SQL 개념 및 실습 기초

박데이터운영실 데이터분석지원부 박성용 주임연구원

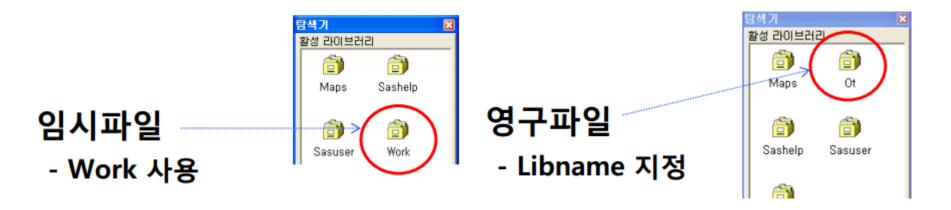


라이브러리 지정

라이브러리 지정 : 자료의 저장 및 저장된 자료의 호출을 위해 사용

→ LIBNAME 라이브러리 이름 '위치' ;

예시) LIBNAME NHIS 'C:\USERS\DESKTOP\실습';



DATA STEP

- SET 구문
 - DATA ABC; SET TEST1; RUN;
- IF ··· THEN ··· 구문
 - DATA ABC; SET TEST1; IF SEX=1 THEN DELETE; RUN;
- KEEP 구문
 - DATA ABC; SET TEST1; KEEP AGE; RUN;
- DROP 구문
 - DATA ABC; SET TEST1; DROP AGE; RUN;
- RENAME 구문
 - DATA ABC(RENAME=(SEX=GENDER)); SET TEST1; RUN;

ID	SEX	AGE
1	1	45
2	1	54
3	2	64
4	2	13

DATA TEST1; INPUT ID SEX \$ AGE; CARDS; 1 1 45 2 1 54 3 2 64 4 2 13 RUN;

SQL 개요

SQL(Structured Query Language) : 여러 데이터셋을 조합해서 사용자가 원하는 변수 및 값을 추출하는 Database 언어

WHY SQL?

- 원하는 변수만 취사 선택 가능
- KEY 변수명이 같지 않아도 연결 가능
- 조인 방식(LEFT, RIGHT, INNER, FULL) 설정 가능
- 요약함수(COUNT, MIN, MAX, SUM, AVG) 사용 가능
- 자료단위 병합 시 GROUP 옵션으로 효율적 병합 가능

SQL 개요

기본구조

PROC SQL

CREATE TABLE AS	새로 만드는 테이블 명	
SELECT FROM	SELECT FROM	
WHERE	조건에 따른 테이블의 row 선택	
GROUP BY	칼럼의 값에 기반하여 데이터 그룹핑	
HAVING	요약된 값에 대해 조건 부여	
ORDER BY	칼럼 값 순으로 정렬	

QUIT;

데이터의 조회 및 검색

SQL 설명

DISTINCT

중복을 제거한 컬럼 값

- SELECT **DISTINCT** 컬럼 FROM 테이블 WHERE 조건

GROUP BY

지정된 칼럼 내 값에 따라 데이터 그룹핑

- SELECT 컬럼 FROM 테이블 GROUP BY 컬럼

HAVING

GROUP BY 결과에 대한 조건

- SELECT 컬럼 FROM 테이블 GROUP BY 컬럼 HAVING 조건

ORDER BY

칼럼 값 순으로 정렬(DEFAULT: 오름차순)

- SELECT 컬럼 FROM 테이블 ORDER BY 컬럼 ※ ORDER BY 컬럼 DESC: 내림차순

※ GROUP BY , ORDER BY 와 자주 쓰이는 집계 함수

AVG, MEAN: 평균

COUNT: 결측 아닌 건 수

MAX : 최대값 MIN: 최소값 SUM: 합계



PROC SQL;

SELECT VAR, AVG(ID)

FROM TABLE NAME **GROUP BY VAR**

ORDER BY VAR;

QUIT;

데이터의 조회 및 검색

SQL 설명

- IS (NOT) NULL 결측인(이 아닌) 경우만
 - SELECT 컬럼 FROM 테이블 WHERE 컬럼 IS (NOT) NULL
- = / ^= OR 〈〉 1개 조건이 같다/같지 않다.
 - SELECT 컬럼1 FROM 테이블 WHERE 컬럼1=컬럼2
 - SELECT 컬럼1 FROM 테이블 WHERE 컬럼1^=컬럼2 (혹은 컬럼1<>컬럼2)
- BETWEEN A AND B A와 B를 포함한 범위의 값
 - SELECT 컬럼 FROM 테이블 WHERE 컬럼 BETWEEN A AND B
- IN / NOT IN 1개 이상 포함되는 조건 / 포함되지 않는 조건
 - SELECT 컬럼 FROM 테이블 WHERE 컬럼 IN (조건) / NOT IN (조건)

데이터의 조회 및 검색

WHERE 절을 이용하여 선택할 행에 대한 조건 표현식 기술 가능

- WHERE 컬럼 LIKE '%문자열%'
 - '문자열'을 포함한 행만 선택
- WHERE 컬럼 IS MISSING / IS NULL
 - MISSING인 행만 선택
- WHERE 컬럼 BETWEEN A AND B
 - A와 B사이만 선택
- WHERE 컬럼 = 'VALUE'
 - 하나의 조건인 경우
- WHERE 컬럼 IN ('VALUE1', 'VALUE2')
 - 여러 조건인 경우

UNION 과 UNION ALL

UNION과 UNION ALL은 데이터의 합집합 개념.

→ 즉, 두 개 이상의 테이블을 합한 결과물을 보고자 할 때 사용

TABLE 1 TABLE 2

ID ID 2
2 3

PROC SQL;

SELECT ID FROM TABLE 1

UNION

SELECT ID FROM TABLE 2; QUIT:

ID 1

2 3

PROC SQL;

SELECT ID FROM TABLE 1

UNION ALL

SELECT ID FROM TABLE 2; QUIT;

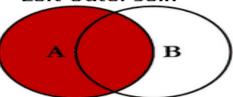
데이터의 JOIN

SQL 설명

- INNER JOIN 두 테이블에 모두 PRIMARY KEY 값이 존재하는 경우만 반환
 - SELECT A.컬럼 FROM 테이블1 AS A INNER JOIN 테이블2 AS B ON A.PRIMARY KEY=B.PRIMARY KEY
- LEFT JOIN 먼저 나열된(왼쪽테이블) 기준으로 조인할 테이블에 KEY 값이 존재하면 반환, 아니면 NULL 처리
 - SELECT A.컬럼 FROM 테이블1 AS A LEFT JOIN 테이블2 AS B ON A.PRIMARY KEY=B.PRIMARY KEY
- RIGHT JOIN 후순위로 나열된(오른쪽테이블) 기준으로 조인할 테이블에 KEY 값이 존재하면 반환, 아니면 NULL 처리
 - SELECT A.컬럼 FROM 테이블1 AS A **RIGHT JOIN** 테이블2 AS B ON A.PRIMARY KEY=B.PRIMARY KEY
- FULL JOIN 양쪽 모두 조건이 일치하지 않는 것들까지 모두 결합하여 반환
 - SELECT A.컬럼 FROM 테이블1 AS A FULL JOIN 테이블2 AS B ON A.PRIMARY KEY=B.PRIMARY KEY

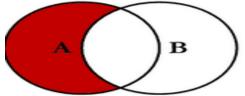
데이터의 JOIN

Left Outer Join



SELECT < select_list> FROM Table_A A LEFT JOIN Table_B B ON A.Key = B.Key

Left Excluding Join

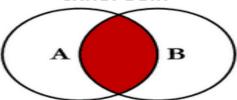


SELECT <select_list> FROM Table_A A LEFT JOIN Table_B B ON A.Key = B.Key WHERE B.Key IS NULL

SELECT <select_list> FROM Table_A A FULL OUTER JOIN TUBLE_B B ON A.Key = B.Key

SQL JOINS

Inner Join



SELECT <select_list> FROM Table_A A INNER JOIN Table_B B ON A.Key = B.Key

OUTER JOIN or FULL OUTER JOIN or FULL JOIN

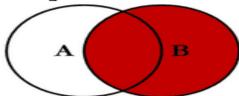


) (A (

Outer Excluding Join

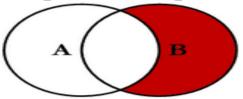
 \mathbf{B}

Right Outer Join



SELECT <select_list> FROM Table_A A RIGHT JOIN Table_B B ON A.Key = B.Key

Right Excluding Join



SELECT <select_list> FROM Table_A A RIGHT JOIN Table_B B ON A.Key = B.Key WHERE A.Key IS NULL

SELECT <select_list>
FROM Table_A A
FULL OUTER JOIN Table_B B
ON A.Key = B.Key
WHERE A.Key IS NULL OR
B.Key IS NULL

'A[.] & 'B' are two sets.

- 1. AnB = Inner Join('n' intersection)
- 2. Au(AnB) = Left Join ('u' Union)
- 3. (AnB)uB = Rigjt Join
- 4. AuBu(AnB) = Outer Join
- 5. A- B = Left Join Excluding Inner Join or Relative Component
- 6. B- A = Right Join Excluding Ineer Join
- 7. (A-B)u(B-A) = Outer Join Excluding Inner Join