mysql workbench

indice

Poceso de creación de una base de datos

1. Análisis -> Universo del discurso
2. Modelo conceptual -> Diagrama entidad relación (EER)
3. Modelo lógico relacional
4. Modelo físico (SQL) -> DDL, DCL, DML

Mysql Workbench y el modelado de datos

Es una herramienta CASE. Nos facilita trabajar en una base de datos los puntos 3 y 4 (modelo lógico y físico)

1. Obtener el modelo lógico a partir del modelo físico (REVERSE ENGINEER)
2. Obtener el modelo físico a partir del modelo lógico (FORWARD ENGINEER)

Estos se guardan a parte en unos ficheros llamados Modelos de Workbench (mwb)

ejercicio

A partir del modelo relacional banco, crear una base de datos llamado banco , usuario y contraseña banco, otorgar privilegios y crear una conexión llamada banco.

Obtener el modelo workbench (.mwb) y la copia de seguridad .sql (forward engineer)

Formato PDF con portada, índice, estilos de página.

Enlace a Drive con .mwb y .sql (creación de las tablas)

Nombre del ciclista y edad de ciclistas que han pertenecido a equipos de españa

select distinct c.nombrec , timestampdiff(year, c.fecna, current\_date) as Edad

from ciclista c, pertenece p, equipo e

where c.nombrec=p.nombrec and p.nombree=e.nombree

and e.nacion like 'España'

order by 2;

Obtener los datos de los ciclistas nacidos en la década de los 80 que han ganado en alguna prueba cuyo km totales sean superiores a los km de todas las pruebas celebradas durante la década de los 80

select distinct c.\*

from prueba p, ciclista c

where p.nombrec=c.nombrec

and year(fecna) between 1980 and 1989

and km > all (select km from prueba where agno between 1980 and 1989);

que paises han ganado más de tres veces a algunos de sus ciclistas el tour de francia.

select nacion, count(\*)

from ciclista c, prueba p

where p.nombrec=c.nombrec

and nombrep like 'TOUR DE FRANCIA'

group by nacion

having count(\*) > 3

order by 2;

Vamos a crear una vista lógica llamada dempe, va a tener los siguientes rasgos:

* El numem que va a llamarse empleado
* A partir de nomem vamos a extraer el apellido y lo vamos a llamar apel
* A partir de nomem vamos a extraer el nombre y o vamos a llamar nombre
* A partir de fecna -> edad
* A partir de fecin -> anti
* A partir del salar -> sbruto=salar; sneto=64%salar; ss=16%salar

Toda vista lógica tiene como sentido que responda a una vista lógica

Vamos a crear un usuario llamado operado con contraseña operador

Vamos a darles privilegios, las actualizaciones afectan a insert, delete y update, a parte de las restricciones, si las columnas de la vista son cálculos de la vista original esa vista no se puede modificar, el único que le podemos darle el privilegio es select, ya que la vista contiene columnas que en realidad son cálculos realizados con expresiones con los datos que están en la tabla temple

Segunda utilidad de las vistas

Facilitar las consultas complejas

Crear una vista reucd en los que aparezcan los datos de los centros y los depatamentos

numce -> ncentro

nomce -> centro

direccion -> direccion

numde -> ndepartamento

nomde -> departamento

direc -> director

tidir -> tipo\_direccion

presu -> presupuesto

ejercicio:

partiendo de la bd de paises, vamos a crear tres vistas logicas

1 - Pais\_ciudad. Aparezcan nombre de pais como pais y el nombre de la ciudad como ciudad. Ej España - Madrid

create view pais\_ciudad as

select p.nombre as Pais, c.nombre as Ciudad

from pais p, ciudad c

where p.cod\_pais=c.cod\_pais;

2 - Pais idioma. nombre como pais y idioma como idioma

create view pais\_idioma as

select distinct p.nombre as Pais, i.nombre as Idioma

from pais p, se\_habla s, idioma i

where p.cod\_pais=s.cod\_pais and s.cod\_idioma=i.cod\_idioma;

3 - Pais\_organizacion. nombre de pais como pais y las siglas de la org como organizacion

create view pais organizacion as

select distinct p.nombre as Pais, o.siglas as Organizacion

from pais p, pertenece\_a pe, organizacion o

where p.cod\_pais=pe.cod\_pais and pe.cod\_organizacion=o.cod\_organizacion;

Práctica

Título: Conector ODBC entre access y mysql

A partir del script del examen del videoclub sql, proceso de creación en mysql workbench, esquema, usuario, privilegios, ejecutar script con capturas (lo mismo con banco).

Configurar la conexión ODBC para conectarse a la base de datos y a continuación desde Access:

1 - Conectar la bd de videoclub con el modo importar

2 - Guardar un fichero de access llamado videoclubdeimportada

3 - Vinculación (hacer capturas) y hacer un cambio en workbench para comprobar que sale en access

Entregar el pdf y el archivo de access de las bases de datos importadas

Aumentar la fecha en 20 años la fecha de nacimiento y la fecha de ingreso en oracle

update temple set fecna=date\_add(fecna, interval 20 year);

ó

update temple set fecna=add\_months(fecna, 240);

current\_date = sysdate

Calcular la diferencia entre fechas

months\_between (en meses)

select trunc(months\_between(fecha\_más\_reciente, fecha\_más\_antigua)/12,0) from nombre\_tabla;

Crear vista lógica llamada demple:

numem -> empleado

a partir de nomem -> nombre

a partir de nomem -> apellido

a partir de fecna -> edad

a partir de fecin -> anti (antigüedad)

Formas de crear una base de datos:

* Manualmente
* MySQL

1. Crear usuario
2. Crear esquema BD
3. Otorgar privilegios al usuario sobre ese esquema
4. Crear la conexión
5. Crear script DDL
6. Ejecutar script

* Oracle

1. Crear usuario
2. Otorgar privilegios sobre la conexión
3. Crear la conexión
4. Crear el script DDL
5. Ejecutar el script

* Modelado de datos
* MySQL

1. Crear usuario
2. Crear esquema BD
3. Otorgar privilegios al usuario sobre ese esquema
4. Crear la conexión
5. Crear un modelo y un diagrama EER
6. Hacer forward engineer

* Oracle

1. En Developer, crear usuario
2. Otorgar privilegios sobre la conexión
3. Crear la conexión
4. En Data Modeler crear un diseño y un diagrama EER
5. Con la opción Export, exportar y guardar el script DDL
6. En Developer, ejecutar el script DDL en la conexión

ALGORITIMIA Y PROGRAMACIÓN

Un algoritmo o un programa es una secuencia ordenada, de pasos finitos que me resuelven un problema

Procesador: Es capaz de ejecutar instrucciones. Estas instrucciones pueden ser

* Acciones elementales
* Estructuras de control
* Otros algoritmos

También necesitaremos:

* Datos de entrada
* Datos de salida

Ejemplo de la

Procesador: Objetos para cocinar como por ejemplo una sartén

Instrucciones: Calentar, batir, etc

Datos de entrada: Alimentos, productos

Datos de salida: Plato preparado

Si haces arroz para dos personas, saldrá arroz para dos personas

Programa: Algoritmo que se lleva al ordenador

PL SQL(Program language of SQL)

Todo algoritmo tiene una serie de características:

* Su número de pasos es finito
* Es flexible, puede tener más de una solución (tenemos que tener en cuenta la eficiencia para resolver el problema)
* Es preciso

Cursor

set serveroutput on;

declare

cursor c1 is

select nombre, capital

from pais

where continente like 'Europa';

nom\_pais varchar2(50);

capi\_pais varchar2(50);

begin

if not c1%isopen then

open c1;

end if;

dbms\_output.put\_line('PAIS\*\*\*\*\*\*\*\*\*\*\*CAPITAL');

loop

fetch c1 into nom\_pais, capi\_pais;

exit when c1%notfound;

dbms\_output.put\_line(nom\_pais || ' - ' || capi\_pais);

end loop;

close c1;

end;

Atributo rowtype

set serveroutput on;

declare

cursor c1 is

select nombre, capital

from pais

where continente like 'Europa';

reg\_pais c1%rowtype;

begin

if not c1%isopen then

open c1;

end if;

dbms\_output.put\_line('PAIS\*\*\*\*\*\*\*\*\*\*\*CAPITAL');

loop

fetch c1 into reg\_pais;

exit when c1%notfound;

dbms\_output.put\_line(reg\_pais.nombre || ' - ' || reg\_pais.capital);

end loop;

close c1;

end;

Cursor que me imprima los nombre de los idiomas que se hablan en España

set serveroutput on;

declare

cursor c1 is

select i.nombre

from pais p, se\_habla s, idioma i

where p.cod\_pais=s.cod\_pais and s.cod\_idioma=i.cod\_idioma

and p.nombre like 'España';

nom\_idioma varchar2(50);

begin

if not c1%isopen then

open c1;

end if;

dbms\_output.put\_line('IDIOMAS QUE SE HABLAN EN ESPAÑA:');

loop

fetch c1 into nom\_idioma;

exit when c1%notfound;

dbms\_output.put\_line(nom\_idioma);

end loop;

close c1;

end;

Mostrar los idiomas si le paso un parámetro con el nombre del país

set serveroutput on;

declare

cursor c1(p\_pais varchar2) is

select i.nombre

from pais p, se\_habla s, idioma i

where p.cod\_pais=s.cod\_pais and s.cod\_idioma=i.cod\_idioma

and p.nombre like p\_pais;

nom\_idioma varchar2(50);

ppais varchar2(50);

begin

ppais:=&Intro\_pais;

if not c1%isopen then

open c1(ppais);

end if;

dbms\_output.put\_line('IDIOMAS QUE SE HABLAN:');

loop

fetch c1 into nom\_idioma;

exit when c1%notfound;

dbms\_output.put\_line(nom\_idioma);

end loop;

close c1;

end;

Le pasamos el parámetro de continente

set serveroutput on;

declare

--Declaro el cursor

cursor c1(p\_conti varchar2) is

select nombre, capital

from pais

where continente like p\_conti;

--Declaro las variables para el volcado del cursor

nom\_pais varchar2(50);

nom\_capital varchar2(50);

--Declaro la variable para pasar el cursor

pconti varchar2(50);

begin

--Leo el valor del parametro

pconti:=&Intro\_continente;

--Abro el cursor pasandole el parametro

if not c1%isopen then

open c1(pconti);

end if;

--Imprimo la cabecera

dbms\_output.put\_line('PAIS\*\*\*\*\*\*\*\*\*\*\*\*CAPITAL');

loop

--Vuelco la fila

fetch c1 into nom\_pais, nom\_capital;

--Compruebo que no esta vacia

exit when c1%notfound;

--Imprimo la fila

dbms\_output.put\_line(nom\_pais || ' - ' || nom\_capital);

end loop;

--Cierro el cursor

close c1;

end;

Ejercicio: Modificar el código para que el cursor reciba el nombre de un idioma y me diga cuantos países por continente hablan este idioma.

set serveroutput on;

declare

cursor c1(pidioma varchar2) is

select continente, count(\*) as total

from pais p, se\_habla s, idioma i

where p.cod\_pais=s.cod\_pais

and s.cod\_idioma=i.cod\_idioma

and i.nombre like pidioma

group by continente;

reg\_pais c1%rowtype;

paraidioma varchar2(50);

begin

paraidioma:=&Dame\_idioma;

if not c1%ISOPEN then

open c1(paraidioma);

end if;

DBMS\_OUTPUT.PUT\_LINE('CONTINENTE TOTAL PAÍSES');

fetch c1 into reg\_pais;

while C1%FOUND loop

DBMS\_OUTPUT.PUT\_LINE(reg\_pais.continente||' '||reg\_pais.total);

fetch c1 into reg\_pais;

end loop;

close c1;

end;

A partir del código de Países de la ONU, modifica para que las siglas de la organización las reciba como parámetro

Ejercicio 1: Bloque de código con un cursor que reciba como parametro 2 extensiones y que me diga el nombre y la capital de los países cuya extensión esté entre esos dos valores

set serveroutput on;

declare

cursor c1(p\_min decimal(10,0), p\_max decimal(10,0)) is

select nombre, capital

from pais

where p\_min<extension

and p\_max>extension;

nom\_pais varchar2(50);

pext\_min decimal(10,0);

pext\_max decimal(10,0);

begin

pext\_min:=&Intro\_extension\_minima;

pext\_max:=&Intro\_extension\_maxima;

if not c1%isopen then

open c1(pext\_min, pext\_max);

end if;

dbms\_output.put\_line('PAIS CAPITAL');

loop

fetch c1 into nom\_pais;

exit when c1%notfound;

dbms\_output.put\_line(nom\_pais);

end loop;

close c1;

end;

Ejercicio 2: Un cursor que reciba el nombre de un continente y me diga el nombre de las ciudades de países de ese continente

set serveroutput on;

declare

cursor c1(p\_continente varchar2) is

select distinct c.nombre

from pais p, ciudad c

where p.cod\_pais=c.cod\_pais

and continente like p\_continente;

nom\_ciudad varchar2(50);

pcontinente varchar2(50);

begin

pcontinente:=&Intro\_continente;

if not c1%isopen then

open c1(pcontinente);

end if;

dbms\_output.put\_line('CIUDADES DEL CONTINENTE:');

loop

fetch c1 into nom\_ciudad;

exit when c1%notfound;

dbms\_output.put\_line(nom\_ciudad);

end loop;

close c1;

end;

Ejercicio 3: Un cursor que reciba el nombre de un idioma y me diga cuantas ciudades hay por cada país que habla ese idioma

set serveroutput on;

declare

cursor c1(p\_idioma varchar2) is

select distinct p.nombre, count(\*) as total

from pais p, se\_habla s, idioma i, ciudad c

where c.cod\_pais=p.cod\_pais

and p.cod\_pais=s.cod\_pais

and s.cod\_idioma=i.cod\_idioma

and i.nombre like p\_idioma

group by p.nombre;

reg\_pais c1%rowtype;

pidioma varchar2(50);

begin

pidioma:=&Intro\_idioma;

if not c1%isopen then

open c1(pidioma);

end if;

dbms\_output.put\_line('PAIS CIUDADES');

fetch c1 into reg\_pais;

while c1%found loop

dbms\_output.put\_line(reg\_pais.nombre||' '||reg\_pais.total);

fetch c1 into reg\_pais;

end loop;

close c1;

end;

Programación Modular

Ejemplo: Parametrizar consultas

CREATE OR REPLACE PROCEDURE PAISES\_ORGA

(

PSIGLAS IN VARCHAR2

) AS

cursor c1(psiglas varchar2) is

select p.nombre

from pais p, pertenece\_a pe, organizacion o

where p.cod\_pais=pe.cod\_pais and pe.cod\_organizacion=o.cod\_organizacion

and siglas like psiglas;

npais varchar2(50);

BEGIN

if not c1%ISOPEN then

open c1(psiglas);

end if;

dbms\_output.put\_line('Paises de la '||psiglas);

fetch c1 into npais;

while C1%FOUND loop

DBMS\_OUTPUT.PUT\_LINE(npais);

fetch c1 into npais;

end loop;

NULL;

END PAISES\_ORGA;

1. Parametrizar consultas
2. Campos calculados
3. Enmascaramiento de funciones
4. Triggers o disparadores

Ejemplo: Sobre la base de datos de empleados, vamos a crear una función que reciba como parametro el numero de un departamento y el numero de hijos y me devuelva el numero de empleadosd de ese departamento .

Para la llammada vamos a crear un vloque de codigo anonimo en el que vamos a controlar el valor devuelto es cero(no hay empleados en ese departamento) o si es diferrente a 0(hay n empleados con esos hijos).

CREATE OR REPLACE FUNCTION EMPLE\_HIJOS\_DEPTO

(

PDEPTO IN NUMBER

, PHIJOS IN NUMBER

) RETURN NUMBER AS

total number;

BEGIN

select count(\*) into total

from temple

where numde=pdepto

and numhi=phijos;

RETURN total;

END EMPLE\_HIJOS\_DEPTO;

set serveroutput on;

declare

paradepto number;

parahijos number;

totalemple number;

begin

paradepto:=&Dame\_departamento;

parahijos:=&Dame\_hijos;

totalemple:=emple\_hijos\_depto(paradepto,parahijos);

if totalemple=0 then

dbms\_output.put\_line('No hay ningún empleado con '||parahijos||' hijos en el departamento '||paradepto);

else

dbms\_output.put\_line('En el departamento '||paradepto||' hay '||totalemple||' empleados con '||parahijos||' hijos');

end if;

end;

Otra forma para tratar los cursores (mediante un bucle for)

Ejemplo:

create or replace

PROCEDURE PAISES\_ORGA\_FOR\_IN

(

PSIGLAS IN VARCHAR2

) AS

cursor c1(psiglas varchar2) is

select p.nombre as npais

from pais p, pertenece\_a pe, organizacion o

where p.cod\_pais=pe.cod\_pais and pe.cod\_organizacion=o.cod\_organizacion

and siglas like psiglas;

BEGIN

dbms\_output.put\_line('Paises de la '||psiglas);

for reg\_pais in c1(psiglas) loop

dbms\_output.put\_line(reg\_pais.npais);

end loop;

NULL;

END PAISES\_ORGA\_FOR\_IN;

Ejercicio: Realiza una función que reciba como parámetro el código de un país y devuelva el número de ciudades que hay de ese país en la base de datos.

A continuación realiza un bloque de código anónimo con un cursor que me recupere los códigos de todos los países, el bloque recorre este cursor y para cada código que recupere al cursor llamara a la función de modo que al final me mostrará una frase del tipo el país de nombre tiene x ciudades.

- - (Función) - -

CREATE OR REPLACE FUNCTION NUM\_CIUDADES\_PAIS

(

PCODIGO IN NUMBER

) RETURN NUMBER AS

total number;

BEGIN

select count(\*) into total

from ciudad

where cod\_pais=pcodigo;

RETURN total;

END NUM\_CIUDADES\_PAIS;

set serveroutput on;

declare

cursor c1 is

select nombre, cod\_pais

from pais;

begin

for reg\_pais in c1 loop

dbms\_output.put\_line(reg\_pais.nombre||' tiene '||num\_ciudades\_pais(reg\_pais.cod\_pais)||' ciudades');

end loop;

end;

Modificación del ejercicio anterior para que reciba un continente

set serveroutput on;

declare

cursor c1(pconti varchar2) is

select nombre, cod\_pais

from pais

where continente like pconti;

paraconti varchar2(50);

begin

paraconti:=&Dame\_continente;

for reg\_pais in c1(paraconti) loop

dbms\_output.put\_line(reg\_pais.nombre||' tiene '||num\_ciudades\_pais(reg\_pais.cod\_pais)||' ciudades');

end loop;

end;

Ejercicio: (BD Empleados) Realiza una función que reciba como argumentos el nombre de un departamento y un número de hijos y devuelva como resultado el número de empleados que tengan ese número de hijos.

Realiza un bloque de código que pida al usuario el nombre de un departamento e invoque a la función controlando la posibilidad que pueda devolver 0, en cuyo caso mostrara ‘No hay ningún empleado con x número de hijos’ y si no ‘En el departamento x hay x empleados con x hijos’

CREATE OR REPLACE FUNCTION DEPTO\_HIJOS

(

PDEPTO IN VARCHAR2

, PHIJOS IN NUMBER

) RETURN NUMBER AS

total number;

BEGIN

select count(\*) into total

from tdepto d, temple e

where d.numde=e.numde

and nomde like pdepto

and numhi=phijos;

RETURN total;

END DEPTO\_HIJOS;

set serveroutput on;

declare

paradepto varchar2(50);

parahijos number;

totalemple number;

begin

paradepto:=&Dime\_departamento;

parahijos:=&Dime\_hijos;

(terminar)

SECUENCIAS

Ejemplo: (BD paises)

Ejercicio 1: En la base de datos de empleados, vamos a crear un procedimiento que reciba como parámetro un número de hijos y que para los empleados que tengan estos hijos o más y que tengan además comisión, aumente la comisión actual en un 2% de su comisión actual por cada hijo.

(PROCEDIMIENTO)

CREATE OR REPLACE PROCEDURE AUMENTA\_COMIS

(

PHIJOS IN NUMBER

) AS

cursor c1 is

select comis

from temple

where numhi>=phijos

and comis is not null

for update;

BEGIN

for reg\_emple in c1 loop

update temple set comis=comis+(0.02\*comis\*numhi) where current of c1;

end loop;

commit;

NULL;

END AUMENTA\_COMIS;

set serveroutput on;

declare

parahijos number;

begin

parahijos:=&Dame\_hijos;

aumenta\_comis(parahijos);

end;

Ejemplo: Crea un procedimiento similar al anterior que reciba un número de hijos y que te recupere las comisiones de los empleados que tienen ese o más número de hijos tengan o no tengan comisión.

El procedimiento aumentará la comisión en un 2% por cada hijo pero declarará una excepción para que cuando se encuentre con un empleado con misión nula lance un mensaje que diga “Empleado con comisión nula, no se puede actualizar”

(PROCEDIMIENTO)

CREATE OR REPLACE PROCEDURE AUMENTA\_COMIS\_EXCEP

(

PHIJOS IN NUMBER

) AS

cursor c1 is

select comis

from temple

where numhi>=phijos

for update;

nula exception;

BEGIN

for reg\_emple in c1 loop

if reg\_emple.comis is null then

raise nula;

else

update temple set comis=comis+(0.02\*comis\*numhi) where current of c1;

end if;

end loop;

commit;

exception

when nula then

commit;

raise\_application\_error(-20000, 'Empleado con comisión NULA. No se puede actualizar');

when others then

commit

dbms\_output.put\_line('Error' || SQLCODE || SQLERRM);

NULL;

END AUMENTA\_COMIS\_EXCEP;

select \* from temple where numhi>=3;

update temple set comis=100 where numem=110;

set serveroutput on;

declare

parahijos number;

begin

parahijos:=&Dame\_hijos;

aumenta\_comis\_excep(parahijos);

end;

Ejercicio:(BD países) Tenemos una función llamada num\_ciudades\_pais. Crea otra función que se llame num\_idiomas\_pais que reciba el código de un país y me devuelva el número de idiomas que se hablan en ese país.

A continuación crear un procedimiento que reciba el nombre de un continente y que con un cursor que recupere el nombre y el código de los países de ese continente me haga un listado con una línea de salida que sea “El país de nombre nombre\_pais tiene x ciudades y hablan z idiomas”

.

Ejercicio: Añadir un nuevo atributo a la tabla país que se llame n\_ciudades que pueda ser nulo.

alter table pais add n\_ciudades number;

Crea un procedimiento llamado rellena\_ciudades que usando un cursor de actualización y la funcion num\_ciudades\_pais, rellene esta columna (n\_ciudades)

(PROCEDIMIENTO RELLENA\_CIUDADES)

CREATE OR REPLACE PROCEDURE RELLENA\_CIUDADES AS

cursor c1 is

select cod\_pais

from pais

for update;

total number;

BEGIN

for reg\_pais in c1 loop

total:=num\_ciudades\_pais(reg\_pais.cod\_pais);

update pais set n\_ciudades=total where current of c1;

end loop;

commit;

NULL;

END RELLENA\_CIUDADES;

(EJECUTAR EN UN BLOQUE ANÓNIMO)

execute rellena\_ciudades;

Utilizando data modeler vamos a crear

producto:

pk id number

nn nombre varchar2

nn pcoste decimal(10,2)

nn pventa ‘’

nn existencias number

contiene

pk fk1 id\_producto number

pk fk2 id\_venta ‘’

nn unidades ‘’

venta:

pk id number

nn fecha date

Ejercicio: Crea un procedimiento que recibe 3 parámetros:

* El nombre de un continente
* 2 extensiones e1 y e2

El procedimiento mostrará el nombre y capital de los países que pertenecen a ese continente y cuya extensión esté comprendida entre e1 ye2.

Además contendrá una excepción que controle que e2 es mayor que e1, en caso contrario mostrará el mensaje de error “Extensiones inválidas”

(Procedimiento)

CREATE OR REPLACE PROCEDURE CONTINENTE\_EXTENSIONES

(

NOM\_CONTINENTE IN VARCHAR2

, E1 IN NUMBER

, E2 IN NUMBER

) AS

cursor c1 is

select nombre, capital

from pais

where continente like nom\_continente

and extension between e1 and e2;

error\_ext exception;

BEGIN

if e2<=e1 then

raise error\_ext;

end if;

for reg\_pais in c1 loop

dbms\_output.put\_line(reg\_pais.nombre||' '||reg\_pais.capital);

end loop;

exception

when error\_ext then

raise\_application\_error(-20000, 'Extensiones inválidas');

when others then

dbms\_output.put\_line('Error'||sqlcode||sqlerrm);

NULL;

END CONTINENTE\_EXTENSIONES;

Ejercicio: Realiza un procedimiento que recibe 2 parámetros:

* El primero será el nombre de un continente
* El segundo, las siglas de una organización

El procedimiento mostrará un listado con el nombre y la capital de los países que pertenecen a ese continente y a esa organización. En el caso de que no haya un país se mostrará la excepción “No hay países en el continente x que pertenezcan a la organización y”

(Procedimiento)

CREATE OR REPLACE PROCEDURE PAIS\_CONTI\_ORGA

(

PCONTI IN VARCHAR2

, PSIGLAS IN VARCHAR2

) AS

cursor c1 is

select p.nombre as npais, capital

from pais p, pertenece\_a pe, organizacion o

where p.cod\_pais=pe.cod\_pais

and pe.cod\_organizacion=o.cod\_organizacion

and continente like pconti

and siglas like psiglas;

total number;

error\_pais exception;

BEGIN

select count(\*) into total

from pais p, pertenece\_a pe, organizacion o

where p.cod\_pais=pe.cod\_pais

and pe.cod\_organizacion=o.cod\_organizacion

and continente like pconti

and siglas like psiglas;

if total=0 then

raise error\_pais;

end if;

exception

when error\_pais then

raise\_application\_error(-20000, 'No hay paises del continente'||pconti||' pertenecientes a la '||psiglas);

when others then

dbms\_output.put\_line('Error'||sqlcode||sqlerrm);

for reg\_pais in c1 loop

dbms\_output.put\_line(reg\_pais.npais||' '||reg\_pais.capital);

end loop;

NULL;

END PAIS\_CONTI\_ORGA;

Ejercicio: Añade un nuevo atributo a la tabla pais llamado rentapcapita definido como un decimal(20,2) y que pueda ser nulo. A Continuación crea un procedimiento llamado rellena\_renta para rellenar el atributo añadido. No recibe parámetros y rellena el atributo anterior de la base de datos como el cociente del producto interior bruto de cada país entre el número de habitantes.

ALTER TABLE pais ADD rentapcapita DECIMAL(20,2);

(PROCEDIMIENTO)

.

ejercicio: Terminar base de productos, añade 6 u 8 productos, implementa los triggers antes\_contiene y despues\_contiene y realiza un procedimiento que imprima la factura de una venta siguiendo el siguiente formato:

ID: (código\_factura)

Fecha: (fecha\_factura)

| id | nombre | punidad  (pventa) | unidades  (contiene) | precio | iva | total |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |

Total: (venta)

**CREATE** **OR** **REPLACE** **PROCEDURE** FACTURA

(

PVENTA **IN** **NUMBER**

) **AS**

**cursor** c1 **is**

**select** id\_producto, unidades

**from** contiene

**where** id\_venta=pventa;

fechaventa **date**;

nproducto **varchar2**(50);

precioventa **decimal**(20,2);

preciosiniva **decimal**(20,2);

ivap **decimal**(20,2);

totalparcial **decimal**(20,2);

totalventa **decimal**(20,2);

**BEGIN**

**select** fecha **into** fechaventa **from** venta **where** **id**=pventa;

**dbms\_output**.**put\_line**('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

**dbms\_output**.**put\_line**('ID\_VENTA: '||pventa);

**dbms\_output**.**put\_line**('FECHA DE VENTA: '||fechaventa);

**dbms\_output**.**put\_line**('Id Nombre P/u Unidades Precio IVA Total');

**for** reg\_venta **in** c1 **loop**

**select** nombre **into** nproducto **from** producto **where** **id**=reg\_venta.id\_producto;

**select** pventa **into** precioventa **from** producto **where** **id**=reg\_venta.id\_producto;

preciosiniva:=valor(precioventa,reg\_venta.unidades);

ivap:=iva(preciosiniva);

totalparcial:=preciosiniva+ivap;

**dbms\_output**.**put\_line**(reg\_venta.id\_producto||' '||nproducto||' '||precioventa||' '||reg\_venta.unidades||' '||preciosiniva||' '||ivap||' '||totalparcial);

**end loop**;

**select** total **into** totalventa **from** venta **where** **id**=pventa;

**dbms\_output**.**put\_line**('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*TOTALVENTA: '||totalventa);

**NULL**;

**END** FACTURA;

Ejercicio: La conexión países, realiza un procedimiento que reciba el nombre de un idioma y a continuación haga un listado al siguiente:

El idioma X en el continente Y lo hablan

nombre, capital de los Z

.

Obtener por continente cuantos paises hablan el idioma castellano y cual es el nombre y capital de cada uno de esto países

(Procedimiento)

**CREATE** **OR** **REPLACE** **PROCEDURE** PAISES\_IDIOMA

(

PIDIOMA **IN** **VARCHAR2**

) **AS**

**cursor** c1 **is**

**select** continente, **count**(\*) **as** total

**from** pais p, se\_habla **s**, idioma **i**

**where** p.cod\_pais=**s**.cod\_pais

**and** **s**.cod\_idioma=**i**.cod\_idioma

**and** **i**.nombre **like** pidioma

**group** **by** continente;

**cursor** c2(pconti **varchar2**) **is**

**select** p.nombre **as** npais, capital

**from** pais p, se\_habla **s**, idioma **i**

**where** p.cod\_pais=**s**.cod\_pais

**and** **s**.cod\_idioma=**i**.cod\_idioma

**and** continente **like** pconti;

**BEGIN**

**for** reg\_pais **in** c1 **loop**

**dbms\_output**.**put\_line**('En '||reg\_pais.continente||' hay '||reg\_pais.total||' paises que hablan '||pidioma);

**for** reg\_pais1 **in** c2(reg\_pais.continente) **loop**

**dbms\_output**.**put\_line**(reg\_pais1.npais||' '||reg\_pais1.capital);

**end loop**;

**end loop**;

**NULL**;

**END** PAISES\_IDIOMA;