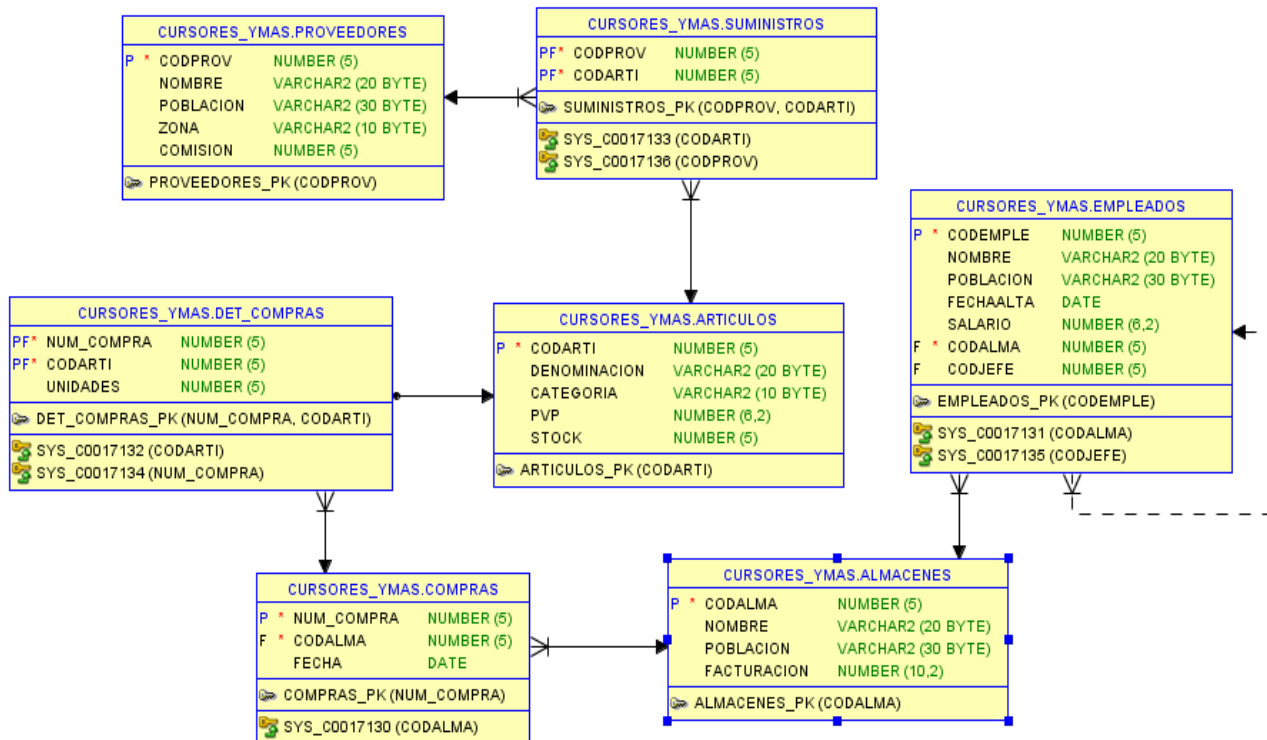


PARA REALIZAR LAS CONSULTAS UTILIZA LA BASES DE DATOS BDALMACEN



1. Realizar un bloque PL para insertar registros en la tabla SUMINISTROS. El código de proveedor y de artículo leerlo de teclado.

Comprobar todos los posibles errores:

- ✓ Que no exista el artículo
- ✓ Que no exista el proveedor
- ✓ Que exista ese proveedor con ese artículo.

Declare

```

Cp Number(5);
Ca Number(5);
Np Number(5):=0;
na number(5):=0;
  
```

Begin

```

Cp:= &valor_Para_Codproveedor;
Ca:= &valor_Para_Codarticulo;
  
```

```

/*Comprobamos si existe el proveedor en la tabla proveedores */
Select Count(*) Into Np From Proveedores Where Codprov=Cp;
  
```

```

If Np = 0 Then
  
```

```

    Dbms_Output.Put_Line('No existe el proveedor: ' || Cp || ', no se inserta suministros');
  end if;
  
```

```

/*Comprobamos si existe el artículo en la tabla artículos */
Select Count(*) Into Na From Articulos Where Codarti=Ca;
  
```

```

If Na = 0 Then
  
```

```

    Dbms_Output.Put_Line('No existe el artículo: ' || Ca || ', no se inserta suministros');
  End If;
  
```

EJERCICIO 1. COMPROBACIONES TEMA 10.

BASES DE DATOS.

```
If Na!=0 And Np!=0 Then
  /* Grabo el suministro */
  Insert Into Suministros Values (Cp, Ca);
  Dbms_Output.Put_Line('Suministro grabado, proveedor: ' || Cp || ', artículo: '
  || Ca);
  commit;
end if;

Exception
  When Dup_Val_On_Index Then
    Dbms_Output.Put_Line('Error, suministro ya existe, proveedor: ' || Cp || ', artículo:'
    || Ca);
End;
```

2. Realizar un bloque PL para insertar registros en la tabla ARTICULOS. Leer de teclado DENOMINACIÓN, CATEGORÍA y PRECIO. Insertar un artículo en la tabla ARTICULOS, teniendo en cuenta:

- ✓ El código de artículo será el valor máximo del código actual más 1.
- ✓ El stock debe ser la media de stock de los artículos suministrados por proveedores de la zona CENTRO.
- ✓ El precio Tecleado tiene que ser > 0.
- ✓ La categoría debe ser A, B o C.
- ✓ Visualizar todos los mensajes de error que ocurran.

Declare

Codar Articulos.Codarti%Type;

Den Articulos.Denominacion%Type;

Cat Articulos.Categoria%Type;

Pvp Articulos.Pvp%Type;

Stock Articulos.Stock%Type;

sw number(1):=0;

Begin

Den:= '&Denominacion';

Cat:= Upper('&Categoria');

Pvp:= &precio_Articulo;

/* Comprobar categoría */

If Cat Not In ('A','B','C') Then

Sw:=1;

Dbms_Output.Put_Line('Categoría Errónea : ' || cat);

End If;

If pvp <=0 Then

Sw:=1;

Dbms_Output.Put_Line('Precio erróneo : ' || pvp);

End If;

If Sw=0 Then

/*resto de calculos y grabar */

Select Max(Codarti)+1 Into Codar From Articulos;

Select Avg(Stock) Into Stock From Articulos

Join Suministros Using(Codarti)

Join Proveedores Using(Codprov)

Where Upper(Zona) = 'CENTRO';

/* insertamos */

Insert Into Articulos Values (Codar, Den, Cat,Pvp,Stock);

Dbms_Output.Put_Line('REGISTRO GRABADO:');

I.E.S. RIBERA DEL TAJO.

CURSO 2017 / 2018.

```

Dbms_Output.Put_Line('Codarti :'||Codar || ', Deno: ' || Den
|| ', categoria: ' || Cat || ', Precio: ' || Pvp
|| ', Stock: ' || Stock);
commit;
Else
Dbms_Output.Put_Line('NO SE GRABA, ERROR DATOS. ');
End If;
End;

```

3. Leer de teclado la subida, la categoría y el código de almacén.

- ✓ Hacer un bloque PL, que actualice el stock de los artículos de una categoría leída, y que sean comprados por el almacén leído.
- ✓ La actualización consiste en sumar la subida al stock.
- ✓ Comprobar que el almacén exista.
- ✓ Comprobar que la subida sea >0
- ✓ Comprobar que la categoría se A, B, C
- ✓ Visualizar todos los mensajes de error.
- ✓ Visualizar también los artículos que se actualizan

```

Declare
Subida number(5);
Cat Articulos.Categoria%Type;
alma Almacenes.Codalma%Type;
sw number(1):=0;
Begin
alma:= &Codigo_almacen;
Cat:= Upper('&Categoria');
Subida:= &Subida_stock;
/* Comprobar categoría */
If Cat Not In ('A','B','C') Then
Sw:=1;
Dbms_Output.Put_Line('Categoría Errónea : ' || cat );
End If;
/* Comprobar subida */
If Subida <=0 Then
Sw:=1;
Dbms_Output.Put_Line('Subida errónea : ' || Subida );
End If;

/* Comprobar almacén, si no existe sale por when no-data-found */
select codalma into alma from almacenes where codalma=alma;

If Sw=0 Then
/*actualizar */
Update Articulos Set Stock=Stock + Subida
Where Categoria = Cat And
Codarti In
(Select DISTINCT Codarti From Compras
Join Det_Compras Using(Num_Compra)
WHERE CODALMA=ALMA);

Dbms_Output.Put_Line('REGISTROS ACTUALIZADOS:' || SQL%ROWCOUNT);
commit;

```

```
Else
  Dbms_Output.Put_Line('NO SE Actualiza, ERROR DATOS.');
```

End If;

Exception

When No_Data_Found Then

```
  Dbms_Output.Put_Line('NO EXISTE EL ALMACÉN,'||alma||', NO SE ACTUALIZA.');
```

End;

4. Realizar un bloque PL para insertar datos en la tabla DET_COMPRAS se leen de teclado NUM_COMPRA, CODARTI, UNIDADES.

Antes de insertar comprobar:

- ✓ Que el NUM_COMPRA + CODARTI no exista en la tabla DET_COMPRAS, comprobar error de clave duplicada.
- ✓ Si no existe NUM_COMPRA + CODARTI, comprobar que exista el num_compra. Comprobar que exista el Codarti. Comprobar que unidades sean >0. Y grabar.
- ✓ No se insertará si existe NUM_COMPRA + CODARTI. O si el num_compra no existe en COMPRA. O si el codarti no existe en ARTICULOS. O cuando unidades<=0.

SOLUCIÓN 1.

```
Declare
  C_Arti number(5);
  N_Compra Number(5);
  Nc number(5);
  Nar number(5);
  Uni number(5);
  sw number(1):=0;

Begin
  C_arti:=&Codigo_Articulo;
  N_Compra :=&Numero_compra;
  Uni:=&Unidades;

  /* Comprobar si existe el detalle */
  Select NUM_COMPRA, CODArTI into nc, nar
    From det_compras where NUM_COMPRA = N_Compra
      And codarti = C_arti;

  Dbms_output.put_line('Ya existe ese detalle, codarti= '||
  C_arti || ', y num compra= '||n_compra);

EXCEPTION
  WHEN NO_DATA_FOUND THEN

  /* COMPROBAR POR SEPARADO EL NUM COMPRA Y EL ARTICULO*/
  begin
    Select num_compra Into nc From Compras
      Where num_compra = n_compra;
  Exception
    When No_Data_Found Then
      Dbms_Output.Put_Line('ERROR COMPRA NO EXISTE: '||n_compra||', NO SE
  INSERTA');
```

```

Sw:=1;
End;

/* COMPROBAR EL ARTÍCULO */
begin
Select CODARTI Into nar From articulos
Where CODARTI = c_arti;
Exception
When No_Data_Found Then
Dbms_Output.Put_Line('ERROR ARTICULO NO EXISTE: ' || c_arti || ', NO SE INSERTA');
Sw:=1;
End;

If Uni <=0 then
Dbms_Output.Put_Line('ERROR, las unidades tienen que ser >0, NO SE INSERTA');
Sw:=1;
End if;

If Sw=0 Then
Insert Into Det_Compras Values(N_Compra,C_Arti, Uni);
commit;

Dbms_Output.Put_Line('Reg insertado:' || ' ', Num compra:' || n_compra || ' ', Num
articulo:' || c_arti || ' ', Unidades:' || Uni);
End if;
End;

```

SOLUCIÓN 2. Utilizamos la excepción DUP_VAL_ON_INDEX para controlar clave duplicada.

Para comprobar si existe utilizamos el contador.

```

Declare
C_Arti number(5);
N_Compra Number(5);
Nc number(5);
Nar number(5);
Uni number(5);
sw number(1):=0;

Begin
C_arti:=&Codigo_Articulo;
N_Compra :=&Numero_compra;
Uni:=&Unidades;

/* comprobar si existe el articulo con contador */
Select count(*) into nar from articulos where codarti=c_arti;

/* comprobar si existe la compra con contador */
Select count(*) into nc from compras where num_compra=n_compra;

If nar = 0 then
Dbms_output.put_line('El artículo ' || c_arti || ' No existe');
Sw:=1;
End if;

```

EJERCICIO 1. COMPROBACIONES TEMA 10.
BASES DE DATOS.

```
If nc = 0 then
    Dbms_output.put_line('La compra ' || n_compra || ' No existe');
    Sw:=1;
End if;

If uni <= 0 then
    Dbms_output.put_line('Las unidades son menor= que 0. ');
    Sw:=1;
End if;

If sw = 0 then
    Insert Into Det_Compras Values(N_Compra,C_Arti, Uni);
    commit;
    Dbms_Output.Put_Line('Reg insertado: ' || ' ', Num compra: ' || n_compra ||
    ', Num articulo: ' || c_arti || ' ', Unidades: ' || Uni);
End if;

Exception
When Dup_val_on_index then
Dbms_output.put_line('Ya existe ese detalle, codarti= ' || C_arti || ', y num compra= ' ||
n_compra);
End;
```