

# Programación en Bases de Datos

Informática 5  
Universidad Católica Ntra. Sra. de la Asunción

# Programación Base de Datos

---

- Para programar en las bases de datos los SGDBs proveen del lenguaje SQL (Structured Query Language), el cual se divide en 2 sublenguajes:
  - DDL : Data Definition Language  
Create, Alter, Drop
  - DML : Data Manipulation Language.  
Select, Insert, Update, Delete
- Los objetos que se utilizan para la programación son los cursores, vistas, tablas temporales, procedimientos almacenados, funciones y triggers.

# Cursores

- Los cursores son un recurso de programación que permiten recorrer un juego de resultados (result set) previamente definido.
- El juego de resultados definido para un cursor queda residente en la memoria asignada al Sistema Gerenciador de Base de Datos (\*).
- Los cursores no se definen como un objeto de base de datos (no son persistentes), están definidos dentro de una sesión de conexión a la base de datos.
- Los cursores son utilizados principalmente dentro de otros objetos tales como los triggers y procedimientos almacenados.

(\*) La administración de recursos del Sistema Gerenciador de Base de Datos forma parte de otra unidad que se desarrollará posteriormente.

# Cursores

- Las siguientes sentencias son utilizadas para el manejo de cursores.

SENTENCIA	DESCRIPCIÓN	SINTAXIS
Declaración →	Se hace en base una sentencia Select. Deja la declaración en memoria pueden ser declarados de sólo lectura o de actualización.	<code>declare nombre_cursor cursor for select from where order by group by [for read only   update]</code>
Open →	Ejecuta la sentencia Select de la declaración. Deja el juego de resultados en la memoria	<code>open nombre_cursor</code>
Fetch →	Ubica el control sobre el registro a continuación, es factible volcar el registro en variables.	<code>fetch next nombre_cursor into nombre_variable</code>
Close →	Cierra el cursor, saca de memoria el juego de resultados.	<code>close nombre_cursor</code>
Deallocate →	Saca de memoria la declaración del cursor	<code>deallocate cursor nombre_cursor</code>

# Cursores - Declaración.

- La declaración de un cursor se realiza en base a una sentencia select y define el cursor en la memoria, ejemplo:

```
declare csr_productos cursor for  
select producto,  
       nombre  
from   productos
```



Sistema Gerenciador de Bases de Datos.



Base de Datos



Declaración



Espacio de Memoria

# Cursores - Open.

- La apertura de un cursor ejecuta la sentencia select previamente definida en la declaracion y ubica el juego de resultados en memoria.

Open csr\_productos



Sistema Gerenciador de Bases de Datos.



Base de Datos



Juego de Resultados



Espacio de Memoria

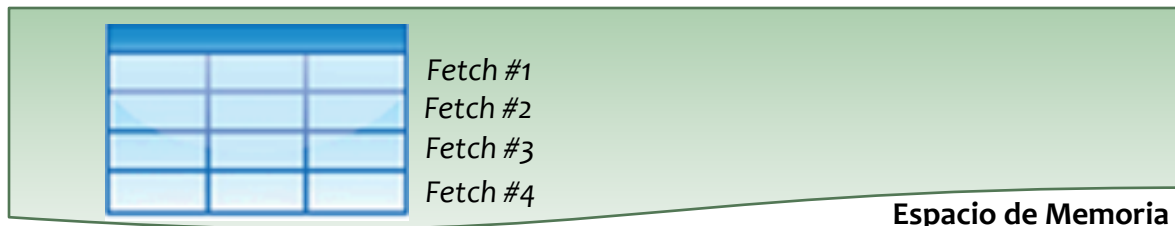
# Cursores - Fetch.

- La sentencia fetch se utiliza para recorrer las filas del juego de resultados, cada SGBD posee una variable de sistema que le permite identificar si pudo llevarse a cabo satisfactoriamente el comando.

fetch csr\_productos



Sistema Gerenciador de Bases de Datos.





# Cursores - Close.

- La sentencia close libera del espacio de memoria el juego de resultados del cursor.

Close csr\_productos



Sistema Gerenciador de Bases de Datos.



Juego de Resultados



Espacio de Memoria



# Cursores - Deallocate.

- La sentencia deallocate libera del espacio de memoria la declaracion del cursor.

Deallocate cursor csr\_productos



Sistema Gerenciador de Bases de Datos.



Declaración



Espacio de Memoria

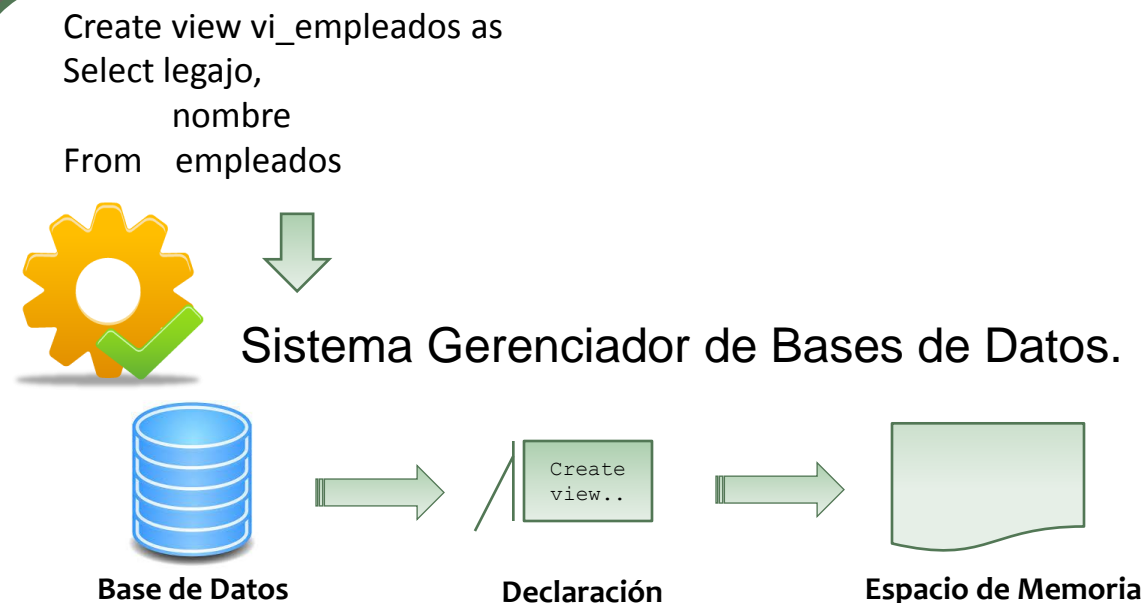
# Cursores - Resumen

---

- Los cursores además de proveer el mecanismo para recorrer un juego de resultados, los mismos pueden ser definidos para ser actualizables.
- En este caso se incluye la sentencia FOR UPDATE en la declaración del cursor.
- Para que un cursor sea actualizable debe estar declarado sobre una sola tabla y debe incluir el campo de la clave primaria.
- El update realizado en el cursor con la sentencia [where current of nombre-cursor] actualiza la tabla base por la cual fue definido el cursor.

# Vistas

- Las vistas son un tipo de objeto de la base de datos y se construyen en base a una sentencia select.



# Vistas

- Al crear una vista, en el objeto se almacena la sentencia con la cual fue creada la misma.
- Las vistas no poseen datos propios, el juego de resultados que se visualiza al proyectar los atributos de una vista corresponden a la sentencia select por la cual fueron definidos (\*).
- Las vistas poseen 2 utilidades principales:
  - Permiten reusar sentencias SELECT complejas en varios programas.
  - Permiten abstraer de datos sensibles a los usuarios que acceden a los datos.

(\*) Las vistas materializadas tienen un manejo distinto y se utilizan para entornos de DataWare/BI ver:  
[http://es.wikipedia.org/wiki/Vista\\_materializada](http://es.wikipedia.org/wiki/Vista_materializada)

# Vistas - Resumen

---

- Las vistas además de ser utilizadas como mecanismos de reusabilidad de código y/o proveer seguridad de acceso a los datos, las mismas pueden ser definidas para ser actualizables.
- En este caso se incluye la sentencia FOR UPDATE en la creación de la vista.
- Para que una vista sea actualizable debe estar creada sobre una sola tabla y debe incluir el campo de la clave primaria.
- El update realizado en la vista, actualiza la tabla base por la cual fue definida la misma.

# Tablas temporales

---

- Las tablas temporales poseen las mismas características que las tablas base o también denominadas tablas de usuario, la única diferencia es que las tablas temporales persisten en la base de datos mientras la sesión de conexión este activa.
- Cuando una tabla temporal es creada dentro de un procedimiento almacenado el tiempo de persistencia se define hasta que el procedimiento finalice la ejecución.
- Las tablas temporales se utilizan para almacenar resultados intermedios y posibilitar la ejecución de sentencias SELECT más específicas y de menor complejidad (evitar resolver en un solo query una consulta compleja).



# Procedimientos Almacenados

---

- Los procedimientos almacenados son programas residentes en la base de datos con el objetivo principal de reutilizar código, dentro de una programación estructurada.
- Los procedimientos poseen parámetros de entrada/salida y en el cuerpo del mismo se pueden escribir las sentencias SQL (sólo DML).
- Se pueden utilizar los procedimientos como mecanismos de seguridad en la manipulación de datos de las tablas importantes del sistema.
- Otra utilidad de los procedimientos almacenados es la de implementar la lógica del negocio.



# Procedimientos Almacenados

---

- Ejemplo: Calcular Saldo Alumno

**Programa 1**

*Lógica Calculo Saldo*

**Programa 2**

*Lógica Calculo Saldo*

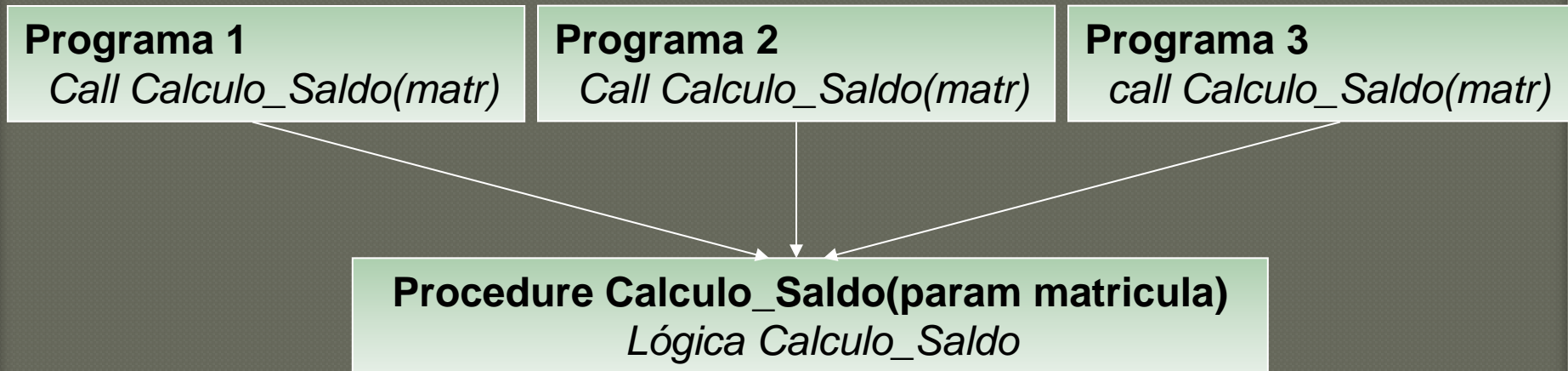
**Programa 3**

*Lógica Calculo Saldo*

- Si cambia la lógica de calculo de saldo, en cuantos programas se debería realizar los cambios.?

# Procedimientos Almacenados

- Ejemplo: Calcular Saldo Alumno



- Si cambia la lógica de calculo de saldo, en cuantos programas se debería realizar los cambios.?

# Funciones

- Las funciones son programas residentes en la base de datos, pueden venir incorporadas en los SGBDs, funciones Built-in o bien ser creadas por el usuario.
- Las funciones solo poseen parámetros de entrada, en el cuerpo del mismo se pueden programar las sentencias SQL (solo DML) y solo pueden devolver un valor de retorno (\*).
- La principal utilidad de las funciones, es que pueden ser incorporadas en las sentencias SELECT, ya sea en la lista de columnas como en el where, en remplazo de los sub-selects.

(\*) En SQL Server se pueden definir funciones que devuelvan un juego de resultados (Table Evaluated Functions) ver: <http://technet.microsoft.com/es-ar/library/ms191165%28v=sql.105%29.aspx>.

# Funciones

---

- Las funciones built-in están incorporadas en todos los SGBDs, los nombres suelen variar entre cada uno de ellos.
- Se dividen en:
  - Funciones Matemáticas
  - Funciones de Cadena
  - Funciones de Fecha
  - Funciones de Conversión.
- Además se encuentran las funciones de conjunto, las cuales están incorporadas en el estándar (ANSI)  
Count(), Min(), Max(), Avg(), Sum()