

파일명 : 이름_chap08_ex1.txt

Quiz 1. SQL 함수 중 INITCAP이란 함수가 있다. 이 함수는 매개변수로 전달한 문자열에서 앞 글자만 대문자로 변환하는 함수이다. INITCAP과 똑같이 동작하는 my_initcap 이란 이름으로 함수를 만들어보자. (단 여기서는 공백 한 글자로 단어 사이를 구분한다고 가정한다)

Quiz 2. 날짜형 SQL 함수 중에는 해당 월 마지막 일자를 반환하는 LAST_DAY란 함수가 있다.

매개변수로 문자형으로 일자를 받아, 해당 일자의 월 마지막 날짜를 문자형으로 반환하는 함수를 my_last_day란 이름으로 만들어 보자.

Quiz 3. 오른쪽의 프로시저는 이번 장에서 학습했던 my_new_job_proc 프로시저이다. 이 프로시저는 JOBS 테이블에 기존 데이터가 없으면 INSERT, 있으면 UPDATE를 수행하는데 IF문을 사용해 구현하였다. IF문을 제거하고 동일한 로직을 처리하도록 MERGE문을 사용해 my_new_job_proc2 란 프로시저를 생성해 보자.

```
CREATE OR REPLACE PROCEDURE my_new_job_proc
( p_job_id   IN JOBS.JOB_ID%TYPE,
  p_job_title IN JOBS.JOB_TITLE%TYPE,
  p_min_sal   IN JOBS.MIN_SALARY%TYPE,
  p_max_sal   IN JOBS.MAX_SALARY%TYPE )
IS
    vn_cnt NUMBER := 0;
BEGIN
    -- 동일한 job_id가 있는지 체크
    SELECT COUNT(*)
      INTO vn_cnt
    FROM JOBS
   WHERE job_id = p_job_id;

    -- 없으면 INSERT
    IF vn_cnt = 0 THEN
        INSERT INTO JOBS ( job_id, job_title, min_salary, max_salary, create_date, update_date)
          VALUES ( p_job_id, p_job_title, p_min_sal, p_max_sal, SYSDATE, SYSDATE);
    ELSE -- 있으면 UPDATE
        UPDATE JOBS
          SET job_title = p_job_title,
              min_salary = p_min_sal,
              max_salary = p_max_sal,
              update_date = SYSDATE
         WHERE job_id = p_job_id;
    END IF;
    COMMIT;
END ;
```

Quiz 4. 부서 테이블의 복사본 테이블을 다음과 같이 만들어보자.

```
CREATE TABLE ch09_departments AS  
SELECT DEPARTMENT_ID, DEPARTMENT_NAME, PARENT_ID  
FROM DEPARTMENTS;
```

위 테이블을 대상으로 다음과 같은 처리를 하는 프로시저를 my_dept_manage_proc 란 이름으로 만들어 보자.

(1) 매개변수 : 부서번호, 부서명, 상위부서번호, 동작 flag

(2) 동작 flag 매개변수 값은 'upsert' -> 데이터가 있으면 UPDATE, 아니면 INSERT

'delete' -> 해당 부서 삭제

(3) 삭제 시, 만약 해당 부서에 속한 사원이 존재하는지 사원테이블을 체크해 존재하면 경고메시지와 함께 delete를 하지 않는다.