МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**
Кафедра математической кибернетики и компьютерных наук

СОЗДАНИЕ АНАЛИТИЧЕСКОЙ СИСТЕМЫ СРЕДСТВАМИ CUBEJS НА ОСНОВЕ ХРАНИЛИЩА ДАННЫХ ДЛЯ ИНТЕРНЕТ-МАГАЗИНА

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студентки 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные технологии
факультета КНиИТ
Барбулат Марии Сергеевны

Научный руководитель
доцент, к. ф.-м. н. _____ М. И. Сафрончик

Заведующий кафедрой
к. ф.-м. н., доцент _____ С. В. Миронов

Саратов 2026

ВВЕДЕНИЕ

Актуальность темы. Цифровая трансформация экономики существенно изменила подходы к организации торговли и взаимодействию с потребителями. Электронная коммерция является одним из наиболее динамично развивающихся сегментов современной экономики. Развитие цифровых технологий и интернет-инфраструктуры способствует росту онлайн-торговли и увеличению количества цифровых транзакций.

Рост электронной коммерции сопровождается увеличением объёмов данных, формируемых в процессе работы интернет-магазинов. Действия пользователей фиксируются в информационных системах и используются для анализа и оптимизации бизнес-процессов.

Извлечение полезной информации из постоянно растущих объёмов данных для принятия управленческих решений помогает выявлять скрытые тенденции и аномалии, прогнозировать развитие ситуации, оптимизировать бизнес-процессы и снижать затраты. Для решения подобных задач используются аналитические системы на основе хранилищ данных и OLAP-технологий многомерного анализа, позволяющие консолидировать информацию из различных источников и исследовать её с разных ракурсов.

В связи с этим разработка аналитической системы средствами Cube.js на основе хранилища данных для интернет-магазина является актуальной задачей, направленной на повышение эффективности анализа данных и поддержку принятия управленческих решений.

Цель бакалаврской работы — разработка и реализация аналитической системы для интернет-магазина на основе реляционного хранилища данных и OLAP-платформы Cube.js.

Поставленная цель определила следующие задачи:

- провести анализ предметной области интернет-магазинов и изучить современные подходы к обработке и анализу данных;
- рассмотреть архитектуру аналитических систем и особенности построения хранилищ данных;
- спроектировать структуру хранилища данных на базе СУБД PostgreSQL;
- реализовать процессы загрузки, преобразования и интеграции данных с использованием Apache NiFi;

— разработать семантический слой для подготовки данных к аналитическому использованию;
— разработать аналитическое API с применением платформы Cube.js;
— реализовать пользовательский интерфейс для визуализации и анализа данных;
— провести тестирование разработанной системы и продемонстрировать её работоспособность на основе выполнения аналитических запросов и визуализации данных.

Методологические основы исследования представлены в работах У. Инмона и Р. Кимбалла в области проектирования хранилищ данных, а также в исследованиях, посвящённых вопросам интеграции данных, ETL-процессам и архитектуре аналитических систем. Техническая реализация базировалась на официальной документации используемых технологий: PostgreSQL, Apache NiFi, Cube.js, React и Docker.

Теоретическая значимость бакалаврской работы заключается в систематизации подходов к построению аналитических систем на основе хранилищ данных, анализе архитектурных решений для интеграции и обработки данных, а также в обосновании применения многомерной модели данных и семантического слоя для организации аналитической обработки информации в электронной коммерции.

Практическая значимость работы заключается в создании аналитической системы для интернет-магазина, обеспечивающей интеграцию данных из различных источников, их преобразование и загрузку в хранилище данных, а также предоставление инструментов для выполнения аналитических запросов и визуализации результатов. Разработанная система может использоваться для анализа показателей деятельности интернет-магазина и поддержки принятия управленческих решений.

Структура и объём работы. Бакалаврская работа состоит из введения, двух разделов, заключения, списка использованных источников и трёх приложений. Общий объем работы — 74 страницы, из них 64 страницы — основное содержание, включая 34 рисунка, список использованных источников из 25 наименований и цифровой носитель в качестве приложения.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Аналитические системы в электронной коммерции» посвящён анализу предметной области и теоретическим основам построения аналитических систем для интернет-магазинов.

В разделе рассмотрены особенности аналитики в электронной коммерции, обусловленные большими объёмами данных. Анализ позволяет выявлять тенденции спроса, оценивать популярность товаров, анализировать поведение пользователей и повышать эффективность бизнес-процессов с использованием KPI: количество заказов, конверсия, средний чек, показатель удержания клиентов (Retention Rate), стоимость привлечения клиента (CAC) и пожизненная ценность клиента (LTV).

В разделе проведён анализ принципов построения аналитических систем. Рассмотрена концепция хранилищ данных (DWH) — централизованного хранилища больших объёмов информации для отчётов и анализа, с основными свойствами: предметная ориентированность, интегрированность, поддержка хронологии и неизменяемость.

Описаны различия между OLTP-системами (текущие данные, нормализованная структура) и OLAP-системами (исторические данные, выборка больших объёмов), обосновано разделение транзакционной и аналитической обработки. Высокая нормализация OLTP требует множества соединений таблиц, поэтому для ускорения запросов применяется денормализация.

Сформулированы основные требования к аналитическим системам: масштабируемость, производительность, актуальность, надёжность, консистентность и безопасность данных. Система должна обеспечивать обработку растущих объёмов данных без потери скорости, минимальную задержку обработки, контроль качества и согласованность информации из разных источников, а также гибкость при изменении бизнес-требований.

Рассмотрена послойная организация: слой загрузки STG (первичное сохранение без преобразований), слой очистки и нормализации ODS, аналитический слой DWH (интеграция, историчность, выполнение запросов) и слой метаданных META (структура данных, статусы ETL, управление процессами).

Выполнен обзор архитектурных подходов: Lambda (сочетает пакетную и потоковую обработку), Kappa (единый потоковый конвейер), Data Vault (разделение на хабы, линки и сателлиты), Lakehouse (гибкость озёр данных с анали-

тикой хранилищ) и Data Mesh (децентрализованное управление данными).

Также рассмотрена тенденция перехода к облачным хранилищам (Google BigQuery, Snowflake, Amazon Redshift), обеспечивающим гибкое масштабирование и производительность, однако их использование связано с зависимостью от интернет-соединения и требованиями к безопасности. Классические хранилища остаются популярным решением благодаря централизованному хранению, согласованности данных, эффективности запросов и меньшей сложности реализации.

Проведён сравнительный анализ подходов Инмона, использующего стратегию «сверху вниз» с нормализованной структурой для высокой целостности данных, и Кимбалла, применяющего стратегию «снизу вверх» на основе многомерной модели для высокой производительности запросов. Обоснован выбор модели типа «звезда» для реализации хранилища.

Особое внимание уделено процессам ETL и ELT. ETL включает извлечение, преобразование (очистка, устранение дублирования, нормализация, приведение к единому формату) и загрузку данных. ELT отличается тем, что преобразование выполняется после загрузки в хранилище. Выбор подхода зависит от архитектуры системы и требований к производительности.

Проанализированы проблемы качества данных, приводящие к искажению аналитических показателей. Для их решения применяются подходы Data Quality Management: контроль, очистка и валидация на различных этапах обработки.

Рассмотрены принципы OLAP и многомерного анализа. Основой является многомерная модель в виде OLAP-куба с измерениями и мерами. Подход позволяет выполнять анализ по нескольким направлениям, получать срезы данных и формировать агрегированные представления. Поддержка иерархий обеспечивает анализ на разных уровнях детализации.

Рассмотрены операции OLAP-анализа: drill-down (детализация), roll-up (агрегирование), slice/dice (фильтрация по измерениям), pivot (изменение структуры представления). Их использование обеспечивает гибкую работу с данными и упрощает построение аналитических отчётов.

Рассмотрены технологические подходы к организации API-доступа: REST API (стандартизированный механизм на HTTP/JSON) и GraphQL (получение только необходимых данных, актуально для интерфейсов с большим количеством метрик). Использование API обеспечивает разделение клиентской и сер-

верной частей, упрощает сопровождение и интеграцию с внешними сервисами.

Рассмотрены методы оптимизации аналитических запросов, включая использование материализованных представлений и предагрегаций, позволяющих предварительно вычислять и сохранять результаты сложных запросов, что особенно эффективно для запросов с агрегатными функциями и операциями JOIN. Также описаны современные средства визуализации данных: BI-системы Power BI, Tableau, Grafana и Apache Superset, обеспечивающие интерактивную работу с данными, построение аналитических панелей, фильтрацию, детализацию и экспорт результатов.

Второй раздел «Проектирование и реализация аналитической системы на основе хранилища данных» посвящён практической реализации разработанной системы.

В разделе представлена общая архитектура аналитической системы, построенная по многоуровневому принципу и включающая источники данных, подсистему интеграции и подготовки данных, аналитическое хранилище, семантический слой и пользовательский интерфейс. Взаимодействие компонентов системы осуществляется посредством API, обеспечивающего передачу аналитических запросов и получение результатов обработки данных.

В качестве основного источника данных используется операционная база данных интернет-магазина, разработанная в рамках курсовой работы и предназначенная для обеспечения транзакционной обработки данных (OLTP). Структура базы данных включает основные сущности системы: пользователей, товары, категории, производителей и заказы. Дополнительно в системе поддерживается загрузка данных из файлов форматов JSON, CSV и Excel. Использование нескольких источников данных позволяет выполнять объединение и последующий анализ информации в единой аналитической среде.

Проектирование хранилища данных выполнено на основе многомерной модели типа «звезда» с использованием многоуровневой архитектуры хранения данных. В структуре хранилища выделяются четыре основных уровня: слой загрузки `STG` (Staging Layer), слой предварительной обработки `ODS` (Operational Data Store), аналитический слой `DWH` и слой метаданных `META`.

Слой `STG` используется для первичной загрузки данных из различных источников без существенных преобразований. Слой `ODS` предназначен для очистки, проверки и приведения информации к единому формату. Слой `DWH` содержит

подготовленные аналитические данные и обеспечивает выполнение запросов. Слой META хранит служебную информацию о процессах загрузки и обработки данных.

Модель данных типа «звезда» включает таблицу фактов fact_sales и связанные с ней таблицы измерений. Таблица фактов содержит данные о продажах, включая количество товаров, цену за единицу и итоговую сумму, а также внешние ключи, обеспечивающие связь с таблицами измерений.

Таблицы измерений включают:

- dim_customer — сведения о клиентах (персональные и социально-демографические характеристики);
- dim_product — информация о товарах, категориях и производителях;
- dim_date — анализ по временным интервалам (дни, месяцы, кварталы, годы);
- dim_geography — географические данные для анализа по регионам и городам;
- dim_delivery — способы доставки;
- dim_payment_method — способы оплаты;
- dim_promotion — информация об акциях и скидках.

Для реализации хранилища данных выбран стек технологий на основе PostgreSQL с расширением Citus и модулем citus_columnar, обеспечивающим колоночное хранение данных и ускорение операций выборки и агрегации. Колоночное хранение особенно эффективно для аналитических запросов, выполняющих сканирование большого количества строк с выборкой ограниченного набора столбцов. Развёртывание базы данных выполняется с использованием Docker и Docker Compose, что обеспечивает изоляцию компонентов и воспроизводимость конфигурации.

ETL-процессы обеспечивают перенос данных из операционных и внешних источников в слои STG, ODS и DWH для последующей аналитической обработки. Для реализации ETL-процессов выбран инструмент Apache NiFi, обеспечивающий визуальное проектирование потоков обработки данных и интеграцию с различными источниками информации. Apache NiFi поддерживает широкий спектр процессоров для работы с базами данных, файловыми системами и API, что позволяет гибко настраивать конвейеры обработки данных. Развёртывание Apache NiFi выполняется с использованием Docker, а настройка взаимодействия

сервисов осуществляется посредством Docker Compose.

ETL-процессы реализованы в Apache NiFi в виде трёх групп процессов, каждая из которых отвечает за определённый этап обработки данных.

Группа PG_01_STG_Load обеспечивает загрузку исходных данных в слой STG с использованием различных процессоров. Для инкрементальной загрузки из OLTP-базы данных применяется процессор QueryDatabaseTableRecord, который извлекает только новые или изменённые записи на основе отслеживания максимального значения идентификатора. Для загрузки JSON-файлов используются процессоры GetFile, SplitJson и PutDatabaseRecord: GetFile выполняет чтение файлов, SplitJson разделяет JSON-данные на отдельные записи, а PutDatabaseRecord обеспечивает их загрузку в таблицы слоя STG. Примером таких данных является файл payments.json, содержащий информацию о платежах пользователей. Для загрузки CSV- и Excel-файлов применяются процессоры GetFile, ConvertRecord и PutDatabaseRecord. В качестве источника используется файл customer_profile.csv, содержащий информацию о клиентах интернет-магазина: пол, семейное положение, наличие детей и уровень образования.

Для контроля выполнения ETL-процессов используется таблица meta.etl_run, в которой фиксируются статусы выполнения загрузок и информация о возникающих ошибках. Перед началом загрузки в таблицу добавляется запись со статусом RUNNING, после успешного завершения статус меняется на SUCCESS, а при ошибке — на FAILED с сохранением сообщения об ошибке.

Группа PG_02_STG_to_ODS выполняет преобразование данных и их перенос в слой ODS с использованием процессоров ExecuteSQL, содержащих SQL-скрипты формирования таблиц слоя ODS. На данном этапе выполняются очистка данных, приведение типов, устранение дублирования и объединение данных из различных источников. В системе формируются таблицы ods.customer (клиенты), ods.product (товары), ods.payment (платежи), ods.order_header (заказы), ods.customer_geo (география клиентов), ods.delivery_geo (география доставки) и ods.sales (продажи). В процессе преобразования используются функции split_part(), TRIM() и NULLIF(), обеспечивающие разбиение адреса на отдельные элементы и обработку пустых значений.

Группа PG_03_ODS_to_DWH обеспечивает загрузку данных в аналитический слой DWH с использованием процессоров ExecuteSQL. Формируются табли-

цы измерений dwh.dim_customer, dwh.dim_product, dwh.dim_payment_method, dwh.dim_geography, dwh.dim_delivery, dwh.dim_promotion, dwh.dim_date и таблица фактов dwh.fact_sales. Для предотвращения дублирования записей используется конструкция ON CONFLICT, обеспечивающая обновление данных при повторной загрузке.

Семантический слой реализован с использованием платформы Cube.js, обеспечивающей подключение к аналитическому хранилищу PostgreSQL/Citus, описание аналитических моделей и предоставление API для работы с данными. Cube.js выступает промежуточным слоем между аналитическим хранилищем данных и пользовательским интерфейсом, обеспечивая обработку аналитических запросов и генерацию SQL-запросов к таблицам DWH.

Аналитическая модель построена по схеме «звезда» и включает куб Sales, реализованный в Cube.js, объединяющий данные о продажах, товарах, клиентах, доставке, географии и способах оплаты. В модели реализованы метрики: revenue (выручка), ordersCount (количество заказов), avgCheck (средний чек), totalQuantity (количество проданных товаров). Измерения включают product, customer, date, geography, paymentMethod, delivery. Реализованы иерархии данных: временная (годы, кварталы, месяцы, дни), географическая (федеральные округа, регионы, города, районы) и товарная (категории товаров, товары).

Для тестирования аналитической модели и выполнения запросов используется инструмент Cube Playground, позволяющий формировать аналитические запросы без написания SQL-кода. Приведены примеры выполнения аналитических запросов: сравнение выручки по возрастным группам клиентов для категории «Посуда» за 2024–2025 годы, динамика изменения среднего чека по федеральным округам России в 2024 году, анализ количества заказов для различных категорий клиентов в 2022 году, динамика продаж производителей товаров по субботам в 2025 году.

Пользовательский интерфейс системы реализован на React и TypeScript. В системе предусмотрены три категории пользователей: пользователь (клиент интернет-магазина с доступом к личному кабинету), аналитик (сотрудник, работающий с аналитическими панелями) и администратор (управление системой и ролями). Аутентификация выполняется с использованием формы входа с проверкой логина, пароля и статуса активности учётной записи. Разграниче-

ние прав реализовано с использованием параметров `is_superuser`, `is_staff` и `is_active`.

Панель администратора предоставляет средства управления учётными записями, ролями и параметрами доступа пользователей. Интерфейс содержит сводные показатели системы и таблицу пользователей с информацией о ролях и параметрах доступа. Реализована возможность добавления новых пользователей с назначением роли и параметров доступа, а также изменение параметров доступа посредством переключения логических параметров ролей и активности пользователя. Учётная запись главного администратора защищена от изменения и блокировки для предотвращения потери доступа к управлению системой.

Интерфейс аналитика предоставляет доступ к агрегированным данным интернет-магазина и средствам бизнес-аналитики. В системе реализованы средства визуализации данных: столбчатые, линейные диаграммы, диаграммы с областями. Реализован механизм детализации данных с переходом между уровнями иерархии измерений. Например, после выбора категории «Бытовая техника» система автоматически переходит к уровню детализации «Товар». Также реализован анализ динамики показателей за различные временные периоды с отображением сводных показателей текущего и предыдущего периодов и величины изменения показателя. Предусмотрен экспорт результатов анализа в форматы Excel, PDF и PNG.

Интерфейс клиента предоставляет доступ к персональным аналитическим данным с фильтрацией запросов по идентификатору клиента. В интерфейсе отображаются основные показатели: сумма покупок, количество заказов, средний чек и количество приобретённых товаров. Также предусмотрен экспорт результатов анализа в форматы Excel, PDF и PNG. Разграничение доступа реализовано посредством фильтрации запросов по идентификатору клиента с использованием фильтра `Sales.userId`.

В результате выполнения второго раздела спроектирована и реализована аналитическая система для интернет-магазина, включающая хранилище данных на основе PostgreSQL/Citus, ETL-процессы в Apache NiFi, семантический слой в Cube.js и пользовательский интерфейс на React. Разработанная система обеспечивает полный цикл работы с данными: от их извлечения из операционных источников до визуализации результатов анализа, и может использоваться для поддержки принятия управленческих решений.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы была разработана аналитическая система для интернет-магазина на основе хранилища данных и платформы Cube.js.

В рамках работы были рассмотрены современные подходы к построению аналитических систем, особенности аналитики в электронной коммерции, а также методы организации хранилищ данных и процессов обработки информации. Спроектирована архитектура аналитической системы и разработана структура хранилища данных на базе PostgreSQL с использованием многомерной модели типа «звезда».

Для загрузки и обработки данных реализованы ETL-процессы с использованием Apache NiFi. Выполнена интеграция данных из различных источников, реализованы механизмы очистки, преобразования и подготовки данных для аналитической обработки.

На основе платформы Cube.js разработан семантический слой и аналитическое API, обеспечивающие формирование аналитических запросов и доступ к данным хранилища.

Дополнительно реализован пользовательский интерфейс системы с использованием React и TypeScript. Разработаны панель администратора, интерфейс аналитика и личный кабинет пользователя. В системе реализованы средства визуализации данных, фильтрации, детализации и анализа динамики показателей, а также экспорт результатов анализа в форматы Excel, PDF и PNG.

В результате выполнения работы была достигнута поставленная цель — разработана и реализована аналитическая система, обеспечивающая хранение, обработку, анализ и визуализацию данных интернет-магазина.