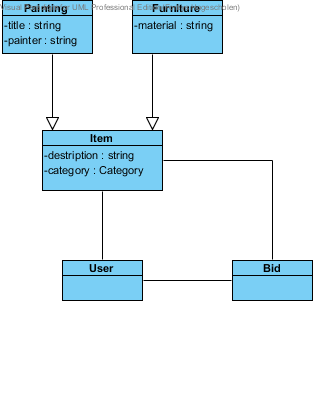
## Opgave overerving

Dit is een opgave waarbij je kennismaakt met overerving. De veiling gaat nu speciale variaties van items ondersteunen , waarvoor extra gegevens worden bijgehouden (zie onderstaande figuur).

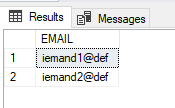
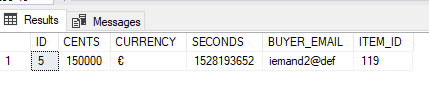
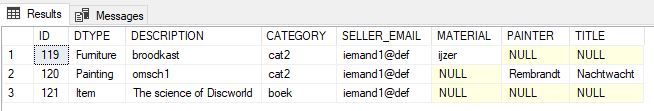
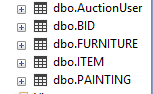
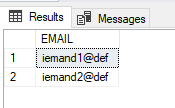
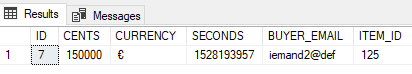
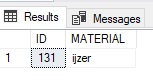
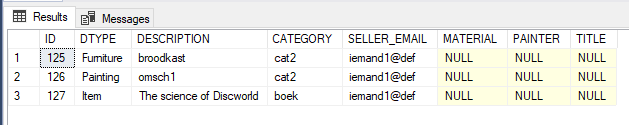
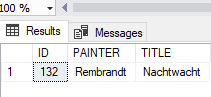


Figuur 1

### Voorbereiding.

Kopieer je project uit de vorige opgave naar een nieuw NetBeans project.

### Opdracht 1

1. Schrijf de Javaklassen *Furniture* en *Painitng*. Deze klassen dienen o.a. een constructor te bevatten die alle attributen via parameters een waarde geeft, en verder de gebruikelijke getters. (Beide zijn het gemakkelijkst te maken door met de rechtermuisknop in de tekst van de klasse te klikken en **Insert Code …** te kiezen.)
2. Maak van deze klassen entiteiten door toevoegen van de annotatie @Entity en nog wat andere, inmiddels bekende handelingen. Definieer geen identifier. Dat doet de superklasse *Item* al.
3. Download de FurnitureAndPainintingTest van Sharepoint, unzip en voeg hem toe aan je project.
4. Voeg aan *SellerMgr*  de methoden *offerFurniture*  en *offerPaining*  toe die in de test worden aangeroepen en zorg dat ze het verwachte gedrag vertonen.
5. Voer de test uit. Verbeter eventueel de code totdat de test slaagt. Noteer de inhoud van de database na uitvoering van de test.
6. 
7. 
8. 
9. 
10. Voeg in *Item* de annotatie @Inheritance (strategy = InheritanceType.JOINED) toe en voer de test opnieuw uit. Noteer ook nu de inhoud van de database.
11. 
12. 
13. 
14. 
15. 
16. 
17. Voer de test nu uit voor de derde variant, InheritanceType.TABLE\_PER\_CLASS. Noteer ook nu de inhoud van de database.
18. Als je de figuur goed bekijkt, zie je dat *Item* cursief is geschreven. In UML betekent dat dat de klasse abstract is. Er kunnen dan wel Meubels en Schilderijen, maar geen andere Items worden gecreëerd. We gaan eens kijken of dat verschil maakt in de database. Maak klasse *Item* abstract, en commentarieer de methode *SellerMgr.offerItem* uit. Doe dit ook met het deel van de test waar deze methode wordt aangeroepen (voor item3). Als je de test nu wilt uitvoeren, meldt NetBeans waarschijnlijk dat er compileerfouten optreden in het project. Dit komt door de aanroepen van *offerItem* in de andere testen, maar die wil je nu niet uitvoeren. Klik daarom op op **Run Anyway.** Zijn de resultaten anders, afgezien van het ontbreken van *item3*? Beantwoord deze vraag voor beide mogelijke inheritancetypen.

## Beoordeling

Zie blokboek.