

# Rutinas



- Funciones (Functions)
- Procedimientos Almacenados (Stored Procedures)
- Disparadores (Triggers)

# Procedimientos Almacenados

Stored Procedures

# Procedimientos almacenados

- Un procedimiento es un conjunto de instrucciones que se guardan en el servidor para un posterior uso, ya que se ejecutarán frecuentemente.
- A diferencia de las funciones, los procedimientos son rutinas que no regresan ningún tipo de valor.
- Simplemente se llaman desde el cliente con un comando y las instrucciones dentro del procedimiento se ejecutarán.
- La llamada es mediante la instrucción:
  - `Call procedimiento([params]);`

# Procedimientos almacenados

- Ventajas de usar Procedimientos en MySQL
- **Seguridad:** Los procedimientos ocultan el nombre de las tablas a usuarios que no tengan los privilegios para manipular datos. Simplemente llaman los procedimientos sin conocer la estructura de la base de datos.
- **Estándares de código:** En un equipo de desarrollo usar el mismo procedimiento permite crear sinergia en las fases de construcción. Si cada programador crea su propio procedimiento para realizar la misma tarea, entonces podrían existir problemas de integridad y pérdida de tiempo
- **Velocidad:** Es mucho más fácil ejecutar un programa ya definido mediante ciertos parámetros, que reescribir de nuevo las instrucciones.

# Procedimientos almacenados

Sintaxis:

**CREATE PROCEDURE** nombre ([parámetro1,parámetro2,...])

**BEGIN**

instrucciones

**END**

Los procedimientos almacenados pueden o no recibir parámetros

# Procedimientos almacenados

Los parámetros pueden ser de tipo:

- **IN:** Datos de sólo entrada, es un dato que debe ser introducido en la llamada del procedimiento para definir alguna acción del bloque de instrucciones.
- **OUT:** parámetro de salida es un espacio de memoria en el cual el procedimiento devolverá almacenado su resultado.
- **INOUT:** parámetro de entrada/salida.

Por defecto, si no se indica el tipo de parámetro, MySQL asigna IN.

# Procedimientos almacenados

Ejemplos: (base de datos pixup)

## 1. Procedimiento almacenado para buscar artistas por palabra clave

```
Delimiter $$  
CREATE PROCEDURE busca_artista(palabra char(3))  
begin  
select * from artista where nombre like concat("%",palabra,"%");  
end;  
$$  
Delimiter ;
```

Ejecutar el procedimiento:

```
call busca_artista('los');
```

# Procedimientos almacenados

## 2. Procedimiento almacenado para buscar artistas por letra inicial

```
Delimiter $$  
CREATE PROCEDURE busca_artistaInicial(IN letra char(1))  
begin  
select * from artista where nombre like concat(letra,"%");  
end;  
$$  
Delimiter ;
```

Ejecutar el procedimiento:

```
call busca_artistaInicial('z');
```



# Procedimientos almacenados

3. Procedimiento almacenado para buscar artistas que inicien con alguna letra específica y que diga cuantos encontró

```
Delimiter //
```

```
create procedure busca_artista2(IN letra char(3), OUT total int)
```

```
begin
```

```
select * from artista where nombre like concat(letra,"%");
```

```
select count(*) into total from artista where nombre like
```

```
concat(letra,"%");
```

```
end;
```

```
//
```

```
Delimiter ;
```

Ejecutar el procedimiento:

```
call busca_artista2('m',@cantidad);
```

```
select @cantidad;
```

# Procedimientos almacenados

## 4. Procedimiento almacenado para mostrar lista de discos con disquera

```
delimiter //  
create procedure muestra_discos()  
begin  
SELECT id_disco,disco,artista,nombre as disquera  
FROM disquera  
JOIN v_infoDiscos USING(id_disquera)  
ORDER BY 2;  
end;  
//  
delimiter ;
```

Ejecutar el procedimiento:

```
call muestra_discos();
```

# Procedimientos almacenados

5. Procedimiento almacenado para actualizar precio de carritos según el modelo (usar la base de datos collectionhw)

```
Delimiter //  
create procedure actualizar_precios(anyo int, porcentaje int)  
Begin  
-- alter table carritos add precioNuevo decimal(10,2) after precio;  
update carritos set precioNuevo = precio * (100-porcentaje)/100 where  
modelo = anyo;  
end;  
//  
Delimiter ;
```

Ejecutar el procedimiento:

```
call actualizar_precios(2011,10);
```

Listar los carritos 2005 y 2011

```
select sku,nombre,precio,precioNuevo,modelo from carritos where modelo in (2005,2011);
```

¿Preguntas?