



## Background

- ✓ SELECT
- ✓ FUNCTION
- ✓ ORDERING, GROUPING

## Goal

- ✓ 테이블에 저장된 데이터를 다양한 방법으로 SELECT 한다.
- ✓ 제공되는 주요 FUNCTION 을 이용한다.
- ✓ 정렬, 집계 데이터를 추출한다.

## Problem

지난 DB\_WS\_01 에서 생성한 테이블에 추가로 데이터를 등록하고 신규로 계좌 테이블을 생성한다. 제공되는 DB\_WS\_01.sql 파일을 실행하여 완성한다. 그리고 아래에 이어지는 실습을 수행한다. 각 작업을 수행하는 sql 문을 작성한다.

1. 고객번호 ( user\_seq ) 가 111 인 계좌 ( account ) 테이블을 조회한다.
2. 잔고 ( balance ) 가 5000 미만인 계좌 ( account ) 테이블을 조회한다.
3. 잔고 ( balance ) 가 5000 이상 10000 이하인 계좌 ( account ) 테이블을 조회한다.
4. 계좌번호 ( accountNumber ) 중 '4' 를 포함하는 계좌 ( account ) 테이블을 조회한다.
5. 고객명 ( name ) 이 '삼' 을 시작하는 고객 ( users ) 테이블을 조회한다.
6. 계좌 ( account ) 테이블에 있는 고객 번호 ( user\_seq ) 를 중복없이 조회한다.
7. 잔고 ( balance ) 가 없는 ( null ) 계좌 ( account ) 테이블을 조회한다.



8. 고객번호 ( user\_seq ) 가 있는 계좌 ( account ) 테이블을 조회한다.
9. 고객번호 ( user\_seq ) 가 있고, 잔고가 4000 이하인 계좌 ( account ) 테이블을 조회한다.
10. 계좌 ( account ) 테이블을 고객번호 ( user\_seq ) 기준으로 오름차순으로 정렬하여 조회한다.
11. 계좌 ( account ) 테이블을 고객번호 ( user\_seq ) 기준으로 오름차순으로, 그리고 그 안에서 잔고( balance ) 기준으로 내림차순으로 정렬하여 조회한다.
12. 계좌 ( account ) 테이블을 조회하되, balance 값이 없으면 0 으로 표시한다.
13. 고객 ( users ) 테이블을 조회하되, email은 @포함 뒷 부분은 빼고 앞 아이디만 표시한다. 컬럼명도 email 대신 email\_id 로 변경한다.
14. 고객 ( users ) 테이블을 조회하되, phone 의 '-' 를 제외하고 표시한다. 컬럼명도 phone\_short 로 변경한다.
15. 계좌 ( account ) 테이블에서 고객번호 ( user\_seq ) 가 222 인 건 수를 조회한다.
16. 계좌 ( account ) 테이블의 전체 잔고 ( balance ) 의 합을 조회하고 balance\_sum 으로 표시한다.
17. 계좌 ( account ) 테이블의 잔고 중 최소값과 최대값을 조회하고 각각 balance\_min, balance\_max 로 표시한다.
18. 계좌 ( account ) 테이블에서 고객번호와 고객번호 ( user\_seq ) 별 계좌 건수를 조회하고 user\_account\_cnt 로 표시한다. 단, 고객번호 ( user\_seq ) 가 없는 건은 제외한다.



19. 계좌 ( account ) 테이블에서 고객번호와 고객번호 ( user\_seq ) 별 잔고의 합을 조회하고 user\_balance\_sum 로 표시한다. 단, 고객번호 ( user\_seq ) 가 없는 건은 제외한다.

20. 위 19번의 결과 중 user\_balance\_sum 이 10000 이하인 건만을 조회한다.

### ❖ 제출방법

- 파일명 : " db02\_반\_성명.sql" 으로 작성 후, 제출