

객체지향 데이터베이스의 동향과 전망



김 형 주
(서울대 컴퓨터공학과 교수)

객체지향 데이터베이스에 이르기까지

1960년대와 70년대 초까지는 IBM의 IMS, MRI System Corp.에서 설계 제작한 System 2000등으로 대표되는 계층형 데이터베이스시스템과 Callname Corp.의 IDMS, CINCOM System의 TOTAL의 네트워크 데이터베이스시스템이 데이터베이스 기술을 주도하였다.

이후 1970년 E. F. Codd에 의해 제안된 관계형 데이터모델은 튼튼한 이론적 기반과 선언적 질의를 통한 사용의 용이성을 바탕으로 기존의 계층형 데이터베이스와 네트워크 데이터베이스를 대치하기 시작하였다. 1976년 최초의 관계형 데이터베이스 시스템인 IBM의 Syste/R과 Berkeley대학의 Ingres가 등장한 이후, 70년대 말과 80

년대 초에 이르기까지 데이터베이스 스키마에 관한 연구, 계층형이나 네트워크 데이터베이스에 비해 상대적으로 떨어지는 성능의 향상을 위한 질의어 자동 최적화에 관한 연구를 비롯하여 동시성 제어, 교착 상태 방지 등 관계형데이터베이스에 관한 수 많은 연구의 결과로 관계형 데이터베이스 시스템은 현재는 거의 모든 컴퓨터 응용분야에서 사용되고 있다.

80년대 초까지는 전세계 컴퓨터 응용분야의 80% 이상이 주로 숫자나 문자 등 정형화된 형태의 자료를 처리하는 사무 중심의 응용이었다. 그래서 그 당시 많은 사람들은 관계형 데이터베이스로서 데이터베이스에 관련된 여러 중요한 문제들이 모두 해결되었다고 생각하였다.

그러나 컴퓨터 하드웨어와 소프트웨어 기술의 발전에 따라 과거에는 현실적으로 불가능하였던 새로운 응용분야가 등장함에 따라 데이터베이스 분야에는 새로운 문제점들이 노출되기 시작하였다. 그 예로 VLSI CAD 분야의 사람들은 기존의 관계형 데이터베이스가 CAD 응용분야에 적합하지 않음을 지적하여 왔는데, 그 이유로는 관계형 데이터 모델이 CAD분야의 설계 패러다임과는 많은 차이가 있어 CAD 응용시스템들이 요구하는 계층적 구조, 또는 수 일 내지 수 십일 간의 작업을 다루기 위한 장기간 트랜잭션, 그리고 이미지 데이터와 같이 길이가 긴 데이터의 처리도

지원하지 않는다는 것이다. 다른 예로 지금까지 NASA는 우주로부터 방대한 양의 정보를 수집해오고 있는데, 이들의 추정으로는 수 년간 인공위성으로부터 송신된 자료의 양이 약 10의 16제곱 바이트 정도라 한다. 이 데이터들은 모두 미래 과학자들의 연구에 필요한 자료가 되기 때문에 모두 저장해 두어야 하고, 관계된 여러 자료들은 빠르게 접근할 수 있어야 하는데 현재의 DBMS로서 매우 어렵다는 것이다.

이렇듯 CAD/CAM/CAE, 인공지능 전문가시스템 셀, 멀티미디어 사무정보 시스템, CASE 등 새로운 응용분야에서는 모두 다양한 형태의 방대한 자료처리기능을 요구하고 있다. 그러나 기존의 데이터베이스 시스템으로는 이러한 새로운 응용분야를 지원할 수 없기 때문에 이를 극복하기 위한 다양한 시도가 이루어지고 있다.

1) System/R Engineering Extension (1982-84)이나 Extending Ingres (1981-83)와 같이 기존의 관계형 데이터베이스 시스템에 새로운 기능들을 추가함으로써 여러 응용의 요구를 만족시키려는 의도가 있다. 이러한 방법은 지금까지 연구된 관계형 데이터베이스에 관한 많은 결과들을 그대로 사용할 수 있다는 장점을 가지고 있다.

2) 기존의 관계형 데이터모델 자체를 확장함으로써 그 한계를 극복하려는 시도가 있다. 이러한 시도의 결과로는 비정형 관계데이터모델(Non-1NF Relational Data Model)과 E. F. Codd의 RM/T 모델 등을 들 수 있다.

3) 데이터베이스 시스템을 생성할 수 있는 소프트웨어 도구(software toolbox)를

제공하여 사용자가 이로부터 필요로 하는 기능들을 선택하여 각 응용에 적합한 데이터베이스 관리시스템을 만들 수 있도록 하는 것이다. 이 중에는 Wisconsin 대학의 EXODUS와 Texas 주립대학교의 GENESIS 등이 있다.

4) 기존의 데이터베이스가 사용하는 데이터모델이 아닌 전혀 새로운 데이터모델로 계산능력을 높이려는 방향이 있다. 이 새로운 데이터모델에는 논리 패러다임, 함수 패러다임, 그리고 의미적 데이터모델 등이 있다.

특히 의미적 데이터 모델은 관계 데이터 모델의 표현능력의 한계를 극복하기 위해 시도되었는데, 이 데이터 모델은 분류화(classification, INSTANCE-OF), 집합화(aggregation, IS-PART-OF), 일반화(generalization, IS-A)를 그 이론적 기반으로 한다. 이 개념들은 각각 클래스(class), 속성(attribute), 클래스 계층구조(class hierarchy) 등의 개념을 통해 객체지향 개념과 이어지고 있다.

객체지향 데이터베이스의 최근 동향

현재 세계 곳곳의 연구소, 대학, 기업 등에서 약 30여 개의 객체지향 데이터베이스 시스템이 개발되었거나 개발중에 있다. 이들은 대부분 객체지향 데이터베이스의 핵심개념인 객체, 객체 식별자, 애트리뷰트, 메소드, 클래스, 클래스 계층구조 상속 등의 개념을 포함하며, 내포된 애트리뷰트(nested attribute)나 집합 값 애트리뷰트 등도 제공하고 있다. 그리고 또한 많은 시스

템들이 다중 사용자를 지원하기 위해서 워크스테이션-서버 모델을 채택하고 있다. 또한 많은 시스템이 프로그래밍 언어와 데이터베이스 시스템을 통합한 형태를 취하고 있으며, 주기억장치에 있는 객체와 디스크에 있는 지속성 객체와의 일치성을 제공하기 위해 객체 버퍼를 관리한다.

이 밖에 다중상속, 버전제어, 멀티미디어 데이터관리 등을 제공하는 시스템도 있다.

지금부터는 현재 개발된 객체지향 데이터베이스 시스템에 대해 간략하게 알아보겠다. 다음은 김 원 박사의 "Introduction to Object-Oriented Database"의 내용에서 참고한 것이다.

1) 상업용 시스템

① Gem Stone

Service-Logic Development에서 개발된 GemStone은 VAX나 SUN 워크스테이션을 서버로 사용하며, IBM PC, Macintosh 등을 의뢰자로 사용한다. GemStone은 응용 프로그램과의 인터페이스로 C, C++, Smalltalk를 제공하는 한편, Smalltalk-80에 데이터 정의와 데이터 처리기능 등을 추가한 형태의 OPAL 언어를 제공하고 있다.

GemStone에서는 객체가 다른 객체에 의해 참조되고 있는 한 삭제되지 않으며, 지속성 객체의 경우 클래스 계층구조의 루트로부터 그 객체에 도달할 수 있는 경로가 존재하지 않는 경우에 한하여 자동적으로 삭제된다. 또한 클래스의 인스턴스들의 모임을 집합으로 분리하여 같은 클래스에 속한 인스턴스일지라도 사용자에게 따라 관심의 대상이 되는 인스턴스들을 모아서 그

집합에 대해 질의를 행하고, 권한 부여의 단위로도 사용한다. 그리고 GemStone은 ORION과 IRIS와 함께 동적 스키마 변경을 지원하는 시스템 중에 하나이다.

② ONTOS

ONTOS는 Ontologic사에서 만든 객체지향 데이터베이스 시스템으로 C++을 확장한 형태이다. ONTOS의 전신은 VBASE인데, VBASE에서는 형(type), 애트리뷰트(attribute), 메소드(method) 등을 정의하는 TDL(Type Definition Language)과 응용 프로그램 및 메소드의 실행부를 만들 수 있는 COP(C Object Processor)로 구성되어 있었다. 그러나 Ontologic사는 여러가지 이유로 VBASE를 포기하고 C++를 확장하는 형태를 취하게 되었다. ONTOS는 분산 서버 클라이언트 모델을 채택하였으며, 따라서 서버가 여러개 존재한다.

③ VERSANT

VERSANT는 Versant Object Technology사에서 개발한 분산 데이터베이스 시스템이다. C와 C++에 대한 인터페이스가 제공되며, 버전제어(version control) 기능이 제공된다.

특히 데이터베이스 응용 프로그램을 여러사람이 공동 개발할 경우, 이를 효과적으로 지원하기 위해서 그룹 데이터베이스와 개인 데이터베이스 사이의 체크인(check in), 체크아웃(check out) 기능을 제공하고 있다.

이 밖에도 Itasca의 Itasca, O2 Technology의 O2, Object Database의 Object Database, Object Design의 ObjectStore, Objectivity의

Objectivity, Postgres를 상품화한 Miro 등이 상품화되어서 판매되고 있는 객체지향 데이터베이스 시스템들이다.

이들 객체지향 데이터베이스 시스템들은 1987년초부터 판매되기 시작했으나 대부분은 아직 평가중이다. 그리고 꽤 많은 제품이 인위적으로 제품 설치건수를 올리려고 무상으로 보급되기도 했다. 과거 5년동안은 일반적인 객체지향 기술, 특히 객체지향 데이터베이스 기술이 태동하는 시기였으므로 기술이 성숙하지 않은 관계로 아직 중대한 응용분야에 적용되지 않고 있다.

2) 산업적 연구 프로토타입 (Industrial Research Prototype)

① ORION

ORION은 MCC에서 개발된 시스템으로 객체지향 데이터베이스시스템 연구의 선구자적 역할을 했다고 할 수 있다. ORION은 버전 (version), 동적 스키마 변경, 멀티미디어 데이터처리 등의 기능을 제공하고 있으며, 또한 복합 객체(composite object) 개념의 지원을 통해 IS-PART-OF의 의미를 갖는 데이터들을 묶어서 이를 검색, 권한부여, 동시성제어(concurrency control) 등의 단위로 사용한다.

COMMON LISP를 확장한 형태의 ORION은 3개의 시스템을 가지고 있는데, 이중 ORION-1은 단일 사용자용, ORION-SX는 서버/클라이언트 모델, 그리고 ORION-2는 분산 데이터베이스시스템이다.

② IRIS

IRIS는 Hewlett-Packard사에서 개발되고 HP68000 워크스테이션에서 사용된다.

IRIS는 C와 LISP 인터페이스를 제공하며, SQL을 확장한 OSQL도 지원한다.

현재는 관계형 저장장치(HP SQL)상에서 객체 처리기 (object manager)가 작동되고 있다. IRIS의 데이터 모델은 함수형 데이터 모델인 DAPLEX를 기반으로 하고 있으므로, 함수(function)라고 불러주는 애트리뷰트(attribute)는 단일 값(single-value) 또는 다중 값(multi-value)을 가질 수 있으며, 각 애트리뷰트에 대해 역 관계 애트리뷰트(inverse attribute)를 정의할 수 있다.

또 IRIS는 지속성 객체를 명시적으로 삭제할 수 있는데, 이때 참조 무결성(referential integrity)을 보장하기 위해서 해당 객체를 참조하는 링크(link)는 모두 제거되어야 한다. 이와 더불어 IRIS에서는 동적 스키마 변경, 버전, 질의어, 질의어 최적화 등이 제공된다.

③ O2

Altair, France에서 개발된 O2는 인터페이스로 각각 C와 Basic에 삽입한 형태의 CO2와 BasicO2를 제공하고 있다. O2의 객체 처리기(Object manager)는 WISS (Wisconsin Storage System)상에서 작동한다. 또 O2는 객체와 값을 구별하여 문자, 문자와 같은 기본 데이터 형은 물론이고 집합, 리스트, 튜플(tuple)과 같은 복합 값도 정의하여 처리할 수 있다. 또한 개발 단계와 실행 단계를 구별하여 개발 단계에서는 스키마 변경 등을 통해 유연성을 제공하며, 실행 단계에서는 효율성을 중시하여 성능을 향상시킨다.

④ Jasmin

Jasmin은 Fujitsu사에서 개발되었으며 객체 처리기(object manager)는 내포형 관계 데이터베이스 시스템상에서 구현되었다. IRIS와 마찬가지로 Jasmin은 DAPLEX 데이터 모델을 기반으로 하며 C언어를 확장한 형태이다. 또한 디몬(daemon)을 사용하여 무결성을 보장할 수 있도록 하였고, 데이터 정의언어(DDL)와 데이터 처리언어(DML) 그리고 응용 프로그램을 위한 Jasmin/C언어로 구성되어 있다.

3) 대학 연구프로토타입 (University Research Prototypes)

① ENCORE/Observer

ENCORE/Observer 시스템은 장기간 트랜잭션을 위한 동시성 제어와 동적 스키마 변경을 처음으로 제안하였다.

여기서 ENCORE는 전위 객체 처리기(front-end object manager)로서 데이터 모델을 제공하는데 다중 상속, 다중값 속성(multi-valued attribute), 버전(version) 등을 제공한다. Observer는 후위 객체 서버(back-end object server)라 할 수 있는데 멀티미디어 데이터를 저장하고 동시성 제어, 복구, 트랜잭션 관리 등을 처리한다. 또한 객체의 집단화(clustering)를 효율적으로 하기 위해서 객체를 이중으로 유지하기도 한다.

② EXTRA/EXODUS

EXTRA/EXODUS 시스템은 Wisconsin 대학에서 개발되었다. EXODUS는 확장형 데이터베이스 시스템으로서 엄밀히 말하면

데이터베이스 시스템 생성기라 할 수 있다. 데이터베이스를 어디에 사용할 것인지에 따라 정의하고자 하는 데이터 형이 다를 수 있고, 데이터에 대한 연산도 달라질 수 있으므로 응용 분야에 알맞는 데이터베이스 시스템을 필요에 따라 EXODUS에서 제공되는 E언어를 사용하여 생성하는 시스템이다. E언어는 C++에 지속성 데이터를 처리할 수 있도록 확장한 형태로서, 사용자의 필요에 따라 확장한 데이터 형과 연산을 추가로 정의하고, 이를 시스템이 정의한 형(type)이나 연산과 같은 방법으로 사용할 수 있도록 하는 확장성을 제공한다. EXTRA 데이터모델은 EXODUS를 이용하여 만든 데이터 모델로서 값과 객체를 구별하여 정의, 사용할 수 있고, 다중 상속, 복합객체(composite object) 등을 지원한다.

또 GemStone과 같이 집합에 대해서 질의를 행한다.

③ POSTGRES

POSTGRES는 관계형 데이터베이스 시스템에서 주도적인 역할을 한 INGRES의 후계자이다. INGRES에 ADT(abstract data type), 프로시저(procedure), 트리거(trigger) 등의 새로운 기능들을 첨가하는 일이 수행되었다. POSTGRES는 이런 새로운 기능들과 복합객체, 재귀적 질의어 처리, 역사가 첨가된 데이터의 관리, 버전(version)관리 등의 기능을 첨가하여 고안되었다.

POSTGRES의 철학은 잘 정의된 관계 모델을 유지하면서 기능을 확장하는 것이다. 하지만 POSTGRES 데이터모델은 기본

적인 객체지향 개념을 포함하고 있다. 다중 상속을 지원하는 “관계” 계층구조를 지원하고, “프로시저”를 관계(relation)의 속성의 데이터 타입으로 제공한다(프로시저는 메소드이다).

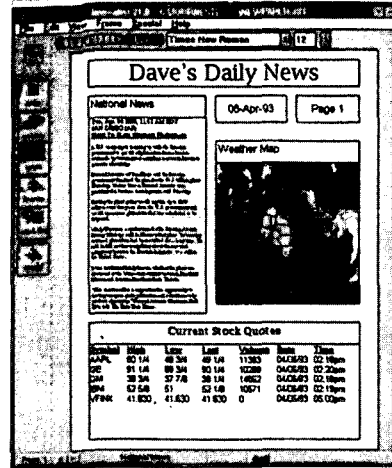
POSTGRES의 또다른 특징은 하드웨어 기술의 발전을 데이터베이스 구조에 반영했다는 것이다. WORM(write once read many) 디스크를 압축 저장 장치로 사용하고, 로그를 저장하는데 안정적인 주기억 장치를 사용한다. 이는 복구 모델을 단순하게 만든다.

향후 발전전망

객체지향 데이터베이스는 프로그램 언어에 지속성을 지원하는 저장 시스템을 첨가하는 형태로 개발되기 시작했다. 그래서 관계형 데이터베이스가 가지는 복잡한 질의 기능을 제공하지 못한다. 또한 많은 시스템들이 일반 데이터베이스시스템 사용자들이 많이 사용하는 기능이 결여되어 있다. 비절차적 질의어, 자동 질의어 처리와 최적화, 동시성 제어, 권한검사 동적 스키마 변경, 시스템 데이터 구조의 변경 등의 기능이 제대로 제공되지 못하고 있다.

그러므로 객체지향 데이터베이스는 현존하는 관계형 데이터베이스의 모든 기능을 지원할 수 있도록 수정되어야 한다.

그리고 현존하는 객체지향 데이터베이스 시스템의 많은 수가 관계형 데이터베이스 위에 객체 관리기(Object Manager)라는 층을 두고서 저장 시스템이나 질의어 처리, 최적화 등을 관계형 데이터베이스에 의존하는 형태를 취하고 있다. 이런 방법은 개



〈해외 전자신문 검색 예〉

발이 용이하고 여러 시스템 위에서 구성될 수 있는 장점이 있지만 객체지향 데이터 모델의 특성을 제대로 살리지 못하므로 불리하고 비효율성을 야기시킨다. 그러므로 객체지향 데이터베이스를 저장 시스템부터 새로이 설계하고 질의어 처리기, 동시성 제어, 권한 검사 등을 처리하는 부분도 새로이 설계되어야 한다.

그러나 현재 중대한 응용분야에 사용중인 데이터베이스시스템 정도의 기능을 갖추도록 하는 것은 향후 3~4년 내에는 힘들 것으로 전망된다.

하지만 객체지향 데이터베이스는 유연성을 그 특징으로 하므로 중대한 응용분야가 아닌 디자인 환경에서는 그 효율성을 계속 늘려나갈 것으로 생각된다. CAD, CASE 분야에서의 활용이 기대되고, OIS(Office Information System)에서의 활용도 기대된다. 이런 응용분야에서 사용되어지면서 객체지향 데이터베이스의 기술이 성숙되어나갈 것이다.

다음 표는 객체지향 시스템의 시장규모

<표 1>

	1988	1989	1990	1991	1992	1993	1994	1995
Smalltalk	4	10	19	39	83	146	227	327
C++	21	40	64	111	184	285	398	547
AI	101	138	185	241	307	385	474	574
Hybrid Toolkit	45	61	82	108	139	176	218	267
databases	3	3	7	10	22	48	96	152
CASE tools	0	1	3	8	16	28	46	70
End-user Tools	2	3	5	8	16	32	64	128

* Smalltalk와 C++와 AI에 대한것은 미국과 유럽시장에 대한 수치이며, 나머지는 미국시장에 대한 수치임.

에 대한 Ovum사의 예측이다. 객체지향 데이터베이스 시장은 앞으로 비약적인 발전을 하리라고 기대된다.

국내 기술수준 및 결론

국내에서는 아직도 데이터베이스가 도입기를 벗어나지 못하고 있다. 아직도 화일 시스템만으로 꾸려나가는 곳이 많고, 데이터베이스를 사용한다고 해도 복잡한 기능은 사용되지 않고 단순 조회가 사용패턴의 대부분을 이룬다. 이제 겨우 관계형 데이터베이스 시장이 형성된 정도이다. 그런 반면에 인사관리 등의 분야에 쓰기 위해서 멀티미디어를 지원하는 데이터베이스의 요구는 상당히 큰 상태라고 할 수 있다.

국내 기술에 의한 관계형 데이터베이스 관리 시스템도 개발에 성공한 것은 두가지 정도이다.

한국전자통신연구소에서 “바다”라는 시제품 개발에 성공했고 이를 대우통신에서 상품화해서 “한바다”라는 상품이 나와있

다.

그리고 삼성에서 “CODA”라는 관계형 데이터베이스 시제품 개발에 성공했고, 곧 이를 상품화할 계획이라고 알고 있다.

하지만 이 제품들도 성능면에서 외국제품들(informix, Oracle, Sybase, Ingres 등)에 비해 좋은 성능을 내지는 못하는 것으로 알려지고 있다.

그리고 서울대학교 컴퓨터공학과 객체지향시스템연구실에서 관계형 데이터베이스 관리 시스템과 객체지향 데이터베이스 관리 시스템을 개발 중에 있다.

아직은 국내 데이터베이스 시장은 멀티미디어 기능을 보강한 관계형 데이터베이스 시스템이 주도적일 것 같고 객체지향 데이터베이스시스템이 제대로 쓰이기까지는 시간이 걸릴 것으로 생각된다.

하지만 관계형 데이터베이스가 외국 제품에 의해 시장이 점령된 전례를 보더라도 지금부터 미리미리 객체지향 데이터베이스 기술을 확보하고 시스템을 구축하는 노력을 기울여야 할 것으로 생각된다. [6]