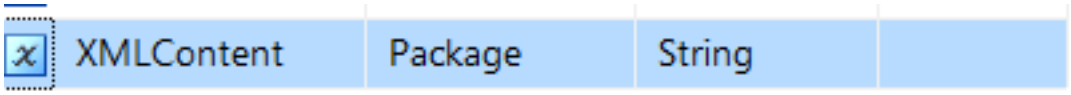
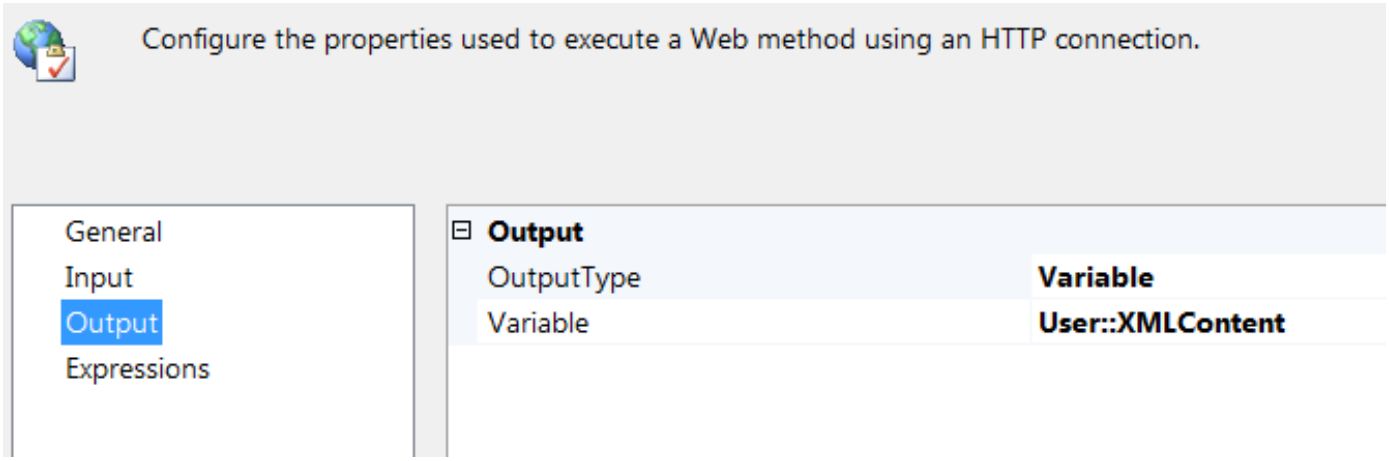
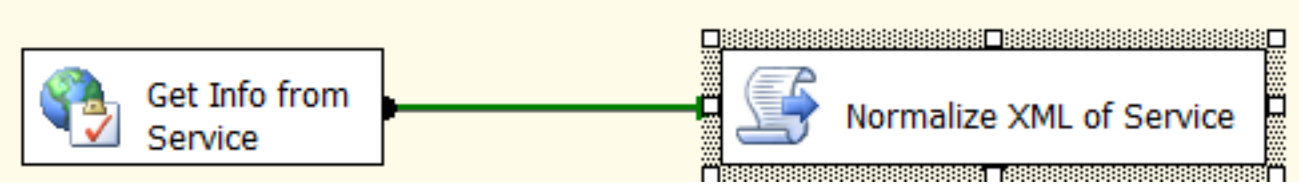
عنوان: **نحوه انتقال اطلاعات استخراج شده از وب سرویس به SQL Server به کمک SSIS**
نویسنده: محمد باقر سیف اللهی
تاریخ: ۱۳۹۲/۰۲/۳۰ ۲۲:۱۵
آدرس: www.dotnettips.info
گروه‌ها: SQL Server, Web Service, SSIS, data mining

ممکن است در مواقعی نیاز به اطلاعات استخراج شده از وب سرویسی داشته باشید که در همان مقطع زمانی به آن دسترسی ندارید . مسلما برای این منظور باید آن اطلاعات را ذخیره کرده تا در صورت نیاز بتوان به آنها رجوع کرد . یکی از راه حل‌ها ذخیره آن در پایگاه داده (در اینجا Sql Server) است که در این پست به کمک امکانات BIDS در پکیج‌های SSIS و کوئری‌های SQL این مشکل را برطرف میکنیم. برای مشاهده نحوه استخراج اطلاعات از وب سرویس به اینجا مراجعه کنید .

تنها تفاوتی که در این پست در کار با سرویس با پست اشاره شده در بالا وجود دارد ذخیره اطلاعات استخراج شده است که در آن پست در یک فایل xml ذخیره شدند ولی در اینجا ما نیاز داریم تا اطلاعات را در یک متغیر با حوزه کاری Package ذخیره کنیم (به این معنی که مختص به همان flow نباشد و در تمام پکیج دیده شود)



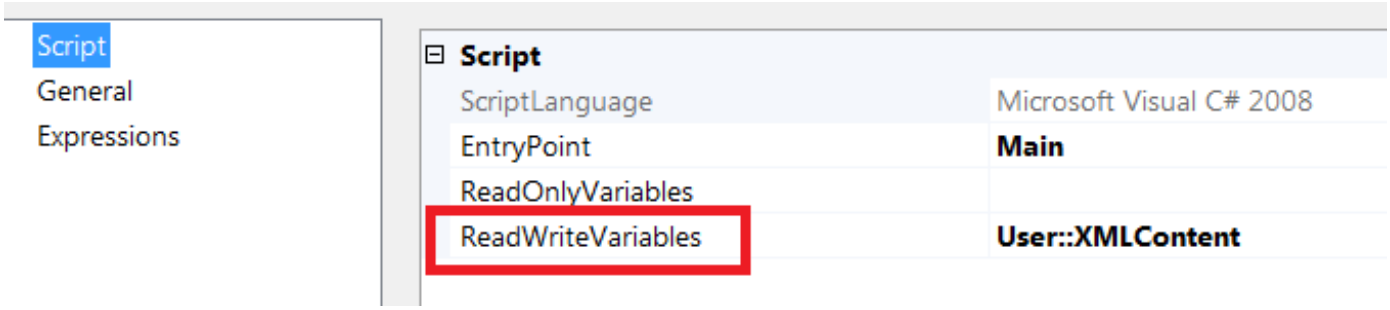




به دلیل اینکه هدر xml خروجی از سرویس دارای چندین namespace هست هنگام کار با آنها به مشکل خواهیم خورد. (هم هنگام کار با xml taskها و هم هنگام کار با xml در sql)

```
▼<ArrayOfUserInfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

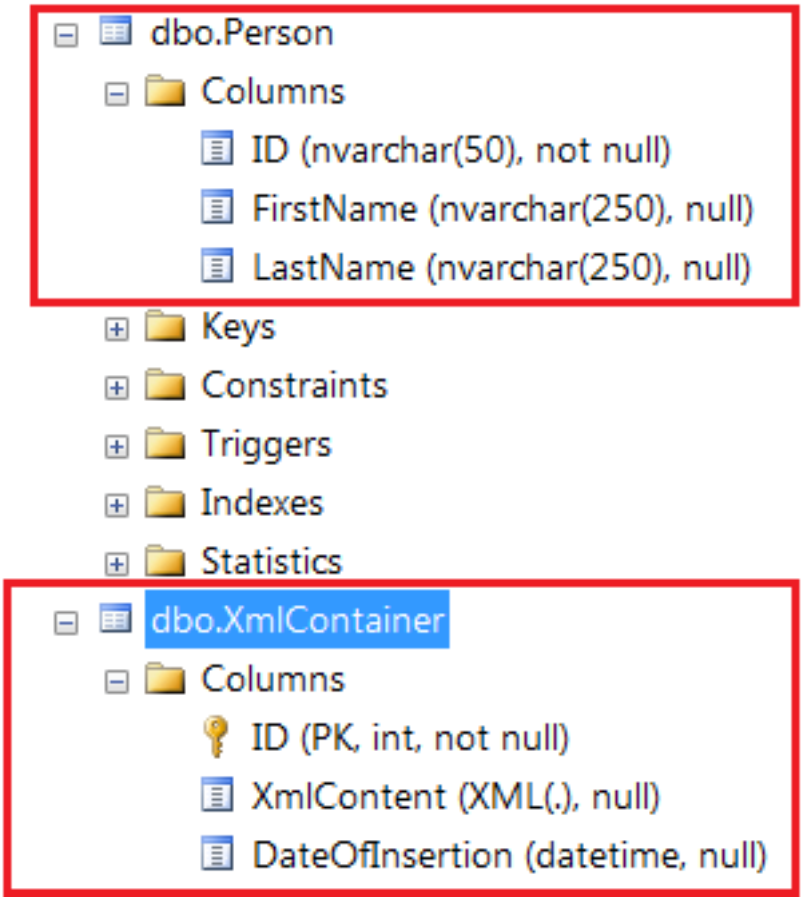
به همین دلیل باید این قسمت از محتوا را حذف کرد . برای همین پس از گرفتن اطلاعات از سرویس آن را به کمک یک Script task حذف می‌کنیم




```
public void Main()
{
    //xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://tempuri.org/"
    object o = "null";
    byte[] emptyBytes = new byte[0];
    try
    {
        Dts.VariableDispenser.LockForRead("User::XMLContent");
        Variable xmlc = Dts.Variables["User::XMLContent"];
        o = xmlc.Value;
        if (o != null)
        {
            string s = o.ToString();
            s = s.Replace("xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"", "");
            s = s.Replace("xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\"", "");
            s = s.Replace("xmlns=\"http://tempuri.org/\"", "");
            Dts.VariableDispenser.Reset();
            Dts.VariableDispenser.LockForWrite("User::XMLContent");
            Dts.Variables["User::XMLContent"].Value = s;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error SSIS : " + Environment.NewLine + Environment.NewLine + ex.Message, "Error on Package...");
        Dts.TaskResult = (int)ScriptResults.Failure;
    }
    finally
    {
        Dts.VariableDispenser.Reset();
    }

    Dts.Log("nameSpace Removed Successfully...", 0, emptyBytes);
    Dts.TaskResult = (int)ScriptResults.Success;
}
```

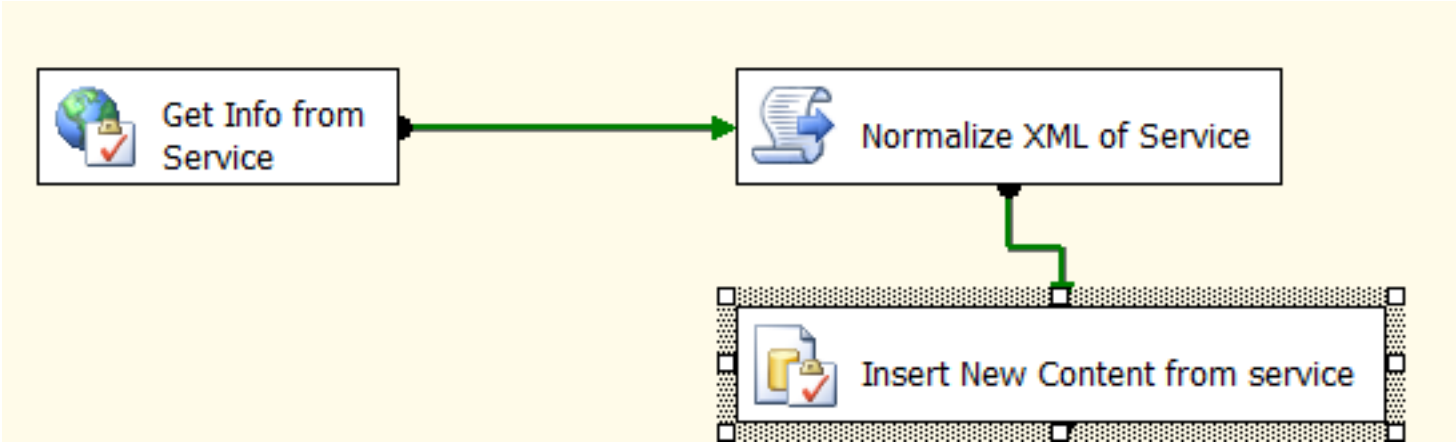
در این مرحله اطلاعات استخراج شده را باید در SQL درج کنیم . برای همین ساختاری که باید اطلاعات را در SQL نگه دارد را در دیتابیس ایجاد می‌کنیم :

جدول person برای نگهداری اطلاعات سرویس و XmlContainer برای نگهداری xmlهای سرویس .(برای داشتن History)

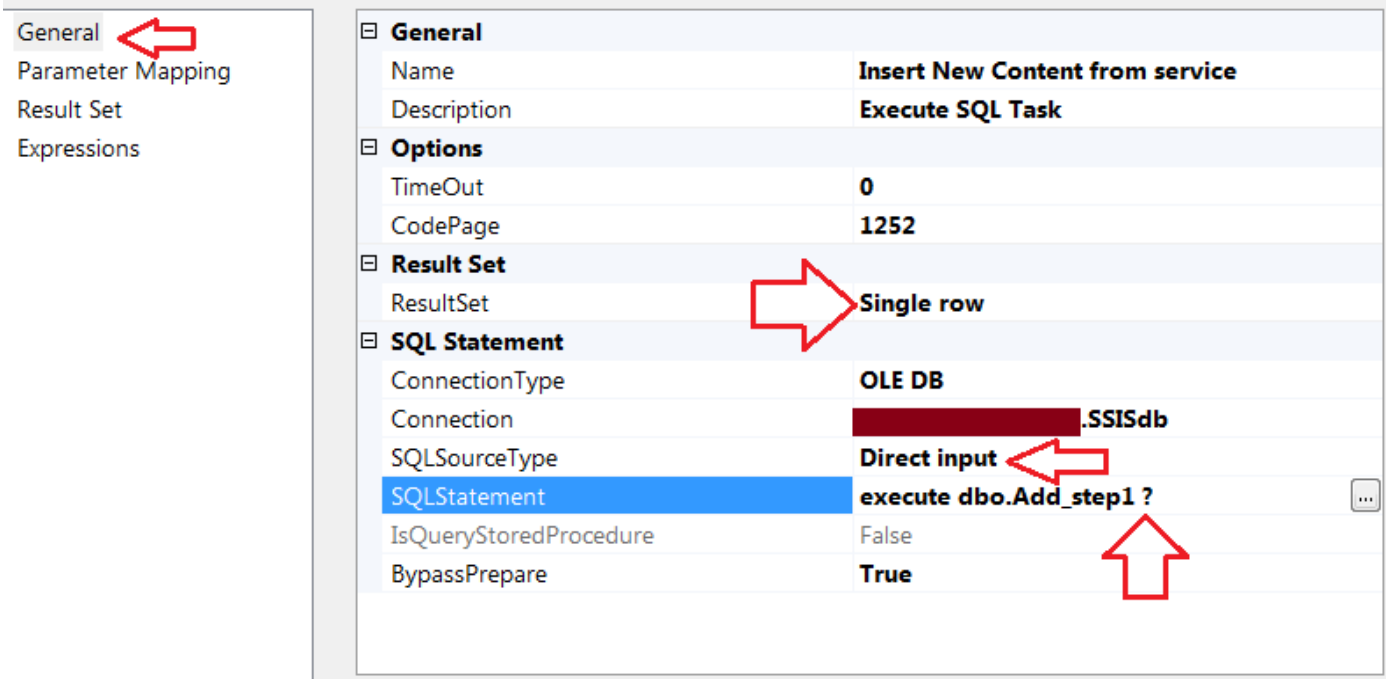


برای درج هم از SP استفاده می‌کنیم :

```
ALTER PROCEDURE [dbo].[Add_step1]

    @xml XML
AS
BEGIN
    SET NOCOUNT ON;

INSERT INTO dbo.XmlContainer
values( @xml, GETDATE())

DECLARE @id INT
SET @id = SCOPE_IDENTITY() ;

SELECT  @id AS Result


END
```

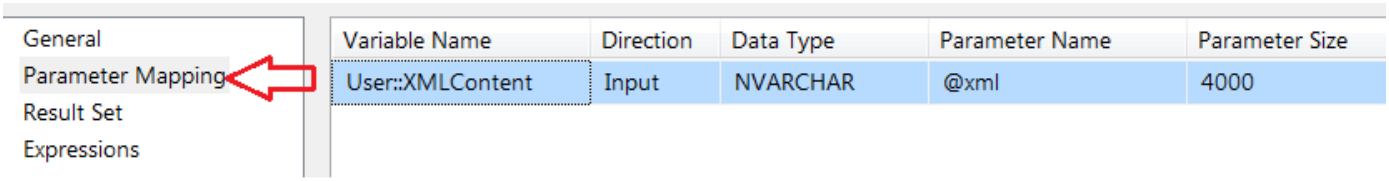


و پیکربندی SQL Task :

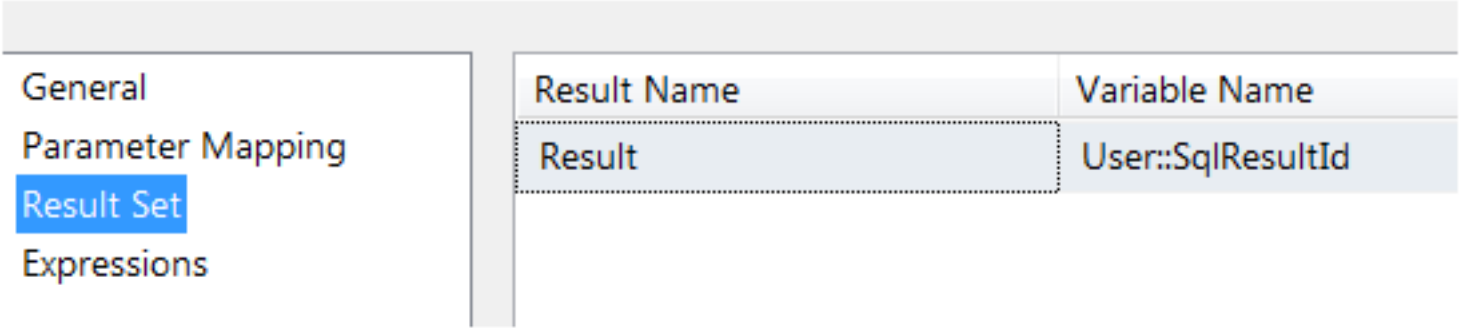


نکته اول ایجاد کانکشن به پایگاه داده است که در اینجا از نمایش جزییات آن صرف نظر شده است . نکته دیگری پارامتر SP است که چون یک پارامتر دارد یک علامت سوال قرار میگیرد . اگر چند پارامتر بود به صورت علامت‌های سوال با ویرگول از هم جدا می‌شوند . ?,?,?,?

سپس در بخش parameter mapping به ترتیب مقدار دهی می‌شوند :



و مقدار خروجی SP که شناسه آیتم درج شده است را در یک متغیر نگهداری می‌کنیم :



برای قدم بعد می‌خواهیم اطلاعات موجود در XML استخراج شده در پایگاه داده را در جدول مربوطه ذخیره کنیم . برای این کار این SP را می‌نویسیم :

```
ALTER PROCEDURE [dbo].[transform_Step2]
 @id int
AS
BEGIN

    SET NOCOUNT ON;
    DECLARE @trans CHAR(50),@xml XML
    SET @trans = 'MyTrans'
    BEGIN TRY
    BEGIN TRANSACTION @trans

    DELETE FROM Person -- empty table
    SELECT @xml = xc.XmlContent FROM XmlContainer xc
    WHERE xc.ID = @id
    DECLARE  @dt DATETIME , @hoc int;
    EXEC sp_xml_preparedocument @hoc OUTPUT, @xml
    INSERT INTO Person (ID,FirstName,LastName)
    SELECT ID, FirstName, LastName
    FROM OPENXML(@hoc, 'ArrayOfUserInfo/UserInfo')
    WITH
    (
    ID [nvarchar](50) 'ID',
    FirstName [nvarchar](250) 'FirstName',
    LastName [nvarchar](250) 'LastName'
    )
    EXEC sp_xml_removedocument @hoc -- empty cache
    IF (@@ERROR <> 0 )
    BEGIN
        ROLLBACK TRANSACTION @trans
    END
    ELSE
    BEGIN
    COMMIT TRANSACTION @trans
     SELECT 'OK' AS  'ERROR'
    END

    END TRY
    begin CATCH
        ROLLBACK TRANSACTION @trans
        SELECT  ERROR_MESSAGE()  AS  'ERROR'
    END CATCH
END
```

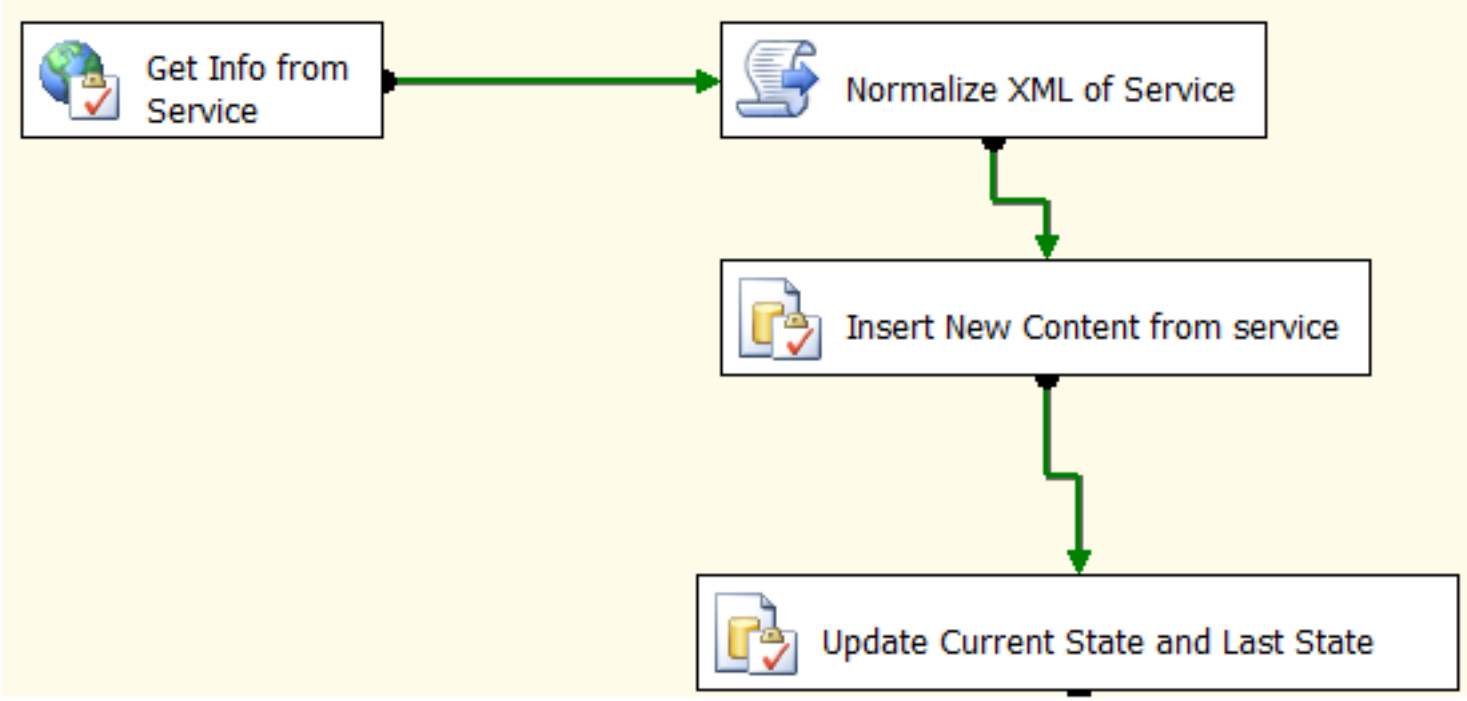
نکات مهم موارد زیر هستند :

1 - استفاده از OpenXml برای parse کردن xml

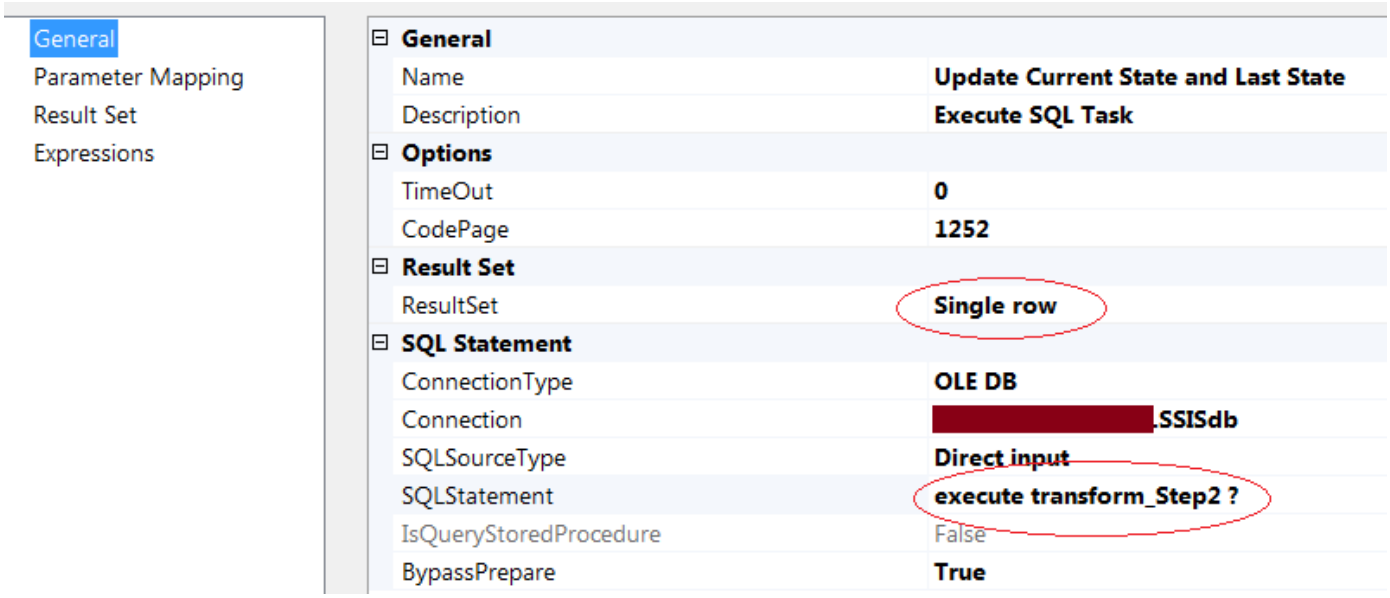
2 - استفاده از sp سیستمی sp_xml_preparedocument و sp_xml_removedocument (بیشتر)

3- اطلاعات شناسه و نام و نام خانوادگی innerهای نودهای xml هستند . اگر این موارد به صورت attribute باشند باید قبل از نام آنها علامت @ قرار بگیرند.

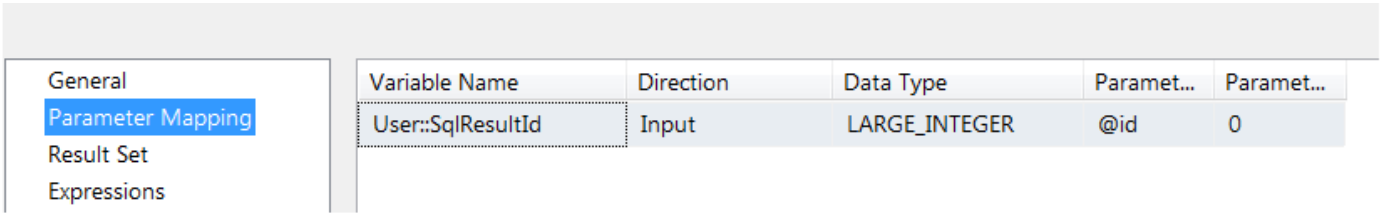
پس از ایجاد این Sp باید آن را در package فراخوانی کنیم :



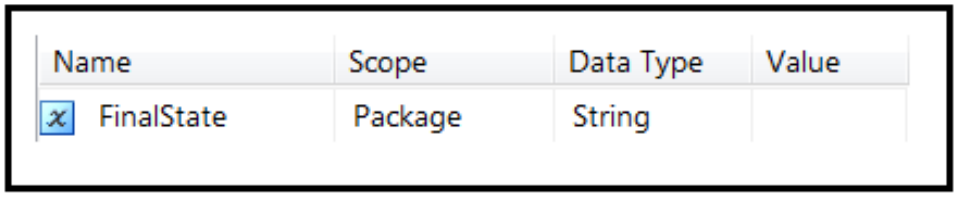
و پیکربندی این SQL Task :

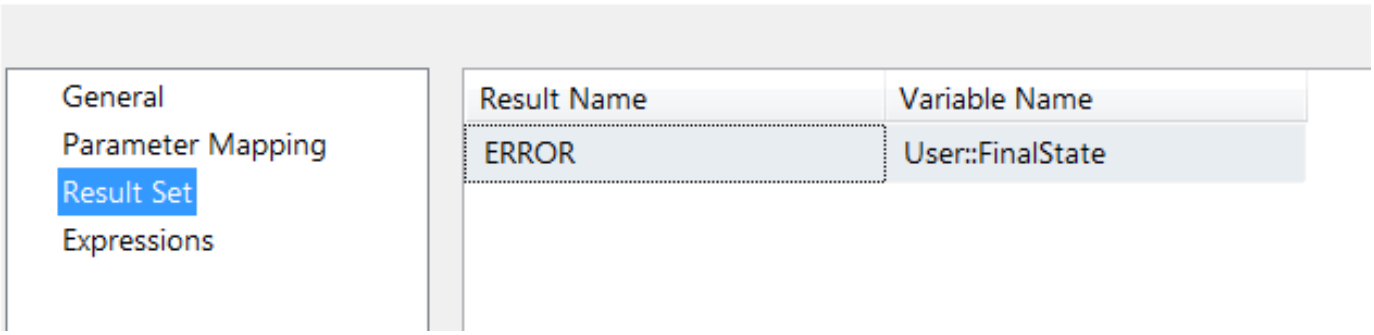


و پارامترهای این SP :

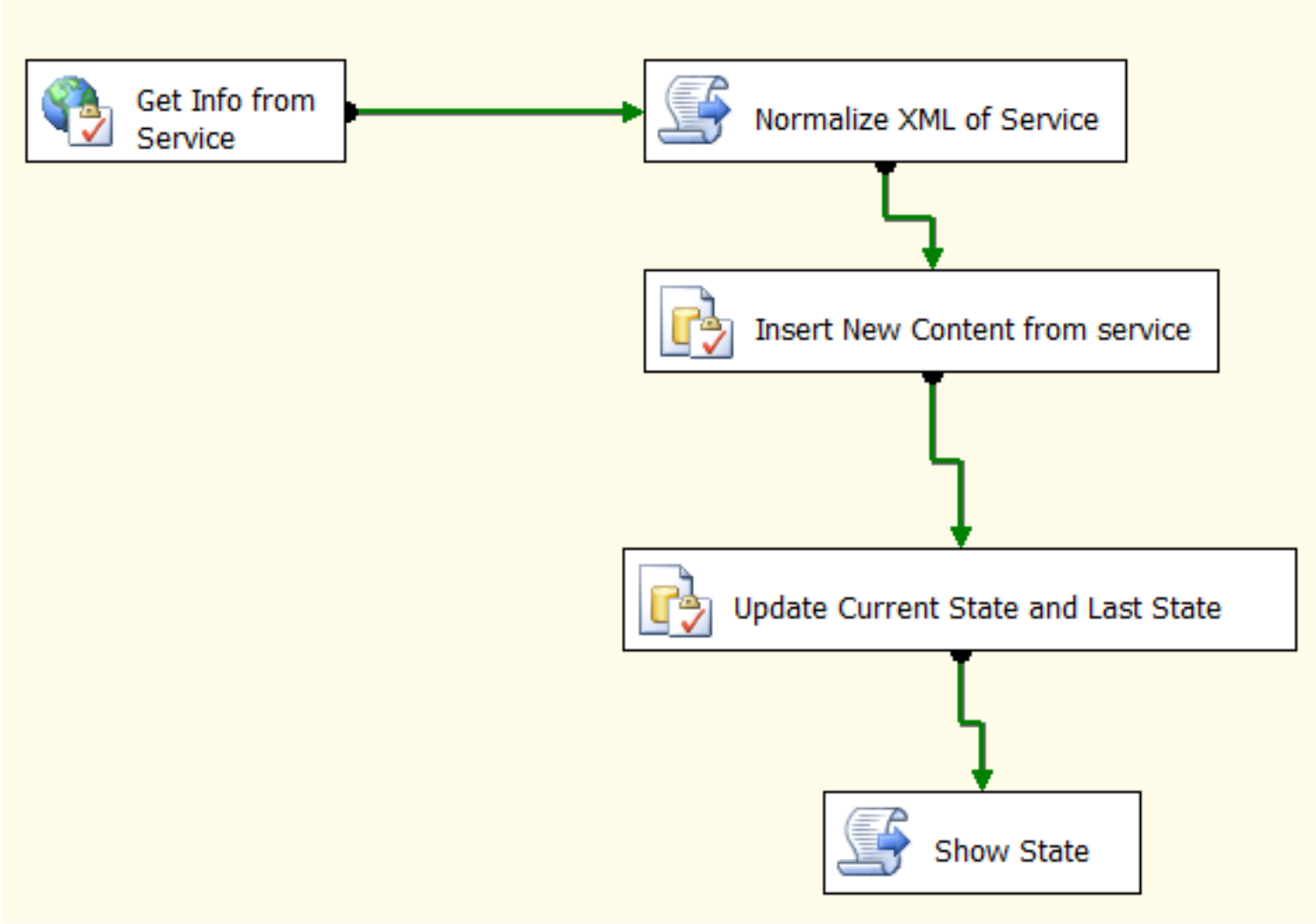


و ذخیره نتیجه تراکنش در متغیری در پکیج :

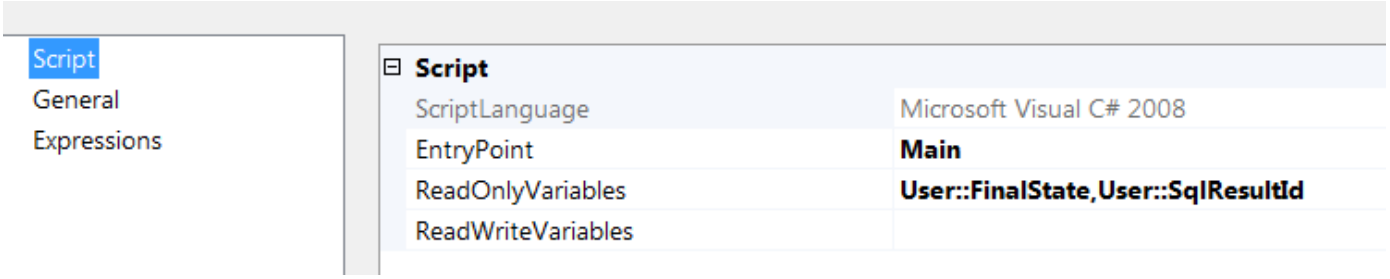




و اکنون پکیج ما ظاهری شبیه به این مورد خواهد داشت :

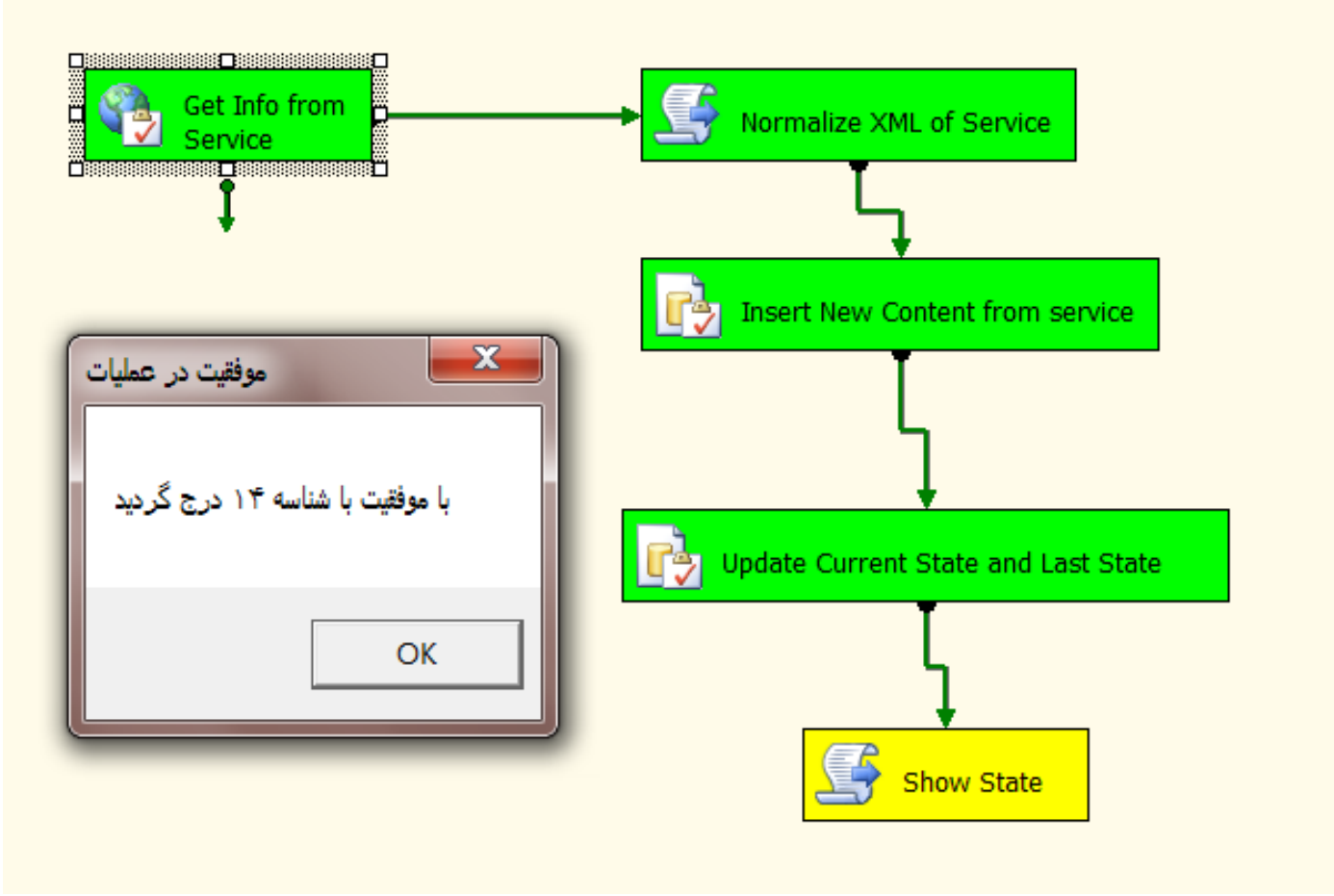


در نهایت به عنوان یک facility می‌توانیم وضعیت تراکنش را به کاربر نمایش دهیم (به کمک Script Task) :



```
public void Main()
{

    const string _id = "User::SqlResultId";
    const string _err = "User::FinalState";

    Dts.VariableDispenser.LockForRead(_id);
    Dts.VariableDispenser.LockForRead(_err);


    string id = Convert.ToString( Dts.Variables[_id].Value);
    string error = Convert.ToString(Dts.Variables[_err].Value);

    string msg = "با موفقیت با شناسه "+id+" درج گردید";
    Dts.VariableDispenser.Reset();

    if (error.Equals("OK"))
    {
        MessageBox.Show(msg,"موفقیت در عملیات");
        Dts.TaskResult = (int)ScriptResults.Success;
    }
    else
    {
        MessageBox.Show("Error : "+Environment.NewLine+error, "خطا در انجام عملیات");
        Dts.TaskResult = (int)ScriptResults.Failure;
    }


    // TODO: Add your code here
}
```

و تمام ...



Results | Messages

	ID	XmlContent	DateOfInsertion
1	7	<ArrayOfUserInfo><UserInfo><ID>bd36d158-f2bd-40d...	2013-05-16 21:36:44.84
2	8	<ArrayOfUserInfo><UserInfo><ID>5848557b-6482-4d8...	2013-05-16 21:44:21.49
3	9	<ArrayOfUserInfo><UserInfo><ID>5f465898-bce8-4153...	2013-05-16 21:45:02.61
4	10	<ArrayOfUserInfo><UserInfo><ID>0836a55b-3d8f-4ce...	2013-05-16 21:53:19.83
5	11	<ArrayOfUserInfo><UserInfo><ID>73670cd5-1d09-4a9...	2013-05-16 22:35:52.70
6	12	<ArrayOfUserInfo><UserInfo><ID>316e06a0-4063-479...	2013-05-16 22:36:30.89
7	13	<ArrayOfUserInfo><UserInfo><ID>c56ab672-9415-418...	2013-05-16 22:38:56.86
8	14	<ArrayOfUserInfo><UserInfo><ID>c991b4ee-a389-427...	2013-05-17 10:55:12.27

موفق باشید

نظرات خوانندگان

نویسنده: Amir
تاریخ: ۱۳۹۲/۱۰/۱۸ ۰:۴۴

با سلام وسپاس از مطالب مفیدتون

سوالی دارم ممنون میشم اگه جواب بدید

برای انتقال داده‌های هشت سرور که از نوع پارادوکس به SQL Server 2012 هستند آیا به غیر از روش SSIS راه دیگری وجود دارد ؟ ناگفته نماند که داده‌ها حجم قابل توجهی دارند و با روش SSIS سرعت شبکه را کاهش می‌دهد.

نکته دیگر، داده‌ها به صورت کامل به SQL Server منتقل نمیشوند و بعضی جداول منتقل شده یا خالی هستند یا جداول تکراری داریم در SQL Server که در نتیجه باعث قابل اطمینان نبودن Data Warehouse میشود.

با سپاس

نویسنده: محمد باقر سیف اللهی
تاریخ: ۱۳۹۲/۱۰/۱۸ ۱۲:۲۹

سلام
اجازه بدید اول به این نکته اشاره کنم که SSIS روش نیست بلکه یک ابزاره. روش مد نظر برای Migration داده ، ETL نام داره (Extract , Transform , Load) نه SSIS.
حالا برای انجام عمل ETL ابزارهای دیگری هم میشه پیدا کرد مثل informatica یا DB2 Infosphere که می‌تونید تعدادی از اون‌های رو اینجا ببینید
در مورد مشکلاتی که در باره داده‌ها یا شبکه می‌فرمایید هم باید یک plan مناسب برای این مهاجرت داده ای ایجاد کنید . جهت اطلاع عرض کنم که این پروسه ممکنه گاها تا یک ماه طول بکشه بسته به پلنی که طراحی شده و شرایط و محدودیت هایی که موجوده. اطلاعات بیشتر رو می‌تونید از اینجا کسب کنید
موفق باشید

نویسنده: Amiir
تاریخ: ۱۳۹۲/۱۰/۱۸ ۱۳:۴۵

ممنون از راهنمایتون

عنوان: **تعریف انبار داده Data Warehouse**
نویسنده: محمد رجبی
تاریخ: ۱۳۹۳/۰۲/۱۵ ۱۰:۳۰
آدرس: www.dotnettips.info
گروه‌ها: data mining, data warehouse

در این مقاله در ادامه‌ی مطلبی که تحت عنوان «آموزش مفاهیم Data Warehouse» توسط آقای شاه قلی منتشر شده بود، به بررسی بیشتر مفهوم انبار داده (Data Warehouse) پرداخته می‌شود.

مقدمه

در سازمان ها، داده‌ها و اطلاعات معمولاً به دو شکل در سیستم‌ها پیاده سازی می‌گردد:

- **سیستم‌های عملیاتی OLTP:**

این سیستم‌ها باعث می‌گردند تا چرخ کسب و کار بگردد. وجود این سیستم‌ها سبب می‌شود تا داده‌های مربوط به کسب و کار، به بانک اطلاعاتی وارد شوند. این سیستم‌ها عموماً:

o به دلیل کوتاهی عملیات دارای سرعت قابل توجهی می‌باشند.
o محیطی جهت ورود داده‌ها می‌باشند.
o معمولاً اپراتورها، استفاده کننده‌های آن هستند.

- **سیستم‌های اطلاعاتی DSS، DW/BI ، OLAP :**

این سیستم‌ها باعث می‌گردند تا چرخش کسب و کار را بنگرید. فلسفه بکارگیری این سیستم‌ها در سازمان این است که اطلاعات مورد نیاز مدیران، از درون داده‌های سیستم‌های عملیاتی موجود، استخراج گردد. این سیستم‌ها عموماً:

o به دلیل آنالیز حجم انبوهی از داده ها، معمولاً کندتر از سیستم‌های عملیاتی می‌باشند.
o محیطی جهت تولید گزارشات تحلیلی و آماری می‌باشند.
o معمولاً مدیران و تصمیم گیرندگان سازمان ها، استفاده کنندگان آن می‌باشند.

سیستم‌های عملیاتی در جامعه ما سابقه بیشتری داشته و متخصصین فناوری اطلاعات عموماً با طراحی و تولید چنین سیستم هایی آشنایی کافی دارند. متاسفانه جایگاه سیستم‌های اطلاعاتی در جامعه ما کمتر شناخته شده و متخصصین فناوری اطلاعات بندرت با مفاهیم و نحوه پیاده سازی آن آشنایی دارند.
این نکته حائز اهمیت است که سیستم‌های اطلاعاتی یک سیستم یا محصول نیستند که بتوان آنها را خریداری کرد. بلکه یک راهبرد (Solution, Approach) هستند و در حقیقت هر راهبردی مربوط به یک نوع کسب و کار (Business) و یا سازمان می‌باشد و نمی‌توان فرمول واحدی را برای حتی سازمان‌های مشابه، ارائه نمود.

گارتنر در ابتدای سال 2011 گزارشی را منتشر کرده که نشان میدهد بازار BI با 9.7 % رشد، ارزشی بالغ بر 10.8 بیلیون دلار داشته، ولی متاسفانه پروژه‌های آن به طور متوسط با 75% شکست مواجه شده است. در حالیکه 4 سال پیش، این رقم حدود 50% بود. این موسسه BI را پنجمین اولویت مدیران IT ذکر کرده است.

مفاهیم و مباحث مربوط به Data Warehouse به اواسط دهه 1980 برمی گردد، به زمانی که IBM تحقیقاتی را در این زمینه شروع کرد و نتیجه آنرا «Information Warehouse» نامید و هنوز هم در برخی منابع از این واژه بجای Data Warehouse استفاده می‌شود. از این پس برای راحتی از اختصار DW بجای Data Warehouse استفاده می‌شود. انبارهای داده جهت رفع نیاز رو به رشد مدیریت داده‌ها و اطلاعات سازمانی که توسط پایگاه‌های داده سیستم‌های عملیاتی غیر ممکن بود، ساخته شدند.

انبار داده به مجموعه ای از داده‌ها گفته می‌شود که از منابع مختلف اطلاعاتی سازمان جمع آوری، دسته بندی و ذخیره می‌شود. در واقع یک انبار داده مخزن اصلی کلیه داده‌های حال و گذشته یک سازمان می‌باشد که برای همیشه جهت انجام عملیات گزارش گیری و آنالیز در دسترس مدیران می‌باشد. انباره‌های داده حاوی داده هایی هستند که به مرور زمان از سیستم‌های عملیاتی آنلاین سازمان، استخراج می‌شوند. بنابراین سوابق کلیه اطلاعات و یا بخش عظیمی از آنها را می‌توان در انباره داده‌ها مشاهده نمود.
از آنجائیکه انجام عملیات آماری و گزارشات پیچیده دارای بار کاری بسیار سنگینی برای سرورهای پایگاه داده می‌باشند، وجود انبار داده سبب می‌گردد که این گونه عملیات تاثیری بر فعالیت برنامه‌های کاربردی سازمان نداشته باشد.

همانگونه که پایگاه داده سیستم‌های عملیاتی سازمان (برنامه‌های کاربردی) به گونه ای طراحی می‌شوند که انجام تغییر، حذف و اضافه داده به سرعت صورت پذیرد، در مقابل انبار داده‌ها دارای معماری ویژه ای می‌باشند که موجب تسریع انجام عملیات آماری و گزارش گیری می‌شود. در حقیقت می‌توان اینگونه بیان نمود که انباره داده یک مخزن فعال و هوشمند از اطلاعات است که قادر است اطلاعات را از محیط‌های گوناگون جمع آوری و مدیریت کرده و نهایتا پخش نماید و در صورت لزوم نیز سیاست‌های تجاری را روی آنها اجرا نماید.

:Bill Inmon

او را پدر DW می‌نامند، از دیدگاه او DW هسته مرکزی چیزی است که او آنرا CIF اختصار (Corporate Information Factory) می‌نامد، که پایه و اساس BI بر مبنای آن قرار دارد. وی از طرفداران Top-Down Design می‌باشد که معتقد است در زمان طراحی باید با دیدی سازمانی، CIF را مدل سازی، ولی بصورت دپارتمانی پیاده سازی کرد (Think Globally, Implement Locally). در این نوع طراحی از DW به Data Mart خواهیم رسید.

:Ralph Kimball Ph.D

به نظر وی DW چیزی نیست جز یک کپی از داده‌های عملیاتی که به طرز خاصی برای گزارشات و تحلیل‌های آماری، آماده و ساختمند شده است. به بیان دیگر DW سیستمی است جهت استخراج، پالایش، تطبیق و تحویل اطلاعات منابع داده ای به یک بانک اطلاعاتی Dimensional و اجرای Query و گزارشات آماری و تحلیلی برای اهداف تصمیم گیری و استراتژیک سازمان.
وی معرفی کننده یکی از اساسی‌ترین مفاهیم طراحی یعنی Dimensional Modeling است؛ ماحصل چنین ایده ای، اساس شکل گیری مدلی است که امروزه کارشناسان آنرا به نام Cube می‌شناسند. وی از طرفداران Bottom-Up Design است که در این نگرش از Data Mart به DW می‌رسیم. این روش به نظر عملی‌تر از روشی می‌باشد که به یکباره DW جامع و کامل برای اهداف سازمانی طراحی و پیاده سازی گردد.

تعریف انبار داده :

W.H.Inmon پدر DW آنرا چنین تعریف می‌کند:

```
The Data Warehouse is a collection of
```

Integrated

```
,
```

Subject-Oriented

```
databases designed to support the DSS function, where each unit of data is
```

Non-Volatile

```
and
```

relevant

```
to some moment in
```

Time

از تعریف فوق دو مورد دیگر نیز به طور ضمنی استنباط می‌شود:

o انبار داده به طور فیزیکی، کاملاً جدا از سایر سیستم‌های عملیاتی است.
o داده‌های DW مجموعه ای Aggregated و Atomic از داده‌های تراکنش‌های سیستم‌های عملیاتی است که سوای کاربرد آنها در سیستم‌های عملیاتی، برای مقاصد مدیریتی نیز استفاده خواهد شد.

به بیان دیگر DW راهبردی است که دسترسی آسان به اطلاعات درست (Right Information)، در زمانی درست (Right Time) ، به کاربران درست (Right Users)، را فراهم می‌آورد تا «تصمیم گیری سازمانی» قابل انجام باشد. DW صرفاً یک محصول نرم افزاری و

یا سخت افزاری نیست که بتوان آنرا خریداری نمود بلکه فراتر از آن و در حقیقت یک محیط پردازشی می‌باشد که کاربران می‌توانند از درون آن اطلاعات مورد نیاز خود را بیابند.
DW اطلاعات خود را از سایر بانک‌های اطلاعاتی از نوع OLTP و یا سایر DWهای لایه پایین‌تر و به صورت دسته ای (Batch) و یا انبوه (Bulk Loading) جمع آوری می‌کند. یک DW به صورت سنتی باید شامل داده‌های Historic سازمان باشد و می‌توان اینگونه بیان نمود که در DW هرچه داده‌های قدیمی‌تری موجود باشد، اعتبار تحلیل‌های آماری سیستم افزایش خواهد یافت.

داده‌های سیستم عملیاتی را نمی‌توان بلافاصله درون بانک اطلاعاتی DW لود نمود، چنین داده هایی باید آماده سازی، پالایش و همگون گردند تا شرایط لود در DW را داشته باشند. حداقل کاری که انتظار داریم یک DW در مورد داده‌ها برای ما برآورده سازد شامل موارد زیر است:

- o استخراج داده‌ها از منابع مختلف (مبدإ)
- o تبدیل داده‌ها به فرمتی یکسان
- o لود داده‌ها به جداول مربوطه (مقصد)

با هر با اجرای پروسه فوق یکی از سه مورد زیر، بسته به نیاز طراحی و محدودیت‌های تکنولوژی رخ خواهد داد:

- o تمام داده‌ها در DW با داده‌های جدید جایگزین خواهند گردید(Full Load, Initial Load, Full Refresh).
- o داده‌های جدید به داده‌های موجود اضافه خواهند گردید Incremental Load (Inserted data).
- o نسخه جدیدی از داده‌های کنونی به سیستم اضافه خواهند گردید Incremental Load (Updated data).

ویژگی‌های داده‌های درون DW

داده‌های DW از نگاه Inmon دارای 4 ویژگی اصلی زیر هستند:

o فقط خواندنی (Non-Volatile):

هیچ رکوردی و یا داده ای Update نخواهد شد و صرفاً رکوردهایی که محتوای مقادیر جدید داده‌ها هستند، به سیستم اضافه خواهند شد.

o موضوع گرا (Subject-Oriented):

منظور از «موضوع» پایه‌های اساسی یک کسب و کار هستند، به شکلی که با حذف یکی از این پایه ها، شاید ماهیت آن کسب و کار از ریشه دگرگون شود. برای مثال موضوعاتی چون «مشتری» و یا «بیمه نامه» برای شرکت‌های بیمه.

o جامع (Integrated):

باید تمامی کدهایی که در سیستم‌های عملیاتی وجود دارند و معانی یکسانی دارند، برای مثال کد جنسیت، در DW به یک روش ذخیره و نمایش داده شوند.

o زمانگرا (Time Variant):

هر رکورد باید حاوی فیلد و یا کلیدی باشد که نمایانگر این باشد که این رکورد در چه زمانی ایجاد، استخراج و ذخیره شده است. از آنجا که داده‌های درون سیستم‌های عملیاتی آخرین و به روزترین داده هر سیستم میباشد، نیازی به وجود چنین عنصری در سیستم‌های OLTP احساس نمی‌گردد، ولی چون در DW تمام داده‌های نسخ قدیمی داده‌های سیستم‌های عملیاتی موجود می‌باشد، باید حتماً مشخص گردد که هر داده ای در سیستم‌های عملیاتی در چه زمانی، چه مقادیری داشته است. این عنصر زمانی کمک می‌کند تا بتوانیم:

- o گذشته را آنالیز کنیم.
- o اطلاعات مربوط به حال حاضر را بدست آوریم.
- o آینده را پیش بینی کنیم.

منبع: کتاب آقای خشایار جام سحر با عنوان بانک داده تجمیعی

Comparison Kimball vs. Inmon Inmon

```
Continuous & Discrete Dimension Management
```

```
Define data management via dates in your data
```

Continuous time

When is a record active

Start and end dates

Discrete time

A point in time

Snapshot

Kimball

Slowly Changing Dimension Management

Define data management via versioning

Type I

Change record as required

No History

Type II

Manage all changes

History is recorded

Type III

Some history is parallel

Limit to defined history

Inmon	Kimball
Subject-Oriented Integrated Non-Volatile Time-Variant	Business-Process-Oriented Stresses Dimensional Model, Not E-R
Top-Down	Bottom-Up and Evolutionary
Integration Achieved via an Assumed Enterprise Data Model	Integration Achieved via Conformed Dimensions
Characterizes Data marts as Aggregates	Star Schemas Enforce Query Semantics

	Inmon	Kimball
Overall approach	Top-down	Bottom-up
Architectural structure	Enterprise-wide DW feeds departmental DBs	Data marts model a business process;enterprise is achieved with conformed dims
Complexity of method	Quite complex	Fairly simple
Data orientation	Subject or data driven	Process oriented
Tools	Traditional ERDs and DIS	Dimensional modeling; departs from traditional relational modeling
End user accessibility	Low	High
Timeframe	Continuous & Discrete	Slowly Changing
Methods	Timestamps	Dimension keys

عنوان: **بررسی داده کاوی و OLAP**
نویسنده: محمد رجبی
تاریخ: ۱۳۹۳/۰۲/۱۶ ۱۰:۳۵
آدرس: www.dotnettips.info
گروه‌ها: data mining, OLAP, SSAS

بررسی OLAP واژه OLAP در اوایل سال‌های 1990 شکل گرفت. E.F.Codd بنیانگذار مدل داده‌ی رابطه‌ای، این واژه را در فرهنگ نامه کاربران بانک‌های اطلاعاتی توصیف نمود.
مشابه یک بانک اطلاعاتی رابطه‌ای که شامل تعدادی جدول می‌باشد، یک بانک اطلاعاتی OLAP شامل تعدادی Cube است. هر Cube مجموعه ای از Dimensionها و Measure هاست. Dimension یک شیء تحلیلی است که محورهای مختصات را برای پرسش‌های تحلیلی تعریف می‌کند و از Member هایی تشکیل شده است که Member هر Dimension در قالب سلسله مراتب می‌تواند تعریف شود؛ در حالیکه Measure یک مقدار عددی است که در مختصات Cube تعریف می‌شود که این مقادیر از جداول تراکنشی بدست می‌آید (جدول Fact) که جزئیات هر رکورد تراکنشی در آنها ذخیره می‌شود. Measureها حاوی اطلاعاتی هستند که از پیش، محاسبات تجمیعی بر روی آنها براساس سلسله مراتب تعریف شده در Dimension انجام شده است.
ساختار OLAP شبیه به یک مکعب روبیک از داده‌ها است که می‌توان آنرا در جهات مختلف چرخانید تا بتوان سناریوهای «قبلا چه شده» و «چه می‌شد اگر ...» را بررسی نمود. مدل چند بعدی OLAP طریقه نمایش دادن داده‌ها را در مقایسه با بانک‌های اطلاعاتی رابطه‌ای تسهیل می‌کند. غالبا OLAP داده‌ها را از یک انباره داده استخراج می‌کند.

ابزارهای OLAP را به چند دسته تقسیم می‌کنند:

OLAP رو میزی: ابزارهای ساده و مستقل که روی کامپیوترهای شخصی نصب شده و مکعب‌های کوچکی می‌سازند و آنها را نیز بر روی سیستم به شکل فایل ذخیره می‌کنند. بیشتر این ابزارها با صفحات گسترده ای نظیر Excel کار می‌کنند. به این ترتیب کسانی که در سفر هستند قادر به استفاده از این دسته از محصولات هستند. (در حال حاضر Web OLAP در حال جایگزین کردن این محصولات است)

MOLAP : بجای ذخیره کردن اطلاعات در رکوردهای کلید دار، این دسته از ابزارها، بانک‌های اطلاعاتی خاصی را برای خود طراحی کرده‌اند؛ بطوری که داده‌ها را به شکل آرایه‌های مرتب شده بر اساس ابعاد داده ذخیره می‌کنند. در حال حاضر نیز دو استاندارد برای این نوع ابزار وجود دارد. سرعت این ابزار بالا و سایز بانک اطلاعاتی آن نسبتا کوچک است.

ROLAP : این ابزارها با ایجاد یک بستر روی بانک‌های رابطه‌ای اطلاعات را ذخیره و بازیابی می‌کنند. بطوری که اساس بهینه سازی برخی بانک‌های مانند MicreoStrategy، Red Brick و ... بر همین اساس استوار است. اندازه بانک اطلاعاتی این ابزار قابل توجه می‌باشد.

HOLAP : در اینجا منظور از hybrid ترکیبی از MOLAP و ROLAP است. ابزار دارای بانک اطلاعاتی بزرگ و راندمان بالاتر نسبت به ROLAP می‌باشد.

مقایسه گزینه‌های ذخیره سازی در OLAP:

MOLAP: این نوع ذخیره‌سازی بیشترین کاربرد در ذخیره اطلاعات را دارد. همچنین به صورت پیش فرض جهت ذخیره‌سازی اطلاعات انتخاب شده است. در این نوع تنها زمانی داده‌های منتقل شده به Cube به روز می‌شوند که Cube پردازش شود و این امر باعث تاخیر بالا در پردازش و انتقال داده‌ها می‌شود.

ROLAP: در ذخیره‌سازی ROLAP زمان انتقال بالا نیست که از مزایای این نوع ذخیره‌سازی نسبت به MOLAP است. در ROLAP اطلاعات و پیش‌محاسبه‌ها در یک حالت رابطه‌ای ذخیره می‌شوند و این به معنای زمان انتقال نزدیک به صفر میان منبع داده (بانک اطلاعاتی رابطه‌ای) و Cube می‌باشد. از معایب این روش می‌توان به کارایی پایین آن اشاره کرد زیرا زمان پاسخ برای پرس‌وجوهای اجرا شده توسط کاربران طولانی است. دلیل این کارایی پایین بکار نبردن تکنیک‌های ذخیره‌سازی چند بعدی است.

HOLAP : این نوع ذخیره‌سازی چیزی مابین دو حالت قبلی است. ذخیره اطلاعات با روش ROLAP انجام می‌شود، بنابراین زمان انتقال تقزیبا صفر است. از طرفی برای بالابردن کارایی، پیش‌محاسبه‌ها به صورت MOLAP انجام می‌گیرد در این حالت SSAS آماده است تا تغییری در اطلاعات مبداء رخ دهد و زمانی که تغییرات را ثبت کرد نوبت به پردازش مجدد پیش‌محاسبه‌ها می‌شود. با این نوع ذخیره‌سازی زمان انتقال داده‌ها به Cube را نزدیک به صفر و زمان پاسخ برای اجرای کوئری‌های کاربر را زمانی بین نوع ROLAP و MOLAP می‌رسانیم.

این سه روش ذخیره‌سازی انعطاف‌پذیری مورد نیاز را برای اجرای پروژه فراهم می‌کند. انتخاب هر یک از این روش‌ها به نوع پروژه، حجم داده‌ها و ... بستگی دارد. در پایان می‌توان نتیجه گرفت که بهتر است زمان پردازش طولانی‌تری داشته باشیم تا اینکه کاربر نهایی در هنگام ایجاد گزارشات زمان زیادی را منتظر بماند.

بررسی داده کاوی

حجم زیاد اطلاعات، مدیران مجموعه‌ها را در تحلیل و یافتن اطلاعات مفید دچار چالش کرده است. داده کاوی، ابزار مناسب برای تجزیه و تحلیل اطلاعات و کشف و استخراج روابط پنهان در مجموعه‌های داده‌ای سنگین را فراهم می‌کند. گروه مشاوره‌ای گارتنر داده کاوی را استخراج نیمه اتوماتیک الگوها، تغییرات، وابستگی‌ها، نابهنجاری‌ها و دیگر ساختارهای معنی دار آماری از پایگاه‌های بزرگ داده تعریف می‌کند. داده کاوی، تلاشی برای یافتن قوانین، الگوها و یا میل احتمالی داده به مُدلی، در بین انبوهی از داده‌ها است.

داده کاوی فرآیندی پیچیده جهت شناسایی الگوها و مدل‌های صحیح، جدید و به صورت بالقوه مفید، در حجم وسیعی از داده می‌باشد؛ به طریقی که این الگوها و مدلها برای انسانها قابل درک باشند. داده کاوی به صورت یک محصول قابل خریداری نمی‌باشد، بلکه یک رشته علمی و فرآیندی است که بایستی به صورت یک پروژه پیاده سازی شود.

به بیانی دیگر داده کاوی، فرآیند کشف الگوهای پنهان، جالب توجه، غیر منتظره و با ارزش از داخل مجموعه وسیعی از داده‌هاست و فعالیتی در ارتباط با تحلیل دقیق داده‌های سنگین بی ساختار است که علم آمار ناتوان از تحلیل آنهاست. بعضی مواقع دانش کشف شده توسط داده کاوی عجیب به نظر می‌رسد؛ مثلا ارتباط افراد دارای کارت اعتباری و جنسیت با داشتن دفترچه تامین اجتماعی یا سن، جنسیت و درآمد اشخاص با پیش بینی خوش حسابی او در بازپرداخت اقساط وام. داده کاوی در حوزه‌های تصمیم گیری، پیش بینی، و تخمین مورد استفاده قرار می‌گیرد.

پایه و اساس این تکنیک، ریشه در علوم زیر دارد:

علم آمار و احتمال
کامپیوتر (تکنولوژی اطلاعات)
هوش مصنوعی (تکنیکهای یادگیری ماشین)

ارتباط داده کاوی و OLAP

OLAP و داده کاوی فن آوری‌های تحلیلی در خانواده BI به شمار می‌آیند. OLAP در زمینه تجمیع مقادیر عظیم داده‌های تراکنشی بر پایه تعاریف ابعادی مناسب است.

سوالات موضوعی که در ادامه به آن اشاره می‌شود توسط OLAP پاسخ داده می‌شوند:
مقدار فروش کل تولیدات در سه ماهه گذشته در یک منطقه بخصوص چقدر بوده است؟

کدامیک از محصولات جزء ده محصول پر فروش تمامی فروشگاه‌ها در ماه گذشته بودند؟

کدامیک از محصولات برای مشتریان زن و مشتریان مرد فروش قابل توجهی داشته است؟

تفاوت میزان فروش روزانه در هنگام تبلیغات در مقایسه با دوره زمانی عادی چیست؟

فن آوری OLAP بر پایه محاسبات تجمیعی است. سرویس دهنده OLAP نوع خاصی از سرویس دهنده‌ی بانک اطلاعاتی محسوب می‌گردد که با داده‌های چند بعدی سروکار دارد. بسیاری از مشکلات و مخاطرات نظیر ایندکس گذاری، ذخیره سازی داده‌ها و ...

که در RDBMSها وجود دارد در سرویس دهنده‌ی OLAP نیز وجود دارد.
داده کاوی در یافتن الگوهای پنهان از یک مجموعه داده توسط تحلیل همبستگی میان مقادیر مشخصه‌ها مناسب است.

تکنیک‌های داده کاوی دو گونه هستند: نظارت شده و نظارت نشده. در داده کاوی نظارت شده کاربر می‌بایست مشخصه‌ی هدف و مجموعه داده‌ی ورودی را تعیین نماید. الگوریتم‌های داده کاوی نظارت شده شامل درخت تصمیم، نیو بیز و شبکه‌های عصبی هستند. تکنیک‌های داده کاوی نظارت نشده نیازی به تعیین مشخصه‌ی قابل پیش بینی ندارد. خوشه بندی مثال خوبی از داده کاوی نظارت نشده می‌باشد و به گروه بندی نقاط داده ای ناهمگن به زیر گروه هایی می‌پردازد که در آنها نقاط داده ای کم و بیش مشابه و همگن هستند.

در زیر نمونه ای از سوالات پاسخ داده شده توسط داده کاوی ارائه شده است: مشخصات مشتریانی که تمایل به خرید جدیدترین مدل را دارند، چیست؟

چه کالاهایی باید به این دسته از مشتریان خاص توصیه و پیشنهاد گردد؟

برآورد میزان فروش مدلی خاص در سه ماهه آینده چیست؟

چگونه باید مشتریان را تقسیم بندی کرد؟

یکی از فرآیندهای اصلی داده کاوی، تحلیل همبستگی میان مشخصه‌ها و مقادیر آنها است. محققین آمار در این موارد قرن‌ها مطالعه داشته‌اند. OLAP و داده کاوی دو فن آوری مختلف هستند اما فعالیت‌های یکدیگر را تکمیل می‌کنند. OLAP فعالیت هایی نظیر خلاصه سازی، تحلیل تغییرات در طول زمان و تحلیل‌های What If را پشتیبانی می‌نماید. همچنین می‌توان آنرا برای تحلیل نتایج داده کاوی در سطوح مختلف و مجزا استفاده کرد. داده کاوی نیز می‌تواند در ساخت Cubeهای مفیدتر سودمند باشد.

تفاوت میان OLAP و داده کاوی ارتباطی به تفاوت میان داده‌های تلخیص شده و داده‌های تشریحی ندارد. در واقع تمایز قابل توجهی میان مدل سازی توصیفی و تشریحی وجود دارد. توابع و الگوریتم هایی که معمولاً در ابزارهای OLAP یافت می‌شود، توابع مدل سازی توصیفی به شمار می‌آیند. در حالیکه توابعی که در آنچه که اصطلاحاً بسته داده کاوی نامیده می‌شود، یافت می‌شود توابع یا الگوهای مدل سازی تشریحی هستند.

الگوریتم‌های داده کاوی موجود در SSAS و زمینه کاری متناظر
این الگوریتم‌ها را به 5 دسته تقسیم می‌توان نمود:

پیش بینی توالی وقایع
برای مثال جهت تجزیه و تحلیل مجموعه ای از شرایط آب و هوایی که منجر به وقوع پدیده خاصی می‌شود. از الگوریتم زیر استفاده می‌شود:
Microsoft Sequence Clustering Algorithm
یافتن گروهی از موارد مشترک در تراکنش ها معروفترین مثال در خصوص تجزیه و تحلیل سبد بازار است. از الگوریتم‌های زیر استفاده می‌شود:
Microsoft Association Algorithm
Microsoft Decision Trees Algorithm

یافتن گروهی از موارد مشابه معمول‌ترین کاربرد زمینه بخش بندی داده‌های مشتریان به منظور یافتن گروه‌های مجزا از مشتریان است. از الگوریتم‌های زیر استفاده می‌شود:
Microsoft Clustering Algorithm
Microsoft Sequence Clustering Algorithm

پیش بینی صفات گسسته به عنوان مثال، پیش بینی اینکه یک مشتری خاص، تمایلی به خرید محصول جدید دارد یا خیر. از

الگوریتم‌های زیر استفاده می‌شود:

Microsoft Decision Trees Algorithm

Microsoft Naive Bayes Algorithm

Microsoft Clustering Algorithm

Microsoft Neural Network Algorithm

پیش بینی صفات پیوسته پیش بینی درآمد در ماه آینده مثالی از آن می‌باشد. از الگوریتم‌های زیر استفاده می‌شود:

Microsoft Decision Trees Algorithm

Microsoft Time Series Algorithm