



ОРИГИНАЛЬНАЯ СТАТЬЯ

DOI: 10.21045/1811-0185-2023-S-85-92

УДК: 61.007

ТЕХНОЛОГИЯ ИМПОРТА ДАННЫХ В МИС «ИНТЕРИН PROMIS ALPHA PG»

Д.В. Бельшев

ФГБУН «Институт программных систем им. А.К. Айламазяна»
Российской академии наук, г. Переславль-Залесский, Россия.
<https://orcid.org/0000-0002-0437-4814>

✉ Автор для корреспонденции: Бельшев Д.В.

АННОТАЦИЯ

Одним из ключевых этапов запуска в эксплуатацию информационной системы является миграция унаследованных данных, существует немало средств для решения этой задачи применительно к особенностям программных продуктов и проектов. В статье рассматриваются методология и средства миграции данных при реализации проектов с применением медицинской информационной системы Interin PROMIS Alpha PG.

Ключевые слова: миграция данных, импортозамещение, PostgreSQL, медицинские информационные системы.

Для цитирования: Бельшев Д.В. Технология импорта данных в МИС «Интерин PROMIS Alpha PG». Менеджер здравоохранения. 2023; S:85–92. DOI: 10.21045/1811-0185-2023-S-85-92

Введение

Для любой организации накопленные в процессе ее работы данные, безусловно, имеют ценность, тем большую ценность они приобретают, если эти данные относятся к здоровью людей, о чем, в частности, свидетельствует Приказ Министерства здравоохранения РФ от 3 августа 2023 г. № 408 «Об утверждении Перечня документов, образующихся в деятельности Министерства здравоохранения Российской Федерации и подведомственных ему организаций, с указанием сроков хранения», где указывается, что «сроки хранения документов не зависят от вида носителя и ограничения доступа к ним. Исключение составляет медицинская документация пациентов, срок хранения которой на бумажном носителе составляет 25 лет, а в случае ведения ее в МИС МО – 50 лет».

Совершенно естественно, что медицинские организации в силу разных обстоятельств за 50 лет накопления и хранения данных конкретной медицинской карты, неоднократно будут решать задачу модернизации своей ИТ-инфраструктуры, в том числе с обновлением или заменой информационных систем. В настоящее время одним из драйверов, стимулирующих данный процесс, является импортозамещение западных технологических решений. Также очевидно, что для исполнения вышеупомянутого Приказа МЗ, да и просто из необходимости обеспечения непрерывности диагностики и лечения пациентов в медицинской организации,

требуется обеспечить корректный перенос данных, накопленных в процессе модернизации информационных систем. Вряд ли можно предположить, что хотя бы какие-то проекты модернизации могут обойтись без этапа переноса унаследованных данных, поэтому все разработчики имеют собственные методы решения данной задачи. Мы рассмотрим соответствующие инструменты и методические приемы на примере медицинской информационной системы Интерин PROMIS Alpha PG [1].

Особенности миграции данных в проекте внедрения МИС

Модернизация, а тем более замена информационной системы в работающей медицинской организации – это достаточно сложный проект, сопряженный с множеством рисков, главный из которых – полная или частичная недоступность компонентов информационной системы в процессе перехода или после его завершения. Серьезность данного риска, как и методы его купирования существенно зависят от условий работы информационной системы, в частности:

- размера и типа медицинской организации;
- наличия распределенных подразделений;
- количества и глубины интеграционных обменов со смежными системами;
- глубины проработки и специфики функциональных возможностей заменяемой или модернизируемой ИС;

© Бельшев Д.В., 2023 г.





- глубины внедрения АРМ на рабочих местах пользователей;
- степени связности информационных процессов внутри организации.



Самым очевидным с точки зрения миграции данных способом является «в ночь на выходные остановить работу, загрузить все данные из старой системы в новую, и с понедельника работать по-новому». Это вполне рабочий вариант, если организация небольшая, процессы детально разобраны, софт типовой, смежники готовы, пользователи обучены, ресурсов для поддержки достаточно, данных не очень много. Но может быть ситуация, когда всё сложнее, и возможности подобного одномоментного перехода в силу объективных или субъективных причин нет, тогда необходима стратегия постепенного перехода из состояния «AS-IS» в «TO-BE», причем этот период может быть растянут на достаточно длительный срок с возвратами назад в силу выяснения важных нюансов, которые упустили на предыдущих этапах.

Таким образом, если всё-таки приходится работать в режиме постепенного перехода, то ключевым фактором является анализ взаимозависимостей процессов, чтобы обеспечить консистентность данных в параллельно работающих старой и новой системах. Нередко встречается ограничение, при котором команда, реализующая проект по миграции на новую систему, не имеет возможности вносить изменения в существующую систему или даже слабо представляет ее внутреннее устройство и особенности рабочих процессов. Самая банальная причина: замещаемое ПО разработано производителем, который уже недоступен, поддержкой не занимается или не хочет участвовать в процессе миграции. Но даже если возможность корректировки старой информационной системы есть, то она обычно сильно ограничена, чтобы не нарушить уже работающие процессы, особенно в условиях, что никакого резона дальнейшего развития уходящей МИС нет. Описанные ограничения диктуют необходимость организации поддержки трансформации моделей данных и восстановления процессной связности на стороне принимающей системы, согласуя технические решения с организационными мероприятиями, чтобы непрерывность деятельности организации не была нарушена.

Приведем для иллюстрации поэтапной миграции данных два чувствительных процесса, обладающих достаточно сильной связностью: круглосуточный стационар и больничная аптека.

Поэтапная миграция данных в круглосуточном стационаре

Лечение пациента в стационаре – это непрерывный ветвящийся процесс, в котором одни события порождают другие, и для регистрации новых данных требуется наличие предыдущих. Ключевой сущностью в стационаре является электронная история болезни, к ней уже привязываются документы, назначения, услуги, результаты лабораторных и диагностических исследований и т.п. Если стоит задача поддержки непрерывной работы с госпитализированными пациентами, то необходимо обеспечить синхронизацию в двух информационных системах данных о движении пациента по стационару в объеме поступления, переводов и выписки. Однократная загрузка движения растянутую во времени задачу перехода не решает, поэтому все изменения в старой системе должны непрерывно зеркалироваться в новой ИС. Вторая чувствительная задача в такой параллельной работе – это медицинская и финансовая отчетность, которая на разных этапах процесса может вестись

- а) только в старой системе с перегрузкой в новую;
- б) параллельно в двух системах для контроля корректности данных;
- в) только в новой системе после выверки и полноценной миграции учетных данных.

Фактически, замещение стационара при плавном переходе идет с конца:

1. Выполняется первоначальная загрузка данных, в первую очередь, справочников и накопленной информации за чувствительные периоды (например, несколько последних лет, если накопленный объем не позволяет выполнить всю загрузку в отведенное время, то загрузку архивов можно растянуть на более длительный срок).

2. Формируются статистические отчеты, целью которых является верификация результатов загрузки данных и самих алгоритмов расчета.

3. Запускается фоновый процесс догрузки новых данных из действующей ИС.

4. В новой ИС в реальном времени начинает отражаться движение пациентов по стационару от поступления до выписки.

5. Запускается фоновая загрузка документооборота, результатов диагностических и лабораторных исследований из старой системы и через донстройку интеграционных процессов с внешними ИС.

6. В новой системе группами (отделениями, корпусами, филиалами) начинают работать врачи-клиницисты, постепенно в нее переводятся врачи



диагностических служб, тем самым формирование электронной ИБ начинается в новой ИС.

7. В новую систему переходит экономика и статистика сначала через анализ импортированных из старой ИС данных, а затем осуществляется поэтапный перевод внесения данных в новую ИС.

8. После переноса всех интеграционных процессов со смежными системами, завязанными на работу с пациентами, на новую ИС, происходит отключение Приемного отделения и прекращается регистрация движения в старой системе.

9. Осуществляется перенос всех оставшихся архивов и выверка данных по отчетам из двух систем.

Поэтапная миграция данных в больничной аптеке

Процесс материального учета в круглосуточном стационаре в случае его достаточной автоматизации является сильно связанным процессом, в котором задействованы различные клинические и вспомогательные подразделения, а также могут быть включены смежные информационные системы, при этом требуется непрерывность движения товарно-материальных ценностей (ТМЦ), начиная от поставщиков, через распределение по центрам материального учета, списания на пациентов и итоговой отчетности с синхронизацией данных с бухгалтерией.

В различных медицинских организациях могут применяться свои учетные политики и способ перехода должен, в первую очередь, опираться на них, тем не менее описываемая ниже схема является тоже достаточно частой. Наибольшая трудоемкость перехода лежит в процессе персонифицированного списания ТМЦ на пациента, поскольку в этой работе задействовано большое количество материально ответственных лиц – это в первую очередь старшие медицинские сестры отделений, процедурные, операционные медсестры. В условиях, когда возможность массового обучения, запуска и поддержки большого количества посменно работающих сотрудников ограничена, деление на этапы происходит через отсечение самых ресурсоемких частей процесса и может выглядеть следующим образом:

1. Поступление ТМЦ оформляется в старой системе и в ней же выполняется отпуск по требованиям в центры материального учета (на склады отделений).

2. Приходы в центры материального учета синхронизируются и поступление ТМЦ, отправленное на соответствующий склад из центральной аптеки старой ИС, появляется в соответствующем складе новой ИС.

3. Списание производится автономно в двух системах, но на каждом конкретном складе только в одной, чтобы избежать конфликтов по остаткам;

4. Итоговый акт списания по пациенту формируется из данных двух систем либо через повторение соответствующих проводок в новой ИС, либо простым объединением на уровне отчетов, получаемых из двух информационных систем.

5. Выгрузка данных в бухгалтерскую систему (если такая имеется в проекте) выполняется из двух систем, каждая отчитывается по своим складам и по своему расходу, в итоге консолидированная отчетность собирается в полном объеме.

6. Опытный участок, в котором происходит работа в новой системе, расширяется по мере освоения сотрудниками функционала новой информационной системы и решения возникающих в процессе работы нюансов, что постепенно выводит старую ИС из процесса расходования ТМЦ.

7. После завершения перехода центров материального учета в новую систему завершается перевод центральной аптеки и отключается старая ИС в части материального учета.

Итоги поэтапной миграции

В текущей главе мы обратили внимание на некоторые организационные особенности миграционных процессов, которых несомненно больше, и для каждого проекта они являются индивидуальными. Вместе с тем нельзя не обратить внимание на общие черты в организации миграционных процессов в сложных проектах, в частности:

1. Поэтапная миграция, несмотря на дополнительные технические издержки (по сравнению с однократной загрузкой), обладает более высокой толерантностью к рискам и позволяет повысить устойчивость к возникающим в процессе реализации проекта непредвиденным обстоятельствам, которые обязательно возникают, как бы тщательно ни была проведена аналитика и тестирование ПО.

2. Возможность распределить ввод в эксплуатацию новой системы во времени предоставляет лимитировать нагрузку на проектную команду, особенно если проект является территориально распределенным, а количество пользователей велико.

3. Параллельная работа двух систем позволяет на реальных данных провести апробацию и настройку интеграционных процессов со сторонними системами, снижая риски сбоев при обмене данными, особенно когда имеющееся в распоряжении





описание протоколов взаимодействия не отражает всех особенностей фактической работы.

4. Синхронизация данных в реальном времени позволяет в более щадящем режиме отладить работу статистических и аналитических подразделений за счет параллельной работы с данными в двух системах, что лучше верифицирует данные и уточняет алгоритмы расчетов. И то, и другое крайне важно, особенно если модели данных в двух системах различны, и требуется корректный перевод сущностей из одной модели в другую без потери и искажения смысла реквизитов.

Технические решения поддержки миграции данных

Описанные выше организационные особенности процесса миграции между системами предъявляют ряд требований к инструментальным средствам, обеспечивающих перенос данных:

1. Трансляция моделей данных из передающей в получающую систему.
2. Эффективный транспорт (выгрузка, передача, загрузка) больших объемов данных.
3. Обеспечение многократной аддитивной загрузки данных для отслеживания изменений в процессе параллельной эксплуатации систем.

Существует масса способов обеспечить импорт данных в базу данных любой информационной системы, есть промышленные системы, обеспечивающие данный процесс, каждый производитель ИС так или иначе подобные инструменты предлагает, учитывая свои особенности. В случае с МИС Интерин PROMIS Alpha PG такие механизмы тоже есть, в их основе лежит технология JSON-хранилищ данных [2]– [4], обладающих следующими свойствами:

- хранилища размещают данные в реляционных таблицах с объектным расширением, позволяющим формировать произвольную JSON-структуру;
- манипуляции с данными происходят как в объектной, так и реляционной парадигмах: разработчик может манипулировать данными в хранилищах как с объектами, не опускаясь до уровня базовых операций с реляционными таблицами, но ему доступна и полноценная работа с теми же данными через обычный SQL.

Ввиду системной поддержки JSON в СУБД PostgreSQL, манипуляция с данными как с JSON-объектами становится эффективным средством работы с данными сложной структуры, а способность

хранилищ вести обработку объектов целиком и самостоятельно распределять их по реляционным сущностям, снимает с разработчика необходимость проведения рутинных манипуляций. Эти особенности платформы используются при реализации задач миграции данных в ИС. В качестве примера рассмотрим задачу модернизации МИС Интерин PROMIS для Oracle до МИС Интерин PROMIS Alpha PG. Обе системы реализованы одной компанией-разработчиком, но в разных технологиях и с достаточно большим временным разбросом, что в итоге привело к существенному изменению подходов к представлению данных [5].

Трансляция моделей данных

Выше уже упоминалось, что в общем случае миграция данных предполагает, что модель данных у передающей и получающих информационных систем различаются. Решать эту задачу можно по-разному, например, через некий универсальный транспортный формат, поддерживаемый обеими системами, но рассчитывать в общем случае на это вряд ли стоит, поэтому рассматривается решение, которое основывается на поддержке актуальной объектной модели МИС Интерин PROMIS Alpha PG, описанной в терминах типов документов платформы Интерин Alpha. Под типом документа понимается метаописание структуры объекта, включающее сведения о полях объекта, технических данных и принадлежности к тому или иному хранилищу. Соответственно, чтобы избежать двойной трансляции (в универсальный формат, а затем во внутренний), данные готовятся сразу в структуре, близкой к целевому типу документа. Точного соответствия не всегда удастся добиться в силу того, что в зависимости от способа подготовки данных и их транспортировки, может не поддерживаться структура, особенно если она иерархическая, целевого объекта, и она должна быть восстановлена на этапе обработки и размещения данных в хранилище.

В качестве примера трансляции данных рассмотрим сведения о персоне в двух МИС. В версии для Oracle персона хранится в разветвленной реляционной структуре, в то время как в версии для PostgreSQL персона представляется одним объектом. Процесс трансляции состоит из следующих этапов, общим для всех типов объектов:

- на стороне передающей системы формируется представление (VIEW), собирающее по реляционной модели сведения о персонах, и представляет их в виде плоской таблицы



- со всеми требуемыми атрибутами, которые сразу имеют названия, соответствующие целевому объекту в принимающей МИС;
- выполняется транспорт данных из передающей в принимающую системы;
- на принимающей стороне процедура конвертирует плоский список атрибутов в JSON-объект, при необходимости изменяя иерархию сущностей;
- в новом объекте выполняется трансляция идентификаторов из значений передающей системы во внутренние справочники и связанные объекты принимающей системы.

Для каждого целевого объекта формируется соответствующий ему набор данных и процедура трансформации, которая в большинстве случаев оставляет структуру без изменений.

Трансляция внешних ссылок

Вторая проблема, которую необходимо решать в процессе загрузки данных – это трансляция ссылок на внутренние данные. Так, в системе-источнике существует своя система внешних ссылок, которая может или переходить в систему-приемник в неизменном виде (если структура данных и способ идентификации не меняется), либо ссылки должны быть пересчитаны, если структура данных или первичные ключи изменены. В общем случае рассматривается второй вариант, поскольку при использовании хранилищ в качестве первичных ключей применяются GUID, в то время как в передающей системе в большинстве своем применяются числовые последовательности. Чтобы обеспечивать связность данных в системе-источнике и системе-приемнике, все хранилища в системе-приемнике имеют атрибут «EXTERNAL_ID», позволяющий хранить данные о внешнем идентификаторе.

Для решения задачи трансляции внешних ключей используется типовая схема:

- Каждый экспортируемый объект в системе-источнике снабжается полем «EXTERNAL_ID», в котором записывается первичный ключ выгружаемой записи. Если запись собственного первичного ключа не имеет, то формируется составной ключ, однозначно эту запись идентифицирующий.
- Если у выгружаемого объекта есть внешние ссылки на другие сущности, то в результирующий объект вводится поле, которое содержит внешний ключ в том виде, как он есть в системе-источнике, с названием «[ИМЯ_ЦЕЛЕВОГО_АТРИБУТА]_EXT».

- На стороне системы-приемника для разрешения внешних ключей, пришедших в виде внутренних ссылок передающей системы, выполняется поиск по внешнему ключу «EXTERNAL_ID» в соответствующем внутреннем хранилище, и если данные найдены, то соответствующее поле формируемого объекта заполняется первичным ключом найденной записи. Если значение не найдено, то генерируется исключение, прерывающее импорт данной записи и фиксирующее это событие в журнале импорта.

Анализ изменений объектов

При обсуждении организационных вопросов миграции данных большое внимание уделялось возможности аддитивного внесения изменений в уже загруженные данные для обеспечения синхронности данных в двух системах в течение продолжительного времени параллельной работы систем. В текущей постановке двусторонней интеграции не предполагается, система-источник является мастер-системой, изменения данных в которой транслируются в систему-приемник, при этом работа в системе-приемнике тоже возможна с некоторыми ограничениями: нет смысла менять или удалять импортированные данные. Технически их модификация не запрещена, но интеграционный сервис приведет загруженные данные к состоянию, идентичному системе-источнику. Как выше отмечалось, отвечает за такую связь поле «EXTERNAL_ID», присутствующее во всех хранилищах и обеспечивающее отбор записи по внешнему идентификатору и ее восстановление.

Применяемый способ синхронизации данных по внешнему идентификатору решает следующие задачи:

- добавление отсутствующих записей;
- отметку об удалении записей, которые в системе-источнике помечены как удаленные (в МИС никакие значимые данные не удаляются, а изменяют статус на «удаленный»);
- редактирование существующих записей в случае, если произошло их изменение на стороне системы источника или приемника. Для исключения избыточной модификации данных производится анализ всех значимых полей объектов, и новая версия объекта вносится только в случае выявления различий.





Транспортные задачи импорта данных

Вопросы доставки данных из одной системы в другую также являются существенными для организации работы, особенно если требуется непрерывная синхронизация данных. Весь процесс импорта данных можно разделить на два существенных этапа:

1. Однократная первоначальная загрузка большого объема данных.
2. Регулярная актуализация ограниченного объема данных в течение длительного времени.

В целом, задача репликации является классической, для ее решения существует масса ETL-инструментов (extract, transform, load – извлечение, преобразование, загрузка) [6]. Многие имеющиеся ETL-инструменты в основном ориентированы на решение конкретных задач, а универсальные решения достаточно громоздки, чтобы их разворачивать и применять в рамках миграционного проекта, срок жизни которого соизмерим со временем настройки соответствующей системы репликации. Поэтому необходимо соблюсти баланс между решаемой задачей и затратами на ее реализацию. Нами рассмотрен минимально необходимый объем технологий, который обеспечивает баланс между эффективностью и себестоимостью решения.

Каждый из описанных выше этапов обладает своими особенностями, которые накладывают ограничения на используемые технологии работы с данными. Так, при первоначальной массовой загрузке требуется, в первую очередь, решить вопросы, связанные с большим объемом данных, реализовать их извлечение, передачу и вставку наиболее эффективным способом. Поскольку этот этап в идеальном случае выполняется в промышленной среде однократно (в реальности больше, так как требуется ряд тестовых загрузок, а также частичные загрузки идут на этапе отладки моделей), то накладные расходы на его проведение не так чувствительны. Конкретные применяемые технологии зависят от используемых СУБД, но какой бы ни была технология на передающей стороне, принимающая система в рассматриваемом нами случае построена на PostgreSQL, в которой наиболее производительным средством внесения данных является операция COPY. Таким образом, предпочтительным на данном этапе является формирование dump-файлов с данными для их последующей массовой загрузки. Работа с файлами в ряде случаев может быть вынужденной мерой, поскольку доступ к инфраструктуре передающей

системы может иметь ряд ограничений, поэтому работа с файлами позволяет снизить зависимость от внешних факторов. Массовая загрузка вводит в процесс импорта дополнительный этап обработки, когда данные в «сыром» виде размещаются на принимающей стороне во временных хранилищах, чтобы затем пройти трансформацию в необходимые структуры и осуществить пересчет ссылок.

Второй этап миграции требует обеспечения непрерывной синхронизации тех же данных, которые первоначально были импортированы. Одна из задач, которую нужно решить на этом этапе связана с отслеживанием происходящих изменений в передающей системе. Делать это можно или на системном, или на прикладном уровне. В случае с Oracle существуют решения типа Oracle GoldenGate или Quest Shareplex, позволяющие выполнять чтение изменений в базе данных из redo/archive логов, не требуя внесения каких-либо корректировок в действующую ИС или понимания сущности тех или иных изменяемых объектов. Это здорово, но дорого и трудоемко для решения достаточно компактной задачи. Вместе с тем, какой бы ни была технология отслеживания изменений, если базы различны, то придется описывать состав объектов, а он у нас уже описан на предыдущем шаге, когда выполнялась трансформация моделей. Поэтому нами рассмотрен может не самый эффективный, но достаточно простой вариант использования тех же представлений, которые создавались для первоначальной выгрузки, только ограниченных по времени модификации данных. За счет этого сокращается объем выборки, и уменьшается время обработки данных, тем самым обеспечивается достаточный контроль изменений. Если не пытаться максимизировать производительность, сокращая объем анализируемых данных, а выбирать период «с запасом», соотнося результирующее множество с производительностью системы передачи и обработки данных, то можно подобрать решение, которое будет достаточно надежно отслеживать изменения, но при этом не требовать излишних вычислительных ресурсов.

Решая вопрос транспорта данных до принимающей системы, приходится ориентироваться на имеющиеся возможности и ограничения инфраструктуры проекта. Так, если есть возможность непосредственного доступа между серверами передающей и принимающей систем, то задачу фоновой синхронизации удобно решать через технологию FDW[7], поддерживаемую PostgreSQL



для достаточно большого количества источников данных, включая основные СУБД. Так, в идеальном варианте принимающая сторона с определенной периодичностью опрашивает передающую через настроенные fdw-таблицы, настроенные на ранее созданные представления на стороне передающей стороны, анализирует изменения и фиксирует их в хранилищах принимающей системы. Если непосредственного доступа нет и настроить доступ без промежуточных выгрузок данных в файлы не получается, то актуализация данных строится через настройку внешних процессов выгрузки данных из передающей системы, передачу их в виде файлов для обработки принимающей системе и дальнейшую загрузку и обработку.

Выводы

Повсеместный опыт эксплуатации информационных систем, в том числе в медицинских организациях, приводит к существенному накоплению данных, которые необходимо поддерживать в доступном состоянии в течение длительного времени, в том числе при модернизации существующих или миграции на новые информационные системы. Возможно, в перспективе внедрения

централизованных сервисов в сфере здравоохранения актуальность заботы о данных на уровне медицинских организаций постепенно будет снижаться, и локальная МИС МО будет не более чем инструментом для работы с облачными хранилищами, но в настоящее время миграция унаследованных данных является насущной проектной задачей, некоторый опыт и подходы к ее решению нашли свое отражение в данной работе.

Помимо технических вопросов миграции важной является место импорта данных в организации проекта модернизации информационной системы в организации. На примерах показано, как наличие возможности синхронизации данных или их определенной части между двумя системами может влиять на выбор стратегии реализации проекта, снижать риски этого процесса, лимитировать нагрузку на проектную команду и персонал заказчика. Несомненно, выбирая стратегию реализации проекта необходимо ориентироваться на имеющиеся условия и ограничения, но иметь в запасе методологию и инструментальные средства плавной поэтапной миграции позволяет более гибко учесть имеющиеся риски и требования заказчика, а также сокращать себестоимость работ.



СПИСОК ИСТОЧНИКОВ

1. Медицинская информационная система «Интерин» <http://www.interin.ru> [Healthcare Information System «Interin»].
2. *Белышев Д.В., Кочуров Е.В.* Модульные хранилища данных в платформе «Интерин IPS» для PostgreSQL. // Врач и информационные технологии. – 2021. – № 55. – С. 32–43.
3. *Белышев Д.В., Кочуров Е.В.* Анализ методов хранения данных в современных медицинских информационных системах. // Программные системы: теория и приложения: электрон. научн. журн. – 2016. – № 2(29). – С. 85–103. [Belyshev DV, Kochurov EV. “Analysis of Data Storage Methods for Modern Healthcare Information Systems”, Program systems: theory and applications, 2016, 7:2(29), P. 85–103. (In Russ)].
4. *Белышев Д.В., Кочуров Е.В.* Перспективные методы работы с данными в медицинских информационных системах. // Программные системы: теория и приложения: электрон. научн. журн. – 2016. – № 3(30). – С. 79–97. [Belyshev DV, Kochurov EV. “Advanced Methods of Data Management in Healthcare Information Systems”, Program systems: theory and applications, 2016, 7:3(30), P. 79–97. (In Russ).]
5. *Белышев Д.В.* Опыт импортозамещения в медицинской информационной системе «Интерин PROMIS Alpha» // Программные системы: теория и приложения. – 2022. – Т. 13. – № 4(55). – С. 93–109.
6. 8 лучших ETL-инструментов для современного дата-стека <https://renta.im/ru/blog/best-etl-tools/>
7. Foreign Data Wrappers. https://wiki.postgresql.org/wiki/Foreign_data_wrappers





ORIGINAL PAPER

TECHNOLOGY FOR DATA IMPORT TO INTERIN PROMIS ALPHA PG

D.V. Belyshev ✉

Federal Ailamazyan A.K. Program Systems Institute of Russian Academy of Sciences,
Pereslavl-Zalessky, Russia.
<https://orcid.org/0000-0002-0437-4814>

✉ Corresponding author: Belyshev D.V.

ABSTRACT

The key step in launching an information system is the legacy data transfer. There are many tools and techniques to help us solve this problem in terms of special features of software products and projects. The article studies the methodology and means of migrating data when implementing the Interin PROMIS Alpha PG medical information system.

Keywords: data import, import substitution, postgresql, healthcare information systems.

For citation: Belyshev D.V. Technology for data import to Interin PROMIS Alpha PG. *Manager Zdravoohranenia*. 2023; 5:85–92. DOI: 10.21045/1811-0185-2023-S-85-92

REFERENCES

1. Medical information system "Interin" <http://www.interin.ru> [Healthcare Information System "Interin"].
2. Belyshev D.V., Kochurov E.V. Modular data warehouses in the Interin IPS platform for PostgreSQL. // Doctor and information technology. – 2021. – No. S5. – P. 32–43.
3. Belyshev D.V., Kochurov E.V. Analysis of data storage methods in modern medical information systems. // Software systems: theory and applications: electron. scientific magazine. – 2016. – No. 2(29). – P. 85–103. [Belyshev DV, Kochurov EV. "Analysis of Data Storage Methods for Modern Healthcare Information Systems", Program systems: theory and applications, 2016, 7:2(29), P. 85–103. (In Russ)].
4. Belyshev D.V., Kochurov E.V. Promising methods for working with data in medical information systems. // Software systems: theory and applications: electron. scientific magazine. – 2016. – No. 3(30). – P. 79–97. [Belyshev DV, Kochurov EV. "Advanced Methods of Data Management in Healthcare Information Systems," Program systems: theory and applications, 2016, 7:3(30), P. 79–97. (In Russ)].
5. Belyshev D.V. Experience of import substitution in the medical information system "Interin PROMIS Alpha" // Software systems: theory and applications. – 2022. – T. 13. – No. 4(55). – P. 93–109.
6. 8 best ETL tools for a modern data stack <https://renta.im/ru/blog/best-etl-tools/>
7. Foreign Data Wrappers. https://wiki.postgresql.org/wiki/Foreign_data_wrappers

ИНФОРМАЦИЯ ОБ АВТОРАХ / ABOUT THE AUTHORS

Бельшев Дмитрий Владимирович – к.т.н., зав.лаб. Исследовательского центра медицинской информатики, ФГБУН «Институт программных систем им. А.К. Айламазяна» Российской академии наук, г. Переславль-Залесский, Россия.

Dmitry V. Belyshev – Ph.D., head of laboratory. Research Center for Medical Informatics, Federal Ailamazyan A.K. Program Systems Institute of Russian Academy of Sciences, Pereslavl-Zalessky, Russia.

E-mail: belyshev@cron.botik.ru