

Bases de Dados

**Gestão de um site de leilões**

Neste trabalho decidimos criar uma base de dados para o auxílio da gestão de um site de leilões.

Na base de dados que esquematizámos existe a classe **User** da qual derivam as classes **Buyer** e **Vendor** que herdam os atributos da casse **User**.

No site existem **Advertisements** que são criados por um **Vendor** e podem incluir um ou mais **Items**. E até 3 **Items** podem estar presentes em **Advertisements**. Se um **Item** for vendido será eliminado de todos os **Advertisements** em que estiver listado.

Os **Advertisements** são caracterizados por **Tags**. Isto será útil para pesquisar apenas os **Advertisements** relevantes para um certo **User**.

Para fazer uma compra, um **Buyer** terá de realizar uma **Bid** por um determinado **Item**. Se a **Bid** feita for superior à highestBid associada a esse **Item**, então o **Buyer** tornar-se-á o highestBidder desse **Item**. A **Bid** só poderá ser feita entre o startTime e o closeTime associados ao **Item**.

Quando terminar o leilão será efetuada a **Sale** sobre a qual o site ganha uma siteCommission. O **Item** será enviado até à transferLimitDate para um **Address** a determinar pelo **Buyer** ao concluir o leilão.

Ao ser finalizada a **Sale**, o **Buyer** poderá ser feita entre o startTime e o closeTime associados ao **Item**.

Quando terminar o leilão será efetuada a **Sale** sobre a qual o site ganha uma siteCommission. O **Item** será enviado até à transferLimitDate para um **Address** a determinar pelo **Buyer** ao concluir o leilão.

Ao ser finalizada a **Sale**, o **Buyer** poderá votar no **Vendor** e a rating do **Vendor** será atualizada.

**Atributos das Classes:**

**User:**

**Advertisement:**

* advertisementID: int
* pageNumber: int
* description: string
* exposingFee: double

**Item:**

* itemID: int
* name: string
* shippingCoverage: bool
* startTime: string
* closeTime: string
* userID: int
* Name: string
* Password: string
* nib: int
* email: string
* rating: double

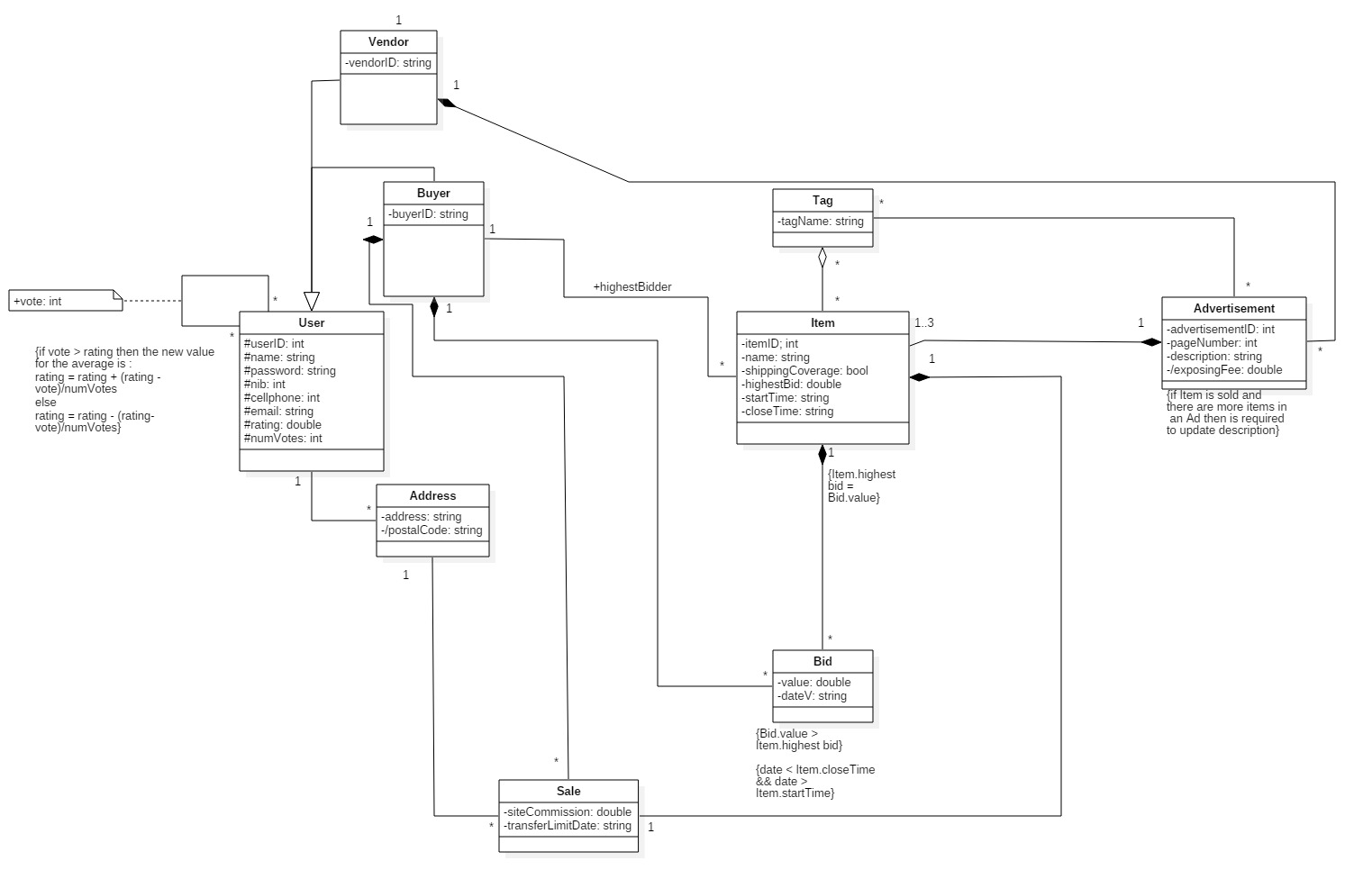
**Tag:**

* name: string

**Bid:**

* value: double
* dateV: string
* numVotes: int

**Sale:**

* siteCommission: double
* transferLimitDate: string

1

Grupo 501:

Christopher Fernandes de Abreu – up201604735

Tiago Pinho Cardoso – up201605762

Vasco Trindade Enes – up201505519

**Modelo Relacional:**

**User** (userID, name, password, nib, cellphone, email, rating, numVotes)

**Buyer** (buyerID , name, password, nib, cellphone, email)

**Vendor** (vendorID , name, password, nib, cellphone, email, rating, numVotes)

**Vote** (buyerID->Buyer,vendorID->Vendor, vote)

**Tag** (name)

**Advertisement** (advertisementID, pageNumber, description, /exposingFee, vendorID-> Vendor)

**Address** (address, /postalCode, userID->Buyer)

**Sale** (siteCommission, transferLimitDate, itemID->item, address->Address, buyerID->Buyer)

**Item** (itemID, name, shippingCoverage, startTime, closeTime, highestBid, advertisementID-> Advertisement, buyerID->Buyer, name->Tag)

**ItemTag**(itemID->Item, name->Tag)

**Advertisement** (advertisementID, pageNumber, description, /exposingFee, vendorID-> Vendor)

**AdvertisementTag**(advertisementID ->Advertisement, name->Tag)

**Bid** (value, dateV , itemID-> Item, userID-> Buyer)

**Restrições**

As restrições estão presentes no ficheiro create.sql

**BCNF:**

Todas as relações estão na forma normal de Boyce-Codd uma vez que o conjunto das primary keys em cada uma das relações é a chave que dá todos os os outros atributos dentro dessa relação. Havendo apenas uma dependência funcional em cada relação do género:

primaryKeyA, primaryKeyB,… -> NonKeyAtributeA, NonKeyAtributeB, …

Sendo a chave desta relação { primaryKeyA, primaryKeyB,… } correspondente ao lado esquerdo desta depência ilustrando assim o que foi referido anteriormente.

**3NF:**

Estas relações também elas estão na terceira forma normal porque em todas as relações a sua única dependência functional possui no lado esquerdo uma superkey visto que nenhum dos seus atributos serve como superkey dessa relação.