|  |  |
| --- | --- |
| Институт (факультет) | Институт информационных технологий |
| Кафедра | МПО ЭВМ |

КУРСОВАЯ РАБОТА

|  |  |
| --- | --- |
| по дисциплине | Проектирование баз данных |
| на тему | Проектирование базы данных |
| «Стоматологическая клиника» | |

|  |
| --- |
| Выполнил студент группы |
| 1ПИб-02-1оп-22 |
| *направления подготовки (специальности)* |
| 09.03.04 Программная инженерия |
| *шифр, наименование* |
| Микуцких Григорий Андреевич |
| *фамилия, имя, отчество* |

|  |
| --- |
| Руководитель |
| Селяничев О.Л. |
| *фамилия, имя, отчество* |
| доцент |
| *должность* |

|  |
| --- |
| Дата представления работы |
| «\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. |
|  |
| Заключение о допуске к защите |
|  |
|  |
|  |
|  |
| Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| количество баллов |
| Подпись преподавателя\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

|  |  |
| --- | --- |
| Череповец, | 2025 |
|  | *год* |

Оглавление

[Введение 3](#_Toc201495769)

[1. Описание предметной области 4](#_Toc201495770)

[2. Инфологическое проектирование 5](#_Toc201495771)

[3. Даталогическое проектирование 10](#_Toc201495772)

[3.1. Логическое проектирование 10](#_Toc201495773)

[3.2. ER-диаграмма 22](#_Toc201495774)

[3.3. Физическое проектирование 27](#_Toc201495775)

[4. Описание программы 34](#_Toc201495776)

[Заключение 42](#_Toc201495777)

[Список литературы 43](#_Toc201495778)

[Приложение 1. Техническое задание 44](#_Toc201495779)

[Приложение 2. Текст программы 49](#_Toc201495780)

[Приложение 3. Руководство пользователя 50](#_Toc201495781)

# Введение

Современный этап развития информационных технологий характеризуется постоянным повышением спроса на высокие технологии для автоматизации рутинных задач. Не являются исключением и стоматологические клиники. Они оказывают комплекс жизненно необходимых человеку услуг для поддержания здоровья полости рта. Прейскурант, лечащие врачи, используемое оборудование, ведение расписания и учёта клиентов – это лишь часть того, что должно быть структурировано и организовано так, чтобы оказывать услуги максимально качественно и с минимумом издержек.

С ростом объёма данных о пациентах, процедурах, платежах, расписании и ресурсах клиники возникают проблемы, связанные с тратой времени на поиск и обработку информации, недостаточной безопасностью конфиденциальной информации и ограниченной масштабируемостью системы, поэтому традиционные методы хранения и обработки информации, такие как бумажные журналы и электронные таблицы, всё чаще оказываются неэффективными и не отвечают постоянно растущим требованиям.

Например, при записи пациента на приём, администратору приходится искать информацию о пациенте в бумажной картотеке, затем просматривать журнал записей, чтобы найти свободное время у врача. Во время приёма врач заполняет бумажную медицинскую карту пациента. Расчёт стоимости оказанных услуг производится также вручную (возможно, с использованием калькулятора) согласно прейскуранту. После завершения приёма администратору необходимо внести данные об оплате в учётный журнал.

Отказаться от использования бумажных каталогов позволит внедрение современной базы данных, которая переведёт всю информацию в электронный формат. Данные о пациентах, расписании, истории болезней и другая важная информация будут хранится в доступном структурированном виде. При записи пациента на приём администратор сможет быстро найти информацию о пациенте и его предыдущих посещениях, а во время приёма врач получит доступ ко всей истории болезни пациента, сможет самостоятельно внести данные о текущем состоянии, диагнозе и проведённых процедурах, не обременяя этим другого сотрудника. Электронная система учёта позволит легко формировать отчёты по предоставленным услугам и платежам за любой период времени. Кроме того, данные в электронном виде могут быть сохранены в резервных копиях, обеспечивая защиту от потери информации.

Таким образом, необходимо создать базу данных и написать программу для автоматизации работы с ней. Целевыми пользователями программы являются администраторы в стоматологических клиниках. Разрабатываемая база данных будет содержать в себе критически важные данные и связи для работы стоматологической клиники, ожидая, что остальная информация будет фиксироваться в других местах.

# Описание предметной области

Предметная область – это часть реального мира, которая подлежит изучению с целью автоматизации организации управления. Предметной областью информационной системы (ИС) является совокупность объектов, свойства которых и отношения, между которыми интересны для пользователей ИС. Любая предметная область может быть разбита на фрагменты. Каждый фрагмент оперирует со своими объектами и с множеством пользователей, которые имеют свои взгляды на предметную область, поэтому выявление предметной области и её анализ является неотъемлемой частью разработки любой ИС [5].

Предметной областью решаемой задачи является стоматологическая клиника – медицинское учреждение, специализирующееся на диагностике, лечении и профилактике заболеваний зубов, полости рта и челюстно-лицевой области.

В стоматологических клиниках ведётся учёт следующей информации:

1. данные пациентов – ФИО, дата рождения, контактные данные, противопоказания, согласия на обработку персональных данных и медицинское вмешательство, срок действия договоров;
2. данные сотрудников – ФИО, должность, оклад (тип и размер), договор, дата заключения договора;
3. расписание приёмов – время приёма, пациент, лечащий врач, кабинет, заключение после приёма;
4. прейскурант – услуга, единица измерения, стоимость;
5. кабинеты – номер кабинета, название, закреплённый сотрудник;
6. стоматологические материалы, оборудование, инструменты и хозяйственные принадлежности – название, артикул, технический паспорт, из какой поставки, за каким кабинетом закреплён, срок годности;
7. поставки – поставщик, контактные данные, дата поставки.

Составление рабочего графика требует различной информации (время работы каждого сотрудника, их квалификация, организация предварительных записей и многое другое). Распространены случаи, когда записи на приёмы стоят на месяцы вперёд.

О клиенте необходимо обязательно знать его контактные данные, а срок заключения договоров и согласий ограничен.

Все необходимые стоматологические ресурсы могут быть из разных поставок, поэтому важно знать кто и когда осуществил поставку для выбора надёжных партнёров, необходима возможность быстрой связи с поставщиком.

Ведение медицинских карт необходимо организовать таким образом, чтобы можно было собрать информацию о конкретном пациенте, так как пациент имеет право знать об оказанных ему медицинских услугах.

Вышеперечисленное крайне сложно организовать традиционными методами (ведение бумажных журналов и каталогов). Перед тем, как создать базу данных и написать программу для работы с ней, необходимо структурировать знания о предметной области, то есть провести инфологическое проектирование.

# Инфологическое проектирование

Инфологическое проектирование – это процесс построения семантической модели предметной области.

Инфологическое моделирование ставит своей целью создание точного выражения человеческих представлений о предметной области, являющейся частью реального мира, и способствует его пониманию. Оно применимо в любой области, ограниченной границами конкретного применения [4].

Базы данных (БД) – это системы хранения информации, обращение к которым может осуществляться через средства управления базами данных. Основное предназначение БД – предоставление пользователю необходимой информации в нужном месте и в нужное время [5]. Базы данных всегда проектируются под конкретное назначение системы.

Система управления базами данных (СУБД) – совокупность программ и языковых средств, предназначенных для управления данными в базе данных, ведения БД и обеспечения взаимодействия её с прикладными программами.

Проектирование базы данных – это поиск способов удовлетворения функциональных требований средствами имеющейся компьютерной технологии с учётом заданных ограничений [6].

Процесс проектирования базы данных охватывает несколько основных сфер:

1. проектирование объектов БД (таблицы, представления, индексы, хранимые процедуры, функции) для представления данных предметной области;
2. проектирование интерфейса взаимодействия с базой данных (формы, отчёты и т.д.), то есть проектирование приложений, которые будут сопровождать данные в БД и реализовывать вопросно-ответные отношения на этих данных;
3. проектирование баз данных под конкретную вычислительную среду или информационную технологию (архитектура «клиент-сервер», параллельные архитектуры, распределённая вычислительная среда);
4. проектирование баз данных под назначение системы (интеллектуальный анализ данных, OLAP, OLTP и так далее) [3].

Инфологическое моделирование проводится на начальном этапе проектирования БД, его результаты используются на заключительном даталогическом этапе, когда решается проблема конструирования непосредственно базы данных в среде СУБД. Также инфологическая модель служит средством коммуникации на разных стадиях жизненного цикла БД.

Перед началом разработки базы данных необходимо выбрать модель данных и определить основные объекты предметной области.

Модель данных (МД) – совокупность правил порождения структур данных в БД, операций над ними, ограничений целостности, определяющих допустимые связи и значения данных, последовательность их изменения [3, с. 14].

Существуют следующие модели данных:

1. иерархическая;
2. сетевая;
3. реляционная;
4. постреляционная (объектно-ориентированная, объектно-реляционная).

Иерархическая МД – это модель, в которой абстрактные понятия находятся в отношении предшествования таким образом, что каждому понятию соответствует только один предшественник (родитель). Только одна часть, называемая корнем модели, не имеет предшественника. Структура данных представляет собой иерархическое дерево, концевые вершины которого задают конкретные данные, а все вершины более высокого уровня определяют признаки классификации конкретных данных. Иерархическая БД состоит из упорядоченного набора деревьев; или, более точно, из упорядоченного набора нескольких экземпляров одного типа дерева [3, с. 15].

Достоинство: данные, отражающие общие свойства совокупности конкретных данных не дублируются.

Недостаток связан с дублированием данных в случае их однотипности.

Сетевая МД так же, как и иерархическая, обладает весьма выразительными свойствами. Необходимость сетевой модели проявляется тогда, когда одни и те же конкретные данные в рамках одной и той же предметной области классифицируются не одной, а несколькими системами классификации, то есть предметная область разбита на части, связанные между собой бинарными связями. Если в иерархических структурах запись-потомок должна иметь ровно одного предка, то в сетевых структурах данных потомок может иметь любое число предков. Связи между записями в сетевой МД выполняются в виде указателей, то есть каждая запись хранит ссылку на другую однотипную запись (или признак конца списка) и ссылки на списки подчинённых записей [3, с. 18].

Достоинство: данные имеют чёткую структуру.

Недостаток: большое количество дополнительной информации о связях.

Реляционная МД – это модель, в которой данные можно представить в виде отношений, изменяющихся во времени. Традиционно в реляционных системах отношением называют таблицу, кортежем – строку таблицы (запись), а атрибутом – столбец (поле) [3, с. 19].

Реляционная МД была предложена Э. Коддом и основана на понятии отношения (relation). Она является наиболее распространённой и практически все современные СУБД ориентированы на такое представление данных [3, с. 18].

Достоинство реляционной модели заключается в простоте для понимания, наглядности и удобстве физической реализации на ЭВМ.

Недостатки: модель не допускает представления объектов со сложной структурой, поскольку в её рамках возможно моделирование лишь с помощью двумерных таблиц. Данные об объектах содержатся, как правило, во многих таблицах, что значительно замедляет обработку данных [3, с. 19].

Постреляционная МД в общем случае представляет собой расширенную реляционную модель, снимающую ограничение неделимости значений полей. То есть, допускаются многозначные поля, значения которых состоят из подзначений. Набор значений многозначных полей считается самостоятельной таблицей, встроенной в основную [3, с. 19].

Достоинство: возможность представления связанных реляционных таблиц одной постреляционной таблицей.

Недостаток: сложность в обеспечении целостности данных [3, с. 19].

Объектно-ориентированная МД представляет структуру, которую можно изобразить графически в виде дерева, узлами которого являются объекты.

Каждый объект характеризуется уникальным идентификатором, состоянием и поведением. Состояние объекта определяется множеством значений его свойств (атрибутов). Поведение объекта описывают методы, называемые процедурами. То есть, составной частью описания объекта являются процедуры, способные производить действия над атрибутами объекта в случае наступления тех или иных событий. Объекты могут объединяться в классы. Экземпляры одного класса отличаются лишь значениями своих свойств, но не своими методами. Методы устанавливаются при определении класса [3, с. 21].

Для выполнения действий над объектами применяются объектно-ориентированные механизмы – наследование, инкапсуляция, полиморфизм.

Основным достоинством является способность отображать информацию о сложных объектах. Эта модель обычно применяется для сложных предметных областей, при моделировании которых не хватает функциональности реляционной модели [3, с. 21].

Недостаток: неудобство обработки больших массивов данных.

Объектно-реляционная МД является гибридной моделью, сочетающей возможности реляционной модели с объектными свойствами данных. В ней используются такие объектно-ориентированные компоненты, как инкапсуляция, полиморфизм, наследование и тому подобное. Отличительная особенность объектно-реляционной модели от объектно-ориентированной состоит в том, что она основана на стратегии реляционной модели [3, с. 21].

Была выбрана реляционная МД. Она достаточна для моделирования большинства предметных областей, но при реальном проектировании обнаруживается её ограниченность – нет достаточных средств для отражения смысла данных. Проектирование начинается с выделения сущностей и связей между ними, а модель не предлагает аппарат для разделения сущностей и связей.

Немного забегая вперёд, необходимо сказать, что по инфологической модели строят логическую модель. А результатом логического проектирования реляционной модели является ER-диаграмма (entity-relationship diagram). Основными понятиями ER-модели являются сущность, связь, атрибут, первичный ключ и внешний ключ.

Сущность – реальный или представляемый объект, информация о котором должна сохраняться и быть доступна. В диаграммах ER-модели сущность представляется в виде прямоугольника, содержащего имя сущности. При этом имя сущности — это имя типа, а не некоторого конкретного его экземпляра [2].

Связь — это графически изображаемая ассоциация, устанавливаемая между двумя сущностями. Эта ассоциация всегда является бинарной. В любой связи выделяются два конца, на каждом из которых указывается имя конца связи, степень конца связи (сколько экземпляров данной сущности связывается), обязательность связи (любой ли экземпляр данной сущности должен участвовать в данной связи) [2].

Связь представляется в виде линии, связывающей две сущности или ведущей от сущности к ней же самой. При этом в месте «стыковки» связи с сущностью используются трёхточечный вход в прямоугольник сущности, если для этой сущности в связи могут использоваться много (many) экземпляров сущности, и одноточечный вход, если в связи может участвовать только один экземпляр сущности. Обязательный конец связи изображается сплошной линией, а необязательный - прерывистой линией (неидентифицирующая связь) [3, с. 60].

Атрибут — любая деталь, которая служит для уточнения, идентификации, классификации, числовой характеристики или выражения состояния сущности. Имена атрибутов заносятся в прямоугольник, изображающий сущность, под именем сущности и изображаются малыми буквами, возможно, с примерами [2].

Первичный ключ – это атрибут или комбинация атрибутов, позволяющая однозначно идентифицировать экземпляр сущности.

Внешний ключ – это копия существующего первичного ключа в экземпляре другой сущности, образующая связь между экземплярами («один-к-одному», «один-ко-многим», «многие-ко-многим»).

В рамках инфологического проектирования была выделена сущность «Информация о работе клиники» со следующими атрибутами (табл. 1).

Таблица 1

Информация о работе клиники

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ФИО пациента | Паспорт пациента | СНИЛС пациента | ИНН пациента | Номер телефона пациента |
| Иванов Алексей Сергеевич | 4510 123456 | 123-456-789 00 | 770123456789 | +7 (915) 123-45-67 |
| Смирнова Ирина Петровна | 4012 987654 | 112-233-445 66 | 503212345678 | +7 (926) 234-56-78 |
| Сидоров Михаил Юрьевич | 4508 334455 | 156-789-234 55 | 502134567890 | +7 (903) 567-89-01 |
| Петрова Ольга Викторовна | 4610 112233 | 145-678-321 44 | 780123456701 | +7 (985) 456-78-90 |
| Сидоров Михаил Юрьевич | 4508 334455 | 156-789-234 55 | 502134567890 | +7 (903) 567-89-01 |
| Иванов Иван Иванович | 1111 111111 | 123-456-789 10 | 109876543210 | - |
| Иванов Иван Иванович | 1111 111111 | 123-456-789 10 | 109876543210 | - |
| Иванов Иван Иванович | 1111 111111 | 123-456-789 10 | 109876543210 | - |

Продолжение табл. 1

|  |  |  |  |
| --- | --- | --- | --- |
| Электронная почта  пациента | Противопоказания  пациента | Согласие на обработку персональных данных | Согласие на медицинское вмешательство |
| Ivanov@mail.ru | Аллергия на лидокаин | - | - |
| smirnova.irina@yandex.ru | Хроническая гипертония | - | - |
| sidormih@yandex.ru | Нет | - | - |
| petrova.olga@mail.ru | Сахарный диабет | - | - |
| sidormih@yandex.ru | Нет | - | - |
| iivanovich@gmail.com | Нет | - | - |
| iivanovich@gmail.com | Нет | - | - |
| iivanovich@gmail.com | Нет | - | - |

Продолжение табл. 1

|  |  |  |  |
| --- | --- | --- | --- |
| Срок договора пациента | План лечения | Дата и время приёма | Заключение |
| 01.06.2024 – 01.06.2025 | Лечение кариеса 1-го зуба | 2025-06-01 10:00 | Успешное лечение |
| 01.01.2025 – 01.07.2025 | Удаление зуба мудрости | 2025-06-02 14:30 | Рекомендован контроль через 1 месяц |
| 01.12.2023 – 01.06.2025 | Профессиональная чистка зубов | 2025-06-03 09:00 | Зубы в хорошем состоянии |
| 01.09.2024 – 01.09.2025 | Протезирование 2-го зуба | 2025-06-04 11:15 | Протез установлен |
| 01.12.2023 – 01.06.2025 | Отбеливание зубов | 2025-06-05 15:45 | Процедура завершена |
| 05.05.2025 – 06.06.2026 | Лечение кариеса 2-го зуба | 2025-06-06 16:35 | - |
| 05.05.2025 – 06.06.2026 | Лечение кариеса 2-го зуба | 2025-06-06 16:35 | - |
| 05.05.2025 – 06.06.2026 | Профессиональная чистка зубов | 2025-06-06 14:00 | Повторить через месяц |

Продолжение табл. 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Кабинет | ФИО сотрудника | Паспорт сотрудника | СНИЛС сотрудника | ИНН сотрудника |
| 200 Консультационно-диагностический кабинет | Васильева Наталья Петровна | 4508 556677 | 156-789-123 55 | 502134567891 |
| 102а Кабинет ортодонтии | Петров Сергей Иванович | 4012 223344 | 112-344-556 66 | 503212345679 |
| 202б Кабинет стерилизационной обработки | Кузнецова Ольга Викторовна | 4511 998877 | 134-123-456 33 | 773405678902 |
| 100 Кабинет имплантологии | Сидоров Михаил Юрьевич | 4610 334455 | 145-987-654 44 | 780123456702 |
| 200 Консультационно-диагностический кабинет | Васильева Наталья Петровна | 4508 556677 | 156-789-123 55 | 502134567891 |
| 101 Стоматологический кабинет (терапевтический) | Петров Сергей Иванович | 4012 223344 | 112-344-556 66 | 503212345679 |
| 101 Стоматологический кабинет (терапевтический) | Кузнецова Ольга Викторовна | 4511 998877 | 134-123-456 33 | 773405678902 |
| 202б Кабинет стерилизационной обработки | Кузнецова Ольга Викторовна | 4511 998877 | 134-123-456 33 | 773405678902 |

Продолжение табл. 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Номер телефона сотрудника | Электронная почта сотрудника | Должность | Договор сотрудника | Дата заключения договора сотрудника | Форма оклада | Оклад, ₽ |
| +7 (903) 567-89-02 | vasilieva.nat@mail.ru | Терапевт | - | 18.04.2022 | Процент от выручки | 10 |
| +7 (926) 234-56-89 | petrov.sergey@yandex.ru | Ортодонт | - | 17.03.2023 | Фиксированная | 60 000 |
| +7 (999) 345-67-90 | kuznetsova.olga@gmail.com | Медицинская сестра | - | 19.02.2024 | Почасовая ставка | 350 |
| +7 (985) 456-78-91 | sidorov.mihail@mail.ru | Имплантолог | - | 10.01.2025 | Сдельная оплата | 1 500 |
| +7 (903) 567-89-02 | vasilieva.nat@mail.ru | Гигиенист | - | 01.01.2025 | Фиксированная | 55 000 |
| +7 (926) 234-56-89 | petrov.sergey@yandex.ru | Стоматолог | - | 17.03.2024 | Фиксированная | 80 000 |
| +7 (999) 345-67-90 | petrov.sergey@yandex.ru | Стоматолог | - | 17.03.2024 | Фиксированная | 80 000 |
| +7 (999) 345-67-90 | kuznetsova.olga@gmail.com | Медицинская сестра | - | 19.02.2024 | Почасовая ставка | 350 |

Продолжение табл. 1

|  |  |
| --- | --- |
| Услуга | Ресурс |
| Лечение кариеса, 3 500 ₽, 1 зуб | - |
| Атравматичное удаление зуба под имплантацию, 5 000 ₽, 1 зуб | Бормашина "DentPro X5"; NSK-V370-KIT; ЗАО "Дентал-Импорт"; +7 (495) 123-45-67; 2023-06-15; 01.06.2023 - 01.06.2028; В эксплуатации |
| Подготовка пациента к приёму, 400 ₽, 1 процедура | Перчатки нитриловые; WDP-MPX-01; ИП Смирнов С.В;. +7 (812) 234-56-78; 2025-03-05; 01.03.2025 - 01.03.2026; Почти закончились |
| Установка импланта (Straumann), 55 000 ₽, 1 имплант | - |
| Профессиональная гигиена полости рта (Air Flow + ультразвук), 4 000 ₽, 1 процедура | - |
| Лечение кариеса, 3 500 ₽, 1 зуб | Зеркало стоматологическое; 3M-FZ550-A2; ООО "МедТехСнаб"; +7 (903) 345-67-89; 2025-01-10; 10.11.2024 - 10.11.2029; В наличии |
| Лечение кариеса, 3 500 ₽, 1 зуб | Цемент стоматологический; GC-FIXGP-001; ООО "СтомаПоставка"; +7 (926) 456-78-90; 2024-10-01; 20.09.2024 - 20.09.2026; В наличии |
| Подготовка пациента к приёму, 400 ₽, 1 процедура | Световая лампа "CureLight 3000"; MPX/WDP-2024-0110; ООО "ТехДентЛайт"; +7 (499) 567-89-01; 2022-11-25; 10.11.2022 - 10.11.2027; Требует обслуживания |

В таблицах присутствует избыточность и дублирование информации, что может привести к аномалиям (будут описаны далее). Проблемы должны и будут решены в процессе даталогического проектирования.

1. Даталогическое проектирование

При проектировании БД решается две проблемы:

1. «как отобразить объекты предметной области в абстрактные модели?»;
2. «как обеспечить эффективность выполнения запросов к БД, то есть как, имея ввиду особенности конкретной СУБД, расположить данные во внешней памяти и создание каких дополнительных структур может потребоваться?».

Проектирование реляционных БД производится методом последовательных приближений к удовлетворительному набору схем отношений и начинается с представления предметной области в виде одного или нескольких отношений.

1. Логическое проектирование

Для реляционной модели данных даталогическая модель – набор схем отношений, отражающих сущности и их связи. На этапе логического проектирования учитывается специфика конкретной модели данных, но может не учитываться специфика конкретной СУБД [3].

Реляционная БД содержит как структурную, так и семантическую информацию. Структура базы данных определяется числом и видом включённых в неё отношений, и связями типа «один ко многим», существующими между кортежами этих отношений. Семантическая часть описывает множество функциональных зависимостей, существующих между атрибутами этих отношений. Если даны два атрибута X и Y некоторого отношения, то говорят, что Y функционально зависит от X, если в любой момент времени каждому значению X соответствует ровно одно значение Y [2].

Избыточная функциональная зависимость – зависимость, заключающая в себе такую информацию, которая может быть получена на основе других зависимостей, имеющихся в базе данных. Корректной считается такая схема БД, в которой отсутствуют избыточные функциональные зависимости. В противном случае приходится прибегать к процедуре декомпозиции (разложения) имеющегося множества отношений. При этом порождаемое множество содержит большее число отношений, которые являются проекциями отношений исходного множества. Обратимый пошаговый процесс замены данной совокупности отношений другой схемой с устранением избыточных функциональных зависимостей называется нормализацией [2].

Нормализация – это декомпозиция отношения, находящегося в предыдущей нормальной форме, в два или более отношения, удовлетворяющих следующей нормальной форме. Цель нормализации: экономия памяти, устранение избыточности, исключение противоречия, упрощение управления.

Основные свойства НФ:

1. каждая следующая НФ в некотором смысле лучше предыдущей;
2. при переходе к следующей НФ свойства предыдущих сохраняются.

Процесс проектирования – процесс нормализации схем отношений, причём каждая следующая нормальная форма обладает свойствами лучшими, чем предыдущая. Каждой нормальной форме соответствует некоторый определённый набор ограничений [2].

В теории реляционных баз данных выделяются следующие нормальные формы в порядке их усложнения:

1. первая нормальная форма (1НФ);
2. вторая нормальная форма (2НФ);
3. третья нормальная форма (3НФ);
4. нормальная форма Бойса-Кодда (НФБК);
5. четвертая нормальная форма (4НФ);
6. пятая нормальная форма (5НФ).

Нормализация начинается с построения ненормализованной формы. В ненормализованной форме таблица может хранить информацию о всех объектах предметной области. Приводить данные к нормальным формам можно только последовательно [2].

Проблемы ненормализованных таблиц:

* избыточность данных (повторение информации);
* аномалии обновления (в разных местах разные данные);
* аномалии удаления;
* аномалии ввода.

В исходной таблице они присутствуют (см. табл. 1). Необходимо её нормализовать прежде, чем строить логическую модель.

Определение 1НФ – значения атрибутов отношения должны быть атомарны.

Критика: атрибуты «Кабинет», «Ресурс», «ФИО», «Срок договора» и «Дата и время приёма» неатомарны.

Предложение:

1. атрибут «Кабинет» разбить на «Номер кабинета» и «Название кабинета»;
2. атрибут «Ресурс» разбить на «Название ресурса», «Артикул», «Номер технического паспорта», «Название поставщика», «Номер телефона поставщика», «Дата поставки», «Дата производства» и «Срок годности»;
3. атрибут «ФИО» разбить на «Фамилия», «Имя» и «Отчество»;
4. атрибут «Срок договора пациента» разбить на «Дата заключения договора пациента» и «Дата окончания договора пациента».

1НФ исходной таблицы представлена в табл. 2.

Таблица 2

Информация о работе клиники

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Фамилия пациента | Имя пациента | Отчество пациента | Паспорт пациента | СНИЛС пациента | ИНН пациента | Номер телефона пациента | Электронная почта пациента |
| Иванов | Алексей | Сергеевич | 4510 123456 | 123-456-789 00 | 770123456789 | +7 (915) 123-45-67 | Ivanov@mail.ru |
| Смирнова | Ирина | Петровна | 4012 987654 | 112-233-445 66 | 503212345678 | +7 (926) 234-56-78 | smirnova.irina@yandex.ru |
| Сидоров | Михаил | Юрьевич | 4508 334455 | 156-789-234 55 | 502134567890 | +7 (903) 567-89-01 | sidormih@yandex.ru |
| Петрова | Ольга | Викторовна | 4610 112233 | 145-678-321 44 | 780123456701 | +7 (985) 456-78-90 | petrova.olga@mail.ru |
| Сидоров | Михаил | Юрьевич | 4508 334455 | 156-789-234 55 | 502134567890 | +7 (903) 567-89-01 | sidormih@yandex.ru |
| Иванов | Иван | Иванович | 1111 111111 | 123-456-789 10 | 109876543210 | - | iivanovich@gmail.com |
| Иванов | Иван | Иванович | 1111 111111 | 123-456-789 10 | 109876543210 | - | iivanovich@gmail.com |
| Иванов | Иван | Иванович | 1111 111111 | 123-456-789 10 | 109876543210 | - | iivanovich@gmail.com |

Продолжение табл. 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Противопоказания пациента | Согласие на обработку персональных данных | Согласие на медицинское вмешательство пациента | Дата заключения договора пациента | Дата окончания договора пациента | План лечения |
| Аллергия на лидокаин | - | - | 01.06.2024 | 01.06.2025 | Лечение кариеса 1-го зуба |
| Хроническая гипертония | - | - | 01.01.2025 | 01.07.2025 | Удаление зуба мудрости |
| Нет | - | - | 01.12.2023 | 01.06.2025 | Профессиональная чистка зубов |
| Сахарный диабет | - | - | 01.09.2024 | 01.09.2025 | Протезирование 2-го зуба |
| Нет | - | - | 01.12.2023 | 01.06.2025 | Отбеливание зубов |
| Нет | - | - | 05.05.2025 | 06.06.2026 | Лечение кариеса 2-го зуба |
| Нет | - | - | 05.05.2025 | 06.06.2026 | Лечение кариеса 2-го зуба |
| Нет | - | - | 05.05.2025 | 06.06.2026 | Профессиональная чистка зубов |

Продолжение табл. 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Дата и время приёма | Заключение приёма | Номер кабинета | Название кабинета | Фамилия сотрудника | Имя сотрудника | Отчество сотрудника |
| 2025-06-01 10:00 | Успешное лечение | 200 | Консультационно-диагностический кабинет | Васильева | Наталья | Петровна |
| 2025-06-02 14:30 | Рекомендован контроль через 1 месяц | 102а | Кабинет ортодонтии | Петров | Сергей | Иванович |
| 2025-06-03 09:00 | Зубы в хорошем состоянии | 202б | Кабинет стерилизационной обработки | Кузнецова | Ольга | Викторовна |
| 2025-06-04 11:15 | Протез установлен | 100 | Кабинет имплантологии | Сидоров | Михаил | Юрьевич |
| 2025-06-05 15:45 | Процедура завершена | 200 | Консультационно-диагностический кабинет | Васильева | Наталья | Петровна |
| 2025-06-06 16:35 | - | 101 | Стоматологический кабинет (терапевтический) | Петров | Сергей | Иванович |
| 2025-06-06 16:35 | - | 101 | Стоматологический кабинет (терапевтический) | Петров | Сергей | Иванович |
| 2025-06-06 14:00 | Повторить через месяц | 202б | Кабинет стерилизационной обработки | Кузнецова | Ольга | Викторовна |

Продолжение табл. 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Паспорт сотрудника | СНИЛС сотрудника | ИНН сотрудника | Номер телефона сотрудника | Электронная почта сотрудника | Должность |
| 4508 556677 | 156-789-123 55 | 502134567891 | +7 (903) 567-89-02 | vasilieva.nat@mail.ru | Терапевт |
| 4012 223344 | 112-344-556 66 | 503212345679 | +7 (926) 234-56-89 | petrov.sergey@yandex.ru | Ортодонт |
| 4511 998877 | 134-123-456 33 | 773405678902 | +7 (999) 345-67-90 | kuznetsova.olga@gmail.com | Медицинская сестра |
| 4610 334455 | 145-987-654 44 | 780123456702 | +7 (985) 456-78-91 | sidorov.mihail@mail.ru | Имплантолог |
| 4508 556677 | 156-789-123 55 | 502134567891 | +7 (903) 567-89-02 | vasilieva.nat@mail.ru | Гигиенист |
| 4012 223344 | 112-344-556 66 | 503212345679 | +7 (926) 234-56-89 | petrov.sergey@yandex.ru | Стоматолог |
| 4012 223344 | 112-344-556 66 | 503212345679 | +7 (926) 234-56-89 | petrov.sergey@yandex.ru | Стоматолог |
| 4511 998877 | 134-123-456 33 | 773405678902 | +7 (999) 345-67-90 | kuznetsova.olga@gmail.com | Медицинская сестра |

Продолжение табл. 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Договор  сотрудника | Дата заключения  договора сотрудника | Форма оклада | Оклад, ₽ | Название услуги | Стоимость, ₽ | Ед. измерения |
| - | 18.04.2022 | Процент от выручки | 10 | Лечение кариеса | 3 500 | 1 зуб |
| - | 17.03.2023 | Фиксированная | 60 000 | Атравматичное удаление зуба под имплантацию | 5 000 | 1 процедура |
| - | 19.02.2024 | Почасовая ставка | 350 | Подготовка пациента к приёму | 400 | 1 имплант |
| - | 10.01.2025 | Сдельная оплата | 1 500 | Установка импланта (Straumann) | 55 000 |  |
| - | 01.01.2025 | Фиксированная | 55 000 | Профессиональная гигиена полости рта (Air Flow + ультразвук) | 4 000 | - |
| - | 17.03.2024 | Фиксированная | 80 000 | Лечение кариеса | 3 500 | 1 зуб |
| - | 17.03.2024 | Фиксированная | 80 000 | Лечение кариеса | 3 500 | 1 зуб |
| - | 01.01.2025 | Фиксированная | 55 000 | Профессиональная гигиена полости рта (Air Flow + ультразвук) | 4 000 | 1 процедура |

Продолжение табл. 2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Название  ресурса | Дата  производства | Срок  годности | Артикул | Дата  поставки | Название  поставщика | Номер телефона  поставщика | Статус  ресурса | Номер технического паспорта |
| - | - | - | - | - | - |  | - | - |
| Зеркало стоматологическое | 10.11.2024 | 10.11.2029 | 3M-FZ550-A2 | 2025-01-10 | ООО "МедТехСнаб" | +7 (495) 123-45-67 | В наличии | - |
| Бормашина "DentPro X5" | 01.06.2023 | 01.06.2028 | - | 2023-06-15 | ЗАО "Дентал-Импорт" | +7 (812) 234-56-78 | В эксплуатации | NSK-V370-KIT |
| - | - | - | - | - | - |  | - | - |
| - | - | - | - | - | - |  | - | - |
| Перчатки нитриловые | 01.03.2025 | 01.03.2026 | WDP-MPX-01 | 2025-03-05 | ИП Смирнов С.В | +7 (903) 345-67-89 | Почти закончились | - |
| Цемент стоматологический | 20.09.2024 | 20.09.2026 | GC-FIXGP-001 | 2024-10-01 | ООО "СтомаПоставка" | +7 (926) 456-78-90 | В наличии | - |
| Световая лампа "CureLight 3000" | 10.11.2022 | 10.11.2027 | - | 2022-11-25 | ООО "ТехДентЛайт" | +7 (499) 567-89-01 | Требует обслуживания | MPX/WDP-2024-0110 |

Определение 2НФ – тогда и только тогда, когда удовлетворяет требованиям 1НФ и каждый неключевой атрибут полностью зависит от первичного ключа.

Для перехода к 2НФ необходимо:

1) определить на какие части можно разбить первичный ключа так, чтобы некоторые из не ключевых полей зависели от одной их этих частей, причём эти части могут содержать несколько атрибутов;

2) создать новую сущность для каждой части такого ключа и группы зависящих от неё атрибутов и переместить их в эту сущность;

3) удалить из исходной сущности атрибуты, перемещённые в другую сущность, кроме тех, которые станут внешним ключом.

Критика:

* атрибуты «Фамилия», «Имя», «Отчество», «Паспорт», «СНИЛС», «ИНН», «Номер телефона» и «Электронная почта» имеют и пациенты, и сотрудники;
* информация о стоматологических ресурсах требует дублирование всех остальных данных, что может привести к аномалиям.

Предложение – ввести первичный ключ и заменить на внешний те атрибуты, информация которых может дублироваться, для следующих атрибутов:

* 1. атрибуты «Фамилия», «Имя», «Отчество», «Паспорт», «СНИЛС», «ИНН», «Номер телефона» и «Электронная почта» перемещены в новую сущность «Личности», и теперь зависят от нового атрибута внешнего ключа «Id личности»;
  2. атрибуты «Должность», «Формы оклада», «Оклад», «Дата заключения договора», «Договор», «Номер кабинета» перемещены в новую сущность «Сотрудники», заменив их на новый внешний ключ «Id сотрудника»;
  3. атрибуты «Название ресурса», «Дата производства», «Срок годности», «Артикул», «Дата поставки», «Название поставщика», «Статус ресурса», «Номер технического паспорта оборудования» перемещены в отдельную сущность «Информация о ресурсах» и зависят от нового атрибута «Id ресурса».

Образованы следующие связи:

* личности сотрудника и пациента определяются по «Id личности», тем самым удалена излишняя и дублирующаяся информация;
* сущность «Информация о работе клиники» вместо данных сотрудника содержит внешний ключ «Id сотрудника»;
* для идентификации ресурса используется «Id ресурса», который используется в сущности «Информация о работе клиники».

2НФ представлена в табл. 3-6.

Таблица 3

Личности

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id личности | Фамилия | Имя | Отчество | Паспорт |
| 1 | Иванов | Алексей | Сергеевич | 4510 123456 |
| 2 | Смирнова | Ирина | Петровна | 4012 987654 |
| 3 | Сидоров | Михаил | Юрьевич | 4508 334455 |
| 4 | Петрова | Ольга | Викторовна | 4610 112233 |
| 5 | Петров | Сергей | Иванович | 4012 223344 |
| 6 | Кузнецова | Ольга | Викторовна | 4511 998877 |
| 7 | Васильева | Наталья | Петровна | 4508 556677 |
| 10 | Иванов | Иван | Иванович | 1111 111111 |

Продолжение табл. 3

|  |  |  |  |
| --- | --- | --- | --- |
| СНИЛС | ИНН | Номер телефона | Электронная почта |
| 123-456-789 00 | 770123456789 | +7 (915) 123-45-67 | Ivanov@mail.ru |
| 112-233-445 66 | 503212345678 | +7 (926) 234-56-78 | smirnova.irina@yandex.ru |
| 156-789-234 55 | 502134567890 | +7 (903) 567-89-01 | sidormih@yandex.ru |
| 145-678-321 44 | 780123456701 | +7 (985) 456-78-90 | petrova.olga@mail.ru |
| 112-344-556 66 | 503212345679 | +7 (926) 234-56-89 | petrov.serge@yandex.ru |
| 134-123-456 33 | 773405678902 | +7 (999) 345-67-90 | kuznetsova.olg@gmail.com |
| 156-789-123 55 | 502134567891 | +7 (903) 567-89-02 | vasilieva.nat@mail.ru |
| 123-456-789 10 | 109876543210 | - | iivanovich@gmail.com |

Таблица 4

Сотрудники

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id  сотрудника | Id  личности | Должность | Номер  кабинета | Название кабинета |
| 1 | 7 | Терапевт | 200 | Консультационно-диагностический |
| 3 | 5 | Ортодонт | 102а | Кабинет ортодонтии |
| 4 | 6 | Медсестра | 202б | Кабинет стерилизационной обработки |
| 5 | 3 | Имплантолог | 100 | Кабинет имплантологии |
| 2 | 7 | Гигиенист | 200 | Консультационно-диагностический |
| 10 | 5 | Стоматолог | 101 | Стоматологический кабинет (терапевтический) |

Продолжение табл. 4

|  |  |  |  |
| --- | --- | --- | --- |
| Договор | Дата заключения договора | Форма оклада | Оклад, ₽ |
| - | 18.04.2022 | Процент от выручки | 10 |
| - | 17.03.2023 | Фиксированная | 60 000 |
| - | 19.02.2024 | Почасовая ставка | 350 |
| - | 10.01.2025 | Сдельная оплата | 1 500 |
| - | 01.01.2025 | Фиксированная | 55 000 |
| - | 17.03.2024 | Фиксированная | 80 000 |

Таблица 5

Информация о работе клиники

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id  личности | Противопоказания  пациента | Согласие на обработку  персональных данных | Согласие на  медицинское  вмешательство | Дата заключения  договора |
| 1 | Аллергия на лидокаин | - | - | 01.06.2024 |
| 2 | Хроническая гипертония | - | - | 01.01.2025 |
| 3 | Нет | - | - | 01.12.2023 |
| 4 | Сахарный диабет | - | - | 01.09.2024 |
| 3 | Нет | - | - | 01.12.2023 |
| 10 | Нет | - | - | 05.05.2025 |
| 10 | Нет | - | - | 05.05.2025 |
| 10 | Нет | - | - | 05.05.2025 |

Продолжение табл. 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Дата окончания  договора | План лечения | Дата и время  приёма | Id  сотрудника | Заключение  приёма |
| 01.06.2025 | Лечение кариеса 1-го зуба | 2025-06-01 10:00 | 1 | Успешное лечение |
| 01.07.2025 | Удаление зуба мудрости | 2025-06-02 14:30 | 3 | Рекомендован контроль через 1 месяц |
| 01.06.2025 | Профессиональная чистка зубов | 2025-06-03 09:00 | 4 | Зубы в хорошем состоянии |
| 01.09.2025 | Протезирование 2-го зуба | 2025-06-04 11:15 | 5 | Протез установлен |
| 01.06.2025 | Отбеливание зубов | 2025-06-05 15:45 | 2 | Процедура завершена |
| 06.06.2026 | Лечение кариеса 2-го зуба | 2025-06-06 16:35 | 10 | - |
| 06.06.2026 | Лечение кариеса 2-го зуба | 2025-06-06 16:35 | 10 | - |
| 06.06.2026 | Профессиональная чистка зубов | 2025-06-06 14:00 | 4 | Повторить через месяц |

Продолжение табл. 5

|  |  |  |
| --- | --- | --- |
| Название услуги | Стоимость | Ед. измерения |
| Лечение кариеса | 3 500 | 1 зуб |
| Атравматичное удаление зуба под имплантацию | 5 000 | 1 зуб |
| Подготовка пациента к приёму | 400 | 1 процедура |
| Установка импланта (Straumann) | 55 000 | 1 имплант |
| Профессиональная гигиена полости рта (Air Flow + ультразвук) | 4 000 | 1 процедура |
| Лечение кариеса | 3 500 | 1 зуб |
| Лечение кариеса | 3 500 | 1 зуб |
| Профессиональная гигиена полости рта (Air Flow + ультразвук) | 4 000 | 1 процедура |

Таблица 6

Информация о ресурсах

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id ресурса | Название ресурса | Дата производства | Срок годности | Артикул |
| 2002 | Зеркало стоматологическое | 10.11.2024 | 10.11.2029 | 3M-FZ550-A2 |
| 4001 | Бормашина "DentPro X5" | 01.06.2023 | 01.06.2028 | - |
| 3003 | Перчатки нитриловые | 01.03.2025 | 01.03.2026 | WDP-MPX-01 |
| 1001 | Цемент стоматологический | 20.09.2024 | 20.09.2026 | GC-FIXGP-001 |
| 4002 | Световая лампа "CureLight 3000" | 10.11.2022 | 10.11.2027 | - |

Продолжение табл. 6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Дата  поставки | Название  поставщика | Номер телефона  поставщика | Статус ресурса | Номер технического  паспорта |
| 2025-01-10 | ООО "МедТехСнаб" | +7 (495) 123-45-67 | В наличии | - |
| 2023-06-15 | ЗАО "ДенталИмпорт" | +7 (812) 234-56-78 | В эксплуатации | NSK-V370-KIT |
| 2025-03-05 | ИП Смирнов С.В. | +7 (903) 345-67-89 | Почти закончились | - |
| 2024-10-01 | ООО "СтомаПоставка" | +7 (926) 456-78-90 | В наличии | - |
| 2022-11-25 | ООО "ТехДентЛайт" | +7 (499) 567-89-01 | Требует обслуживания | MPX/WDP-2024-0110 |

Определение 3НФ – тогда и только тогда, когда удовлетворяет 2НФ, каждый неключевой атрибут нетранзитивно зависит от первичного ключа и ни один из неключевых атрибутов не зависит функционально от любого другого неключевого атрибута.

Для перехода к 3НФ необходимо:

1) определить все атрибуты, от которых зависят другие;

2) создать новую сущность для каждого такого атрибута или группы зависящих от него атрибутов и переместить их в эту сущность; атрибут, от которого зависят все остальные атрибуты станет первичным ключом новой сущности;

3) удалить перемещённые атрибуты из исходной, оставив лишь те, которые станут внешними ключами.

Критика:

1. внешние ключи ссылаются на избыточную информацию;
2. сущность «Информации о ресурсах» содержит атрибуты «Технический паспорт» и «Артикул», которые могут быть пустыми у разных типов ресурсов, а также лишнюю информацию о поставщике;
3. сущность «Информация о работе клиники» содержит дублирующуюся информацию о пациентах, приёмах и услугах;
4. сущность «Сотрудники» содержит избыточную информацию о кабинетах и формах оклада.

Предложение:

1. сущность «Сотрудники» разделить на сущности «Сотрудники», «Кабинеты» и «Формы оклада» со следующими атрибутами:

* «Сотрудники» – «Id сотрудника», «Id личности», «Id должности», «Оклад», «Id форма оклада», «Дата заключения договора», «Договор»;
* «Кабинеты» – «Номер кабинета», «Название кабинета»;
* «Формы оклада» – «Id формы оклада», «Название».

1. сущность «Информация о ресурсах» разделить на следующие таблицы - «Стоматологические материалы», «Оборудование», «Хозяйственные принадлежности», «Инструменты», решив проблему наличия/отсутствия атрибутов «Артикул» и «Технический паспорт», со следующими атрибутами:

* «Стоматологические материалы» – «Id стом. материала», «Название», «Номер кабинета», «Дата производства», «Срок годности», «Id статуса», «Артикул», «Id поставки»;
* «Оборудование» – «Id стом. материала», «Название», «Номер кабинета», «Дата производства», «Срок годности», «Номер технического паспорта оборудования», «Id статуса», «Id поставки»;
* «Хозяйственные принадлежности» – «Id стом. материала», «Название», «Номер кабинета», «Дата производства», «Срок годности», «Артикул», «Id статуса», «Id поставки»;
* «Инструменты» – «Id стом. материала», «Название», «Номер кабинета», «Дата производства», «Срок годности», «Артикул», «Id статуса», «Id поставки».

1. создать сущность «Статусы ресурсов», переместить атрибут «Статус ресурса» из сущности «Информация о ресурсах», заменив на «Id статуса»;
2. создать «Медицинские карты» с атрибутами «Id приёма» и «Id услуги»;
3. «Информация о работе клиники» разбить на сущности: «Пациенты», «Планы лечения», «Приёмы» и «Прейскурант» со следующими атрибутами:

* «Пациенты» – «Id пациента», «Противопоказания», «Id личности», «Дата заключения», «Дата окончания», «Согласие на обработку персональных данных», «Согласие на медицинское вмешательство»;
* «Планы лечения» – «Id планы лечения», «Id пациента», «План»;
* «Приёмы» – «Id приёма», «Id сотрудника», «Номер кабинета», «Id пациента», «Дата и время», «Заключение»;
* «Прейскурант» – «Id услуги», «Название», «Стоимость», «Единица измерения».

1. создать сущность «Поставки», переместить «Название поставщика», «Номер телефона поставщика» и «Дата поставки», заменив их на «Id поставки».

Таким образом, образовались связи между следующими сущностями:

* сущности «Стоматологические материалы», «Хозяйственные принадлежности», «Оборудование», «Инструменты» зависят от сущностей «Статусы ресурсов» (внешний ключ «Id статуса») и «Поставки» (внешний ключ «Id поставки»);
* сущность «Планы лечения» зависит от «Пациенты» («Id пациента»);
* сущность «Приёмы» зависят от сущностей «Пациенты» («Id пациента»), «Сотрудники» («Id сотрудника») и «Кабинеты» («Номер кабинета»);
* сущность «Сотрудники» зависит от «Личности» («Id личности»), «Должности» («Id должности»), «Формы оклада» («Id формы оклада»);
* сущность «Пациенты» зависит от сущности «Личности» («Id личности»);
* сущность «Медицинские карты» зависит от сущностей «Приёмы» («Id приёма») и «Прейскурант» («Id услуги»).

3НФ представлена в табл. 7-22.

Таблица 7

Кабинеты

|  |  |
| --- | --- |
| Номер кабинета | Название кабинета |
| 101 | Стоматологический кабинет (терапевтический) |
| 102а | Кабинет ортодонтии |
| 202б | Кабинет стерилизационной обработки |
| 202б | Кабинет стерилизационной обработки |
| 202б | Кабинет стерилизационной обработки |

Таблица 8

Стоматологические материалы

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Id стом. материала | Название | Номер кабинета | Дата производства | Срок годности | Артикул | Id статуса | Id поставки |
| 1001 | Цемент стоматологический | 202б | 20.09.2024 | 20.09.2026 | GC-FIXGP-01 | 5001 | 9000 |

Таблица 9

Инструменты

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Id стом. материала | Название | Номер кабинета | Дата производства | Срок годности | Артикул | Id статуса | Id поставки |
| 2002 | Зеркало стоматологическое | 101 | 10.11.2024 | 10.11.2029 | 3M-FZ550-A2 | 5001 | 9001 |

Таблица 10

Хозяйственные принадлежности

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Id стом. материала | Название | Номер кабинета | Дата производства | Срок годности | Артикул | Id статуса | Id поставки |
| 3003 | Перчатки нитриловые | 202б | 01.03.2025 | 01.03.2026 | WDP-MPX-01 | 5002 | 9002 |

Таблица 11

Оборудование

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Id стом. материала | Название | Номер кабинета | Дата производства | Срок годности | Номер технического паспорта | Id статуса | Id поставки |
| 4001 | Бормашина "DentPro X5" | 102а | 01.06.2023 | 01.06.2028 | NSK-V370-KIT | 5003 | 9003 |
| 4002 | Световая лампа "CureLight 3000" | 202б | 10.11.2022 | 10.11.2027 | MPX/WDP-2024-0110 | 5004 | 9004 |

Таблица 12

Статусы ресурсов

|  |  |
| --- | --- |
| Id статуса | Название |
| 5001 | В наличии |
| 5002 | Почти закончились |
| 5003 | В эксплуатации |
| 5004 | Требует обслуживания |

Таблица 13

Поставки

|  |  |  |  |
| --- | --- | --- | --- |
| Id поставки | Название поставщика | Номер телефона поставщика | Дата поставки |
| 9000 | ООО "Стома-Поставка" | +7 (495) 123-45-67 | 2025-01-10 |
| 9001 | ООО "МедТехСнаб" | +7 (812) 234-56-78 | 2023-06-15 |
| 9002 | ИП Смирнов С.В. | +7 (903) 345-67-89 | 2025-03-05 |
| 9003 | ЗАО "ДенталИмпорт" | +7 (926) 456-78-90 | 2024-10-01 |
| 9004 | ООО "ТехДентЛайт" | +7 (499) 567-89-01 | 2022-11-25 |

Таблица 14

Должности

|  |  |
| --- | --- |
| Id должности | Название |
| 1 | Терапевт |
| 2 | Ортодонт |
| 3 | Медсестра |
| 4 | Имплантолог |
| 5 | Гигиенист |
| 10 | Стоматолог |

Таблица 15

Формы оклада

|  |  |
| --- | --- |
| Id формы оклада | Название |
| 1 | Процент от выручки |
| 2 | Фиксированная |
| 3 | Почасовая ставка |
| 4 | Сдельная оплата |

Таблица 16

Сотрудники

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id сотрудника | Id личности | Id должности | Id формы оклада | Оклад | Дата заключения договора | Договор |
| 1 | 7 | 1 | 1 | 10 | 18.04.2022 | - |
| 3 | 5 | 2 | 2 | 60 000 | 17.03.2023 | - |
| 4 | 6 | 3 | 3 | 350 | 19.02.2024 | - |
| 5 | 3 | 4 | 4 | 1 500 | 10.01.2025 | - |
| 2 | 7 | 5 | 2 | 55 000 | 01.01.2025 | - |
| 10 | 5 | 10 | 2 | 80 000 | 17.03.2024 | - |

Таблица 17

Планы лечения

|  |  |  |
| --- | --- | --- |
| Id плана | Id пациента | План |
| 100 | 771 | Лечение кариеса 1-го зуба |
| 200 | 772 | Удаление зуба мудрости |
| 300 | 773 | Профессиональная чистка зубов |
| 400 | 774 | Протезирование 2-го зуба |
| 500 | 773 | Отбеливание зубов |
| 600 | 780 | Лечение кариеса 2-го зуба |
| 700 | 780 | Профессиональная чистка зубов |

Таблица 18

Приёмы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id  приёма | Id  сотрудника | Номер  кабинета | Id  пациента | Дата и время | Заключение |
| 10 | 1 | 200 | 771 | 2025-06-01 10:00 | Успешное лечение |
| 20 | 3 | 102а | 772 | 2025-06-02 14:30 | Рекомендован контроль через 1 месяц |
| 30 | 4 | 202б | 773 | 2025-06-03 09:00 | Зубы в хорошем состоянии |
| 40 | 5 | 100 | 774 | 2025-06-04 11:15 | Протез установлен |
| 50 | 2 | 200 | 773 | 2025-06-05 15:45 | Процедура завершена |
| 600 | 3 | 101 | 780 | 2025-06-06 16:35 | - |
| 700 | 4 | 202б | 780 | 2025-06-06 14:00 | Повторить через месяц |

Таблица 19

Личности

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id личности | Фамилия | Имя | Отчество | Паспорт |
| 1 | Иванов | Алексей | Сергеевич | 4510 123456 |
| 2 | Смирнова | Ирина | Петровна | 4012 987654 |
| 3 | Сидоров | Михаил | Юрьевич | 4508 334455 |
| 4 | Петрова | Ольга | Викторовна | 4610 112233 |
| 5 | Петров | Сергей | Иванович | 4012 223344 |
| 6 | Кузнецова | Ольга | Викторовна | 4511 998877 |
| 7 | Васильева | Наталья | Петровна | 4508 556677 |
| 10 | Иванов | Иван | Иванович | 1111 111111 |

Продолжение табл. 19

|  |  |  |  |
| --- | --- | --- | --- |
| СНИЛС | ИНН | Номер телефона | Электронная почта |
| 123-456-789 00 | 770123456789 | +7 (915) 123-45-67 | Ivanov@mail.ru |
| 112-233-445 66 | 503212345678 | +7 (926) 234-56-78 | smirnova.irina@yandex.ru |
| 156-789-234 55 | 502134567890 | +7 (903) 567-89-01 | sidormih@yandex.ru |
| 145-678-321 44 | 780123456701 | +7 (985) 456-78-90 | petrova.olga@mail.ru |
| 112-344-556 66 | 503212345679 | +7 (926) 234-56-89 | petrov.serge@yandex.ru |
| 134-123-456 33 | 773405678902 | +7 (999) 345-67-90 | kuznetsova.olg@gmail.com |
| 156-789-123 55 | 502134567891 | +7 (903) 567-89-02 | vasilieva.nat@mail.ru |
| 123-456-789 10 | 109876543210 | - | iivanovich@gmail.com |

Таблица 20

Пациенты

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id пациента | Противопоказания | Id личности | Дата заключения | Дата окончания | Согласие на обработку персональных данных | Согласие на медицинское вмешательство |
| 771 | Аллергия на лидокаин | 1 | 01.06.2024 | 01.06.2025 | - | - |
| 772 | Хроническая гипертония | 2 | 01.01.2025 | 01.07.2025 | - | - |
| 773 | Нет | 3 | 01.12.2023 | 01.06.2025 | - | - |
| 774 | Сахарный диабет | 4 | 01.09.2024 | 01.09.2025 | - | - |
| 780 | Нет | 10 | 05.05.2025 | 06.06.2026 | - | - |

Таблица 21

Прейскурант

|  |  |  |  |
| --- | --- | --- | --- |
| Id услуги | Название | Стоимость, ₽ | Ед. измерения |
| 8001 | Лечение кариеса | 3 500 | 1 зуб |
| 8002 | Атравматичное удаление зуба под имплантацию | 5 000 | 1 зуб |
| 8003 | Подготовка пациента к приёму | 400 | 1 процедура |
| 8004 | Установка импланта (Straumann) | 55 000 | 1 имплант |
| 8005 | Профессиональная гигиена полости рта (Air Flow + ультразвук) | 4 000 | 1 процедура |

Таблица 22

Медицинские карты

|  |  |
| --- | --- |
| Id приёма | Id услуги |
| 10 | 8001 |
| 20 | 8002 |
| 30 | 8003 |
| 40 | 8004 |
| 50 | 8005 |
| 60 | 8001 |
| 70 | 8005 |

Чаще всего нормализацию завершают на 3НФ (продолжение может привести к созданию множества таблиц, ключей и связей, что усложнит работу с базой и снизит производительность СУБД).

Результатом логического проектирования является ER-диаграмма. В свою очередь, ER-диаграмма – это логическая модель предметной области. Она отображает отношения набора сущностей, хранящиеся в БД. Можно сказать, что ER-диаграммы объясняют логическую структуру базы данных [2].

1. ER-диаграмма

Для проектирования схем отношении используют так называемые «CASE-средства» – программные средства, поддерживающая процессы создания и сопровождения информационных систем, в частности баз данных.

Обычно для проектирования БД используют программу ERWin – средство моделирования диаграмм «сущность-связь» (рис. 1).

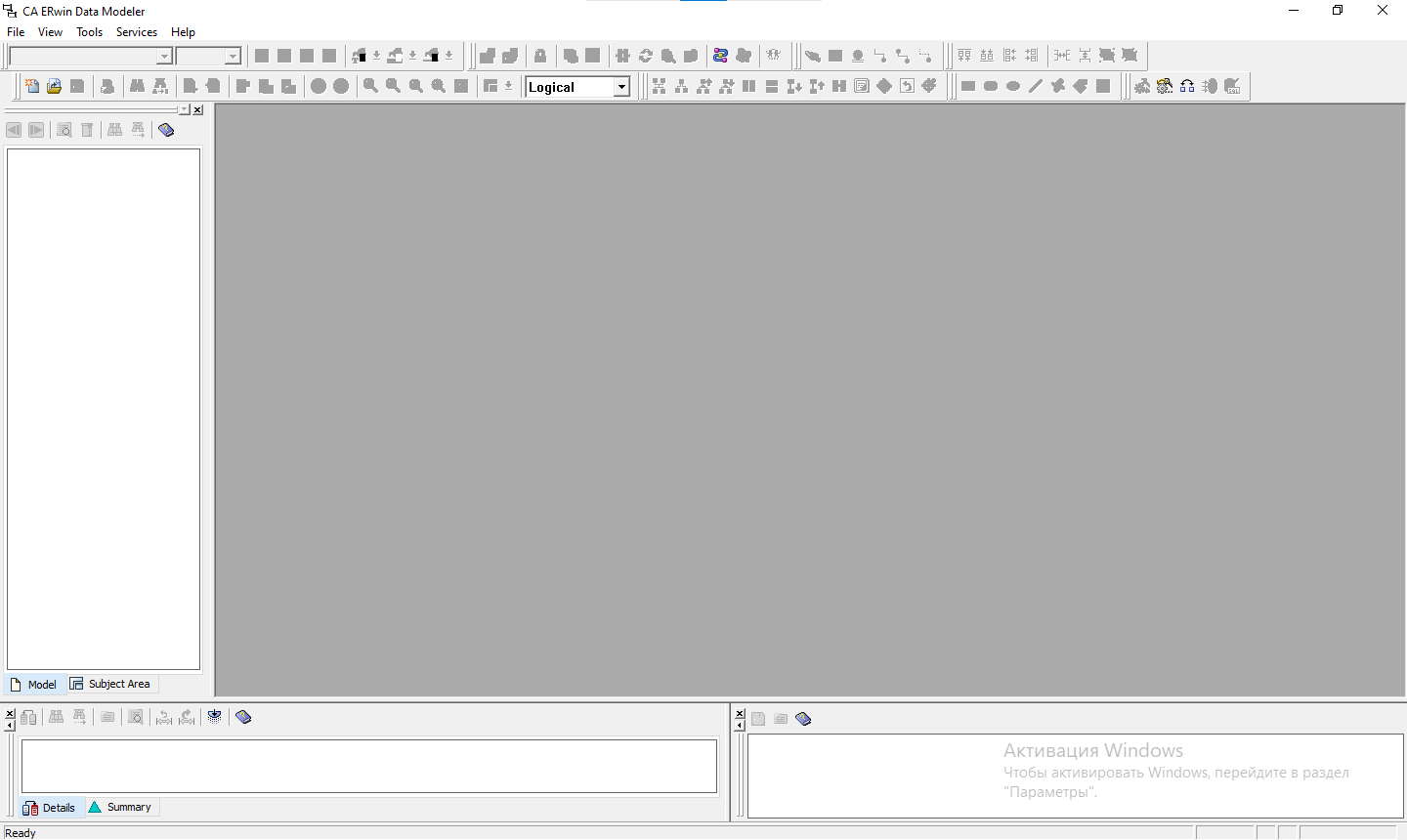


Рис. 1. Пример рабочего окна программы «ERwin»

Программа поддерживает множество СУБД, такие как «Oracle», «dBASE», «FoxPro», «Microsoft Access» и другие [7]. Модели представлены в нотациях «IDEF1Х» или «IE» («куриная лапка», «Crow’s Foot»), отражающих сущности предметной области и присущие им атрибуты, связи между ними и ограничения, есть поддержка индексов [7].

Пакет может осуществлять «прямой инжиниринг» (автоматическое создание запросов для конкретной СУБД) и «обратный инжиниринг» (преобразование БД в объекты для проектирования). Таким образом, появляется возможность как создание новой, так и редактирования уже существующей БД, имея возможность поддерживать синхронизацию между несколькими СУБД одновременно.

В ходе логического проектирования была создана диаграмма «сущность-связь» (или «ERD») следующим образом (рис. 2-10).

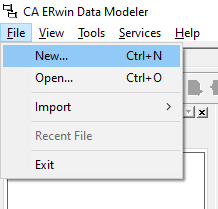


Рис. 2. Кнопка создания новой модели «New…»

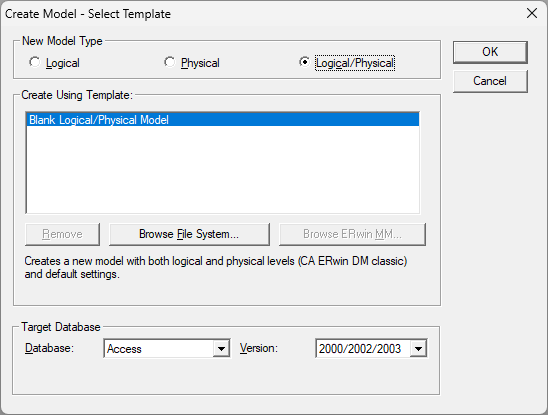


Рис. 3. Выбран «Logical/Physical»-тип модели и целевая БД «Access»

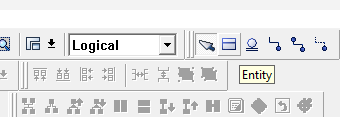


Рис. 4. Инструмент «Entity»

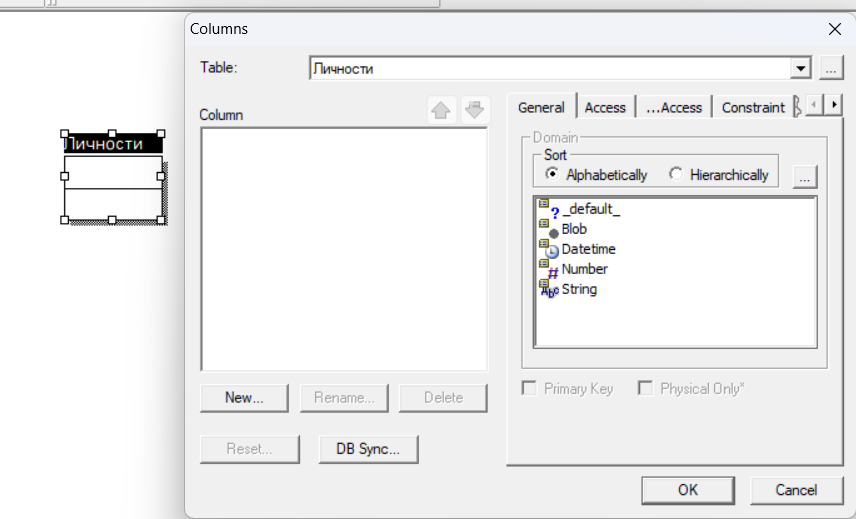


Рис. 5. Окно создания новых атрибутов новой сущности «Личности»

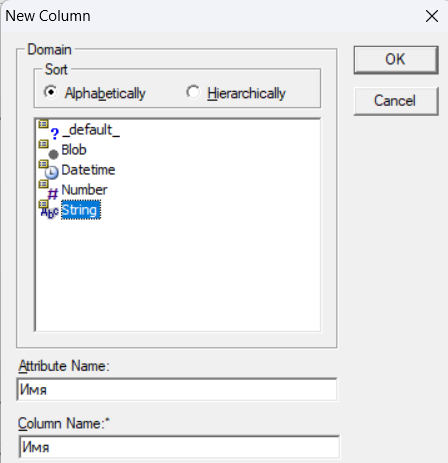


Рис. 6. Окно настройки нового атрибута

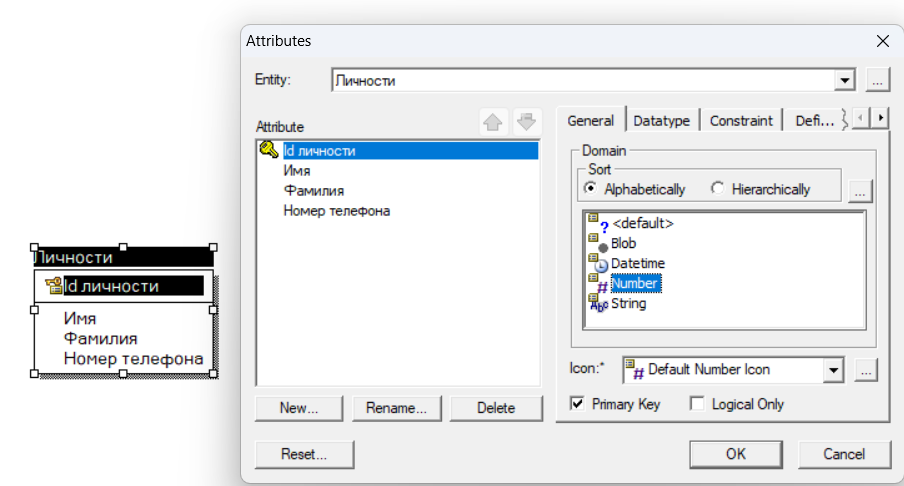


Рис. 7. Пример созданных атрибутов

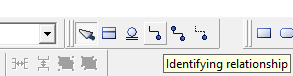


Рис. 8. Инструмент «Relationship»

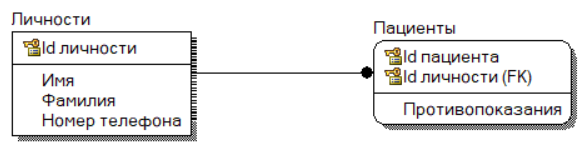


Рис. 9. Пример созданной связи между «Личностями» и «Пациентами»

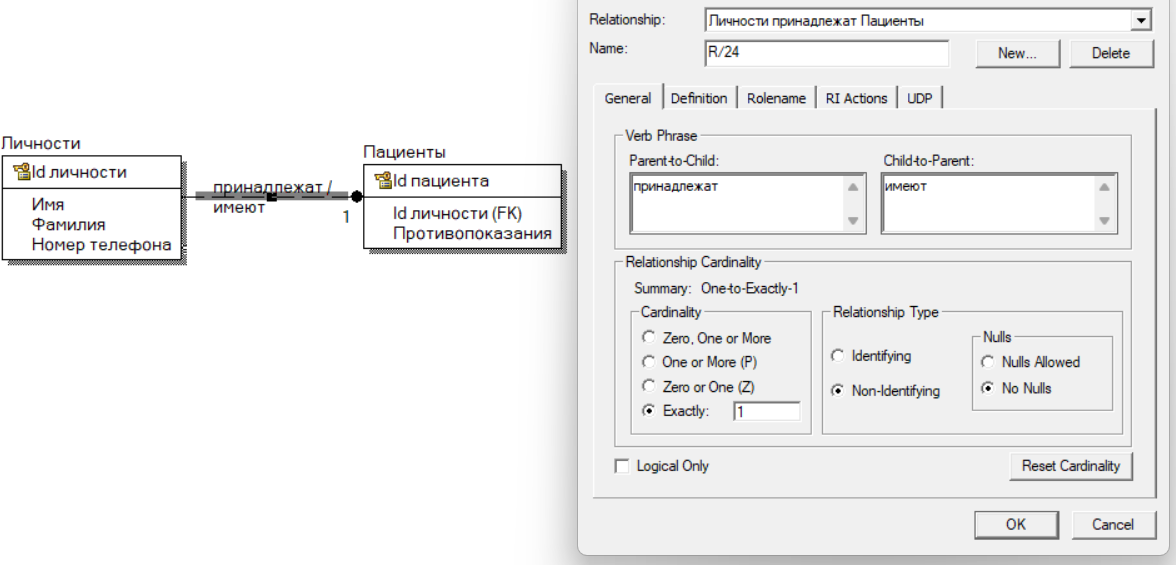


Рис. 10. Пример настройки итоговой связи

Итоговая ER-диаграмма представлена на рис. 11.

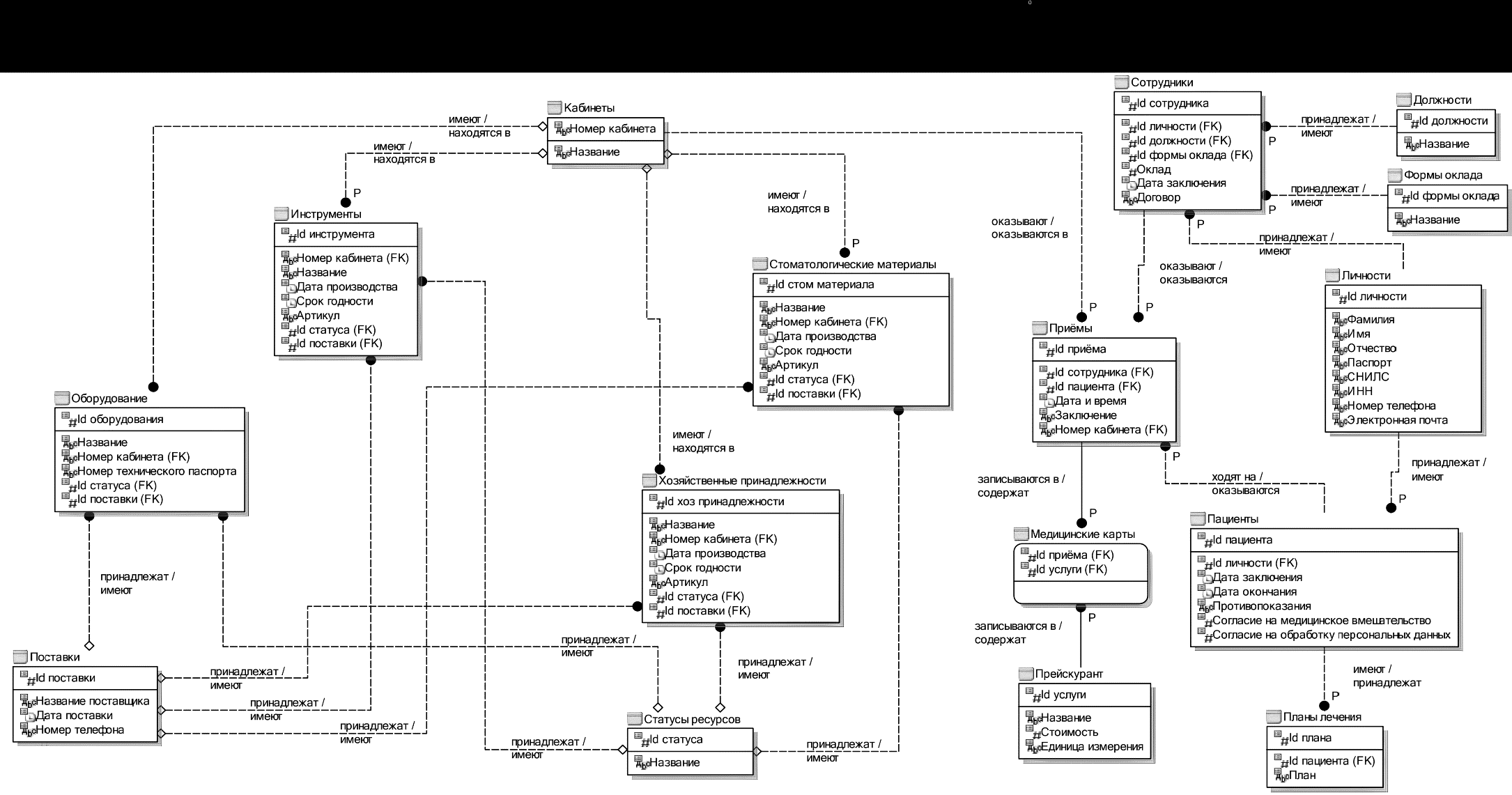


Рис. 11. ER-диаграмма «Стоматологической клиники» (нотация IDEF1X)

1. Физическое проектирование

Физическое проектирование – это создание схемы базы данных для конкретной СУБД. Физическая модель содержит всю информацию, необходимую для реализации конкретной БД [2].

Результатам физического проектирования является сама база данных. Так как стандартов на объекты БД не существует (нет стандарта на типы данных), физическая модель зависит от конкретной реализации СУБД. Следовательно, одной и той же логической модели могут соответствовать несколько разных физических моделей [2].Перед физическим проектированием необходимо выбрать СУБД. Примеры существующих СУБД описаны ниже.

* Oracle — ведущая коммерческая мульти-модельная система управления реляционными базами данных, поддерживающая документное хранение, графовые базы данных, RDF-хранилища, пространственные данные и векторные функции. Широко используется в корпоративных приложениях благодаря своей надёжности и масштабируемости. Когда использовать: крупные корпорации, банки, государственные учреждения [8].
* MySQL — это открытая реляционная СУБД, также поддерживающая документное хранение и пространственные данные. Популярна среди web-разработчиков и используется в многочисленных web-приложениях и онлайн-сервисах. Когда использовать: web-сайты, CMS, электронная коммерция, где важна скорость и простота [8].
* Microsoft SQL Server — это реляционная мульти-модельная СУБД от Microsoft, включающая поддержку документарного хранения, графовых баз данных и пространственных данных. Интегрируется с другими продуктами Microsoft и широко применяется в бизнес-среде. Когда использовать: корпоративные приложения, требующие сложной логики и аналитики [8].
* Microsoft Access — настольная реляционная система управления базами данных с графическим интерфейсом от корпорации Microsoft, входящая в состав пакета «Microsoft Office». Имеет широкий спектр функций, включая связанные запросы, связь с внешними таблицами и базами данных. Подходит для одиночных и малых рабочих групп, так как не требует разворачивать локальный сервер – всё работает из одного файла [8].
* PostgreSQL — это открытая реляционная мульти-модельная СУБД, поддерживающая документное хранение, графовые базы данных, пространственные данные и векторные функции. Известна своей расширяемостью и соответствием стандартам SQL, что делает её популярной среди разработчиков и исследователей. Поддерживает полнотекстовый поиск, репликацию, хранимые процедуры, JSON и многое другое. Когда использовать: сложные бизнес-приложения, требующие надёжности и гибкости [8].
* MongoDB – это документно-ориентированная мульти-модельная СУБД, поддерживающая хранение пространственных данных, функции поиска, временные ряды и векторные данные. Предназначена для работы с неструктурированными данными и широко используется в приложениях, требующих гибкости в структуре данных. Когда использовать: быстро меняющиеся схемы, хранение неструктурированных данных [8].

Несмотря на разнообразие возможностей остальных СУБД, была выбрана СУБД «Microsoft Access». У неё простой интерфейс и есть возможность внести изменения в схему отношений и содержимое с помощью одноимённой программы.

Перед физическим моделированием необходимо заранее создать файл базы данных формата «.mdb» и подключиться к нему (рис. 12-13).

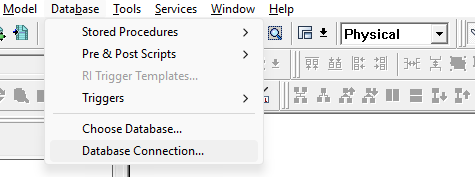


Рис. 12. Переход в «Database Connection»

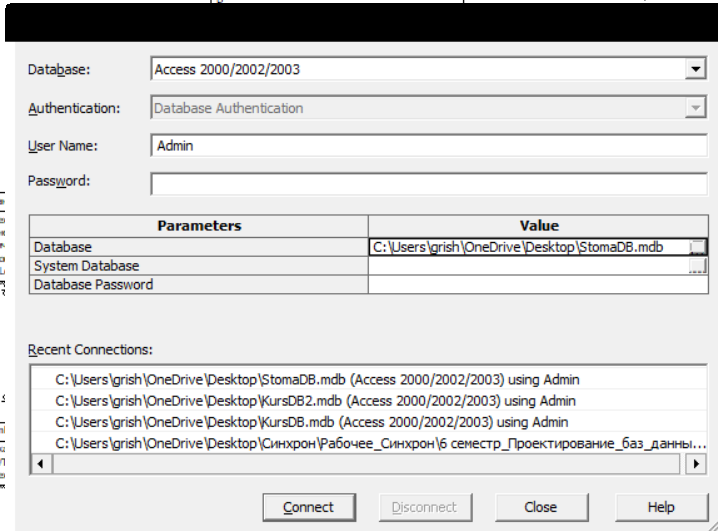


Рис. 13. Пример данных для подключения

Необходимо определиться с типами данных для физической модели. Примеры типов данных представлены ниже (рис. 14-16).

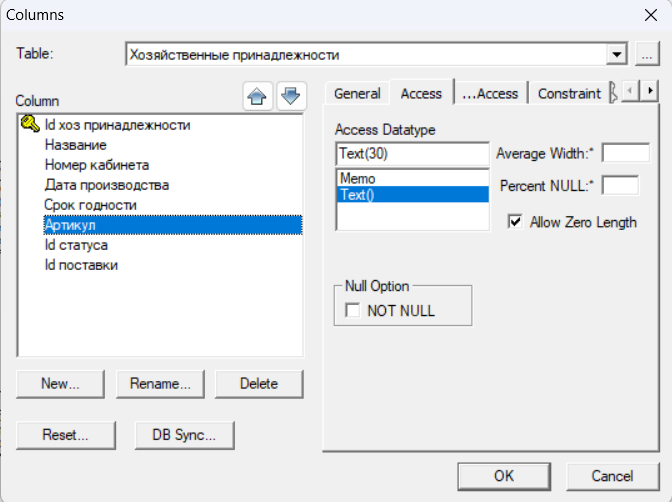


Рис. 14. Тип данных для «Хозяйственные принадлежности.Артикул»

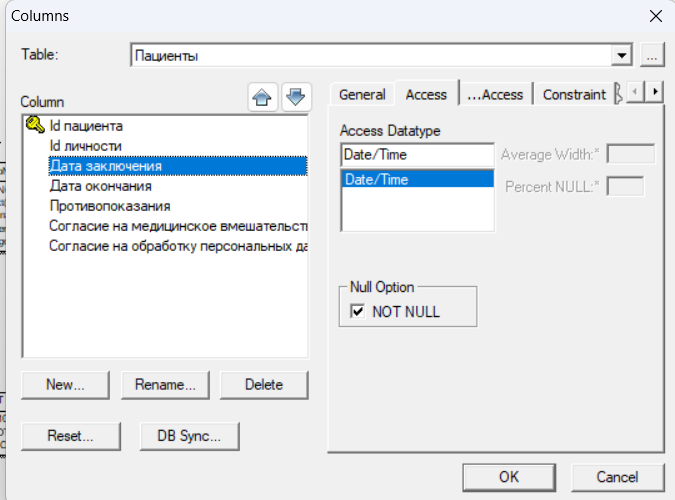


Рис. 15. Тип данных для «Пациенты.Дата заключения»

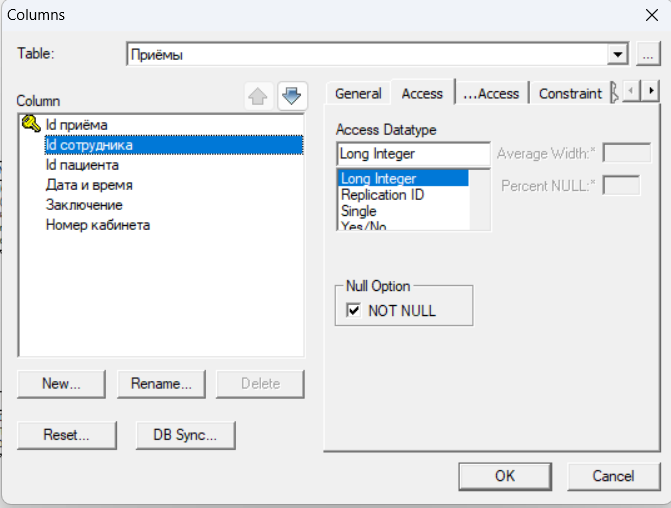


Рис. 16. Тип данных для «Приёмы.Id сотрудника»

Была создана следующая физическая модель (рис. 17).

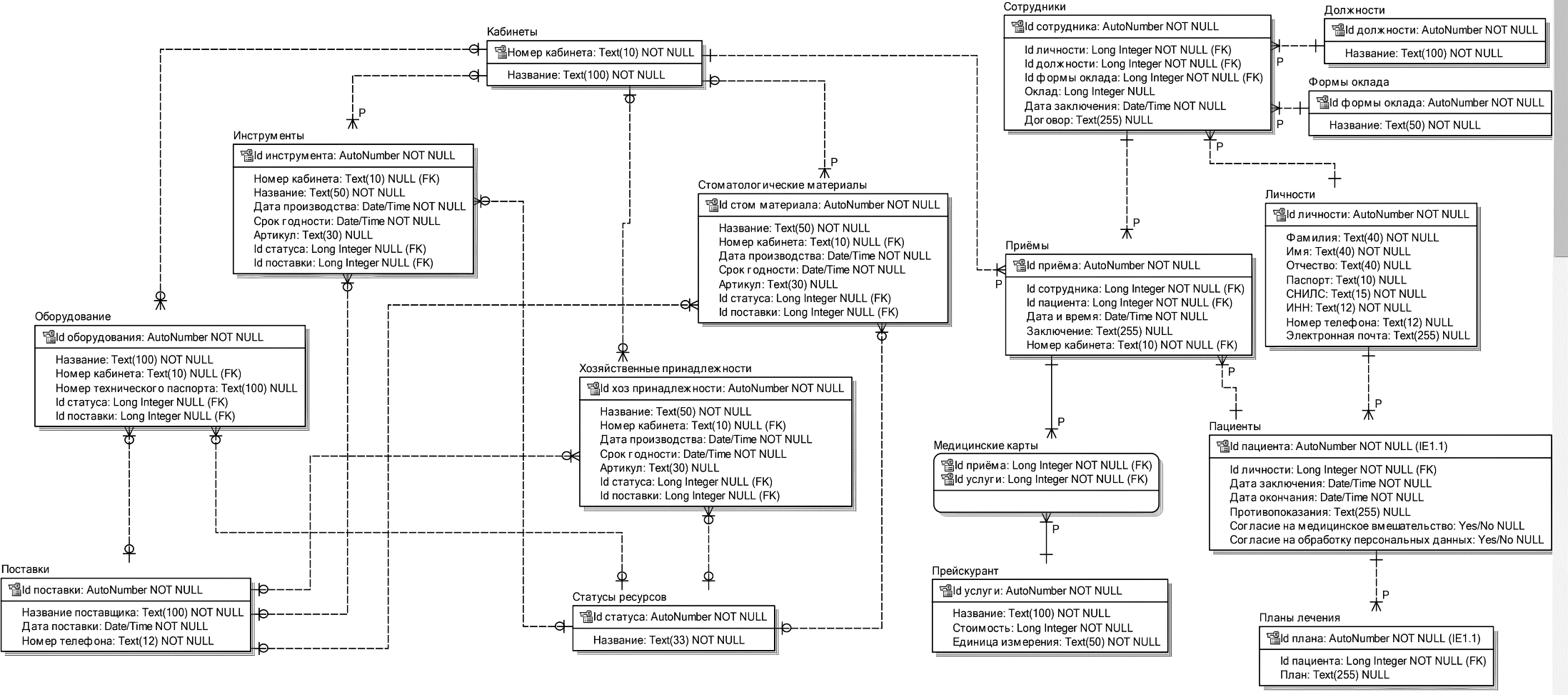


Рис. 17. Физическое представление ER-диаграммы (Crow’s Foot)

В ходе физического проектирования была создана база данных «Microsoft Access» из логической модели с помощью «прямого инжиниринга» (рис. 18-23).

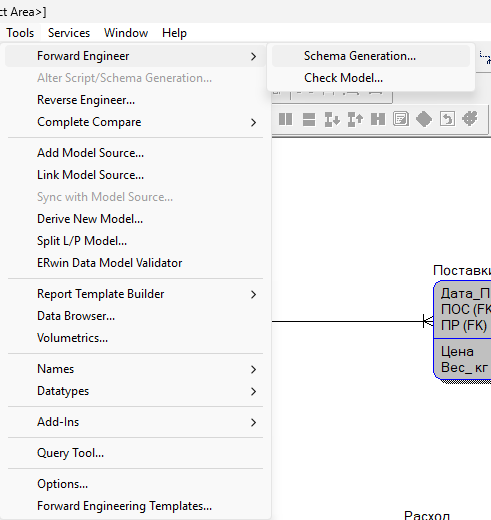


Рис. 18. «Schema Generation…»



Рис. 19. «Generate…»

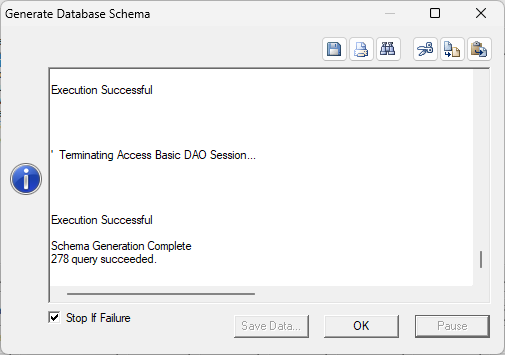


Рис. 20. Окно после генерации (прошла успешно)

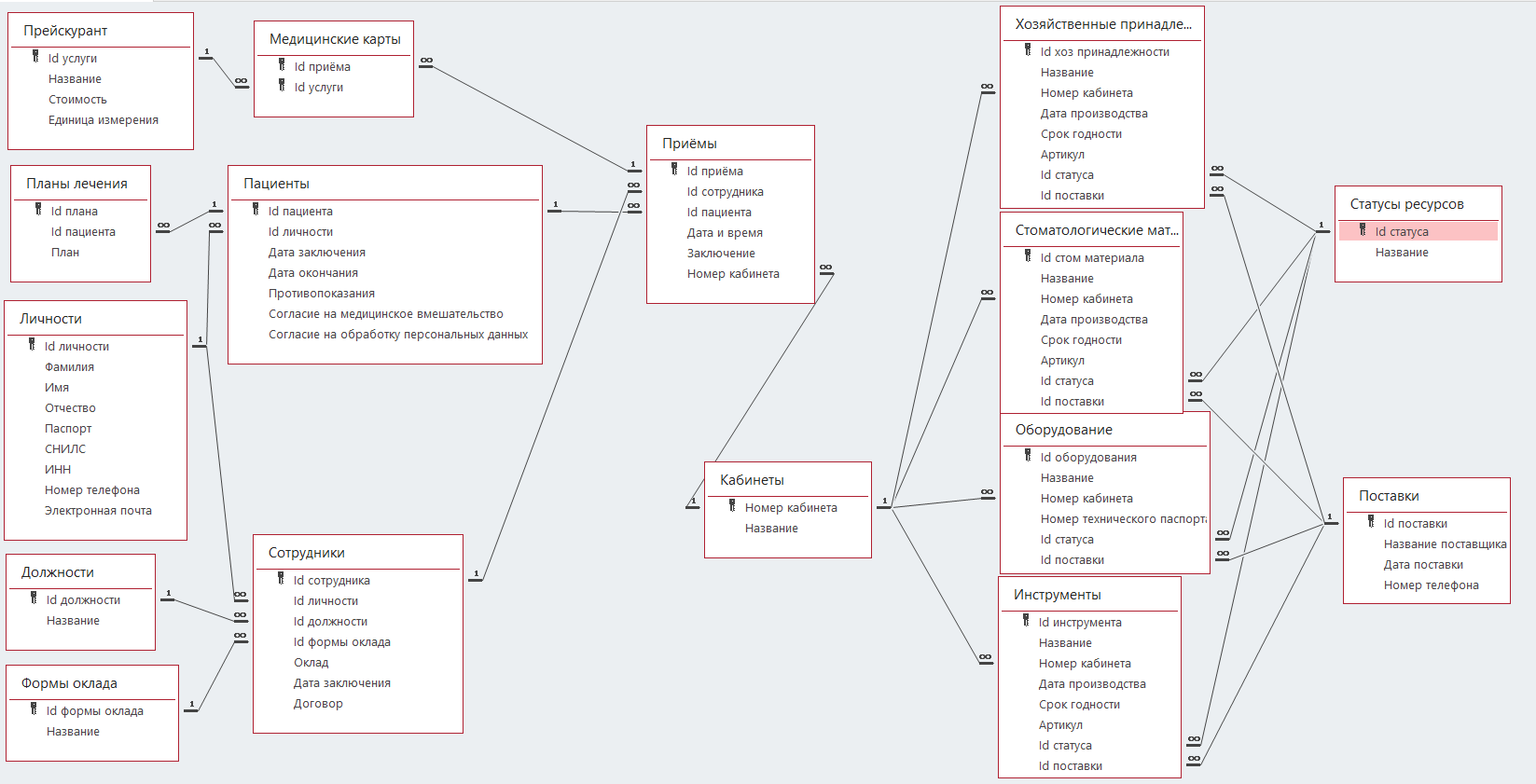


Рис. 21. Схема данных в «Microsoft Access»

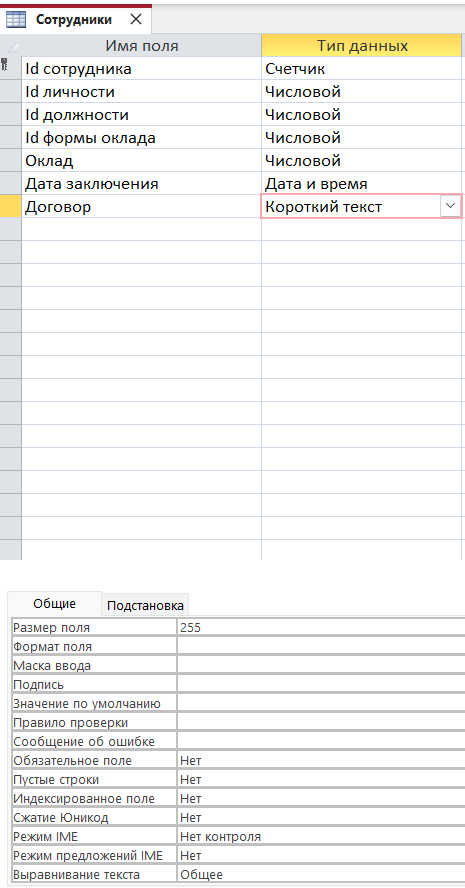


Рис. 22. Таблица «Сотрудники» в «Конструкторе таблиц»

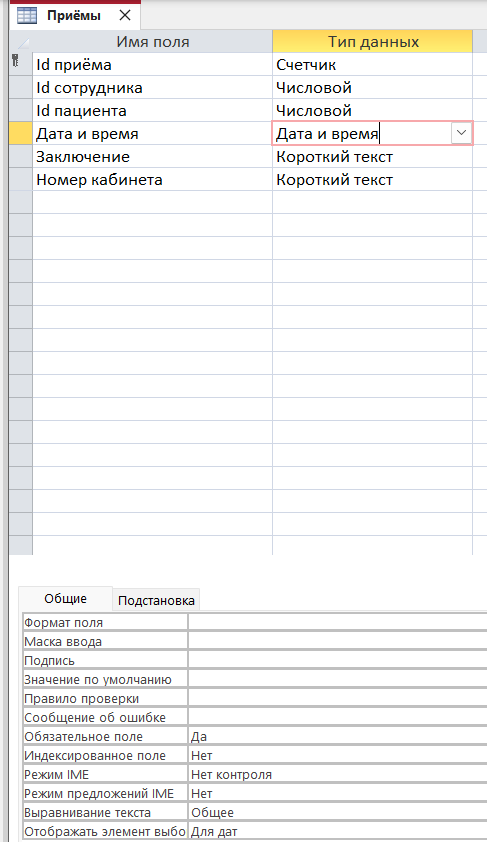


Рис. 23. Таблица «Приёмы» в «Конструкторе таблиц»

Нотации «Microsoft Access» и «ERwin» отличаются от нотаций Мартина (Crow’s Foot) и IDEF1X (см. рис. 17, 21), однако они близки к общепринятым (знак бесконечности «∞» вместо «куриной лапки», не показывается «сила» сущностей в виде скруглённых углов прямоугольников, отображение является ли связь идентифицирующей отсутствует в «Microsoft Access»).

1. Описание программы

Для написания программы будут использоваться язык высокого уровня «C#», библиотека «OLE DB» и платформа «WPF»

Язык «C#» – это один из самых мощных, быстро развивающихся и востребованных языков в ИТ-отрасли. В настоящий момент на нем пишутся самые различные приложения: от небольших десктопных программ до крупных web-порталов и web-сервисов, обслуживающих ежедневно миллионы пользователей. Объектно-ориентированный подход позволяет решить задачи по построению крупных, но в тоже время гибких, масштабируемых и расширяемых приложений [9].

«WPF» – это комплексный набор функций разработки приложений, которые включают в себя язык «XAML», элементы управления, привязку к данным, макет, двумерную и трёхмерную графику, анимацию, стили, шаблоны, документы, мультимедиа, текст и типографические функции. «WPF» является частью «.NET», поэтому можно создавать приложения, включающие другие элементы «.NET API» [10].

«OLE DB» — это высокопроизводительная технология баз данных на основе «COM». Он предоставляет общий способ доступа к данным независимо от формы, в которой он хранится. В типичной бизнес-ситуации большое количество информации не хранится в корпоративных базах данных. Эти сведения находятся в файловых системах, индексированные последовательные файлы, личные базы данных (например, «Access»), электронные таблицы (например, «Excel»), приложения планирования проектов и электронная почта (например, «Outlook»). «OLE DB» позволяет получить доступ к любому типу хранилища данных, если в хранилище данных есть поставщик «OLE DB» [11].

Для разработки программы использовалась среда Visual Studio 2022, так как она поддерживает множество языков программирования и библиотек, удобную отладку и тестирование, а также подсветку синтаксиса (рис. 24).

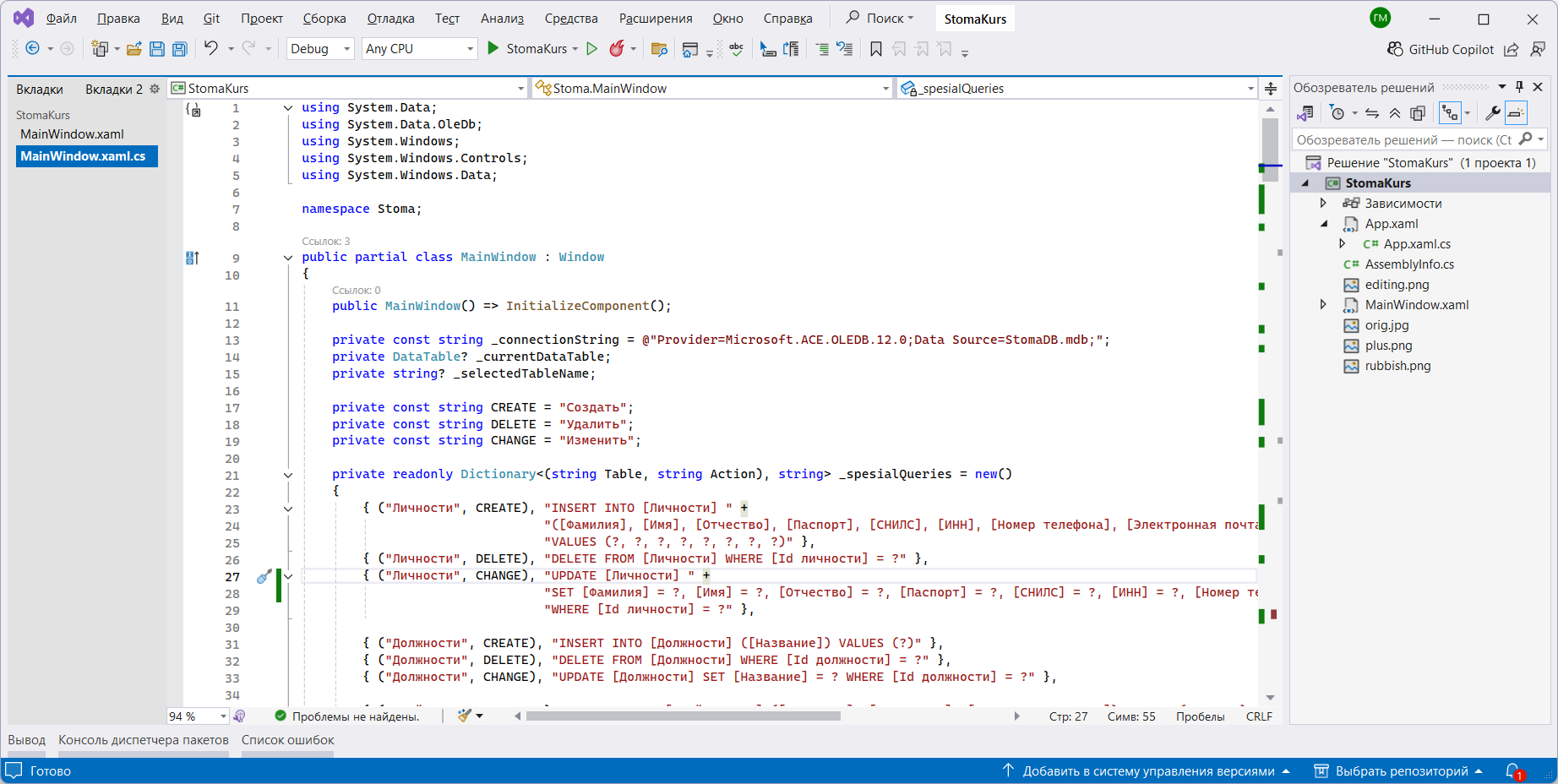


Рис. 24. Пример рабочего окна Visual Studio 2022

«OLE DB» позволяет разрабатывать приложения, которые обращаются к различным источникам данных, независимо от того, является ли они СУБД или нет. «OLE DB» обеспечивает универсальный доступ с помощью COM-интерфейсов, поддерживающих соответствующие функции СУБД для данного источника данных [11].

Чтобы использовать «OLE DB» в программе на «C#», необходимо подключить соответствующую библиотеку (рис. 25).



Рис. 25. Код для подключения библиотеки OleDb

Для взаимодействия с базой данных используют специальный язык – SQL. SQL («Structured Query Language» — «язык структурированных запросов») – это непроцедурный декларативный язык программирования, используемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных [2].

Запрос к БД осуществляется вручную написанными SQL-запросами. Преимущество: ручной контроль и специфика запросов. Недостаток: нужно знать синтаксис SQL-запросов и низкое повторное использование кода.

Пример запросов на чтение всех данных из таблиц БД представлен на рис. 10.



Рис. 26. Пример запросов на чтение таблиц БД

Валидация – это процесс проверки данных на соответствие ожидаемым форматам, типам, диапазонам значений и бизнес-правилам перед их сохранением в БД.

Значение по умолчанию – это значение, которое нужно ввести в поле записи, если никакое другое значение не задано явным образом во время ввода данных.

В программе не были реализованы валидация вводимых пользователем данных и значения по умолчанию явным образом, однако реализован вывод предупреждений от самой базы данных. Пример конструкции на добавление данных в БД (не зависит от конкретной таблицы) представлен на рис. 27.



Рис. 27. Конструкция добавления данных в таблицу БД

Изменение и удаление данных из таблиц БД реализованы аналогичным образом (рис. 28-29).



Рис. 28. Конструкция удаления данных из таблиц БД



Рис. 29. Конструкция изменения данных в таблице БД

Реализованы следующие запросы добавления, удаления и изменения данных в таблицах БД (рис. 30).



Рис. 30. Пример результата запроса всех записей из таблицы «Личности»

Для вывода данных в виде таблиц использовался компонент «DataGrid». Для взаимодействия с данными использовался тип «DataTable». Для вывода списка всех доступных запросов использовался компонент «ListBox».

Загрузка данных из таблиц базы данных осуществляется следующими конструкциями (рис. 31-32).



Рис. 31. Конструкция инициализации и реакции на выбор запроса из списка



Рис. 32. Конструкция загрузки данных из таблиц БД

Итоговый интерфейс программы в среде разработки представлен на рис. 33.

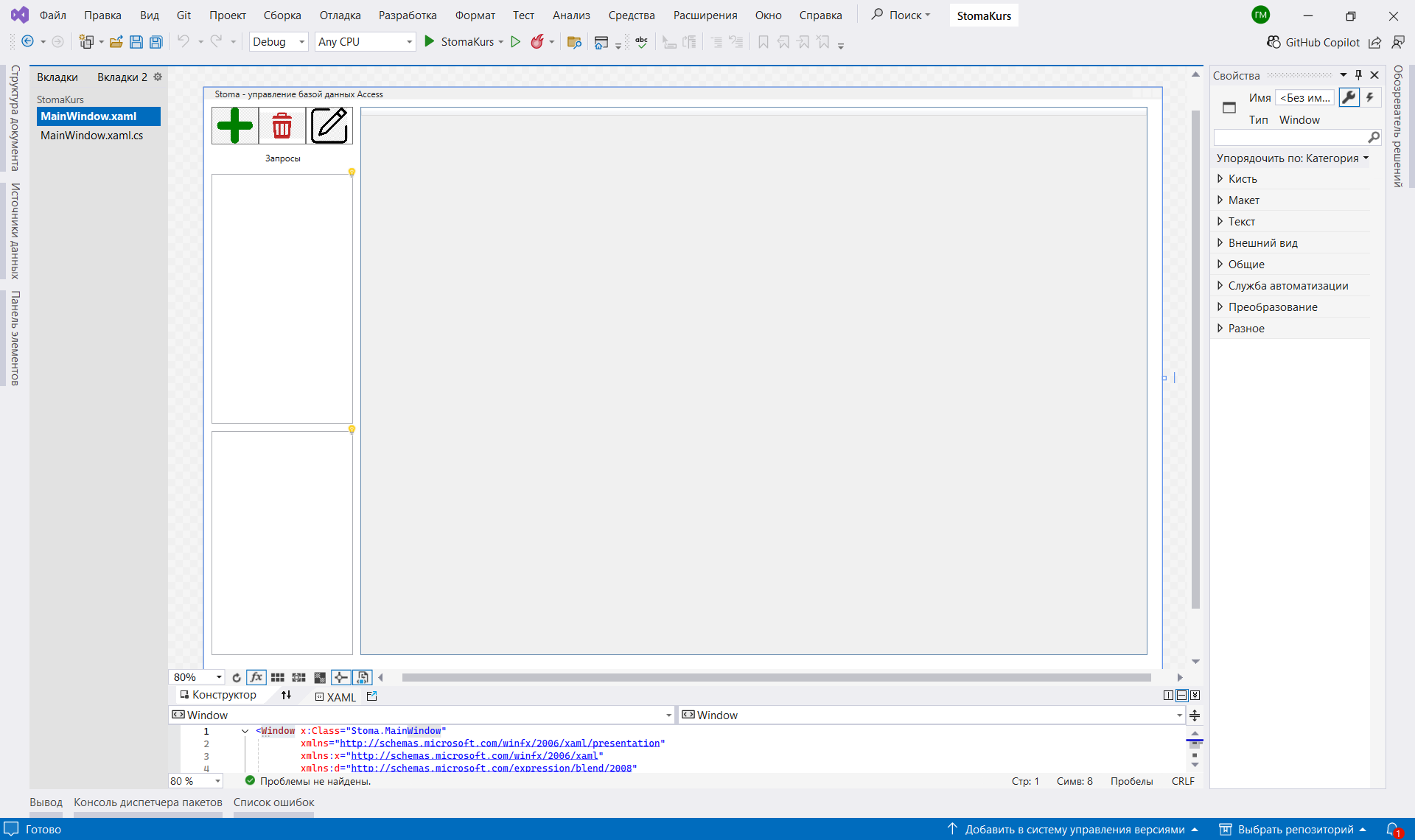


Рис. 33. Интерфейс программы в среде Visual Studio 2022

Пример вывода данных из таблицы «Личности» представлен на рис. 34-35.

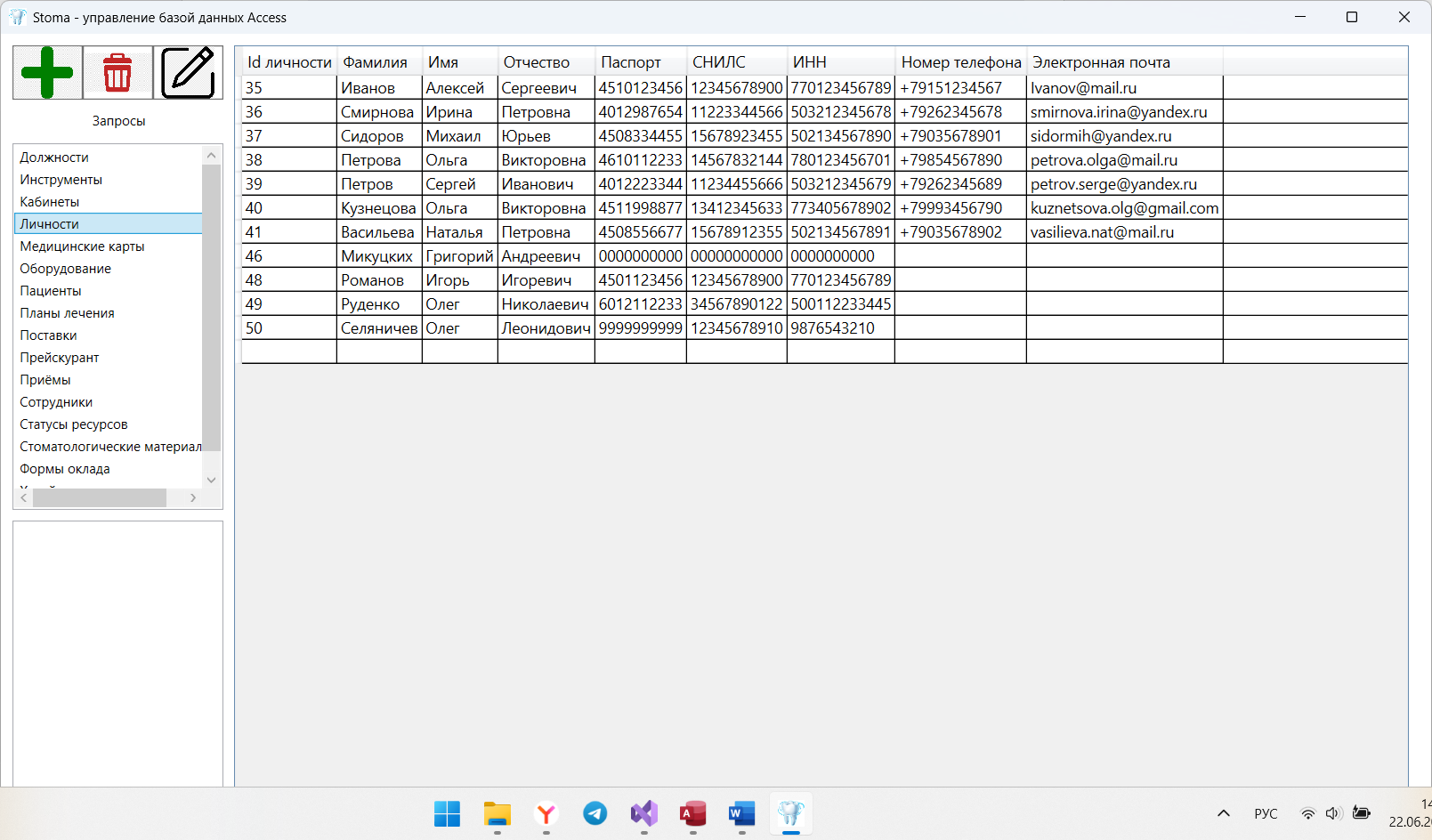


Рис. 34. Пример вывода данных из таблицы «Личности»



Рис. 35. Пример содержимого таблицы «Личности» в «Microsoft Access»

Чтобы было удобнее выбирать данные типа «Сотрудник» (вместо «Id сотрудника»), «Поставка» (вместо «Id поставки»), «Личность» (вместо «Id личности») и прочее, что требует ввода «Id», реализован ввод таковых через выпадающие списки (рис. 36, см. рис. 32).



Рис. 36. Пример конструкции, формирующей выпадающие списки

Запуск программы осуществляется запуском исполняемого файла «Stoma.exe». Рядом с исполняемым файлов находится база данных «Microsoft Access» «Стоматологическая клиника» «StomaDB.mdb» (рис. 37)

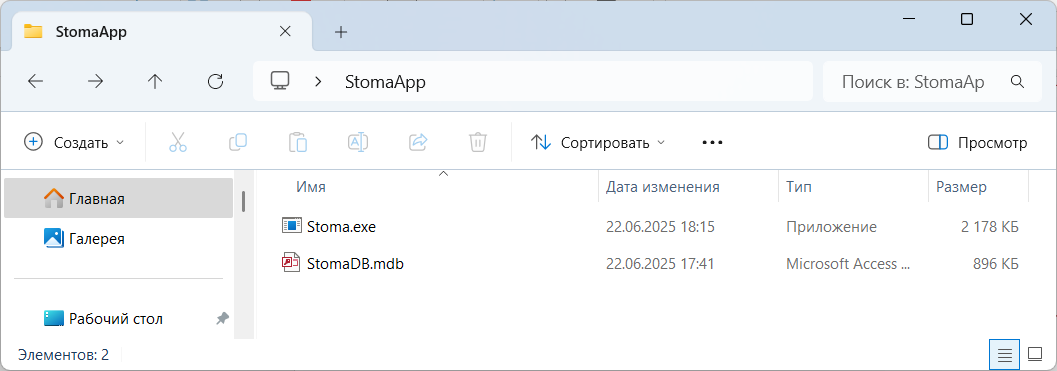


Рис. 37. Итоговый каталог программы

Подробнее про код программы в разделе «Приложение 2. Текст программы».

Подробнее про установку и использование программы в разделе «Приложение 3. Руководство пользователя».

# Заключение

В ходе работы были созданы база данных и программа для взаимодействия с ней в рамках предметной области «Стоматологическая клиника»

На этапе проектирования был проведён анализ предметной области, в ходе которого определены основные элементы системы и установлены связи между ними, после чего построена третья нормальная форма разрабатываемой базы данных.

На основе нормализованной базы данных была создана логическая модель – ER-диаграмма. По логической модели создана физическая модель – база данных «Microsoft Access».

Разработано приложение, написанное на языке высокого уровня «C#» на платформе «WPF», которое обеспечивает пользователю базовые возможности взаимодействия с БД – просмотр, добавление, изменение и удаление.

# Список литературы

1. Методика и организация самостоятельной работы студентов / Е.В. Ершов, д-р техн. наук, проф.; Л.Н. Виноградова, канд. техн. наук; В.В. Селивановских, канд. техн. наук, доцент; О.Л. Селяничев, канд. техн. наук, доцент; О.В. Юдина, канд. техн. наук; И.А. Варфоломеев, канд. техн. наук; Е.В. Майтама; О.С. Сальникова; Н.Е. Сивков; вед. ред. Г.В. Иванова, вед. тех. редактор М.Н. Авдюхова Е.В. – ФГБОУ ВПО «Череповецкий государственный университет» 162600 г. Череповец, пр. Луначарского, 5., 2015. – 243 c.
2. Лекции «Базы данных» ; О.Л. Селяничев.
3. Основы проектирования баз данных: учеб. пособие / Д.А. Попова-Коварцева, Е.В. Сопченко. – Самара: Изд-во Самарского университета, 2019. – 112 с.: ил.
4. Сухомлинов, А.И. Инфологическое моделирование : учебно-методическое пособие : для студентов, обучающихся по направлению 09.03.03 «Прикладная информатика» / А.И. Сухомлинов. - Владивосток : Издательство Дальневосточного федерального университета, 2021. – [36 с.].
5. Научная электронная библиотека – Электронный ресурс. – URL: «https://monographies.ru/en/book/section?id=16234».

Дата обращения: 18.05.25.

1. НОУ ИНТУИТ | Основы проектирования реляционных баз данных. Лекция 3: Что такое проектирование баз данных – Электронный ресурс. – URL: «https://intuit.ru/studies/courses/1095/191/lecture/4971».

Дата обращения: 18.05.25.

1. Проектирование баз данных с помощью CASE-средства ERwin – Электронный ресурс. – URL: «https://studfile.net/preview/7814448/».

Дата обращения: 20.06.25.

1. Самые популярные СУБД в начале 2025 года. | Python.Study – Электронный ресурс. – URL: «https://db-engines.com/en/».

Дата обращения: 18.05.25.

1. C# и .NET | Введение – Электронный ресурс. – URL: «https://metanit.com/sharp/tutorial/1.1.php».

Дата обращения: 18.05.25.

1. Что такое Windows Presentation Foundation - WPF | Microsoft Learn – Электронный ресурс. – URL: «https://learn.microsoft.com/ru-ru/dotnet/desktop/wpf/overview/».

Дата обращения: 20.06.25.

1. Общие сведения о программировании OLE DB | Microsoft Learn – Электронный ресурс. – URL: «https://learn.microsoft.com/ru-ru/cpp/data/oledb/ole-db-programming-overview?view=msvc-170».

Дата обращения: 18.05.25.

# Приложение 1

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий

наименование института (факультета)

Математическое и программное обеспечение ЭВМ

наименование кафедры

Проектирование баз данных

наименование дисциплины в соответствии с учебным планом

УТВЕРЖДАЮ

Зав. кафедрой МПО ЭВМ,

д.т.н., профессор \_\_\_\_\_\_\_\_\_\_ Ершов Е.В.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г.

ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

«СТОМАТОЛОГИЧЕСКАЯ КЛИНИКА»

Техническое задание на курсовую работу

Листов 5

|  |  |
| --- | --- |
| Руководитель | Селяничев О.Л. |
|  | Ф.И.О преподавателя |
| Исполнитель |  |
| студент | 1ПИб-02-1оп-22 |
|  | группа |
|  | Микуцких Г.А. |
|  | Фамилия, имя, отчество |

2025 год

Введение

Стоматологические клиники оказывают широкий спектр услуг, имеют разнообразное оборудование, штат сотрудников различных специализаций (терапевты, хирурги, ортодонты, ортопеды и др.), инструменты (одноразовые и многоразовые), расходники и препараты. Для всего нужен строгий учёт и чёткая организация, которая раньше велась в бумажном виде без какой-либо возможности автоматизации процессов и специфики запросов.

Таким образом, целью курсовой работы является создание базы данных «Стоматологическая клиника» и программное обеспечение для работы с ней.

1. Основания для разработки

Основанием для разработки является задание на курсовую работу по дисциплине «Проектирование баз данных», выданное на кафедре МПО ЭВМ ИИТ ЧГУ.

Дата утверждения: 26 февраля 2025 года.

Наименование темы разработки: «Проектирование базы данных "Стоматологическая клиника"».

2. Назначение разработки

База данных позволит автоматизировать процессы хранения ключевой информации внутри стоматологической клиники.

3. Требования к программе

3.1. Требования к функциональным характеристикам

Программа должна обладать следующими основными качествами:

* способ взаимодействия с программой – мышь и клавиатура;
* результаты работы программы представляются в виде текста и таблиц;
* реализация CRUD-операций («Create», «Read», «Update», «Delete») над содержимым базы данных;
* база данных должна состоять из 10-15 реляционных таблиц;
* в программе предусмотреть возможность реализации 25-30 запросов.

3.2. Требования к надёжности

К программе предъявляются следующие требования надёжности:

1. обеспечивать целостность и согласованность данных;
2. предотвращать возможные конфликты и ошибки, возникающие при взаимодействии пользователя с базой данных или в ходе выполнения программы не должны влиять на состояние базы данных.

3.3. Условия эксплуатации

Предъявляются с условиями эксплуатации персонального компьютера пользователя, в частности:

* температура окружающей среды – +5-35 °С, относительная влажность – (60±20)%;
* ПК размещён не ближе 1 метра от источников сильных электромагнитных излучений;
* электропитание от однофазной сети переменного тока напряжением (220±20)В и частотой 49–51 Гц;
* нет подключения к бесперебойному источнику питания.

3.4. Требования к составу и параметрам технических средств

ЭВМ должна соответствовать следующим параметрам:

* 64-разрядный (x64) процессор с тактовой частотой 1 ГГц или выше;
* 2 Гб ОЗУ, 128 Гб свободного места на накопителе (HDD, SSD);
* графическое устройство DirectX 9 с драйвером WDDM 1.0 или более поздней версии;
* поддержка клавиатуры и компьютерной мыши;
* поддержка монитора разрешением 1920×1080 пикселей.

3.5. Требования к информационной и программной совместимости

Программа должна корректно работать на ОС Windows 10 и выше с установленным пакетом «Microsoft Access».

3.6. Требования к маркировке и упаковке

Архив с готовым к запуску ПО, документация и исходный код программы поставляются на USB-флеш-накопителе.

3.7. Требования к транспортированию и хранению

Всем пользователям запрещается осуществлять действия под видом оригинального ПО, нарушающие статью 273 «Создание, использование и распространение вредоносных компьютерных программ».

3.8. Специальные требования

Схожий опыт использования с программами пакета «Microsoft Office», в частности «Word» и «Access».

4. Требование к программной документации

4.1. Содержание расчётно-пояснительной записки

Программная документация должна содержать расчётно-пояснительную записку, содержание которой:

Титульный лист

Оглавление

Введение

1. Описание предметной области

2. Инфологическое проектирование

3. Даталогическое проектирование

4. Описание программы

Заключение

Список литературы

Приложение 1. Техническое задание

Приложение 2. Текст программы

Приложение 3. Руководство пользователя

4.3. Требования к оформлению

Элементы курсовой работы должны быть оформлены в соответствии с табл. П1.1.

Таблица П1.1

Требования к оформлению

|  |  |
| --- | --- |
| Элемент | Требования |
| Документ | 1. Печать на отдельных листах формата А4 (210х297 мм); оборотная сторона не заполняется; листы нумеруются. Печать возможна ч/б.  2. Файлы предъявляются на компакт-диске: РПЗ с ТЗ; программный код.  3. Листы и диск в конверте вложены в пластиковую папку скоросшивателя. |
| Страницы | 1. Ориентация – книжная; отдельные страницы, при необходимости, альбомная.  2. Поля: верхнее, нижнее – по 2 см, левое – 3 см, правое – 1 см. |
| Абзацы | Межстрочный интервал – 1.0, перед и после абзаца – 0. |
| Шрифты | Кегль – 14. В таблицах шрифт 12. Шрифт программного кода – 8 (возможно в 2 колонки). |
| Рисунки | Подписывается под ним по центру: Рис. Х. Название  В приложениях: Рис. ПX.X. Название |
| Таблицы | 1. Подписывается: над таблицей, выравнивание по правому: «Таблица Х».  2. В следующей строке по центру Название  3. Надписи в «шапке» (имена столбцов, полей) – по центру.  4. В теле таблицы (записи) текстовые значения – выровнены по левому краю, числа, даты – по правому. |

5. Стадии и этапы разработки

Курсовая работа и программа будут разрабатываться в течение следующих этапов (табл. П1.2).

Таблица П1.2

Стадии и этапы разработки

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование  этапа разработки | Сроки разработки | Результат выполнения | Отметка о выполнении |
| Оформление технического задания | 28.02.25 | Составлено техническое задание |  |
| Изучение предметной области | 01.03.25 – 30.03.25 | Изучена проблема предметной области |  |
| Инфологическое проектирование | 04.03.25 – 30.03.25 | Определены основные сущности и их атрибуты |  |
| Даталогическое проектирование | 01.04.25 – 30.04.25 | Проведена нормализация, построены логическая и физическая модели |  |
| Кодирование | 15.04.25 – 20.05.25 | Программа написана |  |
| Тестирование | 20.04.25 – 20.05.25 | Ошибки программы исправлены |  |
| Оформление расчётно-пояснительной записки | 15.04.25 – 22.05.25 | Оформлена расчётно-пояснительная записка |  |

6. Порядок контроля и приёмки

Курсовая работа будет приниматься в течение этапов в табл. П1.3.

Таблица П1.3

Порядок контроля и приёмки

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование контрольного этапа выполнения курсовой работы | Сроки  контроля | Результат выполнения | Отметка о приёмке результата контрольного этапа |
| Сдача технического задания | 28.02.25 | Техническое задание принято |  |
| Показ логической модели | 30.03.25 | Логическая модель утверждена |  |
| Показ физической модели | 30.04.25 | Физическая модель принята |  |
| Демонстрация приложения | 20.05.25 | Приложение принято |  |
| Исправление ошибок | 20.05.25 | Ошибки учтены и исправлены |  |
| Сдача расчётно-пояснительной записки | 22.05.25 | Расчётно-пояснительная записка принята |  |
| Защита курсовой работы | 22.05.25 | Курсовая работа защищена |  |

# Приложение 2

Текст программы.

Файл «MainWindow.xaml».

<Window x:Class="Stoma.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:Stoma"

mc:Ignorable="d"

Title="Stoma - управление базой данных Access" Loaded="InitTableDefault" Width="1300" Height="790" Icon="/orig.jpg">

<Grid x:Name="All" Margin="10,10,10,10">

<Grid.RowDefinitions>

<RowDefinition Height="51\*"/>

<RowDefinition Height="37\*"/>

<RowDefinition Height="347\*"/>

<RowDefinition Height="319\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="200\*"/>

<ColumnDefinition Width="1100\*"/>

</Grid.ColumnDefinitions>

<ListBox SelectionChanged="TablesDefault\_SelectionChanged" x:Name="TablesDefault" Margin="0,3,5,5" RenderTransformOrigin="0.5,0.5" Grid.Row="2" SelectedIndex="0">

<ListBox.RenderTransform>

<TransformGroup>

<ScaleTransform ScaleY="1"/>

<SkewTransform/>

<RotateTransform/>

<TranslateTransform/>

</TransformGroup>

</ListBox.RenderTransform>

</ListBox>

<DataGrid x:Name="Tables" FontSize="14" AutoGenerateColumns="False" Grid.Column="1" Margin="5,0,10,10" Grid.RowSpan="4"/>

<Grid Margin="0,0,5,0">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<Button x:Name="Add" Click="Add\_Click" Grid.Column="0">

<Button.Background>

<ImageBrush Stretch="Uniform" ImageSource="/plus.png"/>

</Button.Background>

</Button>

<Button x:Name="Delete" Click="Delete\_Click" Grid.Column="1">

<Button.Background>

<ImageBrush Stretch="Uniform" ImageSource="/rubbish.png"/>

</Button.Background>

</Button>

<Button x:Name="Edit" Click="Edit\_Click" Grid.Column="2">

<Button.Background>

<ImageBrush ImageSource="/editing.png" Stretch="Uniform"/>

</Button.Background>

</Button>

</Grid>

<Label Content="Запросы" HorizontalAlignment="Center" Width="60" Grid.Row="1" Margin="0,5,0,5"/>

<ListBox SelectionChanged="TablesDefault\_SelectionChanged" x:Name="TablesSpecials" Margin="0,5,5,10" RenderTransformOrigin="1,1" Grid.Row="3" SelectedIndex="0">

<ListBox.RenderTransform>

<TransformGroup>

<ScaleTransform ScaleY="1"/>

<SkewTransform/>

<RotateTransform/>

<TranslateTransform/>

</TransformGroup>

</ListBox.RenderTransform>

</ListBox>

</Grid></Window>

Файл «MainWindow.xaml.cs».

using System.Data;

using System.Data.OleDb;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

namespace Stoma;

public partial class MainWindow : Window

{

public MainWindow() => InitializeComponent();

private const string \_connectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=StomaDB.mdb;";

private DataTable? \_currentDataTable;

private string? \_selectedTableName;

private const string CREATE = "Создать";

private const string DELETE = "Удалить";

private const string CHANGE = "Изменить";

private readonly Dictionary<(string Table, string Action), string> \_spesialQueries = new()

{

{ ("Личности", CREATE), "INSERT INTO [Личности] " +

"([Фамилия], [Имя], [Отчество], [Паспорт], [СНИЛС], [ИНН], [Номер телефона], [Электронная почта]) " +

"VALUES (?, ?, ?, ?, ?, ?, ?, ?)" },

{ ("Личности", DELETE), "DELETE FROM [Личности] WHERE [Id личности] = ?" },

{ ("Личности", CHANGE), "UPDATE [Личности] " +

"SET [Фамилия] = ?, [Имя] = ?, [Отчество] = ?, [Паспорт] = ?, [СНИЛС] = ?, [ИНН] = ?, [Номер телефона] = ?, [Электронная почта] = ? " +

"WHERE [Id личности] = ?" },

{ ("Должности", CREATE), "INSERT INTO [Должности] ([Название]) VALUES (?)" },

{ ("Должности", DELETE), "DELETE FROM [Должности] WHERE [Id должности] = ?" },

{ ("Должности", CHANGE), "UPDATE [Должности] SET [Название] = ? WHERE [Id должности] = ?" },

{ ("Кабинеты", CREATE), "INSERT INTO [Кабинеты] ([Номер кабинета], [Название]) VALUES (?, ?)" },

{ ("Кабинеты", DELETE), "DELETE FROM [Кабинеты] WHERE [Номер кабинета] = ?" },

{ ("Кабинеты", CHANGE), "UPDATE [Кабинеты] SET [Номер кабинета] = ?, [Название] = ? WHERE [Номер кабинета] = ?" },

{ ("Прейскурант", CREATE), "INSERT INTO [Прейскурант] ([Название], [Стоимость], [Единица измерения]) VALUES (?, ?, ?)" },

{ ("Прейскурант", DELETE), "DELETE FROM [Прейскурант] WHERE [Id услуги] = ?" },

{ ("Приёмы", CREATE), "INSERT INTO [Приёмы] ([Id сотрудника], [Id пациента], [Дата и время], [Заключение], [Номер кабинета]) VALUES (?, ?, ?, ?, ?)" },

{ ("Приёмы", DELETE), "DELETE FROM [Приёмы] WHERE [Id приёма] = ?" },

{ ("Приёмы", CHANGE), "UPDATE [Приёмы] " +

"SET [Id сотрудника] = ?, [Id пациента] = ?, [Дата и время] = ?, [Заключение] = ? " +

"WHERE [Id приёма] = ?" },

{ ("Сотрудники", CREATE), "INSERT INTO [Сотрудники] " +

"([Id личности], [Id должности], [Id формы оклада], [Оклад], [Дата заключения], [Договор]) " +

"VALUES (?, ?, ?, ?, ?, ?)" },

{ ("Сотрудники", DELETE), "DELETE FROM [Сотрудники] WHERE [Id сотрудника] = ?" },

{ ("Сотрудники", CHANGE), "UPDATE [Сотрудники] " +

"SET [Id личности] = ?, [Id должности] = ?, [Id формы оклада] = ?, [Оклад] = ?, [Дата заключения] = ?, [Договор] = ? " +

"WHERE [Id сотрудника] = ?" },

{ ("Пациенты", CREATE), "INSERT INTO [Пациенты] " +

"([Id личности], [Дата заключения], [Дата окончания], [Противопоказания], " +

"[Согласие на обработку персональных данных], [Согласие на медицинское вмешательство]) " +

"VALUES (?, ?, ?, ?, ?, ?)" },

{ ("Пациенты", DELETE), "DELETE FROM [Пациенты] WHERE [Id пациента] = ?" },

{ ("Поставки", CREATE), "INSERT INTO [Поставки] ([Название поставщика], [Дата поставки], [Номер телефона]) VALUES (?, ?, ?)" },

{ ("Поставки", DELETE), "DELETE FROM [Поставки] WHERE [Id поставки] = ?" },

{ ("Медицинские карты", CREATE), "INSERT INTO [Медицинские карты] ([Id приёма], [Id услуги]) VALUES (?, ?)" },

};

// Инициализация списка всех доступных таблиц

private void InitTableDefault(object? sender, RoutedEventArgs? e)

{

TablesDefault.Items.Clear();

try

{

using var connection = new OleDbConnection(\_connectionString);

connection.Open();

var tables = connection.GetOleDbSchemaTable(OleDbSchemaGuid.Tables, new string?[] { null, null, null, "TABLE" })!;

if (tables is null)

{

MessageBox.Show("Пустое соединение", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

foreach (var tableName in from DataRow? row in tables.Rows

let tableName = row?["TABLE\_NAME"].ToString()

where !tableName.StartsWith("MSys")

select tableName)

{

TablesDefault.Items.Add(tableName);

}

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка подключения к базе данных:\n\n{ex.Message}", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private readonly Dictionary<string, string> \_defaultQueries = new()

{

["Личности"] = "SELECT \* FROM [Личности]",

["Должности"] = "SELECT \* FROM [Должности]",

["Прейскурант"] = "SELECT \* FROM [Прейскурант]",

["Приёмы"] = "SELECT \* FROM [Приёмы]",

["Стоматологические материалы"] = @$"SELECT \* FROM [Стоматологические материалы]",

["Инструменты"] = "SELECT \* FROM [Инструменты]",

["Оборудование"] = "SELECT \* FROM [Оборудование]",

["Хозяйственные принадлежности"] = "SELECT \* FROM [Хозяйственные принадлежности]",

["Сотрудники"] = "SELECT \* FROM [Сотрудники]",

["Медицинские карты"] = "SELECT \* FROM [Медицинские карты]",

["Кабинеты"] = "SELECT \* FROM [Кабинеты]",

["Пациенты"] = "SELECT \* FROM [Пациенты]",

["Статусы ресурсов"] = "SELECT \* FROM [Статусы ресурсов]",

["Формы оклада"] = "SELECT \* FROM [Формы оклада]",

["Поставки"] = "SELECT \* FROM [Поставки]",

["Планы лечения"] = "SELECT \* FROM [Планы лечения]",

};

// Реакция на выбор запроса таблицы

private void TablesDefault\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (TablesDefault.SelectedItem is not string selectedItem) return;

try

{

if (!\_defaultQueries.TryGetValue(\_selectedTableName = selectedItem, out var query))

{

MessageBox.Show("Запрос не реализован...", "Предупреждение", MessageBoxButton.OK, MessageBoxImage.Warning);

return;

}

LoadTableData(query);

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка при загрузке:\n\n{ex.Message}", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void LoadTableData(string query)

{

try

{

using var connection = new OleDbConnection(\_connectionString);

connection.Open();

var adapter = new OleDbDataAdapter(query, connection);

adapter.Fill(\_currentDataTable = new());

Tables.ItemsSource = null;

Tables.Columns.Clear();

var foreignTables = LoadDataset(connection);

foreach (DataColumn column in \_currentDataTable.Columns)

{

string columnName = column.ColumnName;

if (foreignTables.ContainsKey((\_selectedTableName!, columnName)))

{

var (source, displayMember) = foreignTables[(\_selectedTableName!, columnName)];

var comboBoxColumn = new DataGridComboBoxColumn

{

Header = new TextBlock { Text = displayMember },

SelectedValuePath = columnName,

DisplayMemberPath = displayMember,

SelectedValueBinding = new Binding($"[{columnName}]"),

ItemsSource = source.DefaultView

};

Tables.Columns.Add(comboBoxColumn);

}

else if (column.DataType == typeof(DateTime))

{

var binding = new Binding($"[{columnName}]");

if (columnName.ToLower().Contains("дата")) binding.StringFormat = "dd.MM.yyyy";

else binding.StringFormat = "HH:mm";

var dateColumn = new DataGridTextColumn

{

Header = new TextBlock { Text = columnName },

Binding = binding

};

Tables.Columns.Add(dateColumn);

}

else

{

var textColumn = new DataGridTextColumn

{

Header = new TextBlock { Text = columnName },

Binding = new Binding($"[{columnName}]")

};

Tables.Columns.Add(textColumn);

}

}

Tables.ItemsSource = \_currentDataTable.DefaultView;

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка при загрузке таблицы:\n\n{ex.Message}", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private Dictionary<(string Table, string Column), (DataTable Source, string DisplayMember)>

LoadDataset(OleDbConnection connection)

{

var dict = new Dictionary<(string, string), (DataTable, string)>();

Add("Приёмы", "Id сотрудника",

"SELECT [Id сотрудника], [Личности].[Фамилия] & ' ' & [Личности].[Имя] & ' ' & [Личности].[Отчество] & ': ' & [Должности].[Название] AS [Сотрудник] " +

"FROM ([Сотрудники] " +

"INNER JOIN [Личности] ON [Сотрудники].[Id личности] = [Личности].[Id личности]) " +

"INNER JOIN [Должности] ON [Сотрудники].[Id должности] = [Должности].[Id должности]", "Сотрудник");

Add("Приёмы", "Id пациента",

"SELECT [Id пациента], [Личности].[Фамилия] & ' ' & [Личности].[Имя] & ' ' & [Личности].[Отчество] " +

"AS [Пациент] FROM [Пациенты] " +

"INNER JOIN [Личности] ON [Пациенты].[Id личности] = [Личности].[Id личности]", "Пациент");

Add("Приёмы", "Номер кабинета",

"SELECT [Номер кабинета], [Номер кабинета] & '\t' & [Название] AS [Кабинет] FROM [Кабинеты]", "Кабинет");

Add("Сотрудники", "Id личности",

"SELECT [Id личности], [Фамилия] & ' ' & [Имя] & ' ' & [Отчество] AS [ФИО] " +

"FROM [Личности]", "ФИО");

Add("Пациенты", "Id личности",

"SELECT [Id личности], [Фамилия] & ' ' & [Имя] & ' ' & [Отчество] AS [ФИО] " +

"FROM [Личности]", "ФИО");

Add("Сотрудники", "Id должности",

"SELECT [Id должности], [Название] AS [Должность] " +

"FROM [Должности]", "Должность");

Add("Сотрудники", "Id формы оклада",

"SELECT [Id формы оклада], [Название] AS [Тип оклада] " +

"FROM [Формы оклада]", "Тип оклада");

Add("Медицинские карты", "Id приёма",

"SELECT [Id приёма], [Личности].[Фамилия] & ' ' & [Личности].[Имя] & ' ' & [Личности].[Отчество] & '\t' & [Приёмы].[Дата и время] & '\t' & [Приёмы].[Номер кабинета] AS [Приём] " +

"FROM ([Приёмы] " +

"INNER JOIN [Пациенты] ON [Приёмы].[Id пациента] = [Пациенты].[Id пациента]) " +

"INNER JOIN [Личности] ON [Пациенты].[Id личности] = [Личности].[Id личности]", "Приём");

Add("Медицинские карты", "Id услуги",

"SELECT [Id услуги], [Название] & ' - ' & [Стоимость] AS [Услуга]" +

"FROM [Прейскурант]", "Услуга");

return dict;

void Add(string table, string column, string query, string display)

{

var tableData = new DataTable();

var adapter = new OleDbDataAdapter(query, connection);

adapter.Fill(tableData);

if (tableData.Rows.Count <= 0)

{

MessageBox.Show($"!!! Нет данных для {table}.{column} !!!", "Предупреждение", MessageBoxButton.OK, MessageBoxImage.Warning);

}

dict[(table, column)] = (tableData, display);

}

}

private readonly Dictionary<string, int> \_countOfPrKeys = new()

{

["Личности"] = 1,

["Должности"] = 1,

["Прейскурант"] = 1,

["Приёмы"] = 1,

["Стоматологические материалы"] = 1,

["Сотрудники"] = 1,

["Медицинские карты"] = 0,

["Кабинеты"] = 0,

["Пациенты"] = 1,

["Статусы ресурсов"] = 1,

["Формы оклада"] = 1,

["Поставки"] = 1,

["Планы лечения"] = 1,

};

private void Add\_Click(object sender, RoutedEventArgs e)

{

if (string.IsNullOrEmpty(\_selectedTableName) || \_currentDataTable is null)

{

MessageBox.Show("Сначала выберите таблицу!", "Предупреждение", MessageBoxButton.OK, MessageBoxImage.Warning);

return;

}

try

{

var changes = \_currentDataTable.GetChanges(DataRowState.Added);

if (changes is null || changes.Rows.Count <= 0)

{

MessageBox.Show($"Не введены строки чтобы добавить...", "Предупреждение", MessageBoxButton.OK, MessageBoxImage.Warning);

return;

}

using var connection = new OleDbConnection(\_connectionString);

connection.Open();

foreach (DataRow row in changes.Rows)

{

if (!\_spesialQueries.TryGetValue((\_selectedTableName, CREATE), out var query))

{

MessageBox.Show("Запрос не реализован...", "Предупреждение", MessageBoxButton.OK, MessageBoxImage.Warning);

return;

}

var command = new OleDbCommand(query, connection);

for (int i = \_countOfPrKeys[\_selectedTableName]; i < changes.Columns.Count; ++i)

{

command.Parameters.AddWithValue("?", row[i] ?? "");

}

command.ExecuteNonQuery();

}

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка при добавлении:\n\n{ex.Message}", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

LoadTableData($"SELECT \* FROM [{\_selectedTableName}]");

}

private void Edit\_Click(object sender, RoutedEventArgs e)

{

if (string.IsNullOrEmpty(\_selectedTableName) || \_currentDataTable is null)

{

MessageBox.Show("Сначала выберите таблицу!", "Предупреждение", MessageBoxButton.OK, MessageBoxImage.Warning);

return;

}

try

{

var changes = \_currentDataTable.GetChanges(DataRowState.Modified);

if (changes is null || changes.Rows.Count <= 0)

{

MessageBox.Show("Не выбраны строки чтобы изменить...", "Предупреждение", MessageBoxButton.OK, MessageBoxImage.Warning);

return;

}

using var connection = new OleDbConnection(\_connectionString);

connection.Open();

foreach (DataRow row in changes.Rows)

{

if (!\_spesialQueries.TryGetValue((\_selectedTableName, CHANGE), out var query))

{

MessageBox.Show("Запрос не реализован...", "Предупреждение", MessageBoxButton.OK, MessageBoxImage.Warning);

return;

}

var command = new OleDbCommand(query, connection);

for (int i = \_countOfPrKeys[\_selectedTableName]; i < changes.Columns.Count; ++i)

{

command.Parameters.AddWithValue("?", row[i] ?? "");

}

command.Parameters.AddWithValue("?", row[0, DataRowVersion.Original] ?? "");

command.ExecuteNonQuery();

}

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка при изменении:\n\n{ex.Message}", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

LoadTableData($"SELECT \* FROM [{\_selectedTableName}]");

}

private void Delete\_Click(object sender, RoutedEventArgs e)

{

if (string.IsNullOrEmpty(\_selectedTableName) || \_currentDataTable is null)

{

MessageBox.Show("Сначала выберите таблицу!", "Предупреждение", MessageBoxButton.OK, MessageBoxImage.Warning);

return;

}

try

{

var changes = \_currentDataTable.GetChanges(DataRowState.Deleted);

if (changes is null || changes.Rows.Count <= 0)

{

MessageBox.Show("Не выбраны строки чтобы удалить...", "Предупреждение", MessageBoxButton.OK, MessageBoxImage.Warning);

return;

}

using var connection = new OleDbConnection(\_connectionString);

connection.Open();

foreach (DataRow row in changes.Rows)

{

if (!\_spesialQueries.TryGetValue((\_selectedTableName, DELETE), out var query))

{

MessageBox.Show("Запрос не реализован...", "Предупреждение", MessageBoxButton.OK, MessageBoxImage.Warning);

return;

}

var command = new OleDbCommand(query, connection);

command.Parameters.AddWithValue("?", row[0, DataRowVersion.Original]);

command.ExecuteNonQuery();

}

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка при удалении:\n\n{ex.Message}", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

LoadTableData($"SELECT \* FROM [{\_selectedTableName}]");

}

}

# Приложение 3

Руководство пользователя

Общие сведения о программе

Программа осуществляет взаимодействие с базой данных «Microsoft Access» «Стоматологическая клиника». Реализованы операции создания, чтения, изменения и удаления записей в базе данных.

Установка программы

Программа поставляется в архиве «StomaApp.zip». Необходимо распаковать в желаемую папку (рис. П3.1). В комплекте идут исполняемый файл приложения и файл базы данных. Для работы приложения требуется установленным пакет «Microsoft Access».

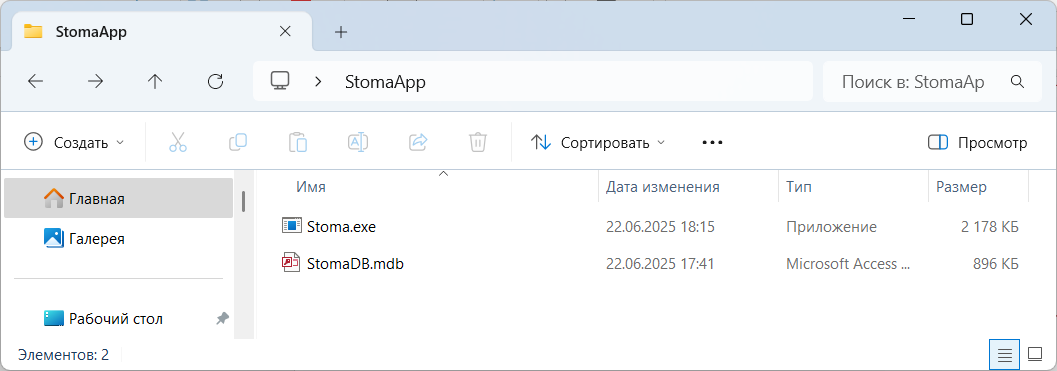


Рис. П3.1. Содержимое распакованного архива

Запуск программы

Запуск программы осуществляется запуском исполняемого файла «Stoma.exe», после чего откроется единственное рабочее окно, в котором осуществляются все доступные операции над базой данных (рис. П3.2).

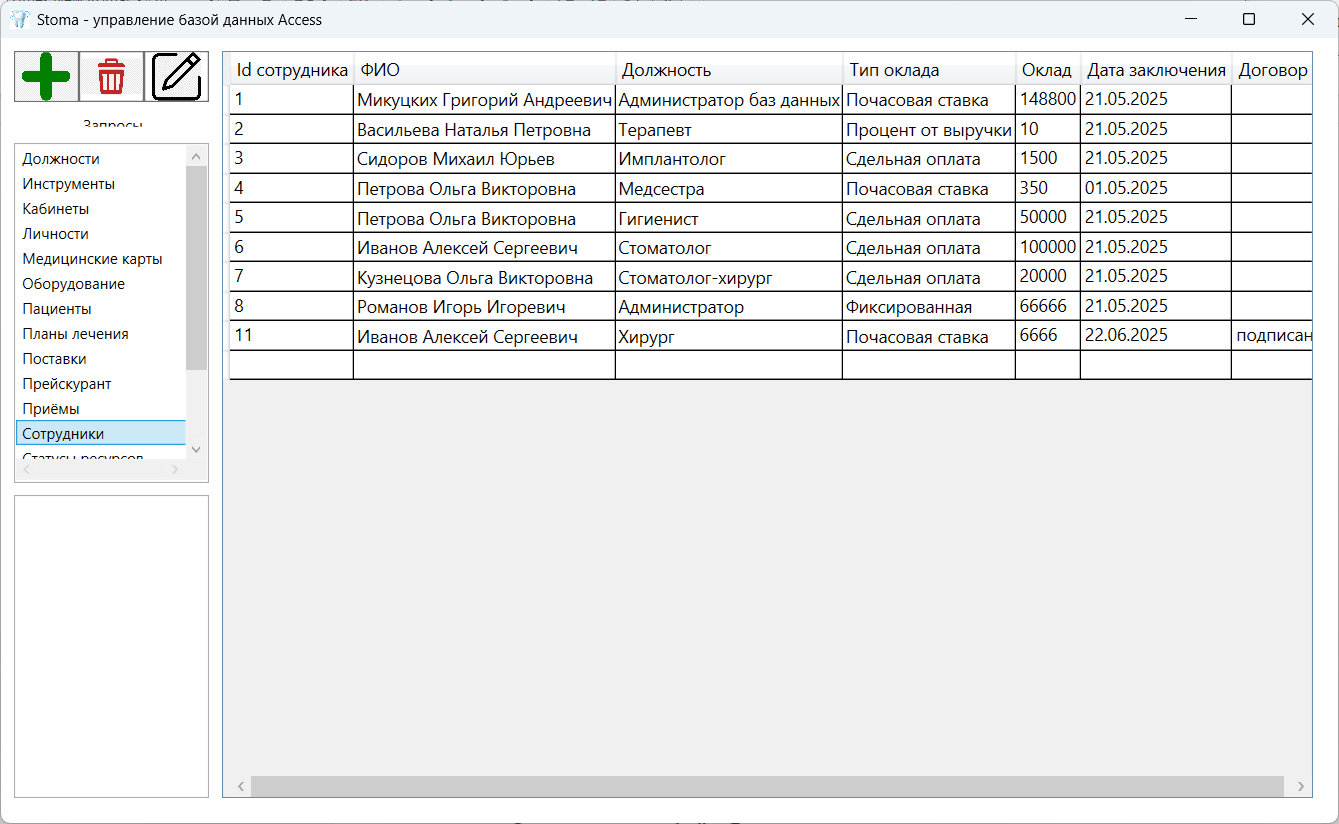


Рис. П3.2. Главное рабочее окно

Описание интерфейса

В левой части окна приложения в один столбец расположены все запросы, доступные для взаимодействия с базой данных. В рабочей области полученные ответы отображаются в виде таблице.

В левом верхнем углу программы расположены кнопки фиксации изменений (рис. П3.3), которые произведены над таблицей в программе:

* добавлены данные (знак плюс «➕»);
* удалены данные (знак корзины «🗑️»);
* изменены данные (знак ручки со стикером «📝»).

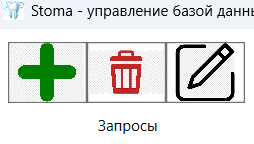


Рис. П3.3. Кнопки фиксации изменений

Приёмы работы в программе

Нажатия на кнопки в списке доступных запросов осуществляют отправку запросов на выборку к базе данных (см. рис. П3.2). Обновление содержимого базы данных не повлияет на отображаемые данные, поэтому необходимо повторно отправить запрос, чтобы обновления отобразились.

Взаимодействие с данными осуществляется следующим образом:

* для ввода новых данных необходимо ввести запись в конец текущей таблицы, затем зафиксировать добавление (рис. П3.4-П3.5);

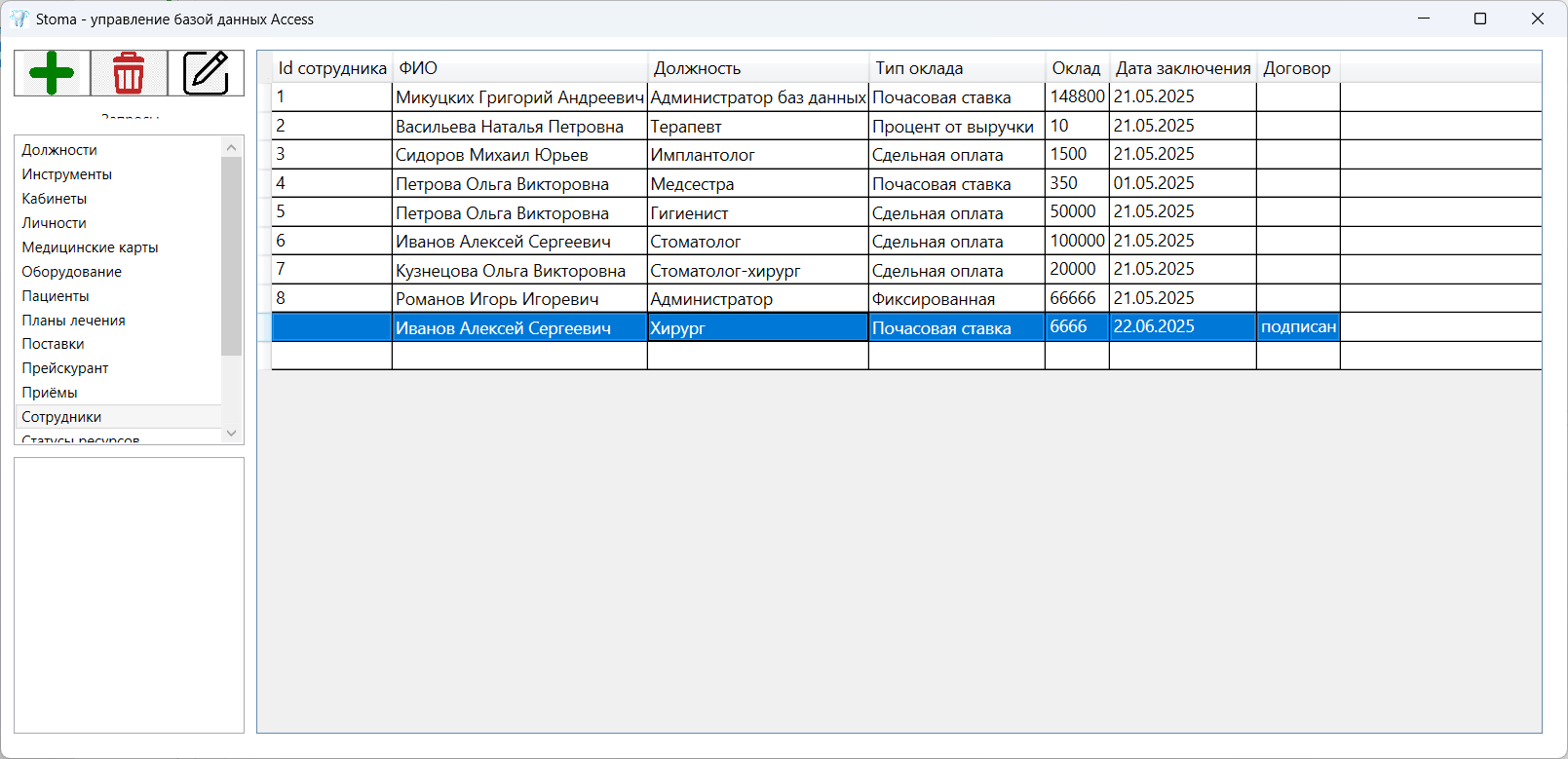


Рис. П3.4. Новая строка создана

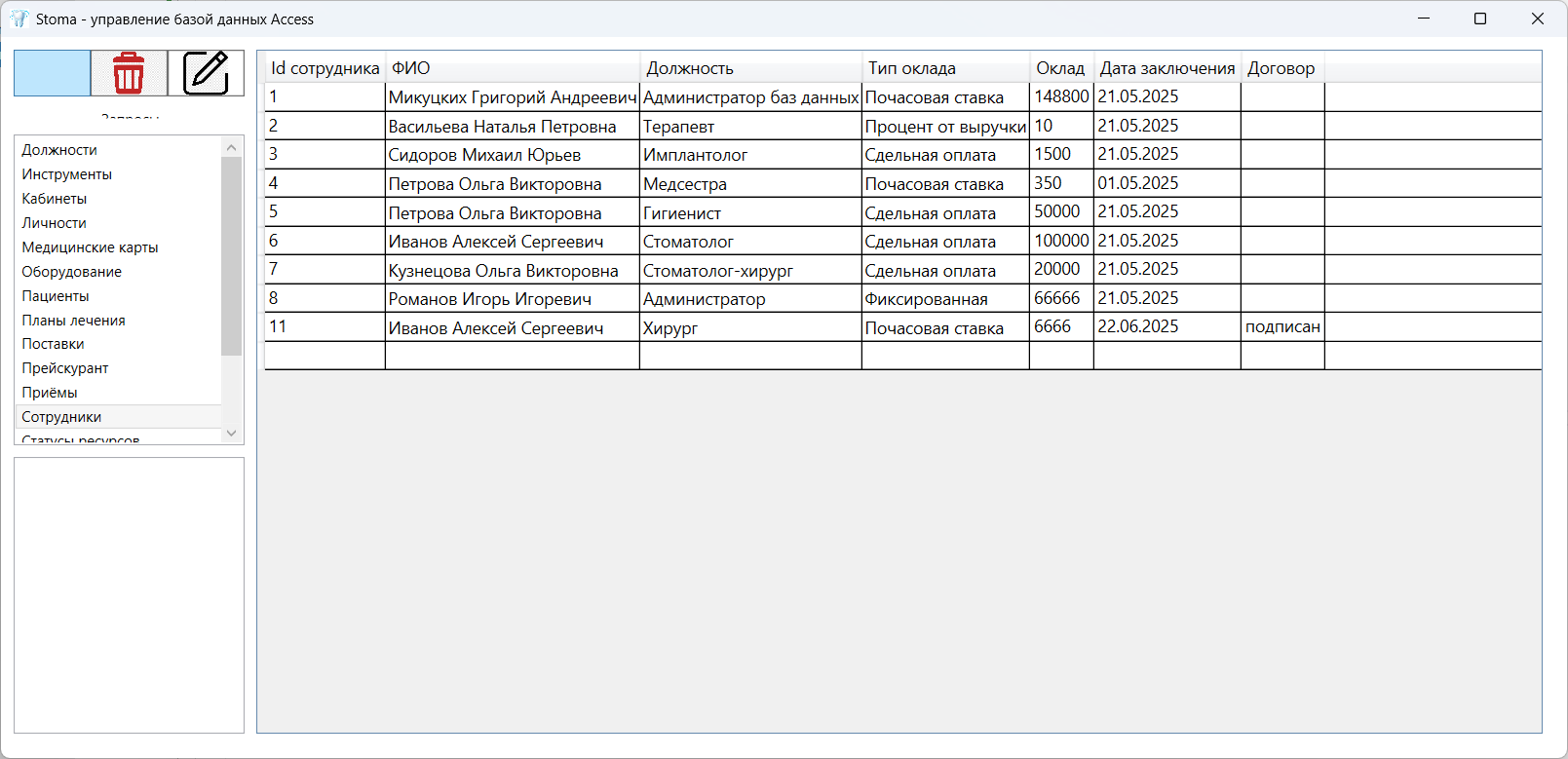


Рис. П3.5. Новая запись введена в базу данных

* для изменения записи необходимо изменить какие-либо данные (только в текущей области), затем зафиксировать изменения (рис. П3.6-П3.7);

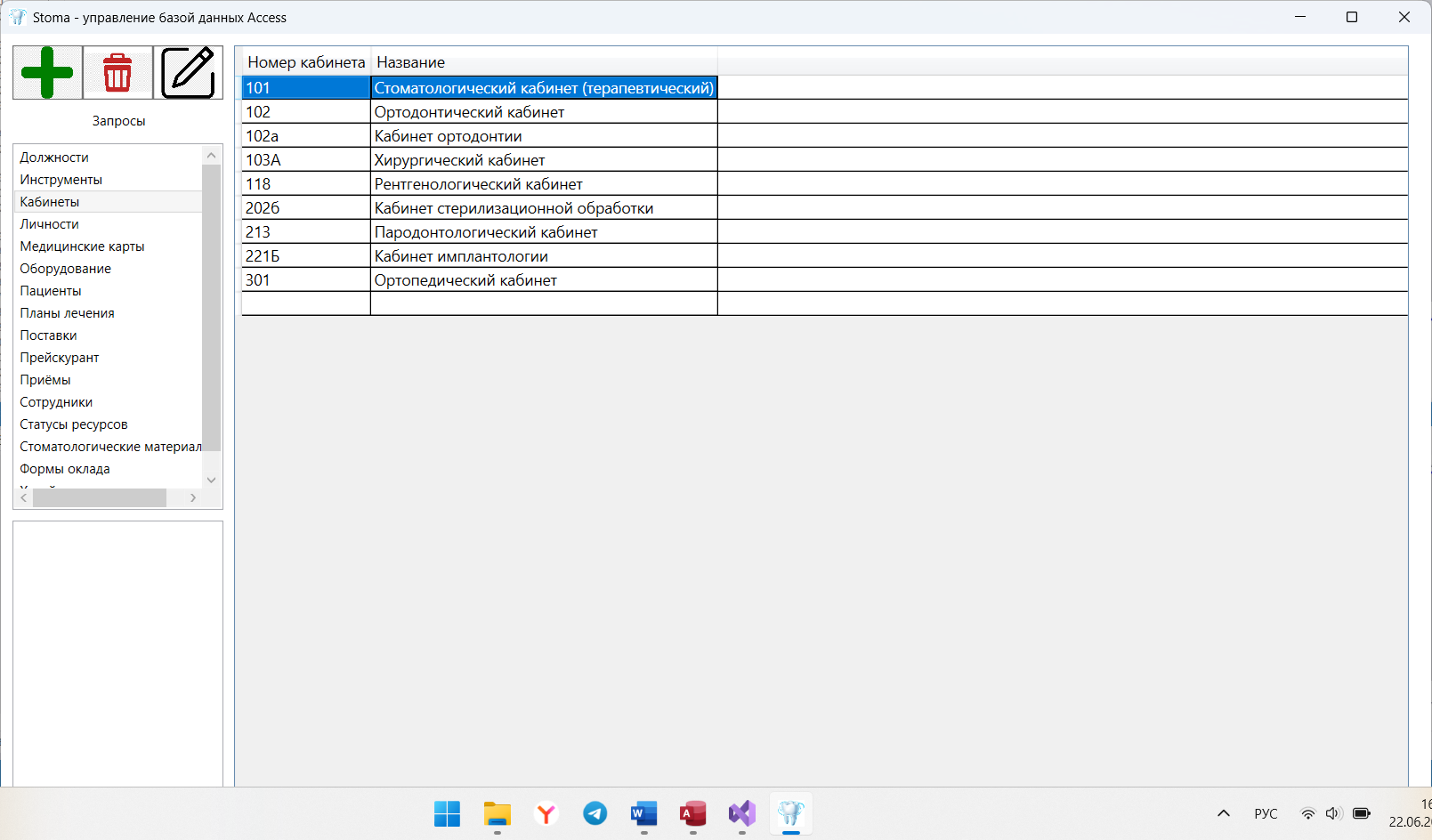


Рис. П3.6. Выбрана запись для изменения

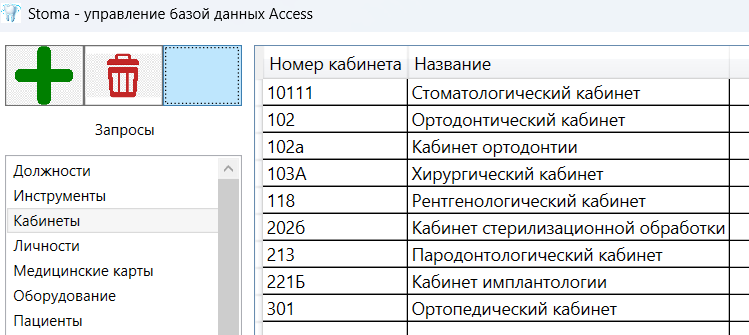


Рис. П3.7. Запись изменена

* для удаления необходимо выбрать строку, нажать клавишу «Del» на клавиатуре, затем зафиксировать удаление (рис. П3.8-П3.9).

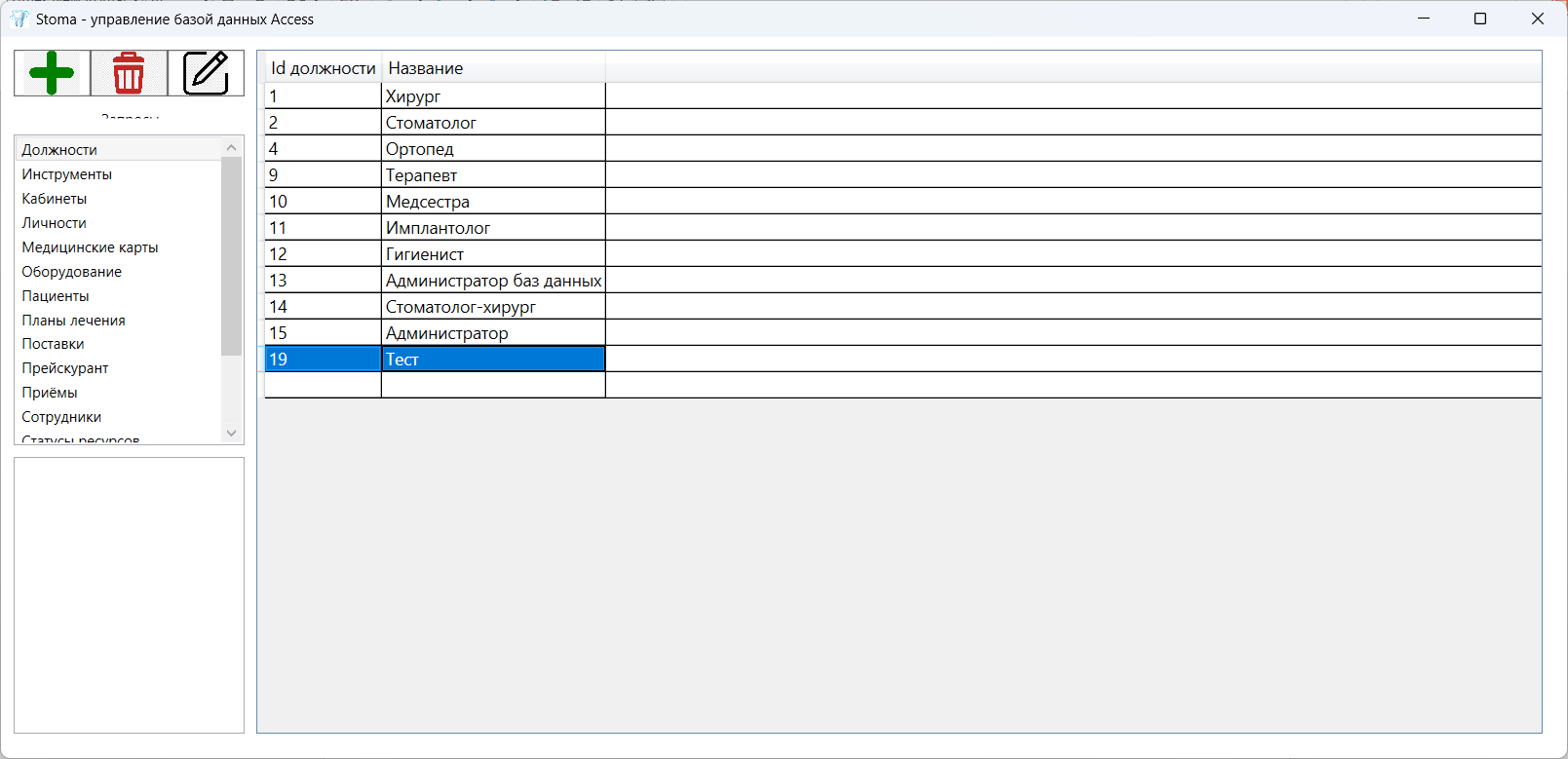


Рис. П3.8. Пример записи для удаления

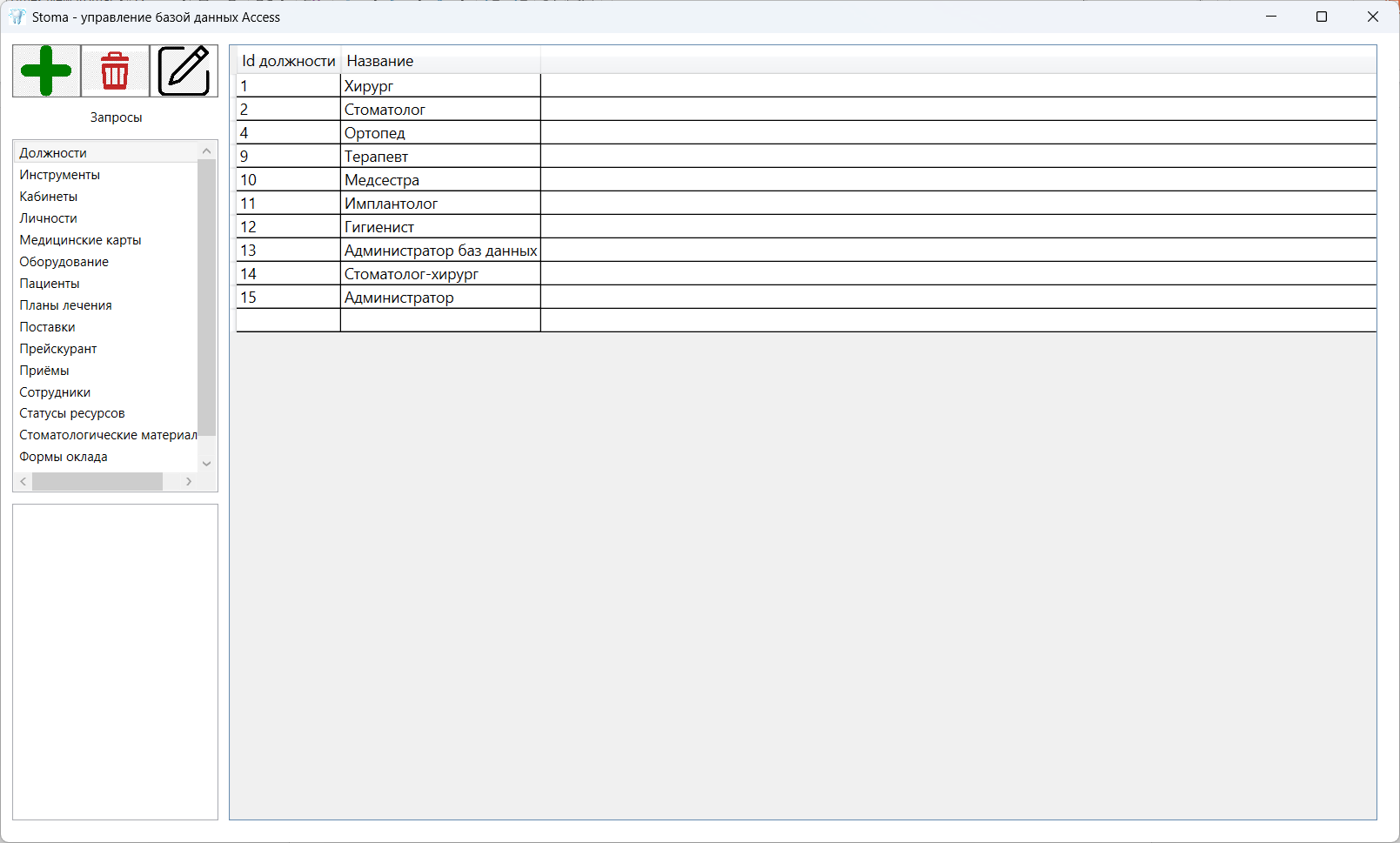


Рис. П3.9. Запись удалена из базы данных

Для примера работы с приложением, занесём в базу данных результаты приёма пациента. Сначала нужно перейти в раздел «Приёмы» (рис. П3.10).

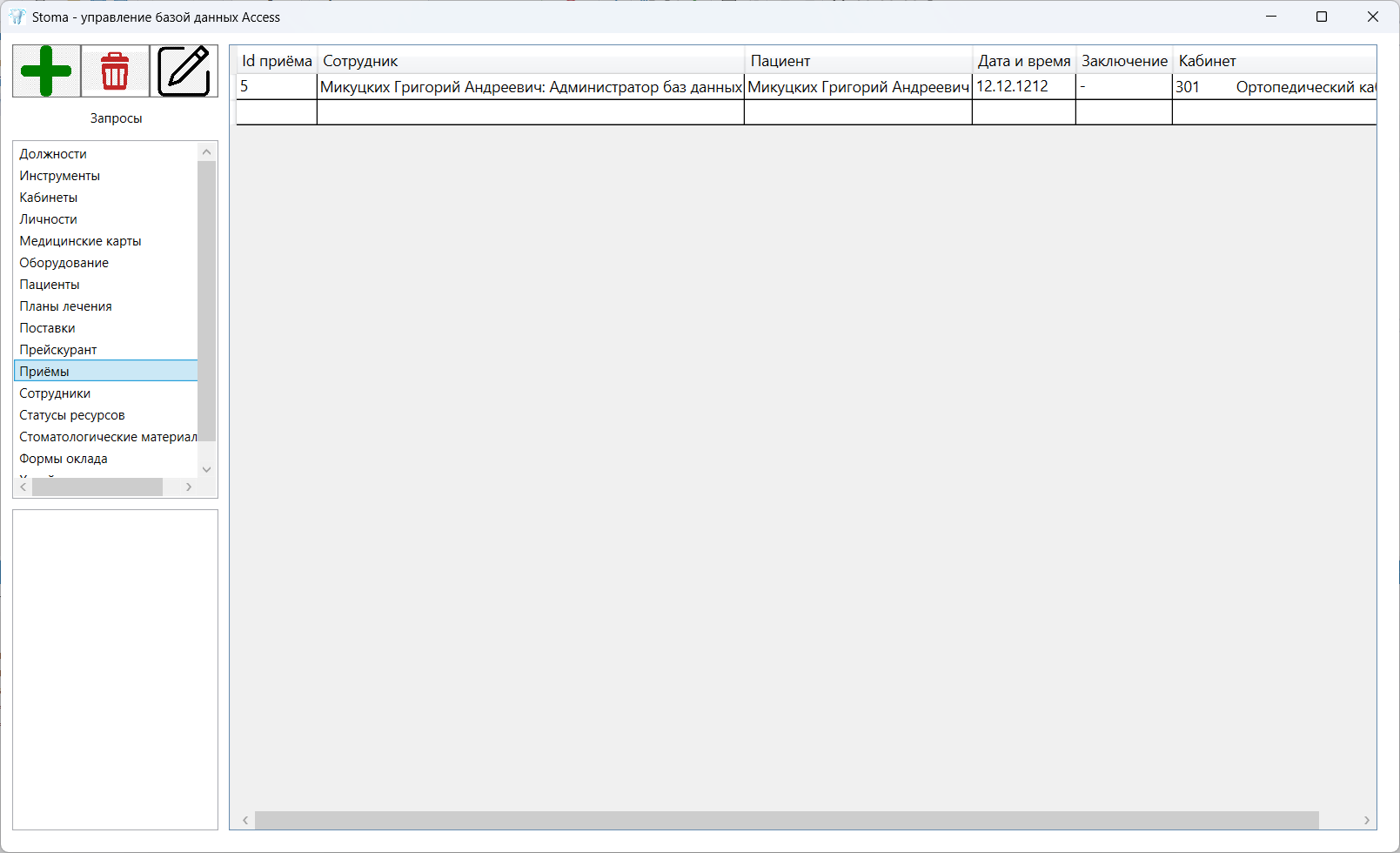


Рис. П3.10. Раздел «Приёмы»

Нужно ввести в новую (пустую) строку данные о приёме. Для удобства, выбор перечисляемых параметров (сотрудников, кабинетов, пациентов и прочее) осуществляется через выпадающие списки (рис. П3.11).

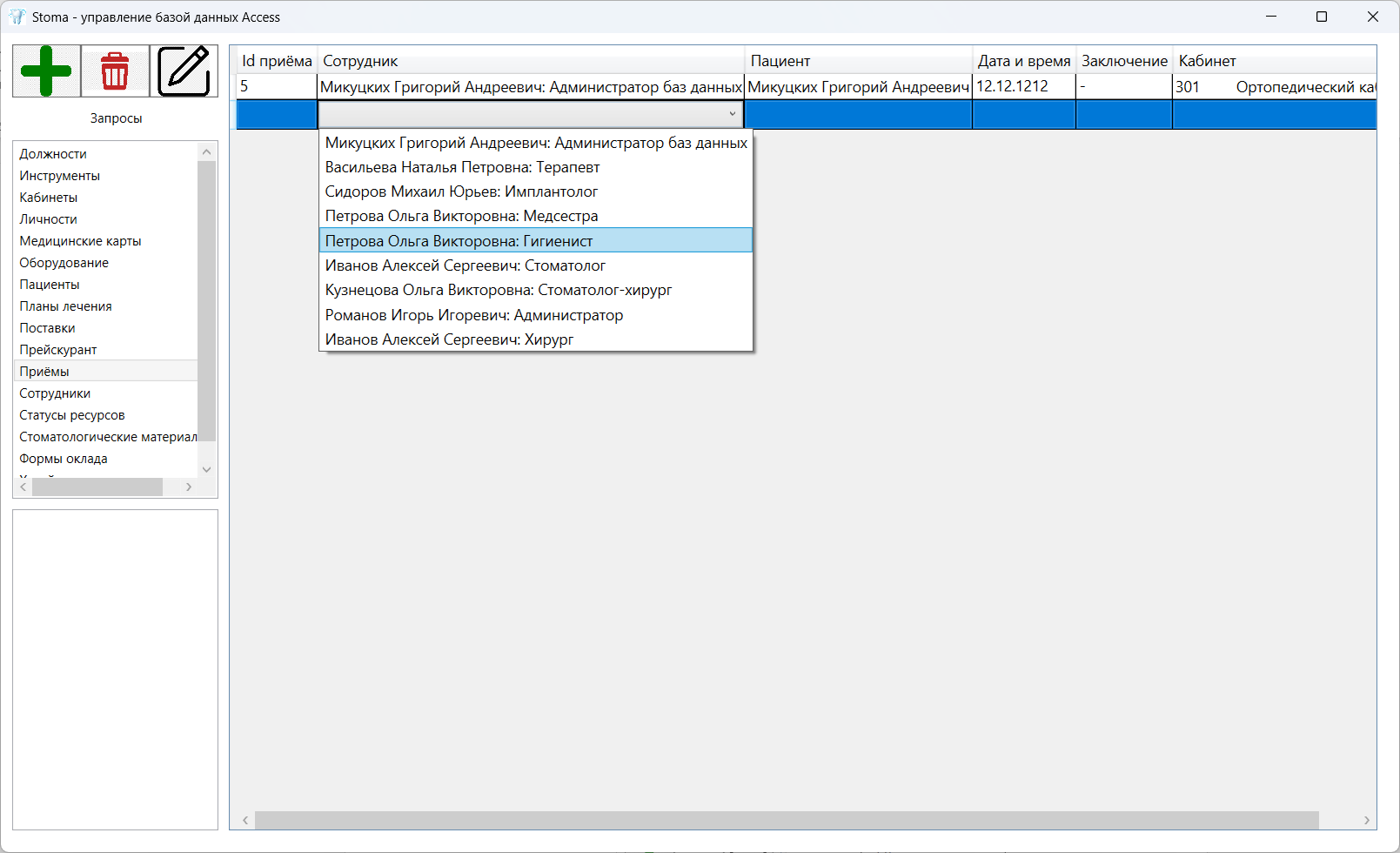


Рис. П3.11. Выпадающий список выбора сотрудника

Для примера были внесены следующие данные (рис. П3.12).

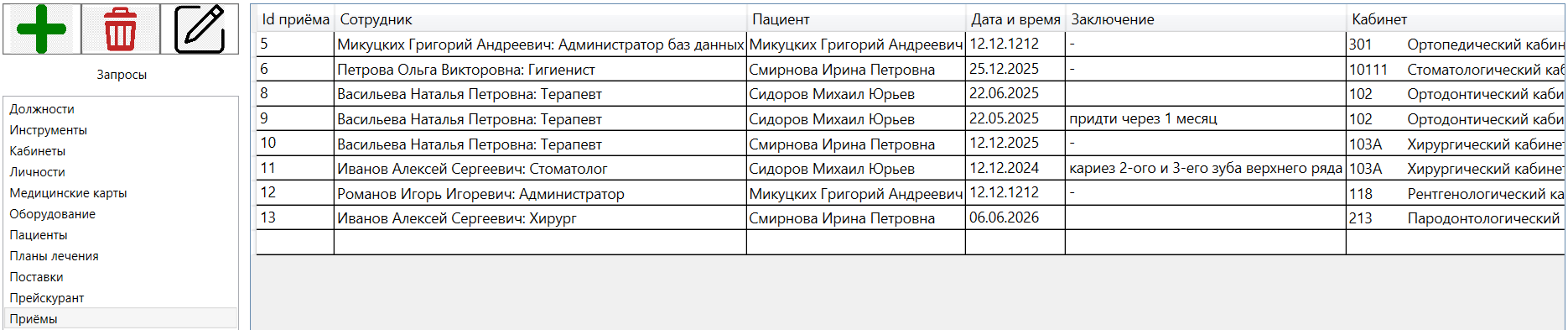


Рис. П3.12. Новые записи успешно добавлены

Теперь можно ввести в таблицу «Медицинские карты» какие услуги были оказаны в ходе приёма (рис. П3.13-П3.14).

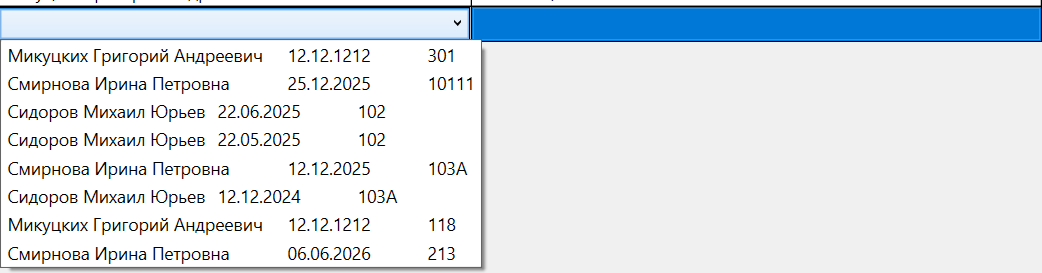


Рис. П3.13. Выпадающий список приёмов

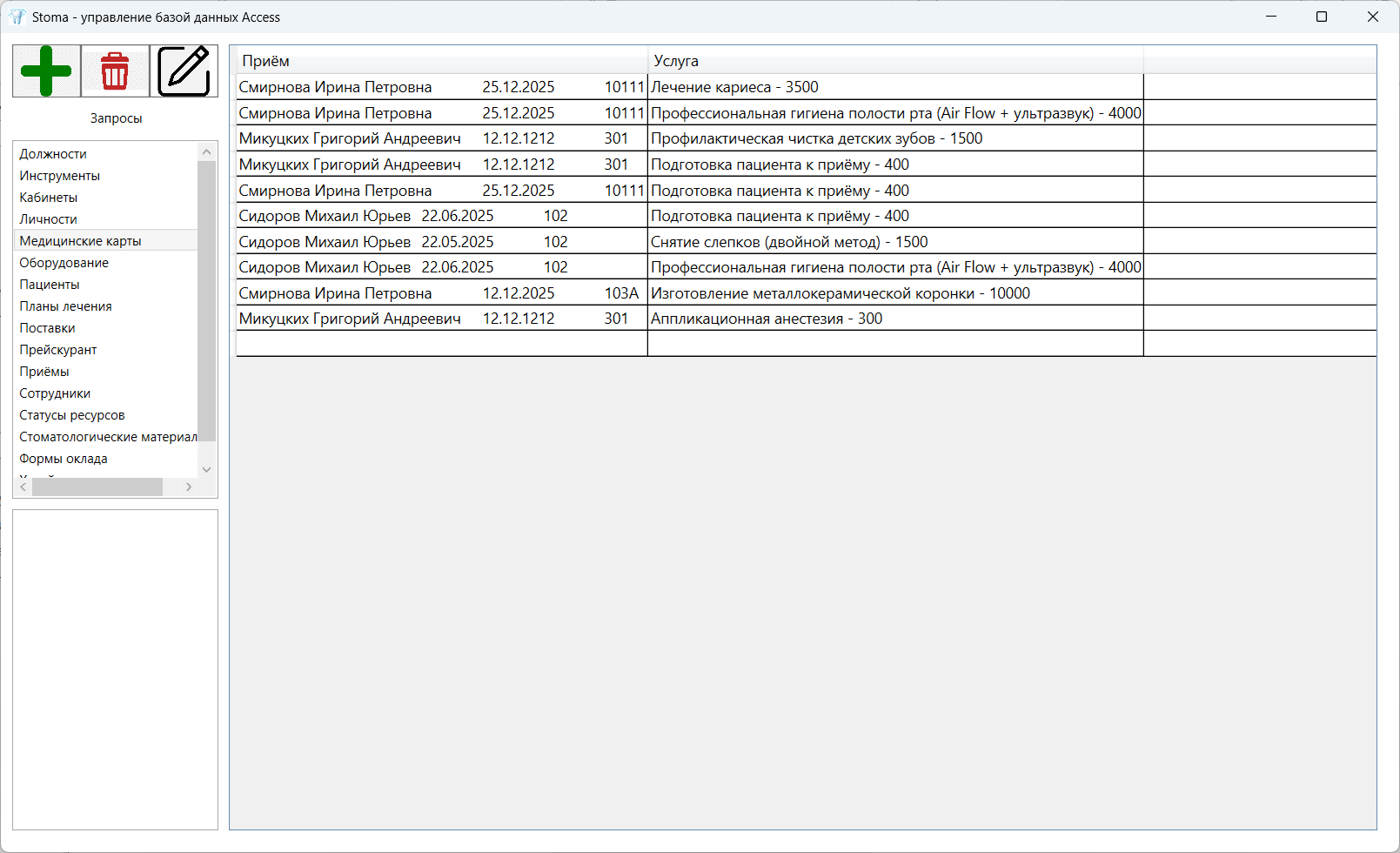


Рис. П3.14. Пример заполненной медицинской карты

Дополнительные примеры результатов вывода данных из базы данных представлены на рис. П3.15.-П3.17.

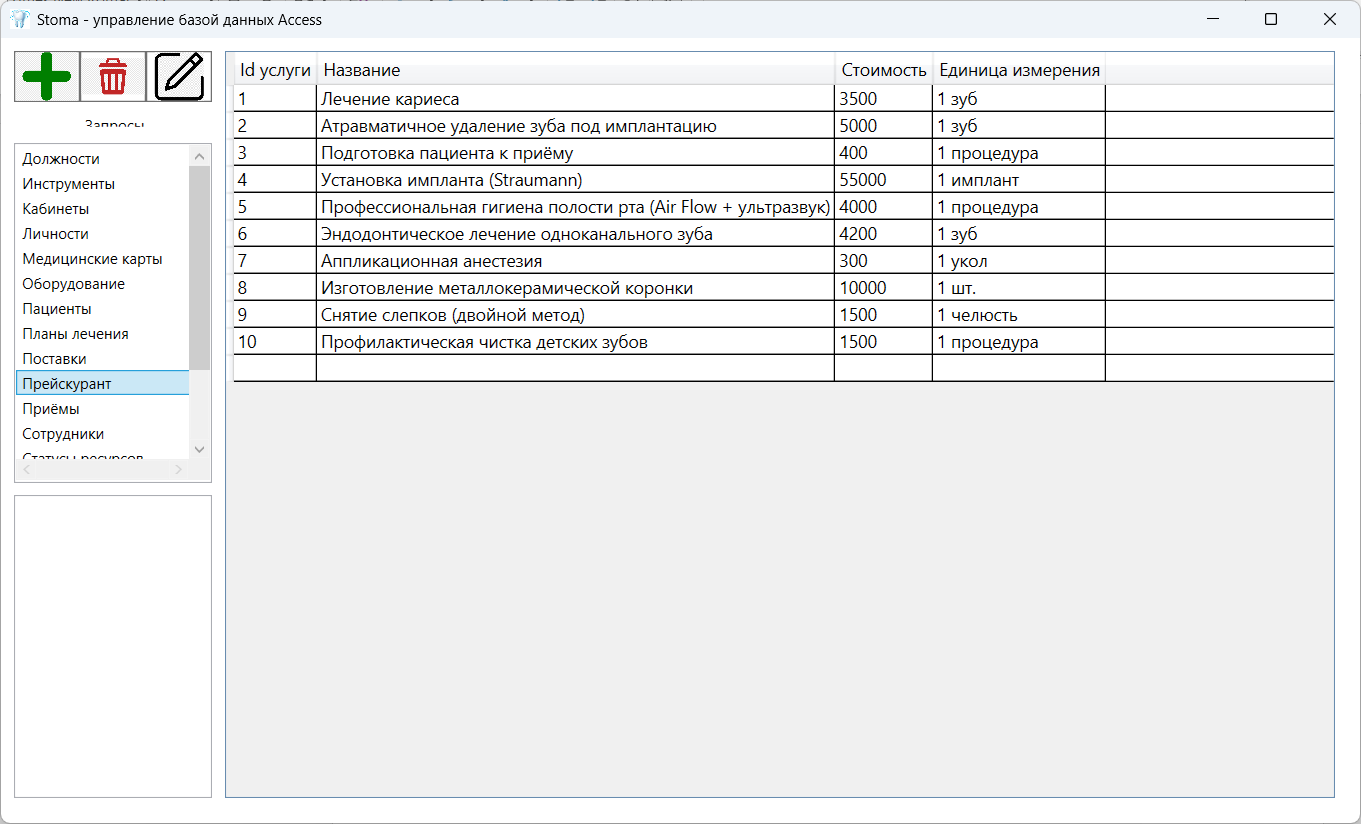


Рис. П3.15. Пример вывода всех записей из таблицы «Прейскурант»

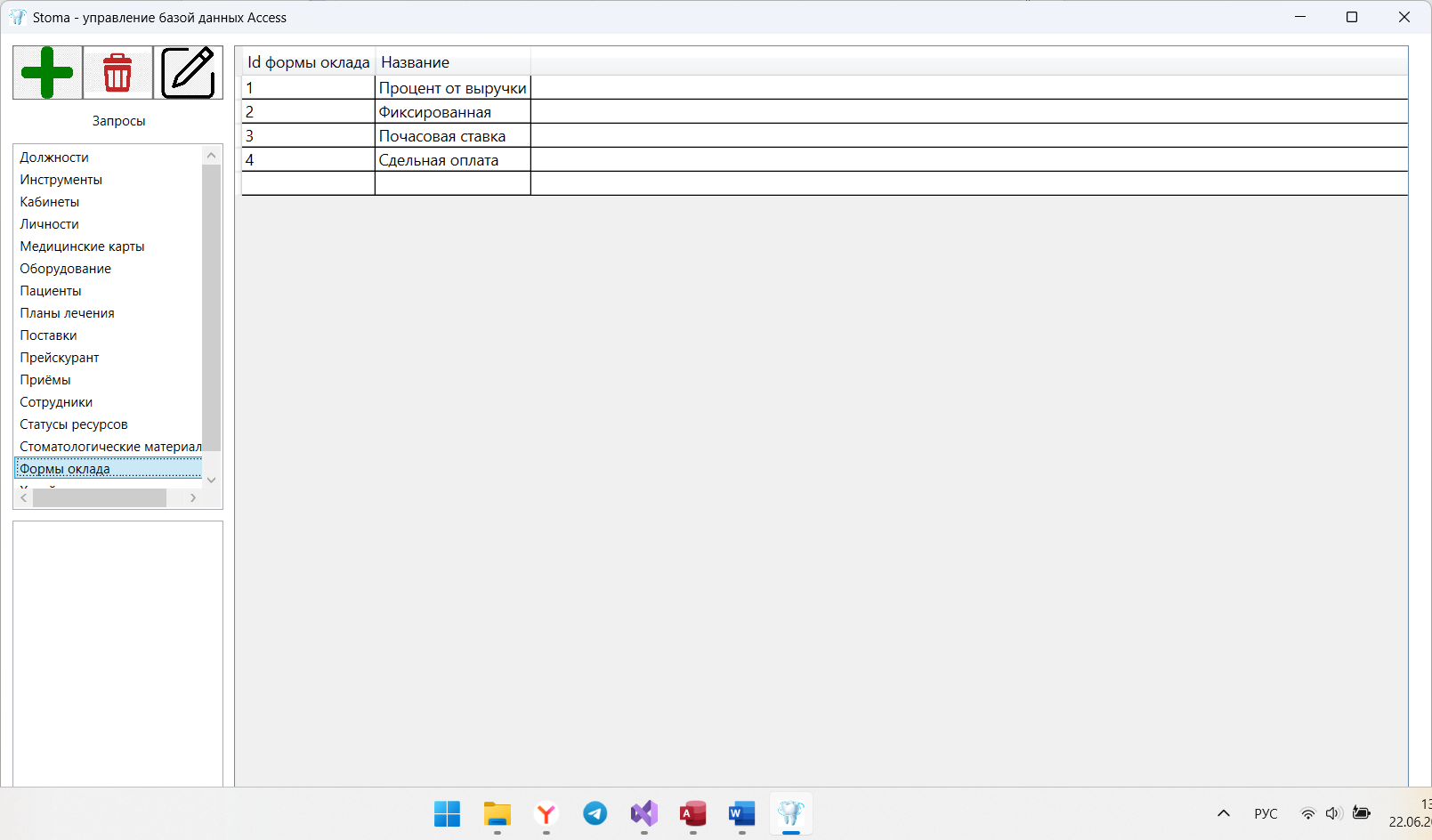


Рис. П3.16. Пример вывода всех записей из таблицы «Формы оклада»

Завершение работы

Для завершения работы программы необходимо закрыть рабочее окно. Закрытие осуществляется нажатием крестика «❌» в правом верхнем углу окна приложения (рис. П3.17).

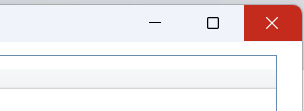


Рис. П3.17. Пример кнопки закрытия программы