# 제약 조건

- 데이터의 무결성을 지키기 위해 제한하는 조건

- 데이터의 무결성이란 ‘데이터에 결함이 없음’이란 의미

- 예로 네이버 회원의 아이디가 중복되면 어떤 일이 일어날까요? 이메일, 블로그, 쇼핑 기록 등 상당한 혼란이 일어날 것입니다 이런 것이 바로 데이터의 결함이고, 이런 결함이 없는 것을 데이터의 무결성이라고 표현함

**1. 기본 키(PRIMARY KEY)**

- 데이터를 구분할 수 있는 식별자

- 기본 키로 지정하면 클러스터형 인덱스가 자동으로 생성됨

- 기본 키는 1개만 가질 수 있음

1번째 방법)

CREATE TABLE member

( mem\_id CHAR(8) NOT NULL PRIMARY KEY,

mem\_name VARCHAR(10) NOT NULL,

height TINYINT UNSIGNED NULL

);

2번째 방법)

CREATE TABLE member

( mem\_id CHAR(8) NOT NULL,

mem\_name VARCHAR(10) NOT NULL,

height TINYINT UNSIGNED NULL,

PRIMARY KEY (mem\_id)

);

3번째 방법)

CREATE TABLE member

( mem\_id CHAR(8) NOT NULL,

mem\_name VARCHAR(10) NOT NULL,

height TINYINT UNSIGNED NULL

);

ALTER TABLE member

ADD CONSTRAINT

PRIMARY KEY (mem\_id);

**2. 외래 키(FOREIGN KEY)**

- 두 테이블 사이의 관계를 연결해주고, 그 결과 데이터의 무결성을 보장해주는 역할

- 외래 키가 설정된 열은 꼭 다른 테이블의 기본 키와 연결됨

- 예시로 사용하고 있는 테이블로 설명해보면 기본 키가 있는 회원 테이블을 기준 테이블이라고 부르며, 외래 키가 있는 구매 테이블을 참조 테이블이라고 함

1번째 방법)

CREATE TABLE buy

( num INT AUTO\_INCREMENT NOT NULL PRIMARY KEY,

mem\_id CHAR(8) NOT NULL,

prod\_name CHAR(6) NOT NULL,

FOREIGN KEY(mem\_id) REFERENCES member(mem\_id)

);

2번째 방법) 이름을 변경해도 괜찮지만 보통 열의 이름과 같은 이름으로 외래 키를 지정한다

CREATE TABLE buy

( num INT AUTO\_INCREMENT NOT NULL PRIMARY KEY,

user\_id CHAR(8) NOT NULL,

prod\_name CHAR(6) NOT NULL,

FOREIGN KEY(user\_id) REFERENCES member(mem\_id)

);

3번째 방법)

CREATE TABLE buy

( num INT AUTO\_INCREMENT NOT NULL PRIMARY KEY,

mem\_id CHAR(8) NOT NULL,

prod\_name CHAR(6) NOT NULL

);

ALTER TABLE buy

ADD CONSTRAINT

FOREIGN KEY(mem\_id) REFERENCES member(mem\_id);

※ 기준 테이블의 열에 변경이 생긴다면? -> 오류 발생

UPDATE member SET mem\_id = 'PINK' WHERE mem\_id='BLK'; (이름이 변경되지 않고 에러)

DELETE FROM member WHERE mem\_id='BLK'; (삭제가 되지 않고 에러)

-----> 꼭 변경하고 싶다면 ON UPDATE CASCADE/ ON DELETE CASCADE문 사용

DROP TABLE IF EXISTS buy;

CREATE TABLE buy

( num INT AUTO\_INCREMENT NOT NULL PRIMARY KEY,

mem\_id CHAR(8) NOT NULL,

prod\_name CHAR(6) NOT NULL

);

ALTER TABLE buy

ADD CONSTRAINT

FOREIGN KEY(mem\_id) REFERENCES member(mem\_id)

ON UPDATE CASCADE

ON DELETE CASCADE;

INSERT INTO buy VALUES(NULL, 'BLK', '지갑');

INSERT INTO buy VALUES(NULL, 'BLK', '맥북');

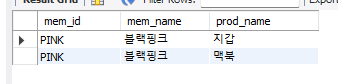
UPDATE member SET mem\_id = 'PINK' WHERE mem\_id='BLK';

SELECT M.mem\_id, M.mem\_name, B.prod\_name

FROM buy B

INNER JOIN member M

ON B.mem\_id = M.mem\_id;



**3. 고유 키(UNIQUE)**

- ‘중복되지 않는 유일한 값’을 입력해야 하는 조건

- 기본 키 제약조건과 거의 비슷하지만, 차이점은 고유 키 제약조건은 NULL 값을 허용한다는 것

- 기본 키는 테이블에 1개만 설정해야 하지만, 고유 키는 여러 개를 설정해도 됨

Ex)

DROP TABLE IF EXISTS buy, member;

CREATE TABLE member

( mem\_id CHAR(8) NOT NULL PRIMARY KEY,

mem\_name VARCHAR(10) NOT NULL,

height TINYINT UNSIGNED NULL,

email CHAR(30) NULL UNIQUE

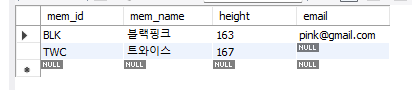
);

INSERT INTO member VALUES('BLK', '블랙핑크', 163, 'pink@gmail.com');

INSERT INTO member VALUES('TWC', '트와이스', 167, NULL);

INSERT INTO member VALUES('APN', '에이핑크', 164, 'pink@gmail.com');

SELECT \* FROM member;



----> NULL값은 허용하지만 중복 값은 허용하지 않으니 INSERT 세번째 부분은 오류로 삽입X

**4. 체크**

- 입력되는 데이터를 점검하는 기능

Ex)

DROP TABLE IF EXISTS member;

CREATE TABLE member

( mem\_id CHAR(8) NOT NULL PRIMARY KEY,

mem\_name VARCHAR(10) NOT NULL,

height TINYINT UNSIGNED NULL CHECK (height >= 100),

phone1 CHAR(3) NULL

);

INSERT INTO member VALUES('BLK', '블랙핑크', 163, NULL);

INSERT INTO member VALUES('TWC', '트와이스', 99, NULL); // 키가 100이상만 입력 가능하므로 에러

ALTER TABLE member

ADD CONSTRAINT

CHECK (phone1 IN ('02', '031', '032', '054', '055', '061' )) ; // 국번이 02, 031, 032 등 여기 적혀 있는 것들만 가능하게 설정

INSERT INTO member VALUES('TWC', '트와이스', 167, '02');

INSERT INTO member VALUES('OMY', '오마이걸', 167, '010'); // 저 목록 중에 010은 없으므로 에러

**5. 기본 값(DEFAULT)**

- 값을 입력하지 않았을 때 자동으로 입력될 값을 미리 지정해 놓는 방법

Ex)

DROP TABLE IF EXISTS member;

CREATE TABLE member

( mem\_id CHAR(8) NOT NULL PRIMARY KEY,

mem\_name VARCHAR(10) NOT NULL,

height TINYINT UNSIGNED NULL DEFAULT 160, // 기본 값을 160으로 지정

phone1 CHAR(3) NULL

);

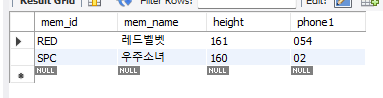
ALTER TABLE member

ALTER COLUMN phone1 SET DEFAULT '02'; // 번호의 기본 값을 02로 지정

INSERT INTO member VALUES('RED', '레드벨벳', 161, '054');

INSERT INTO member VALUES('SPC', '우주소녀', default, default);

SELECT \* FROM member;



**6. 널 값 허용**

- 널 값을 허용하려면 생략하거나 NULL을 사용하고, 허용하지 않으려면 NOT NULL을 사용