**Учреждение образования**

**«Международный государственный экологический институт**

**имени А. Д. Сахарова» Белорусского государственного университета**

Факультет мониторинга окружающей среды

Кафедра информационных технологий в экологии и медицине

**Проектирование и создание баз данных по семантическому описанию**

Курсовой проект по дисциплине «Базы данных»

Студента 3-го курса направления специальности А81ИСТ2

«Информационные системы и технологии (в здравоохранении)»

Шундрика Владислава Владимировича

\_\_\_\_\_\_\_\_\_\_\_В.В. Шундрик

Научный руководитель:

к.ф.-м.н., доцент

\_\_\_\_\_\_\_\_\_\_\_В.А. Иванюкович

Минск 2020

**Учреждение образования**

**«Международный государственный экологический институт**

**имени А. Д. Сахарова» Белорусского государственного университета**

Факультет мониторинга окружающей среды

Кафедра информационных технологий в экологии и медицине

**Проектирование и создание базы данных по семантическому описанию**

**Пояснительная записка к курсовому проекту**

Студент 3 курса \_\_\_\_\_\_\_\_\_\_\_ В.В. Шундрик

Минск 2020

# Содержание

[**Введение** 5](#_Toc59661342)

[1 **Постановка** **задачи** 6](#_Toc59661343)

[2 **Построение** **инфологической** **модели** 7](#_Toc59661344)

[2.1 Связи между сущностями 8](#_Toc59661345)

[3 **Нормализация** **таблиц** 9](#_Toc59661346)

[4 **Запросы** **на** **создание** **таблиц** 12](#_Toc59661347)

[5 **Создание** **запросов** **к** **базе** **данных** 17](#_Toc59661348)

[5.1 Запрос №1 17](#_Toc59661349)

[5.2 Запрос №2 17](#_Toc59661350)

[5.3 Запрос №3 18](#_Toc59661351)

[5.4 Запрос №4 18](#_Toc59661352)

[5.5 Запрос №5 19](#_Toc59661353)

[5.6 Запрос №6 19](#_Toc59661354)

[5.7 Запрос №7 20](#_Toc59661355)

[5.8 Запрос №8 20](#_Toc59661356)

[**Заключение** 21](#_Toc59661357)

[Список **литературы** 22](#_Toc59661358)

# Введение

Основные идеи современной информационной технологии базируются на концепции, согласно которой данные должны быть организованы в базы данных с целью адекватного отображения изменяющегося реального мира и удовлетворения информационных потребностей пользователей. Эти базы данных создаются и функционируют под управлением специальных программных комплексов, называемых системами управления базами данных (СУБД).

Увеличение объема и структурной сложности хранимых данных, расширение круга пользователей информационных систем привели к широкому распространению наиболее удобных и сравнительно простых для понимания реляционных (табличных) СУБД. Для обеспечения одновременного доступа к данным множества пользователей, нередко расположенных достаточно далеко друг от друга и от места хранения баз данных, созданы сетевые мультипользовательские версии БД основанных на реляционной структуре. В них тем или иным путем решаются специфические проблемы параллельных процессов, целостности (правильности) и безопасности данных, а также санкционирования доступа.

Тема курсового проекта – построение базы данных «Редколлегия научного журнала».

Предмет исследования – разработка БД, основанной на инфологической модели по семантическому описанию.

Для выполнения задания будут использоваться возможности программы СУБД MS SQL SERVER 2019 EXPRESS.

# Постановка задачи

Необходимо создать базу данных для хранения сведений о работе редколлегии научного журнала:

1. Состав редколлегии (ФИО, возраст, ученая степень, ученое звание, специальность, место основной работы, дата начала и окончания работы в редколлегии).
2. График заседаний (по годам).
3. Присутствующие на заседании.
4. Учет полученных статей.
5. Сведения об авторах (ФИО, возраст, ученая степень, ученое звание, специальность, место основной работы, прежние статьи).
6. Сведения об рецензентах (ФИО, возраст, ученая степень, ученое звание, специальность, место основной работы, прежние статьи).
7. Сведения об организациях, где работают авторы и рецензенты.
8. Переписка с авторами (замечания рецензента, дата, ответы авторов).
9. Учет опубликованных статей.

Создать 5 запросов к базе данных.

# Построение инфологической модели

Для того чтобы построить инфологическую модель необходимо выделить сущности и атрибуты базы данных, согласно семантическому описанию.

Требования: необходимо выделить сущности и создать между ними связи, согласно семантическому описанию.

Условия: у одного автора может быть несколько статей, у одной статьи может быть несколько авторов, у одной статьи более одного рецензента, у одного рецензента может быть более одной статьи, в одной организации может работать несколько человек, статья может быть рассмотрена только один раз, на одном заседании может быть рассмотрено несколько статей, один автор и рецензент могут участвовать в нескольких переписках, в одной переписке может участвовать только один автор и рецензент, один сотрудник может участвовать в нескольких заседаниях, на одном заседании может быть несколько сотрудников.

При проектировании требуемой базы данных нужно выделить 8 сущностей:

* Сущность 1 – Организации;
* Сущность 2 – Заседания;
* Сущность 3 – Авторы;
* Сущность 4 – Рецензенты;
* Сущность 5 – Состав\_редколлегии;
* Сущность 6 – Учет\_Статей;
* Сущность 7 – Переписка;
* Сущность 8 – Макет\_Статьи;

Для соблюдения условий о соавторстве, рецензировании одной статьи несколькими рецензентами и посещаемости заседаний редколлегии, добавляются три сущности , содержащие информацию об авторах и рецензентах статей и о посещаемости заседаний.

## Связи между сущностями

Виды связей:

Связь «один-к-одному» (1:1): B каждый момент времени каждому представителю сущности А соответствует 0 или 1 представителей сущности В, а каждому представителю сущности B соответствует 0 или 1 представителей сущности А.

Связь «один-ко-многим» (1:М): одному представителю сущности А соответствуют 0, 1 или несколько представителей сущности В, а любому представителю сущности В соответствует 0 или 1 представителей сущности А.

Связь «многие-ко-многим» (N:М): каждому представителю сущности А может соответствовать множество представителей сущности В, а каждому представителю сущности В может соответствовать множество представителей сущности А.

Инфологическая модель, созданная на основе семантического описания и выделенных сущностей показана на рисунке 1:

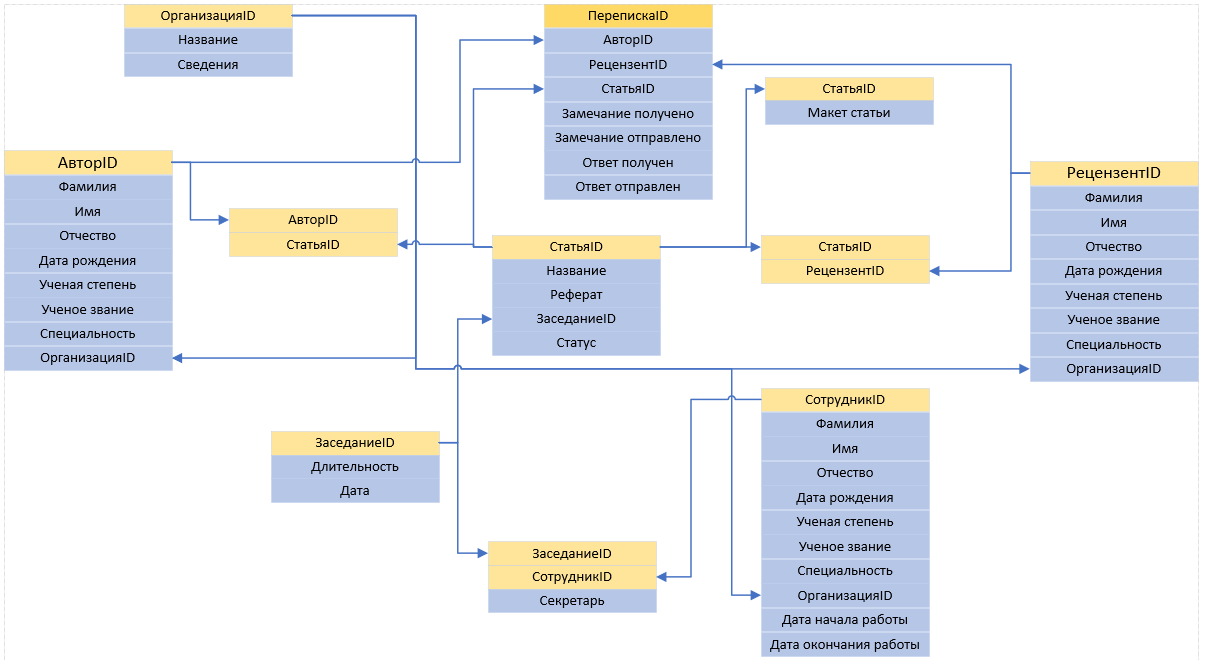


Рисунок 1 – Инфологическая модель базы данных

# Нормализация таблиц

Цель нормализации: исключить избыточное дублирование данных, которое является причиной аномалий, возникших при добавлении или редактировании и удалении кортежей (строк таблицы).

Нормальная форма – требование, предъявляемое к структуре таблиц.

Первая нормальная форма – таблица находится в первой нормальной форме (1НФ) тогда и только тогда, когда ни одна из ее строк не содержит в любом своем поле более одного значения и ни одно из ее ключевых полей не пусто.

Вторая нормальная форма – таблица находится во второй нормальной форме (2НФ), если она удовлетворяет определению 1НФ и все ее поля, не входящие в первичный ключ, связаны неприводимой зависимостью с первичным ключом.

Третья нормальная форма – таблица находится в третьей нормальной форме (3НФ), если она находится в 2НФ и ни одно из ее не ключевых полей не зависит функционально от любого другого не ключевого поля.

Таблицы БД «Редколлегия научного журнала» соответствуют 1 НФ, так как все поля каждой таблицы содержат неделимую информацию и ключевое поле не нулевое.

Данные таблицы находятся во 2НФ так как они выполняют условия 1НФ, а также все их поля, не входящие в первичный ключ, связаны неприводимой зависимостью с первичным ключом. В таблице «ЗаседанияСостав» не ключевое поле связано неприводимой зависимостью с первичным ключом. Таблицы «АвторыСтатья» и «СтатьиРецензенты» не содержат не ключевых полей, а все остальные таблицы базы данных не содержат составной первичный ключ, значит все их не ключевые поля связаны неприводимой зависимостью с первичным ключом.

В таблицах БД «Редколлегия научного журнала» выполняются условия 1НФ и 2НФ, отсутствуют функциональные зависимости одних не ключевых полей от других не ключевых полей, значит таблицы находятся в 3НФ.

Из всего следует, что таблицы базы данных соответствуют трем нормальным формам.

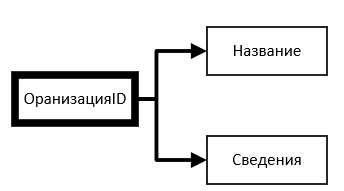
Функциональные диаграммы для всех таблиц базы данных приведены ниже:

Рисунок 2– Функциональная диаграмма для таблицы «Организации»

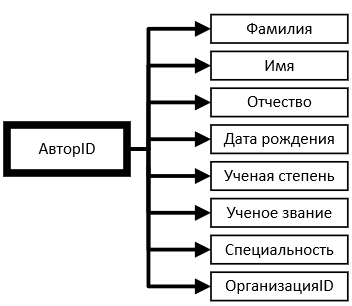


Рисунок 3 – Функциональная диаграмма для таблицы «Авторы»

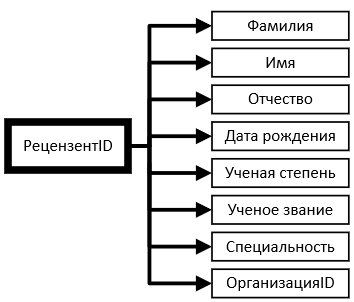


Рисунок 4 – Функциональная диаграмма для таблицы «Рецензенты»

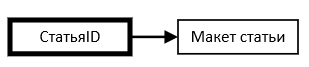


Рисунок 5 – Функциональная диаграмма для таблицы «Макет статьи»

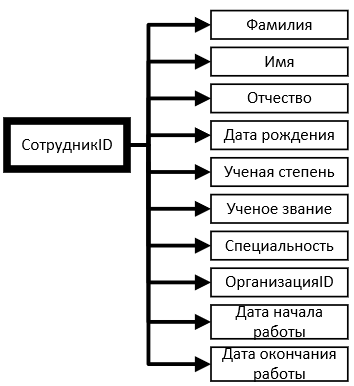


Рисунок 6 – Функциональная диаграмма для таблицы «Состав\_редколлегии»

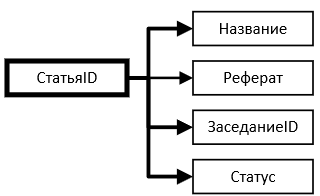


Рисунок 7 – Функциональная диаграмма для таблицы «Учет\_Статей»

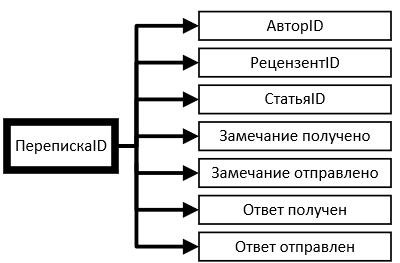


Рисунок 8 – Функциональная диаграмма для таблицы «Переписка»

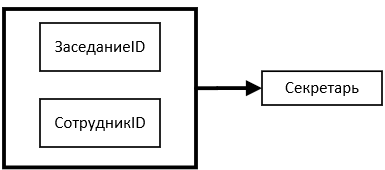


Рисунок 9 – Функциональная диаграмма для таблицы «Переписка»

# Запросы на создание таблиц

Для работы с БД требуется создать 11 таблиц. Должны быть предусмотрены связи между таблицами (первичные, вторичные и составные ключи).

Запрос на создание таблицы «Организации»:

CREATE TABLE Организации

(

ОрганизацияID bigint IDENTITY(1,1) NOT NULL,

Название nvarchar(50) NOT NULL,

Сведения nvarchar(200) NULL,

PRIMARY KEY(ОрганизацияID)

);

Запрос на создание таблицы «Заседания»:

CREATE TABLE Заседания

(

ЗаседаниеID bigint IDENTITY(1,1) NOT NULL,

Длительость int NULL,

Дата date NOT NULL,

PRIMARY KEY(ЗаседаниеID)

);

Запрос на создание таблицы «Авторы»:

CREATE TABLE Авторы

(

АвторID bigint IDENTITY(1,1) NOT NULL,

Фамилия nvarchar(30) NOT NULL,

Имя nvarchar(25) NOT NULL,

Отчество nvarchar(30) NOT NULL,

Дата\_рождения date NOT NULL,

Ученая\_степень nvarchar(40) NOT NULL,

Ученое\_звание nvarchar(15) NOT NULL,

Специальность nvarchar(40) NOT NULL,

ОрганизацияID bigint NOT NULL,

PRIMARY KEY(АвторID),

FOREIGN KEY (ОрганизацияID) REFERENCES Организации (ОрганизацияID)

ON UPDATE CASCADE

ON DELETE CASCADE

);

Запрос на создание таблицы «Рецензенты»:

CREATE TABLE Рецензенты

(

РецензентID bigint IDENTITY(1,1) NOT NULL,

Фамилия nvarchar(30) NOT NULL,

Имя nvarchar(25) NOT NULL,

Отчество nvarchar(30) NOT NULL,

Дата\_рождения date NOT NULL,

Ученая\_степень nvarchar(40) NOT NULL,

Ученое\_звание nvarchar(15) NOT NULL,

Специальность nvarchar(40) NOT NULL,

ОрганизацияID bigint NOT NULL,

PRIMARY KEY(АвторID),

FOREIGN KEY (ОрганизацияID) REFERENCES Организации (ОрганизацияID)

ON UPDATE CASCADE

ON DELETE CASCADE

);

Запрос на создание таблицы «Состав\_редколлегии»:

CREATE TABLE Авторы

(

АвторID bigint IDENTITY(1,1) NOT NULL,

Фамилия nvarchar(30) NOT NULL,

Имя nvarchar(25) NOT NULL,

Отчество nvarchar(30) NOT NULL,

Дата\_рождения date NOT NULL,

Ученая\_степень nvarchar(40) NOT NULL,

Ученое\_звание nvarchar(15) NOT NULL,

Специальность nvarchar(40) NOT NULL,

ОрганизацияID bigint NOT NULL,

Дата\_начала\_работы date NOT NULL,

Дата\_окончания\_работы date NULL,

PRIMARY KEY(СотрудникID),

FOREIGN KEY (ОрганизацияID) REFERENCES Организации (ОрганизацияID)

ON UPDATE CASCADE

ON DELETE CASCADE

);

Запрос на создание таблицы «Учет\_Статей»:

CREATE TABLE Учет\_Статей

(

СтатьяID bigint IDENTITY(1,1) NOT NULL,

Название nvarchar(200) NOT NULL,

Реферат nvarchar(500) NOT NULL,

ЗаседаниеID bigint NULL,

Статус nvarchar(25) NULL,

PRIMARY KEY(СтатьяID),

FOREIGN KEY (ЗаседаниеID) REFERENCES Заседания (ЗаседаниеID)

ON UPDATE CASCADE

ON DELETE CASCADE

);

Запрос на создание таблицы «Переписка»:

CREATE TABLE Переписка

(

ПерепискаID bigint IDENTITY(1,1) NOT NULL,

АвторID bigint NOT NULL,

РецензентID bigint NOT NULL,

СтатьяID bigint NOT NULL,

Замечание\_получено date NOT NULL,

Замечание\_отправлено date NOT NULL,

Ответ\_получен date NOT NULL,

Ответ\_отправлен date NOT NULL,

PRIMARY KEY(ПерепискаID),

FOREIGN KEY (АвторID) REFERENCES Авторы (АвторID)

ON UPDATE CASCADE

ON DELETE CASCADE ,

FOREIGN KEY (РецензентID) REFERENCES Рецензенты (РецензентID),

FOREIGN KEY (СтатьяID) REFERENCES Учет\_Статей (СтатьяID)

ON UPDATE CASCADE

ON DELETE CASCADE

);

Запрос на создание таблицы «АвторыСтатья»:

CREATE TABLE АвторыСтатья(

АвторID bigint NOT NULL,

СтатьяID bigint NOT NULL,

PRIMARY KEY(АвторID, СтатьяID),

FOREIGN KEY (АвторID) REFERENCES Авторы (АвторID)

ON UPDATE CASCADE

ON DELETE CASCADE ,

FOREIGN KEY (СтатьяID) REFERENCES Учет\_Статей (СтатьяID)

ON UPDATE CASCADE

ON DELETE CASCADE

);

Запрос на создание таблицы «СтатьиРецензенты»:

CREATE TABLE СтатьиРецензенты

(

СтатьяID bigint NOT NULL,

РецензентID bigint NOT NULL,

PRIMARY KEY(СтатьяID, РецензентID),

FOREIGN KEY (СтатьяID) REFERENCES Учет\_Статей (СтатьяID)

ON UPDATE CASCADE

ON DELETE CASCADE,

FOREIGN KEY (РецензентID) REFERENCES Рецензенты (РецензентID)

ON UPDATE CASCADE

ON DELETE CASCADE

);

Запрос на создание таблицы «Макет\_Статьи»:

CREATE TABLE Макет\_Статьи(

СтатьяID bigint NOT NULL,

Макет nvarchar(100) NULL,

PRIMARY KEY(СтатьяID),

FOREIGN KEY (СтатьяID) REFERENCES Учет\_Статей (СтатьяID)

ON UPDATE CASCADE

ON DELETE CASCADE

);

Запрос на создание таблицы «ЗаседанияСостав»:

CREATE TABLE ЗаседанияСостав(

ЗаседаниеID bigint NOT NULL,

СотрудникID bigint NOT NULL,

Секретарь nvarchar(5) NOT NULL,

PRIMARY KEY(ЗаседаниеID, СотрудникID),

FOREIGN KEY (ЗаседаниеID) REFERENCES Заседания (ЗаседаниеID)

ON UPDATE CASCADE

ON DELETE CASCADE,

FOREIGN KEY (СотрудникID) REFERENCES Состав\_редколлегии (СотрудникID)

ON UPDATE CASCADE

ON DELETE CASCADE

);

В результате выполнения данных запросов в базе данных было создано 11 таблиц с соответствующими связями между ними.

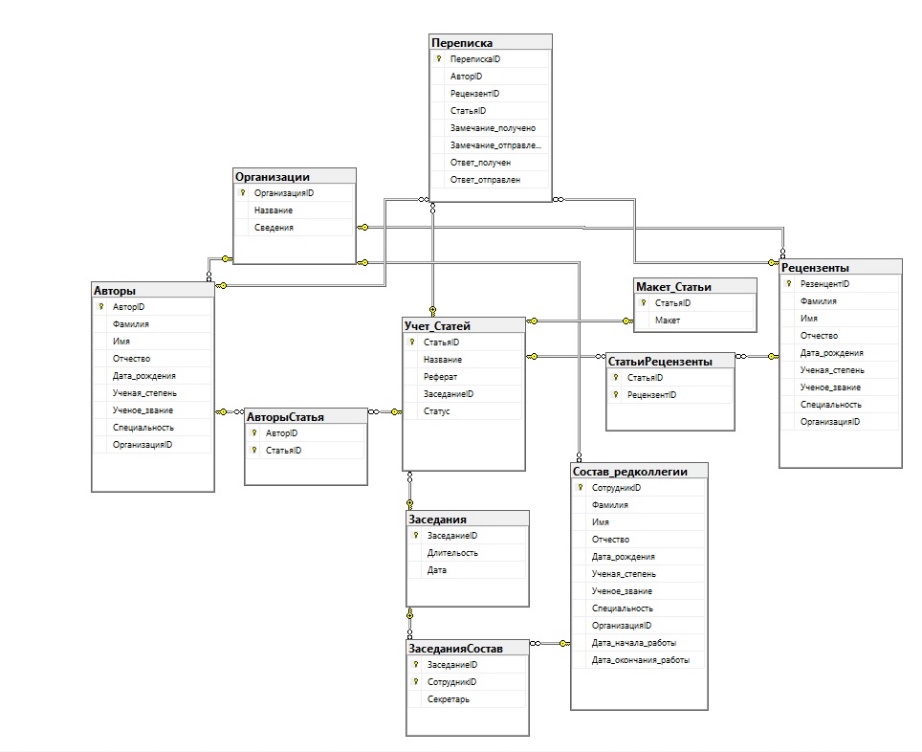


Рисунок 10 – Диаграмма базы данных

# Создание запросов к базе данных

## Запрос №1

Показать всю информации о переписках, в которых участвовали рецензенты с буквой «р» в фамилии:

SELECT DISTINCT Р.Фамилия, Р.Имя, Р.Отчество, П.ПерепискаID, П.АвторID, П.СтатьяID, П.Замечание\_получено, П.Замечание\_отправлено, П.Ответ\_получен, П.Ответ\_отправлен

FROM Рецензенты AS Р INNER JOIN Переписка AS П ON П.РецензентID = Р.РезенцентID

WHERE Фамилия LIKE '%р%'

ORDER BY Р.Фамилия;

Результат выполнения запроса приведен на рисунке 11:

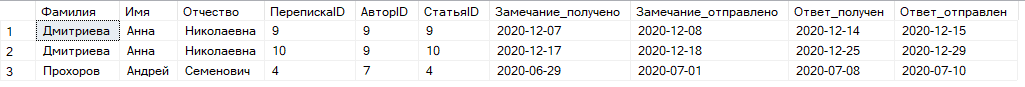


Рисунок 11

## Запрос №2

Показать ФИО, даты рождения, специальности и названия статей тех авторов, чьи года рождения находятся в указанном интервале:

SELECT А.АвторID, А.Фамилия, А.Имя, А.Отчество, А.Дата\_рождения, А.Специальность, У.Название

FROM (Авторы AS А INNER JOIN АвторыСтатья AS АС ON А.АвторID=АС.АвторID) INNER JOIN Учет\_Статей AS У ON АС.СтатьяID=У.СтатьяID

WHERE YEAR(А.Дата\_рождения) BETWEEN 1940 AND 1960

ORDER BY А.Фамилия;

Результат выполнения запроса приведен на рисунке 12:

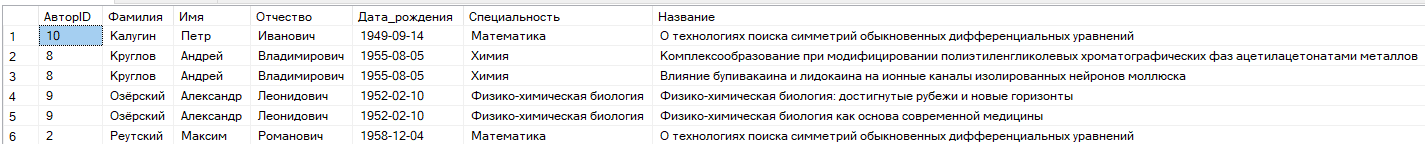


Рисунок 12

## Запрос №3

Показать количество рассмотренных статей на каждом заседании редколлегии. При выводе список отсортировать по количеству рассмотренных статей:

SELECT У.ЗаседаниеID, COUNT(\*) AS kolvo

FROM Учет\_Статей AS У

WHERE У.Статус<>'Не рассмотрена' AND

EXISTS (SELECT \*

FROM Заседания AS З

WHERE У.ЗаседаниеID=З.ЗаседаниеID)

GROUP BY У.ЗаседаниеID

ORDER BY kolvo DESC, ЗаседаниеID ASC

Результат выполнения запроса приведен на рисунке 13:

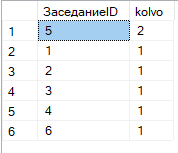


Рисунок 13

## Запрос №4

Показать: есть ли заданного автора и заданного рецензента переписка, и если есть, то вывести информацию об этой переписке:

SELECT П.ПерепискаID, П.СтатьяID, П.Замечание\_получено, П.Замечание\_отправлено, П.Ответ\_получен, П.Ответ\_отправлен

FROM Переписка AS П

WHERE EXISTS(SELECT \*

FROM Авторы AS А WHERE

А.Фамилия='Круглов' AND А.Имя='Андрей'

AND А.АвторID=П.АвторID )

AND EXISTS

(SELECT \*

FROM Рецензенты AS Р WHERE

Р.Фамилия='Давидович' AND Р.Имя='Елена'

AND Р.РезенцентID=П.РецензентID );

Результат выполнения запроса приведен на рисунке 14:

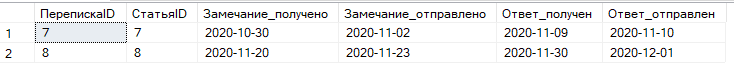


Рисунок 14

## Запрос №5

Вывести информацию о том, какие заседания посетил заданный сотрудник редколлегии и являлся ли он секретарём:

SELECT СР.СотрудникID, З.ЗаседаниеID, З.Длительность,З.Дата,ЗС.Секретарь

FROM (Состав\_редколлегии AS СР INNER JOIN ЗаседанияСостав AS ЗС ON СР.СотрудникID=ЗС.СотрудникID)

INNER JOIN Заседания AS З ON ЗС.ЗаседаниеID=З.ЗаседаниеID

WHERE СР.Фамилия='Радужная' AND СР.Имя = 'Ульяна' AND СР.Отчество = 'Ростиславовна' ;

Результат выполнения запроса приведен на рисунке 15:

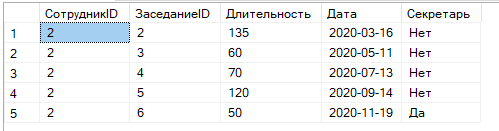


Рисунок 15

## Запрос №6

Вывести информацию (ФИО, ученая степень и звание, место основной работы) о сотрудниках редколлегии, которые посетили конкретное заседание:

SELECT З.ЗаседаниеID, З.Длительность, СР.Фамилия, СР.Имя,СР.Отчество,СР.Ученая\_степень,СР.Ученое\_звание,О.ОрганизацияID, О.Название

FROM ((Заседания AS З INNER JOIN ЗаседанияСостав AS ЗС ON З.ЗаседаниеID=ЗС.ЗаседаниеID)

INNER JOIN Состав\_редколлегии AS СР ON ЗС.СотрудникID=СР.СотрудникID)

INNER JOIN Организации AS О ON СР.ОрганизацияID=О.ОрганизацияID

WHERE З.Дата='2020-07-13'

ORDER BY СР.Фамилия;

Результат выполнения запроса приведен на рисунке 16:

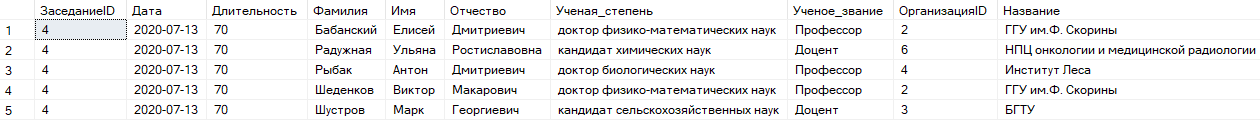


Рисунок 16

## Запрос №7

Вывести информацию о тех заседаниях редколлегии, которые посетило более 4 сотрудников:

SELECT З.ЗаседаниеID, З.Дата, З.Длительность, COUNT(\*) AS kolvo

FROM ЗаседанияСостав AS ЗС INNER JOIN Заседания AS З ON ЗС.ЗаседаниеID=З.ЗаседаниеID

GROUP BY З.ЗаседаниеID, З.Дата,З.Длительность

HAVING COUNT(\*)>4;

Результат выполнения запроса приведен на рисунке 17:

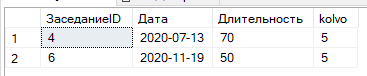


Рисунок 17

## Запрос №8

Показать ФИО и номера авторов, написавших наибольшее количество статей:

SELECT А.АвторID, А.Фамилия,А.Имя,А.Отчество, count(\*) AS Stati

FROM (Авторы AS А INNER JOIN АвторыСтатья AS АС ON А.АвторID=АС.АвторID)

GROUP BY А.АвторID, А.Фамилия,А.Имя,А.Отчество

HAVING count(\*)=(SELECT MAX(kol) AS MKOL

FROM (SELECT count(\*) AS kol

FROM АвторыСтатья AS ас

GROUP BY АвторID) AS D);

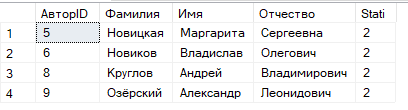
Результат выполнения запроса приведен на рисунке 18:

Рисунок 18

# Заключение

В ходе выполнения курсового проекта была разработана база данных по семантическому описанию. Проверены таблицы базы данных на соответствие первым трём нормальным формам. В том числе были сформулированы SQL-запросы к базе данных.

# Список литературы

1. В. А. Иванюкович: Реляционная модель данных и основы языка программирования SQL. Учебно-методическое пособие / -Мн.: МГЭИ им. А.Д. Сахарова БГУ, 2017. – 67 с
2. В. А. Иванюкович: Системы баз данных. Учебное пособие / -Мн.:Международный государственный экологический институт им. А.Д. Сахарова БГУ, 2016. – 123 с.