

به نام خدا

گزارش پروژه‌ی نهایی درس طراحی پایگاه داده

فاز اول

چمران معینی

۹۹۳۱۰۵۳

آدرس را در یک خانه ذخیره نمی‌کنیم تا مقادیر همه‌ی سلول‌ها اتمیک باشد، پس جدول جداگانه‌ای برای آدرس درست می‌کنیم و برای هر آدرس یک آی‌دی می‌سازیم، و آی‌دی آدرس را به جای خود آدرس، در کنار دیگر داده‌های مشتریان و نمایندگی‌ها قرار می‌دهیم. بررسی می‌کنیم که همه‌ی داده‌ها به کلید اصلی وابسته باشند.

```
CREATE TABLE Address
(
    AddressID int NOT NULL PRIMARY KEY,
    Street    VARCHAR(50) ,
    City      VARCHAR(50) ,
    Province  VARCHAR(50)
);

CREATE TABLE Customer
(
    CustomerID int NOT NULL PRIMARY KEY,
    FirstName  VARCHAR(50) ,
    LastName   VARCHAR(50) ,
    NationalID VARCHAR(50) ,
    AddressID  int NOT NULL,
    PhoneNumber VARCHAR(50) ,
    FOREIGN KEY (AddressID) REFERENCES Address (AddressID)
);

CREATE TABLE Supplier
(
    SupplierID int NOT NULL PRIMARY KEY,
    ContractStart DATE,
    ContractEnd   DATE,
    ManagerName   VARCHAR(50) ,
    AddressID     int NOT NULL,
    FOREIGN KEY (AddressID) REFERENCES Address (AddressID)
);
```

برای هر یک از موجودیت‌های برند و مدل و ماشین، یک جدول جداگانه می‌سازیم تا در همه‌شان، تمام ستون‌ها وابسته به کلید اصلی باشند.

در هر یک از موجودیت‌های برند و مدل و ماشین، یک کلید خارجی هم قرار می‌دهیم تا برند هر مدل و مدل هر ماشین را به واسطه‌ی آن، همواره بتوانیم تشخیص دهیم.

```
CREATE TABLE Brand
(
  BrandID int NOT NULL PRIMARY KEY,
  Title VARCHAR(50)
);
```

```
CREATE TABLE Model
(
  ModelID int NOT NULL PRIMARY KEY,
  BrandID int NOT NULL,
  Title VARCHAR(50),
  FOREIGN KEY (BrandID) REFERENCES Brand (BrandID)
);
```

```
CREATE DOMAIN Transmission AS VARCHAR(50) CHECK (VALUE IN ('manual',
'automatic'));
CREATE DOMAIN Color AS VARCHAR(50) CHECK (VALUE IN ('blue', 'black', 'white',
'red'));
```

```
ALTER DOMAIN Color ADD CONSTRAINT ColorCheck CHECK (VALUE IN ('blue', 'black',
'white', 'red', 'green', 'yellow'));
```

```
CREATE TABLE Car
(
  VIN int NOT NULL PRIMARY KEY,
  Color Color,
  Transmission Transmission,
  ModelID int NOT NULL,
  ProductID int NOT NULL,
  FOREIGN KEY (ModelID) REFERENCES Model (ModelID),
  FOREIGN KEY (ProductID) REFERENCES Product (ProductID)
);
```

برای قطعات یدکی نیز جدول جدیدی می‌سازیم.

لازم است بدانیم که هر قطعه، برای چه مدل‌هایی به کار می‌آید. اگر بخواهیم این را نیز در جدول قطعات یدکی ذخیره کنیم، دیتای ما از نرمال بودن فاصله می‌گیرد، پس جدول جدیدی درست می‌کنیم که در آن کلید هر قطعه را در کنار کلید هر مدل قرار می‌دهیم تا به واسطه‌ی آن همه‌ی مدل‌هایی که یک قطعه‌ی یدکی برای آن مدل کارآمد است، دسترسی داشته باشیم و در عین حال داده‌هایمان را تا حد ممکن نرمال نگه داریم.

```
CREATE TABLE SparePart
(
    SparePartID int NOT NULL PRIMARY KEY,
    Title        VARCHAR(50) ,
    Price        int,
    Quantity     int,
    SupplierID   int NOT NULL,
    ProductID    int NOT NULL,
    FOREIGN KEY (SupplierID) REFERENCES Supplier (SupplierID) ,
    FOREIGN KEY (ProductID) REFERENCES Product (ProductID)
);
```

```
CREATE TABLE ModelSparePart
(
    ModelID      int NOT NULL,
    SparePartID  int NOT NULL,
    FOREIGN KEY (ModelID) REFERENCES Model (ModelID) ,
    FOREIGN KEY (SparePartID) REFERENCES SparePart (SparePartID)
);
```

محصولات را هم به طور جداگانه و سفارشات را هم جداگانه ذخیره می‌کنیم تا با وجود چند خودرو در یک خرید، باز هم جدول‌های نرمالی داشته باشیم.

از آنجایی که در هر خرید می‌توانیم خودرو یا قطعه یدکی خریده باشیم، جدول جدیدی برای محصولات می‌سازیم، که نوع محصول (ماشین / قطعه) را در خود ذخیره دارد و کلید اصلی آن را در کنار دیگر داده‌های ماشین‌ها و قطعات ذخیره می‌کنیم.

```
CREATE DOMAIN ProductType AS VARCHAR(50) CHECK (VALUE IN ('Car', 'SparePart'));
```

```
CREATE TABLE Product
(
    ProductID int NOT NULL PRIMARY KEY,
    ProductType ProductType
);
```

```
CREATE TABLE Orders
(
    OrderID int NOT NULL PRIMARY KEY,
    CustomerID int NOT NULL,
    FOREIGN KEY (CustomerID) REFERENCES Customer (CustomerID)
);
```

```
CREATE TABLE OrderProduct
(
    OrderID int NOT NULL,
    ProductID int NOT NULL,
    Quantity int NOT NULL,
    FOREIGN KEY (OrderID) REFERENCES Orders (OrderID),
    FOREIGN KEY (ProductID) REFERENCES Product (ProductID)
);
```