

Examen-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 Double Cheeseburger

PÍDE YA

--1

```
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));
```

--2

```
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
```

--3

--Crear secuencia

--Crear un trigger

```
CREATE OR REPLACE TRIGGER TR_RESPUESTAS
INSTEAD OF UPDATE ON V_RESPUESTAS
FOR EACH ROW
DECLARE
```

```
id_usuario NUMBER;
resp_anterior VARCHAR2(100);
```

BEGIN

```
SELECT ID INTO id_usuario FROM USUARIOS WHERE USUARIO_ORACLE = USER;
BEGIN
```

```
SELECT R.RESPUESTA INTO resp_anterior FROM RESPUESTAS R, PREGUNTAS P
WHERE R.ID_PREGUNTA = P.ID;
```

```
UPDATE RESPUESTAS SET RESPUESTA = :NEW.RESPUESTA WHERE
ID_PREGUNTA = :OLD.ID_PREGUNTA;
```

```
EXCEPTION WHEN no_data_found THEN
```

```
--si no hay respuesta
```

```
INSERT INTO RESPUESTAS VALUES (SEC_RESPUESTAS.NEXTVAL,
:OLD.ID_PREGUNTA, id_usuario, :NEW.RESPUESTA);
END;
```

```
EXCEPTION WHEN no_data_found THEN
```

```
raise_application_error(-20001, 'No se ha encontrado el usuario');
```

WUOLAH

```

COMMIT;

END TR_RESPUESTAS;

UPDATE V_RESPUESTAS SET respuesta = 'Tengo 21 años' WHERE id_pregunta = 1;

--4
--Crear un procedimiento
create or replace PROCEDURE PR_CREA_VISTAS_RESPUESTAS AS
  CURSOR c_usuarios IS SELECT * FROM USUARIOS;
  err_msg VARCHAR2(100);
  sentencia VARCHAR2(1000);
BEGIN

  FOR var_c IN c_usuarios LOOP
    sentencia := 'CREATE OR REPLACE VIEW V_' || var_c.nombre || '_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 = ' || var_c.id;
    dbms_output.put_line(sentencia);
    BEGIN

      EXECUTE IMMEDIATE sentencia;
      COMMIT;

    EXCEPTION
      when others then
        err_msg := substr(sqlerrm, 1, 100);
        insert into errores values (var_c.usuario_oracle, err_msg);

    END;
  END LOOP;

END PR_CREA_VISTAS_RESPUESTAS;

BEGIN
  PR_CREA_VISTAS_RESPUESTAS;
END;
/

```