**ПРЕДИСЛОВИЕ**

В КП вам предстоит написать работу, которая будет занимать около 70 страниц печатного текста. Структура курсового проекта представлена ниже. Написанное в скобках и выделенное бордовым цветом – пометки, в самой структуре их быть не должно:

Введение

1 Предметная область

1.1 Анализ предметной области (1 лабораторная работа)

1.2 Требования, предъявляемые к базе данных

2 Проектирование БД

2.1 Концептуальное проектирование

2.1.1 Определение сущностей (2 лабораторная работа)

2.1.2 Определение связей (2 лабораторная работа)

2.1.3 Определение атрибутов (3 лабораторная работа)

2.1.4 Определение первичных ключей (3 лабораторная работа)

2.1.5 Построение ER-модели (2 лабораторная работа)

2.2 Выбор СУБД

2.2.1 Процесс установки Microsoft SQL Server (6 лабораторная работа)

2.3 Логическое проектирование

2.3.1 Выбор модели данных

2.3.2 Определение таблиц (3 лабораторная работа)

2.3.3 Нормализация (4 лабораторная работа)

2.3.4 Определение требований поддержки целостности (3 лабораторная работа)

3 Физическое проектирование

3.1 Создание БД и таблиц (7-9 лабораторные работа)

3.2 Схема данных (10 лабораторная работа)

3.3 Триггеры (12 лабораторная работа)

3.4 Функции и процедуры (11, 13-14 лабораторные работы)

3.5 Представления (10 лабораторная работа)

4 Администрирование БД

4.1 Создание пользователей и ограничение их прав (7 лабораторная работа)

4.2 Создание резервной копии БД. Создание заданий

Заключение

Список используемых источников

Приложения

Большая часть работы была сделана на протяжении двух семестров. Что-то придется еще изучить и сделать. Вместе с пособием на портале прикреплен архив – данный архив содержит практические примеры, с помощью которых вы сможете выполнить недостающую часть работы и литературу, с помощью которой вы можете заполнить водой этот чудный водоём, под названием «курсовой проект». Главное, при добавлении воды - не захлебнитесь. По ходу методических рекомендаций будет написано, в каком практическом пособии можно посмотреть пример выполнения.

**!** Если вам не нужна оценка «4» и «5» - главы, выделенные зеленым цветом, можно не делать.

**!** Обратите внимание, наличие выделенных зелёным цветом глав оценку «4» или «5» не гарантируют (но без них больше 3 получить невозможно).

Положительная оценка ставится за знания в вашей голове! Поэтому не нужно бездумно копировать чьи-то работы.

В случае, если у вас возникнут вопросы – убедительная просьба не атаковать личку. Сначала попробуйте разобраться самостоятельно. Немного подумайте, напрягите извилинки. Это полезно, честное слово.

Если все равно сложно понять – в этом случае приходим на занятия и задаём свои вопросы по **заранее сформированному списку**.

Когда у вас закончится практика – *ЗАНЯТИЙ НЕ БУДЕТ*. Также, в ваших интересах сделать работу раньше, так как можно получить освобождение от хождения на пары, особенно по субботам это большой плюс, не так ли?)

Вечером по часику выделять и за март курсовая напишется)

Для выделения текста в методичке используется 4 цвета:

* Красный – текст, который нужно заменить по образцу.
* Синий – пояснения.
* Зеленый – рекомендации.
* Черный – текст, который должен быть, менять его не нужно.

**КОМИТЕТ ПО НАУКЕ И ВЫСШЕЙ ШКОЛЕ**

Санкт-Петербургское государственное бюджетное

профессиональное образовательное учреждение

«Санкт-Петербургский технический колледж управления и коммерции»

КУРСОВОЙ ПРОЕКТ

|  |  |
| --- | --- |
| По учебной дисциплине | СУБД |
| На тему | Проектирование и реализация базы данных «Социальная сеть» |

|  |  |  |  |
| --- | --- | --- | --- |
| Студента | Козлова В. П. | | |
|  | Фамилия И.О. | | |
| 3 | курса | 9СК-31 | группы |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Оценка | | |  | | |
| Преподаватель | | | | | |
| Т. И. Курдюмова | | | | | |
| подпись И.О.Фамилия | | | | | |
| « |  | » | |  | 2023 г. |

Санкт-Петербург

2023

|  |
| --- |
| СОДЕРЖАНИЕ  [ВВЕДЕНИЕ 3](#_heading=h.30j0zll)  [1 ПРЕДМЕТНАЯ ОБЛАСТЬ 5](#_heading=h.1fob9te)  [1.1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 5](#_heading=h.3znysh7)  [1.2 ТРЕБОВАНИЯ, ПРЕДЪЯВЛЯЕМЫЕ К БАЗЕ ДАННЫХ 8](#_heading=h.2et92p0)  [2. ПРОЕКТИРОВАНИЕ БД 9](#_heading=h.tyjcwt)  [2.1 КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ 9](#_heading=h.3dy6vkm)  [2.1.1 Определение сущностей 9](#_heading=h.1t3h5sf)  [2.1.2 Определение связей 9](#_heading=h.4d34og8)  [2.1.3 Определение атрибутов 10](#_heading=h.2s8eyo1)  [2.1.4 Определение первичных ключей 10](#_heading=h.17dp8vu)  [2.2 ВЫБОР СУБД 11](#_heading=h.3rdcrjn)  [2.2.1 Процесс установки Microsoft SQL Server 11](#_heading=h.26in1rg)  [2.3 ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ 11](#_heading=h.lnxbz9)  [2.3.1 Выбор модели данных 11](#_heading=h.35nkun2)  [2.3.2 Определение таблиц 12](#_heading=h.1ksv4uv)  [2.3.3 Нормализация 12](#_heading=h.44sinio)  [3 ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ 14](#_heading=h.2jxsxqh)  [3.1 СОЗДАНИЕ БД И ТАБЛИЦ 14](#_heading=h.z337ya)  [3.2 СХЕМА ДАННЫХ 18](#_heading=h.3j2qqm3)  [3.3 ТРИГГЕРЫ 19](#_heading=h.1y810tw)  [3.4 ФУНКЦИИ И ПРОЦЕДУРЫ 19](#_heading=h.4i7ojhp)  [3.5 ПРЕДСТАВЛЕНИЯ 20](#_heading=h.2xcytpi)  [4 АДМИНИСТРИРОВАНИЕ БД 20](#_heading=h.1ci93xb)  [4.1 СОЗДАНИЕ ПОЛЬЗОВАТЕЛЕЙ И ОГРАНИЧЕНИЕ ИХ ПРАВ 20](#_heading=h.3whwml4)  [4.2 СОЗДАНИЕ РЕЗЕРВНОЙ КОПИИ БД. ВОССТАНОВЛЕНИЕ БД 21](#_heading=h.2bn6wsx)  [ЗАКЛЮЧЕНИЕ 22](#_heading=h.qsh70q)  [СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 23](#_heading=h.3as4poj)  [ПРИЛОЖЕНИЯ 24](#_heading=h.1pxezwc) |

ВВЕДЕНИЕ

Целью данной курсовой работы является разработка и создание базы данных предметной области для социальной сети.

База данных — это совокупность структурированных данных, организованных и хранимых на компьютере с целью эффективного доступа, управления и обновления.

База данных может включать в себя различные типы данных, такие как текст, числа, изображения, звук и видео, которые могут быть организованы в таблицы, поля и записи. Эти данные могут быть связаны между собой, что позволяет создавать связи между данными и улучшать их целостность и надежность.

Базы данных могут использоваться для хранения и обработки информации в различных областях, таких как бизнес, наука, государственное управление, здравоохранение и др. Они могут быть использованы для хранения информации о клиентах, заказах, продуктах, инвентаре, научных данных и т.д.

Существуют различные типы баз данных, такие как реляционные, иерархические, сетевые, объектно-ориентированные и др., которые отличаются своей структурой, возможностями и применением.

Создание базы данных включает несколько этапов и зависит от типа базы данных и используемой системы управления базами данных (СУБД).

ТУТ МОЖНО ВСТАВИТЬ ЕЩЕ ВОДЫ

СУБД (Система Управления Базами Данных) - это программное обеспечение, предназначенное для управления базами данных, обеспечивающее доступ к данным, обработку данных и хранение данных на диске.

СУБД позволяет пользователям создавать, изменять и удалять данные в базе данных, а также выполнять запросы для поиска, сортировки, фильтрации и анализа данных. СУБД позволяет пользователям обрабатывать большие объемы данных, сохранять и защищать данные от несанкционированного доступа, обеспечивать целостность данных и поддерживать соответствие со стандартами безопасности и конфиденциальности.

СУБД могут иметь различные типы и характеристики, такие как реляционные, иерархические, сетевые, объектно-ориентированные и др. Каждый тип СУБД имеет свои особенности, структуру и способ хранения данных.

СУБД широко применяются во многих областях, таких как бизнес, здравоохранение, образование, наука и т.д. Все приложения, которые хранят данные, используют некоторый вид СУБД для управления этими данными.

Подводя все вышесказанное, СУБД позволяет:

* Добавлять в таблицу одну или несколько записей
* Удалять из таблицы одну или несколько записей
* Обновлять значения некоторых полей в одной или нескольких записях
* Находить одну или несколько записей, удовлетворяющих заданному условию
* Управлять данными – защита от несанкционированного доступа, поддержка мультипользовательского режима работы с данными и обеспечение целостности и согласованности данных.

Предметом исследования является: База данных социальной сети

Цель исследования: необходимость выбора такой темы диктуется тем, что

Задачи исследования:

1. Анализ предметной области;
2. Составление модели базы данных;
3. Выбор СУБД;
4. Создание базы данных.
5. Администрирование базы данных.

Разработка и реализация базы данных позволит анализировать поведение пользователей: их активность, привычки, мысли, предпочтения и другие данные, которые могут быть полезны при различных психологических исследованиях. Еще одной причиной для создания является необходимость дать пользователям дать возможность отслеживать деятельность своих друзей, знакомых, находить новых пользователей. Словом, социализировать и повысить коммуникабельность.

ДОПИСАТЬ ЗАЧЕМ Я СОЗДАЮ БД?

Изучение данного вопроса проводилось с помощью научной литературы, сравнения и анализа информации предметной области.

В ходе исследований применялись ранее полученные теоретические знания, а также материал из интернета и научной литературы.

Результаты исследований позволят совершенствовать работу компании «DeepNet» за счет внедрения базы данных.

Структура работы соответствует логике исследования и включает в себя введение, теоретическую часть, практическую часть, заключение, список литературы.

1 ПРЕДМЕТНАЯ ОБЛАСТЬ

Предметная область — это сфера деятельности, которая описывает некоторую проблемную область или конкретную предметную область, для которой разрабатывается система или приложение. Например, предметная область может быть связана с финансовой отчетностью, здравоохранением, автомобильной промышленностью, онлайн-торговлей и т. д.

Анализ предметной области - это процесс изучения предметной области, включающий в себя выявление и анализ ее основных характеристик, свойств и функций. Целью анализа предметной области является понимание основных требований к системе или приложению, которые должны быть учтены при проектировании базы данных и разработке программного обеспечения.

Анализ предметной области включает в себя нижеследующие пункты:

* + - 1. Изучение бизнес-процессов и требований пользователя: Целью является понимание того, какие задачи и функции должна выполнять система или приложение, какие данные и ресурсы необходимы для этих задач, а также какие ограничения и требования к безопасности и конфиденциальности должны быть учтены.
      2. Выявление сущностей и их связей: Целью является выявление всех объектов (сущностей) в предметной области и их связей между собой, которые могут быть представлены в базе данных. Это важно для определения структуры базы данных и связей между таблицами.
      3. Определение атрибутов и свойств: Целью является определение атрибутов (характеристик) каждой сущности и их свойств, которые необходимы для описания объекта в базе данных. Это важно для определения полей таблицы базы данных.
      4. Описание бизнес-правил и правил целостности данных: Целью является определение всех бизнес-правил и правил целостности данных, которые должны быть учтены при разработке базы данных и приложения. Например, бизнес-правило может требовать, чтобы определенные поля были заполнены, а правило целостности может требовать, чтобы определенные связи между сущностями были сохранены в базе данных.

1.1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Компания «DeepNet» представляет Вам сеть нового поколения.

Если Вы знаете социальную сеть Vk (vk.com), то вы сразу же почувствуете себя в своей тарелке, ибо принцип дизайна практически идентичен. Но, хочется уточнить, что все же есть некоторые различия. К примеру, у нас есть кастомизация любых частей сервиса под нужды и желания пользователей.

Что это значит? Это значит, что у нас есть мастерская, куда пользователи могут загружать свои работы (дизайны). ты тоже можешь загрузить свою работу туда. А все, потому что у нашей соцсети есть открытый исходный код на github.com, что является еще одним абсолютным преимуществом нашей новой социальной сети. Это значит, что мы не воруем Ваши данные, и не продаем информацию на рынках. Безопасность и приватность клиентов – наш принцип работы.

Если Вы хороший программист, то Вы сможете это проверить и оценить, если Вы обычный рядовой пользователь, Вы будете слушать, что говорят хорошие программисты. А они будут хвалить нас.

У нас Вы можете:

Загружать файлы любого типа с ограничением до 1 гб

Слушать любую музыку, ну и, конечно, добавлять свою

Иметь свой профиль

Вступать в группы (каналы) по интересам, а также их создавать

Создавать беседы для общения с большим кол-вом людей

Писать сообщения другим пользователям

Делиться фотографиями

Лайкать посты и писать комментарии к посту

Для того, чтобы Вы могли воспользоваться всеми данными преимуществами, над данным проектом трудилось и трудится огромное кол-во людей: высококлассные программисты, специалисты по информационной и компьютерной безопасности, бухгалтеры, маркетологи, знаменитости, аналитики, контент-менеджеры, UX/UI дизайнеры

Наши программисты всегда ищут способы оптимизировать сервис и добавить новые фишки, а специалисты по безопасности постоянно обновляют методы защиты данных пользователей, тестируют систему на устойчивость перед атаками и попытками взлома.

Маркетолог - специалист по продвижению продукта на рынке, который может разрабатывать маркетинговые стратегии, исследовать целевую аудиторию, проводить анализ конкурентов и создавать рекламные кампании.

UX/UI дизайнер - дизайнер пользовательского интерфейса и опыта пользователя, который может создавать привлекательный и удобный интерфейс для социальной сети, обеспечивая удобство и доступность использования для пользователей.

Разработчик - специалист по созданию программного обеспечения, который может разрабатывать и поддерживать функциональность социальной сети, обеспечивая ее стабильную работу и безопасность.

Контент-менеджер - специалист по созданию контента, который может создавать и публиковать контент на социальной сети, обеспечивая ее актуальность и интересность для пользователей.

Аналитик - специалист по анализу данных, который может анализировать статистику использования социальной сети, оценивать эффективность маркетинговых кампаний и разрабатывать стратегии для повышения роста активности пользователей.

У нас работает очень много талантливых людей!

Еще одним несомненным плюсом будет то, что ты тоже можешь внести что-то новое в наш проект. Это значит, что ты можешь устроиться работать в нашу компанию. Даже если ты только студент, и не имеешь опыта работы, мы все равно тебя можем трудоустроить, помочь и все объяснить. Потому что мы понимаем, что будущее именно за молодым поколением, поэтому мы даем возможность именно тебе реализовать свои возможности и потенциал. Возможно, начав помогать нашей компании, ты повлияешь на весь мир!

1.2 ТРЕБОВАНИЯ, ПРЕДЪЯВЛЯЕМЫЕ К БАЗЕ ДАННЫХ

Правильно спроектированная база данных должна удовлетворять следующим требованиям:

1. Целостность данных: Данные должны быть целостными и точными, что означает, что они должны соответствовать бизнес-правилам и ограничениям, заданным для предметной области. Например, если база данных содержит информацию о заказах и клиентах, то каждый заказ должен быть привязан к соответствующему клиенту, чтобы гарантировать целостность данных.
2. Эффективность: База данных должна быть эффективной в использовании ресурсов и обеспечивать быстрый доступ к данным. Например, база данных должна быть оптимизирована для выполнения запросов и поиска данных.
3. Безопасность: База данных должна быть защищена от несанкционированного доступа и должна обеспечивать конфиденциальность и целостность данных. Например, база данных должна быть защищена паролями, шифрованием и другими мерами безопасности.
4. Гибкость: База данных должна быть гибкой и расширяемой, что означает, что она должна позволять быстро и легко вносить изменения в структуру данных и связи между таблицами. Например, если в бизнес-процессах возникают новые требования, база данных должна легко адаптироваться к этим изменениям.
5. Простота использования: База данных должна быть легко понятной и простой в использовании. Например, она должна иметь понятные и легко доступные интерфейсы для ввода и вывода данных, а также быть удобной для поиска и обработки информации.
6. Надежность: База данных должна быть надежной и устойчивой к сбоям, что означает, что она должна иметь механизмы резервного копирования и восстановления данных.

2. ПРОЕКТИРОВАНИЕ БД

Проектирование базы данных (БД) — это процесс создания и определения структуры, хранения, организации и управления данными в базе данных. Проектирование БД является важной частью разработки программного обеспечения и выполняется перед созданием физической БД.

Существует 3 вида процедур проектирования: логическое, физическое и концептуальное.

Рассмотрим все этапы проектирования:

Планирование разработки БД

Определение требований к системе

Сбор и анализ требований пользователей

Проектирование БД

Концептуальное проектирование БД

Логическое проектирование БД

Физическое проектирование БД

Разработка применений

Реализация

Загрузка данных

Тестирование

Эксплуатация и сопровождение

2.1 КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ

Концептуальное проектирование — это первый этап процесса проектирования базы данных. Он охватывает определение основных сущностей и связей между ними, которые будут храниться в базе данных. Цель концептуального проектирования - создание высокоуровневой модели данных, которая описывает структуру базы данных без учета специфических деталей реализации.

Концептуальное проектирование базы данных основывается на анализе предметной области, в которой будет использоваться база данных, и учете требований пользователей. В результате этого процесса создается диаграмма ER (Entity-Relationship), которая отображает сущности, их атрибуты и связи между ними.

Этапы:

1. Сбор информации о предметной области.
2. Определение сущностей и их документирование.
3. Создание ER-модели предметной области.
4. Определение атрибутов и их документирование
5. Определение значений атрибутов и их документирование
6. Определение первичных ключей для сущностей и их документирование.
7. Обсуждение концептуальной модели данных с конечными пользователями

2.1.1 Определение сущностей

На основе вышеописанной предметной области можно составить перечень сущностей компании ООО «DeepNet»:

* Сущность «Пользователь» хранит всю информацию о пользователях;
* Сущность «Пост» хранит все записи о постах, которые были когда-либо созданы;
* Сущность «Комментарий» содержит всю информацию о комментарии, который оставляют пользователи;
* Сущность «Лайк» содержит информацию о лайках, которые ставят пользователи;
* Сущность «Сообщество» содержит информациях о сообществах;
* Сущность «Участник сообщества» содержит информацию об участниках сообщества;
* Сущность «Беседа» содержит информацию о беседах;
* Сущность «Участник беседы» содержит информацию об участниках беседы;
* Сущность «Файл» содержит информацию о файлах внутри данной социальной сети;
* Сущность «Сообщение» содержит информацию о сообщениях, отправленных пользователями;

2.1.2 Определение связей

Существует 3 типа связи между сущностями:

1. Один к одному (One-to-One): каждая запись в первой таблице связана с одной и только одной записью во второй таблице, и наоборот. Такая связь часто используется, когда две таблицы имеют схожие атрибуты, но могут быть разделены по логическим причинам.
2. Один ко многим (One-to-Many): каждая запись в первой таблице связана с несколькими записями во второй таблице, но каждая запись во второй таблице связана только с одной записью в первой таблице. Этот тип связи является наиболее распространенным в проектировании баз данных.
3. Многие ко многим (Many-to-Many): каждая запись в первой таблице может быть связана с несколькими записями во второй таблице, и наоборот. Такие связи реализуются с помощью дополнительной таблицы, которая содержит первичные ключи обеих таблиц.

В данной работе будут использованы следующие типы связей:

1. Пользователь создает пост (рисунок 1).

Изображение выглядит как часы

Автоматически созданное описание

Рисунок 1 – связь между сущностями Пользователь и Пост

1. Пользователь имеет комментарий (рисунок 2).

Изображение выглядит как часы

Автоматически созданное описание

Рисунок 2 – связь между сущностями Пост и Комментарий

1. Пользователь добавляет (пишет) файл (рисунок 3).

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 3 – связь между сущностями Пользователь и Файл

1. Пользователь ставит лайк (рисунок 4).

Изображение выглядит как часы

Автоматически созданное описание

Рисунок 4 – связь между сущностями Пользователь и Лайк

1. Пользователь имеет комментарий (рисунок 5).

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 5 – связь между сущностями Пост и Комментарий

1. Пользователь является участником беседы (рисунок 6).

Изображение выглядит как часы

Автоматически созданное описание

Рисунок 6 – связь между сущностями Пользователь и Участник беседы

1. Беседа имеет участников беседы (рисунок 7).

Изображение выглядит как часы

Автоматически созданное описание

Рисунок 7 – связь между сущностями Беседа и Участник беседы

1. Участник беседы получает сообщение (рисунок 8).

Изображение выглядит как седзи

Автоматически созданное описание

Рисунок 8 – связь между сущностями Участник беседы и Сообщение

1. Пользователь является участником сообщества (рисунок 9).

Изображение выглядит как текст, часы

Автоматически созданное описание

Рисунок 9 – связь между сущностями Пользователь и Участник сообщества.

1. Сообщество имеет участников сообщества (рисунок 10).

Изображение выглядит как часы

Автоматически созданное описание

Рисунок 10 – связь между сущностями Сообщество и Участник сообщества.

1. Пост имеет лайк (рисунок 11).

Изображение выглядит как часы

Автоматически созданное описание

Рисунок 11 – связь между сущностями Пост и Лайк.

2.1.3 Определение атрибутов

Для каждой сущности необходимо выделить атрибуты, которые они будут содержать.

Таблица 1 – Атрибуты сущности «Участник сообщества»

|  |  |
| --- | --- |
| Наименование атрибута | Описание атрибута |
| Id пользователя | Идентификатор, по которому можно понять, какой пользователь учувствует в связке сообщество-пользователь |
| Id сообщества | Идентификатор, по которому можно понять к какому сообществу принадлежит пользователь |

Таблица 2 – Атрибуты сущности «Сообщество»

|  |  |
| --- | --- |
| Наименование атрибута | Описание атрибута |
| Id сообщества | Идентификатор, по которому можно найти и определить сообщество |
| Аватарка | Картинка, которую можно установить для сообщества |
| Название сообщества | Название, которое устанавливает пользователь для придания индивидуальности сообщества и выражения основной тематики сообщества |
| Описание сообщества | Поясняющий текст для сообщества. К примеру, чем оно занимается, зачем было создано |
| Дата создания | Показывает, когда было создано сообщество |

Таблица 3 – Атрибуты сущности «Беседа»

|  |  |
| --- | --- |
| Наименование атрибута | Описание атрибута |
| Id беседы | Идентификатор, по которому можно найти и определить беседу |
| Аватарка | Картинка, которую можно установить для беседы |
| Название беседы | Название, которое устанавливает пользователь для придания индивидуальности беседы и выражения основной темы беседы |

Таблица 4 – Атрибуты сущности «Пользователь»

|  |  |
| --- | --- |
| Наименование атрибута | Описание атрибута |
| Id пользователя | Идентификатор, по которому можно определить пользователя, и посмотреть информацию о нем |
| Аватарка | Картинка, которую пользователь может установить для своего профиля |
| Имя | Имя, которое захочет использовать пользователь |
| Фамилия | Фамилия, которую можно использовать пользователь |
| Адрес электронной почты | Адрес почты, который пользователь для авторизации и получения уведомлений |
| Хэш пароля | Хаотичный набор символов и чисел, в которые зашифрован пароль пользователя |
| Дата регистрации | Дата регистрации пользователя |

Таблица 5 – Атрибуты сущности «Файл»

|  |  |
| --- | --- |
| Наименование атрибута | Описание атрибута |
| Id файла | Идентификатор, по которому можно обратиться к определенному файлу и узнать информацию о нем |
| Id пользователя | Идентификатор пользователя, который показывает кто именно загрузил файл |
| Имя файла | Имя файла |
| Расширение файла | Расширение, в котором сохранен файл |
| Адрес электронной почты | Название тип файла текстом, чтобы быстро понять к какому типу он относится |
| Размер файла | Размер файла |
| Дата загрузки | Когда был загружен файл |

Таблица 6 – Атрибуты сущности «Комментарий»

|  |  |
| --- | --- |
| Наименование атрибута | Описание атрибута |
| Id пользователя | Идентификатор пользователя, который показывает кто именно написал комментарий |
| Id поста | Идентификатор поста, который показывает к какому посту относится комментарий |
| Текст комментария | Текст комментария |
| Дата публикации | Когда был написан и опубликован комментарий |

Таблица 7 – Атрибуты сущности «Участник беседы»

|  |  |
| --- | --- |
| Наименование атрибута | Описание атрибута |
| Id пользователя | Идентификатор пользователя, который показывает кто именно является участником беседы |
| Id беседы | Идентификатор беседы, который показывает к какой именно беседе относится пользователь |

Таблица 8 – Атрибуты сущности «Лайк»

|  |  |
| --- | --- |
| Наименование атрибута | Описание атрибута |
| Id пользователя | Идентификатор пользователя, который показывает кто именно поставил лайк |
| Id поста | Идентификатор поста, который показывает к какому именно посту был поставлен лайк пользователем |

Таблица 9 – Атрибуты сущности «Пост»

|  |  |
| --- | --- |
| Наименование атрибута | Описание атрибута |
| Id поста | Идентификатор поста, по которому можно определить пост |
| Id пользователя | Идентификатор пользователя, который показывает, какой именно пользователь создал пост |
| Текст поста | Текст комментария |
| Дата публикации | Когда был создан и опубликован пост |

Таблица 10 – Атрибуты сущности «Сообщение»

|  |  |
| --- | --- |
| Наименование атрибута | Описание атрибута |
| Id сообщения | Идентификатор поста, по которому можно определить сообщение |
| Id пользователя | Идентификатор пользователя, который показывает, кто именно написал сообщение |
| Id беседы | Идентификатор беседы, который показывает, в какую именно беседу было отправлено сообщение |
| Текст поста | Текст сообщения |
| Дата публикации | Когда было создано сообщение |

2.1.4 Определение первичных ключей

Первичный ключ (Primary key) – это уникальный идентификатор, используемый в базах данных для идентификации и связывания отдельных записей в таблице. Первичный ключ должен быть уникальным для каждой записи в таблице и не может содержать значений NULL (пустых значений).

Как правило, первичный ключ состоит из одного или нескольких столбцов в таблице, и он может быть создан автоматически или задан пользователем при создании таблицы. Первичный ключ обеспечивает быстрый доступ к данным и упрощает выполнение операций обновления, вставки и удаления записей в таблице.

Первичный ключ необходим в базах данных для нескольких целей:

* + - 1. Уникальность записей: Первичный ключ гарантирует уникальность каждой записи в таблице. Это позволяет предотвратить дублирование данных и обеспечивает целостность данных в базе данных.
      2. Связь между таблицами: Первичный ключ может использоваться для связывания данных в разных таблицах базы данных. Например, ключевое поле "идентификатор сотрудника" может быть использовано для связывания записей с таблицей "отделы", чтобы определить, в каком отделе работает каждый сотрудник.
      3. Ускорение поиска данных: Использование первичного ключа позволяет быстро находить конкретную запись в таблице. Без первичного ключа база данных должна была бы сканировать всю таблицу, чтобы найти нужную запись, что приводит к замедлению производительности базы данных.
      4. Облегчение выполнения операций изменения данных: Первичный ключ облегчает выполнение операций вставки, обновления и удаления данных в таблице. Без первичного ключа база данных может случайно обновить несколько записей, что может привести к некорректным результатам.

Для сущности «Участник сообщества» в качестве первичных ключей были выбраны атрибуты «Код пользователя» и «Код сообщества».

Для сущности «Сообщество» в качестве первичного ключа был выбран атрибут «Код сообщества».

Для сущности «Участник беседы» в качестве первичных ключей были выбраны атрибуты «Код беседы» и «Код пользователя».

Для сущности «Беседа» в качестве первичного ключа был выбран атрибут «Код беседы».

Для сущности «Файл» в качестве первичного ключа был выбран атрибут «Код файла».

Для сущности «Сообщение» в качестве первичного ключа был выбран атрибут «Код сообщения».

Для сущности «Пост» в качестве первичного ключа был выбран атрибут «Код поста».

Для сущности «Пользователь» в качестве первичного ключа был выбран атрибут «Код пользователя».

Для сущности «Комментарий» в качестве первичных ключей были выбраны атрибуты «Код пользователя» и «Код поста».

Для сущности «Лайк» в качестве первичных ключей были выбраны атрибуты «Код поста» и «Код пользователя».

2.1.5 Построение ER-модели

Исходя из полученных связей, была создана ER-диаграмма, представленная на рисунке 12.

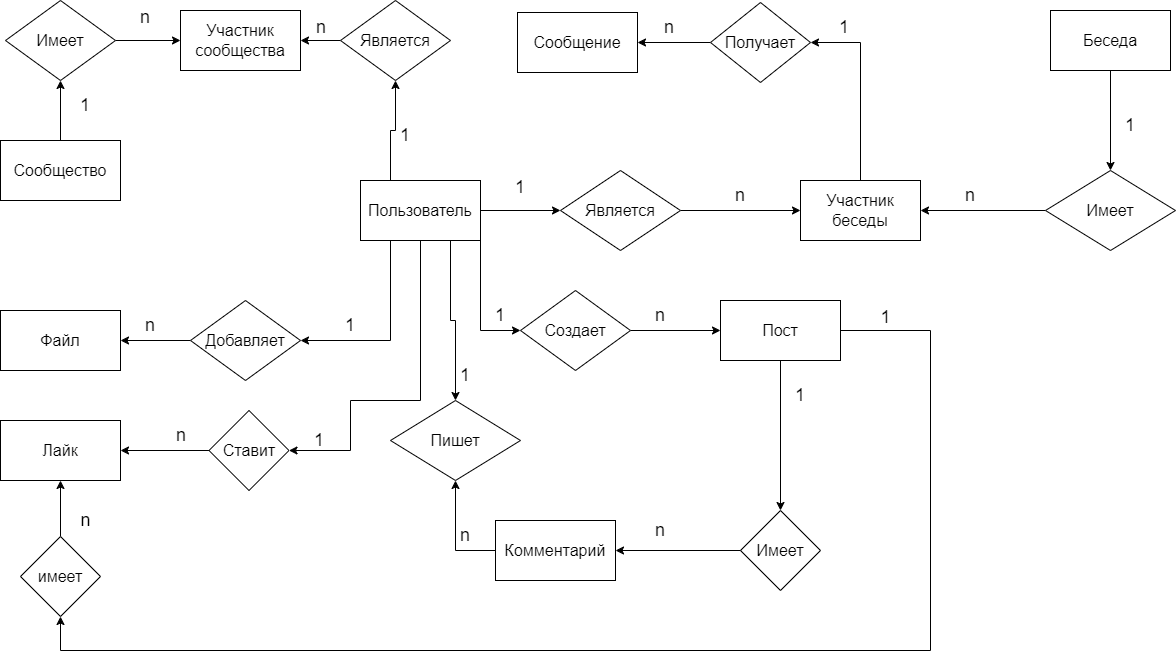


Рисунок 12 – Концептуальная модель.

2.2 ВЫБОР СУБД

СУБД (Система Управления Базами Данных) — это комплекс программных и аппаратных средств, предназначенных для создания, управления и использования баз данных. СУБД обеспечивает множество функций для хранения, организации, поиска, обновления и анализа данных, а также для обеспечения их безопасности и целостности.

Реляционные СУБД – это наиболее распространенный тип СУБД. Реляционные СУБД ориентированы на работу с реляционными базами данных, которые хранят данные в виде таблиц, связанных между собой по ключевым полям. Примерами реляционных СУБД являются Oracle, MySQL, PostgreSQL, Microsoft SQL Server и другие.

Для работы с базами данными я выбрал СУБД Microsoft SQL Server. Коротко пройдемся по минусам и плюсам данной СУБД.

Плюсы:

1. Функциональность – широкий набор функций, включая инструменты для аналитики, обработки данных и отчетности.
2. Безопасность - широкий набор инструментов для обеспечения безопасности данных.
3. Производительность - Microsoft SQL Server имеет высокую производительность и может обрабатывать большие объемы данных с большим количеством пользователей.
4. Интеграция – данная СУБД может интегрироваться с другими продуктами Microsoft, что позволяет удобно пользоваться им в экосистеме Microsoft
5. Поддержка – широкое сообщество разработчиков и большая поддержка со стороны Microsoft

Минусы:

Цена: Microsoft SQL Server является коммерческим продуктом и требует приобретения соответствующей лицензии

Ограничения на платформу – работает только на ОС Windows.

Сложность: может быть сложным для настройки и управления, особенно для непрофессионалов

Требования к высокопроизводительной конфигурации устройства, на которое будет устанавливаться данный сервер.

2.2.1 Процесс установки Microsoft SQL Server

Порядок действий для установки Microsoft SQL Server

1. Перейти по указанной ниже ссылке, и скачать файл установки

<https://go.microsoft.com/fwlink/?linkid=866662>

1. При открытии Вы увидите то, что показано на рисунке 13.

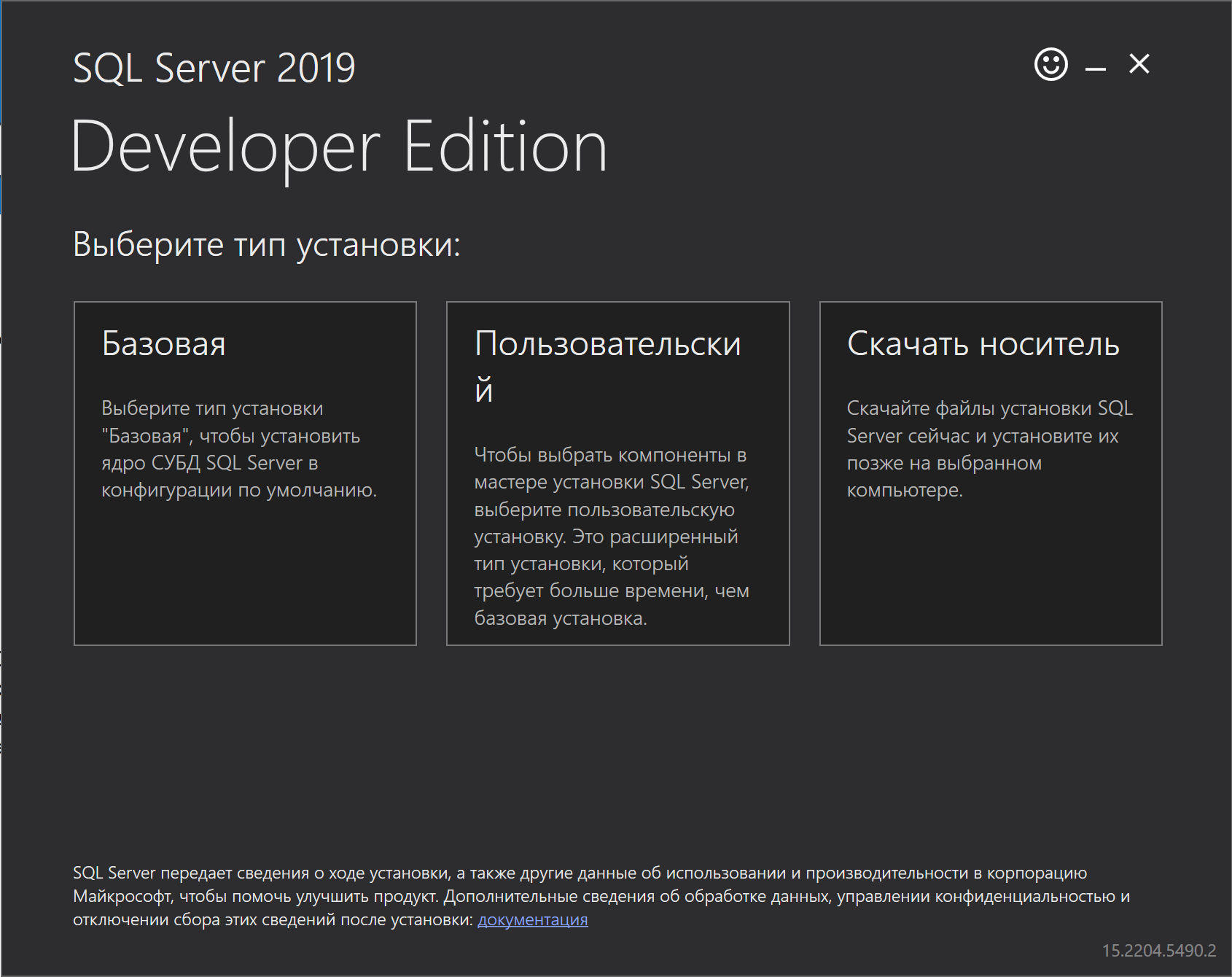


Рисунок 13 – начальное меню установки.

1. Выбрав базовый тип установки, мы видим это (рисунок 14):

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 14 – условия лицензии.

1. Подтвердив, мы выбираем место установки – рисунок 15.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 15 – выбор расположения SQL Server.

1. Далее начнется установка. В течение 5 минут Вы увидите такое окно, как на рисунке 16.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 16 – установленная программа.

Далее нам надо среда установить SQL Server Management Studio (SSMS). SSMS – это основной инструмент, с помощью которого осуществляется разработка баз данных в Microsoft SQL Server.

В окне программы нажать «Установить SSMS» (рисунок 16). После откроется веб-страница, которая будет выглядеть, как на рисунке 17. На ней следует нажать на текст «Бесплатное скачивание SQL Server Management Studio (SSMS) 18.12.1».

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 17 – скачивание программы с сайта.

Далее, открывая эту программу, мы видим, что нам снова предлагают указать путь. Я выбрал тот же. Это видно на рисунке 18.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 18 – выбор места установки SSMS.

Ожидая некоторый короткий срок времени, мы видим картинку, как на рисунке 19.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 19 – установленная программа.

Установленная программа для работы с БД находится по запросу «management». Тогда Вы увидите такие же результаты, как на рисунке 20.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 20 – найденная программа.

После запуска мы можем начать выполнять действия по проектированию баз данных – рисунок 21.

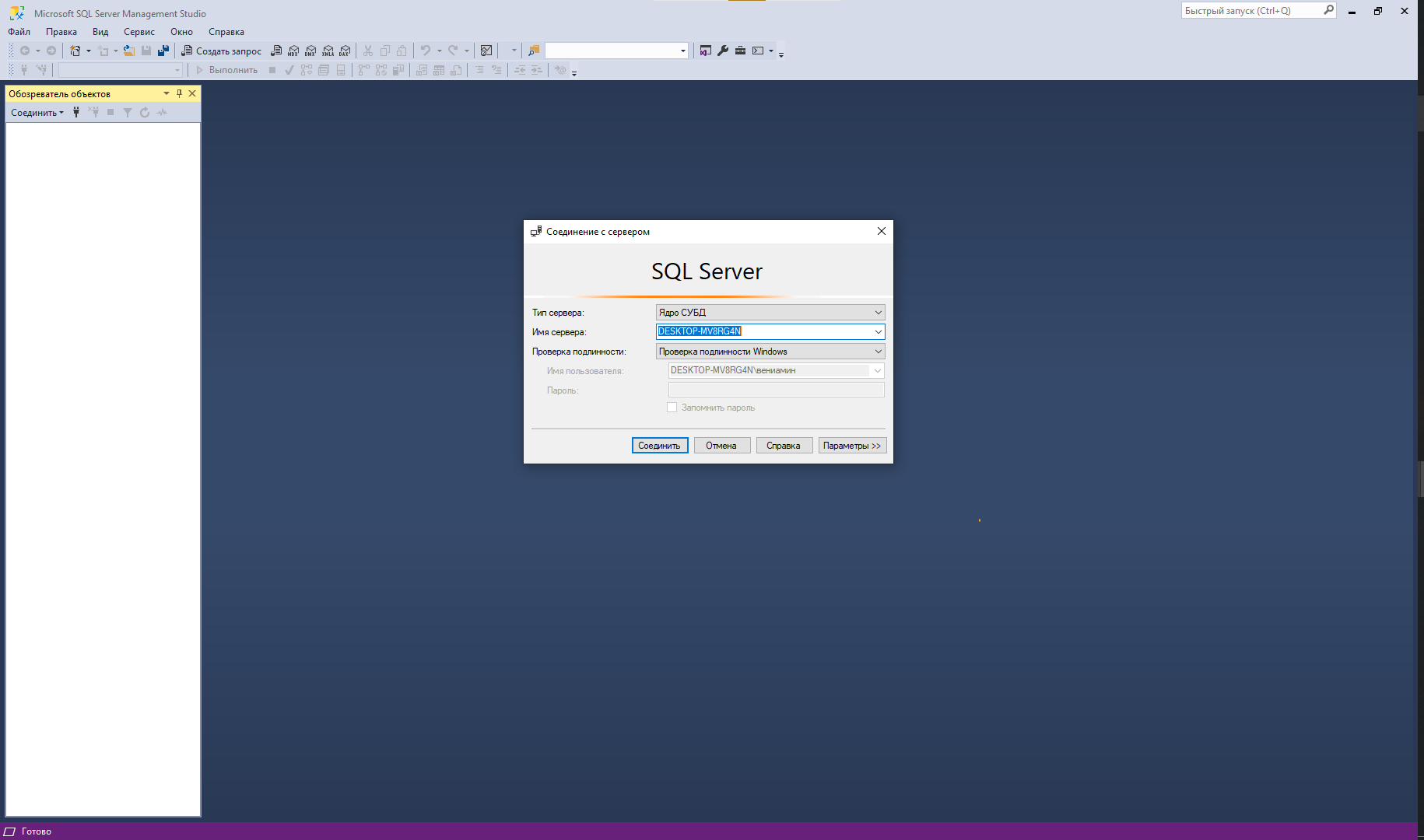


Рисунок 21 – открытая программа.

2.3 ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

Логическое проектирование — это процесс проектирования базы данных на более высоком уровне абстракции, чем физическое проектирование. В процессе логического проектирования определяются таблицы, связи и атрибуты, которые будут использоваться для хранения и управления данными в базе данных.

Результатом логического проектирования является описание структуры данных и их отношений в базе данных, представленное в виде логической модели данных. Логическая модель данных определяет таблицы, связи между таблицами, атрибуты и ограничения целостности данных. Она описывает, как данные будут храниться и использоваться в базе данных, но не содержит информацию о том, как данные будут физически реализованы в базе данных. Она служит основой для дальнейшего физического проектирования базы данных, то есть создания таблиц, индексов и других объектов базы данных на конкретной платформе. Также логическая модель данных может быть использована в качестве документации для программистов, которые будут работать с базой данных, и для пользователей, которые будут использовать данные, хранящиеся в базе данных.

Этапы:

1. Определение сущностей: определение всех объектов, которые будут храниться в базе данных. Например, если база данных используется для учета клиентов, то сущностями могут быть Клиенты, Заказы, Продукты и т.д.
2. Определение связей: определение отношений между сущностями. Например, Клиенты могут иметь несколько Заказов, а каждый Заказ может иметь несколько Продуктов.
3. Определение атрибутов: определение всех характеристик каждой сущности, которые будут храниться в базе данных. Например, для сущности Клиенты это могут быть Имя, Фамилия, Адрес и т.д.
4. Нормализация: процесс устранения дублирующейся информации в базе данных и уменьшения избыточности данных.
5. Создание схемы базы данных: разработка конкретной структуры базы данных, которая определяет таблицы, их связи, атрибуты и ограничения, такие как первичные ключи, внешние ключи и т. д.

2.3.1 Выбор модели данных

Существует 5 видов моделей данных:

1. Реляционная модель данных — это наиболее распространенная модель данных, используемая в коммерческих и научных приложениях. В реляционной модели данные хранятся в виде таблиц, которые состоят из строк (записей) и столбцов (атрибутов). Связи между таблицами устанавливаются посредством первичных и внешних ключей
2. Иерархическая модель данных - в этой модели данные организованы в виде иерархии, состоящей из родительских и дочерних узлов. Каждый узел может иметь несколько дочерних узлов, но только одного родительского узла.
3. Сетевая модель данных - в этой модели данные организованы в виде графа, состоящего из узлов и связей между ними. Каждый узел может иметь несколько связей с другими узлами.
4. Объектно-ориентированная модель данных - в этой модели данные организованы в виде объектов, каждый из которых может иметь свойства и методы. Связи между объектами устанавливаются посредством ссылок.
5. NoSQL-модели данных — это группа различных моделей данных, которые не используют реляционную модель. Они используются для работы с большими объемами данных и обеспечивают гибкость в обработке данных. Примеры таких моделей включают документно-ориентированные, ключ-значение, графовые и колоночные базы данных.

В моей курсовой работе, при проектировании, мне следует воспользоваться реляционной базой данных, так как мне надо определять уникальные данных для пользователей, файлов и других сущностей

2.3.2 Определение таблиц

Каждое реляционное отношение соответствует одной сущности и в него вносятся все атрибуты этой сущности. Для каждого отношения определяются первичный ключ и внешние ключи. (в соответствии со схемой БД)

Ниже представлены все сущности и атрибуты:

Таблица N – Таблица «Клиенты»

|  |  |  |  |
| --- | --- | --- | --- |
| Код клиента | Первичный ключ | Идентификатор | число (int) |
| Телефон | Зависит от кода клиента | Содержит телефонный номер клиента | число |
| Паспортные данные | Зависит от кода клиента | Содержит паспортные данные клиента | число |

Таблица N – Таблица «Товары»

|  |  |  |  |
| --- | --- | --- | --- |
| Код товара | Первичный ключ | Идентификатор | число |
| Название | Зависит от кода товара | Содержит название товара | текст (char(30) |
| Описание | Зависит от кода товара | Содержит описание товара | текст |

2.3.3 Нормализация

\*И что же такое эта «нормализация»? Я слышала про какие-то нормальные формы. Кто их придумал? И сколько их? И что, прям все эти формы используются на практике?

А как узнать, какая это нормальная форма? И как вообще привести к нормальной форме? Тут опять теория. Теории много не бывает.

Дальше вставляем процесс преобразования к 3 НФ нескольких таблиц, как пример, что убрали, что разделили, почему, что получилось в итоге. НЕ ВСЕ, НЕСКОЛЬКО ПРИМЕРОВ. Это было в 3 лабораторной работе.

После приведения примеров необходимо сказать, что все они в 3 нормальной форме, а также упомяните о требованиях поддержки целостности. Кстати, что это такое?\*

Полученные таблицы были объединены в логическую модель, представленную на рисунке N (из 6 лабораторной работы).



Рисунок 1 – Логическая модель

Сделайте схемы красивыми, читабельными. Кроме того, проверьте, действительно ли все таблицы находятся в 3НФ. Может вы что-то упустили?

3 ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

\*Что такое физическое проектирование? Что получается в результате?\*

3.1 СОЗДАНИЕ БД И ТАБЛИЦ

\*Как создать БД? Тут часть оставляем из того, что я вставила, заменяем данные на свои (8 лабораторная работа)\*

В Microsoft SQL Server существует два способа создания БД: графический и с помощью SQL запроса на языке T-SQL. В рамках данного курсового проекта была создана БД при помощи запроса, представленного в листинге 1. Результат работы запроса показан на рисунках N-M.

Листинг 1 – Запрос, создающий БД «Название БД»

|  |
| --- |
| USE NAME\_1  CREATE DATABASE name;  GO |

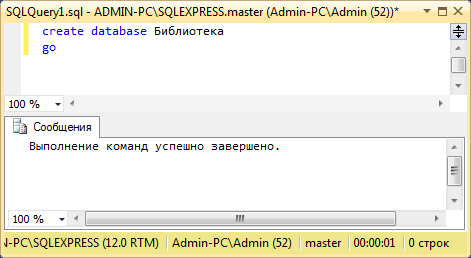


Рисунок N – Результат выполнения запроса на создание БД «Название БД»

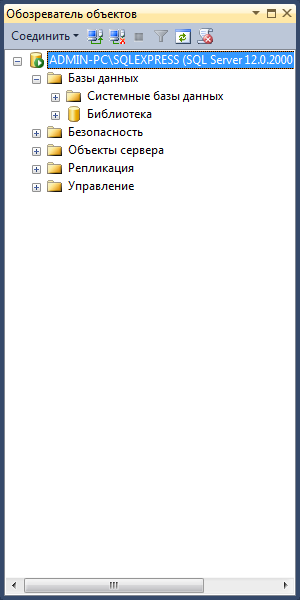


Рисунок N – Созданная БД «Название БД» в Microsoft SQL Server

Также были созданы таблицы при помощи запросов, как показано в листинге 2. \*тут вставляем запрос из 8 лабораторной работы\*

Листинг 2 – Запрос, создающий таблицы «Название таблицы 1», «Таблицы 2»

|  |
| --- |
| use Библиотека  GO    CREATE TABLE [dbo].[Читатель] (  [Код читателя] [int] NOT NULL ,  [Фамилия] [varchar] (20) NOT NULL ,  [Имя] [varchar] (20) NOT NULL ,  [Домашний адрес] [varchar] (20) NOT NULL ,  [Номер паспорта] [int] NOT NULL ,  [Телефон] [varchar] (10) NULL ,  ) ON [PRIMARY]  GO    CREATE TABLE [dbo].[Книги] (  [Код книги] [int] NOT NULL ,  [Название книги] [varchar] (50) NOT NULL ,  [Автор] [varchar] (50) NOT NULL ,  [Год издания] [smallint] NOT NULL ,  [Число страниц] [real] NOT NULL ,  [Цена] [money] NULL,  [Код издательства] [int] NOT NULL  ) ON [PRIMARY]  GO |

Затем, в каждой из таблиц был назначен первичный ключ. Сделать это можно как графическим способом, так и при помощи запроса. В листинге 3 представлен запрос, назначающий первичный ключ в таблицах «Таблица 1», «Таблица 2».

Листинг 3 – Запрос, назначающий первичные ключи для таблиц «Название таблицы 1», «Таблицы 2»

|  |
| --- |
| use Библиотека  GO  ALTER TABLE [dbo].[Читатель] WITH NOCHECK ADD  CONSTRAINT [PK\_Читатель] PRIMARY KEY NONCLUSTERED  (  [Код читателя]  ) ON [PRIMARY]  GO    ALTER TABLE [dbo].[Книги] WITH NOCHECK ADD  CONSTRAINT [PK\_Книги] PRIMARY KEY NONCLUSTERED  (  [Код книги]  ) ON [PRIMARY]  GO |

Для того, чтобы заполнить таблицу данными можно использовать запрос, показаный в листинге 4 или открыть таблицу в режиме заполнения и заполнить ее вручную (тут текст из 8 лабы).

Листинг 4 – Запрос, заполняющий таблицы «Название таблицы 1», «Название таблицы 1»

|  |
| --- |
| USE Библиотека  GO  INSERT INTO [Книги] VALUES  ('1', 'Мастер и Маргарита', 'Н.В. Гоголь', '1967', '504', '1500')  INSERT INTO [Читатель] VALUES  ('123', 'Иванов', 'Иван', 'ул.Лесная,д.21,кв708', '0814754585', '789845654') |

Таким образом было создано и заполнено 10 таблиц, представленных на рисунках N-M (тут скрины из 7 лабы).

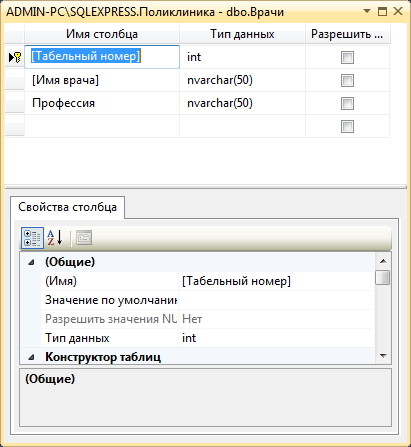


Рисунок 1 – Таблица «Врачи» в режиме проекта

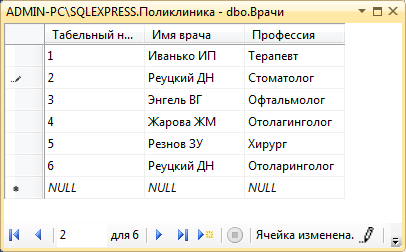


Рисунок 1 – Таблица «Врачи» в режиме просмотра

И так все таблицы.

3.2 СХЕМА ДАННЫХ

\*Тут просто схема из 9 лабы\*

На основе полученных таблиц была создана схема данных, представленная на рисунке 1.

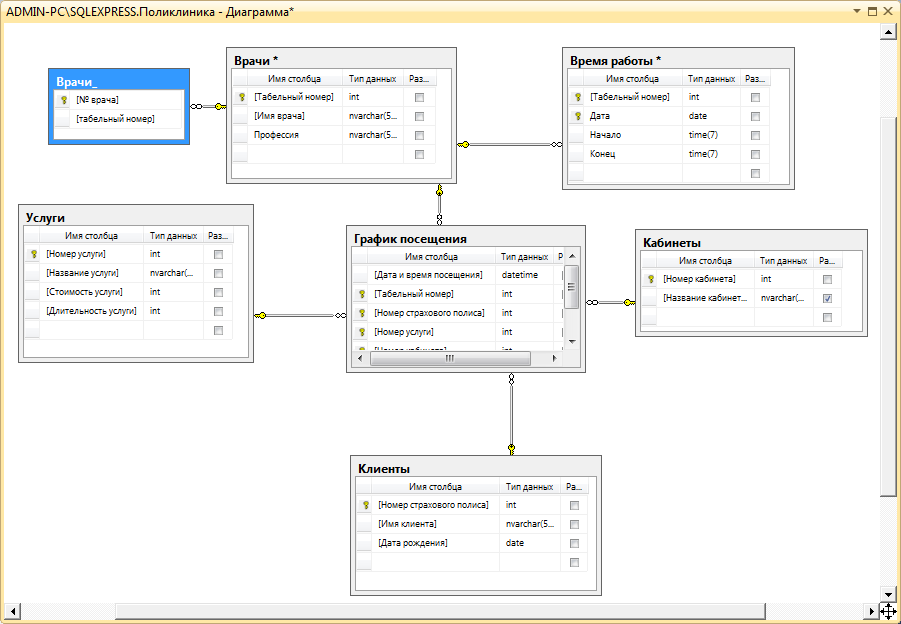


Рисунок 1 – Схема данных

3.3 ТРИГГЕРЫ

\*тут опять теория, что такое триггеры? Для чего используются? Добавляем теорию и затем вставляем 10 лабораторную, но обязательно с расписыванием шагов: создание архивных таблиц, какая информация в них будет храниться. Затем текст триггеров, результат работы\*

3.4 ФУНКЦИИ И ПРОЦЕДУРЫ

\*В архиве есть папка «Примеры для выполнения заданий». Откройте файл «Функции и процедуры». Вам необходимо создать одну процедуру, которая будет реагировать на добавление данных в любую таблицу (на ваш выбор). Лучше всего на таблицу «Клиенты» или «Сотрудники» если таковые в вашей БД имеются. Кроме того, необходимо использовать ВСЕ перечисленные функции (их 3). Мы рассматривали их на лекции. Можно перейти на портал и посмотреть там.

1 функция – функция работы с датами. С ее помощью можно посчитать, сколько лет прошло с сегодняшней даты до даты, которая записана в каком-либо поле в вашей таблице. С помощью такой функции можно вычислить возраст человека, опыт работы, опыт вождения и прочее. Опять же, на ваш выбор.

2 функция – подсчитывает и выводит данные по заданному условия. Используйте с какой-либо вашей таблицей также по своему усмотрению. Например, можно выводить список клиентов, который покупали (или оформляли заказы) в вашей компании несколько раз, или наоборот только один раз.

3 функция – подсчитывает и выводит в виде предложения данные. Также по своему усмотрению используем, можно подсчитать, сколько заказов выполнил тот или иной сотрудник, сколько заказов оформил каждый клиент, какой автомобиль сколько раз арендовали и т.д.

Вставляем текст запроса и результат его работы, так же, как делали в предыдущих главах. ОБЯЗАТЕЛЬНО описываем для чего они нужны\*

3.5 ПРЕДСТАВЛЕНИЯ

\*что такое представления, для чего используются? Опять же немного теории и затем лабораторная 9 – показываем какие представления создали и что они отображают (ПОДРОБНОЕ ОПИСАНИЕ). В противном случае на защите буду спрашивать и придется рассказывать\*

4 АДМИНИСТРИРОВАНИЕ БД

После создания физического проекта базы данных организуется непрерывное слежение за ее функционированием. Полученные сведения об уровне производительности базы данных используются для ее настройки. Для этого привлекаются и средства выбранной СУБД.

Администрирование SQL сервера - это сложный процесс, направленный на постоянную поддержку баз данных с целью поддержания и увеличения их эффективности. Критерий эффективности все больше зависит от безопасности баз данных, а корректное управление учетными записями пользователей гарантирует эффективное функционирование всей компании.

\*Тут ничего не надо менять, так пусть остается\*

4.1 СОЗДАНИЕ ПОЛЬЗОВАТЕЛЕЙ И ОГРАНИЧЕНИЕ ИХ ПРАВ

Пользователь, создающий БД считается ее владельцем. У него есть все права на созданную им БД. Все остальные пользователи (кроме суперпользователи с неограниченными правами), доступа к БД не имею, пока владелец или суперпользователь не выдаст им права.

Для того, чтобы создать пользователя нужно \*дальше сами, это было в 1 лабораторной. Как создали пользователя? Как определили ему права на создание БД?\*

После того, как БЖ была создана, можно выдать права просмотра или редактирования другим пользователям, в случае необходимости. Существует несколько способов выдачи прав. Рассмотрим графический способ.

\*Далее открываем файл под названием «Выдача прав пользователю» и выполняем задание. Скриншоты вставляем в курсовую\*

4.2 СОЗДАНИЕ РЕЗЕРВНОЙ КОПИИ БД. ВОССТАНОВЛЕНИЕ БД

\*В архиве в папке «Примеры для выполнения заданий» откройте файл «Резервное копирование». В нём пошагово расписывается, как создать резервную копию базы данных при помощи графического интерфейса. Сделать это можно и при помощи запроса, но мы этот способ рассматривать не будем.

Откройте файл и пошагово создайте резервную копию вашей базы данных. Затем попробуйте восстановить базу данных под другим именем. Опишите процесс, представьте скриншоты резервного копирования и результаты восстановления.

Полученный файл можно использовать для переноса БД на другое устройство.

P.S. С компьютерами в колледже это не сработает, ибо надо нести копию на сервер, а доступ к серверу вам никто не даст. Поэтому выполняем работу через Horizon\*

ЗАКЛЮЧЕНИЕ

\*Что у вас получилось? Какие выводы можно сделать? Что не получилось?\*

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

\*Ну тут все по стандарту, литература, которой пользовались. Можно включить интернет ресурсы, такие как сайт microsoft.com, docs.microsoft.com, edoc.bseu.by, intuit.ru\*

ПРИЛОЖЕНИЯ

\*приложение по желанию, если хотите выносить скрины отдельно\*

Приложение 1

Приложение 2