1. Crear un procedure
2. Tipos de datos
3. Operadores
4. Estructuras secuenciales (IF, FOR LOOP INVERTIDO, WHILE, DO-WHILE)
5. Definir variables
6. Estructura de bloques
   1. Ver procedimientos y funciones
   2. Ver como se crearon
   3. Llamar a procedimiento
   4. Llamar funcion
   5. Borrar
7. TRIGGERS
8. Crear procedimiento

PL = Procedural Language (Oracle 90) CASE-SENSITIVE(no distingue de mayusculas y minusculas)

Transact-sql (Microsoft)

CREATE OR REPLACE **PROCEDURE**

BEGIN

…

END

CREATE OR REPLACE FUNCTION

BEGIN

…

END

**Tipos de datos**

**ORACLE MySQL**

NUMBER INTEGER

CHAR

VARCHAR2 VARCHAR

BOOLEAN BOOLEAN

**Operadores**

**ORACLE MySQL (Lo mismo, menos..)**

:=

+,-,\*,/,\*\* ^(Exponente)

=,<>,>,<,>=,<= !=, <>

AND, OR, NOT AND, $$;NOT,!OR, ||;;XOR

CONCATENACIÓN ES DOBLE TUBERÍA ||

**Condicional IF**

IF (expresión) THEN

– instrucción

ELSE IF(expresiones) THEN

– Instrucciones

ELSE

–Instrucciones

END IF;

**Control de Flujo FOR**

**FOR** contador **IN**

inicio.. fin **LOOP**

**END LOOP;**

–instrucciones

**Control de Flujo FOR INVERTIDO**

**FOR** contador **IN**

**[REVERSE] i**nicio.. fin **LOOP**

–instrucciones

**END LOOP;**

**Control de flujo while**

**WHILE** (EXPRESIÓN) **LOOP**

– Instrucciones

**END LOOP;**

**– Control de flujo Loop (Equivalente a do - while)**

**LOOP**

– Instrucciones

**IF**(expresión) **THEN**

– Instrucciones

**EXIT;**

**END IF;**

**END LOOP;**

**DEFINICIÓN DE VARIABLES**

nombreVariable [CONSTANT] < tipo\_dato> [NOT NULL] O [:= valor inicial];

nombre varchar2 := ‘Juan’;

fecha\_nacimiento DATE;

**ESTRUCTURA DE LOS BLOQUES (oracle)**

[declare |is | as] (opciones)

/\*Parte declarativa \*/

BEGIN

/\* Parte de ejecución \*/

[exceptio]

/\* Parte excepciones \*/

END;

**Ver procedimientos y funciones:**

show [PROCEDURE | function]

status WHERE dB= ‘SAKILA’;

show PROCEDURE status where db = 'sakila';

show FUNCTION status where db = 'sakila';

**Ver como se crearon:**

SHOW CREATE create <nombre;

SHOW CREATE function get\_customer\_balance;

\g → al final de cada sentencia para que aparezca ordenado

**Llamar a un PROCEDIMIENTO:**

CALL n.basededatos].nombre\_procedimiento[parámetros]

SET @resultado = null;

CALL sakila.film\_in\_stock(19,1,@resultado);

SHOW PROCEDURE STATUS WHERE Name = 'procedimiento03';

**1.- Variable vacía**

SET @resultado = null;

CALL sakila.film\_in\_stock(19,1,@resultado);

**– Llamar a una función:** (los parámetros de entrada son los in y de salida los out)

SHOW CREATE function inventory\_in\_stock;

LAS FUNCIONES EN UN SELECT

SELECT inventory\_in\_stock(9); – 9 es el valor que le damos

**BORRAR:**

DROP PROCEDURE nombre;

DROP FUNCTION nombre;

**TRIGGERS:**

Son procedimientos automatizados, se ejecutan cuando ocurren eventos

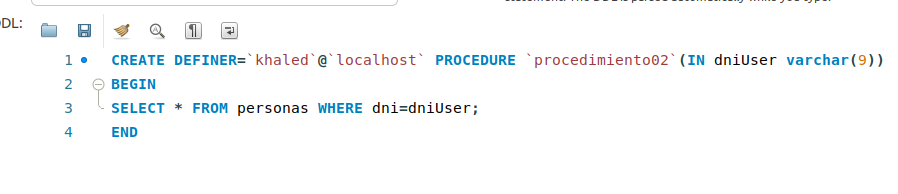
**Características de como crear procedimientos**

delimiter//

CREATE PROCEDURE procedimiento01 (IN dniUser varchar(9)) BEGIN SELECT \* FROM Persona WHERE dni=dniUser; END//

delimiter ; (espacio antes de ;)

CALL procedimiento01('00000000A');



CREATE PROCEDURE procedimiento01 (IN dniUser varchar(9))

BEGIN

SELECT \* FROM Persona WHERE dni=dniUser; END

**COMMENT:** para comentarios

**Meter en variable:** select COUNT(\*) FROM Persons **INTO numero**

*ERROR 1318 (42000): Incorrect number of arguments for PROCEDURE Ciudadano.procedimiento03; expected 1, got 0*

* Crear variables

**ACTIVIDADES**

1.- Lo use el usuario Juan y en caso de que no tenga teléfono que aparezca el mensaje

2.- Procedimiento para contar número de personas que hay en la tabla persona y al fecha de la creación con esas dos variables

3-. Procedimiento para alistar los libros de un autor, pasando parámetro nombre autor, mostrar título editorial y precio

4.- crear un procedimiento para buscar los teléfonos de una persona por su nombre y apellidos y el resultado deberá mostrar nombres apellidos y teléfonos

SELECT routine\_name, routine\_type

FROM information\_schema.routines

WHERE routine\_type = 'PROCEDURE';

5. CONTAR LOS TELEFONOS DE UNA PERSONA PASANDO EL DNI

**Estructuras alternaitvas**