



Background

- ✓ SELECT
- ✓ FUNCTION
- ✓ ORDERING, GROUPING

Goal

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

Problem

제공되는 load_sample.sql 파일을 실행하고 테이블 생성 및 데이터를 확인한다. employees 테이블 기준으로 아래의 과제를 수행한다. 각 작업을 수행하는 sql 문을 작성한다.

1. job_id 가 'IT_PROG' 인 employees 테이블을 조회한다.
2. salary 가 5000 이상 인 employees 테이블을 조회한다..
3. salary 가 5000 이상 10000 이하 인 employees 테이블을 조회한다.
4. last_name 이 'M' 으로 시작하는 employees 테이블 중 employee_id, first_name, last_name, email 을 조회한다.
5. email 에 'AN' 을 포함하는 employees 테이블 중 employee_id, first_name, last_name, email 을 조회한다.
6. employees 테이블에 있는 job_id 를 중복없이 조회한다.



7. manager_id 가 없는 (null) employees 테이블을 조회한다.
8. commission_pct 가 있는 employees 테이블을 조회한다
9. job_id 가 'FI_ACCOUNT' 또는 'IT_PROG' 이고, salary 가 8000 이상 인 employees 테이블을 조회한다.
10. employees 테이블을 salary 기준으로 오름차순으로 정렬하여 조회한다.
11. employees 테이블을 job_id 기준으로 오름차순으로, 그리고 그 안에서 salary 기준으로 내림차순으로 정렬하여 조회한다.
12. employees 테이블에서 employee_id, email, salary, commission_pct을 조회하되, commission_pct 값이 없으면 0 으로 표시한다.
13. employees 테이블을 employee_id, first_name, last_name 을 조회하되, first_name 과 last_name 을 합쳐서 (중간에 공백문자 1개 포함) fullname 으로 조회한다.
14. employees 테이블을 employee_id, first_name, last_name 을 조회하되, last_name 의 길이가 5 이하인 건만 조회한다.
15. employees 테이블에서 salary 가 5000 이상인 건수를 조회하고 high_salary_cnt 로 표시한다.
16. employees 테이블에서 job_id 가 'IT_PROG' 인 조건에 해당하는 salary 의 전체 합을 구하고 it_prog_salary_sum 으로 표시한다.



17. employees 테이블에서 hire_date가 가장 빠른 값과 가장 늦은 값을 구하고 각각 hire_date_min, hire_date_max 로 표시한다.

18. employees 테이블에서 job_id 와 job_id 별 employee 건수를 조회하고 job_id_employee_cnt 로 표시한다.

19. employees 테이블에서 department_id별 salary 의 합을 조회하고 department_id_salary_sum 로 표시한다. 단, department_id 가 없는 건은 제외한다.

20. 위 19번의 결과 중 department_id_salary_sum 이 50000 이하인 건만을 조회한다.

❖ 제출방법

- 파일명 : "db_hw_01_반_성명.sql" 으로 작성 후, 제출