

본 자료와 관련 영상 콘텐츠는 저작권법 제25조 2항에 의해 보호를 받습니다.

본 콘텐츠 및 콘텐츠 일부 문구 등을 외부에 공개하거나, 요약해서 게시하지 말아주세요.

Copyright [잔재미코딩](#) Dave Lee

### SQL 문제 1

고객의 지불 내역에서 각 고객별로 해당 고객의 지불 금액에 따른 순위를 출력하세요. (고객 ID, rental ID, 해당 고객의 지불 금액에 따른 순위를 출력하되, 동일한 값이 있을 경우 같은 순위를 부여하지만 다음 순위는 건너뛰지 않습니다)

### SQL 문제 2

영화 대여 내역에서 각 고객별로 해당 고객의 대여날짜시간 순으로 정렬하여, 고객 ID, rental ID 와 함께, 각 대여날짜시간, 바로 다음 대여날짜시간을 출력하세요.

### SQL 문제 3

영화 정보에서 각 등급별로 대여 기간(rental\_duration)이 가장 긴 영화의 제목을 출력하세요.

### SQL 문제 4

고객 정보에서 각 고객을 활동 상태(active)가 높은 순으로 정렬하고, 이를 기준으로 상위 1/3, 중간 1/3, 하위 1/3의 세 그룹으로 나누고, 각 그룹 내에서 고객의 순서를 customer\_id 가 낮은 순으로 정렬해서 다음과 같은 항목으로 출력하세요.

출력항목: customer\_id, first\_name, last\_name, active, active\_group, group\_row\_number(각 그룹 내에서 고객의 순서를 customer\_id 가 낮은 순으로 정렬하였을 때의 행 번호)

### SQL 문제 5

영화 대여 내역에서 고객별로 대여 순서(rental\_date 순으로 정렬), 이전 대여와의 간격(DAY 기반), 첫 번째 대여 일시(가장 rental\_date 가 오래된 일시)를 다음과 같은 항목으로 출력하세요.

출력항목: customer\_id, rental\_id, rental\_date, rental\_order, prev\_rental\_gap, first\_rental\_date

### SQL 문제 6

고객의 지불 내역에서 각 고객의 총 지불 금액에 따른 순위(지불금액이 높은 순, 동일한 값이 있을 경우 같은 순위를 부여하지만 다음 순위는 건너뛰지 않음)와 백분위 순위(지불금액이 높은 순)를 출력하세요.

### SQL 문제 7

영화 정보에서 각 등급별로 영화를 대여 기간에 따라 4개의 그룹으로 나누고, 각 그룹 내에서 rental\_duration 이 낮은 순으로 영화의 순서를 다음 항목으로 출력하세요.

film\_id, title, rating, rental\_duration, rental\_duration\_group, group\_row\_number

### SQL 문제 8

배우 정보에서 각 배우의 출연 영화 수에 따른 누적 분포를 다음과 같이 출력하세요.

actor\_id, first\_name, last\_name, film\_count, film\_count\_cume\_dist

### SQL 문제 9

Sakila 데이터베이스의 영화 대여 내역을 바탕으로 다음 항목들을 출력하는 SQL 쿼리를 작성하세요.

1. 고객별 대여 순위 (rental\_rank): 고객별로 대여 내역을 rental\_date가 빠른 순서대로 정렬한 후, 각 대여에 대한 순위를 매깁니다.
  2. 이전 대여와의 간격 (previous\_rental\_gap): 각 대여일을 기준으로 이전 대여일과의 간격을 일(DAY) 단위로 계산합니다. (단, 고객별로 계산)
  3. 다음 대여와의 간격 (next\_rental\_gap): 각 대여일을 기준으로 다음 대여일과의 간격을 일(DAY) 단위로 계산합니다. (단, 고객별로 계산)
  4. 고객별 첫 번째 및 마지막 대여 일자 (first\_rental\_date, last\_rental\_date): 고객별로 대여 내역을 rental\_date 기준으로 정렬한 후, 첫 번째 대여일과 마지막 대여일을 출력합니다.
  5. 고객별 대여 건의 백분위 순위 및 누적 분포 (rental\_percentile\_rank, rental\_cumulative\_dist): 고객별로 대여 내역을 rental\_date 기준으로 정렬한 후, 각 대여 건에 대해 백분위 순위와 누적 분포를 계산합니다.
  6. 고객별 대여 내역의 3개 그룹 분할 (rental\_group): 고객별 대여 내역을 최대한 균등하게 3개의 그룹으로 나눈 후, 각 대여 건이 속한 그룹 값을 출력합니다. (그룹 번호는 1 ~ 3)
  7. 고객과 그룹별 대여 순서 (group\_rental\_rank): 고객별로 대여 내역을 rental\_date 기준으로 정렬한 후, 각 그룹 내에서 대여 건에 대해 순차적인 순서를 매깁니다. (순서는 1부터 시작)
- 위의 항목들을 customer\_id, rental\_date 와 함께 모두 포함하여 출력하는 SQL 쿼리를 작성하세요.

최종 출력 항목: customer\_id, rental\_date, rental\_rank, previous\_rental\_gap,

next\_rental\_gap, first\_rental\_date, last\_rental\_date, rental\_percentile\_rank,  
rental\_cumulative\_dist, rental\_group, group\_rental\_rank

## SQL 문제 10

Sakila 데이터베이스의 결제 내역을 바탕으로 다음 항목들을 출력하는 SQL 쿼리를 작성하세요.

1. 고객별 누적 결제 금액 (cumulative\_amount): 고객별 결제 내역을 payment\_date 기준으로 정렬한 후, 누적 결제 금액을 계산합니다.
  2. 이전 결제 금액 차이 (prev\_payment\_diff): 고객별로 현재 결제 금액과 이전 결제 금액의 차이를 계산합니다.
  3. 다음 결제 금액 차이 (next\_payment\_diff): 고객별로 다음 결제 금액과 현재 결제 금액의 차이를 계산합니다.
  4. 고객별 첫 번째 및 마지막 결제 일자 (first\_payment\_date, last\_payment\_date): 고객별로 결제 내역을 payment\_date 기준으로 정렬한 후, 첫 번째 결제일과 마지막 결제일을 출력합니다.
  5. 전체 결제 금액에 따른 그룹 (total\_amount\_group): 누적 결제 금액을 기준으로 전체 결제 내역을 5개의 그룹으로 나누고, 각 결제가 속한 그룹을 출력합니다.
  6. 총 결제 금액 순위 (total\_amount\_rank): 누적 결제 금액을 기준으로 내림차순으로 정렬하여 순위를 매깁니다. 이때, 동일한 누적 결제 금액을 가진 고객이 있을 경우 동일 순위를 부여하고, 그 다음 순위는 건너 뛰지 않고, 연속된 순위로 부여합니다.
  7. 고객별 결제 금액의 백분위 순위 및 누적 분포 (payment\_amount\_pct\_rank, payment\_amount\_cume\_dist): 고객별 결제 금액을 기준으로 백분위 순위와 누적 분포를 계산합니다.
  8. 그룹 내 결제 건 순서 (group\_row\_number): 결제 금액 그룹별로 결제 내역을 누적 결제 금액 기준으로 내림차순 정렬한 후 순서를 매깁니다.
- 위의 항목들을 customer\_id, payment\_date 와 함께 모두 포함하여 출력하는 SQL 쿼리를 작성하세요.

최종 출력 항목: customer\_id, payment\_date, cumulative\_amount, prev\_payment\_diff, next\_payment\_diff, first\_payment\_date, last\_payment\_date, total\_amount\_rank, payment\_amount\_pct\_rank, payment\_amount\_cume\_dist, total\_amount\_group, group\_row\_number

본 자료와 관련 영상 콘텐츠는 저작권법 제25조 2항에 의해 보호를 받습니다.

본 콘텐츠 및 콘텐츠 일부 문구 등을 외부에 공개하거나, 요약해서 게시하지 말아주세요.

Copyright [잔재미코딩](#) Dave Lee