

# 오디세우스 객체관계형 DBMS를 위한 오브젝트 뷰

(Object Views in the ODYSSEUS Object-Relational DBMS)

이재길<sup>\*</sup> 한욱신<sup>\*\*</sup> 이민재<sup>\*</sup> 이종학<sup>\*\*\*</sup> 황규영<sup>\*\*\*\*</sup>  
(Jae-Gil Lee) (Wook-Shin Han) (Min-Jae Lee) (Jong-Hak Lee) (Kyu-Young Whang)

**요약** 뷰는 데이터베이스 관리 시스템에서 논리적 데이터 독립성을 지원하기 위한 필수적인 요소이다. 객체지향/객체관계형 데이터베이스의 오브젝트 뷰는 객체지향 개념을 지원해야 하므로 관계형 데이터베이스의 뷰와는 다른 요구 기능을 가진다. 현재 대부분의 상용 객체지향/객체관계형 데이터베이스 관리 시스템에서 오브젝트 뷰의 기능을 일부 지원하고 있지만, 구현 방법에 대해서는 충분히 알려져 있지 않다. 본 논문에서는 객체지향/객체관계형 데이터베이스를 위한 오브젝트 뷰의 구현 방법을 제안하고, 이를 오디세우스 객체관계형 데이터베이스 관리 시스템에 구현한다. 이를 위해 먼저 오브젝트 뷰의 요구 사항을 분석한다. 다음으로, 오브젝트 뷰의 구현을 위해 관계형 데이터베이스에서 뷰의 구현을 위해 사용하는 질의 수정 알고리즘을 확장하는 방법을 제안한다. 다음으로, 제안한 뷰의 기능들을 상용 객체관계형 데이터베이스 관리 시스템의 뷰의 기능들과 비교한다. 비교 결과, 제안한 방법이 기존의 오브젝트 뷰에 비해 객체지향 개념인 객체 식별자, 상속, 메소드, 복합 객체들을 모두 잘 지원함을 보인다. 마지막으로, 확장된 질의 수정 방법을 오디세우스 객체관계형 데이터베이스 관리 시스템에 구현하기 위한 세부 방법들을 제시한다.

**키워드** : 객체지향/객체관계형 데이터베이스, 뷰

**Abstract** Views are essential in providing logical data independence for database systems. Object views in object-oriented/object-relational databases have requirements quite different from those of relational databases due to support for object-oriented concepts. Although many commercial object-oriented/object-relational database systems support object views, implementation techniques have not been discussed sufficiently in the literature. In this paper, we devise a technique for implementing views in object-oriented/object-relational databases and apply it to the ODYSSEUS object-relational database system. We first analyze the requirements of object views. Next, to implement object views, we extend the existing query modification algorithm that has been proposed for implementing views in relational databases. Next, we compare the features of the proposed object view with those of object views in commercial object-relational database systems. It is shown that the proposed object view supports all object-oriented concepts such as object identifiers, inheritance, methods, and composite objects, while existing object views support part of them. Last, we propose detailed techniques for implementing the extended query modification algorithm in the ODYSSEUS object-relational database system.

**Key words** : object-oriented/object-relational database, view

• 본 연구는 첨단정보기술연구센터를 통하여 한국과학재단으로부터 지원  
을 받았음

\* 학생회원 : 한국과학기술원 전자전산학과/첨단정보기술연구센터

jglee@mozart.kaist.ac.kr

mjlee@mozart.kaist.ac.kr

\*\* 종신회원 : 경북대학교 컴퓨터공학과 교수

wshan@knu.ac.kr

\*\*\* 종신회원 : 대구가톨릭대학교 컴퓨터정보통신공학부

jhlee11@cuth.cataegu.ac.kr

\*\*\*\* 종신회원 : 한국과학기술원 전자전산학과 교수

첨단정보기술연구센터 소장

kywhang@mozart.kaist.ac.kr

논문접수 : 2002년 12월 11일

심사완료 : 2003년 8월 12일

## 1. 서론

객체관계형 데이터베이스 관리 시스템(Object-Relational Database Management System: ORDBMS)은 관계형 데이터베이스 관리 시스템(RDBMS)을 확장하여 타입(type), 객체(object), 참조(reference), 메소드(method), 상속(inheritance) 등의 객체지향 모델링 기능을 추가한 시스템이다[1]. ORDBMS는 RDBMS에 비해 풍부한 시멘틱스를 제공하므로 XML, CAD/CAM과 같은

복잡한 응용을 지원하는데 적합한 것으로 알려져 있다. 이미 대부분의 상용 DBMS 개발사들은 자신의 RDBMS를 ORDBMS로 확장하였으며, 현재에도 지속적으로 ORDBMS의 기능을 추가하고 있다. 이러한 작업은 주로 기본 테이블(base table)에 객체, 참조, 상속과 같은 객체지향 개념을 추가하는데 집중되었다. 그러나 기본 테이블 뿐만 아니라 뷰에도 객체지향 개념을 적용하는 것이 필요하다[2]. 왜냐하면 뷰(view)는 논리적 데이터 독립성(logical data independance)을 지원하기 위한 필수적인 요소이기 때문이다.

최근 상용 DBMS 개발사들은 이러한 필요성을 인지하여 객체지향 개념을 지원하는 오브젝트 뷰(object view)를 제공하기 시작하였다. ORACLE9i[3]와 IBM DB2[4]에서는 기존의 RDBMS에서 제공되던 뷰를 유지하는 한편 객체지향 개념이 추가된 오브젝트 뷰를 제공한다. 그러나 오브젝트 뷰의 구현 방법에 대해서는 많은 연구가 수행되지 않았고 상용 DBMS 개발사들도 구현 방법을 자세히 공개하지 않았기 때문에, 오브젝트 뷰를 구현하는 데 어려움이 있다.

OODBMS/ORDBMS의 오브젝트 뷰에는 객체 식별자(object identifiers), 상속(inheritance), 메소드(methods), 복합 객체(composite objects)와 같은 객체지향 개념들이 반영되어야 한다[5]. OODBMS/ORDBMS에서 뷰를 제공하고자 하는 목적은 뷔를 가상 클래스(virtual class) 혹은 가상 테이블(vitural table)처럼 사용할 수 있도록 제공하는 것이다[6]. 따라서 뷔를 클래스와 동일한 방법으로 사용할 수 있도록 제공하기 위해서는 클래스에 적용되는 객체지향 개념들을 뷔에도 적용시켜야 한다.

본 논문은 OODBMS/ORDBMS의 오브젝트 뷔를 위한 새로운 구현 방법을 제안하고, 제안한 방법을 이용하여 오브젝트 뷔를 구현하는 것을 목적으로 한다. 제2절에서는 관련 연구로서 기존 RDBMS의 뷔와 그 구현 방법에 대해서 살펴본다. 제3절에서는 오브젝트 뷔를 위한 새로운 구현 방법으로 RDBMS의 질의 수정 방법을 확장하는 방법에 대해서 설명한다. 이를 위해 먼저 오브젝트 뷔의 요구 사항을 분석한다. 다음으로, 오브젝트 뷔의 구현을 위해 RDBMS에서 뷔의 구현을 위해 사용하는 질의 수정 알고리즘을 확장하는 방법을 제안한다. 다음으로, 제안한 방법에서 지원하는 뷔의 기능들을 상용 객체관계형 데이터베이스 관리 시스템의 뷔의 기능들과 비교한다. 제4절에서는 확장된 질의의 수정 방법을 한국과학기술원에서 개발하고 있는 오디세우스[7] 객체관계형 데이터베이스 관리 시스템에 구현하기 위한 세

부 방법들을 제시한다. 마지막으로 제5절에서는 결론을 내린다.

## 2. 관련 연구

본 장에서는 기존 RDBMS의 뷔에 대해서 소개하고 그 구현 방법에 대해서 다룬다. 제2.1절에서는 RDBMS에서의 뷔의 개념에 대해서 설명한다. 제2.2절에서는 RDBMS에서 뷔를 구현하기 위해서 사용하는 대표적인 두 가지 구현 방법인 질의 수정과 뷔의 구체화에 대해서 설명하고 각 방법을 비교한다.

### 2.1 RDBMS의 뷔

RDBMS에서 뷔는 하나 혹은 그 이상의 기본 테이블(base table)로부터 유도된(derived) 논리적인 테이블로 정의된다[8]. 기본 테이블이란 테이블의 투플(tuple)들이 디스크와 같은 물리적인 저장 장소에 존재하는 테이블이다. 뷔는 기본 테이블과 마찬가지로 투플들의 집합으로 구성되지만, 뷔의 투플들은 기본 테이블의 투플들로부터 유도된 것으로 물리적인 저장 장소에 존재하지 않을 수 있다.

그림 1은 consumer 기본 테이블을 사용해 big\_consumer 뷔를 정의한 예로, big\_consumer 뷔의 투플을 유도하기 위해 SQL[9]의 SELECT 질의식을 사용한다. 그림에서 사용된 SQL 질의식은 consumer 테이블에서 quantity 속성의 값이 10 이상인 투플의 name, quantity, age 속성 값을 big\_consumer 뷔의 투플로 정의한다. 그림 1에서 consumer 테이블에 색칠되어 있는 부분은 뷔를 정의하는 질의식을 만족하는 값들로 big\_consumer 뷔의 투플을 구성한다.

CREATE VIEW big_consumer (vname, vquantity, vage) AS SELECT name, quantity, age FROM consumer WHERE quantity >= 10			
유 정 의 질 의식			
consumer 테이블			big_consumer 뷔
name	quantity	age	vname
Lee	12	25	Lee
Song	9	29	Kim
Kim	14	24	Park
Yoo	5	30	
Park	11	27	

그림 1 SQL에서의 뷔 정의 예

뷰는 질의식에서 기본 테이블과 동일한 방식으로 질의 대상이 될 수 있다. 기본 테이블에 대해 SELECT, UPDATE, INSERT, DELETE 질의를 수행하면 기본 테이블에 대한 데이터의 검색, 생성, 추가, 삭제 연산이 수행된다. 뷔에 대해서도 기본 테이블과 마찬가지로 SELECT, UPDATE, INSERT, DELETE 질의가 가능

하다. 뷰에 대한 SELECT 질의는 기본 테이블로부터 유도된 데이터를 검색함으로써 수행된다. 뷔에 대한 UPDATE 질의는 뷔에 생성된 데이터가 유도되도록 기본 테이블을 생성함으로써 수행된다. 뷔에 대한 INSERT 질의는 뷔에 추가된 데이터가 유도되도록 기본 테이블에 데이터를 추가함으로써 수행된다. 뷔에 대한 DELETE 질의는 뷔에서 삭제된 데이터가 유도되지 않도록 기본 테이블에서 데이터를 삭제함으로써 수행된다.

## 2.2 RDBMS의 뷔 구현 방법

RDBMS의 뷔 구현을 위해 사용되는 방법으로는 뷔의 투플들이 물리적 저장 장소에 존재하는지 여부에 따라 1) 질의 수정(query modification)[10]과 2) 뷔의 구체화(view materialization)[11,12] 방법이 있다. 질의 수정 방법은 뷔의 투플들을 뷔의 정의로부터 유도해내는 방법이고, 뷔의 구체화는 별도의 물리적 저장 장소에 뷔의 투플들을 저장하는 방법이다. 본 절에서는 각 방법에 대해서 설명하고, 각 방법의 장단점에 대해서 비교한다.

### 2.2.1 질의 수정

질의 수정[10] 방법에서는 뷔에 대한 질의를 기본 테이블에 대한 질의로 수정하여 질의를 수행한다. 따라서 질의 수정 방법에서 뷔의 투플들은 별도의 물리적 저장 장소에 생성되지 않고, 가상적으로 존재한다.

알고리즘 1은 참고 문헌 [10]에서 제안한 질의 수정 알고리즘이다. 알고리즘 1은 질의 수정을 위해 1) 질의식에 사용된 뷔는 그 뷔를 유도한 기본 테이블로 대치하고, 2) 질의식에 사용된 뷔의 속성은 그 속성을 유도한 기본 테이블의 속성으로 대치하며, 3) 뷔를 유도하는 실례선 조건을 질의식의 조건에 추가하는 과정을 수행한다. 알고리즘 1은 QUEL을 기준으로 작성되어 있으며, QUEL의 뷔 정의 DDL 문법은 아래와 같다.

**RANGE OF  $X_1$  IS  $R_1$**

**RANGE OF  $X_2$  IS  $R_2$**

**RANGE OF  $X_n$  IS  $R_n$**

```
RANGE OF e IS emp
DEFINE research(vname=e.name, vsalary=e.salary)
WHERE e.dept = "research"
```

(a) research 뷔를 정의하는 질의식

```
RANGE OF e IS emp
RETRIEVE INTO lee(e.salary)
WHERE e.name = "Lee" AND e.dept = "research"
```

(c) 질의 수정을 거친 질의식

```
DEFINE VIEWNAME( $D_1 = f_1, \dots, D_j = f_j$ )
WHERE  $Q_v$ 
```

알고리즘 1 RDBMS의 뷔를 위한 질의 수정 알고리즘[10]: 질의식에 사용된 각각의 뷔의 투플 변수 V에 대하여 뷔 정의의 WHERE 절에 사용된 조건이  $Q_v$ 일 때, 아래의 과정을 수행한다.

1. RANGE 절에서 V를 삭제하고 뷔 정의에 있는 모든 투플 변수를 추가한다.
2. DELETE, APPEND, REPLACE 절의식인 경우 V 가 두개 이상의 기본 테이블로 정의되어 있으면 수행을 중단한다. 그렇지 않으면 V를 V가 정의된 기본 테이블의 이름으로 대치한다.
3.  $Q_v$ 에 나타난 테이블을 REPLACE 하려는 경우 수행을 중단한다.
4. V가 사용된 질의식의 WHERE 절에  $Q_v$ 를 AND로 연결한다.
5. V가 사용된 질의식에서 V가 집단화 함수 내에 사용된 경우 집단화 함수의 WHERE 절에  $Q_v$ 를 AND로 연결한다.
6. V가 사용된 질의식에서  $V.D_i$ 를 뷔 정의를 참조해  $f_j$ 로 대치한다.

□

그림 2는 알고리즘 1을 적용하여 뷔에 대한 질의식을 기본 테이블에 대한 질의식으로 수정한 예이다. 그림 2(a)는 뷔의 정의로서, 고용자 중 부서가 "research"인 고용자의 이름과 월급을 research 뷔로 정의한다. 그림 2(b)는 뷔에 대한 질의식으로서, research 뷔에서 이름이 "Lee"인 고용자의 월급을 구하는 질의식이다. 그림 2(c)는 research 뷔에 대한 질의식을 emp 기본 테이블에 질의식으로 수정한 결과이다. 즉, 그림 2(b)의 RANGE 절에 있는 research 뷔는 알고리즘 1의 단계 1에 의해 그림 2(c)의 emp 테이블로 대치된다. 그림 2(a)의 WHERE 절은 알고리즘 1의 단계 4에 의해 그림 2(c)의 WHERE 절에 AND로 연결되며, 그림 2(b)

```
RANGE OF r IS research
RETRIEVE INTO lee(r.vsalary)
WHERE r.vname = "Lee"
```

(b) research 뷔에 대한 질의식

그림 2 질의 수정 알고리즘 수행 예

의 RETRIEVE, WHERE 절의 vsalary, vname 속성은 알고리즘 1의 단계 6에 의해 그림 2(c)의 salary, name 속성으로 대치된다.

### 2.2.2 뷰의 구체화

뷰의 구체화[11,12] 방법에서는 뷰의 투플들을 물리적인 저장 장소에 존재하는 테이블에 구체화하고, 뷔에 대한 질의를 구체화된 테이블에 대한 질의로 수행한다. 뷔의 투플은 뷔를 정의하는 질의식을 수행하여 그 결과를 테이블에 저장함으로써 구체화된다.

구체화된 테이블과 기본 테이블의 데이터들 사이에는 일관성이 유지되어야 한다. 즉, 구체화된 테이블의 데이터는 기본 테이블의 데이터로부터 유도된 것이기 때문에 기본 테이블의 데이터가 변경되면 구체화된 뷔의 데이터도 변경시켜 주어야 하고, 마찬가지로 구체화된 뷔의 데이터를 변경시킬 경우에는 그 뷔를 정의하는 데 사용된 기본 테이블의 데이터도 변경시켜야 한다.

### 2.2.3 구현 방법 비교

뷰의 두 가지 구현 방법인 질의 수정과 뷔의 구체화는 뷔의 종류와 뷔가 사용되는 환경에 따라 장단점이 있다[13]. 뷔의 구체화 방법은 뷔의 투플들이 이미 구체화된 테이블에 저장되어 있기 때문에 검색 연산이 효율적이라는 장점이 있는 반면, 질의 수정 방법은 뷔의 투플들을 얻어내기 위해 매번 질의 수정을 수행해야 하므로 검색 연산이 상대적으로 비효율적이라는 단점이 있다. 그러나, 질의 수정 방법은 기본 테이블의 변경시 별도의 처리가 필요 없다는 장점이 있는 반면, 뷔의 구체화 방법은 기본 테이블의 데이터가 변경되면 그에 따라 구체화된 테이블의 데이터도 변경해 주어야 한다는 단점이 있다.

참고 문헌 [13]에서는 두 가지 방법으로 뷔를 구현한 경우에 뷔의 사용 환경에 따른 수행 성능을 실험하였다. 참고 문헌 [13]의 실험 결과에 의하면 선택(selection)과 프로젝션(projection)만을 뷔에 포함하거나, 갱신 연산의 비율이 비교적 높은 경우는 질의 수정 방법이 더 좋은 수행 성능을 보인다. 반면, 뷔가 조인이나 집단화와 같은 복잡한 연산을 포함하고 갱신 연산의 비율이 비교적 낮은 경우는 뷔의 구체화 방법이 더 좋은 수행 성능을 보인다.

현재 대부분의 상용 DBMS에서는 뷔의 구현 방법으로 질의 수정 방법을 채택하고 있다. ORACLE, DB2와 같은 주요 상용 DBMS에서 뷔 구현을 위해 질의 수정 방법을 사용한다[3,14]. 단, ORACLE9i는 데이터 웨어하우징과 같은 응용을 위해 구체화된 뷔까지도 지원하고 있다[3]. 따라서 본 논문에서는 대부분의 응용 분야에서 사용되는 구현 방법인 질의 수정 방법을 오브젝트 뷔의 구현에 적용시킬 수 있는 방법에 대해서 설명한다.

## 3. 오브젝트 뷔를 위한 새로운 구현 방법

본 절에서는 오브젝트 뷔를 위한 새로운 구현 방법을 제안한다. 기존의 RDBMS의 뷔 구현 방법은 오브젝트 뷔에 추가되는 객체지향 개념이 고려되어 있지 않기 때문에 수정 없이 오브젝트 뷔의 구현에 사용될 수 없다. 그리고 대부분의 상용 OODBMS/ORDBMS 개발사에서는 오브젝트 뷔의 구현 방법에 대해서 충분히 공개하지 않고 있다. 따라서 오브젝트 뷔를 위한 새로운 구현 방법에 대한 연구가 필요하다. 제3.1절에서는 오브젝트 뷔의 개념과 요구 사항에 대해서 기술한다. 제3.2절에서는 오브젝트 뷔를 위한 새로운 구현 방법으로 제2.2.1절에서 소개한 질의 수정 방법을 확장하는 방법에 대해서 기술한다. 제3.3절에서는 상용 ORDBMS에서 제공하는 뷔와 본 논문에서 제안한 구현 방법의 기능을 비교한다.

### 3.1 오브젝트 뷔의 요구 사항

ODBMS/ORDBMS에서 오브젝트 뷔는 하나 혹은 그 이상의 저장 클래스(stored class)로부터 유도된 논리적인 클래스로 정의된다[5]. OODBMS/ORDBMS의 오브젝트 뷔는 RDBMS의 뷔에서 지원되는 기능을 모두 지원해야 하고 객체지향 개념도 반영해야 한다[5]. 오브젝트 뷔에 추가되어야 하는 객체지향 개념은 다음과 같다[6].

첫째, 뷔 객체에 유일한 객체 식별자[15]가 부여된다. 따라서 뷔에 대해서도 객체 식별자를 사용하는 연산을 수행할 수 있다.

둘째, 뷔에 상속의 개념이 추가된다. OODBMS/ORDBMS에서 클래스들은 상속으로 인해 상속 계층 구조를 형성한다. 뷔도 정의에 따라 일종의 클래스이므로, 데이터베이스 내의 다른 클래스 혹은 뷔와 상속 계층 구조를 형성할 수 있다.

셋째, 클래스의 정의에 메소드가 포함될 수 있으므로 뷔 정의에서도 메소드가 포함될 수 있다.

넷째, 복합 객체에 뷔 객체가 포함될 수 있다. OODBMS/ORDBMS에서 클래스 속성의 도메인으로 임의의 클래스를 지정할 수 있으므로, 그림 3과 같이 클래스-복합 계층 구조(class-composite hierarchy)를 구성할 수 있다. 복합 객체는 클래스-복합 계층 구조를 구성하는 클래스들의 객체들의 집합으로 정의된다[15]. 뷔도 정의에 따라 일종의 클래스이므로, 뷔 객체도 클래스 객체와 마찬가지로 복합 객체에 포함될 수 있다. 즉, 클래스 객체가 뷔 객체를 참조할 수 있고 뷔 객체가 클래스 객체를 참조할 수 있다. 그림 3에서 product 클래스, big\_consumer 뷔, address 클래스 객체들의 집합은 복합 객체를 구성한다. product 클래스의 consumer 속

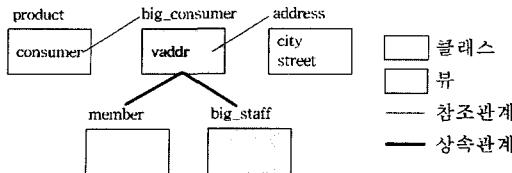


그림 3 클래스 계층 구조와 클래스-복합 계층 구조의 예

성은 big\_consumer 뷰의 객체 식별자를 가지며 big\_consumer 뷰의 vaddr 속성은 address 클래스의 객체 식별자를 가진다.

### 3.2 오브젝트 뷰 구현을 위한 질의 수정 방법의 확장

본 절에서는 뷰를 사용하는 질의식에 객체 식별자, 상속, 메소드, 복합 객체가 포함되어 있는 경우 어떻게 질의 수정을 해야하는지 설명하고, 이를 위해 기존의 RDBMS의 질의 수정 방법을 확장하는 방법을 제안한다.

#### 3.2.1 객체 식별자를 위한 확장

OODBMS/ORDBMS의 질의식에는 객체 식별자가 사용될 수 있으므로 질의식에 사용된 객체 식별자를 질의 수정 할 수 있도록 RDBMS의 질의 수정 방법을 확장해야 한다. 객체 식별자를 사용하는 질의는 크게 질의 결과로 객체 식별자를 반환하는 형태와 객체 식별자로 객체의 컬럼 값을 액세스하는 형태로 나누어진다. 각각의 경우에 대한 예는 그림 4와 같다. 그림 4(a)는 big\_consumer 뷰에서 vname 속성의 값이 "Lee"인 객체의 객체 식별자를 반환하는 질의식이며, 그림 4(b)는 객체 식별자 "VOID1"이 가리키는 객체의 vname, vage 컬럼의 값을 반환하는 질의식이다.

```
SELECT big_consumer
FROM big_consumer
WHERE vname = "Lee"
```

(a) 객체 식별자 반환

```
SELECT vname, vage
FROM OBJECT "VOID1"
```

(b) 컬럼 값 액세스

그림 4 질의식에서의 객체 식별자 사용 예

위에서 설명한 두 가지 질의식을 처리하기 위해서는 다음과 같은 사항을 고려해야 한다. 첫째, 뷰 객체 식별자가 유일하게 부여되어야 한다. 뷰 객체 식별자로 대응

되는 기본 테이블의 객체 식별자를 부여하면 데이터베이스에 동일한 객체 식별자를 가지는 여러 개의 객체가 존재하게 되므로, 객체 식별자를 사용한 질의를 수행할 수 없게된다. 따라서, 뷰 객체들에게도 유일한 식별자가 부여되어야 한다. 둘째, 뷰 객체 식별자를 사용하여 뷰 객체의 컬럼 값을 액세스 할 수 있어야 한다.

뷰 객체의 객체 식별자는 앞에서 설명한 두 가지 요건을 만족시키기 위해 뷰 객체를 유도한 클래스 객체의 객체 식별자와 뷰의 클래스 식별자를 함께 사용해서 부여된다. 이와 같은 방식으로 객체 식별자를 부여하면 객체 식별자로 뷰 객체의 컬럼 값을 액세스 할 때 실제로는 뷰 객체를 유도한 클래스 객체의 컬럼 값을 액세스하게 되며, 객체 식별자에 포함된 뷰의 클래스 식별자로 뷰 객체가 어느 뷰에 속하는지 판별할 수 있다. 뷰 객체의 객체 식별자 부여 방법은 그림 5와 같다. 오른쪽의 색칠된 뷰 객체가 왼쪽의 색칠된 클래스 객체로부터 유도되었을 경우 색칠된 뷰 객체의 객체 식별자는 색칠된 클래스 객체의 객체 식별자에 오른쪽 뷰의 클래스 식별자를 추가해서 부여된다. 뷰 객체의 객체 식별자는 객체 식별자를 반환하는 질의식을 처리하는 시점에서 부여된다.

뷰 객체의 객체 식별자는 하나의 클래스로부터만 유도된 뷰에 한해서 부여된다. 뷰 객체의 객체 식별자는 뷰 객체를 유도한 클래스 객체의 객체 식별자를 기반으로 해서 부여되므로 뷰 객체와 클래스 객체 사이에 일대일 대응이 성립해야만 한다. 두 개 이상의 클래스로부터 뷰가 유도되면, 뷰 객체와 클래스 객체 사이에 일대일 대응이 성립하지 않으므로 어느 클래스 객체를 사용하여 뷰 객체의 객체 식별자를 부여할지 결정할 수 있게 된다.

#### 3.2.2 상속을 위한 확장

OODBMS/ORDBMS의 질의식은 상속 계층 구조에 대해 질의를 수행할 수 있으므로 뷰가 포함되어 있는 상속 계층 구조에 대해서도 질의를 수행하도록 RDBMS의 질의 수정 방법을 확장해야 한다. 상속 계층 구조에 대해 질의를 수행하기 위해서는 질의 대상의 클래스/뷰를 포함하여 서브클래스/서브뷰도 질의 대상으로 처리해야 한다. 이러한 형태에 대한 질의식의 예는 그림

consumer 클래스			
name	quantity	age	goods
Lee	12	25	dictionary
Song	9	29	magazine
Kim	14	24	novel
Yoo	5	30	manual
Park	11	27	novel

big_consumer 뷰			
vname	vquantity	vage	VOID3
Lee	12	25	VOID3 = OID3 + class id of big_consumer
Kim	14	24	
Park	11	27	

$$\text{VOID3} = \text{OID3} + \text{class id of big_consumer}$$

그림 5 뷰 객체의 객체 식별자 부여 방법

6과 같다. 그림 6(a)에 있는 질의식의 FROM 절에서 질의 대상 클래스/뷰 다음에 있는 '\*'는 질의 대상 클래스/뷰의 서브클래스/서브뷰에 대해서도 질의를 수행하라는 의미이다. 즉, 그림 6(b)와 같이 big\_consumer 뷰의 서브클래스로 member 클래스, 서브뷰로 big\_staff 뷰가 있다고 가정하면, 그림 6(a)에 있는 질의식은 big\_consumer 뷰, member 클래스, big\_staff 뷰를 대상으로 수행되어야 한다.

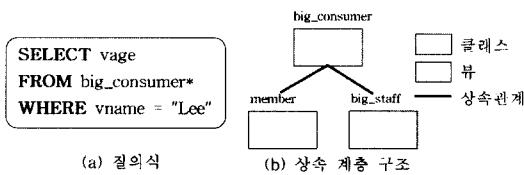


그림 6 질의식에서의 상속 사용 예

뷰가 포함되어 있는 상속 계층 구조에 대해서 질의를 수행하기 위해서는 상속 계층 구조에 나타난 뷰를 뷰 유도에 사용된 클래스로 수정해야 한다. 예를 들어, 그림 6(a)에 있는 질의식에 사용된 big\_consumer 뷔가 consumer 클래스를 사용하여 유도되었다고 가정하면, 그림 6(a)에 있는 질의식은 consumer 클래스에 대한 질의식으로 수정되고 그림 6(b)에 있는 상속 계층 구조도 big\_consumer 뷔를 consumer 클래스로 대치하여 consumer 클래스에 대한 상속 관계를 나타내도록 해야 한다.

뷰가 포함된 상속 계층 구조에서 뷔를 클래스로 대치하면 뷔 유도에 사용되지 않은 클래스 객체까지 질의식의 수행 결과로 나올 수 있다. 왜냐하면, 상속 계층 구조에서 대치된 뷔는 질의식에 나타나지 않을 수 있으므로 질의식에 뷔를 유도하는 조건을 추가할 수 없기 때문이다. 예를 들어, 그림 6(b)의 상속 계층 구조에서 big\_staff 뷔가 staff 클래스로부터 유도되었다고 가정할 때, big\_staff 뷔를 단순히 staff 클래스로 대치하면 그림 6(a)의 질의식의 결과로 big\_staff 뷔에 유도되지 않은 staff 클래스의 객체까지 나오게 된다. 이로 인해 수정 전과 수정 후의 상속 계층 구조에 대한 질의 수행 결과가 동일하지 않게 나온다. 따라서 뷔 유도에 사용되지 않은 클래스 객체는 질의 수행 결과에 나올 수 없도록 질의를 수행해야 한다.

뷰가 포함된 상속 계층 구조는 위와 같은 문제점을 고려하여 다음과 같이 질의 수정된다. 상속 계층 구조에 포함된 뷔를 그 뷔를 유도한 클래스로 대치하고 대치된 뷔의 정의에 있는 WHERE 절을 해당 클래스의 질의 그래프(query graph)의 노드에 저장한다. 대치된 뷔를 유도하는 조건을 질의식에 명시하지 못할 수 있으므로,

질의식의 질의 대상 클래스/뷰를 표현하는 내부 자료 구조인 질의 그래프에 대치된 뷔를 유도하는 조건을 명시한다. 상속 계층 구조를 따라 질의를 수행하면서 질의 그래프의 노드에 조건이 명시되어 있으면, 질의식의 WHERE 절에 명시된 조건과 질의 그래프에 명시된 조건을 AND로 연결하여 질의식을 수행한다. 예를 들어, 그림 6(b)의 상속 계층 구조에서 big\_consumer, big\_staff 뷔가 각각 consumer, staff 클래스로부터 유도되었다고 가정할 때, 그림 7(b)와 같이 consumer, staff 클래스로 대치된다. big\_staff 뷔가 대치된 staff 클래스에 대응되는 질의 그래프의 노드에는 big\_staff 뷔의 정의에 있는 WHERE 절인  $salary \geq 1000$  이 저장된다. 그림 7(a)의 질의식은 staff 클래스에 대해서 내부적으로  $salary \geq 1000$  조건이 추가되어 수행된다.

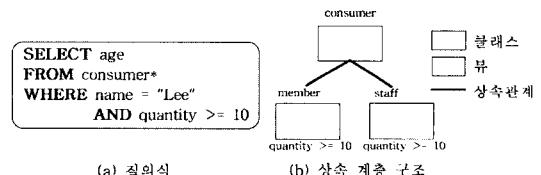


그림 7 상속 계층 구조의 수정 예

### 3.2.3 메소드를 위한 확장

OODBMS/ORDBMS의 질의식은 메소드를 호출할 수 있으므로 질의식에 사용된 뷔 객체의 메소드 호출에 대해 질의 수정을 할 수 있도록 RDBMS의 질의 수정 방법을 확장해야 한다. 그림 8은 메소드가 사용된 질의식의 예이다. 그림 8의 질의식은 big\_consumer 뷔에서 보너스 점수가 1,000이상인 객체의 vname 컬럼 값을 반환하는 질의식이다. 보너스 점수를 계산하기 위해 big\_consumer 뷔 객체의 vquantity, vage 컬럼 값을 인수로 해서 bonus\_score라는 메소드를 호출한다.

```
SELECT vname
FROM big_consumer
WHERE bonus_score(vquantity, vage) >= 1000
```

그림 8 질의식에서의 메소드 사용 예

뷰 객체의 메소드는 뷔 객체가 실제 존재하지 않는다는 점에서 클래스의 메소드와 다르게 처리된다. 일반적으로 메소드는 객체와 인수를 입력으로 받아 객체에 대해 지정된 작업을 수행한다. 뷔 객체의 메소드가 사용된 질의식에서 메소드의 수행 대상인 뷔 객체가 실제 존재하지 않으므로 뷔 객체의 메소드는 뷔 객체를 유도한 클래스 객체를 사용하여 수행되도록 수정되어야 한다. 그리고, 뷔 객체의 메소드를 구현한 메소드 본체(body)

에서는 뷔 객체의 컬럼을 액세스하므로, 메소드 본체 내의 뷔 객체의 컬럼은 클래스 객체의 컬럼으로 수정되어야 한다.

뷰 객체에 대한 메소드 호출은 위와 같은 문제점을 고려하여 다음과 같이 질의 수정된다. 일반적으로 메소드를 호출하는 과정에서 메소드가 수행될 객체에 대한 객체 식별자를 메소드 본체에 전달한다. 따라서, 뷰 객체의 메소드인 경우에는 제3.2.1절에서 설명한 방법으로 뷰 객체의 객체 식별자를 뷰 객체를 유도한 클래스 객체의 객체 식별자로 변환한 후 메소드 본체에 전달한다. 즉, 뷰 객체의 메소드는 뷰 객체를 유도한 클래스를 대상으로 수행된다. 그리고, 뷰 객체의 메소드를 실행하는 도중 메소드 본체에서 뷰 객체의 컬럼을 액세스하려는 경우에는 뷔 정의를 참조하여 대응되는 클래스 객체의 컬럼을 액세스하도록 수정하여 실행한다. 그럼 9는 위와 같은 뷔 객체에 대한 메소드 호출의 질의 수정 방법을 나타낸다.

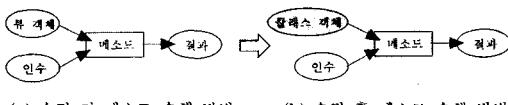
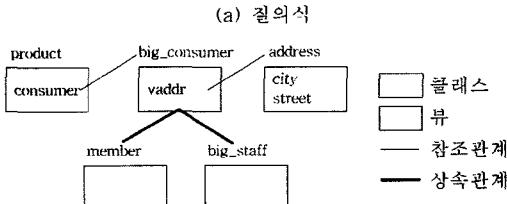


그림 9 뷔 객체의 메소드의 수정 방법

#### 3.2.4 복합 객체를 위한 확장

OODBMS/ORDBMS의 질의식은 경로식(path expression)을 통해 복합 객체의 속성을 액세스할 수 있으므로 질의식에 사용된 경로식에 나타난 뷰를 질의 수정할 수 있도록 RDBMS의 질의 수정 방법을 확장해야 한다. 경로식은 복합 객체내의 객체에 대한 연속적인 참조를 나타내는 식으로 정의된다[15]. 그림 10은 경로식이 사용된 질의식의 예를 나타낸다. 그림 10(b)와 같이 복합 객체가 구성되어 있을 때, address 클래스 객체의

```
SELECT product.consumer.vaddr.street  
FROM product  
WHERE product.consumer.vaddr.city = "Daejon"
```



### (b) 클래스 계층 구조 및 클래스-복합 계층 구조

#### 그림 10 질의식에서의 경로식 사용 예

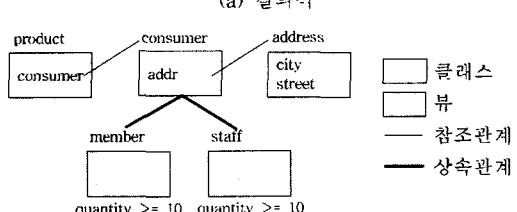
city 속성을 액세스하기 위해서는 그림 10(a)의 WHERE 절에 나타난 바와 같이 product.consumer.vaddr.city라는 경로식을 사용한다.

뷰의 속성이 포함된 경우 경로식에서 참조하는 뷰 객체는 실제 존재하지 않으므로 경로식은 그 뷰를 유도한 클래스 객체를 참조하도록 수정되어야 한다. 이를 위해 경로식에 있는 뷰의 속성을 그와 대응하는 클래스의 속성으로 수정한다. 예를 들어, 위 경로식 `product.consumer.vaddr.city` 는 `big_consumer` 뷰의 `vaddr` 속성이 `consumer` 클래스의 `addr` 속성에 대응한다고 가정하면, `product.consumer.addr.city`로 수정된다. 수정 후의 경로식에서 `product` 클래스 객체는 `big_consumer` 뷰 객체를 참조하지 않고 `consumer` 클래스 객체를 참조한다.

경로식에 나온 뷔의 속성을 클래스의 속성으로 대치하면 뷔 유도에 사용되지 않은 클래스 객체까지 참조하게 되므로 수정 전과 동일한 객체들을 참조하지 않게되는 문제가 발생한다. 이로 인해 수정 전과 수정 후의 경로식이 사용된 절의 수행 결과가 동일하지 않게 나온다. 따라서 뷔 유도에 사용되지 않은 클래스 객체는 절의 수행 결과에 나올 수 없도록 절의를 수행해야 한다.

뷰가 포함된 경로식은 위와 같은 문제점을 고려하여 다음과 같이 질의 수정된다. 경로식을 탐색하면서 뷰의 속성이 발견되면 이를 대응되는 클래스의 속성으로 대치하고 대치된 뷰 정의에 있는 WHERE 절을 질의식의 WHERE 절에 AND로 연결한다. 이로 인해 뷰의 정의를 만족시키지 못하는 클래스의 객체는 참조하지 않으므로 경로식의 수정 전과 동일한 질의 수행 결과를 내줄 수 있다. 예를 들어, 그림 10에서 `big_consumer` 뷰의 `vaddr` 속성이 `consumer` 클래스의 `addr` 속성에 대응된다고 가정할 때, 그림 11(a)와 같이 질의식에 사용

```
SELECT product.consumer.vaddr.street  
FROM product  
WHERE product.consumer.vaddr.city = "Daejon"  
      AND product.consumer.quantity >= 10
```



#### (b) 클래스 계층 구조 및 클래스-불합 계층 구조

그림 11 경로식의 질의 수정 예

된 경로식 내의 vaddr 속성은 addr 속성으로 대체되며 big\_consumer 뷰 정의의 WHERE 절인 *quantity >= 10* 이 질의식의 WHERE 절에 AND로 연결된다.

경로식에서 참조하는 뷰 객체에는 서브클래스/서브뷰가 존재할 수 있으므로 경로식의 처리에는 상속의 처리방법이 추가로 고려되어야 한다. 경로식에서 참조하는 뷰 객체에 대해 서브클래스/서브뷰가 존재하면 뷰 객체뿐만 아니라 서브클래스/서브뷰의 객체도 참조할 수 있도록 제3.2.2절에서 설명한 방식으로 질의 수정을 해야 한다. 예를 들어, 그림 11(b)와 같이 consumer 클래스의 상속 계층 구조를 수정한다.

### 3.3 ORDBMS 뷰와의 기능 비교

본 절에서는 타 ORDBMS 뷰의 기능과 제3.2절에서 제안한 새로운 구현 방법의 기능에 대해서 비교한다. 비교 항목은 오브젝트 뷰에 추가되는 객체지향 개념인 객체 식별자, 상속, 메소드, 복합 객체의 지원 여부이며, 비교 대상은 상용 ORDBMS인 ORACLE9i의 오브젝트 뷰(object view)[3], IBM DB2 Version 7의 타입 뷰(typed view)[4]이다.

비교 결과는 표 1과 같다. 본 논문에서 제안하는 방법은 네 가지 객체지향 개념을 모두 지원하지만, 상용 ORDBMS들은 일부 개념만을 지원하였다.

표 1 오브젝트 뷰의 기능 비교

기능	ORACLE9i	IBM DB2 Version 7	본 논문의 방법
객체 식별자	O	△	O
상속	△	O	O
메소드	X	X	O
복합 객체	O	O	O

각 항목 별 오브젝트 뷰의 기능 비교는 다음과 같다.

- 객체 식별자는 ORACLE9i, IBM DB2 Version 7, 그리고 본 논문의 방법 모두에서 제공하고 있다. 단, ORACLE9i, 본 논문의 방법은 시스템이 자동적으로 뷰 객체에 객체 식별자를 부여하는 방법을 사용하지만, IBM DB2 Version 7은 사용자가 뷰 객체에 객체 식별자를 부여하는 수동적인 방법을 사용한다.
- 상속은 ORACLE9i, IBM DB2 Version 7, 그리고 본 논문의 방법에서 제공하고 있다. 단, ORACLE9i에서는 클래스와 뷰가 각각 별도의 상속 계층 구조를 유지하도록 제한하고 있지만 IBM DB2 Version 7과 본 논문의 방법은 그러한 제한을 두지 않는다.
- 메소드는 본 논문의 방법에서만 제공하고 있다. ORACLE9i, IBM DB2 Version 7에서는 클래스 객체

에 대한 메소드는 제공하지만, 뷰 객체에 대한 메소드는 현재까지 제공하고 있지 않다.

- 복합 객체는 세가지 모든 방법에서 제공하고 있다.

## 4. 오디세우스에서의 뷰 구현

본 절에서는 오디세우스에서 뷰를 구현하는 방법에 대해서 다룬다. 제4.1절에서는 오디세우스에 대해서 간단히 소개하며 제4.2절에서는 오디세우스에 확장된 질의 수정 방법을 구현하는 방법에 대해서 설명한다.

### 4.1 오디세우스 소개

본 절에서는 뷰를 구현하는 기반 시스템인 오디세우스/OOSQL(ODYSSEUS/OOSQL)[7]을 소개한다. 제4.1.1절에서는 오디세우스의 전체 아키텍쳐에 대해서 간단히 설명한다. 제4.1.2절에서는 오디세우스에서 뷔를 지원하기 위해 확장되어야 하는 모듈인 질의 처리기에 대해서 설명한다.

#### 4.1.1 아키텍처

오디세우스는 클라이언트/서버 구조로 구성된다. 오디세우스 서버에는 COSMOS, OOSQL 질의 처리기가 포함되어 있다. COSMOS는 대용량 멀티미디어 객체를 지원하는 멀티미디어 객체 저장 시스템이며, OOSQL 질의 처리기는 오디세우스의 질의어인 OOSQL[16]을 처리할 수 있도록 해준다. 그리고 오디세우스 클라이언트에는 C/C++ CLI(Call Level Interface)가 포함되어 있어 사용자는 C/C++ CLI를 사용하여 응용프로그램을 개발할 수 있으며, 산업체 표준인 ODBC/JDBC 드라이버도 포함되어 있다.

오디세우스 서버에서 뷔를 지원하기 위해서는 OOSQL 질의 처리기가 확장되어야 한다. OOSQL 질의 처리기는 뷔를 정의할 수 있도록 확장되어야 하며 뷔가 사용된 질의식이 입력되었을 때 이를 질의 수정 하도록 확장되어야 한다. 다음 절에서는 뷔의 구현에 관련된 OOSQL 질의 처리기에 대해서 간단히 설명한다.

#### 4.1.2 OOSQL 질의 처리기

OOSQL은 오디세우스에서 제공하는 객체지향 질의어로서 상속과 경로식 등의 객체지향 개념을 지원한다. OOSQL의 SELECT 질의식의 구조는 그림 12와 같다. SELECT 절에는 프로젝션 필 속성이나 수식 혹은 집계 함수(aggregate functions)를 기술한다. FROM 절에는

```

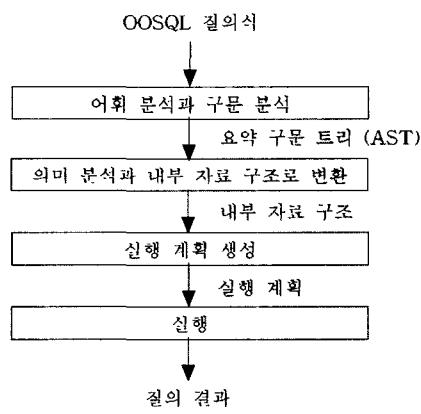
SELECT      < 프로젝션 리스트 >
FROM        < 클래스 리스트 >
[ WHERE     < 검색 조건 > ]
[ GROUP BY  < 그룹화 속성 > ]
[ HAVING    < 그룹 조건 > ]
[ ORDER BY  < 속성 리스트 > ]

```

그림 12 OOSQL의 SELECT 질의식 구조

검색의 대상이 되는 클래스들을 명시한다. WHERE 절에는 FROM 절에 나타난 클래스의 객체들이 선택될 조건을 명시한다. GROUP BY 절에는 그룹화 속성(grouping attributes)들을 명시한다. HAVING 절에는 그룹들이 선택될 조건을 명시한다. 마지막으로 ORDER BY 절에는 정렬에 사용되는 속성과 정렬 방법을 명시한다.

OOSQL 질의 처리기는 그림 13에 나타난 바와 같이 어휘 분석과 구문 분석, 의미 분석과 내부 자료 구조로의 변환, 실행 계획 생성, 실행이라는 과정을 거쳐 주어진 OOSQL 질의를 수행한다. 어휘 분석과 구문 분석에서는 주어진 질의식이 OOSQL 문법에 맞게 쓰여졌는지를 검사한 후 이를 요약 구문 트리(Abstract Syntax Tree: AST)로 변환한다. 의미 분석에서는 주어진 AST가 의미에 맞는지를 검사한 후 AST를 내부 자료 구조로 변환한다. 실행 계획 생성에서는 내부 자료 구조를 실행 계획으로 변환하며, 실행에서는 실행 계획에 따라 저장 시스템을 액세스하여 사용자가 내린 질의식을 만족하는 객체들을 저장시스템으로부터 가져오게 된다.



#### 4.2 확장된 질의 수정 방법의 구현

OOSQL 질의 처리기는 제2.2.1절에서 설명한 질의 수정 알고리즘인 알고리즘 1과 제3.2절에서 설명한 확장 방법을 사용해 뷰에 대한 질의식을 클래스에 대한 질의식으로 수정하도록 확장된다. 질의 수정은 기본적으로 알고리즘 1의 단계를 따르며 질의식에 객체 식별자, 상속, 메소드, 복합 객체가 존재하면 이를 처리하기 위해 확장한 방법을 사용한다.

질의 수정은 구현을 보다 용이하게 하기 위해 모든 뷰가 클래스만을 사용해서 정의되어 있다는 가정하에 수행된다. 만약 뷔를 정의하는 질의식에 또 다른 뷔가 포함되어 있다면 그 질의식에 질의 수정 알고리즘을 재귀적으로 수행한다. 즉, 뷔 유도 계층 구조(view-

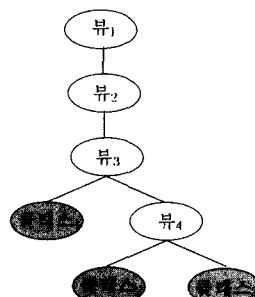


그림 14 뷔 유도 계층 구조의 예

derivation hierachy)[6]에서 끝단(leaf) 노드로부터 질의 수정을 수행하며 올라와야 한다. 예를 들어 그림 14와 같은 뷔 유도 계층 구조에서 뷔1에 대해서 질의를 수행하고자 한다면 먼저 뷔4, 뷔3, 뷔2의 순서로 질의 수정을 수행한다.

질의 수정은 효율적인 수행을 위해 그림 13의 OOSQL 질의 처리 과정 중 의미 분석 단계에서 수행된다. 질의 처리기에서는 의미 분석 도중에 현재 질의식에 뷔가 사용되었는지 아닌지 판별해 낼 수 있다. 의미 분석 중에 뷔가 질의식에 사용되었다고 판단되면, 질의 수정을 사용하여 뷔에 대한 질의를 클래스에 대한 질의로 변환한 상태의 내부 자료 구조로 변환한다.

질의 수정의 구현을 위해 의미 분석의 각 단계 별로 질의 수정을 수행한다. OOSQL 질의 처리기에서는 의미 분석이 질의식의 FROM 절, SELECT 절, WHERE 절의 순서로 이루어지기 때문에 질의 수정도 이 순서를 따라 이루어진다. FROM 절에는 객체 식별자와 상속이 사용될 수 있고, SELECT 절과 WHERE 절에는 경로식과 메소드가 사용될 수 있으므로 이를 고려하도록 질의 수정이 확장된다. 다음은 의미 분석의 각 단계 별로 확장된 질의 수정 방법을 구현하기 위한 사항을 기술한다.

##### FROM 절의 질의 수정

- 질의 대상으로 뷔가 있으면, 뷔의 정의에 사용된 클래스로 대치한다.
- 대치된 뷔가 있으면, 상속에 대한 질의 수정을 수행한다.
- 뷔의 객체 식별자가 있으면, 객체 식별자에 대한 질의 수정을 수행한다.

##### SELECT 절의 질의 수정

- 뷔가 포함된 경로식이 있으면, 경로식에 대한 질의 수정을 수행한다.
- 뷔 객체의 메소드가 있으면, 메소드에 대한 질의 수정을 수행한다.

##### WHERE 절의 질의 수정

- 뷔가 포함된 경로식이 있으면, 경로식에 대한 질의 수정을 수행한다.

- 뷰 객체의 메소드가 있으면, 메소드에 대한 질의 수정을 수행한다.
- WHERE 절에 뷰 정의에 있는 WHERE 절을 AND로 연결한다.

INSERT, UPDATE, DELETE 질의를 위한 질의 수정의 구현에는 갱신 가능한 뷰(updatable view)[17]에 대해서만 INSERT, UPDATE, DELETE 질의를 수행하도록 뷰의 갱신 가능 여부를 검사하는 과정이 추가된다. 갱신 가능한 뷰란 뷰에 대한 갱신이 클래스에 대한 갱신으로 유일하게 변환되어서 클래스의 데이터를 갱신할 수 있는 뷰를 의미한다[17,18].

## 5. 결 론

본 논문에서는 오브젝트 뷰를 위한 새로운 구현 방법을 제안하였다. 그리고, 이를 사용하여 ORDBMS인 오디세우스에 오브젝트 뷰를 구현하였다.

본 논문의 공헌은 다음과 같이 요약될 수 있다. 첫째, 오브젝트 뷰의 요구 사항을 분석하여 오브젝트 뷰에도 클래스에 적용되는 객체지향 개념인 객체 식별자, 상속, 메소드, 복합 객체가 지원되어야 함을 밝혀내었다. 둘째, 분석 결과를 토대로 기존 RDBMS에서 뷰 구현을 위해 사용하던 질의 수정 알고리즘을 오브젝트 뷰 구현을 위해 사용할 수 있도록 확장하는 방법을 제안하였다. 이를 위해, 객체 식별자, 상속, 메소드, 복합 객체와 같은 객체지향 개념들을 지원하도록 질의 수정 알고리즘을 확장하는 방법을 제안하였다. 셋째, 본 논문에서 제안한 방법이 지원하는 뷰의 기능을 상용 ORDBMS에서 제공하는 뷰의 기능들과 비교하였다. 본 방법은 객체 식별자, 상속, 메소드, 복합 객체를 모두 지원하는 반면, 네 가지 기능을 모두 지원하는 상용 ORDBMS는 없는 것으로 나타났다. 넷째, 확장된 질의 수정 방법을 오디세우스 ORDBMS에 구현하는 방법을 제시하였다. 이를 위해, 질의 처리기의 의미 분석 모듈을 확장하여 질의 수정을 구현하는 방법을 제안하였다.

현재 상용 ORDBMS에는 오브젝트 뷰가 구현되어 있으나 상세한 구현 방법이 발표되어 있지 않기 때문에, 실제로 오브젝트 뷰를 구현하기에는 어려움이 있다. 본 논문은 이러한 오브젝트 뷰의 상세한 구현 방법과 구현 경험을 제시하였다는 점에서 큰 의미가 있다.

## 참 고 문 헌

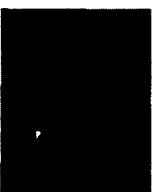
- [1] Stonebraker, M. and Moore, D., *Object-Relational DBMSs: The Next Great Wave*, Morgan Kaufmann, 1999.
- [2] Kim, W., "Object-oriented database systems: Promises, reality, and future," In Proc. 19th Int. Conf. on Very Large Data Bases, Dublin, Ireland, pp. 676~687, Aug. 1993.
- [3] Oracle, Oracle9i SQL Reference Manual, 2001.
- [4] IBM, DB2 SQL Reference Manual, 2001.
- [5] Motschnig-Pitrik, R., "Requirements and Comparison of View Mechanisms for Object-Oriented Databases," *Int'l Journal on Information Systems*, Vol. 21, No. 3, pp. 229~252, 1996.
- [6] Kim, W., *Modern Database Systems*, ACM Press, 1995.
- [7] 한옥신, 이민재, 이재길, 박상영, 황규영, "오디세우스 객체관계형 멀티미디어 DBMS의 아키텍처", 한국정보과학회 추계학술발표회, 27권, 2호, pp. 45~47, 2000년 10월.
- [8] Ceri, S. and Widom, J., "Deriving Production Rules for Incremental View Maintenance," In Proc. 17th Int. Conf. on Very Large Data Bases, Barcelona, Spain, pp. 577~589, Sept. 1991.
- [9] Date, C., *A Guide to the SQL Standard*, Addison Wesley, 1987.
- [10] Stonebraker, M., "Implementation of Integrity Constraints and Views by Query Modification," In Proc. Int'l Conf. on Management of Data, ACM SIGMOD, pp. 65~78, May 1975.
- [11] Shmueli, O. and Itai, A., "Maintenance of Views," In Proc. Int'l Conf. on Management of Data, ACM SIGMOD, pp. 240~256, June 1984.
- [12] Blakeley, J., Larson, P., and Tompa, F., "Efficiently Updating Materialized Views," In Proc. Int'l Conf. on Management of Data, ACM SIGMOD, pp. 61~71, June 1986.
- [13] Hanson, E., "A Performance Analysis of View Materialization Strategies," In Proc. Int'l Conf. on Management of Data, ACM SIGMOD, pp. 440~453, May 1987.
- [14] Carey, M., Rielau, S., and Vance, B., "Object View Hierarchies in DB2 UDB," In Proc. Int'l Conf. on Extending Database Technology, EDBT, pp. 478~492, Mar. 2000.
- [15] Kim, W., *Introduction to Object-Oriented Databases*, MIT Press, 1990.
- [16] 우준호, ODYSSEUS 객체지향 데이터베이스 시스템을 위한 질의 처리기의 설계 및 구현, 석사 학위 논문, KAIST 전산학과, 1995.
- [17] Dayal, U. and Bernstein, P., "On the Correct Translation of Update Operations on Relational Views," *ACM Trans. on Database Systems*, Vol. 8, No. 2, pp. 381~416, 1982.
- [18] Bancilhon, F. and Spyros, N., "Update Semantics of Relational Views," *ACM Trans. on Database Systems*, Vol. 6, No. 4, pp. 557~575, 1981.



### 이재길

1999년 2월~현재 한국과학기술원 전자전산학과 전산학전공 박사과정. 1997년 3월~1999년 2월 한국과학기술원 전자전산학과 전산학전공 석사. 1993년 3월~1997년 2월 한국과학기술원 전자전산학과 전산학전공 학사. 2001년 7월~2001

년 8월 Visiting Scholar, 미국 HP Labs. 관심분야는 객체관계형 데이터베이스 시스템, 정보 검색, 질의 최적화, XML 데이터베이스, 데이터베이스 보안



### 한욱신

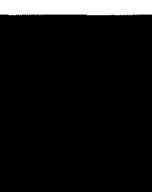
2003년 3월~현재 전임강사, 컴퓨터공학과, 경북대학교. 2002년 9월~2003년 2월 연구교수, 첨단정보기술연구센터, KAIST. 2001년 9월~2002년 8월 포스트닥, 첨단정보기술연구센터, KAIST. 1996년 3월~2001년 8월 Ph.D., 전산학과, KAIST.

1994년 3월~1996년 2월 MS, 전산학과, KAIST. 1990년 3월~1994년 2월 BS, 컴퓨터공학과, 경북대학교. 2002년 7월 방문연구원, 미국 HP Labs. 2001년 7월~2001년 8월 Visiting Scholar, 미국 HP Labs



### 이민재

1995년 2월 한국과학기술원 전자전산학과 전산학전공 학사. 1997년 2월 한국과학기술원 전자전산학과 전산학전공 석사. 1997년 3월~현재 한국과학기술원 전자전산학과 전산학전공 박사과정. 관심분야는 지리 정보 시스템, 객체 관계형 데이터베이스 시스템



### 이종학

1982년 경북대학교 전자공학과(전자계산전공) 졸업(학사). 1984년 한국과학기술원 전산학과 졸업(공학석사). 1997년 한국과학기술원 전산학과 졸업(공학박사) 1991년 정보처리기술사. 1984년~1987년 금성통신(주) 부설연구소 주임연구원

1987년~1998년 한국통신 연구개발본부 선임연구원. 1998년~현재 대구가톨릭대학교 컴퓨터정보통신공학부 교수. 관심분야는 데이터베이스 시스템, 객체 데이터베이스, 트랜잭션 프로세싱, 지리정보 시스템 등



### 황규영

1973년 서울대학교 전자공학과 졸업(B.S.). 1975년 한국과학기술원 전기 및 전자학과 졸업(M.S.). 1982년 Stanford University(M.S.). 1983년 Stanford University(Ph.D.). 1975년~1978년 국방과학연구소(ADD), 선임연구원. 1983년~

1990년 IBM T.J. Watson Research Center, Research Staff Member. 1992년~1994년 한국정보과학회 데이타베이스 연구회(SIGDB) 운영위원장. 1995년 한국정보과학회 이사 겸 논문지 편집위원장. 1999년~2000년 한국정보과학회 부회장. Editor-in-Chief: the VLDB Journal, 2003년~현재 Editor: Distributed and Parallel Databases: An International Journal, 1991년~1995년 Editor: International Journal of Geographical Information Systems 1994년~현재 Associate Editor: IEEE Data Engineering Bulletin, 1990년~1993년 IEEE Transactions on Knowledge and Data Engineering, 2002년~현재. 1998년~2004년 Trustee, The VLDB Endowment. 1999년~2005년 Steering Committee Member, DASFAA. 1990년~현재 한국과학기술원 전자전산학과 전산학전공 교수 1999년~현재 첨단정보기술연구센터(과학재단 우수연구센터) 소장. 관심분야는 데이터베이스 시스템, 멀티미디어 GIS