

**Chapter 05.** 매출 관련 추가 분석

# 매출 관련 추가 분석 문제 모음

## 🎯 17. 2020년 7월 일별 매출의 전일 대비 증감폭, 증감률을 구해주세요

- with문 : 서브쿼리를 사용해서 임시테이블 처럼 사용할 수 있는 구문  
옵티마이저가 인라인 뷰나 임시테이블로 판단
- window 함수(행 순서 함수) 이해
  - lag: 이전 행을 가져옴
  - lead: 특정 위치의 행을 가져옴 (default는 1: 다음 행을 가져옴)

```
1 • with tbl_revenue as (  
2   select date_format(purchased_at - interval 9 hour, '%Y-%m-%d') as d_date  
3     , sum(price) as revenue  
4   from fastcampus.tbl_purchase  
5   where purchased_at >= '2020-07-01'  
6     and purchased_at < '2020-08-01'  
7   group by 1  
8   order by 1  
9 )  
10  
11 select *  
12   , revenue - lag(revenue) over(order by d_date asc) as diff_revenue  
13   , round((revenue - lag(revenue) over(order by d_date asc))/lag(revenue) over(order by d_date asc)*100,2) as chg_revenue  
14 from tbl_revenue
```

## 🎯 18. 일별로 많이 구매한 고객들한테 소정의 선물을 줄려고해요, 7월에 일별로 구매 금액 기준으로 가장 많이 지출한 고객 Top3를 뽑아주세요

- Rank 함수 이해
  - ex. rank() over (partition by c1, c2 order by price desc)
  - rank: 동일한 값이면 같은 순위를 매기고, 다음 값에서는 다음 순위를 매김
  - dense\_rank: 동일한 값이면 같은 순위를 매기고, 동일한 순위를 하나의 rank로 계산
  - row\_number: 중복 허용 없음. 동일한 순위에 대해서 고유의 순위 부여

```
1 • select *
2   from
3   (select date_format(purchased_at - interval 9 hour, '%Y-%m-%d') as d_date
4      , customer_id
5      , sum(price) as revenue
6      , dense_rank() over (partition by date_format(purchased_at - interval 9 hour, '%Y-%m-%d') order by sum(price) desc) as rank_rev
7   from fastcampus.tbl_purchase
8   where purchased_at >= '2020-07-01'
9   and purchased_at < '2020-08-01'
10  group by 1, 2) foo
11
12  where rank_rev < 4
```