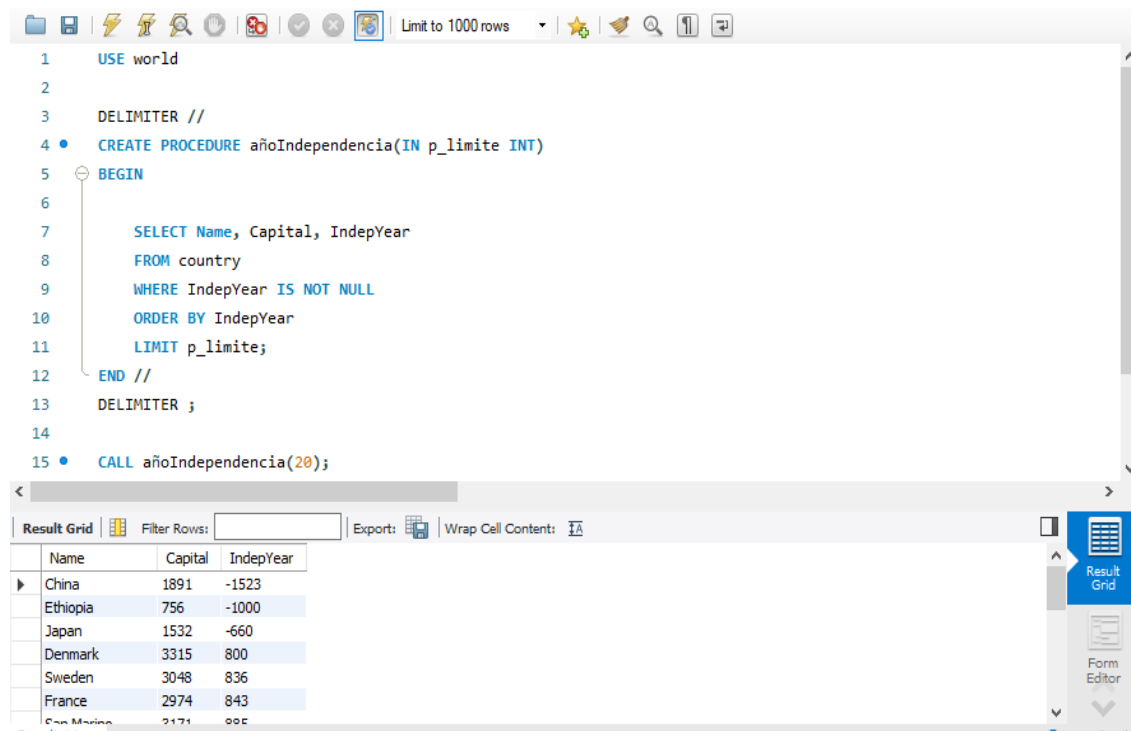


Ejercicios en Store Procedures

Ejercicio 33: Listar los nombres, las capitales y el año de independencia (sin nulos) de los 20 países más antiguos. (Se esperan 3 columnas y 20 registros).



The screenshot shows a SQL IDE interface. The top pane contains a SQL script for a stored procedure named `añoIndependencia`. The script uses the `world` database, sets the delimiter to `//`, and creates the procedure to select the first 20 countries from the `country` table, ordered by their independence year. The bottom pane shows the results of the procedure call, displaying a table with 3 columns: `Name`, `Capital`, and `IndepYear`. The results are sorted by `IndepYear` in ascending order.

```
1  USE world
2
3  DELIMITER //
4  CREATE PROCEDURE añoIndependencia(IN p_limite INT)
5  BEGIN
6
7      SELECT Name, Capital, IndepYear
8      FROM country
9      WHERE IndepYear IS NOT NULL
10     ORDER BY IndepYear
11     LIMIT p_limite;
12 END //
13 DELIMITER ;
14
15 CALL añoIndependencia(20);
```

Name	Capital	IndepYear
China	1891	-1523
Ethiopia	756	-1000
Japan	1532	-660
Denmark	3315	800
Sweden	3048	836
France	2974	843
San Marino	2171	895

Ejercicio 51: Insertar un nuevo cliente con los siguientes datos:

'12169851', 'Luis Enrique', 'Pérez', 'HOMBRE', '1954-06-01',
'perez_luisenrique@ma1l.com', '1157319468', 'Pujol 416', 'Caballito',
'Uruguay'

Insertar o actualizar.

```
1  DELIMITER $$
2
3  CREATE PROCEDURE spInsertarActualizarCliente(
4      _dni CHAR(8),
5      _nombre VARCHAR(50),
6      _apellido VARCHAR(50),
7      _genero ENUM('HOMBRE', 'MUJER'),
8      _fecha_nacimiento DATE,
9      _mail VARCHAR(100),
10     _movil VARCHAR(15),
11     _domicilio VARCHAR(255),
12     _barrio VARCHAR(50),
13     _nacionalidad VARCHAR(50),
14     OUT estado TINYINT
15 )
16 BEGIN
17     DECLARE dni_count INT;
18
19     -- Verificar si el cliente con el DNI proporcionado ya existe
20     SELECT COUNT(*) INTO dni_count FROM inupde.clientes WHERE dni = _dni;
21
22     IF dni_count = 1 THEN
23         -- El cliente existe, realizar la ACTUALIZACIÓN
24
25         UPDATE inupde.clientes
26         SET
27             nombre = _nombre,
28             apellido = _apellido,
29             genero = _genero,
30             fecha_nacimiento = _fecha_nacimiento,
31             mail = _mail,
32             movil = _movil,
33             domicilio = _domicilio,
34             barrio = _barrio,
35             nacionalidad = _nacionalidad
36         WHERE dni = _dni;
37
38         SET estado = 1; -- Cliente existente. Aplicar ACTUALIZACIÓN
39     ELSE
40         -- El cliente no existe, realizar la INSERCIÓN
41         INSERT INTO inupde.clientes (
42             dni, nombre, apellido, genero, fecha_nacimiento,
43             mail, movil, domicilio, barrio, nacionalidad
44         )
45         VALUES (
46             _dni, _nombre, _apellido, _genero, _fecha_nacimiento,
47             _mail, _movil, _domicilio, _barrio, _nacionalidad
48         );
```

```

48
49         SET estado = 2; -- Nuevo cliente. Aplicar INSERCIÓN
50     END IF;
51 END $$
52 DELIMITER ;
53 • USE inupde;
54 • SET @dni = '12169851';
55 • SET @nombre = 'Luis Enrique';
56 • SET @apellido = 'Pérez';
57 • SET @genero = 'HOMBRE';
58 • SET @fecha_nacimiento = '1954-06-01';
59 • SET @mail = 'perez_luisenrique@mail.com';
60 • SET @movil = '1157319468';
61 • SET @domicilio = 'Pujol 416';
62 • SET @barrio = 'Caballito';
63 • SET @nacionalidad = 'Uruguay';
64
65 • CALL spInsertarActualizarCliente(
66     @dni, @nombre, @apellido, @genero, @fecha_nacimiento,
67     @mail, @movil, @domicilio, @barrio, @nacionalidad, @retorno
68 );
69
70 • SELECT @retorno;
71

```

```
70 • SELECT @retorno;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
@retorno	1			

Ejercicio 54: Modificar el apellido de la cliente 85 por 'Chaves'. (Se espera 1 registro afectado).

Actualización con estado.

```

1  DELIMITER $$
2  • CREATE PROCEDURE spActualizarApellidoCliente(
3      _idCliente INT,
4      _nuevoApellido VARCHAR(50),
5      OUT estado TINYINT
6  )
7  BEGIN
8      -- Verificar si el cliente con el ID proporcionado existe
9      DECLARE cliente_count INT;
10     SELECT COUNT(*) INTO cliente_count FROM inupde.clientes WHERE idCliente = _idCliente;
11
12     IF cliente_count = 1 THEN
13         -- El cliente existe, realizar la ACTUALIZACIÓN del apellido
14         UPDATE inupde.clientes
15         SET apellido = _nuevoApellido
16         WHERE idCliente = _idCliente;
17
18         SET estado = 1; -- Actualización exitosa
19     ELSE
20         -- El cliente no existe
21         SET estado = 0; -- Cliente no encontrado
22     END IF;
23     END
24     <<
27 • USE inupde;
28 • SET @idCliente = 85;
29 • SET @nuevoApellido = 'Chaves';
30 • CALL spActualizarApellidoCliente(@idCliente, @nuevoApellido, @retorno);
31 • SELECT @retorno;

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	@retorno			
▶	1			

Ejercicio 58: Eliminar al cliente 9804. (Se espera 1 registro afectado).

Eliminación sin estado.

```
DELIMITER $$  
CREATE PROCEDURE spEliminarCliente(  
    _idCliente INT  
)  
BEGIN  
    -- Eliminar al cliente con el ID proporcionado  
    DELETE FROM inupde.clientes  
    WHERE idCliente = _idCliente;  
END $$  
DELIMITER ;  
  
USE inupde;  
SET @idClienteEliminar = 9804;  
CALL spEliminarCliente(@idClienteEliminar);
```