

DISEÑO ORIENTADO A OBJETOS

Prof. Esp. Ing. Agustín Fernandez

Mapeo de Objetos a Relacional

- **OO**: paradigma más utilizado
- **BD**: almacenamiento preferido y lo será por un buen tiempo.

Mapeo de Objetos a Relacional

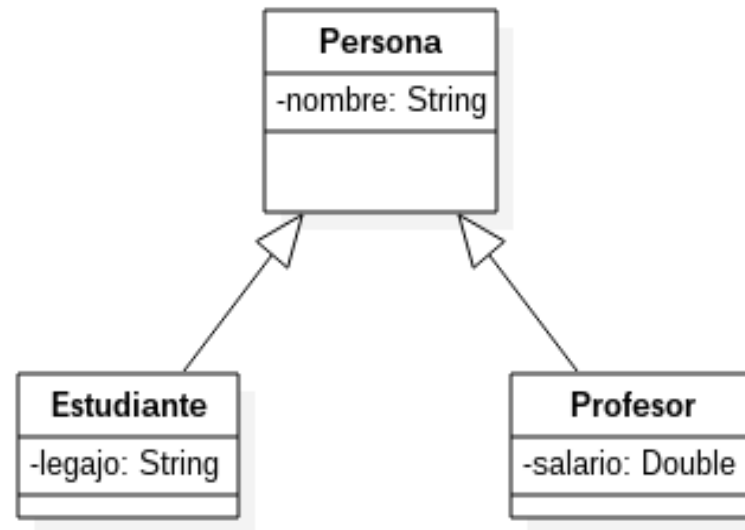
- **Orientación Objetos:**
- Principios de Ingeniería de Software (acoplamiento, cohesión, encapsulamiento, reuso).
- Objetos con datos y comportamiento.
- Se recorre los objetos a través de relaciones.
- **Base de Datos:**
- Teorías matemáticas.
- Solamente datos.
- «Datos duplicados» para hacer JOIN de las tablas

Mapeo de Objetos a Relacional

- **Reglas Básicas:**
- Atributo simple => cero o varias columnas
- No todos los atributos son persistentes (ej: calculados)
- Atributos objetos => los atributos de estos objetos serán mapeados a su propia tabla.
- Varios atributos => se agrupan en una columna.
- Surge el concepto de OID.

Mapeo de Objetos a Relacional

- Clases y Tablas: Herencia



Mapeo de Objetos a Relacional

- **Solución 1 => Una sola tabla:**

| Persona | | |
|---------|---------|--|
| PK | id | |
| | nombre | |
| | legajo | |
| | salario | |
| | tipo | |

Mapeo de Objetos a Relacional

- **Solución 1 => Una sola tabla:**
- Ventajas:
 - Simple.
 - Soporta polimorfismo cuando una persona cambia de rol.
 - Consultas ad-hoc fáciles.
- Desventajas:
 - Modificar la tabla cuando se modifica cualquiera de las clases y un error afecta toda la jerarquía.
 - Espacio vacío.
 - No permite herencia múltiple.

Mapeo de Objetos a Relacional

- **Solución 2 => Una tabla para cada clase concreta:**

| Estudiante | | |
|------------|--------|--|
| PK | id | |
| | nombre | |
| | legajo | |

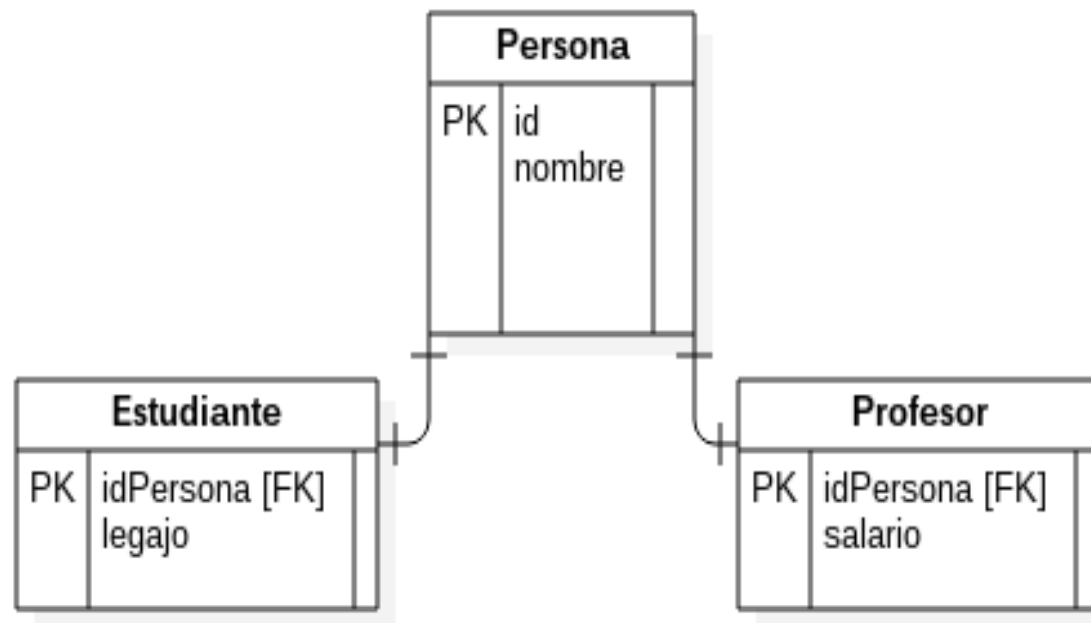
| Profesor | | |
|----------|---------|--|
| PK | id | |
| | nombre | |
| | salario | |

Mapeo de Objetos a Relacional

- **Solución 2 => Una tabla para cada clase concreta:**
- Ventajas:
 - Fácil para reportes y no se desperdicia espacio.
 - Si se alteran las clases concretas sólo se toca una tabla.
 - Permite la herencia múltiple.
- Desventajas:
 - Modificar la clase abstracta implica modificar tantas tablas como subclases.
 - Cambio de rol implica transferir datos entre tablas.
 - Herencia múltiple implica datos replicados que pueden ocasionar inconsistencias.

Mapeo de Objetos a Relacional

- Solución 3 => Una tabla para cada clase:



Mapeo de Objetos a Relacional

- **Solución 3 => Una tabla para cada clase concreta:**
- Ventajas:
 - Fácil de extender con nuevas clases.
 - Si se modifica una clase solo se modifica una tabla.
 - Más parecido al dominio y buen soporte al polimorfismo
 - Fácil modificar la superclase.
- Desventajas:
 - Muchas tablas y relaciones.
 - Más tiempo R/W porque hay que acceder a varias tablas.
 - Más complejas las consultas.

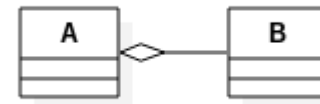
Mapeo de Objetos a Relacional

- **Relaciones: Asociación, agregación, composición**

- **Asociación:** Hay una relación entre los objetos.



- **Agregación:** Una instancia de B puede pertenecer a varias instancias de A.
- B puede existir sin que exista A.



- **Composición:** Una instancia de B solo puede pertenecer a una instancia de A.
- No puede B existir si no es dentro de A.



Mapeo de Objetos a Relacional

- **Relaciones: Asociación, agregación, composición**
- Desde la perspectiva de BD, la única diferencia es cuán ligados están los objetos.
- En agregación/composición, lo que le pasa al todo afecta a los objetos y se guarda y recupera todo junto, pero no así en la asociación.

Mapeo de Objetos a Relacional

- **Relaciones: Asociación, agregación, composición**
- La implementación se traduce en Foreign Keys
- Relaciones 1-1 y 1-N, alcanza con incluir la clave de la tabla del 1 en la tabla del N.
- Si la relación es unidireccional, se agrega en una tabla.

Mapeo de Objetos a Relacional

- **Relaciones: Asociación, agregación, composición**
- Para las relaciones N a M se necesita una tabla asociativa, cuyo único objetivo es guardar la relación.
- PK es la combinación de las dos PK.
- Se suele nombrar con los nombres de las dos entidades que relaciona.