| **OpERACIÓN** | **DESCRIPCIÓN** | **EJEMPLO** |
| --- | --- | --- |
| SELECT | Consultar datos | SELECT \* FROM USUARIO; |
| INSERT INTO | Insertar nuevos datos | INSERT INTO LIBRO VALUES ('123', 'Título', 'Autor', 'Género', SYSDATE); |
| UPDATE | Modificar datos existentes | UPDATE USUARIO SET SALDO\_MULTA = 5 WHERE ID\_USUARIO = 1; |
| DELETE | Eliminar datos de una tabla | DELETE FROM PRESTAMO WHERE ID\_PRESTAMO = 3; |
| MERGE | Insertar o actualizar según exista el dato | MERGE INTO USUARIO ... *(menos común en exámenes básicos)* |
| COMMIT | Confirmar los cambios | COMMIT; |
| ROLLBACK | Deshacer los cambios si no se ha hecho COMMIT | ROLLBACK; |

OPERACIONES DML (Data Manipulation Language)

🧱 OPERACIONES DDL (Data Definition Language)

| **OPERACIÓN** | **DESCRIPCIÓN** | **EJEMPLO** |
| --- | --- | --- |
| CREATE TABLE | Crear una nueva tabla | CREATE TABLE USUARIO (...); |
| ALTER TABLE | Modificar una tabla existente (añadir, quitar columnas, claves...) | ALTER TABLE USUARIO ADD TELEFONO VARCHAR2(15); |
| DROP TABLE | Eliminar completamente una tabla | DROP TABLE DEVOLUCION; |
| TRUNCATE | Borrar todos los registros de una tabla (sin poder recuperarlos) | TRUNCATE TABLE LIBRO; |
| CREATE VIEW | Crear una vista (consulta guardada) | CREATE VIEW V\_USUARIOS\_MULTADOS AS SELECT \* FROM USUARIO WHERE SALDO\_MULTA > 0; |
| DROP VIEW | Eliminar una vista | DROP VIEW V\_USUARIOS\_MULTADOS; |

**👥 ADMINISTRACIÓN DE USUARIOS Y PRIVILEGIOS (DDL)**

**🔐 USUARIOS**

| **OPERACIÓN** | **DESCRIPCIÓN** | **EJEMPLO** |
| --- | --- | --- |
| CREATE USER | Crear un nuevo usuario | CREATE USER bryan IDENTIFIED BY contraseña123; |
| ALTER USER | Cambiar contraseña o propiedades | ALTER USER bryan IDENTIFIED BY nueva123; |
| DROP USER | Eliminar usuario | DROP USER bryan CASCADE; |

**👑 PRIVILEGIOS DE SISTEMA**

Son permisos generales sobre la base de datos (crear tablas, usuarios, etc.)

| **OPERACIÓN** | **DESCRIPCIÓN** | **EJEMPLO** |
| --- | --- | --- |
| GRANT | Conceder privilegios | GRANT CREATE SESSION TO bryan; |
| REVOKE | Quitar privilegios | REVOKE CREATE SESSION FROM bryan; |
| GRANT ... WITH ADMIN OPTION | Permitir que el usuario dé permisos a otros | GRANT CREATE TABLE TO bryan WITH ADMIN OPTION; |

**🎯 PRIVILEGIOS DE OBJETO**

Permiten usar una tabla específica (leerla, modificarla, etc.)

| **OPERACIÓN** | **DESCRIPCIÓN** | **EJEMPLO** |
| --- | --- | --- |
| GRANT SELECT ON tabla TO usuario | Permitir consultar una tabla | GRANT SELECT ON LIBRO TO bryan; |
| GRANT INSERT, UPDATE ON tabla TO usuario | Insertar y modificar datos | GRANT INSERT, UPDATE ON USUARIO TO bryan; |
| REVOKE ALL ON tabla FROM usuario | Quitar todos los permisos | REVOKE ALL ON LIBRO FROM bryan; |

**🧾 CONSULTAR PRIVILEGIOS**

| **CONSULTA** | **RESULTADO** |
| --- | --- |
| SELECT \* FROM USER\_TAB\_PRIVS; | Privilegios sobre objetos que tienes |
| SELECT \* FROM SESSION\_PRIVS; | Privilegios de sistema de tu sesión |
| SELECT \* FROM DBA\_USERS; | Todos los usuarios del sistema (requiere permiso) |

**🧠 TARJETA DE REPASO — SINTAXIS COMPLETA DE UN TRIGGER EN ORACLE**

**📌 ESTRUCTURA BÁSICA DE UN TRIGGER**

CREATE [OR REPLACE] TRIGGER nombre\_trigger

[BEFORE | AFTER | INSTEAD OF]

[INSERT OR UPDATE OR DELETE]

ON nombre\_tabla

[FOR EACH ROW]

[WHEN (condición)] -- ⚠️ NO USAR :NEW o :OLD aquí directamente

DECLARE

-- Variables locales

BEGIN

-- Lógica del trigger

EXCEPTION

-- Manejadores de errores

END;

/

**🧱 OPCIONES EXPLICADAS**

| **OPCIÓN** | **DESCRIPCIÓN** |
| --- | --- |
| BEFORE | El trigger se ejecuta **antes** de la acción DML. Útil para validaciones. |
| AFTER | Se ejecuta **después** del cambio. Ideal para actualizar otras tablas. |
| INSTEAD OF | Solo para vistas. Sustituye la acción sobre una vista. |
| INSERT, UPDATE, DELETE | Define qué tipo de evento activa el trigger. |
| FOR EACH ROW | El trigger se ejecuta **una vez por cada fila afectada**. |
| WHEN (condición) | Condición **simple** para activar el trigger. **No se puede usar :NEW, :OLD con operadores lógicos o aritméticos aquí.** |

**🧠 TARJETA DE REPASO — FUNCIONES EN PL/SQL**

**📌 ¿Qué es una función?**

Una **función** es un bloque de código que **devuelve un valor**. Se puede usar en SELECT, dentro de procedimientos, vistas, etc.

**🧱 SINTAXIS COMPLETA**

CREATE [OR REPLACE] FUNCTION nombre\_funcion (

parametro1 [IN | OUT | IN OUT] tipo,

...

)

RETURN tipo

IS

-- Declaración de variables

BEGIN

-- Lógica de la función

RETURN valor;

EXCEPTION

WHEN others THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Mensaje de error');

END;

/

**✅ OPCIONES EXPLICADAS**

| **OPCIÓN** | **SIGNIFICADO** |
| --- | --- |
| IN | El parámetro **entra** a la función (por defecto). |
| OUT | La función **devuelve** algo por ese parámetro (no se usa mucho en funciones). |
| IN OUT | Entra y sale. (Muy raro en funciones, más usado en procedimientos.) |
| RETURN | Tipo del valor final que devuelve la función. |

**🔁 ¿Dónde usar funciones?**

| **CONTEXTO** | **EJEMPLO** |
| --- | --- |
| En un SELECT | SELECT nombre\_funcion(5) FROM DUAL; |
| En PL/SQL | v\_resultado := nombre\_funcion(param); |
| En un paquete | FUNCTION dentro del PACKAGE y PACKAGE BODY |

**⚠️ ERRORES COMUNES**

| **ERROR** | **SOLUCIÓN** |
| --- | --- |
| No usar RETURN | Toda función necesita devolver algo con RETURN valor; |
| SELECT dentro sin INTO | Si haces un SELECT, debe ser SELECT ... INTO variable |
| Usar DBMS\_OUTPUT sin activar | Necesitas SET SERVEROUTPUT ON para ver resultados en consola |

**✅ EJEMPLO REALISTA**

CREATE OR REPLACE FUNCTION contar\_libros\_genero(p\_genero IN VARCHAR2)

RETURN NUMBER

IS

v\_total NUMBER := 0;

BEGIN

SELECT COUNT(\*) INTO v\_total

FROM LIBRO

WHERE GENERO = p\_genero;

RETURN v\_total;

END;

/

**🧠 TARJETA DE REPASO — PROCEDIMIENTOS EN PL/SQL**

**📌 ¿Qué es un procedimiento?**

Un **procedimiento** es un bloque de código que **ejecuta acciones**, puede **recibir datos** y **devolver resultados por parámetros**, pero **NO devuelve valor con RETURN** como una función.

**🧱 SINTAXIS COMPLETA**

CREATE [OR REPLACE] PROCEDURE nombre\_procedimiento (

parametro1 [IN | OUT | IN OUT] tipo,

...

)

IS

-- Declaración de variables

BEGIN

-- Lógica del procedimiento

EXCEPTION

WHEN OTHERS THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Mensaje de error');

END;

/

**✅ TIPOS DE PARÁMETROS**

| **TIPO** | **SIGNIFICADO** | **EJEMPLO** |
| --- | --- | --- |
| IN | El valor entra, pero no puede cambiarse | IN p\_nombre VARCHAR2 |
| OUT | El valor sale, el procedimiento lo rellena | OUT p\_resultado NUMBER |
| IN OUT | El valor entra y puede salir modificado | IN OUT p\_valor NUMBER |

**🧩 ¿CUÁNDO USAR UN PROCEDIMIENTO?**

“Si me piden ejecutar acciones (mostrar, actualizar, insertar...) y devolver resultados por variables, uso un procedimiento.”

**🧾 LLAMAR A UN PROCEDIMIENTO DESDE PL/SQL**

DECLARE

v\_total NUMBER;

BEGIN

nombre\_procedimiento('valor', v\_total);

DBMS\_OUTPUT.PUT\_LINE('Total: ' || v\_total);

END;

**⚠️ ERRORES COMUNES**

| **ERROR** | **SOLUCIÓN** |
| --- | --- |
| Usar RETURN en un procedimiento | ❌ Solo se usa en funciones. Usa OUT para devolver valores. |
| No declarar los tipos correctamente | Asegúrate de usar %TYPE si es necesario. |
| No usar BEGIN / END | Todo el cuerpo del procedimiento debe estar dentro de BEGIN ... END; |

**✅ EJEMPLO REALISTA**

CREATE OR REPLACE PROCEDURE obtener\_datos\_libro (

p\_isbn IN LIBRO.ISBN%TYPE,

p\_titulo OUT LIBRO.TITULO%TYPE,

p\_autor OUT LIBRO.AUTOR%TYPE

)

IS

BEGIN

SELECT TITULO, AUTOR INTO p\_titulo, p\_autor

FROM LIBRO

WHERE ISBN = p\_isbn;

END;

/

**🧠 TARJETA DE REPASO — PAQUETES EN PL/SQL (PACKAGE)**

**📌 ¿Qué es un paquete?**

Un **PACKAGE** es una **estructura que agrupa** procedimientos, funciones, cursores y tipos en un solo módulo.

🔁 Es como una “carpeta” con herramientas que puedes reutilizar.

**🧱 ESTRUCTURA COMPLETA**

**1. Especificación (lo que se ve desde fuera)**

CREATE OR REPLACE PACKAGE nombre\_paquete IS

PROCEDURE nombre\_procedimiento(...);

FUNCTION nombre\_funcion(...) RETURN tipo;

-- Tipos, cursores, variables públicas opcionales

END;

/

**2. Cuerpo (la lógica real)**

CREATE OR REPLACE PACKAGE BODY nombre\_paquete IS

PROCEDURE nombre\_procedimiento(...) IS

BEGIN

-- lógica

END;

FUNCTION nombre\_funcion(...) RETURN tipo IS

BEGIN

-- lógica

RETURN valor;

END;

END;

/

**📦 ¿QUÉ PUEDE INCLUIR UN PAQUETE?**

| **ELEMENTO** | **USO PRINCIPAL** |
| --- | --- |
| PROCEDURES | Ejecutar acciones complejas |
| FUNCTIONS | Devolver valores reutilizables |
| CURSORES | Recorrer datos y devolver resultados línea a línea |
| TIPOS | Declarar registros o colecciones |
| VARIABLES | Compartir datos dentro del paquete |

**⚠️ ERRORES COMUNES**

| **ERROR** | **SOLUCIÓN** |
| --- | --- |
| Declarar una función en el body pero no en el spec | ¡Debe estar en ambas partes! |
| No cerrar END; o el paquete con / | Usa siempre / al final del package y del body. |
| Llamar un objeto no público | Solo se puede usar lo declarado en el PACKAGE (spec) |

**📌 EJEMPLO DE LLAMADA**

BEGIN

nombre\_paquete.nombre\_procedimiento(...);

v := nombre\_paquete.nombre\_funcion(...);

END;