**T.C.**

**ESKİŞEHİR OSMANGAZİ ÜNİVERSİTESİ**

**MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**DERSİN ADI:**

VERİ TABANI YÖNETİM

SİSTEMLERİ LABORATUVARI

**DENEY #5**

**Doç. Ahmet YAZICI**

Gizem Taşcı - 152120131100

**ESKİŞEHİR**

ÇÖZÜMLER

1. CREATE DATABASE db\_TheatreManage;

* “CREATE DATABASE” komutu kullanılarak veri tabanımızda db\_TheatreManage adında bir database oluşturulmuştur.

CREATE TABLE tbl\_Company

(

Name varchar(20) ,

Country varchar(20),

PRIMARY KEY(Name)

)

CREATE TABLE tbl\_Production

(

Name varchar(20),

Company\_Name varchar(20) NOT NULL,

Director varchar(20),

PRIMARY KEY(Name,Director),

FOREIGN KEY(Company\_Name) REFERENCES tbl\_Company(Name)

)

CREATE TABLE tbl\_Diary

(

Diary\_Date date,

Seats int,

Price int,

Director varchar(20),

Production\_Name varchar(20),

PRIMARY KEY(Diary\_Date),

FOREIGN KEY(Production\_Name,Director) REFERENCEStbl\_Production(Name,Director)

)

* Tablolar oluşturulurken ER diyagramının bize göstermiş olduğu ilişkiler göz önünde bulundurulmuştur.
* Diary-Production İlişkisi : Many-to-One ilişkisi bulunmaktadır. Bu yüzden tablolar oluşturulurken “Many” olan Diary tablosu üzerinde, Production tablosunun PRIMARY KEY’leri, FOREIGN KEY kullanılarak referans edilmiştir.
* Production-Company İlişkisi : Many-To-One ilişkisi bulunmaktadır.
* Bu yüzden tablolar oluşturulurken Production tablosu üzerinde, Company tablosunun PRIMARY KEY’i, FOREIGN KEY kullanılarak referans edilmiştir.
* Aynı zamanda burada bulunan kalın çizgi bütün yapıtlar mutlaka bir şirkete ait olmalıdır anlamına gelmektedir. Bu yüzden hiçbir yapıtın Production tablosunda tanımladığımız Company\_Name değeri boş kalmasına izin verilmemelidir. Bu kısıt NOT NULL tanımlanarak oluşturulmuştur.

INSERT INTO tbl\_Company(Name,Country) VALUES ('A','Ankara')

INSERT INTO tbl\_Company(Name,Country) VALUES ('B','Eskişehir')

INSERT INTO tbl\_Company(Name,Country) VALUES ('C','Eskişehir')

INSERT INTO tbl\_Company(Name,Country) VALUES ('D','İzmir')

INSERT INTO tbl\_Company(Name,Country) VALUES ('E','İstanbul')

INSERT INTO tbl\_Production(Name,Company\_Name,Director) VALUES ('PA1','A','DA1')

INSERT INTO tbl\_Production(Name,Company\_Name,Director) VALUES ('PA2','A','DA2')

INSERT INTO tbl\_Production(Name,Company\_Name,Director) VALUES ('PB1','B','DB1')

INSERT INTO tbl\_Production(Name,Company\_Name,Director) VALUES ('PB2','B','DB2')

INSERT INTO tbl\_Production(Name,Company\_Name,Director) VALUES ('PC','C','DC')

INSERT INTO tbl\_Diary(Diary\_Date,Seats,Price,Production\_Name,Director) VALUES ('2015-11-23',10,10,'PA1','DA1')

INSERT INTO tbl\_Diary(Diary\_Date,Seats,Price,Production\_Name,Director) VALUES ('2015-11-22',20,20,'PA2','DA2')

INSERT INTO tbl\_Diary(Diary\_Date,Seats,Price,Production\_Name,Director) VALUES ('2015-11-24',30,30,'PB1','DB1')

INSERT INTO tbl\_Diary(Diary\_Date,Seats,Price,Production\_Name,Director) VALUES ('2015-11-26',40,40,'PB2','DB2')

INSERT INTO tbl\_Diary(Diary\_Date,Seats,Price,Production\_Name,Director) VALUES ('2015-11-25',50,50,'PC','DC')

* Her bir tabloya ayrı ayrı 5 adet veri, “INSERT INTO” komutu kullanılarak girilmiştir.
* Burada en önemli nokta, birbirinden referans aldığımız verilerin aynı olmasına dikkat edilmesidir.

1. SELECT MAX(Price) FROM tbl\_Diary WHERE Production\_Name IN

SELECT Name FROM tbl\_Production WHERE Company\_Name ='A')

* İsmi “A” olan şirketin, yapmış olduğu yapıtlar arasından en yüksek ücrete sahip olanın listelenmesi amaçlanmıştır.
* İç sorguda , şirketin adı Production tablosunda aranıp, bu şirketin yapmış olduğu yapıtları listeler.
* Dış sorguda ise IN komutu kullanılarak bu oluşturulan yeni tablo içerisindeki verilerden MAX komutunu kullanarak en yüksek ücrete sahip olanı getirir.

1. SELECT COUNT(\*) FROM tbl\_Production WHERE Company\_Name = 'B';

* İsmi “B” olan şirketin yaptığı yapıtların sayısını bulmak amaçlanmıştır.
* COUNT(\*) komutu kullanılarak , oluşturulan tablonun tüm satır sayısını getirilir.

1. SELECT Diary\_Date FROM tbl\_Diary WHERE Production\_Name IN

(SELECT Name FROM tbl\_Production WHERE Company\_Name='A' OR Company\_Name='B')

* İsmi “A” ya da “B” olan şirketlerin, yapmış oldukları yapıtların tarihlerini listelemek amaçlanmıştır.
* A ya da B sorgusu OR komutu kullanılarak iç sorguda sağlanarak, bu şirketlerin yaptıkları yapıtların adı tablo haline getirilmiştir.
* Dış sorguda ise IN komutu ile bu tablo içerisindeki veriler baz alınarak, Diary tablosundan tarih bilgileri listelenmiştir.

1. SELECT \* FROM tbl\_Diary ORDER BY Seats

* ORDER BY komutu kullanılarak , yapıtların tüm bilgileri(\*) koltuk sayılarına göre küçükten büyüğe (ASCENDING) sıralanmıştır.

1. SELECT Diary\_Date,Production\_Name FROM tbl\_Diary WHERE Diary\_Date BETWEEN '2015-11-23' AND '2015-11-29'

* 23/11/2015 ile 29/11/2015 tarihleri arasında bulunan yapıtların isimleri ve tarihleri listelenmek amaçlanmıştır.
* BETWEEN … AND komutu kullanılarak Diary tablosundan alınan tarihler ve bu yapıtların adları listelenmiştir.

1. SELECT Director FROM tbl\_Production WHERE Company\_Name='A'

UNION

SELECT Director FROM tbl\_Production WHERE Company\_Name='B'

* İsmi “A” ya da “B” olan şirketlerin yöneticilerinin adları tablo haline getirilmek amaçlanmıştır.
* UNION komutu kullanılarak ilk sorguda şirket adı A olanlar, ikici sorguda şirket adı B olanlar bulunmuş ve bunların birleşimi alınmıştır.

1. SELECT Name FROM tbl\_Production Where Company\_Name='B'

AND Company\_Name != 'C'

* İsmi “B” olan şirketin yaptığı ama aynısını ismi “C” olan şirketin yapmamış olduğu yapıtların adlarından tablo oluşturulması amaçlanmıştır.
* Burada “=!” karşılaştırma operatörü kullanılabilirken aynı zamanda NOT EXIST komutu kullanılarak da işlem iç içe iki sorguyla da sağlanabilirdi.

SELECT Name FROM tbl\_Production WHERE Company\_Name IN

(SELECT Name FROM tbl\_Company WHERE Country='Eskişehir')

* Eskişehir ‘de bulunan yapıtların tablo haline getirilmesi amaçlanmıştır.
* IN komutu kullanılarak iç sorguda oluşturulan Eskişehir’de bulunan şirketlerin adı tablosundan, Production tablosu yardımıyla yapıtların adları listelenmiştir.

SELECT AVG(Price), Production\_Name FROM tbl\_Diary, tbl\_Production

WHERE tbl\_Diary.Production\_Name=tbl\_Production.Name

GROUP BY Production\_Name

HAVING AVG(Price) > 10

* GROUP BY ve HAVING komutları sayesinde ortalama ücreti 10’dan büyük olma özelliğine sahip yapıtların adları gruplandırılmıştır.
* Bu gruplandırma içerisinden yapıtın ortalama ücreti ve adı Diary ile Production tablosu yardımıyla listelenmiştir.