

Итоговый проект

Итоговый индивидуальный проект

Цель проекта

Создать индивидуальный проект, который объединит все изученные темы курса и закрепит навыки построения масштабируемой и надёжной системы для бизнес-приложения. Проект будет включать разработку структуры базы данных, её шардирование, настройку мониторинга и интеграцию через JDBC, обеспечивая эффективное управление и анализ данных в реальной среде.

Бизнес-контекст

В современном бизнесе, где объёмы данных постоянно растут, системы хранения данных должны быть масштабируемыми и устойчивыми к нагрузкам. Использование PostgreSQL требует продуманного подхода к распределению данных, чтобы избежать переполнения базы и обеспечить высокую производительность. Шардирование – ключевой механизм, позволяющий разделить данные на логические и физические части, упрощая масштабирование и поддерживая стабильность работы.

Мониторинг технических и бизнес-метрик позволяет своевременно обнаруживать проблемы и принимать решения для оптимизации работы системы, а интеграция через JDBC гарантирует корректный и эффективный доступ к данным из приложений.

Задачи проекта

1. База данных и шардирование

- Спроектируйте и создайте реляционную базу данных с **минимум 9 таблицами**, сгруппированными в **3 домена** (вертикальное шардирование), отражающими разные бизнес-области приложения.
- Применяйте шардирование **только к тем таблицам, которые потенциально могут значительно вырасти** (например, таблицы с транзакциями, логами или историческими данными). Не обязательно шардировать все таблицы.
- Для каждого домена реализуйте **горизонтальное шардирование**: разделите данные каждой шардируемой таблицы минимум на 2 шарда по ключу, обеспечивающему равномерное распределение нагрузки и данных. Например, это может быть шардирование по `user_id`, дате, географии или другому подходящему бизнес-ключу.
- Вертикальное шардирование помогает изолировать бизнес-области, снижает сложность запросов и повышает управляемость данных.

- Обеспечьте целостность данных и корректную маршрутизацию запросов в приложении при работе с шардами.
- В отчёте опишите структуру БД, логику вертикального и горизонтального шардирования с конкретными примерами ключей и бизнес-обоснованием.
- Постройте и приложите **ER-диаграмму** с обозначением доменов и шардинга.
- Приложите скрипты для создания и наполнения базы тестовыми данными.

2. Мониторинг и метрики

- Настройте мониторинг с помощью Grafana, Prometheus, Alertmanager, Node Exporter, Postgres Exporter.
- Создайте **2 дашборда в Grafana**:
 - **Технический дашборд** – отображение метрик состояния системы (память, CPU, дисковое пространство и прочее).
 - **Бизнес-дашборд** – минимум 3 бизнес-метрики, отражающих ключевые показатели работы вашего приложения.
- Настройте Alertmanager для отправки оповещений в Telegram при критических событиях.

3. JDBC и интеграция

- Реализуйте подключение к одной из шардированных баз через JDBC в Java-приложении.
- Напишите пример запроса и обработки результата из шардированной БД.

4. Отчёт

Для подтверждения проделанной работы оформите отчёт, включив следующие элементы:

- **Описание архитектуры и компонентов проекта**
 - Включите схему архитектуры с указанием используемых сервисов и контейнеров (PostgreSQL, Prometheus, Grafana, JDBC-приложение).
 - Опишите логику шардирования и мониторинга, приведите примеры конфигураций.
- **Документация Docker-среды**
 - Приложите Dockerfile(ы) и docker-compose.yml с подробными комментариями.
 - Укажите команды для запуска и остановки контейнеров.
 - Опишите, как воспроизвести рабочую среду из исходных файлов.
- **Логирование и цифровой след**
 - Включите логи запуска контейнеров и приложений (например, сохранённый вывод `docker logs`).
 - Сделайте скриншоты или экспорт дашбордов Grafana с отображением метрик и alert-уведомлений.
 - Приложите примеры запросов JDBC и результаты их выполнения (логи или выводы).

- **Использование системы контроля версий**

- Рекомендуется хранить проект в Git-репозитории.
- Укажите ссылку на репозиторий с историей коммитов как доказательство работы.

- **Автоматизация и тесты (если есть)**

- Опишите и приложите автоматические тесты или скрипты для наполнения БД, а также результаты их выполнения.

- **Выводы**

- Кратко опишите сложности, с которыми столкнулись, и методы их решения.
- Сделайте выводы о работе шардирования и мониторинга, оцените эффективность и перспективы улучшения.

Рекомендации

- Планируйте структуру и шардирование исходя из бизнес-логики и объёмов данных.
 - Шардирование применяйте только там, где данные действительно вырастут и могут создать нагрузку.
 - Выбирайте ключи для горизонтального шардирования, обеспечивающие равномерное распределение нагрузки (например, user_id, дата, география).
 - Разделяйте технический и бизнес-аспекты мониторинга.
 - Обеспечьте корректную маршрутизацию запросов и сохранение целостности данных.
 - Используйте материалы курса.
 - Активно пользуйтесь системой контроля версий для фиксации изменений.
 - Соблюдайте академическую честность.
-