

Microsoft®  
**SQL Server™ 2005**

**Analysis Services를 이용한 실시간  
비즈니스 인텔리전스**

## 요약

---

본 백서는 오늘날 기업들이 직면하고 있는 실시간 비즈니스 인텔리전스 데이터를 수집하고 분석하는 데 있어 많은 장애 요소들을 해결할 수 있는 SQL Server 2005 Analysis Services의 주요 특징 및 기능에 대해 설명합니다.

본 문서는 예비 문서이며 여기에서 설명된 소프트웨어의 최종 상용 버전이 출시되기 전에 상당 부분 변경될 수 있습니다. 이 문서에 포함된 정보는 문서 발행 시에 논의된 문제들에 대한 Microsoft Corporation의 당시 관점을 나타냅니다. Microsoft는 변화하는 시장 상황에 부응해야 하므로 이를 Microsoft 측의 공약으로 해석해서는 안되며 발행일 이후 소개된 어떠한 정보에 대해서도 Microsoft는 그 정확성을 보증하지 않습니다. 이 문서는 오직 정보를 제공하기 위한 것입니다. Microsoft는 이 설명서에서 어떠한 명시적이거나 묵시적인 보증도 하지 않습니다. 해당 저작권법을 준수하는 것은 사용자의 책임입니다. 저작권에서의 권리와는 별도로, 이 설명서의 어떠한 부분도 Microsoft의 명시적인 서면 승인 없이는 어떠한 형식이나 수단(전기적, 기계적, 복사기에 의한 복사, 디스크 복사 또는 다른 방법) 또는 목적으로도 복제되거나, 검색 시스템에 저장 또는 도입되거나, 전송될 수 없습니다. Microsoft가 이 설명서 본안에 관련된 특허권, 상표권, 저작권 또는 기타 지적 재산권 등을 보유할 수도 있습니다. 서면 사용권 계약에 따라 Microsoft로부터 귀하에게 명시적으로 제공된 권리 이외에, 이 설명서의 제공은 귀하에게 이러한 특허권, 상표권, 저작권 또는 기타 지적 재산권 등에 대한 어떠한 사용권도 허여하지 않습니다.

© 2005 Microsoft Corporation. 전원 보유.

Microsoft는 미국 및 기타 국가에서 Microsoft Corporation의 등록 상표입니다. 여기에 인용된 실제 회사와 제품 이름은 해당 소유자의 상표일 수 있습니다.

# Contents

---

|                                      |    |
|--------------------------------------|----|
| 서론.....                              | 3  |
| 실시간 BI 확보의 장애 요소.....                | 3  |
| 장애 요인 해결 .....                       | 5  |
| Analysis Services로 데이터 전송 .....      | 6  |
| 일관된 데이터 읽기 .....                     | 6  |
| 이기종 데이터 소스 결합.....                   | 7  |
| 캐싱 정책 정의 .....                       | 7  |
| 알림 스키마.....                          | 9  |
| 증분 프로세싱.....                         | 9  |
| Analysis Services로부터 알림 메시지 수신 ..... | 10 |
| 결론 .....                             | 10 |
| 부록 .....                             | 11 |
| 사전 캐싱 설정 .....                       | 11 |
| 사전 캐싱을 사용한 증분 프로세싱 .....             | 12 |



# 서론

비즈니스 인텔리전스(BI: Business Intelligence) 시스템의 목표는 더욱 정확하고 풍부한 정보를 활용하여 신속하고 효과적인 의사결정을 내릴 수 있도록 한다는 것입니다. 따라서 실시간 비즈니스 인텔리전스와 관련한 모든 논의는 이러한 의사결정을 지원할 정보를 얼마나 실시간에 가깝게 확보할 수 있는지의 맥락에서 이루어져야 합니다. 실시간 정보를 구성하는 요소들은 단일 기업 내에서조차도 다양한 비즈니스 활동에 따라 매우 다양합니다. 아울렛 유통망을 예로 들어 보겠습니다.

- 익월 판매를 예측하기 위해 이전 데이터를 사용하는 분석가들은 지난 달의 정보만을 필요로 할 수도 있습니다.
- 캠페인의 성공도를 평가하는 것은 물론, 캠페인 진행 기간을 결정해야 하는 마케팅 관리자는 일일 단위의 보다 최신 정보를 필요로 할 것입니다.
- 매장 관리자는 예를 들어 부패하기 쉬운 식품을 판매할 때 하루에도 여러 차례 의사 결정을 내려야 할 수 있습니다. 이 관리자는 일 단위가 아닌 시간 단위의 정보를 필요로 할 것입니다.

많은 경우 최신 정보와 이전 정보를 적절히 함께 사용해야 합니다. 예를 들어 매장 관리자는 오늘 오전 중의 특정 물품 판매 개수를 파악하는 것도 유용하지만 지난 해 같은 날 동일 품목에 대한 평균 판매 개수와 비교해 이 정보를 사용한다면 더욱 효과적일 것입니다.

보고서에는 이기종 시스템에서 입수된 데이터가 포함될 수 있습니다. 유통 매장에서 당일 매장 판매 데이터는 판매 시점에 바로 업데이트 되는 소스 시스템에서 입수되어 엔터프라이즈 데이터 웨어하우스의 이전 데이터와 결합됩니다. 그러나 이는 더 이상 "1개 버전의 사실 데이터"를 보유하고 있지 않다는 것을 의미합니다. 2개 소스 시스템에서 입수된 데이터를 비교하는 경우 데이터 웨어하우스 로드 중에 동일한 비즈니스 정책을 적용하도록 많은 주의를 기울여야 합니다. 대표적인 정책으로 적용된 통합, 할인율 처리 및 환전 방식 등을 들 수 있습니다. 이는 DW 로드 프로세스에 구현된 비즈니스 규칙이 2개 시스템의 데이터를 결합한 보고서 또는 응용 프로그램에도 구현되어야 한다는 것을 의미합니다. 이는 시스템의 기민성을 떨어뜨리고 오류 발생률을 높일 수 있습니다. 뿐만 아니라 궁극적으로 부정확한 정보를 바탕으로 의사결정을 내리게 될 수도 있습니다.

가장 이상적인 것은 서로 다른 상황의 모든 의사 결정자를 지원할 수 있을 만큼 충분히 실시간에 가까운 데이터를 제공하는 단일 소스를 보유하는 것입니다. 그렇다면 비즈니스 인텔리전스 시스템에서 실시간 데이터를 제공하는 것을 가로막는 요인은 무엇일까요?

## 실시간 BI 확보의 장애 요소

실시간 BI 데이터를 입수하는 데 어려움을 가중시키는 많은 심각한 문제들이 있으며 이들 중 일부는 기술적인 접근방식만으로는 해결할 수 없습니다. 많은 EII(Enterprise Information Interchange) 기술과 관련한 표현들이 문제의 심각성을 호도하고 있기 때문에 이 점에 유념해야 할 것입니다.

실시간 데이터 확보의 장애 요소들은 다음과 같습니다.

- 이전 데이터를 볼 수 있어야 합니다. 종종 원래의 소스 시스템이 이전 데이터를 전혀 제공하지 않는 경우가 있기 때문에 전체 기록을 포함한 별도의 데이터 사본을 DW에 보관해야 합니다.
- 시스템 전반에서 비즈니스 프로세스와 조정해야 합니다. 특정 비즈니스 프로세스가 완료되기 전까지는 특정 분석을 수행하는 것이 의미가 없습니다. 앞서 제시한 유통 체인의 사례에서 보면 모든 시간대에 전매장이 마감할 때까지 매장별 비교를 수행하는 것이 무의미합니다. 마찬가지로 일부 레거시 시스템들이 업데이트되는 메커니즘은 달성 가능한 대기시간의 한계치 보다 조금 낮게 설정될 수 있습니다.
- 데이터 품질을 보장하기 위해서는 데이터 변환 프로세스를 자주 실행해야 합니다. 이러한 변환의 복잡성과 데이터 볼륨에 따라 로드 시간을 특정 수준 이하로 낮추는 것은 비용 효율적인 방법이 아닐 수 있습니다.
- 변환 프로세스를 통해 일괄적으로 처리되어 개별 스토어를 생성하는 중요한 변환이 없는 경우에도 종종 여러 데이터 소스의 데이터를 통합해야 합니다.
- 만족스러운 사용자 성능을 제공한다는 목표는 실시간 목표와 상충될 수 있습니다. 소스 시스템에서 직접 보고한다면 분명히 실시간 정보를 제공하게 되지만 일반적으로 공통의 집계 레벨 쿼리는 빈번하게 업데이트가 이루어지는 시스템에서는 제대로 수행되지 않습니다. 뿐만 아니라 트랜잭션 시스템에 이와 같은 쿼리 로드를 부가하는 것을 허용할 수도 없습니다. 따라서 사전에 계산된 집계(Analysis Services 사용과 같이)를 제공하는 데 많은 주의를 기울이면서 개별 매장을 관리해야 합니다. 소스 데이터를 업데이트할 때마다 이들 집계를 유지하는 데 많은 시간이 소요됩니다.
- 계속 업데이트되는 시스템의 데이터에 대해 지속적으로 보고하기 위해서는 데이터 일관성의 문제를 고려해야 합니다.
- 보고서의 데이터가 충분히 최신 데이터인 경우에도 해당 보고서가 의사 결정자들이 필요로 할 때 즉시 제공된다는 것을 어떻게 보장할 수 있을까요?

이와 같은 고려사항은 그림 1에서 제시된 바와 같이 사용자가 종종 원래 소스 데이터로부터 멀리 떨어져 있다는 것을 의미합니다. 체인 내 각 링크는 추가 지연 시간을 나타냅니다.

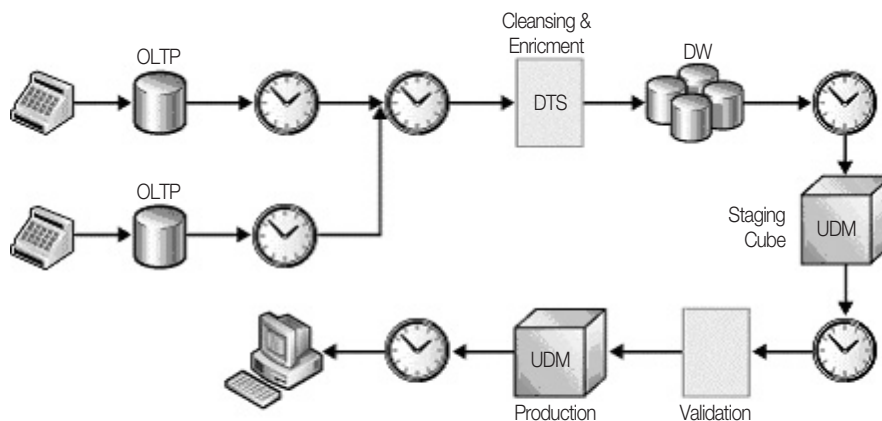


그림 1: 실시간 BI의 장애 요소

각 매장의 소스 트랜잭션 시스템은 POS 레지스터에서 즉시 업데이트됩니다. 매일 모든 매장의 데이터가 통합되며 광범위한 변환 작업이 적용되어 DW로 로드되기 전에 데이터를 정리합니다. 일단 로드되면 모든 사용자 액세스를 위해 사용되는 Analysis Services 큐브를 다시 처리하는 프로세스가 실행됩니다. 큐브가 스테이징 서버에서 처리되고 검증 작업이 완료되면 사용자 보고용으로 사용되는 프로덕션 서버로 전송됩니다. 때때로 큐브를 새로 고친 다음, 해당 관리자가 큐브를 기초로 보고서를 실행하고 변경된 정보를 확인한 후 최종 작업을 개시하게 됩니다.

## 장애 요인 해결

일부 장애 요소들은 기술적으로 처리하기 어렵지만 SQL Server Analysis Services는 물론, 최종 사용자에게 보다 신속하게 정보를 제공하도록 돕는 SQL Server 제품의 여타 구성요소들은 많은 특징과 기능을 갖추고 있습니다.

- Analysis Services로 데이터를 전달. Analysis Services에 변경 사항들을 업데이트하는 작업은 Analysis Services로 데이터를 직접 전달하는 SSIS(SQL Server Integration Services) 전환 기능이 추가되면서 한결 쉬어졌습니다. 이는 데이터를 추출하여 정리하는 동일한 작업을 통해 큐브에 대한 새로 고침을 직접 실행할 수 있다는 것을 의미합니다.
- 이기종 데이터 소스 결합. 이기종 소스의 데이터에서 Analysis Services 큐브를 만들 수 있습니다. 데이터는 Analysis Services에 의해 결합되어 일원화된 방식으로 사용자에게 제공됩니다.
- 일관된 데이터 읽기. SQL Server 2005 관계형 엔진은 추가 트랜잭션 분리 레벨을 포함하여 독자들이 데이터의 스냅샷 뷰를 획득할 수 있도록 지원합니다. 이는 Analysis Services에 의해 활용되기 때문에 프로세스에 많은 시간이 소요되는 경우에도 일관된 데이터 뷰가 큐를 통해 나타나도록 큐브가 처리될 수 있습니다.
- Analysis Services 큐브의 증분 업데이트. Analysis Services 2000에서와 마찬가지로 큐브의 작은 부분에만 실제 영향이 미치도록 업데이트 영향을 제한할 수 있는 파티션을 사용하면 Analysis Services 큐브 새로 고침에 소요되는 시간을 단축할 수 있습니다. 뿐만 아니라 변경 데이터에 대한 큐브 처리가 가능하기 때문에 새로운 팩트 행을 기존 캐시에 추가할 수도 있습니다. Analysis Services 2005는 차원을 증분 업데이트 할 수 있도록 개발되어 새로운 차원 레코드가 추가되지만 할 뿐 업데이트 되지 않는 경우도 지원할 수 있습니다.
- 캐싱 정책 정의. Analysis Services 2005는 소스 데이터의 변경을 반영하도록 큐브 새로 고침 방법을 정의할 수 있습니다. 이들 사전 캐싱 정책을 통해 최신 데이터에 대한 비즈니스 요구와 고성능 요구의 균형을 유지할 수 있습니다.
- Analysis Services로부터 알림 메시지 입수. Notification Services와 통합하면 관심 있는 데이터의 변경 시 알림 메시지가 전달되도록 신청할 수 있습니다. 예를 들어 관리자는 특정 KPI(Key Performance Indicator)에 관심 등록을 하면 KPI 상태가 변경될 때마다 전자 메일로 알림 메시지를 받게 됩니다.

이들 기술을 독립적으로 사용하거나 함께 사용하여 훨씬 손쉽게 실시간 BI를 제공할 수 있습니다. 일부 간단한 사례에 적용할 수 있는 단순한 아키텍처의 사례가 아래 제시되어 있으며 이 사례에서는 Analysis Services 큐브가 소스 데이터베이스에서 직접 생성됩니다.

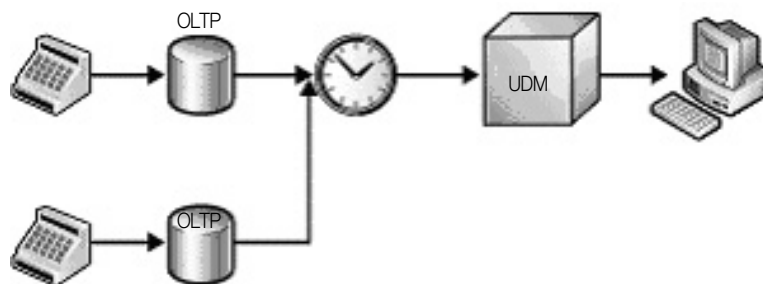


그림 2: 단순한 아키텍처

### 참고

이처럼 간단한 사례에서도 주로 실제 소스 OLTP 데이터베이스의 복제본인 데이터베이스 위에서 큐브를 생성하는 것이 더욱 실용적인 방법입니다. 이러한 복제본을 유지 관리하는 데 드는 비용은 트랜잭션 시스템의 추가 오버헤드를 줄여 얻게되는 가치보다 더 큽니다.

각 매장의 소스 트랜잭션 시스템은 POS 레지스터에서 즉시 업데이트됩니다. 사전 캐싱 정책으로 매 20분마다 변경분에 대한 새로 고침을 실행하여 큐브에 새로운 데이터가 반영되도록 보장합니다. 사용자가 상태가 변경된 KPI에 대한 알림을 수신하도록 신청한 경우 관련 보고서로 연결되는 URL이 포함된 전자 메일 메시지가 직접 전송됩니다.

본 백서의 나머지 부분에서는 Analysis Services 2005의 새로운 사항을 위주로 이러한 기능에 대해 자세히 다루고 있습니다.

## Analysis Services로 데이터 전송

SQL Server 2005에서 SSIS는 파이프라인 작업을 개시하여 데이터를 읽고 일련의 변환 작업을 거쳐 변환한 다음, 그 결과를 몇몇 수신지로 출력할 수 있습니다. Analysis Services는 가능한 수신지 중 하나로 포함되어 변환된 데이터가 큐브로 직접 전송되도록 합니다. 이는 파티션과 차원 모두의 로드를 처리하며 증분 로드할 수 있는 기능도 포함하고 있습니다. Analysis Services 2005에서 새롭게 선보인 차원은 증분 로드될 수 있습니다. type II 차원과 마찬가지로 차원 열이 추가되지만 하고 업데이트는 되지 않는 경우에 특히 유용합니다.

## 일관된 데이터 읽기

SQL Server 2005에서 스냅샷 분리로 알려진 새로운 트랜잭션 분리 레벨이 처음으로 소개되었습니다. 이러한 트랜잭션을 시작하는 고객들은 다른 사용자에 의해 업데이트가 커밋되고 있어도, 트랜잭션 과정 중에 읽은 모든 데이터가 트랜잭션 시작 시의 상태와 동일하도록 보장받을 수 있습니다.

큐브에서 변경되는 데이터를 처리할 때 다른 사용자의 변경 사항과 분리되지 않는다면 문제가 발생할 수 있습니다. 예를 들어 제품 차원이 처리된 다음 파티션 처리가 완료되기 전에 새로운 제품이 입력되고 해당 제품의 판매가 이루어질 것입니다. 그러면 파티션 처리에서 판매가 선택되어 명백한 참조 무결성 오류가 야기됩니다.

Analysis Services 2005에서는 개발자의 재량에 따라 스냅샷 분리를 이용할 수 있습니다. 데이터 소스에 이러한 정책 세트가 있다면 Analysis Services는 처리 작업을 시작할 때 스냅샷 트랜잭션을 개시하며 모든 객체는 동일한 스냅샷에서 처리됩니다. 스냅샷을 입수하는 데 대규모의 고정 오버헤드가 요구되지 않습니다. 그러나 트랜잭션 과정 중에 수행되는 모든 업데이트에 대한 약간의 오버헤드로 인해 어느 정도 비용이 발생합니다.



## 이기종 데이터 소스 결합

Analysis Services의 큐브는 여러 데이터 소스에서 공급될 수 있습니다. 예를 들어 제품 및 판매 데이터는 주요 엔터프라이즈 DW에서 가져오고 건적은 소형 부서별 데이터베이스에서 가져올 수 있습니다. Analysis Services는 이기종 소스 전반의 데이터를 조인하는 데 필요한 쿼리를 발행합니다.

Analysis Services에서 SQL Server의 분산 쿼리 프로세서를 활용하는 경우 데이터 소스 중 하나는 SQL Server가 될 것입니다. SQL Server 데이터베이스가 반드시 데이터를 포함할 필요는 없습니다. 예를 들어 2개 Oracle 데이터베이스의 데이터가 제3의 SQL Server 데이터베이스의 쿼리 프로세서를 사용하여 함께 조인될 수도 있습니다.

## 캐싱 정책 정의

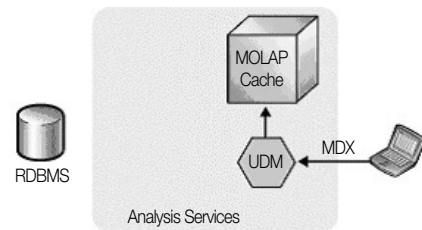
Analysis Services는 기반 데이터의 캐시를 분석 쿼리에 최적화된 형식으로 유지하고 종종 사전 계산된 총계도 포함함으로써 높은 쿼리 성능을 제공합니다. 이러한 캐시로 인해 다음과 같은 많은 중요한 질문들이 제기되고 있습니다.

- 기반 데이터가 변경되어 캐시에 지난 데이터가 포함되어 있는 경우 어떻게 해야 하는가?
- 캐시에 대한 새로 고침을 얼마나 자주 실행해야 하는가? 캐시 새로 고침을 실행하는 도중에 어떻게 쿼리에 응답해야 하는가? 기반 소스로 되돌아 가지 않고 지난 캐시를 이용한다면 최신 데이터를 입수할 수는 있지만 쿼리 성능이 크게 떨어질 수 있다.
- 소스 데이터에 변경이 이루어졌다는 것을 어떻게 알 수 있는가? 얼마나 자주 확인해야 하는가?

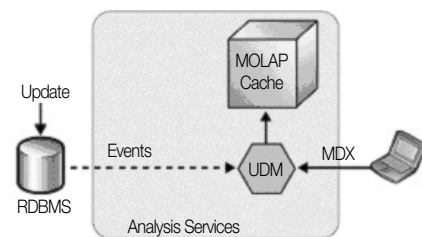
새로운 Analysis Services 2005의 사전 캐싱 기능은 큐브 디자이너가 실시간 데이터와 성능에 대한 비즈니스 요구의 균형을 이룰 수 있도록 정책을 정의하는 방법을 제공합니다. 시스템은 명시적인 프로세싱을 요구하지 않고 "자동 조종" 모드로 실행됩니다.

정책 설정을 통해 다양한 시스템 운영을 지원할 수 있습니다. 다음 예에서는 소스 데이터에 대한 업데이트가 완료되는 즉시 큐브 MOLAP(Multidimensional OLAP) 캐시의 새로 고침을 실행되도록 정책을 지정한 큐브를 보여줍니다. 또한 캐시 새로 고침이 단기간 내에 완료되지 않으면 최신 데이터에 대한 강력한 요구 때문에 시스템이 ROLAP(Relational OLAP) 모드로 되돌아가 캐시 새로 고침이 완료될 때까지 기반 데이터베이스로 직접 쿼리를 전송하게 됩니다.

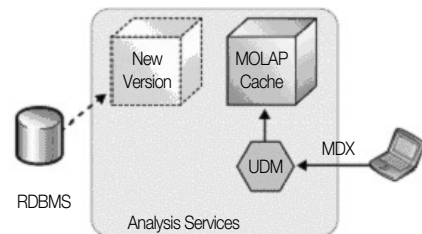
1. 큐브는 관계형 데이터베이스의 데이터에서 처리됩니다. MOLAP 캐시를 사용하여 큐브에 대한 모든 사용자 쿼리가 충족됩니다.



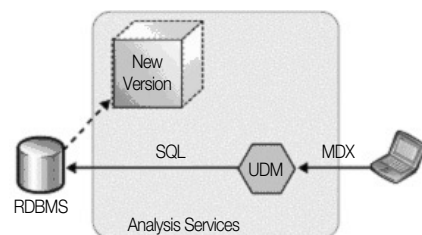
2. 기반 데이터베이스에 업데이트가 수행됩니다. 추후 이벤트를 통해 Analysis Services에 통지됩니다.



3. 새로운 MOLAP 캐시가 재구축되도록 프로세싱이 시작됩니다. 처음, 새로운 캐시가 구축되는 동안에는 사용자 쿼리에 대한 응답은 이전의 쓸모 없어진 MOLAP 캐시를 사용하여 이루어집니다.



4. 시간이 지나도 새로운 MOLAP 캐시는 아직 완료되지 않은 상태입니다. 정책은 최대 허용 가능한 대기시간을 지정합니다. 해당 한계에 도달하면 시스템은 ROLAP 모드로 전환되고 모든 새로운 사용자 요청은 기반 관계형 소스에 대해 최신 데이터를 검색하도록 하는 SQL 쿼리를 발행함으로써 충족될 것입니다.



5. 궁극적으로, 새로운 MOLAP 캐시가 구축되면, 이제 시스템은 원래의 MOLAP 모드로 전환됩니다.

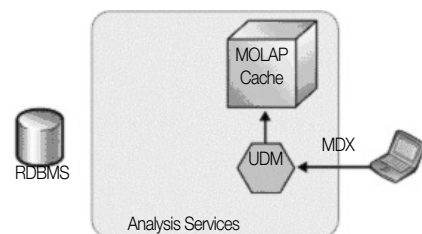


그림 3: 사전 캐싱을 이용한 동작 예제



여러 다양한 정책으로 매우 다른 동작을 수행할 수 있습니다. 다음과 같은 예를 들 수 있습니다.

- **자동 MOLAP(Automatic MOLAP)**, 그림 3에서 제시된 바와 같이 동작은 일관된 고성능에 대한 요구가 실시간 데이터에 대한 요구보다 크기 때문에 ROLAP 모드로 되돌아 가지 않습니다.
- **계획된 MOLAP(Scheduled MOLAP)**, 큐브는 소스 데이터의 업데이트에 관계없이 주기적으로 새로 고침을 실행합니다.
- **점증식 MOLAP(Incremental MOLAP)**, 소스 데이터베이스에 새로운 데이터가 추가될 때마다 큐브가 변경분에 대한 새로 고침을 실행합니다.
- **실시간 ROLAP(Real-time MOLAP)**, 큐브가 항상 ROLAP 모드에 있기 때문에 소스 데이터에 대한 모든 업데이트가 쿼리 결과에 즉시 반영됩니다.

개별 Analysis Services 객체에 각각 다른 사전 캐싱 설정이 이루어질 수 있기 때문에 큐브 내 여러 부분에 유연하게 정책을 적용하고 성능과 대기시간에 대한 우선 순위를 부여할 수 있습니다.

사전 캐싱 설정은 부록에서 더욱 자세하게 설명되어 있습니다.

## 알림 스키마

Analysis Services에 기반 소스의 업데이트를 알리는 다음과 같은 3가지 알림 스키마가 제공되고 있습니다.

- **추적 이벤트**, 이는 기반 데이터 소스가 SQL Server인 경우에만 해당됩니다. Analysis Services는 필수 테이블에 대한 추적 이벤트를 수신하도록 등록합니다. Analysis Services의 서비스 계정이 SQL Server 데이터베이스 상의 관리 권한을 보유해야만 이 스키마를 사용할 수 있습니다. 업데이트 시점에 데이터베이스를 사용할 수 없다면, 이벤트 전달이 보장되지 않습니다.
- **클라이언트 초기화**, 클라이언트는 지정된 테이블이 업데이트됐다는 것을 알리는 알림 메시지를 Analysis Services로 전송할 수 있습니다. 이는 실제로 업데이트를 수행 중인 애플리케이션이 Analysis Services 큐브의 영향을 인식하는 경우에 효과적입니다.
- **폴링**, 가장 일반적으로 활용할 수 있는 방법은 폴링을 사용하는 것입니다. 폴링 쿼리는 각 테이블에 대해 정의되기 때문에 변경이 수행되어 테이블이 업데이트되었음을 나타내는 단일 값(예: 하나의 열과 함께 단일 행)을 반환합니다. 예를 들어 LastUpdated 타임스탬프 열의 최대 값을 반환하는 쿼리가 적합한 폴링 쿼리라고 할 수 있습니다. 디자이너는 이러한 쿼리와 함께 Analysis Services가 폴링 쿼리를 전송해야 하는 간격을 지정합니다.

## 점증식 프로세싱

일반적으로 캐시의 새로 고침을 실행할 때 영향 받는 Analysis Services 개체를 완벽하게 처리하여 캐시가 완벽하게 재로드 되도록 합니다. 보다 정확성을 기하기 위해 파티션에서는 Full 옵션을 처리하고 차원에서는 Update 프로세서 옵션을 사용합니다. 이들 옵션으로 관련 파티션 데이터가 보존됩니다. 그러나 행의 추가만 이루어지고 업데이트는 되지 않는 경우 파티션과 차원 모두에서 증분 프로세싱을 수행할 수도 있습니다.

이 기능은 폴링 알림 스키마를 사용할 때 이용할 수 있습니다. 업데이트가 이루어졌음을 알리는 폴링 쿼리와 함께, 디자이너는 새로운 행들을 반환하는 증분 프로세싱 쿼리도 제공합니다. 이 쿼리는 폴링 쿼리에 의해 반환될 때 이전 값과 새 값을 가져오도록 매개 변수화로 지정됩니다. 부록에서 예제가 제공됩니다.

## Analysis Services로부터 알림 메시지 수신

Notification Services는 SQL Server에서 새로 선 보이는 서비스로서 클라이언트가 이벤트(예: 쿼리 변경 값)에 관심 분야를 등록하고 이벤트가 발생하는 경우에 알림 메시지를 수신하는 방식(예: 전자 메일 또는 인스턴트 메신저)을 정의할 수 있는 확장 가능한 인프라를 제공합니다. 그 다음 이를 지원하기 위해 Notification Services는 알림 메시지 전달을 보장하고 전환이 이루어질 때 오직 한 번만 알림이 발생하도록 하는 것과 같은 표준 서비스를 제공합니다. 예를 들어 KPI가 OK에서 Bad로 변경되는 경우 사용자는 한 차례 메시지를 받게 되며 KPI가 Bad 상태로 남아 있더라도 반복해서 알림 메시지를 받지 않습니다.

SQL Server 2005의 경우, Notification Services에 Analysis Services를 위한 특별 어댑터가 포함되어 있습니다. 이 어댑터를 사용하여 고객은 폴링 주기, 알림 방법 등과 함께 관심 데이터를 정의하는 MDX 쿼리를 지정함으로써 이벤트에 가입할 수 있습니다. 예를 들어 매장 관리자는 특정 제품에 대한 시간별 판매가 지정된 수준 이하로 떨어질 경우에 알림 메시지를 요청할 수 있습니다.

이를 통해 지정된 이벤트가 발생하면 사용자에게 신속하게 알리는 BI 애플리케이션을 쉽게 구축할 수 있습니다.

## 결론

여타 SQL Server 2005 제품과 함께 Analysis Services 2005는 실시간 BI 데이터를 사용할 수 있도록 돕는 많은 기능을 제공합니다. 대표적으로 다음과 같은 기능을 들 수 있습니다.

- 캐싱 정책을 정의한 다음 총계를 명시적으로 다시 처리하지 않고도 시스템을 '자동 조종(auto-pilot)' 모드로 실행하고 보다 간편하게 총계 세트를 유지 관리할 수 있습니다.
- 변환에서 새 레코드의 변경분 추가에 이르기까지 직접 Analysis Services를 로드함으로써 효율성이 크게 향상됩니다.
- 스냅샷 분리 기능으로 데이터가 변경되는 동안에도 일관된 뷰를 확보할 수 있습니다.
- 여러 소스의 데이터를 결합할 수 있는 Analysis Services 큐브의 기능을 이용해 여러 이기종 소스에서 입수되는 데이터를 쉽게 통합할 수 있습니다.
- Notification Services를 통해 주요 이벤트를 사용자에게 알릴 수 있습니다.

완벽한 실시간 BI 데이터를 확보하는 데 많은 장벽이 존재하지만, 이러한 기능을 독립적으로 사용하거나 함께 사용하여 보다 신속하고 효과적인 비즈니스 의사결정을 지원할 수 있을 만큼 충분히 실시간에 가까운 데이터를 확보할 수 있습니다.

문서 번호, SQL-200509-16

## 부록

부록에서는 사전 캐싱 설정에 대한 추가적인 설명과 함께 사전 캐싱을 사용하여 큐브를 변경분에 대해 새로 고칠 수 있는 폴링 쿼리 및 증분 프로세싱 쿼리의 사용에 대해 자세히 설명합니다.

### 사전 캐싱 설정

| 설정                                                                 | 설명                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 사일런스 간격(Silence Interval) 및 사일런스 무효화 간격(Silence Override Interval) | 일반적으로 새로운 캐시가 처리되는 도중에 새 업데이트가 발생하면 처리 작업이 취소되고 다시 시작하게 됩니다. 그렇기 때문에 기반 데이터베이스에 대한 업데이트가 일괄적으로 수행된다면 배치에서 첫 번째 업데이트가 수행되는 즉시 캐시 새로 고침을 시작하는 것은 효율적이지 못할 수 있습니다. 사일런스 간격(Silence Interval)은 새로 고침을 시작하기 전에 추가 업데이트 없이 경과되어야 하는 적절한 시간을 지정합니다. 사일런스 무효화 간격(Silence Override Interval)은 경과되어야 하는 시간의 상한을 지정합니다. 이 기간이 경과하면 지체없이 새로 고침이 시작됩니다. 사일런스 간격(Silence Interval)을 '무한'으로 설정하면 이 작업은 업데이트 이벤트가 아닌 Force Rebuild Interval 설정에 의해 처리됩니다. |
| Force Rebuild Interval                                             | Force Rebuild Interval이 지정되어 있지 않고 새로운 캐시가 처리되는 동안 또 다른 업데이트가 수행되면, 프로세싱이 취소되고 다시 시작하게 됩니다. Force Rebuild Interval은 마지막 캐시의 새로 고침이 실행된 이후의 시간 길이를 지정합니다. 여기에서 이후의 다른 업데이트에 관계없이 새로운 캐시의 새로 고침이 시작되고 완료됩니다. 따라서 연속 업데이트에서 계속 다시 시작하는 것을 방지할 수 있습니다. Force Rebuild Interval은 일반적으로 무한대의 사일런스 간격(Silence Interval)과 함께 사용되어 주기적으로 캐시 새로 고침을 실행하게 됩니다.                                                                                        |
| 대기 시간                                                              | 대기 시간 설정은 ROLAP 모드로 되돌아 가기 전에 캐시를 얼마나 오랫동안 새로 고치지 않고 유지할 수 있는지를 지정합니다. 종종 일관된 고성능이 필요한 경우에 이 설정은 '무한대'로 설정될 수 있으며 이는 쿼리가 항상 MOLAP 캐시를 사용할 것임을 의미합니다.                                                                                                                                                                                                                                                                                      |
| 온라인 모드                                                             | 온라인 모드는 처음 캐시가 작성되기 전에도 처음 프로세싱 동안 큐브가 ROLAP 모드에서 즉시 온라인 상태로 전환될 수 있는지 여부를 제어합니다.                                                                                                                                                                                                                                                                                                                                                         |
| 총계 스토리지                                                            | 일반적으로 ROLAP 큐브의 경우, 총계는 기반 데이터베이스 내에서 작성되는 구체화된 뷰로 나타납니다. 이 설정은 MOLAP 큐브가 ROLAP로 되돌아 가는 경우에도 이러한 뷰가 생성되는지 여부를 제어합니다.                                                                                                                                                                                                                                                                                                                      |

## 사전 캐싱을 사용한 증분 프로세싱

아래 테이블에서는 팩트 테이블 내에 새로운 판매 행이 존재하며 파티션으로 이들 행이 점증적으로 추가되는지를 탐지하는 데 사용할 수 있는 폴링 쿼리와 이와 관련된 증분 프로세싱 쿼리의 예를 보여줍니다.

|            |                                                                                                                                                                                                                                                                      |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 폴링 쿼리      | SELECT max(LastUpdated) from Sales                                                                                                                                                                                                                                   |
| 증분 프로세싱 쿼리 | SELECT * from Sales WHERE<br>LastUpdated > coalesce(?, -1) AND<br>LastUpdated <= ?.<br>주: 시작 시 이전 값이 Null이기 때문에 'coalesce'만 필요로 합니다.                                                                                                                                 |
| 예제         | 마지막 폴링 쿼리에서 , max(LastUpdated)는 3/18/2005 1pm입니다.<br>다음 폴링 쿼리에서 이 값은 3/18/2005 1.15 pm으로 변경됩니다.<br>Analysis Services에서는 새로운 행을 처리하도록 다음 쿼리를 전송합니다.<br>SELECT * from Sales WHERE<br>LastUpdated > coalesce(3/18/2005 1pm, -1) AND<br>LastUpdated <= 3/18/2005 1.15 pm |

문서 번호, SQL-200509-16



**Microsoft®**

**한국마이크로소프트(유)**

서울특별시 강남구 대치동 892번지 포스코센터 | 전화 : 080-985-2000 | 인터넷 : [www.microsoft.com/korea](http://www.microsoft.com/korea)