

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Таныгин Максим Олегович
Должность: и.о. декана факультета фундаментальной и прикладной информатики
Дата подписания: 21.09.2023 12:44:06
Уникальный программный ключ:
65ab2aa0d384efe8480e6a4c688eddbc475e411a

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра программной инженерии

УТВЕРЖДАЮ

Проректор по учебной работе

О.Г. Локтионова



2020 г.

БАЗЫ ДАННЫХ

Методические указания к выполнению курсовой работы
для студентов
направления подготовки бакалавров
09.03.04 «Программная инженерия»

Курск 2020

УДК 004.65

Составители: В. Г. Белов, Т.М. Белова

Рецензент

Кандидат технических наук, доцент А.В. Малышев

Базы данных: методические указания к выполнению курсовой работы для студентов направления подготовки 09.03.04 «Программная инженерия»/Юго-Зап. гос. ун-т; сост. В. Г. Белов, Т.М. Белова. Курск, 2020. 74 с.

Методические указания на конкретном примере раскрывают структуру, содержание и порядок выполнения курсовой работы по дисциплине «Базы данных» в рамках реализации ФГОС ВО.

Методические указания предназначены для студентов, обучающихся по направлению подготовки бакалавров 09.04.03 – Программная инженерия всех форм обучения.

Текст печатается в авторской редакции.

Подписано в печать . Формат 60x84 1/16.
Усл. печ. л. . Уч.-изд. л. . Тираж 100 экз. Заказ .
Бесплатно.

Юго-Западный государственный университет
305040, Курск, ул.50 лет Октября, 94.

Содержание

Введение.....	4
1 Техническое задание.....	5
1.1 Основание для разработки.....	5
1.2 Назначение разработки.....	5
1.3 Рассмотренные альтернативы.....	5
1.4 Стандарты.....	5
1.5 Требования пользователей к программному изделию.....	5
1.6 Описание прецедентов.....	5
2 Технический проект.....	6
2.1 Неформальное описание предметной области.....	6
2.2 Словарь предметной области.....	7
2.3 Построение функциональных зависимостей.....	14
2.3.1 Исходное множество функциональных зависимостей.....	14
2.4 Синтез схемы базы данных на основании функциональных зависимостей.....	20
2.4.1 Построение избыточного покрытия.....	20
2.4.2 Построение леворедуцированного покрытия.....	20
2.4.3 Построение праворедуцированного покрытия.....	20
2.4.4 Минимальное покрытие.....	23
2.4.5 Минимальное кольцевое покрытие.....	29
2.5 ER-схема базы данных.....	35
2.6 Построение запросов на языке SQL.....	35
2.6.1 Словесная формулировка запроса.....	35
2.6.2 Исходный запрос.....	35
2.6.3 Оптимизированный запрос (опущенная селекция).....	43
2.6.4 Оптимизированный запрос (опущенная проекция).....	52
3 Рабочий проект.....	62
3.1 Разработка приложения в FoxPro.....	62
3.2 Работа с таблицами.....	63
3.3 Разработка запроса.....	83
Заключение.....	84
Список использованных источников.....	85
Приложение А Схемы базы данных.....	86
Приложение Б Требования к оформлению курсовой работы.....	87

Введение

Необходимо спроектировать базу данных для социальной сети книжной направленности. Также целью данной работы является ознакомление с основными принципами построения, использования и оптимизации реляционных баз данных.

Данная тема актуальна, так как большое количество людей сталкивается с потребностью в удобном доступе к информации по книгам, каталогизации прочитанных книг и общении на книжные темы.

В настоящее время существует небольшое количество аналогичных программных продуктов.

Необходимо предоставить пользователю информацию о книгах, авторах, выпускающем составе книг и персонажах, возможность составлять списки прочитанного, отмеченного прочтению, ставить оценки и присваивать метки.

Данная задача будет решаться с помощью анализа предметной области, ее формализации с помощью функциональных зависимостей. Затем необходимо провести этапы минимизации системы функциональных зависимостей, описывающих предметную область, и на основании полученной редуцированной системы спроектировать требуемую модель базы данных.

Предполагается, что проектируемый продукт будет иметь общественно-социальную значимость среди читающих людей.

1 Техническое задание

1.1.1 Основание для разработки

Основанием для разработки является задание на курсовую работу по предмету «Базы данных».

1.2 Назначение разработки

Целью данной разработки является автоматизация каталогизации книг.

1.3 Рассмотренные альтернативы

Существуют другие программы для каталогизации книг, однако стоимость разрабатываемой программы гораздо ниже стоимости других альтернативы.

1.4 Стандарты

Данная работа оформлена в соответствии со стандартом СТУ 04.02.30-2008.

1.5 Требования пользователей к программному изделию

Пользователю необходимо предоставить информацию о книгах, авторах, издательствах, выпускающих составах книг и персонажах литературных произведений, а также возможность составлять списки прочитанного, отмеченного прочтению, ставить оценки и присваивать метки.

1.6 Описание прецедентов

Данная программа позволяет получить информацию о книгах, авторах, издательствах, выпускающих составах книг и персонажах литературных произведений, а также возможность составлять списки прочитанного, отмеченного прочтению, ставить оценки и присваивать метки.

2 Технический проект

2.1 Неформальное описание предметной области

Книжная социальная сеть представляет собой электронный каталог книг, в котором пользователи могут найти информацию о книгах, авторах, издательстве, выпускающих коллективах книг и популярных персонажах, описанных в литературных произведениях.

Пользователи могут помечать книги, как прочитанные, составлять списки имеющихся у них книг (в бумажном и электронном формате), оценивать книги и присваивать им метки.

У издательства есть название и сайт. Издательство выпускает книжные серии. Серия может быть совместным проектом нескольких издательств. Книжные серии состоят из книг (изданий). У издания есть характеристики, которые включают название, ISBN, количество страниц, тип обложки, массу, ширину, высоту, количество страниц в листе и язык издания. ISBN уникален для каждого издания. Издание содержит в себе литературные произведения. У произведения есть название на языке оригинала. У произведения могут быть альтернативные названия (варианты перевода). Произведение может относиться к одному или нескольким жанрам. В произведении участвуют персонажи. Персонажи могут участвовать в нескольких произведениях. У человека может быть по одному имени каждого одинарного типа. У человека может быть несколько имен множественного типа. Для множественных имен должен быть определен порядок следования. Человек может быть автором произведения. Человек может быть редактором издания. Человек может быть переводчиком издания. Человек может быть составителем издания (если оно включает в себя несколько произведений). Человек может быть иллюстратором издания. Человек может выполнять несколько ролей. У издательства, серии, издания, произведения, персонажа и человека есть описание. У пользователя есть имя (nickname), адрес электронной почты и дата регистрации. У пользователя могут быть необязательные характеристики разных типов (строковый, дата). У пользователя может быть по одному значению каждой характеристики. Пользователь может писать рецензии на издания и произведения. Пользователь

может оценивать издания и произведения. Пользователь может присваивать метки изданиям и произведениям.

2.2 Словарь предметной области

Издательство, книжная серия, книга, издание, произведение, личность, автор, редактор, переводчик, составитель, иллюстратор, персонаж, жанр, пользователь, характеристика, оценка, метка, отзыв.

Словарь понятий и терминов для сущности «Тип базового объекта» приведен в таблице 2.1.

Таблица 2.1 – Тип базового объекта

Наименование термина		Примеры значений
Полное	Краткое	
ID	ID	12345
Название	Title	Series Publisher Book Character Personality

Словарь понятий и терминов для сущности «Базовый объект» приведен в таблице 2.2.

Таблица 2.2 – Базовый объект

Наименование термина		Примеры значений
Полное	Краткое	
ID	ID	12345
ID типа	TypeID	12345
Описание	Description	Описание объекта (книги, серии, издательства, персонажа) возможно очень большого размера

Словарь понятий и терминов для сущности «Издательство» приведен в таблице 2.3.

Таблица 2.3 – Издательство

Наименование термина		Примеры значений
Полное	Краткое	
ID	ID	12345
Название	Title	Hodder and Stoughton
Сайт	Website	www.hodder.co.uk

Словарь понятий и терминов для сущности «Серия» приведен в таблице 2.4.

Таблица 2.4 – Серия

Наименование термина		Примеры значений
Полное	Краткое	
ID	ID	12345
Название	Title	Фантастический боевик

Словарь понятий и терминов для сущности «Тип книги» приведен в таблице 2.5.

Таблица 2.5 – Тип книги

Наименование термина		Примеры значений
Полное	Краткое	
ID	ID	12345
Название	Title	Publishing Work

Словарь понятий и терминов для сущности «Книга» приведен в таблице 2.5.

Таблица 2.5 – Книга

Наименование термина		Примеры значений
Полное	Краткое	
ID	ID	12345
ID типа	TypeID	12345

Словарь понятий и терминов для сущности «Издание» приведен в таблице 2.6.

Таблица 2.6 – Издание

Наименование термина		Примеры значений
Полное	Краткое	
ID	ID	12345
ISBN	ISBN	9785170791583
Название	Title	Океан в конце дороги
ID серии	SeriesID	12345
Количество страниц	PageCount	320
Тип обложки	CoverType	7Бц - твердая, целлофанированная (или лакированная)
Масса	Weight	392
Ширина	Width	200
Высота	Height	130
Количество страниц в листе	SleevePageCount	32
Язык издания	PublishingLanguage	Русский

Словарь понятий и терминов для сущности «Произведение» приведен в таблице 2.7.

Таблица 2.7 – Произведение

Наименование термина		Примеры значений
Полное	Краткое	
ID	ID	12345
Название	Title	The Ocean at the End of the Lane
Язык оригинала	OriginLanguage	Английский

Словарь понятий и терминов для сущности «Персонаж» приведен в таблице 2.8.

Таблица 2.8 – Персонаж

Наименование термина		Примеры значений
Полное	Краткое	
ID	ID	12345
Имя	Name	Джек Фрост

Словарь понятий и терминов для сущности «Жанр» приведен в таблице 2.9.

Таблица 2.9 – Жанр

Наименование термина		Примеры значений
Полное	Краткое	
ID	ID	12345
Название	Title	Стимпанк

Словарь понятий и терминов для сущности «Другие названия» приведен в таблице 2.10.

Таблица 2.10 – Другие названия

Наименование термина		Примеры значений
Полное	Краткое	
ID произведения	WorkID	12345
Название	Title	Бог света

Словарь понятий и терминов для сущности «Личность» приведен в таблице 2.11.

Таблица 2.11 – Личность

Наименование термина		Примеры значений
Полное	Краткое	
ID	ID	12345

Словарь понятий и терминов для сущности «Тип одинарных имен» приведен в таблице 2.12.

Таблица 2.12 – Тип одинарных имен

Наименование термина		Примеры значений
Полное	Краткое	
ID	ID	12345
Название	Title	First Name Last Name ...

Словарь понятий и терминов для сущности «Одинарные имена» приведен в таблице 2.13.

Таблица 2.13 – Одинарные имена

Наименование термина		Примеры значений
Полное	Краткое	
ID типа	TypeID	12345
ID человека	PersonalityID	12345
Имя	Name	Терри

Словарь понятий и терминов для сущности «Тип множественных имен» приведен в таблице 2.14.

Таблица 2.14 – Тип множественных имен

Наименование термина		Примеры значений
Полное	Краткое	
ID	ID	12345
Название	Title	Second Name

Словарь понятий и терминов для сущности «Множественные имена» приведен в таблице 2.15.

Таблица 2.15 – Множественные имена

Наименование термина		Примеры значений
Полное	Краткое	
1	2	3
ID типа	TypeID	12345
ID человека	PersonalityID	12345

Окончание таблицы 2.15

1	2	3
Порядковый номер	Number	1
Имя	Name	Джон

Словарь понятий и терминов для сущности «Пользователь» приведен в таблице 2.16.

Таблица 2.16 – Пользователь

Наименование термина		Примеры значений
Полное	Краткое	
ID	ID	12345
Nickname	Nickname	John1234593
E-mail	E-mail	somemailbox@domain.com
Дата регистрации	RegistrationDate	31.05.2014

Словарь понятий и терминов для сущности «Метка» приведен в таблице 2.17.

Таблица 2.17 – Метка

Наименование термина		Примеры значений
Полное	Краткое	
ID пользователя	PersonalityID	12345
ID книги	BookID	12345
Метка	Tag	попаданцы

Словарь понятий и терминов для сущности «Отзыв» приведен в таблице 2.18.

Таблица 2.18 – Отзыв

Наименование термина		Примеры значений
Полное	Краткое	
1	2	3
ID пользователя	PersonalityID	12345

Окончание таблицы 2.18

1	2	3
ID книги	BookID	12345
Текст	Text	Большой и развернутый отзыв на книгу с аргументами и обоснованием оценки

Словарь понятий и терминов для сущности «Оценка» приведен в таблице 2.19.

Таблица 2.19 – Оценка

Наименование термина		Примеры значений
Полное	Краткое	
ID пользователя	PersonalityID	12345
ID книги	BookID	12345
Оценка	Mark	8

Словарь понятий и терминов для сущности «Строковая характеристика пользователя» приведен в таблице 2.20.

Таблица 2.20 – Строковая характеристика пользователя

Наименование термина		Примеры значений
Полное	Краткое	
ID типа	TypeID	12345
ID пользователя	UserID	12345
Значение	Value	Курск

Словарь понятий и терминов для сущности «Тип строковой характеристики пользователя» приведен в таблице 2.21.

Таблица 2.21 – Тип строковой характеристики пользователя

Наименование термина		Примеры значений
Полное	Краткое	
ID	ID	12345
Название	Title	Имя

Словарь понятий и терминов для сущности «Дата пользователя» приведен в таблице 2.22.

Таблица 2.22 – Дата пользователя

Наименование термина		Примеры значений
Полное	Краткое	
ID типа	TypeID	12345
ID пользователя	UserID	12345
Значение	Value	13.05.1993

Словарь понятий и терминов для сущности «Тип даты пользователя» приведен в таблице 2.23.

Таблица 2.23 – Тип даты пользователя

Наименование термина		Примеры значений
Полное	Краткое	
ID	ID	12345
Название	Title	Дата рождения

2.3 Построение функциональных зависимостей

2.3.1 Исходное множество функциональных зависимостей

2.3.1.1 Неформальное описание функциональных зависимостей

Ниже приведены ограничения для таблиц БД

Не может быть более одного типа базового объекта с одинаковым ID:

BaseObjectTypeID ->BaseObjectTypeTitle.

Не может быть более одного базового объекта с одинаковым ID:

BaseObjectID ->BaseObjectTypeID, BaseObjectDescription.

Базовый объект может быть издателем, книжной серией, книгой, человеком или персонажем:

PublisherID -> BaseObjectID,

SeriesID -> BaseObjectID,

BookID -> BaseObjectID,

CharacterID -> BaseObjectID,

PersonalityID -> BaseObjectID.

Не может быть более одного издателя с одинаковым ID:

PublisherID ->PublisherTitle, PublisherWebsite.

Не может быть более одной серии с одинаковым ID:

SeriesID ->SeriesTitle.

У серии может быть несколько издателей, издатели могут издавать несколько серий:

SeriesID, PublisherID ->PublisherPublishesSeries.

Не может быть более одного типа книги с одинаковым ID:

BookTypeID ->BookTypeTitle.

Не может быть более одной книги с одинаковым ID:

BookID ->BookTypeID.

Книга может быть изданием или произведением:

PublicationID ->BookID,

WorkID ->BookID.

Не может быть более одного издания с одинаковым ID:

PublicationID -> SeriesID, PublicationISBN, PublicationPageCount,
PublicationCoverType, PublicationWeight, PublicationWidth, PublicationHeight,
PublicationSleevePageCount, PublishingLanguage, PublicationTitle.

Не может быть более одного издания с одинаковым ISBN:

PublicationISBN -> SeriesID, PublicationID, PublicationPageCount,
PublicationCoverType, PublicationWeight, PublicationWidth, PublicationHeight,
PublicationSleevePageCount, PublishingLanguage, PublicationTitle.

Не может быть более одного произведения с одинаковым ID:

WorkID ->WorkTitle, WorkOriginLanguage.

Издание может включать в себя одно или несколько произведений, произведение может быть издано несколько раз:

PublicationID, WorkID ->PublicationContainsWork.

У произведения может быть несколько альтернативных названий (например, переводных):

WorkID, AltTitle ->AlternativeTitle.

Не может быть более одного жанра с одинаковым ID:

GenreID ->GenreTitle.

Произведение может быть отнесено к одному или нескольким жанрам:

GenreID, WorkID ->WorkBelongsToGenre.

Не может быть более одного персонажа с одинаковым ID:

CharacterID ->CharacterName.

Персонаж может участвовать в нескольких произведениях, в произведении могут участвовать несколько персонажей:

CharacterID, WorkID ->WorkContainsCharacter.

Не может быть более одной личности с одинаковым ID:

PersonalityID ->Personality.

У личности может быть по одному имени каждого одинарного типа:

SingleNamesTypeID, PersonalityID ->SingleName.

Не может быть более одного одинарного типа имени с одинаковым ID:

SingleNameTypeID ->SingleNameTitle.

У личности может быть несколько имен каждого множественного типа, для них должен быть определен порядок:

MultipleNamesTypeID, PersonalityID, MultipleNamesNumber ->MultipleName.

Не может быть более одного множественного типа имени с одинаковым ID:

MultipleNameTypesID ->MultipleNameTitle.

Личность может быть автором произведения, у произведения может быть несколько авторов:

PersonalityID, WorkID ->Authorship.

Личность может быть редактором издания, у издания может быть несколько редакторов:

PersonalityID, PublicationID ->Editorship.

Личность может быть составителем издания, у издания может быть несколько составителей:

PersonalityID, PublicationID ->Compilership.

Личность может быть переводчиком издания, у издания может быть несколько переводчиков:

PersonalityID, PublicationID ->Translatorship.

Личность может быть иллюстратором издания, у издания может быть несколько иллюстраторов:

PersonalityID, PublicationID ->Illustratorship.

Не может быть более одного пользователя с одинаковым ID:

UserID ->UserNickname, UserEmail, UserRegistrationDate.

Не может быть более одного пользователя с одинаковым никнеймом:

UserNickname ->UserID, UserEmail, UserRegistrationDate.

Не может быть более одного пользователя с одинаковым адресом электронной почты:

UserEmail ->UserNickname, UserID, UserRegistrationDate.

Не может быть более одного типа строковой характеристики пользователя с одинаковым ID:

UserStringInfoTypeID ->UserStringInfoTypeTitle.

Не может быть более одного типа даты пользователя с одинаковым ID:

UserDateInfoTypeID ->UserDateInfoTypeTitle.

У пользователя может быть по одному значению каждой строковой характеристики:

UserStringInfoTypeID, UserID ->UserStringInfoValue.

У пользователя может быть по одному значению каждой даты:

UserDateInfoTypeID, UserID ->UserDateInfoValue.

Пользователь может написать отзывы на несколько книг, на одну книгу могут написать отзыв несколько пользователей:

UserID, BookID ->ReviewText.

Пользователь может оценить несколько книг, одну книгу могут оценить несколько пользователей:

UserID, BookID ->MarkValue.

Пользователь может присвоить метку нескольким книгам, одной книге может быть присвоено несколько меток от одного или нескольких пользователей:

UserID, BookID, TagValue ->Tag.

Пользователь может поместить одну или несколько книг в список прочитанного, одна книга может быть в списках прочитанного у нескольких пользователей:

UserID, BookID ->Read.

Пользователь может поместить одну или несколько книг в список к прочтению, одна книга может быть в списках к прочтению у нескольких пользователей:

UserID, BookID ->WantToRead.

Пользователь может поместить одну или несколько книг в список своих электронных книг, одна книга может быть в списках электронных книг у нескольких пользователей:

UserID, BookID ->HaveDigitalBook.

Пользователь может поместить одну или несколько книг в список своих бумажных книг, одна книга может быть в списках бумажных книг у нескольких пользователей:

UserID, BookID -> HavePaperBook.

2.3.1.2 Формальное описание функциональных зависимостей

Формальное описание функциональных зависимостей представлено на рисунках 2.1-2.2.

BaseObjectTypeID -> BaseObjectTypeTitle
BaseObjectID -> BaseObjectTypeID, BaseObjectDescription
PublisherID -> BaseObjectID
SeriesID -> BaseObjectID
BookID -> BaseObjectID
CharacterID -> BaseObjectID
PersonalityID -> BaseObjectID
PublisherID -> PublisherTitle, PublisherWebsite
SeriesID -> SeriesTitle
SeriesID, PublisherID -> PublisherPublishesSeries
BookTypeID -> BookTypeTitle
BookID -> BookTypeID
PublicationID -> BookID
WorkID -> BookID
PublicationID -> SeriesID, PublicationISBN, PublicationPageCount,
PublicationCoverType, PublicationWeight, PublicationWidth, PublicationHeight,
PublicationSleevePageCount, PublishingLanguage, PublicationTitle
PublicationISBN -> SeriesID, PublicationID, PublicationPageCount,
PublicationCoverType, PublicationWeight, PublicationWidth, PublicationHeight,
PublicationSleevePageCount, PublishingLanguage, PublicationTitle
WorkID -> WorkTitle, WorkOriginLanguage
PublicationID, WorkID -> PublicationContainsWork
WorkID, AltTitle -> AlternativeTitle
GenreID -> GenreTitle
GenreID, WorkID -> WorkBelongsToGenre
CharacterID -> CharacterName
CharacterID, WorkID -> WorkContainsCharacter
PersonalityID -> Personality
SingleNameTypesID, PersonalityID -> SingleName
SingleNameTypeID -> SingleNameTitle
MultipleNamesTypeID, PersonalityID, MultipleNamesNumber -> MultipleName
MultipleNamesTypeID -> MultipleNameTitle

Рисунок 2.1

PersonalityID, WorkID -> Authorship
PersonalityID, PublicationID -> Editorship
PersonalityID, PublicationID -> Compilership

```

PersonalityID, PublicationID -> Translatorship
PersonalityID, PublicationID -> Illustratorship
UserID -> UserNickname, UserEmail, UserRegistrationDate
UserNickname -> UserID, UserEmail, UserRegistrationDate
UserEmail -> UserID, UserNickname, UserRegistrationDate
UserStringInfoTypeID -> UserStringInfoTypeTitle
UserDataInfoTypeID -> UserDataInfoTypeTitle
UserDataInfoTypeID, UserID -> UserDataInfoValue
UserStringInfoTypeID, UserID -> UserStringInfoValue
UserID, BookID -> ReviewText
UserID, BookID -> MarkValue
UserID, BookID, TagValue -> Tag
UserID, BookID -> Read
UserID, BookID -> WantToRead
UserID, BookID -> HaveDigitalBook
UserID, BookID -> HavePaperBook

```

Рисунок 2.2

2.4 Синтез схемы базы данных на основании функциональных зависимостей

2.4.1 Построение избыточного покрытия

В результате построения избыточного покрытия множество функциональных зависимостей не изменяется.

2.4.2 Построение леворедуцированного покрытия

В результате построения леворедуцированного покрытия множество функциональных зависимостей не изменяется.

2.4.3 Построение праворедуцированного покрытия

Построение праворедуцированного покрытия приведено на рисунках 2.3-2.5.

```

1. PublicationID -> SeriesID, PublicationISBN, PublicationPageCount,
PublicationCoverType, PublicationWeight, PublicationWidth,
PublicationHeight, PublicationSleevePageCount, PublishingLanguage,
PublicationTitle
1.1. Если удаляем SeriesID

```

Проверяем: $f \mid = \text{PublicationID} \rightarrow \text{SeriesID}$
Да $\Rightarrow g=f$

1.2. Если удаляем `PublicationPageCount`
Проверяем: $f \mid = \text{PublicationID} \rightarrow \text{PublicationPageCount}$
Да $\Rightarrow g=f$

1.3. Если удаляем `PublicationCoverType`
Проверяем: $f \mid = \text{PublicationID} \rightarrow \text{PublicationCoverType}$
Да $\Rightarrow g=f$

1.4. Если удаляем `PublicationWeight`
Проверяем: $f \mid = \text{PublicationID} \rightarrow \text{PublicationWeight}$
Да $\Rightarrow g=f$

1.5. Если удаляем `PublicationWidth`
Проверяем: $f \mid = \text{PublicationID} \rightarrow \text{PublicationWidth}$
Да $\Rightarrow g=f$

1.6. Если удаляем `PublicationHeight`
Проверяем: $f \mid = \text{PublicationID} \rightarrow \text{PublicationHeight}$
Да $\Rightarrow g=f$

1.7. Если удаляем `PublicationSleevePageCount`
Проверяем: $f \mid = \text{PublicationID} \rightarrow \text{PublicationSleevePageCount}$
Да $\Rightarrow g=f$

1.8. Если удаляем `PublishingLanguage`
Проверяем: $f \mid = \text{PublicationID} \rightarrow \text{PublishingLanguage}$
Да $\Rightarrow g=f$

1.9. Если удаляем `PublicationTitle`
Проверяем: $f \mid = \text{PublicationID} \rightarrow \text{PublicationTitle}$
Да $\Rightarrow g=f$

2. `UserID` \rightarrow `UserNickname`, `UserEmail`, `UserRegistrationDate`

2.1. Если удаляем `UserNickname`
Проверяем: $f \mid = \text{UserID} \rightarrow \text{UserNickname}$
Да $\Rightarrow g=f$

Рисунок 2.3

2.2. Если удаляем `UserRegistrationDate`
Проверяем: $f \mid = \text{UserID} \rightarrow \text{UserRegistrationDate}$
Да $\Rightarrow g=f$

3. `UserNickname` \rightarrow `UserID`, `UserEmail`, `UserRegistrationDate`

3.1. Если удаляем `UserID`

Проверяем: f |= UserNickname -> UserID
Да => g=f
3.2. Если удаляем UserRegistrationDate
Проверяем: f |= UserNickname -> UserRegistrationDate
Да=>g=f
Удаление зависимостей вида X->
Праворедуцированное покрытие
g:
BaseObjectTypeID -> BaseObjectTypeTitle
BaseObjectID -> BaseObjectTypeID, BaseObjectDescription
PublisherID -> BaseObjectID
SeriesID -> BaseObjectID
BookID -> BaseObjectID
CharacterID -> BaseObjectID
PersonalityID -> BaseObjectID
PublisherID -> PublisherTitle, PublisherWebsite
SeriesID -> SeriesTitle
SeriesID, PublisherID -> PublisherPublishesSeries
BookTypeID -> BookTypeTitle
BookID -> BookTypeID
PublicationID -> BookID
WorkID -> BookID
PublicationID -> PublicationISBN
PublicationISBN -> SeriesID, PublicationID, PublicationPageCount,
PublicationCoverType, PublicationWeight, PublicationWidth,
PublicationHeight, PublicationSleevePageCount, PublishingLanguage,
PublicationTitle
WorkID -> WorkTitle, WorkOriginLanguage
PublicationID, WorkID -> PublicationContainsWork
WorkID, AltTitle -> AlternativeTitle
GenreID -> GenreTitle

Рисунок 2.4

GenreID, WorkID -> WorkBelongsToGenre
CharacterID -> CharacterName
CharacterID, WorkID -> WorkContainsCharacter
PersonalityID -> Personality

```
SingleNameTypesID, PersonalityID -> SingleName
SingleNameTypeID -> SingleNameTitle
MultipleNamesTypeID, PersonalityID, MultipleNamesNumber ->
MultipleName
MultipleNamesTypeID -> MultipleNameTitle
PersonalityID, WorkID -> Authorship
PersonalityID, PublicationID -> Editorship
PersonalityID, PublicationID -> Compilership
PersonalityID, PublicationID -> Translatorship
PersonalityID, PublicationID -> Illustratorship
UserID -> UserEmail
UserNickname -> UserEmail
UserEmail -> UserID, UserNickname, UserRegistrationDate
UserStringInfoTypeID -> UserStringInfoTypeTitle
UserDataInfoTypeID -> UserDataInfoTypeTitle
UserDataInfoTypeID, UserID -> UserDataInfoValue
UserStringInfoTypeID, UserID -> UserStringInfoValue
UserID, BookID -> ReviewText
UserID, BookID -> MarkValue
UserID, BookID, TagValue -> Tag
UserID, BookID -> Read
UserID, BookID -> WantToRead
UserID, BookID -> HaveDigitalBook
UserID, BookID -> HavePaperBook
```

Рисунок 2.5

2.4.4 Минимальное покрытие

Построение классов эквивалентности и минимального покрытия представлено на рисунках 2.6-2.11.

Построение классов эквивалентности

1. Ef(BaseObjectTypeID):
BaseObjectTypeID ->BaseObjectTypeTitle
2. Ef(BaseObjectID):
BaseObjectID -> BaseObjectTypeID, BaseObjectDescription
3. Ef(PublisherID):
PublisherID -> BaseObjectID
PublisherID -> PublisherTitle, PublisherWebsite
4. Ef(SeriesID):
SeriesID -> BaseObjectID
SeriesID -> SeriesTitle
5. Ef(BookID):
BookID -> BaseObjectID
BookID -> BookTypeID
6. Ef(CharacterID):
CharacterID -> BaseObjectID
CharacterID -> CharacterName
7. Ef(PersonalityID):
PersonalityID -> BaseObjectID
PersonalityID -> Personality
8. Ef(SeriesID, PublisherID):
SeriesID, PublisherID -> PublisherPublishesSeries
9. Ef(BookTypeID):
BookTypeID -> BookTypeTitle
10. Ef(PublicationID):
PublicationID -> BookID
PublicationID -> PublicationISBN
PublicationISBN -> SeriesID, PublicationID, PublicationPageCount,
PublicationCoverType, PublicationWeight, PublicationWidth,
PublicationHeight, PublicationSleevePageCount, PublishingLanguage,
PublicationTitle
11. Ef(WorkID):
WorkID -> BookID
WorkID -> WorkTitle, WorkOriginLanguage
12. Ef(PublicationID, WorkID):
PublicationID, WorkID -> PublicationContainsWork

Рисунок 2.6

13. Ef(WorkID, AltTitle):
WorkID, AltTitle -> AlternativeTitle
14. Ef(GenreID):
GenreID -> GenreTitle
15. Ef(GenreID, WorkID):
GenreID, WorkID -> WorkBelongsToGenre
16. Ef(CharacterID, WorkID):
CharacterID, WorkID -> WorkContainsCharacter
17. Ef(SingleNameTypesID, PersonalityID):
SingleNameTypesID, PersonalityID -> SingleName
18. Ef(SingleNameTypeID):
SingleNameTypeID -> SingleNameTitle
19. Ef(MultipleNamesTypeID, PersonalityID, MultipleNamesNumber):
MultipleNamesTypeID, PersonalityID, MultipleNamesNumber ->
MultipleName
20. Ef(MultipleNamesTypeID):
MultipleNamesTypeID -> MultipleNameTitle
21. Ef(PersonalityID, WorkID):
PersonalityID, WorkID -> Authorship
22. Ef(PersonalityID, PublicationID):
PersonalityID, PublicationID -> Editorship
PersonalityID, PublicationID -> Compilership
PersonalityID, PublicationID -> Translatorship
PersonalityID, PublicationID -> Illustratorship
23. Ef(UserID):
UserID -> UserEmail
UserNickname -> UserEmail
UserEmail -> UserID, UserNickname, UserRegistrationDate
24. Ef(UserStringInfoTypeID):
UserStringInfoTypeID -> UserStringInfoTypeTitle
25. Ef(UserDateInfoTypeID):
UserDateInfoTypeID -> UserDateInfoTypeTitle
26. Ef(UserDateInfoTypeID, UserID):
UserDateInfoTypeID, UserID -> UserDateInfoValue
27. Ef(UserStringInfoTypeID, UserID):
UserStringInfoTypeID, UserID -> UserStringInfoValue

Рисунок 2.7

28. Ef(UserID, BookID):

UserID, BookID -> ReviewText

UserID, BookID -> MarkValue

UserID, BookID -> Read

UserID, BookID -> WantToRead

UserID, BookID -> HaveDigitalBook

UserID, BookID -> HavePaperBook

29. Ef(UserID, BookID, TagValue):

UserID, BookID, TagValue ->Tag

Построение минимального покрытия на основе

прямой функциональной определяемости

1. Проверка: $f \setminus \text{Ef}(\text{PublisherID}) \mid = \text{PublisherID} \rightarrow \text{PublisherID}?$

Да => заменяем PublisherID -> BaseObjectID и PublisherID -> PublisherTitle, PublisherWebsite на PublisherID -> BaseObjectID, PublisherTitle, PublisherWebsite.

2. Проверка: $f \setminus \text{Ef}(\text{SeriesID}) \mid = \text{SeriesID} \rightarrow \text{SeriesID}?$

Да => заменяем SeriesID -> BaseObjectID и SeriesID -> SeriesTitle на SeriesID -> BaseObjectID, SeriesTitle.

3. Проверка: $f \setminus \text{Ef}(\text{BookID}) \mid = \text{BookID} \rightarrow \text{BookID}?$

Да => заменяем BookID -> BaseObjectID и BookID -> BookTypeID на BookID -> BaseObjectID, BookTypeID.

4. Проверка: $f \setminus \text{Ef}(\text{CharacterID}) \mid = \text{CharacterID} \rightarrow \text{CharacterID}?$

Да => заменяем CharacterID -> BaseObjectID и CharacterID -> CharacterName на CharacterID -> BaseObjectID, CharacterName.

5. Проверка: $f \setminus \text{Ef}(\text{PersonalityID}) \mid = \text{PersonalityID} \rightarrow \text{PersonalityID}?$

Да => заменяем PersonalityID -> BaseObjectID и PersonalityID -> Personality на PersonalityID -> BaseObjectID, Personality.

6. Проверка: $f \setminus \text{Ef}(\text{PublicationID}) \mid = \text{PublicationID} \rightarrow \text{PublicationID}?$

Да => заменяем PublicationID -> BookID и PublicationID -> PublicationISBN на PublicationID -> BookID, PublicationISBN.

7. Проверка: $f \setminus \text{Ef}(\text{WorkID}) \mid = \text{WorkID} \rightarrow \text{WorkID}?$

Да => заменяем WorkID -> BookID и WorkID -> WorkTitle, WorkOriginLanguage на WorkID -> BookID, WorkTitle, WorkOriginLanguage.

Рисунок 2.8

8. Проверка: $f \setminus \text{Ef}(\text{PersonalityID}, \text{PublicationID}) \mid = \text{PersonalityID}, \text{PublicationID} \rightarrow \text{PersonalityID}, \text{PublicationID}$?

Да => заменяем PersonalityID, PublicationID -> Editorship и PersonalityID, PublicationID -> Compilership на PersonalityID, PublicationID -> Editorship, Compilership.

9. Проверка: $f \setminus \text{Ef}(\text{PersonalityID}, \text{PublicationID}) \mid = \text{PersonalityID}, \text{PublicationID} \rightarrow \text{PersonalityID}, \text{PublicationID}$?

Да => заменяем PersonalityID, PublicationID -> Editorship, Compilership и PersonalityID, PublicationID -> Translatorship на PersonalityID, PublicationID -> Editorship, Compilership, Translatorship.

10. Проверка: $f \setminus \text{Ef}(\text{PersonalityID}, \text{PublicationID}) \mid = \text{PersonalityID}, \text{PublicationID} \rightarrow \text{PersonalityID}, \text{PublicationID}$?

Да => заменяем PersonalityID, PublicationID -> Editorship, Compilership, Translatorship и PersonalityID, PublicationID -> Illustratorship на PersonalityID, PublicationID -> Editorship, Compilership, Translatorship, Illustratorship.

11. Проверка: $f \setminus \text{Ef}(\text{UserID}, \text{BookID}) \mid = \text{UserID}, \text{BookID} \rightarrow \text{UserID}, \text{BookID}$?

Да => заменяем UserID, BookID -> ReviewText и UserID, BookID -> MarkValue на UserID, BookID -> ReviewText, MarkValue.

12. Проверка: $f \setminus \text{Ef}(\text{UserID}, \text{BookID}) \mid = \text{UserID}, \text{BookID} \rightarrow \text{UserID}, \text{BookID}$?

Да => заменяем UserID, BookID -> ReviewText, MarkValue и UserID, BookID -> Read на UserID, BookID -> ReviewText, MarkValue, Read.

13. Проверка: $f \setminus \text{Ef}(\text{UserID}, \text{BookID}) \mid = \text{UserID}, \text{BookID} \rightarrow \text{UserID}, \text{BookID}$?

Да => заменяем UserID, BookID -> ReviewText, MarkValue, Read и UserID, BookID -> WantToRead на UserID, BookID -> ReviewText, MarkValue, Read, WantToRead.

14. Проверка: $f \setminus \text{Ef}(\text{UserID}, \text{BookID}) \mid = \text{UserID}, \text{BookID} \rightarrow \text{UserID}, \text{BookID}$?

Да => заменяем UserID, BookID -> ReviewText, MarkValue, Read, WantToRead и UserID, BookID -> HaveDigitalBook на UserID, BookID -> ReviewText, MarkValue, Read, WantToRead, HaveDigitalBook.

Рисунок 2.9

15. Проверка: $f \setminus \text{Ef}(\text{UserID}, \text{BookID}) \mid = \text{UserID}, \text{BookID} \rightarrow \text{UserID}, \text{BookID}?$

Да => заменяем UserID, BookID -> ReviewText, MarkValue, Read, WantToRead, HaveDigitalBook и UserID, BookID -> HavePaperBook на UserID, BookID -> ReviewText, MarkValue, Read, WantToRead, HaveDigitalBook, HavePaperBook.

Редуцированное минимальное покрытие

g:

BaseObjectTypeID -> BaseObjectTypeTitle

BaseObjectID -> BaseObjectTypeID, BaseObjectDescription

SeriesID, PublisherID -> PublisherPublishesSeries

BookTypeID -> BookTypeTitle

PublicationISBN -> SeriesID, PublicationID, PublicationPageCount, PublicationCoverType, PublicationWeight, PublicationWidth, PublicationHeight, PublicationSleevePageCount, PublishingLanguage, PublicationTitle

PublicationID, WorkID -> PublicationContainsWork

WorkID, AltTitle -> AlternativeTitle

GenreID -> GenreTitle

GenreID, WorkID -> WorkBelongsToGenre

CharacterID, WorkID -> WorkContainsCharacter

SingleNameTypesID, PersonalityID -> SingleName

SingleNameTypeID -> SingleNameTitle

MultipleNamesTypeID, PersonalityID, MultipleNamesNumber -> MultipleName

MultipleNamesTypeID -> MultipleNameTitle

PersonalityID, WorkID -> Authorship

UserID -> UserEmail

UserNickname -> UserEmail

UserEmail -> UserID, UserNickname, UserRegistrationDate

UserStringInfoTypeID -> UserStringInfoTypeTitle

UserDateInfoTypeID -> UserDateInfoTypeTitle

UserDateInfoTypeID, UserID -> UserDateInfoValue

UserStringInfoTypeID, UserID -> UserStringInfoValue

UserID, BookID, TagValue -> Tag

PublisherID -> BaseObjectID, PublisherTitle, PublisherWebsite

Рисунок 2.10

```
SeriesID -> BaseObjectID, SeriesTitle
BookID -> BaseObjectID, BookTypeID
CharacterID -> BaseObjectID, CharacterName
PersonalityID -> BaseObjectID, Personality
PublicationID -> BookID, PublicationISBN
WorkID -> BookID, WorkTitle, WorkOriginLanguage
PersonalityID, PublicationID -> Editorship, Compilership, Translatorship,
Illustratorship
UserID, BookID -> ReviewText, MarkValue, Read, WantToRead,
HaveDigitalBook, HavePaperBook
```

Рисунок 2.11

2.4.5 Минимальное кольцевое покрытие

Построение минимального кольцевого покрытия представлено на рисунках 2.12-2.18.

```
Минимальноекольцевоепокрытие
( BaseObjectTypeID; ) -> BaseObjectTypeTitle
( BaseObjectID; ) -> BaseObjectTypeID, BaseObjectDescription
( SeriesID, PublisherID; ) -> PublisherPublishesSeries
( BookTypeID; ) -> BookTypeTitle
( PublicationISBN; PublicationID; ) -> SeriesID, PublicationPageCount,
PublicationCoverType, PublicationWeight, PublicationWidth,
PublicationHeight, PublicationSleevePageCount, PublishingLanguage,
PublicationTitle, BookID
( PublicationID, WorkID; ) -> PublicationContainsWork
( WorkID, AltTitle; ) -> AlternativeTitle
( GenreID; ) -> GenreTitle
( GenreID, WorkID; ) -> WorkBelongsToGenre
( CharacterID, WorkID; ) -> WorkContainsCharacter
( SingleNameTypesID, PersonalityID; ) -> SingleName
( SingleNameTypeID; ) -> SingleNameTitle
( MultipleNamesTypeID, PersonalityID, MultipleNamesNumber; ) ->
MultipleName
( MultipleNamesTypeID; ) -> MultipleNameTitle
( PersonalityID, WorkID; ) -> Authorship
```

Рисунок 2.12

```
( UserID; UserNickname; UserEmail; ) -> UserRegistrationDate
( UserStringInfoTypeID; ) -> UserStringInfoTypeTitle
( UserDateInfoTypeID; ) -> UserDateInfoTypeTitle
( UserDateInfoTypeID, UserID; ) -> UserDateInfoValue
( UserStringInfoTypeID, UserID; ) -> UserStringInfoValue
( UserID, BookID, TagValue; ) -> Tag
( PublisherID; ) -> BaseObjectID, PublisherTitle, PublisherWebsite
( SeriesID; ) -> BaseObjectID, SeriesTitle
( BookID; ) -> BaseObjectID, BookTypeID
( CharacterID; ) -> BaseObjectID, CharacterName
( PersonalityID; ) -> BaseObjectID, Personality
( WorkID; ) -> BookID, WorkTitle, WorkOriginLanguage
( PersonalityID, PublicationID; ) -> Editorship, Compilership,
Translatorship,
Illustratorship
( UserID, BookID; ) -> ReviewText, MarkValue, Read, WantToRead,
HaveDigitalBook, HavePaperBook
Естественное характеристическое множество для кольцевого покрытия
f(C):
BaseObjectTypeID -> BaseObjectTypeTitle
    BaseObjectID -> BaseObjectTypeID, BaseObjectDescription
    SeriesID, PublisherID -> PublisherPublishesSeries
    BookTypeID -> BookTypeTitle
    PublicationISBN -> PublicationID
    PublicationID -> PublicationISBN
    PublicationID -> SeriesID, PublicationPageCount,
PublicationCoverType, PublicationWeight, PublicationWidth,
PublicationHeight, PublicationSleevePageCount, PublishingLanguage,
PublicationTitle, BookID
    PublicationID, WorkID -> PublicationContainsWork
    WorkID, AltTitle -> AlternativeTitle
    GenreID -> GenreTitle
    GenreID, WorkID -> WorkBelongsToGenre
    CharacterID, WorkID -> WorkContainsCharacter
    SingleNameTypesID, PersonalityID -> SingleName
    SingleNameTypeID -> SingleNameTitle
```

Рисунок 2.13

MultipleNamesTypeID, PersonalityID, MultipleNamesNumber ->
MultipleName
MultipleNamesTypeID -> MultipleNameTitle
PersonalityID, WorkID -> Authorship
UserID -> UserNickname
UserNickname -> UserEmail
UserEmail -> UserID
UserEmail -> UserRegistrationDate
UserStringInfoTypeID -> UserStringInfoTypeTitle
UserDataInfoTypeID -> UserDataInfoTypeTitle
UserDataInfoTypeID, UserID -> UserDataInfoValue
UserStringInfoTypeID, UserID -> UserStringInfoValue
UserID, BookID, TagValue -> Tag
PublisherID -> BaseObjectID, PublisherTitle, PublisherWebsite
SeriesID -> BaseObjectID, SeriesTitle
BookID -> BaseObjectID, BookTypeID
CharacterID -> BaseObjectID, CharacterName
PersonalityID -> BaseObjectID, Personality
WorkID -> BookID, WorkTitle, WorkOriginLanguage
PersonalityID, PublicationID -> Editorship, Compilership, Translatorship,
Illustratorship
UserID, BookID -> ReviewText, MarkValue, Read, WantToRead,
HaveDigitalBook, HavePaperBook
Минимальноередуцированноекольцевоепокрытие
C
(BaseObjectTypeID;) -> BaseObjectTypeTitle
(BaseObjectID;) ->BaseObjectTypeID, BaseObjectDescription
(SeriesID, PublisherID;) -> PublisherPublishesSeries
(BookTypeID;) -> BookTypeTitle
(PublicationISBN; PublicationID;) -> SeriesID, PublicationPageCount,
PublicationCoverType, PublicationWeight, PublicationWidth,
PublicationHeight, PublicationSleevePageCount, PublishingLanguage,
PublicationTitle, BookID
(PublicationID, WorkID;) -> PublicationContainsWork
(WorkID, AltTitle;) -> AlternativeTitle
(GenreID;) -> GenreTitle

Рисунок 2.14

```
( GenreID, WorkID; ) -> WorkBelongsToGenre
( CharacterID, WorkID; ) -> WorkContainsCharacter
( SingleNameTypeID, PersonalityID; ) -> SingleName
( SingleNameTypeID; ) -> SingleNameTitle
( MultipleNamesTypeID, PersonalityID, MultipleNamesNumber; ) ->
MultipleName
( MultipleNamesTypeID; ) -> MultipleNameTitle
( PersonalityID, WorkID; ) -> Authorship
( UserID; UserNickname; UserEmail; ) -> UserRegistrationDate
( UserStringInfoTypeID; ) -> UserStringInfoTypeTitle
( UserDateInfoTypeID; ) -> UserDateInfoTypeTitle
( UserDateInfoTypeID, UserID; ) -> UserDateInfoValue
( UserStringInfoTypeID, UserID; ) -> UserStringInfoValue
( UserID, BookID, TagValue; ) -> Tag
( PublisherID; ) -> BaseObjectID, PublisherTitle, PublisherWebsite
( SeriesID; ) -> BaseObjectID, SeriesTitle
( BookID; ) -> BaseObjectID, BookTypeID
( CharacterID; ) -> BaseObjectID, CharacterName
( PersonalityID; ) -> BaseObjectID, Personality
( WorkID; ) -> BookID, WorkTitle, WorkOriginLanguage
( PersonalityID, PublicationID; ) -> Editorship, Compilership,
Translatorship, Illustratorship
( UserID, BookID; ) -> ReviewText, MarkValue, Read, WantToRead,
HaveDigitalBook, HavePaperBook
Естественное характеристическое множество
f(C):
BaseObjectTypeID -> BaseObjectTypeTitle
BaseObjectID -> BaseObjectTypeID, BaseObjectDescription
SeriesID, PublisherID -> PublisherPublishesSeries
BookTypeID -> BookTypeTitle
PublicationISBN -> PublicationID
PublicationID -> PublicationISBN
PublicationID -> SeriesID, PublicationPageCount,
PublicationCoverType, PublicationWeight, PublicationWidth,
PublicationHeight, PublicationSleevePageCount, PublishingLanguage,
PublicationTitle, BookID
```


Рисунок 2.15

```

PublicationID, WorkID -> PublicationContainsWork
WorkID, AltTitle -> AlternativeTitle
GenreID -> GenreTitle
GenreID, WorkID -> WorkBelongsToGenre
CharacterID, WorkID -> WorkContainsCharacter
SingleNameTypesID, PersonalityID -> SingleName
SingleNameTypeID -> SingleNameTitle
MultipleNamesTypeID, PersonalityID, MultipleNamesNumber ->
MultipleName
MultipleNamesTypeID -> MultipleNameTitle
PersonalityID, WorkID -> Authorship
UserID -> UserNickname
UserNickname -> UserEmail
UserEmail -> UserID
UserEmail -> UserRegistrationDate
UserStringInfoTypeID -> UserStringInfoTypeTitle
UserDataInfoTypeID -> UserDataInfoTypeTitle
UserDataInfoTypeID, UserID -> UserDataInfoValue
UserStringInfoTypeID, UserID -> UserStringInfoValue
UserID, BookID, TagValue -> Tag
PublisherID -> BaseObjectID, PublisherTitle, PublisherWebsite
SeriesID -> BaseObjectID, SeriesTitle
BookID -> BaseObjectID, BookTypeID
CharacterID -> BaseObjectID, CharacterName
PersonalityID -> BaseObjectID, Personality
WorkID -> BookID, WorkTitle, WorkOriginLanguage
PersonalityID, PublicationID -> Editorship, Compilership, Translatorship,
Illustratorship
UserID, BookID -> ReviewText, MarkValue, Read, WantToRead,
HaveDigitalBook, HavePaperBook
R0 = ( BaseObjectTypeID, BaseObjectTypeTitle )   K0 = {
BaseObjectTypeID }
R1 = ( BaseObjectID, BaseObjectTypeID, BaseObjectDescription )   K1 =
{ BaseObjectID }
R2 = ( SeriesID, PublisherID, PublisherPublishesSeries )   K2 = { SeriesID,
PublisherID }

```

Рисунок 2.16

R3 = (BookTypeID, BookTypeTitle) K3 = { BookTypeID }
 R4 = (PublicationISBN; PublicationID, SeriesID, PublicationPageCount,
 PublicationCoverType, PublicationWeight, PublicationWidth,
 PublicationHeight, PublicationSleevePageCount, PublishingLanguage,
 PublicationTitle, BookID) K4 = { PublicationISBN; PublicationID }
 R5 = (PublicationID, WorkID, PublicationContainsWork) K5 = {
 PublicationID, WorkID }
 R6 = (WorkID, AltTitle, AlternativeTitle) K6 = { WorkID, AltTitle }
 R7 = (GenreID, GenreTitle) K7 = { GenreID }
 R8 = (GenreID, WorkID, WorkBelongsToGenre) K8 = { GenreID,
 WorkID }
 R9 = (CharacterID, WorkID, WorkContainsCharacter) K9 = {
 CharacterID, WorkID }
 R10 = (SingleNameTypeID, PersonalityID, SingleName) K10 = {
 SingleNameTypeID, PersonalityID }
 R11 = (SingleNameTypeID, SingleNameTitle) K11 = {
 SingleNameTypeID }
 R12 = (MultipleNamesTypeID, PersonalityID, MultipleNamesNumber,
 MultipleName) K12 = { MultipleNamesTypeID, PersonalityID,
 MultipleNamesNumber }
 R13 = (MultipleNamesTypeID, MultipleNameTitle) K13 = {
 MultipleNamesTypeID }
 R14 = (PersonalityID, WorkID, Authorship) K14 = { PersonalityID,
 WorkID }
 R15 = (UserID; UserNickname; UserEmail, UserRegistrationDate) K15 =
 { UserID; UserNickname; UserEmail }
 R16 = (UserStringInfoTypeID, UserStringInfoTypeTitle) K16 = {
 UserStringInfoTypeID }
 R17 = (UserDateInfoTypeID, UserDateInfoTypeTitle) K17 = {
 UserDateInfoTypeID }
 R18 = (UserDateInfoTypeID, UserID, UserDateInfoValue) K18 = {
 UserDateInfoTypeID, UserID }
 R19 = (UserStringInfoTypeID, UserID, UserStringInfoValue) K19 = {
 UserStringInfoTypeID, UserID }
 R20 = (UserID, BookID, TagValue, Tag) K20 = { UserID, BookID,
 TagValue }

Рисунок 2.17

```
R21 = ( PublisherID, BaseObjectID, PublisherTitle, PublisherWebsite )
K21 = { PublisherID }
R22 = ( SeriesID, BaseObjectID, SeriesTitle )   K22 = { SeriesID }
R23 = ( BookID, BaseObjectID, BookTypeID )     K23 = { BookID }
R24 = ( CharacterID, BaseObjectID, CharacterName )   K24 = {
CharacterID }
R25 = ( PersonalityID, BaseObjectID, Personality )   K25 = { PersonalityID }
R26 = ( WorkID, BookID, WorkTitle, WorkOriginLanguage )   K26 = {
WorkID }
R27 = ( PersonalityID, PublicationID, Editorship, Compilership,
Translatorship, Illustratorship )   K27 = { PersonalityID, PublicationID }
R28 = ( UserID, BookID, ReviewText, MarkValue, Read, WantToRead,
HaveDigitalBook, HavePaperBook )   K28 = { UserID, BookID }
```

Рисунок 2.18

2.5 ER-схема базы данных

ER-схемы базы данных приведены в приложении А.

2.6 Построение запросов на языке SQL

2.6.1 Словесная формулировка запроса

Выбрать всех людей, связанных с подготовкой определенного издания и вывести их фамилии, имена и роли.

2.6.2 Исходный запрос

2.6.2.1 Текст запроса на языке SQL

Текст исходной комбинации запросов на языке SQL представлен на рисунках 2.19-2.20.

```
SELECT *, 'Редактор' AS Роль FROM editorship AS ed ;
      INNER JOIN single_names AS pnames ON pnames.personality_id = ed.personality_id
AND pnames.type_id = 0;
      INNER JOIN single_names AS lnames ON lnames.personality_id = ed.personality_id
AND lnames.type_id = 1;
UNION ALL ;
SELECT *, 'Составитель' AS Роль FROM compilership AS comp ;
```

Рисунок 2.19

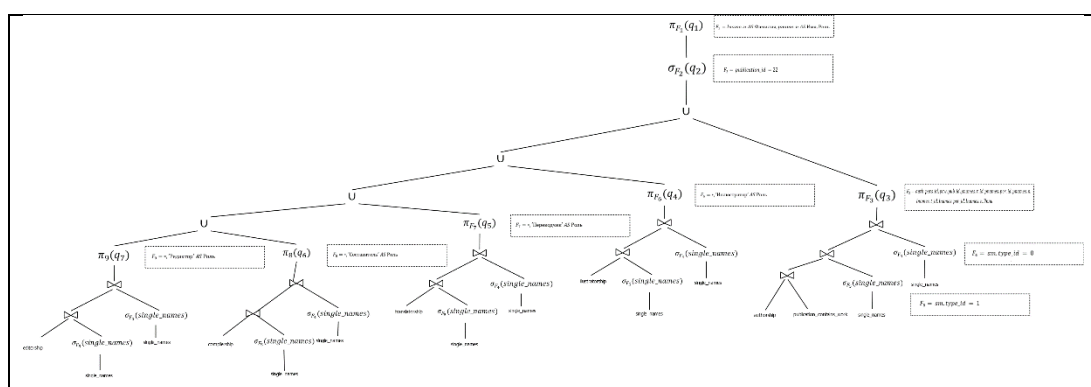
```

INNER JOIN single_names AS pnames ON pnames.personality_id =
comp.personality_id AND pnames.type_id = 0;
INNER JOIN single_names AS lnames ON lnames.personality_id =
comp.personality_id AND lnames.type_id = 1;
UNION ALL ;
SELECT *, 'Переводчик' AS Роль FROM translatorship AS trans ;
INNER JOIN single_names AS pnames ON pnames.personality_id =
trans.personality_id AND pnames.type_id = 0;
INNER JOIN single_names AS lnames ON lnames.personality_id =
trans.personality_id AND lnames.type_id = 1;
UNION ALL ;
SELECT *, 'Иллюстратор' AS Роль FROM Illustratorship AS Illustr ;
INNER JOIN single_names AS pnames ON pnames.personality_id =
Illustr.personality_id AND pnames.type_id = 0;
INNER JOIN single_names AS lnames ON lnames.personality_id =
Illustr.personality_id AND lnames.type_id = 1;
UNION ALL ;
SELECT auth.personality_id, pcw.publication_id, pnames.type_id,
pnames.personality_id, pnames.name, lnames.type_id, lnames.personality_id,
lnames.name, 'Автор' AS Роль FROM authorship AS auth ;
INNER JOIN publication_contains_work AS pcw ON auth.work_id = pcw.work_id ;
INNER JOIN single_names AS pnames ON pnames.personality_id =
auth.personality_id AND pnames.type_id = 0;
INNER JOIN single_names AS lnames ON lnames.personality_id =
auth.personality_id AND lnames.type_id = 1;
INTO TABLE book_team
SELECT Name_b AS Фамилия, Name_a AS Имя, Роль FROM book_team ;
WHERE publicatio = 22

```

Рисунок 2.20

2.6.2.2 Дерево исходного запроса



2.6.2.3 Расчет сложности запроса

Введем псевдонимы для таблиц, участвующих в запросе:

authorship = auth,

editorship = ed,

compilership = comp,

translatorship = trans,

illustratorship = ill,

single_names = sn,

publication_contains_work = pcw.

Введем псевдонимы для полей, участвующих в запросе:

publication_id = pub_id,

personality_id = pers_id,

work_id = w_id,

type_id = t_id,

name = n.

Сложность запроса вычисляется по формуле VL:

$$\begin{aligned}
 VL = & T(q_1) * L(q_1) + T(q_2) * L(q_2) + T(q_{2_1}) * L(q_{2_1}) + T(q_{2_2}) * L(q_{2_2}) + \\
 & T(q_{2_3}) * L(q_{2_3}) + T(q_3) * L(q_3) + T(q_{3_1}) * L(q_{3_1}) + T(q_{3_2}) * L(q_{3_2}) + \\
 & T(q_4) * L(q_4) + T(q_{4_1}) * L(q_{4_1}) + T(q_5) * L(q_5) + T(q_{5_1}) * L(q_{5_1}) + \\
 & T(q_6) * L(q_6) + T(q_{6_1}) * L(q_{6_1}) + T(q_7) * L(q_7) + T(q_{7_1}) * L(q_{7_1}) + \\
 & T(\pi_{F_9}(q_7)) * L(\pi_{F_9}(q_7)) + T(\pi_{F_8}(q_6)) * L(\pi_{F_8}(q_6)) + T(\pi_{F_7}(q_5)) * \\
 & L(\pi_{F_7}(q_5)) + T(\pi_{F_6}(q_4)) * L(\pi_{F_6}(q_4)) + T(\pi_{F_3}(q_3)) * L(\pi_{F_3}(q_3)) + \\
 & T(\pi_{F_1}(q_1)) * L(\pi_{F_1}(q_1)) + T(\sigma_{F_4}(sn)) * L(\sigma_{F_4}(sn)) + T(\sigma_{F_5}(sn)) * \\
 & L(\sigma_{F_5}(sn)) + T(\sigma_{F_4}(sn)) * L(\sigma_{F_4}(sn)) + T(\sigma_{F_5}(sn)) * L(\sigma_{F_5}(sn)) + \\
 & T(\sigma_{F_4}(sn)) * L(\sigma_{F_4}(sn)) + T(\sigma_{F_5}(sn)) * L(\sigma_{F_5}(sn)) + T(\sigma_{F_4}(sn)) *
 \end{aligned}$$

$$L(\sigma_{F_4}(sn)) + T(\sigma_{F_5}(sn)) * L(\sigma_{F_5}(sn)) + T(\sigma_{F_4}(sn)) * L(\sigma_{F_4}(sn)) + \\ T(\sigma_{F_5}(sn)) * L(\sigma_{F_5}(sn)) + T(s) * L(s),$$

где $T(r)$ – кол-во записей в таблице r ,

$L(r)$ – кол-во байт на запись в таблице r ,

$\triangleright\triangleleft$ - операция естественного соединения,

\cup - операция объединения,

$T(s)$ – кол-во записей в результирующей таблице,

$L(s)$ – кол-во байт в строке результирующей проекции,

$\pi_{F_i}(r)$ - проекция таблицы r по множеству столбцов F_i ,

$\sigma_{F_i}(r)$ - селекция таблицы r по множеству условий F_i ,

$$q_1 = \sigma_{F_2}(q_2),$$

$$q_2 = q_{2_1} \cup \pi_{F_3}(q_3),$$

$$q_{2_1} = q_{2_2} \cup \pi_{F_6}(q_4),$$

$$q_{2_2} = q_{2_3} \cup \pi_{F_7}(q_5),$$

$$q_{2_3} = \pi_{F_9}(q_7) \cup \pi_{F_8}(q_6),$$

$$q_3 = q_{3_1} \triangleright\triangleleft \sigma_{F_4}(sn),$$

$$q_3 = q_{3_1} \triangleright\triangleleft \sigma_{F_4}(sn),$$

$$q_{3_1} = q_{3_2} \triangleright\triangleleft \sigma_{F_5}(sn),$$

$$q_{3_2} = auth \triangleright\triangleleft pcw,$$

$$q_4 = q_{4_1} \triangleright\triangleleft \sigma_{F_4}(sn),$$

$$q_{4_1} = ill \triangleright\triangleleft \sigma_{F_5}(sn),$$

$$q_5 = q_{5_1} \triangleright\triangleleft \sigma_{F_4}(sn),$$

$$q_{5_1} = trans \triangleright\triangleleft \sigma_{F_5}(sn),$$

$$q_6 = q_{6_1} \triangleright\triangleleft \sigma_{F_4}(sn),$$

$$q_{6_1} = comp \triangleright\triangleleft \sigma_{F_5}(sn),$$

$$q_7 = q_{7_1} \triangleright\triangleleft \sigma_{F_4}(sn),$$

$$q_{7_1} = ed \triangleright\triangleleft \sigma_{F_5}(sn).$$

Посчитаем L для каждой таблицы в запросе:

$$L(sn) = L(sn.t_id) + \\ + L(sn.pers_id) + L(sn.n) = 10 + 10 + 32 = 52 \text{ (байт)},$$

$$L(\sigma_{F_4}(sn)) = L(sn) = 52,$$

$$L(\sigma_{F_5}(sn)) = L(sn) = 52,$$

$$L(ed) = L(ed.pers_id) + \\ + L(ed.pub_id) = 10 + 10 = 20,$$

$$L(comp) = L(comp.pers_id) + \\ + L(comp.pub_id) = 10 + 10 = 20,$$

$$L(trans) = L(trans.pers_id) + \\ + L(trans.pub_id) = 10 + 10 = 20,$$

$$L(ill) = L(ill.pers_id) + \\ + L(ill.pub_id) = 10 + 10 = 20,$$

$$L(auth) = L(auth.pers_id) + \\ + L(auth.w_id) = 10 + 10 = 20,$$

$$L(pcw) = \\ = L(pcw.pub_id) + \\ + L(pcw.w_id) = 10 + 10 = 20,$$

$$L(q_{7_1}) = L(ed) + L(\sigma_{F_5}(sn)) = 20 + 52 = 72,$$

$$L(q_7) = L(q_{7_1}) + L(\sigma_{F_4}(sn)) = 72 + 52 = 124,$$

$$L(\pi_{F_9}(q_7)) = L(q_7) + L(\text{Роль}) = 124 + 11 = 135,$$

$$L(q_{6_1}) = L(comp) + L(\sigma_{F_5}(sn)) = 20 + 52 = 72,$$

$$L(q_6) = L(q_{6_1}) + L(\sigma_{F_4}(sn)) = 72 + 52 = 124,$$

$$L(\pi_8(q_6)) = L(q_6) + L(\text{Роль}) = 124 + 11 = 135,$$

$$L(q_{5_1}) = L(trans) + L(\sigma_{F_5}(sn)) = 20 + 52 = 72,$$

$$L(q_5) = L(q_{5_1}) + L(\sigma_{F_4}(sn)) = 72 + 52 = 124,$$

$$L(\pi_{F_7}(q_5)) = L(q_6) + L(\text{Роль}) = 124 + 11 = 135,$$

$$L(q_{4_1}) = L(\text{ill}) + L(\sigma_{F_5}(sn)) = 20 + 52 = 72,$$

$$L(q_4) = L(q_{4_1}) + L(\sigma_{F_4}(sn)) = 72 + 52 = 124,$$

$$L(\pi_{F_6}(q_4)) = L(q_4) + L(\text{Роль}) = 124 + 11 = 135,$$

$$L(q_{3_2}) = L(\text{auth}) + L(\text{pcw}) = 20 + 20 = \\ = 40,$$

$$L(q_{3_1}) = L(q_{3_2}) + L(\sigma_{F_5}(sn)) = 40 + 52 = 92,$$

$$L(q_3) = L(q_{3_1}) + L(\sigma_{F_4}(sn)) = 92 + 52 = 144,$$

$$L(\pi_{F_3}(q_3)) = L(q_3) + L(\text{Роль}) - L(\text{auth.w_id}) - L(\text{pcw.w_id}) = 144 + 11 - \\ 10 - 10 = 135,$$

$$L(q_{2_3}) = L(q_{2_2}) = L(q_{2_1}) = L(q_2) = L(\pi_{F_6}(q_4)) = 135,$$

$$L(q_1) = L(q_2) = 135,$$

$$L(s) = L(sn.n) + L(sn.n) + L(\text{Роль}) = 32 + 32 + 11 = 75.$$

Для того, чтобы рассчитать T необходимы следующие данные:

Количество записей в таблице:

$$T(ed) = 600000,$$

$$T(comp) = 600000,$$

$$T(trans) = 600000,$$

$$T(ill) = 600000,$$

$$T(auth) = 600000,$$

$$T(sn) = 400000,$$

$$T(pcw) = 300000.$$

Количество различных записей в столбцах таблицы:

$$V(ed.pub_id) = 100000,$$

$$V(ed.pers_id) = 100000,$$

$$V(comp.pub_id) = 100000,$$

$$V(comp.pers_id) = 100000,$$

$$\begin{aligned}
V(\text{trans. pub_id}) &= 100000, \\
V(\text{trans. pers_id}) &= 100000, \\
V(\text{ill. pub_id}) &= 100000, \\
V(\text{ill. pers_id}) &= 100000, \\
V(\text{auth. w_id}) &= 100000, \\
V(\text{auth. pers_id}) &= 100000, \\
V(\text{sn. t_id}) &= 2, \\
V(\text{sn. pers_id}) &= 100000, \\
V(\text{sn. n}) &= 1000, \\
V(\text{pcw. pub_id}) &= 100000, \\
V(\text{pcw. w_id}) &= 100000.
\end{aligned}$$

Расчитаем T и V для результатов операций:

$$\begin{aligned}
T(\sigma_{F_4}(\text{sn})) &= 200000, \\
V(\sigma_{F_4}(\text{sn}).\text{t_id}) &= 1, \\
V(\sigma_{F_4}(\text{sn}).\text{pers_id}) &= 200000, \\
V(\sigma_{F_4}(\text{sn}).\text{n}) &= 1000, \\
T(\sigma_{F_5}(\text{sn})) &= 200000, \\
V(\sigma_{F_5}(\text{sn}).\text{type_id}) &= 1, \\
V(\sigma_{F_5}(\text{sn}).\text{pers_id}) &= 200000, \\
V(\sigma_{F_5}(\text{sn}).\text{n}) &= 1000,
\end{aligned}$$

$$\begin{aligned}
T(q_{3_2}) &= \frac{T(\text{auth}) * T(\text{pcw})}{\max(V(\text{auth. w_id}), V(\text{pcw. w_id}))} = \frac{600000 * 300000}{\max(100000, 100000)} \\
&= 1800000,
\end{aligned}$$

$$V(q_{3_2}.\text{pers_id}) = 100000,$$

$$\begin{aligned}
T(q_{3_1}) &= \frac{T(q_{3_2}) * T(\sigma_{F_5}(\text{sn}))}{\max(V(q_{3_2}.\text{pers_id}), V(\sigma_{F_5}(\text{sn}).\text{pers_id}))} \\
&= \frac{1800000 * 200000}{\max(100000, 200000)} = 1800000,
\end{aligned}$$

$$V(q_{3_1}.\text{pers_id}) = 100000,$$

$$T(q_3) = \frac{T(q_{3_1}) * T(\sigma_{F_4}(sn))}{\max(V(q_{3_1}.pers_id), V(\sigma_{F_4}(sn).pers_id))} = \frac{1800000 * 200000}{\max(100000, 200000)}$$

$$= 1800000,$$

$$T(q_{4_1}) = \frac{T(ill) * T(\sigma_{F_5}(sn))}{\max(V(ill.pers_id), V(\sigma_{F_5}(sn).pers_id))} = \frac{600000 * 200000}{\max(100000, 200000)}$$

$$= 600000,$$

$$V(q_{4_1}.pers_id) = 100000,$$

$$T(q_4) = \frac{T(q_{4_1}) * T(\sigma_{F_4}(sn))}{\max(V(q_{4_1}.pers_id), V(\sigma_{F_4}(sn).pers_id))} = \frac{600000 * 200000}{\max(100000, 200000)}$$

$$= 600000,$$

$$T(q_{5_1}) = \frac{T(trans) * T(\sigma_{F_5}(sn))}{\max(V(trans.pers_id), V(\sigma_{F_5}(sn).pers_id))}$$

$$= \frac{600000 * 200000}{\max(100000, 200000)} = 600000,$$

$$V(q_{5_1}.pers_id) = 100000,$$

$$T(q_5) = \frac{T(q_{5_1}) * T(\sigma_{F_4}(sn))}{\max(V(q_{5_1}.pers_id), V(\sigma_{F_4}(sn).pers_id))} = \frac{600000 * 200000}{\max(100000, 200000)}$$

$$= 600000,$$

$$T(q_{6_1}) = \frac{T(comp) * T(\sigma_{F_5}(sn))}{\max(V(comp.pers_id), V(\sigma_{F_5}(sn).pers_id))}$$

$$= \frac{600000 * 200000}{\max(100000, 200000)} = 600000,$$

$$V(q_{6_1}.pers_id) = 100000,$$

$$T(q_6) = \frac{T(q_{6_1}) * T(\sigma_{F_4}(sn))}{\max(V(q_{6_1}.pers_id), V(\sigma_{F_4}(sn).pers_id))} = \frac{600000 * 200000}{\max(100000, 200000)}$$

$$= 600000,$$

$$T(q_{7_1}) = \frac{T(ed) * T(\sigma_{F_5}(sn))}{\max(V(ed.pers_id), V(\sigma_{F_5}(sn).pers_id))}$$

$$= \frac{600000 * 200000}{\max(100000, 200000)} = 600000,$$

$$V(q_{7_1}.pers_id) = 100000,$$

$$T(q_7) = \frac{T(q_{7_1}) * T(\sigma_{F_4}(sn))}{\max(V(q_{7_1}.pers_id), V(\sigma_{F_4}(sn).pers_id))} = \frac{600000 * 200000}{\max(100000, 200000)} \\ = 600000,$$

$$T(\pi_{F_9}(q_7)) = T(q_7) = 600000,$$

$$T(\pi_{F_8}(q_6)) = T(q_6) = 600000,$$

$$T(\pi_{F_7}(q_5)) = T(q_5) = 600000,$$

$$T(\pi_{F_6}(q_4)) = T(q_4) = 600000,$$

$$T(\pi_{F_3}(q_3)) = T(q_3) = 1800000,$$

$$T(q_{2_3}) = T(\pi_{F_9}(q_7)) + T(\pi_{F_8}(q_6)) = 600000 + 600000 = 1200000,$$

$$T(q_{2_2}) = T(q_{2_3}) + T(\pi_{F_7}(q_5)) = 1200000 + 600000 = 1800000,$$

$$T(q_{2_1}) = T(q_{2_2}) + T(\pi_{F_6}(q_4)) = 1800000 + 600000 = 2400000,$$

$$T(q_2) = T(q_{2_1}) + T(\pi_{F_3}(q_3)) = 2400000 + 1800000 = 4200000,$$

$$T(q_1) = 30.$$

$$T(s) = T(q_1) = 30.$$

Необходимо подставить эти значения в VL и получить результат:

$$VL = 30 * 135 + 4200000 * 135 + 2400000 * 135 + 1800000 * 135 + \\ 1200000 * 135 + 1800000 * 144 + 1800000 * 92 + 1800000 * 40 + \\ 600000 * 124 + 600000 * 72 + 600000 * 124 + 600000 * 72 + 600000 * \\ 124 + 600000 * 72 + 600000 * 124 + 600000 * 72 + 600000 * 135 + \\ 600000 * 135 + 600000 * 135 + 600000 * 135 + 1800000 * 135 + 200000 * \\ 52 + 200000 * 52 + 200000 * 52 + 200000 * 52 + 200000 * 52 + 200000 * \\ 52 + 200000 * 52 + 200000 * 52 + 200000 * 52 + 200000 * 52 + 30 * 75 = \\ 2934206300.$$

2.6.3 Оптимизированный запрос (опущенная селекция)

2.6.3.1 Текст оптимизированного запроса

Текст оптимизированной комбинации запросов с опущенной селекцией на языке SQL представлен на рисунках 2.22-2.23.

```

SELECT *, 'Редактор' AS Роль FROM editorship AS ed ;
    INNER JOIN single_names AS pnames ON pnames.personality_id = ed.personality_id
AND pnames.type_id = 0 AND ed.publication_id = 22;
    INNER JOIN single_names AS lnames ON lnames.personality_id = ed.personality_id
AND lnames.type_id = 1;
UNION ALL ;
SELECT *, 'Составитель' AS Роль FROM compilership AS comp ;
    INNER JOIN single_names AS pnames ON pnames.personality_id =
comp.personality_id AND pnames.type_id = 0 AND comp.publication_id = 22;
    INNER JOIN single_names AS lnames ON lnames.personality_id =
comp.personality_id AND lnames.type_id = 1;
UNION ALL ;
SELECT *, 'Переводчик' AS Роль FROM translatorship AS trans ;
    INNER JOIN single_names AS pnames ON pnames.personality_id =
trans.personality_id AND pnames.type_id = 0 AND trans.publication_id = 22;
    INNER JOIN single_names AS lnames ON lnames.personality_id =
trans.personality_id AND lnames.type_id = 1;
UNION ALL ;
SELECT *, 'Иллюстратор' AS Роль FROM Illustratorship AS Illustr ;
    INNER JOIN single_names AS pnames ON pnames.personality_id =
Illustr.personality_id AND pnames.type_id = 0 AND Illustr.publication_id = 22;
    INNER JOIN single_names AS lnames ON lnames.personality_id =
Illustr.personality_id AND lnames.type_id = 1;
UNION ALL ;
SELECT      auth.personality_id,      pcw.publication_id,      pnames.type_id,
pnames.personality_id,      pnames.name,      lnames.type_id,      lnames.personality_id,
lnames.name, 'Автор' AS Роль FROM authorship AS auth ;
    INNER JOIN publication_contains_work AS pcw ON auth.work_id = pcw.work_id AND
pcw.publication_id = 22;
    INNER JOIN single_names AS pnames ON pnames.personality_id =
auth.personality_id AND pnames.type_id = 0;

```

Рисунок 2.22

```

    INNER JOIN single_names AS lnames ON lnames.personality_id =
auth.personality_id AND lnames.type_id = 1;
INTO TABLE book_team
SELECT Name_b AS Фамилия, Name_a AS Имя, Роль FROM book_team

```

Рисунок 2.23

2.6.3.2 Дерево оптимизированного запроса

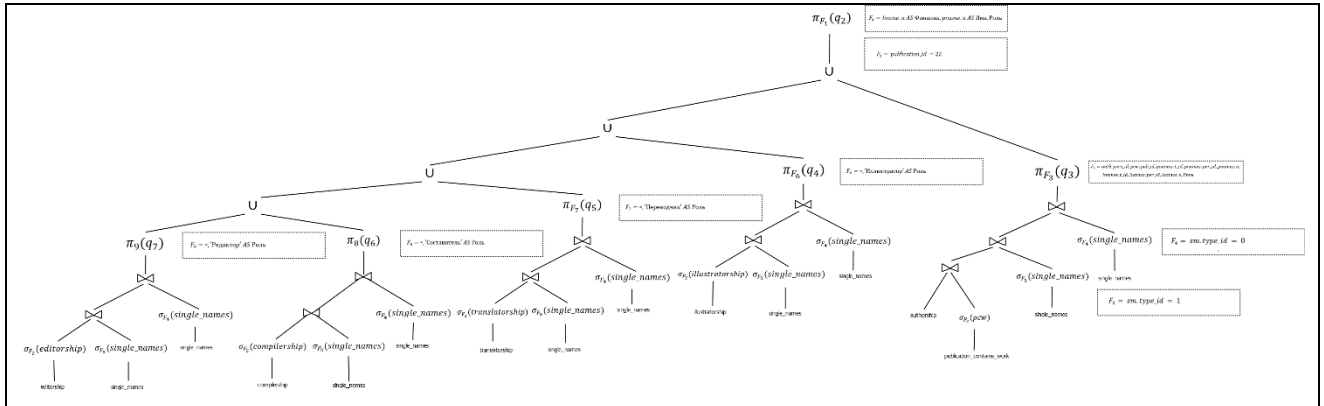


Рисунок 2.24

2.6.3.3 Расчет сложности запроса

Введем псевдонимы для таблиц, участвующих в запросе:

authorship = auth,

editorship = ed,

compilership = comp,

translatorship = trans,

illustratorship = ill,

single_names = sn,

publication_contains_work = pcw.

Введем псевдонимы для полей, участвующих в запросе:

publication_id = pub_id,

personality_id = pers_id,

work_id = w_id,

type_id = t_id,

name = n.

Сложность запроса вычисляется по формуле VL:

$$\begin{aligned}
 VL = & T(q_2) * L(q_2) + T(q_{2_1}) * L(q_{2_1}) + T(q_{2_2}) * L(q_{2_2}) + T(q_{2_3}) * \\
 & L(q_{2_3}) + T(q_3) * L(q_3) + T(q_{3_1}) * L(q_{3_1}) + T(q_{3_2}) * L(q_{3_2}) + T(q_4) * \\
 & L(q_4) + T(q_{4_1}) * L(q_{4_1}) + T(q_5) * L(q_5) + T(q_{5_1}) * L(q_{5_1}) + T(q_6) * \\
 & L(q_6) + T(q_{6_1}) * L(q_{6_1}) + T(q_7) * L(q_7) + T(q_{7_1}) * L(q_{7_1}) + T(\sigma_{F_2}(ed)) *
 \end{aligned}$$

$$\begin{aligned}
& L(\sigma_{F_2}(ed)) + T(\sigma_{F_2}(comp)) * L(\sigma_{F_2}(comp)) + T(\sigma_{F_2}(trans)) * \\
& L(\sigma_{F_2}(trans)) + T(\sigma_{F_2}(pcw)) * L(\sigma_{F_2}(pcw)) + T(\sigma_{F_2}(ed)) * L(\sigma_{F_2}(ed)) + \\
& T(\pi_{F_9}(q_7)) * L(\pi_{F_9}(q_7)) + T(\pi_{F_8}(q_6)) * L(\pi_{F_8}(q_6)) + T(\pi_{F_7}(q_5)) * \\
& L(\pi_{F_7}(q_5)) + T(\pi_{F_6}(q_4)) * L(\pi_{F_6}(q_4)) + T(\pi_{F_3}(q_3)) * L(\pi_{F_3}(q_3)) + \\
& T(\pi_{F_1}(q_1)) * L(\pi_{F_1}(q_1)) + T(\sigma_{F_4}(sn)) * L(\sigma_{F_4}(sn)) + T(\sigma_{F_5}(sn)) * \\
& L(\sigma_{F_5}(sn)) + T(\sigma_{F_4}(sn)) * L(\sigma_{F_4}(sn)) + T(\sigma_{F_5}(sn)) * L(\sigma_{F_5}(sn)) + \\
& T(\sigma_{F_4}(sn)) * L(\sigma_{F_4}(sn)) + T(\sigma_{F_5}(sn)) * L(\sigma_{F_5}(sn)) + T(\sigma_{F_4}(sn)) * \\
& L(\sigma_{F_4}(sn)) + T(\sigma_{F_5}(sn)) * L(\sigma_{F_5}(sn)) + T(\sigma_{F_4}(sn)) * L(\sigma_{F_4}(sn)) + \\
& T(\sigma_{F_5}(sn)) * L(\sigma_{F_5}(sn)) + T(s) * L(s),
\end{aligned}$$

где $T(r)$ – кол-во записей в таблице r ,

$L(r)$ – кол-во байт на запись в таблице r ,

$\triangleright\triangleleft$ - операция естественного соединения,

\cup - операция объединения,

$T(s)$ – кол-во записей в результирующей таблице,

$L(s)$ – кол-во байт в строке результирующей проекции,

$T(s)$ – кол-во записей в результирующей таблице,

$L(s)$ – кол-во байт в строке результирующей проекции,

$\pi_{F_i}(r)$ - проекция таблицы r по множеству столбцов F_i ,

$\sigma_{F_i}(r)$ - селекция таблицы r по множеству условий F_i ,

$$q_2 = q_{2_1} \cup \pi_{F_3}(q_3),$$

$$q_{2_1} = q_{2_2} \cup \pi_{F_6}(q_4),$$

$$q_{2_2} = q_{2_3} \cup \pi_{F_7}(q_5),$$

$$q_{2_3} = \pi_{F_9}(q_7) \cup \pi_{F_8}(q_6),$$

$$q_3 = q_{3_1} \triangleright\triangleleft \sigma_{F_4}(sn),$$

$$q_3 = q_{3_1} \triangleright\triangleleft \sigma_{F_4}(sn),$$

$$q_{3_1} = q_{3_2} \triangleright\triangleleft \sigma_{F_5}(sn),$$

$$q_{3_2} = auth \triangleright \triangleleft \sigma_{F_2}(pcw),$$

$$q_4 = q_{4_1} \triangleright \triangleleft \sigma_{F_4}(sn),$$

$$q_{4_1} = \sigma_{F_2}(ill) \triangleright \triangleleft \sigma_{F_5}(sn),$$

$$q_5 = q_{5_1} \triangleright \triangleleft \sigma_{F_4}(sn),$$

$$q_{5_1} = \sigma_{F_2}(trans) \triangleright \triangleleft \sigma_{F_5}(sn),$$

$$q_6 = q_{6_1} \triangleright \triangleleft \sigma_{F_4}(sn),$$

$$q_{6_1} = \sigma_{F_2}(comp) \triangleright \triangleleft \sigma_{F_5}(sn),$$

$$q_7 = q_{7_1} \triangleright \triangleleft \sigma_{F_4}(sn),$$

$$q_{7_1} = \sigma_{F_2}(ed) \triangleright \triangleleft \sigma_{F_5}(sn).$$

Посчитаем L для каждой таблицы в запросе:

$$L(sn) = L(sn.t_id) + L(sn.pers_id) + L(sn.n) = 10 + 10 + 32 = 52,$$

$$L(\sigma_{F_4}(sn)) = L(sn) = 52,$$

$$L(\sigma_{F_5}(sn)) = L(sn) = 52,$$

$$L(ed) = L(ed.pers_id) + L(ed.pub_id) = 10 + 10 = 20,$$

$$L(comp) = L(comp.pers_id) + L(comp.pub_id) = 10 + 10 = 20,$$

$$L(trans) = L(trans.pers_id) + L(trans.pub_id) = 10 + 10 = 20,$$

$$L(ill) = L(ill.pers_id) + L(ill.pub_id) = 10 + 10 = 20,$$

$$L(auth) = L(auth.pers_id) + L(auth.w_id) = 10 + 10 = 20,$$

$$L(pcw) = L(pcw.pub_id) + L(pcw.w_id) = 10 + 10 = 20,$$

$$L(\sigma_{F_2}(ed)) = L(ed) = 20,$$

$$L(\sigma_{F_2}(comp)) = L(comp) = 20,$$

$$L(\sigma_{F_2}(trans)) = L(trans) = 20,$$

$$L(\sigma_{F_2}(ill)) = L(ill) = 20,$$

$$L(\sigma_{F_2}(pcw)) = L(pcw) = 20,$$

$$L(q_{7_1}) = L(\sigma_{F_2}(ed)) + L(\sigma_{F_5}(sn)) = 20 + 52 = 72,$$

$$L(q_7) = L(q_{7_1}) + L(\sigma_{F_4}(sn)) = 72 + 52 = 124,$$

$$\begin{aligned}
L(\pi_{F_9}(q_7)) &= L(q_7) + L(\text{Роль}) = 124 + 11 = 135, \\
L(q_{6_1}) &= L(\sigma_{F_2}(\text{comp})) + L(\sigma_{F_5}(\text{sn})) = 20 + 52 = 72, \\
L(q_6) &= L(q_{6_1}) + L(\sigma_{F_4}(\text{sn})) = 72 + 52 = 124, \\
L(\pi_8(q_6)) &= L(q_6) + L(\text{Роль}) = 124 + 11 = 135, \\
L(q_{5_1}) &= L(\sigma_{F_2}(\text{trans})) + L(\sigma_{F_5}(\text{sn})) = 20 + 52 = 72, \\
L(q_5) &= L(q_{5_1}) + L(\sigma_{F_4}(\text{sn})) = 72 + 52 = 124, \\
L(\pi_{F_7}(q_5)) &= L(q_5) + L(\text{Роль}) = 124 + 11 = 135, \\
L(q_{4_1}) &= L(\sigma_{F_2}(\text{ill})) + L(\sigma_{F_5}(\text{sn})) = 20 + 52 = 72, \\
L(q_4) &= L(q_{4_1}) + L(\sigma_{F_4}(\text{sn})) = 72 + 52 = 124, \\
L(\pi_{F_6}(q_4)) &= L(q_4) + L(\text{Роль}) = 124 + 11 = 135, \\
L(q_{3_2}) &= L(\text{auth}) + L(\sigma_{F_2}(\text{pcw})) = 20 + 20 = 40, \\
L(q_{3_1}) &= L(q_{3_2}) + L(\sigma_{F_5}(\text{sn})) = 40 + 52 = 92, \\
L(q_3) &= L(q_{3_1}) + L(\sigma_{F_4}(\text{sn})) = 92 + 52 = 144, \\
L(\pi_{F_3}(q_3)) &= L(q_3) + L(\text{Роль}) - L(\text{auth.w_id}) - L(\text{pcw.w_id}) = 144 + 11 - \\
&10 - 10 = 135, \\
L(q_{2_3}) &= L(q_{2_2}) = L(q_{2_1}) = L(q_2) = L(\pi_{F_6}(q_4)) = 135, \\
L(s) &= L(\text{sn.n}) + L(\text{sn.n}) + L(\text{Роль}) = 32 + 32 + 11 = 75.
\end{aligned}$$

Для того, чтобы рассчитать T необходимы следующие данные:

Количество записей в таблице:

$$T(\text{ed}) = 500000,$$

$$T(\text{comp}) = 500000,$$

$$T(\text{trans}) = 500000,$$

$$T(\text{ill}) = 500000,$$

$$T(\text{auth}) = 500000,$$

$$T(\text{sn}) = 400000,$$

$$T(\text{pcw}) = 300000.$$

Количество различных записей в столбцах таблицы:

$$V(ed.pub_id) = 100000,$$

$$V(ed.pers_id) = 100000,$$

$$V(comp.pub_id) = 100000,$$

$$V(comp.pers_id) = 100000,$$

$$V(trans.pub_id) = 100000,$$

$$V(trans.pers_id) = 100000,$$

$$V(ill.pub_id) = 100000,$$

$$V(ill.pers_id) = 100000,$$

$$V(auth.w_id) = 100000,$$

$$V(auth.pers_id) = 100000,$$

$$V(sn.t_id) = 2,$$

$$V(sn.pers_id) = 100000,$$

$$V(sn.n) = 1000,$$

$$V(pcw.pub_id) = 100000,$$

$$V(pcw.w_id) = 100000.$$

Расчитаем T и V для результатов операций:

$$T(\sigma_{F_2}(ed)) = 5,$$

$$T(\sigma_{F_2}(comp)) = 5,$$

$$T(\sigma_{F_2}(trans)) = 5,$$

$$T(\sigma_{F_2}(ill)) = 5,$$

$$T(\sigma_{F_2}(pcw)) = 2,$$

$$T(\sigma_{F_4}(sn)) = 200000,$$

$$V((\sigma_{F_2}(ed)).pub_id) = 1,$$

$$V((\sigma_{F_2}(ed)).pers_id) = 5,$$

$$V((\sigma_{F_2}(comp)).pub_id) = 1,$$

$$V((\sigma_{F_2}(comp)).pers_id) = 5,$$

$$V((\sigma_{F_2}(trans)).pub_id) = 1,$$

$$V((\sigma_{F_2}(trans)).pers_id) = 5,$$

$$V((\sigma_{F_2}(ill)).pub_id) = 1,$$

$$V((\sigma_{F_2}(ill)).pers_id) = 5,$$

$$V((\sigma_{F_2}(pcw)).pub_id) = 1,$$

$$V((\sigma_{F_2}(pcw)).w_id) = 2,$$

$$V((\sigma_{F_4}(sn)).t_id) = 1,$$

$$V((\sigma_{F_4}(sn)).pers_id) = 200000,$$

$$V((\sigma_{F_4}(sn)).n) = 1000,$$

$$T(\sigma_{F_5}(sn)) = 200000,$$

$$V((\sigma_{F_5}(sn)).type_id) = 1,$$

$$V((\sigma_{F_5}(sn)).pers_id) = 200000,$$

$$V((\sigma_{F_5}(sn)).n) = 1000,$$

$$T(q_{3_2}) = \frac{T(auth) * T(\sigma_{F_2}(pcw))}{\max(V(auth.w_id), V(\sigma_{F_2}(pcw)).w_id)} = \frac{500000 * 2}{\max(100000, 2)} = 10,$$

$$V(q_{3_2}.pers_id) = 10,$$

$$\begin{aligned} T(q_{3_1}) &= \frac{T(q_{3_2}) * T(\sigma_{F_5}(sn))}{\max(V(\sigma_{F_2}(q_{3_2}).pers_id), V(\sigma_{F_5}(sn).pers_id))} \\ &= \frac{10 * 200000}{\max(10, 200000)} = 10, \end{aligned}$$

$$V(q_{3_1}.pers_id) = 10,$$

$$\begin{aligned} T(q_3) &= \frac{T(q_{3_1}) * T(\sigma_{F_4}(sn))}{\max(V(q_{3_1}.pers_id), V(\sigma_{F_4}(sn).pers_id))} = \frac{10 * 200000}{\max(10, 200000)} \\ &= 10, \end{aligned}$$

$$\begin{aligned} T(q_{4_1}) &= \frac{T(\sigma_{F_2}(ill)) * T(\sigma_{F_5}(sn))}{\max(V(\sigma_{F_2}(ill)).pers_id), V(\sigma_{F_5}(sn).pers_id))} = \frac{5 * 200000}{\max(5, 200000)} \\ &= 5, \end{aligned}$$

$$V(q_{4_1}.pers_id) = 5,$$

$$T(q_4) = \frac{T(q_{4_1}) * T(\sigma_{F_4}(sn))}{\max(V(q_{4_1}.pers_id), V(\sigma_{F_4}(sn).pers_id))} = \frac{5 * 200000}{\max(5, 200000)} = 5,$$

$$\begin{aligned} T(q_{5_1}) &= \frac{T(\sigma_{F_2}(trans)) * T(\sigma_{F_5}(sn))}{\max(V(\sigma_{F_2}(trans).pers_id), V(\sigma_{F_5}(sn).pers_id))} \\ &= \frac{5 * 200000}{\max(5, 200000)} = 5, \end{aligned}$$

$$V(q_{5_1}.pers_id) = 5,$$

$$T(q_5) = \frac{T(q_{5_1}) * T(\sigma_{F_4}(sn))}{\max(V(q_{5_1}.pers_id), V(\sigma_{F_4}(sn).pers_id))} = \frac{5 * 200000}{\max(5, 200000)} = 5,$$

$$\begin{aligned} T(q_{6_1}) &= \frac{T(\sigma_{F_2}(comp)) * T(\sigma_{F_5}(sn))}{\max(V(\sigma_{F_2}(comp).pers_id), V(\sigma_{F_5}(sn).pers_id))} \\ &= \frac{5 * 200000}{\max(5, 200000)} = 5, \end{aligned}$$

$$V(q_{6_1}.pers_id) = 5,$$

$$T(q_6) = \frac{T(q_{6_1}) * T(\sigma_{F_4}(sn))}{\max(V(q_{6_1}.pers_id), V(\sigma_{F_4}(sn).pers_id))} = \frac{5 * 200000}{\max(5, 200000)} = 5,$$

$$\begin{aligned} T(q_{7_1}) &= \frac{T(\sigma_{F_2}(ed)) * T(\sigma_{F_5}(sn))}{\max(V(\sigma_{F_2}(ed).pers_id), V(\sigma_{F_5}(sn).pers_id))} = \frac{5 * 200000}{\max(5, 200000)} \\ &= 5, \end{aligned}$$

$$V(q_{7_1}.pers_id) = 5,$$

$$T(q_7) = \frac{T(q_{7_1}) * T(\sigma_{F_4}(sn))}{\max(V(q_{7_1}.pers_id), V(\sigma_{F_4}(sn).pers_id))} = \frac{5 * 200000}{\max(5, 200000)} = 5,$$

$$T(\pi_{F_9}(q_7)) = T(q_7) = 5,$$

$$T(\pi_{F_8}(q_6)) = T(q_6) = 5,$$

$$T(\pi_{F_7}(q_5)) = T(q_5) = 5,$$

$$T(\pi_{F_6}(q_4)) = T(q_4) = 5,$$

$$T(\pi_{F_3}(q_3)) = T(q_3) = 12,$$

$$T(q_{2_3}) = T(\pi_{F_9}(q_7)) + T(\pi_{F_8}(q_6)) = 5 + 5 = 10,$$

$$T(q_{2_2}) = T(q_{2_3}) + T(\pi_{F_7}(q_5)) = 10 + 5 = 15,$$

$$T(q_{2_1}) = T(q_{2_2}) + T(\pi_{F_6}(q_4)) = 15 + 5 = 20,$$

$$T(q_2) = T(q_{2_1}) + T(\pi_{F_3}(q_3)) = 20 + 10 = 30,$$

$$T(s) = T(q_2) = 30.$$

Необходимо подставить эти значения в VL и получить результат:

$$\begin{aligned} VL = & 30 * 135 + 20 * 135 + 15 * 135 + 10 * 135 + 10 * 144 + 10 * 92 + 10 * \\ & 40 + 5 * 124 + 5 * 72 + 5 * 124 + 5 * 72 + 5 * 124 + 5 * 72 + 5 * 124 + 5 * \\ & 72 + 5 * 20 + 5 * 20 + 5 * 20 + 2 * 20 + 5 * 20 + 5 * 135 + 5 * 135 + 5 * \\ & 135 + 5 * 135 + 10 * 135 + 200000 * 52 + 200000 * 52 + 200000 * 52 + \\ & 200000 * 52 + 200000 * 52 + 200000 * 52 + 200000 * 52 + 200000 * 52 + \\ & 200000 * 52 + 200000 * 52 + 30 * 75 = 4050 + 2700 + 2025 + 1350 + \\ & 1440 + 920 + 400 + 2480 + 1440 + 400 + 40 + 2700 + 1350 + \\ & 104000000 + 2250 = 104023545. \end{aligned}$$

2.6.4 Оптимизированный запрос (опущенная проекция)

2.6.4.1 Текст оптимизированного запроса

Текст оптимизированной комбинации запросов с опущенной проекцией на языке SQL представлен на рисунках 2.25-2.26.

```
SELECT lnames.name AS Фамилия, pnames.name AS Имя, 'Редактор' AS Роль FROM editorship AS ed
;
    INNER JOIN single_names AS pnames ON pnames.personality_id = ed.personality_id AND
pnames.type_id = 0 AND ed.publication_id = 22;
    INNER JOIN single_names AS lnames ON lnames.personality_id = ed.personality_id AND
lnames.type_id = 1;
UNION ALL ;
SELECT lnames.name AS Фамилия, pnames.name AS Имя, 'Составитель' AS Роль FROM compilership
AS comp ;
    INNER JOIN single_names AS pnames ON pnames.personality_id = comp.personality_id AND
pnames.type_id = 0 AND comp.publication_id = 22;
    INNER JOIN single_names AS lnames ON lnames.personality_id = comp.personality_id AND
lnames.type_id = 1;
```

```

UNION ALL ;
SELECT lnames.name AS Фамилия, pnames.name AS Имя, 'Переводчик' AS Роль FROM translatorship
AS trans ;
    INNER JOIN single_names AS pnames ON pnames.personality_id = trans.personality_id AND
pnames.type_id = 0 AND trans.publication_id = 22;
    INNER JOIN single_names AS lnames ON lnames.personality_id = trans.personality_id AND
lnames.type_id = 1;
UNION ALL ;
SELECT lnames.name AS Фамилия, pnames.name AS Имя, 'Иллюстратор' AS Роль FROM
Illustratorship AS Illustr ;
    INNER JOIN single_names AS pnames ON pnames.personality_id = Illustr.personality_id
AND pnames.type_id = 0 AND Illustr.publication_id = 22;
    INNER JOIN single_names AS lnames ON lnames.personality_id = Illustr.personality_id
AND lnames.type_id = 1;
UNION ALL ;
SELECT lnames.name AS Фамилия, pnames.name AS Имя, 'Автор' AS Роль FROM authorship AS auth ;

```

Рисунок 2.25

```

INNER JOIN publication_contains_work AS pcw ON auth.work_id = pcw.work_id AND
pcw.publication_id = 22;
INNER JOIN single_names AS pnames ON pnames.personality_id = auth.personality_id AND
pnames.type_id = 0;
INNER JOIN single_names AS lnames ON lnames.personality_id = auth.personality_id AND
lnames.type_id = 1;

```

Рисунок 2.26

2.6.4.2 Дерево оптимизированного запроса

Дерево оптимизированного запроса представлено на рисунке 2.27.

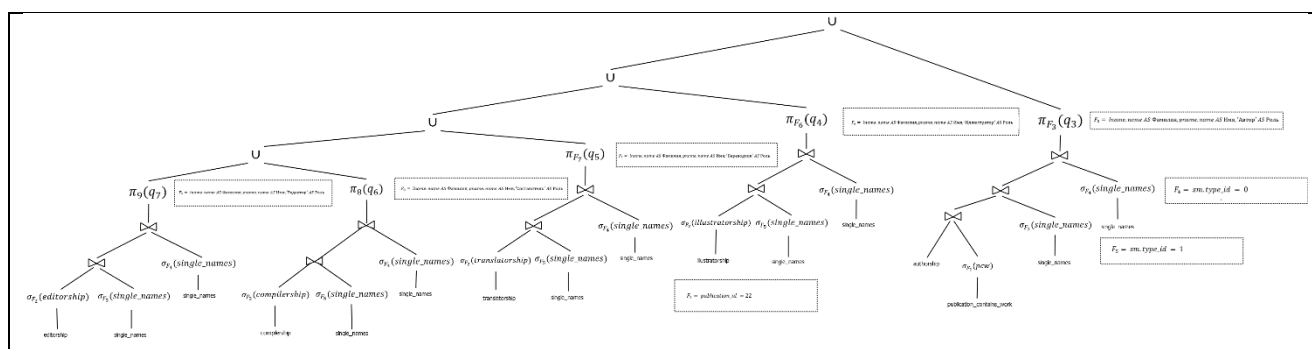


Рисунок 2.27

2.6.4.3 Расчет сложности запроса

Введем псевдонимы для таблиц, участвующих в запросе:

authorship = auth,

editorship = ed,

compilership = comp,

translatorship = trans,

illustratorship = ill,

single_names = sn,

publication_contains_work = pcw.

Введем псевдонимы для полей, участвующих в запросе:

publication_id = pub_id,

personality_id = pers_id,

work_id = w_id,

type_id = t_id,

name = n.

Дальнейший спуск проекций ниже уровня объединений нецелесообразен, так как расходы на создание промежуточных таблиц превысит выигрыш от оптимизации.

Сложность запроса вычисляется по формуле VL:

$$\begin{aligned} VL = & T(q_{2_1}) * L(q_{2_1}) + T(q_{2_2}) * L(q_{2_2}) + T(q_{2_3}) * L(q_{2_3}) + T(q_3) * \\ & L(q_3) + T(q_{3_1}) * L(q_{3_1}) + T(q_{3_2}) * L(q_{3_2}) + T(q_4) * L(q_4) + T(q_{4_1}) * \\ & L(q_{4_1}) + T(q_5) * L(q_5) + T(q_{5_1}) * L(q_{5_1}) + T(q_6) * L(q_6) + T(q_{6_1}) * \\ & L(q_{6_1}) + T(q_7) * L(q_7) + T(q_{7_1}) * L(q_{7_1}) + T(\sigma_{F_2}(ed)) * L(\sigma_{F_2}(ed)) + \\ & T(\sigma_{F_2}(comp)) * L(\sigma_{F_2}(comp)) + T(\sigma_{F_2}(trans)) * L(\sigma_{F_2}(trans)) + \\ & T(\sigma_{F_2}(pcw)) * L(\sigma_{F_2}(pcw)) + T(\sigma_{F_2}(ed)) * L(\sigma_{F_2}(ed)) + T(\pi_{F_9}(q_7)) * \\ & L(\pi_{F_9}(q_7)) + T(\pi_{F_8}(q_6)) * L(\pi_{F_8}(q_6)) + T(\pi_{F_7}(q_5)) * L(\pi_{F_7}(q_5)) + \\ & T(\pi_{F_6}(q_4)) * L(\pi_{F_6}(q_4)) + T(\pi_{F_3}(q_3)) * L(\pi_{F_3}(q_3)) + T(\pi_{F_1}(q_1)) * \\ & L(\pi_{F_1}(q_1)) + T(\sigma_{F_4}(sn)) * L(\sigma_{F_4}(sn)) + T(\sigma_{F_5}(sn)) * L(\sigma_{F_5}(sn)) + \end{aligned}$$

$$\begin{aligned}
& T(\sigma_{F_4}(sn)) * L(\sigma_{F_4}(sn)) + T(\sigma_{F_5}(sn)) * L(\sigma_{F_5}(sn)) + T(\sigma_{F_4}(sn)) * \\
& L(\sigma_{F_4}(sn)) + T(\sigma_{F_5}(sn)) * L(\sigma_{F_5}(sn)) + T(\sigma_{F_4}(sn)) * L(\sigma_{F_4}(sn)) + \\
& T(\sigma_{F_5}(sn)) * L(\sigma_{F_5}(sn)) + T(\sigma_{F_4}(sn)) * L(\sigma_{F_4}(sn)) + T(\sigma_{F_5}(sn)) * \\
& L(\sigma_{F_5}(sn)) + T(s) * L(s),
\end{aligned}$$

где $T(r)$ – кол-во записей в таблице r ,

$L(r)$ – кол-во байт на запись в таблице r ,

$\triangleright\triangleleft$ - операция естественного соединения,

\cup - операция объединения,

$T(s)$ – кол-во записей в результирующей таблице,

$L(s)$ – кол-во байт в строке результирующей проекции,

$T(s)$ – кол-во записей в результирующей таблице,

$L(s)$ – кол-во байт в строке результирующей проекции,

$\pi_{F_i}(r)$ - проекция таблицы r по множеству столбцов F_i ,

$\sigma_{F_i}(r)$ - селекция таблицы r по множеству условий F_i ,

$$s = q_{2_1} \cup \pi_{F_3}(q_3),$$

$$q_{2_1} = q_{2_2} \cup \pi_{F_6}(q_4),$$

$$q_{2_2} = q_{2_3} \cup \pi_{F_7}(q_5),$$

$$q_{2_3} = \pi_{F_9}(q_7) \cup \pi_{F_8}(q_6),$$

$$q_3 = q_{3_1} \triangleright\triangleleft \sigma_{F_4}(sn),$$

$$q_3 = q_{3_1} \triangleright\triangleleft \sigma_{F_4}(sn),$$

$$q_{3_1} = q_{3_2} \triangleright\triangleleft \sigma_{F_5}(sn),$$

$$q_{3_2} = auth \triangleright\triangleleft \sigma_{F_2}(pcw),$$

$$q_4 = q_{4_1} \triangleright\triangleleft \sigma_{F_4}(sn),$$

$$q_{4_1} = \sigma_{F_2}(ill) \triangleright\triangleleft \sigma_{F_5}(sn),$$

$$q_5 = q_{5_1} \triangleright\triangleleft \sigma_{F_4}(sn),$$

$$q_{5_1} = \sigma_{F_2}(trans) \triangleright\triangleleft \sigma_{F_5}(sn),$$

$$q_6 = q_{6_1} \triangleright\triangleleft \sigma_{F_4}(sn),$$

$$q_{6_1} = \sigma_{F_2}(\text{comp}) \triangleright \triangleleft \sigma_{F_5}(sn),$$

$$q_7 = q_{7_1} \triangleright \triangleleft \sigma_{F_4}(sn),$$

$$q_{7_1} = \sigma_{F_2}(\text{ed}) \triangleright \triangleleft \sigma_{F_5}(sn).$$

Посчитаем L для каждой таблицы в запросе:

$$L(sn) = L(sn.t_id) + L(sn.pers_id) + L(sn.n) = 10 + 10 + 32 = 52,$$

$$L(\sigma_{F_4}(sn)) = L(sn) = 52,$$

$$L(\sigma_{F_5}(sn)) = L(sn) = 52,$$

$$L(ed) = L(ed.pers_id) + L(ed.pub_id) = 10 + 10 = 20,$$

$$L(comp) = L(comp.pers_id) + L(comp.pub_id) = 10 + 10 = 20,$$

$$L(trans) = L(trans.pers_id) + L(trans.pub_id) = 10 + 10 = 20,$$

$$L(ill) = L(ill.pers_id) + L(ill.pub_id) = 10 + 10 = 20,$$

$$L(auth) = L(auth.pers_id) + L(auth.w_id) = 10 + 10 = 20,$$

$$L(pcw) = L(pcw.pub_id) + L(pcw.w_id) = 10 + 10 = 20,$$

$$L(\sigma_{F_2}(ed)) = L(ed) = 20,$$

$$L(\sigma_{F_2}(comp)) = L(comp) = 20,$$

$$L(\sigma_{F_2}(trans)) = L(trans) = 20,$$

$$L(\sigma_{F_2}(ill)) = L(ill) = 20,$$

$$L(\sigma_{F_2}(pcw)) = L(pcw) = 20,$$

$$L(q_{7_1}) = L(\sigma_{F_2}(ed)) + L(\sigma_{F_5}(sn)) = 20 + 52 = 72,$$

$$L(q_7) = L(q_{7_1}) + L(\sigma_{F_4}(sn)) = 72 + 52 = 124,$$

$$L(\pi_{F_9}(q_7)) = L(sn.n) + L(sn.n) + L(\text{Роль}) = 32 + 32 + 11 = 75,$$

$$L(q_{6_1}) = L(\sigma_{F_2}(comp)) + L(\sigma_{F_5}(sn)) = 20 + 52 = 72,$$

$$L(q_6) = L(q_{6_1}) + L(\sigma_{F_4}(sn)) = 72 + 52 = 124,$$

$$L(\pi_8(q_6)) = L(sn.n) + L(sn.n) + L(\text{Роль}) = 32 + 32 + 11 = 75,$$

$$L(q_{5_1}) = L(\sigma_{F_2}(trans)) + L(\sigma_{F_5}(sn)) = 20 + 52 = 72,$$

$$L(q_5) = L(q_{5_1}) + L(\sigma_{F_4}(sn)) = 72 + 52 = 124,$$

$$L(\pi_{F_7}(q_5)) = L(sn.n) + L(sn.n) + L(Роль) = 32 + 32 + 11 = 75,$$

$$L(q_{4_1}) = L(\sigma_{F_2}(ill)) + L(\sigma_{F_5}(sn)) = 20 + 52 = 72,$$

$$L(q_4) = L(q_{4_1}) + L(\sigma_{F_4}(sn)) = 72 + 52 = 124,$$

$$L(\pi_{F_6}(q_4)) = L(sn.n) + L(sn.n) + L(Роль) = 32 + 32 + 11 = 75,$$

$$L(q_{3_2}) = L(auth) + L(\sigma_{F_2}(pcw)) = 20 + 20 = 40,$$

$$L(q_{3_1}) = L(q_{3_2}) + L(\sigma_{F_5}(sn)) = 40 + 52 = 92,$$

$$L(q_3) = L(q_{3_1}) + L(\sigma_{F_4}(sn)) = 92 + 52 = 144,$$

$$L(\pi_{F_3}(q_3)) = L(sn.n) + L(sn.n) + L(Роль) = 32 + 32 + 11 = 75,$$

$$L(q_{2_3}) = L(q_{2_2}) = L(q_{2_1}) = L(q_2) = L(s) = L(sn.n) + L(sn.n) + L(Роль) = 75.$$

Для того, чтобы рассчитать T необходимы следующие данные:

Количество записей в таблице:

$$T(ed) = 500000,$$

$$T(comp) = 500000,$$

$$T(trans) = 500000,$$

$$T(ill) = 500000,$$

$$T(auth) = 500000,$$

$$T(sn) = 400000,$$

$$T(pcw) = 300000.$$

Количество различных записей в столбцах таблицы:

$$V(ed.pub_id) = 100000,$$

$$V(ed.pers_id) = 100000,$$

$$V(comp.pub_id) = 100000,$$

$$V(comp.pers_id) = 100000,$$

$$V(trans.pub_id) = 100000,$$

$$V(trans.pers_id) = 100000,$$

$$\begin{aligned}
V(ill.pub_id) &= 100000, \\
V(ill.pers_id) &= 100000, \\
V(auth.w_id) &= 100000, \\
V(auth.pers_id) &= 100000, \\
V(sn.t_id) &= 2, \\
V(sn.pers_id) &= 100000, \\
V(sn.n) &= 1000, \\
V(pcw.pub_id) &= 100000, \\
V(pcw.w_id) &= 100000.
\end{aligned}$$

Расчитаем T и V для результатов операций:

$$\begin{aligned}
T(\sigma_{F_2}(ed)) &= 5, \\
T(\sigma_{F_2}(comp)) &= 5, \\
T(\sigma_{F_2}(trans)) &= 5, \\
T(\sigma_{F_2}(ill)) &= 5, \\
T(\sigma_{F_2}(pcw)) &= 2, \\
T(\sigma_{F_4}(sn)) &= 200000, \\
V((\sigma_{F_2}(ed)).pub_id) &= 1, \\
V((\sigma_{F_2}(ed)).pers_id) &= 5, \\
V((\sigma_{F_2}(comp)).pub_id) &= 1, \\
V((\sigma_{F_2}(comp)).pers_id) &= 5, \\
V((\sigma_{F_2}(trans)).pub_id) &= 1, \\
V((\sigma_{F_2}(trans)).pers_id) &= 5, \\
V((\sigma_{F_2}(ill)).pub_id) &= 1, \\
V((\sigma_{F_2}(ill)).pers_id) &= 5, \\
V((\sigma_{F_2}(pcw)).pub_id) &= 1, \\
V((\sigma_{F_2}(pcw)).w_id) &= 2, \\
V((\sigma_{F_4}(sn)).t_id) &= 1,
\end{aligned}$$

$$V((\sigma_{F_4}(sn)).pers_id) = 200000,$$

$$V((\sigma_{F_4}(sn)).n) = 1000,$$

$$T(\sigma_{F_5}(sn)) = 200000,$$

$$V((\sigma_{F_5}(sn)).type_id) = 1,$$

$$V((\sigma_{F_5}(sn)).pers_id) = 200000,$$

$$V((\sigma_{F_5}(sn)).n) = 1000,$$

$$T(q_{3_2}) = \frac{T(auth) * T(\sigma_{F_2}(pcw))}{\max(V(auth.w_id), V(\sigma_{F_2}(pcw)).w_id)} = \frac{500000 * 2}{\max(100000, 2)} = 10,$$

$$V(q_{3_2}.pers_id) = 10,$$

$$\begin{aligned} T(q_{3_1}) &= \frac{T(q_{3_2}) * T(\sigma_{F_5}(sn))}{\max(V(\sigma_{F_2}(q_{3_2}).pers_id), V(\sigma_{F_5}(sn).pers_id))} \\ &= \frac{10 * 200000}{\max(10, 200000)} = 10, \end{aligned}$$

$$V(q_{3_1}.pers_id) = 10,$$

$$\begin{aligned} T(q_3) &= \frac{T(q_{3_1}) * T(\sigma_{F_4}(sn))}{\max(V(q_{3_1}.pers_id), V(\sigma_{F_4}(sn).pers_id))} = \frac{10 * 200000}{\max(10, 200000)} \\ &= 10, \end{aligned}$$

$$\begin{aligned} T(q_{4_1}) &= \frac{T(\sigma_{F_2}(ill)) * T(\sigma_{F_5}(sn))}{\max(V(\sigma_{F_2}(ill)).pers_id), V(\sigma_{F_5}(sn).pers_id))} = \frac{5 * 200000}{\max(5, 200000)} \\ &= 5, \end{aligned}$$

$$V(q_{4_1}.pers_id) = 5,$$

$$T(q_4) = \frac{T(q_{4_1}) * T(\sigma_{F_4}(sn))}{\max(V(q_{4_1}.pers_id), V(\sigma_{F_4}(sn).pers_id))} = \frac{5 * 200000}{\max(5, 200000)} = 5,$$

$$\begin{aligned} T(q_{5_1}) &= \frac{T(\sigma_{F_2}(trans)) * T(\sigma_{F_5}(sn))}{\max(V(\sigma_{F_2}(trans)).pers_id), V(\sigma_{F_5}(sn).pers_id))} \\ &= \frac{5 * 200000}{\max(5, 200000)} = 5, \end{aligned}$$

$$V(q_{5_1}.pers_id) = 5,$$

$$T(q_5) = \frac{T(q_{5_1}) * T(\sigma_{F_4}(sn))}{\max(V(q_{5_1}.pers_id), V(\sigma_{F_4}(sn).pers_id))} = \frac{5 * 200000}{\max(5, 200000)} = 5,$$

$$T(q_{6_1}) = \frac{T(\sigma_{F_2}(comp)) * T(\sigma_{F_5}(sn))}{\max(V(\sigma_{F_2}(comp)).pers_id, V(\sigma_{F_5}(sn).pers_id))}$$

$$= \frac{5 * 200000}{\max(5, 200000)} = 5,$$

$$V(q_{6_1}.pers_id) = 5,$$

$$T(q_6) = \frac{T(q_{6_1}) * T(\sigma_{F_4}(sn))}{\max(V(q_{6_1}.pers_id), V(\sigma_{F_4}(sn).pers_id))} = \frac{5 * 200000}{\max(5, 200000)} = 5,$$

$$T(q_{7_1}) = \frac{T(\sigma_{F_2}(ed)) * T(\sigma_{F_5}(sn))}{\max(V(\sigma_{F_2}(ed)).pers_id, V(\sigma_{F_5}(sn).pers_id))} = \frac{5 * 200000}{\max(5, 200000)}$$

$$= 5,$$

$$V(q_{7_1}.pers_id) = 5,$$

$$T(q_7) = \frac{T(q_{7_1}) * T(\sigma_{F_4}(sn))}{\max(V(q_{7_1}.pers_id), V(\sigma_{F_4}(sn).pers_id))} = \frac{5 * 200000}{\max(5, 200000)} = 5,$$

$$T(\pi_{F_9}(q_7)) = T(q_7) = 5,$$

$$T(\pi_{F_8}(q_6)) = T(q_6) = 5,$$

$$T(\pi_{F_7}(q_5)) = T(q_5) = 5,$$

$$T(\pi_{F_6}(q_4)) = T(q_4) = 5,$$

$$T(\pi_{F_3}(q_3)) = T(q_3) = 12,$$

$$T(q_{2_3}) = T(\pi_{F_9}(q_7)) + T(\pi_{F_8}(q_6)) = 5 + 5 = 10,$$

$$T(q_{2_2}) = T(q_{2_3}) + T(\pi_{F_7}(q_5)) = 10 + 5 = 15,$$

$$T(q_{2_1}) = T(q_{2_2}) + T(\pi_{F_6}(q_4)) = 15 + 5 = 20,$$

$$T(s) = T(q_{2_1}) + T(\pi_{F_3}(q_3)) = 20 + 10 = 30.$$

Необходимо подставить эти значения в VL и получить результат:

$$\begin{aligned}
VL = & 30 * 75 + 20 * 75 + 15 * 75 + 10 * 75 + 10 * 144 + 10 * 92 + 10 * 40 + \\
& 5 * 124 + 5 * 72 + 5 * 124 + 5 * 72 + 5 * 124 + 5 * 72 + 5 * 124 + 5 * 72 + \\
& 5 * 20 + 5 * 20 + 5 * 20 + 2 * 20 + 5 * 20 + 5 * 135 + 5 * 135 + 5 * 135 + 5 * \\
& 135 + 10 * 135 + 200000 * 52 + 200000 * 52 + 200000 * 52 + 200000 * \\
& 52 + 200000 * 52 + 200000 * 52 + 200000 * 52 + 200000 * 52 + 200000 * \\
& 52 + 200000 * 52 = 104016795.
\end{aligned}$$

3 Рабочий проект

3.1 Разработка приложения в FoxPro

Чтобы создать приложение в VisualFoxpro необходимо прежде всего создать новый проект. Это можно сделать из меню File->New.

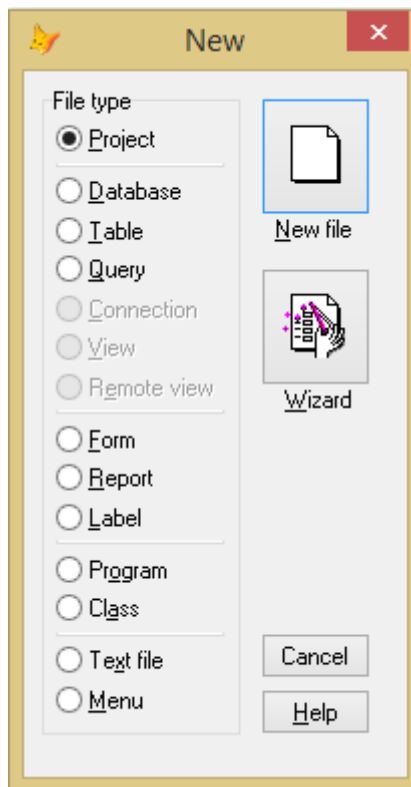


Рисунок 3.1

Необходимо отметить пункт Project и нажать кнопку NewFile.

После этого необходимо создать базу данных. Для этого надо в меню File->New отметить пункт Database и нажать кнопку NewFile. После этого в окне ProjectManager нужно выбрать созданную базу и нажать кнопку Open, чтобы открыть ее для дальнейшей работы. После этого необходимо добавить в базу таблицы и заполнить их. Это можно сделать несколькими способами. В данной работе используется программа, написанная на языке FoxPro. Чтобы добавить в проект программный код необходимо в меню File->New отметить пункт Program и нажать кнопку NewFile. Откроется окно, в котором можно ввести текст программы. Текст некоторых вспомогательных функций приведен на рисунке 3.2.

```

FUNCTION RANDOM
* Returns a random number between any 2 numbers.
* Syntax: =RANDOM(Lower Value, Higher Value)
* Example: lnX=RANDOM(5,10)
PARAMETER anLow,anHigh
lnRnumber=INT((RAND()*(anHigh-anLow+1))+anLow)
RETURN lnRnumber

FUNCTION RAND_DATE
year = RANDOM(100, 9999)
month = RANDOM(1, 12)
DO CASE
CASE month = 2
IF (year % 100) = 0
IF (year % 400) = 0
day = RANDOM(1,29)
ELSE
day = RANDOM(1,28)
ENDIF
ELSE
IF (year % 4) = 0
day = RANDOM(1,29)
ELSE
day = RANDOM(1,28)
ENDIF
ENDIF
CASE (month = 2) OR (month = 4) OR (month = 6) OR (month = 9) OR (month = 11)
day = RANDOM(1,30)
OTHERWISE
day = RANDOM(1,31)
ENDCASE
RETURN DATE(year, month, day)

```

Рисунок 3.2

3.2 Работа с таблицами

Добавить таблицу в базу можно несколькими способами, в том числе через TableWizard или добавление пустой таблицы и редактирование ее свойств. В данной

работе используется программа на языке FoxPro. Ее текст приведен на рисунках 3.3-3.11.

```
CREATE TABLE Alternative_Title ;
( ;
    Work_ID          Numeric NOT NULL, ;
    Title            Varchar(128) NOT NULL, ;
    PRIMARY KEY STR(Work_ID)+Title TAG PK ;
);
CREATE TABLE Authorship ;
( ;
    Personality_ID   Numeric NOT NULL, ;
    Work_ID          Numeric NOT NULL, ;
    PRIMARY KEY STR(Personality_ID)+STR(Work_ID) TAG PK ;
);
CREATE TABLE Base_Object ;
( ;
    ID               Numeric NOT NULL, ;
    Type_ID         Numeric NOT NULL, ;
    Description      Memo NOT NULL, ;
    PRIMARY KEY ID TAG PK ;
);
CREATE TABLE Base_Object_Type ;
( ;
    ID               Numeric NOT NULL, ;
    Title            Varchar(20) NOT NULL, ;
    PRIMARY KEY ID TAG PK ;
);
CREATE TABLE Book ;
( ;
    ID               Numeric NOT NULL, ;
    Type_ID         Numeric NOT NULL, ;
    PRIMARY KEY ID TAG PK ;
);
CREATE TABLE Book_Type ;
( ;
    ID               Numeric NOT NULL, ;
    Title            Varchar(20) NULL, ;
    PRIMARY KEY ID TAG PK ;
);
```

Рисунок 3.3


```

CREATE TABLE Character ;
( ;
    ID                Numeric NOT NULL, ;
    Name              Varchar(128) NOT NULL, ;
    PRIMARY KEY ID TAG PK ;
);
CREATE TABLE Compilership ;
( ;
    Personality_ID    Numeric NOT NULL, ;
    Publication_ID    Numeric NOT NULL, ;
    PRIMARY KEY STR(Personality_ID)+STR(Publication_ID) TAG PK ;
);
CREATE TABLE Editorship ;
( ;
    Personality_ID    Numeric NOT NULL, ;
    Publication_ID    Numeric NOT NULL, ;
    PRIMARY KEY STR(Personality_ID)+STR(Publication_ID) TAG PK ;
);
CREATE TABLE Genre ;
( ;
    ID                Numeric NOT NULL, ;
    Title             Varchar(20) NULL, ;
    PRIMARY KEY ID TAG PK ;
);
CREATE TABLE Have_Digital_Book ;
( ;
    User_ID           Numeric NOT NULL, ;
    Book_ID           Numeric NOT NULL, ;
    PRIMARY KEY STR(User_ID)+STR(Book_ID) TAG PK ;
);
CREATE TABLE Have_Paper_Book ;
( ;
    User_ID           Numeric NOT NULL, ;
    Book_ID           Numeric NOT NULL, ;
    PRIMARY KEY STR(User_ID)+STR(Book_ID) TAG PK ;
);
CREATE TABLE Illustratorship ;
( ;
    Personality_ID    Numeric NOT NULL, ;

```

Рисунок 3.4

```

        Publication_ID      Numeric NOT NULL, ;
        PRIMARY KEY STR(Personality_ID)+STR(Publication_ID) TAG PK ;
    );
CREATE TABLE Mark ;
( ;
    User_ID                Numeric NOT NULL, ;
    Book_ID                Numeric NOT NULL, ;
    Mark                   Numeric NOT NULL, ;
    PRIMARY KEY STR(User_ID)+STR(Book_ID) TAG PK ;
);
CREATE TABLE Multiple_Names ;
( ;
    Type_ID                Numeric NOT NULL, ;
    Number                 Numeric NOT NULL, ;
    Personality_ID         Numeric NOT NULL, ;
    Name                   Varchar(32) NOT NULL, ;
    PRIMARY KEY STR(Type_ID)+STR(Personality_ID)+STR(Number) TAG PK ;
);
CREATE TABLE Multiple_Name_Types ;
( ;
    ID                     Numeric NOT NULL, ;
    Name_Title             Varchar(20) NOT NULL, ;
    PRIMARY KEY ID TAG PK ;
);
CREATE TABLE Personality ;
( ;
    ID                     Numeric NOT NULL, ;
    PRIMARY KEY ID TAG PK ;
);
CREATE TABLE Publication ;
( ;
    ID                     Numeric NOT NULL, ;
    Series_ID              Numeric NOT NULL, ;
    ISBN                   Character(13) NOT NULL, ;
    Page_Count             Numeric NOT NULL, ;
    Cover_Type             Varchar(128) NOT NULL, ;
    Weight                 Numeric NOT NULL, ;
    Width                  Numeric NOT NULL, ;
    Height                 Numeric NOT NULL, ;

```

Рисунок3.5

```

        Sleeve_Page_Count      Numeric NOT NULL, ;
        PublishingLanguage      Varchar(32) NOT NULL, ;
        Title                   Varchar(100) NOT NULL, ;
        PRIMARY KEY ID TAG PK, ;
        UNIQUE ISBN TAG AK1 ;
);
CREATE TABLE Publication_Contains_Work ;
( ;
        Publication_ID          Numeric NOT NULL, ;
        Work_ID                 Numeric NOT NULL, ;
        PRIMARY KEY STR(Publication_ID)+STR(Work_ID) TAG PK ;
);
CREATE TABLE Publisher ;
( ;
        ID                     Numeric NOT NULL, ;
        Title                   Varchar(100) NOT NULL, ;
        Website                 Varchar(100) NOT NULL, ;
        PRIMARY KEY ID TAG PK ;
);
CREATE TABLE Publisher_Publishes_Series ;
( ;
        Series_ID              Numeric NOT NULL, ;
        Publisher_ID           Numeric NOT NULL, ;
        PRIMARY KEY STR(Series_ID)+STR(Publisher_ID) TAG PK ;
);
CREATE TABLE Read ;
( ;
        User_ID                Numeric NOT NULL, ;
        Book_ID                Numeric NOT NULL, ;
        PRIMARY KEY STR(User_ID)+STR(Book_ID) TAG PK ;
);
CREATE TABLE Review ;
( ;
        User_ID                Numeric NOT NULL, ;
        Book_ID                Numeric NOT NULL, ;
        Text                   Memo NOT NULL, ;
        PRIMARY KEY STR(User_ID)+STR(Book_ID) TAG PK ;
);

```

Рисунок 3.6

```

CREATE TABLE Series ;
( ;
    ID                Numeric NOT NULL, ;
    Title             Varchar(100) NOT NULL, ;
    PRIMARY KEY ID TAG PK ;
);
CREATE TABLE Single_Names ;
( ;
    Type_ID           Numeric NOT NULL, ;
    Personality_ID    Numeric NOT NULL, ;
    Name              Varchar(32) NOT NULL, ;
    PRIMARY KEY STR(Type_ID)+STR(Personality_ID) TAG PK;
);
CREATE TABLE Single_Name_Types ;
( ;
    ID                Numeric NOT NULL, ;
    Name_Title        Varchar(20) NOT NULL, ;
    PRIMARY KEY ID TAG PK ;
);
CREATE TABLE Tag ;
( ;
    User_ID           Numeric NOT NULL, ;
    Book_ID           Numeric NOT NULL, ;
    Tag               Varchar(20) NOT NULL, ;
    PRIMARY KEY STR(User_ID)+STR(Book_ID) TAG PK ;
);
CREATE TABLE Translatorship ;
( ;
    Personality_ID    Numeric NOT NULL, ;
    Publication_ID    Numeric NOT NULL, ;
    PRIMARY KEY STR(Personality_ID)+STR(Publication_ID) TAG PK;
);
CREATE TABLE User ;
( ;
    ID                Numeric NOT NULL, ;
    Nickname          Varchar(32) NOT NULL, ;
    Email             Varchar(64) NOT NULL, ;
    Registration_Date Date NOT NULL, ;
    PRIMARY KEY ID TAG PK, ;

```

Рисунок 3.7

```

        UNIQUE Nickname TAG AK1, ;
        UNIQUE Email TAG AK2 ;
    );
CREATE TABLE User_Date_Info ;
( ;
    Type_ID          Numeric NOT NULL, ;
    User_ID          Numeric NOT NULL, ;
    Value            Date NOT NULL, ;
    PRIMARY KEY STR(Type_ID)+STR(User_ID) TAG PK ;
);
CREATE TABLE User_Date_Info_Type ;
( ;
    ID              Numeric NOT NULL, ;
    Type            Varchar(20) NOT NULL, ;
    PRIMARY KEY ID TAG PK ;
);
CREATE TABLE User_String_Info ;
( ;
    Type_ID          Numeric NOT NULL, ;
    User_ID          Numeric NOT NULL, ;
    Value            Varchar(32) NOT NULL, ;
    PRIMARY KEY STR(Type_ID)+STR(User_ID) TAG PK ;
);
CREATE TABLE User_String_Info_Type ;
( ;
    ID              Numeric NOT NULL, ;
    Type            Varchar(20) NOT NULL, ;
    PRIMARY KEY ID TAG PK ;
);
CREATE TABLE Want_To_Read ;
( ;
    User_ID          Numeric NOT NULL, ;
    Book_ID          Numeric NOT NULL, ;
    PRIMARY KEY STR(User_ID)+STR(Book_ID) TAG PK ;
);
CREATE TABLE Work ;
( ;
    ID              Numeric NOT NULL, ;
    Title            Varchar(128) NOT NULL, ;

```

Рисунок 3.8

```

        Origin_Language      Varchar(20) NOT NULL, ;
        PRIMARY KEY ID TAG PK ;
);
CREATE TABLE Work_Belongs_To_Genre ;
( ;
        Genre_ID              Numeric NOT NULL, ;
        Work_ID                Numeric NOT NULL, ;
        PRIMARY KEY STR(Genre_ID)+STR(Work_ID) TAG PK ;
);
CREATE TABLE Work_Contains_Character ;
( ;
        Character_ID          Numeric NOT NULL, ;
        Work_ID                Numeric NOT NULL, ;
        PRIMARY KEY STR(Character_ID)+STR(Work_ID) TAG PK ;
);
ALTER TABLE Base_Object ;
        ADD FOREIGN KEY Type_ID TAG Type_ID REFERENCES Base_Object_Type ;
ALTER TABLE Series ;
        ADD FOREIGN KEY ID TAG Base_ID REFERENCES Base_Object ;
ALTER TABLE Publisher ;
        ADD FOREIGN KEY ID TAG Base_ID REFERENCES Base_Object ;
ALTER TABLE Personality ;
        ADD FOREIGN KEY ID TAG Base_ID REFERENCES Base_Object ;
ALTER TABLE Book ;
        ADD FOREIGN KEY ID TAG Base_ID REFERENCES Base_Object ;
        ADD FOREIGN KEY Type_ID TAG Type_ID REFERENCES Book_Type ;
ALTER TABLE Character ;
        ADD FOREIGN KEY ID TAG Base_ID REFERENCES Base_Object ;
ALTER TABLE Publisher_Publishes_Series ;
        ADD FOREIGN KEY Series_ID TAG Series_ID REFERENCES Series ;
        ADD FOREIGN KEY Publisher_ID TAG Publ_ID REFERENCES Publisher ;
ALTER TABLE Publication ;
        ADD FOREIGN KEY ID TAG Book_ID REFERENCES Book ;
        ADD FOREIGN KEY Series_ID TAG Series_ID REFERENCES Series ;
ALTER TABLE Work ;
        ADD FOREIGN KEY ID TAG Book_ID REFERENCES Book ;
ALTER TABLE Publication_Contains_Work ;
        ADD FOREIGN KEY Publication_ID TAG Publ_ID REFERENCES Publication ;
        ADD FOREIGN KEY Work_ID TAG Work_ID REFERENCES Work ;

```

Рисунок 3.9

```

ALTER TABLE Work_Contains_Character ;
    ADD FOREIGN KEY Character_ID TAG Char_ID REFERENCES Character ;
    ADD FOREIGN KEY Work_ID TAG Work_ID REFERENCES Work ;
ALTER TABLE Alternative_Title ;
    ADD FOREIGN KEY Work_ID TAG Work_ID REFERENCES Work ;
ALTER TABLE Work_Belongs_To_Genre ;
    ADD FOREIGN KEY Genre_ID TAG Genre_ID REFERENCES Genre ;
    ADD FOREIGN KEY Work_ID TAG Work_ID REFERENCES Work ;
ALTER TABLE Authorship ;
    ADD FOREIGN KEY Personality_ID TAG Pers_ID REFERENCES Personality ;
    ADD FOREIGN KEY Work_ID TAG Work_ID REFERENCES Work ;
ALTER TABLE Editorship ;
    ADD FOREIGN KEY Personality_ID TAG Pers_ID REFERENCES Personality ;
    ADD FOREIGN KEY Publication_ID TAG Publ_ID REFERENCES Publication ;
ALTER TABLE Compilership ;
    ADD FOREIGN KEY Personality_ID TAG Pers_ID REFERENCES Personality ;
    ADD FOREIGN KEY Publication_ID TAG Publ_ID REFERENCES Publication ;
ALTER TABLE Translatorship ;
    ADD FOREIGN KEY Personality_ID TAG Pers_ID REFERENCES Personality ;
    ADD FOREIGN KEY Publication_ID TAG Publ_ID REFERENCES Publication ;
ALTER TABLE Illustratorship ;
    ADD FOREIGN KEY Personality_ID TAG Pers_ID REFERENCES Personality ;
    ADD FOREIGN KEY Publication_ID TAG Publ_ID REFERENCES Publication ;
ALTER TABLE Single_Names ;
    ADD FOREIGN KEY Personality_ID TAG Pers_ID REFERENCES Personality ;
    ADD FOREIGN KEY Type_ID TAG Type_ID REFERENCES Single_Name_Types ;
ALTER TABLE Multiple_Names ;
    ADD FOREIGN KEY Personality_ID TAG Pers_ID REFERENCES Personality ;
    ADD FOREIGN KEY Type_ID TAG Type_ID REFERENCES Multiple_Name_Types ;
ALTER TABLE Review ;
    ADD FOREIGN KEY User_ID TAG User_ID REFERENCES User ;
    ADD FOREIGN KEY Book_ID TAG Book_ID REFERENCES Book ;
ALTER TABLE Tag ;
    ADD FOREIGN KEY User_ID TAG User_ID REFERENCES User ;
    ADD FOREIGN KEY Book_ID TAG Book_ID REFERENCES Book ;
ALTER TABLE Mark ;
    ADD FOREIGN KEY User_ID TAG User_ID REFERENCES User ;
    ADD FOREIGN KEY Book_ID TAG Book_ID REFERENCES Book ;
ALTER TABLE Read ;

```

Рисунок 3.10

```

        ADD FOREIGN KEY User_ID TAG User_ID REFERENCES User ;
        ADD FOREIGN KEY Book_ID TAG Book_ID REFERENCES Book ;
ALTER TABLE Want_To_Read ;
        ADD FOREIGN KEY User_ID TAG User_ID REFERENCES User ;
        ADD FOREIGN KEY Book_ID TAG Book_ID REFERENCES Book ;
ALTER TABLE Have_Paper_Book ;
        ADD FOREIGN KEY User_ID TAG User_ID REFERENCES User ;
        ADD FOREIGN KEY Book_ID TAG Book_ID REFERENCES Book ;
ALTER TABLE Have_Digital_Book ;
        ADD FOREIGN KEY User_ID TAG User_ID REFERENCES User ;
        ADD FOREIGN KEY Book_ID TAG Book_ID REFERENCES Book ;
ALTER TABLE User_String_Info ;
        ADD FOREIGN KEY User_ID TAG User_ID REFERENCES User ;
        ADD FOREIGN KEY Type_ID TAG Type_ID REFERENCES User_String_Info_Type ;
ALTER TABLE User_Date_Info ;
        ADD FOREIGN KEY User_ID TAG User_ID REFERENCES User ;
        ADD FOREIGN KEY Type_ID TAG Type_ID REFERENCES User_Date_Info_Type ;

```

Рисунок 3.11

Данная программа создает таблицы, добавляет в них нужные поля, создает первичные, альтернативные и внешние ключи. К сожалению FoxPro не поддерживает создание триггеров поддержки ссылочной целостности (Referential Integrity) программным путем, поэтому для их настройки необходимо открыть базу, в окне Database Designer выбрать одну из созданных связей и дважды кликнуть по ней левой кнопкой мыши. Откроется окно EditRelation.

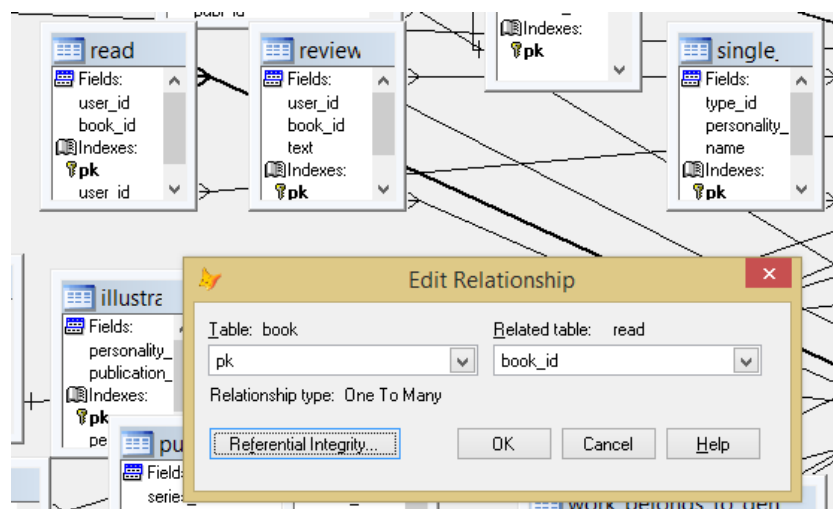


Рисунок 3.12

Необходимо нажать кнопку ReferentialIntegrity. Откроется окно, в котором можно будет отредактировать настройки ссылочной целостности для всех связей.

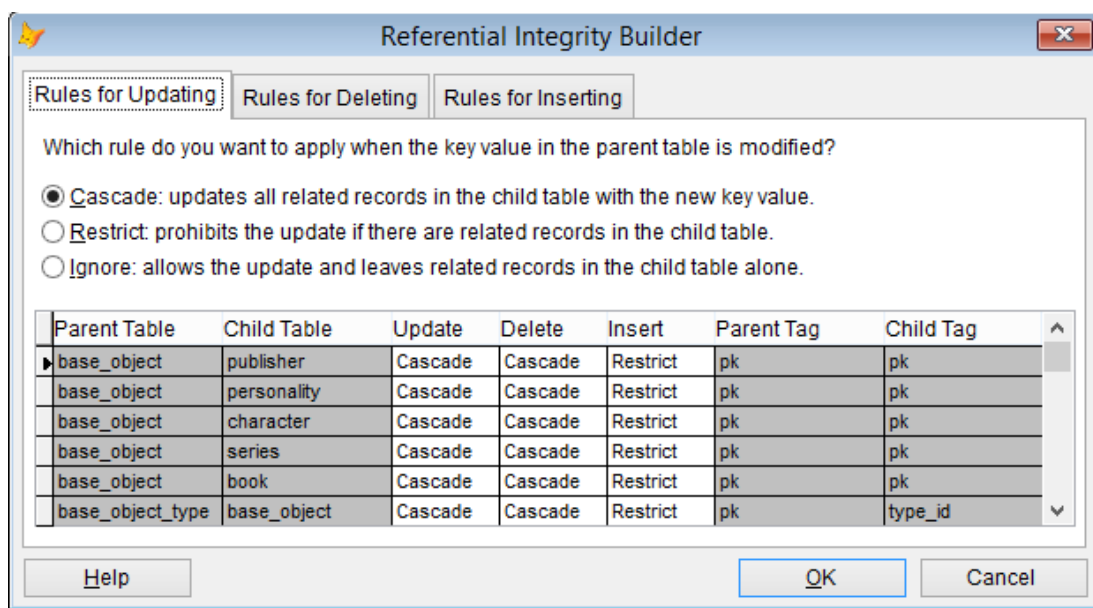


Рисунок 3.13

Для заполнения таблиц тестовыми данными в данной работе используется программа на FoxPro. Ее текст приведен на рисунках.

```

CD d:\[projects]\[db]
SET PROCEDURE TO udf.prg
CLOSE TABLES ALL
INSERT INTO Base_Object_Type VALUES ( 0, 'Book' )
INSERT INTO Base_Object_Type VALUES ( 1, 'Personality' )
INSERT INTO Base_Object_Type VALUES ( 2, 'Series' )
INSERT INTO Base_Object_Type VALUES ( 3, 'Publisher' )
INSERT INTO Base_Object_Type VALUES ( 4, 'Character' )
INSERT INTO Book_Type VALUES ( 0, 'Work' )
INSERT INTO Book_Type VALUES ( 1, 'Publication' )
INSERT INTO Single_Name_Types VALUES ( 0, 'First Name' )
INSERT INTO Single_Name_Types VALUES ( 1, 'Last Name' )
INSERT INTO Single_Name_Types VALUES ( 2, 'Patronymic' )
INSERT INTO Multiple_Name_Types VALUES ( 0, 'Second Name' )
INSERT INTO User_Date_Info_Type VALUES ( 0, 'Birthdate' )
INSERT INTO User_String_Info_Type VALUES ( 0, 'Name' )
INSERT INTO User_String_Info_Type VALUES ( 1, 'City' )
INSERT INTO User_String_Info_Type VALUES ( 2, 'Profession' )

```

Рисунок 3.14

```

INSERT INTO Genre VALUES (0, 'Fantastique')
INSERT INTO Genre VALUES (1, 'Detective')
INSERT INTO Genre VALUES (2, 'Humor')
INSERT INTO Genre VALUES (3, 'Mystery')
INSERT INTO Base_Object VALUES (0, 2, 'Default series')
INSERT INTO Series VALUES (0, 'Title of default series')
FOR I = 1 TO 1000000
  obj_type = RANDOM(0, 4)
  DO CASE
    CASE obj_type = 0
      INSERT INTO Base_Object VALUES (I, obj_type, 'Book with ID '+STR(I))
      book_type = RANDOM(0,1)
      INSERT INTO Book VALUES (I, book_type)
      DO CASE
        CASE book_type = 0
          INSERT INTO Work VALUES (I, 'Title of Work with ID '+STR(I),
'Origin Language of Work')
          alt_title_count = RANDOM(0,2)
          IF alt_title_count != 0
            FOR J=1 TO alt_title_count
              INSERT INTO Alternative_Title VALUES (I,
'Alternative title #' +STR(J))
            ENDFOR
          ENDIF
        CASE book_type = 1
          CLOSE TABLES ALL
          USE series
          COUNT TO series_rec_count
          rand_rec = RANDOM(1, series_rec_count)
          GO rand_rec IN series
          s_id = series.id
          USE IN series
          INSERT INTO Publication VALUES (I, s_id,
STR(I)+'1234567891011', 0, 'Cover type', 0, 0, 0, 0, 'Publishing language', 'Title of
Publication with id '+STR(I))
          ENDCASE
        CASE obj_type = 1
          INSERT INTO Base_Object VALUES (I, obj_type, 'Personality with ID
'+STR(I))

```

Рисунок 3.15

```

        INSERT INTO Personality VALUES (I)
    CASE obj_type = 2
        INSERT INTO Base_Object VALUES (I, obj_type, 'Series with ID '+STR(I))
        INSERT INTO Series VALUES (I, 'Title of Series with ID '+STR(I))
    CASE obj_type = 3
        INSERT INTO Base_Object VALUES (I, obj_type, 'Publisher with ID '+STR(I))
        INSERT INTO Publisher VALUES (I, 'Title of Publisher with ID '+STR(I),
'www.publisher'+STR(I)+' .com')
    CASE obj_type = 4
        INSERT INTO Base_Object VALUES (I, obj_type, 'Character with ID '+STR(I))
        INSERT INTO Character VALUES (I, 'Name of Character with ID '+STR(I))
    ENDCASE
ENDFOR
FOR I = 0 TO 10000
    INSERT INTO user VALUES (I, 'User'+STR(I), 'User'+STR(I)+'@somemail.com', RAND_DATE())
    b_info_name = RANDOM(0,1)
    IF b_info_name = 1
        INSERT INTO user_string_info VALUES (0, I, 'Name of user'+STR(I))
    ENDIF
    b_info_city = RANDOM(0,1)
    IF b_info_city = 1
        INSERT INTO user_string_info VALUES (1, I, 'City of user'+STR(I))
    ENDIF
    b_info_profession = RANDOM(0,1)
    IF b_info_profession = 1
        INSERT INTO user_string_info VALUES (2, I, 'Profession of user'+STR(I))
    ENDIF
    b_info_date = RANDOM(0,1)
    IF b_info_date = 1
        INSERT INTO user_date_info VALUES (0, I, RAND_DATE())
    ENDIF
ENDFOR
CLOSE TABLES ALL
USE publisher IN 1

```

Рисунок3.16

```

COUNT TO pub_rec_count
USE series IN 2
SELECT 2
SCAN
    publisher_count = RANDOM(1,2)
    s_id = id
    SELECT 1
    FOR I = 1 TO publisher_count
        rand_rec = RANDOM(1, pub_rec_count)
        GO rand_rec IN publisher
        p_id = publisher.id
        INSERT INTO Publisher_Publishes_Series (series_id, publisher_id) VALUES (s_id,
p_id)
    ENDFOR
    SELECT 2
ENDSCAN
CLOSE TABLES ALL
USE work IN 1
COUNT TO work_rec_count
USE character IN 2
SELECT 2
SCAN
    work_count = RANDOM(1,5)
    c_id = id
    SELECT 1
    FOR I = 1 TO work_count
        rand_rec = RANDOM(1, work_rec_count)
        GO rand_rec IN work
        w_id = work.id
        INSERT INTO Work_Contains_Character (character_id, work_id) VALUES (c_id, w_id)

    ENDFOR
    SELECT 2
ENDSCAN
CLOSE TABLES ALL
USE work IN 1
COUNT TO work_rec_count
USE publication IN 2
SELECT 2

```

Рисунок 3.17

```

COUNT TO pub_rec_count
USE series IN 2
SELECT 2
SCAN
    publisher_count = RANDOM(1,2)
    s_id = id
    SELECT 1
    FOR I = 1 TO publisher_count
        rand_rec = RANDOM(1, pub_rec_count)
        GO rand_rec IN publisher
        p_id = publisher.id
        INSERT INTO Publisher_Publishes_Series (series_id, publisher_id) VALUES (s_id,
p_id)
    ENDFOR
    SELECT 2
ENDSCAN
CLOSE TABLES ALL
USE work IN 1
COUNT TO work_rec_count
USE character IN 2
SELECT 2
SCAN
    work_count = RANDOM(1,5)
    c_id = id
    SELECT 1
    FOR I = 1 TO work_count
        rand_rec = RANDOM(1, work_rec_count)
        GO rand_rec IN work
        w_id = work.id
        INSERT INTO Work_Contains_Character (character_id, work_id) VALUES (c_id, w_id)

    ENDFOR
    SELECT 2
ENDSCAN
CLOSE TABLES ALL
USE work IN 1
COUNT TO work_rec_count
USE publication IN 2
SELECT 2

```

Рисунок 3.18

```

SCAN
    work_count = RANDOM(1,5)
    p_id = id
    SELECT 1
    FOR I = 1 TO work_count
        rand_rec = RANDOM(1, work_rec_count)
        GO rand_rec IN work
        w_id = work.id
        INSERT INTO Publication_Contains_Work (publication_id, work_id) VALUES (p_id,
w_id)
    ENDFOR
    SELECT 2
ENDSCAN
CLOSE TABLES ALL
USE genre IN 1
COUNT TO genre_rec_count
USE work IN 2
SELECT 2
SCAN
    genre_count = RANDOM(1,2)
    w_id = id
    SELECT 1
    FOR I = 1 TO genre_count
        rand_rec = RANDOM(1, genre_rec_count)
        GO rand_rec IN genre
        g_id = genre.id
        INSERT INTO Work_Belongs_To_Genre (work_id, genre_id) VALUES (w_id, g_id)

    ENDFOR
    SELECT 2
ENDSCAN
CLOSE TABLES ALL
USE book IN 1
COUNT TO book_rec_count
USE user IN 2
SELECT 2
SCAN
    book_count = RANDOM(1,30)
    u_id = id

```

Рисунок 3.19

```

SELECT 1
FOR I = 1 TO book_count
    rand_rec = RANDOM(1, book_rec_count)
    GO rand_rec IN book
    b_id = book.id
    is_read = RANDOM(0,1)
    IF is_read = 1
        INSERT INTO read (user_id, book_id) VALUES (u_id, b_id)
    ELSE
        INSERT INTO want_to_read (user_id, book_id) VALUES (u_id, b_id)
    ENDIF
ENDFOR
paperbook_count = RANDOM(1,30)
FOR I = 1 TO paperbook_count
    rand_rec = RANDOM(1, book_rec_count)
    GO rand_rec IN book
    b_id = book.id
    INSERT INTO have_paper_book (user_id, book_id) VALUES (u_id, b_id)
ENDFOR
digitalbook_count = RANDOM(1,30)
FOR I = 1 TO digitalbook_count
    rand_rec = RANDOM(1, book_rec_count)
    GO rand_rec IN book
    b_id = book.id
    INSERT INTO have_digital_book (user_id, book_id) VALUES (u_id, b_id)
ENDFOR
review_count = RANDOM(1,30)
FOR I = 1 TO review_count
    rand_rec = RANDOM(1, book_rec_count)
    GO rand_rec IN book
    b_id = book.id
    INSERT INTO review (user_id, book_id, text) VALUES (u_id, b_id, 'Текстотзыва')
ENDFOR
tag_count = RANDOM(1,30)
FOR I = 1 TO tag_count
    rand_rec = RANDOM(1, book_rec_count)
    GO rand_rec IN book
    b_id = book.id

```

Рисунок 3.20

```

        INSERT INTO tag (user_id, book_id, tag) VALUES (u_id, b_id, 'tag'+STR(1))
    ENDFOR
    mark_count = RANDOM(1,30)
    FOR I = 1 TO mark_count
        rand_rec = RANDOM(1, book_rec_count)
        GO rand_rec IN book
        b_id = book.id
        INSERT INTO mark (user_id, book_id, mark) VALUES (u_id, b_id, RANDOM(1,10))
    ENDFOR
    SELECT 2
ENDSCAN
CLOSE TABLES ALL
USE publication IN 1
COUNT TO pub_rec_count
USE personality IN 2
SELECT 2
SCAN
    per_id = id
    SELECT 1
    editor_count = RANDOM(0,10)
    IF editor_count != 0
        FOR I = 1 TO editor_count
            rand_rec = RANDOM(1, pub_rec_count)
            GO rand_rec IN publication
            pub_id = publication.id
            INSERT INTO editorship (personality_id, publication_id) VALUES (per_id,
pub_id)
        ENDFOR
    ENDIF
    compiler_count = RANDOM(0,10)
    IF compiler_count != 0
        FOR I = 1 TO compiler_count
            rand_rec = RANDOM(1, pub_rec_count)
            GO rand_rec IN publication
            pub_id = publication.id
            INSERT INTO compilership (personality_id, publication_id) VALUES (per_id,
pub_id)
        ENDFOR

```

Рисунок 3.21


```

ENDIF
translator_count = RANDOM(0,10)
IF translator_count != 0
    FOR I = 1 TO translator_count
        rand_rec = RANDOM(1, pub_rec_count)
        GO rand_rec IN publication
        pub_id = publication.id
        INSERT INTO translatorship (personality_id, publication_id) VALUES
(per_id, pub_id)
    ENDFOR
ENDIF
illustrator_count = RANDOM(0,10)
IF illustrator_count != 0
    FOR I = 1 TO illustrator_count
        rand_rec = RANDOM(1, pub_rec_count)
        GO rand_rec IN publication
        pub_id = publication.id
        INSERT INTO illustratorship (personality_id, publication_id) VALUES
(per_id, pub_id)
    ENDFOR
ENDIF
USE work IN 1
COUNT TO work_rec_count
author_count = RANDOM(0,10)
IF author_count != 0
    FOR I = 1 TO author_count
        rand_rec = RANDOM(1, work_rec_count)
        GO rand_rec IN work
        w_id = work.id
        INSERT INTO authorship (personality_id, work_id) VALUES (per_id, w_id)
    ENDFOR
ENDIF
USE publication IN 1
SELECT 2
ENDSCAN
CLOSE TABLES ALL
CREATE TABLE names ;
( ;

```

Рисунок 3.22

```

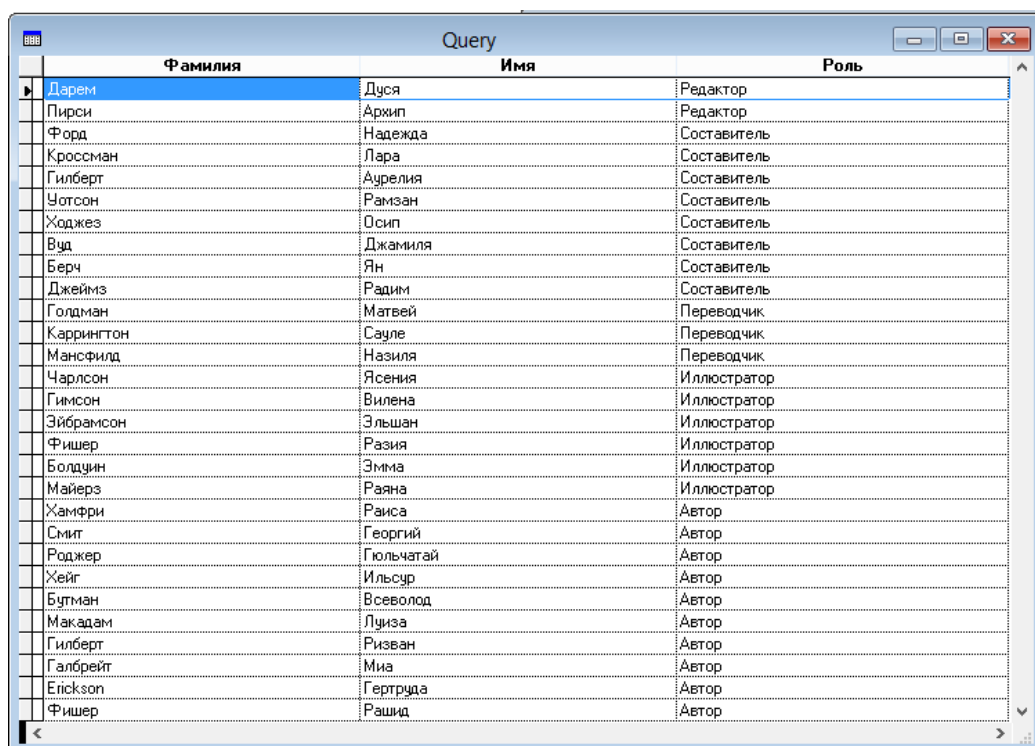
        name                VARCHAR(32) NOT NULL, ;
        PRIMARY KEY name TAG PK ;
    )
APPEND FROM names.txt DELIMITED
USE IN names
CREATE TABLE families ;
( ;
    family_name            VARCHAR(32) NOT NULL, ;
    PRIMARY KEY family_name TAG PK ;
)
APPEND FROM families.txt DELIMITED
USE IN families
CLOSE TABLES ALL
SELECT 1
USE names IN 1
COUNT TO names_rec_count
SELECT 2
USE families IN 2
COUNT TO families_rec_count
USE personality IN 3
SELECT 3
SCAN
    p_id = id
    SELECT 1
    rand_rec = RANDOM(1, names_rec_count)
    GO rand_rec IN names
    pname = names.name
    INSERT INTO single_names VALUES (0, p_id, pname)
    SELECT 2
    rand_rec = RANDOM(1, families_rec_count)
    GO rand_rec IN families
    fname = families.family_name
    INSERT INTO single_names VALUES (1, p_id, fname)
    SELECT 3
ENDSCAN
CLOSE TABLES ALL
REMOVE TABLE names DELETE
REMOVE TABLE families DELETE

```

Рисунок 3.23

3.3 Разработка запроса

Создать запрос в VisualFoxPro можно несколькими способами: с помощью QueryDesigner'a или программным путем. Так как QueryDesigner не поддерживает возможность объединений (Union), в данной работе была использована программа на FoxPro, выполняющая необходимый запрос. Тексты программ представлены в подразделе 2.6 на рисунках 2.19-2.20, 2.22-2.23, 2.25-2.26. Результаты запроса на заполненной базе данных представлены на рисунке 3.24.



Фамилия	Имя	Роль
Дарем	Дуся	Редактор
Пирси	Архип	Редактор
Форд	Надежда	Составитель
Кроссман	Лара	Составитель
Гилберт	Аурелия	Составитель
Уотсон	Рамзан	Составитель
Ходжез	Осип	Составитель
Вуд	Джамия	Составитель
Берч	Ян	Составитель
Джеймс	Радин	Составитель
Голдман	Матвей	Переводчик
Каррингтон	Сауле	Переводчик
Мансфилд	Назиля	Переводчик
Чарлсон	Ясения	Иллюстратор
Гимсон	Вилена	Иллюстратор
Зибрамсон	Эльшан	Иллюстратор
Фишер	Разия	Иллюстратор
Болдуин	Эмма	Иллюстратор
Майерз	Раяна	Иллюстратор
Хамфри	Раиса	Автор
Смит	Георгий	Автор
Роджер	Гюльчатай	Автор
Хейг	Ильсур	Автор
Бутман	Всеволод	Автор
Макадам	Луиза	Автор
Гилберт	Ризван	Автор
Галбрейт	Миа	Автор
Erickson	Гертруда	Автор
Фишер	Рашид	Автор

Рисунок 3.24

Заключение

Таким образом, в результате выполнения КП получен оптимизированный проект базы данных, а также построены обычные и оптимизированные SQL-запросы. В результате работы была спроектирована база данных для книжной социальной сети. Также произошло ознакомление с основными принципами построения, использования и оптимизации реляционных баз данных.

Была спроектирована система, предоставляющая пользователю возможность получить информацию о книгах, авторах, издательствах, выпускающих составах книг и персонажах литературных произведений, а также возможность составлять списки прочитанного, отмеченного прочтению, ставить оценки и присваивать метки.

Данная задача была решена с помощью анализа предметной области, ее формализации с помощью функциональных зависимостей. Были проведены этапы минимизации системы функциональных зависимостей, описывающих предметную область, и на основании полученной редуцированной системы спроектирована требуемая модель базы данных.

Список использованных источников

1. Кузнецов, С. Введение в реляционные базы данных [Электронный ресурс]. / С. Кузнецов. - 2-е изд., исправ. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 248 с. : ил. - Режим доступа: <http://biblioclub.ru/index.php?page=book&id=429088>
2. Гудов, А.М. Базы данных и системы управления базами данных. Программирование на языке PL/SQL : учебное пособие [Электронный ресурс]/ А.М. Гудов, С.Ю. Завозкин, Т.С. Рейн ; Министерство образования и науки Российской Федерации, Государственное образовательное учреждение высшего профессионального образования «Кемеровский государственный университет». - Кемерово : Кемеровский государственный университет, 2010. - 134 с. : ил.,табл., схем. ; - Режим доступа:<http://biblioclub.ru/index.php?page=book&id=232497>.
3. Гущин, А.Н. Базы данных [Электронный ресурс] : учебник / А.Н. Гущин. - М. : Директ-Медиа, 2014. - 266 с. // Режим доступа - <http://biblioclub.ru/index.php?page=book&id=222149>
4. Щелоков, С.А. Базы данных : учебное пособие / С.А. Щелоков ; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Оренбургский государственный университет», Кафедра программного обеспечения вычислительной техники и автоматизированных систем. - Оренбург : Оренбургский государственный университет, 2014. - 298 с. : ил. - Библиогр. в кн. ; [Электронный ресурс]. - Режим доступа: <http://biblioclub.ru/index.php?page=book&id=260752> .

Приложение А

Схемы базы данных

На рисунке А.1 приведена структурная схема базы данных на логическом уровне.

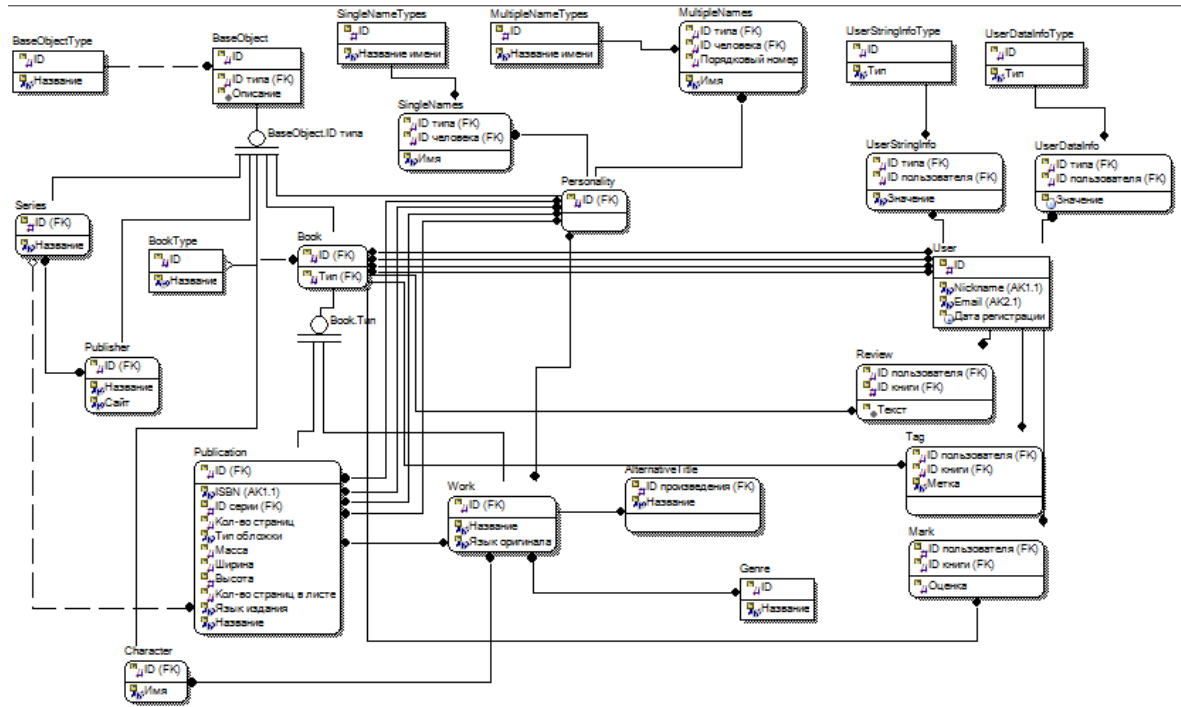


Рисунок А.1

На рисунке А.2 приведена структурная схема базы данных на физическом уровне.

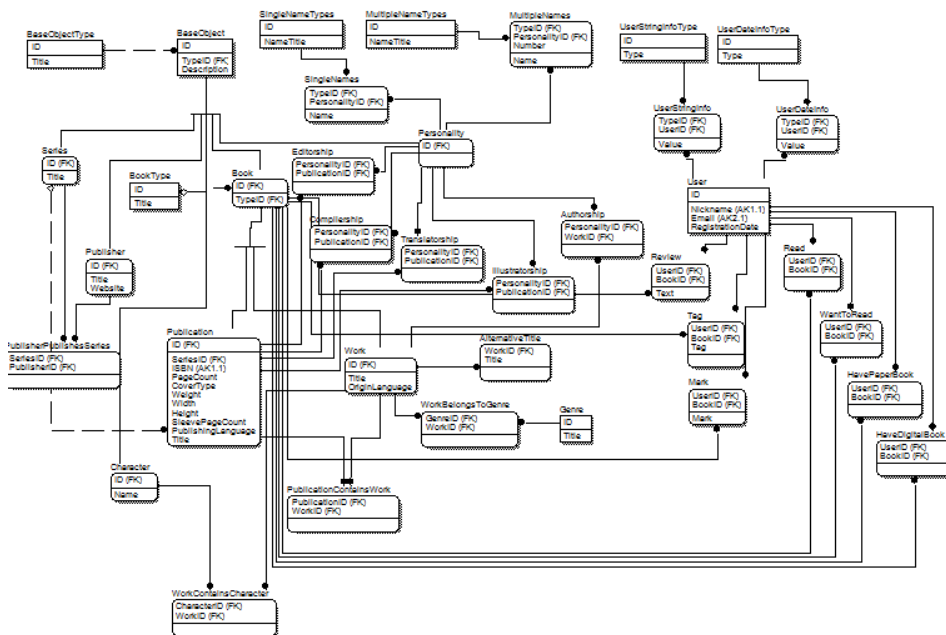


Рисунок А.2

Приложение Б

Требования к оформлению курсовой работы

Курсовая работа должна быть оформлена в соответствии со стандартом университета СТУ 04.02.030 – 2017 "Курсовые работы (проекты). Выпускные квалификационные работы. Общие требования к структуре и оформлению".

Структура КР (КП) включает:

- титульный лист;
- задание;
- реферат;
- содержание;
- обозначения и сокращения (при необходимости);
- введение;
- основная часть, в том числе исследовательский раздел;
- заключение;
- список использованных источников;
- приложения (при необходимости).

После приложений могут быть помещены самостоятельные конструкторские, технологические, программные и другие проектные материалы, выполненные в ходе проектирования согласно заданию. В структуру основной части работы может входить экспериментальная часть с приложением электронного носителя с результатами работы.

Объем основной части должен быть не менее 40 листов (одна страница на одной стороне листа).

Титульный лист должен быть первым листом курсовой работы, но номер страницы на нем не ставится. На всех остальных листах, начиная со второго ставятся номера страниц. Вторым листом должен быть лист задания на курсовую работу.

Формы титульного листа и листов задания приведены ниже.

Форма титульного листа на курсовую работу (проект)
Минобрнауки России
Юго- Западный государственный университет

Кафедра _____

КУРСОВАЯ РАБОТА (ПРОЕКТ)

подисциплине« _____ »
(наименование дисциплины)

на тему« _____ »
_____»

Направление подготовки(специальность) _____
(код, наименование)

Автор работы(проекта) _____
(инициалы, фамилия) (подпись, дата)

Группа _____

Руководитель работы(проекта) _____
(инициалы, фамилия) (подпись, дата)

Работа (проект) защищена _____
(дата)

Оценка _____

Члены комиссии _____
(подпись, дата) (инициалы, фамилия)

_____ (подпись, дата) (инициалы, фамилия)

_____ (подпись, дата) (инициалы, фамилия)

Курск, 20 ____ г.

Форма задания на курсовую работу (проект)

Минобрнауки России

Юго - Западный государственный университет

Кафедра _____

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ (ПРОЕКТ)

Студент _____ шифр _____ группа _____
(фамилия, инициалы)

1. Тема _____

2. Срок представления работы (проекта) к защите «_____» _____ 20__ г.

3. Исходные данные (для проектирования, для научного исследования):

4. Содержание пояснительной записки курсовой работы (проекта):

4.1. _____

4.2. _____

4.3. _____

4.4. _____

4.5. _____

5. Перечень графического материала:

Руководитель работы(проекта) _____
(подпись, дата) (инициалы, фамилия)

Задание принял к исполнению _____
(подпись, дата) (инициалы, фамилия)

