

# Laboratorio IV – SQL

Lenguaje de Definición de Datos (DDL)  
CREATE TABLE

Ing. Gregorio Nicolás Tkachuk



# Objetivos

- ▶ Que el estudiante logre:
  - Conocer y utilizar el comando CREATE TABLE.
  - Habilidad para utilizar las distintas variantes y restricciones del comando CREATE TABLE.

# DDL: Contenidos

- ▶ Create Table
- ▶ Tipos de datos de columna
- ▶ Restricción de integridad
- ▶ Valor por defecto (DEFAULT)
- ▶ Restricción de Columna
- ▶ Restricción de clave primaria
- ▶ Restricción de clave foránea

# SQL – CREATE TABLE

```
CREATE TABLE <nombre_tabla>(  
  <nombre_columna-1> <tipo>  
    [<valor por defecto>] [<restricciones de columna>],  
  <nombre_columna-2> <tipo>  
    [<valor por defecto>] [<restricciones de columna>],  
  ...  
  <nombre_columna-n> <tipo>  
    [<valor por defecto>] [<restricciones de columna>],  
  [<restricciones de tabla>],  
  ....  
  [<restricciones de tabla>]  
);
```

# SQL – CREATE TABLE

**<nombre\_columna-1>**: Nombre asignado a una columna o campo del registro de la tabla.

**< tipo>** : Tipo de dato con el cual se declara la columna (Ejem. INT, CHAR).

**[<valor por defecto>]** : Clausula DEFAULT, permite asignar un valor por defecto a la columna cuando se inserta un registro.

**[<restricciones de columna>]**, : Restricción declarada y aplicable a una columna en particular. (Ejem. NOT NULL, UNIQUE, PRIMARY KEY)

**[<restricciones de tabla>]**, :Restricción declarada de manera separada que puede aplicarse a una o más columnas. (Ejem. UNIQUE(), PRIMARY KEY () )

# SQL – CREATE TABLE (Cont.)

< **tipo** > : Tipo de dato con el cual se declara la columna Pueden ser:

SMALLINT[(*length*)]

INT[(*length*)]

BIGINT[(*length*)]

REAL[(*length*,*decimals*)]

DOUBLE[(*length*,*decimals*)]

FLOAT[(*length*,*decimals*)]

DECIMAL(*length*,*decimals*)

NUMERIC(*length*,*decimals*)

DATE

TIME

TIMESTAMP

DATETIME

CHAR(*length*)

VARCHAR(*length*)

BLOB

TEXT

# SQL – DEFAULT valor por defecto

```
CREATE TABLE <nombre_tabla>(  
  <nombre_columna-1> < tipo>  
    [<valor por defecto>]  
    DEFAULT <valor expresión>
```

Permite asignar a la columna un valor por defecto cuando se realiza un INSERT y no se especifica valor.

<*valor expresión*> es una expresión simple de literales, funciones y valores.

La expresión debe ser evaluada a un tipo de dato que coincida o que pueda ser implícitamente convertida al tipo de dato de la columna.

# SQL – CREATE TABLE Ejemplo

```
CREATE TABLE Productor (  
  proId INT DEFAULT 0,  
  proNombre VarChar(40) DEFAULT 'SECRETO',  
  proSexo Char(1) DEFAULT 'M' );
```

Se puede agregar un registro sin indicar los valores.  
`INSERT INTO Productor VALUE();`

MySQL: no admite para una columna de tipo fecha una función como `NOW()` o `CURRENT_DATE`.

Solo `CURRENT_TIMESTAMP` como `DEFAULT` para las columnas `TIMESTAMP` y `DATETIME`



# SQL – Restricciones de columnas

```
CREATE TABLE <nombre_tabla>(
  <nombre_columna-1> < tipo>
  [<restricciones de columna>] )
```

**NOT NULL:** Restricción de **obligatoriedad**, la columna no puede tomar valores nulos.

**UNIQUE:** Restricción de **unicidad**, los valores para la columna no se pueden repetir.

**PRIMARY KEY:** Restricción de clave primaria, permite declarar uno o varios atributos como clave primaria de una relación, o sea que se identifican unívocamente en toda la tabla.

# SQL – Restricciones de columnas (Cont.)

## FOREIGN KEY (<*columnas*>) REFERENCES

<*referenced table*>[(<*referenced columns*>)]

La restricción de clave foránea, también denominada integridad referencial, se utiliza para crear una relación entre una tabla y otra.

**CHECK (<*condición*>):** Restricción de verificación, es de tipo general, permite agregar una condición que deben cumplir los valores de una columna.

# SQL – Restricciones de columnas (Cont.)

**CONSTRAINT <nombre> <restricción>:** Permite asignarle un nombre a una restricción, cuando se viola la restricción el motor de base de datos muestra el nombre asignado, haciendo mas claro el mensaje de error.

# SQL – Restricción de columna NOT NULL, Ejemplo

**NOT NULL:** Restricción de obligatoriedad, la columna no puede tomar valores nulos.

```
CREATE TABLE cliente (  
  cliId CHAR(5) NOT NULL,  
  cliNombre VARCHAR(30) DEFAULT 'SECRETO' NOT NULL,  
  cliDireccion VARCHAR(20) DEFAULT 'Santiago',  
  cliTelefono CHAR(15) NULL  
);
```

# SQL – Restricción UNIQUE

**UNIQUE:** Restricción de unicidad, los valores para la columna no se pueden repetir.

Permite definir claves alternativas.

Se aplica solo a los valores no nulos.

**Ejemplo como restricción a nivel de columna:**

```
CREATE TABLE zona (  
  zonId CHAR(5) NOT NULL,  
  zonDescripcion CHAR(30) NOT NULL UNIQUE );
```

**Ejemplo como restricción a nivel de tabla:**

```
CREATE TABLE zona (  
  zonId CHAR(5) NOT NULL,  
  zonDescripcion CHAR(30) NOT NULL,  
  UNIQUE(zonId, zonDescripcion) );
```

# SQL – Restricción PRIMARY KEY

**PRIMARY KEY:** Restricción de clave primaria, permite declarar uno o varios atributos como clave primaria de una relación, o sea que se identifican unívocamente en toda la tabla.

Los valores de las columnas de la clave primaria no pueden ser nulos.

No se permiten dos tuplas con los mismos valores para todos los atributos de la clave primaria.

En el modelo relacional es muy recomendable que todas las tablas tengan clave primaria.

# SQL – Restricción PRIMARY KEY, Ejemplo

Ejemplo como restricción a nivel de columna:

```
CREATE TABLE zona (  
  zonId CHAR(5) PRIMARY KEY NOT NULL,  
  zonDescripcion CHAR(30) NOT NULL );
```

Ejemplo como restricción a nivel de tabla:

```
CREATE TABLE comprobante (  
  comTipo CHAR(1) NOT NULL DEFAULT 'R',  
  comNumero INT NOT NULL,  
  comDescripcion VARCHAR(30),  
  comFecha DATE,  
  PRIMARY KEY (comTipo , comNumero )  
);
```

# SQL – Restricciones de Clave Foránea

## FOREIGN KEY (<*columnas*>) REFERENCES

<*referenced table*>[(<*referenced columns*>)]

La restricción de clave foránea, también denominada **integridad referencial**, se utiliza para crear una relación entre una tabla y otra.

La integridad referencial nos indica que los valores de la clave foránea en la relación hija, deben corresponderse con los valores de la clave primaria en la relación padre o bien ser nulos si el atributo lo permite.



# SQL – Restricciones de Clave Foranea, Ejemplo

```
CREATE TABLE provincia (  
codProv int NOT NULL,  
descripcion varchar(30) NOT NULL,  
PRIMARY KEY(codProv) );
```

```
CREATE TABLE padron (  
documento int NOT NULL,  
nombre varchar(30) NOT NULL,  
dirección varchar(40) NOT NULL,  
codProv int,  
PRIMARY KEY(documento),  
FOREIGN KEY(codProv) REFERENCES provincia(codProv) );
```

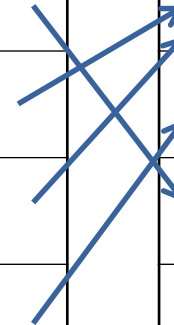
# SQL – Ejemplo de relación 1 a N

**Padron**

Documento (PK)	Nombre	Direccion	CodProv (FK)
20111222	PEREZ, MARTIN	Las Moras 223	3
25000111	PAZ, MONICA	Bs As 400	1
30999444	GARCIA, DIEGO	Chaco 1300	1
35444000	LOPEZ, MAGALI	Jujuy 770	2

**Provincia**

CodProv (PK)	Descripcion
1	Santiago del Estero
2	Chaco
3	Salta
4	Tucuman



# SQL – Restricción de verificación CHECK

**CHECK (<condición>):** Restricción de **verificación**, es de tipo general, permite agregar una condición que deben cumplir los valores de una columna.

**(<condición>):** es una expresión de tipo booleana que se evalúa previo a cualquier cambio que se produzca en una tupla.

Las restricciones de verificación siempre llevan implícitas un rechazo en caso de que no se cumpla la condición especificada y que también se comprueban ante una inserción, borrado o modificación.

# SQL – Restricción de verificación CHECK, Ejemplo

```
CREATE TABLE producto (  
    codigo int NOT NULL,  
    nombre varchar(50) NOT NULL,  
    precio decimal(10,2) CHECK (precio > 0),  
    PRIMARY KEY(codigo) );
```

```
tipoDoc CHAR(3) CHECK (tipoDoc IN  
    ('DNI','LC','CI','DUI','PAS') )
```

Nota: No se aplica en MySql

# SQL – Nombre a Restricciones

**CONSTRAINT <nombre> <restricción>:** Permite asignarle un nombre a una restricción, cuando se viola la restricción el motor de base de datos muestra el nombre asignado, haciendo mas claro el mensaje de error.

```
CREATE TABLE padron ( documento int NOT NULL, nombre  
    varchar(30) NOT NULL, dirección varchar(40) NOT NULL,  
    codProv int, PRIMARY KEY(documento),  
    CONSTRAINT errorCodProvFK FOREIGN KEY(codProv)  
    REFERENCES provincia(codProv) );
```

# Preguntas ?