

Examen-PLSQL-y-Triggers-Julio-20...



Juandf03



Administración de Bases de Datos



3º Grado en Ingeniería Informática



**Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga**



Estamos de
Aniversario

De la universidad al
mercado laboral:
especialízate con los posgrados
de EOI y marca la diferencia.



EOI Escuela de
organización
industrial



saber más

¡TUS NEURONAS PIDEN DÓMINO'S!

3 MEDIANAS

2 INGREDIENTES

7'25€
C/U
DOMICILIO
COD:DEL725



**PIDE
AQUÍ**

Domino's



LENGUAJES Y
CIENCIAS DE LA
COMPUTACIÓN
UNIVERSIDAD DE MÁLAGA

ADMINISTRACIÓN DE BASES DE DATOS.

12 DE JULIO DE 2024

PL/SQL Y TRIGGERS. VALOR DE LA PRUEBA: 3.5 PUNTOS

NOTA: ES MUY IMPORTANTE que siga exactamente la especificación de este ejercicio y que llame a cada objeto **EXACTAMENTE** como se pide.

- Para comenzar el examen conéctese a la dirección: <https://docenciabd.lcc.uma.es/>
- Utiliza el usuario que comienza por UBDXXX para conectarte a la aplicación. La aplicación suministrará un usuario y contraseña con la que debes conectarte a Apolo en Afrodita.lcc.uma.es.
- Tenga cuidado con las comillas si va a copiar y pegar del enunciado, puesto que a veces el código ASCII del pdf no es el mismo que se utiliza en SQL.
- **NO OLVIDE** finalizar las transacciones.

Encontrará en el esquema que se le ha proporcionado una copia de algunas de las tablas con las que ha hecho el trabajo en grupo.

Además, ya se ha creado una tabla denominada `CONTROL_ENTRENA` con los atributos

```
Fecha DATE NOT NULL,  
Usuario VARCHAR2(35),  
Entrenador_nuevo Number NOT NULL,  
Cliente_nuevo Number not null,  
Entrenador_Antes Number,  
Cliente_Antes Number
```

Ejercicio 1 (1 puntos). Crea un Trigger denominado `TR_ENTRENA` que cuando se modifique un registro de/en la tabla `ENTRENA`, inserte un registro en `CONTROL_ENTRENA` con la fecha del sistema, el usuario que ha realizado la operación (el `USER`), el código del entrenador nuevo, del cliente nuevo y los valores que tenían antes ambos atributos.

Crear el siguiente paquete:

```
CREATE OR REPLACE PACKAGE PK_LIFEFIT AS  
  
PROCEDURE P_CREA_PAREJAS (p_centro centro.id%type);  
  
PROCEDURE P_CREA_PLAN (p_entrenador ENTRENADOR.ID%TYPE,  
p_cliente CLIENTE.ID%TYPE, INICIO DATE, FIN DATE);  
  
PROCEDURE P_SESIONES_CLIENTES (p_cliente CLIENTE.ID%TYPE);  
  
END;
```

Consulta condiciones,
suplementos y tiendas
adheridas en la web.
Se podrá solicitar el
carnet de estudiante
para aplicar las pro-
mociones. Exclusivo
para menores de 25
años. Válida hasta el
30/06/2025.

WUOLAH



Crea el BODY del paquete con:

Ejercicio 2 (0.75 puntos). Crea en el paquete un procedimiento denominado `P_CREA_PAREJAS` (`p_centro centro.id%type`); que inserta en la tabla `ENTRENA` todas las posibles parejas de entrenador y cliente que pertenecen al mismo centro. Así, por ejemplo, si `p_centro` vale 50 y en el centro 50 están los entrenadores 11 y 12 y los clientes 30 y 40, se generan las filas: 11, 30; 11, 40; 12, 30 y 12, 40 en la tabla `ENTRENA`. Si el centro no se encuentra en la tabla de centros se deberá elevar la excepción `ESC.PK_PRUEBA_EXAMEN.CENTRO_NO_EXISTE` declarado en el paquete correspondiente del usuario `ESC`.

Ejercicio 3 (0.75 puntos). Crea en el paquete un procedimiento denominado `P_CREA_PLAN` (`p_entrenador ENTRENADOR.ID%TYPE`, `p_cliente CLIENTE.ID%TYPE`, `INICIO DATE`, `FIN DATE`); que inserta en la tabla `PLAN` los datos introducidos como parámetros y en la tabla `SESION` tantas filas como días transcurran entre `INICIO` y `FIN`, incrementando en un día cada una de las filas. Se supone que cada sesión dura una hora. Así por ejemplo si se llama con `P_CREA_PLAN` (11, 30, `to_date ('12/07/2024 09:00', 'dd/mm/yyyy hh24:mi')`, `to_date ('11/07/2025 09:00', 'dd/mm/yyyy hh24:mi')`);

Crearé una sola fila en `PLAN` y 365 filas en la tabla `SESION` con los valores de los días consecutivos entre las 2 fechas y poniendo como fecha de fin la de inicio más una hora.

Ejercicio 4 (1 puntos). Crea en el paquete un procedimiento denominado `P_SESIONES_CLIENTES` (`p_cliente CLIENTE.ID%TYPE`) que crea una vista denominada `V_SESIONES_pcliente` siendo `pcliente` el código del cliente pasado como parámetro, con todas las sesiones del cliente. Así, por ejemplo, si el código es **13**, la vista será:

```
create view v_sesiones_13 (e_nombre, e_apellidos, c_nombre,  
c_apellidos, inicio, fin) as
```

...

```
where cliente.id = 13;
```

NOTA: Se recomienda mostrar por `DBMS_OUTPUT` el código de la sentencia a ejecutar antes de la llamada con `EXECUTE IMMEDIATE` para asegurarse de que se ha construido bien. Ejemplo:

```
Sentencia := 'CREATE VIEW v_sesiones_'||p_cliente||'(e_nombre,  
e_apellidos, c_nombre, c_apellidos, inicio, fin) AS ..... ';
```

```
DBMS_OUTPUT.PUT_LINE (Sentencia);
```

```
EXECUTE IMMEDIATE Sentencia;
```