**Práctica 3. Control Semántico de los Datos**

Valor 60 puntos (debes de documentar cada uno de los apartados e integrar los conceptos necesarios para el desarrollo de la practica (por cada día de retraso tendrá un valor de menos 2 puntos))

**Fecha**: 26 de julio de 2017 Fecha de entrega 28 y 31 de julio 2017  
**Conceptos**: Trigger, View (creación, consultas, inserciones, actualizaciones y borrados)

**OBJETIVOS**

* Crear triggers para realizar acciones ante eventos
* Crearse Vistas sobre la Base de Datos Empresa.
* Realizar consultas utilizando las Vistas.
* Calibrar el efecto de diversas actualizaciones o borrados hechos sobre vistas.

**Nota:** Para esta práctica utilizaremos la base de datos de empleados que creamos en la practica 1. Se adjunta el script [practica1.sql](http://www.it.uc3m.es/jcano/clases/basessdd/practicas/4/practica1.sql) y ejecutarlo para crear tanto las tablas como las tuplas.

**EJERCICIO 1. Triggers (valor 15 puntos)**

Un *trigger* es un procedimiento escrito en PL/SQL que se ejecuta automáticamente sobre la base de datos cuando se produce un evento de manipulación de datos (INSERT, UPDATE o DELETE). Sigue la siguiente estructura:

CREATE [OR REPLACE] TRIGGER trigger\_name

{BEFORE | AFTER | INSTEAD OF} trigger\_event

ON table\_name

[FOR EACH ROW [WHEN trigger\_condition]]

BEGIN

trigger\_body

END trigger\_name;

donde:

* OR REPLACE indica si el trigger reemplazará un trigger que ya se haya creado con el mismo nombre.
* trigger\_name es el nombre que se le dará al trigger.
* BEFORE, AFTER y INSTEAD OF indica en qué momento se debe ejecutar el trigger: antes de la manipulación de la base de datos que hace el evento, después, o en lugar de, respectivamente.
* trigger\_event indica el evento que causa la ejecución del trigger.
* table\_name indica la tabla a la que el trigger referencia.
* FOR EACH ROW indica si el trigger se ejecutará por cada tupla modificada (si hay varias tuplas indicadas) o una única vez.
* trigger\_condition indica una condición que se debe cumplir además del evento para que se ejecute el trigger.
* trigger\_body contiene las sentencias SQL y PL/SQL que llevan a cabo la tarea del trigger.

A continuación, se muestra un ejemplo de trigger que se activará en el caso de que insertemos una nueva tupla en la tabla Proyecto:

CREATE OR REPLACE TRIGGER trnuevoproyecto

BEFORE INSERT

ON Proyecto

DECLARE

contador integer;

BEGIN

select count(\*) into contador from proyecto where numerop= 50;

if (contador = 0) then

insert into proyecto values('PruebaTrigger',50,'Stafford',1);

end if;

END trnuevoproyecto;

/

**Ejercicios**

1. Carga el trigger en tu base de datos e inserta un nuevo dato en la tabla Proyecto. ¿Qué es lo que sucede? Modifica el trigger anterior para que funcione correctamente.
2. Crea un nuevo trigger llamado creafamiliar que, en el caso de que se inserte un nuevo familiar dependiente y no exista la persona de la que depende en la base de datos, cree una nueva entrada con los siguientes datos:

* Nombre: Antonio
* Apellidos: Pérez
* Número de Seguridad Social: xxx (el que aparezca al insertar la tupla del dependiente)

**EJERCICIO 2. Creación de Vistas (valor 15 puntos)**

En este ejercicio vamos a crear **Vistas que simulen**el diseño distribuido de la Base de Datos Empleados.

Dividimos la Base de Datos en tres sitios:

1. Sitio 1 (central)  
   Para el departamento 1 (de Dirección).
2. Sitio 2 (investigación)  
   Para el departamento 5 (de Investigación).
3. Sitio 3 (administración)  
   Para el departamento 4 (de Administración).

El objetivo inicial es crearse una vista de cada una de las tablas del **departamento 5** y del **departamento 4** con los siguientes nombres:

* Para el **departamento nº 5**:

**EMPD5(NOMBRE,INC, APELLIDO, NSS, SALARIO, NSS\_SUPERV, ND) para empleados del Departamento nº5**

**DP5(NOMBRED, NUMEROD, NSS\_JEFE, FECHA\_INICIO\_JEFE) para el Departamento nº5**

**LOCS\_DEPT5(NUMEROD, LOCALIZACIOND) para el Departamento nº5**

**PROYS5 (NOMBREP, NUMEROP, LOCALIZACIONP, NUMD) para proyectos del Departamento nº5**

**TRABAJA\_EN5(NSSE, NP, HORAS) para empleados y/o proyectos del Departamento nº5**

* Para el **departamento nº 4**:

**EMPD4(NOMBRE,INC, APELLIDO, NSS, SALARIO, NSS\_SUPERV, ND) para empleados del Departamento nº4**

**DP4(NOMBRED, NUMEROD, NSS\_JEFE, FECHA\_INICIO\_JEFE) para el Departamento nº4**

**LOCS\_DEPT4(NUMEROD, LOCALIZACIOND) para el Departamento nº4**

**PROYS4 (NOMBREP, NUMEROP, LOCALIZACIONP, NUMD) para proyectos del Departamento nº4**

**TRABAJA\_EN4(NSSE, NP, HORAS) para empleados y/o proyectos del Departamento nº4**

Se dará la definición de la primera de las vistas:

**CREATE VIEW EMPD5(NOMBRE,INC,APELLIDO,NSS,SALARIO,NSS\_SUPERV,ND)**

**AS SELECT NOMBRE,INC,APELLIDO,NSS,SALARIO,NSS\_SUPERV,ND**

**FROM EMPLEADO**

**WHERE ND=5;**

Nótese que para **TRABAJA\_EN5** y **TRABAJA\_EN4**, deberemos hacer uso de la **unión de selecciones**, de forma que se refleje la siguiente tabla:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | **PROYECTO** | | |
| **NUMD = 5** | **NUMD= 4** | **NUMD= 1** |
| **EMPLEADO** | **ND= 5** | **G1: v** | **G2: v x** | **G3: v** |
| **ND= 4** | **G4: v x** | **G5:   x** | **G6: x** |
| **ND= 1** | **G7: v** | **G8:   x** | **G9:** |

Siendo:

* **Sitio 2 (ND=5): v**
* **Sitio 3 (ND=4): x**

De forma que se debe hacer la unión de consultas con las siguientes condiciones:

* Para G1:

**C1= C AND (NP IN (SELECT NUMEROP FROM PROYECTO WHERE NUMD=5))**

* Para G2:

**C2= C AND (NP IN (SELECT NUMEROP FROM PROYECTO WHERE NUMD=4))**

* Para G3:

**C3= C AND (NP IN (SELECT NUMEROP FROM PROYECTO WHERE NUMD=1))**

donde **C= NSSE IN (SELECT NSS FROM EMPLEADO WHERE ND=5)**

* Para G4:
  + **C4= C' AND (NP IN (SELECT NUMEROP FROM PROYECTO WHERE NUMD=5))**
* Para G5:
  + **C5= C' AND (NP IN (SELECT NUMEROP FROM PROYECTO WHERE NUMD=4))**
* Para G6:
  + **C6= C' AND (NP IN (SELECT NUMEROP FROM PROYECTO WHERE NUMD=1))**

donde **C'= NSSE IN (SELECT NSS FROM EMPLEADO WHERE ND=4)**

* Para G7:
  + **C7= C'' AND (NP IN (SELECT NUMEROP FROM PROYECTO WHERE NUMD=5))**
* Para G8:
  + **C8= C'' AND (NP IN (SELECT NUMEROP FROM PROYECTO WHERE NUMD=4))**
* Para G9:
  + **C9= C'' AND (NP IN (SELECT NUMEROP FROM PROYECTO WHERE NUMD=1))**

donde **C''= NSSE IN (SELECT NSS FROM EMPLEADO WHERE ND=1)**

**EJERCICIO 3. CONSULTA SOBRE LAS VISTAS (valor 15 puntos)**

|  |
| --- |
| **Apartado 1. Visualización de todos los datos de cada una de las Vistas**  Una vez generadas las vistas, vamos a comprobar que los datos que se hayan en ella se corresponden con los datos vistos en clase. Para ello consultemos todos los campos de cada una de las relaciones (fácil). Como muestra, se les va a dar el resultado de las consultas sobre las tablas más complicadas.  **SQL> SELECT \* FROM TRABAJA\_EN5;**  **NSSE NP HORAS**  **--------- ---------- ----------**  **123456789 1 32,5**  **123456789 2 7,5**  **333445555 2 10**  **333445555 3 10**  **333445555 10 10**  **333445555 20 10**  **453453453 1 20**  **453453453 2 20**  **666884444 3 40**  **9 filas seleccionadas.**  **SQL> SELECT \* FROM TRABAJA\_EN4;**  **NSSE NP HORAS**  **--------- ---------- ----------**  **987654321 20 15**  **987654321 30 20**  **987987987 10 35**  **987987987 30 5**  **999887777 10 10**  **999887777 30 30**  **333445555 10 10**  **7 filas seleccionadas.**  **Apartado 2. Composición de consultas utilizando sólo Vistas**  ¿Seríamos capaces de utilizar sólo vistas para diseñar las siguientes consultas?  **Consulta 1**  ***Recupere los nombres y apellidos de los jefes de los empleados departamento 5***  Resultado de la consulta:  **NOMBRE APELLIDO**  --------------- ---------------  Franklin Wong  **Consulta 2**  ***Recupere los nombres y apellidos de todos los empleados que trabajen para un proyecto del departamento 4***  **RESULTADO CORREGIDO** Resultado de la consulta:  **NOMBRE APELLIDO**  **--------------- ---------------**  Ahmad Jabbar  Alicia Zelaya  Franklin Wong  Jennifer Wallace  **Consulta 3**  ***¿Cuánto se gasta el departamento 4 en los salarios de sus empleados?***  Resultado de la consulta:  **SUM(SALARIO)**  ------------  93000  **Consulta 4**  ***¿Cuántas horas se invierten en los proyectos realizados en el departamento 5?***  Resultado de la consulta:  **SUM(HORAS)**  **----------**  **140**  **Nótese que**los proyectos realizados en el departamento 5, no coincide exactamente con los proyectos de la tabla **TRABAJA\_EN5**, ya que esta tabla contiene también aquellos proyectos en los que trabajan empleados del departamento 5, proyectos que puede que no se elaboren en el departamento 5.  **EJERCICIO 4. INSERCIONES, ACTUALIZACIONES Y BORRADOS DESDE LAS VISTAS (valor 15 puntos)**  Ahora vamos a demostrar finalmente que se puede insertar y actualizar sobre algunas de las Vistas creadas en esta práctica. Apartado 1. Inserciones, actualizaciones y borrados sobre Vistas que implican una tabla. En nuestro caso son **EMPD5**, **DP5**, **LOCS\_DEPT5**, **PROYS5** y **EMPD4**, **DP4**, **LOCS\_DEPT5**, **PROYS5.**Sobre estas tablas, realizar lo siguiente:   * Insertar tuplas en el rango **NSS=[100000000..200000000],** y **NUMEROP=[40..50]** * Actualizar el valor de algunos de los atributos de las mismas. * Borrarlas.   Comprobar con selecciones que en todos los casos, estas tres operaciones son correctas cuando se hacen sobre vistas de una sola tabla. Apartado 2. Inserciones, actualizaciones y borrados sobre Vistas que implican dos tablas (o más). En nuestro caso tenemos **TRABAJA\_EN5** y **TRABAJA\_EN4.**Sobre estas tablas, realizar lo siguiente:   * Insertar tuplas en el rango **NSS=[100000000..200000000]** y **NUMEROP=[40..50]**. * Actualizar el valor de algunos de los atributos de las mismas. * Borrarlas.   Comprobar con selecciones que, en todos los casos, estas tres operaciones son correctas cuando se hacen sobre vistas de una sola tabla. En concreto, contestemos a estas preguntas.   1. ¿objetivo de Oracle como gestor de base de datos y tareas que se pueden realizar? 2. Características de Oracle 3. ¿Qué es lo que deja y qué no? ¿Por qué? 4. ¿Dan resultados correctos? Investigar sus tablas base, para ver cómo les afecta las inserciones en las Vistas a estas tablas. |
|  |
|  |