

Implementación de la base de datos de DataSafe, empresa de seguridad bancaria

Código del proyecto: ICX0_P3

Producto 5 - Procedimientos y disparadores Parte 1

Alumnos:	Ana Real Tovar Jesús Real Tovar Raúl Vargas Molinero Marc Sánchez Sanz Enric Loren Parrondo
Consultor/ra:	Rita de la Torre Chirivella
Fecha de entrega:	17 de mayo de 2019

1. Realizar los siguientes procedimientos almacenados y Funciones.

1.1. Un procedimiento almacenado que retorne el número de trabajadores fijos que hay en la entidad bancaria.

```
1 DELIMITER //
```

```
2 CREATE PROCEDURE empleadosFijos (OUT empleados INT)
```

```
3 BEGIN
```

```
4     SELECT COUNT(*) INTO empleados FROM fijo;
```

```
5 END
```

```
6 //
```

```
1 CALL empleadosFijos (@numero_empleados_fijos);
```

```
2 SELECT @numero_empleados_fijos;
```

	@numero_empleados_fijos
▶	6

1.2. Una función que devuelva el doble del tipo de interés de los préstamos.

```
1 DELIMITER //
```

```
2 CREATE FUNCTION dobleInteres (id INT) RETURNS DOUBLE DETERMINISTIC
```

```
3 BEGIN
```

```
4     SET @var = 0;
```

```
5     SELECT tipo_interes*2 INTO @var FROM tipoprestamo WHERE tipoprestamo.id = id;
```

```
6     RETURN @var;
```

```
7 END
```

```
8 //
```

```
1 SELECT dobleInteres(1);
```

	dobleInteres(1)
▶	0.6

1.3. Inventar una función relacionada con la seguridad en el banco.

```
1 /* Valida un empleado según su código de empleado, dni y nss */
```

```
2 DELIMITER //
```

```
3 CREATE FUNCTION validaEmpleado (codigo_empleado VARCHAR(8), dni VARCHAR(10),
```

```
4 ↪ nss VARCHAR(20) ) RETURNS TINYINT(1) DETERMINISTIC
```

```
5 BEGIN
```

```
6     declare validado TINYINT(1);
```

```
7     IF (SELECT COUNT(e.codigo_empleado) FROM empleado AS e WHERE e.
```

```
8     ↪ codigo_empleado LIKE codigo_empleado AND e.dni LIKE dni AND e.nss LIKE nss ) THEN
```

```
9         SET validado = 1;
```

```
10    ELSE
```

```
11        SET validado = 0;
```

Realizar los siguientes procedimientos almacenados y Funciones.

```
10     END IF;
11     RETURN validado;
12 END
13 //
```

```
1 SELECT validaEmpleado('ars11234', '36584125P', '7279568404') AS valido;
```

	valido
▶	1

1.4. Inventar un procedimiento almacenado que tenga un parámetro de entrada.

```
1 DELIMITER //
2 CREATE PROCEDURE insertaOActualizaTitulo(IN nombre_titulo VARCHAR(20), IN
↪ numero DECIMAL(10,0))
3 BEGIN
4     IF (SELECT COUNT(t.nombre_titulo) FROM titulo AS t WHERE t.nombre_titulo LIKE
↪ nombre_titulo) THEN
5         UPDATE titulo AS tt
6         SET tt.numero = numero
7         WHERE tt.nombre_titulo LIKE nombre_titulo;
8     ELSEIF (SELECT COUNT(t.nombre_titulo) FROM titulo AS t WHERE t.numero LIKE
↪ numero) THEN
9         UPDATE titulo AS tt
10        SET tt.nombre_titulo = nombre_titulo
11        WHERE tt.numero = numero;
12    ELSE
13        INSERT INTO titulo (nombre_titulo, numero)
14        VALUES (nombre_titulo, numero);
15    END IF;
16 END
17 //
```

```
1 CALL insertaOActualizaTitulo('PGTH', '14000');
2 SELECT * FROM icxp3_7.titulo;
```

	id	nombre_titulo	numero
▶	1	E.S.O.	3000
	2	Bachiller	4000
	3	CFGS_DAM	5000
	4	CFGS_DAW	5001
	5	CFGS_ASIR	5002
	6	CFGS_AF	6000
	7	CFGS_MP	7000
	8	CFGS_GVEC	8000
	9	G_ADE	12000
	10	G_II	13000
	11	PGTH	14000