Laboratorio de Computación III

Vistas, procedimientos almacenados y funciones de usuario

Vistas

- Son elementos que se almacenan en nuestra base de datos.
- Son "tablas virtuales" cuya información está definida por una consulta de selección.
- A una vista se le puede realizar una consulta de selección con todas las características que vimos en clases previas.

Vistas

Creación

```
Create View nombre_vista AS
Select col1, col2, col3 From Tabla1
Inner Join . . .
Where  . . .
```

Modificación

```
Alter View nombre_vista AS
Select col1, col2, col3 From Tabla1
Inner Join . . .
Where  . . .
```

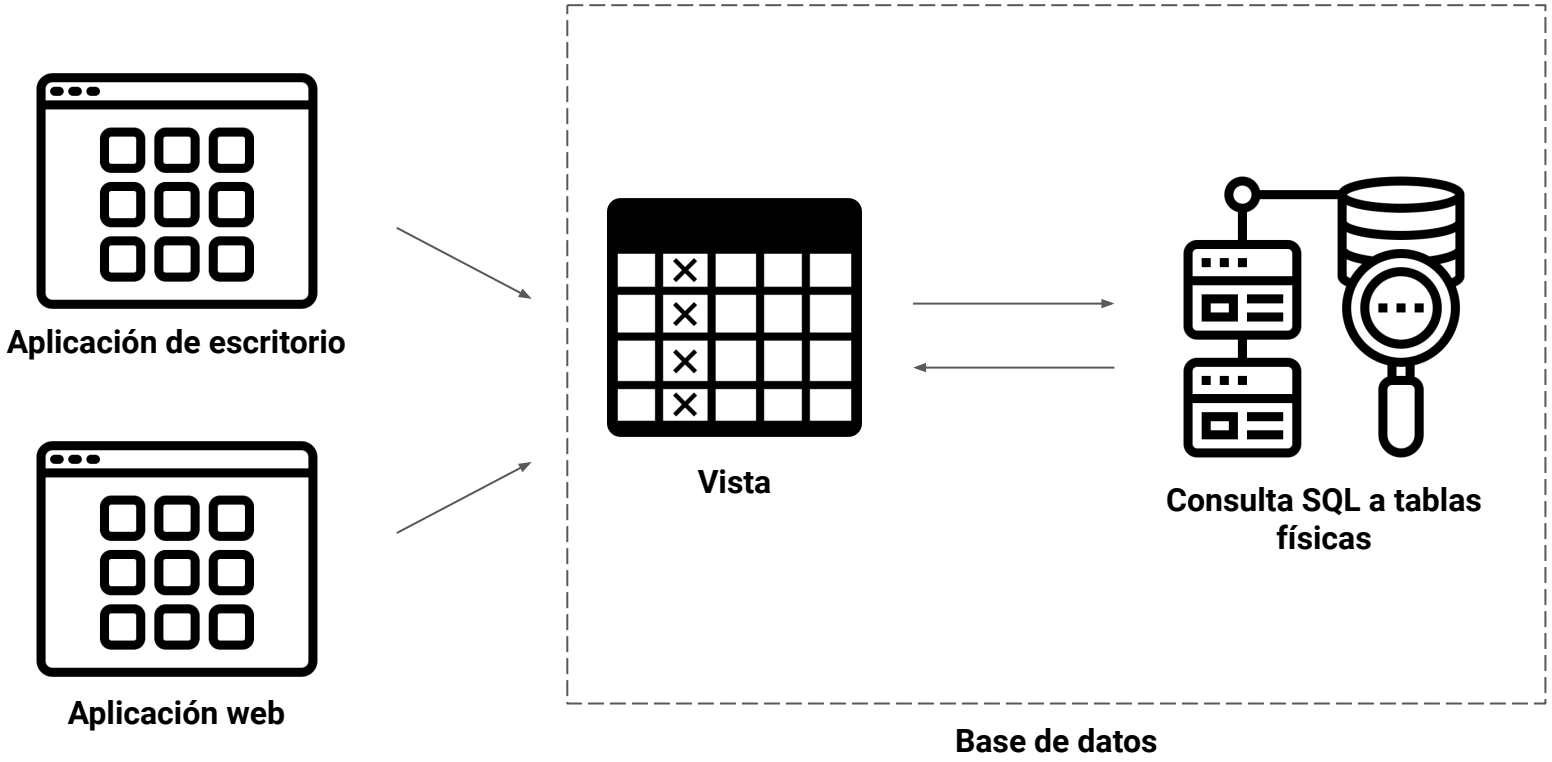
Eliminación

```
Drop View nombre_vista
```

Ventajas de usar vistas

- Simplifican consultas complejas. Abstrayéndolas de la complejidad de la base de datos y su normalización.
- Proporcionan un mecanismo de acceso a la información sin la necesidad de consultar directamente las tablas físicas de la base de datos.
- El impacto de la modificación de una vista es inmediato tanto en la base de datos como en las aplicaciones que las consumen.

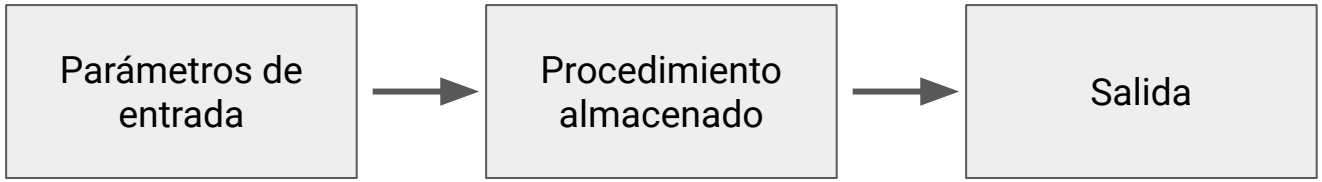
Esquema de trabajo con una vista



Procedimientos almacenados

- Un procedimiento cuyo algoritmo permanece encapsulado. Tanto su código como sus reglas de negocio se encuentran alojados como objeto de la base de datos.

- Como toda función, pueden recibir parámetros y devolver resultados. Éstos pueden ser valores escalares o sets de datos de una consulta de selección.



Procedimientos almacenados

La sintaxis básica para crear un procedimiento almacenado que no recibe parámetros es la siguiente:

```
Create Procedure nombre_procedimiento
As
Begin
     Select * From Tabla
End
```

Su ejecución se realiza de la siguiente manera:

```
Exec nombre_procedimiento
```

Procedimientos almacenados

La sintaxis básica para crear un procedimiento almacenado que recibe parámetros es la siguiente:

```
Create Procedure nombre_procedimiento(
  @param1 int,
  @param2 varchar(30)
)
as
begin
  update Tabla Set Col1 = @param2 where ID = @param1
end
```

Su ejecución se realiza de la siguiente manera:

```
Exec nombre_procedimiento 100, 'Algún string'
```

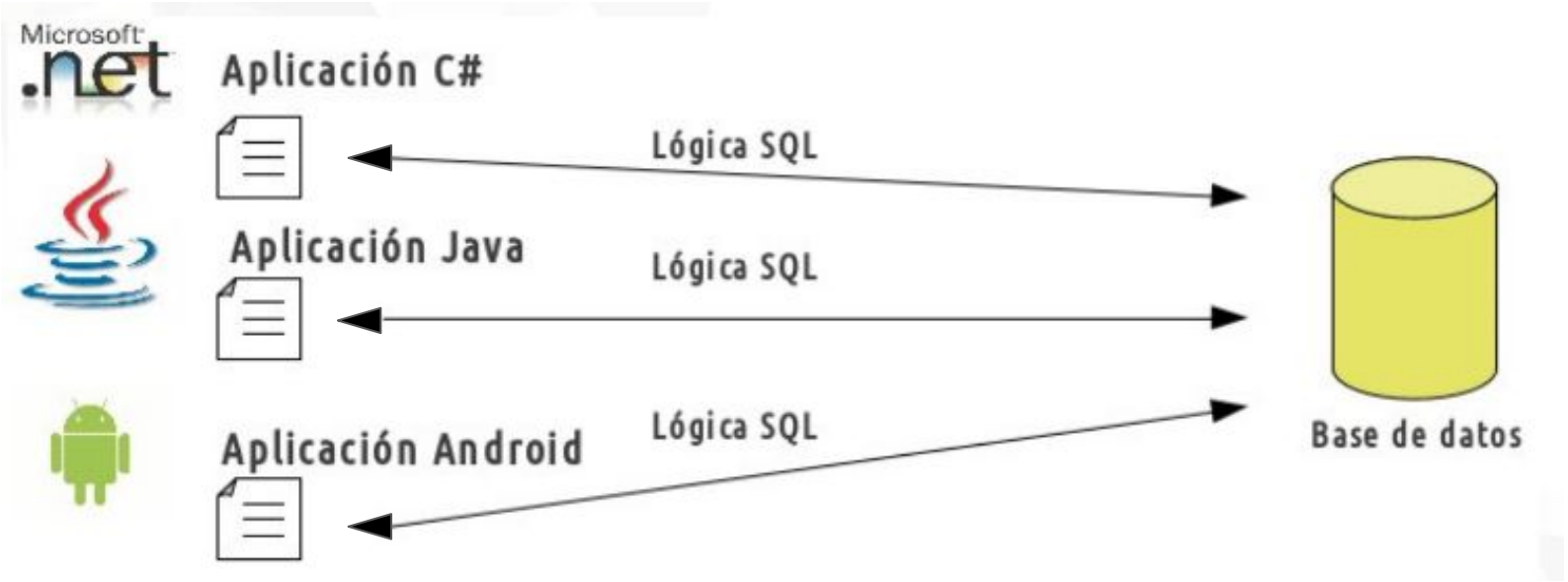
Ventajas de usar procedimientos almacenados

- Como tanto el procedimiento almacenado como los datos que manipulan están en la base de datos. El acceso entre ellos es inmediato.

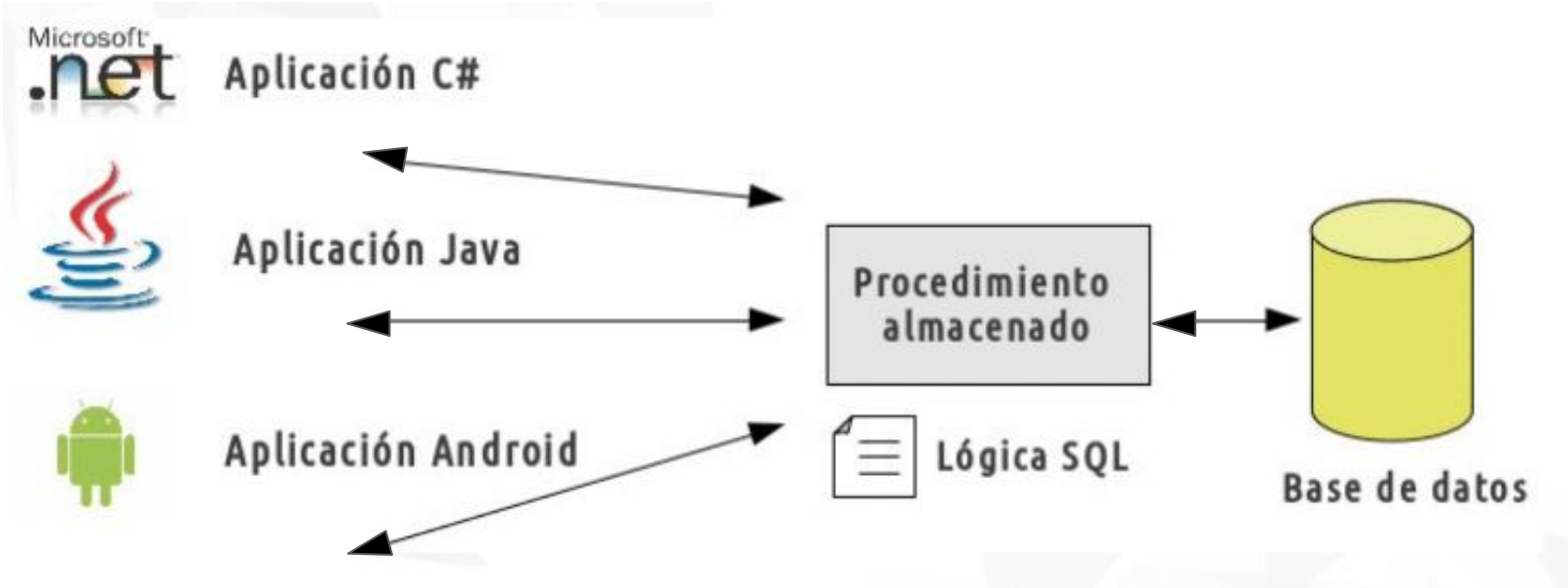
- **Se centraliza la lógica de negocio en la base de datos y no en las aplicaciones.**

- Provee una interfaz encapsulada para la ejecución de un proceso. Sólo es necesario enviar los parámetros del procedimiento y analizar su valor de retorno.

Esquema de trabajo sin procedimientos almacenados



Esquema de trabajo con procedimientos almacenados



Funciones de usuario

- Permite crear una función de usuario que puede ser utilizada como las funciones del lenguaje T-SQL.
- Pueden ser utilizadas en sentencias SQL (Select, Update, Delete, Insert).
- Pueden recibir parámetros y devolver un valor escalar. Existen funciones de usuario que pueden devolver tablas.
- No admiten la modificación de los datos de la base de datos. Es decir, son de sólo lectura.

Funciones de usuario

La sintaxis básica para crear una función de usuario que recibe parámetros es la siguiente:

```
Create Procedure nombre_funcion(
  @param1 int
)
as
begin
  select count(*) tabla where ID = @param1
end
```

Su utilización podría realizarse de la siguiente manera:

```
SELECT  Nombre, dbo.nombre_funcion(ColumnaX) From MiTabla
```

Ejemplos