

# MODELO DE DATOS RELACIONAL

---

**Profesora: Mónica Caniupán**  
**Base de Datos - UBB**  
**2020**

## Introducción

- El modelo relacional se basa en el concepto de **relación**
- Una **relación** es una tabla con filas y columnas
- Una BD es una colección de una o más **relaciones distintas**

## Esquema e Instancia

- Un **esquema**:
  - Nombre de la relación,
  - Nombre de los atributos,
  - Dominios:
    - Tiene un nombre e.g. integer, string, etc.
    - Un conjunto de valores posibles
- Una **Instancia**:
  - Conjunto de **tuplas distintas**,
  - Una tabla con filas y columnas
    - Las filas especifican la **cardinalidad** de la relación
    - Las columnas especifican la **aridad o grado** de una relación

## Relación ALUMNOS

### 1. Esquema:

**ALUMNOS**(ID: integer, **NOMBRE**: varchar(20), **EDAD**: integer, **DIRECCION**: varchar(20))

### 2. Instancia:

ID	NOMBRE	EDAD	DIRECCION
11	Pedro	20	Rengo 1, Concepción
12	María	22	Prat 42, Concepción
13	Antonio	21	Prat 33, Concepción

## Relación ALUMNOS

T u p l a s	Atributos			
	ID	NOMBRE	EDAD	DIRECCION
	11	Pedro	20	Rengo 1, Concepción
	12	María	22	Prat 42, Concepción
	13	Antonio	21	Prat 33, Concepción

- **Aridad= 4**, la relación ALUMNOS tiene 4 atributos
- **Cardinalidad= 3**, existen tres tuplas, **todas distintas!!!**
- Generalmente nos referimos a la instancia de una relación como la **relación**

## Importante

- No existe un orden entre los atributos de una relación
  - Tampoco entre las tuplas
- Los SGBDs exigen orden en los atributos cuando se insertan tuplas
  - Los atributos deben tomar valores de acuerdo al correspondiente dominio del atributo

## Restricciones de Integridad

- Una Restricción de Integridad (RI):
  - Es una **condición** especificada sobre la BD
  - Debe ser satisfecha por toda instancia de la BD
  - Las RI son definidas junto con el esquema de la BD
- Una BD es consistente si **satisface** todas sus RI
- Un SGBD no debería permitir instancias inconsistentes

## Clave Primaria

- Un conjunto de atributos es una clave primaria si:
  - Dos tuplas distintas no tienen el mismo valor para los atributos en la clave
  - Ningún subconjunto de atributos en la clave identifica de manera unívoca a cada tupla
- **Ejemplo:** ALUMNOS(ID, NOMBRE, EDAD, DIRECCION)
  - El conjunto {ID, NOMBRE} no es una clave, sí lo es {ID}
  - {ID, NOMBRE} es una **SUPER CLAVE**, es decir, un conjunto de atributos que contiene una clave

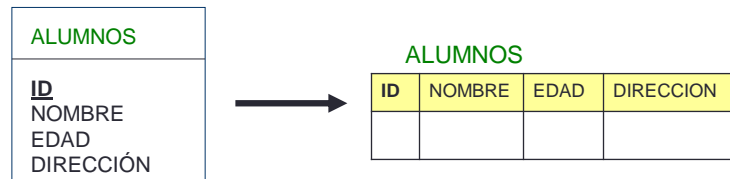
## Clave Candidata

- Un conjunto de atributos que identifica una tupla de manera unívoca es una **clave candidata** o simplemente **clave**
- Una relación puede tener varias claves candidatas, en cuyo caso, se elige una de ellas para ser la **clave primaria** de la relación

## Del MER al Modelo Relacional

1. Cada entidad es una relación (tabla) con el mismo nombre
2. Los atributos de la entidad pasan a ser los atributos de la relación
3. El identificador de la entidad es la **clave primaria** de la relación

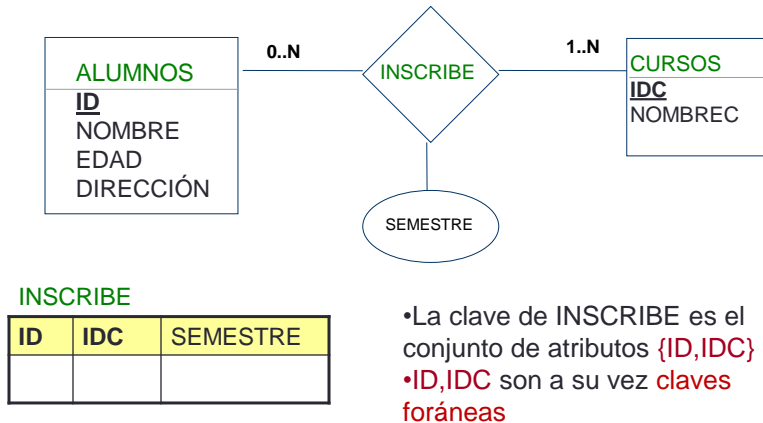
## Entidad a Relación



## Relaciones en MER

- ¿Deberíamos mapear cada relación en el MER a una tabla?
- La **cardinalidad** de las relaciones es fundamental para tomar una decisión

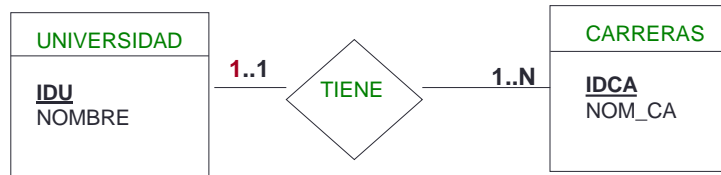
## Relaciones N a N



## Relaciones N a N

1. Si una relación binaria **R** entre las entidades  $E_1$  y  $E_2$  es **N a N** (muchos a muchos), entonces **R** es mapeada a una tabla **T**
2. La tabla **T** contiene las claves primarias de las tablas asociadas a las entidades  $E_1$  y  $E_2$
3. Si la relación **R** tiene atributos, estos pasan a ser atributos de la tabla **T**

## Relaciones N a 1



### CARRERAS

IDCA	NOM_CA	IDU

- La clave de CARRERAS es **IDCA**
- **IDU** es clave foránea en CARRERAS

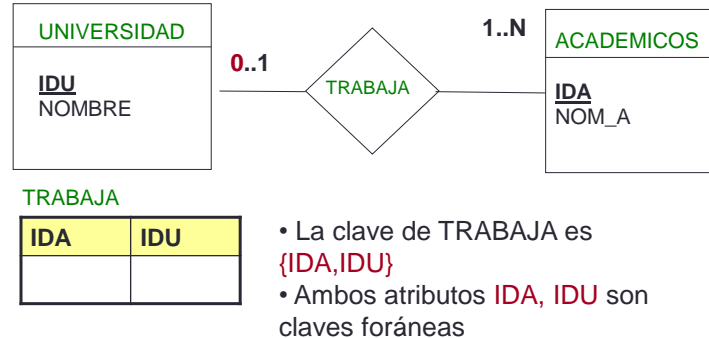
## Relaciones N a 1

1. Si una relación binaria **R** entre las entidades  $E_1$  y  $E_2$  es **N a 1**, **R** no se transforma en una tabla
2. Si la **cardinalidad mínima** de la relación entre **R** y  $E_2$  es 1, entonces la tabla  $E_2$  contendrá la clave primaria de la tabla  $E_1$
3. Los atributos de la relación **R** (si los hay) también se incluyen en la tabla  $E_2$
4. Sin embargo, si la **cardinalidad mínima** de la relación entre **R** y  $E_2$  es 0, se recomienda crear una tabla para **R**



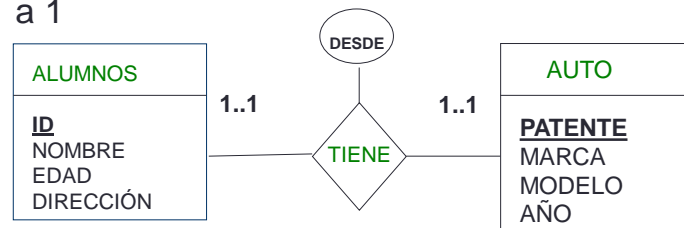
## Relaciones N a 1 con mínimo 0

- Si la **cardinalidad mínima** de la relación entre  $R$  y  $E_2$  es 0, se recomienda crear una tabla para  $R$



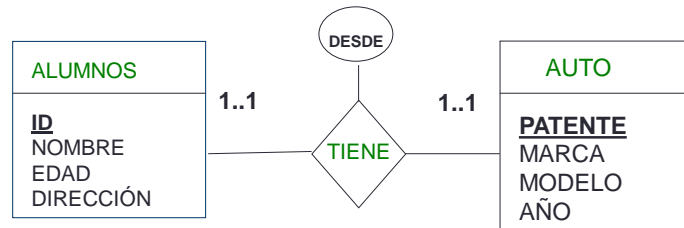
## Relaciones 1 a 1

1. Las relaciones tienen cardinalidad mínima igual a 1



- Se puede crear una sola tabla que combina ambas entidades, para evitar claves foráneas
- Los atributos de la relación pasan a ser atributos de esta única tabla

## Relaciones 1 a 1

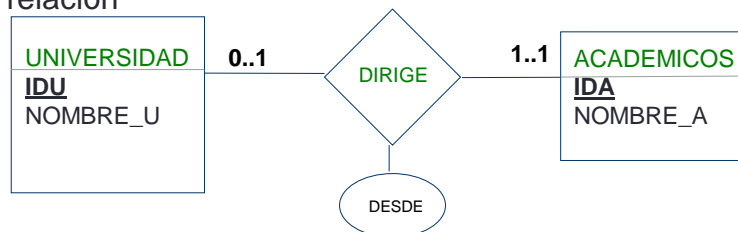


### ALUMNOS

ID	NOMBRE	EDAD	DIRECCION	PATENTE	MARCA	MODELO	AÑO	DESDE

## Relaciones 1 a 1 con mínimo 0

2. Una de las entidades tiene participación opcional en la relación

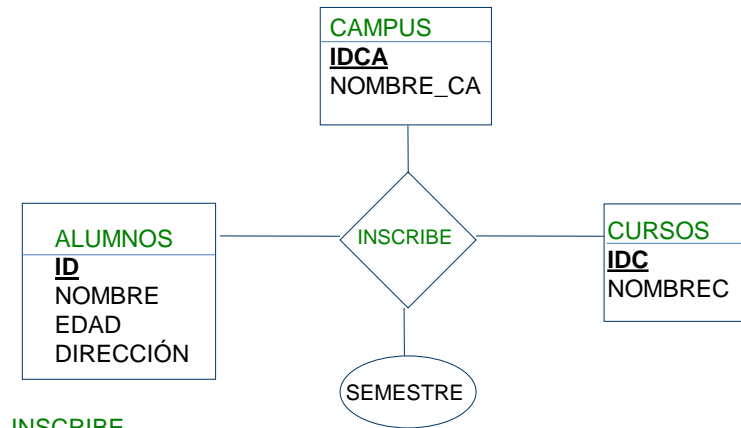


### UNIVERSIDAD

IDU	NOMBRE	IDA	DESDE

- La clave de esta relación es IDU
- IDA es clave foránea en UNIVERSIDAD

## Relaciones n-arias



INSCRIBE

ID	IDC	IDCA	SEMESTRE

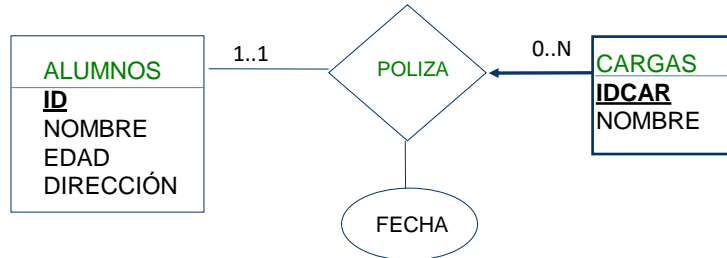
•La clave de INSCRIBE es el conjunto de atributos {ID, IDC, IDCA}

## Relaciones n-arias

- Crear una tabla **T** por la relación **R** entre entidades incluyendo como columnas las claves primarias de las entidades participantes
- Esta colección de claves primarias es la clave primaria de la tabla **T**
- **Se van a obtener *n* claves foráneas!!!**
- Los atributos de la relación **R** se agregan a la tabla **T**

## Relaciones Débiles

- Se tratan como relaciones N a 1



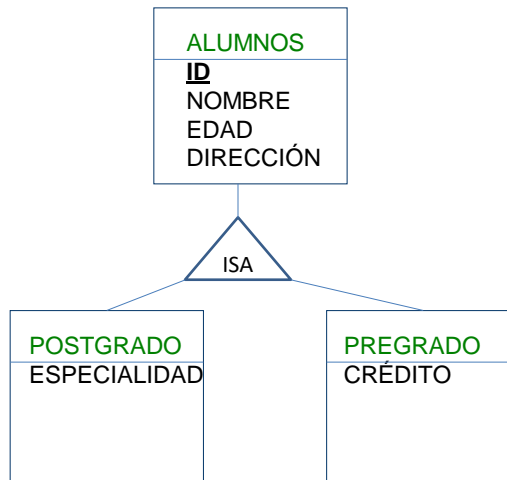
- Pero, la clave primaria de la tabla dependiente contiene la clave primaria de la tabla “fuerte”

**CARGAS**

IDCAR	ID	NOMBRE	FECHA

## Jerarquías

- Hay dos opciones:
  1. Crear tablas para cada entidad con las correspondientes claves foráneas
  2. Crear solo las tablas para las entidades “hijas” que contienen todos los atributos del padre



## Creación y Modificación de BDs

- En SQL una **relación** es una **Tabla**
- El Lenguaje de Definición de Datos es el subconjunto de **SQL** que permite:
  - Crear/eliminar/actualizar tablas
  - Insertar/eliminar/actualizar tuplas en las tablas

## LDD

- CREATE TABLE
- INSERT INTO TABLE
- DELETE FROM TABLE
- UPDATE TABLE
- DROP TABLE
- ALTER TABLE

## CREATE

- La siguiente instrucción crea la relación ALUMNOS:

```
CREATE TABLE ALUMNOS (  
    ID integer,  
    NOMBRE varchar(20),  
    EDAD integer,  
    DIRECCION varchar(20));
```

## INSERT

- La siguiente instrucción inserta una tupla en la relación ALUMNOS:

```
INSERT  
INTO ALUMNOS(ID, NOMBRE, EDAD, DIRECCION)  
VALUES (11, 'Pedro', 20, 'Rengo 1, Concepcion');
```

- Indicar el nombre de los atributos es opcional
- Si se omiten, los valores ingresados deben ser del tipo de datos para el atributo en la posición correspondiente

## DELETE

- La siguiente instrucción elimina todas las tuplas de ALUMNOS:

```
DELETE  
FROM ALUMNOS;
```

- La siguiente instrucción elimina todas las tuplas con NOMBRE “Pedro”:

```
DELETE  
FROM ALUMNOS  
WHERE NOMBRE = 'Pedro';
```

## UPDATE

- La siguiente instrucción actualiza la edad del alumno “Pedro”:

```
UPDATE ALUMNOS  
SET EDAD= EDAD+1  
WHERE NOMBRE = 'Pedro';
```

## DROP

- La siguiente instrucción elimina el esquema y las tuplas de la relación ALUMNOS

**DROP TABLE ALUMNOS;**

## ALTER TABLE

- La siguiente instrucción agrega un atributo a la relación ALUMNOS:

**ALTER TABLE ALUMNOS  
ADD COLUMN APELLIDOS varchar(20);**

- Esta columna tendrá el valor **NULL** para todas las tuplas en la tabla
- **NULL** en SQL significa que el valor es desconocido o no aplicable
- Para modificar el valor del atributo se debe usar el comando **UPDATE TABLE**



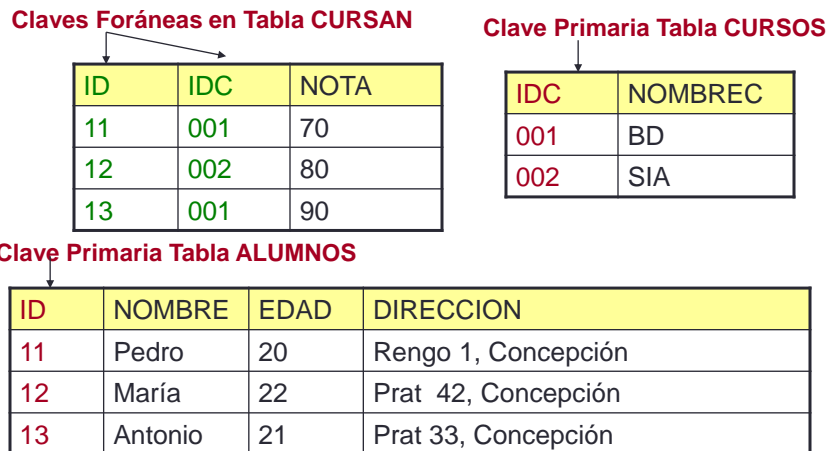
# Especificación de Claves con SQL

- Claves primarias con **PRIMARY KEY**

```
CREATE TABLE ALUMNOS (  
    ID integer,  
    NOMBRE varchar(20),  
    EDAD integer,  
    DIRECCION varchar(20),  
    PRIMARY KEY (ID));
```

## Claves Foráneas

Claves Foráneas en Tabla CURSAN



ID	IDC	NOTA
11	001	70
12	002	80
13	001	90

Clave Primaria Tabla CURSOS

IDC	NOMBREC
001	BD
002	SIA

Clave Primaria Tabla ALUMNOS

ID	NOMBRE	EDAD	DIRECCION
11	Pedro	20	Rengo 1, Concepción
12	María	22	Prat 42, Concepción
13	Antonio	21	Prat 33, Concepción

## Especificación Claves Foráneas

```
CREATE TABLE CURSAN(  
    ID integer,  
    IDC integer,  
    NOTA integer,  
    PRIMARY KEY (ID, IDC),  
    FOREIGN KEY (ID) REFERENCES ALUMNOS,  
    FOREIGN KEY (IDC) REFERENCES CURSOS);
```

## Claves Foráneas

- Las claves foráneas pueden hacer referencia a la misma relación
- **ALUMNOS\_2**(ID: integer, NOMBRE: varchar(20),  
IDCG: integer)
  - **IDCG** hace referencia al **ID** de otro alumno de la relación **ALUMNOS**
- **CREATE TABLE ALUMNOS\_2**(  
 **ID** integer,  
 **NOMBRE** varchar(20),  
 **IDCG** integer,  
 **PRIMARY KEY (ID),**  
 **FOREIGN KEY (IDCG) REFERENCES ALUMNOS);**

## Claves Foráneas

- ALUMNOS\_2

ID	NOMBRE	IDCG
11	Pedro	13
12	María	NULL
13	Antonio	11

- Cada Alumno tiene un compañero de grupo
- Si un Alumno no tiene, el campo IDCG debe tomar un valor por defecto que no sea **NULL (nulo)**
  - Un atributo que es parte de una clave no puede tomar el valor **NULL!!!**

## Operaciones que afectan las RI

- ¿Qué acciones deberíamos tomar si se inserta una tupla en **CURSAN** con un **ID** que no existe en la tabla **ALUMNOS**?

**CURSAN**

ID	IDC	NOTA
11	001	70
12	002	80
13	001	90
14	001	89

## Operaciones que afectan las RI

- La operación debe ser prohibida

ID	IDC	NOTA
11	001	70
12	002	80
13	001	90
14	001	89

## Operaciones que afectan las RI

- ¿Qué acciones deberíamos tomar si se elimina una tupla en la tabla **ALUMNOS**?

CURSAN

ID	IDC	NOTA
11	001	70
12	002	80
13	001	90

ALUMNOS

ID	NOMBRE	EDAD	DIRECCION
11	Pedro	20	Rengo 1, Concepción
12	María	22	Prat 42, Concepción
13	Antonio	21	Prat 33, Concepción

## Operaciones que afectan las RI

1. Eliminar todas las tuplas en la tabla **CURSAN** que hacen referencia al **ID** eliminado

ID	IDC	NOTA
11	001	70
12	002	80
<del>13</del>	<del>001</del>	<del>90</del>

2. No permitir la eliminación de la tupla en **ALUMNOS** si existen tuplas en **CURSAN** que hacen referencia a ella

## Operaciones que afectan las RI

3. Asignar a la columna **ID** el valor de otro **ID** por defecto, por cada fila afectada

CURSAN		
ID	IDC	NOTA
11	001	70
12	002	80
12	001	90

¿Problemas de esta solución?

## Operaciones que afectan las RI

4. Asignar a la columna **ID** el valor **NULL**

**CURSAN**

ID	IDC	NOTA
11	001	70
12	002	80
NULL	001	90

¿Problemas de esta solución?

## Operaciones que afectan las RI

- Las acciones nombradas anteriormente también se aplican cuando la clave primaria de una relación se modifica a través de **UPDATE**

## Satisfacción de RI con SQL

- SQL nos permite especificar cada una de estas situaciones
- Para ello nos entrega cuatro opciones para las operaciones **DELETE** y **UPDATE**

## Satisfacción de RI con SQL

1. **NO ACTION**: tanto **DELETE** como **UPDATE** se prohíben si hay tuplas afectadas (es la opción por defecto)
  - Se permite una modificación de **ID** en **ALUMNOS** sólo si **no** existen tuplas en **CURSAN** que hagan referencia al **ID**
2. **CASCADE**: si se elimina (modifica) una tupla en **ALUMNOS** se eliminan (modifican) todas las tuplas en **CURSAN** que hagan referencia al alumno eliminado (modificado)

## Satisfacción de RI con SQL

- Ejemplo:

```
CREATE TABLE CURSAN(  
    ID integer,  
    IDC integer,  
    NOTA integer,  
    PRIMARY KEY (ID, IDC),  
    FOREIGN KEY (ID) REFERENCES ALUMNOS,  
    ON DELETE CASCADE,  
    ON UPDATE NO ACTION);
```

- Se puede especificar lo mismo para el atributo IDC

## Eliminación en Cascada

ALUMNOS

ID	NOMBRE	EDAD	DIRECCION
11	Pedro	20	Rengo 1, Concepción
12	María	22	Prat 42, Concepción
13	Antonio	21	Prat 33, Concepción

CURSAN

ID	IDC	NOTA
11	001	70
12	002	80
13	001	90

- Si se elimina la tupla en Alumnos con ID 13 se eliminan todas las tuplas en Cursan para ese alumno



## Satisfacción de RI con SQL

3. **ON DELETE SET DEFAULT:** al eliminarse una tupla en **ALUMNOS** se asigna el **ID** del alumno por defecto a las tuplas afectadas en **CURSAN**

- El valor por defecto se especifica cuando se crea la tabla ALUMNOS

```
CREATE TABLE ALUMNOS (  
  ID integer DEFAULT 4545 ...
```

4. **ON DELETE SET NULL**