

[20-06-04]

프로그래머스

: Problem Count - 7

1. LEVEL 3. 없어진 기록 찾기

<https://programmers.co.kr/learn/courses/30/lessons/59042>

없어진 기록 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

`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. `ANIMAL_OUTS` 테이블의 `ANIMAL_ID` 는 `ANIMAL_INS` 의 `ANIMAL_ID` 의 외래 키입니다.

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

천재지변으로 인해 일부 데이터가 유실되었습니다. 입양을 간 기록은 있는데, 보호소에 들어온 기록이 없는 동물의 ID와 이름 ID 순으로 조회하는 SQL문을 작성해주세요.

예시

예를 들어, `ANIMAL_INS` 테이블과 `ANIMAL_OUTS` 테이블이 다음과 같다면

`ANIMAL_INS`

ANIMAL_ID	ANIMAL_TYPE	DATETIME	INTAKE_CONDITION	NAME	SEX_UPON_INTAKE
A352713	Cat	2017-04-13 16:29:00	Normal	Gia	Spayed Female
A350375	Cat	2017-03-06 15:01:00	Normal	Meo	Neutered Male

`ANIMAL_OUTS`

ANIMAL_ID	ANIMAL_TYPE	DATETIME	NAME	SEX_UPON_OUTCOME
A349733	Dog	2017-09-27 19:09:00	Allie	Spayed Female
A352713	Cat	2017-04-25 12:25:00	Gia	Spayed Female
A349990	Cat	2018-02-02 14:18:00	Spice	Spayed Female

`ANIMAL_OUTS` 테이블에서

- Allie의 ID는 `ANIMAL_INS` 에 없으므로, Allie의 데이터는 유실되었습니다.
- Gia의 ID는 `ANIMAL_INS` 에 있으므로, Gia의 데이터는 유실되지 않았습니다.
- Spice의 ID는 `ANIMAL_INS` 에 없으므로, Spice의 데이터는 유실되었습니다.

따라서 SQL문을 실행하면 다음과 같이 나와야 합니다.

ANIMAL_ID	NAME
A349733	Allie
A349990	Spice

=> Solve

solution.sql

```
1  SELECT B.ANIMAL_ID, B.NAME
2  FROM ANIMAL_INS A
3  RIGHT JOIN ANIMAL_OUTS B ON B.ANIMAL_ID = A.ANIMAL_ID
4  WHERE A.ANIMAL_ID IS NULL
5  ORDER BY B.ANIMAL_ID;
6
7  SELECT A.ANIMAL_ID, A.NAME
8  FROM ANIMAL_OUTS A
9  LEFT JOIN ANIMAL_INS B ON A.ANIMAL_ID = B.ANIMAL_ID
10 WHERE B.ANIMAL_ID IS NULL
11 ORDER BY A.ANIMAL_ID;
```

2. LEVEL 3. 있었는데요 없었습니다

<https://programmers.co.kr/learn/courses/30/lessons/59043>

있었는데요 없었습니다

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_INTAKE는 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

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_OUTCOME는 각각 동물의 아이디, 생물 종, 입양일, 이름, 성별 및 중성화 여부를 나타냅니다. ANIMAL_OUTS 테이블의 ANIMAL_ID는 ANIMAL_INS의 ANIMAL_ID의 외래 키입니다.

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

관리자의 실수로 일부 동물의 입양일이 잘못 입력되었습니다. 보호 시작일보다 입양일이 더 빠른 동물의 아이디와 이름을 조회하는 SQL문을 작성해주세요. 이때 결과는 보호 시작일이 빠른 순으로 조회해야합니다.

예시

예를 들어, ANIMAL_INS 테이블과 ANIMAL_OUTS 테이블이 다음과 같다면

ANIMAL_INS

ANIMAL_ID	ANIMAL_TYPE	DATETIME	INTAKE_CONDITION	NAME	SEX_UPON_INTAKE
A350276	Cat	2017-08-13 13:50:00	Normal	Jewel	Spayed Female
A381217	Dog	2017-07-08 09:41:00	Sick	Cherokee	Neutered Male

ANIMAL_OUTS

ANIMAL_ID	ANIMAL_TYPE	DATETIME	NAME	SEX_UPON_OUTCOME
A350276	Cat	2018-01-28 17:51:00	Jewel	Spayed Female
A381217	Dog	2017-06-09 18:51:00	Cherokee	Neutered Male

SQL문을 실행하면 다음과 같이 나와야 합니다.

ANIMAL_ID	NAME
A381217	Cherokee

=> Solve

solution.sql

```
1  SELECT A.ANIMAL_ID, A.NAME
2  FROM ANIMAL_INS A
3  JOIN ANIMAL_OUTS B ON A.ANIMAL_ID=B.ANIMAL_ID
4  WHERE A.DATETIME > B.DATETIME
5  ORDER BY A.DATETIME
```

3. LEVEL 3. 오랜 기간 보호한 동물(1)

<https://programmers.co.kr/learn/courses/30/lessons/59044>

오랜 기간 보호한 동물(1)

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

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. **ANIMAL_OUTS** 테이블의 **ANIMAL_ID** 는 **ANIMAL_INS** 의 **ANIMAL_ID** 의 외래 키입니다.

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

아직 입양을 못 간 동물 중, 가장 오래 보호소에 있었던 동물 3마리의 이름과 보호 시작일을 조회하는 SQL문을 작성해주세요. 이때 결과는 보호 시작일 순으로 조회해야 합니다.

예시

예를 들어, **ANIMAL_INS** 테이블과 **ANIMAL_OUTS** 테이블이 다음과 같다면

ANIMAL_INS

ANIMAL_ID	ANIMAL_TYPE	DATETIME	INTAKE_CONDITION	NAME	SEX_UPON_INTAKE
A354597	Cat	2014-05-02 12:16:00	Normal	Ariel	Spayed Female
A373687	Dog	2014-03-20 12:31:00	Normal	Rosie	Spayed Female
A412697	Dog	2016-01-03 16:25:00	Normal	Jackie	Neutered Male
A413789	Dog	2016-04-19 13:28:00	Normal	Benji	Spayed Female
A414198	Dog	2015-01-29 15:01:00	Normal	Shelly	Spayed Female

ANIMAL_OUTS

ANIMAL_ID	ANIMAL_TYPE	DATETIME	NAME	SEX_UPON_OUTCOME
A354597	Cat	2014-05-02 12:16:00	Ariel	Spayed Female
A373687	Dog	2014-03-20 12:31:00	Rosie	Spayed Female

SQL문을 실행하면 다음과 같이 나와야 합니다.

NAME	DATETIME
Shelly	2015-01-29 15:01:00
Jackie	2016-01-03 16:25:00
Benji	2016-04-19 13:28:00

* 입양을 가지 못한 동물이 3마리 이상인 경우만 입력으로 주어집니다.

=> Solve

solution.sql

```
1 SELECT A.NAME, A.DATETIME
2 FROM ANIMAL_INS A
3 LEFT JOIN ANIMAL_OUTS B ON A.ANIMAL_ID = B.ANIMAL_ID
4 WHERE B.ANIMAL_ID IS NULL
5 ORDER BY A.DATETIME
6 LIMIT 3;
```

4. LEVEL 3. 오랜 기간 보호한 동물(2)

<https://programmers.co.kr/learn/courses/30/lessons/59411>

오랜 기간 보호한 동물(2)

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

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. ANIMAL_OUTS 테이블의 ANIMAL_ID 는 ANIMAL_INS 의 ANIMAL_ID 의 외래 키입니다.

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

입양을 간 동물 중, 보호 기간이 가장 길었던 동물 두 마리의 아이디와 이름을 조회하는 SQL문을 작성해주세요. 이때 결과는 보호 기간이 긴 순으로 조회해야 합니다.

예시

예를 들어, ANIMAL_INS 테이블과 ANIMAL_OUTS 테이블이 다음과 같다면

ANIMAL_INS

ANIMAL_ID	ANIMAL_TYPE	DATETIME	INTAKE_CONDITION	NAME	SEX_UPON_INTAKE
A354597	Cat	2014-05-02 12:16:00	Normal	Ariel	Spayed Female
A362707	Dog	2016-01-27 12:27:00	Sick	Girly Girl	Spayed Female
A370507	Cat	2014-10-27 14:43:00	Normal	Emily	Spayed Female
A414513	Dog	2016-06-07 09:17:00	Normal	Rocky	Neutered Male

ANIMAL_OUTS

ANIMAL_ID	ANIMAL_TYPE	DATETIME	NAME	SEX_UPON_OUTCOME
A354597	Cat	2014-06-03 12:30:00	Ariel	Spayed Female
A362707	Dog	2017-01-10 10:44:00	Girly Girl	Spayed Female
A370507	Cat	2015-08-15 09:24:00	Emily	Spayed Female

SQL문을 실행하면 다음과 같이 나와야 합니다.

ANIMAL_ID	NAME
A362707	Girly Girl
A370507	Emily

* 입양을 간 동물이 2마리 이상인 경우만 입력으로 주어집니다.

=> Solve

solution.sql

```
1 SELECT A.ANIMAL_ID, A.NAME
2 FROM ANIMAL_OUTS A
3 JOIN ANIMAL_INS B ON A.ANIMAL_ID=B.ANIMAL_ID
4 ORDER BY A.DATETIME-B.DATETIME DESC
5 LIMIT 2;
```

5. LEVEL 4. 입양 시각 구하기

<https://programmers.co.kr/learn/courses/30/lessons/59413>

입양 시각 구하기(2)

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. 0시부터 23시까지, 각 시간대별로 입양이 몇 건이나 발생했는지 조회하는 SQL문을 작성해주세요. 이때 결과는 시간대 순으로 정렬해야 합니다.

예시

SQL문을 실행하면 다음과 같이 나와야 합니다.

hour	count
0	0
1	0
2	0
3	0
4	0
5	0
6	0

=> Solve

```
solution.sql

1  SET @CUR = -1 ;
2  SELECT (@CUR := @CUR +1) AS HOUR,
3  (
4      SELECT COUNT(DATETIME) AS COUNT
5      FROM ANIMAL_OUTS
6      WHERE HOUR(DATETIME) = @CUR
7  ) AS COUNT
8  FROM ANIMAL_OUTS
9  WHERE @CUR <23 ;
```

6. LEVEL 4. 보호소에서 중성화한 동물

<https://programmers.co.kr/learn/courses/30/lessons/59045>

보호소에서 중성화한 동물

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

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. ANIMAL_OUTS 테이블의 ANIMAL_ID 는 ANIMAL_INS 의 ANIMAL_ID 의 외래 키입니다.

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

보호소에서 중성화 수술을 거친 동물 정보를 알아보려 합니다. 보호소에 들어올 당시에는 중성화`되지 않았지만, 보호소를 나갈 당시에는 중성화된 동물의 아이디와 생물 종, 이름을 조회하는 아이디 순으로 조회하는 SQL 문을 작성해주세요.

예시

예를 들어, ANIMAL_INS 테이블과 ANIMAL_OUTS 테이블이 다음과 같다면

ANIMAL_INS

ANIMAL_ID	ANIMAL_TYPE	DATETIME	INTAKE_CONDITION	NAME	SEX_UPON_INTAKE
A367438	Dog	2015-09-10 16:01:00	Normal	Cookie	Spayed Female
A382192	Dog	2015-03-13 13:14:00	Normal	Maxwell 2	Intact Male
A405494	Dog	2014-05-16 14:17:00	Normal	Kaila	Spayed Female
A410330	Dog	2016-09-11 14:09:00	Sick	Chewy	Intact Female

ANIMAL_OUTS

ANIMAL_ID	ANIMAL_TYPE	DATETIME	NAME	SEX_UPON_OUTCOME
A367438	Dog	2015-09-12 13:30:00	Cookie	Spayed Female
A382192	Dog	2015-03-16 13:46:00	Maxwell 2	Neutered Male
A405494	Dog	2014-05-20 11:44:00	Kaila	Spayed Female
A410330	Dog	2016-09-13 13:46:00	Chewy	Spayed Female

- Cookie는 보호소에 들어올 당시에 이미 중성화되어있었습니다.
- Maxwell 2는 보호소에 들어온 후 중성화되었습니다.
- Kaila는 보호소에 들어올 당시에 이미 중성화되어있었습니다.
- Chewy는 보호소에 들어온 후 중성화되었습니다.

따라서 SQL문을 실행하면 다음과 같이 나와야 합니다.

ANIMAL_ID	ANIMAL_TYPE	NAME
A382192	Dog	Maxwell 2
A410330	Dog	Chewy

=> Solve

solution.sql

```
1  SELECT O.ANIMAL_ID,O.ANIMAL_TYPE,O.NAME
2  FROM
3  (
4      SELECT*
5      FROM ANIMAL_OUTS
6      WHERE SEX_UPON_OUTCOME LIKE 'Spayed%' OR SEX_UPON_OUTCOME LIKE 'Neutered%'
7  ) AS O
8  JOIN(
9      SELECT *
10     FROM ANIMAL_INS
11     WHERE SEX_UPON_INTAKE LIKE 'Intact%'
12 ) A ON A.ANIMAL_ID = O.ANIMAL_ID
13 ORDER BY O.ANIMAL_ID
```

solution.sql

```
1  SELECT O.ANIMAL_ID,O.ANIMAL_TYPE,O.NAME
2  FROM ANIMAL_OUTS O
3  RIGHT JOIN(
4      SELECT *
5      FROM ANIMAL_INS
6      WHERE SEX_UPON_INTAKE LIKE 'Intact%'
7  ) A ON A.ANIMAL_ID = O.ANIMAL_ID
8  WHERE O.SEX_UPON_OUTCOME LIKE 'Spayed%' OR O.SEX_UPON_OUTCOME LIKE 'Neutered%'
9  ORDER BY O.ANIMAL_ID
```

7. LEVEL 4. 우유와 요거트가 담긴 장바구니

<https://programmers.co.kr/learn/courses/30/lessons/62284>

우유와 요거트가 담긴 장바구니

문제 설명

`CART_PRODUCTS` 테이블은 장바구니에 담긴 상품 정보를 담은 테이블입니다. `CART_PRODUCTS` 테이블의 구조는 다음과 같으며, `ID`, `CART_ID`, `NAME`, `PRICE` 는 각각 테이블의 아이디, 장바구니의 아이디, 상품 종류, 가격을 나타냅니다.

NAME	TYPE
ID	INT
CART_ID	INT
NAME	VARCHAR
PRICE	INT

데이터 분석 팀에서는 우유와 요거트를 동시에 구입한 장바구니가 있는지 알아보려 합니다. 우유와 요거트를 동시에 구입한 장바구니의 아이디를 조회하는 SQL 문을 작성해주세요. 이때 결과는 장바구니의 아이디 순으로 나와야 합니다.

예시

예를 들어 `CART_PRODUCTS` 테이블이 다음과 같다면

CART_PRODUCTS 테이블			
ID	CART_ID	NAME	PRICE
5488	83	알루미늄 호일	2980
8435	83	우유	1880
5489	286	반찬류	3700
5491	286	요거트	2980
5504	286	우유	1880
8437	448	요거트	2980
8454	448	우유	1880
5490	1034	케첩	1380
5492	1034	일회용품	1080

- 83번 장바구니에는 우유만 있고 요거트가 없습니다.
- 286번 장바구니에는 우유와 요거트가 모두 있습니다.
- 448번 장바구니에는 우유와 요거트가 모두 있습니다.
- 1034번 장바구니에는 우유와 요거트가 모두 없습니다.

따라서 SQL 문을 실행하면 다음과 같이 나와야 합니다.

CART_ID
286
448

=> Solve

solution.sql

```
1  SELECT A.CART_ID
2  FROM (
3      SELECT*
4      FROM CART_PRODUCTS
5      WHERE NAME='우유'
6  ) A, (
7      SELECT*
8      FROM CART_PRODUCTS
9      WHERE NAME='요거트'
10 ) B
11 WHERE A.CART_ID = B.CART_ID
12 ORDER BY A.ID
```