

Gestión y Modelación de Datos

Sesión Practica: Introducción a PL

MARÍA CONSTANZA PABÓN

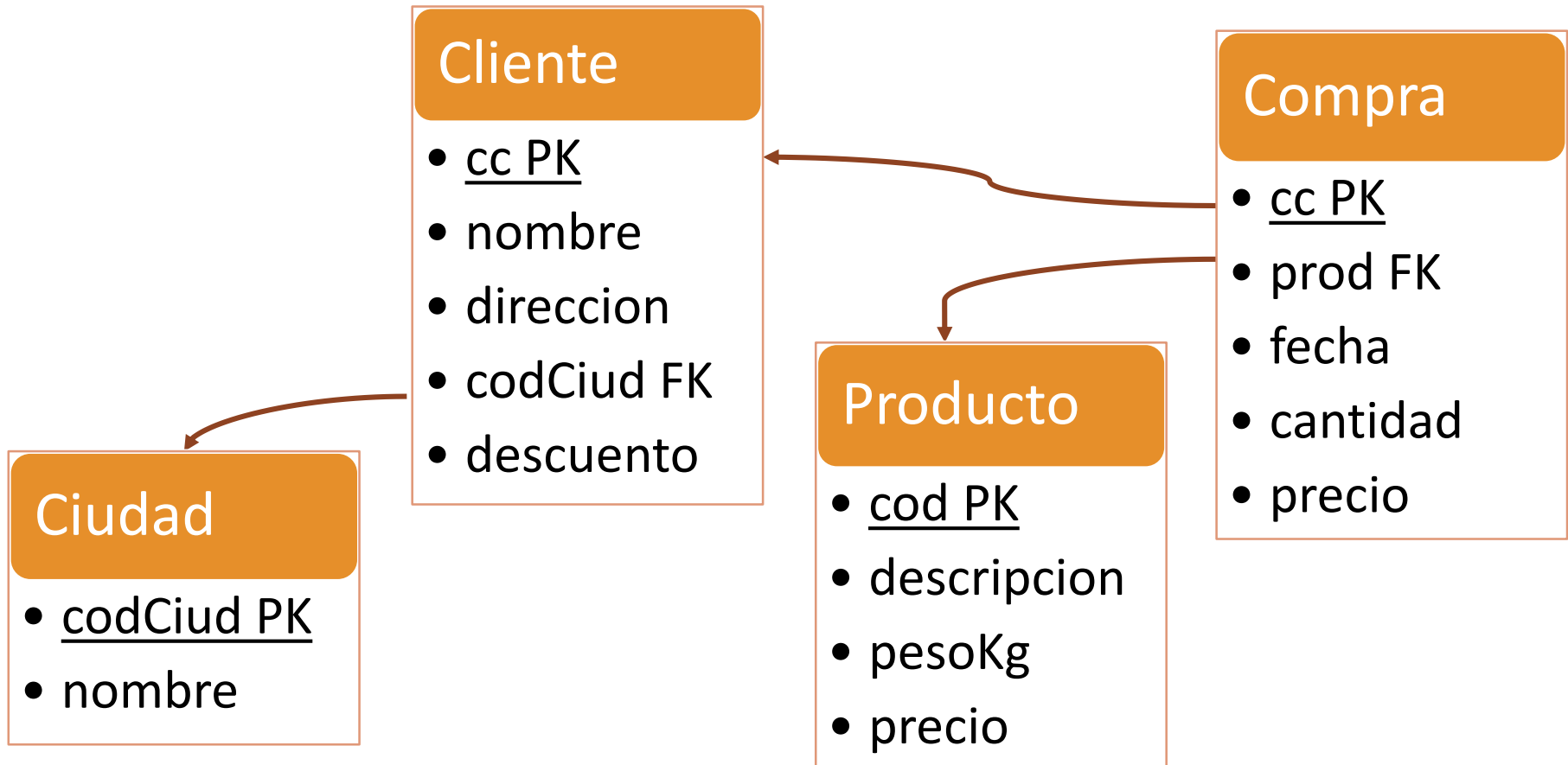
MCPABON@JAVERIANACALI.EDU.CO



PL/SQL (Procedural Language Extension to SQL)

- SQL tiene expresividad limitada
 - No es equivalente a una máquina de Turing
- PL permite usar herramientas generales de programación con SQL
 - Ciclos, funciones y procedimientos
- Reduce la carga de comunicación entre el cliente y la base de datos
- Mejora el rendimiento de los programas
- PL/SQL es soportado por la mayoría de los motores de bases de datos relacionales

Modelo para los ejemplos



Bloques

- La unidad básica de un programa en PL/SQL es el bloque
- Hay dos tipos de bloques:
 - Bloques anónimos
 - Son scripts que se ejecutan directamente (ej. en SQL*PLUS o Developer)
 - Bloques nombrados
 - Funciones y procedimientos almacenados

Bloques anónimos

```
DECLARE /* opcional */  
        /* declaración de variables */  
  
BEGIN   /* obligatorio */  
        /* sentencias, porción ejecutable */  
  
EXCEPTION /* opcional */  
        /* manejo de excepciones */  
  
END;  
  
/      /* En SQL*Plus, para que se ejecute */
```

Si el bloque se ejecuta correctamente, se tendrá el siguiente mensaje:
PL/SQL procedure successfully completed

Bloques anónimos

DECLARE

ccCliente NUMBER(10) := 256;

descuento NUMBER(10,2);

BEGIN

SELECT SUM(precio*cantidad) * 0.1 INTO descuento

FROM compra

WHERE cc = ccCliente

AND fecha BETWEEN '01/07/2020' AND '30/09/2020';

INSERT INTO descuentos VALUES (ccCliente, descuento);

END;

Previamente:

CREATE TABLE descuentos (cc NUMBER(10), descuento NUMBER(10));

Bloques anónimos

DECLARE

nomCliente cliente.Nombre%TYPE := 'Carlos Torres';

ccCliente cliente.CC%TYPE;

descuento NUMBER(10,0);

BEGIN

SELECT cc, SUM(precio*cantidad) * 0.1 INTO ccCliente, descuento

FROM compra NATURAL JOIN cliente

WHERE nombre = nomCliente

AND fecha BETWEEN '01/07/2020' AND '30/09/2020'

GROUP BY cc;

IF descuento IS NOT NULL THEN

INSERT INTO descuentos VALUES (ccCliente, descuento);

END IF;

END;

Bloques nombrados

```
CREATE [OR REPLACE] PROCEDURE procedure_name  
  [ (parameter [mode] datatype,  
    parameter [mode] datatype, ... ) ]  
IS|AS  
  /* declaración de variables */  
BEGIN  
  /* sentencias, porción ejecutable */  
EXCEPTION  
  /* manejo de excepciones */  
END;
```

Mode:

IN: en el llamado se pasa un valor

OUT: el procedimiento retorna un valor

IN OUT: combina los dos anteriores

Por defecto el modo es IN

Bloques nombrados

```
CREATE [OR REPLACE] FUNCTION procedure_name  
[ (parameter [mode] datatype,  
  parameter [mode] datatype, ...)]  
RETURN datatype  
IS|AS  
  /* declaración de variables */  
BEGIN  
  /* sentencias, porción ejecutable */  
EXCEPTION  
  /* manejo de excepciones */  
END;
```

El bloque de sentencias debe incluir la cláusula **RETURN**

Bloques nombrados

```
CREATE OR REPLACE PROCEDURE dctoP (ccCliente NUMBER)
AS
    descuento NUMBER(10);
BEGIN
    SELECT SUM(precio*cantidad) * 0.1 INTO descuento
    FROM compra
    WHERE cc = ccCliente
    AND fecha BETWEEN '01/07/2020' AND '30/09/2020';
    IF descuento IS NOT NULL THEN
        INSERT INTO descuentos VALUES (ccCliente, descuento);
    END IF;
END;
```

Sin tamaño

Ejecución:

```
BEGIN
    dctoP( 256 );
END;
```

Bloques nombrados


```
CREATE OR REPLACE FUNCTION dctoF (ccCliente NUMBER)
RETURN NUMBER
AS
    descuento NUMBER(10,2);
BEGIN
    SELECT SUM(precio*cantidad) * 0.1 INTO descuento
    FROM compra
    WHERE cc = ccCliente
    AND fecha BETWEEN '01/07/2020' AND '30/09/2020';

    RETURN descuento;
END;
```

Ejemplo ejecución:

```
BEGIN
    dbms_output.put_line(dctoF(256));
END;
```

En developer ir a ver-
Salida DBMS, y agregar
conexión a su bd



Salida por consola

- DBMS_OUTPUT.PUT_LINE (expresión)
- En Developer: abrir la salida de DBMS (menú Ver), y en la ventana que se abre, usar el **+** para agregar la conexión
- En SQL*PLUS: ejecutar **SET SERVEROUTPUT ON** antes de usar las salidas por dbms (la salida aparece antes del mensaje de ejecución exitosa)

Errores de procedimientos y funciones en SQL*PLUS

- SQL*PLUS no despliega los errores de compilación de los procedimientos y las funciones
- Para verlos, ejecute:
`SHOW ERRORS PROCEDURE nombre_procedimiento`
`SHOW ERRORS FUNCTION nombre_funcion`

Declaración de variables y constantes

```
variable_name [ CONSTANT ] data_type [ NOT NULL ] [ {  
    DEFAULT | := } expression ];
```

- Ejemplos:

```
DECLARE
```

```
    birthday DATE;
```

```
    age      NUMBER(2) NOT NULL := 27;
```

```
    name     VARCHAR2(13) := 'Levi';
```

```
    magic    CONSTANT NUMBER DEFAULT 77;
```

```
    valid    BOOLEAN NOT NULL := TRUE;
```

```
    nomCliente cliente.Nombre%TYPE
```

Sentencias

- Asignación (:=)

nombre := 'Sofia';

- SELECT INTO

```
SELECT cc INTO ccCliente  
FROM compra NATURAL JOIN cliente  
WHERE nombre = nomCliente
```

Sentencias

- IF – THEN – ELSEIF – ELSE - END IF

```
IF valor > 300000 THEN
    descuento := 20000;
ELSIF valor > 150000 THEN
    descuento := 150000;
ELSE bono := 10000;
END IF;
```

- RETURN expression

```
RETURN descuento;
```


Sentencias

- WHILE – END LOOP

```
WHILE exp_booleana LOOP  
    /* sentencias */  
END LOOP;
```

- FOR – LOOP – END LOOP

```
FOR variable IN [REVERSE] rango LOOP  
    /* sentencias */  
END LOOP;
```

```
FOR i IN 1..3 LOOP  
    DBMS_OUTPUT.PUT_LINE (i);  
END LOOP;
```

Sentencias

- LOOP – EXIT – END LOOP

```
LOOP  
  /* sentencias */  
  EXIT WHEN exp_booleana  
END LOOP;
```

- Llamado a procedimientos y funciones

```
dctoP(256);  
rta := dctoF(256);
```

Cursor Implícito

```
BEGIN
  FOR pers IN ( SELECT nombre, direccion
                FROM cliente
                WHERE codCiud = 7601)
  LOOP
    DBMS_OUTPUT.PUT_LINE
      (pers.nombre || '|' || pers.direccion);
  END LOOP;
END;
```

Se crea un registro con
los atributos del SELECT

Notación:
registro.atributo

Borrar funciones y procedimientos

DROP FUNCTION [*schema.*]*function_name*;

```
DROP FUNCTION dctoF;
```

DROP PROCEDURE [*schema.*]*procedure_name*;

```
DROP PROCEDURE dctoP;
```

Comentarios

-- comentario línea

/ Comentario de
Bloque */*

Referencias

Manual de Oracle. En línea:

Oracle Database PL/SQL Language Reference

https://docs.oracle.com/cd/E11882_01/appdev.112/e25519/toc.htm