

CHAPTER 07

SQL 연습문제 풀이

문제 → 정답 형식 · 객관식 22문항 + SQL 작성 8문항

데이터베이스 개론 · IT CookBook



SQL — 데이터 정의·조작·제어

DDL · 정의어

Data Definition

CREATE
ALTER
DROP

스키마(구조)를 정의·변경·삭제

DML · 조작어

Data Manipulation

SELECT INSERT
UPDATE DELETE

데이터를 검색·삽입·수정·삭제

DCL · 제어어

Data Control

GRANT REVOKE
COMMIT ROLLBACK

권한 부여·취소, 트랜잭션 제어

검색 핵심 절 WHERE · GROUP BY · HAVING · ORDER BY · JOIN · 서브쿼리 · LIKE · IS NULL · 집계함수
(COUNT·SUM·AVG·MIN·MAX)

01

객관식

Multiple Choice · Q1–Q22



DDL과 DML 구분

1. SQL 명령어 중 성격이 다른 것은?

- ① ALTER ② DROP ③ CREATE ④ INSERT

2. 고객 테이블 생성 후 누락된 주소 속성을 추가하는 명령어는?

- ① CREATE ② ALTER ③ ADD ④ MODIFY

DDL과 DML 구분

1. SQL 명령어 중 성격이 다른 것은?

- ① ALTER ② DROP ③ CREATE ④ INSERT

정답 ④

ALTER·DROP·CREATE는 DDL, INSERT는 DML이다.

2. 고객 테이블 생성 후 누락된 주소 속성을 추가하는 명령어는?

- ① CREATE ② ALTER ③ ADD ④ MODIFY

정답 ②

ALTER TABLE ... ADD 로 속성을 추가한다.

제약조건 키워드

3. 학생 테이블을 참조 제약과 함께 제거할 때 빈칸은? (DROP TABLE 학생 __ ;)

- ① ALL ② CASCADE ③ RESTRICT ④ DELETE

4. 속성의 값 범위·제약조건을 지정해 무결성을 유지하는 키워드는?

- ① PRIMARY KEY ② DEFAULT ③ NOT NULL ④ CHECK

제약조건 키워드

3. 학생 테이블을 참조 제약과 함께 제거할 때 빈칸은? (DROP TABLE 학생 __ ;)

- ① ALL ② **CASCADE** ③ RESTRICT ④ DELETE

정답 ②

CASCADE는 참조하는 제약·객체까지 함께 제거한다.

4. 속성의 값 범위·제약조건을 지정해 무결성을 유지하는 키워드는?

- ① PRIMARY KEY ② DEFAULT ③ NOT NULL ④ **CHECK**

정답 ④

CHECK (조건) 으로 허용 값의 범위를 제한한다.

외래키와 집계함수

5. 외래키를 지정하여 참조 무결성과 관련되는 키워드는?

- ① PRIMARY KEY ② DEFAULT ③ NOT NULL ④ FOREIGN KEY

6. 어떤 데이터 타입에도 사용할 수 있는 집계함수는?

- ① AVG ② COUNT ③ SUM ④ STDDEV

외래키와 집계함수

5. 외래키를 지정하여 참조 무결성과 관련되는 키워드는?

- ① PRIMARY KEY ② DEFAULT ③ NOT NULL ④ FOREIGN KEY

정답 ④

FOREIGN KEY ... REFERENCES 로 외래키를 정의한다.

6. 어떤 데이터 타입에도 사용할 수 있는 집계함수는?

- ① AVG ② COUNT ③ SUM ④ STDDEV

정답 ②

AVG·SUM·STDDEV는 숫자형만, COUNT는 모든 타입에 사용 가능.

LIKE 패턴과 HAVING

7. WHERE 고객아이디 LIKE '_T%' 의 의미는?

- ① T로 시작 ② T로 끝남 ③ T를 포함 ④ 두 번째 문자가 T

8. HAVING 키워드를 함께 사용할 수 있는 절은?

- ① LIKE 절 ② WHERE 절 ③ GROUP BY 절 ④ ORDER BY 절

LIKE 패턴과 HAVING

7. WHERE 고객아이디 LIKE '_T%' 의 의미는?

- ① T로 시작 ② T로 끝남 ③ T를 포함 ④ 두 번째 문자가 T

정답 ④ _ 는 임의의 한 문자, % 는 0개 이상 → 두 번째 글자가 T.

8. HAVING 키워드를 함께 사용할 수 있는 절은?

- ① LIKE 절 ② WHERE 절 ③ GROUP BY 절 ④ ORDER BY 절

정답 ③ HAVING은 GROUP BY로 묶은 그룹에 조건을 건다.

NULL 비교와 DELETE

9. 주소가 널이 아닌 고객을 검색하는 WHERE 조건은?

- ① 주소 != NULL ② 주소 <> NULL ③ 주소 IS NOT NULL ④ NOT(주소=NULL)

10. DELETE 명령문에 대한 설명으로 옳지 않은 것은?

- ① 튜플 삭제에 사용 ② WHERE 없으면 DROP TABLE과 동일 ③ DML에 해당 ④ DELETE FROM 테이블 [WHERE 조건] 형식

NULL 비교와 DELETE

9. 주소가 널이 아닌 고객을 검색하는 WHERE 조건은?

- ① 주소 != NULL ② 주소 <> NULL ③ 주소 IS NOT NULL ④ NOT(주소=NULL)

정답 ③

NULL 비교는 != 가 아니라 IS (NOT) NULL 로 한다.

10. DELETE 명령문에 대한 설명으로 옳지 않은 것은?

- ① 튜플 삭제에 사용 ② WHERE 없으면 DROP TABLE과 동일 ③ DML에 해당 ④ DELETE FROM 테이블 [WHERE 조건] 형식

정답 ②

DELETE는 행만 지우고 테이블 구조는 남으므로 DROP TABLE과 다르다.

11. 올바른 INSERT 문 고르기

제품(제품코드, 제품명, 가격, 제조일자) 테이블에 (100, '깨끗세제', 5000) 을 삽입하는 옳은 SQL은?

① INSERT 제품 INTO VALUES (100, '깨끗세제', 5000);

② INSERT FROM 제품 VALUES (100, '깨끗세제', 5000);

③ INSERT INTO 제품(제품코드, 제품명, 가격) VALUES (100, '깨끗세제', 5000);

④ INSERT TO 제품(제품코드, 제품명, 가격) VALUES (100, '깨끗세제', 5000);

11. 올바른 INSERT 문 고르기

제품(제품코드, 제품명, 가격, 제조일자) 테이블에 (100, '깨끗세제', 5000) 을 삽입하는 옳은 SQL은?

① INSERT 제품 INTO VALUES (100, '깨끗세제', 5000);

② INSERT FROM 제품 VALUES (100, '깨끗세제', 5000);

③ INSERT INTO 제품(제품코드, 제품명, 가격) VALUES (100, '깨끗세제', 5000);

정답

④ INSERT TO 제품(제품코드, 제품명, 가격) VALUES (100, '깨끗세제', 5000);

형식: INSERT INTO 테이블(열 목록) VALUES (값 목록);

UPDATE 절과 INNER JOIN

12. UPDATE 학생 ① 점수=점수+5 ② 성명='홍길동'; 에서 ①, ②는?

13. WHERE 조인을 'FROM 학생 ① 학과 ② ...' 형태로 바꿀 때 ①, ②는?

UPDATE 절과 INNER JOIN

12. UPDATE 학생 ① 점수=점수+5 ② 성명='홍길동'; 에서 ①, ②는?

① SET · ② WHERE

UPDATE 테이블 SET 열=값 WHERE 조건 — 값은 SET, 대상 행은 WHERE.

13. WHERE 조인을 'FROM 학생 ① 학과 ② ...' 형태로 바꿀 때 ①, ②는?

① INNER JOIN · ② ON

WHERE의 조인 조건을 INNER JOIN ... ON 으로 옮긴 동일 질의.

14. 부속 질의(subquery) 결과

〈사원 테이블〉

사원번호	성명	소속부서
25	김미순	총무과
56	박이준	자재과
23	이형주	자재과
43	오형우	총무과

SQL

```
SELECT 성명
FROM 사원
WHERE 소속부서 =
      (SELECT 소속부서 FROM 사원
       WHERE 성명='오형우');
```

→ 이 SQL의 결과 테이블 모습은? 보기 ①~④ 중 고르시오.

14. 부속 질의(subquery) 결과

〈사원 테이블〉

사원번호	성명	소속부서
25	김미순	총무과
56	박이준	자재과
23	이형주	자재과
43	오형우	총무과

SQL

```
SELECT 성명
FROM 사원
WHERE 소속부서 =
      (SELECT 소속부서 FROM 사원
       WHERE 성명='오형우');
```

정답 ①

부속 질의가 '오형우'의 소속부서 = 총무과 를 먼저 구한 뒤,
총무과 사원의 성명을 검색한다.

결과 → 김미순, 오형우

NOT EXISTS와 FK 옵션

15. NOT EXISTS (SELECT * FROM 가입 WHERE 학생.학번=가입.학번) 의 결과는?

- ① 모든 동아리 가입 학생 ② 어떤 동아리도 가입 안 한 학생 ③ 없는 동아리에 가입한 학생 ④ 아무 동아리나 가입한 학생

16. FOREIGN KEY 정의 시 ON DELETE/UPDATE 옵션으로 선택할 수 없는 것은?

- ① NO ACTION ② CHECK ③ CASCADE ④ SET NULL

NOT EXISTS와 FK 옵션

15. NOT EXISTS (SELECT * FROM 가입 WHERE 학생.학번=가입.학번) 의 결과는?

- ① 모든 동아리 가입 학생 ② 어떤 동아리도 가입 안 한 학생 ③ 없는 동아리에 가입한 학생 ④ 아무 동아리나 가입한 학생

정답 ②

가입에 대응 튜플이 '없는' 학생 → 어떤 동아리에도 가입하지 않은 학생.

16. FOREIGN KEY 정의 시 ON DELETE/UPDATE 옵션으로 선택할 수 없는 것은?

- ① NO ACTION ② CHECK ③ CASCADE ④ SET NULL

정답 ②

옵션은 NO ACTION·CASCADE·SET NULL·SET DEFAULT — CHECK는 옵션이 아니다.

뷰 삭제와 뷰의 특성

17. 뷰 V1로 정의한 V2가 있을 때 V1을 CASCADE로 삭제하면?

- ① V1만 삭제 ② V2만 삭제 ③ V1·V2 모두 삭제 ④ 둘 다 삭제 안 됨

18. 뷰에 대한 설명으로 옳지 않은 것은?

- ① 물리적 저장 없는 가상 테이블 ② 삽입·삭제·수정이 항상 허용되진 않음 ③ 검색은 기본 테이블과 약간 차이
④ ALTER로 정의 변경 불가

뷰 삭제와 뷰의 특성

17. 뷰 V1로 정의한 V2가 있을 때 V1을 CASCADE로 삭제하면?

- ① V1만 삭제 ② V2만 삭제 ③ **V1·V2 모두 삭제** ④ 둘 다 삭제 안 됨

정답 ③

CASCADE 삭제는 그 뷰에 의존하는 뷰(V2)까지 연쇄 삭제한다.

18. 뷰에 대한 설명으로 옳지 않은 것은?

- ① 물리적 저장 없는 가상 테이블 ② 삽입·삭제·수정이 항상 허용되진 않음 ③ **검색은 기본 테이블과 약간 차이**
④ ALTER로 정의 변경 불가

정답 ③

뷰의 검색(SELECT)은 기본 테이블과 동일하므로 ③이 틀림.

뷰의 독립성

19. 뷰에 대한 설명으로 옳지 않은 것은?

- ① 데이터 보안 제공 ② 검색은 SELECT 사용 ③ WITH CHECK OPTION으로 제약 ④ 물리적 독립성 제공

뷰의 독립성

정답

19. 뷰에 대한 설명으로 옳지 않은 것은?

- ① 데이터 보안 제공 ② 검색은 SELECT 사용 ③ WITH CHECK OPTION으로 제약 ④ 물리적 독립성 제공

정답 ④

뷰는 '논리적' 데이터 독립성을 제공하므로 ④가 틀림.

20. DDL과 DML 분류

다음 SQL 명령어를 DDL과 DML로 분류하시오.

① CREATE ② SELECT ③ DROP ④ ALTER ⑤ DELETE ⑥ INSERT ⑦ UPDATE

정의어(DDL)와 조작어(DML)로 나누어 보세요.

20. DDL과 DML 분류

다음 SQL 명령어를 DDL과 DML로 분류하시오.

① CREATE ② SELECT ③ DROP ④ ALTER ⑤ DELETE ⑥ INSERT ⑦ UPDATE

DDL ① CREATE · ③ DROP · ④ ALTER (스키마 정의·변경·삭제)

DML ② SELECT · ⑤ DELETE · ⑥ INSERT · ⑦ UPDATE (데이터 조작)

21. 뷰에 대한 옳은 설명 모두 고르기

- (A) 뷰는 물리적으로 저장 장치에 저장된다.
- (B) 삽입·수정·삭제 연산이 항상 허용되는 것은 아니다.
- (C) CREATE 문을 이용해 생성할 수 있다.
- (D) ALTER 문을 이용해 변경할 수 있다.
- (E) DROP 문을 이용해 삭제할 수 있다.
- (F) 기본 테이블이 제거되어도 뷰는 자동 삭제되지 않는다.
- (G) 뷰를 기반으로 새로운 뷰를 만들 수 있다.

21. 뷰에 대한 옳은 설명 모두 고르기

(A) 뷰는 물리적으로 저장 장치에 저장된다.

(B) 삽입·수정·삭제 연산이 항상 허용되는 것은 아니다.

(C) CREATE 문을 이용해 생성할 수 있다.

(D) ALTER 문을 이용해 변경할 수 있다.

(E) DROP 문을 이용해 삭제할 수 있다.

(F) 기본 테이블이 제거되어도 뷰는 자동 삭제되지 않는다.

(G) 뷰를 기반으로 새로운 뷰를 만들 수 있다.

정답 (B)(C)(E)(G)

CREATE로 생성·DROP로 삭제, 삽입/수정/삭제에 제약, 뷰 기반 뷰 생성 가능. (A·D·F는 부적합)

삽입 SQL의 특성

22. 삽입 SQL(embedded SQL)에 대한 설명으로 옳지 않은 것은?

- ① 프로그램 안에 삽입하는 SQL ② 명령문 위치 어디든 삽입 가능 ③ 호스트 변수는 속성과 이름이 달라야 함
- ④ 변수와 속성의 타입이 일치해야 함

삽입 SQL의 특성

22. 삽입 SQL(embedded SQL)에 대한 설명으로 옳지 않은 것은?

- ① 프로그램 안에 삽입하는 SQL ② 명령문 위치 어디든 삽입 가능 ③ 호스트 변수는 속성과 이름이 달라야 함
- ④ 변수와 속성의 타입이 일치해야 함

정답 ③

호스트 변수는 이름이 같아도 되며 콜론(:)으로 구별 → ③이 틀림.

02

단답·SQL 작성

Short Answer & SQL Writing · Q23-Q30



WITH CHECK OPTION & 관계대수 → SQL

23 뷰 생성 시 WITH CHECK OPTION의 의미를 설명하시오.

24 관계대수 $\pi(\text{도서명, 가격})(\sigma(\text{출판사}=\text{'한빛'} \wedge \text{재고량} \geq 50)(\text{도서}))$ 를 SQL로 작성하시오.

WITH CHECK OPTION & 관계대수 → SQL

23 WITH CHECK OPTION 의 의미

뷰에 대한 삽입·수정 연산이 실행될 때, 뷰의 정의 조건을 위반하면 그 연산을 거부하도록 지시한다.

24 관계대수 → SQL

SQL

SELECT 도서명, 가격

FROM 도서

WHERE 출판사='한빛' AND 재고량>=50;

SELECT 실행 결과의 튜플 수

제품 테이블 대한식품 50개 · 민국푸드 30개 · 한빛제과 50개 (전체 130행)

다음 각 SQL의 결과 튜플 수는?

가

```
SELECT 제조업체 FROM 제품;
```

?

나

```
SELECT DISTINCT 제조업체 FROM 제품;
```

?

다

```
SELECT 제품명 FROM 제품 WHERE 제조업체='대한식품';
```

?

SELECT 실행 결과의 튜플 수

제품 테이블 대한식품 50개 · 민국푸드 30개 · 한빛제과 50개 (전체 130행)

다음 각 SQL의 결과 튜플 수는?

가

```
SELECT 제조업체 FROM 제품;
```

130

나

```
SELECT DISTINCT 제조업체 FROM 제품;
```

3

다

```
SELECT 제품명 FROM 제품 WHERE 제조업체='대한식품';
```

50

DISTINCT는 중복을 제거하므로 제조업체 종류 수(3)만 반환한다.

테이블 생성과 카디널리티

SETUP

```
CREATE TABLE 부서 ( 부서코드 CHAR(3) PRIMARY KEY, 부서명 VARCHAR(10) );  
CREATE TABLE 사원 ( 사원번호 INT PRIMARY KEY, 소속부서 VARCHAR(10),  
    FOREIGN KEY(소속부서) REFERENCES 부서(부서코드) ON DELETE CASCADE );  
INSERT 부서 → (P1, 개발부), (P2, 홍보부)  
INSERT 사원 → (100, P1), (200, P2), (300, P2)
```

- (1) 위 INSERT 직후 부서·사원 테이블의 카디널리티는 각각?
- (2) DELETE FROM 부서 WHERE 부서코드='P2'; 실행 후 두 테이블의 카디널리티는?

테이블 생성과 카디널리티

(1) 생성·삽입 직후

부서 테이블 = 2 사원 테이블 = 3 (INSERT한 행 수 그대로)

(2) DELETE FROM 부서 WHERE 부서코드='P2'; 실행 후

부서 테이블 = 1 사원 테이블 = 1

ON DELETE CASCADE → 홍보부(P2)를 지우면 P2 소속 사원(200-300)도 함께 삭제된다.

학생 테이블 — 문제

〈학생 테이블〉

학번	이름	학점	학과
101	이진아	3.5	컴퓨터공학과
102	양기섭	4.1	컴퓨터공학과
103	박수정	3.7	호텔관광경영학과
104	홍민호	2.8	호텔관광경영학과
105	오연주	3.2	호텔관광경영학과
106	김우리	2.5	건축학과
107	조유근	4.3	건축학과

- (1) 학과별 학생 수가 2명을 초과하는 학과와 그 인원은?
- (2) 최대 20자 가변 문자열 '연락처' 속성을 추가하시오.
- (3) ORDER BY 학과 DESC, 학점 ASC 시 가장 먼저 출력되는 학생은?
- (4) 김우리와 같은 학과 학생들의 평균 학점('평균 학점')은?
- (5) 학번 105 학생의 학점을 4.5, 학과를 전자과로 변경하시오.

학생 테이블 — 정답 (1)~(3)

(1) GROUP BY 학과 HAVING count(*) > 2

학과	결과
호텔관광경영학과	3

3명인 호텔관광경영학과만 조건(>2)을 만족.

(2) 연락처 속성 추가

```
ALTER TABLE 학생 ADD 연락처 VARCHAR(20);
```

(3) ORDER BY 학과 DESC, 학점 ASC — 첫 행

홍민호 (학과 내림차순 → 호텔관광경영학과 먼저, 그 중 학점 오름차순 최저 2.8)

학생 테이블 — 정답 (4)~(5)

(4) 김우리와 같은 학과 학생들의 평균 학점

SQL

```
SELECT AVG(학점) AS "평균 학점"  
FROM 학생  
WHERE 학과 = (SELECT 학과 FROM 학생 WHERE 이름='김우리');
```

(5) 학번 105 학생의 학점·학과 변경

SQL

```
UPDATE 학생  
SET 학점=4.5, 학과='전자과'  
WHERE 학번=105;
```

환자·의사 테이블 — 문제

〈환자〉

환자번호	환자이름	나이	담당의사
P001	오우진	31	D002
P002	채광주	50	D001
P003	김용욱	43	D003

〈의사〉

의사번호	의사이름	소속	근무연수
D001	정지영	내과	5
D002	김선주	피부과	10
D003	정성호	정형외과	15

- (1) 환자 테이블 생성 — 환자이름 NOT NULL, 담당의사는 의사(의사번호) 참조 FK.
- (2) 의사 테이블 생성 — 소속 기본값 '내과', 근무연수 1~40 CHECK.
- (3) D001 의사가 담당하고 나이 30세 이상인 환자의 번호·이름.
- (4) 소속별 의사 수와 평균 근무연수.
- (5) 김용욱 환자를 담당하는 의사의 이름·소속·근무연수.

환자·의사 — 정답 (1)~(2) 생성

(1) 환자 테이블

```
CREATE TABLE 환자 (  
  환자번호 CHAR(4) NOT NULL,  
  환자이름 VARCHAR(10) NOT NULL,  
  나이 INT,  
  담당의사 CHAR(4),  
  PRIMARY KEY(환자번호),  
  FOREIGN KEY(담당의사)  
    REFERENCES 의사(의사번호)  
);
```

(2) 의사 테이블

```
CREATE TABLE 의사 (  
  의사번호 CHAR(4) NOT NULL,  
  의사이름 VARCHAR(10),  
  소속 VARCHAR(20) DEFAULT '내과',  
  근무연수 INT,  
  PRIMARY KEY(의사번호),  
  CHECK (근무연수 >= 1  
        AND 근무연수 <= 40)  
);
```

환자·의사 — 정답 (3)~(5) 검색

(3) D001 담당 & 나이 ≥ 30 환자

```
SELECT 환자번호, 환자이름  
FROM 환자 WHERE 담당의사='D001' AND 나이>=30;
```

(4) 소속별 의사 수·평균 근무연수

```
SELECT 소속, COUNT(*) AS "의사 수", AVG(근무연수) AS "평균 근무연수"  
FROM 의사 GROUP BY 소속;
```

(5) 김용욱 담당 의사의 이름·소속·근무연수

```
SELECT 의사.의사이름, 의사.소속, 의사.근무연수 FROM 의사, 환자 WHERE 환자.환자이름='김용욱' AND 의사.의사번호=환자.담당의사;
```

학생·과목·수강 — 문제

SCHEMA

학생(학번, 이름, 학년)

과목(과목번호, 과목이름)

수강(학번, 과목번호, 중간성적, 기말성적, 학점)

- (1) 과목번호 'A%' & 기말성적 ≥ 80 학생의 이름·중간성적 (이름 ASC, 중간성적 DESC).
- (2) 3명 이상 수강한 과목별 학생수·기말성적 평균 ('학생수' / '성적평균').
- (3) 개설된 과목의 수 (중복 제외).
- (4) A003을 수강하지 않는 학생의 이름·학년 — NOT IN.
- (5) 같은 조건 — NOT EXISTS.

학생.과목.수강 — 정답 (1)~(2)

(1) 조인 + 정렬

```
SELECT 학생.이름, 수강.중간성적  
FROM 수강, 학생  
WHERE 수강.과목번호 LIKE 'A%' AND 수강.기말성적>=80 AND 학생.학번=수강.학번  
ORDER BY 학생.이름 ASC, 수강.중간성적 DESC;
```

(2) 그룹별 집계 + HAVING

```
SELECT 과목번호, COUNT(*) AS 학생수, AVG(기말성적) AS 성적평균  
FROM 수강  
GROUP BY 과목번호 HAVING COUNT(*)>=3;
```

학생.과목.수강 — 정답 (3)~(5)

(3) 개설 과목 수 (DISTINCT)

```
SELECT COUNT(DISTINCT 과목번호) FROM 수강;
```

(4) A003 미수강 — NOT IN

```
SELECT 이름, 학년  
FROM 학생 WHERE 학번 NOT IN (SELECT 학번 FROM 수강 WHERE 과목번호='A003');
```

(5) A003 미수강 — NOT EXISTS

```
SELECT 이름, 학년  
FROM 학생 WHERE NOT EXISTS (SELECT * FROM 수강 WHERE 과목번호='A003' AND 학생.학번=수강.학번);
```

고객·판매자·제품·주문 — 문제

SCHEMA

고객(고객번호, 고객이름, 거주도시, 할인율)

판매자(판매자번호, 판매자이름, 수수료, 판매지역)

제품(제품번호, 제품명, 재고량, 가격)

주문(주문번호, 고객번호, 제품번호, 판매자번호, 주문수량)

- (1) 고객 (C005, 채희성, 대구, 할인율 미정) 삽입.
- (2) 방지호 고객의 주문수량을 20% 증가.
- (3) 재고량 50 미만 제품 삭제.
- (4) 가격이 최저가인 제품명 (중복 없이).
- (5) 제품명에 '부'가 포함된 제품을 주문한 고객의 이름.
- (6) 주문수량 합계 300 이상 고객별 주문횟수·주문수량 합계.
- (7) 서울 지역 모든 판매자보다 수수료가 비싼 판매자의 이름·수수료·판매지역.

정답 (1)~(3)

(1) 고객 삽입 (할인을 NULL)

```
INSERT INTO 고객 VALUES ('C005', '채희성', '대구', NULL);
```

(2) 방지호 주문수량 20% 증가

```
UPDATE 주문 SET 주문수량=주문수량*1.2  
WHERE 고객번호 IN (SELECT 고객번호 FROM 고객 WHERE 고객이름='방지호');
```

(3) 재고량 50 미만 제품 삭제

```
DELETE FROM 제품 WHERE 재고량<50;
```

정답 (4)~(5)

(4) 최저가 제품명 (중복 없이)

SQL

```
SELECT DISTINCT 제품명  
FROM 제품  
WHERE 가격 = (SELECT MIN(가격) FROM 제품);
```

(5) 제품명에 '부' 포함 제품을 주문한 고객 이름

SQL

```
SELECT 고객.고객이름  
FROM 고객, 제품, 주문  
WHERE 제품.제품명 LIKE '%부%' AND 고객.고객번호=주문.고객번호 AND 제품.제품번호=주문.제품번호;
```

정답 (6)~(7)

(6) 주문수량 합계 ≥ 300 고객별 주문횟수·합계

SQL

```
SELECT 고객번호, COUNT(*) AS 주문횟수, SUM(주문수량) AS 주문수량합계
FROM 주문
GROUP BY 고객번호 HAVING SUM(주문수량) >= 300;
```

(7) 서울 판매자 전원보다 수수료 큰 판매자

SQL

```
SELECT 판매자이름, 수수료, 판매지역
FROM 판매자
WHERE 수수료 > ALL (SELECT 수수료 FROM 판매자 WHERE 판매지역 = '서울');
```

7장 SQL 핵심 정리

정의 · 제약 (DDL)

- CREATE / ALTER / DROP
- PRIMARY KEY · FOREIGN KEY
- CHECK · DEFAULT · NOT NULL
- ON DELETE CASCADE

검색 · 조작 (DML)

- SELECT 4절: WHERE·GROUP BY·HAVING·ORDER BY
- JOIN(INNER/OUTER) · 부속질의
- IN · EXISTS · ALL · LIKE
- 집계: COUNT·SUM·AVG·MIN·MAX

뷰 · 권한

- CREATE / DROP VIEW
- WITH CHECK OPTION
- 논리적 독립성 · 보안 제공
- GRANT · REVOKE (DCL)

한 줄 요약 구조는 DDL, 데이터는 DML, 권한·트랜잭션은 DCL — NULL 비교는 IS NULL, 그룹 조건은 HAVING을 잊지 말 것.

수고하셨습니다

End of Chapter 7 — SQL

데이터베이스 개론 · 7장 SQL 연습문제 (문제 → 정답)

