Преглед НЦД 10 (2007), 47–58

Goran Gavrilović
(National Museum in Belgrade)

MUSEUM INFORMATION SYSTEM OF SERBIA RECENT APPROACH TO DATABASE MODELING

Abstract: The paper offers an illustration of the main parameters for museum database projection (case study of Integrated Museum Information System of Serbia). The simple case of museum data model development and implementation was described. The main aim is to present the advantages of ORM (Object Role Modeling) methodology by using Microsoft *Visio* as an eligible programme support in formalization of museum business rules.
Keywords: museum database modeling, Museum Information System, Serbia, National Museum in Belgrade

One of the consequences of fast growing world economy in the second part of the 20th century is the change of traditional way of business systems management (derived from the traditional economic theory by Adam Smith). New insights, innovative management methods and techniques, different way of thinking and completely new business philosophy are covered by the term *reengineering* (Hammer and Champy 1993). The outstanding progress in information technologies and widespread use of them, especially through last decades, influenced more innovations and recent changes in business systems management. Today, one can hardly find the field of up-to-date activity without use of modern communication technology, computers and accessory equipment.

Named methods of business activities and new information technologies are applicable in functioning of modern museums too, and they can be used in different ways. One of the core museum activities is the scientific validation and the treatment of objects that are kept there. Generally speaking, an object that is kept in museum is a source of many scientific and historical data, described and presented by museum experts in traditional ways, in form of classic documents, but up-to-date methods with reliance on computer techniques and database are used more and more.

The development of global communications, primarily Internet, makes the change of information without the problems of long distances and the factor of time. The precondition is to prepare the data to be comparable. Therefore many international (global) agreements, conventions and standards are established.

One of these standards is Conceptual Reference Model (CRM), which has been developed by a working group for standard documents included in activities of CIDOC (Comité International pour la Documentation) – one of twenty ICOM (the International Council for Museums) committees. The main task of CRM is to create the frame for information exchange on – ISO standard TC46. CIDOC's members are primarily museum professionals working in the field of cultural heritage information and technology.

The use of that standard is one of the basic and starting parameters in creating museum database.

Further, equally important parameter is to define the scope for database. It is determined by the answers to the following question: is it better to develop one unified database with universal physical structure for business systems of the same type (in named case – *the museums*, where identical business rules are applicable, on the whole territory of Serbia), or to make it separately, for individual institutions or even their organizational parts? In practice, both ways are used, although the development of unified database is more efficient. As mentioned before, a museum object is the source of scientific data and the premise for various scientific researches, so the necessity for precise data formation is evident. The accuracy is important, having in mind the structure of database, as well as the efficiency of database usage. So, the unified database, i.e. the cohesive (integrated) information system is the best solution.[1]

Building of integrated museum database in Serbia started in 1996. when the Strategic Study of MISS (Museum Information System of Serbia) development was formed. The IBM BSP (Business System Planning) methodology has been used for the analyses of museum business system. The Study defined basic elements of the Information System structure development: constituents, then main and supporting resources, as well as development mode. All business processes are identified as well as data classes, and their relationships through – processes (the BSP graphical pattern). Through the logical ties of business processes, several subsystems are evident:

0100 Documentation and information center
0200 Museum fund acquisition
0300 Museum fund storage
0400 Scientific research
0500 Museum fund presentation
0600 Museum fund usage
0700 Museum library
0800 Publishing
0900 Technical preservation
1000 Cultural and educational activities
1100 Museum archive

The Study presents a base for implementing following MISS activities:

1. Preparing main projects
2. Preparing operational projects
3. Programming and implementation

---

[1] One unfavorable example from the practice illustrates the need to choose the concept of unified museum database development as the best solution. There are different kinds of museum objects: archeological, historical, ethnological, technical etc. The diversification is made on the basis of attributes that are specific only for one group of museum objects. Business requirements for a group of museum experts operating with special group of museum objects enable model and database development by neglecting mutual attributes with other kinds of museum objects. That is the way of developing different programs as: numismatic, ethnological, technical etc. Let's imagine a hypothetical situation in so-called *complex type of museum*, storing different kinds of museum objects with its own special programme/s/; for example, one wants to know which objects entered our museum in 1993. Obviously, we have to put the same question as many times as the databases or programs we have, including an additional processing of all answers, in order to produce an integral form. The problem is even more complex if we want to involve further museums from different territories! This is possible only with making integral database.

The basic tasks of these activities are: precise physical database structure definition, database generation, as well as creating of one or more user applications in a scope of module, i. e. implemented subsystem.

One of the possible scenarios to execute this part of job is to be presented in the following text. Practical application was based on the Microsoft *Visio* program. Personal author's experience was derived from the Microsoft Official Course 2090A, Modeling Business Requirements to Create Database Using Microsoft Visual Studio .NET Enterprise Architect. In order to understand logic and advantages of the engine of Visio, several basic definition for applying ORM methodology will be directly cited from corresponding Course 2090A textbook (Microsoft 2002)[2].

There are few actual methodologies that are simplified process of modeling data requirements for databases:

Entity Relationship (ER) modeling – proceeded from 1976 and is still the most widely used approach for data modeling. It pictures the world (part of reality which is the object of modeling) as relationships between entities which have own attributes. ER often requires from modeler to make decision about the relative importance of attributes, specially in early phases of modeling process.

Unified Modeling Language (UML) – object-oriented methodology for designing code for object oriented programs, but also can be used for modeling data. It encapsulates both data and behavior within the object. The main disadvantage is disability for modeling all business rules and constraints.

Object Role Modeling (ORM) – is fact-oriented methodology that views the world in terms of objects playing roles in relationships between objects. It began in 1970s. It is rich modeling methodology for modeling very complex data-related business requirements.

By its complexity, ORM presents excellent base for setting up museum database in scope of MISS and will be shortly presented furthermore.

ORM Description

Generally, ORM methodology consists of two phases:

Modeling Business requirements

Process of collecting, analyzing data and modeling database is starting by researching business requirements in language of people working in business system. In this initial phase technical terms are not used. We can divide this phase in two parts:

External modeling phase – obtain business information that exists in the real world within the system, in our case – inside the museums. This very information is managed through classical medium, such as museum written documentation, inventory books, reports, catalogues, etc.

---

[2] All figures used in the following text were grabbed from the Microsoft *Visio* software, in order to demonstrate important points in the course of building the project example.

Conceptual modeling phase – conceptual model describes business requirements with all details and constraints from the perspective of work on concrete jobs, but with no details about implementation.

Database modeling

The relational database model is created after collecting business requirements by conceptual model. There are also two steps in this phase:

- Creating logical database model
- Creating physical database model

During the modeling process, the focus of works is moved from one job or step to another, just like constituents, too. Thus during the external and conceptual phases the brunt is on modeler work with experts from the business system. Later, in logical phase, modeler is more oriented to co-operation with developers (computer science specialist). Finally, detailed physical structure of database is completely defined in physical phase, depending upon DBMS choice.

ORM is used for modeling business requirements in conceptual phase.

ER modeling is used for generation of logical database model.

For physical definition of database we can use Transact SQL DDL (Data Definition Language), if we want to create SQL database.

Domain experts are the people who provide business requirements to the modeler:

- **Subject matter experts (SMEs)** – Understand the area of the business and its data and data-related business requirements. (In the case of museums – heads of different museum sectors, such as restoration sector, documentation sector, etc...)
- **Knowledge workers** – Have direct knowledge about any kind of job, why data is recorded and used. (In the case of museums – painters/icon conservators, art historians...)
- **Business analysts** – See the whole picture of business that includes many SMEs and knowledge workers (such as museum advisers).
- **System architects** – Understand how front-end and back-end IT system communicate data.
- **Developers** – Implement business logic through developing application.

Conceptual phase

Through few steps of making conceptual model, the main aim of this initial phase of modeling is to formalize completely business requirements that we are interested in. All aspects of reality that we are going to model are covered by the term **UoD (Universe of Discourse)** and represent the area of modeler speaking with business analysts. The critical point in whole projecting database is precise and explicit description of data that are objects of modeling. Therefore the main task of ORM is full transposition of business requirements into a model.

Business Managers have key role and responsibility in decision making for UoD. Domain expert have knowledge about concrete job, therefore they are only persons who can determinate UoD and its scope.

We can distinguish two types of models, one open, and second – close. If it is possible to expand UoD, the model is opened, vice versa the model is closed. It should be emphasized that in our case, museum data models are open!

The best way for understanding UoD is to understand data examples which are used in. Data Use Case is snapshot of data values and their relationships inside UoD. As mentioned before, examples of Use Case are placed in traditional museum documentation.

Use Case is provided by Domain Experts and it totally represents business facts and requirements. Those are two main characteristics of Use Case.

The aim of this text is not to explain ORM completely, but to present main parameters for creating museum database and, especially, basic advantages in using ORM in conceptual phase of modeling. Therefore we must understand few further fundamental definitions which will help us to understand Relational Databases concept, as well as logic of ORM functioning:

An *Instance* is a single example of a thing of interest in UoD. On practical level it is single data value or row in table.

Population is combination of instances of given type of interest in UoD. In database terms, all rows in table make up the table’s population.

A *Set* is group of instances, but is not necessarily the same as population. A set could be part of population, or combination of instances from more populations.

Member – each instance of set or population is member of that set or population.

Collection is the other term of *Set* but can includes duplicates of instances.

Domain experts provide information about relationships between instance group of data. Minimum and maximum number of occurrences of a given instance in relationships between two populations is determined by Cardinality. There are three types:

- one-to-one (1:1) – For each instance for one population, there is exactly one instance in another population.
- one-to-many (1:n) – For each instance for one population, there may be more than one duplicate instance in another.
- many-to-many (n:m) – Each instance in one population may be a duplicate, and there may be more than one correlated duplicate instance in another.

Mandatory cardinality implies that at least one instance is involved on a given side of relationship.

Optional cardinality implies that instances could exist in one population that does not have correlated instances in another population.

A population is **functionally dependent** from another if, for every instance in the second population, there is exactly one instance in the first population.

Functional dependencie has two key characteristics:

- Represent a parent-child relationship between two populations. Child instance can not exist without parent instance
- Instances in parent population are unique

In the text below we will see the short description of CSDP (Conceptual Scheme Design Process) which consists of seven steps.

In initial phase, modeler collects and analyzes external data and mark of components for defining basic elements that are needed for creating conceptual model, such as **fact instance, object types, predicates and fact types**.

Fact instance is separate observation about relationships between two or more data. One simplified example from museum practice will illustrate the main phases of database creation. Fact instance in our example is:

The museum object named ring ***is made*** *of materials named gold and silver.*

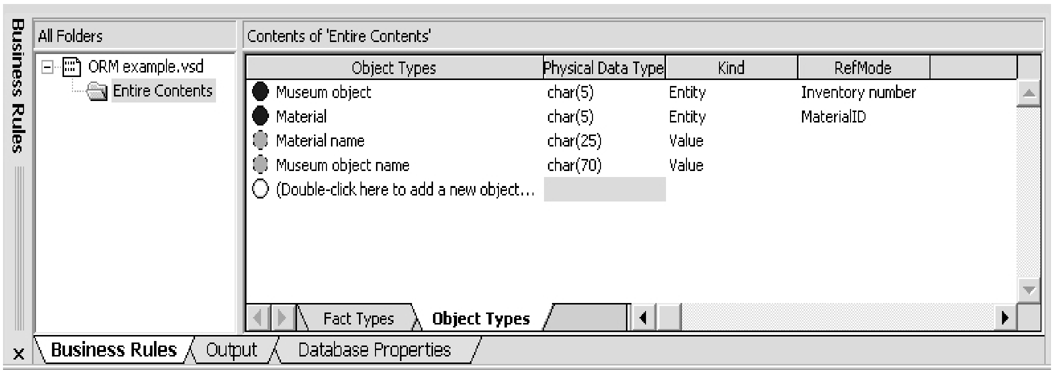
Object types represent set of all instances of given object. In our example they are: museum object, material, name of museum object and name of material. Basic division of object types is on *entities* and *value types*. In our example entity types are *Museum object* and *Material,* and value types are *Name of Museum object* and *Name of Material*.

We also determine the data that uniquely identify each instance in the population.

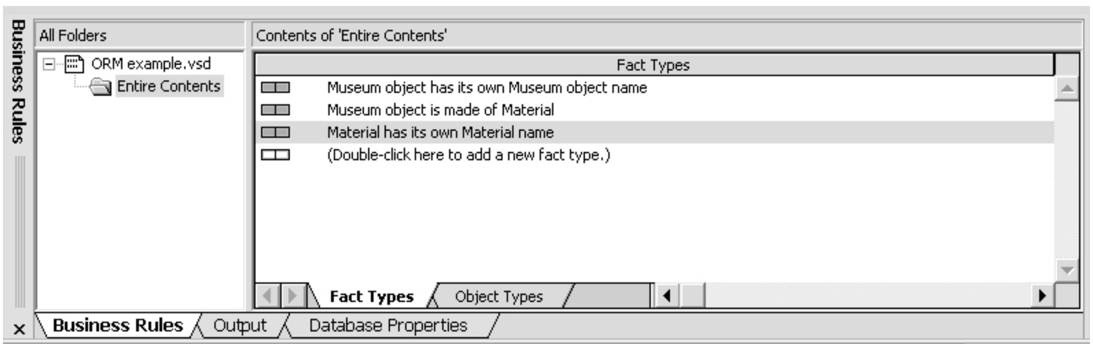
We will accept that unique identifier for Museum object is Inventory Number and for Material is MaterialId.

Predicates are phrases that modeler uses to describe relationships between object types. They consist of roles that object types have in concrete relationship.

The creation of ORM source model is starting by entering identified object types. We can see how it looks in *Visio*:

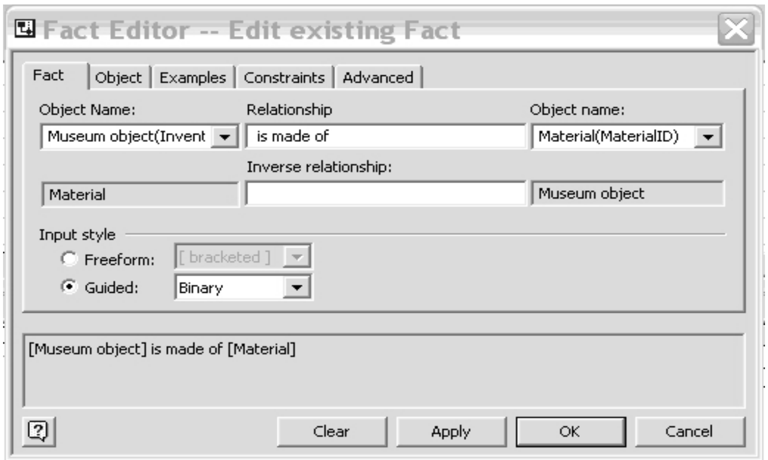


Entities are symbolized by full black and value types by gray circles. *Fact types*, that represent relationships between evidenced objects are also entered in this phase:

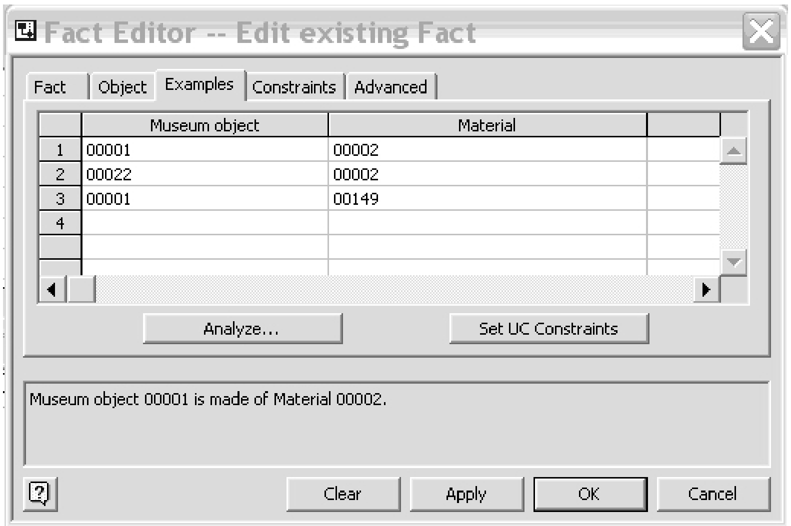


Small rectangulars on the left side of Fact Types list symbolize predicates consisting of two roles (Binary predicates) for each fact type. The number of roles in fact type called *Arity* may be larger than two, depending on the number of object types included in given fact type.

Separate fact types are entered by using Fact Editor:



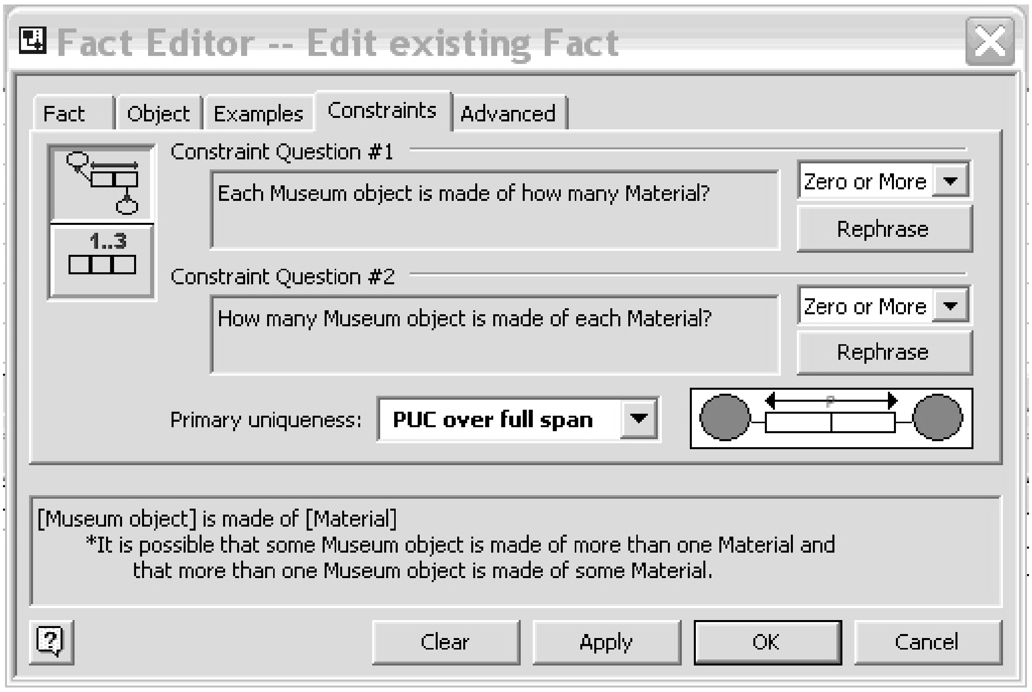
We are also entering example data in Fact Editor:



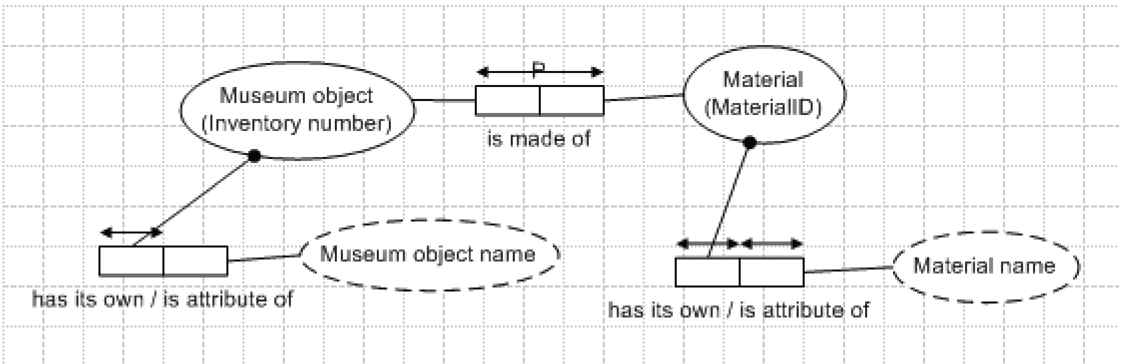
Example data serve to illustrate precisely specific relation between entities. The number of entered examples must be for one larger than arity of predicate.

It enables enveloping of all possible relationships between Fact Types in concrete Fact Instance. Thus, we can notice that in our example Museum object *0001* is made of two materials: *00002* and *00149*. At the same time, two museum objects, *00001* and *00022*, are made of the same material, *00002*. It means that one Museum object may be made of one or more materials, as well as more than one museum object may be made of one material (many-to-many relationship).

Implementation of this constraint is made in Constraints Tab in Fact Editor. Visio verbalized this relationship based on our answer on question #1 and question #2, as we can see in the bottom of window shown below. Also, for the purpose of this example, uniqueness over both roles in predicate is accented with Primary uniqueness (PUC over full span). Practically, it means that in final database inside the table evidencing materials for each Museum Object, combination of data from both columns (with primary keys from both tables) will be unique.



We create graphical scheme of ORM source model by dragging defined fact instances from evidence list to the drawing area:



There are several definitions and tunings in the next steps of creation conceptual model.

CSDP steps from 4 to 7 include implementation of different types of constraints: Internal and External uniqueness constraint, Mandatory role constraint, Value and Set constraints, Frequency and Ring constraints.

At any time of model developing process, we can control our model by *Model Error Checking.* In Output Window, *Visio* will inform us about every anomaly in forms of *Warnings* or *Errors*.

All business rules and constraints evidenced in this phase will be implemented in final database.

Formalization of UoD by using ORM enables very fine tuning of model, especially on level of implementing Primitive Fact Types.

Generating a Relational Logical Model

ER is the basic method in this phase of development. The logical data model is the presentation of the conceptual business data requirements by using tables, columns and cross-table relationships. The components are: tables, columns, primary keys and relationships.

Tables represent the existence of things of interest and make internal structure of future database. They mast have unique names. Rows of data must be unique, too. In our example we can easily notice tables as “Material”, “Museum objects” and “Museum Object is made of material”. Each of these tables will became a physically separate table in final database.

Columns represent a division of data inside table. The names of columns must be unique.

The unique identifier of each row in table is Primary key. To simplify our example, we argue that primary key of museum object is its inventory number (true, but only in a scope of collection to which object belongs).

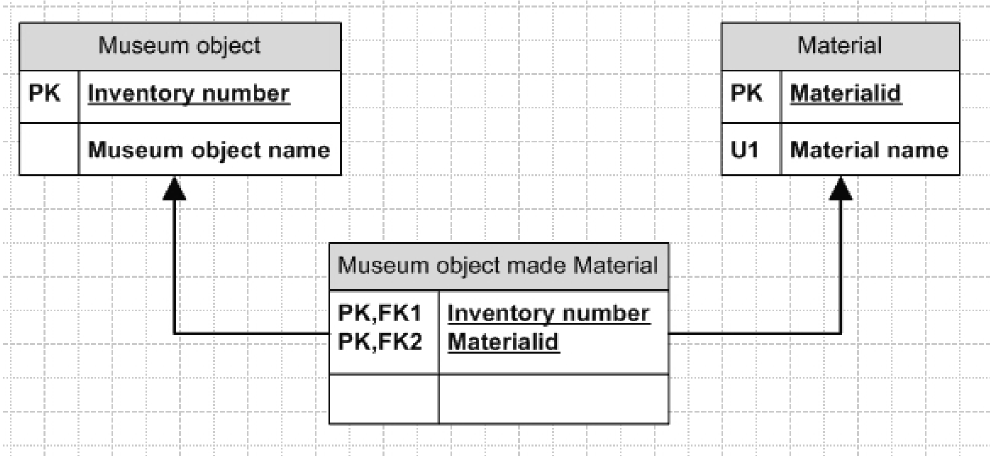
A **relationship** represents link between two tables. Relationships in the relational logical model are enforced by foreign key constraint in the physical database.

Normalization is the method used to store data in separate but linked tables, and in this way redundancy and inconsistencies of data are avoided. *Visio* through CSDP steps enables generating database relational logical model that is in Fifth normal form. Parts of code are located inside of database making it “intelligent”. It means that business logic is implemented into a database making client very “thin”.

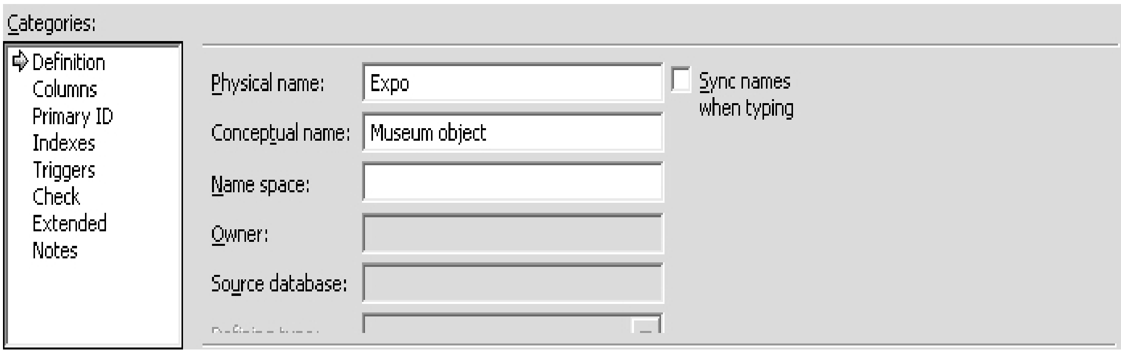
In the technical sense, the creation of logical model in *Visio* is simplified. It is necessary to import finished ORM source model to a newly open project for model creating. A list of tables in the Tables and Views evidence window are created by activating proper menu option. Logical model, as presented in the figure below, was created by dragging tables into a drawing area of screen. Tables are symbolized by rectangles having conceptual table name, primary key (PK) and foreign key (FK). The symbol “U1” shows uniqueness on column “the name of material” in the table “Material” what particularly means that it will not be possible for users to input more than once the same names of material. This constraint was defined earlier in creating conceptual model (in “constraint tab” in fact editor) when defining the fact type “The material has its own name of material”. Of course, there is not the same constraint on table “Museum object”, because repeating of the objects names is very frequent.

As mentioned before, the relationship between “Museum object” and “Material” is more – to – more. We can notice that *Visio* itself adds the third table “Museum object is made of material”, containing only primary keys of both tables. In the third table all materials for each museum object are evidenced separately.

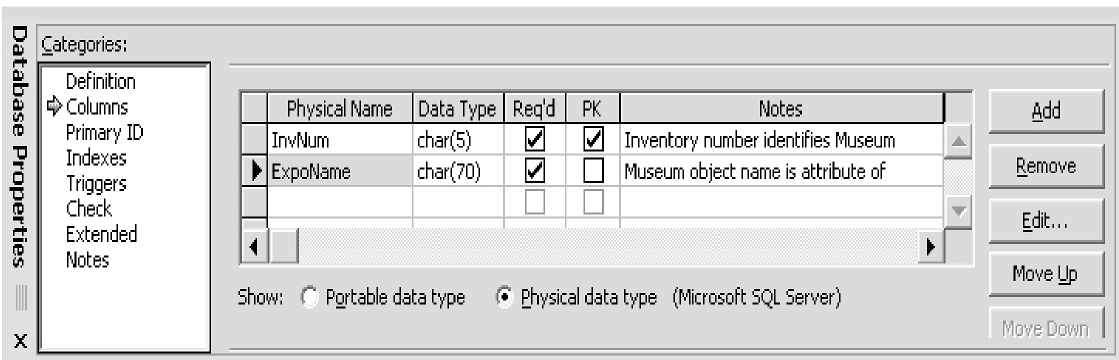
Relationships between tables are shown by arrows.



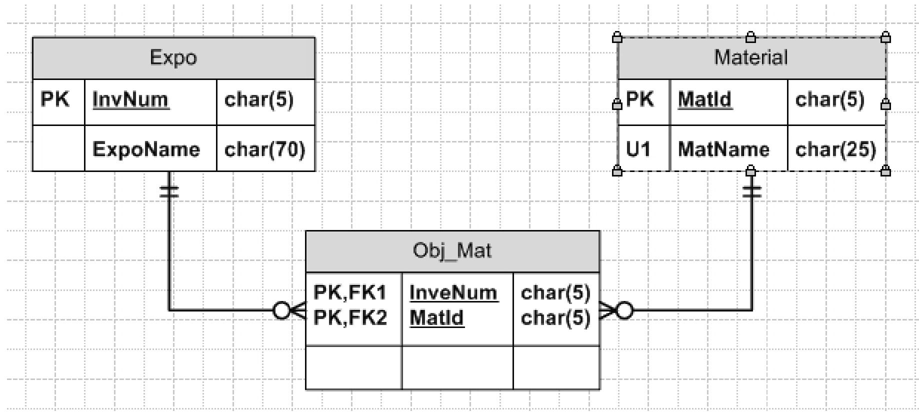
The next step is to determine data type which will be stored in corresponding columns. Data type definition is depending of DBMS (Database Management System) choice. The physical names of columns, as they will be named in final database, are defined, too.



Physical names of columns in table "Museum object", as well as data types, are shown on the following figure. The fact instance definition text from conceptual phase is stored now in column "notes" inside of window "Database property".



Now we can see the model graphical scheme of a new database after specifying physical names.



Generating database is made by simply choosing option "Generate" from "database" menu. This option starts the wizard (Generate Wizard) that puts us several questions.

After all these steps Visio enables physical generation of database together with code segments (implementation of defined constraints in CSDP) bearing business requirements directly to database.

Finally, the creation of reports on both levels (conceptual and physical) is possible.

Visio enables *Reverse engineering*. It means that all changes in the structure of the database in use are reflected reverse to relational logical level. It is not recommended to use Reverse engineering up to ORM source model because some business requirements might be lost.

However, we can see the possible way of implementation on user application level (the part of user interface especially designed for this text using Microsoft FoxPro 9 and its database).

Literature

http:www.**cidoc**.icom.org

Group of authors, 2002. *Microsoft Official Course 2090A. Modeling Business Requirements to Create Database Using Microsoft Visual Studio .NET Enterprise Architect*. Microsoft® Corporation

Hammer, M. and Champy, C. 1993. *Reengineering the Corporation: A Manifesto for Business Revolution*. New York: Harper Business

Neda Jevremović, mr Zoran Cvetković, *Museum Information System of Serbia*, Muzeološke sveske 8, Belgrade – National museum, 1996.

Goran Gavrilović, *Operativni sistem i sistem za upravljanje podacima*, Muzeološke sveske 8, str. 99, Belgrade – National museum, 1996.

http://www.iso.org/iso/en/ISOOnline.frontpage