

<p>1. PL/SQL문은 프로그래밍 언어처럼 변수에 값을 할당하고 예외처리도 할 수 있으며 특정 기능을 처리하는 함수나 프로시저를 만들 수 있는 기능을 제공한다. PL/SQL문은 모든 코드가 DB 내부에서 만들어져 처리됨으로써 수행 속도와 성능측면에서 큰 장점이 있다.</p>
<p>2. PL/SQL문 구조</p> <p>가. 기본단위를 블록이라고 하는데, 블록은 선언부,실행부,예외처리부로 구성된다.</p> <p>나. 이 블록은 다시 이름이 없는 블록과 이름이 있는 블록으로 구분할 수 있는데,전자에 속하는 것이 익명블록이라 한다. 블록구조는 다음과 같다.</p> <p>이름부 --블록의 명칭이 온다.생략하면 익명블록이다.</p> <p>IS</p> <p>선언부 --변수,상수등을 선언</p> <p>BEGIN</p> <p>실행부 --실제 로직</p> <p>EXCEPTION</p> <p>예외 처리부</p> <p>END;</p>
<p>3. PL/SQL 문법</p> <p>가. 변수에 값을 할당할 때 일반 언어에 사용하는 대입연산자 = 대신 :=을 사용한다.</p> <p>나. 변수선언 규칙</p> <p>변수명 데이터타입 := 초기값;</p> <p>다. 변수에 초기값을 할당하지 않으면 타입에 상관없이 변수 초기값으로 NULL이 대입된다. 선언부에서 선언만하고 실행부에서 초기값을 할당하기도 한다.</p> <p>라. 변수 선언할 때 사용하는 데이터 타입으로 하나로는 SQL 데이터 타입이고,다른 하나로는 PL/SQL 데이터 타입이다. SQL타입은 PL/SQL타입에 포함이 되어 있다.</p> <p>PL/SQL타입으로 BOOLEAN, PLS_INTEGER, BINARY_INTEGER등이 있다.</p> <p>BOOLEAN타입 값으로 TRUE,FALSE,NULL 3가지를 가진다.</p> <p>PLS_INTEGER 범위는 -2,147,483,648~2,147,483,647까지 범위를 가진 숫자 타입으로 number타입에 비해 내부적으로 저장 공간을 덜 차지한다.</p> <p>BINARY_INTEGER타입은 PLS_INTEGER타입과 동일하다.</p> <p>마. 상수는 변수와는 달리 한번 값을 할당하면 변하지 않는다.</p> <p>상수명 CONSTANT 데이터타입 := 상수값;</p> <p>상수는 선언할 때 반드시 초기화를 해야 한다.</p> <p>바.문자열 연결 연산자로 을 사용한다. ** 제곱연산자 이다. 2**3은 2의 3제곱을 뜻한다. 같지 않다 연산으로 <>, !=, ~= , ^= 4가지 모두 사용한다.</p>

사. PL/SQL과 SQL문 주석기호로 함께 사용한다. 한줄 전체를 주석 처리할때는 맨 앞에 --를 명시하고, 여러 줄 코드에 주석문을 붙일 때는 처음과 끝에 /* */를 붙인다. 변수명 테이블명.컬럼명%TYPE;을 사용하면 해당 변수에 컬럼 타입을 자동으로 가져온다. %TYPE 키워드를 사용하면, 일일이 변수 타입을 찾는 수고가 제거되고, 데이터 타입을 잘못 선언할 위험도 없앨 수 있으므로 매우 유용한 기능 중 하나이다.

PL/SQL 제어문

1. 제어문에는 특정 조건에 맞을 때 처리하는 조건문, 반복 처리 시 사용하는 반복문, GOTO등과 같은 순차적 제어문이 있다.

2. 특정조건에 따라 처리를 하는 것을 조건문이라 하는데 그 대표적인 것이 바로 IF 문이며 그 구조는 다음과 같다.

<조건이 1개일 경우>

```
IF 조건 THEN
    조건 처리;
END IF;
```

<조건이 2개일 경우>

```
IF 조건 THEN
    조건 처리 1;
ELSE
    조건 처리 2;
END IF;
```

<조건이 n개인 경우>

```
IF 조건1 THEN
    조건 처리1;
ELSIF 조건2 THEN
    조건 처리2;
...
ELSE
    조건 처리n;
END IF;
```

CASE 문

<유형1>

CASE 표현식

```
WHEN 결과1 THEN
    처리문1;
WHEN 결과2 THEN
    처리문2;
```

```

...
ELSE
    기타 처리문;
END CASE;

<유형2>
CASE WHEN 표현식1 THEN
    처리문1;
    WHEN 표현식2 THEN
        처리문2;
    ...
    ELSE
        기타 처리문;
END CASE;

```

LOOP 문

1. LOOP문은 루프를 돌며 반복해서 로직을 처리하는 반복문이다.

2. 형식)

```

LOOP
    처리문;
    EXIT [WHEN 조건];
END LOOP;

```

이 반복문은 특정 조건을 만족할 때만 루프를 돌며 로직을 처리해야 하는데, LOOP문은 특정 조건이란 것이 없다. 그 대신 '이런 경우에는 루프를 빠져 나가라'라는 EXIT를 사용해 루프를 종료한다. 따라서 EXIT를 누락하면 무한루프에 빠지게 되니 사용할 때 주의해야 한다. 일반적으로 LOOP문에서는 IF문을 사용해 루프를 빠져나가는 조건을 걸기도 하지만, EXIT WHEN 조건을 사용해 WHEN절에서 명시한 조건에 맞을 때 빠져 나가라고 명시할 수 있다.

while 반복문

1. 형식)

```

while 조건
loop
    처리문;
end loop;

```

2. while문은 loop문과 비슷하지만 while 다음에 조건을 붙여 해당 조건에 만족할 때만 루프를 돌면서 로직을 처리한다.

3. while 문에서도 반복문(루프)을 돌다가 특정 조건에 부합하면 exit를 써서 반복문을 빠져나올 수 있다.

FOR 문

1. FOR문 기본 유형

FOR 인덱스 IN [REVERSE] 초기값 .. 최종값

LOOP

처리문;

END LOOP;

2. 인덱스는 초기값에서 시작해 최종값까지 루프를 돌며 1씩 증가되는데, 인덱스는 참조는 가능하지만 변경할 수는 없고 참조도 오직 반복문 루프안에서만 가능하다.

3. REVERSE를 명시하면 순서가 거꾸로 된다.

CONTINUE 문

1. 이 문은 반복문은 아니지만, 반복문 내에서 특정 조건에 부합할 때 처리 로직을 건너뛰고 상단의 루프 조건으로 건너가 루프를 계속 수행할 때 사용한다.

2. exit는 루프를 빠져나올 때 사용하고, continue문은 제어 범위가 조건절로 넘어간다. 참고로 continue문은 오라클 11g 버전부터 지원하고 있다.

GOTO 문

GOTO문을 만나면 GOTO문이 지정하는 라벨로 제어가 넘어간다.

NULL 문

1. PL/SQL에서는 NULL문을 사용할 수 있다. NULL문은 아무것도 처리하지 않는 문장이다. 이 문은 주로 IF,CASE문을 작성할 때 주로 사용한다. 조건에 따라 처리 로직을 작성하고 앞에서 작성한 모든 조건에 부합되지 않을 때, 즉 ELSE 절을 수행할 때 아무것도 처리하지 않고 싶은 경우 null문을 사용한다.