

UNO. Anuncios.

Fecha para el 2do examen 3 diciembre, hora tentativa sigue siendo 930am, los estudiantes de la tarde quedaron en consultar con el profe del cual reciben clases de 930 a 1120, si va usar ese espacio durante la semana de exámenes, si sí, haríamos el examen a las 1130. Resolucion sobre esto queda pendiente.

Creo que por enredos con la sincronización de la nube de las 2 compus, se han perdido o estan distribuidos los registros de asistencia de algunas clases (al menos 2), entonces el quiz por asistencia se informa la otra semana mientras recuperamos la información.

Revisión de TP3: Miercoles 26 y Jueves 27.

DOS. Plan para las siguientes 3 clases.

- a) Tema categorización respecto de modelado conceptual (uso de Union, implementación de polimorfismo),
- b) lo mínimo necesario de cálculo relacional y álgebra relacional (en 2do examen habrá pregunta al respecto)
- c) ACID respecto de transacciones de BD, Atomico, Consistente, Isolated y Durable, usamos estos 4 temas para cubrir lo que en el programa de curso se llama Concurrencia y Recuperación de errores, dedicamos la clase que viene y la clase presencial, que es la última lectiva. El ex final, parte teórica, enfatizará en este tema. La A referida a transacciones de BD, la hemos conversado bastante.

TRES. Categorización, último tema referido al modelado conceptual.

Es una implementación de polimorfismo, lo cual NO se implementa en SQL de manera natural, sin embargo en la vida real surgen situaciones que impelen al polimorfismo.

Desde nuestro contexto de BD, el 'polimorfismo' refiere a una tabla (clase) que contiene las PK de tablas o clases diferentes, y tales PK son o de naturaleza diferente (diferente tipo) o son del mismo tipo con diferente significado. La llave que une tales PK, se llama llave subrogada, definida mas adelante.

El ejemplo típico para este problema es de asociar préstamos con garantías en un sistema de créditos. Una garantía, es el activo que toma el banco o financiera, si el deudor no paga el préstamo.

Sistema de crédito es un seguimiento de asuntos, referido a todo el proceso desde que se hace la solicitud de un crédito hasta que termina el proceso de entregar al dinero del crédito e iniciar su cobro (sistema de recuperación del crédito, o cuentas por cobrar). Las garantías de un préstamo pueden ser:

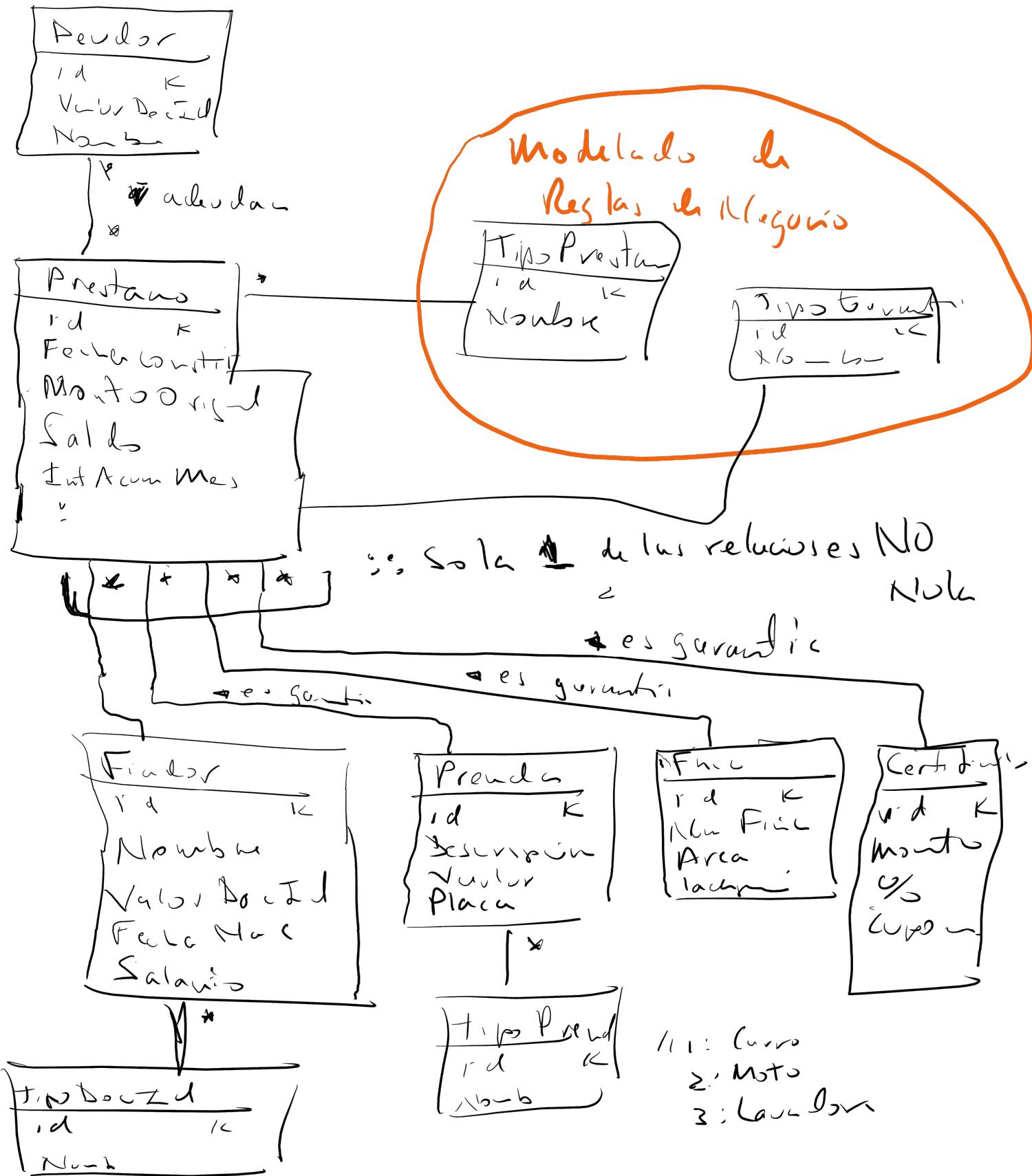
- una finca (si el préstamo es hipotecario), si el deudor no paga el banco le embarga la finca.
- uno o más fiadores, si el préstamo es fiduciario, si el deudor no paga, se embarga el salario de los fiadores.
- Prendario, maquinaria o vehículo o un reloj Rolex, si el deudor no paga se le embarga el vehículo.

- Ahorros a plazo o acciones: si el deudor no paga, se le embarga una inversión a plazo, tal como un certificado, o acciones de una empresa, o derechos de autor sobre algo.

La característica especial del problema es que las garantías son de diversa naturaleza, para cada tipo de garantía los atributos son diversos, las llaves naturales de cada tipo de garantía son diferentes. Es difícil modelar un CE que se llame garantía que cubra todas las posibilidades de llaves y atributos.

- Nota al margen, algo similar sucede con los CC de cobro de la tarea programada, no es buen diseño que los CC se implementen en una sola tabla.

El modelado sería de esta forma:



Típicamente, toda aplicación tiene un modulo o parte que representa las reglas de negocio (en servicios municipales, es para concepto de cobro como se cobra?, en un POS para cada tipo de descuento, como se descuenta?, en un sistema de tarjetas de crédito (tipo de tarjeta: normal, clasic, platinum, extraplatinum), tipo de cuenta de ahorro,

Limitaciones o problemas del modelo:

El diseño es poco mantenible, pues si aparece un nuevo tipo de garantía, será necesario hacer 2 cambios estructurales:

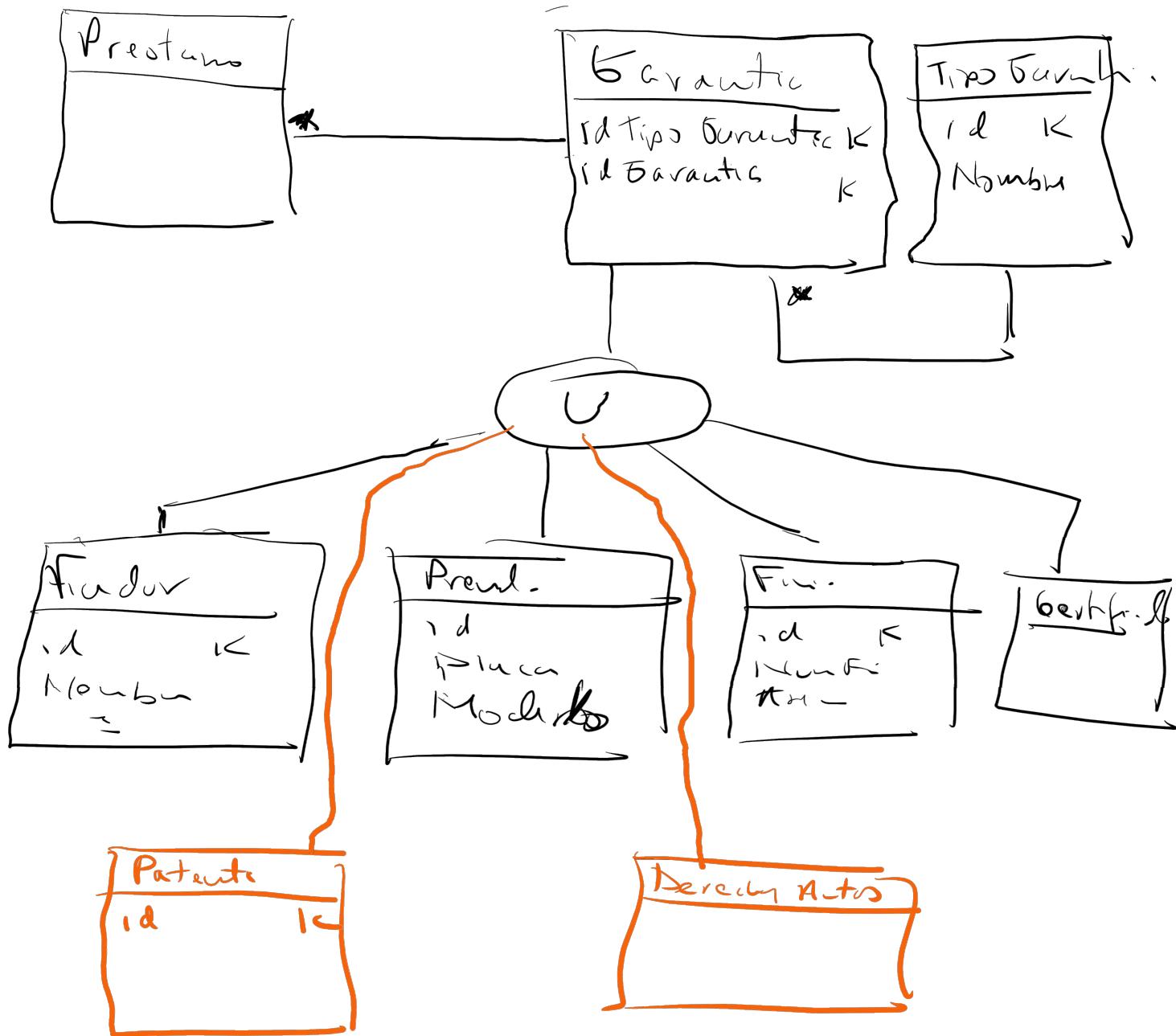
- Hay que crear una nueva tabla para el nuevo tipo de garantía, por ejemplo: derechos de autor sobre software desarrollado (si no pago el prestamo, el banco se hace dueño del software), acciones de una empresa, derechos de autor sobre una canción que ud compuso.
- Hay que crear una nueva columna en Prestamo con el FK de la nueva tabla que representa la garantía.

Hay que asegurar que el prestamos solo se asocia con 1 tipo de garantía, lo cual es solo posible hacerlo programáticamente, si hay pulgas en la programación, la BD queda inconsistente.

La alternativa es un nuevo constructor de modelado en UML, llamado categorización que se anota con una U dentro de un círculo, el dibujo se parece al de herencia solo que la herencia, en lugar de circulo, utiliza un triángulo equilátero apuntando arriba.

El nombre de categorización es poco afortunado, mejor se si llamara polimorfismo o Union.

La Union depende de un CE en la cual se representa la unión de todas las llaves que estan en la categorización, en nuestro caso, de este curso, ya que todas las llaves son enteras y simples, si bien son de un mismo tipo, refieren a objetos muy diferentes, por tanto son un atributo variante, o sea que este CE que ‘une llaves’ no cumple la 1era forma normal, por otro lado esas llaves son PK en los CE de la unión, pero NO PUEDEN SER FKs, pues las FK refieren a solo una tabla, en una unión tenemos llaves de muchas tablas, ese columna que ‘une llaves’ se llama ‘llave subrogada’. Si las PK de cada tabla en la categorización tienen diferente tipo: cedula para fiadores, num de finca para finca, numero placa de carro para prenda, etc, etc., Entonces la llave subrogada será de tipo de variante, por ejemplo: varchar(64) y dependiendo del tipo de garantía hacer un convert.



Este modelo, aunque traiciona 1nf, es mas flexible que el anterior, nuevos tipos de garantías se agregan a la unión, sin necesidad de modificar la estructura de otras tablas (la tabla prestamos)

La llave en garantía idTipoGarantia + IdGarantia, donde IdGarantia es PK local a cada CE que representa una garantía (id de Fiador, o Id Prenda, o Id deFinca), el IdTipoGarantia es necesario para asegurar que las llaves son únicas, pues la PK id = 30, puede existir como id en Fiador o id en Patentes, no de otra que la PK en garantía sea compuesta, no hay opción.

Este modelo tiene una limitación: la única manera de asegurar que la llave subrogada refiere a una PK que realmente existe, es programáticamente, pues las BD relacionales no ofrecen algo así como ‘Integridad referencial a una de varias tablas’, entonces si se borra en Fiador también hay que borrar la referencia de la PK en garantía, eso solo se puede hacer programáticamente.

O sea que la Union, se implementa en BD SQL programáticamente, o dicho de otra forma: SQL no tiene estatutos tipo DDL (Data Definition Language) para implementar categorización.

CUATRO. Lenguajes de manipulación teóricos en modelo relacional.

Para qué sirve conocer de este tema:

- Para poder formalizar operaciones sobre tablas o conjuntos relación (relación o asociación de dominios de atributos). Una tabla es la implementación de un conjunto relación, la tabla es más un concepto físico, mientras que conjunto relación es un concepto teórico.
- Para ser vanidoso con el conocimiento.

Hay de 2 tipos: procedural y no-procedural (declarativo?)

El SQL en realidad es una mezcla de lenguaje procedural y declarativo!!!

Algebra relacional basado en operaciones que son típicamente operaciones sobre conjuntos tales como: Select (operación sobre filas, contradictoriamente respecto de SQL), project (operación sobre columnas como el Select de SQL), producto cartesiano, join, natural join, unión, intersección, diferencia, division, y outer join.

Calculo relacional, que hay de 2 tipos: de variable tuplo y de variable dominio, es declarativo y se usan cuantificadores de Para todo y Existe, además de declarar variables a través de pertenece.

Algebra relacional al ser procedural especifica el cómo, el cálculo relacional al ser declarativo especifica el QUÉ. Ambos tratan con conjuntos.