



## 2020년 2회 정보처리기사 실기 시험 100% 합격전략집

1  
일차

2  
일차

3  
일차

4  
일차

5  
일차

6  
일차

7  
일차

8  
일차

9  
일차

10  
일차

11  
일차

12  
일차

13  
일차

14  
일차

15  
일차

16  
일차

17  
일차

18  
일차

19  
일차

20  
일차

### 8장 SQL 응용

핵심 104 DDL – CREATE TABLE

핵심 105 DDL – CREATE VIEW

핵심 106 DDL – ALTER TABLE

핵심 107 DDL – DROP TABLE

핵심 108 DCL – GRANT / REVOKE

핵심 109 COMMIT / ROLLBACK / SAVEPOINT

핵심 110 DML – 삽입, 삭제, 갱신문

핵심 111 DML – SELECT

핵심 112 하위 질의



## 2020년 2회 정보처리기사 실기 대비용 핵심요약

### 8장 | SQL 응용

#### [핵심104] DDL – CREATE TABLE

##### DDL(Data Define Language, 데이터 정의어)

- DB 구조, 데이터 형식, 접근 방식 등 DB를 구축하거나 수정할 목적으로 사용하는 언어이다.
- DDL은 번역한 결과가 데이터 사전(Data Dictionary)이라는 특별한 파일에 여러 개의 테이블로서 저장된다.
- DDL에는 CREATE SCHEMA, CREATE DOMAIN, CREATE TABLE, CREATE VIEW, CREATE INDEX, ALTER TABLE, DROP 등이 있다.

##### CREATE TABLE

- 테이블을 정의하는 명령문이다.
- 표기 형식

```
CREATE TABLE 테이블명
(속성명 데이터_타입 [DEFAULT 기본값] [NOT NULL], ...
[ PRIMARY KEY(기본키_속성명, ...)]
[ UNIQUE(대체키_속성명, ...)]
[ FOREIGN KEY(외래키_속성명, ...)
[REFERENCES 참조테이블(기본키_속성명, ...)
[ON DELETE 옵션]
[ON UPDATE 옵션]
[ CONSTRAINT 제약조건명] [CHECK (조건식)];
```

- PRIMARY KEY : 기본키로 사용할 속성 또는 속성의 집합을 지정함
- UNIQUE : 대체키로 사용할 속성 또는 속성의 집합을 지정하는 것으로 UNIQUE로 지정한 속성은 중복된 값을 가질 수 없음
- FOREIGN KEY ~ REFERENCES ~
  - ON DELETE 옵션 : 참조 테이블의 튜플이 삭제되었을 때 기본 테이블에 취해야 할 사항을 지정함. 옵션에는 NO ACTION, CASCADE, SET NULL, SET DEFAULT가 있음
  - ON UPDATE 옵션 : 참조 테이블의 참조 속성 값이 변경되었을 때 기본 테이블에 취해야 할 사항을 지

정함. 옵션에는 NO ACTION, CASCADE, SET NULL, SET DEFAULT가 있음

- CONSTRAINT : 제약 조건의 이름을 지정함. 이름을 지정할 필요가 없으면 CHECK절만 사용하여 속성 값에 대한 제약 조건을 명시함
- CHECK : 속성 값에 대한 제약 조건을 정의함

##### 다른 테이블을 이용한 테이블 정의

- 기존 테이블의 정보를 이용해 새로운 테이블을 정의할 수 있다.
- 표기 형식

```
CREATE TABLE 신규테이블명 AS SELECT 속성명[, 속성명, ...] FROM 기존테이블명;
```

1. 아래의 <요구사항>을 만족하는 테이블 <Instructor>를 정의하는 SQL의 괄호(① ~ ③)에 알맞은 명령을 쓰시오.

##### <요구사항>

- 'id(문자 5)', 'name(문자 15)', 'dept(문자 15)' 속성을 가진다.
- 'id' 속성은 기본키이다.
- 'name' 속성은 Null이 올 수 없다.
- 'dept' 속성은 <Department> 테이블의 'name' 속성을 참조하는 외래키이다.
  - <Department> 테이블에서 튜플이 삭제되면 관련된 모든 튜플의 'dept' 속성의 값은 NULL로 변경되어야 한다.
  - <Department> 테이블의 'name' 속성이 변경되면 <Instructor> 테이블의 관련된 모든 속성 값도 같은 값으로 변경되어야 한다.

##### <SQL문>

```
CREATE TABLE Instructor
(
  id CHAR(5),
  name CHAR(15) ( ① ),
  dept CHAR(15),
  PRIMARY KEY(id),
  FOREIGN KEY(dept) ( ② )
    Department(name)
    ON DELETE SET NULL
    ON UPDATE ( ③ )
);
```

답

- ① :  
② :  
③ :

2. <사원> 테이블의 '사번', '성명', '직급' 속성을 이용하여 <진급사원> 테이블을 정의하는 SQL문을 작성하시오.

답 :

해설 1.

```
1 CREATE TABLE Instructor
2 ( id CHAR(5),
3   name CHAR(15) NOT NULL,
4   dept CHAR(15),
5   PRIMARY KEY(id),
6   FOREIGN KEY(dept) REFERENCES
      Department(name)
7     ON DELETE SET NULL
8     ON UPDATE CASCADE
9 );
```

- ① 생성한 테이블 이름은 <Instructor>이다.
- ② 'id' 속성은 문자 5자리이다.
- ③ 'name' 속성은 문자 15자리이며, NULL 값을 갖지 않는다.
- ④ 'dept' 속성은 문자 15자리이다.
- ⑤ 'id' 속성은 기본키이다.
- ⑥ 'dept' 속성은 <Department> 테이블의 기본키인 'name' 속성을 참조하는 외래키이다.
- ⑦ <Department> 테이블에서 튜플이 삭제되면 관련된 모든 튜플의 'dept' 속성의 값을 NULL로 변경한다.
- ⑧ <Department> 테이블에서 튜플이 변경되면 관련된 모든 튜플의 'dept' 속성의 값도 같은 값으로 변경한다.

정답 1. ① Not Null ② References ③ Cascade

2. CREATE TABLE 진급사원 AS SELECT 사번, 성명, 직급 FROM 사원;

## [핵심105] DDL – CREATE VIEW

- 뷰(View)를 정의하는 명령문이다.
- 표기 형식

```
CREATE VIEW 뷰명[(속성명, 속성명, ...)]
AS SELECT문;
```

- SELECT문을 서브 쿼리로 사용하여 SELECT문의 결과로서 뷰를 생성한다.
- 서브 쿼리인 SELECT문에는 UNION이나 ORDER BY 절을 사용할 수 없다.
- 속성명을 기술하지 않으면 SELECT문의 속성명이 자동으로 사용된다.

2020년 1회 기능사 실기

1. <학생> 테이블에서 2학년 학생들의 성명, 사진, 학년으로 구성된 뷰를 '출석부'라는 이름으로 생성하려고 한다. <출석부> 뷰를 생성하는 SQL문의 괄호에 알맞은 명령을 쓰시오.

<SQL문>

```
CREATE VIEW 출석부
(      ) 성명, 사진, 학년
FROM 학생
WHERE 학년 = 2;
```

답 :

해설 1.

- CREATE VIEW 출석부 : 생성한 뷰의 이름은 <출석부>이다.
- AS SELECT 성명, 사진, 학년 : '성명', '사진', '학년' 속성을 가져온다.
- FROM 학생 : <학생> 테이블에서 속성을 가져와 뷰를 생성한다.
- WHERE 학년 = 2 : '학년' 속성의 값이 2인 자료만을 뷰로 생성한다.

정답 1. AS SELECT

## [핵심106] DDL – ALTER TABLE

- 테이블에 대한 정의를 변경하는 명령문이다.
- 표기 형식

```
ALTER TABLE 테이블명 ADD 속성명 데이터_타입 [DEFAULT '기본값'];
ALTER TABLE 테이블명 ALTER | MODIFY 속성명 [SET DEFAULT '기본값'];
ALTER TABLE 테이블명 DROP COLUMN 속성명 [CASCADE];
```

- ADD : 새로운 속성(열)을 추가할 때 사용
- ALTER | MODIFY : 특정 속성의 정의를 변경할 때 사용
- DROP COLUMN : 특정 속성을 삭제할 때 사용

2020년 2회 기능사 실기

1. 테이블의 구조를 변경할 때 사용하는 DDL 명령어를 쓰시오.  
답:

2020년 2회 기능사 실기

2. <HRD> 테이블의 '주민번호' 속성에 "900351"을 입력할 때 오류가 발생하여 <HRD> 테이블의 '주민번호' 속성의 크기를 6으로 변경하는 SQL문을 작성하였다. 괄호에 알맞은 명령을 쓰시오.

<SQL문>

ALTER TABLE HRD ( ) 주민번호 NUMBER(6);

답:

정답 1. ALTER 2. ALTER 또는 MODIFY

### [핵심107] DDL – DROP TABLE

- 기본 테이블을 제거하는 명령문이다.
- 표기 형식

DROP TABLE 테이블명 [CASCADE | RESTRICTED];

- CASCADE : 제거할 요소를 참조하는 다른 모든 개체를 함께 제거함. 즉 주 테이블의 데이터 제거 시 각 외래키와 관계를 맺고 있는 모든 데이터를 제거하는 참조 무결성 제약 조건을 설정하기 위해 사용됨
- RESTRICTED : 다른 개체가 제거할 요소를 참조중일 때는 제거를 취소함

2020년 1회 기능사 실기

1. SQL 중 데이터 정의어(DDL)에서 테이블 구조를 제거할 때 사용하는 명령어를 쓰시오.

답:

2. 다음 <처리 조건>에 부합하는 SQL문을 작성하시오.  
<처리 조건>

- <사원> 테이블을 제거한다.
- <사원> 테이블을 참조하는 모든 데이터도 함께 제거한다.

답:

정답 1. DROP 2. DROP TABLE 사원 CASCADE;

### [핵심108] DCL – GRANT / REVOKE

DCL(Data Control Language, 데이터 제어어)

- 데이터의 보안, 무결성, 회복, 병행 제어 등을 정의하는 데 사용하는 언어이다.
- DCL은 데이터베이스 관리자(DBA)가 데이터 관리를 목적으로 사용한다.
- DCL에는 GRANT, REVOKE, COMMIT, ROLLBACK, SAVEPOINT 등이 있다.

#### GRANT / REVOKE

- 데이터베이스 관리자가 데이터베이스 사용자에게 권한을 부여하거나 취소하기 위한 명령어이다.
- GRANT : 권한 부여를 위한 명령어
- REVOKE : 권한 취소를 위한 명령어
- 사용자등급 지정 및 해제

- GRANT 사용자등급 TO 사용자\_ID\_리스트 [IDENTIFIED BY 암호];
- REVOKE 사용자등급 FROM 사용자\_ID\_리스트;

#### • 테이블 및 속성에 대한 권한 부여 및 취소

- GRANT 권한\_리스트 ON 개체 TO 사용자 [WITH GRANT OPTION];
- REVOKE [GRANT OPTION FOR] 권한\_리스트 ON 개체 FROM 사용자 [CASCADE];

- 권한 종류 : ALL, SELECT, INSERT, DELETE, UPDATE, ALTER 등
- WITH GRANT OPTION : 부여받은 권한을 다른 사용자에게 다시 부여할 수 있는 권한을 부여함
- GRANT OPTION FOR : 다른 사용자에게 권한을 부여할 수 있는 권한을 취소함
- CASCADE : 권한 취소 시 권한을 부여받았던 사용자가 다른 사용자에게 부여한 권한도 연쇄적으로 취소함

2020년 1회 기공사 실기, 2020년 1, 2회 기사 필기

1. 관리자가 사용자에게 데이터베이스에 대한 권한을 부여 또는 회수하거나 데이터 보안, 무결성 유지, 병행수행 제어 등의 기능을 수행하기 위해 사용하는 언어를 영문 약어로 쓰시오.

답 :

2. 김영웅에게 <사원> 테이블에 대한 접근 및 조작에 관한 모든 권한을 부여하는 SQL문을 작성하시오.

답 :

3. 이민지에게 부여된 <고객> 테이블에 대한 SELECT, INSERT, DELETE 권한을 취소하는 SQL문을 작성하시오.

답 :

**정답** 1. DCL 2. GRANT ALL ON 사원 TO 김영웅;  
3. REVOKE SELECT, INSERT, DELETE ON 고객 FROM 이민지;

### [핵심109] COMMIT / ROLLBACK / SAVEPOINT

|           |  |
|-----------|--|
| COMMIT    | 트랜잭션이 성공적으로 끝나면 데이터베이스가 새로운 일관성(Consistency) 상태를 가지기 위해 변경된 모든 내용을 데이터베이스에 반영하여야 하는데, 이때 사용하는 명령어  |
| ROLLBACK  | 아직 COMMIT되지 않은 변경된 모든 내용들을 취소하고 데이터베이스를 이전 상태로 되돌리는 명령어  |
| SAVEPOINT | 트랜잭션 내에 ROLLBACK 할 위치인 저장점을 지정하는 명령어로, 저장점을 지정할 때는 이름을 부여하며, ROLLBACK 시 지정된 저장점까지의 트랜잭션 처리 내용이 취소됨 |

1. 트랜잭션은 데이터베이스에서 하나의 논리적 기능을 수행하기 위한 일련의 연산 집합으로서 작업의 단위가 된다. 트랜잭션의 연산 ROLLBACK과 COMMIT 중 ROLLBACK의 개념을 간략히 서술하시오.

답 :

2. DCL에서 사용되는 명령어 중 다음 설명에 해당하는 명령어를 쓰시오.

- 트랜잭션 내에 ROLLBACK 할 위치인 저장점을 지정하는 명령어이다.
- 저장점을 지정할 때는 이름을 부여한다.
- ROLLBACK 시 지정된 저장점까지의 트랜잭션 처리 내용이 취소된다.

답 :

**정답** 1. ROLLBACK은 아직 COMMIT되지 않은 변경된 모든 내용들을 취소하고 데이터베이스를 이전 상태로 되돌리는 명령어이다.

2. SAVEPOINT

### [핵심110] DML – 삽입, 삭제, 갱신문

DML(Data Manipulation Language, 데이터 조작어)

- 데이터베이스 사용자가 응용 프로그램이나 질의어를 통해 저장된 데이터를 실질적으로 관리하는데 사용되는 언어이다.
- DML에는 SELECT, INSERT, DELETE, UPDATE 등이 있다.

#### 삽입문(INSERT INTO ~)

기본 테이블에 새로운 튜플을 삽입할 때 사용한다.

```
INSERT INTO 테이블명(속성명1, 속성명2, ...)
VALUES (데이터1, 데이터2, ...);
```

#### 삭제문(DELETE FROM ~)

기본 테이블에 있는 튜플들 중에서 특정 튜플을 삭제할 때 사용한다.

```
DELETE
FROM 테이블명
[WHERE 조건];
```



## 갱신문 (UPDATE ~ SET ~)

기본 테이블에 있는 튜플들 중에서 특정 튜플의 내용을 변경할 때 사용한다.

UPDATE 테이블명

SET 속성명 = 데이터[, 속성명=데이터]

[WHERE 조건];

1. 다음은 <학생> 테이블에 학번이 20200823, 성명이 '김치수', 학년이 3, 과목이 '정보처리'인 학생의 정보를 입력하는 SQL문이다. 괄호 ①, ②에 알맞은 명령어를 쓰시오.

<SQL문>

( ① ) 학생  
( ② ) (20200823, '김치수', 3, '정보처리');

답

① :

② :

2. 다음 <처리 조건>에 부합하는 SQL문을 작성하시오.

<처리 조건>

1. <사원> 테이블에서 이름이 '홍길동'인 튜플을 삭제하시오.  
2. 문자형은 싱글(작은)따옴표로 입력하고 문장의 끝에는 세미 콜론(:)을 반드시 표기하시오.

답 :

3. 다음은 <고객> 테이블에서 구매금액이 1,000,000 이상인 고객의 등급을 'VIP'로 변경하는 SQL문이다. 괄호 ①, ②에 알맞은 명령어를 쓰시오.

<SQL문>

UPDATE 고객 ( ① ) 등급 = 'VIP' ( ② ) 구매금액 >= 1000000;

답

① :

② :

해설 1.

- INSERT INTO 학생 : <학생> 테이블에 삽입한다.
- VALUES (20200823, '김치수', 3, '정보처리') : 첫 번째 필드부터 순서대로 20200823, '김치수', 3, '정보처리'를 삽입한다.

2.

- DELETE : 삭제한다.
- FROM 사원 : <사원> 테이블을 대상으로 한다.
- WHERE 이름 = '홍길동' : '이름'이 '홍길동'인 사원만을 대상으로 한다.

3.

- UPDATE 고객 : <고객> 테이블을 갱신한다.
- SET 등급 = 'VIP' : '등급'을 'VIP'로 갱신한다.
- WHERE 구매금액 >= 1000000 : '구매금액'이 1,000,000 이상인 고객만을 대상으로 한다.

정답 1. ① INSERT INTO ② VALUES

2. DELETE FROM 사원 WHERE 이름 = '홍길동';

3. ① SET ② WHERE

## [핵심 111] DML – SELECT

SELECT [PREDICATE] [테이블명].[속성명 [AS 별칭]], [테이블명].[속성명, ...] [, 그룹함수(속성명) [AS 별칭]]  
FROM 테이블명[, 테이블명, ...]  
[WHERE 조건]  
[GROUP BY 속성명, 속성명, ...]  
[HAVING 조건]  
[ORDER BY 속성명 [ASC | DESC]];

### • SELECT절

– PREDICATE : 불러올 튜플 수를 제한할 명령어를 기술함

▶ ALL : 모든 튜플을 검색할 때 지정하는 것으로, 주로 생략함

▶ DISTINCT : 중복된 튜플이 있으면 그 중 첫 번째만 개만 검색함

▶ DISTINCTROW : 중복된 튜플을 검색하지만 선택된 속성의 값이 아닌, 튜플 전체를 대상으로 함

– 속성명 : 검색하여 불러올 속성(열) 및 수식들을 지정함

▶ 기본 테이블을 구성하는 모든 속성을 지정할 때는 '\*'를 기술함

▶ 두 개 이상의 테이블을 대상으로 검색할 때는 '테이블명.속성명'으로 표현함



- AS : 속성 및 연산의 이름을 다른 제목으로 표시하기 위해 사용함
- FROM절 : 질의에 의해 검색될 데이터들을 포함하는 테이블명을 기술함
- WHERE절 : 검색할 조건 기술함
- GROUP BY절 : 특정 속성을 기준으로 그룹화하여 검색할 때 그룹화 할 속성을 지정함
  - 일반적으로 GROUP BY절은 그룹 함수와 함께 사용됨
  - 그룹 함수의 종류

|            |                  |
|------------|------------------|
| COUNT(속성명) | 그룹별 튜플 수를 구하는 함수 |
| MAX(속성명)   | 그룹별 최대값을 구하는 함수  |
| MIN(속성명)   | 그룹별 최소값을 구하는 함수  |
| SUM(속성명)   | 그룹별 합계를 구하는 함수   |
| AVG(속성명)   | 그룹별 평균을 구하는 함수   |

- HAVING절 : GROUP BY와 함께 사용되며, 그룹에 대한 조건을 지정함
- ORDER BY절 : 특정 속성을 기준으로 정렬하여 검색할 때 사용함
  - 속성명 : 정렬의 기준이 되는 속성명을 기술함
  - [ASC | DESC] : 정렬 방식으로서 'ASC'는 오름차순, 'DESC'는 내림차순임. 생략하면 오름차순으로 지정됨

#### 잠깐만요 ① SELEC문의 실행 작동 순서

FROM → WHERE → GROUP BY → HAVING → SELECT → DISTINCT → ORDER BY

2020년 1회 기사 실기, 2020년 1, 2회 기사 필기

1. 학생(STUDENT) 테이블에 전기과 학생이 50명, 전산과 학생이 100명, 전자과 학생이 50명 있다고 할 때, 다음 SQL문 ①, ②, ③의 실행 결과로 표시되는 튜플의 수를 쓰시오. (단, DEPT 필드는 학과를 의미한다)

- ① SELECT DEPT FROM STUDENT;
- ② SELECT DISTINCT DEPT FROM STUDENT;
- ③ SELECT COUNT(DISTINCT DEPT) FROM STUDENT WHERE DEPT = '전산과';

#### 답

- ① :
- ② :
- ③ :

2020년 2회 기능사 실기

2. 다음은 SELECT문의 실행 작동 순서를 나열한 것이다. 괄호 (① ~ ③)에 알맞은 명령어를 보기에서 찾아 쓰시오.

FROM → ( ① ) → ( ② ) → ( ③ ) → SELECT  
→ ORDER BY

#### (보기)

GROUP BY, HAVING, WHERE

#### 답

- ① :
- ② :
- ③ :

#### 해설 1.

①

- SELECT DEPT : 'DEPT'를 표시한다.
- FROM STUDENT; : <STUDENT> 테이블을 대상으로 검색한다.
- ※ <STUDENT> 테이블에서 'DEPT'를 검색합니다. 총 200개의 튜플이 들어 있고 검색 조건이 없으므로 튜플의 수는 200입니다.

②

- SELECT DISTINCT DEPT : 'DEPT'를 표시하되, 같은 'DEPT' 속성의 값은 한 번만 표시한다.
- FROM STUDENT; : <STUDENT> 테이블을 대상으로 검색한다.
- ※ <STUDENT> 테이블에서 'DEPT'를 검색하는 데 중복된 결과는 처음의 한 개만 검색에 포함시킵니다. 전기과 50개 튜플의 'DEPT' 속성의 값이 같으므로 1개, 전산과 100개 튜플의 'DEPT' 속성의 값이 같으므로 1개, 전자과 50개 튜플의 'DEPT' 속성의 값이 같으므로 1개를 검색에 포함시키므로 3개의 튜플이 검색됩니다.

③

- SELECT COUNT(DISTINCT DEPT) : 'DEPT'의 개수를 표시하되, 같은 'DEPT' 속성의 값은 한 번만 계산한다.
- FROM STUDENT; : <STUDENT> 테이블을 대상으로 검색한다.
- WHERE DEPT = '전산과'; : 'DEPT'가 "전산과"인 자료만을 대상으로 검색한다.
- ※ <STUDENT> 테이블에서 'DEPT' 속성의 값이 '전산과'인 튜플에 대해 중복을 제거하고 개수를 세므로 10이 검색 결과로 표시됩니다.

정답 1. ① 200 ② 3 ③ 1

2. ① WHERE ② GROUP BY ③ HAVING

## [핵심112] 하위 질의

조건절에 주어진 질의를 먼저 수행하여 그 검색 결과를 조건절의 피연산자로 사용한다.

### <사원>

| 이름  | 부서  | 생일       | 주소  | 기본급 |
|-----|-----|----------|-----|-----|
| 홍길동 | 기획  | 04/05/61 | 망원동 | 120 |
| 임꺽정 | 인터넷 | 01/09/69 | 서교동 | 80  |
| 황진이 | 편집  | 07/21/75 | 합정동 | 100 |
| 김선달 | 편집  | 10/22/73 | 망원동 | 90  |
| 성춘향 | 기획  | 02/20/64 | 대흥동 | 100 |
| 장길산 | 편집  | 03/11/67 | 상암동 | 120 |
| 일지매 | 기획  | 04/29/78 | 연남동 | 110 |
| 강건달 | 인터넷 | 12/11/80 |     | 90  |

### <여가활동>

| 이름  | 취미    | 경력 |
|-----|-------|----|
| 김선달 | 당구    | 10 |
| 성춘향 | 나이트댄스 | 5  |
| 일지매 | 태권    | 15 |
| 임꺽정 | 씨름    | 8  |

**예제** '취미'가 "나이트댄스"인 사원의 '이름'과 '주소'를 검색하시오.

SELECT 이름, 주소

FROM 사원

WHERE 이름 = (SELECT 이름 FROM 여가활동 WHERE 취미 = '나이트댄스');

- 먼저 "SELECT 이름 FROM 여가활동 WHERE 취미 = '나이트댄스'"를 수행하여 <여가활동> 테이블에서 '성춘향'을 찾는다. 그런 다음 하위 질의에 해당하는 피연산자의 자리에 '성춘향'을 대입하면 질의문은 "SELECT 이름, 주소 FROM 사원 WHERE 이름 = '성춘향'"과 같다.
- WHERE 절의 조건에 IN 연산자를 이용하여 다음과 같이 조건을 지정해도 된다.
- WHERE 이름 IN (SELECT 이름 FROM 여가활동 WHERE 취미 = '나이트댄스');

### <결과>

| 이름  | 주소  |
|-----|-----|
| 성춘향 | 대흥동 |

2020년 2회 기공사 실기, 2020년 1, 2회 기사 필기

1. 다음 SQL문의 실행 결과를 쓰시오.

```
SELECT 가격 FROM 도서가격
WHERE 책번호 = (SELECT 책번호 FROM 도서 WHERE 책명 = '데이터베이스');
```

### <도서>

| 책번호  | 책명     |
|------|--------|
| 1111 | 데이터베이스 |
| 2222 | 운영체제   |
| 3333 | 자료구조   |

### <도서가격>

| 책번호  | 가격    |
|------|-------|
| 1111 | 18000 |
| 2222 | 20000 |
| 3333 | 10000 |
| 4444 | 15000 |

**답 :**

#### 해설 1.

② SELECT 가격 FROM 도서가격 WHERE 책번호 =

① (SELECT 책번호 FROM 도서 WHERE 책명 = '데이터베이스');

①

• SELECT 책번호 : '책번호'를 표시한다.

• FROM 도서 : <도서> 테이블을 대상으로 검색한다.

• WHERE 책명 = '데이터베이스' : '책명'이 "데이터베이스"인 자료만을 대상으로 검색한다.

※ '책명'이 "데이터베이스"인 도서의 책번호 "1111"이 검색되므로 ②의 질의 문은 SELECT 가격 FROM 도서가격 WHERE 책번호 = '1111'이 됩니다.

②

• SELECT 가격 : '가격'을 표시한다.

• FROM 도서가격 : <도서가격> 테이블을 대상으로 검색한다.

• WHERE 책번호 = '1111' : '책번호'가 "1111"인 자료만을 대상으로 검색한다.

※ '책번호'가 "1111"인 도서의 가격은 18000입니다.

**정답** 1. 18000

