Interface system of a software package for electricity metering

T V Zhidchenko, M N Seredina, N M Udintsova and V V Seregina

Azov-Black Sea Engineering Institute of the Federal State Budgetary Educational Institution of Higher Education Don State Agrarian University, 21 Lenina, Zernograd, 347740, Russia

E-mail: tvzh@inbox.ru

Abstract. In almost all spheres of human activity, information has become an important resource, and information systems are a necessary tool. The article discusses information flows for building an interface system, computing and communication equipment of an information system for metering electrical energy. The article presents an analysis of information from the organizational structure of departments. The equipment of the system is displayed using functional levels, the information flows circulating at the levels are presented. The concepts of a database and a database management system, as a set of language and software tools, are used for the development and design of an interface system. The article presents the development of an interface system in a declarative, non-procedural SQL programming language, taking into account the peculiarities of the operation of electricity metering for various consumers. The advantages of this programming language are: relational framework, portability between platform architectures, high-level structure, the ability to develop interactive queries. The structure of the information system is considered at four functional levels. An infological database model has been developed for accounting for electrical energy, in which logical connections between objects are established. One of the priority areas of digital development of agriculture is energy engineering, monitoring and IT-technologies for data processing, commissioning of information systems and storage of information in databases. This leads to the need to use the accumulated data with technologically different tools. The solution to this problem is reduced to the use of specialized software tools. The work also shows that it is possible to further build up this system to organize a centralized database, the availability of which can satisfy the work of power sales departments.

1. Introduction
The development of new algorithms and the modeling of interfaces are related to the task of building a holistic integration platform.
The power sales department includes the following departments: the electricity sales department, the electricity sales accounting department, the lead inspector, the contractual and legal department and other departments (Figure 1).

For further correct construction of actions for the development of the structure of the accounting information system, the principle of functional levels is used (Figure 2). The structure shows that the functional information system for electricity metering contains four levels [2].

2. Purpose, objects and research methods
The first level is primary measuring devices with telemetric or digital outputs. All of them are sealed by the energy sales company and regulated by a certain calibration interval, which controls the energy sales department. Therefore, in the power sales department there is a center of control and measuring devices and their records are kept. Also, recently, developers and suppliers of control and metering devices for electricity are installing self-testing systems in metering devices, which greatly facilitates the task of the control service for measuring instruments.

The second level is multiplexers and concentrators, which are data collection and preparation devices. Such devices can be paired with other equipment that allows data transfer.

The third level is a personal computer of the information accumulation and processing department.

The fourth level is the department for aggregating and structuring the collected data and processing all information.

In the power sales department, the first, second and third levels are implemented according to the proposed version of ABB VEI Metronica. An automated workstation "Consumer" from ABB VEI Metronica is used as software for the third level. With the help of an automated workstation "Consumer" installed on a laptop, metering devices are interrogated.

There are several automated workplaces in the information structure of the power sales department. Information flows circulating in the system are shown in Figure 3.
It is one component of the information system - the database management system - a set of language and software, with the help of which the database is created and maintained during operation.

To work with information stored in a database, the database management system provides programs and users with the following two types of languages:

- Data description language - a high-level non-procedural language designed to describe a logical data structure;
• The language of manipulation [5] data - a set of structures that provide the implementation of the basic operations for working with data: input, modification and retrieval of data on request.

The creation of a standard database language that could function in various computer systems was driven by the need to store and process them in modern database management systems.

Organizing data, retrieving information, modifying data, controlling access, sharing data, ensuring data integrity are functions that the programming language SQL now allows for implementation in database [1] management systems.

3. Results
The main purpose of the SQL language is to prepare and execute queries. As a result of fetching data [4] from one or more tables, a set of records can be obtained, called a view (in essence, this is a table).

For the fourth level, the developed script for creating tables [3] will look like this:

```sql
CREATE TABLE Subscribers (  
    [Apartment number]      SmallInt,  
    Bank                    Text(50) WITH COMP,  
    [Checking account]      Text(50) WITH COMP,  
    [Consumer attribute]    Text(15) WITH COMP,  
    [Contract number]       Text(25) WITH COMP,  
    [Corresponding account] Text(50) WITH COMP,  
    [Date of conclusion of the contract] DateTime,  
    [Date of liquidation of the contract] DateTime,  
    [Device code]            SmallInt,  
    Fax                     Text(50) WITH COMP,  
    House                   SmallInt,  
    [Individual tax number] Text(10) WITH COMP,  
    [Installed power, kW]   Single,  
    Locality                SmallInt,  
    [Middle name]           Text(20) WITH COMP,  
    Name                    Text(20) WITH COMP,  
    [Name of the property]  Text(50) WITH COMP,  
    [Number of residents]   SmallInt,  
    [Number of rooms]       SmallInt,  
    Phone                   Text(10) WITH COMP,  
    Street                  SmallInt,  
    [Subscriber code]       SmallInt,  
    Surname                 Text(20) WITH COMP,  
    CONSTRAINT PrimaryKey PRIMARY KEY ([Subscriber code])
);

CREATE TABLE [Counter readings] (  
    [Accounting point code] SmallInt,  
    [Benefit code]           SmallInt,  
    [Consumer attribute]    Text(20),  
    [Current readings]      SmallInt,  
    [Date of taking readings] DateTime,  
    [Previous readings]     SmallInt,  
    [Subscriber code]       SmallInt,  
    [Tariff code]           SmallInt,
```

CONSTRAINT PrimaryKey PRIMARY KEY ([Subscriber code]),
CONSTRAINT [SubscribersCounter readings] UNIQUE ([Subscriber code])
);

CREATE TABLE [Metering devices] (  
[Date of instrument verification] DateTime,
[Device code]           SmallInt,
[Device type]           Text(50) WITH COMP,
[Electricity type]      Text(50) WITH COMP,
[Installation date of metering device] DateTime,
[Instrument scale value] Single,
[Instrument verification period] Long,
[Phase of the device]   SmallInt,
CONSTRAINT PrimaryKey PRIMARY KEY ([Device code])
);

Since the structure of the information system is thought out at the design stage and relationships between tables are outlined, the creation of relational relationships between tables is performed using key or indexed fields. The main advantages of database management systems are realized when working not with individual tables, but with groups of interconnected tables. Links are characterized by two main purposes. One is to ensure data integrity, and the other is to automate database maintenance tasks.

The relational relationship between tables is intended to:

• in the key field of the main table, if any fields of other tables are associated with this field, exclude the possibility of deleting or changing data;

• in the key field of the main table when deleting (or changing) data, the corresponding data in the fields of related tables will be deleted or changed automatically.

Figure 4. Data schema

The essence of the connectivity of tables is to establish a correspondence between the single-valued fields of the main and additional tables. The connectivity fields of the main table can be regular and
key. Key fields are most often used as the link fields of the subordinate table. The following four main
types of relationship are established between two tables:
  • unambiguous;
  • unambiguous to multi-valued;
  • multi-valued.
The developed diagram of the relationship between the tables is shown in Figure 4.

ALTER TABLE [Counter readings]
  ADD CONSTRAINT [RateCounter readings] FOREIGN KEY
  ([Tariff code]) REFERENCES Rate ON UPDATE CASCADE ON DELETE
  CASCADE;
ALTER TABLE [Counter readings]
  ADD CONSTRAINT [Subscriber benefitsCounter readings] FOREIGN KEY
  ([Benefit code]) REFERENCES [Subscriber benefits]
  ON UPDATE CASCADE ON DELETE CASCADE;
ALTER TABLE [Counter readings]
  ADD CONSTRAINT [SubscribersCounter readings] FOREIGN KEY
  ([Subscriber code]) REFERENCES Subscribers ON UPDATE CASCADE
  ON DELETE CASCADE;

Forms are objects with the help of which new data is entered into the database or viewed. For the
convenience of the user-operator, forms have been developed for the existing base tables. Unlike
queries, which are specialized tools for fetching and analyzing data, forms are tools for entering data.
Their main purpose is to leave open only those fields that he is supposed to fill out. To automate input
when working with big data, you can place special controls in the form: counters, drop-down lists,
radio buttons, check boxes, and other elements. The convenience and benefits of forms are revealed
when data is entered from completed forms. To make it repeat the design of the form by graphic
means, the form is designed in the same way - this significantly simplifies the operator's work, reduces
his fatigue and prevents the appearance of typing errors.

Let’s create a form, the script of which is given below:

CREATE PROCEDURE [~sq_fSubscribers and counter readings] AS
  SELECT [Subscribers].[Subscriber code] AS [Subscribers_Subscriber code],
          [Subscribers].[Device code],
          [Subscribers].[Contract number], [Subscribers].[Surname],
          [Subscribers].[Name], [Subscribers].[Middle name],
          [Subscribers].[Locality], [Subscribers].[Street],
          [Subscribers].[House], [Subscribers].[Apartment number],
          [Subscribers].[Number of rooms], [Subscribers].[Number of residents],
          [Subscribers].[Date of conclusion of the contract], [Subscribers].[Phone],
          [Subscribers].[Individual tax number], [Subscribers].[Name of the property],
          [Subscribers].[Installed power, kW], [Subscribers].[Checking account],
          [Subscribers].[Bank], [Subscribers].[Corresponding account],
          [Subscribers].[Date of liquidation of the contract],
          [Subscribers].[Fax], [Subscribers].[Consumer attribute] AS [Subscribers_Consumer
          attribute],
          [Counter readings].[Subscriber code] AS [Counter readings_Subscriber code],
          [Counter readings].[Consumer attribute] AS [Counter readings].[Consumer attribute] AS [Counter
readings_Consumer attribute], [Counter readings].[Accounting point code], [Counter readings].[Date of taking readings], [Counter readings].[Previous readings], [Counter readings].[Current readings], [Counter readings].[Benefit code], [Counter readings].[Tariff code]
FROM Subscribers INNER JOIN [Counter readings] ON [Subscribers].[Subscriber code] = [Counter readings].[Subscriber code];

You can use queries to view, analyze, and modify data from multiple tables. They are also used as a data source for forms and reports.

The most commonly used query is fetch. When it is executed, data that meet the selection conditions are selected from one or several tables and displayed in a specific order.

Let's create a request to determine the payment for electricity for each subscriber.

CREATE VIEW [Electricity payment] AS
SELECT [Power consumption].[Subscriber code], [Power consumption].[Contract number], [Power consumption].Surname, [Power consumption].[Name of the property], [Power consumption].[Tariff amount 1], [Power consumption].[Tariff amount 2], [Power consumption].[Standard kW * h], [Power consumption].[Percentage of benefit], IF([Power consumption] <= [Standard kW * h], [Power consumption]*[Tariff amount 1] * (1 - [Percentage of benefit]/100), ([Standard kW * h] *[Tariff amount 1] + ([Power consumption]-[Standard kW * h])*[Tariff amount 2])* (1 - [Percentage of benefit]/100)) AS [Electricity payment]
FROM [Power consumption];

4. Conclusions
As a result of the work, an information system was developed that allows for the analysis and accounting of electrical energy consumed by subscribers, and also allows organizing the accounting of electrical energy consumption and calculating payment for it. The work also shows that it is possible to further build up this system to organize a centralized database, the availability of which can satisfy the work of power sales departments.

References
[1] Grinchenko S N 2020 On the system hierarchy of artificial intelligence Informatics and its applications 15 №1 pp 111-115
[2] Zatsman I M 2019 Third-order interfaces in computer science Informatics and its applications 13 pp 82-89
[3] Rzheutskaya S U 2010 SQL language 160
[4] Shirikova O A 2015 Computer mathematics systems and their applications Organization of queries in SQL taking into account the peculiarities of the applied database 16 277-278
[5] Fehily Chris. 2017 SQL 452