

Tarea de clase 2024-05-28 Programación de bases de datos

Se entregará por cada apartado el bloque de código pl/sql realizado (ya sea anónimo, procedimiento almacenado,...)

Nota: Todos los apartados se puntuarán como: ok (puntuando la nota especificada) en el caso que cumplan exactamente todo lo establecido en el enunciado, o como mal puntuando 0ptos

Nota2: Si algún ejercicio nombra tablas sin especificarse la base de datos, es porque está pensado para la base de datos por defecto: northwind o porque no es importante la base de datos donde se realice

Ejercicio 1) (1pto) Apartado único: crear una función/procedimiento (lo que consideres más apropiado de las dos opciones): cantidad_empleados() que ejecute una consulta sobre la tabla employees y devuelva la cantidad de empleados que tiene la empresa

Ayuda. Si ejecutamos:
`select cantidad_empleados()
from dual;`

Se espera que muestre: 9

Ejercicio 2) (1pto) Apartado único: Crear una función/procedimiento (lo que consideres más apropiado de las dos opciones) llamada: precio_medio_categoria() que reciba un único parámetro: nombrecategoria y devuelva el precio medio de los productos para esa categoría (obtendrá la información de la tabla categories)

Ayuda. Si ejecutamos:
`select precio_medio_categoria('Beverages')
from dual;`

debe mostrar: 37,97912

Ejercicio 3) (total: 3ptos)

a) (0.75ptos) Crear un procedimiento almacenado: pedidos_entre() que reciba dos fechas y muestre en pantalla (mediante uno o varios dbms_output.put_line) el id de pedido (orderid) y la fecha, de cada pedido realizado entre las dos fechas ordenados de fecha más antigua a fecha más reciente

b) (0.75ptos) adicionalmente a lo anterior, da igual el orden de las dos fechas

Esto es:

```
call pedidos_entre('29/05/97','01/07/97');
```

muestra los mismos resultados que:

```
call pedidos_entre('01/07/97', '29/05/97');
```

c) (0.75ptos) adicionalmente al apartado a), si el pedido es del cliente con companynome: "Wartian Herkku" entonces también muestra su nombre: companynome , su dirección (address) y teléfono (phone)
Esos datos: dirección, teléfono, deberán obtenerse como consulta en la base de datos. No será válido que se escriba directamente esos datos como hardcode (esto es, estará mal si se escribe directamente en el código: Torikatu 38 y 981-443655 u otro dato almacenado en la base de datos. Lo único almacenado en la base de datos que se puede poner directamente en el código es: "Wartian Herkku"). Un poco más abajo se puede ver una salida esperada parecida (se acomoda exactamente al apartado d), que es una ampliación del actual apartado)

d) (0.75ptos) adicionalmente al apartado c) muestra la cantidad de pedidos obtenidos y los que son de Wartian Herkku siguiendo el estilo establecido más abajo

Si hacemos:

```
call pedidos_entre('29/05/97','01/07/97');
```

La salida esperada es:

```
10552 29/05/97
10554 30/05/97
10553 30/05/97 Wartian Herkku Torikatu 38 981-443655
10555 02/06/97
10556 03/06/97
10557 03/06/97
10558 04/06/97
10559 05/06/97
10560 06/06/97
10561 06/06/97
10562 09/06/97
10563 10/06/97
10564 10/06/97
10565 11/06/97
10567 12/06/97
10566 12/06/97
10568 13/06/97
10569 16/06/97
10570 17/06/97
10571 17/06/97
10572 18/06/97
10573 19/06/97
10574 19/06/97
10575 20/06/97
10576 23/06/97
10577 23/06/97
10578 24/06/97
10579 25/06/97
```

10581 26/06/97
10580 26/06/97
10582 27/06/97
10584 30/06/97
10583 30/06/97 Wartian Herkku Torikatu 38 981-443655
10585 01/07/97
son 34 en este periodo. De ellos son de Wartian Herkku: 2

Ejercicio 4) (total: 3ptos)

primero crear una tabla llamada: BACKUP con los campos:
FECHA, DML, PRODUCTID, PRODUCTNAMEOLD, PRODUCTNAMENew, UNITPRICEOLD, UNITPRICENew

donde PRODUCTNAMENew Y PRODUCTNAMEOLD son del mismo tipo que
PRODUCTS.PRODUCTNAME.
UNITPRICEOLD y UNITPRICENew son del mismo tipo que PRODUCTS.UNITPRICE
FECHA es de tipo DATE

Como se puede encontrar en los apuntes, con SYSDATE se obtiene la fecha actual. Se quiere responder a
inserciones, modificaciones y borrado de la tabla PRODUCTS

a) (0.75pts) si se inserta un producto en la tabla PRODUCTS con los siguientes datos para
productid, productname, unitprice:
100, 'almendras', 40

se debe generar automáticamente una nueva fila en la tabla BACKUP donde se registre en la columna fecha la
fecha en ese momento, en PRODUCTID: 100, en PRODUCTNAMENew: 'almendras', en
PRODUCTPRICENew: 40

en la columna DML: 'insert' y: null en los otros campos: PRODUCTNAMEOLD, UNITPRICEOLD

Veamos esos campos de nuevo:

DML, PRODUCTID, PRODUCTNAMEOLD, PRODUCTNAMENew, UNITPRICEOLD, UNITPRICENew
'update', 100, 'almendras', 'pasas', 80, 40

Nota importante: tener en cuenta que en northwind hay autoincrementales (en oracle están puestos mediante
sequence y trigger), así que cuando se inserta una nueva fila en la tabla PRODUCTS no hace falta poner
PRODUCTID. Lo solicitado debe ser válido para casos en el que el insert que se ejecute en la tabla products se
esté resolviendo con autoincremental

b) (0.25ptos) si se modifica un producto ya existente en la tabla PRODUCTS con los siguientes datos para PRODUCTID, PRODUCTNAME, UNITPRICE:

100, 'pasas', 80

cuando en la tabla PRODUCTS en esos campos estaba:

100, 'almendras', 40

se agregará automáticamente una nueva fila en la tabla BACKUP con la fecha actual en el campo FECHA y en el resto de campos:

DML, PRODUCTID, PRODUCTNAMEOLD, PRODUCTNAMENew, UNITPRICEOLD, UNITPRICENew
'update', 100, 'almendras', 'pasas', 80, 40

c) (0.25ptos) si se borra un producto ya existente de la tabla PRODUCTS, se agregará automáticamente una nueva fila en la tabla BACKUP con la fecha actual en el campo FECHA y para el resto de campos (supongamos que en la tabla PRODUCTS había una fila que decía lo siguiente para los campos: productid, productname y unitprice:

productid, productname, unitprice

100, 'pasas', 80

)

nos quedaría:

DML, PRODUCTID, PRODUCTNAMEOLD, PRODUCTNAMENew, UNITPRICEOLD, UNITPRICENew
'delete', 100, 'pasas', NULL, 80, NULL

d) (0.75ptos) Si se modifica un precio (UNITPRICE) en la tabla PRODUCTS y se trata de poner en negativo se lanzará una excepción que impida la modificación

e) (1pto) crear un procedimiento llamado: reconstruir_products(fechadada) que permita reconstruir lo que había en los campos: PRODUCTID, PRODUCTNAME, UNITPRICE en una fecha dada tomando la información almacenada en la tabla: backup

Para esto, previamente crearemos una tabla

PRODUCTS_REBUILD(PRODUCTID,PRODUCTNAME,UNITPRICE) con los tipos de datos correspondientes de la tabla PRODUCTS

Por ejemplo, supongamos que tenemos en la tabla backup:

FECHA, DML, PRODUCTID, PRODUCTNAMEOLD, PRODUCTNAMENew, UNITPRICEOLD, UNITPRICENew
'15/01/2024', 'insert', 100, NULL, 'almendras', NULL, 40
'16/01/2024', 'update', 100, 'almendras', 'almendras', 40, 60
'17/01/2024', 'update', 100, 'almendras', 'pasas', 60, 60
'17/01/2024', 'insert', 110, NULL, 'orejones', NULL, 90
'18/01/2024', 'delete', 100, 'pasas', NULL, 60, NULL

y ejecutamos:

call reconstruir_products('17/01/2024')

debe aparecer almacenado en PRODUCTS_REBUILD:

PRODUCTID, PRODUCTNAME, UNITPRICE

100, 'pasas', 60

110, 'orejones', 90

Nota: Debe ser válido para múltiples ejecuciones del procedimiento. Esto implica que el procedimiento debe borrar el contenido (no la estructura) de la tabla PRODUCTS_REBUILD para que no tenga datos de la ejecución anterior

Ejercicio 5) (total: 2ptos)

Nota: Este ejercicio tiene dos versiones (apartados a) y b)) si el apartado b) entregado es correcto no hace falta presentar el apartado a), esto es: se puntuará 2ptos por el ejercicio. Si no se consigue el apartado b) entonces aún se puede puntuar 1pto si se consigue entregar un apartado a) correcto

a) (1pto) Crear un bloque pl/sql anónimo que muestre un informe basado en la tabla employees donde vaya estableciendo por cada: "TITLE" diferente las personas que lo tienen:

Sales Representative:

Nancy Davolio
Janet Leverling
Margaret Peacock
Michael Suyama
Robert King
Anne Dodsworth

Sales Manager:

Steven Buchanan

Inside Sales Coordinator:

Laura Callahan

Vice President, Sales:

Andrew Fuller

b) (2pto) Crear bloque pl/sql anónimo que es una versión avanzada del anterior. Ahora mostrará las cantidades de personas en la empresas y por cada title, además de pedidos gestionados por cada empleado

total empleados: 9

Sales Representative: 6 empleados

Nancy Davolio: 123 pedidos

Janet Leverling: 127 pedidos

Margaret Peacock: 156 pedidos

Michael Suyama: 67 pedidos

Robert King: 72 pedidos

Anne Dodsworth: 43 pedidos

Sales Manager: 1 empleados

Steven Buchanan: 42 pedidos

Inside Sales Coordinator: 1 empleados

Laura Callahan: 104 pedidos

Vice President, Sales: 1 empleados

Andrew Fuller: 96 pedidos