**T.C.**

**ESKİŞEHİR OSMANGAZİ ÜNİVERSİTESİ**

**MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**DERSİN ADI:**

VERİ TABANI YÖNETİM

SİSTEMLERİ LABORATUVARI

**DENEY #1**

**Yrd. Doç. Ahmet YAZICI**

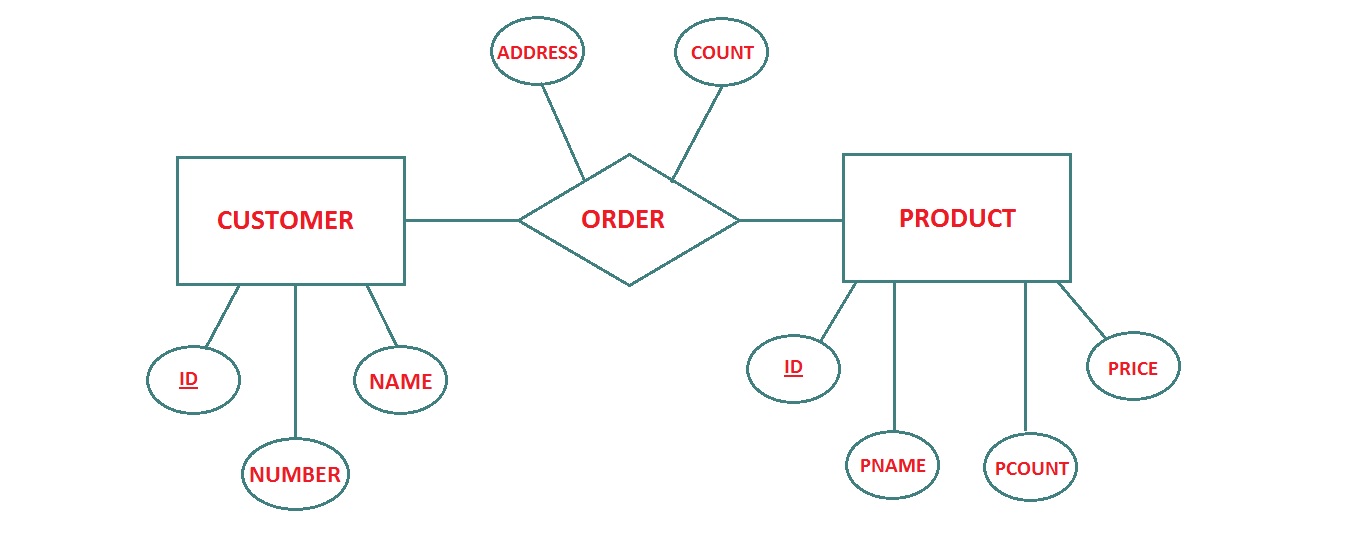
Gizem Taşcı - 152120131100

**ESKİŞEHİR**

**6.Ekim.2015**

**ÇÖZÜMLER**

1. ER DIAGRAM

****

ER (Entity-Relationship)diyagramı varlık-ilişki diyagramıdır. Oluşturulmuş bu şemada veritabanında bulunan tüm tablo ve ilişkiler belirtilir.

Yukarıda ouşturulmuş diyagramda sırasıyla;

* Varlıklar (Entity) : Customer, Product
* İlişkiler (Relationships) : Order
* Nitelikler (Attribute)
* Customer için : CustomerID, Name, IdentityNumber
* Product için : ProductID, ProductName, ProductCount, Price
* Order için : OrderAddress, OrderCount

1. CREATE DATABASE db\_Lab1;

“Create Database” komutu yardımıyla, SQL sorgu ekranında db\_lab1 adında bir database oluşturulmuştur.

1. CREATE TABLE tbl\_Customer(CustomerID int NOT NULL,

IdentitiyNumber varchar(11),

Name varchar(20),

PRIMARY KEY(CustomerID));

CREATE TABLE tbl\_Order (OrderID int NOT NULL,

OrderCount int,

OrderAddress varchar (20),

CustomerID int ,

ProductID int,

FOREIGN KEY (CustomerID) REFERENCES tbl\_Customer (CustomerID),

FOREIGN KEY (ProductID) REFERENCES tbl\_Product (ProductID));

CREATE TABLE tbl\_Product (ProductID int NOT NULL ,

ProductName varchar(100),

Price decimal (10,2),

ProductCount int ,

PRIMARY KEY (ProductID));

Database oluşturulduktan sonra sırasıyla;

* Fig. 1. tbl\_Customer kullanılarak, tbl\_Customer adında tablo oluşturulmuş,ER diagramında da altı çizili olarak belirttiğimiz CustomerID, Primary Key olarak belirtilmiştir. Böylece constraint yani bir nevi kısıtlama yapılarak,bu kolon içerisinde oluşan her bir kayıt için benzersiz bir değer atanmış olur.
* Fig. 2. tbl\_Order kullanılarak, tbl\_Order adında tablo oluşturulmuş, CustomerID ve aynı zamanda ProductID Foreign Key olarak belirtilmiştir. Bu sayede tabloya girilebilecek değerleri başka bir tablonun belli bir alanında yer alan veriler ile sınırlandırmaya ve ilişkilendirme işlemi yapıldı.
* Fig. 3. tbl\_Product kullanılarak, tbl\_Product adında bir tablo oluşturulmuş, aynen tbl\_Customer içinde yapılan Primary Key uygulaması burda da aynı amaçla ProductID için yapılmıştır.

1. USE db\_Lab1;

ALTER TABLE tbl\_Customer

ADD Surname varchar(50);

“Use” komutu ile oluşturulmuş database kullanılarak, değişim işlemi yapılması beklenilen tablo “Alter Table” komutu ile belirtilmiştir. “Add” komutu yardımıyla tbl\_Customer içerisine Surname kolonu eklenmiştir.

1. USE db\_Lab1;

ALTER TABLE tbl\_Order

ALTER COLUMN OrderAddress varchar (50);

“Use” komutu ile oluşturulmuş database kullanılarak, değişim işlemi yapılması beklenilen tablo “Alter Table” komutu ile belirtilmiştir.

“Alter” komutu kullanılarak bu sefer boyutu varchar(20) verilmiş olan OrderAddress boyutu varchar(50) olarak değiştirilmiştir.

1. USE db\_Lab1;

ALTER TABLE tbl\_Customer;

DROP COLUMN Surname;

“Use” komutu ile oluşturulmuş database kullanılarak, değişim işlemi yapılması beklenilen tablo “Alter Table” komutu ile belirtilmiştir.

“Drop” komutu kullanılarak istenilen tbl\_Customer tablosundan , daha önce sonradan eklenmiş olan “Surname” , databaseden silinme işlemine tabi tutulmuştur.

1. USE db\_Lab1;

DROP TABLE tbl\_Product;

“Use” komutu ile oluşturulmuş database kullanılarak, değişim işlemi yapılması beklenilen tablo “Alter Table” komutu ile belirtilmiştir. “Drop” komutu bir tablonun verilerini silebildiği gibi, aynı zamanda tüm tabloyu da veritabanından silebilir. Bu işlemde tbl\_Product kendi oluşturduğumuz db\_lab1 veritabanından silinmiştir.