**1. PL/SQL에서 SQL문장**

데이터베이스에서 정보를 추출할 필요가 있을 때 또는 데이터베이스로 변경된 내용을 적용할 필요가 있을 때 SQL을 사용합니다. PL/SQL은 SQL에 있는 DML과 TRANSACTION 제어 명령을 모두 지원합니다. 테이블의 행에서 질의된 값을 변수에 할당 시키기 위해 SELECT문장을 사용합니다. DML문장은 다중 행 처리를 할 수 있지만 SELECT문장은 하나의 행만을 처리할 수 있습니다.

**1.1 PL/SQL에서 SQL문장 사용**

1. SELECT 명령어를 사용하여 데이터베이스에서 한 행의 데이터를 추출합니다.
2. DML명령어를 사용하여 데이터베이스의 행에 대해 갱신할 수 있습니다.
3. COMMIT,ROLLBACK,SAVEPOINT명령어로 TRANSACTION을 제어 합니다.
4. 암시적인 커서로 DML결과를 결정합니다.

**1.1.1 SQL과 PL/SQL 문장의 유형 비교**

1. PL/SQL 블록은 TRANSACTION 단위가 아닙니다. COMMIT,ROLLBACK,SAVEPOINT는 블록과는 독립적이지만 블록에서 이 명령어를 사용할 수 있습니다.
2. PL/SQL은 CREATE TABLE, ALTER TABLE, DROP TABLE같은 DDL을 지원하지 않습니다.
3. PL/SQL은 GRANT,REVOKE과 같은 DCL을 지원하지 않습니다.

**1.2 PL/SQL에서 SELECT문장**

데이터베이스에서 데이터를 읽어 들이기 위해 SELECT문장을 사용합니다. SELECT문장은 INTO절이 필요한데, INTO절에는 데이터를 저장할 변수를 기술한다. SELECT 절에 있는 Column수와 INTO절에 있는 변수의 수는 좌측부터 1대1대응을 하며 개수와 데이터의 형, 길이가 일치하여야 한다. SELECT 문은 INTO절에 의해 하나의 행만을 저장할 수 있다. 그러므로 SELECT 문장에서 조건을 만족하는 ROW가 한 개도 없거나 여러 행이 있으면 에러를 발생한다.

**1.2.1 Syntax**

|  |
| --- |
| **SELECT** select\_list  **INTO** {variable\_name1[,variable\_name2,..] | record\_name}  **FROM** table\_name  **WHERE** condition; |

select\_list 열의 목록이며 행 함수, 그룹 함수, 표현식을 기술할 수 있다.

variable\_name 읽어 들인 값을 저장하기 위한 스칼라 변수

record\_name 읽어 들인 값을 저장하기 위한 PL/SQL RECORD 변수

Condition PL/SQL 변수와 상수를 포함하여 열명,표현식,상수,비교 연산자로

구성되며 오직 하나의 값을 RETURN할 수 있는 조건이어야 한다.

♣ 참고

질의는 하나의 행만 RETURN 해야 합니다. PL/SQL 블록 내의 SELECT 문장은 다음 규칙을 적용하는 Embedded SQL의 ANSI범주에 속합니다. 질의의 결과는 하나의 행만을 RETURN해야 하고 하나의 행 이상 또는 행이 없는 것은 에러를 생성합니다. PL/SQL은 NO\_DATA\_FOUND와 TOO\_MANY\_ROWS를 예외로 블록의 예외 섹션에서 추적할 수 있는 표준 예외를 조성하여 처리 합니다.

1. SELECT문 사용시 한개 이상의 ROW가 검색되면 Oracle8 Server는 미리 정해진 EXCEPTION인 TOO\_MANY\_ROWS라고 부르는 에러 번호 -1422를 발생한다.
2. SELECT문 사용시 아무런 ROW도 검색되지 않으면 Oracle8 Server는 미리 정해진 EXCEPTION인 NO\_DATA\_FOUND라고 부르는 에러 번호 +1403인 발생한다.

문제1) 이름을 입력받아 급여와 입사일을 출력하는 SCRIPT를 작성하여라.

|  |
| --- |
| SET VERIFY OFF  SET SERVEROUTPUT ON  ACCEPT p\_name PROMPT '사원의 이름을 입력하시오 : '  DECLARE  v\_name emp.ename%TYPE := UPPER('&p\_name');  v\_sal emp.sal%TYPE;  v\_hiredate emp.hiredate%TYPE;  BEGIN  SELECT sal,hiredate  INTO v\_sal,v\_hiredate  FROM emp  WHERE ename = v\_name;  DBMS\_OUTPUT.PUT\_LINE('급 여 : ' || LTRIM(TO\_CHAR(v\_sal,'$999,999')));  DBMS\_OUTPUT.PUT\_LINE('입사일 : ' || TO\_CHAR(v\_hiredate,'YYYY-MM-DD'));  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  DBMS\_OUTPUT.PUT\_LINE(v\_name || '는 자료가 없습니다.');  WHEN TOO\_MANY\_ROWS THEN  DBMS\_OUTPUT.PUT\_LINE(v\_name || '는 자료가 여러행이 존재합니다.');  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('기타 에러가 발생했습니다.');  END;  /  SET VERIFY ON  SET SERVEROUTPUT OFF |

♣ 참고

위 문제에서 SET VERIFY OFF은 old와 new값을 출력하지 않고, SET SERVEROUTPUT ON은 DBMS\_OUTPUT이라는 PACKAGE내의 PUT\_LINE함수를 사용한다. 이 함수를 SQL\*Plus에서 사용하려면 환경 변수를 사용하기 위하여 사용하였다.

문제2) 부서번호를 입력받아 급여의 합을 출력하는 SCRIPT를 작성하여라.

|  |
| --- |
| SET VERIFY OFF  SET SERVEROUTPUT ON  ACCEPT p\_deptno PROMPT '부서번호를 입력하시오(급여의 합을 구함) : '  DECLARE  v\_sal\_total NUMBER;  BEGIN  SELECT SUM(sal)  INTO v\_sal\_total  FROM emp  WHERE deptno = &p\_deptno;  DBMS\_OUTPUT.PUT\_LINE(&p\_deptno || '번 부서 급여의 합 : ' ||  LTRIM(TO\_CHAR(v\_sal\_total,'$99,999,999')));  END;  /  SET VERIFY ON  SET SERVEROUTPUT OFF |

☞ Guidelines

1. 세미콜론(;)으로 개별 SQL문장을 종료한다.
2. SELECT문장이 PL/SQL에 내장될 때 INTO절을 사용한다.
3. WHERE절은 선택적이며 입력변수, 상수, 리터럴, 또는 PL/SQL의 표현식을 지정하기 위해 사용될 수 있다.
4. SELECT절에서의 데이터베이스 열과 INTO절에서의 출력 변수의 수를 동일하게 좌측부터 1대1대응되게 지정해야 한다.
5. 식별자의 데이터형과 열의 데이터형을 갖도록 보증하기 위해 %TYPE 속성을 사용합니다. INTO절의 변수의 데이터형과 변수의 수는 SELECT에서 기술한 Column의 수와 일치하요야 합니다.
6. 그룹 함수는 테이블의 행 그룹에 적용되기 때문에 SUM같은 그룹 함수는 SQL에서 사용합니다.

**1.3 PL/SQL을 이용한 데이터 조작**

DML명령어를 사용하여 데이터베이스 테이블에 대한 내용을 변경할 수 있다.

1. INSERT문장은 테이블에 데이터의 새로운 행을 추가한다.
2. UPDATE문장은 테이블에 존재하는 행을 수정한다.
3. DELETE문장은 테이블에서 원치 않는 행을 제거한다.

**1.3.1 데이터 삽입**

1. USER와 SYSDATE같은 SQL함수를 사용합니다.
2. 데이터베이스 시퀀스를 사용하여 기본키 값을 생성합니다.
3. PL/SQL 블록에서 값을 얻거나 DEFAULT값을 이용합니다.

문제3) 초기값이 8000부터 9999까지 1씩 증가하는 SEQUENCE(EMPNO\_SEQUENCE)를 생성하여 EMP 테이블에 등록하는 SCRIPT를 작성하여라. 단 이름은 JONG, 업무는 MANAGER,부서번호는 10이다.

|  |
| --- |
| SQL> **CREATE SEQUENCE empno\_sequence**  2 **INCREMENT BY 1**  3 **START WITH 8000**  4  **MAXVALUE 9999**  5  **NOCYCLE**  6 **NOCACHE;**  Sequence created.  SQL> **DECLARE**  2  **v\_empno emp.empno%TYPE;**  3 **BEGIN**  4  **SELECT empno\_sequence.NEXTVAL**  5  **INTO v\_empno**  6 **FROM dual;**  7 **INSERT INTO emp(empno,ename,job,deptno)**  8 **VALUES (v\_empno,'JONG','MANAGER',10);**  9  **END;**  10 **/**  PL/SQL procedure successfully completed. |

**1.3.2 데이터 갱신**

1. 지정 연산자 좌측에 있는 식별자는 항상 데이터베이스 열이지만 오른쪽에 있는 식별자도 데이터베이스 열 또는 PL/SQL에서 사용되는 변수도 기술 가능하다.
2. PL/SQL에서의 SELECT문장과 달리 수정된 행이 없으면 에러가 발생하지 않는다.

문제4) 사원번호가 7369인 사원의 급여에 1000을 더하여 갱신하여라.

|  |
| --- |
| DECLARE  v\_sal emp.sal%TYPE := 1000;  BEGIN  UPDATE emp  SET sal = sal + v\_sal  WHERE empno = 7369;  END;  / |

**1.3.3 데이터 삭제**

PL/SQL에서 SQL의 DELETE문장을 사용하여 필요 없는 자료를 삭제할 수 있다.

문제5) 사원번호가 7654인 사원의 정보를 삭제하여라.

|  |
| --- |
| DECLARE  v\_empno emp.empno%TYPE := 7654;  BEGIN  DELETE emp  WHERE empno = v\_empno;  END;  / |

**1.4 이름 지정 규약**

1. WHERE절에서 모호성을 피하기 위해 이름 지정 규약을 사용한다.
2. 데이터베이스 열과 식별자는 다른 이름을 가져야 한다.
3. PL/SQL이 테이블의 열을 첫번째로 조사하기 때문에 구문 오류가 발생할 수도 있다.

**1.5 COMMIT과 ROLLBACK 문장**

COMMIT또는 ROLLBACK SQL문장으로 트랜잭션 논리를 제어 함으로써 데이터베이스를 영구적으로 변경하게 합니다. ORACLE SERVER에서와 마찬가지로 DML 트랜잭션은 COMMIT또는 ROLLBACK을 수행한 다음에 시작하고 성공적인 COMMIT또는 ROLLBACK 다음에 종료합니다.

|  |
| --- |
| DECLARE  v\_empno emp.empno%TYPE := 7934;  BEGIN  DELETE emp  WHERE empno = v\_empno;  COMMIT;  END;  / |

**1.6 SQL CURSOR**

SQL문장을 실행할 때마다 ORACLE SERVER은 명령이 분석되고 실행되는 곳에서 메모리 영역을 개방합니다. 이 영역을 CURSOR라 합니다. 블록의 실행 부분이 SQL문장을 실행할 때 PL/SQL은 SQL식별자를 가지는 암시적 CURSOR를 생성합니다. PL/SQL은 자동적으로 이 CURSOR를 관리합니다. 명시적 CURSOR는 명시적으로 선언되고 프로그래머에 의해 명명됩니다.

1. CURSOR는 개별 SQL 작업 영역입니다.
2. CURSOR에는 임시적 커서와 명시적 커서가 있습니다.
3. ORACLE SERVER은 SQL문장을 분석하고 실행하기 위해 암시적 커서를 사용합니다.
4. 명시적 커서는 프로그램에 의해 명시적으로 선언 됩니다.

**1.6.1 CURSOR의 속성**

SQL CURSOR의 속성을 사용하여 SQL문장의 결과를 테스트할 수 있다.

|  |  |
| --- | --- |
| 종 류 | 설 명 |
| SQL%ROWCOUNT | 가장 최근의 SQL문장에 의해 영향을 받은 행의 수 |
| SQL%FOUND | 가장 최근의 SQL문장이 하나 또는 그 이상의 행에 영향을 미친다면 TRUE로 평가한다. |
| SQL%NOTFOUND | 가장 최근의 SQL문장이 어떤 행에도 영향을 미치지 않았다면 TRUE로 평가한다. |
| SQL%ISOPEN | PL/SQL이 실행된 후에 즉시 암시적 커서를 닫기 때문에 항상 FALSE로 평가된다. |

문제6) ITEM 테이블에서 ORDID가 605인 자료를 모두 삭제하여라.

|  |
| --- |
| VARIABLE rows\_deleted VARCHAR2(60)  DECLARE  v\_ordid NUMBER := 605;  BEGIN  DELETE FROM item  WHERE ordid = v\_ordid;  IF SQL%FOUND THEN  :rows\_deleted := SQL%ROWCOUNT || ' rows deleted.';  ELSE  :rows\_deleted := '삭제한 자료가 없습니다.';  END IF;  END;  /  PRINT rows\_deleted |

**◈ 연 습 문 제 ◈**

1. EMP 테이블에서 급여가 최고인 사원의 이름,급여를 출력하는 SCRIPT를 생성하여라. 단 SQL\*Plus변수를 이용하여 화면에 출력한다.

2. DEPT 테이블에 등록하는 SCRIPT를 작성하여라. 단 아래의 표를 참고하여라.

|  |
| --- |
| 부서번호를 입력하시오. : 50  부서명을 입력하시오. : 현대교육센터  근무지를 입력하시오. : 마북리  PL/SQL procedure successfully completed. |

3. DEPT 테이블에 등록되어 있는 자료 중 근무지를 변경하는 SCRIPT를 작성하여라. 단 아래의 표를 참고하여라

|  |
| --- |
| 부서번호를 입력하시오. : 50  근무지를 입력하시오. : 강남  PL/SQL procedure successfully completed. |

4. EMP 테이블에서 사원번호를 입력받아 자료를 삭제하는 SCRIPT를 작성하여라.

5. EMP 테이블에서 하나의 자료를 입력할 수 있는 SCRIPT를 작성하여라. 단 사원번호는 SEQUENCE를 생성하여 입력한다.

6. SQL%ROWCOUNT란 무엇을 의미하는가.

7. SQL CURSOR란 ?