SQL

- 정의 : SQL(/ˈɛs kjuː ˈɛl/,[1] or /ˈsiːkwəl/, Structured Query Language[2][3][4][5], 구조화 질의어, S-Q-L[6])는 관계형 데이터베이스 관리 시스템(RDBMS)의 데이터를 관리하기 위해 설계된 **특수 목적의 프로그래밍 언어**
- SQL은 문법의 3가지로 구분
  1. 데이터 정의 언어 (DDL)
  - CREATE, DROP, ALTER
  1. 데이터 조작 언어 (DML)
  - INSERT INTO, UPDATE ~SET, DELETE FROM, SELECT ~FROM ~WHERE

  3. 데이터 제어 언어 (DCL)
  - GRANT, REVOKE, SET TRANSACTION

프로그래머스 SQL

SELECT

- SELECT * FROM TABLE WHERE 조건 ORDER BY 기준 컬럼명, 다음 컬럼명 (ASC - DEFAULT, DESC 내림차순)

EX) 모든 동물의 아이디와 이름, 보호 시작일을 이름 순으로 조회, 단 이름이 같은 동물 중에서는 보호를 나중에 시작한 동물을 먼저 보여야 함

- 해결 방법

1. 이름을 먼저 사전 순으로 정렬, ORDER BY NAME
2. 이름이 같을 경우에는 나중에 온 동물이 먼저 나와야 되기 때문에 ORDER BY NAME, **DATETIME DESC**

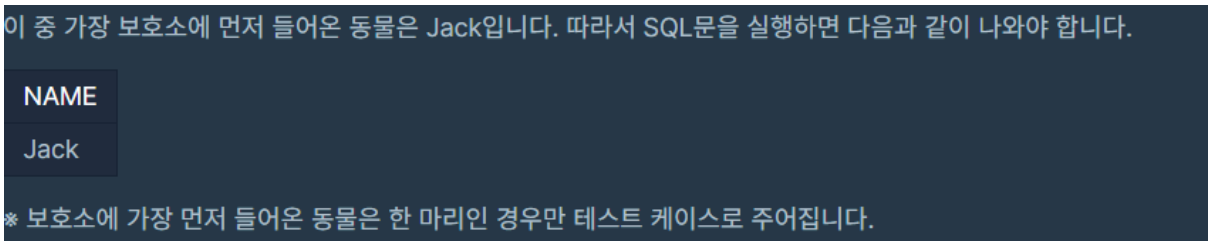
- - 코드를 입력하세요
  SELECT ANIMAL_ID, NAME, DATETIME FROM ANIMAL_INS ORDER BY NAME, DATETIME DESC

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

- 상위 N개의 레코드

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

- SELECT * FROM TABLE ORDER BY 컬럼명 LIMIT 갯수
- SELECT NAME FROM ANIMAL_INS ORDER BY DATETIME LIMIT 1



SUM, MAX, MIN, COUNT

- 가장 최근에 들어온 동물은 언제 들어왔는지 조회하는 SQL 문을 작성해주세요.
- SELECT MAX(DATETIME) FROM ANIMAL_INS
- SELECT DATETIME FROM ANIMAL_INS ORDER BY DATETIME DESC LIMIT 1

- 중복제거, 및 개수 세기
- 동물 보호소에 들어온 동물의 이름은 몇 개인지 조회하는 SQL 문을 작성해주세요. 이때 이름이 NULL인 경우는 집계하지 않으며 중복되는 이름은 하나로 칩니다
- DISTINCT는 중복 제거

- SELECT COUNT(DISTINCT NAME) FROM ANIMAL_INS

GROUP BY

- COUNT 함수로 데이터를 조회하면 전체 갯수만을 가져온다.
- GROUP BY : 유형별로 갯수를 알고 싶을 때는 컬럼에 데이터를 그룹화 할 수 있는 GROUP BY를 사용

  특정 컬럼을 그룹화 하는 GROUP BY

  특정 컬럼을 그룹화한 결과에 조건을 거는 HAVING
- WHERE, HAVING 차이

  WHERE 는 그룹화 하기 전, HAVING은 그룹화 후에 조건
- SELECT ANIMAL_TYPE, COUNT(ANIMAL_TYPE) as count FROM ANIMAL_INS GROUP BY ANIMAL_TYPE ORDER BY ANIMAL_TYPE
- 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.
- SELECT HOUR(DATETIME) AS HOUR, COUNT(HOUR(DATETIME)) AS COUNT FROM ANIMAL_OUTS GROUP BY HOUR HAVING HOUR >=9 AND HOUR <=19 ORDER BY HOUR ASC

JOIN

- 여러 테이블에 흩어져 있는 정보 중 사용자가 필요한 정보만 가져와서 가상의 테이블처럼 만들어서 결과를 보여주는 것, 2개의 테이블을 조합하여 하나의 열로 표현하는 것
- 종류는 4가지
  1. INNER JOIN
  2. CROSS JOIN
  3. OUTER JOIN
  4. SELF JOIN

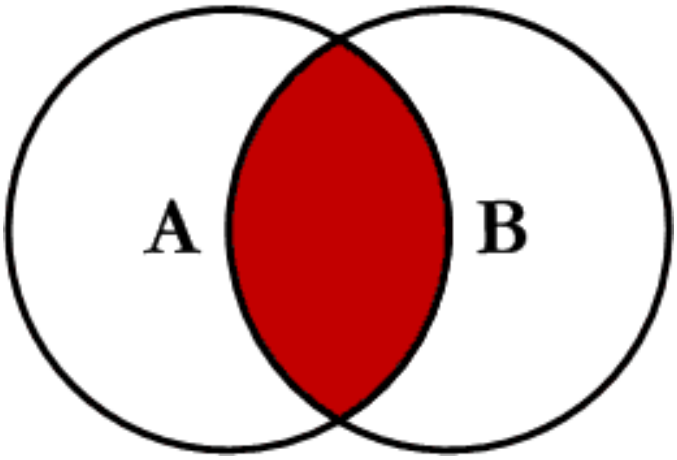
- INNER JOIN(내부 조인)

- 키 값이 있는 테이블의 칼럼 값을 비교 후 조건에 맞는 값을 가져오는 것, 서로 연관된 내용만 검색
- 방법에 따라 2가지가 존재
- 명시적 조인 : JOIN 키워드를 사용하고 ON 키워드를 사용

  EX) SELECT * FROM EMPLOYEE INNER JOIN DEPARTMENT ON EMPLOYEE.DEPARTMENTID=DEPARTMENT.DEPARTMENTID

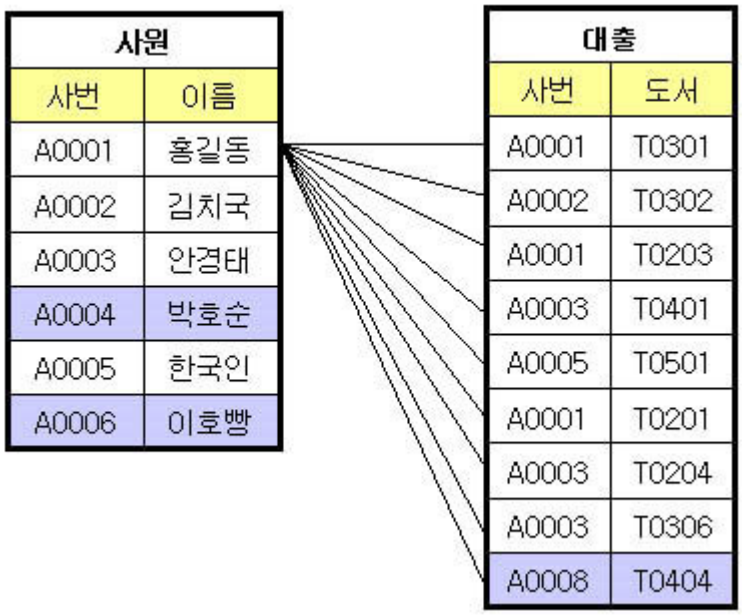
  SELECT I.ANIMAL_ID, I.NAME
  FROM ANIMAL_INS I
  JOIN ANIMAL_OUTS O
  ON I.ANIMAL_ID = O.ANIMAL_ID
  ORDER BY DATEDIFF(O.DATETIME,I.DATETIME)
  LIMIT 2
- 암시적 조인 : SELECT 구문의 FROM절에서 ,를 사용하여 단순히 조인을 위한 여러 테이블을 나열

  EX) SELECT * FROM EMPLOYEE, DEPARTMENT WHERE EMPLOYEE.DPARTMENTID=DEPARTMENT.DEPARTMENTID



- CROSS JOIN
  - 카디션 곱이라고 하며, 조인되는 두 테이블에서 곱집합을 반환
  - M열을 가진 테이블 * N열을 가진 테이블이 교차되면 M*N개의 테이블이 생성
  - 6*8 =48 개 생성

- SELELCT * FROM EMPLOYEE CROSS JOIN DEPARTMENT



사원

사번	이름
A0001	홍길동
A0002	김치국
A0003	안경태
A0004	박호순
A0005	한국인
A0006	이호빵

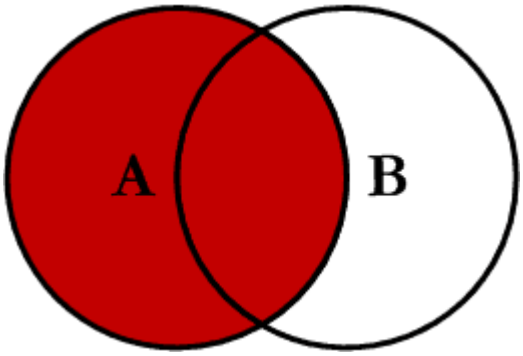
대출

사번	도서
A0001	T0301
A0002	T0302
A0001	T0203
A0003	T0401
A0005	T0501
A0001	T0201
A0003	T0204
A0003	T0306
A0008	T0404

- OUTER JOIN
  - 여러 테이블에서 한 쪽에는 데이터가 있고 한 쪽에는 데이터가 없는 경우, 데이터가 있는 쪽 테이블의 내용을 전부 출력하는 방법
  - 즉, 조인 조건에 만족하지 않아도 해당 행을 출력하고 싶을 때 사용
  - 종류로는 LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN

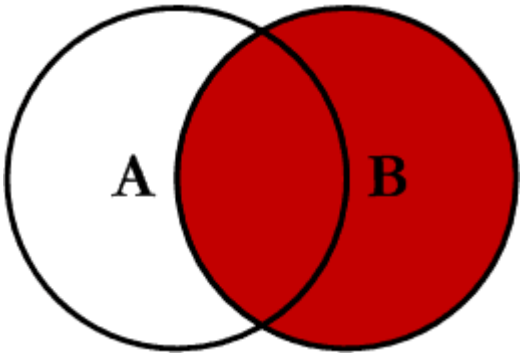
1. LEFT OUTER JOIN
   - 조인 문의 왼쪽에 있는 테이블의 모든 결과를 가져온 후 오른쪽 테이블의 데이터를 매칭하고, 데이터가 없는 경우 NULL을 표시

   SELECT * FROM EMPLOYEE E LEFT OUTER JOIN DEPART D ON E.DEPARTMETNID=D.DEPARTMENTID



2. RIGHT OUTER JOIN

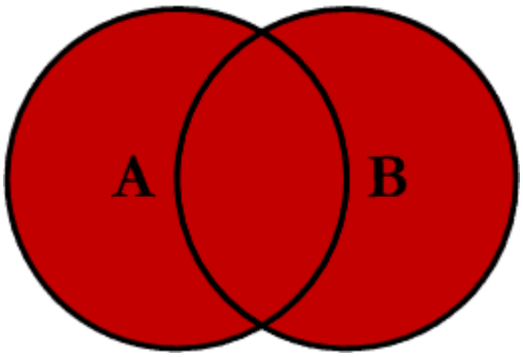
- 조인 문의 오른쪽에 있는 테이블의 모든 결과를 가져온 후 왼쪽 테이블의 데이터를 매칭하고, 데이터가 없는 경우 NULL을 표시



SELECT * FROM EMPLOYEE E RIGHT OUTER JOIN DEPARTMENT D ON E.DepartmentID = D.DepartmentID;

3. FULL OUTER JOIN

- Full Outer Join은 LEFT OUTER JOIN과 RIGHT OUTER JOIN을 합친 것이다.
- 양쪽 모두 조건이 일치하지 않는 것들까지 모두 결합하여 출력



SELECT * FROM EMPLOYEE E FULL OUTER JOIN DEPARTMENT D ON E.DepartmentID = D.DepartmentID

4. SELF JOIN

- 자기자신을 JOIN
- **select F.empno, F.ename, S.empno, S.ename, S.job**

from emp F inner join Emp S

on F.job = S.job

where F.empno < S.empno

order by F.empno, S.empno;

특정 문자열

- LIKE는 특정 문자열을 포함 시킬 수 있음

  SELECT ANIMAL_ID, NAME FROM ANIMAL_INS **WHERE NAME LIKE '%EL%'** AND ANIMAL_TYPE='DOG' ORDER BY NAME

- 여러 문자열을 포함하고 싶을 때는 IN을 사용

  SELECT ANIMAL_ID, NAME, SEX_UPON_INTAKE FROM ANIMAL_INS WHERE **NAME IN('LUCY','ELLA','PICKLE','ROGAN','SABRINA','MITTY')** ORDER BY ANIMAL_ID

출처:

https://clairdelunes.tistory.com/22

[거꾸로 바라본 세상]