

1

문제: 키가 가장 큰 학생의 이름과 키를 구하세요.

목적: 단일 행 서브쿼리(MAX)

정답:

```
SELECT student_name, height
FROM student
WHERE height = (SELECT MAX(height) FROM student);
```

2

문제: '정보통신학' 학과에 속한 학생 이름을 구하세요.

목적: 서브쿼리로 학과 ID 찾기(= 스칼라)

정답:

```
SELECT student_name
FROM student
WHERE department_id = (
    SELECT department_id FROM department WHERE
    department_name='정보통신학'
);
```

3

문제: '가교수'가 담당하는 과목 이름을 구하세요.

목적: 스칼라 서브쿼리(교수→과목)

정답:

```
SELECT course_name
FROM course
WHERE professor_id = (
    SELECT professor_id FROM professor WHERE professor_name='가교수'
);
```

4

문제: 각 학생의 학과 이름을 학생 이름과 함께 조회하세요.

목적: SELECT절 스칼라 서브쿼리

정답:

```
SELECT s.student_name,  
       (SELECT department_name FROM department d WHERE  
d.department_id=s.department_id) AS dept_name  
FROM student s;
```

5

문제: 가장 키가 작은 학생의 이름과 키를 구하세요.

목적: 단일 행 서브쿼리(MIN)

정답:

```
SELECT student_name, height  
FROM student  
WHERE height = (SELECT MIN(height) FROM student);
```

6

문제: '국문학' 학과의 학생 수를 구하세요.

목적: 스칼라 서브쿼리 + 집계

정답:

```
SELECT COUNT(*) AS cnt  
FROM student  
WHERE department_id = (  
    SELECT department_id FROM department WHERE department_name='국문학'  
);
```

7

문제: '정보통신학' 학과에 속한 교수 이름을 구하세요.

목적: 스칼라 서브쿼리 필터

정답:

```
SELECT professor_name  
FROM professor  
WHERE department_id = (  
    SELECT department_id FROM department WHERE department_name='정보통신학'  
);
```

```
SELECT department_id FROM department WHERE  
department_name='정보통신학'  
);
```

8

문제: 과목을 담당하지 않는 교수 이름을 구하세요.

목적: **NOT EXISTS**

정답:

```
SELECT professor_name  
FROM professor p  
WHERE NOT EXISTS (  
    SELECT 1 FROM course c WHERE c.professor_id = p.professor_id  
);
```

9

문제: '교양영어'를 수강하는 학생 이름을 구하세요.

목적: **IN** + 스칼라 서브쿼리

정답:

```
SELECT student_name  
FROM student  
WHERE student_id IN (  
    SELECT student_id  
    FROM student_course  
    WHERE course_id = (SELECT course_id FROM course WHERE  
course_name='교양영어')  
);
```

10

문제: 전체 평균 키보다 큰 학생들의 이름과 키를 구하세요.

목적: 단일 행 서브쿼리(**AVG**) 비교

정답:

```
SELECT student_name, height  
FROM student  
WHERE height > (SELECT AVG(height) FROM student);
```

11

문제: 자기 학과 평균 키보다 큰 학생의 이름과 키를 구하세요.

목적: 상관 서브쿼리 + 집계 함수

정답:

```
SELECT s.student_name, s.height
FROM student s
WHERE s.height > (
    SELECT AVG(height) FROM student WHERE department_id =
s.department_id
);
```

12

문제: '스티브잡스' 교수가 담당하는 과목을 듣는 학생 이름을 구하세요.

목적: 다단계 서브쿼리(IN-IN-==)

정답:

```
SELECT student_name
FROM student
WHERE student_id IN (
    SELECT student_id
    FROM student_course
    WHERE course_id IN (
        SELECT course_id
        FROM course
        WHERE professor_id = (
            SELECT professor_id FROM professor WHERE
professor_name='스티브잡스'
        )
    )
);
```

13

문제: '수학' 학과 학생 중 가장 키가 큰 학생의 이름과 키를 구하세요.

목적: 스칼라 서브쿼리 + MAX

정답:

```

SELECT student_name, height
FROM student
WHERE department_id = (SELECT department_id FROM department WHERE
department_name='수학')
AND height = (
SELECT MAX(height)
FROM student
WHERE department_id = (SELECT department_id FROM department
WHERE department_name='수학')
);

```

14

문제: 수강생이 한 명도 없는 과목 이름을 구하세요.

목적: NOT EXISTS

정답:

```

SELECT c.course_name
FROM course c
WHERE NOT EXISTS (
SELECT 1 FROM student_course sc WHERE sc.course_id = c.course_id
);

```

15

문제: 두 과목 이상 수강하는 학생의 이름과 수강 과목 수를 구하세요.

목적: 서브쿼리(집계) + JOIN

정답:

```

SELECT s.student_name, t.course_count
FROM student s
JOIN (
SELECT student_id, COUNT(*) AS course_count
FROM student_course
GROUP BY student_id
HAVING COUNT(*) >= 2
) t ON s.student_id = t.student_id;

```

16

문제: 개설된 과목이 존재하는 학과 이름을 구하세요.

목적: **IN** + 서브쿼리(JOIN 내부)

정답:

```
SELECT DISTINCT d.department_name
FROM department d
WHERE d.department_id IN (
    SELECT p.department_id
    FROM professor p
    JOIN course c ON c.professor_id = p.professor_id
);
```

17

문제: '다교수'가 담당하는 과목을 수강하는 학생 이름을 구하세요.

목적: 다단계 서브쿼리

정답:

```
SELECT student_name
FROM student
WHERE student_id IN (
    SELECT student_id
    FROM student_course
    WHERE course_id IN (
        SELECT course_id FROM course
        WHERE professor_id = (SELECT professor_id FROM professor WHERE
professor_name='다교수')
    )
);
```

18

문제: 학생이 수강한 과목의 담당 교수 이름을 학생 이름과 함께 조회하세요.

목적: **SELECT**절 대신 **JOIN**으로 해결(참고: 서브쿼리 사고 훈련)

정답:

```
SELECT s.student_name, p.professor_name
FROM student s
JOIN student_course sc ON sc.student_id = s.student_id
JOIN course c ON c.course_id = sc.course_id
JOIN professor p ON p.professor_id = c.professor_id;
```

19

문제: 학생 이름, 학과 이름, 수강 과목 이름을 한 번에 조회하세요.

목적: 다테이블 결합(서브쿼리 대체 사고)

정답:

```
SELECT s.student_name, d.department_name, c.course_name
FROM student s
JOIN department d ON d.department_id = s.department_id
JOIN student_course sc ON sc.student_id = s.student_id
JOIN course c ON c.course_id = sc.course_id;
```

20

문제: '모바일공학' 학과 소속 학생이 수강한 과목 이름을 구하세요.

목적: 중첩 IN 서브쿼리

정답:

```
SELECT DISTINCT c.course_name
FROM course c
WHERE c.course_id IN (
    SELECT sc.course_id
    FROM student_course sc
    WHERE sc.student_id IN (
        SELECT student_id
        FROM student
        WHERE department_id = (
            SELECT department_id FROM department WHERE
department_name='모바일공학'
        )
    )
);
```

21

문제: 어떤 과목도 수강하지 않는 학생 이름을 구하세요.

목적: NOT EXISTS (반정규화 확인)

정답:

```
SELECT s.student_name
FROM student s
WHERE NOT EXISTS (
    SELECT 1 FROM student_course sc WHERE sc.student_id = s.student_id
);
```

22

문제: 한 과목이라도 수강한 학생이 있는 학과 이름을 구하세요.

목적: EXISTS + 상관 서브쿼리

정답:

```
SELECT DISTINCT d.department_name
FROM department d
WHERE EXISTS (
    SELECT 1
    FROM student s
    JOIN student_course sc ON sc.student_id = s.student_id
    WHERE s.department_id = d.department_id
);
```

23

문제: 두 명 이상이 수강하는 과목 이름을 구하세요.

목적: 서브쿼리(집계 HAVING)

정답:

```
SELECT c.course_name
FROM course c
WHERE c.course_id IN (
    SELECT course_id
    FROM student_course
    GROUP BY course_id
    HAVING COUNT(*) >= 2
);
```

24

문제: '가교수'와 같은 학과에 속한 학생 이름을 구하세요.

목적: 스칼라 서브쿼리(학과 동치)

정답:

```
SELECT student_name
FROM student
WHERE department_id = (
    SELECT department_id FROM professor WHERE professor_name='가교수'
);
```

25

문제: 학과별 평균 키보다 큰 학생의 이름, 학과명, 키를 구하세요.

목적: 상관 서브쿼리 + 집계 함수

정답:

```
SELECT s.student_name, d.department_name, s.height
FROM student s
JOIN department d ON d.department_id = s.department_id
WHERE s.height > (
    SELECT AVG(height) FROM student WHERE department_id =
s.department_id
);
```

26

문제: 가장 많은 과목을 담당하는 교수 이름을 구하세요.

목적: 서브쿼리(집계 정렬 + LIMIT 1)

정답:

```
SELECT professor_name
FROM professor
WHERE professor_id = (
    SELECT professor_id
    FROM course
    GROUP BY professor_id
    ORDER BY COUNT(*) DESC
    LIMIT 1
);
```

27

문제: 수강생 수가 가장 많은 과목 이름을 구하세요.

목적: 서브쿼리(집계 정렬 + **LIMIT 1**)

정답:

```
SELECT course_name
FROM course
WHERE course_id = (
    SELECT course_id
    FROM student_course
    GROUP BY course_id
    ORDER BY COUNT(*) DESC
    LIMIT 1
);
```

28

문제: 수강생 수가 가장 적은 과목 이름을 구하세요.

목적: 서브쿼리(집계 정렬 + **LIMIT 1**)

정답:

```
SELECT course_name
FROM course
WHERE course_id = (
    SELECT course_id
    FROM student_course
    GROUP BY course_id
    ORDER BY COUNT(*) ASC
    LIMIT 1
);
```

29

문제: 개설된 모든 과목을 모두 수강하는 학생 이름을 구하세요.

목적: 이중 **NOT EXISTS**(관계 나눗셈 패턴)

정답:

```
SELECT s.student_name
FROM student s
WHERE NOT EXISTS (
```

```

SELECT 1
FROM course c
WHERE NOT EXISTS (
    SELECT 1
    FROM student_course sc
    WHERE sc.course_id = c.course_id
        AND sc.student_id = s.student_id
)
);

```

30

문제: 수강 과목 수가 전체 평균보다 많은 학생 이름을 구하세요.

목적: 서브쿼리(파생 테이블 평균 비교)

정답:

```

SELECT s.student_name
FROM student s
JOIN (
    SELECT student_id, COUNT(*) AS cnt
    FROM student_course
    GROUP BY student_id
) t ON t.student_id = s.student_id
WHERE t.cnt > (
    SELECT AVG(cnt)
    FROM (
        SELECT COUNT(*) AS cnt
        FROM student_course
        GROUP BY student_id
    ) sub
);

```