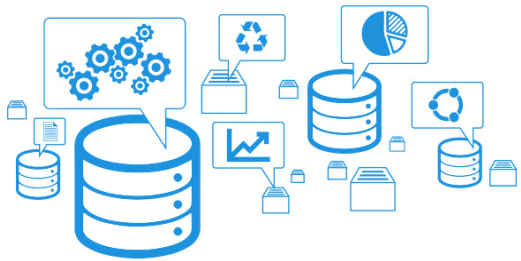
Universidad de Playa Ancha

Ingeniería en Informática

Sistemas de Bases de Datos

PL/SQL 01
Introducción

Docente Tatiana Ilabaca W.
Segundo semestre de 2023



Objetivos

Conocer las características básicas del lenguaje de programación PL/SQL, el lenguaje de las bases de datos Oracle

Implementar bloques anónimos

¿Qué es PL/SQL?

- Es un sofisticado lenguaje de programación que se utiliza para acceder a bases de datos Oracle
- Es un lenguaje particular del sistema gestor de bases de datos Oracle y no un lenguaje estándar
- El código puede ser procesado de forma rápida y eficiente.
- Puede combinar código (PL) con sentencias SQL

¿Qué es PL/SQL?

- Potencia el lenguaje SQL, agregando estructuras y objetos, tales como:
  - Bloques
  - Manejo de errores y excepciones
  - Creación de procedimientos y funciones
  - Definición de variables y tipo
  - Estructuras de bucle
  - Cursores
  - etc.

- Un programa PL/SQL se estructura utilizando bloques.

Bloque

- Unidad básica de todo programa PL/SQL.
- Un programa, por lo menos, debe tener uno.
- Se compone de:
  - Una sección declarativa optativa
  - Una sección de ejecución obligatoria
  - Una sección de control de errores optativa
- Tipos
  - Bloques anónimos
  - Bloques nominados
  - Subprogramas
  - Disparadores (Triggers)

Bloque - tipos

- Bloques anónimos. Se construyen de manera dinámica y se ejecutan una sola vez.
- Bloques nominados. Bloques anónimos que se identifican por un nombre.
- Subprogramas. Son bloques nominados que se almacenan en la base de datos. Se ejecutan bajo demanda. Comprende procedimientos, funciones y paquetes.
- Disparadores. Son bloques nominados que se almacenan en la base de datos. No se ejecutan por petición sino ante un evento para el que se ha programado.

Bloque Sintaxis

```
DECLARE
        -- Las declaraciones son opcionales
BEGIN
        --  El cuerpo del programa es necesario
EXCEPTION
        -- Los manejadores de excepciones son opcionales
END;
```

Bloque Sintaxis

- Sección declarativa (DECLARE). Se definen variables, constantes, cursores, excepciones, etc.
- Sección de ejecución (BEGIN-END). Contiene las sentencias ejecutables del bloque.
  - Sólo admite sentencias DML de SQL o instrucciones SQL dinámicas (ensamblar código en tiempo de ejecución).
  - Las sentencias (CREATE, ALTER, DROP, etc.) no están permitidas directamente, salvo que se indique en una instrucción EXECUTE INMEDIATE.
  - Toda instrucción SELECT, no dinámica, debe llevar INTO.
- Sección control de errores (EXCEPTION). Se definen los controles programados para detectar los errores de ejecución del bloque y el tratamiento de los mismos.

Bloque anónimo Ejemplos

- Bloque mínimo / Salida estándar

```
SET SERVEROUTPUT ON

BEGIN
DBMS_OUTPUT.PUT_LINE('Bloque mínimo');
END;
```

- Bloque con declaración de variables

```
SET SERVEROUTPUT ON

DECLARE
fechaAnt   DATE;
fechaPost  DATE;
BEGIN

fechaAnt:= TO_DATE(SYSDATE)-1;
DBMS_OUTPUT.PUT_LINE('Fecha de ayer: '||fechaAnt);

fechaPost:=TO_DATE(SYSDATE)+1;
DBMS_OUTPUT.PUT_LINE('Fecha de mañana:'||fechaPost);

END;
```

Bloque anónimo Ejemplos

- Bloque con SELECT..INTO

```
SET SERVEROUTPUT ON

DECLARE
fecPR  DATE;
genPR     CHAR(1);
BEGIN

SELECT fechaNacimPR, generoPR INTO fecPR, genPR
FROM PERSONA
WHERE rutPR='&RUT_PERSONA';

DBMS_OUTPUT.PUT_LINE('Fecha de Nacimiento: '||fecPR);
DBMS_OUTPUT.PUT_LINE('Género: '||genPR);
END;
```

Bloque anónimo Ejemplos

- Bloque con SELECT..INTO y EXCEPTION

```
SET SERVEROUTPUT ON

DECLARE
fecPR  DATE;
genPR     CHAR(1);
BEGIN

SELECT fechaNacimPR, generoPR INTO fecPR, genPR
FROM PERSONA
WHERE rutPR='&RUT_PERSONA';

DBMS_OUTPUT.PUT_LINE('Fecha de Nacimiento: '||fecPR);
DBMS_OUTPUT.PUT_LINE('Género: '||genPR);

EXCEPTION
WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.PUT_LINE('Rut no encontrado');
END;
```

Declaración Variables Constantes

```
variable [CONSTANT] tipodatos [ [NOT NULL] {DEFAULT|:=} expresión];
```

```
SET SERVEROUTPUT ON

DECLARE
genero  CHAR(1):='F';                -- Una variable con inicialización
suma    NUMBER(3,0):=0;              -- Una variable con inicialización
iva     CONSTANT NUMBER(2,0):=19;    -- Una constante
BEGIN
DBMS_OUTPUT.PUT_LINE('Género: '||genero);

suma:=suma+10;
DBMS_OUTPUT.PUT_LINE('Suma: '||suma);

--  Observa lo que ocurre con esta sentencia
--  Quita el comentario y ejecuta.
--  iva:=iva+1;

DBMS_OUTPUT.PUT_LINE('IVA: '||iva);
END;
```

variable := expresión

Asignación

```
SET SERVEROUTPUT ON

DECLARE
genero  CHAR(1):='F';                -- Una variable con inicialización
suma    NUMBER(3,0):=0;              -- Una variable con inicialización
iva     CONSTANT NUMBER(2,0):=19;    -- Una constante
BEGIN
DBMS_OUTPUT.PUT_LINE('Género: '||genero);

suma:=suma+10;
DBMS_OUTPUT.PUT_LINE('Suma: '||suma);

--  Observa lo que ocurre con esta sentencia
--  Quita el comentario y ejecuta.
--  iva:=iva+1;

DBMS_OUTPUT.PUT_LINE('IVA: '||iva);
END;
```

Salida estándar

- Corresponde a la ejecución del procedimiento DBMS_OUTPUT.PUT_LINE
- Requiere que se haya ejecutado previamente

```
SET SERVEROUTPUT ON
```

```
SET SERVEROUTPUT ON

BEGIN
DBMS_OUTPUT.PUT_LINE('Bloque mínimo');
END;
```

Actividad

1. Implementa cada uno de los ejemplos de esta diapositiva.
2. Guarda cada programa con el nombre Ejemplox.sql
3. Escribe el código de cada ejemplo en una hoja de trabajo distinta.
4. A medida que vayas avanzando ejecuta y observa las salidas de cada programa.