

Enunciado-Junio-2016-2017.pdf



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

Menú King Ahorro

Molar sin gastar mucho,
mola mucho más.



Menú Cheeseburger



Menú Crispy Chicken



Menú Cheddar Lover
Doble Cheeseburger

PÍDE YA

UNIVERSIDAD DE MÁLAGA

ETSI Informática
Aulas TIC | Programación Docente

campus virtual | enseñanza virtual y laboratorios tecnológicos

Contacta | Idioma | Salir

CV ▶ ETSI Informática ▶ Mis asignaturas en este Centro ▶ Curso académico 2018-2019 ▶ Grado en Ingeniería Informática ▶ Administración de Bases de Datos (2018-19, Todos ... ▶ Tema inicial ▶ Prueba PL/SQL Junio 2017

Identificación

Hola Antonio
antonioalcaidefacultad@gmail.com
Mis últimas entradas

Salir

Navegación

ETSI Informática

- Mi área personal
- Panel de mensajes personales y notificaciones
- Mi información personal
- Asignatura actual
 - Administración de Bases de Datos (2018-19, Todos ...
 - Participantes
 - Insignias
 - Tema inicial
 - Prueba PL/SQL Junio 2017
 - Mis asignaturas en este Centro
 - Asignaturas

Administración

Administración de la asignatura

Mis ajustes de información y preferencias personales

Docencia Administración de Bases de Datos ETSI Informática
Universidad de Málaga

Administración de Bases de Datos
Prueba de PL/SQL

Enlaces documentación:

- Manual referencia PL/SQL
- Manual referencia SQL
- Manual referencia Database 10g-Diccionario Datos-

Cree las siguientes tablas
create table usuarios (Id Number primary key,
Nombre VARCHAR2(50) not null,
Apellidos VARCHAR2(50) not null,
Usuario_Oracle VARCHAR2(12));

create table preguntas (Id Number primary key,
enunciado VARCHAR2(100) not null);

create table respuestas (Id Number primary key,
id_pregunta number references preguntas (id),
id_usuario references usuarios (id),
respuesta VARCHAR2(100) , constraint uk_pregunta_usuario unique (id_pregunta, id_usuario));

Cree la siguiente Vista:
CREATE OR REPLACE VIEW V_RESPUESTAS(id_pregunta,enunciado,respuesta) AS
SELECT p.Id, p.Enunciado, null respuesta
from preguntas p where p.Id not in
(select r.id_pregunta from respuestas r join usuarios u on u.id = r.id_usuario
where user = u.usuario_oracle) --preguntas no contestadas por el user
union
(SELECT p.Id, p.Enunciado, r.respuesta
from preguntas p join respuestas r on r.id_pregunta = p.id
join usuarios u on u.id = r.id_usuario where user = u.usuario_oracle) --si contestadas

Realice los siguientes pasos (1 punto):

- Cree una secuencia SEC_RESPUESTAS que comience en 1 y avance de 1 en 1
- Cree un trigger TR_RESPUESTAS que cada vez que se modifique V_RESPUESTAS compruebe si el usuario ya ha contestado a la pregunta. Si no la ha hecho, debe insertar una fila en la tabla RESPUESTAS utilizando como id_respuestas el siguiente valor de SEC_RESPUESTAS. Si ya lo había hecho, debe modificar la respuesta.

Un ejemplo de actualización de la vista sería,
update v_respuestas set respuesta = 'Mi respuesta' where id_pregunta = 1;
y el trigger tendría que insertar la fila nueva en RESPUESTAS (si no existe) o modificar la fila correspondiente.
Si el usuario que está intentando hacer la modificación no existe en la tabla usuarios, se debe elevar una excepción.

Crea una tabla llamada ERRORES con los atributos usuario VARCHAR2(50) y descripcion VARCHAR2(100)

Cree el siguiente procedimiento (1 Punto):

- PR_CREA_VISTAS_RESPUESTAS que crea una vista a cada usuario con el formato V_USUARIO_RESPUESTA

El procedimiento recorre la tabla usuarios y crea una vista para cada uno con la sentencia: SELECT p.Id, p.Enunciado, r.Respuesta from preguntas p left outer join respuestas r on r.id_pregunta = p.id where r.id_usuario = ID_USUARIO
Es decir, si en la tabla hay 2 filas con los usuarios ESC, con id 1 y PEPE, con id 2, crearía 2 vistas:
V_ESC_RESPUESTA AS SELECT p.Id, p.Enunciado, r.Respuesta from preguntas p left outer join respuestas r on r.id_pregunta = p.id where r.id_usuario = 1.
y
V_PEPE_RESPUESTA AS SELECT p.Id, p.Enunciado, r.Respuesta from preguntas p left outer join respuestas r on r.id_pregunta = p.id where r.id_usuario = 2.

Si la vista no se puede crear, se inserta en errores el usuario de la fila correspondiente, y el mensaje de Oracle (SQLERRM), pero se sigue procesando el cursor.

Junio 2017 E. Soler

Última modificación: lunes, 27 de mayo de 2019, 09:23

Universidad de Málaga · Avda. Cervantes, 2. 29071 MÁLAGA · Tel. 952131000 · info@uma.es

© Todos los derechos reservados

WUOLAH