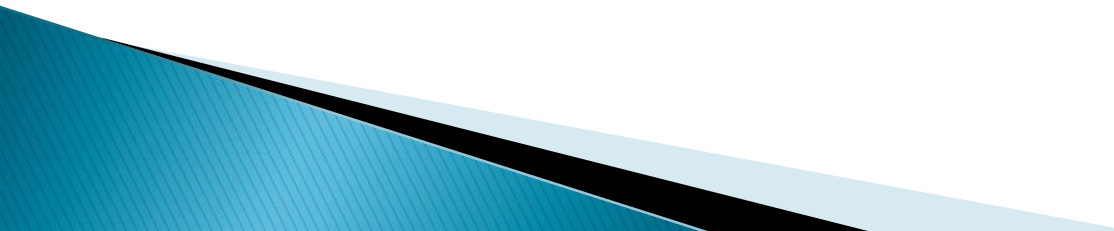


UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE COMPUTACIÓN

# **Tema V. Programación con SQL**

## **Funciones**

Ing. Lucila Patricia  
Arellano Mendoza  
2018



# Funciones

Una función es similar a un procedimiento. La llamada a una función se realiza como parte de una expresión.

```
CREATE [OR REPLACE] FUNCTION <nombreFuncion>(  
[argumento [{IN\OUT\INOUT}] tipo, ....  
argumento [{IN\OUT\INOUT}] tipo)]  
RETURN tipo_retorno  
IS  
BEGIN  
    Declaraciones ...  
END <nombreFuncion>;  
/
```

Para ver la descripción de una función se hace con la siguiente sintaxis:

DESCRIBE <nombreFunción>

Para borrar una función

DROP FUNCTION <nombreFunción>;

Habr  un archivo spool y realice lo siguiente:

**Ejercicio 13.** La siguiente funci  devuelve el saldo seg n el n mero de cuenta solicitado de la tabla ctaAhorro.

```
CREATE OR REPLACE FUNCTION ftObtieneSaldo(  
cuenta IN CHAR  
)  
RETURN NUMBER  
IS  
    vSaldo NUMBER(11,2);  
BEGIN  
    SELECT saldo INTO vsaldo  
    FROM ctaAhorro  
    WHERE claveCta =cuenta;  
    RETURN(vSaldo);  
END ftObtieneSaldo;
```

## Para ejecutarlo

```
DECLARE
```

```
    vSaldo NUMBER (11,2);
```

```
BEGIN
```

```
    vSaldo:=ftObtieneSaldo('Ah4');  --Llamada a la función
```

```
    DBMS_OUTPUT.PUT_LINE('El saldo es : '|| vSaldo);
```

```
END;
```

```
/
```

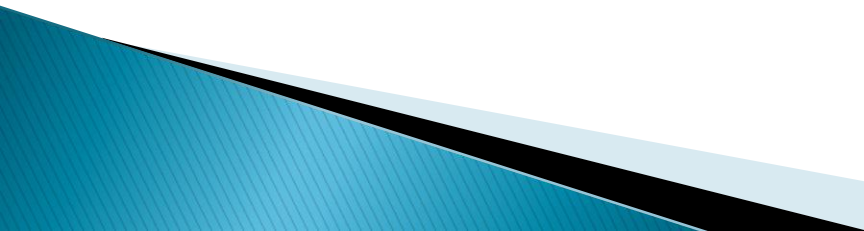
## Ejemplo:

Bonificar en \$1000 el monto de las cuentas de ahorro de los empleados, cuya cuenta tenga una antigüedad de 5 años. Almacene el número y nombre del empleado, el número de cuenta, nuevo saldo, la fecha en la que obtuvo la cuenta y la fecha en la que obtuvo la bonificación, en una tabla nueva llamada PREMIADOS

1. Crear la tabla PREMIADOS
2. Crear la función que calcule la antigüedad
3. Crear el procedimiento

## Tabla PREMIADOS

```
CREATE TABLE premiados(  
claveEmp CHAR(4) NOT NULL,  
nombre VARCHAR2(20) NOT NULL,  
cta CHAR(4) NOT NULL,  
saldo NUMBER(11,2) NOT NULL,  
fechaCuenta DATE NOT NULL,  
fechaBon DATE NOT NULL  
);
```



```
CREATE OR REPLACE FUNCTION ftAntiguedad(  
    reg IN CHAR)  
RETURN NUMBER  
IS  
    vFechaCta ctaAhorro.fechaCta%TYPE;  
    vAntiguedad NUMBER(2);  
BEGIN  
    SELECT fechaCta  
    INTO vFechaCta  
    FROM ctaAhorro  
    WHERE claveCta=reg;  
    vAntiguedad:=TRUNC(MONTHS_BETWEEN(SYSDATE,  
    vFechaCta)/12);  
    RETURN(vAntiguedad);  
END ftAntiguedad;
```



```
CREATE OR REPLACE PROCEDURE spBonificaCta  
AS
```

```
    vEmp empleado.numEmp%TYPE;  
    vNombre empleado.nombreEmp%TYPE;  
    vCta ctaAhorro.claveCta%TYPE;  
    vSaldo ctaAhorro.saldo%TYPE;  
    vFecha ctaAhorro.fechaCta%TYPE;  
    vSaldoNvo ctaAhorro.saldo%TYPE;  
    vAntiguedad NUMBER(2);
```

```
CURSOR curBonifica IS
```

```
    SELECT numEmp, nombreEmp, claveCta, saldo, fechaCta  
    FROM ctaAhorro  
    NATURAL JOIN empleado;
```

BEGIN

OPEN curBonifica ;

LOOP

FETCH curBonifica

INTO vEmp, vNombre, vCta, vSaldo, vFecha;

vAntiguedad:=ftAntiguedad(vCta);      --Llamada a función

IF vAntiguedad >=5 THEN

    vSaldoNvo:=vSaldo+ 1000;

    UPDATE ctaAhorro SET saldo=vSaldoNvo

    WHERE numEmp=vEmp;

    INSERT INTO premiados(claveEmp, nombre, cta, saldo,  
        fechaCuenta, fechaBon)

```
VALUES(vEmp, vNombre, vCta, vSaldoNvo, vFecha, SYSDATE);  
COMMIT;  
END IF;  
EXIT WHEN curBonifica %NOTFOUND;  
END LOOP;  
CLOSE curBonifica ;  
END spBonificaCta;  
/
```

```
SQL> EXEC spBonificaCta;
```

```
SQL> SELECT * FROM premiados;
```

Cierre su archivo spool

