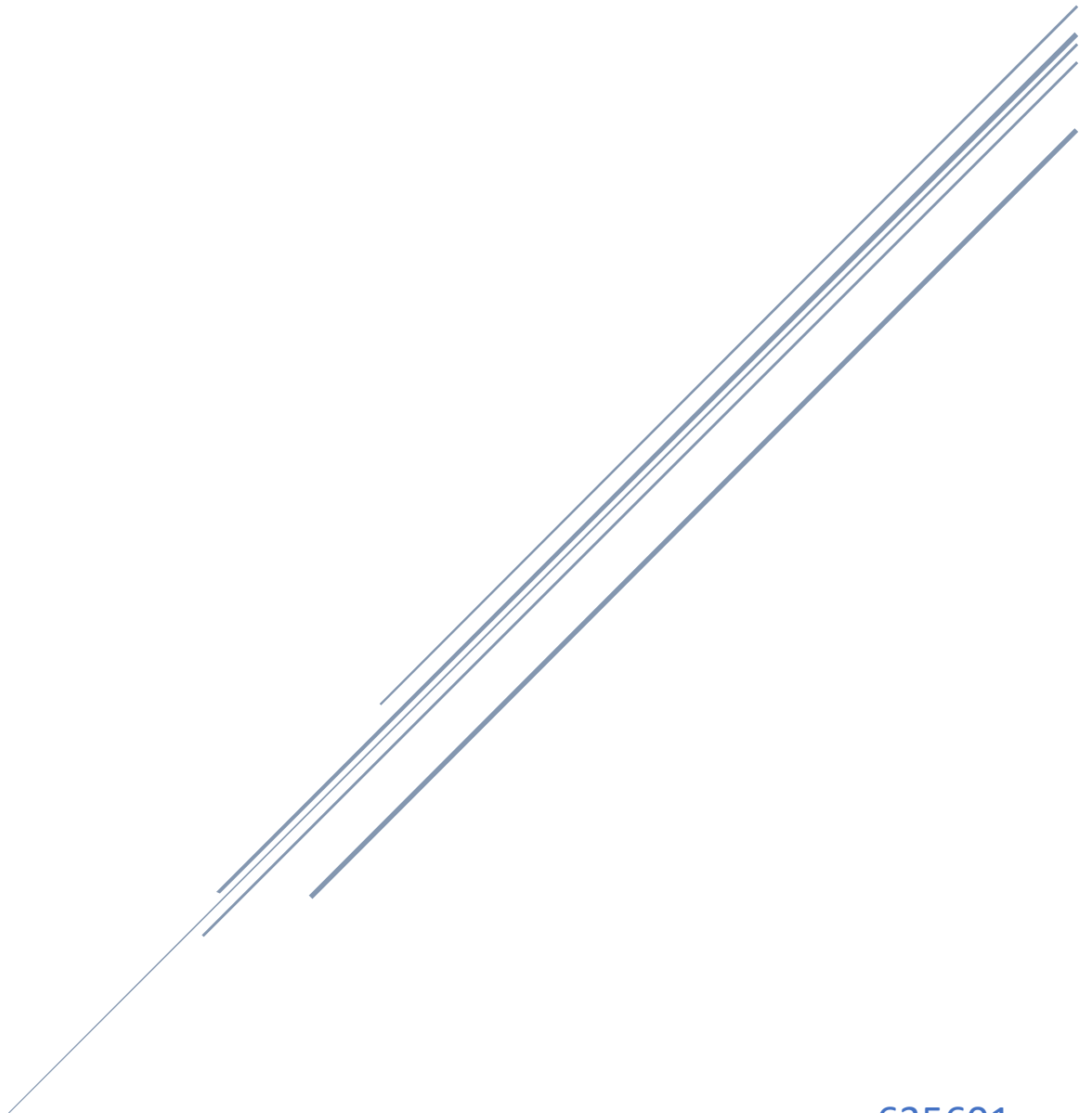


AUDITORIAS PL-SQL

Jonathan Garzon



635601
2342

Creación tablas

```
CREATE TABLE emple (  
    id NUMBER,  
    nombre VARCHAR2(50),  
    apellido VARCHAR(50)  
);  
  
select * from emple;
```

```
CREATE TABLE auditareemple (  
    id NUMBER,  
    nombre VARCHAR2 (50),  
    appellido VARCHAR2(50),  
    operacion VARCHAR2(100)  
);
```

```
SELECT * FROM auditareemple;
```

```
set SERVEROUTPUT on
```

Creación de procedimientos

```
CREATE OR REPLACE PROCEDURE p_insert_empleados  
(v_id emple.id%TYPE, v_nombre emple.nombre%TYPE, v_apellido emple.apellido%TYPE)  
AS  
BEGIN  
    SAVEPOINT i_emple;  
    INSERT INTO emple (id, nombre, apellido)  
    VALUES (v_id, v_nombre, v_apellido);  
EXCEPTION  
    when dup_val_on_index THEN
```

```
DBMS_OUTPUT.PUT_LINE('Ocurrio un error' || sqlcode);  
ROLLBACK TO SAVEPOINT i_emple;  
END p_insert_empleados;
```

```
EXECUTE p_insert_empleados(1,'Alex', 'Garzon');
```

```
CREATE OR REPLACE PROCEDURE p_delete_empleados  
(v_id NUMBER)  
AS
```

```
p_id_v NUMBER;  
BEGIN  
    SELECT id INTO p_id_v FROM emple WHERE id=v_id;  
    SAVEPOINT d;  
    IF v_id = p_id_v THEN  
        DELETE FROM emple WHERE id = v_id;  
    ELSE  
        DBMS_OUTPUT.PUT_LINE('Id no existe');  
    END IF;
```

```
EXCEPTION  
    WHEN NO_DATA_FOUND THEN  
        DBMS_OUTPUT.PUT_LINE("");  
        ROLLBACK TO SAVEPOINT D;  
END p_delete_empleados;
```

```
EXECUTE p_delete_empleados(1);
```

```
CREATE OR REPLACE PROCEDURE p_update_empleados  
(p_id NUMBER, p_name VARCHAR2, p_apellido VARCHAR2, dato NUMBER)  
IS
```

```

cont NUMBER;

c_id emple.id%TYPE;

CURSOR id_cu IS

    SELECT id FROM emple WHERE id = p_id;

BEGIN

    OPEN id_cu;

    FETCH id_cu INTO c_id;

    SELECT COUNT(*) INTO cont FROM emple;

IF cont > 0 THEN

    IF p_id = c_id THEN

        IF DATO = 1 THEN

            UPDATE emple

                SET nombre = p_name

            WHERE id = p_id;

            ELSE IF DATO = 2 THEN

                UPDATE emple

                    SET apellido = p_apellido

                WHERE id = p_id;

            ELSE IF DATO = 3 THEN

                UPDATE emple

                    SET nombre = p_name,

                    apellido = p_apellido

                WHERE id = p_id;

            END IF;

        END IF;

    END IF;

ELSE

    DBMS_OUTPUT.PUT_LINE ('ERROR, EL ID NO EXISTE');

END IF;

```

```

ELSE
    DBMS_OUTPUT.PUT_LINE ('ERROR, NO HAY DATOS');
END IF;

CLOSE id_cu;

END p_update_empleados;

```

Creacion triggers

```

CREATE OR REPLACE TRIGGER t_control_ia
    AFTER INSERT
    ON emple
    FOR EACH ROW
BEGIN
    INSERT INTO auditareemple
        (id, nombre, apellido, operacion)
        VALUES (:new.id, :new.nombre, :new.apellido, 'insert');
END t_control_ia;

CREATE OR REPLACE TRIGGER t_contro_bd
-- puede ser after o before y la operacion update, insert o delete
    BEFORE DELETE
    -- nombre de la tabla
    ON emple
    -- se puede recorrer por fila ROW o por orden STATEMENT
    FOR EACH ROW
    -- when para condicionar para ejecutar el trigger
BEGIN
    INSERT INTO auditareemple

```

```
(id, nombre, apellido, operacion)
VALUES (:old.id, :old.nombre, :old.apellido, 'Delete');
END t_contro_bd;
```

```
CREATE OR REPLACE TRIGGER t_control_bu
BEFORE UPDATE
ON emple
FOR EACH ROW
BEGIN
INSERT INTO auditaremp
(id, nombre, apellido, operacion)
VALUES (:old.id, :old.nombre, :old.apellido, 'Update');
END t_contro_bu;
```

Pruebas de ejecución.

Ejecución

```
EXECUTE p_insert_empleados(1,'Alex', 'Garzon');

CREATE OR REPLACE PROCEDURE p_delete_empleados
(v_id NUMBER)
AS
p_id_v NUMBER;
BEGIN
    SELECT id INTO p_id_v FROM emple WHERE id=v_id;
    SAVEPOINT d;
    IF v_id = p_id_v THEN
```

Salida de Script x Resultado de la Consulta x

Tarea terminada en 0,156 segundos

Trigger T_CONTROL_BU compilado

Procedimiento PL/SQL terminado correctamente.

Procedimiento PL/SQL terminado correctamente.

Evidencia

```
select * from emple;
```

```
CREATE TABLE auditareemple (
    id NUMBER,
    nombre VARCHAR2 (50),
    apellido VARCHAR2(50),
    operacion VARCHAR2(100)
);

SELECT * FROM auditareemple;

set SERVEROUTPUT on

CREATE OR REPLACE PROCEDURE p_insert,
```

Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 1 en

| ID | NOMBRE | APELLIDO |
|----|--------|----------|
| 1 | 1 Alex | Garzon |

Tabla auditoria

```
SELECT * FROM auditaremp;

set SERVEROUTPUT on

CREATE OR REPLACE PROCEDURE p_insert_
(v_id emple.id%TYPE, v_nombre emple.n
AS
BEGIN
    SAVEPOINT i_emple;
    INSERT INTO emple (id, nombre, ap
```

Salida de Script x Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 2 en 0

| ID | NOMBRE | APPELLIDO | OPERACION |
|----|--------|-----------|-----------|
| 1 | 1 Alex | Soto | Delete |
| 2 | 1 Alex | Garzon | insert |

Eliminar datos

```
EXECUTE p_delete_empleados(1);

--TRIGGER
--Auditoria

CREATE OR REPLACE TRIGGER t_control_ia
AFTER INSERT
ON emple
```

Salida de Script x Resultado de la Consulta x

Tarea terminada en 0,078 segundos

Trigger T_CONTROL_BU compilado

Procedimiento PL/SQL terminado correctamente.

Procedimiento PL/SQL terminado correctamente.

Procedimiento PL/SQL terminado correctamente.

Evidencia

```
select * from emple;
```

```
CREATE TABLE auditareemple (  
    id NUMBER,  
    nombre VARCHAR2 (50),  
    apellido VARCHAR2 (50),  
    operacion VARCHAR2 (100)  
);  
  
SELECT * FROM auditareemple;
```

Salida de Script x Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 0 en 0

| ID | NOMBRE | APELLIDO |
|----|--------|----------|
|----|--------|----------|

Tabla auditoria

```
SELECT * FROM auditareemple;
```

```
set SERVEROUTPUT on
```

Salida de Script x Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 3 en 3

| | ID | NOMBRE | APELLIDO | OPERACION |
|---|----|--------|----------|-----------|
| 1 | 1 | Alex | Soto | Delete |
| 2 | 1 | Alex | Garzon | insert |
| 3 | 1 | Alex | Garzon | Delete |

Al momento de hacer update

```
CREATE OR REPLACE TRIGGER t_control_bu
BEFORE UPDATE
  ON emple
  FOR EACH ROW
BEGIN
  INSERT INTO auditaremp
    (id, nombre, apellido, operacion)
  VALUES (:old.id, :old.nombre, :old.apellido, 'Update');
END t_control_bu;

--De sustitucion
-- SOLO APLICA A LAS VISTAS
```

Salida de Script x Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 7 en 0 segundos

| ID | NOMBRE | APPELLIDO | OPERACION |
|----|--------|-----------|-----------|
| 2 | 1 Alex | Garzon | insert |
| 3 | 1 Alex | Garzon | Delete |
| 4 | 1 Alex | Garzon | insert |
| 5 | 1 Alex | Garzon | Update |
| 6 | 1 ALEX | Garzon | Update |
| 7 | 1 ALEX | ALFONSO | Update |