



SQL을 이용한 데이터 핸들링 : 기초부터 실전까지 (기초)

목차

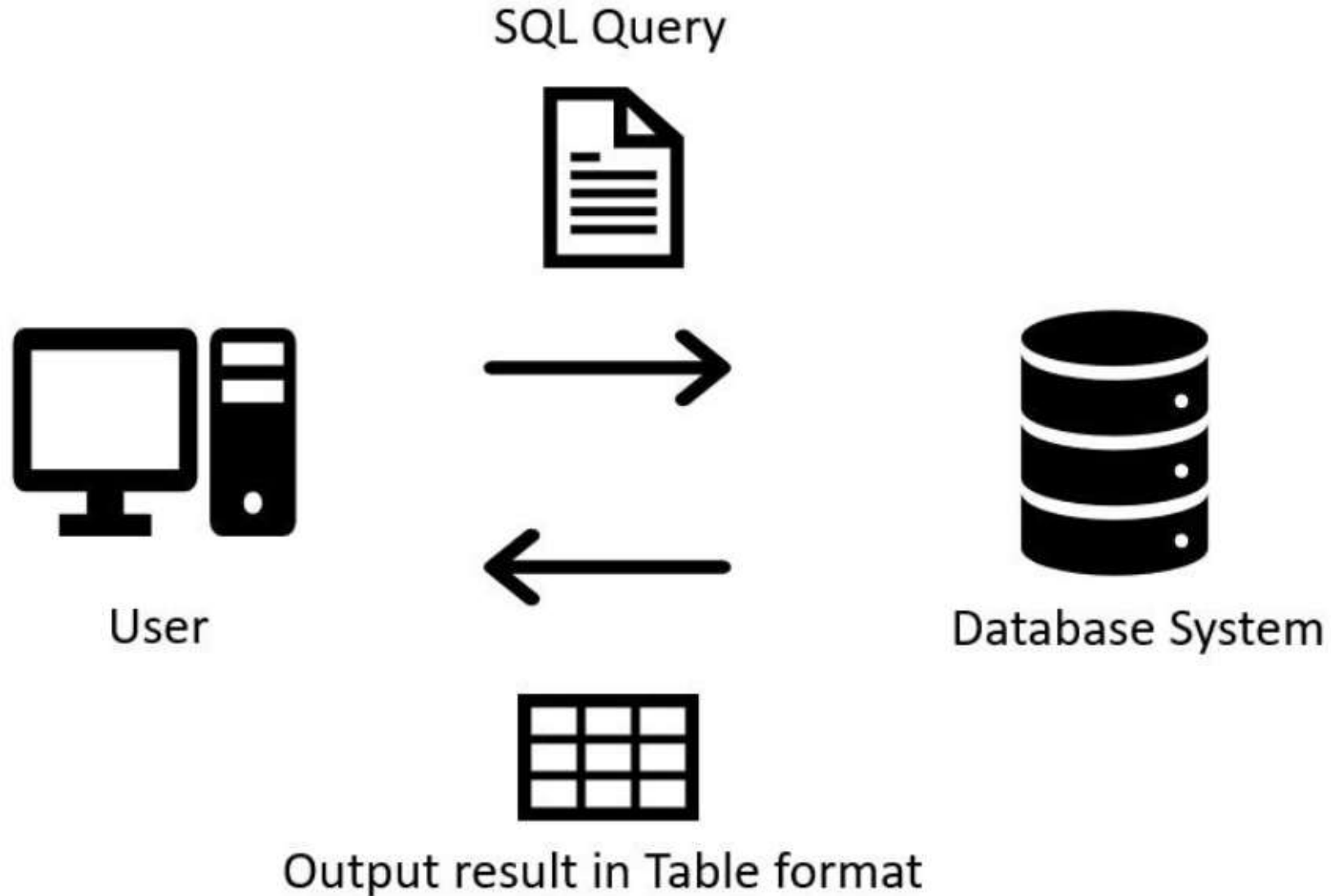
1회차 : SQL기본 문법

- SQL 개요 및 활용
- SQL을 이용한 데이터 조회(비교연산자, 논리연산자 활용)
- 조건에 맞는 데이터 그룹핑 및 정렬(Group by, Having, Order by)
- 데이터 결합하기 (Inner Join, Outer Join)
- 빅데이터센터 플랫폼 안내

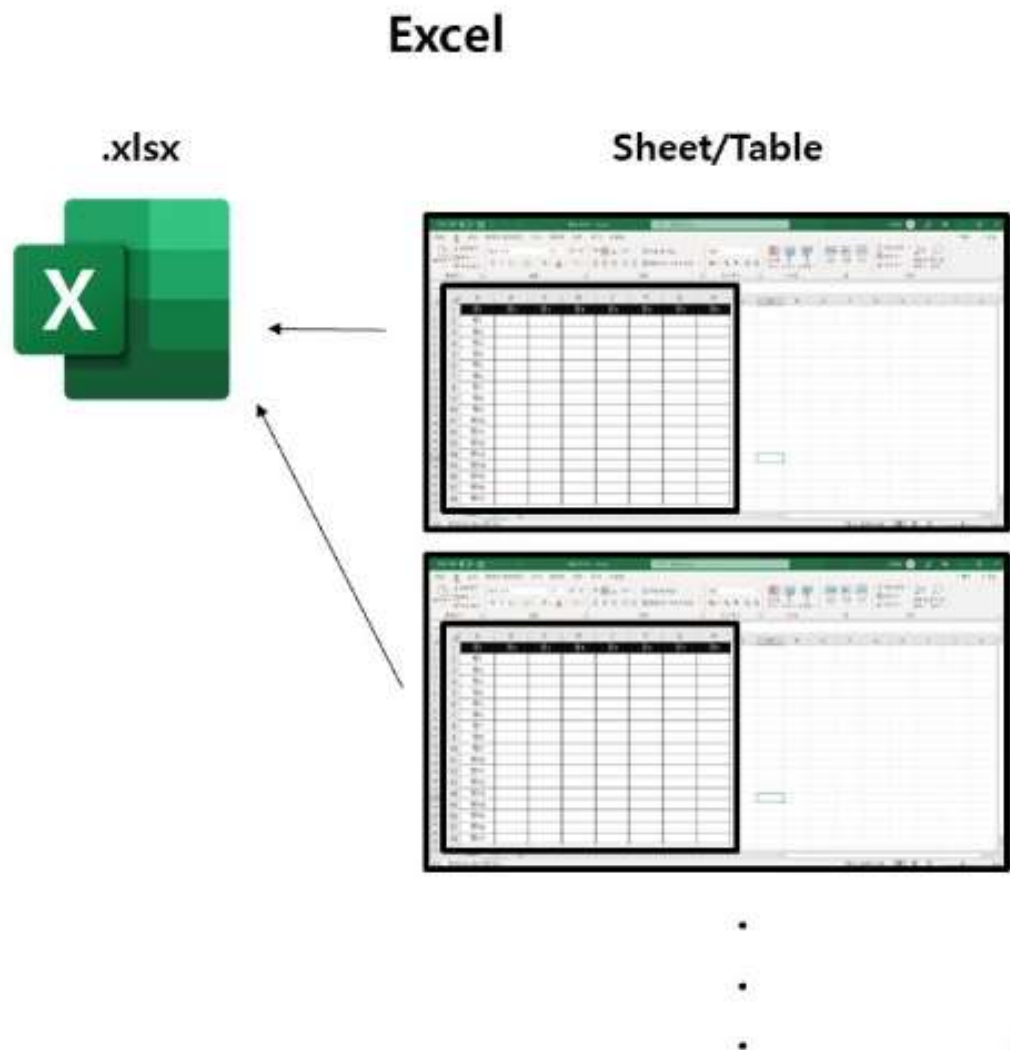
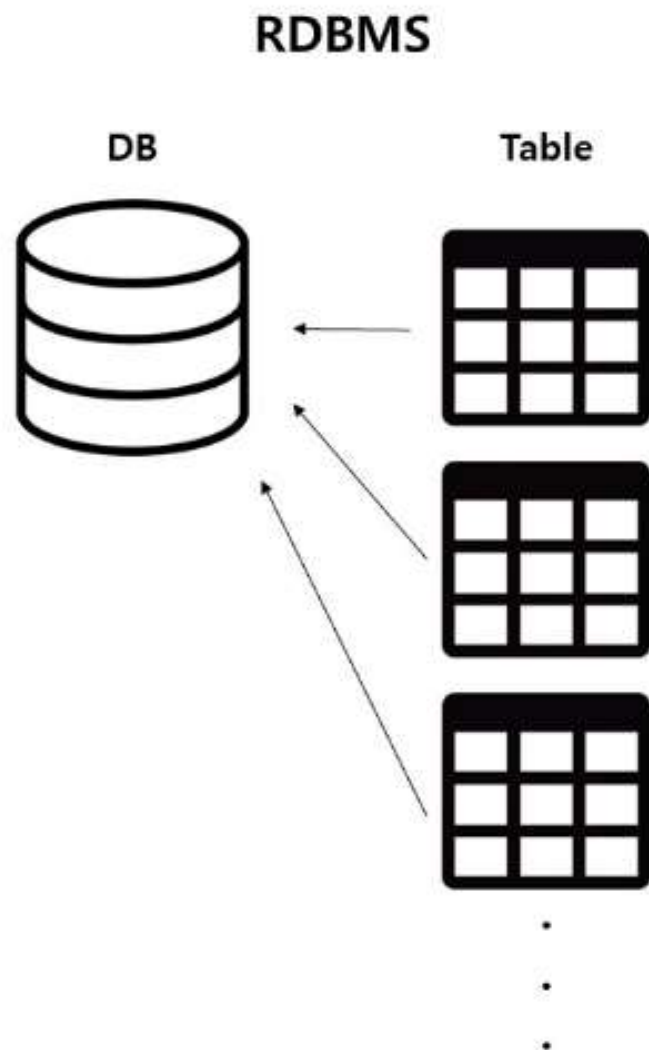
2회차 : 대장암 환자 데이터 추출

- 원내 의료 데이터 소개
- 가상 시나리오 소개
- 시나리오에 따른 실습 코드 소개

SQL이란 무엇인가요?



DB란 무엇인가요?



어느 파일과 시트에서
데이터를 조회할지
선택하듯이

데이터베이스에 있는
수많은 테이블 중
어떤 테이블에서
데이터를 조회할 것인가?

테이블 구조 파악하기

	A	B	C	D	E	F
1	person_id	sex	height	weight	sbp	dbp
2	1	남자	180	70	98	65
3	2	여자	162	50	100	69
4	3	여자	158	45	110	70
5	4	남자	172	58	101	71
6	5	여자	166	48	109	63
7	6	남자	176	60	98	64
8	7	남자	174	68	97	59
9	8	남자	178	70	103	60
10	9	여자	163	50	108	65
11	10	여자	170	57	99	67

< >

Vital sheet

Lab data

+

⋮

<

← Column name

↑ Table name

SQL 문법_SELECT, FROM

❖ SELECT + FROM

1	person_id	sex	height	weight	sbp	dbp
2	1	남자	180	70	98	65
3	2	여자	162	50	100	69
4	3	여자	158	45	110	70
5	4	남자	172	58	101	71
6	5	여자	166	48	109	63
7	6	남자	176	60	98	64
8	7	남자	174	68	97	59
9	8	남자	178	70	103	60
10	9	여자	163	50	108	65
11	10	여자	170	57	99	67

< >

Vital sheet

Lab data

+

⋮

<

← Select 컬럼 이름

↑ From 테이블 이름

SQL 문법_SELECT, FROM

❖ SELECT + FROM

- person_id, sbp, dbp만 조회할 때

1	person_id	sex	height	weight	sbp	dbp
2	1	남자	180	70	98	65
3	2	여자	162	50	100	69
4	3	여자	158	45	110	70
5	4	남자	172	58	101	71
6	5	여자	166	48	109	63
7	6	남자	176	60	98	64
8	7	남자	174	68	97	59
9	8	남자	178	70	103	60
10	9	여자	163	50	108	65
11	10	여자	170	57	99	67

◀ ▶

Vital sheet

Lab data

+

⋮

◀

← **Select** person_id, sbp, dbp

➡ **Select** person_id, sbp, dbp
From Vital sheet ;

↖ **From** Vital sheet

SQL 문법_SELECT, FROM

❖ SELECT + FROM

- person_id, sbp, dbp만 조회할 때

1	person_id	sex	height	weight	sbp	dbp
2	1	남자	180	70	98	65
3	2	여자	162	50	100	69
4	3	여자	158	45	110	70
5	4	남자	172	58	101	71
6	5	여자	166	48	109	63
7	6	남자	176	60	98	64
8	7	남자	174	68	97	59
9	8	남자	178	70	103	60
10	9	여자	163	50	108	65
11	10	여자	170	57	99	67

< Vital sheet Lab data (+) >

→
Select person_id, sbp, dbp
From Vital sheet ;

1	person_id	sbp	dbp
2	1	98	65
3	2	100	69
4	3	110	70
5	4	101	71
6	5	109	63
7	6	98	64
8	7	97	59
9	8	103	60
10	9	108	65
11	10	99	67

< Vital sheet Lab data >

SQL 문법_WHERE

❖ SELECT + FROM + WHERE

1	person_id	sex	height	weight	sbp	dbp
2	1	남자	180	70	98	65
3	2	여자	162	50	100	69
4	3	여자	158	45	110	70
5	4	남자	172	58	101	71
6	5	여자	166	48	109	63
7	6	남자	176	60	98	64
8	7	남자	174	68	97	59
9	8	남자	178	70	103	60
10	9	여자	163	50	108	65
11	10	여자	170	57	99	67

Navigation: Vital sheet | Lab data | (+) | < |

← Select 컬럼 이름

← Where 조건

↑ From 테이블 이름

SQL기본 문법_WHERE

❖ SELECT + FROM + WHERE

- person_id, sex, sbp, dbp만 조회할 때 + 여자만

1	person_id	sex	height	weight	sbp	dbp
2	1	남자	180	70	98	65
3	2	여자	162	50	100	69
4	3	여자	158	45	110	70
5	4	남자	172	58	101	71
6	5	여자	168	48	105	65
7	6	남자	176	60	98	64
8	7	남자	174	68	97	59
9	8	남자	178	70	103	60
10	9	여자	163	50	108	65
11	10	여자	170	57	99	67

Vital sheet

Lab data

+

⋮

⏪

← **Select** person_id, sex ,sbp, dbp

← **Where** sex='여자'

➡ **Select** person_id, sex, sbp, dbp
From Vital sheet
Where sex='여자';

↑ **From** Vital sheet

SQL기본 문법_WHERE

❖ SELECT + FROM + WHERE

- person_id, sex, sbp, dbp만 조회할 때 + 여자만

1	person_id	sex	height	weight	sbp	dbp
2	1	남자	180	70	98	65
3	2	여자	162	50	100	69
4	3	여자	158	45	110	70
5	4	남자	172	58	101	71
6	5	여자	166	48	109	63
7	6	남자	176	60	98	64
8	7	남자	174	68	97	59
9	8	남자	178	70	103	60
10	9	여자	163	50	108	65
11	10	여자	170	57	99	67

→
Select person_id, sex, sbp, dbp
From Vital sheet
Where sex='여자';

1	person_id	sex	sbp	dbp
3	2	여자	100	69
4	3	여자	110	70
6	5	여자	109	63
10	9	여자	108	65
11	10	여자	99	67

SQL기본 문법_WHERE

❖ SELECT + FROM + WHERE

- person_id, sex, sbp, dbp만 조회할 때 + sbp가 100이상인 사람만

1	person_id	sex	height	weight	sbp	dbp
2	1	남자	180	70	98	65
3	2	여자	162	50	100	69
4	3	여자	158	45	110	70
5	4	남자	172	58	101	71
6	5	여자	166	48	109	69
7	6	남자	176	60	98	64
8	7	남자	174	68	97	59
9	8	남자	178	70	103	60
10	9	여자	163	50	108	65
11	10	여자	170	57	99	67

Vital sheet

Lab data

+

⋮

◀

← **Select** person_id, sex, sbp, dbp

← **Where** sbp >= 100

➔ **Select** person_id, sex, sbp, dbp
From Vital sheet
Where sbp >= 100;

↑ **From** Vital sheet

SQL기본 문법_WHERE

- **SELECT + FROM + WHERE**

- person_id, sex, sbp, dbp만 조회할 때 + sbp가 100이상인 사람만

1	person_id	sex	height	weight	sbp	dbp
2	1	남자	180	70	98	65
3	2	여자	162	50	100	69
4	3	여자	158	45	110	70
5	4	남자	172	58	101	71
6	5	여자	166	48	109	63
7	6	남자	176	60	98	64
8	7	남자	174	68	97	59
9	8	남자	178	70	103	60
10	9	여자	163	50	108	65
11	10	여자	170	57	99	67

→

Select person_id, sex, sbp, dbp
From Vital sheet
Where sbp >= 100;

1	person_id	sex	sbp	dbp
3	2	여자	100	69
4	3	여자	110	70
5	4	남자	101	71
6	5	여자	109	63
9	8	남자	103	60
10	9	여자	108	65

SQL 문법_ 집계 함수

- COUNT(* or 컬럼 이름) : 특정 열의 행의 **개수**를 세는 함수

예시)

	person_id	height
1	101	180
2	102	
3	103	162

Count(*) => 3

Count(height) => 2

- AVG(컬럼 이름) : 특정 열의 **평균**을 계산해주는 함수
- MIN (컬럼 이름) : 특정 열의 **최솟값**을 반환해주는 함수
- MAX(컬럼 이름) : 특정 열의 **최대값**을 반환해주는 함수
- SUM(컬럼 이름) 특정 열의 **합계**를 계산해주는 함수

SQL 문법_GROUP BY

❖ SELECT + FROM + GROUP BY

- 남자, 여자의 각각 인원수 파악

1	person_id	sex	height	weight	sbp	dbp
2	1	남자	180	70	98	65
3	2	여자	162	50	100	69
4	3	여자	159	45	110	70
5	4	남자	172	58	101	71
6	5	여자	166	45	109	63
7	6	남자	176	60	98	64
8	7	남자	174	68	97	59
9	8	남자	178	70	103	60
10	9	여자	163	50	108	65
11	10	여자	170	57	99	67

◀ ▶

Vital sheet

Lab data

+

sex	count(*)
남자	5
여자	5

SQL 문법_GROUP BY

❖ SELECT + FROM + GROUP BY

- 남자, 여자의 각각 인원수 파악

1	person_id	sex	height	weight	sbp	dbp
2	1	남자	180	70	98	65
3	2	여자	162	50	100	69
4	3	여자	159	45	110	70
5	4	남자	172	58	101	71
6	5	여자	166	45	109	63
7	6	남자	176	60	98	64
8	7	남자	174	68	97	59
9	8	남자	178	70	103	60
10	9	여자	163	50	108	65
11	10	여자	170	57	99	67

◀ ▶

Vital sheet

Lab data

⊕



sex	count(*)
남자	5
여자	5



Select sex, count(*)
From Vital sheet
Group by sex;

SQL 문법_GROUP BY

❖ SELECT + FROM + GROUP BY

- 남자, 여자의 각각 평균 체중 구하기

1	person_id	sex	height	weight	sbp	dbp
2	1	남자	180	70	98	65
3	2	여자	162	50	100	69
4	3	여자	159	45	110	70
5	4	남자	172	58	101	71
6	5	여자	166	45	109	63
7	6	남자	176	60	98	64
8	7	남자	174	68	97	59
9	8	남자	178	70	103	60
10	9	여자	163	50	108	65
11	10	여자	170	57	99	67

◀ ▶

Vital sheet

Lab data

⊕

sex	avg(weight)
남자	65.2
여자	49.4



Select sex, avg(weight)
From Vital sheet
Group by sex;

SQL 문법_GROUP BY

❖ SELECT + FROM + GROUP BY + HAVING

- 남자, 여자의 각각 평균 체중 구하기 + **남자만 조회**

1	person_id	sex	height	weight	sbp	dbp
2	1	남자	180	70	98	65
3	2	여자	162	50	100	69
4	3	여자	159	45	110	70
5	4	남자	172	58	101	71
6	5	여자	166	45	109	63
7	6	남자	176	60	98	64
8	7	남자	174	68	97	59
9	8	남자	178	70	103	60
10	9	여자	163	50	108	65
11	10	여자	170	57	99	67

◀ ▶

Vital sheet

Lab data

⊕

sex	avg(weight)
남자	65.2
여자	49.4

sex	avg(weight)
남자	65.2



Select sex, avg(weight)
From Vital sheet
Group by sex
Having sex='남자';

SQL 문법_ORDER BY

❖SELECT + FROM + ORDER BY

- sbp기준으로 오름차순

1	person_id	sex	height	weight	sbp	dbp
2	1	남자	180	70	98	65
3	2	여자	162	50	100	69
4	3	여자	159	45	110	70
5	4	남자	172	58	101	71
6	5	여자	166	45	109	63
7	6	남자	176	60	98	64
8	7	남자	174	68	97	59
9	8	남자	178	70	103	60
10	9	여자	163	50	108	65
11	10	여자	170	57	99	67

◀ ▶

Vital sheet

Lab data

+

ORDER BY 정렬 할 컬럼

asc - 오름차순(기본 설정)

desc - 내림차순

SQL 문법_ORDER BY

❖SELECT + FROM + ORDER BY

- sbp기준으로 오름차순

1	person_id	sex	height	weight	sbp	dbp
2	7	남자	174	68	97	59
3	1	남자	180	70	98	65
4	6	남자	176	60	98	64
5	10	여자	170	57	99	67
6	2	여자	162	50	100	69
7	4	남자	172	58	101	71
8	8	남자	178	70	103	60
9	9	여자	163	50	108	65
10	5	여자	166	45	109	63
11	3	여자	159	45	110	70

◀ ▶

Vital sheet

Lab data

⊕

ORDER BY 정렬 할 컬럼

asc - 오름차순(기본 설정)

desc - 내림차순



Select *

From Vital sheet

ORDER BY sbp (asc);

SQL 문법_ORDER BY

❖SELECT + FROM + ORDER BY

- sbp기준으로 내림차순

1	person_id	sex	height	weight	sbp	dbp
2	3	여자	159	45	110	70
3	5	여자	166	45	109	63
4	9	여자	163	50	108	65
5	8	남자	178	70	103	60
6	4	남자	172	58	101	71
7	2	여자	162	50	100	69
8	10	여자	170	57	99	67
9	1	남자	180	70	98	65
10	6	남자	176	60	98	64
11	7	남자	174	68	97	59

◀ ▶

Vital sheet

Lab data

⊕

ORDER BY 정렬 할 컬럼

asc - 오름차순(기본 설정)

desc - 내림차순



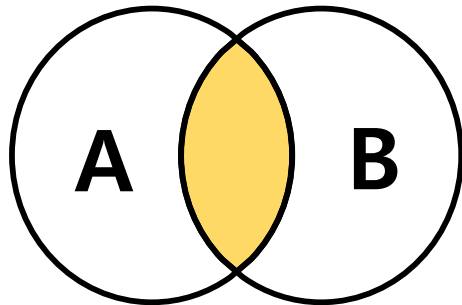
Select *

From Vital sheet

ORDER BY sbp desc;

SQL 문법_JOIN

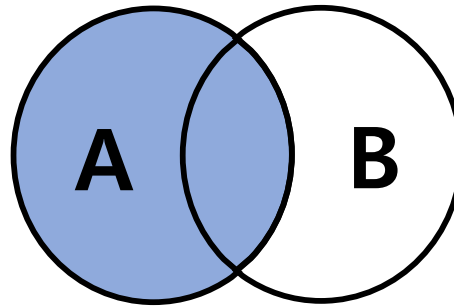
Inner Join



```
SELECT *  
FROM A  
INNER JOIN B  
ON A.컬럼이름 = B.컬럼이름;
```

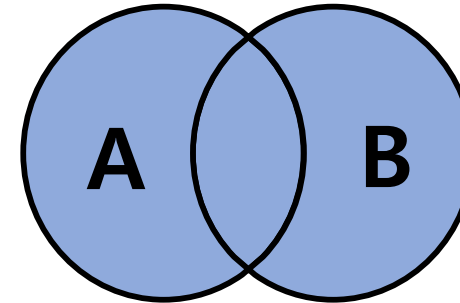
Outer Join

LEFT OUTER JOIN



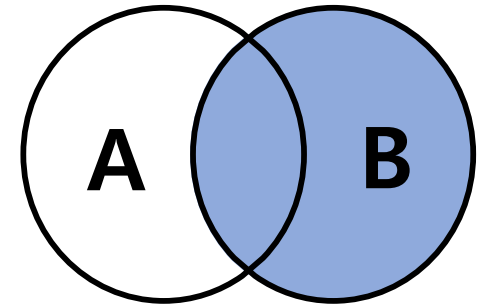
```
SELECT *  
FROM A  
LEFT OUTER JOIN B  
ON A.컬럼이름 = B.컬럼이름;
```

FULL OUTER JOIN



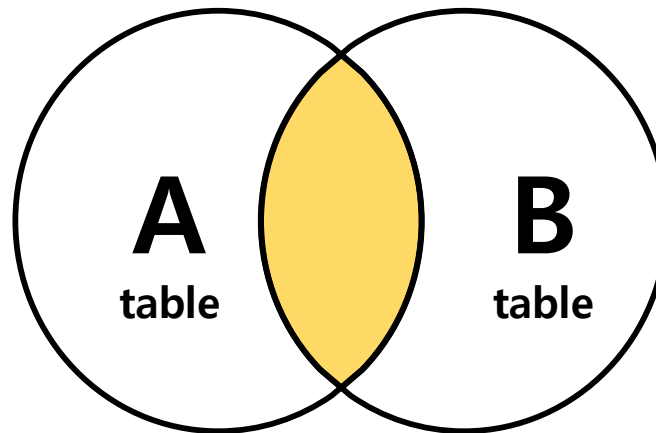
```
SELECT *  
FROM A  
FULL OUTER JOIN B  
ON A.컬럼이름 = B.컬럼이름;
```

RIGHT OUTER JOIN



```
SELECT *  
FROM A  
RIGHT OUTER JOIN B  
ON A.컬럼이름 = B.컬럼이름;
```

Inner Join



<A테이블과 B테이블의 INNER JOIN>

```
SELECT *  
FROM A table  
INNER JOIN B table  
ON A.컬럼이름 = B.컬럼이름;
```

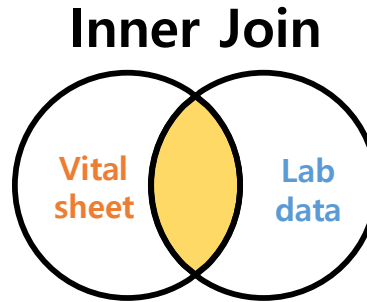

SQL 문법 INNER JOIN

❖ SELECT + FROM + INNER JOIN + ON

- Vital sheet + Lab data 테이블 inner join으로 합쳐보기

<Vital sheet>

1	person_id	sex	height	weight	sbp	dbp
2	1	남자	180	70	98	65
3	2	여자	162	50	100	69
4	3	여자	159	45	110	70
5	4	남자	172	58	101	71
6	5	여자	166	45	109	63
7	6	남자	176	60	98	64
8	7	남자	174	68	97	59
9	8	남자	178	70	103	60
10	9	여자	163	50	108	65
11	10	여자	170	57	99	67



<Lab data>

1	person_id	rbc	wbc	hb
2	3	5	6	13.9
3	5	5.8	7.9	14.2
4	6	5.2	7.3	15.9
5	10	5.4	8.2	14.6
6	15	4.8	8.6	15.3

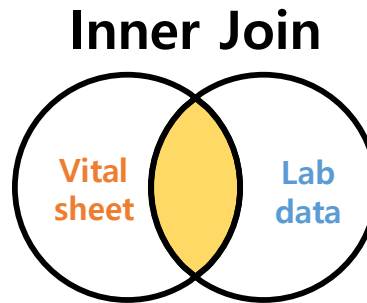
SQL 문법 INNER JOIN

❖ SELECT + FROM + INNER JOIN + ON

- person_id로 Vital sheet + Lab data 테이블 inner join으로 합쳐보기

<Vital sheet>

	person_id	sex	height	weight	sbp	dbp
1						
2	1	남자	180	70	98	65
3	2	여자	162	50	100	69
4	3	여자	159	45	110	70
5	4	남자	172	58	101	71
6	5	여자	166	45	109	63
7	6	남자	176	60	98	64
8	7	남자	174	68	97	59
9	8	남자	178	70	103	60
10	9	여자	163	50	108	65
11	10	여자	170	57	99	67



<Lab data>

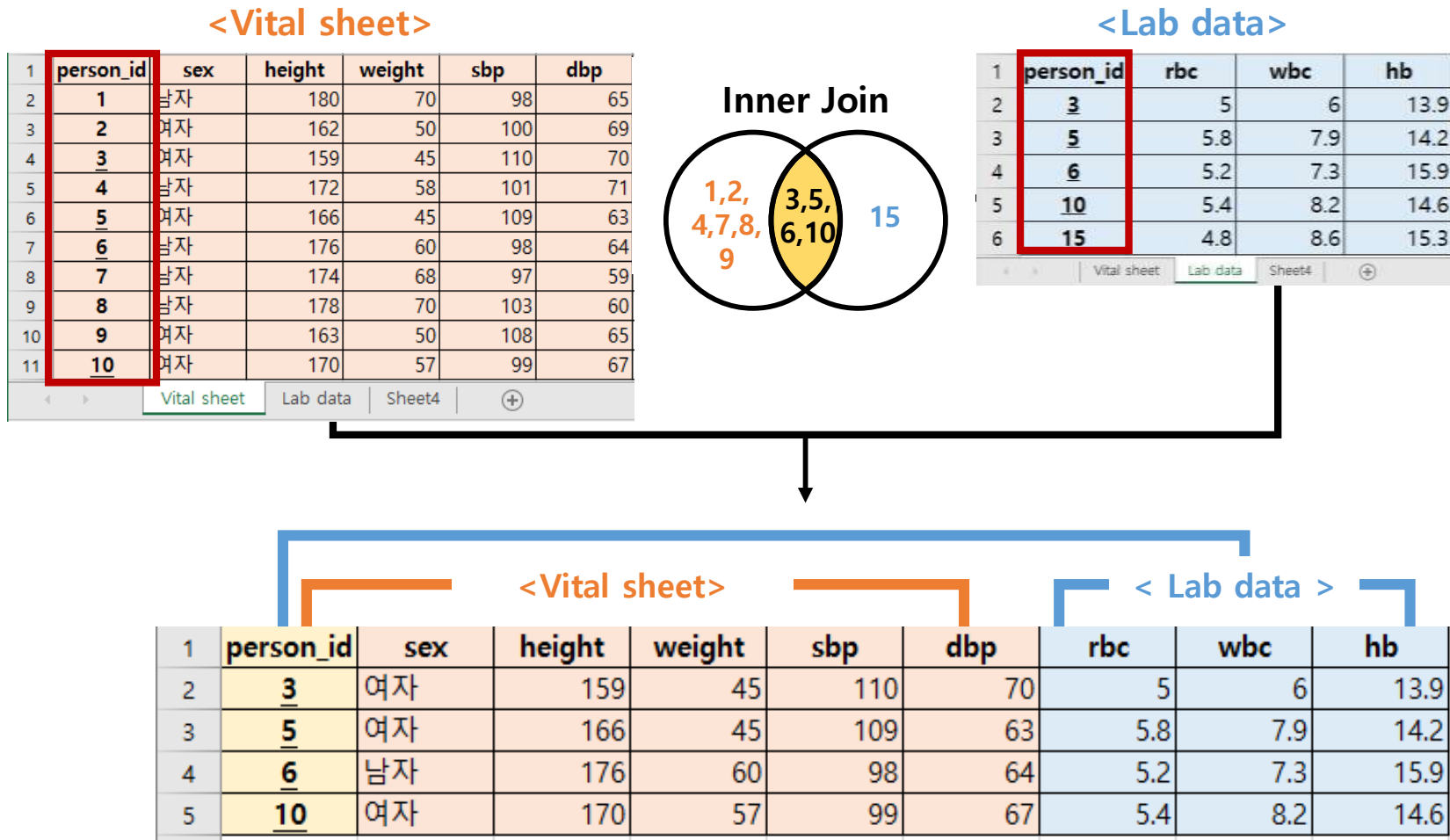
	person_id	rbc	wbc	hb
1				
2	3	5	6	13.9
3	5	5.8	7.9	14.2
4	6	5.2	7.3	15.9
5	10	5.4	8.2	14.6
6	15	4.8	8.6	15.3

➡ **Select** *
From Vital sheet
INNER JOIN Lab data
ON Vital sheet.person_id = Lab data .person_id;

SQL 문법 INNER JOIN

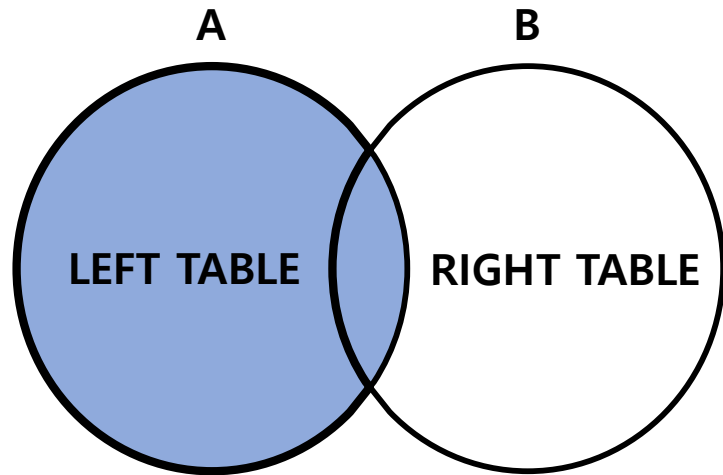
❖ SELECT + FROM + INNER JOIN + ON

- person_id로 Vital sheet + Lab data 테이블 inner join으로 합쳐보기

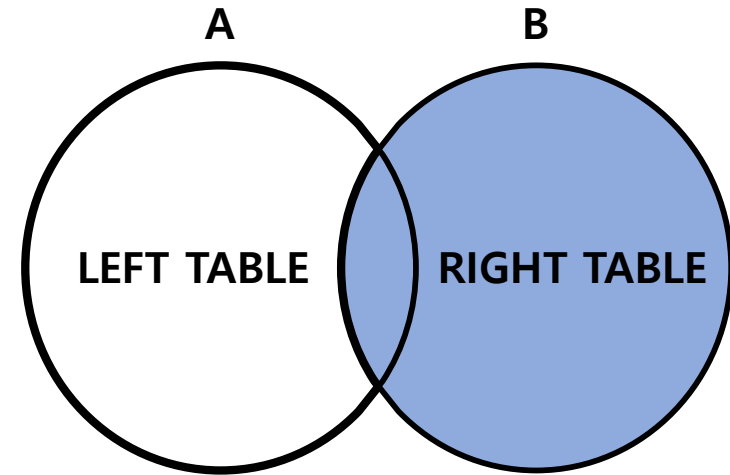


SQL 문법_OUTER JOIN

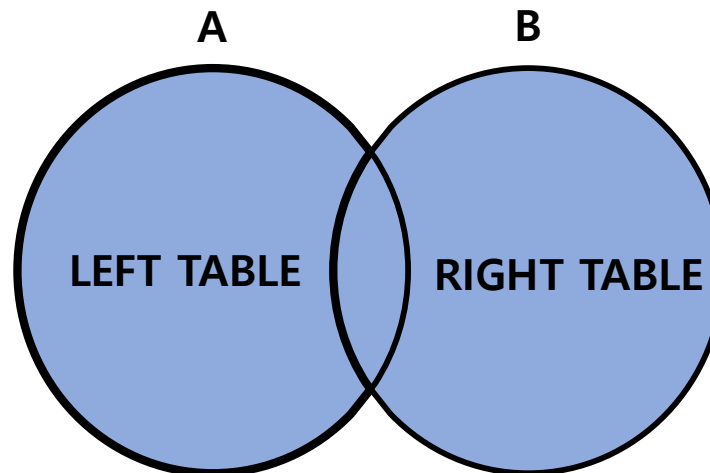
LEFT OUTER JOIN



RIGHT OUTER JOIN

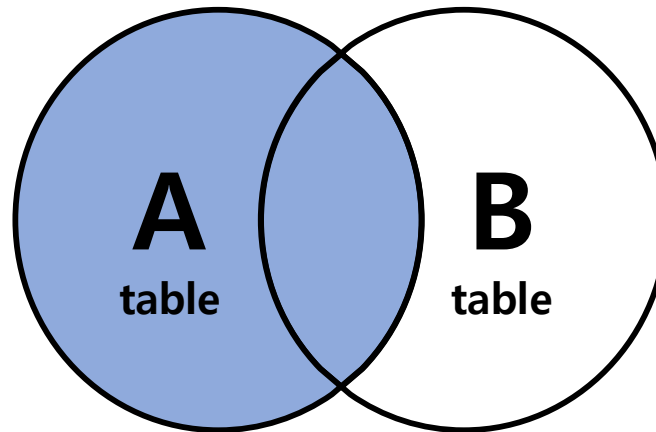


FULL OUTER JOIN



데이터 결합하기 (Outer Join)

Outer Join



<A테이블과 B테이블의 LEFT OUTER JOIN>

```
SELECT *  
FROM A  
LEFT (OUTER) JOIN B  
ON A.컬럼이름 = B.컬럼이름;
```


SQL 문법_OUTER JOIN

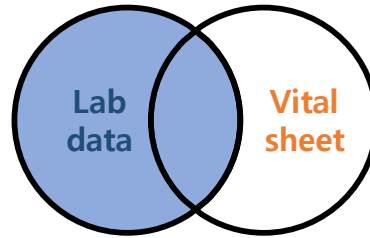
❖SELECT + FROM + LEFT OUTER JOIN

- person_id로 Vital sheet + Lab data 테이블 left outer join으로 합쳐보기

<Lab data>

1	person_id	rbc	wbc	hb
2	3	5	6	13.9
3	5	5.8	7.9	14.2
4	6	5.2	7.3	15.9
5	10	5.4	8.2	14.6
6	15	4.8	8.6	15.3

Left (outer) Join



<Vital sheet>

1	person_id	sex	height	weight	sbp	dbp
2	1	남자	180	70	98	65
3	2	여자	162	50	100	69
4	3	여자	159	45	110	70
5	4	남자	172	58	101	71
6	5	여자	166	45	109	63
7	6	남자	176	60	98	64
8	7	남자	174	68	97	59
9	8	남자	178	70	103	60
10	9	여자	163	50	108	65
11	10	여자	170	57	99	67



Select *

From Lab data

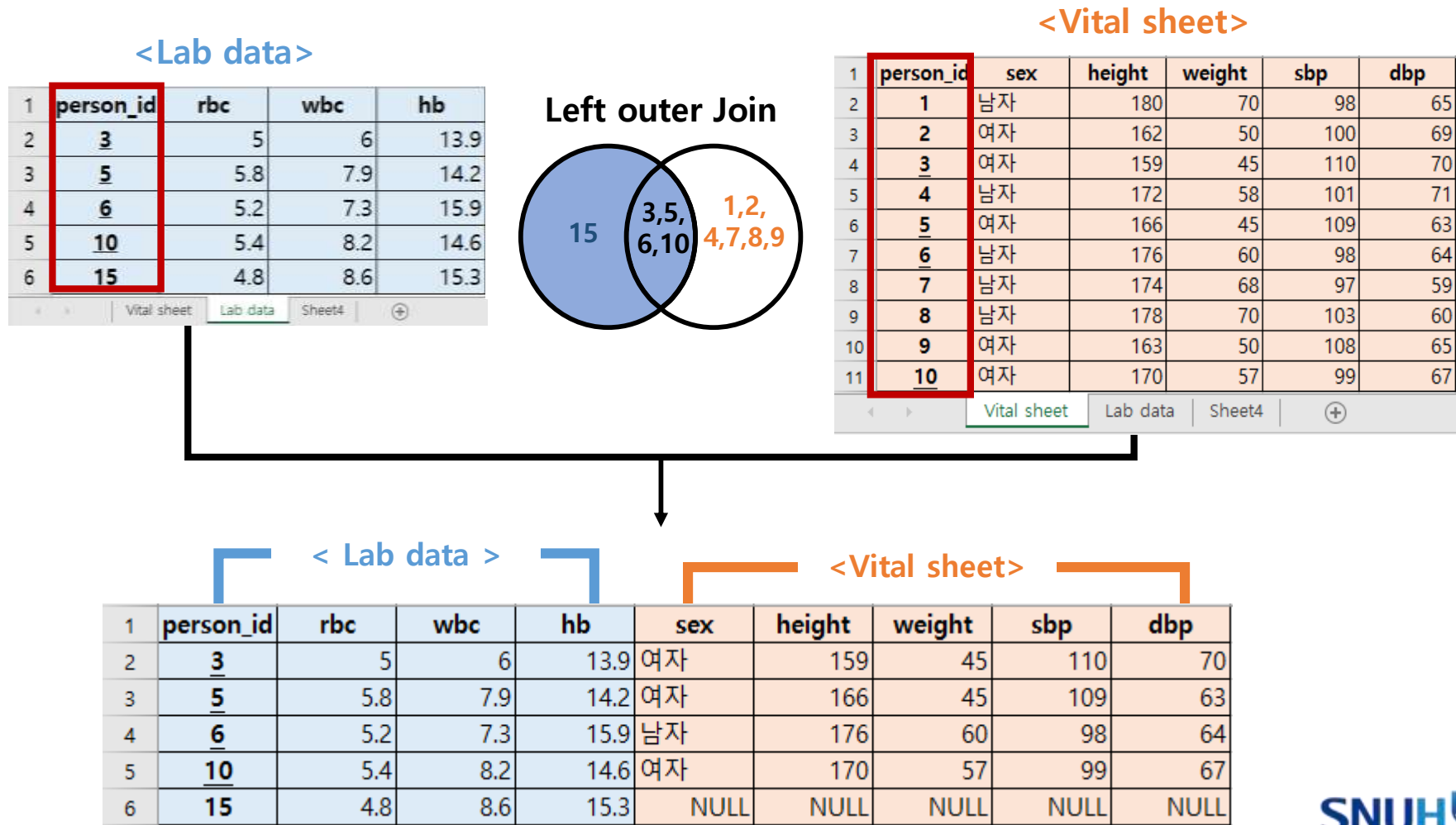
LEFT (OUTER) JOIN Vital sheet

ON Lab data .person_id = Vital sheet.person_id;

SQL 문법_OUTER JOIN

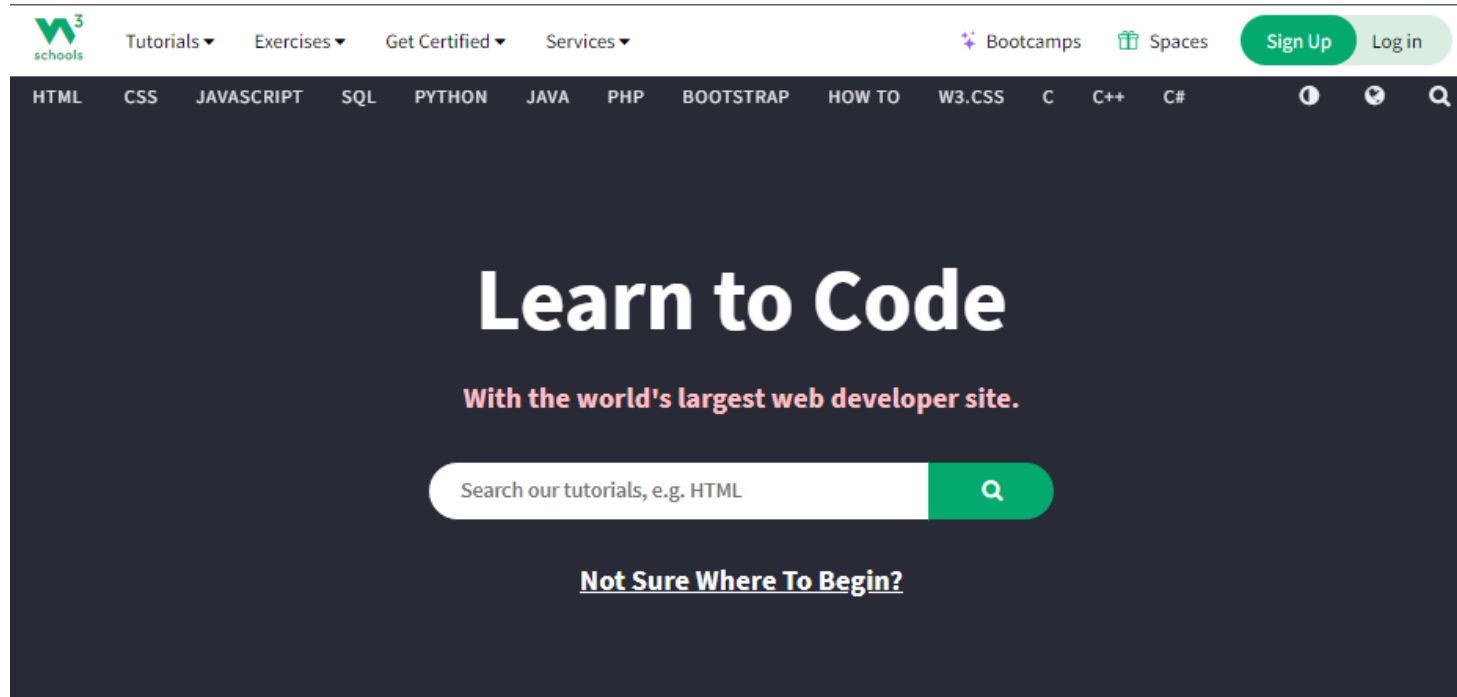
❖ SELECT + FROM + LEFT OUTER JOIN

- person_id로 Vital sheet + Lab data 테이블 left outer join으로 합쳐보기



실습 환경

- W3 schools



https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all

SQL 조회 기본 구조

SELECT [column]

FROM [table_name]

WHERE [expr]

GROUP BY [column]

HAVING [column]

ORDER BY [column]

SELECT : 테이블 내의 **조회할 컬럼명**들을 입력

FROM : 조회할 **테이블명**들을 입력

WHERE : 데이터 중 **조건**에 맞는 일부의 데이터만을 조회하기 위한 조건 입력

GROUP BY : 데이터들을 작은 **그룹으로 분류**하여 소그룹에 대한 항목별로 통계 정보를 얻을 때 사용됩니다.

HAVING : WHERE 절과 비슷하지만 그룹을 나타내는 결과 집합의 행에 **조건**이 적용되는 차이가 있습니다.

ORDER BY : 조회할 데이터들을 어떤 기준으로 **정렬**할 것인지 정렬 방법 입력

테이블 설명

<Customers table>

컬럼 이름	예시
CustomerID	1
CustomerName	Alfreds Futterkiste
ContactName	Maria Anders
Address	Obere Str. 57
City	Berlin
PostalCode	12209
Country	Germany

<Order table>

컬럼 이름	예시
OrderID	1
CustomerID	Alfreds Futterkiste
EmployeeID	Maria Anders
OrderDate	Obere Str. 57
ShipperID	Berlin

<OrderDetails table>

컬럼 이름	예시
OrderDetailID	1
OrderID	Alfreds Futterkiste
ProductID	Maria Anders
Quantity	Obere Str. 57

<Categories table>

컬럼 이름	예시
CategoryID	1
CategoryName	Alfreds Futterkiste
Description	Maria Anders

SQL을 활용한 데이터 조회

SELECT <컬럼 목록>

FROM <테이블 명>;

+ 별표 기호(*)는 전체 컬럼의 모든 내용을 다 보여주는 명령어

<Customers table>

컬럼 이름	예시
CustomerID	1
CustomerName	Alfreds Futterkiste
ContactName	Maria Anders
Address	Obere Str. 57
City	Berlin
PostalCode	12209
Country	Germany

- Customers 테이블에서
CustomerID, CustomerName, Country 컬럼 추출.

```
SELECT CustomerID, CustomerName, Country  
FROM Customers;
```

Number of Records: 91

CustomerID	CustomerName	Country
1	Alfreds Futterkiste	Germany
2	Ana Trujillo Emparedados y helados	Mexico
3	Antonio Moreno Taquería	Mexico
4	Around the Horn	UK
5	Berglunds snabbköp	Sweden
6	Blauer See Delikatessen	Germany

SQL을 활용한 데이터 조회

SELECT <컬럼 목록>

FROM <테이블 명>

WHERE <조건문>;

- OrderDetails 테이블에서
Quantity가 40이상인 모든 컬럼 추출.

```
SELECT *  
FROM Customers  
WHERE Quantity >= 40;
```

Number of Records: 109

OrderDetailID	OrderID	ProductID	Quantity
5	10249	51	40
12	10252	20	40
14	10252	60	40
16	10253	39	42
17	10253	49	40
30	10258	2	50

SQL을 활용한 데이터 조회_논리연산자

- AND : 둘다 True이면 True, 하나라도 False이면 False. WHERE 절에 여러 개의 조건이 존재.

예) where age >= 70 and sbp >=140 : 나이가 70이상 **이면서** sbp가 140이상인 경우

- OR : 하나가 True이면 True. WHERE 절에서 여러 조건을 합치기 위해 사용.

예) where age >= 70 or sbp >=140 : 나이가 70이상 **이거나** sbp가 140이상인 경우

- LIKE : 해당 값과 유사한 값을 찾으려 할 때 사용 ('%' : 모든 글자 / '_' : 한 글자)

예) 대장암으로 진단받은 환자를 찾으시오

where 진단명컬럼 like '%C18%';

- IN (값1, 값2, ...) : WHERE절 내에서 선택하고자 하는 값만 고르는 연산자

예) 환자 ID = 10208, 10209 > where 컬럼명 in (10208, 10209);

SQL을 활용한 데이터 조회_논리연산자

- AND

```
SELECT *  
FROM Customers  
WHERE Country='UK' AND City='London';
```

Number of Records: 6

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
11	B's Beverages	Victoria Ashworth	Fauntleroy Circus	London	EC2 5NT	UK
16	Consolidated Holdings	Elizabeth Brown	Berkeley Gardens 12 Brewery	London	WX1 6LT	UK
19	Eastern Connection	Ann Devon	35 King George	London	WX3 6FW	UK
53	North/South	Simon Crowther	South House 300 Queensbridge	London	SW7 1RZ	UK
72	Seven Seas Imports	Hari Kumar	90 Wadhurst Rd.	London	OX15 4NB	UK

- OR

```
SELECT *  
FROM Customers  
WHERE Country='UK' OR City='London';
```

Number of Records: 7

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
11	B's Beverages	Victoria Ashworth	Fauntleroy Circus	London	EC2 5NT	UK
16	Consolidated Holdings	Elizabeth Brown	Berkeley Gardens 12 Brewery	London	WX1 6LT	UK
19	Eastern Connection	Ann Devon	35 King George	London	WX3 6FW	UK
38	Island Trading	Helen Bennett	Garden House Crowther Way	Cowes	PO31 7PJ	UK
53	North/South	Simon Crowther	South House 300 Queensbridge	London	SW7 1RZ	UK
72	Seven Seas Imports	Hari Kumar	90 Wadhurst Rd.	London	OX15 4NB	UK

SQL을 활용한 데이터 조회_논리연산자

- LIKE

1. %c% : c가 포함되어 있는 모든 값

2. c% : c로 시작하는 모든 값

3. %c : c로 끝나는 모든 값

4. %c_ : 앞에 다른 글자들이 나온 후
c다음 한글자만 존재 하는 모든 값

Number of Records: 8

CategoryID	CategoryName
1	Beverages
2	Condiments
3	Confections
4	Dairy Products
5	Grains/Cereals
6	Meat/Poultry
7	Produce
8	Seafood

%C%	C%	%C	%C_
CategoryName	CategoryName	No result.	CategoryName
Condiments	Condiments		Produce
Confections	Confections		
Dairy Products			
Grains/Cereals			
Produce			

SQL을 활용한 데이터 조회_논리연산자

SELECT <컬럼 목록>

FROM <테이블 명>

WHERE 컬럼명 **IN** (값1, 값2, ...);

- Customers 테이블에서 CustomerID가 4,11,16,19,500인 사람 추출.

```
SELECT *  
FROM Customers  
WHERE CustomerID IN (4,11,16,19,500);
```

Number of Records: 4

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
11	B's Beverages	Victoria Ashworth	Fauntleroy Circus	London	EC2 5NT	UK
16	Consolidated Holdings	Elizabeth Brown	Berkeley Gardens 12 Brewery	London	WX1 6LT	UK
19	Eastern Connection	Ann Devon	35 King George	London	WX3 6FW	UK

조건에 맞는 데이터 그룹핑 및 정렬

SELECT <컬럼 목록>

FROM <테이블 명>

GROUP BY <그룹화 할 컬럼명>;

- OrderDetails테이블에서 OrderID별 총 주문량 구해보기

```
SELECT OrderID, ProductID, SUM(Quantity) AS SUM  
FROM OrderDetails  
GROUP BY OrderID;
```

Number of Records: 196

OrderDetailID	OrderID	ProductID	Quantity
1	10248	11	12
2	10248	42	10
3	10248	72	5

OrderID	ProductID	SUM
10248	11	27
10249	14	49
10250	41	60
10251	22	41
10252	20	105

조건에 맞는 데이터 그룹핑 및 정렬

SELECT <컬럼 목록>

FROM <테이블 명>

WHERE <조건식>

GROUP BY <그룹화 할 컬럼명>;

- ProductID가 30초과인 OrderID별 총 주문량 구해보기

```
SELECT OrderID, ProductID, SUM(Quantity) AS SUM
FROM OrderDetails
WHERE ProductID > 30
GROUP BY OrderID;
```

Number of Records: 177

WHERE ProductID > 30

<OrderDetails Table>

OrderDetailID	OrderID	ProductID	Quantity
1	10248	11	12
2	10248	42	10
3	10248	72	5

OrderID	ProductID	SUM
10248	42	15
10249	51	40
10250	41	60
10251	57	35
10252	33	65

조건에 맞는 데이터 그룹핑 및 정렬

SELECT <컬럼 목록>

FROM <테이블 명>

WHERE <조건식>

GROUP BY <그룹화 할 컬럼명>

HAVING <조건식>;

Having SUM >= 40;

Number of Records: 177

OrderID	ProductID	SUM
10248	42	15
10249	51	<u>40</u>
10250	41	<u>60</u>
10251	57	35
10252	33	<u>65</u>

- ProductID가 30초과인 OrderID별 총 주문량에서 40이상

```
SELECT OrderID, ProductID, SUM(Quantity) AS SUM
FROM OrderDetails
WHERE ProductID > 30
GROUP BY OrderID
HAVING SUM >= 40;
```

Number of Records: 85

OrderID	ProductID	SUM
10249	51	40
10250	41	60
10252	33	65
10253	31	102

조건에 맞는 데이터 그룹핑 및 정렬

SELECT <컬럼 목록>

FROM <테이블 명>

WHERE <조건식>

ORDER BY <컬럼 OR 컬럼 순서>;

- desc:내림차순
- (asc):오름차순(기본설정)

- ProductID가 30초과이며, Quantity기준으로 오름차순 정렬

```
SELECT *  
FROM OrderDetails  
WHERE ProductID > 30  
ORDER BY Quantity;
```

→ ORDER BY 4; 와 동일

Number of Records: 342

1	2	3	4
OrderDetailID	OrderID	ProductID	Quantity
34	10259	37	1
162	10308	69	1
43	10262	56	2
91	10282	57	2

조건에 맞는 데이터 그룹핑 및 정렬

SELECT <컬럼 목록>

FROM <테이블 명>

WHERE <조건식>

ORDER BY <컬럼 OR 컬럼 순서>;

- desc:내림차순
- (asc):오름차순(기본설정)

- ProductID가 30초과이며, Quantity기준으로 내림차순 정렬

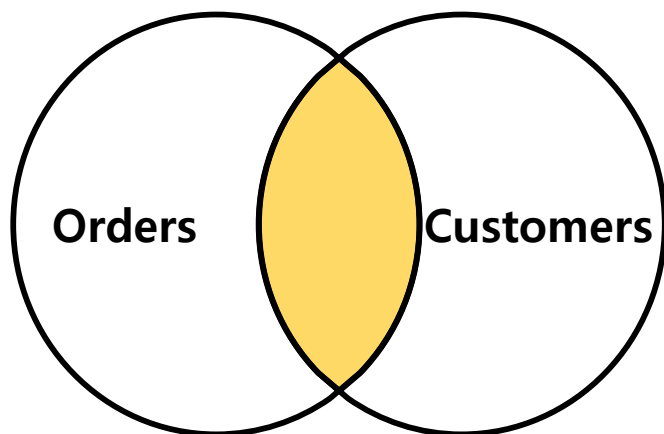
```
SELECT *  
FROM OrderDetails  
WHERE ProductID > 30  
ORDER BY Quantity desc;
```

Number of Records: 342

OrderDetailID	OrderID	ProductID	Quantity
401	10398	55	120
103	10286	35	100
512	10440	61	90
201	10324	63	80

데이터 결합하기 (Inner Join)

Inner Join



< Customers table과 Order table의 INNER JOIN >

```
SELECT *  
FROM Orders  
INNER JOIN Customers  
ON Orders.CustomerID = Customers.CustomerID;
```

<Orders table>

컬럼 이름	예시
OrderID	1
<u>CustomerID</u>	Alfreds Futterkiste
EmployeeID	Maria Anders
OrderDate	Obere Str. 57
ShipperID	Berlin

<Customers table>

컬럼 이름	예시
<u>CustomerID</u>	1
CustomerName	Alfreds Futterkiste
ContactName	Maria Anders
Address	Obere Str. 57
City	Berlin
PostalCode	12209
Country	Germany

데이터 결합하기 (Inner Join)

<Orders Table>

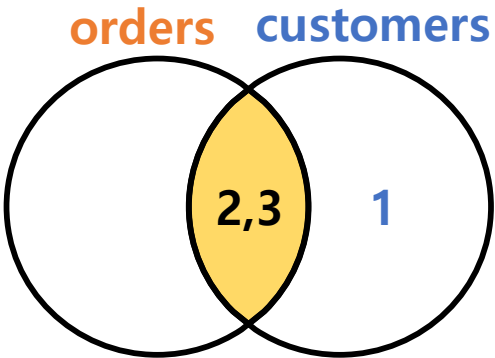
OrderID	Customer ID	EmployeeID	OrderDate
10308	2	7	1996-09-18
10365	3	3	1996-11-27

<Customers Table>

Customer ID	CustomerName	Address
1	Alfreds Futterkiste	Obere Str. 57
2	Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222
3	Antonio Moreno Taquería	Mataderos 2312

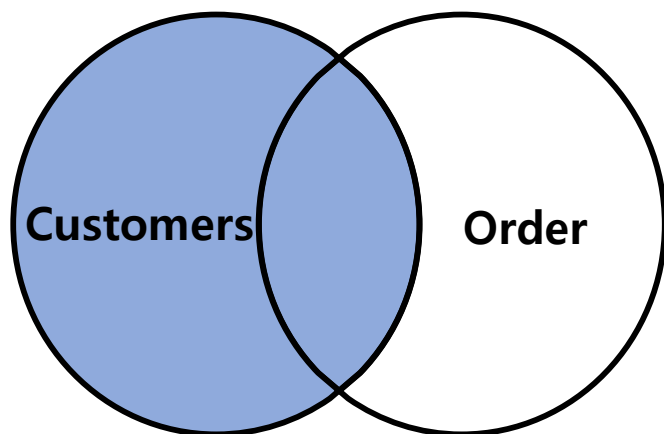
```
SELECT OrderID, Customers.CustomerID, EmployeeID, OrderDate, CustomerName, Address
FROM Orders
INNER JOIN Customers
ON Orders.CustomerID = Customers.CustomerID
ORDER BY Orders.CustomerID;
```

OrderID	CustomerID	EmployeeID	OrderDate	CustomerName	Address
10308	2	7	1996-09-18	Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222
10365	3	3	1996-11-27	Antonio Moreno Taquería	Mataderos 2312



데이터 결합하기 (Outer Join)

Outer Join



< Customers table과 Order table의 OUTER JOIN >

SELECT *

FROM Customers

LEFT (OUTER) JOIN Orders

ON Customers.CustomerID = Orders.CustomerID;

<Customers table>

컬럼 이름	예시
<u>CustomerID</u>	1
CustomerName	Alfreds Futterkiste
ContactName	Maria Anders
Address	Obere Str. 57
City	Berlin
PostalCode	12209
Country	Germany

<Orders table>

컬럼 이름	예시
OrderID	1
<u>CustomerID</u>	Alfreds Futterkiste
EmployeeID	Maria Anders
OrderDate	Obere Str. 57
ShipperID	Berlin

데이터 결합하기 (Outer Join)

<Customers Table>

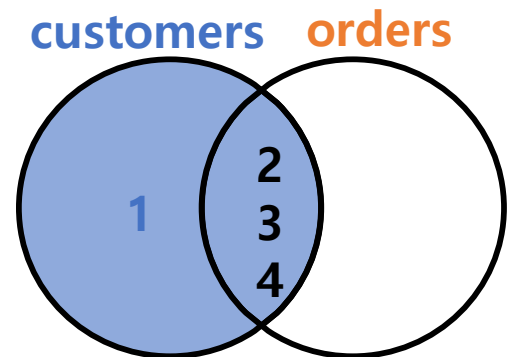
CustomerID	CustomerName	City
1	Alfreds Futterkiste	Berlin
2	Ana Trujillo Emparedados	México D.F
3	Antonio Moreno Taquería	México D.F
4	Around the Horn	London

<Orders Table>

OrderID	CustomerID	OrderDate
10308	2	1996-09-18
10365	3	1996-11-27
10355	4	1996-11-15
10383	4	1996-12-16

```
SELECT Customers.CustomerID, CustomerName, City, OrderID, OrderDate
FROM Customers
LEFT OUTER JOIN Orders
ON Customers.CustomerID = Orders.CustomerID;
```

CustomerID	CustomerName	City	OrderID	OrderDate
1	Alfreds Futterkiste	Berlin	<i>null</i>	<i>null</i>
2	Ana Trujillo Emparedados	México D.F	10308	1996-09-18
3	Antonio Moreno Taquería	México D.F	10365	1996-11-27
4	Around the Horn	London	10355	1996-11-15
4	Around the Horn	London	10383	1996-12-16



데이터 결합하기 (Outer Join)

<Customers Table>

CustomerID	CustomerName	City
1	Alfreds Futterkiste	Berlin
2	Ana Trujillo Emparedados	México D.F
3	Antonio Moreno Taquería	México D.F
4	Around the Horn	London

<Order Table>

OrderID	CustomerID	OrderDate
10308	2	1996-09-18
10365	3	1996-11-27
10355	4	1996-11-15
10383	4	1996-12-16

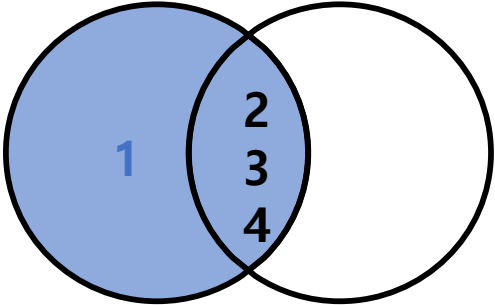
```
SELECT Customers.CustomerID, CustomerName, City, OrderID, OrderDate
FROM Customers
LEFT OUTER JOIN Orders
ON Customers.CustomerID = Orders.CustomerID
```

CustomerID	CustomerName	City	OrderID	OrderDate
1	Alfreds Futterkiste	Berlin	null	null
2	Ana Trujillo Emparedados	México D.F	10308	1996-09-18
3	Antonio Moreno Taquería	México D.F	10365	1996-11-27
4	Around the Horn	London	10355	1996-11-15
4	Around the Horn	London	10383	1996-12-16

```
SELECT * FROM 위 테이블
WHERE ORDERID=NULL;
```

CustomerID	CustomerName	City	OrderID	OrderDate
1	Alfreds Futterkiste	Berlin	null	null

customers orders



데이터 결합하기 (Outer Join)

- WITH문 작성 방법

WITH 테이블 별칭 AS (SUB QUERY)
MAIN QUERY ;

```
SELECT Customers.CustomerID, CustomerName, City, OrderID, OrderDate
FROM Customers
LEFT OUTER JOIN Orders
ON Orders.CustomerID = Customers.CustomerID;
```

CustomerID	CustomerName	City	OrderID	OrderDate
1	Alfreds Futterkiste	Berlin	null	null
2	Ana Trujillo Emparedados	México D.F	10308	1996-09-18
3	Antonio Moreno Taquería	México D.F	10365	1996-11-27
4	Around the Horn	London	10355	1996-11-15
4	Around the Horn	London	10383	1996-12-16

```
SELECT * FROM 위 테이블
WHERE ORDERID=NULL;
```

CustomerID	CustomerName	City	OrderID	OrderDate
1	Alfreds Futterkiste	Berlin	null	null

테이블 별칭

```
WITH NEW_TABLE AS
{
  (SELECT Customers.CustomerID, CustomerName, City,
    OrderID, OrderDate
   FROM Customers
  LEFT OUTER JOIN Orders
  ON Orders.CustomerID = Customers.CustomerID)
} SUB QUERY
{
  SELECT * FROM NEW_TABLE
  WHERE ORDERID=NULL ;
} MAIN QUERY
```

빅데이터 센터 원내 의료 데이터 소개

CDM

- 여러 기관에서 공통으로 사용하는 표준화된 공통 모델

MIMIC

- Beth Israel Deaconess Medical Center의 ICU 환자 데이터

NIA 암 빅데이터

- 유방암, 대장암, 위암, 폐암, 간암 총 5가지 환자정보

소방 구급데이터

- 경기도 내 소방구급환자 데이터 모델

E-Claim

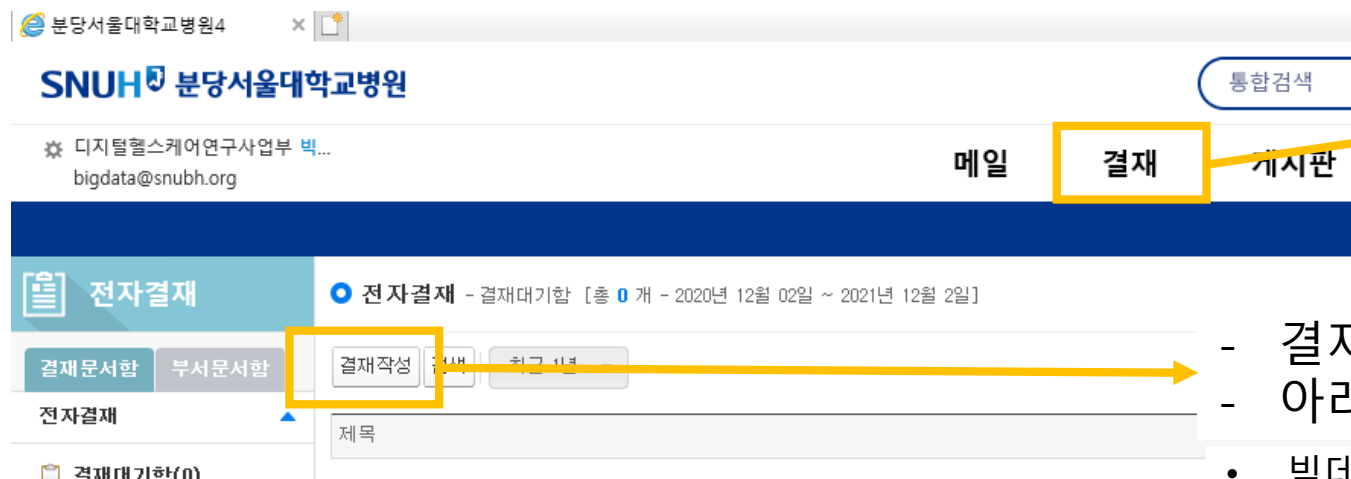
- 원내청구자료를 건강보험공단 표본 코호트 2.0 구조로 변환됨
- 원내 의료데이터

공공데이터

- 대기오염 연구자료
- 신청 시 추가 예정

빅데이터 DB사용 절차 안내

1. DB 이용 계정 신청 방법



- 그룹웨어 결재 클릭

- 결재작성 클릭
- 아래 신청하고자 하는 데이터 신청서 확인 후 작성
 - 빅데이터 플랫폼 사용자 계정 신청서
 - CDM연구를 위한 원격 접속 계정 신청서

빅데이터 DB사용 절차 안내

1. 빅데이터 플랫폼 계정 신청 방법

양식명	빅데이터 플랫폼 사용자 계정발급 신청서	
결재선	담당자	연구행정파트 '이용석' 원내번호: 4207
	신청	신청자 -> 신청부서 부서장
	수신	연구행정파트 '이용석'

계정 관련	<input type="checkbox"/> 계정 신청 <input type="checkbox"/> 계정 갱신 <input type="checkbox"/> 접속 정보 변경 (IP 최대 2까지 유지 가능) ※ CDM 계정 신규발급의 경우 기존 결재 양식 ' <u>CDM(Common Data Model) 연구를 위한 원격 접속 계정 신청서</u> '를 활용하여 주시기 바랍니다. ※ CDM 계정 갱신의 경우 본 양식을 통해 신청(계정 갱신 체크)
연산 서비스 및 데이터	아래 항목들에 대해서 연구 내용에 따라 선택 (연구 계획서 내용에 합당하면 중복 가능) <input type="checkbox"/> 딥러닝용 GPU 서버 사용 <input type="checkbox"/> E-ClaimDB 사용 <input type="checkbox"/> 공공 데이터 사용 <input type="checkbox"/> MIMIC 4 데이터셋 사용 ※ 책임 연구자가 PhysioNet의 Credentialed user여야 하며, 하단 링크에 나오는 "Signed Data Use Agreement (for MIMIC-IV)"를 필히 첨부하시기 바랍니다. https://physionet.org/settings/agreements <input type="checkbox"/> NIA 암 빅데이터셋 사용 <input type="checkbox"/> 유방암 <input type="checkbox"/> 대장암 <input type="checkbox"/> 위암 <input type="checkbox"/> 폐암 <input type="checkbox"/> 간암 (IRB 내용에 맞게 다중 선택 가능) <input type="checkbox"/> + CDM 확장 (NIA 데이터 환자의 CDM 의료정보 연계 시 추가 체크) <input type="checkbox"/> 소방 구급 데이터셋 사용 <input type="checkbox"/> + CDM 확장 (소방 구급 데이터 환자의 CDM 의료정보 연계 시 추가 체크)

-
- 계정 신청 또는 계정 갱신 선택
 - 연산 서비스 및 데이터 선택

빅데이터 DB사용 절차 안내

2. CDM연구를 위한 원격 접속 계정 신청서

CDM(Common Data Model) 연구를 위한 원격 접속 계정 신청서

작성부서	디지털헬스케어연구사업부	작성일	2023-06-16	작성자	빅데이터센터공용
문서번호	디지털헬스케어연구사업부-	보존기한		공개유무	

제 목

- 제목

※ "CDM 연구를 위한 원격 접속 계정 신청서" 작성

작성시 유의사항	<p>1) 원내(SNUBH)에서 받은 IRB 심의면제서, IRB 연구계획서를 반드시 첨부하셔야 합니다. 2) 사번이 있는 원내 연구자에 한해 계정 발급 가능합니다. (IRB 연구계획서에 신청자 포함 되어 있어야 함) 3) 아래 링크 주소에 첨부되어 있는 정보보호 서약서 양식을 다운받아 서명 후, 스캔하여 반드시 첨부해주시기 바랍니다. https://group.snubh.org/myoffice/ezBoardSTD/BoardItemView_Cross.aspx?ShowAdjacent=&ItemID={C7927EFE-E8F1-44F0-9332-370348AA7CE1}&BoardID={5f16f1d5-86a1-b515-ea5a-afb27aae9b37}&location=GENERAL</p>
비고	계정발급은 매월 첫째주와 셋째주 금요일 일괄 발급되며, 디지털헬스케어연구사업부 내부 결제 후 최종승인 일을 기준으로 합니다.

- 정보보호 서약서 작성

※ 반드시 첨부

빅데이터센터 클라우드 설문조사



빅데이터센터 설문조사

https://docs.google.com/forms/d/e/1FAIpQLSdmJqol_sNKgYYNhg4I-gF7kxKKVXbOfdnhy4_N1nqAn1R2yg/viewform?pli=1