SQL 기초와 데이터 분석

하홍석

programmers

3. 여러 테이블 결합하여 사용하기

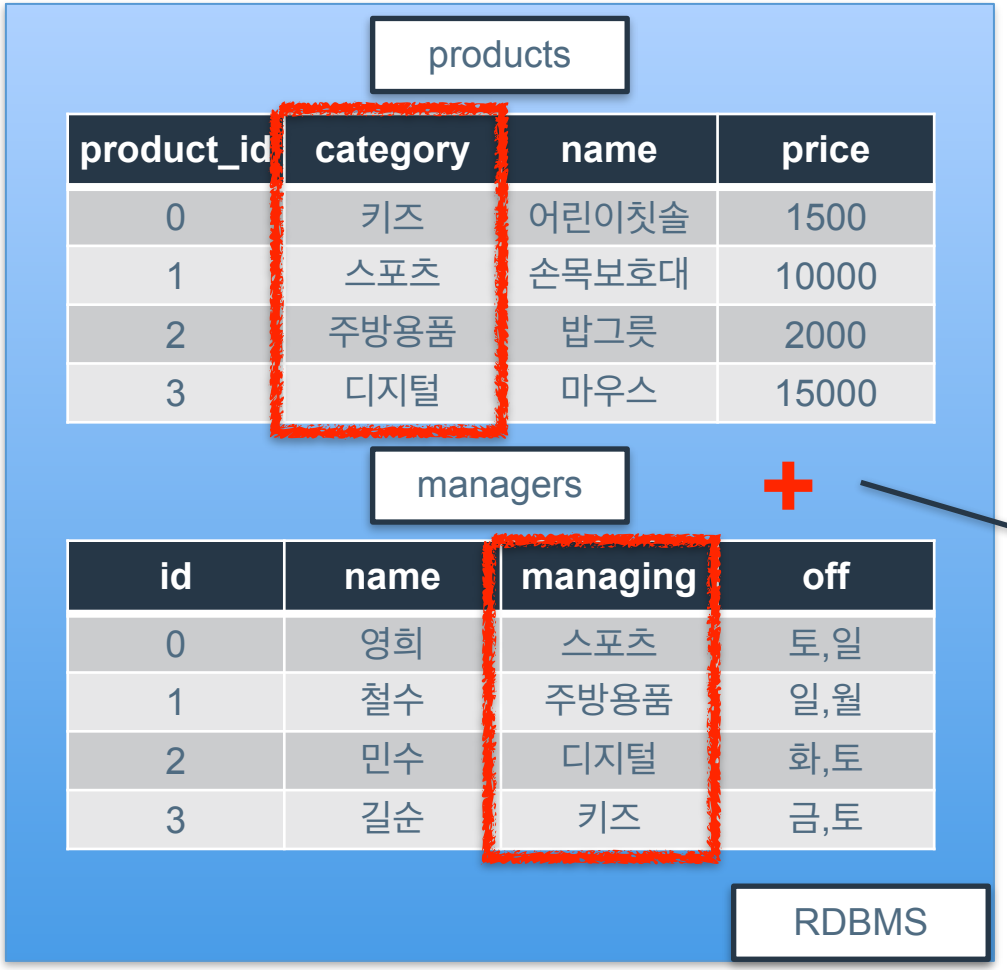
3. 여러 테이블 결합하여 사용하기

1. 다양한 JOINS
2. UNION
3. WITH
4. Subquery

다양한 JOINS

다양한 JOINS : 개요

- JOIN : 두 개 이상의 테이블을 특정 key를 기준으로 결합하는 것



Query

product_id	category	name	price	manager_name
0	키즈	어린이칫솔	1500	길순
1	스포츠	손목보호대	10000	영희
2	주방용품	밥그릇	2000	철수
3	디지털	마우스	15000	민수

다양한 JOINS : INNER JOIN

products

product_id	category	name	price
0	키즈	어린이칫솔	1500
1	스포츠	손목보호대	10000
2	주방용품	밥그릇	2000
3	디지털	마우스	15000

managers

id	name	managing	off
0	영희	스포츠	토,일
1	철수	주방용품	일,월
2	민수	디지털	화,토
3	길순	키즈	금,토

RDBMS

Query

```
SELECT products.*, managers.name as
manager_name
FROM products INNER JOIN managers on
products.category = managers.managing
```

product_id	category	name	price	manager_name
0	키즈	어린이칫솔	1500	길순
1	스포츠	손목보호대	10000	영희
2	주방용품	밥그릇	2000	철수
3	디지털	마우스	15000	민수

다양한 JOINS : LEFT JOIN

clicks

clk_index	product_id	user_name	date
0	0	영희	20231014
1	0	철수	20231014
2	0	영희	20231014
3	0	길순	20231014

orders

odr_index	product_id	user_name	date
0	0	영희	20231014
1	2	길순	20231014
2	3	길동	20231015
3	1	길동	20231015

RDBMS

Query

```
SELECT clicks.*, odr_index
FROM clicks LEFT JOIN orders
    on clicks.user_name = orders.user_name
    and clicks.product_id = orders.product_id
    and clicks.date = orders.date
```

clk_index	product_id	user_name	Date	odr_index
0	0	영희	20231014	0
2	0	영희	20231014	0
1	0	철수	20231014	(null)
3	0	길순	20231014	(null)

다양한 JOINS : RIGHT JOIN

clicks

clk_index	product_id	user_name	date
0	0	영희	20231014
1	0	철수	20231014
2	0	영희	20231014
6	2	길순	20231014
12	1	길동	20231015
14	3	길동	20231015

orders

odr_index	product_id	user_name	date
0	0	영희	20231014
1	2	길순	20231014
2	3	길동	20231015
3	1	길동	20231015

RDBMS

Query

```
SELECT orders.*, clk_index
FROM clicks RIGHT JOIN orders
    on clicks.user_name = orders.user_name
    and clicks.product_id = orders.product_id
    and clicks.date = orders.date
```

odr_index	product_id	user_name	Date	clk_index
0	0	영희	20231014	0
0	0	영희	20231014	2
1	2	길순	20231014	6
3	1	길동	20231015	12
2	3	길동	20231015	14

다양한 JOINS : FULL OUTER JOIN

clicks

clk_index	product_id	user_name	date
0	0	영희	20231014
1	0	철수	20231014
2	0	영희	20231014
3	0	길순	20231014

orders

odr_index	product_id	user_name	date
0	0	영희	20231014
1	2	길순	20231014
2	3	길동	20231015
3	1	길동	20231015

RDBMS

Query

```
SELECT orders.*, clk_index
FROM clicks LEFT JOIN orders
    on clicks.user_name = orders.user_name
    and clicks.product_id = orders.product_id
    and clicks.date = orders.date
UNION
SELECT orders.*, clk_index
FROM clicks RIGHT JOIN orders
    on clicks.user_name = orders.user_name
    and clicks.product_id = orders.product_id
    and clicks.date = orders.date
```

odr_index	product_id	user_name	Date	clk_index
0	0	영희	20231014	0
0	0	영희	20231014	2
(null)	(null)	(null)	(null)	1
(null)	(null)	(null)	(null)	3
1	2	길순	20231014	(null)
2	3	길동	20231015	(null)
3	1	길동	20231015	(null)

다양한 JOINS : FULL OUTER JOIN

odr_index	product_id	user_name	Date	clk_index
0	0	영희	20231014	0
0	0	영희	20231014	2
(null)	(null)	(null)	(null)	1
(null)	(null)	(null)	(null)	3
1	2	길순	20231014	(null)
2	3	길동	20231015	(null)
3	1	길동	20231015	(null)

INNER JOIN

LEFT JOIN

RIGHT JOIN

다양한 JOINS : CROSS JOIN (=Cartesian product)

clicks

clk_index	product_id	user_name	date
0	0	영희	20231014
1	0	철수	20231014
2	0	영희	20231014

orders

odr_index	product_id	user_name	date
0	0	영희	20231014
1	2	길순	20231014
2	3	길동	20231015

RDBMS

Query

```
SELECT clicks.*, odr_index,
orders.product_id as
odr_product_id,
orders.user_name as
odr_user_name,
orders.date as odr_date
FROM clicks CROSS JOIN orders
```

Result

다양한 JOINS : CROSS JOIN (=Cartesian product)

clk_index	product_id	user_name	date	odr_index	odr_product_id	odr_user_name	odr_date
0	0	영희	20231014	0	0	영희	20231014
1	0	철수	20231014	0	0	영희	20231014
2	0	영희	20231014	0	0	영희	20231014
0	0	영희	20231014	1	2	길순	20231014
1	0	철수	20231014	1	2	길순	20231014
2	0	영희	20231014	1	2	길순	20231014
0	0	영희	20231014	2	3	길동	20231015
1	0	철수	20231014	2	3	길동	20231015
2	0	영희	20231014	2	3	길동	20231015

다양한 JOINS : Alias (=별칭)

clicks

clk_index	product_id	user_name	date
0	0	영희	20231014
1	0	철수	20231014
2	0	영희	20231014

orders

odr_index	product_id	user_name	date
0	0	영희	20231014
1	2	길순	20231014
2	3	길동	20231015

RDBMS

Query

```
SELECT c.*, o.odr_index, p.name
FROM clicks c LEFT JOIN orders o
    on c.user_name = o.user_name
    and c.product_id =
o.product_id
    and c.date = o.date
INNER JOIN products p on
c.product_id = p.product_id
```

clk_index	product_id	user_name	date	odr_index	name
0	0	영희	20231014	0	어린이칫솔
2	0	영희	20231014	0	어린이칫솔
1	0	철수	20231014	(null)	어린이칫솔

다양한 JOINS : SELF JOIN

managers_v2

id	name	managing	off	substitute
0	영희	스포츠	토,일	1
1	철수	주방용품	일,월	2
2	민수	디지털	화,토	3
3	길순	키즈	금,토	0

RDBMS

Query

```
SELECT m1.*, m2.id as sub_id,
m2.name as sub_name
FROM managers_v2 m1 INNER JOIN
managers_v2 m2
ON m1.substitute = m2.id
```

id	name	managing	off	substitute	sub_id	sub_name
0	영희	스포츠	토,일	1	1	철수
1	철수	주방용품	일,월	2	2	민수
2	민수	디지털	화,토	3	3	길순
3	길순	키즈	금,토	0	0	영희

다양한 JOINS : 필터링

clicks

clk_index	product_id	user_name	date
0	0	영희	20231014
4	1	철수	20231014
5	1	길동	20231014
12	1	길동	20231015

orders

odr_index	product_id	user_name	date
0	0	영희	20231014
3	1	길동	20231015

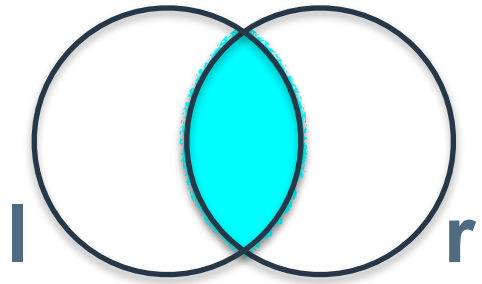
RDBMS

Query

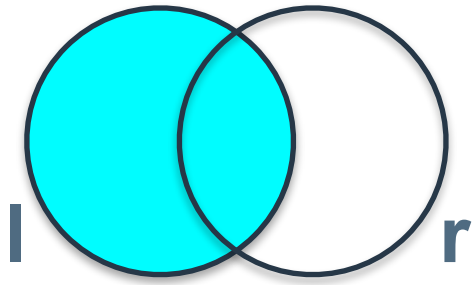
```
SELECT c.*, o.odr_index, p.name
FROM clicks c LEFT JOIN orders o
    on c.user_name = o.user_name
    and c.product_id = o.product_id
INNER JOIN products p on
c.product_id = p.product_id
WHERE p.category = '스포츠'
```

clk_index	product_id	user_name	date	odr_index	name
5	1	길동	20231014	3	손목보호대
4	1	철수	20231014	(null)	손목보호대
12	1	길동	20231015	3	손목보호대

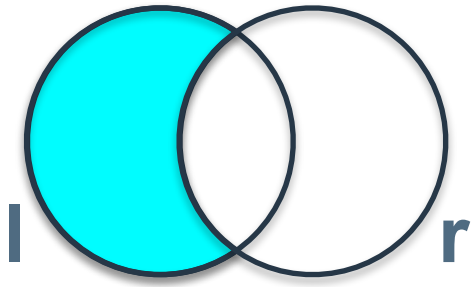
다양한 JOINS : 정리 (1)




```
SELECT *
FROM l INNER JOIN r
on l.key = r.key
```

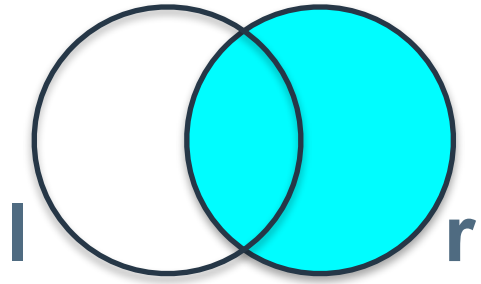



```
SELECT *
FROM l LEFT JOIN r
on l.key = r.key
```

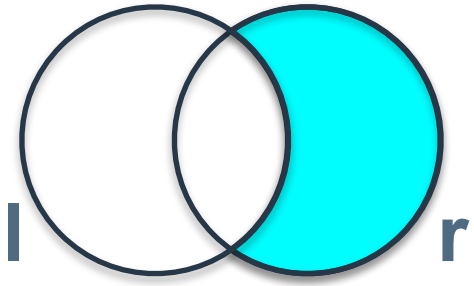



```
SELECT *
FROM l LEFT JOIN r
on l.key = r.key
WHERE r.key is NULL
```

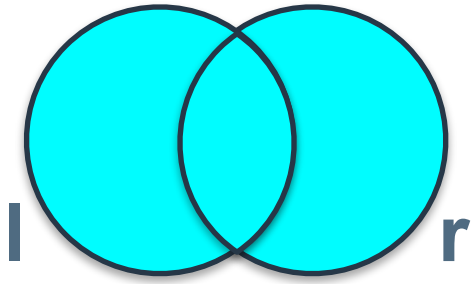
다양한 JOINS : 정리 (2)




```
SELECT *
FROM l RIGHT JOIN r
on l.key = r.key
```

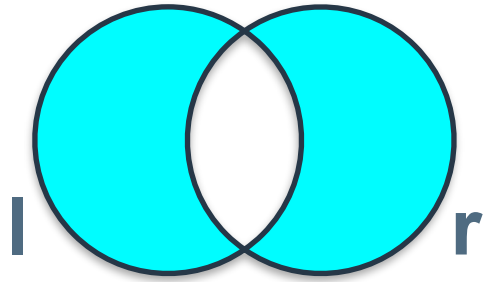



```
SELECT *
FROM l RIGHT JOIN r
on l.key = r.key
WHERE l.key is NULL
```




```
SELECT *
FROM l LEFT JOIN r
on l.key = r.key
UNION
SELECT *
FROM l RIGHT JOIN r
on l.key = r.key
```

다양한 JOINS : 정리




```
SELECT *
FROM l LEFT JOIN r
on l.key = r.key
UNION
SELECT *
FROM l RIGHT JOIN r
on l.key = r.key
WHERE l.key is NULL
OR r.key is NULL
```

UNION

UNION

RDBMS

products

product_id	category	name	price
0	키즈	어린이칫솔	1500
1	스포츠	손목보호대	10000
2	주방용품	밥그릇	2000
3	디지털	마우스	15000

products_B

product_id	category	name	price
33	키즈	어린이칫솔	2000
34	스포츠	무릎보호대	13000
35	주방용품	국자	15000

Query

```
SELECT *
FROM products
UNION
SELECT *
FROM products_B
```

product_id	category	name	price
0	키즈	어린이칫솔	1500
1	스포츠	손목보호대	10000
2	주방용품	밥그릇	2000
3	디지털	마우스	15000
33	키즈	어린이칫솔	2000
34	스포츠	무릎보호대	13000
35	주방용품	국자	15000

UNION : UNION ALL

products

product_id	category	name	price
0	키즈	어린이칫솔	1500
1	스포츠	손목보호대	10000
2	주방용품	밥그릇	2000
3	디지털	마우스	15000

products_B

product_id	category	name	price
33	키즈	어린이칫솔	2000
34	스포츠	무릎보호대	13000
35	주방용품	국자	15000

RDBMS

Query

```
SELECT category
FROM products
UNION ALL
SELECT category
FROM products_B
```

category
키즈
스포츠
주방용품
디지털
키즈
스포츠
주방용품

WITH

WITH

1. CTE (Common Table Expression) 라고도 부르며, MySQL 8.0 버전 이상에서 지원한다.
2. 임시 결과 집합을 생성하여 복잡한 쿼리를 쉽게 작성할 수 있도록 돕는 기능을 한다.
3. 복잡한 쿼리에서 하위 쿼리를 사용해 같은 결과를 여러 번 계산해야 하는 경우를 줄여 준다.
    1. CTE는 같은 쿼리 블록을 여러 번 사용할 수 있도록 함
4. 쿼리 가독성을 높여 유지보수를 용이하게 한다.
5. DB Optimizer는 CTE를 단순한 뷰나 서브쿼리보다 더 효율적으로 처리한다.

* https://www.programiz.com/sql/online-compiler/

```
WITH user_orders as (
    SELECT user_name, SUM(price) as total_purchase
    FROM orders o INNER JOIN products p on
o.product_id = p.product_id
    GROUP BY 1
    ORDER BY 2 DESC
)
SELECT user_orders.*
FROM user_orders uo INNER JOIN managers m ON
uo.user_name = m.user_name
```

Subquery

programmers

Subquery

1. 유저별 평균 구매 가격과, 전체 평균 구매 가격을 비교하기

```
SELECT user_name, AVG(price) as avg_price,
(SELECT AVG(price)
 FROM orders o INNER JOIN products p
 on o.product_id = p.product_id) as total_avg_price
FROM orders o INNER JOIN products p
on o.product_id = p.product_id
GROUP BY 1
ORDER BY 2 DESC
```

2. 스포츠/주방용품 매니저들의 클릭 이력을 가져오기

```
SELECT c.*
FROM (SELECT name
      FROM managers
      WHERE managing in ('스포츠', '주방용품')) a
INNER JOIN clicks c on a.name = c.user_name
```

Subquery

3. 가장 비싼 상품에 대한 클릭 이력 가져오기

```
SELECT c.*
FROM clicks c
WHERE product_id = (SELECT product_id FROM
products ORDER BY price DESC LIMIT 1)
```

```
SELECT c.*
FROM clicks c INNER JOIN products p on
c.product_id = p.product_id
WHERE price >= ALL(SELECT price FROM products)
```

Subquery

4. 매니저들 중에 상품을 구매한 사람이 있는 경우, 구매 테이블 전체를 출력하기

```
SELECT o.*
FROM orders o
WHERE EXISTS (SELECT user_name FROM orders o
inner join managers m on o.user_name = m.name)
```

End of Document