

Persistencia de objetos

Bases de datos II

MAPEO OBJETO-RELACIONAL

Mapeo objeto - relacional

- Escenario de trabajo:
 - Contamos con un diseño orientado a objetos.
 - Contamos con una base de datos relacional.
 - Se intenta persistir la información administrada por el sistema O.O. en una base de datos relacional.
- ¿Hoy por hoy es una selección correcta de paradigmas?
- ¿Posibles inconvenientes?

Mapeo objeto - relacional

- Ventajas de la orientación a objetos:
 - Permite diseñar soluciones robustas y flexibles.
 - Contiene conceptos como abstracción, herencia, polimorfismo, encapsulamiento.
 - La mayoría de las plataformas de desarrollo hoy por hoy son o están tendiendo a ser orientadas a objetos.
 - Múltiples técnicas de diseño disponibles (MVC, double dispatching, patrones de diseño, etc).
 - Ideal para aplicaciones con mucho “comportamiento”.

Mapeo objeto - relacional

- Ventajas de las bases de datos relacionales:
 - Tecnología madura (+ de 40 años).
 - Gran base de instalaciones.
 - Mucha experiencia adquirida.
 - Lenguaje estándar: SQL 92.
 - Base matemática (álgebra relacional y cálculo de tuplas).
 - Performance en aplicaciones orientadas a datos.

Mapeo objeto - relacional

- ¿Cualidades deseables de la integración?
 - Transparencia para el sistema OO de los detalles de la persistencia.
 - Control de todas las propiedades ACID.
 - Performance sin compromisos.
 - Integración fácil y sin limitaciones.
 - Soporte para múltiples bases de datos (sql 92).

Mapeo objeto - relacional

- Estrategias de mapeo
 - OIDs
 - Principio de unicidad
 - Sin valores del dominio
 - Mapeo de atributos a columnas
 - Mapeo de clases a tablas
 - Mapeo de relaciones
 - Uno a uno
 - Uno a muchos
 - Muchos a muchos
 - Asociación vs agregación (asociación fuerte)

Mapeo objeto - relacional

- Mapeo de OIDs

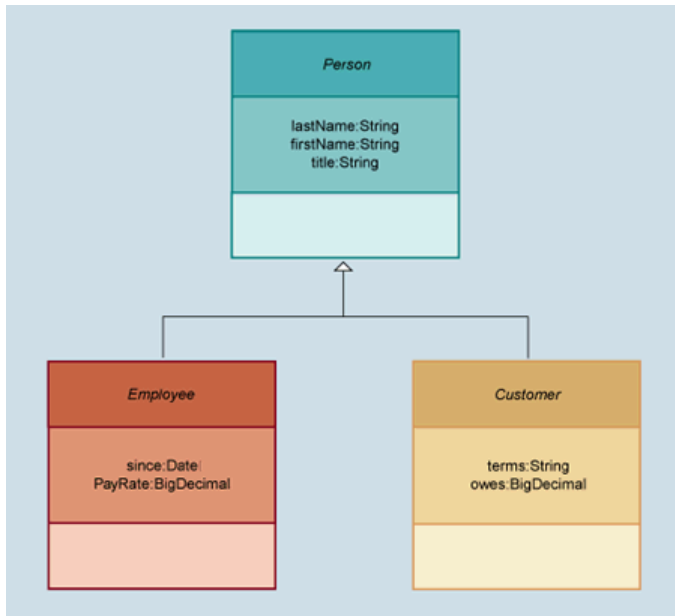
- En general el OID de un objeto no es accesible dentro de los sistemas OO.
- Al mapearlo al concepto de clave primaria de una BDR se debe considerar lo siguiente:
 - Debe ser único
 - No debe tener valores relacionados con el dominio.
- Existen diferentes estrategias para su generación.
- Impacto negativo en el diseño de objetos (aparece en las clases!!).

Mapeo objeto - relacional

- Mapeo de atributos a columnas
 - Los atributos de las clases se mapean a cero o más columnas de una o más tablas.
 - No todos los tipos son directamente mapeables.
 - Esto significa:
 - Existen atributos que no se desea persistir.
 - Existe la posibilidad de tener diferente granularidad entre el diseño OO y el de la base de datos.
 - ¿Qué pasa con los atributos multivaluados (colecciones por ejemplo)?

Mapecto objeto - relacional

- Mapecto de clases a tablas
 - Es importante tratar de respetar el concepto de herencia.
 - Existen fundamentalmente tres alternativas diferentes:

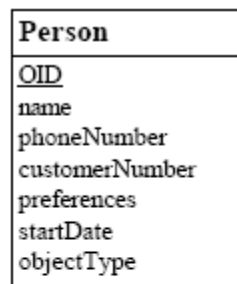


- Mapecto de toda la jerarquía a una sola tabla [1].
- Una tabla por clase concreta [2].
- Una tabla para cada clase [3].

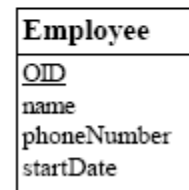
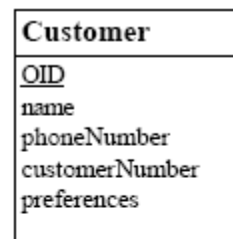
Mapeo objeto - relacional

- Mapeo de clases a tablas

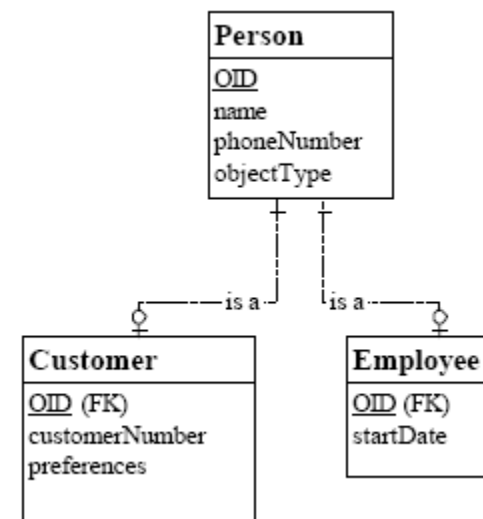
[1]



[2]



[3]



Mapeo objeto - relacional

- Mapeo de clases a tablas
 - Una tabla para toda la jerarquía
 - Ventajas:
 - Las consultas sobre todas las instancias son simples (no se requieren joins).
 - No se repite información inútilmente para instancias con 2 o más roles.
 - Desventajas:
 - Cada vez que se modifica cualquier clase hay que alterar todas las demás.
 - Se desperdicia mucho espacio de almacenamiento.

Mapeo objeto - relacional

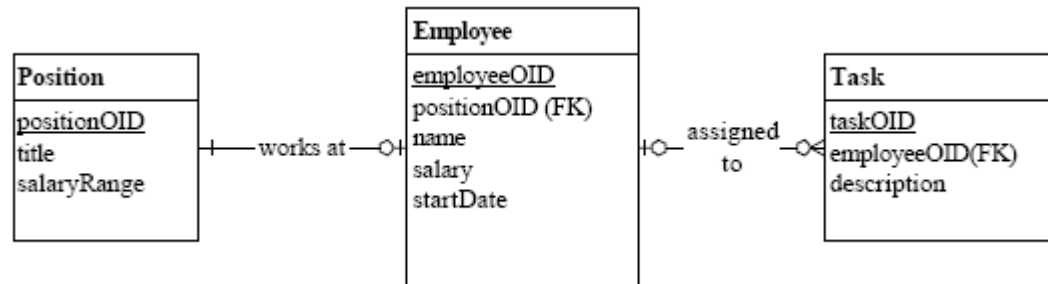
- Mapeo de clases a tablas
 - Una clase para cada clase concreta
 - Ventajas:
 - Los cambios en una clase no afectan a otras tablas mas que a la propia.
 - Desventajas:
 - Cada vez que se modifica una superclase hay que asegurarse de modificar cada una de las subclases.
 - Es complicado cuando hay instancias con varios roles.

Mapeo objeto - relacional

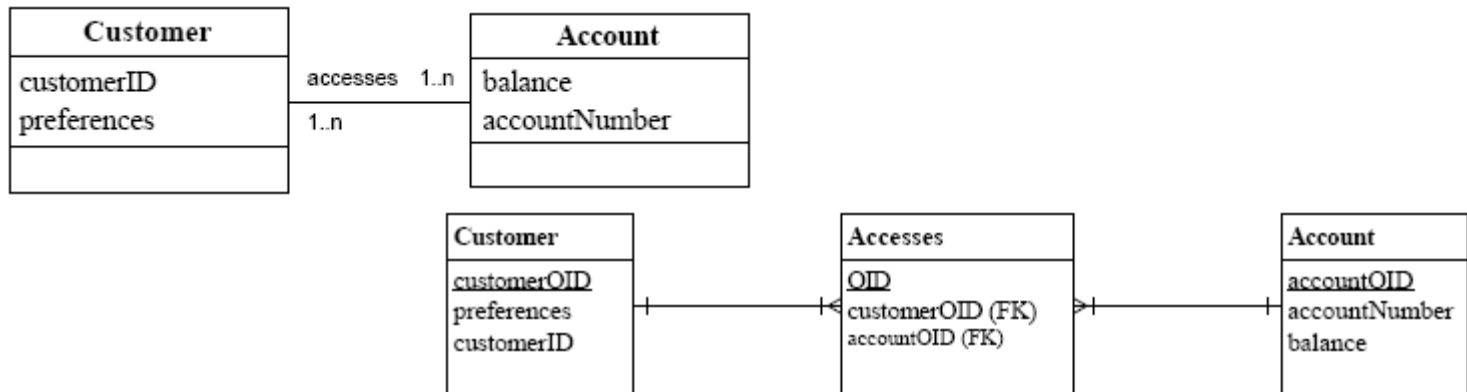
- Mapeo de clases a tablas
 - Una tabla para cada clase
 - Ventajas:
 - Es la opción que más se asemeja al paradigma OO.
 - Desventajas:
 - Implica más tablas para mantener.
 - El acceso es más lento ya que siempre se requiere un join.
 - Es difícil soportar múltiples roles.

Mapecto objeto - relacional

- Mapecto de relaciones
 - En una base de datos relacional se mantienen mediante el uso de claves foráneas.
 - uno a muchos



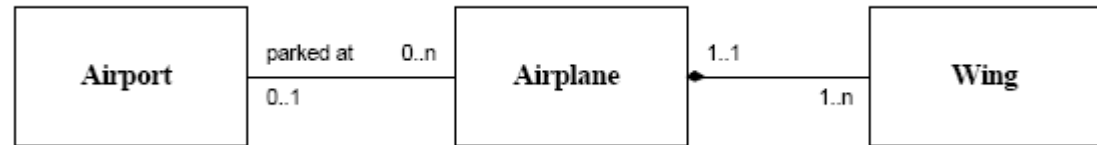
- Muchos a muchos



Mapeo objeto - relacional

- Agregación vs Composición

- La composición es una asociación “fuerte” que incorpora el concepto de “parte de” en un diseño OO.



- En términos de la base de datos, estructuralmente no hay diferencias.
- Se requieren elementos adicionales (S.P., Triggers) para mantener la consistencia.

Mapeo objeto - relacional

- Otras consideraciones
 - Operaciones CRUD y el modelo de objetos.
 - Transacciones y las propiedades ACID.
 - Esquemas de Locking
 - Pesimista y optimista.
 - Versionamiento.
 - Pseudo lenguajes de consulta.
 - Triggers.
 - Stored procedures.