

Modelo Entidad - Relación

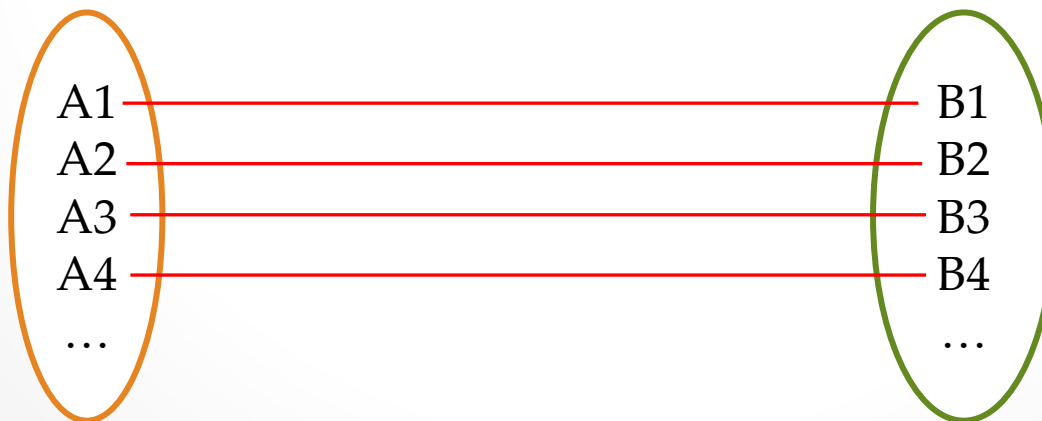
Emerson E. Garay Gómez

www.slideshare.net/emergar

www.youtube.com/user/emergaray

Relación por Cardinalidad

- **Relación de uno a uno (1 : 1)**
- Cuando un registro de una tabla sólo puede estar relacionado con un único registro de la otra tabla y viceversa.
- Este tipo de relación es poco común.

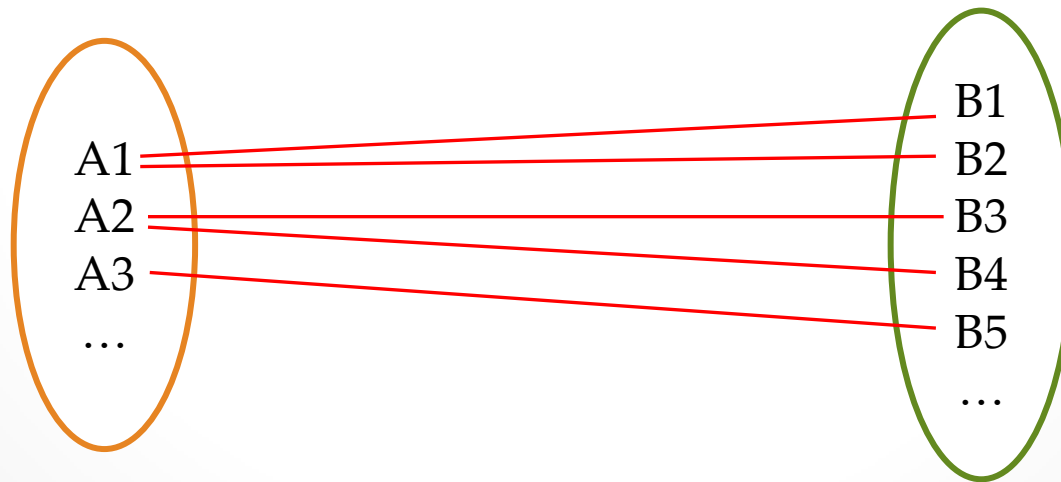


Relación por Cardinalidad

- **Relación de uno a muchos (1 : n)**
- Cuando un registro de una tabla (**tabla secundaria**) sólo puede estar relacionado con un único registro de la otra tabla (**tabla principal**) y un registro de la tabla principal puede tener más de un registro relacionado en la tabla secundaria.

Relación por Cardinalidad

- **Relación de uno a muchos (1 : n)**
- En este caso la **llave foránea** (que es una copia de la llave principal del lado 1) se ubica en la tabla secundaria, es decir, del lado **n**.

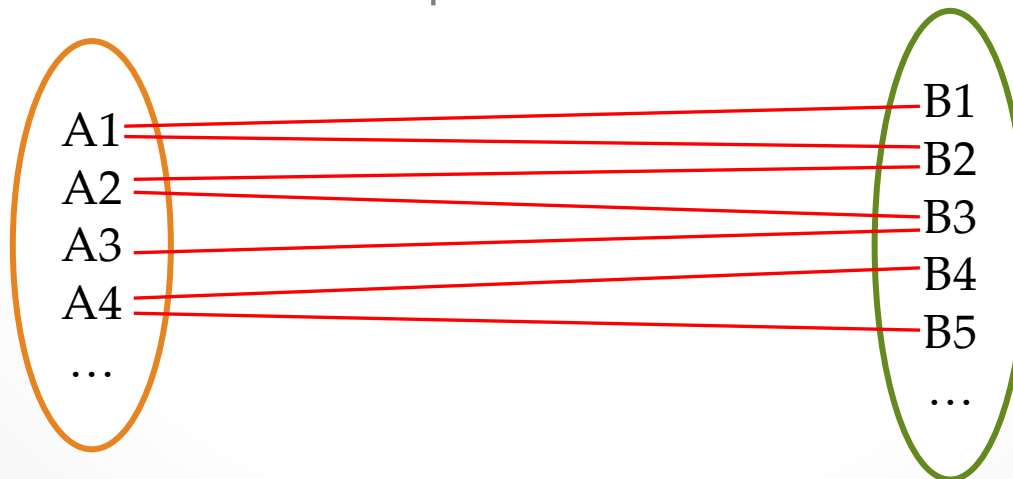


Relación por Cardinalidad

- **Relación Muchos a Muchos ($n : m$)**
- Cuando un registro de una tabla puede estar relacionado con más de un registro de la otra tabla y viceversa. En este caso las dos tablas no pueden estar relacionadas directamente, se tiene que añadir una tabla entre las dos (Tabla débil o de vinculación) que incluya los pares de valores relacionados entre sí.

Relación por Cardinalidad

- **Relación Muchos a Muchos ($n : m$)**
- El nombre de **tabla débil** proviene de que con sus atributos propios no se puede encontrar la clave, por estar asociada a otra entidad. La llave de esta tabla se conforma por la unión de los campos claves de las tablas que relaciona.



Matriculas Estudiantiles

- En la universidad de la región **UniSabanera** se lleva actualmente el proceso de **matriculas** estudiantiles de forma manual, que son organizadas por las secretarias en planillas cada semestre, sin embargo, esta forma conlleva mucho tiempo debido a la cantidad de **estudiantes** (antiguos y nuevos) que se matriculan en cada **programa** cada semestre en la Universidad, por tal, con el objetivo de agilizar el proceso de las matriculas, se ha decidido que se cree una aplicación con bases de datos.

Situación

- En la universidad los datos que se llevan para hacer el proceso de la matrícula son: consecutivo, año, nivel(semestre), periodo del año, identificación, nombre, apellidos, dirección y teléfonos del estudiante, programa y fecha de la matrícula.



- Para hacer el proceso de la matrícula de forma sistematizada, teniendo en cuenta la situación descrita. Las entidades que se necesitan dentro de la base de datos para llevar esa información serán:

Análisis de la situación

- Tabla de programas(**tblProgramas**), para almacenar la información básica de cada uno de los programas que ofrece la universidad, como son: **Código, Nombre, Título** que otorga, **Numero de Semestres**, **Numero de Créditos** académicos, **Registro del MEN**.



Definición de Tablas

- Tabla de estudiantes(**tblEstudiantes**), para manejar la información básica de cada estudiante que se ha matriculado(antiguo) o se va a matricular(nuevo), como son: **Identificación, Nombre, Apellidos, Dirección, Teléfonos.**

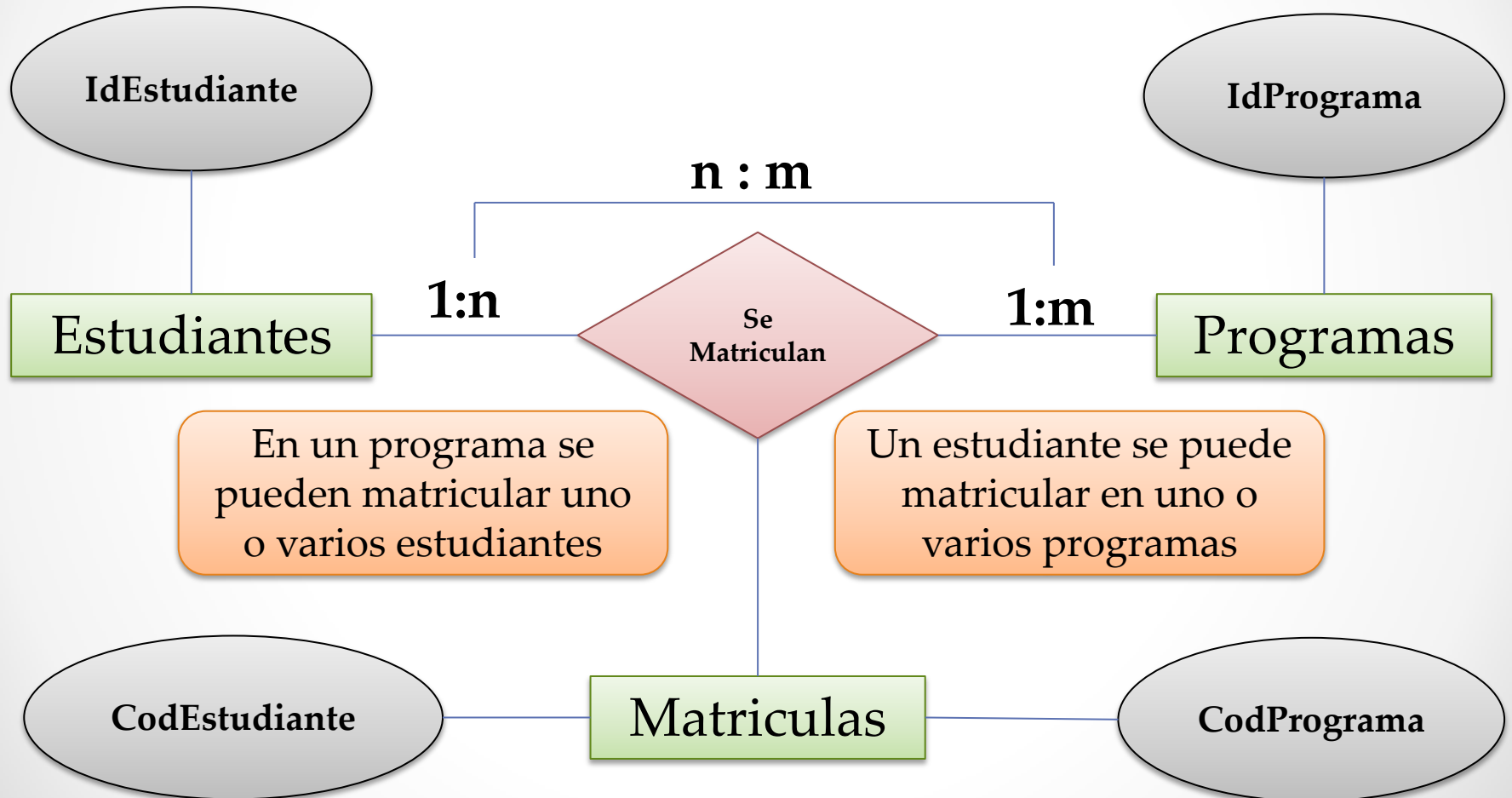


Definición de Tablas

- Tabla de matriculas(**tblMatriculas**), para almacenar la información de cada registro de matricula, como son: **Código** o consecutivo, **Año**, **Periodo**, **Nivel**(semestre), **fecha**, **programa** que estudia y **estudiante** que se matricula.

[illegible]

Diagrama Entidad - Relación



Análisis del Diagrama Entidad - Relación

- Del anterior diagrama la nueva tabla generada (**Matriculas**) por definición se debe componer su clave principal con las claves primarias de las dos tablas que se relacionan y le dan origen.
- **CodEstudiante y CodPrograma**, sin embargo no es suficiente para identificar las diferentes matriculas que el estudiante hará cada semestre, porque el código del estudiante y del programa que estudia se va a repetir semestre a semestre.

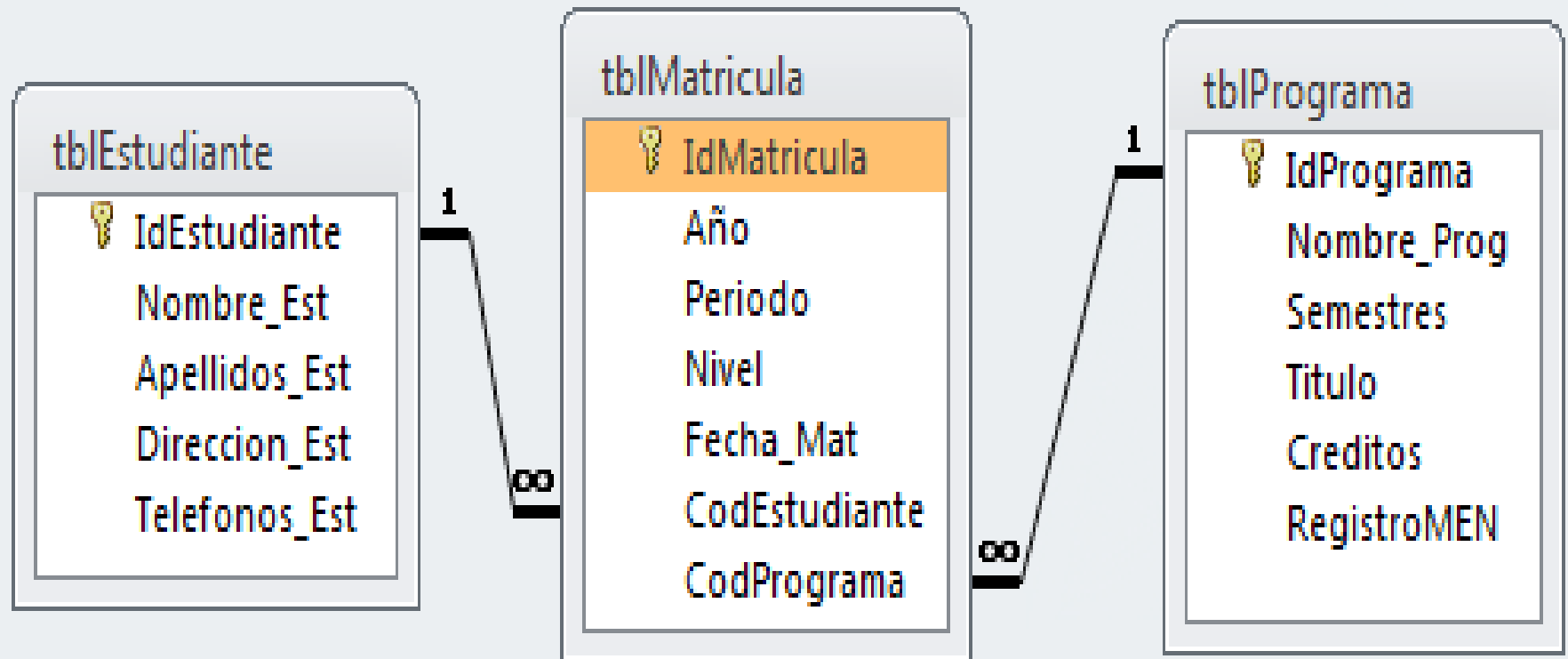
Análisis del Diagrama Entidad - Relación

- Por lo cual, se tienen dos opciones:
- 1. Adicionar otro campo a **CodEstudiante y CodPrograma** para identificar inequívocamente cada registro de matricula de cada estudiante, las claves candidatas son dos combinaciones:
 - **CodEstudiante, CodPrograma, Año y Periodo**
 - **CodEstudiante, CodPrograma y Fecha**
- 2. Colocar otro campo como llave principal, que para este caso puede ser el **numero de la matricula**, ya que es un consecutivo, por lo tanto identifica inequívocamente a cada registro.

Análisis del Diagrama Entidad - Relación

- Resolvemos tomar la opción 2 debió a que es mas optima, por tener un solo campo y no **tres** o **cuatros** como en las otras dos claves candidatas a llave principal.
- Eso si, para saber si un estudiante esta matriculado, se debe busca que coincidan los campos: **CodEstudiante, CodPrograma, Año y Periodo**, en caso de no tener el numero de la matricula

Modelo Entidad - Relación



Estructura

Tabla Programas

Identificador del Campo	Tipo Dato / Longitud	Restricción	Rango	Descripción
IdPrograma	Entero(4)	PK	1 a 9999	Código del Programa de estudios
Nombre_Prog	Texto(50)	No Nulo		Nombre del programa
Semestres	Entero(2)	No Nulo	1 a 12	Semestres a cursar
Titulo	Texto(30)	No Nulo		Titulo que otorga
Créditos	Entero(3)	No Nulo	1 a 999	Cantidad de créditos académicos
Registro_MEN	Texto(30)	No Nulo		Código del registro del Ministerio de Edu. Nal.

Estructura

Tabla Estudiantes

Identificador del Campo	Tipo Dato / Longitud	Restricción	Rango	Descripción
IdEstudiante	Entero(10)	PK	1.000.000 a 1.999.999.999	Numero de Identificación
Nombre_Est	Texto(25)	No Nulo		Nombre del estudiante
Apellidos_Est	Texto(25)	No Nulo	1 a 12	Apellidos del estudiante
Direccion_Est	Texto(50)	No Nulo		Dirección de residencia
Telefonos_Est	Texto(50)	No Nulo	1 a 999	Teléfonos de contacto

Estructura Tabla Matriculas

Identificador del Campo	Tipo Dato / Longitud	Restricción	Rango	Descripción
IdMatricula	Entero(10)	PK , AI	1 a 1.000.000.000	Numero consecutivo de la matricula
Año	Entero(4)	No Nulo	2000 a 9999	Año valido de la matricula
Periodo	Texto(1)	No Nulo	{ "A", "B" }	Periodo semestral
Nivel	Entero(2)	No Nulo	1 a 12	Semestre en el que esta el estudiante
Fecha_Mat	Fecha	No Nulo		Fecha de la matricula
CodEstudiante	Entero(10)	FK	1.000.000 a 1.999.999.999	Numero de Id del estudiante
CodPrograma	Entero(4)	FK	1 a 9999	Código del Programa de estudios

Script SQL

BDUnisabanera

//Crear la BD de la Universidad

create database bdunisabanera;

//Activar el uso de la BD creada

use bdunisabanera;

Script SQL

Tabla Programas

//Crear la tabla Programas

```
create table tblprogramas (  
idprograma int(4) not null primary key,  
nombre_pro varchar(50) not null,  
titulo varchar(30) not null,  
creditos int(3) not null,  
semestres int(2) not null,  
registro_men varchar(30) not null  
)ENGINE=InnoDB;
```

Script SQL

Tabla Estudiantes

//Crear la tabla Estudiantes

```
create table tblestudiantes (  
idestudiante int(10) not null primary key,  
nombre_est varchar(25) not null,  
apellidos_est varchar(25) not null,  
direccion varchar(50) not null,  
telefonos varchar(30) not null  
)ENGINE=InnoDB;
```

Script SQL

Tabla Matriculas

```
create table tblmatriculas (  
  idmatricula int(10) not null primary key auto_increment,  
  año int(4) not null,  
  periodo int(1) not null,  
  nivel int(2) not null,  
  fecha_mat date not null,  
  codprograma int(4) not null,  
  codestudiante int(10) not null,  
  foreign key (codprograma) references  
  tblprogramas(idprograma) on delete cascade on update  
  cascade,  
  foreign key (codestudiante) references  
  tblestudiantes(idestudiante) on delete cascade on update  
  cascade  
)ENGINE=InnoDB;
```