

BASE DE DATOS

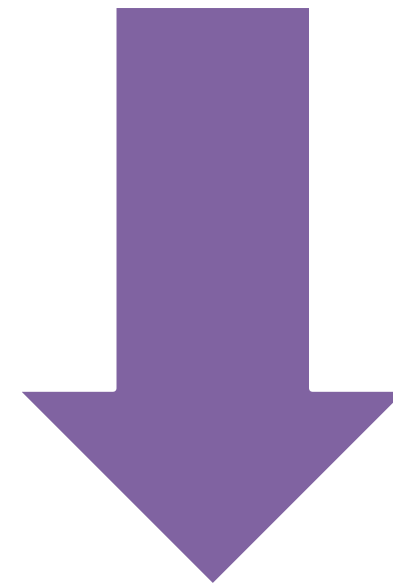
2

CONCEPTO

*Una base de datos es
una colección de datos relacionados.*



Representa algún aspecto
del mundo real



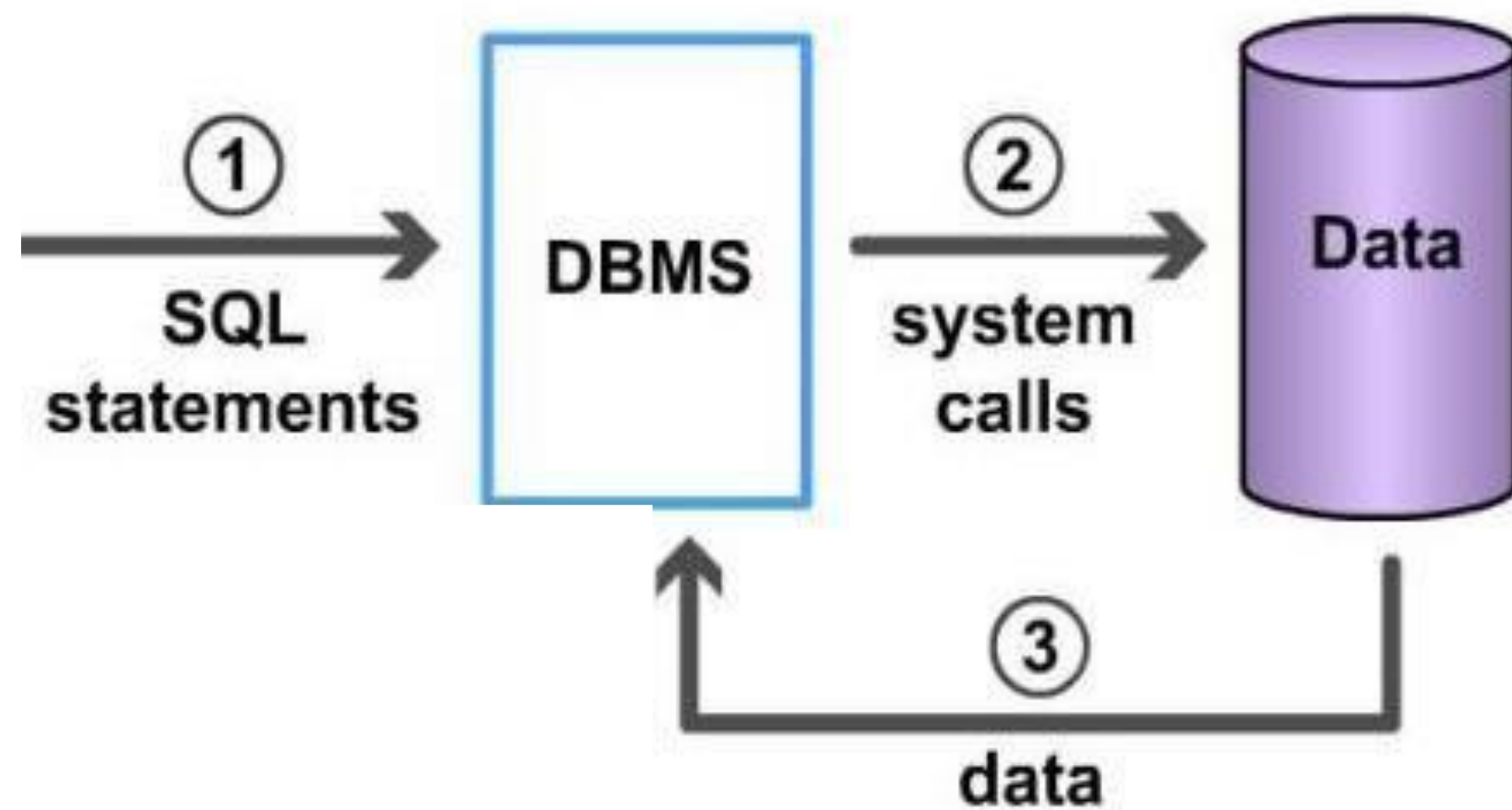
Es una colección de
datos lógicamente
coherente



Tiene un propósito
específico.

DBMS

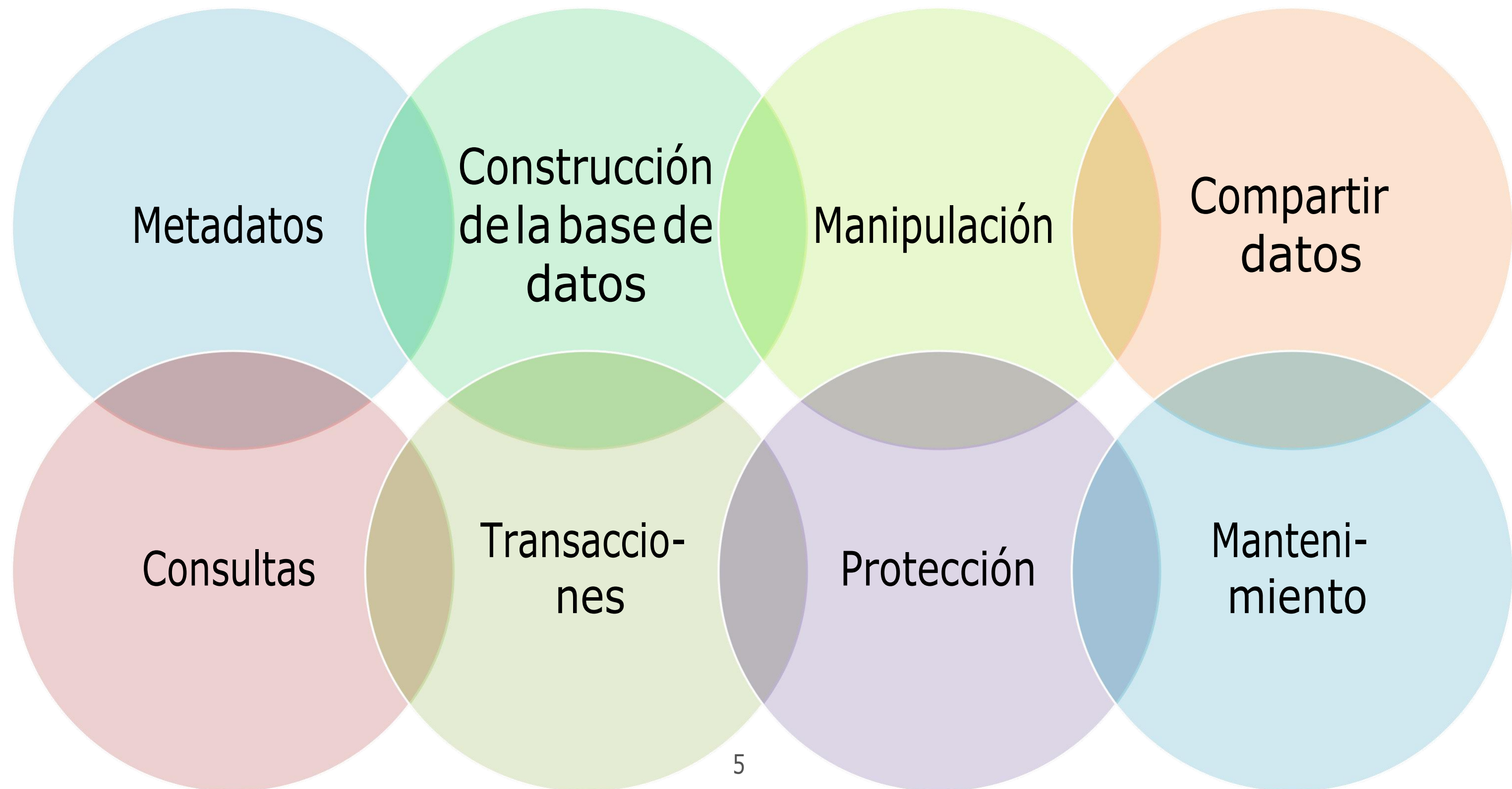
Data Base Management System
Sistema de Administración de Base de Datos



- ✓ Permiten definir la estructuras de datos a almacenar.
- ✓ Proveen de mecanismos de manipulación de datos.
- ✓ Se encargan de garantizar la fiabilidad de la información almacenada.
- ✓ Manejan la concurrencia de usuarios accediendo a una misma base de datos

DBMS

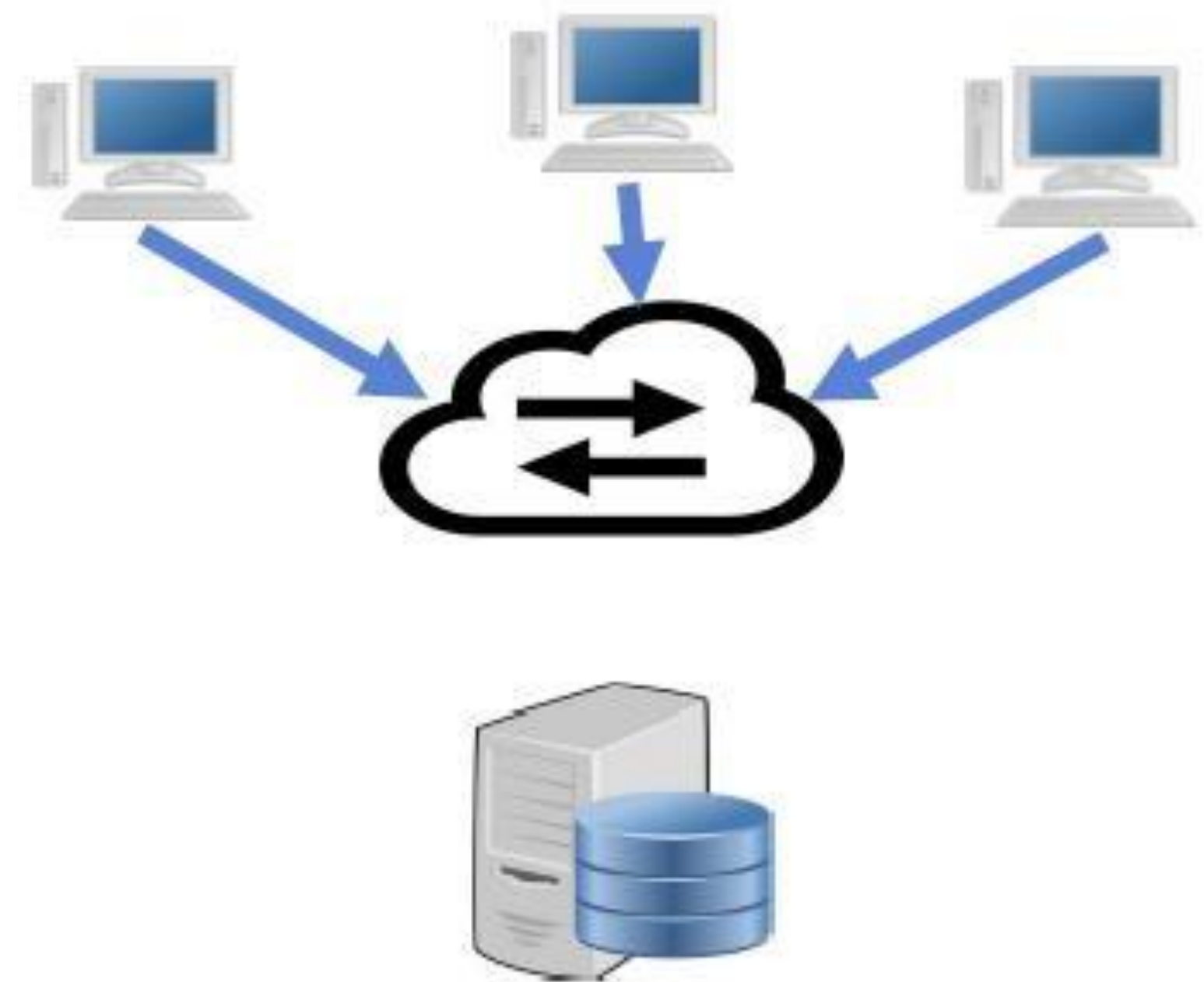
Funciones



Sistemas de Base de Datos Centralizados y Distribuidos



Servidor de BD instalado localmente en la computadora



Servidor de BD externo
Comunicación TCP/IP

Clasificación de Bases de Datos

MODELO DE DATOS

- Modelo de datos: es una serie de conceptos que sirven para describir un conjunto de datos y las operaciones para manipularlos.
- Tipos de modelos de datos:
 - Modelos conceptuales
 - Modelos lógicos.



- Expresividad
- Simplicidad
- Minimalidad
- Formalidad

ESQUEMA

- Un esquema es un conjunto de representaciones lingüísticas o gráficas que describen la estructura de los datos de interés.
- Se utilizan modelos de datos para describir un esquema.

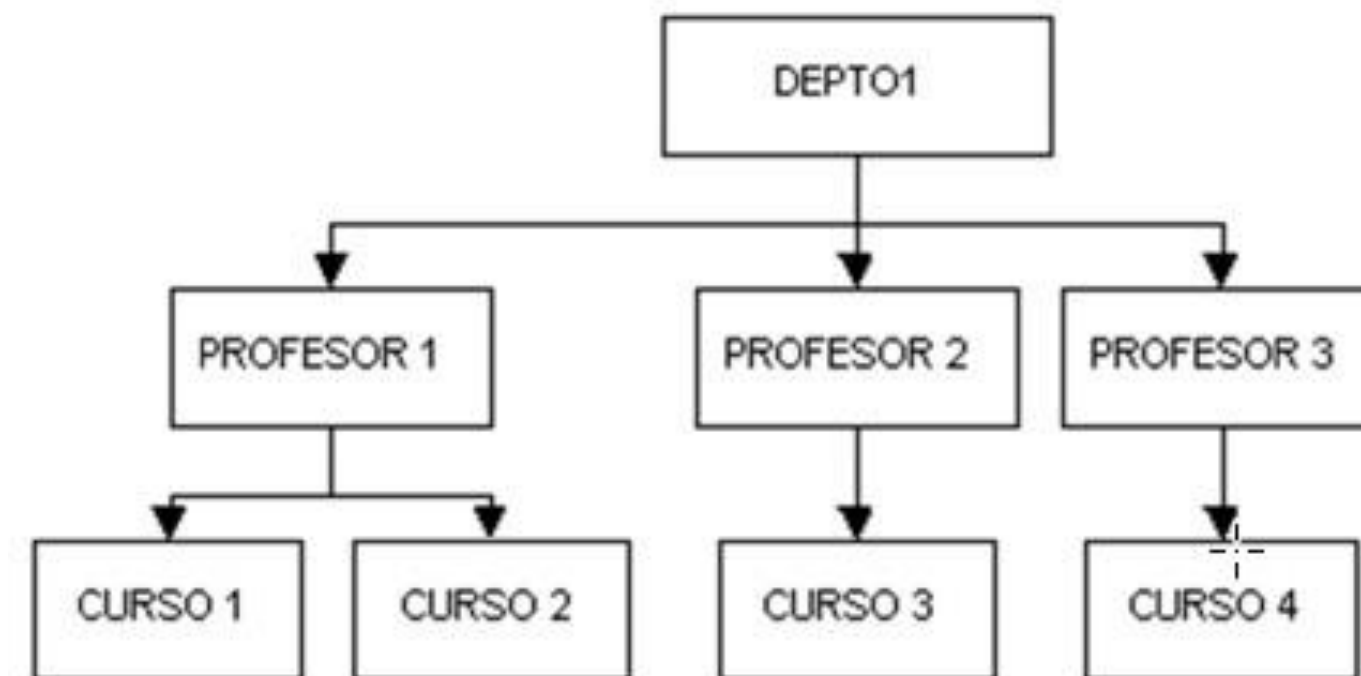
Clasificación de Bases de Datos

Cronológicamente los modelos de datos fueron:

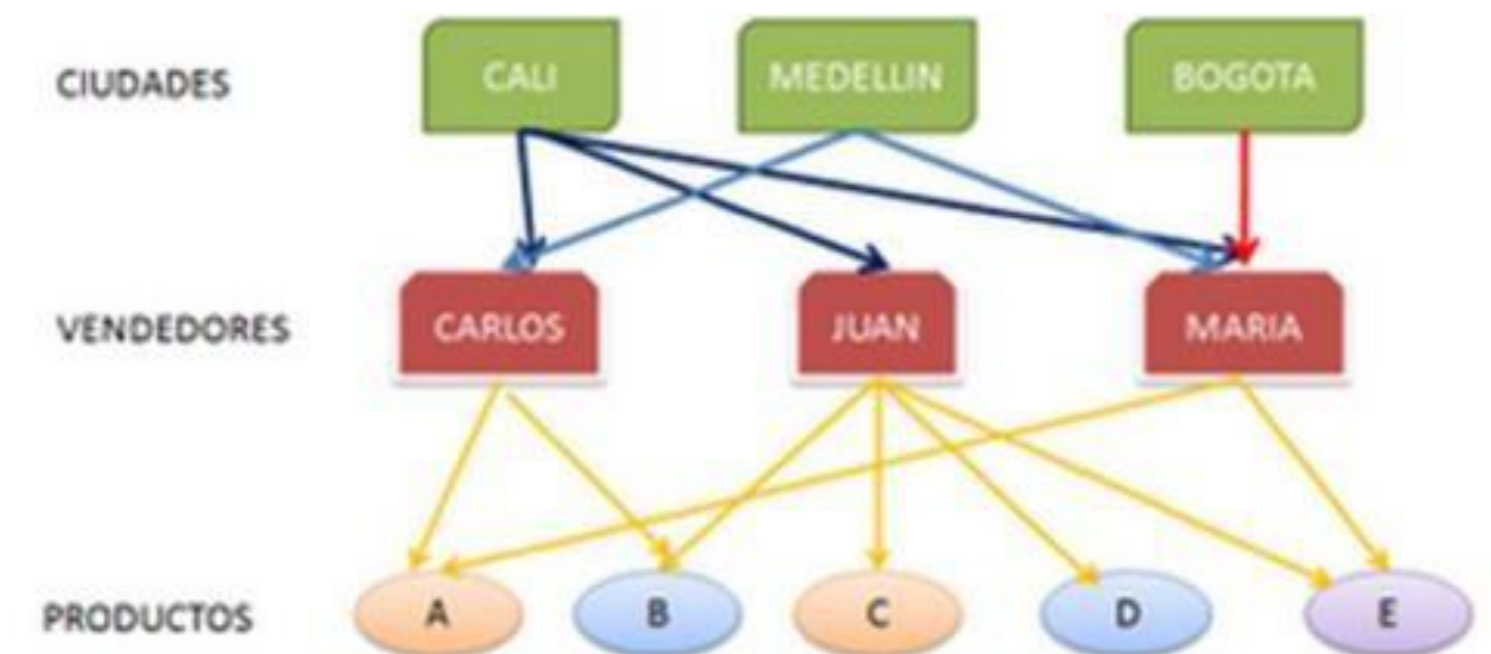
- Basados en sistemas de archivos convencionales



- Jerárquico

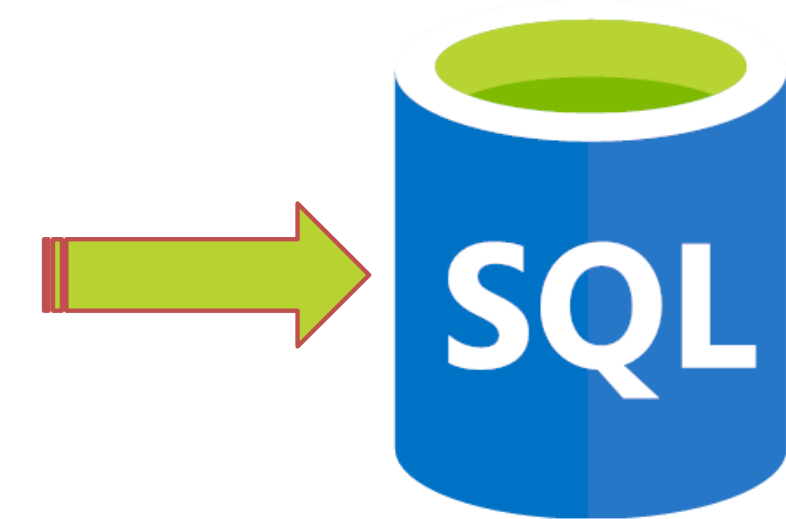
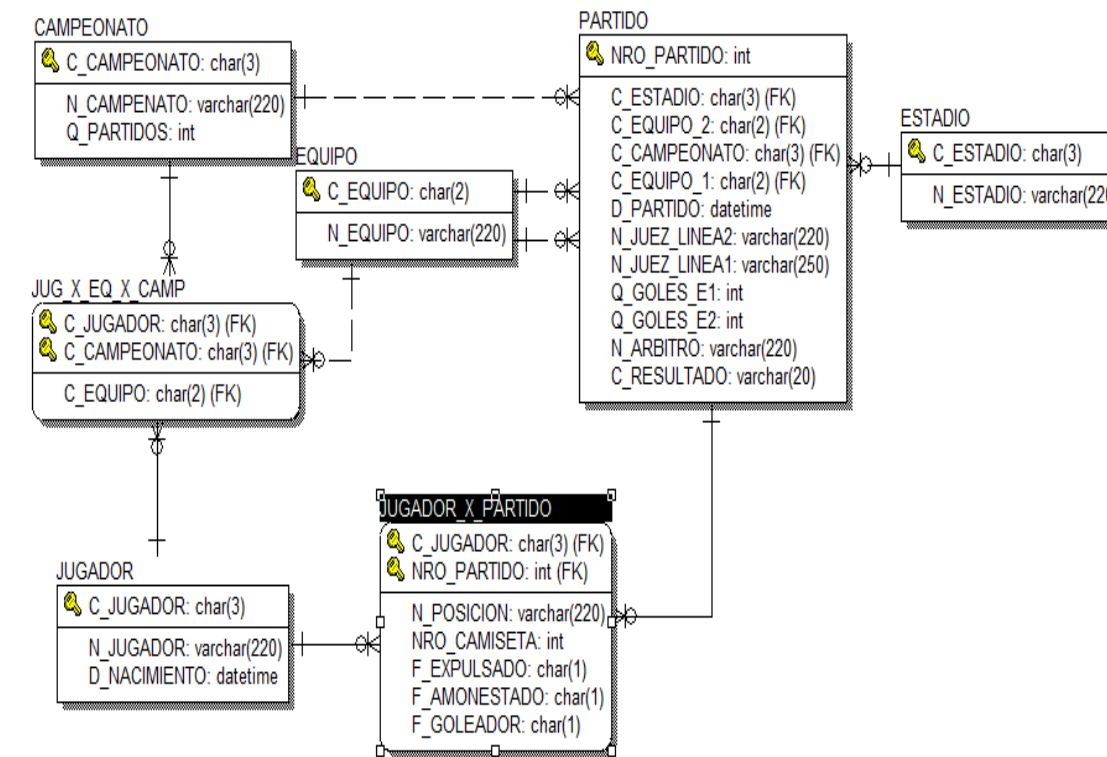


- De Red

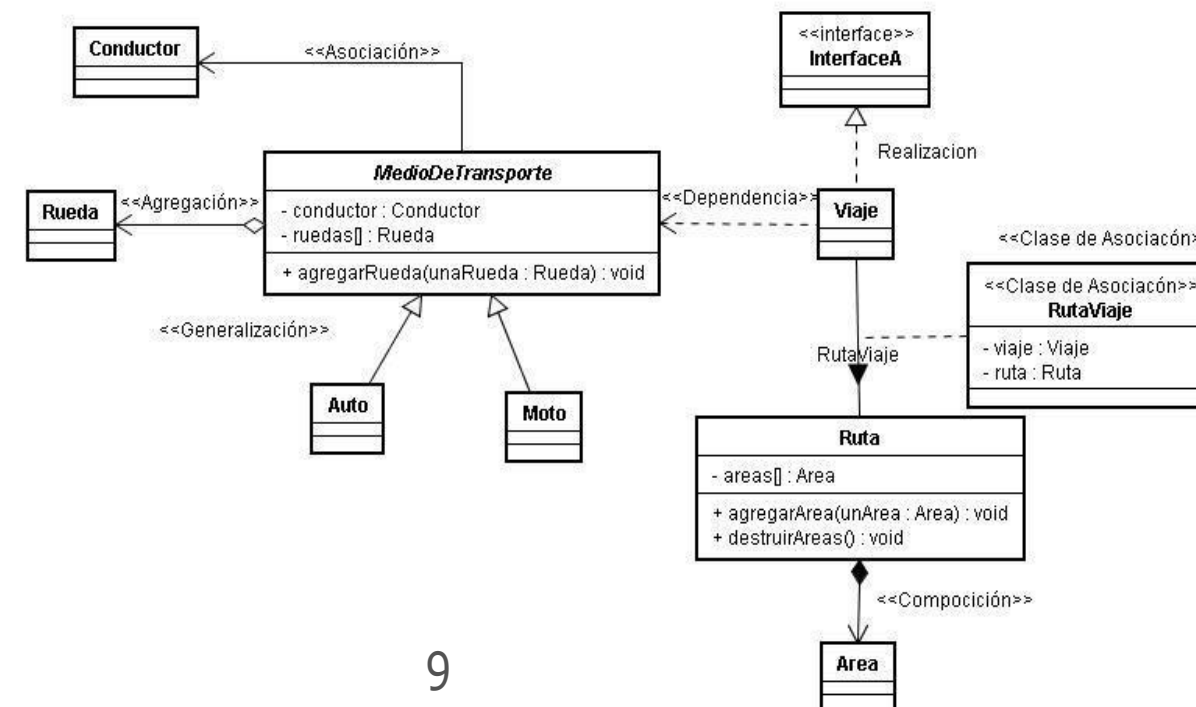


Clasificación de Bases de Datos

- Entidad-Relación

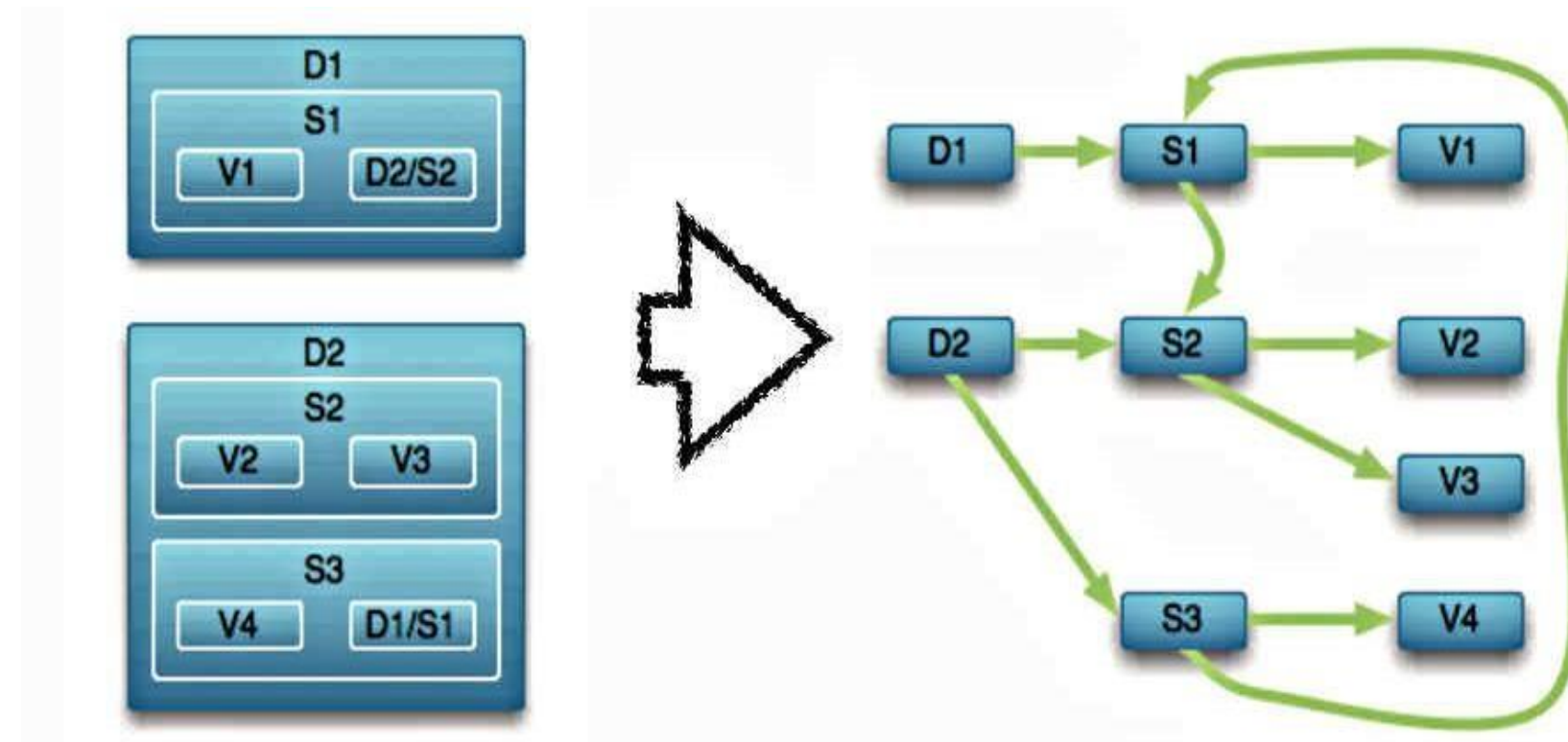


- Orientado a Objetos

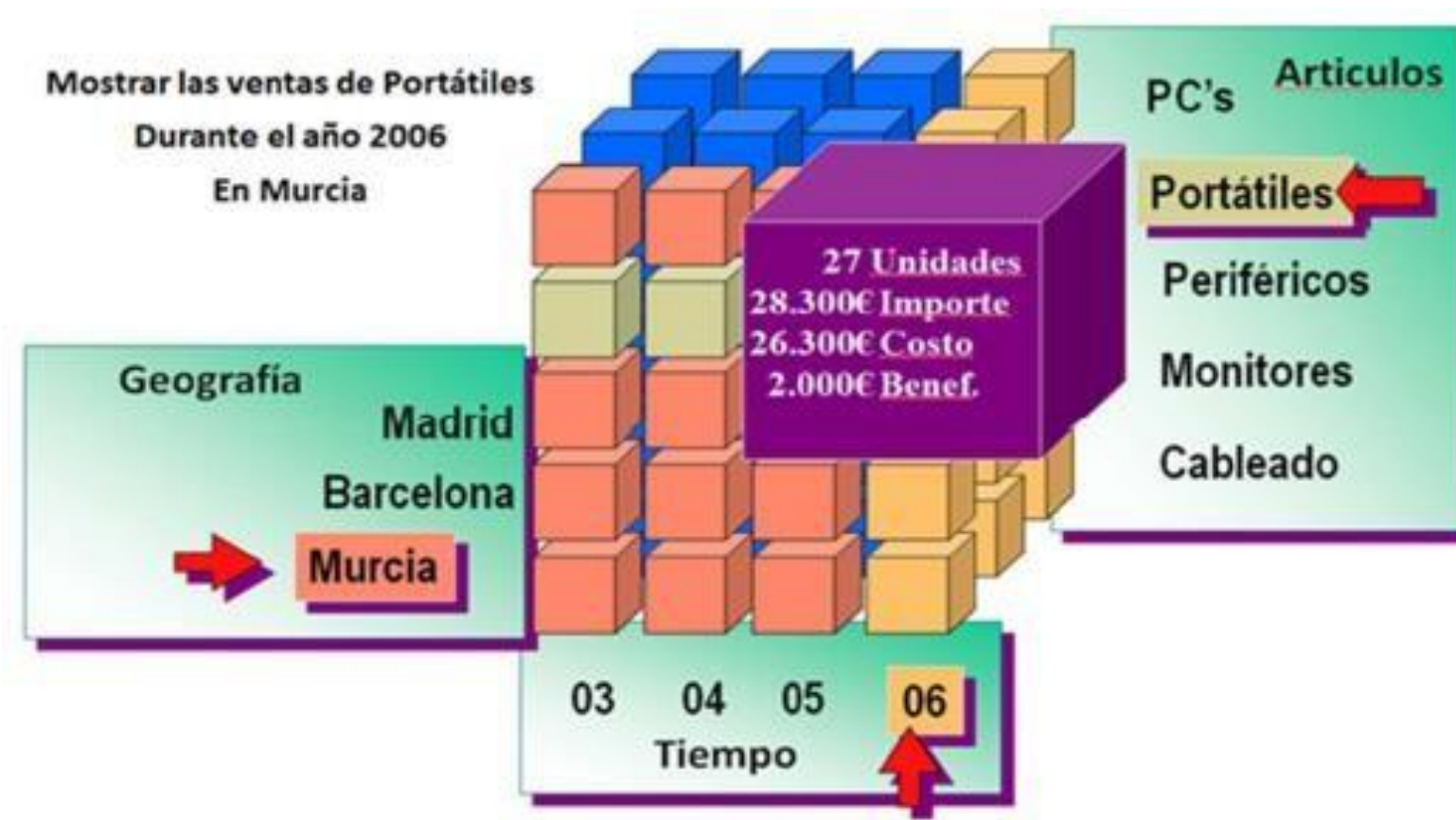


Clasificación de Bases de Datos

- Orientado a Documentos



- Multidimensional



Modelo Entidad Relación

- Caso de Estudio

“Sistema de Gestión de Ventas y Reservas de Entradas para un Complejo de Cines”

1. Un complejo de cines está integrado por varios cines.
2. Cada cine cuenta con una cantidad de salas
3. La programación de las funciones de las películas de las salas se renueva en forma semanal.
4. No todas las salas tienen la misma capacidad.
5. La programación es la que determina qué películas van a proyectarse y los horarios para cada función de cada una de las salas, para todos los cines.
6. Una entrada es válida únicamente para la fecha, hora y función indicadas en la misma.



Modelo Entidad Relación

ENTIDAD

- ✓ Es el **elemento básico** representado por el modelo entidad relación.
- ✓ Representa una **cosa del mundo real**
- ✓ Puede tener **existencia física o existencia conceptual**.
- ✓ Cada ejemplo de la entidad se denomina **instancia**.
- ✓ *Cosas tangibles: Artículo, Repuesto, Rodado.*
- ✓ *Roles desempeñados por personas u organizaciones: Cliente, Proveedor.*
- ✓ *Incidentes: Siniestros, Anulación de Venta*
- ✓ *Interacciones: Representan alguna transacción: Compra, Pedido, Venta*

Ejemplos de ENTIDAD

RESPONDER

¿Cuáles son las entidades del caso de estudio?

Modelo Entidad Relación

ATRIBUTOS

- Una entidad tiene atributos.
- Un atributo es una abstracción que identifica características, propiedades de una entidad.
- Los atributos deben ser: completos, plenamente elaborados, mutuamente independientes



Ejemplo Entidad: Aula

Atributos:

numero: 1, piso: 2, ancho: 4 mts,
largo: 6 mts, bancos:25

RESPONDER

¿Cuáles son los atributos de las entidades identificadas?

Modelo Entidad Relación

CLASIFICACIÓN ATRIBUTOS

- Identificadores
- Descriptivos
- Referenciales

IDENTIFICADOR UNICO

- Uno o más atributos que identifican unívocamente cada instancia de una entidad.
- Un atributo, es identificador único si posee



Unicidad: en cualquier momento dado no existen 2 instancias con el mismo valor de A.

Minimidad: Si A es compuesto, no será posible eliminar ningún componente de A sin destruir la propiedad de unicidad.

Modelo Entidad Relación

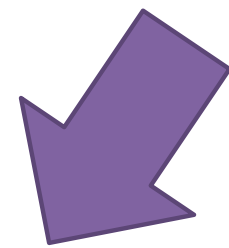
IDENTIFICADOR
UNICO

Pelicula
@id_pelicula
anio_estreno
disponible
duracion
fecha_ingreso
nombre
titulo_original

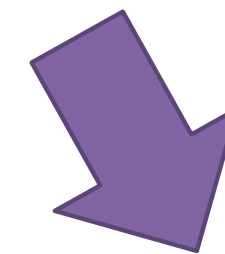
Modelo Entidad Relación

ATRIBUTO REFERENCIAL

- ✓ Un atributo referencial se utiliza para poder establecer **relaciones** entre diferentes entidades.
- ✓ Se dice que un atributo j , ó un conjunto de atributos, de una entidad B es un atributo referencial si y solo si satisface dos propiedades:



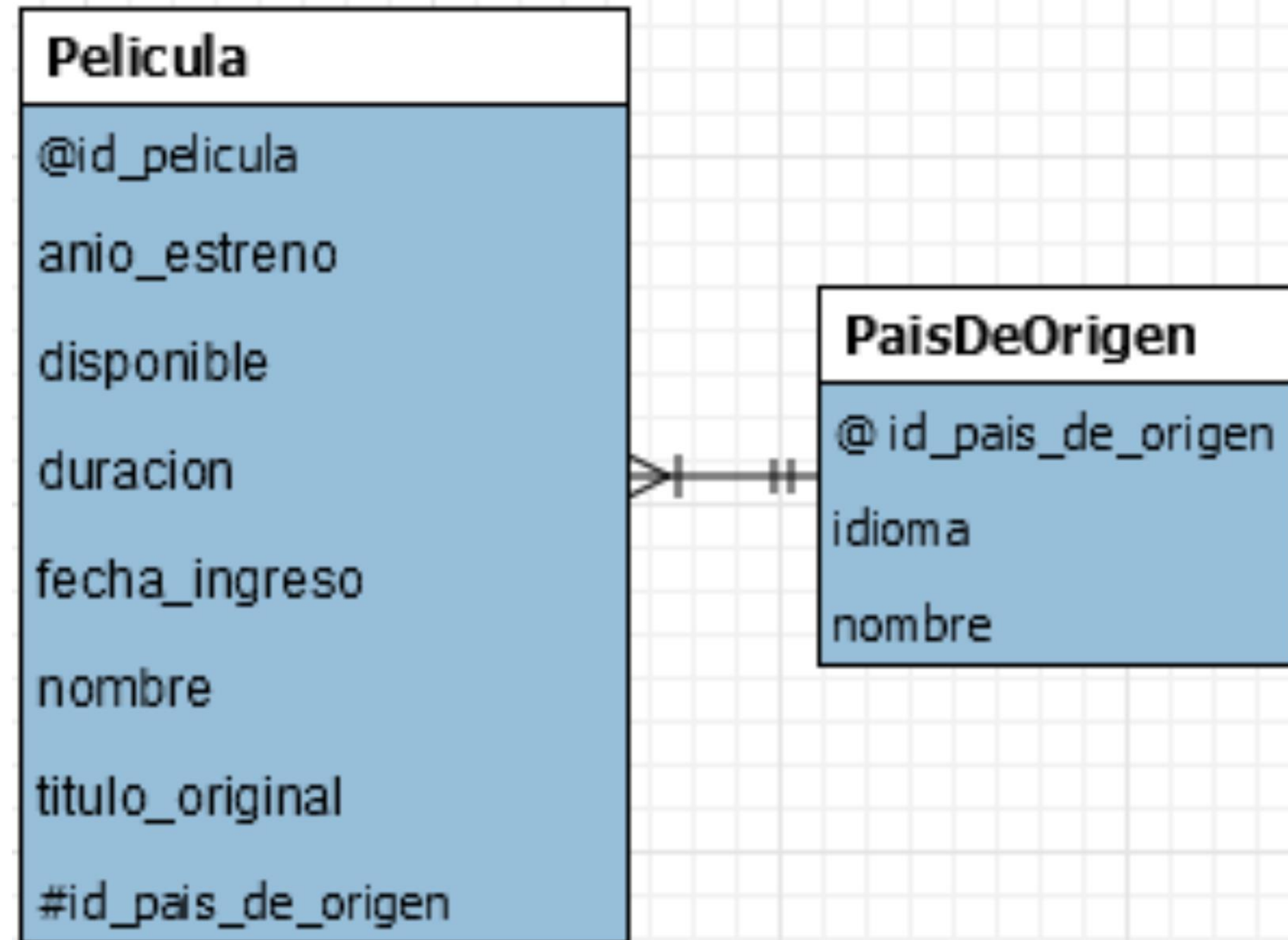
Cada valor j es nulo del todo o no nulo del todo. – En caso de ser un atributo compuesto, formado por más de un atributo.



Existe una entidad A con atributo identificador j tal que:
Cada valor no nulo de j es en la entidad B idéntico al valor j en alguna instancia de la entidad A .

Modelo Entidad Relación

ATRIBUTO
REFERENCIAL



Modelo Entidad Relación

REGLAS DE INTEGRIDAD

1. Integridad de Entidades: Ningún componente del atributo identificador en una entidad aceptará NULOS



Película: {**NULL**, 2014, true, 122, 01/08/2014, "Relatos Salvajes", "Relatos Salvajes", 1}



Película: {**2**, 2014, true, 122, 01/08/2014, "Relatos Salvajes", "Relatos Salvajes", 1}

Modelo Entidad Relación

REGLAS DE INTEGRIDAD

2. Integridad Referencial: Un modelo de datos no debe contener valores en sus atributos referenciales para los cuales no exista un valor concordante en el (ó los) atributos identificadores en la entidad objetivo.



Película: { 2, 2014, true, 122, 01/08/2014, "Relatos Salvajes", "Relatos Salvajes", 4 }

No existe en la tabla PaisDeOrigen una instancia con id_pais_de_origen = 4.



Película: { 2, 2014, true, 122, 01/08/2014, "Relatos Salvajes", "Relatos Salvajes", 1 }

Modelo Entidad Relación

RELACIONES

- Una relación es la abstracción de un conjunto de asociaciones que existen entre las instancias de dos entidades.
- Tienen sentido bidireccional.

CARDINALIDAD

- Indica para una instancia de una entidad A con cuántas instancias de la entidad B se relaciona.

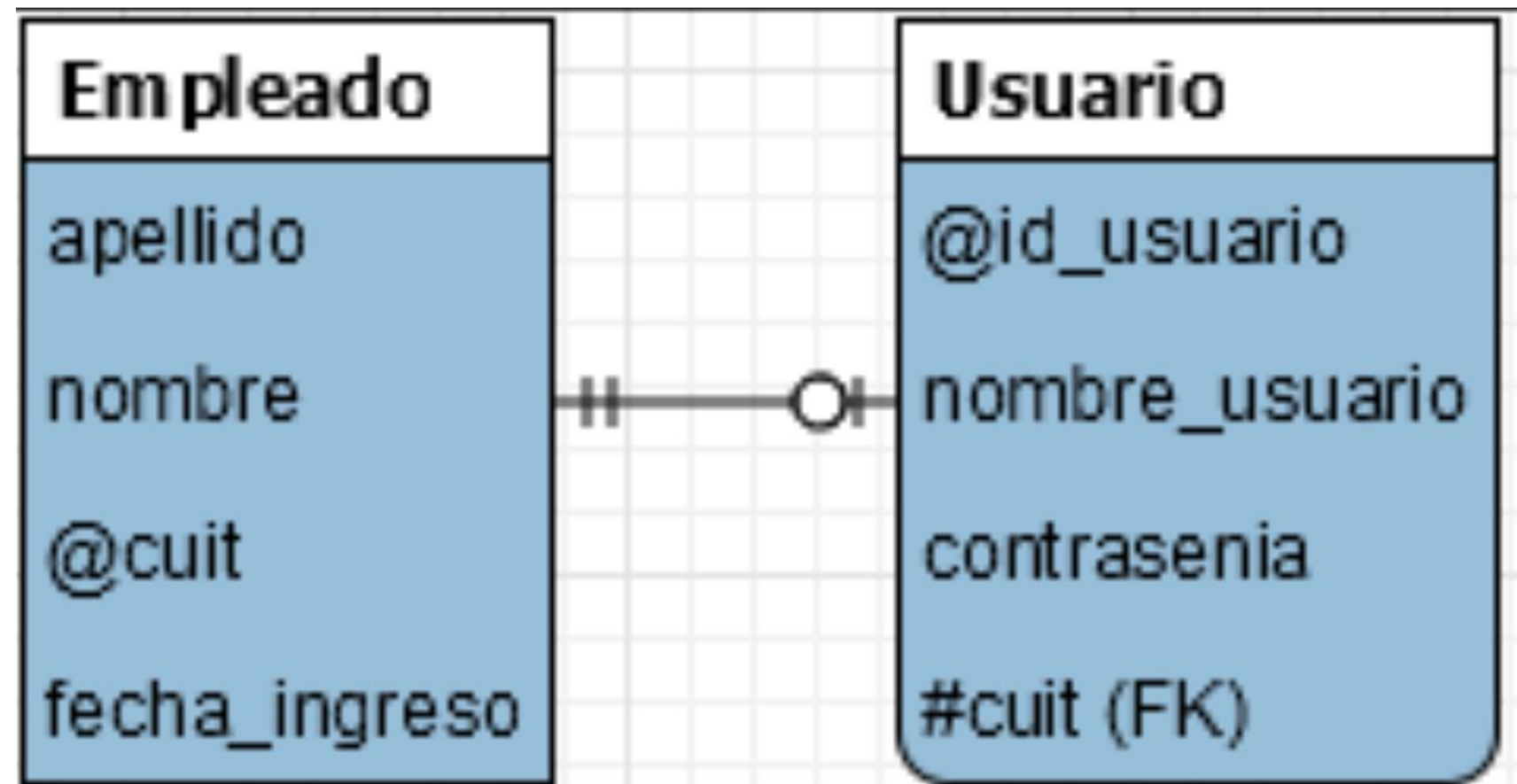
OPCIONALIDAD

- Indica para una instancia de una entidad A, si la relación con instancias de la entidad B, es opcional u obligatoria.

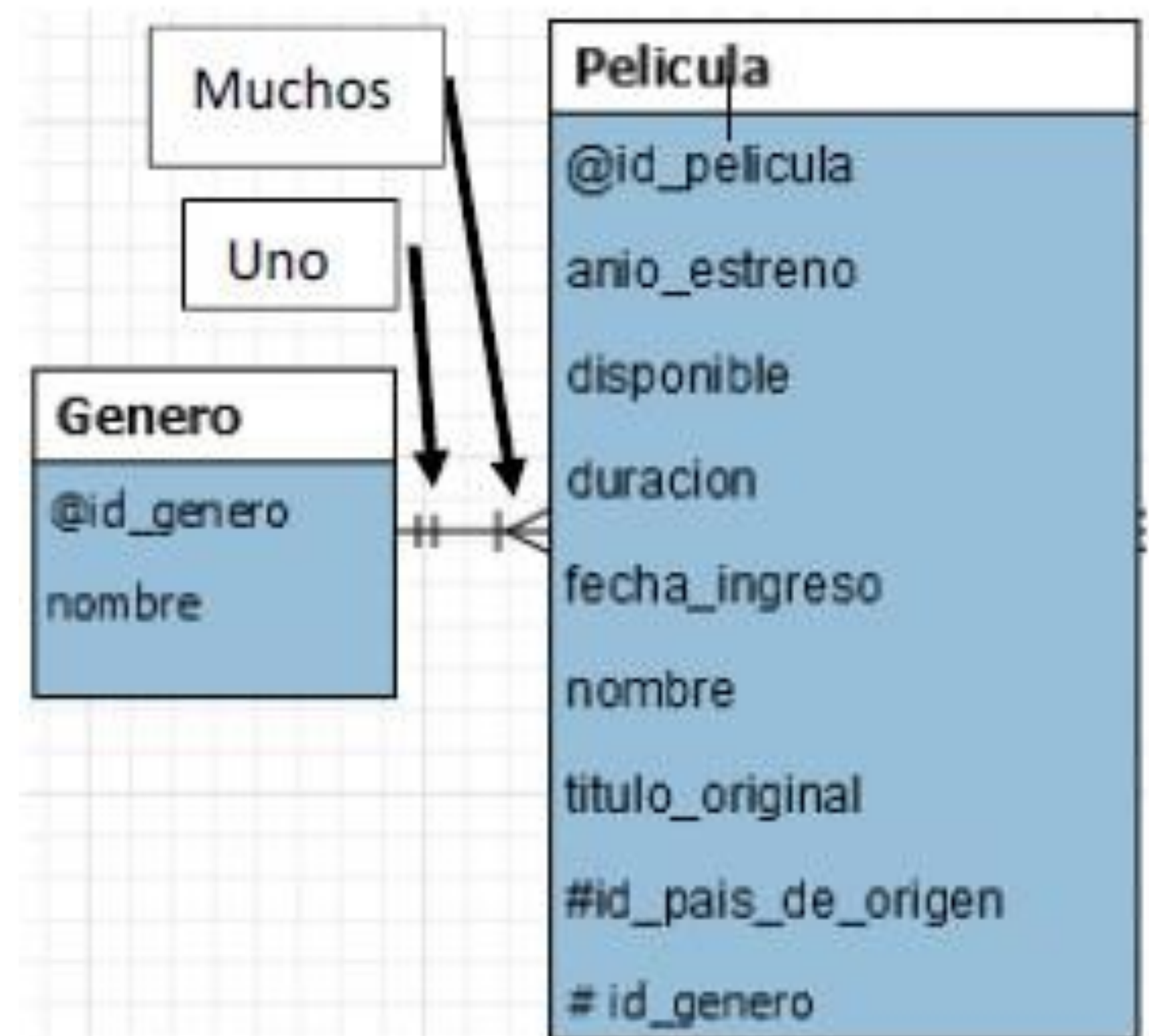
Modelo Entidad Relación

OPCIONALIDAD

Uno a uno



Uno a Muchos

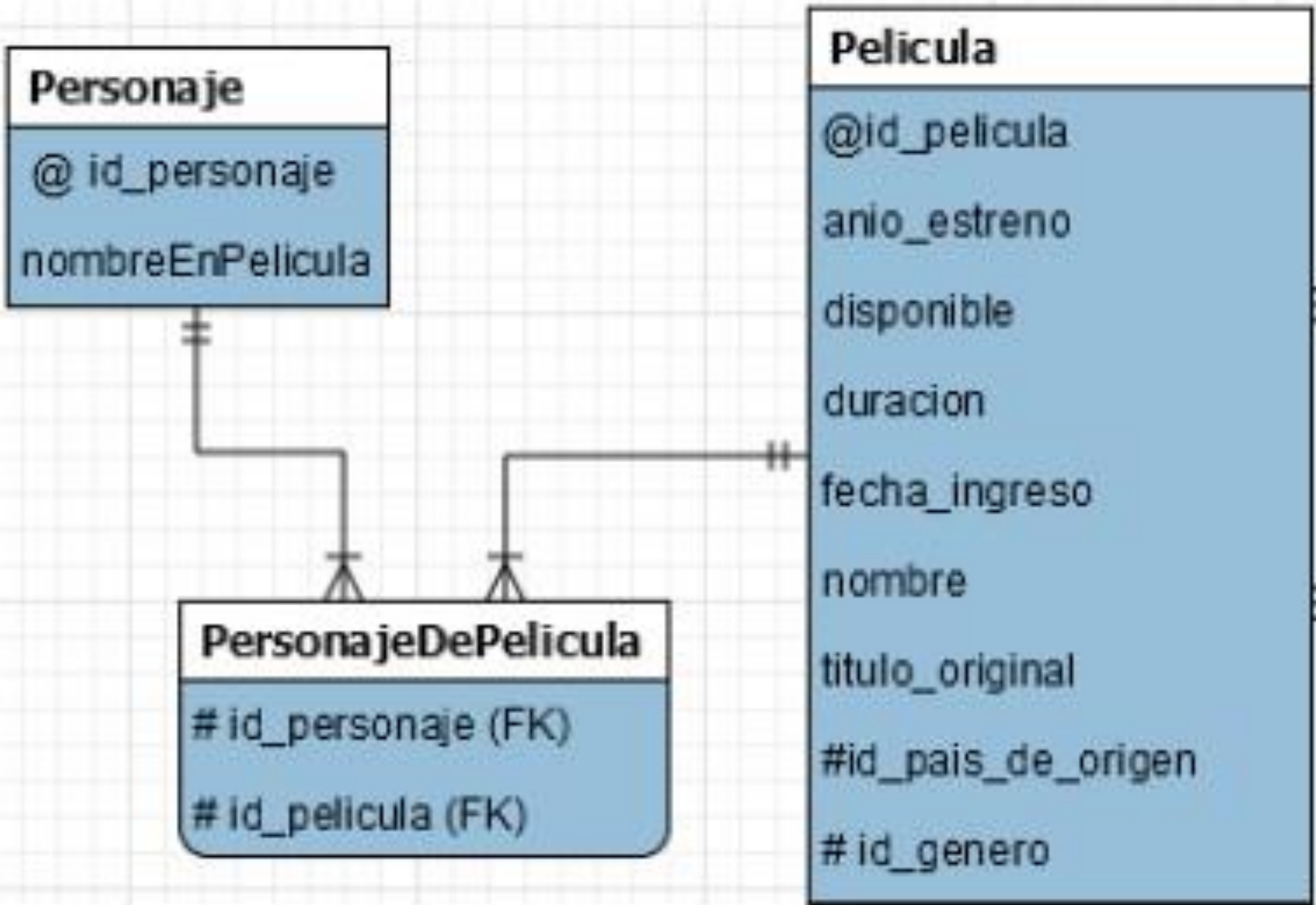
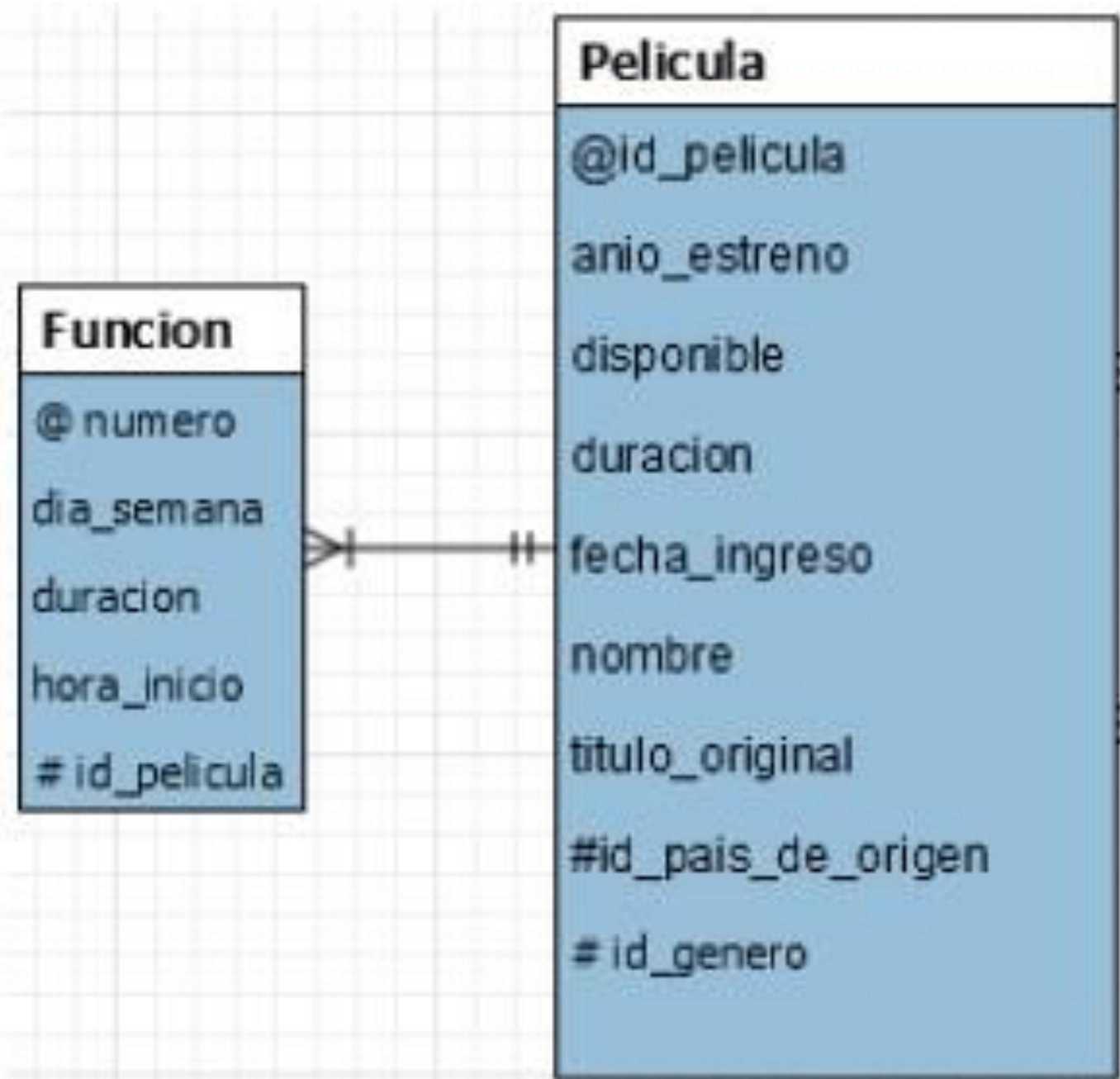


Modelo Entidad Relación

OPCIONALIDAD

Muchos a muchos

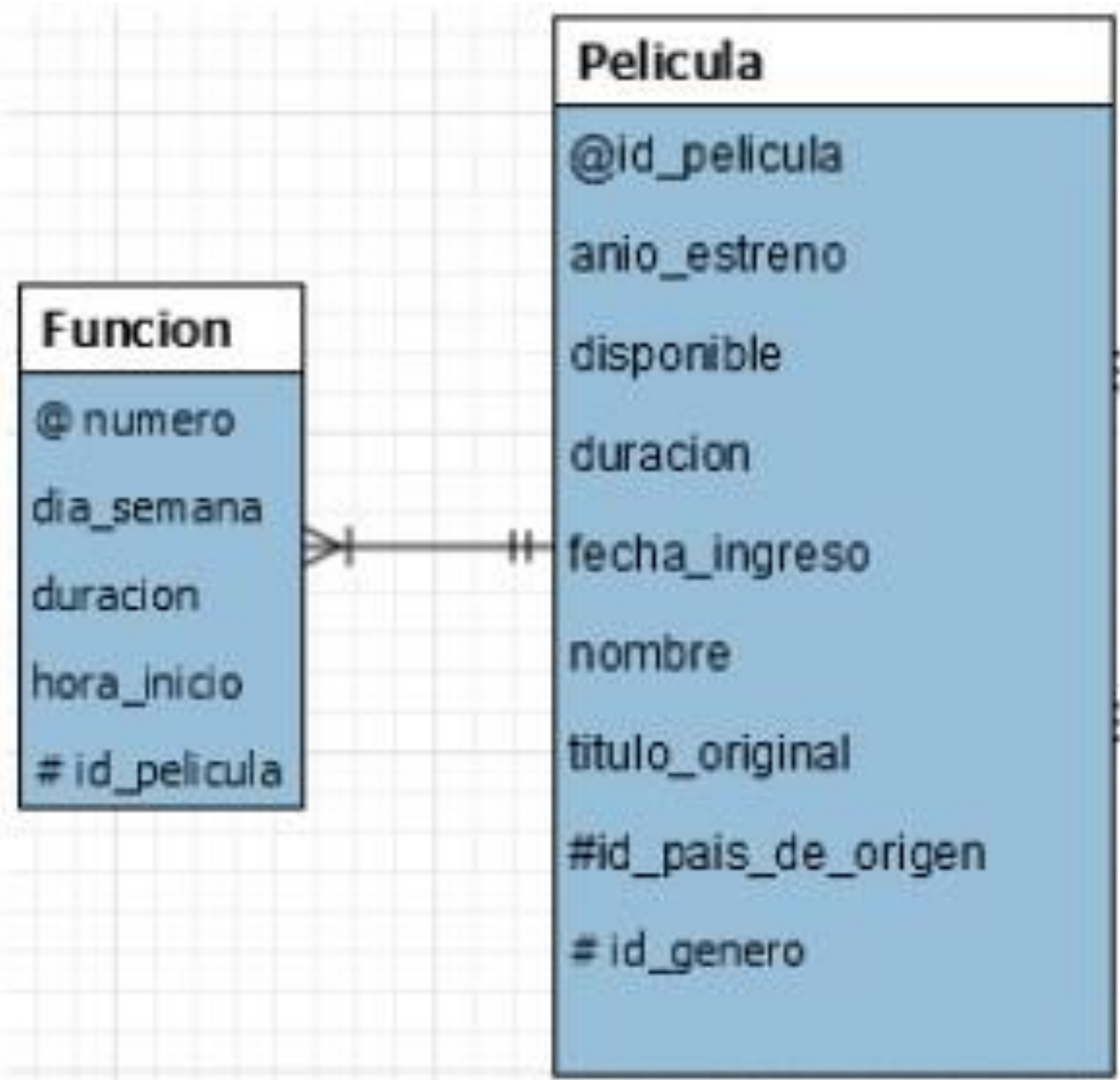
Muchos a Uno



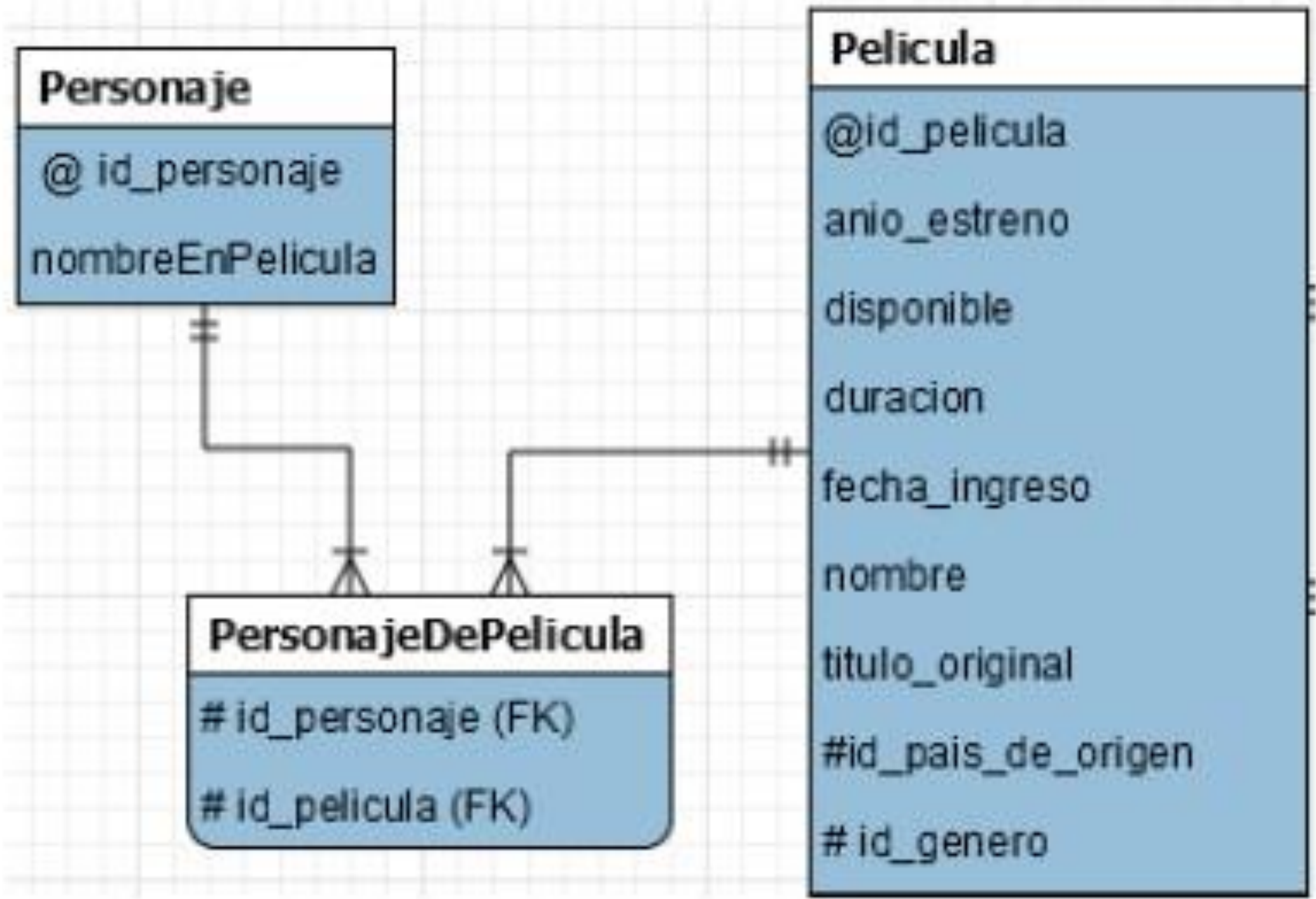
Modelo Entidad Relación

OPCIONALIDAD

Muchos a Uno

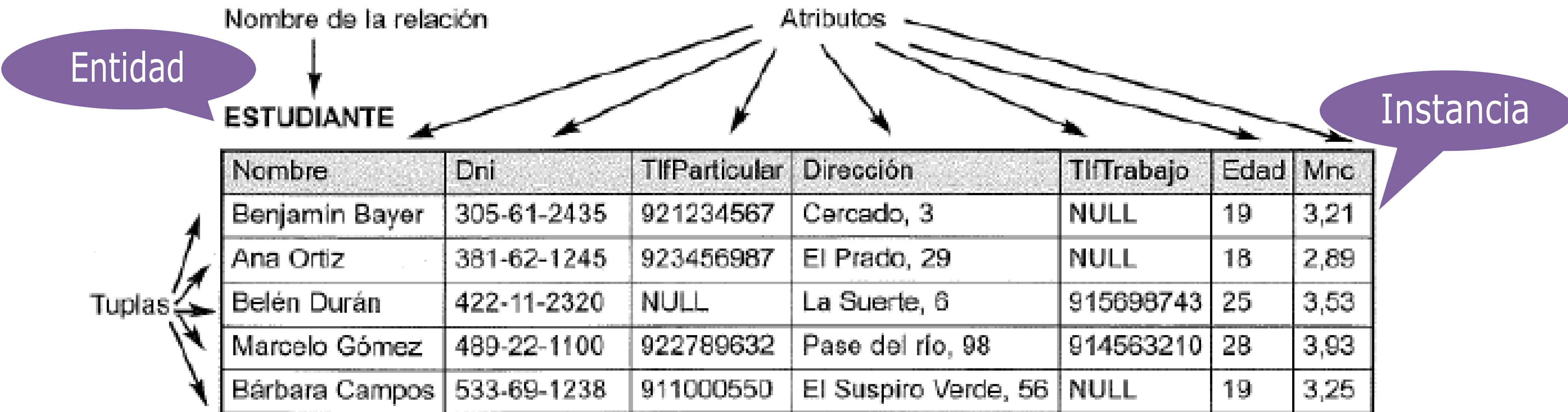


Muchos a muchos

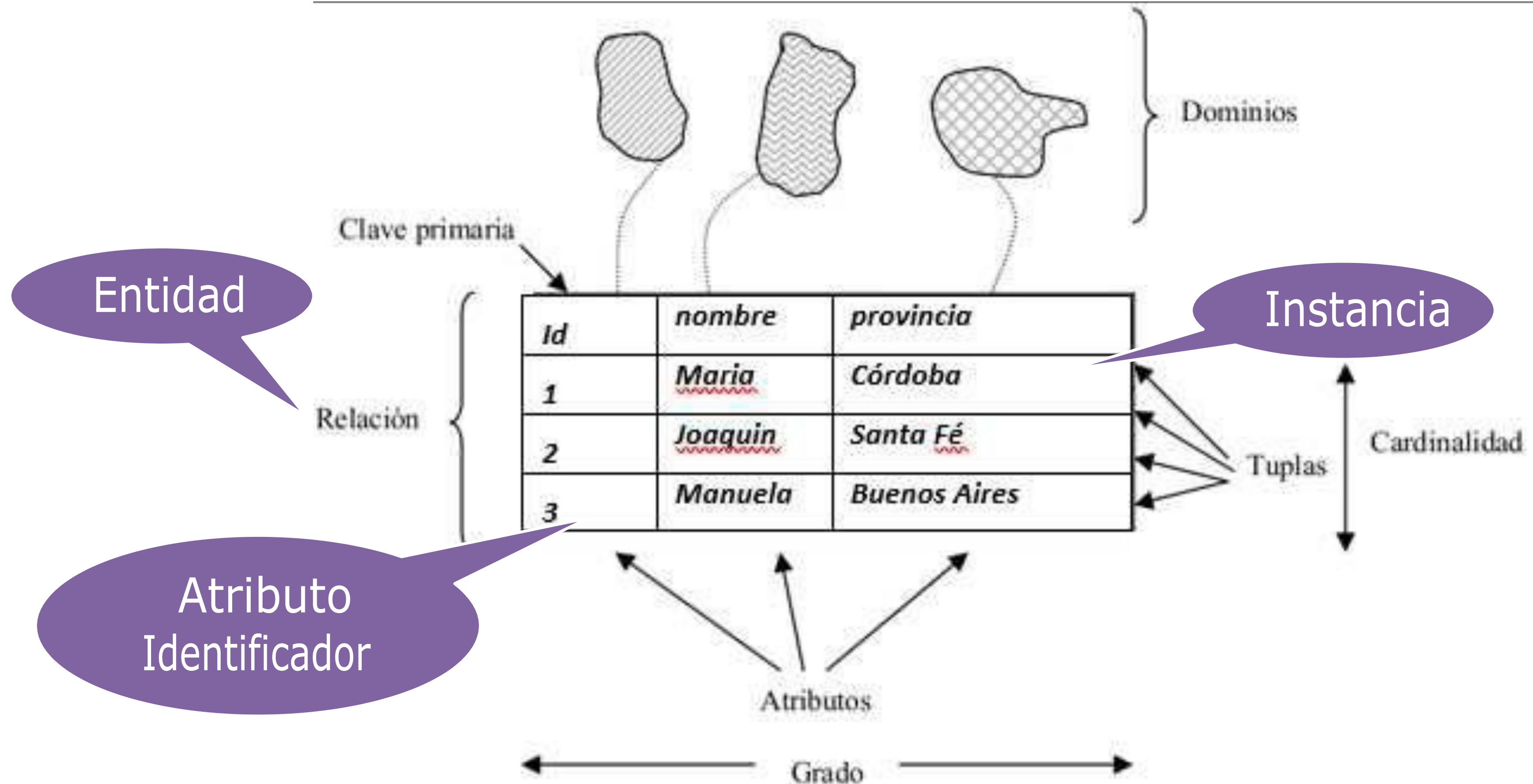


Base de Datos Relacional

- El modelo relacional representa la base de datos como una **colección de relaciones**.
- Cada una de estas relaciones se parece a una **tabla de valores**.



Base de Datos Relacional



Base de Datos Relacional

Entidad

- Una relación corresponde a lo que conocemos como tabla.
- Un atributo corresponde a una columna o campo. El número de atributos se denomina grado.
- Una tupla corresponde a una fila o registro de esa tabla.
- El número de tuplas de una tabla se denomina cardinalidad.
- La clave primaria es un identificador único para la tabla, es decir una columna o combinación de columnas con la siguiente propiedad:

Instancia

"Nunca existen dos filas de la tabla con el mismo valor en esa columna o combinación de columnas."

Atributo
Identificador

- Un dominio es una colección de valores, de los cuales uno o más atributos (columnas) obtienen sus valores reales.

RELACIONES

Base de Datos Relacional

Propiedades de las Relaciones:

- No existen tuplas repetidas.
- Las tuplas no tienen que estar ordenadas.
- Los atributos no tienen que estar ordenados.
- Todos los valores de los atributos son atómicos

Base de Datos Relacional

REGLA DE INTEGRIDAD Clave Primaria

Un atributo a (posiblemente compuesto) de la relación R es una clave candidata de R , sí y solo sí satisface las siguientes propiedades:

- Unicidad: no existen dos tuplas de R con el mismo valor de a , en un momento dado.
- Minimalidad: si a es un atributo compuesto no puedo eliminar un componente de a si destruir la propiedad de unicidad.

<i>id_cliente</i>	<i>nombre_cliente</i>	<i>calle_cliente</i>	<i>ciudad_cliente</i>
19.283.746	González	Arenal, 12	La Granja
67.789.901	López	Mayor, 3	Peguerinos
18.273.609	Abril	Preciados, 123	Valsaín
32.112.312	Santos	Mayor, 100	Peguerinos
33.666.999	Rupérez	Ramblas, 175	León
01.928.374	Gómez	Carretas, 72	Cerceda

REGLA DE INTEGRIDAD Clave Primaria

Base de Datos Relacional

- Toda relación tendrá por fuerza una clave primaria.
- La clave primaria es la que tiene verdadera importancia; las demás, claves candidatas y alternativas, son sólo conceptos que surgen en el proceso de decisión.
- Al nombrar al atributo identificador, es conveniente hacerlo como clave primaria, porque hay muchos tipos de claves.

¿Por qué son importantes las Claves Primarias?

- Son importantes porque constituyen el mecanismo de direccionamiento a nivel de tuplas, básico en un sistema relacional.
- Es el único modo garantizado por el sistema para localizar una tupla específica.
- Ningún componente de la clave primaria de una relación base puede aceptar nulos.

Base de Datos Relacional

REGLA DE INTEGRIDAD Relaciones

En una base de datos relacional, nunca registraremos información de algo que no podamos identificar.

- Para las claves primarias compuestas: cada valor individual de la clave primaria debe ser no nulo en su totalidad.
- Esta regla se aplica a las relaciones base únicamente.
- Se aplica sólo a las claves primarias.

REGLA DE INTEGRIDAD Claves Ajenas

La clave ajena (FOREIGN KEY, en inglés) es un atributo (quizás compuesto) de una relación R2 cuyos valores deben concordar con los de la clave primaria de alguna otra relación R1.

Motores de Bases de Datos Relacionales

Es el servicio que permite interactuar con la base de datos, permitiendo almacenar los datos, procesarlos y recuperarlos.

The Oracle logo, featuring the word "ORACLE" in a red, serif, all-caps font.

Oracle

PostgreSQL



PostgreSQL



MySQL



SQL Server

Lenguaje de Consulta SQL

Structured Query Language Lenguaje de Consulta Estructurado

SQL brinda diferentes sentencias para definir datos, realizar consultas y actualizaciones.

Lenguaje de definición de datos: DDL (Data Definition Language). Permiten crear y definir nuevas bases de datos, campos e índices

Lenguaje de manipulación de datos: DML (Data Manipulation Language). Permiten generar consultas para ordenar, filtrar y extraer datos de la base de datos

Lenguaje de Consulta SQL

DDL

- ✓ CREATE: Crea nuevas tablas, campos e índices.
- ✓ DROP: Elimina tablas e índices.
- ✓ ALTER: Modifica las tablas agregando campos o cambiando la definición de los campos.

DML

- ✓ SELECT: Consulta registros de la base de datos que satisfagan un criterio determinado.
- ✓ INSERT: Carga lotes de datos en la base de datos en una única operación.
- ✓ UPDATE: Modifica los valores de los campos y registros especificados.
- ✓ DELETE: Elimina registros de una tabla de una base de datos.

Lenguaje de Consulta SQL

DDL

Create Database

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] nombre_base_datos
```

Create Table

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] nombre_de_tabla  
nombre_de_columna tipo_de_dato [NOT NULL | NULL] [DEFAULT  
valor_por_defecto]  
[AUTO_INCREMENT] [UNIQUE | [PRIMARY] KEY  
[CONSTRAINT nombre_relación FOREIGN KEY (nombre_columna)  
REFERENCES nombre_de_tabla (nombre_columna)]  
[ON DELETE opciones_de_referencia]  
[ON UPDATE opciones_de_referencia]
```

Create Index

```
CREATE INDEX nombre_indice  
ON nombre_tabla (col_name [(length)] [ASC |  
DESC])  
[opcion_algoritmo | opcion_bloqueo] ...
```

Lenguaje de Consulta SQL

DDL

Alter Table

```
ALTER [IGNORE] TABLE nombre_tabla
ADD [COLUMN] nombre_columna columna_definicion
[FIRST | AFTER col_nombre]
| ADD [COLUMN] (nombre_columna definición_columna)
| ADD {INDEX|KEY} [nombre_indice]
[tipo_indice] (nombre_columna_indice) [indice_opcion] ...
| ADD [CONSTRAINT] PRIMARY KEY
[indice_type] (indice_col_nombre, ...) [indice_opcion] ...
| ADD [CONSTRAINT] [nombre_relación] FOREIGN KEY
(nombre_columna)
REFERENCES nombre_de_tabla (nombre_columna)
```

Drop Table

```
DROP [TEMPORARY] TABLE [IF EXISTS]
nombre_tabla [,nombre_tabla] ...
```

Lenguaje de Consulta SQL

DML

SELECT

SELECT <lista de atributos>
FROM <lista de tablas>
WHERE <condición>;

```
SELECT nombre_columna, nombre_columna FROM nombre_tabla;
```

```
SELECT * FROM nombre_tabla;
```

```
SELECT DISTINCT nombre_columna, nombre_columna FROM nombre_tabla;
```

```
SELECT nombre_columna, nombre_columna FROM nombre_tabla WHERE  
nombre_columna operador valor;
```

```
SELECT nro, fecha, monto, cliente_nombre FROM facturas WHERE monto >200;
```

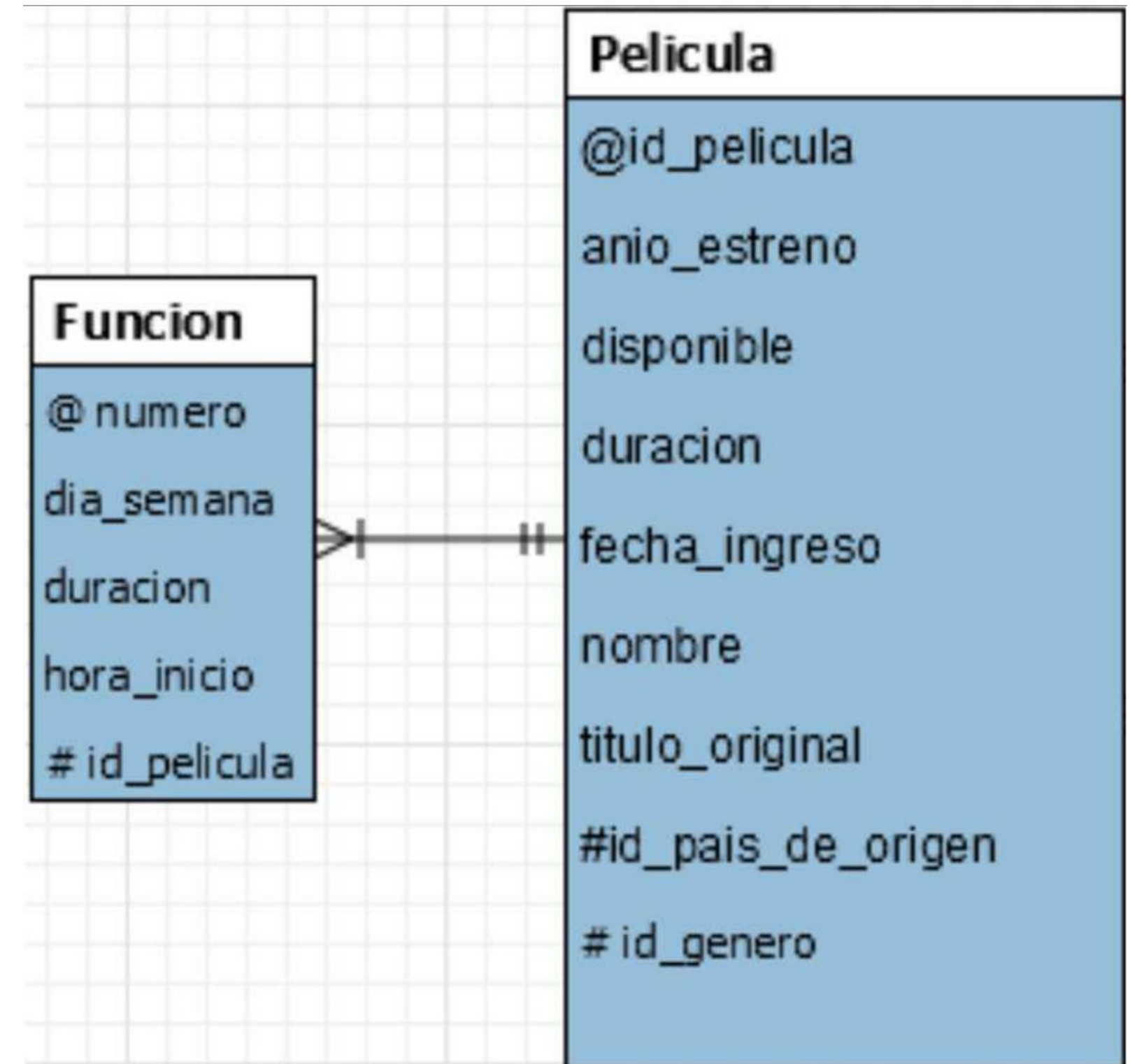
```
SELECT SUM (sueldo), MAX (sueldo), MIN (sueldo), AVG(sueldo)FROM  
empleados
```

Lenguaje de Consulta SQL

DML

Consulta multi-tablas.

```
SELECT Película.nombre, Función.dia_semana,  
Función.hora_inicio  
FROM Película INNER JOIN Funcion ON  
Funcion.id_pelicula =  
Película.id_pelicula
```



Lenguaje de Consulta SQL

DML

Consulta multi-tablas

```
SELECT pel.nombre, fun.hora_inicio, fun.dia_semana AS dia,  
FROM Pelicula pel INNER JOIN Funcion fun  
ON fun.id_pelicula = pel.id_pelicula
```

Abreviación del nombre de la
tabla.

Renombrado de
columna

LEFT JOIN != RIGHT JOIN != INNER JOIN

Lenguaje de Consulta SQL

DML

Subconsultas

Una subconsulta es una consulta anidada en una instrucción SELECT, INSERT, UPDATE o DELETE, o bien en otra subconsulta.

```
SELECT * FROM clientes WHERE id IN  
(SELECT cliente_id FROM facturas WHERE fecha BETWEEN  
2016-10-10 AND 2017-01-01);
```

Lenguaje de Consulta SQL

DML

Subconsultas

Operadores para Subconsultas

- `SELECT * FROM tabla WHERE columna ANY ([SUBCONSULTA])`
- `SELECT * FROM tabla WHERE columna ALL ([SUBCONSULTA])`
- `SELECT * FROM tabla WHERE columna IN ([SUBCONSULTA])`
- `SELECT * FROM tabla WHERE EXISTS ([SUBCONSULTA])`

Lenguaje de Consulta SQL

DML

UPDATE

```
UPDATE Pelicula  
SET disponible=1, duracion=160  
WHERE id_pelicula=4
```

```
UPDATE nombre_tabla SET  
col_nombre_1={valor1|DEFAULT} ,  
col_nombre_2={valor2|DEFAULT}  
[WHERE condicion]  
[ORDER BY ...]  
[LIMIT cantidad_filas]
```

Lenguaje de Consulta SQL

DML

INSERT

```
INSERT INTO pelicula  
VALUES (2, 2014, true, 122,  
01/08/2014, "Relatos Salvajes",  
"Relatos Salvajes",1)
```

```
INSERT INTO nombre_tabla  
(columna1, columna2,  
columna3,...)  
VALUES (valor, valor2,  
valor3,...);
```

DELETE

```
DELETE FROM Pelicula  
WHERE anio_estreno <1990
```

```
DELETE FROM nombre_tabla  
[WHERE condicion]  
[ORDER BY ...]  
[LIMIT cantidad_filas]
```