1. Relational Schema

> The Relational Schema includes the relation schemas, attributes, domains, primary keys, foreign keys and other integrity rules: UNIQUE, NOT NULL, CHECK. Relation schemas are specified in the compact notation:

Relation reference	Relation Compact Notation
R01	auction(state **NN**, title **NN**, description, sellingReason, pathToPhoto, startingPrice **CK** startingPrice > 0, minimumSellingPrice, buyNow, startDate, limitDate, refusalDate, /numberOfBids, reasonOfRefusal, auctionID **NN**, normalUserID -> authenticated_User **NN**, authenticatedUserID -> authenticated_User **NN**)
R02	authenticated_User(typeOfUser, username **NN UK**, password **NN**, pathToPhoto, authenticatedUserID **NN**)
R03	add_Credits(value **NN**, TIMESTAMP WITH TIME zone **NN**, addCreditsID **NN**, normalUserID -> normal_User **NN**)
R04	bid (auctionID -> auction, normalUserID -> authenticated_User, TIMESTAMP WITH TIME zone **NN**, value **NN**)
R05	category(name **NN UK**, subcategories -> category, categoryID **NN**)
R06	city(name **NN UK**, cityID **NN**, countryID -> country **NN**)
R07	comment(date **NN**, description **NN**, commentID **NN**, auctionID -> auction **NN**, authenticatedUserID -> authenticated_User , normalUserID -> authenticated_User **NN**)

R08	country(name **NN UK**, countryID **NN**)
R09	normal_User(state, completeName **NN**, email **NN UK**, birthDate, /rating **CK** /rating >= 0 && /rating <= 5 , address **NN**, postalCode **NN**, balance **CK** balance >= 0, authenticatedUserID -> authenticated_User, cityID -> city **NN**)
R10	notification(date **NN**, description **NN**, type **NN**, notificationID **NN**, auctionID -> auction, authenticatedUserID -> authenticated_User **NN**)
R11	report(date **NN**, reason **NN**, auctionID -> auction, normalUserID -> authenticated_User)
R012	win(date **NN**, finalPrice **NN**, rate **CK** rate >= 0 && rate <= 5, auctionID -> auction, normalUserID -> authenticated_User)
R013	pertains_to (auctionID -> auction, categoryID -> Category)
R014	blocks (normalUserID -> authenticated_User, authenticatedUserID -> authenticated_User)
R015	remove_Moderator(authenticatedUserID -> authenticated_User, removedMod -> authenticated_User)
R016	create_Moderator(authenticatedUserID -> authenticated_User, addedMod -> authenticated_User)

2. Domains

The specification of additional domains can also be made in a compact form, using the notation:

Domain Name	**Domain Specification**

Auction State	ENUM ('Active', 'Rejected', 'Pending')
Normal User State	ENUM ('Seller', 'Bidder')
Type of User	ENUM ('Moderator', 'Administrator', 'Normal')

3. Functional Dependencies and schema validation

> To validate the Relational Schema obtained from the Conceptual Model, all functional dependencies are identified and the normalization of all relation schemas is accomplished. Should it be necessary, in case the scheme is not in the Boyce-Codd Normal Form (BCNF), the relational schema is refined using normalization.

Table R01 (auction)

Keys:{ auctionID }	
Functional Dependencies	
FD0101:	{auctionID} → {state, title, description, sellingReason, pathToPhoto, startingPrice, minimumSellingPrice, buyNow, startDate, limitDate, refusalDate, /numberOfBids, reasonOfRefusal, normalUserID, authenticatedUserID}
Normal Form:	BCNF

Table R02 (authenticated_User)

Keys:{ authenticatedUserID, username }	
Functional Dependencies	
FD0201:	{authenticatedUserID} → {username, typeOfUser, password, pathToPhoto}
FD0202:	{username} → {authenticatedUserID, typeOfUser, password, pathToPhoto}

Normal Form:	BCNF

Table R03 (add_Credits)

Keys:{ addCreditsID }	
Functional Dependencies	
FD0301:	{addCreditsID} → {value, TIMESTAMP WITH TIME zone, normalUserID}
Normal Form:	BCNF

Table R04 (bid)

Keys:{ (auctionID, normalUserID) }	
Functional Dependencies	
FD0401:	{ (auctionID, normalUserID) } → {date, value}
Normal Form:	BCNF

Table R05 (category)

Keys:{ categoryID, name }	
Functional Dependencies	
FD0501:	{ categoryID } → {name, subcategories}
FD0502:	{ name } → {categoryID, subcategories}
Normal Form:	BCNF

Table R06 (city)

Keys:{ cityID, name}	
Functional Dependencies	

FD0601:	{ name } → {cityID, countryID}
FD0602:	{ cityID } → {name, countryID}
Normal Form:	BCNF

Table R07 (comment)

Keys:{ commentID }	
Functional Dependencies	
FD0701:	{ commentID } → {date, description, auctionID, authenticatedUserID, normalUserID}
Normal Form:	BCNF

Table R08 (country)

Keys:{ countryID }	
Functional Dependencies	
FD0801:	{ countryID } → {name}
Normal Form:	BCNF

Table R09 (normal_User)

Keys:{ authenticatedUserID, email }	
Functional Dependencies	
FD0901:	{ authenticatedUserID } → {state, completeName, email, birthDate, rating, address, postalCode, balance, cityID}
FD0902:	{ email } → {state, completeName, birthDate, rating, address, postalCode, balance, authenticatedUserID, cityID}
Normal Form:	BCNF

Table R10 (notification)

Keys:{ notificationID }	
Functional Dependencies	
FD1001:	{ notificationID } → {date, description, type, auctionID, authenticatedUserID}
Normal Form:	BCNF

Table R11 (report)

Keys:{ (auctionID, normalUserID) }	
Functional Dependencies	
FD1101:	{ (auctionID, normalUserID) } → {date, reason}
Normal Form:	BCNF

Table R12 (win)

Keys:{auctionID}	
Functional Dependencies	
FD1201:	{auctionID} → {date, finalPrice, rate, auctionID, normalUserID}
Normal Form:	BCNF

Table R13 (pertains_to)

Keys:{auctionID}	
Functional Dependencies	
FD1301:	{auctionID} → {categoryID)
Normal Form:	BCNF

Table R14 (blocks)	
Keys:{normalUserID}	
Functional Dependencies	
FD1401:	{normalUserID} → authenticated_User
Normal Form:	BCNF

Table R15 (remove_Moderator)	
Keys:{removedMod}	
Functional Dependencies	
FD1501:	{removedMod} → {authenticatedUserID}
Normal Form:	BCNF

Table R16 (create_Moderator)	
Keys:{addedMod}	
Functional Dependencies	
FD1601:	{addedMod} → authenticatedUserID
Normal Form:	BCNF

AS all relations schemas are in the Boyce-Codd Normal Form (BCNF), the relational schema is also in the BCNF and therefore there is no need to be refined using normalisation.

4. SQL Code

> SQL code necessary to build (and rebuild) the database.

> PCAuctions.sql

```
ALTER TABLE Add_Credits DROP CONSTRAINT add_credits;
```

```
ALTER TABLE Auction DROP CONSTRAINT win;
ALTER TABLE Auction DROP CONSTRAINT create;
ALTER TABLE Auction DROP CONSTRAINT accepts;
ALTER TABLE Auction DROP CONSTRAINT rejects;
ALTER TABLE Authenticated_User DROP CONSTRAINT add;
ALTER TABLE Authenticated_User DROP CONSTRAINT
remove;
ALTER TABLE Bid DROP CONSTRAINT Auction;
ALTER TABLE Bid DROP CONSTRAINT Normal_User;
ALTER TABLE Category DROP CONSTRAINT has;
ALTER TABLE City DROP CONSTRAINT FK_City_Country;
ALTER TABLE Comment DROP CONSTRAINT to;
ALTER TABLE Comment DROP CONSTRAINT removes;
ALTER TABLE Comment DROP CONSTRAINT adds;
ALTER TABLE Normal_User DROP CONSTRAINT
FK_Normal_User_Authenticated_User;
ALTER TABLE Normal_User DROP CONSTRAINT blocks;
ALTER TABLE Normal_User DROP CONSTRAINT
FK_Normal_User_City;
ALTER TABLE Notification DROP CONSTRAINT pertains_to;
ALTER TABLE Notification DROP CONSTRAINT receives;
ALTER TABLE pertains_to DROP CONSTRAINT Category;
ALTER TABLE pertains_to DROP CONSTRAINT Auction;
ALTER TABLE Report DROP CONSTRAINT Auction;
ALTER TABLE Report DROP CONSTRAINT Normal_User;

DROP TABLE IF EXISTS Add_Credits;
DROP TABLE IF EXISTS Auction;
DROP TABLE IF EXISTS Authenticated_User;
DROP TABLE IF EXISTS Bid;
DROP TABLE IF EXISTS Category;
DROP TABLE IF EXISTS City;
DROP TABLE IF EXISTS Comment;
DROP TABLE IF EXISTS Country;
DROP TABLE IF EXISTS Normal_User;
DROP TABLE IF EXISTS Notification;
DROP TABLE IF EXISTS pertains_to;
DROP TABLE IF EXISTS Report;
DROP TABLE IF EXISTS Win;
```

```
DROP TABLE IF EXISTS Block;
DROP TABLE IF EXISTS Remove_Moderator;
DROP TABLE IF EXISTS Create_Moderator;

CREATE TABLE Add_Credits (
     value Int NOT NULL,
     TIMESTAMP WITH TIME zone TIMESTAMP WITH TIME
zone NOT NULL,
     add_CreditsID integer NOT NULL,
     normalUserID integer NOT NULL
);

CREATE TABLE Auction (
     state Enum NOT NULL,
     title varchar(50) NOT NULL,
   description varchar(50) NULL,
     sellingReason varchar(50) NULL,
     pathToPhoto varchar(50) NULL,
     startingPrice Int NOT NULL,
     minimumSellingPrice Int NULL,
     buyNow Int NULL,
     startDate TIMESTAMP WITH TIME zone NULL,
     limitDate TIMESTAMP WITH TIME zone NOT NULL,
     refusalDate TIMESTAMP WITH TIME zone NULL,
     /numberOfBids Int NULL,
     reasonOfRefusal varchar(50) NULL,
     auctionID integer NOT NULL,
     normalUserID integer NOT NULL,
     Authenticated_UserID integer NOT NULL
);

CREATE TABLE Authenticated_User (
    typeOfUser Enum NOT NULL,
    username varchar(50) NOT NULL UNIQUE,
    password varchar(50) NOT NULL,
    pathToPhoto varchar(50) NULL,
    Authenticated_UserID integer NOT NULL
);
```

```
CREATE TABLE Bid (
     "date" TIMESTAMP WITH TIME zone NOT NULL,
     value Int NOT NULL,
     bidID integer NOT NULL,
   auctionID integer NULL,
   normalUserID integer NULL
);

CREATE TABLE Category (
     name varchar(50) NOT NULL UNIQUE,
     subcategories NULL,
     categoryID integer NOT NULL
);

CREATE TABLE City (
     name varchar(50) NOT NULL,
     cityID integer NOT NULL,
     countryID integer NOT NULL
);

CREATE TABLE Comment (
     "date" TIMESTAMP WITH TIME zone NOT NULL,
     description varchar(50) NOT NULL,
     commentID integer NOT NULL,
     auctionID integer NOT NULL,
     Authenticated_UserID integer NULL,
     normalUserID integer NOT NULL
);

CREATE TABLE Country (
     name varchar(50) NOT NULL UNIQUE,
     countryID integer NOT NULL
);

CREATE TABLE Normal_User (
     state Enum NULL,
     completeName varchar(50) NOT NULL,
     email Email NOT NULL UNIQUE,
     birthDate TIMESTAMP WITH TIME zone NULL,
```

```
     /rating Int NULL,
     address varchar(50) NOT NULL,
     postalCode varchar(50) NOT NULL,
     balance Int NULL,
     normalUserID integer NOT NULL,
     Authenticated_UserID integer NULL,
     cityID integer NOT NULL
);

CREATE TABLE Notification (
     "date" TIMESTAMP WITH TIME zone NOT NULL,
     description varchar(50) NOT NULL,
     type varchar(50) NOT NULL,
     notificationID integer NOT NULL,
     auctionID integer NULL,
     Authenticated_UserID integer NOT NULL
);

CREATE TABLE pertains_to (
     categoryID integer NULL,
     auctionID integer NULL
);

CREATE TABLE Report (
     "date" TIMESTAMP WITH TIME zone NOT NULL,
     reason varchar(50) NOT NULL,
     reportID integer NOT NULL,
   auctionID integer NULL,
   NormalUserID integer NULL
);

CREATE TABLE Win (
     "date" TIMESTAMP WITH TIME zone NOT NULL,
     finalPrice Int NOT NULL,
     rate Int NULL,
     winID integer NOT NULL
);

CREATE TABLE Blocks (
```

```
          normalUserID integer NULL,
          authenticatedUserID integer NULL
);

CREATE TABLE Remove_Moderator (
    authenticatedUserID integer NULL,
          removeMod integer NULL
);

  CREATE TABLE Create_Moderator (
          authenticatedUserID integer,
          addedMod integer NULL
);

ALTER TABLE Add_Credits ADD CONSTRAINT PK_Add_Credits
 PRIMARY KEY (add_CreditsID);

ALTER TABLE Auction ADD CONSTRAINT PK_Auction
PRIMARY KEY (auctionID);

ALTER TABLE Authenticated_User ADD CONSTRAINT
PK_Authenticated_User
PRIMARY KEY (Authenticated_UserID);

ALTER TABLE Bid ADD CONSTRAINT PK_Bid
PRIMARY KEY (bidID);

ALTER TABLE Category ADD CONSTRAINT PK_Category
PRIMARY KEY (categoryID);

ALTER TABLE City ADD CONSTRAINT PK_City
PRIMARY KEY (cityID);

ALTER TABLE Comment ADD CONSTRAINT PK_Comment
PRIMARY KEY (commentID);

ALTER TABLE Country ADD CONSTRAINT PK_Country
PRIMARY KEY (countryID);
```

```
ALTER TABLE Normal_User ADD CONSTRAINT PK_Normal_User
PRIMARY KEY (normalUserID);

ALTER TABLE Notification ADD CONSTRAINT
PK_Notification
PRIMARY KEY (notificationID);

ALTER TABLE Report ADD CONSTRAINT PK_Report
PRIMARY KEY (reportID);

ALTER TABLE Win ADD CONSTRAINT PK_Win
PRIMARY KEY (winID);

ALTER TABLE Add_Credits ADD CONSTRAINT add_credits
FOREIGN KEY (normalUserID) REFERENCES Normal_User
(normalUserID);

ALTER TABLE Auction ADD CONSTRAINT win
FOREIGN KEY (normalUserID) REFERENCES Normal_User
(normalUserID);

ALTER TABLE Auction ADD CONSTRAINT create
FOREIGN KEY (normalUserID) REFERENCES Normal_User
(normalUserID);

ALTER TABLE Auction ADD CONSTRAINT accepts
FOREIGN KEY (Authenticated_UserID) REFERENCES
Authenticated_User (Authenticated_UserID);

ALTER TABLE Auction ADD CONSTRAINT rejects
FOREIGN KEY (Authenticated_UserID) REFERENCES
Authenticated_User (Authenticated_UserID);

ALTER TABLE Authenticated_User ADD CONSTRAINT add
FOREIGN KEY (Authenticated_UserID) REFERENCES
Authenticated_User (Authenticated_UserID);

ALTER TABLE Authenticated_User ADD CONSTRAINT remove
FOREIGN KEY (Authenticated_UserID) REFERENCES
```

```
Authenticated_User (Authenticated_UserID);

ALTER TABLE Bid ADD CONSTRAINT Auction
FOREIGN KEY (auctionID) REFERENCES Auction
(auctionID);

ALTER TABLE Bid ADD CONSTRAINT Normal_User
FOREIGN KEY (normalUserID) REFERENCES Normal_User
(normalUserID);

ALTER TABLE Category ADD CONSTRAINT has
FOREIGN KEY (categoryID) REFERENCES Category
(categoryID);

ALTER TABLE City ADD CONSTRAINT FK_City_Country
FOREIGN KEY (countryID) REFERENCES Country
(countryID);

ALTER TABLE Comment ADD CONSTRAINT to
FOREIGN KEY (auctionID) REFERENCES Auction
(auctionID);

ALTER TABLE Comment ADD CONSTRAINT removes
FOREIGN KEY (Authenticated_UserID) REFERENCES
Authenticated_User (Authenticated_UserID);

ALTER TABLE Comment ADD CONSTRAINT adds
FOREIGN KEY (normalUserID) REFERENCES Normal_User
(normalUserID);

ALTER TABLE Normal_User ADD CONSTRAINT
FK_Normal_User_Authenticated_User
FOREIGN KEY (normalUserID) REFERENCES
Authenticated_User (Authenticated_UserID);

ALTER TABLE Normal_User ADD CONSTRAINT blocks
FOREIGN KEY (Authenticated_UserID) REFERENCES
Authenticated_User (Authenticated_UserID);
```

```
ALTER TABLE Normal_User ADD CONSTRAINT
FK_Normal_User_City
FOREIGN KEY (cityID) REFERENCES City (cityID);

ALTER TABLE Notification ADD CONSTRAINT pertains_to
FOREIGN KEY (auctionID) REFERENCES Auction
(auctionID);

ALTER TABLE Notification ADD CONSTRAINT receives
FOREIGN KEY (Authenticated_UserID) REFERENCES
Authenticated_User (Authenticated_UserID);

ALTER TABLE pertains_to ADD CONSTRAINT Category
FOREIGN KEY (categoryID) REFERENCES Category
(categoryID);

ALTER TABLE pertains_to ADD CONSTRAINT Auction
FOREIGN KEY (auctionID) REFERENCES Auction
(auctionID);

ALTER TABLE Report ADD CONSTRAINT Auction
FOREIGN KEY (auctionID) REFERENCES Auction
(auctionID);

ALTER TABLE Report ADD CONSTRAINT Normal_User
FOREIGN KEY (normalUserID) REFERENCES Normal_User
(normalUserID);
```

GROUP1716, 18/03/2018

- Diogo Peixoto Pereira, diogopeixotopereira@gmail.com
- Igor Bernardo Amorim Silveira, igorasilveira@gmail.com
- Nádia de Sousa Varela de Carvalho, nadiacarvalho118@gmail.com
- Pedro Miguel Ferraz Nogueira da Silva, pedronogueirasilva5@gmail.com