

Laboratorio IV – SQL

Lenguaje de Manipulación de Datos
(DML – *Data Manipulation Language*)

Ing. Gregorio Nicolás Tkachuk



Objetivos

- ▶ Que el estudiante logre:
 - Conocer y utilizar las sentencias básicas del Lenguaje de Manipulación de Datos (DML)
 - Habilidad para actualizar Bases de Datos.

DML: Contenidos

- ▶ Insert
- ▶ Delete
- ▶ Update
- ▶ Alias de columna
- ▶ Alias de tablas

DML

- ▶ El **Lenguaje de Manipulación de Datos** (*Data Manipulation Language*, o *DML* en inglés) es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios llevar a cabo las tareas de consulta o manipulación de los datos, organizados por el modelo de datos adecuado.
- ▶ Sentencias: Insert; Delete; Update; Select

SQL – INSERT – Sentencia para agregar una fila a una Tabla (1)

```
INSERT INTO <nombre_tabla>  
[( < atributo1 >, . . . , <atributoN> )]  
VALUES (<expresion1 >, ..,<expresionN>);
```

Ejemplo:

- ▶ **INSERT INTO alumno
(aluDoc, aluNom, aluDir)
VALUES (30111222, 'Lopez Martina', 'Av. Roca 77');**
- ▶ **INSERT INTO carrera VALUES (10, "PUI");**

SQL – INSERT –Sentencia para agregar varias filas a una Tabla (2)

**INSERT INTO <nombre_tabla>
[(< atributo1 >, . . . , <atributoN>)]
(sentencia SELECT);**

- ▶ Los atributos de ambas listas de atributos deben coincidir en el mismo dominio. La sentencia "SELECT" especifica una consulta cuyo resultado (filas) se insertará en la tabla especificada.

SQL – INSERT –Sentencia para agregar varias filas a una Tabla (3)

**INSERT INTO <nombre_tabla>
[(< atributo1 >, . . . , <atributoN>)]
(sentencia SELECT);**

Ejemplo: Agregar en la tabla alumno los ingresantes de la comisión 3.

▶ **INSERT INTO alumno
(aluDoc, aluNom, aluDir)
(SELECT doc, nom, dir FROM ingresante WHERE
comision = 3);**

SQL – DELETE –Sentencia para borrar registros de una Tabla

**DELETE FROM <nombre_tabla>
[WHERE <condicion>];**

Si no se especifica [WHERE <condición>], se eliminan todas las filas de la tabla.

Ejemplo:

DELETE FROM libro;

DELETE FROM alumno

WHERE aluDoc = 30111222;

SQL – UPDATE –Sentencia para actualizar datos de una Tabla (1)

```
UPDATE <nombre_tabla>  
SET <atributo1> = <valor1>,  
    <atributo2> = <valor2>,  
    ...  
[ WHERE <condicion> ];
```

- ▶ Si no se especifica [WHERE <condición>], la modificación afectará a todas las filas de la tabla.
- ▶ El valor que se asigne a un atributo puede ser una constante, o el resultado de una subconsulta (que deberá ir entre paréntesis).

SQL – UPDATE –Sentencia para actualizar datos de una Tabla (2)

```
UPDATE <nombre_tabla>  
SET <atributo1> = <valor1>, ...  
[ WHERE <condicion> ];
```

Ejemplo: aumentar un 10% el precio de los artículos del rubro 15.

```
UPDATE articulo  
SET artPrecio = artPrecio + (artPrecio * 10/100 )  
WHERE artRubro = 15;
```

SQL – NULL

NULL valor por defecto que toma una columna cuando no se ha asignado explícitamente un valor. Significa “no hay dato”

Toda columna puede tomar valor NULL, siempre que ésta no posea una restricción que lo impida (por ejem. NOT NULL)

El valor NULL de una columna de tipo numérica o carácter no es igual a 0 (cero) o “.”.

SQL – ALIAS de Columna y de Tabla

SQL permite que se definan alias para las columnas y las tablas; pudiendo los mismos además ser utilizados en las otras cláusulas. La palabra reservada es “**ALIAS**” o simplemente “**AS**” en su formato abreviado.

Alias de columna: se utiliza generalmente para dar mayor legibilidad al resultado de la consulta. Se puede utilizar mas de una palabra como alias en este caso deben ir entre comillas.

Alias de Tabla: se utiliza generalmente para simplificar o abreviar el nombre de una tabla en la escritura de una consulta. No se puede utilizar mas de una palabra.

SQL – ALIAS de Columna y de Tabla (cont.)

Se utilizan:

En el FROM para tablas:

FROM tabla [[AS] Alias] [, ...]

En el SELECT para columnas

SELECT expresión [[AS] Alias] [, ...]

```
SELECT p.Nombre, (p.Precio * p.Cantidad) AS Importe  
FROM Producto AS p  
WHERE p.cantidad > 100;
```

Preguntas ?