

# 제 9 장

## 무결성과 보안



SQL Server

ORACLE®

# 목 차

9.1 무결성 서브시스템

9.2 무결성 규정

9.3 무결성 규정의 명세

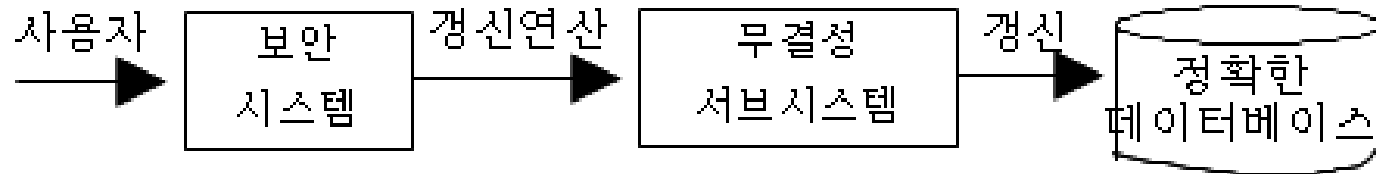
9.4 보안의 개념

9.5 데이터베이스 보안과 구현

9.6 권한 부여 명세 기법

## 9.1 무결성 서브시스템

- 정밀성(accuracy), 정확성(correctness)
- 데이터베이스에 저장된 데이터 값을 정확하게 유지함
- 일관성(consistency)
  - : 병행트랜잭션들의 상호작용으로부터 의미상의 모순이 없도록 함
- 신뢰성(reliability)
  - : H/W, S/W시스템의 오동작으로부터 오류가 없도록 함



무결성 서브시스템

## 9.2 무결성 규정

- ① 규정 이름 : 규정을 참조할 때 사용하는 식별자
- ② 검사 시기 : 트랜잭션의 접근 유형 및 데이터와 함께 검사해야 될 시기 를 명세
- ③ 제약 조건 : 데이터가 만족해야 될 제약 조건
- ④ 위반 조치 : 제약 조건 검사 결과, 위반이 발견되었을 때 취해야 될 대응 조치를 명세

```
R1 : AFTER UPDATING STUDENT.YEAR  
      CHECK(STUDENT.YEAR>0)  
      ELSE  
        DO  
          PRINT "R1 violated"  
          REJECT  
      END;
```

## 9.2 무결성 규정

- 도메인 무결성 규정

```
CREATE DOMAIN NAME CHAR(20);  
CREATE DOMAIN WEIGHT DECIMAL(5,1)  
    CHECK(WEIGHT > 0 AND WEIGHT < 300 );
```

```
CREATE DOMAIN SNO CHAR(5)  
    CHECK(SUBSTR(SNO, 1, 1)= 'S'  
        AND IS_NUMBER(SUBSTR(SNO, 2, 4)))  
    ELSE  
        REJECT;
```

## 9.2 무결성 규정

- 릴레이션 무결성 규정

**R2: WHEN UPDATE EMP.SAL  
CHECK(EMP.NEWSAL>EMP.OLDSAL);**

**R3: AFTER UPDATING EMP.SAL  
CHECK(AVG(EMP.SAL)≤300);**

**R4: AFTER UPDATING EMP.SAL  
CHECK(EMP.SAL≤500);**

**R5: AFTER UPDATING EMP.SEX  
CHECK(SEX='M' OR SEX='F');**

**R6: WHEN COMMIT  
CHECK(SUM(ACCONUT.BALANCE)=SUMMARY.TOTAL);**

## 9.3 무결성 규정의 명세

- 트리거(trigger)

```
DEFINE TRIGGER STUDINS  
ON INSERTION OF STUDENT:  
(UPDATE DEPR  
    SET NSRUD=NSTUD + 1  
    WHERE DNO=NEW_STUDENT.DNO);
```

```
DEFINE TRIGGER STUDEL  
ON DELETION OF STUDENT:  
(UPDATE DEPR  
    SET NSRUD=NSTUD - 1  
    WHERE DNO=OLD_STUDENT.DNO);
```

## 9.3 무결성 규정의 명세

- SQL에서의 무결성 규정

```
CREATE DOMAIN DYEAR SMALLINT  
CONSTRAINT CONST-YEAR  
CHECK(VALUE IN (1,2,3,4));
```

- 테이블 제약 조건

```
CREATE TABLE ENROL  
  (SNO DSNO NOT NULL,  
   CNO DSNO NOT NULL,  
   GRADE INTEGER NOT NULL,  
   PRIMARY KEY (SNO,CNO),  
   FOREIGN KEY (SNO) REFERENCES STUDENT  
     ON DELETE CASCADE  
     ON UPDATE CASCADE,  
   FOREIGN KEY (CNO) REFERENCES COURSE  
     ON DELETE CASCADE  
     ON UPDATE CASCADE,  
   CHECK(GRADE ≥ 0 AND GRADE ≤ 100) );
```



## 9.3 무결성 규정의 명세

- 일반 제약 조건

```
CREATE ASSERTION CONST_YEAR  
CHECK ((SELECT MIN(STUDENT.YEAR)  
        FROM STUDENT > 0);
```

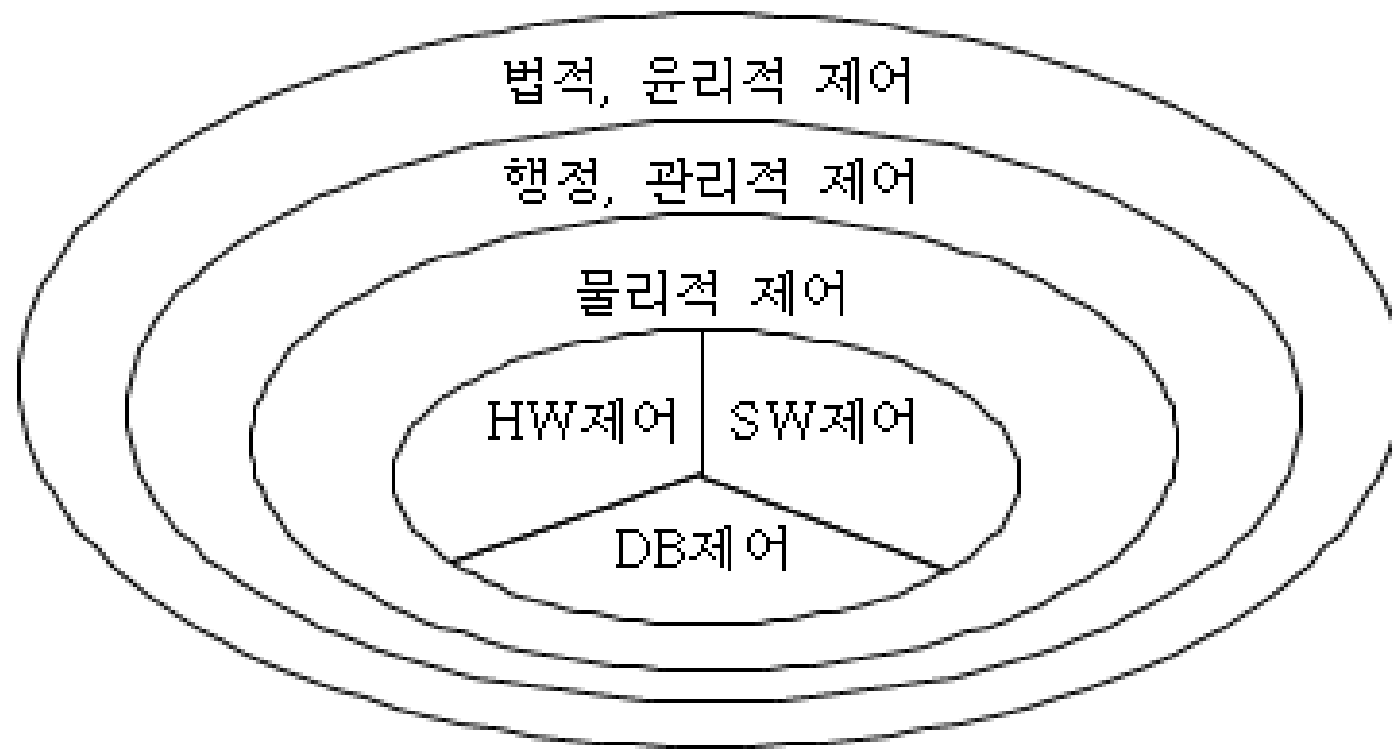
```
CREATE ASSERTION CONST_CREDIT  
CHECK (NOT EXISTS (SELECT * FROM COURSE  
                   WHERE NOT (COURSE.CREDIT < 5)));
```

```
DROP ASSERTION constraint;
```

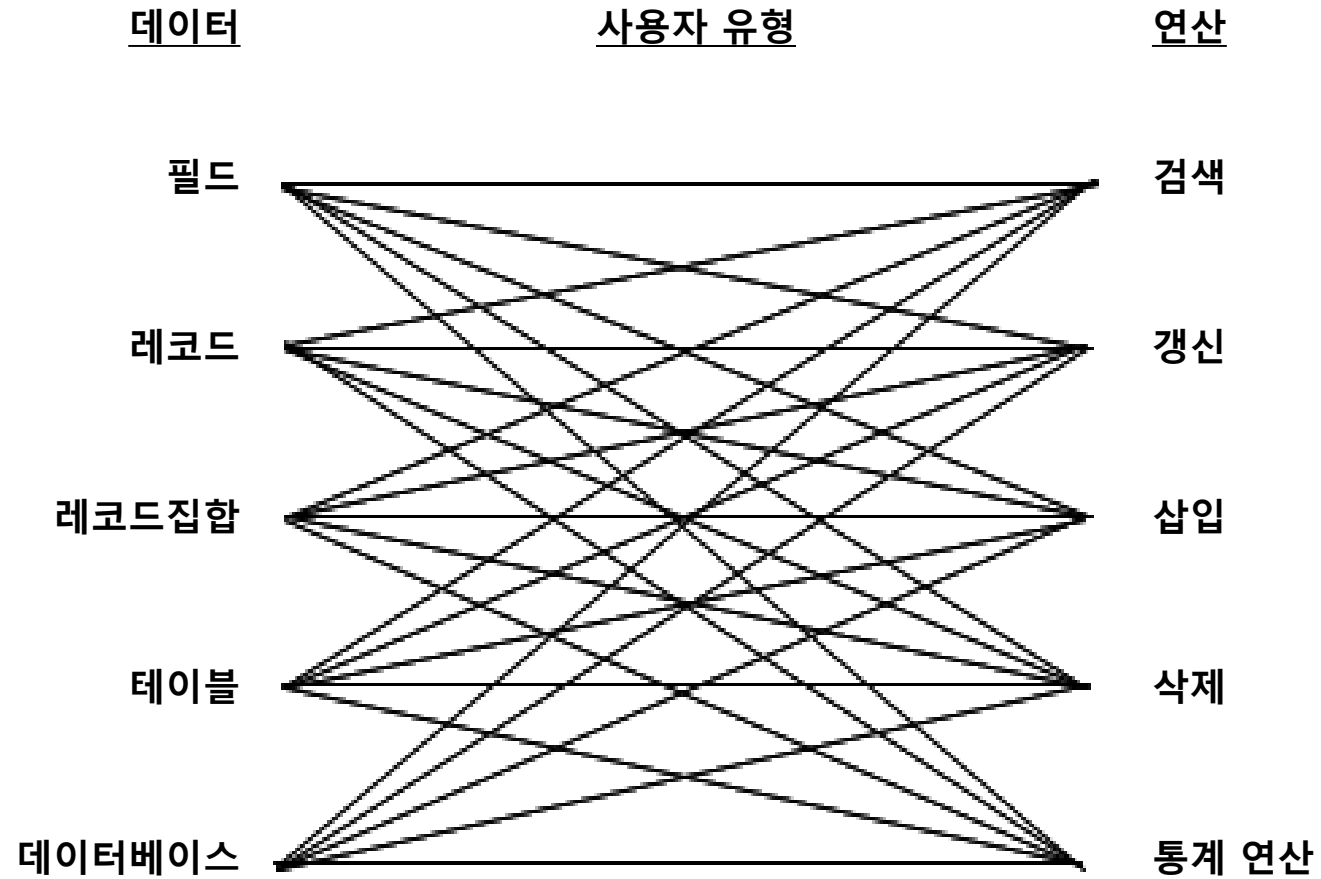
```
SET CONSTRAINTS constraint_list { DEFERRED | IMMEDIATE}
```

## 9.4 보안의 개념

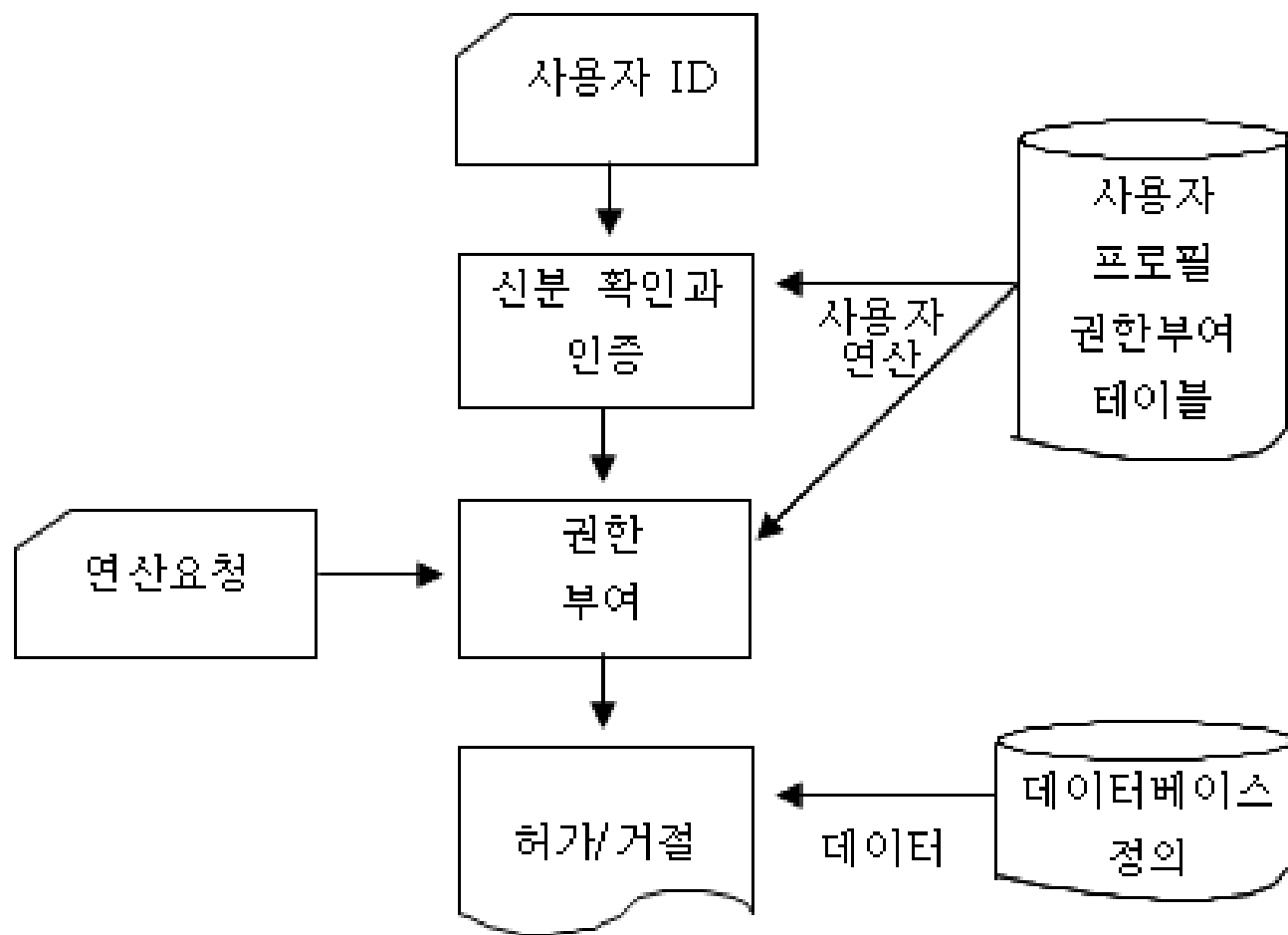
- 법적, 윤리적 통제
- 행정, 관리적 통제
- 물리적 통제
- 기술적 통제



## 9.4 보안의 개념



## 9.5 데이터베이스 보안과 구현



## 9.5 데이터베이스 보안과 구현

### ● 권한 부여 테이블

데이터 사용자	데이터 1	데이터 2	데이터 3
U1	검색	×	검색, 갱신
U2	×	검색	검색, 갱신
U3	검색	검색, 갱신	×
U4	검색, 갱신	×	×

## 9.6 권한 부여 명세 기법

- 뷰 기법

```
CREATE VIEW ST1  
  AS SELECT SNO, NAME, YEAR  
  FROM STUDENT
```

```
CREATE VIEW ST2  
  AS SELECT *  
  FROM STUDENT  
  WHERE YEAR=1
```

## 9.6 권한 부여 명세 기법

- GRANT/REVOKE

DBA : GRANT INSERT, DELETE ON STUDENT TO U1;

DBA : GRANT SELECT ON STUDENT TO U2 WITH GRANT OPTION;

U2 : GRANT SELECT ON STUDENT TO U3;

DBA : REVOKE SELECT ON STUDENT FROM U2 CASCADE;

- 강제적 접근 제어

9장을 마치며.....

질의 & 응답

