



Background

- ✓ SELECT
- ✓ FUNCTION
- ✓ ORDERING, GROUPING

Goal

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

Problem

1 – 10 번은 users 와 account 테이블을 이용하고, 11 – 15 번은 지난 과제에서 실행한 hr_cre.sql, hr_popul.sql 에 의해 만들어진 employees 등 테이블을 이용한다. 아래. 각 작업을 수행하는 sql 문을 작성한다.

1. 계좌 (account) 테이블을 조회하되, 고객명 (name) 도 함께 조회한다. (단, ANSI JOIN 을 사용)
2. 위 1번 query 를 table alias 를 사용하고, 오라클 Join 표기법을 사용한다.
3. 고객번호 (user_seq) 가 111 인 고객의 계좌 중 잔고 (balance) 가 2000 이상 계좌 (account) 테이블을 조회하되, 고객번호, 고객명, 계좌번호, 잔고 순으로 조회한다.
4. 계좌 (account) 테이블의 잔고 (balance) 가 5000 이상인 고객의 고객번호 (user_seq), 고객명 (name), 잔고 (balance) 를 조회한다. (단 Subquery 를 사용하지 않고 join 으로만 작성)
5. 위 4번 문제를 Subquery 를 이용해서 작성한다.



6. 고객 (users) 테이블에서 고객번호 (user_seq), 고객명 (name) 조회하되, 고객명 뒤에 계좌의 수도 account_cnt 로 함께 조회한다. (단, 계좌가 없는 고객은 제외한다.)
7. 위 6번 문제를 조회하되 계좌가 없는 고객도 함께 조회하고 계좌의 수에 0 으로 표시한다.
8. 각 계좌별 잔고 (balance) 가 전체 평균 잔고보다 적은 고객의 고객번호 (user_seq), 고객명 (name) 을 조회한다.
9. 고객의 잔고의 합이 전체 평균 잔고 + 5000 이하인 고객의 고객번호 (user_seq), 고객명 (name) 을 조회한다.
10. 계좌 (account) 테이블에서 balance 기준 내림차순으로 정렬하되, 상위 5 건만 조회한다.
11. salary 가 10000 이상인 사원의 employee_id, first_name, last_name, salary, job_title, department_name 을 조회한다.
12. job_id 가 IT_PROG 인 사원의 평균 salary 보다 salary 가 더 많은 사원의 총 수를 구한다.
13. 입사일자가 '97/06/25' 이후 입사한 직원 중 부서의 최소 salary 가 8000 이상고 부서의 최대 salary 가 20000 이하 인 부서에 해당하는 직원 전체를 조회한다.
14. manager_id 가 108 인 department 소속 직원들의 평균 salary 를 구하여, department_id, department_avg_salary 로 조회한다.



15. manager_id 가 없는 사원이 manager 로 등록되어 있는 사원의 employee_id, first_name, last_name, salary, job_title 조회한다.

❖ 제출방법

- 파일명 : " db03_반_성명.sql" 으로 작성 후, 제출