

[20-05-31]

프로그래머스

: Problem Count - 10

1. LEVEL 1. 역순 정렬하기

역순 정렬하기

문제 설명

`ANIMAL_INS` 테이블은 동물 보호소에 들어온 동물의 정보를 담은 테이블입니다. `ANIMAL_INS` 테이블 구조는 다음과 같으며, `ANIMAL_ID`, `ANIMAL_TYPE`, `DATETIME`, `INTAKE_CONDITION`, `NAME`, `SEX_UPON_INTAKE` 는 각각 동물의 아이디, 생물 종, 보호 시작일, 보호 시작 시 상태, 이름, 성별 및 중성화 여부를 나타냅니다.

NAME	TYPE	NULLABLE
ANIMAL_ID	VARCHAR(N)	FALSE
ANIMAL_TYPE	VARCHAR(N)	FALSE
DATETIME	DATETIME	FALSE
INTAKE_CONDITION	VARCHAR(N)	FALSE
NAME	VARCHAR(N)	TRUE
SEX_UPON_INTAKE	VARCHAR(N)	FALSE

동물 보호소에 들어온 모든 동물의 이름과 보호 시작일을 조회하는 SQL문을 작성해주세요. 이때 결과는 `ANIMAL_ID` 역순으로 보여주세요. SQL을 실행하면 다음과 같이 출력되어야 합니다.

NAME	DATETIME
Rocky	2016-06-07 09:17:00
Shelly	2015-01-29 15:01:00
Benji	2016-04-19 13:28:00
Jackie	2016-01-03 16:25:00
*Sam	2016-03-13 11:17:00

=> Solve

solution.sql

```
1 SELECT NAME, DATETIME
2 FROM ANIMAL_INS
3 ORDER BY ANIMAL_ID DESC
```

2. LEVEL 1. 아픈 동물 찾기

아픈 동물 찾기

문제 설명

`ANIMAL_INS` 테이블은 동물 보호소에 들어온 동물의 정보를 담은 테이블입니다. `ANIMAL_INS` 테이블 구조는 다음과 같으며, `ANIMAL_ID`, `ANIMAL_TYPE`, `DATETIME`, `INTAKE_CONDITION`, `NAME`, `SEX_UPON_INTAKE` 는 각각 동물의 아이디, 생물 종, 보호 시작일, 보호 시작 시 상태, 이름, 성별 및 중성화 여부를 나타냅니다.

NAME	TYPE	NULLABLE
<code>ANIMAL_ID</code>	<code>VARCHAR(N)</code>	FALSE
<code>ANIMAL_TYPE</code>	<code>VARCHAR(N)</code>	FALSE
<code>DATETIME</code>	<code>DATETIME</code>	FALSE
<code>INTAKE_CONDITION</code>	<code>VARCHAR(N)</code>	FALSE
<code>NAME</code>	<code>VARCHAR(N)</code>	TRUE
<code>SEX_UPON_INTAKE</code>	<code>VARCHAR(N)</code>	FALSE

동물 보호소에 들어온 동물 중 아픈 동물¹의 아이디와 이름을 조회하는 SQL 문을 작성해주세요. 이때 결과는 아이디 순으로 조회해주세요.

예시

예를 들어 `ANIMAL_INS` 테이블이 다음과 같다면

ANIMAL_ID	ANIMAL_TYPE	DATETIME	INTAKE_CONDITION	NAME	SEX_UPON_INTAKE
A365172	Dog	2014-08-26 12:53:00	Normal	Diablo	Neutered Male
A367012	Dog	2015-09-16 09:06:00	Sick	Miller	Neutered Male
A365302	Dog	2017-01-08 16:34:00	Aged	Minnie	Spayed Female
A381217	Dog	2017-07-08 09:41:00	Sick	Cherokee	Neutered Male

이 중 아픈 동물은 Miller와 Cherokee입니다. 따라서 SQL문을 실행하면 다음과 같이 나와야 합니다.

ANIMAL_ID	NAME
A367012	Miller
A381217	Cherokee

=> Solve

solution.sql

```
1 SELECT ANIMAL_ID, NAME
2 FROM ANIMAL_INS
3 WHERE INTAKE_CONDITION LIKE 'SICK'
4 ORDER BY ANIMAL_ID ASC;
```

3. LEVEL 1. 어린 동물 찾기

어린 동물 찾기

문제 설명

`ANIMAL_INS` 테이블은 동물 보호소에 들어온 동물의 정보를 담은 테이블입니다. `ANIMAL_INS` 테이블 구조는 다음과 같으며, `ANIMAL_ID`, `ANIMAL_TYPE`, `DATETIME`, `INTAKE_CONDITION`, `NAME`, `SEX_UPON_INTAKE` 는 각각 동물의 아이디, 생물 종, 보호 시작일, 보호 시작 시 상태, 이름, 성별 및 중성화 여부를 나타냅니다.

NAME	TYPE	NULLABLE
ANIMAL_ID	VARCHAR(N)	FALSE
ANIMAL_TYPE	VARCHAR(N)	FALSE
DATETIME	DATETIME	FALSE
INTAKE_CONDITION	VARCHAR(N)	FALSE
NAME	VARCHAR(N)	TRUE
SEX_UPON_INTAKE	VARCHAR(N)	FALSE

동물 보호소에 들어온 동물 중 **젊은 동물**의 아이디와 이름을 조회하는 SQL 문을 작성해주세요. 이때 결과는 아이디 순으로 조회해주세요.

예시

예를 들어 `ANIMAL_INS` 테이블이 다음과 같다면

ANIMAL_ID	ANIMAL_TYPE	DATETIME	INTAKE_CONDITION	NAME	SEX_UPON_INTAKE
A365172	Dog	2014-08-26 12:53:00	Normal	Diablo	Neutered Male
A367012	Dog	2015-09-16 09:06:00	Sick	Miller	Neutered Male
A365302	Dog	2017-01-08 16:34:00	Aged	Minnie	Spayed Female
A381217	Dog	2017-07-08 09:41:00	Sick	Cherokee	Neutered Male

이 중 젊은 동물은 Diablo, Miller, Cherokee입니다. 따라서 SQL문을 실행하면 다음과 같이 나와야 합니다.

ANIMAL_ID	NAME
A365172	Diablo
A367012	Miller
A381217	Cherokee

=> Solve

solution.sql

```
1 SELECT ANIMAL_ID, NAME
2 FROM ANIMAL_INS
3 WHERE INTAKE_CONDITION <> 'Aged'
4 ORDER BY ANIMAL_ID
```

4. LEVEL 1. 동물의 아이디와 이름

동물의 아이디와 이름

문제 설명

ANIMAL_INS

 테이블은 동물 보호소에 들어온 동물의 정보를 담은 테이블입니다.

ANIMAL_INS

 테이블 구조는 다음과 같으며,

ANIMAL_ID

,

ANIMAL_TYPE

,

DATETIME

,

INTAKE_CONDITION

,

NAME

,

SEX_UPON_INTAKE

 는 각각 동물의 아이디, 생물 종, 보호 시작일, 보호 시작 시 상태, 이름, 성별 및 중성화 여부를 나타냅니다.

NAME	TYPE	NULLABLE
ANIMAL_ID	VARCHAR(N)	FALSE
ANIMAL_TYPE	VARCHAR(N)	FALSE
DATETIME	DATETIME	FALSE
INTAKE_CONDITION	VARCHAR(N)	FALSE
NAME	VARCHAR(N)	TRUE
SEX_UPON_INTAKE	VARCHAR(N)	FALSE

동물 보호소에 들어온 모든 동물의 아이디와 이름을

ANIMAL_ID

순으로 조회하는 SQL문을 작성해주세요. SQL을 실행하면 다음과 같이 출력되어야 합니다.

ANIMAL_ID	NAME
A349996	Sugar
A350276	Jewel
A350375	Meo
A352555	Harley
A352713	Gia
A352872	Peanutbutter
A353259	Bj

((이하 생략))

=> Solve

solution.sql

1

2

3

SELECT ANIMAL_ID, NAME

FROM ANIMAL_INS

ORDER BY ANIMAL_ID

5. LEVEL 1. 여러 기준으로 정렬하기

여러 기준으로 정렬하기

문제 설명

`ANIMAL_INS` 테이블은 동물 보호소에 들어온 동물의 정보를 담은 테이블입니다. `ANIMAL_INS` 테이블 구조는 다음과 같으며, `ANIMAL_ID`, `ANIMAL_TYPE`, `DATETIME`, `INTAKE_CONDITION`, `NAME`, `SEX_UPON_INTAKE` 는 각각 동물의 아이디, 생물 종, 보호 시작일, 보호 시작 시 상태, 이름, 성별 및 중성화 여부를 나타냅니다.

NAME	TYPE	NULLABLE
ANIMAL_ID	VARCHAR(N)	FALSE
ANIMAL_TYPE	VARCHAR(N)	FALSE
DATETIME	DATETIME	FALSE
INTAKE_CONDITION	VARCHAR(N)	FALSE
NAME	VARCHAR(N)	TRUE
SEX_UPON_INTAKE	VARCHAR(N)	FALSE

동물 보호소에 들어온 모든 동물의 아이디와 이름, 보호 시작일을 이름 순으로 조회하는 SQL문을 작성해주세요. 단, 이름이 같은 동물 중에서는 보호를 나중에 시작한 동물을 먼저 보여줘야 합니다.

예시

예를 들어, `ANIMAL_INS` 테이블이 다음과 같다면

ANIMAL_ID	ANIMAL_TYPE	DATETIME	INTAKE_CONDITION	NAME	SEX_UPON_INTAKE
A349996	Cat	2018-01-22 14:32:00	Normal	Sugar	Neutered Male
A350276	Cat	2017-08-13 13:50:00	Normal	Jewel	Spayed Female
A396810	Dog	2016-08-22 16:13:00	Injured	Raven	Spayed Female
A410668	Cat	2015-11-19 13:41:00	Normal	Raven	Spayed Female

- 이름을 사전 순으로 정렬하면 다음과 같으며, 'Jewel', 'Raven', 'Sugar'
- 'Raven'이라는 이름을 가진 개와 고양이가 있으므로, 이 중에서는 보호를 나중에 시작한 고양이를 먼저 조회합니다.

따라서 SQL문을 실행하면 다음과 같이 나와야 합니다.

ANIMAL_ID	NAME	DATETIME
A350276	Jewel	2017-08-13 13:50:00
A396810	Raven	2016-08-22 16:13:00
A410668	Raven	2015-11-19 13:41:00
A349996	Sugar	2018-01-22 14:32:00

=> Solve

solution.sql

```
1 SELECT ANIMAL_ID, NAME, DATETIME
2 FROM ANIMAL_INS
3 ORDER BY NAME ASC, DATETIME DESC
```

6. LEVEL 1. 상위 n개 레코드

상위 n개 레코드

문제 설명

`ANIMAL_INS` 테이블은 동물 보호소에 들어온 동물의 정보를 담은 테이블입니다. `ANIMAL_INS` 테이블 구조는 다음과 같으며, `ANIMAL_ID`, `ANIMAL_TYPE`, `DATETIME`, `INTAKE_CONDITION`, `NAME`, `SEX_UPON_INTAKE` 는 각각 동물의 아이디, 생물 종, 보호 시작일, 보호 시작 시 상태, 이름, 성별 및 중성화 여부를 나타냅니다.

NAME	TYPE	NULLABLE
ANIMAL_ID	VARCHAR(N)	FALSE
ANIMAL_TYPE	VARCHAR(N)	FALSE
DATETIME	DATETIME	FALSE
INTAKE_CONDITION	VARCHAR(N)	FALSE
NAME	VARCHAR(N)	TRUE
SEX_UPON_INTAKE	VARCHAR(N)	FALSE

동물 보호소에 가장 먼저 들어온 동물의 이름을 조회하는 SQL 문을 작성해주세요.

예시

예를 들어 `ANIMAL_INS` 테이블이 다음과 같다면

ANIMAL_ID	ANIMAL_TYPE	DATETIME	INTAKE_CONDITION	NAME	SEX_UPON_INTAKE
A399552	Dog	2013-10-14 15:38:00	Normal	Jack	Neutered Male
A379998	Dog	2013-10-23 11:42:00	Normal	Disciple	Intact Male
A370852	Dog	2013-11-03 15:04:00	Normal	Katie	Spayed Female
A403564	Dog	2013-11-18 17:03:00	Normal	Anna	Spayed Female

이 중 가장 보호소에 먼저 들어온 동물은 Jack입니다. 따라서 SQL문을 실행하면 다음과 같이 나와야 합니다.

NAME
Jack

※ 보호소에 가장 먼저 들어온 동물은 한 마리인 경우만 테스트 케이스로 주어집니다.

=> Solve

solution.sql

```
1 SELECT NAME
2 FROM ANIMAL_INS
3 ORDER BY DATETIME
4 LIMIT 1
```

7. LEVEL 1. 이름이 있는 동물의 아이디

이름이 있는 동물의 아이디

문제 설명

`ANIMAL_INS` 테이블은 동물 보호소에 들어온 동물의 정보를 담은 테이블입니다. `ANIMAL_INS` 테이블 구조는 다음과 같으며, `ANIMAL_ID`, `ANIMAL_TYPE`, `DATETIME`, `INTAKE_CONDITION`, `NAME`, `SEX_UPON_INTAKE` 는 각각 동물의 아이디, 생물 종, 보호 시작일, 보호 시작 시 상태, 이름, 성별 및 중성화 여부를 나타냅니다.

NAME	TYPE	NULLABLE
<code>ANIMAL_ID</code>	<code>VARCHAR(N)</code>	FALSE
<code>ANIMAL_TYPE</code>	<code>VARCHAR(N)</code>	FALSE
<code>DATETIME</code>	<code>DATETIME</code>	FALSE
<code>INTAKE_CONDITION</code>	<code>VARCHAR(N)</code>	FALSE
<code>NAME</code>	<code>VARCHAR(N)</code>	TRUE
<code>SEX_UPON_INTAKE</code>	<code>VARCHAR(N)</code>	FALSE

동물 보호소에 들어온 동물 중, 이름이 있는 동물의 ID를 조회하는 SQL 문을 작성해주세요. 단, ID는 오름차순 정렬되어야 합니다.

예시

예를 들어 `ANIMAL_INS` 테이블이 다음과 같다면

<code>ANIMAL_ID</code>	<code>ANIMAL_TYPE</code>	<code>DATETIME</code>	<code>INTAKE_CONDITION</code>	<code>NAME</code>	<code>SEX_UPON_INTAKE</code>
A434523	Cat	2015-11-20 14:18:00	Normal	NULL	Spayed Female
A562649	Dog	2014-03-20 18:06:00	Sick	NULL	Spayed Female
A524634	Dog	2015-01-02 18:54:00	Normal	*Belle	Intact Female
A465637	Dog	2017-06-04 08:17:00	Injured	*Commander	Neutered Male

이름이 있는 동물의 ID는 A524634와 A465637입니다. 따라서 SQL을 실행하면 다음과 같이 출력되어야 합니다.

<code>ANIMAL_ID</code>
A465637
A524634

=> Solve

solution.sql

```
1 SELECT ANIMAL_ID
2 FROM ANIMAL_INS
3 WHERE NAME IS NOT NULL
4 ORDER BY ANIMAL_ID ASC
```


8. LEVEL 2. 동명 동물 수 찾기

동명 동물 수 찾기

문제 설명

`ANIMAL_INS` 테이블은 동물 보호소에 들어온 동물의 정보를 담은 테이블입니다. `ANIMAL_INS` 테이블 구조는 다음과 같으며, `ANIMAL_ID`, `ANIMAL_TYPE`, `DATETIME`, `INTAKE_CONDITION`, `NAME`, `SEX_UPON_INTAKE` 는 각각 동물의 아이디, 생물 종, 보호 시작일, 보호 시작 시 상태, 이름, 성별 및 중성화 여부를 나타냅니다.

NAME	TYPE	NULLABLE
ANIMAL_ID	VARCHAR(N)	FALSE
ANIMAL_TYPE	VARCHAR(N)	FALSE
DATETIME	DATETIME	FALSE
INTAKE_CONDITION	VARCHAR(N)	FALSE
NAME	VARCHAR(N)	TRUE
SEX_UPON_INTAKE	VARCHAR(N)	FALSE

동물 보호소에 들어온 동물 이름 중 두 번 이상 쓰인 이름과 해당 이름이 쓰인 횟수를 조회하는 SQL문을 작성해주세요. 이때 결과는 이름이 없는 동물은 집계에서 제외하며, 결과는 이름 순으로 조회해주세요.

예시

예를 들어 `ANIMAL_INS` 테이블이 다음과 같다면

ANIMAL_ID	ANIMAL_TYPE	DATETIME	INTAKE_CONDITION	NAME	SEX_UPON_INTAKE
A396810	Dog	2016-08-22 16:13:00	Injured	Raven	Spayed Female
A377750	Dog	2017-10-25 17:17:00	Normal	Lucy	Spayed Female
A355688	Dog	2014-01-26 13:48:00	Normal	Shadow	Neutered Male
A399421	Dog	2015-08-25 14:08:00	Normal	Lucy	Spayed Female
A400680	Dog	2017-06-17 13:29:00	Normal	Lucy	Spayed Female
A410668	Cat	2015-11-19 13:41:00	Normal	Raven	Spayed Female

- Raven 이름은 2번 쓰였습니다.
- Lucy 이름은 3번 쓰였습니다
- Shadow 이름은 1번 쓰였습니다.

따라서 SQL문을 실행하면 다음과 같이 나와야 합니다.

NAME	COUNT
Lucy	3
Raven	2

=> Solve

sql

```
1  SELECT NAME, COUNT(NAME)
2  FROM ANIMAL_INS
3  GROUP BY NAME
4  HAVING COUNT(NAME) >= 2
5
```


9. LEVEL 2. NULL 처리하기

NULL 처리하기

문제 설명

`ANIMAL_INS` 테이블은 동물 보호소에 들어온 동물의 정보를 담은 테이블입니다. `ANIMAL_INS` 테이블 구조는 다음과 같으며, `ANIMAL_ID`, `ANIMAL_TYPE`, `DATETIME`, `INTAKE_CONDITION`, `NAME`, `SEX_UPON_INTAKE` 는 각각 동물의 아이디, 생물 종, 보호 시작일, 보호 시작 시 상태, 이름, 성별 및 중성화 여부를 나타냅니다.

NAME	TYPE	NULLABLE
<code>ANIMAL_ID</code>	<code>VARCHAR(N)</code>	FALSE
<code>ANIMAL_TYPE</code>	<code>VARCHAR(N)</code>	FALSE
<code>DATETIME</code>	<code>DATETIME</code>	FALSE
<code>INTAKE_CONDITION</code>	<code>VARCHAR(N)</code>	FALSE
<code>NAME</code>	<code>VARCHAR(N)</code>	TRUE
<code>SEX_UPON_INTAKE</code>	<code>VARCHAR(N)</code>	FALSE

입양 게시판에 동물 정보를 게시하려 합니다. 동물의 생물 종, 이름, 성별 및 중성화 여부를 아이디 순으로 조회하는 SQL 문을 작성해주세요. 이때 프로그래밍을 모르는 사람들은 NULL이라는 기호를 모르기 때문에, 이름이 없는 동물의 이름은 "No name"으로 표시해 주세요.

예시

예를 들어 `ANIMAL_INS` 테이블이 다음과 같다면

<code>ANIMAL_ID</code>	<code>ANIMAL_TYPE</code>	<code>DATETIME</code>	<code>INTAKE_CONDITION</code>	<code>NAME</code>	<code>SEX_UPON_INTAKE</code>
A350276	Cat	2017-08-13 13:50:00	Normal	Jewel	Spayed Female
A350375	Cat	2017-03-06 15:01:00	Normal	Meo	Neutered Male
A368930	Dog	2014-06-08 13:20:00	Normal	NULL	Spayed Female

마지막 줄의 개는 이름이 없기 때문에, 이 개의 이름은 "No name"으로 표시합니다. 따라서 SQL 문을 실행하면 다음과 같이 나와야 합니다.

<code>ANIMAL_TYPE</code>	<code>NAME</code>	<code>SEX_UPON_INTAKE</code>
Cat	Jewel	Spayed Female
Cat	Meo	Neutered Male
Dog	No name	Spayed Female

※ 컬럼 이름은 일치하지 않아도 됩니다.

=> Solve

solution.sql

```
1 SELECT ANIMAL_TYPE, IFNULL(NAME, "No name") AS NAME, SEX_UPON_INTAKE
2 FROM ANIMAL_INS
3 ORDER BY ANIMAL_ID
```

10. LEVEL 2. 입양 시각 구하기(1)

입양 시각 구하기(1)

문제 설명

`ANIMAL_OUTS` 테이블은 동물 보호소에서 입양 보낸 동물의 정보를 담은 테이블입니다. `ANIMAL_OUTS` 테이블 구조는 다음과 같으며, `ANIMAL_ID`, `ANIMAL_TYPE`, `DATETIME`, `NAME`, `SEX_UPON_OUTCOME`는 각각 동물의 아이디, 생물 종, 입양일, 이름, 성별 및 중성화 여부를 나타냅니다.

NAME	TYPE	NULLABLE
ANIMAL_ID	VARCHAR(N)	FALSE
ANIMAL_TYPE	VARCHAR(N)	FALSE
DATETIME	DATETIME	FALSE
NAME	VARCHAR(N)	TRUE
SEX_UPON_OUTCOME	VARCHAR(N)	FALSE

보호소에서는 몇 시에 입양이 가장 활발하게 일어나는지 알아보려 합니다. 09:00부터 19:59까지, 각 시간대별로 입양이 몇 건이나 발생했는지 조회하는 SQL문을 작성해주세요. 이때 결과는 시간대 순으로 정렬해야 합니다.

예시

SQL문을 실행하면 다음과 같이 나와야 합니다.

hour	count
9	1
10	2
11	13
12	10
13	14
14	9
15	7
16	10
17	12
18	16
19	2

=> Solve

solution.sql

```
1 SELECT HOUR(DATETIME) AS HOUR, COUNT(DATETIME) AS COUNT
2 FROM ANIMAL_OUTS
3 GROUP BY HOUR(DATETIME)
4 HAVING HOUR >= 9 AND HOUR < 20
```