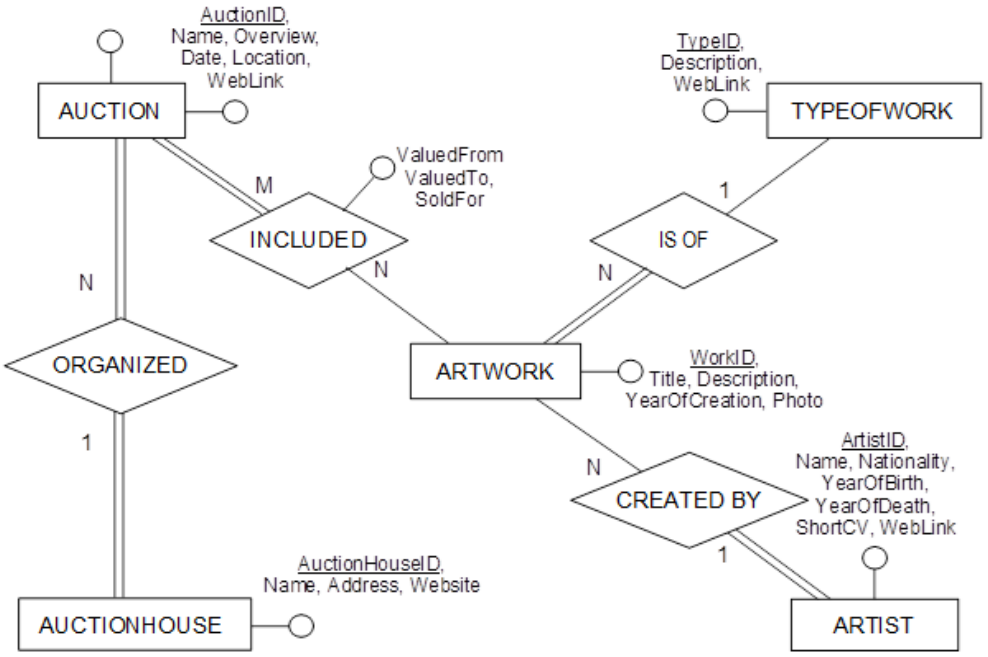
The AUCTIONS case study

The Auctions auction system utilizes a database that records the history of art auctions by auction houses. In addition to the details of the auctions held and the auction houses that organized them, the database records details of the artworks, their artists-creators, and the selling price of the artworks in the recorded auctions. More information about the process followed in art auctions can be found on the Internet, for example at the following link to the online encyclopedia Wikipedia: https://en.wikipedia.org/wiki/Art_auction.

The Auctions database was designed in a way that is summarized in the Entity-Relationship Diagram (ERD) shown below.



Assignment 1 – Functional dependencies and relational schema

Exercise 1 refers to identifying functional dependencies between the attributes of the database. Specifically, you are asked to reason about and document the functional dependency (X → Y) that exists between the attributes:

AuctionID, ArtWorkID, ValuedFrom, ValuedTo, SoldFor

This should be approached by illustrating and proving that X (a subset of the mentioned attributes) is a candidate key of a universal relation involving all mentioned attributes.

Then, in the area of the course in eclass you will find the file 'Auctions-No-Constraints.zip' which contains the script defining the tables AUCTIONHOUSE, AUCTION, ARTIST, TYPEOFWORK and ARTWORK, using PostgreSQL for your convenience, but without specifying primary keys

and referential integrity constraints. First, based on the script given to you, the ER diagram above as well as what you know about converting ER models to relational schema, you should implement the required modifications to the database so that the revised script is error-free with the data that you will find in the file 'Auctions-Data.zip' in eclass. Next, you should enrich the database script by creating the INCLUDED table of the database by writing the necessary command in SQL language and enter the data that you will find in the file 'Auctions-Data-II.zip' in eclass. Attention: the name of the table and its fields should be given exactly as they are presented, without any changes. To create/define the table, you will need to decide the type of each field, according to the data that needs to be entered (see file 'Auctions-Data-II.zip'.)

Assignment 2 – Schema improvements and new data types

Having arrived at a script that executes successfully in the current exercise, you will work on relational schema improvements using new data types. The task consists of two parts. In the first part, you will create a new data type called FYLLO and define it as an ENUM with values 'Male' and 'Female'. Then (a) appropriately modify the definition and the snapshot of the ARTIST table so that the gender of each artist is determined by an attribute of type FYLLO (without adding or removing data from the table and (b) formulate the following queries in SQL:

1. Find the living artists {ArtistName, BirthYear} who are male.
2. Find the artists {ArtistName, BirthYear} who are female.
3. Find for each female artist the total number of different works that have been sold at auctions {ArtistCode, ArtistName, TotalNumberOfWorks}. Print the results in descending order of total number of works.

The second part concerns the modification of the relational schema and the database data so as to create a new complex data type titled ADDRESS that is defined by the following STREET, POLH and CODE, (all of type VARCHAR (60)). In addition, you should define a multivalued attribute that will record landline and mobile phone (in any order you want). Then appropriately modify the definition and the snapshot of the AUCTIONHOUSE table so that the address is defined by a new attribute of type ADDRESS. Check the changes by formulating the queries in SQL that calculate:

4. Find the AUCTIONHOUSE {AUCTIONHOUSEID, name, website} that state an address in New York City.
5. Find the mobile phone number of the person in charge of AUCTIONHOUSE that states a landline number of 2810379190. In your answer, include the name of AUCTIONHOUSE and the mobile phone number of the person in charge

Assignment 3 – Specialization hierarchies, JSON and XML

So far, you have experimented with the new possibilities offered by enriching data types and creating data types for some of the uses that PostgreSQL supports. In particular, you have examined complex types (e.g. ADDRESS), enumerated types, multi-valued types, and the combination of these to define user-defined data types. The current section is dedicated to handling

special problems and data such as specialization hierarchies and inheritance. Specifically, you will need to implement a number of tasks.

First, you should modify the existing schema to create a hierarchy of specialization for auction participants. In particular, participants (PARTICIPANT) need to be distinguished into experts (EXPERT) and novices (NOVICE). The two tables should inherit from the PARTICIPANT table. The EXPERT table will also have the LIST_OF_TYPES attributes of type ARRAY with values from the types of works present in the TYPEOFWORK table. The NOVICE table will also have the DATE_OF_BIRTH attribute of type DATE which records the date of birth as well as the TypeID attribute which will receive values from the types of works present in the TYPEOFWORK table. Secondly, you should import / modify your data set to comply / conform to the new schema. You will need to import at least one EXPERT and at least one NOVICE participant in an auction. Finaly, after completing the above, formulate the SQL statements for the queries:

1. Print the types of works for which the expert has expertise with a nickname of your choice
2. Print the nicknames of novices (NOVICE)
3. Print the nickname, first name and last name of all EXPERT performances
4. Print all the details of those who are EXPERT
5. Print the performances of type NOVICE
6. Print the nickname, date of birth and exhibit type of the novices of the database
7. Print the performances from the EXPERT table that specialize in more than one exhibit type
8. Print the nicknames and content of the messages submitted by experts for all messages that do not include a reference to the work with code 'AW22'
9. Print the nicknames of EXPERTS (only) and the contents of the messages they posted (on any wall) which refer to the project with the code of your choice (which also appears in a message from NOVICE)
10. Print the codes and contents of the messages referring to the project with the code of your choice and which (messages) were posted by a NOVICE on the wall of an EXPERT

Assignment 4 – Graphs in Postgres and path queries

Assignment 4 has a dual objective, which concerns on the one hand the familiarization of students with graph development and on the other hand the analysis of alternative ways of representing graphs with relational technology. Progressively, this analysis will allow the understanding and practice of some advanced techniques of PostgreSQL such as the recursive operation for traversal and transitive closure of graphs. The students are still working on the current implementation of the AUCTIONS database where the following will need to be implemented:

Creating a property graph and implementing it in PostgreSQL

You will start by developing (initially on paper or in a drawing or slide tool such as PowerPoint) a property graph (see theory) that will capture the following:

- Follows relationships between collectors that you have already implemented (i.e. who follows whom, in the sense that a follow is found in the Twitter service)
- Interest relationships between collectors and auction items (note that this relationship is reflected in the current version of your implementation with the tags in the content of the messages that collectors post on a wall).
- For collectors registered in an auction, let's say 'AU01', the messages they post, the wall they post them on, the content of the message (except for the exhibits to which these messages refer, since this is question b above).
- The final version of the property graph that will be developed must definitely reflect the data that exists in your database at the time of submitting Part C and any new ones that you may wish to add

Having designed the indicative property graph (with a small number of nodes so that at least two representations of each node are recorded), you should then represent the property graph using relational technology. It is at your discretion to choose a technique from those you have learned so far, namely the use of a normalized relation and an unnormalized relation, while you should study the way in which the alternative nodes and edges that will exist in your graph will be supported (see PostgreSQL inheritance mechanism).

Recursive path queries

To become familiar with graph traversal using recursive SQL you should devise SQL statements to answer the following questions:

1. For each / one node calculate the next ones (in a direction of your choice)
2. For each / one node calculate the previous ones (in a direction of your choice)
3. For each / one node calculate the next ones regardless of direction
4. For each / one node calculate the previous ones regardless of direction
5. For two collectors participating in the same auction of your choice, calculate all the exhibits of common interest that are exhibited in the auction

Assignment 5 - NoSQL

The last assignment requires you to choose a NoSQL system of your choice and re-examine (a subset of) the Auctions database in order to gain hands-on experience with a system of a different type and philosophy than the one you used to develop the database. Your goal should not be to create yet another different implementation of the database in another computing environment (hence the reference to the subset above) but to migrate a portion of your data to a NoSQL environment in order to create a new data source complementary to PostgreSQL. The data set you choose should be realistic and reflect real-world conditions. For example, it could be a set of reviews and ratings for Apps from an open data set. Students who can present an illustrative

scenario of utilizing the data from all the sources they create will receive an additional bonus during the final grading.