|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное автономное**  **образовательное учреждение высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

**ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*фамилия, имя, отчество*

Группа\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Тип практики Преддипломная практика

Название предприятия НУК ИУ МГТУ им. Н. Э. Баумана

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*подпись, дата фамилия, и.о.*

Руководитель практики **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кострицкий А. С.**

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*2025 г.*

**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное автономное образовательное учреждение высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э.Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой ИУ7

(Индекс)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.В. Рудаков

(И.О.Фамилия)

«14» мая 2025 г.

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ**

**на учебную / производственную практику**

Студент группы \_\_\_\_ИУ7-85Б\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Зуев Тимофей Александрович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, имя, отчество)

Сроки прохождения практики с 14.05.2025 г. по 27.05.2025 г.

##### Задание:

1. Реализовать на практике метод (программно-алгоритмический комплекс), разработанный в ходе выполнения выпускной квалификационной работы, в виде соответствующего программного обеспечения (ПО).

2. Реализовать разработанные в ходе выполнения технологической части выпускной квалификационной работы алгоритм/алгоритмы тестирования созданного ПО.

3. Исследовать характеристики разработанного программного обеспечения.

4. Рассмотреть при необходимости вопросы автоматического развёртывания разработанного ПО.

Дата выдачи задания «14» мая 2025 г.

**Руководитель практики от университета**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. С. Кострицкий

(Подпись, дата) (И.О.Фамилия)

**Руководитель практики от предприятия**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Подпись, дата) (И.О.Фамилия)

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

# Содержание

[Содержание 3](#_Toc199486930)

[Введение 4](#_Toc199486931)

[Цель работы 4](#_Toc199486932)

[Задачи 4](#_Toc199486933)

[Формализованная постановка задачи 5](#_Toc199486934)

[Функциональное тестирование 6](#_Toc199486935)

[Алгоритм тестирования 6](#_Toc199486936)

[Классы эквивалентности 7](#_Toc199486937)

[Результаты тестирования 9](#_Toc199486938)

[Модульное тестирование 14](#_Toc199486939)

[Тестирование графического интерфейса 14](#_Toc199486940)

[Исследование характеристик разработанного программного обеспечения 15](#_Toc199486941)

[Постановка и условия исследования 15](#_Toc199486942)

[Результаты исследования 15](#_Toc199486943)

[Выводы 17](#_Toc199486944)

[Заключение 18](#_Toc199486945)

# ВВЕДЕНИЕ

В условиях стремительного роста объёмов структурированных данных и усложнения требований к их целостности и непротиворечивости задача эффективного проектирования баз данных приобретает все большую актуальность. Основным инструментом обеспечения этих свойств является анализ формализованных семантических связей между атрибутами и служащие основой для процедур нормализации. Однако традиционные методы нормализации, основанные на ручном анализе, требуют значительных профессиональных ресурсов и не всегда применимы в динамических и масштабных системах. В связи с этим разработка автоматизированных методов нормализации является актуальной задачей в области проектирования баз данных.

## Цель работы

Цель работы — реализовать на практике метод, разработанный в ходе выполнения выпускной квалификационной работы; реализовать алгоритм тестирования созданного ПО, а также исследовать его характеристики.

## Задачи

Для достижения поставленной цели выделены следующие задачи:

1. Реализовать на практике метод (программно-алгоритмический комплекс), разработанный в ходе выполнения выпускной квалификационной работы, в виде соответствующего программного обеспечения (ПО).

2. Реализовать разработанные в ходе выполнения технологической части выпускной квалификационной работы алгоритм/алгоритмы тестирования созданного ПО.

3. Исследовать характеристики разработанного программного обеспечения.

# Формализованная постановка задачи

Во время выполнения выпускной квалификационной работы, задача автоматической нормализации в реляционных базах данных с использованием анализа функциональных зависимостей была формализована в виде нотации IDEF0 нулевого уровня. Формальная постановка задачи изображена на рисунке 1.

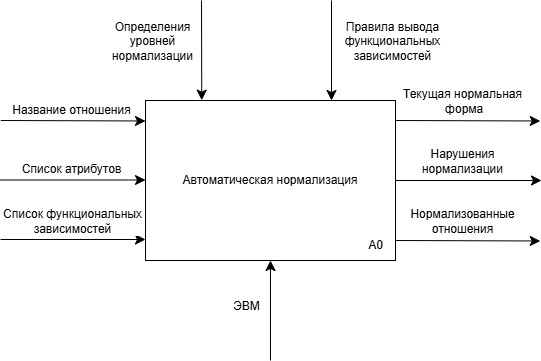


Рисунок 1. – Формальная постановка задачи

# Функциональное тестирование

## Алгоритм тестирования

На рисунке 2 изображена схема алгоритма тестирования разработанной реализации метода автоматической нормализации в реляционных базах данных.

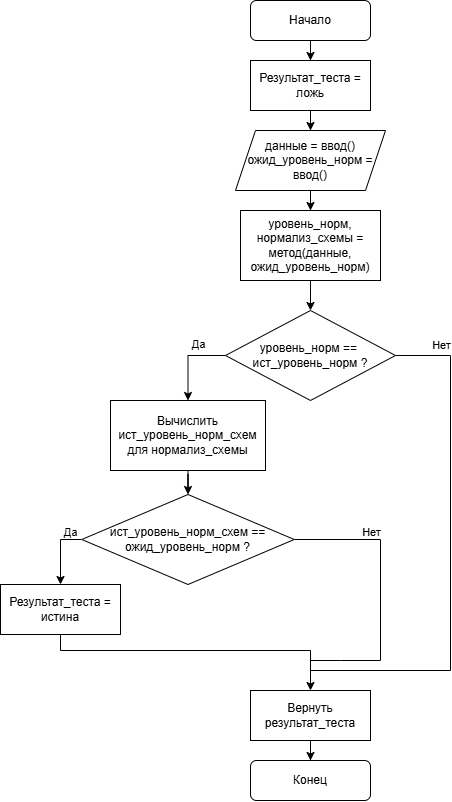


Рисунок 2. – Схема алгоритма тестирования

## Классы эквивалентности

В результате анализа предметной области для тестирования разработанной реализации метода автоматической нормализации были выделены следующие группы классов эквивалентности:

1. Группа классов эквивалентности по исходной нормальной форме отношения представлена в таблице 2.1.

|  |  |
| --- | --- |
| **Описание класса эквивалентности** | **Ожидаемый результат работы метода** |
| Отношение в целевой НФ | НФ отношения определена корректно, декомпозиции отношения к целевой НФ не происходит |
| Исходное отношение в 1НФ, есть нарушение 2НФ | НФ отношения определена как 1НФ, в нормализованных отношениях устранены частичные зависимости |
| Исходное отношение в 2НФ, есть нарушение 3НФ | НФ отношения определена как 2НФ, в нормализованных отношениях устранены транзитивные зависимости |
| Исходное отношение в 3НФ, есть нарушение НФБК | НФ отношения определена как 3НФ, в нормализованных отношениях устранены ФЗ, где детерминант не является суперключом. |
| Исходное отношение в НФБК, есть нарушение 4НФ | НФ отношения определена как 4НФ, в нормализованных отношениях устранены многозначные ФЗ |

Таблица 2.1. – Группа классов эквивалентности по исходной нормальной форме отношения.

1. Группа классов эквивалентности по структуре и количеству ключей представлена в таблице 2.2.

|  |  |
| --- | --- |
| **Описание класса эквивалентности** | **Ожидаемый результат работы метода** |
| В отношении один простой кандидатный ключ | Кандидатный ключ находится корректно |
| В отношении один составной кандидатный ключ | Кандидатный ключ находится корректно |
| Несколько пересекающихся кандидатных ключей | Кандидатный ключ находится корректно |
| Несколько непересекающихся кандидатных ключей | Кандидатный ключ находится корректно |

Таблица 2.2. – Группа классов эквивалентности по структуре и количеству ключей.

1. Группа классов эквивалентности по потере ФЗ при декомпозиции представлена в таблице 2.3.

|  |  |
| --- | --- |
| **Описание класса эквивалентности** | **Ожидаемый результат работы метода** |
| Декомпозиция без потери зависимостей | При нормализации все исходные ФЗ сохранены в результирующих нормализованных отношениях |
| Декомпозиция с потерей зависимостей | При нормализации потеряны ФЗ, нарушающие целевую НФ |

Таблица 2.3. – Группа классов эквивалентности по потере ФЗ при декомпозиции.

1. Группа классов эквивалентности по количеству и типу атрибутов и ФЗ представлена в таблице 2.4.

|  |  |
| --- | --- |
| **Описание класса эквивалентности** | **Ожидаемый результат работы метода** |
| Отношение без атрибутов | Программа сообщает, что для анализа требуются атрибуты |
| Отношение с атрибутами, но без заданных ФЗ | НФ отношения определяется как 4НФ, декомпозиция не требуется |
| Отношение с одним атрибутом | НФ отношения определяется как 4НФ, декомпозиция не требуется |
| Отношение, где заданы только тривиальные ФЗ | НФ отношения определяется как 4НФ, декомпозиция не требуется |

Таблица 2.3. – Группа классов эквивалентности по количеству и типу атрибутов и ФЗ.

## Результаты тестирования

Ниже приведены результаты тестирования разработанной реализации метода автоматической нормализации, объединенные по группам классов эквивалентности.

1. Результаты тестирования по группе классов эквивалентности по исходной нормальной форме отношения представлен в таблице 2. 4.

|  |  |  |  |
| --- | --- | --- | --- |
| **Класс эквивалентности и входные данные** | **Ожидаемый результат** | **Результат работы метода** | **Соответствие результата ожидаемому** |
| Отношение уже в целевой НФ (НФБК).  Отношение: *R(a(PK),b, c)*.  ФЗ: *a → (b, c).*  Целевой уровень: НФБК | НФ: НФБК.  R1: *(a(PK), b, c)*. | НФ: НФБК.  *R1*: *R(a(PK), b, c)*. | + |
| Нарушение 2НФ.  Отношение: *R(a(PK), b(PK), c, d, e)*.  ФЗ: *(a, b) → (e);*  *(b) → (c, d)*.  Целевой уровень: 2НФ | НФ: 1НФ  Нарушение:  *(b) → (c, d).*  R1: *(b, c, d)*.  R2: *(a, b, e).* | НФ: 1НФ  Нарушение:  *(b) → (c, d).*  R1: *(b, c, d)*.  R2: *(a, b, e).* | + |
| Нарушение 3НФ.  Отношение: *R(a(PK), b, c)*. ФЗ: *a → b*; *b → c*. Целевой уровень: 3НФ | НФ: 3НФ.  Нарушение:  *a → b*; *b → c* R1: *R(a(PK), b)* R2: *R(b(PK), c)* | НФ: 3НФ.  Нарушение:  *a → b*; *b → c* R1: *R(a(PK), b)* R2: *R(b(PK), c)* | + |
| Нарушение НФБК.  Отношение: *R(a(PK), b, c(PK), d(PK)).* ФЗ: *a → b; b → a* Целевой уровень: НФБК | НФ: НФБК. *R1: R(a(PK), b) R2: R(a(PK), c, d)* | НФ: НФБК. *R1: R(a(PK), b) R2: R(a(PK), c, d)* | + |
| Нарушение 4НФ.  Отношение: *R(a(PK), b, c).* ФЗ: *a → b; a → c.* Многозначные зависимости: *b →→ c.* Целевой уровень: 4НФ | НФ: 4НФ. *R1: R(b(PK), c(PK)) R2: R(a(PK), b)* | НФ: 4НФ. *R1: R(b(PK), c(PK)) R2: R(a(PK), b)* | + |

Таблица 2.4. – Результат тестирования по группе классов эквивалентности по исходной нормальной форме отношения.

1. Результаты тестирования по группе классов эквивалентности по структуре и количеству ключей представлен в таблице 2.5.

|  |  |  |  |
| --- | --- | --- | --- |
| **Класс эквивалентности и входные данные** | **Ожидаемый результат** | **Результат работы метода** | **Соответствие результата ожидаемому** |
| Один простой кандидатный ключ.  Отношение: *R(a, b, c).* ФЗ: *a → (b, c).* | Кандидатные ключи: *{a}* | Кандидатные ключи: *{a}* | + |
| Один составной кандидатный ключ.  Отношение: *R(a, b, c).* ФЗ: *(a, b) → c.* | Кандидатные ключи: *{a, b}* | Кандидатные ключи: *{a, b}* | + |
| Несколько пересекающихся кандидатных ключей.  Отношение: *R(a, b, c).* ФЗ: *(a, b) → c; (a, c) → b* | Кандидатные ключи: *{a, b}, {a, c}* | Кандидатные ключи: *{a, b}, {a, c}* | + |
| Несколько непересекающихся кандидатных ключей.  Отношение: *R(a, b, c, d).*  ФЗ: *(a, b) → (c, d); (c, d) → (a, b)* | Кандидатные ключи: *{a, b}, {c, d}* | Кандидатные ключи: *{a, b}, {c, d}* | + |

Таблица 2.5. – Результат тестирования по группе классов эквивалентности по структуре и количеству ключей.

1. Результаты тестирования по группе классов эквивалентности по потере ФЗ при декомпозиции представлен в таблице 2.6.

|  |  |  |  |
| --- | --- | --- | --- |
| **Класс эквивалентности и входные данные** | **Ожидаемый результат** | **Результат работы метода** | **Соответствие результата ожидаемому** |
| Декомпозиция без потери зависимостей. Отношение: *R(a(PK), b, c).* ФЗ: *a → b; a → c* | НФ: 3НФ. R1: *R(a(PK), b)* R2: *R(a(PK), c)* Сохранено зависимостей: 2 Потеряно зависимостей: 0 | НФ: 3НФ. R1: *R(a(PK), b)* R2: *R(a(PK), c)* Сохранено зависимостей: 2 Потеряно зависимостей: 0 | + |
| Декомпозиция с потерей зависимостей. Отношение: *R(a, b, c).* ФЗ: *(a, b) → c; c → a.* | НФ: НФБК. *R1: R1(c(PK), a) R2: R2(c(PK), b)* Сохранено зависимостей: 1 *(c → a)* Потеряно зависимостей: 1 *(a, b → c)* | НФ: НФБК. *R1: R1(c(PK), a) R2: R2(c(PK), b)* Сохранено зависимостей: 1 *(c → a)* Потеряно зависимостей: 1 *(a, b → c)* | + |

Таблица 2.6. – Результат тестирования по группе классов эквивалентности по потере ФЗ при декомпозиции.

1. Результаты тестирования по группе классов эквивалентности по количеству и типу атрибутов и ФЗ представлен в таблице 2.7.

|  |  |  |  |
| --- | --- | --- | --- |
| **Класс эквивалентности и входные данные** | **Ожидаемый результат** | **Результат работы метода** | **Соответствие результата ожидаемому** |
| Отношение без атрибутов. Отношение: *R()*. ФЗ: — | Ошибка: для анализа требуются атрибуты. | Ошибка: для анализа требуются атрибуты. | + |
| Отношение с атрибутами, но без заданных ФЗ. Отношение: *R(a, b, c).* ФЗ: — Целевой уровень: 4НФ | НФ: 4НФ. *R1: R(a(PK), b(PK), c(PK))* | НФ: 4НФ. *R1: R(a(PK), b(PK), c(PK))* | + |
| Отношение с одним атрибутом. Отношение: *R(a).* ФЗ: — Целевой уровень: 4НФ | НФ: 4НФ. *R1: R(a(PK))* | НФ: 4НФ. *R1: R(a(PK))* | + |
| Отношение, где заданы только тривиальные ФЗ. Отношение: *R(a, b).*  ФЗ: *a → a; b → b* | НФ: 4НФ. R1: *R(a(PK), b(PK))* | НФ: 4НФ. R1: *R(a(PK), b(PK))* | + |

Таблица 2.7. – Результат тестирования по группе классов эквивалентности по количеству и типу атрибутов и ФЗ.

# Модульное тестирование

Модульное тестирование не проводилось.

# Тестирование графического интерфейса

Тестирование графического интерфейса не проводилось.

# Исследование характеристик разработанного программного обеспечения

## Постановка и условия исследования

В данном разделе будет проведено исследование зависимости временных характеристик разработанной реализации метода автоматической нормализации реляционных баз данных от сложности входных данных, а также будет проведено сравнение с алгоритмом JMathNorm.

В рамках исследования зависимости временных характеристик разработанной реализации метода автоматической нормализации реляционных баз данных от сложности входных данных, будет замерено время выполнения анализа отношения и нормализации отношения до заданной целевой нормальной формы от следующих факторов:

1. Количество атрибутов в исходном отношении (N). В этом случае количество функциональных зависимостей фиксируется (M = 5), а N варьируется в рамках выбранного диапазона;
2. Количество функциональных зависимостей (ФЗ) в исходном отношении (M). В этом случае количество атрибутов фиксируется (N = 5), а M варьируется в рамках выбранного диапазона.

Так как JMathNorm не предусматривает анализ на текущий уровень нормализации, а только декомпозицию, сравнение алгоритмов должно проходить непосредственно по времени декомпозиции в зависимости от входных данных

Время на графике – среднее из 10 замеров

## Результаты исследования

Результаты исследования зависимости временных характеристик разработанной реализации метода автоматической нормализации реляционных баз данных от сложности входных данных и сравнения с алгоритмом JMathNorm представлены на рисунках 3.1 и 3.2.

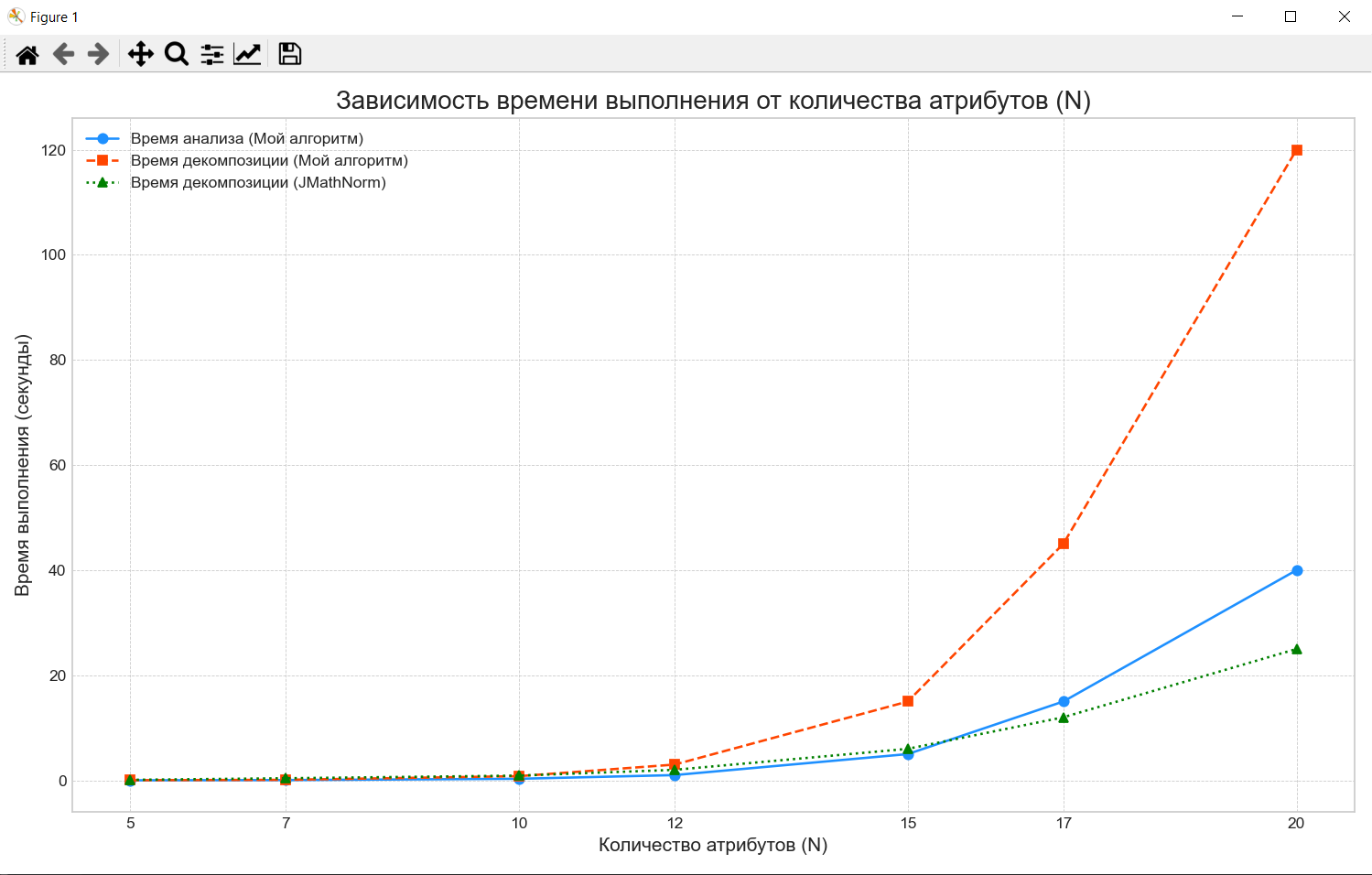


Рисунок 3.1. – График зависимости временных характеристик разработанной реализации метода от количества атрибутов и сравнение с алгоритмом JMathNorm

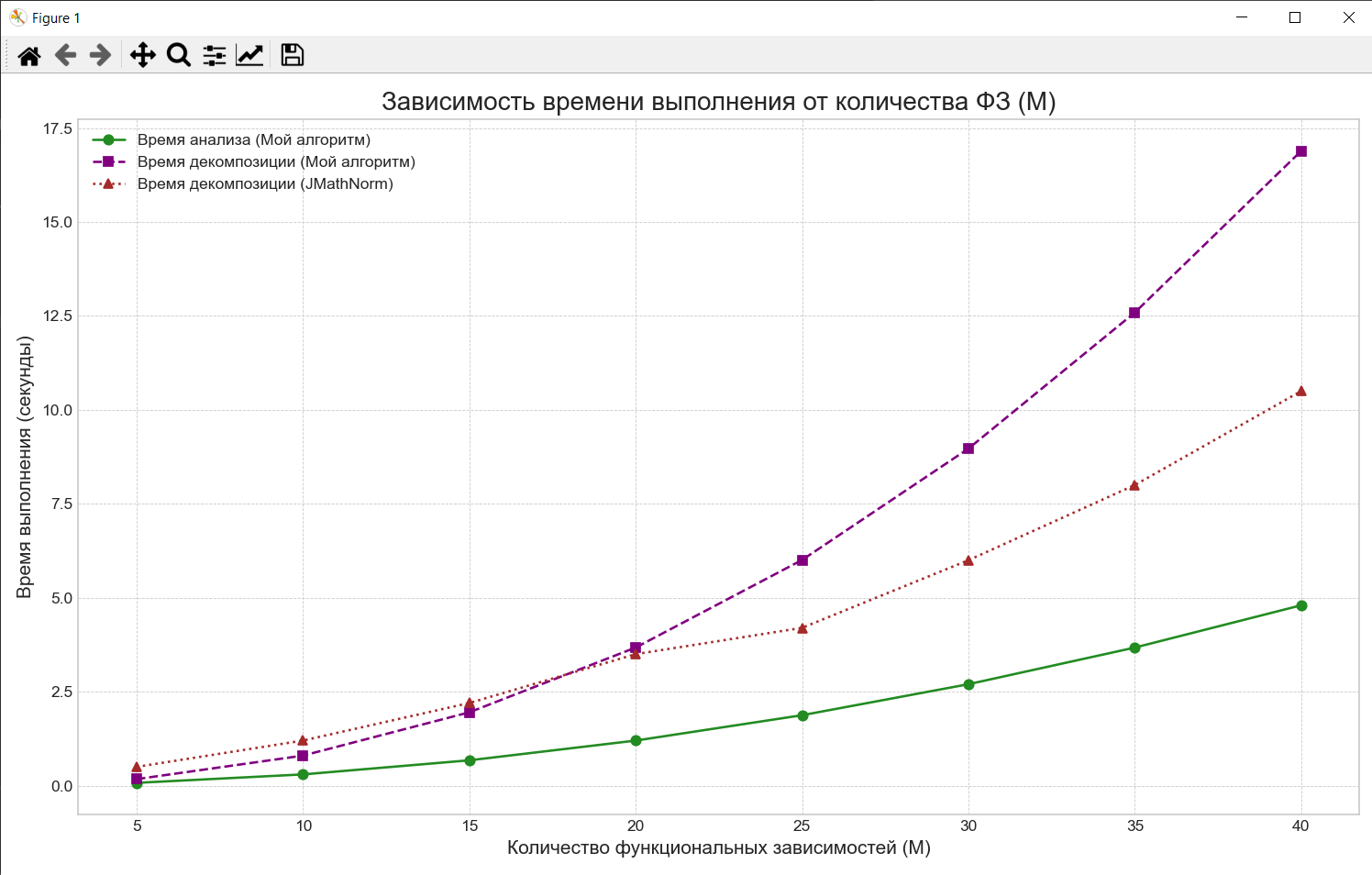


Рисунок 3.2. – График зависимости временных характеристик разработанной реализации метода от количества функциональных зависимостей и сравнение с алгоритмом JMathNorm.

## Выводы

В результате проведенного исследования можно сделать вывод о том, что время анализа и декомпозиции отношений растет по степенному закону. Это связано с комбинаторным характером разработанного алгоритма.

В сравнении с алгоритмом JMathNorm, разработанный метод показывает лучшую эффективность на N и M меньших десяти, но на больших значениях начинает отставать. Это может быть связано с выбранными средствами реализации в виду того, что алгоритм JMathNorm реализован на языке программирования Rust.

# Заключение

В рамках данной работы были выполнены все поставленные задачи. Была формализована постановка задачи разработки метода автоматической нормализации в реляционных базах данных с использованием анализа функциональных зависимостей была формализована в виде диаграммы в нотации IDEF0. Был описан алгоритм тестирования разработанной реализации метода, сформулированы и описаны классы эквивалентности. Было проведено тестирование по сформулированных классам эквивалентности, а также проведено исследование характеристик разработанной реализации метода автоматической нормализации в реляционных базах данных и проведено сравнение с алгоритмом JMathNorm.