

A sample project to the subject Database and information systems

Information system of auctions

(c) 2012-2019, version: 0.15, 20190325

Radim Bača, Peter Chovanec, Michal Krátký

Department of Computer Science, FEECS, VŠB – Technical University of Ostrava

<http://dbedu.cs.vsb.cz/>

General rules

- The goal of the analysis is to describe a future system, the goal **is not to describe as many pages as possible** (a good analysis can include 20 pages, a bad analysis 100 pages).
- A developer must be able to implement the system according to the analysis – **the analysis must be complete**.
- Analysis has not to include a description of trivial aspects of the system, e.g. trivial operations (**CreateReadUpdateDelete**) for codebooks, we must describe only complicated parts of the system in more detail, e.g. complicated functions.
- The same problem as an *incomplete analysis* is an analysis describing everything in the system – we call it *analysis-paralysis*. Why? We are not able to find complex (and important) features of the system.
- Functional analysis is as important as the data analysis. We are not able to implement the system without the it.
- Why? The number of functions **is not equal** to the number of tables $\times 4$ (since we have 4 database operations) or 5 (since we often use one select retrieving a record for the primary key and one select retrieving all records of a table); a system can include a number of other functions.
- Complicated functions are described by a tool like the minispecification, sequence UML diagram or data flow diagram (DFD). We select the best one for a concrete situation; however, it seems that doing the minispecification takes the lowest time.
- We define SQL commands of the functions in the analysis, as a result the analysis is more concrete and the implementation of such a function is possible without any doubt. Moreover, it helps us to tune data and physical design of the system.

- **Summary:** Two ways of poor analysis:
 - An analysis describing nothing.
 - An analysis describing everything.
- **A good way to start a functional analysis** is to design forms of the user interface, and, in such a way, to identify functions of the system.
- **A good way to design a function** is to minimize a number of database operations (e.g., to use one complex select instead of a number of selects).

1 Assignment Specification

WHY?

We need an information system for the management of electronic auctions in the company. The system should simplify the depreciation and sale of obsolete items for a reasonable price.

FOR WHAT?

An electronic auction system will solve the problem of selling obsolete items. The main aim is to create an auction information system and ensure smooth automatic bidding. Moreover, it will be possible to monitor auctions using an automatical notification and comment on individual auctions, if the user participated in them.

WHO?

The main role will be **auction administrator**, which will be the person in charge of managing and depreciating the assets in the company. In addition, the system can be used only by the users created by the administrator. The system recognizes two categories of users. The first category will be basic **customers** which will be people who do not necessarily have to work in the company. The second category will be **auctioneer**, which must be a person from the company and typically he will be a co-worker of auction administrator who will auction items from a particular category. In summary, we will call all roles in the system **user**.

INPUTS:

The system will mainly concern auctions and bids for them. In the case of the auction, we will be particularly interested in the name of the auction, its description, the beginning of the auction, the end of the auction, the category, the minimum bid and the auctioneer who created the auction. Each auction belongs to exactly one category. Only an auctioneer or auction administrator can create a new auction. Any auction can be updated or deleted by the auction administrator and the auctioneer (only own auctions).

In the case of the user, we will be interested in the user's login, name and surname, address, telephone number and date of the last visit in the system. The user can make many bids at the auction, where we record the bid and the date. The auctioneer cannot bid on his own auction. The user can also watch the selected auctions and can also add a comment to the auction in which he won.

OUTPUTS:

The main outputs that will be available to all registered users will include: (1) an auction detail with a list of bids at the auction and (2) a list of active (i.e. unfinished) auctions sorted by the end date. Furthermore, the user will be able to view his user data, the list of the observed auctions and the list of unfinished auctions to which he has bid. In the case of the auctioneer, everyone will be able to view a list of comments on the auctions created by him.

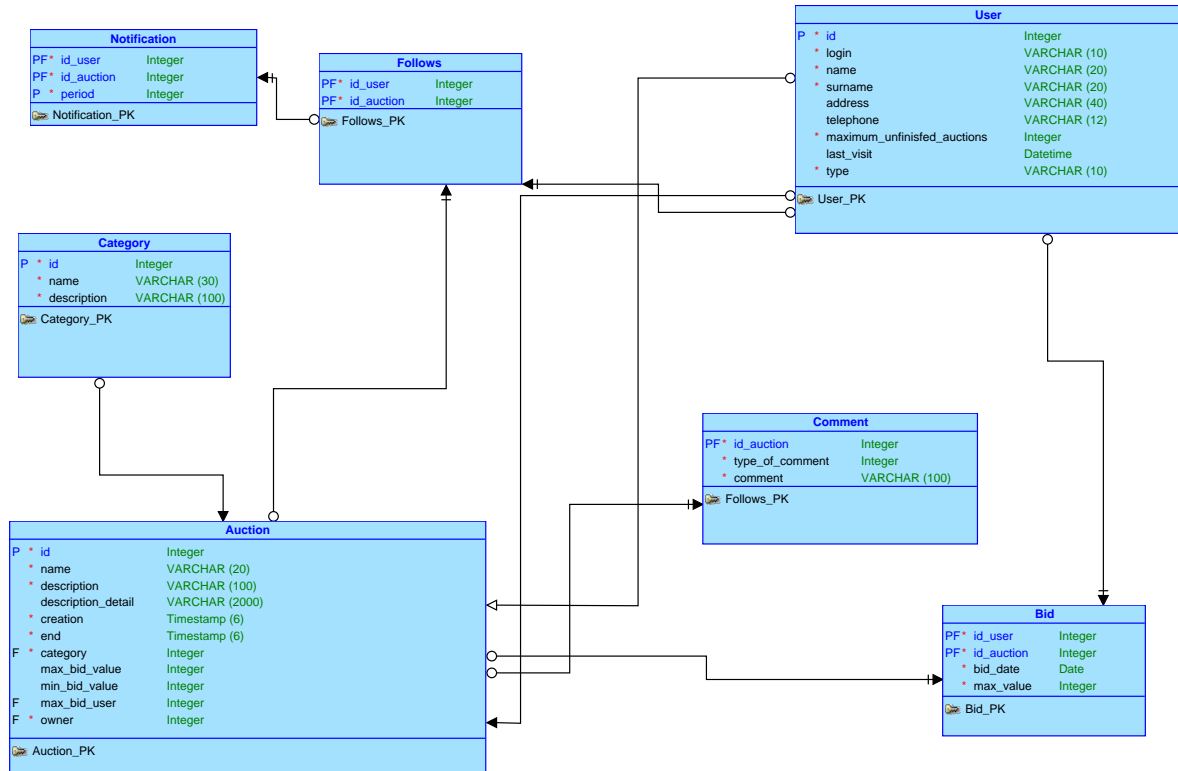
FUNCTIONS:

The auction administrator will be able to create and delete users in the system and will also be the only one in charge of category management. Each user can then update his data, bid and turn on/off the auction monitoring. Auctioneers can then also create a new auction. The system will not allow access to operations that are not allowed for the given user role.

The main function of the system will be to deal with bidding on auctions, which will be accessible to all users. In the case of a bid, it will be necessary to compare the current bid with the maximum of other bids. The current bid at the auction will always be equal to the second-highest bid + 10, and if there is only one bid at the auction, the current bid will be equal to the minimum required at the auction. In other words, the user with the highest bid does not pay his maximum offered price, but only ten crowns more than the maximum of the second-highest price.

2 Conceptual Model

ER Diagram



Entity Types

Legend: **Table**, Primary Key, *foreign key*, attribute

User(id, login, name, surname, address, telephone, type, maximum_unfinished_auctions, last_visit)

Auction(id, name, description, description_detail, creation, end, owner, category, id_user_max_bid, max_bid_value, min_bid_value)

Bid(id_user, id_auction, bid_date, max_value)

Comment(id_user, id_auction, type_of_comment, comment)

Follows(id_user, id_auction)

Category(id, name, description)

Notification(id_user, id_auction, period)

3 Data Models

Description of tables is depicted in the following tables.

Table **User**

	Data type	Length	Key	Null	Index	IC	Description
id	Int		Primary	N	A		Login of a user
login	Varchar	10		N			User name
name	Varchar	20		N			Last name of the user
surname	Varchar	20		N			Street and city of the user
address	Varchar	40		A			
telephone	Number	12		A			
type	Varchar	10		N		1	User type
maximum_unfinished_auctions	Integer			N		2	The maximum auctions owned by the user at once
last_visit	Timestamp			A			Timerstamp of the last visit of the system

Table Auction

	Data type	Length	Key	Null	Index	IC	Description
id	Int		Primary	N	A		
name	Varchar	20		N			Auction name
description	Varchar	100		N			A short description
description_detail	Varchar	2000		A			A long description
creation	Timestamp			N		3, 4	Date of creation
end	Timestamp			N		3, 4	Auction finish date
owner	Int		FK (User)	N			Bidder
category	Int		FK (Category)	N			Category of the auction
max_bid_user	Int		FK (User)	N			Who has the highest bid for the auction
max_bid_value	Int			N			The value of the highest bid of the auction
min_bid_value	Int			N			The minimal value of bid for the auction

Table Bid

	Data type	Length	Key	Null	Index	IC	Description
id_user	Int		PK, FK (User)	N	A		
id_auction	Int		PK, FK (Auction)	N	A		
bid_date	Timestamp					5	Bid date
max_value	Int						Maximal value of the bid

Table Notification

	Data type	Length	Key	Null	Index	IC	Description
id_user	Int		PK, FK (User)	N	A		
id_auction	Int		PK, FK (Auction)	N	A		
period	Int	Primary	N	A			The number of hours before the end of the auction, when the user has been notified

Table Comment

	Data type	Length	Key	Null	Index	IC	Description
id_auction	Int		PK, FK (Auction)	N	A		
type_of_comment	Int						
comment	Varchar	100					Content of the comment

Table Follows

	Data type	Length	Key	Null	Index	IC	Description
id_user	Int		PK, FK (User)	N	A		
id_auction	Int		PK, FK (Auction)	N	A		

Table Category

	Data type	Length	Key	Null	Index	IC	Description
id	Int		Primary	N	A		
name	Varchar	30		N			
description	Varchar	100		N			The description of the auctions in the specified category

Integrity Constraints:

1. Type: admin, auctioneer or user.
2. maximum_unfinished_auctions < 100.
3. creation < end.
4. end - creation < 21 days.
5. bid_date < #auction.end, where #auction is the auction defined by id_auction (i.e. the bid auction).