

DOI: 10.12731/2227-930X-2025-15-2-360

EDN: XVPFOM



УДК 004.65

Научная статья |

Системный анализ, управление и обработка информации, статистика

## РАЗРАБОТКА БАЗЫ ДАННЫХ ДЛЯ ЭФФЕКТИВНОГО УПРАВЛЕНИЯ АВТОЗАПРАВОЧНЫМИ СТАНЦИЯМИ

*Г.А. Гареева, А.Г. Файзуллина, З.Ш. Аглямова,  
Ю.Н. Бурханова, Р.В. Спесваков*

### *Аннотация*

**Обоснование.** Современные автозаправочные станции сталкиваются с рядом проблем, связанных с управлением данными, учетом запасов и обслуживанием клиентов. Традиционные методы учета и управления часто оказываются неэффективными, что приводит к увеличению временных затрат на обработку информации и повышению вероятности ошибок. Отсутствие централизованной системы для хранения и обработки данных затрудняет анализ продаж и управление запасами, что негативно сказывается на общей эффективности работы станции.

Создание базы данных для автозаправочной станции решает эти проблемы, обеспечивая автоматизацию процессов учета, управления запасами и обслуживания клиентов. База данных позволяет централизованно хранить информацию о продажах, клиентах, топливе и финансовых операциях, что значительно упрощает доступ к данным и их обработку.

**Цель** – разработать инструмент, который обеспечит автоматизацию учета и управления на автозаправочной станции, включая хранение информации о клиентах, топливе и транзакциях, а также поддержку аналитики для принятия обоснованных управленческих решений. Это позволит повысить эффективность работы станции, улучшить качество обслуживания клиентов и снизить вероятность ошибок в учете.

**Метод и методология.** Для разработки базы данных автозаправочной станции использовался язык программирования Java для создания интерфейса и логики приложения, а также PostgreSQL для хранения и управления данными. Подход основывался на использовании реляционной модели данных, что обеспечивает надежность, целостность и масштабируемость системы.

**Результаты.** База данных поддерживает хранение информации о клиентах, топливе, продажах и финансовых операциях. Данные централизованно управляются в PostgreSQL, что позволяет легко выполнять операции создания, редактирования и удаления записей. Система также включает функции для анализа данных, что способствует принятию обоснованных управленческих решений. Архитектура базы данных обеспечивает модульность, позволяя добавлять новые функции без значительных изменений в существующем коде.

**Область применения.** Разработанная база данных может быть использована в управлении автозаправочными станциями, для автоматизации процессов учета и обслуживания клиентов, а также в аналитике для повышения эффективности бизнес-процессов.

**Выводы.** Созданная база данных значительно упрощает управление данными на автозаправочной станции, снижает вероятность ошибок и повышает общую эффективность работы. Автоматизация процессов учета и обслуживания клиентов позволяет сэкономить время и ресурсы.

**Ключевые слова:** автозаправочная станция; база данных; PostgreSQL; DBeaver; автоматизация; управление данными; аналитика; эффективность; интеграция; реляционная модель; учет клиентов; управление запасами

**Для цитирования.** Гареева, Г. А., Файзуллина, А. Г., Аглямова, З. Ш., Бурханова, Ю. Н., & Спееваков, Р. В. (2025). Разработка базы данных для эффективного управления автозаправочными станциями. *International Journal of Advanced Studies*, 15(2), 64–85. <https://doi.org/10.12731/2227-930X-2025-15-2-360>

Original article |

System Analysis, Management and Information Processing, Statistics

## DEVELOPMENT OF A DATABASE FOR EFFICIENT MANAGEMENT OF GAS STATIONS

*G.A. Gareeva, A.G. Faizullina, Z.Sh. Aglyamova,  
Yu.N. Burkhanova, R.V. Spevakov*

### *Abstract*

**Background.** Modern gasoline stations face a number of challenges related to data management, inventory accounting, and customer service. Traditional accounting and management methods are often inefficient, resulting in increased time spent on processing information and an increased likelihood of errors. The lack of a centralized system for storing and processing data makes it difficult to analyze sales and manage inventory, which negatively affects the overall efficiency of the station.

Creating a database for a gas station solves these problems by providing automation of accounting, inventory management and customer service processes. The database allows centralized storage of information on sales, customers, fuel and financial transactions, which greatly simplifies data access and processing.

**Purpose.** The objective is to develop a tool that will provide automation of accounting and management at a filling station, including storage of information on customers, fuel and transactions, as well as support for analytics to make informed management decisions. This will increase the efficiency of the station, improve customer service and reduce the likelihood of accounting errors.

**Method and methodology.** To develop the gas station database, the Java programming language was used to create the interface and application logic, and PostgreSQL was used to store and manage the data. The approach was based on the use of a relational data model, which ensures the reliability, integrity and scalability of the system.

**Results.** The database supports storage of customer, fuel, sales and financial transaction information. Data is centrally managed in PostgreSQL, allowing easy operations for creating, editing and deleting records. The system also includes functions to analyze the data to facilitate informed management decisions. The database architecture provides modularity, allowing new functions to be added without significant changes to existing code.

**Scope of the results.** The developed database can be used in the management of gas stations, for automation of accounting and customer service processes, as well as in analytics to improve the efficiency of business processes.

**Conclusions.** The created database significantly simplifies data management at the gasoline station, reduces the probability of errors and increases the overall efficiency of work. Automation of accounting and customer service processes saves time and resources.

**Keywords:** gas station; database; PostgreSQL; DBeaver; automation; data management; analytics; efficiency; integration; relational model; customer accounting; inventory management

**For citation.** Gareeva, G. A., Faizullina, A. G., Aglyamova, Z. Sh., Burkhanova, Yu. N., & Spevakov, R. V. (2025). Development of a database for efficient management of gas stations. *International Journal of Advanced Studies*, 15(2), 64–85. <https://doi.org/10.12731/2227-930X-2025-15-2-360>

## Введение

Проблема эффективного управления автозаправочными станциями становится все более актуальной в современном мире, где автомобильный транспорт играет ключевую роль в экономике и повседневной жизни [1].

Автозаправочные станции не только обеспечивают заправку автомобилей топливом, но и предлагают широкий спектр дополнительных услуг, что требует системного подхода к организации данных [4].

Существующие методы управления часто не позволяют оптимизировать процессы учета и управления, что приводит к снижению качества обслуживания клиентов. Решение данной проблемы требует разработки специализированной базы данных, которая позволит автоматизировать процессы управления, повысить уровень обслуживания и оптимизировать внутренние процессы на автозаправочной станции.

### **Цель и задачи**

Целью работы является создание модели базы данных для автозаправочной станции, которая позволит автоматизировать процессы учета и управления, повысить уровень обслуживания клиентов и оптимизировать внутренние процессы. Для достижения цели решены следующие задачи:

- Проведен анализ предметной области и определены ключевые сущности и их взаимосвязи;
- Разработаны логическая и физическая структуры базы данных;
- Реализована интеграция с СУБД PostgreSQL для централизованного хранения данных.

### **Материалы и методы разработки**

Процесс разработки включал:

- Выбор инструментов. Использование средств СУБД DBEAVER и PostgreSQL для реализации базы данных;
- Архитектура программы. Построение ER-модели предметной области и преобразование ее в реляционную схему;
- Интеграция с базой данных. Использование SQL для взаимодействия с PostgreSQL и выполнения запросов к базе данных.

### **Реализация программы**

Реализация программы включала в себя множество этапов, направленных на создание специализированного инструмента для автоматизации процессов учета и управления на автозаправочной

станции. Каждый из этапов был тщательно проработан с учетом требований к эффективности, удобству использования и интеграции с существующими системами [5].

Первым шагом был проведен анализ ключевых аспектов функционирования автозаправочной станции, включая управление запасами топлива, учет клиентов и их бонусных карт, а также управление финансовыми операциями и персоналом.

На основе анализа были выделены основные сущности и их взаимосвязи, что позволило определить структуру базы данных, которая будет храниться в PostgreSQL [11]. На рисунке 1 представлена сущность-связь в нотации Чена.

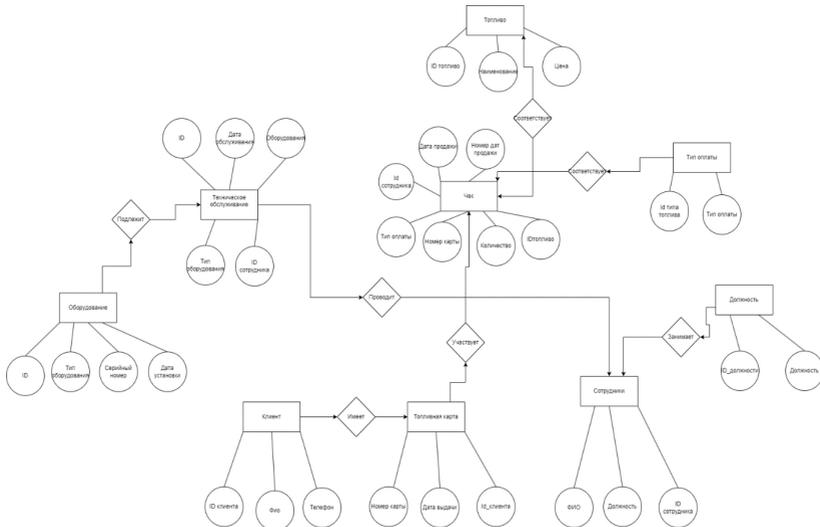
Эти сущности включают управление запасами топлива, учет клиентов и их бонусных карт, а также управление финансовыми операциями и персоналом.

Каждая из сущностей была детализирована для определения необходимых атрибутов и взаимосвязей [6].

Например, управление запасами топлива включает в себя такие атрибуты, как название топлива, уникальный номер позиции, поставщик и количество. Учет клиентов включает информацию о клиентах, их контактные данные и бонусные карты. Управление финансовыми операциями охватывает учет транзакций, контроль цен и торговых наценок.

Результаты анализа позволили определить структуру базы данных, которая будет хранить информацию о каждой из сущностей. Для каждой сущности была предусмотрена возможность интеграции с другими сущностями, что обеспечит целостность и согласованность данных. Это позволит пользователям легко взаимодействовать с системой и получать необходимую информацию для принятия решений в реальном времени [3].

Таким образом, проведенный анализ стал основой для разработки модели базы данных, которая будет эффективно поддерживать автоматизацию процессов на автозаправочной станции и обеспечивать удобное использование в реальных условиях.



**Рис.1.** Модель сущность-связь «Автозаправочной станции» в нотации Чена

Создание таблиц в реляционной базе данных включает в себя определение структуры таблицы, которая включает в себя:

1. Имя таблицы: уникальный идентификатор, который используется для обращения к таблице [7].
2. Столбцы: каждый столбец представляет собой атрибут сущности и имеет определенный тип данных (например, целое число, строка, дата и т.д.). При создании таблицы необходимо определить имя каждого столбца, его тип данных и дополнительные ограничения (например, уникальность, обязательность заполнения) [8].
3. Первичный ключ: это уникальный идентификатор для каждой записи в таблице, который обеспечивает целостность данных и позволяет однозначно идентифицировать каждую строку [9].

Основной сущностью системы автозаправочной станции является топливо. Программный код таблицы «Топливо» представлен на рисунке 2.

```
CREATE TABLE Топливо (  
    ТопливоID SERIAL PRIMARY KEY, -- Уникальный идентификатор для каждого типа топлива  
    Название VARCHAR(100) NOT NULL, -- Название топлива (например, "Бензин", "Дизель")  
    Тип VARCHAR(50), -- Тип топлива (например, "Сжиженный", "Газ", "Электрический")  
    Цена DECIMAL(10, 2) NOT NULL, -- Цена за единицу топлива (например, за литр)  
    Количество DECIMAL(10, 2) NOT NULL, -- Доступное количество топлива на складе  
    ДатаПоставки TIMESTAMP DEFAULT CURRENT_TIMESTAMP -- Дата последней поставки топлива  
);
```

Рис. 2. Программная реализация таблицы «Топливо»

Система включает таблицу должностей, где каждая должность имеет уникальный идентификатор и наименование, например, «Менеджер» или «Оператор». Система включает таблицу должностей, где каждая должность имеет уникальный идентификатор и наименование, например, «Менеджер» или «Оператор». На рисунке 3 представлен код таблицы «Должность».

```
CREATE TABLE Должность (  
    ID_Должности SERIAL PRIMARY KEY, -- Уникальный идентификатор должности  
    Наименование VARCHAR(100) NOT NULL -- Наименование должности (например, "Менеджер", "Разработчик")  
);
```

Рис. 3. Программная реализация таблицы «Должность»

Сотрудники системы имеют уникальные идентификаторы, фамилии, имена и отчества, а также ссылаются на свои должности через внешний ключ. Данная таблица представлена на рисунке 4.

```
CREATE TABLE Сотрудники (  
    ID_Сотрудника SERIAL PRIMARY KEY, -- Уникальный идентификатор сотрудника  
    ФИО VARCHAR(100) NOT NULL, -- Фамилия Имя Отчество сотрудника  
    ID_Должности INT NOT NULL, -- Идентификатор должности (внешний ключ)  
    FOREIGN KEY (ID_Должности) REFERENCES Должность(ID_Должности) -- Связь с таблицей Должность  
);
```

Рис. 4. Программная реализация таблицы «Сотрудники»

Для учета оборудования в системе предусмотрена таблица оборудования, где каждый тип оборудования имеет уникальный идентификатор, название, серийный номер и дату установки. Записи о техническом обслуживании оборудования фиксируют дату обслуживания, название оборудования и ссылаются на конкретное обо-

рудование и сотрудника, проводившего обслуживание. Программная реализация таблицы «Оборудование» показана на рисунке 5.

```
CREATE TABLE Оборудование (  
    ID_типа_Оборудования SERIAL PRIMARY KEY,  
    Тип_оборудования VARCHAR(255) NOT NULL,  
    Серийный_номер VARCHAR(255) NOT NULL,  
    Дата_установки DATE NOT NULL  
);
```

**Рис. 5.** Программная реализация таблицы «Оборудование»

Система также включает таблицу типов оплаты, где каждый тип имеет уникальный идентификатор и наименование. При продаже топлива создается запись в таблице чеков, которая содержит уникальный номер чека, дату продажи, информацию о проданном топливе, количество, номер топливной карты, тип оплаты и идентификатор сотрудника, который осуществил продажу. На рисунке 6 представлен программный код таблицы «Техническое обслуживание».

Записи о техническом обслуживании содержат уникальный идентификатор записи (ID\_записи), дату обслуживания (Дата\_обслуживания), описание услуги (Описание) и стоимость обслуживания (Стоимость). Эта таблица фиксирует все проведенные работы по обслуживанию оборудования.

```
CREATE TABLE Техническое_обслуживание (  
    ID_записи SERIAL PRIMARY KEY,  
    Дата_обслуживания DATE NOT NULL,  
    Оборудование VARCHAR(255) NOT NULL,  
    ID_Типа_Оборудования INT REFERENCES Оборудование(ID_типа_оборудования),  
    ID_сотрудника INT REFERENCES Сотрудники(ID_Сотрудника)  
);
```

**Рис. 6.** Программный код таблицы «Техническое обслуживание»

На рисунке 7 представлен программный код таблицы «Тип оплаты». Каждый тип оплаты имеет уникальный идентификатор (ID\_типа\_оплаты) и наименование типа оплаты (Тип\_оплаты). Эта

таблица используется для учета различных способов оплаты, доступных клиентам.

```
CREATE TABLE Тип_оплаты (  
    ID_типа_оплаты SERIAL PRIMARY KEY,  
    Тип_оплаты VARCHAR(255) NOT NULL  
);
```

**Рис. 7.** Программный код таблицы «Тип оплаты»

На рисунке 8 представлен программный код таблицы «Чек». Каждый чек содержит уникальный номер чека (Номер\_Чека), дату продажи (Дата\_продажи), информацию о проданном топливе (ID\_топлива), количество топлива, проданного клиенту, номер карты, тип оплаты и идентификатор сотрудника, который осуществил продажу. Эта таблица фиксирует все продажи топлива клиентам.

```
CREATE TABLE Чек (  
    Номер_Чека SERIAL PRIMARY KEY,  
    Дата_продажи DATE NOT NULL,  
    ID_топлива INT REFERENCES Топливо(ID_Топлива), -- Исправлено имя столбца на ID_Торлива  
    Количество INT NOT NULL,  
    Номер_карты INT REFERENCES Топливная_карта(Номер_карты),  
    ID_типа_оплаты INT REFERENCES Тип_оплаты(ID_типа_оплаты),  
    ID_сотрудника INT REFERENCES Сотрудники(ID_Сотрудника)  
);
```

**Рис. 8.** Программный код таблицы «Чек»

Таким образом, система обеспечивает полный учет клиентов, топлива, сотрудников и операций по продаже топлива. Все сущности связаны между собой через внешние ключи, что позволяет легко отслеживать взаимодействия между ними и получать необходимую отчетность. На рисунке 9 представлена физическая модель базы данных.

Далее представлены результаты нескольких запросов:

1. Данный SQL-запрос выполняет выборку данных из нескольких связанных таблиц в реляционной базе данных. Он извлекает информацию о продажах топлива, включая детали о чеке, типе

топлива и клиенте. На рисунке 10 показан запрос, а на рисунке 11 его результат.



Рис. 9. Физическая модель базы данных «Автозаправочной станции»

```

SELECT
    Чек.Номер_Чека,
    Чек.Дата_продажи,
    Топливо.Наименование AS Топливо,
    Чек.Количество,
    Клиент.ФИО AS Клиент
FROM
    Чек
INNER JOIN
    Топливо ON Чек.ID_топлива = Топливо.ID_Топлива
INNER JOIN
    Топливная_карта ON Чек.Номер_карты = Топливная_карта.Номер_карты
INNER JOIN
    Клиент ON Топливная_карта.ID_Клиента = Клиент.ID_Клиента;
    
```

Рис. 10. Код запроса 1

123 Номер_Че	Дата_прод	А2 Топливо	123 Количеств	А: Клиент
1	2024-02-10	АИ-92	20	Иванов Иван Иванович
2	2024-02-11	АИ-95	30	Петров Петр Петрович
3	2024-02-12	АИ-100	15	Сидоров Сидор Сидорович

Рис. 11. Результат запроса 1

1. Запрос 2 выбирает всех сотрудников (таблица Сотрудники) и информацию о техническом обслуживании (таблица Техническое обслуживание), включая тех, у кого нет записей об обслуживании. На рисунке 12 представлен запрос 2.

```

SELECT
    Сотрудники.ID_Сотрудника,
    Сотрудники.ФИО,
    Техническое_обслуживание.Дата_обслуживания,
    Техническое_обслуживание.Оборудование
FROM
    Сотрудники
LEFT JOIN
    Техническое_обслуживание ON Сотрудники.ID_Сотрудника = Техническое_обслуживание.ID_сотрудника;

```

Рис. 12. Код запроса 2

```

SELECT
    Оборудование.id_типа_оборудования,
    Оборудование.Тип_оборудования,
    Оборудование.Серийный_номер,
    Техническое_обслуживание.Дата_обслуживания,
    CASE
        WHEN Техническое_обслуживание.Дата_обслуживания >= CURRENT_DATE - INTERVAL '30 days' THEN 'Недавнее обслуживание'
        ELSE 'Старое обслуживание'
    END AS Статус_обслуживания
FROM
    Оборудование
RIGHT JOIN
    Техническое_обслуживание ON Оборудование.id_типа_оборудования = Техническое_обслуживание.id_Типа_Оборудования;

```

Рис. 13. Код запроса 3

2. Предположим, что необходимо добавить в запрос информацию о статусе обслуживания оборудования (рис. 13). Если дата обслуживания находится в пределах последних 30 дней, то можно пометить его как «Недавнее обслуживание», иначе - «Старое

обслуживание». Используется CASE в PostgreSQL - это условное выражение, которое позволяет создавать условные запросы [11].

3. Запрос 4 создается CTE для подсчета количества чеков, связанных с каждым клиентом, а затем извлекаются данные из этого CTE (рис. 14).

```

WITH Чеки_по_клиентам AS (
    SELECT
        Клиент.ID_Клиента,
        Клиент.ФИО,
        COUNT(Чек.Номер_Чека) AS Количество_чеков
    FROM
        Клиент
    LEFT JOIN
        Топливная_карта ON Клиент.ID_Клиента = Топливная_карта.ID_Клиента
    LEFT JOIN
        Чек ON Топливная_карта.Номер_карты = Чек.Номер_карты
    GROUP BY
        Клиент.ID_Клиента, Клиент.ФИО
)

SELECT
    ФИО,
    Количество_чеков
FROM
    Чеки_по_клиентам
ORDER BY
    Количество_чеков DESC;

```

ент 1 ×

Н Чеки\_по\_клиентам AS ( SELECT К्ल... Введите SQL выражение чтобы отфильтровать результат

ФИО	Количество_чеков
Петров Петр Петрович	1
Сидоров Сидор Сидорович	1
Иванов Иван Иванович	1

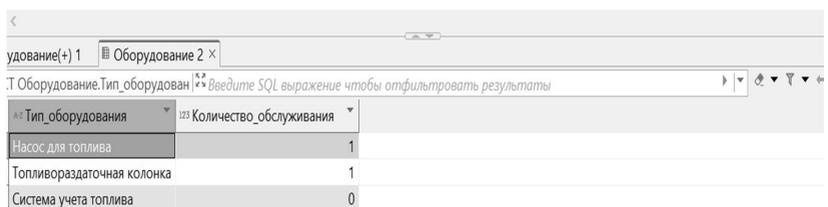
Рис. 14. Код запроса 4 и его результат

4. Этот запрос подсчитывает количество записей о техническом обслуживании для каждого типа оборудования и сортирует результаты по количеству в порядке убывания. На рисунке 15 представлен данный запрос.

```

SELECT
    Оборудование.Тип_оборудования,
    COUNT(Техническое_обслуживание.ID_записи) AS Количество_обслуживания
FROM
    Оборудование
LEFT JOIN
    Техническое_обслуживание ON Оборудование.id_типа_оборудования = Техническое_обслуживание."id_Типа_Оборудования"
GROUP BY
    Оборудование.Тип_оборудования
ORDER BY
    Количество_обслуживания DESC;

```



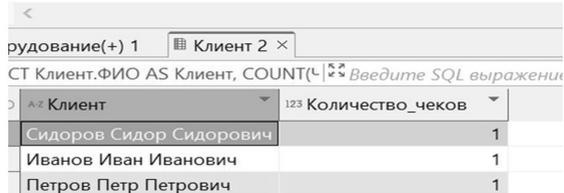
Тип_оборудования	Количество_обслуживания
Насос для топлива	1
Топливораздаточная колонка	1
Система учета топлива	0

Рис. 15. Запрос 5 и его результат

```

SELECT
    Клиент.ФИО AS Клиент,
    COUNT(Чек.Номер_Чека) AS Количество_чеков
FROM
    Клиент
LEFT JOIN
    Чек ON Клиент.ID_Клиента = Чек.Номер_карты
GROUP BY
    Клиент.ФИО
ORDER BY
    Количество_чеков DESC;

```



Клиент	Количество_чеков
Сидоров Сидор Сидорович	1
Иванов Иван Иванович	1
Петров Петр Петрович	1

Рис. 16. Запрос 6 и его результат

Чтобы подсчитать количество чеков для каждого клиента и отсортировать результаты по количеству чеков в порядке убывания, построен запрос 6 (рис. 16).

### **Область применения**

База данных автозаправочной станции может использоваться в следующих целях:

- в управлении запасами - для учета и контроля наличия топлива и других ресурсов на автозаправочной станции;
- в обслуживании клиентов - для хранения информации о клиентах, их бонусных картах и истории покупок, что позволяет улучшить качество обслуживания;
- в финансовом учете - для автоматизации процессов учета продаж, анализа доходов и расходов, а также формирования отчетности;
- в аналитике - для проведения анализа продаж, выявления тенденций и оптимизации бизнес-процессов на основе собранных данных;
- в интеграции с другими системами - для обмена данными с системами управления, бухгалтерскими программами и другими приложениями, что обеспечивает комплексный подход к управлению автозаправочной станцией.

### **Заключение**

В ходе разработки базы данных для автозаправочной станции была создана эффективная система управления, способствующая автоматизации процессов учета и обслуживания клиентов. Реализованные функции позволяют не только контролировать запасы топлива и управлять финансовыми операциями, но и обеспечивать высокий уровень сервиса для клиентов.

Использование реляционной модели данных и интеграция с современными инструментами, такими как PostgreSQL и DBeaver, обеспечили надежность и масштабируемость системы [10]. Вне-

дрение базы данных позволяет значительно сократить время на обработку информации, минимизировать ошибки и повысить общую эффективность работы автозаправочной станции.

База данных также открывает возможности для дальнейшего анализа данных, что может способствовать принятию более обоснованных управленческих решений и оптимизации бизнес-процессов. В будущем планируется расширение функционала системы, включая интеграцию с мобильными приложениями и системами лояльности, что позволит улучшить взаимодействие с клиентами и повысить конкурентоспособность автозаправочной станции на рынке.

Таким образом, разработанная база данных является важным инструментом для успешного функционирования автозаправочной станции, обеспечивая надежное хранение и обработку данных, а также способствуя улучшению качества обслуживания клиентов и повышению эффективности бизнес-процессов.

### **Список литературы**

1. Остроух, А. В., & Суркова, Н. Е. (2019). *Проектирование информационных систем*. Монография. Санкт-Петербург: Лань. 164 с.
2. Громов, А. И., Фляйшман, А., & Шмидт, В. (2021). *Управление бизнес-процессами: современные методы*. Монография (под ред. А. И. Громова). Москва: Издательство Юрайт. 367 с.
3. Бондаренко, И. С. (2019). *Базы данных: создание баз данных в среде SQL Server*. Лабораторный практикум. Москва: Изд. Дом НИТУ «МИСиС». 39 с.
4. Линец, Г. И., & Братченко, Н. Ю. (2021). *Базы данных*. Учебник. Ставрополь: Изд-во СКФУ. 170 с.
5. Файли, К. (2023). *SQL. Руководство для использования с любыми SQL СУБД*. Учебное пособие (2-е изд.; пер. с англ. А. В. Хаванова). Москва: ДМК Пресс. 454 с.
6. Ёсу, М. (2021). *Принципы организации распределенных баз данных*. Учебник (пер. с англ. А. А. Слинкина). Москва: ДМК Пресс. 672 с.

7. Харрингтон, Д. (2023). *Проектирование объектно-ориентированных баз данных*. Практическое руководство (2-е изд.; пер. с англ. А. А. Слинкина). Москва: ДМК Пресс. 273 с.
8. Льюис, Д. (2023). *Ядро Oracle. Внутреннее устройство для администраторов и разработчиков баз данных*. Практическое руководство (2-е изд.; пер. с англ. А. Н. Киселева). Москва: ДМК Пресс. 373 с.
9. Мартишин, С. А., Симонов, В. Л., & Храпченко, М. В. (2024). *Базы данных. Практическое применение СУБД SQL и NoSQL типа для проектирования информационных систем*. Учебное пособие. Москва: ФОРУМ: ИНФРА-М. 368 с.
10. Мартишин, С. А., Симонов, В. Л., & Храпченко, М. В. (2022). *Базы данных: проектирование и разработка информационных систем с использованием СУБД MySQL и языка Go*. Учебное пособие. Москва: ИНФРА-М. 325 с. <https://doi.org/10.12737/1830834> EDN: <https://elibrary.ru/PNOJTV>
11. Митин, А. И. (2020). *Работа с базами данных Microsoft SQL Server: сценарии практических занятий*. Учебно-методическое пособие. Москва: Директ-Медиа. 143 с. EDN: <https://elibrary.ru/SEKKTY>
12. Amin Al Ka'bi. (2021). Management of energy consumption using programmable logic controllers (PLCs). *Proceedings on Engineering Sciences*, 3(3), 267–272. <https://doi.org/10.24874/pes03.03.003> EDN: <https://elibrary.ru/MTGQZK>
13. Walters III, E. G., & Bryla, E. J. (2016). The impact of PLC program architecture on production line efficiency: Case study of a control system rewrite. *Machines*, 4(2), 13. <https://doi.org/10.3390/machines4020013>
14. Martin A. Sehr et al. (2024). Programmable Logic Controllers in the context of Industry 4.0. *IEEE Journals & Magazine*. Получено с <https://ieeexplore.ieee.org/document/9134804>
15. Tiago Cruz et al. (2024). Virtualizing Programmable Logic Controllers: Toward a convergent approach. *IEEE Journals & Magazine*. Получено с <https://ieeexplore.ieee.org/document/7564414>

### References

1. Ostroukh, A. V., & Surkova, N. E. (2019). *Design of Information Systems*. Monograph. St. Petersburg: Lan'. 164 p.
2. Gromov, A. I., Fleishman, A., & Shmidt, V. (2021). *Business Process Management: Modern Methods* (Edited by A. I. Gromov). Moscow: Yurait Publishing House. 367 p.
3. Bondarenko, I. S. (2019). *Databases: Creating Databases in SQL Server Environment*. Lab manual. Moscow: MISiS Publishing House. 39 p.
4. Linec, G. I., & Bratchenko, N. Y. (2021). *Databases*. Textbook. Stavropol: SFU Publishing House. 170 p.
5. Fiayli, K. (2023). *SQL: User Guide for Any SQL DBMS* (2nd ed.; translated by A. V. Khavanov). Moscow: DMK Press. 454 p.
6. Yo, M. (2021). *Principles of Distributed Database Organization* (translated by A. A. Slinkin). Moscow: DMK Press. 672 p.
7. Harrington, D. (2023). *Object-Oriented Database Design: Practical Guide* (2nd ed.; translated by A. A. Slinkin). Moscow: DMK Press. 273 p.
8. Lewis, D. (2023). *Oracle Kernel: Internal Workings for DBAs and Developers* (2nd ed.; translated by A. N. Kiselev). Moscow: DMK Press. 373 p.
9. Martishin, S. A., Simonov, V. L., & Khrapchenko, M. V. (2024). *Databases: Practical Application of SQL and NoSQL DBMS for Information Systems Design*. Moscow: Forum INFRA-M. 368 p.
10. Martishin, S. A., Simonov, V. L., & Khrapchenko, M. V. (2022). *Databases: Design and Development of Information Systems Using MySQL DBMS and Go Programming Language*. Moscow: INFRA-M. 325 p. <https://doi.org/10.12737/1830834> EDN: <https://elibrary.ru/PNOJTV>
11. Mitin, A. I. (2020). *Microsoft SQL Server Databases: Practical Session Scenarios*. Tutorial. Moscow: Direct-Media. 143 p. EDN: <https://elibrary.ru/SEKKTY>
12. Amin Al Ka'bi. (2021). Management of energy consumption using programmable logic controllers (PLCs). *Proceedings on Engineering Sciences*, 3(3), 267–272. <https://doi.org/10.24874/pes03.03.003> EDN: <https://elibrary.ru/MTGQZK>

13. Walters III, E. G., & Bryla, E. J. (2016). The impact of PLC program architecture on production line efficiency: Case study of a control system rewrite. *Machines*, 4(2), 13. <https://doi.org/10.3390/machines4020013>
14. Martin A. Sehr et al. (2024). Programmable Logic Controllers in the context of Industry 4.0. *IEEE Journals & Magazine*. Retrieved from <https://ieeexplore.ieee.org/document/9134804>
15. Tiago Cruz et al. (2024). Virtualizing Programmable Logic Controllers: Toward a convergent approach. *IEEE Journals & Magazine*. Retrieved from <https://ieeexplore.ieee.org/document/7564414>

### **ВКЛАД АВТОРОВ**

**Гареева Г.А.:** разработка базы данных для автоматизации управления данными, обработка результатов исследований.

**Файзуллина А.Г.:** формулирование основных направлений исследования, разработка теоретических предпосылок, формирование общих выводов.

**Аглямова З.Ш.:** проведение сбора данных, подготовка начального варианта статьи.

**Бурханова Ю.Н.:** научное редактирование текста статьи и окончательное утверждение версии для публикации.

**Спеваков Р.В.:** анализ и интерпретация полученных данных, литературный анализ.

### **AUTHORS CONTRIBUTION**

**Gulnara A. Gareeva:** development of the templateizer program for automation of data management, processing of research results.

**Aigul G. Faizullina:** formulation of the main directions of the research, development of theoretical assumptions, formation of general conclusions.

**Zulfina Sh. Aglyamova:** carrying out data collection, preparation of the initial version of the article.

**Yulia N. Burkhanova:** scientific editing of the text of the article and final approval of the version for publication.

**Ruslan V. Spevakov:** analysis and interpretation of the obtained data, literary analysis.

### **ДАННЫЕ ОБ АВТОРАХ**

**Гареева Гульнара Альбертовна**, кандидат педагогических наук, доцент, заведующий кафедрой информационных систем  
*Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ*  
ул. Академика Королева, 1, г. Набережные Челны, 423814, Российская Федерация  
[gagareeva1977@mail.ru](mailto:gagareeva1977@mail.ru)

**Файзуллина Айгуль Гинатулловна**, старший преподаватель кафедры бизнес-информатики и математических методов в экономике  
*Казанский федеральный университет Набережночелнинский институт*  
проспект Мира, 68/19, г. Набережные Челны 423812, Российская Федерация  
[dlya\\_pisem\\_t@mail.ru](mailto:dlya_pisem_t@mail.ru)

**Аглямова Зульфина Шамилевна**, кандидат педагогических наук, доцент кафедры высшей математики, моделирования и анализа данных  
*Казанский инновационный университет им. В.Г. Тимирязова пр. Московский, 67, г. Набережные Челны, 423822, Российская Федерация*  
[aglamova@chl.ieml.ru](mailto:aglamova@chl.ieml.ru)

**Бурханова Юлия Николаевна**, кандидат педагогических наук, доцент, доцент кафедры высшей математики, моделирования и анализа данных; доцент кафедры математики, физики и методики их обучения

*Казанский инновационный университет им. В.Г. Тимирязова; Набережночелнинский государственный педагогический университет*

*пр. Московский, 67, г. Набережные Челны, 423822, Российская Федерация; ул. им. Низаметдинова Р.М., 28, г. Набережные Челны, 423806, Российская Федерация*  
*ulin2703@mail.ru*

**Спеваков Руслан Витальевич**, кандидат экономических наук, доцент кафедры высшей математики, моделирования и анализа данных

*Казанский инновационный университет им. В.Г. Тимирязова пр. Московский, 67, г. Набережные Челны, 423822, Российская Федерация*  
*spevakov@mail.ru*

#### **DATA ABOUT THE AUTHORS**

**Gulnara A. Gareeva**, Candidate of Pedagogical sciences, Associate professor, Head of the Department of Information Systems

*Kazan National Research Technical University named after A.N. Tupolev-KAI*

*1, Akademika Koroleva Str., Naberezhnye Chelny, 423814, Russian Federation*

*gagareeva1977@mail.ru*

*SPIN-code: 3279-8465*

*Scopus Author ID: 36801593200*

*ResearcherID: M-1728-2015*

*ORCID: <https://orcid.org/0000-0002-8539-4541>*

**Aigul G. Faizullina**, Senior Lecturer of the Department of Business Informatics and Mathematical Methods in Economics

*Kazan Federal University Naberezhnochelninsk Institute*

*68/19, Prospekt Mira, Naberezhnye Chelny 423812, Russian Federation*  
*dlya\_pisem\_t@mail.ru*

**Zulfina Sh. Aglyamova**, Candidate of Pedagogical Sciences, Associate Professor of the Department of Higher Mathematics, Modeling and Data Analysis  
*Kazan Innovation University named after V.G. Timiryasov*  
*67, Moskovsky Ave., Naberezhnye Chelny, 423822, Russian Federation*  
*aglamova@chl.ieml.ru*

**Yulia N. Burkhanova**, Candidate of Pedagogical Sciences, Associate Professor, Associate Professor of the Department of Higher Mathematics, Modeling and Data Analysis; Associate Professor of the Department of Mathematics, Physics and Methods of their Education  
*Kazan Innovation University named after V.G. Timiryasov; Naberezhnochelny State Pedagogical University*  
*67, Moskovsky Ave., Naberezhnye Chelny, 423822, Russian Federation; 28, R.M. Nizametdinov Str., Naberezhnye Chelny, 423806, Russian Federation*  
*ulin2703@mail.ru*

**Ruslan V. Spevakov**, PhD in Economics, Associate Professor of the Department of Higher Mathematics, Modeling and Data Analysis  
*Kazan Innovation University named after V.G. Timiryasov*  
*67, Moskovsky Ave., Naberezhnye Chelny, 423822, Russian Federation*  
*spevakov@mail.ru*

Поступила 01.03.2025  
После рецензирования 18.03.2025  
Принята 23.03.2025

Received 01.03.2025  
Revised 18.03.2025  
Accepted 23.03.2025