

EJERICICIO 1

```
declare
  i number;
  CURSOR c1 is select * from RECURSO order by descripcion;
begin
  i := 0;
  FOR r_aux IN c1 LOOP
    if r_aux.nombre_formato = 'MP3' then
      update recurso set descripcion = upper(descripcion) where codigo = r_aux.codigo;
      i:=i+1;
    end if;

    if i>=2 then
      exit;
    end if;
  END LOOP;
  commit;
end;
```

Variable para contar la iteración del bucle por la que vamos y así poder salir después de la segunda

Nos devuelve los recursos ordenados por descripción, es fundamental el ORDER BY DESCRIPCION para quedarnos con los dos primeros según el orden

r_uax es una variable del tipo RECURSO%ROWTYPE donde se irán "depositando" los registros de RECURSO conforme lo vayamos iterando. Si no se declara, el sistema lo hace automáticamente (CUIDADO: SÓLO OCURRE ESTO EN EL BUCLE FOR)

Sólo contamos si es MP3, podríamos haber añadido esa condición en el where del cursor y así no poner este IF
CURSOR c1 is select * from RECURSO where nombre_formato = 'MP3' order by descripcion

Hacemos el update !!!IMPORTANTE!!! indicando el código del RECURSO que estamos iterando

Si hemos actualizado a dos recursos, entonces salimos del bucle

El commit fuera del bucle para que o se actualicen TODAS o ninguna

EJERICICIO 2

```
declare
  i number;
  CURSOR c1 is select * from RECURSO order by descripcion;
  r_aux RECURSO%ROWTYPE;

begin
  i := 0;
  Open c1;
  fetch C1 into r_aux;
  while C1%FOUND and i<=2 loop

    if r_aux.nombre_formato = 'MP3' then
      update recurso set descripcion = upper(descripcion) where codigo = r_aux.codigo;
      i:=i+1;
    end if;

    fetch C1 into r_aux;

  END LOOP;
  Close C1;
  commit;

end;
```

Variable para contar la iteración del bucle por la que vamos y así poder salir después de la segunda

Nos devuelve los recursos ordenados por descripción, es fundamental el ORDER BY DESCRIPCION para quedarnos con los dos primeros según el orden

Ahora, al no usar el FOR si necesitamos definir expresamente r_aux, ahí iremos colocando el registro en cada iteración.

Al no usar el FOR necesitamos ABRIR EL CURSOR Y HACER EL PRIMER FETCH (recuperamos primer fila), si NO recuperamos nada entonces C1%FOUND=false

Mientras hayan registros y no hayamos actualizado a dos

Solo contamos si es MP3, podríamos haber añadido esa condición en el where del cursor y así no poner este IF
CURSOR c1 is select * from RECURSO where nombre_formato = 'MP3' order by descripcion

Hacemos el update !!!IMPORTANTE!!! indicando el código del RECURSO que estamos iterando

Le pedimos otro registro
IMPORTANTÍSIMO, SI NOS OLVIDAMOS DE ESTE FECH DELTRO DEL WHILE ENTRARÍAMOS EN UN BUCLE INFINITO

No se nos olvida cerrar el cursor

El commit fuera del bucle para que o se actualicen TODAS o ninguna

EJERCICIO 3

```
create table categoria_portales
( nombre varchar2(50),
  categoría varchar2(15),
  CONSTRAINT pk_categoria_porta PRIMARY KEY ( nombre ));
```

Creamos la tabla FUERA del PL, dentro de un PL no pueden haber sentencias DDL

DECLARE

```
CURSOR c1 is select * from portales;
nporta integer;
wcat varchar2(15);
```

Variable para poner la categoría que vamos a enviar en el insert

BEGIN

```
select count(*) into nporta from portales;
if nporta = 0 then
  raise_application_error(-20001, 'Tabla PORTALES vacía');
```

Antes de recorrer el cursor, nos aseguramos que haya filas en portales tal y como pide el enunciado

```
end if;
```

```
for rpo in c1 loop
```

Al estar en un FOR no es necesario declarar rpo como variable tipo %ROWTYPE, aunque podríamos haberlo hecho

```
  if rpo.num_usu is null then
    wcat:='sin datos';
```

```
  elsif rpo.num_usu >=1 and rpo.num_usu < 51 then
    wcat:='pequeño';
```

En función de valor de num_usu ponemos un valor u otro en wcat

```
  elsif rpo.num_usu >=51 and rpo.num_usu < 100 then
    wcat:='mediano';
```

```
  else
```

```
    wcat:='grande';
```

```
  end if;
```

Hacemos el insert

```
  insert into categoria_portales(nombre,categoría) values(rpo.nombre, wcat );
```

```
end loop;
```

```
commit;
```

Commit fuera del bucle

```
END;
```

EJERCICIO 3: Solución alternativa usando excepciones para controlar el error lógico "No hay portales en la tabla PORTALES".

```
create table categoria_portales
( nombre varchar2(50),
  categoría varchar2(15), CONSTRAINT pk_categoria_porta PRIMARY KEY ( nombre ));
DECLARE
  CURSOR c1 is select * from portales;
  nporta integer;
  wcat varchar2(15);
  no_hay_portales EXCEPTION;
BEGIN
  select count(*) into nporta from portales;
  if nporta = 0 then
    RAISE no_hay_portales;
  end if;
  for rpo in c1 loop
    if rpo.num_usu is null then
      wcat:='sin datos';
    elsif rpo.num_usu >=1 and rpo.num_usu < 51 then
      wcat:='pequeño';
    elsif rpo.num_usu >=51 and rpo.num_usu < 100 then
      wcat:='mediano';
    else
      wcat:='grande';
    end if;
    insert into categoria_portales(nombre,categoría) values(rpo.nombre, wcat );
  end loop;
  commit;
EXCEPTION
  WHEN no_hay_portales THEN
    raise_application_error(-20001, 'Tabla PORTALES vacía');
END;
```

Creamos la tabla FUERA del PL, dentro de un PL no pueden haber sentencias DDL

Variable para poner la categoría que vamos a enviar en el insert

Variable para disparar la excepción si la tabla PORTALES está vacía

Antes de recorrer el cursor, nos aseguramos que haya filas en portales tal y como pide el enunciado

Al estar en un FOR no es necesario declarar rpo como variable tipo %ROWTYPE, aunque podríamos haberlo hecho

En función de valor de num_usu ponemos un valor u otro en wcat

Hacemos el insert

Commit fuera del bucle

Gestionamos la excepción definida por nosotros

EJERCICIO 4

```
create table categoria_portales
( nombre varchar2(50),
  categoría varchar2(15), CONSTRAINT pk_categoria_porta PRIMARY KEY ( nombre ));
DECLARE
  CURSOR c1 is select * from portales;
  nporta integer;
  wcat varchar2(15);
  rpo PORTALES%ROWTYPE;
BEGIN
  open c1;
  fetch C1 into rpo ;
  if C1%NOTFOUND THEN
    raise_application_error(-20001, 'Tabla PORTALES vacía');
  else
    while C1%FOUND loop
      if rpo.num_usu is null then
        wcat:='sin datos';
      elsif rpo.num_usu >=1 and rpo.num_usu < 51 then
        wcat:='pequeño';
      elsif rpo.num_usu >=51 and rpo.num_usu < 100 then
        wcat:='mediano';
      else
        wcat:='grande';
      end if;
      insert into categoria_portales(nombre,categoría) values(rpo.nombre, wcat );
      fetch C1 into rpo ;
    end loop;
    close c1;
    commit;
  end if;
END;
```

Creamos la tabla FUERA del PL, dentro de un PL no pueden haber sentencias DDL

Variable para poner la categoría que vamos a enviar en el insert

Al ser con while, debemos definirla

Podemos usar la variable C1%NOTFOUND para saber si hay filas en portales en vez de hacer un select count

En función de valor de num_usu ponemos un valor u otro en wcat

Hacemos el insert

OJO!!!! Pedimos la siguiente, sino BUCLE INFINITO

No se nos olvida cerrar el cursor

Commit fuera del bucle