

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 004.65

ПРИНЦИПЫ И ВАРИАНТЫ МИГРАЦИИ СПРАВОЧНЫХ БАЗ ДАННЫХ ИЗ MICROSOFT ACCESS В КЛИЕНТ-СЕРВЕРНУЮ СУБД PostgreSQL

К. Д. АВРУСЕВИЧ

(Представлено: канд. техн. наук, доц. В. М. ЧЕРТКОВ)

В статье анализируются принципы, архитектурные различия и проблемы при переносе справочников с файловой СУБД Microsoft Access на промышленную платформу PostgreSQL. Рассмотрены варианты миграции схемы, трансформации типов данных и вопросы обеспечения целостности, необходимые для построения масштабируемой корпоративной информационной системы.

Введение. В условиях необходимости цифровой трансформации критически важным является переход от устаревших локальных информационных систем (ИС) к надежным и масштабируемым клиент-серверным решениям. Многие предприятия, особенно в части ведения специализированных справочников или первичного учета, исторически использовали Microsoft Access. Однако данная файловая СУБД не способна обеспечить необходимую транзакционную целостность, многопользовательский доступ и производительность, которые требуются для обработки больших объемов информации.

Построение серверной части программного комплекса требует миграции критически важных данных в промышленную СУБД, такую как PostgreSQL. PostgreSQL обеспечивает централизованное управление данными и высокую надежность, что является необходимым условием для автоматизации ключевых бизнес-процессов.

Основная часть. Принципиальное различие между исходной (Access) и целевой (PostgreSQL) архитектурами лежит в модели доступа к данным. Access функционирует как файловая система, где данные и логика (VBA, макросы) хранятся совместно, что ведет к высокому риску повреждения файла при одновременном доступе и снижению надежности. Напротив, PostgreSQL использует клиент-серверную модель, в которой СУБД централизованно управляет транзакциями (ACID), безопасностью и обеспечением целостности данных.

Успешная миграция данных требует не простого копирования строк, а глубокой трансформации архитектурных принципов, включая перенос функционала управления целостностью с уровня приложения (Access) на уровень серверного ядра (PostgreSQL).

Трансформация типов данных и целостности. Одной из основных технических проблем является трансформация типов данных. Специфические типы Access должны быть тщательно сопоставлены с типами SQL. Например, Access использует AutoNumber, который необходимо преобразовать в PostgreSQL SERIAL или IDENTITY. Критически важно при этом корректно учесть текущее значение счетчика для обеспечения непрерывности нумерации.

Дополнительные сложности возникают с нестандартизированными для SQL типами, такими как OLE (Object Linking and Embedding) Object или Hyperlink, которые часто использовались в Access. Принципом миграции должно стать либо хранение самого объекта в виде byte, либо стандартизация хранения путей к внешним файлам в виде text. Некорректная трансформация типов данных, например, Access Date/Time в PostgreSQL Timestamp without time zone, может привести к потере информации о часовом поясе. Это влечет за собой риск нарушения надежности справочных данных, что недопустимо для точного планирования ремонтов и учета, требуемого в диссертационном исследовании.

Перенос логики и ограничений. В Access ограничения целостности (например, Foreign Keys) часто реализовывались на уровне интерфейса или с помощью кода VBA. В промышленной СУБД такая частичная реализация недопустима. Поэтому основной принцип миграции требует переноса всей бизнес-логики, касающейся валидации и взаимосвязи справочников, в хранимые процедуры и функции PostgreSQL или триггеры. Это обеспечивает централизованную целостность и надежность данных, критически важную для последующего автоматизированного управления активами.

Варианты и технические проблемы переноса. Существует два основных подхода к миграции, каждый из которых имеет свои преимущества и недостатки применительно к критически важным справочным данным.

Вариант A: Использование внешних ETL-инструментов. Инструменты для извлечения, преобразования и загрузки (Extract, Transform, Load) данных автоматизируют процесс экспортации схемы

(DDL – Data Definition Language) и содержимого. Однако они часто не справляются с конвертацией сложных типов данных, а также игнорируют встроенную логику Access. Кроме того, возникают проблемы с кодировками, особенно при работе с кириллицей (переход от Windows-1251 к UTF-8), что может привести к искажению текстовых справочных полей.

Вариант Б: Скриптовый подход (Data Definition Language/ Data Manipulation Language). Этот принцип предполагает ручное создание DDL-скриптов в PostgreSQL и импорт данных через текстовые форматы, такие как CSV. Несмотря на высокую трудоемкость, данный подход обеспечивает максимальный контроль над схемой, индексами, ограничениями целостности и, что наиболее важно, над кодировкой. Для критически важных справочников, являющихся основой программного комплекса, скриптовый метод считается наиболее безопасным.

Неправильное управление кодировками при переходе с Access на PostgreSQL является частой технической проблемой, приводящей к некорректному отображению названий деталей или типов вагонов. Обязательная конвертация на этапе экспорта данных является принципиальным требованием. Кроме того, необходимо пересмотреть подход к хранению ссылок на внешние файлы или OLE-объекты, заменив их стандартизованными путями или непосредственным хранением бинарных данных.

Заключение. Успешный перенос справочных данных в PostgreSQL требует комплексного трансформационного подхода, включающего не только экспорт данных, но и перенос механизмов обеспечения целостности и логики на серверный уровень. Скриптовый метод в сочетании с тщательным преобразованием типов данных и контролем кодировок является оптимальным принципом для миграции критически важных справочников, которые могут стать надежной основой для серверной части программного комплекса. Этот подход обеспечивает необходимую надежность и масштабируемость, подтверждая техническую готовность проекта к обработке больших объемов информации.

ЛИТЕРАТУРА

1. Моргунов, Е. П. PostgreSQL. Основы языка SQL : учеб. пособие / Е. П. Моргунов ; под ред. Е. Рогова, П. В. Лузанова. – СПб. : БХВ-Петербург, 2018. – 336 с.
2. Шёниг, Х.-Ю. PostgreSQL 13. Мастерство разработки / Ханс-Юрген Шёниг. – [М.] : [Издательство], [Год]. – 450 с.