**Ejercicio 1:**

**Crea un procedimiento llamado busca que muestre todos los alumnos que su apellido tenga la cadena que se indique.**

Ejemplo de ejecución : CALL busca('R%');\*/

DELIMITER //

CREATE PROCEDURE busca(IN param VARCHAR(20))

BEGIN

SELECT \* FROM alumno WHERE apellido LIKE param;

END //

DELIMITER ;

CALL busca('R%');

**Ejercicio 2:**

**Crea un procedimiento que muestre los profesores por nombre y apellido. Si se deja alguno de los parámetros en blanco se busque por el otro**

**Ejemplo ejecucion: CALL busca2(NULL,'Ribas');\*/**

DELIMITER //

CREATE PROCEDURE busca2 (IN param VARCHAR(20), IN param2 VARCHAR(20))

BEGIN

IF (param IS NULL)

THEN

SELECT \* FROM profesor WHERE apellido LIKE param2;

ELSE IF (param2 IS NULL)

THEN

SELECT \* FROM profesor WHERE nom LIKE param;

ELSE

SELECT \* FROM profesor WHERE nom LIKE param AND apellido LIKE param2;

END IF;

END //

DELIMITER ;

CALL busca2(NULL, 'R%');

**Ejercicio 3:**

**Crea un procedimiento que incremente el valor del cod de las dos tablas con el valor que se le indique.**

**Ejemplo ejecucion: CALL actualitza(20);\*/**

DELIMITER //

CREATE PROCEDURE actualitza(IN par INT)

BEGIN

UPDATE alumno SET codi=(codi+par);

UPDATE profesor SET codi = (codi+par);

END //

DELIMITER ;

CALL actualitza(5);

**Ejercicio 4**

**Crea un procedimiento que dado un nombre, un apellido, la fecha nacimiento i si es profesor o alumno, lo añada a la taula correspondiente. Si no, que no haga nada.**

**Ejemplo ejecucion: CALL insertar('Xavier','Vila ','1975/02/03','profesor');**

DELIMITER //

CREATE PROCEDURE insertar (IN par\_nom VARCHAR(20), IN par\_cog VARCHAR(20), IN par\_dat DATE, IN par\_tipus VARCHAR(20))

BEGIN

CASE par\_tipus

WHEN 'profesor'

THEN

INSERT INTO profesor VALUES (NULL,par\_nom,par\_cog,par\_dat,NULL);

WHEN 'alumno'

THEN

INSERT INTO alumno VALUES(NULL,par\_nom,par\_cog,par\_dat,NULL);

END CASE;

END //

DELIMITER ;

CALL insertar('Pepe','Motos','1975/02/03','alumno');

**\*Ejercicio 5:**

**Crea un procedimiento que añada la fecha de nacimiento a los alumnos . Indicar el cod del alumno y la fecha de nacimiento.**

**Ejemplo ejecucion: CALL fecha\_nac(21,'1995/03/19');\*/**

DELIMITER //

CREATE PROCEDURE fecha\_nac(IN cod INT, IN dat DATE)

BEGIN

UPDATE alumno SET fecha\_nac = dat WHERE codi = cod LIMIT 1;

END //

DELIMITER ;

CALL fecha\_nac(6,'1999/03/19');

**/\*Ejercicio 6:**

**Crea un procedimiento que, dada la tabla que se le indique y el codigo de la persona, calcule la edad de una persona a partir de la fecha de nacimiento.**

**Ejemplo ejecucion: CALL edad('profesor',20);\*/**

DELIMITER //

CREATE PROCEDURE edad(IN tabla VARCHAR(20), IN cod INT)

BEGIN

IF (tabla = 'profesor')

THEN

SELECT YEAR(NOW())-YEAR(fecha\_nac)

FROM profesor WHERE cod = cod;

ELSE IF (tabla = 'alumno')

THEN

SELECT YEAR(NOW())-YEAR(fecha\_nac)

FROM alumno WHERE cod = cod;

END IF;

END //

DELIMITER ;

CALL edad('profesor',6);

**Ejercicio 7:**

**Crea un procedimiento que elimine los registros de la tabla alumnos que tengan el codigo entre los dos que se indican.**

**Ejemplo ejecucion: CALL elimina(20,22);\*/**

DELIMITER //

CREATE PROCEDURE elimina(IN par1 INT, IN par2 INT)

BEGIN

DECLARE par3 INT DEFAULT par1;

WHILE (par3<=par2) DO

DELETE FROM alumno WHERE codi = par3 ;

SET par3=par3+1;

END WHILE;

END //

DELIMITER ;

CALL elimina(6,8);

**bd empresa**

1. Crear un procedure en el que dada una factura, obtengamos el importe total de la factura en todas las líneas de factura.

**CREATE PROCEDURE** `**total\_factura**`(**IN** `fact` **INT**, **OUT** `total` **FLOAT**)

**BEGIN**

**select sum**(importe) **into** total **from** lineas\_factura **where** factura = fact;

**END**

**CALL** `total\_factura`('2', @y);

**select** @y;

2. Crear un procedure en el que dado un vendedor, obtengamos el importe total de todas sus facturas.

**CREATE PROCEDURE** `**total\_vendedores**`(**IN** `vend` **INT**, **OUT** `total` **FLOAT**)

**BEGIN**

**select sum**(importe) **into** total

**from** facturas, lineas\_factura

**where** facturas.vendedor=vend **and** lineas\_factura.factura = facturas.id;

**END**

**CALL** `total\_vendedores`('1', @x);

**select** @x;

3. Crear un procedure en el que dado un vendedor y una factura, obtengamos el

importe total de esa factura.

**CREATE PROCEDURE** `**total\_vendedores1**`(**IN** `vend` **INT**, **IN** `fact` **INT**, **OUT** `total` **FLOAT**)

**BEGIN**

**select sum**(importe) **into** total **from** lineas\_factura,facturas **where** lineas\_factura.factura =

fact **and** facturas.vendedor=vend **and** facturas.id=fact;

**END**

**CALL** `total\_vendedores1`('1','1',@x);

**select** @x;

4. Crear un procedure en el que mostremos el nº de clientes que han hecho compras en el mes de marzo.

(he modificado la solucion para que sea en un mes cualquiera)

**CREATE PROCEDURE** `**compras\_clientes**`(**IN** `mes` **INT**, **OUT** `num` **INT**)

**BEGIN**

**SELECT count**(\*) **INTO** num

**FROM** clientes **WHERE** Id **IN** (

**SELECT** cliente

**FROM** facturas

**WHERE month(**fecha)=mes);

**END**

**CALL** `compras\_clientes`('3', @x);

**select** @x;