

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»**

Кафедра теоретических основ
компьютерной безопасности и
криптографии

Дифференциальное резервное копирование баз данных

АВТОРЕФЕРАТ
дипломной работы

студента 6 курса 631 группы
специальности 10.05.01 Компьютерная безопасность
факультета компьютерных наук и информационных технологий

Ухова Александра Андреевича

Научный руководитель

Старший преподаватель

_____ А. А. Лобов

19.01.2026 г.

Заведующий кафедрой

д. ф.-м. н., профессор

_____ М. Б. Абросимов

19.01.2026 г.

Саратов 2026

ВВЕДЕНИЕ

В современном мире информационных технологий объемы данных разрастаются до огромных масштабов. В процессе эксплуатации информационных систем накапливаются пользовательские данные, транзакционные записи, журналы событий, метрики и аналитика. В этих условиях наличие надежного резервного копирования становится одним из ключевых факторов обеспечения непрерывности бизнеса и устойчивости ИТ-инфраструктуры при инцидентах.

Актуальность резервного копирования усиливается ростом числа инцидентов и утечек данных. Так, в 2024 году Россия заняла второе место в мире по количеству утечек данных, обеспечив 8,5% всех зарегистрированных инцидентов.

Реальным примеры показывают, что современные атаки часто выходят за рамки утечки или шифрования данных и включают разрушение инфраструктуры и резервных копий. В качестве показательных инцидентов можно выделить атаку на «Аэрофлот» в июле 2025 года, в результате которой было уничтожено около 7000 физических и виртуальных серверов и похищено 22 ТБ данных, а также атаку на Rutube в мае 2022-ого, когда злоумышленники уничтожили более 75% баз данных и значительную часть резервной инфраструктуры. Эти случаи подтверждают необходимость не только наличия резервных копий, но и их корректной организации, изоляции и проверяемости.

Стоит еще отметить, что помимо хакерских атак существуют и иные причины повреждения информационной структуры компаний: человеческий фактор (ошибки миграций, случайное удаление данных, некорректные скрипты обновления), сбои оборудования и природными явлениями (пожары, наводнения и др.).

Дополнительную сложность формирует архитектурная тенденция последних лет – переход компаний к polyglot persistence, то есть использованию нескольких СУБД в одной организации по принципу «подходящий инструмент для подходящей задачи». В реальной инфраструктуре одновременно могут

использоваться PostgreSQL для OLTP–нагрузок, MariaDB/MySQL и решения с колоночными расширениями для аналитики, SQLite как встраиваемая БД, а также корпоративные СУБД (Oracle, SQL Server) для систем с повышенными требованиями к надежности и функциональности. Следствием этого становится практическая проблема: отсутствие унифицированного инструмента резервного копирования и восстановления в подобной рабочей среде, что приводит к необходимости сопровождать множество разрозненных средств, несовместимые интерфейсы и неодинаковые политики бэкапа и валидации.

Также стоит отметить, что существуют различные типы резервного копирования, каждый из которых имеет свои собственные плюсы и минусы:

- Полная резервная копия – сохранение всех данных полностью.
- Дифференциальная резервная копия – сохранение лишь изменений данных относительно полной копии.
- Инкрементальная резервная копия – сохранение изменений данных относительно других инкрементальных копий.

Таким образом, актуальной является задача разработки подхода, который, с одной стороны, снижает стоимость регулярного резервного копирования по сравнению с полными копиями, а с другой – упрощает восстановление и управление копиями по сравнению с инкрементными цепочками, и при этом может применяться в нескольких реляционных СУБД через единые принципы. Связь данной работы с существующими исследованиями и разработками заключается в развитии методов дифференциального резервного копирования и поиске универсальных механизмов фиксации изменений, применимых вне зависимости от конкретных СУБД и их специализированных средств журналирования или репликации.

Научная новизна работы заключается в разработке унифицированного подхода к дифференциальному резервному копированию для различных реляционных СУБД, основанного на применении временных меток как универсального механизма отслеживания изменений данных. В практическом аспекте предлагаемый подход ориентирован на упрощение эксплуатации

резервных копий в условиях ограниченных ресурсов и гетерогенной инфраструктуры, что важно для администраторов БД и DevOps-инженеров.

Цель работы: разработать и протестировать унифицированный инструмент дифференциального резервного копирования для различных реляционных систем управления базами данных, основанный на применении временных меток.

Для достижения поставленной цели необходимо решить следующие задачи:

- 1) Изучить существующие методы резервного копирования и провести анализ подходов (полное, инкрементное, дифференциальное копирование).
- 2) Выполнить сравнительный анализ выбранных реляционных СУБД с точки зрения возможностей и ограничений реализации дифференциального резервного копирования.
- 3) Разработать архитектурное решение дифференциального резервного копирования на основе временных меток для отслеживания изменений данных.
- 4) Реализовать прототип предложенного решения.
- 5) Провести тестирование и анализ производительности (время выполнения полной копии, размер копий, скорость восстановления).

Дипломная работа состоит из введения, 4 разделов, заключения, списка использованных источников и 1 приложения. Общий объем работы – 112 страниц, из них 41 страница – основное содержание, включая 11 рисунков и 7 таблиц, список использованных источников из 20 наименований.

КРАТКОЕ СОДЕРЖАНИЕ

1. Теоретические основы резервного копирования

Раздел предоставляет основной теоретический материал, для дальнейшего более глубокого ознакомления с резервным копированием

1.1 Резервное копирование

Резервное копирование представляет собой процесс создания копий данных с целью восстановления при потере или повреждении оригинала. Выполняет четыре функции: защита от аппаратных сбоев, восстановление после логических ошибок, Point-in-Time Recovery, соответствие нормативным требованиям.

1.2 Компоненты процесса резервного копирования

Процесс включает четыре компонента: источник данных, механизм копирования (логический/физический), хранилище (локальное/облачное), восстановление.

1.3 Метрики оценки качества

Качество оценивается по метрикам: RPO (максимальная потеря данных во времени), RTO (максимальное время восстановления), Storage efficiency (отношение полезных данных к общему объёму), Backup speed (объём данных в единицу времени), Recovery speed (скорость восстановления).

1.4 Дельта кодирование

Дельта кодирование сохраняет только различия между версиями, снижая объём хранения в инкрементных и дифференциальных схемах.

1.5 Факторы выбора стратегии резервного копирования

Выбор стратегии зависит от размера БД, частоты изменений, требований к RTO и бюджета на хранилище.

2. Виды резервного копирования

Раздел систематизирует три основных типа резервного копирования.

2.1 Полное резервное копирование

Полное копирование создаёт полную копию БД на момент времени в один этап восстановления. Преимущества: простота, быстрое восстановление, независимость, гарантированная консистентность. Недостатки: большой объём, длительное выполнение, высокая нагрузка. Применяется для начальных точек восстановления и долгосрочного архивирования.

2.2 Инкрементное резервное копирование

Инкрементное копирование сохраняет изменения с момента последнего копирования любого типа. Восстановление требует цепочки копий. Преимущества: минимальный объём, быстрое выполнение, низкая нагрузка. Недостатки: сложное восстановление, риск потери при повреждении звена, зависимость между копиями.

2.3 Дифференциальное резервное копирование

Дифференциальное копирование сохраняет изменения только от последней полной копии. Восстановление требует двух этапов (полная + дифференциальная). Преимущества: упрощённое восстановление, быстрее инкрементного, меньше риск. Недостатки: размер растёт со временем. Типовая стратегия: еженедельные полные + ежедневные дифференциальные копии.

2.4 Сравнительный анализ методов резервного копирования

Полное копирование характеризуется долгим временем и полным размером БД. Инкрементное имеет короткое время и минимальный размер, но требует N операций восстановления. Дифференциальное занимает среднее положение. Выбор зависит от баланса между объёмом хранилища и временем восстановления.

3. Сравнение реализаций резервного копирования в различных реляционных СУБД

Раздел анализирует механизмы резервного копирования в пяти СУБД.

3.1 SQLite

SQLite использует Online Backup API для полного копирования с порционным копированием (минимальные блокировки). Дифференциального копирования нет. Применяются файловые сnapshotы (Btrfs/ZFS), WAL mode или временные метки.

3.2 MySQL

MySQL имеет два подхода полного копирования: логическое (mysqldump, SQL-скрипты, переносимость) и физическое (Percona XtraBackup, .ibd файлы, быстрее для больших БД). Дифференциального копирования нет. Point-in-Time Recovery обеспечивается binlog.

3.3 MariaDB

MariaDB наследует MySQL методы: логическое (mariadb-dump) и физическое (mariabackup). Поддерживает инкрементальное копирование через LSN. Point-in-Time Recovery через binlog.

3.4 PostgreSQL

PostgreSQL использует base backup (pg_basebackup, физическое копирование) и pg_dump (логическое). Дифференциального копирования нет. Point-in-Time Recovery через WAL-архивирование (archive_command).

3.5 SQL Server

SQL Server имеет встроёное дифференциальное копирование (параметр DIFFERENTIAL, отслеживание через DCM – битовая карта изменённых страниц). Восстановление в два этапа (полная + дифференциальная). Также поддерживает резервное копирование транзакционных логов.

3.6 Сравнительный анализ реализаций в СУБД

Анализ показывает различия в подходах: SQL Server имеет встроёное дифференциальное копирование (DCM), MySQL/MariaDB используют LSN для инкрементного копирования, PostgreSQL использует WAL-архивирование,

SQLite требует внешних решений. Отсутствие универсального механизма обосновал необходимость унифицированного подхода.

4. Реализация инструмента для резервного копирования

Раздел описывает разработанный инструмент и результаты тестирования.

4.1 Описание программы

Разработан унифицированный инструмент дифференциального копирования на основе временных меток. Преимущества: универсальность (работает для всех СУБД), точность отслеживания, надёжность (независимость от binlog/WAL), простота, гибкость без сложной инфраструктуры.

4.2 Подготовительная работа программы

Подготовительная работа добавляет два столбца (`timestamp` – время изменения, `is_deleted` – флаг удаления) и триггеры для обновления при `INSERT/UPDATE/DELETE`.

4.3 Тестирование программы

Производится тестирование программы. Также описывается алгоритм, по которому происходит изменение данных и их фиксация с помощью создания резервных копий.

4.4 Результаты тестирования

Представлены результаты тестирования для SQLite и PostgreSQL. Инструмент работает корректно и эффективно.

ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломной работы была рассмотрена задача разработки и реализации механизма дифференциального резервного копирования реляционных баз данных, ориентированного на оптимизацию объема хранилища, сокращение времени выполнения операций и повышение гибкости стратегий восстановления данных.

Дифференциальное резервное копирование как подход к сохранению только изменений относительно последней полной копии базы данных исследовалось в контексте практического применения к различным реляционным СУБД. Были определены ключевые требования к решению: минимизация объема создаваемых копий, возможность быстрого восстановления через комбинацию полной и дифференциальной копии, контроль целостности данных на всех этапах, а также масштабируемость под разные СУБД и сценарии эксплуатации.

В рамках работы проанализированы технические аспекты дифференциального резервного копирования: механизмы отслеживания изменений (например, через журналы транзакций, битовые карты блоков или таблицы версий), способы определения модифицированных данных с момента последней полной копии, стратегии оптимизации объема дифференциальных копий и методики верификации целостности восстановленной информации.

На основе полученных знаний разработана архитектура инструмента дифференциального резервного копирования, предусматривающая модульную поддержку различных СУБД через адаптеры, единый формат конфигурации и журналирования операций. Реализованы ключевые сценарии: создание полных копий как основы, построение дифференциальных копий с отслеживанием изменений, восстановление данных на актуальное состояние (комбинация полной и дифференциальной копии), а также проверка корректности результата через сверку контрольных сумм, валидацию схемы и выполнение контрольных запросов.

Работоспособность разработанного решения подтверждена на тестовых

наборах данных различных размеров и уровней активности. Проведены серии экспериментов, включающие создание полных и дифференциальных копий, операции восстановления, измерение объема сохраняемых данных, времени выполнения и успешности восстановления. Полученные результаты демонстрируют, что дифференциальное резервное копирование обеспечивает значительное сокращение объема копий (по сравнению с полным копированием) при сохранении надежности восстановления и может быть рекомендовано для внедрения в системы администрирования данных организаций.

Таким образом, поставленные задачи исследования выполнены, а цель работы – разработка и реализация механизма дифференциального резервного копирования реляционных баз данных достигнута. В качестве перспективных направлений дальнейшего развития целесообразно рассмотреть: интеграцию инкрементного копирования для дополнительной оптимизации объема и частоты операций; автоматизацию управления политиками хранения и расписаниями резервного копирования; реализацию шифрования и управления ключами для защиты конфиденциальных данных; углубленную поддержку сценариев восстановления на момент времени (PITR) с использованием дифференциальных копий и журналов; оптимизацию обработки больших объемов данных через параллелизм и распределенные вычисления; а также интеграцию с системами мониторинга и оповещения для автоматизации контроля состояния резервных копий.