Федеральное агентство связи

Ордена Трудового Красного Знамени

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Московский технический университет связи и информатики»

Разрешаю

допустить к защите

Зав. кафедрой

            2020 г.

**БАКАЛАВРСКАЯ РАБОТА**

НА ТЕМУ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Рецензент: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва 2020 г.

**Содержание**

[Введение 3](#_Toc30404483)

[1. Анализ предметной области и постановка задачи 7](#_Toc30404484)

[1.1 Описание организации 7](#_Toc30404485)

[2. Выбор технологии 11](#_Toc30404486)

[2.1. Язык программирования информационной системы 11](#_Toc30404487)

[2.2. Среда разработки информационной системы 14](#_Toc30404488)

[2.3. Современные технологии разработки веб-приложений. Особенности платформы ASP.NET Core 18](#_Toc30404489)

[Список использованных источников 29](#_Toc30404490)

# Введение

Туризм в начале 21 века стал одним из ведущих направлений социально-экономической и культурной жизни многих государств и регионов мира. Сегодня туризм является одной из наиболее высокодоходных и динамично развивающихся отраслей экономики, оказывающих сильное мультипликативное влияние на совокупную деятельность различных секторов экономики. Туристическая индустрия во всем мире сейчас переживает период активного развития. При этом параллельно с традиционными формами отдыха получают развитие новые, всякого рода «зеленые» путешествия, в том числе и сельский туризм.

Россия обладает уникальными природно-климатическими и культурно историческими особенностями, позволяющими развивать практически все популярные виды туризма, одним из которых является сельский туризм. Доля сельского туризма в России пока невелика и на сегодняшний день составляет, по экспертным оценкам, 1,5–2 %. Однако, в России есть все предпосылки для развития этого вида туризма. [9]

Единственное закрепленное в официальном документе (ГОСТ Р 56641-2015) на федеральном уровне определение сельского туризма звучит следующим образом: «Сельский туризм — деятельность по организации отдыха в сельской местности или в малых городах (при отсутствии промышленных зон и застройки) с предоставлением услуг гостеприимства в частном секторе с возможностью трудового участия, ориентированная на использование природных, культурно-исторических и других ресурсов, традиционных для данной местности».

Сегодня понятие «сельский туризм» трактуется профессиональным сообществом как «комплексный туризм в сельской местности, способствующий устойчивому развитию сельских территорий». Основной ресурс сельского туризма — это сельская территория, а основная задача — сохранение ее традиций и вклад в устойчивое развитие посещаемых регионов.

Сельский туризм является важным фактором в обеспечении устойчивого развития сельских территорий, роста благосостояния и занятости сельского населения. Он позволяет сохранить и продемонстрировать бытовые, культурные и национальные особенности аграрного региона, дает уникальную для современных горожан возможность приобщиться к иной культуре, узнать на собственном опыте, что такое сельская жизнь и сельский труд. Учитывая высокую социально-экономическую значимость данного направления и его мультипликативный эффект на региональном и муниципальном уровнях.

Сельский туризм предполагает временное пребывание туристов в сельской местности с целью отдыха и/или участия в сельскохозяйственных работах; сектор туристической отрасли, ориентированный на использование природных, культурно-исторических и иных ресурсов сельской местности и ее специфики для создания комплексного туристского продукта. От сельского туризма ожидают спокойствия и размеренности сельской жизни, чистого воздуха, тишины и натуральных продуктов, комфортных условий проживания, домашней атмосферы, приемлемых цен, ощущения близости с природой, получения новых впечатлений, возможности развлечения для детей и проведение досуга для взрослых.

Современное состояние сельского туризма Российской Федерации весьма скромное. Лидерами в организации сельского туризма являются Алтайский край, Чувашская Республика, Республика Карелия, Владимирская, Вологодская, Ленинградская, Ярославская, Псковская, Астраханская, Калужская области.

Создать хороший туристический продукт недостаточно, важно добиться того, чтобы о нем узнали будущие клиенты. Именно поэтому очень важно проводить информатизацию сферу сельского туризма, что позволит популяризировать и развивать этот элемент туристической деятельность. Одним из способов продвижения услуг сельского туризма является веб-сайт. Создание собственного сайта – ответственный этап. Сложность состоит не столько в том, чтобы один раз сделать красивый и удобный в пользовании сайт, но в том, чтобы поддерживать его и обновляться необходимой регулярностью. Полноценный сайт состоит из нескольких страниц (основных блоков или разделов), между которыми пользователь перемещается по гиперссылкам. Сайты бывают сложными (состоящими из большого количества разделов и подразделов) и простыми. Последние обычно называют «сайт-визитка». Как правило, не большим объектам сельского туризма для популяризации своей деятельности достаточно именно «сайта-визитки», который, как правило является одностраничным веб-приложением, или веб-приложением, содержащим до 10 страниц. Существует множество инструментов и способов разработки веб-приложений. Так, с помощью языка гипертекстовой разметки и каскадных таблиц стилей можно создавать простые статические веб-страницы, которые можно использовать в качестве сайтов-визиток. Подключая возможности Javascript, с простой статической веб-страницы можно сделать динамический сайт, который реагирует на действия пользователя. Значительное расширение функционала веб-приложения происходит при использовании серверных языков программирования, которые позволят выполнять серверные сценарии и соединятся с базой данных. Ключевым звеном в подобных системах является использованный серверный язык программирования и система управления базами данных. В среде веб-разработки используется множество технологий, при этом, наиболее используемыми являются PHP, ASP.NET, Perl, Phyton, которые позволяют создавать функциональные веб-приложения. Одной из новых и набирающих популярность технологий, с помощью которых можно создавать серверные приложения является технология .NET. Серверные приложения на платформе .NET можно разрабатывать в двух поддерживаемых реализациях: .NET Framework и .NET Core. В них используется множество одинаковых компонентов, а код можно использовать как в одной среде, так и в другой. При этом, между этими двумя средами существуют фундаментальные различия, и выбор платформы для реализации проекта зависит от поставленной задачи.

Таким образом, целью работы является Разработка прототипа сайта компании по предоставлению услуг сельского туризма

Для достижения цели необходимо решить следующие задачи:

* рассмотреть предметную область;
* выявить объекты и определить связи между ними;
* провести анализ информационных потоков;
* обосновать выбор технологии разработки проекта;
* разработать структура проекта;
* выбрать СУБД для реализации проекта;
* реализовать сайт на базе выбранной технологии;
* проверить работоспособность созданного сайта.

# Анализ предметной области и постановка задачи

## 1.1 Описание организации

Компания представляет два объекта: земельная территория, находящаяся рядом с селом и офис, находящийся в городе. Место непосредственного отдыха находится в близи села так как сельская местность придает месту отдыха значимый социальный и исторический фактор. А городское местонахождение офиса полезно в качестве поиска квалифицированных работников бухгалтерии, кадрового учета, юриспруденции.

Множество компаний открывают места для отдыха рядом или внутри старинных губерний, старинных мест обитания предков. Таковой выбор придаёт месту большую социальную огласку, интерес среди людей. Также выбор сельской местности для установки базы отдыха сельского туризма сможет посодействовать поиску квалифицированных работников для объекта и увеличить количество рабочих мест, развитием инфраструктуры.

У многих организаций есть желание отправлять своих работников на отдых после трудных рабочих сессий. Такой организацией в моём творческом примере является далее – X Банк. Вдохновившись идеей развития сельского туризма и экологически чистого оздоровительного отдыха X Банк (Рисунок 1.1) приходит к мнению о создании компании по предоставлению услуг сельского туризма. Данная компания принесет ещё больший доход и льготный отдых для работников Банка и всех его структур.



Рисунок 1.1 логотип банка

У организации имеется помещение и штат сотрудников. Помещение арендуется в бизнес комплексе (Рисунок 1.2).



Рисунок 1.2 Бизнес комплекс

Под организацию может быть выделено несколько этажей. Имеется всё необходимое для работы сотрудников. Так же имеется своя сеть и серверное оборудование. В дальнейшем серверное оборудование будет необходимо для использования в проекте проекта.

В случае локальной сети (Рисунок 1.3), она находится в одном сегменте, а интернет-серверы – в другом. Маршрутизатор выполняет роль шлюза в интернет, также и для серверов.



Рисунок 1.3 Топологическая карта сети

Серверная оснащена оборудованием SUPERMICRO SYS-6017R X9DRI-LN4F+ (Рисунок 1.4), чип сет Intel C602 Сокет LGA201.

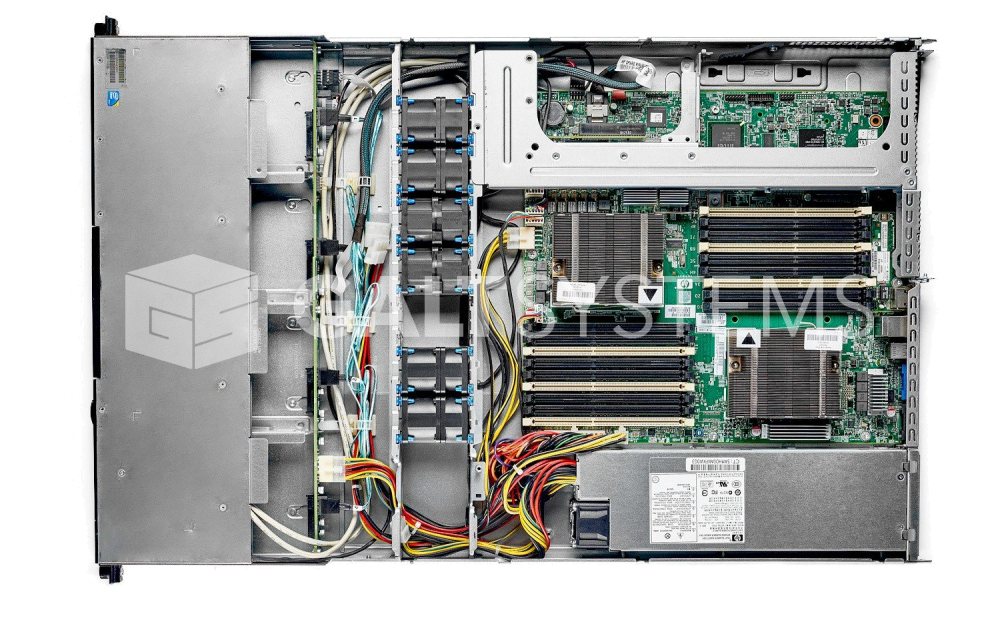


Рисунок 1.4 Сервер

У организации уже имеется свой сайт, сайт предназначен для банковской структуры и не подходит по своему внутреннему составляющему для компании сельского туризма.

# 2. Выбор технологии

## 2.1. Язык программирования информационной системы

Языком программирования для информационной системы выбран C#. Язык программирования C# — это универсальный объектно-ориентированный язык программирования. Первая общедоступная версия этого языка появилась в феврале 2002 года одновременно с выходом среды разработки Microsoft Visual Studio .NET. Начиная с того времени, язык C#, как и программная платформа Microsoft .NET, постоянно развиваются и приобретают всё более и более широкие дополнительные функциональные возможности. Для создания приложений для платформы .NET с использованием языка C# существуют различные средства разработки. Наиболее популярным из них является интегрированная среда разработки (ИСР) Microsoft Visual Studio.

В настоящее время язык программирования C# получает широкое распространение не только как универсальный язык профессиональной разработки программного обеспечения, но и как средство начального обучения программированию. В этом смысле язык C# составляет серьёзную конкуренцию таким языкам программирования, как BASIC и Pascal.

Отличительной особенностью языка программирования C# является то, что он построен на принципах объектно-ориентированного программирования (ООП). При этом любая, даже самая простая программа, является объектно-ориентированной. В языке C# удачно сочетаются испытанные средства программирования с самыми последними новшествами, и предоставляется возможность для эффективного написания программ, предназначенных для вычислительной среды современных предприятий.

С# представляет собой систему на основе специального байт-кода, с обещаниями много-платформенности. В 2000 г. ЕСМА по запросу Microsoft одобрила С# как стандартный язык. В связи с этим, сейчас С# — это язык, определенный в стандарте, который Microsoft не сможет менять по своему усмотрению. В этом является преимущество С#, так как он получил явную независимость от Microsoft.

Теория, лежащая в основе С# состоит в том, что язык может быть использован для создания много платформенных программ. В Windows обычная программа состоит из двоичного исполняемого файла. То есть исходный код программы компилируется в машинный формат. .NET все еще предоставляет такой тип разработки программного обеспечения, хотя он специально называется неуправляемым кодом.

С# содержит большое количество повторно используемого кода. Программы на С# компилируются в специальный байт-код, который может понять и выполнить отдельная программа. Common Language Runtime (CLR) — это программа, которая работает с байт-кодом С#. Однако CLR пошла дальше подхода виртуальной машины, и на самом деле компилирует приложения «на лету» в машинный код, тем самым увеличивая производительность. В случае часто вызываемых приложений для веб-серверов, таких как ASP.NET, результаты компиляции кэшируются и повторно используются, опять же для увеличения производительности.

Таким образом, С# действует по-другому: программы, написанные на этом языке, сначала «компилируются» в CLR-совместимый формат (похоже на то, что делает Java), а затем компилируются в машинный код, когда CLR первый раз их выполняет.

С# предоставляет встроенную поддержку СОМ-компонентов и Windows API, а также ограниченное использование указателей. Это означает, что хотя программы на этом языке могут быть не такими быстрыми, как приложения, выполняемые напрямую (из-за интерпретации CLR), их производительность обычно не такая низкая, в сравнении с другими технологии.

С# реализует модель безопасности формате песочницы, но основной контроль устанавливается сборкой, которая позволяет определить, какие операции могут выполняться некоторой программой или классом. С# поддерживает указатели для прямых манипуляций с памятью, хотя их использование также может контролироваться настройками безопасности.

Теоретически С# переносим, но пока никакие операционные системы производства не Microsoft не могут использовать его. Это не означает, что он полностью непереносим. У Microsoft также есть операционная система для карманных устройств, которая называется Windows Mobile (ранее она называлась Windows СЕ, Windows PocketPC, PocketPC 2000 и т. д), которая распространяется с .NET Framework и CLR для выполнения программ на С#. Как и Java, С# можно использовать на веб-серверах, настольных компьютерах и карманных устройствах, если на них есть соответствующая версия Windows.

С# реализует автоматический сбор мусора. В С# можно объявлять деструкторы, важно отметить, что они вызываются, когда процесс сбора мусора определяет, что объект больше недоступен коду и что необходима память. Деструкторы в С# имеют формат: ~ClassName();

Процесс сбора мусора вызовет этот метод автоматически, когда возникнет необходимость в этом.

Язык С# был разработан, чтобы упростить создание компонентов или дополнительных модулей, которые могут быть привязаны к среде разработки программ. Это означает, что можно легко создавать новые элементы управления и легко использовать их как встроенную часть компонентов среды разработки.

С# не поддерживает множественное наследование, в отличие от C++. С# поддерживает интерфейсы похожим на Java образом. Как и в Java, интерфейсы определяют код, который должен быть написан, а не сам повторно используемый код. С # не поддерживает шаблоны или параметризованные типы.

## 2.2. Выбор базы данных

Для хранения данных разрабатываемой информационной системы предназначена база данных. БД представляет собой хранилище огромных массивов информации в связанном виде для совместного использования. Сама по себе БД не может обслужить запросы пользователей на поиск и обработку информации, так как это просто «склад информации» Чтобы эффективно использовать подобный склад, необходима система управления базой данных.

Лучшим вариантом хранения данных является использование реляционной базы данных. Реляционная база данных – база данных, основывающаяся на реляционной модели. Слово «реляционный» произошло от английского слова «relation» (отношение). Для работы с реляционными БД используют реляционные системы управления базами данных. Использовать реляционные базы данных предложил в 1970 году доктор Кодд, работающий в компании IBM.

Реляционная модель организует данные в виде двумерных таблиц. Каждая реляционная таблица, представляющая собой двумерный массив, должна иметь следующие свойства:

1. каждый табличный элемент является только одним элементом данных;
2. все ячейки в столбце таблицы являются однородными – все элементы столбца должны иметь одинаковый тип (INT, FLOAT, VARCHAR и т. д.);
3. каждый столбец обладает уникальным именем, идентифицирующее данный столбец;
4. в таблице не может быть одинаковых строк;
5. порядок следования строк и столбцов произволен.

Почти каждая система управления реляционными БД обеспечивает работу с языком SQL, с помощью которого возможно определение и модификация структуру сведений, добавление, изменение и удаление данных, а также совершение самых разнообразных выборок данных.

Больше всего распространились серверы реляционных БД, которые базируются на клиент-серверной архитектуре. Данными серверами обеспечивается устойчивая работа с базами данных сразу множества клиентов (ими могут являться десятки, сотни, а также тысячи и млн. клиентов – все зависимо от применяемого оборудования и ПО). Помимо того, реляционная модель данных реализуется так называемыми настольными базами данных, к примеру, dBASE, FoxPro, а также Clarion и Paradox, Access. Почти каждая ведущая настольная БД на данный момент поддерживает возможность работать как клиенты серверов БД с помощью технологий ODBC, а также BDE, ADO и других.

Реляционные БД основаны на строгой теории реляционных БД, основывающейся также на теории множеств, а также теории отношений. Реляционная БД является набором таблиц, между которыми имеются заданные связи. Строки таблиц носят название записей, а элементы, которые включены в запись — полей. В теории реляционных БД таблицы носят название отношений, записи — кортежей, а поля — атрибутов.

В таблице реляционной БД не могут содержаться повторяющиеся записи (строки). Данное требование следует из теории множеств. Наименьший набор полей, дающий возможность отличия записи от любой иной записи — ключ. Все значения ключа в рамках таблицы должны являться уникальными. Каждая таблица должна обладать хотя бы одним ключом, что прямо вытекает из того, что в таблице не могут содержаться повторяющиеся записи. Ключи таблицы могут включать одно поле – эти ключи носят название атомарных или простых ключей. Ключи могут включать несколько полей, то есть составные ключи. Таблица БД может обладать одним ключом или несколькими ключами. Один из ключей назначают как первичный ключ, а остальные — потенциальные (в теории реляционных БД) либо альтернативные (в определенных реализациях отдельных БД).

Нередко применяют суррогатный первичный ключ – ключ, включающий поле (или поля), не несущие сведения из предметной сферы, а выступают как замена (суррогат) для естественных (натуральных) первичных ключей. Как суррогатные ключи зачастую применяются счетчики (генераторы, последовательности) autoincrement либо глобально–уникальные идентификаторы (GUID). В правильности созданная структура БД должна отвечать специальным правилам, основанным на теории отношений и называющимся как нормальные формы.

Понятие нормальной формы было введено Эдгаром Коддом при создании реляционной модели баз данных. Нормализация – это процесс преобразования базы данных, к виду, который отвечает нормальным формам и обеспечивает минимальную избыточность. Цель нормализации – защита базы данных от структурных и логических проблем, являющимися аномалиями данных. К примеру, если есть несколько одинаковых записей в таблице, то в этом случае есть риск нарушения целостности данных при последующем обновлении таблицы. Таблица, прошедшая нормализацию, меньше подвергается этим проблемам, поскольку структура такой таблицы определяет связи между данными, таким образом исключается необходимость существования записей с информацией, которая повторяется. Избыточность можно устранить при помощи разбиения отношений (таблиц) таким образом, чтобы в каждом отношении хранились только первичные факты (факты, которые не выводятся из других хранимых фактов). Таким образом, нормализация не ставит цель уменьшения или увеличения производительности работы или же уменьшения, или увеличения объёма базы данных. Конечная цель нормализации – уменьшение потенциальной противоречивости, хранимой в БД информации. Нормализация применима к таблице, которая представляет собой правильное отношение.

Реляционные СУБД бывают системами, имеющих открытый код (OpenSource) и имеющих закрытый. В соответствии с исследованиями фирмы IDC, предприятия из Европы чаще предпочитают открытое программное обеспечение (ПО). Идут на это они ввиду увеличения качества и гибкости этого ПО.

Система управления БД является программным обеспечением, при помощи которого пользователи имеют возможность создания, определения и поддержания базы данных. Специфика СУБД с открытым кодом состоит в том, что они обладают при себе исходным кодом, который может быть изменен, оптимизирован, ускорен за счет того или иного алгоритма. СУБД представляют собой оболочку, с помощью которой после организации структуры таблиц и заполнения их данными формируется определенная БД. Программные средства включают систему управления вводом-выводом, хранение и обработку информации, создание и тестирование БД, трансляторы [10].

Выбор СУБД для реализации определенной задачи – это довольно сложный процесс, который требует навыков, знаний и внимания профессионала в этой области. Неправильный выбор СУБД может заблокировать процесс реализации данной технологии, а замена СУБД может стоить слишком дорого.

СУБД относится к очень сложной части программного обеспечения, которую трудно понять во всей полноте и является жизненно важной для определения критериев, которые могут быть использованы администратором базы данных в процессе их отбора. Многие практики–разработчики баз данных в своих работах отмечают, что выбор критериев использования СУБД заключается в первичном анализе организационных потребностей и необходимых функций СУБД. Практики–разработчики выделяют ряд факторов выбора СУБД для реализации задачи, в частности:

* требования к применению (это ограничение, которые вводятся в базу данных в приложении);
* функции и инструменты СУБД (встроенный набор инструментов, который помогает облегчить задачу разработки приложений);
* модели СУБД (иерархические, сетевые, реляционные, объектно–ориентированные);
* переносимость СУБД (системы и языки программирования, платформы)
* требования СУБД к оборудованию (минимальная скорость процессора, объем оперативной памяти, дискового пространства и т. д.);
* стоимость СУБД (затраты на приобретение и обслуживание, эксплуатационные расходы, стоимость лицензий, затраты на установку, стоимость обучения и преобразования расходов);
* сохранение целостности данных (защита данных в базе данных от несанкционированного доступа, изменения или уничтожения данных).

В качестве СУБД выбрана реляционная база данных MSSQL. Microsoft SQL Server — система управления реляционными базами данных (СУРБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL (T-SQL), создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных, размером от персональных до крупных баз данных масштаба предприятия, успешно конкурирует с другими СУБД. Разработчик СУБД - Microsoft - позиционирует этот продукт в качестве «решения для того, чтобы управлять и анализировать корпоративные данные». MS SQL Server уже многие годы выступает как одна из лучших новейших многопользовательских СУБД, с наиболее оптимальным использованием возможностей ОС Windows и обеспечением высокой масштабируемости и надежности.

Языком реляционных баз данных является SQL. SQL – это язык структурированных запросов (Structured Query Language), первый и единственный язык работы с базами данных, который получил широкое распространение и поддерживается всеми производителями коммерческих реляционных СУБД.

SQL является важнейшим компонентом методологии и технологии реляционных баз данных, неотъемлемой частью реляционной СУБД. Без SQL немыслимы разработка и эксплуатация реляционных баз данных.

В составе СУБД SQL выполняет множество функций. Его можно использовать в интерактивном режиме по принципу «запрос-ответ». SQL является подъязыком баз данных в обстановке прикладной программы, составленной на языке программирования. SQL — это язык запросов в приложениях многопользовательской клиент-серверной системы. SQL — это язык администрирования баз данных.

SQL нельзя отнести к конкретному виду языков. Он содержит в себе возможности и языка реляционного исчисления (исчисления кортежей), и алгебраического языка и, несомненно, является реляционно полным.

В силу исторических причин SQL стал стандартным реляционным языком и в настоящее время поддерживается практически всеми системами баз данных. Поэтому каждый специалист по базам данных должен быть знаком с ним.

Язык SQL является языком декларативного типа. В нем отсутствуют какие-либо команды управления ходом вычислительного процесса типа IF-THEN-ELSE, SWITCH, WHILE, DO-WHILE, FOR, GO TO и др. Управление ходом выполнения процесса обработки данных может выполняться интерактивно, в результате действий самого пользователя или при помощи процедурных языков программирования высокого уровня. В связи с этим различают две разновидности SQL — интерактивный и вложенный. По большей части обе формы работают одинаково, но используются различно.

Интерактивный SQL используется для функционирования непосредственно в базе данных и реализуется непосредственно при вводе пользователем отдельных команд. При такой форме SQL при вводе команды она сейчас же выполнится, и пользователь может увидеть результат немедленно.

Вложенный SQL состоит из команд SQL внедренных внутрь программ, написанных на процедурных языках программирования высокого уровня, что делает программы более мощными и эффективными. Однако при таком способе управления процессом обработки данных приходится иметь дело со структурой SQL и стилем управления, который требует некоторых расширений к интерактивному SQL. На практике существуют два основных способа использования SQL в программах:

* + внедрение SQL-операторов в исходный текст программы с последующей ее компиляцией и компоновкой;
  + использование интерфейса прикладного программирования, специализирующегося на работе с базами данных.

База данных в SQL представляет собой набор взаимосвязанных таблиц. При этом различают базовые таблицы и таблицы-представления. Базовая Таблица (TABLE) — это основной структурный элемент базы данных. Ей соответствуют реальные хранимые данные. Концептуальная модель базы данных представляет собой совокупность взаимосвязанных базовых таблиц. Представление (VIEW) является виртуальной таблицей, которая выглядит как реально существующая Таблица. Представление не содержит собственных данных, в нем скомбинированы данные из одной или нескольких связанных базовых таблиц. Средством наполнения таблицы-представления являются запросы, в которых реализуются информационные потребности пользователей базы данных. С помощью представления реализуются локальные представления (внешняя модель данных) базы данных, используемые в прикладных программах или запросах пользователей.

Средством эффективного доступа к хранимым данным являются индексы (INDEX). При помощи индексов осуществляется доступ к данным, упорядоченным по определенным критериям. В качестве критерия упорядочения могут использоваться один или несколько столбцов таблицы. Наглядным образом индекса является Таблица, первый столбец которой содержит значения индексируемых полей, а второй — ссылки на соответствующие строки. При создании базовых таблиц автоматически создаются индексы по первичному ключу, по альтернативным ключам и по внешним ключам.

Как было написано выше, в качестве СУБД для приложения используется Microsoft SQL Server. MS SQL Server выступает как одна из лучших новейших многопользовательских СУБД, с оптимальным использованием возможностей ОС Windows и обеспечением высокой масштабируемости и надежности.

# Список использованных источников

1. Зашеловский А. Е. Среды разработки программного обеспечения, ориентированные на процессы [Электронный ресурс] / А. Е. Зашеловский, О. Ф. Абрамова. – Режим доступа: <http://www.scienceforum.ru/2015/pdf/15128.pdf>. (дат обращения: 02.12.2019).
2. Зиборов В. Visual C# 2012 на примерах / В. Зиборов. – СПб.: БХВПетербург, 2013. ‒ 473 с.
3. Косенко В. В. Популярные языки программирования с позиций системного программирования / В. В. Косенко // Информационные технологии и вычислительные системы. – 2013. - № 1. – С. 54-59
4. Либерти Д. Программирование на C#. Создание .NET приложений. Программирование на C# / Д. Либерти. – М.: Бином 2010. ‒ 684 с.
5. Рихтер Джеффри. CLR via C# программирование на платформе Microsoft .Net Framework 4.0 на языке C# / Д. Рихтер – Издательство Питер, 3-е издание, 2012. – 928 с.