Sistemas de Informação 2

Semestre Verão 15/16

Primeira Fase

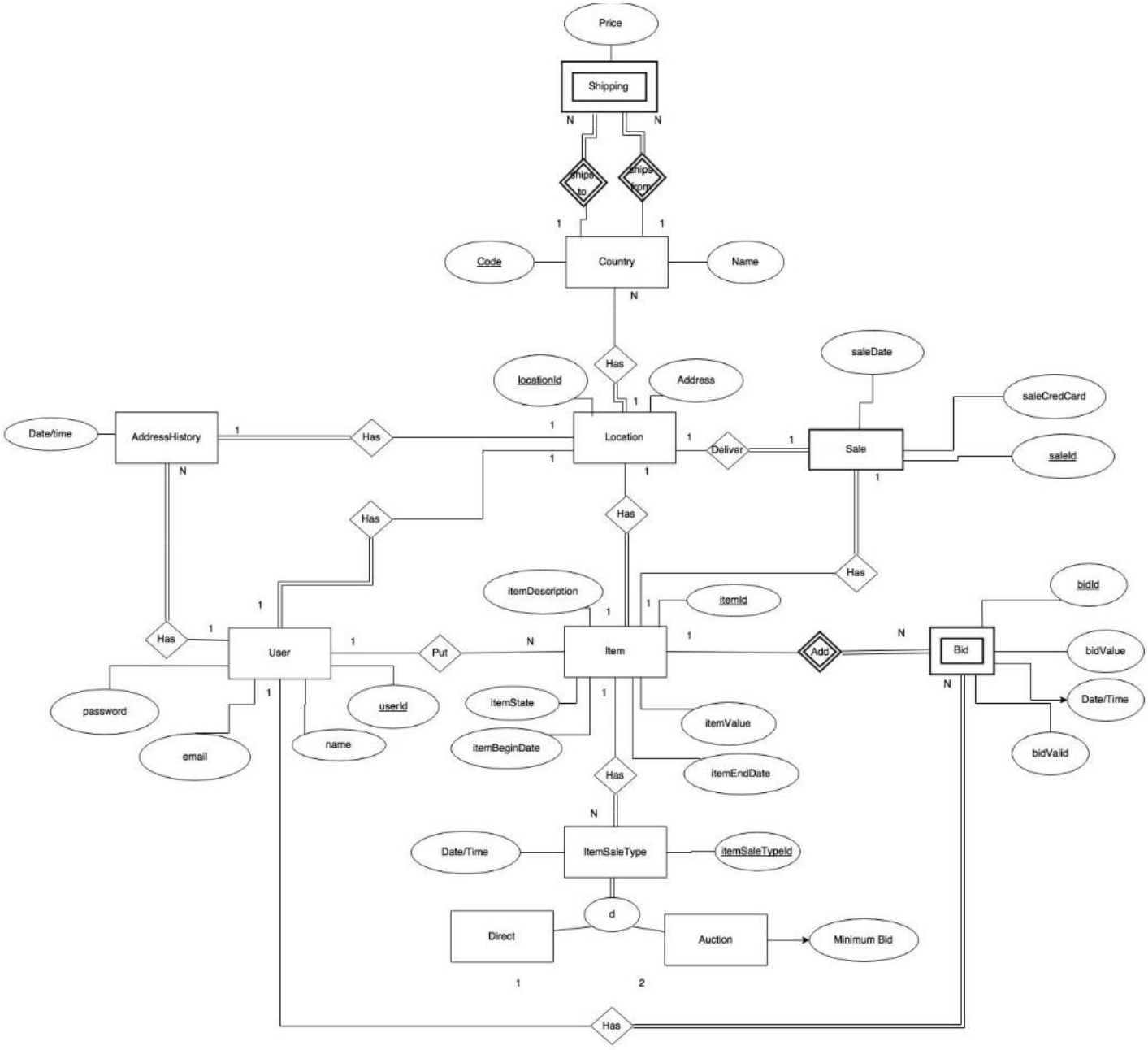
Trabalho realizado por:

40622: Pedro Gregório

40637: Francisco Dias

40641: António Cunha

Modelo EA


Price
Shipping
N
N
ships to
ships from
1
1
Code
Country
Name
N
Has
locationId
Address
saleDate
saleCredCard
saleId
Date/time
AddressHistory
Has
Location
Deliver
Sale
Has
Has
Has
itemDescription
itemId
bidId
Has
User
Put
Item
Add
Bid
bidValue
Date/Time
bidValid
password
email
name
userId
itemState
itemBeginDate
itemValue
itemEndDate
Has
Date/Time
ItemSaleType
itemSaleTypeId
d
Direct
Auction
Minimum Bid
1
2
Has

Restrições de Integridade

- Item.ItemState, só pode ter os valores "Novo", "Usado", "Como Novo" ou "Velharia vintage"
- Venda directa tem o valor identificador " Direct" e Leilão tem o valor identificador “Auction”
- Quando a venda é do tipo Auction, a licitação mínima entre 1€ e 10% do valor mínimo de venda
- Não é possível alterar o valor Users.userEmail

Modelo Relacional

Shipping - {CodeFrom, CodeTo, Price}

PK = {CodeFrom, CodeTo}

FK = {CodeFrom, CodeTo}

Country - {Code, Name}

PK = {Code}

AK = {Name}

Location - {LocationId, LocationCountry, Address}

PK = {LocationId}

FK = {LocationCountry}

AddressHistory - {addrHistoryId, addrHistoryUser, addrHistoryLocation, addrHistoryDateTime}

PK = {addrHistoryId}

FK = {addrHistoryUser, addrHistoryLocation}

Users - {userId, userLocation, userEmail, userName, userPassword}

PK = {userId}

FK = {userLocation}

AK = {userEmail}

Item - {itemId, itemLocationId, itemValue, itemDescription, itemState, itemBeginDate, itemEndDate, itemSaleType, itemUserId}

PK = {itemId}

FK = {itemSaleType}, {itemLocation}, {itemUserId}

ItemSaleType - {itemSaleTypeId, itemSaleTypeItemId, itemSaleTypeDesc, auctionMinimumBid, itemSaleTypeDate}

PK = {itemSaleTypeId}

FK = {itemSaleTypeItemId}

Bid - {bidId, bidValue, bidDateTime, bidValid, bidItemId}

PK = {bidId, bidItemId}

FK = {bidItemId}

Sale - {saleId, saleDate, saleLocation, saleCredCard, saleItemId}

PK = {saleId}

FK = {saleLocation}, {saleItemId}

Inserir, remover e actualizar informação de um artigo

Foram criados três procedimentos armazenados para cumprir este requisito, sendo que foi considerado que na actualização de um artigo, é apenas possível alterar a sua descrição, pois da forma como o projecto foi modelado, considerou-se que alterações a outros atributos de um artigo, poderia comprometer todas as licitações feitas sobre este artigo

Inserir, remover e actualizar um utilizador

Foram criados três procedimentos armazenados para cumprir este requisito, considerou-se que as alterações possíveis de se realizar sobre um utilizador seriam a alteração do userName e de userPassword e userLocation, visto que não deveria ser possível a alteração do e-mail de utilizador e o userId é interno à base de dados.

Para impossibilitar a alteração do e-mail, criámos um trigger sobre a table Users, onEmailUpdate, que caso tenha sido alterado o valor de e-mail, executa um rollback.

Na alteração da morada de um utilizador, é colocada a antiga morada na tabela AddressHistory, foi criado para tal um procedimento armazenado que trata da criação da nova morada, e armazenamento da anterior.

Inserir, remover e actualizar um local

Foram criados três procedimentos armazenados para cumprir este requisito, considerou-se que alterar um local poderia ser alterar a informação do País ou a morada referente ao local, ou ambas.

Inserir uma licitação

No âmbito do projecto foi considerado que uma venda directa estaria terminada (não seria possível realizar mais licitações) a partir do momento que existisse uma licitação que satisfizesse o valor estabelecido pelo anunciante do artigo.

Desta forma foi criado um procedimento armazenado, que dado um itemId, valor, userId, insere uma licitação, tendo em conta que:

- Na tentativa de inserção de uma licitação, a data da criação tem que anteceder a data final de venda do artigo definida na sua criação.
- No caso de se tratar de uma compra *Direct,* o valor da licitação tem que ser igual ao valor do artigo, e não pode existir prévia licitação sobre este artigo.
- No caso de se tratar de uma compra *Auction*, o valor da primeira licitação tem que ser igual ou superior ao valor do artigo. Para todas as outras, o valor tem que no mínimo ser superior ao valor da última licitação válida (bidAlive = 1), acrescido ao valor de licitação mínimo, definido a quando a inserção do artigo.

De forma a garantir o bom funcionamento da criação das licitações, foi estabelecido o nível de isolamento *REPEATABLE READ* para que não seja possível que outro cliente insira uma licitação que possa invalidar a licitação que está a ser inserida.

Retirar uma licitação

Foi criado um procedimento armazenado, que dado um bidId remove uma licitação, tendo em conta que, é considerado a remoção de uma licitação, a afetação do atributo bidAlive com o valor 0, mantendo desta forma, todas as licitações na base de dados.

Caso o artigo já tenha sido vendido, é impossibilitada a remoção da licitação.

Concluir a compra de um leilão e realizar a compra de um artigo de venda directa

Foi criado um procedimento armazenado que independentemente do tipo de venda do artigo finaliza a sua compra, verificando se o utilizador passado, é o vencedor do artigo (corresponde à última licitação máxima). Caso seja leilão, se a data atual já ultrapassa a data final definida no artigo, mas não, ultrapassando 2 dias dessa mesma data.

Determinar o valor de licitação de um artigo

Foi realizada a função BiddingPrice que a partir de um artigo determina se este é do tipo leilão ou venda directa e calcula o valor de licitação. Considerou-se que para a venda directa existiria o valor de licitação se ainda ninguém tivesse licitado(comprado) o artigo, e que esse valor de licitação seria o valor de venda do artigo, já não sendo possível realizar outras licitações.

Obter as N ultimas licitações

Foi realizada a função LastNBids que retorna numa tabela, as últimas n licitações, ordenadas pela data de licitação (mais recentes para mais antigas) e que fossem válidas.

Obter os portes, dadas duas localizações

Foi realizada a função ShippingPrice que dado o país destino e país partida, retorna o valor, que se encontra tabelado em Shipping.

Listar os leilões que não foram concluídos

Foi realizada a função NotConcludedAuction, que retorna uma tabela que contém todos os leilões, cuja a data de fim não tenha sido atingida, no momento da chamada à função.

Verificar a password de um utilizador

Foi realizada a função CheckPassword, que dado um userId, e uma password, é convertido o parâmetro password, através da função realizada MD5, verificando se corresponde à codificação guardada para o mesmo utilizador.