Transboundary water management adaptation in the Amudarya basin to climate change uncertainties

Planning zone model

Report

2.8.1 Planning zone model

2.8.2 Database and interface

Project coordinator _____ V.A.Dukhovniy

Responsible for item 2.8 _____ A.G.Sorokn

Executor _____ R.R.Khafazov

1. Objectives and tasks

Objective – adapting the Aral Sea Basin Management model.

Tasks:

1. analyze modules and architecture of the planning zone subsystem – PZM;
2. apply IDEF methodology for modeling of the PZM subsystem;
3. apply IDEF0 methodology for design of the function model of planning zone;
4. apply IDEF1X methodology for design of the information model of planning zone;
5. integrate the function and information models into the PZM subsystem;
6. develop server and client parts of the model using the modern web-technologies.

2. IDEF family methodology[1]

The project on development of the information system (IS) consists of the following steps: the analysis (before developing IS, it is necessary to understand and describe business logic of subject domain), design (determine modules and architecture of the future system), coding, testing and maintenance. It is well known that it is more costly to correct errors made at previous stages than those made at the current stage. Thus initial stages of the project development are very critical. Hence, the availability of effective means of automation for initial stages of the project development is extremely important.

2.1 Methodology of IDEF0 functional modeling

First, when developing IS, one should understand how the system, which is to be automated, operates. Thus, a model describing such operation is to be developed. This model must be in conformity with the subject domain; it should contain knowledge of all actors of business processes in the system. IDEF0 is the most suitable language for modeling business processes; it was proposed 20 years ago and called SADT (Structured analysis and design technique). In early 1970s, the US Air Force applied the SADT subset designed for process modeling in a variety of projects as part of the ICAM (Integrated computer-aided manufacturing). Later this SADT subset has been standardized and called IDEF0.

IDEF0 is derived from the graphic modeling language of business processes. In IDEF0 notation, the model is represented by hierarchically systematized and interrelated diagrams.

Each diagram is a unit of system description and placed separately. The model contains three types of diagram:

1. context diagram;
2. decomposition diagram;
3. node-trees.

[1] For more information on IDEF standards, please, visit http://www.idef.com

2.1.1 Context diagram

The context diagram is the starting node tree diagram, which describes the system you are modeling in its most complete form and it’s interactions with external environment.

A context diagram of the planning zone model is shown in Fig.2.1.

2.1.2 Decomposition of context diagram

After general description of the system in the context diagram, it is decomposed into large fragments. This process is called a functional decomposition; the diagrams describing each fragment and their interrelations are called decomposition diagrams.

A decomposition diagram of the planning zone model is shown in Fig.2.2.

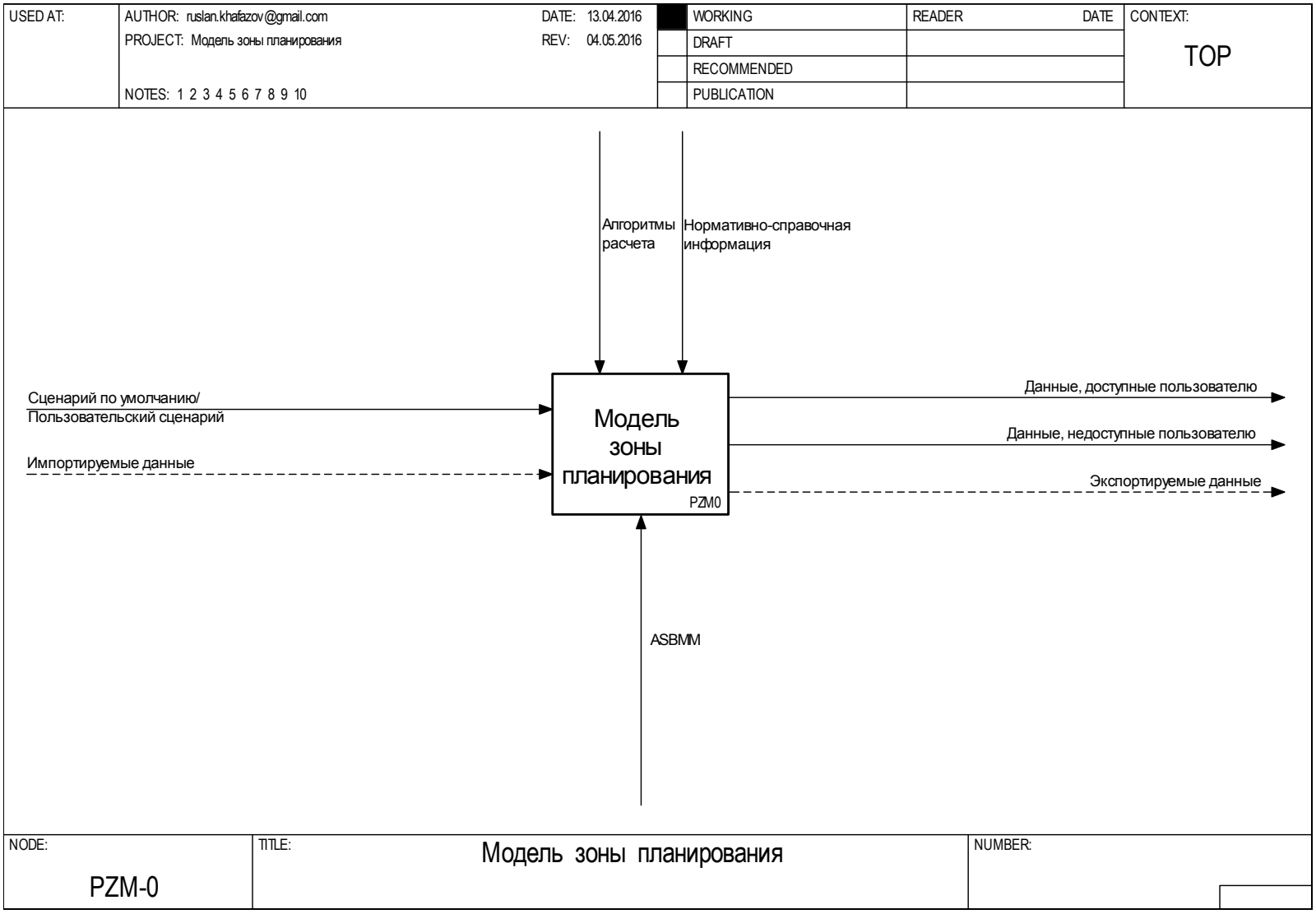

Fig. 2.1 – Context diagram

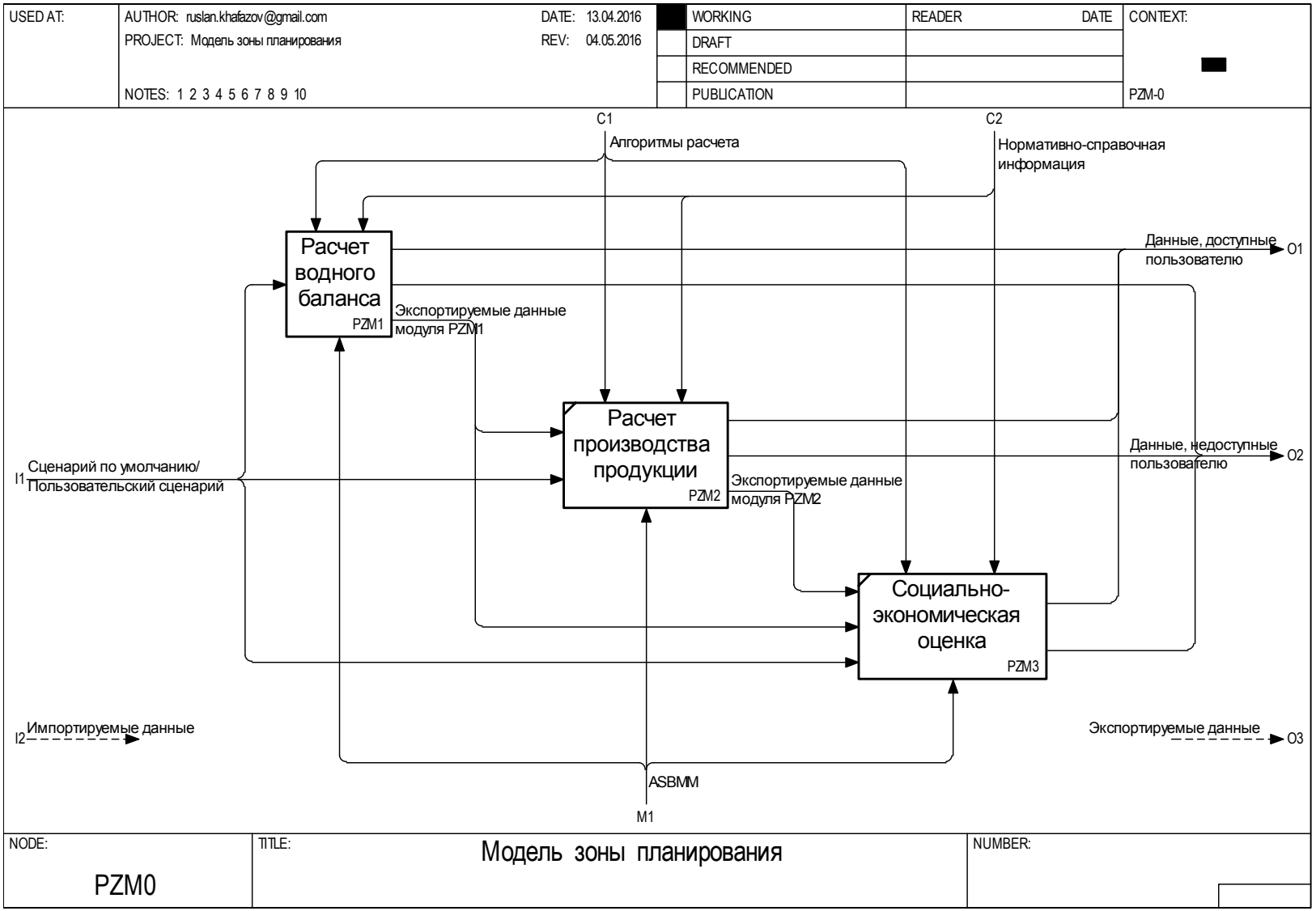

Fig. 2.2 – Decomposition diagram

2.1.3 Node tree diagram

The node tree diagram shows the hierarchical relationship between activities. These diagrams may be as much as possible as the tree may be rooted at a chosen node.

A node tree diagram of the planning zone model is shown in Figure 2.3.

2.1.4 Decomposition of the “Water balance calculation” module

After the decomposition of the context diagram, large fragments of the system are decomposed into smaller fragments, until the required level of detailing is reached.

The decomposition of the “Water balance calculation” module is presented in Fig.2.4.

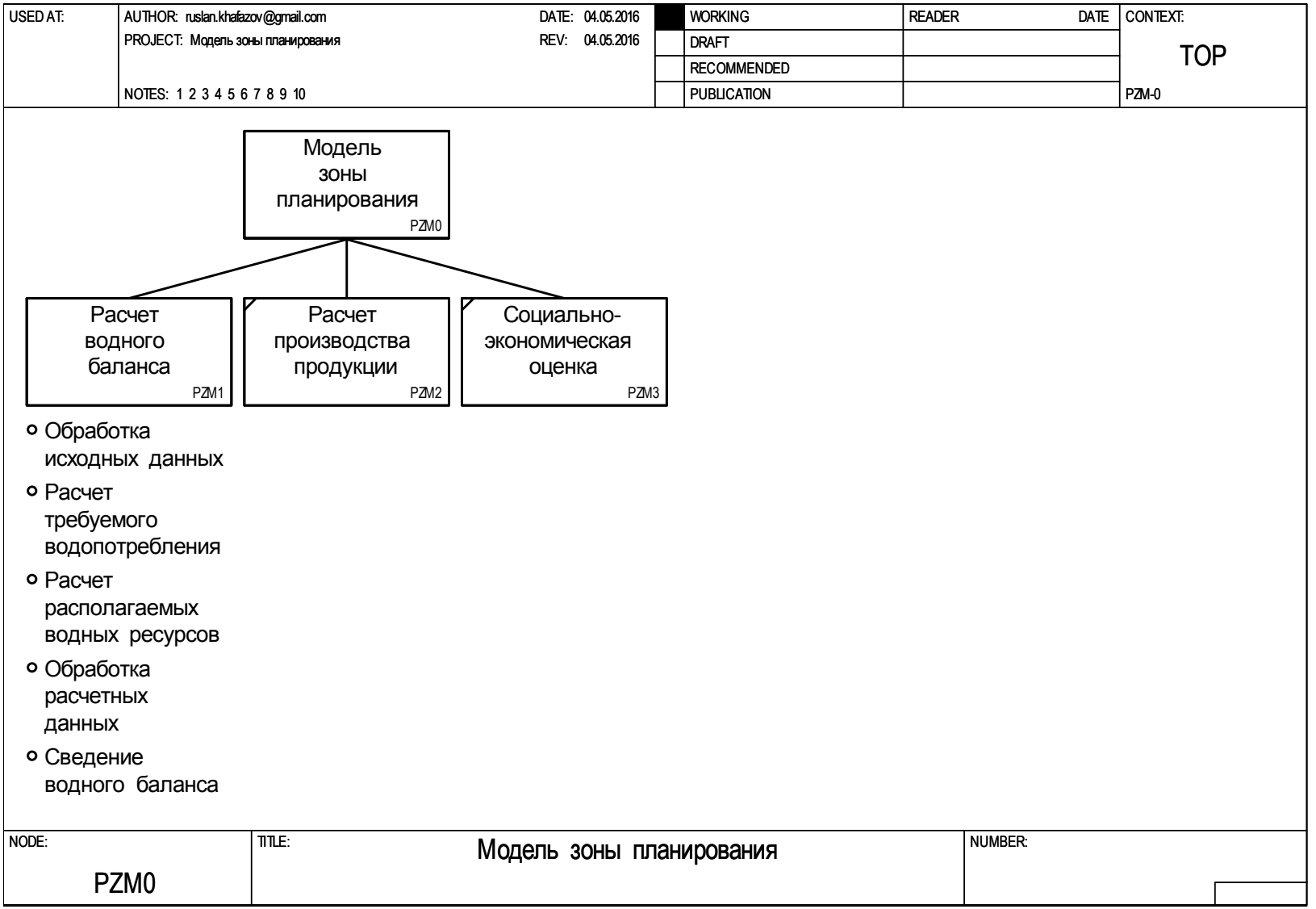

Fig. 2.3 – Node trees

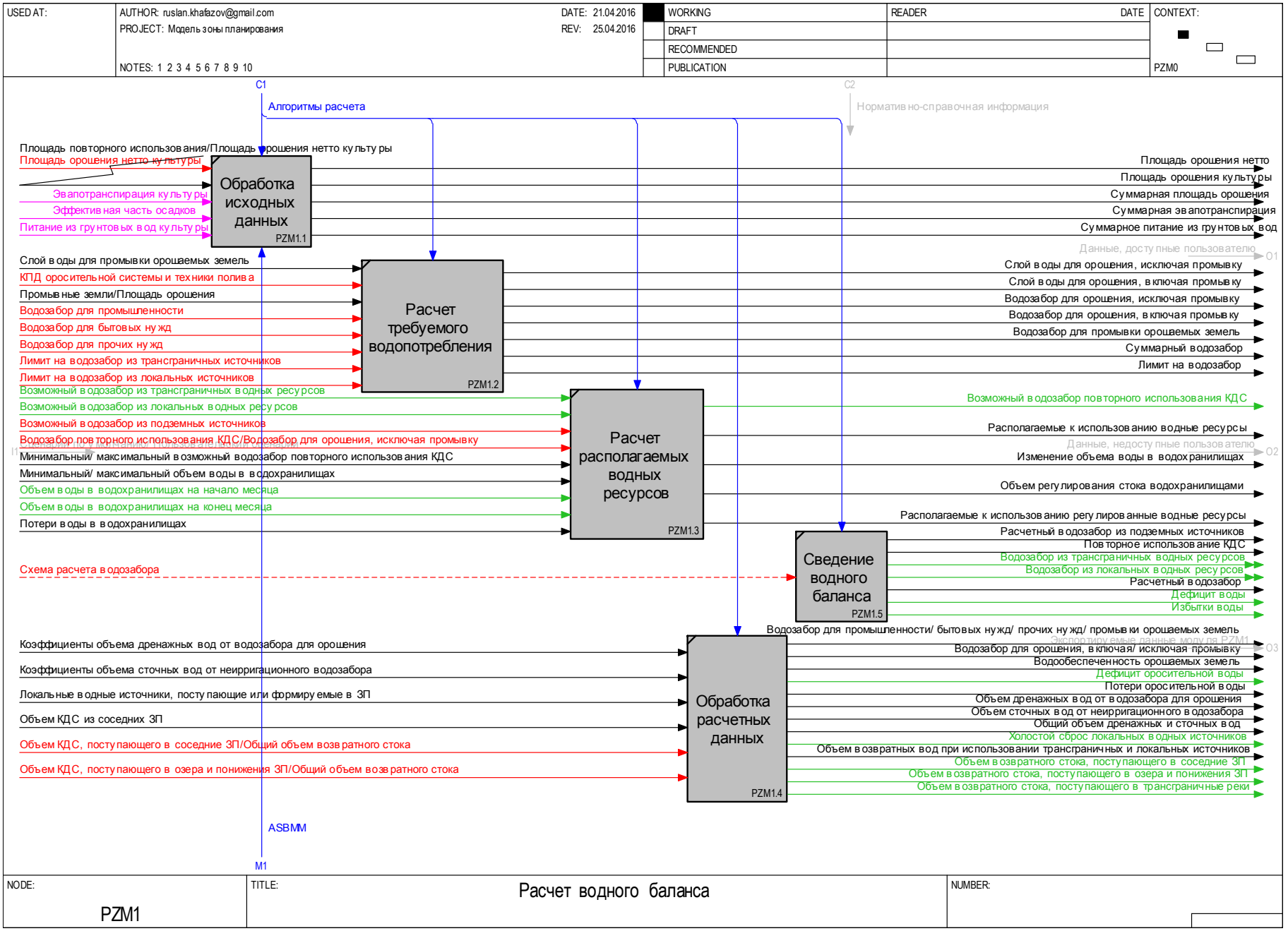

Fig. 2.4 – Decomposition of the “Water balance calculation” module

2.2 IDEF1X Information Modeling Methodology

The information modeling methodology is used to produce and information model which represents the structure of information within the system or environment. IDEF1 method developed by T.Ramey is also based on P.Chen approach and permits the construction of relational database. The IDEF1X methodology has been developed on the basis of upgrading of the IDEF1 methodology.

2.2.1 Logical model of database

IDEF1X methodology has two levels of model representation – logical and physical. Logical level presents an abstract view of data, where data is represented as it is in the real world and may be called as in the real world.

Three sublevels of the logical level are defined in the database model and differ from each other by the depth of provided information on data:

1. entity relationship diagram (ERD);
2. key based model (KB);
3. fully attributed model (FA).

2.2.1.1 Conceptual model of database

The entity relationship diagram contains entities and relationships reflecting basic business rules of subject domain. Such diagram does not give detailed data – it only contains basic entities and relationships between them.

The key-based model gives more detailed presentation of the data. It includes the description of all entities and primary keys and serves for representation of the structure of data and keys that correspond to the subject domain.

The conceptual model of database is presented in Figure 2.5.

2.2.1.2 Fully attributed model of database

The fully attributed model represents the most detailed data structure (the data in third normal form) and includes all entities, attributes and relationships.

The fully attributed model of database for the planning zone model is shown in Figure 2.6.

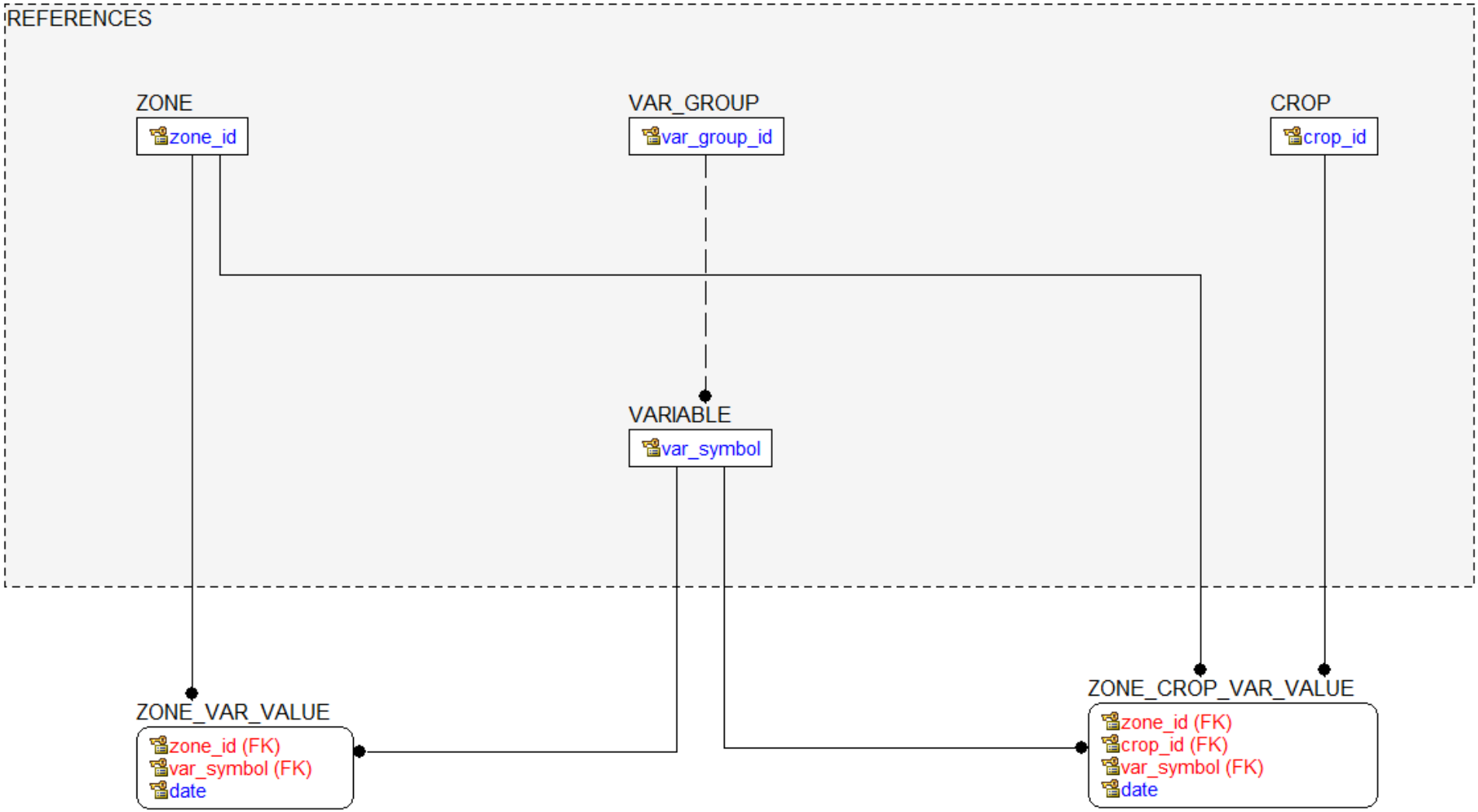

Fig. 2.5 – Conceptual model of database

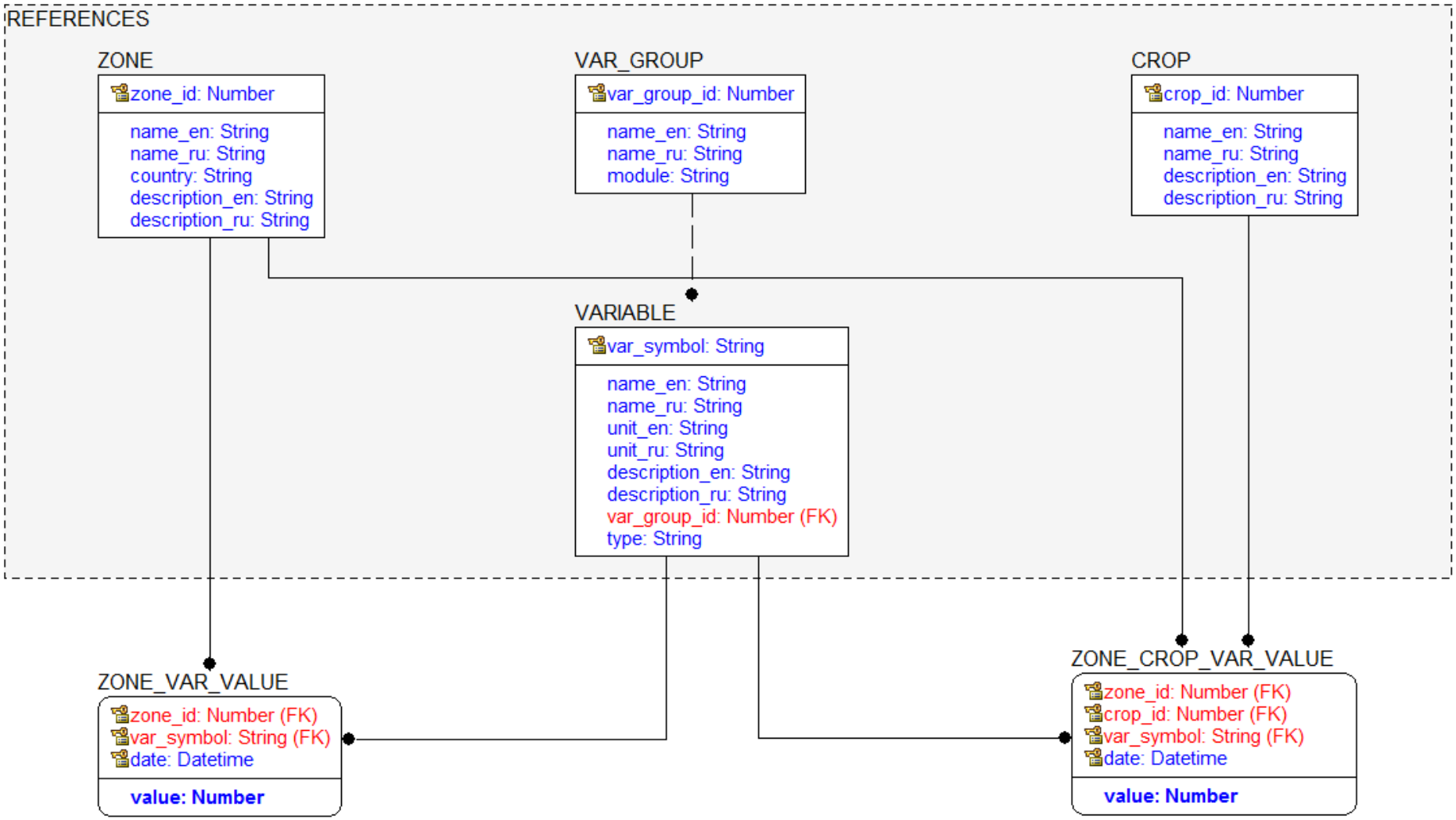

Fig. 2.6 – Fully attributed model of database

2.2.2 Physical model of database

The logical level of the database model is universal and not connected to a particular DBMS. The physical level of the database model, on the contrary, depends on a particular DBMS, actually reflecting the system catalogue. The physical level of the model contains information about all database objects. As there are no standards for database objects (e.g. no standard for data types), the physical level depends on the particular implementation of DBMS. Hence, the same logical level of the model may correspond to several different physical levels of the model. If the data type of attribute is not important at the logical level of the model, then at the physical level of the model, it is important to describe all information about particular physical objects – tables, columns, indexes, procedures, etc.

The physical model of database for the planning zone model is shown in Figures 2.7 and 2.8.

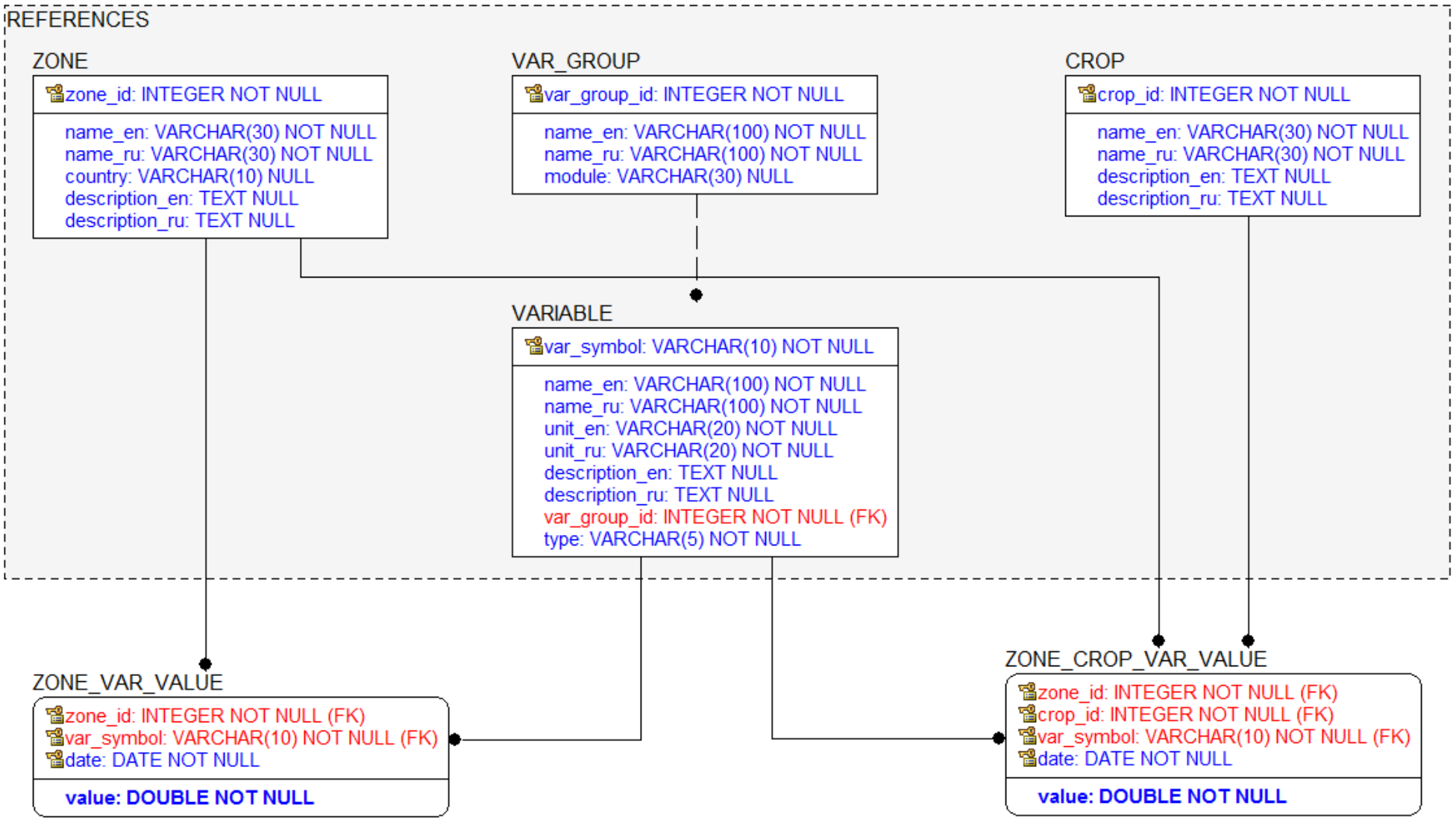

Fig. 2.7 - Physical model of database

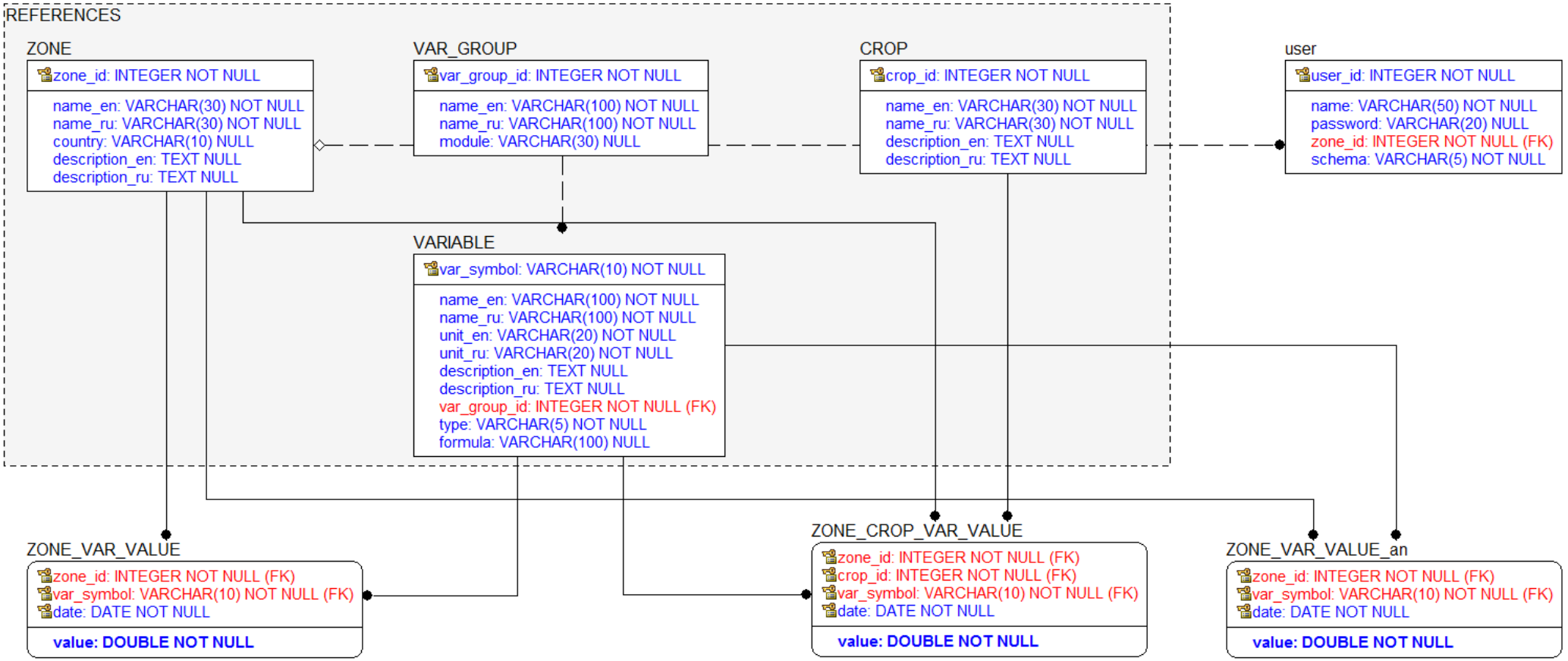

Fig. 2.8 – Physical model of database

3. Means and methodology for development of server and client sides of the model[2]

3.1 Means and methods for development of the server side of the model

The model's server side is implemented in form of DBMS stored procedures MariaDB 10.1 (MySQL).

The stored procedure is an object of database that represents a series of SQL statements, which are compiled once and stored on the server. The stored procedures are very similar to ordinary procedures of high-level languages; they may have input and output parameters and local variables; numerical calculations and operations on symbolic data may be done there; the results of these operations may be treated as variables and parameters. Standard manipulations with databases (such as DDL and DML) are also allowable in the stored procedures. Moreover, cycles and branching are also possible in the stored procedures, i.e. the process control statements may be used there.

Tables and stored procedures of DBMS for the planning zone model are presented in Figures 3.1 and 3.2.

[2] 2.8.2 Database, interface

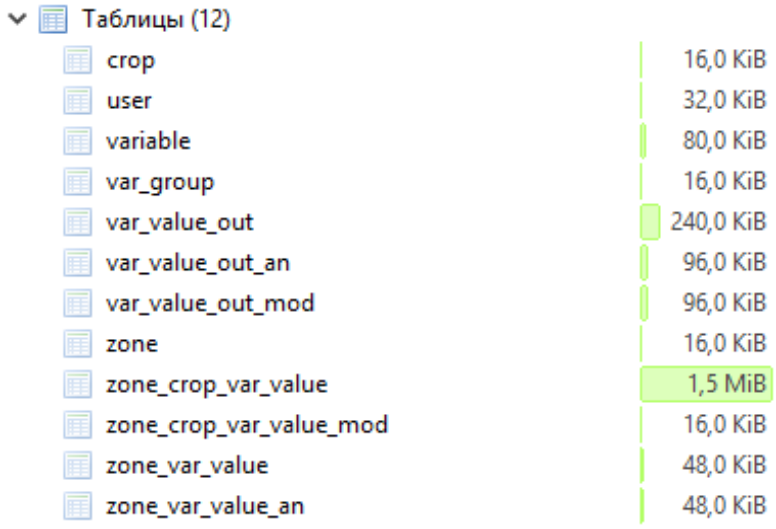

Fig. 3.1 – Tables of planning zone model DBMS

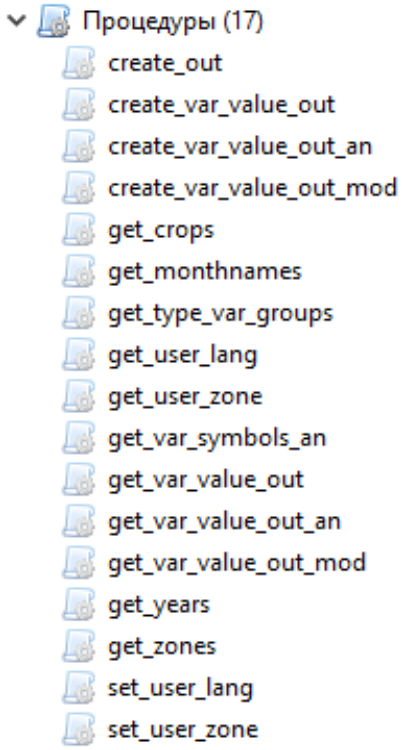

Fig. 3.2 – Stored procedures of planning zone model DBMS

3.1.1 Description of major stored procedures

The major stored procedures of the planning zone model DBMS are described in Table3.1.

Table 3.1 - Description of major stored procedures of the planning zone model DBMS

Stored procedure	**create_var_value_***	**get_var_value_***
out	Calculation of water balance	Selection of water balance calculation results
out_mod	Calculation of irrigated agriculture output and socio-economic assessment	Selection of calculation results
out_an	Analysis of simulated and actual data	Selection of analysis results
get_*, set_*	Additional procedures on the model's client side	

3.2 Means and methods for development of the model's client side

The client side of the model is implemented through the Yii 2.0 web-framework.

Yii is an object-oriented, component-based MVC PHP web application framework.

The MVC (Model-View-Controller) design pattern is widely used in web programing and separates the business logic from the user interface so that developers could easily change particular parts of the web-application without changing the other parts. In the MVC architecture, the model represents data and rules of business logic; the view is responsible for the user interface (for instance, the text, entry field); the controller ensures interaction between the model and view (Fig.3.3).

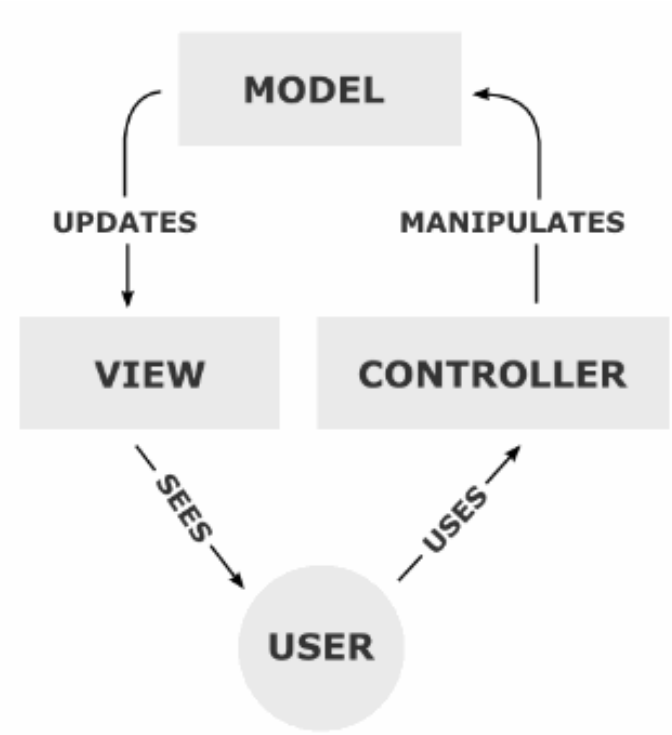

Fig. 3.2 – MVC design pattern

It is important that both the view and controller depend on the model; however, the model (active) does not depend on both. Hence, this division allows developing the model independently from visual view and creating several different views for a model.

3.2.1 Description of MVC major components

The MVC major components are described in Figure 3.2.

Table 3.2 - Description of MVC major components

<table>
<tr><th>Model</th><th colspan="2">Controller</th><th colspan="2">View</th></tr>
<tr><td rowspan="4">Model is implemented in form of DBMS stored procedures</td><td colspan="2">Controller SiteController</td><td rowspan="2">Calculation</td><td rowspan="2">Representation of water balance calculation results</td></tr>
<tr><td>actionCalculation</td><td>Method (controller) for model calculations</td></tr>
<tr><td rowspan="2">actionAnalysis</td><td rowspan="2">Method (controller) for analysis of simulated and actual data</td><td>Calculationmod</td><td>Representation of calculation results</td></tr>
<tr><td>Analysis</td><td>Representation of analysis results</td></tr>
</table>