

ОСОБЕННОСТИ ПОСТРОЕНИЯ СЕРВЕРНОЙ БАЗЫ ДАННЫХ ИНТЕГРИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ АНАЛИЗА ПОТЕНЦИАЛА ВОЗОБНОВЛЯЕМЫХ ИСТОЧНИКОВ ЭНЕРГИИ

Тонконогов Б.А.¹, Кундас С.П.², Мороз А.Е.³, Масловская Е.С.²

¹Учреждение образования «Международный государственный экологический институт имени А. Д. Сахарова» Белорусского государственного университета,

г. Минск, Республика Беларусь, boristonkonogov@iseu.by

²Белорусский национальный технический университет,

г. Минск, Республика Беларусь, kundas@tut.by

³ООО «Сенсотроника»,

г. Минск, Республика Беларусь, marozaliaksei@gmail.com

Рассмотрены особенности построения серверной базы данных интегрированной информационной системы для анализа потенциала возобновляемых источников энергии, содержащей информацию о потенциале возобновляемых источников энергии и применяемом оборудовании возобновляемой энергетики.

Введение

Оценка возможностей использования потенциала возобновляемых источников энергии является актуальной научной и технической задачей, важной для обеспечения энергетической безопасности Республики Беларусь. Страна не обладает в достаточном количестве собственными углеводородными источниками энергии, поэтому важным направлением в энергетической политике, в том числе в энергосбережении для нашей страны, является использование возобновляемых источников энергии, которые позволяют сократить импорт дорогостоящих углеводородных ресурсов и улучшить экологические показатели энергетики [1]. Для решения указанной задачи авторами настоящей работы создана интегрированная информационная система [2], реализующая определенные расчетные методы и содержащая серверную базу данных потенциала возобновляемых источников энергии и оборудования возобновляемой энергетики, которая является объектом рассмотрения в настоящей статье.

1. Выбор технологий и средств реализации базы данных

В настоящее время создание баз данных в составе различных Web-ориентированных интегрированных информационных систем, кроме сбора информации для последующего хранения, может преследовать следующие цели [2, 3]:

- предоставление пользователям хранимой информации в качестве справочной;
- использование информации для проведения анализа и предоставление его результатов пользователям.

В зависимости от цели создания, существует 2 типа баз данных: *многомерные* и *реляционные* [4 - 6].

Многомерная модель баз данных характеризуется представлением на физическом уровне ее многомерной структуры, то есть хранение информации происходит не в виде реляционных таблиц, а в виде упорядоченных многомерных массивов: гиперкубов и поликубов. К достоинствам таких баз данных можно отнести высокую производительность, достигаемую за счет разреженности данных. Однако разреженность хранимой информации приводит к серьезному недостатку – неэкономному расходованию внешней памяти.

Для реляционных баз данных характерно преобразование информации в многомерную модель через промежуточный слой метаданных. В этом случае гиперкуб эмулируется системой управления базами данных на логическом уровне. При использовании реляционной модели баз данных размер хранилища не является таким критичным параметром, как в случае многомерных баз данных, и, кроме того, внесение в структуру новых измерений не влечет за собой ее физической реорганизации. Главный недостаток реляционных систем управления

базами данных по сравнению с многомерными – меньшая производительность. Однако при тщательной проработке баз данных, они по производительности могут приблизиться к многомерным.

Ввиду того, что информация, хранящаяся в базе данных, носит динамический характер, то для создания серверной базы данных оборудования возобновляемой энергетики и потенциала возобновляемых источников энергии целесообразным было использование реляционной модели баз данных. В качестве инструментов, с помощью которых была осуществлена разработка, выбраны достаточно производительный сервер баз данных *Microsoft SQL Server* и система управления базами данных *Microsoft SQL Server Management Studio*.

2. Проектирование базы данных

На основе проведенного анализа требуемого содержимого серверной базы данных информацию, которую необходимо хранить в ней, можно классифицировать следующим образом:

- 1) данные о *потенциале возобновляемых источников энергии*, в том числе, метеорологические и климатические данные (среднемесячная величина прямой и рассеянной солнечной радиации для каждого часа безоблачного неба, среднемесячная скорость ветра, почасовое значение температуры окружающего воздуха в среднем для каждого месяца, информация, необходимая для расчета энергетической и экономической эффективности использования возобновляемых источников энергии и др.);
- 2) данные об *объектах* (географические координаты, установленное оборудование, локальный потенциал возобновляемых источников энергии и так далее);
- 3) справочные данные об *оборудовании*, выпускаемом различными производителями (технические параметры и характеристики, информация о производителях и так далее).

Хранение параметров оборудования, применяемого в области возобновляемой энергетики. В базе данных предусмотрено хранение параметров оборудования, используя которые проводятся различные аналитические расчеты эффективности его использования. Таблицы хранят следующую обязательную информацию:

- данные о производителе;
- название модели;
- цена;
- срок службы.

Если дополнительная информация о производителе и названии модели носит информационный характер, то цена и срок службы необходимы при проведении экономической оценки эффективности использования выбранного оборудования. Также предусмотрено хранение подробной информации о производителе оборудования (для обеспечения обратной связи):

- Web-сайт;
- адрес;
- адрес электронной почты.

Для оценки энергетической эффективности использования оборудования необходимо хранить дополнительные параметры. Эта информация содержится в следующих таблицах с соответствующими полями:

- *sun_batteries* – солнечные батареи;
- *sun_collectors* – солнечные коллекторы;
- *windmill* – ветроэнергетическое оборудование;
- *hydroelectric* – малые гидроэлектростанции;
- *pumps* – тепловые насосы;
- *boilers* – котлоагрегаты;
- *biogas* – биогазовые станции;

- *briquette* – оборудование по брикетированию топлива (не используется при проведении аналитических расчетов).

Также для упрощения таких операций как поиск оборудования, получение списков недавно добавленного оборудования и его сортировка по производителем были добавлены вспомогательные таблицы *added_equipment* и *equipment_type*. Таблица *added_equipment* заполняется по вызову триггеров при операциях вставки, обновления и удаления в таблицах оборудования (*sun_collectorus*, *sun_batteries*, *hydroelectric*, *pumps*, *windmill*, *biogas*, *boilers*).

Хранение информации об объектах. Для эффективного проведения анализа в базе данных необходимо хранить следующую информацию об объектах:

- географическое положение населенного пункта;
- количество ресурсов, имеющихся на объекте;
- установленное оборудование.

Информация о местоположении, типе и размерах каждого объекта хранится в следующих таблицах:

- *objects* – основная информация об объектах;
- *regions* и *oblasts* – информация о принадлежности объекта к району и области;
- *object_types* – информация, определяющая тип объекта.

Хранение информации об установленном на объектах оборудовании. Информация об установленном оборудовании хранится в таблицах, которые являются связующими между таблицей объектов и таблицами, хранящими информацию об оборудовании. Организация хранения этой информации осуществляется путем добавления записи в соответствующую связующую таблицу, содержащую информацию об идентификаторах объекта и производителя оборудования, названии установленной модели и их количестве.

Хранение информации о потенциале возобновляемых источников энергии. Для проведения аналитических расчетов в базе данных необходимо хранить следующие метеорологические данные:

- удельный тепловой поток, прямого H_B и рассеянного H_D солнечного излучения, падающего на 1 м^2 горизонтальной поверхности в каждый час безоблачного неба (по месяцам в зависимости от времени суток);
- среднемесячная температура окружающего воздуха для каждого часа;
- среднемесячные значения скорости ветра и коэффициенты, зависящие от типов поверхностей для определенной местности.

Все метеорологические данные представляют собой усредненные по всей территории региона значения. Это обусловлено несколькими факторами:

- отсутствие подробных метеорологических данных отдельно для каждого объекта;
- изменения данных показателей в различных точках региона не оказывают значительных влияний на результаты анализа.

Для хранения метеорологических данных использовалась определенная схема их организации. Таблицы *rad_s*, *rad_d*, *wind_speed* и *temperature* связаны с таблицей *month*, что позволяет хранить информацию отдельно для каждого месяца. Эти таблицы также связаны с таблицей *regions*, которая в свою очередь связана с таблицей *objects*.

Для оценки эффективности использования потенциала ветровой энергии необходимо располагать информацией о рельефе местности на каждом объекте. Данная информация хранится в таблице *surfaces*, которая связана с таблицей объектов через таблицу *surf_obj*.

Для хранения информации о наличии доступного для использования древесного топлива созданы таблицы:

- *biomass_type* – содержит информацию о древесных породах;
- *bt_obj* – содержит информацию о наличии древесной породы на объекте.

Для хранения информации о доступном для использования в биогазовых установках количестве биомассы созданы таблицы:

- *biogas_make* – содержит информацию о сырье для биогазовых установок и его характеристиках;
- *biostuff* – содержит информацию о количестве сырья на объекте.

Таким образом, разработана общая реляционная схема данных, содержащая указанную информацию. Также были реализованы некоторые функциональные объекты базы данных: хранимые процедуры, функции, триггеры и представления [4].

Хранимые процедуры и функции – это отдельные программы, написанные на языке T-SQL. Эти процедуры являются частью метаданных базы данных. Они могут получать входные параметры, возвращать значения приложению и быть вызваны явно из Web-приложения или подстановкой вместо имени таблицы в инструкции SELECT. В рамках рассматриваемой задачи хранимые процедуры и функции обеспечивают следующие возможности:

- модульность – хранимые процедуры могут быть общими для приложений (программных модулей), которые обращаются к той же самой базе данных, что позволяет избегать повторяющегося кода и уменьшает размер приложений;
- упрощение сопровождения приложений – при обновлении процедур, изменения автоматически отражаются во всех приложениях, которые используют их, без необходимости их повторной компиляции и сборки;
- повышение эффективности работы – для удаленных клиентов процедуры выполняются сервером, а не самими клиентами, что снижает сетевой трафик и повышает производительность.

Триггеры – это отдельные программы, ассоциированные с таблицами или представлениями, которые автоматически выполняют определенные действия при завершении операций (возможно с сообщениями об ошибках), установке значений в записях, к которым происходят обращения, при добавлении, изменении или удалении строк в таблицах или представлениях и другие операции. Когда триггер вызван, он имеет непосредственный доступ к добавлению, изменению или уничтожению данных. Триггеру могут быть также доступны данные из других таблиц. В созданной базе данных триггеры обеспечивают следующие возможности:

- автоматическое ограничение ввода данных – гарантия того, что пользователь ввел только допустимые значения в поля столбцов;
- упрощение сопровождения приложений – изменение в триггерах автоматически отражаются во всех приложениях, которые используют таблицы со связанными с ними триггерами;
- автоматическое документирование изменений таблицы – приложение может управлять журналом изменений с помощью триггеров, которые выполняются всякий раз, когда происходят изменения таблиц.

Представления – это виртуальные таблицы, которые не сохранены физически в базе данных, но ведут себя точно также как реальные таблицы. В базе данных представления содержат данные (информационные выборки) из одной или более таблиц или других представлений и используется для хранения часто используемых запросов или множества запросов в базе данных. Однако представления могут обеспечить ограниченные средства защиты, так как предоставляют доступ пользователей к подмножеству доступных данных при скрытии других связанных и чувствительных данных.

3. Программная реализация алгоритмов работы аналитических модулей базы данных (хранимых процедур, функций и представлений)

В составе разработанной базы данных на основе созданных специальных программных алгоритмов реализованы соответствующие хранимые процедуры, функции и представления:

- *dbo.all_energy* – хранимая процедура для оценки эффективности использования оборудования, информация о котором хранится в базе данных;
- *dbo.sun_battery* – пользовательская функция, проводящая расчет энергетической эффективности использования солнечных батарей;

- *dbo.sun_collector* – пользовательская функция, проводящая расчет энергетической эффективности использования солнечных коллекторов;
- *dbo.wep* – пользовательская функция, проводящая расчет энергетической эффективности использования ветроэнергетического оборудования;
- *dbo.mhes* – пользовательская функция проводящая расчет энергетической эффективности использования мини-ГЭС;
- *dbo.blrs* – пользовательская функция, проводящая расчет энергетической эффективности использования котлоагрегатов;
- *dbo.bgs* – пользовательская функция, проводящая расчет энергетической эффективности использования биогазовых установок;
- *dbo.pmps* – пользовательская функция, проводящая расчет энергетической эффективности использования тепловых насосов;
- *dbo.sc_view* – представление для отображения солнечных коллекторов.

4. Программный механизм интеграции, администрирования и защиты базы данных в составе информационной системы

Программный механизм интеграции базы данных оборудования возобновляемой энергетики и потенциала возобновляемых источников энергии в состав информационной системы, обладающей Web-ориентированным графическим интерфейсом пользователя, предполагает реализацию программных модулей для взаимодействия серверной части Web-приложения с базой данных. В частности, был реализован модуль доступа к данным с использованием *ORM-библиотеки Entity Framework*. Общая функциональная блок-схема алгоритма работы указанного механизма представлена на рисунке 1.

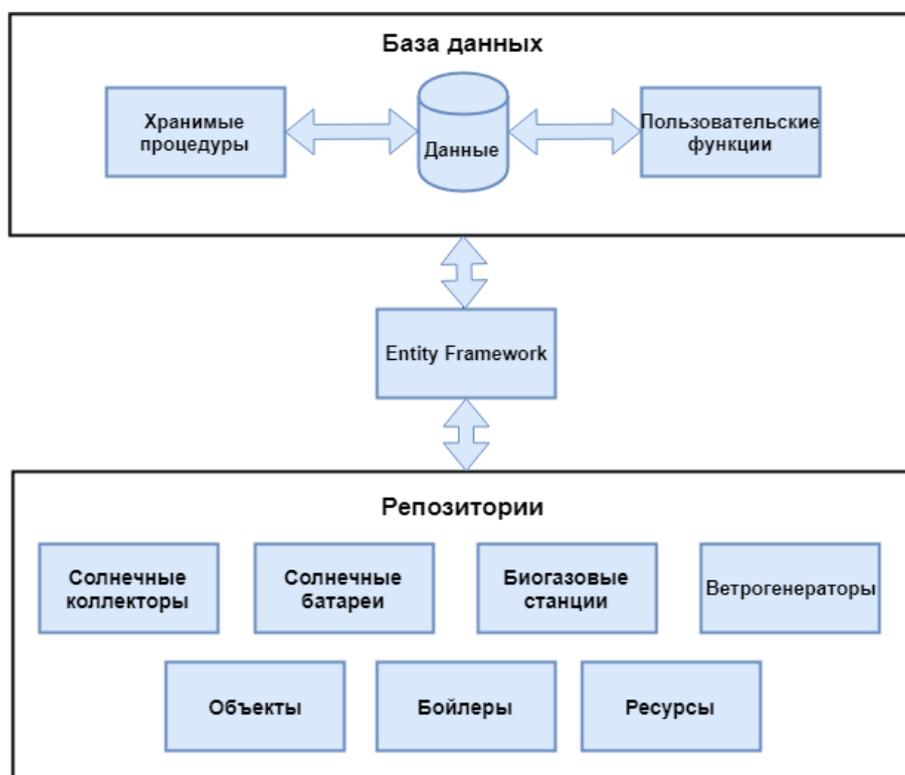


Рисунок 1 – Блок-схема алгоритма интеграции базы данных в состав информационной системы

В приведенной блок-схеме элемент «*Ресурсы*» представлен соответствующим программным функционалом, соответствующим энергетическому потенциалу различных возобновляемых источников энергии:

- солнечная радиация;
- ветровой поток;

- биогаз;
- биомасса;
- водные ресурсы.

При создании приложения использовалась технология *Database First*, так как база данных уже была создана на предыдущих этапах. На основе существующей базы данных при использовании Entity Framework были реализованы следующие классы моделей для хранения информации о:

- *SunBattery* – солнечных батареях;
- *SunCollector* – солнечных коллекторах;
- *Windmill* – ветроэнергетическом оборудовании;
- *Biogas* – биогазовых установках;
- *Boiler* – водогрейных котлах;
- *MHES* – мини-ГЭС;
- *Locations* – объектах на карте;
- *SunBatteryObject*, *SunCollectorObject*, *WindmillObject*, *BiogasObject*, *BoilerObject* и *MHESObject* – установленном оборудовании на объектах;
- *Sun*, *Wind*, *Biog*, *Biom* и *Water* – суммарном (в зависимости от районов) энергетическом (ресурсном) потенциале солнечной радиации, ветрового потока, биогаза (твердых коммунальных отходов), биомассы (древесного топлива) и водных ресурсов.

Расчет энергетического (ресурсного) потенциала в зависимости от условий и параметров оборудования производится хранимыми процедурами и пользовательскими функциями, хранящимися как объекты базы данных. Сгенерированные с помощью Entity Framework классы соответствуют таблицам базы данных, а поля этих классов – полям таблиц. На рисунке 2 показан пример класса, содержащего информацию об оборудовании, использующем возобновляемые источники энергии – *SunCollector*.

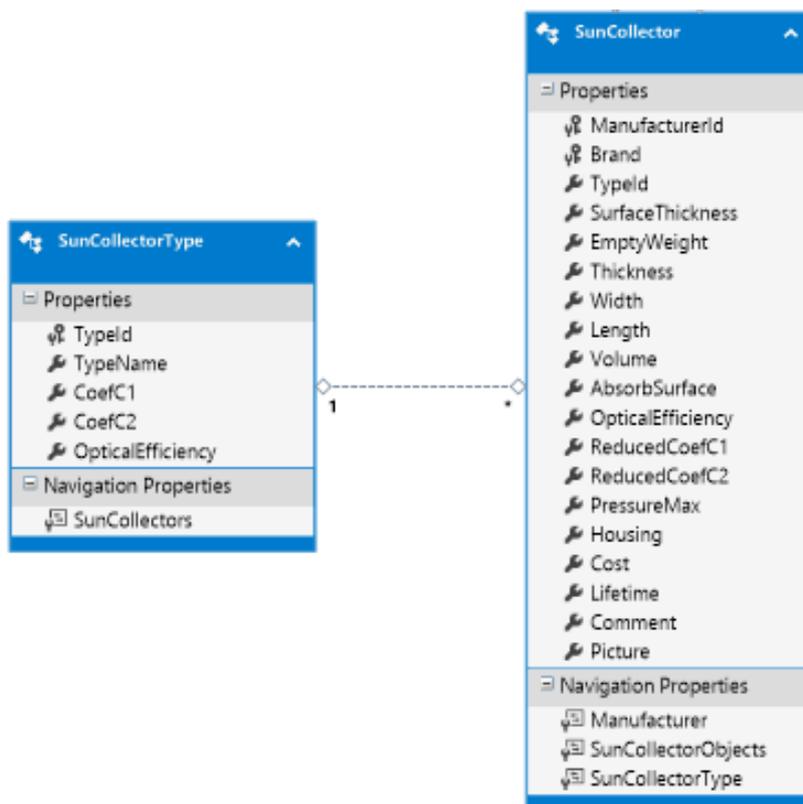


Рисунок 2 – *SunCollector* – класс, представляющий солнечные коллекторы

Указанный класс содержит такие поля как:

- производитель;

- модель;
- оптическая эффективность;
- поглощающая поверхность;
- объем теплоносителя в коллекторе и прочие.

Информация о типах солнечных коллекторов была вынесена в отдельную таблицу в базе данных для сокращения избыточности. Типы солнечных коллекторов представлены классом *SunCollectorType*. Классы для остальных типов оборудования построены схожим образом, то есть имеют связь «многие к одному» с классом производителя и «один ко многим» с классом, представляющим соответствующее оборудование, установленное на объектах.

Модуль, отвечающий за логику работы с данными, реализован с помощью шаблона проектирования «Репозиторий». Использование такого подхода позволяет избежать проблем при изменении источника данных. В таком случае редактированию подвергнется только код в промежуточном слое данных, то есть код репозитория, а код представления затронут не будет. Приложение становится более гибким и легко адаптируется к изменениям. Серверная часть Web-приложения извлекает необходимую информацию из базы данных через объекты репозитория.

Например, в коде интерфейса *IWindmillsRepository* репозитория для ветроэнергетических ресурсов и оборудования метод *SelectByParameters* позволяет производить подбор ветроэнергетического оборудования по заданным параметрам. Расчет ресурсного потенциала производится с помощью хранимых процедур базы данных. Вызовы процедур для расчета энергетического потенциала инкапсулированы в методы репозитория соответствующих ресурсов. Реализация методов *GetEnergyPotential* и *GetEconomicPotential* указанного репозитория предназначены для расчета энергетического потенциала и экономической эффективности соответственно.

В целях разграничения прав пользователей и защиты от несанкционированного доступа к содержимому серверной базы данных реализован алгоритм, использующий систему авторизации пользователей *ASP.NET Identity* [6]. Эта система состоит из библиотеки *Microsoft.AspNet.Identity* с описанием основных классов и абстракций в виде интерфейсов хранилища пользователей и так далее. *Microsoft.AspNet.Identity.EntityFramework* – библиотека, которая содержит реализацию пользователя и его роли, а также хранилищ на основании *Entity Framework*. *ASP.NET Identity* использует технологию *Code First* при создании таблиц в базе данных. Так как приложение уже имеет существующую базу данных, то таблицы, необходимые *ASP.NET Identity* для работы с пользователями, создавались вручную в существующей базе данных.

Заключение

1. Разработан программный механизм интеграции базы данных оборудования возобновляемой энергетики и потенциала возобновляемых источников энергии в состав информационной системы, которая обладает Web-ориентированным графическим интерфейсом пользователя и представляет собой серверный ресурс. Для этих целей созданы специализированные программные модули для взаимодействия серверной части Web-приложения с базой данных и реализованы некоторые классы моделей объектов базы данных.
2. Использована оптимизированная реляционная структура (модель) данных, состоящая из таблиц, включающих поля для хранения параметров оборудования, применяемого в области возобновляемой энергетики, и информации об объектах, установленном оборудовании и ресурсном потенциале возобновляемых источников энергии.
3. Реализованы функциональные объекты базы данных (хранимые процедуры, функции, триггеры и представления) на основе созданных специальных программных алгоритмов.
4. Реализован модуль доступа к данным с использованием ORM-библиотеки *Entity Framework* и технологии *Database First*. и оригинальные хранимые процедуры и поль-

зовательские функции, являющиеся объектами базы данных и предназначенные для расчета энергетического (ресурсного) потенциала в зависимости от условий и параметров оборудования.

5. Разработан алгоритм, использующий систему авторизации пользователей ASP.NET Identity, и выполнена программная реализация администрирования и защиты серверной базы данных в целях разграничения прав пользователей и защиты от несанкционированного доступа к ее содержимому.

Список литературы

1. Кундас, С. П. Возобновляемые источники энергии / С. П. Кундас, С. С Позняк, Л. В. Шенец. – Минск: МГЭУ им. А. Д. Сахарова, 2009. – 390 с.
2. Тонконогов Б. А., Кундас С. П., Мороз А. Е. Архитектура и функциональность интегрированной информационной системы для анализа потенциала возобновляемых источников энергии // Системный анализ и прикладная информатика. № 4, 2017. – С. 4 - 15.
3. Moroz A. E., Tonkonogov B. A. Architecture of integrated information system for analysis of potential of renewable energy sources // Actual environmental problems: Proceedings of the VII International Scientific Conference of young scientists, graduates, master and PhD students, November 23 - 24, 2017, Minsk, Republic of Belarus / ISEI BSU; the general editorship: Professor S. A. Maskevitch, Professor S. S. Poznjak. – Minsk: RUE “Information-Computing Center of the Ministry of finance of the Republic of Belarus”, 2017. – 220 p. – P. 173 - 174.
4. Дьюсон, Р. SQL Server 2008 для начинающих разработчиков / Р. Дьюсон. – СПб.: БХВ-Петербург, 2009. – 704 с.: ил.
5. Уолтерс, Р. SQL Server 2008. Ускоренный курс для профессионалов / Р. Уолтерс [и др.]; пер. с англ. Н. А. Мухина. – М.: Вильямс, 2009. – 768 с.: ил.
6. Максимов, Н. В. Базы данных: учебное пособие [рек. УМО РФ] / Н. В. Максимов, И. И. Попов, О. Л. Голицына. – 4-е изд., перераб. и доп. – М.: ФОРУМ: ИНФРА-М, 2016. – 400 с.: ил.