<http://www.sqlleader.com/mboard.asp?exec=view&strBoardID=SS2005SSAS&intSeq=3277>

**Many-to-Many 차원 관계**

**Premier Field Engineer**

**한대성**

|  |
| --- |
| *요즘 한참 Analysis Services에 대해 이리저리 공부하고 있는데, 이전부터 잘 이해가 안되었던 Many-to-many Dimension Relationship에 대해 좀 정리해봤습니다. 책이나 사이트 들에서는 AdventureWorksDW DB를 이용해서 설명하는데 구조가 잘 이해가 안되어서 단순화 한 예로 설명드립니다.*  *본 글 역시 다른 분들에겐 어떨지 모르겠네요~^^* |

다음과 같은 매출 테이블을 생각해 봅시다.

|  |  |  |  |
| --- | --- | --- | --- |
| **SalesID** | **판매일** | **상품** | **판매액** |
| 1 | 2009-06-01 | 자전거 | 200,000 |
| 2 | 2009-06-01 | 자동차 | 3,000,000 |
| 3 | 2009-06-02 | 오토바이 | 1,500,000 |
| 4 | 2009-06-02 | 자전거 | 300,000 |

이 정보를 이용하여 큐브(Cube)로 만들기 위해 **판매일**이라는 차원과 **상품**이라는 차원을 만들어서 간단한 다차원 분석을 할 수 있습니다.

이 때, 만약, **판매원**이라는 차원이 추가되는 경우를 생각해 봅시다.

그런데, 판매원은 판매 건, 즉 SalesID 하나에 한 명이 매핑되는 것은 아닙니다. SalesID가 2인 매출, 즉 2009년 6월 1일에 자동차를 파는데 두 명의 판매원이 같이 고객을 설득해서 팔았다고 합니다. 또한 6월 2일에 오토바이를 팔 때에는 판매원 세 명이 협동해서 팔았다고 합니다.

즉, 다음과 같이 나누어지는 것이지요.

|  |  |
| --- | --- |
| **SalesID** | **판매원** |
| 1 | 장원주 |
| 2 | 한대성 |
| 백도훈 |
| 3 | 한대성 |
| 백도훈 |
| 장원주 |
| 4 | 윤성룡 |
| 백도훈 |

매출 테이블을 기준으로 봤을 때에는 매출 건 하나가 여러 판매원 차원과 연결되는 것입니다.

즉 300만 원짜리 2번 매출은 판매일과 상품이라는 차원에는 1:1로 연결되지만, 판매원 차원에 대해서는 1:3, 즉 1:Many로 연결됩니다.

이 경우를 **Many-to-Many 차원 관계** 또는 **다대다 차원 관계**라고 합니다.

자, 이 차원관계를 이용해서 그냥 일반적인 방법으로 연결을 맺는다면 다음과 같은 결과가 초래(?)됩니다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 자전거 | 자동차 | 오토바이 | **합계** |
| 장원주 | 200,000 |  | 1,500,000 | 1,700,000 |
| 백도훈 | 300,000 | 3,000,000 | 1,500,000 | 4,800,000 |
| 한대성 |  | 3,000,000 | 1,500,000 | 4,500,000 |
| 윤성룡 | 300,000 |  |  | 300,000 |
| **합계** | 800,000 | 6,000,000 | 4,500,000 | 11,300,000 |
| **실제합계** | 500,000 | 3,000,000 | 1,500,000 | 5,000,000 |

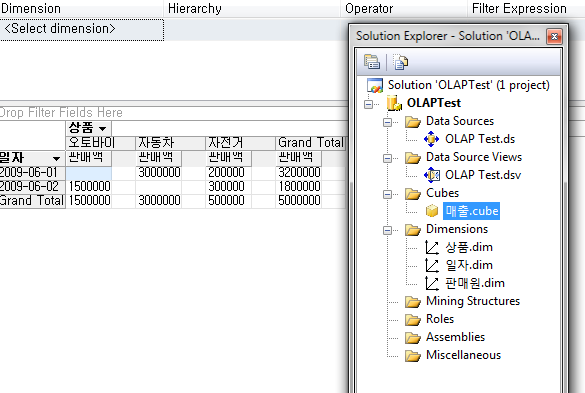
분명 자전거의 매출은 500,000원인데, 판매원 별 합계를 해보니 800,000원으로 나옵니다. 4번 자전거 팔 때 {백도훈}과 {윤성룡}이 판매했기 때문에 각각 300,000원으로 나오는 것은 맞습니다. 그렇다고 합계를 단순한 합으로 할 수는 없는 것이죠.

위의 **실제합계**와 같이 나오게 하도록 설정하는 방법이 SSAS에서 제공됩니다. 바로 **Many-to-many 차원 관계**로 설정하는 것이지요.

다음과 같은 스크립트로 테스트 데이터를 만듭니다.

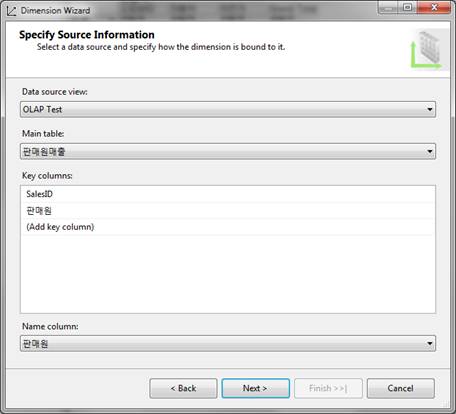
|  |
| --- |
| CREATE DATABASE OLAPTest  GO    USE OLAPTest  GO      CREATE TABLE 일자(일자 CHAR(10) PRIMARY KEY)  GO    INSERT 일자 VALUES ('2009-06-01')  INSERT 일자 VALUES ('2009-06-02')  GO      CREATE TABLE 상품(상품 VARCHAR(20) PRIMARY KEY)  GO    INSERT 상품 VALUES ('자전거')  INSERT 상품 VALUES ('자동차')  INSERT 상품 VALUES ('오토바이')  GO    CREATE TABLE 판매원(판매원 VARCHAR(20) PRIMARY KEY)  GO    INSERT 판매원 VALUES ('장원주')  INSERT 판매원 VALUES ('백도훈')  INSERT 판매원 VALUES ('윤성룡')  INSERT 판매원 VALUES ('한대성')  GO      CREATE TABLE 매출  (  SalesID INT IDENTITY PRIMARY KEY,  판매일 CHAR(10) REFERENCES 일자(일자),  상품 VARCHAR(20) REFERENCES 상품(상품),  판매액 MONEY  )  GO  INSERT 매출(판매일, 상품, 판매액) VALUES('2009-06-01','자전거',200000)  INSERT 매출(판매일, 상품, 판매액) VALUES('2009-06-01','자동차',3000000)  INSERT 매출(판매일, 상품, 판매액) VALUES('2009-06-02','오토바이',1500000)  INSERT 매출(판매일, 상품, 판매액) VALUES('2009-06-02','자전거',300000)  GO    CREATE TABLE 판매원매출  (  SalesID INT REFERENCES 매출(SalesID),  판매원 VARCHAR(20) REFERENCES 판매원(판매원),  CONSTRAINT PK\_판매원매출 PRIMARY KEY(SalesID, 판매원)  )  GO    INSERT 판매원매출  SELECT 1,'장원주' UNION ALL  SELECT 2,'한대성' UNION ALL  SELECT 2,'백도훈' UNION ALL  SELECT 3,'한대성' UNION ALL  SELECT 3,'백도훈' UNION ALL  SELECT 3,'장원주' UNION ALL  SELECT 4,'윤성룡' UNION ALL  SELECT 4,'백도훈'  GO |

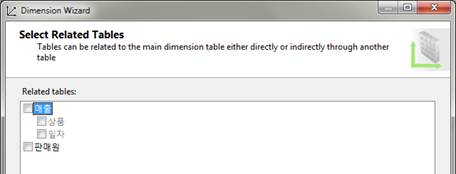
이 데이터로 간단한 Analysis Services DB를 만듭니다.

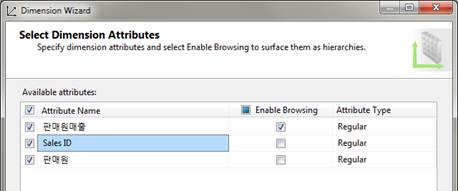


   **Many-to-many 차원 구성하기**

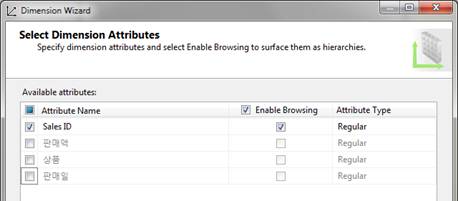
a.     판매원매출 테이블을 이용해서 다음과 같은 차원을 만듭니다.

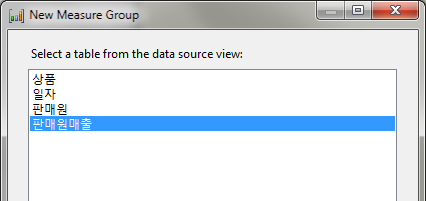
 

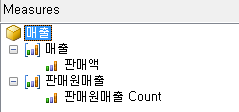
*  
(Attribute Name은 위의 그림과 같이 변경)*

b.     동일한 방법으로 **매출** 테이블을 이용하여 **매출** 차원을 만듭니다.

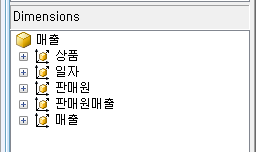


c.     큐브에서 **새 측정값 그룹**을 선택한 후, **판매원매출** 테이블로 측정값 그룹을 만듭니다. 이 그룹에는 카운트만 들어갑니다.

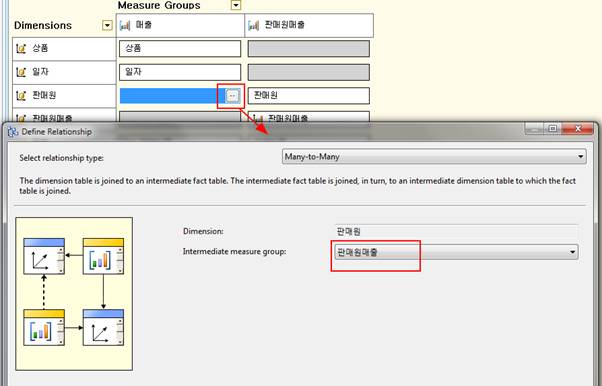
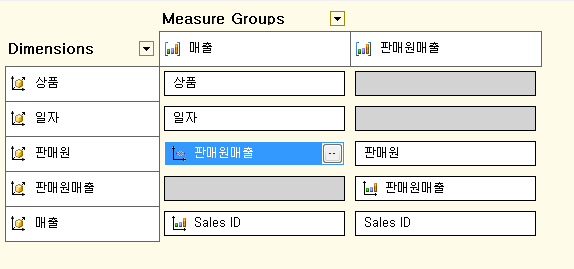


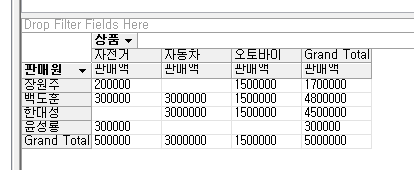


d.     Dimensions에 **판매원**, **판매원매출, 매출** 차원을 추가합니다.



e.     **Dimension Usage** 탭에서 **매출** 측정값 그룹의 판매원 차원에 대해 다음과 같이 설정합니다.

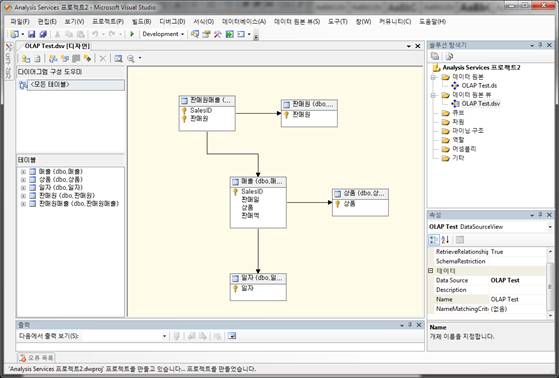
f.      큐브를 프로세싱 한 후, 데이터를 확인해 봅니다.  
  
판매원 별로 실적이 나오지만, 합계는 실제 매출 합계로 나오는 것을 확인하실 수 있습니다.

SQL Server 2005에서 상세 요청이 있어서 보충 설명합니다. 참고하세요.

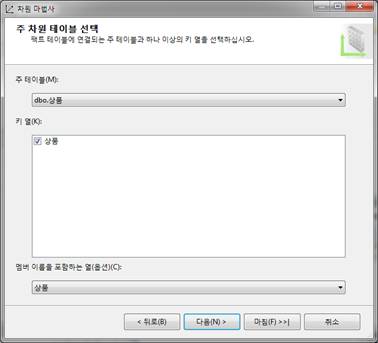
1.     스크립트를 이용해서 데이터베이스와 테이블을 만듭니다.

2.     데이터 원본을 추가합니다.

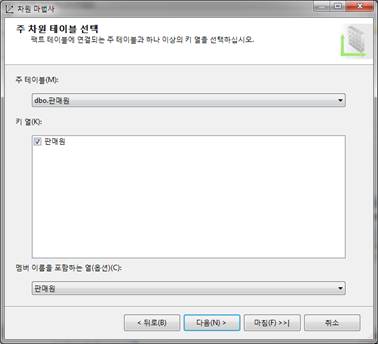
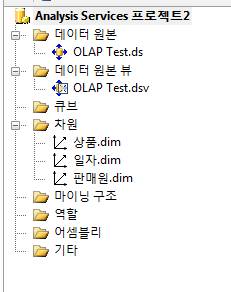
3.     데이터 원본 뷰를 생성합니다.



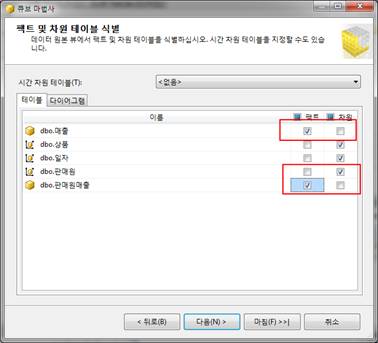
4.     상품 차원을 생성합니다.

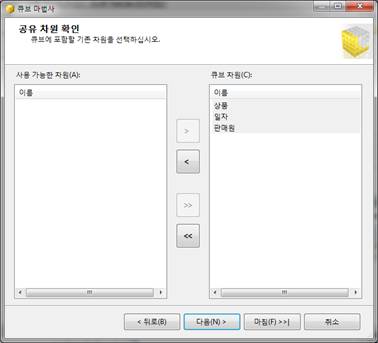
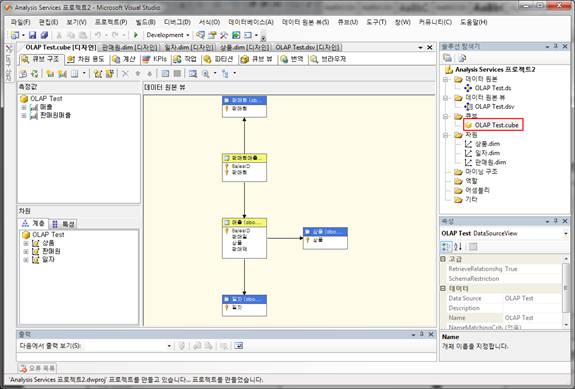
  
계층 등은 생성하지 않아도 됩니다.

5.     동일한 방법으로 일자와 판매원 차원을 생성합니다.

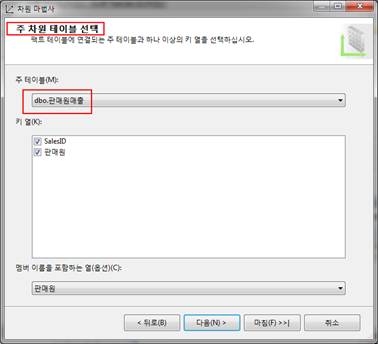
  
  
   
  


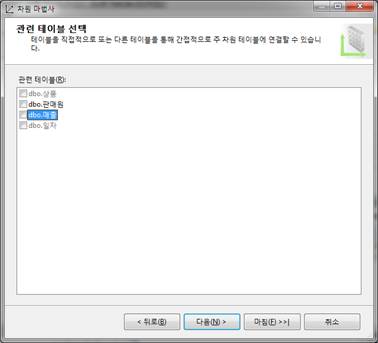
6.     큐브를 생성합니다.



7.     **판매원매출** 테이블을 이용해서 **판매원매출이라는 차원**을 만듭니다. (a 단계)

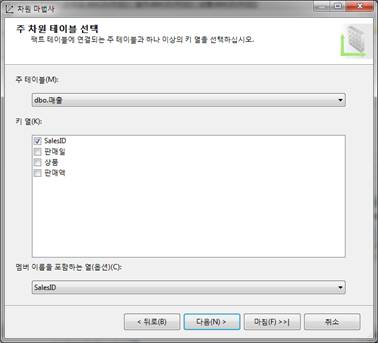
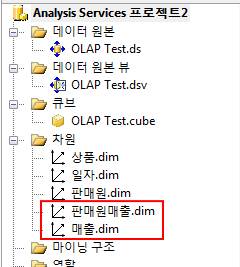
  
  
관련 테이블에서 모든 체크를 제거합니다.



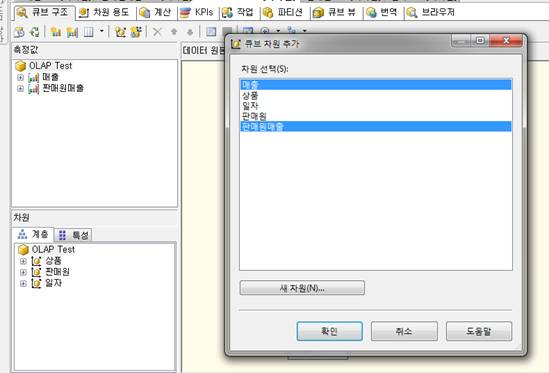
차원 특성에서는 기본 설정대로 모든 항목을 선택합니다.



8.     동일한 방법으로 **매출 테이블을 이용해서 매출 차원**을 생성합니다.

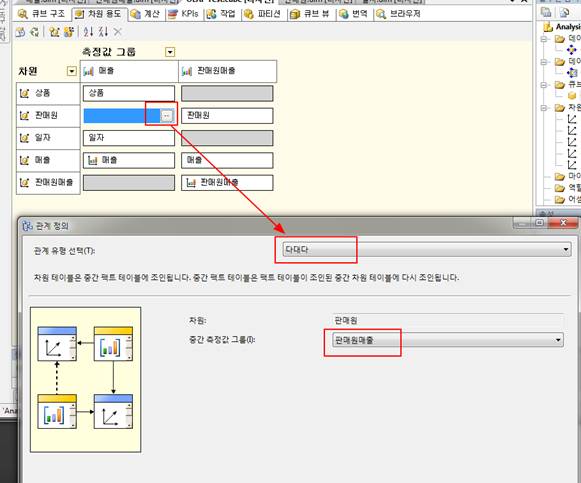
  
  


9.     큐브에서 다음과 같이 {매출, 판매원매출} 차원을 추가합니다.



10.   큐브의 **차원 용도** 탭을 선택합니다.

11.   아래와 같이 판매원과 매출 그룹 관계를 설정합니다.





12.   배포 및 처리 후 데이터를 조회해 봅니다.

