



데이터 무결성을 위한 제약조건




배재대학교 컴퓨터공학과
김 창 수

Contents

 데이터 무결성 제약조건의 개념

 데이터 무결성 제약조건의 종류

 무결성 제약조건의 생성 방법

데이터 무결성 제약조건의 개념

❖ 데이터 무결성 제약조건의 개념

- 데이터의 정확성과 일관성을 보장
- 데이터의 정확성을 유지하여 다양한 종류의 업무규칙 고려한 예
 - student 테이블에서 학년 데이터는 1,2,3,4 중의 하나의 값만 입력
 - 모든 학번은 유일
 - student 테이블의 지도교수 번호는 professor 테이블의 교수 번호중의 하나와 반드시 일치

❖ 데이터 무결성 제약조건의 장점

- 테이블 생성시 무결성 제약조건을 정의 가능
- 테이블에 대해 정의, 데이터 덱서너리에 저장되므로 응용 프로그램에서 입력된 모든 데이터에 대해 동일하게 적용
- 제약조건을 활성화, 비활성화 할 수 있는 융통성

데이터 무결성 제약조건의 종류

무결성 제약조건 종류

무결성 제약조건	설명
NOT NULL	해당 칼럼 값은 NULL을 포함할 수 없음
고유키	테이블내서 해당 칼럼 값은 항상 유일해야함
기본키	해당 칼럼 값은 반드시 존재해야 하며, 유일해야함 UNIQUE, NOT NULL 제약조건을 결합한 형태
참조	해당 칼럼 값은 참조되는 테이블의 칼럼 값 중의 하나와 일치하거나 NULL을 가짐
CHECK	해당 칼럼에 저장 가능한 데이터 값의 범위나 조건 지정

DEPTNO	DNAME	COLLEGE	LOC
101	컴퓨터공학과	100	1호관
102	멀티미디어학과	100	2호관
.....
100	정보미디어학부	10	
10	공과대학		

① 기본 키 무결성 제약조건

- STUDNO 컬럼에는 값이 반드시 존재해야 함
- STUDNO 컬럼의 모든 값은 반드시 유일해야 함

② 고유 키 무결성 제약조건

- USERID, IDNUM 컬럼의 모든 값은 반드시 유일해야 함

학생 테이블

STUDNO	NAME	USERID	GRADE	IDNUM	BIRTHDATE	TEL	HEIGHT	WEIGHT	DEPTNO	PROFNO
10101	전인하	jun123	4	7904021369824	79/04/02	051)781-2158	176	72	101	9903
20101	이동훈	Dals	1	8312101128467	83/12/10	055)426-1752	172	64	201	
.....	
10203	윤진옥	Samba7	3	7904021358671	79/04/02	053)487-2698	171	70	102	9905
10107	이광훈	huriky	4	8109131276431	81/10/13	055)736-4981	175	92	101	9903
.....	

③ NOT NULL 무결성 제약조건

- NAME 컬럼에는 값이 반드시 존재해야 함
- 컬럼 값으로 NULL을 가질 수 없음

④ CHECK 무결성 제약조건

- GRADE 컬럼 값은 반드시 1, 2, 3, 4 중 하나여야 함

⑤ 참조키 무결성 제약조건

- 학생 테이블의 DEPTNO컬럼의 값은 부서 테이블의 DEPTNO컬럼의 값 중 하나와 일치해야 함

입력 예 (1)

30101	홍길동	gdhong	1	8605051827463	86/05/05		182	78	201	
-------	-----	--------	---	---------------	----------	--	-----	----	-----	--

입력성공

입력 예 (2)

10101		guest	5	8011301898765					500	9903
-------	--	-------	---	---------------	--	--	--	--	-----	------

입력실패



NOT NULL 무결성 제약조건

- ❖ 해당 칼럼의 값이 NULL을 가질 수 없다는 것을 정의하는 제약조건
- ❖ NOT NULL 무결성 제약조건이 지정된 칼럼은 반드시 값을 입력, 생략 시 오류 발생
- ❖ 기존 데이터를 NULL로 수정하는 경우에도 오류 발생

NOT NULL 무결성 제약조건

STUDNO	NAME	USERID	GRADE	IDNUM	BIRTHDATE	TEL	HEIGHT	WEIGHT	DEPTNO	PROFNO
10101	전인하	jun123	4	7904021369824	79/04/02		176	72	101	9903
20101	이동훈		1	8312101128467		055)426-1752	172	64	201	
.....	
10203	윤진옥	Samba7	3		79/04/02	053)487-2698	171		102	9905
10107	이광훈	huriky		8109131276431	81/10/13	055)736-4981		92	101	9903
.....	

NOT NULL 무결성 제약조건 존재

- STUDNO, NAME 컬럼에는 NULL을 입력할 수 없음

NOT NULL 무결성 제약조건이 없음

- 이 컬럼들의 어느 열에든지 NULL을 입력할 수 있음
 - NULL을 의미

입력 예 (1)

30101	홍길동	gdhong	1	8605051827463	86/05/05		182	78	201	
-------	-----	--------	---	---------------	----------	--	-----	----	-----	--

입력성공

입력 예 (2)

		chsung	4	8909071928374					101	9903
--	--	--------	---	---------------	--	--	--	--	-----	------

입력실패

- STUDNO 값 과 NAME 값은 NULL을 허용하지 않음

고유 키 무결성 제약조건

- ❖ 한 테이블 내에서 칼럼이 동일한 값을 가질 수 없음을 정의하는 제약조건
- ❖ NULL 은 고유 키 제약조건에 위배되지 않으므로 입력할수 있다.
- ❖ 고유 키 무결성 제약조건이 정의 되면 테이블에 조약조건과 같은 이름의 인덱스가 자동적으로 생성

고유 키 무결성 제약조건

고유 키 무결성 제약조건 존재 : 컬럼의 모든 값이 유일해야 하므로 중복될 수 없음

STUDNO	NAME	USERID	GRADE	IDNUM	BIRTHDATE	TEL	HEIGHT	WEIGHT	DEPTNO	PROFNO
10101	전인하	jun123	4	7904021369824	79/04/02	051)781-2158	176	72	101	9903
20101	이동훈	Dals	1	8312101128467	83/12/10	055)426-1752	172	64	201	
.....	
10203	윤진옥	Samba7	3	7904021358671	79/04/02	053)487-2698	171	70	102	9905
10107	이광훈	huriky	4	8109131276431	81/10/13	055)736-4981	175	92	101	9903
.....	

고유 키 무결성 제약조건이 없음

- GRADE(학년) 컬럼의 값은 중복될 수 있음

입력 예 (1)

30101	홍길동	gdhong	1	8605051827463	86/05/05		182	78	201	
-------	-----	--------	---	---------------	----------	--	-----	----	-----	--

입력성공

입력 예 (2)

30102	성춘향	chsung	2				1752	182	78	201
-------	-----	--------	---	--	--	--	------	-----	----	-----

입력성공

- 고유키 무결성은 NULL 값을 허용하므로 입력 성공

입력 예 (3)

30103	이동훈	jun123	4	8312101128467					101	9903
-------	-----	--------	---	---------------	--	--	--	--	-----	------

입력실패

- USERID 값 'jun123' 과 IDNUM 값 '8312101128467' 값이 이미 존재함

기본 키 무결성 제약조건

- ❖ 기본 키는 테이블내에 저장된 행을 유일하게 식별할 수 있는 식별자로 사용
- ❖ 관계형 데이터 모델에서 테이블은 행을 원소로 가지는 집합으로 정의되어 집합내의 모든 행을 구별할 수 있는 식별자가 필요
- ❖ 기본 키 무결성 제약조건은 하나 이상의 칼럼에 의해 테이블의 모든 행을 구별하기 위한 식별자를 정의하기 위한 제약 조건
- ❖ 고유 키 제약조건과 NOT NULL 제약조건을 결합한 개념

기본 키 무결성 제약조건

기본키 무결성 제약조건 존재 : 컬럼의 모든 값이 유일하고, NULL을 허용하지 않음

STUDNO	NAME	USERID	GRADE	IDNUM	BIRTHDATE	TEL	HEIGHT	WEIGHT	DEPTNO	PROFNO
10101	전인하	jun123	4	7904021369824	79/04/02	051)781-2158	176	72	101	9903
20101	이동훈	Dals	1	8312101128467	83/12/10	055)426-1752	172	64	201	
.....	
10203	윤진옥	Samba7	3	7904021358671	79/04/02	053)487-2698	171	70	102	9905
10107	이광훈	huriky	4	8109131276431	81/10/13	055)736-4981	175	92	101	9903
.....	

입력 예 (1)

30101	홍길동	gdhong	1	8605051827463	86/05/05		182	78	201	
-------	-----	--------	---	---------------	----------	--	-----	----	-----	--

입력성공

입력 예 (2)

	성춘향	chsung	2		86/05/05		182	78	201	
--	-----	--------	---	--	----------	--	-----	----	-----	--

입력실패

- 기본 키 무결성 제약조건은 NULL을 허용하지 않음

입력 예 (3)

10101		jun123	4	8312101128467					101	9903
-------	--	--------	---	---------------	--	--	--	--	-----	------

입력실패

- 기본 키 무결성 제약조건은 중복을 허용하지 않음, STUDNO 의 값 10101 은 이미 존재함

참조 무결성 제약조건

- 한 테이블의 칼럼 값이 자신, 다른 테이블의 칼럼 값 중에 하나와 일치시키기 위한 제약조건
- 용어
 - 자식 테이블 : 다른 테이블의 칼럼 값을 참조하는 테이블
 - 부모 테이블 : 다른 테이블에 의해 참조되는 테이블
 - 외래 키 : 부모 테이블의 칼럼 값을 참조하는 자식 테이블의 칼럼
 - 참조 키 : 자식 테이블에서 참조하는 부모 테이블의 칼럼

DML 명령문	부모 테이블	자식 테이블
INSERT	참조 키 값이 고유한 경우에만 가능	외래 키 값이 참조 키 값 중의 하나와 일치하거나 null인 경우에 가능
UPDATE	참조 키 값을 참조하는 자식 테이블의 칼럼 값이 없는 경우에만 가능	수정되는 외래 키 값이 참조 키 값 중의 하나와 일치할 경우에 가능
DELETE RESTRICT	참조 키 값을 참조하는 자식 테이블의 칼럼 값이 없는 경우에만 가능	항상 가능
DELETE CASCADE	항상 가능	항상 가능

참조 키
(referenced key)

DEPTNO	DNAME	COLLEGE	LOC
101	컴퓨터공학과	100	1호관
102	멀티미디어학과	100	2호관
.....
100	정보미디어학부	10	
10	공과대학		

학생 테이블

STUDNO	NAME	USERID	GRADE	IDNUM	BIRTHDATE	TEL	HEIGHT	WEIGHT	DEPTNO	PROFNO
10101	전인하	jun123	4	7904021369824	79/04/02	051)781-2158	176	72	101	9903
20101	이동훈	Dals	1	8312101128467	83/12/10	055)426-1752	172	64	201	
.....	
10203	윤진옥	Samba7	3	7904021358671	79/04/02	053)487-2698	171	70	102	9905
10107	이광훈	huriky	4	8109131276431	81/10/13	055)736-4981	175	92	101	9903
.....	

외래 키 (foreign key)

입력 예 (1)

30101	홍길동	gdhong	1	86050101123456789		055)426-1752	182	78	201	
-------	-----	--------	---	-------------------	--	--------------	-----	----	-----	--

입력성공

입력 예 (2)

30102	성춘향	chsung	2	86050101123456789			182	78		
-------	-----	--------	---	-------------------	--	--	-----	----	--	--

입력성공

- 참조 무결성 제약조건은 NULL을 허용하므로 입력 성공

입력 예 (3)

30103	이동훈	guest	4	731010112777					500	9903
-------	-----	-------	---	--------------	--	--	--	--	-----	------

입력실패

- DEPTNO의 값 500 은 부서 테이블에 존재하지 않음

CHECK 무결성 제약조건

- ❖ 컬럼에서 허용 가능한 데이터의 범위나 조건을 지정
- ❖ 데이터 입력이나 수정 시 실수로 부정확한 값을 입력 예방
- ❖ 하나의 컬럼에 여러 개의 CHECK 무결성 제약조건을 지정 가능
- ❖ CURRVAL, NEXTVAL과 같은 가상 컬럼이나 SYSDATE, USER와 같은 함수는 사용 불가능
- ❖ 예
 - student 테이블에서 학년 데이터는 1,2,3,4 중의 하나의 값만 입력 가능

CHECK 무결성 제약조건

CHECK 무결성 제약조건 존재 : 지정한 조건에 부합되는 값만 허용



STUDNO	NAME	USERID	GRADE	IDNUM	BIRTHDATE	TEL	HEIGHT	WEIGHT	DEPTNO	PROFNO
10101	전인하	jun123	4	7904021369824	79/04/02	051)781-2158	176	72	101	9903
20101	이동훈	Dals	1	8312101128467	83/12/10	055)426-1752	172	64	201	
.....	
10203	윤진옥	Samba7	3	7904021358671	79/04/02	053)487-2698	171	70	102	9905
10107	이광훈	huriky	4	8109131276431	81/10/13	055)736-4981	175	92	101	9903
.....	

입력 예 (1)

30101	홍길동	gdhong	1	8605051827463	86/05/05	051)781-2158	182	78	201	
-------	-----	--------	---	---------------	----------	--------------	-----	----	-----	--

입력성공

입력 예 (2)

30102	성춘향	chsung			86/05/05	051)781-2158	78	201		
-------	-----	--------	--	--	----------	--------------	----	-----	--	--

입력성공

- GRADE는 check 무결성 제약조건만 가지므로 NULL은 허용 가능

입력 예 (3)

30103	이동훈	mrlee	6		86/05/05	051)781-2158	78	201		
-------	-----	-------	---	--	----------	--------------	----	-----	--	--

입력실패

- GRADE의 값은 1, 2, 3, 4 만 허용되므로 6은 조건에 부합하지 않음

무결성 제약조건 생성 방법

❖ 무결성 제약조건 생성

- 테이블 생성과 동시에 정의
- 테이블을 생성한 이후에 추가, 삭제 할수 있음
- 제약조건명을 지정하지 않으면 SYS_Cn 형태로 자동생성
- 칼럼 레벨 : 칼럼 정의시 해당 칼럼별로 지정
- 테이블 레벨 : 제약조건이 하나 이상의 칼럼을 참조, 지정

❖ 사용법

```
CREATE TABLE [schema.] table  
(column datatype[DEFAULT expression]  
[column_constraint],  
[table_constraint][,...]);
```


칼럼 레벨에서 제약조건 정의

❖ column_constraint : 칼럼 레벨에서 제약 조건을 정의하는 경우

```
column_constraint ::=  
{ [NOT] NULL  
| PRIMARY KEY [USING INDEX index_clause]  
| UNIQUE [USING INDEX index_clause]  
| REFERENCES [schema.] table [ (column [.column] ...) ]  
    [ON DELETE CASCADE]  
| CHECK (condition) }  
defer_spec ::=  
[NOT DEFERRABLE  
| DEFERRABLE [INITIALLY {IMMEDIATE | DEFERRED}]]  
[DISABLE | ENABLE [VALIDATE | NOVALIDATE]]
```

테이블 레벨에서 제약조건 정의

❖ table_constraint : 테이블 레벨에서 제약 조건을 정의하는 경우

```
table_constraint ::=  
{ {UNIQUE | PRIMARY KEY} (column[,column]...)   
| PRIMARY KEY[USING INDEX index_clause]   
| UNIQUE[USING INDEX index_clause]   
| FOREIGN KEY (column[.column]...)   
      REFERENCES[schema.] table[(column[.column]...)]   
      [ON DELETE CASCADE]   
| CHECK (condition) }  
defer_spec ::=   
[NOT DEFERRABLE   
| DEFERRABLE[INITIALLY{IMMEDIATE | DEFERRED}]]   
[DISABLE | ENABLE]
```

무결성 제약조건 생성문에서의 키워드

❖ ON DELETE CASCADE

- 부모 테이블에서 외래 키가 참조하는 기본 키나 고유 키를 포함한 행을 삭제할 경우 자식 테이블의 외래 키를 포함하는 행도 함께 삭제

❖ USING INDEX

- 기본 키나 고유 키 무결성 제약조건 생성시 묵시적으로 생성되는 인덱스에 대한 스토리지 파라미터 정의

❖ NOT DEFERRABLE :

- 하나의 DML 명령문이 처리될 때마다 제약조건 위반 여부를 검사, defer_spec을 명시하지 않을 경우에는 기본 값

❖ DEFERRABLE

- 트랜잭션내의 모든 DML 명령문에 대한 제약조건 검사를 트랜잭션 종료 시 까지 연기

❖ INITIALLY IMMEDIATE

- DML 명령문이 종료될 때마다 제약조건 검사

❖ INITIALLY DEFERRED

- 트랜잭션이 끝날 때만 제약조건 검사

무결성 제약조건 생성 예

강좌(subject) 테이블 인스턴스

칼럼이름	데이터 타입	Key Type	NN/ Unique	FK table	FK column	설명
SUBNO	NUMBER(5)	PK	NN,U			강좌번호
SUBNAME	VARCHAR2(20)		NN			강좌이름
TERM	VARCHAR2(1)					학기
TYPE	VARCHAR2(5)					필수/선택구분

```

SQL> CREATE TABLE subject
2  (subno  NUMBER(5)
3     CONSTRAINT subject_no_pk PRIMARY KEY
4     DEFERRABLE INITIALLY DEFERRED
5     USING INDEX TABLESPACE indx,
6  subname VARCHAR2(20)
7     CONSTRAINT subject_name_nn NOT NULL,
8  term    VARCHAR2(1)
9     CONSTRAINT subject_term_ck CHECK (term in ('1', '2')),
10 type    VARCHAR2(1));
    
```

인덱스 테이블과 다른 테이블스페이스에 저장하는 것이 성능향상을 위해 효율적이다.

칼럼레벨 제약조건들

테이블이 생성되었습니다.

무결성 제약조건 생성 예

수강(sugang) 테이블 인스턴스

칼럼이름	데이터 타입	Key Type	NN/ Unique	FK table	FK column	설명
STUDNO	NUMBER(5)	PK1,FK	NN,U	student	studno	학번
SUBNO	NUMBER(5)	PK2,FK	NN,U	subject	subno	강좌 번호
REGDATE	DATE					등록일
RESULT	NUMBER(3)					평가 결과

```
SQL> ALTER TABLE student
2 ADD CONSTRAINT stud_no_pk PRIMARY KEY(studno);
```

studno 칼럼에 기본키 제약조건 추가

테이블이 변경되었습니다.

```
SQL> CREATE TABLE sugang
2 (studno NUMBER(5)
3 CONSTRAINT sugang_studno_fk REFERENCES student(studno),
4 subno NUMBER(5)
5 CONSTRAINT sugang_subno_fk REFERENCES subject(subno),
6 regdate DATE,
7 result NUMBER(3),
8 CONSTRAINT sugang_pk PRIMARY KEY(studno, subno));
```

칼럼레벨 제약조건들

테이블레벨 제약조건

무결성 제약조건 조회

❖ USER_CONSTRAINTS 데이터 사전에서 무결성 제약조건 조회

```
SQL> SELECT  constraint_name, constraint_type
2  FROM    user_constraints
3  WHERE   table_name IN ('SUBJECT', 'SUGANG');
```

CONSTRAINT_NAME	C

SUBJECT_NAME_NN	C
SUBJECT_TERM_CK	C
SUBJECT_NO_PK	P
SUGANG_PK	P
SUGANG_STUDNO_FK	R
SUGANG_SUBNO_FK	R

- C : check 또는 not null
- P : primary key
- U : unique key
- R : foreign key

6 개의 행이 선택되었습니다.

기존 테이블에 무결성 제약조건 추가

❖ NULL을 제외한 무결성 제약조건 추가

- ALTER ... ADD CONSTRAINT 명령문 사용

❖ 사용법

```
ALTER TABLE table  
ADD [CONSTRAINT constraint_name] constraint_type (column);
```

❖ NULL 무결성 제약 조건 추가

- ALTER TABLE...MODIFY 명령문 사용

❖ 사용법

```
ALTER TABLE table  
MODIFY (column CONSTRAINT constraint_name constraint_type);
```

무결성 제약조건 추가 예

❖ 사용 예

- 학생 테이블 인스턴스를 참조하여 studno에 기본 키, idnum에 고유 키, name에 NOT NULL 무결성 제약조건을 추가하여라.

```
SQL> ALTER TABLE student
2 ADD CONSTRAINT stud_idnum_uk UNIQUE(idnum);
```

idnum 칼럼에 고유키
제약조건 추가

테이블이 변경되었습니다.

```
SQL> ALTER TABLE student
2 MODIFY (name CONSTRAINT stud_name_nn NOT NULL);
```

name 칼럼에 NOT
NULL 제약조건 추가

테이블이 변경되었습니다.

```
SQL> ALTER TABLE student ADD CONSTRAINT stud_deptno_fk
2 FOREIGN KEY(deptno) REFERENCES department(deptno);
FOREIGN KEY(deptno) REFERENCES department(deptno)
*
```

department 테이블의 deptno 칼
럼에 기본키 또는 고유키 제약조
건이 없으면 외래키 정의시 오류

2행에 오류:

ORA-02270: 이 열목록에 대한 유일 혹은 일차 키가 일치하지 않습니다

무결성 제약조건 추가 예

❖ 실습 예1

- 부서 테이블 인스턴스를 참조하여 deptno에 기본 키, dname에 NOT NULL 무결성 제약조건을 추가하여라.

❖ 실습 예2

- 교수 테이블 인스턴스를 참조하여 profno에 기본 키, name에 NOT NULL, deptno에 참조 무결성 제약조건을 추가하여라.

무결성 제약조건에 의한 DML 명령문의 영향

❖ 즉시 제약조건(immediate constraints)에 위배되는 데이터 입력시

- 테이블에 데이터를 먼저 입력한 다음 무결성 제약조건을 위반하는 명령문을 롤백

❖ 지연 제약조건(deferred constraints)에 위배되는 데이터 입력시

- 트랜잭션내의 DML 명령문에서 제약조건 검사를 COMMIT 시점에서 한꺼번에 처리하여 트랜잭션의 처리 성능을 향상시키기 위해 사용

즉시 제약조건 위반되는 데이터 입력 1

❖ 사용 예

- 강좌 테이블에서 무결성 제약조건에 위반되는 데이터의 예를 입력 하려라.

```
SQL> INSERT INTO subject VALUES(1, 'SQL', '1', '필수');
```

1 개의 행이 만들어졌습니다.

```
SQL> INSERT INTO subject VALUES(2, '', '2', '필수');
```

```
INSERT INTO subject VALUES(2, '', '2', '필수');
```

*

1행에 오류:

ORA-01400: NULL을 ("SCOTT"."SUBJECT"."SUBNAME") 안에 삽입할 수 없습니다

name에 NULL 입력

```
SQL> INSERT INTO subject VALUES(3, 'JAVA', '3', '선택');
```

```
INSERT INTO subject VALUES(3, 'JAVA', '3', '선택');
```

*

1행에 오류:

ORA-02290: 체크 제약조건 (SCOTT.SUBJECT_TERM_CHK)이 위반되었습니다

1이나 2이외의 값은 입력할 수 없다

```
SQL> COMMIT;
```

커밋이 완료되었습니다.

```
SQL> SELECT * FROM subject;
```

SUBNO	SUBNAME	T	TYPE
1	SQL	1	필수

DML문 실행시마다 무결성 제약조건 위반 여부를 확인하여 위반되면 해당 DML 명령문을 롤백시킨다

지연 제약조건 위반되는 데이터 입력

❖ 사용 예

- 강좌 테이블의 무결성 제약조건(subject_id_pk)에 위반되는 데이터를 입력하여 오류 메시지를 확인하여라.

```
SQL> INSERT INTO subject VALUES(4, '데이터베이스', '1', '필수');
```

1 개의 행이 만들어졌습니다.

subno에 중복값 입력

```
SQL> INSERT INTO subject VALUES(4, '데이터모델링', '2', '선택');
```

1 개의 행이 만들어졌습니다.

```
SQL> COMMIT;
```

```
COMMIT
```

```
*
```

1행에 오류:

ORA-02091: 트랜잭션이 롤백되었습니다

ORA-00001: 무결성 제약 조건 (SCOTT.SUBJECT_NO_PK)에 위반됩니다

제약조건이 deferrable initially deferred로 지정된 경우에는 DML문 실행시점에는 무결성 제약조건 위반 여부를 확인하지 않고 트랜잭션 종료시 확인하여 위반되면 해당 트랜잭션을 롤백시킨다

```
SQL> SELECT * FROM subject;
```

SUBNO	SUBNAME	T	TYPE
1	SQL	1	필수

무결성 제약조건 삭제

❖ 무결성 제약조건 삭제시

- ALTER TABLE ... DROP CONSTRAINT 명령문 사용
- 부모 테이블의 기본키 무결성 제약조건을 삭제하는 경우
 - 참조 무결성 제약조건을 먼저 삭제한 후 삭제하거나 CASCADE 옵션 사용

❖ 사용법

```
ALTER TABLE table  
DROP CONSTRAINT constraint_name [CASCADE]);
```

- cascade : 삭제되는 칼럼을 참조하는 참조 무결성 제약조건도 함께 삭제

무결성 제약조건 삭제 예

❖ 사용 예

- 강좌 테이블의 subject_pk_ck 무결성 제약조건을 삭제하여라.

```
SQL> SELECT constraint_name, constraint_type  
2 FROM user_constraints  
3 WHERE table_name = 'SUBJECT';
```

CONSTRAINT_NAME	C

SUBJECT_NAME_NN	C
SUBJECT_TERM CK	C
SUBJECT_NO_PK	P

```
SQL> ALTER TABLE subject  
2 DROP CONSTRAINT subject_term_ck;
```

테이블이 변경되었습니다.

```
SQL> SELECT constraint_name, constraint_type  
2 FROM user_constraints  
3 WHERE table_name = 'SUBJECT';
```

CONSTRAINT_NAME	C

SUBJECT_NAME_NN	C
SUBJECT_NO_PK	P

무결성 제약조건 활성화 및 비활성화

❖ 대용량 데이터 초기 입력시

- 무결성 제약조건 위반 여부를 검사하는 과정으로 인해 처리 시간이 오래 걸리는 경우가 발생
- 이러한 경우에는 기존의 무결성 제약조건을 일시적으로 비활성화하여 데이터를 입력한 다음, 비활성화된 무결성 제약조건은 다시 활성화
- ALTER TABLE 명령문에서 ENABLE 또는 DISABLE 절 사용

무결성 제약조건의 비활성화

❖ 제약조건의 비활성화

```
ALTER TABLE table
DISABLE CONSTRAINT constraint_name [CASCADE]);
```

❖ 비활성화 사용 예

- 수강 테이블의 sugang_pk, sugang_studno_라 무결성 제약조건을 비활성화 하여라.

```
SQL> ALTER TABLE sugang
2 DISABLE CONSTRAINT sugang_Pk;
```

테이블이 변경되었습니다.

```
SQL> ALTER TABLE sugang
2 DISABLE CONSTRAINT sugang_studno_fk;
```

테이블이 변경되었습니다.

```
SQL> SELECT constraint_name, status
2 FROM user_constraints
3 WHERE table_name IN ('SUGANG', 'SUBJECT');
```

CONSTRAINT_NAME	STATUS
SUBJECT_NAME_NN	ENABLED
SUBJECT_NO_PK	ENABLED
SUGANG_PK	DISABLED
SUGANG_STUDNO_FK	DISABLED
SUGANG_SUBNO_FK	ENABLED

무결성 제약조건의 활성화

❖ 제약조건의 활성화

```
ALTER TABLE table  
ENABLE [NOVALIDATE] CONSTRAINT constraint_name [CASCADE]);
```

- NOVALIDATE : 기존 데이터에 대해서는 제약조건을 적용하지 않고, 새로 입력되는 데이터나 수정하는 데이터에 대해서만 제약조건을 검사하기 위한 옵션

무결성 제약조건의 활성화

❖ 활성화 사용법

- 수강 테이블의 sugang_pk, dugang_studno_fk 무결성 제약조건을 활성화 하여라.

```
SQL> ALTER TABLE sugang
2  ENABLE CONSTRAINT sugang_Pk;
```

테이블이 변경되었습니다.

```
SQL> ALTER TABLE sugang
2  ENABLE CONSTRAINT sugang_studno_fk;
```

테이블이 변경되었습니다.

```
SQL> SELECT constraint_name, status
2  FROM user_constraints
3  WHERE table_name = 'SUGANG';
```

CONSTRAINT_NAME	STATUS
SUGANG_PK	ENABLED
SUGANG_STUDNO_FK	ENABLED
SUGANG_SUBNO_FK	ENABLED

무결성 제약조건 조회 1

❖ USER_CONSTRAINTS

- 제약조건이 설정된 테이블 이름, 무결성 제약조건 이름, 무결성 제약조건 종류 및 활성화 상태정보를 저장

❖ 사용 예

- student, professor, department 테이블에 정의된 모든 종류의 무결성 제약조건을 조회하여라.

```
SQL> SELECT table_name, constraint_name, constraint_type, status
2 FROM user_constraints
3 WHERE table_name IN ('STUDENT', 'PROFESSOR', 'DEPARTMENT');
```

TABLE_NAME	CONSTRAINT_NAME	C STATUS
DEPARTMENT	DEPT_NO_PK	P ENABLED
DEPARTMENT	DEPT_NAME_NN	C ENABLED
PROFESSOR	PROF_NAME_NN	C ENABLED
PROFESSOR	PROF_DEPTNO_FK	R ENABLED
STUDENT	STUD_NO_PK	P ENABLED
STUDENT	STUD_IDNUM_UK	U ENABLED
STUDENT	STUD_NAME_NN	C ENABLED
STUDENT	STUD_DEPTNO_FK	R ENABLED

무결성 제약조건 조회 2

❖ USER_CONS_COLUMNS

- 제약조건이 설정된 칼럼 이름을 저장