

# Base de Datos

DC. FCEyN  
2024

Emilio Platzer  
[tute@dc.uba.ar](mailto:tute@dc.uba.ar)

# Formas Normales - Dependencia Funcional

Dependencia funcional:  $X \rightarrow Y$

Donde: X e Y son conjuntos de atributos en R

Cuando  $\forall t_1, t_2$  Si  $t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$

Atención: La dependencia funcional es una propiedad de los datos,  
no es una definición de diseño de cómo quisiéramos que sean.

Atención 2: Los datos son los del universo que el conjunto de tuplas.

# Formas Normales - Claves Candidatas

X es clave candidata de R si y solo sí:

- $X \rightarrow R$
- $\nexists Y \subset X / Y \rightarrow R$

# Formas Normales - Descomposición

## Modelo de ejemplo 1: Profesores y Materias

El Departamento de Computación tiene una base de datos de profesores, donde se registran:

- sus datos personales
- las materias que está en condiciones de dictar
- la materia que dicta o dictó cada cuatrimestre
- su materia preferida

En cada cuatrimestre hay un docente por materia.

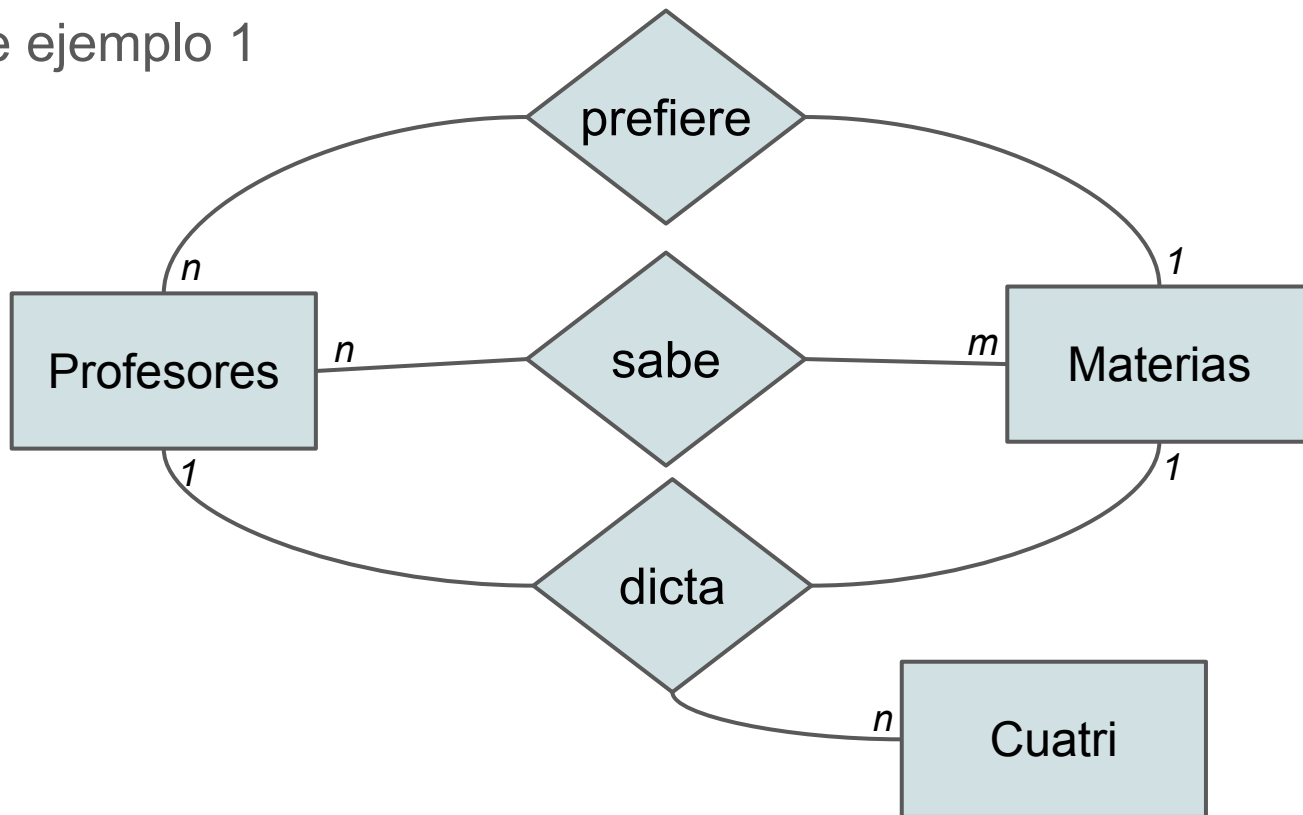
Todas las materias se dictan todos los cuatrimestres.

Dentro de lo posible se tratará de respetar las preferencias de los profesores.

<u>idMat</u>	nombre	labo
MN	Métodos	no
BD	Base	Sí
TJ	Juegos	no
A1	Álgebra 1	no
R	Redes	no
AE1	Aglo 1	sí

# Formas Normales - Descomposición

Modelo de ejemplo 1



# Formas Normales - Violaciones a la 1FN, descomposiciones

Tabla Profesores

<u>idProf</u>	apellido	nombre	sabe	C_23_2	C_23_1	C_22_2
1	Méndez	Isabel	MN, MD, LC, AE3	MN	MN	MN
2	Zabala	Paula	A1, MD, AE1, AE2, AE3	A1	A1	AE1
3	Righetti	Claudio	TC, R, ACD, SI	ACD	ACD	R
4	Platzer	Emilio	BD, O2, TJ, MD	BD	null	null

idMat	nombre	labo
MN	Métodos	no
BD	Base	Sí
TJ	Juegos	no
A1	Álgebra 1	no

# Formas Normales - Violaciones a la 2FN, descomposiciones

Tabla ProfesoresD

<u>idProf</u>	apellido	nombre	<u>Cuatri</u>	<u>idMat</u>	nomMat	labo
1	Méndez	Isabel	23_2	MN	Métodos	no
2	Zabala	Paula	23_2	A1	Algebra 1	no
3	Righetti	Claudio	23_2	ACD	ArquiCom	no
3	Righetti	Claudio	23_1	R	Redes	no
4	Platzer	Emilio	23_2	BD	BD	sí

**Atención** hay casos donde la violación es aparente, lo que se está reflejando es el atributo vigente en ese cuatrimestre

# Formas Normales - Violaciones a la 3FN, descomposiciones

Tabla Profesores

<u>idProf</u>	apellido	nombre	idPref	nomMat
0	Loiseau	Irene	MD	Discreta
1	Méndez	Isabel	MN	Métodos
2	Zabala	Paula	MD	Discreta
3	Righetti	Claudio	R	Redes
4	Platzer	Emilio	VG	VideoGames

**Atención** hay casos donde la violación es aparente por ejemplo cuando se registran datos declarados



# Formas Normales - Descomposición

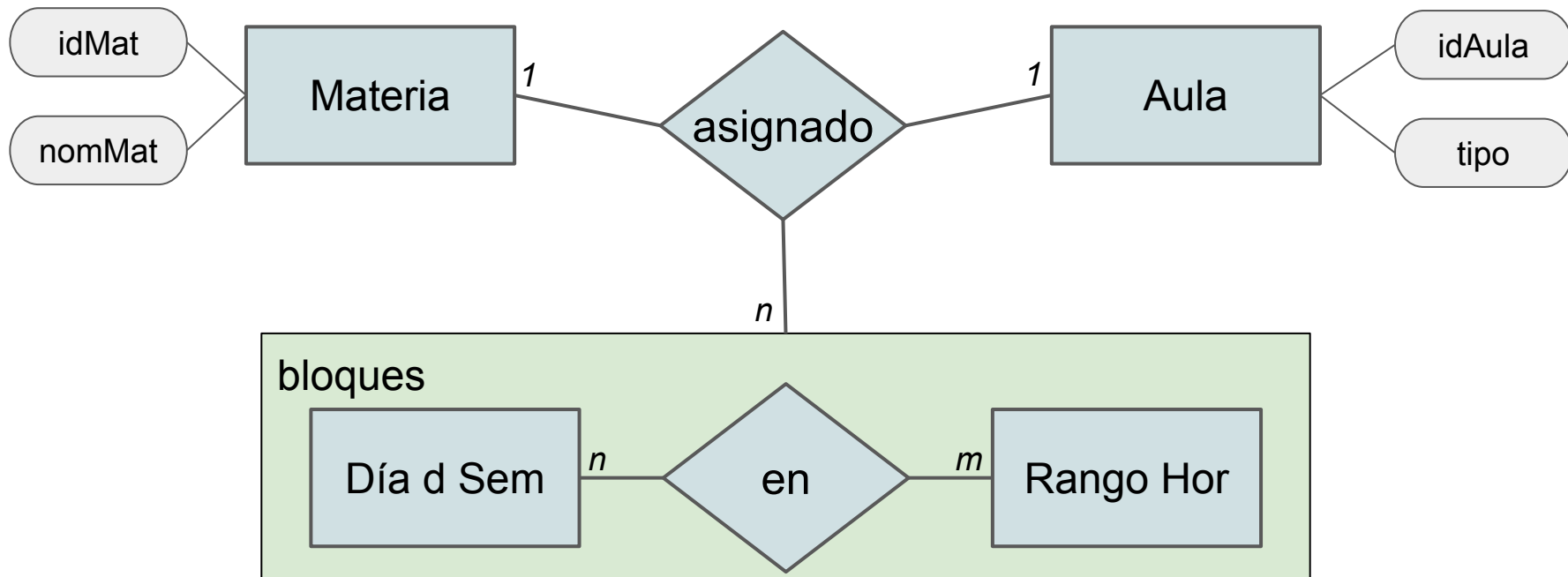
## Modelo de ejemplo 2: Asignación de Materias a Aulas

Cada materia se dicta en ciertos días de la semana, en determinado horario cada día. Ej BD miércoles de 19 a 22 y viernes de 17 a 22. Algunas clases se dan en laboratorios, otras en aulas normales. Las restricciones son:

- En cada aula debe dictarse al menos dos horas seguidas de cada materia.
- Una materia podría, el mismo día, tener clases en un laboratorio y en un aula normal (obviamente no al mismo tiempo).

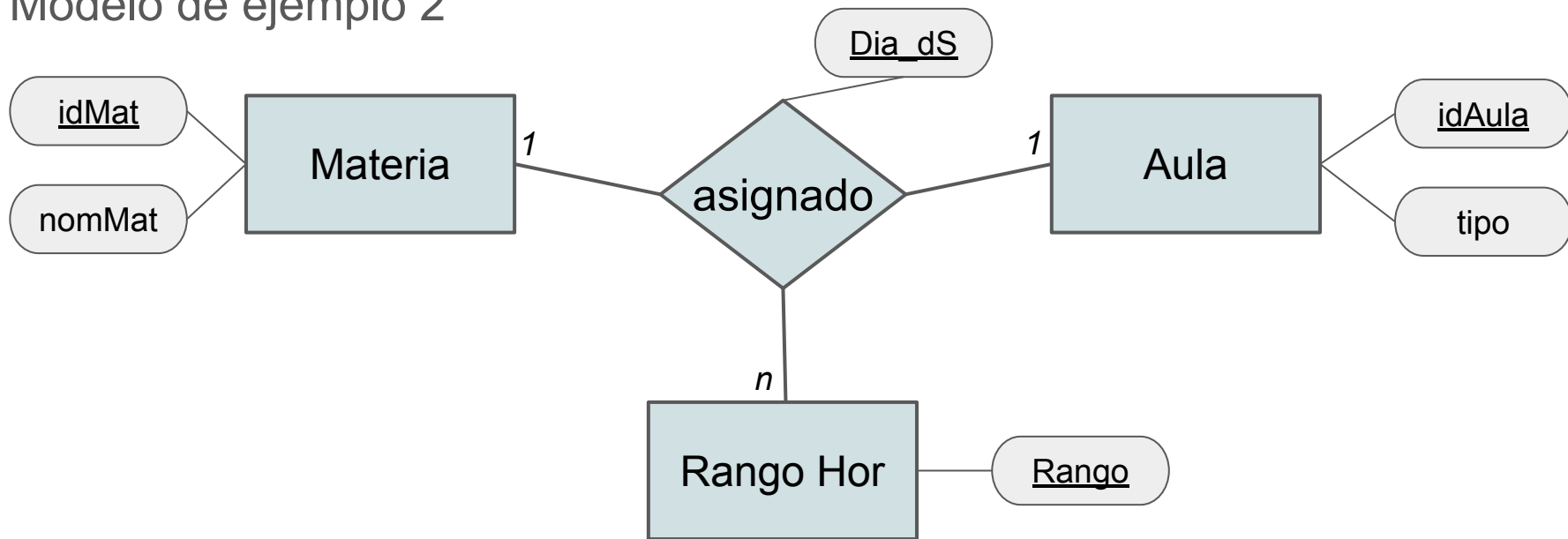
# Formas Normales - Descomposición

## Modelo de ejemplo 2



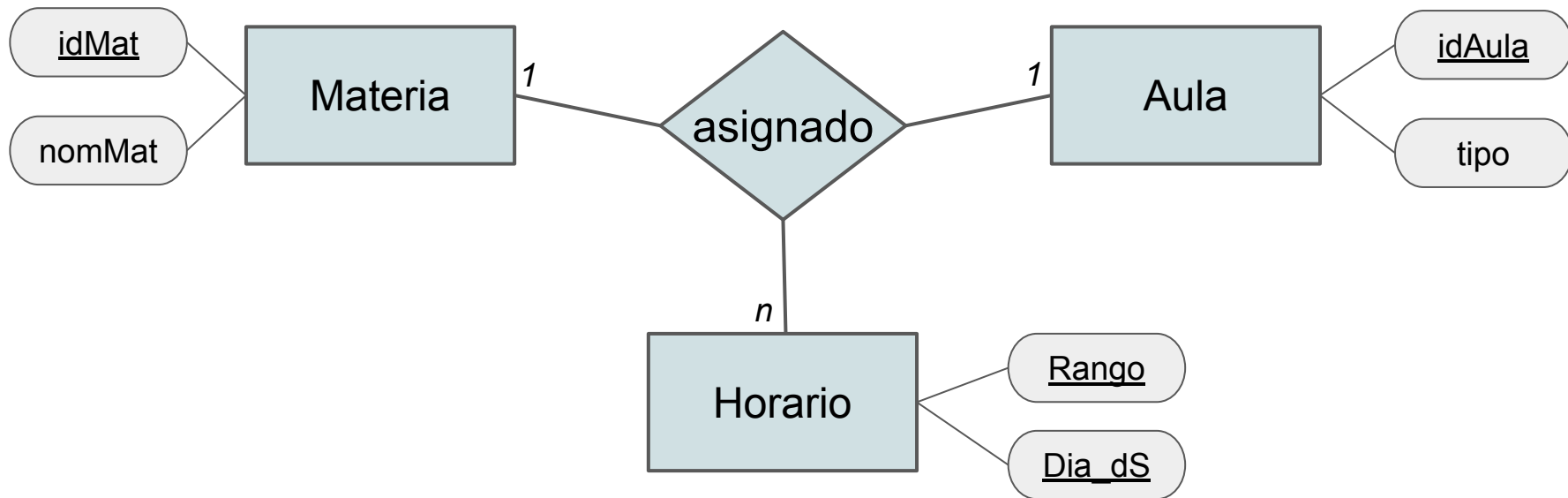
# Formas Normales - Descomposición

## Modelo de ejemplo 2



# Formas Normales - Descomposición

## Modelo de ejemplo 2



# Formas Normales - Violaciones a la BCNF, descomposiciones

## Tabla Asignaciones

idMat	<u>idAula</u>	tipo	<u>dia</u>	<u>turno</u>
BD		1 nor	mie	[19; 22)
BD		1 nor	vie	[17; 19)
BD		2 labo	vie	[19; 22)
MN		1 nor	mie	[17; 19)
MN		3 nor	vie	[17; 19)

**Atención:** Una tabla en 3FN que no está en BCNF no es necesariamente un error de diseño. Puede ser solo un detalle de implementación o puede ser mejorar la expresividad de la solución.

# Expresividad de lenguajes de consulta

- Se puede demostrar que AR, CRT tienen la misma expresividad y la Lógica de Primer Orden (para consultas seguras)
- SQL tiene al menos el mismo poder de expresión. Extensiones:
  - Aritmética compleja
  - Recursión
  - Agregación
  - Funciones almacenadas

# Expresividad de lenguajes de consulta - Otros conceptos

- Lenguajes imperativos (con o sin OO)
- Lenguajes funcionales
- Lenguajes Lógicos (PROLOG)
- SQL

# Expresividad de lenguajes de consulta - Null en SQL

¿Qué pasa cuando no hay dato? (Información faltante, missing data)

- El valor de un atributo de una entidad es desconocido
- No corresponde por la naturaleza de los valores de otros atributos (por ejemplo un docente ad honorem no tiene sueldo)
- Not classified (cuando no hay una categoría que corresponda a los valores posibles del atributo)
- Dato recibido erróneo (cuando al momento de insertar se sabe que hay un dato pero el dato es inconsistente, o con error de representación para el tipo de dato o el resultado de algo imposible (ej div 0))



# Expresividad de lenguajes de consulta - Aritmética de Null

En SQL la semántica de null equivale a desconocido

- $a + \text{null} \Rightarrow \text{null}$
- $0 * \text{null} \Rightarrow \text{null}$  (¿por qué si 0 por cualquier cosa es 0?)
- $a = \text{null} \Rightarrow \text{null}$
- $\text{null} = \text{null} \Rightarrow \text{null}$  (desconocido = desconocido no es true?)
- $\text{null} <> a \Rightarrow \text{null}$
- $\text{'nombre'} || \text{' '} || \text{'apellido'} \Rightarrow \text{'nombre apellido'}$
- $\text{null} || \text{' '} || \text{'apellido'} \Rightarrow \text{null}$
- $\text{null is null} \Rightarrow \text{true}$
- $\text{null is not null} \Rightarrow \text{false}$

# Expresividad de lenguajes de consulta - Lógica de 3 estados

null equivale a desconocido en expresiones lógicas

AND	true	false	null
true	true	false	null
false	false	false	false
null	null	false	null

OR	true	false	null
true	true	true	true
false	true	false	null
null	true	null	null

=	true	false	null
true	true	false	null
false	false	true	null
null	null	null	null

IS	true	false	null
true	true	false	false
false	false	true	false
null	false	false	true

salvo para las siguientes operaciones:

- IS (= entre bool)
- IS DISTINCT FROM ( $\neq$  any type)
- CONCAT
- todas las funciones de agregación: SUM, AVG, MIN, MAX

NOT	true	false	null
	false	true	null

# Expresividad de lenguajes de consulta - Anomalías con NULL

$SUM(a + b) \not\equiv SUM(a) + SUM(b)$

y si a y b no pueden contener números negativos:

$SUM(a + b) < SUM(a) + SUM(b)$

$CASE\ WHEN\ a > 0\ THEN\ b\ ELSE\ c\ END \not\equiv CASE\ WHEN\ a \leq 0\ THEN\ b\ ELSE\ c\ END$

$if\ A > 0\ then\ P1()\ else\ P2()\ end\ if; \not\equiv if\ A \leq 0\ then\ P2()\ else\ P1()\ end\ if;$

# Expresividad de lenguajes de consulta - NULL en claves

- Las claves primarias no pueden contener NULL (no importa si son compuestas o no)
- Las claves únicas pueden contener NULL (se entiende que son claves parciales). Postgres puede definir claves parciales con cualquier otro filtro (además del filtro por defecto IS NOT NULL)
- Los campos que hacen referencia a claves foráneas pueden ser null (salvo que el campo esté definido como not null)
  - En claves compuestas hay 3 comportamientos posibles: FULL MATCH (o son todas null o todas no null), SIMPLE MATCH (si hay un null se ignora la relación), PARTIAL MATCH (si hay un null se busca si hay al menos un registro compatible en la tabla relacionada).