



Universidad
Inca Garcilaso de la Vega

Nuevos Tiempos. Nuevas Ideas

Facultad de Ingeniería de Sistemas, Cómputo y Telecomunicaciones

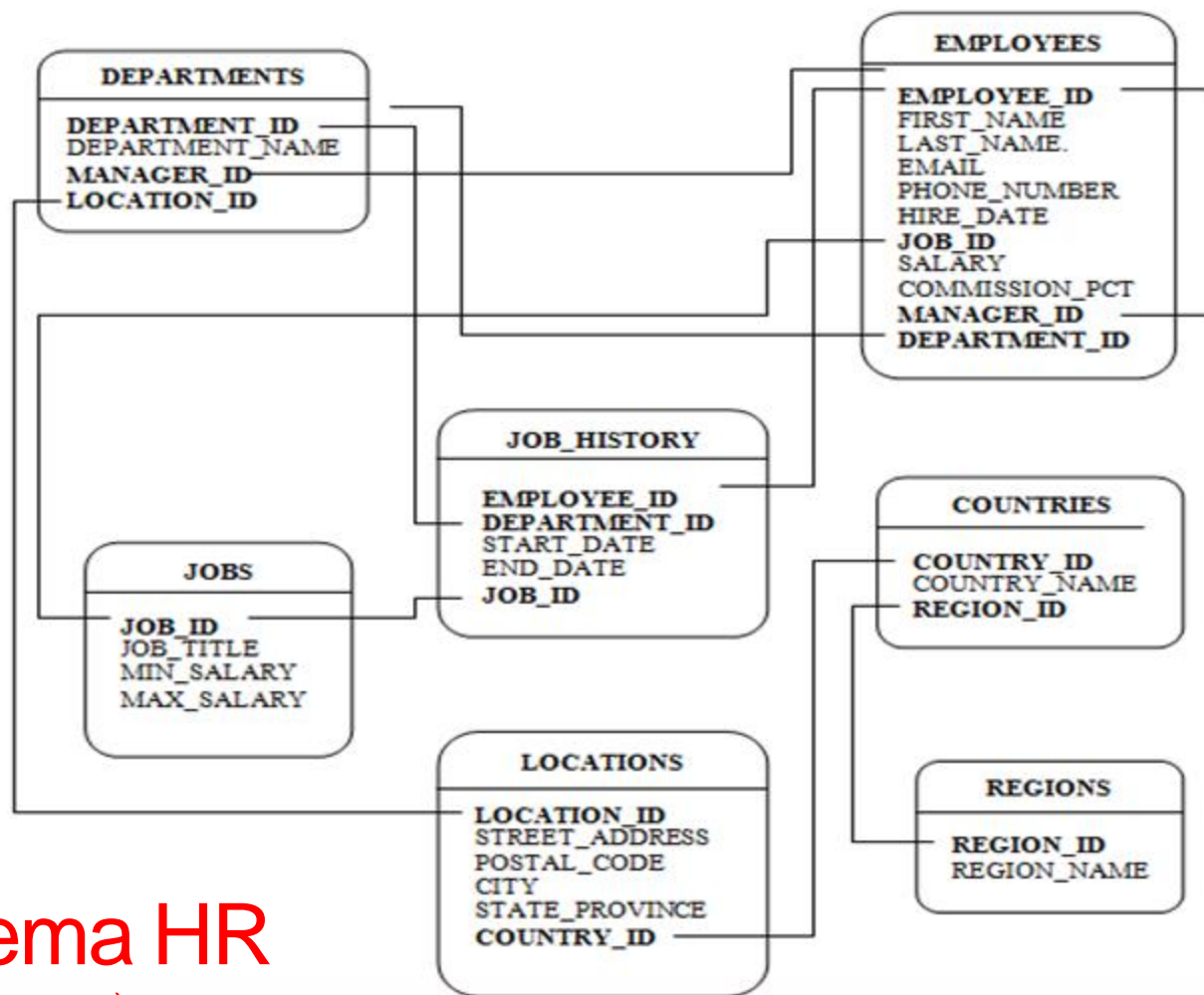
Introducción a Pl/Sql

Asignatura: Gestión de Base de Datos

Semana 04

Docente: Mg. Christian Almóguer Martínez

Mail: almoguer@uigv.edu.pe



Esquema HR

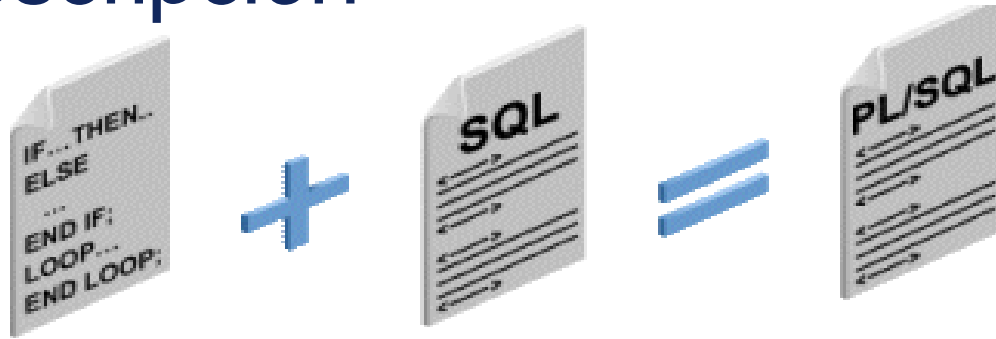
(Recursos Humanos)

AGENDA

- Descripción.
- Ventajas.
- Característica.
- Tipos de Bloques en PL/Sql
- Descripción de los bloques anónimos.

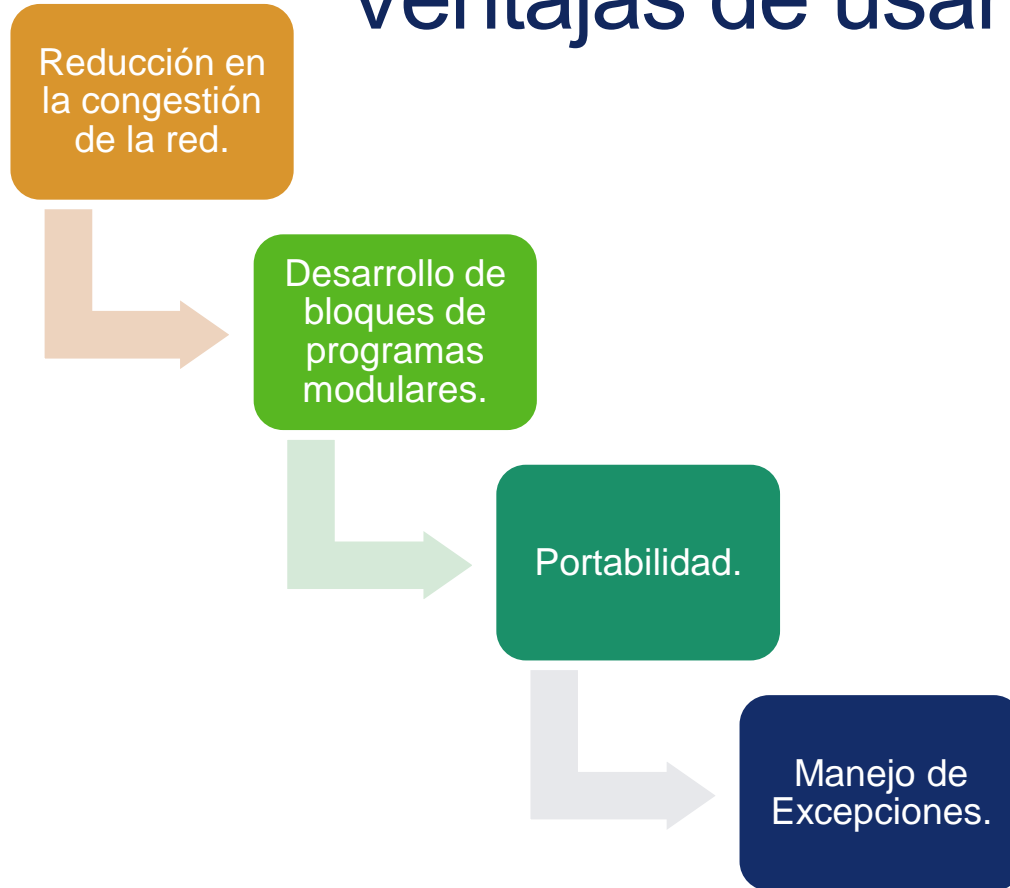


Descripción



- Lenguaje de programación que proporciona Oracle para extender el SQL estándar con otros tipos de instrucciones.
- Es menos potente que los lenguajes para aplicaciones, pero se especializa en el manejo de la base de datos.
- Lenguaje portable, procedural y de transacción.

Ventajas de usar PL/SQL...



Características de PL/SQL

- **Incorpora todos los comandos de SQL:**
select, insert, update y delete.
- **Manejo de sentencias:**
 - **Control condicional:**
IF...THEN...ELSE...ENDIF
 - **Ciclos:**
FOR ... LOOP
WHILE ... LOOP
- **Incluye opciones avanzadas:**
 - Manejo de cursores.
 - Manejo de excepciones.
 - Procedimientos y funciones predefinidos.
 - Disparadores

Tipo de Bloques en PL/SQL

BLOQUES

Tipos de bloques

Bloque Anónimo

```
[DECLARE]

BEGIN
  -- statements

EXCEPTION

END;
```

Procedimiento

```
PROCEDURE name

IS/AS

BEGIN
  --statements

[EXCEPTION]

END;
```

Función

```
FUNTION name
RETURN datatype
IS/AS
BEGIN
  --statements

RETURN value;
[EXCEPTION]

END;
```

BLOQUES ANÓNIMOS

Descripción

PL/SQL, lenguaje estructurado en bloques, los cuales representan expresiones lógicas.

- Un bloque permite agrupar en forma lógica un grupo de sentencias.
- Un bloque en PL/SQL tiene tres partes: una sección de declaración, sección de ejecución y otras de manejos de excepciones.
- Sólo el bloque de ejecución es obligatorio.



Tipos de Datos

Tipo de dato / Sintaxis	Oracle 8i	Oracle 9i	Descripción
dec(p, e)	La precisión máxima es de 38 dígitos.	La precisión máxima es de 38 dígitos.	Donde p es la precisión y e la escala. Por ejemplo: dec(3,1) es un número que tiene 2 dígitos antes del decimal y un dígito después del decimal.
decimal(p, e)	La precisión máxima es de 38 dígitos.	La precisión máxima es de 38 dígitos.	Donde p es la precisión y e la escala. Por ejemplo: decimal(3,1) es un número que tiene 2 dígitos antes del decimal y un dígito después del decimal.
double precision			
float			
int			
integer			
numeric(p, e)	La precisión máxima es de 38 dígitos.	La precisión máxima es de 38 dígitos.	Donde p es la precisión y e la escala. Por ejemplo: numeric(7,2) es un número que tiene 5 dígitos antes del decimal y 2 dígitos después del decimal.
number(p, e)	La precisión máxima es de 38 dígitos.	La precisión máxima es de 38 dígitos.	Donde p es la precisión y e la escala. Por ejemplo: number(7,2) es un número que tiene 5 dígitos antes del decimal y 2 dígitos después del decimal.

Tipos de Datos

Tipo de dato / Sintaxis	Oracle 8i	Oracle 9i	Descripción
real			
smallint			
char (tamaño)	Hasta 32767 bytes en PLSQL. Hasta 2000 bytes en Oracle 8i.	Hasta 32767 bytes en PLSQL. Hasta 2000 bytes en Oracle 9i.	Donde tamaño es el número de caracteres a almacenar. Son cadenas de ancho fijo. Se rellena con espacios.
varchar2 (tamaño)	Hasta 32767 bytes en PLSQL. Hasta 4000 bytes en Oracle 8i.	Hasta 32767 bytes en PLSQL. Hasta 4000 bytes en Oracle 9i.	Donde tamaño es el número de caracteres a almacenar. Son cadenas de ancho variable.
long	Hasta 2 gigabytes.	Hasta 2 gigabytes.	Son cadenas de ancho variable.
raw	Hasta 32767 bytes en PLSQL. Hasta 2000 bytes en Oracle 8i.	Hasta 32767 bytes en PLSQL. Hasta 2000 bytes en Oracle 9i.	Son cadenas binarias de ancho variable.
long raw	Hasta 2 gigabytes.	Hasta 2 gigabytes.	Son cadenas binarias de ancho variable.
date	Una fecha entre el 1 de Enero de 4712 A.C. y el 31 de Diciembre de 9999 D.C.	Una fecha entre el 1 de Enero de 4712 A.C. y el 31 de Diciembre de 9999 D.C.	
timestamp (fractional seconds precision)	No soportado por Oracle 8i.	fractional seconds precision debe ser un número entre 0 y 9. (El valor por defecto es 6)	Incluye año, mes día, hora, minutos y segundos. Por ejemplo: timestamp(6)
timestamp (fractional seconds precision) with time zone	No soportado por Oracle 8i.	fractional seconds precision debe ser un número entre 0 y 9. (El valor por defecto es 6)	Incluye año, mes día, hora, minutos y segundos; con un valor de desplazamiento de zona horaria. Por ejemplo: timestamp(5) with time zone

Tipos de Datos

Tipo de dato / Sintaxis	Oracle 8i	Oracle 9i	Descripción
timestamp (<i>fractional seconds precision</i>) with time zone	No soportado por Oracle 8i.	fractional seconds precision debe ser un número entre 0 y 9. (El valor por defecto es 6)	Incluye año, mes día, hora, minutos y segundos; con un valor de desplazamiento de zona horaria. Por ejemplo: timestamp(5) with time zone
timestamp (<i>fractional seconds precision</i>) with local time zone	No soportado por Oracle 8i.	fractional seconds precision debe ser un número entre 0 y 9. (El valor por defecto es 6)	Incluye año, mes día, hora, minutos y segundos; con una zona horaria expresada como la zona horaria actual. Por ejemplo: timestamp(4) with local time zone
interval year (<i>year precision</i>) to month	No soportado por Oracle 8i.	year precision debe ser un número entre 0 y 9. (El valor por defecto es 2)	Período de tiempo almacenado en años y meses. Por ejemplo: interval year(4) to month
interval day (<i>day precision</i>) to second (<i>fractional seconds precision</i>)	No soportado por Oracle 8i.	day precision debe ser un número entre 0 y 9. (El valor por defecto es 2) fractional seconds precision debe ser un número entre 0 y 9. (El valor por defecto es 6)	Incluye año, mes día, hora, minutos y segundos. Por ejemplo: interval day(2) to second(6)

Tipos de Datos

Tipo de dato / Sintaxis	Oracle 8i	Oracle 9i	Descripción
rowid	El formato del campo rowid es: BBBBBBB.RRRR.FFFFF donde BBBBBBB es el bloque en el fichero de la base de datos; RRRR es la fila del bloque; FFFFF es el fichero de la base de datos.	El formato del campo rowid es: BBBBBBB.RRRR.FFFFF donde BBBBBBB es el bloque en el fichero de la base de datos; RRRR es la fila del bloque; FFFFF es el fichero de la base de datos.	Datos binarios de ancho fijo. Cada registro de la base de datos tiene una dirección física o rowid.
urrowid [tamaño]	Hasta 2000 bytes.	Hasta 2000 bytes.	Rowid universal. Donde tamaño es opcional.
boolean	Válido en PLSQL, este tipo de datos no existe en Oracle 8i.	Válido en PLSQL, este tipo de datos no existe en Oracle 9i.	
nchar (tamaño)	Hasta 32767 bytes en PLSQL. Hasta 2000 bytes en Oracle 8i.	Hasta 32767 bytes en PLSQL. Hasta 2000 bytes en Oracle 9i.	Donde tamaño es el número de caracteres a almacenar. Cadena NLS de ancho fijo.
nvarchar2 (tamaño)	Hasta 32767 bytes en PLSQL. Hasta 4000 bytes en Oracle 8i.	Hasta 32767 bytes en PLSQL. Hasta 4000 bytes en Oracle 9i.	Donde tamaño es el número de caracteres a almacenar. Cadena NLS de ancho variable.
bfile	Hasta 4 gigabytes.	Hasta 4 gigabytes.	Localizadores de archivo apuntan a un objeto binario de sólo lectura fuera de la base de datos.
blob	Hasta 4 gigabytes.	Hasta 4 gigabytes.	Localizadores LOB apuntan a un gran objeto binario dentro de la base de datos.
clob	Hasta 4 gigabytes.	Hasta 4 gigabytes.	Localizadores LOB apuntan a un gran objeto de caracteres dentro de la base de datos.
nclob	Hasta 4 gigabytes.	Hasta 4 gigabytes.	Localizadores LOB apuntan a un gran objeto NLS de caracteres dentro de la base de datos.

Declaración de Variables

- Estas podrán ser utilizadas en cualquier expresión de un programa.
- Toda variable y constante que se utilice en PL/SQL debe estar declarada antes de ser utilizada en una expresión.
- Las variables pueden corresponder a cualquier tipo de dato de SQL.

Declaración de Variables

DECLARE

```
fec_nac    date;  
cantidad   smallint := 10;  
valor      number(5,2);  
precio     number(4) :=450;
```

Declaración de Constantes

- Similar a la declaración de variables, se debe incorporar la palabra reservada “constant” y luego asignar el valor deseado.

DECLARE

```
dolar constant number (2,2) := 3.25;
```

```
lim_credito constant real := 1500;
```


Mostrar Valores

- Para poder visualizar valores dentro de un bloque PL/SQL se debe ejecutar la instrucción:
 - `dbms_output.put_line(expresión);`
 - `http.print (expresión);`

Usar pipe || para concatenar

Asignación de Valores

Hay dos formas de asignar valores a las variables de memoria:

- **Con el operador de asignación (:=)**
promedio := total/cant;
- **Con la sentencia Select y la orden Into**
Select sum(salary) into total
from employees
where hire_date=sysdate;

Ejemplo 1: Mostrar Mensaje

```
DECLARE
--declarando variable
mensaje VARCHAR2(20);

BEGIN
-- Asignación de un valor a la variable
  mensaje := 'Sistemas y Cómputo';
-- Mostrar mensaje
  dbms_output.put_line ( ' Facultad : ' || mensaje );
END;
```

Ejemplo 2: Mostrar el puesto y salario del empleado Neena

```
DECLARE
    puesto varchar2(20);
    sal    number(8,2);
BEGIN
    --Buscar y asignar
    select job_id, salary INTO puesto, sal
    from employees
    where first_name = 'Neena';
    -- mostrar
    dbms_output.put_line ( 'El puesto es ' || puesto );
    dbms_output.put_line ( 'El salario es ' || sal );
END;
```

Ejemplo 3: Mostrar el nombre y fecha de contrato del primer empleado contratado

```
DECLARE
    nombre varchar2(20);
    fecha   date;
BEGIN
    --Buscar y asignar
    select first_name, hire_date INTO nombre, fecha
    from employees
    where hire_date =(select min(hire_date)
                      from employees);

    -- mostrar
    dbms_output.put_line ( 'El nombres es  ' || nombre);
    dbms_output.put_line ( 'La fecha es   ' || fecha );
END;
```

Ejemplo 4: Mostrar el salario diario para el empleado Bruce sabiendo que trabaja 25 días al mes.

```
DECLARE
mensual          number(5);
dias             number(2):=25;
sal_dia  number(5,2);
BEGIN
    select salary into mensual
    from employees
    where first_name='Bruce';
    -- cálculo
    sal_dia:= mensual / dias;
    -- mostrar
    dbms_output.put_line ('Salario Diario : ' || sal_dia);
END;
```

Ejemplo 5: Mostrar los siguientes campos para el primer empleado contratado:
Nombre, nombre de departamento, ciudad y país.

```
DECLARE
```

```
nombre varchar2(15);
```

```
depa   varchar2(15);
```

```
ciudad varchar2(15);
```

```
pais   varchar2(25);
```

```
BEGIN
```

```
-- Consulta Principal
```

```
SELECT first_name, Department_name, city, country_name
```

```
INTO nombre, depa, ciudad, pais
```

```
FROM employees e, departments d, locations l, countries c
```

```
WHERE e.department_id=d.department_id and
```

```
      d.location_id=l.location_id and l.country_id=c.country_id and
```

```
      hire_date =(select min(hire_date) from employees);
```

```
-- Muestra Información
```

```
dbms_output.put_line ( 'Nombre : '||nombre);
```

```
dbms_output.put_line ( 'Departamento : '|| depa);
```

```
dbms_output.put_line ( 'Ciudad : '|| ciudad);
```

```
dbms_output.put_line ( 'País ' ||pais);
```

```
END;
```

Ejemplo 6: Mostrar el apellido del empleado que menos gana y su monto de incremento; sabiendo que es de 15%.

```
DECLARE
nombre varchar2(15);
incre number(3,2) := 0.15;
monto number(8,2);
BEGIN
  -- Consulta Principal
  SELECT last_name, salary * incre INTO nombre, monto
  FROM employees
  WHERE salary =(select min(salary) from employees);

  -- Muestra Información
  dbms_output.put_line ( 'Empleado : ' || nombre || ', su aumento del
                          15% es : ' || monto|| ' dólares');
END;
```