

# Open Information System: Conceptual schema

Thomas Perale – 0546990  
Maximilien Romain – 0543411  
Felipe Rojas – 0542569  
Lehal Sherik – 0543118

27 October 2017

## 1 Project subject

Our OWL have seven classes, Category, Ebook, Purchase, Publisher, Author, User, Publisher and Role.

The *User* class represent both an admin or a customer of the platfrom, it have seven attributes:

- userId
- information
- name
- email
- password
- lastName

A *User* of the system can either have the *AdminRole* or just have the *ClientRole*, relation like *hasAdminRole* or *hasClientRole* from the *User* domain in the *AdminRole/ClientRole* range. Both *AdminRole/ClientRole* are subclasses of *Role* they are used to be more specific what permission a *User* have but are characterized by the attributes:

- roleId
- description

Having one of these role imply the *User* can interract with the *Ebook* has shown with the relations *hasPurchased* and *manage*. *Ebook* are the main component of the platform because it's simply what we sell on it. They are defined by:

- ISBN
- title

- year
- version

But also a remote *Author* class bound to *Ebook* by the *hasWritten* class who have got the following attributes:

- authorId
- firstName
- lastName

Also defined with the *Publisher* class bound from *Ebook* by the relation *isPublishedBy*, it is made of:

- publisherId
- name

And finally *Category* class bound by *Ebook* by the *hasCategory* relation, categories are just a way to do search on ebooks by genre so the *Category* class is defined by:

- categoryId
- description

To enable customers to read ebooks they must first create *Purchase* as you can see in our ontology a *User* make by the *isMaking* relation *Purchase*. *Purchase* are a way to modelize ebooks the user paid to have for each *Purchase* the user have to make a transaction and they can contain more than one *Ebook* has we can see with the *isPartOf* relation.

Other privilege a customer have is to rate the *Ebook* he bought modeled by the *Rating* class. *User* and *Rating* are bound together by the *hasRated* relation and make *Ebook* in relation with it with *hasRating*. The attributes for *Rating* are:

- ratingId
- number

## 2 Rules description

First rule, if the category of a book is also a subCategory of another category, it implies that the book belongs to both category and subcategory.

$Ebook(?x), Category(?y), Category(?z), subCategoryOf(?y, ?z), hasCategory(?x, ?y), differentFrom(?y, ?z) \rightarrow hasCategory(?x, ?z)$

Second rule, if an user has Purchased an ebook, and ebook is part of a purchase, then we infer that a user “HasPurchased” an ebook.

$User(?x), Purchase(?y), Ebook(?z), isMaking(?x, ?y), isPartOf(?z, ?y) \rightarrow hasPurchased(?x, ?z)$

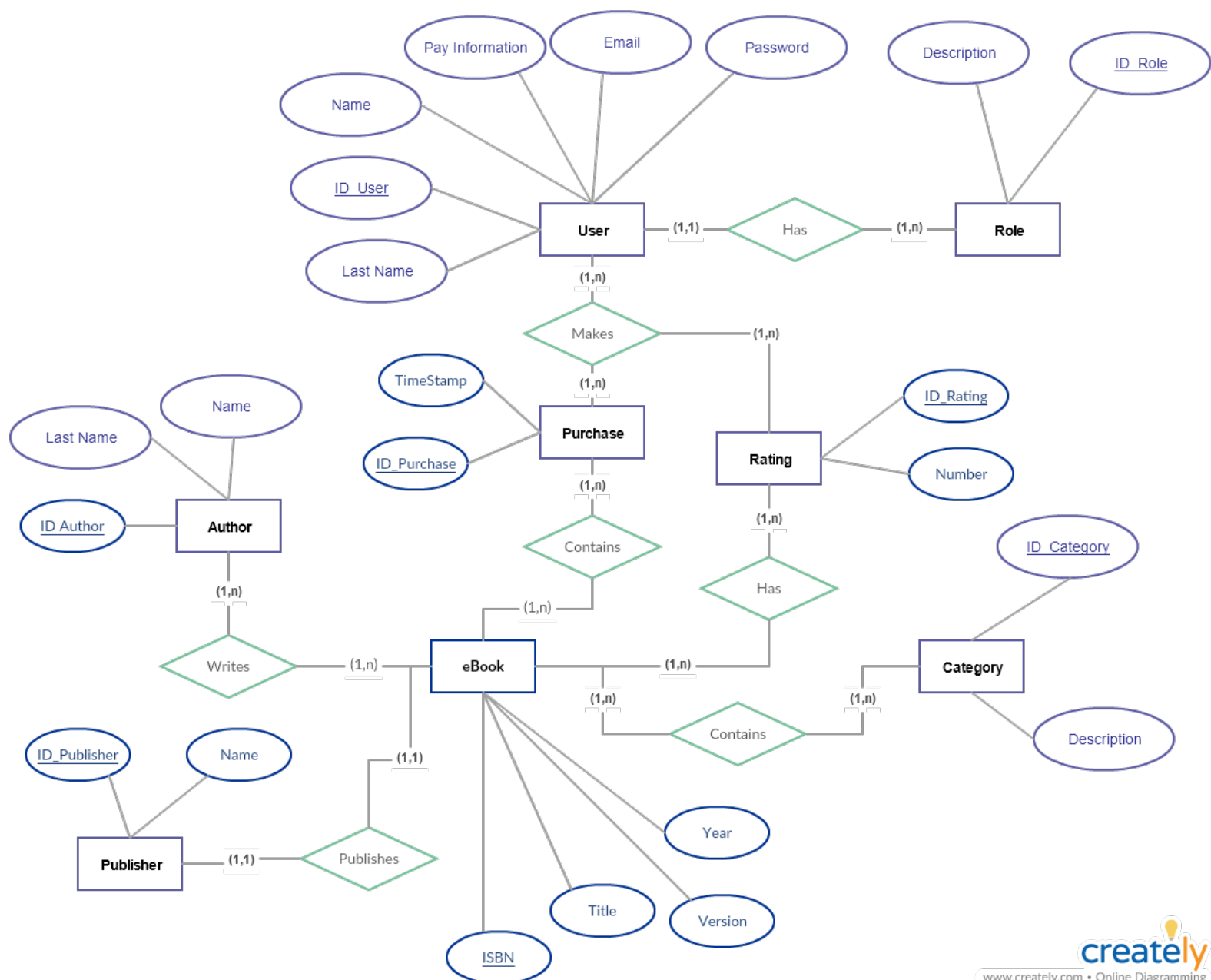
Third rule, if a user has rated an ebook and an ebook has been rating, infers that user purchased the book.

$User(?x), Rating(?y), Ebook(?z), hasRating(?x, ?y), hasRating(?z, ?y) \rightarrow hasPurchased(?x, ?z)$

This last rule allows to infer that if an user has the role admin, that user can manage the ebooks on the system

$User(?admin), Ebook(?book), AdminRole(?role), hasAdminRole(?admin, ?role) \rightarrow manage(?admin, ?book)$

### 3 ER Schema



## 4 WebOwl visualisation

