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(ID **NN**, state **NN**, title **NN**, description, sellingReason, pathToPhoto, startingPrice **CK** startingPrice > 0, minimumSellingPrice, buyNow, startDate NN, limitDate NN, refusalDate, /numberOfBids, reasonOfRefusal, auctionWinner -> authenticated_User, auctionCreator -> authenticated_User **NN**, responsibleModerator -> authenticated_User, rate **CK** rate >= 0 && rate <= 5 || NULL, finalPrice, finalDate)
R02	authenticated_User(ID **NN**, typeOfUser **NN**, username **NN UK**, password **NN**, pathToPhoto, completeName, email **UK**, birthDate, /rating **CK** /rating >= 0 && /rating <= 5 , address, postalCode, balance **CK** balance >= 0, city -> city)
R03	add_Credits(addCreditsID **NN**, value **NN**, date **NN**, user -> authenticated_User **NN**, paypalID **NN**)
R04	bid (auctionBidded -> auction **NN**, bidder -> authenticated_User **NN**, date **NN**, value **NN**)
R05	category(categoryID **NN**, name **NN UK**, parent -> category)
R06	city(ID **NN**, name **NN UK**, country -> country **NN**)
	comment(ID **NN**, date **NN**, description **NN**,

R07	auctionCommented -> auction **NN**, moderatorThatErased -> authenticated_User , userCommenter -> authenticated_User **NN**)
R08	country(ID **NN**, name **NN UK**)
R09	notification(ID **NN**, date **NN**, description **NN**, type **NN**, auctionAssociated -> auction, authenticatedUserID -> authenticated_User **NN**)
R10	report(auctionID -> auction **NN**, normalUserID -> authenticated_User **NN**, date **NN**, reason **NN**)
R011	categoryOfAuction (auction -> auction **NN**, category -> Category **NN**)
R012	blocks (blocked -> authenticated_User **NN**, blocker -> authenticated_User **NN**, state **NN**, description **NN**, date **NN**)
R013	edit_Moderator(removedMod -> authenticated_User **NN**, removerAdmin -> authenticated_User **NN**)
R014	edit_Categories(category -> Category **NN**, admin-> authenticated_User **NN**)

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')
Blocking State	ENUM ('Blocked', 'Allowed')
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:{ ID }	
Functional Dependencies	
FD0101:	{ID} → {state, title, description, sellingReason, pathToPhoto, startingPrice, minimumSellingPrice, buyNow, startDate, limitDate, refusalDate, /numberOfBids, reasonOfRefusal, rate, finalPrice, finalDate, auctionWinner, auctionCreator, responsibleModerator}
Normal Form:	BCNF

Table R02 (authenticated_User)

Keys:{ ID }	
Functional Dependencies	
FD0201:	{ID} → {username, typeOfUser, password, pathToPhoto, completeName, email, birthDate, /rating, address, postalCode, balance, city}
Normal Form:	BCNF

Table R03 (add_Credits)

Keys:{ addCreditsID }
Functional Dependencies
FD0301: {addCreditsID} → {value, date, user, paypalID }
Normal Form: BCNF

Table R04 (bid)

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

Table R05 (category)

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

Table R06 (city)

Keys:{ ID }
Functional Dependencies
FD0601: { ID } → {name, country}
Normal Form: BCNF

Table R07 (comment)

Keys:{ commentID }
Functional Dependencies

FD0701:	{ ID } → {date, description, auctionCommented, moderatorThatErased, userCommenter}
Normal Form:	BCNF

Table R08 (country)

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

Table R09 (notification)

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

Table R10 (report)

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

Table R11 (categoryOfAuction)

Keys:	{ (auction,category) }
Functional Dependencies	

FD1301:	{(auction,category)} → {}
Normal Form:	BCNF

Table R12 (blocks)

Keys:{blocked}	
Functional Dependencies	
FD1401:	{blocked} → {blocker,state,description,date}
Normal Form:	BCNF

Table R13 (edit_Moderator)

Keys:{removedMod}	
Functional Dependencies	
FD1501:	{removedMod} → {removerAdmin}
Normal Form:	BCNF

Table R14 (edit_Categories)

Keys:{category}	
Functional Dependencies	
FD1601:	{category} → admin
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 Authenticated_User DROP CONSTRAINT
FK_Authenticated_User_Autheticated_User;
ALTER TABLE Authenticated_User DROP CONSTRAINT
blocks;
ALTER TABLE Authenticated_User DROP CONSTRAINT
FK_Authenticated_User_City;
ALTER TABLE Bid DROP CONSTRAINT Auction;
ALTER TABLE Bid DROP CONSTRAINT Authenticated_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 Notification DROP CONSTRAINT pertains_to;
ALTER TABLE Notification DROP CONSTRAINT receives;
ALTER TABLE CategoryOfAuction DROP CONSTRAINT
Category;
ALTER TABLE CategoryOfAuction DROP CONSTRAINT
Auction;
ALTER TABLE Report DROP CONSTRAINT Auction;
ALTER TABLE Report DROP CONSTRAINT
Authenticated_User;
ALTER TABLE Blocks DROP CONSTRAINT
Authenticated_User;
ALTER TABLE Blocks DROP CONSTRAINT
Authenticated_User;
ALTER TABLE Edit_Moderator DROP CONSTRAINT
Authenticated_User;
```

```
ALTER TABLE Edit_Moderator DROP CONSTRAINT
Authenticated_User;
ALTER TABLE Edit_Categories DROP CONSTRAINT Category;
ALTER TABLE Edit_Categories DROP CONSTRAINT
Authenticated_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 Authenticated_User;
DROP TABLE IF EXISTS Notification;
DROP TABLE IF EXISTS CategoryOfAuction;
DROP TABLE IF EXISTS Report;
DROP TABLE IF EXISTS Blocks;
DROP TABLE IF EXISTS Edit_Moderator;
DROP TABLE IF EXISTS Edit_Categories;

CREATE TYPE AuctionSate AS ENUM ('Active',
'Rejected', 'Pending');
CREATE TYPE BlockingState AS ENUM ('Blocked',
'Allowed');
CREATE TYPE TypeOfUser  AS ENUM ('Moderator',
'Administrator', 'Normal');

CREATE TABLE Add_Credits
(
        ID integer NOT NULL,
        value integer NOT NULL,
        date TIMESTAMP WITH TIME zone NOT NULL,
        paypalID integer NOT NULL,
        user INT NOT NULL
);

CREATE TABLE Auction
```

```
(
        ID integer NOT NULL,
        state AuctionSate NOT NULL,
        title varchar(50) NOT NULL,
        description varchar(50) NULL,
        sellingReason varchar(50) NULL,
        pathToPhoto varchar(50) NULL,
        startingPrice integer NOT NULL CHECK
(startingPrice > 0),
        minimumSellingPrice integer NULL CHECK
(minimumSellingPrice > 0),
        buyNow integer NULL CHECK (buyNow > 0),
        startDate TIMESTAMP WITH TIME zone NOT NULL,
        limitDate TIMESTAMP WITH TIME zone NOT NULL,
        refusalDate TIMESTAMP WITH TIME zone NULL,
        /numberOfBids integer NULL,
        reasonOfRefusal varchar(50) NULL,
  finalDate TIMESTAMP WITH TIME zone NULL,
  finalPrice integer NULL,
        rate integer NULL CHECK (rate >= 0 AND rate
<=5),
  auctionCreator integer NOT NULL,
  auctionWinner integer NULL,
        responsibleModerator integer NULL,
        CHECK ((minimumSellingPrice > startingPrice
OR minimumSellingPrice == NULL)
                                    AND
                                    (buyNow >
startingPrice OR buyNow == NULL)
                                   )
);

CREATE TABLE Authenticated_User
(
        ID integer NOT NULL,
        typeOfUser TypeOfUser NOT NULL,
        username varchar(50) NOT NULL UNIQUE,
        password varchar(50) NOT NULL,
        pathToPhoto varchar(50) NULL,
```

```
        completeName varchar(50) NOT NULL,
        email varchar(50) NULL UNIQUE,
        birthDate TIMESTAMP WITH TIME zone NULL,
        /rating integer NULL CHECK (/rating >= 0 AND
/rating<=5),
        address varchar(50) NULL,
        postalCode varchar(50) NULL,
        balance integer NULL CHECK (balance >= 0),
        city integer NULL
);

CREATE TABLE Bid
(
        date TIMESTAMP WITH TIME zone NOT NULL,
        value integer NOT NULL,
  auctionBidded integer NOT NULL,
  bidder integer NOT NULL
);

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

CREATE TABLE City
(
        ID integer NOT NULL,
        name varchar(50) NOT NULL UNIQUE,
        country integer NOT NULL
);

CREATE TABLE Comment
(
        ID integer NOT NULL,
        date TIMESTAMP WITH TIME zone NOT NULL,
        description varchar (50) NOT NULL,
        auctionCommented integer NOT NULL,
```

```
        moderatorThatErased integer NULL,
        userCommenter integer NOT NULL
);

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

CREATE TABLE Notification
(
        ID integer NOT NULL,
        date TIMESTAMP WITH TIME zone NOT NULL,
        description varchar(50) NOT NULL,
  type varchar(50) NOT NULL,
        auctionAssociated integer NULL,
        authenticated_UserID integer NOT NULL
);

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

CREATE TABLE CategoryOfAuction
(
        category integer NOT NULL,
        auction integer NOT NULL
);

CREATE TABLE Blocks
(
        state BlockingState NOT NULL,
        description varchar(50) NULL,
        date TIMESTAMP WITH TIME zone NOT NULL,
```

```
        blocked integer NULL,
        blocker integer NULL
);

CREATE TABLE Edit_Moderator
(
        removedMod integer NULL,
        removerAdmin integer NULL
);

CREATE TABLE Edit_Categories
(
        category integer NOT NULL,
        admin integer NOT NULL
);


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

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

ALTER TABLE Authenticated_User ADD CONSTRAINT
PK_Autheticated_User
        PRIMARY KEY (ID);

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

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

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

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

```
ALTER TABLE Notification ADD CONSTRAINT
PK_Notification
        PRIMARY KEY (ID);

ALTER TABLE Add_Credits ADD CONSTRAINT add_credits
        FOREIGN KEY (user) REFERENCES
Authenticated_User (ID);

ALTER TABLE Auction ADD CONSTRAINT win
        FOREIGN KEY (auctionWinner) REFERENCES
Authenticated_User (ID);

ALTER TABLE Auction ADD CONSTRAINT create
        FOREIGN KEY (auctionCreator) REFERENCES
Authenticated_User (ID);

ALTER TABLE Auction ADD CONSTRAINT accepts
        FOREIGN KEY (responsibleModerator) REFERENCES
Autheticated_User (authenticatedUserID);

ALTER TABLE Auction ADD CONSTRAINT rejects
        FOREIGN KEY (responsibleModerator) REFERENCES
Autheticated_User (authenticatedUserID);

ALTER TABLE Authenticated_User ADD CONSTRAINT blocks
        FOREIGN KEY (authenticatedUserID) REFERENCES
Autheticated_User (authenticatedUserID);

ALTER TABLE Authenticated_User ADD CONSTRAINT
FK_Authenticated_User_City
        FOREIGN KEY (cityID) REFERENCES City
(cityID);

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

ALTER TABLE Bid ADD CONSTRAINT Authenticated_User
        FOREIGN KEY (bidder) REFERENCES
```

```
Authenticated_User (ID);

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

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

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

ALTER TABLE Comment ADD CONSTRAINT removes
        FOREIGN KEY (moderatorThatErased) REFERENCES
Autheticated_User (authenticatedUserID);

ALTER TABLE Comment ADD CONSTRAINT adds
        FOREIGN KEY (userCommenter) REFERENCES
Authenticated_User (ID);

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

ALTER TABLE Notification ADD CONSTRAINT receives
        FOREIGN KEY (authenticatedUserID) REFERENCES
Autheticated_User (authenticatedUserID);

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

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

ALTER TABLE Report ADD CONSTRAINT Auction
```

```
        FOREIGN KEY (auctionID) REFERENCES Auction
(auctionID);

ALTER TABLE Report ADD CONSTRAINT Authenticated_User
        FOREIGN KEY (normalUserID) REFERENCES
Authenticated_User (ID);

ALTER TABLE Edit_Moderator ADD CONSTRAINT
Authenticated_User
        FOREIGN KEY (removedMod) REFERENCES
Authenticated_User (ID);

ALTER TABLE Edit_Moderator ADD CONSTRAINT
Authenticated_User
        FOREIGN KEY (removerAdmin) REFERENCES
Authenticated_User (ID);

ALTER TABLE Blocks ADD CONSTRAINT Authenticated_User
        FOREIGN KEY (blocked) REFERENCES
Authenticated_User (ID);

ALTER TABLE Blocks ADD CONSTRAINT Authenticated_User
        FOREIGN KEY (blocker) REFERENCES
Authenticated_User (ID);

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

ALTER TABLE Edit_Categories ADD CONSTRAINT
Authenticated_User
  FOREIGN KEY (admin) REFERENCES Authenticated_User
(ID);
```

Changes made to the first submission:

1. Changed some of the classes as suggested by out professor, including eliminating Normal_User and upgrading those elements

in that class to Authenticated_User.
2. Win of an Auction information went up to Auction instead of being its own class.
3. We did a revision of some of the functional dependencies.
4. We reviewed our sql so it was updated in according to our relational references.

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