SQL 기초와 데이터 분석

하홍석

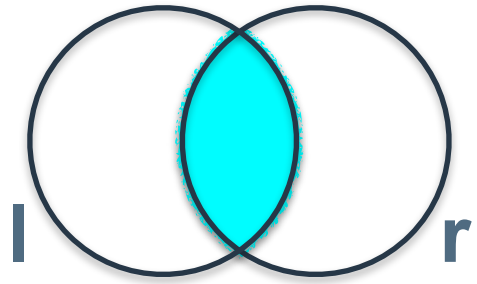
programmers

6. 효율적인 SQL 코드 작성하기

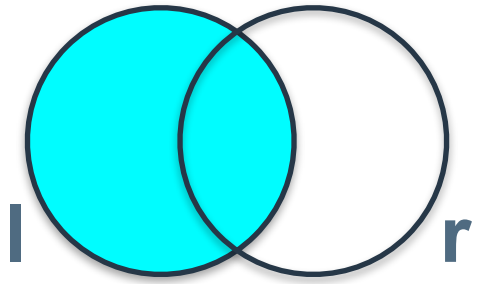
6. 효율적인 SQL 코드 작성하기

1. 테이블을 집합으로 생각하기
2. *, % 사용 지양하기
3. 데이터 타입 잘 확인하기
4. JOIN 시 유의할 점
5. 가독성 높이기

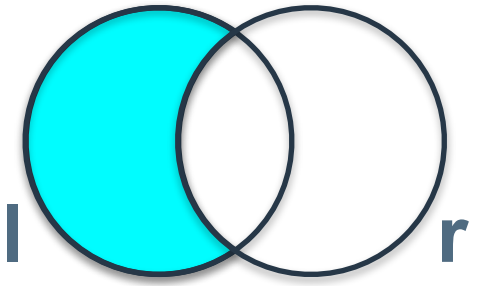
1. 테이블을 집합으로 생각하기



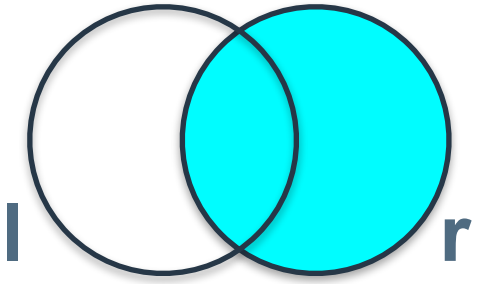
INNER JOIN



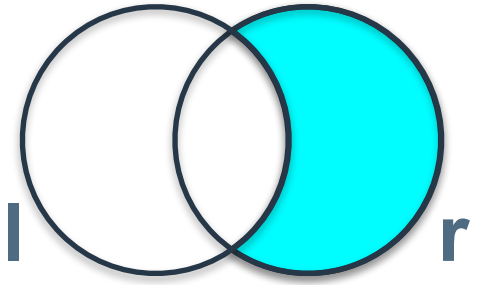
LEFT JOIN



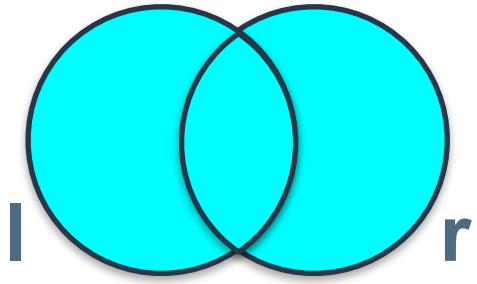
LEFT ANTI JOIN



RIGHT JOIN



RIGHT ANTI JOIN



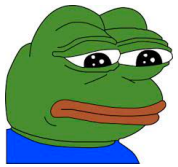
FULL OUTER JOIN

1. 테이블을 집합으로 생각하기

- 최대한 작게 만들어 놓고 JOIN 하기

```
WITH customer as (
    SELECT customer_id, customer_name, customer_tier
    FROM Customers
    WHERE customer_id = 1
),
order_info as (
    SELECT customer_id, order_id, amount
    FROM Orders
    WHERE customer_id = 1
)
SELECT c.customer_id, customer_name, customer_tier,
count(distinct order_id) as odr_cnt, sum(amount) as
total_purchase
FROM customer c INNER JOIN order_info o on
c.customer_id = o.customer_id
GROUP BY 1, 2, 3
ORDER BY 4 DESC
```

```
WITH customer as (
    SELECT customer_id, customer_name, customer_tier
    FROM Customers
),
order_info as (
    SELECT customer_id, order_id, amount
    FROM Orders
)
SELECT c.customer_id, customer_name, customer_tier,
count(distinct order_id) as odr_cnt, sum(amount) as
total_purchase
FROM customer c INNER JOIN order_info o on
c.customer_id = o.customer_id
WHERE c.customer_id = 1
GROUP BY 1, 2, 3
ORDER BY 4 DESC
```



2. *, % 사용 지양하기

1. LIMIT 걸고 조회하기

```
SELECT product_id, category, name
FROM products
LIMIT 10
```

2. 파티션이 있는 테이블인지 확인하고, 파티션을 필터 조건으로 걸고 조회하기

```
SELECT clk_index, user_name, product_id
FROM clicks
WHERE date = '20231104'
LIMIT 10
```

3. 컬럼 수가 많은 테이블을 조회할 때 SELECT * 지양하기

```
SELECT product_id, category_name, sales_yn
FROM products
LIMIT 10
```

4. LIKE 사용 시 % 제한적으로 사용하기

```
SELECT product_id, name
FROM products
WHERE name LIKE '23FW%'
LIMIT 10
```

```
SELECT product_id, name
FROM products
WHERE name LIKE '23FW__'
LIMIT 10
```

3. 데이터 타입 잘 확인하기

1. 비교 연산자를 쓸 때 타입을 확인하기

```
SELECT count(1)
FROM clicks
WHERE date > '20231031'
```

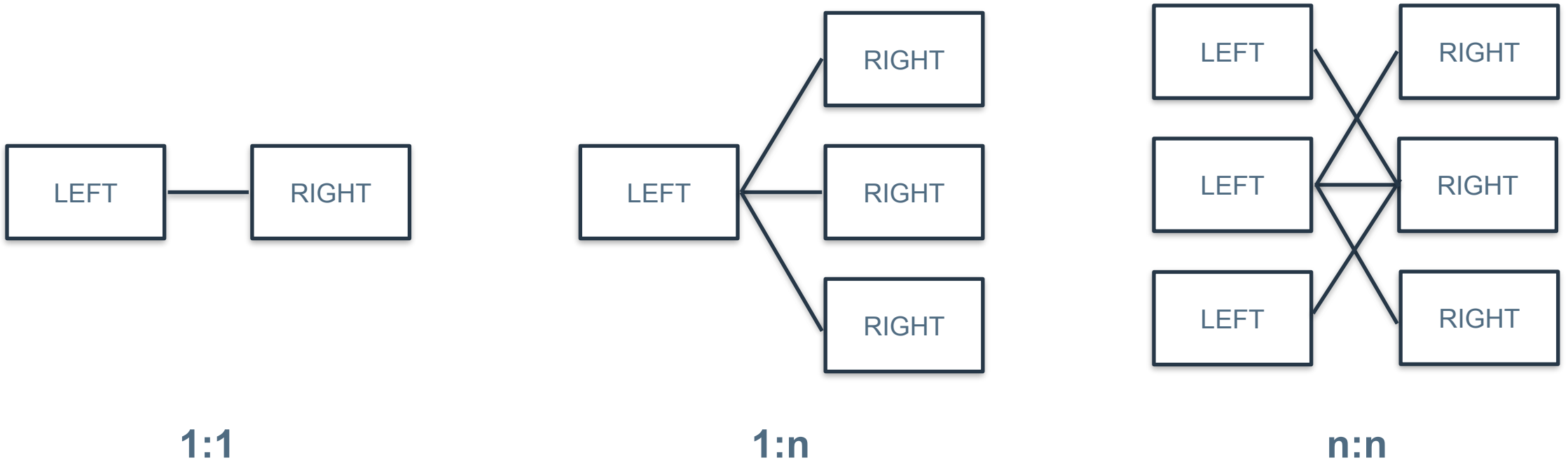
2. WHERE 절에서 왼쪽 컬럼에 함수 적용 지양하기

```
WITH clk as (
    SELECT clk_index, event_index, date
    FROM clicks

),
event as (
    SELECT event_index, DATE_FORMAT(event_end, '%Y%m%d') as event_end_dt
    FROM events
)

SELECT count(1)
FROM clk INNER JOIN event ON clk.event_index = event.event_index
WHERE date <= event_end_dt
```

1. JOIN 하는 테이블 간의 관계를 고려하기



ER Model : https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model

2. 데이터 중복이 있는지 확인하기

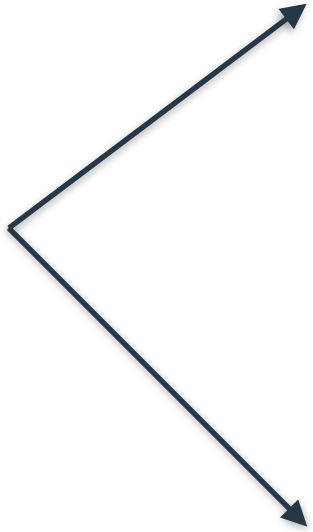
클릭 로그

name	clicked
영희	칫솔
영희	치약
철수	칫솔
철수	보조배터리
길동	보조배터리
길동	치약
길순	칫솔
길순	보조배터리

+

구매 로그

name	bought
영희	칫솔
길순	칫솔
철수	보조배터리
철수	보조배터리



product	clicked	bought	cvr
칫솔	3	2	0.67
치약	2	0	0.00
보조배터리	3	2	0.67

product	clicked (unique)	bought (unique)	cvr
칫솔	3	2	0.67
치약	2	0	0.00
보조배터리	3	1	0.33

3. 여러 가지 쿼리 방식을 고려하자

```
SELECT user_name, product_id
FROM (
    SELECT user_name, clk_index, product_id,
row_number() over(PARTITION BY user_name
ORDER BY clk_index ASC) as rownum
    FROM clicks
    ) w
WHERE rownum = 1
```

0.021 초

```
WITH clk as (
    SELECT user_name, product_id, clk_index
    FROM clicks
),
mini as (
    SELECT user_name, min(clk_index) as min_idx
    FROM clicks
    GROUP BY 1
)
SELECT clk.user_name, clk.product_id
FROM clk INNER JOIN mini ON clk.user_name =
mini.user_name AND clk.clk_index = min_idx
```

0.015 초

```
SELECT c.user_name, product_id
FROM clicks c
    INNER JOIN (
        SELECT user_name, min(clk_index) as
min_idx
        FROM clicks
        GROUP BY 1
    ) g
    ON c.user_name = g.user_name
    AND c.clk_index = min_idx
```

0.016 초

5. 가독성 높이기

1. 서브쿼리 보다는 WITH 구문이 가독성이 좋다
2. WITH 절을 사용할 때, 각 블록 이름을 잘 지정하자
3. 쿼리가 복잡해 지면 중간중간 주석을 작성하자

```
WITH odr_cnt as (
  -- 고객별 구매 수
  SELECT c.customer_id, count(distinct
order_id) as odr_cnt, sum(amount) as
total_purchase
  FROM Customers c INNER JOIN Orders o on
c.customer_id = o.customer_id
  GROUP BY 1
  ORDER BY 2 DESC
),
ship_cnt as (
  -- 고객별 배송 수
  SELECT c.customer_id, count(distinct
shipping_id) as ship_cnt
  FROM Customers c INNER JOIN Shippings s on
c.customer_id = s.customer
  WHERE status = 'Pending'
  GROUP BY 1
  ORDER BY 2 DESC

)
SELECT oc.customer_id, odr_cnt, total_purchase,
COALESCE(ship_cnt, 0) as shipping_cnt
FROM odr_cnt oc LEFT JOIN ship_cnt sc on
oc.customer_id = sc.customer_id
```

End of Document