제 8 장 뷰와 시스템 카탈로그

메타 데이터

8.1 뷰
8.2 관계 DBMS의 시스템 카탈로그
8.3 MS SQL Server의 시스템 카탈로그
- 연습문제

8장. 뷰와 시스템 카탈로그

❑ 뷰와 시스템 카탈로그

- ✓ 관계 데이터베이스 시스템의 뷰(view)는 다른 릴레이션으로부터 유도된 릴레이션(derived relation)으로서 ANSI/SPARC 3단계 아키텍처의 외부 뷰와 다름
- ✓ 뷰는 관계 데이터베이스 시스템에서 데이터베이스의 보안 메카니즘으로서, 복잡한 질의를 간단하게 표현하는 수단으로서, 데이터독립성을 높이기 위해서 사용됨
- ✓ 시스템 카탈로그는 시스템내의 객체(기본 릴레이션, 뷰, 인덱스, 사용자, 접근 권한 등)에 관한 정보를 포함
- ✓ 시스템 카탈로그를 적절히 활용하면 원하는 릴레이션을 데이터베이스에서 찾고, 그 릴레이션에 어떤 애트리뷰트들이 들어 있으며, 각 애트리뷰트의 데이터 타입은 무엇인가 등을 쉽게 파악할 수 있음

8.1 뷰

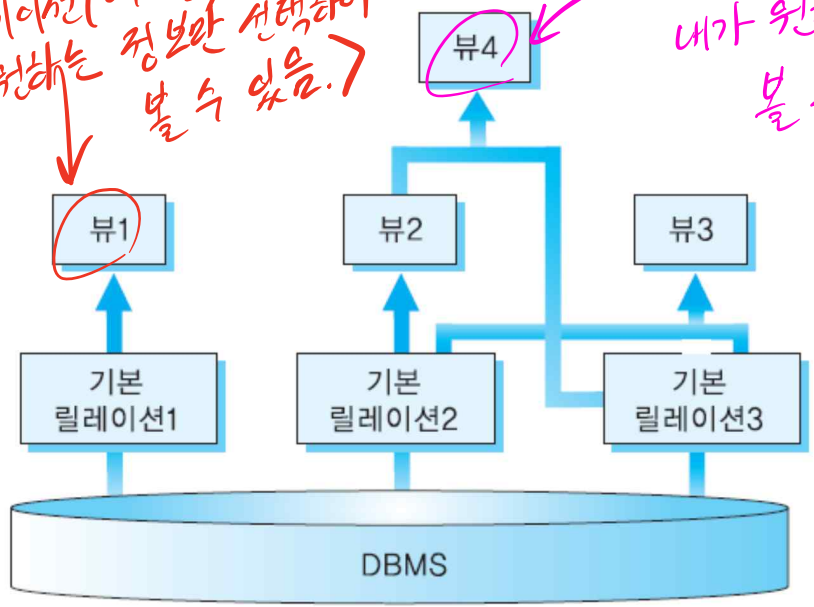
❑ 뷰의 개요

- ✓ ANSI/SPARC 3단계 아키텍처에서 외부 뷰는 특정 사용자가 보는 데이터베이스의 구조
- ✓ 관계 데이터베이스에서의 뷰는 한 사용자의 전체 외부 뷰 대신에 하나의 가상 릴레이션(virtual relation)을 의미
- ✓ 뷰는 기존의 기본 릴레이션(base relation. 실제 릴레이션)에 대한 SELECT문의 형태로 정의됨
- ✓ 사용자는 여러 개의 릴레이션과 뷰를 사용할 수 있음
- ✓ 뷰는 릴레이션으로부터 데이터를 검색하거나 갱신할 수 있는 동적인 창(dynamic window)의 역할

8.1 뷰(계속)

- 스냅샷(snapshot) :- 실제화 뷰(materialized view)
  - ✓ 어느 시점에 SELECT문의 결과를 기본 릴레이션의 형태로 저장해 놓은 것
  - ✓ 스냅샷은 사진을 찍은 것과 같아서 스냅샷을 정의하는 시점의 기본 릴레이션의 내용이 스냅샷에 반영됨
  - ✓ 어떤 시점의 조직체의 현황, 예를 들어 몇년 몇월 시점에 근무하던 사원들의 정보, 재고 정보 등이 스냅샷으로 정의될 수 있음

8.1 뷰(계속)



[그림 8.1] 뷰와 기본 릴레이션의 관계

8.1 뷰(계속)

❑ 뷰의 정의

✓ 뷰를 정의하는 SQL문의 구문

```
CREATE VIEW 뷰이름 [(애트리뷰트(들))]
AS SELECT문
[WITH CHECK OPTION];
```

✓ 뷰의 이름 다음에 애트리뷰트들을 생략하면 뷰를 정의하는데 사용된 SELECT문의 SELECT절에 열거된 애트리뷰트들의 이름과 동일한 애트리뷰트들이 뷰에 포함됨

✓ 뷰를 정의하는 SELECT절에 산술식 또는 집단 함수에 사용된 애트리뷰트가 있는 경우, 뷰의 정의에 조인이 포함되어 있고 두 개 이상의 다른 릴레이션으로부터 가져온 애트리뷰트들의 이름이 같아서 뷰에서 두 개 이상의 애트리뷰트의 이름이 같게 되는 경우에는 뷰를 정의할 때 모든 애트리뷰트들의 이름을 지정해야 함 (이름 지정이 필요한 경우)

8.1 뷰(계속)

예: 한 릴레이션 위에서 뷰를 정의

그림 4.8의 EMPLOYEE 릴레이션에 대해서 “3번 부서에 근무하는 사원들의 사원번호, 사원이름, 직책으로 이루어진 뷰”를 정의해보자. 아래의 뷰의 정의에는 뷰의 애트리뷰트들을 별도로 명시했기 때문에 뷰에는 EMPNO, EMPNAME, TITLE의 세 애트리뷰트가 포함됨

```
CREATE VIEW  EMP_DNO3 (ENO, ENAME, TITLE)
AS SELECT    EMPNO, EMPNAME, TITLE
   FROM      EMPLOYEE
   WHERE     DNO=3;
```

8.1 뷰(계속)

EMPLOYEE

EMPNO	EMPNAME	TITLE	MANAGER	SALARY	DNO
2106	김창섭	대리	1003	2500000	2
3426	박영권	과장	4377	3000000	1
3011	이수민	부장	4377	4000000	3
1003	조민희	과장	4377	3000000	2
3427	최종철	사원	3011	1500000	3
1365	김상원	사원	3426	1500000	1
4377	이성래	사장	∧	5000000	2

[그림 8.2] EMP_DNO3을 통해서 접근할 수 있는 부분(파란색 음영)

emp-dno3

ENO	ENAME	TITLE
3011	이수민	부장
3427	최종철	사원

← VIEW 결과물

8.1 뷰(계속)

예: 두 릴레이션 위에서 뷰를 정의

그림 4.8의 EMPLOYEE와 DEPARTMENT 릴레이션에 대해서 "기획부에 근무하는 사원들의 이름, 직책, 급여로 이루어진 뷰"를 정의해보자. 아래의 뷰의 정의에는 뷰의 애트리뷰트들을 별도로 명시하지 않았기 때문에 뷰에 속하는 애트리뷰트들의 이름은 기본 릴레이션의 애트리뷰트들의 이름과 같다. 즉 뷰에는 EMPNAME, TITLE, SALARY의 세 애트리뷰트가 포함된다.

```
CREATE VIEW EMP_PLANNING
AS SELECT   E.EMPNAME, E.TITLE, E.SALARY
   FROM     EMPLOYEE E, DEPARTMENT D
   WHERE    E.DNO=D.DEPTNO
            AND D.DEPTNAME = '기획';
```

equijoin

8.1 뷰(계속)

❑ 뷰를 사용하여 데이터를 접근할 때 관계 DBMS에서 거치는 과정

✓ 시스템 카탈로그로부터 뷰의 정의, 즉 SELECT문을 검색

✓ 기본 릴레이션에 대한 뷰의 접근 권한을 검사

✓ 뷰에 대한 질의를 기본 릴레이션에 대한 동등한 질의로 변환

```
SELECT  *                          SELECT  EMPNO, EMPNAME, TITLE
FROM    EMP_DNO3                   FROM    EMPLOYEE
WHERE   TITLE='사원';        →     WHERE   TITLE='사원'
                                           AND DNO=3;
```

✓ 질의 변경(query modification)이라고 함

8.1 뷰(계속)

❑ 뷰의 장점

① 뷰는 복잡한 질의를 간단하게 표현할 수 있게 함

- 기획부에 근무하는 사원들 중에서 직책이 부장인 사원의 사원이름과 급여를 검색하는 질의를 기본 릴레이션을 사용하여 표현하면 아래와 같이 다소 복잡한 형태의 질의가 됨

```
SELECT   E.EMPNAME, E.SALARY
FROM     EMPLOYEE E, DEPARTMENT D
WHERE    D.DEPTNAME = '기획'
         AND D.DEPTNO = E.DNO
         AND E.TITLE = '부장';
```

- 뷰에 대해서 같은 결과를 검색하는 질의를 표현하면

```
SELECT   EMPNAME, SALARY
FROM     EMP_PLANNING
WHERE    TITLE = '부장';
```

8.1 뷰(계속)

❑ 뷰의 장점(계속)

② 뷰는 데이터 무결성을 보장하는데 활용됨

- 기본적으로 뷰를 통해 투플을 추가하거나 수정할 때 투플이 뷰를 정의하는 SELECT문의 WHERE절의 기준에 맞지 않으면 뷰의 내용에서 사라짐

```
UPDATE   EMP_DNO3
SET      DNO = 2
WHERE    ENO = 3427;
```

- 이 뷰의 정의할 때 WITH CHECK OPTION을 명시했다고 가정

```
CREATE VIEW  EMP_DNO3 (ENO, ENAME, TITLE)
AS  SELECT   EMPNO, EMPNAME, TITLE
    FROM     EMPLOYEE
    WHERE    DNO = 3
    WITH CHECK OPTION;
```

8.1 뷰(계속)

❑ 뷰의 장점(계속)

3) 뷰는 데이터 독립성을 제공함

- 뷰는 데이터베이스의 구조가 바뀌어도 기존의 질의(응용 프로그램)를 다시 작성할 필요성을 줄이는데 사용될 수 있음
- 예: 응용의 요구사항이 변경되어 기존의 EMPLOYEE 릴레이션이 두 개의 릴레이션 EMP1(EMPNO, EMPNAME, SALARY)과 EMP2(EMPNO, TITLE, MANAGER, DNO)로 분해되었다고 가정하자. 응용 프로그램에서 기존의 EMPLOYEE 릴레이션을 접근하던 SELECT문은 더 이상 수행되지 않으므로, EMP1과 EMP2에 대한 SELECT문으로 변경해야 한다.
- 아래와 같이 EMPLOEE라는 뷰를 정의했다면 응용 프로그램에서 EMPLOYEE 릴레이션을 접근하던 SELECT문은 계속해서 수행될 수 있음

8.1 뷰(계속)

❑ 뷰의 장점(계속)

✓ 뷰는 데이터 독립성을 제공함(계속)

```
CREATE VIEW  EMPLOYEE
AS  SELECT   E1.EMPNO, E1.EMPNAME, E2.TITLE, E2.MANAGER,
             E1.SALARY, E2.DNO
    FROM     EMP1 E1, EMP2 E2
    WHERE    E1.EMPNO = E2.EMPNO;
```

8.1 뷰(계속)

기존 테이블에 존재하는 데이터에 대한 보안

뷰의 장점(계속)

(1) 뷰는 데이터 보안 기능을 제공함

- 뷰는 뷰의 원본이 되는 기본 릴레이션에 직접 접근할 수 있는 권한을 부여하지 않고 뷰를 통해 데이터를 접근하도록 하기 때문에 보안 메커니즘으로 사용할 수 있음
- 뷰는 일반적으로 기본 릴레이션의 일부 애트리뷰트들 또는 일부 투플들을 검색하는 SELECT문으로 정의되므로 뷰를 통해서 기본 릴레이션을 접근하면 기본 릴레이션의 일부만 검색할 수 있음
- 예: EMPLOYEE 릴레이션의 SALARY 애트리뷰트는 숨기고 나머지 애트리뷰트들은 모든 사용자가 접근할 수 있도록 하려면 SALARY 애트리뷰트를 제외하고 EMPLOYEE 릴레이션의 모든 애트리뷰트를 포함하는 뷰를 정의하고, 사용자에게 뷰에 대한 SELECT 권한을 허가

8.1 뷰(계속)

- 뷰의 장점(계속)
  - ✓ 동일한 데이터에 대한 여러 가지 뷰를 제공함
    - 뷰는 사용자들의 그룹이 각자 특정한 기준 및 자격에 따라 구분하여 데이터를 접근하도록 함

8.1 뷰(계속)

-X View로 갱신 잘 안함.

❑ 뷰의 갱신

- ✓ 뷰에 대한 갱신도 기본 릴레이션에 대한 갱신으로 변환됨
- ✓ 아래의 갱신들이 성공적으로 수행될 수 있는가?
- ✓ 갱신 1: 한 릴레이션 위에서 정의된 뷰에 대한 갱신

```
INSERT INTO EMP_DNO3
VALUES (4293, '김정수', '사원');
```

```
INSERT INTO EMPLOYEE
VALUES (4293, '김정수', '사원', , , );
```

8.1 뷰(계속)

❑ 뷰의 갱신(계속)

✓ 갱신 2: 두 개의 릴레이션 위에서 정의된 뷰에 대한 갱신

```
INSERT INTO EMP_PLANNING
VALUES ('박지선', '대리', 2500000);
```

```
INSERT INTO EMPLOYEE
VALUES(, '박지선', '대리', , 2500000, );
```

8.1 뷰(계속)

❑ 뷰의 갱신(계속)

✓ 갱신 3: 집단 함수 등을 포함한 뷰에 대한 갱신

```
CREATE VIEW    EMP_AVGSAL (DNO, AVGSAL)
AS  SELECT     DNO, AVG(SALARY)
    FROM       EMPLOYEE
    GROUP BY   DNO;
```

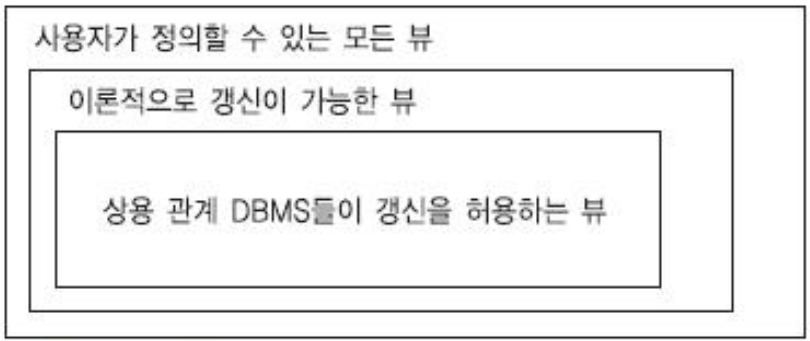
```
UPDATE   EMP_AVGSAL
SET      AVGSAL = 3000000
WHERE    DNO = 2;
```

```
INSERT INTO EMP_AVGSAL
VALUES (3, 3200000);
```

8.1 뷰(계속)

- 갱신이 불가능한 뷰
  - ✓ 한 릴레이션 위에서 정의되었으나 그 릴레이션의 기본 키가 포함되지 않은 뷰
  - ✓ 기본 릴레이션의 애트리뷰트들 중에서 뷰에 포함되지 않은 애트리뷰트에 대해 NOT NULL이 지정되어 있을 때
  - ✓ 집단 함수가 포함된 뷰
  - ✓ 조인으로 정의된 뷰

8.1 뷰(계속)



[그림 8.3] 갱신 가능성 기준에 따른 뷰들의 유형

8.2 관계 DBMS의 시스템 카탈로그

❑ 시스템 카탈로그

- ✓ 시스템 카탈로그는 데이터베이스의 객체(사용자, 릴레이션, 뷰, 인덱스, 권한 등)와 구조들에 관한 모든 데이터를 포함
- ✓ 시스템 카탈로그를 메타 데이터베이스라고 함. 메타데이터는 데이터에 관한 데이터라는 의미
- ✓ 시스템 카탈로그는 사용자 및 질의 최적화 모듈 등 DBMS 자신의 구성요소에 의해서 사용됨
- ✓ 시스템 카탈로그는 관계 DBMS마다 표준화되어 있지 않아서 관계 DBMS마다 서로 다른 형태로 시스템 카탈로그 기능을 제공함
- ✓ 시스템 카탈로그는 데이터 사전(data dictionary) 또는 시스템 테이블이라고도 부름

8.2 관계 DBMS의 시스템 카탈로그(계속)

❑ 시스템 카탈로그가 질의 처리에 어떻게 활용되는가

```
SELECT   EMPNAME, SALARY, SALARY * 1.1
FROM     EMPLOYEE
WHERE    TITLE = '과장' AND DNO = 2;
```

1. SELECT문이 문법적으로 정확한가를 검사함
2. SELECT 문에서 참조하는 EMPLOYEE 릴레이션이 데이터베이스에 존재하는가를 검사함
3. EMPLOYEE 릴레이션에 SELECT절에 열거된 애트리뷰트와 WHERE절에서 조건에 사용된 애트리뷰트가 존재하는가를 확인함
4. SALARY 애트리뷰트가 수식에 사용되었으므로 이 애트리뷰트의 데이터 타입이 숫자형(정수형이나 실수형)인가를 검사하고, TITLE이 문자열과 비교되었으므로 이 애트리뷰트의 데이터 타입이 문자형(CHAR(n) 또는 VARCHAR(n) 등)인가 등을 검사함

where 절 domain 검사!

8.2 관계 DBMS의 시스템 카탈로그(계속)

❑ 시스템 카탈로그가 질의 처리에 어떻게 활용되는가(계속)

- ✓ 이 질의를 입력한 사용자가 EMPLOYEE 릴레이션의 EMPNAME, SALARY 애트리뷰트를 검색할 수 있는 권한이 있는가를 확인함
- ✓ TITLE 애트리뷰트와 DNO 애트리뷰트에 인덱스가 정의되어 있는지 확인함
- ✓ 두 애트리뷰트에 각각 인덱스가 존재한다고 가정하자. DBMS가 두 인덱스 중에서 조건을 만족하는 투플 수가 적은 것을 선택하기 위해서는 관계 데이터베이스 시스템에 데이터베이스 외에 추가로 정보를 유지해야 함
- ✓ 한 릴레이션의 전체 투플 수와 그 릴레이션에 정의된 각 인덱스에 존재하는 상이한 값들의 개수를 유지한다면 어느 인덱스를 사용하는 것이 유리한가를 예상할 수 있음

8.2 관계 DBMS의 시스템 카탈로그(계속)

- 시스템 카탈로그가 질의 처리에 어떻게 활용되는가(계속)
  - ✓ 그림 4.8에서 EMPLOYEE 릴레이션의 전체 투플 수는 7이고, TITLE 애트리뷰트에는 사원, 대리, 과장, 부장, 사장의 다섯 가지 값들이 존재함
  - ✓ DNO 애트리뷰트에는 1, 2, 3의 세 가지 값들이 존재함
  - ✓ 따라서 TITLE 애트리뷰트에 정의된 인덱스가 DNO에 정의된 인덱스보다 대상 투플들을 더 좁혀 주므로 유리함

8.2 관계 DBMS의 시스템 카탈로그(계속)

❑ 질의 최적화

- ✓ DBMS가 질의를 수행하는 여러 가지 방법들 중에서 가장 비용이 적게 드는 방법을 찾는 과정
- ✓ 질의 최적화 모듈이 정확한 결정을 내릴 수 있도록 DBMS는 자체 목적을 위해서 시스템 카탈로그에 다양한 정보를 유지함
- ✓ 사용자가 질의 최적화 모듈을 깊이 있게 이해할 필요는 없지만 질의 최적화 모듈이 정확한 수행 방법을 결정하기 위해서는 릴레이션에 관한 다양한 통계 정보가 정확하게 유지돼야 한다는 것을 알고 있는 것이 바람직

8.2 관계 DBMS의 시스템 카탈로그(계속)

❑ 관계 DBMS의 시스템 카탈로그

- ✓ 시스템 카탈로그에는 릴레이션, 애트리뷰트, 인덱스, 사용자, 권한 등 각 유형마다 별도의 릴레이션이 유지됨
- ✓ 시스템 카탈로그는 사용자 릴레이션처럼 SELECT문을 사용하여 내용을 검색할 수 있음
- ✓ 사용자 릴레이션과 마찬가지 형태로 저장되기 때문에 사용자 릴레이션에 적용되는 회복 기법과 동시성 제어 기법을 동일하게 사용할 수 있음
- ✓ EMPLOYEE 릴레이션과 DEPARTMENT 릴레이션에 대해서 시스템 카탈로그에 어떤 정보들이 유지되는가를 이해하기 쉽도록 시스템 카탈로그를 매우 단순화하여 설명함
- ✓ 릴레이션에 관한 정보를 유지하는 릴레이션의 이름이 SYS_RELATION, 애트리뷰트에 관한 정보를 유지하는 릴레이션의 이름이 SYS_ATTRIBUTE라고 가정

8.2 관계 DBMS의 시스템 카탈로그(계속)

SYS_RELATION

RelId	RelOwner	RelTups	RelAtts	RelWidth
EMPLOYEE	KIM	7	6	36
DEPARTMENT	KIM	4	3	18
...	...	...	...	...

SYS_ATTRIBUTE

AttrelId	AttID	AttName	AttOff	AttType	AttLen	PkorFk
EMPLOYEE	1	EMPNO	0	int	4	Pk
EMPLOYEE	2	EMPNAME	4	char	10	
EMPLOYEE	3	TITLE	14	char	10	
EMPLOYEE	4	MANAGER	24	int	4	Fk
EMPLOYEE	5	SALARY	28	int	4	
EMPLOYEE	6	DNO	32	int	4	Fk
DEPARTMENT	1	DEPTNO	0	int	4	Pk
DEPARTMENT	2	DEPTNAME	4	char	10	
DEPARTMENT	3	FLOOR	14	int	4	
...	...	...	...	...	...	

[그림 8.4] SYS_RELATION과 SYS_ATTRIBUTE의 내용

8.2 관계 DBMS의 시스템 카탈로그(계속)

- 시스템 카탈로그의 갱신
  - ✓ 어떤 사용자도 시스템 카탈로그를 직접 갱신할 수 없음
  - ✓ 즉 DELETE, UPDATE 또는 INSERT문을 사용하여 시스템 카탈로그를 변경할 수 없음

8.2 관계 DBMS의 시스템 카탈로그(계속)

❑ 시스템 카탈로그에 유지되는 통계 정보

✓ 릴레이션마다

투플의 크기, 투플 수, 각 블록의 채우기 비율, 블록킹 인수,
릴레이션의 크기(블록 수)

✓ 뷰마다

뷰의 이름과 정의

✓ 애트리뷰트마다

애트리뷰트의 데이터 타입과 크기, 애트리뷰트 내의 상이한 값들의 수,
애트리뷰트 값의 범위, 선택율 (조건을 만족하는 투플 수/전체 투플 수)

8.2 관계 DBMS의 시스템 카탈로그(계속)

❑ 시스템 카탈로그에 유지되는 통계 정보(계속)

✓ 사용자마다

접근할 수 있는 릴레이션과 권한

✓ 인덱스마다

인덱스된 애트리뷰트(키 애트리뷰트 또는 비 키 애트리뷰트), 클러스터링 인덱스/비 클러스터링 인덱스 여부, 밀집/희소 인덱스 여부, 인덱스의 높이, 1단계 인덱스의 블록 수

8.3 SQL Server의 시스템 카탈로그

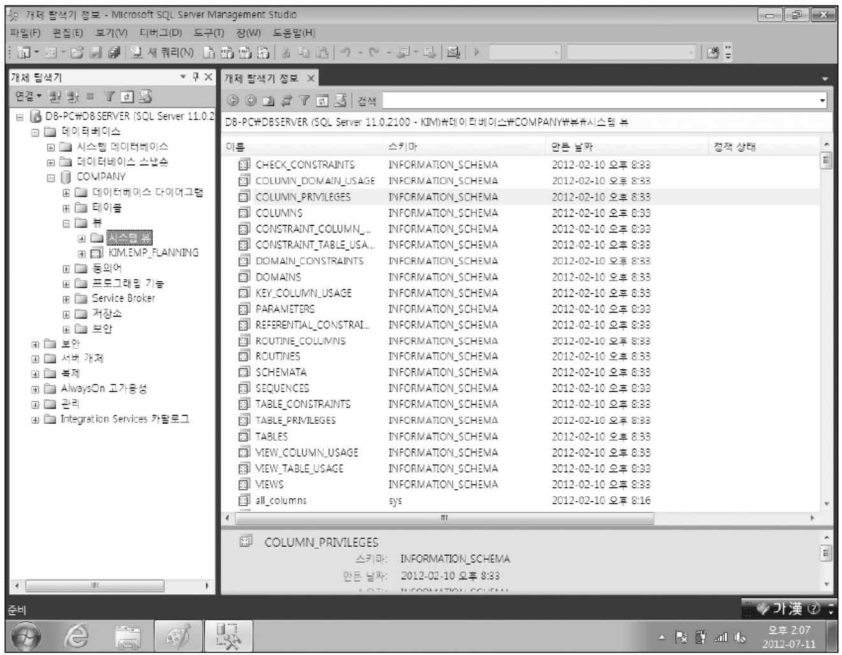
- MS SQL Server의 시스템 카탈로그
  - ✓ SQL Server는 릴레이션의 데이터가 변경될 때 주기적으로 통계 정보를 자동으로 갱신함
  - ✓ 통계 정보는 UPDATE STATISTICS문을 사용하여 수동으로 갱신할 수 있음
  - ✓ 통계 정보를 갱신할 필요성이 있는가를 확인하기 위해서 샘플링 방법을 사용함
  - ✓ 통계 정보가 갱신되는 빈도는 애트리뷰트 또는 인덱스의 크기와 변경되는 데이터의 양에 따라 결정됨
  - ✓ 모든 릴레이션들에 대한 구성을 정의하는 데이터를 시스템 테이블이라고 부르는 특수한 테이블 집합에 저장함
  - ✓ 사용자나 응용 프로그램이 정보 스키마 뷰, Transact-SQL문 및 함수, 시스템 저장 프로시저 등을 사용하여 시스템 테이블에 저장된 정보를 검색함

8.3 SQL Server의 시스템 카탈로그(계속)

〈표 8.1〉 각 데이터베이스에 있는 주요 시스템 테이블들의 이름과 기능

이름	기능
syscolumns	모든 테이블 및 뷰의 모든 열에 대해 한 투플을 포함하고, 저장 프로시저의 각 매개 변수에 대해 한 투플을 포함한다.
syscomments	뷰, 규칙, 기본값, 트리거, CHECK 제약 조건, DEFAULT 제약 조건 및 저장 프로시저에 대한 항목을 포함한다.
sysconstraints	개체의 제약 조건을 나타낸다.
sysdepends	개체(뷰, 프로시저 및 트리거)와 그 정의에 포함된 개체(테이블, 뷰 및 프로시저) 간의 종속 관계에 관한 정보를 포함한다.
sysforeignkeys	릴레이션 정의에 있는 외래 키 제약 조건에 관한 정보를 포함한다.
sysindexes	각 인덱스에 대해 한 투플을 포함한다.
syskeys	각 기본 키, 외래 키에 관해 한 투플을 포함한다.
sysobjects	각 개체(제약 조건, 로그, 규칙, 저장 프로시저 등)에 대해 한 투플을 포함한다.
syspermissions	데이터베이스의 사용자, 그룹, 역할에게 부여 또는 거부된 권한에 대한 정보를 포함한다.
sysprotects	GRANT문과 함께 보안 계정에 적용되는 권한에 관한 정보를 포함한다.
sysreferences	참조되는 애트리뷰트에 대한 외래 키 제약조건에 관해 한 투플을 포함한다.
systypes	각 시스템 정의 및 각 사용자 정의 데이터 형식에 대해 한 투플을 포함한다.
sysusers	데이터베이스 내의 각 사용자, 윈도우 그룹, 역할에 대해 한 투플을 포함한다.

8.3 SQL Server의 시스템 카탈로그(계속)



[그림 8.5] INFORMATION_SCHEMA 스키마에 들어 있는 시스템 뷰

8.3 SQL Server의 시스템 카탈로그(계속)

〈표 8.2〉 시스템 저장 프로시저들의 이름과 기능

이름	기능
sp_column_privileges	릴레이션의 애트리뷰트 권한 정보를 반환한다.
sp_depends	뷰가 종속하는 릴레이션과 뷰에 관한 정보를 표시한다.
sp_fkeys	외래 키 정보를 반환한다.
sp_helpindex	릴레이션에 정의된 모든 인덱스들에 관한 정보를 표시한다.
sp_helprotect	데이터베이스의 개체에 대한 사용자 권한에 관한 정보를 보여준다.
sp_helptext	뷰를 정의하는데 사용된 SELECT문을 표시한다.
sp_pkeys	릴레이션의 기본 키 정보를 반환한다.
sp_spaceused	릴레이션이 사용하는 디스크 공간 정보를 보여준다.
sp_statistics	릴레이션 또는 인덱스된 뷰에 관한 모든 인덱스 및 통계 정보를 반환한다.
sp_tables	질의할 수 있는 개체(FROM 절에 나타날 수 있는 모든 개체)의 목록을 반환한다.
sp_table_privileges	릴레이션에 대한 SELECT, INSERT, DELETE, UPDATE와 같은 권한 목록을 반환한다.