



# 10.02 메모

🕒 Created @2024년 10월 2일 오전 9:23

## SQL 함수

```
SELECT position('T' in 'XTZ'); -- 문자열에서 문자 위치 반환
```

```
select substring('chunjae education', -3, 3); -- -3번부터 3자리
```

```
select "userid", substring("userid", 7) -- 7번째부터의 문자열 반환  
from "learning_analytics"."e_assessment"  
limit 5;
```

```
select upper('abc'); -- 대문자화
```

```
select lower('ABC'); -- 소문자화
```

```
select upper(substring("userid", -12)) -- -12에서부터의 문자열 반  
from "learning_analytics"."e_assessment"  
limit 5;
```

```
select "userid", regexp_extract("userid", '[0-9a-z]+') --정규표  
from "learning_analytics"."e_assessment"  
limit 5;
```

```
select concat('abc', '123'); -- 문자열 합치기
```

```
select 'abc' || '1123'; -- 콘켓과 같은 결과
```

```
select length('chunjae education'); -- 문자열 길이
```

```
select replace('textbook', 'x', 's'); -- 문자열 대체(대체할 문자열
```

```
select rpad('abc', 9, 'xyz'); -- 오른쪽에서 문자열 채우기 (채워질 문  
select lpad('abc', 8, 'xyz'); -- 왼쪽에서 문자열 채우기
```

```
select abs(-3.14); -- 절댓값
```

```
select ceiling(-3.14); -- 올림  
select floor(-3.14); -- 내림
```

```
select round(-3.141592, 2); -- 소수점 둘째자리까지 반올림
```

```
select truncate(-3.141592); -- 소수점을 전부 버림.
```

```
select power(-3.141592, 2); -- 앞 숫자를 뒤 숫자만큼 제공해서 반환
```

```
select mod(5,2); -- 앞 숫자를 뒤 숫자로 나눈 나머지 반환  
select 5%2;
```

```
select now(); -- 현재시간  
select year(now()); -- 현재 년  
select quarter(now()); -- 현재 분기  
select month(now()), week(now()), day(now()), hour(now()), mi
```

```
select day_of_year(now()) as "연중 몇번째 일", day_of_week(now())
```

```

SELECT NOW() AS "현재"
, DATE_TRUNC('year', NOW())
, DATE_TRUNC('quarter', NOW())
, DATE_TRUNC('month', NOW())
, DATE_TRUNC('week', NOW())
, DATE_TRUNC('day', NOW())
, DATE_TRUNC('hour', NOW())
, DATE_TRUNC('minute', NOW())
, DATE_TRUNC('second', NOW())
, DATE_TRUNC('millisecond', NOW())
; -- 남길 것만 두고 초기화

```

```

SELECT NOW() AS "현재"
, DATE_ADD('year', 1, NOW())
, DATE_ADD('quarter', 1, NOW())
, DATE_ADD('month', 1, NOW())
, DATE_ADD('week', 1, NOW())
, DATE_ADD('day', 1, NOW())
, DATE_ADD('hour', 1, NOW())
, DATE_ADD('minute', 1, NOW())
, DATE_ADD('second', 1, NOW())
, DATE_ADD('millisecond', 1, NOW())
; -- 날짜 데이터 더하기

```

```

SELECT NOW() AS "현재"
, DATE_DIFF('year', DATE_ADD('year', 1, NOW()), NOW())
, DATE_DIFF('quarter', DATE_ADD('quarter', 1, NOW()), NOW())
, DATE_DIFF('month', DATE_ADD('month', 1, NOW()), NOW())
, DATE_DIFF('week', DATE_ADD('week', 1, NOW()), NOW())
, DATE_DIFF('day', DATE_ADD('day', 1, NOW()), NOW())
, DATE_DIFF('hour', DATE_ADD('hour', 1, NOW()), NOW())
, DATE_DIFF('minute', DATE_ADD('minute', 1, NOW()), NOW())
, DATE_DIFF('second', DATE_ADD('second', 1, NOW()), NOW())
, DATE_DIFF('millisecond', DATE_ADD('millisecond', 1, NOW()),
; -- 왼쪽에서 오른쪽 빼기

```

```

SELECT NOW()
, DATE_FORMAT(NOW(), '%Y-%m-%d')
, DATE_FORMAT(NOW(), '%Y%m%d')
, DATE_FORMAT(NOW(), '%Y%m')
, DATE_FORMAT(NOW(), '%H:%i:%s')
, DATE_FORMAT(NOW(), '%Y')
, DATE_FORMAT(NOW(), '%T')
; -- 날짜 데이터 포맷 설정

```

```

SELECT NOW()
-- , CONCAT(CURRENT_DATE, CURRENT_TIME)
, CONCAT(DATE_FORMAT(NOW(), '%Y-%m-%d'), DATE_FORMAT(NOW(), '%H:%i:%s'))
; -- 포맷 합치기

```

```

select date_parse('2024-10-02', '%Y-%m-%d');

```

```

select now(),
cast(cast(now() as varchar) as timestamp);

```

## rank

```

select
"proc_ymd", "userid", "mcode", "lecture_type", "eventtime_kst",
row_number() over(partition by "proc_ymd", "userid", "mcode"
-- "proc_ymd", "userid", "mcode", "lecture_type"로 그룹화 한 후
-- eventtime_kst 가 같은 경우 같은 순위임.
, rank() over(partition by "proc_ymd", "userid", "mcode", "lecture_type"
-- 공동순위 다음에 공동순위 수만큼 다음 순위 늘어남 ( 예로 11 11 11 다음
, dense_rank() over(partition by "proc_ymd", "userid", "mcode"
-- 공동 순위 다음에 다음 순위 늘어나지 않음( 11 11 11 다음에 12)
from "learning_analytics"."e_learning_action"
where 1=1
and "userid" = '41f130e0-a18a-404e-8688-99a391eca5c2'

```

```

select
*,
sum("sum_point") over (partition by "userid"
order by "proc_ymd" asc
rows between unbounded preceding and unbounded following) as
-- 여기서는 파티션 내에서 맨 앞부터 맨 끝까지 더함.
from(
select
"proc_ymd", "userid",
sum("point") as "sum_point"
from "learning_analytics"."e_point"
where 1=1
and "point" > 0
and "proc_ym" = '202403'
group by "proc_ymd", "userid"
)

```

## with

```

WITH "temp1" AS (
    SELECT DISTINCT "userid"
    FROM "learning_analytics"."e_assessment"
    WHERE "proc_ymd" = '20230101'
),
"temp2" AS (
    SELECT DISTINCT "userid"
    FROM "learning_analytics"."e_assessment_detail"
    WHERE "proc_ymd" = '20230101'
)

SELECT COUNT(*)
FROM "temp1"
FULL OUTER JOIN "temp2" ON "temp1"."userid" = "temp2"."userid"

SELECT
    "ea"."proc_ymd",
    "ea"."userid",

```

```

        "ea"."mcode",
        "ea"."score",
        "ead"."no",
        "ead"."answer",
        "ead"."correct"
FROM "learning_analytics"."e_assessment" AS "ea"
INNER JOIN "learning_analytics"."e_assessment_detail" AS "ead"
    ON "ea"."userid" = "ead"."userid"
    AND "ea"."proc_ymd" = "ead"."proc_ymd"
    AND "ea"."mcode" = "ead"."mcode"
WHERE "ea"."proc_ymd" = '20230101'
ORDER BY "ea"."proc_ymd", "ea"."userid", "ea"."mcode", "ead".

```

## union

```

select *
from (select "proc_ym",
count() as "count",
'e_learning_action' as "table_name"
from "learning_analytics"."e_learning_action"
group by "proc_ym"
union all
select "proc_ym",
count() as "count",
'e_learning_time' as "table_name"
from "learning_analytics"."e_learning_time"
group by "proc_ym"
union all
select "proc_ym",
count() as "count",
'e_assessment' as "table_name"
from "learning_analytics"."e_assessment"
group by "proc_ym"
union all
select "proc_ym",
count() as "count",
'e_assessment_detail' as "table_name"
from "learning_analytics"."e_assessment_detail"

```

```
group by "proc_ym"  
)  
order by "proc_ym"
```

```
with _temp1 as (  
values (char '202208')  
)  
, _temp2 ("yyyy", "mm") as (  
values (varchar '2022', varchar '08')  
)  
, _temp3 as (  
select *  
from "learning_analytics"."e_assessment"  
where concat("yyyy", "mm") = '202208'  
)  
, _temp4 as (  
select *  
from "learning_analytics"."e_assessment"  
where concat("yyyy", "mm") = (select * from _temp1)  
)  
, _temp5 as(  
select *  
from "learning_analytics"."e_assessment"  
where 1=1  
and "yyyy" = (select "yyyy" from _temp2)  
and "mm" = (select "mm" from _temp2)  
)  
, _temp6 ("년월일", "아이디", "mcode", "점수") as (  
select "proc_ymd", "userid", "mcode", "score"  
from _temp5  
)  
select *  
from _temp6
```

10/10일까지 과제

text\_biz\_dw 데이터베이스를 활용하여  
다음 요구사항을 만족하는 SQL을 작성하세요.  
2022년 분기 별 (총 4분기) 콘텐츠 이용  
실태 조사를 진행하고자 합니다.

초등 3학년, 4학년, 5학년, 6학년을 대상으로 하는 콘텐츠 중  
영상강의 + 문제풀이가 함께 서비스되는 콘텐츠를 대상으로 하여,

1. 콘텐츠 별 학습을 진행한 학생 수
2. 콘텐츠 별 학습을 진행한 학생의 학년 평균 (학생학년별 평균이 아니라 학년값의 평균)
3. 콘텐츠 별 학습 시간
4. 콘텐츠 별

평가문항 평균 갯수, 정답 문항 평균 갯수, 평가점수 평균  
(평균 값은 소수 둘째자리까지 표현 해 주세요.)

을 확인할 수 있는 쿼리를 전달해 주세요.

이 때, WITH 문을 사용하여

사용자의 분기 입력을 통해 분기 별 데이터를 조회할 수 있는 방법을 만들어 주세요.

사용 테이블 : e\_media, e\_content\_meta, e\_test, e\_study, e\_member

#### 평가기준

1. 쿼리가 정상적으로 실행 가능한가?
2. 행수가 답과 일치하는가?
3. 사용자 기능이 구현되었는가?
4. 적절한 컬럼을 선택하였는가?