



MySQL

▼ SUM, MAX, MIN

```
# 최솟값 구하기
SELECT DATETIME
FROM ANIMAL_INS
ORDER BY DATETIME ASC
LIMIT 1

# 중복 제거하기
SELECT COUNT(DISTINCT NAME)
FROM ANIMAL_INS
```

▼ GROUP BY

```
# 고양이와 개는 몇 마리 있을까
SELECT ANIMAL_TYPE, COUNT(*)
FROM ANIMAL_INS
GROUP BY ANIMAL_TYPE
ORDER BY ANIMAL_TYPE

# 입양 시각 구하기(1)
SELECT HOUR(DATETIME) AS HOUR, COUNT(ANIMAL_ID)
FROM ANIMAL_OUTS
GROUP BY HOUR
HAVING HOUR >= 9 AND HOUR <= 19
ORDER BY HOUR

# 입양 시각 구하기(2)
SET @HOUR := -1;
# 변수 선언, 초기값을 -1로 설정
SELECT (@HOUR := @HOUR + 1) AS HOUR, # +1이 되어 0이 저장
(SELECT COUNT(*) FROM ANIMAL_OUTS WHERE HOUR(DATETIME) = @HOUR) AS COUNT
# 0부터 시작
FROM ANIMAL_OUTS
WHERE @HOUR < 23 # <23일 때까지 @HOUR 값이 계속 +1씩 증가
# SET 옆에 변수명과 초기값을 설정할 수 있다
# @가 붙은 변수는 프로시저가 종료되어도 유지된다
```

▼ IS NULL

```

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

# 동명 동물 수 찾기
SELECT NAME, COUNT(NAME)
FROM ANIMAL_INS
WHERE NAME IS NOT NULL
GROUP BY NAME
HAVING COUNT(NAME) > 1
ORDER BY NAME ASC

# 이름이 없는 동물의 아이디

```

▼ JOIN

```

# 없어진 기록 찾기
SELECT B.ANIMAL_ID, B.NAME
FROM ANIMAL_INS AS A
RIGHT JOIN ANIMAL_OUTS AS B
ON A.ANIMAL_ID = B.ANIMAL_ID
WHERE A.ANIMAL_ID IS NULL
ORDER BY A.ANIMAL_ID ASC

# 있었는데요 없었습니다
SELECT A.ANIMAL_ID, A.NAME
FROM ANIMAL_INS AS A
JOIN ANIMAL_OUTS AS B
ON A.ANIMAL_ID = B.ANIMAL_ID
WHERE A.DATETIME > B.DATETIME
ORDER BY A.DATETIME ASC
# DATETIME 빠르다 = 작다
# 2021-01-01 < 2021-12-31

```

▼ String, Date

```

# 중성화 여부 파악하기
SELECT ANIMAL_ID, NAME,
CASE WHEN SEX_UPON_INTAKE LIKE 'Neutered%' # CASE WHEN; if절과 같다
OR SEX_UPON_INTAKE LIKE 'Sprayed%'
THEN 'O' ELSE 'X'
END 중성화
FROM ANIMAL_INS
ORDER BY ANIMAL_ID

# 오랜 기간 보호한 동물(2)
SELECT A.ANIMAL_ID, A.NAME
FROM ANIMAL_INS A
JOIN ANIMAL_OUTS B
ON A.ANIMAL_ID = B.ANIMAL_ID

```

```
ORDER BY DATEDIFF(B.DATETIME, A.DATETIME) DESC
# DATEDIFF(future date, past date)
LIMIT 2
```

```
# DATETIME에서 DATE로 형 변환
SELECT ANIMAL_ID, NAME,
DATE_FORMAT(DATETIME, '%Y-%m-%d') AS 날짜
FROM ANIMAL_INS
```