Facilitador(a): Ing. Carmen Ortega Asignatura: Base de Datos II

Fecha: 13/09/2020 Grupo: \_\_\_1IL131

Estudiantes: Jacky He, 3-740-114

Yui Lo, 8-929-854

Maria José Martínez PE-15-1146

Dhiraj Lakhwani 3-743-1798

Richie Ieong 8-970-668

1. **TÍTULO** **DE LA EXPERIENCIA**: Crear procedimientos y funciones invocados MySQL
2. **TEMAS:**

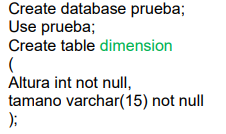
En este ejercicio se utilizarán todas las instrucciones para crear procedimientos y funciones invocados por SQL

1. **OBJETIVO(S):**

Aplicar conceptos para crear procedimientos y funciones invocados por SQL y aplicarlo en un ejemplo práctico.

La actividad le permite al estudiante enriquecer su experiencia en el desarrollo y construcción de conocimientos del curso de Base de Datos II.

1. **METODOLOGÍA:**
2. Divida el salón en grupos.
3. Distribuya el taller a cada grupo.
4. Cada equipo contará con un período de laboratorio (2 períodos de 45 min) para dar respuesta a las preguntas.
5. Primero realice el taller de forma individual y una vez desarrollado haga una reunión video llamada en equipo para que colaborativamente den respuesta en equipo las preguntas del ejercicio.
6. Se deben recibir en sección tareas de la plataforma de clases virtuales la respectiva asignación resuelta o realizar el test online del taller.
7. **PROCEDIMIENTO O ENUNCIADO DE LA EXPERIENCIA:**
8. Crear una función invocada desde SQL que muestre como obtenemos la factorial de un número “x” ingresado como parámetro.
9. Crear una función que retorne en el nombre completo de la prioridad de un cliente, introduciendo como parámetro el campo prioridad.
   1. Para ello la función se llamará EXT\_PRIORIDAD, la cual tiene como parámetro de variable cliente\_prioridad con tipo de dato VARCHAR (5).
   2. El tipo de dato que retornará a la función será VARCHAR(20).
   3. Usar la estructura CASE para determinar en la variable cliente\_prioridad la clase de cliente de la siguiente forma: Si es A es ALTO, si es M es MEDIO, si es B es Bajo de lo contrario es NN
   4. Usar la función creada y construir como podremos consultar registros de los clientes de la base de datos. Estructura de la tabla CLIENTE: NOMBRE, APELLIDO, PRIORIDAD
10. Cuál sería la instrucción, Si quisiéramos añadir una descripción a una función que calcula el promedio de huéspedes diario con respecto a una fecha llamada promedio\_huespedes.
11. Cuál sería la instrucción para borrar una función que retorna en el ingreso neto con respecto a todos los tiquetes aéreos comprados en una sucursal de una aerolínea, llamada ingreso\_neto\_sucursal
12. Escribir un procedimiento que reciba dos números y visualice su suma.
13. Codificar un procedimiento que reciba una cadena y la visualice al revés. Use FOR LOOP.
14. Escribir una función que reciba una fecha y devuelva el año, en número, correspondiente a esa fecha.
15. Codificar un procedimiento que reciba una lista de hasta 5 números y visualice su suma.
16. Implementar un procedimiento que reciba un importe y visualice el desglose del cambio en unidades monetarias de 1, 5, 10, 25, 50, 100, 200, 500, 1000, 2000, 5000 Ptas. en orden inverso al que aparecen aquí enumeradas. Use While, Loop
17. Escribir un procedimiento que modifique la localidad de un departamento. El procedimiento recibirá como parámetros el número del departamento y la localidad nueva.
18. Desarrollar una función que devuelva el número de años completos que hay entre dos fechas que se pasan como argumentos
19. Crea la siguiente base de datos y construir procedimiento Sp\_DecidirDimensión cuyo parámetro de entrada es Altura tipo de dato Integer , la variable1 del procedimiento es tipo CHAR (15), si variable1 es > 10 es Grande, si variable1> 5 es Mediano y de lo contrario es pequeño. Al final INSERT INTO dimension VALUES (altura, variable1);



1. Se necesita calcular el valor de venta de un producto sabiendo que tiene un costo y porcentaje de ganancia, ejemplo si el costo es 2.000 y el porcentaje de ganancia es 15, el valor venta seria 2.300 Primera pregunta antes de crear una función que parámetros necesita y que retorna, en este caso se necesita dos valores uno flotante (el costo y otro entero ( el porcentaje de ganancia) y retornara un valor flotante. CREATE FUNCTION calcularvalorventa
2. Cree una tabla llamada productos (código, nombre, costo, porganancia, cantidad)

Create table productos

(

Codigo int not null primary key,

Nombre varchar(30) not null,

Costo float not null,

Porganancia int not null,

Cantidad float not null

);

* **Inserte varios datos**.

Insert into productos values(1,’arroz libra’,850,15,10);

Insert into productos values(2,’gasesosa 350’,800,20,30);

Insert into productos values(3,’avena alpinavaso’,1300,10,45);

Insert into productos values(4,’menta helada’,25,100,150);

Insert into productos values(5,’salsa tomate fruco1000gr’,5250,30,8);

select codigo, nombre, costo, calcularvalorventa(costo, Porganancia)

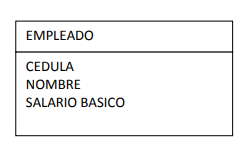
as valorventa from productos;

* Nota: observe que emplea la función calcularvalorventa del ejercicio anterior.
* Crear otra función que calcule la ganancia**. CREATE FUNCTION calcularganancia**
* Y lo podría utilizar así:

select codigo, nombre, costo, calcularvalorventa(costo, Porganancia)

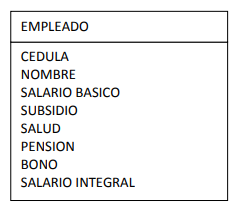
as valorventa, calcularganancia(costo, Porganancia) as ganancia from productos;

1. Dada la siguiente tabla



* 1. **Cree una función para cada punto teniendo en cuenta que:**
* El subsidio de transporte equivale al 7% al salario básico
* La salud que corresponde al 4% al salario básico
* La pensión que corresponde al 4% al salario básico
* Un bono que corresponde al 8% al salario básico
* El salario integral es la suma del salario básico -salud - pension +bono + sub de
* transporte.
  1. **Cree un procedimiento que Inserte información a la tabla y pruebe las funciones.**

1. Dada la siguiente tabla



* 1. **Cree una función para cada punto teniendo en cuenta que:**
* El subsidio de transporte equivale al 7% al salario básico
* La salud que corresponde al 4% al salario básico
* La pensión que corresponde al 4% al salario básico
* Un bono que corresponde al 8% al salario básico
* El salario integral es la suma del salario básico -salud - pension +bono + sub de
* transporte.
  1. **Cree un procedimiento que Inserte información a la tabla y pruebe las funciones.**

**NOTA: DESARROLLE SUS RESPUESTAS EN LA SECCION G. RESULTADOS DE ESTA PLANTILLA.**

1. **RECURSOS:**

Laptop o PC con los recursos, Internet, material de clases preparado por el docente.

1. **RESULTADOS:**
2. **(Yui Lo)**

delimiter //

CREATE FUNCTION F\_Factorial(n INT)

RETURNS INT(11)

Deterministic

BEGIN

DECLARE factorial INT;

SET factorial = n ;

IF n <= 0 THEN

RETURN 1;

END IF;

bucle: LOOP

SET n = n - 1 ;

IF n<1 THEN

LEAVE bucle;

END IF;

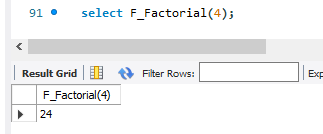
SET factorial = factorial \* n ;

END LOOP bucle;

RETURN factorial;

END//

delimiter ;



1. **(Yui Lo)**

DELIMITER //

CREATE FUNCTION EXT\_PRIORIDAD (cliente\_prioridad VARCHAR(5))

RETURNS VARCHAR(20)

deterministic

BEGIN

CASE cliente\_prioridad

WHEN 'A' THEN RETURN 'ALTO';

WHEN 'M' THEN RETURN 'MEDIO';

WHEN 'B' THEN RETURN 'BAJO';

ELSE RETURN 'NN';

END CASE;

END//

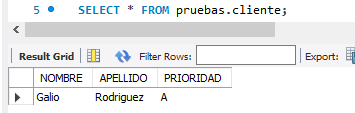
DELIMITER ;

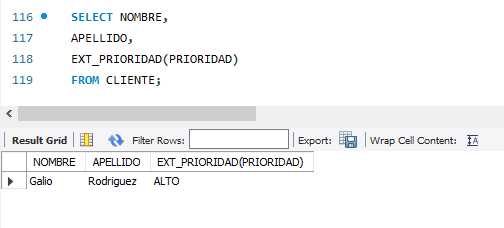
SELECT NOMBRE,

APELLIDO,

EXT\_PRIORIDAD(PRIORIDAD)

FROM CLIENTE;





1. **(Yui Lo)**

ALTER FUNCTION promedio\_huespedes

COMMENT ‘Promedio de huéspedes diario con respecto a la fecha ‘;

1. **Jacky He**

DROP FUNCTION ingreso\_neto\_sucursal

1. **Jacky He**

DELIMITER //

CREATE PROCEDURE SUMAR(IN n1 INT, IN n2 INT)

BEGIN

DECLARE suma, a, b INT;

SET a = n1, b = n2;

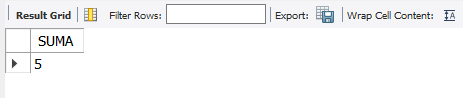
SET suma = a + b;

SELECT suma AS 'SUMA';

END; //

DELIMITER ;

CALL SUMAR(1, 2);



1. **Richie Ieong**

DELIMITER //

CREATE PROCEDURE reves(in cadena varchar(60))

BEGIN

DECLARE cadena\_reves varchar(60);

DECLARE x INT;

SET x=length(cadena);

SET cadena\_reves= '';

reves\_1: LOOP

SET cadena\_reves=CONCAT(cadena\_reves, MID(cadena, x, 1));

SET x=x-1;

IF x=0 THEN

leave reves\_1;

END IF;

END LOOP;

SELECT cadena\_reves;

END; //

DELIMITER //

CALL reves("Estoy usando MySQL");



1. **Jacky He**

DELIMITER //

CREATE FUNCTION AÑO(fecha DATE)

RETURNS INT

DETERMINISTIC BEGIN

DECLARE num INT;

DECLARE f DATE;

SET f = fecha;

SET num = YEAR(f);

RETURN num;

END; //

DELIMITER ;

select año('2020-10-10') AS 'AÑO';



1. **Richie Ieong**

DELIMITER //

CREATE PROCEDURE SUMA(IN n1 INT, IN n2 INT,IN n3 INT,IN n4 INT,IN n5 INT)

BEGIN

DECLARE suma INT;

SET suma = n1+n2+n3+n4+n5;

SELECT suma AS 'SUMA';

END; //

DELIMITER ;

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. **María José Martínez**

DELIMITER //

CREATE PROCEDURE desg\_cambio(IN pago INT)

BEGIN

DECLARE cambio, moneda, dolar1, dolar5, unidades,

dolar10, dolar25, dolar50, dolar100, dolar200, dolar500, dolar1000,

dolar2000, dolar5000 int;

SET moneda=0, dolar1=0, dolar5=0, unidades=0,

dolar10=0, dolar25=0, dolar50=0, dolar100=0, dolar200=0, dolar500=0, dolar1000=0,

dolar2000=0, dolar5000=0;

SET cambio = pago;

while cambio>0 do

if cambio>5000 then

set moneda =5000;

set dolar5000=truncate(cambio/moneda, 0);

elseif cambio>=2000 then

set moneda =2000;

set dolar2000= truncate(cambio/moneda, 0);

elseif cambio>=1000 then

set moneda =1000;

set dolar1000= truncate(cambio/moneda, 0);

elseif cambio>=500 then

set moneda =500;

set dolar500= truncate(cambio/moneda, 0);

elseif cambio>=200 then

set moneda =200;

set dolar200= truncate(cambio/moneda, 0);

elseif cambio>=100 then

set moneda =100;

set dolar100= truncate(cambio/moneda, 0);

elseif cambio>=50 then

set moneda =50;

set dolar50=truncate(cambio/moneda, 0);

elseif cambio>=25 then

set moneda =25;

set dolar25= truncate(cambio/moneda, 0);

elseif cambio>=10 then

set moneda =10;

set dolar10= truncate(cambio/moneda, 0);

elseif cambio>=5 then

set moneda =5;

set dolar5= truncate(cambio/moneda, 0);

else

set moneda= 1;

set dolar1= truncate(cambio/moneda, 0);

end if;

set cambio= MOD(cambio, moneda);

End while;

select concat(" se ingresó: ", pago) as "Resumen de transacción: ",

dolar5000 as "dolares de 5000", dolar2000 as "dolares de 2000",

dolar1000 as "dolares de 1000", dolar500 as "dolares de 500",

dolar200 as "dolares de 200", dolar100 as "dolares de 100",

dolar50 as "dolares de 50", dolar25 as "dolares de 25",

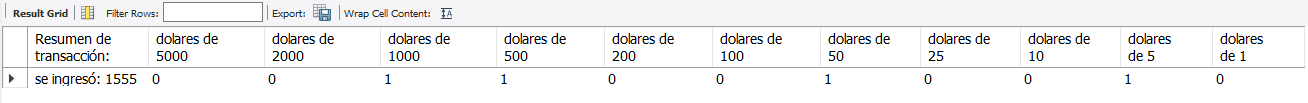
dolar10 as "dolares de 10", dolar5 as "dolares de 5",

dolar1 as "dolares de 1";

END

// DELIMITER ;

CALL desg\_cambio(1555);



1. **Richie Ieong**

DELIMITER //

CREATE PROCEDURE act\_localidad(in n INT,in localidad\_1 varchar(60) )

BEGIN

update departamentos

SET localidad=localidad\_1

where num\_depto=n;

END; //

DELIMITER //

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. **Dhiraj Lakhwani**

DROP procedure IF EXISTS `year`;

DELIMITER $$

CREATE PROCEDURE `year`(IN fechainic DATE, IN fechafin DATE)

BEGIN

DECLARE fecha1, fecha21 DATE;

DECLARE anios INT;

SET fecha1=fechainic, fecha2=fechafin;

SET anios= timestampdiff(YEAR, fechainic, fechafin);

SELECT abs(anios) as 'Diferencia en años';

END$$

DELIMITER ;

A screenshot of a cell phone

Description automatically generated

1. **Dhiraj Lakhwani**

CREATE DATABASE prueba;

USE prueba;

CREATE TABLE dimension (altura int not null,

tamano VARCHAR(10));

DELIMITER $$

CREATE PROCEDURE Sp\_DecidirDimension(Altura int)

Begin

DECLARE variable1 CHAR(15);

IF altura>10 THEN

SET variable1='GRANDE';

ELSE IF altura>5 THEN

SET variable1='MEDIANO';

ELSE

SET variable1='PEQUEÑO';

END IF;

INSERT into dimension VALUES (altura, variable1);

END$$

DELIMITER ;

CALL Sp\_DecidirDimension(15);

CALL Sp\_DecidirDimension(6);

CALL Sp\_DecidirDimension(4);

Graphical user interface, application

Description automatically generated

1. **Dhiraj Lakhwani**  
   DELIMITER //

CREATE FUNCTION calcularvalorventa (costo float, porcen\_gan int)

returns float deterministic

BEGIN

declare respuesta float ;

SET respuesta= costo + (costo \* (porcen\_gan/100));

RETURN respuesta;

END//

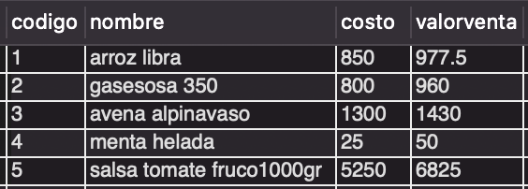
DELIMITER ;

Graphical user interface, application

Description automatically generated

**María José Martínez**

1. **Antes:**



**Creación de la función:**

DELIMITER //

CREATE FUNCTION calcularganancia (costo float, porcen\_gan int)

returns float deterministic

BEGIN

declare ganancia, valorventa float ;

SET valorventa= costo + (costo \* (porcen\_gan/100));

set ganancia= valorventa- costo;

RETURN ganancia;

END//

DELIMITER ;

select codigo, nombre, costo, calcularvalorventa(costo, Porganancia)

as 'valorventa', calcularganancia(costo, Porganancia) as 'ganancia'

from productos;

**Después:**



**María José Martínez**

**Creación de la tabla empleados**

create table empleado(

cedula varchar(12) not null primary key,

nombre varchar(80) not null,

salario\_basico float not null

);

**Creación de una función para cada punto teniendo en cuenta que:**

* El subsidio de transporte equivale al 7% al salario básico

DELIMITER //

CREATE FUNCTION calcsubsidio (salariob float)

returns float deterministic

BEGIN

declare sub float ;

SET sub= salariob\*0.07;

RETURN sub;

END//

DELIMITER ;



* La salud que corresponde al 4% al salario básico

DELIMITER //

CREATE FUNCTION calcsalud (salariob float)

returns float deterministic

BEGIN

declare salud float ;

SET salud= salariob\*0.04;

RETURN salud;

END//

DELIMITER ;



* La pensión que corresponde al 4% al salario básico

DELIMITER //

CREATE FUNCTION calcpension (salariob float)

returns float deterministic

BEGIN

declare pension float ;

SET pension= salariob\*0.04;

RETURN pension;

END//

DELIMITER ;



* Un bono que corresponde al 8% al salario básico

DELIMITER //

CREATE FUNCTION calcbono (salariob float)

returns float deterministic

BEGIN

declare bono float ;

SET bono= salariob\*0.08;

RETURN bono;

END//

DELIMITER ;



* El salario integral es la suma del salario básico -salud - pension +bono + sub de

transporte.

DELIMITER //

CREATE FUNCTION calcsal\_integral(salariob float)

returns float deterministic

BEGIN

declare sal\_int, calcsubsidio, calcsalud, calcpension, calcbono float ;

SET sal\_int= salariob + calcsubsidio (salariob )+ calcsalud (salariob )+

calcpension (salariob )+ calcbono (salariob );

RETURN sal\_int;

END//

DELIMITER ;

**Cree un procedimiento que Inserte información a la tabla y pruebe las funciones.**

**Procedimiento:**

DELIMITER //

create procedure insertarvalores(

in cedula varchar(12),

nombre varchar(80),

salario\_basico float

)

BEGIN

insert into empleado values(cedula, nombre, salario\_basico);

END//

DELIMITER ;



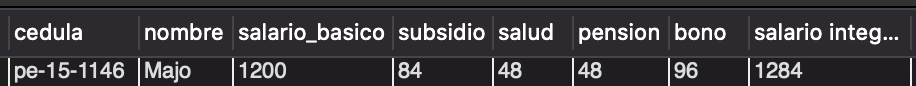
call insertarvalores('pe-15-1146', 'Majo', 1200);



select cedula, nombre, salario\_basico, calcsubsidio (salario\_basico )as 'subsidio', calcsalud (salario\_basico ) as 'salud',

calcpension (salario\_basico ) as 'pension' , calcbono (salario\_basico ) as 'bono',calcsal\_integral(salario\_basico) as 'salario integral'

from empleado;



**María José Martínez**

1. **la tabla empleados como ya existía se realiza un alter table add column**

Alter table empleado

add column subsidio float not null,

add column salud float not null,

add column pension float not null,

add column bono float not null,

add column salario\_integral float not null;



**Creación de una función para cada punto teniendo en cuenta que:**

* El subsidio de transporte equivale al 7% al salario básico

DELIMITER //

CREATE FUNCTION calcsubsidio (salariob float)

returns float deterministic

BEGIN

declare sub float ;

SET sub= salariob\*0.07;

RETURN sub;

END//

DELIMITER ;



* La salud que corresponde al 4% al salario básico

DELIMITER //

CREATE FUNCTION calcsalud (salariob float)

returns float deterministic

BEGIN

declare salud float ;

SET salud= salariob\*0.04;

RETURN salud;

END//

DELIMITER ;



* La pensión que corresponde al 4% al salario básico

DELIMITER //

CREATE FUNCTION calcpension (salariob float)

returns float deterministic

BEGIN

declare pension float ;

SET pension= salariob\*0.04;

RETURN pension;

END//

DELIMITER ;



* Un bono que corresponde al 8% al salario básico

DELIMITER //

CREATE FUNCTION calcbono (salariob float)

returns float deterministic

BEGIN

declare bono float ;

SET bono= salariob\*0.08;

RETURN bono;

END//

DELIMITER ;



* El salario integral es la suma del salario básico -salud - pension +bono + sub de

transporte.

DELIMITER //

CREATE FUNCTION calcsal\_integral(salariob float)

returns float deterministic

BEGIN

declare sal\_int, calcsubsidio, calcsalud, calcpension, calcbono float ;

SET sal\_int= salariob + calcsubsidio (salariob )+ calcsalud (salariob )+

calcpension (salariob )+ calcbono (salariob );

RETURN sal\_int;

END//

DELIMITER ;

**Cree un procedimiento que Inserte información a la tabla y pruebe las funciones.**

**Procedimiento:**

DELIMITER //

Create procedure insertarvalor (in

cedula varchar(12),

nombre varchar(80),

salario\_basico float

)

Begin

Insert into empleado

values(

cedula, nombre, salario\_basico,

calcsubsidio(salario\_basico), calcsalud(salario\_basico),

calcpension(salario\_basico), calcbono(salario\_basico),

calcsal\_integral(salario\_basico));

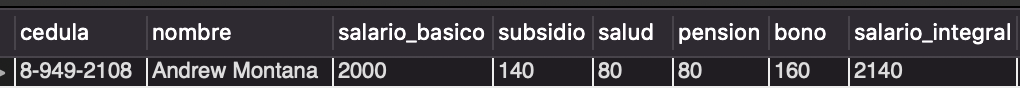
End //

DELIMITER ;



Call insertarvalor('8-949-2108', 'Andrew Montana', 2000.00);

select\* from empleado;



1. **CONSIDERACIONES FINALES – CONCLUSIONES:**

En este laboratorio se cumplió su objetivo de aplicar conceptos para crear procedimientos y funciones invocados por SQL, y pudimos enriquecer nuestras experiencias en el desarrollo y construcción de conocimientos del curso de Base de Datos II.

1. **BIBLIOGRAFIA:**

**El material de clase proporcionado por la profesora**

1. **RÚBRICAS:**

|  |  |
| --- | --- |
| ***Detalle - Evaluar*** | ***Valor de la respuesta*** |
| ***Sección G – (100 PUNTOS)*** | ***100 pts*** |