

Министерство образования Республики Беларусь
Учреждение образования
Белорусский государственный университет информатики и
радиоэлектроники
Кафедра инженерной психологии и эргономики

На правах рукописи

УДК 004.65

Каралкин
Виктор Павлович

КОНТРОЛЬ ВЕРСИЙ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ

Автореферат диссертации на соискание
академической степени магистра технических
наук

1 - 23 80 08 Психология труда, инженерная психология, эргономика

Магистрант В. П. Каралкин

Научный руководитель
Л. Т. Ткачева, кандидат
технических наук, доцент

Заведующий кафедрой ИПиЭ
К. Д. Яшин, кандидат
технических наук, доцент

Нормоконтролер
Т. В. Гордейчук,
ассистент кафедры ИПиЭ,
магистр технических наук

Минск 2016

КРАТКОЕ ВВЕДЕНИЕ

В настоящий момент реляционные базы данных широко распространены в различных сферах промышленной разработки ПО – в различного рода клиент-серверных системах, в частности – в разработке web-приложений. Несмотря на набирающие популярность и развивающиеся no-SQL решения, такие как MongoDB, Redis, CouchDB, в большинстве случаев «ядром» механизма хранения и обработки данных являются реляционные решения.

Одной из наиболее часто используемых СУБД является MySQL. Её несомненными плюсами является свободная лицензия GPL, довольно высокая производительность, кроссплатформенность а также наличие драйверов практически под все популярные среды программирования.

Многие web-системы и web-приложения используют БД не только для хранения пользовательских данных, таких как контент web-сайта, регистрационные данные, торговые транзакции, но и специфические настройки web-приложений, а порой даже часть интерпретируемого программного кода, например, на языке PHP.

Данная магистерская диссертация является исследованием системы, призванной усовершенствовать процесс разработки ПО с использованием реляционных юаз данных, предоставив разработчикам версионировать БД наподобии аналогичной операции с исходным кодом. В основу проекта заложены базовые концепции популярной системы управления версиями Git. Соответственно, проект является распределённой системой контроля версий для реляционной БД.

Система предлагает разработчику удерживать под версионным контролем как структуру БД – схемы таблиц, структуру индексов, триггеры, хранимые процедуры, так и сами данные определённых таблиц. Последнее актуально для случаев, когда в БД хранятся некоторые настройки web-приложения.

Данный подход позволяет управлять БД таким же образом, как это можно делать с исходным кодом при помощи Git. Разработчик может в любой момент переключить web-приложение в любое необходимое ему состояние, переключив в нужную версию исходный код и базу данных.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Объектом исследования диссертации являются реляционные базы данных. Предмет исследования – методы и подходы к контролю версий реляционных баз данных.

Задачи исследования были поставлены следующие:

1. Изучить научную литературу по темам «реляционные базы данных» и «системы контроля версий».
2. Провести теоретическое исследование по проблеме контроля версий реляционных баз данных.
3. Разработать и испытать программное средство для мониторинга процесса продвижения веб-сайтов.

Целью работы является изучение способов контроля версий реляционных баз данных и разработка программного средства, реализующего изученные способы.

Проблема и способы контроля версий реляционных баз данных были изложены в докладе к 51-ой научной конференции аспирантов, магистрантов и студентов.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

В первой главе рассматривается литература по теме магистерской диссертации, в частности, по реляционным базам данных, СУБД MySQL и системе управления версиями Git.

Современные СУБД обладают богатым набором возможностей в сферах хранения и обработки данных. Существуют базовые алгоритмы и операции с данными в реляционной модели БД, используя которые, можно оперировать потоками данных целевого web-приложения, а также расширять эти механизмы, реализуя произвольную функциональность. В книге [1] описаны базовые возможности MySQL и приведено множество примеров работы с потоками данных.

Работе сложных, распределённых, масштабируемых приложений сопутствует непростой процесс развёртывания используемых им ресурсов БД. Высоконагруженные бизнес-системы используют такие возможности СУБД MySQL, как репликация и кластеризация. Важным вопросом в контексте сохранности данных и производительности является процесс резервного копирования баз данных и их реплик. В данной магистерской диссертации эти механизмы учитываются согласованно с драйвером MySQL, используемым ядром разработанной системы контроля версий БД.

Оптимизация запросов, правильные методологии построения индексов БД наряду с вышеназванными механизмами репликации и кластеризации являются неотъемлемой частью процесса проектирования и построения высокоэффективной и производительной БД.

С другой стороны, разрабатывая данный продукт, я анализировал популярные методологии разработки ПО в контексте целевых пользователей систем контроля версий. Существуют различные практики работы с такими системами, они связаны с соответствующими им типами рабочих процессов. Одним из популярных семейств видов рабочего процесса является гибкая методология разработки – серия подходов к разработке программного обеспечения, ориентированных на использование итеративной разработки, динамическое формирование требований и обеспечение их реализации в результате постоянного взаимодействия внутри самоорганизующихся рабочих групп, состоящих из специалистов различного профиля. Большинство концептуальных особенностей разработанная система заимствует от распределённой системы контроля версий Git.

Во второй главе диссертации описано структурное и функциональное проектирование. Программное средство для контроля версий реляционных баз данных является кроссплатформенным консольным приложением с возможностью разработки подключения внешнего графического пользовательского интерфейса в будущем.

Программа написана на языке программирования C++ на базе кроссплатформенного фреймворка Qt. Среда выполнения программы – любая ОС, поддерживаемая кроссплатформенным свободным компилятором GCC. Язык программирования C++ является объектно-ориентированным языком, в частности, фреймворк Qt поощряет использование парадигмы ООП.

Базовые модули Qt предоставляют полезные возможности, которые заметно облегчают программирование за счёт использования стандартизированных алгоритмов, таких как коллекции, структуры данных. Также облегчена работа с файловой системой, а также взаимодействие с системными вызовами IPC для манипуляции дочерними процессами и вызова внешних программ.

В программном средстве, разработанном в рамках данной магистерской диссертации наиболее активно используются компоненты QtCore, QtSql и QtXml. QtCore предоставляет ряд полезных классов для работы с параметризуемыми коллекциями, структурами данных, обработкой и валидацией аргументов командной строки. QtSql отвечает за подключение к серверу баз данных, и осуществление операций с ней. QtXML используется для разбора конфигурационных файлов. Т.к. программное средство обладает консольным (текстовым) пользовательским интерфейсом, из правил сборки проекта были принудительно исключены компоненты QtGui и QtGuiTools в целях исключения ненужных зависимостей и уменьшения физического размера исполняемых файлов.

В третьей главе описана разработка текстового пользовательского интерфейса и программа испытаний разработанного программного средства.

Тестирование программного обеспечения является важной частью жизненного цикла программных продуктов. Задачами современного тестирования является не только обнаружение ошибок в программах, но и выявление причин их возникновения. Такой подход позволяет разработчикам функционировать максимально эффективно, быстро устраняя возникающие ошибки. Понимание важности процесса тестирования приводит к возникновению тенденций, направленных на применение промышленных способов проверки качества программного обеспечения. Наиболее важным направлением здесь является внедрение различных систем

автоматизированного тестирования.

Для модульного тестирования во фреймворке Qt предоставлен модуль QTestLib, которая отвечает за выполнение и проверку тестов до начала построения исходного файла проекта. Методология модульного тестирования предполагает реализацию тестов до начала процесса разработки.

Результатом программы испытаний является покрытие программного средства блочными тестами. Кроме того, было проведено интеграционное тестирование.

Также составлено руководство пользователя, которое одновременно также является программой для use-case-тестирования.

Библиотека БГУМР

ЗАКЛЮЧЕНИЕ

В процессе работы над магистерской диссертацией были рассмотрены основные принципы построения приложений на базе фреймворка Qt, а также концепции функционирования системы управления версиями Git. В результате было создано полноценное программное средство, выполняющее свои непосредственные функции и имеющее определённые достоинства. Благодаря данной разработке, были сделаны попытки внести в процесс разработки ПО, связанный с активным использованием баз данных, некоторые улучшения, сделать этот процесс более гибким и прозрачным.

Программный модуль имеет следующие достоинства:

- кроссплатформенность, что позволяет использовать разработанное программное средство на различных операционных системах;
- отсутствие посторонних зависимостей, возможность простой установки программного средства используя пакетный менеджер;
- интуитивно понятный текстовый пользовательский интерфейс, очень схожий с его аналогом в классическом Git;
- сравнительно быстрое выполнение операций благодаря тому, что большинство операций с рабочей копией носят локальный характер;
- следование концепциям системы управления версиями Git, что существенно облегчает понимание принципов работы программного средства среди программистов, владеющих навыками работы с Git.

В качестве недостатков программного средства можно отметить следующие пункты:

- отсутствие интерактивного режима работы с некоторыми командами, аналоги которых поддерживают такой режим работы в классическом Git;
- необходимость пошагового применения ряда патчей в случае переключения между версиями БД вместо применения одного патча.

Первый недостаток решается путем использования библиотеки nCurses, которая позволяет организовать интерактивный консольный пользовательский интерфейс. Вторую проблему можно решить, реализовав механизм слияния нескольких SQL-скриптов и устранения взаимоуничтожающих операций, например, создание таблицы с последующим её удалением.

СПИСОК ОПУБЛИКОВАННЫХ РАБОТ

[30-А] Каралкин В. Программное средство для контроля версий реляционных баз данных / В. Каралкин // Контроль версий реляционных баз данных: Тезисы докл. к конф. – Минск, 2015.

Библиотека БГУИР