

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ И ЦИФРОВЫХ ТЕХНОЛОГИЙ
КАФЕДРА ИНФОРМАЦИОННЫХ И РОБОТОТЕХНИЧЕСКИХ СИСТЕМ

**АВТОМАТИЗИРОВАННЫЙ МОНИТОРИНГ ПРОЦЕССА
ПОСТАНОВКИ ОНКОЛОГИЧЕСКОГО ДИАГНОЗА НА ОСНОВЕ
СЛАБОСВЯЗАННЫХ МЕДИЦИНСКИХ ДАННЫХ**

Выпускная квалификационная работа
обучающегося по направлению подготовки
09.03.02 Информационные системы и технологии
очной формы обучения, группы 12001509
Заворитной Анастасии Мирославовны

Научный руководитель
проф.
Маторин С.И.

БЕЛГОРОД 2019

РЕФЕРАТ

Автоматизированный мониторинг процесса постановки онкологического диагноза на основе слабосвязанных медицинских данных. – Заворитная Анастасия Мирославовна, выпускная квалификационная работа бакалавра Белгород, Белгородский государственный национальный исследовательский университет (НИУ «БелГУ»), количество страниц 60, включая приложения 80, количество рисунков 22, количество таблиц 5, количество использованных источников 26.

КЛЮЧЕВЫЕ СЛОВА: автоматизированный мониторинг, центр обработки данных (ЦОД), лечебно-профилактическое учреждение (ЛПУ), онкологическое заболевание, пациент, медицинская информационная система (МИС).

ОБЪЕКТ ИССЛЕДОВАНИЯ: процесс постановки онкологического диагноза.

ПРЕДМЕТ ИССЛЕДОВАНИЯ: автоматизация учета пациентов и сроков оказания медицинской помощи по профилю «онкология».

ЦЕЛЬ РАБОТЫ: повышение производительности труда медицинского персонала и снижение сроков постановки диагноза и начала специализированного лечения за счет внедрения автоматизированного мониторинга процесса постановки онкологического диагноза.

ЗАДАЧИ ИССЛЕДОВАНИЯ: анализ предприятия; анализ существующих систем; выбор проектных решений; разработка автоматизированного мониторинга процесса постановки онкологического диагноза; обоснование целесообразности разработки.

ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ: В результате работы был спроектирован и реализован автоматизированный мониторинг процесса постановки онкологического диагноза на основе слабосвязанных медицинских данных.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	5
1 Аналитическая часть.....	7
1.1 Техничко-экономическая характеристика предметной области.....	7
1.1.1 Характеристика предприятия	7
1.1.2 Краткая характеристика подразделения или видов его деятельности ...	8
1.2 Экономическая сущность задачи	10
1.3 Обоснование необходимости и цели использования вычислительной техники для решения задачи.....	13
1.4 Постановка задачи	16
1.4.1 Цель и назначение автоматизированного варианта решения задачи ...	16
1.4.2 Общая характеристика организации решения задачи на ЭВМ.....	17
1.5 Анализ существующих разработок и обоснование выбора технологии проектирования	17
2 Обоснование проектных решений.....	20
2.1 Обоснование проектных решений по техническому обеспечению (ТО)...	20
2.2 Обоснование проектных решений по информационному обеспечению (ИО)	21
2.3 Обоснование проектных решений по программному обеспечению (ПО). 24	
2.4 Обоснование проектных решений по технологическому обеспечению	26
2.5 Обоснование выбора программных средств	26
3 Проектная часть.....	31
3.1 Информационное обеспечение задачи.....	31
3.1.1 Информационная модель и ее описание	31

3.1.2 Характеристика базы данных	35
3.1.3 Характеристика результатной информации	39
3.2 Программное обеспечение задачи.....	42
3.3 Описание контрольного примера реализации проекта	50
3.4 Целесообразность и экономическая эффективность.....	54
ЗАКЛЮЧЕНИЕ	56
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	58
ПРИЛОЖЕНИЕ А	61
ПРИЛОЖЕНИЕ Б.....	72

ВВЕДЕНИЕ

Объем цифровой информации и ее оборот в различных сферах деятельности человека значительно вырос. И объем этой информации не уменьшается, а только увеличивается с каждым днем. Именно поэтому обработка и систематизация информации позволяет ориентироваться в большом потоке информации, производить поиск и получать нужные данные за кратчайшие промежутки времени, с наибольшим эффектом использовать результаты, полученные из разных информационных источников.

Анализ работы медицинских организаций показал, что их эффективная работа практически полностью зависит от уровня оснащенности информационными средствами. Чтобы на должном и высоком уровне решать поставленные перед ними задачи, необходим соответствующий инструментарий и методология внедрения данного информационного обеспечения, в связи с чем высока роль персонального компьютера как инструмента, обеспечивающего эффективность работы с данными, и современных технологий хранения и поиска информации. Именно этим определяется актуальность выпускной квалификационной работы [1].

Объектом исследования является процесс постановки онкологического диагноза.

Предмет исследования – автоматизация учета пациентов и сроков оказания медицинской помощи по профилю «онкология».

Целью работы является повышение производительности труда медицинского персонала и снижение сроков постановки диагноза и начала специализированного лечения за счет внедрения автоматизированного мониторинга процесса постановки онкологического диагноза.

Задачами выпускной квалификационной работы являются:

- анализ деятельности рассматриваемого предприятия;
- обоснование необходимости разработки информационного обеспечения;

- обоснование проектных решений по программному, техническому и информационному обеспечению;

- разработка автоматизированного мониторинга;

- расчет экономических показателей проекта.

Выпускная квалификационная работа состоит из трех разделов.

В первом разделе проведен анализ предметной области изучена экономическая сущность задачи, обоснована постановка задачи, ее цель и назначение.

Во втором разделе произведен выбор по техническому, информационному и программному обеспечению.

В третьем разделе разработано и описано информационное и программное обеспечение, а также проведен обзор экономической составляющей автоматизированной системы.

Выпускная квалификационная работа содержит: 60 страниц, 22 рисунка, 5 таблиц.

1 Аналитическая часть

1.1 Технико-экономическая характеристика предметной области

1.1.1 Характеристика предприятия

Общество с ограниченной ответственностью «СофтТраст» (ООО «СофтТраст») занимается разработкой компьютерного программного обеспечения, оказывает консультационные услуги в данной области и другие сопутствующие услуги.

Основное направление - комплексная автоматизация деятельности лечебно-профилактических и аптечных учреждений, приложения для пациентов и органов управления здравоохранением. На рисунке 1.1 приведена схема организационной структуры ООО «СофтТраст».

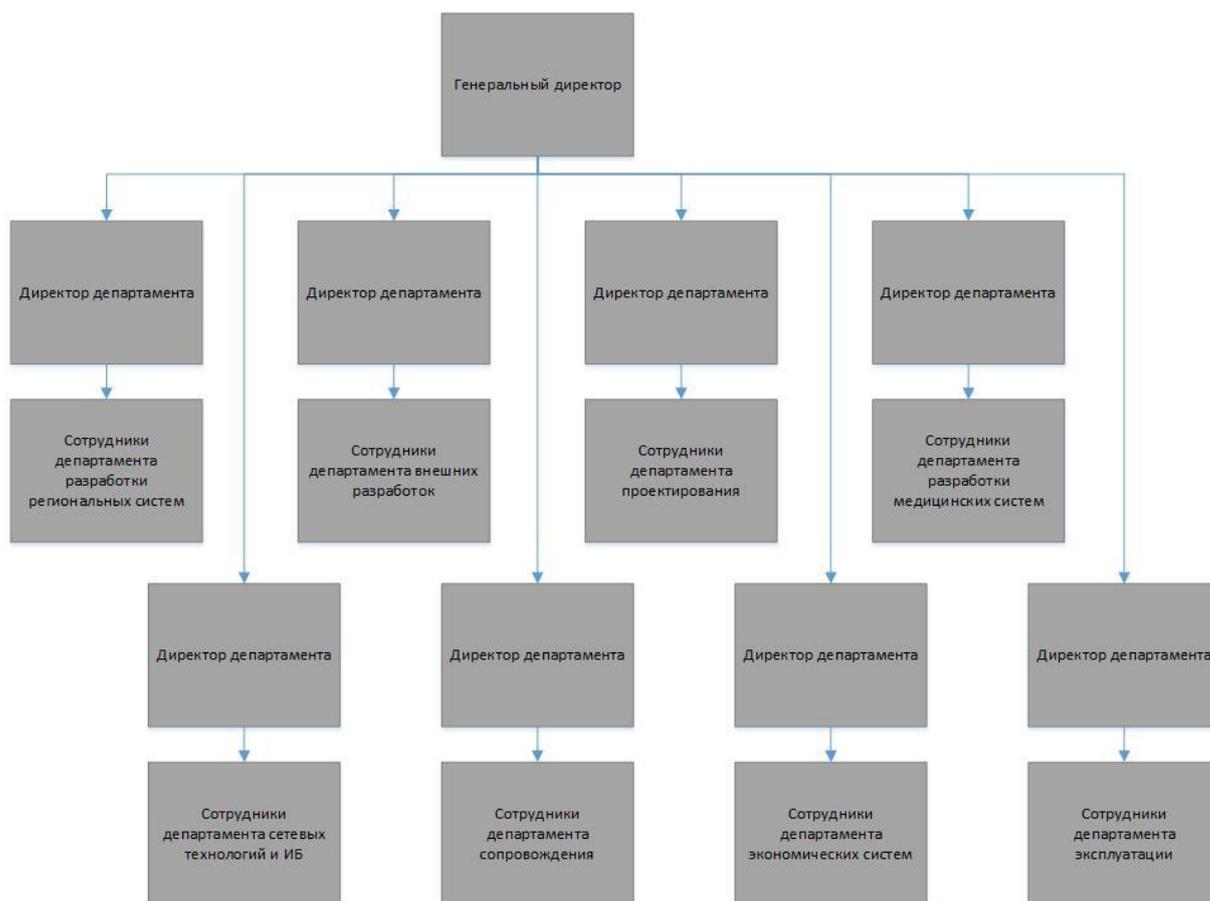


Рисунок 1.1 – Организационная структура управления

В данной структуре реализуется принцип линейного построения структурных подразделений и распределения функций управления между ними, а также принцип единоначалия [2].

Высший (институциональный) уровень. Деятельность руководителя данного уровня обусловлена стратегиями и целями развития организации в целом.

Средний (управленческий) уровень. Данный уровень объединяет руководителей среднего звена и их аппарат. Задачи, вытекающие из функциональной специфики, решают менеджеры этого уровня.

Низший (производственно-технический) уровень. Этот уровень объединяет руководителей нижнего уровня, находящихся непосредственно над исполнителями. Руководители низшего уровня ответственны за работу с людьми, а также обеспечение всеми видами ресурсов производственного процесса.

Всю власть, при такой структуре управления, берет на себя руководитель, возглавляющий коллектив. При подготовке соответствующих решений, планов и программ, а также решении конкретных вопросов ему помогает специальный аппарат, который состоит из функциональных подразделений. Свои решения они реализовывают или через главного руководителя, или через соответствующих руководителей служб.

1.1.2 Краткая характеристика подразделения или видов его деятельности

Департамент разработки региональных систем занимается разработкой основных программных модулей ТрастМед.

ТрастМед — программный комплекс для построения единой государственной системы здравоохранения региона полностью автоматизирует медицинское учреждение: поликлинику, стационар, стоматологическую клинику, больничную аптеку. Основные модули ТрастМед:

а) ТрастМед для пациента (личный кабинет пациента) - веб-сервис для записи на приём к врачу, ведения журнала здоровья, прохождения диспансеризаций и медицинских осмотров.

б) ТрастМед для медицинской организации.

1) ТрастМед: МИС - комплексная автоматизация деятельности лечебно-профилактического учреждения.

2) ТрастМед: Поликлиника - комплексная автоматизация деятельности персонала амбулаторно-поликлинического подразделения.

3) ТрастМед: Стационар - комплексная автоматизация деятельности персонала стационара.

4) ТрастМед: Шаблон - веб-сервис для создания шаблонов медицинских записей

в) ТрастМед для фармации.

1) ТрастМед: Больничная аптека - автоматизация назначения лекарственных препаратов пациентам с последующим автоматическим персонифицированным списанием со складов отделений и больничной аптеки.

2) ТрастМед: РАС - автоматизация ведения предметно-количественного учета лекарственных средств на региональном аптечном складе.

3) ТрастМед: Лекарственное обеспечение - автоматизация деятельности учреждений здравоохранения субъекта Российской Федерации при формировании заявок на лекарственные средства и изделия медицинского назначения.

г) ТрастМед органам управления здравоохранения.

1) ТрастМед: 263 - информационная поддержка застрахованных граждан.

2) ТрастМед: ЦОД SaaS - программный комплекс для централизованной обработки данных и информационного взаимодействия участников программ Обязательного Медицинского Страхования (ОМС) и Обеспечения Необходимыми Лекарственными Средствами (ОНЛС).

3) Мониторинг деятельности учреждений - формирование отчётов и рейтингов медицинских организаций, применяющих ТрастМед.

4) ТрастМед: МЭК - обеспечение медико-экономического контроля персонифицированного учета медицинской помощи, оказанной застрахованным лицам.

5) ТрастМед: ИЭМК - централизованный архив медицинских документов в региональном сегменте Единой Государственной Информационной Системы в сфере Здравоохранения (ЕГИСЗ).

Отдел мониторингов занимается формированием мониторингов деятельности учреждений и программным комплексом для централизованной обработки данных. Отдел взаимодействует с Министерством здравоохранения Российской Федерации и сотрудниками Медицинского информационно-аналитического центра (МИАЦ) и выполняет их поручения, связанные с обработкой информации, подсчета статистики и формированием отчетов.

1.2 Экономическая сущность задачи

Приказом Министерства Здравоохранения от 04.07.2017 г. №379н внесены изменения в порядок оказания медицинской помощи населению по профилю «онкология», утвержденный приказом ведомства от 15 ноября 2012 г. №915н. и установлены предельные сроки диагностики онкологических заболеваний и оказания специализированной медицинской помощи больным.

В данной выпускной квалификационной работе объектом исследования является процесс постановки онкологического диагноза. Входящей в данный объект задачей является формирование автоматизированного мониторинга. Данная задача относится к классу анализа деятельности медицинских организаций и необходима для выявления лечебно-профилактических учреждений и врачей, не соблюдающих сроки постановки онкологического диагноза. Результаты решения данной задачи являются основой для принятия стратегических управленческих решений. Поэтому формирование мониторинга является важной и неотъемлемой частью процесса постановки онкологического диагноза.

Основными бизнес-процессами при постановке онкологического диагноза являются:

- прием у врача первичного звена;
- выдача направления на консультацию к врачу-онкологу;
- первичный прием у врача-онколога;
- выдача направления на исследование;
- проведение исследования;
- повторный прием у врача-онколога;
- выдача направления на госпитализацию;
- фактическая госпитализация.

Выделим следующие задачи автоматизации, в каждой задаче идет учет пациентов с направлением на консультацию от врача первичного звена к врачу-онкологу:

- учет пациентов, находящихся в процессе постановки диагноза;
- учет времени нахождения больного в процессе;
- учет пациентов с нарушением сроков ожидания записи на прием к врачу-онкологу;
- учет пациентов неявившихся на прием к врачу-онкологу;
- учет пациентов с нарушением сроков ожидания записи на исследование;
- учет пациентов неявившихся на исследование;
- учет пациентов с нарушением сроков ожидания результатов исследования;
- учет пациентов с уточненным диагнозом, не имеющих направления на госпитализацию;
- учет пациентов с направлением на госпитализацию, не имеющие фактической госпитализации;
- учет пациентов, по которым не определено следующее мероприятие.

Отметим, что все задачи связаны между собой, а процессы, существующие на предприятии, не могут относиться только к одной из задач [3].

Отдельное внимание следует уделить одной из важных задач автоматизации, а именно: отслеживанию сроков оказания медицинской помощи больным по профилю «онкология».

Сроки, на которые стоит обращать внимание, определяются приказом Министерства здравоохранения РФ от 15 ноября 2012 г. N 915н "Об утверждении Порядка оказания медицинской помощи населению по профилю "онкология".

При подозрении или выявлении у больного онкологического заболевания врачи-терапевты, врачи-терапевты участковые, врачи общей практики (семейные врачи), врачи-специалисты, средние медицинские работники в установленном порядке направляют больного на консультацию в первичный онкологический кабинет или первичное онкологическое отделение медицинской организации для оказания ему первичной специализированной медико-санитарной помощи. Консультация в первичном онкологическом кабинете или первичном онкологическом отделении медицинской организации должна быть проведена не позднее 5 рабочих дней с даты выдачи направления на консультацию.

Врач-онколог первичного онкологического кабинета или первичного онкологического отделения в течение одного дня с момента установления предварительного диагноза злокачественного новообразования организует проведение лабораторного исследования. В случае невозможности взятия в медицинской организации, в составе которой организован первичный онкологический кабинет или первичное онкологическое отделение, биопсийного (операционного) материала, проведения иных диагностических исследований пациент направляется лечащим врачом в онкологический диспансер или в медицинскую организацию, оказывающую медицинскую помощь больным с онкологическими заболеваниями. Срок выполнения патологоанатомических исследований, необходимых для гистологической верификации злокачественного новообразования, не должен превышать 15 рабочих дней с даты поступления биопсийного (операционного) материала в патологоанатомическое бюро (отделение).

Врач-онколог первичного онкологического кабинета или первичного онкологического отделения направляет больного в онкологический диспансер или в медицинские организации, оказывающие медицинскую помощь больным с онкологическими заболеваниями, для уточнения диагноза (в случае невозможности установления диагноза, распространенности онкологического процесса и стадирования заболевания врачом-онкологом первичного онкологического кабинета или первичного онкологического отделения) и оказания специализированной, в том числе высокотехнологичной, медицинской помощи. Срок начала оказания специализированной, за исключением высокотехнологичной, медицинской помощи больным с онкологическими заболеваниями в медицинской организации, оказывающей медицинскую помощь больным с онкологическими заболеваниями, не должен превышать 10 календарных дней с даты гистологической верификации злокачественного новообразования или 15 календарных дней с даты установления предварительного диагноза злокачественного новообразования (в случае отсутствия медицинских показаний для проведения патолого-анатомических исследований в амбулаторных условиях).

1.3 Обоснование необходимости и цели использования вычислительной техники для решения задачи

Модель «КАК ЕСТЬ» - это модель уже существующего процесса или функции. Исследование процессов является обязательной частью любого проекта создания системы [4]. Построение структурно-функциональной модели позволяет выяснить, какие процессы выполняются на предприятии, а также какие используются информационные объекты при выполнении различных функций.

Структурно-функциональная модель «КАК ЕСТЬ» является начальной точкой для выявления потребностей и проблем предприятия, а также разработки проекта автоматизации бизнес-процессов [5].

На рисунке 1.2 представлена контекстная диаграмма «КАК ЕСТЬ» процесса постановки онкологического диагноза.

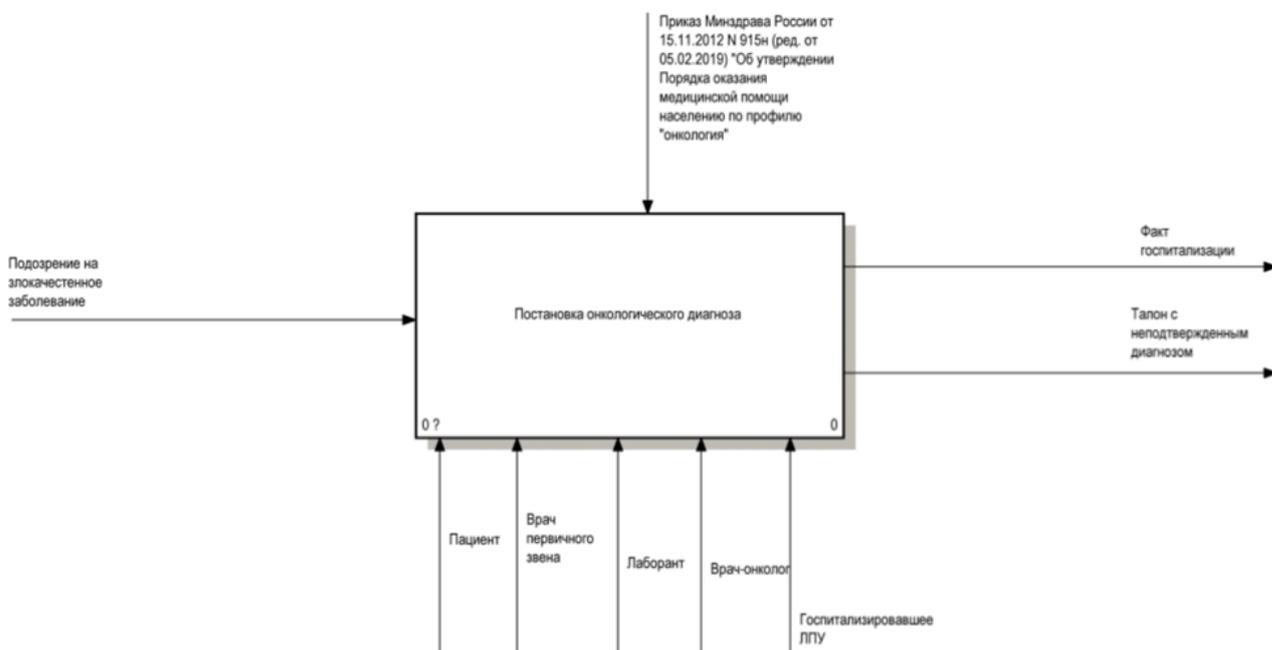


Рисунок 1.2 – Контекстная диаграмма «Постановка онкологического диагноза»

На рисунке 1.3 представлена декомпозиция контекстной диаграммы. Она включает в себя прием у врача первичного звена, постановку диагноза и госпитализацию.

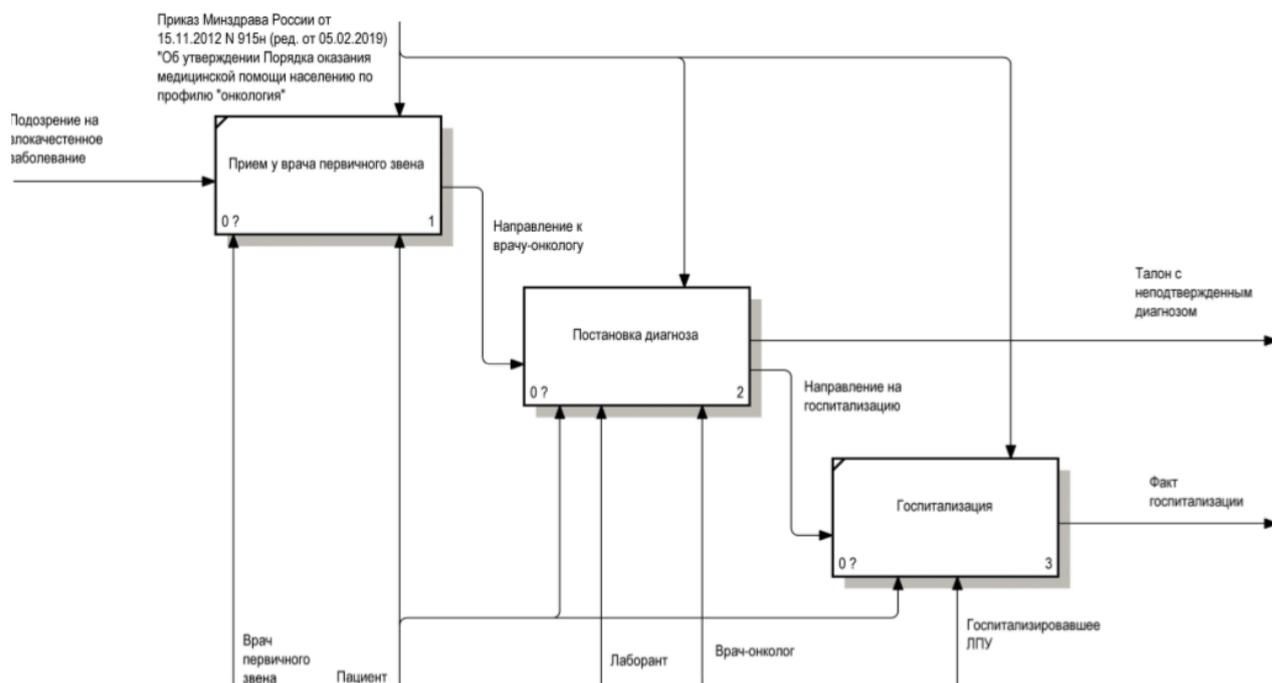


Рисунок 1.3 – Декомпозиция контекстной диаграммы

На рисунке 1.4 показана декомпозиция блока «Постановка диагноза». Постановка диагноза в свою очередь включает в себя первичный прием у врача онколога, проведение лабораторного исследования и повторный прием у врача онколога.

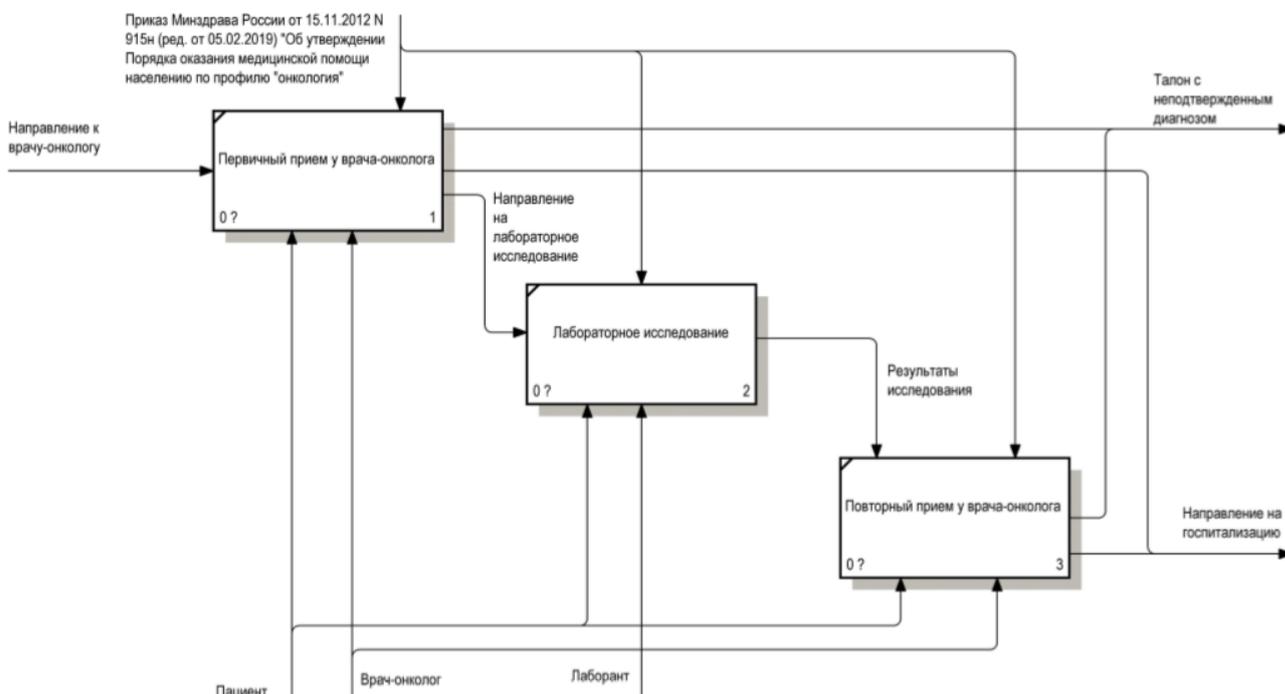


Рисунок 1.4 – Декомпозиция «Постановка диагноза»

Механизмами являются пациент, врач первичного звена, врач-онколог, лаборант, лечебно-профилактическое учреждение госпитализации (ЛПУ). Выходом из процесса постановки диагноза является факт госпитализации либо талон с неподтвержденным диагнозом от врача-онколога.

Все процессы регламентированы Приказом Министерства здравоохранения Российской Федерации от 15 ноября 2012 г. № 915н "Об утверждении Порядка оказания медицинской помощи населению по профилю "онкология".

Анализ поставленной задачи приводит к выводу, что ее решение традиционными средствами учета с использованием бумажных документов возможно, но неэффективно. Прежде всего, такой вывод следует из анализа количества операций при ведении пациента. Проблема автоматизации возникла в связи с необходимостью обработки и анализа больших объемов информации, а

также решения задач управления, с чем традиционным способом, путем бумажного документооборота, справиться уже невозможно.

Основной процесс постановка онкологического диагноза, и автоматизация должна быть средством его оптимизации, инструментом для оказания качественной медицинской помощи. Использование вычислительной техники поднимает эффективность работы врачей на принципиально новый уровень, предоставляя функции, ранее недоступные.

1.4 Постановка задачи

1.4.1 Цель и назначение автоматизированного варианта решения задачи

Целью разработки автоматизированного мониторинга процесса постановки онкологического диагноза с точки зрения получения прямого эффекта от автоматизации решения рассматриваемого комплекса задач может быть:

- повышение производительности труда медицинского персонала;
- снижение сроков постановки диагноза;
- снижение сроков начала специализированного лечения;
- снижение смертности от злокачественных заболеваний;
- повышение доли заболеваний, выявленных на ранних стадиях;
- удобное ведение отчетности.

Назначением автоматизированного варианта решения задачи является:

- автоматизация учета больных злокачественными заболеваниями;
- автоматизация информационного обмена между медицинскими организациями;
- возможность вывода отчетных форм на печать.

Основными функциями мониторинга для учета больных по профилю «онкология» являются:

- учет пациентов с онкологическими заболеваниями;

- учет соблюдения установленных сроков оказания медицинской помощи;
- ведение отчетности.

Автоматизированный мониторинг представляет собой совокупность алгоритмов обработки данных и эффективного отображения необходимой информации на отчетных формах.

Врач в процессе ведения конкретного пациента может отслеживать его этапы и сроки постановки диагноза. Медицинская организация может видеть общую картину работы медицинского персонала. Министерство здравоохранения получит возможность наблюдать динамику по региону в целом.

Таким образом, автоматизированный мониторинг сможет повлиять на улучшение качества и понижение сроков обслуживания пациентов.

1.4.2 Общая характеристика организации решения задачи на ЭВМ

Задача автоматизации мониторинга процесса постановки онкологического диагноза состоит из следующих основных подзадач:

- изучение структуры базы данных;
- создание временных таблиц, для быстрого доступа к информации;
- наполнение и настройка автоматизированного наполнения временных таблиц;
- формирование отчетных форм;
- настройка вывода информации в отчетные формы;
- размещение отчетных форм на ресурсе.

1.5 Анализ существующих разработок и обоснование выбора технологии проектирования

Для автоматизации процесса постановки онкологического диагноза существуют готовые информационные системы, предлагающие различный функционал. Рассмотрим самые крупные из них.

Программа «Канцер-регистр бФВ» предназначена для регистрации и ведения учета больных злокачественными новообразованиями в территориальных учреждениях Российской Федерации.

Программа «Канцер-регистр бФВ» позволяет отправлять обновления баз данных территориальных раковых регистров в базу данных Федерального ракового регистра РФ.

Полностью поддерживается приказ №135 от 19 апреля 1999 г. «О совершенствовании системы Государственного ракового регистра» Министерства здравоохранения Российской Федерации, который регламентирует регистрацию больных злокачественными новообразованиями при формировании Государственного ракового регистра. Программа разработана ФГУ МНИОИ им. П.А.Герцена.

Еще одна информационная система «Арте-ONCO». Специализированный программный комплекс, автоматизирующий деятельность районных онкологов и онкологических учреждений различного уровня (город, край, область, республика) и видов деятельности (поликлиника, дневной и круглосуточный стационары).

Это госпитальная система с полноценной электронной историей болезни, амбулаторной картой и многогранным дополнительным функционалом.

Система в 2003-2005 годах была разработана на базе краевого клинического онкологического диспансера. С 2006 г. она находится в промышленной эксплуатации, постоянно обновляется в соответствии с изменениями медицинского законодательства, а также в связи с разработкой новых модулей. Кроме Краснодарского края (Краснодар и Новороссийск) имеются внедрения в Ростовской и Свердловской областях, в Алтайском крае.

Предприятие ООО «СофтТраст» занимается комплексной автоматизацией лечебно-профилактических учреждений. ТрастМед - программный комплекс, разработанный компанией ООО «СофтТраст», для построения единой государственной системы здравоохранения региона полностью автоматизирует

медицинское учреждение: поликлинику, стационар, стоматологическую клинику, больничную аптеку.

В системе уже имеется возможность ведения пациента по профилю «онкология». И так как, ТрастМед уже внедрен на регионе, то нет необходимости в дополнительном программном обеспечении.

В разработанном комплексе ТрастМед отсутствует лишь мониторинг процесса постановки онкологического диагноза, для учета соблюдения сроков ведения больного.

Преимущество ТрастМед заключается еще в том, что нет необходимости в подключении дополнительных модулей от разных разработчиков, вся система здравоохранения региона ведется одним разработчиком и одной информационной системой. Плюсом является и то, что не требуется дополнительное обучение врачей по работе с новым программным продуктом.

Выводы по первому разделу

В первом разделе дается технико-экономическая характеристика предметной области, выполняется обоснование необходимости и цели создания автоматизированного мониторинга, осуществляется постановка задачи. Также был произведен анализ информационных систем конкурентов.

2 Обоснование проектных решений

2.1 Обоснование проектных решений по техническому обеспечению (ТО)

Техническое обеспечение - совокупность технических средств, компьютерной техники, средств передачи информации, используемых в автоматизированных системах управления и в информационных системах.

Для формирования автоматизированного мониторинга необходимо наличие Центра Обработки Данных (ЦОД), на который ежедневно поступает вся медицинская история со всех медицинских организаций региона с помощью протоколов обмена данными. По структуре ЦОД и локальная база данных каждого ЛПУ одинаковы.

Так как объем результативных данных является достаточно большим, то оперативная память и объем жесткого диска сервера должны быть достаточны, чтобы заархивировать значительные объемы информации, также, процессор должен быть достаточно быстродействующим. Необходимые элементы для технического обеспечения автоматизации функций планирования приведены ниже. Сервер должен быть быстродействующим, а также обладать высокой вычислительной мощностью.

Наиболее подходящими характеристиками обладает четырехпроцессорный сервер Flagman QD420. Сервер STSS Flagman QD420 построен на базе четырех высокопроизводительных процессоров AMD® Opteron® с использованием технологии Dual/Quad Core. Данный сервер обладает высокой вычислительной мощностью. Flagman QD420 может применяться как сервер тяжелых баз данных. Обладая высокими вычислительными возможностями, может использоваться в качестве сервера SQL с числом пользователей свыше тысячи.

Надежность обеспечивается благодаря использованию серверных компонентов, буферизированной памяти с коррекцией ошибок (ECC Registered),

высококачественного отказоустойчивого блока питания с горячей заменой, жестких дисков с возможностью горячей замены, аппаратных RAID-контроллеров, отказоустойчивой системы охлаждения и встроенным средствам диагностики и мониторинга.

В сервере применяется дисковая подсистема SAS для обеспечения максимальной надежности и производительности на приложениях типа СУБД. Сервер имеет шесть дисковых отсеков для установки жестких дисков с горячей заменой.

Основные характеристики сервера Flagman QD420:

- четыре двух-/четырёх- ядерных процессора AMD® Opteron® 8000-series Dual/Quad Core AMD64;
- до 128Gb оперативной памяти DDR-2 PC2-5300 ECC Registered;
- дисковая подсистема SAS/SATA onBoard RAID 0,1,10 (SAS RAID-5,6 - опционально);
- два интегрированных сетевых адаптера 10/100/1000Mbit Ethernet;
- система управления сервером IPMI 2.0 Server Management (опционально);
- корпус 2U Rackmount.

Других технических элементов для технического обеспечения задачи автоматизации не требуется.

2.2 Обоснование проектных решений по информационному обеспечению (ИО)

Информационное обеспечение - это совокупность средств и методов построения информационной базы.

Существует три способа организации информационной базы (ИБ): файловая организация ИБ; интегрированная ИБ, смешанная организация ИБ.

Под файловой организацией ИБ понимается локальное размещение базы на компьютере, доступ к которому других пользователей осуществляется

стандартными методами ОС для обмена данными по сети, что уменьшает скорость обработки данных в локальной базе.

Под смешанной организацией ИБ подразумевается распределённая база данных, хранящаяся на нескольких серверах и реплицирующая изменения в каждой из них по расписанию, данная структура ИБ используется в системах класса ERP для работы в одной ИБ территориально удалённым офисам одновременно.

Интегрированный способ организации ИБ представляет собой совокупность взаимосвязанных и хранящихся вместе данных при такой минимальной избыточности, которая допускает их использование оптимальным образом для любых приложений и при этом обеспечивается независимость данных от программы, а для актуализации данных используется общий способ управления.

В данном проекте наиболее целесообразной организацией информационной базы можно считать смешанную организацию ИБ. И оптимальным выбором будет использование СУБД вместо файлового хранения базы данных.

Существует три модели логической структуры базы данных (по способу установления связей между данными): иерархическая, сетевая и реляционная.

В иерархической модели каждой информационной единице (сегменту), кроме корневого, соответствует один исходный сегмент и между исходным и порожденным сегментом устанавливается только одна связь. В иерархических моделях экземпляру исходного сегмента соответствует в общем случае какое-то число экземпляров порожденного сегмента. Такие структуры удобны для отображения отношений типа «один ко многим» в предметной области. Просмотр иерархической структуры возможен только с корневой вершины. Пропуск сегмента в иерархическом пути при доступе к заданному сегменту не допускается. Основные недостатки иерархической структуры: трудность (неэффективность) отображения отношений типа «многие ко многим»;

длительность доступа к сегментам, находящимся на нижних уровнях иерархии; ориентированность на определенный тип (разрез) запроса.

Сетевые модели графически отображаются в виде графа. Вершинам графа соответствуют составные единицы информации (записи). Экземпляры записей образуют файлы. Структура записи может быть иерархической или линейной в зависимости от системы. Между парой типов записей может быть объявлено несколько связей, имена и направления связей должны быть четко обозначены. Недостатками являются: сложность (очень большое число параметров описания данных и операторов), а также неудобство навигационного доступа.

Реляционная база данных - это множество отношений. Реляционная модель основана на математической логике и является простейшей и наиболее привычной формой представления данных в виде таблицы. Строка таблицы эквивалентна записи файла базы данных, а колонка - полю записи. Доступ к элементу данных осуществляется посредством связи требуемой строки (записи) с требуемой колонкой (полем). Достоинством реляционной модели является сравнительная простота инструментальных средств ее поддержки, недостатком - жесткость структуры данных (например, невозможность задания строк таблицы произвольной длины) и зависимость скорости ее работы от размера базы данных.

Преимущества использования реляционных базы данных состоит в следующем:

- простота - в реляционной модели данных существует всего одна информационная конструкция, которая формализует табличное представление данных, привычное для пользователей;

- теоретическое обоснование - наличие теоретически обоснованных методов нормализации отношений позволяет получать базы данных с заранее заданными свойствами (в основном, с гарантией минимальной избыточности представления данных);

- независимость данных - когда необходимо изменить структуру реляционной базы данных, то это приводит к минимальным изменениям в программном продукте.

Моделью логической структуры базы данных была выбрана именно реляционная, так как она позволяет довольно быстро сформировать связи между таблицами для правильного построения запросов к базе данных и также легко разорвать эти связи и создать новые для построения другого запроса. Кроме того, архитектура построения связи более проста и время выполнения запроса в реляционной модели выше чем при использовании сетевой или иерархической структуры.

2.3 Обоснование проектных решений по программному обеспечению (ПО)

Программное обеспечение (ПО) - совокупность программ системы обработки данных и программных документов, необходимых для эксплуатации этих программ. ПО предназначено для придания вычислительной системе определенных свойств, связанных с увеличением производительности, повышением достоверности получаемых результатов, повышением надежности функционирования системы, улучшения работы пользователя.

Разработка автоматизированного мониторинга будет производиться для региона, в котором используется комплекс ТрастМед, разработанные компанией ООО «СофТраст». Основная масса бизнес-приложений, базируется на уникальной технологической платформе Malibu™. Malibu™ реализован на платформе .NET и требует для работы MS Windows XP/Vista, MS .NET Framework, MS SQL Server/Express.

Основой для веб-приложений, разрабатываемых в компании, является платформа .Net Core Web API и Angular.

Microsoft SQL Server — система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных

размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка [6].

Существуют и другие системы управления реляционными базами данных. Например, такая как, MySQL разработанная в середине 90х (позже приобретённая Oracle), MySQL была одной из первых баз данных с открытым исходным кодом и остаётся таковой и до сегодня. Это значит, что существует несколько альтернатив MySQL. Но различия между этими вариантами не слишком явные; синтаксис и основная функциональность остаётся одинаковой.

А что является отличительной чертой MySQL, так это её популярность среди стартап-сообществ. Открытый код и бесплатность даёт возможность разработчикам легко начать с MySQL и изменять свой код, когда понадобится. MySQL обычно используется вместе с PHP(англ.) и Веб-сервером Apache, в дистрибутивах Linux, что и привело к известной аббревиатуре LAMP (Linux, Apache, MySQL, PHP).

Но в случае работы медицинских организаций, приложения для которых являются крупномасштабными проектами, необходима платформа для решения критически важных задач в масштабе предприятия, обладающая высокой доступностью, повышенной производительностью и безопасностью, т.е. в данном случае идеально подходит Microsoft SQL Server.

Плюсом является и то, что SQL сервер больше подходит для разработчиков, использующих .NET в качестве языка разработки. А платформа .NET как раз используется в компании ООО «СофтТраст» для разработки приложений для медицинских организаций.

Также, SQL сервер поддерживается одной из самых больших компаний в мире. Microsoft создало дополнительные инструменты для SQL сервера, которые привязываются к реляционной СУБД, включая инструменты для анализа данных. Система также имеет сервер отчётов – Служба отчётов SQL Сервера. Это делает SQL сервер швейцарским армейский ножом среди реляционных СУБД.

2.4 Обоснование проектных решений по технологическому обеспечению

Технологический процесс – последовательность технологических операций, необходимых для выполнения определённого вида работ. Технологический процесс состоит из рабочих операций, которые в свою очередь складываются из рабочих движений.

Технологический процесс включает: сбор и регистрацию исходных данных, передачу их на обработку, хранение, подготовку данных к обработке, ввод данных в ЭВМ, обработку информации по заданным алгоритмам, выдачу результирующей информации и передачу её пользователям.

На этапе обработки данных важно использовать нормативно-справочную информацию. Это даст некоторые преимущества в скорости выбора и сортировки. При этом также необходим предварительный просмотр полученных данных перед их оформлением и выдачей. На этом этапе становится актуальным выбор правильного режима – пакетного или диалогового.

Первый вариант предполагает уменьшение вмешательства самого пользователя в процесс решения задачи, предоставляя ему только возможность вводить и корректировать данные, что не всегда удобно для самого исполнителя.

На практике эффективнее и стабильнее показывает себя диалоговый режим, обеспечивающий более гибкую связь и дающий целый ряд преимуществ: удобен при работе, обеспечивает защиту от нежелательного доступа, позволяет непосредственно участвовать в решении задачи, даёт быстрый доступ к данным в любой момент времени и хорошую управляемость процессом, а также отлично осуществляет переход от одной операции к другой.

2.5 Обоснование выбора программных средств

Для разработки автоматического мониторинга была выбрана система SQL Server Reporting Services (SSRS).

SSRS — это программная серверная система, обеспечивающая набор готовых к применению средств и служб для создания, управления и работы с отчетами компании. Функции программирования Microsoft SQL Server Reporting Services помогают расширить и настроить функциональность отчетов.

Службы Reporting Services содержат инструменты и функции, позволяющие создавать отчеты в любых удобных форматах, используя широкий спектр источников информации, а также включают полный набор уже знакомых инструментов, позволяющих осуществлять управление и производить доставку отчетов, обеспечивая безопасность решения. Службы SSRS имеют API-интерфейсы, которые дают возможность разработчикам проводить интеграцию или расширять возможности обработки данных и отчетов в пользовательских приложениях. Отчеты обрабатываются и доставляются быстро и эффективно, что позволяет пользователям получать отчеты как автоматически по подписке, так и другими удобными для них способами.

Инструменты служб SSRS работают и проектируются в среде Microsoft Visual Studio, целиком интегрированы с инструментами и компонентами SQL Server, а также со службами Microsoft Office SharePoint. Режим интеграции с SharePoint позволяет осуществлять совместную работу над документами, такую как проверку отчетов, внесение и отслеживание изменений в документах, что в свою очередь может существенно дополнить контролируемость и гибкость среды отчетов.

С помощью служб SSRS отчеты можно создавать и просматривать в любом удобном для пользователей формате, включая HTML, PDF, CSV, XML, в виде рисунка TIFF, а также в виде документов Microsoft Office или как часть узла SharePoint.

С помощью Report Manager (веб-приложение которое взаимодействует с веб-службой Report Server) пользователи могут просматривать, публиковать, управлять, обрабатывать или осуществлять доступ к отчетам при необходимости.

На рисунке 2.1 представлена схема работы SQL Server Reporting Services.

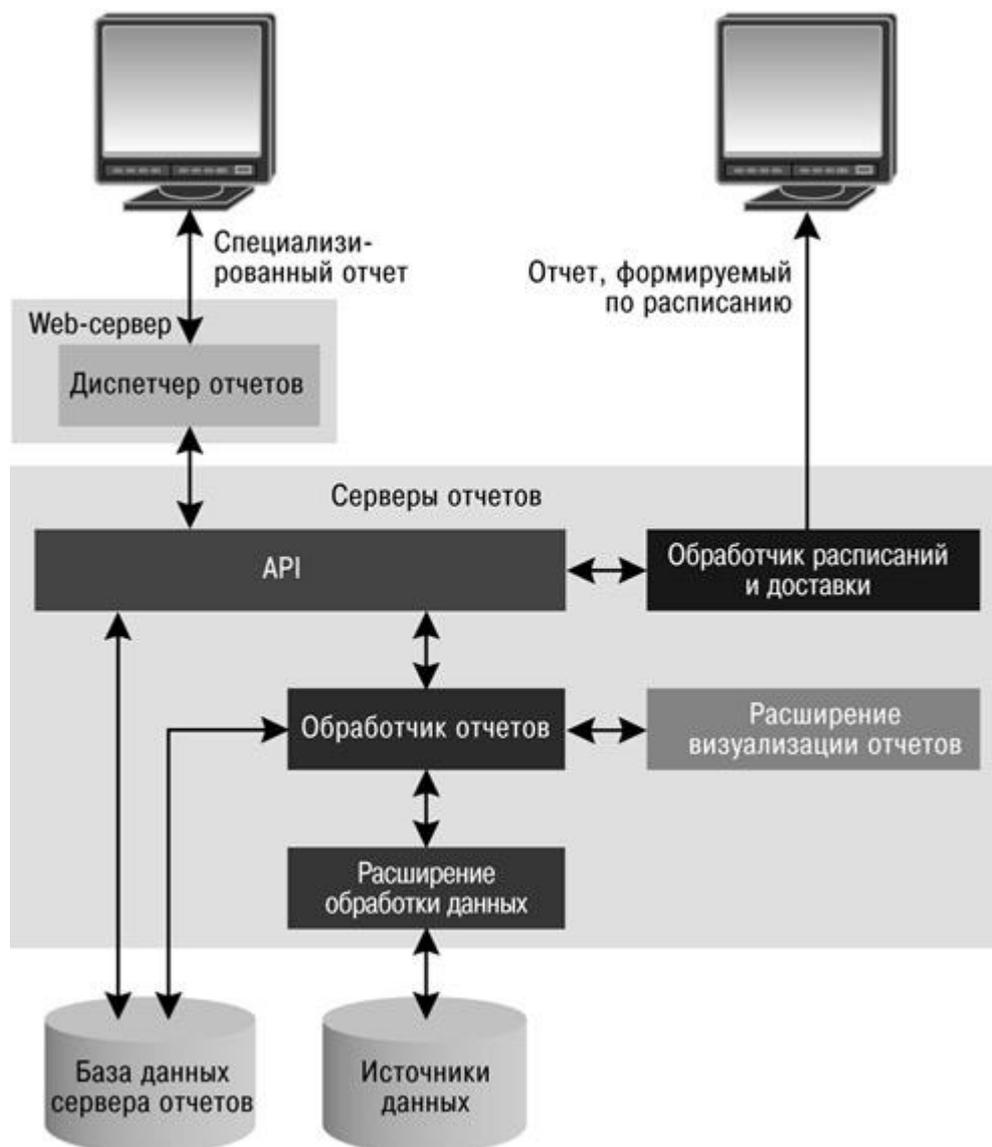


Рисунок 2.1 - Схема работы SQL Server Reporting Services

Помимо интерактивных, табличных, графических, веб-службы Reporting Services также дают возможность создавать нерегламентированные отчеты на базе стандартных моделей, для интерактивного просмотра данных внутри модели, а также экспортировать отчеты в другие приложения.

Microsoft SQL Server 2008 Reporting Services имеет следующие преимущества:

- быстрое и простое создание отчетов с форматированным текстом;
- безопасный доступ к отчетам, папкам, ресурсам и общим источникам данных благодаря ролевой модели безопасности SQL Reporting Services;

- доставки форматированных отчетов пользователям с помощью самых разных механизмов, один из которых автоматическая доставка отчетов посредством подписки;

- простое управление средой отчетов с помощью целого ряда различных средств и технологий, таких как интеграция с SharePoint Server и диспетчер Reporting Services Configuration Manager.

Построитель отчетов Microsoft SQL Server Report Builder - это инструмент создания отчетов с разбиением на страницы, предназначенный для корпоративных пользователей, которые предпочитают работать в автономной среде вместо использования конструктора отчетов в Visual Studio/SSDT. При проектировании отчета с разбиением на страницы создается определение отчета, в котором указано, откуда и какие брать данные, а также как их отображать. При запуске отчета обработчик получает заданное определение отчета, извлекает данные и объединяет их с макетом отчета, чтобы создать отчет. Отчет можно предварительно просмотреть в построителе отчетов. После этого отчет можно опубликовать в собственном режиме или в режиме интеграции с SharePoint на сервере отчетов Службы Reporting Services (2016 и более ранние версии).

Предоставляемые функции от Microsoft SQL Server Report Builder:

- создание отчетов с разбиением на страницы с использованием следующих макетов отчетов: табличных, матричных, с диаграммами и произвольной формы. Табличные отчеты целесообразно применять для данных, представленных в виде столбцов; матричные отчеты (такие, как перекрестные или сводные таблицы) — для сводных данных, а отчеты с диаграммами — для графических данных. Для других форматов данных используется произвольный тип отчета. В отчеты можно внедрять другие отчеты и диаграммы наряду со списками, графическими изображениями и элементами управления для динамических веб-приложений.

- отчеты на основе различных источников данных. Построение отчетов с использованием данных из любых источников, для которых есть управляемый поставщик данных Microsoft .NET Framework, поставщик OLE DB или источник

данных ODBC. Можно создавать отчеты, использующие реляционные и многомерные данные из SQL Server и служб Службы Analysis Services, Oracle, Hyperion и других баз данных. Чтобы получить данные из любого источника XML-данных, можно воспользоваться модулем обработки XML-данных. Для разработки пользовательских источников данных можно использовать функции, возвращающие табличное значение.

- изменение существующих отчетов. С помощью построитель отчетов можно настроить и обновить отчеты, созданные в конструкторе отчетов SQL Server Data Tools (SSDT).

- изменение данных путем фильтрации, группирования и сортировки либо путем добавления формул и выражений.

- добавление диаграмм, датчиков, инфокривых и индикаторов позволяет наглядно отобразить сводные данные и представлять большие объемы статистической информации в легко доступном виде.

- добавление интерактивных функций, например, схем документов, кнопок отображения и скрытия, а также ссылок для детализации вложенных отчетов и детализированных отчетов. С помощью параметров и фильтров можно отфильтровать данные для пользовательских представлений.

- внедрение или размещение ссылок на изображения и другие ресурсы, в том числе на внешнее содержимое.

Выводы по второму разделу

Во втором разделе был произведен выбор и обоснование по проектным решениям. В качестве программного обеспечения была выбрана система управления реляционными базами данных (РСУБД) Microsoft SQL Server. В качестве программного средства была выбрана система SQL Server Reporting Services.

3 Проектная часть

3.1 Информационное обеспечение задачи

3.1.1 Информационная модель и ее описание

Для разработки функциональной модели было подобрано средство проектирования AllFusion ProcessModeler 7. AllFusion – это мощный инструмент моделирования, который используется для анализа, документирования и реорганизации сложных процессов, в том числе, бизнес-процессов. Модель, созданная средствами AllFusion, позволяет четко документировать различные аспекты деятельности - действия, которые необходимо предпринять, способы их осуществления, требующиеся для этого ресурсы и др. Таким образом, формируется целостная картина деятельности предприятия - от моделей организации работы в маленьких отделах до сложных иерархических структур.

AllFusion совмещает в одном инструменте средства моделирования функций (IDEF0), потоков данных (DFD) и потоков работ (IDEF3).

С помощью функционального моделирования (нотация IDEF0), можно провести систематический анализ процессов и систем, сосредоточившись на регулярно решаемых задачах (функциях), свидетельствующих об их правильном выполнении показателях, необходимых для этого ресурсах, результатах и исходных материалах (сырье).

Моделирование потоков данных (DFD), часто используемое при разработке программного обеспечения, сосредоточено вокруг потоков данных, передающихся между различными операциями, включая их хранение, для достижения максимальной доступности и минимального времени ответа. Такое моделирование позволяет рассмотреть конкретный процесс, проанализировать операции, из которых он состоит, а также точки принятия решений, влияющих на его ход.

Моделирование потоков работ (нотация IDEF3) позволяет рассмотреть конкретный процесс, проанализировать операции, из которых он состоит, а также точки принятия решений, влияющих на его ход.

Функциональная модель предназначена для описания существующих бизнес-процессов на предприятии и идеального положения вещей. Также функциональная модель позволяет идентифицировать все информационные объекты, которыми оперирует предприятие в своей деятельности.

Для того, чтобы разработать автоматизированный мониторинг процесса постановки онкологического диагноза необходимо понимать процессы и функции, происходящие при этом. Таким образом, смоделировав процесс, будут выявлены главные функциональные блоки, которые облегчат программную реализацию.

Контекстная диаграмма «Постановка онкологического диагноза» - это диаграмма, расположенная на вершине древовидной структуры диаграмм, представляющая собой самое общее описание системы и ее взаимодействие с внешней средой (как правило, здесь описывается основное назначение моделируемого объекта). Контекстная диаграмма состоит из одного блока, описывающего функцию верхнего уровня, ее входы, выходы, управления, и механизмы, вместе с формулировками цели модели и точки зрения, с которой строится модель.

На рисунке 3.1 представлена контекстная диаграмма бизнес-процесса постановки онкологического диагноза. Ее отличие от диаграммы «КАК ЕСТЬ» в том, что автоматизированный мониторинг участвует на каждом этапе процесса и позволяет просматривать информацию по пациенту, в том числе определяя сколько дней ждет пациент для прохождения того или иного этапа.

На вход поступает подозрение на злокачественное заболевание у пациента. Механизмами являются пациент, врач первичного звена (терапевт, педиатр, врач общей практики), врач-онколог, лаборант, лечебно-профилактическое учреждение госпитализации (ЛПУ), автоматизированный мониторинг. Выходом из процесса постановки диагноза является факт госпитализации либо талон с

неподтвержденным диагнозом от врача-онколога. Все процессы регламентированы Приказом Министерства здравоохранения Российской Федерации от 15 ноября 2012 г. № 915н "Об утверждении Порядка оказания медицинской помощи населению по профилю "онкология".

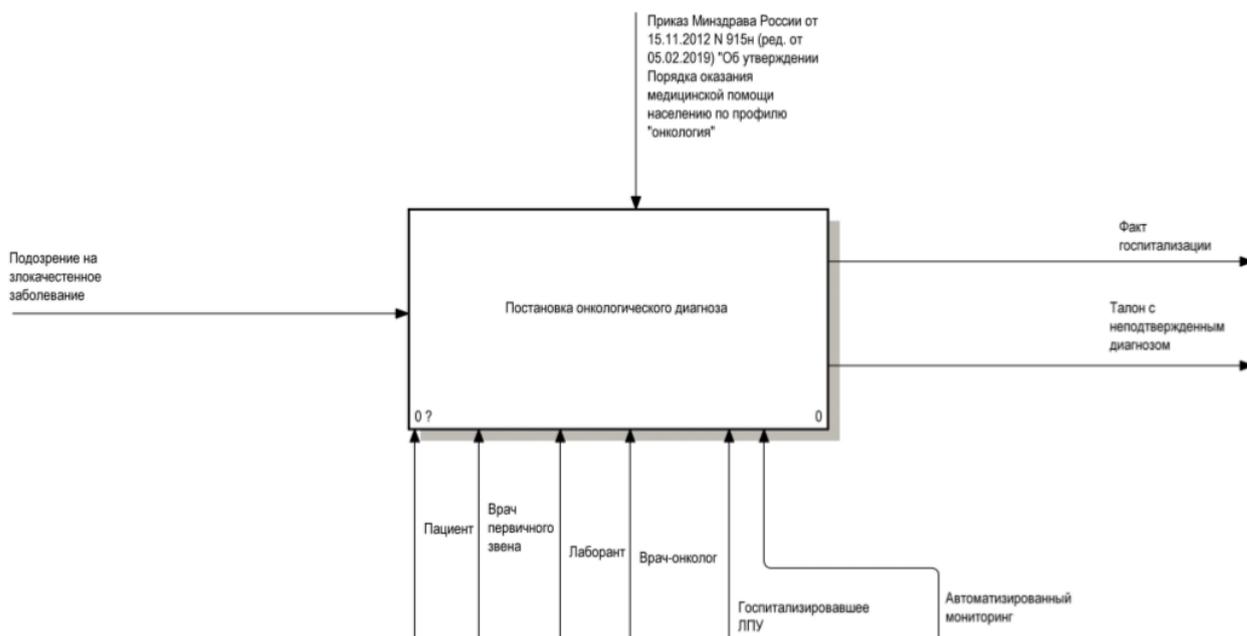


Рисунок 3.1 - Контекстная диаграмма

Диаграмма декомпозиции предназначена для детализации функций и получается при разбиении контекстной диаграммы на крупные подсистемы (функциональная декомпозиция) и описывает каждую подсистему и их взаимодействие.

Декомпозиция контекстной диаграммы представлена на рисунке 3.2. Она включает в себя три основных процесса: прием у врача первичного звена, постановка диагноза, госпитализация.

На приеме у врача первичного звена, врач при подозрении на злокачественное заболевание выписывает направление к врачу-онкологу, с которым пациент должен записаться на прием.

В блоке постановка диагноза пациент должен посетить врача-онколога. Стоит отметить, что пациент может выйти из процесса постановки онкологического диагноза на этапе постановки диагноза, т.к. врач-онколог может опровергнуть подозрение на злокачественное заболевание. В случае

подтверждения диагноза, врач-онколог выписывает направление на госпитализацию.

В блок госпитализации на вход поступает направление от врача-онколога. Выходом из процесса является заполнение талона фактической госпитализации пациента.

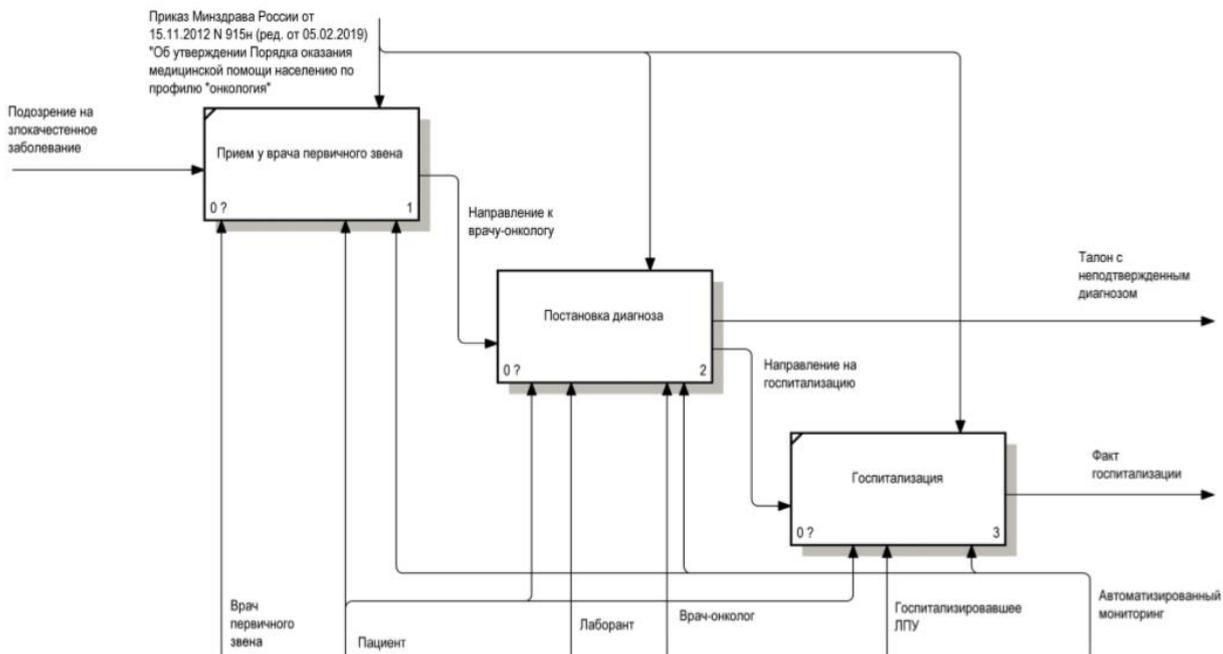


Рисунок 3.2 - Декомпозиция контекстной диаграммы

На рисунке 3.3 представлена декомпозиция процесса постановки диагноза.

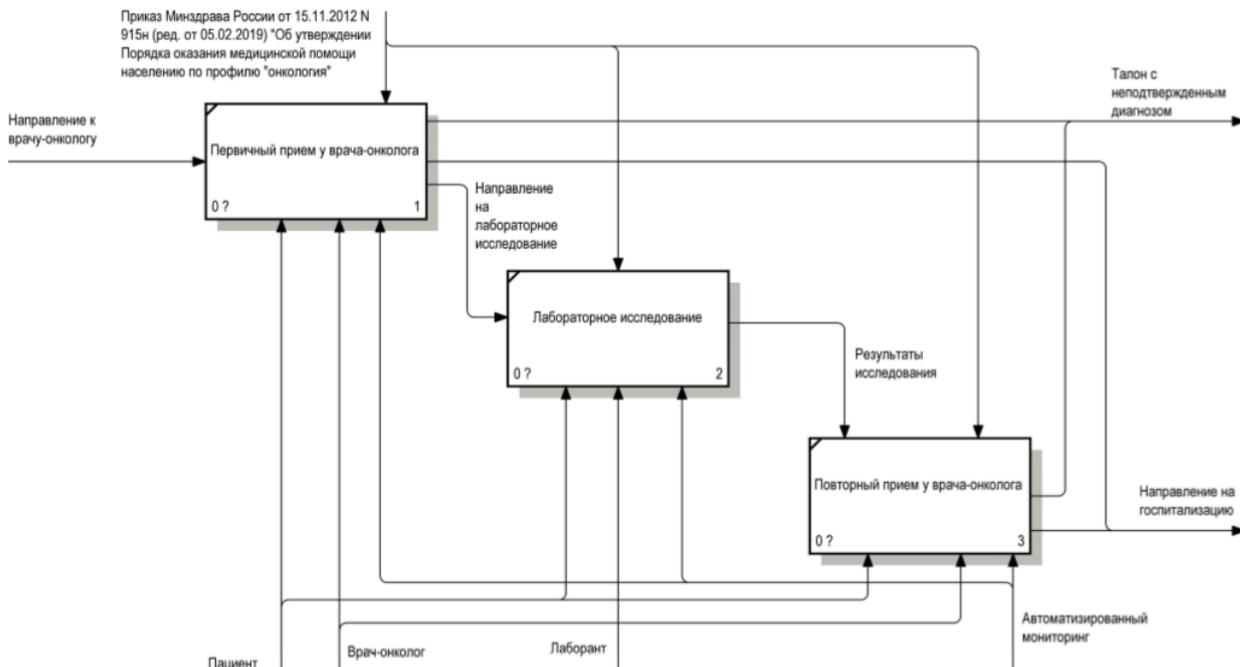


Рисунок 3.3 - Декомпозиция постановки диагноза

На основании выписанного направления к врачу-онкологу, пациент может записаться к нему на прием.

На первичном приеме врач-онколог может сразу же подтвердить или опровергнуть диагноз. В случае подтверждения диагноза, врач выписывает направление на госпитализацию.

В случае, если врач сомневается в диагнозе, он может направить пациента на лабораторное исследование. После проведения лабораторного исследования врач получает его результаты и подтверждает, либо опровергает подозрение на злокачественное заболевание.

3.1.2 Характеристика базы данных

ER-модель – это представление базы данных в виде наглядных графических диаграмм. Данная модель представляет собой набор концепций, которые описывают структуру БД в виде совокупности сущностей, атрибутов и связей. Основная цель разработки такой модели данных заключается в создании пользовательского восприятия данных и согласования большого количества технических аспектов, связанных с проектированием БД [7].

На рисунке 3.4 представлена структура ресурсов медицинских организаций hlt_DocPRVD.

Ресурсом может выступать врач, оборудование или кабинет и определяется справочником hlt_ResourceType.

В hlt_DocPRVD определяется специальность ресурса из справочника oms_PRVS, должность из справочника oms_PRVD и указывается отделение из таблицы oms_Department, в котором работает ресурс, который в свою очередь определяет подразделение и юридическое лицо из справочника oms_LPU. Отделение имеет свой профиль и определяется справочником oms_kl_DepatmentProfile.

Следует заметить, что у одного врача/оборудования/кабинета может быть несколько ресурсов.

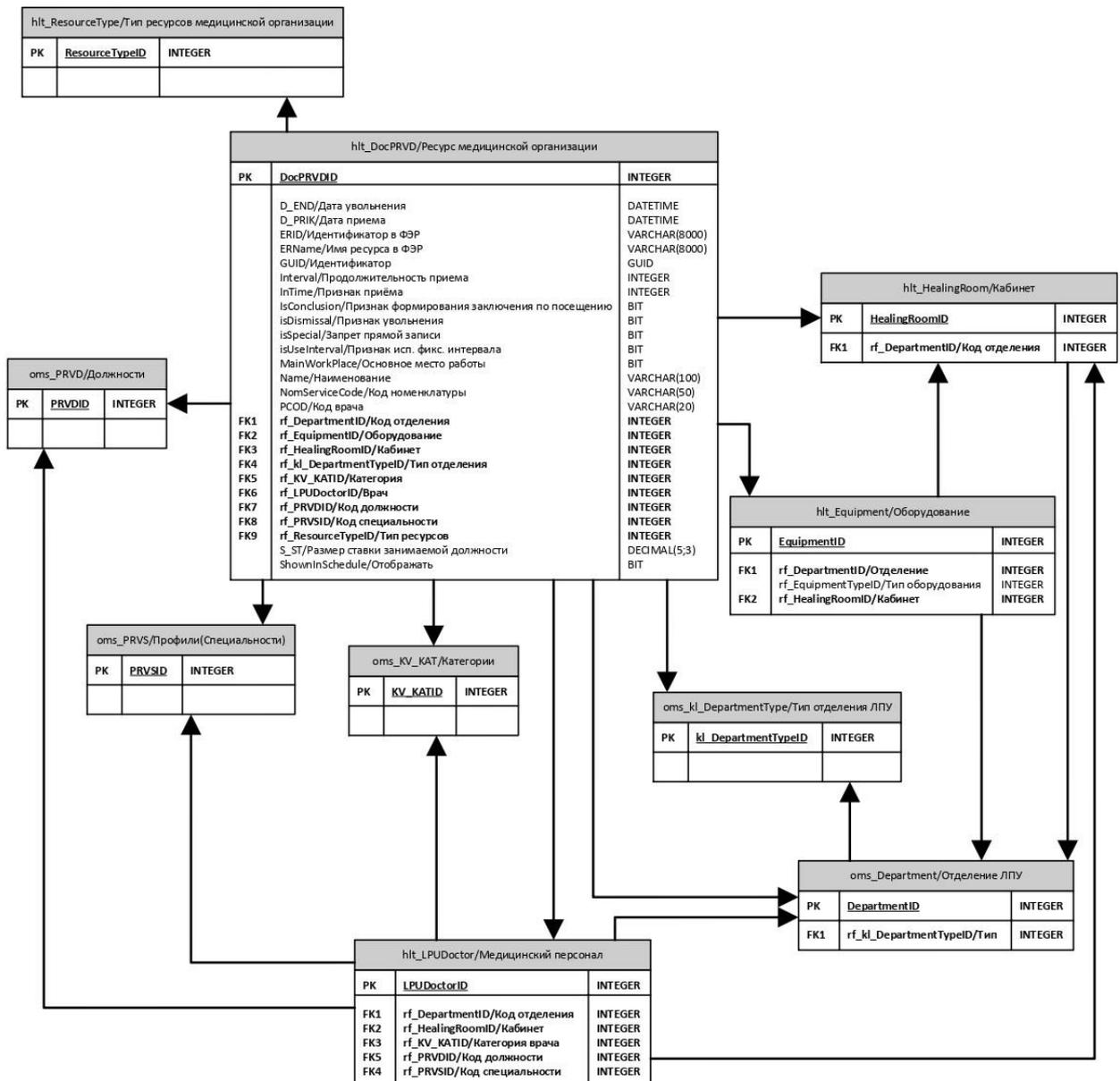


Рисунок 3.4 – ER-модель ресурсов медицинских организаций

В базе данных хранится таблица **hlt_MКАВ**, это медицинская карта амбулаторного больного. У одного человека может быть несколько карт в ЦОД и по одной карте в каждой медицинской организации, которые посещал пациент.

Для распознавания персоны, которой принадлежит несколько карт имеется таблица **gpt_mn_person_mkab**. Она содержит в себе идентификатор карты пациента и идентификатор персоны, таким образом можно определить какие карты принадлежат конкретному человеку, и в случае необходимости смотреть

историю нужного пациента по медицинской истории всех лечебно-профилактических учреждений.

Далее идет немало важная таблица `hlt_Direction` - это направления, которые выписывают врачи. Тип направления определяется справочником `hlt_TypeDirection` и бывает четырех видов:

- на госпитализацию;
- на консультацию;
- на обследование;
- на восстановительное лечение.

Таблица `lbr_LaboratoryResearch` содержит в себе направления на исследования, `lbr_Research` в свою очередь содержит результаты этого исследования.

На рисунке 3.5 представлена ER-схема расписания врача и посещения.

Таблица `hlt_DoctorTimeTable` содержит в себе расписания ресурса медицинской организации. Справочником `hlt_DocBusyType` определяется тип занятости ресурса.

Таблица `hlt_DoctorVisitTable` содержит в себе активные записи пациентов на прием. В `tmp_DoctorVisitTable_Cancel` хранятся отмененные записи на прием.

Таблица `hlt_ActionSchedule` содержит в себе связь между записью на прием и источником записи. Источником записи на прием может являться:

- направление на исследование;
- вызов на дом.

По итогам приема заполняется талон (таблица `hlt_TAP`), где врач указывает диагноз по международному классификатору болезней (МКБ-10) из справочника `oms_МКБ`, результат обращения (справочник `oms_kl_VisitResult`), признак подозрения на злокачественное новообразование (таблица `oms_onco_Sign`), исход заболевания (справочник `oms_kl_StatCureResult`) и многие другие дополнительные поля.

В таблице `hlt_WaitingList` хранятся листы ожидания. Они создаются на пациента в случае, если к врачу нет свободных ячеек приема.

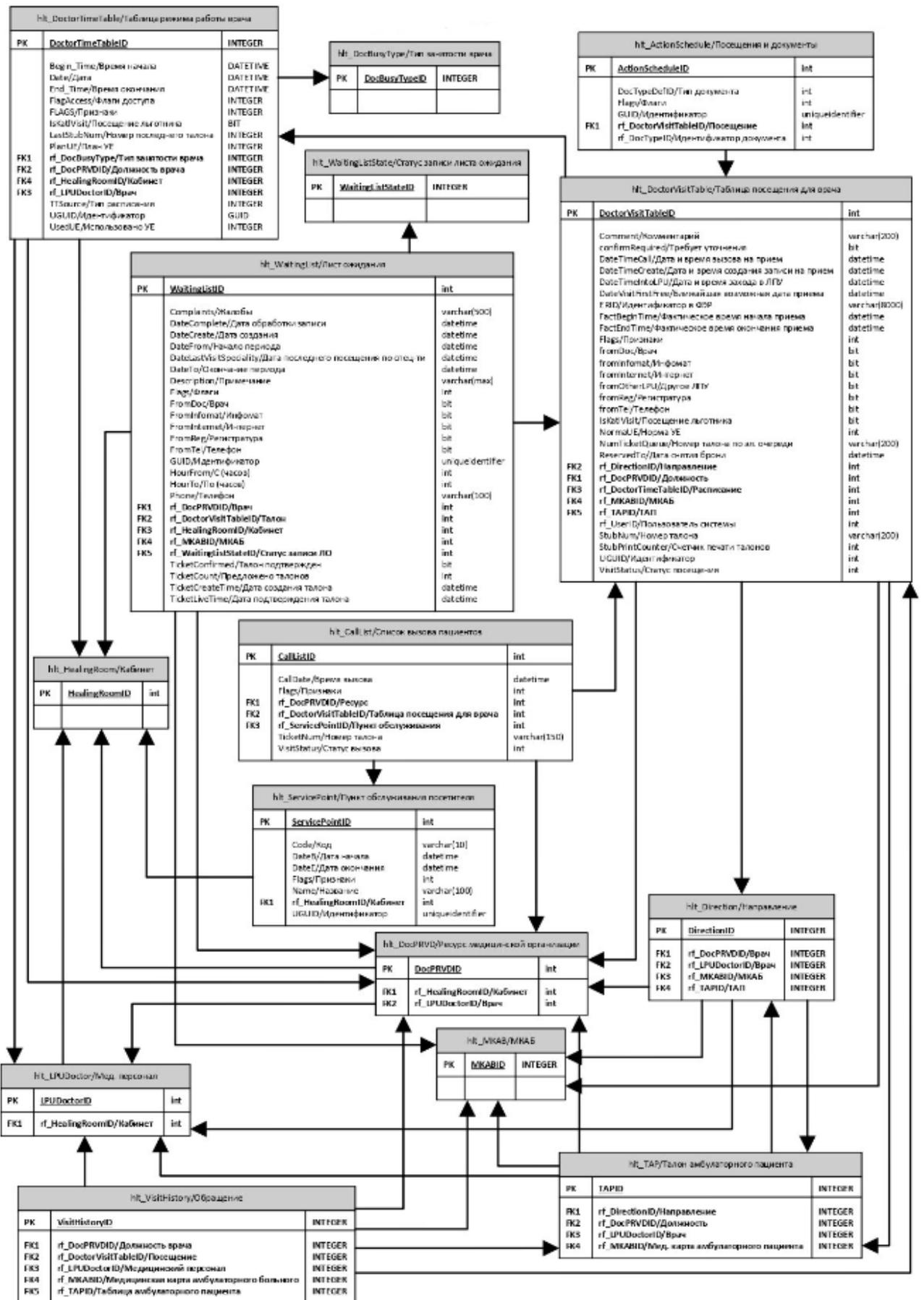


Рисунок 3.5 – ER-модель расписания и посещения

3.1.3 Характеристика результатной информации

Результатом прикладной обработки информации будут выступать 4 временные таблицы.

Описание первой самой сажной таблицы приведено в таблице 3.1. В ней хранится информация по пациенту, его направлению на консультацию к врачу-онкологу, направлению на госпитализацию и фактической госпитализации, если таковая имеется. Все последующие таблицы формируются, отталкиваясь от первой.

Таблица 3.1 – Описание таблицы tmp_dir_onko_new

Таблица	Атрибут	Идентификатор поля	Тип поля
tmp_dir_onko_new	порядковый номер	n	bigint
	номер направления	Num	varchar(50)
	фамилия	FAMILY	varchar(50)
	имя	NAME	varchar(50)
	отчество	OT	varchar(50)
	др	BD	datetime
	серия полиса	S_POL	varchar(16)
	номер полиса	N_POL	varchar(50)
	идентификатор мкаб	rf_MKABID	int
	дата направления	DateDir	datetime
	направивший врач	rf_DocPRVDID	int
	диагноз из направления	rf_MKBDID	int
	код диагноза	DS	varchar(10)
	специальность	PRVS_NAME	varchar(100)
	направившее лпу	rf_LPUSenderID	int
	лпу направления	rf_LPUID	int
	диагноз не С	tap_MKBDID	int
	дата диагноза не С	tap_DateTAP	datetime
	диагноз С	tapC_MKBDID	int
	дата диагноза С	tapC_DateTAP	datetime
	дата направления на госпитализацию	DateDirHosp	datetime
	дата плановой госпитализации	DatePlanHosp	datetime
лпу направления	LPUHosp	int	
дата направления на госпитализацию	DateDirHosp1	datetime	
дата фактической госпитализации	DateFactHosp	datetime	

Продолжение таблицы 3.1

Таблица	Атрибут	Идентификатор поля	Тип поля
tmp_dir_onko_new	лпу фактической госпитализации	LPUHospFact	int
	причина закрытия мкаб	CloseMkabDeath	varchar(10)
	признак исключения	Iskl	int
	новый да/нет	isNew	int

В таблице tmp_zap_onko_new хранятся все записи включая отмененные на прием к врачу-онкологу по направлению. Важными полями являются даты создания записи на прием, приема, отмены записи и статус приема. Описание приведено в таблице 3.2.

Таблица 3.2 – Описание таблицы tmp_zap_onko_new

Таблица	Атрибут	Идентификатор поля	Тип поля
tmp_zap_onko_new	порядковый номер	n	bigint
	фамилия	FAMILY	varchar(50)
	имя	NAME	varchar(50)
	отчество	OT	varchar(50)
	др	BD	datetime
	дата направления	DateDir	datetime
	направивший врач	dirDoc	int
	дата создания записи на прием	DateTimeCreate	datetime
	дата приема	Begin_Time	datetime
	врач према	rf_DocPRVDID	int
	тап	rf_TAPID	int
	статус посещения	VisitStatus	int
	специальность	PRVS_NAME	varchar(100)
	фио	V_FIO	varchar(37)
	идентификатор мкаб	MKABID	int
	диагноз из направления	DS	varchar(10)
	дата отмены	canceldate	datetime
	направившее лпу	rf_LPUSenderID	int
	лпу направления	rf_LPUID	int
	новый да/нет	isNew	int

В таблице tmp_dir_diag_new содержатся направления на исследования от врача-онколога и время активной записи на прием, если таковая имеется.

Важными полями являются даты направления на исследование, приема и признаки проведения и выполнения.

Признак проведения исследования означает явку пациента на исследование.

Признак выполнения говорит о том, что по исследованию имеются результаты и их внесли в информационную систему. Описание таблицы приведено в таблице 3.3.

Таблица 3.3 – Описание таблицы tmp_dir_diag_new

Таблица	Атрибут	Идентификатор поля	Тип поля
tmp_dir_diag_new	порядковый номер	n	bigint
	фамилия	FAMILY	varchar(50)
	имя	NAME	varchar(50)
	отчество	OT	varchar(50)
	др	BD	datetime
	дата направления	Date_Direction	datetime
	дата приема	Begin_Time	datetime
	дата консультации	DateRasp	datetime
	дата направления на консультацию	DateDir	datetime
	направление на исследование	LaboratoryResearchID	int
	признак проведения	IsPerformed	bit
	признак выполнения	isComplete	bit
	идентификатор мкаб	MKABID	int

В таблице tmp_zap_diag_new хранятся все записи включая отмененные на прием по направлению от врача-онколога на исследование. По структуре таблица похожа на таблицу с записями на консультацию к врачу-онкологу и имеет такую же значимость полей. Описание приведено в таблице 3.4.

Таблица 3.4 – Описание таблицы tmp_zap_diag_new

Таблица	Атрибут	Идентификатор поля	Тип поля
tmp_zap_diag_new	порядковый номер	n	bigint
	фамилия	FAMILY	varchar(50)
	имя	NAME	varchar(50)

Продолжение таблицы 3.4

Таблица	Атрибут	Идентификатор поля	Тип поля
tmp_zap_diag_new	отчество	OT	varchar(50)
	др	BD	datetime
	направление на исследование	LaboratoryResearchID	int
	дата создания записи на прием	DateTimeCreate	datetime
	дата приема	Begin_Time	datetime
	врач приема	rf_DocPRVDID	int
	тап	rf_TAPID	int
	статус посещения	VisitStatus	int
	специальность	PRVS_NAME	varchar(100)
	фио	V_FIO	varchar(37)
	идентификатор мкаб	MKABID	int
	дата отмены	canceldate	datetime
	направившее лпу	rf_LPUSenderID	int
	лпу направления	rf_LPUID	int

Связь между таблицами производится по ФИО и дате рождения, это связано с тем, что карты и записи на прием могут приходиться из разных медицинских организаций, соответственно идентификаторы карт будут отличаться.

3.2 Программное обеспечение задачи

Центр обработки данных (ЦОД) хранит в себе медицинскую историю всех медицинских организаций региона. Автоматизированный мониторинг процесса постановки онкологического диагноза на основе слабосвязанных медицинских данных включает в себя сложную логику обработки информации, потому что она охватывает информацию не в пределах одного врача или медицинской организации, а в пределах региона, т.к. движение пациента проводится по разным организациям. Например, направление к врачу онкологу может быть выписано из ЛПУ № 1 в ЛПУ № 2, а госпитализация состоится в ЛПУ № 3. В связи с вышеперечисленными фактами было принято решение на создание временных таблиц с предобработкой нужной информации.

В связи с тем, что мониторинг будет формироваться медицинскими организациями ежедневно, то в нем нужна всегда актуальная информация. Для этого необходимо постоянно актуализировать данные во временных таблицах. Решением данной проблемы является создание «Задания» на сервере баз данных.

Задания (jobs) — это наиболее часто используемое средство автоматизации административных операций. Задания можно определить, как именованные наборы действий, которые можно запланировать для выполнения по расписанию (а можно выполнять и вручную). На многих рабочих серверах на предприятиях существует сложная система заданий, при помощи которых выполняется множество операций: резервное копирование, проверка целостности, дефрагментация и перестроение индексов, загрузка и выгрузка данных, генерация страниц HTML и т. п. Задания могут создаваться и в автоматическом режиме, например, при настройке доставки журналов или репликации [8].

На рисунке 3.6 представлен пример создания «Задания» в среде разработки Microsoft SQL Management Studio.

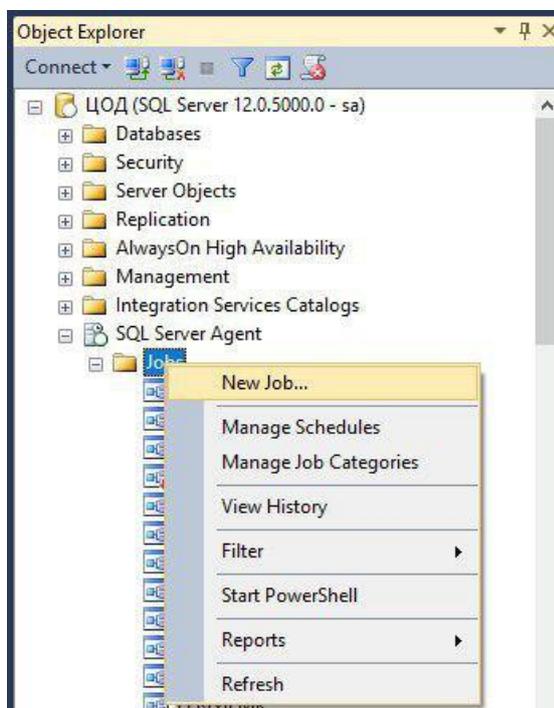


Рисунок 3.6 – Создание «Задания»

Для формирования временных таблиц будут использоваться функции T-SQL.

`ROW_NUMBER` - Нумерует выходные данные результирующего набора. В частности, возвращает последовательный номер строки в секции результирующего набора, 1 соответствует первой строке в каждой из секций.

`MAX/MIN` - Возвращает максимальное/минимальное значение выражения.

`ISNULL` - Заменяет значение `NULL` указанным замещающим значением.

`(NOT) EXISTS` - Указывает вложенный запрос для проверки существования строк.

`CAST/CONVERT` - Эти функции преобразуют выражение одного типа данных в другой.

`(NOT) LIKE` - Определяет, совпадает ли указанная символьная строка с заданным шаблоном. Шаблон может включать обычные символы и символы-шаблоны. Во время сравнения с шаблоном необходимо, чтобы его обычные символы в точности совпадали с символами, указанными в строке. Символы-шаблоны могут совпадать с произвольными элементами символьной строки. Использование символов-шаблонов в отличие от использования операторов сравнения строки (`=` и `!=`) делает оператор `LIKE` более гибким. Если тип данных одного из аргументов не является символьной строкой, компонент `SQL Server Database Engine`, если это возможно, преобразует его в тип данных символьной строки.

`DATEADD` - Эта функция добавляет указанное целое число со знаком к заданному аргументу входного значения `date`, а затем возвращает это измененное значение.

`WITH()` - Табличные подсказки переопределяют поведение оптимизатора запросов по умолчанию на время выполнения инструкции языка обработки данных (`DML`) указанием способа блокировки, одного или более индексов, операции обработки запроса, например просмотра таблицы или поиска в индексе, или других параметров. Табличные указания задаются в предложении `FROM` инструкции `DML` и относятся только к таблицам и представлениям, на которые ссылается это предложение.

GETDATE - Возвращает текущую системную метку времени базы данных в виде значения datetime без смещения часового пояса базы данных. Это значение наследуется от операционной системы компьютера, на котором работает экземпляр SQL Server.

Скрипт для наполнения временных таблиц приведен в приложении А.

После того, как «Задание» было создано и был указан скрипт, который должен выполняться ежедневно, было настроено расписание для выполнения. Выполнение скрипта происходит после полуночи каждый день, в связи с тем, что до полуночи загружается и обновляется вся медицинская история за сутки от медицинских организаций региона, а ночью нагрузка на сервер невысокая, что способствует быстрому формированию временных таблиц.

Автоматизированный мониторинг процесса постановки онкологического диагноза включает в себя комплекс отчетных форм, который состоит из 4 отчетов. Первый отчет, отображающий все проблемные случаи по региону в целом называется «Сводка по ЛПУ по процессу постановки онкологического диагноза». На рисунке 3.7 представлена сформированная форма этого отчета в Microsoft SQL Server Report Builder.

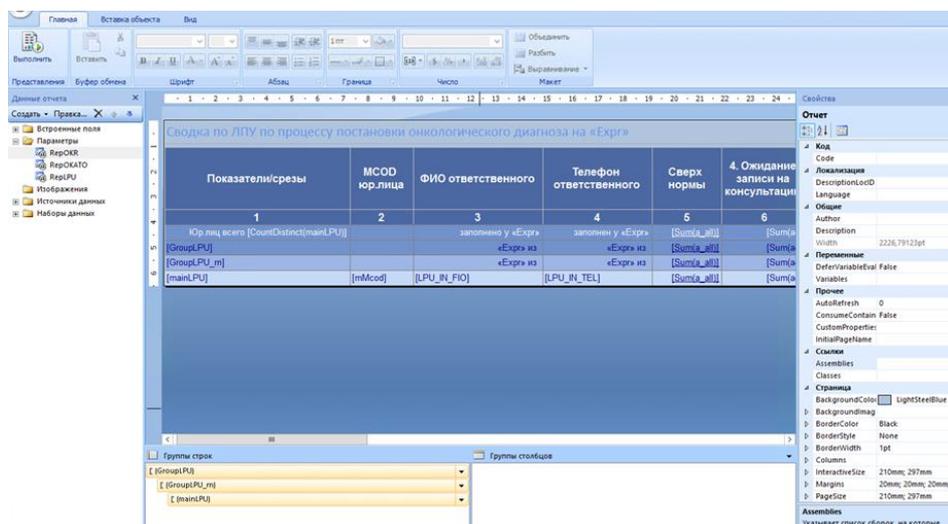


Рисунок 3.7 - Форма отчета «Сводка по ЛПУ по процессу постановки онкологического диагноза»

Слева от формы расположены источники данных, параметры, наборы данных.

Данные отчета могут поступать из нескольких источников данных в организации. Первым шагом при разработке отчета является создание источников и наборов данных, из которых берутся базовые данные отчета. Каждый источник данных содержит сведения о подключении к данным. У каждого набора данных есть команда запроса, определяющая набор полей для использования в качестве данных из источника данных.

Управление источниками данных отчета производится на веб-портале для сервера отчетов, работающего в собственном режиме, либо с помощью страниц приложения на сайте SharePoint, если сервер отчетов развернут в режиме интеграции с SharePoint.

На рисунке 3.8 представлено создание источника данных для отчета.

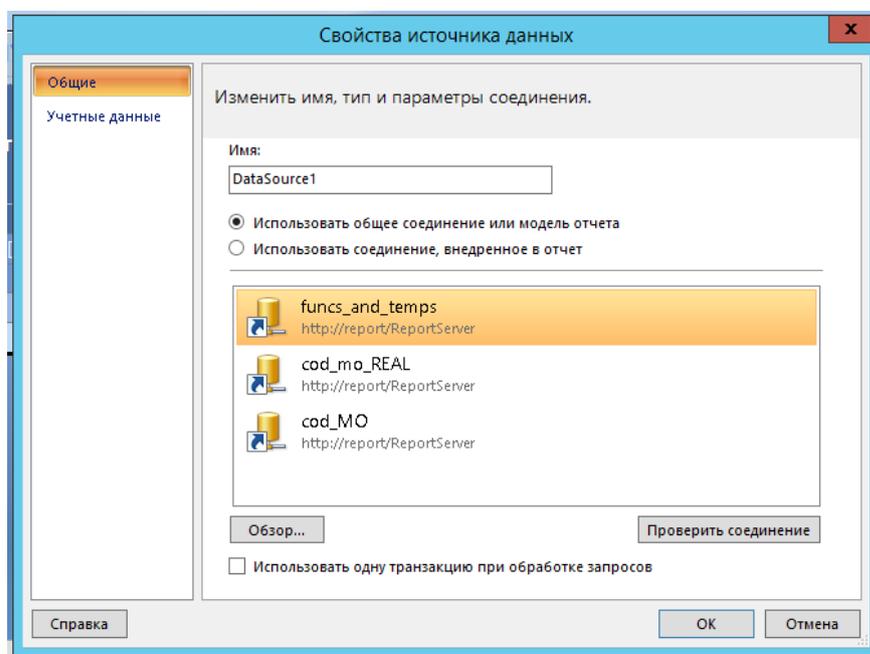


Рисунок 3.8 - Создание источника данных

Чтобы добавить данные в отчет, необходимо создать наборы данных. Каждый набор данных представляет результирующий набор команды запроса к источнику данных. Столбцы в результирующем наборе представляют собой коллекцию полей. Стройки в результирующем наборе представляют данные. Набор данных не содержит фактических данных. Набор данных содержит информацию, необходимую для получения определенного набора данных из источника данных.

Есть два типа наборов данных: внедренные и общие. Внедренный набор данных определяется в отчете и используется только этим отчетом. Общий набор данных определяется на сервере отчетов или на сайте SharePoint и может использоваться несколькими отчетами. В построителе отчетов можно создавать общие наборы данных в режиме «Общий набор данных» либо внедренные наборы данных в режиме «Конструктор отчетов». В конструкторе отчетов в среде SQL Server Data Tools (SSDT) можно создавать общие наборы данных как часть проекта либо внедренные наборы данных как часть отчета.

На рисунке 3.9 показано создание набора данных для мониторинга процесса постановки онкологического диагноза.

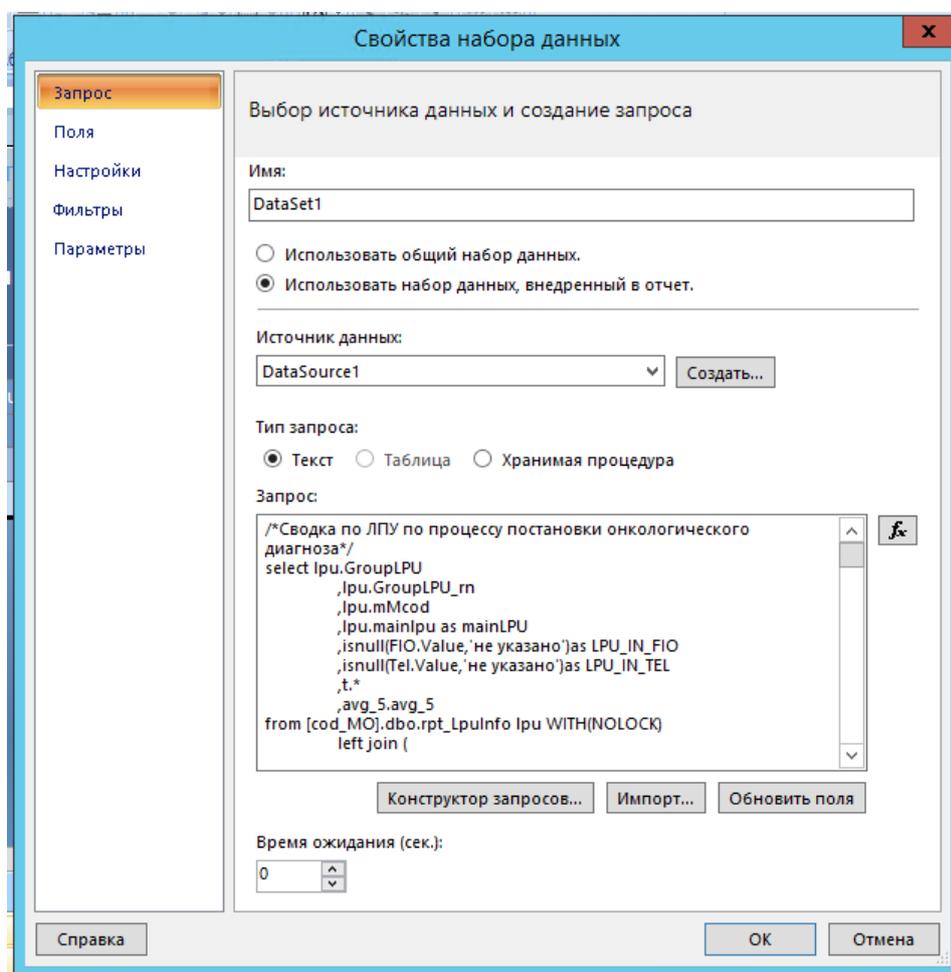


Рисунок 3.9 - Создание набора данных

Слева в окне создания набора данных есть вкладки, с помощью которых можно отфильтровать данные, переименовать результирующие поля, настроить передаваемые параметры.

Параметры отчета позволяют управлять данными отчета, объединять связанные отчеты и изменять внешний вид отчетов. Они часто используются для фильтрации данных отчета как до, так и после того, как данные возвращены отчету. Параметры отчета создаются следующими способами:

- автоматически при определении запроса набора данных, содержащего переменные запроса. Для каждой переменной запроса создается соответствующий параметр запроса набора данных и параметр отчета с тем же именем. Параметр запроса может выступать в качестве ссылки на переменную запроса или выходной параметр хранимой процедуры;

- автоматически при добавлении ссылки на общий набор данных, содержащий параметры запроса;

- вручную при создании параметров отчета на панели данных отчета. Параметры являются одной из встроенных коллекций, которые можно включать в выражения отчета. Так как выражения используются для задания значений повсюду в определении отчета, можно использовать параметры, чтобы управлять видом отчета или чтобы передавать значения в связанные вложенные отчеты или отчеты, которые тоже используют параметры.

Ниже формы отчета находится панель группировки. На панели группировки отображаются группы строк и столбцов для текущей области данных таблица. Панель группирования недоступна для областей данных диаграммы и датчика. Панель группирования состоит из панелей «Группы строк» и «Группы столбцов». Панель группировки: по умолчанию и расширенный. В режиме по умолчанию отображается иерархическое представление динамических элементов групп строк и столбцов. В расширенном режиме отображаются как динамические, так и статические элементы групп строк и столбцов. Группа – это именованный набор данных из набора данных отчета, который отображается в области данных. Группы организованы в иерархии, включающие динамические и статические элементы.

После создания группы можно установить свойства данных, уникальные для области данных, такие как выражения фильтрации и сортировки, разрывы

страниц и групповые переменные для хранения данных, уникальных для определенной области.

На рисунке 3.10 приведен пример создания группировки по ТУ. Сортировка по умолчанию производится по тому же полю, что и группировка.

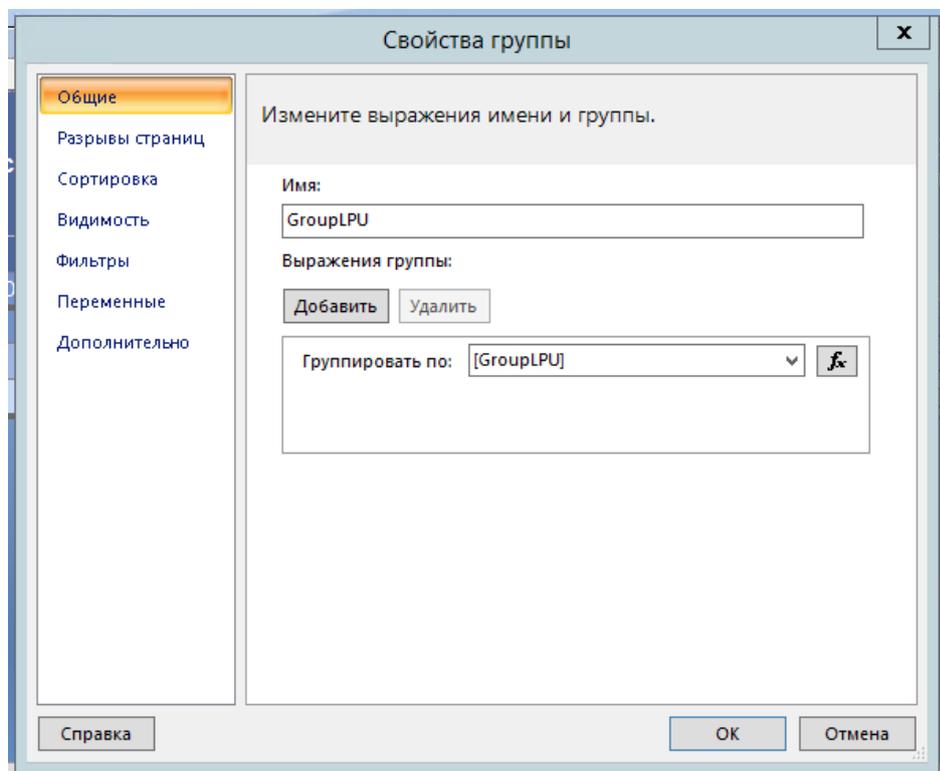


Рисунок 3.10 – Создание группировки

Справа от формы находится панель свойств, которые помогают:

- управлять внешним видом области данных таблица;
- управлять расположением области данных таблица и ее расширением на странице;
- управлять подготовкой к просмотру области данных таблица на странице;
- управлять подготовкой к просмотру групп строк и столбцов таблица.

Аналогичным образом были созданы и настроены еще 3 отчетные формы, содержащие в себе иерархическую детализацию показателей. Скрипты для наборов данных для отчетов приведены в приложении Б.

3.3 Описание контрольного примера реализации проекта

После создания и настройки отчетных форм, они были размещены на ресурсе для доступа пользователям к мониторингу.

Доступ осуществляется через веб-браузер. После авторизации на сервисе пользователь выбирает нужную папку «Мониторинг онкозаболеваний» с отчетом (рисунок 3.11) и запускает отчет «Сводка по ЛПУ по процессу постановки онкологического диагноза» (рисунок 3.12).

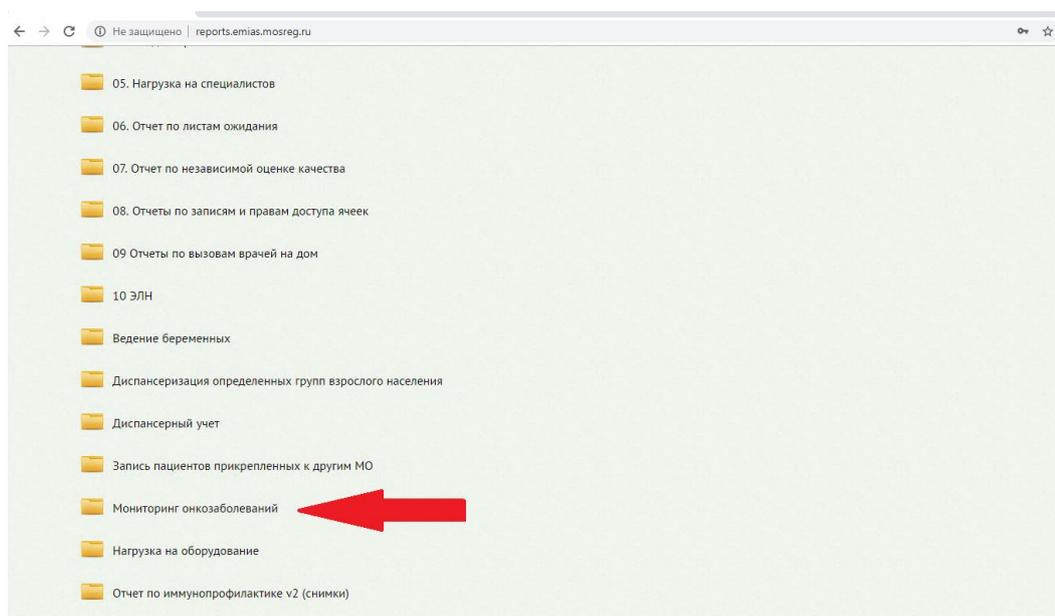


Рисунок 3.11 – Папка с отчетом

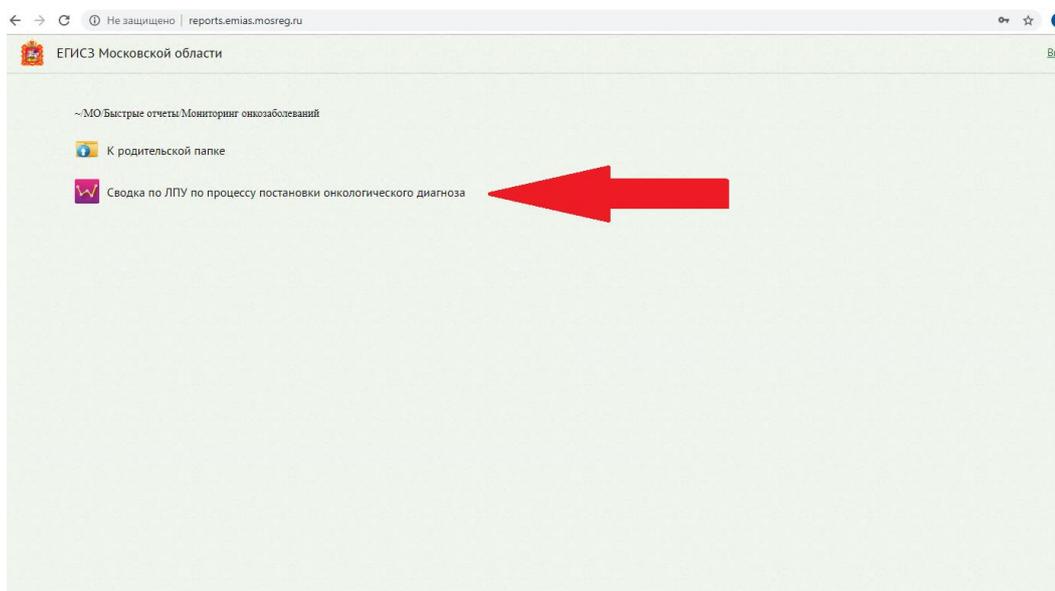


Рисунок 3.12 – Запуск отчета

Следующим действием в параметрах нужно выбрать интересующие ТУ, районы, ЛПУ или оставить их значения по умолчанию. Нажать кнопку «Просмотреть отчет». После того как отчет сформируется, можно раскрыть уровни группировки до нужной медицинской организации, проанализировать показатели и при необходимости нажать на показатель для формирования отчета с детализацией (рисунок 3.13).

1	2	3	4	5	6	7	8	9	10	
Юр.лиц всего	123	заполнено у	73	заполнен у	72	2226	514	504	10	124
01		40 из	63	40 из	63	334	148	141	7	11
Ивантеевка г.о.		1 из	1	1 из	1	3	3	3	0	0
Королев г.о.		3 из	6	3 из	6	23	12	12	0	1
Лосино-Петровский г.о.		3 из	4	3 из	4	27	27	27	0	0
Мытищи г.о.		11 из	14	11 из	14	50	21	21	0	0
Подольск г.о.		0 из	2	0 из	2	-	-	-	-	-
Пушкинский м.р.		8 из	10	8 из	10	74	22	18	4	3
Сергиево-Посадский м.р.		1 из	8	1 из	8	13	4	4	0	1
Троицкий		0 из	1	0 из	1	-	-	-	-	-
Фрязино г.о.		2 из	2	2 из	2	13	11	11	0	0
Щелково г.о.		13 из	17	13 из	17	131	48	45	3	6
[281601] ГБУЗ МО Мытищинский ПТД	281601	Гнедько Николай Иванович	8-495-586-30-40	-	-	-	-	-	-	-
[540101] ГБУЗ МО Щелковская РБ №2	540101	Яценко Александра Константиновна	+7(985)197-79-73	78	8	0	0	0	4	
[540201] ГБУЗ МО "Фряновская больница"	540201	Скрябина Ольга Александровна	8 (496) 563-51-05	-	-	-	-	-	-	-
[540301] ГБУЗ МО Монинская больница	540301	Сафонова Екатерина Анатольевна	89259332537	2	2	0	0	0	0	
[540401] ГБУЗ МО Загорянская поликлиника	540401	Оспанова Зоя Кадековна	84965631761	10	10	0	0	0	0	

Рисунок 3.13 – Отчет «Сводка по ЛПУ по процессу постановки онкологического диагноза»

После нажатия на кнопку детализации сформируется форма отчета «Сводка по процессу постановки онкологического диагноза». Здесь показатели будут более детализированы и отображать более ясную картину по одной медицинской организации (рисунок 3.14).

Показатели здесь делятся на подразделы и отображают количество новых пациентов на каждом из этапов. Каждый из этапов отображается с красной заливкой ячейки в случае, если имеются пациенты с несоблюдением сроков оказания медицинской помощи. Показатели сверх нормы можно детализировать до пациентов нажав на один из них (рисунок 3.15).

ЕГИСЗ Московской области

лпу [540101] ГБУЗ МО Щелковская РБ

Сводка по процессу постановки онкологического диагноза на 5 июня 2019 г.

№ п/п	Показатели	Пациентов	Случаев	Сверх нормы	из них новые	%
1	2	3	4	5	6	7
1	Пациентов в процессе	0	-	-	-	-
2	Время нахождения пациента в процессе	-	-	-	-	-
2.1	до 10 дней	-	0	-	-	-
2.2	от 10 до 30 дней	-	0	-	-	-
2.3	от 31 до 60 дней	-	0	-	-	-
2.4	от 61 до 90 дней	-	0	-	-	-
2.5	свыше 90 дней	-	0	-	-	-
3	Активных направлений на консультацию к онкологу	0	0	-	-	-
4	Ожидания записи на консультацию	8	8	8	0	100.00%
4.1	Первичная консультация	8	8	8	0	100.00%
4.11	Первая запись	0	0	0	0	0
4.12	Повторная запись	8	8	8	0	100.00%
4.2	Повторная консультация	0	0	0	0	0
4.21	Первая запись	0	0	0	0	0
4.22	Повторная запись	0	0	0	0	0
4.3	Отказались от записи	0	0	-	-	-
5	Плановое ожидание проведения консультации	4	4	4	0	100.00%
5.1	до 3 дней	0	0	-	-	-
5.2	более 3 дней	4	4	-	-	-
6	Ожидание проведения консультации	4	4	4	0	100.00%
6.1	до 3 дней	0	0	-	-	-
6.2	более 3 дней	4	4	-	-	-

Рисунок 3.14 – Отчет «Сводка по процессу постановки онкологического диагноза»

ЕГИСЗ Московской области

лпу [540101] ГБУЗ МО Щелковская РБ №2

Детальный отчет по процессу постановки онкологического диагноза на 5 июня 2019 г.
Ожидание записи на консультацию

лпу, допустившее нарушение	Данные пациента				Данные направления на консультацию к врачу онкологу					
	ФИО	Дата рождения	Полис	Адрес (почтовый и e-mail)	Дата направления	Диагноз	Номер направления	ФИО врача	ID врача	Л. прикрепления
	Кол-во сверх нормы - 8									
					26.01.2019	N60.1	541401666	Яценко Александра Константиновна	38320	[541401] ГБУЗ ГП №4
					13.03.2019	C50.0	5414012931	Юнусова Д. Ш. (Эндокринология)	4175	[541401] ГБУЗ ГП №4
					27.03.2019	D34	541201456	Лещенко А. В. (Терапия)	47741	[541201] ГБУЗ п. Свердловски
					04.04.2019	E06.3	541201561	Зайнутдинов Р. Р. (Терапия)	47741	[541201] ГБУЗ п. Свердловски
					04.04.2019	J41.0	541201579	Зайнутдинов Р. Р. (Терапия)	47741	[541201] ГБУЗ п. Свердловски
					14.01.2019	D18.0	5411011218	Зайнутдинов Р. Р. (Терапия)	1021	[541101] ГБУЗ Озерская амбу
					08.04.2019	G43.9	54050112307	Сафина С. У. (Хирургия)	38412	[540501] ГБУЗ больница п. Б.
					26.04.2019	C50.9	5412012086	Жигулина В. М. (Терапия)	51444	[541201] ГБУЗ п. Свердловски

Рисунок 3.15 – Отчет «Детальный отчет по процессу постановки онкологического диагноза»

Детальный отчет отображает всех пациентов с нарушением сроков по одному из выбранных этапов. Для отображения информации по всем

пройденным этапам по одному пациенту необходимо нажать на его ФИО в форме отчета (рисунок 3.16).

ЕГИСЗ Московской области								
Данные направления								
Диагноз	Тип направления	Дата направления	Диагноз	Номер направления	ФИО направившего врача	ППУ прикрепления(направившее)	Дата создания ячейки расписания	Дата приема
C60.0	На консультацию	01.03.2019	C60.0	37110136711	Золотарева Л. А. (Средний медицинский персонал. Лечебное дело)	[371101] ГБУЗ МО "МОБ им. проф. Розанова В. Н." (Поликлиника №11)	01.03.2019 10:48:47	11.03.2019 11:15:00
	На диагностику	11.03.2019	C06.0	37010118-30299	Турлугулова Н. Б. (Онкология)	[370101] ГБУЗ МО "МОБ им. проф. Розанова В. Н."	11.03.2019 11:44:01	11.04.2019 12:30:00
	На диагностику	11.03.2019	C06.0	37010118-30300	Турлугулова Н. Б. (Онкология)	[370101] ГБУЗ МО "МОБ им. проф. Розанова В. Н."	11.03.2019 11:46:25	11.04.2019 12:45:00

Рисунок 3.16 – Отчет «Детальный отчет по движению пациента с онкологическим диагнозом»

Также пользователь может выгрузить отчет в документ формата: XML-файл с данными отчета, CSV (с разделителями-запятыми), PDF, MHTML (веб-архив), Excel, TIFF-файл, Word (рисунок 3.17).

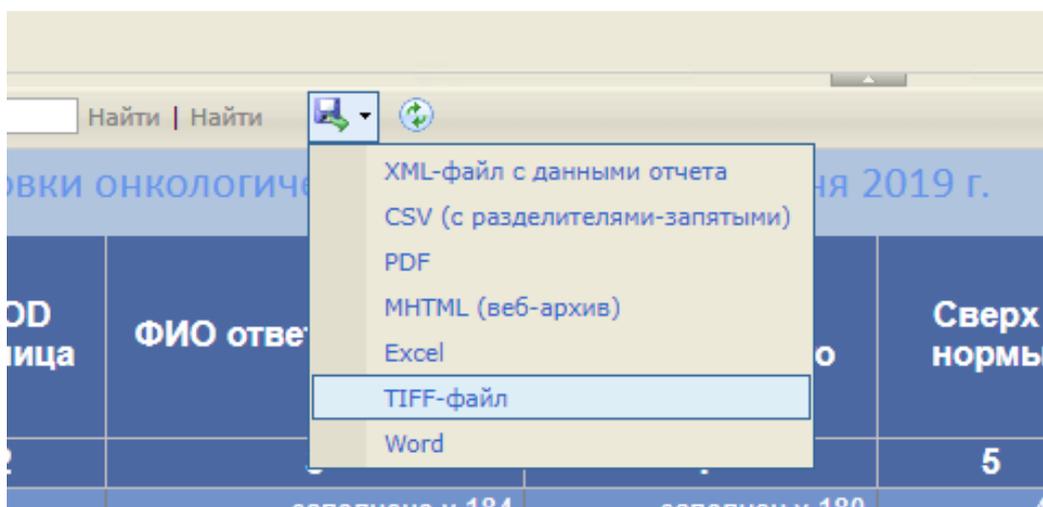


Рисунок 3.17 – Выгрузка отчетной формы

Таким образом, в результате тестирования было выявлено, что все отчетные формы функционируют правильно, ошибок при работе с мониторингами не выявлено. Разработанный автоматизированный мониторинг решает все поставленные задачи.

3.4 Целесообразность и экономическая эффективность

Схема SWOT-анализа включает следующие этапы:

- выявление: сильных сторон разработки; слабых сторон разработки; возможностей разработки; угроз;
- формирование и анализ SWOT-матрицы;
- выработка заключения о перспективности разработки.

В таблице 3.5 приведена SWOT-матрица для анализа сильных и слабых сторон автоматизированного мониторинга.

Таблица 3.5 – SWOT-матрица

Сильные стороны	Возможности		Угрозы		Итого
	Перенос на другие регионы	Совершенство разработки	Появление конкурентов	Быстрое устаревание	
Низкая стоимость разработки	++	0	++	0	+4
Обеспечение сопровождения	+	+	+	+	+4
Оперативная обработка данных	0	+	0	++	+3
Удобная форма представляемых результатов	++	+	0	0	+3
Централизованное хранение данных	+	++	+	++	+6
Итого	+6	+5	+4	+5	+20
Слабые стороны					
Сбои на сервере	0	0	0	-	-1
Некорректная работа врачей в системе	0	0	0	-	-1
Итого	0	0	0	-2	-2
Общий итог	+6	+5	+4	+3	+18

Проанализировав полученную SWOT-матрицу, можно сделать выводы, что самые важные сильные стороны в достаточной степени важны. Наиболее важным достоинством является централизованное хранение данных. В дальнейшем необходимо обращать особое внимание на обеспечение и расширение этой стороны разработки.

Все выделенные слабые стороны разработки являются одинаково опасными.

Из рассмотренных возможностей более реальной представляется возможность переноса автоматизированного мониторинга на другие регионы.

Наиболее опасной угрозой представляется быстрое устаревание разработки, но при сложившихся условиях это не столь существенная угроза. Появление конкурентов представляется маловероятным вследствие наличия сильных сторон разработки.

Заключение о перспективности разработки. Разработка имеет перспективы, так как у нее имеется ряд очень важных достоинств.

Выводы по третьему разделу

В третьем разделе была рассмотрена информационная модель, охарактеризована база данных и результатная информация. Спроектирован автоматизированный мониторинг процесса постановки онкологического диагноза, состоящий из четырех отчетных форм. Также, была обоснована целесообразность и экономическая эффективность разработки.

ЗАКЛЮЧЕНИЕ

В процессе выполнения выпускной квалификационной работы был проанализирован Приказ Министерства Здравоохранения от 04.07.2017 г. №379н, в котором внесены изменения в порядок оказания медицинской помощи населению по профилю «онкология», утвержденный приказом ведомства от 15 ноября 2012 г. №915н. и установлены предельные сроки диагностики онкологических заболеваний и оказания специализированной медицинской помощи больным.

Были выполнены следующие задачи:

- проведен анализ предметной области;
- проведен анализ существующего состояния;
- создана модель «КАК ЕСТЬ»;
- изучены аналогичные информационные системы;
- обоснована актуальность и значимость предмета проектирования;
- проведен анализ стратегии автоматизации;
- создана модель «КАК ДОЛЖНО БЫТЬ»;
- проведен анализ базы данных и спроектированы временные таблицы;
- разработаны 4 отчетные формы;
- выявлена целесообразность разработки информационной системы с

экономической точки зрения.

Изучив деятельность медицинских организаций, для реализации автоматизированного мониторинга был выбран следующий ряд программных продуктов: PCУБД Microsoft SQL Server, программная серверная система SQL Server Reporting Services (SSRS) и построитель отчетов Microsoft SQL Server Report Builder.

Исходя из выявленных требований, был создан автоматизированный мониторинг процесса постановки онкологического диагноза на основе слабосвязанных медицинских данных. Данный программный продукт

разработан непосредственно под особенности функционала медицинских организаций, использующих медицинскую информационную систему ТрастМед.

Разработанный автоматизированный мониторинг внедрен в практическую деятельность ООО «СофтТраст», о чем свидетельствует акт о внедрении.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Игрунова, С. В. Реинжиниринг бизнес-процессов АРМ логиста на примере ООО Торговый Дом «Малахов +» [Текст] / С. В. Игрунова, Е. В. Нестерова, И. В. Свиридова, А. К. Кичигина, Н. Е. Мандрыкина, М. А. Лифинцев. – Саратов: Электронное научно-практическое периодическое издание «Экономика и социум» Институт управления и социально-экономического развития, 2016. – 125 с.
2. Елизарова, О. И. Организация производства и менеджмент [Текст] / О. И. Елизарова - Москва: Московский гос. ун-т печати, 2010. – 134 с.
3. Карминский, А. М. Информационные системы в экономике. Методология создания [Текст] / А. М. Карминский, Б. В. Черников. – М.: Финансы и статистика, 2011. – 330 с.
4. Балдин, К. В. Информационные системы в экономике: Учебное пособие [Текст] / К. В. Балдин. - М.: НИЦ ИНФРА-М, 2013. - 218 с.
5. Дубейковский, В. И. Практика функционального моделирования с AllFusion Process Modeler 4.1. (VPwin) Где? Зачем? Как? [Текст] / В.И. Дубейковский. – М.: ДИАЛОГ-МИФИ, 2014. – 464 с.
6. Дистанционное образование. Современные информационные технологии в коммерческой деятельности. [Электронный ресурс]. Режим доступа: <http://sga-help.ru/>, свободный.
7. Попов, Д. И. Информационные технологии. Базы данных: Учебное пособие [Текст] / Д.И. Попов, Е.Д. Попова - Москва: Московский гос. ун-т печати, 2009. - 117 с.
8. Михеев, Р. Н. MS SQL Server 2005 для администраторов [Текст] / Р. Н. Михеев – СПб.: БХВ-Петербург, 2006. - 544 с.
9. Голицына, О. Л. Базы данных [Текст] / О. Л. Голицына, Н. В. Максимов, И. И. Попов. - М.: Форум, 2015. - 400 с.

10. Тернстрем, Т. Microsoft SQL Server 2008. Разработка баз данных. Учебный курс Microsoft [Текст] / Т. Тернстрем. - М.: Русская Редакция, 2011. - 293 с.
11. Об утверждении Порядка оказания медицинской помощи населению по профилю "онкология" [Текст]: Приказ Министерства здравоохранения Российской Федерации от 15 ноября 2012 г. № 915н // зарегистрирован Министерством юстиции Российской Федерации 17 апреля 2013 г., регистрационный № 28163
12. Баранова, Е.Н. Информационная безопасность и защита информации [Текст] / Е.Н. Баранова, Бабаши А.Н. - М., 2014. - 256 с.
13. Лобел, Л. Разработка приложений на основе Microsoft SQL Server 2008 [Текст] / Л. Лобел, Э. Дж. Браст, С. Форте: Пер. с англ. – М.: Русская Редакция; СПб.: БХВ-Петербург, 2010. – 1024 с.
14. Билл Карвин, Программирование баз данных SQL. Типичные ошибки и их устранение [Текст] / Б. Карвин. – М., 2014. - 336 с.
15. Вигерс, Б. М. Разработка требований к программному обеспечению [Текст] / Б. М. Вингерс. - М., 2014. – 736 с.
16. Маклаков, С. В. ВРwin и Егwin. Case – средства разработки информационных систем [Текст] / С.В. Маклаков – Москва: ДИАЛОГ – МИФИ, 2013. – 230 с.
17. Справочник по Transact-SQL [Электронный ресурс], - Режим доступа: <https://docs.microsoft.com/ru-ru/sql/t-sql/language-reference?view=sql-server-2017>, свободный.
18. Учитель, Ю. Г. SWOT-анализ и синтез - основа формирования стратегии организации [Текст] / Ю. Г. Учитель, М.Ю. Учитель. - Москва: Либроком, 2010. - 328 с.
19. Петрунин, Ю. Ю. Информационные технологии анализа данных: Учебное пособие [Текст] / Ю. Ю. Петрунин. - Москва: КДУ, 2010. - 292. с.

20. Советов, Б. Я. Базы данных: теория и практика: Учебник для бакалавров [Текст] / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовский. - Москва: Юрайт, 2013 - 463с.

21. Что такое службы SQL Server Reporting Services (SSRS)? [Электронный ресурс], - Режим доступа: <https://docs.microsoft.com/ru-ru/sql/reporting-services/create-deploy-and-manage-mobile-and-paginated-reports?view=sql-server-2017>, свободный.

22. ГОСТ 7.32-2001. Отчет о научно-исследовательской работе. Структура и правила оформления [Текст]. - Взамен ГОСТ 7.32 - 91; Введ. 01.07.2002. - М: Стандартинформ, 2008. – 20 с. - (Система стандартов по информации, библиотечному и издательскому делу).

23. ГОСТ 7.1—2003. Библиографическая запись. Библиографическое описание. Общие требования и правила составления [Текст]. - Взамен ГОСТ 7.1—84, ГОСТ 7.16—79, ГОСТ 7.18—79, ГОСТ 7.34—81, ГОСТ 7.40—82; Введ. 01.07.2004. - М: Стандартинформ, 2006. – 52 с. - (Система стандартов по информации, библиотечному и издательскому делу).

24. ГОСТ 2.105-95. Общие требования к текстовым документам [Текст]. - Взамен ГОСТ 2.105-79. ГОСТ 2.906-71; Введ. 01.06.1996. - Минск: Межгосударственный совет по стандартизации, метрологии и сертификации, 2007. – 30 с. - (Единая система конструкторской документации).

25. Култыгин, О. П. Администрирование баз данных СУБД MS SQL Server [Текст] / О. П. Култыгин. – Москва: Московская финансово-промышленная академия, 2012. – 232 с.

26. Корпоративный сайт ООО «СофтТраст» [Электронный ресурс], - Режим доступа: <https://softtrust.ru/>, свободный.

ПРИЛОЖЕНИЕ А

Листинг скрипта для формирования временных таблиц

```
declare @d1 date = convert(date,'01.01.2017',104)
declare @d2 date = convert(date,getdate());

if exists(select * from sys.tables where name = 'tmp_dir_onko_last_new') drop table
tmp_dir_onko_last_new
if exists(select * from sys.tables where name = 'tmp_zap_onko_last_new') drop table
tmp_zap_onko_last_new
if exists(select * from sys.tables where name = 'tmp_dir_diag_last_new') drop table
tmp_dir_diag_last_new
if exists(select * from sys.tables where name = 'tmp_zap_diag_last_new') drop table
tmp_zap_diag_last_new

if exists(select * from sys.tables where name = 'tmp_dir_onko_new') select * into
tmp_dir_onko_last_new from tmp_dir_onko_new
if not exists(select * from sys.tables where name = 'tmp_dir_onko_new')
create table [dbo].[tmp_dir_onko_last_new](
    [n] [bigint] NULL,
    [Num] [varchar](50) NOT NULL,
    [FAMILY] [varchar](50) NOT NULL,
    [NAME] [varchar](50) NOT NULL,
    [OT] [varchar](50) NOT NULL,
    [BD] [datetime] NOT NULL,
    [S_POL] [varchar](16) NOT NULL,
    [N_POL] [varchar](50) NOT NULL,
    [rf_MKABID] [int] NOT NULL,
    [DateDir] [datetime] NOT NULL,
    [rf_DocPRVDID] [int] NOT NULL,
    [rf_MKBID] [int] NOT NULL,
    [DS] [varchar](10) NOT NULL,
    [PRVS_NAME] [varchar](100) NOT NULL,
    [rf_LPUSenderId] [int] NOT NULL,
    [rf_LPUID] [int] NOT NULL,
    [tap_MKBID] [int] NULL,
    [tap_DateTAP] [datetime] NULL,
    [tapC_MKBID] [int] NULL,
    [tapC_DateTAP] [datetime] NULL,
    [DateDirHosp] [datetime] NULL,
    [DatePlanHosp] [datetime] NULL,
    [LPUHosp] [int] NULL,
    [DateDirHosp1] [datetime] NULL,
    [DateFactHosp] [datetime] NULL,
    [LPUHospFact] [int] NULL,
    [CloseMkabDeath] [varchar](10) NULL,
    [Isk1] [int] NULL,
    [isNew] [int] NULL
) on [primary]

if exists(select * from sys.tables where name = 'tmp_zap_onko_new') select * into
tmp_zap_onko_last_new from tmp_zap_onko_new
if not exists(select * from sys.tables where name = 'tmp_zap_onko_new')
create table [dbo].[tmp_zap_onko_last_new](
    [n] [bigint] NULL,
    [FAMILY] [varchar](50) NOT NULL,
    [NAME] [varchar](50) NOT NULL,
    [OT] [varchar](50) NOT NULL,
    [BD] [datetime] NOT NULL,
```

```

        [DateDir] [datetime] NOT NULL,
        [dirDoc] [int] NOT NULL,
        [DateTimeCreate] [datetime] NOT NULL,
        [Begin_Time] [datetime] NOT NULL,
        [rf_DocPRVDID] [int] NOT NULL,
        [rf_TAPID] [int] NOT NULL,
        [VisitStatus] [int] NOT NULL,
        [PRVS_NAME] [varchar](100) NOT NULL,
        [V_FIO] [varchar](37) NULL,
        [MKABID] [int] NOT NULL,
        [DS] [varchar](10) NOT NULL,
        [canceldate] [datetime] NULL,
        [rf_LPUSenderID] [int] NOT NULL,
        [rf_LPUID] [int] NOT NULL,
        [isNew] [int] NULL
    ) on [primary]

if exists(select * from sys.tables where name ='tmp_dir_diag_new') select * into
tmp_dir_diag_last_new from tmp_dir_diag_new
if not exists(select * from sys.tables where name ='tmp_dir_diag_new')
create table [dbo].[tmp_dir_diag_last_new](
    [n] [bigint] NULL,
    [FAMILY] [varchar](50) NOT NULL,
    [NAME] [varchar](50) NOT NULL,
    [OT] [varchar](50) NOT NULL,
    [BD] [datetime] NOT NULL,
    [Date_Direction] [datetime] NOT NULL,
    [Begin_Time] [datetime] NULL,
    [DateRasp] [datetime] NOT NULL,
    [DateDir] [datetime] NOT NULL,
    [LaboratoryResearchID] [int] NOT NULL,
    [IsPerformed] [bit] NOT NULL,
    [isComplete] [bit] NOT NULL,
    [MKABID] [int] NOT NULL,
    [isNew] [int] NOT NULL
) on [primary]

if exists(select * from sys.tables where name ='tmp_zap_diag_new') select * into
tmp_zap_diag_last_new from tmp_zap_diag_new
if not exists(select * from sys.tables where name ='tmp_zap_diag_new')
create table [dbo].[tmp_zap_diag_last_new](
    [n] [bigint] NULL,
    [FAMILY] [varchar](50) NOT NULL,
    [NAME] [varchar](50) NOT NULL,
    [OT] [varchar](50) NOT NULL,
    [BD] [datetime] NOT NULL,
    [LaboratoryResearchID] [int] NOT NULL,
    [DateTimeCreate] [datetime] NULL,
    [Begin_Time] [datetime] NULL,
    [rf_DocPRVDID] [int] NULL,
    [rf_TAPID] [int] NULL,
    [VisitStatus] [int] NULL,
    [PRVS_NAME] [varchar](100) NOT NULL,
    [V_FIO] [varchar](37) NULL,
    [rf_MKABID] [int] NULL,
    [canceldate] [datetime] NULL,
    [rf_LPUSenderID] [int] NOT NULL,
    [rf_LPUID] [int] NOT NULL,
    [isNew] [int] NULL
) on [primary]

if exists(select * from sys.tables where name ='tmp_dir_onko_new') drop table
tmp_dir_onko_new
if exists(select * from sys.tables where name ='tmp_zap_onko_new') drop table
tmp_zap_onko_new

```

```

if exists(select * from sys.tables where name = 'tmp_dir_diag_new') drop table
tmp_dir_diag_new
if exists(select * from sys.tables where name = 'tmp_zap_diag_new') drop table
tmp_zap_diag_new

/*Направления на консультацию*/
select row_number() over(partition by FAMILY,NAME,OT,BD order by DateDir) as n /*Количество
направлений на человека*/
,Num,FAMILY,NAME,OT,BD,S_POL,N_POL,rf_MKABID,DateDir,rf_DocPRVDID,rf_MKABID,DS,PRVS_NA
ME,rf_LPUSenderID,rf_LPUID
,min(case when n_tap = 1 then tap_MKABID else null end) as tap_MKABID
,min(case when n_tap = 1 then tap_DateTAP else null end) as tap_DateTAP
,min(case when n_tapC = 1 then tapC_MKABID else null end) as tapC_MKABID
,min(case when n_tapC = 1 then tapC_DateTAP else null end) as tapC_DateTAP
,min(case when DateDirHosp1 is null
then case when n_hosp = 1 then DateDirHosp else null end
else case when n_hosp2 = 1 then DateDirHosp else null end end) as
DateDirHosp
,min(case when DateDirHosp1 is null
then case when n_hosp = 1 then DatePlanHosp else null end
else case when n_hosp2 = 1 then DatePlanHosp else null end end) as
DatePlanHosp
,min(case when DateDirHosp1 is null
then case when n_hosp = 1 then LPUHosp else null end
else case when n_hosp2 = 1 then LPUHosp else null end end) as LPUHosp
,min(case when n_hosp_fact = 1 then DateDirHosp1 else null end) as DateDirHosp1
,min(case when n_hosp_fact = 1 then DateFactHosp else null end) as DateFactHosp
,min(case when n_hosp_fact = 1 then LPUHospFact else null end) as LPUHospFact
,max(CloseMkabDeath) as CloseMkabDeath
,max(Iskl) as Iskl
,min(isNew) as isNew
into tmp_dir_onko_new
from (
select dir.Num,dir.FAMILY,dir.NAME,dir.OT,dir.BD,dir.S_POL,dir.N_POL
,dir.rf_MKABID /*МКАБ из направления*/
,dir.DateDir /*Дата направления*/
,dir.rf_DocPRVDID /*Врач выписавший направление*/
,dir.rf_MKABID /*Диагноз из направления*/
,dir.DS /*Код диагноза из направления*/
,dir.PRVS_NAME /*Специальность врача к которому направили*/
,dir.rf_LPUSenderID /*Направлен из ЛПУ*/
,dir.rf_LPUID /*Направлен в ЛПУ*/
,mkab.MKABID /*МКАБы по пациенту*/
/*Тап с хорошим диагнозом, далее выбираем с минимальной датой*/
,row_number() over(partition by
dir.FAMILY,dir.NAME,dir.OT,dir.BD,dir.DateDir order by isnull(tap.DateTAP,getdate()) asc) as
n_tap
,tap.rf_MKABID as tap_MKABID
,tap.DateTAP as tap_DateTAP
/*Тап с плохим диагнозом, далее выбираем с минимальной датой*/
,row_number() over(partition by
dir.FAMILY,dir.NAME,dir.OT,dir.BD,dir.DateDir order by isnull(tapC.DateTAP,getdate()) asc)
as n_tapC
,tapC.rf_MKABID as tapC_MKABID
,tapC.DateTAP as tapC_DateTAP
/*Направление на госпитализацию, далее берем минимальную дату если нет
фактической госпитализации, или максимально ближайшую к фактической госпитализации, если она
есть*/
,row_number() over(partition by
dir.FAMILY,dir.NAME,dir.OT,dir.BD,dir.DateDir order by isnull(hosp.DateDirHosp,getdate())
asc) as n_hosp
,row_number() over(partition by
dir.FAMILY,dir.NAME,dir.OT,dir.BD,dir.DateDir order by
isnull(hosp_fact.DateDirHosp,getdate()) asc, isnull(hosp.DateDirHosp,getdate()) desc) as
n_hosp2

```

```

        ,hosp.DateDirHosp /*Дата направления*/
        ,hosp.DatePlanHosp /*Дата планируемой госпитализации*/
        ,hosp.rf_LPUIID as LPUHosp /*Направлен в ЛПУ*/
        /*Фактическая госпитализация*/
        ,row_number() over(partition by
dir.FAMILY,dir.NAME,dir.OT,dir.BD,dir.DateDir order by
isnull(hosp_fact.DateDirHosp,getdate()) asc) as n_hosp_fact
        ,hosp_fact.DateDirHosp as DateDirHosp1 /*Дата направления*/
        ,hosp_fact.DateFactHosp as DateFactHosp /*Дата фактической
госпитализации*/
        ,hosp_fact.LPU as LPUHospFact /*Фактически госпитализировавшее ЛПУ*/
        ,isnull(closemkab.Code,0) as CloseMkabDeath /*МКАБ закрыт по причине
смерти*/
        ,(case when iskl.dat is not null then 1 else 0 end) as Iskl /*Исключить
из отчета (по просьбе МИАЦ)*/
        ,(case when onko_last.rf_MKABID is null then 1 else 0 end) as isNew /*1
- новый пациент 0 - старый пациент*/
        from (
                select row_number() over(partition by
dir.FAMILY,dir.NAME,dir.OT,dir.BD order by dir.Date) as n
                ,dir.Num,dir.FAMILY,dir.NAME,dir.OT,dir.BD,dir.S_POL,dir.N_POL,dir.rf_MKABID
                ,dir.Date as DateDir,dir.rf_DocPRVDID,dir.rf_MKABID
                ,oms_MKB.DS
                ,Oms_PRVS.PRVS_NAME
                ,dir.rf_LPUsenderID,dir.rf_LPUIID
                from [cod_MO].dbo.hlt_Direction dir WITH(NOLOCK) /*Направления*/
                inner join ( /*Берем максимальный ИД направления за день
по человеку*/
                        select max(dir.directionID) as mdirectionID
                        from [cod_MO].dbo.hlt_Direction dir WITH(NOLOCK)
                                inner join [cod_MO].dbo.Oms_PRVS
                                and Oms_PRVS.PRVS_NAME like
                                '%онколог%' /*Специальность врача, к которому направляется человек*/
                                inner join [cod_MO].dbo.hlt_TypeDirection
                                and hlt_TypeDirection.Name = 'На
консультацию' /*Отбираем направления именно на консультацию*/
                                inner join [cod_MO].dbo.hlt_DocPRVD doc
                                and hlt_TypeDirection.Name = 'На консультацию'
                                inner join [cod_MO].dbo.oms_PRVS prvs_send
                                and prvs_send.PRVS_NAME not like
                                '%онколог%'/*Специальность направившего врача не онколог*/
                                where dir.Date > @d1 /*Все направления начиная с
01-01-2017*/
                                and dir.rf_MKABID > 0
                        group by
dir.FAMILY,dir.NAME,dir.OT,dir.BD,cast(dir.Date as date)
                        )dir_max on directionID = mdirectionID
                inner join [cod_MO].dbo.Oms_PRVS WITH(NOLOCK) on PRVSID =
dir.rf_PRVSID
                and Oms_PRVS.PRVS_NAME like '%онколог%'
                /*Специальность врача, к которому направляется человек*/
                inner join [cod_MO].dbo.hlt_TypeDirection WITH(NOLOCK) on
TypeDirectionID = rf_TypeDirectionID
                and hlt_TypeDirection.Name = 'На консультацию'
                /*Отбираем направления именно на консультацию*/
                inner join [cod_MO].dbo.hlt_DocPRVD doc WITH(NOLOCK) on
doc.DocPRVDID = dir.rf_DocPRVDID /*Направивший врач*/
                inner join [cod_MO].dbo.oms_PRVS prvs_send WITH(NOLOCK) on
prvs_send.PRVSID = doc.rf_PRVSID

```

```

                                and prvs_send.PRVS_NAME not like
'%онколог%' /*Специальность направившего врача не онколог*/
                                inner join [cod_MO].dbo.oms_MKB WITH(NOLOCK) on MKBID =
dir.rf_MKBID /*Диагнозы*/
                                )dir

                                left join tmp_dir_onko_last_new dir2 WITH(NOLOCK) on dir.FAMILY =
dir2.FAMILY and dir.NAME = dir2.NAME and dir.OT = dir2.OT and dir.BD = dir2.BD
                                and dir.n = dir2.n-1 /*Следующее направление*/

                                inner join [cod_MO].dbo.rpt_mn_person_mkab m WITH(NOLOCK) on
m.rf_mkabid = dir.rf_MKABID /*МКАБ из направления*/
                                inner join [cod_MO].dbo.rpt_mn_person_mkab person WITH(NOLOCK) on
person.rf_mn_personid = m.rf_mn_personid /*Все МКАБ по пациенту*/
                                inner join [cod_MO].dbo.hlt_MKAB mkab WITH(NOLOCK) on mkab.MKABID =
person.rf_mkabid
                                and mkab.MKABID > 0
                                left join [cod_MO].dbo.hlt_ReasonCloseMKAB closemkab WITH(NOLOCK) on
closemkab.ReasonCloseMKABID = mkab.rf_ReasonCloseMKABID
                                and closemkab.Code = '1' /*Причина закрытия МКАБА - смерть*/

                                left join ( /*Госпитализация из СУПП в отделение с профилем онкология*/
                                select dir.FAM,dir.IM,dir.OT,dir.DR
                                ,dir.DirDate as DateDirHosp
                                ,dir.DateTimeFact as DateFactHosp
                                ,lpu.LPUID as LPU
                                from [TM263].dbo.oms_hs_hospital dir WITH(NOLOCK)
                                inner join [TM263].dbo.oms_kl_DepartmentProfile
WITH(NOLOCK) on kl_DepartmentProfileID = dir.rf_kl_DepartmentProfileID
                                and oms_kl_DepartmentProfile.NAME like '%онколог%'
                                left join [cod_MO].dbo.oms_LPU lpu WITH(NOLOCK) on
lpu.MCod = dir.LPU_1
                                )hosp_fact on hosp_fact.FAM = mkab.FAMILY and hosp_fact.IM = mkab.NAME
and hosp_fact.OT = mkab.OT and hosp_fact.DR = mkab.DATE_BD
                                and hosp_fact.DateDirHosp >= dir.DateDir
                                and hosp_fact.DateDirHosp < isnull(dir2.DateDir, getdate())
                                left join ( /*Направление на госпитализацию в отделение с профилем
онкология*/
                                select dir.rf_MKABID
                                ,dir.Date as DateDirHosp
                                ,dir.DatePlan as DatePlanHosp
                                ,dir.rf_LPUID
                                from [cod_MO].dbo.hlt_direction dir WITH(NOLOCK)
                                inner join [cod_MO].dbo.oms_kl_DepartmentProfile
WITH(NOLOCK) on kl_DepartmentProfileID = dir.rf_kl_DepartmentProfileID
                                and oms_kl_DepartmentProfile.NAME like '%онколог%'
                                inner join [cod_MO].dbo.hlt_TypeDirection WITH(NOLOCK) on
TypeDirectionID = dir.rf_TypeDirectionID
                                and hlt_TypeDirection.Name = 'На госпитализацию'
                                )hosp on hosp.rf_MKABID = mkab.MKABID
                                and hosp.DateDirHosp >= dir.DateDir
                                and hosp.DateDirHosp < isnull(dir2.DateDir, getdate())
                                and hosp.DateDirHosp <= isnull(hosp_fact.DateDirHosp, getdate())
                                left join ( /*Тапы по человеку с плохими диагнозами*/
                                select rf_MKABID,DateTAP,rf_MKBID
                                from [cod_MO].dbo.hlt_TAP WITH(NOLOCK)
                                inner join [cod_MO].dbo.hlt_DocPRVD WITH(NOLOCK) on
DocPRVDID = hlt_TAP.rf_DocPRVDID
                                inner join [cod_MO].dbo.Oms_PRVS WITH(NOLOCK) on PRVSID =
hlt_DocPRVD.rf_PRVSID
                                and Oms_PRVS.PRVS_NAME like '%онколог%'
                                inner join [cod_MO].dbo.Oms_MKB WITH(NOLOCK) on MKBID =
hlt_TAP.rf_MKBID
                                and MKBID > 0
                                and DS like 'C%' /*Диганоз злокачественный*/

```

```

inner join [cod_MO].dbo.oms_kl_VisitResult WITH(NOLOCK) on
kl_VisitResultID = hlt_TAP.rf_kl_VisitResultID
and kl_VisitResultID > 0
and oms_kl_VisitResult.CODE_Federal not in
('304','314')--Лечение продолжено--Динамическое наблюдение
)tapC on tapC.rf_MKABID = mkab.MKABID
and tapC.DateTAP >= dir.DateDir
and tapC.DateTAP < isnull(dir2.DateDir, getdate())
and tapC.DateTAP <= isnull(hosp.DatePlanHosp, getdate())
and tapC.DateTAP <= isnull(hosp_fact.DateFactHosp, getdate())
left join ( /*Тапы по человеку с положительными изменениями диагноза*/
select rf_MKABID,DateTAP,rf_MKABID
from [cod_MO].dbo.hlt_TAP WITH(NOLOCK)
inner join [cod_MO].dbo.hlt_DocPRVD WITH(NOLOCK) on
DocPRVDID = hlt_TAP.rf_DocPRVDID
inner join [cod_MO].dbo.Oms_PRVS WITH(NOLOCK) on PRVSID =
hlt_DocPRVD.rf_PRVSID
and Oms_PRVS.PRVS_NAME like '%онколог%'
inner join [cod_MO].dbo.Oms_MKB WITH(NOLOCK) on MKBID =
rf_MKABID
and MKBID > 0
and DS not like 'C%' and DS not like 'D3[789]%' and
DS not like 'D4[0-8]%' and DS <> 'Z03.1' /*Диганоз доброкачественный*/
inner join [cod_MO].dbo.oms_kl_VisitResult WITH(NOLOCK) on
kl_VisitResultID = rf_kl_VisitResultID
and kl_VisitResultID > 0
and oms_kl_VisitResult.CODE_Federal not in
('304','314')--Лечение продолжено--Динамическое наблюдение
)tap on tap.rf_MKABID = mkab.MKABID
and tap.DateTAP >= dir.DateDir
and tap.DateTAP < isnull(dir2.DateDir, getdate())
and tap.DateTAP > isnull(tapC.DateTAP, dir.DateDir)
and tap.DateTAP <= isnull(hosp.DatePlanHosp, getdate())
and tap.DateTAP <= isnull(hosp_fact.DateFactHosp, getdate())
left join tmp_dir_onko_last_new onko_last WITH(NOLOCK) on dir.rf_MKABID
= onko_last.rf_MKABID
and dir.DateDir = onko_last.DateDir and dir.rf_DocPRVDID =
onko_last.rf_DocPRVDID /*left join с этой же таблицей чтобы узнать новый пациент или нет*/
/*Исключение пациентов*/
left join [cod_MO].dbo.V_hlt_DocPRVD doc WITH(NOLOCK) on
dir.rf_DocPRVDID = doc.DocPRVDID
left join tmp_dir_iskl iskl WITH(NOLOCK) on iskl.dat = dir.DateDir and
iskl.fio = doc.V_FIO and iskl.ds = dir.ds and iskl.num = dir.num
)t
group by Num,FAMILY,NAME,OT,BD,S_POL,N_POL,rf_MKABID,DateDir,rf_DocPRVDID,rf_MKABID
,DS,PRVS_NAME,rf_LPUSenderID,rf_LPUID

/*Информация о записях на консультацию*/
select *
into tmp_zap_onko_new
from (
/*Активные ячейки расписания на консультацию*/
select row_number() over(partition by dir.FAMILY,dir.NAME,dir.OT,dir.BD,dir.DateDir
order by dvt.DateTimeCreate) as n /*Количество консультаций по направлению*/
,dir.FAMILY,dir.NAME,dir.OT,dir.BD
,dir.DateDir /*Дата направления на консультацию*/
,dir.rf_DocPRVDID as dirDoc /*Врач выписавший направление*/
,dvt.DateTimeCreate /*Дата и время создания записи на прием*/
,dtb.Begin_Time /*Дата и время консультации*/
,dtb.rf_DocPRVDID /*Врач проводящий консультацию*/
,dvt.rf_TAPID /*Тап созданный на консультацию*/
,dvt.VisitStatus /*Статус визита*/
,doc.V_PRVSName as PRVS_NAME /*Специальность врача проводящего консультацию*/
,doc.V_FIO /*ФИО врача проводящего консультацию*/
,mkab.MKABID /*МКАБ из посещения*/

```

```

,dir.DS /*Диагноз из направления*/
,null as canceldate /*null - запись не отменена*/
,dir.rf_LPUSenderID /*Направлен из ЛПУ (из направления на консультацию)*/
,dir.rf_LPUIID /*Направлен в ЛПУ (из направления на консультацию)*/
,max(case when onko_last.MKABID is null then 1 else 0 end) as isNew /*1 -
новый пациент 0 - старый пациент*/
from tmp_dir_onko_new dir WITH(NOLOCK) /*Временная таблица с направлениями на
консультацию*/
inner join [cod_MO].dbo.rpt_mn_person_mkab m WITH(NOLOCK) on m.rf_mkabid =
dir.rf_MKABID /*МКАБ из направления*/
inner join [cod_MO].dbo.rpt_mn_person_mkab person WITH(NOLOCK) on
person.rf_mn_personid = m.rf_mn_personid /*Все МКАБ по пациенту*/
inner join [cod_MO].dbo.hlt_MKAB mkab WITH(NOLOCK) on mkab.MKABID =
person.rf_mkabid
and mkab.MKABID > 0
inner join [cod_MO].dbo.hlt_DoctorVisitTable dvt WITH(NOLOCK) on dvt.rf_MKABID
= mkab.MKABID
and DateTimeCreate >= dir.DateDir /*Берем записи на прием дата создания
которых больше даты направления и (меньше плановой даты госпитализации либо меньше текущей
даты+30дней)*/
and dvt.DateTimeCreate <
isnull(dir.DatePlanHosp,isnull(dir.DateDirHosp1,dateadd(day,30,getdate()))))
and dvt.Flags <> 8
left join tmp_dir_onko_new dir2 WITH(NOLOCK) on dir.FAMILY = dir2.FAMILY and
dir.NAME = dir2.NAME and dir.OT = dir2.OT and dir.BD = dir2.BD
and dir.n = dir2.n-1 /*Если есть направление новее*/
and dvt.DateTimeCreate >= dir2.DateDir
inner join [cod_MO].dbo.hlt_DoctorTimeTable dtt WITH(NOLOCK) on
dtt.DoctorTimeTableID = dvt.rf_DoctorTimeTableID /*Режим работы врача*/
and dtt.DoctorTimeTableID > 0
and dir2.n is null /*Нет направления новее*/
inner join [cod_MO].dbo.hlt_DocBusyType dbt WITH(NOLOCK) on dbt.DocBusyTypeID
= dtt.rf_DocBusyType /*Тип занятости врача*/
and dbt.TypeBusy > 0
inner join [cod_MO].dbo.V_hlt_DocPRVD doc WITH(NOLOCK) on doc.DocPRVDID =
dtt.rf_DocPRVDID /*Врач*/
and doc.V_PRVSName like '%онколог%' /*Должность направившего врача
онколог*/
inner join [cod_MO].dbo.oms_Department dep WITH(NOLOCK) on dep.DepartmentID =
doc.rf_DepartmentID
and dep.DepartmentID > 0
and dep.rf_LPUIID > 0 /*Отделение врача*/
inner join [cod_MO].dbo.oms_LPU lpu_dir WITH(NOLOCK) on lpu_dir.LPUIID =
dir.rf_LPUIID /*Подразделение ЛПУ в которое направили*/
left join [cod_MO].dbo.oms_LPU mainl_dir WITH(NOLOCK) on (lpu_dir.rf_MainLPUIID
= mainl_dir.LPUIID and mainl_dir.LPUIID > 0) and mainl_dir.rf_MainLPUIID = 0 /*Главное ЛПУ в
которое направили*/
inner join [cod_MO].dbo.oms_LPU lpu WITH(NOLOCK) on lpu.LPUIID = dep.rf_LPUIID
/*Подразделение врача к которому была запись на консультацию*/
left join [cod_MO].dbo.oms_LPU mainl WITH(NOLOCK) on (lpu.rf_MainLPUIID =
mainl.LPUIID and mainl.LPUIID > 0) and mainl.rf_MainLPUIID = 0 /*Главное ЛПУ врача к которому
была запись на консультацию*/
left join tmp_zap_onko_last_new onko_last WITH(NOLOCK) on mkab.MKABID =
onko_last.MKABID
and dtt.Begin_Time = onko_last.Begin_Time
and dtt.rf_DocPRVDID = onko_last.rf_DocPRVDID /*left join с этой же
таблицей чтобы узнать новый пациент или нет*/
where isnull(mainl.LPUIID,lpu.LPUIID) = isnull(mainl_dir.LPUIID,lpu_dir.LPUIID)
/*консультация в ЛПУ в которое направили*/
group by dir.FAMILY,dir.NAME,dir.OT,dir.BD,dir.DateDir,dir.DS
,dvt.DateTimeCreate,dtt.Begin_Time,dtt.rf_DocPRVDID,dvt.Flags,dvt.rf_TAPID,dvt.Visits
tatus

```

```

,doc.V_PRVSName,doc.V_FIO,mkab.MKABID, dir.rf_LPUSenderID,
dir.rf_LPUID,dir.rf_DocPRVDID
union
/*Отмененные записи на консультацию*/
select * from (
select row_number() over (partition by
onko_last.FAMILY,onko_last.NAME,onko_last.OT,onko_last.BD,onko_last.Begin_Time order by
cancel.DateTimeCreate asc) as n /*Количество раз сколько отменяли запись на ячейку
расписания*/
,onko_last.FAMILY,onko_last.NAME,onko_last.OT,onko_last.BD
,onko_last.DateDir
,onko_last.dirDoc
,onko_last.DateTimeCreate
,onko_last.Begin_Time
,onko_last.rf_DocPRVDID
,onko_last.rf_TAPID
,onko_last.VisitStatus
,onko_last.PRVS_NAME
,onko_last.V_FIO
,onko_last.MKABID
,onko_last.DS
,cancel.DateTimeCreate as canceldate /*Дата отмены*/
,onko_last.rf_LPUSenderID
,onko_last.rf_LPUID
,0 as isNew
from tmp_zap_onko_last_new onko_last WITH(NOLOCK) /*Прошлая версия записей на
консультацию*/
inner join [cod_MO].dbo.tmp_DoctorVisitTable_cancel cancel WITH(NOLOCK)
on onko_last.MKABID = cancel.rf_MKABID /*таблица с отмененными записями*/
and cancel.Begin_Time = onko_last.Begin_Time /*Берем старые
записи которые не отменены, ищем в таблице отмененных записей совпадения по времени, мкабу,
и врачу*/
and cancel.DateTimeCreate >= onko_last.DateTimeCreate
and cancel.dtt_rf_DocPRVDID = onko_last.rf_DocPRVDID
and onko_last.canceldate is null
)asd where n = 1 /*Есть дубли если отменяли несколько раз одну и ту же запись,
поэтому берем самую ближайшую дату отмены к дате создания*/
union all
select * from tmp_zap_onko_last_new WITH(NOLOCK) where canceldate is not null /*Берем
все старые отмененные записи*/
)asd

/*Направления на диагностику*/
select *
into tmp_dir_diag_new
from (
select row_number() over(partition by
zap.FAMILY,zap.NAME,zap.OT,zap.BD,lr.LaboratoryResearchID order by zap.Begin_Time desc) as n
/*Берем n = 1 (ближайшая дата консультации к дате направления на диагностику) */
,zap.FAMILY,zap.NAME,zap.OT,zap.BD
,lr.Date_Direction /*Дата выдачи направления на диагностику*/
,dtt.Begin_Time /*Дата и время записи на диагностику*/
,zap.Begin_Time as DateRasp /*Дата консультации на которой выдали направление
на диагностику*/
,zap.DateDir as DateDir /*Дата направления на консультацию*/
,lr.LaboratoryResearchID /*ИД направления на диагностику*/
,r.IsPerformed /*Признак проведения исследования*/
,r.isComplete /*Признак выполнения*/
,lr.rf_MKABID as MKABID /*МКАБ из направления на исследование*/
,case when onko_last.MKABID is null then 1 else 0 end as isNew /*1 - новый
пациент 0 - старый пациент*/
from tmp_zap_onko_new zap WITH(NOLOCK) /*Таблица консультаций*/
left join tmp_zap_onko_new zap1 WITH(NOLOCK) on zap1.FAMILY+' '+zap1.NAME+'
'+zap1.OT+' '+convert(varchar,zap1.BD,104) = zap.FAMILY+' '+zap.NAME+' '+zap.OT+'
'+convert(varchar,zap.BD,104)

```

```

        and zap1.DateDir = zap.DateDir
        and zap1.n > zap.n
        and zap1.VisitStatus > -1
        and zap1.canceldate is null /*join с самой собой, чтобы найти
промежуток между какими консультациями было выписано направление на диагностику*/
        inner join [cod_MO].dbo.rpt_mn_person_mkab m WITH(NOLOCK) on m.rf_mkabid =
zap.MKABID /*МКАБ из направления*/
        inner join [cod_MO].dbo.rpt_mn_person_mkab person WITH(NOLOCK) on
person.rf_mn_personid = m.rf_mn_personid /*Все МКАБ по пациенту*/
        inner join [cod_MO].dbo.hlt_MKAB mkab WITH(NOLOCK) on mkab.MKABID =
person.rf_mkabid
        and mkab.MKABID > 0
        and zap.VisitStatus > -1 and zap.canceldate is null /*Консультация
состоялась*/

        inner join [cod_MO].dbo.lbr_LaboratoryResearch lr WITH(NOLOCK) on lr.rf_MKABID
= mkab.MKABID /*Лабораторное исследование*/
        and lr.DateCreate >= zap.Begin_Time
        and lr.DateCreate <= isnull(zap1.Begin_Time,getdate()) /*Дата создания
записи больше чем дата проведенной консультации и меньше (даты следующей консультации либо
текущей даты)*/

        inner join [cod_MO].dbo.oms_LPU lpu_dir WITH(NOLOCK) on lpu_dir.LPUID =
zap.rf_LPUID /*Подразделение ЛПУ в котором была консультация*/
        left join [cod_MO].dbo.oms_LPU mainl_dir WITH(NOLOCK) on (lpu_dir.rf_MainLPUID
= mainl_dir.LPUID and mainl_dir.LPUID > 0) and mainl_dir.rf_MainLPUID = 0 /*Главное ЛПУ в
котором была консультация*/
        inner join [cod_MO].dbo.oms_LPU lpu WITH(NOLOCK) on lpu.LPUID =
lr.rf_LPUSenderID /*Подразделение направившее на исследование*/
        left join [cod_MO].dbo.oms_LPU mainl WITH(NOLOCK) on (lpu.rf_MainLPUID =
mainl.LPUID and mainl.LPUID > 0) and mainl.rf_MainLPUID = 0 /*Главное ЛПУ направившее на
исследование*/

        inner join [cod_MO].dbo.v_hlt_DocPRVD doc WITH(NOLOCK) on doc.DocPRVDID =
lr.rf_DocPRVDID
        and doc.V_PRVName like '%онколог%' /*Должность направившего врача
онколог*/
        and (isnull(mainl.LPUID,lpu.LPUID) =
isnull(mainl_dir.LPUID,lpu_dir.LPUID) or lr.rf_DocPRVDID = zap.rf_DocPRVDID) /*ЛПУ где была
консультация = ЛПУ которое направило на диагностику или направивший врач = врач из
консультации*/

        inner join [cod_MO].dbo.lbr_Research r WITH(NOLOCK) on
r.rf_LaboratoryResearchGUID = lr.GUID /*Исследование*/
        left join [cod_MO].dbo.hlt_ActionSchedule sch WITH(NOLOCK) on sch.rf_DocTypeID
= lr.LaboratoryResearchID /*Планирование мероприятий*/
        and sch.DocTypeDefID = (select d.DocTypeDefID from
[cod_MO].dbo.x_DocTypeDef d WITH(NOLOCK) where d.headTable = 'lbr_LaboratoryResearch')
        left join [cod_MO].dbo.hlt_DoctorVisitTable dvt WITH(NOLOCK) on
dvt.DoctorVisitTableID = sch.rf_DoctorVisitTableID /*Посещение по направлению на
исследование*/
        and dvt.FLAGS <> 8
        and dvt.VisitStatus <> -1
        left join [cod_MO].dbo.hlt_DoctorTimeTable dtt WITH(NOLOCK) on
dtt.DoctorTimeTableID = dvt.rf_DoctorTimeTableID /*Расписание врача*/
        left join tmp_dir_diag_last_new onko_last WITH(NOLOCK) on
onko_last.LaboratoryResearchID = lr.LaboratoryResearchID /*left join с этой же таблицей
чтобы узнать новый пациент или нет*/
    )asd
where n = 1 /*Берем n = 1 (ближайшая дата консультации к дате направления на диагностику) */

/*Информация о записях на диагностику*/
select *
into tmp_zap_diag_new
from (

```

```

/*Активные ячейки расписания на диагностику*/
select row_number() over(partition by
dir.FAMILY,dir.NAME,dir.OT,dir.BD,dtf.rf_DocPRVDID order by dtf.DateTimeCreate) as n
/*Количество исследований у врача*/
,dir.FAMILY,dir.NAME,dir.OT,dir.BD
,lr.LaboratoryResearchID /*ИД направления на диагностику*/
,dtf.DateTimeCreate /*Дата и время создания записи на диагностику*/
,dtf.Begin_Time /*Дата и время начала диагностики*/
,dtf.rf_DocPRVDID /*Врач проводивший диагностику*/
,dtf.rf_TAPID /*Созданный ТАП на диагностике*/
,dtf.VisitStatus /*Статус посещения*/
,Oms_PRVS.PRVS_NAME /*Специальность врача*/
,doc.V_FIO /*ФИО врача*/
,dtf.rf_MKABID /*МКАБ из таблицы посещения*/
,null as canceldate
,lr.rf_LPUSenderID /*Направлен из ЛПУ (из направления на консультацию)*/
,lr.rf_LPUID /*Направлен в ЛПУ (из направления на консультацию)*/
,max(case when onko_last.rf_MKABID is null then 1 else 0 end) as isNew /*1 -
новый пациент 0 - старый пациент*/
from tmp_dir_diag_new dir WITH(NOLOCK)
inner join [cod_MO].dbo.lbr_LaboratoryResearch lr WITH(NOLOCK) on
lr.LaboratoryResearchID = dir.LaboratoryResearchID /*Лабораторное исследование*/
inner join [cod_MO].dbo.hlt_ActionSchedule sch WITH(NOLOCK) on
sch.rf_DocTypeID = lr.LaboratoryResearchID /*Планирование мероприятий*/
and sch.DocTypeDefID = (select d.DocTypeDefID from
[cod_MO].dbo.x_DocTypeDef d WITH(NOLOCK) where d.headTable = 'lbr_LaboratoryResearch')
inner join [cod_MO].dbo.hlt_DoctorVisitTable dtf WITH(NOLOCK) on
dtf.DoctorVisitTableID = sch.rf_DoctorVisitTableID /*Посещение*/
and dtf.Flags <> 8
inner join [cod_MO].dbo.hlt_DoctorTimeTable dtf WITH(NOLOCK) on
dtf.DoctorTimeTableID = dtf.rf_DoctorTimeTableId /*Расписание врача*/
inner join [cod_MO].dbo.v_hlt_DocPRVD doc WITH(NOLOCK) on doc.DocPRVDID =
dtf.rf_DocPRVDID /*Врач*/
and doc.rf_LPUDoctorID = 0 /*Потому что оборудование?*/
inner join [cod_MO].dbo.Oms_PRVS WITH(NOLOCK) on PRVSID = doc.rf_PRVSID
/*Специальность врача*/
left join tmp_zap_diag_last_new onko_last WITH(NOLOCK) on
onko_last.LaboratoryResearchID = lr.LaboratoryResearchID /*left join с этой же таблицей
чтобы узнать новый пациент или нет*/
group by dir.FAMILY,dir.NAME,dir.OT,dir.BD,lr.LaboratoryResearchID

,dtf.DateTimeCreate,dtf.Begin_Time,dtf.rf_DocPRVDID,dtf.Flags,dtf.rf_TAPID,dtf.VisitS
tatus
,Oms_PRVS.PRVS_NAME,doc.V_FIO,dtf.rf_MKABID,lr.rf_LPUSenderID, lr.rf_LPUID
union
/*Отмененные ячейки расписания*/
select * from (
Select row_number() over (partition by
onko_last.FAMILY,onko_last.NAME,onko_last.OT,onko_last.BD,onko_last.Begin_Time order by
cancel.DateTimeCreate asc) as n /*Количество раз сколько отменяли запись на ячейку
расписания*/
,onko_last.FAMILY,onko_last.NAME,onko_last.OT,onko_last.BD
,onko_last.LaboratoryResearchID
,onko_last.DateTimeCreate
,onko_last.Begin_Time
,onko_last.rf_DocPRVDID
,onko_last.rf_TAPID
,onko_last.VisitStatus
,onko_last.PRVS_NAME
,onko_last.V_FIO
,onko_last.rf_MKABID
,cancel.DateTimeCreate as canceldate /*Дата отмены*/
,onko_last.rf_LPUSenderID
,onko_last.rf_LPUID
,onko_last.isNew

```

```

        from tmp_zap_diag_last_new onko_last /*Прошлая версия записей на диагностику*/
            inner join [cod_MO].dbo.tmp_DoctorVisitTable_cancel cancel on
onko_last.rf_MKABID = cancel.rf_MKABID /*таблица с отмененными записями*/
            and cancel.Begin_Time = onko_last.Begin_Time /*Берем старые
записи которые не отменены, ищем в таблице отмененных записей совпадения по времени, мкабу,
и врачу*/
            and cancel.DateTimeCreate >= onko_last.DateTimeCreate
            and cancel.dtt_rf_DocPRVDID = onko_last.rf_DocPRVDID
            and onko_last.canceldate is null
    )asd where n = 1 /*Есть дубли если отменяли несколько раз одну и ту же запись,
поэтому берем самую ближайшую дату отмены к дате создания*/
    union all
    select * from tmp_zap_diag_last_new where canceldate is not null /*Берем все старые
отмененные записи*/
)asd

update tmp_zap_onko_new /*меняем n на -1 чтобы выделять отдельно отмененные записи*/
set n = -1
where canceldate is not null

update tmp_zap_diag_new /*меняем n на -1 чтобы выделять отдельно отмененные записи*/
set n = -1
where canceldate is not null or visitstatus = -1

```

ПРИЛОЖЕНИЕ Б

Листинг скрипта для отчета «Сводка по ЛПУ по процессу постановки онкологического диагноза»

```
/*Сводка по ЛПУ по процессу постановки онкологического диагноза*/
select ltrim(rtrim(isnull(окр,'Без указания округа'))) as GroupLPU
      ,ltrim(rtrim(isnull(omain.O_NAME,Oms_OKATO.O_NAME))) as GroupLPU_rn
      ,ltrim(rtrim(isnull(lpu1.mcod,lpu.mcod))) as mMcod
      , '['+ltrim(rtrim(isnull(lpu1.mcod,lpu.mcod)))+']
'+ltrim(rtrim(isnull(lpu1.M_names,lpu.M_names))) as mainLPU
      ,isnull(FIO.Value,'не указано')as LPU_IN_FIO
      ,isnull(Tel.Value,'не указано')as LPU_IN_TEL
      ,t.*
      ,avg_5.avg_5
from [COD_mo].dbo.oms_LPU lpu WITH(NOLOCK)
     left join [cod_MO].dbo.oms_LPU lpu1 WITH(NOLOCK) on (lpu.rf_MainLPUID = lpu1.LPUID
and lpu1.LPUID > 0) and lpu1.rf_MainLPUID = 0
     left join [cod_MO].dbo.Oms_OKATO WITH(NOLOCK) on OKATOID =
isnull(lpu1.rf_OKATOID,lpu.rf_OKATOID)
     left join [cod_MO].dbo.oms_OKATO omain WITH(NOLOCK) on omain.C_OKATO =
left(oms_OKATO.C_OKATO,5)+'000000'
     left join (
         select case when len(oms_TehAtrEnumValue.Name) = 1 then '0' else '' end
+oms_TehAtrEnumValue.Name as окр, rf_LPUID
         from [cod_MO].dbo.oms_tehatrvalue WITH(NOLOCK)
         inner join [cod_MO].dbo.oms_TehAtrEnumValue WITH(NOLOCK) on
oms_TehAtrEnumValue.Code = Value
         inner join [cod_MO].dbo.oms_tehatr t2 WITH(NOLOCK) on t2.tehatrID =
oms_tehatrvalue.rf_tehatrID and t2.Code = '0002'
         where tehatrvalueID in (
             select max(tehatrvalueID)
             from [cod_MO].dbo.oms_tehatrvalue WITH(NOLOCK)
             where oms_tehatrvalue.date_E > getdate()
             group by rf_LPUID,rf_tehatrID
         )
     )tokr on isnull(lpu1.LPUID,lpu.LPUID) = tokr.rf_LPUID
left join (
     select rf_LPUID, Value
     from [cod_MO].dbo.oms_tehatrvalue WITH(NOLOCK)
     where tehatrvalueID in (
         select max(tehatrvalueID)
         from [cod_MO].dbo.oms_tehatrvalue WITH(NOLOCK)
         inner join [cod_MO].dbo.oms_tehatr WITH(NOLOCK) on
tehatrID = rf_tehatrID
         where oms_tehatrvalue.date_E > getdate() and oms_tehatr.Code =
'0026'
         group by rf_LPUID
     )
)FIO on isnull(lpu1.LPUID,lpu.LPUID) = FIO.rf_LPUID
left join (
     select rf_LPUID, Value
     from [cod_MO].dbo.oms_tehatrvalue WITH(NOLOCK)
     where tehatrvalueID in (
         select max(tehatrvalueID)
         from [cod_MO].dbo.oms_tehatrvalue WITH(NOLOCK)
         inner join [cod_MO].dbo.oms_tehatr WITH(NOLOCK) on
tehatrID = rf_tehatrID
```

```

                                where oms_tehatrvalue.date_E > getdate() and oms_tehatr.Code =
'0027'
                                group by rf_LPUIID
                                )
)Tel on isnull(lpu1.LPUIID,lpu.LPUIID) = Tel.rf_LPUIID
left join
(
    select LPU_Mistake
           ,sum(a4+a5+a6+a7+a9+a10+a11+a12+a13+a14+a15+a16+a17) as a_all
           ,sum(a4) as a4
           ,sum(a4_1) as a4_1
           ,sum(a4_2) as a4_2
           ,sum(a5) as a5
           ,sum(a6) as a6
           ,sum(a7) as a7
           ,sum(a9) as a9
           ,sum(a10) as a10
           ,sum(a11) as a11
           ,sum(a12) as a12
           ,sum(a13) as a13
           ,sum(a14) as a14
           ,sum(a15) as a15
           ,sum(a16) as a16
           ,sum(a17) as a17
    from (
        --Ожидание записи на консультацию
        select
            case when zap3.DateRasp is null and zap.DateRasp is null and
zapm.DateRasp is null then dir.rf_LPUSenderID else dir.rf_LPUIID end as LPU_Mistake /*ЛПУ к
которой относится конкретное действие*/
            ,count(1) as a4
            ,count(case when zap.DateRasp is null then dir.FAMILY else null
end) as a4_1
            ,count(case when zap.DateRasp is not null and dir_d.dat is not
null then dir.FAMILY else null end) as a4_2
            ,0 as a5,0 as a6,0 as a7,0 as a9,0 as a10,0 as a11,0 as a12,0 as
a13,0 as a14,0 as a15,0 as a16,0 as a17
        from (
            select FAMILY, Name, OT, BD, max(n) as n
            from tmp_dir_onko_new WITH(NOLOCK)
            group by FAMILY, Name, OT, BD
        )dirm
        inner join tmp_dir_onko_new dir WITH(NOLOCK) on dir.FAMILY =
dirm.FAMILY and dir.Name = dirm.Name and dir.OT = dirm.OT and dir.BD = dirm.BD and dir.n =
dirm.n
        and dir.DateFactHosp is null and dir.DatePlanHosp is null
and dir.tap_MKBID is null and dir.CloseMkabDeath = 0 and dir.Iskl = 0
        left join (
            select FAMILY, Name, OT, BD, DateDir, max(case when
canceldate is null or canceldate > Begin_Time then Begin_Time else canceldate end) as
DateRasp, max(case when VisitStatus < 0 then -1 else n end) as n
            from tmp_zap_onko_new WITH(NOLOCK)
            group by FAMILY, Name, OT, BD, DateDir
            having max(case when VisitStatus < 0 then -1 else n end) <
1
        )zapm on zapm.FAMILY = dir.FAMILY and zapm.Name = dir.NAME and
zapm.OT = dir.OT and zapm.BD = dir.BD
        and zapm.DateDir = dir.DateDir
        left join (
            select FAMILY, Name, OT, BD, DateDir, max(Begin_Time) as
DateRasp
            from tmp_zap_onko_new WITH(NOLOCK)
            where n != -1 and VisitStatus != -1
            group by FAMILY, Name, OT, BD, DateDir

```

```

)zap on zap.FAMILY = dir.FAMILY and zap.Name = dir.NAME and
zap.OT = dir.OT and zap.BD = dir.BD
    and zap.DateDir = dir.DateDir
left join (
    select FAMILY, Name, OT, BD, DateDir, DateRasp,
max(Begin_Time) as dat
    from tmp_dir_diag_new WITH(NOLOCK)
    where Begin_Time <= getdate()
    group by FAMILY, Name, OT, BD, DateDir, DateRasp
)dir_d on dir_d.FAMILY = dir.FAMILY and dir_d.Name = dir.NAME and
dir_d.OT = dir.OT and dir_d.BD = dir.BD
    and dir_d.DateDir = dir.DateDir and dir_d.DateRasp =
zap.DateRasp
left join (
    select FAMILY, Name, OT, BD, DateDir, max(case when
canceldate is null or canceldate > Begin_Time then Begin_Time else canceldate end) as
DateRasp
    from tmp_zap_onko_new WITH(NOLOCK)
    where n = -1 or VisitStatus = -1
    group by FAMILY, Name, OT, BD, DateDir
)zap3 on zap3.FAMILY = dir.FAMILY and zap3.Name = dir.NAME and
zap3.OT = dir.OT and zap3.BD = dir.BD
    and zap3.DateDir = dir.DateDir and zap3.DateRasp >=
dir_d.dat
    where (zapm.n is not null or (zap.DateRasp is not null and dir_d.FAMILY
is not null) or zap.DateRasp is null)
    and
datediff(day,isnull(zapm.DateRasp,(isnull(zap3.DateRasp,isnull(dir_d.dat,dir.DateDir))))),get
date()) > 1
    group by case when zap3.DateRasp is null and zap.DateRasp is null and
zapm.DateRasp is null then dir.rf_LPUSenderID else dir.rf_LPUID end
union all
--Плановое ожидание проведения консультации
select rf_LPUID
    ,0 as a4,0 as a4_1, 0 as a4_2
    ,count(1) as a5
    ,0 as a6,0 as a7,0 as a9,0 as a10,0 as a11,0 as a12,0 as a13,0 as
a14,0 as a15,0 as a16,0 as a17
    from tmp_zap_onko_new WITH(NOLOCK)
    where Begin_Time > getdate() and
datediff(day,DateTimeCreate,Begin_Time) > 3 and canceldate is null
group by rf_LPUID
union all
--Ожидание проведения консультации
select rf_LPUID
    ,0 as a4,0 as a4_1, 0 as a4_2,0 as a5
    ,count(1) as a6
    ,0 as a7,0 as a9,0 as a10,0 as a11,0 as a12,0 as a13,0 as a14,0
as a15,0 as a16,0 as a17
    from tmp_zap_onko_new WITH(NOLOCK)
    where Begin_Time > getdate() and datediff(day,DateTimeCreate,getdate())
> 3 and canceldate is null
group by rf_LPUID
union all
--Ожидание данных об итогах консультации
select zap.rf_LPUID
    ,0 as a4,0 as a4_1, 0 as a4_2,0 as a5,0 as a6
    ,count(1) as a7
    ,0 as a9,0 as a10,0 as a11,0 as a12,0 as a13,0 as a14,0 as a15,0
as a16,0 as a17
    from (
        select FAMILY, Name, OT, BD, max(n) as n
        from tmp_dir_onko_new WITH(NOLOCK)
        group by FAMILY, Name, OT, BD
    )dirm

```

```

        inner join tmp_dir_onko_new dir WITH(NOLOCK) on dir.FAMILY =
dirm.FAMILY and dir.Name = dirm.Name and dir.OT = dirm.OT and dir.BD = dirm.BD and dir.n =
dirm.n
        and dir.DateFactHosp is null and dir.DatePlanHosp is null
and dir.tap_MKBID is null and dir.CloseMkabDeath = 0 and dir.Iskl = 0
        inner join tmp_zap_onko_new zap WITH(NOLOCK) on zap.FAMILY =
dir.FAMILY and zap.Name = dir.NAME and zap.OT = dir.OT and zap.BD = dir.BD and zap.DateDir =
dir.DateDir
        and zap.VisitStatus > -1
        and zap.Begin_Time < getdate() and
datediff(day,zap.Begin_Time,getdate()) > 1 and zap.canceldate is null
        left join [cod_MO].dbo.hlt_SMTAP tap WITH(NOLOCK) on tap.rf_TAPID
= zap.rf_TAPID
        and datediff(day,zap.Begin_Time,tap.Date_P) = 0
where tap.rf_TAPID is null
group by zap.rf_LPUIID
union all
--Ожидание записи на диагностику
select case when zap.Begin_Time is null then lr.rf_LPUSenderID else
lr.rf_LPUIID end as LPU_Mistake
        ,0 as a4,0 as a4_1, 0 as a4_2,0 as a5,0 as a6,0 as a7
        ,count(1) as a9
        ,0 as a10,0 as a11,0 as a12,0 as a13,0 as a14,0 as a15,0 as a16,0
as a17
from (
        select FAMILY, Name, OT, BD, max(n) as n
        from tmp_dir_onko_new WITH(NOLOCK)
        group by FAMILY, Name, OT, BD
        )dirm
        inner join tmp_dir_onko_new dir WITH(NOLOCK) on dir.FAMILY =
dirm.FAMILY and dir.Name = dirm.Name and dir.OT = dirm.OT and dir.BD = dirm.BD and dir.n =
dirm.n
        and dir.DateFactHosp is null and dir.DatePlanHosp is null
and dir.tap_MKBID is null and dir.CloseMkabDeath = 0 and dir.Iskl = 0
        inner join tmp_dir_diag_new dir_d WITH(NOLOCK) on dir_d.FAMILY =
dir.FAMILY and dir_d.Name = dir.Name and dir_d.OT = dir.OT and dir_d.BD = dir.BD and
dir_d.DateDir = dir.DateDir
        and dir_d.IsPerformed = 0 and dir_d.isComplete = 0 and
dir_d.Begin_Time is null
        left join (
        select row_number() over(partition by
FAMILY,Name,OT,BD,LaboratoryResearchID order by Begin_Time desc) as n
        ,FAMILY,Name,OT,BD,LaboratoryResearchID,Begin_Time,canceldate
        from tmp_zap_diag_new WITH(NOLOCK)
        )zap on zap.LaboratoryResearchID = dir_d.LaboratoryResearchID and
zap.n = 1
        inner join [cod_MO].dbo.lbr_LaboratoryResearch lr WITH(NOLOCK) on
lr.LaboratoryResearchID = dir_d.LaboratoryResearchID
        where
datediff(day,isnull(zap.canceldate,isnull(zap.Begin_Time,dir_d.Date_Direction)),getdate()) >
case when zap.Begin_Time is null then 3 else 1 end
        group by case when zap.Begin_Time is null then lr.rf_LPUSenderID else
lr.rf_LPUIID end
        union all
--Плановое Ожидание проведения диагностики
select zap.rf_LPUIID
        ,0 as a4,0 as a4_1, 0 as a4_2,0 as a5,0 as a6,0 as a7,0 as a9
        ,count(1) as a10
        ,0 as a11,0 as a12,0 as a13,0 as a14,0 as a15,0 as a16,0 as a17
from tmp_dir_diag_new dir WITH(NOLOCK)
        inner join tmp_zap_diag_new zap WITH(NOLOCK) on
zap.LaboratoryResearchID = dir.LaboratoryResearchID and zap.Begin_Time = dir.Begin_Time
        where zap.Begin_Time > getdate() and
datediff(day,zap.DateTimeCreate,zap.Begin_Time) > 3

```

```

group by zap.rf_LPUIID
union all
--Ожидание проведения диагностики
select zap.rf_LPUIID
      ,0 as a4,0 as a4_1, 0 as a4_2,0 as a5,0 as a6,0 as a7,0 as a9,0
as a10
      ,count(1) as a11
      ,0 as a12,0 as a13,0 as a14,0 as a15,0 as a16,0 as a17
from tmp_dir_diag_new dir WITH(NOLOCK)
      inner join tmp_zap_diag_new zap WITH(NOLOCK) on
zap.LaboratoryResearchID = dir.LaboratoryResearchID and zap.Begin_Time = dir.Begin_Time
      where zap.Begin_Time > getdate() and
datediff(day,zap.DateTimeCreate,getdate()) > 3
group by zap.rf_LPUIID
union all
--Ожидание данных о проведении диагностики
select lr.rf_LPUIID
      ,0 as a4,0 as a4_1, 0 as a4_2,0 as a5,0 as a6,0 as a7,0 as a9,0
as a10,0 as a11
      ,count(1) as a12
      ,0 as a13,0 as a14,0 as a15,0 as a16,0 as a17
from (
      select FAMILY, Name, OT, BD, max(n) as n
      from tmp_dir_onko_new WITH(NOLOCK)
      group by FAMILY, Name, OT, BD
      )dirm
      inner join tmp_dir_onko_new dir WITH(NOLOCK) on dir.FAMILY =
dirm.FAMILY and dir.Name = dirm.Name and dir.OT = dirm.OT and dir.BD = dirm.BD and dir.n =
dirm.n
      and dir.DateFactHosp is null and dir.DatePlanHosp is null
and dir.tap_MKBBID is null and dir.CloseMkabDeath = 0 and dir.Isk1 = 0
      inner join tmp_dir_diag_new dir_d WITH(NOLOCK) on dir_d.FAMILY =
dir.FAMILY and dir_d.Name = dir.Name and dir_d.OT = dir.OT and dir_d.BD = dir.BD and
dir_d.DateDir = dir.DateDir
      and dir_d.Begin_Time < getdate()
      and dir_d.IsPerformed = 0 and
datediff(day,dir_d.Begin_Time,getdate()) > 1
      inner join [cod_MO].dbo.lbr_LaboratoryResearch lr WITH(NOLOCK) on
lr.LaboratoryResearchID = dir_d.LaboratoryResearchID
      group by lr.rf_LPUIID
union all
--Ожидание данных о результатах диагностики
select lr.rf_LPUIID
      ,0 as a4,0 as a4_1, 0 as a4_2,0 as a5,0 as a6,0 as a7,0 as a9,0
as a10,0 as a11,0 as a12
      ,count(1) as a13
      ,0 as a14,0 as a15,0 as a16,0 as a17
from (
      select FAMILY, Name, OT, BD, max(n) as n
      from tmp_dir_onko_new WITH(NOLOCK)
      group by FAMILY, Name, OT, BD
      )dirm
      inner join tmp_dir_onko_new dir WITH(NOLOCK) on dir.FAMILY =
dirm.FAMILY and dir.Name = dirm.Name and dir.OT = dirm.OT and dir.BD = dirm.BD and dir.n =
dirm.n
      and dir.DateFactHosp is null and dir.DatePlanHosp is null
and dir.tap_MKBBID is null and dir.CloseMkabDeath = 0 and dir.Isk1 = 0
      inner join tmp_dir_diag_new dir_d WITH(NOLOCK) on dir_d.FAMILY =
dir.FAMILY and dir_d.Name = dir.Name and dir_d.OT = dir.OT and dir_d.BD = dir.BD and
dir_d.DateDir = dir.DateDir
      and dir_d.Begin_Time < getdate()
      and dir_d.IsPerformed > 0 and dir_d.isComplete = 0
      inner join [cod_MO].dbo.lbr_LaboratoryResearch lr WITH(NOLOCK) on
lr.LaboratoryResearchID = dir_d.LaboratoryResearchID

```

```

inner join [cod_mo].dbo.lbr_Research r WITH(NOLOCK) on lr.GUID =
r.rf_LaboratoryResearchGUID
where datediff(day,r.DatePerformed,getdate()) > 5
group by lr.rf_LPUIID
union all
--Ожидание направления в стационар
select zap.rf_LPUIID
,0 as a4,0 as a4_1, 0 as a4_2,0 as a5,0 as a6,0 as a7,0 as a9,0
as a10,0 as a11,0 as a12,0 as a13
,count(1) as a14
,0 as a15,0 as a16,0 as a17
from (
select FAMILY, Name, OT, BD, max(n) as n
from tmp_dir_onko_new WITH(NOLOCK)
group by FAMILY, Name, OT, BD
)dirm
inner join tmp_dir_onko_new dir WITH(NOLOCK) on dir.FAMILY =
dirm.FAMILY and dir.Name = dirm.Name and dir.OT = dirm.OT and dir.BD = dirm.BD and dir.n =
dirm.n
and dir.DateFactHosp is null and dir.DatePlanHosp is null
and dir.tap_MKBID is null and dir.CloseMkabDeath = 0 and dir.Iskl = 0
inner join (
select row_number() over(partition by FAMILY,Name,OT,BD
order by Begin_Time desc,DateTimeCreate asc) as n
,FAMILY,Name,OT,BD,DateDir,DateTimeCreate,Begin_Time
from tmp_zap_onko_new WITH(NOLOCK)
where VisitStatus > -1 and canceldate is null
)zapm on zapm.FAMILY+' '+zapm.Name+' '+zapm.OT+'
'+convert(varchar,zapm.BD,104) = dir.FAMILY+' '+dir.Name+' '+dir.OT+'
'+convert(varchar,dir.BD,104)
and zapm.DateDir = dir.DateDir and zapm.n = 1
and zapm.Begin_Time < getdate()
inner join tmp_zap_onko_new zap WITH(NOLOCK) on zap.FAMILY+'
'+zap.Name+' '+zap.OT+' '+convert(varchar,zap.BD,104) = dir.FAMILY+' '+dir.Name+' '+dir.OT+'
'+convert(varchar,dir.BD,104)
and zap.DateDir = dir.DateDir and zap.Begin_Time =
zapm.Begin_Time and zap.DateTimeCreate = zapm.DateTimeCreate
left join tmp_dir_diag_new dir_diag WITH(NOLOCK) on
dir_diag.FAMILY+' '+dir_diag.Name+' '+dir_diag.OT+' '+convert(varchar,dir_diag.BD,104) =
dir.FAMILY+' '+dir.Name+' '+dir.OT+' '+convert(varchar,dir.BD,104)
and dir_diag.DateDir = dir.DateDir
and dir_diag.DateRasp = zap.Begin_Time
inner join [cod_mo].dbo.hlt_TAP tap WITH(NOLOCK) on tap.TAPID =
zap.rf_TAPID
and tap.rf_MKBID > 0 and tap.TAPID > 0
where dir_diag.FAMILY is null
and datediff(day,zap.Begin_Time,getdate()) > 1
group by zap.rf_LPUIID
union all
--Не определено следующее мероприятие
select zap.rf_LPUIID
,0 as a4,0 as a4_1, 0 as a4_2,0 as a5,0 as a6,0 as a7,0 as a9,0
as a10,0 as a11,0 as a12,0 as a13,0 as a14
,count(1) as a15
,0 as a16,0 as a17
from (
select FAMILY, Name, OT, BD, max(n) as n
from tmp_dir_onko_new WITH(NOLOCK)
group by FAMILY, Name, OT, BD
)dirm
inner join tmp_dir_onko_new dir WITH(NOLOCK) on dir.FAMILY =
dirm.FAMILY and dir.Name = dirm.Name and dir.OT = dirm.OT and dir.BD = dirm.BD and dir.n =
dirm.n

```

```

        and dir.DateFactHosp is null and dir.DatePlanHosp is null
and dir.tap_MKBBID is null and dir.CloseMkabDeath = 0 and dir.Iskl = 0
        inner join (
            select row_number() over(partition by FAMILY,Name,OT,BD
order by Begin_Time desc,DateTimeCreate asc) as n
                ,FAMILY,Name,OT,BD,DateDir,DateTimeCreate,Begin_Time
                from tmp_zap_onko_new WITH(NOLOCK)
                where VisitStatus > -1 and canceldate is null
            )zapm on zapm.FAMILY+' '+zapm.Name+' '+zapm.OT+'
'+convert(varchar,zapm.BD,104) = dir.FAMILY+' '+dir.Name+' '+dir.OT+'
'+convert(varchar,dir.BD,104)
                and zapm.DateDir = dir.DateDir and zapm.n = 1
                and zapm.Begin_Time < getdate()
            inner join tmp_zap_onko_new zap WITH(NOLOCK) on zap.FAMILY+'
'+zap.Name+' '+zap.OT+' '+convert(varchar,zap.BD,104) = dir.FAMILY+' '+dir.Name+' '+dir.OT+'
'+convert(varchar,dir.BD,104)
                and zap.DateDir = dir.DateDir and zap.Begin_Time =
zapm.Begin_Time and zap.DateTimeCreate = zapm.DateTimeCreate
            left join tmp_dir_diag_new dir_diag WITH(NOLOCK) on
dir_diag.FAMILY+' '+dir_diag.Name+' '+dir_diag.OT+' '+convert(varchar,dir_diag.BD,104) =
dir.FAMILY+' '+dir.Name+' '+dir.OT+' '+convert(varchar,dir.BD,104)
                and dir_diag.DateDir = dir.DateDir
                and dir_diag.DateRasp = zap.Begin_Time
            left join [cod_mo].dbo.hlt_TAP tap WITH(NOLOCK) on tap.TAPID =
zap.rf_TAPID
                and tap.rf_MKBBID > 0 and tap.TAPID > 0
where dir_diag.FAMILY is null and tap.rf_MKBBID is null
                and datediff(day,zap.Begin_Time,getdate()) > 1
            group by zap.rf_LPUIID
            union all
            --Ожидание госпитализации
            select dir.LPUHosp
                ,0 as a4,0 as a4_1, 0 as a4_2,0 as a5,0 as a6,0 as a7,0 as a9,0
as a10,0 as a11,0 as a13,0 as a12,0 as a14,0 as a15
                ,count(1) as a16
                ,0 as a17
            from (
                select FAMILY, Name, OT, BD, max(n) as n
                from tmp_dir_onko_new WITH(NOLOCK)
                group by FAMILY, Name, OT, BD
            )dirm
            inner join tmp_dir_onko_new dir WITH(NOLOCK) on dir.FAMILY =
dirm.FAMILY and dir.Name = dirm.Name and dir.OT = dirm.OT and dir.BD = dirm.BD and dir.n =
dirm.n
                and dir.DatePlanHosp is not null and dir.DateFactHosp is
null and dir.CloseMkabDeath = 0 and dir.Iskl = 0
                and datediff(day,dir.DateDirHosp,dir.DatePlanHosp) > 5
            group by dir.LPUHosp
            union all
            --Ожидание госпитализации
            select dir.LPUHosp
                ,0 as a4,0 as a4_1, 0 as a4_2,0 as a5,0 as a6,0 as a7,0 as a9,0
as a10,0 as a11,0 as a13,0 as a14,0 as a12,0 as a15,0 as a16
                ,count(1) as a17
            from (
                select FAMILY, Name, OT, BD, max(n) as n
                from tmp_dir_onko_new WITH(NOLOCK)
                group by FAMILY, Name, OT, BD
            )dirm
            inner join tmp_dir_onko_new dir WITH(NOLOCK) on dir.FAMILY =
dirm.FAMILY and dir.Name = dirm.Name and dir.OT = dirm.OT and dir.BD = dirm.BD and dir.n =
dirm.n
                and dir.DatePlanHosp is not null and dir.DateFactHosp is
null and dir.CloseMkabDeath = 0 and dir.Iskl = 0

```

```

                                and datediff(day,dir.DateDirHosp,getdate()) > 5
                                group by dir.LPUHosp
                                )t group by LPU_Mistake
)t on t.LPU_Mistake = lpu.LPUID
left join (
    select rf_LPUIID
           ,avg(datediff(day,DateTimeCreate,Begin_Time)) as avg_5
    from tmp_zap_onko_new
    where Begin_Time > getdate() and datediff(day,DateTimeCreate,Begin_Time) > 3
and canceldate is null
    group by rf_LPUIID
)avg_5 on avg_5.rf_LPUIID = lpu.LPUID
where lpu.DATE_E > getdate()
order by 1, 2, 3

```

Выпускная квалификационная работа выполнена мной совершенно самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

«___» _____ Г.

(подпись)

(Ф.И.О.)