

УДК 621.6:004.6

DOI: 10.15587/2706-5448.2020.212752

## РАЗРАБОТКА БАЗЫ ДАННЫХ НАСОСОВ НЕФТЕБАЗ И МАГИСТРАЛЬНЫХ НЕФТЕПРОВОДОВ

Бузовский В. П., Кологривов М. М., Антонова А. Р.

Объектом исследований является модель данных, которые характеризуют насосные агрегаты магистральных нефтепроводов. В работе рассматривается создание базы данных насосов и их характеристик как составной части программных комплексов, предназначенных для поиска источников энергосбережения на трубопроводном транспорте нефти и нефтепродуктов.

Приведен обзор существующих программ для расчета совместной работы нефтеперекачивающей станции и трубопровода. Показано, что большинство программ являются недоступными для анализа применяемых технологий хранения данных, или предполагают ввод входных данных вручную без использования баз данных. Выполнен сравнительный анализ электронных таблиц и реляционных баз данных. Показано, что реляционные базы данных в большей мере отвечают требованиям удобства работы с данными, доступности, масштабируемости и производительности. Представлена физическая модель реляционной базы данных. Установлены родительские и дочерние сущности, позволяющие полностью отразить всю информацию о насосном агрегате, а именно:

- коэффициенты гидравлической характеристики в зависимости от диаметра ротора рабочего колеса;
- характеристики, зависящие только от марки насоса;
- классификацию насоса по конструктивным особенностям или назначению. Между сущностями определены типы связей и правила ссылочной целостности при удалении или обновлении данных. Для управления данными предложена система SQLite, обеспечивающая манипулирование данными на языке запросов SQL и не требующая разработки дополнительного программного обеспечения. Выполнен анализ возможных способов организации многоязычности за счет базы данных. Предложен текст SQL-запроса, позволяющий выбрать насос с его характеристиками по марке и диаметру рабочего колеса. Предложена структура базы данных, которая позволяет хранить информацию об оборудовании нефтепроводов различного назначения, что дает возможность автоматизированных расчетов сложных технологических процессов. База данных насосных агрегатов представляет собой проект с открытым исходным кодом, который выложен на общедоступный веб-сервис GitHub.

**Ключевые слова:** магистральный нефтепровод, нефтяной насос, гидравлическая характеристика, база данных, модель сущность-связь.

## 1. Введение

Системы распределения нефти и нефтепродуктов подвержены постоянным изменениям, которые вызваны следующими причинами:

- открываются новые нефтяные месторождения и изменяются объёмы нефтедобычи на существующих;
- нефтеперерабатывающие заводы могут наращивать или уменьшать объёмы нефтепереработки, что приводит к изменению объемов поставки нефти;
- страны, которые импортируют нефть или транспортируют её через свою территорию могут проводить диверсификацию поставщиков нефти и нефтепродуктов, что также может оказывать влияние на работу их нефтетранспортных систем.

Например, в Украине компанией «Укртранснафта» принято ряд стратегических направлений. Одно из приоритетных – содействие интеграции Украины в проект Евразийского нефтетранспортного коридора, который предполагает строительство нефтепровода Броды-Плоцк. На данный момент трубопроводная система Украины работает в недогруженном режиме. Вызвано это значительным уменьшением объемов производства нефтепродуктов на местных нефтеперерабатывающих заводах и уменьшением объемов транзита нефти через свою территорию. Тем не менее, масштабы нефтетранспортной системы Украины велики. В работе задействовано 4767,4 км магистральных нефтепроводов, морской нефтяной терминал «Южный» и 28 нефтеперекачивающих станций (НПС) [1].

Нефтетранспортные системы потребляют значительное количество энергии. Энергоресурсы ограничены и дороги, поэтому страны закрепляют стратегии энергосбережения на законодательном уровне.

Например, в Украине принят закон «Об энергосбережении» и ряд других нормативных документов [2], которые предписывают организационно-технические мероприятия по оптимизации режимов работы нефтепроводов на недогруженных участках.

Для обеспечения требуемой производительности на эксплуатируемых трубопроводах с учетом критериев энергоэффективности применяют варианты (оптимизационные) расчеты. Такие расчеты принято выполнять с использованием программных продуктов.

Хотя в литературных источниках имеются упоминания о применении компьютерных программ для решения множества задач транспорта и хранения нефти [3–5], сведения об используемых информационных технологиях при разработке этих программ отсутствуют. Также отсутствуют рекомендации о построении баз данных оборудования, применяемого при транспорте и хранении нефти и нефтепродуктов. Наличие базы данных предоставляет возможность организации единого доступа к характеристикам оборудования и их управления, а также обеспечивает возможность применения общепринятых шаблонов проектирования программных продуктов. Значительный интерес к накоплению и структурированию информации о нефтяном технологическом оборудовании могут проявлять производственные предприятия, проектные организации, научные и учебные учреждения. Ввиду недостаточной

освещенности вопросов, связанных с хранением информации о технологическом оборудовании нефтепроводов, описание принципов построения баз данных представляется актуальной задачей.

## **2. Объект исследований и его технологический аудит**

*Объектом исследований* является модель данных, которые характеризуют насосные агрегаты магистральных нефтепроводов. В настоящее время магистральные насосы в Украине работают в недогруженном состоянии. Приводом насосов являются электрические двигатели, потребляемая мощность которых зависит от производительности насоса. Режимы работы насосов далеки от оптимальных [6], что может приводить к завышенному удельному энергопотреблению привода. В общем случае причины малой загрузки нефтепроводов могут быть следующими:

- недостаточное количество перекачиваемой жидкости в резервуарах головной НПС или переполнение резервуаров конечного пункта;
- ремонт трубопровода без остановки перекачки;
- изменение вязкости перекачиваемой нефти;
- технологические ограничения величины давления на входе и выходе перекачивающих станций (ПС);
- наличие путевых сбросов.

Поскольку большая часть эксплуатационных расходов МН обусловлена работой электродвигателей [7], то и положительный эффект от реализации потенциала энергосбережения насосных агрегатов также является наибольшим [8]. При выборе рациональных режимов эксплуатации нефтепроводов рассматривают целый ряд мероприятий:

- регулирование производительности нефтепровода изменением количества работающих насосов на ПС;
- регулирование изменением схемы работы насосов на ПС;
- байпасирование;
- регулирование путем обтачивания рабочих колес насоса;
- частотное регулирование насоса.

Для поиска режимов, отвечающих критериям наибольшей энергоэффективности и наименьшей капиталоемкости, просчитывают сотни вариантов работы нефтепровода. Для выполнения подобных расчетов, как правило, пишут компьютерные программы. Значительным является не только объем расчетных данных, но и данных, подаваемых на вход программы. К таким относится характеристика насосного оборудования и линейной части МН. Объясняется это тем, что на действующих нефтепроводах работают насосы различных марок и с разнообразным типоразмером рабочих колес. Данные о насосах обычно хранятся в табличном (табл. 1) или в графическом видах в книгах и каталогах [9–11].

Таблица 1

Справочные данные для насоса НМ 2500-230 (Украина) [12]

Типоразмер насоса	Ротор	$D_2$ , мм	Коэффициенты в формуле (1)		Коэффициенты в формуле (2)		
			$H_0$ , м	$10^6 \cdot b$ , $\text{ч}^2/\text{м}^5$	$10^2 \cdot c_0$	$10^4 \cdot c_1$ , $\text{ч}/\text{м}^3$	$10^8 \cdot c_2$ , $\text{ч}^2/\text{м}^6$
НМ 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

При гидравлических расчетах нефтепровода используются данные для описания характеристики насоса:

– зависимости полного напора  $H$ :

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

– коэффициента полезного действия (КПД)  $\eta_n$  от производительности  $Q$  ( $\text{м}^3/\text{ч}$ ):

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

где  $H$ ,  $\eta_n$  – полный напор (м ст. ж.) и КПД насоса при подаче  $Q$  ( $\text{м}^3/\text{ч}$ ) соответственно;  $H_0$ ,  $b$ ,  $c_0$ ,  $c_1$ ,  $c_2$  – эмпирические коэффициенты.

Ручной ввод этих данных в программу представляется неудобным, поскольку требует концентрации внимания и значительных трудовых затрат.

### 3. Цель и задачи исследования

Целью работы является разработка базы данных насосных агрегатов магистральных нефтепроводов.

Для достижения цели необходимо решить следующие задачи:

1. Проектирование физической модели базы данных, определение родительских и дочерних сущностей, типов связей между ними и правил целостности данных с учетом различных вариаций марок насосов.

2. Разработка способов организации многоязычности базы данных насосных агрегатов.

### 4. Исследование существующих решений проблемы

Большинство существующих программ, посвященных расчету режимов работы нефтепроводов недоступны или обходятся без баз данных.

Например, программа NS1 по расчету пропускной способности прилегающего к головной насосной станции участка нефтепровода [13] предлагает пользователю вручную через консоль вводить данные о насосах и

трубопроводе. Чтобы повторить расчет при измененной величине одного из входных параметров, необходимо заново вводить остальные параметры, что представляется неудобным. Ввод данных по насосным агрегатам из книг и каталогов [9, 11, 12] требует значительных трудовых затрат. Скорость чтения низкая, возможны ошибки.

На языке BASIC разработана программа RABT.BAS для расчета совместной работы нефтепровода и НПС [14]. Программа работает без базы данных. Ввод коэффициентов гидравлических характеристик насосов производится вручную.

В работе [15] приведен текст программы по расчету режима работы нефтепровода с применением противотурбулентных присадок. Данные по присадкам считываются с TXT-файлов. Коэффициенты характеристики подпорного и магистрального насосов присваиваются переменным напрямую. База данных не используется.

На Pascal реализована программа для расчета нестационарных режимов магистральных нефтепроводов [16]. В программе отсутствует код, отвечающий за соединение с базой данных. Данные по насосам и трубопроводу задаются вручную путем присваивания переменным.

Программа «Транзит» для расчета режимов работы нефтепровода [17, 18] предоставляет удобный способ задания основных и подпорных насосов. Программа имеет интерфейс. Выбор насосов осуществляется при помощи выпадающего списка.

Программа «Транзит» платная. Её исходный код не доступен. В литературных источниках отсутствуют сведения о реализуемом способе хранения данных, что исключает возможность использования опыта разработчиков программы «Транзит» при проектировании новых баз данных.

В ходе исследований по разработке и совершенствованию моделирования и построения эффективных алгоритмов гидравлического расчета нефтепроводных систем создан промышленный комплекс программ по гидравлическому расчету режимов работы магистральных нефтепроводов [19]. Программный комплекс сдан в промышленную эксплуатацию, однако в свободном доступе его нет и сведения о способе организации данных также отсутствуют.

Широкое распространение получил способ хранения информации при помощи электронных таблиц MS Excel. Например, в программе [20] данные о насосном оборудовании организованы в таблицы, в которой каждому столбцу соответствует определенная характеристика насоса. Благодаря наличию встроенной фильтрации программа позволяет осуществить выбор насоса среди множества модификаций. Из недостатков электронных таблиц следует отметить следующие:

- MS Excel является платным программным продуктом, поэтому доступен не везде;

- многие языки программирования имеют библиотеки для работы с файлами Excel, однако код для выборки данных представляется более громоздким и менее наглядным по сравнению, например, с SQL-запросами к реляционным базам данных;

– при наличии значительного количества строк скорость обработки файлов Excel низкая [21–23].

Для реализации цели работы наиболее предпочтительным представляется использование реляционных баз данных. Их повсеместное [24, 25] использование обусловлено следующими причинами [26]:

– реляционная модель организации данных достаточно простая и в то же время мощная, подходит для решения подавляющего большинства задач хранения и обработки данных;

– эта модель характеризуется простотой структуры данных и удобным для пользователя табличным представлением;

– обеспечивает целостность данных, масштабируемость и независимость от приложения, которое использует данные.

Выбор системы управления реляционными базами данных (СУРБД) не является принципиальным. Для малых проектов удобным представляется использование легковесной СУРБД SQLite. Ее преимущество в упрощенной структуре данных и отсутствии необходимости использования сервера базы данных.

## 5. Методы исследования

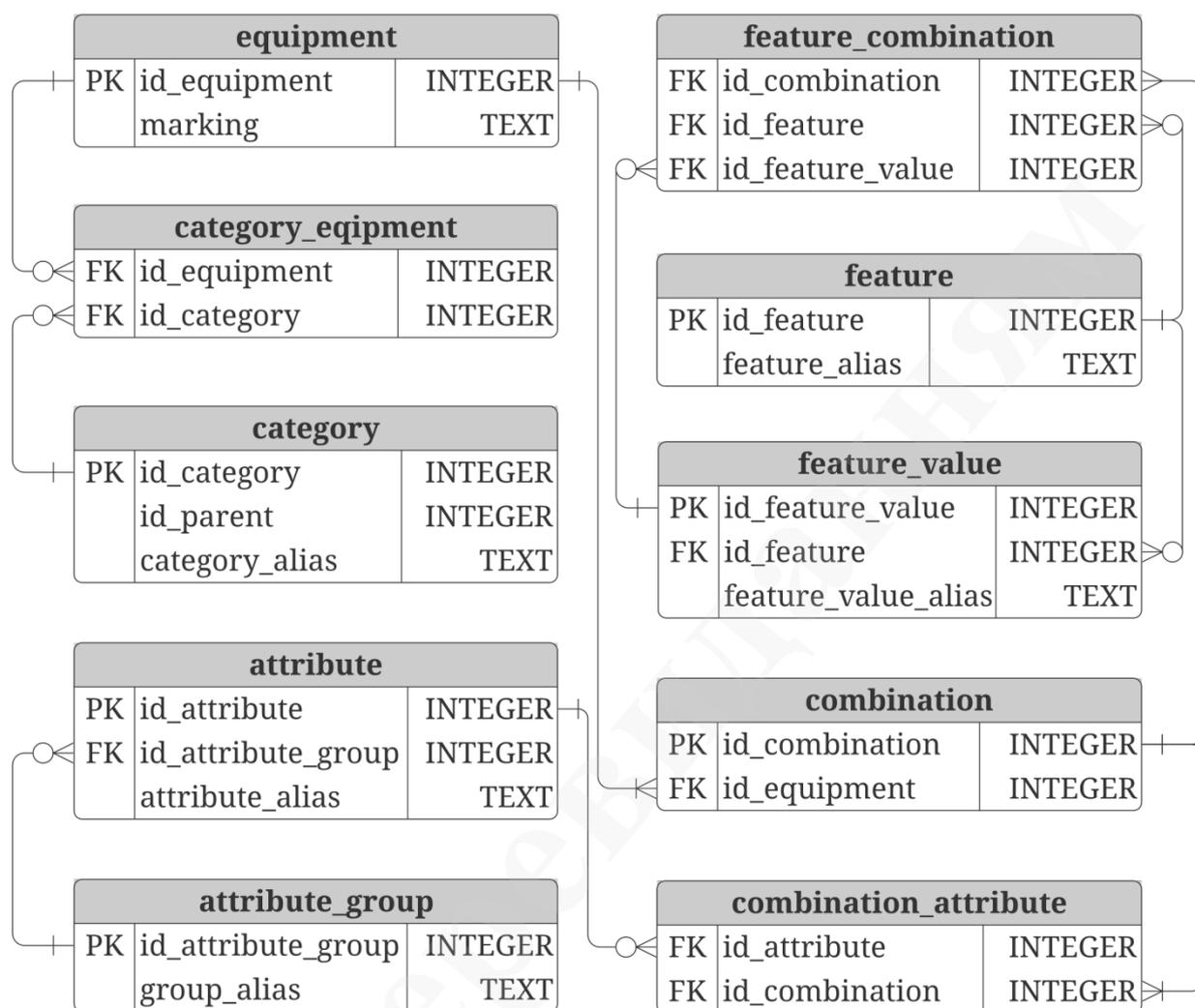
Для удобства структурирования данных разделим характеристики насоса на атрибуты и свойства. Атрибутами будем считать характеристики, которые определяют различные комбинации (вариации) марки оборудования. К примеру, у насоса НМ 2500-230 можно выделить такие атрибуты, как номер ротора и диаметр рабочего колеса  $D_2$ . Остальные характеристики определяются, как свойства. Для примера, комбинация атрибутов «ротор=1» и « $D_2=440$ » однозначно определяет свойство  $H_0=281,5$  м ст. ж (табл. 1).

При создании реляционной базы данных насосных агрегатов определены следующие родительские сущности: «оборудование» (equipment), «категория оборудования» (category), «свойство оборудования» (feature) и «название атрибута» оборудования (attribute\_group).

Для хранения различных вариаций марки насоса, значений свойств и атрибутов определены такие дочерние сущности: «комбинация» (combination), «значение свойства» (feature\_value) и «значение атрибута» (attribute). На рис. 1 представлена физическая модель базы данных, которая описывает типы данных и связи между сущностями.

Между сущностями определены типы связей и правила ссылочной целостности при удалении или обновлении данных родительских сущностей. Подробнее рассмотрим каждую из таблиц.

Таблица «оборудование» содержит марку насоса. Как будет показано далее, саму марку насоса удобно хранить в отдельной таблице, содержащей переводы на различные языки. Уникальность каждой записи обеспечивается за счет поля id\_equipment с первичным ключом (PK) и целочисленными значениями. Поле marking принимает текстовый тип данных и служит для хранения псевдонима марки оборудования.



**Рис. 1.** Физическая модель реляционной базы данных насосных агрегатов магистральных нефтепроводов

При транспортировке нефти и нефтепродуктов могут применяться насосы различных типов. Информацию о возможных типах насосов удобно хранить в таблице «category» (рис. 1). При разработке её структуры следует учитывать критерий масштабируемости, что подразумевает возможность создания неограниченного количества подкатегорий. Например, категория «насосы» может иметь подкатегории «насосы центробежные», «насосы шестеренные», «насосы поршневые», которые можно обозначить как подкатегории второго уровня. В свою очередь, среди «насосов центробежных» можно выделить «насосы секционные типа НМ» и «насосы спиральные типа НМ», которые являются подкатегориями третьего уровня. В поле `id_category` с первичным ключом содержится уникальный целочисленный идентификатор категории. Целочисленная запись поля `id_parent` указывает на идентификатор родительской категории `id_category`. Поле `category_alias` является не обязательным. Оно создано для более понятного визуального восприятия таблицы и содержит уникальные псевдонимы категорий, которые в данном случае соответствуют переводам категорий на английский язык.

Связи между насосами и категориями определены в таблице «category\_equipment» (рис. 1). Её поле id\_equipment с целочисленными значениями указывает на уникальный идентификатор оборудования в таблице equipment. Поле id\_category указывает на идентификатор категории в таблице category. Таким образом, одна строка таблицы category\_equipment однозначно указывает на принадлежность оборудования категории. Следует отметить, что в данной таблице нет уникальных полей по двум причинам: во-первых, одной категории может принадлежать сразу несколько марок оборудования, во-вторых, одна марка оборудования может принадлежать сразу нескольким категориям. Однако уникальными являются записи, представляющие собой комбинацию из двух упомянутых полей. Целостность данных обеспечивается за счет внешних ключей (FK) для каждого из полей. Благодаря таблице category\_equipment между сущностями «насосное оборудование» и «категории» существует связь «многие-ко-многим».

Анализ табл. 1 справочных данных нефтяного магистрального насоса показывает, что установить однозначную связь между маркой насоса и его характеристиками не представляется возможным. Причина в том, что характеристики насоса могут определяться не только его маркой, но и комбинациями других его характеристик, например, ротора и диаметра рабочего колеса. Для того, чтобы определить связь между маркой насоса и его комбинациями, создадим таблицу «combination». Таблица состоит из двух полей. В поле id\_combination записываются уникальные идентификаторы вариаций. Поле id\_equipment также как и id\_combination принимает целочисленный тип данных. Полю присвоен внешний ключ, который ссылается на идентификатор оборудования в таблице equipment. Поскольку одной сущности в таблице «equipment» может соответствовать одна и более сущностей в таблице «combination», а одной сущности в таблице «combination» может соответствовать только одна сущность в таблице «equipment», то связь между ними можно определить как «один-ко-многим».

Названия атрибутов насосов определим как отдельные сущности и поместим их в таблицу «attribute\_group» (рис. 1). Таблица состоит из полей id\_attribute\_group и group\_alias с типами данных INTEGER и TEXT, соответственно. В поле id\_attribute\_group с первичным ключом записываются идентификаторы атрибутов. Поле group\_alias содержит уникальное название атрибута.

Для хранения значений атрибутов предназначена отдельная таблица «attribute» (рис. 1). Каждое значение атрибута имеет свой уникальный идентификатор, который записывается в поле id\_attribute. Таблица attribute также имеет поле id\_attribute\_group с внешним ключом, который указывает на идентификатор названия атрибута id\_attribute\_group в таблице attribute\_group. Таким образом, обеспечивается структурная связь между таблицей атрибутов и таблицей их значений. В поле attribute\_alias записываются значения атрибутов. Следует отметить, что записи в полях id\_attribute\_group и attribute\_alias не являются уникальными, однако уникальными должны быть их комбинации. Для внешнего ключа определено свойство ON UPDATE CASCADE ON DELETE CASCADE, благодаря которому достигается следующее поведение

записей таблицы attribute:

– при изменении какой-либо записи в поле id\_attribute\_group таблицы attribute\_group изменяются все записи в поле id\_attribute\_group таблицы attribute, которые ссылаются на изменяемую запись;

– при удалении какого-либо атрибута в таблице attribute\_group удаляются все строки в таблице attribute, ссылающиеся на удаляемый атрибут.

Между сущностями «attribute» и «attribute\_group» связь определена как «один-ко-многим», поскольку названию атрибута может соответствовать множество его значений, а значению атрибута – только одно название.

Связь между значениями атрибута и вариацией насоса задана при помощи таблицы «combination\_attribute» (рис. 1). Таблица имеет два поля id\_attribute и id\_combination с целочисленными данными, которые благодаря внешним ключам указывают на уникальные идентификаторы атрибута и комбинации, соответственно. Благодаря таблице «combination\_attribute» устанавливается связь «многие-ко-многим» между сущностями в таблицах «attribute» и «combination». Данная связь означает следующее: одному значению атрибута может соответствовать несколько насосов, и наоборот одной вариации насоса может соответствовать сразу несколько атрибутов.

Хранение свойств оборудования организовано при помощи двух таблиц – «feature» и «feature\_value» (рис. 1). Таблица feature состоит из двух полей – feature\_alias и id\_feature. Поле feature\_alias является текстовым и служит для хранения названий свойств оборудования. Целочисленный идентификатор свойства записывается в поле id\_feature. Оба поля являются уникальными.

Для хранения значений свойств предназначена таблица feature\_value. Значения записываются в текстовое поле feature\_value\_alias. Благодаря полю id\_feature и его внешнему ключу каждое значение свойства ссылается на его название в таблице «feature».

Между таблицами «feature» и «feature\_value» установлена связь «один-ко-многим», поскольку одному свойству может соответствовать множество значений.

Связи между комбинациями, свойствами комбинаций и их значениями задаются при помощи таблицы «feature\_combination» (рис. 1). Таблица feature\_combination состоит из трех полей, которые принимают целочисленные значения. Поле id\_combination при помощи внешнего ключа ссылается на уникальный идентификатор в таблице «combination», который в свою очередь связан с идентификатором оборудования. Поле id\_feature таблицы «feature\_combination» ссылается на уникальный идентификатор свойства в таблице «feature». Поле id\_feature\_value таблицы «feature\_combination» указывает на идентификатор значения свойства, которое хранится в таблице «feature\_value». Каждая строка рассматриваемой таблицы является уникальной. Таблица «feature\_combination» обеспечивает связь между сущностью вариации насоса и сущностью свойства насоса по типу «многие-ко-многим», поскольку одной вариации насоса могут соответствовать множество значений свойств и одно свойство может быть у множества вариаций насосов.

При создании программных продуктов, ориентированных на пользователей из различных стран важно в архитектуру базы данных заложить

поддержку многоязычности. Далее, на примере категорий насосов, будут рассмотрены различные варианты организации многоязычности при помощи базы данных.

На рис. 2 показан способ поддержки многоязычности, при котором для каждого языка в таблице категорий создается отдельное поле.

Недостатком такого подхода является необходимость каждый раз вносить изменения в структуру таблицы при добавлении нового языка. Данный способ может быть использован только в тех случаях, когда заранее известно количество поддерживаемых языков и, когда каждая сущность (в данном примере – название категории) должна существовать во всех языковых вариантах.

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

**Рис. 2.** Иллюстрация поддержки многоязычности категорий насосов путем создания полей для каждого языка

Иной способ поддержки многоязычности может быть реализован при помощи информации в сериализованном виде (рис. 3).

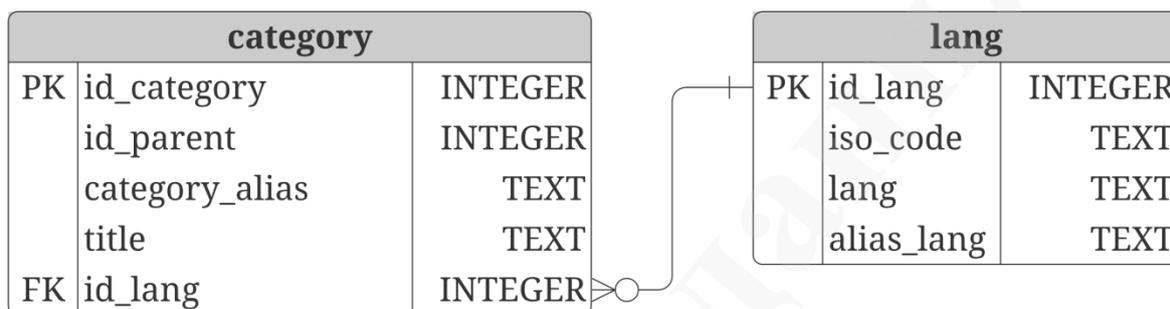
id_category	id_parent	category_alias	title
1	0	pumps	{"uk": "насоси", "ru": "насосы"}
2	1	centrifugal pumps	{"uk": "насоси відцентрові", "ru": "насосы центробежные"}

**Рис. 3.** Способ поддержки многоязычности при помощи сериализованных данных сложной структуры

Суть реализации состоит в том, что в каждое поле, требующее перевода, записывается информация в JSON, XML, binary или другом формате. При этом записанный объект может быть ассоциативным массивом или словарем с ключами в виде идентификаторов языков (например, ru, en, uk) и значениями в виде переводов на соответствующие языки. К недостаткам данной реализации следует отнести невозможность применения языка запросов для управления переводами. К примеру, для удаления всех переводов английского языка следует применить какой-либо язык программирования (Python, PHP) с необходимостью чтения и записи соответствующей строки таблицы. Применение языка запросов SQL в данном случае не представляется возможным.

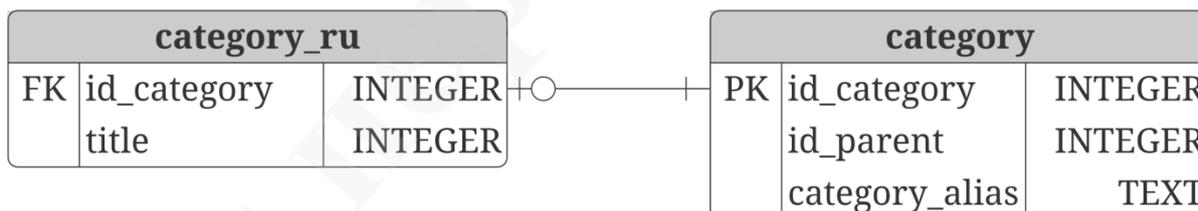
Возможен также вариант создания отдельных записей для каждого перевода в той же таблице, в которой содержится сущность, требующая локализации (рис. 4). В данном случае также создается таблица, в которой описываются доступные языки.

Недостатком является то, что для одной сущности создается сразу несколько объектов в таблице, что приводит к дублированию данных, не требующих перевода и, при наличии нескольких связей с другими таблицами, достаточно сильно усложняет бизнес-логику программного продукта.



**Рис. 4.** Поддержка многоязычности при помощи отдельной записи в таблице для каждого языка

Известен также пример реализации многоязычности, когда для каждого языка создают отдельную таблицу, имеющую поля, требующие перевода (рис. 5).



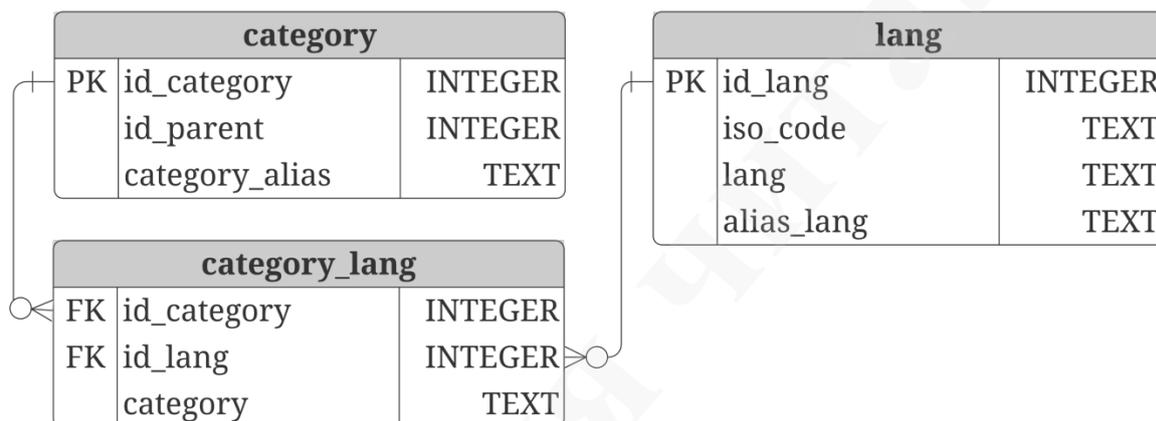
**Рис. 5.** Создание таблицы перевода для каждого языка

Между сущностями устанавливается связь «один-к-одному», поскольку одной строке перевода соответствует одна категория и одна запись категории может быть связана лишь с одной строкой в таблице переводов для соответствующего языка.

Недостатком такой структуры является необходимость создавать новую таблицу при добавлении нового языка. При такой реализации каждая сущность, требующая локализацию, будет иметь набор таблиц с переводами. Например, если требуется добавить два новых языка, то при наличии пяти сущностей, необходимо создать десять новых языковых таблиц, что представляется крайне неудобным.

Наиболее распространенным методом реализации многоязычности является создание двух таблиц для каждой сущности (рис. 6), где в основной таблице находятся поля, которые не зависят от локализации, а во второй – поля, для которых необходимы переводы. Также создается таблица со списком доступных языков. Поскольку заранее не известно, какие языки будут использоваться в проектируемой

базе данных, выберем данный способ поддержки многоязычности, как наиболее гибкий и легко масштабируемый. Общий принцип создания многоязычности опишем на примере категорий. Сущность «категория» хранится в основной таблице «category». Каждый язык также представляет собой отдельную сущность, которая записывается в таблицу «lang». В поле id\_lang с первичным ключом записывается идентификатор языка. Поле iso\_code содержит уникальное двухбуквенное обозначение языка в соответствии с ISO 639-1, например, ru. Поле lang предназначено для хранения названия языка в оригинале, например – русский. В поле alias\_lang записывается название языка на английском, например – russian.



**Рис.6.** Многоязычность с использованием таблиц локализации

В таблице «category\_lang» содержатся поля, которые требуют перевода. Благодаря полям id\_category и id\_lang каждая строка этой таблицы определяет, к какой категории и к какому языку относится перевод. Таблица «category\_lang» обеспечивает связь между сущностью категории и сущностью языка по типу «многие-ко-многим», поскольку одной сущности в таблице «category» может соответствовать несколько языков и одна сущность языка может использоваться для нескольких категорий.

По такому же принципу создаются языковые таблицы для других сущностей, например, комбинаций и свойств.

## 6. Результаты исследования

Рассмотрим структуру данных насоса НМ 2500-230 и две его вариации в зависимости от диаметра ротора рабочего колеса (рис. 7). Каждая вариация имеет свой уникальный идентификатор, который определен в поле id\_combination таблицы «combination». В этой же таблице задается связь между идентификатором марки насоса id\_equipment и идентификатором его вариации (комбинации). Для каждой вариации и языка в таблице «combination\_lang» указывается короткое short\_description и длинное description описание насоса. Каждая строка таблицы combination\_lang при помощи идентификатора id\_lang указывает на язык, для которого созданы соответствующие переводы. Доступные языки определены в таблице «lang». Ссылочная целостность данных обеспечивается благодаря внешним ключам, например, удаление насоса в

таблице «equipment» автоматически приводит к удалению всех его вариаций в таблице «combination» и переводов в таблице «combination\_lang». Аналогично, например, удаление русского языка из таблицы «lang» приводит к удалению всех русских переводов в таблице «combination\_lang».

equipment	
id_equipment	marking
1	nm_2500_230

combination	
id_combination	id_equipment
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	Насос магистральный НМ 2500-230, Дк=425 мм, n=2973 об/хв	НМ 2500-230
1	1	Насос магистральный НМ 2500-230, Дк=425 мм, n=2973 об/мин	НМ 2500-230
2	2	Насос магистральный НМ 2500-230, Дк=405 мм, n=2973 об/мин	НМ 2500-230
2	1	Насос магистральный НМ 2500-230, Дк=405 мм, n=2973 об/хв	НМ 2500-230

**Рис. 7.** Вариации насоса

Таблицы на рис. 8 показывают, от каких параметров (атрибутов) зависят вариации насоса.

Связи между атрибутами и вариациями определены в таблице «combination\_attribute». Например, вариации с идентификатором 1 соответствуют значения атрибутов с идентификаторами 1 и 3. Значения атрибутов определены в таблице «attribute». Так, идентификаторам значений атрибутов 1 и 3 соответствуют значения 0,5 и 425.

Связь таблицы «attribute» с таблицей attribute\_group обеспечивается благодаря внешнему ключу поля id\_attribute\_group. Например, значениям атрибутов с идентификатором 1 и 3 соответствуют атрибуты с идентификатором 1 и 2, где 1 и 2 соответствуют идентификаторам ротора и диаметр рабочего колеса насоса, соответственно.

combination_attribute	
id_attribute	id_combination
1	1
3	1
2	2
4	2

combination	
id_combination	id_equipment
1	1
2	1

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

**Рис. 8.** Атрибуты насоса

Названия и значения характеристик насосов хранятся в таблицах «feature» и «feature\_value», соответственно (рис. 9).

Таблица «feature\_combination» задает связь между вариацией насоса и его характеристиками. Например, у вариации насоса с идентификатором  $id\_combination=1$  имеется два свойства с идентификаторами 1 и 2. На значения свойств указывают идентификаторы в поле  $id\_feature\_value$ . Таким образом можно определить, что у рассматриваемого насоса есть два свойства – эмпирические коэффициенты  $H_0$  и  $b$  в формуле (1) со значениями  $H_0=246,7$  м ст. ж. и  $b=16,8 \cdot 10^{-6}$  ч<sup>2</sup>/м<sup>5</sup>.

Процесс получения конкретного оборудования из базы данных предложенной архитектуры может быть формализован благодаря применению SQL запросов. На рис. 9 отображены таблицы базы данных, которые хранят характеристики насоса НМ 2500-230 в двух вариациях:

- 1) ротор 0.5, диаметр рабочего колеса  $D_2=425$  мм;
- 2) ротор 0.7, диаметр рабочего колеса  $D_2=405$  мм.

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

**Рис. 9.** Характеристики насоса

В качестве примера сформируем запрос для получения характеристик насоса НМ 2500-230 с ротором 0.5 и диаметром рабочего колеса 425 мм (табл. 2).

Предложенный запрос имеет довольно сложную структуру, поскольку содержит вложенный запрос, заключенный между строками 4 и 14 (табл. 2). Задачей вложенного запроса является поиск уникального идентификатора вариации насоса `id_combination`, который в данном примере равен единице (табл. 3).

В общем случае поиск идентификатора `id_combination` осуществляется по двум критериям: по маркировке оборудования и значениям атрибутов. В рассматриваемом примере указанные критерии отражены в строках 11–13 запроса (табл. 2).

**Таблица 2**

SQL запрос для получения характеристик насоса НМ 2500-230 с ротором 0.5 и  $D_2=425$  мм

№ строки	Текст запроса
1	2
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

**Продолжение таблицы 2**

1	2
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

**Таблица 3**

Результат SQL запроса

Поля таблиц						
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

Функция агрегации COUNT после оператора SELECT (строка 5) подсчитывает количество строк, соответствующих каждой вариации насоса. После вызова функции COUNT указывается псевдоним combinations\_count, под которым будет доступен результат работы функции. Благодаря оператору GROUP BY (строка 14) строки с одинаковыми значениями id\_combination объединяются в одну. Поскольку по указанным в строках 11–13 критериям может соответствовать сразу несколько вариаций насоса, для получения необходимой следует использовать оператор HAVING со значением поля combinations\_count равным двум (по числу атрибутов). В результате работы вложенного запроса будет сформирована таблица, доступная по псевдониму comb и состоящая из одной строки. После левого соединения (строка 16) таблицы feature\_combination, связывающая характеристики насоса с его вариациями, образуется таблица, состоящая из двух строк по числу характеристик (H0, b). На этом этапе получены идентификаторы свойств и их

значений. Левым соединением привязываем строки с названиями свойств и их значениями (строки 17 и 18). В итоге получаем результирующую таблицу, содержащую характеристики насоса (табл. 2), где одной строке соответствует одна характеристика.

Структура разработанной базы данных выложена на веб-сервис GitHub [27] и доступна для внесения изменений.

## 7. SWOT-анализ результатов исследования

*Strength.* По сравнению с ручным вводом входных данных в программный продукт их хранение в электронном виде позволяет значительно сократить трудоемкость расчета характеристик нефтяных насосов и режимов работы нефтепроводов. Преимущество реляционных баз данных по сравнению с электронными таблицами в следующем:

- бесплатные;
- позволяют хранить значительные объёмы данных, обеспечивая при этом высокую производительность;
- легко могут использоваться в связке с популярными языками программирования (Python, PHP, Java и др.) и пакетами прикладных математических программ (Matlab, Mathematica, Scilab);
- расширяют возможности для масштабирования проектов, например, может быть реализована поддержка многоязычности со структурой таблиц, которая наилучшим образом отвечает требованиям проекта;
- используют удобный язык запросов SQL.

*Weaknesses.* В реляционных базах данных для добавления, изменения и удаления данных используется язык запросов SQL напрямую или посредством визуального редактора (phpMyadmin, dbviewer, dbeaver). Это в обоих случаях является недостаточно удобным. Для решения этой проблемы данные импортируют из файлов в формате json, xml, csv или xls, что требует дополнительных затрат времени на разработку модулей импорта.

*Opportunities.* Результаты исследования могут быть использованы при проектировании баз данных нефтегазового оборудования различного типа, например, теплообменников, пылеуловителей, регуляторов давления газа газораспределительных станций и пунктов, центробежных нагнетателей, газотурбинных двигателей и нефтяных резервуаров. Благодаря продуманной архитектуре базы данных возможно создание как узкоспециализированных программ, рассчитанных на решение одной задачи, так и больших программных комплексов для расчета сложных технологических процессов.

*Threats.* Работа с реляционными базами данных предъявляет более высокие требования к квалификации разработчика программы, поскольку требуются знания по проектированию архитектур баз данных, составлению таблиц и связей между ними, расстановке индексов и определению ключей.

## 8. Выводы

1. Анализ задач трубопроводного транспорта нефти и нефтепродуктов и современных программ по расчету режимов работы нефтепроводов показал

актуальность создания баз данных насосного оборудования. Показано, что ввиду сложной структуры данных насосов, предпочтительнее использовать реляционные базы данных, чем электронные таблицы Excel. При проектировании физической модели базы данных выделены родительские и дочерние сущности таким образом, чтобы можно было отразить всю информацию о насосном агрегате: конструктивные особенности, область применения, характеристики насоса в зависимости от его вариаций по сменным роторам. В процессе моделирования определены правила целостности данных, и установлены связи типа «один-ко-многим» и «многие-ко-многим» между сущностями физической модели. Разработанная архитектура базы данных обеспечивает модифицируемость, масштабируемость, модульность и эффективное использование в программах для расчета режимов работы нефтепроводов.

2. Выполнен обзор различных способов организации многоязычности в базе данных. Для достижения цели работы наиболее предпочтительным представляется использование структуры, в которой каждая сущность хранится в двух таблицах. В одной таблице хранятся данные, которые не зависят от переводов, а в другой – данные с переводами.

## Литература

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., Hryhorskyi, 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., Mezerya, A. Iu., Kniazieva, V. N. (2015). Analiz rezervov energoberezhennia pri upravlenii nasosnymi agregatami nefteperekachivaiuschikh stantsii Ukrainy. *Integrovani tekhnologii ta energoberezhennia*, 4, 3–14. Available at: [http://repository.kpi.kharkov.ua/bitstream/KhPI-Press/29242/1/ITE\\_2015\\_4\\_Kanyuk\\_Analiz\\_rezervov.pdf](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). *Truboprovidnyi 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 rozrakhunky 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 prisadok na gidravlicheskuu effektivnost nefteproduktprovodov*. Ufa, 162. Available at: <http://earthpapers.net/otsenka-vliyaniya-protivoturbulentnyh-prisadok-na-gidravlicheskyu-effektivnost-nefteproduktprovodov>
16. Al-Dandal, R. S. (2016). *Ratsionalni rezhymy ekspluatatsii mahistralnykh naftoproduktprovodiv z vykorystanniam protyturbulentnykh 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, razrobotannykh 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](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 nefteproduktprovodnykh 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](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](https://john.cs.olemiss.edu/~ychen/publications/conference/vicknair_acmse10.pdf)

26. Kirillov, V. V., Gromov, G. Iu. (2009). *Vvedenie v reliatsionnye bazy dannykh*. Saint Petersburg: BKHV-Peterburg, 464.

27. *Pumps database – GitHub repository*. Available at: [https://github.com/Buzovskiy/equipment\\_database](https://github.com/Buzovskiy/equipment_database)