

# MariaDB PART 2 워크북

---

## 1. 데이터 검색하기 (연습문제)

### ✓ 연습문제 1

급여가 400만원 이상인 교수들의 교수번호, 이름, 급여를 급여 높은 순으로 정렬하여 상위 3명만 조회하시오.

```
SELECT id, name, sal FROM professors
WHERE sal >= 400
ORDER BY sal DESC
LIMIT 0, 3;
```

### ✓ 연습문제 2

담당교수가 정해지지 않은 학생의 학번, 이름, 학년, 성별, 학적 상태를 학년 순으로 조회하시오.

```
SELECT id, name, grade, gender, status FROM students
WHERE professor_id IS NULL;
```

### ✓ 연습문제 3

키가 큰 상위 5명의 3학년 여학생의 이름, 학년, 성별, 키를 조회하시오.

```
SELECT name, grade, gender, height FROM students
WHERE gender = '여' AND grade >= 3
ORDER BY height DESC
LIMIT 0, 5;
```

### ✓ 연습문제 4

학점이 2 또는 3이 아닌 과목의 과목번호, 과목명, 학점을 조회하는 SQL문을 작성하시오. 단, 과목명 가나다 순으로 정렬하시오.

```
SELECT id, name, credit FROM subjects
WHERE credit NOT IN (2, 3)
ORDER BY name ASC;
```

### ✓ 연습문제 5

보직수당(comm)이 등록되어 있지 않은 교수의 교수번호, 이름, 보직수당을 교수번호 내림차순으로 조회하는 SQL문을 작성하시오.

```
SELECT id, name, comm FROM professors  
WHERE comm IS NULL  
ORDER BY id DESC;
```

## 2. SQL 함수 (1) (연습문제)

### ✓ 연습문제 1

학과명(dname)을 모두 소문자로 변환하고, 학과명이 10자 이상인 경우만 조회하시오.

```
SELECT LOWER(dname) AS lower_dname FROM departments
WHERE CHAR_LENGTH(dname) >= 10;
```

### ✓ 연습문제 2

과목명(name)의 앞 3글자와 학점을 결합해 "과목코드: AAA-3" 형식으로 출력하시오.

(예: '데이터베이스' 3학점 → '과목코드: 데이-3')

```
SELECT CONCAT('과목코드: ', LEFT(name, 3), '-', credit) AS subject_code
FROM subjects;
```

### ✓ 연습문제 3

학생의 전화번호(phone) 중, 가운데 4자리를 추출하여 출력하시오.

단, 전화번호는 "010-1234-5678" 형식으로 저장되어 있다고 가정하시오.

```
SELECT name, SUBSTRING(phone, INSTR(phone, '-') + 1, 4) AS mid_number
FROM students;
```

### ✓ 연습문제 4

학과명을 오른쪽으로 15자 고정 폭으로 출력하고, 전화번호는 왼쪽에 '-'를 붙여 10자 고정 폭으로 출력하시오.

두 값을 합쳐 하나의 문자열로 출력하시오.

```
SELECT CONCAT(RPAD(dname, 15, ' '), LPAD(phone, 10, '-')) AS dept_info
FROM departments
WHERE phone IS NOT NULL;
```

### ✓ 연습문제 5

교수의 이메일 중 @ 앞 아이디만 추출하고, 이를 대문자로 변환한 값을 출력하시오.

또한 급여는 원단위로 환산한 후 세 자리마다 쉼표, 소수점 2자리까지 포맷하시오.

결과는 "USERID : 급여" 형식으로 출력하시오.

```
SELECT CONCAT(UPPER(SUBSTRING(email, 1, INSTR(email, '@') - 1)), ' : ',  
FORMAT(sal * 10000, 2)) AS email_salary  
FROM professors  
WHERE email IS NOT NULL;
```

### 3. SQL 함수 (2) (연습문제)

#### ✓ 연습문제 1

학생들의 키와 몸무게를 활용하여 건강 지수를 계산하려고 한다. 키(cm)의 제곱을 몸무게(kg)로 나눈 값을 소수점 둘째 자리까지 반올림하여 **건강지수**로 출력하고, 그 값이 35 이상이면 '**우수**', 30 이상이면 '**양호**', 그보다 작으면 '**주의**'라고 판단하여 **건강등급**으로 함께 표시하시오.

```
SELECT
    name, height, weight,
    ROUND(POWER(height, 2) / weight, 2) AS 건강지수,
    CASE
        WHEN ROUND(POWER(height, 2) / weight, 2) >= 35 THEN '우수'
        WHEN ROUND(POWER(height, 2) / weight, 2) >= 30 THEN '양호'
        ELSE '주의'
    END AS 건강등급
FROM students
ORDER BY 건강지수 DESC
LIMIT 5;
```

#### ✓ 연습문제 2

기준 몸무게 60kg과 비교하여 학생들의 몸무게 차이를 절댓값으로 계산하여 **차이**로 출력하고, 몸무게를 10 단위로 내림 처리하여 **몸무게구간**이라는 이름으로 표시하시오. 차이가 적은 순서대로 상위 5명을 조회하시오.

```
SELECT
    name, weight,
    ABS(weight - 60) AS 차이,
    FLOOR(weight / 10) * 10 AS 몸무게구간
FROM students
ORDER BY 차이 ASC
LIMIT 5;
```

#### ✓ 연습문제 3

학생의 키에 따라 180cm 이상은 '**장신**', 160cm 이상은 '**보통**', 그 이하는 '**단신**'으로 구분하여 **키등급**을 출력하시오. 또한 학생의 ID가 짝수인지 홀수인지를 판단하여 '**짝수**', '**홀수**' 중 하나를 **번호유형**으로 출력하시오.

```
SELECT
    id, name, height,
    CASE
        WHEN height >= 180 THEN '장신'
        WHEN height >= 160 THEN '보통'
        ELSE '단신'
    END AS 키등급,
    CASE
        WHEN id % 2 = 0 THEN '짝수'
        ELSE '홀수'
    END AS 번호유형
FROM students
ORDER BY 키등급, 번호유형
```

```
END AS 키등급,
IF(MOD(id, 2) = 0, '짝수', '홀수') AS 번호유형
FROM students
LIMIT 5;
```

#### ✓ 연습문제 4

학생의 성별 정보를 출력하되, '남'은 'Male', '여'는 'Female', 그 외는 '기타'로 구분하시오. 성별 정보가 NULL인 경우는 '미입력'으로 출력되도록 하시오.

```
SELECT
  name, gender,
  IFNULL(
    CASE
      WHEN gender = '남' THEN 'Male'
      WHEN gender = '여' THEN 'Female'
      ELSE '기타'
    END,
    '미입력'
  ) AS 성별_표기
FROM students
LIMIT 5;
```

#### ✓ 연습문제 5

학생의 키가 170cm 이상이면 NULL, 그보다 작으면 '작음'이라고 표시하되, 이 결과가 NULL일 경우에는 '경고'라는 문자열로 대체하여 키상태 컬럼에 출력하시오.

```
SELECT
  name, height,
  IFNULL(NULLIF(SIGN(170 - height), 0), '경고') AS 키상태
FROM students
LIMIT 5;
```

## 연구과제

⇒ 이름이 없는 동물의 아이디 (Level.1)

<https://school.programmers.co.kr/learn/courses/30/lessons/59039>

```
SELECT ANIMAL_ID FROM ANIMAL_INS WHERE NAME IS NULL;
```

⇒ 이름이 있는 동물의 아이디 (Level.1)

<https://school.programmers.co.kr/learn/courses/30/lessons/59407>

```
SELECT ANIMAL_ID FROM ANIMAL_INS WHERE NAME IS NOT NULL ORDER BY ANIMAL_ID ASC;
```

⇒ 아픈 동물 찾기 (Level.1)

<https://school.programmers.co.kr/learn/courses/30/lessons/59036>

```
SELECT ANIMAL_ID, NAME FROM ANIMAL_INS  
WHERE INTAKE_CONDITION='Sick'  
ORDER BY ANIMAL_ID
```

⇒ 어린 동물 찾기 (Level.1)

<https://school.programmers.co.kr/learn/courses/30/lessons/59037>

```
SELECT ANIMAL_ID, NAME FROM ANIMAL_INS WHERE INTAKE_CONDITION != 'Aged' ORDER  
BY ANIMAL_ID;
```

⇒ 강원도에 위치한 생산공장 목록 출력하기 (Level.1)

<https://school.programmers.co.kr/learn/courses/30/lessons/131112>

```
SELECT FACTORY_ID, FACTORY_NAME, ADDRESS FROM FOOD_FACTORY  
WHERE ADDRESS LIKE '강원도%'  
ORDER BY FACTORY_ID;
```

⇒ Python 개발자 찾기 (Level.1)

<https://school.programmers.co.kr/learn/courses/30/lessons/276013>

```
SELECT ID, EMAIL, FIRST_NAME, LAST_NAME  
FROM DEVELOPER_INFOS  
WHERE SKILL_1='Python' OR SKILL_2='Python' OR SKILL_3='Python'  
ORDER BY ID ASC;
```

⇒ 루시와 엘라 찾기 (Level.2)

<https://school.programmers.co.kr/learn/courses/30/lessons/59046>

```
SELECT ANIMAL_ID, NAME, SEX_UPON_INTAKE FROM ANIMAL_INS
WHERE NAME IN ('Lucy', 'Ella', 'Pickle', 'Rogan', 'Sabrina', 'Mitty')
ORDER BY ANIMAL_ID ASC
```

⇒ 이름에 e이 들어가는 동물 찾기 (Level.2)

<https://school.programmers.co.kr/learn/courses/30/lessons/59047>

```
SELECT ANIMAL_ID, NAME FROM ANIMAL_INS
WHERE ANIMAL_TYPE = 'Dog' AND INSTR(UPPER(NAME), 'EL') > 0
ORDER BY NAME ASC;
```

⇒ 특정 옵션이 포함된 자동차 리스트 구하기 (Level.1)

<https://school.programmers.co.kr/learn/courses/30/lessons/157343>

```
SELECT CAR_ID, CAR_TYPE, DAILY_FEE, OPTIONS
FROM CAR_RENTAL_COMPANY_CAR
WHERE INSTR(OPTIONS, '네비게이션') > 0
ORDER BY CAR_ID DESC;
```

⇒ 경기도에 위치한 식품창고 목록 출력하기 (Level.1)

<https://school.programmers.co.kr/learn/courses/30/lessons/131114>

```
SELECT WAREHOUSE_ID, WAREHOUSE_NAME, ADDRESS, IFNULL(FREEZER_YN, 'N')
FROM FOOD_WAREHOUSE
WHERE LEFT(ADDRESS, 3) = '경기도'
ORDER BY WAREHOUSE_ID ASC;
```

⇒ 12세 이하인 여자 환자 목록 출력하기 (Level.1)

<https://school.programmers.co.kr/learn/courses/30/lessons/132201>

```
SELECT PT_NAME, PT_NO, GEND_CD, AGE, IFNULL(TLNO, 'NONE') AS TLNO
FROM PATIENT
WHERE AGE <= 12 AND GEND_CD='W'
ORDER BY AGE DESC, PT_NAME ASC;
```

⇒ NULL 처리하기 (Level.2)



<https://school.programmers.co.kr/learn/courses/30/lessons/59410>

```
SELECT ANIMAL_TYPE, IFNULL(NAME, 'No name') AS NAME, SEX_UPON_INTAKE
FROM ANIMAL_INS
ORDER BY ANIMAL_ID ASC
```

### ⇒ 중성화 여부 파악하기 (Level.2)

<https://school.programmers.co.kr/learn/courses/30/lessons/59409>

```
SELECT
    ANIMAL_ID, NAME,
    IF(SEX_UPON_INTAKE LIKE 'Neutered%' OR SEX_UPON_INTAKE LIKE 'Spayed%', 'O',
    'X') AS 중성화
FROM ANIMAL_INS
ORDER BY ANIMAL_ID
```

### ⇒ 조건에 부합하는 중고거래 상태 조회하기 (Level.2)

<https://school.programmers.co.kr/learn/courses/30/lessons/164672>

```
SELECT
    BOARD_ID, WRITER_ID, TITLE, PRICE,
    IF(STATUS='SALE', '판매중', IF(STATUS='RESERVED', '예약중', '거래완료')) AS
STATUS
FROM USED_GOODS_BOARD
WHERE DATE_FORMAT(CREATED_DATE, '%Y%m%d') = '20221005'
ORDER BY BOARD_ID DESC;
```

```
SELECT
    BOARD_ID, WRITER_ID, TITLE, PRICE,
    CASE
        WHEN STATUS='SALE' THEN '판매중'
        WHEN STATUS='RESERVED' THEN '예약중'
        ELSE '거래완료'
    END AS STATUS
FROM USED_GOODS_BOARD
WHERE DATE_FORMAT(CREATED_DATE, '%Y%m%d') = '20221005'
ORDER BY BOARD_ID DESC;
```

### ⇒ 대장균의 크기에 따라 분류하기 1 (Level.3)

<https://school.programmers.co.kr/learn/courses/30/lessons/299307>

```
SELECT
    ID,
    IF(SIZE_OF_COLONY <= 100, 'LOW', IF(SIZE_OF_COLONY > 1000, 'HIGH',
    'MEDIUM')) AS SIZE
FROM ECOLI_DATA
ORDER BY ID ASC;
```

```
SELECT
    ID,
    CASE
        WHEN SIZE_OF_COLONY > 1000 THEN 'HIGH'
        WHEN SIZE_OF_COLONY > 100 THEN 'MEDIUM'
        ELSE 'LOW'
    END AS SIZE
FROM ECOLI_DATA
ORDER BY ID ASC;
```