



9.30 메모

🕒 Created	@2024년 9월 30일 오전 8:58
📁 Class	9.30~10.10 데이터베이스 구축

데이터베이스 구축

강사님 : 김지훈

aws 로그인해서 열어보기.

aws athena 사용해서 테이블 미리보기 하면 쿼리문이 자동으로 짜진다.

sql 활용 시의 중요 표현들

AS : 데이터에 별명을 지정

LIMIT : 출력할 데이터의 갯수를 지정

Trim(문자열) : 양쪽 공백 제거

```
select 1; (사칙연산 가능)
```

```
SELECT 'abc'; (문자열 출력 가능)
```

```
select concat('가', '나'); (문자열 합치기)
```

```
select current_timestamp; (현재시간 출력)
```

```
select 1>21; (불린값 출력)
```


```
select 'proc_ymd', 'userid', 'mcode', 'completed_yn'
```

```

from "learning_analytics"."e_milkt_study"
limit 5;

select *, "proc_ym" as "처리일자"
from "learning_analytics"."e_milkt_study"
limit 5;

```

 : 애스터리스크 : 테이블의 모든 컬럼을 조회 시 사용.

주의

db, 테이블명, 선택할 컬럼은 큰따옴표로 쓸 것. (as로 설정한 별명도)

문자열은 작은따옴표로 쓸것.

연산자	예제	의미
=	a = b	a와 b가 같다
!= (<>)	a != b	a와 b가 같지 않다
>	a > b	a가 b보다 크다
>=	a >= b	a가 b보다 크거나 같다
<	a < b	a가 b보다 작다
<=	a <= b	a가 b보다 작거나 같다
AND	a AND b	a와 b 둘 다 'true'이면 true
OR	a OR b	a와 b 둘 중 하나라도 'true'이면 true
NOT	NOT a	a가 아니면 true
BETWEEN	BETWEEN a AND b	a부터 b 범위 내의 데이터를 선택
IN	IN (a, b, c, ...)	IN 이하의 목록에 포함된 데이터를 선택
LIKE	LIKE '문자열'	특정 문자열에 포함된 데이터를 선택 (% : 0개 이상의 문자, _:1개 문자)
IS NULL	IS NULL	데이터가 Null인지 아닌지를 확인

cast() 함수

CAST(표현식 AS 데이터타입) : 기재한 데이터 타입으로 데이터 타입 변경.

like로 문자열 찾을 때 정규식으로 찾을 수도 있다.

```
ex)
regexp_like("l_template_nm", '^.+_.*$')
```

INFORMATION_SCHEMA

```
SELECT *
FROM "information_schema"."columns"
WHERE 1=1
    and "table_schema" = 'learning_analytics'
    AND "table_name" = 'e_learning_time';
```

TABLE 의 컬럼 정보를 확인할 수 있다.

IF

IF(조건, 참일 경우의 값, 거짓일 경우의 값)

```
select *,
if("mm" in ('01', '02', '03', '04', '05', '06'), '상반기', '하반기')
from "learning_analytics"."e_milkt_study"
where 1=1
and "yyyy" = '2023'
and "mm" = '01'
limit 10;
```

CASE

```

SELECT
    *,
    CASE
        WHEN "mm" IN ('01', '02', '03') THEN '1분기'
        WHEN "mm" IN ('04', '05', '06') THEN '2분기'
        WHEN "mm" IN ('07', '08', '09') THEN '3분기'
        WHEN "mm" IN ('10', '11', '12') THEN '4분기'
        ELSE '기타'
    END AS "quarter" (케이스를 끝내고, 별명을 quarter로 정함)
FROM "learning_analytics"."e_milkt_study"
WHERE 1=1
    AND "yyyy" = '2023'
    AND "mm" = '01'
LIMIT 10;

```

DISTINCT

선택한 컬럼의 중복을 제거

```

SELECT DISTINCT "user_grade", "user_grade_nm"
FROM "learning_analytics"."e_milkt_study";

```

group by

```

select "biz_sect_cd_nm",
count("system_id") as "count_member",
count(distinct "system_id") as "count_distinct_memver"
from "learning_analytics"."member_segment"
where 1=1
and "proc_ymd" = '20240601'
group by "biz_sect_cd_nm"
limit 500;

```

SELECT 쿼리 작성 순서 vs 실행 순서

SELECT 쿼리의 작성 순서

1. SELECT
2. FROM
3. WHERE
4. GROUP BY
5. (HAVING)
6. ORDER BY

SELECT 쿼리의 실행 순서

1. FROM
2. WHERE
3. GROUP BY
4. (HAVING)
5. SELECT
6. ORDER BY

그룹 함수?

GROUP BY를 통해 그룹화된 각 그룹별로 값을 계산하여 결과를 출력하는 함수

```
select
  "proc_ymd", "userid",
  count("point") as "count",
  max("point") as "max",
  min("point") as "min",
  avg("point") as "avg",
  sum("point") as "sum"
from "learning_analytics"."e_point"
where 1=1
and "proc_ymd" = '20240304'
```

```
-- and "userid" = 'e629b763-9741-40c4-a85d-ba6f7aae4873'
group by "proc_ymd", "userid"
having max("point") >= 30 (여기서 "max"는 실행 순서상 아직 정의되지
```

ORDER BY

출력될 행들의 정렬 순서를 지정

- **ASC** – ASCENDING (오름차순)
- **DESC** – DESCENDING (내림차순)

```
select
"proc_ymd", "userid",
count("point") as "count",
max("point") as "max",
min("point") as "min",
avg("point") as "avg",
sum("point") as "sum"
from "learning_analytics"."e_point"
where 1=1
and "proc_ymd" = '20240304'
-- and "userid" = 'e629b763-9741-40c4-a85d-ba6f7aae4873'
group by "proc_ymd", "userid"
having max("point") >= 30
order by "count" asc, "sum" desc (여기서는 select절이 진행되어 별명
```

실습

초등 학습생의 월별 풀이이력 집계

초등 학습생의 상세 풀이 이력을 확인할 수 있는 테이블을 확인하세요.

하나의 행이 각 학생이 푼 문제의 답과 정답 여부를 의미합니다.

즉, 하나의 행이 하나의 문제풀이 이력이라고 할 수 있습니다.

초등 학습생의 월별 풀이이력 수를 집계하는 쿼리를

GROUP BY를 사용하여 작성해 주세요.

그리고, 월별 풀이이력 수가 3000만 건을 넘는 달을 확인하는
HAVING 절을 추가해 주세요.

```
select "proc_ym",  
count(*) as "count"  
from "learning_analytics"."e_assessment_detail"  
group by "proc_ym"  
having count(*) > 30000000
```

밀크티 중학 회원별 획득 포인트 확인

밀크티 중학 회원들의

포인트 획득(+) 혹은 사용(-) 내역을 확인할 수 있는 테이블이 있습니다.

2024년 5월달 중에

밀크티 중학 회원별 포인트 "획득(양수)" 의 합계를 확인하려고 합니다.

GROUP BY를 사용하여 유저별 포인트 합계 값을 확인하고,

ORDER BY를 사용하여 포인트 획득량이 많은 유저 순으로 나열하여
상위 100위 유저의 내역을 확인할 수 있도록 Query를 작성해 주세요.

```
select "userid",  
sum("point") as "sum"  
from "learning_analytics"."m_point"  
where 1=1  
and "proc_ym" = '202405'  
and "point" > 0  
group by "userid"  
order by "sum" desc;
```