

Chuleta PL/SQL - Guía Rápida

Estructura básica

```
DECLARE
    -- Declaraciones (variables, constantes, cursores...)
BEGIN
    -- Código ejecutable
EXCEPTION
    -- Manejo de errores
END;
```

Funciones

```
CREATE OR REPLACE FUNCTION suma(a NUMBER, b NUMBER) RETURN NUMBER
IS
BEGIN
    RETURN a + b;
END;
```

Procedimientos

```
CREATE OR REPLACE PROCEDURE saludar(nombre IN VARCHAR2)
IS
BEGIN
    DBMS_OUTPUT.PUT_LINE('Hola, ' || nombre);
END;
```

Cursores

```
-- Cursor explícito
DECLARE
    CURSOR c_empleados IS SELECT nombre FROM empleados;
    v_nombre empleados.nombre%TYPE;
BEGIN
    OPEN c_empleados;
    LOOP
        FETCH c_empleados INTO v_nombre;
        EXIT WHEN c_empleados%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(v_nombre);
    END LOOP;
    CLOSE c_empleados;
END;

-- Cursor FOR
BEGIN
    FOR emp IN (SELECT nombre FROM empleados) LOOP
        DBMS_OUTPUT.PUT_LINE(emp.nombre);
    END LOOP;
END;
```

Chuleta PL/SQL - Guía Rápida

Paquetes

```
-- Especificación
CREATE OR REPLACE PACKAGE nombre_paquete AS
    PROCEDURE proc_ejemplo;
    FUNCTION funcion_ejemplo RETURN NUMBER;
END nombre_paquete;

-- Cuerpo
CREATE OR REPLACE PACKAGE BODY nombre_paquete AS

    PROCEDURE proc_ejemplo IS
    BEGIN
        DBMS_OUTPUT.PUT_LINE('Procedimiento en paquete');
    END;

    FUNCTION funcion_ejemplo RETURN NUMBER IS
    BEGIN
        RETURN 42;
    END;

END nombre_paquete;
```

Manejo de errores

```
BEGIN
    -- Código
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No se encontró info');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
```

Bucles

```
-- LOOP
LOOP
    EXIT WHEN condicion;
END LOOP;

-- WHILE
WHILE condicion LOOP
    -- Código
END LOOP;

-- FOR
FOR i IN 1..10 LOOP
    DBMS_OUTPUT.PUT_LINE(i);
```

Chuleta PL/SQL - Guía Rápida

```
END LOOP;
```

Extras útiles

```
DBMS_OUTPUT.PUT_LINE('texto'); -- Mostrar mensajes
%TYPE                          -- Obtener tipo de una columna
%ROWTYPE                       -- Obtener tipo de una fila completa
RAISE_APPLICATION_ERROR(-20001, 'Mensaje'); -- Error personalizado
```

Triggers

```
-- Trigger BEFORE INSERT
CREATE OR REPLACE TRIGGER trg_before_insert
BEFORE INSERT ON empleados
FOR EACH ROW
BEGIN
    :NEW.fecha_alta := SYSDATE;
END;

-- Trigger AFTER UPDATE
CREATE OR REPLACE TRIGGER trg_after_update
AFTER UPDATE ON empleados
FOR EACH ROW
BEGIN
    DBMS_OUTPUT.PUT_LINE('Empleado actualizado: ' || :OLD.nombre || ' -> ' ||
:NEW.nombre);
END;

-- Trigger INSTEAD OF (para vistas)
CREATE OR REPLACE TRIGGER trg_instead_of_vista
INSTEAD OF INSERT ON vista_empleados
FOR EACH ROW
BEGIN
    INSERT INTO empleados(id, nombre) VALUES (:NEW.id, :NEW.nombre);
END;
```