

# 데이터 웨어하우스 성능 관리를 위한 DBMax의 확장

김 은 주<sup>†</sup> · 용 환 승<sup>††</sup> · 이 상 원<sup>††</sup>

## 요 약

데이터베이스의 사용이 증가하고 다루어야 하는 데이터의 양이 방대해지면서 데이터베이스 시스템을 효율적으로 사용하기 위한 성능 관리 기술이 중요해지고 있다. 특히 데이터 웨어하우스는 대용량의 데이터를 대상으로 복잡한 분석을 수행하거나 전략적 의사 결정을 위해 사용하기 때문에 신속한 질의 응답을 위한 성능 관리가 무엇보다 중요하다. 데이터 웨어하우스는 일반 운영계 시스템과는 다른 목적과 특성을 가지기 때문에 그에 적합한 성능 모니터링 방법 및 튜닝 기술이 필요하다. 본 연구에서는 OLTP(On-Line Transaction Processing)용 오라클 데이터베이스를 위한 국산 성능 관리 도구인 DBMax를 데이터 웨어하우스 환경에 적합하도록 기능을 확장한다. 우선 데이터 웨어하우스 응용 분야를 지원하기 위한 오라클 9i의 요약 관리(summary management) 기능과 ETL(Data Extraction, Transformation and Loading) 기능을 중심으로 데이터 웨어하우스 시스템의 성능 관리시 고려해야 할 요구 사항을 분석하고 이를 지원하는 DBMax의 확장 아키텍처를 설계 및 구현한다. 구체적으로 요약 관리와 ETL 작업을 지원하기 위한 오라클 9i의 다양한 스키마 객체에 대한 정보와 성능 관련 지표를 제시하여 데이터 웨어하우스 환경에서 수행되는 질의에 대한 SQL 튜닝 기능을 강화한다. 또한 사후 분석을 위한 DBMax의 로그 파일에서 의미 있는 SQL 문을 추출하여 잠재적으로 유용한 실제화된 뷰를 추천하는 요약 권고 기능을 추가한다.

## An Extension of the DBMax for Data Warehouse Performance Administration

eun ju Kim<sup>†</sup> · hwan seung Young<sup>††</sup> · sang won Lee<sup>†††</sup>

## ABSTRACT

As the usage of database systems dramatically increases and the amount of data pouring into them is massive, the performance administration techniques for using database systems effectively are getting more important. Especially in data warehouses, the performance management is much more significant mainly because of large volume of data and complex queries. The objectives and characteristics of data warehouses are different from those of other operational systems so adequate techniques for performance monitoring and tuning are needed. In this paper we extend functionalities of the DBMax, a performance administration tool for Oracle database systems, to apply it to data warehouse systems. First we analyze requirements based on summary management and ETL functions they are supported for data warehouse performance improvement in Oracle 9i. Then, we design architecture for extending DBMax functionalities and implement it. In specifics, we support SQL tuning by providing details of schema objects for summary management and ETL processes and statistics information. Also we provide new function that advises useful materialized views on workload extracted from DBMax log files and analyze usage of existing materialized views.

**키워드 :** 데이터 웨어하우스(Data Warehouse), 성능 관리(Performance Administration), 성능 모니터링(Performance Monitoring), Sql 튜닝(Sql Tuning), 오라클 9i(Oracle9i), 요약 관리(Summary Management), 실제화된 뷰(Materialized View), ETL, DBMax

### 1. 서 론

1990년대 중반 이후 전사적 자원 관리(ERP : Enterprise Resource Planning), 고객 관계 관리(CRM : Customer Relationship Management)와 같은 기업의 전략적 정보 시스템이 국내에 활발히 도입되면서 그 근간을 이루는 데이터베이스 시스템의 성능이 기업의 생산성과 직결되는 중요 요인으로 작용하고 있다. 정보시스템의 성능은 데이터베이스의 튜

닝 여하에 따라 평균 수십배 가량의 속도 차이를 낼 수 있는데 이는 결국 수많은 직원들이 질의 결과를 기다려서 확인하고 다음 업무를 처리하는데 걸리는 시간이 평균 수십배 가량 차이가 난다는 것을 의미한다. 현재 대부분의 상용 DBMS(Database Management System) 제품들은 자체적으로 최적의 성능 관리와 튜닝을 위한 도구들[1-4]을 제공하고 있다.

그러나 데이터베이스에 유입되는 데이터의 양이 방대해지고 사용자 및 SQL 응용 프로그램의 질의가 점점 복잡해지면서 데이터베이스를 근간으로 하는 기업의 정보시스템에 성능 문제가 발생한 경우 그 원인을 파악하고 해결하기 위한 작업이 더욱 어려워지고 있다. 현재 대부분의 정보 시스템은 다수의 응용 프로그램이 동시에 하나 이상의 데이터베이스에 접근하여 업무를 처리하는 경우가 많기 때문에 단일 데이터

※ 본 연구는 과학재단 산학협력과제(과제명 : 데이터 웨어하우스 성능 모니터링을 위한 DBMax 기능 확장에 관한 연구) 지원에 의해 수행되었음.

† 준 회원 : 삼성전자

†† 종신회원 : 이화여자대학교 컴퓨터학과 교수

††† 정 회원 : 성균관대학교 정보통신공학부 교수

논문접수 : 2002년 10월 14일, 심사완료 : 2003년 2월 3일

베이스에 대한 성능 모니터링만으로는 문제의 원인을 파악하기 어렵다. 특히 이러한 성능 문제는 수백 기가 바이트(GB)에서 수십 테라 바이트(TB)에 이르는 대용량 데이터 웨어하우스 환경에서 훨씬 더 중요한 문제로 인식되고 있다.

본 연구에서는 데이터베이스 시스템의 성능 관리를 위해 개발된 많은 제품들 중 최초의 국산 소프트웨어인 DBMax를 데이터 웨어하우스 시스템에 적합하도록 기능을 확장한다. 현재 DBMax는 오라클을 이용해 구축한 OLTP(On-Line Transaction Processing)용 시스템에 대한 실시간 성능 모니터링과 SQL 튜닝 기능을 제공하는데 본 연구를 통해 최근 수요가 급증하고 있는 데이터 웨어하우스 시스템에 대한 성능 관리 기능을 추가하고자 한다.

우선 데이터 웨어하우스 시스템의 성능 관리시 고려해야 할 요구 사항을 분석하고, 이를 지원하는 확장 아키텍처를 설계 및 구현한다. 본 연구에서는 오라클 9i가 제공하는 데이터 웨어하우스 핵심 기능중 실체화된 뷰(materialized view)를 이용한 요약 관리(summary management) 기능과 ETL(Data Extraction, Transformation and Loading) 기능을 중심으로 확장 방안을 설계한다. 오라클 9i의 관련 기능에 대한 분석 결과를 토대로 요약 관리와 ETL 성능 지원을 위해 제공하는 다양한 스키마 객체들에 대한 상세 정보를 제공하고 질의 재작성 정보를 표시하여 데이터 웨어하우스 환경에서 수행되는 질의에 대한 SQL 튜닝 기능을 강화한다. 또한 사후 분석을 위한 DBMax의 로그 파일에서 의미 있는 작업 로드를 추출하여 잠재적으로 유용한 실체화된 뷰를 추천하는 요약 권고 기능을 추가한다.

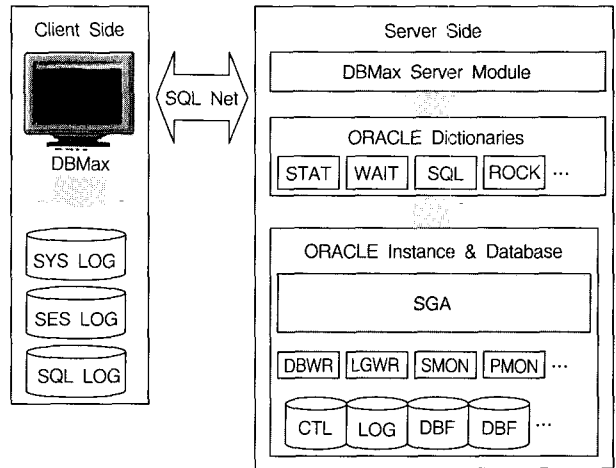
본 논문의 구성은 다음과 같다. 2장에서는 DBMax의 주요 특징과 모니터링 수행 구조를 살펴보고 오라클 9i의 요약 관리 기능과 ETL 기능을 소개한다. 3장에서는 데이터 웨어하우스 시스템의 성능 관리를 위한 요구 사항을 관련 기능별로 분석하고 DBMax가 이를 얼마나 지원하고 있는지 살펴본 후 확장 아키텍처를 설계한다. 4장에서는 3장에서 설계한 확장 기능들을 구현한 프로토타입 시스템의 동작 원리와 구현 방법을 설명하고, 결과 화면을 소개한다. 5장에서는 결론 및 향후 연구 과제를 제시한다.

## 2. 관련 연구

### 2.1 DBMax

DBMax는 (주)엑셈[5]에서 개발한 최초의 국산 성능 관리 도구로 오라클 데이터베이스를 운영하는데 있어서 발생할 수 있는 성능 문제들을 실시간으로 모니터링하여 발생한 문제의 원인을 신속히 찾아내고 지속적으로 분석할 수 있도록 지원한다. 구체적으로 DBMax는 DBA(Database Administrator) 또는 DPA(Database Performance Administrator)로 하여금 ① 다중 데이터베이스를 동시에 사용하는 환경에서, ② 문제가 되는 시스템, 프로그램, SQL 문을 신속히 발견하고, ③ 이들의 튜닝에 필요한 성능 및 통계 정보를 제공하는 역할을 한다[6, 7].

DBMax는 실시간 모니터링을 위해 데이터베이스 인스턴스 및 데이터베이스에서 발생하는 모든 상세 정보와 성능 정보를 실시간으로 저장하고 있는 오라클의 디렉너리에 주기적으로 접근한다. (그림 1)은 DBMax의 모니터링 수행 구조[8]를 나타낸 것이다.



(그림1) DBMax의 모니터링 수행 구조

(그림 1)에서 DBMax는 대상 데이터베이스에 설치되는 PL/SQL로 개발된 서버 모듈을 통해 클라이언트의 디렉너리 조회 요청을 받아들이고 필요한 정보를 조회하여 정해진 형식으로 전송한다. DBMax의 모니터링 클라이언트는 서버 모듈로부터 전송된 디렉너리 정보를 사용자가 보기 쉽도록 가공하고 그래프로 그려내는 역할을 한다. 이 때 DBMax의 서버 모듈과 클라이언트 모듈은 오라클의 SQL \*Net을 통해서로 통신한다[9].

DBMax는 오라클에서 지속적으로 양산하는 성능 관련 통계 정보와 대기 이벤트 정보를 이용하여 CPU, I/O, 메모리 등에 대한 자원 사용량(STAT)과 자원을 사용하기 위한 경합(WAIT) 현상을 실시간으로 보여준다. 우선 데이터베이스 단위로 자원 사용량과 대기 이벤트 정보들을 모니터링하다가 문제 상황이 감지되면 부하 세션 단위로 내려가 수행 SQL별 성능 지표들을 비교한 후 문제 SQL을 신속히 추출한다. 부하 SQL에 대해서는 실행 계획을 분석하고 질의 수행시 접근하는 테이블, 인덱스에 대한 객체 정보를 제공하여 SQL 튜닝을 지원한다. DBMax는 실시간으로 데이터베이스의 성능을 모니터링할 뿐만 아니라 실시간에 모니터링 중인 정보들을 파일에 기록하고 나중에 재생할 수 있도록 하여 사후 분석 기능을 제공한다[9].

### 2.2 오라클 9i의 데이터 웨어하우스 기능

데이터 웨어하우스 응용 분야를 지원하기 위해 오라클 9i의 DBMS 엔진에는 데이터 웨어하우스 분야의 핵심 기술 [10, 11]인 요약 관리 기능과 ETL 기능이 통합되어 있다.

#### 2.2.1 요약 관리 기능

요약 데이터(summary)는 대용량 데이터의 복잡한 분석을

위해 자주 사용되는 집계 정보를 미리 계산하여 테이블의 형태로 저장해둔 것으로 오라클 9i에서는 실체화된 뷰 스키마 객체를 사용하여 요약 데이터를 생성할 수 있다.

실체화된 뷰는 데이터 웨어하우스 시스템의 질의 수행 성능을 향상시키기 위한 주요 기법으로 질의 최적기에 의한 재작성 과정을 통해 질의 응답 시간을 단축시킨다. 오라클 9i에서는 실체화된 뷰의 정의 구문과 질의 텍스트가 일치하는 경우뿐만 아니라 일치하지 않는 경우에도 조인이나 그룹화 조건 등을 비교하여 가능한 경우 질의를 재작성 한다. 실체화된 뷰를 사용할 때 상세 테이블의 데이터가 변경되면 실체화된 뷰에 변경 내용을 지속적으로 반영해야 하는데 오라클 9i는 이를 위해 다양한 갱신(refresh) 모드와 방법을 제공한다. 또한 주어진 작업 로드와 대해 적절한 실체화된 뷰의 생성을 권고하고 기존의 실체화된 뷰의 실제 효용성을 분석하는 요약 권고자를 제공하여 실체화된 뷰의 생성 및 관리를 용이하게 한다[12-14].

### 2.2.2 ETL 기능

ETL은 원본 시스템에서 필요한 데이터를 추출하고, 추출한 데이터를 목표 시스템에서 요구하는 형태로 변환한 후 데이터 웨어하우스 시스템에 로딩하는 전과정을 의미한다. 오라클 9i는 ETL 작업의 성능을 향상시키기 위하여 CDC(Change Data Capture)기능, 외부 테이블(external table), upsert 기능, 다중 테이블 삽입(multi-table insert) 기능, 테이블 함수(table function)의 다섯 가지 확장 기능을 제공한다.

CDC 기능은 데이터 추출 작업의 성능을 향상시키기 위한 기능으로 원본 데이터의 변경 부분만 추출하여 목표 테이블에 반영할 수 있도록 한다. 외부 테이블은 외부에 존재하는 소스 데이터 파일을 데이터 웨어하우스에 로딩하지 않고도 마치 데이터베이스 내의 일반 테이블인 것처럼 사용할 수 있게 한다. 일반적으로 원본 데이터의 변경은 새로운 레코드의 추가와 기존 레코드의 수정 연산을 모두 포함하는데 upsert 기능은 데이터 로딩시 한 번의 연산으로 테이블의 행을 조건적으로 추가하거나 변경할 수 있도록 지원하여 로딩 성능을 향상시킨다.

다중 테이블 삽입 기능과 테이블 함수는 데이터 변환 작업의 성능을 향상시키기 위해 지원하는 ETL 기능이다. 다중 테이블 삽입 기능은 같은 원본 데이터를 다중의 테이블에 서로 다른 변형 규칙을 적용하여 한번에 저장할 수 있도록 한다. 데이터 변환 작업은 보통 여러 단계를 거쳐 수행되는데 외부 테이블은 병렬 방식이나 파이프라인 방식을 사용하여 메모리보다 큰 중간 변환 결과를 별도의 테이블에 저장하지 않고도 데이터를 변환할 수 있기 때문에 데이터 변환 작업을 신속히 처리할 수 있다[12, 14, 15].

## 3. 확장 아키텍처

### 3.1 요구 사항 분석

본 절에서는 2장에서 살펴본 오라클 9i의 요약 관리 기능과 ETL 확장 기능들을 중심으로 데이터 웨어하우스 시스템

의 성능 관리를 위한 요구 사항을 분석한다.

#### 3.1.1 요약 관리 기능

오라클 9i에서 제공하는 요약 관리 컴포넌트[12]와 실체화된 뷰에 대한 이슈[10, 16]를 토대로 요약 관리 기능을 위한 성능 관리 요구 사항을 분석한다. 실체화된 뷰와 관련된 이슈는 어떤 뷰를 실체화할 지에 대한 설계(design) 문제, 실체화된 뷰의 지속적인 갱신을 위한 유지 관리(maintenance) 문제, 성능 향상을 위해 실체화된 뷰를 효율적으로 이용하는 이용(exploitation)에 관한 이슈 세 가지가 있다.

일반적으로 실체화된 뷰와 관련된 성능 문제는 기대했던 질의가 재작성 되지 않은 경우에 많이 발생한다. 질의를 수행할 때, 질의 응답 시간을 단축할 수 있는 요약 데이터를 이용하는 것이 아니라 대용량의 상세 데이터에 접근하여 필요한 결과를 직접 계산해내기 때문에 성능이 저하되는 것이다. 질의 재작성을 보장하는 규칙은 차원, 그룹화 조건, 조인 조건, 질의 재작성 활성화 여부, 제약 조건 등 매우 복잡하기 때문에 부하 SQL에 대한 실행 계획 분석시 질의 재작성 정보를 함께 제공한다면 사용자가 재작성과 관련된 성능 문제의 원인을 신속히 파악하고 효과적인 조치를 취할 수 있을 것이다.

스키마 객체로서의 실체화된 뷰는 일반 테이블과 같은 물리적 속성뿐만 아니라 실체화된 뷰를 생성할 때 지정할 수 있는 다양한 옵션[12, 14]에 대한 논리적 속성과 능력(capability) 정보[12]를 가지고 있다. 부하 SQL에 대한 튜닝시 질의 최적기가 접근하는 실체화된 뷰에 대한 상세 정보를 제공하여 사용자가 참고할 수 있도록 해야 한다.

DBMax는 실시간에 모니터링 중인 성능 정보들을 파일에 기록하고 나중에 재생할 수 있도록 시스템과 세션, SQL 레벨의 로그 파일을 제공한다. DBMax의 로그 파일에는 모니터링 대상 시스템에서 수행되는 유용한 작업 로드가 저장되어 있기 때문에 이를 이용하여 기존의 실체화된 뷰의 효용성을 분석하고, 유용한 실체화된 뷰를 권고하여 시스템의 성능을 향상시킬 수 있도록 지원한다.

기술한 내용을 바탕으로 요약 관리 기능을 위한 확장 방안을 정리하면 다음과 같다.

- 실체화된 뷰의 상세 정보 제공
- 질의 재작성 정보 제공
- 로그 파일을 이용한 요약 권고 기능 지원

#### 3.1.2 ETL

2장에서 소개한 외부 테이블은 별도의 로딩 과정 없이 플랫 파일에 저장되어 있는 외부 데이터를 데이터베이스 내의 일반 테이블인 것처럼 사용할 수 있기 때문에 외부 데이터가 자주 사용되지 않는 매우 큰 데이터일 경우 아주 유용하게 사용할 수 있다. 따라서 일반 테이블 외에 외부 테이블에 대한 상세 정보를 제공하여 SQL 튜닝시 참고할 수 있도록 한다. 스키마 객체로서의 외부 테이블은 기본 컬럼 정보 이외에 원본 데이터 파일에 대한 정보를 부가적으로 가지고 있다.

병렬 방식과 파이프라인 방식을 이용하여 데이터 변환

작업을 수행하는 테이블 함수는 사용자가 목적에 맞게 정의하여 사용하기 때문에 테이블 함수 자체에 결함이 있을 수 있다. 따라서 테이블 함수의 결함으로 인한 성능 문제 발생시 문제 테이블 함수의 상태 정보와 텍스트 원본 정보 등을 제공하여 사용자가 튜닝시 참고할 수 있도록 한다. 또한 테이블 함수가 생성될 때 함께 생성하거나 참조하는 데이터 타입, 패키지 등의 사용자 정의 객체에 대한 정보도 함께 표시하여 테이블 함수의 결함으로 인한 성능 문제를 신속히 해결할 수 있도록 지원해야 한다.

기술한 내용을 바탕으로 ETL 기능을 위한 확장 방안을 정리하면 다음과 같다.

- 외부 테이블에 대한 상세 정보 제공
- 테이블 함수 및 참조 객체에 대한 상세 정보 제공

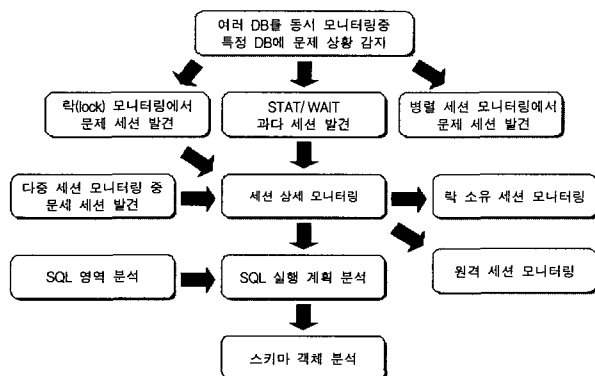
3.2 확장 아키텍처 설계

본 절에서는 요구 사항 분석 결과와 기존 DBMax의 성능 관리 구조를 토대로 DBMax를 데이터 웨어하우스 시스템에 적용할 때 드러나는 취약점을 보완하기 위한 확장 아키텍처를 설계한다.

3.2.1 DBMax의 하향식 문제 접근 방식

데이터베이스의 문제 상황은 특정 세션이 수행하는 특정 SQL에 의해 발생하는데 DBMax는 문제 세션과 문제 SQL 문을 추출하기 위하여 하향식 방식으로 접근한다. 우선 데이터베이스 단위로 성능 정보와 대기 이벤트 정보를 모니터링하다가 문제 상황이 감지되면 부하 세션 단위로 내려가 수행 SQL별 성능 지표들을 비교하고 문제 SQL을 신속히 추출한다. 또한 부하 SQL의 실행 계획을 분석하고 수행시 참조하는 테이블, 인덱스 스키마 객체 정보를 분석하여 SQL 튜닝 기능을 지원한다.

(그림 2)는 하향식 접근 방식을 이용하여 문제 상황을 파악하는 과정을 시나리오 단계별로 설명한 그림 [9]이다.



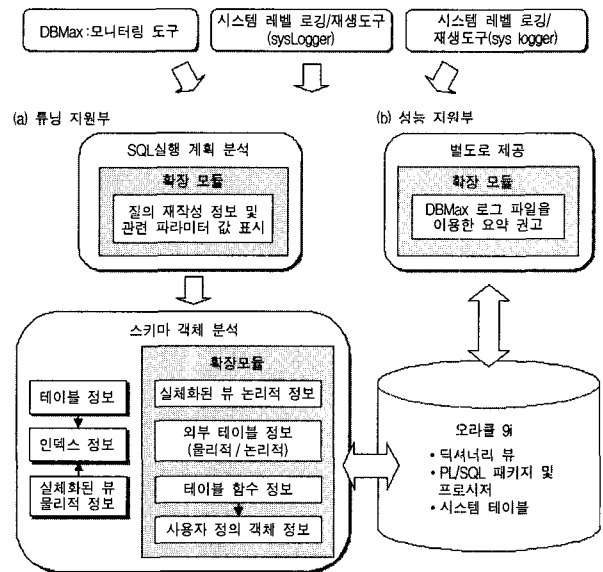
(그림 2) 문제 상황에 대한 하향식 접근

(그림 2)에서 DBMax의 성능 관리 구조를 크게 문제 상황을 감지하기 위한 모니터링부와 실제 SQL 문의 튜닝을 지원하기 위한 튜닝 지원부로 나누어볼 수 있다. 모니터링

부에서는 시스템, 세션, SQL 레벨로 다양한 리소스 정보와 대기 이벤트 정보를 표시하여 문제 상황을 신속히 파악할 수 있도록 하며 튜닝 지원부에서는 문제 SQL에 대한 실행 계획 분석과 스키마 객체 분석을 통해서 관리자가 튜닝시 참고할 수 있는 유용한 정보를 제공한다.

3.2.2 확장 아키텍처

(그림 3)은 DBMax를 데이터 웨어하우스 성능 관리에 적용하기 위한 주요 확장 기능과 각 기능 간의 호출 관계를 표시한 개요도이다.



(그림 3) 확장 아키텍처

본 연구에서는DBMax의 확장 아키텍처를 성능 모니터링 및 로그 기록을 수행하는 기존의 DBMax 모듈과 새롭게 추가되는 확장 모듈이 상호 연동되는 형태로 구성하여 자연스러운 확장을 유도하였다. DBMax의 확장 기능은 SQL 문의 튜닝을 지원하기 위한 튜닝 지원부와 요약 권고를 담당하는 성능 지원부로 나누어 볼 수 있다. 이 때 확장 모듈은 필요한 정보를 추출하기 위하여 오라클 데이터 웨어하우스에서 제공하는 다양한 딕셔너리 뷰와 시스템 테이블들을 조회하며 PL/SQL 패키지와 프로시저를 사용한다.

• 튜닝 지원부

(그림 3)(a)는 튜닝 지원부로 데이터 웨어하우스 환경에서 수행되는 질의에 대한 실행 계획 분석과 스키마 객체 분석을 통하여 성능 문제의 원인을 신속히 파악하고, 적절한 조치를 취할 수 있도록 지원한다.

튜닝 지원부는 실행 계획 분석 단계를 거쳐 요청한 스키마 객체 정보를 표시하는데 이 때 질의 재작성 정보는 중요한 성능 지표로 사용될 수 있다. 기존의 DBMax는 부하 SQL의 실행 계획을 분석하여 질의가 재작성 되었는지의 여부를 확인할 수 있도록 지원하지만 질의 재작성에 대한 상세 정보를 전혀 제공하지 않기 때문에 재작성과 관련된

성능 문제 발생시 근본적인 원인을 파악할 수가 없다. 따라서 실행 계획 분석과 함께 질의 재작성 정보를 분석하고, 재작성 여부에 직접적인 영향을 주는 세션의 파라미터 설정 값을 표시하여 사용자가 손쉽게 재작성과 관련된 성능 문제를 해결할 수 있도록 지원한다.

스키마 객체 분석 모듈에서는 테이블, 인덱스 이외에 데이터 웨어하우스 환경에서 보편적으로 사용하는 실체화된 뷰와 외부 테이블, 테이블 함수에 대한 상세 정보를 제공한다.

● 성능 지원부

(그림 3)(b)는 성능 지원부로 DBMax의 다양한 로그 파일을 이용하여 모니터링하는 대상 시스템의 성능을 향상시킬 수 있도록 지원한다. 로그 파일에 기록된 SQL 문들을 대상으로 시스템의 성능 향상에 도움이 되는 실체화된 뷰를 생성하고, 자주 사용되지 않거나 사용되더라도 차지하는 공간 대비 성능 향상 효과가 낮은 실체화된 뷰를 제거하여 최적화된 상태를 유지할 수 있도록 한다. 확장 모듈의 성능 지원 기능을 이용하면 사용자가 별도의 처리 작업 없이 유용한 요약 권고 사항을 쉽게 획득할 수 있다.

4. 프로토타입 구현

본 장에서는 3장에서 설계한 확장 기능들을 구현한 프로토타입 시스템의 구조와 동작 원리를 설명하고, 결과 화면을 제시한다.

4.1 구현 환경

설계한 확장 기능을 구현하고 테스트한 환경은 <표 1>과 같다.

<표 1> 구현 환경

개발 환경	Windows 2000 Server
사용 DBMS	오라클 9i
개발 언어	PL/SQL 비주얼 베이직 6.0 ADO 2.6 라이브러리
데이터 연결	ODBC

오라클의 덱서너리 뷰와 시스템 테이블에 접근하여 요청된 정보를 추출하는 서버 모듈은 PL/SQL을 사용하여 구현하였으며 모니터링 하고자 하는 대상 시스템의 데이터베이스 내부에 설치하여 실행시의 부하를 최소화하였다. 서버 모듈은 PL/SQL 패키지 형태로 구현하였으며 확장 서버 모듈의 패키지 내부에는 필요한 정보를 조회하는 프로시저들과 추출한 정보를 저장하는 사용자 정의 타입들이 정의되어 있다. 서버 모듈이 전송한 정보를 가공하여 사용자에게 보여주는 인터페이스 모듈은 비주얼 베이직 6.0을 사용하여 구현하였으며 데이터 연결은 ODBC를 이용하였다.

4.2 구현 방법 및 결과

서버 모듈이 참조하는 데이터베이스 객체 목록과 제공

정보, 구현 방법을 설명하고 구현 화면을 제시한다.

4.2.1 튜닝 지원부

가) 실행 계획 분석 모듈 - 질의 재작성 정보 표시 기능

수행된 질의에 대한 재작성 정보와 질의 재작성 여부에 영향을 주는 세션 파라미터 설정 값을 표시한다. DBMax의 실행 계획 분석 화면(클라이언트)에서 부하 SQL에 대한 재작성 정보를 요청하면 재작성 정보를 분석하는 서버 모듈의 프로시저를 호출한다. 이때 부하 SQL 텍스트와 부하 SQL을 수행한 세션 정보를 서버 측에 전달한다. <표 2>는 재작성 정보를 표시하기 위해 호출하는 확장 서버 모듈의 프로시저 목록이다.

<표 2> 재작성 정보 분석 프로시저

서버 모듈 프로시저	입력 파라미터	설명
GET_PARAMS_VALUE	세션 명	초기화 파라미터 설정 값 조회
GET_REWRITE_MSG	부하 SQL 텍스트	질의 재작성 정보 조회
GET_MVIEW_LIST	스키마 명	접근 가능한 실체화된 뷰 목록 조회

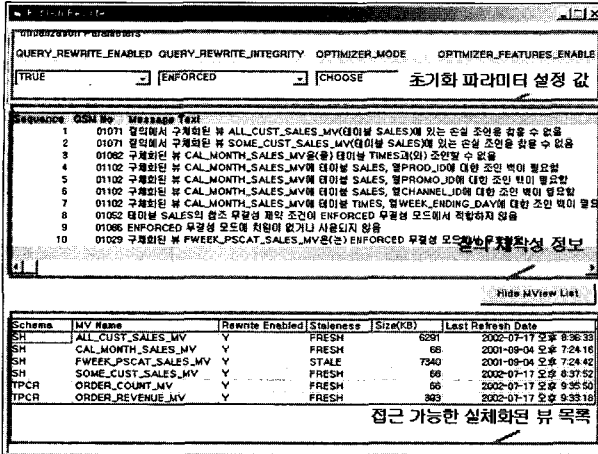
실제 재작성 정보를 추출하는 GET\_REWRITE\_MSG 프로시저는 오라클 9i에서 제공하는 DBMS\_MVIEW.EXPLAIN\_REWRITE 프로시저를 호출하여 재작성 정보를 분석한 후 REWRITE\_TABLE 테이블에서 분석 결과를 추출한다. GET\_PARAMS\_VALUE 프로시저와 GET\_MVIEW\_LIST 프로시저는 전달된 세션 정보를 이용하여 덱서너리 뷰를 조회하고 추출한 정보를 클라이언트 모듈에 전송한다. <표 3>은 각 프로시저들이 필요한 정보를 추출하기 위해 참조하는 객체 목록[17, 18]이다.

<표 3> 질의 재작성 정보 표시를 위한 참조 객체

프로시저	참조 객체
GET_PARAMS_VALUE	V\$PARAMETER
GET_REWRITE_MSG	ALL_MVIEWS
GET_MVIEW_LIST	DBMS_MVIEW.EXPLAIN_REWRITE REWRITE_TABLE

(그림 4)는 부하 SQL에 대한 재작성 정보 표시 화면으로 크게 세 부분으로 구성되어 있다. 첫번째는 부하 SQL을 수행한 세션의 초기화 파라미터(initialization parameter) 중 질의 재작성에 영향을 주는 네 가지 파라미터의 설정 값을 표시하는 부분이다. 두 번째는 실제 재작성 정보 분석 결과를 표시하는 부분이며 세 번째는 현재 세션의 접근 가능한 실체화된 뷰 목록을 표시하고 각각에 대한 상세 정보를 표시하는 부분이다.

(그림 4)에서 초기화 파라미터 설정 값 표시부분은 각각의 파라미터가 가질 수 있는 값의 목록을 콤보 박스 형태로 구성하여 사용자가 값을 변경할 때마다 'ALTER SESSION~SET~'구문을 실행하여 현재 세션의 파라미터 설정 값



(그림 4) 질의 재작성 정보

을 변경하고, GET\_REWRITE\_MSG 프로시저를 호출하여 질의 재작성 정보를 갱신한다. <표 4>은 확장 기능에서 표시하는 초기화 파라미터 목록[12, 17]이다.

<표 4> 초기화 파라미터

파라미터 명	설 명
QUERY_REWRITE_ENABLED	재작성 활성화 여부
QUERY_REWRITE_INTEGRITY	질의 재작성 무결성 모드
OPTIMIZER_MODE	질의 최적기 모드 (규칙 기반/비용 기반)
OPTIMIZER_FEATURE_ENABLE	질의 최적기의 릴리즈 번호

재작성 정보 표시 부분에는 접근 가능한 실체화된 뷰 각각에 대해 재작성 활성화 여부, 무결성 모드, 조인 및 그룹화 조건 등의 다양한 측면에서 재작성 정보를 분석한 결과가 표시된다. 이 메시지를 통해 사용자가 재작성과 관련된 성능 문제의 원인을 쉽게 파악하고 효과적인 조치를 취할 수 있다. 해당 세션이 접근할 수 있는 실체화된 뷰 목록은 질의 재작성 정보 메시지에 언급되어 있는 실체화된 뷰의 상세 정보를 확인할 수 있도록 하기 위해 제공한다. 목록에서 실체화된 뷰 이름을 선택하면 선택한 실체화된 뷰의 물리적 속성과 논리적 속성 및 능력 정보를 표시하는 화면을 호출한다.

나) 스키마 객체 분석 모듈

스키마 객체 분석을 위한 확장 모듈은 실체화된 뷰, 외부 테이블, 테이블 함수와 테이블 함수 생성시 참조하는 사용자 정의 객체에 대한 스키마 정보를 분석하여 사용자가 SQL 튜닝시 참고할 수 있도록 한다.

● 실체화된 뷰

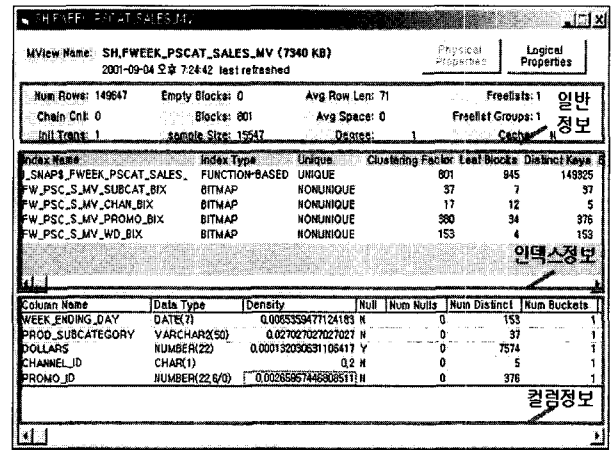
기존 DBMax는 실체화된 뷰를 일반 테이블과 구분하여 인식하지 않기 때문에 실체화된 뷰의 중요한 성능 정보를 표시하지 않는다. 확장 모듈에서는 일반 테이블과 실체화된 뷰를 별도의 객체로 인식하고, 실체화된 뷰에 대한 물리적 속성 이외에 논리적 속성 정보와 능력 정보를 함께 제공한다. DBMax 말단의 실행 계획 분석 화면에서 트리에 표시된 실체화된 뷰

이름을 클릭하면 실체화된 뷰 상세 정보를 추출하는 프로시저를 호출하는데 이때 선택한 실체화된 뷰 명과 스키마 명을 전달한다. 서버 모듈은 디서너리 뷰에서 객체 정보를 조회하여 클라이언트에게 전송한다. <표 5>는 실체화된 뷰의 상세 정보를 조회하는 확장 모듈의 프로시저 목록이다.

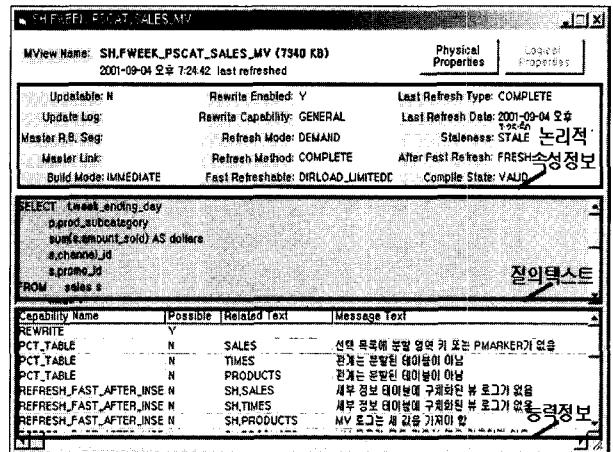
<표 5> 실체화된 뷰 정보 조회 프로시저

구 분	확장 모듈 프로시저	
물리적 정보	일반 정보	GET_TABLE_PROPS
	인덱스 정보	GET_INDEXES
	컬럼 정보	GET_TAB_COLUMNS
논리적 정보	GET_MVIEW_PROPS	
능력 정보	GET_MVIEW_CAPABILITIES	

실체화된 뷰의 물리적 속성 정보와 논리적 속성 정보는 오라클의 디서너리 뷰를 조회하여 얻을 수 있다. 능력 정보를 추출하는 GET\_MVIEW\_CAPABILITIES 프로시저는 오라클 9i의 DBMS\_MVIEW.EXPLAIN\_MVIEW 프로시저를 호출하여 능력 정보를 분석한 후 MV\_CAPABILITIES\_TB



(a) 물리적 속성 정보



(b) 논리적 속성 및 능력 정보

(그림 5) 실체화된 뷰 상세 정보

BLE 테이블에서 분석 결과를 추출한다. <표 6>은 각각의 프로시저가 참조하는 객체 목록[17,18]이며 클라이언트 모듈은 전송된 데이터를 가공하여 (그림 5)의 화면을 표시한다.

<표 6> 실제화된 뷰 정보 표시를 위한 참조 객체

프로시저	참조 객체
GET_TABLE_PROPS	ALL_TABLES / ALL_SEGMENTS
GET_INDEXES	ALL_INDEXES
GET_TAB_COLUMNS	ALL_TAB_COLUMNS
GET_MVIEW_PROPS	ALL_MVIEWS ALL_MVIEW_REFRESH_TIMES
GET_MVIEW_CAPABILITIES	DBMS_MVIEWEXPLAIN_MVIEW MV_CAPABILITIES_TABLE

(그림 5)(a)는 실제화된 뷰의 물리적 속성 정보를 표시하는 화면으로 크기, 로우 수, 블록 수 등의 일반 정보와 인덱스 정보, 구성 컬럼 정보를 표시한다. (그림 5)(b)는 논리적 속성 정보와 능력 정보를 표시하는 화면으로 생성시 지정한 갱신 모드와 방법, 질의 텍스트 정보 등을 제공하고, 오라클 9i에서 정의한 17가지 능력 목록[12]에 대해 선택한 실제화된 뷰가 이를 지원하는지를 표시한다.

● 외부 테이블

기존 DBMax는 외부 테이블에 대한 객체 정보를 전혀 제공하지 않는데 확장 모듈에서는 외부 테이블의 컬럼 정보와 원본 데이터 파일에 대한 상세 정보를 제공하여 부하 SQL에 대한 튜닝 기능을 강화한다. 기존 DBMax 실행 계획 트리에서 외부 테이블이 표시된 행을 클릭하면 외부 테이블의 객체 정보를 조회하는 확장 서버 모듈을 호출한다. 서버 모듈은 덱서너리 뷰에서 외부 테이블에 대한 객체 정보를 조회하여 클라이언트에 전송하고, 클라이언트 모듈은 (그림 6)과 같이 전송 받은 데이터를 가공하여 표시한다.

External Table Name: SH\_SALES\_DELTA\_XT

Num Rows:      Empty Blocks:      Avg Row Len:      Freelists:      F 기본 물리적 정보

Chain Cnt:      Blocks:      Avg Space:      Degree: 1      Cache: N

Column Name	Data Type	Density	Null	Num Nulls	Num Distinct	Num Buckets	Low Value	High Va
PROD_ID	NUMBER(22,6)		Y					
CUST_ID	NUMBER(22)		Y					
TIME_ID	DATE(7)		Y					
CHANNEL_ID	CHAR(1)		Y					
PROMO_ID	NUMBER(22,6)		Y					
QUANTITY_SOLD	NUMBER(22,3)		Y					
AMOUNT_SOLD	NUMBER(22,10)		Y					

컬럼 정보

Type Owner: SYS      Default Directory Owner: SYS

Type Name: ORACLE\_LOADER      Default Directory Name: DATA\_DIR

Access Type: CLOB      Access parameters

Reject Limit: UNLIMITED      RECORDS DELIMITED BY NEWLINE CHARACTERSET

Location: salesDelta.dat      BADFILE log\_dir

외부 테이블 및 원본 정보

(그림 6) 외부 테이블 상세 정보

외부 테이블은 일반 테이블과 달리 외부 플랫폼 파일에 저장된 데이터에 접근하므로 원본 데이터 파일이 존재하는 디렉토리 정보와 파일 이름, 접근 방식 등에 대한 추가의

정보를 가지고 있다. 외부 테이블 상세 정보 표시 화면에서는 덱서너리 뷰 조회를 통해서 컬럼 정보와 원본 데이터 파일에 대한 정보를 함께 표시한다. 외부 테이블에는 인덱스를 생성할 수 없기 때문에 인덱스 정보는 제외한다.

<표 7>은 (그림 6)의 정보를 표시하기 위해 클라이언트 모듈이 호출하는 확장 서버 모듈의 프로시저 목록으로 프로시저 호출 시 스키마 명과 외부 테이블 명을 입력 파라미터로 제공해야 한다.

<표 7> 외부 테이블 정보 조회 프로시저

확장 모듈 프로시저	구분	참조 객체
GET_TABLE_PROPS	기본 테이블 정보	ALL_TABLES
GET_TAB_COLUMNS	컬럼 정보	ALL_TAB_COLUMNS
GET_XT_PROPS	외부 테이블 정보	ALL_EXTERNAL_TABLES
GET_XT_SOURCE	원본 파일 정보	ALL_EXTERNAL_LOCATIONS

● 테이블 함수 및 사용자 정의 객체

기존 DBMax를 통해서는 부하 SQL이 참조하는 테이블 함수 명만을 확인할 수 있는데 확장 모듈에서는 테이블 함수의 상세 스키마 정보를 제공하여 테이블 함수 자체의 결합으로 인한 성능 문제의 원인을 신속히 파악할 수 있도록 지원한다. (그림 7)은 테이블 함수의 상세 정보를 표시하는 화면이다.

Table Function Name: SH\_OBSOLETE\_PRODUCTS\_PIPE

Aggregate: NO      Pipelined: YES      Parallel: YES

Status: VALID      함수 기본 정보 및 상태 정보

DEFINITION      텍스트 원본 정보

```

FUNCTION obsolete_products_pipe(cursor_pkgs from_refcur)
RETURN product_t_tstble
PIPELINED
PARALLEL_ENABLE (PARTITION cur BY ANY) ISB
prod_id NUMBER(6)B
prod_name VARCHAR(50)B
prod_desc VARCHAR(400)B
prod_subcategory VARCHAR(50)B
prod_subcat_desc VARCHAR(200)B
prod_category VARCHAR(50)B
prod_cat_desc VARCHAR(200)B
prod_weight_class NUMBER(2)B
prod_unit_of_measure VARCHAR(20)B
prod_pack_size VARCHAR(30)B
supplier_id NUMBER(6)B
prod_status VARCHAR(20)B
prod_list_price NUMBER(8,2)B
prod_min_price NUMBER(8,2)B
prod_total sales NUMBER(15)B
BEGIN
LOOP
-- Fetch from cursor variableB
FETCH cur INTO prod_id, prod_name, prod_desc, prod_subcategory,B
prod_subcat_desc, prod_category, prod_cat_desc, prod_weight_cla
prod_unit_of_measure, prod_pack_size, supplier_id, prod_status,B
    
```

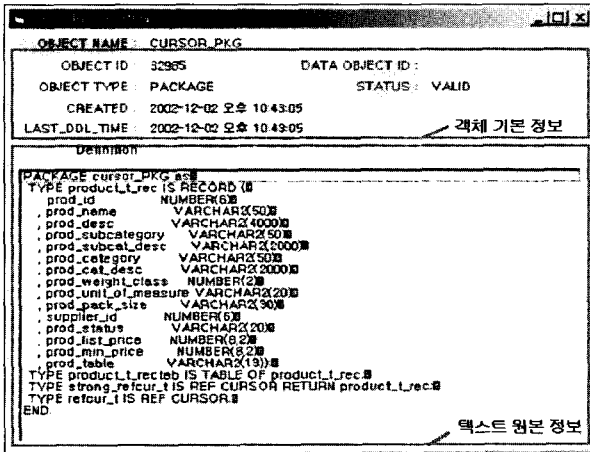
(그림 7) 테이블 함수 상세 정보

테이블 함수가 현재 유효한지 상태 정보를 제공하고, 테이블 함수를 수행할 때 파일프라인 방식과 병렬 방식을 사용할 수 있는지의 여부를 표시한다. 또한 테이블 함수를 정의할 때 사용한 텍스트 원본을 표시하여 해당 함수가 어떤 기능을 수행하는지 확인할 수 있도록 한다. <표 8>은 테이블 함수 정보를 추출하기 위해 호출하는 확장 서버 모듈의 프로시저 목록[17]으로 프로시저 호출 시 스키마 명과 테이블 함수 명을 입력 파라미터로 제공한다.

〈표 8〉 테이블 함수 정보 조회 프로시저

확장 모듈 프로시저	구분	참조 객체
GET_TF_PROPS	테이블	ALL_PROCEDURES
	함수	ALL_OBJECTS
GET_TXT_SOURCE	텍스트 원본 정보	ALL_SOURCE

일반적으로 테이블 함수를 생성할 때에는 이와 관련된 타입이나 패키지들을 함께 생성하기 때문에 테이블 함수 정의 구문에는 이러한 사용자 정의 타입과 패키지 이름이 포함되어 있다. 따라서 정의 구문에서 타입이나 패키지 명이 포함된 열을 클릭하면 클릭한 열의 문자열에서 빈 칸(space)과 점(dot)을 기준으로 사용자 정의 객체 이름을 추출하여 상세 정보를 제공한다. (그림 8)은 (그림 7)의 테이블 함수 정의 구문에서 첫 번째 열을 클릭했을 때 호출되는 CURSOR\_PKG 객체에 대한 상세 정보 표시 화면이다.



(그림 8) 사용자 정의 객체 상세 정보

사용자 정의 객체에 대한 정보로는 객체 타입과 생성 날짜, 현재 상태 등의 정보와 객체 생성시 사용한 텍스트 원본에 대한 정보를 표시한다. 객체에 대한 기본 정보를 추출하기 위해 서버 모듈은 ALL\_OBJECTS 디셔너리 뷰를 조회하고, 객체 생성시 사용한 텍스트 원본에 대한 정보는 ALL\_SOURCE 디셔너리 뷰를 참조한다.

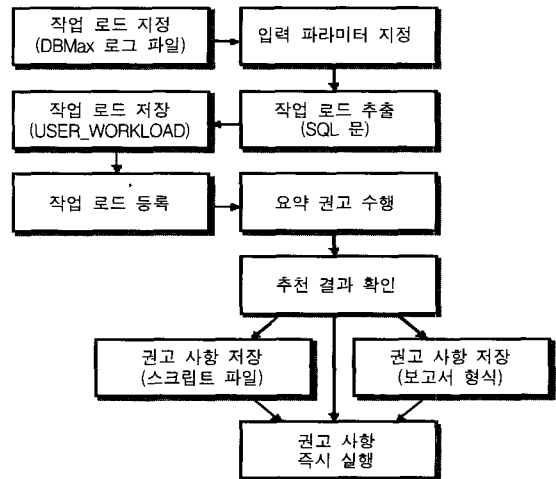
4.2.2 성능 지원부

새롭게 추가된 성능 지원부는 DBMax가 제공하는 시스템과 세션, SQL 레벨의 로그 파일에 기록된 작업 로드를 대상으로 요약 권고 기능을 수행한다. 요약 권고 기능을 수행하기에 앞서 정확한 결과를 얻기 위해서 우선 테이블들에 대한 통계 정보를 수집해야 한다. 그러나 대상 로그 파일에 기록된 모든 테이블들에 대해서 통계 정보를 수집한다는 것은 시간과 성능 상의 문제가 있기 때문에 본 연구에서는 스키마에 대한 통계 정보를 수집한다. 우선 로그 파일에서 스키마 목록을 추출하고, 각 스키마에 대해서 DBMS\_STATS.GATHER\_SCHEMA\_STAT 프로시저를 호출하여 통계 정

보를 수집하였다. DBMax 로그 파일에서 SQL 문을 추출하는 부분은 기존의 로그 재생 모듈에 포함되어 있는 기능을 이용한다.

오라클 9i에서는 요약 권고 기능을 지원하기 위해서 DBMS\_OLAP 패키지를 제공하는데 이 패키지 내에는 요약 권고 대상이 되는 작업 로드를 등록하고, 필터를 적용하고, 실제 요약 권고 기능을 수행하는 다양한 프로시저들이 정의되어 있다. 확장 모듈에서는 DBMS\_OLAP 패키지 내의 프로시저들을 적절히 사용하여 DBMax의 요약 권고 모듈을 구현하였으며 지정한 DBMax 로그 파일에서 SQL 문을 추출하고 작업 로드를 등록하는 모든 과정을 서버 모듈에서 자동으로 처리한다.

(그림 9)은 확장 아키텍처의 요약 권고 흐름을 나타낸 것이며 <표 9>는 요약 권고 기능을 구현하기 위해 확장 서버 모듈의 프로시저가 참조하는 DBMS\_OLAP 패키지의 프로시저[18] 목록이다.



(그림 9) 확장 아키텍처의 요약 권고 흐름

〈표 9〉 요약 권고 기능을 위한 참조 프로시저

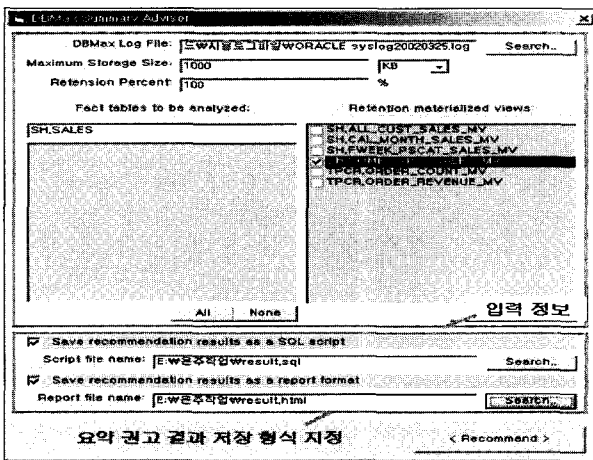
DBMS_OLAP 패키지의 프로시저	설 명
CREATE_ID	작업 로드, 필터, 권고 ID 생성
LOAD_WORKLOAD_USER	로그 파일에서 추출한 사용자 정의 작업 로드 로드
RECOMMEND_MVIEW_STRATEGY	지정한 작업 로드에 대해 실제화된 뷰 분석 및 권고
GENERATE_MVIEW_SCRIPT	요약 권고 결과를 SQL 스크립트로 저장
GENERATE_MVIEW_REPORT	요약 권고 결과를 보고서 형식으로 저장

입력 정보로는 DBMax 로그 파일명과 실제화된 뷰를 저장할 때 사용할 수 있는 최대 저장 공간 크기, 분석할 사실 테이블 및 보유할 실제화된 뷰 목록, 보유 퍼센트 값을 지정한다.

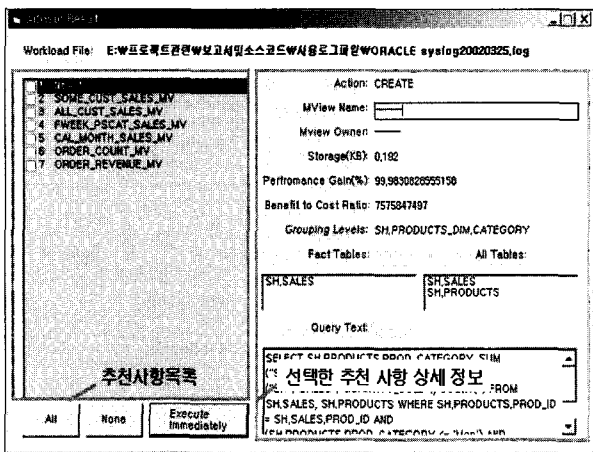


서버 모듈은 사용자가 지정한 로그 파일에서 스키마 정보와 SQL구문을 추출하여 데이터베이스 내의 테이블에 저장하고 LOAD\_WORKLOAD\_USER 프로시저를 호출하여 작업 로드를 등록한다. 사용자가 화면을 통해 입력한 값들을 가지고 RECOMMEND\_MVIEW\_STRATEGY 프로시저를 호출하여 실제 요약 권고 기능을 수행한다. 요약 권고기능이 완료되면 SYSTEM.MVIEW\_RECOMMENDATIONS 테이블에서 요약 권고 결과를 조회하여 사용자에게 제시한다.

(그림 10)은 요약 권고 기능을 위한 사용자 인터페이스로 (그림 10)(a)는 요약 권고 기능을 수행하기 위한 입력 화면이며 (그림 10)(b)는 로그 파일을 이용하여 요약 권고 기능을 수행한 후 표시되는 결과 화면이다.



(a) 입력 화면



(b) 요약 권고 결과

(그림 10) 요약 권고 모듈의 사용자 인터페이스

(그림 10)(a)에서 요약 권고 결과를 스크립트 파일이나 보고서 형식으로 저장하도록 지정할 수 있다. 입력 사항을 지정한 후 추천 버튼을 누르면 요약 권고 기능을 수행하는 확장 서버 모듈의 프로시저를 호출하고 (그림 10)(b)의 결과 화면을 표시한다. (그림 10)(b)는 요약 권고 결과 도출된 추천 사항 목록을 표시하고 특정 추천 사항을 선택하면 우

측에 상세 정보를 표시한다. 추천 사항은 새로운 실체화된 뷰에 대한 생성 권고 뿐만 아니라 지정한 작업 로드들에 대해 비효율적인 실체화된 뷰에 대한 삭제 권고와 보유 권고 사항을 모두 포함하고 있다. 추천 사항 목록에서 특정 권고 사항을 체크하고 즉시 실행 버튼을 클릭하면 권고 결과를 즉시 실행하여 시스템에 반영한다.

### 4.3 비교 평가

기존의 DBMax는 OLTP 시스템의 성능 관리를 목적으로 설계되었기 때문에 데이터 웨어하우스 시스템에 적용할 때 많은 취약점이 드러난다. 데이터 웨어하우스 환경에서 수행되는 대부분의 질의는 데이터 조작보다는 읽기 작업이 주가 되기 때문에 성능 문제가 발생한 경우 대부분은 인덱스 문제나 SQL 문의 결함, 기대했던 질의가 재작성 되지 않은 경우라고 판단할 수 있다. 문제 SQL 문을 추출한 후에 질의가 재작성 되지 않아 성능 문제가 발생했음을 확인한 경우 기존의 DBMax는 재작성 정보를 표시하지 않기 때문에 문제의 원인을 파악할 수가 없다. 그러나 확장 기능을 적용하면 해당 질의에 대한 재작성 정보와 관련 파라미터 값들, 현재 세션이 참조할 수 있는 실체화된 뷰 목록을 표시하기 때문에 재작성과 관련된 성능 문제의 근본적인 원인을 신속히 파악할 수 있다.

기존의 DBMax가 제공하는 시스템/세션 레벨의 로그 파일은 성능 문제 상황을 종료하기 위해 시스템을 다시 시작한 경우 손실되는 중요 정보를 저장하여 사후 분석을 지원한다. 로그 파일에는 해당 데이터베이스에서 수행되는 중요 작업 로드가 저장되어 있는데 기존의 DBMax는 로그를 기록하고 재생하는 기능만 제공하기 때문에 로그 파일을 대상으로 요약 권고 기능을 수행하기 위해서는 관리자가 별도로 처리해야 하는 작업이 많았다. 그러나 DBMax 확장 기능을 이용하면 관리자가 작업 로드로 사용할 로그 파일명을 지정하면 필요 정보를 추출하여 작업 로드로 등록하고, 요약 권고 기능을 수행하는 모든 과정이 자동으로 이루어진다.

따라서 본 논문에서 기술한 DBMax의 확장 기능을 이용하여 데이터 웨어하우스 시스템의 성능을 관리한다면 SQL 튜닝시 보다 풍부한 성능 관련 정보 및 스키마 객체 정보를 확인할 수 있으며 이러한 정보를 이용하여 모니터링 대상 시스템의 성능을 최적화할 수 있을 것이다.

## 5. 결론 및 향후 연구 과제

본 연구에서는 OLTP용 시스템을 위한 국산 성능 관리 도구인 DBMax를 데이터 웨어하우스 시스템에 적합하도록 기능을 확장하였다. 우선 DBMax를 데이터 웨어하우스 시스템에 적용할 때 고려해야 할 요구사항을 오라클 9i의 요약 관리와 ETL 기능을 중심으로 살펴보고 이를 지원하는 확장 아키텍처를 설계 및 구현하였다. 확장 아키텍처는 성능 모니터링과 로그 기록 및 재생을 수행하는 기존의 DBMax 모듈

과 SQL 튜닝, 성능 지원을 위한 확장 모듈이 상호 연동되는 구조로 이루어져 있다.

확장 모듈의 튜닝 지원부에서는 부하 SQL에 대한 실행 계획 분석과 재작성 정보 분석, 스키마 객체 정보 분석을 통해 SQL 튜닝 기능을 강화하였다. 구체적으로 테이블, 인덱스 외에 데이터 웨어하우스 시스템에서 보편적으로 사용되는 실체화된 뷰와 외부 테이블, 테이블 함수에 대한 스키마 정보와 질의 재작성 정보를 분석하여 제공하였다. 성능 지원부는 DBMax가 제공하는 로그 파일을 이용하여 성능 관리 대상 시스템에서 유용하게 사용될 수 있는 실체화된 뷰에 대한 요약 권고 기능을 수행한다.

향후 구현한 요약 권고 기능을 보완하여 특정 필터 기준에 의해 작업 로드를 제한할 수 있도록 한다. 우선 로그 파일에서 SQL 문을 추출할 때 실체화된 뷰가 유용하게 사용될 수 있는 SELECT, INSERT~SELECT, UPDATE~SELECT 구분만을 추출하도록 한다. 또한 작업 로드를 등록한 후 우선 순위나 마지막 수행된 날짜, 질의가 수행된 응용 프로그램 등을 조건으로 작업 로드를 제한할 수 있도록 보완한다. 다음으로 본 연구에서 제시한 확장 아키텍처를 보완하여 요약 관리와 ETL 기능 이외에 오라클이 제공하는 다양한 데이터 웨어하우스 관련 기술들을 분석하고 이들에 대한 성능 관리 기능을 추가한다.

### 참 고 문 헌

[1] Oracle Corp., Oracle Enterprise Manager Database Tuning Pack, June, 2001.  
 [2] Benoit Dageville, Mohamed Zait, SQL Memory Management in Oracle 9i, VLDB 2002.  
 [3] Agrawal S., Chaudhuri S., Narasayya V., Automated Selection of Materialized Views and Indexes for SQL databases, VLDB 2000, pp.496-505.  
 [4] Guy M. Lohman, Sam S. Lightstone, SMART : Making DB2 (more) Automatic, VLDB 2002.  
 [5] (주)엑셈, <http://www.ex-em.com/>.  
 [6] 조종암, 이상원, DBMax : 데이터베이스 성능 모니터링 도구, 한국데이터베이스학회 2001 발표논문집, 제17권 제2호, pp. 287-294, 2001.  
 [7] 조종암, Technical Notes : Rapid Database Problem Identification, Oracle Korea Magazine, Vol.27, pp.93-98, 2001.  
 [8] (주)엑셈, BizMax User's Guide, 2002.  
 [9] (주)엑셈, DBMax 사용자 매뉴얼, 2001.  
 [10] Surajit Chaudhuri, Umesh Dayal, An Overview of Data Warehousing and OLAP Technology, ACM SIGMOD Record, 1997.  
 [11] William A. Giovinazzo, Objected-Oriented Data Warehouse Design, Prentice Hall PTR, 2000.

[12] Oracle Corp, Oracle 9i Data Warehousing Guide, June, 2001  
 [13] Oracle white paper, Oracle 9i Materialized Views, May, 2001.  
 [14] Oracle Corp., Oracle 9i SQL Reference, June, 2001  
 [15] Oracle white paper, ETL Processing within Oracle 9i, May, 2001.  
 [16] Jonathan Coldstein, Per-Ake Larson, Optimizing Queries Using Materialized Views : A Practical, Scalable Solution, ACM SIGMOD, 2001.  
 [17] Oracle Corp., Oracle 9i Reference, June, 2001.  
 [18] Oracle Corp., Oracle 9i Supplied PL/SQL Pack-ages and Types Reference, June, 2001.



### 김 은 주

e-mail : jussi@ewha.ac.kr

2001년 이화여자대학교 컴퓨터학과(학사)

2003년 이화여자대학교 과학기술 대학원

컴퓨터학과(공학석사)

2003년~현재 삼성전자

관심분야 : 데이터베이스 성능 관리, 데이터

웨어하우스, OLAP, XML



### 용 환 승

e-mail : hsyong@ewha.ac.kr

1983년 서울대학교 컴퓨터공학과(학사)

1985년 서울대학교 대학원 컴퓨터공학과

(공학석사)

1994년 서울대학교 대학원 컴퓨터공학과

(공학박사)

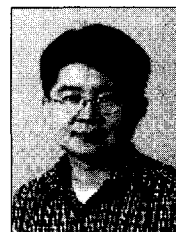
1985년~1989년 한국전자통신연구소 연구원

2002년 IBM T. J. Watson 연구소 객원 연구원

1995년~현재 이화여자대학교 컴퓨터학과 부교수

관심분야 : 데이터베이스, 데이터마이닝, XML, 멀티미디어

처리, 바이오 인포매틱스



### 이 상 원

e-mail : wonlee@ece.skku.ac.kr

1991년 서울대학교 컴퓨터공학과(학사)

1994년 서울대학교 대학원 컴퓨터공학과

(공학석사)

1999년 서울대학교 대학원 컴퓨터공학과

(공학박사)

1999년~2001년 한국 오라클

2001년 이화여자대학교 컴퓨터학과 BK21 연구교수

2002년~현재 성균관대학교 정보통신공학부 조교수

관심분야 : 데이터베이스 튜닝, 데이터웨어하우스, 데이터

마이닝, OLAP