

문제 1의 실행 계획

1. 메뉴의 login 그리고 logout을 제외한 나머지 메뉴에 대해서,
각 메뉴 접근 시간 (LOG_TKTM)을 이용해 무슨 요일인지 알아냄
2. 이후 각 요일별로 접근한 메뉴를 묶고,
각 요일의 메뉴마다 몇번의 접근이 있었는지 카운트
3. 카운트된 횟수를 이용해 각 요일마다 메뉴 간 접근 횟수의 순위를 정함
4. 정해진 순위에서 Top 10까지 표현하고,
특정 요일이 Top 10 미만의 메뉴 순위를 보유할 경우 '-'로 표현

실행 계획 (1)

- 메뉴의 login 그리고 logout을 제외한 나머지 메뉴에 대해서,
각 메뉴 접근 시간 (LOG_TKTM)을 이용해 무슨 요일인지 알아냄
 - STR_TO_DATE을 이용하여 LOG_TKTM을 DATE로 변환 후 WEEKDAY 사용
- 해당 동작을 하는 SQL 코드는 WITH 절의 DAY_MENU_LOG 참고
- 아래는 DAY_MENU_LOG의 실행 결과 일부

DAYOFWEEK	LOG_ID	USR_NO	MENU_NM
4	id001	005	내카드
4	id004	001	세이프박스
4	id005	001	내카드
4	id006	001	이체내역
4	id007	001	세이프박스
4	id008	001	추천
4	id011	001	추천
4	id012	001	내정보
4	id016	001	추천

실행 계획 (2)

- 이후 각 요일별로 접근한 메뉴를 묶고,
각 요일의 메뉴마다 몇번의 접근이 있었는지 카운트
 - GROUP BY를 이용해 요일과 메뉴로 묶은 후 LOG_ID를 COUNT
(LOG_TKTM은 중복 발생 가능성 있음)
- 카운트된 횟수를 이용해 각 요일마다 메뉴 간 접근 횟수의 순위를 정함
 - COUNT된 LOG_ID와 ROW_NUMBER 함수를 이용해 순위를 정하는데,
접근건수 (내림차순) 그리고 메뉴명 (오름차순) 기준으로 정함
- 해당 동작을 하는 SQL 코드는 WITH
절의 RANK_DAY_AND_LOG참고
- 오른쪽은 RANK_DAY_AND_LOG의
실행 결과 일부

DAYOFWEEK	CONTENTS	DAYRANK
0	가이드(9)	1
0	세이프박스(8)	2
0	내신용정보(6)	3
0	내카드(6)	4
0	모임통장(6)	5
0	추천(6)	6
0	카드이용내역(6)	7
0	내정보(5)	8
0	이체내역(5)	9
0	정기예금(5)	10
1	이체내역(22)	1
1	추천(20)	2

실행 계획 (3)

- 정해진 순위에서 Top 10까지 표현하고,
특정 요일이 Top 10 미만의 메뉴 순위를 보유할 경우 ‘-’로 표현
 - Top 10 까지 표현하는 방법의 핵심은 OUTER JOIN에 있음
 - 1 부터 10까지 표현하는 테이블을 RECURSIVE WITH 절을 이용해 만듦
 - 특정 요일에 대한 메뉴 접근 순위 기록을
이전의 RANK_DAY_AND_LOG에서 찾은 뒤, OUTER JOIN을 수행
 - Top 10까지 못미치는 메뉴들은 모두 Null로 처리되고 이를 ‘-’로 치환



실행 계획 (4)

- 각 요일마다 COUNT 된 메뉴 접근 횟수 테이블들을 모두 LEFT JOIN 수행하여
요일별 메뉴 접근 횟수 테이블 생성

Top	Wednesday
1	세이프박스(12)
2	가이드(10)
3	추천(10)
4	정기예금(9)
5	내신용정보(7)
6	이체내역(7)
7	내정보(4)
8	내카드(4)
9	모임통장(4)
10	카드이용내역(3)



Top	Thursday
1	이체내역(5)
2	내카드(4)
3	추천(4)
4	내신용정보(3)
5	세이프박스(3)
6	정기예금(1)
7	카드이용내역(1)
8	-
9	-
10	-

Top	Wednesday	Thursday
1	세이프박스(12)	이체내역(5)
2	가이드(10)	내카드(4)
3	추천(10)	추천(4)
4	정기예금(9)	내신용정보(3)
5	내신용정보(7)	세이프박스(3)
6	이체내역(7)	정기예금(1)
7	내정보(4)	카드이용내역(1)
8	내카드(4)	-
9	모임통장(4)	-
10	카드이용내역(3)	-