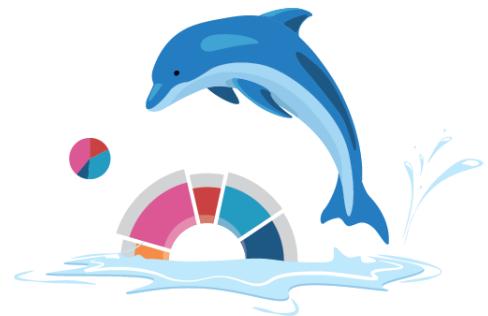


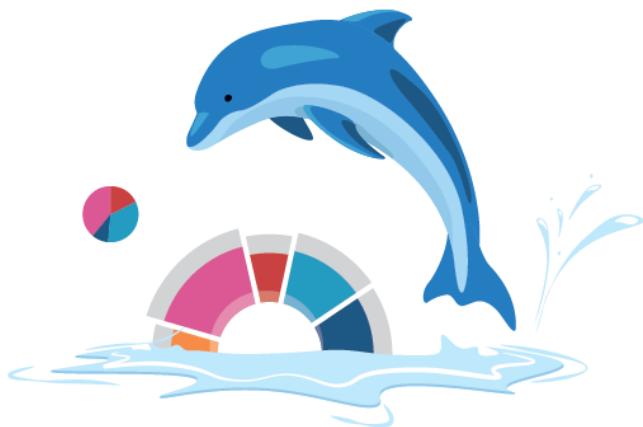


MySQL



Chapter 07

DML: 데이터 조작어



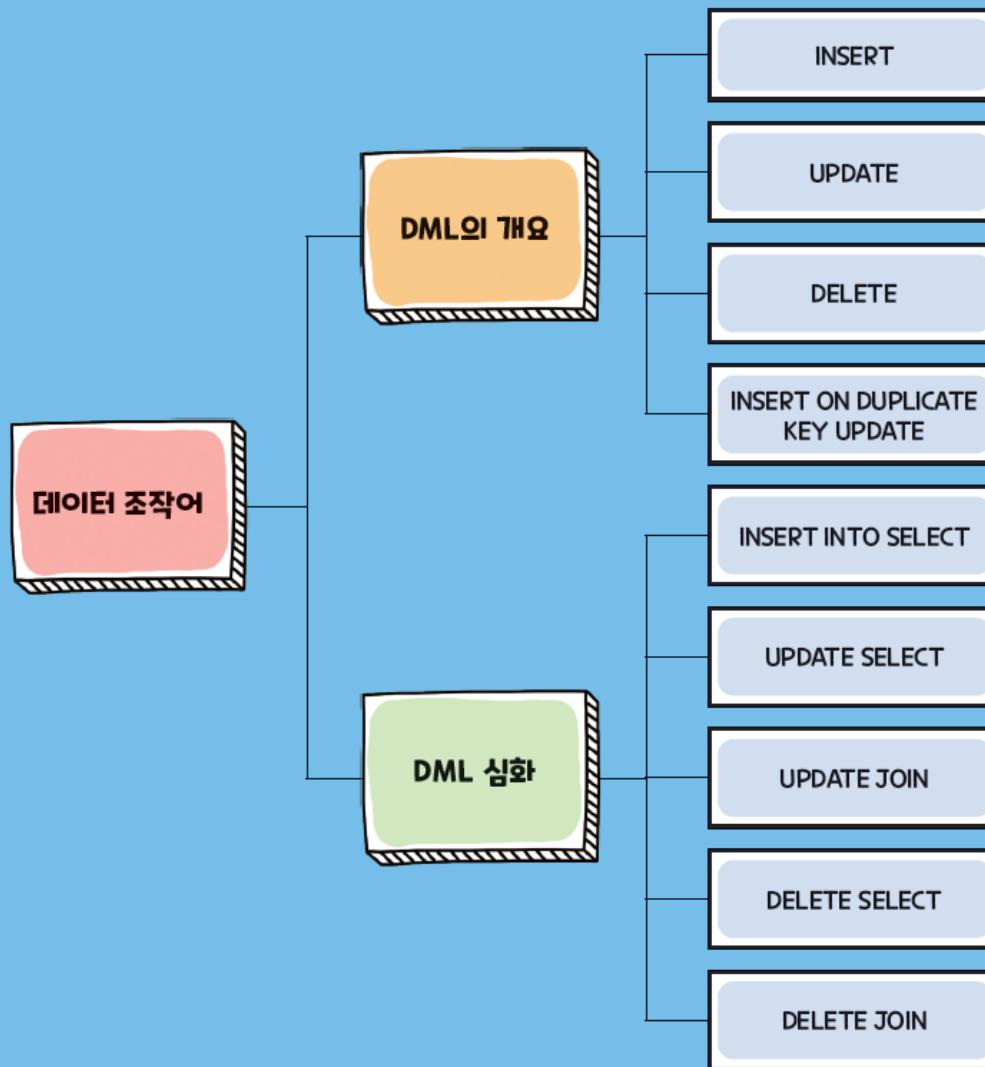
목차

1. DML의 개요
2. DML 심화

학습목표

- 데이터 조작어를 사용하여 SQL문을 작성할 수 있습니다.
- 서브쿼리를 활용한 데이터 조작어 문장을 작성할 수 있습니다.

Preview



Section 01

DML의 개요

1. INSERT

■ 데이터 조작어(Data Manipulation Language, DML)

- 데이터를 관리하는 데 사용하는 언어
- 데이터를 추가(INSERT), 변경(UPDATE), 삭제(DELETE)할 때 사용함

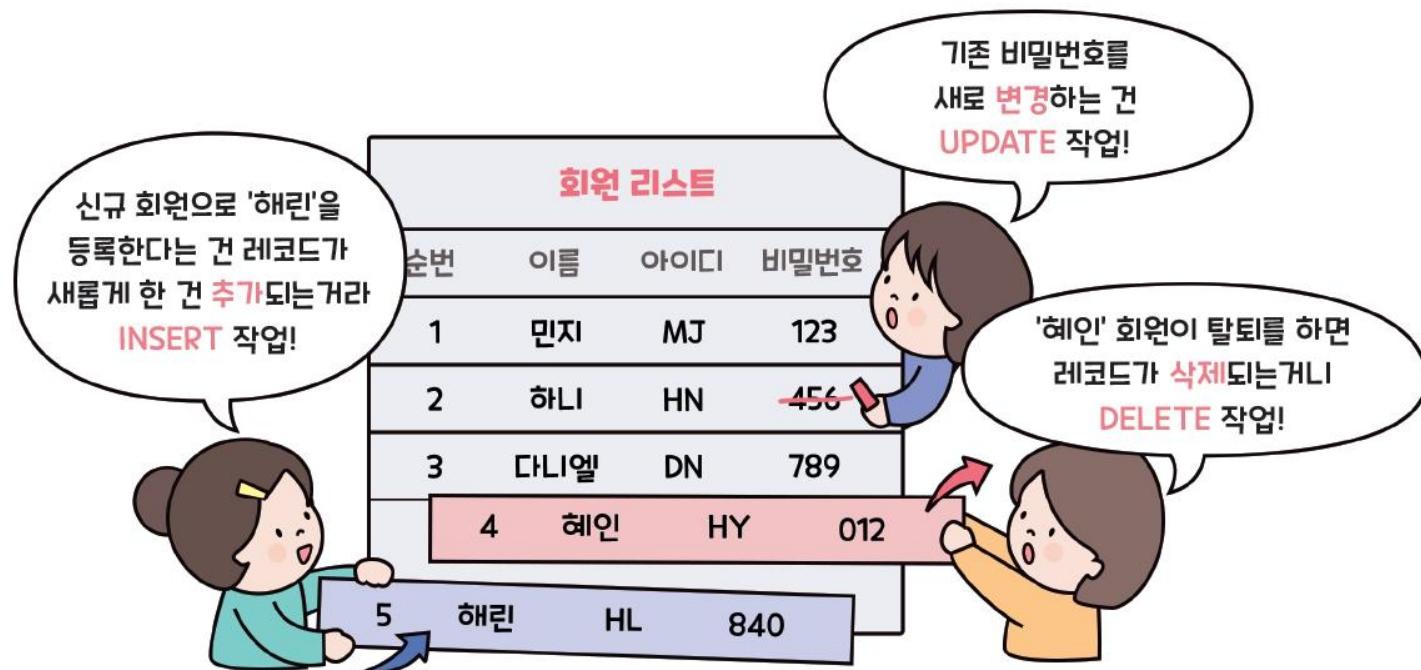


그림 7-1 INSERT, UPDATE, DELETE 작업

1. INSERT

■ INSERT

- 테이블에 새로운 행을 삽입하는 경우에 사용함
- 삽입할 컬럼명이 생략된 경우 삽입할 값이 테이블의 컬럼 수와 같아야 하며, 순서도 동일해야 함
- 형식

```
INSERT INTO 테이블명  
VALUES(값1, 값2, ...);
```

■ [예제 7-1] 부서 테이블에 다음 레코드를 삽입하시오.

- 부서번호: A5, 부서명: 마케팅부

```
INSERT INTO 부서  
VALUES('A5', '마케팅부');
```

▶ 실행결과

#	Time	Action
1	09:28:24	INSERT INTO 부서 VALUES('A5','마케팅부')

부서번호	부서명
A1	영업부
A2	기획부
A3	개발부
A4	홍보부
A5	마케팅부

그림 7-2 [예제 7-1] 레코드 삽입 결과

1. INSERT

■ [예제 7-2] 제품 테이블에 다음 레코드를 추가하시오

- 제품번호: 91, 제품명: 연어피클소스, 단가: 5000, 재고: 40

```
INSERT INTO 제품  
VALUES(91, '연어피클소스', NULL, 5000, 40);
```

▶ 실행결과

제품번호	제품명	포장단위	단가	재고
73	원양 순 상어알	150 g jars	3500	101
74	서울 구이 김	5 kg pkg.	1000	4
75	뉴트리인 맥주	0.5 l bottles	2400	125
76	찰스 초콜릿 드링크	500 ml	600	57
77	뉴트리 샐러드 드레싱	12 boxes	1300	32
78	엘몬트 망고 주스	0.5 l bottles	3000	80
91	연어피클소스	NULL	5000	40

1. INSERT

■ [예제 7-3] 제품 테이블에 다음 레코드를 추가하시오

- 제품번호: 90, 제품명: 연어핫소스, 단가: 4000, 재고: 50

```
INSERT INTO 제품(제품번호, 제품명, 단가, 재고)
```

```
VALUES(90, '연어핫소스', 4000, 50);
```

▶ 실행결과

제품번호	제품명	포장단위	단가	재고
76	원양 소스	150 g jars	101	10
77	뉴트리 샐러드 드레싱	12 boxes	1300	32
78	멜론트 망고 주스	0.5 l bottles	3000	80
90	연어핫소스	HULL	4000	50
91	연어피클소스	HULL	5000	40

1. INSERT

■ [예제 7-4] 사원 테이블에 한 문장으로 세 명의 레코드를 추가하시오.

- 사원번호: E20, 이름: 김사과, 직위: 수습사원, 성별: 남, 입사일: 현재 날짜
- 사원번호: E21, 이름: 박바나나, 직위: 수습사원, 성별: 여, 입사일: 현재 날짜
- 사원번호: E22, 이름: 정오렌지, 직위: 수습사원, 성별: 여, 입사일: 현재 날짜

```
INSERT INTO 사원(사원번호, 이름, 직위, 성별, 입사일)
VALUES('E20', '김사과', '수습사원', '남', CURDATE())
      ,('E21', '박바나나', '수습사원', '여', CURDATE())
      ,('E22', '정오렌지', '수습사원', '여', CURDATE());
```

▶ 실행 결과

사원번호	이름	영문이름	직위	성별	생일	입사일	주소	도시	지역	집전화	상사번호	부서번호
E01	이소미	Lee So Mi	사원	여	1985-12-05	2019-04-13	강남구 역삼동 36-8	서울특별시	경인	(02)578-8988	E06	A1
E02	배재용	Bae Jae Yong	대표이사	남	1973-02-17	2019-01-01	임미동 16-11	부천시	경인	(032)69-0136		A2
E03	우대현	Yoo Dae Hyeon	사원	남	1988-08-27	2019-03-14	광산구 송정동 100-11	광주광역시	초남	(062)73-0256	E06	A1
E04	최소민	Choi So Min	사원	여	1987-09-17	2019-04-15	중구 종양동 57-14	부산광역시	영남	(051)587-4783	E06	A1
E05	안주준	Ahn Ju Hoon	과장	남	1980-03-01	2018-12-29	남구 도파동 276-31	인천광역시	경인	(032)515-0278	E09	A1
E06	이현진	Lee Hyun Jin	대리	남	1983-06-30	2019-09-29	대덕구 이현동 577-1	대전광역시	충부	(042)518-3876	E05	A1
E07	오영수	O Yeong Soo	사원	남	1989-05-27	2019-12-15	문화동 76-77	광주시	충부	(043)19-1784	E06	A1
E08	선하라	Seon Ha Ra	전산팀장	여	1982-01-07	2019-02-16	의진구 고할동 116	전주시	초남	(065)2983-1985	E02	A3
E09	유가을	Yoo Ga Eul	부장	여	1980-01-25	2019-10-29	남구 대명동 19-7	대구광역시	영남	(053)465-1248	E02	A1
E10	정수진	Jung Su Jin	수습사원	여	1993-10-07	2022-03-19	동작구 흙석동 현대아파트 1-1501	서울특별시	경인	(02)824-2898		
E20	김사과	NULL	수습사원	남	NULL	2023-03-03	NULL	NULL	NULL	NULL	NULL	NULL
E21	박바나나	NULL	수습사원	여	NULL	2023-03-03	NULL	NULL	NULL	NULL	NULL	NULL
E22	정오렌지	NULL	수습사원	여	NULL	2023-03-03	NULL	NULL	NULL	NULL	NULL	NULL

2. UPDATE

■ UPDATE

- 기존 행에 있는 데이터 값을 변경할 때 사용함
- UPDATE문에 WHERE절이 없으면 모든 행의 값이 변경되므로 주의해야 함
- 형식

UPDATE 테이블명

SET 컬럼1 = 값1

,컬럼2 = 값2

[WHERE 조건];

만약 UPDATE문을 실행했을 때 오류가 발생한다면
2장 61p의 <하나 더 알기>를 참고하세요.

■ [예제 7-5] 사원번호가 'E20'인 사원의 데이터를 다음과 같이 수정하시오.

- (사원번호: E20, 이름: 김사과) → (사원번호: E20, 이름: 김레몬)

UPDATE 사원

SET 이름 = '김레몬'

WHERE 사원번호 = 'E20';

▶ 실행결과

사원번호	이름
E10	정수진
E20	김사과
E21	박바나나



사원번호	이름
E10	정수진
E20	김레몬
E21	박바나나

2. UPDATE

■ [예제 7-6] 제품번호가 91인 제품에 대하여 포장단위를 추가하시오.

- (제품번호: 91, 포장단위: NULL) → (제품번호: 91, 포장단위: 200 ml bottles)

UPDATE 제품

SET 포장단위 = '200 ml bottles'

WHERE 제품번호 = 91;

▶ 실행결과

제품 번호	제품명	포장단위
77	뉴트리 샐러드 드레싱	12 boxes
78	멜론트 망고 주스	0.5 l bottles
90	연어핫소스	NULL
91	연어피클소스	NULL



제품 번호	제품명	포장단위
77	뉴트리 샐러드 드레싱	12 boxes
78	멜론트 망고 주스	0.5 l bottles
90	연어핫소스	NULL
91	연어피클소스	200 ml bottles

2. UPDATE

■ [예제 7-7] 제품번호가 91인 제품에 대하여 단가를 10% 인상하고, 재고에서 10을 뺀 값으로 변경하시오.

UPDATE 제품

SET 단가 = 단가 * 1.1

,재고 = 재고 - 10

WHERE 제품번호 = 91;

▶ 실행결과

제품번호	제품명	포장단위	단가	재고
77	뉴트리 샐러드 드레싱	12 boxes	1300	32
78	델몬트 망고 주스	0.5 l bottles	3000	80
90	연어핫소스	HULL	4000	50
91	연어피클소스	200 ml bottles	5000	40



제품번호	제품명	포장단위	단가	재고
77	뉴트리 샐러드 드레싱	12 boxes	1300	32
78	델몬트 망고 주스	0.5 l bottles	3000	80
90	연어핫소스	HULL	4000	50
91	연어피클소스	200 ml bottles	5500	30

3. DELETE

■ DELETE

- 기존에 있는 행을 삭제할 때 사용함
- 만약 DELETE문에 WHERE절이 없으면 모든 행이 삭제되므로 주의해야 함
- 형식

```
DELETE FROM 테이블명  
[WHERE 조건];
```

■ [예제 7-8] 제품 테이블에서 제품번호가 91인 레코드를 삭제하시오.

```
DELETE FROM 제품  
WHERE 제품번호 = 91;
```

▶ 실행 결과

제품번호	제품명	포장단위	단가	재고
77	뉴트리 셀러드 드레싱	12 boxes	1300	32
78	멜론트 망고 주스	0.5 l bottles	3000	80
90	연어핫소스	HULL	4000	50
91	연어피클소스	200 ml bottles	5500	30



제품번호	제품명	포장단위	단가	재고
77	뉴트리 셀러드 드레싱	12 boxes	1300	32
78	멜론트 망고 주스	0.5 l bottles	3000	80
90	연어핫소스	HULL	4000	50

3. DELETE

- [예제 7-9] 사원 테이블에서 입사일이 가장 늦은 사원 3명의 레코드를 삭제하시오.

```
DELETE FROM 사원  
ORDER BY 입사일 DESC  
LIMIT 3;
```

- [예제 7-4]에서 추가한 '김레몬', '박바나나', '정오렌지'가 삭제된 것을 확인하는 쿼리

```
SELECT *  
FROM 사원  
WHERE 이름 IN('김레몬', '박바나나', '정오렌지');
```

▶ 실행결과

사원번호	이름	영문이름	직위	성별	생일	입사일	주소	도시	지역	집전화	상사번호	부서번호
NULL												

4. INSERT ON DUPLICATE KEY UPDATE

■ INSERT ON DUPLICATE KEY UPDATE

- 레코드가 없다면 새롭게 추가하고, 이미 있다면 데이터를 변경하는 경우에 사용
- 형식

```
INSERT INTO 테이블명(컬럼1, 컬럼2, 컬럼3, ...) • ①  
VALUES(값1, 값2, 값3, ...)  
ON DUPLICATE KEY UPDATE  
컬럼2 = 값2, 컬럼3 = 값3, ...;
```

4. INSERT ON DUPLICATE KEY UPDATE

■ [예제 7-10] 91번 제품이 없다면 레코드를 추가하고, 이미 존재한다면 값을 변경하시오.

- 제품번호: 91, 제품명: 연어피클핫소스, 단가: 6000, 재고: 50

```
INSERT INTO 제품(제품번호, 제품명, 단가, 재고)
VALUES(91, '연어피클핫소스', 6000, 50)
ON DUPLICATE KEY UPDATE
제품명 = '연어피클핫소스', 단가 = 6000, 재고 = 50;
```

▶ 실행결과

제품번호	제품명	포장단위	단가	재고
75	뉴트리인 맥주	0.5 l bottles	2400	125
76	참스 초콜릿 드링크	500 ml	600	57
77	뉴트리 샐러드 드레싱	12 boxes	1300	32
78	멜론트 망고 주스	0.5 l bottles	3000	80
90	연어핫소스	HULL	4000	50
91	연어피클핫소스	HULL	6000	50

4. INSERT ON DUPLICATE KEY UPDATE

확인문제

레코드가 존재하지 않으면 새롭게 추가하고, 이미 있다면 데이터를 변경하는 작업을 한 문장으로 해결하고자 한다. 빈칸을 채워 문장을 완성하시오.

```
INSERT INTO 고객(고객번호, 담당자명, 고객회사명, 도시)  
VALUES('ZAQAQ', '김지현', '두빛트레이드', '서울특별시')
```

담당자명 = '오성균', 도시 = '인천광역시';

정답

```
ON DUPLICATE KEY UPDATE
```

Section 02

DML 심화

1. INSERT INTO SELECT

■ INSERT INTO SELECT

- SELECT문의 결과를 다른 테이블에 삽입하려면 INSERT문에서 VALUES 대신 SELECT 문장을 넣음
- 이때 SELECT문의 컬럼 수와 INSERT문에 있는 컬럼 수는 동일해야 하고, 순서도 같아야 함
- 형식

```
INSERT INTO 테이블명(컬럼A, 컬럼B)
    SELECT 컬럼1, 컬럼2
    FROM 테이블명
    [WHERE 조건];
```

1. INSERT INTO SELECT

■ [예제 7-11] 고객주문요약 테이블을 만들고, 레코드를 추가하시오.

- 테이블 생성에 관한 내용은 8장에서 다룸

표 7-1 고객주문요약 테이블의 컬럼과 자료형

컬럼명	자료형
고객번호(기본키)	CHAR(5)
고객회사명	VARCHAR(50)
주문건수	INT
최종주문일	DATE

- 고객주문요약 테이블 생성

```
CREATE TABLE 고객주문요약
(
    고객번호 CHAR(5) PRIMARY KEY
    ,고객회사명 VARCHAR(50)
    ,주문건수 INT
    ,최종주문일 DATE
);
```

1. INSERT INTO SELECT

■ [예제 7-11] 고객주문요약 테이블을 만들고, 레코드를 추가하시오.

- 레코드 삽입

```
INSERT INTO 고객주문요약
```

```
    SELECT 고객.고객번호  
          ,고객회사명  
          ,COUNT(*)  
          ,MAX(주문일)  
     FROM 고객  
      ,주문  
    WHERE 고객.고객번호 = 주문.고객번호  
  GROUP BY 고객.고객번호  
        ,고객회사명;
```

▶ 실행결과

고객번호	고객회사명	주문건수	최종주문일
ACDDR	굿모닝서울	6	2022-01-11
ADHTR	엘케이 상사	6	2021-09-27
AIHTR	씨엔그룹	3	2021-09-16
ANKFR	오리안무역	15	2021-12-17
ANRFR	남해종합식품	3	2021-12-01
ANSFR	서보앱엔에프	6	2022-01-07

2. UPDATE SELECT

■ UPDATE SELECT

- UPDATE문에서 서브쿼리를 사용할 수도 있음
- 변경할 값을 얻기 위해서는 SET절에서 SELECT문을 사용함
- 서브쿼리의 SELECT문에는 반드시 별명을 붙여야 하고, 서브쿼리의 결과는 단일 값이어야 함
- 형식

```
UPDATE 테이블명  
SET 컬럼명 = (  
    SELECT *  
    FROM (  
        SELECT 컬럼A  
        FROM ...  
    ) AS 별명  
    WHERE 조건  
);
```

2. UPDATE SELECT

- [예제 7-12] 제품번호가 91인 제품의 단가를 '소스' 제품들의 평균단가로 변경하시오.

```
UPDATE 제품
SET 단가 = (
    SELECT *
    FROM (
        SELECT AVG(단가)
        FROM 제품
        WHERE 제품명 LIKE '%소스%'
    ) AS t
)
WHERE 제품번호 = 91;
```

▶ 실행결과

제품번호	제품명	단가	재고
75	뉴트리인 맥주	2400	125
76	참스 초콜릿 드링크	600	57
77	뉴트리 샐러드 드레싱	1300	32
78	멜론트 망고 주스	3000	80
90	연어핫소스	4000	50
91	연어피클핫소스	6000	50



제품번호	제품명	단가	재고
75	뉴트리인 맥주	2400	125
76	참스 초콜릿 드링크	600	57
77	뉴트리 샐러드 드레싱	1300	32
78	멜론트 망고 주스	3000	80
90	연어핫소스	4000	50
91	연어피클핫소스	2831	50

2. UPDATE SELECT

■ UPDATE SELECT

- WHERE절에서 서브쿼리 사용
- 형식

```
UPDATE 변경할_테이블명  
    ,서브쿼리  
SET 컬럼1 = 변경할 값  
WHERE 컬럼2 IN (서브쿼리.컬럼);
```

2. UPDATE SELECT

■ [예제 7-13] 한 번이라도 주문한 적이 있는 고객의 마일리지를 10% 인상하시오.

```
UPDATE 고객
  (
    SELECT DISTINCT 고객번호
    FROM 주문
    ) AS 주문고객
SET 마일리지 = 마일리지 * 1.1
WHERE 고객.고객번호 IN (주문고객.고객번호);
```

- 결과를 확인하기 위한 SELECT문

```
SELECT *
FROM 고객
WHERE 고객번호 IN (SELECT DISTINCT 고객번호
                     FROM 주문
                    );
```

▶ 실행결과

고객번호	고객회사명	담당자명	담당자직위	주소	도시	지역	전화번호	마일리지
ACDDR	굿모닝서울	이은광	영업 과장	송파구 잠실동 220	서울특별시	NULL	(02)978-1984	17502
ADHTR	얼케이 상사	김병현	영업 사원	동작구 은석 3동 140-3	서울특별시	NULL	(02)345-1945	447
AIHTR	씨엔그룹	김성민	영업 사원	가락동 301	광명시	경기도	(02)31-0345	9667
ANKFR	오리안무역	최지수	마케팅 과장	성북구 길을동 136-11	서울특별시	NULL	(02)123-0345	6731
ANLSD	남해줄화시프	김준	마케팅 과장	성북구 길을동 435	구리시	경기도	(034)756-2004	1111

3. UPDATE JOIN

■ UPDATE JOIN

- INNER JOIN을 사용하여 다른 테이블의 행과 일치하는 행을 수정할 수 있음
- LEFT OUTER JOIN을 사용하여 그렇지 않은 행을 수정할 수도 있음
- 변경할 값에는 상수나 수식뿐만 아니라 조인할 테이블에 있는 컬럼에 기반하여 값을 넣을 수도 있음
- ANSI SQL 표현

```
UPDATE 변경할_테이블A  
[INNER|LEFT| OUTER] JOIN 조인할_테이블B  
ON 조인_조건  
SET 테이블A.컬럼명 = 값  
WHERE 기타_조건;
```

- Non-ANSI SQL 표현

```
UPDATE 변경할_테이블A  
    ,조인할_테이블B  
SET 테이블A.컬럼명 = 값  
WHERE 조인_조건  
AND 기타_조건;
```

3. UPDATE JOIN

■ [예제 7-14] 마일리지등급이 'S'인 고객의 마일리지에 1000점씩 추가하시오.

- ANSI SQL 표현

```
UPDATE 고객
INNER JOIN 마일리지등급
ON 마일리지 BETWEEN 하한마일리지 AND 상한마일리지
SET 마일리지 = 마일리지 + 1000
WHERE 등급명 = 'S';
```

- Non-ANSI SQL 표현

```
UPDATE 고객
,마일리지등급
SET 마일리지 = 마일리지 + 1000
WHERE 마일리지 BETWEEN 하한마일리지 AND 상한마일리지
AND 등급명 = 'S';
```

3. UPDATE JOIN

■ [예제 7-14] 마일리지등급이 'S'인 고객의 마일리지에 1000점씩 추가하시오.

- 결과를 확인하기 위한 SELECT문

```
SELECT 고객번호  
      ,고객회사명  
      ,마일리지  
FROM 고객  
INNER JOIN 마일리지등급  
ON 마일리지 BETWEEN 하한마일리지AND 상한마일리지  
WHERE 등급명 = 'S';
```

▶ 실행결과

고객번호	고객회사명	마일리지
ATRAN	대정 인터내셔널	141669
PRDSU	활인터프라이시스	128588



고객번호	고객회사명	마일리지
ATRAN	대정 인터내셔널	142669
PRDSU	활인터프라이시스	129588

4. DELETE SELECT

■ DELETE SELECT

- DELETE문에서도 삭제할 레코드를 찾기 위하여 서브쿼리를 사용할 수 있음
- 형식

```
DELETE FROM 테이블A  
WHERE 컬럼명 IN (  
    SELECT 컬럼명  
    FROM 테이블B  
);
```

■ [예제 7-15] 주문 테이블에는 존재하나 주문세부 테이블에는 존재하지 않는 주문번호를 주문 테이블에서 삭제하시오.

```
DELETE FROM 주문  
WHERE 주문번호 NOT IN (  
    SELECT DISTINCT 주문번호  
    FROM 주문세부  
);
```

4. DELETE SELECT

■ [예제 7-15] 주문 테이블에는 존재하나 주문세부 테이블에는 존재하지 않는 주문번호를 주문 테이블에서 삭제하시오.

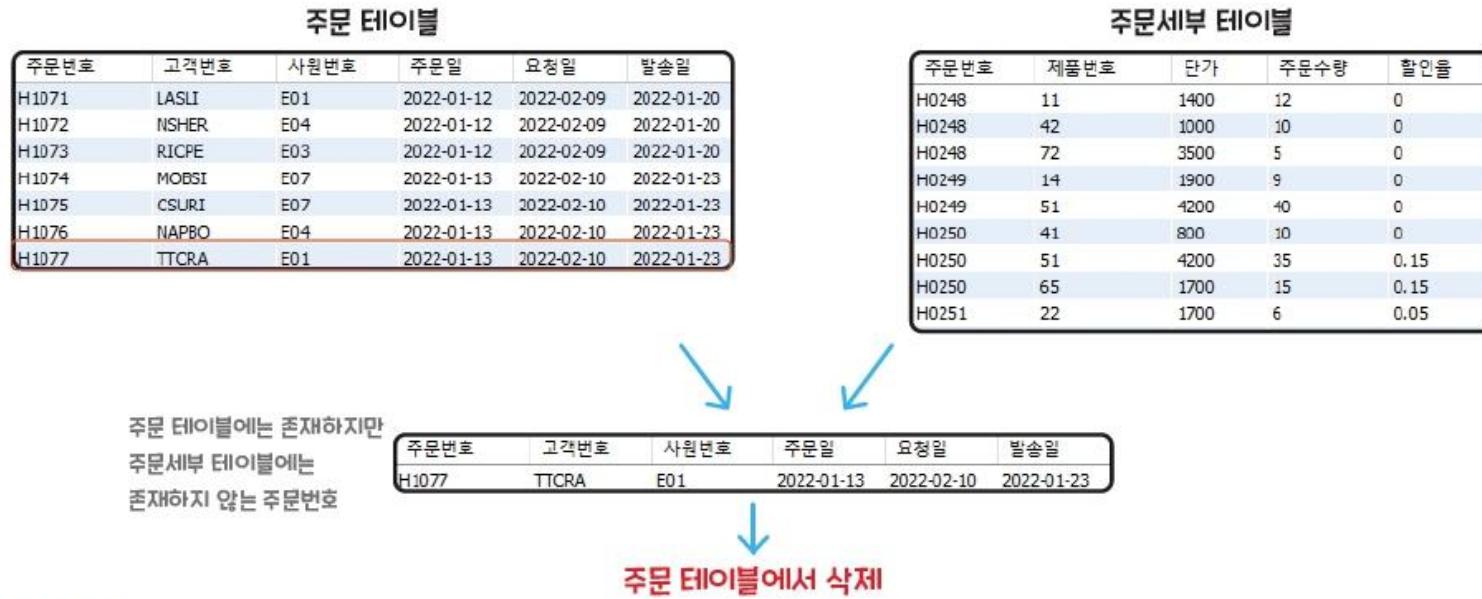


그림 7-3 [예제 7-15] 진행 과정

4. DELETE SELECT

하나 더 알기 ✓

Collation 에러가 발생하는 경우

조인을 하거나 IF, CASE 등으로 값을 비교할 때, 양쪽 테이블의 Collation이 다를 때 다음과 같은 오류가 발생할 수 있습니다.

Error Code: 1267. Illegal mix of collations

이런 오류가 발생할 때는 다음과 같이 테이블의 Collation을 변경시켜주면 문제를 해결할 수 있습니다.

```
ALTER TABLE 고객 CONVERT TO CHAR SET utf8mb4 COLLATE utf8mb4_general_ci;  
ALTER TABLE 주문 CONVERT TO CHAR SET utf8mb4 COLLATE utf8mb4_general_ci;  
ALTER TABLE 주문세부 CONVERT TO CHAR SET utf8mb4 COLLATE utf8mb4_general_ci;  
ALTER TABLE 제품 CONVERT TO CHAR SET utf8mb4 COLLATE utf8mb4_general_ci;  
...
```



5. DELETE JOIN

■ DELETE JOIN

- INNER JOIN을 사용하여 두 테이블에서 일치하는 행을 모두 삭제할 수 있음
- LEFT OUTER JOIN을 사용하여 일치하지 않는 행을 삭제할 수도 있음
- ANSI SQL 표현

```
DELETE 테이블A  
      ,테이블B  
FROM 테이블A  
[INNER|LEFT OUTER] JOIN 테이블B  
ON 조인_조건  
WHERE 삭제할_조건;
```

- Non-ANSI SQL 표현

```
DELETE 테이블A  
      ,테이블B  
FROM 테이블A  
      ,테이블B  
WHERE 조인_조건  
AND 삭제할_조건;
```

5. DELETE JOIN

■ [예제 7-16] 주문번호 'H0248'에 대한 내역을 주문 테이블과 주문세부 테이블에서 모두 삭제하시오. 하나의 문장으로 작업을 수행해보시오.

- 주문 테이블, 주문세부 테이블에 주문번호 'H0248'이 존재하는지 확인하기

```
SELECT *
FROM 주문
WHERE 주문번호 = 'H0248';
```

▶ 실행결과

주문번호	고객번호	사원번호	주문일	요청일	발송일
H0248	NETVI	E04	2020-03-12	2020-04-09	2020-03-24

```
SELECT *
FROM 주문세부
WHERE 주문번호 = 'H0248';
```

▶ 실행결과

주문번호	제품번호	단가	주문수량	할인율
H0248	11	1400	12	0
H0248	42	1000	10	0
H0248	72	3500	5	0

5. DELETE JOIN

■ [예제 7-16] 주문번호 'H0248'에 대한 내역을 주문 테이블과 주문세부 테이블에서 모두 삭제하시오. 하나의 문장으로 작업을 수행해보시오.

- 주문 테이블, 주문세부 테이블에 주문번호 'H0248'를 동시에 삭제하기
- ANSI SQL 표현

```
DELETE 주문
    ,주문세부
FROM 주문
INNER JOIN 주문세부
ON 주문.주문번호 = 주문세부.주문번호
WHERE 주문.주문번호 = 'H0248';
```

- Non-ANSI SQL 표현

```
DELETE 주문
    ,주문세부
FROM 주문
    ,주문세부
WHERE 주문.주문번호 = 주문세부.주문번호
AND 주문.주문번호 = 'H0248';
```

▶ 실행결과

- 주문 테이블에서 주문번호 'H0248' 레코드 확인

주문번호	고객번호	사원번호	주문일	요청일	발송일
NULL	NULL	NULL	NULL	NULL	NULL

- 주문세부 테이블에서 주문번호 'H0248' 레코드 확인

주문번호	제품번호	단가	주문수량	할인율
NULL	NULL	NULL	NULL	NULL

5. DELETE JOIN

■ [예제 7-17] 한 번도 주문한 적이 없는 고객의 정보를 삭제하시오

- 한 번도 주문한 적이 없는 고객 검색하기

```
SELECT 고객.*  
FROM 고객  
LEFT OUTER JOIN 주문  
ON 고객.고객번호 = 주문.고객번호  
WHERE 주문.고객번호 IS NULL;
```

▶ 실행 결과

고객번호	고객회사명	담당자명	담당자직위	주소	도시	지역	전화번호	마일리지
BQQZA	비큐큐푸드	황수영	대표 이사	서초구 서초동 120-3	서울특별시	NULL	(02)456-9876	100
RISPA	한림통상	김정현	대표 이사	송파구 잠실본동 150	서울특별시	NULL	(02)909-2094	6942
SSAFI	대동 무역	정수환	회계 과장	동구 마산동 250-1	대전광역시	NULL	(042)33-3433	975
TTRAN	해피트리푸드	김희원	영업 과장	동작구 읍석동 220-12	서울특별시	NULL	(02)756-9787	448

5. DELETE JOIN

■ [예제 7-17] 한 번도 주문한 적이 없는 고객의 정보를 삭제하시오

- 주문한 적이 없는 고객의 레코드를 고객 테이블에서 삭제하기

```
DELETE 고객
FROM 고객
LEFT JOIN 주문
ON 고객.고객번호 = 주문.고객번호
WHERE 주문.고객번호 IS NULL;
```

- 삭제된 고객 정보가 고객 테이블에 존재하는지 확인하기

```
SELECT *
FROM 고객
WHERE 고객번호 IN ('BQQZA', 'RISPA', 'SSAFI', 'TTRAN');
```

▶ 실행결과

고객번호	고객회사명	담당자명	담당자직위	주소	도시	지역	전화번호	마일리지
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

5. DELETE JOIN

확인문제

다음 지문을 읽고 옳은 것에는 ○, 틀린 것에는 ×를 표시하시오.

1. INSERT문에서 서브쿼리를 사용할 수 있다.()
2. UPDATE문의 SET절에서 서브쿼리를 사용할 수 없다.()
3. UPDATE문의 WHERE절에서 서브쿼리를 사용할 수 있다.()
4. UPDATE문에서 INNER JOIN을 사용하여 다른 테이블의 행과 일치하는 행을 수정할 수 있다.()
5. DELETE문에서 INNER JOIN을 사용하여 두 테이블에서 일치하는 행을 모두 삭제할 수 없다.()

정답

1. ○ 2. × 3. ○ 4. ○ 5. ×

점검문제

점검문제

문제 1

제품 테이블에 레코드를 추가하시오.

- 제품번호: 95, 제품명: 망고베리 아이스크림, 포장단위: 400g, 단가: 800, 재고: 30

▶ 실행결과

제품번호	제품명	포장단위	단가	재고
78	웰몬트 망고 주스	0.5l bottles	3000	80
90	연어핫소스	NULL	4000	50
91	연어피클핫소스	NULL	3283	50
95	망고베리 아이스크림	400g	800	30

문제 2

제품 테이블에 레코드를 추가하시오.

- 제품번호: 96, 제품명: 눈꽃빙수맛 아이스크림, 단가: 2000

▶ 실행결과

제품번호	제품명	포장단위	단가	재고
90	연어핫소스	NULL	4000	50
91	연어피클핫소스	NULL	3283	50
95	망고베리 아이스크림	400g	800	30
96	눈꽃빙수맛 아이스크림	NULL	2000	NULL

점검문제

문제 3

[문제 2]에서 추가한 96번 제품의 재고를 30으로 변경하시오.

▶ 실행결과

제품 번호	제품명	포장단위	단가	재고
90	연어핫소스	HULL	4000	50
91	연어피클핫소스	HULL	3283	50
95	망고베리 아이스크림	400g	800	30
96	눈꽃빙수맛 아이스크림	HULL	2000	30

문제 4

사원이 한 명도 존재하지 않는 부서를 부서 테이블에서 삭제하시오.

▶ 실행결과

부서번호	부서명
A1	영업부
A2	기획부
A3	개발부
A4	홍보부
A5	마케팅부



부서번호	부서명
A1	영업부
A2	기획부
A3	개발부

Thank you!

