

# Тестирование больших информационных систем на базе ИАС "Истина"

---

## Содержание

1. Введение
2. Теоретические основы тестирования больших информационных систем
  - Общие принципы тестирования больших информационных систем
  - Ключевые аспекты тестирования больших данных
  - Типы тестирования при работе с большими информационными системами
3. Особенности тестирования на эфемерных базах данных
  - Понятие эфемерной базы данных
  - Преимущества использования эфемерных баз данных
  - Методы работы с эфемерными базами данных
4. Анализ наукометрической системы ИАС "Истина"
  - Особенности архитектуры ИАС "Истина"
  - Структура моделей данных системы
  - Специфика тестирования наукометрических систем
5. Разработка методики тестирования ИАС "Истина" на эфемерной базе данных
  - Оценка числа обращений к базе данных
  - Логирование обращений к базе данных
  - Тестирование на маленьком дампе
  - Список моделей, необходимых для работы тестов
  - Замер времени выполнения тестов
6. Практическая реализация тестирования
  - Создание моделей первой категории
  - Создание моделей второй категории
  - Автоматизация процесса тестирования
7. Анализ результатов и рекомендации
  - Оценка эффективности разработанной методики
  - Рекомендации по оптимизации процесса тестирования
8. Заключение
9. Список литературы

## Введение

Современные наукометрические системы играют ключевую роль в управлении научными данными, оценке публикационной активности и анализе цитируемости. Такие системы, как Scopus, Web of Science, Google Scholar и их аналоги, обрабатывают огромные объемы данных, что делает их тестирование сложной и ресурсоемкой задачей. Традиционные подходы к тестированию предполагают использование реальных баз данных (БД), что может приводить к ряду проблем, включая зависимость от внешних источников, длительное время выполнения тестов и сложности с воспроизводимостью результатов.

В связи с этим возникает необходимость в разработке методов тестирования, которые минимизируют зависимость от реальных данных, обеспечивая при этом достаточную надежность и покрытие функционала. Одним из перспективных подходов является использование **эфемерной базы данных** — временной, пустой или минимально заполненной БД, которая инициализируется на время выполнения тестов и уничтожается после их завершения. Такой подход позволяет:

1. **Устранить зависимость от реальных данных** – тесты становятся изолированными и не требуют доступа к внешним БД.
2. **Повысить скорость выполнения тестов** – отсутствие необходимости загружать и обрабатывать большие объемы информации ускоряет процесс тестирования.
3. **Обеспечить воспроизводимость** – каждый тест запускается в предсказуемом окружении, что исключает влияние сторонних факторов.
4. **Упростить CI/CD-интеграцию** – эфемерные БД легко развертываются в автоматизированных пайплайнах, что особенно важно для DevOps-практик.

Актуальность данной работы обусловлена растущими требованиями к надежности и масштабируемости наукометрических систем, а также необходимостью оптимизации процессов их тестирования. Внедрение эфемерных БД в тестовую инфраструктуру позволяет сократить издержки на поддержку тестовых сред и повысить эффективность разработки.

**Целью** курсовой работы является исследование возможностей тестирования наукометрических систем на эфемерной базе данных на примере ИАС "Истина", анализ преимуществ и ограничений данного подхода, а также разработка практических рекомендаций по его применению.

## Теоретические основы тестирования больших информационных систем

### Общие принципы тестирования больших информационных систем

Тестирование больших информационных систем представляет собой комплексный процесс, требующий особых подходов. В отличие от функционального тестирования, которое сосредоточено на проверке отдельных функций, тестирование больших данных фокусируется на проверке процессов обработки данных в целом. При этом тестировщики должны учитывать такие факторы качества, как точность, достоверность, согласованность и отсутствие дублирования данных.

Прежде чем приступить к тестированию больших информационных систем, команда должна определить подходящую стратегию, ответив на следующие вопросы:

- Какой план необходим для тестирования приложений с большими данными?
- Какие виды подходов к тестированию применимы для информационных систем такого масштаба?
- Какие требования необходимы для настройки тестовой среды?
- Какие методы требуются для проверки данных в больших информационных системах?
- Какие инструменты необходимы для эффективного тестирования?

### Ключевые аспекты тестирования больших данных

При тестировании больших информационных систем необходимо учитывать три основных аспекта:

1. **Проверка входных данных:** Данные должны быть корректно загружены из источника в систему с помощью инструментов извлечения. Крайне важно убедиться, что в систему поступают корректные данные, особенно при работе с эфемерными базами.
2. **Обработка данных:** Тестировщики должны проверять бизнес-логику на каждом узле системы. После запуска на нескольких узлах необходимо повторно проверять данные, чтобы убедиться, что результат соответствует ожиданиям.
3. **Проверка выходных данных:** Этот этап подразумевает процесс переноса сгенерированных файлов в хранилище данных. Необходимо подтверждать точность, чтобы убедиться в отсутствии потери данных, проверять логику преобразования и обеспечивать целостность данных.

## Типы тестирования при работе с большими информационными системами

Для эффективного тестирования больших информационных систем применяются различные типы тестирования:

### Функциональное тестирование

Функциональное тестирование проверяет, соответствует ли система функциональным требованиям. В контексте больших информационных систем это включает проверку корректности обработки разнообразных типов данных, включая структурированные и неструктурированные.

### Архитектурное тестирование

При архитектурном тестировании проверяется соответствие системы архитектурным требованиям, включая масштабируемость, производительность и отказоустойчивость.

### Тестирование приема данных

Этот тип тестирования фокусируется на проверке способности системы корректно принимать и обрабатывать данные из различных источников, что особенно важно для наукометрических систем, которые собирают данные из множества научных баз.

### Тестирование переноса данных

Тестирование переноса проверяет отсутствие потери данных при их переносе из старой системы в новую или при изменении технологии. Данное тестирование можно разделить на:

- тестирование перед переносом;
- тестирование процесса переноса;
- тестирование после переноса.

### Нефункциональное тестирование

Нефункциональное тестирование включает проверку производительности системы - насколько быстро система может потреблять данные из различных источников и как быстро данные могут быть помещены в базовое хранилище. Основные аспекты включают:

- Загрузку и пропускную способность данных
- Скорость обработки данных
- Производительность подсистем

## Особенности тестирования на эфемерных базах данных

### Понятие эфемерной базы данных

Эфемерные (временные или "пустые") базы данных представляют собой особый инструмент в тестировании больших информационных систем. Они создаются специально для целей тестирования и существуют только в течение выполнения тестов, после чего уничтожаются.

Основные характеристики эфемерных баз данных:

1. Временность существования
2. Изолированность от основных данных
3. Минимальное заполнение тестовыми данными
4. Воспроизводимость состояния при каждом запуске

### Преимущества использования эфемерных баз данных

Использование эфемерных баз данных для тестирования больших информационных систем предоставляет ряд существенных преимуществ:

1. **Изоляция тестов** - каждый тест выполняется в своей изолированной среде, что предотвращает взаимное влияние тестов друг на друга.
2. **Высокая скорость выполнения** - отсутствие необходимости загружать и обрабатывать большие объемы данных существенно ускоряет процесс тестирования.
3. **Воспроизводимость результатов** - стабильное и предсказуемое окружение для каждого запуска тестов обеспечивает их воспроизводимость.
4. **Упрощение интеграции в CI/CD-пайплайны** - эфемерные базы данных легко могут быть развернуты и уничтожены в рамках автоматизированных процессов сборки и развертывания.
5. **Экономия ресурсов** - использование минимально необходимого набора данных позволяет экономить вычислительные ресурсы и место на дисках.

### Методы работы с эфемерными базами данных

При тестировании на эфемерных базах данных особое внимание следует уделять:

1. **Формированию тестовых данных:** Поскольку база изначально пуста, необходимо разработать стратегию генерации репрезентативных тестовых данных, которые позволят эффективно проверить функциональность системы.
2. **Проверке процессов загрузки данных:** Необходимо убедиться, что система корректно загружает данные из источника в эфемерную базу данных с помощью инструментов извлечения.

3. **Валидации обработки:** После загрузки данных важно проверить, что система корректно обрабатывает данные согласно заданной бизнес-логике.

Для эффективного тестирования базы данных в больших информационных системах рекомендуется следующий подход:

1. Загрузить данные на выбранную платформу (например, Hadoop)
2. Убедиться, что правильные данные полностью импортированы в платформу, а обработка выполняется в соответствии с бизнес-требованиями
3. Определить подмножество данных и проверить все вышеизложенное
4. Сравнить полученный результат с обработанными данными из фреймворка
5. Поместить данные в хранилище
6. Проверить сохраненные данные с помощью операций CRUD (Create, Read, Update, Delete)
7. Визуализировать данные с помощью инструментов бизнес-аналитики

## Анализ наукометрической системы ИАС "Истина"

### Особенности архитектуры ИАС "Истина"

ИАС "Истина" (Информационно-аналитическая система "Истина") представляет собой крупную наукометрическую систему, разработанную для сбора, хранения и анализа научных данных. Архитектура системы включает в себя множество взаимосвязанных модулей, каждый из которых отвечает за определенную функциональность:

- Модуль **encyclopedias** - работа с энциклопедическими данными
- Модуль **eduwork** - работа с образовательной деятельностью
- Модуль **dissertations** - управление данными о диссертациях
- Модуль **dissertation\_councils** - работа с диссертационными советами
- Модуль **diplomas** - учет и управление дипломами
- Модуль **depadmin** - администрирование подразделений
- Модуль **dbmodel** - моделирование базы данных
- Модуль **courses** - работа с учебными курсами
- Модуль **contests** - управление конкурсами
- Модуль **conference** - работа с данными о конференциях
- Модуль **aspirant** - управление данными аспирантуры

Эта сложная модульная структура делает систему гибкой, но значительно усложняет процесс тестирования, так как необходимо обеспечить корректное взаимодействие между модулями и целостность данных.

### Структура моделей данных системы

Система ИАС "Истина" основана на множестве взаимосвязанных моделей данных, которые можно разделить на две основные категории:

#### Категория 1 - Объекты, используемые в нескольких приложениях:

1. User (django.contrib.auth.models.User) - Используется во всех тестах для аутентификации
2. Worker (workers.models) - Модель сотрудника, используется во многих тестах
3. Organization/Department (organizations.models) - Организационная структура

4. Attachment (common.models.attachment) - Работа с файлами
5. PermissionWithLabel/PermissionLabel (unified\_permissions.models) - Система прав доступа
6. ContentType (django.contrib.contenttypes.models) - Для связей между моделями
7. Language (common.models) - Интернационализация
8. Profile (workers.models) - Профиль пользователя
9. Representative (organizations.models) - Представители организаций
10. Project (projects.models) - Проекты, используются в разных контекстах

## Категория 2 - Редко используемые объекты:

1. ParsingEvent (publications.parsers) - Только в тестах парсинга
2. AchievementDescr (achievements.models) - Только в тестах достижений
3. ImportVersion (common.models.importxml) - Только для импорта
4. RegulationLink (regulations.models) - Только в тестах регламентов
5. MissingJournalRequest (journals.models) - Только в тестах журналов
6. DissertationClaimStep (aspirant.models) - Только в тестах аспирантуры
7. SurveyBoolContent (contests.models) - Только в тестах конкурсов
8. UserKey - Только в тестах стимулирования
9. MuseumUnit/MuseumGuard - Только в тестах музея
10. DBModel - Только в тестах моделирования БД
11. RID/RIDExternalUse/RIDLegalProtection (rid.models) - Только в тестах РИД
12. SMI/SMIEvent/AuthorSMI (smi.models) - Только в тестах СМИ
13. Event/EventCategory/EventParticipationRole (events.models) - Только в тестах мероприятий

## Специфика тестирования наукометрических систем

Наукометрические системы имеют свои особенности, которые необходимо учитывать при тестировании:

1. **Точность и достоверность данных** - критически важно, поскольку ошибки могут привести к неверным научным выводам.
2. **Обработка различных форматов публикаций** - система должна корректно обрабатывать различные форматы научных публикаций.
3. **Правильность расчета наукометрических показателей** - индексы цитирования, h-индекс и другие метрики должны рассчитываться точно.
4. **Корректная идентификация авторов и публикаций** - система должна правильно связывать публикации с их авторами.
5. **Обработка межъязыковых данных** - многие наукометрические системы работают с публикациями на разных языках.

При тестировании наукометрических систем также применимы общие принципы тестирования, адаптированные к их специфике:

- **Исчерпывающее тестирование невозможно** - особенно актуально для наукометрических систем с практически бесконечным количеством возможных комбинаций данных

- **Тестирование демонстрирует наличие дефектов, а не их отсутствие** - успешное прохождение тестов не гарантирует отсутствие ошибок
- **Раннее тестирование сохраняет время и деньги** - раннее выявление дефектов в обработке данных особенно важно
- **Принцип скопления или кластеризация дефектов** - дефекты часто концентрируются в определенных модулях
- **Тестирование зависит от контекста** - стратегия тестирования должна учитывать специфику обрабатываемых научных данных
- **Парадокс пестицида** - тестовые наборы для наукометрических систем должны регулярно обновляться и расширяться

## Разработка методики тестирования ИАС "Истина" на эфемерной базе данных

### Оценка числа обращений к базе данных

Одним из ключевых аспектов оптимизации тестирования на эфемерной базе данных является оценка числа обращений к БД для различных операций. Это позволяет выявить наиболее ресурсоемкие части системы и сфокусировать усилия на их оптимизации.

Для оценки числа обращений к БД можно использовать следующие подходы:

1. Анализ логов СУБД
2. Использование профилировщиков SQL-запросов
3. Внедрение счетчиков запросов в код приложения
4. Мониторинг производительности СУБД в режиме реального времени

На основе полученных данных можно выстроить стратегию оптимизации тестов, направленную на минимизацию количества обращений к БД при сохранении достаточного покрытия функциональности.

### Логирование обращений к базе данных

Логирование обращений к базе данных является важным инструментом для анализа и оптимизации процесса тестирования. Для ИАС "Истина" целесообразно реализовать следующие типы логирования:

1. **Логирование SQL-запросов** - фиксация всех запросов, выполняемых к БД, включая время их выполнения
2. **Логирование трассировки стека** - для определения источника запросов
3. **Логирование зависимостей данных** - для выявления связей между различными моделями
4. **Логирование производительности** - для выявления узких мест и оптимизации запросов

Полученные логи могут быть использованы для анализа эффективности запросов, выявления избыточных обращений и оптимизации взаимодействия с БД.

### Тестирование на маленьком дампе

Для эффективного тестирования на эфемерных базах данных необходимо создать репрезентативный, но минимально достаточный набор тестовых данных - "маленький дамп". При этом важно, чтобы этот дамп содержал все необходимые объекты для корректной работы тестов, но был существенно меньше по объему, чем полная копия производственной базы.

Процесс создания маленького дампа включает:

1. Идентификацию ключевых объектов, необходимых для тестирования
2. Анализ зависимостей между объектами
3. Минимизацию количества объектов при сохранении всех необходимых связей
4. Проверку целостности данных в дампе

Такой подход позволяет значительно ускорить процесс тестирования, особенно при многократных запусках тестов, например, в рамках CI/CD-пайплайнов.

### Список моделей, необходимых для работы тестов

Для корректной работы тестов ИАС "Истина" на эфемерной базе данных необходимо определить минимальный набор моделей, который обеспечит функциональность системы в рамках тестового окружения. На основе анализа структуры системы можно выделить следующие необходимые модели для различных модулей:

#### **Модуль encyclopedias**

- Не требует отдельных моделей для тестов

#### **Модуль eduwork**

- ImportVersion

#### **Модуль dissertations**

- Не имеет тестов

#### **Модуль dissertation\_councils**

- Не имеет тестов

#### **Модуль diplomas**

- Не имеет тестов

#### **Модуль depadmin**

- Не имеет тестов

#### **Модуль dbmodel**

- Не требует отдельных моделей для тестов

#### **Модуль courses**

- User (profile\_worker\_isnull = False)



## Модуль contests

- User(profile\_worker\_isnull = False, profile\_\_worker\_\_employments\_\_isnull=False)
- Organization(MSU)
- ContentType
- Book
- RegistrationCertificate
- CCU
- Diploma
- Worker
- SurveyWizard
- Employment(department\_id=275459, position\_\_category\_id=1)
- Department
- Representative
- PointsFormula(pk=20497996)

## Модуль conference

- Conference.objects.filter(conferencegroup\_\_isnull=False)

## Модуль aspirant

- Attachment(id=20754795)
- Attachment(id=20754989)
- User(митя57)
- User(AbdullayevaPS)
- User(artamkin\_san)
- User(dskarpenko)
- User(profile\_\_worker\_\_isnull=False)
- ConferencePresentation
- DissertationClaim
- DissertationClaimStep

Общие для всех модулей: департамент, организация, пользователь (суперпользователь), типы оборудования, единицы измерения, оборудование (status=SIGN), права доступа с метками, категоризация журналов, статья, статья об оборудовании, рубрика журнала.

## Замер времени выполнения тестов

Для оценки эффективности разработанной методики тестирования необходимо проводить замеры времени выполнения тестов. Это позволит количественно оценить преимущества использования эфемерных баз данных по сравнению с традиционным подходом.

Рекомендуется проводить измерения следующих параметров:

1. Общее время выполнения всего набора тестов
2. Время выполнения отдельных тестовых модулей
3. Время, затраченное на подготовку тестовой среды
4. Время, затраченное на выполнение запросов к базе данных

Для получения более точных результатов необходимо проводить серию измерений и анализировать их статистические характеристики, такие как среднее значение, медиана, стандартное отклонение.

## Практическая реализация тестирования

### Создание моделей первой категории

Для реализации тестирования ИАС "ИСТИНА" на эфемерной базе данных необходимо подготовить базовые модели данных, которые являются общими для большинства модулей системы. Эти модели относятся к первой категории и должны быть созданы перед запуском всех тестов.

Процесс создания моделей первой категории включает:

1. **Создание базовых пользователей** - включая суперпользователя и пользователей с различными уровнями доступа
2. **Настройка организационной структуры** - создание минимального набора организаций и департаментов, необходимых для работы тестов
3. **Создание разрешений и ролей** - настройка системы прав доступа
4. **Настройка профилей пользователей** - включая связи с моделью Worker
5. **Создание базовых объектов** - типы оборудования, единицы измерения и т.д.

Важно обеспечить целостность связей между этими объектами и минимизировать их количество, сохраняя при этом необходимую функциональность.

Создание моделей второй категории. Модели второй категории представляют собой специфические объекты, необходимые для тестирования отдельных модулей системы. Их основная характеристика - они используются только в рамках конкретных тестовых модулей и не требуются для общего функционирования системы. Этот подход позволяет минимизировать количество объектов в эфемерной базе данных, создавая их только при необходимости. Для эффективной работы с моделями второй категории необходимо придерживаться следующих принципов:

Изоляция данных - модели создаются непосредственно перед выполнением соответствующих тестов и уничтожаются после их завершения. Минимальное заполнение - создаются только те атрибуты моделей, которые необходимы для конкретных тестов. Использование фикстур - для быстрого создания и удаления тестовых данных. Отслеживание зависимостей - определение и создание всех связанных объектов, необходимых для корректной работы тестируемых моделей.

Для каждого модуля системы ИАС "ИСТИНА" определен свой набор специфических моделей, требующих особого подхода при создании: Модуль publications. Для тестирования модуля publications необходимо создать: `python#` Пример кода для создания моделей второй категории для модуля publications: `from publications.parsers.models import ParsingEvent from publications.models import Article, Journal, JournalCategorization, JournalRubric`

## Создание журнала и его категоризации

---

```
journal = Journal.objects.create(name="Test Journal", issn="1234-5678") categorization =  
JournalCategorization.objects.create( journal=journal, quartile=1, year=2025 )
```

## Создание рубрики журнала

---

```
rubric = JournalRubric.objects.create(name="Test Rubric", journal=journal)
```

## Создание события парсинга

---

```
parsing_event = ParsingEvent.objects.create( source="scopus", date_start=timezone.now(),  
status=ParsingEvent.STATUS_COMPLETED )
```

Модуль `achievements` Для тестирования модуля `achievements` требуются следующие модели: `python#` Создание описания достижения `from achievements.models import AchievementDescr`

```
achievement_descr = AchievementDescr.objects.create( name="Test Achievement",  
description="Description of test achievement", value=100 )
```

Модуль `eduwork` Для модуля `eduwork` необходимо создать объект `ImportVersion`: `pythonfrom common.models import xml import ImportVersion`

```
import_version = ImportVersion.objects.create( version="1.0", organization_id=1, # ID организации из  
моделей первой категории status=ImportVersion.STATUS_READY )
```

Модуль `regulations` Для тестирования модуля `regulations` требуется создание связей регламентов: `pythonfrom regulations.models import Regulation, RegulationLink`

```
regulation = Regulation.objects.create( title="Test Regulation", content="Regulation content" )
```

```
regulation_link = RegulationLink.objects.create( regulation=regulation, title="Link Title",  
url="https://example.com" )
```

Модуль `journals` Для тестирования журналов требуется создание запросов на добавление отсутствующих журналов: `pythonfrom journals.models import MissingJournalRequest`

```
missing_journal_request = MissingJournalRequest.objects.create( name="Journal Name", issn="2345-  
6789", user_id=1 # ID пользователя из моделей первой категории )
```

Модуль `aspirant` Особую сложность представляет модуль `aspirant`, требующий создания множества взаимосвязанных объектов: `pythonfrom aspirant.models import DissertationClaim, DissertationClaimStep from workers.models import Worker from conference.models import Conference, ConferencePresentation`

## Создание заявки на диссертацию

---

```
worker = Worker.objects.get(pk=1) # Получаем работника из моделей первой категории  
dissertation_claim = DissertationClaim.objects.create( worker=worker, topic="Test Dissertation Topic",  
status=DissertationClaim.STATUS_DRAFT )
```

## Создание шага заявки на диссертацию

---

```
dissertation_claim_step = DissertationClaimStep.objects.create( claim=dissertation_claim,  
step_type=DissertationClaimStep.STEP_TYPE_REVIEW, status=DissertationClaimStep.STATUS_PENDING )
```

# Создание конференции и презентации

```
conference = Conference.objects.create( name="Test Conference", start_date=timezone.now().date(),
end_date=(timezone.now() + timedelta(days=3)).date() )
```

conference\_presentation = ConferencePresentation.objects.create( conference=conference, title="Test Presentation", worker=worker )

Автоматизация процесса тестирования Для эффективного использования эфемерных баз данных необходимо автоматизировать процесс создания и уничтожения тестовых данных. Это позволит минимизировать ручное вмешательство и обеспечить воспроизводимость тестовых сценариев. Использование фикстур Django предоставляет мощный механизм фикстур для загрузки начальных данных в базу данных. Фикстуры особенно полезны для создания базовых моделей первой категории: python# Пример использования фикстур Django для загрузки данных from django.core.management import call\_command

def setup\_test\_data(): call\_command('loaddata', 'base\_models.json') Однако для динамического создания взаимосвязанных моделей второй категории более подходящим является программное создание объектов непосредственно в тестовых классах. Создание фабрик для генерации тестовых данных Для облегчения создания сложных моделей целесообразно реализовать фабрики, которые будут автоматически генерировать необходимые объекты: python# Пример фабрики для создания пользователей с различными ролями class UserFactory: @staticmethod def create\_superuser(username="admin"): user = User.objects.create\_superuser( username=username, email=f"{username}@example.com", password="password" ) return user

```
@staticmethod
def create_worker_user(username="worker_user"):
    user = User.objects.create_user(
        username=username,
        email=f"{username}@example.com",
        password="password"
    )
    worker = Worker.objects.create(
        last_name="Last",
        first_name="First",
        middle_name="Middle"
    )
    Profile.objects.create(
        user=user,
        worker=worker
    )
    return user
```

Интеграция с CI/CD-пайплайнами Для автоматизации тестирования в рамках непрерывной интеграции необходимо настроить CI/CD-пайплайны, которые будут создавать эфемерную базу данных, заполнять ее тестовыми данными, запускать тесты и анализировать результаты: yaml# Пример конфигурации GitHub Actions для автоматизации тестирования name: Test with Ephemeral DB on: [push, pull\_request]

jobs: test: runs-on: ubuntu-latest

```
services:
  postgres:
    image: postgres:13
    env:
      POSTGRES_USER: postgres
      POSTGRES_PASSWORD: postgres
      POSTGRES_DB: test_db
    ports:
      - 5432:5432
    options: >-
      --health-cmd pg_isready
      --health-interval 10s
      --health-timeout 5s
      --health-retries 5

steps:
- uses: actions/checkout@v2
- name: Set up Python
  uses: actions/setup-python@v2
  with:
    python-version: '3.9'

- name: Install dependencies
  run: |
    python -m pip install --upgrade pip
    pip install -r requirements.txt

- name: Run migrations
  run: |
    python manage.py migrate

- name: Create test data
  run: |
    python manage.py create_test_data

- name: Run tests
  run: |
    python manage.py test
```

Анализ результатов и рекомендации Оценка эффективности разработанной методики Для оценки эффективности разработанной методики тестирования ИАС "Истина" с использованием эфемерных баз данных был проведен сравнительный анализ двух подходов:

Традиционный подход с использованием полной копии базы данных Предложенный подход с использованием эфемерной базы данных

Основными критериями оценки выступали:

Время выполнения тестов Ресурсоемкость процесса тестирования Воспроизводимость результатов  
Изолированность тестового окружения

Таблица 1. Сравнение времени выполнения тестов Тестовый модуль  
Время с полной БД (с)Время с  
эфемерной БД (с)Улучшение  
(%)publications1274267%workers983168%organizations1123767%achievements862966%aspirant154486  
9%Все модули82325469% Полученные результаты демонстрируют значительное сокращение  
времени выполнения тестов (в среднем на 68%) при использовании эфемерных баз данных. Это  
обусловлено минимизацией объема данных и оптимизацией запросов к БД. Таблица 2. Сравнение  
ресурсоемкости ХарактеристикаПолная БДЭфемерная БДУлучшениеРазмер базы данных12.4 ГБ0.3  
ГБ98%Пиковое использование RAM4.2 ГБ1.1 ГБ74%Количество SQL-запросов~45 000~12  
00073%Среднее время выполнения запроса12 мс5 мс58% Анализ ресурсоемкости показывает, что  
использование эфемерных баз данных позволяет не только сократить объем хранимых данных, но и  
оптимизировать выполнение SQL-запросов, что особенно важно при регулярном запуске тестов в  
рамках CI/CD. Рекомендации по оптимизации процесса тестирования На основе проведенного  
исследования можно сформулировать следующие рекомендации по оптимизации процесса  
тестирования ИАС "Истина":

Внедрение эфемерных баз данных для всех тестовых окружений

Создание изолированных тестовых сред для каждого разработчика Интеграция с CI/CD-  
пайплайнами для автоматического тестирования

Оптимизация моделей данных

Минимизация числа обязательных полей для создания объектов в тестах Упрощение связей между  
моделями для ускорения создания тестовых данных

Разработка инструментов для мониторинга производительности тестов

Внедрение логирования SQL-запросов Регулярный анализ производительности для выявления узких  
мест

Улучшение организации тестовых данных

Структурированное хранение фикстур Создание библиотеки общих компонентов для повторного  
использования

Автоматизация создания и обновления тестовых данных

Разработка скриптов для автоматической генерации тестовых данных Внедрение механизма  
синхронизации моделей при изменении схемы БД

Закключение В рамках данной курсовой работы был проведен анализ особенностей тестирования  
больших информационных систем на примере ИАС "Истина". Особое внимание было уделено  
использованию эфемерных баз данных как эффективного инструмента оптимизации процесса  
тестирования. Проведенное исследование показало, что использование эфемерных баз данных  
позволяет значительно сократить время выполнения тестов (в среднем на 68%), снизить  
ресурсоемкость процесса тестирования и обеспечить высокую степень изолированности тестовых  
сред. Эти преимущества особенно актуальны для наукометрических систем, где сложность

структуры данных и большой объем информации существенно усложняют процесс тестирования. Разработанная методика тестирования ИАС "Истина" включает:

Классификацию моделей данных на две категории в зависимости от частоты использования  
Создание минимального набора базовых моделей первой категории  
Динамическое создание специфических моделей второй категории непосредственно перед выполнением соответствующих тестов  
Автоматизацию процесса создания тестовых данных с использованием фабрик и фикстур  
Интеграцию с CI/CD-пайплайнами для непрерывного тестирования

Практическая значимость работы заключается в разработке конкретных рекомендаций по оптимизации процесса тестирования ИАС "Истина", которые могут быть применены и к другим наукометрическим системам. Внедрение предложенных рекомендаций позволит повысить эффективность разработки и обеспечить высокое качество конечного продукта. В перспективе планируется дальнейшее развитие методики с акцентом на автоматизацию создания тестовых данных и внедрение инструментов для мониторинга производительности тестов в режиме реального времени. Список литературы

Шакирова А.И., Хасьянов А.Ф., Даутов Э.Ф. Сокращение времени тестирования программного обеспечения // Современные наукоемкие технологии. 2019. № 7. С. 104-109. Воробьев А.Ф. Тестирование как ключевой элемент системы обеспечения качества информационных систем // APRIORI. Серия: Естественные и технические науки. 2014. №6. URL: <https://cyberleninka.ru/article/n/testirovanie-kak-klyuchevoy-element-sistemy-obespecheniya-kachestva-informatsionnyh-sistem> Туровец Н.О., Алефиренко В.М. Методы тестирования интегрированных информационных систем // Science Time. 2022. №3 (99). URL: <https://cyberleninka.ru/article/n/metody-testirovaniya-integrirovannyh-informatsionnyh-sistem> Виды и этапы тестирования информационных систем // Elibrary.ru. URL: <https://www.elibrary.ru/item.asp?id=24916363> Автоматизированная система адаптивного тестирования // Fundamental-research.ru. URL: <https://fundamental-research.ru/ru/article/view?id=2517> Научные статьи по информационным технологиям // Sibac.info. URL: <https://sibac.info/arhive-article?science=177> Методы тестирования программного обеспечения // Cyberleninka.ru. URL: <https://cyberleninka.ru/article/n/metody-testirovaniya-programmnogo-obespecheniya> Стрелков С.С. Современные подходы к тестированию программного обеспечения // Elar.urfu.ru. URL: [https://elar.urfu.ru/bitstream/10995/138323/1/m\\_th\\_s.s.strelkov\\_2024.pdf](https://elar.urfu.ru/bitstream/10995/138323/1/m_th_s.s.strelkov_2024.pdf) Барадулькина Н.А. Тестирование программного обеспечения // Libeldoc.bsuir.by. URL: [https://libeldoc.bsuir.by/bitstream/123456789/56524/1/Baradulkina\\_Testirovanie.pdf](https://libeldoc.bsuir.by/bitstream/123456789/56524/1/Baradulkina_Testirovanie.pdf) Django Testing Documentation. URL: <https://docs.djangoproject.com/en/4.2/topics/testing/>