

**Министерство науки и высшего образования Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ**  
**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«Национальный исследовательский университет ИТМО»**  
**(Университет ИТМО)**

**Факультет      Инфокоммуникационных технологий**

**Образовательная программа**  
**Программирование в инфокоммуникационных системах**

**О Т Ч Е Т**

**по дисциплине «Разработка баз данных»**

Обучающиеся: Гуляева Алиса Павловна группа К33202  
Потитова Валентина Александровна группа К33212  
Долматов Дмитрий Алексеевич группа К33212

Проверено:  
Осетрова Ирина Станиславовна

Санкт-Петербург  
2023

## **СОДЕРЖАНИЕ**

<b>ВВЕДЕНИЕ</b>	<b>3</b>
<b>ВЫПОЛНЕНИЕ ПЕРВОГО ЭТАПА РАБОТЫ</b>	<b>4</b>
1 ER-диаграмма	4
2 Информация об организациях	5
3 Отношения между организациями	10
4 Структура доменов в логической модели	12
5 Окончательные результаты по первой части работы	12
<b>ВЫПОЛНЕНИЕ ВТОРОГО ЭТАПА РАБОТЫ</b>	<b>14</b>
6 Вторая часть работы. Создание логической модели хранилища	14
7 Окончательные результаты по второй части работы	19
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>20</b>

## ВВЕДЕНИЕ

Ни одна больница, а тем более государство, не может поддерживать систему здравоохранения без наличия качественной донорской крови. Главная задача каждого банка крови – следить за качеством крови и контролировать людей, которые сдают кровь, то есть "доноров". Но это трудная задача. Существующие системы не могут удовлетворить потребности в поддержании качества крови и отслеживании доноров.

Система структурирования информации банком крови позволит нам следить за качеством крови, а также отслеживать наличие крови по запросу приёмщика. С помощью этой системы поиск доступной крови станет простым и сэкономит значительное количество времени по сравнению с рукописной системой на бумажных носителях. Она будет накапливать, управлять, восстанавливать и анализировать информацию, связанную с административным управлением и управлением запасами в банке крови.

## ВЫПОЛНЕНИЕ ПЕРВОГО ЭТАПА РАБОТЫ

Целью работы является создание ER-диаграммы для системы управления донорством крови.

Для достижения поставленной цели необходимо выполнить следующие задачи:

- Изучение предметной области,
- Определение сущностей и их атрибутов,
- Выявление связей между сущностями,
- Построение ER-диаграммы,
- Определение ограничений целостности группы атрибутов сущностей.

### **1 ER-диаграмма**

В ходе работы была разработана структура проекта и на ее основе построена ER-диаграмма в приложении Workbench SQL (рисунок 1) [1].

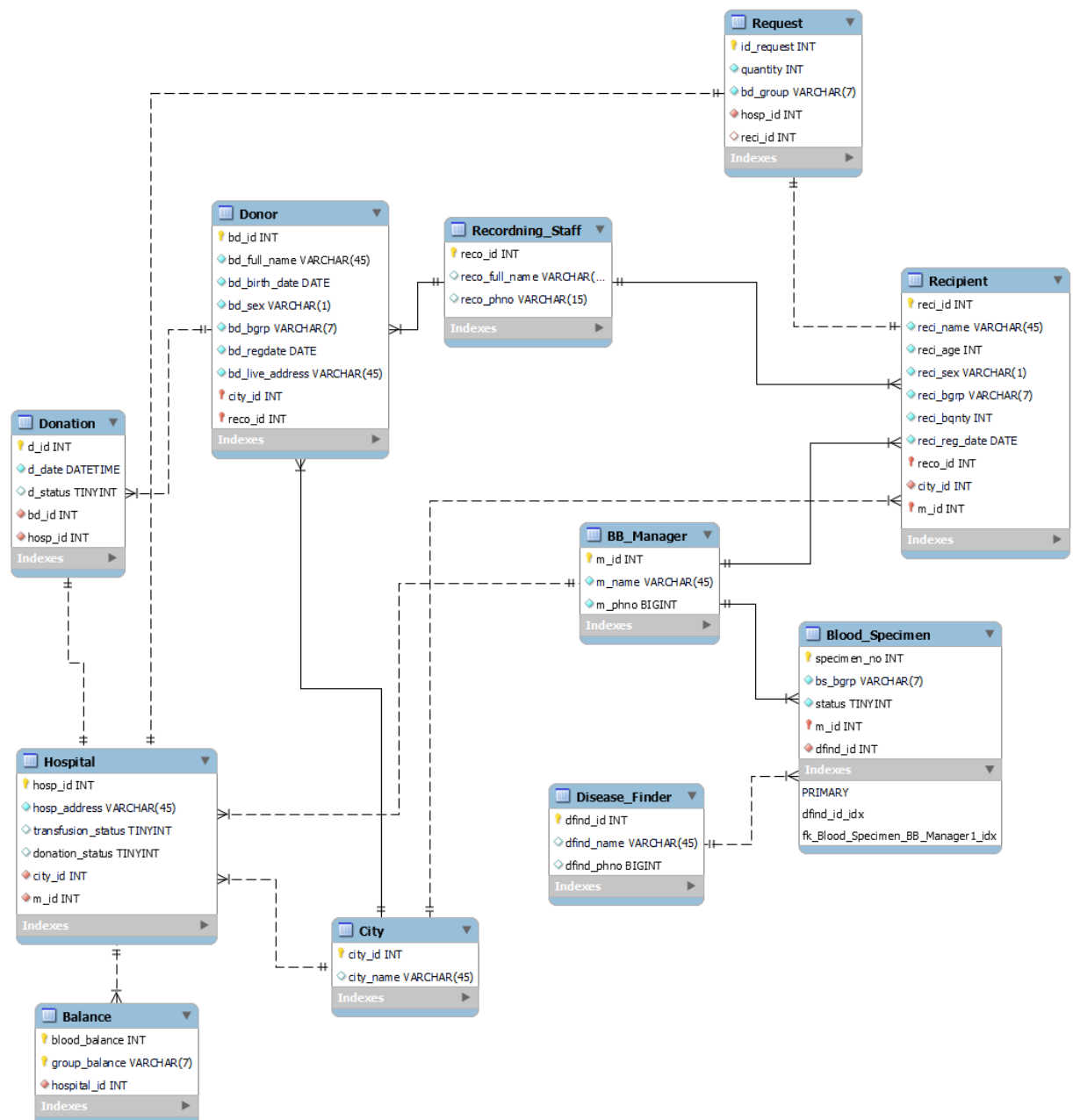


Рисунок 1 – ER-диаграмма

## 2 Информация об организациях

Всего у нас восемь сущностей, и информация о каждой из них в виде краткого описания и тип данных приведена ниже (Таблицы 1 – 8) [2].

Таблица 1 – Donor

№	Атрибут	Описание	Тип данных	Ограничения
1	bd_id	Идентификатор донора	int	PK

2	bd_name	Имя донора	varchar(45)	
3	bd_sex	Пол донора	varchar(6)	
4	bd_age	Возраст донора	int	
5	bd_bgrp	Группа крови донора	varchar(6)	
6	bd_regdate	Дата регистрации донора	date	
7	reco_id	Идентификатор регистрирующего	int	FK
8	city_id	Уникальный идентификатор города	int	FK

Донор – это человек, который сдает кровь, при сдаче крови генерируется идентификатор донора (bd\_id), который используется как первичный ключ для идентификации информации о доноре.

Таблица 2 – Recipient

№	Атрибут	Описание	Тип данных	Ограничения
1	reci_id	Идентификатор получателя	int	PK
2	reci_name	Имя получателя	varchar(45)	
3	reci_age	Возраст получателя	int	
4	reci_bgrp	Группа крови получателя	varchar(6)	
5	reci_bqnty	Необходимое кол-во крови	int	
6	reci_sex	Пол получателя	varchar(6)	

7	reci_reg_date	Дата регистрации получателя	date	
8	reco_id	Идентификатор регистрирующего	int	FK
9	city_id	Уникальный идентификатор города	int	FK
10	m_id	Уникальный идентификационный номер	int	FK

Получатель – это человек, который получает кровь из банка крови. Когда кровь сдается получателю, генерируется идентификатор получателя (reci\_id), который используется как первичный ключ для сущности Recipient, чтобы идентифицировать информацию о получателях крови.

Таблица 3 – BV\_Manager

№	Атрибут	Описание	Тип данных	Ограничения
1	m_id	Уникальный идентификационный номер	int	PK
2	m_name	Имя менеджера	varchar(45)	
3	m_phno	Номер телефона менеджера	bigint	

Менеджер банка крови – это человек, который заботится о доступных образцах крови в банке крови, он также отвечает за обработку запросов на кровь от получателей и больниц. Менеджер банка крови имеет уникальный идентификационный номер (m\_id), используемый в качестве первичного ключа, а также имя и номер телефона менеджера банка крови будут храниться в базе данных под сущностью BV\_Manager.

Таблица 4 – Recording\_Staff

№	Атрибут	Описание	Тип данных	Ограничения
1	reco_id	Идентификатор регистрирующего	int	PK
2	reco_name	Имя регистрирующего	varchar(45)	
3	reco_phno	Номер телефона регистрирующего	bigint	

Регистрирующий персонал – это человек, который регистрирует доноров и реципиентов крови, и сущность Recording\_Staff имеет атрибут reco\_id, который является первичным ключом.

Таблица 5 – Blood\_Specimen

№	Атрибуты	Описание	Тип данных	Ограничения
1	specimen_no	Образец крови, взятый у донора	int	PK
2	b_grp	Группа крови образца крови	varchar(6)	PK
3	status	Статус, который показывает, заражена ли кровь или нет	int	
4	m_id	Уникальный идентификационный номер	int	FK
5	dfind_id	Уникальный идентификатор врача, нашедшего болезнь	int	FK

В базе данных в сущности Blood\_Specimen мы будем хранить информацию об образцах крови, имеющихся в банке крови. В этой сущности specimen\_no и b\_grp вместе будут первичным ключом наряду с атрибутом status, который покажет, заражена ли кровь или нет.



Таблица 6 – Disease\_Finder

№	Атрибуты	Описание	Тип данных	Ограничения
1	dfind_id	Уникальный идентификатор врача, нашедшего болезнь	int	PK
2	dfind_name	Имя врача, нашедшего болезнь	varchar(45)	
3	dfind_phno	Номер телефона врача, нашедшего болезнь	bigint	

В базе данных под сущностью Disease\_Finder мы будем хранить информацию о враче, который проверяет кровь на наличие каких-либо загрязнений. Для хранения этой информации у нас есть уникальный идентификационный номер (dfind\_id) в качестве первичного ключа. Имя и номер телефона врача также будут храниться в той же сущности.

Таблица 7 – Hospital

№	Атрибуты	Описание	Тип данных	Ограничения
1	hosp_id	Идентификатор больницы	int	PK
2	hosp_name	Название больницы	varchar(45)	
3	hosp_needed_bgrp	Необходимая группа крови	varchar(6)	PK
4	hosp_needed_qnty	Количество необходимой группы крови	int	

5	city_id	Уникальный идентификатор города	int	
6	m_id	Идентификатор менеджера банка крови	int	

В базе данных в сущности Hospital мы будем хранить информацию о больницах. В этой сущности hosp\_id и hosp\_needed\_bgrp являются первичными ключами. Мы будем хранить название больницы и количество крови, требуемое в больнице.

Таблица 8 – City

№	Атрибут	Описание	Тип данных	Ограничения
1	city_id	Идентификатор города	int	PK
2	city_name	Название города	varchar	

Сущность City будет хранить информацию о городах, в которых находятся доноры, получатели и больницы. Уникальный идентификационный номер (city\_id) будет использоваться в качестве первичного ключа для определения информации о городе.

### 3 Отношения между организациями

#### 1) City и Hospital:

- Отношения – "в";
- Тип отношений – 1 ко многим;
- Объяснение – В городе может быть много больниц. Одна больница будет принадлежать одному городу.

#### 2) City и Donor:

- Отношения – "живёт в";

- Тип отношений – 1 ко многим;
- Объяснение – В одном городе может жить много доноров. Один донор будет принадлежать одному городу.

### 3) City и Recipient:

- Отношения – "живёт в";
- Тип отношений – 1 ко многим;
- Объяснение – В одном городе может жить много получателей. Один получатель будет принадлежать одному городу.

### 4) Recording\_Staff и Donor:

- Отношение – "регистрирует";
- Тип отношения – 1 ко многим;
- Объяснение – Один сотрудник регистратуры может зарегистрировать много доноров. Один донор будет регистрироваться одним сотрудником регистратуры.

### 5) Recording\_Staff и Recipient:

- Отношение – "записывает";
- Тип отношения – 1 ко многим;
- Объяснение – Один записывающий сотрудник может записать много получателей. Один получатель будет записан одним сотрудником по записи.

### 6) Hospital\_Info и BV\_Manager:

- Отношение – "запрашивает";
- Тип отношения – 1 ко многим;
- Объяснение – Один менеджер банка крови может обрабатывать запросы от многих больниц. Одна больница направляет запрос менеджеру банка крови. Внешний ключ: m\_id

### 7) BV\_Manager и Blood\_Specimen:

- Отношение – "обрабатывает";
- Тип отношения – 1 ко многим;

- Объяснение – Один менеджер банка крови может управлять многими образцами крови, а одним образцом будет управлять один менеджер.

#### 8) Recipient и BB\_Manager:

- Отношение – "запрашивает у";
- Тип отношения – 1 ко многим;
- Объяснение – Один получатель может запросить кровь у одного менеджера, а один менеджер может обрабатывать запросы от многих получателей.

#### 9) Disease\_finder и Blood\_Specimen:

- Отношение – "проверяет";
- Тип отношения – 1 ко многим;
- Объяснение – Специалист по выявлению заболеваний может проверить много образцов крови. Один образец крови проверяется одним специалистом по выявлению заболеваний.

### **4 Структура доменов в логической модели**

Основными доменами на атрибуты сущности "Donor" должны заключаться в следующем:

- bd\_weight больше или равен 50 кг для мужчин и женщин;
- bd\_age больше или равен 18 лет;
- bd\_reg\_date с прошлой сдачи цельной крови должно пройти, как минимум, 2 месяца и сдавать можно не более 5 раз за год от фактического дня сдачи цельной крови для bd\_sex мужчин и не более 4 раз за год от фактического дня сдачи цельной крови для bd\_sex женщин.

### **5 Окончательные результаты по первой части работы**

В ходе данной работы мы создали ER-диаграмму для системы управления донорством крови с соблюдением первых трёх нормальных форм.

Для этого были выполнены следующие задачи:

- Определение сущностей и их атрибутов,

- Выявления связей между сущностями,
- Построение ER-диаграммы,
- Определение ограничений целостности группы атрибутов сущностей.

Лабораторная работа выполнена полностью, так как были выполнены соответствующие поставленные задачи.

Планируемые результаты достигнуты, они соответствуют цели и изначально запланированным результатам.

## ВЫПОЛНЕНИЕ ВТОРОГО ЭТАПА РАБОТЫ

Целью второго этапа работы является разработка схемы хранилища данных, предназначенного для анализа эффективности банка крови.

В процессе достижения цели должны быть выполнены следующие задачи:

- 1) Определить показатели для анализа данных,
- 2) Разработать схему хранилища данных: измерения и факт,
- 3) Задokumentировать хранилище данных,
- 4) Создать запросы к хранилищу данных,
- 5) Оформить результаты в виде второй части отчёта.

### **6 Создание логической модели хранилища**

В данной части работы акцентируем создание логической схемы ХД на факте получения донорской крови. Созданная схема будет иметь вид "Звезда", состоящая из единственной таблицы фактов, Donation Fact, и трёх таблиц измерений: Recording\_Staff\_Dimension, City\_Dimension, Donor\_Dimension, Date\_Info. Структура последующих таблиц представлена ниже (Таблицы 9 – 13). А логическая схема представлена на рисунке 2.

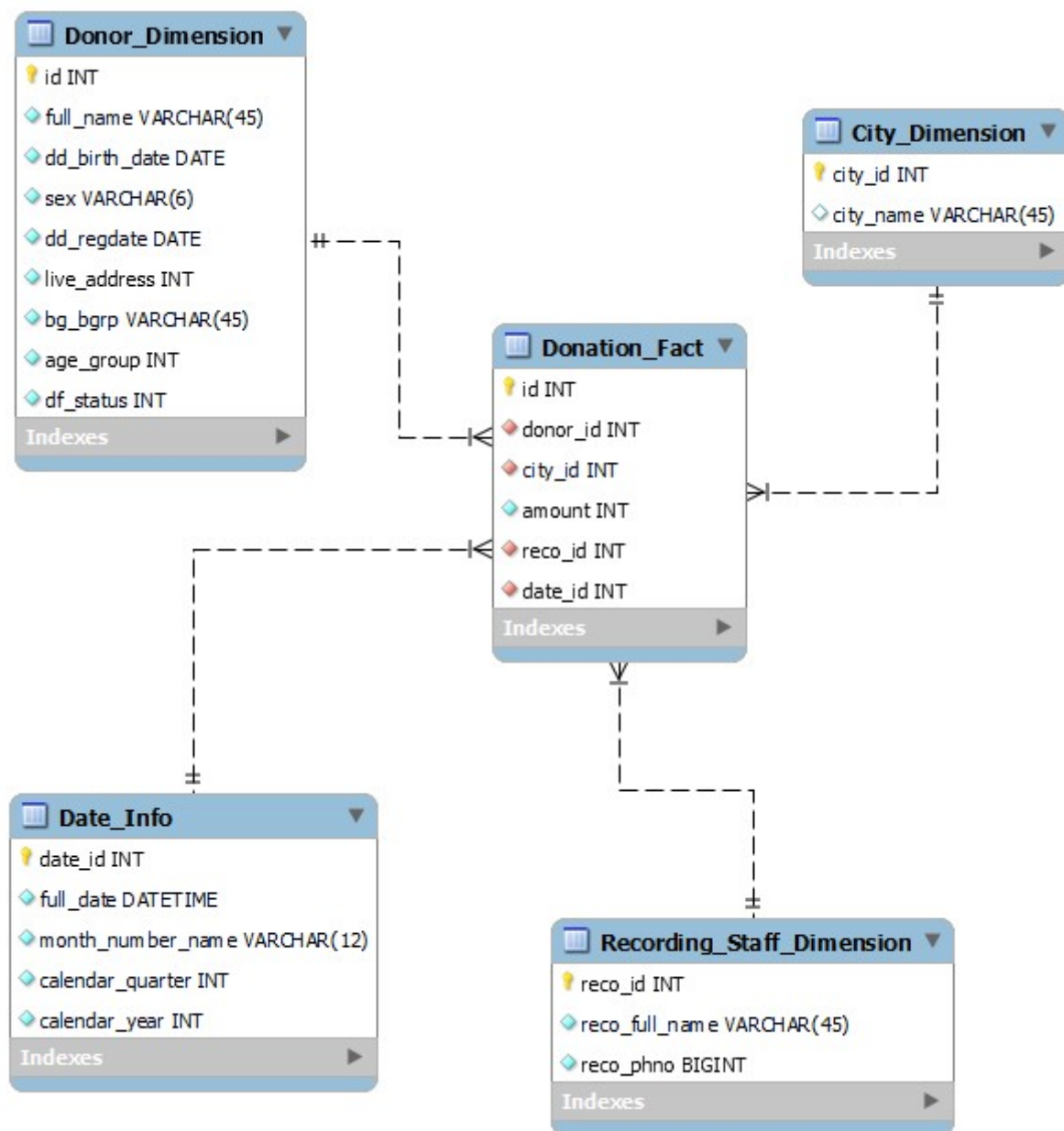


Рисунок 2 – Логическая схема "Звезда"

Таблица 9 – Donation\_Fact

№	Атрибуты	Описание	Тип данных	Ограничения
1	id	Идентификатор донации	INT	PK
2	donor_id	Идентификатор донора	INT	FK, NN

3	city_id	Идентификатор города	INT	FK, NN
4	amount	Количество крови	BIGINT	NN
5	reco_id	Идентификатор сотрудника	INT	FK, NN
6	date_id	Идентификатор даты	INT	FK, NN

Таблица 10 – Donor\_Dimension

№	Атрибуты	Описание	Тип данных	Ограничения
1	id	Идентификатор донора	INT	PK
2	full_name	Полное имя донора	VARCHAR(45)	NN
3	dd_birth_date	Дата рождения донора	DATE	NN
4	sex	Идентификатор пола донора	VARCHAR(6)	NN
5	dd_redgate	Дата регистрации донора	DATE	NN
6	dd_live_address	Место проживания донора	VARCHAR(45)	NN
7	bg_bgrp	Идентификатор группы донора	VARCHAR(6)	NN



8	age_group	Возрастная группа	INT	NN
9	status	Статус донации	INT	NN

Таблица 11 – City\_Dimension

№	Атрибуты	Описание	Тип данных	Ограничения
1	city_id	Идентификатор города	INT	PK
2	city_name	Полное название города	VARCHAR(45)	NN

Таблица 12 – Recording\_Staff\_Dimension

№	Атрибут	Описание	Тип данных	Ограничения
1	reco_id	Идентификатор регистрирующего	INT	PK
2	reco_name	Имя регистрирующего	VARCHAR(45)	NN
3	reco_phno	Номер телефона регистрирующего	BIGINT	NN

Таблица 13 - Date\_Info

№	Атрибут	Описание	Тип данных	Ограничения
1	date_id	Идентификатор даты	INT	PK
2	full_date	Полная дата	DATETIME	NN

		совершённой донации		
3	month_number_name	Номер месяца в прописном формате	VARCHAR(12)	NN
4	calendar_quarter	Номер календарного квартала	INT	NN
5	calendar_year	Номер календарного года	INT	NN

Стоит добавить, что таблица фактов частично состоит из внешних ключей таблиц измерений: bd\_id, city\_id, reco\_id, date\_id. Самый главный атрибут таблицы фактов - это общее количество крови. Также довольно полезным будет добавление избыточности DATETIME для минимизации использования функций.

Для концептуального осмысления необходимости в использовании данной системы для анализа данных создадим SQL запросы, которые могут быть к ней применимы. Они представлены на рисунке 3.

```

1 / Первый запрос
2 SELECT COUNT(*) AS total_donations, AVG(d_status) AS average_status
3 FROM Donation_Fact;
4
5 // Второй запрос
6 SELECT dd.bd_bgrp, AVG(dd.d_status) AS avg_donation_status
7 FROM Donation_Fact AS df
8 JOIN Donor_Dimension AS dd ON df.bd_id = d.bd_id
9 GROUP BY d.bd_bgrp
10
11
12 // Третий запрос
13 SELECT c.city_name, COUNT(*) AS donor_count
14 FROM Donation_Fact AS df
15 JOIN City_Dimension AS c ON df.city_id = c.city_id
16 GROUP BY c.city_name
17
18 // Четвёртый запрос
19 SELECT df.calendar_year AS donation_year, df.month_number_name AS donation_month, COUNT(*) AS donation_count
20 FROM Donation_Fact AS df JOIN Donor_Dimension ON Donor_Dimension.id = Donation_Fact.id
21 GROUP BY calendar_quarter, month_number_name
22 ORDER BY calendar_quarter, month_number_name

```

Рисунок 3 – Возможные запросы для анализа полученных данных из хранилища

Первый и второй запрос приводят статистику, насколько образцы крови из донации соответствовали требованиям безопасности для её передачи реципиенту. Второй запрос группирует результаты первого по группам крови. Третий запрос приводит статистику по городам о донациях. Четвёртый запрос группирует статистику по сдаче в пределах месяца для выявления сезонности и более точного планирования поставки средств, создании субсидий для привлечения населения.

## **7 Окончательные результаты по второй части работы**

Анализируя созданную логическую схему для хранилища данных и соответствующие потенциальные SQL-запросы, можно сделать вывод о практической применимости данной таблицы фактов для последующего исследования аналитических гипотез. В частности:

1. Первый и второй запросы предоставляют возможность определить коэффициент удачных донаций, что может быть полезно для предварительного планирования объема доступной чистой крови для медицинских учреждений или городов,
2. Третий запрос агрегирует информацию о сданных донациях по городу, что может служить надежным ориентиром для определения необходимости поставки медицинского оборудования, такого как шприцы, спирт, пакеты для крови,
3. Четвертый запрос предоставляет исторические данные о сезонности сдачи крови, что позволяет оценить планируемый профицит или дефицит крови, а также рассмотреть возможные стимулы для населения сдавать кровь в более дефицитные месяцы или периоды.

Таким образом, проведенный анализ логической схемы ХД, прикладной области и соответствующих SQL-запросов позволяет сделать вывод о практической ценности данной таблицы фактов в общей логической схеме "Звезда" для разработки аналитических гипотез и применения их в реальной жизни.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Создание ER-диаграммы в среде MySQL Workbench [Электронный ресурс]. URL:  
[https://github.com/kolei/PiRIS/blob/master/articles/5\\_1\\_1\\_1\\_erd\\_workbench.md](https://github.com/kolei/PiRIS/blob/master/articles/5_1_1_1_erd_workbench.md) (дата обращения: 05.03.23).
2. Сущности и атрибуты [Электронный ресурс]. URL:  
<https://studfile.net/preview/4545059/page:2/> (дата обращения: 05.03.23).
3. Связи (отношения) между сущностями [Электронный ресурс]. URL:  
<https://studfile.net/preview/4050050/page:7/> (дата обращения: 05.03.23).
4. Среда MySQL Workbench для графического представления логической модели базы данных [Электронный ресурс]. URL:  
<https://www.mysql.com/products/workbench/> (дата обращения: 05.03.2023)