

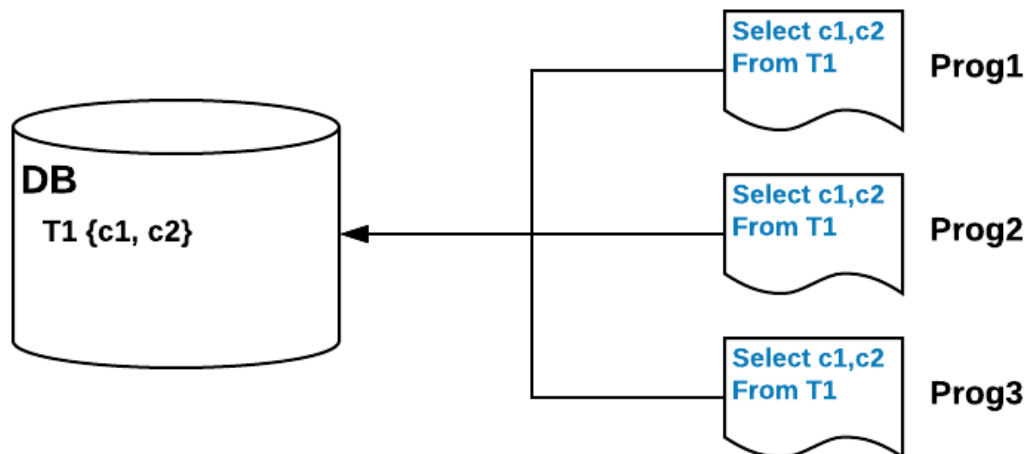
PROCEDIMIENTOS ALMACENADOS

Problema

Cuando una aplicación (sistema Web o de cualquier tipo) hace uso de una Base de Datos para consultar, insertar, actualizar o borrar; en la aplicación debe escribirse el código SQL que accederá a la BD. Si múltiples aplicaciones desean hacer las mismas consultas, estos códigos deben escribirse en cada aplicación.

¿Qué pasa si yo cambio, por ejemplo, un nombre de un campo en mi BD (por ejemplo, *Fecha_Nacimiento* lo cambio a *FechNac*)?

Entonces todas las aplicaciones deberían actualizar su consulta. Qué tedioso!!
En la Figura se aprecia esta situación problemática

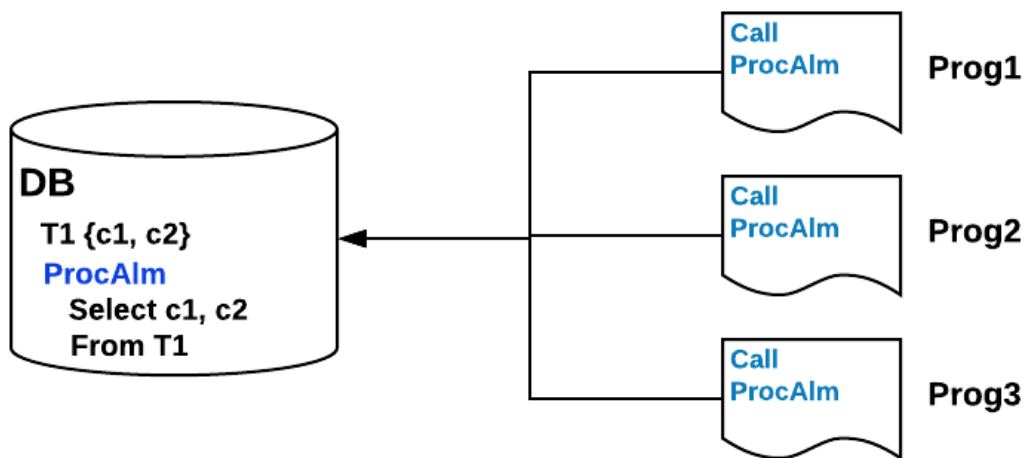


Los 3 programas Prog1, Prog2 y Prog 3, hacen la misma consulta a la base de datos. Si por necesidad esta consulta debe cambiar, por ejemplo que *c1* se llame ahora *c1New*, entonces habrá que editar los 3 programas para hacer el cambio en cada uno de los SELECT.

Solución

Escribir un procedimiento (o función) en la misma BD el cual se encargue de realizar la consulta. Entonces, las aplicaciones (programas) lo que hacen ahora es llamar a ese procedimiento. Así, si cambia el nombre de un campo en la BD, solamente actualizo mi procedimiento, sin tocar los programas.

La figura muestra la nueva situación, ahora con procedimientos almacenados.



Podemos ver que si **c1** llegara a cambiar a **c1New**, entonces habría solo que actualizar el Procedimiento Almacenado.

Esto también da mayor seguridad, ya que los procedimientos almacenados se ejecutan en el servidor de BD, lo cual hace más difícil que el sistema sea vulnerado.

Un procedimiento almacenado puede ser simple (ejm. `SELECT * FROM Clientes`), como también podría ser muy complejo (más de 100 o 1000 líneas de código). Entonces, al incluir este código directamente en la BD la eficiencia y seguridad aumentan notablemente. Hay sistemas donde todas las consultas se escriben en procedimientos almacenados, y en las aplicaciones simplemente se llama a estos procedimientos.

Creando un Procedimiento Almacenado en MySql

La sintaxis es

```
CREATE PROCEDURE Nombre(parámetros)
Aquí va el código/consulta del procedimiento
almacenado
```

Ejemplo

```
CREATE PROCEDURE MuestraEmpleados()
SELECT * FROM empleados
WHERE Departamento="Finanzas"
```

Ejecutando un Procedimiento almacenado

Para ejecutar un procedimiento almacenado, la sintaxis es:

```
CALL Nombre (parámetros)
```

Para el ejemplo anterior, la llamada sería:

```
CALL MuestraEmpleados ()
```

Este es un procedimiento almacenado que no tiene parámetros.

Usando parámetros

Los parámetros permiten hacer más flexible un procedimiento almacenado. Por ejemplo, el procedimiento anterior solo muestra los empleados de *Finanzas*. Podríamos, por ejemplo, hacer que este procedimiento muestre los empleados de distintos Departamentos, pasando como parámetro el Nombre del Departamento que queremos listar. De este modo, el ejemplo anterior, con parámetros, nos quedaría de la siguiente manera:

```
CREATE PROCEDURE MuestraEmpleados (NomDepto VARCHAR (30) )  
SELECT * FROM empleados  
WHERE Departamento = NomDepto;
```

Un ejemplo de llamada a este procedimiento almacenado sería:

```
CALL MuestraEmpleados ("Marketing")
```

Ejercicio

Crear un procedimiento almacenado que permita aumentar el sueldo en un porcentaje a todos los empleados de un departamento cualquiera.

En este caso, hay dos parámetros que manejar. Por un lado el **porcentaje** de aumento de sueldo; y por otro, el **departamento** al cual pertenecen los empleados que recibirán el aumento. De este modo, el procedimiento queda:

```
CREATE PROCEDURE ActualizaSueldo (Porcentaje INT,  
                                  NomDepto VARCHAR (30) )  
  
UPDATE empleados  
SET      sueldo = sueldo * (1 + Porcentaje)  
WHERE    Departamento = NomDepto;
```

Refuerce este contenido con el siguiente video

[https://www.youtube.com/embed/ Gy8-hCA8a0](https://www.youtube.com/embed/Gy8-hCA8a0)