



**Buzovskyi V.,
Kologrivov M.,
Antonova A.**

OIL STORAGES AND MAIN OIL PIPELINES PUMPS DATABASE DEVELOPMENT

The object of research is a data model that characterizes the pumping units of main oil pipelines. The paper considers the creation of pumps database and their characteristics as a component of software systems designed to search for sources of energy saving in pipeline transportation of oil and oil products.

An overview of existing programs for calculating the joint operation of an oil pumping station and a pipeline is given. It is shown that most programs are inaccessible for analyzing the applied data storage technologies, or involve manual input of entry data without using databases. Comparative analysis of spreadsheets and relational databases is performed. It is shown that relational databases better meet the requirements of data convenience, availability, scalability, and performance. A physical model of a relational database is presented. The parent and child entities have been established, which make it possible to fully reflect all information about the pumping unit, namely:

- coefficients of hydraulic characteristics depending on the diameter of the impeller rotor;
- characteristics depending only on the pump make;

– classification of the pump by design or purpose. Relationship types and referential integrity rules are defined between entities when deleting or updating data. For data management, the SQLite system is proposed, which provides data manipulation in the SQL query language and does not require the development of additional software. The analysis of possible ways of organizing multilingualism using a database is carried out. The text of the SQL query is proposed, which allows to select the pump with its characteristics by the pump make and impeller diameter. A database structure is proposed that allows storing information about equipment for oil pipelines for various purposes, which makes it possible for automated calculations of complex technological processes. The pumping unit database is an open source project that is posted on the public web service GitHub.

Keywords: oil pipeline, pumping units, hydraulic characteristic, database, entity-relationship model.

Received date: 05.05.2020

Accepted date: 18.06.2020

Published date: 31.10.2020

Copyright © 2020, Buzovskyi V., Kologrivov M., Antonova A.

This is an open access article under the CC BY license

(<http://creativecommons.org/licenses/by/4.0>)

1. Introduction

Oil and petroleum distribution systems are subject to constant changes, which are caused by the following reasons:

- new oil fields are discovered and the volumes of oil production on existing oil fields are changed;
- refineries can increase or decrease the volume of oil refining, which leads to a change in the volume of oil supplies;
- countries that import oil or transport it through their territory can diversify their suppliers of oil and oil products, which can also affect the operation of their oil transportation systems.

For example, in Ukraine, Ukrtransnafta has adopted a number of strategic directions. One of the priorities is to promote Ukraine's integration into the Eurasian oil transport corridor project, which involves the construction of the Brody-Plock oil pipeline. At the moment, the pipeline system of Ukraine is operating in an underloaded mode. This is caused by a significant decrease in the production of petroleum products at local oil refineries and a decrease in the volume of oil transit through its territory. Nevertheless, the scope of Ukraine's oil transportation system is large. 4767.4 km of main pipelines, the «Yuzhny» offshore oil terminal and 28 oil pumping stations (OPS) are being involved [1].

Oil transportation systems consume significant amounts of energy. Energy resources are limited and expensive, so countries legislate energy conservation strategies.

For example, Ukraine has adopted a law «On Energy Saving» and a number of other regulatory documents [2], which prescribe organizational and technical measures to optimize the operating modes of oil pipelines in underloaded sections.

To ensure the required performance on the operating pipelines, taking into account the criteria for energy efficiency, alternative (optimization) calculations are used. Such calculations are usually performed with software products.

There are a lot of references of computer programs using for solving many problems of oil transportation and storage, though [3–5], there is no evidence about the information technologies used in the development of these programs. There are also no recommendations for building equipment databases, used in oil transportation and storage. A database provides the ability to organize a single access to the characteristics of equipment and their control, and also provides the ability to apply generally accepted design patterns for software products. Production enterprises, design organizations, scientific and educational institutions can show significant interest in the accumulation and structuring of information about oil technological equipment.

In view of insufficient coverage of issues related to the storage of information about technological equipment of oil pipelines, the description of the principles of building databases seems to be an urgent task.

2. The object of research and its technological audit

The object of research is a data model that characterizes the pumping units of main oil pipelines. Currently, mainline pumps in Ukraine are operating in an underloaded condition. The pumps are driven by electric motors, the power consumption of which depends on the pump performance. Pump operating modes are far from optimal [6], which can lead to overestimated specific power consumption of the drive. In general, the reasons for the low load of oil pipelines can be as follows:

- insufficient amount of pumped liquid in the tanks of the head oil pumping station or overfilling of the tanks of the final destination;
- repair of the pipeline without stopping the pumping;
- change in the viscosity of the pumped oil;
- technological limitations of the pressure value at the inlet and outlet of pumping stations (PS);
- presence of travel discharges.

Since most of the operating costs of the oil pipeline are due to the operation of electric motors [7], then the positive effect of realizing the energy saving potential of pumping units is also the greatest [8]. When choosing rational modes of oil pipelines operation, a number of measures are considered:

- regulation of the pipeline performance by changing the number of operating pumps at the pumping station;
- regulation by changing the pump operation scheme at the pumping station;
- bypassing;
- regulation by turning the pump impellers;
- frequency regulation of the pump.

To search for modes that meet the criteria of the highest energy efficiency and the lowest capital intensity, hundreds of options for the operation of the oil pipeline are calculated. To perform such calculations, as a rule, computer programs are written. Not only the amount of calculated data is significant, but also the data supplied to the input of the program. These include the characteristics of pumping equipment and the linear part of the main pipeline. This is explained by the fact that pumps of various brands and with various standard sizes of impellers operate on the existing oil pipelines. Pump data is usually stored in a tabular (Table 1) or graphically in books and catalogs [9–11].

In the hydraulic calculations of the oil pipeline, data are used to describe the characteristics of the pump:

- total head dependence H :

$$H = H_0 - b \cdot Q^2; \quad (1)$$

- dependency of the efficiency η_p from productivity Q (m^3/hr):

$$\eta_p = c_0 + c_1 \cdot Q + c_2 \cdot Q^2, \quad (2)$$

where H , η_p – total head (m) and pump efficiency at productivity Q (m^3/hr) respectively; H_0 , b , c_0 , c_1 , c_2 – empirical coefficients.

Manual entry of these data into the program is inconvenient, since it requires concentration of attention and significant labor costs.

3. The aim and objectives of research

The aim of research is to develop a database of pumping units for main oil pipelines.

To achieve the aim, it is necessary to solve the following objectives:

1. Design a physical database model, defining parent and child entities, types of relationships between them and data integrity rules, taking into account various variations of pump marks.
2. Develop the methods for organizing a multilingual database of pumping units.

4. Research of existing solutions of the problem

Most of the existing programs devoted to the calculation of the operating modes of oil pipelines are not available or work without databases.

For example, the NS1 program for calculating the productivity of an oil pipeline section adjacent to the main pumping station [13] prompts the user to manually enter data for pumps and pipeline through the console. In order to repeat the calculation with a changed value of one of the input parameters, it is necessary to re-enter the remaining parameters, which seems inconvenient. Entering data for pumping units from books and catalogs [9, 11, 12] requires significant labor costs. Reading speed is slow, errors are possible.

The RABT.BAS program has been developed in the BASIC language for calculating the joint operation of an oil pipeline and a pump station [14]. The program works without a database. The input of the coefficients of hydraulic characteristics of the pumps is done manually.

Table 1

Reference data for the pump NM 2500-230 (Ukraine) [12]

Pump mark	Impeller	D_2 , mm	Coefficients in the formula (1)		Coefficients in the formula (2)		
			H_0 , m	$10^6 \cdot b$, hr^2/m^5	$10^2 \cdot c_0$	$10^4 \cdot c_1$, hr/m^3	$10^6 \cdot c_2$, hr^2/m^5
NM 2500-230	0.5	425	246.7	16.8	24.8	644	-16.9
	0.7	405	248.7	7.61	-79.35	15.8	-37
	1	440	281.5	7.84	26.2	485	-9.7
		405	258.8	8.59	26.2	485	-9.7
		385	235.9	8.32	26.2	485	-9.7
	1.25	450	371.0	14.9	18.8	403	-6.2

In [15], the text of the program for calculating the operating mode of an oil pipeline with the use of anti-turbulent additives is given. Additive data is read from TXT files. The characteristic coefficients of the booster and main pumps are assigned directly to the variables. Database is not used.

A program for calculating non-stationary modes of main oil pipelines has been implemented in Pascal [16]. The program lacks the code responsible for connecting to the database. Pump and piping data are set manually by assigning variables.

«Transit» program for calculating the operating modes of the oil pipeline [17, 18] provides a convenient way to define main and booster pumps. The program has an interface. The selection of pumps is carried out using the drop-down list.

The «Transit» program is paid. Its source code is not available. Literary sources lack information about the implemented method of data storage, which excludes the possibility of using developers' experience of the «Transit» program for designing new databases.

During the research on the development and improvement of modeling and construction of effective algorithms for oil pipeline systems hydraulic calculation, an industrial complex of programs was created for the hydraulic calculation of main oil pipelines operating modes [19]. The software package was put into commercial operation, but it is not freely available and there is no information about the method of data organizing.

The method of storing information using MS Excel spreadsheets has become widespread. For example, in the program [20], pumping equipment data is organized into tables, in which each column corresponds to a certain characteristic of the pump. Thanks to the built-in filtration, the program allows to select a pump among many modifications. Among the disadvantages of spreadsheets, the following should be noted:

- MS Excel is a paid software product, therefore it is not available everywhere;
- many programming languages have libraries for working with Excel files, but the code for fetching data seems to be more cumbersome and less intuitive in comparison, for example, with SQL queries to relational databases;
- Excel file processing speed is slow when there are a significant number of rows [21–23].

To achieve the goal of the work, the most preferable is the use of relational databases. Their widespread use [24, 25] is due to the following reasons [26]:

- the relational data organization model is quite simple and at the same time powerful, suitable for solving the vast majority of data storage and processing tasks;
- this model is characterized by simplicity of data structure and user-friendly tabular presentation;
- ensures data integrity, scalability and independence from the application that uses the data.

The choice of a relational database management system (RDBMS) is not critical. For small projects, it seems convenient to use a lightweight RDBMS SQLite. Its advantage is a simplified data structure and no need to use a database server.

5. Methods of research

For the convenience of structuring the data, the pump characteristics will be divided into attributes and properties. Attributes will be considered characteristics that determine various combinations (variations) of the equipment mark. For example, for the pump NM 2500-230 such attributes as rotor number and impeller diameter can be distinguished D_2 . The rest of the characteristics are defined as properties. For example, the combination of the «impeller=1» and « $D_2=440$ » attributes uniquely determines the property $H_0=281.5$ m (Table 1).

When creating a relational database of pumping units, the following parent entities are defined: equipment, category, feature and attribute_group.

To store different variations of pump mark, feature values and attributes, the following child entities are defined: combination, feature_value, and attribute. Fig. 1 shows a physical database model that describes data types and relationships between entities.

Relationship types and referential integrity rules are defined between entities when deleting or updating data from parent entities. Let's take a closer look at each of the tables.

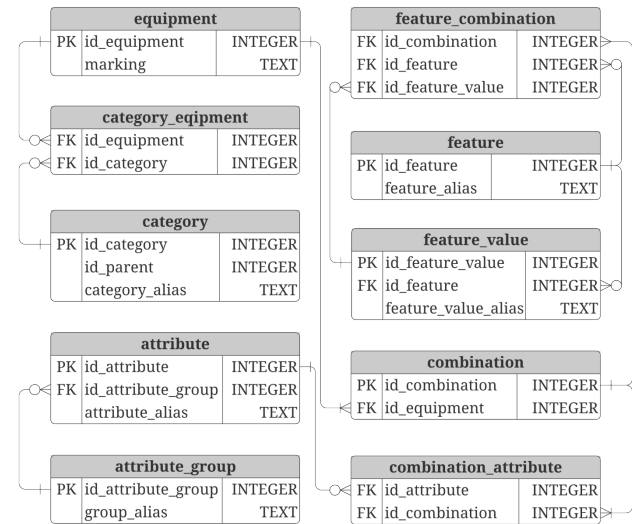


Fig. 1. Relational database physical model of main oil pipelines pumping units

The table «equipment» contains the mark of the pump. As will be shown below, it is convenient to store the pump mark itself in a separate table containing translations into various languages. The uniqueness of each record is ensured by the id_equipment field with a primary key (PK) and integer values. The «marking» field takes a text data type and is used to store the alias of the equipment brand.

When transporting oil and oil products, pumps of various types can be used. It is convenient to store information about possible types of pumps in the table «category» (Fig. 1). When developing its structure, one should take into account the criterion of scalability, which implies the possibility of creating an unlimited number of subcategories. For example, the category «pumps» can have subcategories «centrifugal pumps», «gear pumps», «piston pumps», which can be designated as subcategories of the second level. In turn, among the «centrifugal pumps» sectional pumps of the NM type and spiral pumps of the NM type can be distinguished, which are subcategories of the third level. The id_category field with the primary key contains a unique integer category identifier. An integer entry for the id_parent field points to the id_category of the parent category. The category_alias field is optional. It was created for a more understandable visual perception of the table and contains unique category aliases, which in this case correspond to the translations of categories into English.

Links between pumps and categories are defined in the «category_equipment» table (Fig. 1). Its id_equipment field with integer values indicates a unique equipment identifier in the equipment table. The id_category field points to the category ID in the category table. Thus, one row of

the category_equipment table unambiguously indicates the equipment belonging to the category. It should be noted that there are no unique fields in this table for two reasons: firstly, several marks of equipment can belong to one category at once, and secondly, one mark of equipment can belong to several categories at once. However, records that are a combination of the two mentioned fields are unique. Data integrity is ensured by foreign keys (FK) for each of the fields. The category_equipment table provides a many-to-many relationship between the pumping equipment and categories entities.

The Table 1 analysis of the main oil pump reference data shows that it is not possible to establish an unambiguous relationship between the brand of the pump and its characteristics. The reason is that pump performance can be determined not only by its mark, but also by combinations of other pump characteristics, such as impeller and impeller diameter. In order to determine the relationship between the mark of the pump and its combinations a table «combination» should be created. The table has two fields. The id_combination field contains unique identifiers of variations. The id_equipment field, like id_combination, takes an integer data type. The field is assigned a foreign key that refers to the equipment ID in the equipment table. Since one entity in the «equipment» table can correspond to one or more entities in the «combination» table, and one entity in the «combination» table can correspond only to one entity in the «equipment» table, the relationship between them can be defined as «one-to-many».

The names of the pump attributes are defined as separate entities and are placed in the «attribute_group» table (Fig. 1). The table consists of fields id_attribute_group and group_alias with data types INTERGER and TEXT, respectively. The id_attribute_group field with the primary key contains attribute IDs. The group_alias field contains a unique name for the attribute.

A separate table «attribute» is intended for storing attribute values (Fig. 1). Each attribute value has its own unique identifier, which is recorded in the id_attribute field. The attribute table also has an id_attribute_group field with a foreign key that points to the id of the attribute name id_attribute_group in the attribute_group table. This provides a structural link between the attribute table and the attribute value table. The attribute_alias field contains the attribute values. It should be noted that entries in the id_attribute_group and attribute_alias fields are not unique, but their combinations must be unique. The foreign key has the ON UPDATE CASCADE ON DELETE CASCADE property, which achieves the following behavior for the attribute table entries:

- changing any entry in the id_attribute_group field of the attribute_group table changes all entries in the id_attribute_group field of the attribute table that refer to the modified entry;
- removing any attribute in the attribute_group table removes all rows in the attribute table that refer to the removed attribute.

The relationship between the entities «attribute» and «attribute_group» is defined as «one-to-many», since the name of an attribute can correspond to many of its values, and the value of an attribute – only one name.

The relationship between attribute values and pump variation is specified using the «combination_attribute» table (Fig. 1). The table has two fields id_attribute id_com-

ination and integer data, which due to foreign keys indicate unique identifiers and attribute combinations, respectively. The combination_attribute table establishes a many-to-many relationship between entities in the attribute and combination tables. This relationship means the following: several pumps can correspond to one attribute value, and vice versa, several attributes can correspond to one pump variation at once.

Storage of equipment properties is organized using two tables – «feature» and «feature_value» (Fig. 1). The feature table consists of two fields – feature_alias and id_feature. The feature_alias field is textual and is used to store the names of equipment properties. The integer identifier of the property is written in the id_feature field. Both fields are unique.

The feature_value table is used to store property values. Values are written into the text field feature_value_alias. Due to the field id_feature and its foreign key each value of the property refers to its name in «feature» table.

There is a one-to-many relationship between the feature and feature_value tables, since multiple values can correspond to one property.

Connections between combinations, combination properties and their values are set using the «feature_combination» table (Fig. 1). Feature_combination table consists of three fields that take integer values. The id_combination field, using a foreign key, refers to a unique identifier in the «combination» table, which in turn is associated with the equipment identifier. The id_feature field of the «feature_combination» table refers to the unique identifier of the property in the «feature» table. The id_feature_value field of the feature_combination table indicates the identifier of the property value that is stored in the feature_value table. Each row of the table is unique. The feature_combination table provides a relationship between the pump variation entity and the many-to-many pump property entity, since multiple property values can correspond to a single pump variation and multiple pump variations can have one property.

When creating software products aimed at users from different countries, it is important to include support for multilingualism in the database architecture. Further, using the example of pump categories, various options for organizing multilingualism using a database are to be considered.

Fig. 2 shows a way to support multilingualism, in which a separate field is created for each language in the category table.

The disadvantage of this approach is the need to make changes to the table structure every time when adding a new language. This method can be used only in cases where the number of supported languages is known in advance and when each entity (in this example, the category name) must exist in all language variants.

id_category	id_parent	category_alias	title_uk	title_ru
1	0	pumps	насоси	насосы
2	1	centrifugal pumps	насоси відцентрові	насосы центробежные

Fig. 2. Illustration of support for multilingual pump categories by creating fields for each language

Another way of multilingualism can be implemented using information in a serialized form (Fig. 3).

The essence of the implementation is that information in JSON, XML, binary, or another format is written

to each field that requires translation. In this case, the written object can be an associative array or dictionary with keys in the form of language identifiers (for example, ru, en, uk) and values in the form of translations into the corresponding languages. The disadvantages of this implementation include the impossibility of using the query language for translation management. For example, in order to remove all translations of the English language a programming language (Python, PHP) should be used with the need to read and write the corresponding row of the table. The use of the SQL query language in this case is not possible.

id_category	id_parent	category_alias	title
1	0	pumps	{"uk": "насоси", "ru": "насосы"}
2	1	centrifugal pumps	{"uk": "насоси відцентрові", "ru": "насосы центробежные"}

Fig. 3. A way of multilingualism with complex serialized data

It is also possible to create separate records for each translation in the same table that contains the entity that needs localization (Fig. 4). In this case, a table is also created that describes the available languages.

The disadvantage is that for one entity several objects are created in the table at once, which leads to duplicate data that does not require translation and, in the presence of several relationships with other tables, greatly complicates the business logic of the software product.

There is also an example of multilingualism implementation, when for each language a separate table is created with fields that require translation (Fig. 5).

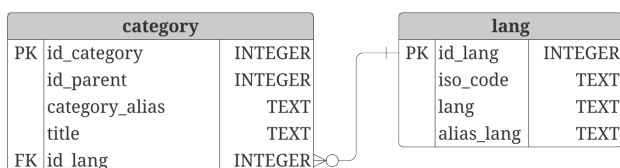


Fig. 4. Multilingual support with a separate table entry for each language

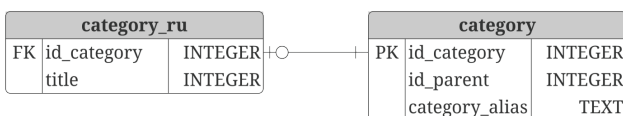


Fig. 5. Creating a translation table for each language

A one-to-one relationship is established between entities, since one translation line corresponds to one category and one category record can be associated with only one line in the translation table for the corresponding language.

The disadvantage of this structure is the need to create a new table when adding a new language. With this implementation, each entity requiring localization will have a set of tables with translations. For example, if two new languages are required to be added, assuming there are five entities, ten new language tables should be created, which seems extremely inconvenient.

The most common method for implementing multilingualism is to create two tables for each entity (Fig. 6), where the main table contains fields that do not depend on localization, and the second contains fields that require

translations. A table is also created with a list of available languages. Since it is not known in advance which languages will be used in the designed database, this method of multilingual support is chosen as the most flexible and easily scalable. The general principle of creating multilingualism is described using the example of categories. The entity «category» is stored in the main table «category». Each language is also a separate entity, which is recorded in the «lang» table. In the id_lang field with the primary key the language identifier is recorded. The iso_code field contains a unique two-letter language designator in accordance with ISO 639-1, for example, ru. The lang field is intended to store the name of the language in the original, for example – Russian. The alias_lang field contains the name of the language in English, for example – russian.

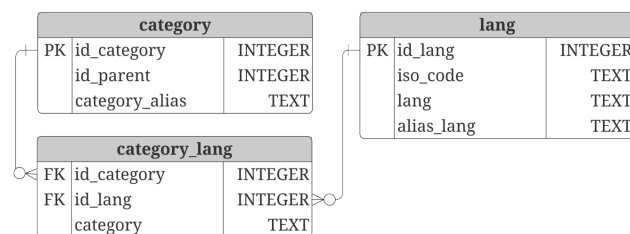


Fig. 6. Multilingualism using localization tables

The «category_lang» table contains fields that require translation. Because of both fields id_category and id_lang each row of the table determines which category and which language the translation relates to. The «category_lang» table provides a many-to-many relationship between the category entity and the language entity, since several languages can correspond to one entity in the «category» table and one language entity can be used for several categories.

By the same principle, language tables are created for other entities, for example, combinations and properties.

6. Research results

As an example let's consider the data structure of the pump NM 2500-230 and its two variations depending on the diameter of the impeller (Fig. 7). Each variation has its own unique identifier, which is defined in the id_combination field of the «combination» table. This table specifies the relationship between the pump mark id_equipment identifier and its variation identifier (combination). For each variation and language in the «combination_lang» table a short «short_description» and a long «description» of the pump are indicated. Each row of the «combination_lang» table with the «id_lang» identifier indicates the language for which the corresponding translations are created. The available languages are defined in the «lang» table. Referential data integrity is ensured by foreign keys, for example, deleting a pump in the equipment table automatically removes all its variations in the combination table and translations in the combination_lang table. Likewise, for example, deleting the Russian language from the «lang» table removes all Russian translations in the «combination_lang» table.

The tables in Fig. 8 show which parameters (attributes) the pump variations depend on.

The relationships between attributes and variations are defined in the «combination_attribute» table. For example,

variations with identifier 1 correspond to the values of attributes with identifiers 1 and 3. Attribute values are defined in the «attribute» table. So, identifiers of attribute values 1 and 3 correspond to values 0.5 and 425.

equipment		combination	
id_equipment	marking	id_combination	id_equipment
1	nm_2500_230	1	1
		2	1

lang			
id_lang	iso_code	lang	alias_lang
1	uk	українська	ukrainian
2	1	русский	russian

combination_lang			
id_combination	id_lang	description	short_description
1	2	Main pump NM 2500-230, Dw=425 mm, n=2973 rpm	NM 2500-230
1	1	Main pump NM 2500-230, Dw=425 mm, n=2973 rpm	NM 2500-230
2	2	Main pump NM 2500-230, Dw=405 mm, n=2973 rpm	NM 2500-230
2	1	Main pump NM 2500-230, Dw=405 mm, n=2973 rpm	NM 2500-230

Fig. 7. Pump variations

combination_attribute		combination	
id_attribute	id_combination	id_combination	id_equipment
1	1	1	1
3	1		
2	2		
4	2		

attribute_group		
id_attribute_group	group_alias	
1	rotor	
2	D2	

attribute		
id_attribute	id_attribute_group	attribute_alias
1	1	0.5
2	1	0.7
3	2	425
4	2	405

Fig. 8. Pump attributes

The relationship of the «attribute» table to the attribute_group table is provided by the foreign key of the id_attribute_group field. For example, the values of attributes with identifiers 1 and 3 correspond to attributes with identifiers 1 and 2, where 1 and 2 correspond to the identifiers of the rotor and the diameter of the pump impeller, respectively.

The names and values of pumps characteristics are stored in the tables «feature» and «feature_value», respectively (Fig. 9).

The feature_combination table defines the relationship between pump variation and pump performance. For example, a pump variation with id_combination=1 has two properties with IDs 1 and 2. Property values are indicated by identifiers in the id_feature_value field. Thus, it can be determined that the reviewed pump has two properties - empirical coefficients H_0 and b in formula (1) with values $H_0=246.7$ m and $b=16.8 \cdot 10^{-6}$ hr²/m⁵.

The process of obtaining specific equipment from the database of the proposed architecture can be formalized through the use of SQL queries. Fig. 9 shows the tables

of the database, which store the characteristics of the pump NM 2500-230 in two variations:

- 1) rotor 0.5, impeller diameter $D_2=425$ mm;
- 2) rotor 0.7, impeller diameter $D_2=405$ mm.

combination		feature	
id_combination	id_equipment	id_feature	feature_alias
1	1	1	H0
2	1	2	b

feature_combination		
id_feature	id_combination	id_feature_value
1	1	1
2	1	3
1	2	2
2	2	4

feature_value		
id_feature_value	id_feature	feature_value_alias
1	1	246.7
2	1	248.7
3	2	16.8e-6
4	2	7.61e-6

Fig. 9. Pump characteristics

As an example, Table 2 shows a request for obtaining the characteristics of the pump NM 2500-230 with a rotor of 0.5 and an impeller diameter of 425 mm.

The proposed query has a quite complex structure, since it contains a subquery enclosed between lines 4 and 14 (Table 2). The task of the nested query is to find the unique identifier of the pump variation id_combination, which in this example is equal to one (Table 3).

In general, the search for the id_combination identifier is carried out according to two criteria: equipment labeling and attribute values. In this example, the specified criteria are reflected in lines 11–13 of the query (Table 2).

Table 2

SQL query for obtaining characteristics of pump NM 2500-230 with rotor 0.5 and $D_2=425$ mm

Row	Query
1	SELECT
2	comb.id_combination, comb.id_equipment, comb.marking, fc.id_feature, fc.id_feature_value, f.feature_alias, fv.feature_value_alias
3	FROM (
4	SELECT
5	count(c.id_combination) AS combinations_count, e.*, c.id_combination
6	FROM equipment e
7	LEFT JOIN combination c ON c.id_equipment=e.id_equipment
8	INNER JOIN combination_attribute ca ON ca.id_combination=c.id_combination
9	LEFT JOIN attribute a ON a.id_attribute=ca.id_attribute
10	LEFT JOIN attribute_group ag ON ag.id_attribute_group=a.id_attribute_group
11	WHERE e.marking='nm_2500_230'
12	AND (ag.group_alias='rotor' AND a.attribute_alias='0.5')
13	OR (ag.group_alias='D2' AND a.attribute_alias='425')
14	GROUP BY ca.id_combination HAVING combinations_count=2
15) AS comb
16	LEFT JOIN feature_combination fc ON fc.id_combination=comb.id_combination
17	LEFT JOIN feature f ON f.id_feature=fc.id_feature
18	LEFT JOIN feature_value fv ON fv.id_feature_value=fc.id_feature_value

Table 3

SQL query result

Table fields						
id_combination	id_equipment	marking	id_feature	id_feature_value	feature_alias	feature_value_alias
1	1	nm_2500_230	1	1	H0	246.7
1	1	nm_2500_230	2	3	b	16.8e-6

The COUNT aggregation function after the SELECT statement (line 5) counts the number of rows corresponding to each pump variation. After calling the COUNT function, the combinations_count alias is specified, under which the result of the function will be available. Due to the GROUP BY clause (line 14), rows with the same id_combination values are combined into one. Since several pump variations can meet the criteria specified in lines 11–13 at once, to obtain the required one, it is necessary use the HAVING operator with the combinations_count field value equal to two (by the number of attributes). As a result of the work of the nested query, a table will be formed accessible by the alias «comb» and consisting of one row. After the left join (line 16) of the feature_combination table, which connects the pump characteristics with its variations, a table is formed, consisting of two rows according to the number of characteristics (H0, b). At this stage, identifiers of properties and their values are obtained. With the left join, bind the lines with the names of the properties and their values (lines 17 and 18). Finally, the resulting table containing the characteristics of the pump is obtained (Table 2). In this table one line corresponds to one characteristic.

The structure of the developed database is posted on the GitHub web service [27] and is available for making changes.

7. SWOT analysis of research results

Strength. Compared to manual input of entry data into a software product, their storage in electronic form can significantly reduce the complexity of calculating the characteristics of oil pumps and operating modes of oil pipelines. The advantage of relational databases over spreadsheets is as follows:

- free;
- allow to store significant amounts of data and provide high performance;
- can be easily used in conjunction with popular programming languages (Python, PHP, Java, etc.) and packages of applied mathematical programs (Matlab, Mathematica, Scilab);
- expand the scope for scaling projects, for example, multilingual support can be implemented with a table structure that best suits the requirements of the project;
- use a convenient query language SQL.

Weaknesses. Relational databases use the SQL query language directly or through a visual editor to add, modify and delete data (phpMyadmin, dbviewer, dbeaver). This is not convenient enough in both cases. To address the problem, data may be imported from files in json, xml, csv or xls format, which requires additional time-consuming development of import modules.

Opportunities. The research results can be used in the design of databases of oil and gas equipment of various types, for example, heat exchangers, dust collectors, gas

pressure regulators of gas distribution stations, centrifugal blowers, gas turbine engines and oil reservoirs. Due to the well-thought-out architecture of the database, it is possible to create both highly specialized programs designed to solve one problem, and large software systems for calculating complex technological processes.

Threats. Working with relational databases places higher demands on the qualifications of a program developer, as it requires knowledge of designing database architectures, composing tables and relationships between them, placing indexes and determining keys.

8. Conclusions

1. Analysis of the tasks of pipeline transportation of oil and oil products and modern programs for calculating the operating modes of oil pipelines showed the relevance of creating databases of pumping equipment. It is shown that due to the complex data structure of the pumps, it is preferable to use relational databases than Excel spreadsheets. When designing the physical model of the database, parent and child entities were selected in such a way that it was possible to reflect all information about the pumping unit: design features, scope, pump characteristics, depending on its variations for replaceable rotors. In the process of modeling, data integrity rules are determined, and relationships of the «one-to-many» and «many-to-many» types are established between the entities of the physical model. The developed database architecture provides modifiability, scalability, modularity and effective use in programs for calculating the operating modes of oil pipelines.

2. An overview of various ways of multilingualism organizing in a database has been made. To achieve the aim of research, it seems most preferable to use a structure in which each entity is stored in two tables. One table stores data that does not depend on translations, and the other contains data with translations.

References

1. *Scheme of Ukraine's Main Oil Pipelines System.* Ukrtransnafta. Available at: <https://www.ukrtransnafta.com/en/principal-scheme-of-pp-in-ukraine/>
2. Kniazieva, V., Kanyuk, G., Mezerya, A., Andreev, A. (2019). Analysis of normative documents for ensuring efficient work of pumped installations of main oil pipelines. *Bulletin of the National Technical University «KhPI» Series: New Solutions in Modern Technologies*, 5 (1330), 27–33. doi: <http://doi.org/10.20998/2413-4295.2019.05.04>
3. Serediuk, M. D. (2019). Mathematical modeling of the characteristics of oil pumps, taking into account the viscosity of the transported oil. *Oil and Gas Power Engineering*, 1 (31), 54–64. doi: [http://doi.org/10.31471/1993-9868-2019-1\(31\)-54-64](http://doi.org/10.31471/1993-9868-2019-1(31)-54-64)
4. Serediuk, M. D., Hryhoriyskiy, S. Ya. (2018). Vyznachennia propusknoi zdatnosti ta enerhovytratnosti dvonytkovykh naftoprovodnykh system. *Mizhnarodnyi naukovyi zhurnal Internauka*, 1 (3 (43)), 81–87.

5. Serediuk, M. D., Hanzha, M. Ye. (2017). Vybir enerhoefektyvnykh rezhymiv ekspluatatsii mahistralnykh hazoprovodiv za yikh nepovnoho zavantazhennia. *Naukovyi visnyk IFNTUNH*, 1 (42), 67–72.
6. Serediuk, M. D., Hrudz, V. Ya. (2007). Shliakhy pidvyshchennia efektyvnosti ta zmeshennia enerhovytratnosti protsesiv transportuvannia ta zberihannia nafty i hazu. *Naftohazova enerhetyka*, 2 (3), 24–31. Available at: <http://elar.nung.edu.ua/bitstream/123456789/1329/3/1705p.pdf>
7. Kutukov, S. E. (2002). *Informatsionno-analiticheskie sistemy magistralnykh truboprovodov*. Moscow: SIP RIA, 324.
8. Kaniuk, G. I., Andreev, A. V., Mezeria, A. Iu., Kniازهva, V. N. (2015). Analiz rezervov energosberezheniia pri upravlenii nasosnymi agregatami nefteperekachivaiushchikh stantsii Ukrainy. *Integrovani tekhnologii ta energoberezheniia*, 4, 3–14. Available at: http://repository.kpi.kharkov.ua/bitstream/KhPI-Press/29242/1/ITE_2015_4_Kanyuk_Analiz_rezervov.pdf
9. Serediuk, M. D., Yakymiv, Y. V., Lisafin, V. P. (2002). *Truboprovodnyi transport nafty i naftoproduktiv*. Ivano-Frankivsk: Kremenchuk, 517.
10. McAllister, E. W. (2013). *Pipeline rules of thumb handbook: a manual of quick, accurate solutions to everyday pipeline engineering problems*. Gulf Professional Publishing, 806.
11. Bykov, L. I., Mustafin, F. M., Rafikov, S. K., Nechval, A. M., Lavrentev, A. E. (2006). *Tipovye raschety pri sooruzhenii i remonte gazonefteprovodov*. Saint Petersburg: Nedra, 824.
12. Tugunov, P. I., Novoselov, V. F., Korshak, A. A., Shammazov, M. A. (2002). *Tipovye raschety pri proektirovanii i ekspluatatsii neftebaz i nefteprovodov*. Ufa: OOO «DizainPoligrafServis», 658.
13. Serediuk, M. D., Liuta, N. V. (2004). *Tekhnolohichni rozrakhunkiv rezhymiv roboty nasosnykh stantsii mahistralnykh truboprovodiv*. Ivano-Frankivsk: IFNTUNH, 151. Available at: <http://194.44.112.13/chytalna/473/index.html>
14. Nechval, A. M. (2005). *Osnovnye zadachi pri proektirovanii i ekspluatatsii magistralnykh nefteprovodov*. Ufa: Izd-vo UGNTU, 81.
15. Muratova, V. I. (2014). *Otsenka vliianiia protivoturbulentnykh prysadok na gidravlicheskuuiu effektivnost nefteproduktoprovodov*. Ufa, 162. Available at: <http://earthpapers.net/otsenka-vlianiya-protivoturbulentnykh-prysadok-na-gidravlicheskuuiu-effektivnost-nefteproduktoprovodov>
16. Al-Dandal, R. S. (2016). *Ratsionalni rezhymy ekspluatatsii mahistralnykh naftoproduktoprovodiv z vykorystanniam proty-turbulentnykh prysadok*. Ivano-Frankivsk, 141. Available at: <http://elar.nung.edu.ua/handle/123456789/6952>
17. Didkovskaia, A. S., Lure, M. V. (2002). *Kompiuternii praktikum po truboprovodnomu transportu nefti i nefteproduktov*. Moscow: GUP Izd-vo «Neft i gaz» RGU nefti i gaza im. I. M. Gubkina, 128.
18. *Informatsiia o paketakh kompiuternykh programm, razrabotannykh v nauchno-issledovatel'skoi laboratorii neftegazovoi gidrodinamiki RGU nefti i gaza im. I. M. Gubkina*. Available at: https://www.gubkin.ru/faculty/pipeline_network_design/chairs_and_departments/designing_and_operation_gasoil_pipeline/lurie/lab3.php?special_version=Y
19. Bakhtizin, R. N., Pirogov, A. N., Nechval, A. M., Pirogov, N. E., Sukharnikov, L. V. (2018). Modelirovanie i algoritmy gidravlicheskogo rascheta statsionarnogo rezhima raboty nefteproduktoprovodnykh sistem. *Transport i khranenie nefteproduktov i uglevodorodnogo syria*, 3, 27–31. doi: <http://doi.org/10.24411/0131-4270-2018-10302>
20. *Pump selection program*. Available at: http://www.agrovodcom.ru/prog_korvet.php
21. Linoff, G. S. (Ed.) (2015). *Data analysis using SQL and Excel*. John Wiley & Sons, 745. doi: <http://doi.org/10.1002/9781119183419>
22. Bendre, M., Wattanawaroon, T., Mack, K., Chang, K., Parameswaran, A. (2019). Anti-freeze for large and complex spreadsheets: Asynchronous formula computation. *Proceedings of the 2019 International Conference on Management of Data*. Amsterdam, 1277–1294. Available at: <https://people.eecs.berkeley.edu/~adityagp/papers/dataspread-async.pdf>
23. Bendre, M., Wattanawaroon, T., Rahman, S., Mack, K., Liu, Y., Zhu, S. et. al. (2019). Faster, higher, stronger: Redesigning spreadsheets for scale. *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, 1972–1975. Available at: <https://people.eecs.berkeley.edu/~adityagp/papers/dataspread-demo2.pdf>
24. Malecha, G., Morrisett, G., Shinnar, A., Wisnesky, R. (2010). Toward a verified relational database management system. *Proceedings of the 37th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, 237–248. doi: <http://doi.org/10.1145/1706299.1706329>
25. Vicknair, C., Macias, M., Zhao, Z., Nan, X., Chen, Y., Wilkins, D. (2010). A comparison of a graph database and a relational database: a data provenance perspective. *Proceedings of the 48th annual Southeast regional conference*, 1–6. Available at: https://john.cs.olemiss.edu/~ychen/publications/conference/vicknair_acmse10.pdf
26. Kirillov, V. V., Gromov, G. Iu. (2009). *Vvedenie v relatsionnye bazy dannykh*. Saint Petersburg: BKHV-Peterburg, 464.
27. *Pumps database – GitHub repository*. Available at: https://github.com/Buzovskiy/equipment_database

Buzovskiy Vitalii, PhD, Assistant, Department of Oil and Gas Technologies, Engineering and Heat Power Engineering, Odessa National Academy of Food Technologies, Ukraine, ORCID: <http://orcid.org/0000-0001-6718-5001>, e-mail: buzovskiy.v@gmail.com

Kologrivov Mykhail, PhD, Associate Professor, Department of Oil and Gas Technologies, Engineering and Heat Power Engineering, Odessa National Academy of Food Technologies, Ukraine, ORCID: <http://orcid.org/0000-0003-1959-8615>, e-mail: klgrvmm@rambler.ru

Antonova Alfiia, PhD, Associate Professor, Department of Information Technologies and Cybersecurity, Odessa National Academy of Food Technologies, Ukraine, ORCID: <http://orcid.org/0000-0003-3599-0983>, e-mail: allaantonova62@gmail.com