

Ejercicio 1. Sobre la BD 'test', implementar los siguientes procedimientos:

- Utilizando el comando IF. Crear un procedimiento 'siDivisible' que reciba 2 números y muestre si el primero es divisible por el segundo. **Ejemplo**, si introducimos los números 4 y 2. El procedimiento deberá mostrar el mensaje: 4 es divisible por 2.

```
mysql> call siDivisible(4,2);
+-----+
| Resultado |
+-----+
| 4 es divisible por 2 |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql>
```

```
mysql> call siDivisible(5,2);
+-----+
| Resultado |
+-----+
| 5 NO es divisible por 2 |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql>
```

- Utilizando el comando CASE. Crear un procedimiento 'notasPrimaria' que reciba dos parámetros:
 - Número (nota de un alumno).
 - Nombre (nombre del alumno).

El procedimiento deberá mostrar un mensaje con el siguiente formato y según el siguiente baremo:

Entre 9 y 10 = Sobresaliente

Entre 7 y 8 = Notable

6 = Bien

5 = Suficiente

4-1 = Insuficiente

NOTA: Tener en cuenta que el usuario puede introducir una nota que no sea un valor entre 1-10, en ese caso el procedimiento deberá mostrar el siguiente mensaje: 'Nota incorrecta, debe introducir una nota entre 1 y 10'.

Ejemplo, suponemos que introducimos la nota 8,5 para Laura Pérez, el procedimiento mostrará: 'La nota de Laura Pérez es NOTABLE'.

```
mysql> call notasPrimaria(8.5,'Laura Pérez');
+-----+
| Nota |
+-----+
| Laura Pérez tiene un NOTABLE |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

mysql>
```

```
mysql> call notasPrimaria(0.0,'Laura Pérez');
+-----+
| Nota |
+-----+
| Nota incorrecta, debe introducir una nota 1 y 10 |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql>
```

- Utilizando el bucle LOOP. Crear un procedimiento que muestre la suma de los números pares comprendidos entre 1-20 excepto el 10 (2-4-6-8-12-14-16-18-20). SUMA: 100

```
mysql> call bucleLOOP();
+-----+
| SUMA |
+-----+
| SUMA: 100 |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.02 sec)

mysql>
```

4. Utilizando el bucle REPEAT. Crear un **procedimiento** que muestre la suma de los números pares comprendidos entre 100-90 (100-98-96-94-92-90). SUMA: 570

```
mysql> call bucleREPEAT();
+-----+
| SUMA   |
+-----+
| SUMA: 570 |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.02 sec)

mysql>
```

5. Utilizando el bucle WHILE. Crear un **procedimiento** que muestre la suma de los 100 primeros números naturales (1-2.....99-100). SUMA: 5050

```
mysql> call bucleWHILE();
+-----+
| SUMA   |
+-----+
| SUMA: 5050 |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql>
```

6. Crear un **procedimiento** 'operacionesAritméticas', que reciba 2 números enteros o decimales y que muestre (select) la suma, resta, producto, división;

```
mysql> call operacionesAritmeticas(5.5,10);
+-----+
| SUMA   |
+-----+
| La suma de: 5.50 + 10.00 es: 15.50 |
+-----+
1 row in set (0.02 sec)

+-----+
| RESTA  |
+-----+
| La resta de: 5.50 - 10.00 es: -4.50 |
+-----+
1 row in set (0.02 sec)

+-----+
| PRODUCTO |
+-----+
| El producto de : 5.50 * 10.00 es: 55.0000 |
+-----+
1 row in set (0.02 sec)

+-----+
| DIVISION |
+-----+
| La division entre: 5.50 y 10.00 es: 0.550000 |
+-----+
1 row in set (0.03 sec)

Query OK, 0 rows affected (0.03 sec)
```

```
mysql> call operacionesAritmeticas(5.5,0);
+-----+
| SUMA   |
+-----+
| La suma de: 5.50 + 0.00 es: 5.50 |
+-----+
1 row in set (0.00 sec)

+-----+
| RESTA  |
+-----+
| La resta de: 5.50 - 0.00 es: 5.50 |
+-----+
1 row in set (0.00 sec)

+-----+
| PRODUCTO |
+-----+
| El producto de : 5.50 * 0.00 es: 0.0000 |
+-----+
1 row in set (0.02 sec)

+-----+
| DIVISION |
+-----+
| La division entre: 5.50 y 0.00 es: INDETERMINADA |
+-----+
1 row in set (0.02 sec)

Query OK, 0 rows affected (0.03 sec)

mysql>
```

7. Crear un **procedimiento** `'primeros20Impares'` que muestre los 20 primeros números impares.

```
mysql> call primeros20Impares();
+-----+
| R                                           |
+-----+
| Primeros 20 números impares: 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

mysql>
```

8. Crear un **procedimiento** `'fechaLarga'` que reciba una fecha (dd-mm-aaaa) y muestre la fecha en formato largo. Por ejemplo: 31-10-2014 deberá mostrar: Viernes, 31 de Octubre de 2014. Puede resultar interesante crear sendos procedimientos que retornen el día y mes en castellano/gallego.

Funciones útiles:

`dayname(fecha)`→devuelve el día de la semana en letra.

`month(fecha)`→devuelve el mes en número.

```
mysql> call fechaLarga('2014/10/31');
+-----+
| FECHALARGA                                |
+-----+
| Viernes, 31 de Octubre de 2014          |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql>
```

Ejercicio 2. Sobre la BD 'kadoo', implementar los siguientes procedimientos:

9. Crear un **procedimiento** de nombre **`empleadosPorDpto`** que acepte *un número de departamento* como argumento y visualice todos los datos de los empleados que en él trabajan. Ejemplo de funcionamiento.

```
mysql> call empleadosPorDpto(100);
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| NUMEM | NUMDE | EXTEL | FECNA   | FECIN   | SALAR   | COMIS | NUMHI | NOMEM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 210   | 100   | 200   | 1960-09-28 | 1979-01-22 | 2283.85 | NULL  | 2     | PILAR |
| 250   | 100   | 250   | 1966-10-27 | 1987-03-01 | 2704.55 | NULL  | 0     | ADRIANA |
| 260   | 100   | 220   | 1963-12-03 | 1988-07-12 | 4327.29 | NULL  | 6     | ANTONIO |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

Query OK, 0 rows affected (0.03 sec)

mysql>
```

10. Crear un procedimiento de nombre **empleadosPorCentro** que acepte un *número de centro* como argumento y visualice todos los datos de los empleados que en él trabajan. Ejemplo de funcionamiento.

```
mysql> call ejercicio2_2(10);
```

NUMCE	NUMEN	NUMDE	EXTTEL	FECNA	FECIN	SALAR	COMIS	NUMHI	NOHEM
10	210	100	200	1960-09-28	1979-01-22	2283.85	NULL	2	PILAR
10	250	100	250	1966-10-27	1987-03-01	2704.55	NULL	0	ADRIANA
10	260	100	220	1963-12-03	1988-07-12	4327.29	NULL	6	ANTONIO
10	290	120	910	1967-10-30	1988-02-14	1622.73	NULL	3	GLORIA
10	110	121	350	1949-11-10	1970-02-15	1863.14	NULL	3	CESAR
10	150	121	340	1950-08-10	1968-01-15	2644.45	NULL	0	JULIO
10	190	121	350	1952-05-12	1982-02-11	1803.04	NULL	4	JULIANA
10	370	121	360	1987-06-22	2007-01-20	1141.92	NULL	1	FABIOLA
10	285	122	620	1969-10-25	1988-02-15	2283.85	NULL	0	OTILIA
10	320	122	620	1977-12-25	1998-02-05	2434.10	NULL	2	CORNELIO
10	350	122	610	1969-04-13	2004-09-10	2704.55	NULL	1	AURELIO
10	410	122	660	1988-07-14	2008-10-13	1051.77	NULL	0	AZUCENA
10	430	122	650	1987-02-26	2006-11-19	1262.13	NULL	1	VALERIANA
10	280	130	410	1968-01-11	1991-10-08	1742.94	NULL	5	DOROTEA
10	310	130	480	1966-11-21	1991-01-15	2524.25	NULL	0	AUGUSTO
10	420	130	450	1986-10-22	2008-11-19	2404.05	NULL	0	CLAUDIA

```
16 rows in set (0.00 sec)

Query OK, 0 rows affected (0.09 sec)

mysql>
```

Sí el centro no existe

```
mysql> call ejercicio2_2(100);
```

datos
El Centro 100 no existe

```
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

mysql>
```

11. Crear un procedimiento de nombre **dirpresuPorDpto** que acepte un *número de departamento* y devuelva mediante dos parámetros de tipo OUT su número de *director* y *presupuesto*. Ejemplo de funcionamiento.

```
mysql> call dirpresuPorDpto(100, @director, @presupuesto);
Query OK, 1 row affected (0.02 sec)

mysql> select @director, @presupuesto;
```

@director	@presupuesto
260	1200000.00

```
1 row in set (0.00 sec)

mysql>
```

Sí el departamento no existe

```
mysql> call dirpresuPorDpto(10, @director,@presupuesto);
Query OK, 0 rows affected (0.00 sec)

mysql> select @director,@presupuesto;
+-----+-----+
| @director | @presupuesto |
+-----+-----+
|          0 |          0.00 |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```