

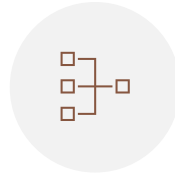
RAPID APPLICATION DEVELOPMENT (RAD)

(SERİ UYGULAMA GELİŞTİRME)

Rapid Application Development Nedir?



RAD, çok kısa bir zaman dilimi içinde, uygulamalar geliştirme konusuna odaklanmış bir yazılım geliştirme metodolojisidir.



Yüksek seviyede düşünüldüğünde, RAD; prototipleri, case tool larını kullanarak ihtiyaçlara göre sistemin tasarlandığı, iteratif bir yazılım geliştirme tekniğidir.



RAD deyimi, son zamanlarda, 60 ile 90 gün içerisinde tasarlanıp geliştirilebilen uygulamalar için kullanılan ticari bir terim haline gelmiştir.



Fakat gerçekte bu deyim, uygulama prototiplemeyi ve aşamalı uygulama geliştirmeyi kapsayan bir yazılım geliştirme sürecini ifade eder.



RAD yaklaşık 20 yıldır var olan bir kavram olmasına rağmen şu an ilk ortaya atıldığı kadar geçerli bir kavramdır.

PROBLEM

- ❑ Waterfall metodu gibi geliştirme süreçleri, kullanıcı ihtiyaçlarını karşılamayan sonuçlar ortaya çıkarmaktaydı.
- ❑ Geliştirme süreçleri çok uzundu ve sistem tamamlanana kadar ihtiyaçlar değişmekteydi.
- ❑ Geliştirme sürecinin bir aşaması tamamlanmadan, diğer aşamaya geçilememekteydi.
- ❑ Özellikle belirli gereksinimler üzerine yapılan bir yazılım süreci, bir noktada dondurulduğunda toplam geliştirme süreçleri çok fazla uzayacaktı.

RAD ÇÖZÜMÜ

- ❑ 1986'da Barry Boehm 'Yazılım Geliştirme ve İyileştirmede bir Spiral Model' başlıklı yazısında, prototipleme ve iteratif geliştirme kavramlarını ilk defa tanımlamış ve risk azaltma konusuna odaklanmıştır.
- ❑ 1980 lerin sonlarına doğru, Scott Shultz ve James Martin, prototipleme ve iteratif geliştirme kavramlarını bir araya getirerek, RIPP (Rapid Iterative Production Prototyping) "Seri iteratif geliştirme prototiplemesi" metodunu ortaya atmışlardır.
- ❑ James Martin PIPP yaklaşımını daha da genişleterek formülize etmiş ve 1991 yılında RAD (Rapid Application Development) "Seri Uygulama Geliştirme" isminde bir kitap yayınlamıştır.

Avantajları

Artan Hız:

- İsminde de belirtildiği gibi RAD ın temel avantajı, uygulama geliştirme hızını arttırarak proje teslim süresini kısaltmasıdır.
- Hafif bir sürüm olacak olan ilk versiyon hızlı bir şekilde tasarlanıp geliştirilirken, vakit kısıtlamaları göz önüne alındığında kısa vadede yazılıma eklenemeyecek olan gelişmiş özellikler ile ilgili olarak ileriki versiyonlarda planlamalar yapılır.

Artan Kalite:

- Artan kalite RAD metodolojisinin temel hedeflerindendir. RAD dan önce kalite ölçüsü, belki de daha çok sezgisel olarak, bir uygulama tamamlandığında özelleştirmelere ne kadar uyduğu ve ne kadar az aksayan tarafı bulunduğu ile ilgili bir derecelendirme idi.
- RAD a göre ise kalite, tamamlanan çalışma kullanıcı ihtiyaçlarını ile ne ölçüde karşıladığına ve bakım onarım maliyetlerinin ne kadar düşük olduğuna bağlıdır.

Dezavantajları

Azaltılan Özellikler

- Zaman kısıtlamaları sebebi ile sisteme konulacak olan özelliklerden bazıları sonraki versiyonlara bırakılabileceğinden dolayı RAD sistemin ilk versiyonlarında klasik yöntemlere göre daha az opsiyonlara sahip çözümler üretebilir.
- Bu durum, mümkün olduğu kadar hızlı bir şekilde müşteri ile açık diyaloglar kurularak, teslim edilecek sistem ve zamanlaması konusunda bilgi alışverişi yapılarak yönetilmelidir.

Azalan Ölçeklenebilirlik

- RAD öncelikle, süreçler tamamlandığında sistemin tümünü kapsayacak olan, çekirdek bir prototip geliştirmeye odaklandığı için, teslim edilen çözüm ölçeklenebilirlik açısından başlangıçta tüm sistem olarak tasarlanan çözüme göre eksik kalabilmektedir.

RAD'ın Temel Elemanları



Prototipleme

RAD in temelde, kullanıcı gereksinimlerinin ortaya çıkaran ve tüm sistemin geliştirilmesi için temel hareket noktası olacak olan ilk system prototipini geliştirmeye dayanır. Amaç, özellikler açısından hafif fakat çok kısa sürede mümkünse günler mertebesinde bitirilebilecek bir versiyon hazırlamaktır.



İteratif Geliştirme

İteratif geliştirme, küçük geliştirme periyodları içinde sistemin niteliklerinin versiyonlar ilerledikçe arttırılmasını ifade etmektedir. Her versiyon, bir sonraki versiyon için gereksinimleri belirlemek üzere müşteri ile birlikte gözden geçirilir.



Versiyon Planlama

Versiyon planlama, üzerinde çalışılan versiyonu tam zamanında çıkarabilmek için bazı özelliklerin ileriki versiyonlarda gerçekleştirilmek üzere ertelenmesi işlemidir.



Ekip Elemanları

RAD metodolojisi; tecrübeli, çok yönlü, farklı roller üstlenebilecek, motivasyonu yüksek elemanlardan oluşan ufak ekipler ile çalışmayı tavsiye eder.



Yönetim Yaklaşımı

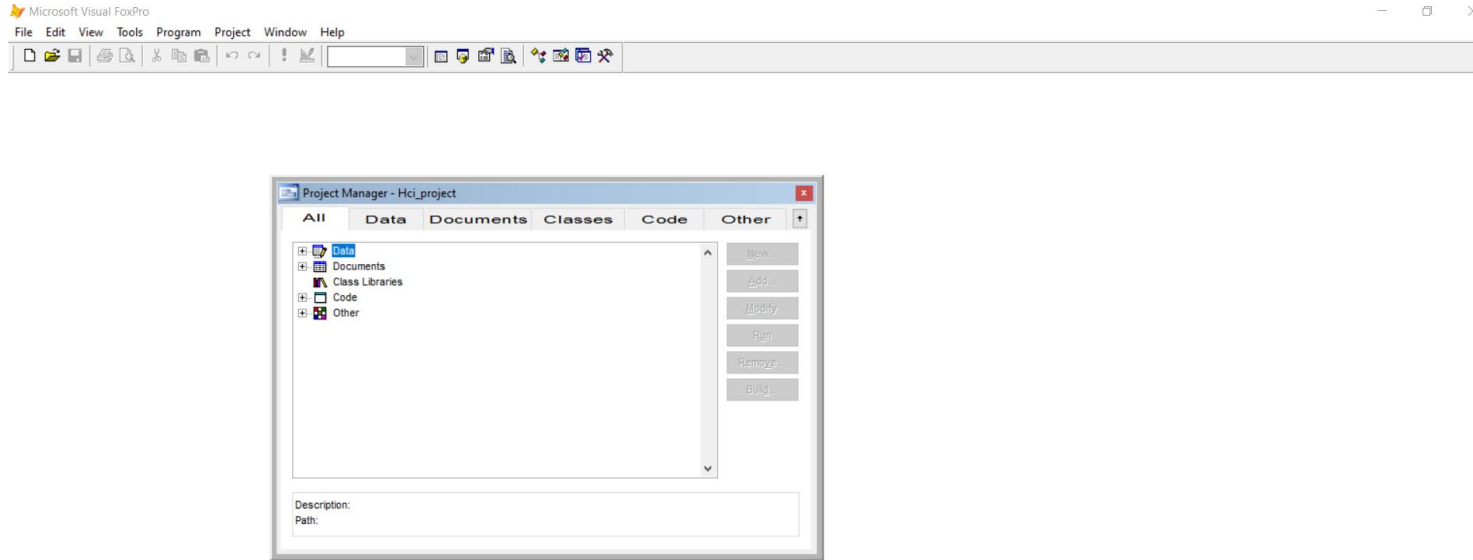
Aktif ve dahili bir yönetim anlayışı, geliştirme süreçlerinin uzamasına neden olan riskleri, müşteri yanlış anlamalarını ve aşılan zaman sınırlarını azaltma açısından hayati önem taşımaktadır.

FoxPro Nedir?



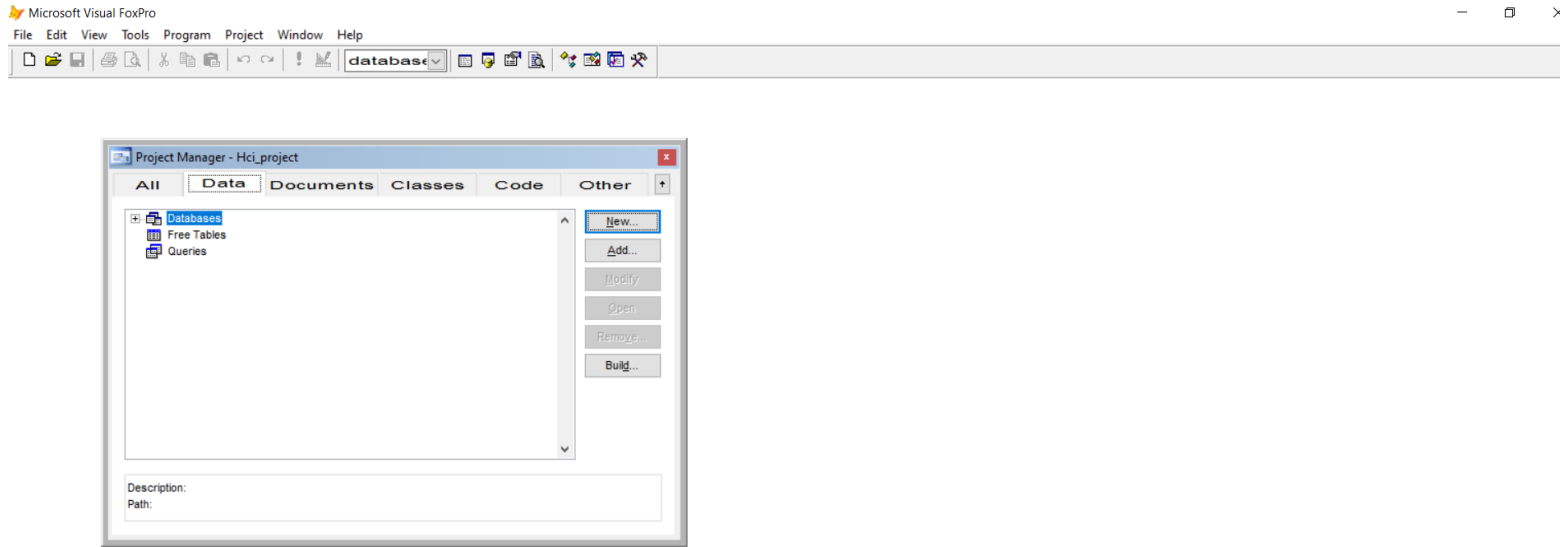
- Microsoft şirketine ait bir programdır.
- Veri tabanı uygulamaları kolaylıkla geliştirilebilir.
- Design time ve run time oldukça hızlıdır.
- Server olmayan veri tabanlarına göre performansı 10 kat güçlendiren rushmore teknolojisi kullanılmaktadır.
- Rushmore teknolojisi, kayıt kümelerine çok verimli bir şekilde erişilmesini sağlayan bir veri erişim tekniğidir.
- Rushmore sorgu optimizasyonu, bir dizi kaydı hızlı bir şekilde bulmak için dizinleri verimli bir şekilde kullanır.
- Bütün komutların ya da fonksiyonların ilk 4 karakterleriyle kodlama yapılabilmektedir.

Çalışma Ortamı



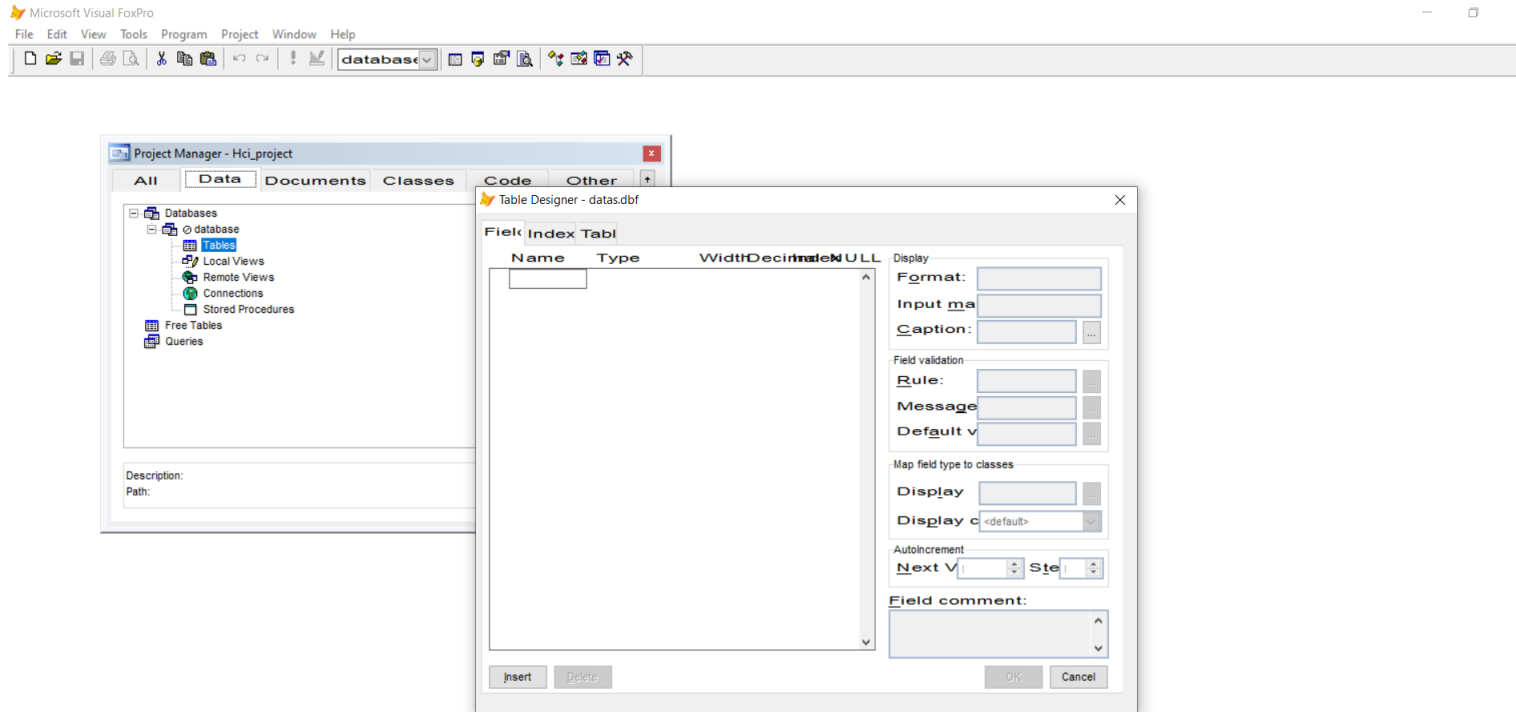
➤ Eylemleri seçme seçeneğine sahip olacağımız çalışma alanı gösterilmektedir.

Çalışma Ortamı



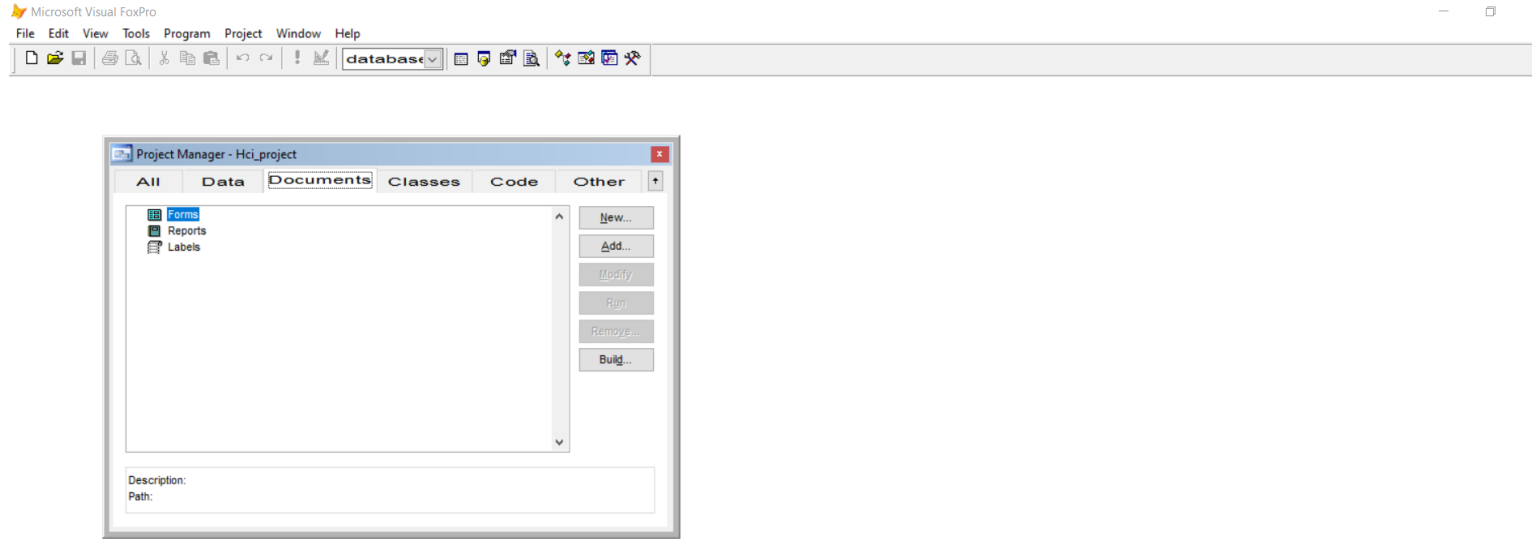
➤ Yeni bir veri tabanı oluşturmak için "Data" sekmesine gitmeli ve "Databases" seçeneğini seçmeliyiz

Çalışma Ortamı



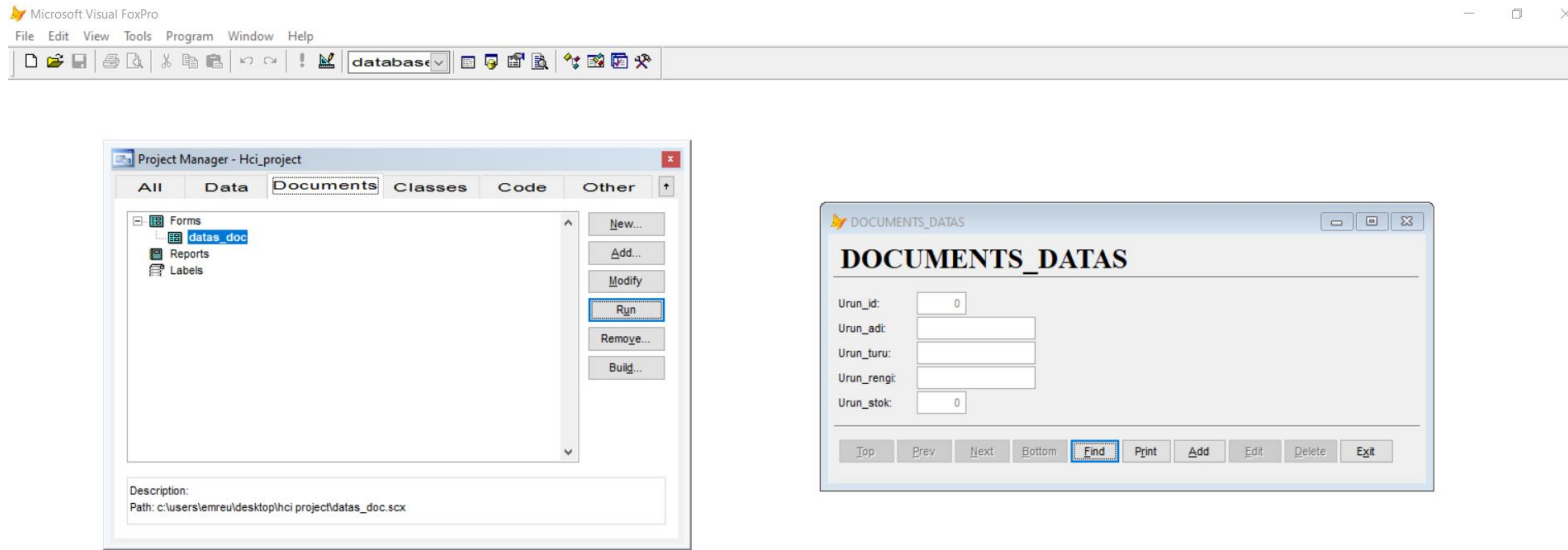
➤ Bu alanda alanlarımızı oluşturabilir, veri türünü ve istenirse 'NULL' seçeneğini atayabiliriz.

Çalışma Ortamı



- Alanlara değer girmek için "Belgeler" sekmesinden oluşturabileceğimiz bir "Form" gerekli olacaktır.

Çalışma Ortamı



➤ Form aracılığıyla tablolara kayıt ekleyerek, silerek veya değiştirerek tablolarımızla etkileşim kurabiliriz.

Çalışma Ortamı

Microsoft Visual FoxPro

File Edit View Tools Program Table Window Help

Properties - Desktop

Screen

All Da Me La: Ott Fa

ActiveContr: 0

ActiveForm: 0

AllowOutput: T. - True (Defa

AlwaysOnB: F. - False (Defa

AlwaysOnT: F. - False (Defa

AutoCenter: F. - False (Defa

AutoCompTi: (None)

BackColor: 255,255,255

BaseClass: Form

BindControls: T. - True (Defa

BorderStyle: 3 - Sizable (Def

BufferMode: 0 - None (Defau

Caption: Microsoft Visi

Class: Form

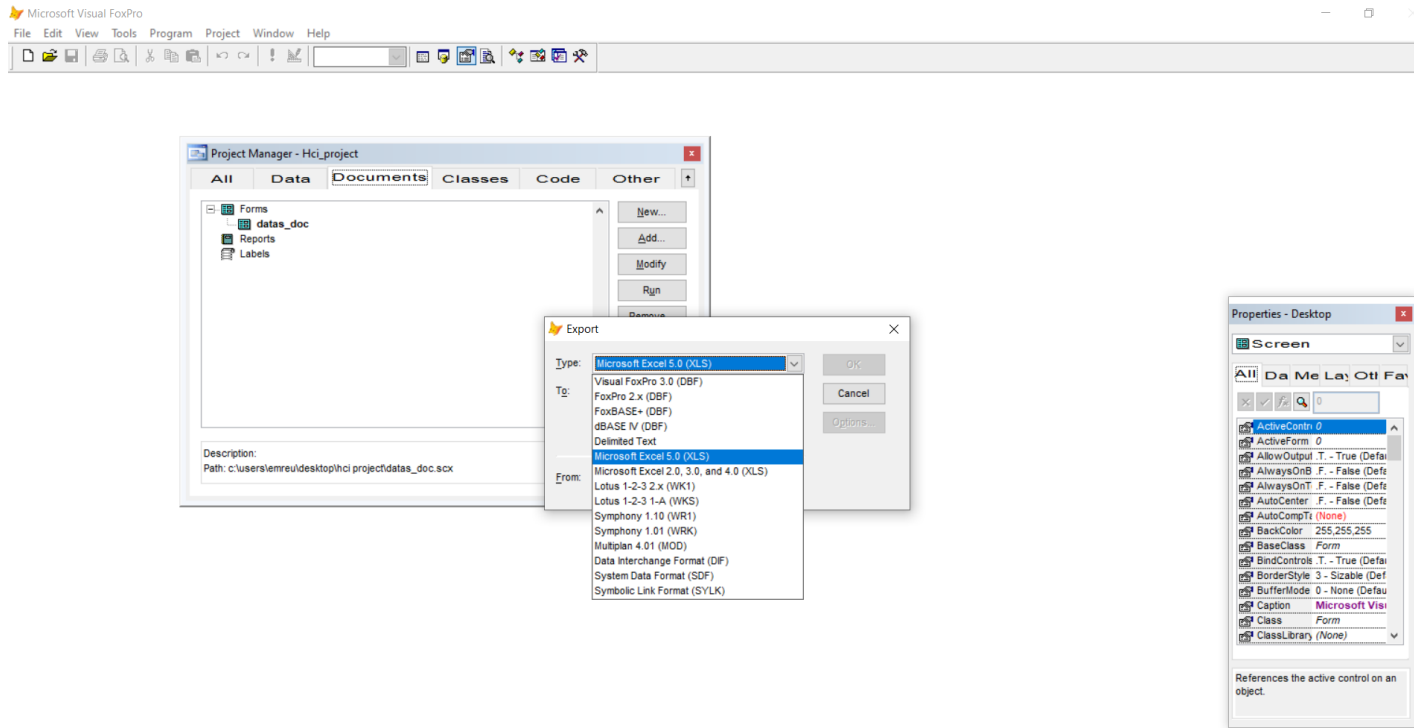
ClassLibrary: (None)

References the active control on an object.

Urun_id	Urun_adi	Urun_turu	Urun_rengi	Urun_stok
1	pantolon	kumaş	siyah	25
2	kot	pantolon	mavi	20
3	gömlek	kumaş	beyaz	45

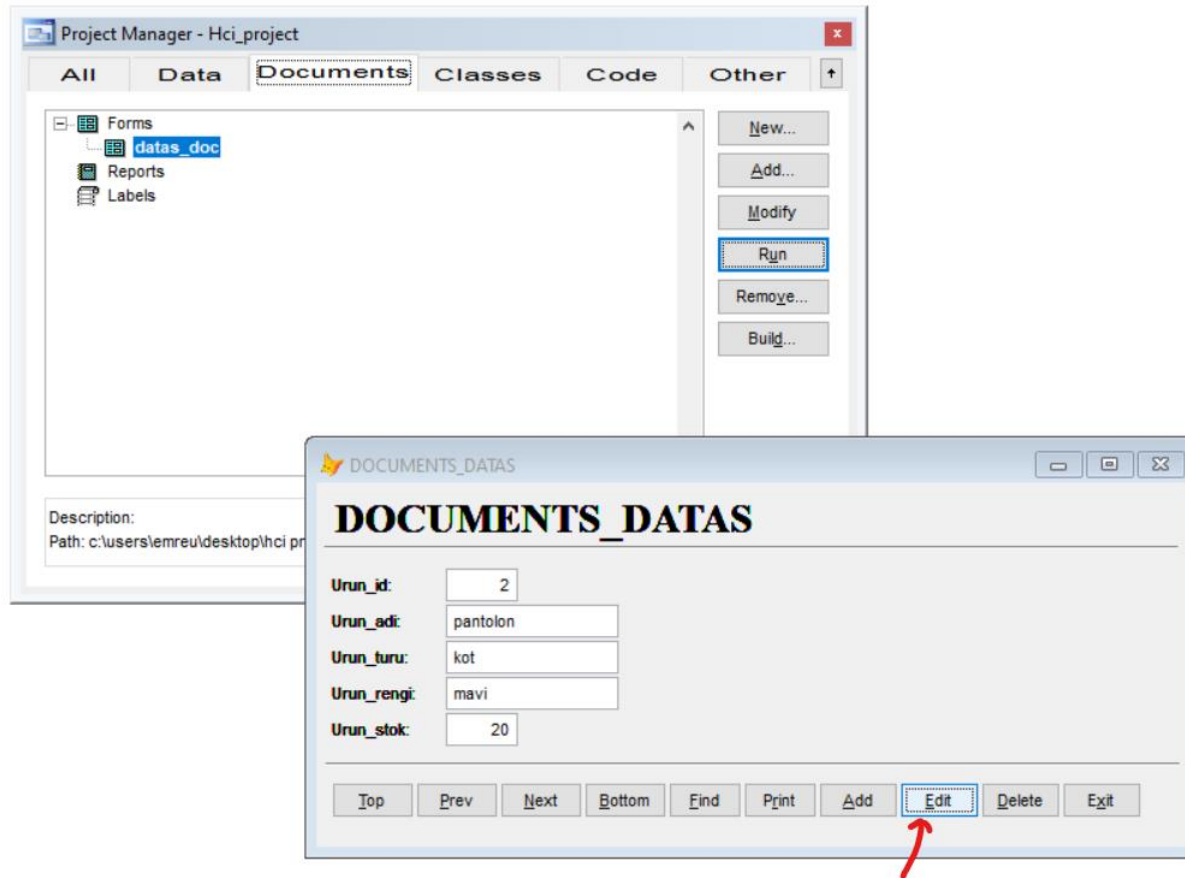
➤ Eklediğimiz kayıtları, tablo detaylarına inerek gözlemleyebiliriz.

Çalışma Ortamı



➤ Hazırlamış olduğumuz çalışmamızın export işlemiyle farklı platformlarda çalışmasını sağlayabiliriz.

Çalışma Ortamı



➤ Veri tabanında yer alan bir bilgiyi güncelleyebiliriz.

FoxPro ve .Net Karşılaştırması

İlişkisel veri tabanı yönetimi için sıkı kuralları bulunmamaktadır.

.NET ile farklı bir ilişkisel Veritabanı Yönetim Sistemi (RDBMS)' ne bağlanması gerekmektedir. Bu da sürekli kurulum yapmak gerektirdiği için zaman israfına sebep olur.

FoxPro, DOS döneminden beri yerleşik zengin raporlama yeteneğine sahipken, .NET geliştiricilerinin Crystal Reports, SSRS veya diğer üçüncü taraf varyantlarına bağımlı olması gerekir.

FoxPro aynı zamanda .NET yeteneklerine de sahiptir. Fakat .NET'e bağımlı değildir. Bu nedenle son yürütülebilir dosyanın .NET Framework yüklü olması gerekmez.

İLGİYLE DİNLEDİĞİNİZ TEŞEKKÜR EDERİM

Emre USTA

