МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева»

(Самарский университет)

Институт информатики и кибернетики

Кафедра информационных систем и технологий

**ОТЧЕТ ПО ПРАКТИКЕ**

Вид практики производственная

(учебная, производственная)

Тип практики научно-исследовательская работа

(в соответствии с ОПОП ВО)

Сроки прохождения практики: с 06.02.2023 по 10.06.2023

(в соответствии с календарным учебным графиком)

по направлению подготовки 09.03.01 Информатика и вычислительная техника

(уровень бакалавриата)

направленность (профиль) «Информационные системы»

Обучающийся группы № 6303-090301D\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. П. Андреев

Руководитель практики,

к. т. н., Доцент кафедры ИСТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е. И. Чигарина

Дата сдачи 10.06.2023

Дата защиты 10.06.2023

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Самара 2023

**Содержание**

1. Задание(я) для выполнения определенных видов работ, связанных с будущей профессиональной деятельностью (сбор и анализ данных и материалов, проведение исследований).
2. Описательная часть.
3. Заключение.
4. Список использованных источников.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение

высшего образования «Самарский национальный исследовательский университет

имени академика С.П. Королева»

(Самарский университет)

Институт информатики и кибернетики

Кафедра информационных систем и технологий

**Задание(я) для выполнения определенных видов работ, связанных с будущей профессиональной деятельностью (сбор и анализ данных и материалов, проведение исследований)**

Обучающемуся \_\_\_\_\_\_\_\_\_\_\_\_\_\_Андрееву Дмитрию Павловичу\_\_\_\_\_\_\_\_\_

группы 6303-090301D.

Направление на практику оформлено приказом по университету

от 03.02.2023 г. № 76-ПР

на \_\_\_\_\_\_\_\_\_\_кафедру информационных систем и технологий\_\_\_\_\_\_\_\_\_\_\_\_\_

(наименование профильной организации или структурного подразделения университета)

Тема НИР:\_\_\_\_\_ «Разработка автоматизированной информационной\_\_   
системы учёта пациентов клинической железнодорожной больницы\_\_\_   
города Самары» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| Планируемые результаты освоения образовательной программы (компетенции) | Планируемые результаты практики | Содержание задания |
| ПК-6 Способен осуществлять разработку требований и проектирование программного обеспечения  ПК 6.2 Разрабатывает технические спецификации на программные компоненты и их взаимодействие | Знать: технологии разработки технических спецификаций программных компонентов и их взаимодействия.  Уметь: обосновывать выбор технологии разработки технических спецификаций программных компонентов.  Владеть: навыками разработки технических спецификаций программных компонентов. | Провести разработку информационно-логического проекта информационной системы учёта пациентов клинической железнодорожной больницы города Самары  Сделать описание технической спецификации на программные компоненты и их взаимодествие в информационной системе учёта пациентов клинической железнодорожной больницы города Самары |

Дата выдачи задания 06.02.2023.

Срок представления на кафедру отчета о практике 10.06.2023.

Руководитель практики,

к. т. н., Доцент кафедры ИСТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е. И. Чигарина

*(подпись)*

Задание принял к исполнению

обучающийся группы № 6303-090301D \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. П. Андреев

*(подпись)*

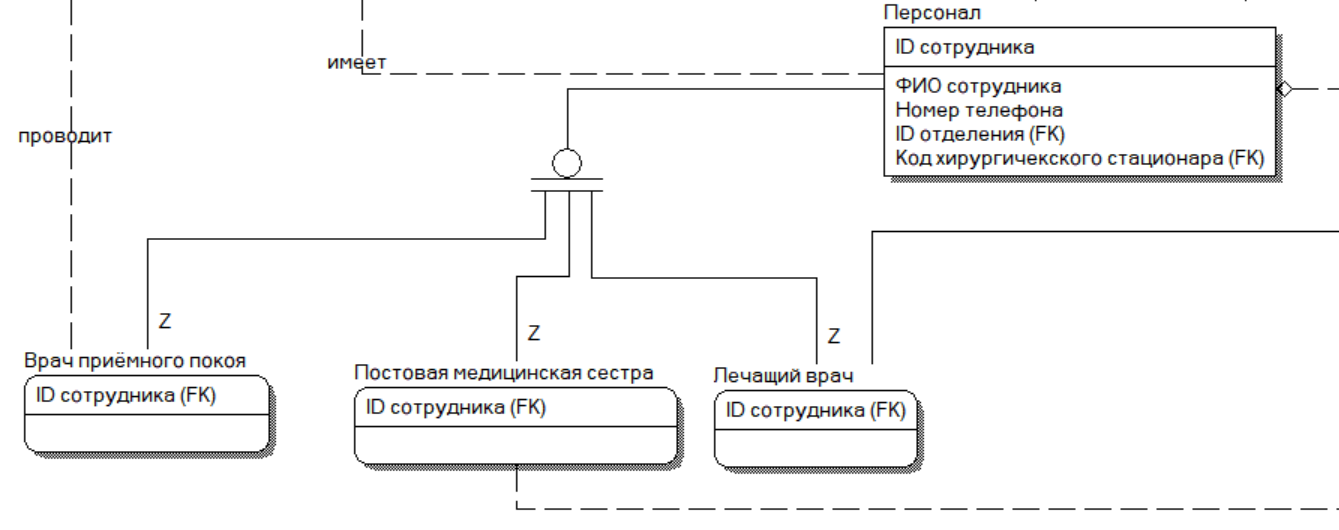
**Описательная часть**

1. Описание информационной модели АИС учёта пациентов клинической железнодорожной больницы города Самары по методологии IDEF1x.

Методология моделирования IDEF1X, являясь расширением стандарта IDEF1, предназначена для описания данных (информации). В ее основе лежит язык семантического моделирования, основанного на концепции "сущность — связь", позволяющей определять данные и связи между ними. Методология используется для создания информационной модели предметной области с помощью идентификации ее сущностей и связей между ними. Чаще всего такая методология используется для описания данных в целях последующей автоматизации их обработки с помощью систем управления базами данных. Таким образом, можно говорить о том, что модели данных в нотации IDEF1X используются для создания баз данных.

На рисунках 1-3 изображена логическая модель базы данных автоматизированной информационной системы учёта пациентов клинической железнодорожной больницы города Самары.

Рисунок 1 – Подсхема «Работа с пациентом»

Рисунок 2 – Подсхема «Работа с сотрудниками»

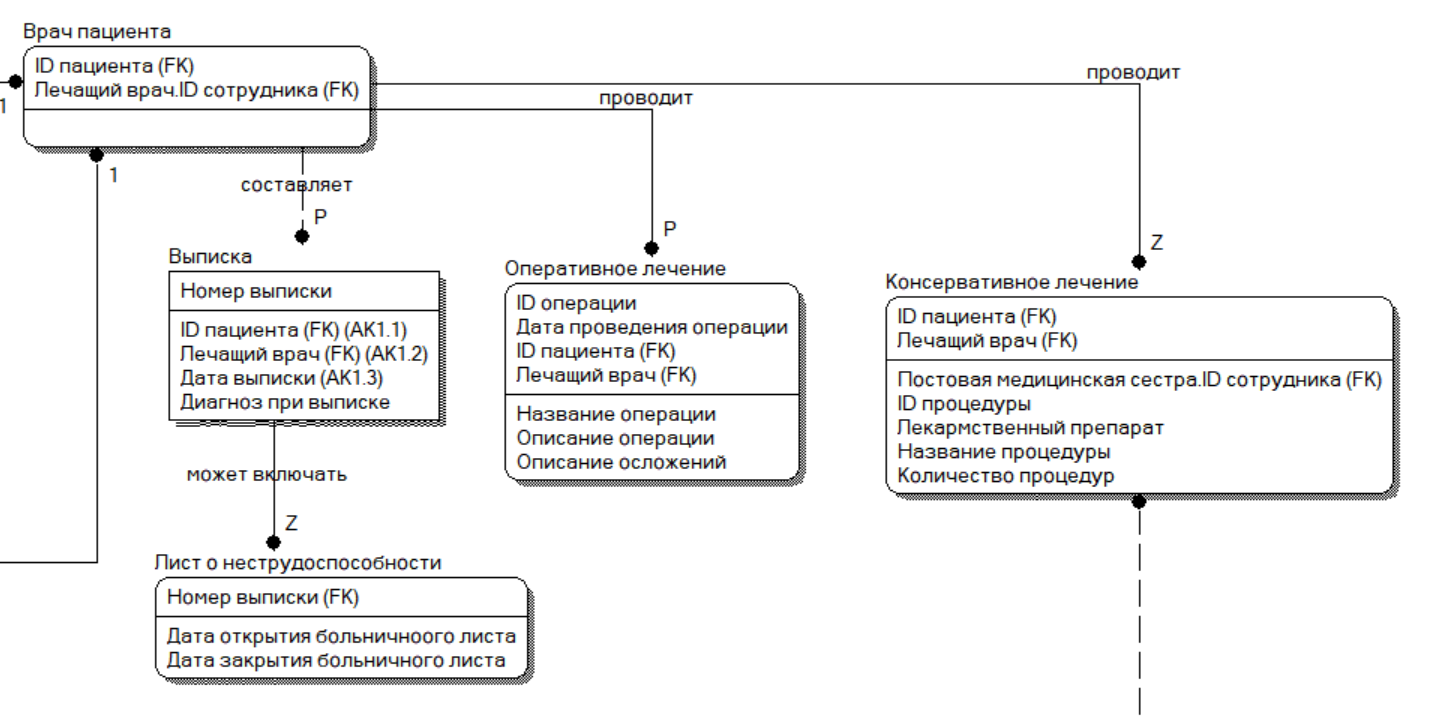


Рисунок 3 – Подсхема «Работа с данными о пациенте и назначенным ему леччением»

Для хранения информации о пациенте была добавлена сущность «Пациент», которая содержит такие атрибуты как ОМС пациента, СНИЛС пациента, диагноз с которым поступил пациент.

О пациенте содержится дополнительная информация в сущности «Дополнительная информация», в которой содержится информация о группе крови пациента, резус факторе, а так же о возможном наличии аллергий.

Для того чтобы описать отделение, была создана сущность «Вид отделения», где хранится ID отделения и его название.

Сущность «Отделение» содержит информацию о отделении в котором будет лежать пациент, для этого были описаны следующие атрибуты: ID отделения, код хирургического стационара, а так же ID сотрудника, который под собой подразумевает ID сотрудника, который является заведующим отделением.

Для хранения информации о палатах в медицинском учреждении была создана сущность «Палата», которая содержит информацию о номере палаты, ID отделения, коде хирургического стационара, количестве свободных мест в палате, количестве мест в палате.

Для хранения информации о сотрудниках больницы была создана сущность «Персонал», которая содержит такие атрибуты как ID сотрудника, ФИО сотрудника, номер телефона, ID отделения, к которому причислен сотрудник, код хирургического отделения.

Сущность «Паспортные данные» содержит информацию о паспортных данных как сотрудников, так и сотрудников больницы, содержит такие атрибуты как код паспорта, дата выдачи паспорта, номер паспорта, кем выдан паспорт, серия паспорта, адрес прописки, ОМС пациента, ID сотрудника.

Появилась необходимость так же учитывать в каком хирургическом стационаре находится пациент или же работает сотрудник больницы, котому была добавлена сущность «Хирургический стационар», который содержит в себе код хирургического стационара, название хирургического стационара, адрес хирургического стационара, телефон регистратуры, ID сотрудника

Такие сущности как «Врач приёмного покоя», «Постовая медицинская сестра», «Лечащий врач» описывают сотрудников хирургического стационара, все они содержат в себе ID сотрудника.

Была создана сущность «Первичный осмотр» для хранения информации о первом осмотре пациента, когда он ложится в хирургический стационар. Данная сущность хранит в себе информацию о ОМС пациента, дате поступления, диагноз, враче приёмного покоя, который проводил осмотр.

Для того чтобы регулировать попадание и присваивание пациенту койко-места в палате была создана сущность «Пациент в палате», она включает в себя такие поля как ID пациента, ОМС пациента, дата поступления, номер палаты, ID отделения, Код хирургического стационара.

Для регулирования какой врач какого пациента лечит была создана сущность «Врач пациента», которая содержит информацию о ID пациента, а так же о лечащем враче.

Лечащий врач создаёт такой документ как «Выписка», потому была создана сущность с таким же названием, она включает в себя такие поля как номер выписки, ID пациента, лечащий врач пациента, дата выписки, диагноз при выписке.

Лечащий врач так же по необходимости пишет такой документ как «Лист о нетрудоспособности», который содержит в себе информацию о номере выписки, дате открытия листа о нетрудоспособности, дате закрытия листа о нетрудоспособности.

Для хранения информации о оперативном лечении пациента, бала создана сущность «Оперативное лечение», которая хранит информацию о ID операции, дата проведения операции, ID пациента, лечащий врач, название операции, описание операции, описание осложнений.

Для ведении данных по консервативному лечению была создана сущность «Консервативное лечение», которая содержит такие атрибуты как ID пациента, лечащего врача, постовой медицинской сестры, ID процедуры, лекарственный препарат, название процедуры, количество процедур.

2. Нормализация отношений реляционной базы данных АИС учёта пациентов клинической железнодорожной больницы города Самары.

2.1 Схема отношений:

1. Пациент (ОМС пациента, Диагноз с которым поступил, СНИЛС, ФИО пациента)
2. Дополнительные сведения (ОМС пациента (FK), Аллергия, Резус фактор, Группа крови)
3. Вид отделения (ID отделения, Название отделения)
4. Отделение (Код хирургического стационара (FK), ID отделения (FK), Заведующий отделением.ID сотрудника (FK))
5. Палата (Номер палаты, Код хирургического стационара (FK), ID отделения (FK), Количество свободных мест в палате, Количество мест в палате)
6. Персонал (ID сотрудника, ФИО, Номер телефона, ID отделения (FK), Код хирургического стационара (FK))
7. Паспортные данные (Код паспорта, Дата выдачи паспорта, Номер паспорта, Кем выдан паспорт, Серия паспорта, Адрес прописки, ID сотрудника (FK), ОМС пациента (FK))
8. Хирургический стационар (Код хирургического стационара, Название хирургического стационара, Адрес хирургического стационара, Телефон регистратуры, Главный врач.ID сотрудика (FK))
9. Врач приёмного покоя (ID сотрудника (FK))
10. Постовая медицинская сестра (ID сотрудника (FK))
11. Лечащий врач (ID сотрудника (FK))
12. Первичный осмотр (ОМС пациента (FK), Дата поступления, Врач приёмного покоя.ID сотрудника, Диагноз)
13. Пациент в палате (ID пациента, ОМС пациента (FK), Код хирургического стационара (FK), Дата поступления (FK), Номер палаты (FK), ID Отделения (FK))
14. Врач пациента (Лечащий врач.ID сотрудника (FK), ID пациента (FK))
15. Выписка (Номер выписки, ID пациента (FK), Лечащий врач (FK), Дата выписки, Диагноз при выписке)
16. Лист о нетрудоспособности (Номер выписки (FK), Дата начала больничного, Дата окончания больничного)
17. Оперативное лечение (ID операции, Дата проведения операции, Лечащий врач (FK), ID пациента (FK), Название операции, Описание осложнений, Описание операции)
18. Консервативное лечение (Лечащий врач (FK), ID пациента (FK), ID процедуры, Медицинская сестра.ID сотрудника (FK), Лекарственный препарат, Название процедуры, Количество процедур)

2.2 Проверка на первую нормальную форму

Проверка на 1-ю нормальную форму. Все отношения находятся в 1-й нормальной форме, т.к. значения атрибутов во всех отношениях атомарны, то есть не являются списком или множеством, и могут быть описаны определенным типом данных.

2.3 Проверка на вторую нормальную форму

Проверка на 2-ю нормальную форму. Отношения «Пациент», «Дополнительные сведения», «Вид отделения», «Персонал», «Хирургический стационар», «Лист о нетрудоспособности», «Пациент в палате» находятся во 2-й нормальной форме, так как не имеют составных ключей. Отношения «Врач приёмного покоя», «Постовая медицинская сестра», «Лечащий врач», «Врач пациента» находятся во 2-й нормальной форме, так как отсутствуют атрибуты, не входящие в ключ и не являющиеся ключами. Требуется проверить остальные отношения.

Проверим отношение «Отделение», в котором один составной ключ: «Код хирургического стационара, ID отделения» и атрибут «Заведующий отделением.ID сотрудника», не входящий в ключ и не являющийся ключами.

По определению, от ключа функционально зависят все атрибуты отношения:

«Код хирургического стационара, ID отделения» --> «Заведующий отделением.ID сотрудника»

Проверяем наличие частных функциональных зависимостей:

«Код хирургического стационара» -/-> «Заведующий отделением.ID сотрудника»

«ID отделения» -/-> «Заведующий отделением.ID сотрудника»

В результате проверки видно, что частичные функциональные зависимости отсутствуют и отношение «Отделение» находится во второй нормальной форме.

Проверим отношение «Палата», в котором один составной ключ: «Номер палаты, Код хирургического стационара, ID отделения» и атрибуты «Количество свободных мест в палате, Количество мест в палате», не входящие в ключ и не являющиеся ключами.

По определению, от ключа функционально зависят все атрибуты отношения:

«Номер палаты, Код хирургического стационара, ID отделения» --> «Количество свободных мест в палате»

«Номер палаты, Код хирургического стационара, ID отделения»--> «Количество мест в палате»

Проверяем наличие частных функциональных зависимостей:

«Номер палаты» -/-> «Количество свободных мест в палате»

«Код хирургического стационара» -/-> «Количество свободных мест в палате»

«ID отделения» -/-> «Количество свободных мест в палате»

«Номер палаты» -/-> «Количество мест в палате»

«Код хирургического стационара» -/-> «Количество мест в палате»

«ID отделения» -/-> «Количество мест в палате»

В результате проверки видно, что частичные функциональные зависимости отсутствуют и отношение «Палата» находится во второй нормальной форме.

Проверим отношение «Паспортные данные», в котором один составной ключ: «Кем выдан паспорт, Номер паспорта, Серия паспорта, Дата выдачи паспорта» и атрибуты «Адрес прописки, ID сотрудника, ОМС пациента» , не входящие в ключ и не являющиеся ключами.

По определению, от ключа функционально зависят все атрибуты отношения:

«Кем выдан паспорт, Номер паспорта, Серия паспорта, Дата выдачи паспорта» --> «Адрес прописки»

«Кем выдан паспорт, Номер паспорта, Серия паспорта, Дата выдачи паспорта»--> «ID сотрудника»

«Кем выдан паспорт, Номер паспорта, Серия паспорта, Дата выдачи паспорта»--> «ОМС пациента»

Проверяем наличие частных функциональных зависимостей:

«Кем выдан паспорт» -/-> «Адрес прописки»

«Номер паспорта» -/-> «Адрес прописки»

«Серия паспорта» -/-> «Адрес прописки»

«Дата выдачи паспорта» -/-> «Адрес прописки»

«Кем выдан паспорт» -/-> «ID сотрудника»

«Номер паспорта» -/-> «ID сотрудника»

«Серия паспорта» -/-> «ID сотрудника»

«Дата выдачи паспорта» -/-> «ID сотрудника»

«Кем выдан паспорт» -/-> «ОМС пациента»

«Номер паспорта» -/-> «ОМС пациента»

«Серия паспорта» -/-> «ОМС пациента»

«Дата выдачи паспорта» -/-> «ОМС пациента»

В результате проверки видно, что частичные функциональные зависимости отсутствуют и отношение «Паспортные данные» находится во второй нормальной форме.

Проверим отношение «Первичный осмотр», в котором один составной ключ: «ОМС пациента, Дата поступления» и атрибуты «Врач приёмного покоя.ID сотрудника, Диагноз», не входящие в ключ и не являющиеся ключами.

По определению, от ключа функционально зависят все атрибуты отношения:

«ОМС пациента, Дата поступления» --> «Врач приёмного покоя.ID сотрудника»

«ОМС пациента, Дата поступления» --> «Диагноз»

Проверяем наличие частных функциональных зависимостей:

«ОМС пациента» -/-> «Врач приёмного покоя.ID сотрудника»

«Дата поступления» -/-> «Врач приёмного покоя.ID сотрудника»

«ОМС пациента» -/-> «Диагноз»

«Дата поступления» -/-> «Диагноз»

В результате проверки видно, что частичные функциональные зависимости отсутствуют и отношение «Первичный осмотр» находится во второй нормальной форме.

Проверим отношение «Выписка», в котором один составной ключ: «Лечащий врач, ID пациента, Дата выписки» и атрибут «Диагноз при выписке», не входящие в ключ и не являющиеся ключами.

По определению, от ключа функционально зависят все атрибуты отношения:

«Лечащий врач, ID пациента, Дата выписки» --> «Диагноз при выписке»

Проверяем наличие частных функциональных зависимостей:

«Лечащий врач» -/-> «Диагноз при выписке»

«ID пациента» -/-> «Диагноз при выписке»

«Дата выписки» -/-> «Диагноз при выписке»

В результате проверки видно, что частичные функциональные зависимости отсутствуют и отношение «Выписка» находится во второй нормальной форме.

Проверим отношение «Оперативное лечение», в котором один составной ключ: «ID операции, Дата проведения операции, Лечащий врач, ID пациента» и атрибуты «Название операции, Описание операции, Описание осложнений», не входящие в ключ и не являющиеся ключами.

По определению, от ключа функционально зависят все атрибуты отношения:

«ID операции, Дата проведения операции, Лечащий врач, ID пациента»--> «Название операции»

«ID операции, Дата проведения операции, Лечащий врач, ID пациента»--> «Описание операции»

«ID операции, Дата проведения операции, Лечащий врач, ID пациента»--> «Описание осложнений»

Проверяем наличие частных функциональных зависимостей:

«ID операции» -/-> «Название операции»

«Дата проведения операции» -/-> «Название операции»

«Лечащий врач» -/-> «Название операции»

«ID пациента» -/-> «Название операции»

«ID операции» -/-> «Описание операции»

«Дата проведения операции» -/-> «Описание операции»

«Лечащий врач» -/-> «Описание операции»

«ID пациента» -/-> «Описание операции»

«ID операции» -/-> «Описание осложнений»

«Дата проведения операции» -/-> «Описание осложнений»

«Лечащий врач» -/-> «Описание осложнений»

«ID пациента» -/-> «Описание осложнений»

В результате проверки видно, что частичные функциональные зависимости отсутствуют и отношение «Оперативное лечение» находится во второй нормальной форме.

Проверим отношение «Консервативное лечение», в котором один составной ключ: «Лечащий врач, ID пациента» и атрибуты «ID процедуры, Медицинская сестра.ID сотрудника, Лекарственный препарат, Название процедуры, Количество процедур», не входящие в ключ и не являющиеся ключами.

По определению, от ключа функционально зависят все атрибуты отношения:

«Лечащий врач, ID пациента» --> «ID процедуры»

«Лечащий врач, ID пациента» --> «Медицинская сестра.ID сотрудника»

«Лечащий врач, ID пациента» --> «Лекарственный препарат»

«Лечащий врач, ID пациента» --> «Название процедуры»

«Лечащий врач, ID пациента» --> «Количество процедур»

Проверяем наличие частных функциональных зависимостей:

«Лечащий врач» -/-> «ID процедуры»

«Лечащий врач» -/-> «Медицинская сестра.ID сотрудника»

«Лечащий врач» -/-> «Лекарственный препарат»

«Лечащий врач» -/-> «Название процедуры»

«Лечащий врач» -/-> «Количество процедур»

«ID пациента» -/-> «ID процедуры»

«ID пациента» -/-> «Медицинская сестра.ID сотрудника»

«ID пациента» -/-> «Лекарственный препарат»

«ID пациента» -/-> «Название процедуры»

«ID пациента» -/-> «Количество процедур»

В результате проверки видно, что частичные функциональные зависимости отсутствуют и отношение «Консервативное лечение» находится во второй нормальной форме.

Проверим отношение «Хирургический стационар», в котором три альтернативных ключа: «Название хирургического стационара, Адрес хирургического стационара, Телефон регистратуры» и атрибут «Главный врач.ID сотрудника», не входящие в ключ и не являющиеся ключами.

По определению, от ключа функционально зависят все атрибуты отношения:

«Название хирургического стационара, Адрес хирургического стационара, Телефон регистратуры» --> «Главный врач.ID сотрудника»

Проверяем наличие частных функциональных зависимостей:

«Название хирургического стационара» -/-> «Главный врач.ID сотрудника»

«Адрес хирургического стационара» -/-> «Главный врач.ID сотрудника»

«Телефон регистратуры» -/-> «Главный врач.ID сотрудника»

В результате проверки видно, что частичные функциональные зависимости отсутствуют и отношение «Хирургический стационар» находится во второй нормальной форме.

2.4 Проверка на третью нормальную форму

Проверка на 3-ю нормальную форму. Отношения «Пациент», «Вид отделения», «Отделение», «Палата», « Хирургический стационар», «Первичный осмотр», «Пациент в палате», «Врач пациента», «Выписка» , «Лист о нетрудоспособности», «Врач приёмного покоя», «Постовая медицинская сестра», «Лечащий врач» находятся в 3-й нормальной форме т.к. они находятся во второй нормальной форме и число атрибутов в этих отношениях, которые не являются потенциальными ключами, меньше двух. Требуется проверить отношения остальные отношения на 3-ю нормальную форму.

Проверим на наличие транзитивных зависимостей отношение «Дополнительные сведения» - оно имеет первичный ключ «ОМС пациента» и атрибуты: «Аллергия, Резус фактор, Группа крови», не входящие в ключ и не являющиеся ключами.

«ОМС пациента» -> Аллергия -/-> Резус фактор

«ОМС пациента» -> Аллергия -/-> Группа крови

«ОМС пациента» -> Резус фактор -/-> Аллергия

«ОМС пациента» -> Резус фактор -/-> Группа крови

«ОМС пациента» -> Группа крови -/-> Резус фактор

«ОМС пациента» -> Группа крови -/-> Аллергия

Таким образом, в отношении «Дополнительные сведения» отсутствуют транзитивные функциональные зависимости, и оно в 3 нормальной форме.

Проверим на наличие транзитивных зависимостей отношение «Персонал» - оно имеет первичный ключ «ID сотрудника» и атрибуты: «ФИО, Номер телефона, ID отделения, Код хирургического стационара», не входящие в ключ и не являющиеся ключами.

«ID сотрудника» -> ФИО -/-> Номер телефона

«ID сотрудника» -> ФИО -/-> ID отделения

«ID сотрудника» -> ФИО -/-> Код хирургического стационара

«ID сотрудника» -> Номер телефона -/-> ФИО

«ID сотрудника» -> Номер телефона -/-> ID отделения

«ID сотрудника» -> Номер телефона -/-> Код хирургического стационара

«ID сотрудника» -> Код хирургического стационара -/-> ФИО

«ID сотрудника» -> Код хирургического стационара -/-> ID отделения

«ID сотрудника» -> Код хирургического стационара -/-> Номер телефона

Таким образом, в отношении «Персонал» отсутствуют

транзитивные функциональные зависимости, и оно в 3 нормальной форме.

Проверим на наличие транзитивных зависимостей отношение «Паспортные данные» - оно имеет первичный ключ «Код паспорта», альтернативный составной ключ «Кем выдан паспорт, Номер паспорта, Серия паспорта, Дата выдачи паспорта» и атрибуты: «Адрес прописки, ID сотрудника, ОМС пациента», не входящие в ключ и не являющиеся ключами.

«Код паспорта» -> Адрес прописки -/-> ID сотрудника

«Код паспорта» -> Адрес прописки -/-> ОМС пациента

«Код паспорта» -> ID сотрудника -/-> Адрес прописки

«Код паспорта» -> ID сотрудника -/-> ОМС пациента

«Код паспорта» -> ОМС пациента -/-> Адрес прописки

«Код паспорта» -> ОМС пациента -/-> ID сотрудника

«Кем выдан паспорт, Номер паспорта, Серия паспорта, Дата выдачи паспорта» -> Адрес прописки -/-> ID сотрудника

«Кем выдан паспорт, Номер паспорта, Серия паспорта, Дата выдачи паспорта» -> Адрес прописки -/-> ОМС пациента

«Кем выдан паспорт, Номер паспорта, Серия паспорта, Дата выдачи паспорта» -> ID сотрудника -/-> Адрес прописки

«Кем выдан паспорт, Номер паспорта, Серия паспорта, Дата выдачи паспорта» -> ID сотрудника -/-> ОМС пациента

«Кем выдан паспорт, Номер паспорта, Серия паспорта, Дата выдачи паспорта» -> ОМС пациента -/-> Адрес прописки

«Кем выдан паспорт, Номер паспорта, Серия паспорта, Дата выдачи паспорта» -> ОМС пациента -/-> ID сотрудника

Таким образом, в отношении «Паспортные данные» отсутствуют

транзитивные зависимости, и оно в 3 нормальной форме.

Проверим на наличие транзитивных зависимостей отношение «Оперативное лечение» - оно имеет первичный ключ «ID операции, Дата проведения операции, Лечащий врач, ID пациента» и атрибуты: «Название операции, Описание операции, Описание осложнений», не входящие в ключ и не являющиеся ключами.

«ID операции, Дата проведения операции, Лечащий врач, ID пациента» -> Название операции -/-> Описание операции

«ID операции, Дата проведения операции, Лечащий врач, ID пациента» -> Название операции -/-> Описание осложнений

«ID операции, Дата проведения операции, Лечащий врач, ID пациента» -> Описание операции -/-> Название операции

«ID операции, Дата проведения операции, Лечащий врач, ID пациента» -> Описание операции -/-> Описание осложнений

«ID операции, Дата проведения операции, Лечащий врач, ID пациента» -> Описание осложнений -/-> Название операции

«ID операции, Дата проведения операции, Лечащий врач, ID пациента» -> Описание осложнений -/-> Описание операции

Таким образом, в отношении «Оперативное лечение» отсутствуют транзитивные функциональные зависимости, и оно в 3 нормальной форме.

Проверим на наличие транзитивных зависимостей отношение «Консервативное лечение» - оно имеет первичный ключ «Лечащий врач, ID пациента» и атрибуты: «ID процедуры, Медицинская сестра.ID сотрудника, Лекарственный препарат, Название процедуры, Количество процедур», не входящие в ключ и не являющиеся ключами.

«Лечащий врач, ID пациента» -> ID процедуры -/-> Медицинская сестра.ID сотрудника

«Лечащий врач, ID пациента» -> ID процедуры -/-> Лекарственный препарат

«Лечащий врач, ID пациента» -> ID процедуры -/-> Название процедуры

«Лечащий врач, ID пациента» -> ID процедуры -/-> Количество процедур

«Лечащий врач, ID пациента» -> Медицинская сестра.ID сотрудника -/-> ID процедуры

«Лечащий врач, ID пациента» -> Медицинская сестра.ID сотрудника -/-> Лекарственный препарат

«Лечащий врач, ID пациента» -> Медицинская сестра.ID сотрудника -/-> Название процедуры

«Лечащий врач, ID пациента» -> Медицинская сестра.ID сотрудника -/-> Количество процедур

«Лечащий врач, ID пациента» -> Лекарственный препарат -/-> Медицинская сестра.ID сотрудника

«Лечащий врач, ID пациента» -> Лекарственный препарат -/-> ID процедуры

«Лечащий врач, ID пациента» -> Лекарственный препарат -/-> Название процедуры

«Лечащий врач, ID пациента» -> Лекарственный препарат -/-> Количество процедур

«Лечащий врач, ID пациента» -> Название процедуры -/-> Медицинская сестра.ID сотрудника

«Лечащий врач, ID пациента» -> Название процедуры -/-> Лекарственный препарат

«Лечащий врач, ID пациента» -> Название процедуры -/-> ID процедуры

«Лечащий врач, ID пациента» -> Название процедуры -/-> Количество процедур

«Лечащий врач, ID пациента» -> Количество процедур -/-> Медицинская сестра.ID сотрудника

«Лечащий врач, ID пациента» -> Количество процедур -/-> Лекарственный препарат

«Лечащий врач, ID пациента» -> Количество процедур -/-> Название процедуры

«Лечащий врач, ID пациента» -> Количество процедур -/-> ID пациента

Таким образом, в отношении «Консервативное лечение» отсутствуют транзитивные функциональные зависимости, и оно в 3 нормальной форме.

Проверим на наличие транзитивных зависимостей отношение «Пациент» - оно имеет первичный ключ «ОМС пациента», альтернативный ключ «СНИЛС» и атрибуты: «Диагноз с которым поступил, ФИО пациента», не входящие в ключ и не являющиеся ключами.

«ОМС пациента» -> Диагноз с которым поступил -/-> ФИО пациента

«ОМС пациента» -> ФИО пациента -/-> Диагноз с которым поступил

Таким образом, в отношении «Пациент» отсутствуют

транзитивные зависимости, и оно в 3 нормальной форме.

Проверим на наличие транзитивных зависимостей отношение «Палата» - оно имеет первичный ключ «Номер палаты, Код хирургического отделения, ID отделения» и атрибуты: «Количество свободных мест в палате, Количество мест в палате», не входящие в ключ и не являющиеся ключами.

«Номер палаты, Код хирургического отделения, ID отделения» -> Количество свободных мест в палате -/-> Количество мест в палате

«Номер палаты, Код хирургического отделения, ID отделения» -> Количество мест в палате -/-> свободных мест в палате

Таким образом, в отношении «Палата» отсутствуют

транзитивные зависимости, и оно в 3 нормальной форме.

Проверим на наличие транзитивных зависимостей отношение «Первичный осмотр» - оно имеет первичный ключ «ОМС пациента, Дата поступления» и атрибуты: «Врач приёмного отделения.ID сотрудника, Диагноз», не входящие в ключ и не являющиеся ключами.

«ОМС пациента, Дата поступления» -> Врач приёмного отделения.ID сотрудника -/-> Диагноз

«ОМС пациента, Дата поступления» -> Диагноз -/-> Врач приёмного отделения.ID сотрудника

Таким образом, в отношении «Первичный осмотр» отсутствуют

транзитивные зависимости, и оно в 3 нормальной форме.

Проверим на наличие транзитивных зависимостей отношение «Лист о нетрудоспособности» - оно имеет первичный ключ «Номер выписки» и атрибуты: «Дата начала больничного, Дата окончания больничного», не входящие в ключ и не являющиеся ключами.

«Номер выписки» -> Дата начала больничного -/-> Дата окончания больничного

«Номер выписки» -> Дата окончания больничного -/-> Дата начала больничного

Таким образом, в отношении «Лист о нетрудоспособности» отсутствуют транзитивные зависимости, и оно в 3 нормальной форме.

3. Выбор средств реализации АИС учёта пациентов клинической железнодорожной больницы города Самары.

3.1 Выбор операционной системы

Для реализации информационной системы была выбрана Windows 10, выбор был обусловлен, простотой использования, производительностью а так же безопасностью операционной системы. Из-за всех этих достоинств для Windows 10 поддерживается огромное множество приложений и программных обеспечений.

3.2 Выбор средства реализации базы данных

В качестве языка программирования и среды разработки были выбраны   
язык программирования С# и среда разработки ПО Microsoft Visual Studio   
2022.

C# — объектно-ориентированный язык программирования общего назначения. Разработан в в компании [Microsoft](https://ru.wikipedia.org/wiki/Microsoft) как язык разработки приложений для платформы Microsoft .NET Framework и .NET Core.

Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, переменные, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Visual Studio — это комплексная интегрированная среда разработки   
(IDE), которую можно использовать для написания, редактирования, отладки   
и сборки кода, а затем для публикации приложения. Помимо редактирования   
и отладки кода. Visual Studio включает графический конструктор, через который создавать графический интерфейс становится очень просто.

3.3 Выбор средства реализации приложения системы

Для разработки автоматизированной информационной системы была   
выбрана система управления базой данных (СУБД) Microsoft SQL Server. Её основные преимущества: масштабируемость и надежность. SQL Server 2014 обеспечивает практически неограниченный рост объемов хранения за счет увеличения надежности и масштабируемости системы, используя все преимущества мультипроцессорной обработки файлов.

Это безопасная, надежная, масштабируемая платформа, защищающая информацию в приложениях и повышающая её доступность. Скорость создания решений. SQL Server 2014 в сочетании с .NET Framework уменьшает время разработки, внедрения и выхода на рынок современных приложений, ускоряет процесс поиска информации, упрощает управление, позволяет использовать создаваемые пользователем функции в других приложениях, предоставляет широкие возможности для создания Web-приложений.

Для управления Microsoft SQL Server была выбрана среда SQL Server   
Management Studio 14. Среда SQL Server Management Studio

3.4 Выбор технологии доступа к данным

В качестве технологии доступа к базе данных была выбрана технология ADO.NET.

ADO.NET — технология, предоставляющая доступ и управление данными, хранящимся в базе данных или других источниках, основанных на платформе .NET Framework и входящая в состав .NET Framework 2.0, представляет собой набор библиотек, через которые можно отправлять запросы к базам данных, устанавливать подключения, получать ответ от базы данных и производить ряд других операций.

3.5 Выбор средства генерации отчётов

SAP Crystal Reports – это инструмент бизнес-аналитики, который используется для создания отчетов из источников данных как SAP, так и сторонних производителей. Это позволяет конечным пользователям создавать отчеты, которые включают визуализации и внедрять новые бизнес-требования в отчеты, чтобы уменьшить зависимость от разработчиков отчетов.

SAP Crystal Reports может подключаться к любому источнику данных, который включает в себя реляционные базы данных, такие как Oracle, системы источников данных OLAP, такие как BW, или также с данными XML. Существует возможность создать простой отчет или использовать сложный или специализированный инструмент Crystal Reports для создания отчетов для конечных пользователей.

4. Описание физической модели базы данных АИС учёта пациентов клинической железнодорожной больницы города Самары.

В соответствии с логической моделью была создана физическая модель базы данных, представленная на рисунках 4-6.

Для каждого атрибута каждой сущности был определен тип данных в соответствии с его функционалом.

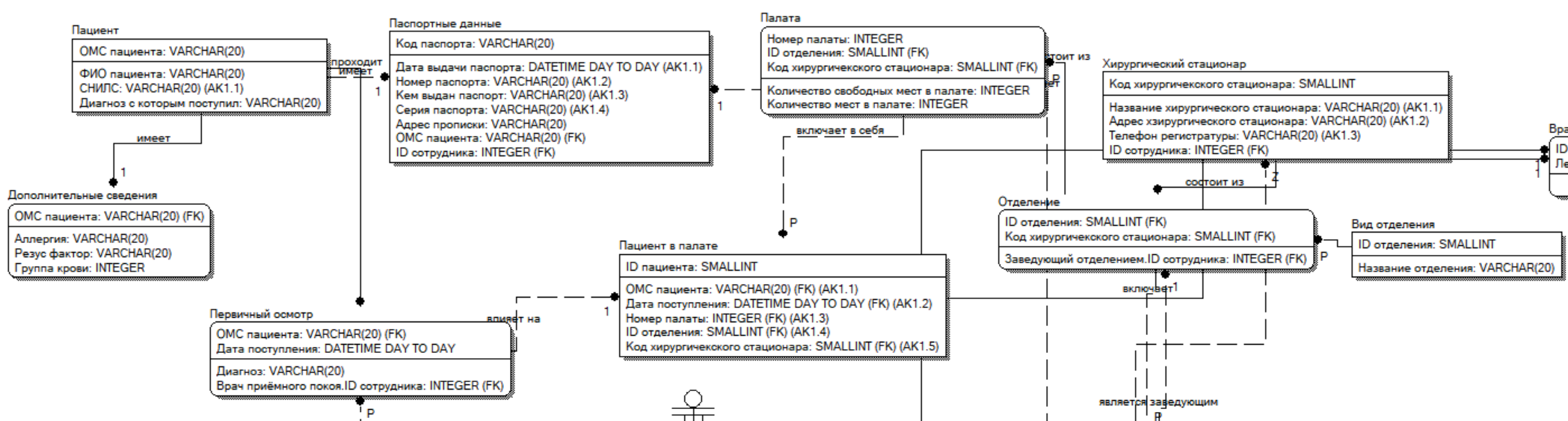
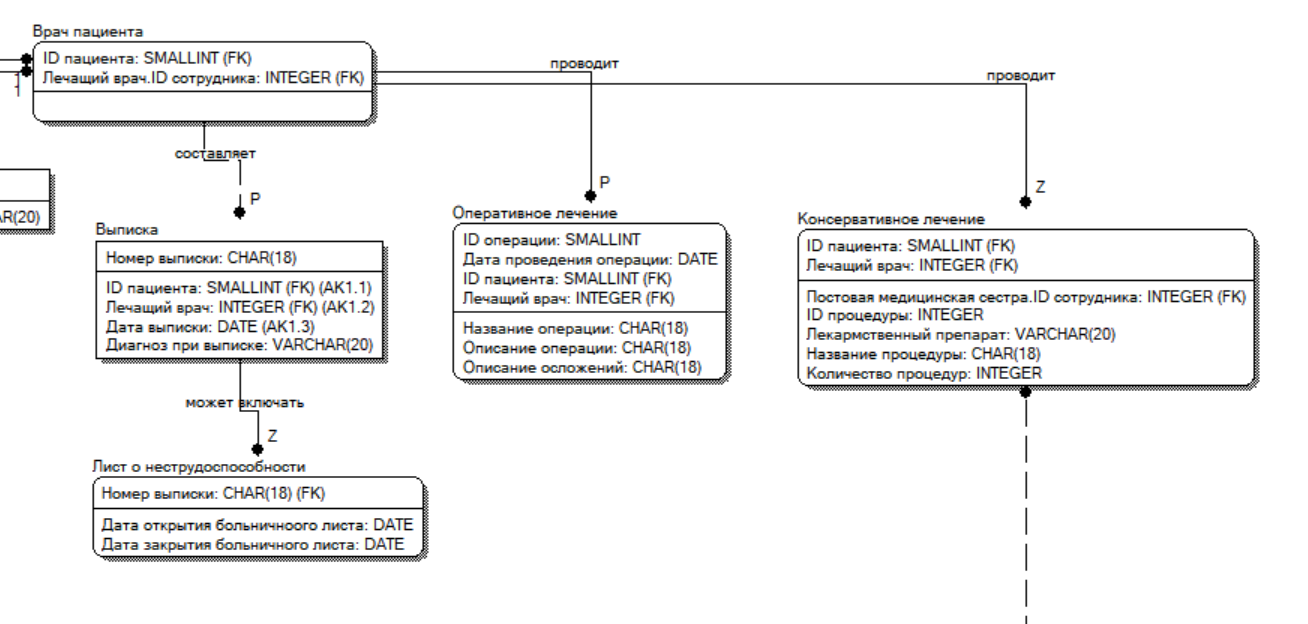
Рисунок 4 – Подсхема «Работа с пациентом»

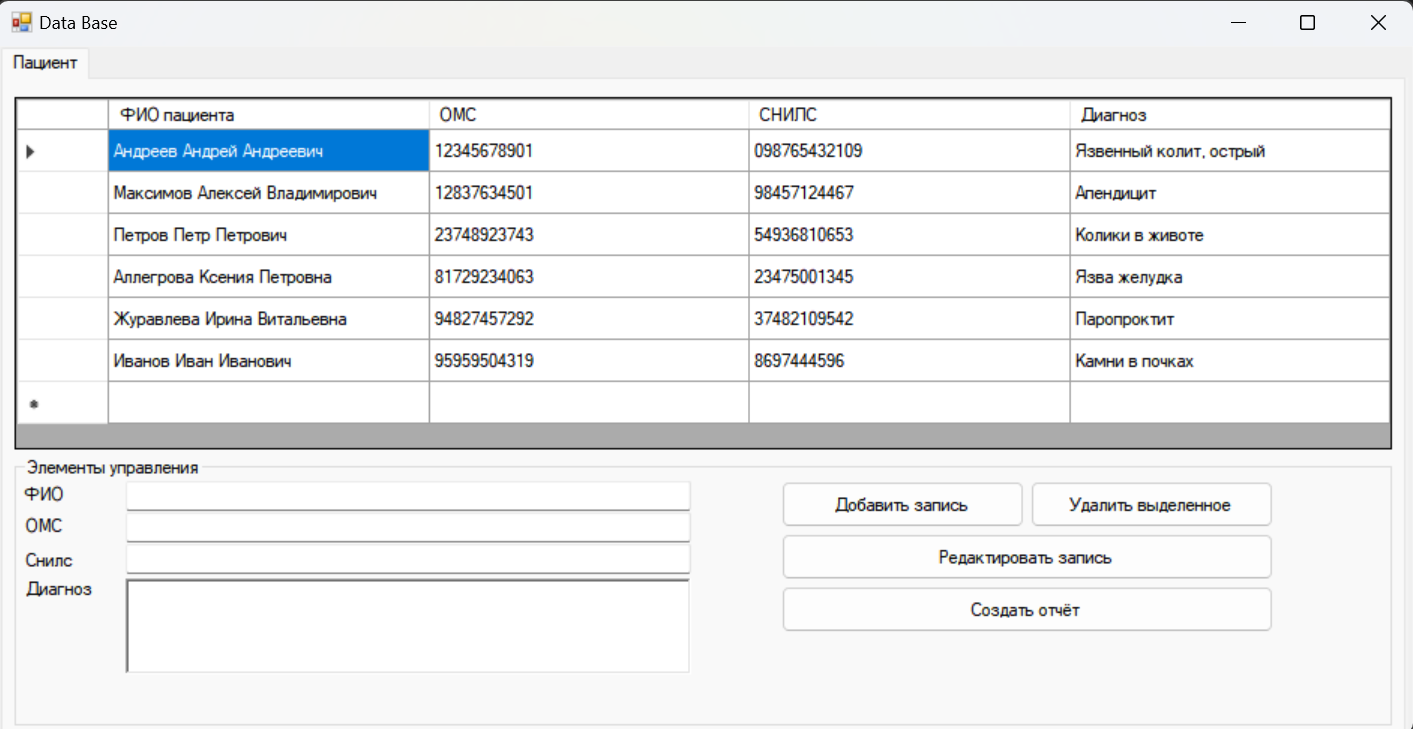
Рисунок 5 – Подсхема «Работа с сотрудниками»

Рисунок 6 – Подсхема «Работа с данными о пациенте и назначенным ему леччением»

Обеспечение ограничения целостности (подумать над ограничениями {логические})

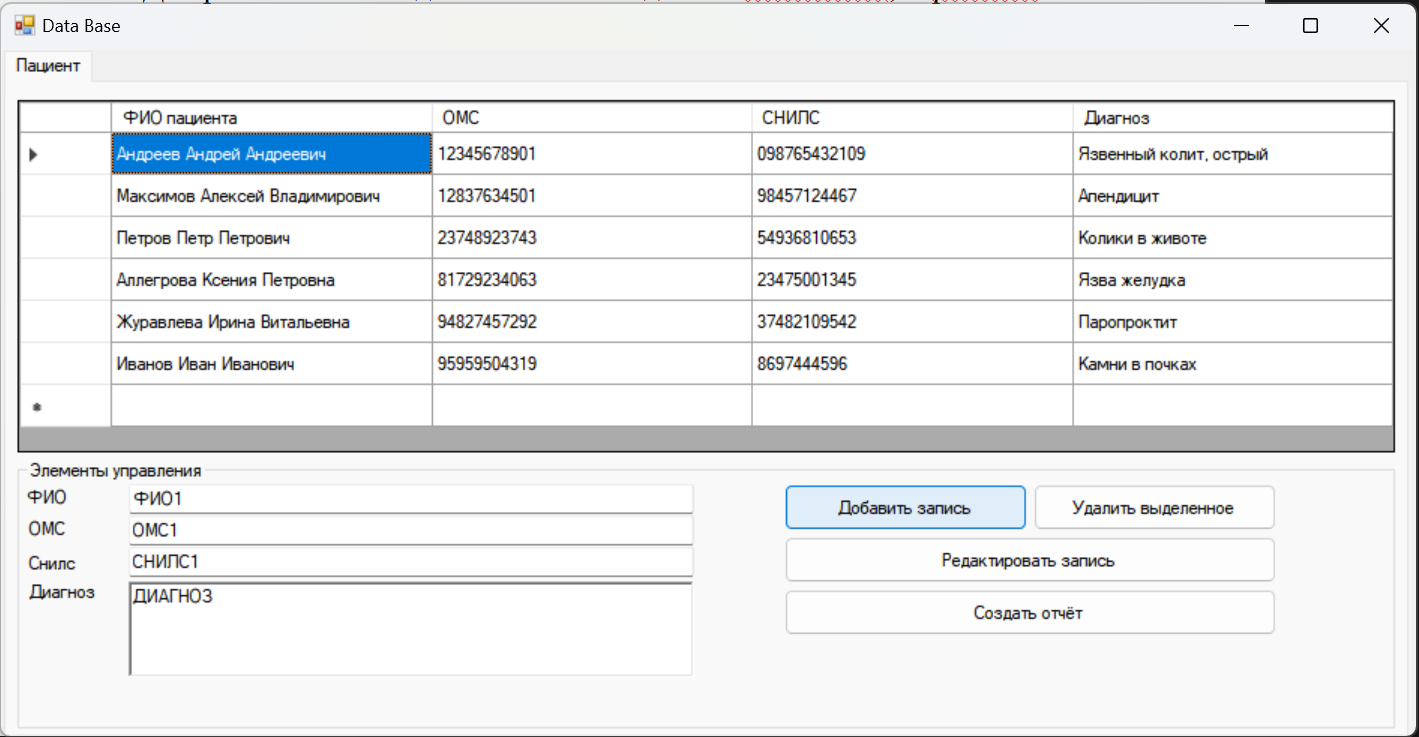
Обеспечение ссылочной целоснтости (рестрикт и каскейд)

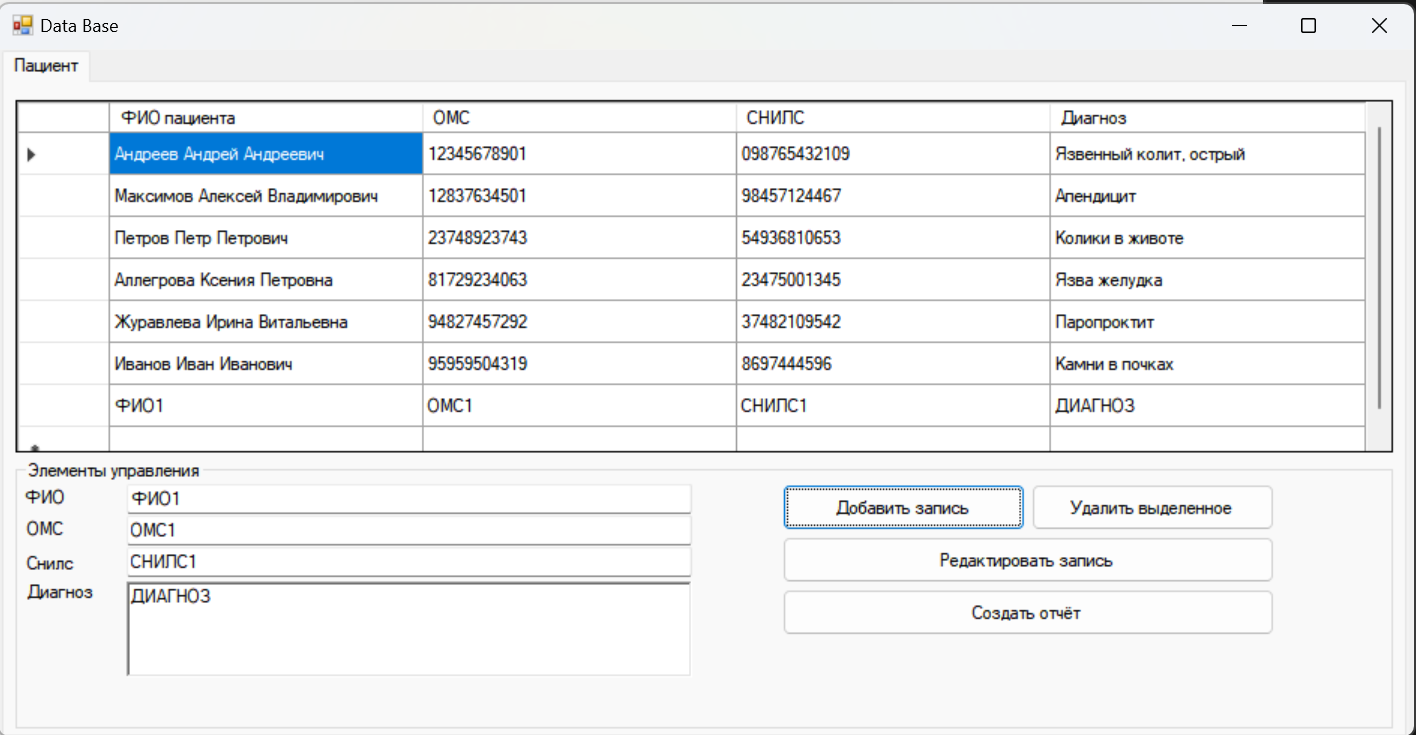
**5. Реализация работы со справочником «Пациент» в АИС учёта пациентов клинической железнодорожной больницы города Самары.**

Рисунок 7 — Основное окно программы

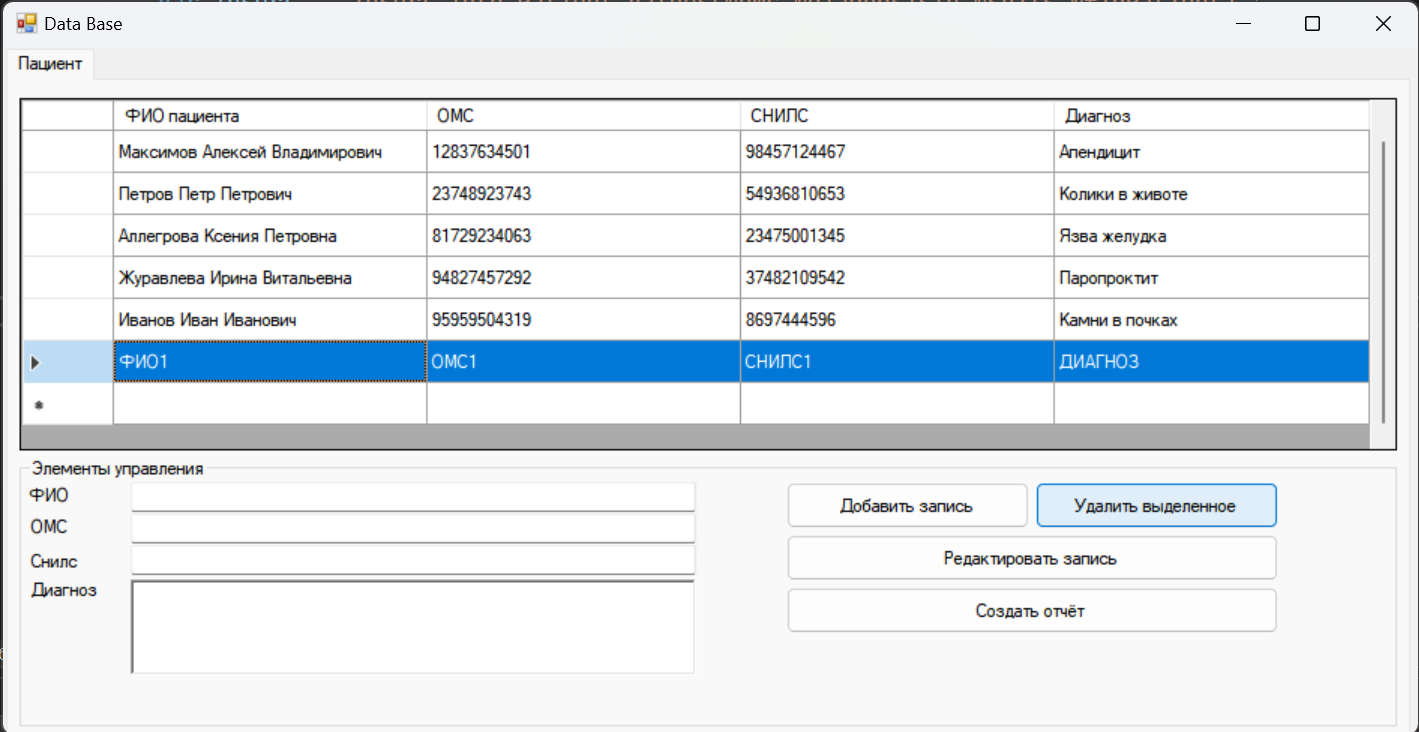
Для работы с базой данных были созданы 3 текстбокса, 1 ричбокс а так же 4 кнопки, которые отвечают за выполнение операций ведения над базой данных

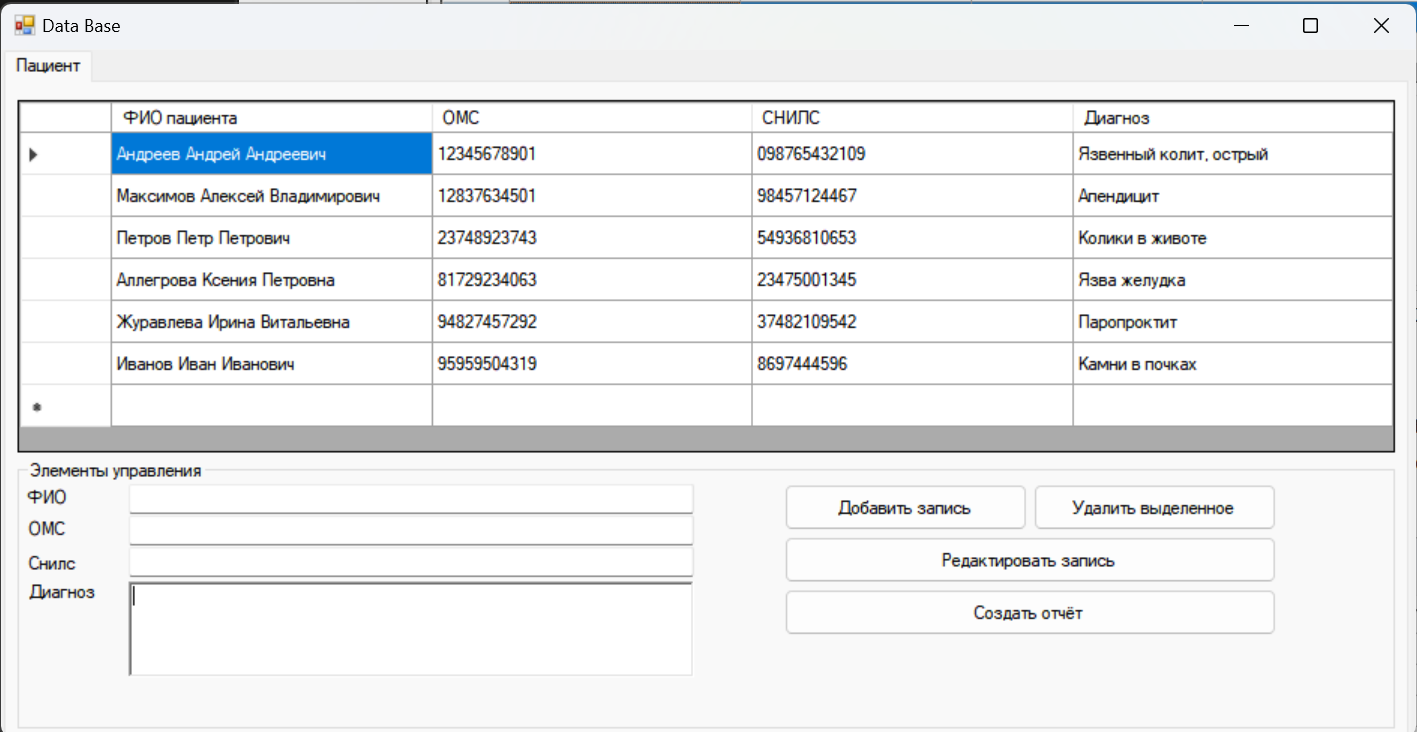
Для добавления новой записи необходимо вписать данные о пациенте в пустые поля, и нажать на кнопку «Добавить запись»

Рисунок 8 — Окно программы до нажатия на копку «Добавить запись»

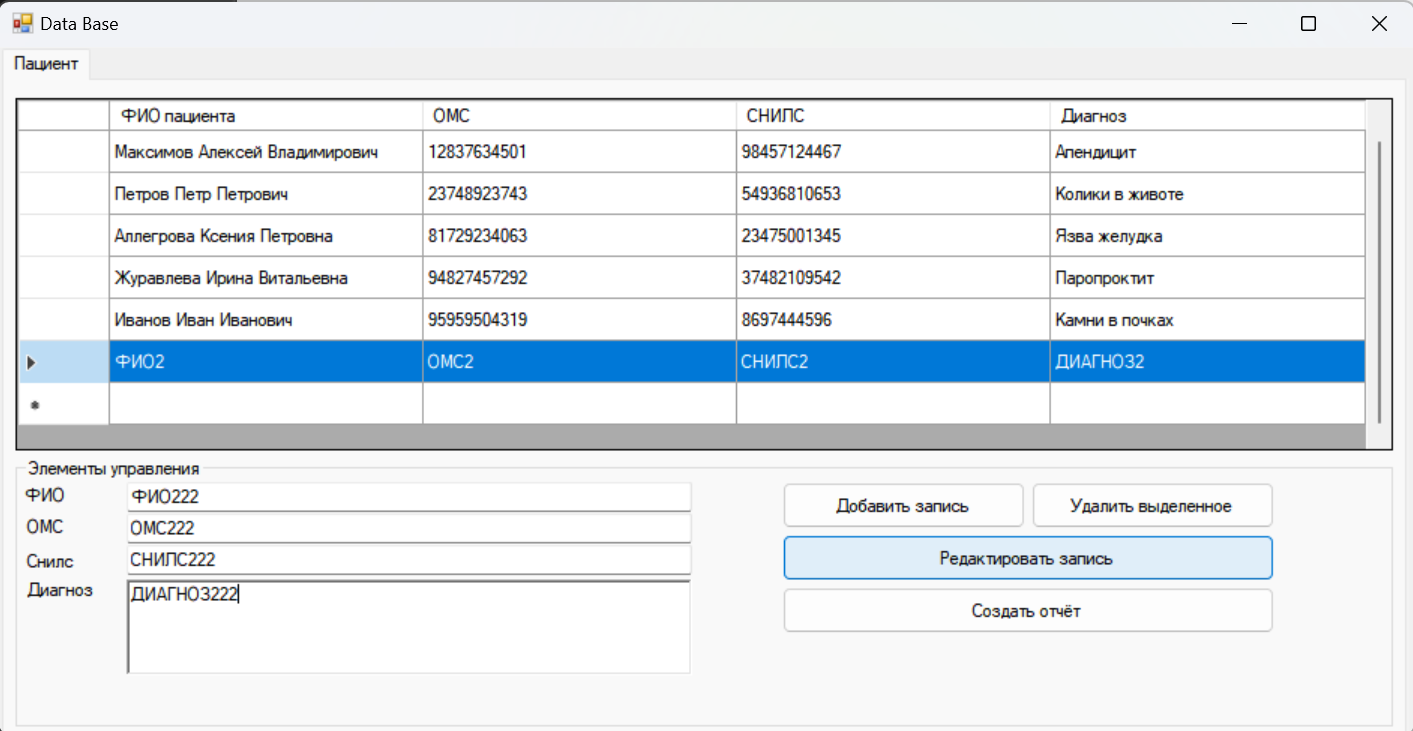
Рисунок 9 — Окно программы после нажатия на кнопку «Добавить запись»

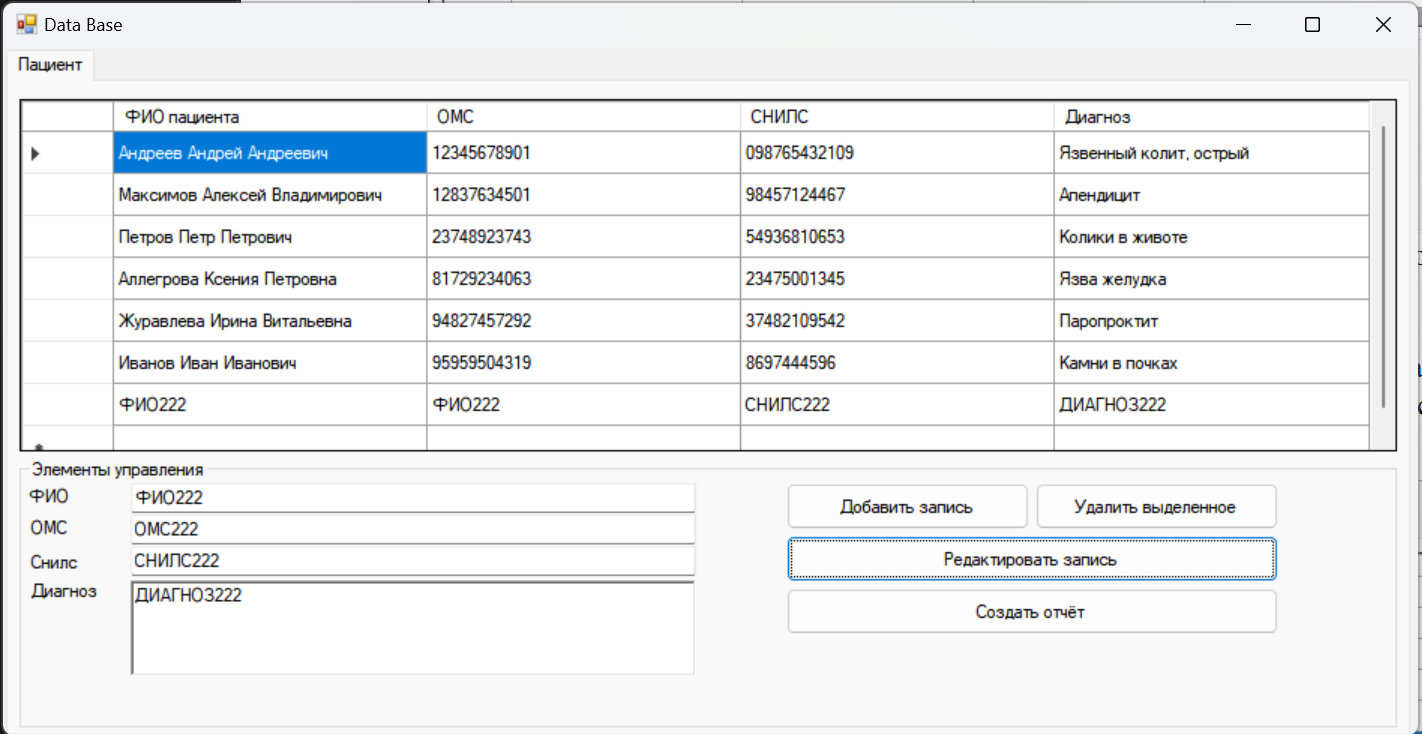
Для удаления данных о пациентах из базы данных была добавлена кнопка «Удалить выделенное»

Рисунок 9 — Окно программы до нажатия на кнопку «Удалить выделенное»

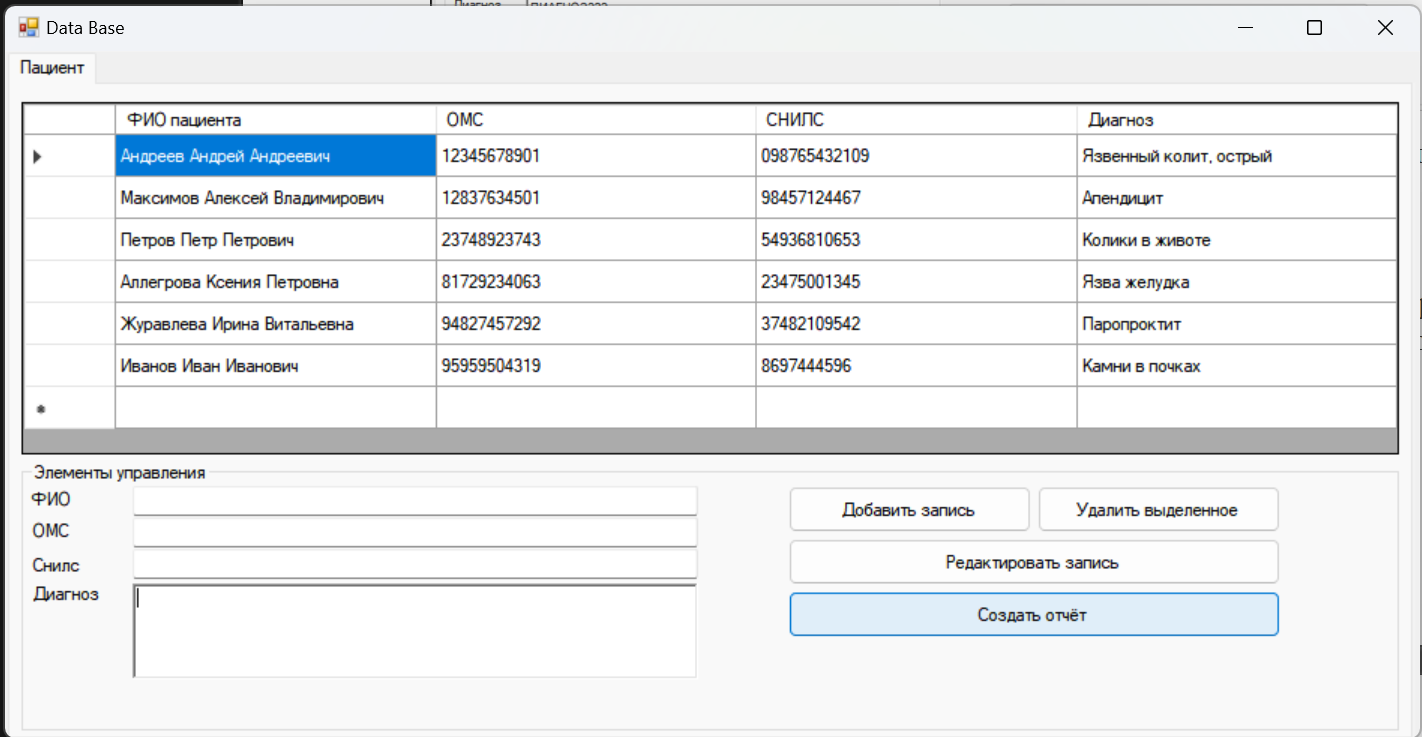
Рисунок 10 — Окно программы после нажатия на кнопку «Удалить выделенное»

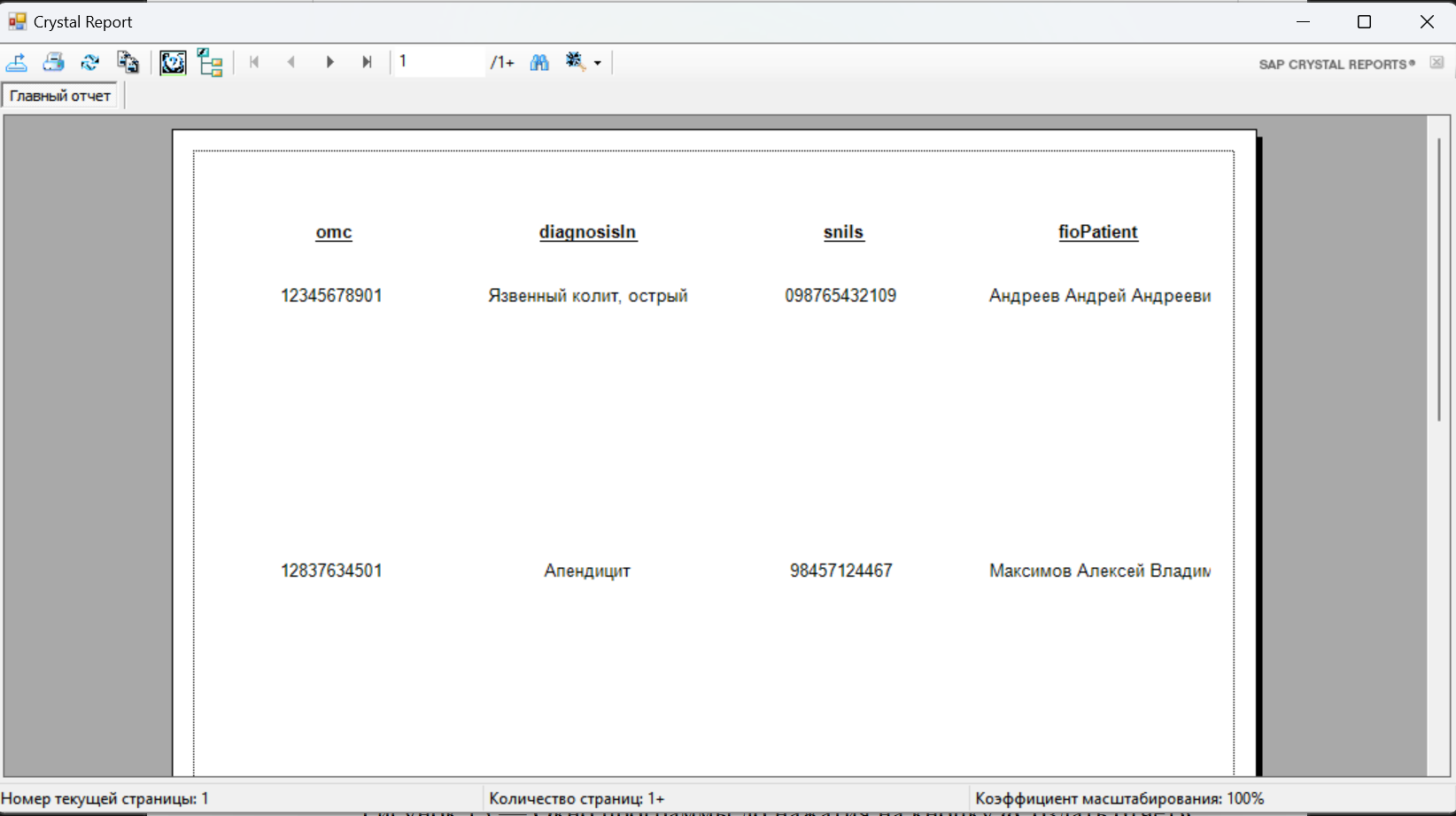
Так же в случае если данные о пациенте были заполнены неверно или каким либо образом поменялись, была добавлена функция редактирования записи о пациенте, для этого была добавлена кнопка «Редактировать запись»

Рисунок 11 — Окно программы до нажатия на кнопку «Редактировать запись»

Рисунок 12 — Окно программы после нажатия на кнопку «Редактировать запись»

Так же была реализована функция создания отчётов с помощью Crystal report, для того чтобы программа создала отчёт, необходимо нажать на кнопку «Создать отчёт»

Рисунок 13 — Окно программы до нажатия на кнопку «Создать отчёт»

Рисунок 14 — Дополнительное окно программы, созданное для отчёта

# ЗАКЛЮЧЕНИЕ

В процессе выполнения научно-исследовательской работы был освоен индикатор ПК-6.2 компетенции ПК-6, и решены все поставленные задачи:

- была проведена разработка информационно-логического проекта информационной системы системы учёта пациентов клинической железнодорожной больницы  
города Самары.

- было сделано описание информационно-логического проекта разрабатываемой информационной системы системы учёта пациентов клинической железнодорожной больницы  
города Самары.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**