

# Modelos y bases de datos

## Paso al modelo relacional

CEIS

2023-2

# Agenda

## Modelo Relacional

- Relación-Tabla

- Tres aspectos

## De Conceptual a Logico

- Patrones

- Conceptos

- Relaciones

## De lógico a código

- Diseño lógico

- Código

# Agenda

## Modelo Relacional

### Relación-Tabla

Tres aspectos

## De Conceptual a Logico

Patrones

Conceptos

Relaciones

## De lógico a código

Diseño lógico

Código

# Modelo relacional

Relación

Tabla

# Modelo relacional

## Relación

Es una propiedad que asigna un valor de verdad a combinaciones de  $k$  datos ( $k$ -tuplas)

## Capitales

$\text{capitales}(p, c)$ :-  $c$  es la capital de  $p$

## Tabla

Es una estructura bidimensional (filas, columnas) que contiene información

CAPITALES

PAIS	CAPITAL
COLOMBIA	BOGOTA
FRANCIA	PARIS
ESPAÑA	MADRID
LESOTO	MASERU

# Modelo relacional

## Relación

Es una propiedad que asigna un valor de verdad a combinaciones de  $k$  datos ( $k$ -tuplas)

## Capitales

$\text{capitales}(p, c)$ :-  $c$  es la capital de  $p$

## Tabla

Es una estructura bidimensional (filas, columnas) que contiene información

CAPITALES

PAIS	CAPITAL
COLOMBIA	BOGOTA
FRANCIA	PARIS
ESPAÑA	MADRID
LESOTO	MASERU

Todo es relación - Todo es tabla

# Modelo relacional

## Capitales

$\text{capitales}(p,c)$ :- c es la capital de p

CAPITALES

PAIS	CAPITAL
COLOMBIA	BOGOTA
FRANCIA	PARIS
ESPAÑA	MADRID
LESOTO	MASERU

## Tres consecuencias

- ▶ Una relación tiene asignada un **predicado** (la propiedad)
- ▶ Las tuplas de la relación denotan proposiciones verdaderas derivadas de ese predicado
- ▶ Si cierta tupla no existe en una relación, podemos afirmar que la proposición correspondiente es falsa

# Modelo relacional

## Capitales

$\text{capitales}(p,c)$ :- c es la capital de p

CAPITALES

PAIS	CAPITAL
COLOMBIA	BOGOTA
FRANCIA	PARIS
ESPAÑA	MADRID
LESOTO	MASERU

## Tres consecuencias

- ▶ Una relación tiene asignada un **predicado** (la propiedad)  
*¿Cuál sería la propiedad?*
- ▶ Las tuplas de la relación denotan proposiciones verdaderas derivadas de ese predicado
- ▶ Si cierta tupla no existe en una relación, podemos afirmar que la proposición correspondiente es falsa



# Modelo relacional

## Capitales

$\text{capitales}(p,c)$ :- c es la capital de p

CAPITALES

PAIS	CAPITAL
COLOMBIA	BOGOTA
FRANCIA	PARIS
ESPAÑA	MADRID
LESOTO	MASERU

## Tres consecuencias

- ▶ Una relación tiene asignada un **predicado** (la propiedad)
- ▶ Las tuplas de la relación denotan proposiciones verdaderas derivadas de ese predicado  
*¿Bogotá es la capital de Colombia?*
- ▶ Si cierta tupla no existe en una relación, podemos afirmar que la proposición correspondiente es falsa

# Modelo relacional

## Capitales

$\text{capitales}(p, c)$ :-  $c$  es la capital de  $p$

CAPITALES

PAIS	CAPITAL
COLOMBIA	BOGOTA
FRANCIA	PARIS
ESPAÑA	MADRID
LESOTO	MASERU

## Tres consecuencias

- ▶ Una relación tiene asignada un **predicado** (la propiedad)
- ▶ Las tuplas de la relación denotan proposiciones verdaderas derivadas de ese predicado

¿Maseru es la capital de Lesoto?

- ▶ Si cierta tupla no existe en una relación, podemos afirmar que la proposición correspondiente es falsa

# Modelo relacional

## Capitales

$\text{capitales}(p,c)$ :- c es la capital de p

CAPITALES

PAIS	CAPITAL
COLOMBIA	BOGOTA
FRANCIA	PARIS
ESPAÑA	MADRID
LESOTO	MASERU

## Tres consecuencias

- ▶ Una relación tiene asignada un **predicado** (la propiedad)
- ▶ Las tuplas de la relación denotan proposiciones verdaderas derivadas de ese predicado
- ▶ Si cierta tupla no existe en una relación, podemos afirmar que la proposición correspondiente es falsa

¿Cali es la capital de Colombia?

# Modelo relacional

## Capitales

$\text{capitales}(p, c)$ :-  $c$  es la capital de  $p$

CAPITALES

PAIS	CAPITAL
COLOMBIA	BOGOTA
FRANCIA	PARIS
ESPAÑA	MADRID
LESOTO	MASERU

## Tres consecuencias

- ▶ Una relación tiene asignada un **predicado** (la propiedad)
- ▶ Las tuplas de la relación denotan proposiciones verdaderas derivadas de ese predicado
- ▶ Si cierta tupla no existe en una relación, podemos afirmar que la proposición correspondiente es falsa

¿Quito es la capital de Ecuador?

# La regla de oro

## La regla de oro

**Nunca debe permitirse una operación de actualización que deje cualquier variable de relación en un estado que viole su propio predicado.**

# La regla de oro

## La regla de oro

**Nunca debe permitirse una operación de actualización que deje cualquier variable de relación en un estado que viole su propio predicado.**

PAIS	CAPITAL
COLOMBIA	BOGOTA
FRANCIA	PARIS
ESPAÑA	MADRID
LESOTO	MASERU
BRASIL	RIO DE JANEIRO

¿Rio de Janeiro es la capital de Brasil ?

# Modelo relacional

## Capitales

$\text{capitales}(p, c)$ :-  $c$  es la capital de  $p$

PAIS	CAPITAL
COLOMBIA	BOGOTA
FRANCIA	PARIS
ESPAÑA	MADRID
LESOTO	MASERU

## Propiedades de relaciones

- ▶ No existen tuplas duplicadas
- ▶ Los tuplas no tienen orden
- ▶ Los atributos no tienen orden
- ▶ Cada tupla tiene un único valor para cada atributo

# Modelo relacional

## Capitales

$\text{capitales}(p, c)$ :-  $c$  es la capital de  $p$

PAIS	CAPITAL
COLOMBIA	BOGOTA
FRANCIA	PARIS
ESPAÑA	MADRID
LESOTO	MASERU

## Propiedades de relaciones

- ▶ No existen tuplas duplicadas
- ▶ Los tuplas no tienen orden
- ▶ Los atributos no tienen orden
- ▶ Cada tupla tiene un único valor para cada atributo

¿Se cumplen en SQL?



# Modelo relacional

## Capitales

capitales(p,c):- c es la capital de p

PAIS	CAPITAL
COLOMBIA	BOGOTA
FRANCIA	PARIS
ESPAÑA	MADRID
LESOTO	MASERU

## Informal-Formal

<i>Término relacional formal</i>	<i>Equivalente informal</i>
relación	tabla
tupia	fila o registro
cardinalidad	número de filas
atributo	columna o campo
grado	número de columnas
clave primaria	identificador único
dominio	conjunto de valores válidos

# Agenda

## Modelo Relacional

Relación-Tabla

Tres aspectos

## De Conceptual a Logico

Patrones

Conceptos

Relaciones

## De lógico a código

Diseño lógico

Código

# Modelo relacional

## Aspectos basicos

- ▶ **Aspecto estructural**

Los usuarios perciben la información de la base de datos como tablas y nada más que tablas

- ▶ **Aspecto de integridad**

Estas tablas satisfacen ciertas restricciones de integridad

- ▶ **Aspecto de manipulación**

Las operaciones disponibles permiten derivar tablas a partir de tablas

# Integridad-Claves

## Clave candidata

El valor no se puede repetir

## Clave externa

El valor debe existir en la tabla referenciada, sino es nulo.

El valor debe ser una clave candidata en la tabla referenciada

# Integridad-Claves

## Clave candidata

El valor no se puede repetir

- ▶ **Clave primaria**
  - ▶ Es el indentificador seleccionado
  - ▶ Ningún componente de la clave primaria puede se nulo.
- ▶ **Clave única**
  - ▶ Puede tener valores nulos

## Clave externa

El valor debe existir en la tabla referenciada, sino es nulo.

El valor debe ser una clave candidata en la tabla referenciada

# Modelo relacional Mini

## Equipos

Personas(cedula, alias, nombre, nacimiento !)

Hinchas(persona, equipo)

Equipos(nombre, pais)

           Primaria  
           Unica  
           Foranea  
! Puede no conocerse

¿ A conceptual?

# Modelo relacional Mini

## Equipos

Personas(cedula, alias, nombre, nacimiento !)

Hinchas(persona, equipo)

Equipos(nombre, pais)

           Primaria  
           Unica  
           Foranea  
           !  
           Puede no conocerse

## ¿ A conceptual?

1. ¿Número de relaciones en relacional?
2. ¿Número de asociaciones en conceptual?

# Agenda

## Modelo Relacional

Relación-Tabla

Tres aspectos

## De Conceptual a Logico

Patrones

Conceptos

Relaciones

## De lógico a código

Diseño lógico

Código



# Patrones

## Definición

Un patrón es un par problema/solución con nombre que se puede aplicar en nuevos contextos, con consejos acerca de cómo aplicarlos en nuevas situaciones y discusiones sobre sus puntos fuertes y débiles.

# Patrones

## De Persistencia

1. Representar conceptos como tablas
2. Identificador de conceptos
3. Representar relaciones como tablas
4. Representar relaciones de herencia como tablas

# Agenda

## Modelo Relacional

Relación-Tabla

Tres aspectos

## De Conceptual a Logico

Patrones

**Conceptos**

Relaciones

## De lógico a código

Diseño lógico

Código

# Patrones

## De Persistencia

1. Representar conceptos como tablas
2. Identificador de conceptos
3. Representar relaciones como tablas
4. Representar relaciones de herencia como tablas

Asistente
codigo : _ nombre : _

Profesor
<<I>> cedula : _ <<U>> nombre : _

Proyecto
id : _ fecha : _[0..1]

# Patrones

## De Persistencia

1. Representar conceptos como tablas
2. Identificador de conceptos
3. Representar relaciones como tablas
4. Representar relaciones de herencia como tablas

Asistente
codigo : _ nombre : _

Profesor
<< >> cedula : _ <<U>> nombre : _

Universidad
nombre : _ inicio : _ carreras : _[1..*]

Proyecto
id : _ fecha : _[0..1]

# Estructura

## Nombre

Representación de objetos como tablas

*Representing Objects as Tables*

## Problema

¿ Cómo representar los conceptos en un esquema de base de datos relacional ?

## Solución

Definir una tabla para cada concepto persistente. Los atributos del concepto que son tipos primitivos deben ser las columnas de la tabla. Los atributos del concepto que son colecciones deben representarse en tablas aparte.

## Referencia

[Brown,1996]

# Estructura

## Nombre

Identificador de Objeto

*Object Identifier*

## Problema

¿ Cómo relacionar adecuadamente los objetos con las tuplas ?

## Solución

Asignar un identificador a cada concepto y a la tupla correspondiente.

## Referencia

[Brown,1996]

# Agenda

## Modelo Relacional

Relación-Tabla

Tres aspectos

## De Conceptual a Logico

Patrones

Conceptos

**Relaciones**

## De lógico a código

Diseño lógico

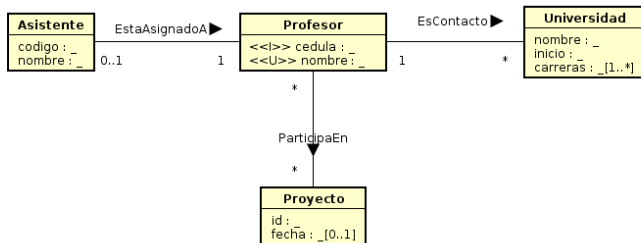
Código



# Patrones

## De Persistencia

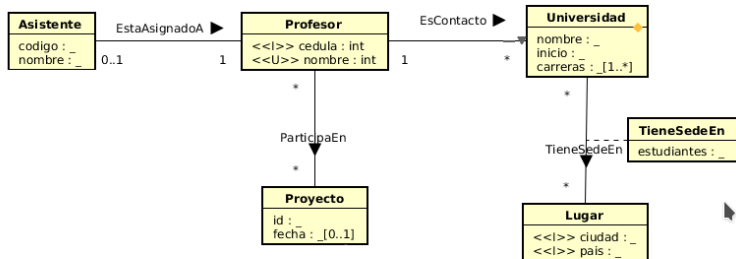
1. Representar conceptos como tablas
2. Identificador de conceptos
3. Representar relaciones como tablas
4. Representar relaciones de herencia como tablas



# Patrones

## De Persistencia

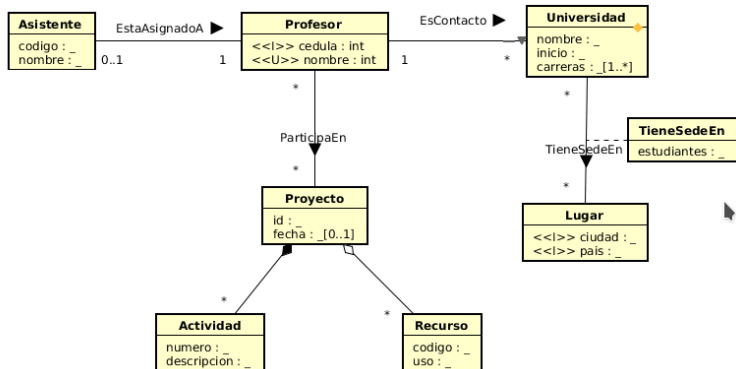
1. Representar conceptos como tablas
2. Identificador de conceptos
3. Representar relaciones como tablas
4. Representar relaciones de herencia como tablas



# Patrones

## De Persistencia

1. Representar conceptos como tablas
2. Identificador de conceptos
3. Representar relaciones como tablas
4. Representar relaciones de herencia como tablas



# Estructura

## Nombre

Representación de relaciones como tablas

*Representing Object Relationships as Tables*

## Problema

¿ Cómo representar una relación en un esquema de base de datos relacional ?

## Solución

- ▶ Crear una tabla asociativa para registrar los identificadores de cada uno de los objetos de la relación
- ▶ Para relaciones uno a uno o uno a muchos, Colocar una clave foránea en una de las tablas para representar la relación entre los objetos

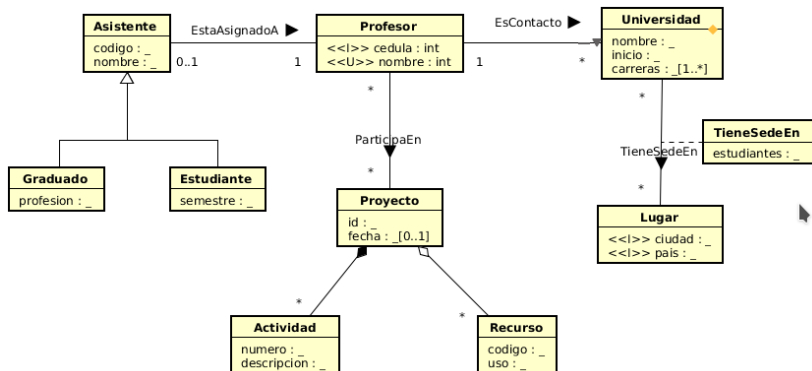
## Referencia

[Brown,1996]

# Patrones

## De Persistencia

1. Representar conceptos como tablas
2. Identificador de conceptos
3. Representar relaciones como tablas
4. Representar relaciones de herencia como tablas



# Estructura

## Nombre

Representación de herencia en una base de datos relacional

*Representing Inheritance in a Relational Database*

## Problema

¿ Cómo representar una relación de herencia en un esquema de base de datos relacional ?

## Solución

- ▶ Una tabla para el superconcepto y una para cada subconcepto (con los atributos propios)
- ▶ Una tabla para cada subconcepto (con todos los atributos) y, si no es abstracto, una para el superconcepto.
- ▶ Sólo una tabla con toda la información

## Referencia

[Brown,1996]

# Agenda

## Modelo Relacional

Relación-Tabla

Tres aspectos

## De Conceptual a Logico

Patrones

Conceptos

Relaciones

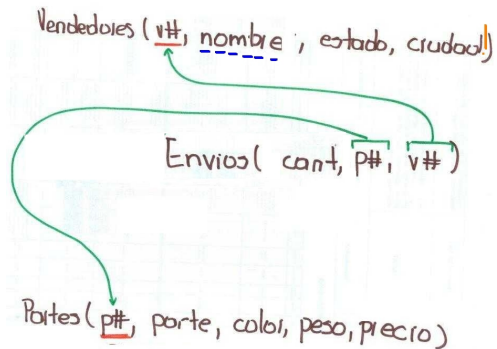
## De lógico a código

Diseño lógico

Código

# Diseño Lógico

## Modelo mini





# Agenda

## Modelo Relacional

Relación-Tabla

Tres aspectos

## De Conceptual a Logico

Patrones

Conceptos

Relaciones

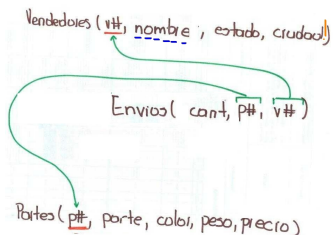
## De lógico a código

Diseño lógico

Código

# Aspecto estructural

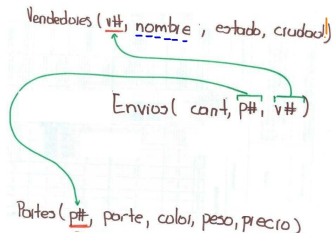
## Tablas



```
CREATE TABLE VENEDORES(  
  v#          CHAR(2) NOT NULL,  
  nombre      VARCHAR(20) NOT NULL,  
  estado      NUMBER(2) NOT NULL,  
  ciudad      VARCHAR(10));
```

# Aspecto estructural

## Tablas



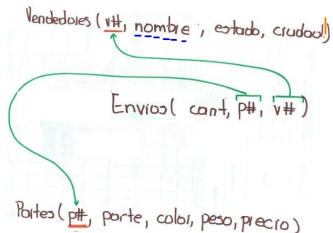
```
CREATE TABLE VENDEDORES(  
  v#          CHAR(2) NOT NULL,  
  nombre      VARCHAR(20) NOT NULL,  
  estado      NUMBER(2) NOT NULL,  
  ciudad      VARCHAR(10));
```

## Estándar de Codificación

- ▶ Indentación para creación de tablas
- ▶ Nombre de tablas en plural

# Aspecto estructural

## Tablas



```
CREATE TABLE VENDEDORES(  
  v#          CHAR(2) NOT NULL,  
  nombre      VARCHAR(20) NOT NULL,  
  estado      NUMBER(2) NOT NULL,  
  ciudad      VARCHAR(10));
```

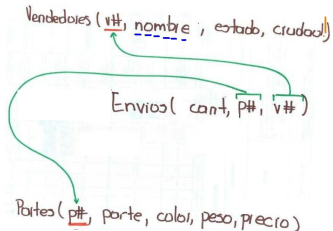
### Estándar de Codificación

- Indentación para creación de tablas
- Nombre de tablas en plural

- ¿Creación de la tabla PARTES?
- ¿Creación de la tabla ENVIOS?

# Aspecto integridad

## Claves candidatas



```
CREATE TABLE VENDEDORES{
```

```
  v#          CHAR(2) NOT NULL,
```

```
  nombre      VARCHAR(20) NOT NULL,
```

```
  estado      NUMBER(2) NOT NULL,
```

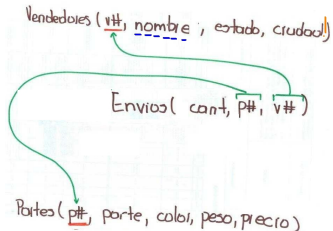
```
  ciudad      VARCHAR(10) );
```

```
ALTER TABLE VENDEDORES ADD CONSTRAINT PK_VENDEDORES  
PRIMARY KEY (v#);
```

```
ALTER TABLE VENDEDORES ADD CONSTRAINT UK_VENDEDORES_NOMBRE  
UNIQUE (nombre);
```

# Aspecto integridad

## Claves candidatas



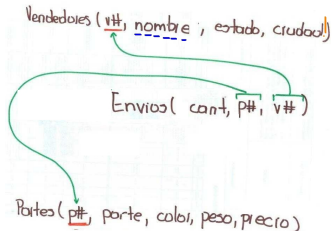
```
CREATE TABLE VENDEDORES(  
  v#          CHAR(2) NOT NULL,  
  nombre      VARCHAR(20) NOT NULL,  
  estado      NUMBER(2) NOT NULL,  
  ciudad      VARCHAR(10) );  
  
ALTER TABLE VENDEDORES ADD CONSTRAINT PK_VENDEDORES  
  PRIMARY KEY (v#);  
  
ALTER TABLE VENDEDORES ADD CONSTRAINT UK_VENDEDORES_NOMBRE  
  UNIQUE (nombre);
```

## Estándar de Codificación

- ▶ Indentación para creación de claves candidatas
- ▶ Nombre de claves  
*PK\_tabla, UK\_tabla\_atributo*

# Aspecto integridad

## Claves candidatas



```
CREATE TABLE VENDEDORES(  
  v#          CHAR(2) NOT NULL,  
  nombre      VARCHAR(20) NOT NULL,  
  estado      NUMBER(2) NOT NULL,  
  ciudad      VARCHAR(10) );  
  
ALTER TABLE VENDEDORES ADD CONSTRAINT PK_VENDEDORES  
  PRIMARY KEY (v#);  
  
ALTER TABLE VENDEDORES ADD CONSTRAINT UK_VENDEDORES_NOMBRE  
  UNIQUE (nombre);
```

## Estándar de Codificación

- ▶ Indentación para creación de claves candidatas
- ▶ Nombre de claves  
*PK\_tabla, UK\_tabla\_atributo*

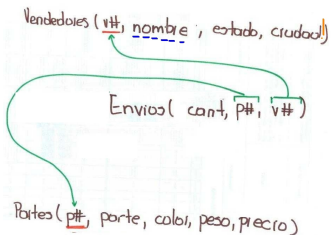
- ▶ ¿Creación de claves candidatas de la tabla de PARTES?
- ▶ ¿Creación de claves candidatas de la tabla de ENVIOS?

# Aspecto integridad

## Claves foraneas

```
CREATE TABLE ENVIOS(  
  v#          CHAR(2) NOT NULL,  
  p#          CHAR(2) NOT NULL,  
  cant        NUMBER(5) NOT NULL);
```

```
ALTER TABLE ENVIOS ADD CONSTRAINT FK_ENVIOS_PARTES  
  FOREIGN KEY (p#) REFERENCES PARTES(p#);
```

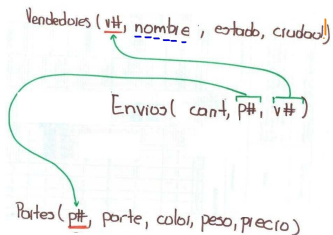


► ¿Qué falta en la tabla ENVIOS?



# Aspecto integridad

## Claves foraneas



```
CREATE TABLE ENVIOS(  
  v#          CHAR(2) NOT NULL,  
  p#          CHAR(2) NOT NULL,  
  cant        NUMBER(5) NOT NULL);
```

```
ALTER TABLE ENVIOS ADD CONSTRAINT FK_ENVIOS_PARTES  
  FOREIGN KEY (p#) REFERENCES PARTES(p#);
```

## Estándar de Codificación

- ▶ Indentación para creación de clave foráneas
- ▶ Nombre de clave foránea

*FK\_tablaBase\_tablaReferenciada*

- ▶ ¿Qué falta en la tabla ENVIOS?