

Ch07. 데이터베이스 언어 SQL-2

SQL의 역할을 이해하고, 이를 기능별로 분류해본다.

SQL의 데이터 정의 기능을 예제를 통해 익힌다.

SQL의 데이터 조작 기능을 예제를 통해 익힌다.

뷰의 개념과 장점을 이해한다.

삽입 SQL의 역할을 이해한다

복음 앞부분 그 앞부분



명지대학교
MYONGJI UNIVERSITY

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

- 그룹별 검색

```
SELECT  [ ALL | DISTINCT ] 속성_리스트  
FROM    테이블_리스트  
[ WHERE 조건 ]  
[ GROUP BY 속성_리스트 [ HAVING 조건 ] ]  
[ ORDER BY 속성_리스트 [ ASC | DESC ] ];
```

- GROUP BY 키워드를 이용해 특정 속성의 값이 같은 튜플을 모아 그룹을 만들고, 그룹별로 검색
 - › GROUP BY 키워드와 함께 그룹을 나누는 기준이 되는 속성을 지정
- HAVING 키워드를 함께 이용해 그룹에 대한 조건을 작성
- 그룹을 나누는 기준이 되는 속성을 SELECT 절에도 작성하는 것이 좋음

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

- 그룹별 검색

예제 7-32

주문 테이블에서 주문제품별 수량의 합계를 검색해보자.

```
▶▶ SELECT  주문제품, SUM(수량) AS 총주문수량
FROM      주문
GROUP BY  주문제품;
```

결과 테이블

	주문제품	총주문수량
1	p03	52
2	p02	58
3	p06	81
4	p04	15
5	p01	24

수량	배송지	주문일자
10	서울시 마포구	2019-01-01
5	인천시 계양구	2019-01-10
45	경기도 부천시	2019-01-11
8	부산시 금정구	2019-02-01
36	경기도 용인시	2019-02-20
19	충청북도 보은군	2019-03-02
22	서울시 영등포구	2019-03-15
50	강원도 춘천시	2019-04-10
15	전라남도 목포시	2019-04-11
20	경기도 안양시	2019-05-22

o10

carrot

p03

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

- 그룹별 검색

주문제품	수량
p03	10
p01	5
p06	45
p02	8
p06	36
p01	19
p03	22
p02	50
p04	15
p03	20

동일 제품을 주문한 튜플을 모아 그룹으로 만들고,
그룹별로 수량의 합계를 계산

	주문제품	총주문수량
1	p03	52
2	p02	58
3	p06	81
4	p04	15
5	p01	24

그림 7-10 주문제품별 수량의 합계를 계산하는 과정

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

■ 그룹별 검색

예제 7-33

제품 테이블에서 제조업체별로 제조한 제품의 개수와 제품 중 가장 비싼 단가를 검색하되, 제품의 개수는 제품수라는 이름으로 출력하고 가장 비싼 단가는 최고가라는 이름으로 출력해 보자.

▶▶ SELECT 제조업체, COUNT(*) AS 제품수, MAX(단가) AS 최고가
FROM 제품
GROUP BY 제조업체;

결과 테이블

	제조업체	제품수	최고가
1	대한식품	2	4500
2	민국푸드	2	5500
3	한빛제과	3	2600

제품 테이블

제품번호	제품명	재고량	단가	제조업체
p01	그냥만두	5000	4500	대한식품
p02	매운쫄면	2500	5500	민국푸드
p03	콩떡파이	3600	2600	한빛제과
p04	맛난초콜릿	1250	2500	한빛제과
p05	얼큰라면	2200	1200	대한식품
p06	통통우동	1000	1550	민국푸드
p07	달콤비스킷	1650	1500	한빛제과

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

- 그룹별 검색

예제 7-34

제품 테이블에서 제품을 3개 이상 제조한 제조업체별로 제품의 개수와, 제품 중 가장 비싼 단가를 검색해보자.

```
▶▶ SELECT  제조업체, COUNT(*) AS 제품수, MAX(단가) AS 최고가
FROM      제품
GROUP BY  제조업체 HAVING COUNT(*) >= 3;
```

결과 테이블

	제조업체	제품수	최고가
1	한빛제과	3	2600

제품 테이블

제품번호	제품명	재고량	단가	제조업체
p01	그냥만두	5000	4500	대한식품
p02	매운쫄면	2500	5500	민국푸드
p03	콩떡파이	3600	2600	한빛제과
p04	맛난초콜릿	1250	2500	한빛제과
p05	얼큰라면	2200	1200	대한식품
p06	통통우동	1000	1550	민국푸드
p07	달콤비스킷	1650	1500	한빛제과

집계 함수를 이용한 조건은 WHERE 절에는 작성할 수 없고 HAVING 절에서 작성

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

- 그룹별 검색

예제 7-35

고객 테이블에서 적립금 평균이 1,000원 이상인 등급에 대해 등급별 고객수와 적립금 평균을 검색해보자.

AVG(적립금), COUNT(고객아이디)

▶▶ SELECT 등급, COUNT(*) AS 고객수, AVG(적립금) AS 평균적립금
FROM 고객
GROUP BY 등급 HAVING AVG(적립금) >= 1000;

결과 테이블

	등급	고객수	평균적립금
1	gold	3	3500
2	vip	1	2500

고객 테이블

고객아이디	고객이름	나이	등급	직업	적립금
apple	정소화	20	gold	학생	1000
banana	김선우	25	vip	간호사	2500
carrot	고명석	28	gold	교사	4500
orange	김용욱	22	silver	학생	0
melon	성원용	35	gold	회사원	5000
peach	오형준	NULL	silver	의사	300
pear	채광주	31	silver	회사원	500

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

■ 그룹별 검색

예제 7-36

주문 테이블에서 각 주문고객이 주문한 제품의 총주문수량을 주문제품별로 검색해보자.

▶▶ SELECT 주문제품, 주문고객, SUM(수량) AS 총주문수량
FROM 주문
GROUP BY 주문제품, 주문고객;

결과 테이블

하나의 그룹

	주문제품	주문고객	총주문수량
1	p02	carrot	8
2	p01	banana	19
3	p06	melon	36
4	p03	apple	32
5	p01	melon	5
6	p02	pear	50
7	p03	carrot	20
8	p06	banana	45
9	p04	banana	15

주문 테이블

주문번호	주문고객	주문제품	수량	배송지	주문일자
o01	apple	p03	10	서울시 마포구	2022-01-01
o02	melon	p01	5	인천시 계양구	2022-01-10
o03	banana	p06	45	경기도 부천시	2022-01-11
o04	carrot	p02	8	부산시 금정구	2022-02-01
o05	melon	p06	36	경기도 용인시	2022-02-20
o06	banana	p01	19	충청북도 보은군	2022-03-02
o07	apple	p03	22	서울시 영등포구	2022-03-15
o08	pear	p02	50	강원도 춘천시	2022-04-10
o09	banana	p04	15	전라남도 목포시	2022-04-11
o10	carrot	p03	20	경기도 안양시	2022-05-22

그림 7-6 데이터 조작 예제에서 사용하는 판매 데이터베이스 : 고객, 제품, 주문 테이블

집계 함수나 GROUP BY 절에 명시된 속성 외의 속성은 SELECT 절에 작성 불가

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

- 여러 테이블에 대한 조인 검색
 - 조인 검색: 여러 개의 테이블을 연결하여 데이터를 검색하는 것
 - 조인 속성: 조인 검색을 위해 테이블을 연결해주는 속성
 - › 연결하려는 테이블 간에 조인 속성의 이름은 달라도 되지만 도메인은 같아야 함
 - › 일반적으로 외래키를 조인 속성으로 이용함
 - FROM 절에 검색에 필요한 모든 테이블을 나열
 - WHERE 절에 조인 속성의 값이 같아야 함을 의미하는 조인 조건을 제시
 - 같은 이름의 속성이 서로 다른 테이블에 존재할 수 있기 때문에 속성 이름 앞에 해당 속성이 소속된 테이블의 이름을 표시
 - 예) 주문.주문고객

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

- 여러 테이블에 대한 조인 검색

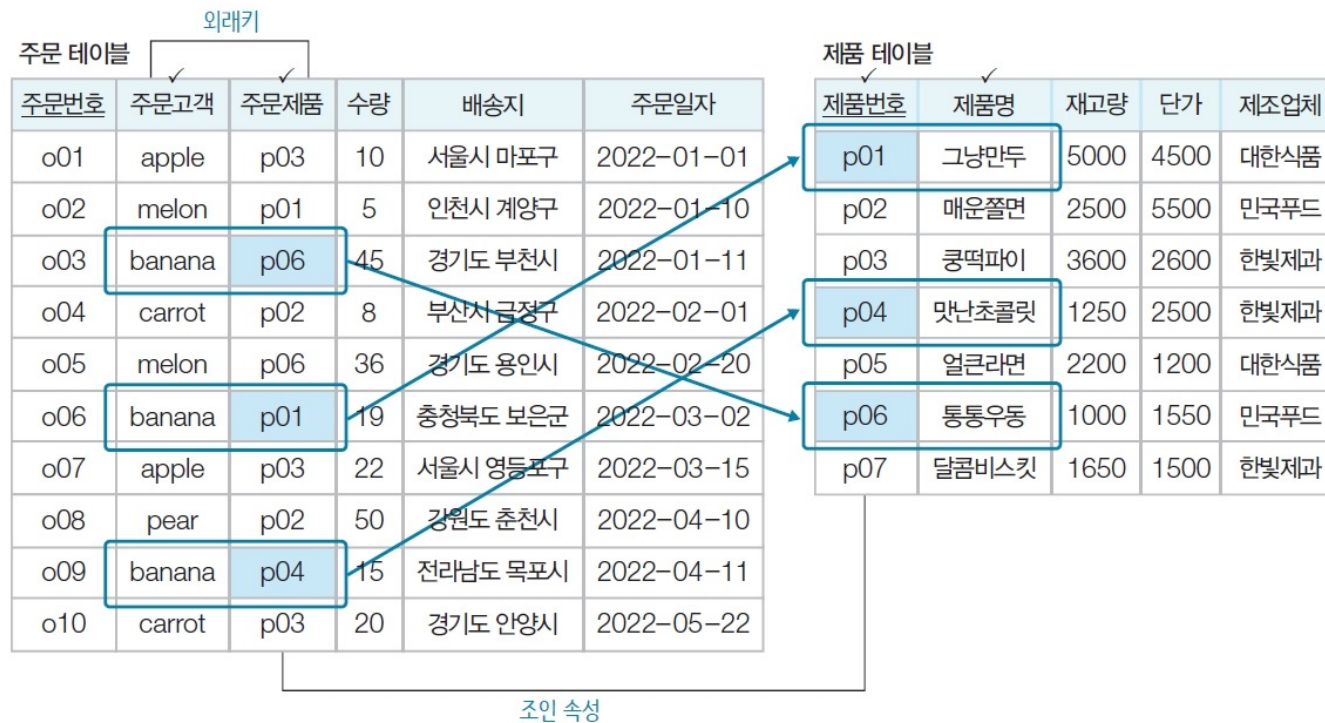


그림 7-11 2개의 테이블을 이용한 조인 검색 예 : 주문 테이블과 제품 테이블

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

- 여러 테이블에 대한 조인 검색

예제 7-37

판매 데이터베이스에서 banana 고객이 주문한 제품의 이름을 검색해보자.

▶▶ SELECT 제품.제품명
FROM 제품, 주문
WHERE 주문.주문고객 = 'banana' AND 제품.제품번호 = 주문.주문제품;

결과 테이블

	제품명
1	그냥만두
2	맛난초콜릿
3	통통우동

제품 테이블

제품번호	제품명	재고량	단가	제조업체
p01	그냥만두	5000	4500	대한식품
p02	매운쫄면	2500	5500	민국푸드
p03	쿵떡파이	3600	2600	한빛제과
p04	맛난초콜릿	1250	2500	한빛제과
p05	얼큰라면	2200	1200	대한식품
p06	통통우동	1000	1550	민국푸드
p07	달콤비스킷	1650	1500	한빛제과

주문 테이블

주문번호	주문고객	주문제품	수량	배송지	주문일자
o01	apple	p03	10	서울시 마포구	2022-01-01
o02	melon	p01	5	인천시 계양구	2022-01-10
o03	banana	p06	45	경기도 부천시	2022-01-11
o04	carrot	p02	8	부산시 금정구	2022-02-01
o05	melon	p06	36	경기도 용인시	2022-02-20
o06	banana	p01	19	충청북도 보은군	2022-03-02
o07	apple	p03	22	서울시 영등포구	2022-03-15
o08	pear	p02	50	강원도 춘천시	2022-04-10
o09	banana	p04	15	전라남도 목포시	2022-04-11
o10	carrot	p03	20	경기도 안양시	2022-05-22

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

- 여러 테이블에 대한 조인 검색

예제 7-38

판매 데이터베이스에서 나이가 30세 이상인 고객이 주문한 제품의 번호와 주문일자를 검색해보자.

```
▶▶ SELECT   주문.주문제품, 주문.주문일자
FROM       고객, 주문
WHERE      고객.나이 >= 30 AND 고객.고객아이디 = 주문.주문고객;
```

결과 테이블

	주문제품	주문일자
1	p01	22/01/10
2	p06	22/02/20
3	p02	22/04/10

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

- 여러 테이블에 대한 조인 검색

```
SELECT  주문제품, 주문일자
FROM    고객 c, 주문 o
WHERE   c.나이 >= 30 AND o.주문고객 = c.고객아이디;
```

이름이 같은 속성이 없다면
테이블 이름 없이 속성 이름으로만 작성해도 된다.

테이블의 이름을 대신하는 단순한 별명을 제시하여
질의문을 작성하는 것도 좋다.

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

- 여러 테이블에 대한 조인 검색

예제 7-39

판매 데이터베이스에서 고명석 고객이 주문한 제품의 제품명을 검색해보자.

```

▶▶ SELECT   제품.제품명
FROM         고객, 제품, 주문
WHERE        고객.고객이름 = '고명석' AND 고객.고객아이디 = 주문.주문고객
            AND 제품.제품번호 = 주문.주문제품;
    
```

결과 테이블

	제품명
1	매운쫄면
2	콩떡파이

제품 테이블

제품번호	제품명	재고량	단가	제조업체
p01	그냥만두	5000	4500	대한식품
p02	매운쫄면	2500	5500	민국푸드
p03	콩떡파이	3600	2600	한빛제과
p04	맛난초콜릿	1250	2500	한빛제과
p05	얼큰라면	2200	1200	대한식품
p06	통통우동	1000	1550	민국푸드
p07	달콤비스킷	1650	1500	한빛제과

고객 테이블

고객아이디	고객이름	나이	등급	직업	적립금
apple	정소화	20	gold	학생	1000
banana	김선우	25	vip	간호사	2500
carrot	고명석	28	gold	교사	4500
orange	김용욱	22	silver	학생	0
melon	성원웅	35	gold	회사원	5000
peach	오형준	NULL	silver	의사	300
pear	채광주	31	silver	회사원	500

주문 테이블

주문번호	주문고객	주문제품	수량	배송지	주문일자
o01	apple	p03	10	서울시 마포구	2022-01-01
o02	melon	p01	5	인천시 계양구	2022-01-10
o03	banana	p06	45	경기도 부천시	2022-01-11
o04	carrot	p02	8	부산시 금정구	2022-02-01
o05	melon	p06	36	경기도 용인시	2022-02-20
o06	banana	p01	19	충청북도 보은군	2022-03-02
o07	apple	p03	22	서울시 영등포구	2022-03-15
o08	pear	p02	50	강원도 춘천시	2022-04-10
o09	banana	p04	15	전라남도 목포시	2022-04-11
o10	carrot	p03	20	경기도 안양시	2022-05-22

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

- 여러 테이블에 대한 조인 검색
 - 표준 SQL에서는 INNER JOIN과 ON 키워드를 이용해 작성하는 방법도 제

```
SELECT  속성_리스트  
FROM    테이블1 INNER JOIN 테이블2 ON  조인조건  
[ WHERE 검색조건 ]
```

(예제 7-38) 나이가 30세 이상인 고객이 주문한 제품의 번호와 주문일자 검색

```
SELECT  주문.주문제품, 주문.주문일자  
FROM    고객 INNER JOIN 주문 ON  고객.고객아이디 = 주문.주문고객  
WHERE   고객.나이 >= 30;
```

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문



- 외부 조인 검색
 - 조인 조건을 만족하지 않는 튜플에 대해서도 검색을 수행
 - OUTER JOIN과 ON 키워드를 이용해 작성

```
SELECT 속성_리스트  
FROM 테이블1 LEFT | RIGHT | FULL OUTER JOIN 테이블2 ON 조인조건  
[ WHERE 검색조건 ]
```

- 분류
 - › 모든 튜플을 검색 대상으로 하는 테이블이 무엇이냐에 따라 분류
 - › 왼쪽 외부 조인, 오른쪽 외부 조인, 완전 외부 조인

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

- 외부 조인 검색

```
SELECT  고객.고객이름, 주문.주문제품, 주문.주문일자
FROM    고객 LEFT OUTER JOIN  주문 ON  고객.고객아이디 = 주문.주문고객;
```

주문하지 않은 고객도 포함해서
고객이름, 주문제품, 주문일자를 검색

인덱스, 오른쪽 순서로
나열하면 결과값 달라짐

결과 테이블

	고객이름	주문제품	주문일자
1	정소화	p03	2022-01-01
2	정소화	p03	2022-03-15
3	김선우	p06	2022-01-11
4	김선우	p01	2022-03-02
5	김선우	p04	2022-04-11
6	고명석	p02	2022-02-01
7	고명석	p03	2022-05-22
8	김용욱	(null)	(null)
9	성원용	p01	2022-01-10
10	성원용	p06	2022-02-20
11	오형준	(null)	(null)
12	채광주	p02	2022-04-10

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

- 외부 조인 검색

고객 테이블의 모든 튜플을
검색 대상으로 하기 때문에 결과는 같음

```
SELECT  고객.고객이름, 주문.주문제품, 주문.주문일자  
FROM    고객 LEFT OUTER JOIN  주문 ON  고객.고객아이디 = 주문.주문고객;
```

=

```
SELECT  고객.고객이름, 주문.주문제품, 주문.주문일자  
FROM    주문 RIGHT OUTER JOIN  고객 ON  주문.주문고객 = 고객.고객아이디;
```

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

- 부속 질의문을 이용한 검색
 - SELECT 문 안에 또 다른 SELECT 문을 포함하는 질의
 - › 상위 질의문(주 질의문): 다른 SELECT 문을 포함하는 SELECT 문
 - › 부속 질의문(서브 질의문): 다른 SELECT 문 안에 들어 있는 SELECT 문
 - 괄호로 묶어서 작성, ORDER BY 절을 사용할 수 없음
 - 단일 행 부속 질의문: 하나의 행을 결과로 반환
 - 다중 행 부속 질의문: 하나 이상의 행을 결과로 반환
 - 부속 질의문을 먼저 수행하고, 그 결과를 이용해 상위 질의문을 수행
 - 부속 질의문과 상위 질의문을 연결하는 연산자가 필요
 - › 단일 행 부속 질의문은 비교 연산자(=, <>, >, >=, <, <=) 사용 가능
 - › 다중 행 부속 질의문은 비교 연산자 사용 불가

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

- 부속 질의문을 이용한 검색

제품 테이블

제품번호	제품명	재고량	단가	제조업체
p01	그냥만두	5000	4500	대한식품
p02	매운쫄면	2500	5500	민국푸드
p03	쿵떡파이	3600	2600	한빛제과
p04	맛난초콜릿	1250	2500	한빛제과
p05	얼큰라면	2200	1200	대한식품
p06	통통우동	1000	1550	민국푸드
p07	달콤비스킷	1650	1500	한빛제과

예제 7-40

판매 데이터베이스에서 (달콤비스킷을 생산한 제조업체)가 만든 제품들의 제품명과 단가를 검색해보자.



② { SELECT 제품명, 단가
FROM 제품
WHERE 제조업체 =

상!

(SELECT 제조업체
FROM 제품
WHERE 제품명 = '달콤비스킷');

① select 제품명, 단가
from 제품
where 제조업체

과'

결과 테이블

	제품명	단가
1	쿵떡파이	2600
2	맛난초콜릿	2500
3	달콤비스킷	1500

department — dept_no ①
 dept_name employee First-last
 부서명 → 부서 dept_emp) 직원
 ↳ emp_no

① Finance dept_no \Rightarrow ①

② dept_emp의 emp_no 찾기
 select emp_no from dept_emp

24 25 40

18/2/11

3월 31일 - 12월 2016

44 24

5월 30일

AND
department dept no
= 'doo 2'

441 high \rightarrow like?
% 1101?

2인 join, exist 둘다 사용 가능

Handwritten: $\frac{1}{2} \frac{d}{dt} \left(\frac{1}{2} \frac{d}{dt} \right)$

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

- 부속 질의문을 이용한 검색

예제 7-41

판매 데이터베이스에서 적립금이 가장 많은 고객의 고객이름과 적립금을 검색해보자.

```
▶▶ SELECT  고객이름, 적립금
FROM      고객
WHERE     적립금 = (SELECT MAX(적립금) FROM 고객);
```

결과 테이블

	고객이름	적립금
1	성원용	5000

```
SELECT  고객이름, 적립금
FROM      고객
WHERE     적립금 = 5000 ;
```

최대 적립금은 단일 값이므로 단일 행 부속 질의문
(비교 연산자 = 이용)

고객 테이블

고객아이디	고객이름	나이	등급	직업	적립금
apple	정소화	20	gold	학생	1000
banana	김선우	25	vip	간호사	2500
carrot	고명석	28	gold	교사	4500
orange	김용욱	22	silver	학생	0
melon	성원용	35	gold	회사원	5000
peach	오형준	NULL	silver	의사	300
pear	채광주	31	silver	회사원	500

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

- 부속 질의문을 이용한 검색

표 7-7 다중 행 부속 질의문에 사용 가능한 연산자

연산자	설명
IN	부속 질의문의 결과 값 중 일치하는 것이 있으면 검색 조건이 참
NOT IN	부속 질의문의 결과 값 중 일치하는 것이 없으면 검색 조건이 참
EXISTS	부속 질의문의 결과 값이 하나라도 존재하면 검색 조건이 참
NOT EXISTS	부속 질의문의 결과 값이 하나도 존재하지 않으면 검색 조건이 참
ALL	부속 질의문의 결과 값 모두와 비교한 결과가 참이면 검색 조건을 만족(비교 연산자와 함께 사용)
ANY 또는 SOME	부속 질의문의 결과 값 중 하나라도 비교한 결과가 참이면 검색 조건을 만족(비교 연산자와 함께 사용)

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

- 부속 질의문을 이용한 검색

예제 7-42

판매 데이터베이스에서 banana 고객이 주문한 제품의 제품명과 제조업체를 검색해보자.

```
▶▶ SELECT  제품명, 제조업체
FROM      제품
WHERE     제품번호 IN (SELECT 주문제품
                        FROM 주문
                        WHERE 주문고객 = 'banana');
```

결과 테이블

	제품명	제조업체
1	그냥만두	대한식품
2	맛난초콜릿	한빛제과
3	통통우동	민국푸드

```
SELECT  제품명, 제조업체
FROM      제품
WHERE     제품번호 IN ('p01', 'p04', 'p06')
```

9월 finance 한때

제품 테이블

제품번호	제품명	재고량	단가	제조업체
p01	그냥만두	5000	4500	대한식품
p02	매운쫄면	2500	5500	민국푸드
p03	쿵떡파이	3600	2600	한빛제과
p04	맛난초콜릿	1250	2500	한빛제과
p05	얼큰라면	2200	1200	대한식품
p06	통통우동	1000	1550	민국푸드
p07	달콤비스킷	1650	1500	한빛제과

'banana' 고객이 주문한 제품은 여러 개이므로
→ 다중 행 부속 질의문 (IN 연산자 이용)

주문 테이블

주문번호	주문고객	주문제품	수량	배송지	주문일자
o01	apple	p03	10	서울시 마포구	2022-01-01
o02	melon	p01	5	인천시 계양구	2022-01-10
o03	banana	p06	45	경기도 부천시	2022-01-11
o04	carrot	p02	8	부산시 금정구	2022-02-01
o05	melon	p06	36	경기도 용인시	2022-02-20
o06	banana	p01	19	충청북도 보은군	2022-03-02
o07	apple	p03	22	서울시 영등포구	2022-03-15
o08	pear	p02	50	강원도 춘천시	2022-04-10
o09	banana	p04	15	전라남도 목포시	2022-04-11
o10	carrot	p03	20	경기도 안양시	2022-05-22

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

- 부속 질의문을 이용한 검색

예제 7-43

판매 데이터베이스에서 banana 고객이 주문하지 않은 제품의 제품명과 제조업체를 검색해 보자.

```

▶▶ SELECT  제품명, 제조업체
FROM      제품
WHERE     제품번호 NOT IN (SELECT 주문제품
                           FROM  주문
                           WHERE 주문고객 = 'banana');
    
```

결과 테이블

	제품명	제조업체
1	쿵떡파이	한빛제과
2	매운쫄면	민국푸드
3	얼큰라면	대한식품
4	달콤비스킷	한빛제과

제품 테이블

제품번호	제품명	재고량	단가	제조업체
p01	그냥만두	5000	4500	대한식품
p02	매운쫄면	2500	5500	민국푸드
p03	쿵떡파이	3600	2600	한빛제과
p04	맛난초콜릿	1250	2500	한빛제과
p05	얼큰라면	2200	1200	대한식품
p06	통통우동	1000	1550	민국푸드
p07	달콤비스킷	1650	1500	한빛제과

부속 질의문의 결과 값 중에서
일치하는 것이 없어야 조건이 참이 되는
NOT IN 연산자 이용

주문 테이블

주문번호	주문고객	주문제품	수량	배송지	주문일자
o01	apple	p03	10	서울시 마포구	2022-01-01
o02	melon	p01	5	인천시 계양구	2022-01-10
o03	banana	p06	45	경기도 부천시	2022-01-11
o04	carrot	p02	8	부산시 금정구	2022-02-01
o05	melon	p06	36	경기도 용인시	2022-02-20
o06	banana	p01	19	충청북도 보은군	2022-03-02
o07	apple	p03	22	서울시 영등포구	2022-03-15
o08	pear	p02	50	강원도 춘천시	2022-04-10
o09	banana	p04	15	전라남도 목포시	2022-04-11
o10	carrot	p03	20	경기도 안양시	2022-05-22

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

- 부속 질의문을 이용한 검색

예제 7-44

판매 데이터베이스에서 대한식품이 제조한 모든 제품의 단가보다 비싼 제품의 제품명, 단가,
제조업체를 검색해보자.

```
▶▶ SELECT  제품명, 단가, 제조업체
FROM      제품
WHERE     단가 > ALL ( SELECT 단가
                        FROM      제품
                        WHERE     제조업체 = '대한식품');
```

결과 테이블

	제품명	단가	제조업체
1	매운쫄면	5500	민국푸드

제품 테이블

제품번호	제품명	재고량	단가	제조업체
p01	그냥만두	5000	4500	대한식품
p02	매운쫄면	2500	5500	민국푸드
p03	콩떡파이	3600	2600	한빛제과
p04	맛난초콜릿	1250	2500	한빛제과
p05	얼큰라면	2200	1200	대한식품
p06	통통우동	1000	1550	민국푸드
p07	달콤비스킷	1650	1500	한빛제과

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

- 부속 질의문을 이용한 검색

예제 7-45

판매 데이터베이스에서 2022년 3월 15일에 제품을 주문한 고객의 고객이름을 검색해보자.

```

▶▶ SELECT  고객이름
FROM      고객
WHERE     EXISTS (SELECT *
                  FROM 주문
                  WHERE 주문일자 = '2022-03-15'
                  AND 주문.주문고객 = 고객.고객아이디);
    
```

Handwritten notes:
 select 고객이름
 From 고객
 WHERE select *
 FROM 주문
 EXIST (내부
 AND 주문.주문고객 = 고객.고객아이디);

결과 테이블

고객이름
1 정소화

고객 테이블

고객아이디	고객이름	나이	등급	직업	적립금
apple	정소화	20	gold	학생	1000
banana	김선우	25	vip	간호사	2500
carrot	고명석	28	gold	교사	4500
orange	김용욱	22	silver	학생	0
melon	성원용	35	gold	회사원	5000
peach	오형준	NULL	silver	의사	300
pear	채광주	31	silver	회사원	500

주문 테이블

주문번호	주문고객	주문제품	수량	배송지	주문일자
o01	apple	p03	10	서울시 마포구	2022-01-01
o02	melon	p01	5	인천시 계양구	2022-01-10
o03	banana	p06	45	경기도 부천시	2022-01-11
o04	carrot	p02	8	부산시 금정구	2022-02-01
o05	melon	p06	36	경기도 용인시	2022-02-20
o06	banana	p01	19	충청북도 보은군	2022-03-02
o07	apple	p03	22	서울시 영등포구	2022-03-15
o08	pear	p02	50	강원도 춘천시	2022-04-10
o09	banana	p04	15	전라남도 목포시	2022-04-11
o10	carrot	p03	20	경기도 안양시	2022-05-22

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

- 부속 질의문을 이용한 검색

예제 7-46

판매 데이터베이스에서 2022년 3월 15일에 제품을 주문하지 않은 고객의 고객이름을 검색해보자.

```
▶▶ SELECT  고객이름
FROM      고객
WHERE     NOT EXISTS (SELECT *
                      FROM  주문
                      WHERE  주문일자 = '2022-03-15'
                      AND   주문.주문고객 = 고객.고객아이디);
```

결과 테이블

	고객이름
1	채광주
2	성원용
3	고명석
4	오형준
5	김선우
6	김용욱

고객 테이블

고객아이디	고객이름	나이	등급	직업	적립금
apple	정소화	20	gold	학생	1000
banana	김선우	25	vip	간호사	2500
carrot	고명석	28	gold	교사	4500
orange	김용욱	22	silver	학생	0
melon	성원용	35	gold	회사원	5000
peach	오형준	NULL	silver	의사	300
pear	채광주	31	silver	회사원	500

주문 테이블

주문번호	주문고객	주문제품	수량	배송지	주문일자
o01	apple	p03	10	서울시 마포구	2022-01-01
o02	melon	p01	5	인천시 계양구	2022-01-10
o03	banana	p06	45	경기도 부천시	2022-01-11
o04	carrot	p02	8	부산시 금정구	2022-02-01
o05	melon	p06	36	경기도 용인시	2022-02-20
o06	banana	p01	19	충청북도 보은군	2022-03-02
o07	apple	p03	22	서울시 영등포구	2022-03-15
o08	pear	p02	50	강원도 춘천시	2022-04-10
o09	banana	p04	15	전라남도 목포시	2022-04-11
o10	carrot	p03	20	경기도 안양시	2022-05-22

03 SQL을 이용한 데이터 조작

데이터 검색 : SELECT 문

- 질의 내용은 다양하게 표현 가능하므로 사용자가 자유롭게 선택

① 조인 질의를 이용한 SELECT 문

```
SELECT  제품.제품명, 제품.제조업체
FROM    제품, 주문
WHERE   제품.제품번호 = 주문.주문제품 AND  주문.주문고객 = 'banana';
```

② EXISTS 연산자를 이용한 SELECT 문

```
SELECT  제품명, 제조업체
FROM    제품
WHERE   EXISTS (SELECT  *
                  FROM    주문
                  WHERE   제품.제품번호 = 주문.주문제품
                  AND     주문.주문고객 = 'banana');
```

[예제 7-42]

```
SELECT  제품명, 제조업체
FROM    제품
WHERE   제품번호 IN (SELECT  주문제품
                     FROM    주문
                     WHERE   주문고객 = 'banana');
```

03 SQL을 이용한 데이터 조작

데이터 수정 : UPDATE 문

- 테이블에 저장된 튜플에서 특정 속성의 값을 수정

```
INSERT  
INTO   테이블_이름[(속성_리스트)]  
VALUES (속성값_리스트);
```

- SET 키워드 다음에 속성 값을 어떻게 수정할 것인지를 지정
- WHERE 절에 제시된 조건을 만족하는 튜플에 대해서만 속성 값을 수정
 - WHERE 절을 생략하면 테이블에 존재하는 모든 튜플을 대상으로 수정

03 SQL을 이용한 데이터 조작

데이터 삽입 : INSERT 문

■ 데이터 직접 삽입

예제 7-47

판매 데이터베이스의 고객 테이블에 고객아이디가 strawberry, 고객이름이 최유경, 나이가 30세, 등급이 vip, 직업이 공무원, 적립금이 100원인 새로운 고객의 정보를 삽입해보자. 그런 다음 고객 테이블에 있는 모든 내용을 검색하여 삽입된 새로운 투플을 확인해보자.

▶▶ INSERT

INTO 고객(고객아이디, 고객이름, 나이, 등급, 직업, 적립금)

VALUES ('strawberry', '최유경', 30, 'vip', '공무원', 100);

SELECT * FROM 고객;

고객아이디, 고객이름, 나이, 등급, 직업, 적립금

↑ ↑ ↑ ↑ ↑ ↑

'strawberry', '최유경', 30, 'vip', '공무원', 100

결과 테이블

	고객아이디	고객이름	나이	등급	직업	적립금
1	apple	정소화	20	gold	학생	1000
2	banana	김선우	25	vip	간호사	2500
3	carrot	고명석	28	gold	교사	4500
4	orange	김용욱	22	silver	학생	0
5	melon	성원용	35	gold	회사원	5000
6	peach	오형준	(null)	silver	의사	300
7	pear	채광주	31	silver	회사원	500
8	strawberry	최유경	30	vip	공무원	100

고객 테이블

고객아이디	고객이름	나이	등급	직업	적립금
apple	정소화	20	gold	학생	1000
banana	김선우	25	vip	간호사	2500
carrot	고명석	28	gold	교사	4500
orange	김용욱	22	silver	학생	0
melon	성원용	35	gold	회사원	5000
peach	오형준	NULL	silver	의사	300
pear	채광주	31	silver	회사원	500

03 SQL을 이용한 데이터 조작

데이터 삽입 : INSERT 문

- 데이터 직접 삽입

```
INSERT  
INTO      고객(고객아이디, 고객이름, 나이, 등급, 직업, 적립금)  
VALUES    ('strawberry', '최유경', 30, 'vip', '공무원', 100);
```

=

```
INSERT  
INTO      고객  
VALUES    ('strawberry', '최유경', 30, 'vip', '공무원', 100);
```


03 SQL을 이용한 데이터 조작

데이터 삽입 : INSERT 문

- 데이터 직접 삽입

예제 7-48

판매 데이터베이스의 고객 테이블에 고객아이디가 tomato, 고객이름이 정은심, 나이가 36세, 등급이 gold, 적립금은 4,000원, 직업은 아직 모르는 새로운 고객의 정보를 삽입해보자. 그런 다음 고객 테이블에 있는 모든 내용을 검색하여, 삽입된 정은심 고객의 직업 속성이 널 값인지 확인해보자.

▶▶ INSERT

INTO 고객(고객아이디, 고객이름, 나이, 등급, 적립금)

VALUES ('tomato', '정은심', 36, 'gold', 4000);

SELECT * FROM 고객;

결과 테이블

	고객아이디	고객이름	나이	등급	직업	적립금
1	apple	정소화	20	gold	학생	1000
2	banana	김선우	25	vip	간호사	2500
3	carrot	고명석	28	gold	교사	4500
4	orange	김용욱	22	silver	학생	0
5	melon	성원웅	35	gold	회사원	5000
6	peach	오형준	(null)	silver	의사	300
7	pear	채광주	31	silver	회사원	500
8	strawberry	최유경	30	vip	공무원	100
9	tomato	정은심	36	gold	(null)	4000

직업 속성에 널 값을 삽입

03 SQL을 이용한 데이터 조작

데이터 삽입 : INSERT 문

- 데이터 직접 삽입

```
INSERT  
INTO      고객(고객아이디, 고객이름, 나이, 등급, 적립금)  
VALUES    ('tomato', '정은심', 36, 'gold', 4000);
```

=

```
INSERT  
INTO      고객  
VALUES    ('tomato', '정은심', 36, 'gold', NULL, 4000);
```

03 SQL을 이용한 데이터 조작

데이터 삽입 : INSERT 문

- 부속 질의문을 이용한 데이터 삽입
 - SELECT 문을 이용해 다른 테이블에서 검색한 데이터를 삽입

```
INSERT  
INTO 테이블_이름[(속성_리스트)]  
SELECT 문;
```

한빛제과에서 제조한 제품의 제품명,
재고량, 단가를 제품 테이블에서 검색하여
한빛제품 테이블에 삽입

```
예) INSERT  
    INTO 한빛제품(제품명, 재고량, 단가)  
    SELECT 제품명, 재고량, 단가  
    FROM 제품  
    WHERE 제조업체 = '한빛제과';
```

03 SQL을 이용한 데이터 조작

데이터 수정 : UPDATE 문

- 테이블에 저장된 튜플에서 특정 속성의 값을 수정

```
UPDATE 테이블_이름  
SET      속성_이름1 = 값1, 속성_이름2 = 값2, ...  
[WHERE 조건];
```

3번 inner join

- SET 키워드 다음에 속성 값을 어떻게 수정할 것인지를 지정
- WHERE 절에 제시된 조건을 만족하는 튜플만 속성 값을 수정
 - WHERE 절을 생략하면 테이블에 존재하는 모든 튜플을 대상으로 수정

03 SQL을 이용한 데이터 조작

데이터 수정 : UPDATE 문

예제 7-49

제품 테이블에서 제품번호가 p03인 제품의 제품명을 통큰파이로 수정해보자.

제품 테이블

제품번호	제품명	재고량	단가	제조업체
p01	그냥만두	5000	4500	대한식품
p02	매운짬면	2500	5500	민국푸드
p03	쿵떡파이	3600	2600	한빛제과
p04	맛난초콜릿	1250	2500	한빛제과
p05	얼큰라면	2200	1200	대한식품
p06	통통우동	1000	1550	민국푸드
p07	달콤비스킷	1650	1500	한빛제과

▶▶ UPDATE 제품

SET 제품명 = '통큰파이' 결과 테이블

WHERE 제품번호 = 'p03';

SELECT * FROM 제품;

	제품번호	제품명	재고량	단가	제조업체
1	p01	그냥만두	5000	4500	대한식품
2	p02	매운짬면	2500	5500	민국푸드
3	p03	통큰파이	3600	2600	한빛제과
4	p04	맛난초콜릿	1250	2500	한빛제과
5	p05	얼큰라면	2200	1200	대한식품
6	p06	통통우동	1000	1550	민국푸드
7	p07	달콤비스킷	1650	1500	한빛제과

03 SQL을 이용한 데이터 조작

데이터 수정 : UPDATE 문

예제 7-50

제품 테이블에 있는 모든 제품의 단가를 10% 인상해보자. 그런 다음 제품 테이블의 모든 내용을 검색하여 인상 내용을 확인해보자.

이제부터 UPDATE

▶ UPDATE 제품
 SET 단가 = 단가 * 1.1; 결과 테이블

SELECT * FROM 제품;

	제품번호	제품명	재고량	단가	제조업체
1	p01	그냥만두	5000	4950	대한식품
2	p02	매운짬면	2500	6050	민국푸드
3	p03	통큰파이	3600	2860	한빛제과
4	p04	맛난초콜릿	1250	2750	한빛제과
5	p05	얼큰라면	2200	1320	대한식품
6	p06	통통우동	1000	1705	민국푸드
7	p07	달콤비스킷	1650	1650	한빛제과

제품 테이블

제품번호	제품명	재고량	단가	제조업체
p01	그냥만두	5000	4500	대한식품
p02	매운짬면	2500	5500	민국푸드
p03	쿵떡파이	3600	2600	한빛제과
p04	맛난초콜릿	1250	2500	한빛제과
p05	얼큰라면	2200	1200	대한식품
p06	통통우동	1000	1550	민국푸드
p07	달콤비스킷	1650	1500	한빛제과

select는 원래 있는것만
보여줌

03 SQL을 이용한 데이터 조작

데이터 수정 : UPDATE 문

예제 7-51

판매 데이터베이스에서 정소화 고객이 주문한 제품의 주문수량을 5개로 수정해보자. 그런 다음 주문 테이블의 모든 내용을 검색하여 수정 내용을 확인해보자.

```
▶▶ UPDATE   주문
   SET       수량 = 5
  WHERE      주문고객 IN (SELECT 고객아이디
                           FROM   고객
                           WHERE  고객이름 = '정소화');
```

SELECT * FROM 주문;

결과 테이블

	주문번호	주문고객	주문제품	수량	배송지	주문일자
1	o01	apple	p03	5	서울시 마포구	22/01/01
2	o02	melon	p01	5	인천시 계양구	22/01/10
3	o03	banana	p06	45	경기도 부천시	22/01/11
4	o04	carrot	p02	8	부산시 금정구	22/02/01
5	o05	melon	p06	36	경기도 용인시	22/02/20
6	o06	banana	p01	19	충청북도 보은군	22/03/02
7	o07	apple	p03	5	서울시 영등포구	22/03/15
8	o08	pear	p02	50	강원도 춘천시	22/04/10
9	o09	banana	p04	15	전라남도 목포시	22/04/11
10	o10	carrot	p03	20	경기도 안양시	22/05/22

부속 질의문을 이용한 UPDATE 문

주문 테이블

주문번호	주문고객	주문제품	수량	배송지	주문일자
o01	apple	p03	10	서울시 마포구	2022-01-01
o02	melon	p01	5	인천시 계양구	2022-01-10
o03	banana	p06	45	경기도 부천시	2022-01-11
o04	carrot	p02	8	부산시 금정구	2022-02-01
o05	melon	p06	36	경기도 용인시	2022-02-20
o06	banana	p01	19	충청북도 보은군	2022-03-02
o07	apple	p03	22	서울시 영등포구	2022-03-15
o08	pear	p02	50	강원도 춘천시	2022-04-10
o09	banana	p04	15	전라남도 목포시	2022-04-11
o10	carrot	p03	20	경기도 안양시	2022-05-22

03 SQL을 이용한 데이터 조작

데이터 삭제 : DELETE 문

- 테이블에 저장된 데이터를 삭제

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

- WHERE 절에 제시한 조건을 만족하는 튜플만 삭제
 - WHERE 절을 생략하면 테이블에 존재하는 모든 튜플을 삭제해 빈 테이블이 됨

03 SQL을 이용한 데이터 조작

데이터 삭제 : DELETE 문

예제 7-52

주문 테이블에서 주문일자가 2022년 5월 22일인 주문 내역을 삭제해보자. 그런 다음 주문 테이블의 모든 내용을 검색하여 삭제 여부를 확인해보자.

▶▶ DELETE

```
FROM 주문
WHERE 주문일자 = '2022-05-22';
```

```
SELECT * FROM 주문;
```

결과 테이블

	주문번호	주문고객	주문제품	수량	배송지	주문일자
1	o01	apple	p03	5	서울시 마포구	22/01/01
2	o02	melon	p01	5	인천시 계양구	22/01/10
3	o03	banana	p06	45	경기도 부천시	22/01/11
4	o04	carrot	p02	8	부산시 금정구	22/02/01
5	o05	melon	p06	36	경기도 용인시	22/02/20
6	o06	banana	p01	19	충청북도 보은군	22/03/02
7	o07	apple	p03	5	서울시 영등포구	22/03/15
8	o08	pear	p02	50	강원도 춘천시	22/04/10
9	o09	banana	p04	15	전라남도 목포시	22/04/11

주문 테이블

주문번호	주문고객	주문제품	수량	배송지	주문일자
o01	apple	p03	10	서울시 마포구	2022-01-01
o02	melon	p01	5	인천시 계양구	2022-01-10
o03	banana	p06	45	경기도 부천시	2022-01-11
o04	carrot	p02	8	부산시 금정구	2022-02-01
o05	melon	p06	36	경기도 용인시	2022-02-20
o06	banana	p01	19	충청북도 보은군	2022-03-02
o07	apple	p03	22	서울시 영등포구	2022-03-15
o08	pear	p02	50	강원도 춘천시	2022-04-10
o09	banana	p04	15	전라남도 목포시	2022-04-11
o10	carrot	p03	20	경기도 안양시	2022-05-22

03 SQL을 이용한 데이터 조작

데이터 삭제 : DELETE 문

예제 7-53

판매 데이터베이스에서 정소화 고객이 주문한 내역을 주문 테이블에서 삭제해보자. 그런 다음 주문 테이블의 모든 내용을 검색하여 삭제 여부를 확인해보자.

▶▶ DELETE

```
FROM      주문
WHERE      주문고객 IN (SELECT 고객아이디
                        FROM      고객
                        WHERE      고객이름 = '정소화');
```

부속 질의문을 이용한 DELETE 문

SELECT * FROM 주문;

결과 테이블

	주문번호	주문고객	주문제품	수량	배송지	주문일자
1	o02	melon	p01	5	인천시 계양구	22/01/10
2	o03	banana	p06	45	경기도 부천시	22/01/11
3	o04	carrot	p02	8	부산시 금정구	22/02/01
4	o05	melon	p06	36	경기도 용인시	22/02/20
5	o06	banana	p01	19	충청북도 보은군	22/03/02
6	o08	pear	p02	50	강원도 춘천시	22/04/10
7	o09	banana	p04	15	전라남도 목포시	22/04/11

03 SQL을 이용한 데이터 조작

데이터 삭제 : DELETE 문

예제 7-54

판매 데이터베이스의 주문 테이블에 존재하는 모든 튜플을 삭제해보자. 그런 다음 주문 테이블의 모든 내용을 검색하여 삭제 여부를 확인해보자.

```
▶▶ DELETE
FROM    주문;

SELECT * FROM 주문;

결과 테이블
```

주문번호	주문고객	주문제품	수량	배송지
------	------	------	----	-----

빈 테이블이 남음, DROP TABLE과는 다름

주문 테이블

주문번호	주문고객	주문제품	수량	배송지	주문일자
o01	apple	p03	10	서울시 마포구	2022-01-01
o02	melon	p01	5	인천시 계양구	2022-01-10
o03	banana	p06	45	경기도 부천시	2022-01-11
o04	carrot	p02	8	부산시 금정구	2022-02-01
o05	melon	p06	36	경기도 용인시	2022-02-20
o06	banana	p01	19	충청북도 보은군	2022-03-02
o07	apple	p03	22	서울시 영등포구	2022-03-15
o08	pear	p02	50	강원도 춘천시	2022-04-10
o09	banana	p04	15	전라남도 목포시	2022-04-11
o10	carrot	p03	20	경기도 안양시	2022-05-22

그림 7-6 데이터 조작 예제에서 사용하는 판매 데이터베이스 : 고객, 제품, 주문 테이블