

네트워크 컴퓨팅을 위한 아키텍처

개방형 표준 기반의 오라클 네트워크 컴퓨팅 아키텍처는 정보 처리 부서가 당연하게 되는 상호 호환성 문제를 덜어주고, 솔루션을 실행 전개하는데 더 노력을 기울일 수 있도록 해주기 위한 것이며, 그 핵심을 이루는 것이 바로 가장 강력한 객체 관계형 데이터베이스 서버인 Oracle8이다.

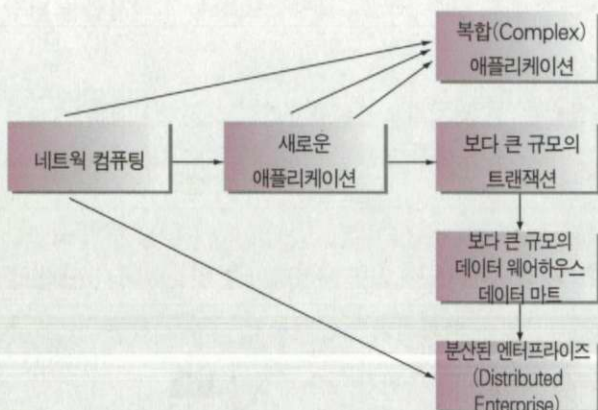
최기영/한국오라클 마케팅 대리

네트워크 컴퓨팅 모델

메인프레임에서 클라이언트/서버로, 다시 네트워크 컴퓨팅 환경으로의 컴퓨팅 환경 변화가 가속화되면서, 보다 가벼운(thin) 클라이언트/다계층(N-tier) 구조의 무거운(fat) 서버로의 전이와 함께 웹의 활용, 다양한 비정형 데이터 타입의 지원, 대규모 사용자 지원 등이 필수적인 조건으로 거론되고 있다.

또한 분산 컴퓨팅 모델과 기존 시스템을 모두 지원하면서 앞의 조건들을 만족시킬 수 있어야 한다는 것이 네트워크 컴퓨팅 환경에 적합한 데이터베이스 시스템에 대한 요구 사항들이다.

〈그림 1〉의 다이어그램을 바탕으로 다시 표현한다면,



〈그림 1〉 네트워크 컴퓨팅 환경에 적합한 데이터베이스 시스템의 요구 사항

새로운 애플리케이션의 요구사항들을 만족시킬 수 있는 객체 솔루션으로 네트워크상에 산재한 어떠한 객체들이라도 지원할 수 있어야 하고, 보다 뛰어난 확장성과 고가용성으로 OLTP를 위한 뛰어난 솔루션, 보다 나은 품질의 웨어하우스를 구축하며 향상된 관리를 제공하는 데이터 웨어하우스 솔루션, 보다 나은 전개 및 손쉬운 분산 관리를 위한 분산된(Distributed) 솔루션 등이 모두 만족될 수 있는 데이터베이스가 진정한 네트워크 컴퓨팅을 위한 데이터베이스라는 것이다.

일반적으로 기업들이 정보 시스템을 개방형, 네트워크 기반의 아키텍처로 발전시켜 나아갈 때는 다음과 같이 두 가지의 근본적인 문제들과 마주치게 된다.

첫째, 복수의 계층(multi-tier), 복수의 플랫폼(multi-platform) 환경에서 기존의 데이터와 새로운 데이터 사이에 제기되는 상호 호환성(interoperability)과 관리 문제.

둘째, 그러한 환경 하에서 실행 전개를 위해 가장 적합한 제품들을 구별해 내는 문제.

개방형 표준 기반의 오라클 네트워크 컴퓨팅 아키텍처(Network Computing Architecture)는 위와 같이 정보 처리 부서가 당연하게 되는 상호 호환성 문제를 덜어주고, 솔루션을 실행 전개하는데 더 노력을 기울일 수 있도록 해주기 위한 것이며, 그 핵심을 이루는 것이 바로 가장 강력한 객체 관계형 데이터베이스 서버인 Oracle8이다.

보다 강력한 데이터베이스 서버

네트워크 기반의 애플리케이션들은 네트워크 데이터베이스에 대해 매우 다른 성능을 요구하는데, Oracle8은 이를 해결할 수 있도록 독특하게 설계되었다.

우선, 네트워크 데이터베이스는 반드시 수직적인 확장성이 뛰어나야만 한다. 멀티미디어 타입과 데이터 웨어하우스가 전체적인 데이터의 양을 크게 증가시키므로, 데이터베이스는 대규모의 데이터를 용이하게 관리할 수 있어야만 한다.

Oracle8은 숫자, 문자 및 멀티미디어 데이터를 포함하여 PetaByte(1024 TeraByte) 크기의 데이터베이스를 지원할 수 있도록 설계되어 있다. 게다가 성능, 데이터의 관리, 가용성에 있어서 큰 진척을 이루어 낸 분할(partition) 및 향상된(Advanced) 병렬 처리와 같은 대용량 데이터베이스 처리 기능이 제공된다.

그리고, 네트워크 데이터베이스는 반드시 수평적인 확장성도 뛰어나야 한다. 웹상에 전개된 애플리케이션들은 사용자 수에 있어서 엄청난 증가를 초래하므로 단순히 네트워크 자체가 병목현상의 원인이 되는 것은 아니다. 수만 명의 사용자를 지원할 수 있도록 설계된 향상된 네트워크 기술로 Oracle8은 어떠한 사용자 규모라도 지원할 수 있다. 마지막으로, 네트워크 기반의 애플리케이션들은 빈번하게 복수의 데이터를 저장하게 되므로, 네트워크 데이터베이스는 네트워크 전반에 걸쳐 애플리케이션들을 지원해줄 수 있어야만 한다.

복수의 시스템간에 발생하는 상호 종속성 때문에 일어나는 지연 문제들은 Oracle8의 향상된 질의(Advanced Queuing) 기술로 해소된다. 이 향상된 질의 기술은 병목현상을 제거하고 네트워크 기반 애플리케이션의 전반적인 확장성을 크게 늘려준다.

또한, 사용과 관리의 용이성도 네트워크 기반 애플리케이션에서 강조되는 측면이다.

Oracle8의 서버가 관리해주는 백업 및 복구 기능은 관리자의 업무를 현저하게 단순화시켜, 사람의 실수로 인해 문제가 발생할 수 있는 소지를 줄임으로써 손실을 최소화할 수 있도록 많은 부분이 자동화되었다. 그리고 오라클 엔터프라이즈 매니저

(Oracle Enterprise Manager)가 Oracle8 데이터베이스 및 시스템, 애플리케이션들의 관리, 이벤트 모니터링, 튜닝을 중앙 집중 콘솔에서 드래그앤드롭으로 완벽하게 관리할 수 있도록 해준다.

네트워크 기반의 아키텍처는 다양한 하드웨어와 소프트웨어 플랫폼으로 구성되므로, 일반적으로 많이 사용되는 주요 플랫폼(메인프레임, 병렬 서버, 데스크 탑 시스템 등)을 지원할 수 있는가 하는 문제도 고려되어야 될 사항이다. Oracle8은 모든 주요 플랫폼에 대해 공급이 되며, 궁극적으로 운영체계에 구애받지 않고 네트워크 컴퓨팅 아키텍처의 개방형 표준을 따르게 된다.

Oracle8, OLTP·데이터웨어하우스를 위한 기능

Oracle8은 네트워크에 기반한 웹 전자상거래든 전통적인 기업 애플리케이션이든 구애받지 않고 어떤 기업이라도 강력함과 강건함 그리고 유연성을 제공받을 수 있도록 해준다.

이 글에서는 Oracle8이 가지고 있는 객체관계형 데이터베이스 서버로서의 특징을 주로 설명하겠지만, 그 외에 간과할 수 없는 OLTP 및 데이터 웨어하우스를 위한 Oracle8의 주요 특징들을 간략하게 먼저 살펴보기로 하자(표 1).

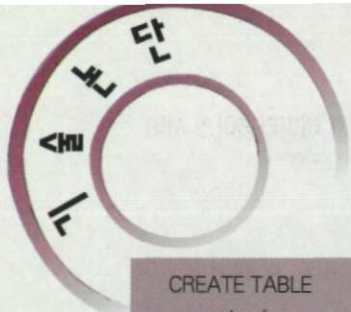
테이블 파티션(Partition) : 성능과 관리 기능 동시 개선

키값을 기준으로 데이터를 저장하며 한 테이블에 64K 파티션까지 가능하다. 각각의 파티션은 각각의 테이블 스페이스를 가질 수 있다.

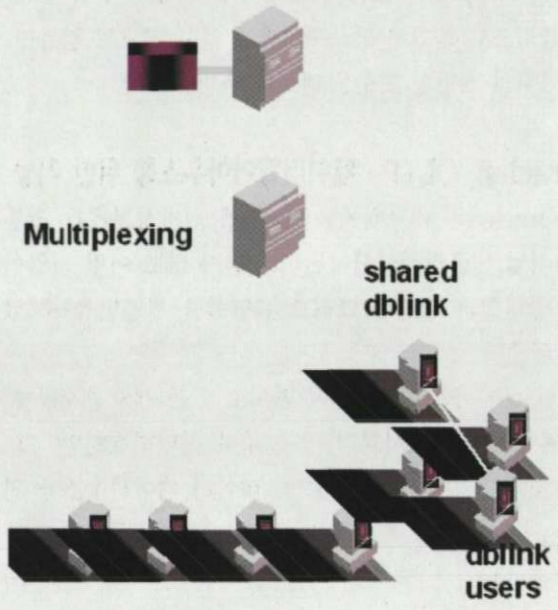
파티션 단위로 저장, 파티션 단위 인덱스 생성, import/export, 로딩, 파티션 단위 truncate, 이동, 테이블과 파티션 간의 상호 전환 기능 등을 통해 관리가 더욱 쉬워진다.

〈표 1〉 VLDB를 위해 크기의 한계점을 극복, 개선한 Oracle8

Size 타입	Oracle7	Oracle8
데이터베이스 크기	2-32 TeraByte	512 PetaByte
데이터베이스당 데이터 file들의 수	1-4 K	64-256 M
테이블 당 칼럼 수	254	1000
테이블 당 확장 영역	무제한 (7.3이후)	무제한
테이블 당 LOB 칼럼 수	하나	1000
칼럼 크기 : CHAR VARCHAR	254바이트	2000바이트
	2000바이트	4000바이트



```
CREATE TABLE
  sales (acct , rep , amt , weekno )
  PARTITION BY RANGE(weekno) ...
  ( PARTITION P1 VALUE LESS THAN(4),
    TABLESPACE data1,
  PARTITION P2 VALUE LESS THAN(8),
    TABLESPACE data2, ...
  PARTITION P52 VALUE LESS THAN(MAXVALUE)
    TABLESPACE data52 );
```



〈그림 2〉 Net8의 멀티플렉싱(Multiplexing) 기능

한편, 디스크에 장애가 발생할 경우 해당되는 파티션만 영향을 받게 되므로 보다 나은 가용성을 보장받을 수 있으며, 파티션 단위로 백업 및 복구가 가능해진다.

또한, SQL문 처리에 불필요한 파티션은 이용하지 않을 수 있고, 파티션을 인식하여 조회문을 최적화할 수 있으므로 성능 상의 개선도 제공되게 된다.

데이터 이중화(Replication)

Oracle8에서는 혁신적으로 데이터 이중화의 성능을 향상시켰다. 변화된 정보를 병렬적으로 전달하고 내부 트리거, 마스터 로그 인덱스들을 사용한다. LOB 칼럼도 복제가 가능하며, 서브쿼리 스냅샷(Subquery snapshot)들은 데이터 서브세팅(subsetting)이 가능하고 프라이머리 키 스냅샷(Primary key snapshot)은 관리가 쉬워지는 스냅샷 기능 향상이 제공된다.

Net8

기존의 Oracle 미들웨어인 SQL*Net을 더욱 보강하고 TP 모니터적인 기능과 통합성을 강화한 것이 Net8이다. Net8은 간단하게 구성할 수 있으며, tnsnames.ora 없이 동적으로 목표를 찾으므로 정적인 구성 파일이 필요 없다. 네트워크에 대한 단 한번의 'Single Network Sign-on'으로 오라클 보안 기능과 글로벌 사용자 이름, 역할, 상호 사용자 권리 부여가 가능하다.

Net8의 멀티플렉싱(Multiplexing) 기능은 하나의 접속으로 여러 개의 세션에 접근 가능하도록 해준다.

커넥션 풀링(Connection pooling) 기능은 보다 많은 수의 사용자를 융통성있게 지원할 수 있도록 idle 시간에는 자동적으로 물리적인 연결이 제거되게 해준다.

DBLINK concentration은 여러 dblink 사용자가 서버 간에 연결을 공유하도록 해주며, 향상된 DB Queue로 비동기적인 메시지 전달 방식으로 TP 모니터 기능을 통합 제공한다.

기타 향상된 기능

리버스 키(Reverse key) 인덱스와 I/O 공간을 줄인 인덱스만의 테이블 지원과 같은 새로운 인덱스 타입이 지원된다.

기존의 오라클 병렬 서버(Oracle Parallel Server)는 향상된 성능과 함께 튜닝과 이식이 쉬워지며 메모리, 코드 패스(path)를 감소시켰다. 조회시 오버헤드도 감소하였고 ping을 개선하였다.

캐쉬를 이용하여 체크 포인트를 빠르게 하였으며, PL/SQL의 수행 속도도 더욱 빨라졌다. 제약 조건에 대한 체크를 문장 수행 뒤가 아닌 커밋(Commit)시 수행하도록 하여 커밋 전까지는 제약 조건을 무시할 수 있도록 해준다. 만일 문제가 발생하면 롤백(Roll-back)을 하게 된다.

Oracle8, 객체 관계형 DBMS

객체 관계형(Object-Relational) 데이터베이스는 기존의 관계형 데이터베이스상에 구축된 정보들을 '그대로' 사용할 수 있고, 동시에 객체 기반의 정보 저장 및 검색, 응용 프로그램의 작성이 가능하다.

객체 관계형 데이터베이스는 기본적으로 ANSI

(American National Standard Institute)에서 제정한 관계형 데이터베이스의 표준 모델인 SQL92 데이터 모델에 기반하고 있으며, 이 바탕 위에 객체 지향 개념을 지원하기 위해 확장된 것이다.

따라서, 객체 관계형 모델은 현재 관계형 데이터베이스를 사용하고 있는 사용자가 객체 지향 개념을 필요로 할 때에 가장 적합한 모델이라고 볼 수 있다. 현재로서는 이러한 객체 관계형을 위한 표준은 없으며, SQL3라고 불리는 표준화 작업이 현재 진행되고 있다.

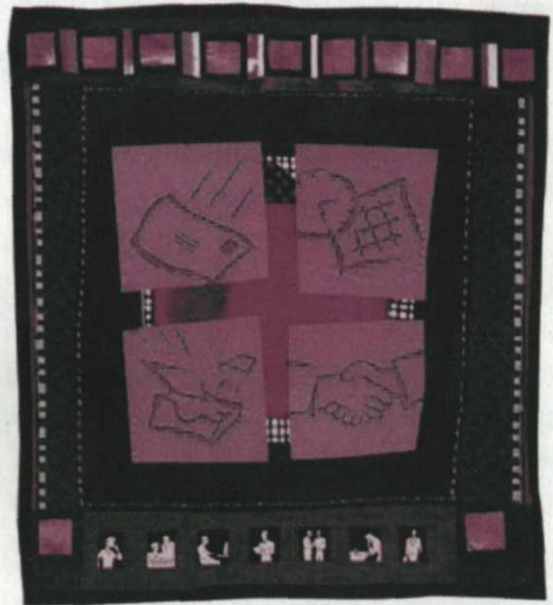
기본적으로 Oracle8이 ORDBMS(Object Relational DataBaseManagement System)로 개발된 것은 고객의 개발 생산성을 높이고, 풍부한 데이터 타입들을 지원하여 멀티미디어 데이터의 효과적인 처리 및 관리와 같은 다양성을 제공하자는 것이다. 그러면서도 서로 다른 제품 기술들이 단순히 조합되고 연결되는 것이 아닌 기존의 현업에서 입증된 서버(Oracle 7.3)에서 발전되어 강력한 단일의 서버로 가능하게 한다는 것이 특징이다.

이 단일 서버라는 것을 다시 부연 설명하자면, 네트워크 컴퓨팅 시대의 OLTP 시스템과 분산 시스템, 네트워크 컴퓨팅 시대의 데이터 웨어하우스 시스템, 네트워크 컴퓨팅 시대의 애플리케이션 개발과 객체지향의 조건들을 Oracle8, 하나의 서버가 모두 만족시켜줄 수 있다는 것이다.

이의 검증을 위해, Oracle8은 지난 1996년 8월부터 10개월여의 기간 동안, 전세계 28개국 270여 Beta 고객들을 대상으로 충분한 테스트의 과정을 거쳤으며, 이 Beta 고객들은 47%가 새로이 추가된 객체 관계형 기술을 활용하고 있고, 53%는 Oracle8의 보다 강력한 대용량 데이터베이스, OLTP, 데이터웨어하우스의 용도를 주목적으로 활용하고 있다.

특히, 많은 사람들이 관심을 가지고 있는 객체 관련 기술을 채택할 때 오라클은 객체지향 응용 프로그램의 쉬운 개발을 가능하도록 하면서 신뢰성 있는 데이터베이스를 공급한다는 원칙에 충실했다.

이전의 것을 무시하고 완전히 새롭게 바꾸는 혁명적인 방법이 개발은 더 쉽고 빠를 수 있겠지만, 기존에 많이 출시된 객체지향 데이터베이스들이 외면당하는 이유는 완전히 새롭게 만든 것이 가지는 허점때문이라고 볼 수 있다.



Oracle8에서는 혁신적으로 데이터 이중화의 성능을 향상시켰다. 변형된 정보를 병렬적으로 전달하고 내부 트리거, 마스터 로그 인덱스들을 사용한다.

반면에 차곡차곡 새로운 기능을 추가해가며 조화를 이루어 나아가는 것은, 개발에 소요되는 시간은 더디지만 이전의 것을 충분히 포괄하기 때문에 무리가 없다. 이미 사용자들이 쓰고 있는 오라클의 장점을 모두 포함하면서 객체지향 데이터베이스로 발전시키는 것이 오라클의 선택이었다.

높은 병렬성, 수천명의 동시 사용자 환경에서의 데이터 무결성, 하드웨어에 따라 적합한 성능을 내는 확장성, 분산 환경에서 트랜잭션 관리, 진정한 로우레벨 록(Row-Level Lock) 메커니즘, 백업과 복구의 견고성 등등, 오라클을 데이터베이스 업계에서 최고의 자리로 올려놓았던 강력한 기능을 가지면서 새로운 객체지향 데이터베이스를 개발한 것이다.

그러나 이것이 단순한 Wrapper 수준의 제품을 만드는 것은 결코 아니다.

데이터베이스 엔진은 이전의 것을 그대로 사용하면서 외부에서 보는 인터페이스만 객체지향 특징을 지원하도록 하는 'Mapping layer' 만 개발함으로써 ORDBMS적



인 제품을 공급하는 회사도 있지만, 오라클의 분명한 입장은 우수한 관계형 데이터베이스를 기반으로 커널 차원에서 객체지향 기능을 추가하는 것이다.

객체관계형 데이터베이스(ORDBMS)로서의 Oracle8이 설계된 기본 방향은 다음과 같다.

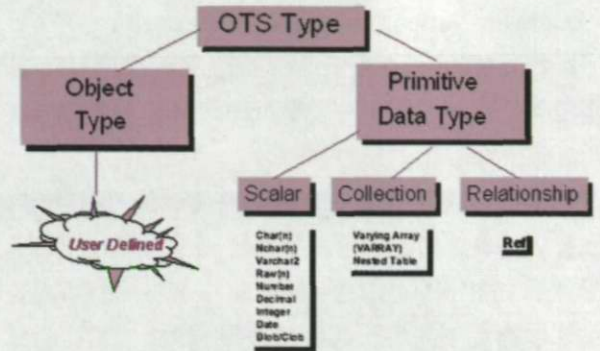
- 비즈니스 환경에서 생기는 객체를 사용자가 모델링할 수 있는 능력을 제공한다. 이것은 사용자 정의 타입을 데이터베이스 내에서 구현할 수 있도록 하는 타입 시스템이 존재함으로써 가능하다. 이런 타입들은 객체 분석 결과에서 나온 응용 프로그램 객체와 밀접히 연관되고 number, char와 같은 내장 데이터 타입처럼 다루어진다.
- 데이터베이스 내에 존재하는 데이터를 접근하기 위해서 필요한 다양한 방법을 제공한다. 이것은 이전에 응용 프로그램에서 정의한 객체와 데이터베이스에 저장된 객체 구조가 상이함으로써 생기는 임피던스 불일치를 극복할 수 있다.
- 멀티미디어 데이터 타입을 데이터베이스 내에서 관리하도록 지원함으로써 데이터베이스가 관리할 수 있는 데이터 처리를 최대한 확장할 수 있도록 지원한다. 이것은 다른 도메인에 존재하는 데이터 카트리지를 개발할 수 있도록 하는 기본 틀을 제공한다.

Oracle8, 객체 타입과 객체 뷰

Oracle8의 객체 기능 중 가장 기본적인 것이 객체 타입에 관한 것이다.

기본적으로 속성(attribute)과 메소드(method : function)로 구성되는 객체를 새로운 타입이나 테이블로 정의하기 위한 SQL DDL(CREATE TYPE, ALTER TYPE, DROP TYPE), SQL DML(Path notation, 연산자 지원, 새로운 구문 제공), 타입 메소드(Constructor, Member method, ORDER, MAP)의 세부분으로 나누어 질 수 있고 이는 다시 SQL, PL/SQL, Pro*C, OCI(Oracle Call Interface) 등의 인터페이스 차원에서 정의 및 지원된다. 또한, Oracle8은 C++, Java, CORBA/ORB 등과 쉽게 매핑되도록 해준다.

이때, 속성의 타입은 스칼라, 다른 객체 타입, Nested 테이블, Array, 객체 레퍼런스 또는 LOB로 이루어지고, 메소드는 타입에 대한 정의의 일부로서 Constructor들, 멤버 메소드(Member method) 등의 형식으로 구성되



〈그림 3〉 Oracle8에서 객체 타입은 OTS 방법을 통해 확장성이 뛰어나다.

며 사용이 쉽고 효율적인 PL/SQL, C/C++로도 만들어진다. 또한 이 메소드를 ADT(Abstract Data Type)에 대한 SQL 질의 내에서 사용할 수 있다.

객체 뷰(view)는 관계형 SQL 스키마에 대한 객체 뷰를 말하며 union, WITH OID 구문 내에서의 디폴트(DEFAULT) 구문, LOB 및 칼럼들을 포함하는데 객체 뷰 인스턴스(view instance)를 통하여 액세스하도록 한다. 프라이머리 키(primary key)가 OID가 되게 하여 관계형 테이블에 대해 객체 뷰를 제공하는 것이며 'INSTEAD OF' 트리거를 통해 업데이트시켜 줄 수 있다. 또한 객체 타입에 대해서는 기존의 관계형 뷰를 제공해 유연성을 높여준다.

개방형 시스템(Open Type System)

Oracle8에서 객체 타입은 〈그림 3〉과 같이 OTS(Open Type System)라는 체계적인 방법을 통해 확장성이 뛰어나면서도 포괄적이고 쉽게 처리해준다.

객체 타입의 관리는 OTM(Open Type Manager)으로 공통적인 인터페이스를 제공하여 객체의 타입들을 동적으로 create, access, update, delete 할 수 있도록 해주며, OTS의 스펙에 따라 타입의 semantics를 구현한다.

객체 프로그래밍 인터페이스로는 기본적으로 객체 캐시로 확장된 콜 레벨(call level) 인터페이스를 제공하는데, 데이터베이스 타입 정의로부터 C와 C++의 헤더 파일들을 생성하기 위한 스키마 틀들을 제공한다. CORBA를 지원하기 위해서는 ADT 정보로부터 IDL(Interface Definition Language)을 생성한다. Pro*C/C++는 자연스러운 객체 액세스를 통해 C/C++를 위한 프리컴파일러(pre-compiler)를 제공한다.

복잡한 객체 검색은 객체 검색의 지원과 단일 네트워크 라운드 트립(round-trip) 내에서의 객체 참조를 통해 성능 개선을 하며, 사용자측 즉, 클라이언트쪽의 객체 캐싱도 지원한다. 스토리지 최적화(Storage optimization)은 Packed/Unpacked Storage로, 사용자 정의 객체에 대해서는 ADT에 대한 인덱스들의 정의로, 또 row/ADT locking보다 작은 Attribute locking을 지원한다.

Oracle8의 확장성, 데이터 카트리지

Oracle8은 표준 객체들은 데이터베이스 내에 포함되게 하고, 추가로 확장되는 객체 타입들은 데이터베이스 외부에 데이터 카트리지 형태로 플러그인 될 수 있게 함으로써 거의 무한적인 객체의 확장성과 프로그램 가능성을 보장한다.

Oracle8 데이터 카트리지는 PL/SQL, C/C++ 또는 Java로 IDL 매핑과 데이터베이스 서버 내에 삽입하는 방식으로 작성될 수 있으며, 질의 최적화, 액세스 메소드와 데이터베이스 관리 유틸리티들 같은 서버에서 가능한 서비스들과 함께 자기 자신을 등록할 수 있다.

기본적으로 Oracle8에서 제공할 컨텍스트 카트리지, 스페셜 카트리지, 비디오 카트리지, 이미지 카트리지, 메시징 카트리지, 도큐먼트 카트리지 외에, ISV (Independent Software Vendor), SI(System Integrator), ORB 공급사, 개발 전문업체, 시스템 공급사 등 다양한 회사들을 통해 카트리지를 제공하게 될 것이다.

Oracle8의 객체 기능 발전 방향

Oracle8은 데이터베이스 내에서 사용자 객체를 정의할 수 있을 뿐만 아니라 다양한 언어로 그 타입과 연결되는 메소드를 구현할 수 있다. PL/SQL, C, C++ 뿐만 아니라 최근 네트워크 환경에서 표준으로 부상하는 JAVA도 지원한다. 특히 JAVA는 JDBC, J/SQL과 같이 데이터베이스와 접할 수 있는 다양한 인터페이스를 고안하고 있다.

뿐만 아니라 OID로 관리되는 객체 타입, 참조 타입, 콜렉션 타입, 객체들 간의 인터페이스, 상속(Inheritance), 다형성(Polymorphism), 사용자 정의 인덱스와 같은 기능을 이후의 버전에서 차례차례 추가될

예정이다.

그러나 오라클은 기존의 ODBMS가 가지는 기능에 대해서 신중한 입장을 취하고 있다. 현재 ODBMS가 가지는 기능은 제품마다 상이하고 그 기능도 객체 기술의 범주에서 구현된 것이 있는가 하면 객체 기술과는 상관없이 구현된 것들도 있다. 또한, ODBMS만이 가지는 것은 아닌데 현재의 ODBMS가 가지고 있기 때문에 마치 그것이 ODBMS의 고유 기능인 것처럼 비쳐지는 것들도 있다. 이 모든 범주에 대해서 그 실익을 따져서 오라클은 구현 여부를 판단할 예정이다.

어떤 하나의 기술은 그 걸 모습에 따라 판단하는 것이 아니라, 그 기능이 추가됨으로 해서 생기는 관련 문제를 어떻게 극복했느냐에 따라 그 판단은 달라질 수 있다. 객체지향 데이터베이스는 개발시에 생기는 복잡한 문제, 예를 들어 Rule 시스템의 문제, 질의 최적화의 문제, 각 언어의 상이함으로 생기는 인터페이스의 문제, 세션 관리의 문제들을 복합적으로 고려하지 않는다면 형상은 있으나 사용할 수 없는 제품이 될 수도 있다.

Oracle8은 표준 객체들은 데이터베이스 내에 포함되게 하고, 추가로 확장되는 객체 타입들은 데이터베이스 외부에 데이터 카트리지 형태로 플러그인 될 수 있게 함으로써 거의 무한적인 객체의 확장성과 프로그램 가능성을 보장한다.

오라클은 이런 선부른 판단을 내리지 않도록 최대한 노력하고 있으며, 이를 위한 방법론을 산업 및 업계 표준 수준을 통해 제공한다. 즉, Oracle8을 NCA, 세도나(sedona)와 통합된 환경으로 제공하여 정보관리를 위한 진정한 객체 기술, 새로운 차원의 개발 생산성과 유연성을 산업 및 업계 표준에 따라 가능하도록 하는 것이다.

이상과 같이 Oracle8의 객체 기술 중 기본적인 사항들과 OLTP 및 데이터 웨어하우스를 위해 추가, 향상된 기능들을 간략하게 살펴보았다.

보다 우수한 성능 및 확장성, 가용성, 다양성을 고려한 강력한 데이터베이스 서버로 짧은 시간에 복잡한 질의를 신속하게 처리하고, 보다 대용량의 데이터베이스를 효과적으로 관리하며, 보다 많은 수의 온라인 사용자를 지원할 수 있는 것이 네트워크 컴퓨팅의 필수 요구 사항이라면, 이에 대한 탄탄한 솔루션이 바로 Oracle8인 것이다. **DC**