

SQL Pogramming

- Day 24 -

2023. 04

목차

Day 1. 데이터베이스와 SQL
Day 2. 테이블 / 인덱스
Day 3. DDL / DML / DCL / TCL
Day 4. SELECT 기본문형 익히기1
Day 5. SELECT 기본문형 익히기2
Day 6. 서브쿼리 / 스칼라쿼리
Day 7. 뷰 / 인라인뷰
Day 8. 내장함수 일반
Day 9. 내장함수 CASE
Day 10. 조인 기본
Day 11. 조인 활용1
Day 12. 조인 활용2

Day 13. 데이터 압축하기1
Day 14. 데이터 압축하기2
Day 15. 데이터 늘리기1
Day 16. 데이터 늘리기2
Day 17. 인덱스 이해하기
Day 18. SELECT 중요성
Day 19. 분석함수1
Day 20. 분석함수2
Day 21. 분석함수3
Day 22. 실전연습1
Day 23. 실전연습2
Day 24. 프로시저 만들기1
Day 25. 프로시저 만들기2
Day 26. SQL 리뷰하기

Day 24. 프로시저만들기1

■ PROCEDURE(절차형 SQL)

- ▶ 일반적인 개발 언어처럼 SQL에도 절차 지향적인 프로그램이 가능하도록 DBMS 벤더별로 절차형 SQL을 제공하고 있음
- ▶ 오라클 → PL/SQL (Procedural Language/SQL)
- ▶ SQL SERVER → T-SQL
- ▶ DB2 → SQL/PL
- ▶ 일련의 쿼리를 마치 하나의 함수처럼 실행하기 위한 쿼리의 집합
- ▶ 데이터베이스 벤더별로 프로시저 작성 언어의 문법적인 특징들은 서로 다르지만 일반적인 개발자라면 단시간의 투자만으로도 습득이 가능함
- ▶ 한 번의 서버 콜에 의해 여러 SQL문을 실행할 수 있음
 - 프로시저로 작성하지 않고 각 프로그램 언어(C++, JAVA, C# 등)에서 로직을 구성하고 실행한다면 SQL을 한 번 실행할 때마다 서버 콜이 발생하여 프로시저로 작성하는 것 보다 훨씬 큰 부하가 발생할 수 있음
 - 예를 들어 FOR/WHILE문 등에 의해 SQL의 수많은 반복 수행이 발생한다고 가정하면 더욱 더 신중한 결정이 필요함

■ SQL vs. PROCEDURE

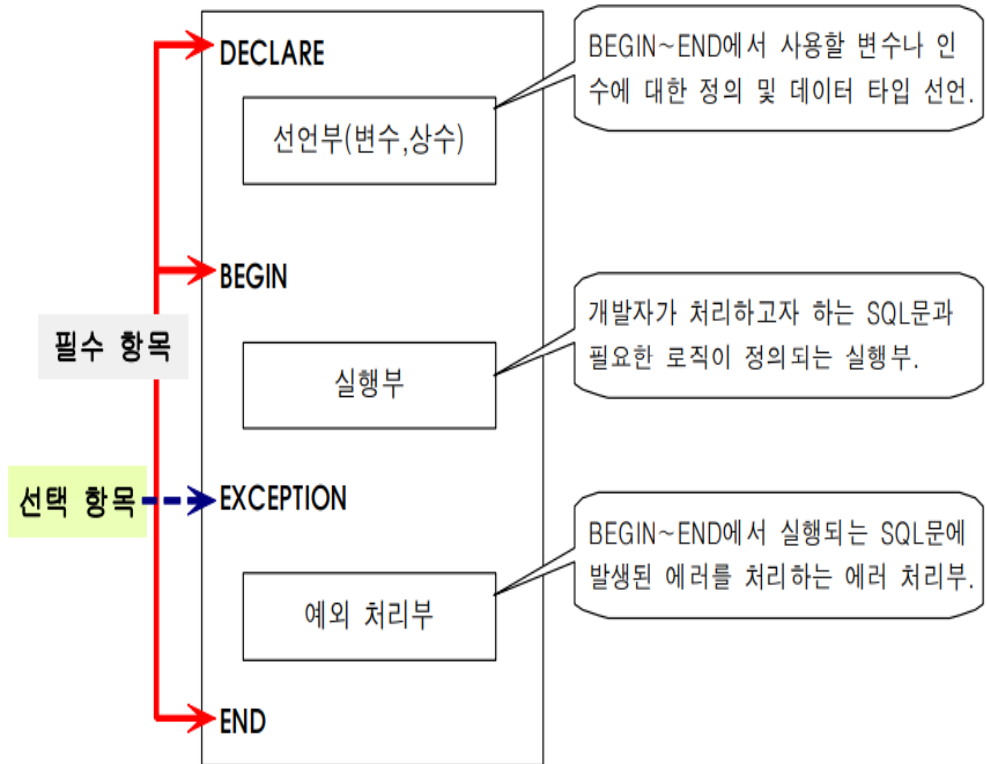
- ▶ 프로시저는 절차형 프로그램을 작성할 수 있는 언어이기 때문에 일반적인 프로그램 언어에서 구사할 수 있는 대부분의 로직을 동일하게 구사할 수 있음
- ▶ 이 말은 어떤 면에 있어서는 개발자에게 안심을 줄 수 있을지 모르겠지만, 또 다른 면에서는 SQL 개발 능력이 뛰어 나지 않아도 이러한 절차적 언어를 이용하여 간단한 SQL들을 조합하여 개발하는 방법으로 원하는 결과를 추출할 수도 있다는 의미이기 때문에 또 다른 위험성을 노출할 수 있다는 의미이기도 함
- ▶ 조금 복잡하더라도 하나의 SQL로 작성하는 것과 작성하기 쉬운 간단한 SQL들로 분리하여 반복문에 포함시켜 SQL들을 여러 번 실행시키는 것은 실행 퍼포먼스에 있어 하늘과 땅의 차이를 초래할 수 있다는 것을 이해해야 함
- ▶ 즉, 우리가 지금까지 배운 다양한 기법들을 적용하고, 참신한 아이디어가 가미된 잘 정제된 SQL을 만들어 내고, 프로시저에서 도움 받을 수 있는 기능들을 잘 혼합하여 멋진 프로그램을 만들어 내는 것을 목표로 삼아야 함

Day 24. 프로시저만들기1

■ PL/SQL 특징

- ▶ IF, LOOP 등의 절차적 언어를 사용하여 절차적인 프로그램이 가능하도록 함
- ▶ DBMS 정의 예러나 사용자 정의 예러를 정의하여 사용할 수 있음
- ▶ PL/SQL은 ORACLE에 내장되어 있으므로 ORACLE과 PL/SQL을 지원하는 어떤 서버로도 프로그램을 옮길 수 있음
- ▶ PL/SQL은 응용 프로그램의 성능을 향상시킴
- ▶ PL/SQL은 여러 SQL 문장을 하나의 블록으로 묶고 한 번에 블록 전부를 서버로 보내기 때문에 통신량을 줄일 수 있음

■ PL/SQL 블록 구조



Day 24. 프로시저만들기1

■ PL/SQL Basic Syntax

```
CREATE [OR REPLACE] Procedure [Procedure_name]
( argument1 [mode] data_type1,
  argument2 [mode] date_type2,
  ... .. )
IS [AS]
... ..
BEGIN
... ..
EXCEPTION
... ..
END;
/
```

■ PL/SQL PROCEDURE Example

```
CREATE OR REPLACE Procedure p_DEPT_insert
( v_DEPTNO in number,
  v_dname in varchar2,
  v_loc in varchar2,
  v_result out varchar2)
IS
cnt number := 0;
BEGIN
  SELECT COUNT(*) INTO CNT
  FROM DEPT
  WHERE DEPTNO = v_DEPTNO;
  if cnt > 0 then
    v_result := '이미 등록된 부서번호이다';
  else
    INSERT INTO DEPT (DEPTNO, DNAME, LOC)
    VALUES (v_DEPTNO, v_dname, v_loc);
    COMMIT;
    v_result := '입력 완료!!';
  end if;
EXCEPTION
  WHEN OTHERS THEN
    ROLLBACK;
    v_result := 'ERROR 발생';
END;
/
```

Day 24. 프로시저만들기1

실전문제① ▶ 결과창에 텍스트 출력하는 PROCEDURE 만들기

《테이블》	■	■	■
《조건》	■	▶	
《정렬》	■		
《특징》	■ 프로시저 명 → DAY23_1_SKP ■ 파라미터 → 없음 ■ 결과 창에 문자열을 출력하는 함수 이용 → 인터넷 검색		

결과 ▼ 총 건수 : 0건

DBMS Output (disabled)
 Messages | Data Grid | Trace | DBMS Output (disabled) | Query Viewer | Explain Plan | Script Output | PL/SQL Results

Frequency: 5

1 현재 시간은 2019-12-10 13:25:51 입니다.
 2

Day 24. 프로시저만들기1

실전문제② ▶ 테이블에 레코드 1건 입력하는 PROCEDURE 만들기

《테이블》	■ ZZ_TEST_SAMPLE	■	■
《조건》	■	▶	
《정렬》	■		
《특징》	■ 프로시저 명 → DAY23_2_SKP ■ 입력 파라미터 → P_USER_ID(VARCHAR2) ■ 출력 파라미터 → O_MSG(VARCHAR2) → 성공(OK), 실패(오류 메시지)		

결과 ▼ 총 건수 : 1건

ID	CONTENTS	REG_USER_ID	REG_DATETIME
3	현재 시간은 2020-01-28 17:07:59 입니다.	SKP	2020-01-28 오후 5:07:59

Thank you !

ASETEC Location <http://www.asetec.co.kr>

본사. 경기도 성남시 분당구 성남대로 331번길 8, 킨스타워 2201호 TEL.031.609.7000 FAX.031.609.7009
부산. 부산광역시 해운대구 센텀동로 99 TEL.051.506.6352 FAX.051.504.8794