

IDE4DB Veritabanı Geliştirme Platformu

Bitirme Projesi Sunumu

Onur EKER
040970627

Danışman:
Yrd. Doç Dr. Feza BUZLUCA

Sunum İeriđi

- Projenin Tanımı
- Projenin Amacı
- Projenin Analizi
- Projenin özüm Sunduđu Problemler
- Projede Neler Yapılabilir?
- Sonuç ve Öneriler
- Demo

Sunum İeriđi

- **Projenin Tanımı**
- Projenin Amacı
- Projenin Analizi
- Projenin özüm Sunduđu Problemler
- Projede Neler Yapılabilir?
- Sonuç ve Öneriler
- Demo

Projenin Tanımı

Veritabanları günümüz projelerinin tamamı tarafından kullanılan veri ambarlarıdır. Projenin ihtiyaçları ve olanakları doğrultusunda en uygun veritabanı seçilerek kullanılır. Farklı veritabanları için farklı arayüzler kullanmak geliştirme süresini ve maliyeti arttırır.

IDE4DB projesi, veritabanları için geliştirme platformu olmayı hedeflemiş açık kaynak kodlu bir projedir. Veritabanı bağımsız ve işletim sistemi bağımsız olarak işlevlerini gerçekleştirmesi projenin en önemli özelliklerinden biridir.

IDE4DB projesinin alt yapısı, ve büyük bir bölümü bitirme projesi olarak hedeflenerek gerçekleşmiştir. Fakat proje kapsamında hedeflenen diğer modüllerinde geliştirilmesine devam edilecektir.

Sunum İeriđi

- Projenin Tanımı
- **Projenin Amacı**
- Projenin Analizi
- Projenin özüm Sunduđu Problemler
- Projede Neler Yapılabilir?
- Sonuç ve Öneriler
- Demo

Projenin Amacı

- Veritabanı üzerinde bulunan tüm yapısal nesnelerin, veritabanı bağımsız olarak okunması ve modellenmesi
- Modellenen veritabanı yapısal ağacının görsellenmesi
- Veritabanı yapısal modellerinin karşılaştırılarak, iki veritabanı arasındaki fark modelinin oluşturulması.
- Fark modeli ve kullanıcının farklar üzerinde vereceği yapısal birleştirme kararları doğrultusunda yeni veritabanı yapısal modellerinin oluşturulması.
- Veritabanı yapısal modellerinin, veritabanı bağımsız olarak veritabanına uygulanması

Sunum İeriđi

- Projenin Tanımı
- Projenin Amacı
- **Projenin Analizi**
- Projenin özüm Sunduđu Problemler
- Projede Neler Yapılabilir?
- Sonuç ve Öneriler
- Demo

Projenin Analizi (Veritabanının modellenmesi)

- Veritabanı üzerindeki tüm yapısal nesnelere aynı sınıf ile modellenmeli, çünkü ancak bu şekilde yeteri kadar esnek bir modelleme sağlanabilir.
- Yapısal nesnelere temsil eden bu sınıf bir ağacın yaprakları gibi ata ve çocuk bilgilerini içermelidir, çünkü veritabanındaki tüm yapısal nesnelere böyle bir hiyerarşi içerisinde.
- Görselleme ve kodlama aşamasında sağladığı faydadan dolayı “Container” nesnelere de bu yapı ağacının bir parçası olmalıdır.

Projenin Analizi (Veritabanının modellenmesi)

Örneğin:

Veritabanı üzerindeki bir tablonun modellenmesi şu şekilde olmalıdır.

Table 1

Columns

Column 1

Column 2

Column 3

Primary Keys

Primary Key 1

Projenin Analizi (Veritabanının modellenmesi)

IDataObject ve INode arayüzleri

```
<<interface>>  
INode  
{ From common }  
  
Attributes  
  
Operations  
public INode getFirstChild( )  
public INode getLastChild( )  
public INode getNextChild( INode node )  
public INode getPrevChild( INode node )  
public INode getNextSibling( )  
public INode getPrevSibling( )  
public INode getParent( )  
public boolean hasChild( )  
public void addChild( INode node )  
public void deleteChild( INode node )  
public void clear( )
```

```
<<interface>>  
IDataObject  
{ From data }  
  
Attributes  
  
Operations  
public String getType( )  
public String getName( )  
public Map<String, Object> getProperties( )
```

Projenin Analizi (Sürücülerin modellenmesi)

- Veritabanı modellenin okunması ve yazılması aşamasında veritabanı sürücülerine ihtiyaç duyulmaktadır.
- Veritabanı sürücüsü, bağlantı bilgisini kullanarak ilgili veritabanına bağlanır.
- Bağlantı bilgisini bir örneği sürücü tarafından kullanıcıya sağlanmalı. Bu şekilde kullanıcıya bağlantı bilgisinin içermesi gereken verilerin bir örneği sağlanmış olur.
- Sürücüler okuma işlemi okuyucuları vasıtası ile gerçekleştirirler.
- Sürücüler yazma işlemi yazıcıları vasıtası ile gerçekleştirirler.

Projenin Analizi (Sürücülerin modellenmesi)

IDriver, IReader, IWriter arayüzleri

<p><<interface>> IDriver { From driver }</p>
<p>Attributes</p>
<p>Operations</p> <pre>public String getName() public String getCaption() public Boolean getCapability(String key) public IReader getReader() public IWriter getWriter() public ConnectionInfo getSampleConnectionInfo() public void init() public Object connect(ConnectionInfo connectionInfo)</pre>

<p><<interface>> IReader { From reader }</p>
<p>Attributes</p>
<p>Operations</p> <pre>public IDriver getDriver() public void init() public IDataObject readDataObjectTree(ConnectionInfo connectionInfo)</pre>

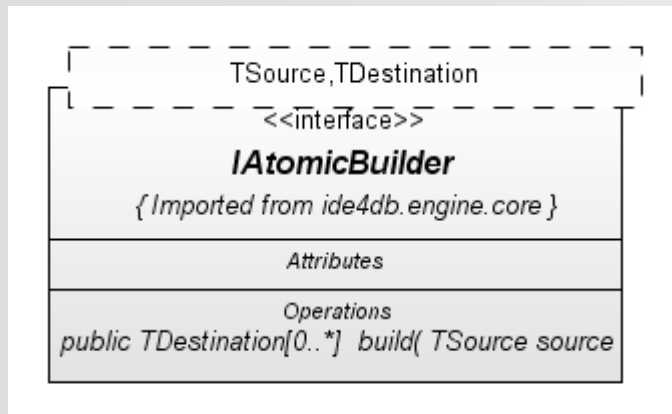
<p><<interface>> IWriter { From writer }</p>
<p>Attributes</p>
<p>Operations</p> <pre>public IDriver getDriver() public void init() public void writeDataObjectTree(IDataObject head, ConnectionInfo connectionInfo)</pre>

Projenin Analizi (Atomik işleme sınıflarının modellenmesi)

- Tüm işlemler esasen, ağaç şeklindeki bir modelin üzerinde gerçekleşecek atomik işlemlerin bütünüdür.
- Atomik işlem, giriş parametre tipi ve çıkış değeri tipi belli olmayan işlem birimidir.
- Atomik işlemlerde ağaç hiyerarşisini desteklemelidir. Örneğin tablo üzerinde işlem yapan bir atomik işlem modülü, kolon üzerinde işlem yapan bir atomik işlem modülünün atasıdır.

Projenin Analizi (Atomik işleme sınıflarının modellenmesi)

IAtomicBuilder arayüzü

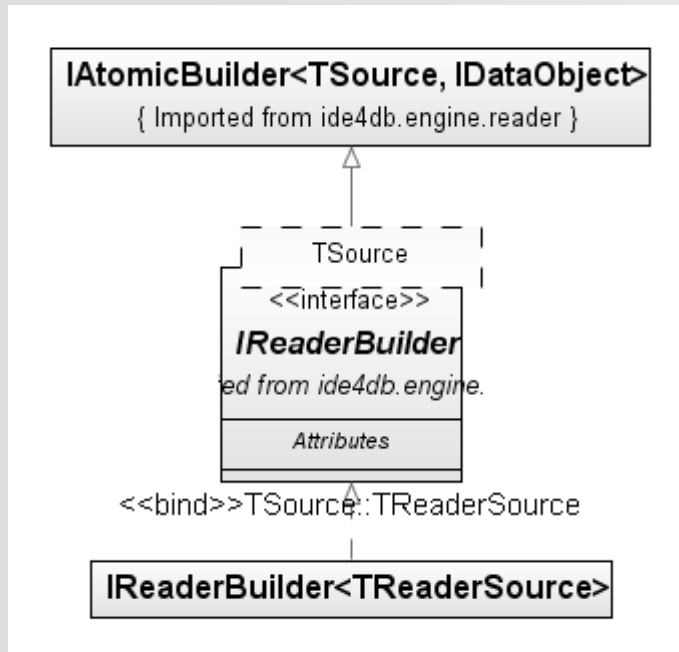


Projenin Analizi (Okuyucu sınıflarının modellenmesi)

- Okuyucu sınıf bağlantı bilgisi alıp, veritabanı ağaç modelini çeviren sınıflardır.
- Okuma işlemi esasen giriş parametre tipi bağlantı bilgisi, vs.. gibi değerleri taşıyan bir tip, çıkış parametre tipi ise veritabanı birim modeli tipi olan bir atomik işlemdir.

Projenin Analizi (Okuyucu sınıflarının modellenmesi)

IReaderBuilder arayüzü

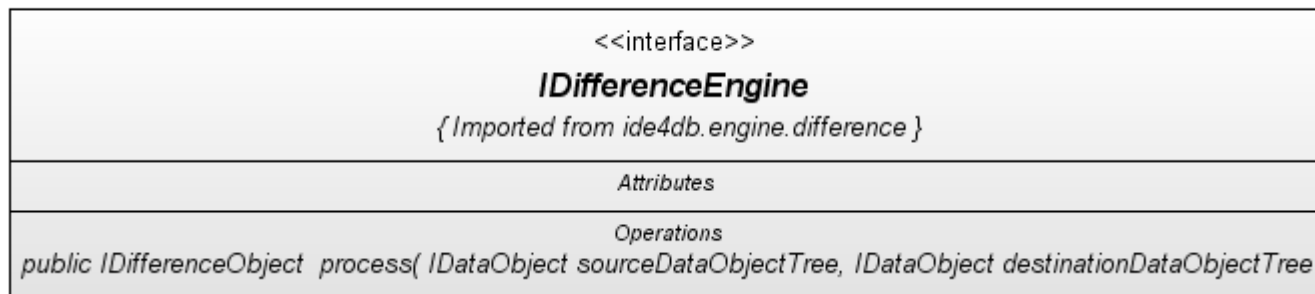
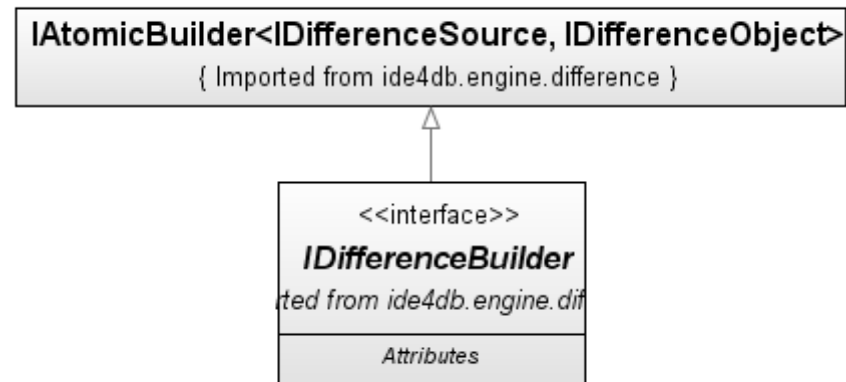
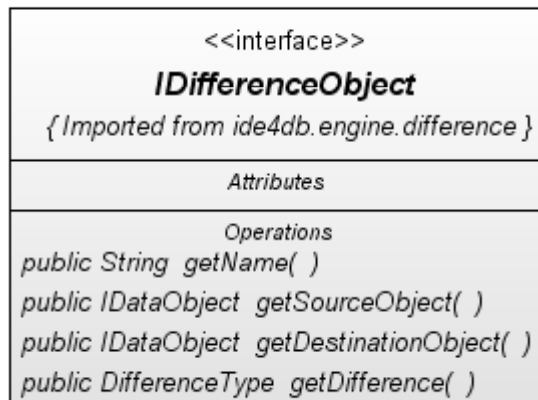


Projenin Analizi (Fark alma motorunun modellenmesi)

- Fark modeli, üzerinde iki veritabanı yapı modelini ve farkı tipini içerir. Ayrıca ağaç yapısını destekler.
- Fark alma motoru, iki farklı veritabanı ağaç modeli arasındaki farkları bulup fark ağaç modelini oluşturan motordur.
- Fark alma işlemi, esas olarak giriş veri tipi iki ağaç modelini içeren bir tip, çıkış veri tipi ise fark birim modeli olan bir atomik işlemdir.

Projenin Analizi (Fark alma motorunun modellenmesi)

IDifferenceObject, IDifferenceEngine, DifferenceBuilder arayüzleri



Projenin Analizi (Birleřtirme motorunun modellenmesi)

- Birleřtirme modeli iersinde fark modelini ve o fark modeli iin kullanıcının karar verdiėi birleřtirme ynn ieren bir modeldir.
- Birleřtirme motoru, birleřtirme aėa modelini ve birleřtirmenin ynn alarak yeni veritabanı yapısal aėa modelini oluřturan motordur.
- Birleřtirme iřlemi, esasen giriř veri tipi birleřtirme modeli ve birleřtirme yn ieren bir tip, ıkıř veri tipi ise veritabanı yapısal birim modeli olan bir atomik iřlemdir.

Projenin Analizi (Birleştirme motorunun modellenmesi)

IMergeObject, IMergeEngine arayüzleri

<p><<interface>> IMergeObject { Imported from ide4db.engine.merge }</p>
<p>Attributes</p>
<p>Operations public MergeType getMergeType() public void setMergeType(MergeType merge) public IDifferenceObject getDifferenceObject()</p>

<p><<interface>> IMergeEngine { Imported from ide4db.engine.merge }</p>
<p>Attributes</p>
<p>Operations public IDataObject process(IMergeObject mergeObject, MergeSelection mergeSelection)</p>

Sunum İeriđi

- Projenin Tanımı
- Projenin Amacı
- Projenin Analizi
- **Projenin özüm Sunduđu Problemler**
- Projede Neler Yapılabilir?
- Sonuç ve Öneriler
- Demo

Projenin Çözüm Sunduğu Problemler (Senaryo 1)

Problem:

Yazılımcı yaşayan bir projenin üzerinde hata düzeltme ve / veya yeni fazın geliştirmesini yapmaktadır. Yenileme süreçlerinin zaman aralığı arttıkça veritabanı üzerinde yapılan değişikliklerin takip edilmesi zorlaşır. Bu durumda hata düzeltme veya faz geliştirilmesinin yaşayan sistemi uygulanması son derece zor ve risk taşıyan bir süreçtir.

Çözüm:

Geliştirici yapılan her değişiklikte veritabanı üzerinde yapılanları tutmak yerine, nihai veritabanları arasındaki farkı alır ve değişiklikleri kontrollü bir şekilde hedef veritabanına aktarır.

Projenin Çözüm Sunduđu Problemler (Senaryo 2)

Problem:

Geliştirilen projenin başka bir veritabanı üzerine aktarılmasına karar verilir. Hedef veritabanına tüm yapının, kaynak veritabanındaki gibi aktarılması hataya açık ve zaman isteyen bir süreçtir.

Çözüm:

Kaynak veritabanı ve boş hedef veritabanı arasında fark alınır. Fark tümü ile hedef veritabanı üzerine senkronize edilir. Bu şekilde kaynak veritabanı tümü ile hedef veritabanı üzerine kopyalanmış olur.

Projenin Çözüm Sunduğu Problemler (Senaryo 3)

Problem:

Proje grubundaki kişiler aynı veritabanı üzerinde geliştirmeler yapar. Tüm geliştirmeler tamamlandıktan sonra nihai veritabanı yapısının oluşturulması ve bunun yaşayan sisteme aktarılması ancak geliştiricilerin kontrolünde gerçekleştirilebilen hataya açık bir işlemdir. Bu işlemin hatasız olması ve daha kısa sürede gerçekleştirilmesi istenmektedir.

Çözüm:

Her geliştiricilerin veritabanı biribirleri ile karşılaştırılarak, çakışmalar tespit edilir. Nihai bir veritabanı oluşturulduktan sonra orjinal veritabanı ile karşılaştırılıp son birleştirme yapılır.

Projenin Çözüm Sunduğu Problemler (Senaryo 4)

Problem:

Projenin geliştirilmesi esnasında kod üzerinde versiyonlama yapılır. Bu olanağı sağlayan programlar sınırsızdır. Ancak kod üzerinde olduğu gibi, veritabanı üzerinde de versiyonlama yapabilmek gerekir. Bu olanağı sağlayan programlar ise şuan çok sınırlı sayıda olup kullanımı yeteri kadar rahat değildir.

Çözüm:

Versiyonlama esnasında, o versiyona ait veritabanı uygun bir sanal sürücü ile (örneğin XStream sürücüsü) kopyalanır ve oluşan dosya veya dosyalarda versiyona dahil edilir.

Sunum İeriđi

- Projenin Tanımı
- Projenin Amacı
- Projenin Analizi
- Projenin özüm Sunduđu Problemler
- **Projede Neler Yapılabilir?**
- Sonuç ve Öneriler
- Demo

Projede Neler Yapılabilir?

- Veritabanı yapısal modelinin ER-Diagram ile gösterilmesi
- Veritabanı verisel modelleme, fark ve birleştirme modüllerinin oluşturulması
- Veritabanı bağımsız SQL motorunun geliştirilmesi
- Versiyon takip sistemleri için sürücülerin yazılması
- Uzaktan erişim için yazılan web-servis sürücüsünün tamamlanması
- Çoklu dil desteği

Sunum İeriđi

- Projenin Tanımı
- Projenin Amacı
- Projenin Analizi
- Projenin özüm Sunduđu Problemler
- Projede Neler Yapılabilir?
- **Sonuç ve Öneriler**
- Demo

Sonuç ve Öneriler

- Veritabanları üzerine geniş bilgi birikimi kazanılmıştır
- Bazı tasarım desenleri ilk defa uygulanarak, daha iyi kavranmıştır
- Nesneye dayalı programlama prensiplerinin uygulanarak pekiştirilmesi sağlanmıştır.
- Netbeans platformu üzerinde geliştirilme yapılması öğrenilmiştir.
- Generic sınıfların kullanımı ve kullanım durumları ile ilgili olarak yoğun pratik ve uygulama yapılmıştır.

Sunum İeriđi

- Projenin Tanımı
- Projenin Amacı
- Projenin Analizi
- Projenin özüm Sunduđu Problemler
- Projede Neler Yapılabilir?
- Sonuç ve Öneriler
- **Demo**

IDE4DB Veritabanı Geliştirme Platformu

Bitirme Projesi Sunumu Sonu

Teşekkürler

Onur EKER
040970627

Danışman:
Yrd. Doç Dr. Feza BUZLUCA