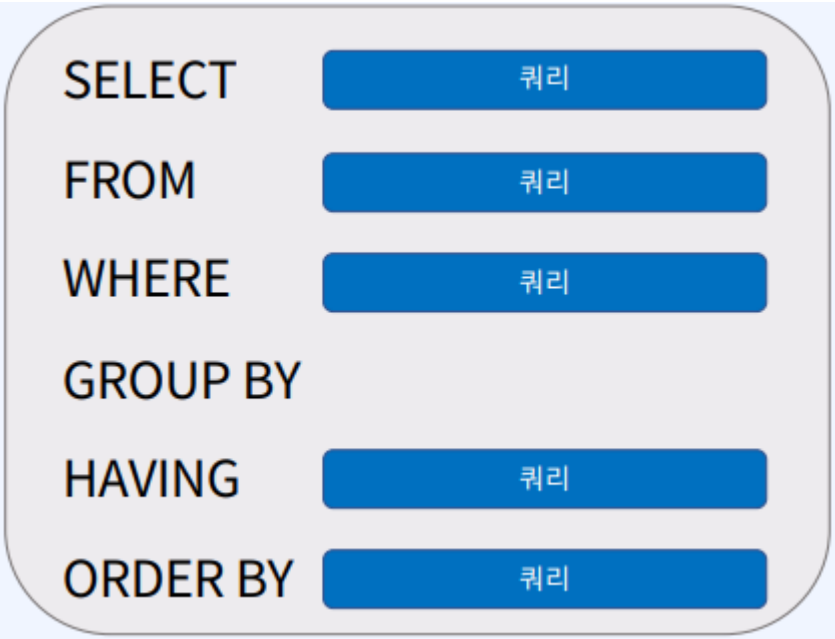
04. Sub Query

▼ Index

[Sub Query]
[SELECT 절]
[FROM 절]
[WHERE 절]

[Sub Query]



- 하나의 Query 내 포함된 또 하나의 Query
- Sub Query는 괄호 안에 존재해야 한다.

▼ Data

- mypokemon

	number	name
▶	10	caterpie
	25	pikachu
	26	raichu
	133	eevee
	152	chikoirita

- ability

	number	type	height	weight	attack	defense	speed
▶	10	bug	0.3	2.9	30	35	45
	25	electric	0.4	6	55	40	90
	26	electric	0.8	30	90	55	110
	133	normal	0.3	6.5	55	50	55
	152	grass	0.9	6.4	49	65	45

[SELECT 절]

> Scalar Sub Query
> Scalar : 단일 차원의 값

- Sub query의 결과는 반드시 하나의 값이어야 한다.

▼ Grammer

```
SELECT [컬럼명],
                (SELECT [컬럼명]
                    FROM [테이블명]
                    WHERE 조건식)
FROM [테이블명]
WHERE 조건식;
```

▼ Example

- 피카츄의 번호와 이름, 키를 가져와 주세요

```
select number, name, (select height from ability where num
from mypokemon
where name = 'pikachu';
```

	number	name	height
▶	25	pikachu	0.4

▼ Desc

```
select height from ability where number = 25;
```

	height
▶	0.4

```
select T1.number, name, height
from pokemon.mypokemon as T1
left join pokemon.ability as T2
on T1.number = T2.number
where name = 'pikachu';
```

	number	name	height
▶	25	pikachu	0.4

[FROM 절]

> Inline View Sub Query
> Inline View : View와 같이 가상의 테이블

- Sub query의 결과는 반드시 하나의 테이블이어야 한다.

▼ Grammer

```
SELECT [컬럼 이름]
FROM (SELECT [컬럼 이름]
            FROM [테이블 이름]
            WHERE 조건식 ) AS [테이블 별명]
WHERE 조건식;
```

▼ Example

- 키 순위가 3순위인 포켓몬의 번호와 키 순위를 가져와 주세요

```
select number, height_rank
from (select number, rank() over(order by height desc) as
        from pokemon.ability) as T1
where height_rank=3;
```

	number	height_rank
▶	25	3

▼ Desc

```
select number, rank() over(order by height desc) as hei
from pokemon.ability;
```

	number	height_rank
▶	152	1
	26	2
	25	3
	10	4
	133	4

```
select T1.number, name, height,
        rank() over (order by height desc) as height_ra
from pokemon.mypokemon as T1
left join pokemon.ability as T2
on T1.number = T2.number;
```

	number	name	height	height_rank
▶	152	chikoirita	0.9	1
	26	raichu	0.8	2
	25	pikachu	0.4	3
	10	caterpie	0.3	4
	133	eevee	0.3	4

[WHERE 절]

> Nested Sub Queries
> Nested : 중첩

- Sub query의 결과는 반드시 하나의 컬럼이어야 한다.
  - 단, Exists는 단독으로 사용하며, 결과값이 여러 컬럼이어도 된다.

▼ 연산자

- 비교 연산자

연산자	활용	의미
=	A = [서브 쿼리]	A와 [서브 쿼리]의 결과값이 같다
!=	A != [서브 쿼리]	A와 [서브 쿼리]의 결과값이 같지 않다
>	A > [서브 쿼리]	A가 [서브 쿼리]의 결과값보다 크다
>=	A >= [서브 쿼리]	A가 [서브 쿼리]의 결과값보다 크거나 작다
<	A < [서브 쿼리]	A가 [서브 쿼리]의 결과값보다 작다
<=	A <= [서브 쿼리]	A가 [서브 쿼리]의 결과값보다 작거나 같다

- 주요 연산자
  - ALL : AND | ANY : OR

연산자	활용	의미
IN	A **IN**([서브 쿼리])	A가 [서브 쿼리]의 결과값 내에 있다.
ALL	A < **ALL**([서브 쿼리])	A가 모든 [서브 쿼리]의 결과값보다 작다.
	A > **ALL**([서브 쿼리])	A가 모든 [서브 쿼리]의 결과값보다 크다.
ANY	A < **ANY**([서브 쿼리])	A가 [서브 쿼리]의 결과값보다 하나라도 작다.
	A > **ANY**([서브 쿼리])	A가 [서브 쿼리]의 결과값보다 하나라도 크다.
※예외 EXISTS	**EXISTS** ([서브 쿼리])	[서브 쿼리]의 결과값이 존재한다.
	NOT EXISTS ([서브 쿼리])	[서브 쿼리]의 결과값이 존재하지 않는다.

▼ Grammer

```
SELECT [컬럼 이름]
FROM [테이블 이름]
WHERE [컬럼 이름] [연산자] ( SELECT [컬럼 이름]
                                                    FR
                                                    WH
```

▼ Example

▼ 키가 평균 키보다 작은 포켓몬의 번호를 가져와 주세요

```
select number
from pokemon.ability
where height <
    (select avg(height) from pokemon.ability);
```

	number
▶	10
	25
	133

▼ 공격력이 모든 전기 포켓몬의 공격력보다 작은 포켓몬의 번호를 가져와 주세요.

```
select number
from ability
where attack < all (select attack
                    from ability
                    where type = 'electric');
```

	number
▶	10
	152

▼ Desc

```
select *
from ability
where type = 'electric';
```

	number	type	height	weight	attack	defense	speed
▶	25	electric	0.4	6	55	40	90
	26	electric	0.8	30	90	55	110

▼ 방어력이 모든 전기 포켓몬의 공격력보다 하나라도 큰 포켓몬의 번호를 가져와 주세요

```
select number
from ability
where defense > any (select attack
                     from ability
                     where type = 'electric');
```

	number
▶	152

▼ bug 타입 포켓몬이 있다면 모든 포켓몬의 번호를 가져와 주세

```
select number
from ability
```

```
where exists(select * from ability where type='bug');
```

	number
▶	10
	25
	26
	133
	152