

CAPITULO III – PARTE I

IMPLEMENTACIÓN DE REGLAS DE INTEGRIDAD DE LAS BASE DE DATOS / MODELO DE BD RELACIONAL

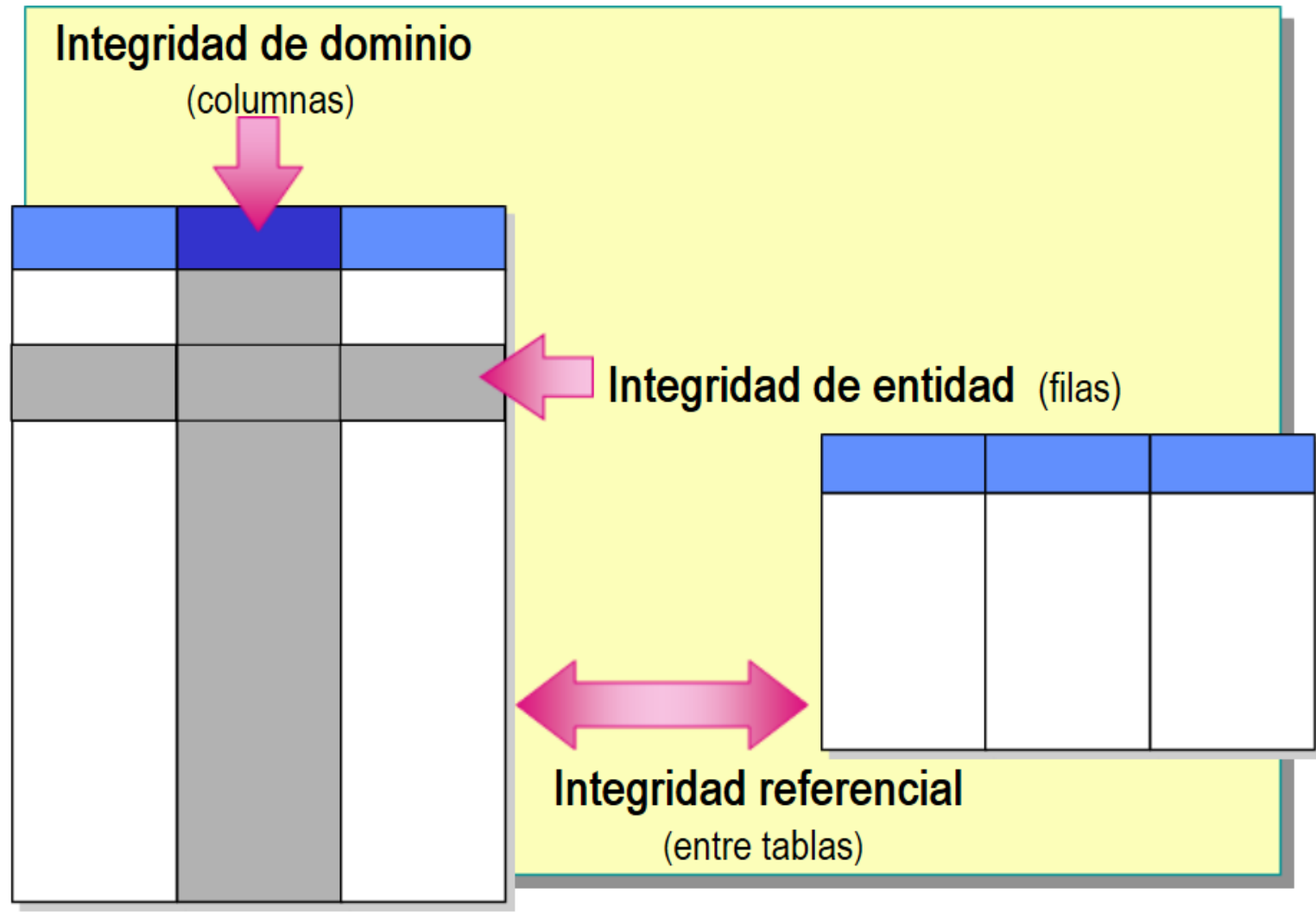
SEGUNDO SEMESTRE 2023

CONCEPTOS

- Integridad
- Datos requeridos (null, not null)

Puesto que una base de datos es generalmente un modelo que refleja una situación del mundo real, ciertos datos pueden inevitablemente faltar, ser desconocidos o no ser aplicables. El lenguaje estándar de base de datos soporta explícitamente los datos que faltan, son desconocidos o son inaplicables, a través del concepto de valor nulo (que no debe confundirse con el valor cero).

Tipos de integridad de datos



Determinación del tipo de restricción que se va a utilizar


Tipo de integridad	Tipo de restricción
Dominio	DEFAULT
	CHECK
	REFERENTIAL
Entidad	PRIMARY KEY
	UNIQUE
Referencial	FOREIGN KEY
	CHECK

EXIGIR LA INTEGRIDAD DE LOS DATOS

Para poder ejecutar esto SQL Server nos permite definir datos y nos entrega herramientas para poder exigir la Integridad Referencial, como:

- PRIMARY KEY
 - FOREIGN KEY
 - UNIQUE
 - CHECK
 - DEFAULT
- 

RESTRICCIONES PRIMARY KEY

- ✘ Una tabla suele tener una columna o una combinación de columnas cuyos valores identifican de forma única cada fila de la tabla.
 - ✘ Estas columnas se denominan claves principales de la tabla y exigen la integridad de entidad de la tabla.
 - ✘ Puede crear una clave principal mediante la definición de una restricción PRIMARY KEY cuando cree o modifique una tabla.
 - ✘ Una tabla sólo puede tener una restricción PRIMARY KEY y ninguna columna a la que se aplique una restricción PRIMARY KEY puede aceptar valores NULL.
- 

RESTRICCIONES PRIMARY KEY

Si se define una restricción PRIMARY KEY para más de una columna, puede haber valores duplicados dentro de la misma columna, pero cada combinación de valores de todas las columnas de la definición de la restricción PRIMARY KEY debe ser única.

Para crear tablas con llaves compuestas :

```
CREATE TABLE student
(
    id INT not null,
    codest varchar (15) not null,
    constraint primariatotal PRIMARY KEY(ID, codest),
    name VARCHAR(50) NOT NULL,
    gender VARCHAR(50) NOT NULL,
    birthday datetime NOT NULL,
    total_score INT NOT NULL,
    city VARCHAR(50) NOT NULL
)
```

CREAR RESTRICCIÓN PRIMARY KEY

Para generar una llave primaria desde la creación de una tabla:

```
1 CREATE TABLE nombreEsquema.nombreTabla
2 (
3     nombreColumna1 INT NOT NULL,
4     nombreColumna2 VARCHAR(100) NOT NULL,
5     nombreColumna3 NVARCHAR(200) NOT NULL,
6     CONSTRAINT PK_nombreRestriccion PRIMARY KEY( nombreColumna1 )
7 );
```

Ejemplo:

```
CREATE TABLE PELICULA(
ID_PELICULA VARCHAR(10) NOT NULL,
TITULO VARCHAR (20) NOT NULL,
PRODUCTORA VARCHAR (15) NOT NULL,
FECHA_ESTRENO DATE NOT NULL,
CONSTRAINT PK_IDPELICULA PRIMARY KEY
(ID_PELICULA)
```


CREAR RESTRICCIÓN PRIMARY KEY

Modificando una tabla:

```
1 ALTER TABLE nombreEsquema.nombreTabla  
2 ADD CONSTRAINT PK_nombreRestriccion PRIMARY KEY( nombreColumna1 );
```

Ejemplo:

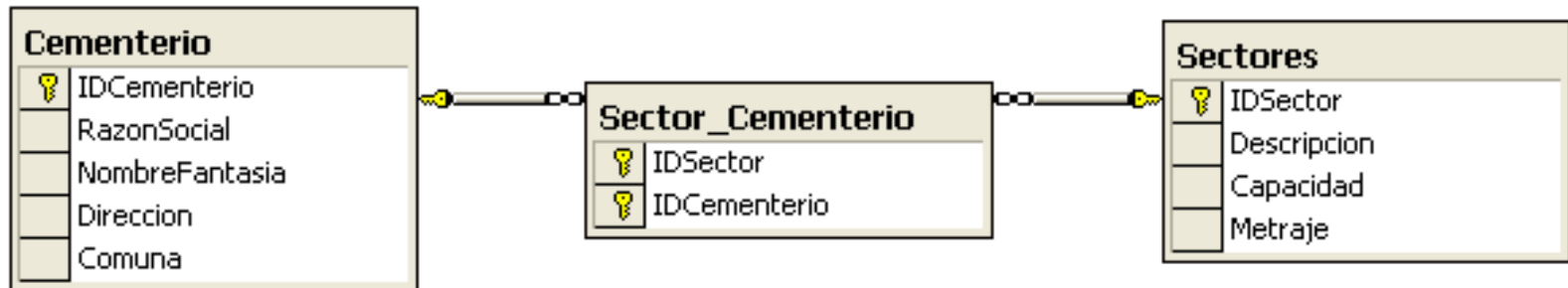
Nombre de la tabla

ALTER TABLE EJEMPLAR ADD CONSTRAINT
PK_NUMERO PRIMARY KEY (NUMERO)

Nombre de la restricción

RESTRICCIONES PRIMARY KEY

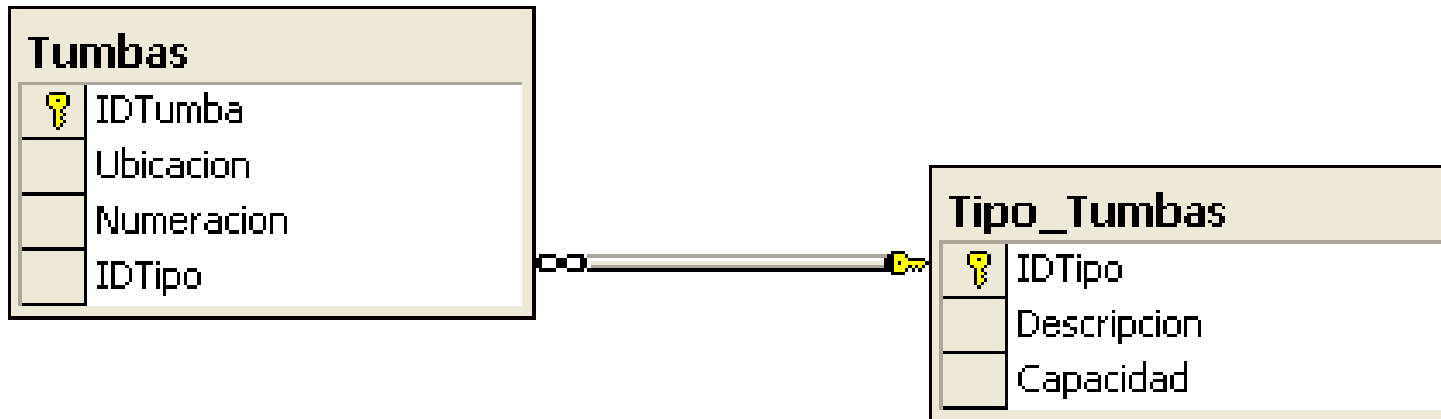
Ejemplo: Esta es una relación de ?



RESTRICCIONES FOREIGN KEY

- ✘ Una clave foránea(FK) es una columna o combinación de columnas que se utiliza para establecer y exigir un vínculo entre los datos de dos tablas.
- ✘ Puede crear una clave externa mediante la definición de una restricción FOREIGN KEY **cuando cree o modifique una tabla o al asignar la relación en el diagrama.**
- ✘ En una referencia de clave externa, se crea un vínculo entre dos tablas cuando las columnas de una de ellas hacen referencia a las columnas de la otra que contienen el valor de clave principal.
- ✘ Esta columna se convierte en una clave externa para la segunda tabla

RESTRICCIONES FOREIGN KEY



CREAR RESTRICCIÓN FOREIGN KEY

Para generar una llave foránea desde la creación de la tabla con constraint

```
CONSTRAINT fk_constraint_name  
FOREIGN KEY (column_1, column2,...)  
REFERENCES parent_table_name(column1,column2,..)
```

Ejemplo:

```
CREATE TABLE ALQUILA (  
DNI VARCHAR(15) NOT NULL,  
ID_PELICULA VARCHAR (10) NOT NULL,  
NUMERO VARCHAR (10) NOT NULL,  
FECHA_ALQUILER DATE NOT NULL,  
FECHA_DEVOLUCION DATE,  
CONSTRAINT FK_DNI FOREIGN KEY (DNI) REFERENCES SOCIO (DNI),  
CONSTRAINT FK_IDPELICULA_AL FOREIGN KEY (ID_PELICULA) REFERENCES PELICULA (ID_PELICULA),  
);  
GO
```

CREAR RESTRICCIÓN FOREIGN KEY

Luego de crear la tabla

Ejemplo:

Nombre de la tabla

```
ALTER TABLE ALQUILA ADD CONSTRAINT FK_NUMERO FOREIGN KEY (NUMERO)  
REFERENCES EJEMPLAR (NUMERO)
```

Nombre de la restricción

RESTRICCIONES UNIQUE

- ✘ Puede utilizar restricciones UNIQUE para garantizar que no se escriben valores duplicados en columnas específicas que no forman parte de una clave principal.
- ✘ Tanto la restricción UNIQUE como la restricción PRIMARY KEY exigen la unicidad
- ✘ A diferencia de PK, si puede aceptar NULL, pero solo una fila puede contenerlo ya que como su nombre lo indica, es de tipo UNIQUE o único.
- ✘ En una tabla se pueden definir varias restricciones UNIQUE, pero sólo una restricción PRIMARY KEY.

```
CONSTRAINT UQ_CORREO UNIQUE (CORREO)
```

CREAR RESTRICCIÓN UNIQUE

```
1 CREATE TABLE nombreEsquema.nombreTabla
2 (
3     nombreColumna1 INT      NULL,
4     nombreColumna2 VARCHAR(100) NOT NULL,
5     nombreColumna3 NVARCHAR(200) NOT NULL,
6     CONSTRAINT UQ_nombreRestriccion UNIQUE( nombreColumna1 ),
7     CONSTRAINT UQ_nombreRestriccion2 UNIQUE( nombreColumna2 ),
8     CONSTRAINT UQ_nombreRestriccion3 UNIQUE( nombreColumna1,nombreColumna2 )
9 );
```

```
CREATE TABLE ACTORES(
ID_ACTOR VARCHAR(5) CONSTRAINT PK_ID_act PRIMARY
KEY NOT NULL,
NOMBRE VARCHAR(30) NOT NULL,
SEXO CHAR(2) NOT NULL
CONSTRAINT U_TELEFONO UNIQUE(TELEFONO));
```


BORRAR RESTRICCIÓN UNIQUE

✖ Para modificar una restricción **UNIQUE**, deberá eliminar la restricción **UNIQUE** existente y, a continuación, volver a crearla con la nueva definición.

Borrar una restricción **UNIQUE** creada.

Para borrar una restricción **UNIQUE** se debe utilizar la instrucción **ALTER TABLE** con su opción **DROP**.

Nomenclatura

ALTER TABLE nombre_tabla

DROP CONSTRAINT UQ_nombre_restriccion

RESTRICCIONES CHECK

Las restricciones CHECK exigen la integridad del dominio mediante la limitación de los valores que puede aceptar una columna.

Este constraint actúa cuando el usuario emplea una instrucción INSERT o UPDATE.

```
CREATE TABLE nombreEsquema.nombreTabla
(
  nombreColumna1 INT      NULL,
  nombreColumna2 VARCHAR(100) NOT NULL,
  nombreColumna3 NVARCHAR(200) NOT NULL,
  --VALORES POSITIVOS
  CONSTRAINT CH_nombreRestriccion CHECK (nombreColumna1>=0),
  -- SOLO VALORES IGUALES A 10 20 30 40
  CONSTRAINT CH_nombreRestriccion2 CHECK (nombreColumna1 IN
(10,20,30,40)),
  --VALORES CONTENIDOS EN UN RANGO
  CONSTRAINT CH_nombreRestriccion3 CHECK (nombreColumna1>=1 AND
nombreColumna1 <=30)
);
```

CREAR RESTRICCIÓN CHECK

Para generar un campo check desde la creación de la tabla con constraint

Ejemplo:

```
CREATE TABLE PELICULA (  
  ID_PELI VARCHAR(5) NOT NULL,  
  ID_DIRECTOR VARCHAR(5) NOT NULL,  
  PRODUCTORA VARCHAR(30) NOT NULL,  
  NACIONALIDAD VARCHAR(30) NOT NULL,  
  FECHA_DE_ESTRENO DATE NOT NULL,  
  ESTADO VARCHAR (25) NOT NULL,  
  CONSTRAINT PK_IDPELICULA PRIMARY KEY (ID_PELI),  
  CONSTRAINT CH_ESTADO CHECK (ESTADO = 'DISPONIBLE' OR ESTADO =  
  'NO DISPONIBLE'))
```

Modificando la tabla:

```
alter table Actores  
add constraint Actores_check  
check(género in('F','M'))
```

RESTRICCIONES CHECK

- ✘ Son similares a las restricciones FOREIGN KEY porque controlan los valores que se insertan en una columna.
- ✘ La diferencia está en que las restricciones FOREIGN KEY obtienen la lista de valores válidos de otra tabla, mientras que las restricciones CHECK determinan los valores válidos a partir de una expresión lógica que no se basa en datos de otra columna.

Otro ej. Alterando la tabla:

```
ALTER TABLE dbo.Employees  
ADD  
CONSTRAINT CK_birthdate  
CHECK (BirthDate > '01-01-1900' AND BirthDate <  
getdate())
```

RESTRICCIONES DEFAULT

Estos constraints trabajan al momento de utilizar la función INSERT y asignan un valor automáticamente a la columna que no se le asignó.

Se puede decir que no es una restricción, ya que solo se ingresa un valor en caso de que ninguno otro sea especificado.

Cada columna de un registro debe contener un valor, aunque sea un valor NULL.

```
CREATE TABLE nombreTabla
(
  nombreColumna1 INT      NULL CONSTRAINT DF_nombreRestriccion DEFAULT(0),
  nombreColumna2 VARCHAR(100) NOT NULL,
  nombreColumna3 NVARCHAR(200) NOT NULL,
);
```

EJEMPLOS CAMPOS DEFAULT

Para generar un campo default desde la creación de la tabla con constraint

```
FECHA_ALQUILER DATE NULL CONSTRAINT  
DF_FECHA DEFAULT (GETDATE())
```

Modificando la tabla:

```
alter table Director  
add salario int not null constraint  
Default_Director default(0)
```