대용량 데이터베이스를 위한 성능 데이터 모델링에 관한 연구
Modeling on Data Performance for Very Large Database

이 종 석* · 이 창 호*

Abstract

데이터는 살아 움직이고 있다. 10년 전만 해도 10GB 정도의 데이터라면 대용량 데이터라고 불리던 시절이 있었다. 하지만 지금은 10TB보다 큰 데이터베이스도 흔하다. 결국, 대용량 데이터베이스(VLDB)의 시대가 개막된 것이다.

VLDB로 변한 데이터베이스에는 백업, 복구, 관리와 같은 문제점이 있지만 그 중에서도 성능 문제를 빼놓을 수 없다. 데이터베이스에 많은 데이터가 있고 그렇게 많은 데이터 중에서 필요한 몇 건의 데이터만 추출하는 것이 쉬운 일이 아니다. 과거에는 데이터가 적었기 때문에 이러한 것이 큰 문제가 아니었지만 이제는 VLDB가 되면서 성능 최적화는 일상적이고도 중요한 이슈가 되었다. 따라서 VLDB가 된 데이터베이스나 VLDB로 변하고 있는 데이터베이스에서 성능 관리를 하고 최적화할 수 있는 전문 기술이 필요하다.

Keywords: Data Modeling, VLDB, EPCglobal Network.

1. 서 론

데이터의 용량이 커질수록 기업의 의사결정의 속도가 빨라질수록 데이터를 처리하는 속도는 빠르게 처리되어야 할 필요성을 반증해 준다. 일반적으로 실무 프로젝트에서 보면 잘못된 테이블 디자인 위에서 개발된 어플리케이션의 성능이 저하되는 경우, 개발자가 구축한 SQL 구문에 대해서만 책망을 하는 경우가 많이 있다. 물론 개발자가 SQL 구문을 잘못 구성하여 성능이 저하되는 경우도 있지만 근본적으로 디자인이 잘못되어 SQL 구문을 잘못 작성하도록 구성될 수밖에 없는 경우도 빈번하게 발생되고 있다.[5][10][16][18][20]

* 남서울대학교 산업경영공학과
* 인하대학교 산업공학과

성능이 저하되는 데이터 모델은 크게 세 가지 경우로 나눌 수 있다. 데이터 모델 구조에 의해 성능이 저하될 수도 있고 데이터가 대용량으로 됨으로 인해 불가피하게 성능이 저하되어 나타나는 경우도 있다. 또한 인덱스 특성을 충분히 고려하지 않고 인덱스를 생성함으로 인해 성능이 저하되어 나타나는 경우도 있다.[5][16][20]

본 논문에서는 대용량 데이터의 효율적인 관리, 처리 및 유지를 위해 데이터 모델의 성능을 향상시킬 수 있는 방안을 조사 연구하고자 한다.

2. 이론적 배경

2.1 성능 데이터 모델링

일반적으로 성능이라고 하면 데이터 조회의 성능을 의미하곤 한다. 그 이유는 데이터 입력, 수정, 삭제는 일시적이고 빈번하지 않고 단건 처리가 많은 반면 데이터 조회의 경우는 반복적이고 빈번하며 여러 건을 처리하는 경우가 많기 때문이다. 이러한 특징은 일반적인 트랜잭션의 성격이 조회의 패턴을 가지고 있다는 것이고 업무에 따라서는 입력, 수정, 삭제의 성능이 중요한 경우도 있다.[20] 데이터 모델링을 할 때 어떤 작업 유형에 따라 성능 향상을 도모해야 하는지 목표를 분명하게 해야 정확한 성능 향상 모델링을 할 수 있다.[16][18]

성능 데이터 모델링이란 데이터베이스 성능 향상을 목적으로 설계단계의 데이터 모델링 때부터 정규화, 반정규화, 테이블통합, 테이블분할, 조인구조, PK, FK 등 여러 가지 성능과 관련된 사항이 데이터 모델링에 반영될 수 있도록 하는 것으로 정의할 수 있다.[5][16]

정규화된 모델이 데이터를 주요 관심사별로 분산시키는 효과가 있기 때문에 그 자체로 성능을 향상시키는 효과가 있다. 각각의 엔티티에 대한 용량산정을 수행하면 어떤 엔티티에 데이터가 집중되는지 파악할 수 있다. 또한 데이터 모델에 발생되는 트랜잭션의 유형을 파악할 필요가 있다. 트랜잭션의 유형을 파악하게 되면 SQL 문장의 조인관계 테이블에서 데이터 조회의 컬럼들을 파악할 수 있게 되어 그에 따라 성능을 고려한 데이터 모델을 설계할 수 있다.

이렇게 파악된 용량산정과 트랜잭션의 유형 데이터를 근거로 정확하게 테이블에 대해 반정규화를 적용하도록 한다. 반정규화는 테이블, 속성, 관계에 대해 포괄적인 반정규화의 방법을 적용해야 한다. 또한 대량 데이터가 처리되는 이력모델에 대해 성능 고려를 하고 PK/FK의 순서가 인덱스 특성에 따라 성능에 영향을 미치는 영향도가 크기 때문에 반드시 PK/FK를 성능이 우수한 순서대로 컬럼의 순서를 조정해야 한다.

2.2 정규화

정규화는 다양한 유형의 검사를 통해 데이터 모델을 좀 더 구조화하고 개선시켜 나가는 절차에 관련된 이론이다. 정규화의 기본 원칙은 하나의 테이블에는 중복된 데이터가 없도록 하는 것이다.[20]

현재 정규화 이론은 1NF, 2NF, 3NF, BCNF, 4NF, 5NF으로 6단계까지 있다. 그러나 일반적으로 3단계까지만 사용한다.[11]

2.3 반정규화

반정규화는 정규화된 엔티티 타입, 속성, 관계를 시스템의 성능 향상, 개발과 운영을 단순화하기 위해 데이터 모델을 통합하는 프로세스를 말한다. 반정규화를 할 때 가장 중요하게 검토해야 할 기준은 각각의 엔티티 타입과 속성, 관계에 대해 데이터의 정합성과 데이터의 무결성을 우선으로 할지 데이터베이스 구성의 단순화와 성능을 우선으로 할지에 달려있다.[20]

대부분의 업무에서는 정확한 데이터의 관리가 중요한 관건이다. 그래서 데이터의 정합성과 무결성을 보장할 수 있는 정규화가 기본적인 전제라 할 수 있다. 그러나 테이블의 복잡성과 시스템의 성능을 고려하지 않을 수 없으므로 기본적으로는 정규화한 테이블을 그대로 유지하는 것을 목표로 하고, 문제가 되는 테이블에 대해서 뷰의 생성, 파티셔닝 테이블 생성, 인덱스 조정, 클러스터링 적용 등 여러 가지 방안을 먼저 조사하도록 한다. 그 다음 반정규화를 고려한다.

반정규화를 무분별하게 진행하면 데이터의 무결성이 깨져 추적이 불가하거나 정합성을 맞출 수 없는 경우가 발생할 수 있다. 따라서 일정한 기준을 선정하여 반정규화를 할 대상을 선정하는 작업을 해야 한다. 반정규화를 하기로 선정된 테이블, 컬럼, 관계에 대해서는 반드시 지속적인 추적 관리를 해야 한다.

3. 성능 향상 전략

3.1 정규화를 통한 성능 향상

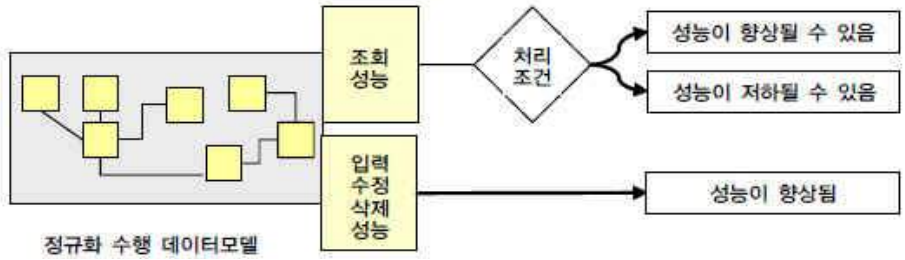
데이터베이스에서 데이터를 처리할 때 성능이라고 하면 조회 성능과 입력, 수정, 삭제 성능의 두 분류로 구분된다. 이 두 가지 성능이 모두 우수하면 좋겠지만 데이터 모델을 구성하는 방식에 따라 두 성능이 Trade-Off 되어 나타나는 경우가 많이 있다.

정규화를 수행한다는 것은 데이터를 결정하는 결정자에 의해 함수적 종속을 가지고 있는 일반 속성을 의존자로 하여 입력, 수정, 삭제 이상을 제거하는 것이다. 데이터의 중복 속성을 제거하고 결정자에 의해 동일한 의미의 일반 속성이 하나의 테이블로 집약되므로 한 테이블의 데이터 용량이 최소화되는 효과가 있다. 따라서 정규화된 테이블은 데이터를 처리할 때 속도가 빨라질 수도 있고 느려질 수도 있는 특성이 있다.

일반적으로 정규화가 잘 되어 있으면 입력, 수정, 삭제의 성능이 향상되고 반정규화

를 많이 하면 조회의 성능이 향상된다고 인식될 수 있다. 그러나 반정규화된 테이블에 대해 입력, 수정, 삭제, 조회 성능이 저하되는 일이 빈번하며, 조회 성능을 올리고자 하는 당초의 목적과 달리 데이터의 중복 탓으로 조회 성능이 더 저하되는 경우도 종종 발생한다. 또한 데이터가 중복되니 데이터를 입력하거나 수정할 때 일관성 있게 처리되지 못하여 데이터의 무결성이 깨지기도 한다.

따라서 데이터 모델링을 할 때 반정규화만이 조회 성능을 향상시킨다는 고정관념은 탈피되어야 한다. 정규화가 적용된 데이터 모델이 설계되어야 데이터 처리의 입력, 수정, 삭제, 조회의 성능이 원활하고 데이터 무결성도 보존될 수 있다. 꼭 필요한 반정규화를 제외한 나머지 테이블에 대해서는 정규화의 형태로 유도하여 데이터 처리의 성능과 데이터 무결성을 보장하는 품질이 우수한 모델을 디자인해야 한다.



[그림 1] 정규화 수행과 성능의 관계

3.2 반정규화를 통한 성능 향상

데이터 무결성이 깨질 수 있는 위험을 무릅쓰고 데이터를 중복하여 반정규화를 적용하는 이유는 데이터를 조회할 때 디스크 I/O량이 많아서 성능이 저하되거나 경로가 너무 멀어 조인으로 인한 성능 저하가 예상되거나 컬럼을 계산하여 읽을 때 성능이 저하될 것이 예상되는 경우 반정규화를 수행하게 된다.

반정규화에 대한 필요성이 결정이 되면 컬럼의 반정규화뿐만 아니라 테이블의 반정규화와 관계의 반정규화를 종합적으로 고려하여 적용해야 한다. 또한 반정규화를 막연하게 중복을 유도하는 것만을 수행하기 보다는 성능을 향상시킬 수 있는 다른 방법들을 고려하고 그 이후에 반정규화를 적용하도록 해야 한다. 반정규화를 적용할 때는 기본적으로 데이터 무결성이 깨질 가능성이 많이 때문에 반드시 데이터 무결성을 보장할 수 있는 방법을 고려한 이후에 반정규화를 적용하도록 해야 한다.

[그림 2] 중복성의 원리를 활용한 성능 향상

3.3 데이터 모델 단순화를 통한 성능 향상

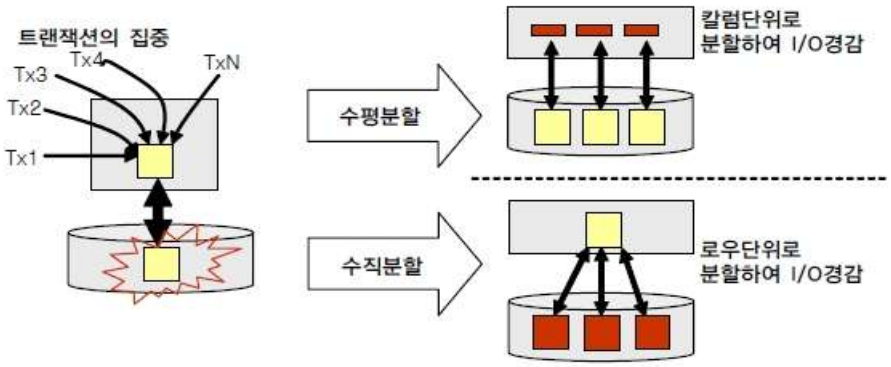
각 모델간의 관계가 연결되어 있지 않고, 통합되어야 할 엔티티 타입이 몇 개로 분리가 되어 있으며, 어떤 엔티티 타입은 과도하게 통합되어 하나의 엔티티 타입으로 표현되어 있으면 이는 전체적으로 품질이 낮은 데이터 모델이 되는 것이다. 이 데이터 모델이 물리적인 테이블로 생성되면 SQL 구문이 길어지고 잦은 조인이 발생할뿐더러 SQL 구문을 한번 실행해서는 명확한 결론이 나오지 않아 두 번 이상 SQL 문장을 실행해야 하는 경우도 생긴다. 결과적으로 낮은 품질로 인한 데이터 모델로 인해 복잡한 SQL 구문이 작성이 되었으며 성능이 저하된 SQL 구문이 작성되게 된 것이다.

따라서 업무 흐름을 정확하게 구별할 수 있는 데이터 모델이 생성되어야 업무 흐름에 맞게 관계가 연결되어 정보 추적성을 보장하게 되고 데이터 무결성도 보장할 수 있고, 엔티티 타입이 적절하게 통합되어 SQL 문장에서 테이블 중복으로 인해 발생하는 불필요한 성능 저하 현상을 예방할 수 있다.

3.4 테이블 수평/수직 분할에 의한 성능 향상

한 테이블에 데이터가 대량으로 집중되거나 하나의 테이블에 여러 개의 컬럼이 존재하여 디스크에 많은 블록을 점유하면 성능 저하를 불러올 수 있다. 하나의 테이블에 대량의 데이터가 존재하면 인덱스의 Tree 구조가 너무 커져 효율성이 떨어져 데이터를 처리(입력, 수정, 삭제, 조회)할 때 디스크 I/O를 많이 유발한다. 또한 한 테이블에 많은 수의 컬럼이 존재하면 데이터가 디스크의 여러 블록에 존재하기 때문에 디스크에서 데이터를 읽는 I/O량이 많아지게 되어 성능이 저하된다.

많은 컬럼을 가지고 있는 테이블에 대해서는 트랜잭션이 발생할 때 어떤 컬럼에 대해 집중적으로 발생하는지 분석하여 테이블을 쪼개면 디스크 I/O가 감소하게 되어 성능이 개선된다. 즉 트랜잭션을 분석하여 적절하게 1:1 관계로 분리함으로써 성능을 향상시킨다.



[그림 3] 테이블 수평/수직 분할에 의한 성능 향상

테이블에 많은 양의 데이터가 예상될 경우 파티셔닝을 적용하거나 PK에 의해 테이블을 분할할 수 있다. 가장 많이 사용하는 파티셔닝의 기준이 범위 파티션이다. 대상 테이블이 날자 또는 숫자 값으로 분리 가능하고, 각 영역별로 트랜잭션이 분리된다면 범위 파티션을 적용한다. 지점, 사업소, 사업장, 핵심적인 코드 값 등으로 PK가 구성되어 있고 대량의 데이터가 있는 테이블이라면 각각의 값에 의해 파티셔닝이 되는 리스트 파티션을 적용할 수 있다. 리스트 파티션은 대용량 데이터를 특정 값에 따라 분리 저장할 수는 있으나 범위 파티션과 같이 데이터 보관주기에 따라 쉽게 삭제하는 기능은 제공되지 않는다.

데이터양이 대용량이 되면 파티셔닝의 적용은 필수적이다. 파티셔닝 기준을 나눌 수 있는 조건에 따라 적절한 파티셔닝 방법을 선택하여 성능을 향상시키도록 한다.

3.5 PK 순서 조정을 통한 성능 향상

PK가 여러 개의 속성으로 구성된 복합식별자 일 때 PK순서에 대해 별로 고려하지 않고 데이터 모델링을 한 경우에 성능 저하 현상이 많이 발생한다. PK 순서를 결정하는 기준은 인덱스 정렬 구조를 이해한 상태에서 인덱스를 효율적으로 이용할 수 있도록 PK 순서를 지정해야 한다. 즉 인덱스의 특징은 여러 개의 속성이 하나의 인덱스로 구성되어 있을 때 앞쪽에 위치한 속성의 값이 비교자로 있어야 인덱스가 좋은 효율을 나타낼 수 있다. 앞쪽에 위치한 속성 값이 가급적 '=' 또는 최소한 범위 'BETWEEN' '< >'가 들어와야 인덱스를 이용할 수 있다.

3.6 FK 인덱스 생성을 통한 성능 향상

FK 인덱스를 적절하게 설계하여 구축하지 않았을 경우 개발 초기에는 데이터양이 얼마 되지 않아 성능 저하가 나타나지 않다가 시스템을 오픈하고 데이터양이 누적될수록 데이터베이스 서버에 심각한 장애 현상을 초래할 수 있다. 그러므로 물리적인 테이블에 FK 제약을 걸었을 때는 반드시 FK 인덱스를 생성하도록 하고, FK제약이 걸

리지 않았을 경우에는 FK 인덱스를 생성하는 것을 기본 정책으로 하되 발생되는 트랜잭션에 의해 거의 활용되지 않았을 때에만 FK 인덱스를 지워야 한다.

4. 결론

정보의 홍수 속에서 범람하는 유효한 데이터를 체계적으로 형상화한다는 것은 쉽지 않다. 이런 상황에서 데이터의 관리는 많은 조직체에서 가장 중요한 작업 중의 하나가 되어 왔고 앞으로도 더 가중될 추세이다. 처리해야 할 정보가 증가함에 따라 데이터의 효율을 최대로 하기 위해 데이터를 어떻게 구성해야 하는가라는 과제는 이미 매우 중요한 문제로 대두되었다.[3]

데이터베이스 응용 프로그램은 원하는 답만 나온다고 해서 역할을 다한 것이 아니며, 적절한 시간 내에 결과를 제공할 수 있어야 하고 다른 프로세스의 작업도 방해하지 않아야 한다. 이를 위해서는 하드웨어, DBMS, 네트워크 등의 모든 영역들이 적절하게 구성되어 있어야 하며, 응용 프로그램 또한 최적화되어 있어야만 한다.[23]

본 논문에서는 VLDB가 된 데이터베이스나 VLDB로 변하고 있는 데이터베이스에서 성능 관리를 하고 성능을 최적화할 수 있는 방법을 DBMS의 관점에서 여러 가지로 알아보았다. 이는 유비쿼터스 시대에 대량으로 발생하게 될 데이터에 대한 관리는 물론 EPCglobal Network를 활용하는 다양한 산업분야에 적용될 수 있으며, 개별 물류센터나 기업에서 물류정보를 효율적으로 관리하는데 활용함은 물론 SCM 전반에 도입하여 시너지 효과를 창출할 수 있을 것으로 기대된다.

5. 참 고 문 헌

[1] 그림으로 배우는 MS-SQL 2000 서버, 홍릉과학출판사, 박정용.
[2] 관계형 데이터 모델링, 오픈메이드, 김기창.
[3] 대용량 데이터베이스솔루션 I, 엔코아, 이화식.
[4] 대용량 데이터베이스솔루션 II, 엔코아, 이화식.
[5] 데이터베이스 설계와 구축: 성능까지 고려한 데이터 모델링, 한빛미디어, 이춘식.
[6] 데이터베이스 튜닝, 브레인코리아, 최용락.
[7] 실무 사례로 다지는 고성능 데이터베이스 튜닝, 비팬북스, 권순용.
[8] 실행 계획으로 배우는 고성능 데이터베이스 튜닝, 비팬북스, 권순용.
[9] 새로 쓴 대용량 데이터베이스 솔루션 I, ㈜엔코아컨설팅, 이화식.
[10] 아는 만큼 보이는 데이터베이스 설계와 구축, 한빛미디어, 이춘식.
[11] 알기 쉽게 해설한 데이터베이스 모델링, 프리렉, 김연홍.
[12] 이종석, 이창호, “시뮬레이션을 이용한 EPCIS의 효율화 방안에 관한 연구”, 대한안전경영과학회지, 제12권 제4호, 2010.
[13] 이창호, 조용철, “EPCIS Event 데이터 크기의 정량적 모델링에 관한 연구”, 대한

안전경영과학회지, 제11권 제4호, 2009.
[14] 전문가를 위한 데이터 모델링 실무, 브레인코리아, 최용락.
[15] 하루 10분씩 핵심만 골라 마스터하는 SQL 핸드북, 정보문화사, 최현호.
[16] SQL 전문가 가이드, 한국데이터베이스진흥원.
[17] SQL Server 2000 성능 튜닝, 정보문화사, 하성희.
[18] http://www.dbguide.net
[19] http://www.infomaster.co.kr
[20] http://itmore.tistory.com

저 자 소 개

이 종 석 : 현재 남서울대학교 산업경영공학과 교수로 재직 중. 인하대학교 산업공학과 공학박사 취득. 주요 연구 관심분야는 EPCglobal Network, RFID를 활용한 응용시스템, 시뮬레이션, SCM, DB 등.
주　　소 : 충남 천안시 성환읍 매주리 21 남서울대학교 산업경영공학과

이 창 호 : 현재 인하대학교 산업공학과 교수로 재직 중. 인하대학교 산업공학과 공학사, 한국과학기술원 산업공학과 공학석사, 한국과학기술원 경영과학과 공학박사 취득. 주요 연구 관심분야는 RFID를 활용한 항공물류 정보시스템, 인천항 물류관리, 항공 산업 관련 스케줄링과 중소기업의 ERP 개발 등.
주　　소 : 인천광역시 남구 용현동 253 인하대학교 산업공학과