**1. PL/SQL 개요**

PL/SQL(Procedural Language/SQL)은 최근의 프로그래밍 언어의 특성을 수용한, SQL의 확장이라 할 수 있다. SQL의 데이터 조작(DML)과 질의문(QUERY)을 블록 구조에 절차적 단위(IF, LOOP, FOR등)로 된 코드를 포함할 수 있으며 절차적 프로그래밍을 가능하게 한 강력한 TRANSACTION 처리 언어이다.

|  |
| --- |
| SET VERIFY OFF  SET SERVEROUTPUT ON  ACCEPT p\_name PROMPT ' 이 름: '  DECLARE  v\_empno emp.empno%TYPE;  v\_name emp.ename%TYPE := UPPER('&p\_name');  v\_sal emp.sal%TYPE;  v\_job emp.job%TYPE;  BEGIN  SELECT empno,job  INTO v\_empno,v\_job  FROM emp  WHERE ename = v\_name;  IF v\_job IN ('MANAGER','ANALYST') THEN  v\_sal := v\_sal \* 1.5;  ELSE  v\_sal := v\_sal \* 1.2;  END IF;  UPDATE emp  SET sal = v\_sal  WHERE empno = v\_empno;  IF SQL%FOUND THEN  DBMS\_OUTPUT.PUT\_LINE(SQL%ROWCOUNT||'개의 행이 갱신되었습니다.');  ELSE  DBMS\_OUTPUT.PUT\_LINE('갱신된 자료가 없습니다.');  END IF;  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 동명 이인입니다.');  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('기타 에러가 발생 했습니다.');  END;  /  SET VERIFY ON  SET SERVEROUTPUT OFF |

**1.1 PL/SQL의 장점**

PL/SQL은 SQL로는 얻을 수 없는 절차적 언어의 기능을 가지고 있다.

**1.1.1 프로그램 개발의 모듈화**

1. 블록 내에서 논리적으로 관련된 문장들의 그룹화할 수 있다.
2. 강력한 프로그램을 작성하기 위해 서브 블록들을 큰 블록에 포함할 수 있다.
3. 복잡한 문제에 대한 프로그래밍이 적절히 나뉘어진 모듈들의 집합으로 구성된다.

**1.1.2 식별자 선언**

1. 변수, 상수 등을 선언하고 SQL과 절차적인 프로그램에서 사용한다.
2. 데이터베이스의 테이블과 Record를 기반으로 하는 dynamic한 변수 선언이 가능하다.
3. 단일형 데이터 타입과 복합형 데이터 타입을 선언할 수 있다.

**1.1.3 절차적 언어 구조로 된 프로그램 작성**

1. IF문은 조건에 따라 일련의 문장을 실행한다.
2. LOOP문을 사용하여 일련의 문장을 반복적으로 실행한다.
3. Explicit Cursor를 이용한 Multi-row 질의 처리한다.

**1.1.4 ERROR 처리**

1. Exception 처리 루틴을 이용하여 Oracle8 Server 에러를 처리한다.
2. 사용자 정의 에러를 선언하고 Exception 처리 루틴으로 처리 가능하다.

**1.1.5 성능 향상**

1. PL/SQL은 응용 프로그램의 성능을 향상 시킬 수 있다.
2. PL/SQL은 여러 SQL문장을 BLOCK으로 묶고 한번에 BLOCK전부를 서버로 전송하기 때문에 통신량을 줄일 수 있다.

ORACLE

with PL/SQL

ORACLE

SERVER

APPLICATION

S

Q

L

S

Q

L

PL/SQL BOLCK

S

Q

L

APPLICATION

**1.1.6 PL/SQL의 처리**

PL/SQL로 작성된 Block을 Oracle Server로 보내면 그 안에 있는 PL/SQL Engine이 SQL문과 Non SQL문을 구분하여 Non SQL문은 PL/SQL Engine내의 Procedural statement executor가 수행하고 SQL문은 SQL statement executor가 처리하게 된다. 즉 Non SQL문은 Client환경에서 처리되고 SQL문은 서버에서 실행하게 된다. 따라서 PL/SQL을 사용하게 되면 서버쪽으로 작업의 양을 줄이게 되므로 Network traffic를 감소시켜 Performance를 증가시키는 이점이 있다.

**1.2 PL/SQL Block 구조**

PL/SQL은 프로그램을 논리적인 블록으로 나누게 하는 구조화된 블록 언어이다. PL/SQL 블록은 선언부(선택적),실행부(필수적),예외 처리부(선택적)로 구성되고 BEGIN과 END는 키워드로 반드시 기술하여야 한다. PL/SQL 블록에서 사용하는 변수는 블록에 대해 논리적으로 선언할 수 있고 사용할 수 있다. 변수들과 그 외의 식별자를 참조하고 선언함으로써 PL/SQL 블록 내에서 값을 저장하고 변경할 수 있다.

|  |
| --- |
| * DECLARE * variables, cursor, user\_defined, exception * BEGIN * SQL,PL/SQL statements; * EXCEPTION * actions to perform when errors occur * END; |

기본적인 PL/SQL Block은 세 부분으로 구성된다.

|  |  |  |
| --- | --- | --- |
| 섹 션 | 설 명 | 포 함 |
| 선언부 | 실행부에서 참조할 모든 변수,상수,CURSOR,EXCEPTION을 선언 | 선택 |
| 실행부 | 데이터베이스의 데이터를 처리할 SQL문과 PL/SQL 블록을 기술 | 필수 |
| 에러 처리부 | 실행부에서 에러가 발생했을 때 수행될 문장을 기술 | 선택 |

☞ Guidelines

1. PL/SQL Block내에서는 한 문장이 종료할 때마다 세미콜론(;)을 기술한다.
2. END뒤에 세미콜론(;)을 사용하여 하나의 Block이 끝났다는 것을 명시한다.
3. PL/SQL Block의 작성은 편집기를 통해 파일로 작성할 수 있고 SQL\*Plus에서 바로 작성할 수 있다.
4. SQL Buffer에서 PL/SQL을 실행하기 위해 “/”을 사용하며 성공적으로 실행 된다면 PL/SQL procedure successfully completed라는 Message가 출력된다.

**1.2.1 PL/SQL 블록의 유형**

PL/SQL의 모든 단위는 하나 또는 그 이상의 블록을 포함합니다. 이 블록은 다른 것으로부터 하나로 완전히 분리되어 중첩될 수 있습니다. 기본 단위(프로시저, 함수,서브 프로그램, 익명의 블록)는 임의의 수의 중첩된 서브 블록을 포함할 수 있는 논리적인 블록으로 구성된다. 그러므로 하나의 블록은 다른 블록의 작은 부분이 되기도 하고 또는 코드 단위의 전체중 일부가 될 수도 있습니다.

|  |  |  |
| --- | --- | --- |
| Anonymous | Procedure | Function |
| [DECLARE]  BEGIN  statements;  statements;  statements;  [EXCEPTION]  END; | CREATE PROCEDURE name  IS  BEGIN  statements;  statements;  [EXCEPTION]  END; | CREATE FUNCTION name  RETURN datatype  IS  BEGIN  statements;  statements;  RETURN value;  [EXCEPTION]  END; |

**가) Anonymous Block(익명 블록)**

이름이 없는 블록을 의미한다. 그것은 실행하기 위해 프로그램 안에서 선언되고 실행 시에 실행을 위해 PL/SQL 엔진으로 전달됩니다. 선행 컴파일러 프로그램과 SQL\*Plus 또는 서버 관리자에서 익명의 블록을 내장할 수 있습니다.

**나) Subprogram(procedure, function)**

Subprogram은 매개변수를 사용할 수 있고 호출할 수 있는 PL/SQL BOLCK이다. Procedure 또는 Function로 선언될 수 있습니다. 일반적으로 어떤 작업을 수행하기 위해 Procedure을 사용하고 값을 계산하기 위해 Function을 사용합니다. 서버 또는 Application 수준에서 Subprogram을 저장할 수 있습니다. Developer/2000을 사용하여 Application의 일부로써 Procedure와 Function을 선언할 수 있고 필요할 때마다 Trigger를 이용하여 사용할 수 있다.

**1.2.2 프로그램의 구성**

|  |  |  |
| --- | --- | --- |
| 프로그램 구성 | 설 명 | 사용 환경 |
| 익명 블록 | Application에 내장되거나 대화식으로 호출 | 모든 PL/SQL환경 |
| 내장된 Procedure or Function | 매개변수를 받아들일 수 있고 이름을 이용하여 반복적으로 호출할 수 있는 이름이 있는 PL/SQL 블록은 Oracle 서버에 저장된다. | Oracle Server |
| Application Procedure or  Function | 매개변수를 받아들일 수 있고 이름을 이용하여 반복적으로 호출할 수 있는 이름이 있는 PL/SQL 블록은 Developer/2000 어플리케이션에 저장되거나 Shared Library에 저장 된다. | Devleoper/2000 |
| Package | 관련된 Procedure or Function을 묶어 이름을 붙인 PL/SQL모듈입니다. | Oracle Server와  Devleoper/2000 |
| Database Trigger | Database Table와 관련된 DML명령문에 의해 Trigger될 때 자동적으로 실행됩니다. | Oracle Server |
| Application Trigger | PL/SQL 블록은 Application Event와 관련되고 자동적으로 실행됩니다. | Devleoper/2000 |

**1.2.3 PL/SQL 환경**

PL/SQL은 별개의 Oracle 제품이 아니라 Oracle8 서버와 다른 Oracle TOOL에 이용되고 있는 프로그래밍 언어이다. PL/SQL 의 블록은 Oracle8 서버나 툴에 내장되는 PL/SQL 엔진에 전달되어 처리된다. 사용하는 엔진은 PL/SQL 이 수행되는 곳에 따라 다르다.

**가) Oracle8 서버에서의 PL/SQL 엔진**

Pro\*프로그램, USER-EXIT, SQL\*Plus, 또는 Server Manager에서 PL/SQL 블록을 사용하면 Oracle8 서버의 PL/SQL 엔진이 처리한다. 그리고 블록에 있는 SQL을 별도의 문장으로 분리하여 SQL 문 실행기로 보낸다. 이는 응용 프로그램의 블록을 한 번에 Oracle8 서버에게 보낸다는 뜻이며 따라서 client/server 환경하에서 많은 성능 향상을 기대할 수 있다.

**나) Oracle 툴에서의 PL/SQL**

Developer/2000을 포함한 많은 Oracle툴은 Oracle7 서버에 있는 엔진과는 별도로 자체 PL/SQL 엔진을 갖고 있다. 이 엔진이 SQL 분장을 찾아서 Oracle7 서버의 SQL 문 실행기로 보내고, PL/SQL engine 은(데이터베이스라기보다는 이미 클라이언트 환경에 있는) 응용 프로그램에 대해 지역적인 데이터를 처리한다. 이로써 Oracle7 서버의 대한 작업량과 요구되는 메모리 커서의 수를 줄인다.

♣ 참고

Developer/2000 응용 프로그램의 부분으로 선언된 프로시저와 함수의 일반적인 구조는 동일하더라도 데이터베이스에 저장된 것과는 다르다. Stored Subprogram 은 데이터베이스 객체이고 데이터 사전에 저장되며 여러 응용 프로그램이 사용할 수 있다. Application subprogram 그 응용 프로그램의 지역적인 PL/SQL 엔진에 블록을 전달한다. 작업은 서버 쪽이 아닌 응용 프로그램 쪽에서 수행된다.

**1.3 SQL\*Plus로 하는 일**

1. SQL 명령과 PL/SQL블록의 입력,편집,저장,검색 및 실행을 해준다.
2. 데이터 베이스의 데이터 access를 가능하게 한다.
3. 계산수행, query 결과를 보고서 양식으로 출력한다.
4. SQL 데이터베이스 언어와 절차적 언어의 확장인 PL/SQL을 SQL\*Plus 프로그램을 이용하여 사용할 수 있다.
5. SQL\*Plus는 SQL명령과 PL/SQL 블럭을 조작하는 등 많은 부가적인 작업을 수행할 수 있게 해 준다.

**◈ 연 습 문 제 ◈**

1. PL/SQL의 특징을 설명하여라.

2. PL/SQL에서 사용하는 BLOCK란

3. PL/SQL의 3가지 유형을 설명하여라.

4. SQL\*Plus와 PL/SQL의 관계를 설명하여라