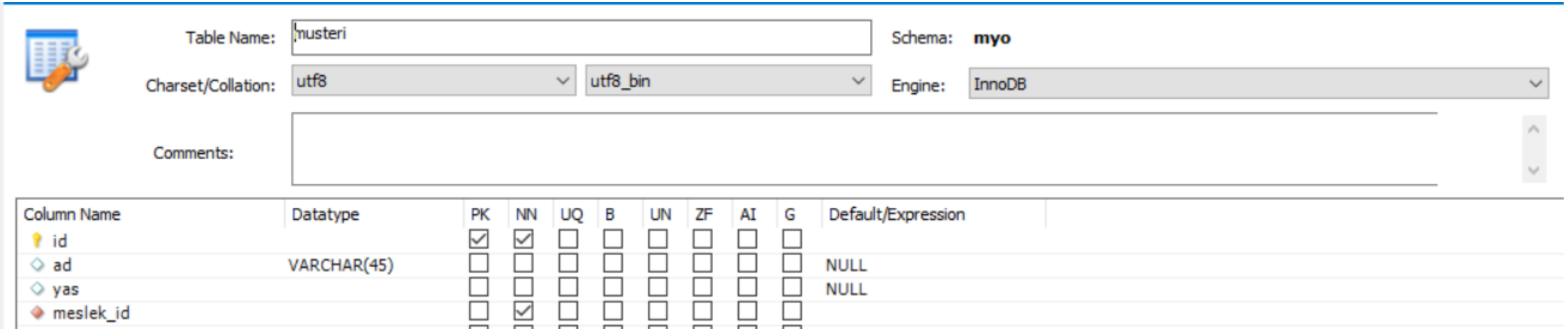
```
CREATE TABLE table_name (
column_name1 data_type(size),
column_name2 data_type(size),
……
PRIMARY KEY (column_name1));
```

Create




```
1  CREATE TABLE `myo`.`meslek` (
2    `id` INT NOT NULL,
3    `aciklama` VARCHAR(45) NULL,
4    PRIMARY KEY (`id`));
5
```

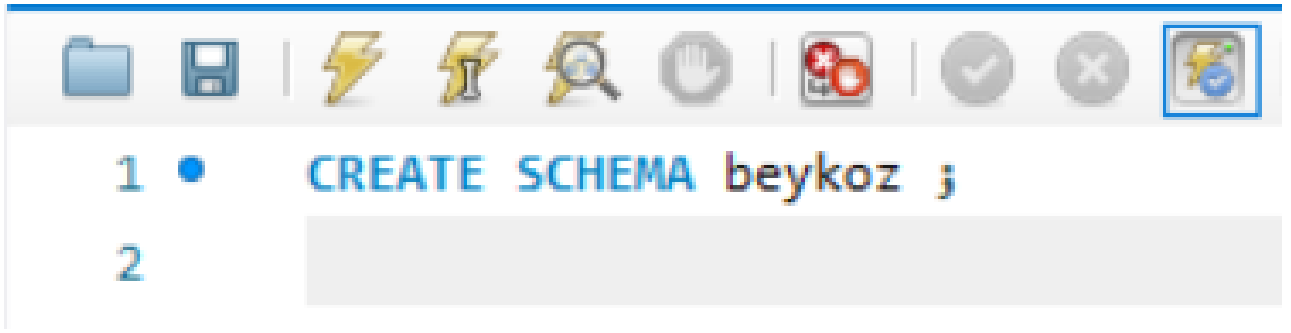
Create

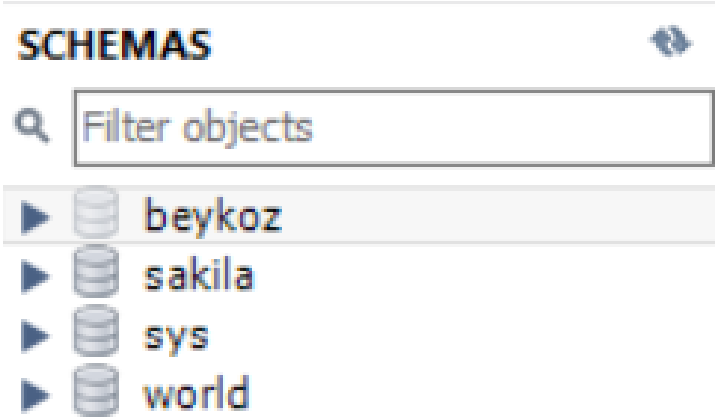



```
1  CREATE TABLE `myo`.`musteri` (
2    `id` INT NOT NULL,
3    `ad` VARCHAR(45) NULL,
4    `yas` INT NULL,
5    `meslek_id` VARCHAR(45) NULL,
6    PRIMARY KEY (`id`));
7
```

BEYKOZ ÜNİVERSİTESİ

Create

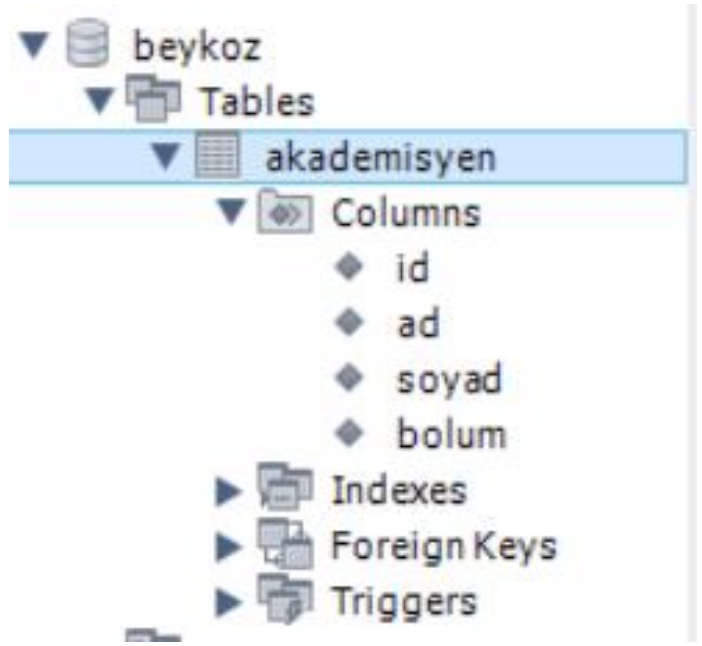




20	20:19:58	CREATE SCHEMA beykoz	1 row(s) affected

Create

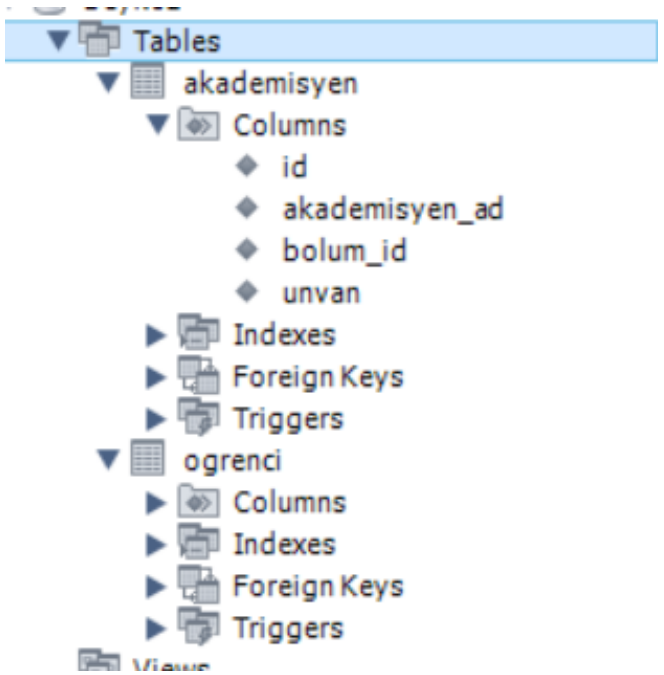
```
CREATE TABLE beykoz.akademisyen (
  id INT NOT NULL,
  ad VARCHAR(45) NULL,
  soyad VARCHAR(45) NULL,
  bolum INT NULL,
  PRIMARY KEY (id));
```




```
27  20:22:40  CREATE TABLE beykoz.akademisyen (  id INT N...  0 row(s) affected
```

Create

```
CREATE TABLE beykoz.ogrenci (
  id INT NOT NULL,
  ad VARCHAR(45) NULL,
  soyad VARCHAR(45) NULL,
  bolum INT NULL,
  PRIMARY KEY (id));
```



35 20:37:13 CREATE TABLE beykoz.ogrenci (id INT ... 0 row(s) affected

Alter

- Daha önceden oluşturulmuş bir nesne özelliğinin değiştirilmesini sağlar.
  - Yeni bir sütun ekleme,
  - Sütun tanımı ve sütun değerinin değiştirilmesi,
  - Var olan bir sütunun silinmesi,
  - Tablo tanımının değiştirilmesi,
  - Tablo kısıtlarının değiştirilmesi, yeni kısıtların eklenmesi ya da var olan kısıtların düşürülmesini yapar.

Alter

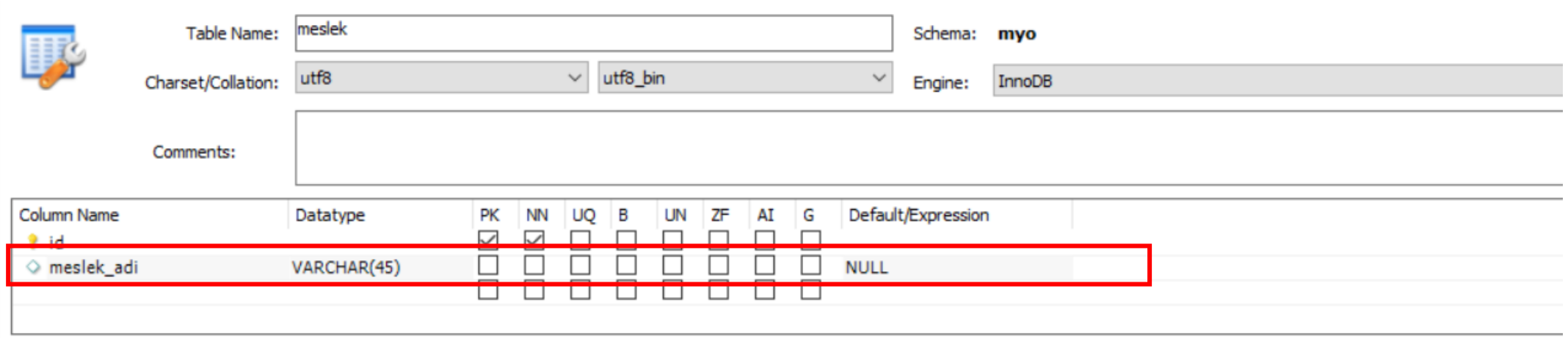
Syntax:

ALTER TABLE table_name **ADD COLUMN** column_name1 data_type(size);

ALTER TABLE table_name **DROP COLUMN** column_name1;

ALTER TABLE table_name **CHANGE COLUMN** old_column_name new_column_name data_type(size);

Alter

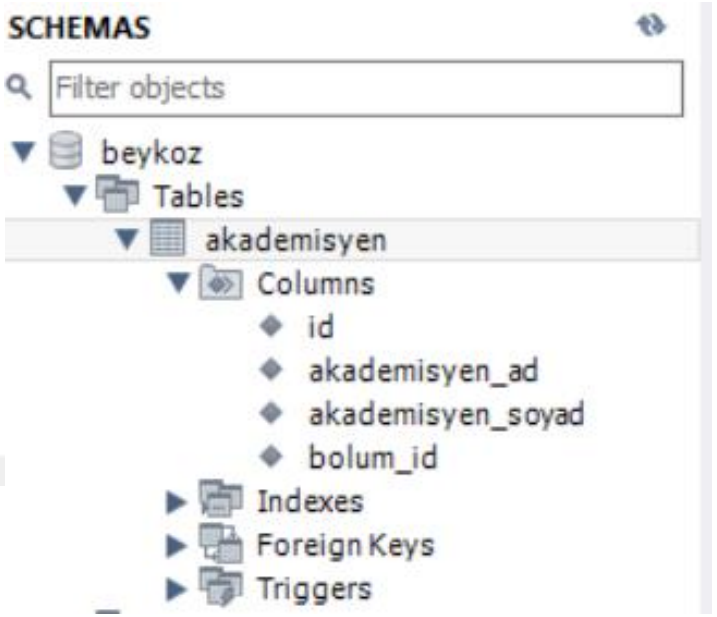



```
1   ALTER TABLE `myo`.`meslek`
2   CHANGE COLUMN `aciklama` `meslek_adi` VARCHAR(45) NULL DEFAULT NULL ;
3
```

Alter

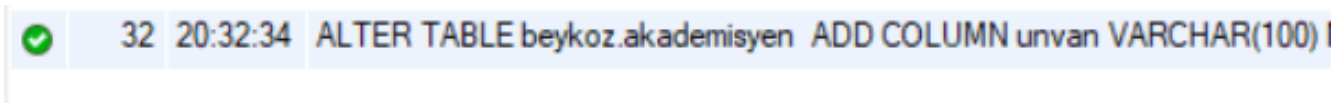
```
ALTER TABLE beykoz.akademisyen
CHANGE COLUMN ad akademisyen_ad VARCHAR(45) NULL DEFAULT NULL ,
CHANGE COLUMN soyad akademisyen_soyad VARCHAR(45) NULL DEFAULT NULL ,
CHANGE COLUMN bolum bolum_id  INT DEFAULT NULL ;
```

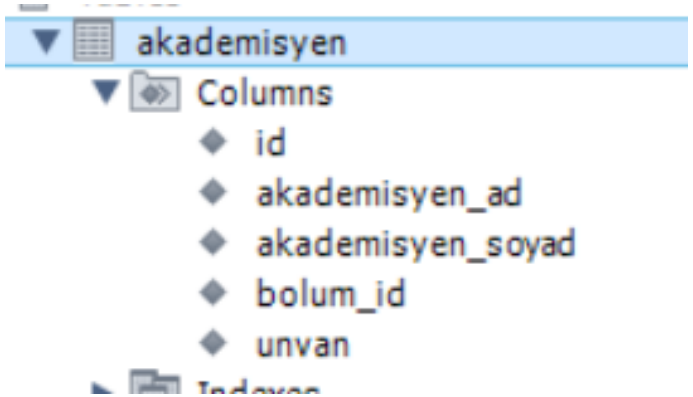
```
29  20:27:34  ALTER TABLE `beykoz`.`akademisyen` CHANGE ...  0 row(s) affected Reco
```



Alter

```
ALTER TABLE beykoz.akademisyen
ADD COLUMN unvan VARCHAR(100) NULL AFTER bolum_id;
```

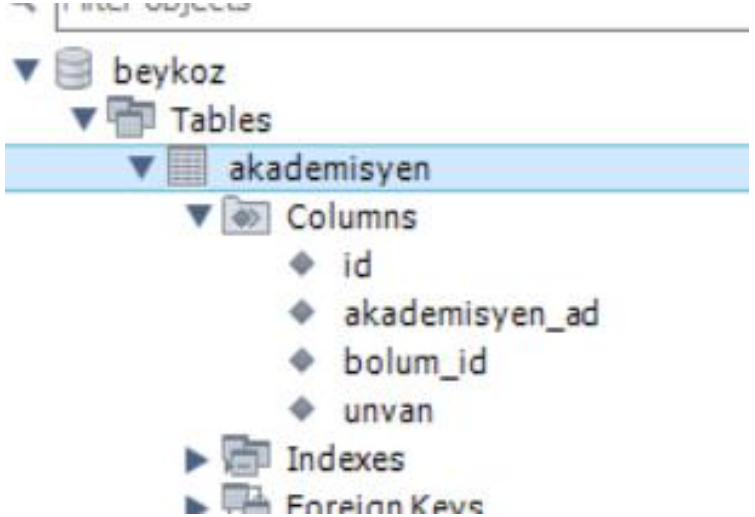




Alter

```
ALTER TABLE beykoz.akademisyen
DROP COLUMN akademisyen_soyad;
```

34	20:34:38	ALTER TABLE beykoz.akademisyen DROP ...	0 row(s) affected Records: 0



Drop

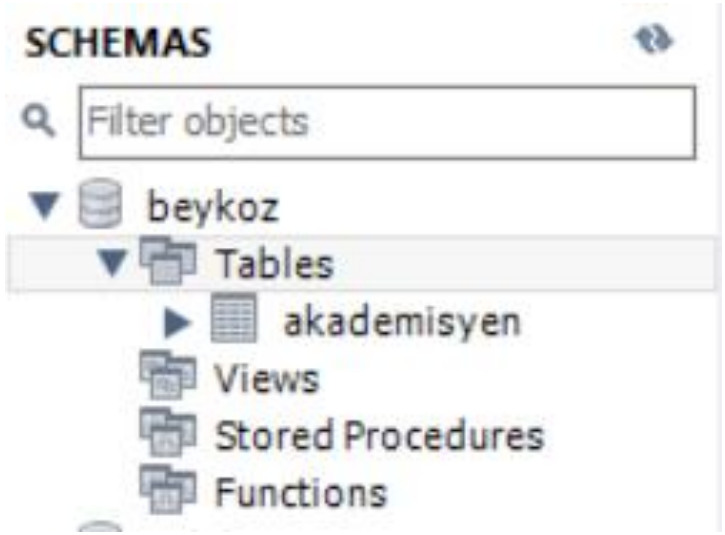
- Bir nesnenin silinmesini sağlayan komuttur. DROP komutu tüm nesneler için kullanılır.
  - Tablo tanımı ve verilerin silinmesi
  - Tablo ile ilgili indexler, kısıtlar, tetikçiler ve yetkilerin kaldırılmasını yapar.

Syntax:

```
DROP DATABASE database_name;
DROP TABLE table_name;
ALTER TABLE table_name DROP INDEX index_name ;
```

Drop

```
DROP TABLE beykoz.ogrenci;
```



✓	36	20:40:37	DROP TABLE beykoz.ogrenci	0 row(s) affected

Truncate

- Bir tablodaki tüm kayıtları, kayıtlar için ayrılan tüm boşluklar da dahil olmak üzere kaldırmak için kullanılır.

Syntax:

```
TRUNCATE TABLE table_name;
```

Truncate

Result Grid | Filter Rows: | Edit:

id	akademisyen_ad	bolum_id	unvan
1	ayşe	bilgisayar	doçent
2	ali	inşaat	öğretim görevlisi
NULL	NULL	NULL	

Result Grid | Filter Rows: | Edit:

id	akademisyen_ad	bolum_id	unvan
NULL	NULL	NULL	NULL

```
TRUNCATE TABLE beykoz.akademisyen;
```

✓	38	20:43:34	TRUNCATE TABLE beykoz.akademisyen	0 row(s) affected

BEYKOZ ÜNİVERSİTESİ

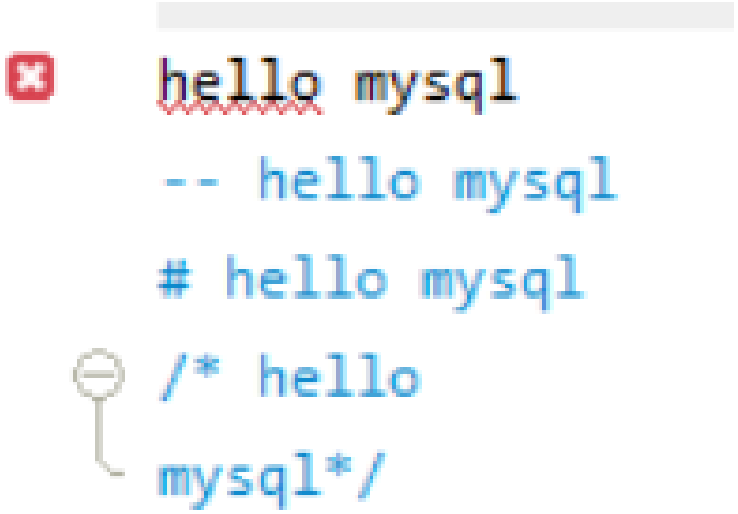
Comment

- Yorum eklemek için kullanılır.

Syntax:

```
-- tek satır yorumlar için
# tek satır yorumlar için
/* çok satırlı
yorumlar için*/
```

Comment



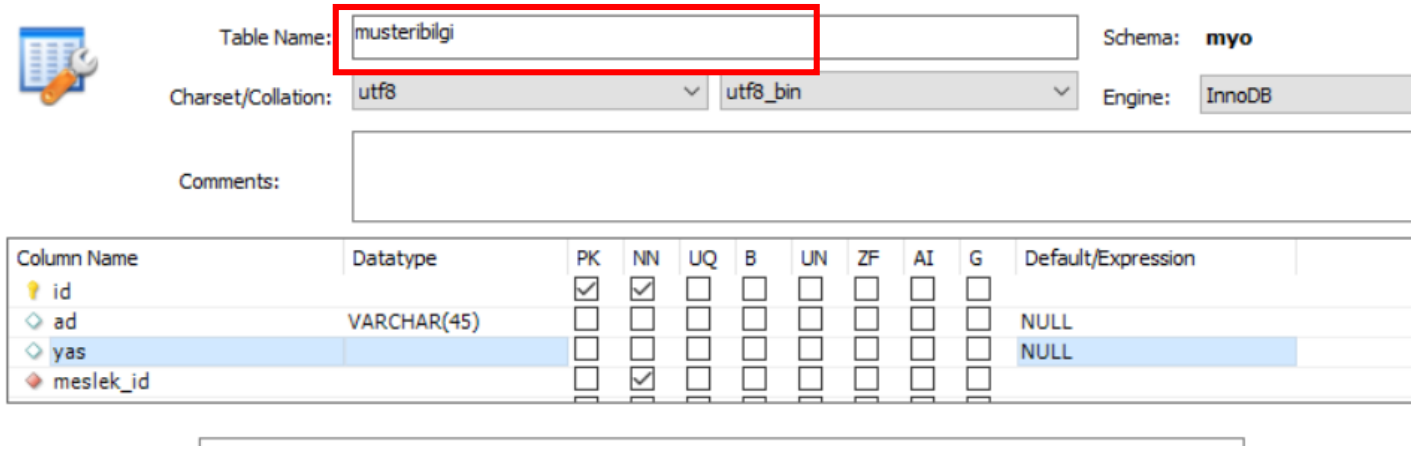
Rename

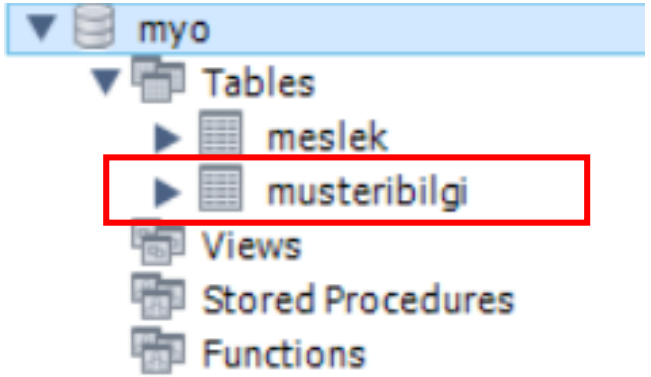
- Veritabanında bulunan bir nesneyi yeniden adlandırmak için kullanılır.

Syntax:

```
RENAME TABLE old_table_name TO new_table_name;
```

Rename



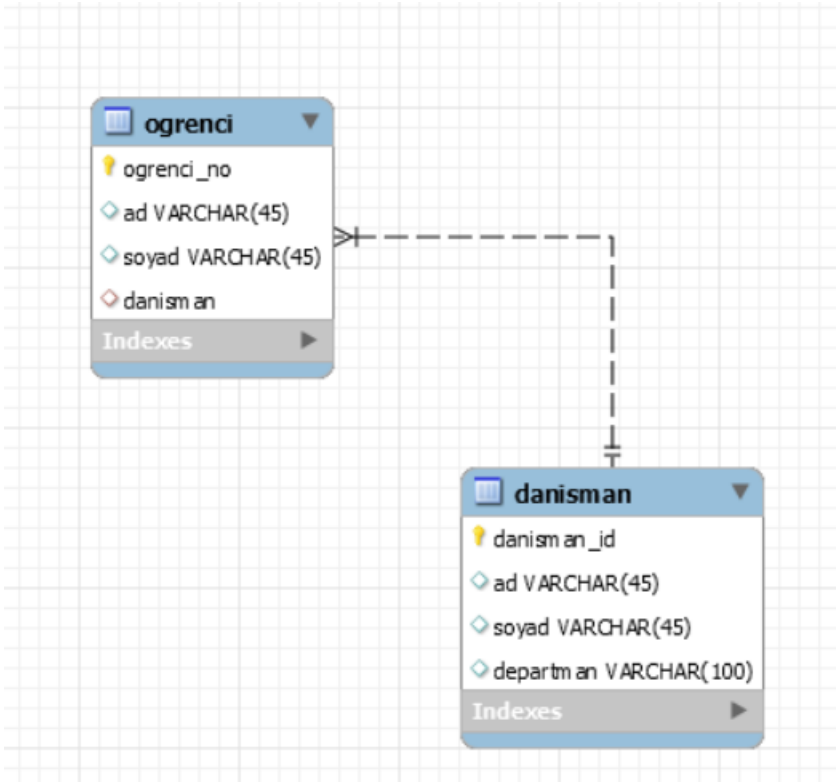



```
RENAME TABLE myo.musteri TO myo.musteriBilgi;
```

Tabloları İlişkilendirmek

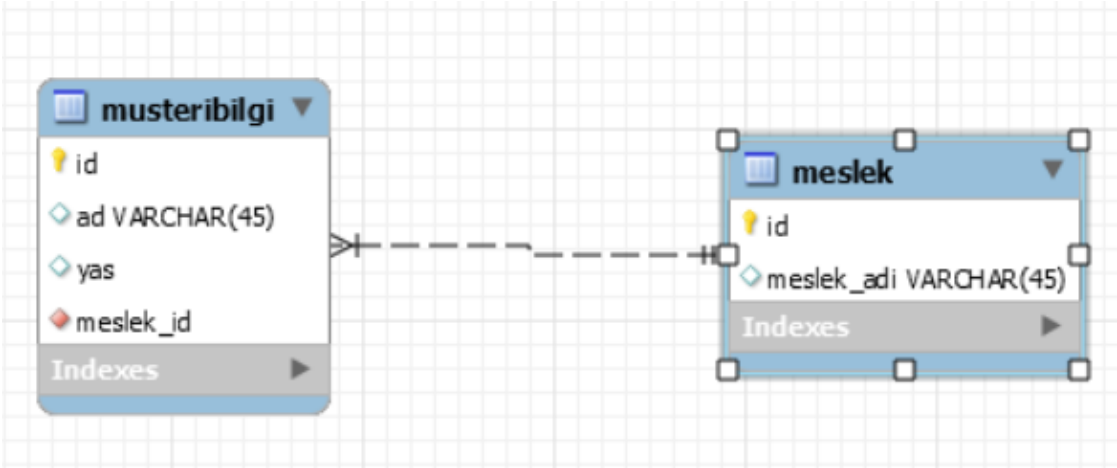
- Mevcut tablolar ‘foreign key’ olarak adlandırdığımız anahtar yardımı ile ilişkilendirerek tam bir veritabanı oluştururuz.
- E-R diyagramlarında gösterildiği gibi 1-1, 1-N veya N-M şeklinde ilişkilendirilen tablolar üzerinde SQL ile sorgulama yapabiliriz.

Tabloları İlişkilendirmek



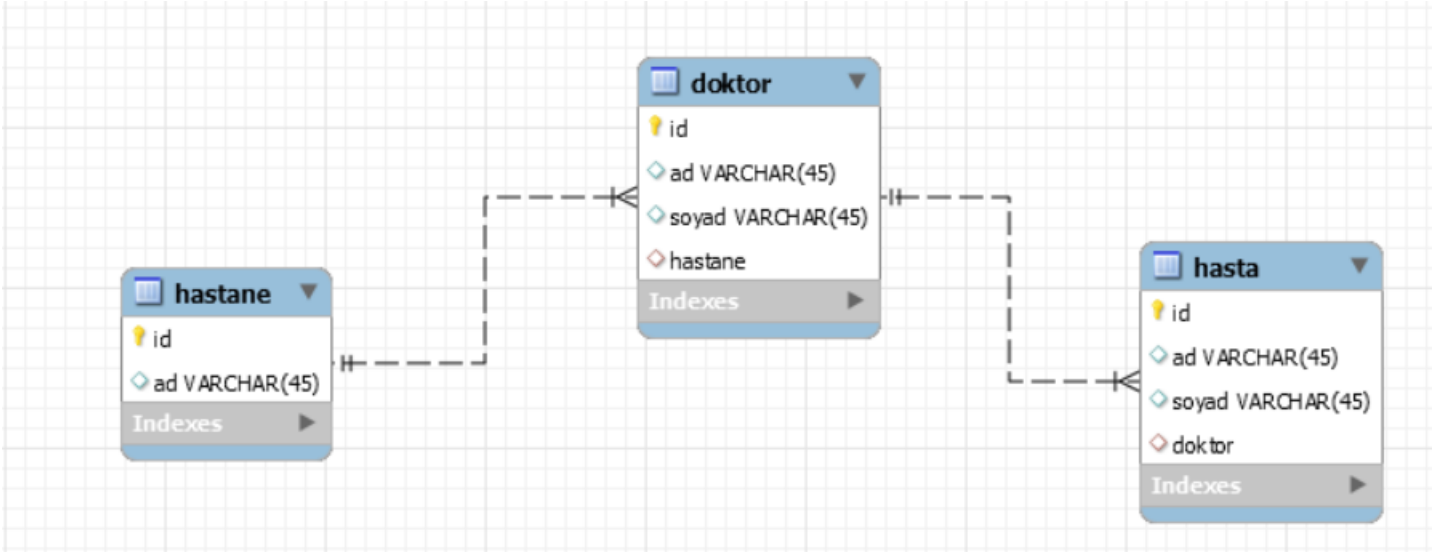

```
ALTER TABLE okul.ogrenci ADD CONSTRAINT fk_key FOREIGN KEY (danisman) REFERENCES danisman(danisman_id);
```

Tabloları İlişkilendirmek




```
ALTER TABLE myo.musteriBilgi ADD CONSTRAINT fk_meslek_id FOREIGN KEY (meslek_id) REFERENCES meslek(id);
```

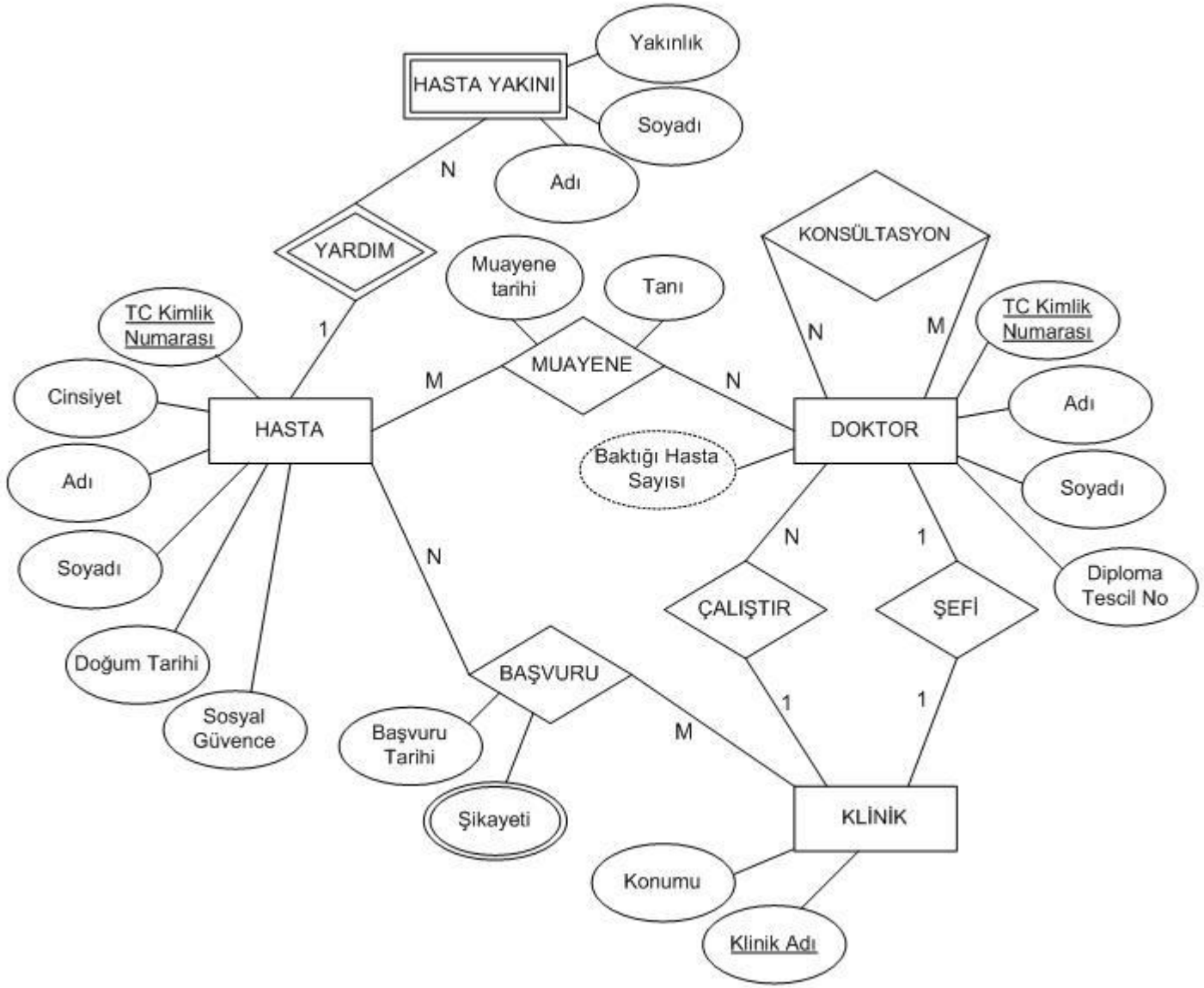
Tabloları İlişkilendirmek

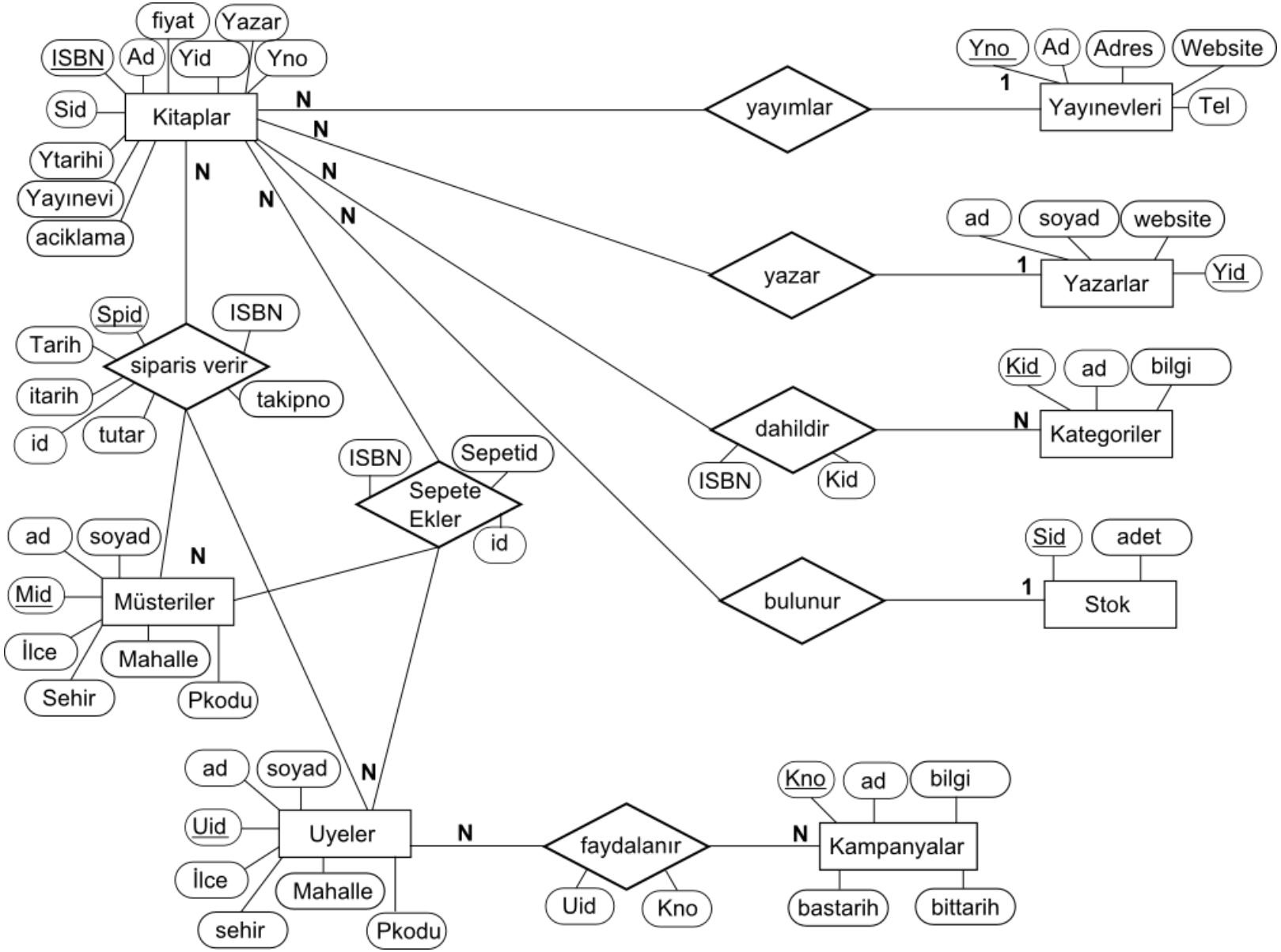



```
ALTER TABLE saglik.hasta ADD CONSTRAINT fk_key_d FOREIGN KEY (doktor) REFERENCES doktor(id);

ALTER TABLE saglik.doktor ADD CONSTRAINT fk_key_h FOREIGN KEY (hastane) REFERENCES hastane(id);
```

BEYKOZ ÜNİVERSİTESİ


HASTA YAKINI
Yakınlık
Soyadı
Adı
N
YARDIM
1
TC Kimlik Numarası
Cinsiyet
Adı
Soyadı
Doğum Tarihi
Sosyal Güvence
HASTA
Muayene tarihi
Tanı
MUAYENE
M
N
KONSÜLTASYON
N
M
TC Kimlik Numarası
Adı
Soyadı
Diploma Tescil No
DOKTOR
Baktığı Hasta Sayısı
N
1
ÇALIŞTIR
ŞEFİ
1
1
N
BAŞVURU
M
Başvuru Tarihi
Şikayeti
KLİNİK
Konumu
Klinik Adı


fiyat
Yazar
ISBN
Ad
Yid
Yno
Sid
Kitaplar
Ytarihi
Yayınevi
aciklama
N
yayımlar
Yno
Ad
Adres
Website
1
Yayınevleri
Tel
yazar
ad
soyad
website
Yazarlar
Yid
Spid
ISBN
Tarih
siparis verir
takipno
itarih
id
tutar
dahildir
Kid
ad
bilgi
Kategoriler
ISBN
Kid
Sepetid
Sepete Ekler
id
ad
soyad
Mid
Müsteriler
İlce
Mahalle
Sehir
Pkodu
bulunur
Sid
adet
Stok
ad
soyad
Uid
Uyeler
İlce
Mahalle
sehir
Pkodu
faydalanır
Uid
Kno
Kno
ad
bilgi
Kampanyalar
bastarih
bittarih

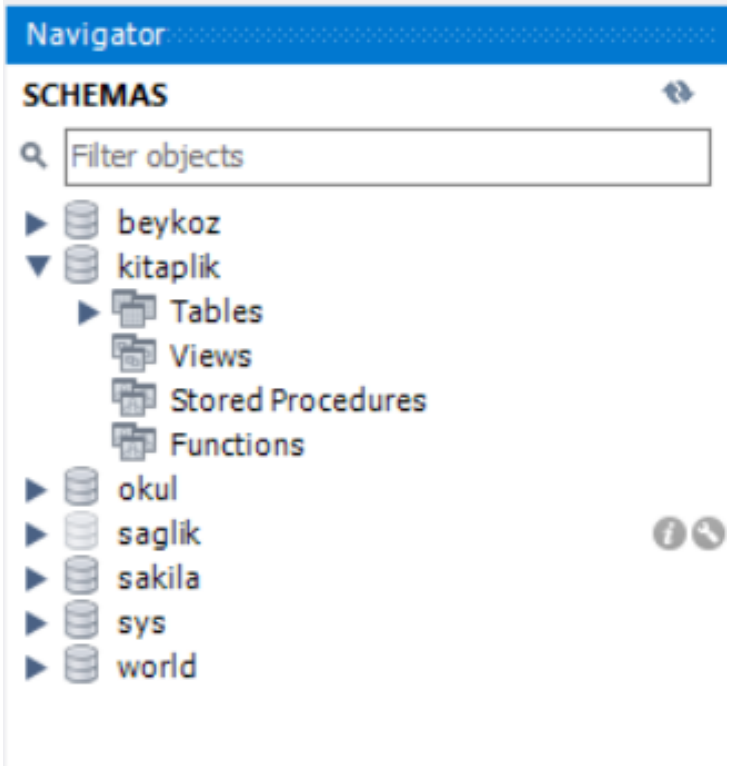
DDL ÖRNEKLERİ

Örnek Veritabanı

- Şema adı: ‘kitaplik’
  - Tablolar:
    - ‘kitap’
      - kitap_id (Primary Key)
      - kitap_ad
      - yazar_id (Foreign Key)
      - kategori_id (Foreign Key)
    - ‘yazar’
      - yazar_id (Primary Key)
      - yazar_adi
    - ‘kategori’
      - kategori_id (Primary Key)
      - kategori_ad

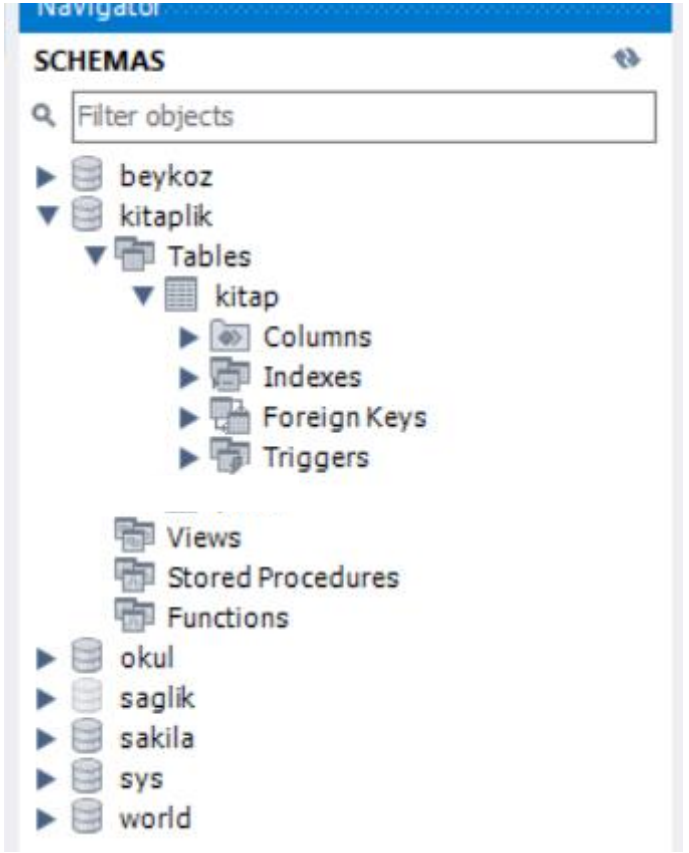
CREATE

```
CREATE SCHEMA kitaplik ;
```



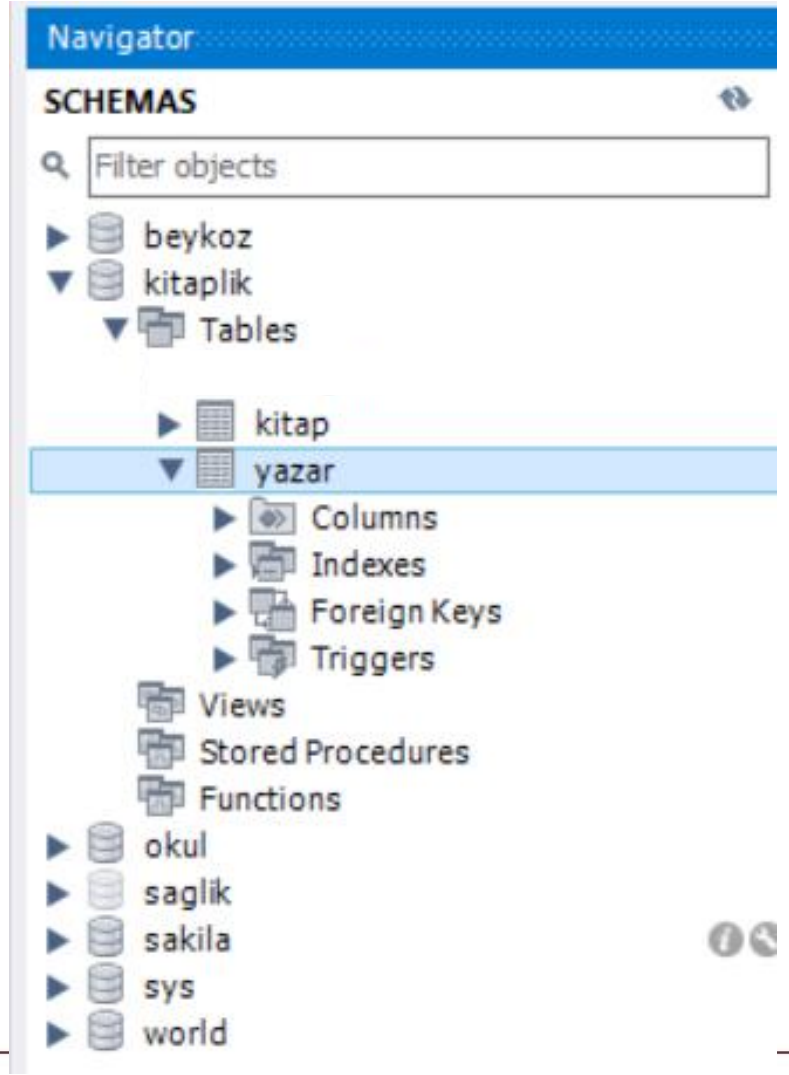
CREATE

```
CREATE TABLE kitaplik.kitap (
   kitap_id INT NOT NULL,
   kitap_ad VARCHAR(50) NULL,
   yazar_id INT NOT NULL,
   kategori_id INT NOT NULL,
   PRIMARY KEY (kitap_id));
```



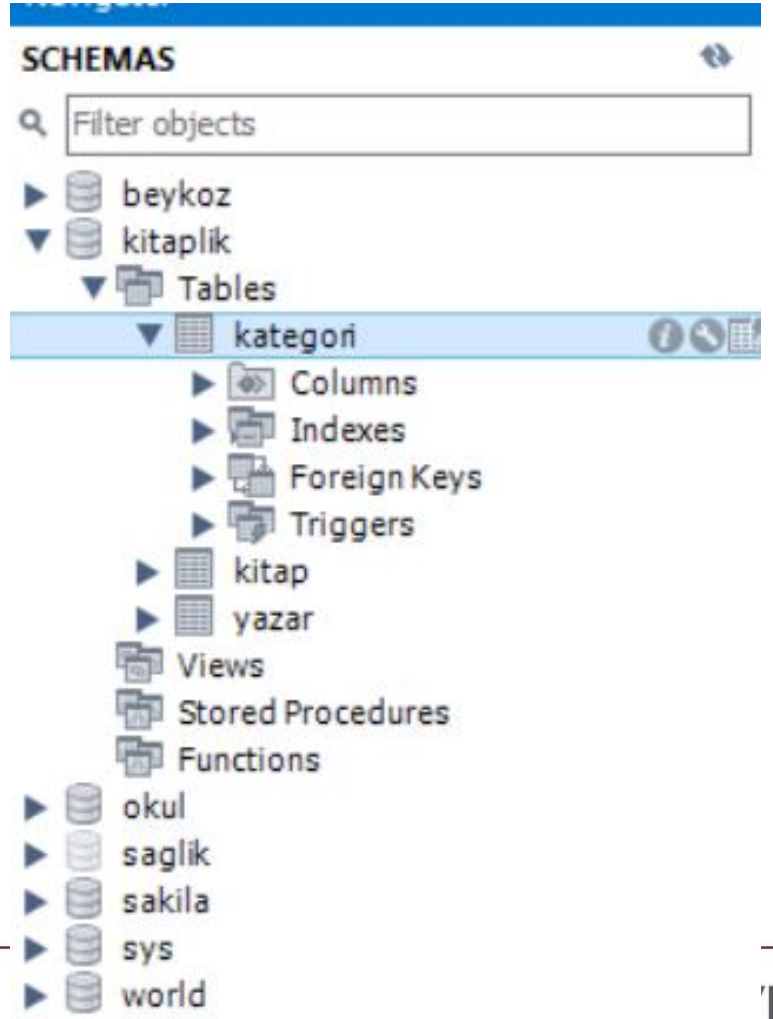
CREATE

```
CREATE TABLE kitaplik.yazar (
  yazar_id INT NOT NULL,
  yazar_ad VARCHAR(100) NULL,
  PRIMARY KEY (yazar_id));
```

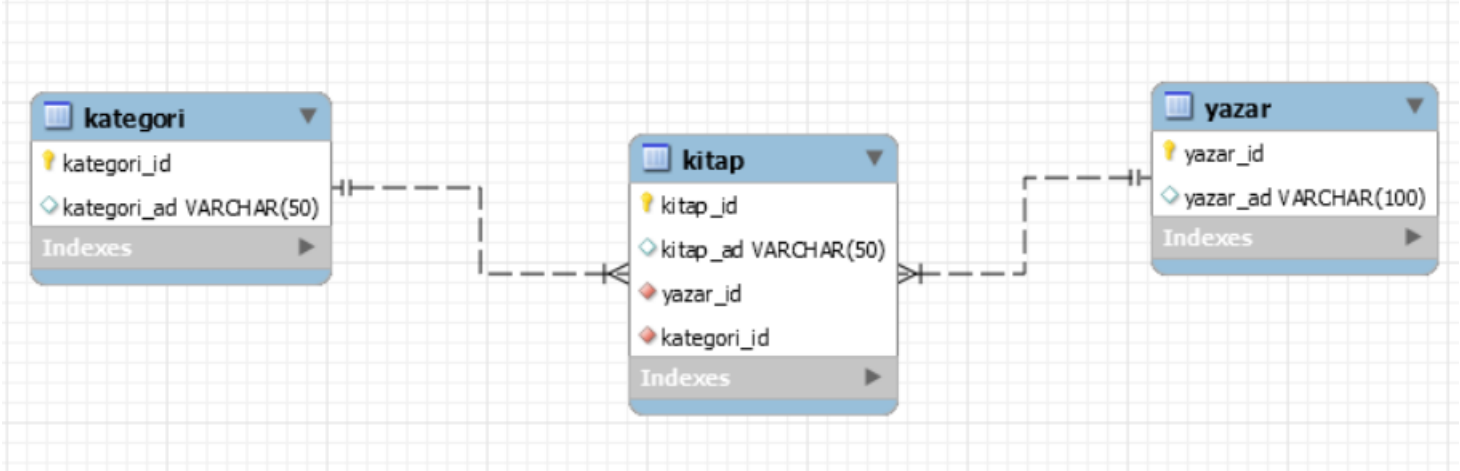


CREATE

```
6
7   CREATE TABLE kitaplik.kategori (
8   kategori_id INT NOT NULL,
9   kategori_ad VARCHAR(50) NULL,
0   PRIMARY KEY (kategori_id));
1
```



Yabancı Anahtar ve E-R Şeması




```
21
22 •  ALTER TABLE kitaplik.kitap ADD CONSTRAINT fk_key_yazar FOREIGN KEY (yazar_id) REFERENCES yazar(yazar_id);
23
24 •  ALTER TABLE kitaplik.kitap ADD CONSTRAINT fk_key_kategori FOREIGN KEY (kategori_id) REFERENCES kategori(kategori_id);
25
```

BEYKOZ ÜNİVERSİTESİ