

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Оренбургский государственный университет»

Кафедра прикладной информатики в экономике и управлении

Н.Ф. Панова

БАЗЫ ДАННЫХ

Методические указания

Рекомендовано к изданию редакционно-издательским советом федерального государственного бюджетного образовательного учреждения высшего образования «Оренбургский государственный университет» для обучающихся по образовательной программе высшего образования по направлению подготовки 09.03.03 Прикладная информатика

Оренбург
2018

УДК 004.6(076.5)
ББК 32.972.1я7
П 16

Рецензент - кандидат технических наук М.А. Токарева

Панова, Н.Ф.

П 16 Базы данных: методические указания/Н.Ф. Панова; Оренбургский гос. ун-т.- Оренбург : ОГУ, 2018. - 38 с.

Методические указания предназначены для выполнения курсового проекта по дисциплине «Базы данных». Методические указания содержат теоретический материал по проектированию и реализации реляционных баз данных, описание этапов выполнения курсового проекта, а также пример разработки курсового проекта.

Предназначены для обучающихся по образовательной программе высшего образования по направлению подготовки 09.03.03 Прикладная информатика очной и заочной форм обучения.

УДК 004.6(076.5)
ББК 32.972.1я7

© Панова Н.Ф., 2018
© ОГУ, 2018

Содержание

Введение	4
1 Теоретические сведения	5
2 Пример разработки курсового проекта.....	8
2.1 Разработка инфологической модели предметной области	8
2.2 Выбор СУБД	17
2.3 Дatalogическое проектирование БД	17
2.4 Проектирование внутреннего уровня БД	21
3 Оформление курсового проекта	35
Список использованных источников	37
Приложение А. ER-диаграмма предметной области.....	38

Введение

Дисциплина «Базы данных» относится к базовой части блока 1 «Дисциплины (модули)», предусмотренных Государственным стандартом подготовки бакалавров по направлению 09.03.03 Прикладная информатика.

Целью освоения дисциплины является овладение современными методами и средствами технологии баз данных, как одной из основных информационных технологий.

Задачами курса являются освещение теоретических и организационно-методических вопросов построения и функционирования систем, основанных на концепции баз данных, в том числе различных методологий моделирования и проектирования баз данных; знакомство с возможностями средств автоматизации проектирования баз данных; знакомство с возможностями современных высокоуровневых языков и средств создания приложений; приобретение практических навыков в проектировании, ведении и использовании баз данных в среде выбранных целевых систем управления базами данных (СУБД); знакомство с основами обеспечения безопасности и целостности баз данных, а также с направлениями и перспективами развития баз данных.

В рамках курса студенты должны получить навыки использования базовых понятий и определений, основ проектирования баз данных, ориентирования в тенденциях развития современных средств проектирования баз данных и уровнях представления данных, знания моделей данных и методов обработки.

Методические указания предназначены для курсового проектирования по дисциплине «Базы данных» для студентов вторых курсов направления 09.03.03 Прикладная информатика и содержат теоретический материал по теории и практике проектирования реляционных баз данных и основам языка SQL. Указания содержат пример разработки курсового проекта, а также описание требований к содержанию и оформлению пояснительной записки.

1 Теоретические сведения

Существует два основных подхода к проектированию БД: нисходящий и восходящий. Восходящий подход начинается с самого нижнего уровня - с уровня определения атрибутов. Эти атрибуты на основе анализа существующих между ними связей группируются в несколько отношений. Полученные отношения далее нормализуются для того, чтобы исключить избыточность (дублирование) данных.

Восходящий подход лучше всего подходит для проектирования простых баз данных с небольшим числом атрибутов. Базы данных, которые насчитывают сотни и тысячи атрибутов, проектируются на основе нисходящего подхода. В этом случае проектирование начинается с разработки высокоуровневых моделей данных, которые называются инфологическими. На рисунке 1 представлены основные этапы проектирования БД с применением нисходящего подхода.



Рисунок 1 - Этапы проектирования БД

При анализе предметной области необходимая для проектирования БД информация может быть собрана следующим образом:

- с помощью наблюдения за работой предприятия (организации);

- посредством изучения документов, которые используются для сбора и представления информации в данной предметной области (ПО);

- путём опроса сотрудников предприятия, особенно специалистов – будущих пользователей БД.

Собранная информация должна включать следующие компоненты: используемую входную и генерируемую выходную документацию, сведения о функциях, которые должны быть включены в создаваемое приложение с указанием их приоритетов. На основании этой информации будут составлены спецификации требований пользователей. Кроме того, должны быть сформулированы семантические ограничения на данные.

Инфологическое моделирование – это процесс создания инфологической концептуальной модели ПО. В рамках этой модели данные описываются в терминах, не связанных ни с каким физическим аспектом их хранения.

В качестве инфологической модели, как правило, используется модель «сущность-связь», которую часто называют ER-диаграммой. Эта модель имеет различные нотации, которые не меняют сути модели. При выполнении курсового проекта используется нотация Ричарда Баркера.

На этапе инфологического проектирования, кроме того, на основе анализа деятельности предприятия (организации), определяются группы пользователей, которые будут использовать проектируемую базу данных в рамках своих должностных обязанностей, конкретизируется перечень их функций. Например, в деканате вуза можно выделить такие группы пользователей как методисты, заместители декана и декан факультета. Методисты имеют права на ввод, обновление и удаление информации в базе данных по студентам. Заместители декана могут только просматривать информацию по обучающимся и формировать различные отчеты. Декан имеет доступ на просмотр данных не только по студентам, но и по сотрудникам.

Если выявлено несколько групп пользователей, то на этапе выбора СУБД анализируются только удаленные (промышленные) СУБД, способные обеспечить параллельный доступ к БД. Немаловажным фактором при выборе

СУБД является предполагаемый объем внешней памяти, необходимо для реализации проектируемой БД. Эта оценка проводится с учетом числа пользователей, интенсивности запросов к БД и количества объектов в каждом классе заданной предметной области.

В процессе выбора СУБД разработчик должен убедиться в том, что данная система действительно обладает достоинствами, способными принести реальную пользу. Сначала на основе требований пользователя определяются критерии оценки СУБД и формируется предварительный список доступных систем. Окончательный список, состоящий из двух-трех продуктов, создается после тщательного изучения функций и возможностей каждой системы. Для оценки можно использовать самые разные параметры. Обычно все параметры объединяются в группы, такие как физические (требования к памяти, требования к внешним устройствам, средства индексирования, простота реорганизации), наличие утилит, доступность (интерфейс для других систем, многопользовательский доступ, совместимость со стандартом SQL, защита данных) и другие параметры (цена, требования к операционной системе, наличие русифицированной версии и т.д.).

Даталогическое проектирование – это процесс преобразования инфологической модели данных в такую даталогическую модель (иерархическую, сетевую, реляционную, объектную), которую поддерживает выбранная СУБД. При этом модель по-прежнему не зависит от особенностей выбранной СУБД и не содержит информации о способе физического хранения данных. Результатом этапа даталогического проектирования реляционных баз данных является совокупность схем отношений. Каждое отношение необходимо проверить на наличие избыточности и при необходимости нормализовать. Достаточно преобразовать отношения к ЗНФ.

Этап проектирования внутреннего уровня – это процесс описания физической реализации БД. Суть этапа заключается в определении схемы хранения, т.е. физической структуры БД. Схема хранения зависит от той физической структуры, которую поддерживает выбранная СУБД. Физическая

структура БД, с одной стороны, должна адекватно отражать логическую структуру БД, а с другой стороны, должна обеспечивать эффективное размещение данных и быстрый доступ к ним. Результаты этого этапа документируются в форме схемы хранения на языке определения данных (DDL, Data Definition Language) выбранной СУБД. Принятые на этом этапе решения оказывают огромное влияние на производительность системы.

2 Пример разработки курсового проекта

В качестве примера предлагается рассмотреть такую предметную область, как музей. Основной функцией в данной организации является функция учета музейных экспонатов, в рамках которой ведутся карточки учета, где отражаются перемещения экспоната как внутри музея, так и вне (различные выставки).

2.1 Разработка инфологической модели предметной области

2.1.1 Анализ предметной области

База данных создается для информационной поддержки деятельности музея. База данных должна содержать сведения о музейных фондах и экспонатах, а также о сотрудниках музея.

В музее имеется несколько фондов: живопись, графика, икона, скульптура, декоративно-прикладное искусство (ДПИ), нумизматика, археология, рукописи и редкая книга и т.п. Необходимо реализовать ведение карточек музейных предметов - инвентарный номер, название, точная или примерная дата создания, автор работы, если он известен (ФИО, дата рождения, страна), выставки, в которых участвовал музейный предмет, ФИО сотрудника музея, отвечающего за данный экспонат. Один сотрудник музея может быть закреплен в качестве ответственного за несколько экспонатов, но за каждый экспонат отвечает только один сотрудник. Необходимо вести учет

движения музейных предметов, т.е. сведения о том, когда он был принят на хранение и откуда (подарен частным лицом или другим музеем, найден сотрудниками в научных экспедициях и т.д.), когда был отдан на реставрацию когда возвращен, передача на выставку, возвращение с выставки, списание и т.п.). Все эти сведения в карточку учета вносит сотрудник, отвечающий за сохранность экспоната. О выставках, на которых был представлен экспонат, необходимо знать адрес, где проводилась выставка, название выставки, дата начала работы, дата окончания работы).

Необходимо обеспечить выполнение следующих запросов к БД:

- определить местонахождение заданных музейных предметов на заданную дату;
- определить фонд, который содержит наибольшее количество предметов;
- сформировать карточку учета заданного предмета – инвентарный номер, название, дата создания, название фонда, ФИО хранителя, отвечающего за предмет.

Исходя из описания предметной области выделим следующие сущности:

- страна;
- передача;
- улица;
- населенный пункт;
- автор;
- источник;
- фонд;
- предмет;
- работник;
- предмет.

Сущность *Передача* является супертипом. Между сущностями *Предмет* и *Передача* имеется связь $M : N$, поскольку один и тот же музейный предмет может неоднократно передаваться как на выставку, так и на

реставрацию. Эта связь в рамках методики Ричарда Баркера разрешается с помощью добавления в модель сущности пересечения *Строка листа передачи*.

ER-диаграмма, отражающая связи между выявленными сущностями, представлена в приложении А.

Разработанную инфологическую модель необходимо документировать, т.е. описать с помощью таблиц, описывающих сущности, связи между сущностями, а также атрибуты сущностей.

Описания сущностей и особенностей их использования представлены в таблице 1.

Таблица 1 – Описание сущностей ER-диаграммы

Имя сущности	Пояснения	Особенности использования
Предмет	Характеризует конкретный музейный экспонат	Каждый экспонат должен быть приписан к одному фонду
Работник	Характеризует работника музея	Каждый работник может следить не более чем за 20-ю предметами
Выставка	Описывает выставки, на которые могут отдаваться предметы	Предмет не может передаваться на выставку, если она завершена
Реставрация	Характеризует степень реставрации предмета	Отсутствуют
Страна	Является справочником стран	Характеризует страну автора предмета, а так же страну проведения выставки
Населённый пункт	Характеризует региональный субъект страны, в которой проводится выставка	Определяется для конкретной страны
Улица	Является справочником улиц, по которым проводятся выставки	Общее для всех стран и населённых пунктов
Источник	Является справочником источников предметов музея	Источником могут быть сотрудники музея
Фонд	Является справочником фондов, к которым приписаны предметы	Отсутствуют
Автор	Кратко характеризует автора предмета	При смене автором фамилии или имени рекомендуется занести новые сведения
Строка листа передачи	Описывает конкретную запись передачи предмета на выставку или реставрацию	Предмет может быть передан, если он находится в музее. Можно передать либо на выставку, либо на реставрацию

Описание атрибутов сущности «Работник» представлено в таблице 2.

Таблица 2 – Описание атрибутов сущности «Работник»

Свойство	Пояснение	Физ.характеристики	Опциональность	Логические ограничения
Код	Уникальный идентификатор	Число, 6 знаков	д/б	>0
Фамилия	Фамилия работника	Символьный, 30 знаков	д/б	-
Имя	Имя работника	Символьный, 30 знаков	д/б	-
Отчество	Отчество	Символьный, 30 знаков	д/б	-
Дата рождения	Дата рождения сотрудника	Дата, гггг/мм/дд	м/б	<Текущей

Описание атрибутов сущности «Фонд» представлено в таблице 3.

Таблица 3 – Описание атрибутов сущности «Фонд»

Свойство	Пояснение	Физ.характеристик и	Опциональность	Логические ограничения
Код	Уникальный идентификатор	Число, 6 знаков	д/б	>0
Наименование	Название фонда	Символьный, 30 знаков	д/б	-

Описание атрибутов сущности «Источник» представлено в таблице 4.

Таблица 4 – Описание атрибутов сущности «Источник»

Свойство	Пояснение	Физ.характеристики	Опциональность	Логические ограничения
Код	Уникальный идентификатор	Число, 6 знаков	д/б	>0
Наименование	Название источника	Символьный, 30 знаков	д/б	-

Описание атрибутов сущности «Реставрация» представлено в таблице 5.

Таблица 5 – Описание атрибутов сущности «Реставрация»

Свойство	Пояснение	Физ.характеристики	Опциональность	Логические ограничения
Код	Уникальный идентификатор	Число, 6 знаков	д/б	>0

Продолжение таблицы 5

Свойство	Пояснение	Физ.характеристики	Опциональность	Логические ограничения
Степень	Степень реставрации	Символьный, 30 знаков	д/б	-

Описание атрибутов сущности «Страна» представлено в таблице 6.

Таблица 6 – Описание атрибутов сущности «Страна»

Свойство	Пояснение	Физ.характеристики	Опциональность	Логические ограничения
Код	Уникальный идентификатор	Число, 6 знаков	д/б	>0
Наименование	Название страны	Символьный, 30 знаков	д/б	-

Описание атрибутов сущности «Улица» представлено в таблице 7.

Таблица 7 – Описание атрибутов сущности «Улица»

Свойство	Пояснение	Физ.характеристики	Опциональность	Логические ограничения
Код	Уникальный идентификатор	Число, 6 знаков	д/б	>0
Наименование	Название улицы	Символьный, 30 знаков	д/б	-

Описание атрибутов сущности «Автор» представлено в таблице 8.

Таблица 8 – Описание атрибутов сущности «Автор»

Свойство	Пояснение	Физ.характеристики	Опциональность	Логические ограничения
Код	Уникальный идентификатор	Число, 6 знаков	д/б	>0
Фамилия	Фамилия автора	Символьный, 30 знаков	д/б	-
Имя	Имя автора	Символьный, 30 знаков	д/б	-

Описание атрибутов сущности «Предмет» представлено в таблице 9

Таблица 9 – Описание атрибутов сущности «Предмет»

Свойство	Пояснение	Физ.характеристики	Опциональность	Логические ограничения
Код	Уникальный идентификатор	Число, 6 знаков	д/б	>0
Инвентарный номер	Номер для хоз. учёта	Символьный, 30 знаков	д/б	-
Название	Название экспоната	Символьный, 30 знаков	д/б	-
Точная дата создания	Известная дата создания	Дата, гггг/мм/дд	м/б	<Текущей
Приблизительная дата создания	Приблизительная дата создания	Дата, гггг/мм/дд	м/б	<Текущей
Дата списания	Дата списания	Дата, гггг/мм/дд	м/б	-

Описание атрибутов сущности «Населённый пункт» представлено в таблице 10.

Таблица 10 – Описание атрибутов сущности «Населённый пункт»

Свойство	Пояснение	Физ.характеристики	Опциональность	Логические ограничения
Код	Уникальный идентификатор	Число, 6 знаков	д/б	>0
Наименование	Название населённого пункта	Символьный, 30 знаков	д/б	-

Описание атрибутов сущности «Выставка» представлено в таблице 11.

Таблица 11 – Описание атрибутов сущности «Выставка»

Свойство	Пояснение	Физ.характеристики	Опциональность	Логические ограничения
Код	Уникальный идентификатор	Число, 6 знаков	д/б	>0
Дата начала	Открытие выставки	Дата, гггг/мм/дд	д/б	-

Продолжение таблицы 11

Свойство	Пояснение	Физ.характеристики	Опциональность	Логические ограничения
Дата завершения	Закрытие выставки	Дата, гггг/мм/дд	д/б	-
Номер дома	Дом проведения выставки	Число, 3 знака	д/б	>0

Описание атрибутов сущности «Строка листа передачи» представлено в таблице 12.

Таблица 12 – Описание атрибутов сущности «Строка листа передачи»

Свойство	Пояснение	Физ.характеристики	Опциональность	Логические ограничения
Дата сдачи	Выдача	Дата, гггг/мм/дд	д/б	>= текущей
Дата получения	Выдача	Дата, гггг/мм/дд	д/б	>= даты сдачи

Описание связей между сущностями приведено в таблице 13.

Таблица 13 – Описание связей

Имя сущности субъекта	Имя связи	Имя сущности объекта	Обязательность	Мощность
Улица	Является адресом	Выставка	м/б	1:М
Выставка	Проводится	Улица	д/б	М:1
Населённый пункт	Является адресом	Выставка	м/б	1:М
Выставка	Проводится	Населённый пункт	д/б	М:1
Страна	Имеет в составе	Населённый пункт	м/б	1:М
Населённый пункт	Входит в состав	Страна	д/б	М:1
Страна	Является Родиной	Автор	м/б	1:М
Автор	Является гражданином	Страна	д/б	М:1
Автор	Является автором	Предмет	м/б	1:М
Предмет	Имеет	Автор	д/б	М:1
Источник	Отдаёт	Предмет	м/б	1:М
Предмет	Принимается	Источник	д/б	М:1
Фонд	Включает	Предмет	м/б	1:М
Предмет	Числится	Фонд	д/б	М:1
Работник	Отвечает	Предмет	м/б	1:М
Предмет	Приписан	Работник	д/б	М:1
Предмет	Прописывается	Строка листа передачи	м/б	1:М

Продолжение таблицы 13

Имя сущности субъекта	Имя связи	Имя сущности объекта	Обязательность	Мощность
Строка листа передачи	Прописывает	Предмет	д/б	М:1
Выставка	Прописывается	Строка листа передачи	м/б	1:М
Строка листа передачи	Прописывает	Выставка	д/б	М:1
Реставрация	Прописывается	Строка листа передачи	м/б	1:М
Строка листа передачи	Прописывает	Реставрация	д/б	М:1

Помимо ограничений на атрибуты, заданных в таблицах 2-12, можно сформулировать дополнительные ограничения семантической целостности:

- музейный предмет нельзя передать на выставку, если он находится на реставрации;
- музейный предмет нельзя передать на реставрацию, если он находится на выставке.

Такие ограничения называются ограничениями перехода. Кроме того, дата передачи на выставку не может быть больше даты окончания выставки. Это ограничение не задано явно, но оно очевидно. В описании предметной области задано явное бизнес-правило - работник музея отвечает не более чем за 20 музейных экспонатов. Перечисленные ограничения невозможно задать декларативным путем, т.е. при описании таблиц. Обеспечить проверку этих ограничений можно программным путем с помощью триггеров, хранимых процедур или функций.

2.1.2 Анализ круга пользователей системы и их информационных потребностей

В данной предметной области можно выделить 2 категории пользователей и соответствующие функции:

- сотрудники, ответственные на экспонаты. Ведут карточки учета;
- сотрудники отдела кадров. Ведут кадровый учет;

- заведующие фондами. Принимают предмет на хранение, формируют распоряжения на перемещение предмета;

- руководитель музея. Визуирует распоряжения на перемещение экспонатов, отслеживает местоположение экспонатов.

2.1.3 Определение требований к объему дисковой памяти

Проектировщику необходимо определить требования к ресурсам вычислительной системы, в том числе к объему жесткого диска, на котором предполагается хранить базу данных. Для этого необходимо иметь представление о границах предметной области, т.е. знать количество экспонатов и сотрудников, а также предполагать ориентировочно, как часто экспонаты передаются на выставки, как часто они реставрируются, другими словами, какова будет интенсивность транзакций в проектируемой системе.

На основе результатов анализа ПО можно приблизительно оценить объём памяти, требуемой для хранения данных. Будем считать, что:

- в музее имеется 5 фондов, в каждом из которых порядка 5 000 экспонатов (для хранения сведений об одном экспонате выделим 1 Кб);

- каждый экспонат один раз в год передается на выставки (0.5 Кб на соответствующую запись);

- в музее работают 50 сотрудников (по 0.5Кб на каждого сотрудника);

Тогда объём памяти для хранения данных за первый год примерно составит:

$$M_d = 5000*1+5000*0.5+50*0.5 = 7525 \text{ Кб},$$

Необходимо учесть, что для хранения таких вспомогательных структур как индексы, генераторы и т.д. требуется дополнительная память. Кроме того, ежегодно объем занимаемой БД памяти будет увеличиваться, поскольку БД отражает реальную работу музея. В связи с этим рекомендуется удвоить вычисленный объем памяти, необходимый для хранения данных:

$$M_d = 2*7525= 15050 \text{ Кб}$$

2.2 Выбор СУБД

Для реализации разрабатываемой базы данных можно использовать одну из таких реляционных СУБД как Interbase, Firebird, MS SQL Server и т.д. В пояснительной записке следует обосновать выбор конкретной СУБД, описав ее функциональные возможности и особенности.

2.3 Даталогическое проектирование БД

На этом этапе инфологическая модель предметной области преобразуется в такую даталогическую модель, которую поддерживает выбранная СУБД.

2.3.1 Преобразование ER-диаграммы в схему реляционной БД

Для ER-диаграммы существует алгоритм однозначного преобразования ее в реляционную модель. Он состоит из следующих этапов:

- каждая сущность становится отношением. Атрибуты сущности становятся атрибутами отношений;
- разрешение связей:
 - а) разрешение связей 1:1;

Если связь обязательная, столбец внешнего ключа добавляется в таблицу на обязательной стороне связи. Эта таблица становится подчиненной, а другая - главной. Внешним ключом становится первичный ключ основной таблицы. Следует добавить также ограничение для внешнего ключа *Not null*.

Если связи необязательны в обоих направлениях, можно выбрать, в какую таблицу поместить столбец внешнего ключа. Если, например, строка одной таблицы создается раньше, чем в другой, можно сделать эту таблицу главной, а другую – подчиненной;

- б) разрешение связей 1:M;

Таблица, стоящая на стороне связи 1, является главной, а таблица, стоящая на стороне связи M, является подчиненной. В подчиненную таблицу добавляется столбец внешнего ключа. Внешним ключом становится первичный

ключ основной таблицы. При этом, если сущность использует связь как часть своего UID, внешний ключ должен входить в состав первичного ключа таблицы.

в) разрешение рекурсивных связей 1:M.

При создании таблицы создается столбец внешнего ключа, указывающего на другую строку этой же самой таблицы.

- преобразование супертипов;

Такое преобразование может быть осуществлено двумя путями:

а) супертип преобразуется в одно отношение, включающее все атрибуты, характерные как для супертипа, так и для всех его подтипов;

б) в соответствии с количеством подтипов создается несколько отношений.

Руководствуясь данным алгоритмом, получим совокупность схем отношений, представленных на рисунке 2.

Можно выделить отношения, содержащие условно-постоянную информацию:

- фонд (код, наименование);

- источник (код, наименование);

- улица (код, наименование);

- реставрация (код, степень);

- страна (код, наименование).

Отношение находится в третьей нормальной форме, если оно находится во второй нормальной форме и среди его атрибутов отсутствуют транзитивные зависимости. Выделенные отношения находятся во второй нормальной форме, так как первичный ключ отношения простой. В отношениях не может существовать транзитивных зависимостей, так как для транзитивности необходимо наличие как минимум трёх атрибутов. Следовательно, отношения находятся в третьей нормальной форме.

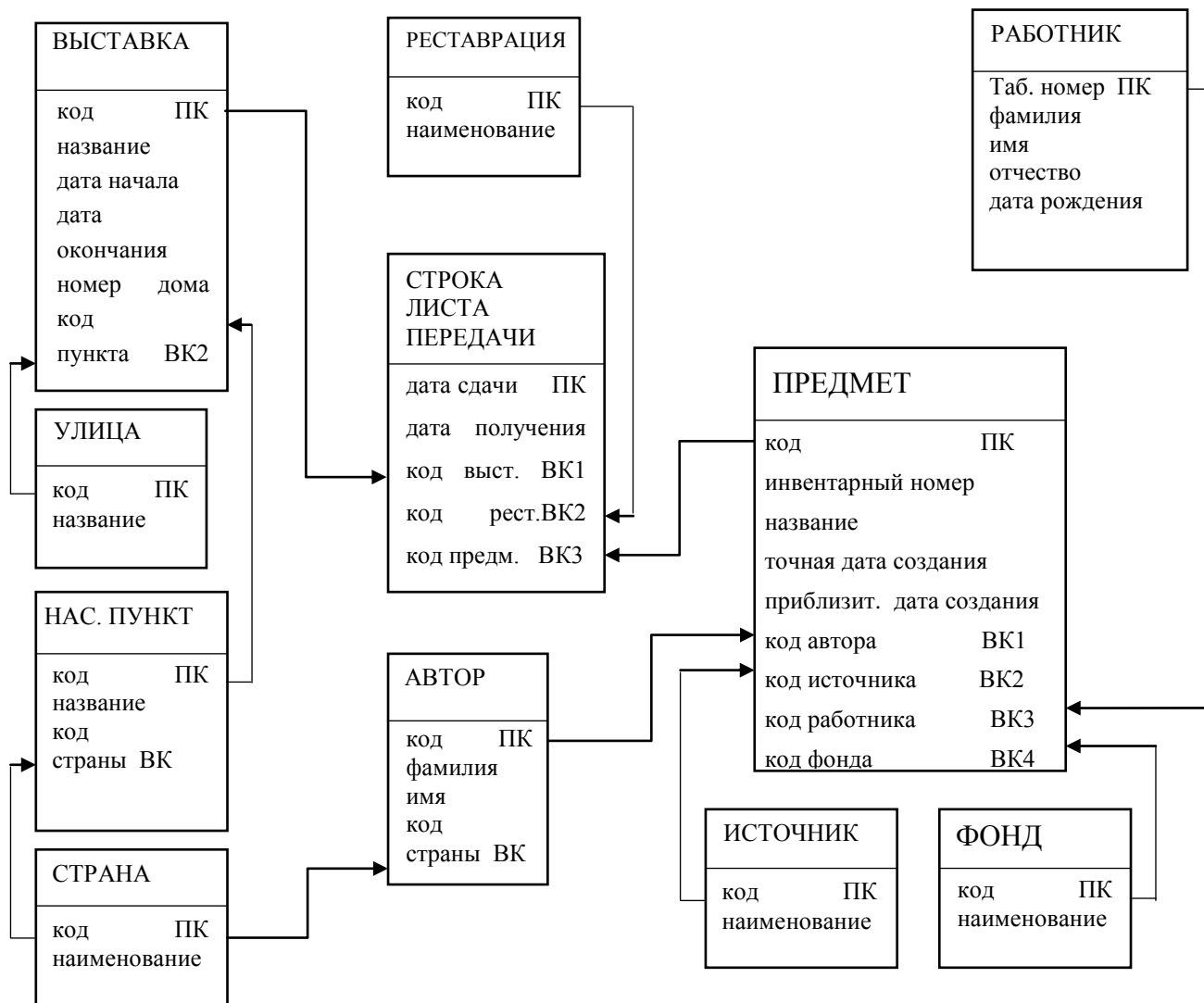


Рисунок 2 - Логическая структура БД

Следующие отношения находятся в третьей нормальной форме, так как первичный ключ каждого отношения простой и в отношении отсутствуют транзитивные зависимости:

- Предмет (код, инвентарный номер, название, точная дата создания, приблизительная дата создания, дата списания, код фонда, код работника, код автора, код источника);
- Работник (код, фамилия, имя, отчество, дата рождения);
- Автор (код, фамилия, имя, код страны);
- Населённый пункт (код, наименование, код страны);
- Выставка (код, название, дата начала, дата завершения, номер дома).

Отношение «Строка листа передачи (дата сдачи, дата получения, код предмета, код выставки, код реставрации) находится во второй нормальной форме, а так как среди атрибутов отсутствуют транзитивные зависимости, отношение по определению находится в третьей нормальной форме.

2.3.2 Описание прав доступа к объектам БД

Опишем для каждой группы пользователей права доступа к каждой таблице. Права доступа должны быть распределены так, чтобы для каждого объекта БД был хотя бы один пользователь, который имеет право добавлять и удалять данные из объекта. Права приведены в таблице 16. Используются следующие сокращения:

s – чтение данных (select);

i – добавление данных (insert);

u – модификация данных (update);

d – удаление данных (delete).

Таблица 16 - Права доступа к таблицам для групп пользователей

Таблицы	Группы пользователей			
	Сотрудники	Заведующие фондами	Директор музея	Сотрудники отдела кадров
Фонд	S	S	SIUD	S
Источник	S	SIUD	SIUD	S
Улица	S	S	S	SIUD
Реставрация	S	SIUD	SIUD	S
Страна	S	S	S	SIUD
Работник	S	S	S	SIUD
Населенный пункт	S	S	S	SIUD
Выставка	S	S	SIUD	S
Строка листа передачи	S	SIUD	S	S
Автор	S	SIUD	SIUD	S
Предмет	SIUD	SIUD	SIUD	S

2.4 Проектирование внутреннего уровня БД

На этом этапе даталогическая модель преобразуется во внутреннюю схему, то есть такую модель, которая может быть реализована в среде выбранной СУБД.

Для реляционной модели этап проектирования внутреннего уровня базы данных включает в себя следующие шаги:

- проектирование структур таблиц;
- проектирование индексов. Индекс (индексный файл) необходим для упорядочивания содержимого таблицы по значениям одного или нескольких столбцов с целью ускорения доступа к данным;
- реализация ограничений реляционной и семантической целостности средствами выбранной СУБД.

Следует помнить, что далеко не все ограничения могут быть реализованы декларативным путем. Сложные ограничения семантической целостности реализуются процедурным путем, т.е. с помощью хранимых процедур и триггеров.

Результаты этапа проектирования внутреннего уровня реляционной базы данных документируются в виде SQL-скриптов, содержащих операторы языка SQL для создания различных объектов БД, таких как таблицы, представления, триггеры, хранимые процедуры и функции.

Одной из важнейших составляющих проекта базы данных является разработка средств защиты БД. Для защиты от несанкционированного доступа каждому пользователю доступ к данным предоставляется только в соответствии с его правами доступа, набор которых также является составной частью проекта БД.

2.4.1 Техническое описание таблиц

Перед написанием SQL-скриптов для создания объектов проектируемой БД полезно выполнить техническое описание БД в терминах выбранной СУБД. Это описание отражает особенности реализации базы данных в среде данной

СУБД и облегчает написание SQL-операторов. В описании должны быть указаны те ограничения семантической целостности, которые будут реализованы декларативным путем.

В данном примере в качестве целевой СУБД используется SQL Server.

Техническое описание таблицы «Работник» представлено в таблице 14.

Таблица 14 – Техническое описание таблицы «Работник»

Имя столбца	Тип данных	Длина	Точность	Тип ключа	Опциональность	Логическое ограничение
idTabelniNomer	int	-	-	Primary key	Not null	>0
Family	Char	20	-	-	Not null	-
Name	Char	20	-	-	Not null	-
Otchestvo	Char	20	-	-	Not null	-
DataRogdeniya	Date	-	-	-	Not null	>1900-01-01 and <getdate()

Техническое описание таблицы «Фонд» представлено в таблице 15.

Таблица 15 – Техническое описание таблицы «Фонд»

Имя столбца	Тип данных	Длина	Точность	Тип ключа	Опциональность	Логическое ограничение
idFond	int	-	-	Primary key	Not null	>0
NameFond	Char	30	-	-	Not null	-

Техническое описание таблицы «Источник» представлено в таблице 16.

Таблица 16 – Техническое описание таблицы «Источник»

Имя столбца	Тип данных	Длина	Точность	Тип ключа	Опциональность	Логическое ограничение
idIstochnic	int	-	-	Primary key	Not null	>0
NameIstochnic	Char	50	-	-	Not null	-

Техническое описание таблицы «Реставрация» представлено в таблице 17.

Таблица 17 – Техническое описание таблицы «Реставрация»

Имя столбца	Тип данных	Длина	Точность	Тип ключа	Опциональность	Логическое ограничение
idRestavratsiya	int	-	-	Primary key	Not null	>0
Stepen	Char	20	-	-	Not null	-

Техническое описание таблицы «Страна» представлено в таблице 18.

Таблица 18 – техническое описание таблицы «Страна»

Имя Столбца	Тип Столбца	Длина	Точность	Тип Ключа	Опциональность	Логическое ограничение
idStrana	int	-	-	Primary key	Not null	>0
NameStrana	Char	30	-	-	Not null	-

Техническое описание таблицы «Улица» представлено в таблице 19.

Таблица 19 – Техническое описание таблицы «Улица»

Имя столбца	Тип данных	Длина	Точность	Тип ключа	Опциональность	Логическое ограничение
idUlica	int	-	-	Primary key	Not null	>0
NameUlica	Char	30	-	-	Not null	-

Техническое описание таблицы «Автор» представлено в таблице 20.

Таблица 20 – Техническое описание таблицы «Автор»

Имя столбца	Тип данных	Длина	Точность	Тип ключа	Опциональность	Логическое ограничение
idAutor	int	-	-	Primary key	Not null	>0
Family	Char	20	-	-	Not null	-
Name	Char	20	-	-	Not null	-
vcStrana	int	10	0	Foreign key	Not null	>0

Техническое описание таблицы «Предмет» представлено в таблице 21.

Таблица 21 – Техническое описание таблицы «Предмет»

Имя столбца	Тип данных	Длина	Точность	Тип ключа	Опциональность	Логическое ограничение
idPredmet	int	-	-	Primary key	Not null	>0

Продолжение таблицы 21

Имя столбца	Тип данных	Длина	Точность	Тип ключа	Опциональность	Логическое ограничение
NamePredmet	Char	30	-	-	Not null	-
TDataSozdaniya	Date	-	-	-	Null	<getdate()
PDataSozdaniya	Date	-	-	-	Null	<getdate()
InventarniNomer	Char	20	-	-	Not null	-
DataSpisaniya	Date	-	-	-	Null	<=getdate()
vcIstochnic	int	-	-	Foreign key	Not null	>0
vcAutor	Int	-	-	Foreign key	Not null	>0
vcRabotnik	Int	-	-	Foreign key	Not null	>0
vcFond	Int	-	-	Foreign key	Not null	>0

Техническое описание таблицы «Населённый пункт» представлено в таблице 22.

Таблица 22 – Техническое описание таблицы «Населённый пункт»

Имя столбца	Тип данных	Длина	Точность	Тип ключа	Опциональность	Логическое ограничение
idNaselPunct	int	-	-	Primary key	Not null	>0
NameNaselPunct	Char	30	-	-	Not null	-
vcStrana	int	-	-	Foreign key	Not null	>0

Техническое описание таблицы «Выставка» представлено в таблице 23.

Таблица 23 – Техническое описание таблицы «Выставка»

Имя Столбца	Тип Столбца	Длина	Точность	Тип Ключа	Опциональность	Логическое ограничение
idVistavka	int	-	-	Primary key	Not null	>0
NameVistavka	Char	20	-	-	Not null	-

Продолжение таблицы 23

Имя Столбца	Тип Столбца	Длина	Точность	Тип Ключа	Опцио-нальность	Логическое ограничение
DataNachala	Date	-	-	-	Not null	-
DataOkonchaniya	Date	-	-	-	Not null	>DataNachala
NomerDoma	smallint	-	-	-	Not null	>0
vcUlica	int	-	-	Foreign key	Not null	>0
vcNaselPunct	int	-	-	Foreign key	Not null	>0

Техническое описание таблицы «Строка листа передачи» представлено в таблице 24.

Таблица 24 – Техническое описание таблицы «Строка листа передачи»

Имя Столбца	Тип Столбца	Длина	Точность	Тип Ключа	Опцио-нальность	Логическое ограничение
DataSdachi	date	-	-	Primary key	Not null	>getdate()
DataPolucheniya	date	-	-	-	Not null	>DataSdachi
vcPredmet	int	-	-	Foreign key, Primary key	Not null	>0
vcVistavka	int	-	-	Foreign key	Null	>0
vcRestavratsiya	int	-	-	Foreign key	Null	>0

2.4.2 Создание таблиц

В этом разделе после каждого скрипта следует разместить скриншот с результатами его выполнения.

SQL-скрипт базы данных «MuseumDB»:

```
create database MuseumDB
```

```
on
```

```
(
```

```
name = Museum_data,
```

```
filename = 'd:\MuseumDB\Museum.mdf',
```

```

size = 10mb,
maxsize= 2gb,
filegrowth = 50mb
)
log on
(
name = Museum_log,
filename = 'd:\MuseumDB\Museum.ldf',
size = 5mb,
maxsize = 2gb,
filegrowth = 5%
)

```

SQL-скрипт таблицы «Работник»:

```

create table Rabotnik
(
idTabelniNomer int identity(1,1) not null,
Family char(20) not null,
Name char(20) not null,
Otchestvo char(20) not null,
DataRogdeniya date not null check(DataRogdeniya>'1900-01-01' and
DataRogdeniya<getdate()),
vcFond int not null,
constraint PK_idTabelniNomer primary key (idTabelniNomer),
)

```

SQL-скрипт таблицы «Фонд»:

```

create table Fond
(
idFond int identity(1,1) not null,
NameFond char(30) not null,
constraint PK_idFond primary key (idFond)
)

```

SQL-скрипт таблицы «Источник»:

```

create table Istochnic
(
idIstochnic int identity(1,1) not null,
NameIstochnic char(50) not null,
constraint PK_idIstochnic primary key (idIstochnic)
)

```

SQL-скрипт таблицы «Страна»:

```

create table Strana

```

```
(
  idStrana int identity(1,1) not null,
  NameStrana char(30) not null,
  constraint PK_idStrana primary key (idStrana)
)
```

SQL-скрипт таблицы «Улица»:

```
create table Ulica
(
  idUlica int identity(1,1) not null,
  NameUlica char(30) not null,
  constraint PK_idUlica primary key (idUlica)
)
```

SQL-скрипт таблицы «Реставрация»:

```
create table Restavratsiya
(
  idRestavratsiya int identity(1,1) not null,
  Stepen char(20) not null,
  constraint PK_idRestavratsiya primary key (idRestavratsiya)
)
```

SQL-скрипт таблицы «Автор»:

```
create table Autor
(
  idAutor int identity(1,1) not null,
  Family char(20) not null,
  Name char(20) not null,
  vcStrana int not null,
  constraint PK_idAutor primary key (idAutor),
  constraint FK_vcAutorStrana foreign key (vcStrana) references Strana
(idStrana) )
```

SQL-скрипт таблицы «Выставка»:

```
create table Vistavka
(
  idVistavka int identity(1,1) not null,
  NameVistavka char(30) not null,
  DataNachala date not null,
  DataOkonchaniya date not null,
  NomerDoma smallint check(NomerDoma > 0) not null,
  vcUlica int not null,
  vcNaselPunct int not null,
  constraint PK_idVistavka primary key (idVistavka),
  constraint FK_VistavkaUlica foreign key (vcUlica) references Ulica (idUlica),
```

```
constraint FK_VistavkaNaselPunct foreign key (vcNaselPunct) references
NaselPunct (idNaselPunct),
constraint RelationDate check(DataOkonchaniya >= DataNachala)
)
```

SQL-скрипт таблицы «Предмет»:

```
create table Predmet
(
idPredmet int identity(1,1) not null,
NamePredmet char(30) not null,
TDataSozdaniya date null check(TDataSozdaniya<getdate()),
PDataSozdaniya date null check(PDataSozdaniya<getdate()),
InventarniNomer char(20) not null,
DataSpisaniya date null check(DataSpisaniya<=getdate()),
vcIstochnic int not null,
vcAutor int null,
vcRabotnik int not null,
vcFond int not null,
constraint PK_idPredmet primary key (idPredmet),
constraint FK_PredmetIstochnic foreign key (vcIstochnic) references Istochnic
(idIstochnic),
constraint FK_PredmetAutor foreign key (vcAutor) references Autor (idAutor),
constraint FK_PredmetRabotnik foreign key (vcRabotnik) references Rabotnik
(idTabelniNomer),
constraint FK_PredmetFond foreign key (vcFond) references Fond (idFond),
constraint DateSozdaniya check(TDataSozdaniya is null and PDataSozdaniya
is not null or TDataSozdaniya is not null and PDataSozdaniya is null)
)
```

SQL-скрипт таблицы «Населённый пункт»:

```
create table NaselPunct
(
idNaselPunct int identity(1,1) not null,
NameNaselPunct char(30) not null,
vcStrana int not null,
constraint PK_idNaselPunct primary key (idNaselPunct),
constraint FK_vcStrana foreign key (vcStrana) references Strana (idStrana)
)
```

SQL-скрипт таблицы «Строка листа передачи»:

```
create table StrokaListaPeredachi
(
DataSdachi date not null,
DataPolucheniya date not null,
vcPredmet int not null,
```

```

vcVistavka int,
vcRestavratsiya int,
constraint FK_StrokaListaPeredachivcPredmet foreign key (vcPredmet)
references Predmet (idPredmet),
constraint FK_StrokaListaPeredachivcVistavka foreign key (vcVistavka)
references Vistavka (idVistavka),
constraint FK_StrokaListaPeredachivcRestavratsiya foreign key
(vcRestavratsiya) references Restavratsiya (idRestavratsiya),
constraint StrokaListaPeredachiDateRelation check(DataPolucheniya >=
DataSdachi),
constraint StrokaListaPeredachiFKRelation check(vcVistavka is null and
vcRestavratsiya is not null or vcRestavratsiya is null and vcVistavka is not null),
constraint PK_StrokaListaPeredachi primary key (DataSdachi,vcPredmet)
)

```

2.4.3 Создание запросов

1. SQL-скрипт запроса на вывод полной информации об экспонатах.

```

select p.idPredmet, p.InventarniNomer, p.NamePredmet, p.TDataSozdaniya,
p.PDataSozdaniya, a.Name, a.Family, i.NameIstochnic, f.NameFond,
r.idTabelniNomer, r.Family, r.Name, r.Otchestvo, r.DataRogdeniya
from Predmet p, Fond f, Rabotnik r, Autor a, Istochnic i
where p.vcFond = f.idFond and p.vcRabotnik = r.idTabelniNomer and
p.vcAutor = a.idAutor and p.vcIstochnic = i.idIstochnic

```

2. SQL-скрипт запроса на определение фонда с максимальным числом предметов :

```

select f.NameFond, f.idFond
from Fond f, Predmet p
where p.vcFond = f.idFond
group by f.NameFond, f.idFond
having count(*) >= all
(select count(*)
from Predmet
group by vcFond)

```

2.4.4 Создание дополнительных объектов БД

1. SQL-скрипт функции «PeremesheniyaPredmeta»:

Данная функция возвращает таблицу, содержащую строки с информацией о всех перемещениях предмета с заданным идентификационным номером

```

create function PeremesheniyaPredmeta
(
    @vcPredmet smallint
)
returns table
as
return
    (select DataSdachi, DataPolucheniya, vcVistavka, vcRestavratsiya
    from StrokaListaPeredachi
    where vcPredmet = @vcPredmet)

```

2. SQL-скрипт триггера «Число предметов у работника »:

Данный триггер обеспечивает выполнение ограничения семантической целостности, заданное в описании предметной области – каждый сотрудник отвечает не более чем за 20 экспонатов

```

create trigger ui_Predmet on dbo.Predmet
for update,insert
as
declare @RabotnikColPredmet smallint
declare @KodRabotnik smallint

select @KodRabotnik = vcRabotnik
from inserted

select @RabotnikColPredmet = count(vcRabotnik)
from Predmet
where vcRabotnik = @KodRabotnik

if @RabotnikColPredmet >= 20
begin
    rollback transaction
    print 'Максимальное число предметов, за которые отвечает
работник, переполнено'
end

```

3. SQL-скрипт триггера « Передача предмета »:

Данный триггер обеспечивает выполнение ограничений переходов. Так предмет не может быть передан на выставку или реставрацию, если он не находится в музее. Кроме того, если предмет передается на выставку, то дата передачи не может быть больше даты завершения выставки. Триггер будет срабатывать при добавлении новой строки в таблицу StrokaListaPeredachi

```

create trigger i_StrokaListaPeredachi on dbo.StrokaListaPeredachi
for insert
as
declare @DataSdachi date
declare @KodVistavki smallint
declare @DataOkonchaniya date
declare @KodPredmeta smallint
declare @Mestopologenie varchar(30)

select @DataSdachi = DataSdachi, @KodVistavki = vcVistavka,
@KodPredmeta = vcPredmet
from inserted;

select @Mestopologenie = dbo.MestopologeniePredmeta(@KodPredmeta,
@DataSdachi);
print @Mestopologenie
if @Mestopologenie <> 'Предмет в музее'
begin
    rollback transaction
    print 'Предметне может быть передан'
end

if @KodVistavki is not null
begin
    select @DataOkonchaniya = DataOkonchaniya
    from Vistavka
    where idVistavka = @KodVistavki;

    if @DataSdachi > @DataOkonchaniya
    begin
        rollback transaction
        print 'Ошибка передачи. Проверьте корректность даты
передачи'
    end
end
end

```

4. SQL-скрипт функции «Местоположение предмета»:
 Данная функция предназначена для определения местонахождения предмета на текущую дату, поскольку с помощью оператора Select реализовать запрос невозможно в виду его сложности.

```

create function MestopologeniePredmeta
(
    @idPredmet int,

```

```

    @Date date
)
returns char(30)
begin
    declare @DataSpisaniya date
    declare @MestoVistavka char(30)
    declare @MestoRestavratsiya char(30)
    declare @Itog char(30)

    select @MestoVistavka = v.NameVistavka
    from PeremesheniyaPredmeta(@idPredmet) PeremPred, Vistavka v
    where PeremPred.vcVistavka = v.idVistavka and
        DataSdachi <= @Date and DataPolucheniya > @Date;

    select @MestoRestavratsiya = r.Stepen
    from PeremesheniyaPredmeta(@idPredmet) PeremPred, Restavratsiya r
    where PeremPred.vcRestavratsiya = r.idRestavratsiya and
        DataSdachi <= @Date and DataPolucheniya > @Date;

    if @MestoVistavka is not null
        set @Itog = 'Выставка: ' + @MestoVistavka
    else
        if @MestoRestavratsiya is not null
            set @Itog = 'Реставрировался: ' + @MestoRestavratsiya
        else
            set @Itog = 'Предметвмузее'

    select @DataSpisaniya = DataSpisaniya
    from Predmet
    where idPredmet = @idPredmet;

    if @DataSpisaniya is not null
        if @Date >= @DataSpisaniya
            set @Itog = 'Предмет списан'

    return @Itog
end

```

Для того, чтобы можно было работать с созданными функциями, необходимо назначить права доступа к ним. Права доступа соответствующих групп представлены в таблице 25.

Здесь X означает право на выполнение функции.

Таблица 25 – Права доступа к функциям

Функции	Группы пользователей			
	Сотрудники	Заведующие фондами	Директор музея	Сотрудники отдела кадров
PeremesheniyaPredmeta	X	X	X	-
MestopologeniePredmeta	X	X	X	-

2.4.5 Назначение прав доступа

Создадим роль Zav_fond для группы пользователей, в которую войдут заведующие фондами музея. В соответствии с таблицей 2 назначим этой группе права:

```
Grant select, insert, update, delete on IstochnictoZav_fond;
Grant select, insert, update, delete on Restavratsia to Zav_fond;
Grant select, insert, update, delete on StrokaListaPeredachi to Zav_fond;
Grant select, insert, update, delete on Autor to Zav_fond;
Grant select, insert, update, delete on Predmet to Zav_fond;
Grant select on Fond to Zav_fond;
Grant select on Ulica to Zav_fond;
Grant select on Strana to Zav_fond;
Grant select on Rabotnic to Zav_fond;
Grant select on NaselPunct to Zav_fond;
Grant select on Vistavka to Zav_fond;
```

Необходимо также назначить права на выполнение функций, определенных пользователем:

```
Grant execute on PeremesheniyaPredmeta to Zav_fond;
Grant execute on MestopologeniePredmeta to Zav_fond;
```

Аналогично назначаются права доступа к объектам БД для остальных групп пользователей.

2.4.6 Создание индексов

Индексы создаются для ускорения доступа к данным. СУБД автоматически создает индексы по столбцам первичных ключей и по тем столбцам, которые описаны как уникальные. Для ускорения выполнения многотабличных запросов, которые требуют соединения исходных таблиц, необходимо также создать индексы по столбцам внешних ключей.

Так в таблице Predmet имеется 3 внешних ключа: vcAutor, vcRabotnik, vcFond для связи с таблицами Autor, Rabotnik и Fond. В то же время эти таблицы участвует в нескольких запросах. Команды создания соответствующих индексов выглядят следующим образом:

```
create nonclustered index Autor_ind on Predmet (vcAutor);  
create nonclustered index Rabotnik_ind on Predmet (vcPredmet);  
create nonclustered index Fond_ind on Predmet (vcFond);
```

3 Оформление курсового проекта

Пояснительная записка к курсовому проекту должна содержать:

1. Титульный лист с указанием всех реквизитов учебного заведения, кафедры.

Записка называется «Пояснительная записка к курсовому проекту по дисциплине «Базы данных» на тему «Название курсового проекта». На титульном листе указываются также фамилия, инициалы, группа исполнителя, фамилия, инициалы руководителя.

2. Лист технического задания.

Лист, верхняя часть которого содержит почти такой же заголовок, что и титульный: «Техническое задание на курсовой проект по дисциплине «Базы данных» на тему «Название курсового проекта».

Целью курсового проекта является получение практических навыков разработки реляционной базы данных и реализации для заданной предметной области.

3. Лист с большим штампом, заполненным по ГОСТ и аннотация.

Аннотация объемом не более 0.5 страницы расположена после титульного листа и предшествует «Содержанию». Аннотация – это краткая характеристика содержания курсового проекта. Аннотация содержит ответы на вопросы: «Каков объект разработки? Что было реализовано? В каком виде представлены результаты?»

Содержание пояснительной записки должно включать следующие пункты:

Введение

Здесь дается общая характеристика проекта, раскрывается актуальность работы, формулируются цель(цели) курсового проектирования, задачи, которые необходимо решить для достижения цели

1 Словесное описание предметной области

2 Проектирование и создание БД

2.1 Инфологическое проектирование

2.1.1 Разработка ER-диаграммы предметной области

2.1.2 Документирование ER-диаграммы

2.1.3 Анализ круга пользователей системы и их информационных потребностей

2.1.4 Определение требований к объему дисковой памяти

2.2 Выбор СУБД

2.3 Дatalogическое проектирование реляционной БД

2.3.1 Схема БД, приведенная к 3НФ

2.3.2 Графическое представление связей между таблицами (логическая структура БД)

2.3.3 Описание прав доступа к объектам БД

2.4 Проектирование внутреннего уровня БД

2.4.1 Техническое описание таблиц

2.4.2 Тексты SQL-скриптов для создания таблиц

2.4.3 Тексты SQL-скриптов для создания запросов

2.4.4 Тексты SQL-скриптов для создания дополнительных объектов БД

2.4.5 Назначение прав доступа

Заключение

Список использованных источников

Приложения

В приложения можно вынести инфологическую и даталогическую модели данных в том случае, когда разместить их по тексту записки затруднительно.

Список использованных источников

1 Советов, Б. Я. Базы данных: теория и практика: учебник для бакалавров / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской.- 2-е изд. - Москва : Юрайт, 2013. - 463 с. - (Бакалавр. Базовый курс). - Прил.: с. 386-458. - Библиогр.: с.459-460. - ISBN 978-5-9916-2913-3.

2 Карпова, И.П. Базы данных: Учебное пособие по курсу "Базы данных"/ И.П. Карпова. - СПб: Издательство «Питер», 2015. - 240 с. - ISBN: 978-5-496-00546-3.

3 Кузнецов, С. Д. Базы данных. Модели и языки : учеб. для вузов / С. Д. Кузнецов . - М. : Бином, 2008. - 720 с. - Прил.: с. 685-700. - Предм. указ.: с. 701-720. - ISBN 978-5-9518-0132-6.

4 Кусмарцева, Н.Н. Разработка и эксплуатация удаленных баз данных [Электронный ресурс]: учебное пособие/ Кусмарцева Н.Н.-Электрон. текстовые данные.- Волгоград: Волгоградский институт бизнеса, Вузовское образование, 2013.- 143 с.- Режим доступа: <http://www.iprbookshop.ru/11343>

5 Королева О.Н. Базы данных [Электронный ресурс]: курс лекций/ Королева О.Н., Мажукин А.В., Королева Т.В.- Электрон. текстовые данные.- М.: Московский гуманитарный университет, 2012.- 66 с.- Режим доступа: <http://www.iprbookshop.ru/14515>

Приложение А

ER-диаграмма предметной области

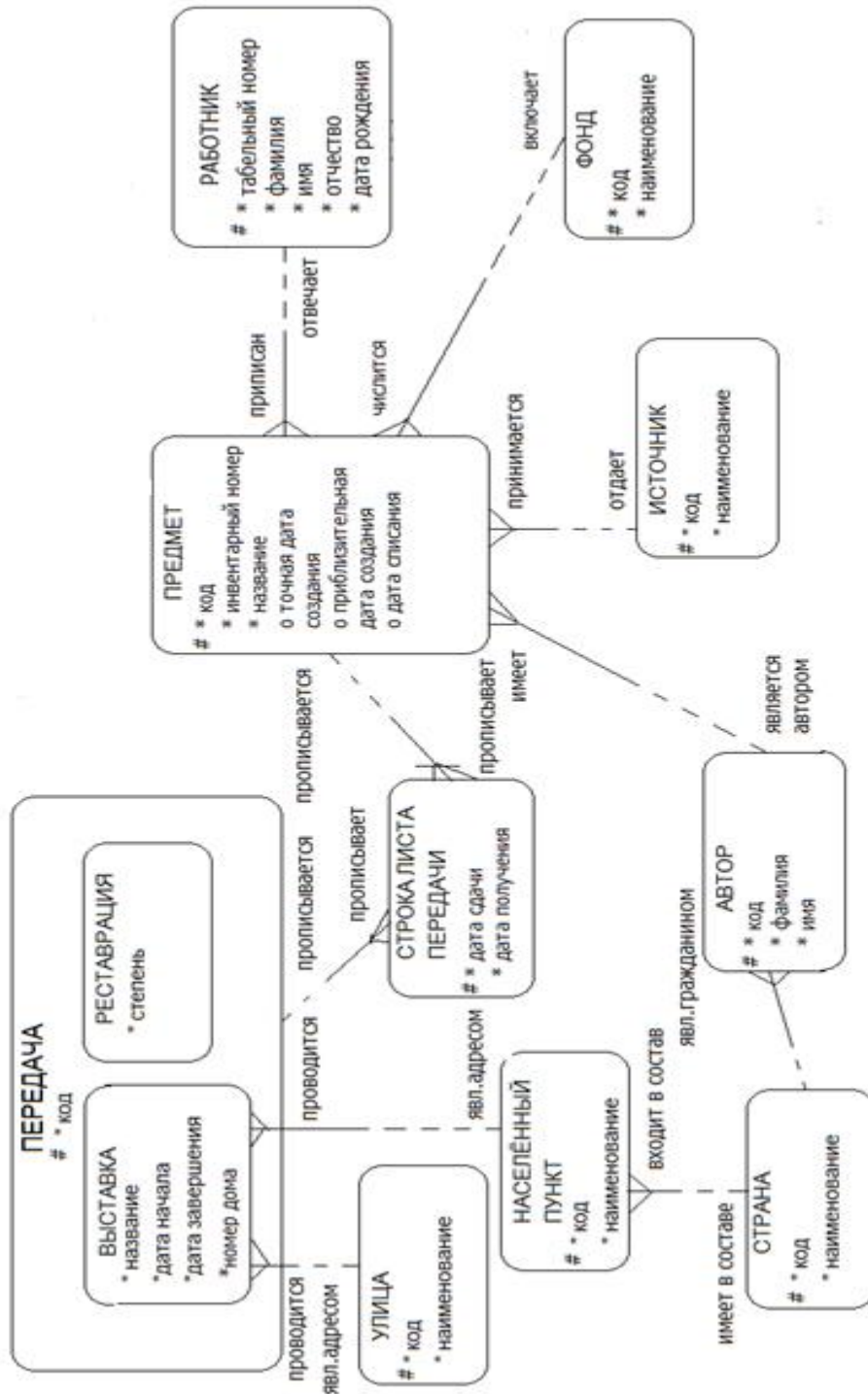


Рисунок А.1 – ER-диаграмма предметной области