

# 조회 속도 향상을 고려한 손익 평가 시스템 구현

김 솔미, 곽 희규, 김 수형  
전남대학교 전산학과  
e-mail:dawool@lycos.co.kr

## The Loss and Profit System For View Performance

Sol-Moi Kim, Hee-Kue Kwag, Soo-Hyung Kim  
Department of Computer Science, Chonnam National University

### 요약

정보의 양이 많아지면서 대용량의 자료에서 필요한 자료를 찾는 일의 중요성이 날로 증대되고 있다. 본 논문에서는 100만건 이상의 손익 자료에서 시계열성 매출액, 사원별 매출액 등과 같은 정보의 조회 시 조회 속도를 향상시키기 위한 여러 방법들을 고려하여 손익 평가 시스템을 구축하고 앞으로의 발전 방향을 알아본다.

### 1. 서론

정보의 홍수 속에 살고 있는 우리는 많은 자료 속에서 유용하고 가치 있는 정보를 빠른 응답시간내에 찾기를 희망한다. 사용자의 요구 또한 단순한 자료의 요약에서 정보의 새로운 가치를 창출하려는 방향으로 흘러가고 있다. 시스템에 저장되는 자료 또한 매년 2배 가까이 증가되고 있다. 이제 자료는 모으는 일보다 검색하는 일이 더 중요하게 된 것이다.

손익 평가 시스템은 사원별 매출 목표와 실적을 관리하는 시스템이다. 유통회사의 손익 평가 시스템의 경우 전국에 체인망이 있어 동시 접속하는 단말기도 많고 동시 접속 시간대 또한 유사하다. 시계열과 같은 연간 자료를 조회할 경우 기존의 시스템에서는 각 일별 매출액 테이블에서 자료를 읽어 계산하거나 월별 요약을 구한 후 조회하여야 하기 때문에 자료의 건수에 많은 영향을 받는다. 시계열과 같이 연간 자료를 조회하는 경우 5분 이상 기다려야 결과가 나왔다. 이러한 문제점을 개선하고자 조회 속도 향상을 위한 다양한 방법에 대해 알아보자.

본 논문에서는 많은 사용자의 접속으로 가중되는 서버의 부담을 줄이기 위해 3-tier 계층으로 개발되었다. 클라이언트 계층에서는 사용자의 인터페이스를 담당하고 개발자들의 개발도구인 PowerBuilder 7.0을 사용하였고, 미들웨어(middle ware)는 자료의 송수신과 처리 조건을 서버에 옮겨주고 처리결과 및

검색된 자료를 클라이언트로 보내주는 TUXEDO를 사용하였다. 서버는 자료를 검색하고 읽고 쓰는 C, Pro\*C, 자료의 저장소인 DB가 존재한다.

본 논문의 구성은 다음과 같다. 2장에서는 속도 개선을 위한 여러 방법(테이블의 활용, 인덱스의 활용, 자료 송수신량의 개선, DB의 메인 메모리 상주 등)을 살펴보고 3장에서는 제시된 방법에 의해 구현된 손익 평가 시스템을 소개하고 4장에서는 이를 요약하고 추후 발전 방향에 대해 논의한다.

### 2. 조회 속도 개선을 위한 방법

조회 속도 개선을 위한 방법으로는 자료량을 조절하는 테이블의 활용, 인덱스의 활용, 자료 송수신량의 개선, DB의 메인 메모리 상주 등이 있다. 이러한 방법들에 대해 구체적으로 살펴보자.

#### 2.1 테이블의 활용

테이블이 가지고 있는 자료의 양에 따라 검색 속도에 많은 차이가 있다. 테이블의 크기가 커지면 검색 속도가 현저하게 저하되기 때문에 테이블의 크기를 조절할 필요가 있다. 이처럼 테이블의 크기를 적당하게 만들기 위한 방법을 살펴보면 테이블의 크기를 분할하는 방법으로 수평적 분할, 수직적 분할 기법이 있으며, 복잡하게 계산된 조회를 위해 계산된 칼럼으로 테이블을 따로 생성하는 방법으로 전체요약, 최소차원 테이블

블 기법 등이 있다.

- 수평적 분할 기법(horizontal partitioning) : 연간 단위나 월간단위로 테이블을 분할하는 방법이다.
- 수직적 분할 기법(vertical partitioning) : 테이블의 한 행(row)이 차지하는 크기가 클 때 하나의 테이블을 2개 이상으로 분할하는 기법이다. 빈번히 발생하는 항목과 그렇지 않은 항목을 분리해서 분할한다.
- 전체 요약(full summary) : 전체 요약이란 모든 조회 가능한 자료를 미리 계산하여 요약 테이블을 만들어 놓는 것을 말한다.
- 희소 요약(sparse summary) : 전체 요약은 많은 디스크 양을 필요로 하고 또한 계산에 따른 잦은 자료의 입출력이 필요해서 오류의 빈도수를 높인다. 만약 연별 매출액을 조회할 경우, 연간 자료까지 요약한 테이블을 두는 것이 아니라 월별 매출액 테이블에서 매출 실적을 충분히 만들어 낼 수 있다. 이처럼 사용빈도수가 낮은 자료는 따로 만들지 않고 필요한 항목만으로 테이블을 만드는 경우를 말한다. 희소 요약 테이블의 크기는 자료의 1/10 이하로 만드는 것을 기준으로 한다.

### 2.3 인덱스의 활용

테이블 내의 자료를 어떻게 읽어오는지에 따라 검색 속도에 현저한 차이가 있다. 인덱스를 사용하지 않고 자료를 읽으면 전체 테이블에서 원하는 자료를 하나씩 찾기 때문에 검색 속도가 현저하게 느려진다. 이 절에서는 생성된 인덱스를 사용할 수 없는 경우와 조인시의 고려사항을 살펴보면 다음과 같다.

- 인덱스를 사용할 수 없는 경우 : 인덱스 칼럼의 변형이 일어난 경우, 부정형(not, <>)으로 조건을 기술한 경우, 인덱스 칼럼이 NULL인 경우, 인덱스 칼럼에 함수를 사용하는 경우
- 조인시 Select 된 자료가 적은 테이블을 우선 조건으로 하여야 한다.
- 여러개의 인덱스가 있는 상태에서 동시에 많은 양의 데이터가 입력, 수정, 삭제된다면 많은 부하가 발생한다.

### 2.3 송수신시 자료량의 개선

3-tier 특성상 미들웨어를 통해 한번에 서버에서 클라이언트로 운반할 자료의 길이가 정해져 있다. 동시에 여러 사용자가 접속할 경우 자료를 입출력하는데 많은 시간을 필요로 하기 때문에 입출력 시간을 줄이기 위해 서버-미들웨어-클라이언트의 자료 운반을 한번만 하는 방법이다.

### 2.4 DB의 메인 메모리 상주

자료의 입출력시간을 최소화하기 위한 방법으로 DB의 메인 메모리 상주 방법을 활용하는데 크게 2가지로 나눌 수 있다. 메인 메모리의 버퍼 크기를 확장하는 방법과 전체 DB를 주기억장치에 상주시키는 방법이 있다.

- 버퍼 크기를 확장하는 방법 : RDBMS를 확장하는 접근방식으로 주기억장치 버퍼를 크게 확장함으로써 디스크 입출력 횟수를 줄여 성능을 향상시키는 방법이다.
- 주기억장치에 상주시키는 방법 : 전체 DB를 주기억장치에 구축/운영함으로써 디스크 입출력으로 인한 부담을 없애 고성능 실시간 요구조건을 맞추주는 방식이다. 하지만 주기억장치가 휘발성이어서 시스템 오류시 자료의 손실이 우려된다.

### 3. 손익 시스템 구현

손익 시스템이란 기존의 회계 시스템과 연계하여 지점별, 부서별, 사원별, 제품별 손익 평가 자료를 제공하는 시스템이다. 이는 경영의 합리화와 효율화, 기업의 이익 극대화를 위한 기초 자료 및 판단 자료로서 모든 사원이 열람 할 수 있다. 또한 매출에 대한 정확한 손익 판단을 위해 순매출, 매출이익, 목표 대비 매출이익, 부서별 사원별 계정별 실적 등의 자료를 조회 할 수 있도록 한다. 손익 시스템은 일마감과 월마감의 마감시점에서 하위계정에 대한 상위계정을 생성하여야 한다. 따라서 하나의 하위계정에 대한 상위계정이 3~4개정도의 단계를 가지기 때문에 마감 후 보통 3~4배정도로 자료량이 증가하게 된다.

조회 속도 향상을 위한 손익 평가 시스템의 구성도를 보면 다음과 같다.

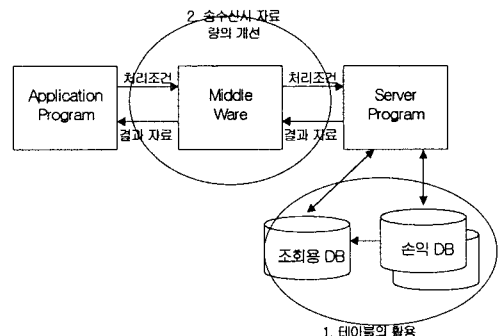


그림 1. 손익 평가 시스템 구성도

그림 3. 시스템의 매출 테이블 분할

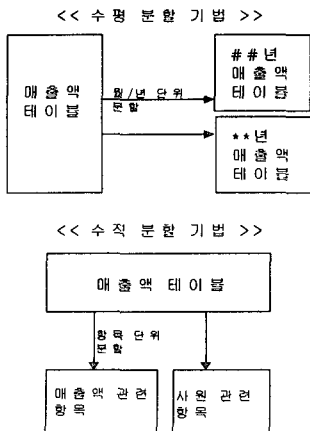
3.1 테이블의 활용

일반적인 손익 평가 시스템의 경우 손익 DB의 매출액 테이블에서 일 매출액의 자료를 읽어 일별 합계를 계산한 다음 이를 조회하면 (그림 2)에 보여준다. 그런데 사원별 매출액의 경우 사원 테이블과 계정테이블, 매출액 관련 테이블을 조인하여야 한다. 조인은 조회 속도 감소의 주요 원인이 되므로 일별 매출 합계를 가진 조회용 테이블을 새로 만든다. 조인 없이 이를 조회하면 속도를 향상 시킬 수 있다.

월	매출액	사원번호	매출액	사원번호	매출액	사원번호
01월	17,283	0	0	17,283	17,283	1000 00
02월	5,793	0	0	5,793	5,793	1000 00
03월	7,118	0	0	7,118	7,118	1000 00
04월	13,108	0	0	13,108	13,108	1000 00
05월	5,974	0	0	5,974	5,974	1000 00
06월	13,808	0	0	13,808	13,808	1000 00
07월	0	0	0	0	0	0 00
08월	7,209	0	0	7,209	7,209	1000 00
09월	6,335	0	0	6,335	6,335	1000 00
10월	5,070	0	0	5,070	5,070	1000 00
11월	315	0	0	315	315	1000 00
12월	4,239	0	0	4,239	4,239	1000 00
합계	83,588	0	0	83,588	83,588	1000 00

그림 2. 테이블 활용의 조회 화면

기존에 일별 매출액 테이블의 행의 수는 1,083,600 × 평균 일별 매출건수(14개월 × 43개 부서 × 60계정 × 30일 × 평균 일별 매출건수)건이었으나 일별 계정의 합계를 내어 월별로 실적일부서 조회용 테이블을 생성하였다. 이때는 일별생성 자료가 없고 월별 합계로 저장되기 때문에 행 수가 36,120건(14개월 × 43개 부서 × 60계정)으로 조회 속도가 현저하게 향상됨을 알 수 있다.



3.2 송수신시 자료량의 개선

응용 프로그램에서 처리조건을 서버에 보낼 때 미들웨어는 중간 다리 역할을 한다. 미들웨어가 한번에 운반할 수 있는 양은 버퍼의 크기처럼 정해져 있다. 기존의 시스템에서는 조회버튼을 누르면 검색 조건에 맞는 자료를 읽어 서버-미들웨어-클라이언트를 반복해서 왕복하면서 자료를 운반하고 사용자의 화면에 뿌려준다. 그러나 본 손익 시스템에서는 한번 운반 가능한 자료의 양만큼만 사용자의 화면으로 보내고 사용자의 요청이 있는 경우(화면의 계속 버튼을 클릭하는 경우)만 다음 결과를 검색하여 운반하는 방법이다. 기존의 시스템은 검색조건에 의한 자료의 건수에 상관없이 조회 가능하였으나 운반 횟수를 1회로 규정하면 조회 속도는 향상되지만 사용자가 원하는 자료를 한번에 모두 볼 수 없는 단점이 있다.

월	매출액	사원번호	매출액	사원번호	매출액	사원번호
01월	17,283	0	0	17,283	17,283	1000 00
02월	5,793	0	0	5,793	5,793	1000 00
03월	7,118	0	0	7,118	7,118	1000 00
04월	13,108	0	0	13,108	13,108	1000 00
05월	5,974	0	0	5,974	5,974	1000 00
06월	13,808	0	0	13,808	13,808	1000 00
07월	0	0	0	0	0	0 00
08월	7,209	0	0	7,209	7,209	1000 00
09월	6,335	0	0	6,335	6,335	1000 00
10월	5,070	0	0	5,070	5,070	1000 00
11월	315	0	0	315	315	1000 00
12월	4,239	0	0	4,239	4,239	1000 00
합계	83,588	0	0	83,588	83,588	1000 00

그림 4. 송수신시 자료량의 개선 예

3.3 인덱스의 활용

인덱스의 활용은 서버 프로그램(C, Pro\* C)에서 대부분 처리된다. 우선 각 프로그램마다 검색시 필요한 조건을 기록한다. 조건들을 살펴보면 한 테이블에 필요한 인덱스들을 알 수 있다. 모든 테이블에 검색시 사용되는 인덱스를 설정한다고 해서 검색 시간이 단축되는 것은 아니다. 인덱스의 개수가 5개 이상이면 검색 시간 단축의 효과를 기대하기 힘들다.

그림 4의 조회 화면을 검색할 경우를 살펴보자. '30213100'보다 큰 부서코드와 명을 부서코드순으로 알고자 할 경우

```

select Null(부서코드, '**'), Null(부서명, '**')
from 부서코드테이블
where Null(부서코드, '**') > '30213100'
order by 부서코드;
  
```

라고 하면 부서코드가 Primary Key라 할지라도 Null 명령어 때문에 인덱스를 활용할 수 없다. 따라서

```
select distinct 부서코드, 부서명
      from 부서코드테이블
      where 부서코드 > '30213100'
      order by 부서코드;
```

로 바꾸어 부서코드와 부서명을 찾아야한다. 물론 Null 값이 들어있는 부서코드는 일지 못할 것이다.

### 3. 결론

본 논문에서는 대용량의 자료에 대한 조회 속도 향상을 고려한 손익 평가 시스템을 구현하였다. 사용자의 요구는 점점 다양해지고 끊임없이 변화하며, 개발자가 끊임없이 변화하는 사용자의 요구에 맞게 처리하는 일보다 사용자가 직접 원하는 자료를 검색하기를 원한다. 이러한 변화에 맞추어 사용자가 직접 자료를 읽고 처리하는 일을 쉽게 할 수 있는 툴이나 방법들(OLAP : On-line Analytical Processing, DW : DataWare Housing, DM : Data Mining)이 등장하고 경영자들의 의사결정을 위한 예측자료나 패턴분석(DM : Data Mining)까지 하기에 이르렀다. 이에 대한 발전이 대용량의 데이터의 활용에 일조할 것이다. 또한 이에 대한 활발한 연구가 이어져야 한다.

### 참고문헌

- [1] Eun-Teong Kim, Hwan-Seung Yong, "Design of Multidimensional Data Model for OLAP Based on Object-Relational DBMS", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.18, No.11, November 1996.
- [2] Pieter Adrians, Dolf Zantinge, "Data Mining", Addison-Wesley 1996.
- [3] 박 성진, "데이터웨어하우스", 마이크로소프트, pp. 246-263 1999년 8월호
- [4] 이 재환, "웹과 DB연동 튜닝 테크닉 총정리", 마이크로소프트, pp. 280-285 1998년 5월호
- [5] 이 은정, "OLAP를 위한 데이터 큐브에서의 효과적인 Max/Min 영역 질의 처리 기법", 한국과학기술원 석사학위 논문, 1997.
- [6] 강현구, 백종명, 손주찬, 박성진, "ERP 개발 도구의 설계 및 구현", 한국과학기술원 석사학위 논문, 1997.
- [7] 정광철, "메인 메모리 상주 DBMS의 개발 방향과 이용 방안", 전자신문, 2000.08.22
- [8] 이화식, "대용량 데이터베이스 솔루션 I", 엔코아, 1996.

- [9] 장동인, "실무자를 위한 데이터웨어하우스", 대청, 1998.