#### 문제 1의 실행 계획

- 1. 메뉴의 login 그리고 logout을 제외한 나머지 메뉴에 대해서, 각 메뉴 접근 시간 (LOG\_TKTM)을 이용해 무슨 요일인지 알아냄
- 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

## 실행계획(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
+	+	   005   001   001   001   001	
4 4	id012	001	내정보
	id016	001	추천

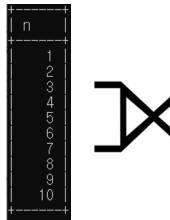
### 실행계획(2)

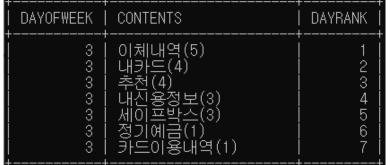
- 이후 각 요일별로 접근한 메뉴를 묶고,
  각 요일의 메뉴마다 몇번의 접근이 있었는지 카운트
  - GROUP BY를 이용해 요일과 메뉴로 묶은 후 LOG\_ID를 COUNT (LOG\_TKTM은 중복 발생 가능성 있음)
- 카운트된 횟수를 이용해 각 요일마다 메뉴 간 접근 횟수의 순위를 정함
  - COUNT된 LOG\_ID와 ROW\_NUMBER 함수를 이용해 순위를 정하는데, 접근건수 (내림차순) 그리고 메뉴명 (오름차순) 기준으로 정함
- 해당 동작을 하는 SQL 코드는 WITH
  절의 RANK\_DAY\_AND\_LOG참고
- 오른쪽은 RANK\_DAY\_AND\_LOG의
  실행 결과 일부

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

#### 실행계획(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 RECURVIE WITH 절을 이용해 만듬
  - 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







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	-

# 실행계획(4)

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

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



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

