

در طول این سری آموزش‌های MDX (البته هنوز نمی‌دانم چند قسمت خواهد بود) تلاش خواهم کرد تمامی موارد موجود در MDX‌ها را به طور کامل با شرح و توضیح مناسب پوشش دهم.

امیدوارم شما دوستان عزیز پس از مطالعه‌ی این مجموعه مقالات به دانش کافی در خصوص MDX Query‌ها دست پیدا کنید.

در قسمت اول این آموزش‌ها در نظر دارم در ابتدا مفاهیم اولیه OLAP و همچنین مفاهیم مورد نیاز در Multi Dimensional Data Base ها برای شما عزیزان توضیح دهم و در قسمت‌های بعدی این مجموعه در خصوص MDX Query‌ها صحبت خواهم کرد.

### انباره داده (Data Warehouse)

عملایک یا چند پایگاه داده می‌باشد که اطلاعات تجمعی شده از دیگر پایگاه‌های داده را در خود نگه داری می‌کند. برای ارایه گزارشاتی که از پایگاه داده‌های OLTP نمی‌توانیم به راحتی بگیریم.

### OLTP (Online transaction processing)

سیستم پردازش تراکنش برخط می‌باشد. که عملایمان سیستم‌هایی می‌باشد که در طول روز دارای تغییرات بسیار زیادی می‌باشد (مانند سیستم‌های حسابداری، انبار داری و ...) که در طول روز دایماً دارای تغییرات در سطح داده می‌باشدند).

### OLAP (OnLine Analysis Processing)

این سیستم‌ها خدماتی در نقش تحلیل‌گر داده و تصمیم گیرنده ارائه می‌کند. چنین سیستم‌هایی می‌توانند، داده را در قالب‌های مختلف برای هماهنگ کردن نیازهای مختلف کاربران مختلف، سازماندهی کنند.

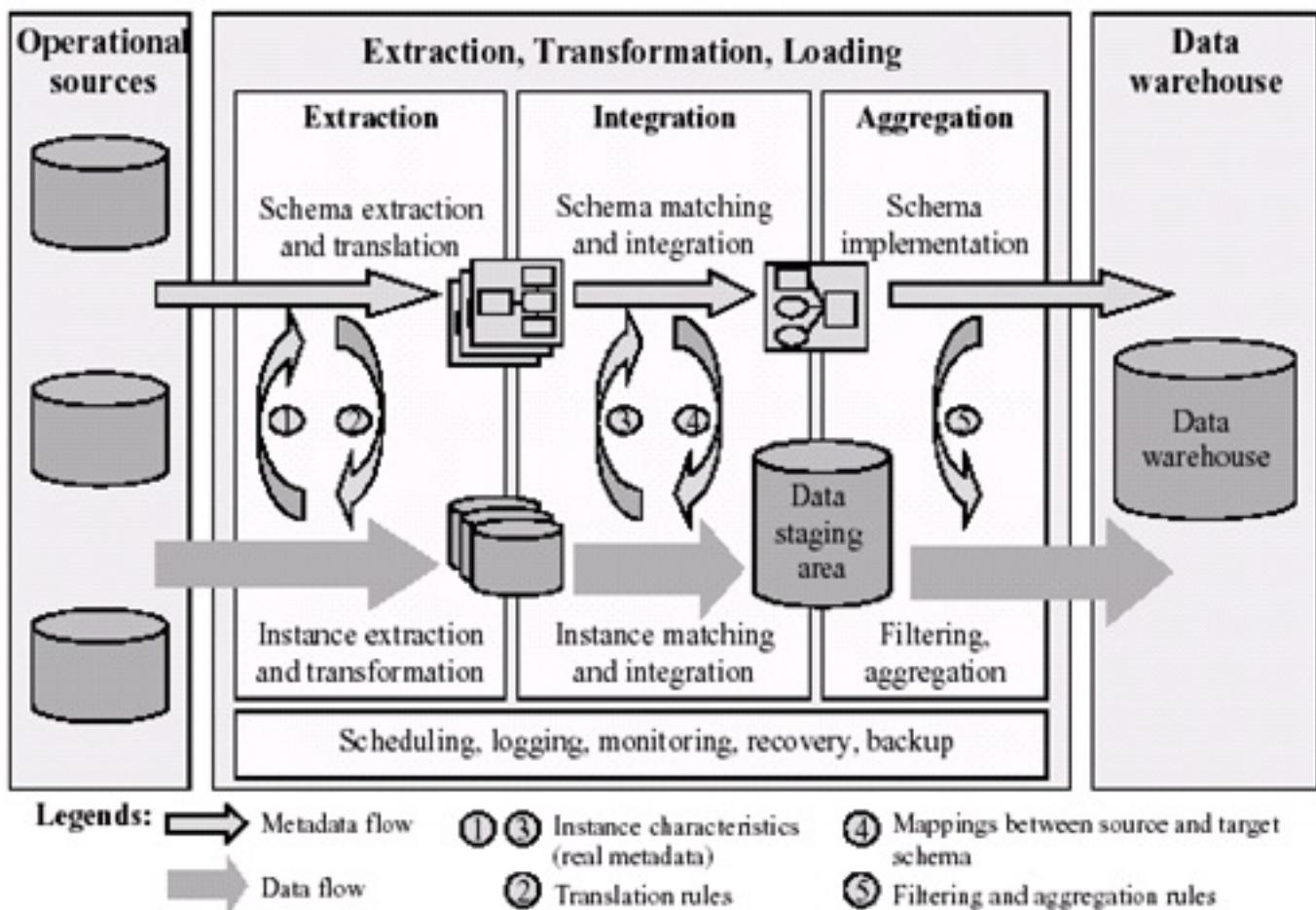
### تفاوت انبار داده (Data Base) و پایگاه داده (Data Warehouse)

وظیفه اصلی سیستم‌های پایگاه داده کاربردی OnLine، پشتیبانی از تراکنش‌های برخط و پردازش کوئری است. این سیستم‌ها، سیستم پردازش تراکنش برخط (OLTP) نامیده می‌شوند و بیشتر عملیات روزمره یک سازمان را پوشش می‌دهند. از سوی دیگر انبار داده، خدماتی در نقش تحلیل‌گر داده و تصمیم گیرنده ارائه می‌کند. چنین سیستم‌هایی می‌توانند داده را در قالب‌های مختلف برای هماهنگ کردن نیازهای مختلف کاربران مختلف، سازماندهی و ارائه می‌کند. این سیستم‌ها با نام سیستم‌های پردازش تحلیلی برخط (OLAP) شناخته می‌شوند.

### موارد تفاوت انبار داده (Data Base) و پایگاه داده (Data Warehouse)

- از لحاظ مدل‌های داده: پایگاه‌های داده برای مدل OLTP بهینه سازی شده‌است. که بر اساس مدل داده رابطه‌ای امکان پردازش تعداد زیادی تراکنش هم‌روند، که اغلب حاوی رکوردهای اندکی هستند را دارد. اما در انبارهای داده که برای پردازش تحلیلی برخط، طراحی شده‌اند امکان پردازش تعداد کمی کوئری پیچیده بر روی تعداد بسیار زیادی رکورد داده فراهم می‌شود. سرورهای OLAP می‌توانند از دو نوع رابطه‌ای (ROLAP) یا چندبعدی باشند (MOLAP).
- از لحاظ کاربران: کاربران پایگاه داده کارمندان دفتری و مسؤولان هستند در حالی که کاربران انبار داده مدیران و تصمیم‌گیرنده‌ها هستند.
- از لحاظ عملیات قابل اجرا بر روی آن‌ها: عملیات انجام شده بر روی پایگاه‌های داده عمدها عمليات Select/Insert/Update/Delete) می‌باشد، در حالی که عملیات روی انبار داده عمدها Select ها می‌باشد.

- از لحاظ مقدار داده‌ها: مقدار داده‌های یک پایگاهداده در حدود چند مگابایت تا چند گیگابایت است در حالی که این مقدار در انبار داده در حدود چند گیگابایت تا چند ترابایت است.
  - از لحاظ زمان پرس و جو: به طور کلی سرعت پرس و جو ها روی انبارهای داده بسیار بالاتر از کوئری مشابه آن روی پایگاه داده می‌باشد.
- مراحل ساخت یک انبارهای داده (Data Warehouse) به شرح زیر می‌باشد



#### • پاکسازی داده (Data Cleaning)

پاکسازی داده‌ها عبارت است از شناسایی و حذف خطاهای ناسازگاری‌های داده‌ای به منظور دستیابی به داده‌های با کیفیت بالاتر.

اگر داده‌ها از منابع یکسان مثل فایل‌ها یا پایگاه‌های داده‌ای گرفته شوند خطاهایی از قبیل اشتباہات تایپی، داده‌های نادرست و فیلدهای بدون مقدار را خواهیم داشت و چنانچه داده‌ها از منابع مختلف مثل پایگاه داده‌های مختلف یا سیستم اطلاعاتی مبتنی بر وب گرفته شوند. با توجه به نمایش‌های داده‌ای مختلف خطاهای بیشتر بوده و پاکسازی داده‌ها اهمیت بیشتری پیدا خواهد کرد. برای دستیابی به داده‌های دقیق و سازگار، بایستی داده‌ها را یکپارچه نموده و تکرارهای آنها را حذف نمود.

وجود خطاهای نویزی، ناسازگاری در داده‌های انبار داده و ناقص بودن داده‌ها امری طبیعی است. فیلدهای یک جدول ممکن است خالی باشند و یا دارای داده‌های خطا دار و ناسازگار باشند. برای هر کدام از این حالت‌ها روش‌هایی جهت پاکسازی و اصلاح داده‌ها ارایه می‌شود.

در این بخش عملیات مختلفی برای پاکسازی داده‌ها قابل انجام است:

- نادیده گرفتن تاپل‌های نادرست

- پرکردن فیلد‌های نادرست به صورت دستی

- پرکردن فیلد‌های نادرست با یک مقدار مشخص

- پرکردن فیلد‌ها با توجه به نوع فیلد و داده‌ها می‌موجود

- پرکردن فیلد‌ها با نزدیکترین مقدار ممکن (مثلاً میانگین فیلد تاپل‌های دیگر می‌تواند به عنوان یک مقدار مناسب در نظر گرفته شود) • **یکپارچه‌سازی (Integration)**

این فاز شامل ترکیب داده‌های دریافتی از منابع اطلاعاتی مختلف، استفاده از متاداده‌ها برای شناسایی و حذف افزونگی داده‌ها، تشخیص و رفع برخوردهای داده ای می‌باشد.

یکپارچه سازی داده‌ها از سه فاز کلی تشکیل شده است:

- **شناسایی فیلد‌های یکسان:** فیلد‌های یکسان که در جدول‌ها می‌باشند.

- **شناسایی افزونگی‌ها می‌موجود در داده‌ها ورودی:** داده‌های ورودی گاهی دارای افزونگی است. مثلاً بخشی از رکورد در جداول مختلف وجود دارد.

- **مشخص کردن برخوردهای داده ای:** مثالی از برخوردهای داده ای یکسان نبودن واحدهای نمایش داده ای است. مثلاً فیلد وزن در یک جدول بر حسب کیلوگرم و در جدولی دیگر بر حسب گرم ذخیره شده است.

- **تبديل داده‌ها (Data Transformation)**

در این فاز، داده‌های ورودی طی مراحل زیر به شکلی که مناسب عمل داده کاوی باشند، در می‌آیند:

- از بین بردن نویز داده‌ها (Smoothing)

- تجمعیح داده‌ها (Aggregation)

- کلی‌سازی (Generalization)

- نرمال‌سازی (Normalization)

- افزودن فیلد‌های جدید

در ادامه به شرح هر یک می‌پردازیم: ۱. از بین بردن نویزهای داده ای (Smoothing): منظور از داده‌های نویزی، داده‌هایی هستند که در خارج از بازه مورد نظر قرار می‌گیرند. مثلاً اگر بازه حقوقی کارمندان بین یک صد هزار تومان و یک میلیون تومان باشد، داده‌های خارج از این بازه به عنوان داده‌های نویزی شناخته شده و در این مرحله اصلاح می‌گردند. برای اصلاح داده‌های نویزی از روش‌های زیر استفاده می‌شود:

- استفاده از مقادیر مجاور برای تعیین یک مقدار مناسب برای فیلد‌های دارای نویز

- دسته بندی داده‌های موجود و مقداردهی فیلد دارای داده نویزی با استفاده از دسته نزدیکتر

- ترکیب روشهای فوق با ملاحظات انسانی، در این روش، اصلاح مقادیر نویزی با استفاده از یکی از روشهای فوق انجام می‌گیرد اما افرادی برای بررسی و اصلاح نیز وجود دارند

**2. تجمعی داده‌ها(Aggregation):** تجمعی داده‌ها به معنی بدست آوردن اطلاعات جدید از ترکیب داده‌های موجود می‌باشد. به عنوان مثال بدست فروش ماهانه از حساب فروش‌های روزانه. **3. کلی سازی(Generalization):** کلی سازی به معنی دسته بندی داده‌های موجود براساس ماهیت و نوع آنها است. به عنوان مثال می‌توان اطلاع رده‌های سنی خاص (جوان، بزرگسال، سالخورد) را از فیلد سن استخراج کرد.

**4. نرمال سازی(Normalization):** منظور از نرمال سازی، تغییر مقیاس داده‌ها است. به عنوان مثالی از نرمال سازی، می‌توان به تغییر بازه یک فیلد از مقادیر موجود به بازه 0 تا 1 اشاره کرد.

**5. افزودن فیلدی‌های جدید:** گاهی اوقات برای سهولت عمل داده کاوی می‌توان فیلدی‌ای به مجموعه فیلدی‌های موجود اضافه کرد. مثلاً می‌توان فیلد میانگین حقوق کارمندان یک شعبه را به مجموعه فیلدی‌های موجود اضافه نمود.

#### • کاهش داده‌ها(Reduction)

در این مرحله، عملیات کاهش داده‌ها انجام می‌گیرد که شامل تکنیکهایی برای نمایش کمینه اطلاعات موجود است

. این فاز از سه بخش تشکیل می‌شود:

• **کاهش دامنه و بعد:** فیلدی‌های نامرتب، نامناسب و تکراری حذف می‌شوند. برای تشخیص فیلدی‌های اضافی، روشهای آماری و تجربی وجود دارند؛ یعنی با اعمال الگوریتمهای آماری و یا تجربی بر روی داده‌های موجود در یک بازه زمانی مشخص، به این نتیجه می‌رسیم که فیلد یا فیلدی‌های خاصی کاربردی در اینباره داده ای و داده کاوی نداشته و آنها را حذف می‌کنیم.

• **فسرده سازی داده‌ها:** از تکنیکهای فشرده سازی برای کاهش اندازه داده‌ها استفاده می‌شود.

• **کدکردن داده‌ها:** داده‌ها در صورت امکان با پارامترها و اطلاعات کوچکتر جایگزین می‌شوند.

### مدل داده‌ای رابطه‌ای (Relational) و چند بعدی (Multidimensional)

**1. مدل داده رابطه‌ای (Relational data modeling)** بر اساس دو مفهوم اساسی موجودیت (entity) و رابطه (relation) (entity) (relation) بنابراین نهاده شده است. از این رو آن را با نام مدل ER نیز می‌شناسند.

• **موجودیت (entity):** نمایانگر همه چیزهایی که در پایگاه داده وجود خارجی دارند یا به تصور در می‌آیند. پدیده‌ها دارای مشخصاتی هستند که به آن‌ها صفت (attribute) گفته می‌شود.

• **رابطه (relation):** پدیده‌ها را به هم می‌پیوندد و چگونگی در ارتباط قرار گرفتن آن‌ها با یکدیگر را مشخص می‌کند.

**2. مدل داده چندبعدی (Multidimensional modeling)** یا MD بر پایه دو ساختار جدولی اصلی بنا نهاده شده است:

#### • جدول حقایق (Fact Table)

#### • جداول ابعاد (Dimension Table)

این ساختار امکان داشتن یک نگرش مدیریتی و تصمیم‌گیری به داده‌های موجود در پایگاه داده را تسهیل می‌کند.

**جدول حقایق :** قلب حجم داده‌ای ما را تشکیل می‌دهد و شامل دو سری فیلد است : کلیدهای خارجی به ابعاد و شاخص‌ها (Measure).

**شاخص‌ها (Measure) :** معیارهایی هستند که بر روی آن‌ها تحلیل انجام می‌گیرد و درون جدول حقایق قرار دارند. شاخص‌ها قبل از شکل‌گیری انبار داده توسط مدیران و تحلیل‌گران به دقت مشخص می‌شوند. چون در مرحله کار با انبار اطلاعات اساسی هر تحلیل بر اساس همین شاخص‌ها شکل می‌گیرد. شاخص‌ها تقریباً همیشه مقادیر عددی را شامل می‌شوند. مثلاً برای یک فروشگاه

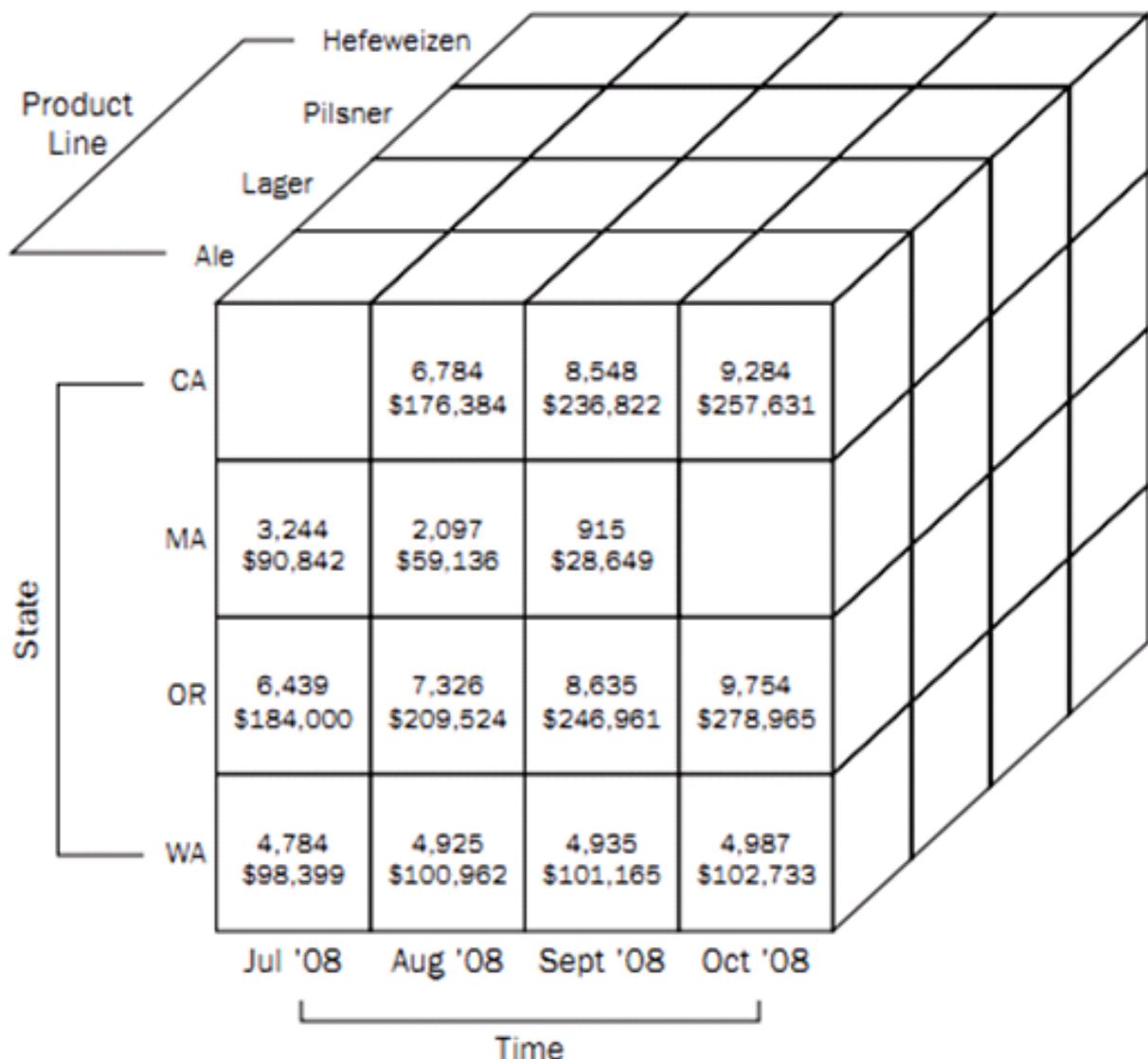
زنگیرهای این شاخص‌ها می‌توانند واحدهای فروخته‌شده کالاهای و مبلغ فروش به تومان باشند.

بعد (**Dimension**) : هر موجودیت در این مدل می‌تواند با یک بعد تعریف شود. ولی بعدها با موجودیت‌های مدل ER متفاوتند زیرا آن‌ها سازمان شاخص‌ها را تعیین می‌کنند. علاوه بر این دارای یک ساختار سلسله مراتبی هستند و به طور کلی برای حمایت از سیستم‌های تصمیم‌گیری سازمان‌دهی شده‌اند.

اجزای بعدها **member** نام دارند و تقریباً همه بعدها، **member**‌های خود را در یک یا چند سطح سلسله مراتبی (**hierarchies**) سازمان‌دهی می‌نمایند، که این سلسله مراتب نمایانگر مسیر تجمعی (integration) و ارتباط بین سطوح پایین‌تر (مثل روز) و سطوح بالاتر (مثل ماه و سال) است. وقتی یک دسته از **member**‌های خاص با هم مفهوم جدیدی را ایجاد می‌کنند، به آنها یک سطح (**Level**) می‌گوییم. ( مثلًا هر سی روز را ماه می‌گوییم. در این حالت ماه یک سطح است.)

### حجم‌های داده‌ای (Data Cube)

حجم‌های داده‌ای یا **Cube** از ارتباط تعدادی بعد با تعدادی شاخص تعریف می‌شود. ترکیب **member**‌های هر بعد از حجم داده‌ای فضای منطقی را تعریف می‌کند که در آن مقادیر شاخص‌ها ظاهر می‌شوند. هر بخش مجزا که شامل یکی از **member**‌های بعد در حجم داده‌ای است، سلول (**cell**) نامیده می‌شود. سلول‌ها شاخص‌های مربوط به تجمعی‌های مختلف را در خود نگهداری می‌نمایند. در واقع مقادیر مربوط به حقایق (**Fact**) که در جدول حقایق (**Fact**) تعریف می‌شوند در حجم داده‌ای (**Data Cube**) در سلول‌ها (**cell**) نمایان می‌گردند.



شماهای داده‌ای (Data Schema) : سه نوع Data Warehouse در طراحی Schema وجود دارد

Star .1

Snowflake .2

Galaxy .3

1. شماهی ستاره‌ای (Star Schema) : متدالترین شما، همین شماهی ستاره‌ای است. که در آن انبارداده با استفاده از اجزای زیر تعریف می‌شود:

- یک جدول مرکزی بزرگ به نام جدول حقایق که شامل حجم زیادی از داده‌های بدون تکرار است.

- مجموعه‌ای از جدول‌های کمکی کوچک‌تر به نام جدول بعد، که به ازای هر بعد یکی از این جداول موجود خواهد بود.

- شکل این شما به صورت یک ستاره است که جدول حقایق در مرکز آن قرار گرفته و هر یک از جداول بعد به وسیله شعاع‌هایی به آن مربوط هستند.

مشکل این مدل احتمال پیشامد افزونگی در آن است.

2. شمای دانهبرفی (Snowflake Schema) : در واقع شمای دانهبرفی، نوعی از شمای ستاره‌ای است که در آن بعضی از جداول بعد نرمال شده‌اند. و به همین خاطر دارای تقسیمات بیشتری به شکل جداول اضافی می‌باشد که از جداول بعد جدا شده‌اند.

تفاوت این دو شما در این است که جداول شمای دانه برای دانه هستند و افزونگی در آن‌ها کاهش یافته است. که این برای کار کردن با داده‌ها و از لحاظ فضای ذخیره‌سازی مفید است. ولی در عوض کارایی را پایین می‌آورد، زیرا در محاسبه کوئری‌ها به join زهای بیشتری نیاز داریم.

3. شمای کهکشانی (galaxy schema) : در کاربردهای پیچیده برای به اشتراک گذاشتن ابعاد نیاز به جداول حقایق چندگانه احساس می‌شود که یک یا چند جدول بعد را در بین خود به اشتراک می‌گذارند. این نوع شما به صورت مجموعه‌ای از شماهای ستاره‌ای است و به همین دلیل شمای کهکشان یا شمای منظومه‌ای نامیده می‌شود. این شما به ما این امکان را می‌دهد که جداول بعد بین جداول حقایق مختلف به اشتراک گذاشته شوند.

#### عملیات بر روی حجم‌های داده‌ای :

- Roll Up (یا Drill-up) : با بالا رفتن در ساختار سلسله مراتبی مفهومی یک حجم داده‌ای، یا با کاهش دادن بعد، یک مجموعه با جزئیات کمتر (خلاصه شده) ایجاد می‌نماید. بالا رفتن در ساختار سلسله مراتبی به معنای حذف قسمتی از جزئیات است. برای مثال اگر قبل از بعد مکان بر حسب شهر بوده آن را با بالا رفتن در ساختار سلسله مراتبی بر حسب کشور در می‌آوریم. ولی وقتی با کاهش دادن بعد سروکار داریم منظور حذف یکی از ابعاد و جایگزین کردن مقادیر کل است. در واقع همان عمل تجمیع (aggregation) است.

- Drill Down : بر عکس عمل Roll-up است و از موقعیتی با جزئیات داده‌ای کم به جزئیات زیاد می‌رود. این کار با پایین آمدن در ساختار سلسله مراتبی (به سمت جزئیات بیشتر) یا با ایجاد ابعاد اضافی انجام می‌گیرد.

The diagram illustrates the relationship between three levels of a sales data cube:

- Number of Autos Sold** (Top Level): A 4x4 grid showing sales by Month (Jul, Aug, Sep, Total) and Color (CA, OR, WA, Total). The total sales are 345.
- Roll up by Month**: An arrow points from the top level to a second grid, indicating the aggregation of data by month.
- Number of Autos Sold** (Middle Level): A 2x4 grid showing sales by Color (CA, OR, WA, Total) for the months. The total sales are 345.
- Drill down by Color**: An arrow points from the middle level to a third grid, indicating the disaggregation of data by color.
- Number of Autos Sold** (Bottom Level): A 4x4 grid showing sales by Color (Red, Blue, Gray, Total) and Month (Jul, Aug, Sep, Total). The total sales are 345.

	CA	OR	WA	Total
Jul	45	33	30	108
Aug	50	36	42	128
Sep	38	31	40	109
Total	133	100	112	345

	CA	OR	WA	Total
133	100	112	345	

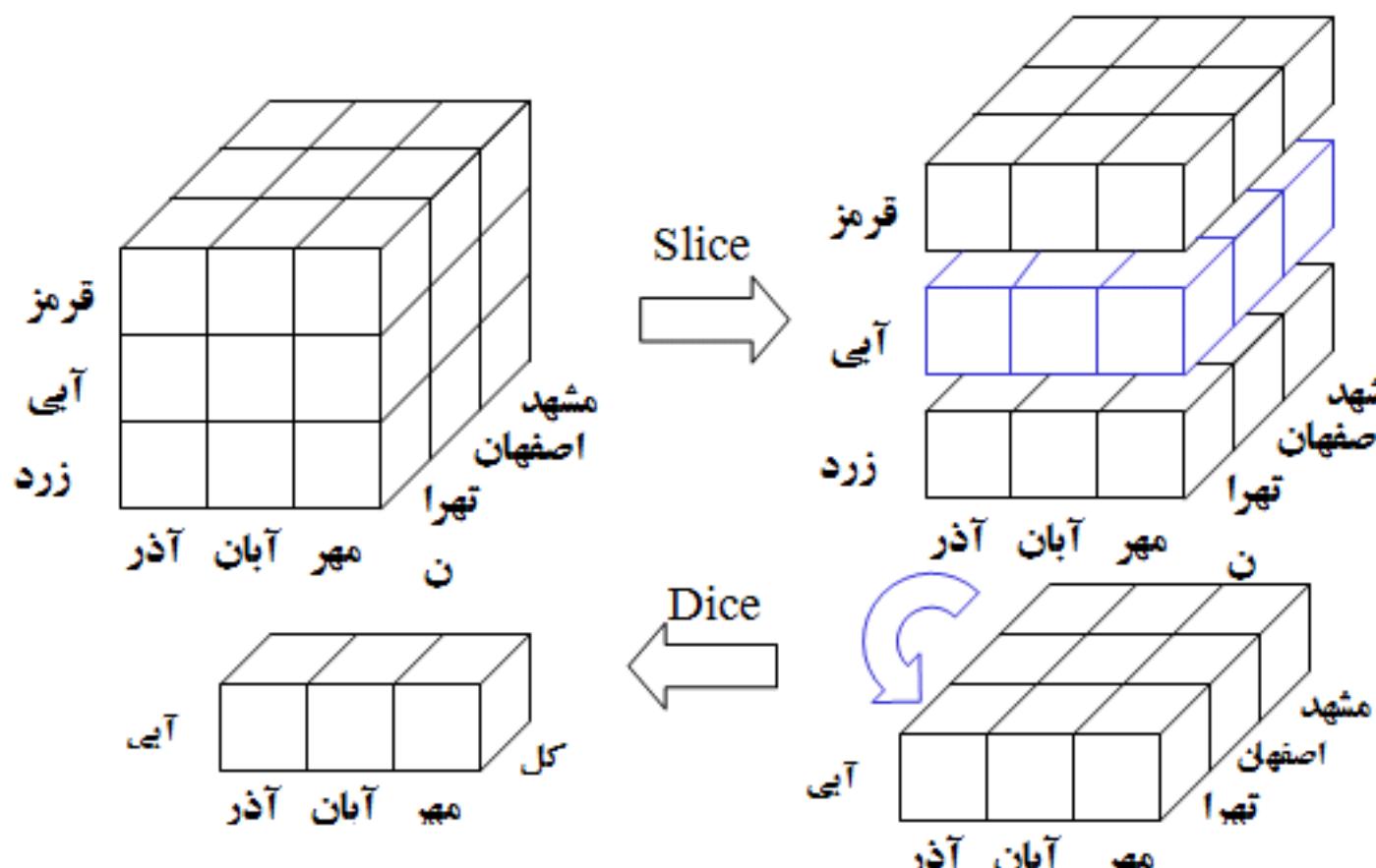
  

	CA	OR	WA	Total
Red	40	29	40	109
Blue	45	31	37	113
Gray	48	40	35	123
Total	133	100	112	345

نمونه‌ای از عملیات Roll Up و Drill Down

• Slice : با انتخاب و اعمال شرط بر روی یک از ابعاد یک subcube به شکل یک برش دو بعدی ایجاد می کند. در واقع همان عمل انتخاب (select) است.

• Dice : با انتخاب قسمتی از ساختار سلسله مراتبی بر روی دو یا چند بعد یک subcube ایجاد می نماید.



نمونه ای از عملیات Slice و Dice

(Rotate) یا (Pivot) : این عملیات بردارهای بعد را در ظاهر می چرخاند.

Region	Sales variance
Africa	105%
Asia	57%
Europe	122%
North America	97%
Pacific	85%
South America	163%

Nation	Sales variance
China	123%
Japan	52%
India	87%
Singapore	95%

نمونه‌ای از عملیات pivot

• Drill-across : نتیجه اجرای کوئری‌هایی که نتیجه اجرای آنها حجم‌های داده‌ای‌های مرکب با بیش از یک fact-table است.

• Ranking : سلول‌هایی را باز می‌گرداند که در بالا یا پایین شرط خاصی واقع هستند. مثلاً ده محصولی که بهترین فروش را داشته‌اند.

#### سرورهای OLAP :

در تکنولوژی OLAP داده‌ها به دو صورت چندبعدی (MOLAP) و رابطه‌ای (Multidimensional OLAP) یا (ROLAP) ذخیره می‌شوند. OLAP پیوندی (HOLAP) تکنولوژی است که دو نوع قبل را با هم ترکیب می‌کند.

MOLAP : روشی است که معمولاً برای تحلیل‌های OLAP در تجارت مورد استفاده قرار می‌گیرد. در MOLAP، داده‌ها با ساختار یک حجم داده‌ای (Data Cube) چند بعدی ذخیره می‌شوند. ذخیره‌سازی در پایگاه‌داده‌های رابطه‌ای انجام نمی‌گیرد، بلکه با یک فرمت خاص انجام می‌شود. اغلب محصولات موفق MOLAP از یک روش چندبعدی استفاده می‌نمایند که در آن یک سری حجم‌های داده‌ای کوچک، انبو و از پیش محاسبه شده، یک حجم داده‌ای بزرگ (hypercube) را می‌سازند.

علاوه بر این MOLAP به شما امکان می‌دهد داده‌های دیدهای (View) تحلیل‌گران را دسته بندی کنید، که این در حذف اشتباهات و برخورد با ترجمه‌های پرغلط کمک بزرگی است.

گذشته از همه این‌ها از آن‌جا که داده‌ها به طور فیزیکی در حجم‌های داده‌ای بزرگ چندبعدی ذخیره می‌شوند، سرعت انجام فعالیت‌ها بسیار زیاد خواهد بود.

از آنجا که یک کپی از داده‌های منبع در کامپیوتر Analysis server ذخیره می‌شود، کوئری‌ها می‌توانند بدون مراجعه به منابع مجدد

محاسبه شوند. کامپیوتر Analysis server ممکن است کامپیوترسرور که تقسیم بندی‌ها در آن انجام شده یا کامپیوتر دیگری باشد. این امر بستگی به این دارد که تقسیم‌بندی‌ها در کجا تعریف شده‌اند. حتی اگر پاسخ کوئری‌ها از روی تقسیمات تجمعی (integration) شده قابل دستیابی نباشند، MOLAP سریع‌ترین پاسخ را فراهم می‌کند. سرعت انجام این کار به طراحی و درصد تجمعی تقسیم‌بندی‌ها بستگی دارد.

**مزایا :** کارایی عالی- حجم‌های داده‌ای MOLAP برای بازیابی سریع داده‌ها ساخته شده‌اند و در فعالیت‌های slice و dice به صورت بهینه پاسخ می‌دهند. ترکیب سادگی و سرعت مزیت اصلی MOLAP است.

در ضمن MOLAP قابلیت محاسبه محاسبات پیچیده را فراهم می‌کند. همه محاسبات از پیش وقتی که حجم‌های داده‌ای ساخته می‌شود، ایجاد می‌شوند. بنابراین نه تنها محاسبات پیچیده انجام شدنی هستند بلکه بسیار سریع هم پاسخ می‌دهند.

**معایب :** عیب این روش این است که تنها برای داده‌هایی با مقدار محدود کارکرد خوبی دارد. از آنجا که همه محاسبات زمانی که حجم‌های داده‌ای ساخته می‌شود، محاسبه می‌گرددند، امکان این که حجم‌های داده‌ای مقدار زیادی از داده‌ها را در خود جای دهد، وجود ندارد. ولی این به این معنا نیست که داده‌های حجم‌های داده‌ای نمی‌توانند از مقدار زیادی داده مشتق شده باشند. داده‌ها می‌توانند از مقدار زیادی داده مشتق شده باشند. اما در این صورت، فقط اطلاعات level خلاصه (level ای که دارای کمترین جزئیات است یعنی سطوح بالاتر) می‌توانند در حجم‌های داده‌ای موجود باشند.

**ROLAP :** محدودیت MOLAP در حجم داده‌های قابل پرس‌وجو و نیاز به روشی که از داده‌های ذخیره‌شده به روش رابطه‌ای حمایت کند، موجب پیشرفت ROLAP شد.

مبنای این روش کارکردن با داده‌هایی که در پایگاه‌داده‌های رابطه‌ای ذخیره‌شده‌اند، برای انجام اعمال slicing و dicing معمولی است. با استفاده از این مدل ذخیره‌سازی می‌توان داده‌ها را بدون ایجاد واقعی تجمعی در پایگاه‌داده‌های رابطه‌ای به هم مربوط کرد.

**مزایا :** با این روش می‌توان به حجم زیادی از داده‌ها را رسیدگی کرد. محدودیت حجم داده در تکنولوژی ROLAP مربوط به محدودیت حجم داده‌های قابل ذخیره‌سازی در پایگاه‌داده‌های رابطه‌ای است. به بیان دیگر، خود ROLAP هیچ محدودیتی بر روی حجم داده‌ها اعمال نمی‌کند.

**معایب :** ممکن است کارایی پایین بیاید. زیرا هر گزارش ROLAP در واقع یک کوئری SQL (یا چند کوئری SQL) در پایگاه داده‌های رابطه‌ای است و اگر حجم داده‌ها زیاد باشد ممکن است زمان پاسخ کوئری طولانی شود. در مجموع ROLAP سنگین است، نگهداری آن سخت است و کند هم هست. بخصوص زمانی که نیاز به آدرس دهی جدول‌های ذخیره شده در سیستم چند بعدی داریم.

این محدودیت ناشی از عملکرد SQL است. زیرا تکنولوژی ROLAP بر پایه عبارات مولد SQL برای پرسش و پاسخ بر روی پایگاه داده رابطه‌ای است و عبارات SQL به همه نیازها پاسخ نمی‌دهند (مثلاً محاسبه حساب‌های پیچیده در SQL مشکل است)، بنابراین فعالیت‌های ROLAP به آن چه SQL قادر به انجام آن است محدود می‌گردد.

**تفاوت MOLAP و ROALP :** تفاوت اصلی این دو در معماری آن‌ها است. محصولات MOLAP داده‌های مورد نیاز را در یک حافظه نهان (cache) مخصوص می‌گذارد. ولی ROLAP تحلیل‌های خود را بدون استفاده از یک حافظه میانی انجام می‌دهد، بدون آن که از یک مرحله میانی برای گذاشت داده‌ها در یک سرور خاص استفاده کند.

با توجه به کند بودن ROLAP در مقایسه با MOLAP، باید توجه داشت که کاربرد این روش بیشتر در پایگاه داده‌های بسیار بزرگی است که گاه‌گاهی پرس و جویی بر روی آن‌ها شکل می‌گیرد، مثل داده‌های تاریخی و کمتر جدید سال‌های گذشته.

**نکته:** اگر از Analysis Services که به وسیله Microsoft OLE DB Provider مهیا شده استفاده می‌کنید، تجمعی‌ها نمی‌توانند برای تقسیم‌بندی از روش ROLAP استفاده نمایند.

**HOLAP :** با توجه به نیاز رو به رشدی که برای کارکردن با داده‌های بلادرنگ (real time) در بخش‌های مختلف در صنعت و تجارت

احساس می‌شود، مدیران تجاری انتظار دارند بتوانند با دامنه وسیعی از اطلاعات که فوراً و بدون حتی لحظه‌ای تأخیر در دسترس باشند، کار کنند. در حال حاضر شبکه اینترنت و سایر کاربردهایی از منابع مختلف مراجعه دارند و نیاز به فعالیت با یک سیستم بلادرنگ هم دارند، همگی از سیستم HOLAP بهره می‌گیرند.

#### : named set

مجموعه‌ای از member بعد یا مجموعه‌ای از عبارات است که برای استفاده مجدد ایجاد می‌شود.

#### Calculated member

memberها هستند که بر اساس داده‌ها نیستند بلکه بر اساس عبارات ارزیابی MDX هستند. آنها دقیقاً به سبک سایر memberهای معمولی هستند. MDX یک مجموعه قوی از عملیاتی را تامین می‌کند که میتوانند برای ساخت Calculated Memberها مورد استفاده قرار گیرند به طوری که به شما امکان داشتن انعطاف زیاد در کار کردن با داده‌های چند بعدی را بدهد.

امیدوارم در این قسمت با مفاهیم نخستین OLAP آشنا شده باشید.

تلash خواهم کرد در قسمت بعدی در خصوص نصب SQL Server Analysis Services و نصب پایگاه داده‌ی Adventure Work DW 2008 شرح کاملی را ارایه کنم.

### نظرات خوانندگان

نویسنده: afshin  
تاریخ: ۱۷:۲۶ ۱۳۹۲/۰۹/۲۰

خیلی ممنون ... لطفاً ادامه بدید ... به بحث عملی هم میپردازید ؟ کلا بحث عملی بر پایه SQL Server هست یا از نرم افزارهای دیگه هم استفاده میشه ؟ در مورد ETS ها هم اگه میشه توضیح بدید .... باز هم تشکر

نویسنده: vahid  
تاریخ: ۲۱:۱۶ ۱۳۹۲/۰۹/۲۰

ممنون از مطلب مفیدتون لطفاً در مورد اینکه چگونه datawarehouse را پیاده سازی کنیم توضیح دهید ممنون

نویسنده: اردلان شاه قلی  
تاریخ: ۲۳:۸ ۱۳۹۲/۰۹/۲۰

به طور کلی زبان MDX بین DBMS یک سان میباشد (تقریباً) و البته من در این مجموعه تلاش دارم در خصوص MDX Query ها آنچه را بلد هستم با شما در میان بگذارم. اگر عمری بود مطالبی در خصوص UI هم اضافه خواهم کرد.

نویسنده: اردلان شاه قلی  
تاریخ: ۲۳:۹ ۱۳۹۲/۰۹/۲۰

تلashem را میکنم بعد از آموزش MDXها یک مطلب هم در این خصوص برای شما بگذارم.

نویسنده: محمد باقر سیف اللهی  
تاریخ: ۱۴:۳ ۱۳۹۲/۰۹/۲۱

سلام، لطفاً تعریفی از Datamart و ارتباطش با Datawarehouse ارائه بدهید. تشکر

نویسنده: اردلان شاه قلی  
تاریخ: ۱۵:۴۹ ۱۳۹۲/۰۹/۲۱

اگر اجازه دهید تعاریف DataMart و ارتباطش با DW را در یک مقاله کوتاه انتشار میدم . اینجا در کامنت بگذارم تعداد کمی بهش دسترسی پیدا خواهد کرد.

نویسنده: علاقه مند  
تاریخ: ۱۱:۱۰ ۱۳۹۲/۱۰/۱۰

سلام، آیا کتاب فارسی تالیفی (یا ترجمه بصورت مناسب و کامل) با موضوع انبار داده‌ها و OLAP در بازار سراغ دارد؟

نویسنده: اردلان شاه قلی  
تاریخ: ۱۶:۱۷ ۱۳۹۲/۱۰/۱۰

نه من از منبع فارسی مطالعه نکردم و اطلاعی هم در این خصوص ندارم . البته انتشارات Wrox دو تا کتاب خود در خصوص SSAS دارد منبع اطلاعاتی هم که بنده اینجا ارایه میکنم همین مرجع‌ها میباشد.

در این قسمت در خصوص نحوه‌ی نصب SSAS صحبت خواهم کرد.

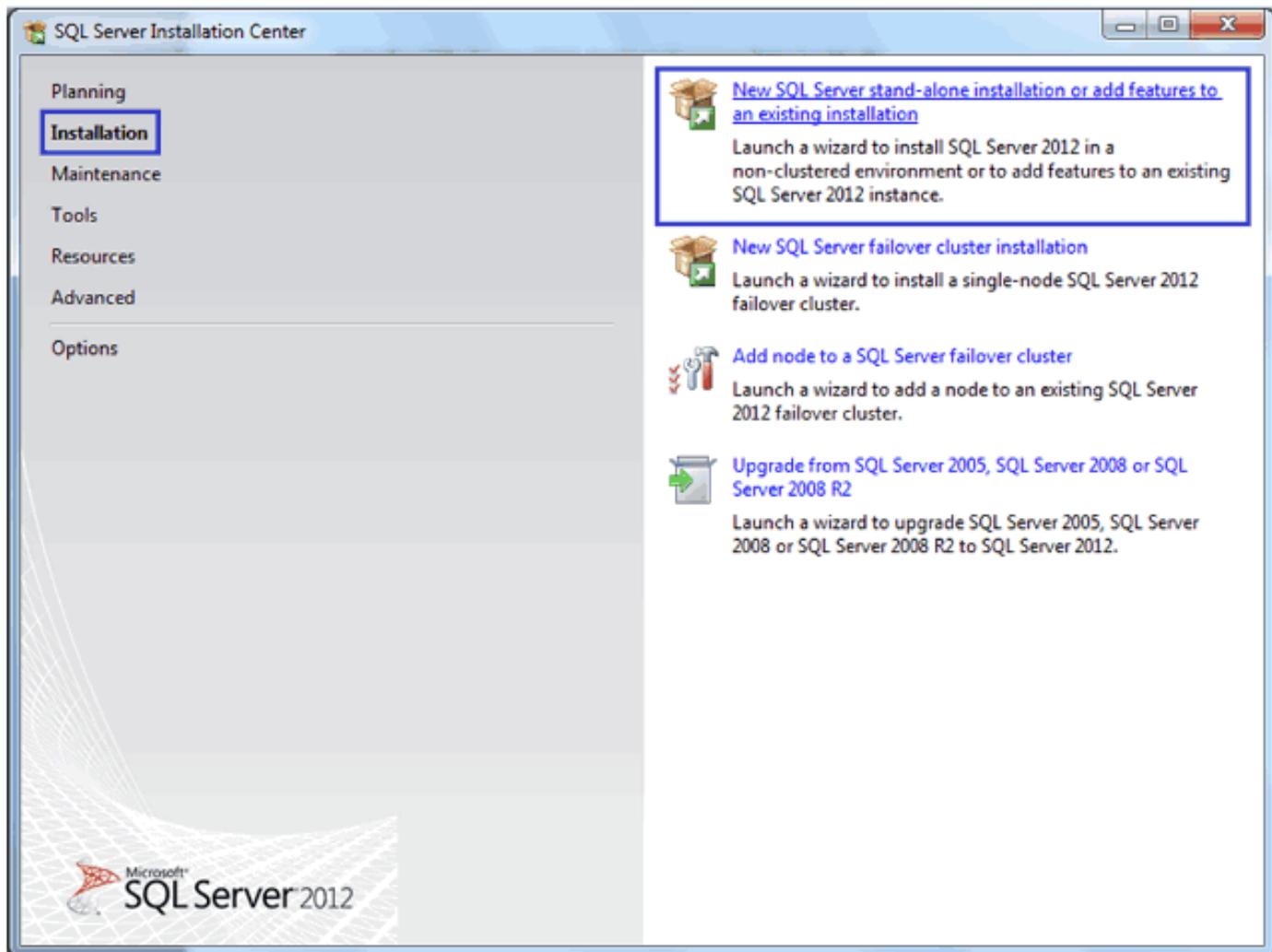
تصور می‌کنم بهتر است در خصوص آنچه در هنگام نصب SQL Server انتخاب می‌کنیم، وقت بیشتری کنیم. بسیار دیده ام که برخی از دوستان و همکاران در هنگام نصب SQL Server Feature ها تمامی آنها را انتخاب کرده در صورتی که تنها به Database Engine نیاز دارند و عملاً با این کار، کارایی Database Server خود را پایین می‌آورند.

بنابر این توصیه می‌کنم در پنجره‌ی Feature Selection فقط آنچه را که نیاز دارد نصب نمایید.

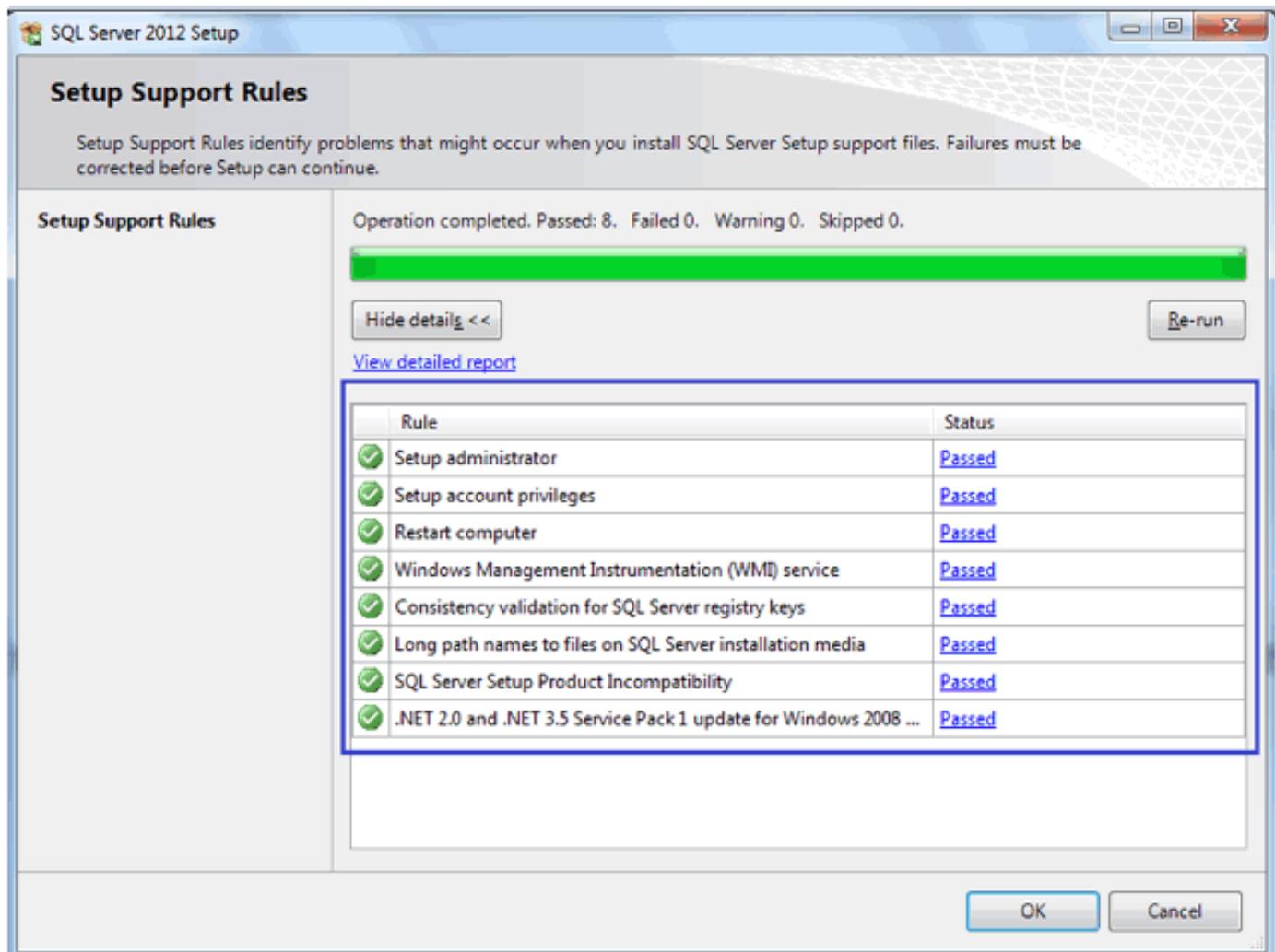
بنابر این در صورتی که شما جزو آن دسته دوستانی می‌باشید که در پنجره‌ی Feature Selection تمامی گزینه‌ها را انتخاب نموده اید، خوب نیازی به نصب مجدد SSAS ندارید و شما ناخواسته این سرویس را برروی Database Server خود نصب نموده اید.

در صورتی که شما قبل این سرویس را برروی سرور خود نصب نکرده اید و فقط Database Engine را نصب نموده اید مراحل زیر را طی نمایید.

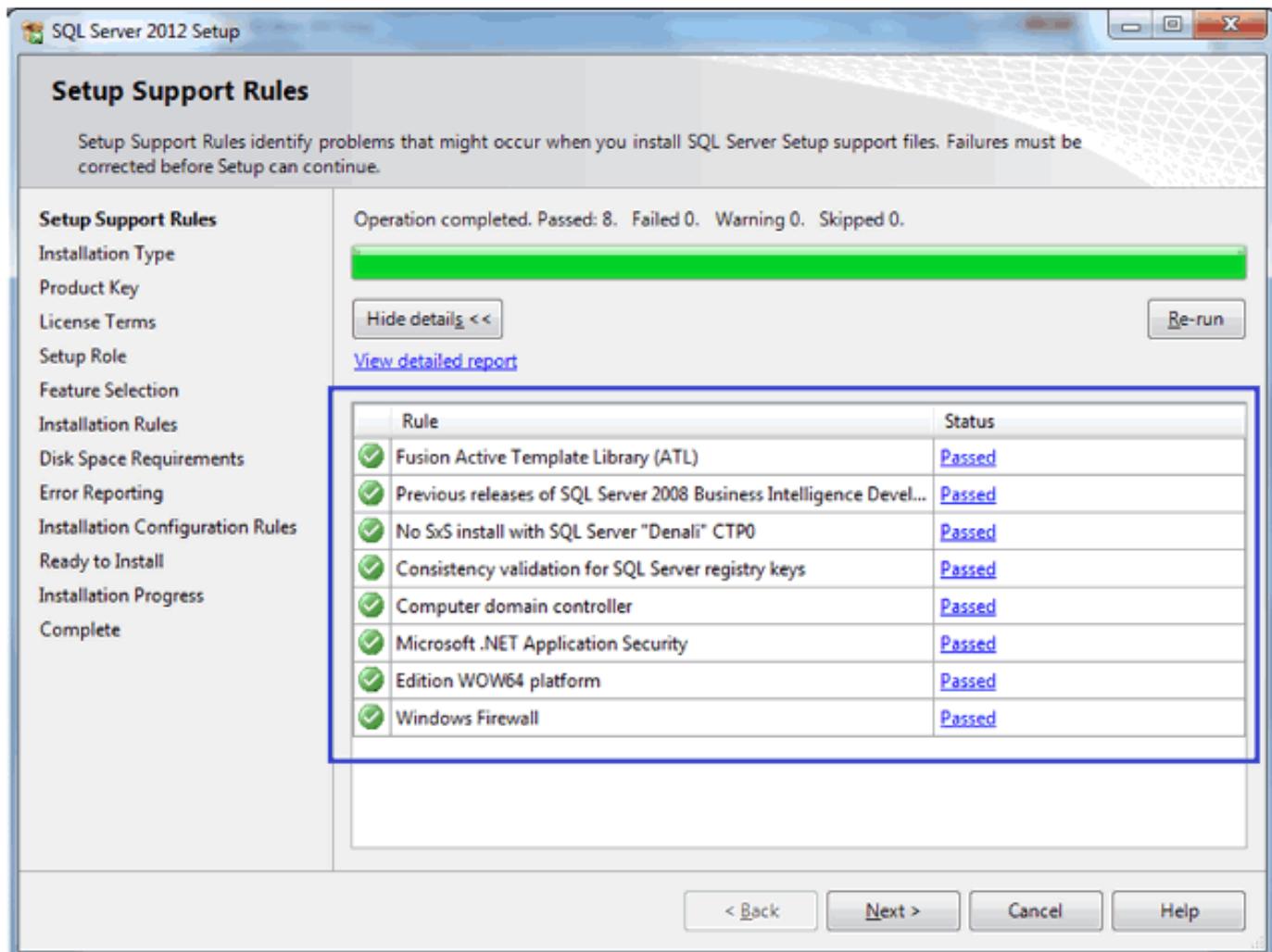
1. در ابتدا Set Up 2012 SQL Server را اجرا نمایید. و در صفحه‌ی ابتدایی برنامه‌ی نصب، مطابق شکل زیر روی New SQL Server stand-alone installation or add features to an existing installation کلیک کنید. و در قسمت سمت راست گزینه‌ی Installation را انتخاب نمایید.



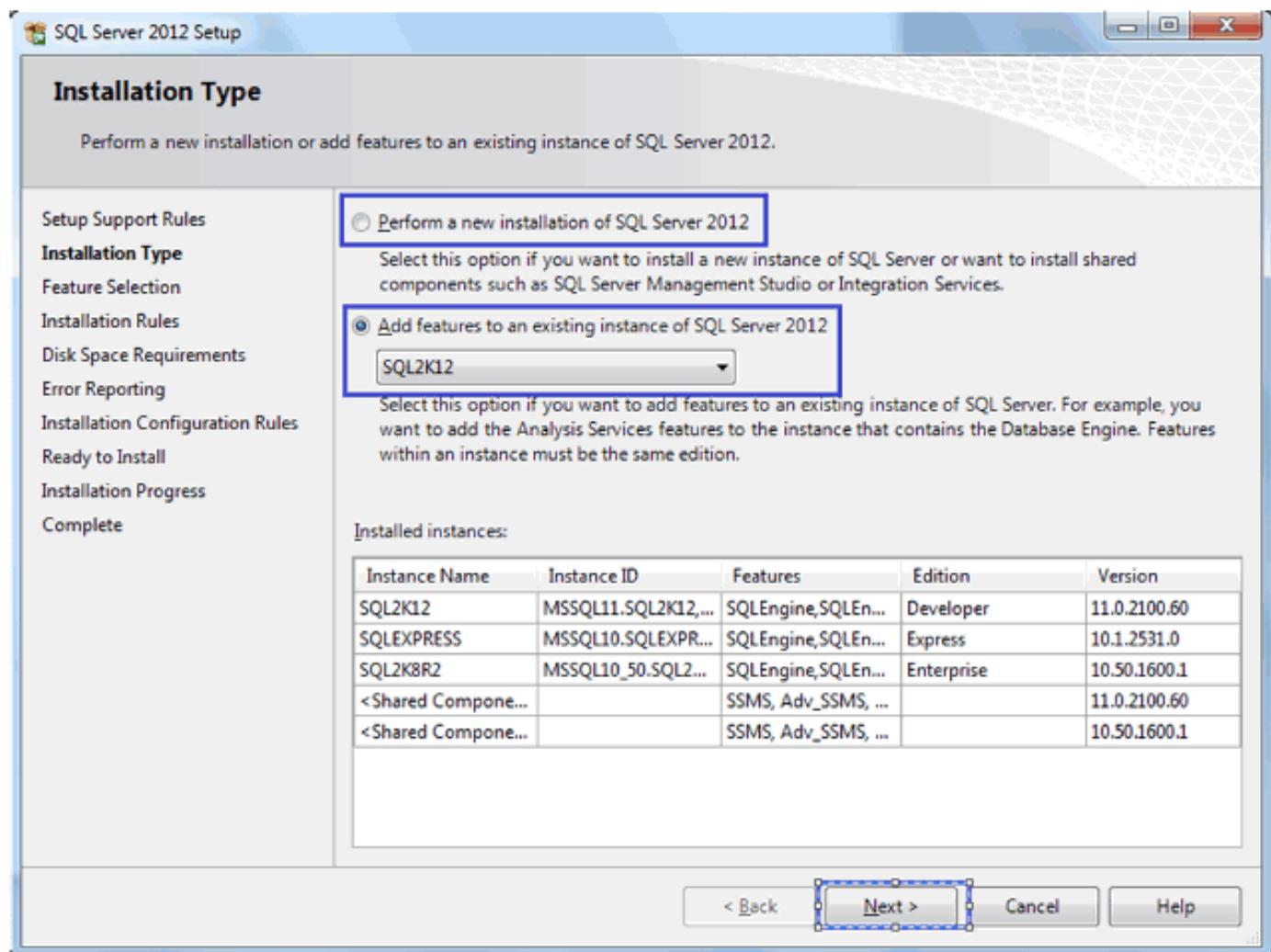
2. د رپنجره‌ی Setup Support Rules مطمئن شوید که تمامی پیش شرایط نصب را دارید (در صورتی که Warning داشته باشد احتمالا در مراحل بعدی، نصب برنامه با مشکل رو برو خواهد شد یا بعد از نصب برخی قسمت‌های برنامه به درستی کار نمی‌کنند.) در صورتی که در قسمتی با Warning رو برو شدید بعد از برطرف کردن مشکل دکمه‌ی Rerun را بزنید به عبارت دیگر نیازی نمی‌باشد مراحل نصب را از ابتدا ادامه دهید. سپس دکمه‌ی OK را بفشارید.



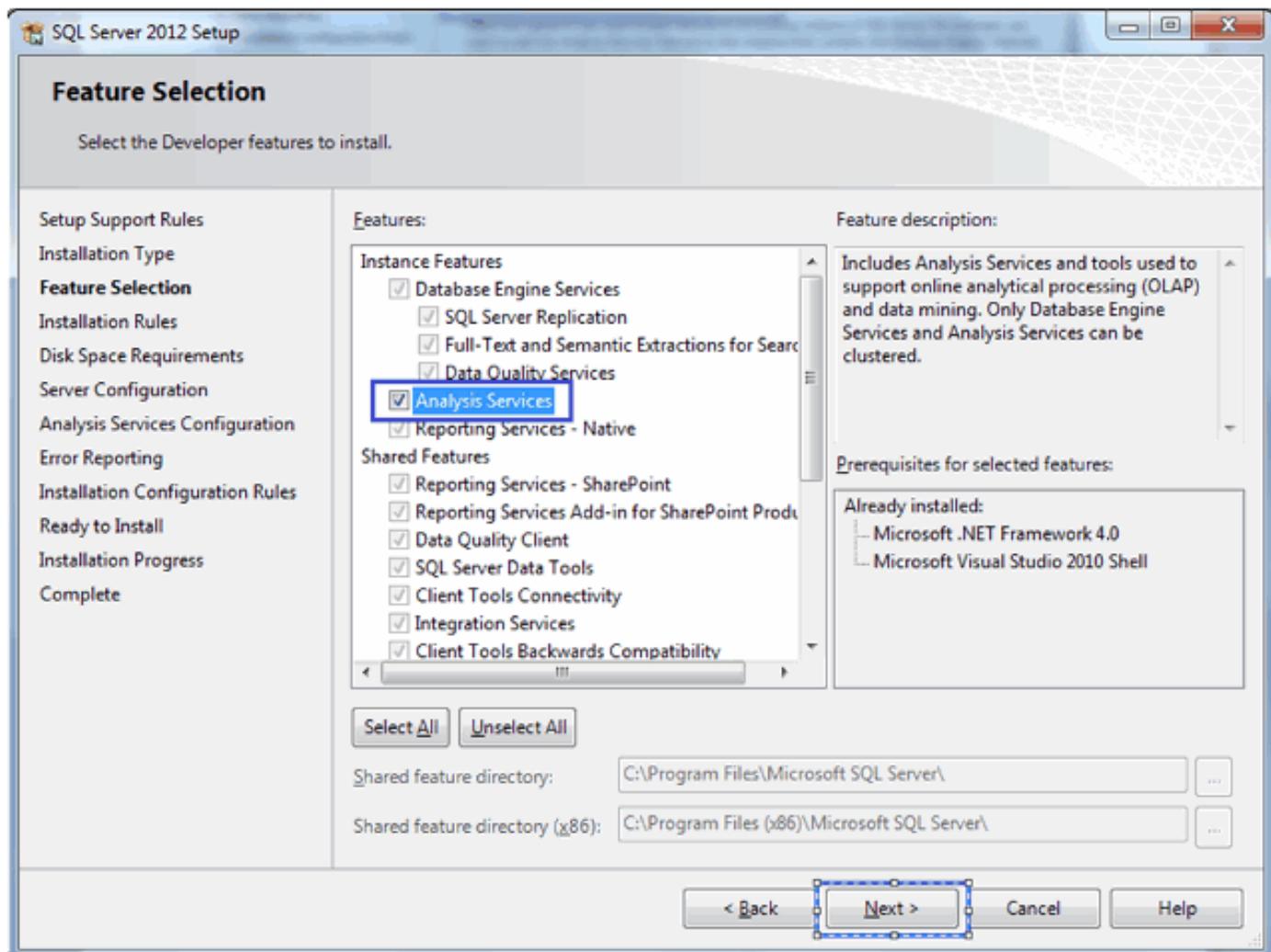
3. در پنجره‌ی بعدی دکمه‌ی Install را بزنید. سپس دوباره صفحه‌ی Setup Support Rules را خواهید داشت. مطمئن شوید تمامی پیش شرایط Passed شده باشند. سپس دکمه‌ی Next را بزنید.



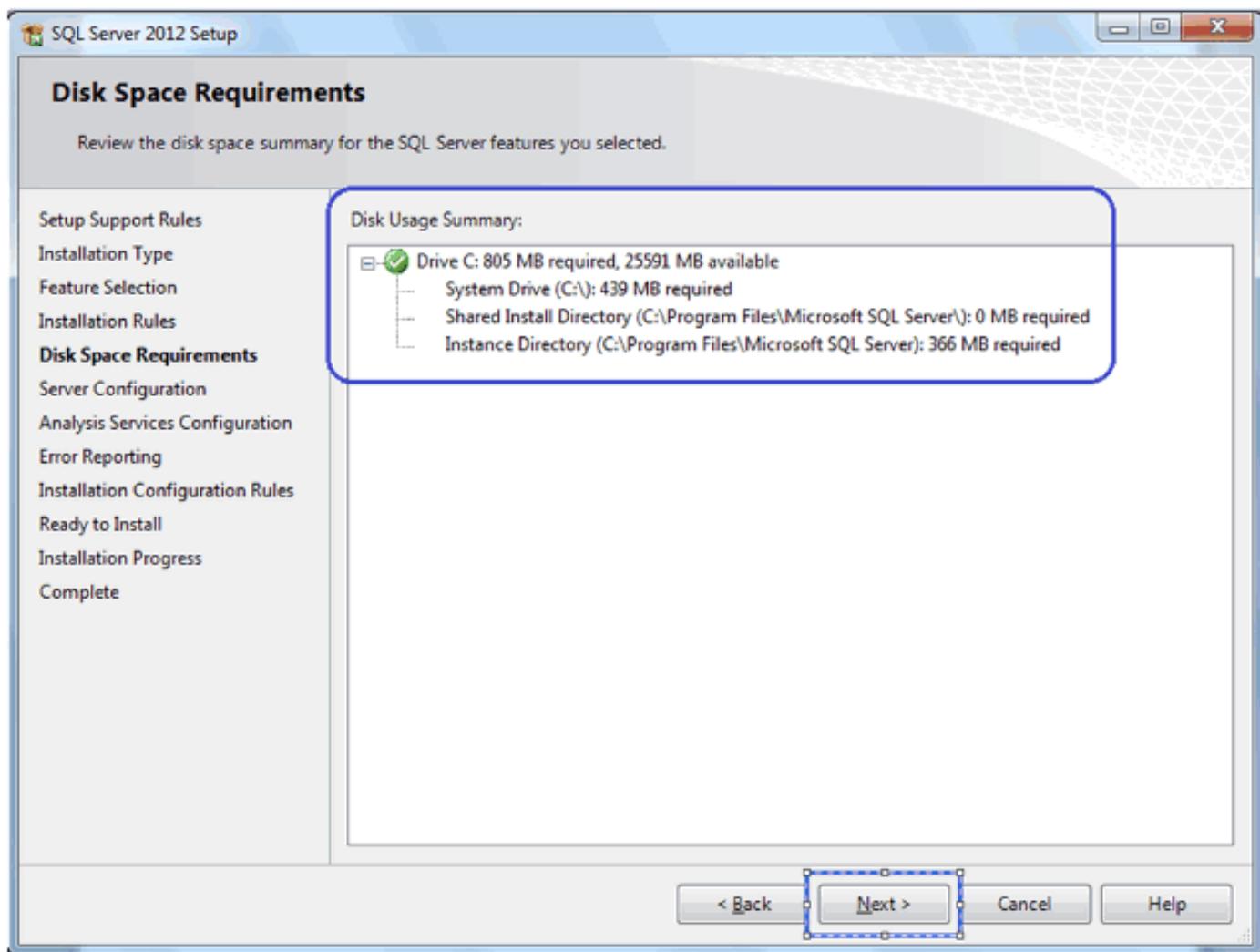
4. در پنجره‌ی بعدی گزینه‌ی Add features to an existing instance of SQL Server 2012 را انتخاب نمایید اگر شما قبلاً فقط DataBase Engine را نصب کرده‌اید و در غیر این صورت Perform a new installation of SQL Server 2012 را انتخاب نمایید. سپس دکمه‌ی Next را بزنید.



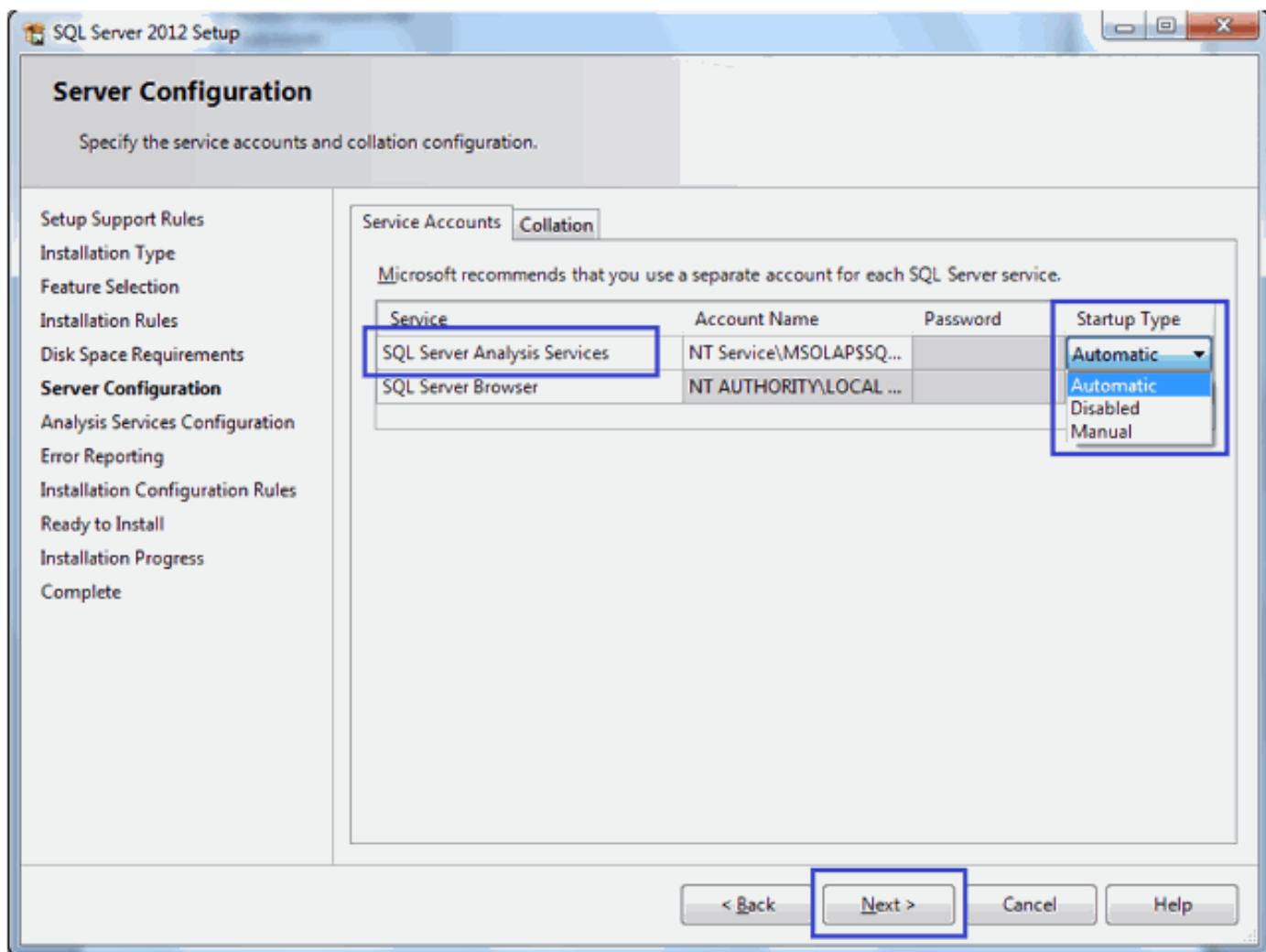
5. در صفحه‌ی Feature Selection گزینه‌ی Analysis Services را مطابق شکل زیر انتخاب نمایید. و سپس دکمه‌ی Next را بزنید.



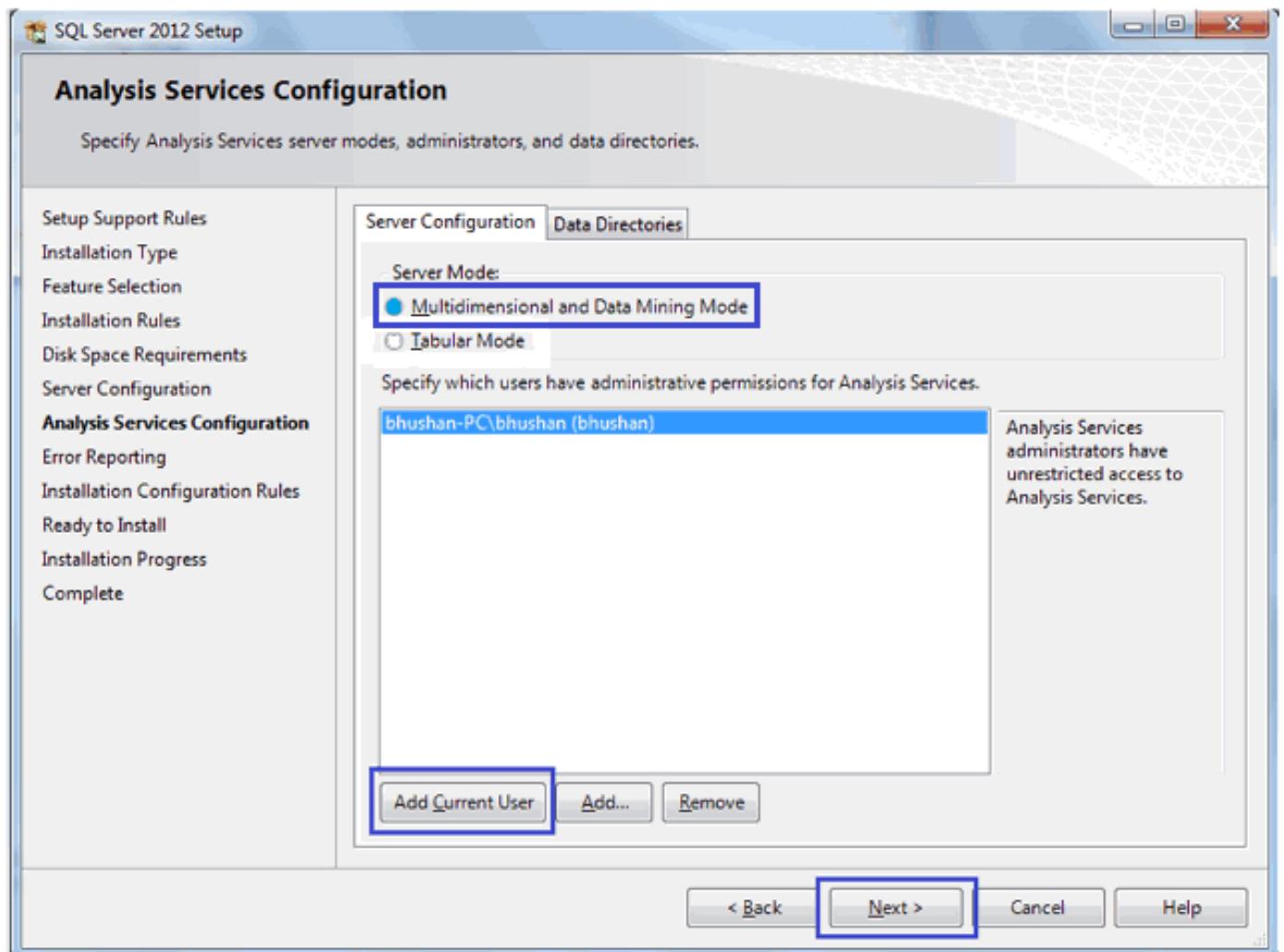
6. در صفحه‌ی بعدی برنامه‌ی نصب به شما توضیحاتی در خصوص مقدار فضای Hard برای نصب سرویس(های) انتخاب شده، نمایش می‌دهد. دکمه‌ی Next را بزنید



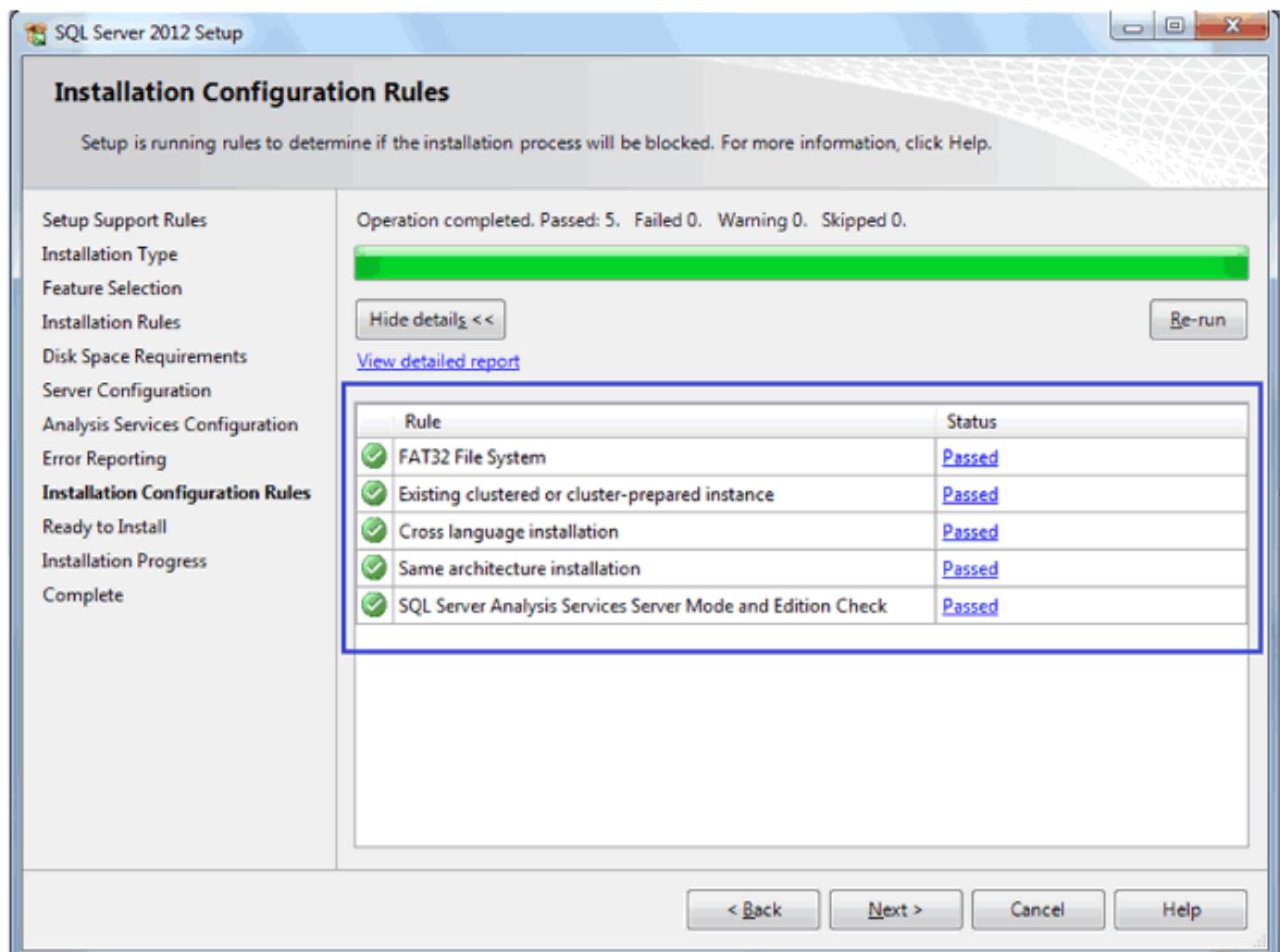
7. در صفحه‌ی SQL Server Analysis Services Server configuration مد Automatic را بر روی حالت تنظیم کنید. سپس دکمه‌ی Next را بزنید.



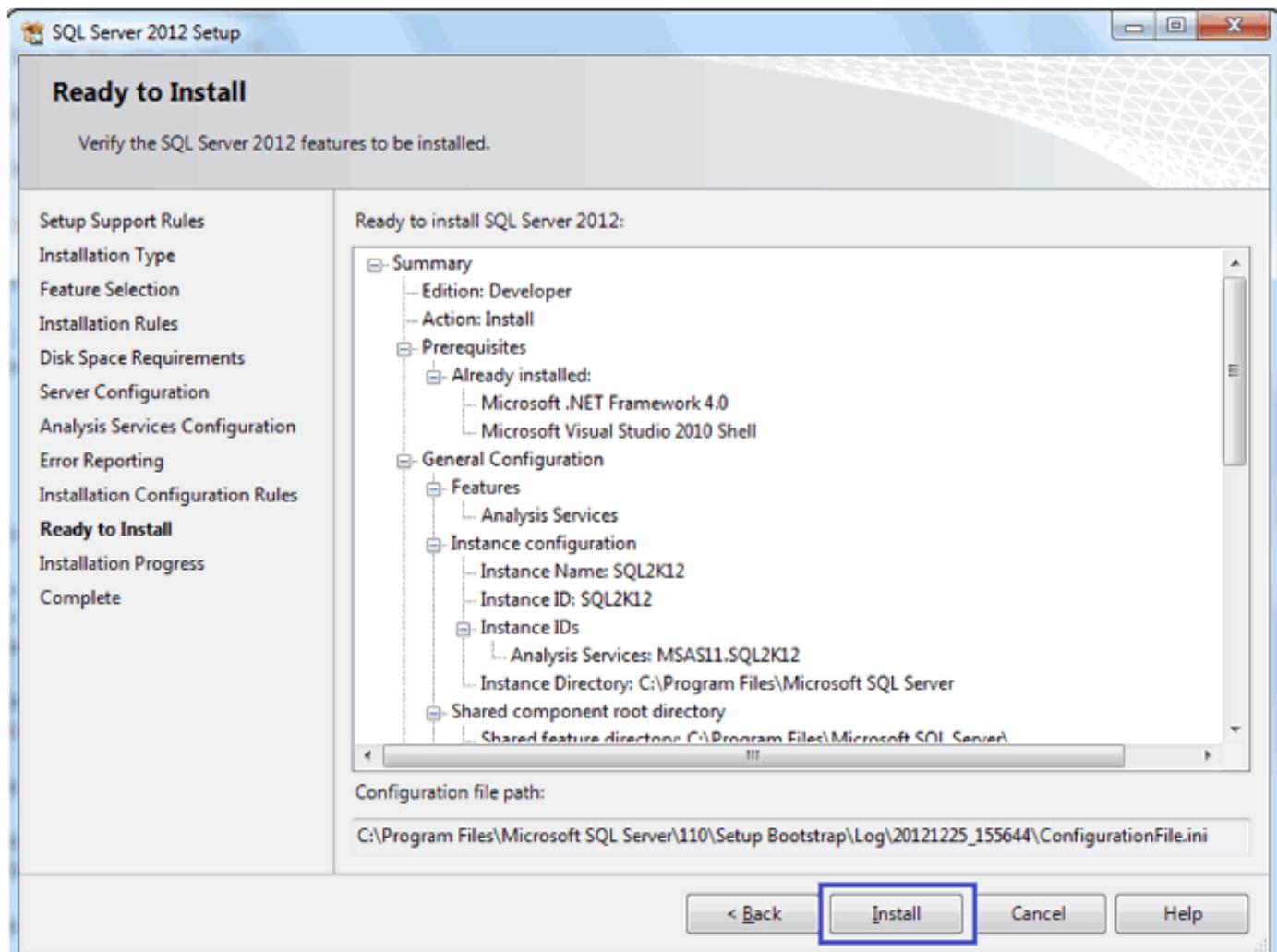
8. سپس در صفحه‌ی Analysis Services Configuration گزینه‌ی Multidimensional and Data Mining Mode را انتخاب نمایید. همچنین برای مشخص نمودن Administrator سرویس SSAS نام کاربر را در قسمت پایین پنجره وارد نمایید. در صورتی که شما با کاربری که عملاً Administrator سرویس SSAS می‌باشد در سیستم عامل ویندوز لاگین نموده اید می‌توانید دکمه‌ی Add Current User را بزنید. سپس دکمه‌ی Next را بزنید.



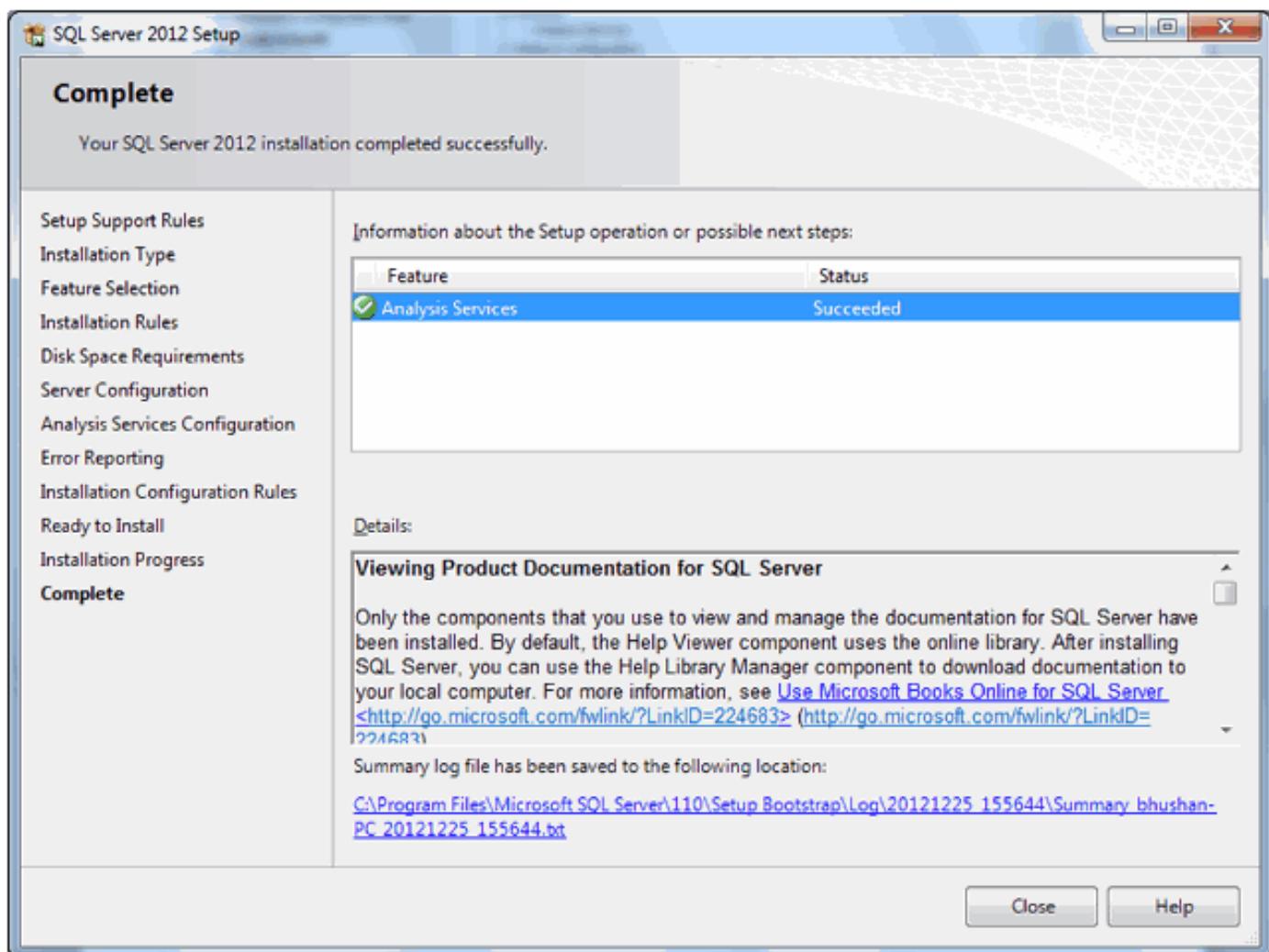
9. در صفحه‌ی بعد بررسی‌های Installation Configuration Rules انجام می‌گردد . دقت داشته باشید که تمامی موارد Passed گردیده باشند. سپس دکمه‌ی Next را بزنید .



. در صفحه‌ی Ready to install دکمه‌ی Install را بفشارید.

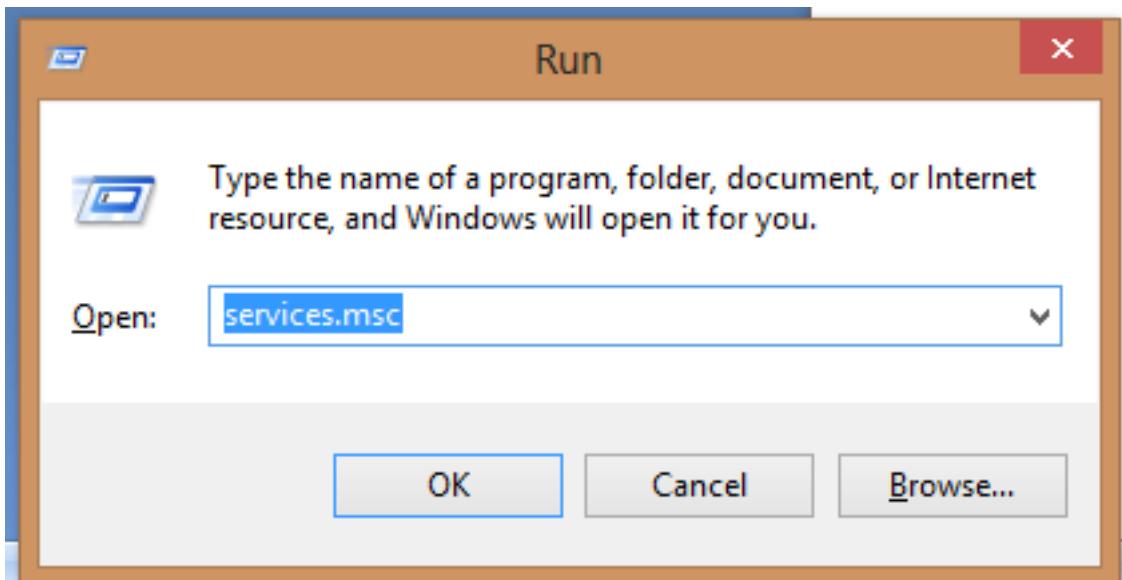


11. در صورتی که نصب با موفقیت انجام شده باشد، صفحه‌ای به شکل زیر خواهد دید.



خوب به شما تبریک می‌گوییم شما هم اکنون سرویس SSAS را برروی سرور خود نصب نموده اید. برای اطمینان از تنظیمات Registry توصیه می‌کنم سیستم عامل خود را Restart نمایید.

برای اطمینان از نصب سرویس SSAS بر روی سیستم خود می‌توانید در پنجره‌ی Run عبارت services.msc را وارد کنید.



سپس در قسمت سرویس‌ها شما می‌توانید سرویس SSAS را مشاهده نمایید مطابق شکل زیر.

SQL Full-text Filter Daemon Launcher (MSSQLSERVER)	Service to launch full-text filter daemon process which will p...	Running	Manual
SQL Server (MSSQLSERVER)	Provides storage, processing and controlled access of data, a...	Running	Automatic
SQL Server Agent (MSSQLSERVER)	Executes jobs, monitors SQL Server, fires alerts, and allows a...		Manual
SQL Server Analysis Services (MSSQLSERVER)	Supplies online analytical processing (OLAP) and data minin...	Running	Automatic
SQL Server Browser	Provides SQL Server connection information to client comp...		Disabled
SQL Server Distributed Replay Client	One or more Distributed Replay client computers that work t...		Manual
SQL Server Distributed Replay Controller	Provides trace replay orchestration across multiple Distributed Replay client computers.		

در قسمت‌های بعدی این سری از آموزش‌های MDX Query تلاش خواهیم کرد طریقه‌ی نصب پایگاه داده‌ی Adventure Work DW و همچنین ساخت پایگاه داده‌ی Multidimensional Adventure Work DW مربوط به دهم.

برای ادامه دادن این سری از مقالات آموزش MDX Query نیاز می‌باشد که پایگاه داده‌ی Adventure Work DW را نصب کرده و سپس توسط SSAS عمل Deploy را انجام دهیم تا پایگاه داده‌ی Multidimensional Database ساخته شود.

در ابتدا می‌بایست فایل نصب پایگاه داده‌ی Adventure Work را دانلود نمایید برای این منظور به آدرس زیر رفته و فایل AdventureWorks2008R2\_SR1.exe را دانلود نمایید.

[http://www.general-files.biz/download/gs4ac37d18h17i0/AdventureWorks2008R2\\_SR1.exe.html](http://www.general-files.biz/download/gs4ac37d18h17i0/AdventureWorks2008R2_SR1.exe.html)

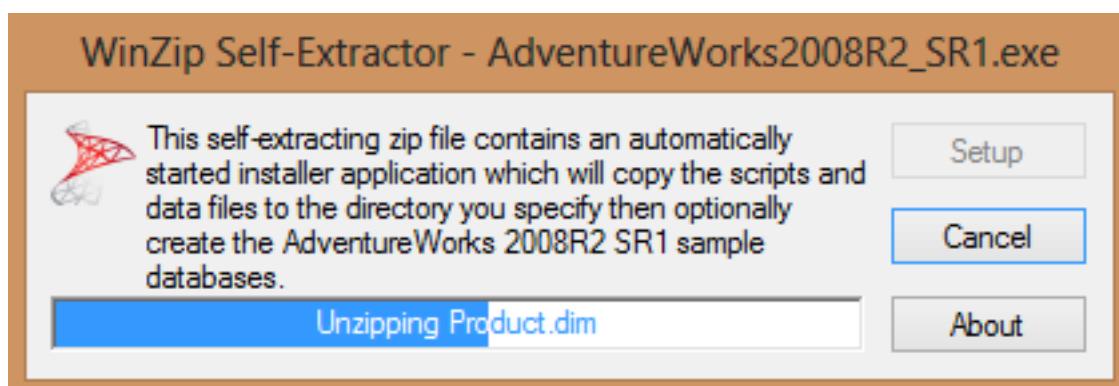
یا به آدرس زیر مراجعه کنید

<https://msftdbprodsamples.codeplex.com/releases/view/59211>

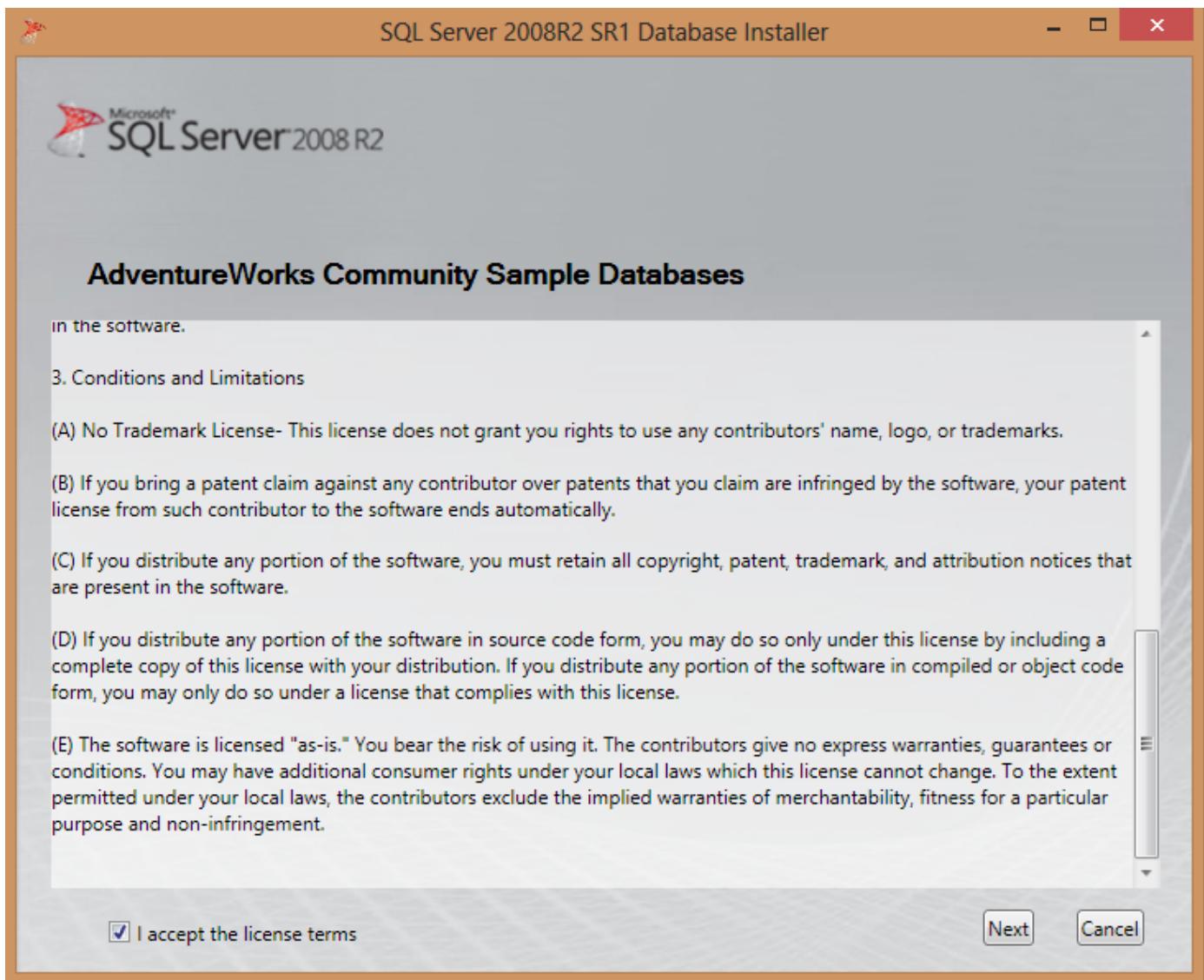
نیاز می‌باشد قبل از شروع به نصب نرم افزار SQL Server Management Studio را ببندید.

سپس مراحل زیر را انجام دهید.

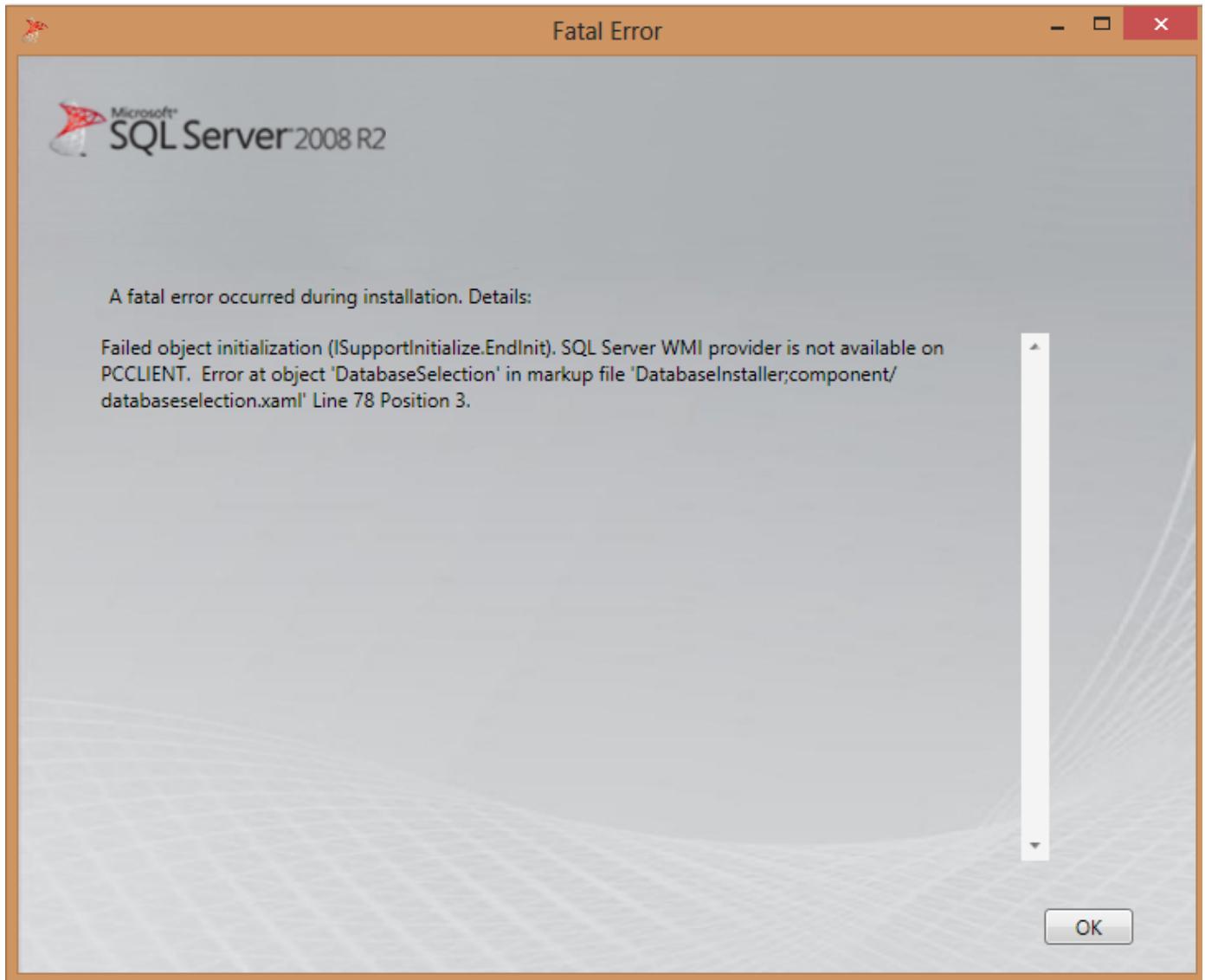
1. فایل AdventureWorks2008R2\_SR1.exe را اجرا نمایید.



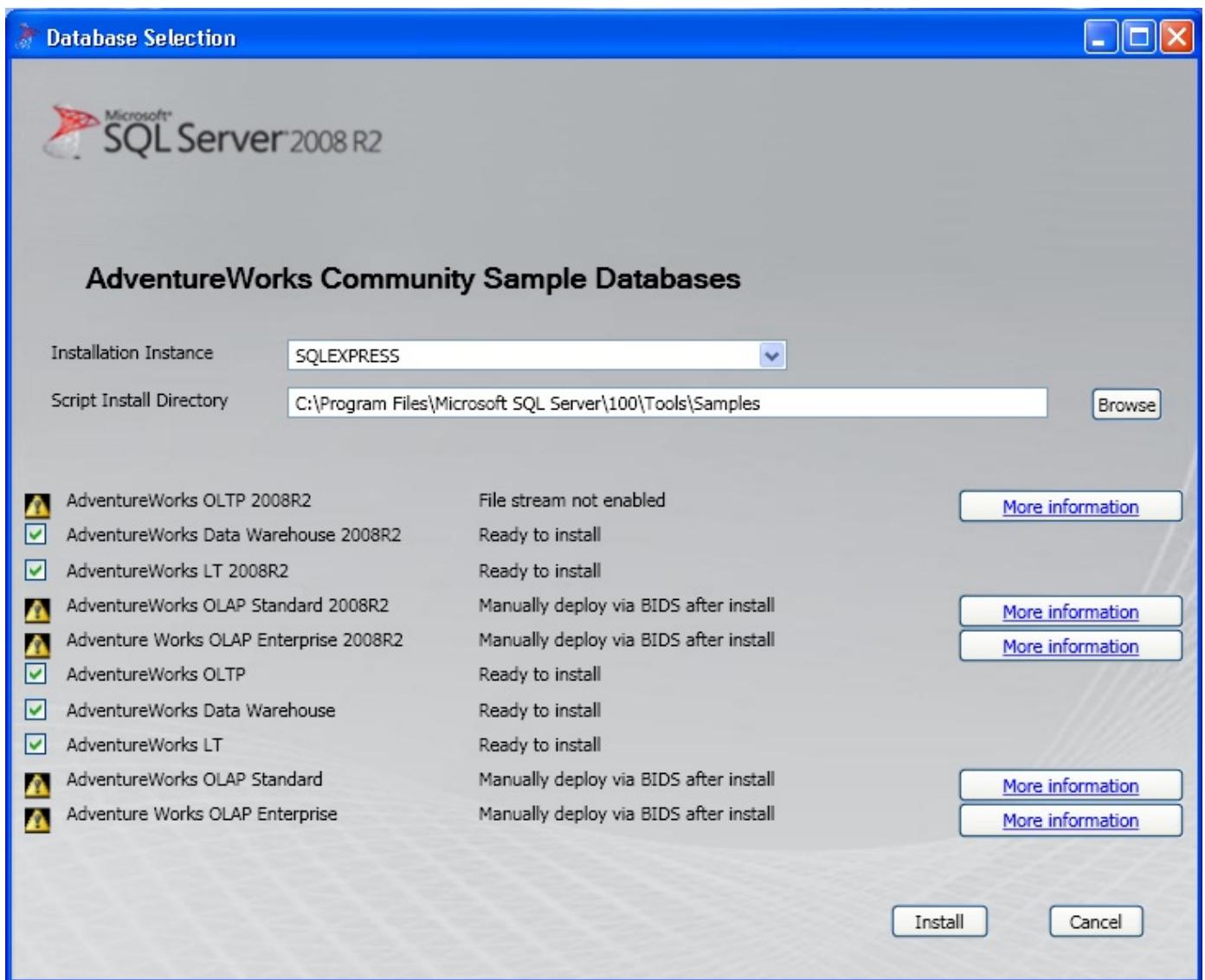
2. کمی صبر کنید تا صفحه‌ی زیر نمایش داده شود. و گزینه‌ی I Accept ... را انتخاب نماید و دکمه‌ی Next را بزنید.



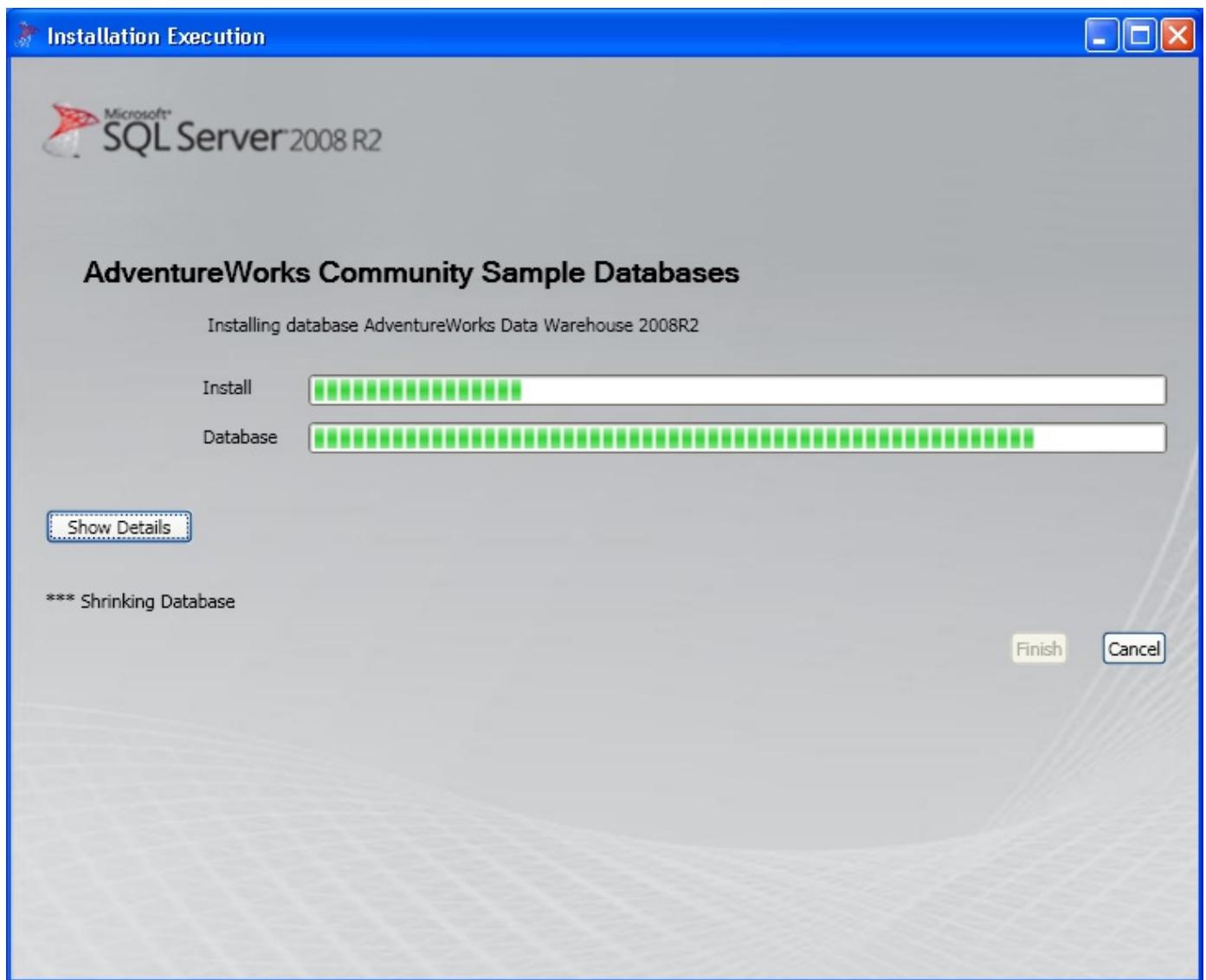
3. در صورتی که از ویندوز 8 استفاده نماید احتمال دارد با خطای زیر مواجه شوید در این صورت به قسمت روش نصب در ویندوز 8 در ادامه این مقاله مراجعه کنید .

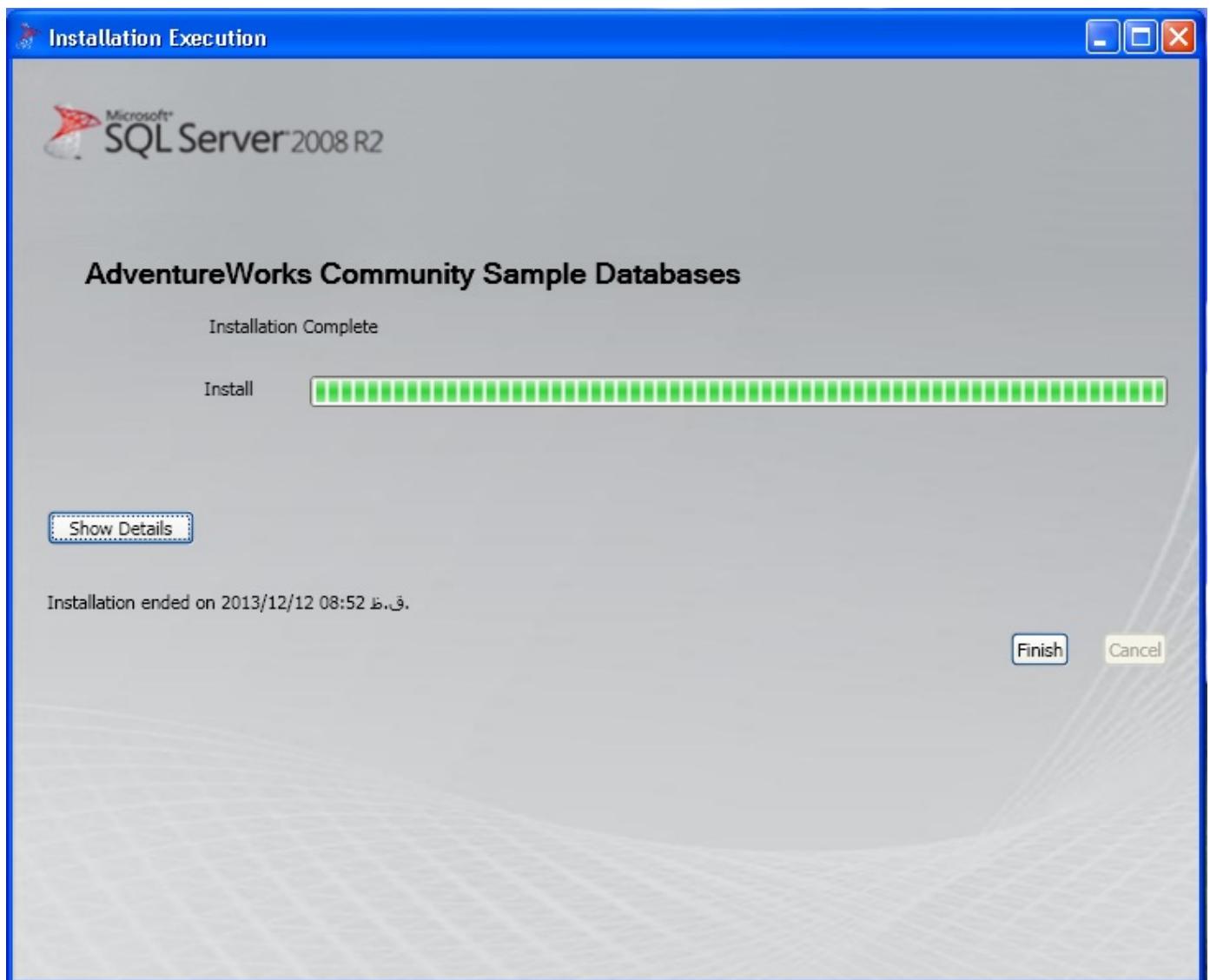


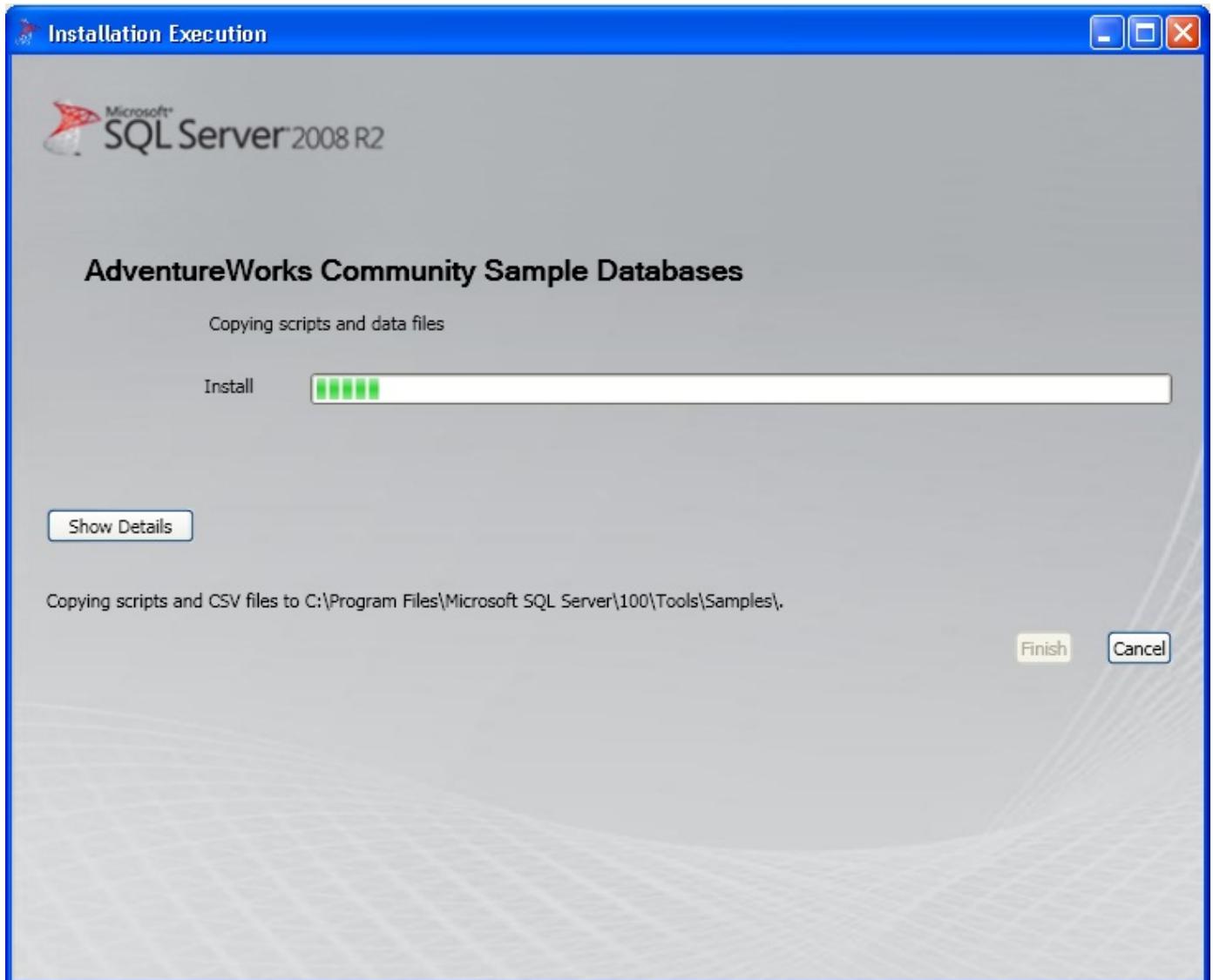
4 در صورتی که از ویندوزهای Server 2003, XP, Win7 استفاده کنید صفحه‌ی زیر را خواهید دید. در این صفحه ابتدا Instance مربوط به SQL سرور خود را انتخاب نمایید (در صورت داشتن چندین Instance روی سرور پایگاه داده) سپس مسیر نصب فایل‌های Sample را مشخص نمایید (بعدا از همین مسیر اقدام به Deploy کردن پایگاه داده‌ی Multidimensional خواهیم کرد) و پیش‌فرضها را بپذیرید و دکمه‌ی Install را بزنید.



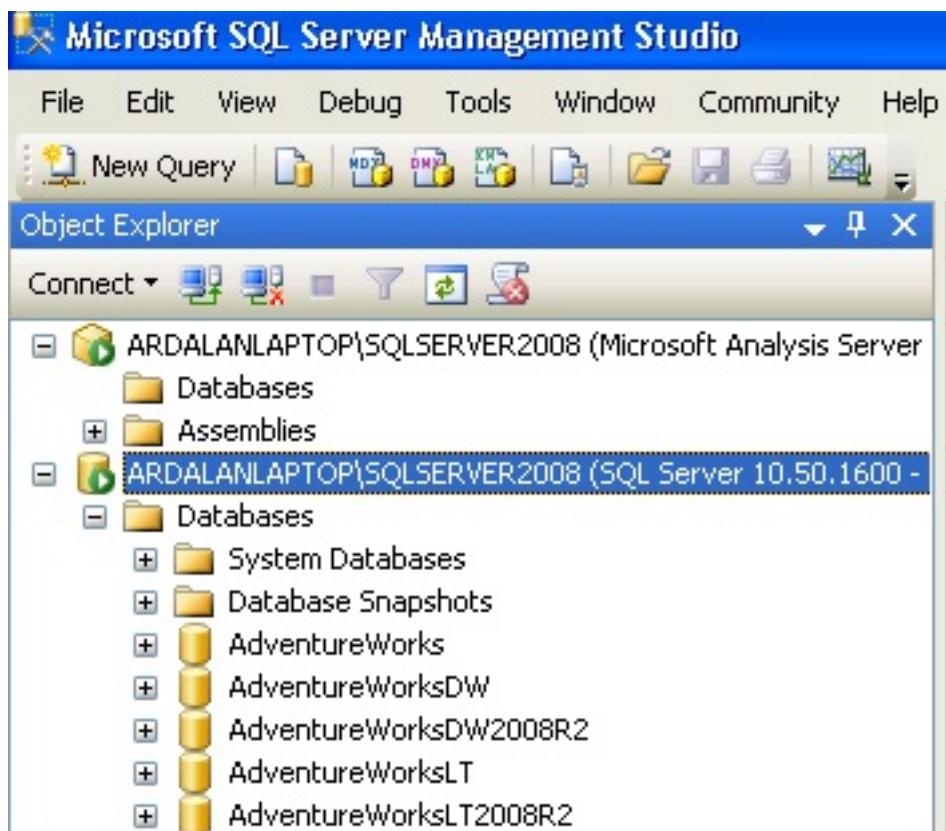
5. کمی صبر کنید تا نصب انجام گردد. و در انتها کلید Finish را بزنید.



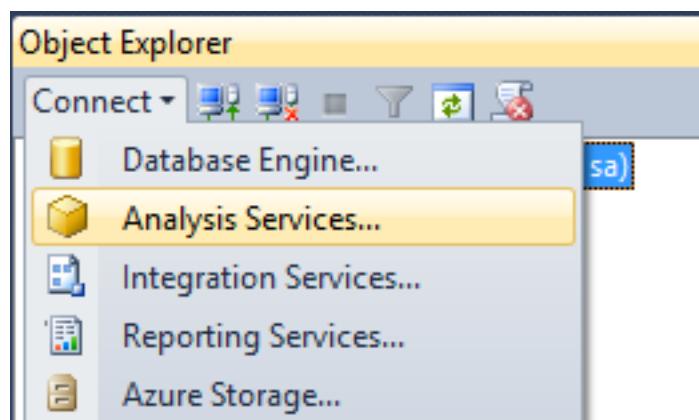




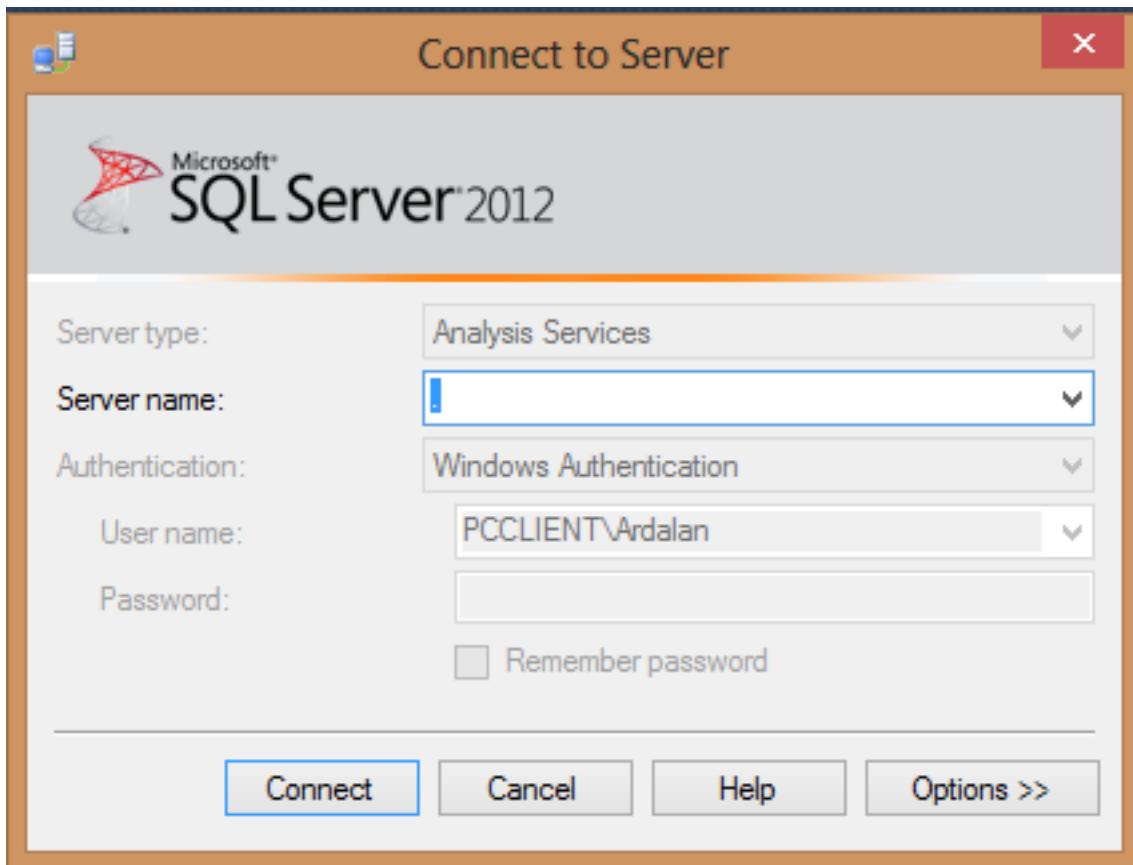
پس از مراحل بالا (به جز ویندوز 8) با باز کردن نرم افزار SQL Server Management Studio و اتصال به سرویس Database Engine در قسمت Database تصویر زیر را خواهید دید (البته امکان دارد شما از قبل دارای پایگاه داده های شخصی بوده باشید که بنابر آنها نیز در لیست شما وجود خواهند داشت)



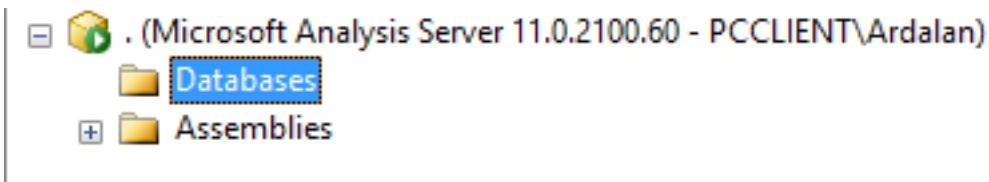
همچنین شما می‌توانید از پنجره‌ی Connect در قسمت Object Explorer اقدام به اتصال به سرویس SSAS نموده.



و در پنجره‌ی باز شده Server Name را انتخاب نمایید (با توجه به اینکه شما در حال حاضر می‌خواهید به SSAS موجود در سیستم Local متصل شوید، بنابر این انتخاب سرور Local با وارد کردن کاراکتر (. ) انجام می‌شود).



بعد از اتصال شکل زیر را خواهید داشت و در شاخه‌ی Database همچنان هیچ Multidimensional Database نخواهد داشت. (بعد از عمل Deploy که در ادامه آموزش داده خواهد شد پایگاه داده‌ی Multidimensional ساخته می‌شود).



تنها روشی که تاکنون برای نصب پایگاه داده‌ی Adventure Work DW بروی ویندوز 8 یافته ام (البته کمی غیر حرفه ای می‌باشد). به صورت زیر می‌باشد.

فایل بالا را (AdventureWorks2008R2\_SR1.exe) روی سیستم عامل‌های (Server 2003,XP,Win 7) نصب کرده (به عنوان یک سیستم عامل واسطه) و سپس سرویس Database Engine را Stop کرده و فایل‌های پایگاه داده را به سیستم عامل ویندوز 8 انتقال داده و به صورت دستی Restore کنیم.

مراحل ایجاد پایگاه داده‌ی Multidimensional در ویندوز‌های مختلف، یکسان می‌باشد.

بعد از نصب پایگاه داده‌ی Adventure Work DW باید به شاخه‌ی نصب Sample بروید (همان مسیری که در مراحل نصب وارد کردیم)

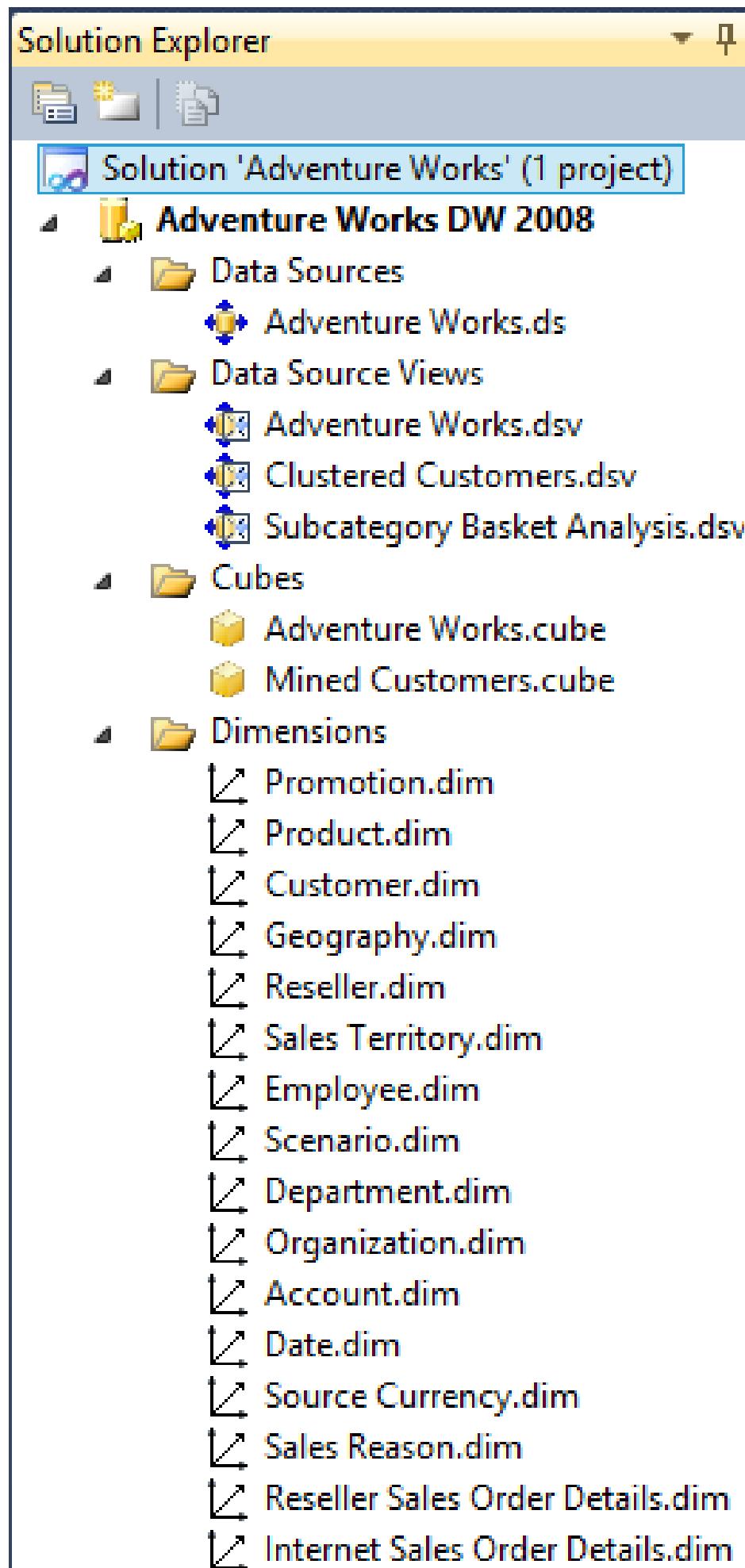
و البته آدرس پیش فرض آن C:\Program Files\Microsoft Sql Server\100\Tools\Sample می باشد).

(در صورتی که در ویندوز 8 مراحل نصب را دنبال می کنید مسیر زیر را در سیستم خود درست نمایید و فایل ها و پوشش های موجود در مسیر فوق در سیستم عامل واسط (همان سیستم عاملی که فایل نصب بر روی آن نصب شده است) را به درون آن منتقال دهید).

سپس به زیر شاخه‌ی Analysis Services Project\enterprise AdventureWorks 2008R2 بروید و فایل Works.sln را با Visual Studio 2010 باز کنید.

احتمال دارد که نیاز باشد روی کل شاخه‌ی Security در قسمت enterprise کاربر جاری را Add کنید و به آن دسترسی Full Control بدهید تا عملیات Convert این پروژه به درستی انجام شود.

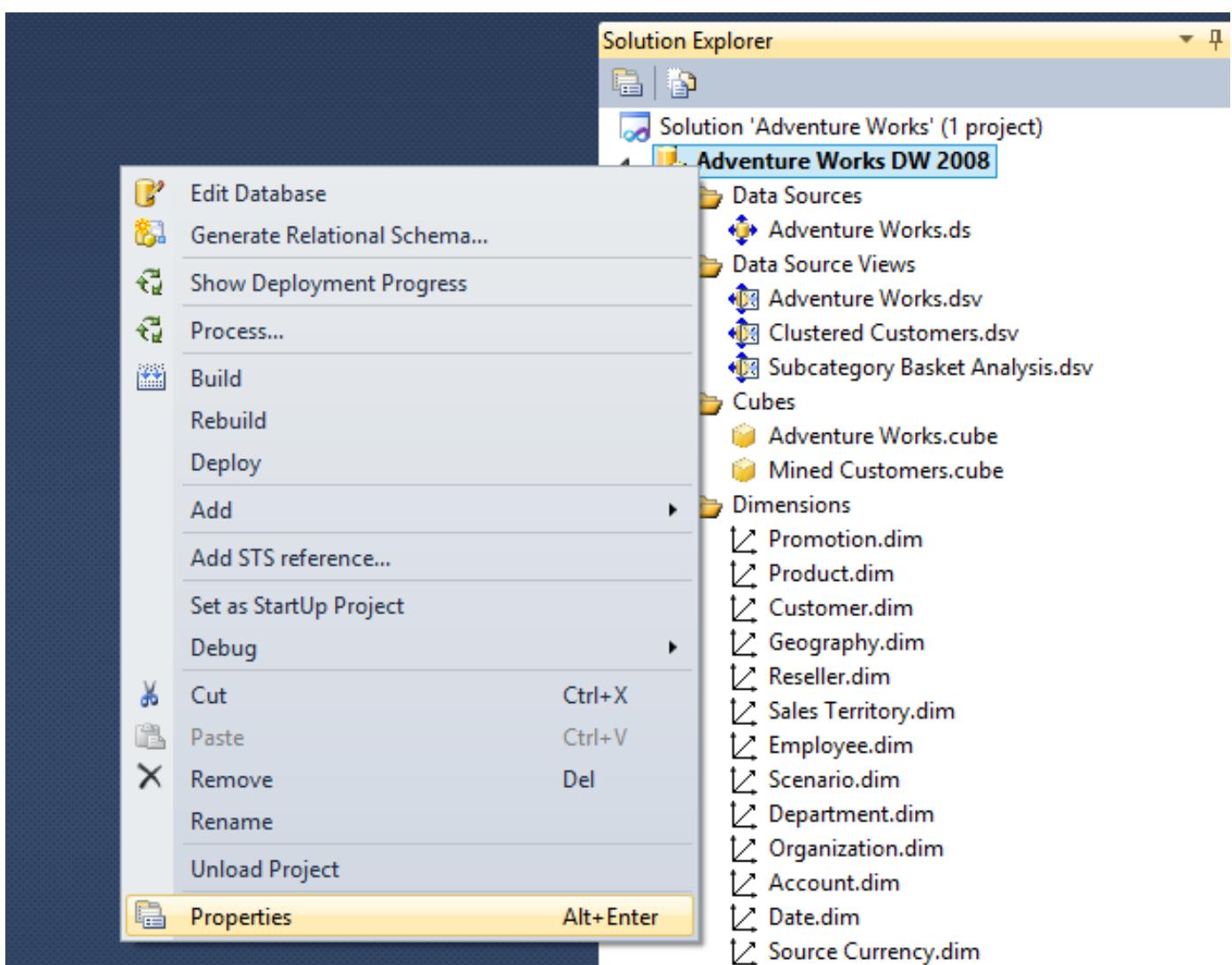
پس از باز کردن پروژه در 2010 Visual Studio صفحه ای مطابق تصویر زیر در پنجره‌ی Solution Explorer خواهد دید.



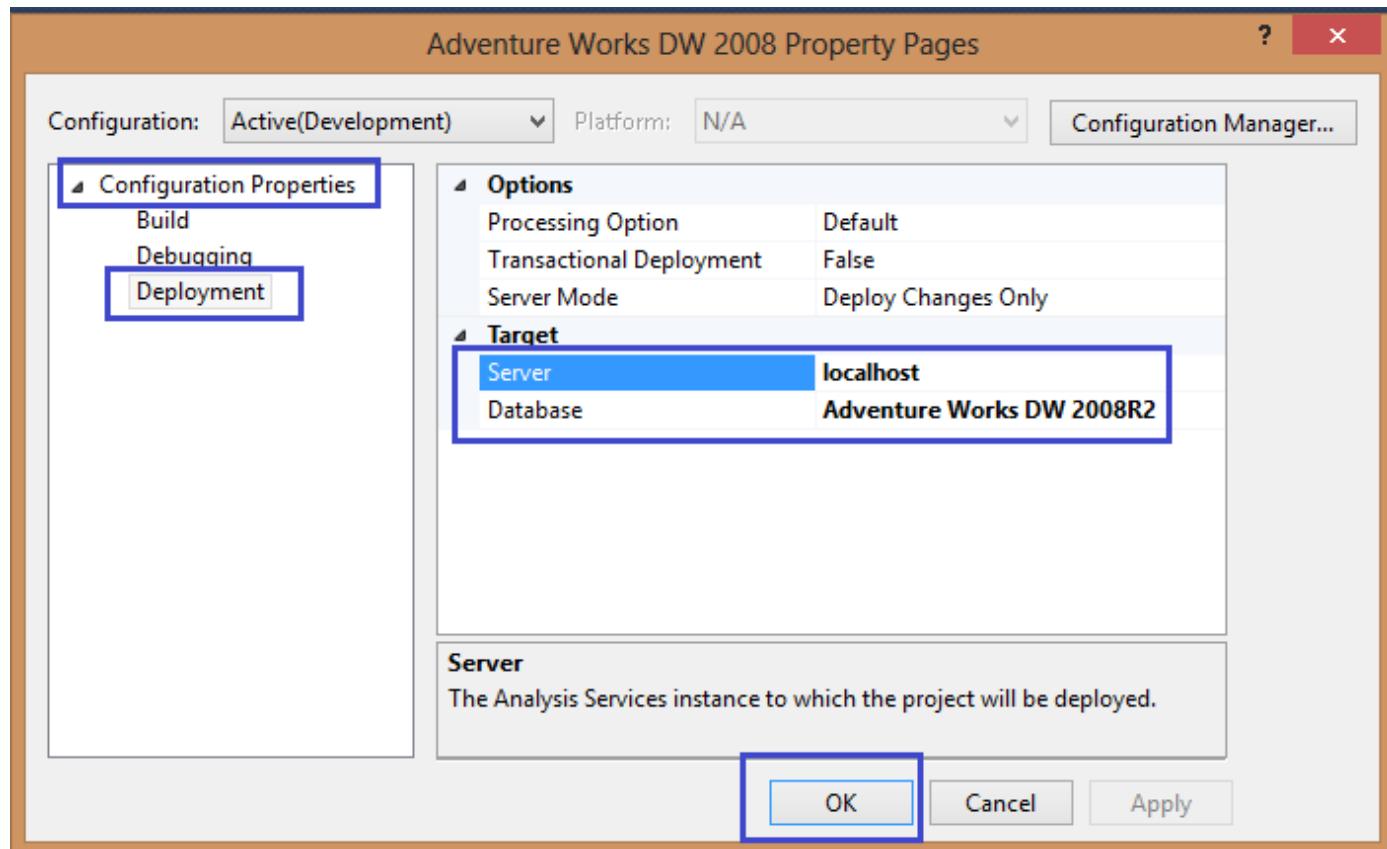
به هیچ عنوان نگران ساختار این پروژه نباشد ، زیرا در مقاله‌های آیند شرح کاملی در این خصوص کار با Business Intelligence Management Studio خواهم داد. فعلا هدف ما ایجاد پایگاه داده‌ی Multidimensional می باشد.

برای ساخت پایگاه داده‌ی Multidimensional مراحل زیر را دنبال نمایید.

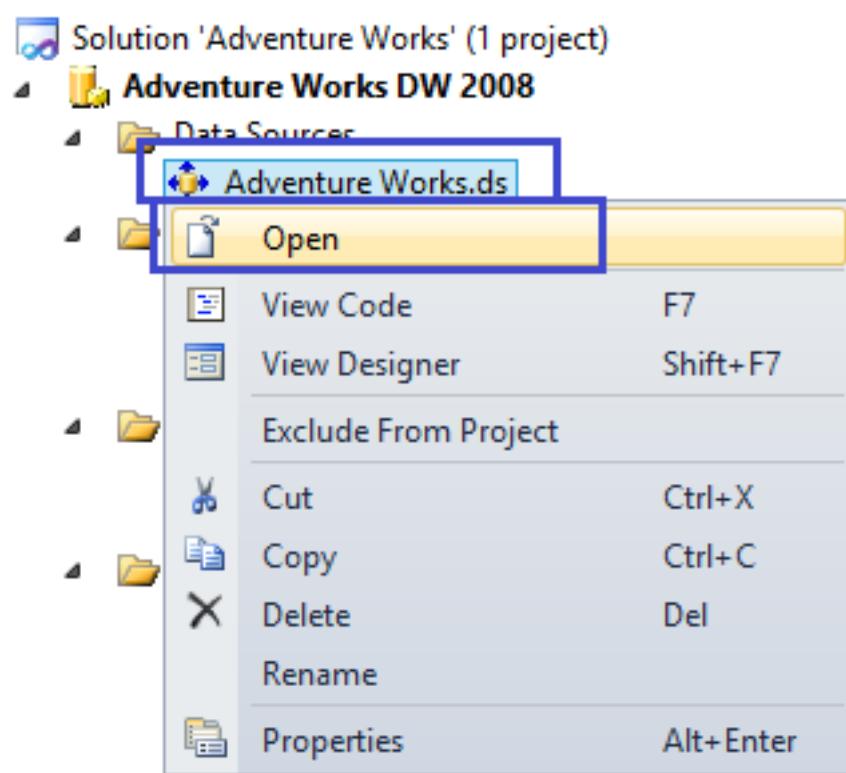
- در ابتدا روی پروژه کلیک راست کرده و گزینه‌ی Properties را انتخاب نمایید.



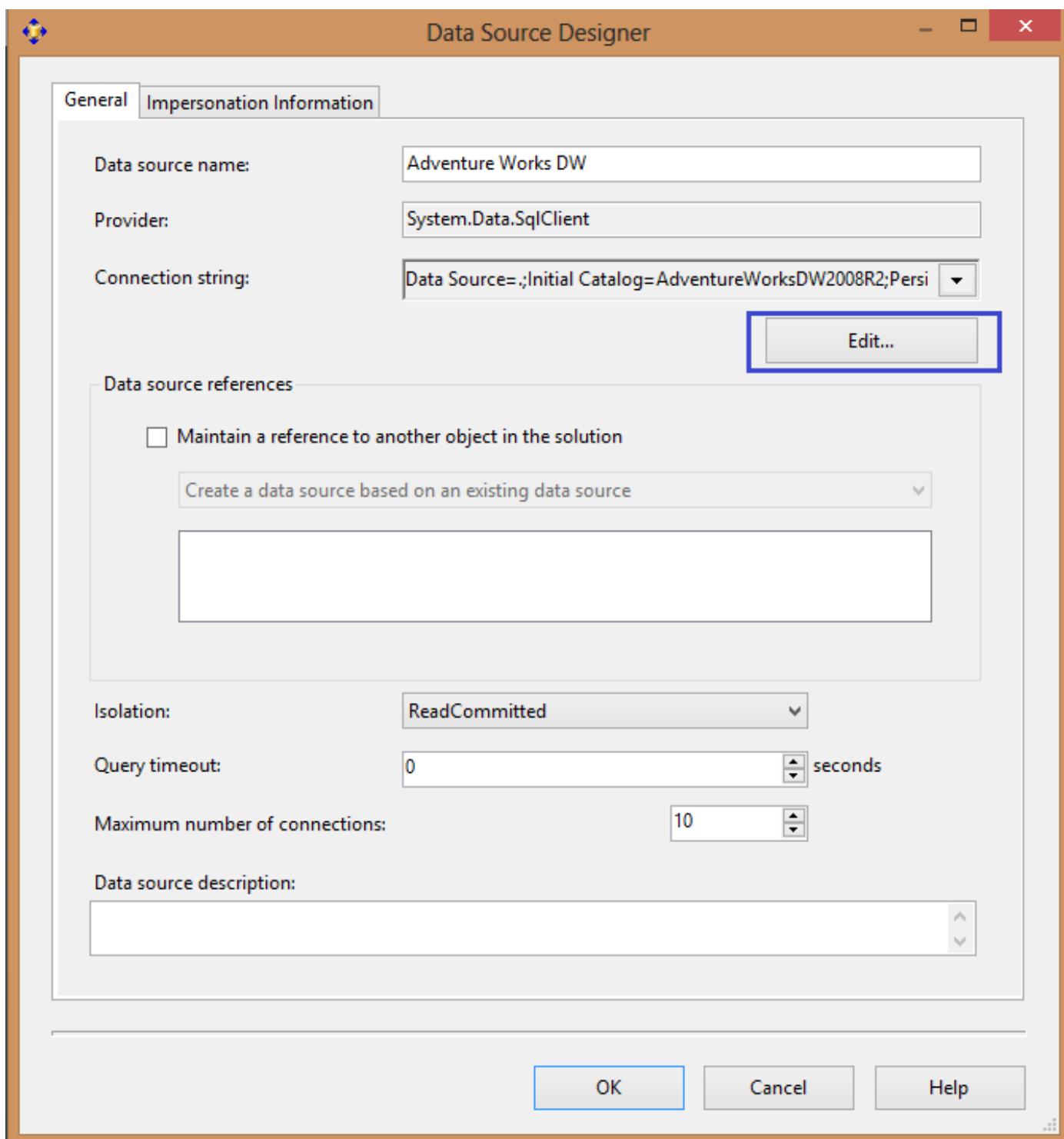
- در قسمت Deployment Configuration Properties منوی Deployment را انتخاب کرده و اطمینان حاصل کنید که سرور شما LocalHost و نام پایگاه داده شما Adventure Works DW 2008R2 باشد.



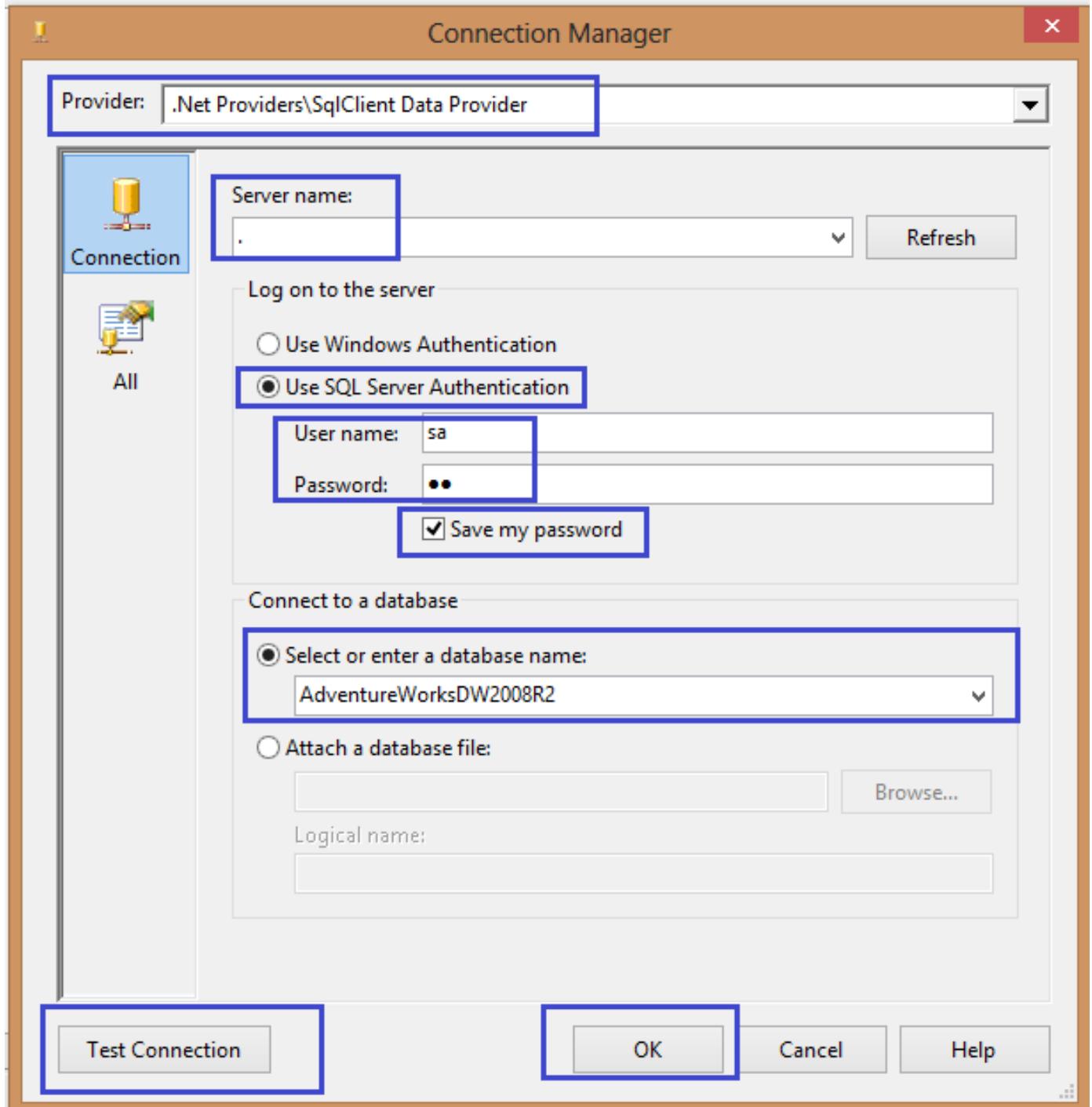
3. سپس روی کلیک راست کنید تا تنظیمات Connection String به DW را انجام دهیم. مطابق شکل زیر



4. سپس در پنجره‌ی باز شده دکمه‌ی Edit را بزنید.



5. و در صفحه باز شده تنظیمات زیر را مطابق تصویر زیر انجام دهید. دقت داشته باشید که تغییرات را از بالا به پایین باید انجام دهید و قبل از زدن دکمه‌ی OK حتما Test Connection را بزنید تا از صحت تنظیمات مطمئناً شوید.



6. سپس دو بار دکمه‌ی OK را در دو صفحه کلیک کنید. (بعد از این مراحل شما آماده‌ی Deploy کردن می‌باشد)
7. در ابتدا پروژه را Build نمایید ( CTR1 + Shift + B ) و اطمینان حاصل کنید که Build با موفقیت انجام می‌شود.
8. در انتهای بررسی نام پروژه کلیک راست نمایید و گزینه‌ی Deploy را انتخاب نمایید. فرایند Deploy کردن می‌تواند کمی زمان بر باشد بنابر این شکیبا باشید و در انتهای پیام Deployment Completed Successfully را دریافت خواهید کرد.

Deployment Progress - Adventure Works DW 2008

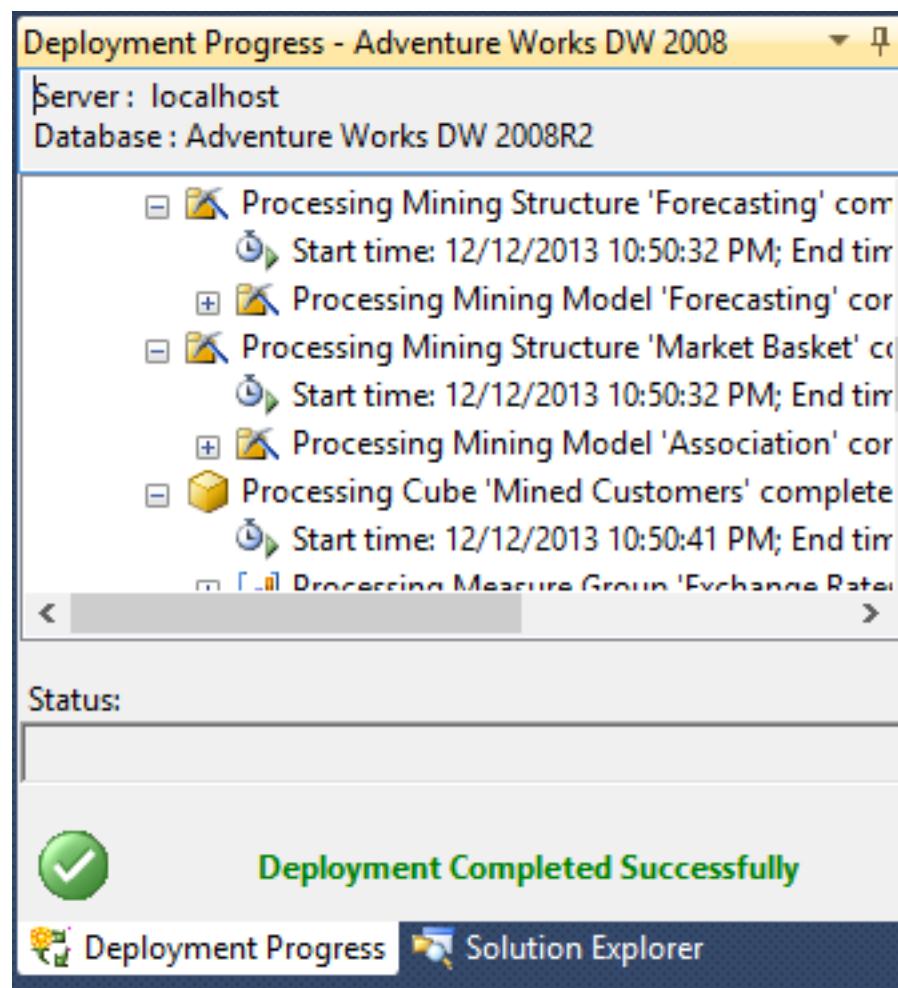
Server: localhost  
Database: Adventure Works DW 2008R2

- Processing Cube 'Adventure Works'.  
⌚ Start time 12/12/2013 10:50:32 PM
- Processing Mining Structure 'Forecasting'.  
⌚ Start time 12/12/2013 10:50:32 PM
  - + Processing Mining Model 'Forecasting'.
- Processing Mining Structure 'Market Basket'.  
⌚ Start time 12/12/2013 10:50:32 PM
  - + Processing Mining Model 'Association'.
- Processing Mining Structure 'Sequence Cluster'

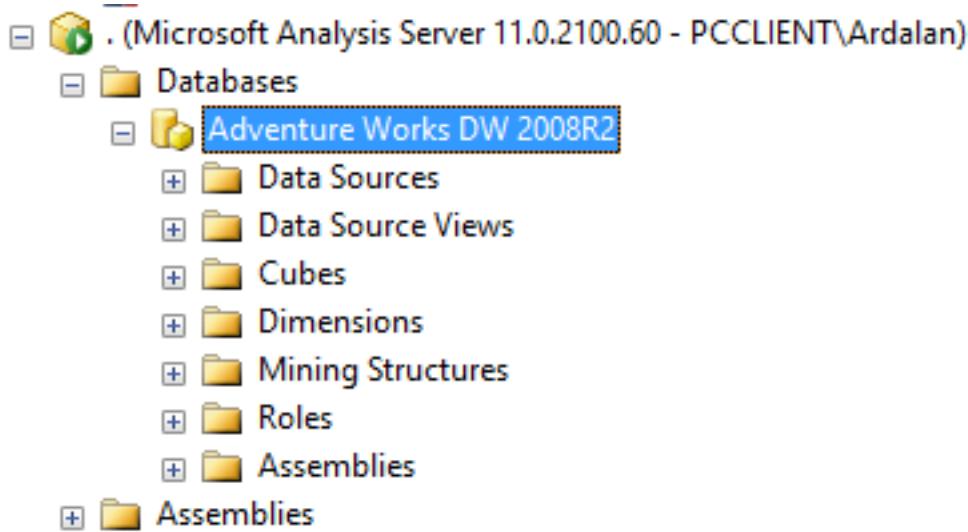
Status:  
Finished building processing schedule.

Deploying and Processing

Deployment Progress Solution Explorer



. حال به ۴ SQL Server Management Studio یک پایگاه DataBase کانکت شوید . در قسمت SSAS به سرویس می خواهد داده با نام Adventure Works DW 2008R2 مشاهده خواهد کرد .



به شما تبریک می‌گوییم اینک شما یک پایگاه داده‌ی Multidimensional را ساخته‌اید.

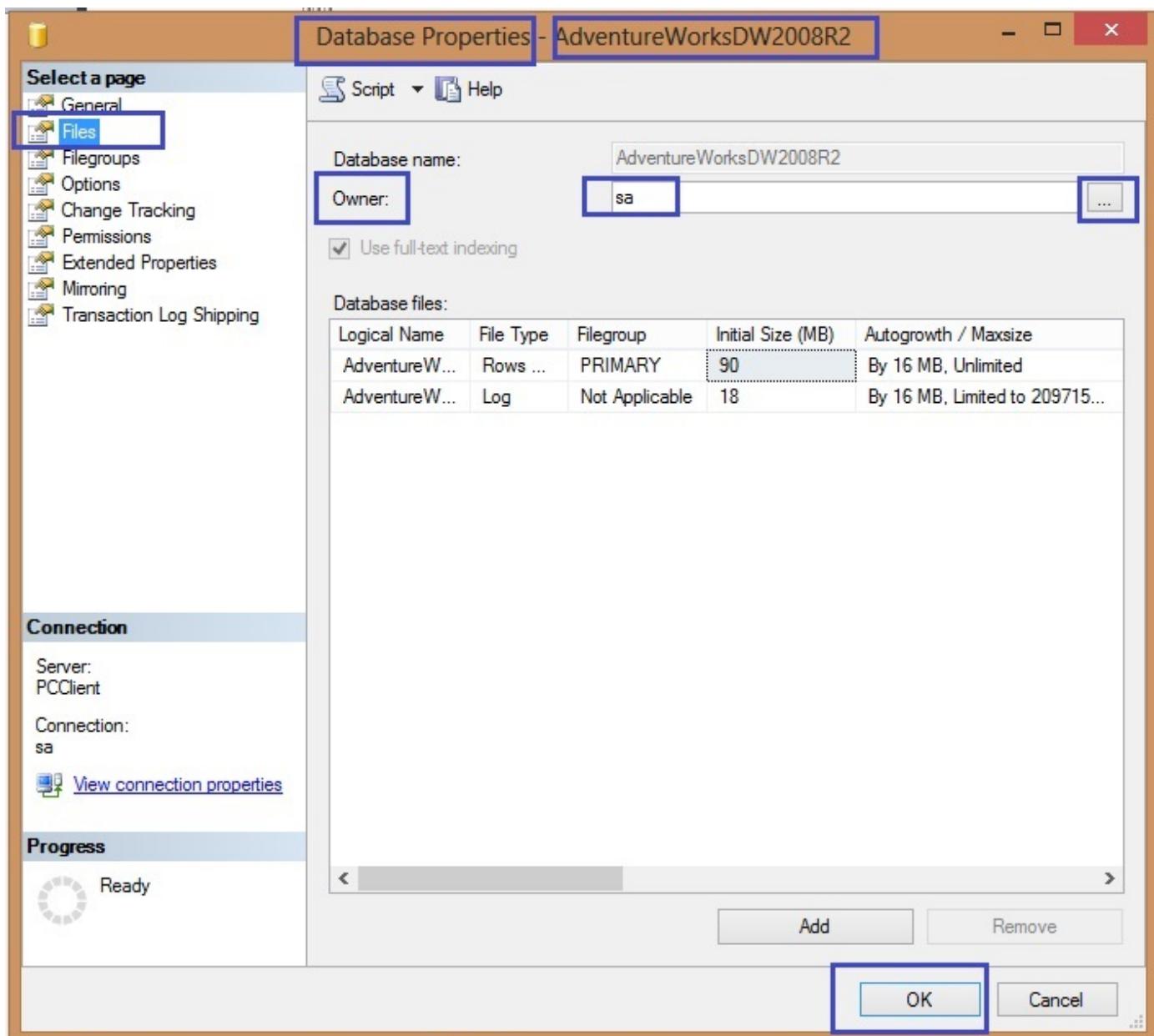
در مقاله‌ی بعدی توضیحاتی در خصوص BIMS (Business Intelligence Management Studio) خواهم داد و همچنین اولین MDX Query را خواهیم نوشت.

در این قسمت تلاش می‌کنم در خصوص محیط BIMS (Business Intelligence Management Studio) و همچنین AdventureWorksDW2008R2 توضیحاتی را ارائه کنم. در ابتدا در خصوص طراحی انجام شده در Data Warehouse مربوط به پایگاه داده‌ی Adventure Works 2008 توضیحاتی ارایه می‌گردد.

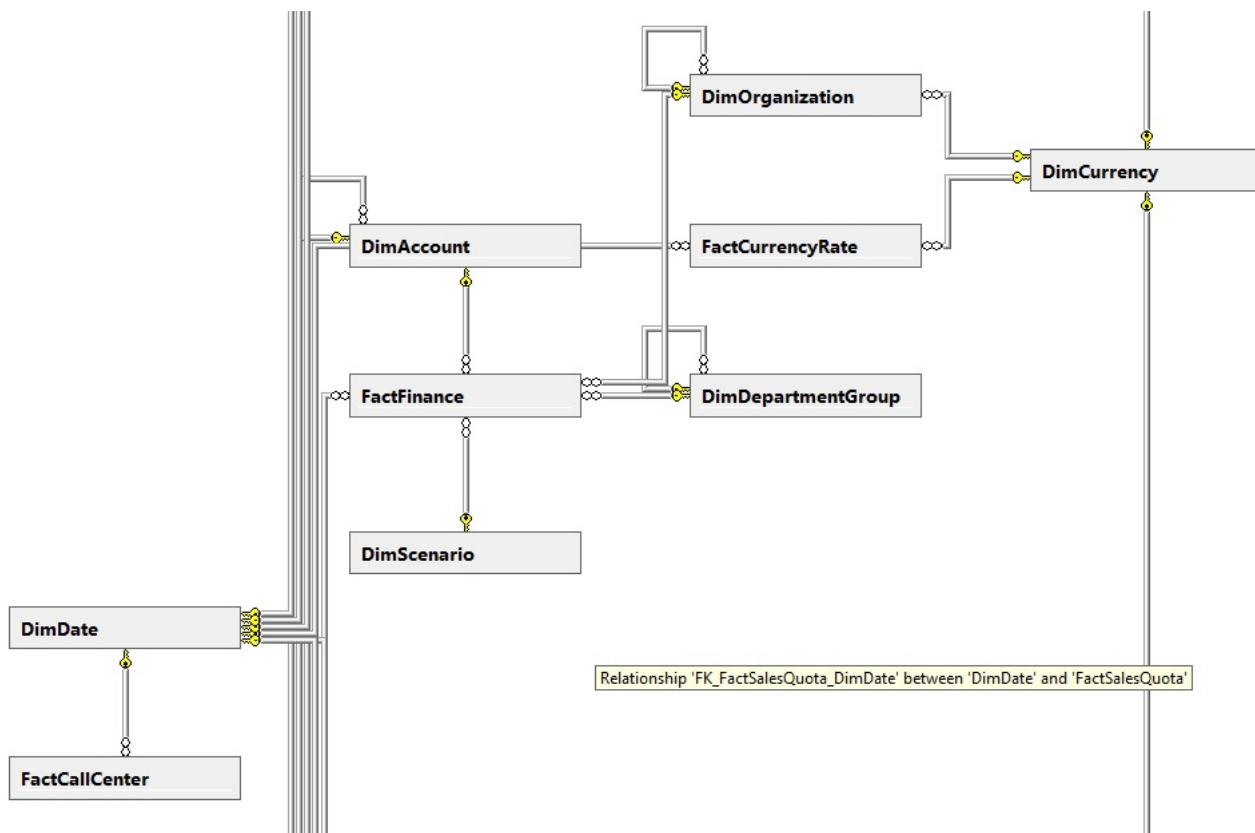
شاید بهترین کار در خصوص آشنایی با یک پایگاه داده نگاه کردن به دیاگرام کلی آن پایگاه داده باشد. بنابر این در ابتدا می‌بایست یک دیاگرام از پایگاه داده‌ی AdventureWorksDW2008R2 بسازیم (این کار را در SQL Server Management Studio انجام می‌دهیم). قبل از ساخت دیاگرام می‌بایست کاربر Sa را به عنوان Owner پایگاه داده معرفی کنیم.

برای این منظور ابتدا Properties پایگاه داده‌ی AdventureWorksDW2008R2 را گرفته و به قسمت Files رفته و با انتخاب دکمه‌ی ... در مقابل Owner و جستجوی کاربر Sa ، اقدام به مشخص کردن مالک پایگاه داده می‌کنیم. و سپس دکمه‌ی Ok را می‌زنیم.

مطابق شکل زیر



سپس یک دیاگرام کلی از پایگاه داده تولید می‌کنیم. مانند شکل زیر



با یک نگاه اجمالی مشخص می‌گردد که نام تمامی جداول پایگاه داده‌ی DW یا با کلمه‌ی Fact شروع شده‌اند.

همان طور که در مقاله‌ی [شماره‌ی یک](#) نیز عنوان شد، چندین روش طراحی DW وجود دارد:

1. ستاره ای

2. دانه برفی

3. کهکشانی

دقیق داشته باشید که جداول Fact دارای فیلدهای عددی نیز می‌باشد که توسط مراحل ETL پر شده‌اند و جداول Dimension دارای ابعادی هستند که به شاخص‌های موجود در یک جدول Fact معنا می‌دهند. به عبارت دیگر شاخص میزان فروش اینترنتی، یک Measure می‌باشد. اما با ارایه دو دایمنشن، به یک واکنش، عمل مانع یک Measure داریم که بر اساس آن دو بعد، ماهیت پیدا کرده است. به عنوان مثال میزان فروش اینترنتی بر اساس سال و ماه و روز و براساس کشور خریدار مشخص می‌شود.

یکی از روش‌های تهیه‌ی DW این می‌باشد که کاربران خبره در هر سیستم، مشخص نمایند چه گزارشاتی مورد نظر آنها می‌باشد. سپس توسط تیم پشتیبانی آن سیستم‌ها، جداول Fact, Dimension مورد نیاز برای حصول گزارش مربوطه تهیه گردد.

شاید ذکر این نکته جالب باشد که برای توسعه‌ی یک پایگاه داده Multidimensional Solution های ماکروسافت نیازی به آشنایی با یک محیط کار ( IDE ) جدید نمی‌باشد. همان طور هم که در مقاله‌ی قبلی اشاره شد، برای Deploy یک پایگاه داده‌ی چند بعدی ( Multidimensional ) از خود محیط Visual Studio .Net استفاده می‌شود. بنابر این آن دسته از برنامه نویسانی که با این محیط آشنا می‌باشند به راحتی می‌توانند به توسعه‌ی پایگاه داده‌ی چند بعدی بپردازند.

لازم به ذکر می‌باشد که اساسا هدف من از شروع این سری مقالات، آموزش MDX Query ها می‌باشد و نه آموزش BIMS ، با این

وجود در این قسمت و در قسمت بعدی، توضیحات مقدماتی کار با BIMS ارایه می‌گردد و همچنین در فرصت مناسب در خصوص یک مجموعه مقاله‌ی جامع ارایه خواهیم کرد.

در ابتدا اجزا BIMS را برای شما توضیح می‌دهم و سپس در خصوص ساخت هر کدام از آنها و ترتیب ساخت آنها توضیحاتی ارایه خواهیم داد.

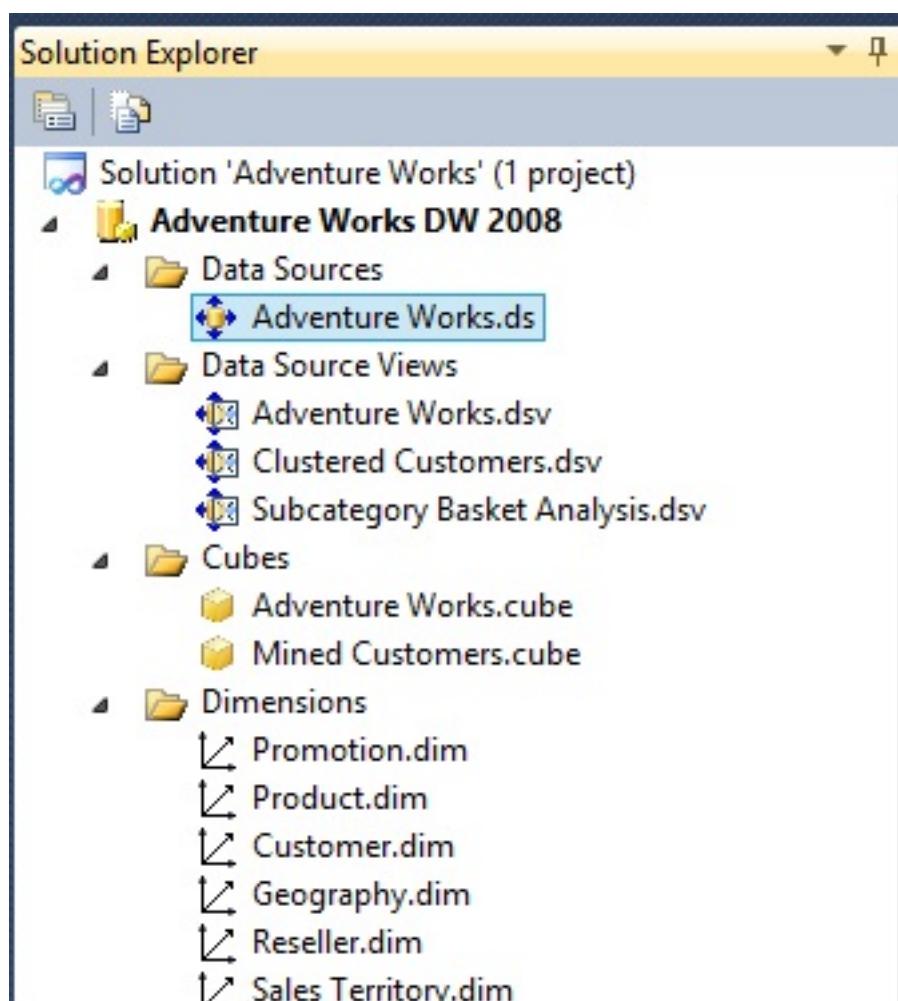
مسیر باز کردن برنامه‌ی SQL Server Business Intelligence Development Studio = BIDS در زیر آمده است:

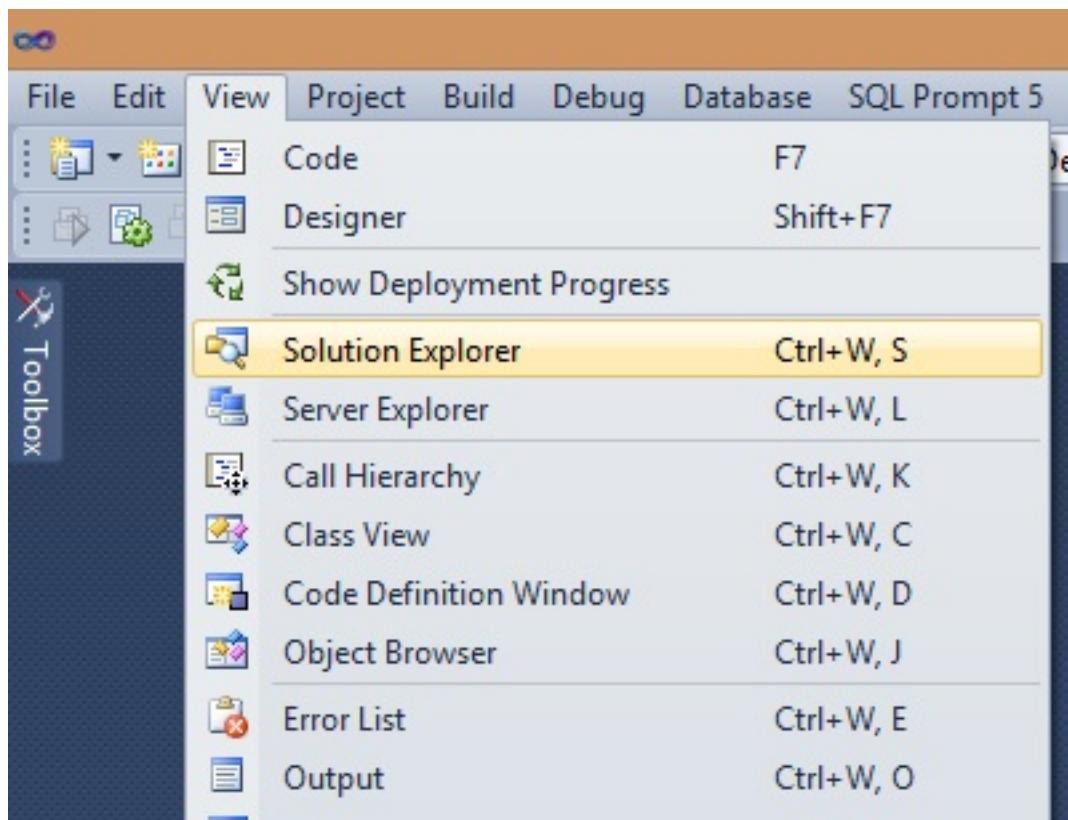
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Microsoft SQL Server 2012\ SQL Server Data Tools

دقیقت داشته باشید که در صورت استفاده از نسخه‌ی SQL Server 2008 می‌باشد مسیر زیر را جستجو نمایید:

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Microsoft SQL Server 2008 R2

با نگاه کردن به محیط BIMS می‌توانید پنجره‌ی Solution Explorer را مشاهده کنید. (در صورت عدم مشاهده، می‌توانید این پنجره را از منوی View باز کنید)





در پنجره‌ی Solution Explorer ابتدا نام Solution و در زیر آن، نام پروژه را خواهیم دید (نام پروژه و نام پایگاه داده‌ی چند بعدی، مشابه یکدیگر می‌باشد) و در زیر نام پروژه، موارد زیر را می‌بینیم:

Data Source .1

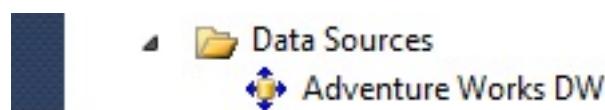
Data Source View .2

Cubes .3

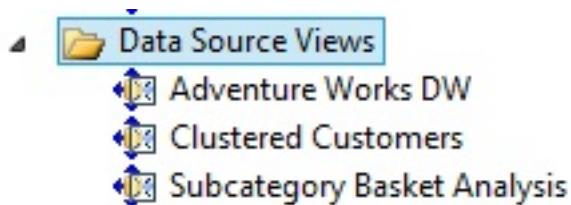
Dimensions .4

.... .5

: عملا برقرار کننده‌ی پروژه با Data Warehouse می‌باشد. دقت داشته باشید که امکان تهیه یک پایگاه داده‌ی چند بعدی از چندین DW وجود دارد و حتا نوع DW ها می‌توانند متفاوت باشد (به عبارت دیگر ما می‌توانیم چندین DW های متفاوت داشته باشیم و همه‌ی آنها را در یک Multidimensional Database تجمیع کنیم). برای انجام چنین کاری باید چندین Data Source تعریف کنیم.

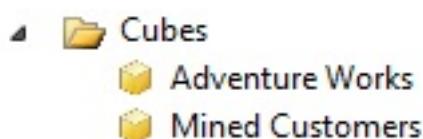


هر Data Source می‌تواند دارای چندین تقسیم بندی با مفاهیم Business باشد. برای هر کدام از این دسته‌بندی‌ها می‌توانیم یک یا چند Data Source View ایجاد کنیم. به عبارت دیگر ایجاد Data Source View‌ها سبب خلاصه شدن تعداد جداول Fact، Dimension براساس یک بیزینس خاص می‌باشد و در ادامه راحت‌تر می‌توانیم Cube‌ها را تولید کنیم.

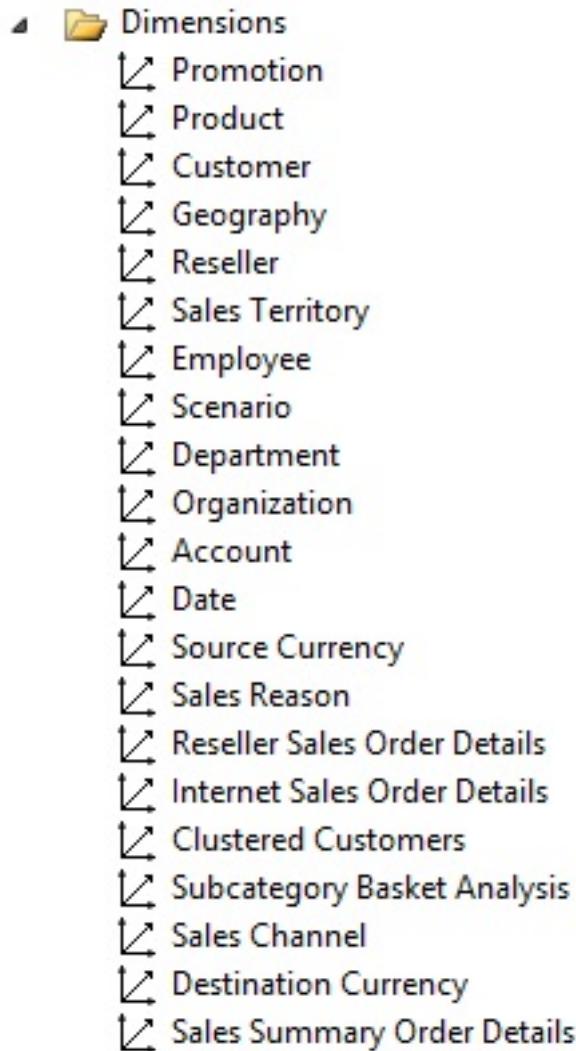


نکته: جداول Fact، Dimension در ساختار Data Warehouse ساخته می‌شوند.

محل تعریف Cube‌ها در این قسمت می‌باشد. در سری آموزش SSAS در خصوص نحوه ساخت Cube‌ها شرح کاملی ارایه خواهیم کرد.



با توجه به این که در روال ساخت Cube ما مشخص می‌کنیم چه Dimension‌هایی داریم، یک سری از Dimensions به صورت پیش فرض در این قسمت قرار می‌گیرند و البته در صورت تغییر در Data Source View می‌توانیم یک Dimension را به صورت دستی در این قسمت ایجاد نماییم و سپس آن را به Cube مورد نظر اضافه نماییم.

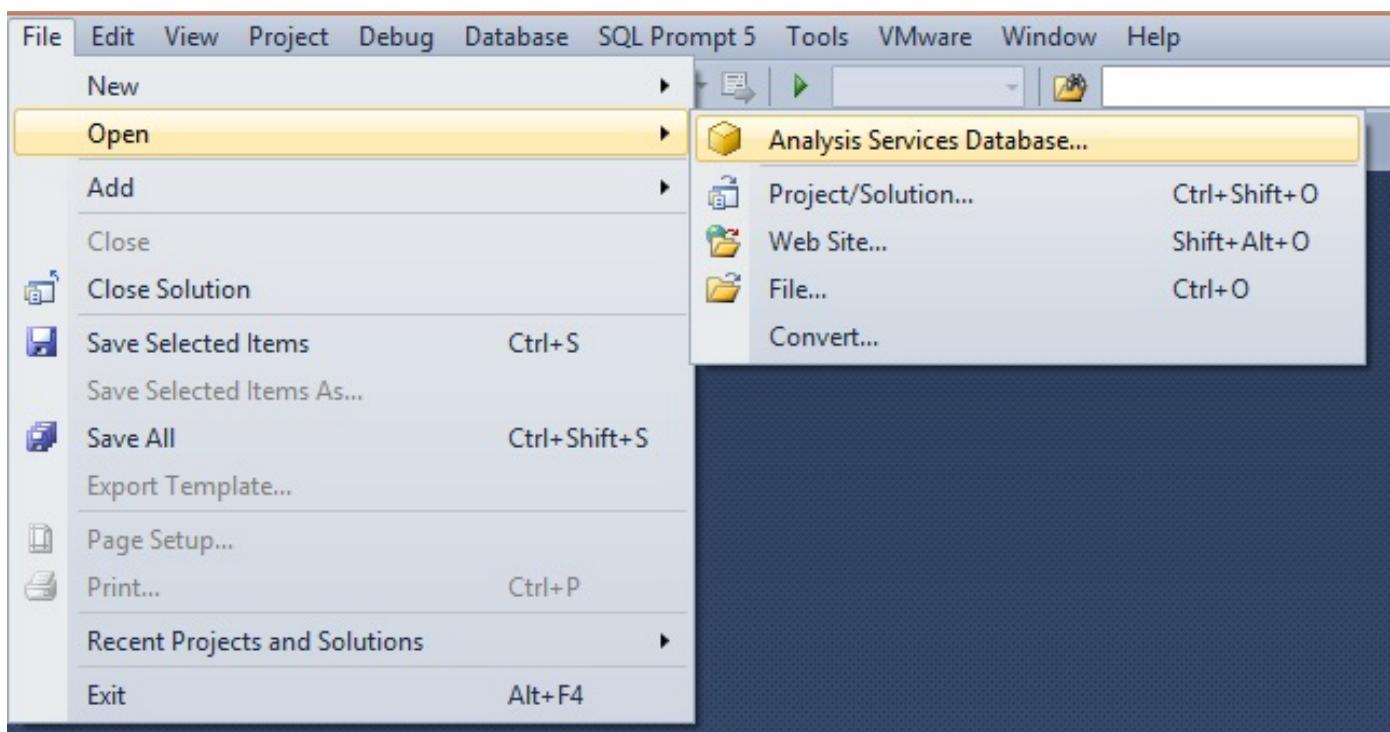


دقت داشته باشید که برای ساخت یک پروژه می‌بایست بعد از ساخت Data Warehouse در برنامه‌ی BIMS اقدام به ساخت یک Data Source View OLTP اقدام به ساخت Business گنیم و سپس با توجه به OLTP های موجود در سیستم‌های مناسب Hierarchy و ... نیاز KPI را در نهایت اقدام به ساخت Cube کنیم. بعد از انجام تنظیمات مختلف در Cube مانند ساخت Cube Deploy کرده و در پایگاه داده‌ی چند بعدی (MDB) ساخته شود.

در قسمت بعدی نحوه‌ی ساخت یک پروژه در SSAS و چگونگی باز کردن یک پایگاه داده را بررسی خواهیم کرد.

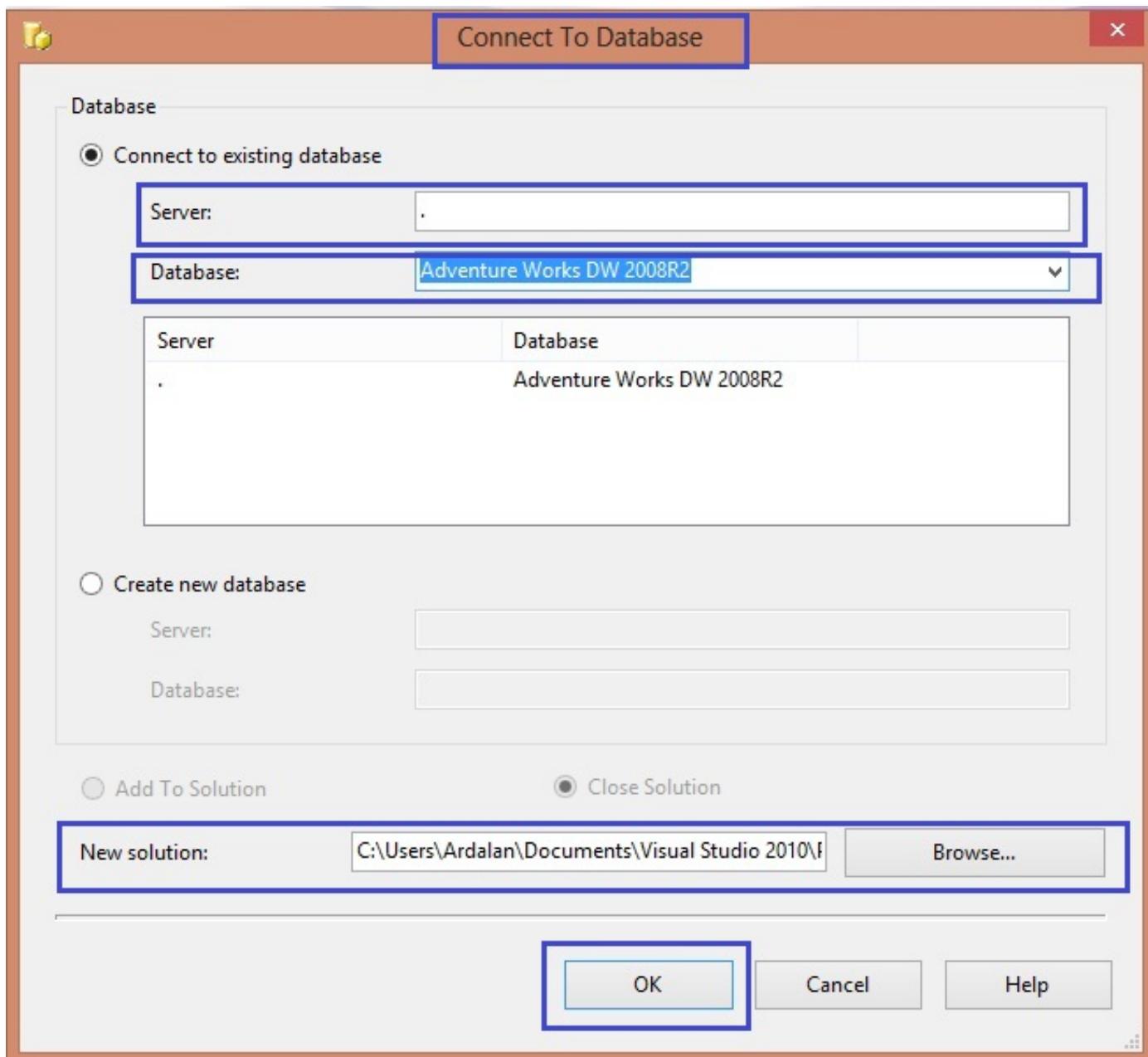
در این قسمت در ابتدا نحوه‌ی باز کردن یک پایگاه داده‌ی چند بعدی را در محیط BIMS بررسی کرده و سپس چگونگی ساخت یک MDB را از پایه بررسی می‌کنیم. برای ادامه دادن این قسمت نیاز می‌باشد که پایگاه داده‌ی AdventureWorkDW2008 را در SSAS نصب کرده باشید.

در ابتدا مطابق شکل زیر منوی File سپس زیر منوی Open را انتخاب نمایید.

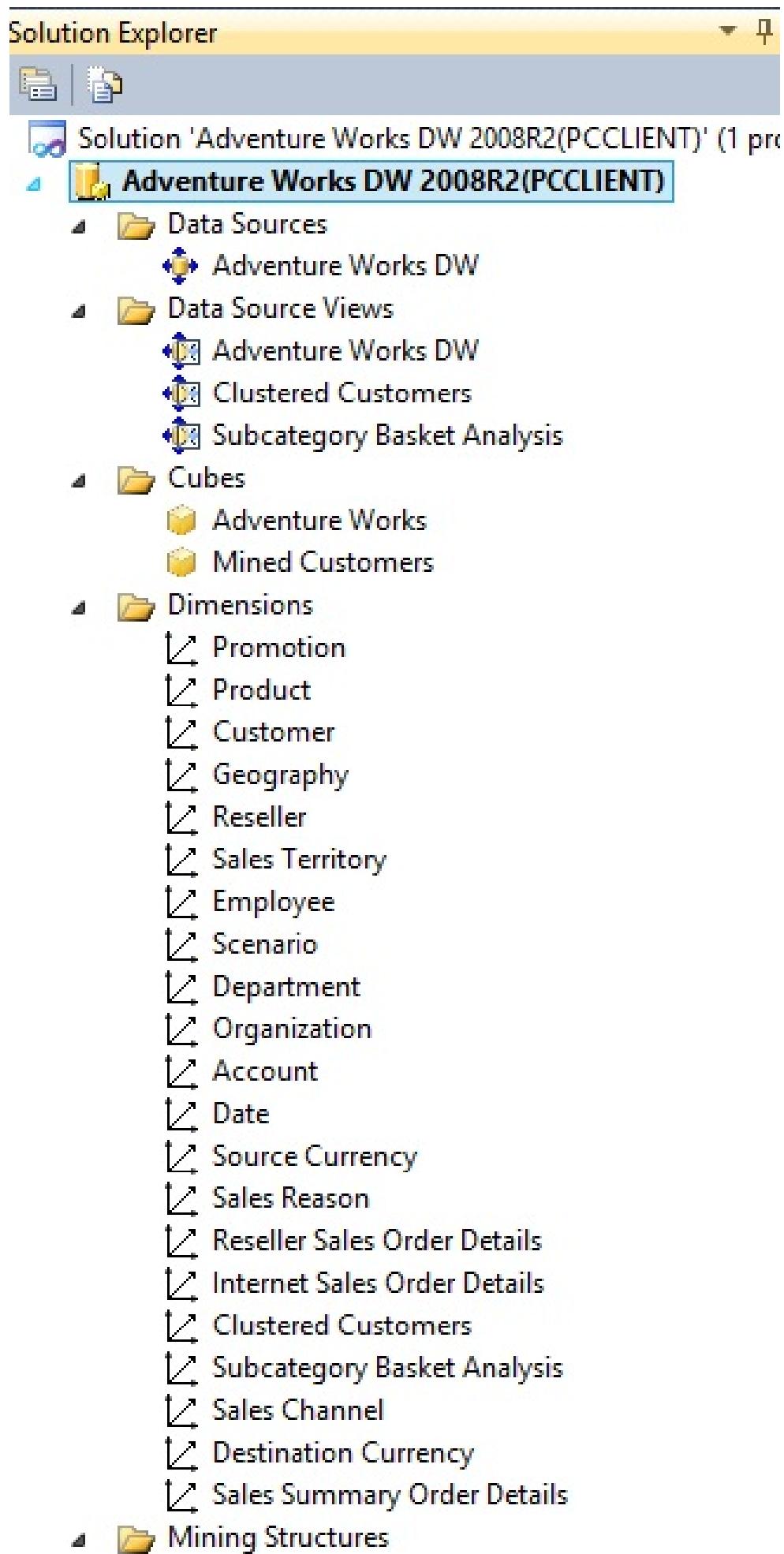


در ادامه می‌بایست نام Server را مشخص نمایید و دقت داشته باشید که در اینجا منظور از نام سرور، نام سرور SSAS می‌باشد (در صورتیکه بر روی خود سرور در حال کار می‌باشید از . به جای نام سرور استفاده کنید). سپس در قسمت Database ، نام پایگاه داده‌ی چند بعدی را انتخاب نمایید. در صورتی که به جز Adventure Work DW 2008 ، پایگاه داده‌های چند بعدی دیگری را در SSAS داشته باشید، یک لیست از آنها را مشاهده خواهید کرد و در صورتیکه لیست شما خالی می‌باشد، احتمال دارد نام سرور اشتباه باشد یا روی سرویس SSAS مربوط به آن سرور هیچ پایگاه داده‌ی چند بعدی نصب نباشد.

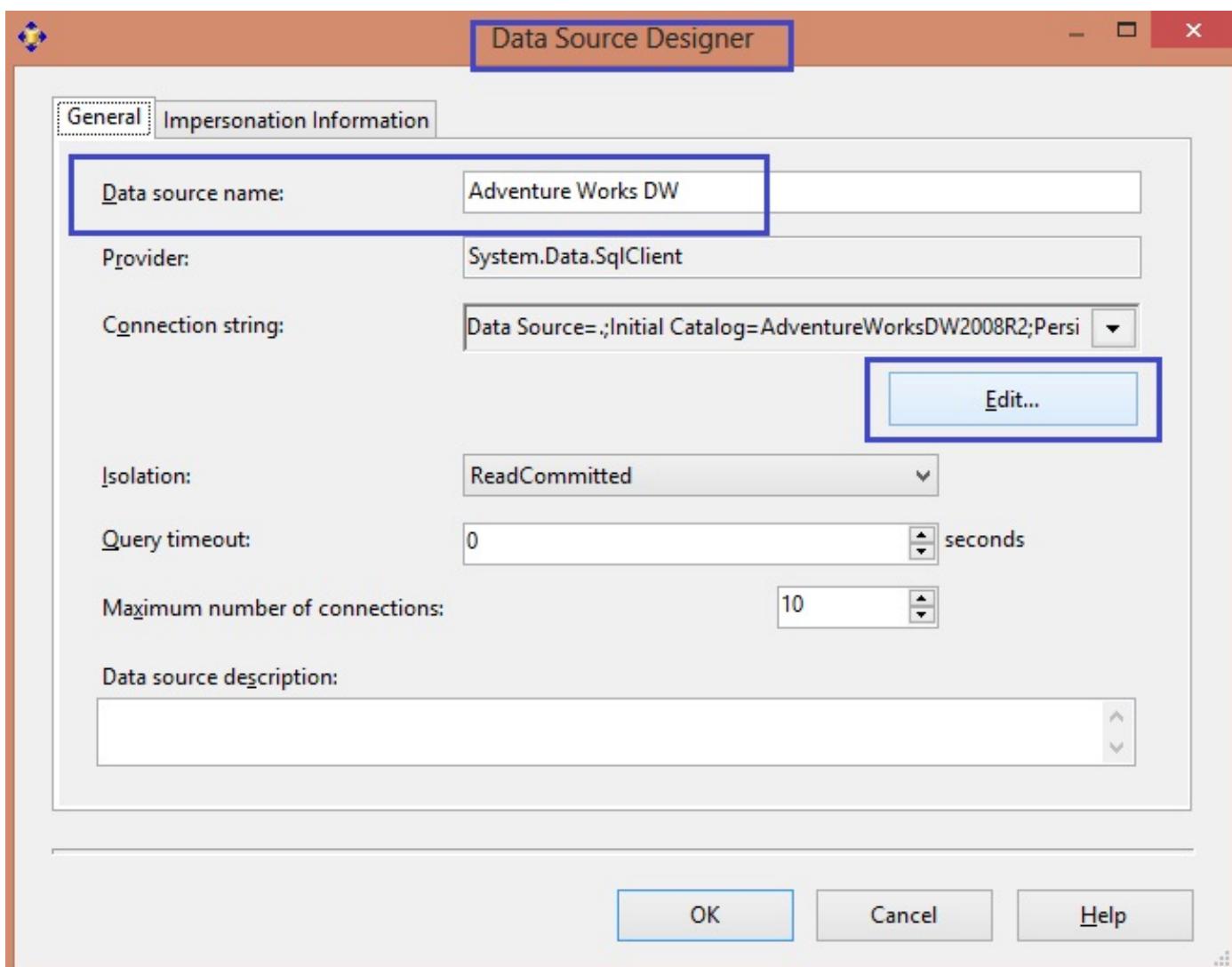
حال مسیری را برای ذخیره سازی پروژه جدید در نظر بگیرید:



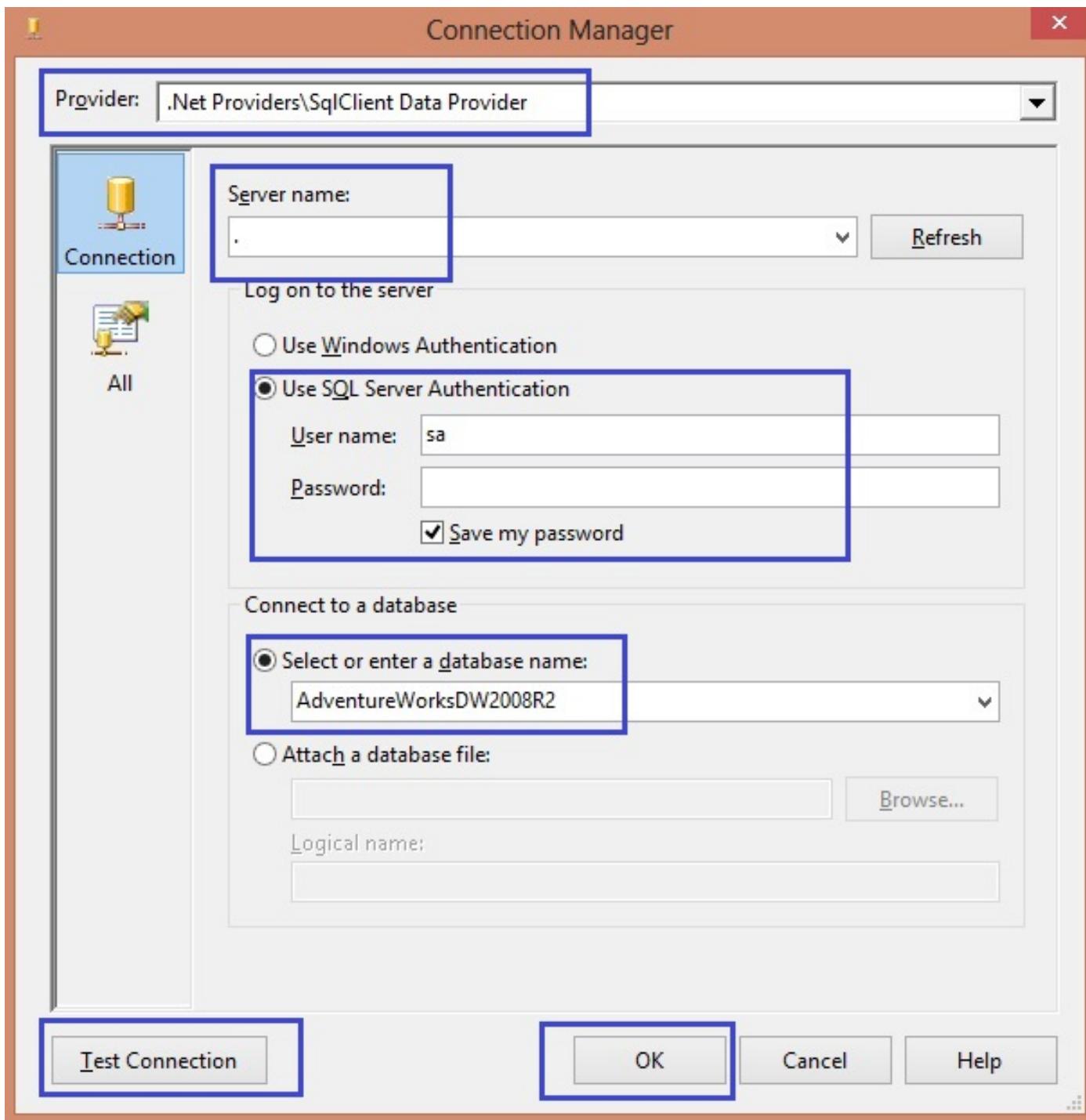
پس از کمی شکیبایی، واکنش اطلاعات از روی پایگاه داده‌ی چند بعدی انتخاب شده انجام می‌شود و یک پروژه در ارتباط با آن پایگاه داده ساخته می‌شود.



همان طور که مشخص می‌باشد، یک شیء درون شاخه‌ی Data Source وجود دارد که مشخص کننده ارتباط این پروژه با پایگاه داده‌ی Data Warehouse است. برای مشاهده این ارتباط، بر روی Adventure Work DW کلیک راست کنید و سپس گزینه‌ی Open را انتخاب نمایید. در ادامه گزینه‌ی Edit را بزنید.



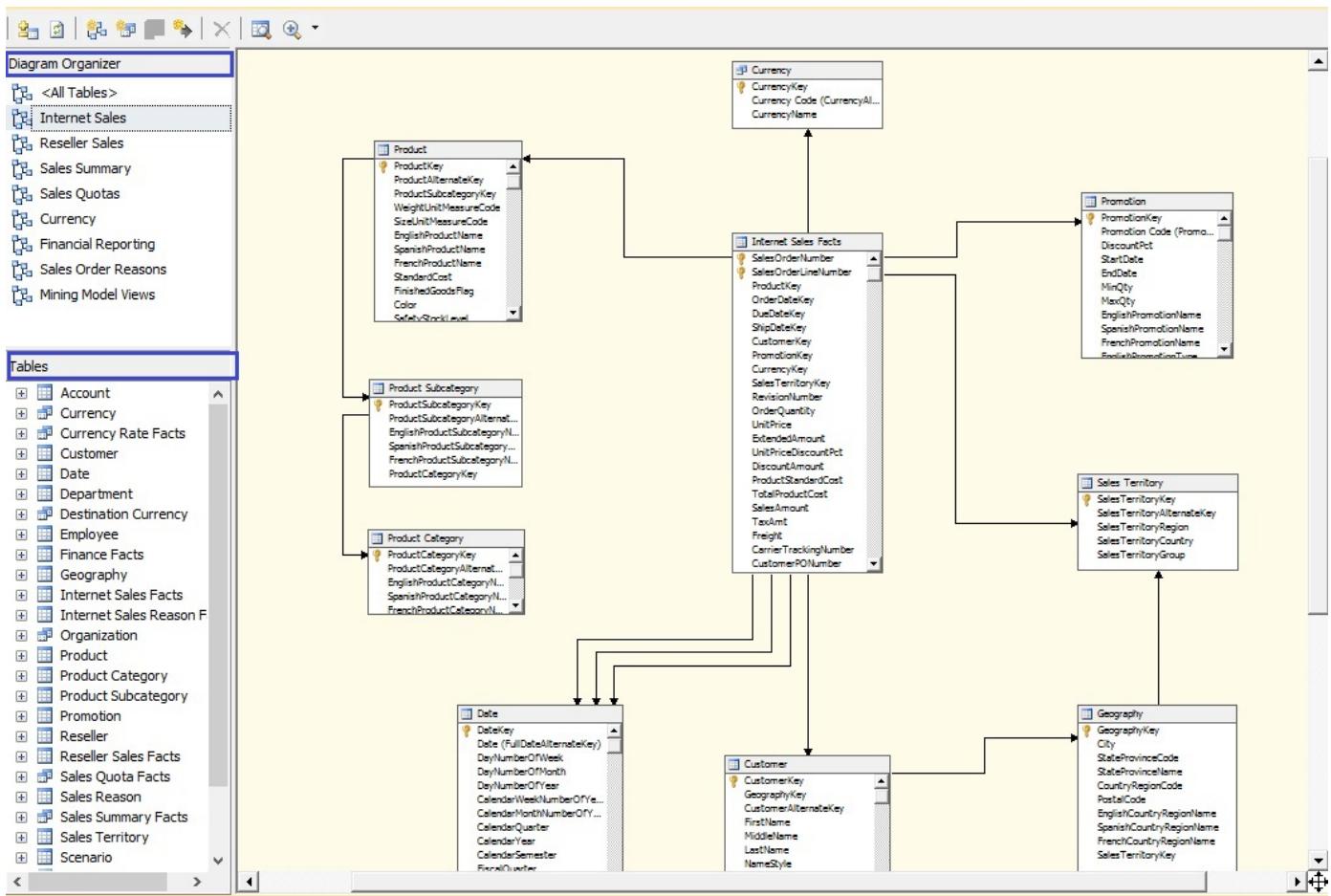
سپس در پنجره‌ی جدید، تنظیمات رشته‌ی ارتباطی با DW را مشاهده نمایید



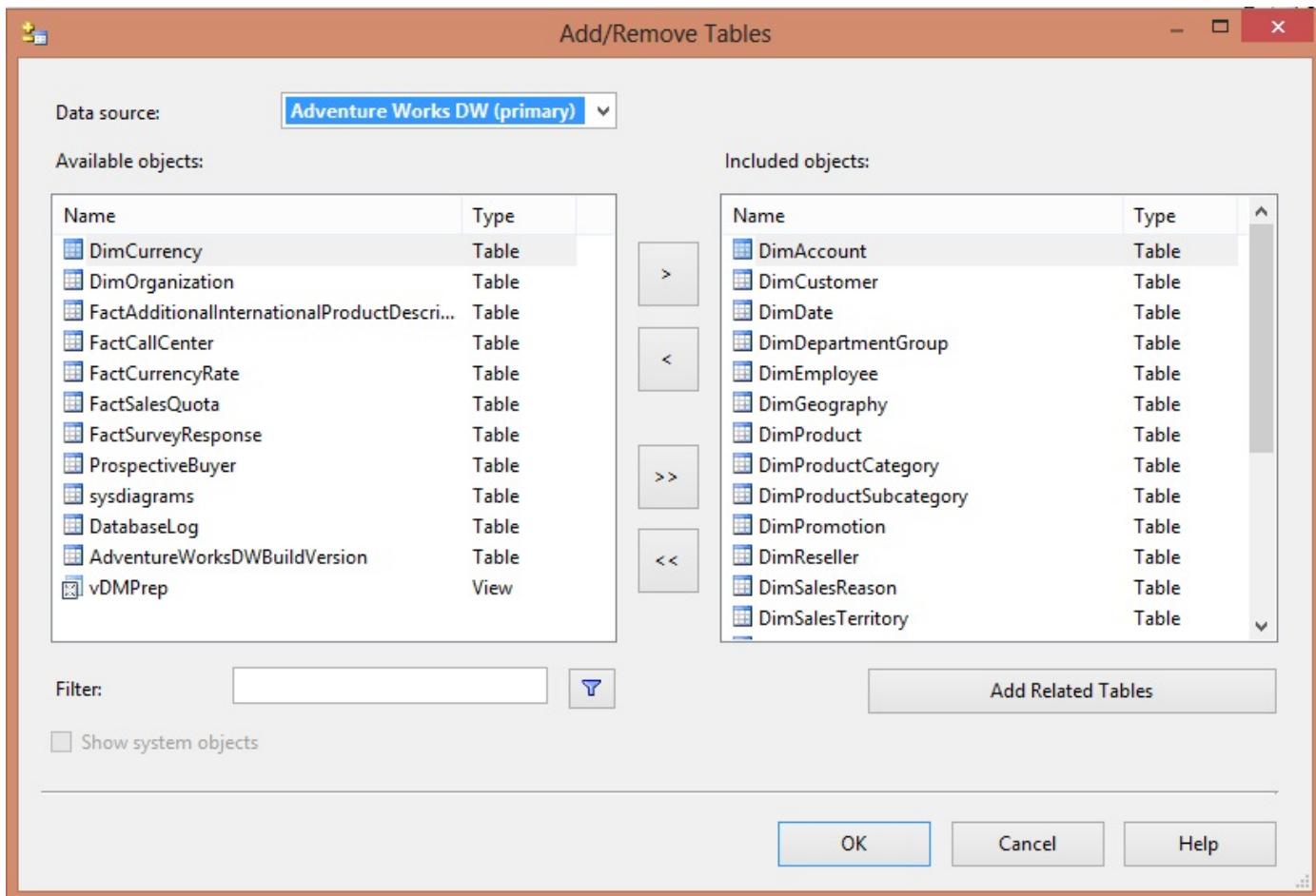
با زدن کلید Test Connection باید پیام Test Connection Succeeded را مشاهده نمایید. اکنون پنجره‌ها را با زدن کلید OK ببندید.

در قسمت Data Source View سه شی تعریف شده است؛ براساس دسته بندی مورد نظر و جاری در Business موجود در . Adventure Work

با کلیک راست کردن بر روی Adventure Works DW و انتخاب گزینه Open، اقدام به باز کردن DSV انتخاب شده کنید. در صفحه باز شده می‌توانید انواع دیاگرام تهیه شده را مشاهده نمایید و همچنین لیستی از جداول موجود در این DSV مشخص می‌باشد.



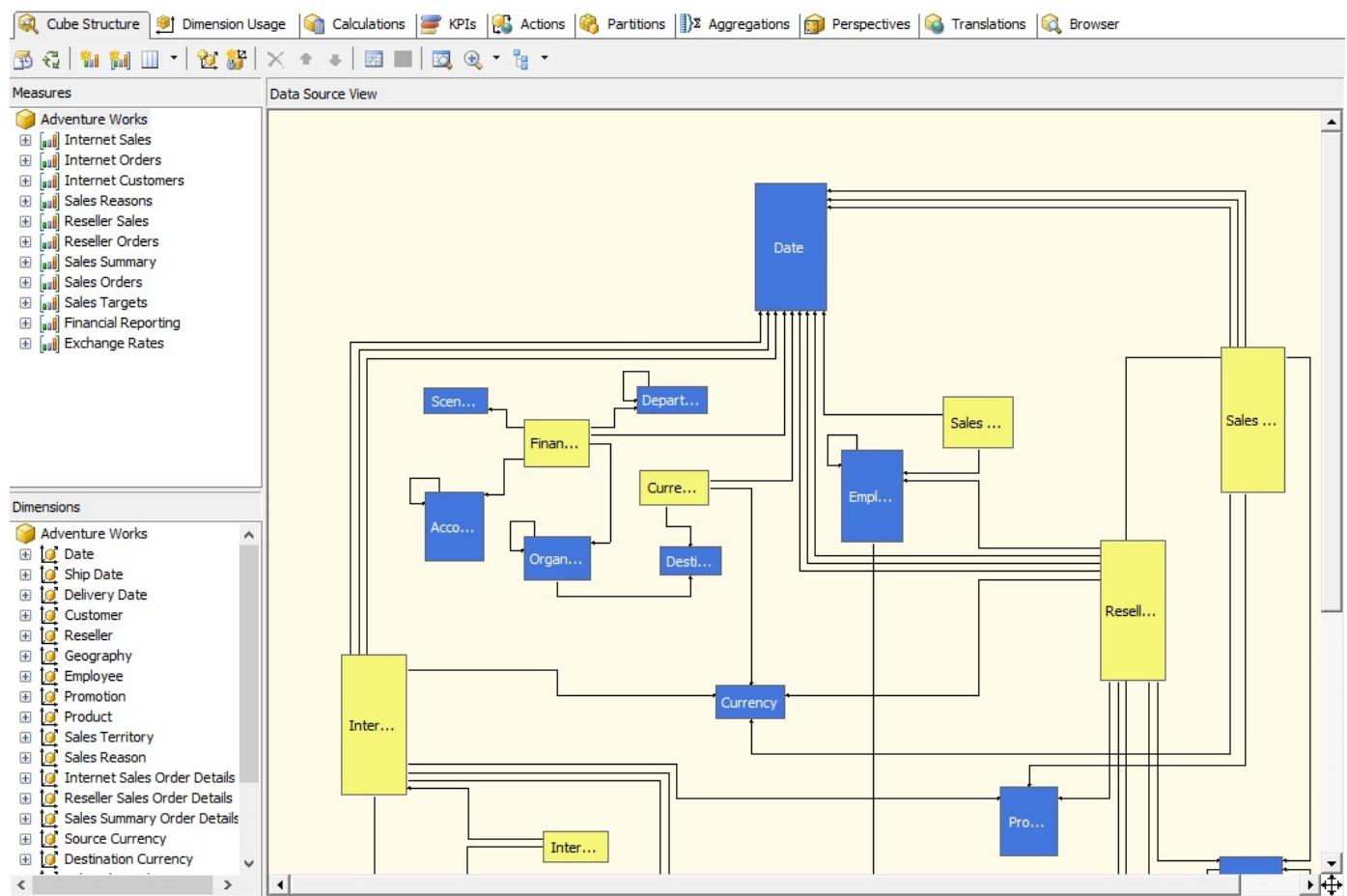
با کلیک راست در فضای خالی دیاگرام ، امکان Add/Remove کردن جداول را به دیاگرام دارد.



در شکل بالا بعد از انتخاب یک جدول در سمت راست و انتقال آن به سمت چپ می‌توانید با زدن دکمه Add Related Table براساس کلیدهای خارجی، جداول مرتبط با جدول انتخاب شده را به صورت خودکار انتخاب نمایید و به قسمت چپ انتقال دهید.

شما در ساخت Cube مشخص می‌نمایید که Cube را از کدام DSV خواهید ساخت. بنابراین انتخاب جداول DSV ها می‌بایست براساس نوع Business شما باشد تا در ساخت Cube به مشکلی برخورد نکنید.

در ساختار درختی موجود در پنجره Solution در شاخه Cube، می‌توانید Adventure Works را باز کنید (کلیک راست و انتخاب . (Open



در شکل بالا در سمت چپ، می‌توانید Measure های موجود در این Cube را مشاهده کنید. همچنین در قسمت بالا چندین Tab وجود دارند که در هر کدام تنظیمات بیشتری را بر روی Cube اعمال می‌کنیم. با توجه به اینکه طراحی Cube ها کاری تخصصی می‌باشد و نیاز به اطلاعات زیادی دارد اجازه دهد مقاله‌ای در خصوص طراحی Cube در SSAS جدایانه انتشار داده شود و فعلا در همین حد بسته کنیم. با این حال در صورت نیاز می‌توانید برای اطلاعات بیشتر در این خصوص کتاب Microsoft SQL Server Analysis Services 2008 With MDX از انتشارات Wrox را مطالعه نمایید.

در شاخه Dimensions در Solution Explorer می‌توانید تمامی بعدهایی که در تمامی Cube های شما استفاده شده‌اند را مشاهده نمایید.

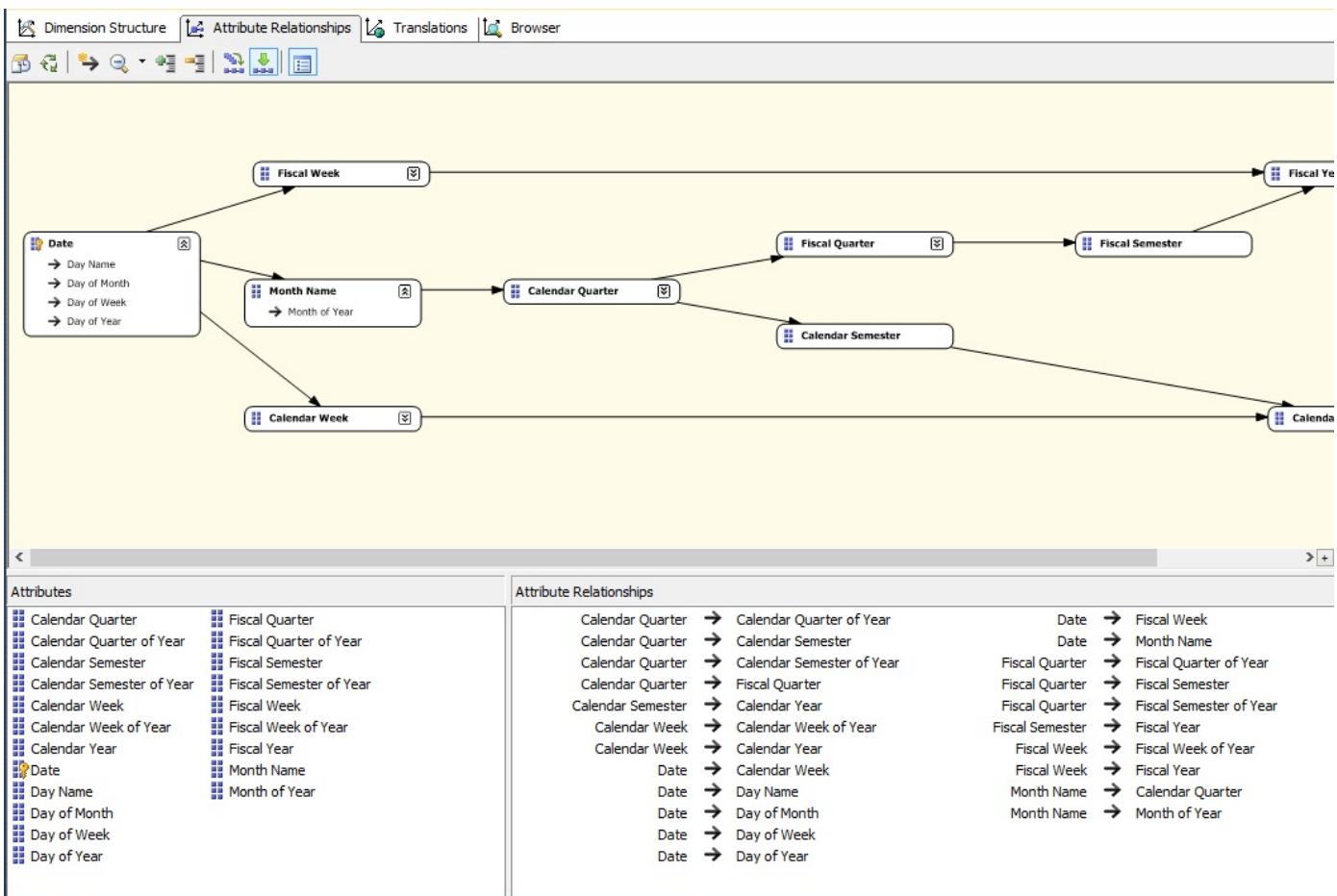
با انتخاب یک بعد (ترجیحاً بعد Date) و با کلیک راست کردن و انتخاب گزینه Open آن را باز نمایید.

The screenshot shows the BIMS Dimension Structure interface with three main tabs:

- Attributes** tab (selected): Displays a hierarchical list of attributes under the Date category, including Calendar Quarter, Calendar Quarter of Year, Calendar Semester, Calendar Semester of Year, Calendar Week, Calendar Week of Year, Calendar Year, Date, Day Name, Day of Month, Day of Week, Day of Year, Fiscal Quarter, Fiscal Quarter of Year, Fiscal Semester, Fiscal Semester of Year, Fiscal Week, Fiscal Week of Year, Fiscal Year, Month Name, and Month of Year.
- Hierarchies** tab: Displays two groups of hierarchies:
  - Fiscal** hierarchy: Includes Fiscal Year, Fiscal Semester, Fiscal Quarter, Month, and Date.
  - Calendar** hierarchy: Includes Calendar Year, Calendar Semester, Calendar Quarter, Month, and Date.
  - Calendar Weeks** hierarchy: Includes Calendar Year and Calendar Week.
  - Fiscal Weeks** hierarchy: Includes Fiscal Year and Fiscal Week.
 A note in the center says: "To create a new hierarchy, drag an attribute here."
- Data Source View** tab: Displays a list of columns from the Date table, including DateKey, Date (FullDateAlternateKey), DayNumberOfWeek, DayNumberOfMonth, DayNumberOfYear, CalendarWeekNumberOfYear (W...), CalendarMonthNumberOfYear (M...), CalendarQuarter, CalendarYear, CalendarSemester, FiscalQuarter, FiscalYear, FiscalSemester, CalendarQuarterDesc, FiscalQuarterDesc, CalendarSemesterDesc, FiscalSemesterDesc, EnglishDayNameOfWeek, SpanishDayNameOfWeek, FrenchDayNameOfWeek, EnglishMonthName, SpanishMonthName, FrenchMonthName, SimpleDate, MonthName, CalendarSemesterOfYear, FiscalSemesterOfYear, CalendarQuarterOfYear, FiscalQuarterOfYear, FiscalYearDesc, CalendarYearDesc, MonthNameValue, and FinancialPeriod.

در پنجره باز شده می‌توانید Tab 4 در بالا را مشاهد نمایید و در Tab نخست، Attribute ها و همچنین ساختار Hierarchies را مشاهد نمایید. در آخر Data source View را مشاهده نمایید.

در میان Attribute relationships می‌توانید ارتباط صفت‌های یک بعد را مشخص نمایید.

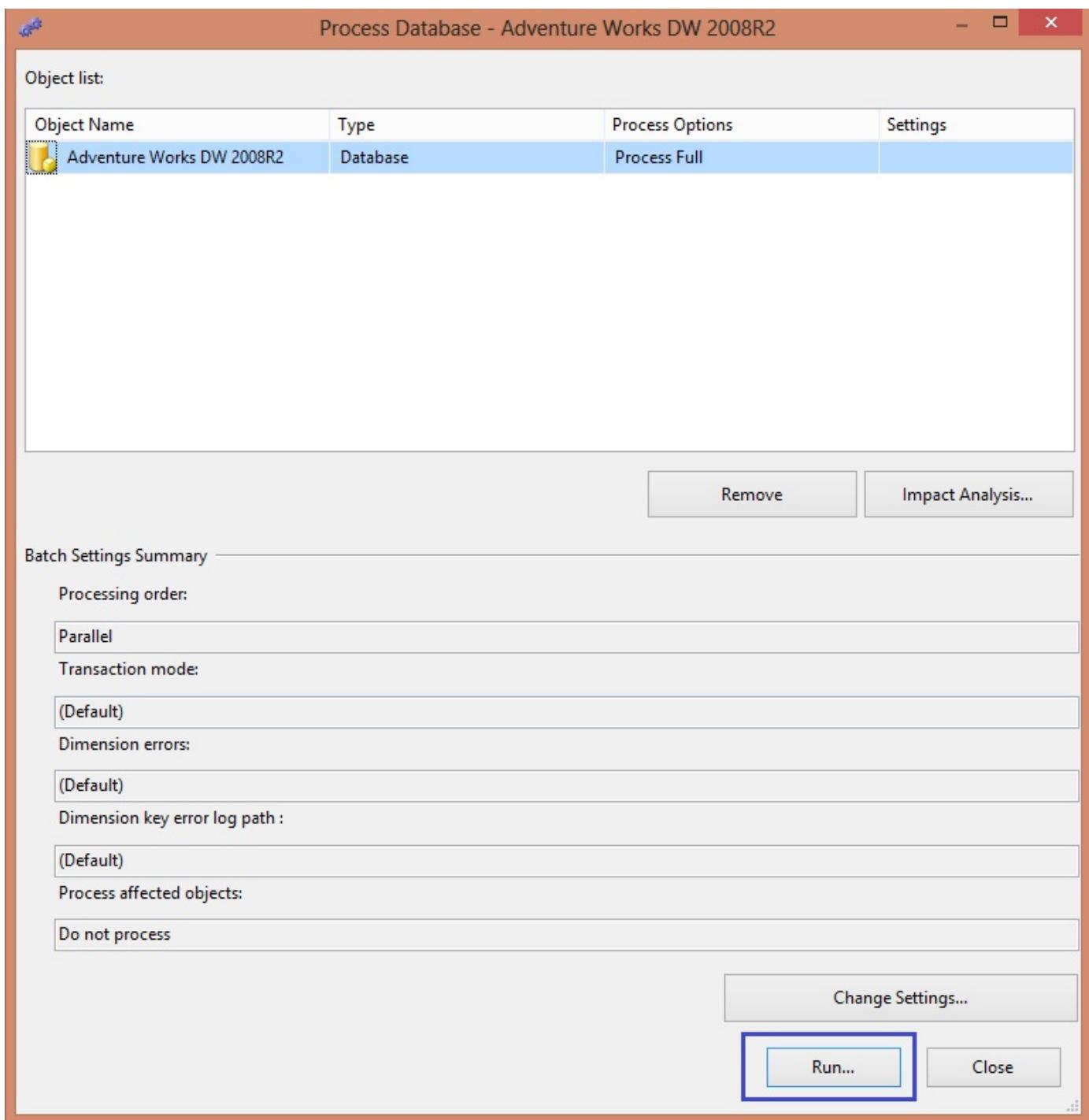


در Tab Browsing می‌توانید محتوای Dimension را بررسی نمایید (البته اگر در پروژه‌ی جدید قرار دارید حتماً می‌بایست پروژه را Deploy کرده باشید). در حالتیکه یک پایگاه داده چند بعدی را باز می‌کنید، نیازی به Deploy کردن نمی‌باشد؛ زیرا حتماً قبل این کار انجام شده است (زیرا شما پایگاه داده‌ی چند بعدی را بعد از Deploy کردن پروژه‌ی SSAS خواهید داشت))

The screenshot shows the BIMS Multidimensional Dimension Structure browser interface. At the top, there are tabs: Dimension Structure, Attribute Relationships, Translations, and Browser (which is highlighted with a blue box). Below the tabs are various icons and a language selection dropdown. The main area displays a hierarchical tree under 'Hierarchy: Calendar'. The 'Current level:' dropdown is set to '(All)'. The tree structure is as follows:

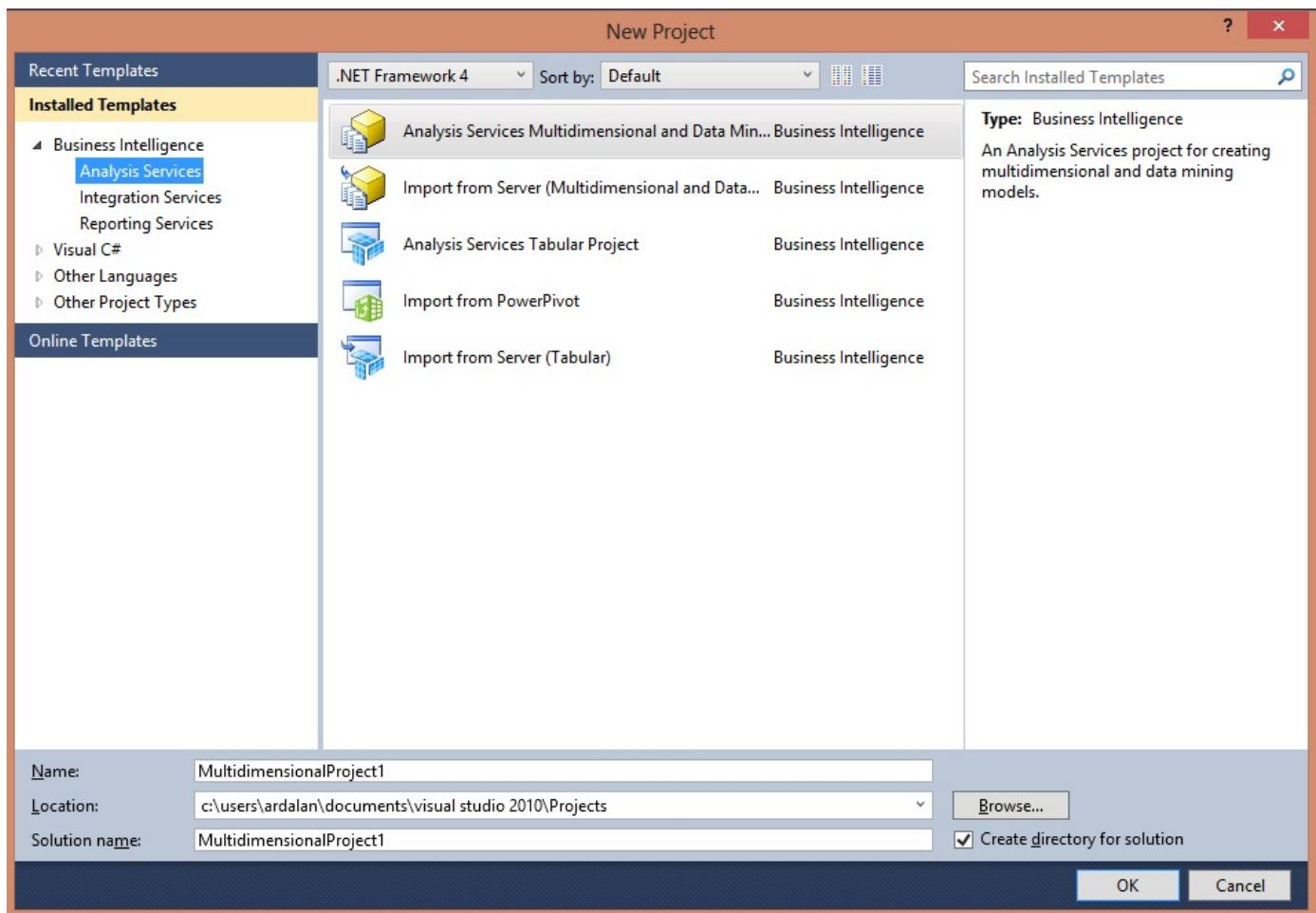
- All Periods
  - CY 2005
    - H2 CY 2005
      - Q3 CY 2005
        - July 2005
        - August 2005
        - September 2005
      - Q4 CY 2005
        - October 2005
        - November 2005
        - December 2005

در صورتیکه مانند روش بالا یک پایگاه داده‌ی چند بعدی را باز کنیم، دیگر نیازی به Deploy کردن نمی‌باشد و فقط برای اعمال تغییرات روی پایگاه داده‌ی چند بعدی باید پروژه را Process کنیم و برای این منظور روی نام پروژه کلیک راست کرده و گزینه‌ی Process را انتخاب کنید. با این کار تغییرات اعمال شده در BIMS روی پایگاه داده‌ی SSAS اعمال می‌گردد و داده‌ها با توجه به ساختار Cube‌ها دوباره پردازش می‌شوند.

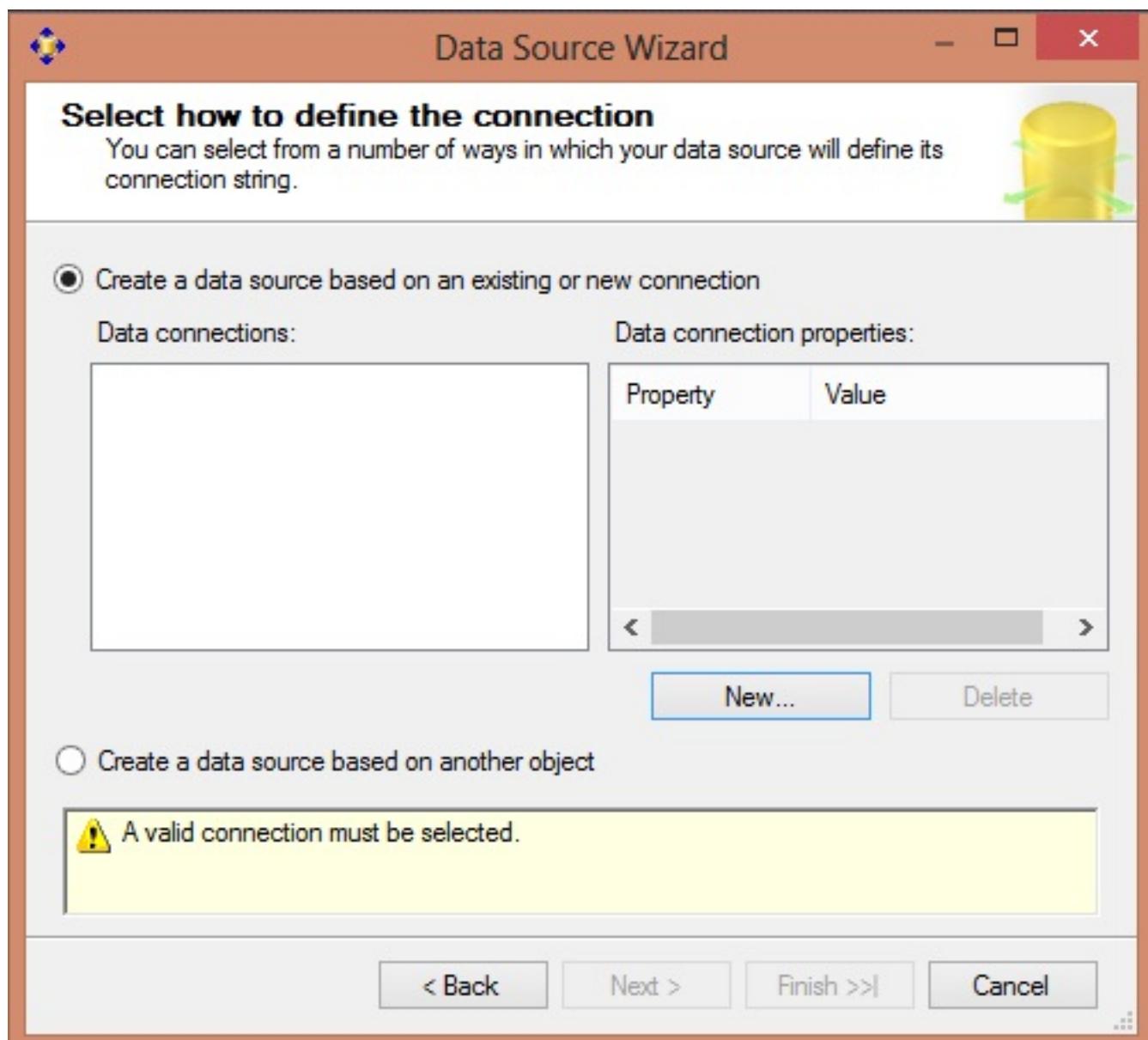


برای ساخت یک پروژه جدید به شکل زیر عمل می‌کنیم :

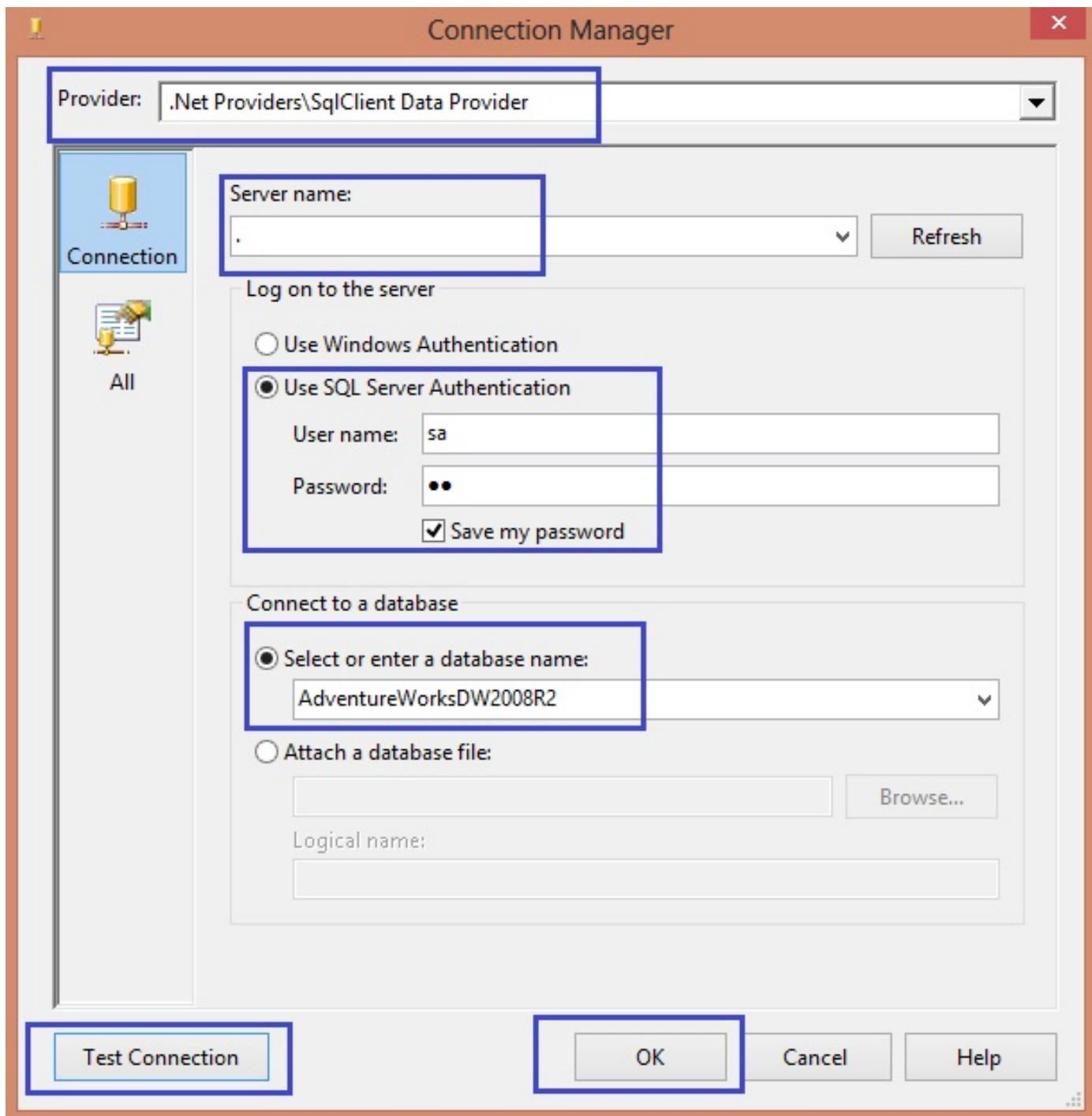
در ابتدا BIMS را باز کرده و سپس به منوی File رفته و در قسمت New گزینه‌ی Project را انتخاب می‌کنیم. سپس در صفحه‌ی باز شده، مطابق شکل زیر عمل کرده و یک پروژه از نوع Analysis Service Multidimensional ... می‌سازیم.



سپس بر روی شاخه **New Data Source** کلیک راست کرده و گزینه **Data Source** را می‌زنیم و پنجره‌های ویزارد را به جلو می‌رویم.



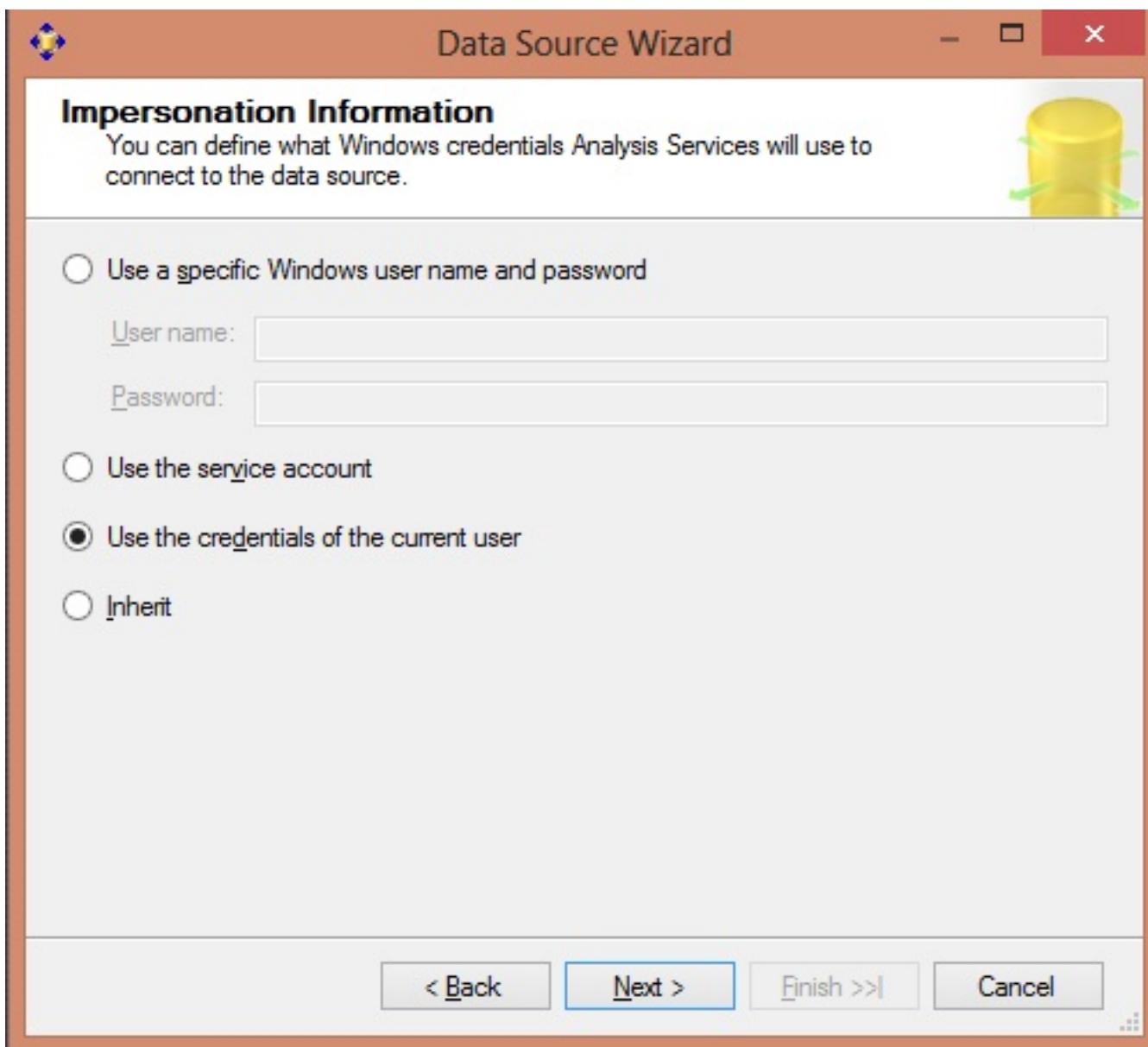
در ابتدا باید یک Connection به DW تولید کنیم. برای این منظور در پنجره‌ی فوق دکمه‌ی New را زده و اطلاعات را مطابق شکل زیر پر می‌کنیم.



و سپس OK را میزنیم.

در صورتی که SSAS در یک سرور دیگر نصب شده است در پنجره‌ی بعدی نیاز می‌باشد نام کاربری را که به سرویس SSAS در آن سرور دسترسی دارد را وارد کنیم.

در صورتی که SSAS روی سیستم Local نصب شده است و کاربری که با آن Login هستیم دسترسی کافی به SSAS را دارد، گزینه‌ی Use the credentials of the current user را انتخاب می‌کنیم.



در صفحه‌ی آخر یک نام برای DS انتخاب می‌کنیم.

سپس نیاز می‌باشد یک DSV بسازیم. برای این منظور روی شاخه‌ی Data Source View کلیک راست کرده و گزینه‌ی New را انتخاب کرده و سپس در پنجره‌ی Data Source Wizard باید ساخته شده در مرحله‌ی قبل را انتخاب کرده و سپس Next را بزنیم. در اینجا بر اساس بیزینس‌های مختلف، راه کارهای گوناگونی را داریم. به عبارت دیگر می‌توان جداول Fact و Dimension را مرتبط با آن را بر اساس زیر سیستم‌های مختلف انتخاب کرده و برای هر کدام از آنها یک DSV بسازیم. به نظر من می‌توانیم تمامی جداول را در این مرحله انتخاب کرده و سپس این تفکیک بندی را در سطح Cube ها انجام داد. به طور کلی دقت داشته باشید به هیچ عنوان DSV و Cube های سیستم را خیلی تفکیک نکنید. زیرا در نوشتن کوئری‌ها و Join بین Cube ها با مشکل و سختی روبرو خواهد شد. (از لحاظ تجربی تفکیک بندی به شرطی صورت گیرد که نیازی به Join کردن Cube ها در MDX Query ها نباشد).

Name	Type
DimEmployee	Table
DimGeography	Table
DimProductCategory	Table
DimProductSubcategory	Table
DimReseller	Table
DimSalesReason	Table
FactAdditionalInternationalProd...	Table
FactCallCenter	Table
FactCurrencyRate	Table
FactResellerSales	Table
FactSalesQuota	Table
FactSurveyResponse	Table
ProspectiveBuyer	Table
sysdiagrams	Table
DatabaseLog	Table
AdventureWorksDWBuildVersion	Table
vDMPrep	View
vTimeSeries	View
vTargetMail	View
vAssocSeqOrders	View
vAssocSeqLineItems	View

Name	Type
DimAccount	Table
DimDepartmentGroup	Table
FactFinance	Table
DimDate	Table
DimOrganization	Table
DimScenario	Table
FactInternetSales	Table
DimSalesTerritory	Table
DimCurrency	Table
DimProduct	Table
FactInternetSalesReason	Table
DimCustomer	Table
DimPromotion	Table

Filter:

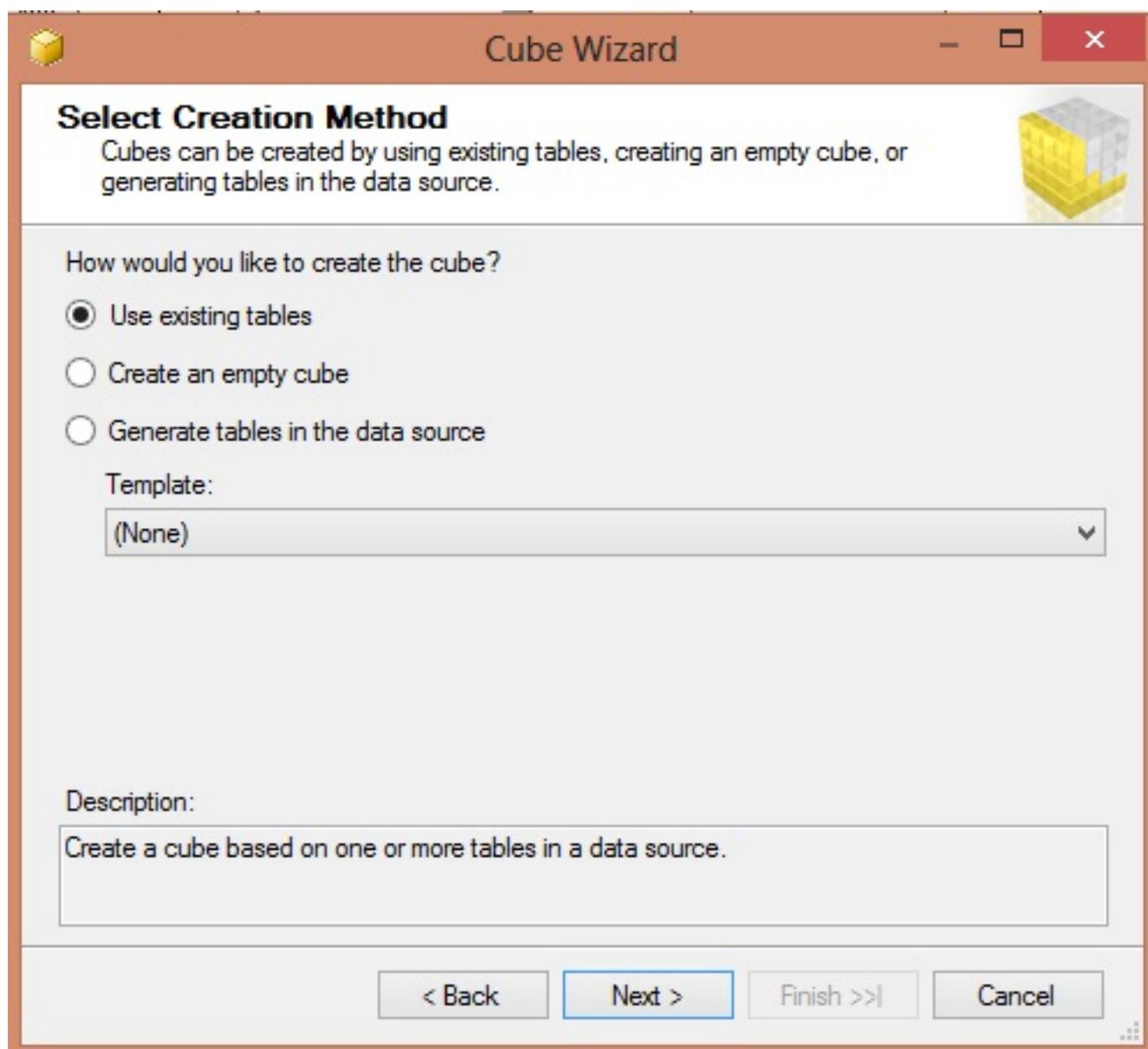
Show system objects

سپس یک نام برای DSV خود انتخاب کرده و Finish را بزنید.

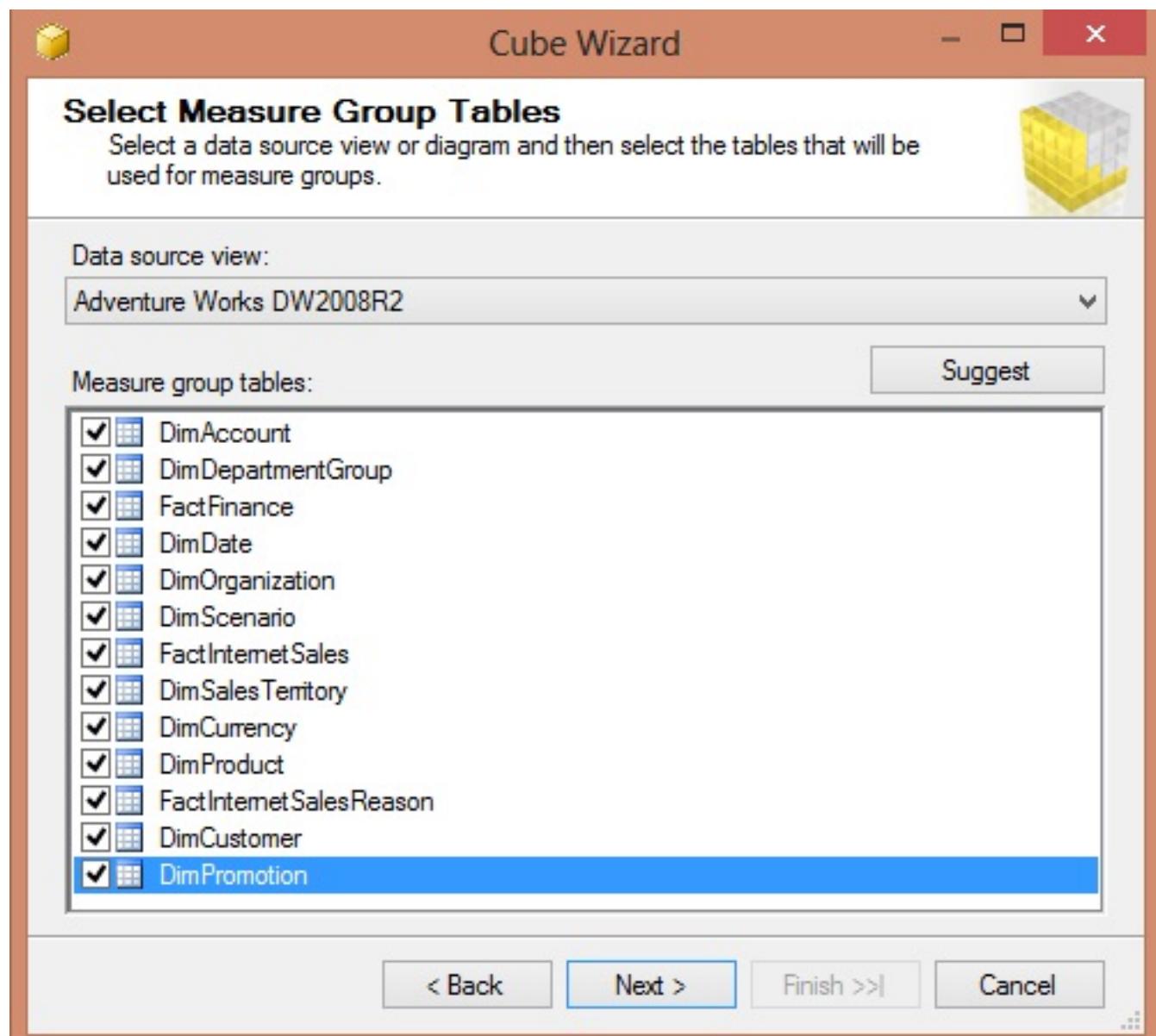
خوب؛ آخرین مرحله ساخت Cube می‌باشد (البته در طراحی Cube مطالب بسیاری وجود دارند که در یک مقاله‌ی دیگر تلاش خواهیم کرد تمامی آن موارد را توضیح دهم).

برای ساخت Cube، روی شاخه‌ی Cube کلیک راست کرده و گزینه‌ی New را بزنید.

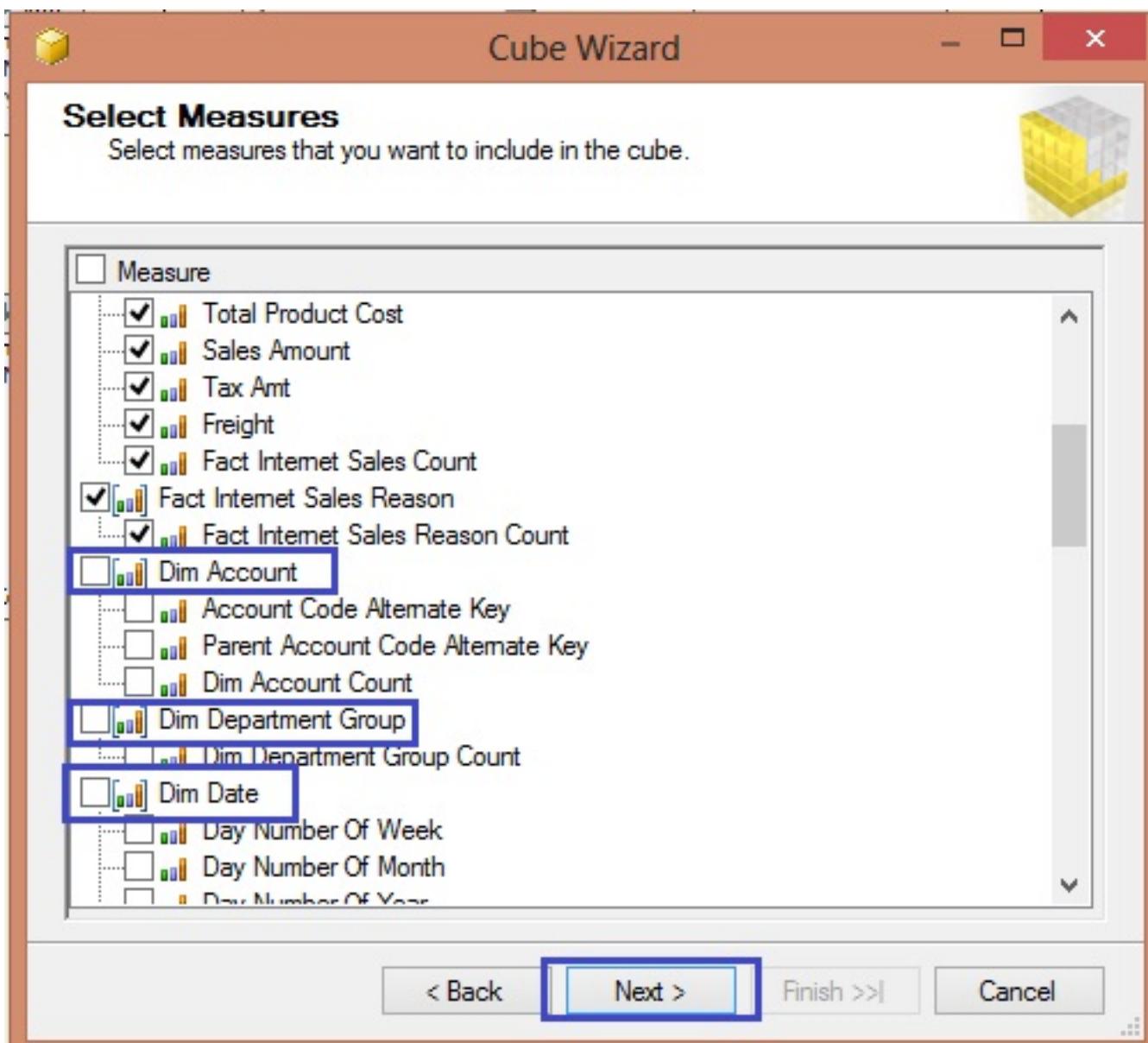
سپس Use Existing Table را انتخاب کرده و Next را بزنید.



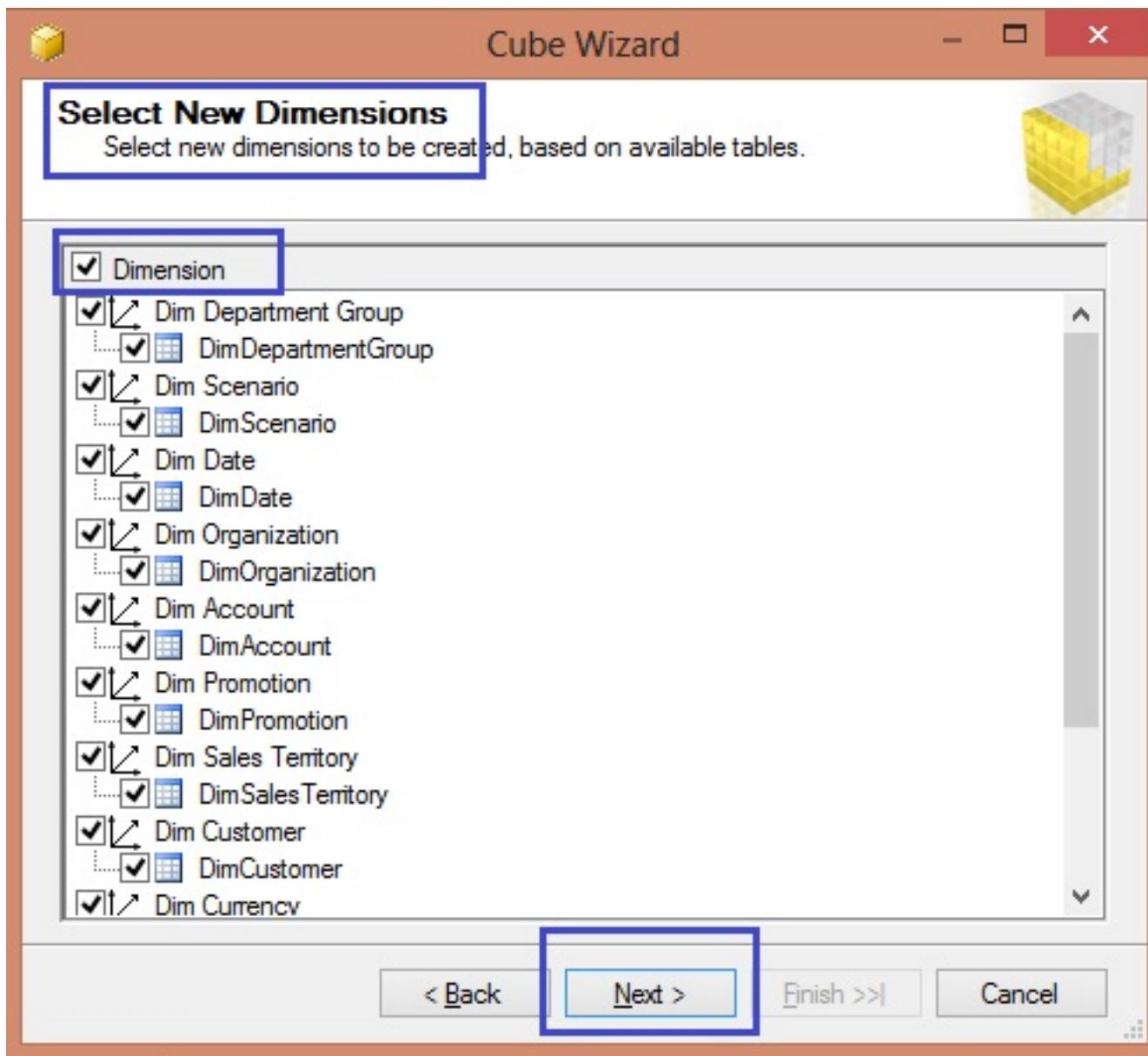
در پنجره‌ی بعدی باید DSV را انتخاب کرد و بعد جداول مورد نیاز در طراحی Cube را انتخاب کنید. فراموش نکنید در صورت انتخاب یک Dimension تمامی Fact های مرتبط با آن را انتخاب نماید. دکمه Next را بزنید.



در پنجره‌ی بعدی باید جداول Fact را انتخاب کرده و دکمه‌ی Next را بزنید.



سپس در پنجره‌ی بعدی دایمنشن را انتخاب نمایید. (ترجیحاً اجازه بدهید خود BIMS برای شما Dimension ها را بسازد، هرچند که خود شما می‌توانید بعداً به صورت دستی Dimension ها را ایجاد کنید).



بعد از زدن دکمه Next نامی برای Cube خود انتخاب نمایید و سپس دکمه Finish را بزنید.

بعد از ساخت Cube، چندین دایمنشن به صورت خودکار ساخته می‌شوند. البته گاهی نیاز می‌باشد که اقدام به ساخت ساختارهای سلسله مراتبی در Dimension ها کنیم (این مورد را در یک مقاله جداگانه آموزش خواهیم داد).

پروژه با کلیدهای ترکیبی Ctrl+Shift+B ساخته می‌شود و بعد از اطمینان از درست بودن ساخت پروژه، آن را باید Deploy کرد.

برای Deploy کردن یک پروژه کافی است بعد از تنظیم کردن رشته‌ی ارتباطی در DS (قبل از توضیح داده شده است) روی پروژه کلیک راست کرده و گزینه Deploy را بزنید.

## نظرات خوانندگان

نویسنده: باغبان  
تاریخ: ۱۳۹۲/۱۰/۰۱ ۲۱:۵۶

ممnon دوست عزیز  
میشه بفرمایید چه موقع باید از این روش استفاده کرد و مزایای اون چی هستش

نویسنده: اردلان شاه قلی  
تاریخ: ۱۳۹۲/۱۰/۰۱ ۲۳:۵۳

متاسفانه سوال شما برایم مشخص نمیباشد؟!  
{ چه موقع باید از این روش استفاده کرد }

با این وجود در این مقاله من دو روش برای کار با پایگاه داده‌ی چند بعدی را توضیح دادم. ابتدا زمانی که یک پایگاه داده وجود دارد و ما میخواهیم در ساختار Cube‌ها و ... در آن تغییر ایجاد کنیم (تغییرات باید درون محیط BIMS انجام شود) در چنین شرایطی با توجه به مواردی که توضیح داده شده پایگاه داده را در محیط BIMS باز کرده و تغییرات را انجام داده و سپس Process میکنیم. اما گاهی برای شروع کار نیاز میباشد طراحی MDB را از روی DW از ابتدا انجام دهیم. خوب در اینحالت چیزی برای باز کردن در BIMS وجود ندارد و باید خودمان از ابتدا آن را پیاده سازی کنیم. راحت‌تر بگم تقریباً مثل اینه که شما بخواهید یک برنامه با #C از ابتدا بنویسید یا اینکه یک برنامه‌ی موجود را باز کنید و تغییراتی در آن ایجاد کنید و سپس ذخیره کنید. امید وارم پاسخ سوال شما داده شده باشد.

## انجام عملیات Pivot توسط MDX

برای این منظور کافی است فقط جای سطر و ستون را با هم عوض کنیم. برای مثال، کوئری های زیر را اجرا نمایید.

```
Select
[Date].[Calendar].[Calendar Year] on columns,
[Product].[Product Categories].[Category] on rows
From [Adventure Works]
```

GO

```
Select
[Product].[Product Categories].[Category] on columns,
[Date].[Calendar].[Calendar Year] on rows
From [Adventure Works]
```

خروجی به صورت زیر می باشد.

The screenshot displays two result sets from SQL Server Management Studio. The top result set shows a pivoted table where categories (Accessories, Bikes, Clothing, Components) are listed as columns and years (CY 2005, CY 2006, CY 2007, CY 2008, CY 2010) are listed as rows. The bottom result set shows the original unpivoted table where years are listed as columns and categories are listed as rows.

	CY 2005	CY 2006	CY 2007	CY 2008	CY 2010
Accessories	\$20,235.36	\$92,735.35	\$296,532.88	\$161,794.33	(null)
Bikes	\$7,395,348.63	\$19,956,014.67	\$25,551,775.07	\$13,399,243.18	(null)
Clothing	\$34,376.34	\$485,587.15	\$871,864.19	\$386,013.16	(null)
Components	\$615,474.98	\$3,610,092.47	\$5,482,497.29	\$2,091,011.92	(null)

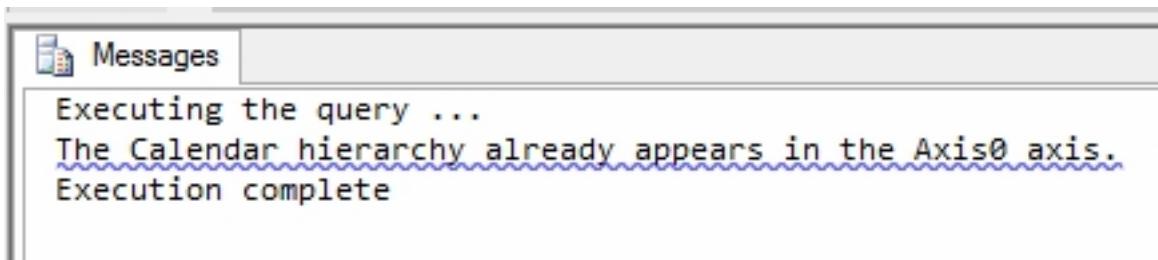
	Accessories	Bikes	Clothing	Components
CY 2005	\$20,235.36	\$7,395,348.63	\$34,376.34	\$615,474.98
CY 2006	\$92,735.35	\$19,956,014.67	\$485,587.15	\$3,610,092.47
CY 2007	\$296,532.88	\$25,551,775.07	\$871,864.19	\$5,482,497.29
CY 2008	\$161,794.33	\$13,399,243.18	\$386,013.16	\$2,091,011.92
CY 2010	(null)	(null)	(null)	(null)

چگونگی استفاده از ساختارهای سلسله مرتبی در محورهای مختلف :

برای روشن شدن مطلب چندین نمونه کوئری زیر را باهم اجرا می کنیم.

در ابتدا به خاطر داشته باشید که امکان استفاده از یک ساختار سلسله مراتبی در دو محور ( Axis ) مجزا وجود ندارد. برای روشن تر شدن این مطلب کوئری زیر را اجرا کنید:

```
Select  
[Date].[Calendar].[Calendar Year] on columns,  
[Date].[Calendar].[Month] on rows  
From [Adventure Works]
```



در توضیح مثال بالا دقت داشته باشید که واکنشی [Calendar Year] و [Month] از یک ساختار سلسله مراتبی یکسان به نام [Date].[Calendar] انجام شده است و این کار در MDX ها غیر مجاز می باشد.

البته استفاده از دو ساختار سلسله مراتبی متفاوت از یک دایمنشن در دو محور مجزا امکان پذیر می باشد برای مثال :

```
Select  
[Date].[Calendar].[Calendar Year] on columns,  
[Date].[Month of Year].[Month of Year] on rows  
From [Adventure Works]
```

در مثال بالا واکنشی از دو ساختار سلسله مراتبی مختلف اما از یک دایمنشن در دو محور مجزا صورت گرفته است.

	CY 2005	CY 2006	CY 2007	CY 2008	CY 2010
January	(null)	\$713,116.69	\$1,317,541.83	\$1,662,547.32	(null)
February	(null)	\$1,900,788.93	\$2,384,846.59	\$2,700,766.80	(null)
March	(null)	\$1,455,280.41	\$1,563,955.08	\$2,739,370.98	(null)
April	(null)	\$882,899.94	\$1,865,278.43	\$2,204,623.41	(null)
May	(null)	\$2,269,116.71	\$2,880,752.68	\$3,315,275.00	(null)
June	(null)	\$1,001,803.77	\$1,987,872.71	\$3,415,479.07	(null)
July	\$489,328.58	\$2,393,689.53	\$2,665,650.54	(null)	(null)
August	\$1,538,408.31	\$3,601,190.71	\$4,212,971.51	(null)	(null)
September	\$1,165,897.08	\$2,885,359.20	\$4,047,574.04	(null)	(null)
October	\$844,721.00	\$1,802,154.21	\$2,282,115.88	(null)	(null)
November	\$2,324,135.80	\$3,053,816.33	\$3,483,161.40	(null)	(null)
December	\$1,702,944.54	\$2,185,213.21	\$3,510,948.73	(null)	(null)

امکان استفاده از دو ساختار سلسله مراتبی یکسان در یک محور مجاز می باشد.

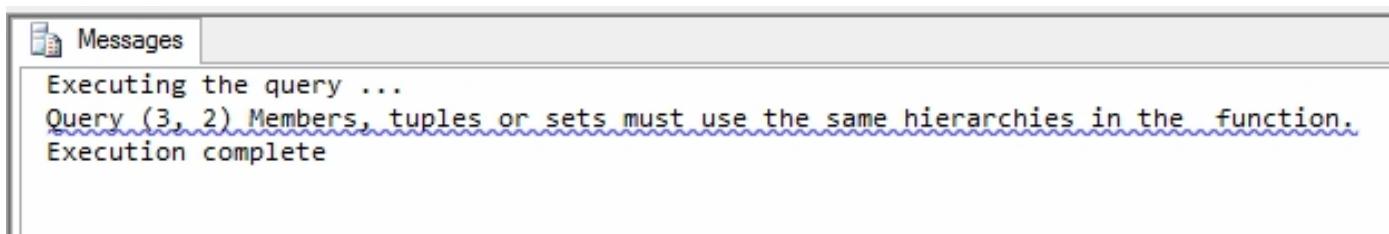
```
Select
[Measures].[Internet Sales Amount] on columns,
{[Date].[Calendar].[Calendar Year],[Date].[Calendar].[Month]} on rows
From [Adventure Works]
```

The screenshot shows the SSMS interface with the 'Results' tab selected. The table has two columns: 'Messages' (containing the query) and 'Results' (containing the output). The output table has a header 'Internet Sales Amount' and contains 15 rows of data:

	Internet Sales Amount
CY 2005	\$3,266,373.66
CY 2006	\$6,530,343.53
CY 2007	\$9,791,060.30
CY 2008	\$9,770,899.74
CY 2010	(null)
July 2005	\$473,388.16
August 2005	\$506,191.69
September 2005	\$473,943.03
October 2005	\$513,329.47
November 2005	\$543,993.41
December 2005	\$755,527.89
January 2006	\$596,746.56
February 2006	\$550,816.69

همچنین استفاده از دو ساختار سلسله مراتبی مختلف از یک دایمنشن در یک محور مجاز نمی باشد.

```
Select
[Measures].[Internet Sales Amount] on columns,
{[Date].[Calendar].[Calendar Year],[Date].[Month of Year]} on rows
From [Adventure Works]
```



در مثال بالا از دو ساختار سلسله مراتبی [Date].[Month of Year] و [Date].[Calendar] استفاده شده است.

در صورتی می توانیم از دو ساختار سلسله مراتبی متفاوت از یک دایمنشن در یک محور استفاده کنیم که آنها را با هم Cross Join کنیم.

```
Select
[Measures].[Internet Sales Amount] on columns,
crossjoin(
[Date].[Calendar].[Calendar Year],
[Date].[Month of Year].[Month of Year]
) on rows
From [Adventure Works]
```

		Internet Sales Amount
CY 2005	July	\$473,388.16
CY 2005	August	\$506,191.69
CY 2005	September	\$473,943.03
CY 2005	October	\$513,329.47
CY 2005	November	\$543,993.41
CY 2005	December	\$755,527.89
CY 2006	January	\$596,746.56
CY 2006	February	\$550,816.69
CY 2006	March	\$644,135.20
CY 2006	April	\$663,692.29
CY 2006	May	\$673,556.20
CY 2006	June	\$676,763.65

دقت داشته باشید که امکان استفاده از دو ساختار سلسله مراتبی یکسان از یک دایمنشن در Cross Join وجود ندارد همان گونه که در دو مثال قبل مشاهده نمودید. (و البته نیازی هم به استفاده از Join در Cross Join وجود ندارد)

```
Select
[Measures].[Internet Sales Amount] on columns,
crossjoin(
[Date].[Calendar].[Calendar Year],
[Date].[Calendar].[Month]
)on rows
From [Adventure Works]
```

به طور کلی استفاده از Cross Join مانند استفاده از () می باشد . بنابر این می توان کلمهی Cross Join را ننوشت .

```
Select
{[Measures].[Internet Sales Amount],[Measures].[Reseller Sales Amount]} on columns,
crossjoin(
[Date].[Calendar].[Calendar Year],
[Date].[Month of Year].[Month of Year]
)on rows
From [Adventure Works]
GO
Select
{[Measures].[Internet Sales Amount],[Measures].[Reseller Sales Amount]} on columns,
(
[Date].[Calendar].[Calendar Year],
[Date].[Month of Year].[Month of Year]
)on rows
From [Adventure Works]
```

دو کوئری بالا مشابه هم می باشند.

می توان از دو Cross Join در دو محور مختلف نیز استفاده کرد.

```
Select
crossjoin(
[Product].[Product Categories].[Category],
{[Measures].[Internet Sales Amount],[Measures].[Reseller Sales Amount]}
)on columns,
crossjoin(
[Date].[Calendar].[Calendar Year],
[Date].[Month of Year].[Month of Year]
)on rows
From [Adventure Works]
```

The screenshot shows a query results grid with two tabs: 'Messages' and 'Results'. The 'Results' tab displays a table with data from the Adventure Works cube. The table has 7 columns: two empty header columns, followed by 'Accessories' (with sub-columns 'Internet Sales Amount' and 'Reseller Sales Amount'), then 'Bikes' (with sub-columns 'Internet Sales Amount' and 'Reseller Sales Amount'), and finally 'Clothing' (with sub-columns 'Internet Sales Amount' and 'Reseller Sales Amount'). The data rows show sales figures for each month from July to December of the year 2005, categorized by product type (Accessories, Bikes, Clothing) and sales channel (Internet Sales Amount and Reseller Sales Amount).

		Accessories	Accessories	Bikes	Bikes	Clothing	Clothing
		Internet Sales Amount	Reseller Sales Amount	Internet Sales Amount	Reseller Sales Amount	Internet Sales Amount	Reseller Sales Amount
CY 2005	July	(null)	\$1,695.67	\$473,388.16	\$453,231.80	(null)	\$2,875.1
CY 2005	August	(null)	\$3,593.20	\$506,191.69	\$1,413,253.52	(null)	\$7,038.5
CY 2005	September	(null)	\$3,250.03	\$473,943.03	\$1,054,995.97	(null)	\$5,138.3
CY 2005	October	(null)	\$1,937.90	\$513,329.47	\$777,394.97	(null)	\$3,583.7
CY 2005	November	(null)	\$5,490.73	\$543,993.41	\$2,152,858.49	(null)	\$8,551.7
CY 2005	December	(null)	\$4,267.84	\$755,527.89	\$1,543,613.88	(null)	\$7,188.8

دقت داشته باشید نوع محصول در ستون ها بالای سر ستون [Reseller Sales Amount] و [Internet Sales Amount] قرار گرفته است.

استفاده از Cross Join های تودر تو نیز وجود دارد.

```
Select
crossjoin(
[Sales Territory].[Sales Territory].[Country],
crossjoin(
[Product].[Product Categories].[Category],
{
[Measures].[Internet Order Count],
[Measures].[Reseller Order Count]
}
)
)on columns ,
crossjoin(
[Date].[Calendar].[Calendar Year],
[Date].[Month of Year].[Month of Year]
) on rows
From [Adventure Works]
```

		France	France	France	France	France
		Accessories	Accessories	Bikes	Bikes	Clothing
		Internet Order Count	Reseller Order Count	Internet Order Count	Reseller Order Count	Internet Order Count
CY 2005	July	(null)	(null)	13	(null)	(null)
CY 2005	August	(null)	(null)	6	(null)	(null)
CY 2005	September	(null)	(null)	11	(null)	(null)
CY 2005	October	(null)	(null)	4	(null)	(null)
CY 2005	November	(null)	(null)	12	(null)	(null)
CY 2005	December	(null)	(null)	13	(null)	(null)
CY 2006	January	(null)	(null)	11	(null)	(null)
CY 2006	February	(null)	(null)	11	(null)	(null)

در کوئری زیر به جای () استفاده شده است

```
Select
crossjoin(
[Sales Territory].[Sales Territory].[Country],
[Product].[Product Categories].[Category],
{
[Measures].[Internet Order Count],
[Measures].[Reseller Order Count]
}
) on columns,
([Date].[Calendar].[Calendar Year],[Date].[Month of Year].[Month of Year]) on rows
From [Adventure Works]
```

استفاده از عملگر \* مانند استفاده از Cross Join می باشد.

```
Select
crossjoin(
[Sales Territory].[Sales Territory].[Country],
[Product].[Product Categories].[Category],
{
[Measures].[Internet Order Count],
[Measures].[Reseller Order Count]
}
) on columns,
[Date].[Calendar].[Calendar Year] * [Date].[Month of Year].[Month of Year] on rows
From [Adventure Works]
```

در مقالات بعدی آموزش MDX Query را ادامه خواهیم داد.

امروز اولین دستورات MDX را خواهیم نوشت . قبل از شروع کار فراموش نکنید موارد زیر را حتما انجام داده باشید :

نصب پایگاه داده Adventure Work DW 2008 و همچنین نصب پایگاه داده چند بعدی 2008 روی SSAS مطالعه قسمت‌های قبلی برای آشنایی با مفاهیم پایه .

در صورتیکه پیش شرایط فوق را نداشته باشید، احتمالا در ادامه با مشکلاتی مواجه خواهید شد؛ زیرا برای آموزش MDX Query ها از پایگاه داده Adventure Work DW 2008 استفاده شده است.

دققت داشته باشید که MDX Query ها تا حدودی شبیه T/SQL می‌باشند؛ اما مطلقاً از نظر مفهومی با هم شباهت ندارند. به عبارت دیگر ما در T/SQL با یک مدل رابطه‌ای سرو کار داریم در حالیکه در MDX ها با یک پایگاه داده چند بعدی کار می‌کنیم. به بیان دیگر در پایگاه داده‌های رابطه‌ای صحبت از جداول، ردیف‌ها، ستون‌ها و ضرب دکارتی مجموعه‌ها می‌باشد، اما در پایگاه داده‌های چند بعدی در خصوص Dimension, Fact, Cube, Tuple ... صحبت می‌کنیم. البته ماکروسافت تلاش کرده است تا حد زیادی Syntax ها شبیه به یکدیگر باشند.

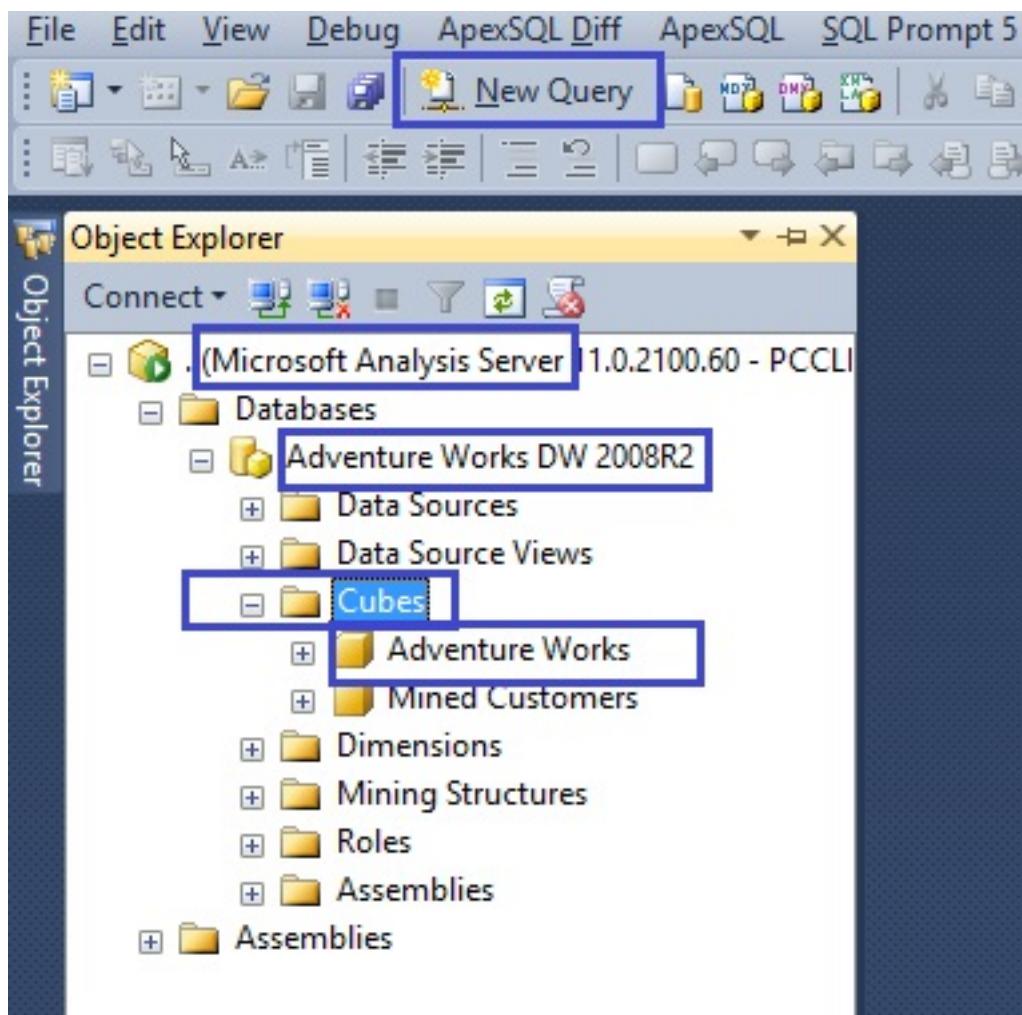
نحوه‌ی نوشتن یک Select در MDX ها به صورت زیر می‌باشد :

```
Select
{} On Columns ,
{} On Rows
From <Cube_Name>
Where <Condition>
```

در ادامه با اجرای هر کوئری، توضیحات لازم در خصوص آن ارایه می‌گردد و با پیگیری این آموزش‌ها می‌توانید مفاهیم، توابع و ... را در MDX Query ها بیاموزید.

برای اجرای دستورات زیر باید Microsoft SQL Server Management Studio را باز نمایید و به سرویس SSAS متصل شوید. سپس پایگاه داده‌ی Adventure Works DW 2008R2 را انتخاب نمایید و از Cubes Adventure Works را انتخاب نمایید.

حال دکمه‌ی New Query را در بالای صفحه بزنید ( Ctrl + N )



سپس در صفحه‌ی باز شده می‌توانید Cube های آن Cube را انتخاب کرده و کمی پایین‌تر Measure Group را خواهیم داشت و در انتها Dimension ها و Measure ها قرار گرفته‌اند. (در هنگام نوشتن Select می‌توان از عمل Drag&Drop برای آسان‌تر شدن نوشتن MDX Query ها نیز استفاده کنید)

متاسفانه هنوز در IDE SQL Server کلیدی برای مرتب سازی دستورات MDX وجود ندارد و البته در نرم افزارهای مانند SQL Server 2008 هم چنین چیزی قرار داده نشده است. بنابر این توصیه می‌شود در نوشتن دستورات MDX تمام تلاش خود را بکنید تا دستوراتی مرتب و خوانا را تولید کنید.

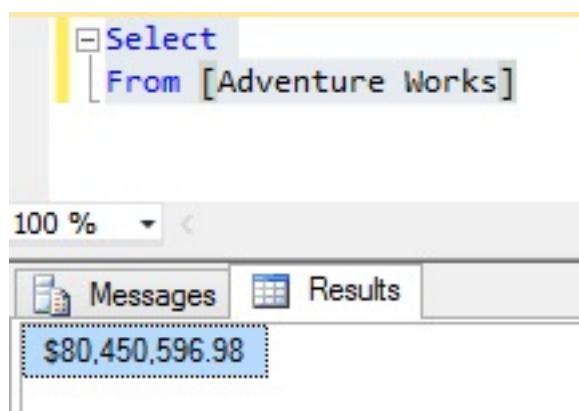
با اجرای دستور زیر اولین کوئری خود را در پایگاه داده‌ی چند بعدی بنویسید (برای اجرا کلید F5 مانند T/SQL کار خواهد کرد).

```
Select
From [Adventure Works]
```

شاید تعجب کنید. کوئری فاقد قسمت Projection می‌باشد! در MDX ها می‌توان هیچ سطر یا ستونی را انتخاب نکرد. اما چگونه؟ و خروجی نمایش داده شده چیست؟

برای توضیح مطلب فوق باید در خصوص Default Measure کمی اطلاعات داشته باشید. در هنگام Deploy کردن پروژه در SSAS برای هر Cube یک Measure به عنوان Default Measure پیش فرض انتخاب شده. بنابر این در صورتیکه هیچ گونه Where Projection یا اعمال نشده باشد، SQL Server به صورت پیش فرض مقدار Masaure پیش فرض را بدون اعمال هیچ بعدی نمایش می‌دهد.

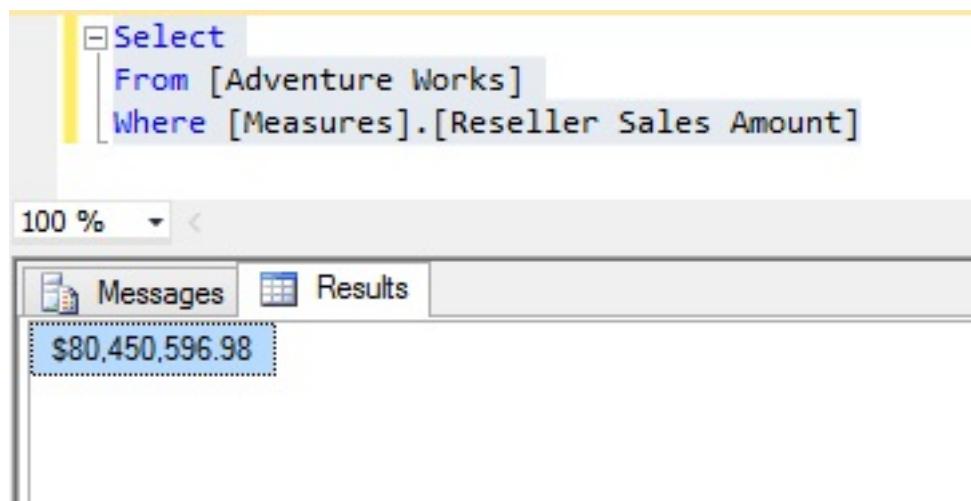
خروجی دستور بالا مشابه تصویر زیر می‌باشد.



حال دستور زیر را اجرا می‌کنیم :

```
Select
From [Adventure Works]
Where [Measures].[Reseller Sales Amount]
```

تصویر خروجی به صورت زیر می‌باشد :



شاید باز هم تعجب کنید. نوشتن نام یک شاخص به جای عبارت شرط؟! آیا خروجی عبارات شرطی نباید Boolean باشند؟

خیر. اگر چنین پرسش هایی در ذهن شما ایجاد شده باشد، به دلیل مقایسه‌ی MDX با T/SQL می‌باشد. در اینجا شرط Where دریف‌های جدول مدل رابطه‌ای اعمال نمی‌شود و عملاً بیانگر واکنش اطلاعات از مدل چند بعدی می‌باشد. با اعمال شرط فوق به SSAS اعلام کرده ایم که خروجی بر اساس شاخص [Measures].[Reseller Sales Amount] باشد. با توجه به این که شاخص انتخاب شده با شاخص پیش فرض یکی می‌باشد خروجی با حالت قبل تفاوتی نخواهد کرد.

برای درک بهتر، کوئری زیر را اجرا کنید :

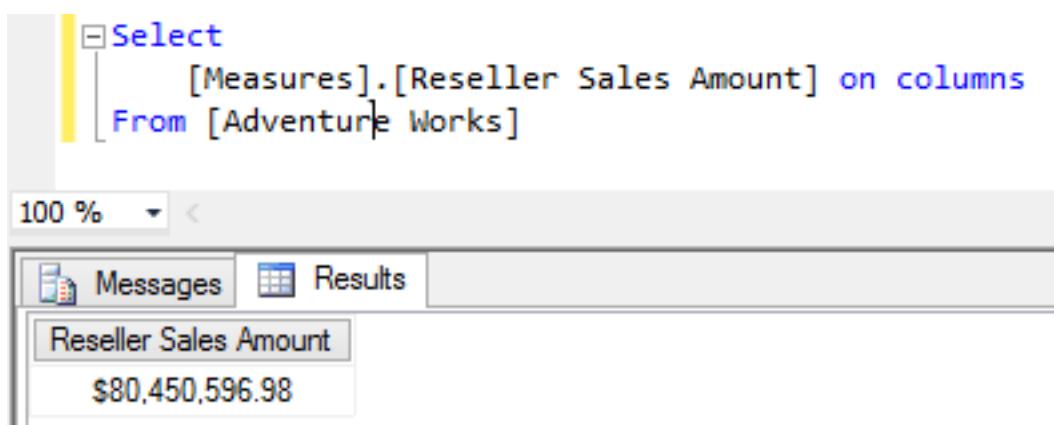
```
Select
From [Adventure Works]
where [Measures].[Internet Sales Amount]
```

استفاده از این شرط سبب استفاده نشدن از شاخص پیش فرض می شود . به عبارت دیگر این کوئری دارای سرجمع مبلغ فروش اینترنتی می باشد.

دستور زیر را اجرا کنید :

```
Select
[Measures].[Reseller Sales Amount] on columns
From [Adventure Works]
```

با اعمال یک شاخص خاص در ستون ، عملا فیلترینگ انجام می شود

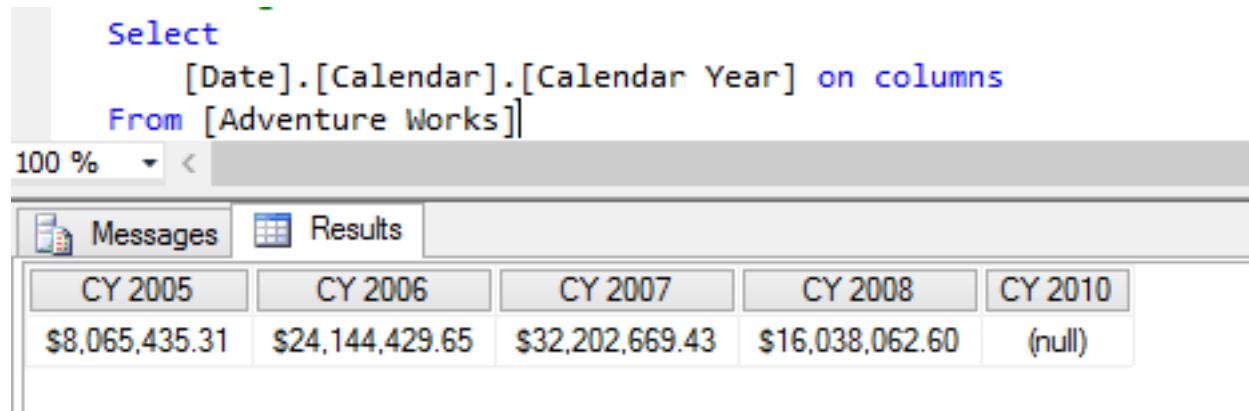


استفاده از یک دایمنشن در ستون :

دستور زیر را اجرا کنید

```
Select
[Date].[Calendar].[Calendar Year] on columns
From [Adventure Works]
```

خروجی به شکل زیر خواهد بود



```

Select
    [Date].[Calendar].[Calendar Year] on columns
From [Adventure Works]
  
```

The screenshot shows the SSMS interface with the 'Results' tab selected. The query results are displayed in a grid:

	CY 2005	CY 2006	CY 2007	CY 2008	CY 2010
	\$8,065,435.31	\$24,144,429.65	\$32,202,669.43	\$16,038,062.60	(null)

همان طور که مشاهده می‌کنید خروجی دارای چندین ستون می‌باشد و دارای مقادیری در هر ستون. اما این مقادیر از کجا آمده‌اند؟

همواره این نکته را به خاطر بسپارید که در صورت عدم ذکر نام یک Measure در کوئری، SSAS پیش فرض استفاده می‌کند. حال کوئری فوق میزان فروش نمایندگان (Reseller Sales Amount) را در هر سال نمایش می‌دهد.

سوال بعدی این می‌باشد که این سال‌ها از کجا آمده‌اند؟ خوب برای درک بهتر این مورد می‌توانیم مانند تصویر زیر به دایمنشن رفته و در ساختار سلسله مراتبی، اعضای سطح [Date].[Calendar].[Calendar Year] را مشاهده کنیم.

Cube:

Adventure Works

Metadata Functions

Measure Group:

<All>

```

Adventure Works
  + Measures
  + KPIs
  + Account
  + Customer
  - Date
    - Calendar
      + Date.Calendar Quarter of Year
      + Date.Calendar Semester of Year
      + Date.Calendar Week of Year
      + Date.Calendar Year
      - Date.Calendar
        - Members
          + Calendar Year
            + Member Properties
              CY 2005
              CY 2006
              CY 2007
              CY 2008
              CY 2010
        - Calendar Semester
  
```

: ایجاد سرجمع ستون‌ها :

کوئری زیر را اجرا نمایید

```

Select
{[Date].[Calendar].[Calendar Year],[Date].[Calendar]} on columns
From [Adventure Works]
  
```

بعد از اجرا تصویر زیر را خواهید دید :

CY 2005	CY 2006	CY 2007	CY 2008	CY 2010	All Periods
\$8,065,435.31	\$24,144,429.65	\$32,202,669.43	\$16,038,062.60	(null)	\$80,450,596.98

سوال اول این می‌باشد که کاربرد `{}` در انتخاب دایمنشن‌ها چیست؟ در پاسخ می‌توان گفت که اگر شاخص‌ها یا بعد‌ها، مرتبط به یک سلسله مرتب باشد آنها را در یک `{}` قرار می‌دهیم ولی اگر سلسله مرتب متفاوت باشد، یا بعد و شاخص باشند باید در `()` قرار بگیرند.

خوب همان طور که مشخص است در ساختار سلسله مرتبی ابتداء سال و بعد یک سطح بالاتر را انتخاب کرده ایم این به معنی نمایش سرجمع در سطح بالاتر از سال می‌باشد (سرجمع تمامی سال‌ها).

#### استفاده از دایمنشن و Measure در سطر و ستون مجرما:

کوئری زیر را اجرا نمایید

```
Select
{[Date].[Calendar].[Calendar Year],[Date].[Calendar]} on columns,
[Product].[Product Categories].[Category] on rows
From [Adventure Works]
```

خروجی مشابه شکل زیر می‌باشد

	CY 2005	CY 2006	CY 2007	CY 2008	CY 2010	All Periods
Accessories	\$20,235.36	\$92,735.35	\$296,532.88	\$161,794.33	(null)	\$571,297.93
Bikes	\$7,395,348.63	\$19,956,014.67	\$25,551,775.07	\$13,399,243.18	(null)	\$66,302,381.56
Clothing	\$34,376.34	\$485,587.15	\$871,864.19	\$386,013.16	(null)	\$1,777,840.84
Components	\$615,474.98	\$3,610,092.47	\$5,482,497.29	\$2,091,011.92	(null)	\$11,799,076.66

در مثال فوق از بعدها در ستون و همزنان، نمایش نوع دسته بندی محصولات در ردیف‌ها استفاده شده است. به عبارت دیگر نتیجه عبارت است از فروش نماینگان فروش ( Reseller Sales Amount ) براساس هر سال به تفکیک نوع دسته بندی محصول

فروخته شده.

(کسانی که چنین گزارشی را با استفاده از T/SQL نوشته اند، احتمالاً از آسانی نوشتتن این گزارش توسط MDX ها شگفت زده شده اند).

**قراردادن فیلد سرجمع در ردیف :**

برای این منظور کوئیری زیر را اجرا نمایید

```
Select {[Date].[Calendar].[Calendar Year],[Date].[Calendar]} on columns,
 {[Product].[Product Categories].[Category],[Product].[Product Categories]}on rows
 From [Adventure Works]
```

خروجی به صورت زیر میباشد

The screenshot shows a query window with the following content:

```
Select {[Date].[Calendar].[Calendar Year],[Date].[Calendar]} on columns,
 {[Product].[Product Categories].[Category],[Product].[Product Categories]}on rows
 From [Adventure Works]
```

Below the query, there is a results grid showing sales data by product category over several years. The columns represent years from CY 2005 to CY 2010, plus an 'All Periods' column. The rows represent product categories: Accessories, Bikes, Clothing, Components, and All Products. The data shows varying sales figures for each category across the years.

	CY 2005	CY 2006	CY 2007	CY 2008	CY 2010	All Periods
Accessories	\$20,235.36	\$92,735.35	\$296,532.88	\$161,794.33	(null)	\$571,297.93
Bikes	\$7,395,348.63	\$19,956,014.67	\$25,551,775.07	\$13,399,243.18	(null)	\$66,302,381.56
Clothing	\$34,376.34	\$485,587.15	\$871,864.19	\$386,013.16	(null)	\$1,777,840.84
Components	\$615,474.98	\$3,610,092.47	\$5,482,497.29	\$2,091,011.92	(null)	\$11,799,076.66
All Products	\$8,065,435.31	\$24,144,429.65	\$32,202,669.43	\$16,038,062.60	(null)	\$80,450,596.98

نحوه‌ی نمایش سرجمع در ردیف، مشابه نمایش سرجمع در ستون میباشد.

**: non empty از تابع استفاده**

برای حذف ستون هایی که کاملاً دارای مقدار null میباشند به صورت زیر عمل میکنیم :

```
Select non empty {[Date].[Calendar].[Calendar Year],[Date].[Calendar]} on columns ,
 {[Product].[Product Categories].[Category],[Product].[Product Categories]} on rows
 From [Adventure Works]
```

خروجی به صورت زیر میباشد:

The screenshot shows the SSAS MDX Query Results window with two tabs: 'Messages' and 'Results'. The 'Results' tab is selected, displaying a table with six columns: CY 2005, CY 2006, CY 2007, CY 2008, All Periods, and Product Categories. The rows list product categories: Accessories, Bikes, Clothing, Components, and All Products. Each cell contains a monetary value.

	CY 2005	CY 2006	CY 2007	CY 2008	All Periods
Accessories	\$20,235.36	\$92,735.35	\$296,532.88	\$161,794.33	\$571,297.93
Bikes	\$7,395,348.63	\$19,956,014.67	\$25,551,775.07	\$13,399,243.18	\$66,302,381.56
Clothing	\$34,376.34	\$485,587.15	\$871,864.19	\$386,013.16	\$1,777,840.84
Components	\$615,474.98	\$3,610,092.47	\$5,482,497.29	\$2,091,011.92	\$11,799,076.66
All Products	\$8,065,435.31	\$24,144,429.65	\$32,202,669.43	\$16,038,062.60	\$80,450,596.98

انتخاب دو دایمنشن در سطر و ستون و مشخص نمودن یک Measure خاص برای کوئری :

برای این کار به صورت زیر عمل خواهیم کرد:

```
Select
{[Date].[Calendar].[Calendar Year],[Date].[Calendar]} on columns,
{[Product].[Product Categories].[Category],[Product].[Product Categories]} on rows
From [Adventure Works]
Where [Measures].[Internet Sales Amount]
```

در اینجا با اعمال شرط Where از SSAS عمل ایم خروجی برای شاخص مشخص شده واکشی شود.

در بالا میزان فروش اینترنتی برای دسته بندی محصولات و در سال های مختلف ارائه و همچنین سر جمع ستون و سطر نیز نمایش داده شده است.

The screenshot shows the SSAS MDX Query Results window with seven columns: CY 2005, CY 2006, CY 2007, CY 2008, CY 2010, All Periods, and Product Categories. The rows list product categories: Accessories, Bikes, Clothing, Components, and All Products. The CY 2010 column contains '(null)' for all categories except All Products, which has a value of \$29,358,677.22.

	CY 2005	CY 2006	CY 2007	CY 2008	CY 2010	All Periods
Accessories	(null)	(null)	\$293,709.71	\$407,050.25	(null)	\$700,759.96
Bikes	\$3,266,373.66	\$6,530,343.53	\$9,359,102.62	\$9,162,324.85	(null)	\$28,318,144.65
Clothing	(null)	(null)	\$138,247.97	\$201,524.64	(null)	\$339,772.61
Components	(null)	(null)	(null)	(null)	(null)	(null)
All Products	\$3,266,373.66	\$6,530,343.53	\$9,791,060.30	\$9,770,899.74	(null)	\$29,358,677.22

در صورتیکه بخواهیم ستون و سطرهایی را که دارای مقدار null در تمامی آن سطر یا ستون می باشند، حذف کنیم به صورت زیر عمل می کنیم:

```
Select
non empty {[Date].[Calendar].[Calendar Year],[Date].[Calendar]} on columns,
non empty {[Product].[Product Categories].[Category],[Product].[Product Categories]} on rows
From [Adventure Works]
Where [Measures].[Internet Sales Amount]
```

	CY 2005	CY 2006	CY 2007	CY 2008	All Periods
Accessories	(null)	(null)	\$293,709.71	\$407,050.25	\$700,759.96
Bikes	\$3,266,373.66	\$6,530,343.53	\$9,359,102.62	\$9,162,324.85	\$28,318,144.65
Clothing	(null)	(null)	\$138,247.97	\$201,524.64	\$339,772.61
All Products	\$3,266,373.66	\$6,530,343.53	\$9,791,060.30	\$9,770,899.74	\$29,358,677.22

اگر در یک دایمنشن فقط یک سلسله مراتب باشد یا اصلا سلسله مراتب وجود نداشته باشد، می توان از نام خود دایمنشن استفاده کرد

```
Select
[Sales Channel] on columns
From [Adventure Works]
```

Messages	Results
	All Sales Channels
	\$80,450,596.98

و دقت داشته باشید دایمنشنی که دارای بیش از یک سلسله مراتب باشد، حتما باید در Select مشخص شود که از کدام سلسله مراتب می خواهیم استفاده کنیم . در غیر این صورت با خطأ مواجه خواهیم شد.

```
Select
[Product] on columns
From [Adventure Works]
```

Messages
Executing the query ...
Query (2, 2) The 'Product' dimension contains more than one hierarchy, therefore the hierarchy must be explicitly specified.
Execution complete

استفاده از فیلد های یک دایمنشن که دارای سلسه مراتب می باشد نیز جایز می باشد

```
Select
[Product].[Category] on columns
From [Adventure Works]
```

All Products
\$80,450,596.98

```
Select
[Product].[Category].[all]    on columns
From [Adventure Works]
--
Select
[Product].[Category].[All] on columns
From [Adventure Works]
--
Select
[Product].[Category].[{(all)}] on columns
From [Adventure Works]
--
Select
[Product].[Category].[all products] on columns
From [Adventure Works]
```

برای به دست آوردن سرجمع کل روی یک صفت از دایمنشن، باید از سه حالت آخر استفاده کرد. حالت اول خطأ دارد و خروجی خالی نمایش داده می شود .

در صورتی که بخواهیم از یک دایمنشن تمامی Member های آن را واکشی کنیم به صورت زیر عمل خواهیم کرد

```
Select
{[Product].[Category].members} on columns
From [Adventure Works]
```

استفاده از Members روی یک خصوصیت در دایمنشن به معنی دریافت سرجمع آن صفت و سپس تک تک اجزای آن صفت می باشد.

All Products	Accessories	Bikes	Clothing	Components
\$80,450,596.98	\$571,297.93	\$66,302,381.56	\$1,777,840.84	\$11,799,076.66

اگر از یک صفت واکشی اطلاعات انجام شود در سطح اعضای آن ، در آن صورت دیگر سرجمع نمایش داده نمی شود و فقط جمع هر عضو در آن صفت نمایش داده می شود .

```
Select
[Product].[Category].[Category].members
-- dimension.hierarchy.level.members
on columns
From [Adventure Works]
```

Accessories	Bikes	Clothing	Components
\$571,297.93	\$66,302,381.56	\$1,777,840.84	\$11,799,076.66

اگر بخواهیم دو ستون را داشته باشیم که هر دو برای یک دایمنشن می‌باشند باید از {} استفاده کرد. دستور اول خطاب خواهد داشت.

```
Select
[Product].[Category].[Category].members,[Product].[Category].[All Products] on columns
From [Adventure Works]
```

در دستور دوم با استفاده از {} خروجی نمایش داده می‌شود که عبارت است از تمامی اعضای سطح [Product].[Category].[Category]. به همراه سرجمع تمامی محصولات.

```
Select
{[Product].[Category].[Category].members,[Product].[Category].[All Products]} on columns
From [Adventure Works]
```

Accessories	Bikes	Clothing	Components	All Products
\$571,297.93	\$66,302,381.56	\$1,777,840.84	\$11,799,076.66	\$80,450,596.98

یک راه کوتاه‌تر برای انتخاب تمامی اعضا و سرجمع آنها

```
Select
{[Product].[Category].[Category],[Product].[Category]}
on columns
From [Adventure Works]
```

می‌توان از کلمات X Members, All استفاده نکرد.

Accessories	Bikes	Clothing	Components	All Products
\$571,297.93	\$66,302,381.56	\$1,777,840.84	\$11,799,076.66	\$80,450,596.98

انتخاب اولین دسته بندی محصول البته این ترتیب بر اساس Key Columns در SSAS می‌باشد.

```
Select
[Product].[Category].&[1]
on columns
From [Adventure Works]
```

Bikes
\$66,302,381.56

انتخاب دقیق یک عضو در خروجی

```
Select
[Product].[Category].[Bikes]
on columns
From [Adventure Works]
```

Bikes
\$66,302,381.56

انتخاب دو عضو از یک دایمنشن

```
Select
{[Product].[Category].[Bikes],[Product].[Category].[Clothing]}
on columns
From [Adventure Works]
```

Bikes	Clothing
\$66,302,381.56	\$1,777,840.84

واکشی تمامی دسته بندی محصولات بر اساس Measure پیش فرض :

```
Select
[Product].[Product Categories].members
on columns
From [Adventure Works]
```

All Products	Accessories	Bike Racks	Hitch Rack - 4-Bike	Bike Stands	All-Purpose Bike Stand	Bottles and Cages	Mountain Bottle Cage	Road Bottl
\$80,450,596.98	\$571,297.93	\$197,736.16	\$197,736.16	(null)	(null)	\$7,476.60	(null)	(null)

در صورتیکه بخواهیم دو Dimension مختلف را در یک سطون یا سطر بیاوریم باید از Join استفاده کنیم. بنابر این دو دستور زیر با خطرا رو برو می‌شوند

```
Select
```

```
[Product].[Product Categories],[Product].[Category]
on columns
From [Adventure Works]

Go

Select
{[Product].[Product Categories],[Product].[Category]}
on columns
From [Adventure Works]
```

```
Executing the query ... 01.
Parser: The statement dialect could not be resolved due to ambiguity.
Execution complete 01.

Executing the query ... 11.
Query (2, 2) Members, tuples or sets must use the same hierarchies in the function.
Execution complete 11.
```

تعريف Axis : به هر کدام از ستون یا سطر یک محور یا Axis گفته می‌شود.

با بررسی مثال فوق به نتایج زیر خواهیم رسید.

1. امکان استفاده از دو سلسله مراتب مختلف از یک دایمنشن در یک Axis وجود ندارد . مگر اینکه آنها را با هم دیگر CrossJoin کنیم .

2. امکان استفاده از دو سلسله مراتب مختلف از یک دایمنشن در دو Axis مختلف وجود دارد .

ترتیب انتخاب Axis ها به صورت زیر می‌باشد:

Columns .1

Rows .2

برای مشخص شدن موضوع کوئری زیر را اجرا کنید

```
Select
[Product].[Product Categories].members
on rows
From [Adventure Works]
```

```
Executing the query ...
Query (2, 2) Axis numbers specified in a query must be sequentially specified, and cannot contain gaps.
Execution complete
```

نمیتوانیم ردیفی را واکشی کنیم بدون اینکه ستونی برای کوئری مشخص کرده باشیم.

البته میتوان ستون خالی ایجاد نماییم مانند مثال زیر :

```
Select
{} on columns,
[Product].[Product Categories].members
on rows
From [Adventure Works]
```

البته در این صورت خروجی فقط نام دسته بندی محصولات خواهد بود زیرا هیچ ستونی مشخص نشده .

All Products
Accessories
Bike Racks
Hitch Rack - 4-Bike
Bike Stands
All-Purpose Bike Stand
Bottles and Cages
Mountain Bottle Cage
Road Bottle Cage
Water Bottle - 30 oz.
Cleaners
Bike Wash - Dissolver
Fenders
Fender Set - Mountain
Helmets
Sport-100 Helmet, Black
Sport-100 Helmet, Black
Sport-100 Helmet, Black

در مقالات بعدی به ادامه مطالب MDX Query خواهیم پرداخت.

## نظرات خوانندگان

نوبسند: علاقه مند  
تاریخ: ۱۳۹۲/۱۰/۱۰ ۰۵:۵۵

با سلام خدمت شما، بسیار سپاسگزارم به خاطر این سری مقالات آموزشی که باعث شد با زبان جذاب و فوق العاده‌ی MDX آشنا شومن.

بنده با گزارشات آماری cross tab آشنا هستم، خروجی مثال هایتان دقیقا مشابه با خروجی گزارشات cross tab است (نمی‌زننه!)

نوشتنيQuery‌های pivoting/cross tabulation با این زبان نسبت به زبان SQL واقعا ساده‌تر است. حقیقتا لذت بردم. فقط خواهشی داشتم، اگر مقدور است به مثال‌هایی بپردازید که در آنها ستون‌ها یا سطرها در Range‌های مختلف گروه بندی شوند.

مثال:

Columns: Age	-10	10 - 20	21 - 30	31-40	+40
Rows: Watching TV (min)					
-20					
20-40					
41-60					
+60					

جدول فوق یک گزارش آماری cross tab است. که مشخص کرده افراد (مرد و زن) در رده‌های سنی مختلف (زیر 10 سال، بین 10 و 20 سال و ...) روزانه چند دقیقه تلویزیون تماشا می‌کنند (کمتر از 20 دقیقه، بین 20 تا 40 دقیقه..). این بازه‌ها را می‌شوند. داده‌هایی مثل تاریخ، روزها، هفته‌ها، ماهها و ... نیز در نظر گرفت.

و سوالی نیز داشتم، آیا می‌توان Query‌های MDX را به T-SQL تبدیل کرد (منظور بصورت خودکار است نه بازنویسی آن)؟ و درخواست پایانی، لطفا به بحث Reporting هم بپردازید که ببینیم این نتیجه در قالب گزارش تجاری چگونه ظاهر می‌شوند.

نوبسند: اردلان شاه قلی  
تاریخ: ۱۳۹۲/۱۰/۱۰ ۱:۳۴

آقای علاقه مند :

در خصوص سوال اول شما ( ستون‌ها یا سطرها در Range‌های مختلف گروه بندی شوند ) این کار را باید با طراحی مناسب جدا اول در DW انجام دهید. به عبارت دیگر جدول Fact در خصوص آمار داشته باشد که یکی از فیلدهای آن مشاهده Dimension تلویزیون می‌باشد و دو عدد Fact که یکی گروه‌های سنی و دیگری ساعت‌ها می‌باشد.

حال باید در ETL واکنشی مناسبی برای پر کردن جدول Fact داشته باشد.

در خصوص تبدیل MDX به T/SQL اطلاع دقیقی ندارم و با قاطعیت نمی‌توانم بگویم نه. اما استدلال بنده این می‌باشد که از آنجایی که روی پایگاه داده‌ی MDB نوشته می‌شوند کاملاً متفاوت از T/SQL می‌باشند که روی مدل رابطه‌ای کار می‌کنند بنابر این تصور می‌کنم چنین امکانی نباشد و در ضمن در جستجوی وب هم چیزی نیافتم.

در خصوص نمایش نهایی این چنین گزارشاتی چندین روش وجود دارد

واکنشی اطلاعات در Excel

استفاده از SharePoint در Performance Point Service (ترجیحاً 2013)

استفاده از کامپوننت‌های OLAP در محیط Visual Studio ..., RadarSoft, Dundas مانند

استفاده از نرم افزار هایی که کارشنان ساختن گزارشات از MDBها میباشد (CubWare)

نوبتده: علاقه مند  
تاریخ: ۱۱:۲ ۱۳۹۲/۱۰/۱۰

سلام مجدد،  
منونم از پاسختتون.

راستش شناختی نسبت به این دو نوع جدولی که فرمودینندارم (Fact, Dimension). اشاره کردین که باید رده بندی های مربوطه یعنی رده سنی و بازه زمانی را باید در جداول Dimension پیاده کرد. آیا این پیاده سازی به معنای عدم داینامیک بودن است؟ منظورم اینه که فرض اگر بخواهیم روزی به جای اینکه سن افراد را ده سال دسته بندی کنیم، تصمیم گرفتیم اینکار را پنج سال پنج سال انجام دهیم آیا باز نیاز هست مجدد یک جدول دیگر طراحی کنیم؟

آیا در این زبان امکان استفاده از عملگرهای ابتدایی جبری مثل تقسیم وجود دارد؟ فرض اگر بخواهیم نسبت مقادیر دو ستون را بدست آوریم (بطور نمونه مقایسه تورم شش ماه اول سال جاری نسبت به شش ماه اول سال گذشته)؟

نوبتده: اردلان شاه قلی  
تاریخ: ۱۶:۲۵ ۱۳۹۲/۱۰/۱۰

در قسمت اول این مجموعه توضیحات لازم را در خصوص مقدمات کار و مفاهیم اولیه خدمتتان ارایه کردم. برای آشنایی با ساخت Data Warehouse لطفا مراجعه کنید به قسمت (مدل داده‌ای رابطه‌ای (Relational) و چند بعدی (Multidimensional)) در سری اول.

در خصوص ایجاد ابعادی مانند بعد تاریخ و بعد سن باید عرض کنم که این ابعاد امکان ارایه انواع گزارشات مختلف را به شما خواهند داد و مطلقا OLAP را از حالت دینامیک خارج نمیکند.

در خصوص نحوه تحلیل و طراحی DW ، امیدوارم به زودی بتوانم مقاله ای را انتشار بدهم.  
در خصوص عملگرهای تقسیم و ... و توابع جمعی به زودی در قسمت‌های بعدی اطلاعات کاملی را خدمتتان ارایه خواهم کرد. به طور خلاصه اینکه OLAP برای ایجاد گزارشات مدیریتی می‌باشد و قطعاً این موارد شما در ادامه پوشش داده خواهد شد. (شما به راحتی می‌توانید سرچمی یک بازه‌ی تاریخی یا سنی را برای یک Measure خاص بدست آورید).

نوبتده: میم  
تاریخ: ۱۱:۴۱ ۱۳۹۳/۰۲/۲۸

با سلام و سپاس فراوان از مطالب پر مغزتان  
سوال من این است که آیا امکان انجام عملیات group by بر روی پایگاه داده چند بعدی وجود دارد؟

اگر دایمنشنی بیشتر از یک ساختار سلسله مراتبی باشد ، باید حتما مشخص شود که اعضای کدام ساختار سلسله مراتبی مورد نظر ما می باشد . بنابراین کوئری های زیر دارای خطای خطا می باشند.

```
Select
[Customer].members on columns
From [Adventure Works]
GO
Select
[Scenario].members on columns
From [Adventure Works]
```

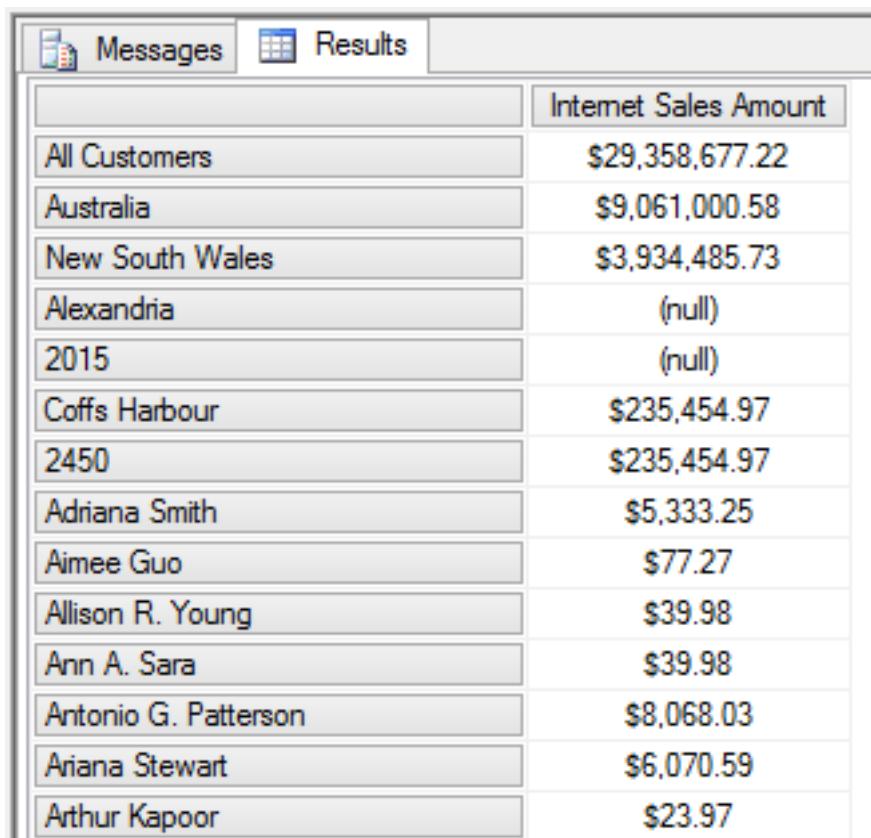
دو کوئری زیر را در نظر بگیرید :

```
Select
[Measures].[Internet Sales Amount] on columns,
[Customer].[Customer Geography].members on rows
From [Adventure Works]
```

```
Select
[Measures].[Internet Sales Amount] on columns,
[Customer].[Customer Geography].[Country].members on rows
From [Adventure Works]
```

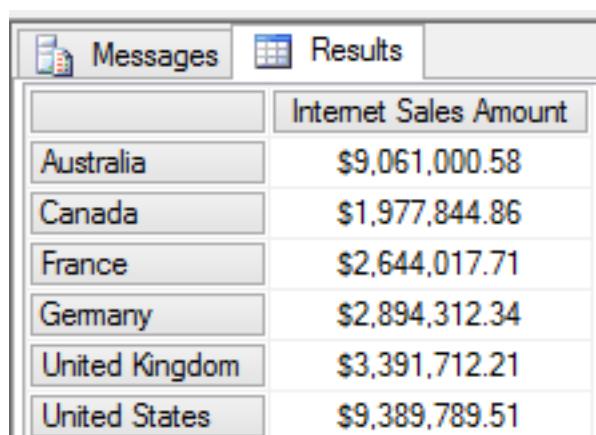
۹

تفاوت این دو کوئری در واکنشی اطلاعات از اعضای سطح ساختار سلسله مراتبی دایمنشن می باشد. برای توضیح بیشتر ابتدا به خروجی آن ها نگاهی بیندازید.



The screenshot shows the 'Results' tab of Analysis Manager. The table has two columns: 'Customer' and 'Internet Sales Amount'. The data includes:

Customer	Internet Sales Amount
All Customers	\$29,358,677.22
Australia	\$9,061,000.58
New South Wales	\$3,934,485.73
Alexandria	(null)
2015	(null)
Coffs Harbour	\$235,454.97
2450	\$235,454.97
Adriana Smith	\$5,333.25
Aimee Guo	\$77.27
Allison R. Young	\$39.98
Ann A. Sara	\$39.98
Antonio G. Patterson	\$8,068.03
Ariana Stewart	\$6,070.59
Arthur Kapoor	\$23.97

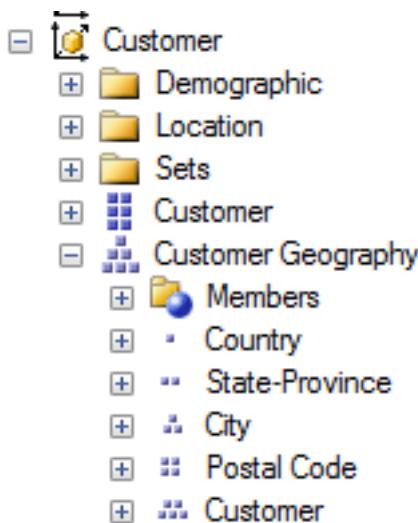


The screenshot shows the 'Results' tab of Analysis Manager. The table has two columns: 'Country' and 'Internet Sales Amount'. The data includes:

Country	Internet Sales Amount
Australia	\$9,061,000.58
Canada	\$1,977,844.86
France	\$2,644,017.71
Germany	\$2,894,312.34
United Kingdom	\$3,391,712.21
United States	\$9,389,789.51

در کوئیری اول تمامی سطوح موجود در ساختار سلسله مراتبی دایمنشن مورد نظر واکشی می‌شود. در کوئیری دوم فقط سطح کشور از ساختار سلسله مراتبی واکشی می‌شود.

ساختار سلسله مراتبی دایمنشن [Customer] در شکل زیر نمایش داده شده است.

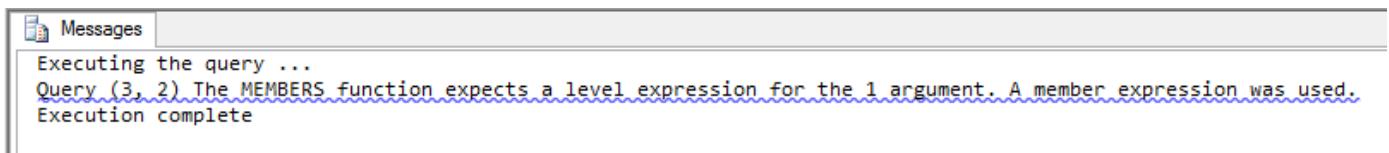


امکان واکشی عضوهای یک عضو از ساختار سلسله مراتبی وجود ندارد

```

Select
[Measures].[Internet Sales Amount] on columns,
[Customer].[Customer Geography].[Country].[France].members on rows
From [Adventure Works]
  
```

دقیق کنید که France خودش عضو سطح Country در ساختار سلسله مراتبی Customer Geography در دایمنشن می‌باشد و دیگر امکان واکشی اعضای این عضو وجود ندارد



بچه‌های یک عضو را به شکل زیر می‌توان واکشی نمود.

```

Select
[Measures].[Internet Sales Amount] on columns,
[Customer].[Customer Geography].[Country].[France].children on rows
From [Adventure Works]
  
```

برای بدست آوردن بچه‌های یک عضو در یک ساختار سلسله مراتبی، از children استفاده می‌شود.

	Internet Sales Amount
Charente-Maritime	\$34,441.73
Essonne	\$279,297.18
Garonne (Haute)	\$54,641.72
Gers	(null)
Hauts de Seine	\$263,416.19
Loir et Cher	\$21,473.74
Loiret	\$91,562.91
Moselle	\$94,046.23
Nord	\$391,400.20
Pas de Calais	\$11,342.92
Seine (Paris)	\$539,725.80
Seine et Marnes	\$109,735.24
Seine Saint Denis	\$379,479.75
Somme	\$29,555.28

دقت داشته باشید، همانگونه که عضو، دارای Members نمیباشد، یک بچه هم دارای Children نمیباشد. بنابراین کوئری زیر با خطای مواجه میشود.

```
Select
[Measures].[Internet Sales Amount] on columns,
[Customer].[Customer Geography].[Country].[France].children.children on rows
From [Adventure Works]
```

البته شاید تصور میشود که باید به عنوان یک ساختار سلسله مراتبی بعد از کشور، استان و سپس شهرها واکشی شوند. اما برای واکشی ساختار سلسله مراتبی باید به صورت زیر عمل کرد.

```
Select
[Measures].[Internet Sales Amount] on columns,
descendants(
[Customer].[Customer Geography].[Country].[France],
[Customer].[Customer Geography].[City]
) on rows
From [Adventure Works]
```

عمل برای یافتن شهرهای کشور فرانسه، باید دو سطح در ساختار سلسله مراتبی پایین برویم. برای این منظور از تابع descendants () استفاده کرده ایم و به خاطر داشته باشید که نوشتن [France].children.children مجاز نمیباشد. همچنین در تابع فوق دو پارامتر ورودی داریم؛ اولی مشخص کننده‌ی یک عضو از سطح مبدأ میباشد و دومین پارامتر مشخص کننده‌ی سطحی است که باید واکشی در آن صورت بگیرد.

The screenshot shows a Microsoft Analysis Services query results window. The top bar has tabs for 'Messages' and 'Results'. The 'Results' tab is selected, displaying a table with two columns: 'City' and 'Internet Sales Amount'. The data is as follows:

	Internet Sales Amount
Saint Ouen	\$34,441.73
Les Ulis	\$181,244.73
Morangis	\$56,432.84
Vemieres Le Buisson	\$41,619.61
Colomiers	\$54,641.72
Aujan Moumede	(null)
Boulogne-Billancourt	\$14,289.24
Colombes	\$90,268.51
Courbevoie	\$38,809.63
Paris La Defense	\$45,350.86
Sèvres	\$39,598.20
Suresnes	\$35,099.73
Saint Ouen	\$21,473.74
Orleans	\$91,562.91

برای مشخص تر شدن موضوع به مثال زیر توجه کنید

```
Select
[Measures].[Internet Sales Amount] on columns,
descendants{
[Customer].[Customer Geography].[Country].[France],
[Customer].[Customer Geography].[Customer]
} on rows
From [Adventure Works]
```

در این مثال واکشی در چهار سطح پایین تر انجام شده است و به این معنی می باشد که ما مشتریان کشور فرانسه را در ردیفها واکشی کردہ ایم.

	Internet Sales Amount
Brad Nath	\$48.97
Bridget C. Nath	\$1,560.43
Chelsea Jordan	\$1,735.98
Clayton Zhang	\$2,049.10
Colin R. Wang	\$47.97
Craig R. Gill	\$23.78
Daren A. Ruiz	\$43.46
Dominic A. Garcia	\$1,382.97
Edwin Zhou	\$2,753.82
Jon Cai	\$88.97
Julie Pal	\$2,778.52
Katie Shan	\$8.99
Levi W. Rodriguez	\$4.99
Nancy Schmidt	\$5,033.14

: Member عدم وجود یک

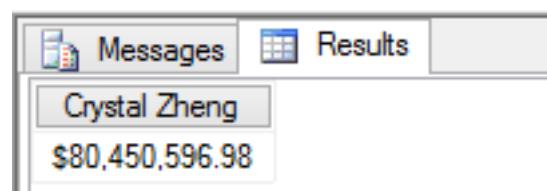
در صورت عدم وجود یک عضو در واکشی، خروجی خالی می باشد و خطای نمایش داده نمی شود.

```
Select
[Customer].[Customer Geography].[Customer].[Crystal Zhen] on columns
From [Adventure Works]
```

مشتری با نام Crystal Zhen وجود ندارد؛ بنابراین نتیجه خالی می باشد.

پیدا کردن یک عضو در مجموعه ای از اعضاء، البته اگر این عضو در آن مجموعه وجود نداشته باشد خروجی خالی می باشد.

```
Select
exists(
[Customer].[Customer Geography].[Customer].[Crystal Zheng],
[Customer].[Customer Geography].members
)on columns
From [Adventure Works]
```



در نوشتن کوئری ها دقت کنید که حتما بین دایمنشن ها و شاخص ها در SSAS ارتباط ایجاد شده باشد. در غیر این صورت خروجی نامعتبر خواهد بود. به عنوان مثال در کوئری زیر بین شاخص [Measures].[Reseller Sales Amount] و مشتری ارتباطی وجود ندارد (این ارتباط باید در هنگام ساخت Cube در SSAS ایجاد می گردید تا در هنگام Deploy کردن محاسبات لازم صورت بگیرد) بنابر این در خروجی شاهد نمایش شاخص پیش فرض Cube می باشیم.

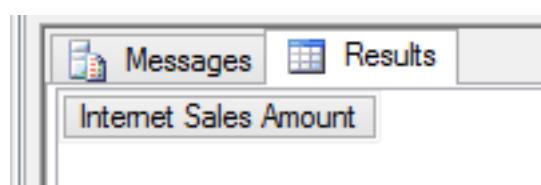
```
Select
[Measures].[Reseller Sales Amount] on columns,
[Customer].[Customer Geography].[Customer].members on rows
From [Adventure Works]
```

	Reseller Sales Amount
Adriana Smith	\$80,450,596.98
Aimee Guo	\$80,450,596.98
Allison R. Young	\$80,450,596.98
Ann A. Sara	\$80,450,596.98
Antonio G. Patterson	\$80,450,596.98
Ariana Stewart	\$80,450,596.98
Arthur Kapoor	\$80,450,596.98
Barbara W. Lal	\$80,450,596.98
Bobby D. Saunders	\$80,450,596.98
Brianna J. Johnson	\$80,450,596.98
Bruce G. Madan	\$80,450,596.98
Bryant L. Perez	\$80,450,596.98
Carla D. Madan	\$80,450,596.98
Carlos Edwards	\$80,450,596.98

قبل اگفته شد که هر سطح دارای Members می باشد و میتوان برای هر عضو را نمایش داد، با این وجود دقت کنید که اعضای آخرین سطح ساختار سلسله مراتبی دارای Children نمی باشند.

```
Select
[Measures].[Internet Sales Amount] on columns,
[Customer].[Customer Geography].[Customer].[Crystal Zheng].children on rows
From [Adventure Works]
```

خروجی خالی می باشد.



بدست آوردن پدر یک عضو در ساختار سلسله مراتبی :

کوئری زیر را اجرا کنید

```
Select
[Measures].[Internet Sales Amount] on columns,
[Customer].[Customer Geography].[Customer].[Crystal Zheng].parent on rows
From [Adventure Works]
```

با توجه به ساختار سلسله مراتبی تعریف شده، پدر این مشتری یک کد پستی می باشد .

برای بدست آوردن پدربرگ یک ساختار سلسله مراتبی داریم :

```
Select
[Measures].[Internet Sales Amount] on columns,
[Customer].[Customer Geography].[Customer].[Crystal Zheng].parent.parent on rows
From [Adventure Works]
```

و برای بدست آوردن پدر در 4 سطح بالاتر داریم

```
Select
[Measures].[Internet Sales Amount] on columns,
[Customer].[Customer Geography].[Customer].[Crystal Zheng].parent.parent.parent.parent
on rows
From [Adventure Works]
```

در صورتیکه با بالا رفتن از سطوح یک ساختار سلسله مراتبی از ریشه رد شویم ، خروجی خالی بر می گردد .

```
Select
[Measures].[Internet Sales Amount] on columns,
[Customer].[Customer Geography].[Customer].[Crystal Zheng].parent
.parent.parent.parent.parent
on rows
From [Adventure Works]
```

برای بدست آوردن نیاز می توانیم از تابع ancestor() استفاده کنیم.

بدست آوردن نیاز برای یک مشتری در سطح شهر

```
Select
[Measures].[Internet Sales Amount] on columns,
ancestor(
[Customer].[Customer Geography].[Customer].[Crystal Zheng],
[Customer].[Customer Geography].[City]
)on rows
From [Adventure Works]
```

Messages		Results
	Internet Sales Amount	
Versailles	\$102,657.25	

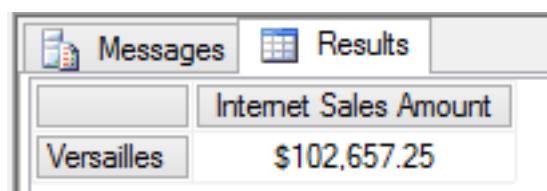
بدست آوردن نیا برای یک مشتری در سطح شهر

```
Select
[Measures].[Internet Sales Amount] on columns,
ancestor(
[Customer].[Customer Geography].[Customer].[Crystal Zheng],
[Customer].[Customer Geography].[City]
) on rows
From [Adventure Works]
```

پارامتر دوم در تابع ancestor می‌تواند هم مشخص کننده‌ی یک سطح باشد توسط نام آن سطح و هم می‌تواند یک عدد باشد که مشخص کننده‌ی تعداد سطحی است که باید در واکشی اطلاعات از ساختار سلسله مراتبی بالا برویم.

```
Select
[Measures].[Internet Sales Amount] on columns,
ancestor(
[Customer].[Customer Geography].[Customer].[Crystal Zheng],
2
) on rows
From [Adventure Works]
```

بدست آوردن نیا برای یک مشتری در دو سطح بالاتر



در صورتی که بخواهیم تمامی سطوح بالاتر از یک سطح را داشته باشیم و بعد هم سر جمع داشته باشیم، به صورت زیر عمل می‌کنیم و از تابع ascendants استفاده می‌کنیم.

```
Select
[Measures].[Internet Sales Amount] on columns,
ascendants(
[Customer].[Customer Geography].[Customer].[Crystal Zheng]
) on rows
From [Adventure Works]
```

The screenshot shows the SSMS interface with the 'Results' tab selected. The results are displayed in a table with two columns: 'Customer' and 'Internet Sales Amount'. The data is as follows:

	Internet Sales Amount
Crystal Zheng	\$60.47
78000	\$102,657.25
Versailles	\$102,657.25
Yveline	\$268,664.80
France	\$2,644,017.71
All Customers	\$29,358,677.22

دقت داشته باشید که در این تابع تمامی سطوح از سطح مشخص شده با سمت ریشه باز گردانده می‌شوند. در حالیکه در تابع `ancestor` فقط سطح مشخص شده در ساختار سلسله مراتبی به سمت ریشه باز گردانده می‌شد.

مرتب سازی اعضای یک ساختار سلسله مراتبی از بالا به پایین به صورت نزولی

```
select
[Measures].[Internet Sales Amount] on columns,
hierarchize(
ascendants(
[Customer].[Customer Geography].[Customer].[Crystal Zheng]
)
) on rows
From [Adventure Works]
```

The screenshot shows the SSMS interface with the 'Results' tab selected. The results are displayed in a table with two columns: 'Customer' and 'Internet Sales Amount'. The data is as follows:

	Internet Sales Amount
All Customers	\$29,358,677.22
France	\$2,644,017.71
Yveline	\$268,664.80
Versailles	\$102,657.25
78000	\$102,657.25
Crystal Zheng	\$60.47

برای مرتب سازی یک ساختار سلسله مراتبی از تابع `hierarchize()` استفاده می‌کنیم. دقت داشته باشید که کار بر روی ساختارهای سلسله مراتبی یکی از قسمت‌های اصلی در MDX Query ها می‌باشد.

ادامه‌ی این مطالب و کار با ساختارهای سلسله مراتبی را در مقالات بعدی دنبال خواهیم کرد.

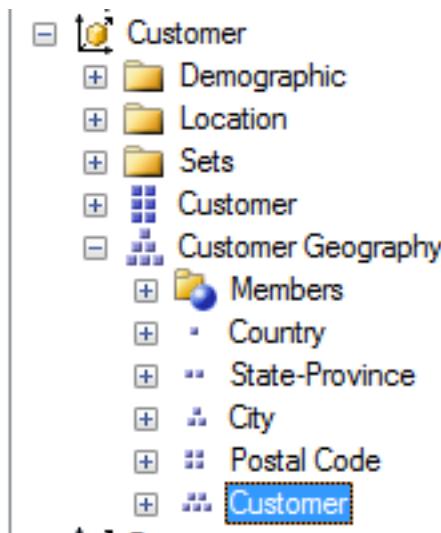
در این قسمت می خواهیم بیشتر در خصوص توابع مرتبط با ساختار سلسله مراتبی صحبت کنیم. برای آشنایی با این توابع و امکانات MDX Query ، مقاله را با بررسی چندین Query دنبال خواهیم کرد.

بدست آوردن تمامی برادران یک سطح خاص :

```
Select
[Measures].[Internet Sales Amount] on columns,
[Customer].[Customer Geography].[Customer].[Crystal Zheng].parent.children on rows
From [Adventure Works]
```

در کوئری بالا تمامی مشتریانی را که دارای کد پستی مشابه با کد پستی [Crystal Zheng] می باشند، واکنشی کرده ایم.

به عبارت دیگر با اعمال [Crystal Zheng].parent ، به کد پستی مشتری دسترسی پیدا کرده ایم (برای درک بیشتر در زیر ساختار سلسله مراتبی موقعیت جغرافیایی مشتریان را ببینید) و سپس با اعمال children به تمامی مشتریان موجود در آن کد پستی رسیده ایم؛ که عملا همان برادران [Crystal Zheng] می باشند.



نتیجه کوئری بالا در زیر نمایش داده شده است

	Internet Sales Amount
Abby Sandberg	\$5,948.23
Adam Hayes	\$597.95
Alexa Peterson	\$3,350.42
Blake Roberts	\$8.99
Caleb Washington	\$1,145.48
Carolyn Sanchez	\$2,419.06
Chloe L. Ross	\$2,049.10
Christian C. Davis	\$7.28
Clayton C. Gao	\$89.97
Colin M. Zheng	\$38.98
Crystal Zheng	\$60.47
Dana Alvarez	\$2,803.52
Daren E. Sai	\$2,420.34
David B. Martin	\$42.97

راه بهتر برای بدست آوردن تمامی برادران یک سطح، استفاده از تابع `siblings` می‌باشد.

```
Select
[Measures].[Internet Sales Amount] on columns,
[Customer].[Customer Geography].[Customer].[Crystal Zheng].siblings on rows
From [Adventure Works]
```

کوئری‌های بالا جواب یکسانی را بر می‌گردانند. به عبارت دیگر تابع `siblings` عملکار دو تابع `parent.children` را انجام میدهد

	Internet Sales Amount
Abby Sandberg	\$5,948.23
Adam Hayes	\$597.95
Alexa Peterson	\$3,350.42
Blake Roberts	\$8.99
Caleb Washington	\$1,145.48
Carolyn Sanchez	\$2,419.06
Chloe L. Ross	\$2,049.10
Christian C. Davis	\$7.28
Clayton C. Gao	\$89.97
Colin M. Zheng	\$38.98
Crystal Zheng	\$60.47
Dana Alvarez	\$2,803.52
Daren E. Sai	\$2,420.34
David B. Martin	\$42.97

برای بدست آوردن برادر ارشد به صورت زیر عمل می کنیم (اولین بچه در ساختار سلسله مراتبی)

```
Select
[Measures].[Internet Sales Amount] on columns,
[Customer].[Customer Geography].[Customer].[Crystal Zheng].parent.firstChild on rows
From [Adventure Works]
```

	Internet Sales Amount
Abby Sandberg	\$5,948.23

و یا از تابع زیر استفاده می کنیم

```
Select
[Measures].[Internet Sales Amount] on columns,
[Customer].[Customer Geography].[Customer].[Crystal Zheng].firstsibling on rows
From [Adventure Works]
```

هر دو کوئری به جواب یکسان خواهند رسید.

و برای بدست آوردن آخرین برادر در ساختار سلسله مراتبی (برادر ته تغایری) از دو روش زیر می توان استفاده کرد.

```
Select
[Measures].[Internet Sales Amount] on columns,
[Customer].[Customer Geography].[Customer].[Crystal Zheng].parent.lastchild on rows
From [Adventure Works]
```

```
Select
[Measures].[Internet Sales Amount] on columns,
[Customer].[Customer Geography].[Customer].[Crystal Zheng].lastsibling on rows
From [Adventure Works]
```

	Internet Sales Amount
Xavier White	\$3.99

برای توضیح بیشتر می‌توان اضافه کرد که در کوئری بالا میزان فروش اینترنتی را برای آخرین مشتری در موقعیت جغرافیایی مشتری با نام [Crystal Zheng] واکشی شده است.

حال تصور کنید بخواهیم میزان فروش اینترنتی را برای تمامی مشتریان ایالت [Yveline] بدست بیاوریم. در این صورت MDX Query به شکل زیر خواهد بود

```
Select
[Measures].[Internet Sales Amount] on columns,
descendants(
[Customer].[Customer Geography].[State-Province].[Yveline]
,[Customer].[Customer Geography].[Customer])
on rows
From [Adventure Works]
```

تابع descendants دارای دو پارامتر می‌باشد. اولی برای مشخص نمودن شروع و مبدا در ساختار سلسله مراتبی و دومین برای مشخص کردن سطح واکشی در ساختار سلسله مراتبی می‌باشد. به عبارت دیگر در کوئری بالا تمامی زاد و رود ایالت [Yveline] در سطح شهر واکشی شده است و میزان فروش اینترنتی آن نمایش داده شده است.

در زیر یک کوئری ترکیبی با استفاده از دو تابع ancestor و descendants نوشته شده است.

```
Select
[Measures].[Internet Sales Amount] on columns,
descendants(
ancestor(
[Customer].[Customer Geography].[Customer].[Crystal Zheng],
[Customer].[Customer Geography].[State-Province]
)
,[Customer].[Customer Geography].[Customer])
on rows
From [Adventure Works]
```

در اینجا ابتدا جد یک مشتری در سطح ایالت بدست آمده سپس زاد و رود آن در سطح مشتری بدست می‌آید.

	Internet Sales Amount
Abby P. Rana	\$36.96
Aidan Washington	\$548.98
Alexa Morris	\$553.97
Alison Sharma	\$60.47
Anne Alonso	\$597.95
Beth R. Alonso	\$5,923.23
Billy L. Jiménez	\$39.98
Bryan E. Bailey	\$3,010.03
Charles J. Bell	\$7.28
Charles M. Smith	\$36.27
Christopher R. Clark	\$866.31

برای بدست آوردن فروش اینترنتی تمامی شهرهای کشور فرانسه می‌توانیم به صورت زیر عمل کنیم.

```
Select
[Measures].[Internet Sales Amount] on columns,
descendants(
[Customer].[Customer Geography].[Country].[France],
[Customer].[Customer Geography].[City]
) on rows
From [Adventure Works]
```

	Internet Sales Amount
Saint Ouen	\$34,441.73
Les Ulis	\$181,244.73
Morangis	\$56,432.84
Vernieres Le Buisson	\$41,619.61
Colomiers	\$54,641.72
Aujan Moumede	(null)
Boulogne-Billancourt	\$14,289.24
Colombes	\$90,268.51
Courbevoie	\$38,809.63
Paris La Defense	\$45,350.86
Sèvres	\$39,598.20

تابع `descendants` دارای یک پارامتر سوم هم می‌باشد که مشخص کننده میزان واکشی سطوح می‌باشد و به صورت پیش فرض `Self` می‌باشد. بنابر این کوئری بالا و پایین، نتیجه یکسان خواهد داشت

```
Select
```

```
[Measures].[Internet Sales Amount] on columns,
descendants(
[Customer].[Customer Geography].[Country].[France],
[Customer].[Customer Geography].[City],
self
) on rows
From [Adventure Works]
```

	Internet Sales Amount
Saint Ouen	\$34,441.73
Les Ulis	\$181,244.73
Morangis	\$56,432.84
Vermieres Le Buisson	\$41,619.61
Colomiers	\$54,641.72
Aujan Moumede	(null)
Boulogne-Billancourt	\$14,289.24
Colombes	\$90,268.51
Courbevoie	\$38,809.63
Paris La Defense	\$45,350.86
Sèvres	\$39,598.20

حال اگر بخواهیم فروش اینترنتی را برای تمامی زاد و رود کشور فرانسه از سطح شهر به پایین واکشی کنیم داریم :

```
Select
[Measures].[Internet Sales Amount] on columns,
descendants(
[Customer].[Customer Geography].[Country].[France],
[Customer].[Customer Geography].[City],
self_and_after
) on rows
From [Adventure Works]
```

در این حالات تمامی زاد و رود کشور فرانسه از سطح شهر به پایین در خروجی قرار می گیرد . به این صورت که ابتدا اولین شهر می آید؛ سپس اولین کد پستی در آن شهر و بعد تمامی مشتری های آن کد پستی و بعد کد پستی بعدی و ...

	Internet Sales Amount
Saint Ouen	\$34,441.73
17490	\$34,441.73
Brad Nath	\$48.97
Bridget C. Nath	\$1,560.43
Chelsea Jordan	\$1,735.98
Clayton Zhang	\$2,049.10
Colin R. Wang	\$47.97
Craig R. Gill	\$23.78
Daren A. Ruiz	\$43.46
Dominic A. Garcia	\$1,382.97
Edwin Zhou	\$2,753.82
Jon Cai	\$88.97
Julie Pal	\$2,778.52
Katie Shan	\$8.99
Levi W. Rodriguez	\$4.99
Nancy Schmidt	\$5,033.14
Peter D. She	\$691.91
Russell A. Sharma	\$6,999.57
Sabrina Serrano	\$3,216.90
Shaun N. Andersen	\$13.98
Theresa H. Alvarez	\$5,923.23
Tony K. Raji	\$7.28
Wesley Sun	\$27.77
Les Ulis	\$181,244.73
91940	\$181,244.73
Alicia D. Shen	\$9,458.18
Alvin K. Liu	\$786.33
All Data Items	\$2,001.04

به دست آوردن تمامی زاد و رود فرانسه از سطح بعد از شهر.

به عبارت دیگر ، خروجی باتوجه به ساختار سلسله مراتبی تعریف شده عبارت است از کد پستی و تمام مشتریان آن کد پستی و سپس کد پستی بعدی .

```
Select
[Measures].[Internet Sales Amount] on columns,
descendants(
[Customer].[Customer Geography].[Country].[France],
[Customer].[Customer Geography].[City],
after
)on rows
From [Adventure Works]
```

در کوئری فوق، خود شهر در خروجی نمایش داده نمی شود.

	Internet Sales Amount
17490	\$34,441.73
Brad Nath	\$48.97
Bridget C. Nath	\$1,560.43
Chelsea Jordan	\$1,735.98
Clayton Zhang	\$2,049.10
Colin R. Wang	\$47.97
Craig R. Gill	\$23.78
Darren A. Ruiz	\$43.46
Dominic A. Garcia	\$1,382.97
Edwin Zhou	\$2,753.82
Jon Cai	\$88.97
Ivilia Daji	\$2,770.57

به دست آوردن زاد و رود فرانسه تا یک سطح قبل از شهر.

در این حالت فرانسه و تمامی ایالت های آن در خروجی آورده می شود .

```
Select
[Measures].[Internet Sales Amount] on columns,
descendants(
[Customer].[Customer Geography].[Country].[France],
[Customer].[Customer Geography].[City],
before
)on rows
From [Adventure Works]
```

	Internet Sales Amount
France	\$2,644,017.71
Charente-Maritime	\$34,441.73
Essonne	\$279,297.18
Garonne (Haute)	\$54,641.72
Gers	(null)
Hauts de Seine	\$263,416.19
Loir et Cher	\$21,473.74
Loiret	\$91,562.91
Moselle	\$94,046.23
Nord	\$391,400.20
Pas de Calais	\$11,342.92
Seine (Paris)	\$539,725.80

همچنین می‌توان دومین پارامتر تابع را به صورت عدد وارد کرد و این عدد بیانگر تعداد سطح پایین‌تر از پارامتر اول در ساختار سلسله مراتبی می‌باشد.

به عنوان مثال :

```
Select
[Measures].[Internet Sales Amount] on columns,
descendants(
[Customer].[Customer Geography].[Country].[France],
2,
before
) on rows
From [Adventure Works]
```

در این حالت فرانسه و تمامی ایالت‌های آن در خروجی قرار می‌گیرد.

در ابتدا دو سطح ارکشور پایین می‌رویم و به شهر می‌رسیم و بعد زاد و رود فرانسه تا یکی قبل از شهر را بر می‌گرداند.

	Internet Sales Amount
France	\$2,644,017.71
Charente-Maritime	\$34,441.73
Essonne	\$279,297.18
Garonne (Haute)	\$54,641.72
Gers	(null)
Hauts de Seine	\$263,416.19
Loir et Cher	\$21,473.74
Loiret	\$91,562.91
Moselle	\$94,046.23
Nord	\$391,400.20
Pas de Calais	\$11,342.92
Seine (Paris)	\$539,725.80

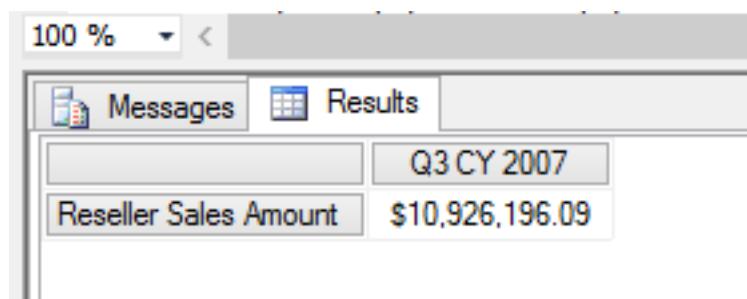
در قسمت های بعدی در خصوص دیگر توابع مرتبط با ساختارهای سلسله مراتبی، توضیحاتی را ارایه خواهیم کرد.

در این قسمت در خصوص توابع مرتبط با ساختار سلسله مراتبی صحبت خواهد شد.

```
Select
{
    [Date].[Calendar].[Calendar Quarter].[Q1 CY 2006],
    cousin(
    [Date].[Calendar].[Calendar Quarter].[Q1 CY 2006],
    [Date].[Calendar].[Calendar Year].[CY 2007]
    )
} on columns,
[Measures].[Reseller Sales Amount] on rows
From [Adventure Works]
```

تابع عمو زاده به این صورت کار می کند که دو پارامتر اول سطح فعلی را مشخص می کند . پارامتر دوم سطح بالاتر از سطح اول را مشخص می کند در ساختار سلسله مراتبی و خروجی برابر است با سطحی برابر سطح پارامتر اول در زیر مجموعه‌ی پارامتر دوم و هم تراز پارامتر اول .

خروجی به صورت زیر می باشد:



خوب حالا به ساختار زیر دقت کنید (ساختار سلسله مراتبی Date )

The screenshot shows the structure of the Date dimension in a Microsoft Analysis Services cube browser. The hierarchy is as follows:

- Date
  - Calendar
    - Date.Calendar Quarter of Year
    - Date.Calendar Semester of Year
    - Date.Calendar Week of Year
    - Date.Calendar Year
    - Date.Calendar
      - Members
        - Calendar Year
          - CY 2005
          - CY 2006
          - CY 2007
          - CY 2008
          - CY 2010
        - Calendar Semester
      - Calendar Quarter
        - Member Properties
          - Q3 CY 2005
          - Q4 CY 2005
          - Q1 CY 2006
          - Q2 CY 2006
          - Q3 CY 2006
          - Q4 CY 2006
          - Q1 CY 2007
          - Q2 CY 2007
          - Q3 CY 2007
          - Q4 CY 2007
          - Q1 CY 2008
          - Q2 CY 2008
          - Q3 CY 2008
          - Q4 CY 2010
    - Month
    - Date

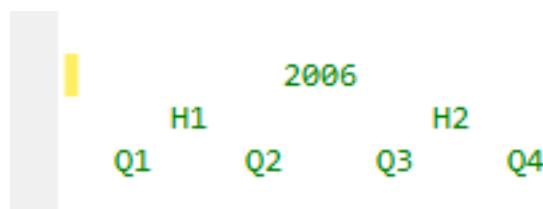
همانطور که مشخص می‌باشد تاریخ‌ها از 2005 تا 2008 و سال 2010 می‌باشند و فصول عبارتند از دو فصل پایانی سال 2005 و تمامی فصول سال 2006 و 2007 و سه فصل اول سال 2008 و فصل چهارم سال 2010. حال دوباره به کوئری نوشته شده دقت کنید. در کوئری بالا فصل همسطح فصل اول سال 2006 در سال 2007 مورد واکشی قرار گرفته است که همان فصل اول در سال 2007 می‌باشد.

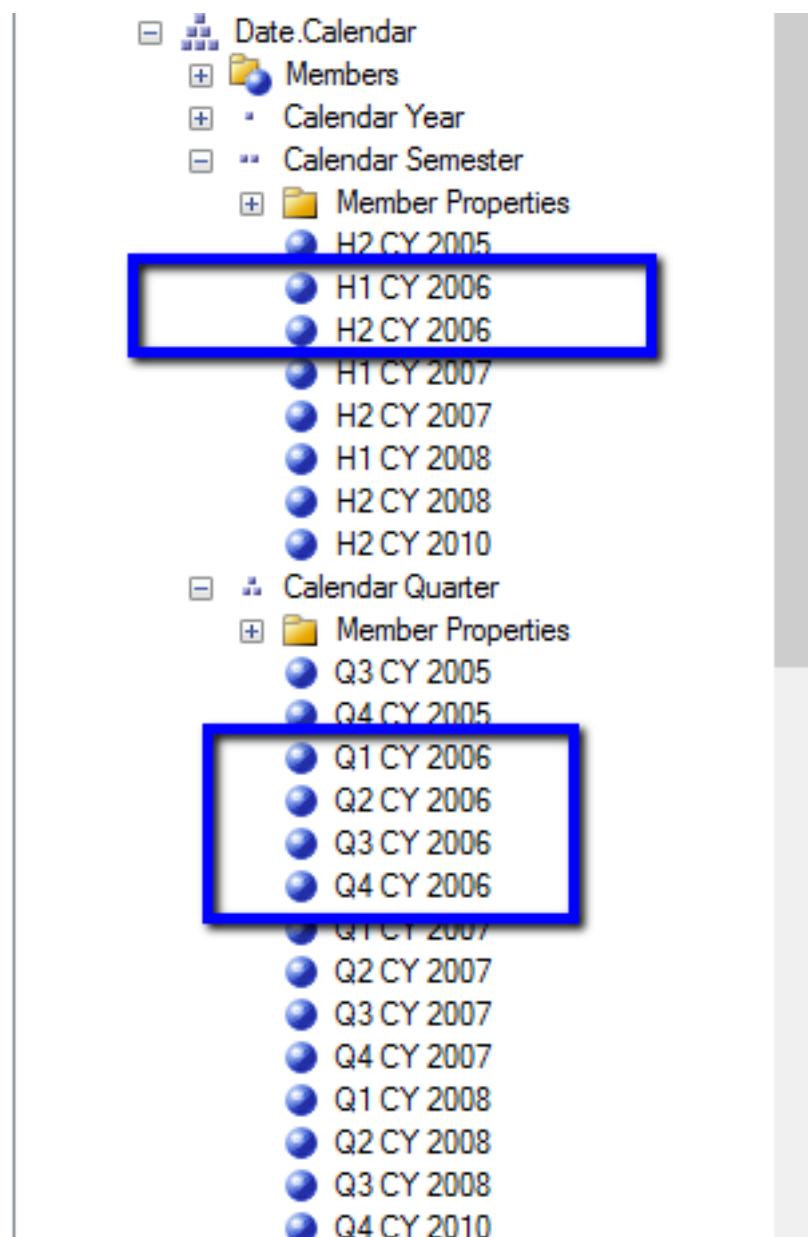
حال به بررسی کوئری زیر خواهیم پرداخت:

```
Select
{
    [Date].[Calendar].[Calendar Quarter].[Q1 CY 2006],
    cousin(
        [Date].[Calendar].[Calendar Quarter].[Q1 CY 2006],
        [Date].[Calendar].[Calendar Semester].[H2 CY 2006]
    )
} on columns,
[Measures].[Reseller Sales Amount] on rows
From [Adventure Works]
```

در این کوئری ما ابتدا ستون فصل اول سال 2006 را بر می‌گردانیم. سپس در تابع پسر عمومی در نیم فصل دوم سال 2006 به دنبال هم سطح فصل اول 2006 می‌گردیم.

نمودار درختی زیر توضیح کاملی به ما خواهد داد:





حال برای ادامه مطلب کار بر روی ساختارهای سلسله مراتبی، ابتدا باید در خصوص نحوه ایجاد Range توضیحاتی ارایه گردد.  
دو کوئری زیر را در نظر گرفته و خروجی آنها را با هم مقایسه نمایید

```
Select
{
  [Date].[Calendar].[Calendar Quarter].[Q1 CY 2006],
  [Date].[Calendar].[Calendar Quarter].[Q2 CY 2006],
  [Date].[Calendar].[Calendar Quarter].[Q3 CY 2006]
} on columns,
[Measures].[Reseller Sales Amount] on rows
From [Adventure Works]
```

9

```
Select
[Date].[Calendar].[Calendar Quarter].[Q1 CY 2006]:
[Date].[Calendar].[Calendar Quarter].[Q3 CY 2006]
on columns,
[Measures].[Reseller Sales Amount] on rows
From [Adventure Works]
```

خروجی‌ها به صورت زیر می‌باشد:

	Q1 CY 2006	Q2 CY 2006	Q3 CY 2006
Reseller Sales Amount	\$4,069,186.04	\$4,153,820.42	\$8,880,239.44

۹

	Q1 CY 2006	Q2 CY 2006	Q3 CY 2006
Reseller Sales Amount	\$4,069,186.04	\$4,153,820.42	\$8,880,239.44

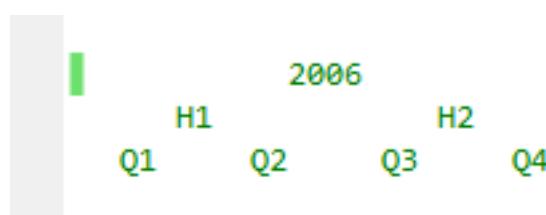
مشخص می‌باشد که از علامت <:> برای ایجاد یک محدوده و جلوگیری از تولید کدهای بلند و طولانی استفاده می‌شود.

حال کوئری زیر را اجرا کنید:

```
Select
[Date].[Calendar].[Calendar Quarter].[Q1 CY 2006]
:
cousin(
[Date].[Calendar].[Calendar Quarter].[Q1 CY 2006],
[Date].[Calendar].[Calendar Semester].[H2 CY 2006]
) on columns,
[Measures].[Reseller Sales Amount] on rows
From [Adventure Works]
```

در این کوئری در ابتدا تابع پسر عمومی گردد، سپس تابع رنج اجرا می‌گردد و در نتیجه، فاصله‌ی بین Q1 CY 2006 تا Q3 CY 2006 را بدست می‌آورد.

نمودار درختی زیر توضیح کاملی به ما خواهد داد:



خروجی به صورت زیر میباشد

The screenshot shows a Microsoft Analysis Services query results window. The window has two tabs at the top: 'Messages' (disabled) and 'Results'. The 'Results' tab is selected, displaying a table with four columns: Q1 CY 2006, Q2 CY 2006, and Q3 CY 2006. The table has one row with the header 'Reseller Sales Amount' and three data cells containing monetary values: \$4,069,186.04, \$4,153,820.42, and \$8,880,239.44 respectively.

	Q1 CY 2006	Q2 CY 2006	Q3 CY 2006
Reseller Sales Amount	\$4,069,186.04	\$4,153,820.42	\$8,880,239.44

در قسمت های بعدی دیگر توابع MDX Query ها را بررسی میکنیم.

## آموزش MDX Query - قسمت یازدهم - استفاده از توابع Lag و Lead

عنوان:

اردلان شاه قلی

نوبسنده: تاریخ: ۱۶:۳۵ ۱۳۹۲/۱۰/۰۶

آدرس: [www.dotnettips.info](http://www.dotnettips.info)

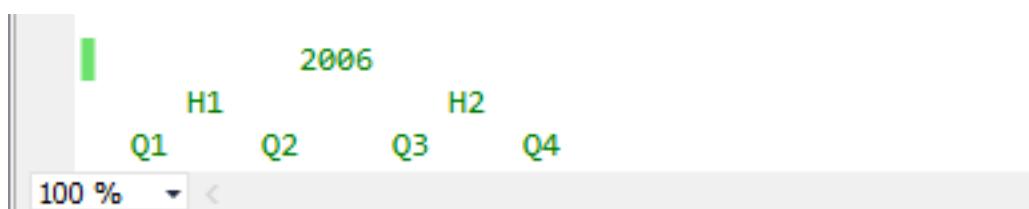
گروهها: SQL Server 2012, OLAP, MDX, SSAS

از دو تابع Lag و Lead برای واکشی فرزندان، در همان سطح استفاده می‌شود. به عبارت دیگر این دو تابع می‌توانند برادران قبل و بعد را پیدا کنند.

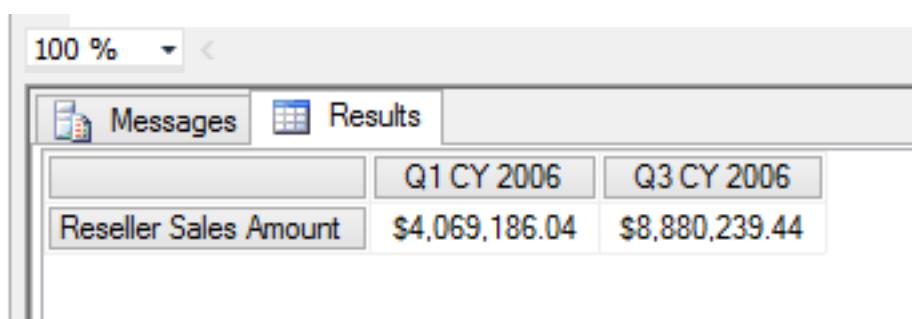
استفاده از تابع Lead():

```
Select
{
    [Date].[Calendar].[Calendar Quarter].[Q1 CY 2006],
    [Date].[Calendar].[Calendar Quarter].[Q1 CY 2006].lead(2)
} on columns,
[Measures].[Reseller Sales Amount] on rows
From [Adventure Works]
```

این تابع، دو سطح بعد از نیم فصل اول 2006 را در خروجی می‌آورد. به عبارت دیگر نیم فصل سوم 2006 را بازگشت می‌دهد.



خروجی به صورت زیر می‌باشد.



استفاده از تابع Lag():

```
Select
{
    [Date].[Calendar].[Calendar Quarter].[Q1 CY 2006],
    [Date].[Calendar].[Calendar Quarter].[Q1 CY 2006].lag(-2)
} on columns,
[Measures].[Reseller Sales Amount] on rows
From [Adventure Works]
```

این تابع برعکس تابع Lead عمل می کند، ولی عدد منفی، کارآیی آن را مشابه تابع Lead می کند . به عبارت دیگر این تابع با عدد مثبت، در همان سطح ، فرزندان قبلی را پیدا کرده و با عدد منفی، فرزندان بعدی را پیدا می کند.  
استفاده از تابع Lead با پارامتر منفی مشابه تابع Lag با پارامتر مثبت می باشد.

```
Select
{
    [Date].[Calendar].[Calendar Quarter].[Q3 CY 2006].lead(-2),
    [Date].[Calendar].[Calendar Quarter].[Q3 CY 2006]
}on columns,
[Measures].[Reseller Sales Amount]on rows
From [Adventure Works]
```

یا

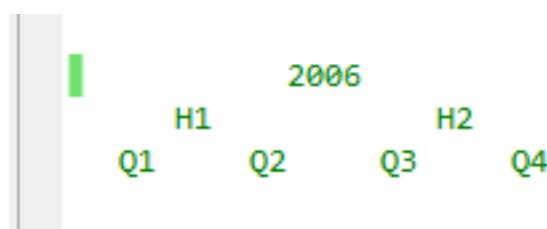
```
Select
{
    [Date].[Calendar].[Calendar Quarter].[Q3 CY 2006].lag(2),
    [Date].[Calendar].[Calendar Quarter].[Q3 CY 2006]
}on columns,
[Measures].[Reseller Sales Amount] on rows
From [Adventure Works]
```

	Q1 CY 2006	Q3 CY 2006
Reseller Sales Amount	\$4,069,186.04	\$8,880,239.44

در کوئری زیر یک حالت ترکیبی بین عملگر : و تابع حرکت در فرزندان در سطحی مشابه را مشاهده می کنیم.

```
Select
[Date].[Calendar].[Calendar Quarter].[Q1 CY 2006]
:
[Date].[Calendar].[Calendar Quarter].[Q1 CY 2006].lead(2)
on columns,
[Measures].[Reseller Sales Amount] on rows
From [Adventure Works]
```

این نمودار به درک بهتر این مثال کمک خواهد کرد



خروجی به شکل زیر می‌باشد.

	Q1 CY 2006	Q2 CY 2006	Q3 CY 2006
Reseller Sales Amount	\$4,069,186.04	\$4,153,820.42	\$8,880,239.44

در قسمتهای بعدی در خصوص توابع Head و Filter توضیح خواهیم داد.

در ادامه به بررسی توابع Head , Filter , TopCount و tail می‌پردازیم

```

Select
{
[Measures].[Internet Sales Amount],
[Measures].[Internet Tax Amount]
} on columns,
head(
[Customer].[Customer Geography].[Country],
2
)on rows
From [Adventure Works]

```

تابع Head, تعداد مشخص شده بر اساس پارامتر اول از آن محور را بر اساس نحوه نمایش تنظیم شده در SSAS, واکشی می‌کند.

	Internet Sales Amount	Internet Tax Amount
Australia	\$9,061,000.58	\$724,880.07
Canada	\$1,977,844.86	\$158,227.59

حال تصور کنید بخواهیم شرط زیر را بر روی کوئری بالا اجرا کنیم

```
( [Measures].[Internet Sales Amount] >= '2500000' )
```

به عبارت دیگر ما می‌خواهیم دو کشوری را انتخاب کنیم که میزان فروش اینترنتی آنها بالای 2500000 باشد.

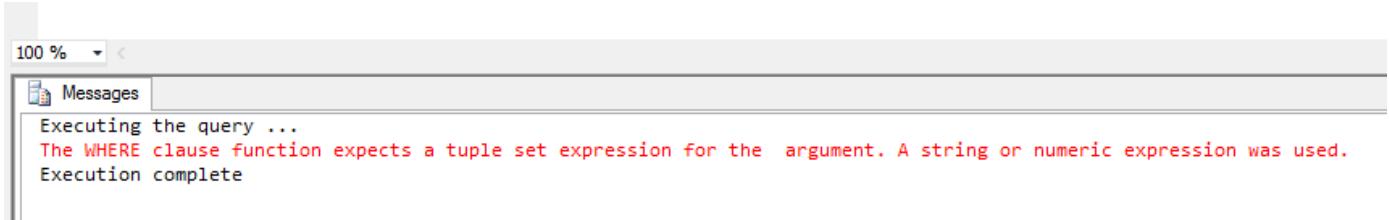
کوئری مشابه زیر می‌باشد

```

Select {
[Measures].[Internet Sales Amount],
[Measures].[Internet Tax Amount]
} on columns,
head(
[Customer].[Customer Geography].[Country],
2
)on rows
From [Adventure Works]
Where
( [Measures].[Internet Sales Amount] >= '2500000' )

```

البته خطای زیر را خواهیم داشت.



به یاد داشته باشیم در صورتیکه بخواهیم ایجاد محدودیت در نمایش داده‌ها را در یک محور داشته باشیم، باید از تابع Filter استفاده کنیم؛ به صورت زیر:

```
Select
Filter(
{
[Measures].[Internet Sales Amount],
[Measures].[Internet Tax Amount]
},
[Measures].[Internet Sales Amount] >= 2644017.71
) on columns,
head(
[Customer].[Customer Geography].[Country],
3
)on rows
From [Adventure Works]
```

تابع Filter دو پارامتر می‌گیرد. پارامتر اول نام ردیف یا ستونی می‌باشد که روی آن می‌خواهیم عمل فیلتر را انجام دهیم. پارامتر دوم شرط فیلترینگ می‌باشد که می‌بایست مانند T/SQL دارای یک خروجی Boolean باشد

	Internet Sales Amount	Internet Tax Amount
Australia	\$9,061,000.58	\$724,880.07
Canada	\$1,977,844.86	\$158,227.59
France	\$2,644,017.71	\$211,521.42

همچنان نتیجه درست نمی‌باشد ! چرا؟

اگر بخواهیم شرط روی Axis ردیف (کشورها) اعمال گردد، باید عملیات فیلترینگ در این Axis انجام شود . بنابر این خروجی بدست آمده صحیح نمی‌باشد زیرا ما عملیات فیلترینگ را روی ستون‌ها انجام داده ایم.

کوئری زیر را اجرا نمایید

```
Select {
[Measures].[Internet Sales Amount]
,[Measures].[Internet Tax Amount]
}
on columns,
head(
Filter(
```

```
[Customer].[Customer Geography].[Country],
[Measures].[Internet Sales Amount] >= 2644017.71
),
3)
on rows
From [Adventure Works]
```

البته توجه کنید که این کوئری، سه کشور اول که در شرط زیر قرار دارند را بر می گرداند و الزاماً این سه کشور از تمام کشور های دیگر بیشتر نمی باشند .

	Internet Sales Amount	Internet Tax Amount
Australia	\$9,061,000.58	\$724,880.07
France	\$2,644,017.71	\$211,521.42
Germany	\$2,894,312.34	\$231,544.99

در این حالت سه کشور که بالاتر از مقدار ذکر شده، فروش اینترنتی دارند، در خروجی قرار می گیرند . البته این سه کشور دارای بالاترین فروش نمی باشند بلکه به ترتیب اسم، از بالا گزینش انجام شده است و بعد از پیدا کردن سه کشور که در شرط قرار بگیرند، جستجو تمام شده است .

اگر بخواهیم سه کشوری را که بالاترین میزان فروش را دارند پیدا کنیم و شرط هم همواره اعمال گردد، کوئری زیر درست می باشد:

```
Select {
[Measures].[Internet Sales Amount]
,[Measures].[Internet Tax Amount]
}
on columns,
TopCount(
Filter(
[Customer].[Customer Geography].[Country],
[Measures].[Internet Sales Amount] >= 2644017.71
),
3, [Measures].[Internet Sales Amount])
on rows
From [Adventure Works]
```

	Internet Sales Amount	Internet Tax Amount
United States	\$9,389,789.51	\$751,183.18
Australia	\$9,061,000.58	\$724,880.07
United Kingdom	\$3,391,712.21	\$271,336.98

در این حالت به جای تابع Head از تابع TopCount استفاده گردیده است . این تابع سه کشوری را که بیشترین فروش اینترنتی را داشته اند و این فروش بالاتر از مقدار ذکر شده در شرط می باشد را بر می گرداند . البته در اینجا تابع topcount دارای سه پارامتر می باشد و در پارامتر سوم اعلام میکند که تعداد بالای مجموعه براساس چه شاخصی باید به دست بیاید .

حال اگر بخواهیم سه ردیف انتهایی جدول را واکسی کنیم داریم:

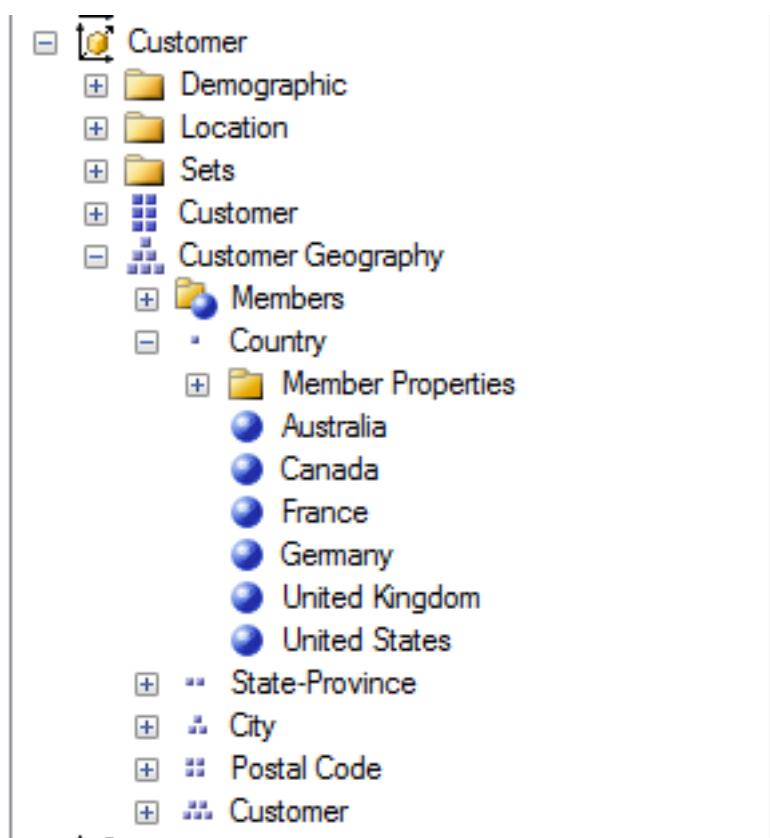
```
Select
{
[Measures].[Internet Sales Amount],
[Measures].[Internet Tax Amount]
}on columns,
tail([Customer].[Customer Geography].[Country],
3)on rows
From [Adventure Works]
```

	Internet Sales Amount	Internet Tax Amount
Germany	\$2,894,312.34	\$231,544.99
United Kingdom	\$3,391,712.21	\$271,336.98
United States	\$9,389,789.51	\$751,183.18

این تابع برعکس تابع Head کار میکند و N ردیف آخر مجموعه را بدست می آورد . البته در بالا فقط 3 ردیف انتهایی را در خروجی آورده ایم و هیچ شرطی اعمال نگردیده است.

کار با توابع nextmember و prevmember

قبل از اجرای کوئری های زیر در ابتدا به ساختار سلسله مراتبی Customer دقت نمایید و ترتیب کشورها را در این ساختار بررسی نمایید.



کوئری زیر را اجرا نمایید :

```

Select
{
[Measures].[Internet Sales Amount],
[Measures].[Internet Tax Amount]
}on columns,
{
[Customer].[Customer Geography].[Country].[Germany],
[Customer].[Customer Geography].[Country].[Germany].prevmember
}on rows
From [Adventure Works]
  
```

این تابع برادر قبلی را بدست می آورد

	Internet Sales Amount	Internet Tax Amount
Germany	\$2,894,312.34	\$231,544.99
France	\$2,644,017.71	\$211,521.42

حال کوئری زیر را اجرا نمایید :

```
Select
{
[Measures].[Internet Sales Amount],
[Measures].[Internet Tax Amount]
} on columns,
{
[Customer].[Customer Geography].[Country].[Germany],
[Customer].[Customer Geography].[Country].[Germany].lag(1)
} on rows
From [Adventure Works]
```

	Internet Sales Amount	Internet Tax Amount
Germany	\$2,894,312.34	\$231,544.99
France	\$2,644,017.71	\$211,521.42

در اینجا میزان فروش اینترنتی و همچنین میزان مالیات اینترنتی برای مشتریان آلمان و کشور قبلی بدست می‌آید.

با این تابع می‌توان برادر قبلی را با اعلام یک فاصله بدست آورد. مثلا 4 برادر قبلی یا .... را توسط این تابع بدست آورد .

کوئری زیر را اجرا نمایید :

```
Select
{
[Measures].[Internet Sales Amount],
[Measures].[Internet Tax Amount]
} on columns,
{
[Customer].[Customer Geography].[Country].[Germany],
[Customer].[Customer Geography].[Country].[Germany].prevmember.prevmember
} on rows
From [Adventure Works]
```

	Internet Sales Amount	Internet Tax Amount
Germany	\$2,894,312.34	\$231,544.99
Canada	\$1,977,844.86	\$158,227.59

در این حالت مشابه تابع Lag(2) عمل کرده ایم. حال با استفاده از تابع Nextmember می توانیم برادر بعدی را بدست بیاوریم.

کوئری زیر را اجرا کنید :

```
select
{
[Measures].[Internet Sales Amount],
[Measures].[Internet Tax Amount]
} on columns,
{
[Customer].[Customer Geography].[Country].[Germany],
[Customer].[Customer Geography].[Country].[Germany].nextmember
} on rows
From [Adventure Works]
```

	Internet Sales Amount	Internet Tax Amount
Germany	\$2,894,312.34	\$231,544.99
United Kingdom	\$3,391,712.21	\$271,336.98

و همچنین در کوئری زیر برادر بعد از برادر بعدی را بدست آورده ایم

```
Select
{
[Measures].[Internet Sales Amount],
[Measures].[Internet Tax Amount]
} on columns,
{
[Customer].[Customer Geography].[Country].[Germany],
[Customer].[Customer Geography].[Country].[Germany].nextmember.nextmember
} on rows
From [Adventure Works]
```

	Internet Sales Amount	Internet Tax Amount
Germany	\$2,894,312.34	\$231,544.99
United States	\$9,389,789.51	\$751,183.18

یک ترکیب از عملگر Range , NextMember در کوئری زیر نوشته شده است.

```
Select
{
[Measures].[Internet Sales Amount],
[Measures].[Internet Tax Amount]
} on columns,
[Customer].[Customer Geography].[Country].[Germany]
:
[Customer].[Customer Geography].[Country].[Germany].nextmember.nextmember
on rows
From [Adventure Works]
```

	Internet Sales Amount	Internet Tax Amount
Germany	\$2,894,312.34	\$231,544.99
United Kingdom	\$3,391,712.21	\$271,336.98
United States	\$9,389,789.51	\$751,183.18

کاربرد تابع Lead برای به دست آوردن برادر بعدی بر اساس عددی می باشد که به آن داده ایم .

در اینجا ترکیبی از رنج و به دست آوردن برادر برادر بعدی را داریم .

```
Select
{
[Measures].[Internet Sales Amount],
[Measures].[Internet Tax Amount]
} on columns,
[Customer].[Customer Geography].[Country].[Germany]
:
[Customer].[Customer Geography].[Country].[Germany].lead(2)
on rows
From [Adventure Works]
```

The screenshot shows the SSMS Results grid with three columns: Internet Sales Amount and Internet Tax Amount. The rows represent countries: Germany, United Kingdom, and United States. The data is as follows:

	Internet Sales Amount	Internet Tax Amount
Germany	\$2,894,312.34	\$231,544.99
United Kingdom	\$3,391,712.21	\$271,336.98
United States	\$9,389,789.51	\$751,183.18

در قسمت بعدی تابع Order را بررسی می‌کنیم.

در زیر در ابتدای کوئری می‌نویسیم و سپس بررسی می‌کنیم که چگونه می‌توان خروجی را مرتب کنیم.

```
Select
[Measures].[Internet Sales Amount] on columns,
[Product].[Product Categories].[Subcategory] on rows
From [Adventure Works]
```

کوئری بالا میزان فروش اینترنتی را بر اساس دسته بندی محصولات واکنشی می‌کند

	Internet Sales Amount
Bike Racks	\$39,360.00
Bike Stands	\$39,591.00
Bottles and Cages	\$56,798.19
Cleaners	\$7,218.60
Fenders	\$46,619.58
Helmets	\$225,335.60
Hydration Packs	\$40,307.67
Lights	(null)
Locks	(null)
Panniers	(null)
Pumps	(null)

در کوئری زیر لیست فروش، براساس میزان فروش اینترنتی مرتب شده است :

```
Select
[Measures].[Internet Sales Amount] on columns,
order(
[Product].[Subcategory].[Subcategory],
[Measures].[Internet Sales Amount]
) on rows
From [Adventure Works]
```

The screenshot shows the SSMS Results grid with the following data:

	Internet Sales Amount
Pedals	(null)
Pumps	(null)
Road Frames	(null)
Saddles	(null)
Tights	(null)
Touring Frames	(null)
Wheels	(null)
Socks	\$5,106.32
Cleaners	\$7,218.60
Caps	\$19,688.10
Gloves	\$35,020.70
Vests	\$35,687.00
Bike Racks	\$39,360.00
Bike Stands	\$39,591.00
Hydration Packs	\$40,307.67
Fenders	\$46,619.58
Bottles and Cages	\$56,798.19
Shorts	\$71,319.81
Jerseys	\$172,950.68
Helmets	\$225,335.60

تابع Order، ستون یا ردیف خاصی را بر اساس ستون یا ردیف دیگری مرتب می کند . این تابع بر روی کل یک Axis اعمال میگردد و دو پارامتر دارد .

پارامتر دوم مشخص می کند مرتب سازی پارامتر اول بر اساس چه شاخصی باشد .

همچنین در کوئری زیر نحوهی مرتب کردن به صورت نزولی را خواهیم دید.

```
Select
[Measures].[Internet Sales Amount] on columns,
order(
[Product].[Subcategory].[Subcategory],
[Measures].[Internet Sales Amount],
desc
) on rows
From [Adventure Works]
```

The screenshot shows the SSMS interface with the 'Results' tab selected. The results are displayed in a grid with two columns: 'Subcategory' and 'Internet Sales Amount'. The data is sorted in descending order of sales amount.

	Internet Sales Amount
Road Bikes	\$14,520,584.04
Mountain Bikes	\$9,952,759.56
Touring Bikes	\$3,844,801.05
Tires and Tubes	\$245,529.32
Helmets	\$225,335.60
Jerseys	\$172,950.68
Shorts	\$71,319.81
Bottles and Cages	\$56,798.19
Fenders	\$46,619.58
Hydration Packs	\$40,307.67
Bike Stands	\$39,591.00
Bike Racks	\$39,360.00
Vests	\$35,687.00
Gloves	\$35,020.70
Caps	\$19,688.10
Cleaners	\$7,218.60
Socks	\$5,106.32
Bib-Shorts	(null)
Bottom Brackets	(null)

مرتب سازی به صورت صعودی به شکل زیر می باشد

```
Select
[Measures].[Internet Sales Amount] on columns,
order(
[Product].[Subcategory].[Subcategory],
[Measures].[Internet Sales Amount],
asc
) on rows
From [Adventure Works]
```

Mountain Frames	(null)
Panniers	(null)
Pedals	(null)
Pumps	(null)
Road Frames	(null)
Saddles	(null)
Tights	(null)
Touring Frames	(null)
Wheels	(null)
Socks	\$5,106.32
Cleaners	\$7,218.60
Caps	\$19,688.10
Gloves	\$35,020.70

از آنجایی که مرتب سازی به صورت پیش فرض صعودی می باشد، ب کار بردن این کلید واژه ، خیلی لازم نمی باشد .

رویکرد تابع Order با ساختار سلسله مراتبی

برای شروع کوئری زیر را اجرا می کنیم.

```
Select
[Measures].[Internet Sales Amount] on columns,
order(
[Product].[Product Categories].[Subcategory],
[Measures].[Internet Sales Amount],
asc
) on rows
From [Adventure Works]
```

Bib-Shorts	(null)
Tights	(null)
Socks	\$5,106.32
Caps	\$19,688.10
Gloves	\$35,020.70
Vests	\$35,687.00
Shorts	\$71,319.81
Jerseys	\$172,950.68
Lights	(null)
Locks	(null)
Panniers	(null)
Pumps	(null)
Cleaners	\$7,218.60
Bike Racks	\$39,360.00
Bike Stands	\$39,591.00
Hydration Packs	\$40,307.67
Fenders	\$46,619.58

البته در این حالت با نگاه کردن به خروجی متوجه میشوید که عملیات مرتب سازی به درستی انجام نشده است و دلیل این اتفاق رویکرد تابع مرتب سازی به ساختار سلسله مراتبی میباشد . عملاً تابع مرتب سازی احترام ویژه ای به ساختار سلسله می گزارد .

حال همان کوئری را به صورت زیر اجرا می کنیم :

```
Select
[Measures].[Internet Sales Amount] on columns,
order(
[Product].[Product Categories].[Subcategory],
[Measures].[Internet Sales Amount],
base
) on rows
From [Adventure Works]
```

Bib-Shorts	(null)
Tights	(null)
Bottom Brackets	(null)
Brakes	(null)
Chains	(null)
Cranksets	(null)
Derailleurs	(null)
Forks	(null)
Handlebars	(null)
Headsets	(null)
Mountain Frames	(null)
Pedals	(null)
Road Frames	(null)
Saddles	(null)
Touring Frames	(null)
Wheels	(null)
Socks	\$5,106.32
Cleaners	\$7,218.60
Caps	\$19,688.10

حرف b در کلمه بی کلیدی `basc` به معنی شکستن ساختار سلسله مرتبی می باشد . در این حالت عملیات مرتب سازی بدون درنظر گرفتن ساختار سلسله مرتبی انجام می گردد .

کوئری های زیر تاثیر مرتب سازی نزولی و ساختار سلسله مرتبی را بررسی می کنند

```
Select
[Measures].[Internet Sales Amount] on columns,
order(
[Product].[Product Categories].[Subcategory],
[Measures].[Internet Sales Amount],
desc
) on rows
From [Adventure Works]
```

	Internet Sales Amount
Road Bikes	\$14,520,584.04
Mountain Bikes	\$9,952,759.56
Touring Bikes	\$3,844,801.05
Tires and Tubes	\$245,529.32
Helmets	\$225,335.60
Bottles and Cages	\$56,798.19
Fenders	\$46,619.58
Hydration Packs	\$40,307.67
Bike Stands	\$39,591.00
Bike Racks	\$39,360.00
Cleaners	\$7,218.60
Lights	(null)
Locks	(null)
Panniers	(null)
Pumps	(null)
Jerseys	\$172,950.68
Shorts	\$71,319.81
Vests	\$35,687.00
Gloves	\$35,020.70

در این حالت مرتب سازی نزولی انجام می گردد اما با درنظر گرفتن ساختار سلسله مراتبی

```
Select
[Measures].[Internet Sales Amount] on columns,
order(
[Product].[Product Categories].[Subcategory],
[Measures].[Internet Sales Amount],
bdesc
) on rows
From [Adventure Works]
```

The screenshot shows a Microsoft Analysis Services query results window. The title bar says "100 %". Below it are two tabs: "Messages" (selected) and "Results". The "Results" tab displays a table with two columns: "Product Categories" and "Internet Sales Amount". The data is as follows:

	Internet Sales Amount
Road Bikes	\$14,520,584.04
Mountain Bikes	\$9,952,759.56
Touring Bikes	\$3,844,801.05
Tires and Tubes	\$245,529.32
Helmets	\$225,335.60
Jerseys	\$172,950.68
Shorts	\$71,319.81
Bottles and Cages	\$56,798.19
Fenders	\$46,619.58
Hydration Packs	\$40,307.67
Bike Stands	\$39,591.00
Bike Racks	\$39,360.00
Vests	\$35,687.00
Gloves	\$35,020.70
Caps	\$19,688.10
Cleaners	\$7,218.60
Socks	\$5,106.32
Lights	(null)

در این حالت عمل مرتب سازی نزولی بدون درنظر گرفتن ساختار سلسله مراتبی انجام میگردد.

می توان مرتب سازی را بر اساس یک ستون که در کوئری ذکر نشده است انجام داد. در این حالت نتیجه در ظاهر مرتب نمی باشد.

```
Select
[Measures].[Internet Sales Amount] on columns,
order(
[Product].[Product Categories].[Subcategory],
[Measures].[Reseller Sales Amount],
bdesc
) on rows
From [Adventure Works]
```

The screenshot shows a Microsoft Analysis Services query results window. The title bar says "100 %". The window has two tabs: "Messages" (selected) and "Results". The "Results" tab displays a table with two columns: "Product Category" and "Internet Sales Amount". The data is as follows:

	Internet Sales Amount
Road Bikes	\$14,520,584.04
Mountain Bikes	\$9,952,759.56
Touring Bikes	\$3,844,801.05
Mountain Frames	(null)
Road Frames	(null)
Touring Frames	(null)
Wheels	(null)
Jerseys	\$172,950.68
Shorts	\$71,319.81
Helmets	\$225,335.60
Vests	\$35,687.00
Gloves	\$35,020.70
Cranksets	(null)
Tights	(null)
Bike Racks	\$39,360.00
Handlebars	(null)

در این کوئری دو ستون وجود دارد اما مرتب سازی براساس یکی از ستون ها انجام می شود

```
Select
{
[Measures].[Internet Sales Amount],
[Measures].[Reseller Sales Amount]
} on columns,
order(
[Product].[Product Categories].[Subcategory],
[Measures].[Reseller Sales Amount],
bdesc
) on rows
From [Adventure Works]
```

100 % <

Messages Results

	Internet Sales Amount	Reseller Sales Amount
Road Bikes	\$14,520,584.04	\$29,358,206.96
Mountain Bikes	\$9,952,759.56	\$26,492,684.38
Touring Bikes	\$3,844,801.05	\$10,451,490.22
Mountain Frames	(null)	\$4,713,672.15
Road Frames	(null)	\$3,849,853.34
Touring Frames	(null)	\$1,642,327.69
Wheels	(null)	\$679,070.07
Jerseys	\$172,950.68	\$579,308.71
Shorts	\$71,319.81	\$342,202.72
Helmets	\$225,335.60	\$258,712.93
Vests	\$35,687.00	\$223,801.37
Gloves	\$35,020.70	\$207,775.17
Cranksets	(null)	\$203,942.62
Tights	(null)	\$201,833.01
Bike Racks	\$39,360.00	\$197,736.16
Handlebars	(null)	\$170,591.32

عملیات مرتب سازی می تواند بر روی هر کدام از Axis ها جداگانه انجام گردد .

```
Select
order(
[Product].[Product Categories].[Subcategory],
[Measures].[Reseller Sales Amount],
bdesc
) on columns,
{
[Measures].[Internet Sales Amount],
[Measures].[Reseller Sales Amount]
} on rows
From [Adventure Works]
```

100 % <

Messages Results

	Road Bikes	Mountain Bikes	Touring Bikes	Mountain Frames	Road Frames	Touring Frames	Wheels	Jerseys	Shorts
Internet Sales Amount	\$14,520,584.04	\$9,952,759.56	\$3,844,801.05	(null)	(null)	(null)	(null)	\$172,950.68	\$71,319.81
Reseller Sales Amount	\$29,358,206.96	\$26,492,684.38	\$10,451,490.22	\$4,713,672.15	\$3,849,853.34	\$1,642,327.69	\$679,070.07	\$579,308.71	\$342,202.72

ازتابع hierarchize هم می توان برای عملیات مرتب سازی استفاده کرد

```
Select
[Measures].[Internet Sales Amount] on columns,
hierarchize(
```

```
{
[Product].[Subcategory].[Touring Bikes],
[Product].[Subcategory],
[Product].[Subcategory].[Mountain Bikes]
}
)on rows
From [Adventure Works]
```

The screenshot shows the SSMS interface with the 'Results' tab selected. The results are presented in a table with two columns: 'Internet Sales Amount' and 'Subcategory'. The data is as follows:

	Internet Sales Amount
All Products	\$29,358,677.22
Mountain Bikes	\$9,952,759.56
Touring Bikes	\$3,844,801.05

برای مرتب سازی ردیف هایی که عملاً ارتباطی با هم ندارند از این تابع استفاده می شود . پیش فرض مرتب سازی نزولی می باشد . البته می توانستیم همچنان از تابع Order استفاده کنیم:

```
Select
[Measures].[Internet Sales Amount] on columns,
Order(
{
[Product].[Subcategory].[Touring Bikes],
[Product].[Subcategory],
[Product].[Subcategory].[Mountain Bikes]
}, [Measures].[Internet Sales Amount] , basc
)on rows
From [Adventure Works]
```

The screenshot shows the SSMS interface with the 'Results' tab selected. The results are presented in a table with two columns: 'Internet Sales Amount' and 'Subcategory'. The data is as follows:

	Internet Sales Amount
Touring Bikes	\$3,844,801.05
Mountain Bikes	\$9,952,759.56
All Products	\$29,358,677.22

حال می خواهیم ببینیم چگونه می توان با استفاده از تابع hierarchize مرتب سازی صعودی را انجام دهیم :

```
Select
[Measures].[Internet Sales Amount] on columns,
hierarchize(
{
[Product].[Subcategory].[Touring Bikes],
[Product].[Subcategory],
[Product].[Subcategory].[Mountain Bikes]
}),
```

```
post
) on rows
From [Adventure Works]
```

	Internet Sales Amount
Mountain Bikes	\$9,952,759.56
Touring Bikes	\$3,844,801.05
All Products	\$29,358,677.22

همان طور که مشخص می‌باشد مرتب سازی به درستی صورت نگرفته است؟! دلیل این اتفاق اولویت دادن به ساختار سلسله مراتبی می‌باشد.

برای بررسی بیشتر ابتدا کوئری زیر را فراخوانی نمایید :

```
Select
[Measures].[Internet Sales Amount] on columns,
non empty order(
crossjoin(
[Product].[Category].[Category],
[Product].[Subcategory].[Subcategory]
),
[Measures].[Internet Sales Amount],
desc
) on rows
From [Adventure Works]
```

		Internet Sales Amount
Accessories	Tires and Tubes	\$245,529.32
Accessories	Helmets	\$225,335.60
Accessories	Bottles and Cages	\$56,798.19
Accessories	Fenders	\$46,619.58
Accessories	Hydration Packs	\$40,307.67
Accessories	Bike Stands	\$39,591.00
Accessories	Bike Racks	\$39,360.00
Accessories	Cleaners	\$7,218.60
Bikes	Road Bikes	\$14,520,584.04
Bikes	Mountain Bikes	\$9,952,759.56
Bikes	Touring Bikes	\$3,844,801.05
Clothing	Jerseys	\$172,950.68
Clothing	Shorts	\$71,319.81
Clothing	Vests	\$35,687.00
Clothing	Gloves	\$35,020.70
Clothing	Caps	\$19,688.10
Clothing	Socks	\$5,106.32

همانطور که مشخص می‌باشد، مرتب سازی براساس فروش اینترنتی و با درنظر گرفتن ساختار سلسله مراتبی صورت گرفته است. برای مرتب سازی بدون در نظر گرفتن ساختار سلسله مراتبی کوئری زیر را بنویسید :

```
Select
[Measures].[Internet Sales Amount] on columns,
non empty order(
crossjoin(
[Product].[Category].[Category],
[Product].[Subcategory].[Subcategory]
),
[Measures].[Internet Sales Amount],
bdesc
) on rows
From [Adventure Works]
```

		Internet Sales Amount
Bikes	Road Bikes	\$14,520,584.04
Bikes	Mountain Bikes	\$9,952,759.56
Bikes	Touring Bikes	\$3,844,801.05
Accessories	Tires and Tubes	\$245,529.32
Accessories	Helmets	\$225,335.60
Clothing	Jerseys	\$172,950.68
Clothing	Shorts	\$71,319.81
Accessories	Bottles and Cages	\$56,798.19
Accessories	Fenders	\$46,619.58
Accessories	Hydration Packs	\$40,307.67
Accessories	Bike Stands	\$39,591.00
Accessories	Bike Racks	\$39,360.00
Clothing	Vests	\$35,687.00
Clothing	Gloves	\$35,020.70
Clothing	Caps	\$19,688.10
Accessories	Cleaners	\$7,218.60
Clothing	Socks	\$5,106.32

در این حالت مرتب سازی بدون در نظر گرفتن ساختار سلسله مراتب انجام می گردد.

در دو کوئری فوق از تابع Non empty برای حذف ردیفهای Null استفاده شده است.

## مفاهیم مقدماتی : Data Warehouse

سیستم‌های مبتنی بر اهداف اصلی سازمان استفاده می‌شوند و این سیستم‌ها کار پردازش و ذخیره کردن داده‌ها را در OLTP Database انجام می‌دهند. مانند تمامی سیستم‌های ...، ERP, MIS

OLTP : پایگاه داده‌ی سیستم‌های OLTP می‌باشد. به طور معمول هر تراکنش کاربر در کمترین زمان ممکن برروی این سیستم‌ها ذخیره می‌گردد و در طول روز بارها دستورات ( Insert/Update/Delete ) برروی آنها انجام می‌شود. این پایگاه‌های داده، همان ها Source System می‌باشند.

ETL ( extract, transform, and load ) : مراحل انتقال داده از OLTP Database به پایگاه داده‌ی Stage می‌باشد. ETL سیستمی می‌باشد که توانایی اتصال به OLTP را دارد و اطلاعات را از OLTP واکشی می‌کند و به پایگاه داده‌ی Stage انتقال می‌دهد. سپس ETL داده‌ها را مجتمع ( integrates ) کرده و از Stage به DDS ( Dimensional Data Source ) انتقال می‌دهد.

عملیات واکشی داده‌ها طبق یک سری قوانین و قواعد می‌باشد .

برای انجام عملیات ETL دو روش وجود دارد

Data .1 مجتمع ( Integrate ) و تمیز ( Data cleansing ) شود و در نهایت وارد Data Warehouse گردد.

Data .2 وارد Data Warehouse گردد سپس مراحل مجتمع سازی و پاک سازی داده‌ها بر روی داده‌ها در خود Data Warehouse انجام گردد.

Consolidates Data : برخی شرکت‌ها داده‌های اصلی خودشان را در چندین پایگاه داده دارند. در این حالت برای انجام عملیات ETL باید داده‌ها تحکیم و مجتمع شوند و سپس در Data Warehouse ذخیره شوند.

به طور کلی موارد زیر در فرایند ETL در نظر گرفته می‌شود:

1. Data availability : برخی داده‌ها در یک سیستم وجود دارند ولی در سیستم دیگری وجود ندارند و یا تفاوت در نگهداری داده‌ها در سیستم‌های مختلف داریم. مثلاً در یک سیستم آدرس در سه فیلد نگه داری می‌شود (کشور-شهر-آدرس) اما در سیستمی دیگر در دو فیلد (کشور-آدرس) نگه داری می‌شود. در این حالت باید ما در ETL راه کارهایی برای مجتمع کردن این موارد در نظر بگیریم.

2. Time ranges : در سیستم‌های مختلف امکان دارد بعدهای زمانی مختلف باشد . مثلاً در یک سیستم بررسی‌ها در بازه‌ی ساعتی و در سیستم دیگر بررسی‌ها در بازه‌ی روزانه یا ماهانه باشد . بنابر این در تجمیع داده‌ها باید این مورد مد نظر گرفته شود.

3. Definitions : تعاریف در سیستم‌های مختلف می‌توانند متفاوت باشند. مثلاً در یک سیستم، مبلغ کل فاکتور شامل مالیات می‌باشد ولی در سیستمی دیگر این مبلغ فاقد مالیات می‌باشد.

4. Conversion : در فرایند ETL باید باز از قواعد موجود در سیستم‌های مختلف آگاهی داشته باشیم. مثلاً در یک سیستم ممکن است دما را به صورت سانتیگراد و در دیگری فارنهایت نگه داری کنند.

5. Matching : باید بررسی لازم را انجام دهیم که کدام داده مرتبط با کدام سیستم می‌باشد. به عبارت دیگر کدام سیستم مالک داده می‌باشد و دقیقاً داده‌ها در کدام سیستم معتبرتر می‌باشند. مثلاً پرسنل، هم در سیستم حسابداری می‌باشند هم در سیستم پرسنلی؛ ولی معمولاً داده‌های اصلی از سیستم پرسنلی می‌آینند.

Periodically : عملیات واکشی داده‌ها (Retrieves Data) و مجتمع سازی داده‌ها (Consolidates Data) در فرآیند ETL فقط یکبار اتفاق نمی‌افتد و این مراحل در بازه‌های زمانی خاص تکرار می‌گردند. این واکشی و انتقال داده‌ها می‌تواند در روز چند بار تکرار شود یا می‌تواند چند روز یک بار اجرا گردد و این بستگی دارد به سیاست موجود در Data Warehouse.

Dimensional (DDS) (Dimensional Data Source) (Data Warehouse) : یک پایگاه داده از نوع نرم‌مال شده (Normalized) یا بعدی (Dimensional) می‌باشد. که داده‌های مجتمع شده و تمیز شده سیستم‌های OLTP را در خود جای داده است. این پایگاه داده برای واکشی‌های سیستم‌های آنالیز داده مورد استفاده قرار می‌گیرد. ورود اطلاعات در Data Warehouse به صورت Batch می‌باشد و به هیچ عنوان مانند پایگاه داده‌های OLTP ویرایش داده‌ها به صورت Online و هر زمان که داده‌ها تغییر می‌کنند، صورت نمی‌گیرد. اطلاعات در Dimensional Data Warehouse معمولاً به صورت تجمعی شده روزانه، ماهانه، فصلی یا سالانه می‌باشد. DDS‌ها مجموعه‌ای از Mart‌ها هستند. و عمدتاً به صورت denormalized Mart می‌باشند.

Dimensional Data Mart : مجموعه‌ای از جداول Fact, Dimension می‌باشند که در یک بیزینس خاص باهم در ارتباط و مشترک می‌باشند.

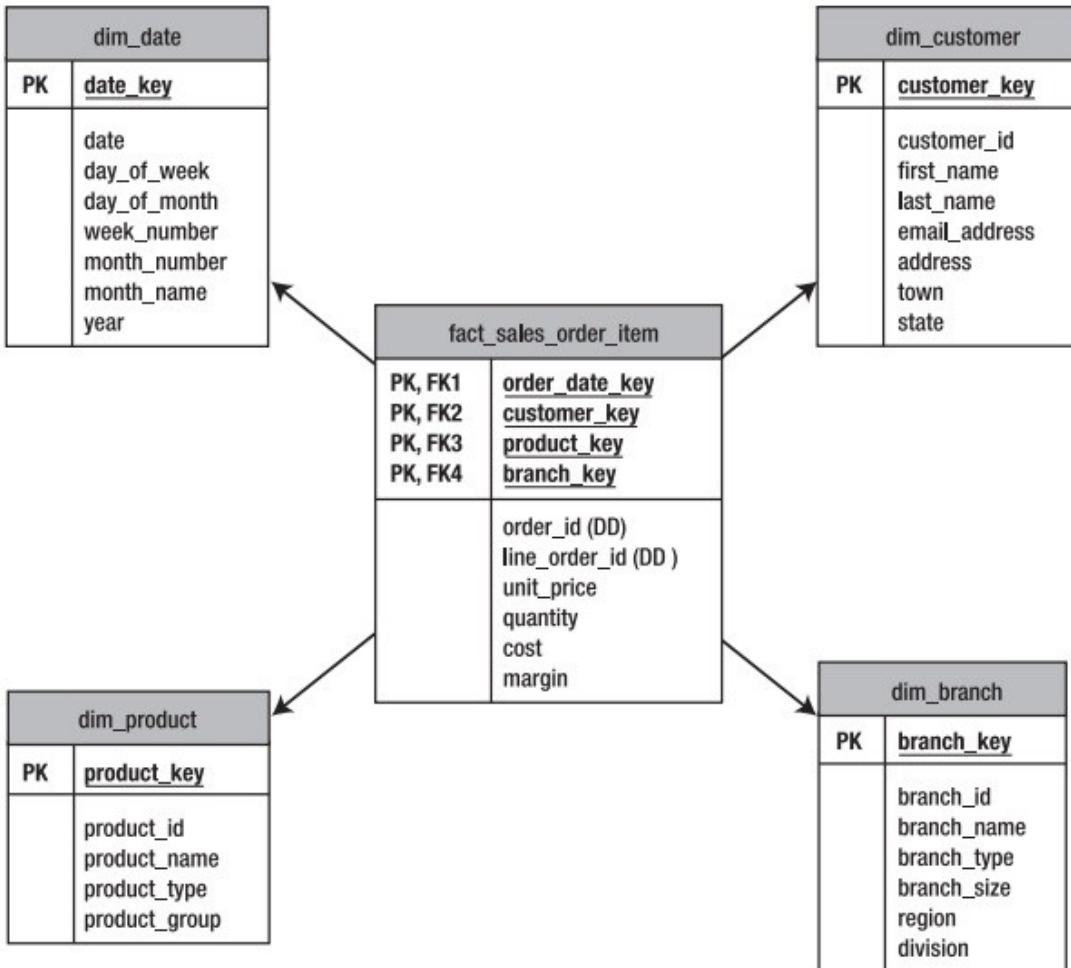
dimensional data store schemas : طراحی‌های مختلفی از جداول Fact, Dimension در DDS وجود دارد که عبارتند از

1. Star schema : ساده‌ترین روش پیاده سازی Data Warehouse

2. Snowflake : در این روش جداول Dimension کمی نرم‌مال سازی بیشتری دارند. سیستم‌های آنالیز داده با این روش بهتر کار می‌کنند.

3. Galaxy schemas : طراحی در این روش بسیار سخت و پیچیده می‌باشد. با این وجود فرآیند ETL در این طراحی ساده‌تر انجام می‌شود.

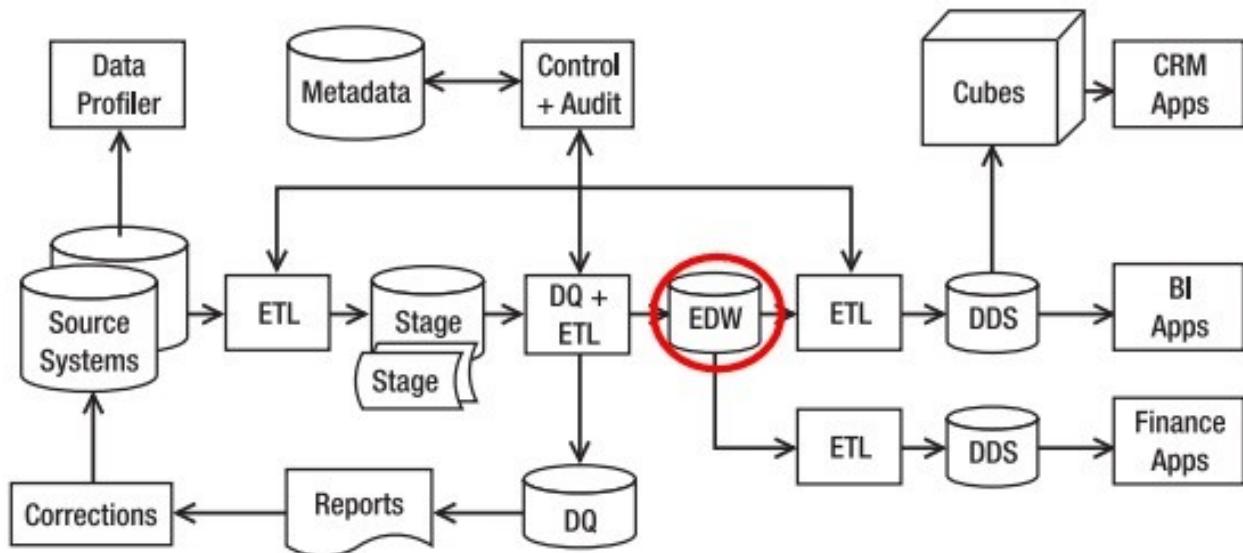
نمونه‌ی طراحی Star به صورت زیر می‌باشد :



: نقوص DDS و NDS

- در DDS ها هیچ گونه نرم‌السازی خاصی انجام نمی‌دهیم و عملای تمامی جداول را دینرمال کردیم، در حالی که در NDS تمامی جداول تا سطح سوم و گاهی تا سطح پنجم نرم‌ال شده اند.
- سرعت واکنشی و پردازش کوئری‌ها روی DDS خیلی بیشتر از NDS ها می‌باشد.
- در صورتی که نیاز باشد Data Warehouse های خیلی بزرگ طراحی کنیم با حجم بسیار زیاد توصیه می‌شود از NDS ها استفاده شود در حالی که برای Data Warehouse های کوچک و متوسط بهتر است از DDS ها استفاده شود.

تصویر طراحی یک (Enterprise Data Source = NDS) EDS در زیر آمده است :



: جداول Data Warehouse میتوانند در طول زمان بسیار بزرگ شوند و دارای تعداد رکورد زیادی گرددند. اینکه حداقل داده‌های چند سال را در Data Warehouse نگه داری کنیم بستگی به سیاست‌های سازمانی دارد که سیستم OLAP برای آن تهیه می‌گردد. استفاده کردن از table partitioning می‌تواند در جبران افزایش تعداد رکورد کمک زیادی به ما بکند.

(SCD) : سه روش برای نگه داری سابقه‌ی تغییرات در جداول Dimension وجود دارد.

1.1 SCD type 1 : هیچ گونه سابقه‌ی تغییراتی را نگه داری نمی‌کنیم

2.2 SCD type 2 : سابقه‌ی تغییرات در ردیف‌ها نگه داری می‌شود. در این روش هر ردیف، شماره ردیف قبلی را دارد و تعداد نامحدودی از تغییرات را نگه داری می‌کنیم.

3.3 SCD type 3 : سابقه‌ی تغییرات در ستون‌ها نگه داری می‌شوند و فقط ردیف جاری و آخرین تغییرات را نگه داری می‌کنیم.

Query : فقط ETL حق تغییرات در Data Warehouse را دارد و کاربر نمی‌تواند Data Warehouse را تغییر دهد. البته کاربران حق کردن از Data Warehouse Query را دارند.

دقت داشته باشید که کوئری‌های پیچیده در NDS ها بسیار کندتر از همان کوئری در DDS می‌باشد.

Business Intelligence : مجموعه‌ای از فعالیت‌ها که در یک سازمان برای شناخت بهتر وضعیت آن سازمان انجام می‌شود. نتایج BI کمک بسیاری برای تصمیم گیری‌های تکنیکی و استراتژیکی درون سازمان می‌کند. همچنین کمک به بهبود فرایندهای Business جاری می‌کند.

فعالیت‌های Business Intelligence در سه دسته بندی قرار می‌گیرند :

1. Reporting : گزارشاتی که از Data Warehouse گرفته می‌شود و به کاربر نمایش داده می‌شود و عمدها این گزارشات به صورت tabular form می‌باشند.

2. OLAP : فعالیت‌های انجام شده روی MDB برای گرفتن گزارشات Drill-Down و ... می‌باشد.

3. Data mining : فرآیند واکشی و داده کاوی داده‌های درون سیستم می‌باشد، که منجر به کشف الگوهای رفتارها و ارتباطات

داده‌ها در سیستم می‌شود. توسط داده کاوی ما متوجه می‌شویم چرا برخی داده‌ها در سیستم تولید شده اند.

زمانی که از داده کاوی برای شرح وقایع گذشته و حال استفاده می‌شود. a. descriptive analytics

زمانی که از داده کاوی برای پیش‌بینی وقایع گذشته استفاده می‌شود. b. predictive analytics

Real time data warehouse : به DW هایی گفته می‌شود که در کمترین زمان، تغییرات OLTP را در خود خواهد داشت. امروزه این نوع DW ها تغییرات 5 دقیقه تا حداقل 1 ساعت قبل را در خود دارند. برای دسترسی به چنین DW هایی دو راه زیر وجود دارد :

1. بر روی هر جدول، Trigger هایی باشد تا تغییرات را به DW منتقل دهد. (البته برای این منظور باید Business ETL مربوط به در این تریگرها نوشت)

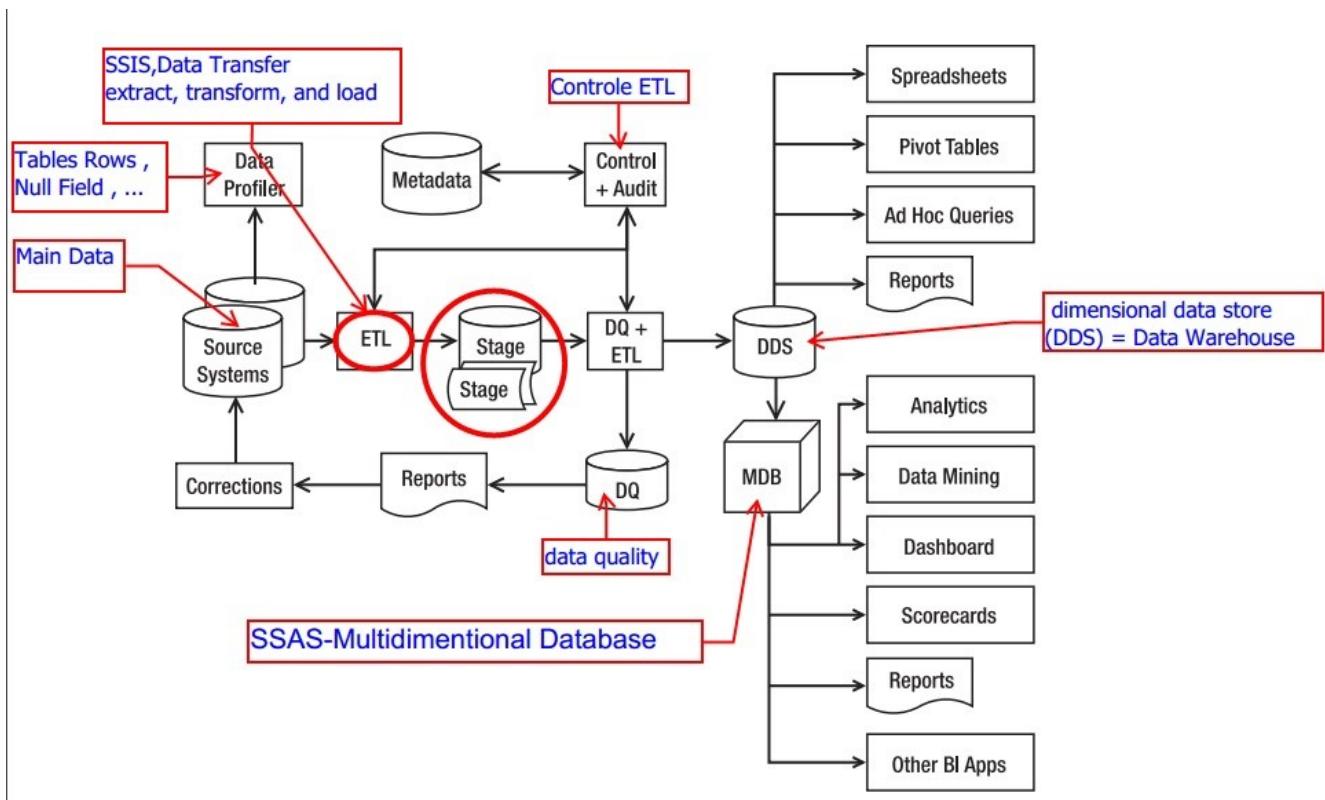
2. سورس برنامه‌های اصلی کاربر (OLTP) تغییر کند تا علاوه بر Data Warehouse ها OLTP Database را هم تغییر دهند.

روش‌های فوق بسیار روی سرعت و کارایی برنامه‌های اصلی تاثیر خواهند گذاشت.

Normalize Dimensional (NDS) : در صورتی که طراحی Data Warehouse به صورت Dimensional نباشد و به صورت NDS (Normalize Data Source) باشد، نوع NDS از نوع Data Warehouse می‌باشد.

: MDB ساخت

OLTP Database -> ETL -> Stage Database -> DDS (Dimensional Data Source = Data Warehouse) -> SSAS -> MDB





منظور از **Source System** همان OLTP Database ها می‌باشد.

به خاطر داشته باشید که **Source System** ها جزئی از **Data Warehouse** نمی‌باشند.

از کاربردهای **Data Warehouse** می‌توان به موارد زیر اشاره کرد

1. Data Mining .

2. استفاده در گزارشات

3. تجمعی داده ها

کمک به درک بهتر **Business** جاری در سازمان می‌کند. همچنین منجر به کشف دانش از درون داده‌ها می‌شود.

برای **Data Mining** می‌توانید از انواع پایگاه داده‌های موجود مانند رابطه‌ای ، سلسله مراتبی و چند بعدی استفاده کرد . حتا می‌توان از فایل‌های Excel ، XML نیز استفاده کرد.

: (Customer Relationship Management (CRM

منظور از مشتری، مصرف کننده‌ی سرویسی است که سازمان شما ارایه می‌کند. یک سیستم CRM شامل تمامی برنامه‌ای می‌باشد که تمام فعالیت‌های مشتری را پشتیبانی می‌کند.

: (Operational Data Store (ODS

این پایگاه داده به صورت رابطه‌ای و نرم‌ال شده می‌باشد و شامل تمامی اطلاعات پایگاه داده ای OLTP می‌باشد که در این پایگاه داده مجمع شده اند. تفاوت ODS با **Data Warehouse** در این می‌باشد که داده‌ها در ODS با هر Transaction به روز می‌شوند (سرعت بروز رسانی اطلاعات در ODS بالاتر از DW می‌باشد).

: (Master Data Management (MDM

در یک نگاه می‌توان داده‌ها را به دو دسته تقسیم کرد

transaction data .1

master data .2

: شامل داده ای transactional در سیستم‌های OLTP می‌باشد.

: توضیح دهنده Business جاری در سازمان می‌باشد.

برای تشخیص این دو نیاز است Business سازمان را به خوبی شناسایی نمایید. به عبارت دیگر رویدادهای Business ی همان master data می‌باشند و شامل پاسخ‌های این سوال‌ها می‌باشد. چه کسی، چه چیزی و کجا در مورد Business transaction data . transaction

(Customer data integration (CDI) : عبارت است از MDM در رابطه با مشتری داده‌ها. کار این قسمت عبارت است از واکشی، پاک سازی ، ذخیره سازی ، نگه داری و به اشتراک گذاشتن داده ای مشتری می‌باشد.

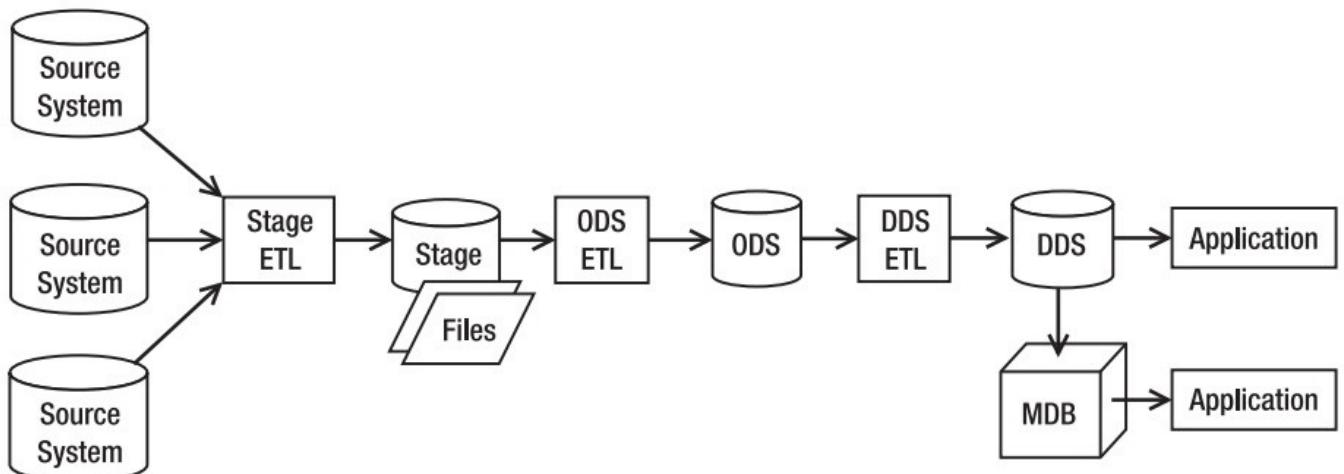
داده ای ذخیره شده در پایگاه داده ، داده‌هایی مانند عکس و فیلم و صوت و Unstructured Data

...

Service-Oriented Architecture (SOA) : یک متد ساخت برنامه می‌باشد که در این روش تمامی اجزا برنامه به صورت ماثولهایی دیده می‌شود که در آنها ارتباطات با دیگر سیستم‌ها به صورت سرویس می‌باشد و این زیر سیستم‌ها را می‌توان در پروژه‌های مختلف به کار برد.

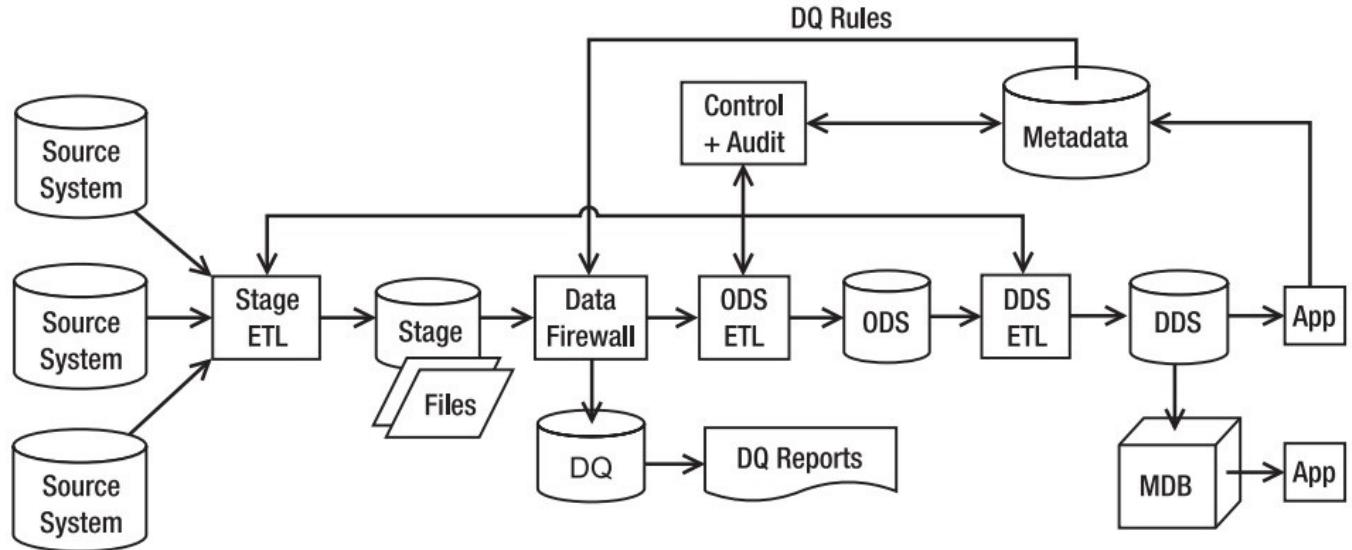
Real-Time Data Warehouse : DW را روی Transaction OLTP می‌شوند در هنگامی که یک ETL توسط روش می‌افتد.

مراحل انتقال داده از OLTP Database به صورت زیر می‌باشد.



Data quality : مکانیسم اطمینان بخشی از این که در DW داده‌ای مناسب و درست وارد می‌شوند. به عبارت دیگر DQ همان firewall برای DW در مقابل داده‌های نامناسب می‌باشد.

برای بهتر مشخص شدن مکان DQ شکل زیر را در نظر بگیرید



نحوهی حرکت داده ای از OLTP به OLAP اولین چیزی میباشد که شما باید به آن فکر کنید و برای آن روشی را انتخاب نمایید قبل از ساخت Data Warehouse.

چهار روش برای معماری انتقال اطلاعات از OLTP به DW وجود دارد (البته به عنوان نمونه و شما میتوانید از روش‌های دیگر و طراحی‌های مختلف و ترکیبی نیز بهره ببرید)

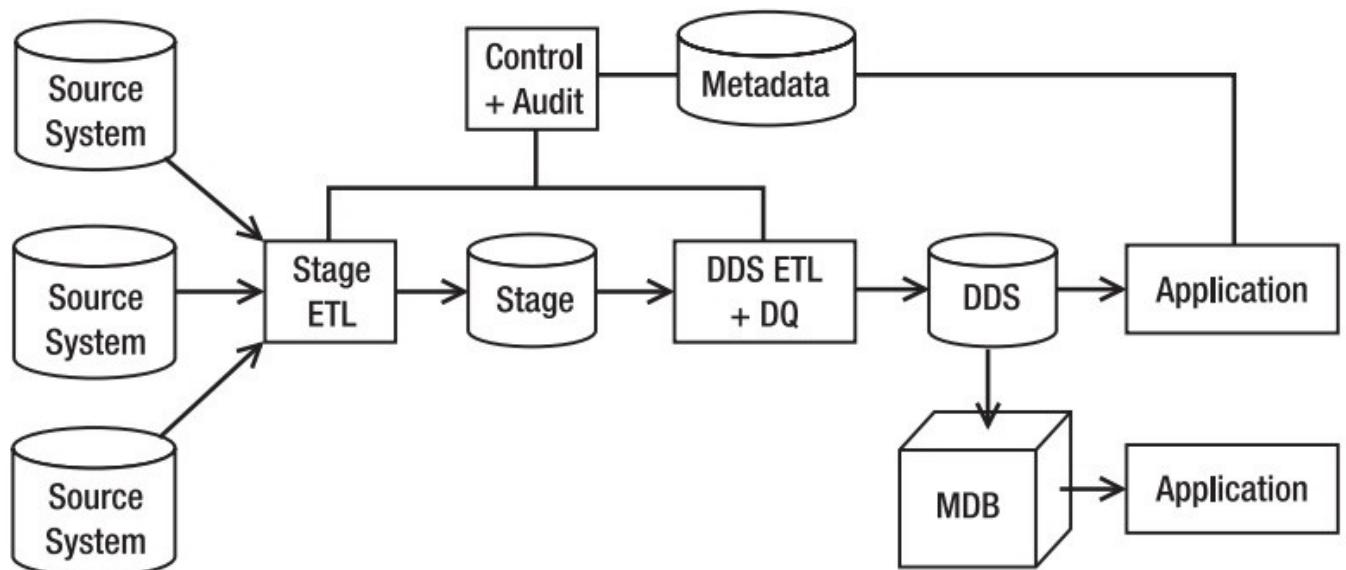
1. single DDS : در این روش فقط Stage , DDS وجود دارد.

2. NDS + DDS : در این روش علاوه بر Stage,DDS از NDS نیز استفاده میشود.

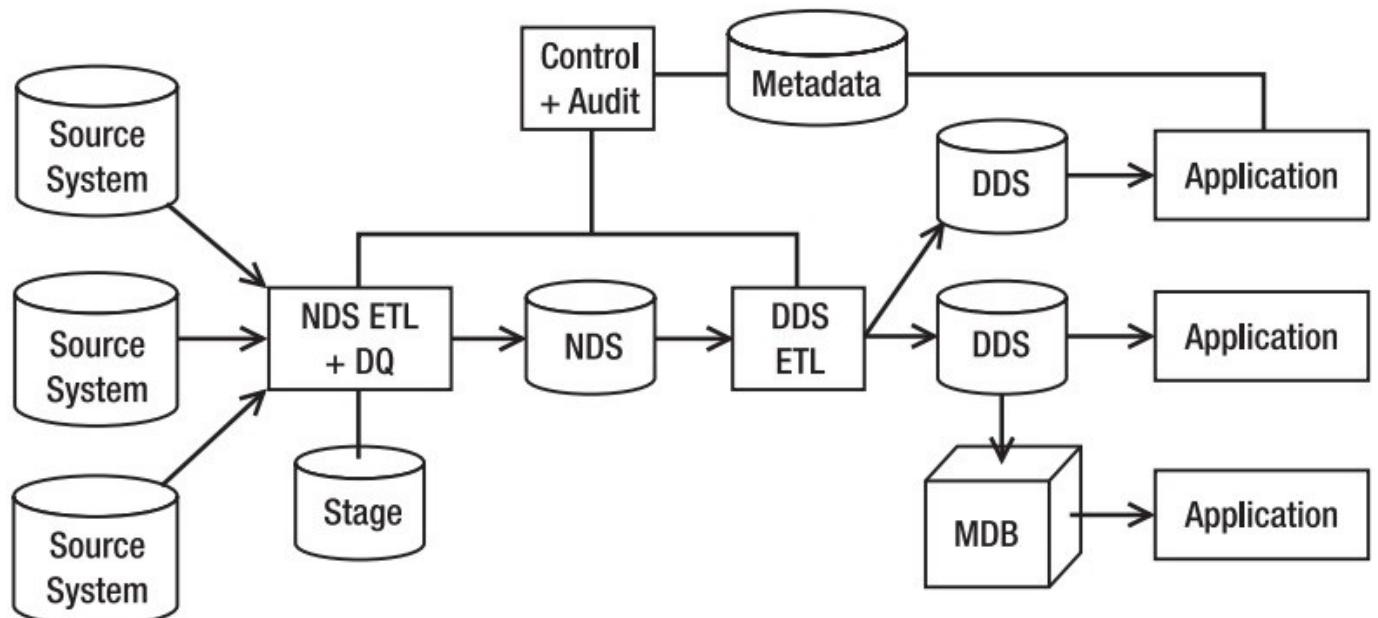
3. ODS + DDS : در این روش از Stage,ODS,DDS استفاده میگردد.

4. federated data warehouse (FDW) : استفاده از چندین DW که با هم تجمعی شده اند.

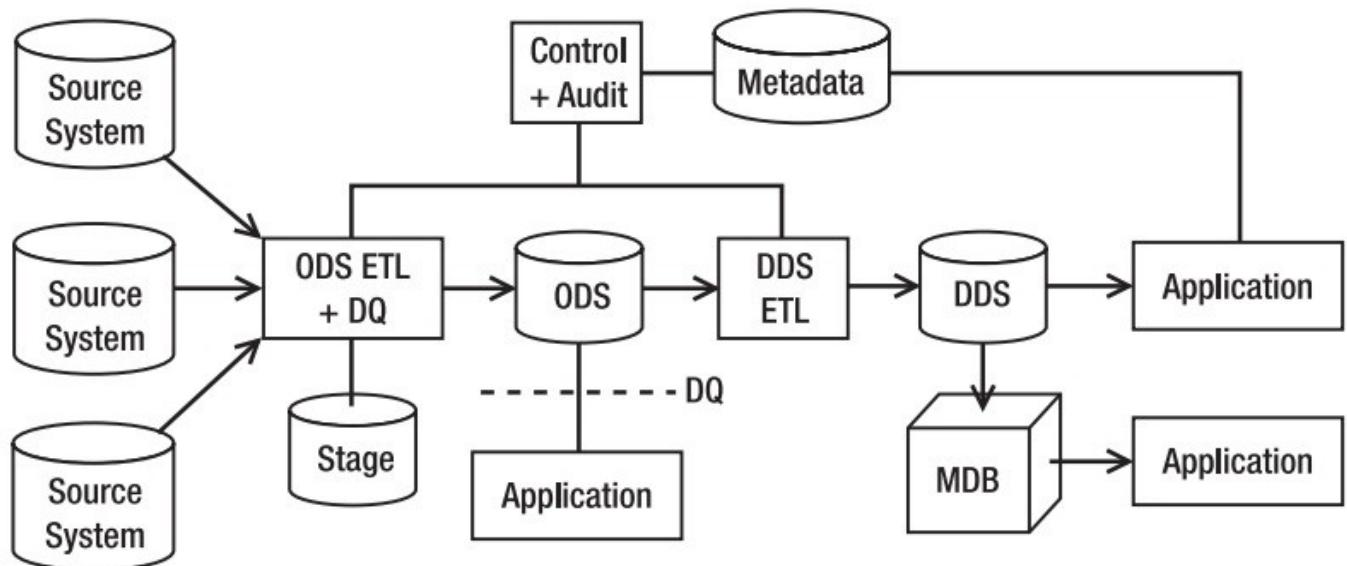
تصویر : Single DDS



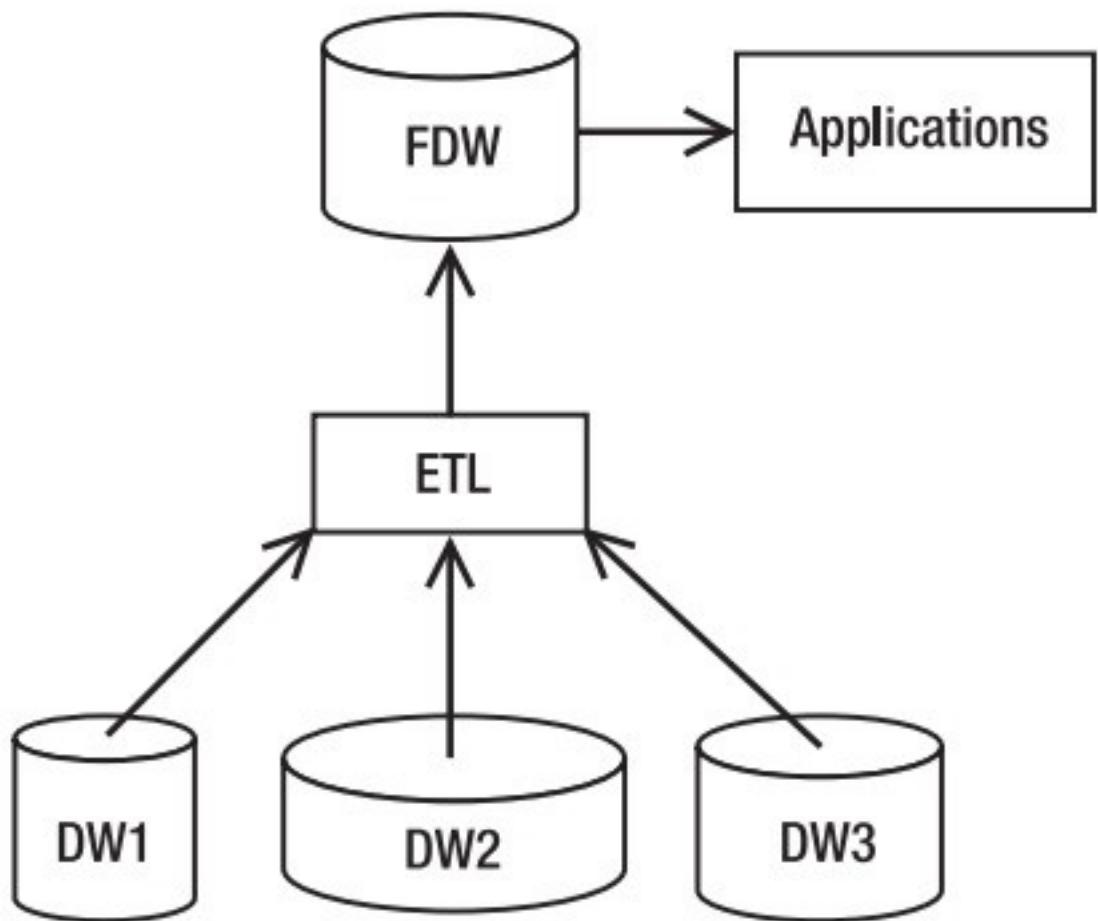
: تصویر NDS + DDS



: تصویر ODS + DDS



: ( federated data warehouse ) FDW تصویر





## نظرات خوانندگان

نویسنده: لیبرتاد  
تاریخ: ۱۳۹۲/۱۰/۱۱ ۱۳:۳

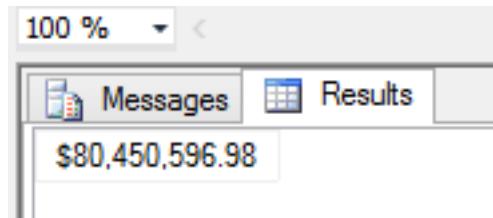
بسیار عالی آیا OLTP و DW می‌توانند بر روی یک سرور باشند مثلاً" بر روی یک SQL Server باشند

نویسنده: اردلان شاه قلی  
تاریخ: ۱۳۹۲/۱۰/۱۱ ۱۵:۴۵

قطعاً امکان پذیر می‌باشد. اما توصیه می‌شود اینچنین نباشد. در ضمن در نظر داشته باشید که در خیلی از موارد، DBMS‌ها سیستم‌های OLTP بر روی SQL Server نمی‌باشند (مثلاً سیستم حسابداری از پایگاه داده‌ی پارادوکس استفاده می‌کند در حالی که سیستم منابع انسانی دارای پایگاه داده‌ی اراکل می‌باشد و ...)

در این قسمت می‌خواهیم بیشتر روی مفاهیم اعمال شرط بر روی خروجی عمل واکشی کار کنیم. برای شروع کوئری ساده‌ی زیر را اجرا و خروجی آن را تفسیر می‌کنیم.

```
Select
From [Adventure Works]
```

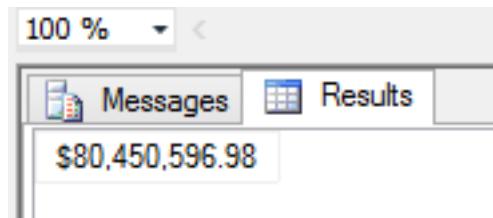


همان طور که مشاهده می‌کنید، خروجی یک عدد می‌باشد. بدون نام ستون یا ردیف؟!

به خاطر بیاورید که هر Cube در SSAS دارای یک Measure است که در صورت عدم اعلام نام یک Measure در کوئری، به صورت پیش فرض مقدار این Measure را برابر می‌گرداند. خوب؛ نام ستون و سطر چرا ذکر نشده است؟ به دلیل عدم اعلام صحیح نام سطر و ستون در کوئری بالا، SSAS نام ستون و سطر خاصی را نمی‌تواند نمایش دهد.

با بررسی کوئری زیر به درک بیشتری از شاخص (Measure) پیش فرض Cube دست پیدا خواهید کرد.

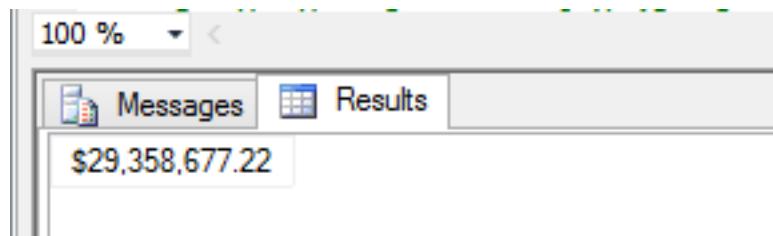
```
Select
From [Adventure Works]
Where
( [Measures].[Reseller Sales Amount] )
```



خروجی همچنان مانند بالا می‌باشد اما در این حالت اعلام شده است که از کدام شاخص باید واکشی انجام شود. دلیل خروجی مشابه، یکسان بودن شاخص پیش فرض و شرط اعلام شده می‌باشد. به بیان دیگر [Measures].[Reseller Sales Amount] در [Adventure Works] به عنوان شاخص پیش فرض معرفی شده است و با اجرای کوئری زیر عمل شرط واکشی برای یک شاخص متفاوت اعمال شده است.

```
Select
From [Adventure Works]
```

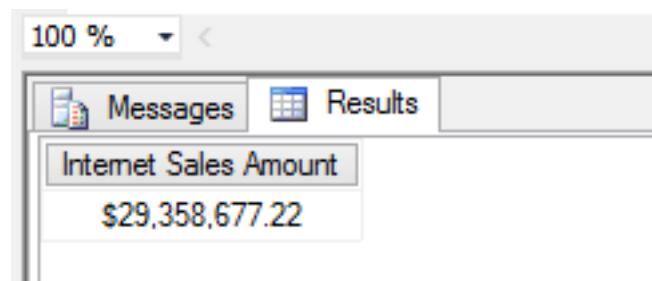
Where [Measures].[Internet Sales Amount]



کوئری زیر را اجرا کنید:

```
Select  
[Measures].[Internet Sales Amount] on columns  
From [Adventure Works]
```

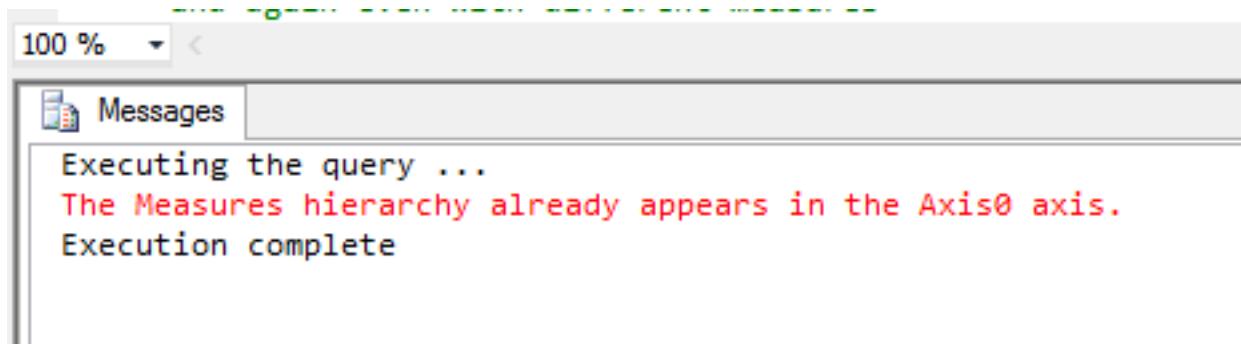
تنها تفاوت دو کوئری بالا ، در آوردن نام ستون می باشد . زیرا در هر دو سرجمع ، یک شاخص واکشی می گردد .



کوئری زیر را اجرا کنید:

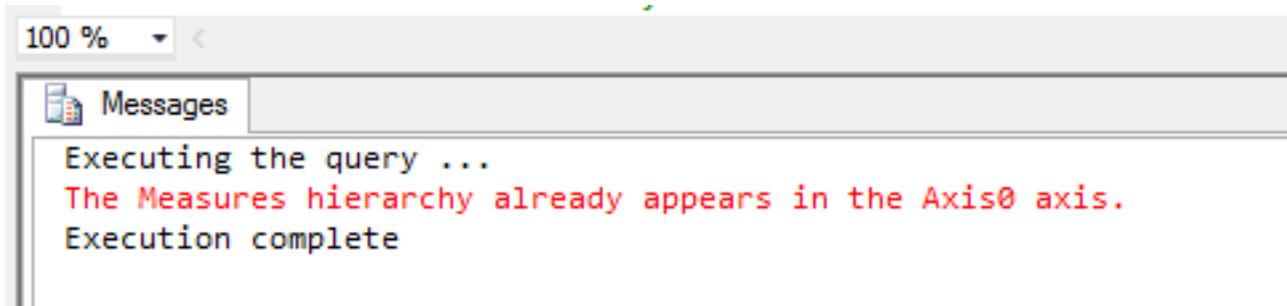
```
select  
[Measures].[Internet Sales Amount] on columns  
From [Adventure Works]  
Where [Measures].[Internet Sales Amount]
```

این کوئری با خطأ مواجه می شود . زیرا در آن کوئری در یک Axis و در شرط ، اعمال انتخاب شاخص شده است که این مورد فقط می بایستی در یکی از این دو قسمت رخ دهد .



و همچنین در صورت انتخاب دو شاخص متفاوت نیز با خطأ بروز خواهیم کرد.

```
Select
[Measures].[Internet Sales Amount] on columns
From [Adventure Works]
Where [Measures].[Reseller Sales Amount]
```



به عبارت دیگر نمی‌توان در خواست فیلتر کردن کوئری را برروی شاخص 1 داد؛ در صورتیکه می‌خواهیم شاخص 2 را واکشی کنیم. اعمال شرط برای واکشی اطلاعات از شاخص، پیش فرض نوشتن این شرط لازم نمی‌باشد؛ زیرا این شاخص به صورت پیش فرض انتخاب شده است.

```
select
{
[Product].[Product Categories].[Category],
[Product].[Product Categories]
}on columns
From [Adventure Works]
Where [Measures].[Reseller Sales Amount]
```

Accessories	Bikes	Clothing	Components	All Products
\$571,297.93	\$66,302,381.56	\$1,777,840.84	\$11,799,076.66	\$80,450,596.98

بنابراین کوئری بالا و کوئری زیر یکسان عمل خواهند کرد:

```
select
{
[Product].[Product Categories].[Category],
[Product].[Product Categories]
}on columns
From [Adventure Works]
```

Accessories	Bikes	Clothing	Components	All Products
\$571,297.93	\$66,302,381.56	\$1,777,840.84	\$11,799,076.66	\$80,450,596.98

حال می‌خواهیم سرجمع فروش نمایندگان فروش محصولات در کشور کانادا را بر اساس دسته بندی محصولات داشته باشیم . برای این منظور کوئری زیر را می‌نویسیم:

```
Select
{
[Product].[Product Categories].[Category],
[Product].[Product Categories]
} on columns
From [Adventure Works]
Where [Customer].[Customer Geography].[Country].[Canada]
```

Accessories	Bikes	Clothing	Components	All Products
\$571,297.93	\$66,302,381.56	\$1,777,840.84	\$11,799,076.66	\$80,450,596.98

با اعمال شرط کشور کانادا، عملاً خروجی فروش نمایندگان فروش در کانادا بر اساس دسته بندی محصولات واکشی می‌گردد. کمی به خروجی دقیق نمایید. مبلغ سرجمع مبالغ فروش نمایندگان فروش برای کانادایی‌ها برابر کل فروش نمایندگان فروش می‌باشد که در کوئیری‌های قبلی بدست آورده‌یم؟!

خروجی این کوئیری مشکوک به نظر می‌رسد. زیرا سرجمع مبالغ فروش نمایندگان فروش برای کانادایی‌ها برابر کل فروش نمایندگان فروش می‌باشد. آیا کانادایی‌ها تمام خرید را انجام داده‌اند؟ خیر.

دلیل این اشکال در این است که هیچ‌گونه ارتباطی بین بعد مشتری و شاخص پیش‌فرض در سیستم وجود ندارد.

مشکل کوئیری بالا در این کوئیری با تغییر بعد در قسمت اعمال شرط برطرف شده؛ اکنون خروجی حقیقی مشاهده می‌شود.

```
Select
{
[Product].[Product Categories].[Category],
[Product].[Product Categories]
} on columns
From [Adventure Works]
Where [Sales Territory].[Sales Territory].[Country].[Canada]
```

	Accessories	Bikes	Clothing	Components	All Products
	\$118,127.35	\$11,636,380.59	\$378,947.63	\$2,244,470.02	\$14,377,925.60

حال اگر بخواهیم دو شرط را به صورت همزمان داشته باشیم به صورت زیر عمل خواهیم کرد:

```
Select
{
[Product].[Product Categories].[Category],
[Product].[Product Categories]
} on columns
From [Adventure Works]
Where (
[Customer].[Customer Geography].[Country].[Canada],
[Measures].[Internet Sales Amount]
)
```

Accessories	Bikes	Clothing	Components	All Products
\$103,377.85	\$1,821,302.39	\$53,164.62	(null)	\$1,977,844.86

در کوئری بالا سرجمع فروش اینترنتی توسط مشتریان کانادایی بدست آمده است.

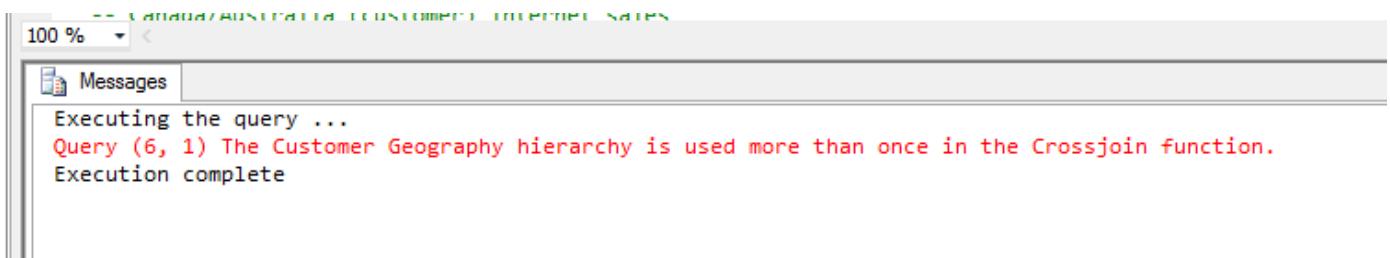
البته می‌توان کوئری فوق را به صورت زیر هم نوشت و در این حالت نام ردیف هم در خروجی قابل مشاهده می‌باشد و البته دیگر نیازی به اعمال شرط، روی نام شاخص نمی‌باشد. زیرا اعمال شرط در ردیف انجام شده است.

```
Select
{
[Product].[Product Categories].[Category],
[Product].[Product Categories]
} on columns,
[Measures].[Internet Sales Amount] On rows
From [Adventure Works]
Where (
[Customer].[Customer Geography].[Country].[Canada]
)
```

Internet Sales Amount	Accessories	Bikes	Clothing	Components	All Products
\$103,377.85	\$1,821,302.39	\$53,164.62	(null)	\$1,977,844.86	

حال اگر بخواهیم فروش اینترنتی را برای استرالیا و کانادا داشته باشیم به صورت زیر عمل می‌کنیم.

```
Select
{
[Product].[Product Categories].[Category],[Product].[Product Categories]
} on columns
From [Adventure Works]
Where
(
[Customer].[Customer Geography].[Country].[Canada],
[Customer].[Customer Geography].[Country].[Australia],
[Measures].[Internet Sales Amount]
)
```



در اینجا ما نیاز داریم میزان فروش اینترنتی کانادا و استرالیا را برای انواع محصولات بدست آوریم ، اما نحوه استفاده از دو ساختار سلسله مراتبی مرتبط با یک دایمنشن را درست رعایت نکرده ایم . بنابر این کوئری زیر را اجرا خواهیم کرد :

```
Select
{
[Product].[Product Categories].[Category],
[Product].[Product Categories]
} on columns
From [Adventure Works]
Where
(
{
[Customer].[Customer Geography].[Country].[Canada],
[Customer].[Customer Geography].[Country].[Australia]
},
[Measures].[Internet Sales Amount]
)
```

Accessories	Bikes	Clothing	Components	All Products
\$242,068.48	\$10,673,352.40	\$123,424.57	(null)	\$11,038,845.45

که همان کوئری بالا می باشد با این تفاوت که از {} استفاده شده است .

درابتدا میزان فروش نمایندگان فروش در انگلستان را بدست می آوریم:

```
Select
{
[Product].[Product Categories].[Category],
[Product].[Product Categories]
} on columns
From [Adventure Works]
Where
[Sales Territory].[Sales Territory].[Country].[United Kingdom]
```

Accessories	Bikes	Clothing	Components	All Products
\$42,593.03	\$3,405,747.21	\$118,828.80	\$711,839.79	\$4,279,008.83

و برای بدست آوردن فروش اینترنتی تمام کشور ها به جز انگلستان بر اساس دسته بندی محصولات کوئری زیر را خواهیم نوشت

```
Select
{
[Product].[Product Categories].[Category],
[Product].[Product Categories]
} on columns
From [Adventure Works]
Where
[Sales Territory].[Sales Territory].[Country]
-
[Sales Territory].[Sales Territory].[Country].[United Kingdom]
```

Accessories	Bikes	Clothing	Components	All Products
\$528,704.90	\$62,896,634.35	\$1,659,012.04	\$11,087,236.86	\$76,171,588.16

البته از تابع Except هم می توان به صورت زیر استفاده کرد

```
Select
{
[Product].[Product Categories].[Category],
[Product].[Product Categories]
} on columns
From [Adventure Works]
Where
except(
[Sales Territory].[Sales Territory].[Country],
[Sales Territory].[Sales Territory].[Country].[United Kingdom]
)
```

Accessories	Bikes	Clothing	Components	All Products
\$528,704.90	\$62,896,634.35	\$1,659,012.04	\$11,087,236.86	\$76,171,588.16

عملگر منها مشابه except کار میکند.

## آموزش MDX Query - قسمت شانزدهم - استفاده از تابع Filter در MDX Query ها

عنوان:

اردلان شاه قلی

نویسنده:

۱۱:۱۵ ۱۳۹۲/۱۰/۱۳

تاریخ:

[www.dotnettips.info](http://www.dotnettips.info)

آدرس:

SQL Server 2012, OLAP, MDX, SSAS

گروهها:

در این قسمت بر روی تابع Filter در MDX Query ها تمرکز خواهیم کرد. برای آشنایی با این تابع یک سری از کوئیری ها را اجرا کرده و به بررسی آنها می پردازیم.

```
Select
{
[Measures].[Internet Sales Amount],
[Measures].[Reseller Sales Amount]
} on columns,
[Product].[Product Categories].[Category] on rows
From [Adventure Works]
```

دقیق کنید که در واکنش، مقدار فروش اینترنتی Component برابر Null می باشد.

	Internet Sales Amount	Reseller Sales Amount
Accessories	\$700,759.96	\$571,297.93
Bikes	\$28,318,144.65	\$66,302,381.56
Clothing	\$339,772.61	\$1,777,840.84
Components	(null)	\$11,799,076.66

```
Select
{
[Measures].[Internet Sales Amount],
[Measures].[Reseller Sales Amount]
} on columns,
filter(
[Product].[Product Categories].[Category],
[Measures].[Internet Sales Amount] > 0
) on rows
From [Adventure Works]
```

	Internet Sales Amount	Reseller Sales Amount
Category	\$700,759.96	\$571,297.93
Bikes	\$28,318,144.65	\$66,302,381.56
Clothing	\$339,772.61	\$1,777,840.84

با توجه به شرطی که اعمال شده است، فقط دسته بندی محصولاتی در خروجی می آید که فروش اینترنتی آنها بیشتر از صفر یا برابر Null نباشند.

استفاده از کلید واژه‌ی Having در هر محور کاری شبیه به انجام عمل فیلترینگ می باشد .

```
Select
{
[Measures].[Internet Sales Amount],
[Measures].[Reseller Sales Amount]
} on columns,
[Product].[Product Categories].[Category]
having [Measures].[Internet Sales Amount] > 0 on rows
From [Adventure Works]
```

	Internet Sales Amount	Reseller Sales Amount
Category	\$700,759.96	\$571,297.93
Bikes	\$28,318,144.65	\$66,302,381.56
Clothing	\$339,772.61	\$1,777,840.84

اگر بخواهیم میزان فروش اینترنتی و میزان فروشنمایندگان فروش را برای محصولاتی واکشی کنیم که میزان فروش اینترنتی آنها بیش از 500000 دلار می باشد ، کوئری زیر را خواهیم داشت :

```
Select
{
[Measures].[Internet Sales Amount],
[Measures].[Reseller Sales Amount]
} on columns,
filter(
[Product].[Product Categories].[Category],
[Measures].[Internet Sales Amount] > 500000
) on rows
From [Adventure Works]
```

	Internet Sales Amount	Reseller Sales Amount
Accessories	\$700,759.96	\$571,297.93
Bikes	\$28,318,144.65	\$66,302,381.56

و برای ایجاد شرط ترکیبی بر روی شاخص، به صورت زیر عمل خواهیم کرد :

```
Select
{
[Measures].[Internet Sales Amount],
[Measures].[Reseller Sales Amount]
} on columns,
filter(
[Product].[Product Categories].[Category],
[Measures].[Internet Sales Amount] > 500000
and
[Measures].[Internet Sales Amount] < 750000
) on rows
From [Adventure Works]
```

	Internet Sales Amount	Reseller Sales Amount
Accessories	\$700,759.96	\$571,297.93

در مثال بالا، دسته بندی محصولاتی در خروجی واکشی شده اند که میزان فروش اینترنتی آنها بیش از 500 هزار و کمتر از 750 هزار می باشد.

استفاده از And , Or در شرط ترکیبی مجاز می باشد

```
Select
{
[Measures].[Internet Sales Amount],
[Measures].[Reseller Sales Amount]
} on columns,
```

```
filter(
[Product].[Product Categories].[Category],
[Measures].[Internet Sales Amount] > 750000
or
[Measures].[Internet Sales Amount] < 500000
) on rows
From [Adventure Works]
```

	Internet Sales Amount	Reseller Sales Amount
Bikes	\$28,318,144.65	\$66,302,381.56
Clothing	\$339,772.61	\$1,777,840.84
Components	(null)	\$11,799,076.66

در زیر با استفاده از And، شرط برروی میزان فروش نمایندگان فروش نیز قرارداده شده است.

```
Select
{
[Measures].[Internet Sales Amount],
[Measures].[Reseller Sales Amount]
} on columns,
filter(
[Product].[Product Categories].[Category],
(
[Measures].[Internet Sales Amount] > 750000
or
[Measures].[Internet Sales Amount] < 500000
)
and
[Measures].[Reseller Sales Amount] < 15000000
) on rows
From [Adventure Works]
```

در هنگام ایجاد شروط ترکیبی حتما از () استفاده کنید.

	Internet Sales Amount	Reseller Sales Amount
Bikes	\$28,318,144.65	\$66,302,381.56
Clothing	\$339,772.61	\$1,777,840.84
Components	(null)	\$11,799,076.66

حال میخواهیم دو شاخص در یک ردیف، با یکدیگر مقایسه شوند و در صورت صحت شرط، آن ردیف در خروجی قرار گیرد

```
Select
{
[Measures].[Internet Sales Amount],
[Measures].[Reseller Sales Amount]
} on columns,
filter(
[Product].[Product Categories].[Category],
[Measures].[Internet Sales Amount]
>
[Measures].[Reseller Sales Amount]
) on rows
From [Adventure Works]
```

The screenshot shows the SSMS interface with the 'Results' tab selected. The results grid has three columns: 'Accessories' (row label), 'Internet Sales Amount' (\$700,759.96), and 'Reseller Sales Amount' (\$571,297.93).

	Internet Sales Amount	Reseller Sales Amount
Accessories	\$700,759.96	\$571,297.93

ایجاد فیلترینگ با استفاده از currentmember و عملگر Is .

```
Select
{
[Measures].[Internet Sales Amount],
[Measures].[Reseller Sales Amount]
} on columns,
filter(
[Product].[Product Categories].[Category],
[Product].[Product Categories].currentmember
is
[Product].[Product Categories].[Category].[Bikes]
) on rows
From [Adventure Works]
```

The screenshot shows the SSMS interface with the 'Results' tab selected. The results grid has three columns: 'Bikes' (row label), 'Internet Sales Amount' (\$28,318,144.65), and 'Reseller Sales Amount' (\$66,302,381.56).

	Internet Sales Amount	Reseller Sales Amount
Bikes	\$28,318,144.65	\$66,302,381.56

البته در مثال بالا می توانیم به جای استفاده از Is از Filtering عمل بر روی شاخص ها انجام شده است. اما امکان اعمال Dimension Filter روی ها نیز وجود دارد.

کوئری زیر را بررسی کنید :

```
Select
{
[Measures].[Internet Sales Amount],
[Measures].[Reseller Sales Amount]
} on columns,
filter(
[Product].[Product Categories].[Category],
[Product].[Product Categories].currentmember
is
[Product].[Product Categories].[Category].[Bikes]
or
[Product].[Product Categories].currentmember
is
[Product].[Product Categories].[Category].[Accessories]
) on rows
From [Adventure Works]
```

	Internet Sales Amount	Reseller Sales Amount
Accessories	\$700,759.96	\$571,297.93
Bikes	\$28,318,144.65	\$66,302,381.56

در کوئری بالا میزان فروش نمایندگان فروش و فروش اینترنتی برای دسته بندی های Bike و Accessories واکنشی شده است.

امکان ایجاد شرایط ترکیبی از شاخص ها و بعدها در یک Filter نیز وجود دارد.

```
Select
{
[Measures].[Internet Sales Amount],
[Measures].[Reseller Sales Amount]
} on columns,
filter(
[Product].[Product Categories].[Category],
(
[Product].[Product Categories].currentmember
is
[Product].[Product Categories].[Category].[Bikes]
and
[Measures].[Reseller Sales Amount] > 1000000
)
or
(
[Product].[Product Categories].currentmember
is
[Product].[Product Categories].[Category].[Accessories]
)
and
```

## آموزش MDX Query - قسمت شانزدهم - استفاده از تابع Filter در MDX Query

```
[Measures].[Reseller Sales Amount] > 750000  
) on rows  
From [Adventure Works]
```

	Internet Sales Amount	Reseller Sales Amount
Bikes	\$28,318,144.65	\$66,302,381.56

همچنین می‌توان از Not درون شرط Filter استفاده کرد

```
Select  
{  
[Measures].[Internet Sales Amount],  
[Measures].[Reseller Sales Amount]  
} on columns,  
filter(  
[Product].[Product Categories].[Category],  
not  
(  
[Product].[Product Categories].currentmember  
is  
[Product].[Product Categories].[Category].[Clothing]  
)  
) on rows  
From [Adventure Works]
```

	Internet Sales Amount	Reseller Sales Amount
Accessories	\$700,759.96	\$571,297.93
Bikes	\$28,318,144.65	\$66,302,381.56
Components	(null)	\$11,799,076.66

در زیر می‌خواهیم به بررسی تابع non Empty بپردازیم . برای این منظور در ابتدا کوئری زیر را اجرا کنید :

```
Select  
{
```

```
[Measures].[Internet Sales Amount],
[Measures].[Reseller Sales Amount]
} on columns,
[Product].[Product Categories].[Subcategory] on rows
From [Adventure Works]
```

The screenshot shows a query results window with two tabs: 'Messages' and 'Results'. The 'Results' tab displays a table with three columns: 'Subcategory' (rows), 'Internet Sales Amount', and 'Reseller Sales Amount'. The data includes:

Subcategory	Internet Sales Amount	Reseller Sales Amount
Bike Racks	\$39,360.00	\$197,736.16
Bike Stands	\$39,591.00	(null)
Bottles and Cages	\$56,798.19	\$7,476.60
Cleaners	\$7,218.60	\$11,188.37
Fenders	\$46,619.58	(null)
Helmets	\$225,335.60	\$258,712.93
Hydration Packs	\$40,307.67	\$65,518.75
Lights	(null)	(null)
Locks	(null)	\$16,225.22
Panniers	(null)	(null)
Pumps	(null)	\$13,514.69
Tires and Tubes	\$245,529.32	\$925.21
Mountain Bikes	\$9,952,759.56	\$26,492,684.38
Road Bikes	\$14,520,584.04	\$29,358,206.96

با استفاده از تابع NonEmpty، ردیف ها یا ستون هایی حذف می گردند که تمامی مقادیر آنها ، در آن ردیف یا در آن ستون برابر Null باشند.

```
Select
{
[Measures].[Internet Sales Amount],
[Measures].[Reseller Sales Amount]
} on columns,
non empty [Product].[Product Categories].[Subcategory] on rows
From [Adventure Works]
```

The screenshot shows the Analysis Studio interface with the 'Results' tab selected. The results are presented in a table with three columns: Product Category, Internet Sales Amount, and Reseller Sales Amount.

	Internet Sales Amount	Reseller Sales Amount
Bike Racks	\$39,360.00	\$197,736.16
Bike Stands	\$39,591.00	(null)
Bottles and Cages	\$56,798.19	\$7,476.60
Cleaners	\$7,218.60	\$11,188.37
Fenders	\$46,619.58	(null)
Helmets	\$225,335.60	\$258,712.93
Hydration Packs	\$40,307.67	\$65,518.75
Locks	(null)	\$16,225.22
Pumps	(null)	\$13,514.69
Tires and Tubes	\$245,529.32	\$925.21
Mountain Bikes	\$9,952,759.56	\$26,492,684.38
Road Bikes	\$14,520,584.04	\$29,358,206.96
Touring Bikes	\$3,844,801.05	\$10,451,490.22
Other	(null)	\$166,729.71

در قسمت آینده برروی توابع Top , Count و ... تمرکز خواهیم کرد.

## نظرات خوانندگان

نویسنده: سوسن  
تاریخ: ۱۳۹۲/۱۰/۲۸ ۲۳:۳۸

سلام و خسته نباشید؛ یه سوال داشتم: کوئری های SSAS م رو به جای اینکه با استفاده از reporting service بخواه نمایش بدم باید با استفاده از یه فرم توی C# نمایش بدم. یعنی با استفاده از ADOMD data adaptor یا CE11SEt ... شبیه اینها. سوال من اینجاست که اگه بخواه روی فرم که قراره کوئری هام تو ش نمایش پیدا کنه یه textbox بذارم و بخواه که مقداری که توی textbox وارد میشه رو بعنوان یه پارامتر توی دستور mdx تاثیر بدم باید چیکار کنم؟ بطور مثال من یه تاریخ توی textbox روی فرم وارد کنم و اونوقت کوئری بجای اینکه مثه قبل تمام محصولات فروخته شده رو نشون اون محصولاتی رو که توی اون تاریخ مشخص شده فروخته شده (توی ذکر شده) رو نشون بده. ممنون

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۲/۱۰/۲۹ ۱۰:۲۴

با استفاده از ADO.NET نمیتوان کوئری های MDX را مستقیماً اجرا کرد. برای اینکار نیاز به Microsoft® SQL Server® 2012 Feature Microsoft.AnalysisServices.AdomdClient.dll Microsoft® SQL Server® 2012 ADOMD.NET Pack مراجعه و قسمت Microsoft® SQL Server® 2008 به صفحه Microsoft® SQL Server® 2008 R2 Feature Pack Analysis Management Objects Microsoft® SQL Server® 2008 R2 Feature Pack نیز در اینجا پروایدر ADOMD قابل دریافت است.

بعد برای استفاده از آن خواهید داشت:

```
using (AdomdConnection conn =
    new AdomdConnection("Data Source=tfsDB;Initial Catalog=Tfs_Analysis; MDX
Compatibility=1;"))
{
    conn.Open();
    using (AdomdCommand cmd = new AdomdCommand(" ..... mdxQuery here ..... ", conn))
    {
        DataSet ds = new DataSet();
        ds.EnforceConstraints = false;
        ds.Tables.Add();
        DataTable dt = ds.Tables[0];
        dt.Load(cmd.ExecuteReader());
        return dt;
    }
}
```

و یا پژوهه MdxClient - ADO.NET data provider for MDX queries سعی کرده این روش را تبدیل به یک پروایدر جدید ADO.NET کند. نکته‌ی جالب این پروایدر امکان تعریف مقادیر دریافتی از کاربر به صورت پارامتر است در کوئری‌ها تعریف شده. [فایل readme](#) آن را برای مشاهده یک سری مثال در مورد نحوه تعریف پارامترها و دریافت داده‌های ارسالی از طریق کاربر، مطالعه کنید.

در این قسمت بر روی توابع Topcount, bottomcount , toppercent, bottompercent, topsum, bottomsum تمرکز خواهیم داشت.

در ابتدا تصور کنید بخواهیم میزان فروش اینترنتی را برای پنج ردیف از دسته بندی‌های محصولات واکشی کنیم.

```
Select
[Measures].[Internet Sales Amount] on columns,
non empty(
topcount([Product].[Product Categories].[Subcategory],5)
) on rows
From [Adventure Works]
```

	Internet Sales Amount
Bike Racks	\$39,360.00
Bike Stands	\$39,591.00
Bottles and Cages	\$56,798.19
Cleaners	\$7,218.60
Fenders	\$46,619.58

در تابع بالا پنج ردیف ابتدایی (به صورت فیزیکی) برگردانده می‌شوند.

در اینجا تابع topcount دارای دو پارامتر می‌باشد که پارامتر دوم آن مشخص کننده‌ی تعداد ردیف واکشی شده و پارامتر اول آن، مشخص کننده‌ی دایمنشنسی می‌باشد که عمل واکشی برای آن صورت می‌گیرد. همچنین در بالا از تابع Non empty برای حذف ردیف‌های دارای مقدار Null استفاده شده است. حال تصور کنید بخواهیم پنج دسته بندی محصولی را دریافت کنیم که دارای بیشترین میزان فروش اینترنتی می‌باشند.

```
Select
[Measures].[Internet Sales Amount] on columns,
non empty(
topcount(
[Product].[Product Categories].[Subcategory],
5,
[Measures].[Internet Sales Amount]
)
) on rows
From [Adventure Works]
```

	Internet Sales Amount
Road Bikes	\$14,520,584.04
Mountain Bikes	\$9,952,759.56
Touring Bikes	\$3,844,801.05
Tires and Tubes	\$245,529.32
Helmets	\$225,335.60

خروجی بر اساس میزان فروش اینترنتی به صورت نزولی مرتب شده است.

تابع Topcount به عنوان پارامتر سوم میتواند نام یک Measure را دریافت کند و خروجی را براساس آن شاخص، برگرداند. امکان واکشی و مرتب سازی در تابع Topcount:

برای یک شاخص متفاوت از شاخص واکشی شده در یک محور دیگر نیز وجود دارد به مثال زیر دقت کنید:

```
Select
[Measures].[Internet Sales Amount] on columns,
topcount(
[Product].[Product Categories].[Subcategory],
5,
[Measures].[Reseller Sales Amount]
) on rows
From [Adventure Works]
```

	Internet Sales Amount
Road Bikes	\$14,520,584.04
Mountain Bikes	\$9,952,759.56
Touring Bikes	\$3,844,801.05
Mountain Frames	(null)
Road Frames	(null)

همانطور که مشخص میباشد، پنج دسته بندی محصولاتی که دارای بیشترین میزان فروش نمایندگان فروش میباشند، در خروجی واکشی شده‌اند؛ در حالیکه در محور ستون میزان فروش اینترنتی واکشی شده است.

برای درک بیشتر همین کوئری را دوباره بازنویسی کرده اما اینبار در محور ستون هر دو شاخص [Measures].[Internet Sales] و [Measures].[Reseller Sales Amount] را واکشی می‌کنیم.

```
Select {[Measures].[Internet Sales Amount],[Measures].[Reseller Sales Amount]} on columns,
topcount(
[Product].[Product Categories].[Subcategory],
5,
[Measures].[Reseller Sales Amount]
) on rows
From [Adventure Works]
```

	Internet Sales Amount	Reseller Sales Amount
Road Bikes	\$14,520,584.04	\$29,358,206.96
Mountain Bikes	\$9,952,759.56	\$26,492,684.38
Touring Bikes	\$3,844,801.05	\$10,451,490.22
Mountain Frames	(null)	\$4,713,672.15
Road Frames	(null)	\$3,849,853.34

با بررسی خروجی دو کوئری بالا تفاوت واکشی را متوجه خواهید شد. در هر دو کوئری واکشی براساس شاخص [Measures].[Reseller Sales Amount] انجام شده است

اما واکشی در محور ستون متفاوت می‌باشد. (دقیقاً مانند T/SQL که می‌توانستیم، مرتب سازی براساس فیلڈی باشد که در قسمت Projection حاضر نبوده و در این حالت در برخی موارد ظاهراً خروجی مرتب نمی‌باشد)

حال تصور کنید بخواهیم 30 دسته بندی محصولاتی را داشته باشیم که دارای کمترین میزان فروش اینترنتی می‌باشند. برای این منظور از تابع bottomcount استفاده می‌کنیم

```
Select
[Measures].[Internet Sales Amount] on columns,
bottomcount(
[Product].[Product Categories].[Subcategory],
30,
[Measures].[Internet Sales Amount]
) on rows
From [Adventure Works]
```

100 %

	Internet Sales Amount
Wheels	(null)
Touring Frames	(null)
Saddles	(null)
Road Frames	(null)
Pedals	(null)
Mountain Frames	(null)
Headsets	(null)
Handlebars	(null)
Forks	(null)
Derailleurs	(null)
Cranksets	(null)
Chains	(null)
Brakes	(null)
Bottom Brackets	(null)
Tights	(null)
Bib-Shorts	(null)
Pumps	(null)
Panniers	(null)
Locks	(null)
Lights	(null)
Socks	\$5,106.32
Cleaners	\$7,218.60
Caos	\$19,688.10

ردیف هایی که دارای مقدار Null می باشند هم در خروجی قرار می گیرند

```
Select
[Measures].[Internet Sales Amount] on columns,
non empty bottomcount(
[Product].[Product Categories].[Subcategory],
30,
[Measures].[Internet Sales Amount]
)on rows
From [Adventure Works]
```

	Internet Sales Amount
Socks	\$5,106.32
Cleaners	\$7,218.60
Caps	\$19,688.10
Gloves	\$35,020.70
Vests	\$35,687.00
Bike Racks	\$39,360.00
Bike Stands	\$39,591.00
Hydration Packs	\$40,307.67
Fenders	\$46,619.58
Bottles and Cages	\$56,798.19

در مثال بالا ردیف های دارای مقدار Null را از خروجی حذف کرده ایم.

گاهی نیاز می باشد که تعداد دسته بندی های محصولاتی را واکنشی کنیم که دارای بیشترین یا کمترین میزان فروش اینترنتی می باشند و سرجمع فروش اینترنتی آنها بیشتر یا کمتر از X درصد از فروش اینترنتی کل می باشد را داشته باشند. به عنوان مثال می خواهیم ببینیم کدام دسته بندی محصولات شامل بیشترین میزان فروش اینترنتی می باشند و سرجمع فروش آنها 53 درصد از کل فروش اینترنتی می باشند.

```
Select
[Measures].[Internet Sales Amount] on columns,
{
toppercent(
[Product].[Product Categories].[Subcategory],
53,
[Measures].[Internet Sales Amount]
),
[Product].[Product Categories]
} on rows
From [Adventure Works]
```

	Internet Sales Amount
Road Bikes	\$14,520,584.04
Mountain Bikes	\$9,952,759.56
All Products	\$29,358,677.22

و یا واکنشی دسته محصولاتی که دارای کمترین میزان فروش اینترنتی میباشند و سرجمع فروش اینترنتی آنها کمتر از 1 درصد کل میزان فروش اینترنتی میباشد.

```
Select
[Measures].[Internet Sales Amount] on columns,
non empty bottompercent(
[Product].[Product Categories].[Subcategory],
--0.01,
1,
[Measures].[Internet Sales Amount]
) on rows
From [Adventure Works]
```

	Internet Sales Amount
Socks	\$5,106.32
Cleaners	\$7,218.60
Caps	\$19,688.10
Gloves	\$35,020.70
Vests	\$35,687.00
Bike Racks	\$39,360.00
Bike Stands	\$39,591.00
Hydration Packs	\$40,307.67
Fenders	\$46,619.58
Bottles and Cages	\$56,798.19

کاربرد تابع Topsum در کوئری زیر نمایش داده شده است

```
Select
[Measures].[Internet Sales Amount] on columns,
topsum(
[Product].[Product Categories].[Subcategory],
25000000,
[Measures].[Internet Sales Amount]
) on rows
From [Adventure Works]
```

	Internet Sales Amount
Road Bikes	\$14,520,584.04
Mountain Bikes	\$9,952,759.56
Touring Bikes	\$3,844,801.05

در این کوئری از تابع TopSum استفاده شده است که عملاً حداکثر تعداد دسته بندی محصولاتی را بازیابی می‌کند که دارای بیشترین میزان فروش بوده اند و همچنین در مجموع بیش از 25000000 فروش داشته باشند.

تابع bottomsum عملاً تعداد دسته بندی محصولاتی را که دارای کمترین میزان فروش بوده اند و همچنین سرجمع میزان فروش اینترنتی آنها 100000 بوده است را برابر می‌گرداند. البته خروجی توسط non empty filter شده است و خروجی هایی که کاملاً Null می‌باشند، حذف گردیده اند.

```
Select
[Measures].[Internet Sales Amount] on columns,
non empty bottomsum(
[Product].[Product Categories].[Subcategory],
100000,
[Measures].[Internet Sales Amount]
) on rows
From [Adventure Works]
```

The screenshot shows the SSMS interface with the 'Results' tab selected. The results are displayed in a table titled 'Internet Sales Amount'. The table contains five rows, each representing a product and its sales amount.

	Internet Sales Amount
Socks	\$5,106.32
Cleaners	\$7,218.60
Caps	\$19,688.10
Gloves	\$35,020.70
Vests	\$35,687.00