

ПРИМЕНЕНИЕ ЭКСПЕРТНОЙ СИСТЕМЫ ДЛЯ ОПТИМИЗАЦИИ УПРАВЛЕНИЯ ОБЪЕКТАМИ В РЕЛЯЦИОННОЙ БАЗЕ ДАННЫХ

Франтасов Д.Н., канд. техн. наук, доцент, начальник управления по обеспечению цифровой трансформации ФГАОУ ВО «Самарский государственный экономический университет», Самара
Лёшин Н.С., студент магистратуры ФГАОУ ВО «Самарский государственный экономический университет», Самара

Прокопов А.В., студент магистратуры ФГАОУ ВО «Самарский государственный экономический университет», Самара

Разработка автоматизированных систем оптимизации реляционных баз данных становится ключевым направлением для повышения их производительности и снижения эксплуатационных затрат. В работе предлагается концепция экспертной системы, формирующей рекомендации на основе анализа метаданных и конфигурационных параметров. Проведен обзор существующих подходов, включая ручные методы, ML/AI и гибридные модели. Представлена архитектура системы, а также модель оценки ее эффективности с учетом технических и организационных факторов. Предложена классификация типов рекомендаций, охватывающая различные аспекты сопровождения СУБД. Рассматриваемый подход обеспечивает баланс между автоматизацией и экспертным контролем, позволяя повысить устойчивость и управляемость информационной инфраструктуры.

Ключевые слова: экспертная система, оптимизация, метаданные, рекомендации.

Цитирование: Франтасов Д.Н., Лёшин Н.С., Прокопов А.В. Применение экспертной системы для оптимизации управления объектами в реляционной базе данных // Информационно-экономические аспекты стандартизации и технического регулирования. 2025. № 4(85). С. 64–75.

ВВЕДЕНИЕ

В условиях стремительного роста объемов данных и сложности инфраструктур реляционные базы данных остаются основой корпоративных информационных систем. Обеспечение их производительности и устойчивости требует значительных усилий со стороны сотрудников, работающих с базой данных [1]. Существующие оптимизации все чаще оказываются неэффективными: они зависят от квалификации персонала, сопряжены с высокими затратами и не гарантируют стабильных результатов.

Одним из перспективных направлений решения данной проблемы является применение экспертных систем, способных автоматически формировать рекомендации по настройке объектов базы данных на основе анализа конфигурации и метаданных. В работе «DB-BERT: A Database Tuning Tool that 'Reads the Manual'» представлена система, которая анализирует текстовые документы и метаданные для оптимизации параметров СУБД [2]. Кроме того, в исследовании «GPTuner: A Manual-Reading Database Tuning System via GPT-Guided Bayesian Optimization» описывается подход, использующий метаданные и документацию для автоматической настройки конфигурации баз дан-

ных [3]. Внедрение подобных систем позволяет достичь следующих преимуществ: повышение общей производительности системы управления данными, сокращение временных затрат на администрирование, уменьшение экономических расходов, связанных с привлечением высококвалифицированных специалистов, а также минимизацию рисков, обусловленных человеческим фактором.

С методологической точки зрения реляционные базы данных реализуют принципы реляционной модели данных [1]. Несмотря на появление за последние пять десятилетий альтернативных парадигм управления данными (документно-ориентированных, графовых, систем типа «ключ-значение» и других) реляционные СУБД сохраняют лидирующие позиции по распространенности и востребованности. Это обусловлено рядом неоспоримых преимуществ, включающих: эффективные механизмы управления данными, строгую систему поддержки целостности и согласованности информации [4]. Согласно актуальным данным рейтинга DB-Engines (апрель 2025 года), к наиболее популярным реализациям реляционных СУБД относятся PostgreSQL, MySQL, Oracle Database и Microsoft SQL Server [5]. Рейтинг использования систем управления базами данных представлен в табл. 1.

Таблица 1

Рейтинг использования систем управления базами данных

РЕЙТИНГ		НАИМЕНОВАНИЕ СУБД	ТИП СУБД
АПРЕЛЬ 2024	АПРЕЛЬ 2025		
1	1	Oracle	Реляционная
2	2	MySQL	Реляционная
3	3	Microsoft SQL Server	Реляционная
4	4	PostgreSQL	Реляционная
5	5	MongoDB	Документно-ориентированная
9	6	Snowflake	Реляционная
6	7	Redis	СУБД типа Ключ-Значение
7	8	Elasticsearch	Поисковая
8	9	IBM Db2	Реляционная
10	10	SQLite	Реляционная

С увеличением накопленного объема данных, развития различных моделей построения хранилищ каждая организация сталкивается с необходимостью в регулярной оптимизации для поддержания высокой производительности, доступности и надежности данных. Основными задачами для достижения данных целей являются:

- оптимизация скорости выполнения запросов;
- снижение затрат на хранение данных;
- потребность наличия в штате сотрудников соответствующей квалификации;
- поддержание целостности и консистентности данных;
- обеспечение масштабируемости и адаптивности к изменениям нагрузки.

В рамках данного исследования проведен анализ существующих подходов для решения обозначенной проблемы. Были систематизированы ключевые процессы, задействованные в процессе оптимизации. На основе полученных результатов разработана концепция решения, предполагающая внедрение экспертной системы, способной оптимизировать рассматриваемые операции.

ЦЕЛИ И ЗАДАЧИ

В качестве ключевого метода исследования применялся системный анализ, включающий:

- изучение принципов работы реляционных СУБД;
- анализ современных подходов к оптимизации баз данных;

- формализацию критериев оценки эффективности структур данных.

Целью данного исследования является разработка и обоснование методологии применения экспертных систем для автоматизированной оптимизации структурных объектов реляционных баз данных. В работе ставится задача создания концепции экспертной системы, способной генерировать рекомендации по повышению эффективности управления данными, что позволит снизить эксплуатационные затраты.

В качестве экспериментальной платформы для разработки экспертной системы была выбрана система управления базами данных PostgreSQL. Данный выбор обусловлен следующими факторами:

- широкое распространение в российских организациях. Согласно исследованию Фонда «Центр стратегических разработок» и аналитического портала TAdviser, PostgreSQL входит в число наиболее популярных СУБД в российских компаниях [6];
- открытость и свободное распространение [7];
- соответствие требованиям импортозамещения в Российской Федерации.

В качестве ключевых этапов проведения исследования можно выделить следующие:

- анализ существующих подходов оптимизации;
- методология формирования базы знаний экспертной системы;
- формализация правил генерации рекомендаций;
- сравнительный анализ эффективности существующих и предложенного подходов;
- проектирование архитектуры экспертной системы, ориентированной на автоматизированную оптимизацию структуры реляционной базы данных;
- разработана модель оценки эффективности экспертной системы после внедрения.

ОСНОВНАЯ ЧАСТЬ

1. Анализ существующих подходов оптимизации

Администрирование реляционных баз данных предполагает выполнение широкого спектра операций по управлению объектами, включая их создание, модификацию и удаление. Данный процесс требует глубоких специализированных знаний о структуре данных, особенностях выполняемых запросов в организации и специфике используемой системы управления базами данных. Сложность существенно возрастает в условиях динамически изменяющихся требований и необходимости постоянного поддержания оптимального уровня производительности системы.

Например, некорректное добавление индексов или партиций приводит к повышенному потреблению памяти и существенному росту времени выполнения запросов. Согласно данным тестирования по стандарту TPC-H, подобные субоптимальные настройки могут увеличивать время отклика системы на 30–70% [8].

Дополнительные системные ограничения связаны со спецификой механизмов оптимизации в различных СУБД, например, таких как PostgreSQL, Oracle и MySQL. Отсутствие унифицированных стандартов управления производительностью усложняет процесс администрирования, а экспоненциальный рост сложности при масштабировании системы (проблема N+1) создает дополнительные операционные риски [4].

Экономические последствия включают повышенные затраты на содержание высококвалифицированных сотрудников, увеличивающие фонд оплаты труда на 15–20%, простои при устранении последствий ошибочных конфигураций – до 7% рабочего времени (по данным Gartner, 2024) [9], а также необходимость постоянного обучения персонала.

Если рассматривать автоматизированный подход, то стоит посмотреть в сторону таких современных решений, как OtterTune, Index Advisor в Microsoft SQL Server, Amazon Aurora Autopilot [10–13], а также аналогичные модули в Oracle и PostgreSQL; они используют методы машинного обучения и эвристического анализа для автоматического подбора конфигураций и оптимальных структур индексов.

Однако несмотря на очевидные преимущества автоматизированные системы не лишены недостатков. Одним из ключевых рисков является непрозрачность принимаемых решений: алгоритмы зачастую действуют как «черный ящик», предлагая изменения без возможности полноценно оценить их логику и последствия.

2. Методология формирования базы знаний экспертной системы

Для того чтобы экспертная система генерировала рекомендации и процесс превратился из ручного в автоматизированный, необходимо найти источники информации, для анализа с последующей генерацией рекомендаций. Конфигурационные файлы и метаданные могут выступать такими источниками.

Метаданные содержат информацию о структуре таблиц, индексов, ключей, а также статистику о частоте выполнения запросов, распределении данных и других аспектах. Использование метаданных позволяет:

1. Анализировать текущее состояние объектов базы данных.
2. Выявлять узкие места в производительности.
3. На основе проанализированной информации генерировать рекомендации.

Примером полезных метаданных являются статистические данные о частоте использования индексов. Анализ таких данных позволяет определить, какие индексы избыточны или, напротив, отсутствуют, но могли бы улучшить производительность.

Формирование базы знаний для экспертной системы оптимизации реляционных СУБД требует комплексного подхода к сбору и анализу исходных данных. Конфигурационные параметры системы и метаданные базы данных являются перспективным решением данной задачи.

Конфигурационные параметры СУБД, включая настройки сервера, параметры планировщика запросов и ограничения ресурсов, представляют фундаментальную основу для анализа. Эти параметры непосредственно влияют на производительность системы и определяют границы возможных оптимизаций. Особое значение имеют динамически изменяемые настройки, требующие постоянного мониторинга и корректировки.

Метаданные системы содержат исчерпывающую информацию о структурных элементах базы данных, статистике выполнения запросов и характеристиках распределения данных. Анализ метаданных позволяет не только оценить текущее состояние системы, но и выявить скрытые резервы повышения производительности.

Процесс формирования базы знаний включает несколько взаимосвязанных этапов:

- первоначальный сбор данных с последующей нормализацией и структурированием полученной информации;
- формализация взаимосвязей между различными компонентами системы;

3. Формализация правил генерации рекомендаций

Следующим этапом создаются правила анализа и механизмы логического вывода, обеспечивающие генерацию обоснованных рекомендаций. Процесс создания правил включает в себя несколько источников:

- систематизированное извлечение знаний из учебной литературы, научных публикаций и официальной документации;
- привлечение экспертов в области администрирования и оптимизации PostgreSQL с целью формализации практического опыта в виде продуктивных эвристик и экспертных правил.

Для каждой рекомендации может быть определен уровень её применения. В рамках настоящего исследования выделяются два основных уровня: объектный и системный. Объектный уровень подразумевает, что рекомендация направлена на конкретные элементы структуры базы данных.

Системный уровень относится к рекомендациям, предполагающим изменение глобальных параметров.

Дополнительно для каждой рекомендации определяется уровень критичности, отражающий степень потенциального влияния неоптимальной настройки на производительность или стабильность базы данных. Предлагается использовать пятиуровневую шкалу:

- 1** — незначительное влияние, рекомендация носит факультативный характер;
- 2–3** — умеренное влияние, изменения могут улучшить эффективность и предупредить потенциальные проблемы;
- 4** — значительное влияние, требуется приоритетное внимание;
- 5** — критическое влияние, рекомендация подлежит немедленному выполнению.

Также введена категоризация по типу рекомендаций, что позволяет структурировать знания в зависимости от области применения. В рамках данной работы выделены следующие категории: изменения структурных объектов, индексация и стратегии доступа к данным, конфигурационные параметры и настройки окружения, автообслуживание, включая очистку неиспользуемого пространства и обновление статистики оптимизатора запросов с помощью выполнения операций VACUUM и ANALYZE, журналирование и мониторинг, расширения и инфраструктура (включая репликацию, бэкапы, архитектуру).

В табл. 2 представлено несколько примеров правил, сформированных в соответствии с предложенной классификацией.

Таблица 2

Примеры правил, сформированных в соответствии с предложенной классификацией

УСЛОВИЕ	РЕКОМЕНДАЦИЯ	УРОВЕНЬ ПРИМЕНЕНИЯ	КРИТИЧНОСТЬ	КАТЕГОРИЯ
Если количество атрибутов в таблице превышает 50	Рекомендуется провести декомпозицию структуры	Объектный	3	Изменения структурных объектов
Если в запросах активно используются приведения типов (::date, CAST()) по индексированным полям	Рекомендуется привести тип поля к наиболее часто используемому в выражениях типу	Объектный	2	Оптимизация запросов / Индексация
Если в таблице присутствуют поля типа json, jsonb или text, которые не используются в активных запросах	Рекомендуется вынести данные поля в отдельную сущность	Объектный	2	Изменения структурных объектов
Если объем таблицы превышает 10 ГБ	Рекомендуется реализовать секционирование	Объектный	3	Хранение / Изменения структурных объектов
Если поле активно используется в WHERE или JOIN и обладает высокой селективностью	Рекомендуется создать индекс	Объектный	4	Индексация и стратегии доступа
Если индекс не участвует в планах выполнения запросов (EXPLAIN) или дублируется более эффективным составным индексом	Рекомендуется удалить или оптимизировать индекс	Объектный	2	Индексация и стратегии доступа
Если объем таблицы превышает 10 ГБ и параметр autovacuum_vacuum_scale_factor установлен в значение по умолчанию (0.2)	Рекомендуется индивидуально снизить данный параметр	Объектный	2	Автообслуживание и VACUUM/ANALYZE
Если отключены параметры autovacuum и/или track_counts	Рекомендуется включить параметры	Системный	5	Конфигурационные параметры
Если для таблицы длительное время не проводился ANALYZE	Рекомендуется принудительный сбор статистики	Объектный	3	Автообслуживание и VACUUM/ANALYZE
Если в системе используется более 100 индексов и сервер обладает многопроцессорной архитектурой	Рекомендуется использовать reindexdb -j N	Системный	2	Обслуживание / Инфраструктура

Важнейшей характеристикой предлагаемой экспертной системы является поддержка механизма добавления пользовательских правил оптимизации. Такая функциональность позволяет адаптировать систему к специфическим требованиям конкретной организации, учитывая уникальные аспекты конфигурации СУБД и бизнес-процессов.

Ключевое значение здесь имеет обязательное требование к документации всех вносимых изменений. Каждое новое правило должно сопровождаться подробным техническим обоснованием, объясняющим его необходимость и ожидаемый эффект.

4. Сравнительный анализ эффективности существующих и предложенного подходов

На основании проведенного исследования был выполнен систематический сравнительный анализ существующих и предложенного подходов к оптимизации реляционных баз данных. Результаты сравнения представлены в табл. 3.

Преимущества и недостатки каждого подхода представлены в табл. 4.

Таблица 3

Сравнение подходов к оптимизации реляционных баз данных

ПОДХОД	РУЧНАЯ ОПТИМИЗАЦИЯ	ЭКСПЕРТНАЯ СИСТЕМА	ИНТЕЛЛЕКТУАЛЬНАЯ АВТОМАТИЗАЦИЯ (ML/AI)
Гибкость	Высокий уровень (учет уникальных нюансов)	Ограниченная (строго формализованные правила)	Адаптивная, обучается на новых данных, но с ограниченной интерпретируемостью
Быстродействие	Зависит от доступности специалиста (часы–дни)	Быстрое формирование рекомендаций	Почти мгновенное принятие решений, работает в фоновом режиме
Точность	Зависит от квалификации	Зависит от полноты базы знаний	Зависит от качества обучающих данных и метрик модели
Экономичность	Высокие постоянные издержки	Разовые затраты на внедрение	Высокие инвестиции, но возможна масштабируемая окупаемость
Надежность	Риск ошибок человека	Повторяемые и стабильные рекомендации	Представляет собой «черный ящик»
Документирование	Зависит от человека	Автоматизированное логирование	Полное логирование, но слабо объяснимо «черный ящик»
Устойчивость к кадровым изменениям	Потеря знаний при ротации	Независимость от кадров	Независимость от кадров

Таблица 4

Преимущества и недостатки различных методов оптимизации СУБД

ПОДХОД	ПРЕИМУЩЕСТВА	НЕДОСТАТКИ
Ручной подход	– возможность разработки нестандартных решений для уникальных случаев; – гибкая адаптация к изменениям требований.	– значительные постоянные затраты на содержание штата специалистов; – сильная зависимость от индивидуальной квалификации персонала; – нестабильность качества выполняемых оптимизаций; – задержки в реагировании на возникающие проблемы.
ML/AI	– постоянная доступность и адаптивность; – высокая скорость реагирования; – накопление статистики и автоматическое обучение; – масштабируемость и снижение затрат при росте объема данных.	– непрозрачность принятия решений; – сложность отладки и аудита; – риск чрезмерной автоматизации без учета бизнес-контекста; – высокая стоимость внедрения и требований к инфраструктуре.
Экспертная система	– постоянная доступность; – стабильное качество формируемых рекомендаций; – систематическое накопление и сохранение экспертных знаний; – наличие процессов обязательного документирования.	– действует в рамках предустановленных правил и алгоритмов; – требует первоначальных инвестиций; – неспособна полностью заменить человеческого эксперта в сложных случаях.

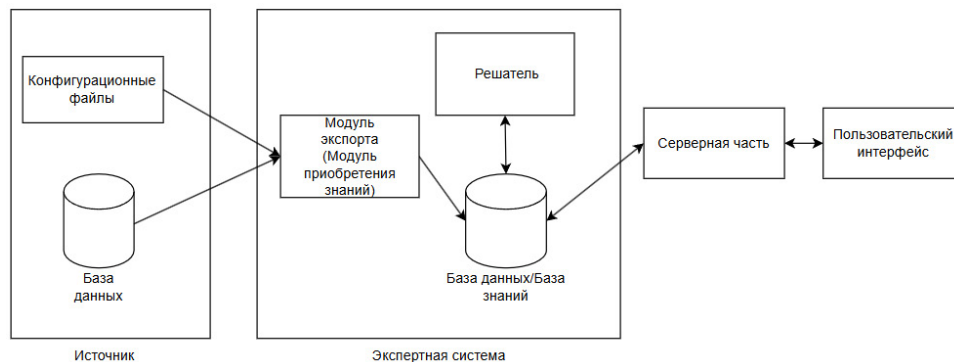
Потенциальным вариантом решения, подходящим для большинства организаций, представляется гибридная модель управления, сочетающая автоматизированную экспертную систему для обработки рутинных операций и наличие небольшого штата сотрудников для решения сложных кейсов и контроля качества.

Поэтапный переход на гибридную модель управления, где 80% рутинных операций выполняет экспертная система,

а 20% сложных кейсов остаются в компетенции администраторов, является наиболее эффективным вариантом.

5. Проектирование архитектуры экспертной системы, ориентированной на автоматизированную оптимизацию структуры реляционной базы данных

На рисунке отображена архитектура экспертной системы.



Архитектура экспертной системы

Общее описание компонентов:

1. Источник данных

Система использует внешнюю PostgreSQL-базу данных и набор конфигурационных файлов (например, .conf, .yaml, .ini) в качестве исходной информации.

2. Модуль экспорта (или модуль приобретения знаний)

Отвечает за сбор метаданных и конфигураций из внешнего источника. Полученная информация передается во внутреннюю базу системы, формируя основу для дальнейшего анализа.

3. Внутренняя база данных / база знаний

Представляет собой хранилище, содержащее: загруженные данные из внешней системы, накопленные знания о правилах оптимизации, статические правила и обновляемые рекомендации. В качестве СУБД используется PostgreSQL.

4. Решатель

Ядро экспертной системы, использующее predetermined правила или алгоритмы для анализа входных данных. Может обращаться к базе знаний и генерировать рекомендации по оптимизации структуры БД.

5. Серверная часть

Служит связующим звеном между логикой приложения и пользовательским интерфейсом. Обеспечивает взаимодействие пользовательского интерфейса с внутренней базой и решателем.

6. Пользовательский интерфейс

Обеспечивает визуализацию полученных рекомендаций и текущего состояния базы. Также пользовательский интерфейс может выступать как активный компонент: запускать процессы, подтверждать изменения или даже дополнять базу знаний.

Представленная архитектура перекликается с подходами, применяемыми в других классах интеллектуальных систем. В частности, в работе Морозова В.П., Родионова Е.А. и Сырина А.И. «Принятие решений в информационных социкиберфизических системах поддержки финансовой инвестиционной деятельности» [14] рассмотрена архитектура информационно-социкиберфизической системы, включающей интеллектуальные модули и механизмы поддержки принятия решений с минимальным участием человека.

5. Модель оценки эффективности системы

Построение модели оценки эффективности экспертной системы после внедрения требует предварительного анализа существующих научных подходов. В рамках данного исследования были рассмотрены ключевые научные публикации, охватывающие различные аспекты оценки производительности, масштабируемости и надежности СУБД.

В исследованиях Т. Тайпалуса «A Systematic Review of Database Performance Comparison Research» и М.А. Хана, О. Альматрафи и М. Хоссейна «A Systematic Literature Review on SQL vs NoSQL DBMSs» рассматриваются методологические аспекты сравнительных исследований производительности СУБД [15, 16]. Авторы подчеркивают важность

применения реалистичных сценариев нагрузки, стандартизированных подходов к тестированию и репрезентативных метрик (время отклика, пропускная способность), а также делает акцент на воспроизводимости результатов. Данный подход оправдан в контексте построения комплексной системы оценки, ориентированной на универсальные метрики.

Исследование О. Османа и В. Кноттенбелта «A Survey of Performance Evaluation of Database Systems using Queueing Models» обосновывает использование теории массового обслуживания для оценки времени ожидания, задержек и пропускной способности СУБД [17]. Такой подход позволяет формализовать поведение СУБД через математические параметры, что особенно ценно для построения количественной модели.

В публикации М. Маабреха «Performance Evaluation of Distributed Database Systems» внимание акцентируется на проблемах распределенных систем: задержках репликации, временной согласованности и устойчивости к сбоям [18]. Эти особенности необходимы для оценки отказоустойчивости и надежности в условиях масштабируемой инфраструктуры.

В исследовании Н.А. Демидовича «Модели и алгоритмы оценки эффективности баз данных» описывается разработка инструментов оценки СУБД, включая устойчивость к ошибкам конфигурации и перегрузкам [19]. Подчеркивается значимость автоматизации мониторинга узких мест, что обосновывает включение подобных критериев в состав предлагаемой модели.

На основе проведенного анализа подходы к оценке СУБД можно классифицировать следующим образом:

1. **Бенчмаркинг** – использование стандартных нагрузочных тестов (TPC-H, YCSB);
2. **Математические модели** – аналитические методы, в том числе модели массового обслуживания;
3. **Экспертные подходы** – систематизация опыта сотрудников;
4. **Гибридные методы** – комбинирование количественных измерений с экспертными оценками.

Однако несмотря на разнообразие подходов, большинство из них имеют ряд существенных ограничений: часто ориентированы на синтетические сценарии, не отражающие специфику реальных эксплуатационных условий, не учитывают когнитивную нагрузку на администраторов при принятии решений на основе рекомендаций, игнорируют экономические параметры, такие как сокращение затрат, возврат инвестиций (ROI), время окупаемости.

С учетом этих недостатков, в данной работе предлагается разработка гибридной модели, ориентированной как на количественные: производительность, нагрузка, надежность, так и на качественные аспекты: качество рекомендаций, простота внедрения.

На основе анализа исследований и практического опыта были сформированы следующие критерии, объединенные в логические группы. Каждому критерию соответствует методика оценки и применимый инструмент замера. Критерии оценки эффективности экспертной системы по оптимизации СУБД представлены в табл. 5.

Таблица 5

Критерии оценки эффективности экспертной системы по оптимизации СУБД

ГРУППА КРИТЕРИЕВ	КРИТЕРИЙ / ПОДКРИТЕРИЙ	МЕТОД ЗАМЕРА
Производительность	Время отклика	pg_stat_statements, лог-анализ, нагрузочные тесты
	Пропускная способность	pgbench, TPC-H, лог транзакций
	ЦП	pg_stat_activity, Prometheus, htop
Загрузка ресурсов	Память	free, top, метрики Prometheus
	I/O	iostat, pg_stat_io, Zabbix
Обнаружение узких мест	Время локализации причины	Сравнение временных меток инцидентов до и после внедрения системы
Надежность	Количество сбоев	Логи ошибок, отчеты Zabbix / Sentry / Prometheus
Доступность	Время простоя	Логи с таймингом, SLA-отчеты
Экономическая эффективность	Снижение затрат на диагностику	Анализ задач в Jira / YouTrack, сравнение FTE-человеко-часов
	Снижение затрат на содержание штата	Анализ расходов по HR-отчетам
	Снижение затрат на вычислительные ресурсы	Сравнение счетов на облако / внутреннюю инфраструктуру
Переносимость рекомендаций	Повторяемость и применимость	Тестирование на нескольких окружениях
Качество рекомендаций	Точность / полнота / релевантность	Precision, Recall, F1-score, экспертная валидация
Интерпретируемость	Понятность рекомендаций	Опрос сотрудников, оценка сложности внедрения
Скорость реакции	Время реакции на аномалию	Разница между временем события и первой рекомендацией
Интеграция	Простота встраивания в инфраструктуру	Анализ документации, интервью с сотрудниками, хронометраж внедрения

Для получения обобщенной метрики, позволяющей оценить влияние системы по совокупности параметров, вводится интегральный показатель E_{total} , рассчитываемый как взвешенная сумма нормализованных компонент:

$$E_{total} = w_{total} \cdot P + w_R \cdot R + w_E \cdot E + w_Q \cdot Q;$$

где:

- w – весовой коэффициент;
- P – обобщенный показатель производительности (время отклика, пропускная способность, загрузка ресурсов);
- R – надежность и доступность (число сбоев, продолжительность простоя);
- E – экономическая эффективность (снижение затрат);
- Q – качество рекомендаций (точность, полнота, интерпретируемость).

Для удобства и масштабируемости предлагается двухуровневая система, где каждая группа включает набор связанных критериев, представленная в таблице 6.

Для повышения универсальности разработанной модели предлагается внедрить адаптивную схему весов, отражающую приоритеты различных типов ИТ-сред. Значения в модели являются весовыми коэффициентами и определяются в зависимости от ключевых требований к системе: надежности, производительности, экономичности и интерпретируемости.

Таблица 6
Двухуровневая система набора связанных критериев

ГРУППА (1 УРОВЕНЬ)	КРИТЕРИЙ (2 УРОВЕНЬ)
1. Производительность (P)	Время отклика
	Пропускная способность
	Загрузка ресурсов (CPU, RAM, I/O)
2. Надежность и доступность (R)	Количество сбоев
	Время простоя
	Время локализации узкого места
3. Экономическая эффективность (E)	Затраты на диагностику
	Содержание штата
	Ресурсы инфраструктуры
4. Качество и удобство рекомендаций (Q)	Точность рекомендаций (F1-score)
	Интерпретируемость
	Переносимость / повторяемость
	Скорость реакции на аномалии
	Интеграция в инфраструктуру

Значения коэффициентов сформированы на основе анализа практических отраслевых требований и являются авторской разработкой, направленной на достижение оптимального баланса характеристик для разных сценариев применения. В табл. 7 представлена модель адаптивной схемы весов относительно конкретной ИТ-среды.

Таблица 7

Модель адаптивной схемы весов

ТИП ОРГАНИЗАЦИИ / СЦЕНАРИЙ	НАДЕЖНОСТЬ (R)	ПРОИЗВОДИТЕЛЬНОСТЬ (P)	ЭКОНОМИЧНОСТЬ (E)	ИНТЕРПРЕТИРУЕМОСТЬ (Q)
Финансовые учреждения, телеком, критичная инфраструктура	0.5	0.3	0.1	0.1
Корпоративные BI-системы, отчетность, DWH	0.4	0.4	0.1	0.1
Стартапы, облачные приложения	0.25	0.25	0.3	0.2
Госсектор, системы с высокими регуляторными требованиями	0.4	0.2	0.2	0.2
Малый бизнес, ограниченные ресурсы	0.2	0.4	0.4	0.2

Для фиксации эффекта от внедрения используются три временные контрольные точки:

- T_0 (исходное состояние) – до начала использования системы;
- T_1 (первичный результат) – через 30 дней;
- T_2 (устойчивый эффект) – через 90 дней после внедрения.

Это позволяет учесть как краткосрочные, так и долгосрочные изменения в ключевых метриках.

Оценка эффективности производится поэтапно, с последовательным усложнением условий внедрения:

- Лабораторный этап** – тестирование системы в контролируемом стенде на синтетических нагрузках.
- Пилотное внедрение в тестовой среде** – имитация реального окружения с «зеркалом» бизнес-нагрузки.
- Внедрение в продуктивной среде (частичное)** – А/В-тестирование: часть экземпляров функционирует с активной системой рекомендаций, часть – в прежнем режиме.

Для демонстрации предлагаемого подхода рассмотрен пример применения экспертной системы в контексте корпоративной BI-платформы. В соответствии с моделью, изложенной выше, все метрики приводятся к единой шкале с последующей агрегацией в интегральный показатель эффективности. В табл. 8 приведены значения ключевых метрик до нормализации, а также соответствующие им интервалы возможных значений, определенные на основании исторических данных и требований SLA.

Таблица 8

Исходные и нормативные значения критериев

КРИТЕРИЙ	ЗНАЧЕНИЕ	МИН	МАКС	НАПРАВЛЕНИЕ
Время отклика (мс)	180	100	400	Меньше лучше
Пропускная способность (транз./с)	250	100	300	Больше лучше
Загрузка CPU (%)	60	20	90	Меньше лучше
Количество сбоев в месяц	2	0	5	Меньше лучше
Время простоя (минут в месяц)	15	0	60	Меньше лучше
F1-score рекомендаций	0.83	0	1	Больше лучше
Интерпретируемость (оценка 1–5)	4.2	1	5	Больше лучше

Для обеспечения сопоставимости разнородных метрик применяется линейная нормализация к диапазону от 0 до 1. Используются следующие формулы:

– Для метрик, где большее значение предпочтительно:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}};$$

– Для метрик, где меньшее значение предпочтительно:

$$X_{norm} = \frac{X_{max} - X}{X_{max} - X_{min}}.$$

Результаты нормализации представлены в табл. 9.

Таблица 9

Нормализованные значения критериев

КРИТЕРИЙ	РАСЧЕТ	ЗНАЧЕНИЕ
Время отклика	$(400 - 180) / (400 - 100)$	0.73
Пропускная способность	$(250 - 100) / (300 - 100)$	0.75
Загрузка CPU	$(90 - 60) / (90 - 20)$	0.43
Количество сбоев	$(5 - 2) / (5 - 0)$	0.60
Время простоя	$(60 - 15) / (60 - 0)$	0.75
F1-score	уже нормализовано	0.83
Интерпретируемость	$(4.2 - 1) / (5 - 1)$	0.80

После нормализации производится группировка по четырем основным блокам:

– P (Производительность): среднее арифметическое из трех показателей: время отклика, пропускная способность и загрузка CPU. Строки 1, 2, 3 табл. 9:

$$P = \frac{0.73 + 0.75 + 0.43}{3} \approx 0.64;$$

– R (Надежность и доступность) среднее арифметическое из двух показателей: количество сбоев и время простоя. Строки 4, 5 табл. 9:

$$R = \frac{0.60 + 0.75}{2} = 0.675;$$

– E (Экономическая эффективность): в этом примере не представлены, т.к. необходимо наличие отчетности от HR-менеджеров. Принимается условное значение: $E=0.7$

– Q (Качество рекомендаций) среднее арифметическое из двух показателей: F1-score и интерпретируемость. Строки 6, 7 табл. 9:

$$Q = \frac{0.83 + 0.80}{2} = 0.815.$$

В соответствии с адаптивной моделью весов для BI-систем, применяются следующие коэффициенты:

- $w_r=0.3$;
- $w_p=0.4$;
- $w_e=0.2$;
- $w_q=0.1$.

Итоговая эффективность рассчитывается по формуле:

$$E_{total} = w_p \cdot P + w_r \cdot R + w_e \cdot E + w_q \cdot Q;$$

$$E_{total} = 0.4 \cdot 0.64 + 0.3 \cdot 0.675 + 0.2 \cdot 0.7 + 0.1 \cdot 0.815 \approx 0.678.$$

Полученное значение $E_{total} \approx 0.678$ отражает интегральную оценку эффективности внедренной экспертной системы. Такой подход позволяет количественно сравнивать различные решения, отслеживать динамику по времени, а также учитывать специфику применения в разных бизнес-сценариях за счет гибкой настройки весов.

СРАВНЕНИЯ С ДРУГИМИ ИССЛЕДОВАНИЯМИ

В рамках данного исследования была предложена концепция экспертной системы, ориентированной на автоматизированную оптимизацию структуры реляционной базы данных, в первую очередь за счет формализации экспертных знаний и логически интерпретируемых правил.

Для сопоставления полученных результатов целесообразно рассмотреть ряд существующих решений в смежных исследованиях.

В работе D. Van Aken и соавт. [20, 21] представлены системы OtterTune и CDBTune, использующие методы машинного обучения для автоматической настройки параметров PostgreSQL и MySQL. Исследование фокусируется на сборе эксплуатационных данных с рабочих СУБД и обучении модели, способной рекомендовать оптимальные значения параметров конфигурации. Полученные результаты показывают значительное улучшение производительности по сравнению с ручными настройками. Однако авторы прямо указывают, что модель работает как «черный ящик» и интерпретировать причины тех или иных рекомендаций затруднительно. Появляющиеся концепции, такие как query superoptimization [22], предлагают более глубокую автоматизацию и адаптивность, но по-прежнему сохраняют характер «черного ящика».

В отличие от этих решений концепция, предложенная в настоящем исследовании, ориентирована на прозрачность и интерпретируемость. Использование централизованной базы знаний и формализованных экспертных правил позволяет отслеживать, уточнять и адаптировать логику принятия решений без участия модели обучения и зависимости от исторических данных.

Таким образом, по сравнению с существующими подходами предложенная система не требует обучающей вы-

борки, допускает экспертную корректировку, охватывает не только параметры конфигурации, но и структуру базы, ориентирована на использование в организациях, где важна повторяемость, документация и объяснимость решений.

ВЫВОДЫ

Предложенная концепция экспертной системы стремится найти баланс между этими подходами. Использование формализованных правил и централизованной базы знаний позволяет обеспечить интерпретируемость решений, а автоматизация обработки метаданных – повысить эффективность анализа. Сходные подходы к анализу и корректировке данных применяются в системах автоматической очистки и логической реконструкции, таких как HoloClean [25]. В отличие от существующих решений разрабатываемая система акцентирует внимание на прозрачности логики принятия решений и возможности включения специфичных экспертных правил.

Таким образом, представленная архитектура является отправной точкой для создания гибкой экспертной системы, способной занять промежуточную нишу между традиционным администрированием БД и сложными интеллектуальными системами на базе ИИ.

Список использованных источников и литературы / References

1. Kimball R., Ross M. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling. 3rd ed. Wiley, 2022.
2. Trummer I. DB-BERT: a Database Tuning Toolkit that «Reads the Manual» // In Proceedings of the 2022 international conference on management of data. 2022. P. 190–203.
3. Lao J., Wang Y., Li Y., Wang J., et al. GPTuner: A Manual-Reading Database Tuning System via GPT-Guided Bayesian Optimization. arXiv preprint arXiv:2311.03157, 2023.
4. Cosa M., Torelli R. Digital Transformation and Flexible Performance Management: A Systematic Literature Review of the Evolution of Performance Measurement Systems // Global Journal of Flexible Systems Management. – 2024. – Vol. 25, No. 3. – P. 445–466.
5. DB-Engines Ranking [Электронный ресурс]. – URL: <https://db-engines.com/en/ranking> (дата обращения: 19.05.2025).
6. СУБД (рынок России) [Электронный ресурс] // TAdviser. – URL: https://www.tadviser.ru/index.php/Статья:СУБД_%28рынок_России%29 (дата обращения: 21.03.2025). / DBMS (Russian Market) [Electronic resource] // TAdviser. – URL: https://www.tadviser.ru/index.php/Статья:СУБД_%28рынок_России%29 (date of access: 21.03.2025).
7. Российские СУБД приходят на смену зарубежным [Электронный ресурс] // Российская газета. 24.04.2022. – URL: <https://rg.ru/2022/04/24/rossijskie-subd-prihodiat-na-smenu-zarubezhnym.html> (дата обращения: 03.04.2025). / Russian DBMSs Replace Foreign Ones [Electronic resource] // Rossiyskaya Gazeta. April 24, 2022. – URL: <https://rg.ru/2022/04/24/rossijskie-subd-prihodiat-na-smenu-zarubezhnym.html> (date of access: 04.03.2025).
8. The Transaction Processing Performance Council [Электронный ресурс]. TPC Benchmark™ H (TPC-H) Standard Specification. – URL: <https://www.tpc.org/tpch/> (дата обращения: 12.04.2025).
9. Gartner [Электронный ресурс]. Database Management System Market Trends. 2024. – URL: <https://www.gartner.com/en/documents/4594399> (дата обращения: 04.03.2025).
10. Pavlo A., Gopalan G., Babu S., et al. Self-driving database management systems [Электронный ресурс] // Proceedings of the 8th Biennial Conference on Innovative Data Systems Research (CIDR 2017). 2017. Vol. 1. P. 1–13. – URL: <https://www.cidrdb.org/cidr2017/papers/p42-pavlo-cidr17.pdf> (дата обращения: 27.03.2025).
11. Gunasekaran K.P., Tiwari K., Acharya R. «Utilizing Deep Learning for Automated Tuning of Database Management Systems» // arXiv preprint arXiv:2306.14349, 2023.
12. Marcus R., Negi P., Mao H., et al. Making learned query optimization practical // In Proceedings of the 2021 International Conference on Management of Data. 2021. P. 1275–1288.
13. Use index advisor [Электронный ресурс] // Microsoft Learn. – URL: <https://cloud.google.com/sql/docs/sqlserver/use-index-advisor> (дата обращения: 02.04.2025).

14. Морозов В.П., Родионов Е.А., Сырин А.И. Принятие решений в информационных социкиберфизических системах поддержки финансовой инвестиционной деятельности // Информационно-экономические аспекты стандартизации и технического регулирования. 2021. № 3 (61). С. 54–60. / Morozov V.P., Rodionov E.A., Syrin A.I. Decision Making in Socio-Cyber-Physical Information Systems for Supporting Financial Investment Activities // Information and Economic Aspects of Standardization and Technical Regulation. 2021. No. 3 (61). Pp. 54–60.
15. Taipalus T. A Systematic Review of Database Performance Comparison Research. // arXiv preprint arXiv:2301.01095, 2023.
16. Khan W., Kumar T., Zhang C., et al. SQL and NoSQL Database Software Architecture Performance Analysis and Assessments – A Systematic Literature Review // Big Data and Cognitive Computing. – 2023. – Т. 7, № 2. – С. 97.
17. Osman O., Knottenbelt W.J. Database system performance evaluation models: A survey // Performance Evaluation. – 2012. – Vol. 69, № 10. – P. 471–493.
18. Maabreh K.S. Performance Evaluation of Distributed Database Systems // International Journal of Computer and Technology. 2014. Т. 13, № 9. С. 4859–4867. – URL: <https://rajpub.com/index.php/ijct/article/view/2383> (дата обращения: 20.03.2025).
19. Демидович Н.А. Модели и алгоритмы оценки эффективности баз данных [Электронный ресурс] // Компьютерные системы и сети: материалы 52-й науч. конф. аспирантов, магистрантов и студентов, Минск, 25–30 апр. 2016 г. / Белорус. гос. ун-т информатики и радиоэлектроники. – Минск, 2016. – С. 54–55. – URL: <https://libeldoc.bsuir.by/handle/123456789/7043> (дата обращения: 04.06.2025). / Demidovich N.A. Models and Algorithms for Database Performance Evaluation [Electronic resource] // Computer Systems and Networks: Proceedings of the 52nd Scientific Conference of Postgraduate and Graduate Students, Minsk, April 25–30, 2016 / Belarusian State University of Informatics and Radioelectronics. – Minsk, 2016. – Pp. 54–55. – URL: <https://libeldoc.bsuir.by/handle/123456789/7043> (date of access: 04.06.2025).
20. Van Aken, D., Yang, D., Brillard, et al. An Inquiry into Machine Learning-based Automatic Configuration Tuning Services on Real-World Database Management Systems [Электронный ресурс]. – Carnegie Mellon University, 2021. – URL: <https://db.cs.cmu.edu/papers/2021/p1241-aken.pdf> (дата обращения: 15.05.2025).
21. Li G., Zhou K., Zhang J., et al. An End-to-End Automatic Cloud Database Tuning System Using Deep Reinforcement Learning // Proceedings of the 2019 International Conference on Management of Data (SIGMOD'19). – 2019. – С. 415–432.
22. Marcus R. Learned Query Superoptimization. // arXiv preprint arXiv:2303.15308, 2023.
23. SQL Optimizer for SQL Server [Электронный ресурс] // Quest Software. – URL: <https://www.quest.com/products/sql-optimizer-for-sql-server/> (дата обращения: 04.06.2025).
24. HypoPG: Hypothetical Indexes for PostgreSQL [Электронный ресурс]. – URL: <https://github.com/HypoPG/hypopg> (дата обращения: 05.05.2025).
25. Rekatsinas T., Chu X., Ilyas I.F., et al. HoloClean: Holistic Data Repairs with Probabilistic Inference. // arXiv preprint arXiv:1702.00820, 2017.

APPLICATION OF AN EXPERT SYSTEM FOR OPTIMIZING THE MANAGEMENT OF OBJECTS IN A RELATIONAL DATABASE

Frantasov D.N., Candidate of Technical Sciences, Associate Professor, Head of the Department for Digital Transformation Support, part-time at Samara State University of Economics, Samara

Leshin N.S., 2nd-year Master's student at Samara State University of Economics, Samara

Prokopov A.V., 2nd-year Master's student at Samara State University of Economics, Samara

The development of automated systems for optimizing relational databases is becoming a key direction for improving performance and reducing operational costs. This work proposes a concept of an expert system that generates recommendations based on the analysis of metadata and configuration parameters. A review of existing approaches is presented, including manual methods, ML/AI, and hybrid models. The system architecture and an evaluation model considering both technical and organizational factors are described. A classification of recommendation types is proposed, covering various aspects of database management system (DBMS) maintenance. The proposed approach ensures a balance between automation and expert control, enhancing the resilience and manageability of the information infrastructure.

Keywords: expert system, optimization, metadata, recommendations.

For citation: Frantasov D.N., Leshin N.S., Prokopov A.V. Application of an Expert System for Optimizing the Management of Objects in a Relational Database. Information and Economic Aspects of Standardization and Technical Regulation. 2025; 4(85): 64–75. (In Russ.).