

# İÇİNDEKİLER

DATA WARE HOUSE (VERİ AMBARI) KURULUMU .....	2
DATA WARE HOUSE: .....	2
AMACIMIZ: .....	2
DATABASE KURULUMU: .....	2
SSIS PROJE OLUŞTURMA: .....	3
PROJE AYARLARI: .....	3
PROJE CONNECTION MANAGER: .....	4
SSIS DATA WARE HOUSE KURULUMU: .....	4
TABLO TEMİZLEME: .....	5
TABLO AKTARIMI: .....	6
SLOWLY CHANGING İLE KAYIT AKTARMA: .....	8
1-DEĞİŞKEN TANILMALA: .....	8
2- DEĞİŞKENİ SORGUYA EKLEME: .....	10
3- SLOWLY CHANGING OLUŞTURMA: .....	11
4- SİLİNER KAYITLAR: .....	13
SSIS PROJE KAYDETME-BUILD-DEPLOY İŞLEMLERİ: .....	17

# DATA WARE HOUSE (VERİ AMBARI) KURULUMU

## DATA WARE HOUSE:

Veri ambarı, iş zekâsı (BI) faaliyetlerine, özellikle de analitiğe olanak tanımak ve bunları desteklemek üzere tasarlanmış bir veri yönetim sistemidir. Veri ambarları yalnızca sorgulama ve analiz amacıyla kurulur ve çoğu zaman geçmişe ait büyük miktarlarda veri içerir. Bir veri ambarındaki veri genellikle uygulama yazılımlarının günlük dosyaları ve işlem uygulama yazılımları gibi çok çeşitli kaynaklardan elde edilir.

Bir veri ambarı, çok sayıda kaynaktan gelen büyük miktardaki verileri merkezi hâle getirir ve birleştirir. Analitik yetenekleri kurumların karar vermeyi geliştirmek için verilerinden değerli iş iç görüleri elde etmelerine olanak tanır. Zaman içinde, veri bilimcileri ve iş analistleri için paha biçilmez bir tarihi kayıt oluşturur. Veri ambarı, bu özellikleri sayesinde bir kuruluşun "tek doğruluk kaynağı" olarak görülebilir.

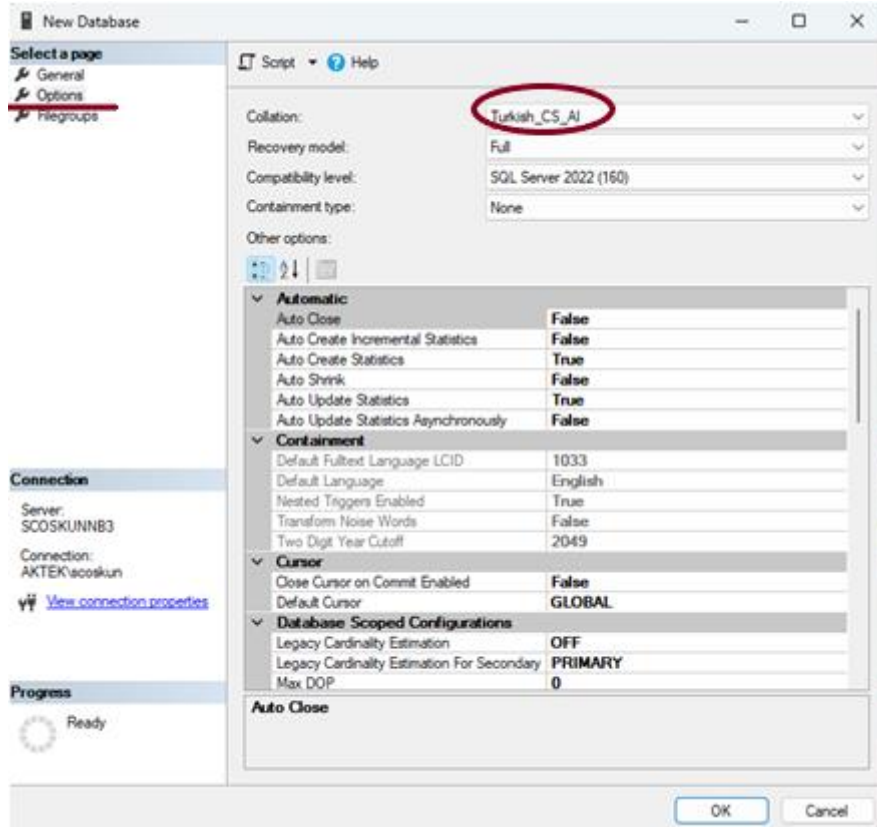
## AMACIMIZ:

DWH ı kurma amacımız şubelerimizdeki verileri tek bir çatı altında toplayıp, bu veriler üzerinden raporlarımızı çalıştırmak ve bu sayede canlı database'i işlem yoğunluğuna tutmamış olup sadece raporlar için kullanabileceğimiz bir database yapmaktır.

Bu işlemler için taşıma, değiştirme ve silme işlemlerini SSIS yardımcılığıyla yapacağız

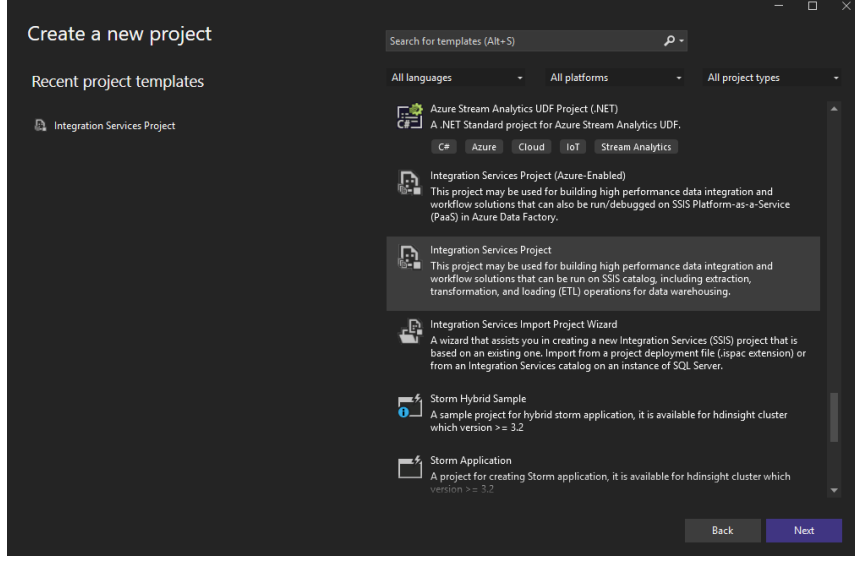
## DATABASE KURULUMU:

SQL tarafında oluşturacağımız database için önemli olan hususlardan bir tanesi, DWH için kuracağımız database dili ile ana database dilinin aynı olması gerekmektedir. Bu ayarı New Database işlemi esnasında Options sekmesinde collaction alanında olması gereken dili seçeriz.



## SSIS PROJE OLUŞTURMA:

Visual Studio 'da Create a new project or solution seçeneğinden "Integration Services Project" ile yeni projemizi oluştururuz.

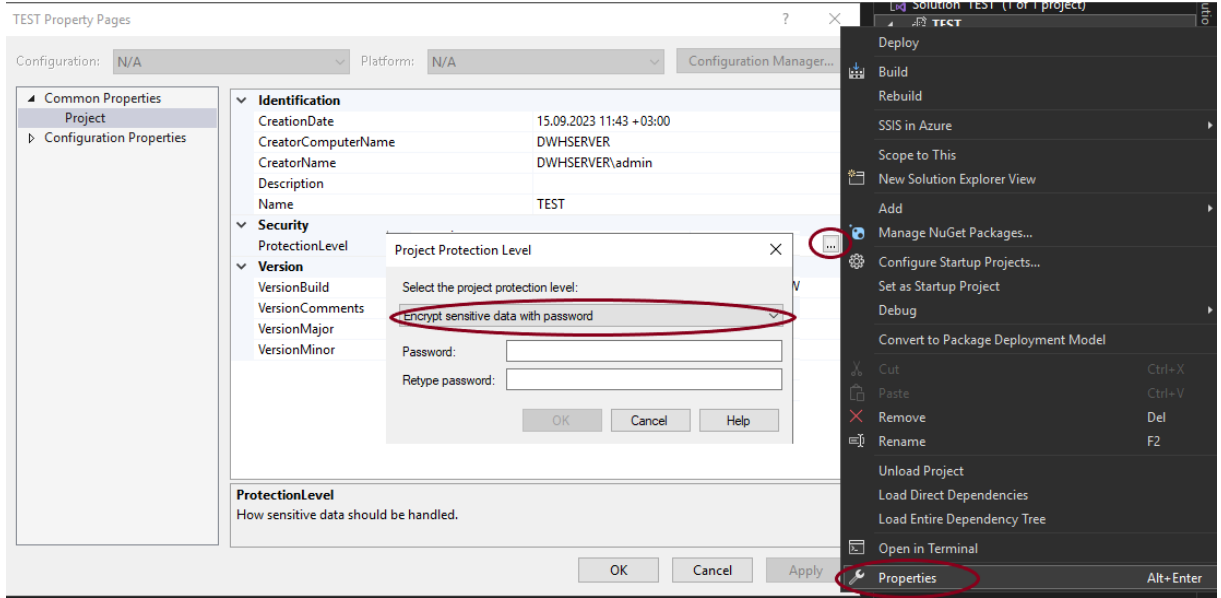


### PROJE AYARLARI:

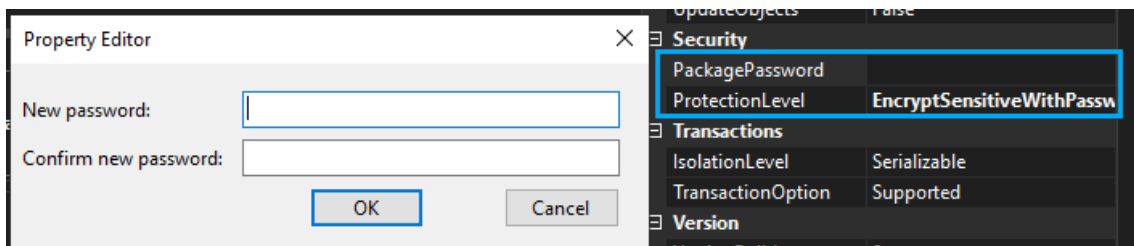
Projemizi birden fazla kişinin kullanacağına varsayarsak şifreli giriş önlemi almak isteyebiliriz.

Bunun için hem projemizin hem de paketimize güvenlik ayarlarımızı değiştirmemiz gerekmektedir.

İlk olarak projemize sağ tıklayarak properties -> ProtectionLevel -> Encrypt sensitive data with password'u seçeriz. Bu özellik ile projemize giriş yapılacak şifreyi belirleriz.



Control flow sayfasına gelerek sağ tık-> properties -> ProtectionLevel -> Encrypt sensitive data with password'u seçeriz PackagePassword ile aynı işlemleri yapmamız gerekmektedir.

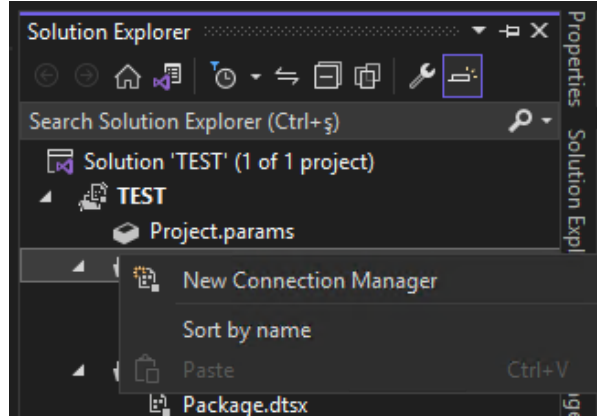


Bu ayarlar ile projemize şifre ile giriş özelliği sağlamaktayız.

## PROJE CONNECTION MANAGER:

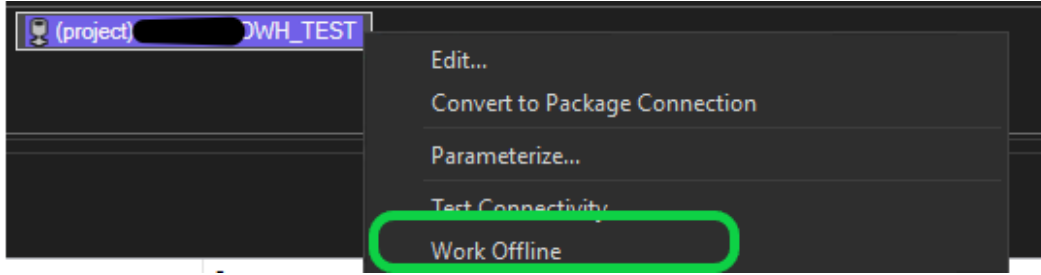
Projemizde kullanacağımız databaseleri tanımlayacağımız alandır.

Solution Explorer'dan Connection managers alanına sağ tık yaparak New connection manager seçilir.



Çıkan seçeneklerde SQL tarafında çalışacağımız için OLE DB seçeriz. New sekmesinden bağlamak istediğimiz databaselerin bilgilerini girdikten sonra connectionlarımız gelecektir. Biz bu alanda ana db miz ve hedefte çalışacağımız db 'yi ekleriz.

Daha sonra connection manager sekmesine gelen bağlantılarımızın ayarlarında work offline işaretli olmaması gerekmektedir.



Bu işlemlerden sonra projemiz aktarım yapısını kurmak için uygun hale gelmiştir.

## SSIS DATA WARE HOUSE KURULUMU:

Aktarım için iki adet databasemiz olacaktır.

- Ana db: Canlı işlemlerin yapıldığı kaynak db miz.
- Dwh db: Kaynaktaki verileri yazdıracağımız hedef db miz.

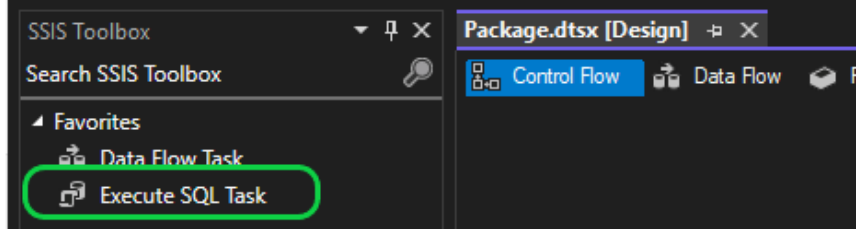
Özet olarak işlem sıralamamız şu şekilde olacaktır.

1. Kaynaktan aktaracağımız tablolarımızı belirleriz.
2. Tablo temizleme: Control Flow dan Execute SQL Task ile belirlediğimiz tabloları dwh db için truncate cümlelerini yazarız. Bizim izleyeceğimiz yolda büyük tablolarımızı bu truncate cümlelerinin dışında bırakırız bunun sebebi sil ve içeri aktar işleminde içeri aktarım sırasında ana db 'de uzun süre meşgul etmesinden kaçınmak istememizdir.
3. Tablo Aktarımı: Aktarım yapacağımız tablolarımızı Data Flow Task içerisinde OLD DB Source ile ana db ye bağlanarak okuma yapacağımız tablomuzu seçeriz. OLE DB Destination ile Dwh db de aktarım yapacağımız tablomuzu belirleriz.
4. Büyük Tabloların Aktarımı: Büyük tablolarımızın kaynak db yi çok fazla meşgul etmesini istemediğimizden dolayı bu tabloların aktarımını SSIS Data Flow fonksiyonu olan SLOWLY CHANGING DIMENSION yapısını kullanırız.

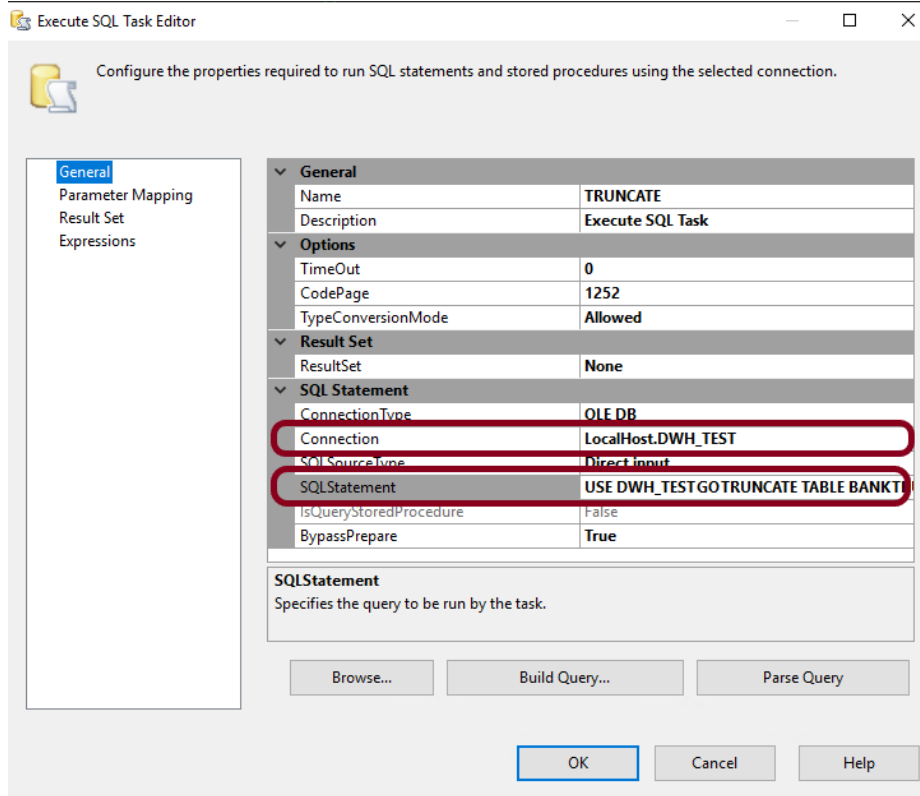
Not: Büyük tabloların aktarımı için direkt olarak temizle ve içeri aktar ile de yapabilirsiniz. Bu işlem sizin alacağınız karara bağlıdır.

## TABLO TEMİZLEME:

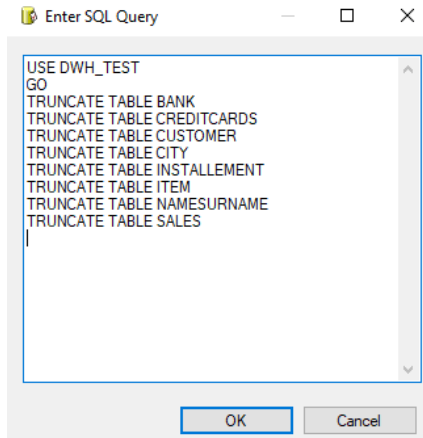
Paketimizi her çalıştırma işleminde ilk olarak ilgili tabloların içerisini temizlemesini istemekteyiz. Bunun için Control Flow içerinden Execute SQL Task 'i kullanırız. Bu Execute SQL Task işlemine TRUNCATE ismini vereceğim.



İlk olarak connection alanından işlem yapacağımız (Dwh db) db 'yi belirleriz. Bu alanda dikkat etmemiz gerekir yanlış bir db seçimi yapmamız ana db üzerinde kayıtların silinmesine neden olacaktır.



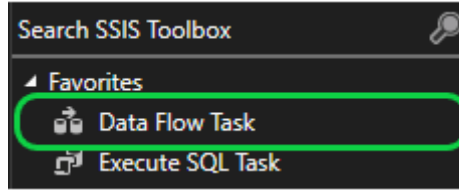
SQL Statement alanından temizleyeceğimiz tabloların SQL komutunu yazacağız.



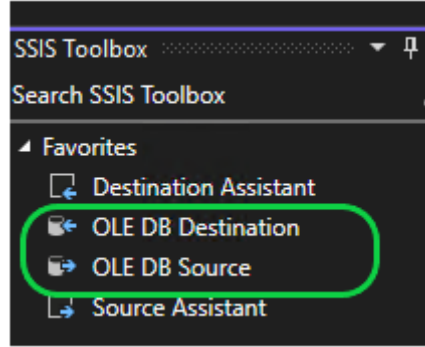
Burada USE ile database belirtme seabim olası yanlış db seçiminde hata alıp sorgumuzu çalışmasını engellemek amaçlıdır.

## TABLO AKTARIMI:

Tablolarımızı kaynaktan hedefe aktarımı yapmak için Data Flow Task kullanırız.

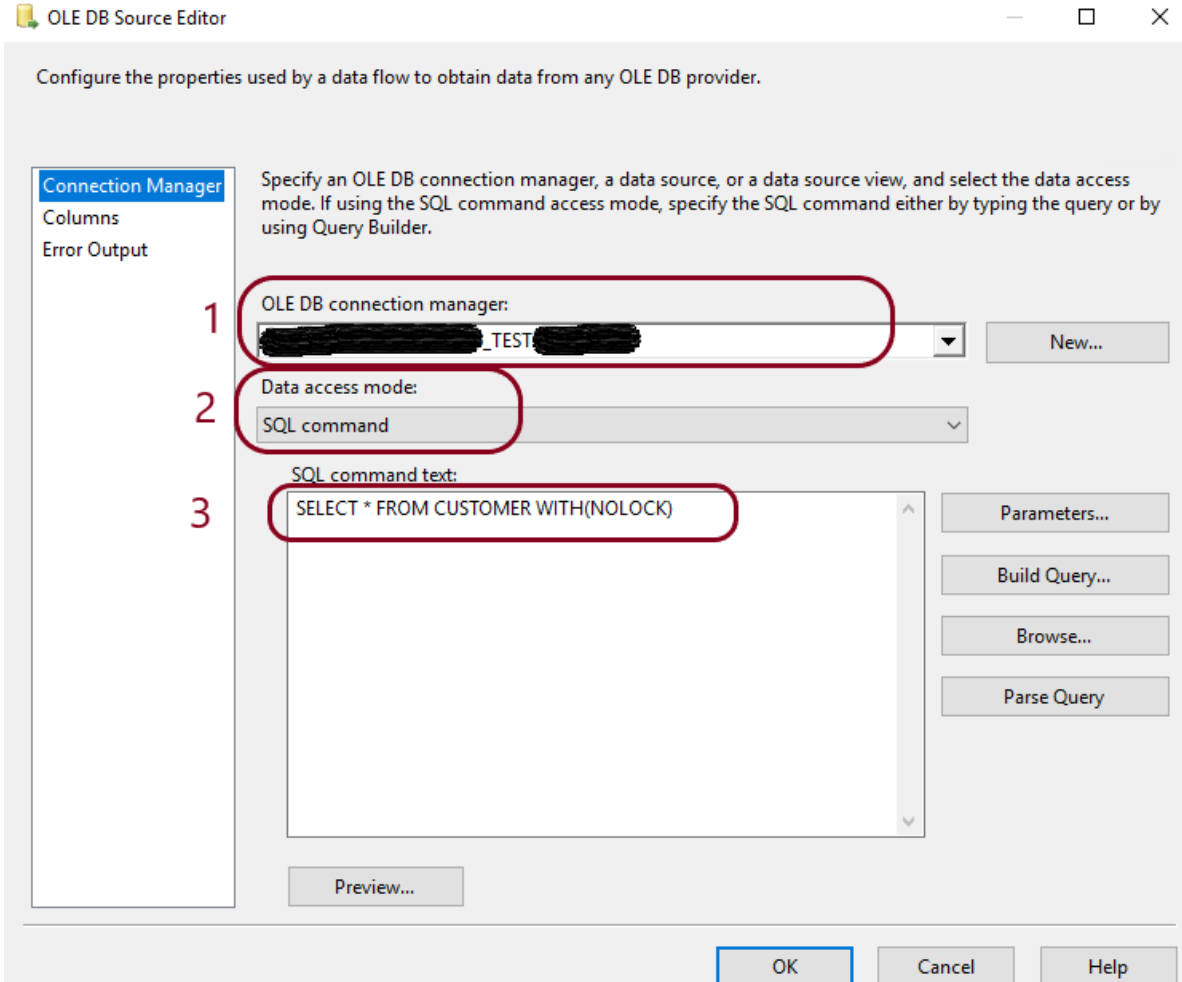


Data Flow Task 'tan kaynaktan veri okumak için OLE DB Source ve hedefe yazdırmak için de OLE DB Destination araçlarını kullanacağız. Data Flow Task için AKTARIM ismini kullanacağım.



OLE DB Source ile kaynaktaki veriyi okumamızı sağlayacaktır. İçerisine girdiğimizde OLE DB Connection manager alanından kaynak db 'i seçeriz.

Data Access mode alanından SQL command seçeriz ve daha sonra kaynaktan alacağımız tablonun sorgusunu yazarız.



Source işlemini bitirdikten sonra Source işleminden çıkan ok ile Destination a bağlarız.



OLE DB Destination kaynaktan okutulan veriyi hedef db ye yazdırmamızı sağlayacaktır. İçerisine girdiğimizde OLE DB Connection manager alanından hedef db 'i seçeriz.

Data Access mode alanında Table or view –fast load seçeneğini seçeriz.

Name of the table or the view alanında eğer aktaracağımız tablo dwh database içerisinde yoksa New butonuyla yeni tablomuzu SSIS aracılığıyla oluştururuz. Eğer tablo DWH db de varsa ilgili tablomuzu seçeriz.

Keep identity alanını işaretledikten sonra mappings alanından mapleme işlemini yaparız.

The screenshot shows the 'OLE DB Destination Editor' dialog box. The title bar says 'OLE DB Destination Editor'. The main text says 'Configure the properties used to insert data into a relational database using an OLE DB provider.' On the left, there is a tree view with 'Connection Manager', 'Mappings', and 'Error Output'. 'Connection Manager' is selected. The main area contains the following fields and options:

- 1. OLE DB connection manager: LocalHost.DWH\_TEST (with a 'New...' button)
- 2. Data access mode: Table or view - fast load (with a dropdown arrow)
- 3. Name of the table or the view: [dbo].[CUSTOMER] (with a 'New...' button)
- 4. Checkboxes: ☒ Keep identity, ☐ Keep nulls, ☒ Table lock, ☒ Check constraints
- Rows per batch: (empty text box)
- Maximum insert commit size: 2147483647
- View Existing (button)
- OK, Cancel, Help (buttons at the bottom)

Bu işlemleri aktarmak istediğimiz her tablo için ayrı ayrı yaparız.

NOT: Büyük tabloları ayırmak istemeyip tüm tabloları bu yapıda işleme almak isteyenler için işlemler buraya kadardır.

16. Sayfa da paketi kaydetme, build ve deploy işlemlerine geçebilirler.

### SLOWLY CHANGING İLE KAYIT AKTARMA:

Kayıt sayısının çok olduğu tabloların aktarımında, aktarma işlemlerinin ana db yi uzun süre meşgul edeceği için pek tercih ettiğimiz bir yöntem değildir.

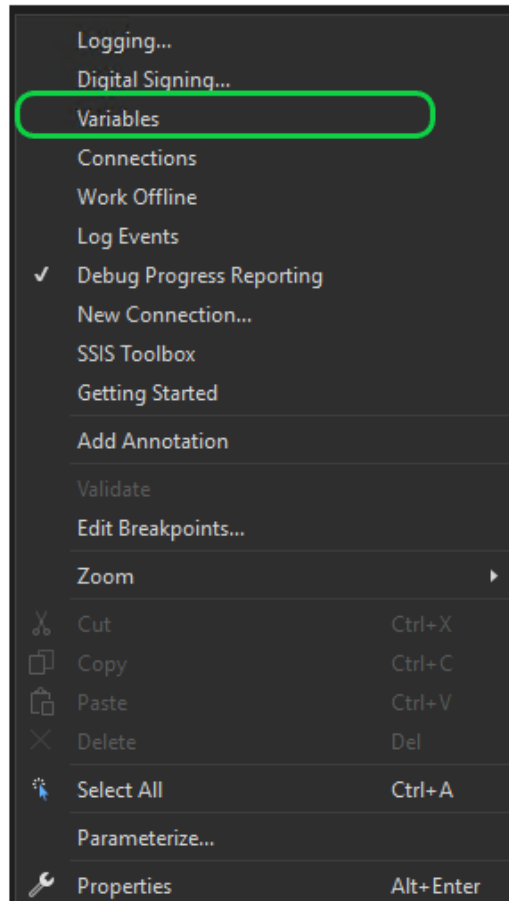
Bu yüzden kaydın çok olduğu tabloları Data Flow Task Fonksiyonu olan SLOWLY CHANGING DIMENSION mimarisini kullanacağız. Bu yapı ana db üzerinde değişen kayıtların güncellenmesi ve yeni oluşan kayıtların da eklenmesini sağlamaktadır.

İlk olarak gerekli kayıtların okunması için OLE DB Source ile ilgili tabloyu okutma işlemi yaparız. Kaynaktan verileri alırken tüm tabloyu okumasını istemediğimiz için where şartımızda dwh db 'de olan son kayıttan sonraki kayıtları getirmesi için bir değişken tanımlamalıyız.

Bu değişken için tanımlayacağımız kolon, tablomuzda ID kolonu veya herhangi bir güncelleme ve ekleme yapıldığında güncel tutulan last update kolonu olabilir. Biz bu yapı için last update kolonunu kullanacağız.

### 1-DEĞİŞKEN TANILMALA:

Control Flow sayfasına sağ tık ile çıkan seçeneklerden Variables i seçeriz



Açılan sayfada paketimize tanımlayacağımız değişkenimizin ismini ve data type ı belirleriz.

Variables			
Name	Scope	Data type	Value
maxdate	Package	String	



Bu deęiřkene deęer tanımlamak için Execute SQL Task ile DWH db 'ye baęlanarak son ID deęeri ya da son last update tarihli kaydı getiren sorguyu yazarız. Bu Execute SQL Task için DEęİřKEN ismini kullanacaęım.

Sorgumuz tek deęer döneceęi için Resul set alanında Single Row olarak seęilmelidir.

Bu sorguyu max komutu ile en yüksek deęeri getiririm. Sorgumuzda gelen deęerin Alias ı mutlaka deęiřken ile aynı isim olmalıdır.

Sorgumuzu ve result set ayarını yaptıktan sonra sol alanda bulunan Result set 'den parametre birleřtirme iřlemini yaparız.

The screenshot shows the 'Execute SQL Task Editor' window. On the left, the 'General' tab is selected, and the 'Result Set' section is highlighted with a red circle and the number 4. The 'General' tab contains the following fields:

General	
Name	DEęİřKEN
Description	Execute SQL Task

Options	
TimeOut	0
CodePage	1252
TypeConversionMode	Allowed

Result Set	
ResultSet	Single row

SQL Statement	
Connection	LocalHost.DWH_TEST
SQLSourceType	Direct input
SQLStatement	SELECT MAX(LASTUPDATE) maxsth FROM CUS
IsQueryStoredProcedure	False
BypassPrepare	True

At the bottom of the 'Execute SQL Task Editor' window, there are three buttons: 'Browse...', 'Build Query...', and 'Parse Query'. Below these buttons are 'OK', 'Cancel', and 'Help' buttons.

On the right, the 'Enter SQL Query' dialog box is open, showing the SQL query: `SELECT MAX(LASTUPDATE) maxdate FROM CUSTOMER`. The query is highlighted with a red circle and the number 2. The dialog box has 'OK' and 'Cancel' buttons at the bottom.

Result Set alanından Alias için kullandıęımız ismi Result Name e yazarız. Variable Name için paketimize tanımladıęımız deęiřkenimizi seęeriz.

Bu iřlemler ile paketimizde oluřturduęumuz deęiřkene yazmıř olduęumuz sorgudan gelen deęeri getirmektedir.

Bu deęer DWH db deki ilgili tablonun son kaydın olduęunu gsteren deęerdir.

The screenshot shows the 'Parameter Mapping' tab in the 'Execute SQL Task Editor' window. The 'Result Set' section is highlighted with a red circle and the number 4. The 'Parameter Mapping' section contains the following fields:

Result Name	Variable Name
maxdate	User::maxdate

## 2- DEĞİŞKENİ SORGUYA EKLEME:

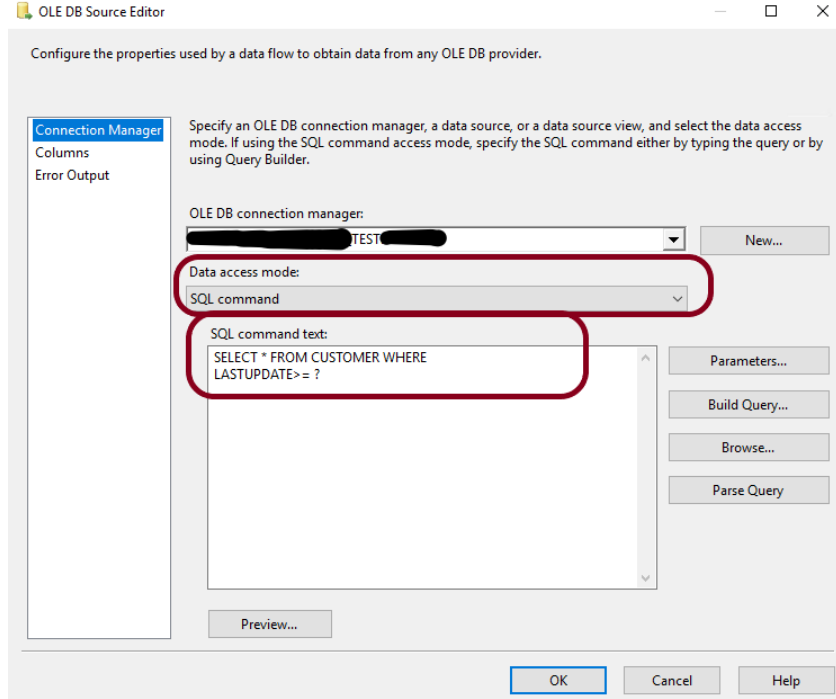
Değişkenlerimizi tanımladıktan sonra SLOWLY CHANGING işlemine özel bir Data Flow Task oluştururuz.

Bu Data Flow Task için SLOWLY CHANGING ismini kullanacağız.

OLE DB Source ile kaynaktan veri okuma işlemi yapacağız. Bu işlem için hazırlayacağımız SQL sorgusunda değişkenimizden gelen değer ile where şartı uygulayacağız.

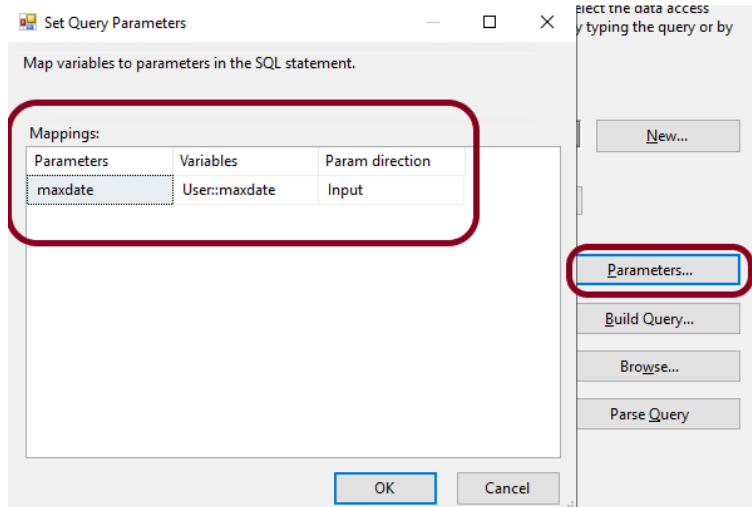
Değişkenimizi sorguya tanımlamak için SSIS de “?” işareti kullanılır.

ÖR: WHERE LASTUPDATE >= ? Bu sorgu ile değişkenimize tanımladığımız DWH db ‘de max last update ‘den büyük ve eşit değerleri getirmektedir.



Parametremizi tanımlamak için OLE DB Source içerisinde sağ alanda bulunan Parameters... ‘e tıklarız.

Açılan pencerede Parametre adını ve hangi parametreyle eşleştireceğini seçeriz.

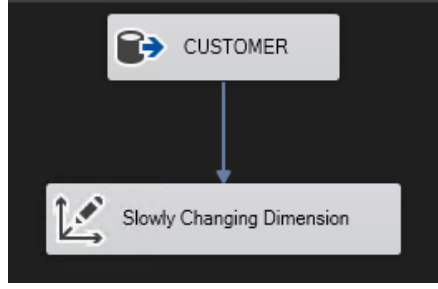


Bu işlemleri yaptığımızda artık ana kaynaktan bizde var olmayan ya da güncellenmiş kayıtları getirmiş olacağız bu sayede kaynaktan daha az veri okuma işlemi yapmaktayız.

### 3- SLOWLY CHANGING OLUŞTURMA:

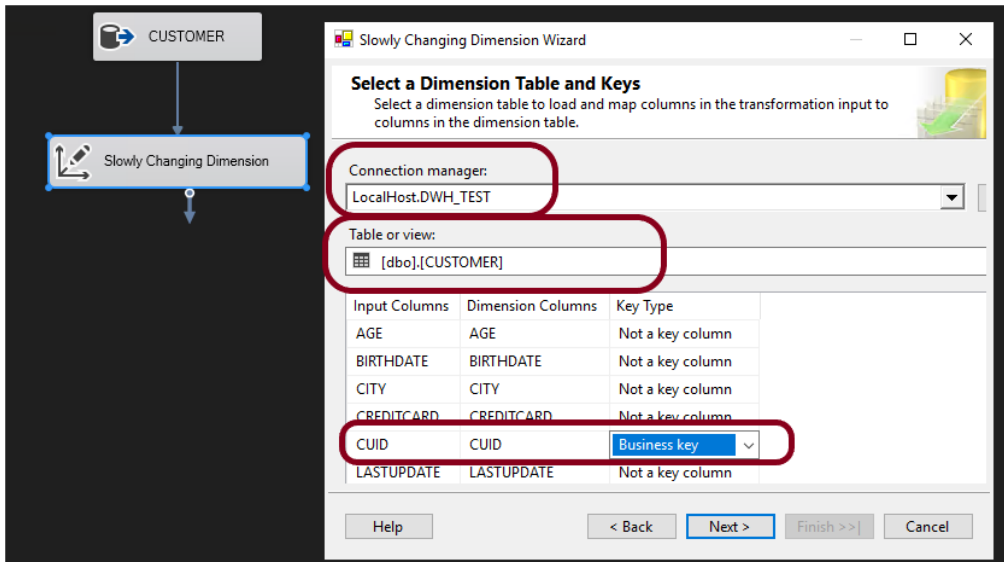
Data Flow içerisinde SSIS Toolbox menüsünde yer alan Slowly Changing Dimension yapısını kullanacağız.

OLE DB Source ile yaptığımız kaynaktan veri okuma işlemini Slowly Changing işlemimize bağlarız.

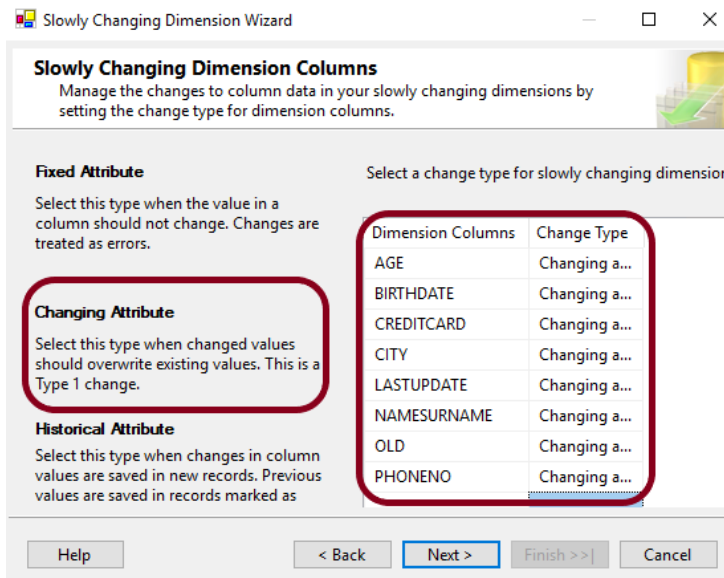


Slowly Changing içerisine girdiğimizde Source ile okuttuğumuz veriyi hangi veri ile karşılaştıracığımızı seçeriz. Bu yüzden Dwh db 'yi ve ilgili tablomu seçerim. Eşlemede kullanmak için en az bir tane business key seçmemiz gerekir.

İşlemlerimiz tamamlandıktan sonra Next butonuyla devam ederiz.



Dimension Columns ekranından Changing Attribute özelliğini kullanacağız. Bu özellik tanımladığımız kolonda kaynaktan verinin değişmesi sonucu hedef db (dwh db) 'de değişiklikleri ilgili kolonda uygulamasıdır.



İşlemlerimiz tamamlanınca Next butonuyla devam ederiz.

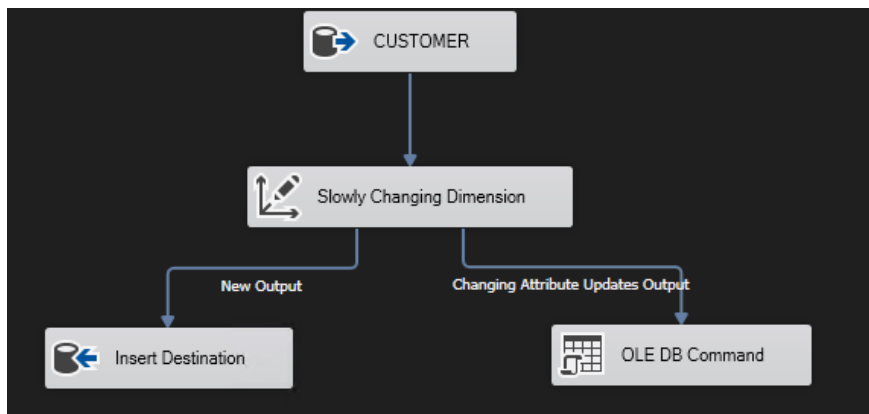
Fixed and Changing Attribute Options alanında herhangi bir alanı seçmeden next ile devam ederiz.

The screenshot shows the 'Slowly Changing Dimension Wizard' window, specifically the 'Fixed and Changing Attribute Options' step. The window has a title bar with the text 'Slowly Changing Dimension Wizard' and standard window controls. The main area is divided into two sections: 'Fixed attributes' and 'Changing attributes'. Under 'Fixed attributes', there is a checkbox labeled 'Fail the transformation if changes are detected in a fixed attribute'. Under 'Changing attributes', there is a checkbox labeled 'Change all the matching records, including outdated records, when changes are detected in a attribute'. At the bottom of the window, there are four buttons: 'Help', '< Back', 'Next >', and 'Finish >>|'. The 'Next >' button is highlighted with a blue border.

Inferred Dimension Members alanında herhangi bir alanı seçmeden next ile devam ederiz.

The screenshot shows the 'Slowly Changing Dimension Wizard' window, specifically the 'Inferred Dimension Members' step. The window has a title bar with the text 'Slowly Changing Dimension Wizard' and standard window controls. The main area is divided into two sections: 'Inferred Dimension Members' and 'When data for the inferred member is loaded, you can update the existing record rather than create'. Under 'Inferred Dimension Members', there is a checkbox labeled 'Enable inferred member support'. Below this, there are two radio buttons: 'All columns with a change type are null' (selected) and 'Use a Boolean column to indicate whether the current record is an inferred member'. Under 'When data for the inferred member is loaded, you can update the existing record rather than create', there is a text box labeled 'Inferred member indicator:'. At the bottom of the window, there are four buttons: 'Help', '< Back', 'Next >', and 'Finish >>|'. The 'Next >' button is highlighted with a blue border.

İşlemlerimiz bittiğinde Finish seçeneği ile sonlarız.



Slowly Changing Dimension için işlemlerimiz buraya kadar bu yapı sayesinde kaynaktaki değişen ve eklenen kayıtları hedef db 'de güncelleme ve ekleme işlemlerini yapacaktır. Ancak kaynaktaki silinen kayıtlar için herhangi bir işlem yapmamaktadır bunun için farklı bir yapı kuracağız.

#### 4- SİLİNER KAYITLAR:

Kaynakta silinen kayıtları hedef db miz olan DWH db 'de silme işlemi için şu adımları uygulayacağız.

- 1- Kaynakta silinen kayıtları trigger ile deleted tablosuna yazdıracakız.
- 2- DWH db 'de sadece bu kayıtlar için bir tablo oluşturacağız.
- 3- Kaynakta deleted tablosundaki bilgileri DWH db de oluşturduğumuz tabloya aktaracağız.
- 4- Bu kayıtları DWH db de sileceğiz.

##### 1.Silinen Kayıtlar Tablosu:

Kaynak db de silinen kayıtların ID bilgisinin tutulduğu bir tablo oluştururuz.

```
CREATE TABLE [dbo].[CUSTOMER_DELETED](  
    [CUID] nvarchar(36) NOT NULL  
)
```

Bizim için önemli olan alan CUID alanıdır bu alan tekil olması gerekmektedir.

Silinen kayıtları trigger ile tablomuza yazdıralım.

```
CREATE TRIGGER [dbo].[Deleted_Customer] on [dbo].[CUSTOMER]  
AFTER DELETE  
AS BEGIN  
  
DECLARE  
    @Cuid nvarchar(36)  
    select @Cuid=CUID  
    from deleted  
  
    insert into [CUSTOMER_DELETED] (CUID) values (@Cuid)  
  
END
```

Trigger ile silinen kayıtlarımızı artık deleted tablomuza yazdırıyoruz.

##### 2.DWH db 'de Deleted Tablosu:

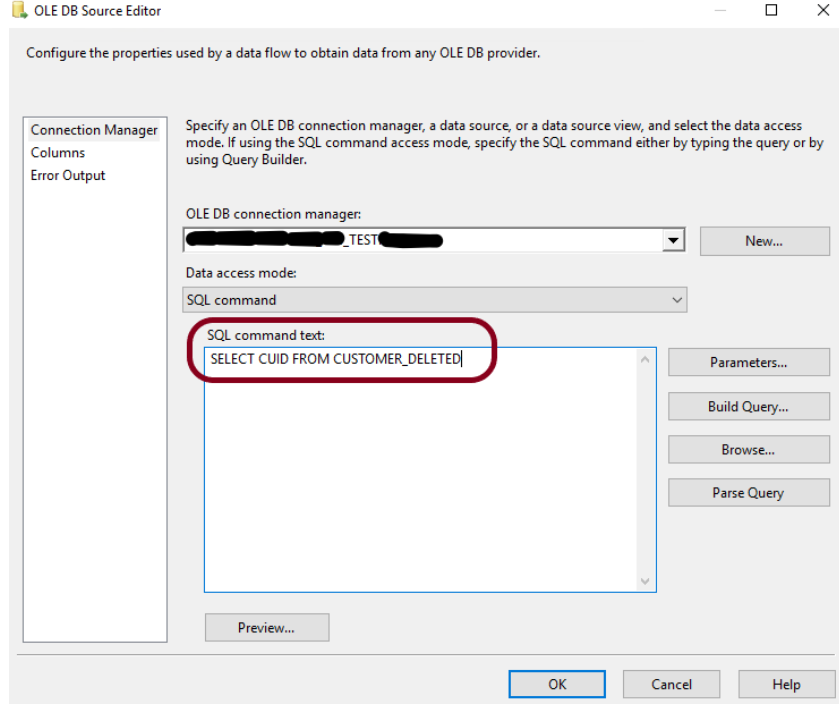
Hedef DB miz olan DWH db 'ye bu kayıtları aktarmak için bir tablo oluştururuz.

```
CREATE TABLE [dbo].[CUID](  
    [CUID] [nvarchar](36) NULL  
) ON [PRIMARY]  
GO
```

### 3.Silinen Kayıtları Aktarma:

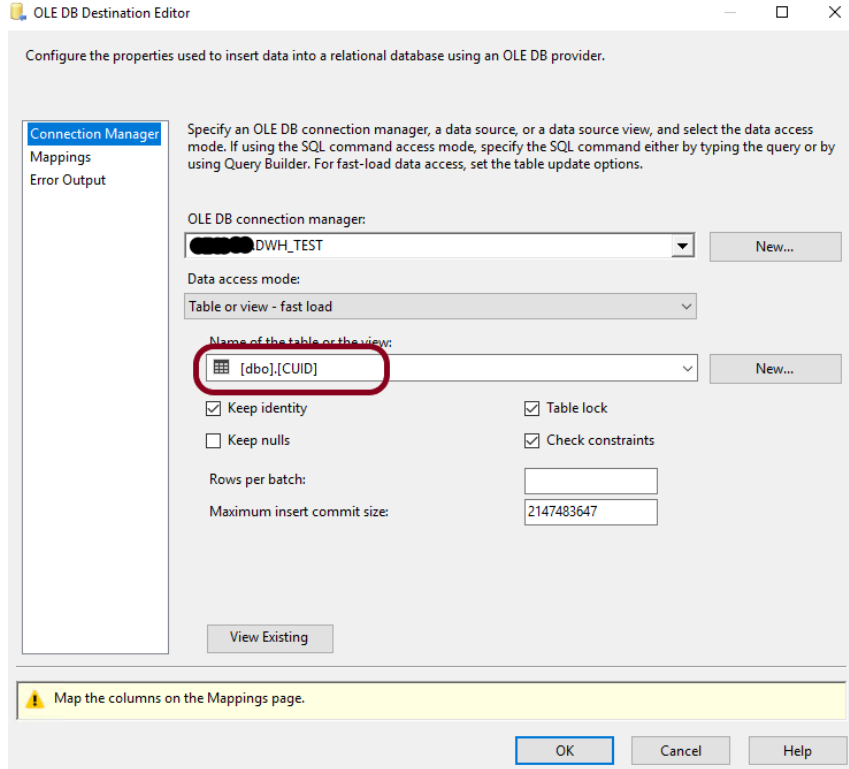
SSIS paketimizde AKTARIM adı verdiğimiz küçük tabloların aktarımı için oluşturduğumuz Data Flow Task içerisine gelerek yeni bir Source ve Destination oluştururuz.

Source işleminde kaynaktan veri okumak için CUSTOMER\_DELETED tablomdan sadece ilgili alanım olan CUID alanını okuturum.



The OLE DB Source Editor dialog box is shown. It has a left sidebar with 'Connection Manager', 'Columns', and 'Error Output'. The main area is titled 'Configure the properties used by a data flow to obtain data from any OLE DB provider.' It contains a dropdown for 'OLE DB connection manager' with a value ending in '\_TEST' and a 'New...' button. Below it is a 'Data access mode' dropdown set to 'SQL command'. The 'SQL command text' field is highlighted with a red circle and contains the text 'SELECT CUID FROM CUSTOMER\_DELETED'. To the right of this field are buttons for 'Parameters...', 'Build Query...', 'Browse...', and 'Parse Query'. At the bottom are 'Preview...', 'OK', 'Cancel', and 'Help' buttons.

Destination işlemimde DWH db için oluşturduğumuz deleted tablomuzaya yazdırırız.



The OLE DB Destination Editor dialog box is shown. It has a left sidebar with 'Connection Manager', 'Mappings', and 'Error Output'. The main area is titled 'Configure the properties used to insert data into a relational database using an OLE DB provider.' It contains a dropdown for 'OLE DB connection manager' with a value ending in 'DWH\_TEST' and a 'New...' button. Below it is a 'Data access mode' dropdown set to 'Table or view - fast load'. The 'Name of the table or the view' field is highlighted with a red circle and contains the text '[dbo].[CUID]'. To the right of this field is a 'New...' button. Below this are checkboxes for 'Keep identity', 'Keep nulls', 'Table lock', and 'Check constraints'. There are also input fields for 'Rows per batch' and 'Maximum insert commit size'. At the bottom is a 'View Existing' button. A yellow warning bar at the bottom says 'Map the columns on the Mappings page.' At the very bottom are 'OK', 'Cancel', and 'Help' buttons.

Mapleme işlemimiz bittikten sonra artık silinen kayıtların tekil anahtarlarını DWH db 'ye aktarmış olduk.

#### 4. Kaynakta Silinen Kayıtları DWH db 'de Silme:

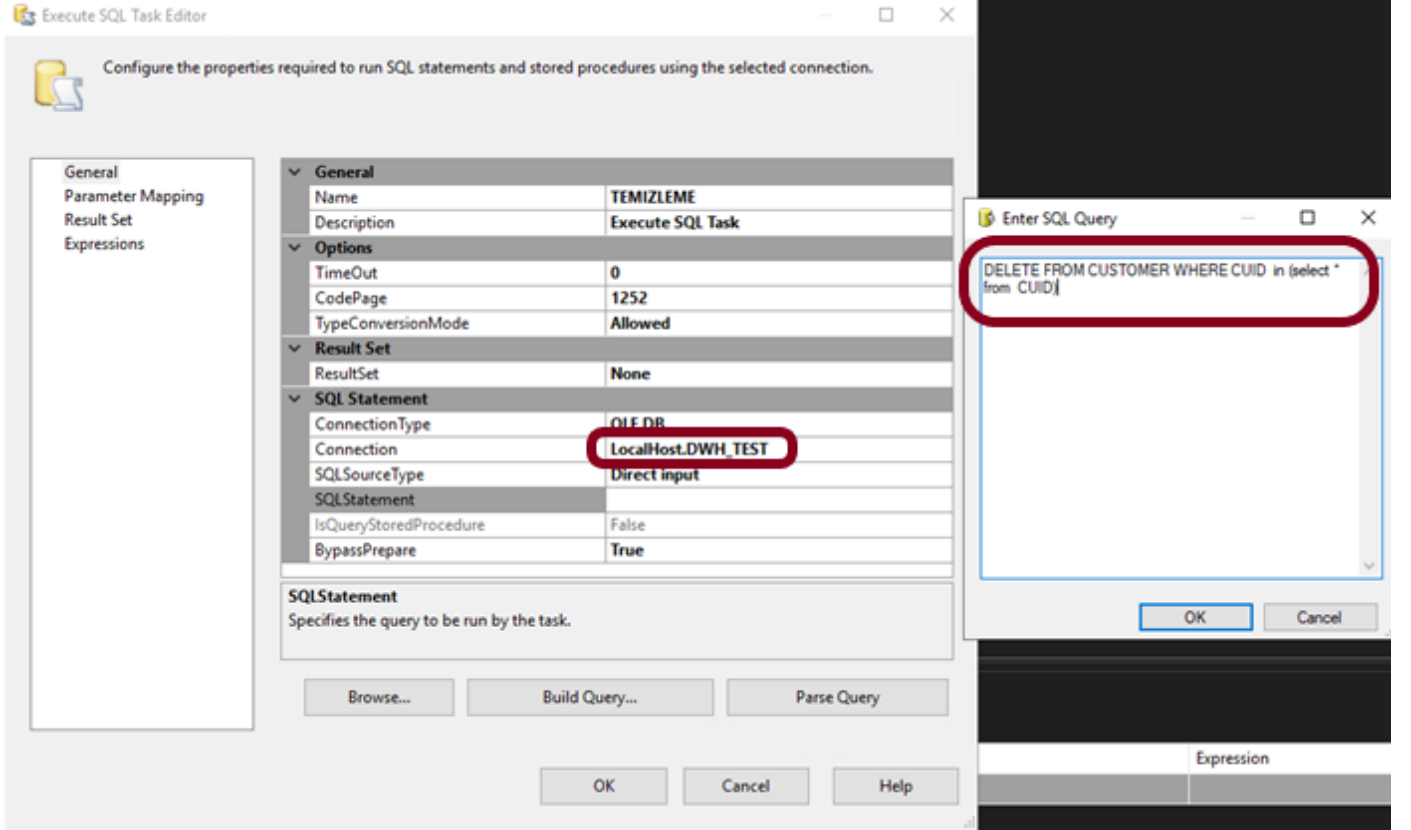
Kaynaktan hedefe aktardığımız silinen kayıtların tekil bilgileri ile DWH db üzerinden bu kayıtları silme işlemi yapan sorgumuzu yazarız.

Bunun için SSIS 'de Data Flow Task üzerinden Execute SQL Task kullanacağız. Bu işleme TEMİZLEME adı vereceğim.

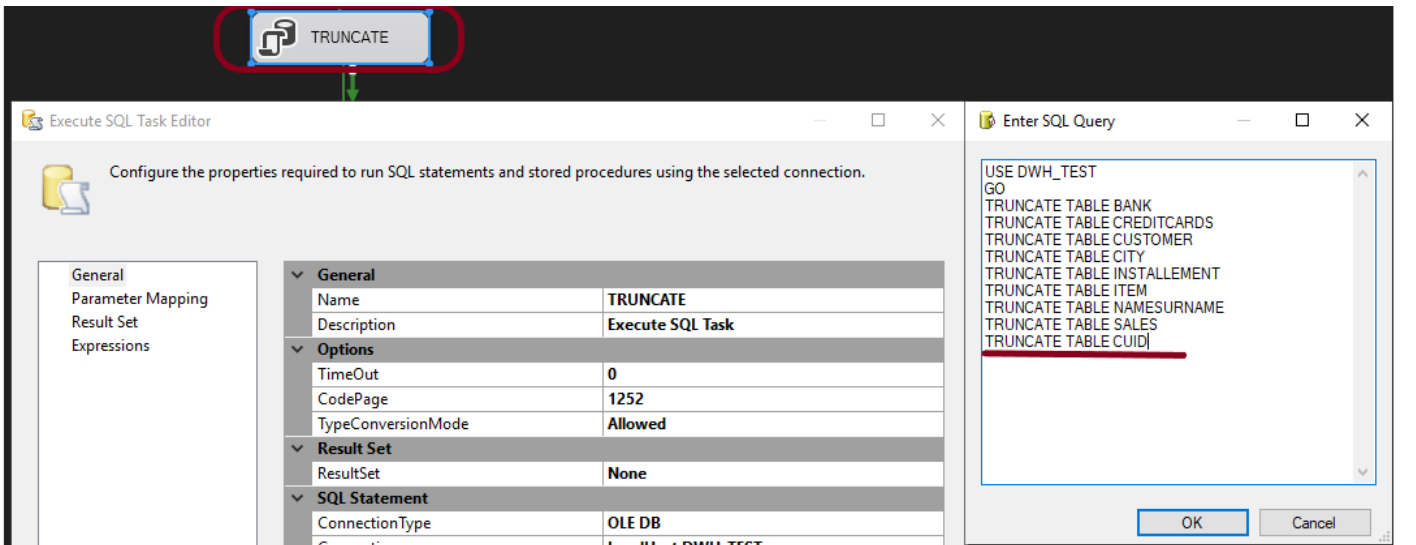
Connection olarak hedef db olan DWH db 'yi seçerim.

SQL Statement alanına silme işlemi için gerekli SQL kodumu yazarım.

(DELETE FROM CUSTOMER WHERE CUID in (select \* from CUID))

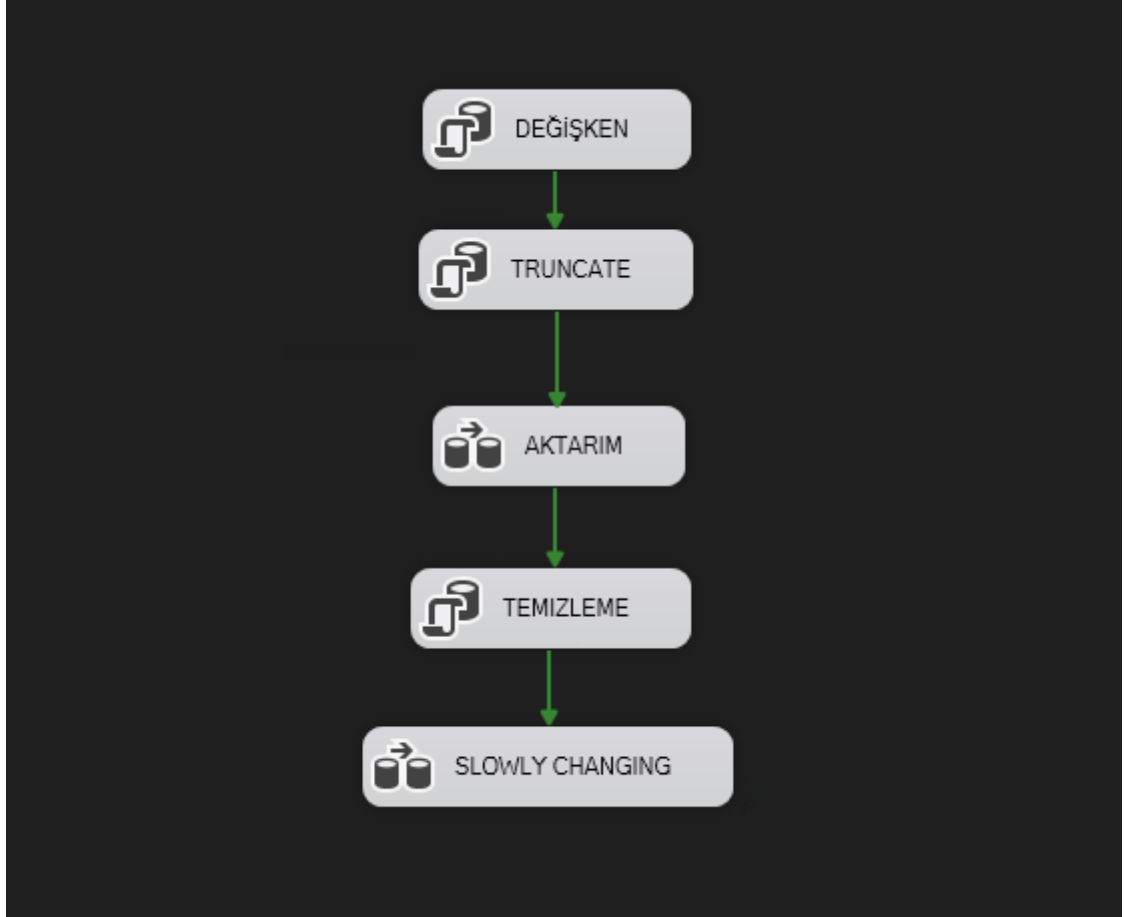


Bu işlemimiz bittiğinde DWH db de oluşturduğumuz tabloların temizlenmesi için TRUNCATE adı verdiğimiz Execute SQL Task in içerisine deleted tablosunu sorgumuzun içerisine ekleriz.



### SON:

SSIS 'de DWH kurulumu için yapacağımız işlemler buraya kadardır. Kurmuş olduğumuz bu yapıda kaynaktan verileri alıp hedef db 'ye yazdırdık büyük tablolarımız için SLOWLY CHANGING mimarisi kullanarak değişen ve eklenen kayıtları getiririz. Silinen kayıtlar için Kanyak db de trigger ile silinenlerin tekil anahtarı olan ID bilgisini bir tabloda tutarak Hedef db ye yazdırırız ve bu kayıtların hedefte silinmesini sağlayan sorgumuzu yazarız.

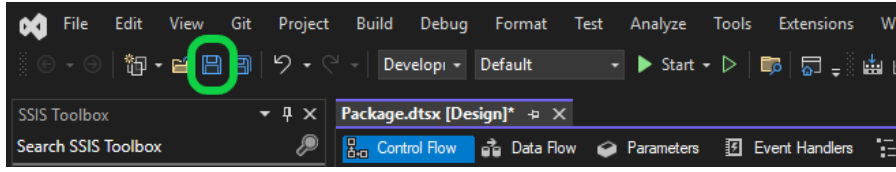


Yapımızın son hali bu şekilde olacaktır.

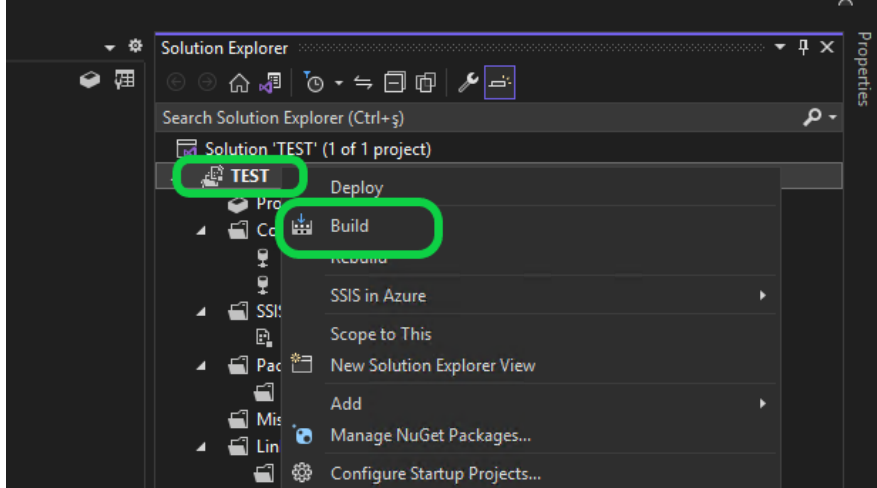


## SSIS PROJE KAYDETME-BUILD-DEPLOY İŞLEMLERİ:

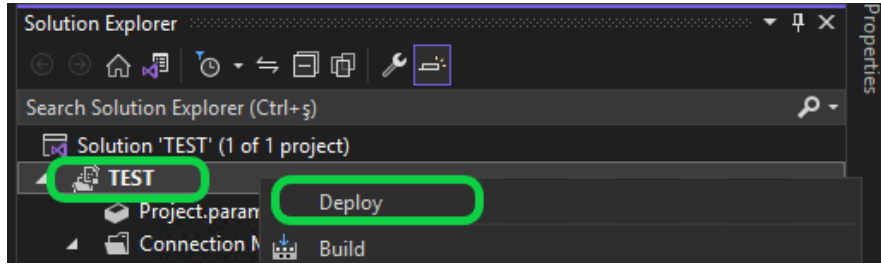
SSIS projemizi kaydetmiş için CTRL+S ya da sağ üst kısımda bulunan kaydetme butonu ile kaydedebiliriz.



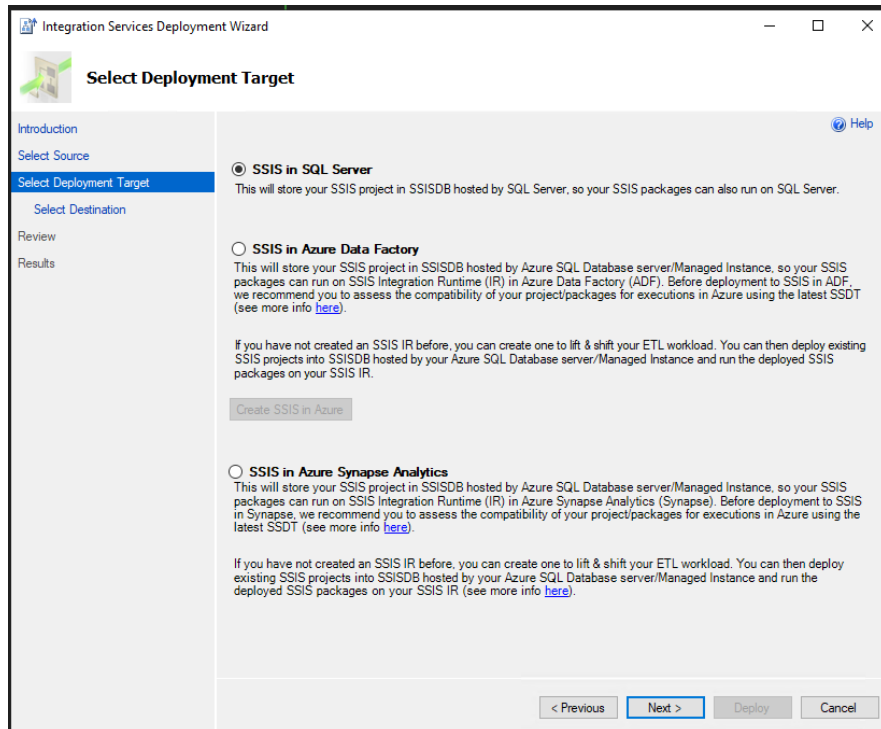
Build işlem için Solution Explorer'dan Projemize sağ tıklayarak build işlemimizi seçeriz.



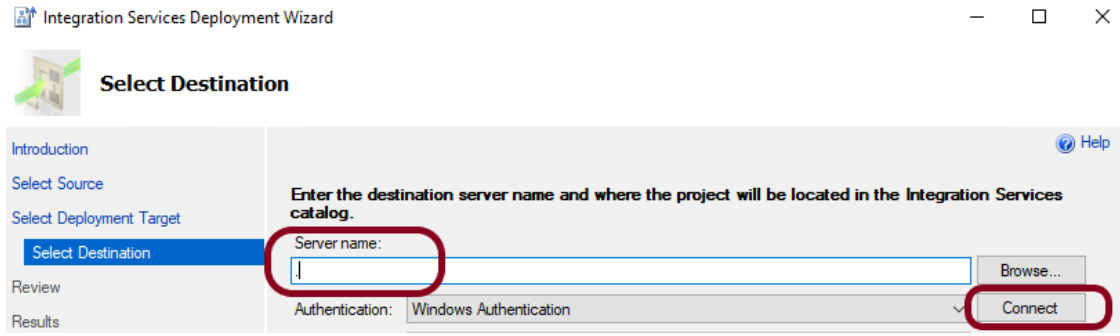
Deploy işlemi için Solution Explorer'dan Projemize sağ tıklayarak deploy işlemimizi seçeriz.



Açılan pencerede SSIS in SQL Server seçeneğini seçeriz ve next ile devam ederiz.



Select Destination sayfasında Server name alanına “.” ya da server name bilgimi yazdıktan sonra connect işlemi yaparız.



Integration Services Deployment Wizard

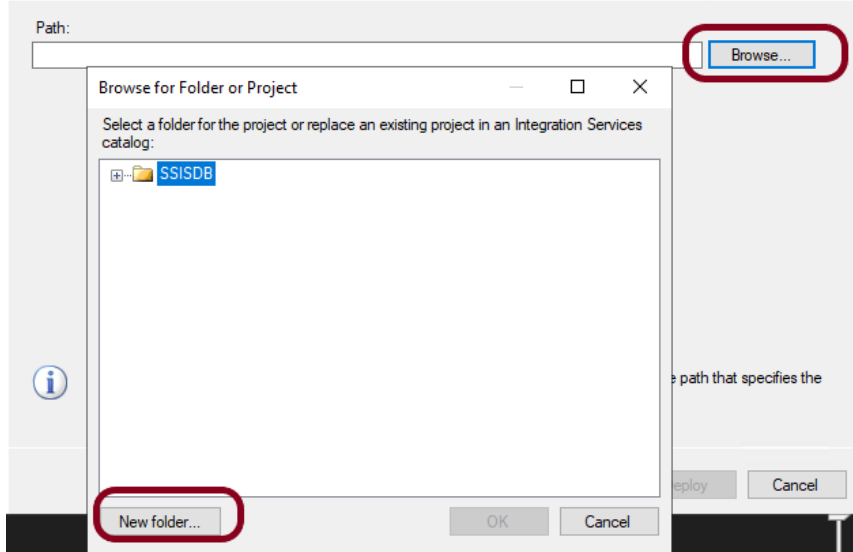
Select Destination

Enter the destination server name and where the project will be located in the Integration Services catalog.

Server name: [ ] Browse...

Authentication: Windows Authentication Connect

Path bölümünden browse ‘a tıklayarak deploy işlemini yapacağımız klasörümüzü seçeriz eğer klasörümüz yoksa New folder ile yeni klasör oluştururuz.



Path: [ ] Browse...

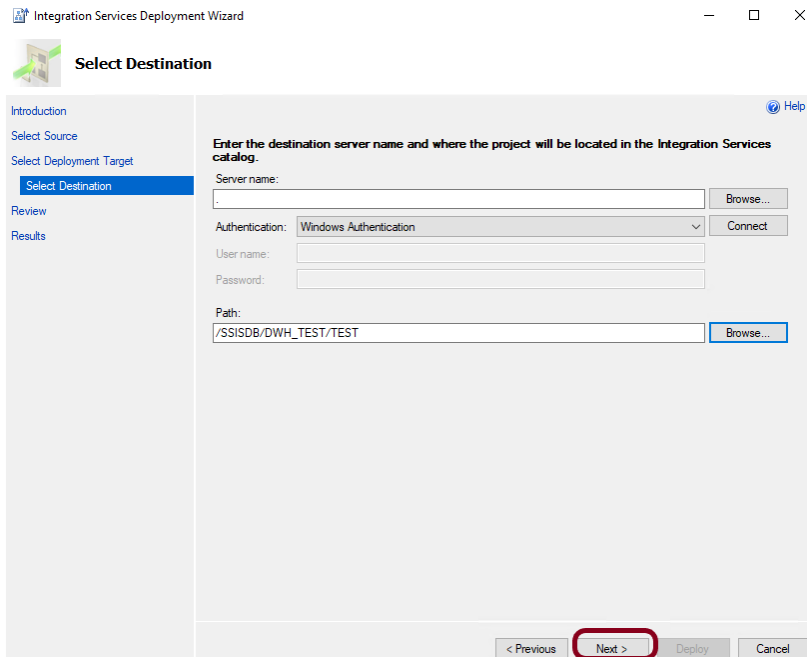
Browse for Folder or Project

Select a folder for the project or replace an existing project in an Integration Services catalog:

SSISDB

New folder... OK Cancel

Klasörümüzü seçtikten sonra next ile devam ederiz.



Integration Services Deployment Wizard

Select Destination

Enter the destination server name and where the project will be located in the Integration Services catalog.

Server name: [ ] Browse...

Authentication: Windows Authentication Connect

User name: [ ]

Password: [ ]

Path: /SSISDB/DWH\_TEST/TEST Browse...

< Previous Next > Deploy Cancel

Çıkan pencerede deploy u seçerek projemizi deploy etmiş oluruz.

Bu işlemleri yapma amacımız SQL üzerinden SSIS projemizle ilgili oluşturacağımız joblar ‘da projemizi kullanmamızı sağlamaktadır.

**HAZIRLAYAN : SERKAN COŞKUN**