МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)

А. С. ЛУКАНОВ, А. С. БАБИЧ

БАЗЫ ДАННЫХ.
ОСНОВЫ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ

Рекомендовано редакционно-издательским советом федерального государственного автономного образовательного учреждения высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева» в качестве учебного-методического пособия для обучающихся по основной образовательной программе высшего образования по направлениям подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем, 02.03.01 Математика и компьютерные науки, 01.03.02 Прикладная математика и информатика, 41.03.05 Документоведение и архивоведение и специальностям 01.05.01 Фундаментальная математика и механика, 01.05.10 Компьютерная безопасность

САМАРА
Издательство Самарского университета
2026

УДК 004.65(075)
ББК 3972.34я7
Л840

Рецензенты: д-р физ.-мат. наук, проф. В. П. Радченко;
канд. физ.-мат. наук, доц. А. Н. Крутов

Луканов, Александр Сергеевич

Л840 **Базы данных. Основы реляционных баз данных**: учебно-методическое пособие / *А.С. Луканов, А.С. Бабич.* – Самара, издательство Самарского университета, 2026. – 46 с.: ил.

ISBN 978-5-7883-2318-3

Рассмотрены теоретические и практические аспекты реляционных баз данных в среде СУБД LibreOffice Base в формате лабораторных работ. Каждая тема содержит базовые теоретические сведения и набор заданий разного уровня.

Учебно-методическое пособие предназначено для обучающихся как академических, так и инженерных направлений подготовки при изучении дисциплины «Базы данных».

Подготовлено на кафедре информатики и вычислительной техники Самарского университета.

УДК 004.65(075)
ББК 3972.34я7

ISBN 978-5-7883-2318-3 © Самарский университет, 2026

ОГЛАВЛЕНИЕ

Введение 4
1 **Фундаментальные понятия реляционной базы данных** 6
1.1 Понятие «Информационная система» 6
1.2 Реляционная база данных. 6
1.3 Потенциальные и внешние ключи РМД. 7
1.4 Класс прикладного программного обеспечения - СУБД 8
1.5 Установка СУБД LibreOffice Base. 9
2 **Создание БД как сложно структурированных и связанных между собой данных**. . 11
2.1 Учебная база данных «Деканат» 11
2.2 Создание базы данных. 11
2.3 Создание реляционных отношений – таблиц базы данных «Деканат» 12
2.4 Заполнение таблиц. 15
2.5 Установка связей между таблицами–отношениями. 17
2.6 Вопросы к собеседованию при сдаче лабораторной работы № 1. 19
3 **Создание форм. Манипулирование данными БД с помощью форм**. 21
3.1 Создание табличных и «линейных» форм. 21
3.2 Усложнение дизайна форм. 25
3.3 Использование субформ. 28
3.4 Вопросы к собеседованию при сдаче лабораторной работы № 2. 29
4 **Создание запросов. Запросы как элемент ИС БД**. 30
4.1 Создание простых запросов. 30
4.2 Запросы с параметрами. 32
4.3 Запросы на основе групповых функций. 34
4.4 Вопросы к собеседованию при сдаче лабораторной работы № 3. 35
5 **Создание отчётов. Отчёты как элемент ИС БД**. 36
5.1 Создание простых отчетов. 36
5.2 Создание отчетов с группированием данных. 39
5.3 Вопросы к собеседованию при сдаче лабораторной работы № 4. 40
6 **Лабораторная работа «Индивидуальное задание»** 41
6.1 Постановка задачи. 41
6.2 Варианты заданий. 41
Список использованных источников. 45

ВВЕДЕНИЕ

На сегодняшний день современные информационные технологии (СИТ) абсолютно определяют все аспекты социальной среды любого человека, будь то образование, производственные отношения или даже элементарные бытовые условия. А среди всех СИТ одно из лидирующих положений занимает технология баз данных, на основе которой и строятся информационные системы на основе баз данных (ИС БД).

Человек в своей повседневной деятельности встречается с ИС БД практически на каждом шагу, например, когда безналично оплачивает проезд в транспорте, или снимает (кладёт) деньги в банкомате, или проходит через стойку контроля в учреждении или делает запрос в WWW поисковой системе и т.д. до бесконечности.

Следовательно, с точки зрения проектирования и реализации ИС БД в любой сфере деятельности человеку необходимо в конкретной предметной области располагать какими-либо сведениями, документами, фактами, т.е. иметь определенную **актуальную** информацию в этой области. Другими словами, любая деятельность человека представляет собой процесс сбора и переработки информации, принятия на ее основе решений и их выполнения. Хранить и перерабатывать большой объем информации практически невозможно без специальных средств обработки информации. В качестве таких средств и используют ИС БД.

Практически неограниченные массивы сложно структурированных и связанных между собой данных, хранящиеся в ИС БД, должны быть оптимальным образом организованы для их компьютерного хранения и обработки, обеспечивать распределённое использование данных многими пользователями при условии сохранения и поддержки всех аспектов **целостности данных**.

Цель данного учебно-методического пособия и состоит в том, чтобы сформировать у студентов базовые концептуальные представления об основных принципах проектирования и реализации ИС БД на основе **реляционных баз данных**.

Все это обуславливает большую потребность в квалифицированных кадрах, способных как создавать информационные системы на основе систем управления базами данных, так и обслуживать соответствующие информационные системы и базы данных.

Отражением потребности в специалистах такого рода является включение курса по базам данных в учебный план целого ряда направлений подготовки, например, 02.03.03 Математическое обеспечение и администрирование информационных систем, 02.03.01 Математика и компьютерные науки, 01.05.01 Фундаментальная математика и механика, 01.03.02 Прикладная математика и информатика, 01.05.10 Компьютерная безопасность, 41.03.05 Документоведение и архивоведение. Т.е. данное пособие может быть рекомендовано даже для гуманитарных направлений подготовки.

Важно подчеркнуть, что в пособии излагается «академический» а не «инженерный» подход к изучению технологий реляционных баз данных (РБД).

Например, РБД, согласно теории, рассматривается в трёх аспектах (или на трех уровнях): концептуальный (логический), внутренний (физический) и внешний (уровень представлений). Данный подход позволяет объяснить, например, что СУБД в процессе реализации РБД на физическом уровне на самом деле создают и обрабатывают **реляционные отношения**, а не **таблицы.**

И это несмотря на то, что интерфейс практически всех СУБД использует только термин «Таблица». А это вполне объяснимо, так как интерфейс это и есть внешний уровень РБД, а на уровне представлений реляционные отношения изображаются в виде таблиц. Более того, структуры данных «реляционное отношение» и «таблица» нельзя отождествлять, иначе это приводит к серьезным методологическим проблемам. Например, термин «потенциальный ключ таблицы» с академической точки зрения является абсолютно некорректным.

Данное пособие есть первая часть полного лабораторного практикума и посвящена только основам реляционных баз данных. Учебно-методическое пособие реализовано в формате «Лабораторный практикум». Фундаментальные теоретические и практические аспекты реляционных баз данных рассматриваются на примере СУБД LibreOffice Base.

В первой главе кратко описаны фундаментальные основы реляционных баз данных и ИС БД. Каждая лабораторная работа содержит базовые теоретические сведения по теме работы, описание интерфейса СУБД LibreOffice Base и варианты заданий. Заключительная работа первой части реализована в формате индивидуального задания, в ходе выполнения которого студенты должны продемонстрировать знания, навыки и умения в области проектирования РБД на основе метода семантического моделирования конкретной предметной области и реализации ИС БД в среде СУБД LibreOffice.

1 ФУНДАМЕНТАЛЬНЫЕ ПОНЯТИЯ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ

1.1 Понятие «Информационная система»

Информационной системой (ИС) называется взаимосвязанная совокупность аппаратных средств, математических методов и моделей, программного обеспечения (ПО), персонала и информационных технологий (в том числе, с использованием сетевых технологий), которая используется для создания, хранения и обработки больших массивов информации с целью описания актуального состояния конкретной предметной области.

Такое всеобъемлющее «определение» термина информационные системы позволяет говорить о практически неограниченном множестве типов и параметров классификации ИС. Но нас, в первую очередь, интересуют информационные системы, построенные на основе технологий баз данных (ИС БД).

Базой данных называется сложно структурированная, поименованная и связанная между собой совокупность больших массивов данных, позволяющих описывать актуальное состояние конкретной предметной области [1-2].

Другими словами, база данных, это СТРУКТУРА ДАННЫХ, а не информационная система (как это принято отождествлять даже в учебниках по БД).

База данных - это основа ИС БД, на «фундаменте» которой строится «здание» информационной системы (Рис 1.1), состоящее из прикладных программ, использующие данные БД для описания актуального состояния конкретной предметной области.

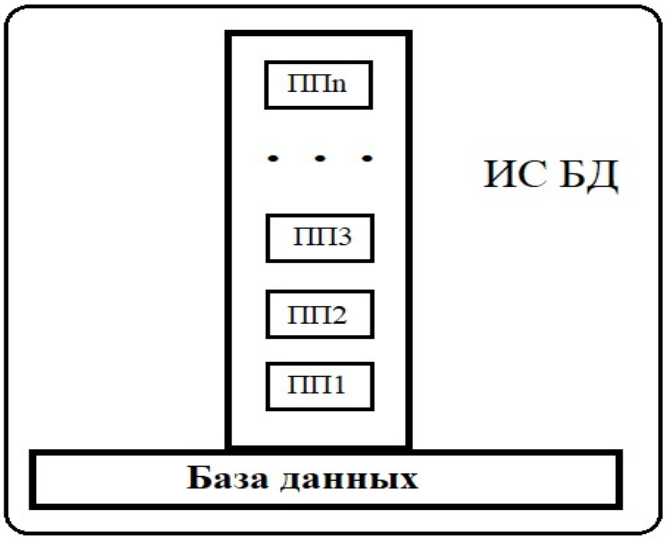


Рис 1.1 «Здание» ИС БД

1.2 Реляционная база данных

Реляционная база данных - это набор реляционных отношений, СВЯЗАННЫХ между собой.

В основе определения структуры данных – РЕЛЯЦИОННОЕ ОТНОШЕНИЕ лежат два фундаментальных постулата реляционной модели данных:

1) Важнейшим элементом реляционной модели является требование простоты используемых в модели данных. Напомним, что к простым типам относят: числовой тип, логический,

символьный, строковый. Однако столь жесткое требование практически не накладывает никаких ограничений на РМД, поскольку это ограничение в манипуляционной части трактуется следующим образом: любая реляционная операция над данными РМД не должна учитывать внутренние структуры данных. При этом одновременно, жестко прописываются правила - операции, которые допускается производить с каждым типом данных. Например, для числового типа допустима операция арифметического сложения, а для строкового типа разрешена операция конкатенации.

2) Понятие домена. Домен является семантическим понятием и с точки зрения РМД рассматривается как подмножество простого типа данных.

Определение домена:

Доменом называется семантическое понятие, которое характеризуется четырьмя фундаментальными свойствами:

1. Домен должен иметь уникальное имя в пределах всей базы данных.
2. Домен определяется либо на простом типе данных, либо на ДРУГОМ ДОМЕНЕ.
3. Домен может иметь ограничения, которые позволяют более конкретно описать подмножество простого типа данных, на котором определен домен.
4. Домен должен иметь семантику.

Определение реляционного отношения (РО):

n-арным реляционным отношением называется **структура данных**, которая определяется через **заголовок отношения** и **тело отношения**.

Заголовок реляционного отношения имеет вид $\mathbf{R(A_1, A_2, \dots, A_n)}$ Где **R** называется именем отношения, $\mathbf{A_i}$ - атрибутами отношения, причем атрибут $\mathbf{A_i}$ определяется как пара вида $\boldsymbol{\{ A_i : D_i \}}$, то есть каждый атрибут отношения должен быть определен на своем домене. Или, другими словами, понятия атрибута и домена практически совпадают, кроме одного тривиального, но системообразующего свойства: имя атрибута должно быть уникально только в пределах данного отношения. Это тривиальное отличие лежит в основе **механизма внешних ключей**.

Тело реляционного отношения есть множество кортежей вида $\mathbf{\{a_1, a_2 \dots, a_n\}}$, где $\mathbf{a_i}$ произвольное значение атрибута $\boldsymbol{A_i \ (a_i = Value(A_i))}$. То есть тело отношения в математическом смысле является декартовым произведением всех допустимых значений атрибутов. Поэтому мощность множества кортежей отношения называют **мощностью отношения**. Число атрибутов в заголовке отношения называют **степенью или арностью** реляционного отношения.

Из определения реляционного отношения следует, что оно как структура данных обладает двумя уникальными и одновременно независимыми фундаментальными свойствами.

1) Заголовок реляционного отношения абсолютно **статичен** и не может быть изменен в течение всего жизненного цикла реляционной базы данных.

2) Напротив, тело отношения абсолютно **динамично**, то есть в любой момент времени ЖЦРБД кортежи могут быть добавлены в отношение, изменены или даже удалены.

Замечание. Эти два свойства структуры данных реляционные отношения фактически и лежат в основе создания ИС БД!

1.3 Потенциальные и внешние ключи РМД

По определению, все кортежи тела отношения уникальны. Однако, легко видеть, что уникальностью могут обладать либо подмножества атрибутов, либо даже отдельно взятый атрибут.

Отсюда следует, что эти уникальные подмножества атрибутов можно и нужно использовать для идентификации кортежей.

Определение потенциального ключа:

Пусть **$R(A_1, A_2, . . . , A_n)$** - произвольное реляционное отношение, **$K = \{ A_i, A_j, ... , A_k \}$** - подмножество атрибутов . Говорят, что подмножество атрибутов **K** является потенциальным ключом, если оно обладает двумя свойствами:

1) **Уникальность**- никакие два картежа отношения **R** не имеют одинаковый набор значений атрибутов из **K**.

2) **Неизбыточность** - никакое подмножество в **K** не обладает свойством уникальности.

Важно подчеркнуть, что реляционное отношение может иметь несколько потенциальных ключей, тогда один из них объявляют первичным ключом, а остальные - альтернативными. Поэтому, если в отношении имеется единственный потенциальный ключ, то термины первичный ключ и потенциальный ключ совпадают.

В РМД между реляционными отношения реализуем только один тип связи - “один-ко-многим” (тип связи “многие-к-одному”, очевидно, является симметричной). Причем связь “один-ко-многим” может быть установлена ТОЛЬКО между родительским и дочерними отношениями.

Определение. Пусть даны два реляционных отношения связных между собой по типу “один-ко-многим”, тогда отношение, которое стоит в связи на стороне «один» называют **родительским**, а отношение, которое стоит на стороне «много», называют **дочерним**.

Для связывания родительского и дочернего отношения используется механизм внешних ключей:

Для того чтобы связать дочернее отношение с родительским в заголовок дочернего отношения достаточно механически добавить атрибуты, определенные на тех же доменах что и атрибуты потенциального ключа родительского отношения.

Определение. Внешним ключом (ВК или FK), называется подмножество атрибутов дочернего отношения, которые определены на тех же доменах, что и атрибуты потенциального ключа соответствующего родительского отношения.

1.4 Класс прикладного программного обеспечения – СУБД

Проблема проектирования и реализации такой сложной структуры данных, как БД и таких сложных информационных систем как ИС БД, оказалась настолько многогранной, что для них был создан специальный класс ПО, получивший название СУБД (система управления базами данных).

Определение СУБД

СУБД это комплекс программных и языковых средств, в состав которых входят инструментальные средства для разработки БД, который предназначен для проектирования и реализации, как БД так и ИС БД и который должен поддерживать следующие основные функции:

1) Проектирование и реализация таких сложно структурированных и связных меж собой данных как БД.

2) Набор функций для ввода, редактирования и удаления практически неограниченного объема сложно структурированных и связанных меж собой данных.

3) Функция долговременного хранения практически неограниченных объемов сложно структурированных и связанных меж собой данных на ВЗУ в виде файлов СУБД;

4) Управление буферами ОП, которые операционная система выделяет процессу СУБД.

5) Поддержка механизма транзакций.

6) Журнализация и восстановление данных БД после аппаратных и программных сбоев.

7) Поддержка языка структурированных запросов Structured Query Language (SQL) и его диалектов для обмена данными БД с оперативной памятью компьютера (Рис. 1.2).

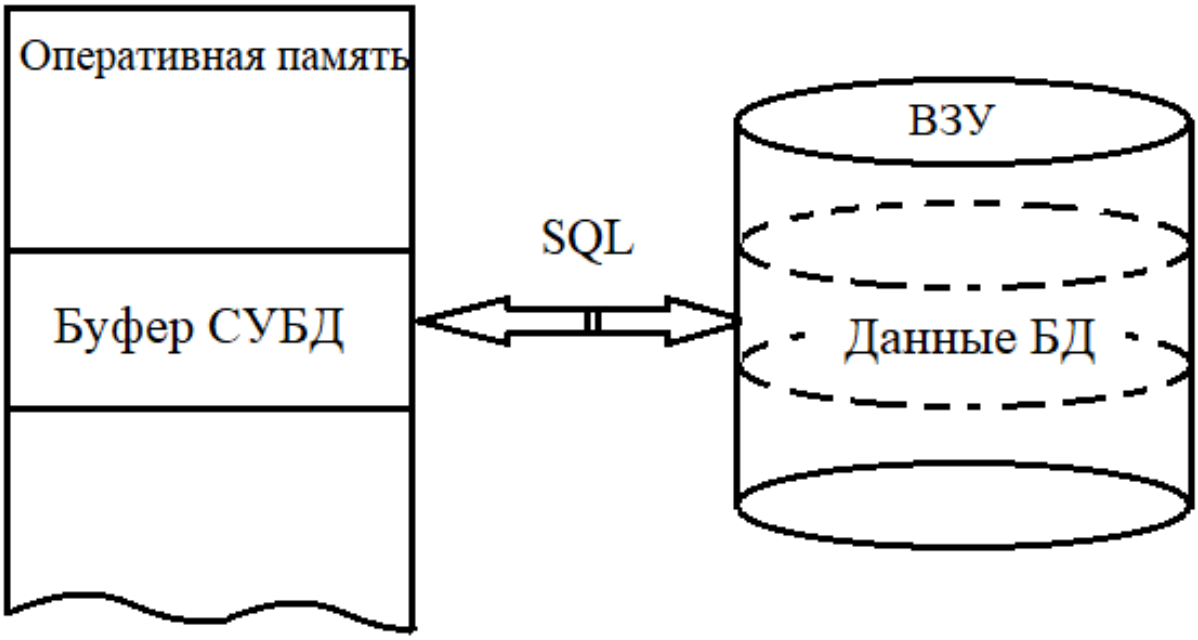


Рис 1.2 Обмен данными БД между ВЗУ и ОП

1.5 Установка СУБД LibreOffice Base

Чтобы установить на свой компьютер СУБД LibreOffice Base, необходимо установить свободно распространяемый, бесплатный офисный пакет LibreOffice, в состав которого традиционно входят такие офисные программы как 1) Текстовый процессор LibreOffice Writer; 2) Электронные таблицы LibreOffice Calc; 3) Электронные презентации LibreOffice Impress; 4) Графический редактор LibreOffice Draw; 5) Локальная СУБД LibreOffice Base.

Дистрибутив офисного пакета LibreOffice рекомендуется скачивать с официального сайта [4].

После скачивания и запуска инсталлятора, выберите «Настройка», если хотите отказаться от ненужных языков интерфейса и прочих компонентов офисного пакета. Если выберете «Типичный», установится LibreOffice с компонентами по умолчанию и языковым настройками, соответствующими параметрам ОС (Рис. 1.3).

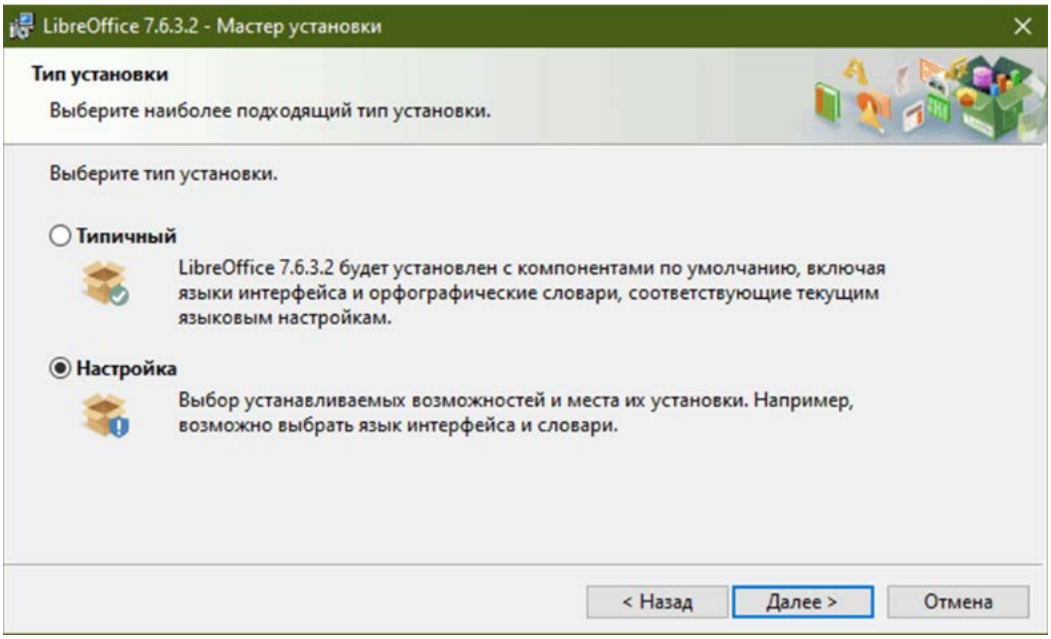


Рис 1.3 Мастер установки LibreOffice

Если выбрали первый вариант, разверните разделы и выберите состояние ненужных составляющих как «Компонент будет недоступен» (Рис. 1.4).

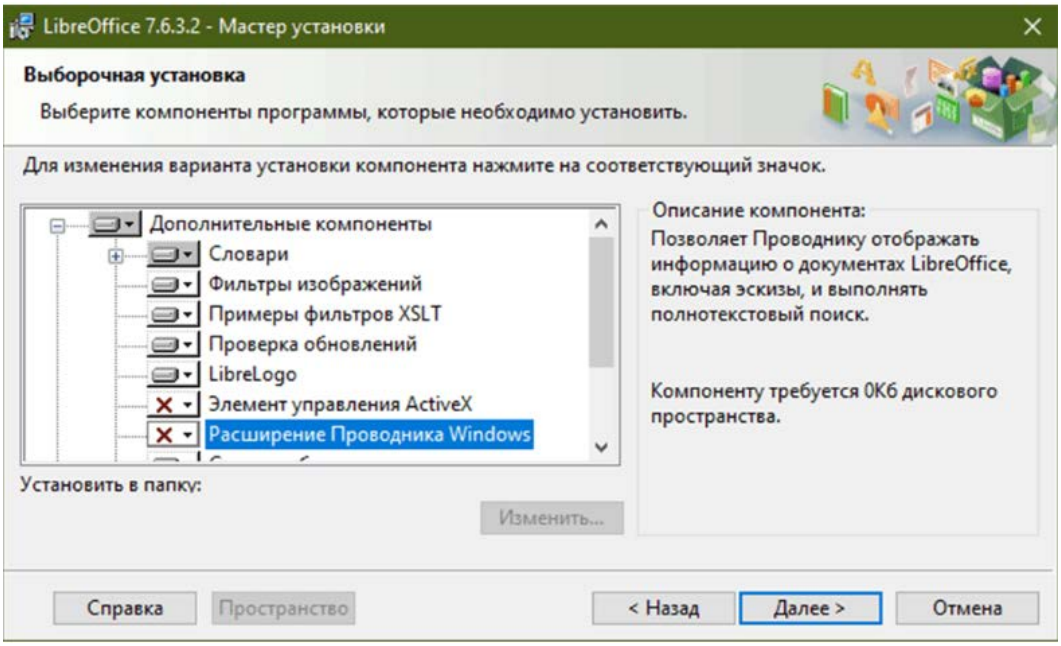


Рис 1.4 Выборочная установка LibreOffice

Важно! Для работы LibreOffice Base нужна среда Java SE (Java Standard Edition) – стандартная платформа для разработки десктопных прикладных программ, в частности ИС БД.

Если на вашем компьютере Java SE ещё не установлена, то последнюю версию дистрибутива нужно скачать с сайта https://www.java.com/ru/download/. Запустите скачанный файл, примите условия лицензионного соглашения и нажмите «Install».

2 СОЗДАНИЕ БД КАК СЛОЖНО СТРУКТУРИРОВАННЫХ И СВЯЗАННЫХ МЕЖДУ СОБОЙ ДАННЫХ

Целью лабораторной работы № 1 является изучение интерфейса, назначения и возможностей СУБД LibreOffice Base, в среде которой и будет создаваться реляционная база данных «Деканат». Предполагается, что студенты знакомы с основными понятиями теории реляционных баз данных [1-2].

2.1 Учебная база данных «Деканат»

В качестве учебной базы данных Части 1 учебного пособия рассмотрим БД «Деканат». Для упрощения задачи, предположим, что в искомой предметной области мы выделили только такие основные сущности как «Преподаватель», «Кафедра», «Должность», «Учёная степень», «Студент», «Специальность», «Дисциплина», «Успеваемость». Предполагается, что на кафедре обязательно должен быть сотрудник с должностью «Заведующий кафедрой». А каждый преподаватель в качестве параметров - атрибутов имеет «Учёную степень» и «Учёное звание». Каждый преподаватель имеет уникальный табельный номер, зависящий от кафедры, на которой он работает, и (неуникальную) должность. Каждый студент идентифицируется уникальным номером зачетной книжки, который зависит от года поступления и номера специальности, на которой он обучается. Также для простоты, разрешим в качестве атрибутов Фамилия, Имя, Отчество (как у преподавателей, так и у студентов) использовать один атрибут ФИО.

В результате искомая модель базы данных будет включать в себя следующие основные сущности:

1) Преподаватель (Таб_номер, ФИО, Кафедра, Должность, Учёная степень, Учёное звание).
2) Кафедра (ID, Наименование, Аудитория).
3) Должность (ID, Наименование, Оклад).
4) Учёная степень (ID, Наименование).
5) Студент (Номер_зачетки, ФИО, Специальность, Дата_рождения, Пол).
6) Специальность (Код_специальности, Наименование).
7) Дисциплина (ID, Название, Семестр).
8) Успеваемость (Оценка, Номер_зачетки, Дисциплина, Преподаватель).

Для простоты, «справочную» сущность «Учёное звание» в модели будем игнорировать. Тем более, что в ВУЗе, как правило, присутствуют только два экземпляра этой сущности: «Доцент» и «Профессор», а учёные звания «Член-корреспондент» и «Академик» в преподавательской среде встречаются редко.

2.2 Создание базы данных

Чтобы создать базу данных в LibreOffice Base, можно либо воспользоваться мастером создания базы данных, где выбрать пункт «Создать новую базу данных» (Рис 2.1). По щелчку на кнопке «Далее» переходите к следующему диалоговому окну, в котором вы можете зарегистрировать базу данных, открыть ее в режиме редактирования и сразу же начать создание реляционных отношений - таблиц. После проведенных манипуляций вам будет предложено сохранить базу данных, после чего можно приступать к созданию таблиц.

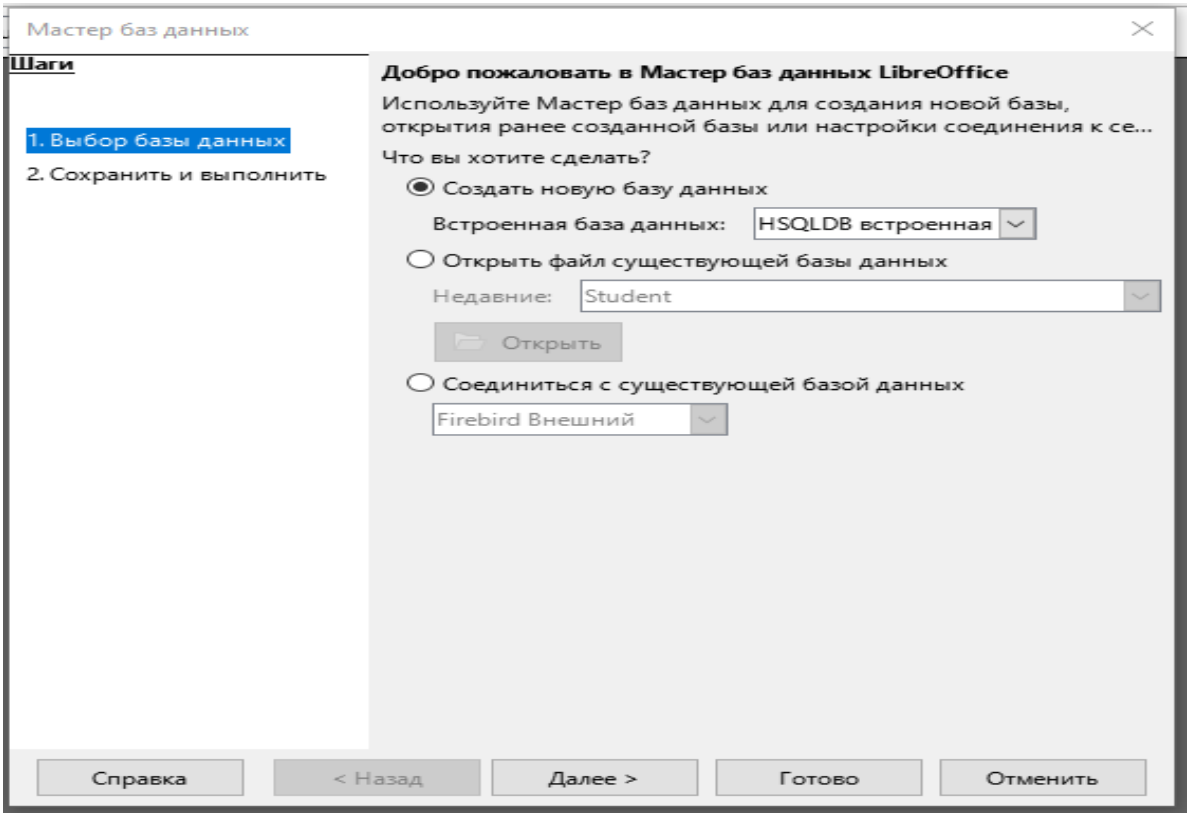


Рис 2.1 Мастер базы данных

Внимание! Здесь и далее ВСЕ объекты, которые будут создаваться при выполнении лабораторных работ, необходимо именовать с суффиксом _FIO. Например, если ФИО студента «Луканов Александр Сергеевич», то создаваемая им база данных должна иметь имя «Деканат_LAS»! Лабораторные работы, не удовлетворяющие данному требованию, преподавателем к отчету не принимаются и не засчитываются!

2.3 Создание реляционных отношений – таблиц базы данных «Деканат»

Замечание 1. Так как на уровне представлений в реляционной модели данных реляционные отношения отображаются в виде таблиц, то все СУБД в интерфейсе вместо термина «реляционное отношение» используют термин «таблица». Поэтому здесь и далее вместо термина «реляционное отношение» будет использоваться термин «таблица».

В левой части окна LibreOffice Base, меню «База данных» позволяет переключаться между режимами работы с отношениями - таблицами, запросами, формами и отчетами (Рис 2.2).

На данном этапе выбираем режим работы с таблицами. В окне «Задачи» необходимо выбрать один из пунктов: 1) Создать таблицу в режиме дизайна; 2) Использовать мастер для создания таблицы и 3) Создать представление. Для начала рекомендуется выбрать пункт «Создать таблицу в режиме дизайна».

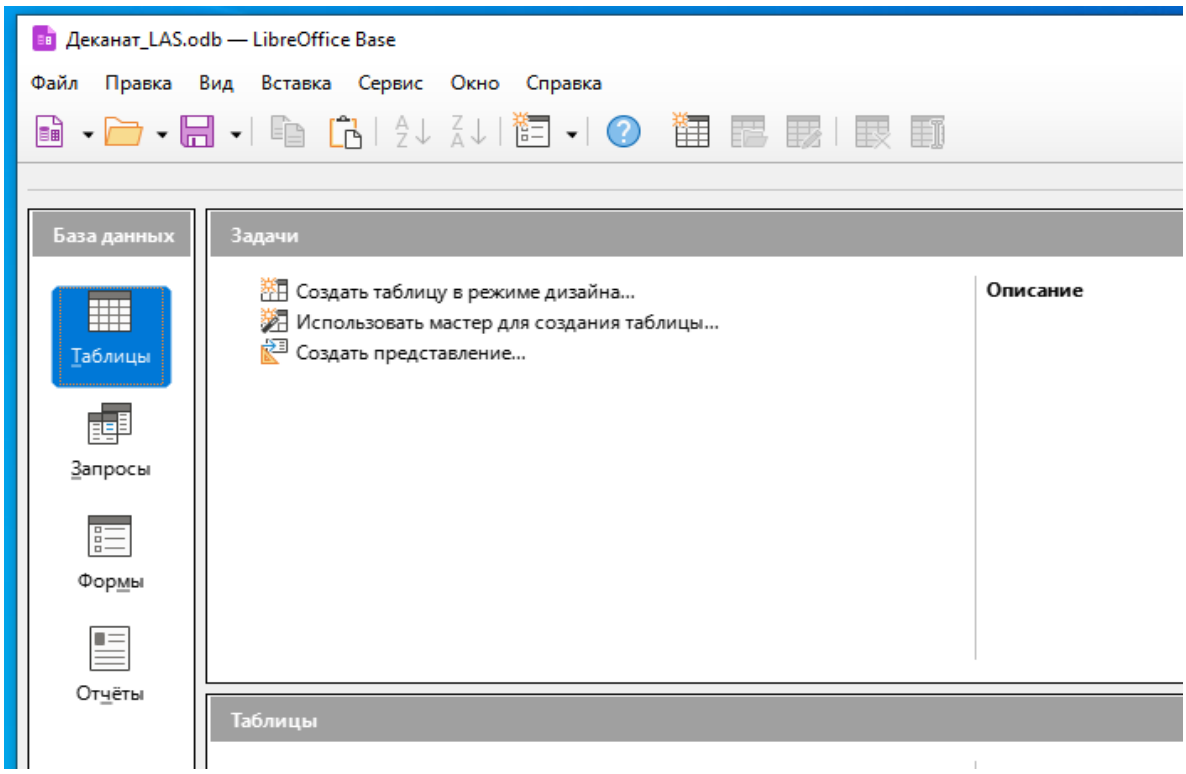


Рис 2.2 Работа с таблицами

При выборе пункта меню «Создать таблицу в режиме дизайна», в новом окне открывается дизайн для создания таблиц. Шаблон «Таблица» включает в себя три столбца: «Имя поля», «Тип поля» и «Описание». Первые два столбца являются обязательными для заполнения, так как «Имя поля» однозначно соответствует атрибуту в заголовке реляционного отношения, а «Тип поля» фактически описывает домен, на котором должен быть определено данное поле таблицы – реляционного отношения (все свойства домена, кроме семантики).

Замечание 2. Столбец «Описание» формально при создании таблицы можно не заполнять, но здесь и далее везде будем использовать этот столбец для ОПИСАНИЯ ДОМЕНА, на котором определено данное поле – атрибут таблицы! Шаблон описания домена должен иметь имеет вид ***<Имя домена\Простой тип\Ограничения\Семантика>***

Если семантика атрибута очевидна (например, «Наименование кафедры»), то её можно не заполнять.

Например, создадим таблицу «Преподаватель» (Рис 2.3).

Замечание 3. Для атрибута «Таб_номер» нельзя использовать простой тип «Числовой», так как номер кафедры может лежать и в диапазоне 1–9.

Замечание 4. По умолчанию для простого типа VARCHAR длина поля установлена равной 100. Необходимо обязательно редактировать поле «Значение по умолчанию» устанавливая «разумную» длину.

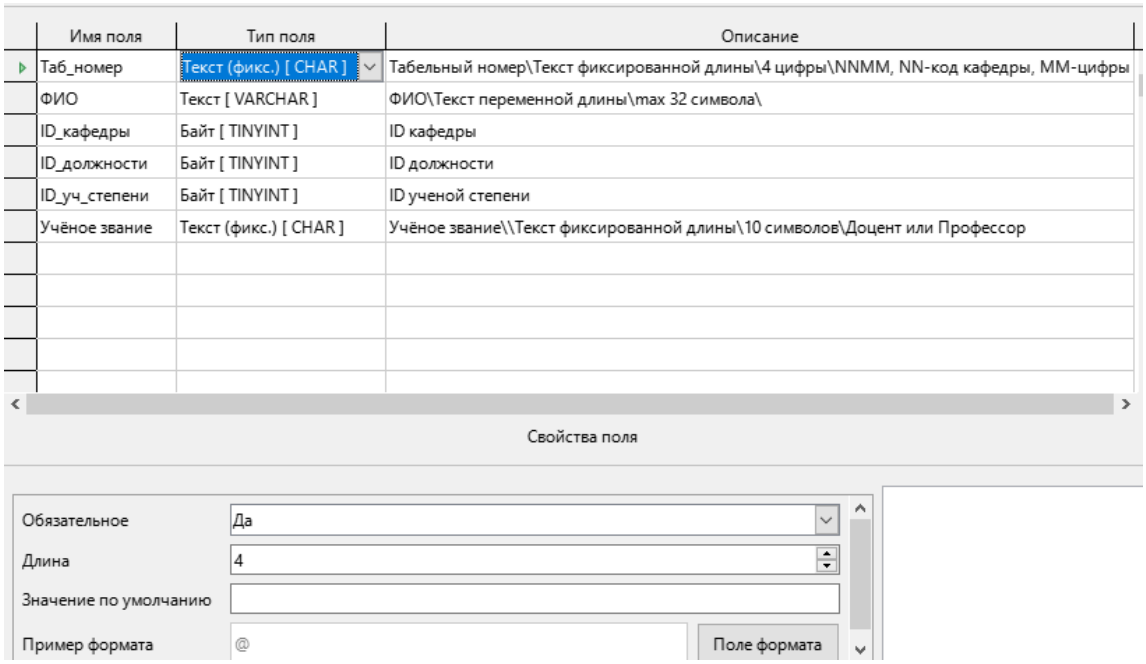


Рис 2.3 Создание таблицы «Преподаватель»

Замечание 5. Для полей – атрибутов таблицы, которые являются ВНЕШНИМИ КЛЮЧАМИ необходимо указывать только ИМЯ ДОМЕНА, на котором определён потенциальный ключ соответствующего родительского отношения. Описание всех 4 свойств домена считается грубой ошибкой.

Замечание 6. LibreOffice Base не допускает сохранение и именование таблиц без указания ПЕРВИЧНОГО КЛЮЧА! Поэтому перед сохранением таблицы выберите искомое поле и, вызвав контекстное меню, выберите «Первичный ключ» (Рис 2.4). В дочерней таблице-отношении первичный ключ изначально может отсутствовать. Тогда поле с именем ID и простым типом TINYINT создается дополнительно к фактическим полям.

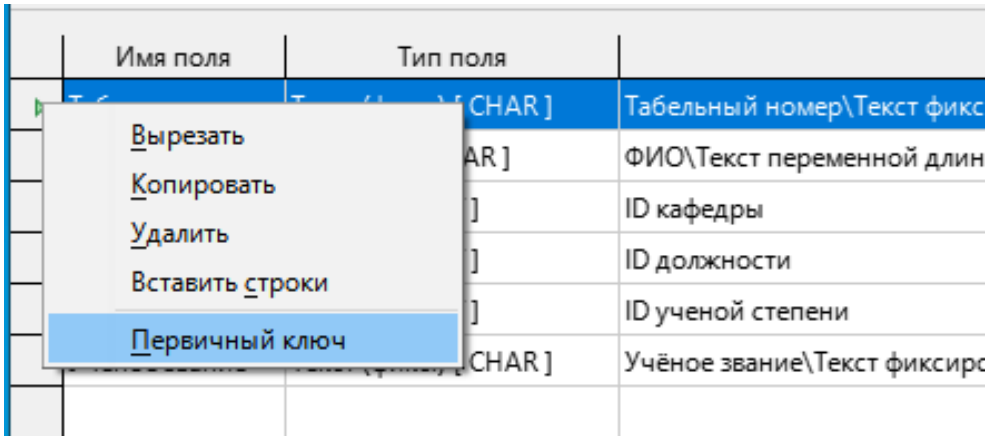


Рис 2.4 Задание первичного ключа

Замечание 7. В любой момент создания базы данных её таблицы можно редактировать. Для этого в окне «Таблицы» нужно вызвать контекстное меню и выбрать «Правка» (Рис 2.5).

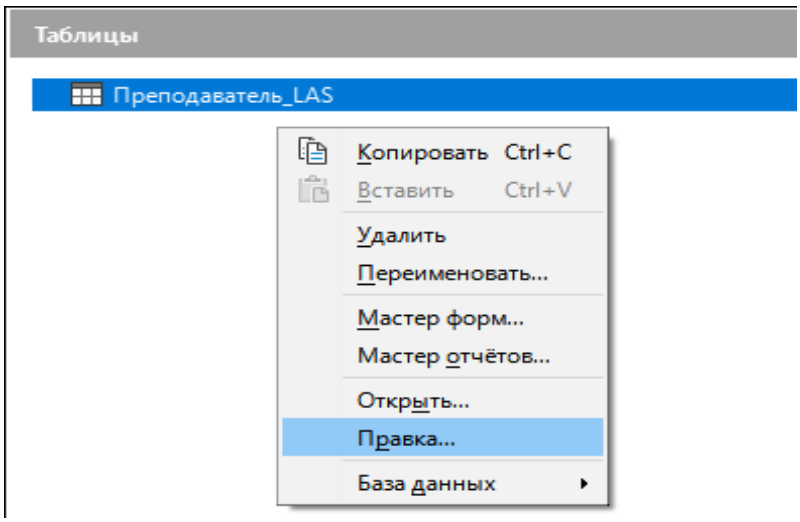


Рис 2.5 Вызов редактирования таблицы

Аналогично создаём и корректно сохраняем таблицы «Кафедра», «Должность», «Учёная степень», «Студент», «Специальность», «Дисциплина» и «Успеваемость».

Замечание 8. Поле «Оклад» в таблице «Должность» необходимо определять на простом типе NUMERIC (10,2) и задать формат «Денежный». Для чего в окне «Свойства поля» в сроке «Пример формата» раскрыть окно «Поле формата» и выбрать категорию «Денежный».

Замечание 9. Поле «Номер_зачетки» в таблице «Студент» необходимо определять на простом типе «CHAR(10)», так как по условию, семантика домена этого атрибута зависит от года поступления студента, кода специальности, на которой он обучается и порядкового номера студента в группе, т.е. первые две цифры слева это последние две цифры года поступления, затем 6 цифр – код специальности и последние две цифры – порядковый номер студента в группе. Например, если студент поступил на первый курс в 2021 году на специальность 02.03.03 - Математическое обеспечение и администрирование информационных систем, то его номер зачетки может принимать ТОЛЬКО такие значения «21020303nn», где nn лежит в диапазоне 01-99.

2.4 Заполнение таблиц

Заполнение таблиц данными и их возможное редактирование в любой реляционной СУБД является рутинной операцией ничем не отличающейся от аналогичных операций в других разделах информационных технологий. Дважды щелкните на ярлыке (значке) искомой таблицы в окне «Таблицы», откроется новое окно заполняемой таблицы с именем «Данные таблицы» (Рис 2.6).

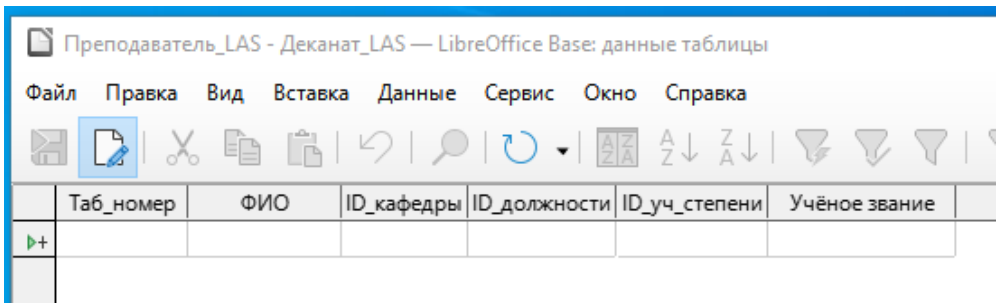


Рис 2.6 Интерфейс для заполнения таблицы

Пример заполнения таблиц приведен на Рис 2.7

Таб_номер	ФИО	ID_кафедры	ID_должности	ID_уч_степени	Учёное звание
0101	Степанов А.Н	1	5	1	Профессор
0102	Сироченко В.П.	1	6	2	Доцент
0103	Луканов А.С.	1	6	2	Доцент
0105	Морозова Е.В.	1	8	10	-
0115	Бабич А.С	1	10	10	-
0205	Иванов Д.В.	2	6	2	Доцент
0211	Осипов М.Н.	2	5	2	Доцент
0808	Щепакина Е.А.	8	5	1	Профессор
0810	Воропаева Н.В.	8	2	2	Доцент
0812	Китаева Е.В.	8	6	2	Доцент
1105	Панов А.Н.	11	5	1	Профессор
1305	Асташкин С.В.	13	5	1	Профессор
1501	Степанова Л.В.	15	5	1	Доцент
1820	Салеев В.А.	18	5	1	Профессор

Рис 2.7 Пример заполнения таблицы «Преподаватель»

Замечание 10. При заполнении таблиц базы данных «Деканат» настоятельно рекомендуется использовать корректные для механико-математического факультета данные. Для этого достаточно использовать официальные данные, взятые, например, по адресу https://ssau.ru/info/struct/op/faculties/mmf/department

*Замечание 11. При вводе данных **минимальное** число кортежей-строк для родительских отношений-таблиц равно **8** (восьми), для дочерних – **14** (четырнадцати)! Данное требование необходимо для корректного выполнения последующих лабораторных работ.*

Замечание 12. Самое главное, что при вводе данных, необходимо поддерживать целостность (корректность и согласованность) данных. Например, лаборант никак не может занимать должность заведующего кафедрой! Но больше всего проблем возникает при сохранении целостности внешних ключей. Поэтому сначала рекомендуется заполнять родительские отношения-таблицы и только затем – дочерние.

Замечание 13. Для примера заполнения таблицы «Преподаватель» (Рис 2.7) были использованы следующие данные «справочных» родительских отношений-таблиц:

Таблица «Кафедра»

ID	Название	Аудитория
1	Информатики и вычислительной математики	303 м/м
2	Безопасности информационных систем	411 м/м
8	Дифференциальных уравнений и теории управления	403 м/м
11	Алгебры и геометрии	406 м/м
13	Функционального анализа и теории функций	410 м/м
15	Математического моделирования в механике	414 м/м
18	Общей и теоретической физики	301 физ
22	Физвоспитания	Корпус 6

Таблица «Должность»

ID	Наименование	Оклад
1	Декан	80 000,00 ₽
2	Зам. декана	60 000,00 ₽
4	Ассистент	32 000,00 ₽
5	Заведующий кафедрой	70 000,00 ₽
6	Доцент	53 500,00 ₽
7	Преподаватель	36 000,00 ₽
8	Старший преподаватель	40 000,00 ₽
9	Лаборант	15 400,00 ₽
10	Педагог-организатор	24 000,00 ₽
11	Методист кафедры	22 000,00 ₽
45	Лаборант-Исследователь	18 000,00 ₽

Таблица «Ученая степень»

ID	Наименование
1	Доктор физико-математических наук
2	Кандидат физико-математических наук
3	Доктор технических наук
4	Кандидат технических наук
5	Доктор философских наук
6	Кандидат философских наук
7	Кандидат исторических наук
8	Доктор исторических наук
10	Без ученой степени

Внимание! После выполнения рассмотренных выше упражнений, в качестве задания лабораторной работы необходимо:

1) Индивидуально добавить ещё 2-3 кафедры

2) Заполнить таблицы так, чтобы на ВСЕХ кафедрах работало НЕ МЕНЕЕ 4 сотрудников.

2.5 Установка связей между таблицами–отношениями

По определению реляционная БД это набор реляционных отношений СВЯЗАННЫХ между собой в ЕДИНУЮ логическую структуру данных.

Для установления связей между таблицами достаточно в главном меню выбрать пункт «Сервис» и кликнуть «Связи» (Рис 2.8).

В результате откроется «рабочий стол» окна «Дизайн связей». Из вложенного окна «Добавить таблицы» необходимо по очереди все созданные таблицы добавить на рабочий стол дизайна связей.

Для связывания родительских таблиц-отношений с дочерними достаточно воспользоваться технологией «drag and drop». Данный механизм реализуется путём «захвата» (нажатием и удержанием левой кнопки мыши) потенциального ключа «родительской» таблицы и перемещении с последующей фиксацией курсора на соответствующий внешний ключ целевой «дочерней» таблицы.

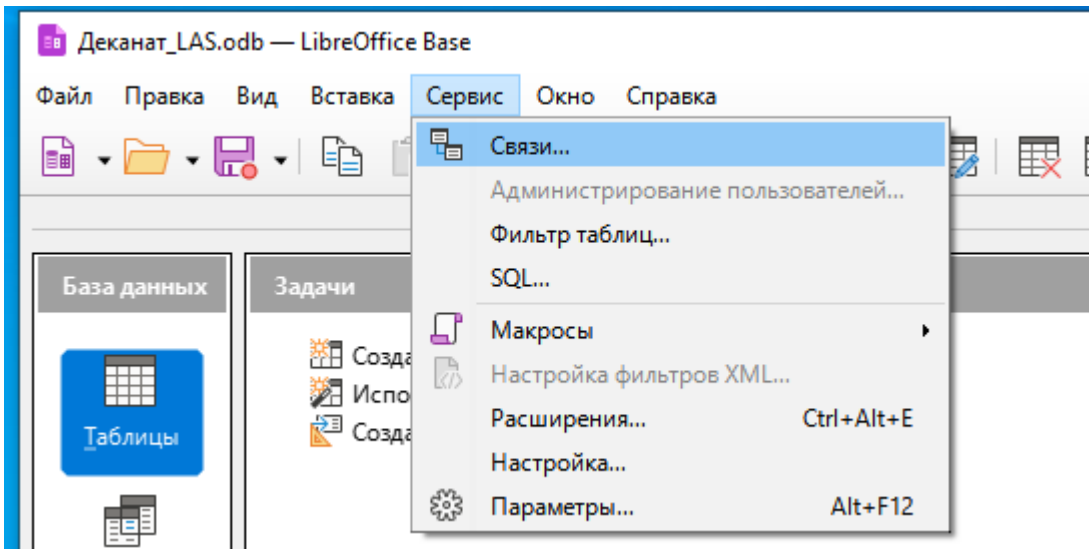


Рис 2.8 Установление связей между таблицами

Когда клавиша мыши будет отпущена, появится новая связь. Данную манипуляцию необходимо повторить для установления всех связей между таблицами – отношениями базы данных «Деканат». В результате будет получена следующая ER-диаграмма базы данных «Деканат» (Рис 1.9)

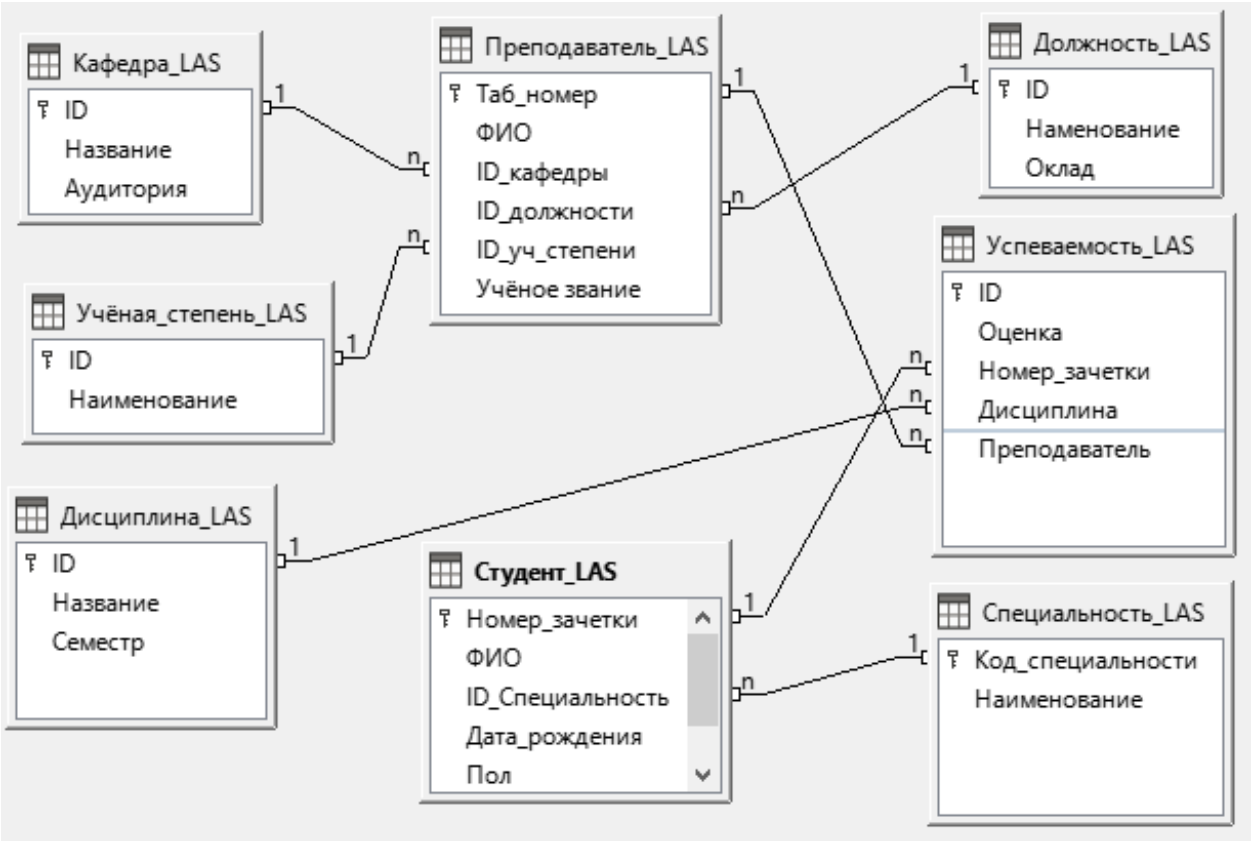


Рис 2.9 Схема связей в базе данных «Деканат»

Созданные связи можно редактировать. Для этого достаточно «выделить» курсором искомую связь и вызвать контекстное меню редактирования связи (Рис 2.10)

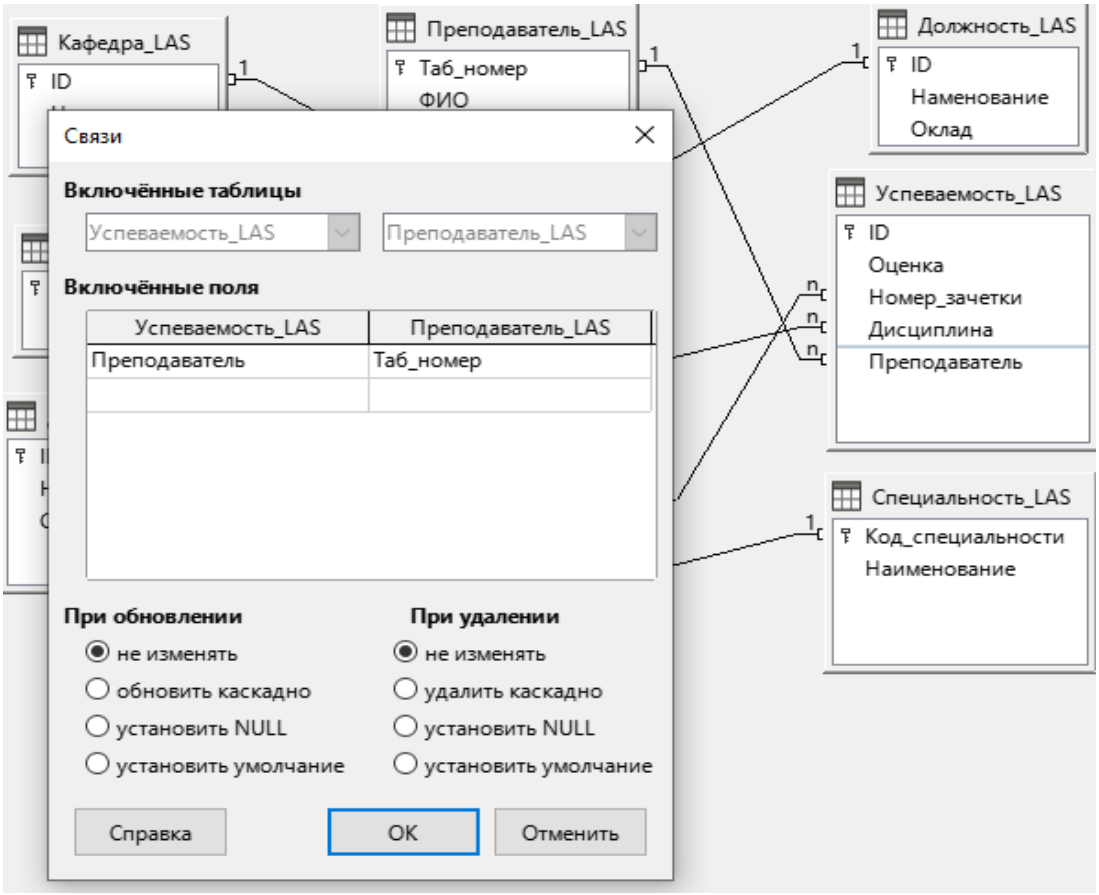


Рис 2.10 Редактирование связи «Преподаватель» - «Успеваемость»

Окно «Связи» подчеркивает, что все современные СУБД реализуют основные стратегии «поддержания ссылочной целостности». По умолчанию установлена стратегия RESTRICT (Не изменять). Поэтому для режима редактирования данных в «РОДИТЕЛЬСКИХ» таблицах рекомендуется установить стратегию CASCADE.

Замечание 14. Стратегию «Удалить каскадно» в учебной базе данных пока рекомендуется не устанавливать.

2.6 Вопросы к собеседованию при сдаче лабораторной работы № 1

1. Дайте определение информационной системы.
2. Что означает термин «ИС БД».
3. Что такое «фундамент» ИС БД?
4. Что такое «здание» ИС БД?
5. Приведите примеры «локальных» (офисных) СУБД.
6. Дайте определение домена.
7. Опишите домен «Табельный номер» преподавателя.
8. Опишите домен «Номер зачетки».
9. Чем атрибут реляционного отношения отличается от домена?
10. Дайте определение реляционного отношения.
11. Что такое потенциальный ключ реляционного отношения?
12. Что такое внешний ключ реляционного отношения?

13. Какой тип связи реализует механизм внешних ключей?
14. Сравните структуры данных «Реляционное отношение» и «Таблица».
15. Почему в интерфейсе реляционных СУБД используется термин «Таблица»?
16. Как добавить таблицу в схему данных?
17. В чем отличие между таблицей в режиме конструктора и таблицей в режиме заполнения?
18. Как сделать поле таблицы ключевым?
19. Что произойдет, если ни одно поле таблицы пользователь не сделал ключевым?
20. Опишите, как установить и изменить связи между таблицами.
21. Что такое ссылочная целостность?
22. Что такое каскадное обновление и каскадное удаление? Приведите пример.
23. Можно ли добавлять/удалять произвольные данные в родительскую таблицу? В дочернюю?

3 СОЗДАНИЕ ФОРМ. МАНИПУЛИРОВАНИЕ ДАННЫМИ БД С ПОМОЩЬЮ ФОРМ

В лабораторной работе № 1 был заложен «фундамент» будущей информационной системы «Деканат». Теперь можно приступать к построению «здания» ИС БД. В классе ПО «Офисные СУБД» простыми элементами информационной системы являются формы, запросы и отчеты. Лабораторная работа № 2 и посвящена созданию и работе с формами, как базовыми элементами интерфейса ИС БД.

3.1 Создание табличных и «линейных» форм

Меню «База данных» позволяет сразу переключиться на работу с формами (Рис 3.1)

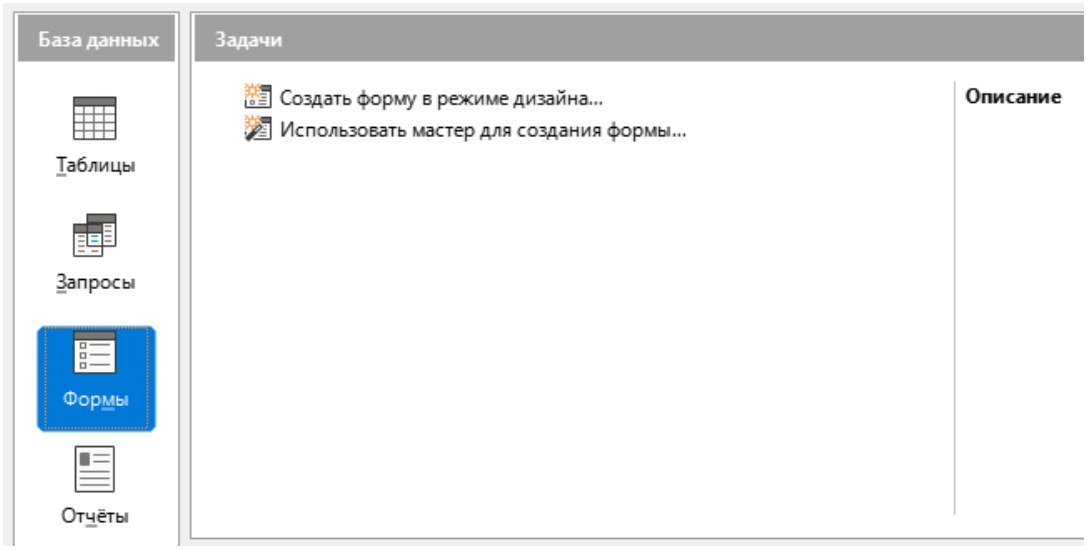


Рис 3.1 Работа с формами

Для создания простейших форм достаточно воспользоваться услугами мастера форм, а затем отредактировать сделанную им заготовку.

Окно «Мастер форм» (Рис 3.2) позволяет сделать вывод о том, что ПОЛНЫЙ цикл создания формы включает в себя восемь шагов:

1) Выбор полей искомой таблицы;
2) Настройка субформы (если она необходима;
3) Добавление полей субформы;
4) Настройка полей связывания субформ;
5) Расположение элементов формы;
6) Задание источников данных, которые будут отображаться в форме;
7) Выбор стиля формы (цвета, границы);
8) Сохранение объекта ИС БД - форма.

Но на самом деле, этих шагов девять! Так как на **нулевом шаге** необходимо выбрать таблицу или запрос, для которой и будет строиться форма!

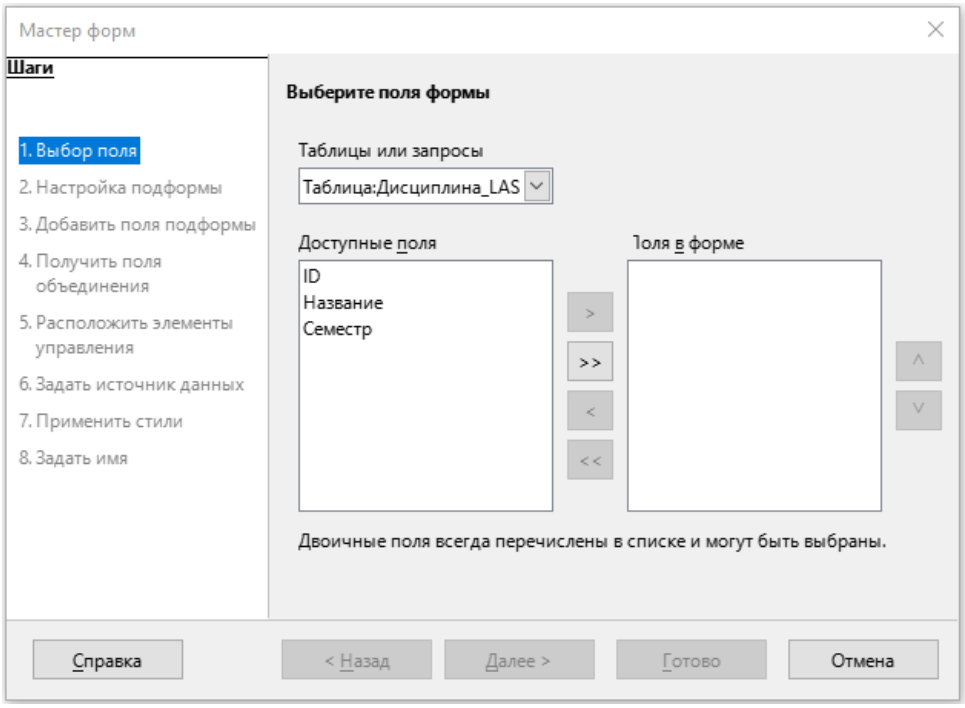


Рис 3.2 Интерфейс окна «Мастер форм»

Очевидно, что субформы (подформы как в интерфейсе) могут понадобиться для построения форм, на которых отображаются данные двух или более связанных между собой таблиц. Поэтому, для начала, создадим форму для «родительской» таблицы, в которой нет внешних ключей. Например, для таблицы «Дисциплина».

1) Итак, выбрав таблицу «Дисциплина» из «выпадающего» списка активных таблиц и запросов, переносим все поля таблицы в окно «Поля формы» нажав кнопку «>>» (Рис 3.3). Кнопка «>» позволяет выбирать поля индивидуально.

Замечание 1. Если на первом шаге нажать кнопку «Готово», то сразу будет создана простейшая форма, имеющий табличный формат.

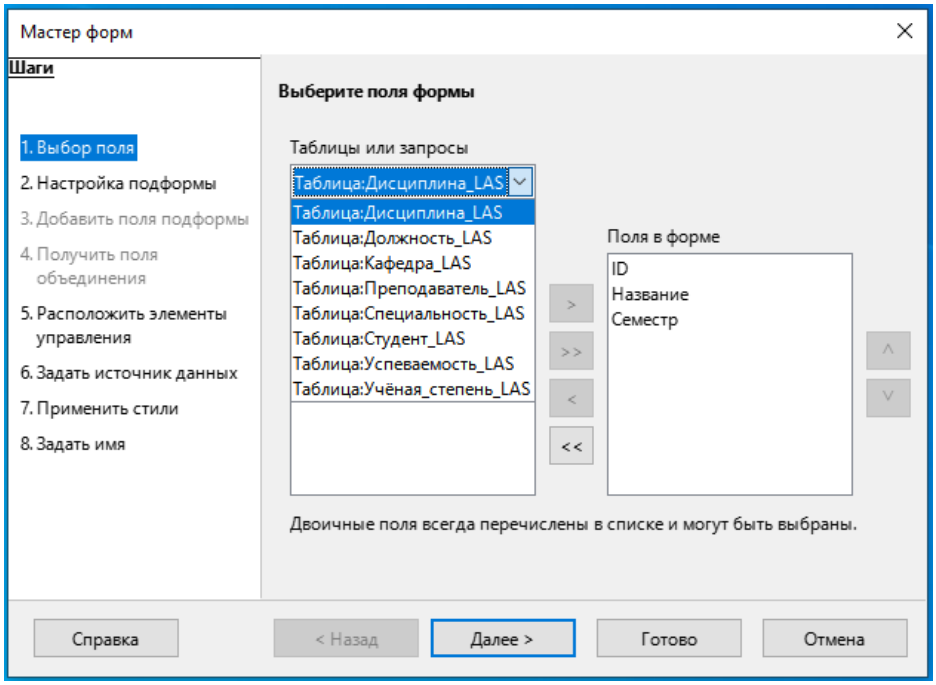


Рис 3.3 Перенос полей таблицы в окно «Поля в форме»

2) Так как субформы нам не понадобятся сразу переходим на шаг 5. (Рис 3.4)

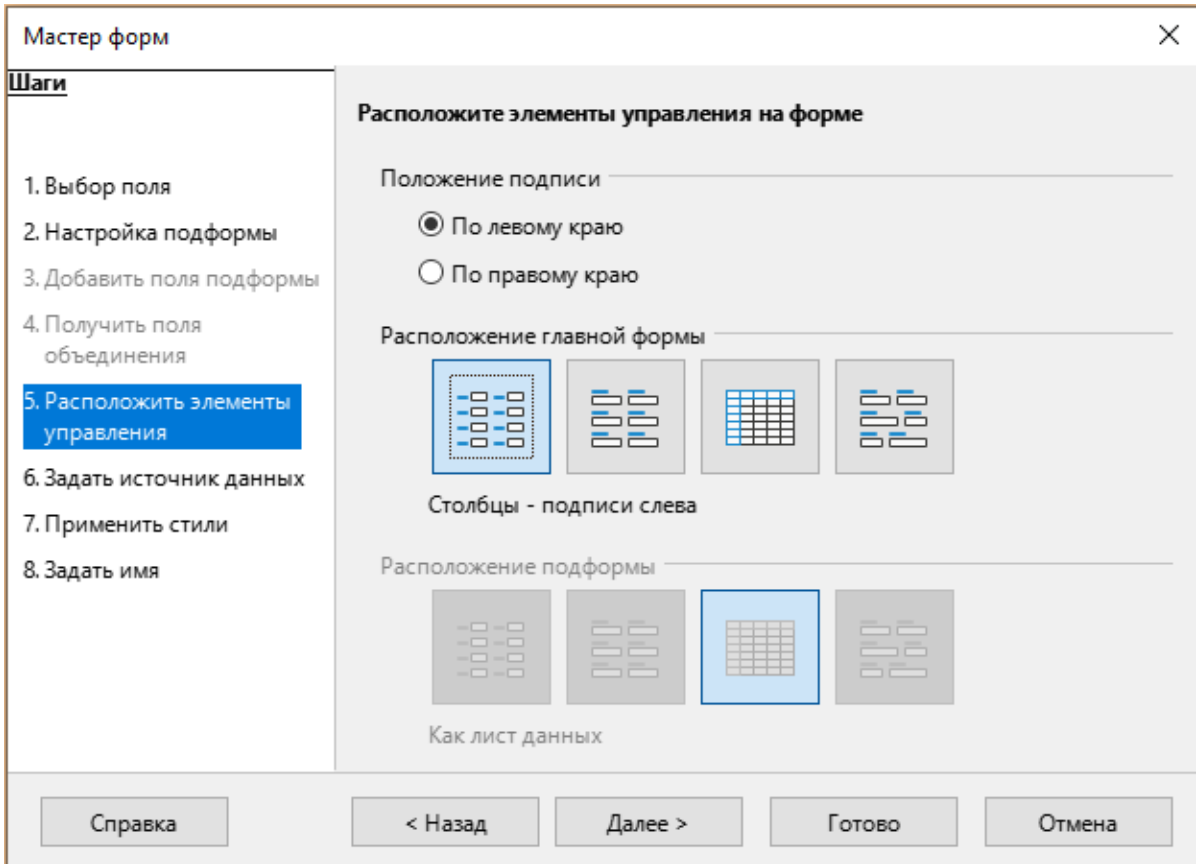


Рис 3.4 Выбор расположения элементов в форме

Возможны четыре варианта: а) «Столбцы – подписи слева»; б) «Столбцы – подписи сверху»; в) «Как лист данных»; г) «Блоки – подписи сверху».

Формы по типу а), б) и г) в дальнейшем будем называть «линейными», а по типу в) – «табличными».

Внимание! Так базовыми функциями любой ИС БД являются а) Ввод и редактирование данных; б) Поиск данных; в) Просмотр данных, то необходимо создавать интерфейс ИС БД, которой будет максимально эффективно реализовывать эти функции. Для простоты будем реализовывать и объединять функции б) и в) в ТАБЛИЧНЫХ формах, а функцию а) только в ЛИНЕЙНЫХ формах.

Отсюда следует, что для выполнения лабораторной работы необходимо, для начала, создать минимум 10 форм по две на каждую «справочную» таблицу БД «Деканат» – одну линейную для ввода и редактирования данных таблицы и вторую – табличную для просмотра данных таблицы.

Например, табличная форма для таблицы «Дисциплина» будет иметь вид, как на Рис 3.5



Рис 3.5 «Табличная» форма для просмотра и поиска данных

А один из трех вариантов «линейной» формы для таблицы «Дисциплина» будет иметь вид, как на Рис 3.6

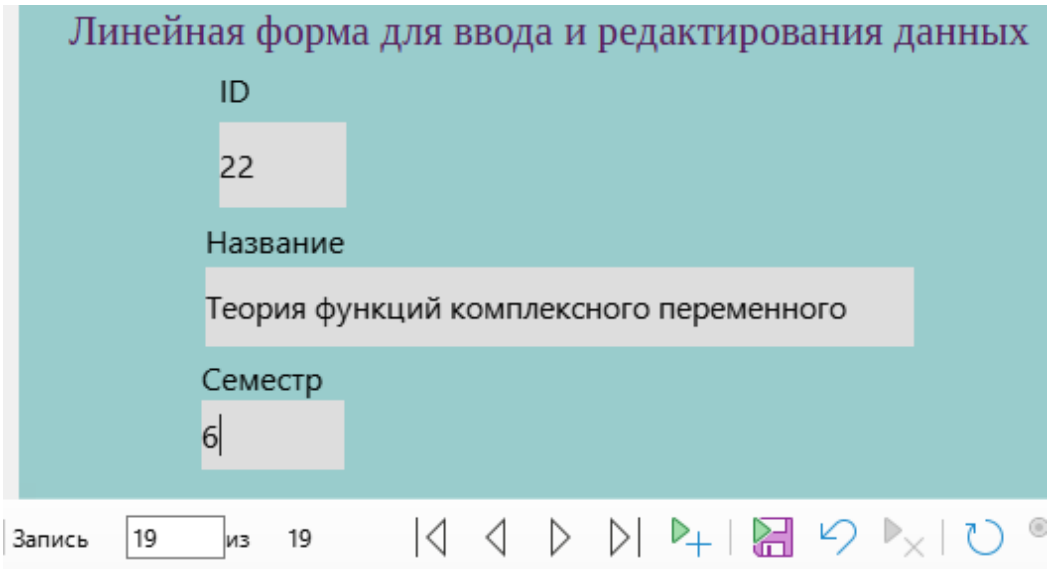


Рис 3.6 «Линейная» форма для ввода и редактирования данных

Очевидно, что в «табличных» формах необходимо запретить ввод, редактирование и удаление данных. Поэтому на шаге 6 необходимо запретить эти операции (Рис 3.7)

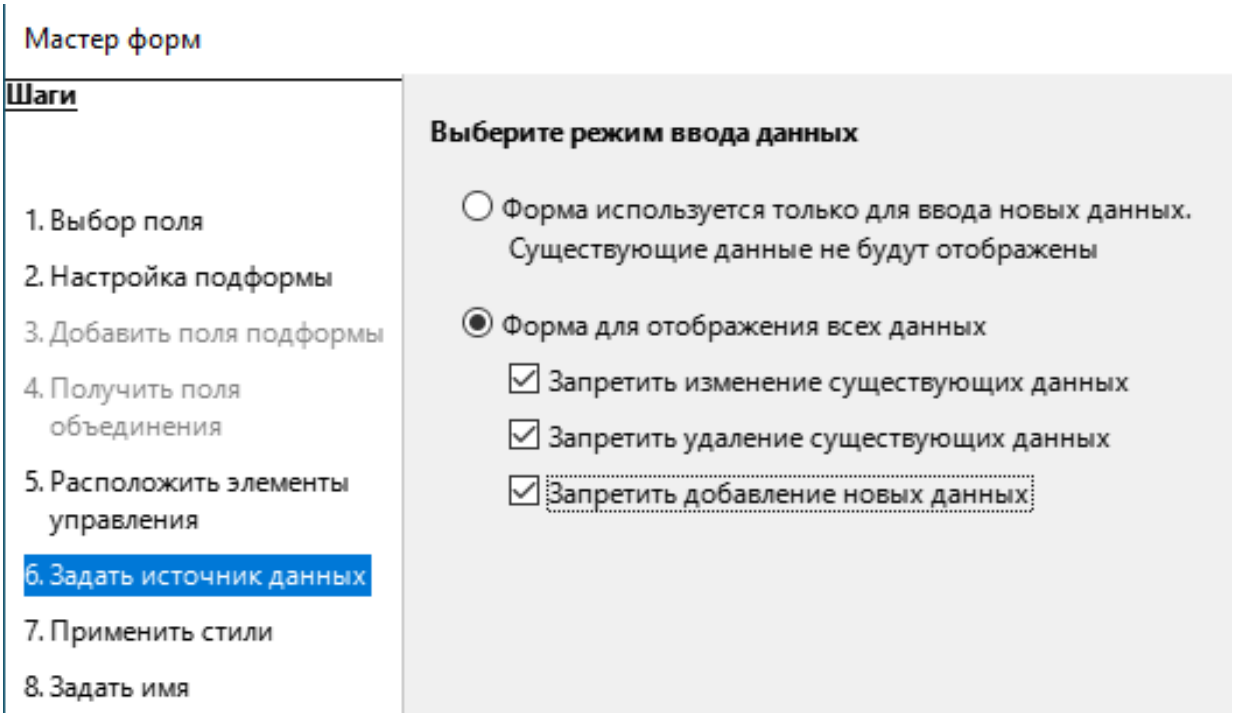


Рис 3.7 Выбор режима для работы с данными

А для «линейных» форм, наоборот разрешить эти операции (т.е. «снять галочки»).

Наконец, существует возможность создавать формы, предназначенные только для ввода новых данных. Для этого необходимо выбрать режим «Форма используется только для ввода новых данных. Существующие данные не будут отображены».

3.2 Усложнение дизайна форм

Даже в учебных базах данных, не говоря уже о профессиональных, большинство таблиц-отношений в качестве данных таблицы содержат коды (ID) этих данных. В результате линейные формы, построенные для таких таблиц, «скрывают» значения этих данных.

Поэтому создадим форму с улучшенным дизайном, которая будет отображать не только коды данных, но и их значения. Для примера возьмем таблицу «Студент».

На первом этапе создадим линейную форму для таблицы «Студент», последовательно повторив шаги п.п. 3.1.

Затем, вызвав контекстное меню в окне «Формы» для формы «Студент», выберем «Правка» (Рис 3.8). В результате откроется окно «Форма базы» данных в режиме дизайна!

В режиме дизайна (кнопка ) становятся активной вертикальная и горизонтальная панели инструментов.

Вертикальная панель позволяет добавлять на рабочий стол формы множество объектов различного «содержательного» типа. Например, «Текстовое поле», «Флажки», «Кнопки», «Переключатели», «Списки», «Поля со списком» и т.д.

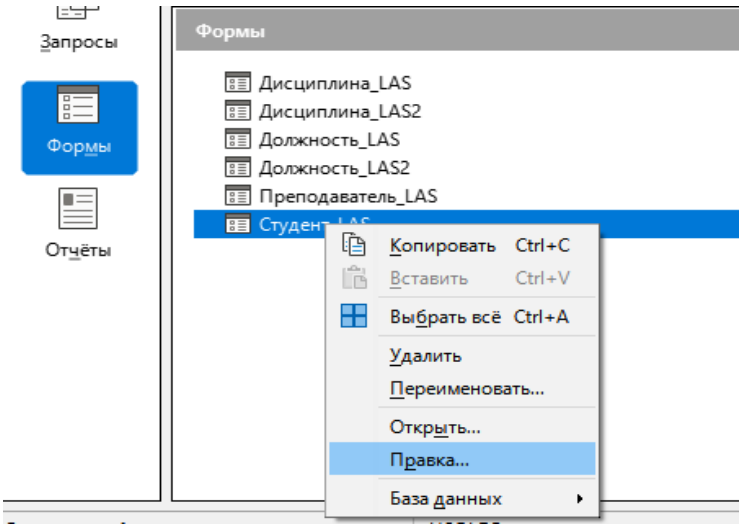


Рис 3.8 Выбор режима «Правка» для редактирования формы

Горизонтальная панель позволяет добавлять на рабочий стол формы множество графических объектов различного типа. Причем эта панель практически идентична панели инструментов векторного графического редактора.

Принцип работы с объектами, размещенными на форме, абсолютно идентичен работе с объектами в офисных средах: объекты можно перемещать, изменять размеры и, главное, изменять параметры (шрифт, цвет, координаты, реакция на событие и т.д.).

Из доступных объектов для решения исходной задачи - отобразить названия специальностей подходят «Список». Щелкните по компоненту «Список» мышкой и переместите ее на рабочий стол формы. Указатель мыши примет форму компонента с плюсом в верхнем левом углу. Теперь растяните компонент в области формы – тут же появится диалоговое окно мастера создания полей со списком (Рис 3.9).

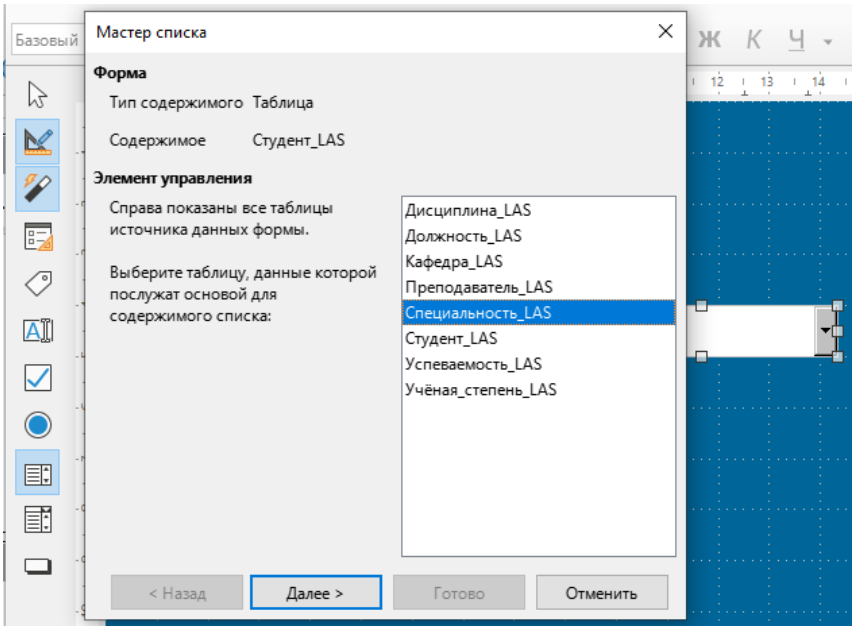


Рис 3.9 Окно «Мастер списка»

В этом окне нужно выбрать таблицу, на основе которой будут переданы данные в исходную форму. Поэтому выбираем таблицу «Специальность» и нажимаем на кнопку «Далее».

На следующем шаге требуется выбрать поля, которые станут значениями поля со списком. Выбираем поле «Наименование» и нажимаем кнопку «Далее» (Рис 3.10).

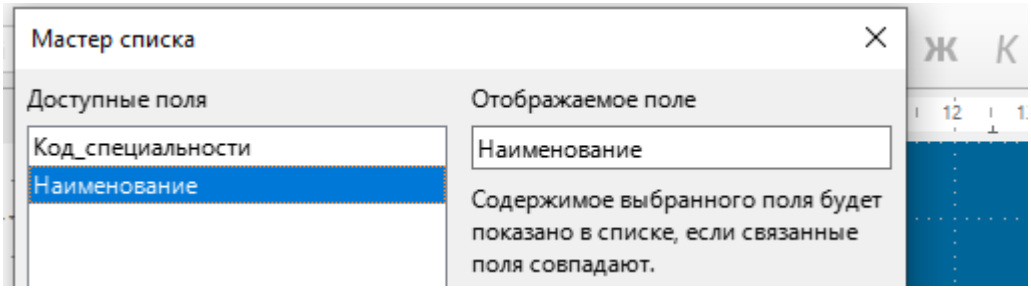


Рис 3.10 Выбор отображаемого поля

На следующем шаге необходимо связать данные «родительской» таблицы «Специальность» с таблицей исходной формы «Студент» (Рис 3.11).

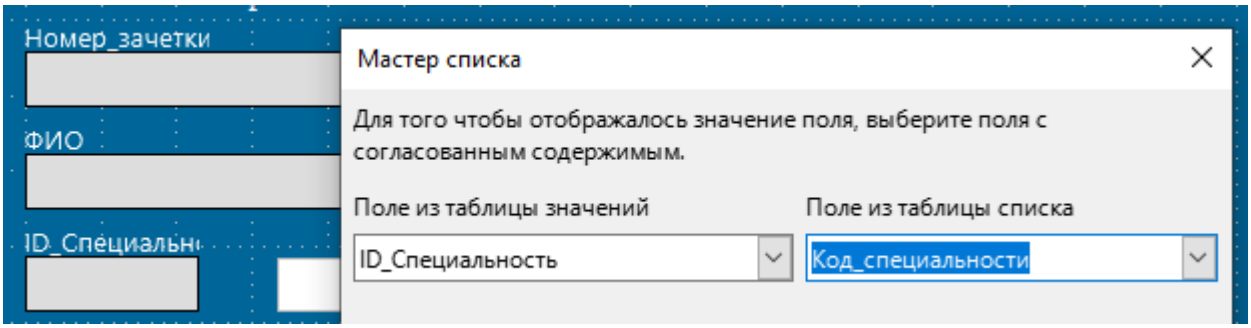


Рис 3.11 Выбор отображаемого поля

Сохраняем построенную форму и открываем её (Рис 3.12).

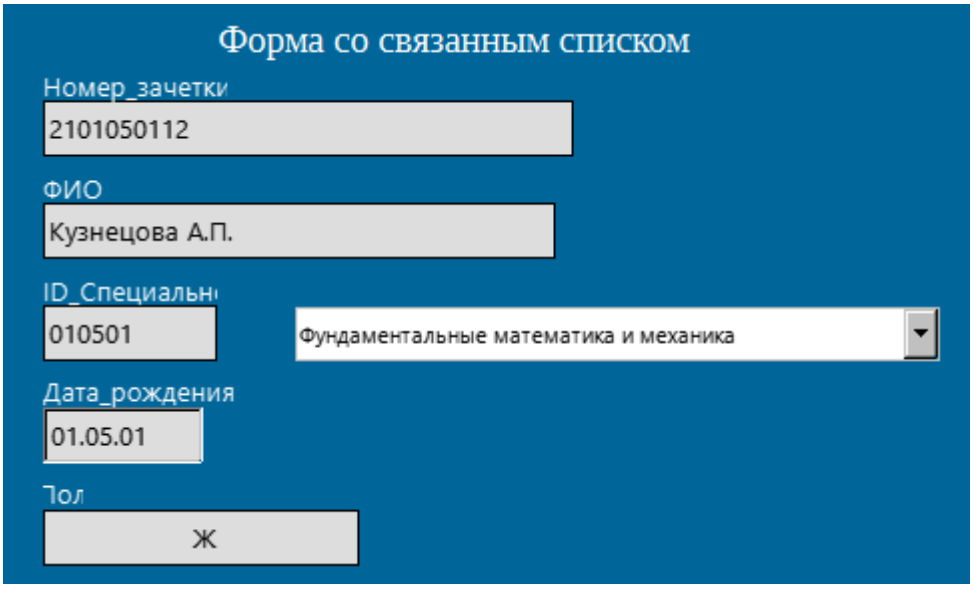


Рис 3.12 Пример формы со связанным списком

Внимание! После выполнения рассмотренных выше упражнений, в качестве задания лабораторной работы необходимо для ИС БД «Деканат» самостоятельно создать 2–3 формы со связанными списками.

3.3 Использование субформ

Построенные в п.п. 2.2 формы со связанными списками фактически позволяют отображать данные, связанные по типу «один ко многим». Для отображения данных связанные по типу «многие ко многим» при построении форм необходимо использовать субформы. Например, рассмотрим вариант одновременного отображения данных таблиц «Успеваемость», «Преподаватель», «Дисциплина» и «Студент» (которые, в совокупности, находятся в связи «многие ко многим») на одной форме.

Для этого реализуем следующий алгоритм (шаги которого подробно рассмотрены в п.п. 3.1 и 3.2):

1) Открываем окно «Мастер форм» и выбираем таблицу «Преподаватель».

2) В окно «Поля в форме» переносим только два поля «Таб_номер» и «ФИО».

3) На втором шаге мастера форм выбираем «Добавить субформу». Проверяем, чтобы по умолчанию была активна радиокнопка «Субформа на основе выбранных полей».

4) На третьем шаге мастера форм выбираем таблицу «Успеваемость» и переносим в окно «Поля в форме» все поля кроме «ID».

5) На четвертом шаге мастера форм настраиваем объединение форм: в первом поле объединения субформ выбираем «Преподаватель», а в первом поле объединения главной формы выбираем «Таб_номер».

6) На пятом шаге мастера форм выбираем типы главной формы и субформы. Здесь рекомендуем оставить табличный тип, но при выполнении самостоятельных заданий можно поэкспериментировать с линейными формами.

7) Так как создаваемая форма служит для просмотра связанных между собой данных, то на шестом шаге мастера запрещаем изменение и удаление существующих данных и добавление новых данных.

8) На седьмом шаге мастера форм выбираем стиль формы и обрамление полей.

9) Сохраняем созданную форму.

10) Открываем созданную форму в режиме редактирования. (это можно сделать и сразу на шаге сохранения формы).

11) В таблице «Успеваемость» столбцы «Номер_зачетки», «Дисциплина» и «Преподаватель» делаем «невидимыми». Для этого для каждого столбца вызываем контекстное меню и выбираем пункт «Скрыть столбец».

12) На рабочий стол формы добавляем объект «Список» для вывода списка дисциплин, по которым получены экзаменационные оценки. Для этого в окне «Мастер списка» выбираем таблицу «Дисциплина», в качестве отображаемого поля выбираем «Название» и связываем объекты по полям «Дисциплина» и «ID».

13) Аналогично, объект «Список» для вывода списка студентов, которые сдавали экзамены по данным дисциплинам. Для этого в окне «Мастер списка» выбираем таблицу «Студент», в качестве отображаемого поля выбираем «ФИО» и связываем объекты по полям «Номер_зачетки» и «Номер_зачетки».

14) Сохраняем окончательную версию формы.

В результате будет создана форма, один из дизайнерских вариантов которой представлен на Рис 3.13

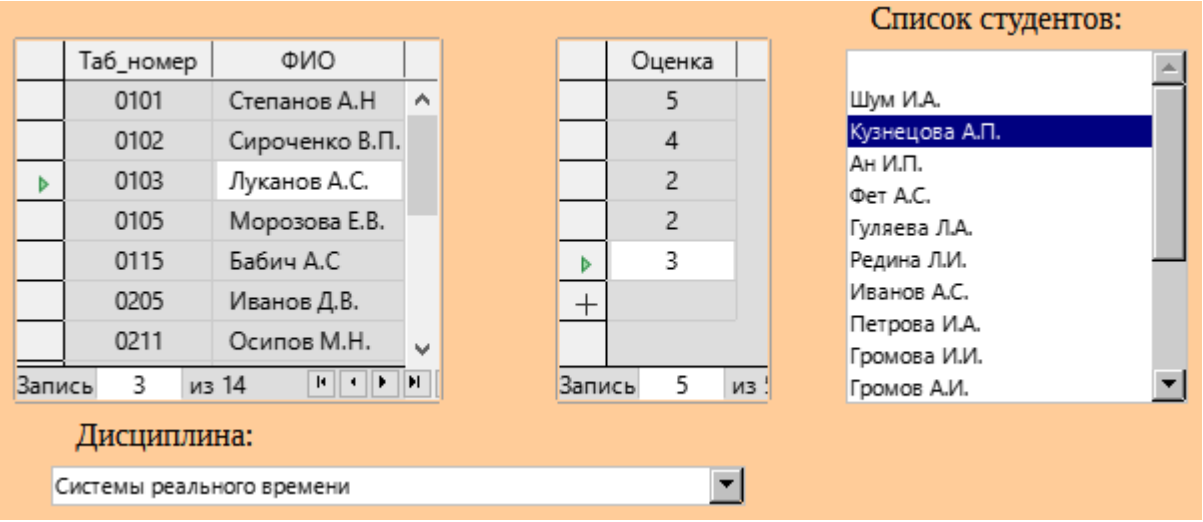


Рис 3.13 Пример формы на основе субформы

Из рисунка видно, что студент Кузнецова А.П. сдавала экзамен преподавателю Луканову А.С. и получила оценку «удовлетворительно» по дисциплине «Системы реального времени».

Внимание! После выполнения рассмотренных выше упражнений, в качестве задания лабораторной работы необходимо для ИС БД «Деканат» самостоятельно создать 1–2 формы на основе субформ.

Замечание 2. Если не на каждом шаге создание форм, то периодически рекомендуется проверять работоспособность формы, а затем вновь возвращаться к редактированию.

Замечание 3. Очевидно, много времени потребуется на разработку дизайна форм. При этом особое внимание следует обратить на расположение элементов формы, их размеры, табличный или линейный тип, а также цвета и размеры шрифтов.

3.4 Вопросы к собеседованию при сдаче лабораторной работы № 2

1. Почему формы являются элементом информационной системы ИС БД?
2. Что такое режим дизайна при создании форм?
3. Что такое мастер для создания форм?
4. Сколько шагов составляет полный цикл мастера и форм?
5. Когда мастер форм использует сокращенный, а не полный цикл?
6. Почему для каждой таблицы рекомендуется создавать минимум 2 формы?
7. Что такое «линейная» форма?
8. Что такое «табличная» форма?
9. Как в форме запретить изменения и удаления существующих данных и добавление новых данных?
10. Какие панели инструментов используют мастер форм в режиме дизайна?
11. Назовите основные объекты, которые позволяет добавлять мастер форм.
12. Как отредактировать форму?
13. Как на рабочий стол формы добавить объект список?
14. Как установить связь между объектом типа список и исходной таблицей?
15. Для чего используются субформы?
16. Почему в субформах не рекомендуется ввод и редактирование данных?

4 СОЗДАНИЕ ЗАПРОСОВ. ЗАПРОСЫ КАК ЭЛЕМЕНТ ИС БД

Лабораторная работа №3 посвящена созданию запросов, которые так же, как и формы позволяют манипулировать данными БД, но в другом аспекте. Если формы, как правило, предназначены для просмотра, ввода и редактирования данных, то запросы позволяют мгновенно найти не только найти искомые данные, но и обработать их. Т.е. запросы также являются неотъемлемой частью ИС БД, как и формы.

4.1 Создание простых запросов

Для создания запросов в меню базы данных необходимо выбрать пункт «Запросы». LibreOffice Base позволяет создавать запросы в 3 режимах: 1) Создать запрос в режиме дизайна, 2) Использовать мастер для создания запроса и 3) Создать запрос в режиме SQL. (Рис 4.1)

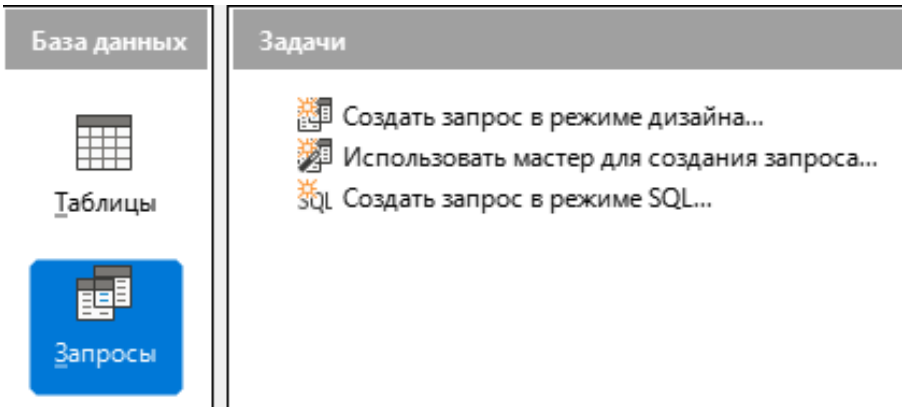


Рис 4.1 Меню для создания запросов

К простому типу запросов, как правило, относят запросы, «движком» которых является оператор SQL SELECT. Поэтому такие запросы и называют выборкой. Другими словами, при создании запроса - выборки достаточно указать целевую таблицу (или представление) и задать условия выборки.

Например, построим запрос для вывода ФИО всех заведующих кафедр всех кафедр.

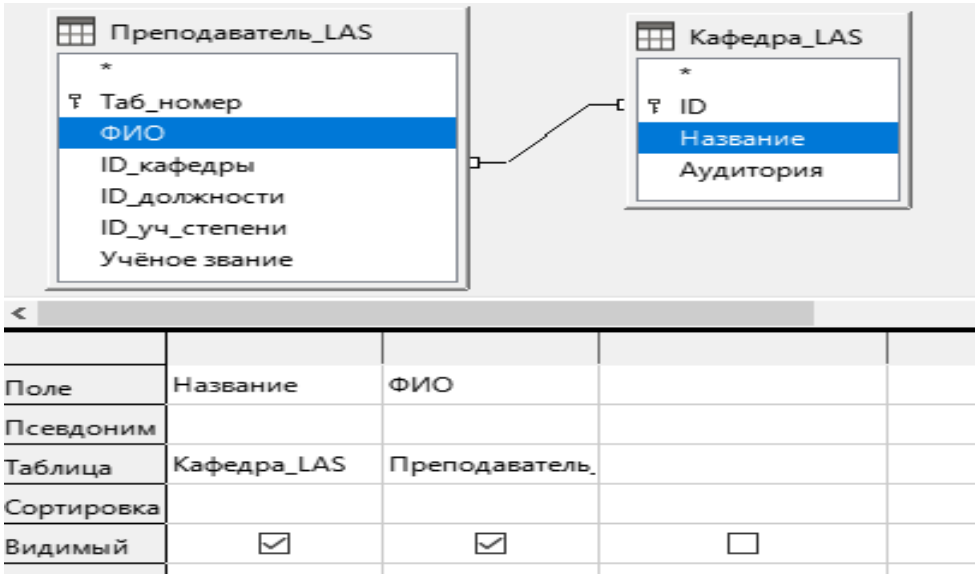


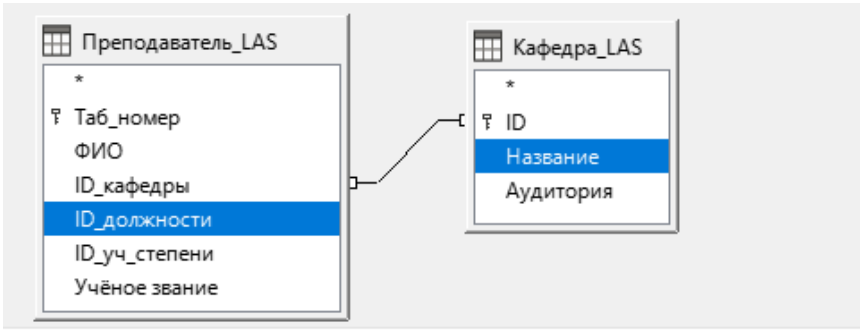
Рис 4.2 Исходные данные для запроса

Выбираем пункт «Создать запрос в режиме дизайна». В окне «Добавить таблицу или запрос» по очереди добавляем таблицы «Преподаватель» и «Кафедра».

В графу «Поле» первой колонки поместим поле «Название» (для этого достаточно сделать двойной щелчок мышью по соответствующему полю в таблице «Кафедра»), в графу «Поле» второй колонки поместим поле «ФИО» (Рис 4.2).

Заполним псевдонимы столбцов как «Название кафедры» и «Заведующий кафедрой» соответственно.

Наконец, и это самое главное, зададим условие выборки. Известно, что код должности «Заведующий кафедрой» равен 5. Поэтому в графу «Поле» третьей колонки поместим поле «ID_должности», делаем столбец «невидимым» (снимаем «галочку») и в строке «Критерий» столбца «ID_должности» задаем условие выборки: « = 5 ». Запрос готов (Рис 4.3).



Поле	Название	ФИО	ID_должности
Псевдоним	Название кафедры	Завдующий кафедрой	
Таблица	Кафедра_LAS	Преподаватель_LAS	Преподаватель
Сортировка			
Видимый	☑	☑	☐
Функция			
Критерий			= 5

Рис 4.3 Готовый запрос

Сохраняем запрос, например, с именем «Зав_кафедрами» и запускаем его. Создается окно с данными, которые удовлетворяют условию запроса (Рис 4.4).

Название кафедры	Заведующий кафедрой
Информатики и вычислительной математики	Степанов А.Н
Безопасности информационных систем	Осипов М.Н.
Дифференциальных уравнений и теории управления	Щепакина Е.А.
Алгебры и геометрии	Панов А.Н.
Функционального анализа и теории функций	Асташкин С.В.
Математического моделирования в механике	Степанова Л.В.
Общей и теоретической физики	Салеев В.А.

Рис 4.4 Результат выполнения запроса

Внимание! После выполнения рассмотренных выше упражнений, в качестве задания лабораторной работы необходимо для ИС БД «Деканат» самостоятельно создать 2–3 запроса-выборки для данных минимум двух таблиц, связанных между собой.

Замечание 1. Просмотреть результаты запроса, в режиме редактирования, можно с помощью кнопки «Выполнить запрос»  *на панели инструментов. Окно «Дизайн запроса» при этом» будет отображать три блока, что очень удобно для отладки запросов.*

Замечание 2. Если нажать кнопку «Режим дизайна вкл./выкл.» *, то открывается окно автоматически созданного запроса на языке SQL. Например, для запроса «Зав_кафедрами» оператор SELECT имеет вид (Рис 4.5)*

```
SELECT "Кафедра_LAS"."Название" AS "Название кафедры",
"Преподаватель_LAS"."ФИО" AS "Заведующий кафедрой" FROM "Преподаватель_LAS",
"Кафедра_LAS" WHERE "Преподаватель_LAS"."ID_кафедры" = "Кафедра_LAS"."ID" AND
"Преподаватель_LAS"."ID_должности" = 5
```

Рис 4.5 Оператор SELECT для запроса «Зав_кафедрами»

Замечание 3. На основе созданных запросов очень удобно создавать новые формы, причем как линейные, так и на основе субформ, так как табличный формат отображения данных стандартного запроса не всегда удобен.

Внимание! В качестве задания лабораторной работы необходимо для ИС БД «Деканат» самостоятельно создать 2-3 формы на основе созданных ранее запросов.

4.2 Запросы с параметрами

Запросы, рассмотренные в п.п. 4.1, в некотором смысле «статичны». Чтобы изменить условие выборки этих запросов, нужно редактировать их непосредственно. Это не всегда удобно. LibreOffice Base позволяет создавать и использовать запросы, в которых можно динамически изменять параметры условия выборки. Например, построим запрос, который позволяет сразу узнать размер зарплаты какого-то конкретного преподавателя.

Чтобы создать запрос с параметром, реализуем сначала обычный запрос-выборку на основе следующих данных: ФИО преподавателя, его должность и оклад. Т.е. запрос будет строится на основе таблиц «Преподаватель» и «Должность».

Затем в колонке «ФИО» в строке «Критерий» необходимо вставить имя параметра с префиксом «двоеточие», например, **:ФИО**. Сохраняем запрос, например, с именем «Запрос_оклада» и выполняем его. При этом появится диалоговое окно «Ввод параметра» с приглашением «ФИО» и пустой строкой «Значение», в которой необходимо ввести ФИО преподавателя (Рис 4.6).

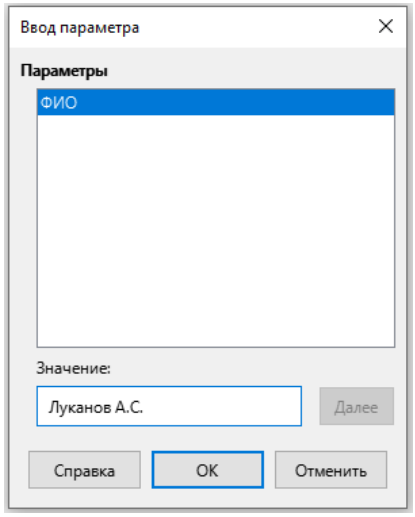


Рис 4.6 Окно для ввода параметров условия выборки

ФИО преподавателя	Должность	Оклад
Луканов А.С.	Доцент	53 500,00 ₽

Рис 4.7 Результат выполнения запроса

Замечание 4. Важно подчеркнуть, что значения текстовых параметров, набираемые с клавиатуры в поле «Значение:» должны абсолютно совпадать со значениями соответствующего параметра в таблице базы данных. Например, в данном примере ФИО преподавателей введены в таблицу «Преподаватель» с префиксом «два пробела» (для «красоты»)! Поэтому, если в поле «Значение:» ввести «Луканов А.С.» или «␣Луканов А.С.» вместо «␣␣Луканов А.С.», то окно выполнение запроса будет пустым.

Частичное решение данной проблемы достигается использованием оператора LIKE.

Например, если при построении запроса на основе таблицы «Студент» в строке «Критерий» поля ФИО прописать LIKE 'Г*', то при выполнении запроса будут выведены данные о следующих студентах (Рис 4.8).

ФИО студента	Дата рождения	Пол	Номер зачетки
Гуляева Л.А.	02.01.02	Ж	2102030310
Громова И.И.	07.06.02	Ж	2201050113
Громов А.И.	02.05.03	М	2202030302
Ге Н.Н.	09.01.02	М	2202030308
Ге Н.Н.	02.08.00	М	2302030105

Рис 4.8 Пример на применение оператора LIKE

Внимание! После выполнения рассмотренных выше упражнений, в качестве задания лабораторной работы для ИС БД «Деканат» необходимо самостоятельно создать 2–3 запроса с параметрами, причем значение параметра условия выборки должно хранится в одной таблице, а результаты выборки берутся из других таблиц. Например, «Вывести список сотрудников всех кафедр, которые работают в должности «Доцент»». (Рис 4.9)

ФИО сотрудника кафедры	Должность	Кафедра
Сироченко В.П.	Доцент	Информатики и вычислительной математики
Луканов А.С.	Доцент	Информатики и вычислительной математики
Иванов Д.В.	Доцент	Безопасности информационных систем
Китаева Е.В.	Доцент	Дифференциальных уравнений и теории управления

Рис 4.9 Пример запроса на основе трёх таблиц

4.3 Запросы на основе групповых функций

Специфика табличного формата хранения и обработки данных БД позволяет применять к данным групповые (итоговые по столбцу) функции. Например, вычислить число сотрудников на кафедре или вычислить средний оклад преподавателей и т.д.

В LibreOffice Base можно применять следующие групповые функции: AVG – вычисляет среднее, Количество, Максимум, Минимум, Сумма, группировка и т.д.

В качестве примера, рассмотрим запрос, который вычисляет общую сумму заработной платы по каждой кафедре.

Запрос создается на основе 3-х таблиц «Кафедра», «Преподаватель» и «Должность». В первую колонку добавляем поле «Название», во вторую колонку поле «Оклад». В первой колонке в строке «Функция» необходимо выбрать функцию «группировка». Во втором столбце в строке «Функция» выбираем функцию «Сумма». В итоге сформирован запрос, как показано на рисунке 4.10.

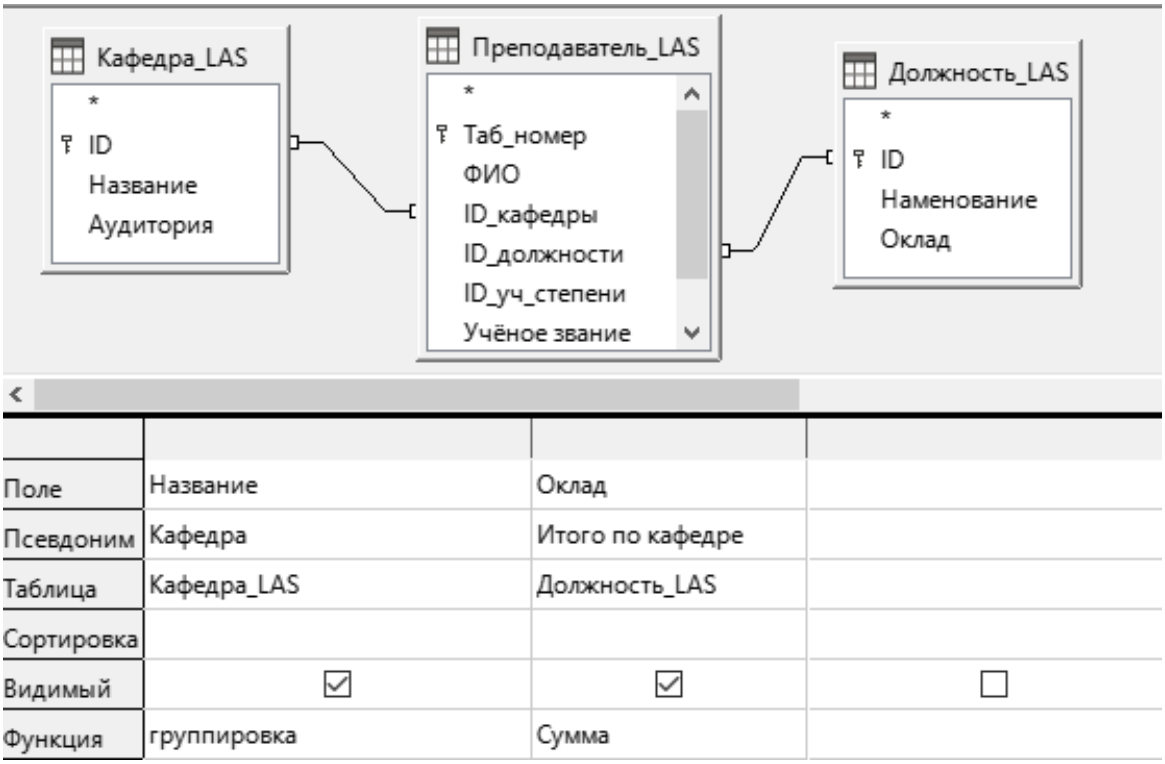


Рис 4.10 Запрос на функции группировки

Один из вариантов результата запроса на группировку представлен на рисунке 4.11

Кафедра	Итого по кафедре
Информатики и вычислительной математики	241 000,00 ₽
Безопасности информационных систем	123 500,00 ₽
Дифференциальных уравнений и теории управления	183 500,00 ₽
Алгебры и геометрии	70 000,00 ₽
Функционального анализа и теории функций	70 000,00 ₽
Математического моделирования в механике	70 000,00 ₽
Общей и теоретической физики	70 000,00 ₽

Рис 4.11 Результат запроса на функции группировки

Внимание! После выполнения рассмотренных выше упражнений, в качестве задания лабораторной работы для ИС БД «Деканат» необходимо самостоятельно создать 2–3 запроса с функциями группировки

4.4 Вопросы к собеседованию при сдаче лабораторной работы № 3

1. Почему запросы являются важным элементом ИС БД?
2. Какой оператор SQL лежит в основе любого запроса?
3. Какие три режима LibreOffice Base предлагает для создания запроса?
4. Расскажите алгоритм создания запроса в режиме дизайна?
5. Нужно ли указывать имена таблиц в строке «Таблица»?
6. Для чего используется строка «Псевдоним»?
7. Для чего используется строка «Видимый»?
8. Для чего используется строка «Функция»?
9. Для чего используется строка «Критерий»?
10. Как задать условие выборки?
11. Как можно увидеть текст SQL-запроса, который автоматически генерирует LibreOffice Base при создании запроса в режиме дизайна?
12. Как посмотреть результаты запроса, в режиме редактирования?
13. Чем запрос с параметрами отличается от «простого» запроса?
14. Как создать запрос с параметрами?
15. Для чего используется оператор LIKE в запросе с параметрами?
16. Какие групповые функции использует LibreOffice Base? Опишите их.
17. Для чего используется функция «группировка»?
18. Как функция «группировка» влияет на оператор SELECT?

5 СОЗДАНИЕ ОТЧЁТОВ. ОТЧЁТЫ КАК ЭЛЕМЕНТ ИС БД

Лабораторная работа № 4 посвящена работе с отчетами, созданными на основе данных БД.

Отчеты является наиболее ярким представителем составляющей информационных систем, так как именно отчеты позволяют зафиксировать либо актуальные состояние искомый предметной области на данный момент, либо представить данные, обработанные по определенному алгоритму.

5.1 Создание простых отчетов

Пункт меню базы данных «Отчеты» позволяет создавать отчет в двух режимах: 1) Создать отчет в режиме дизайна, 2) Использует мастер для создания отчета.

Рассмотрим алгоритм создания отчета в режиме мастера на примере отчета, который формирует данные на основе только одной таблицы для вывода списка преподавателей всех кафедр.

Окно «Мастер отчётов» предлагает выполнить шесть шагов.

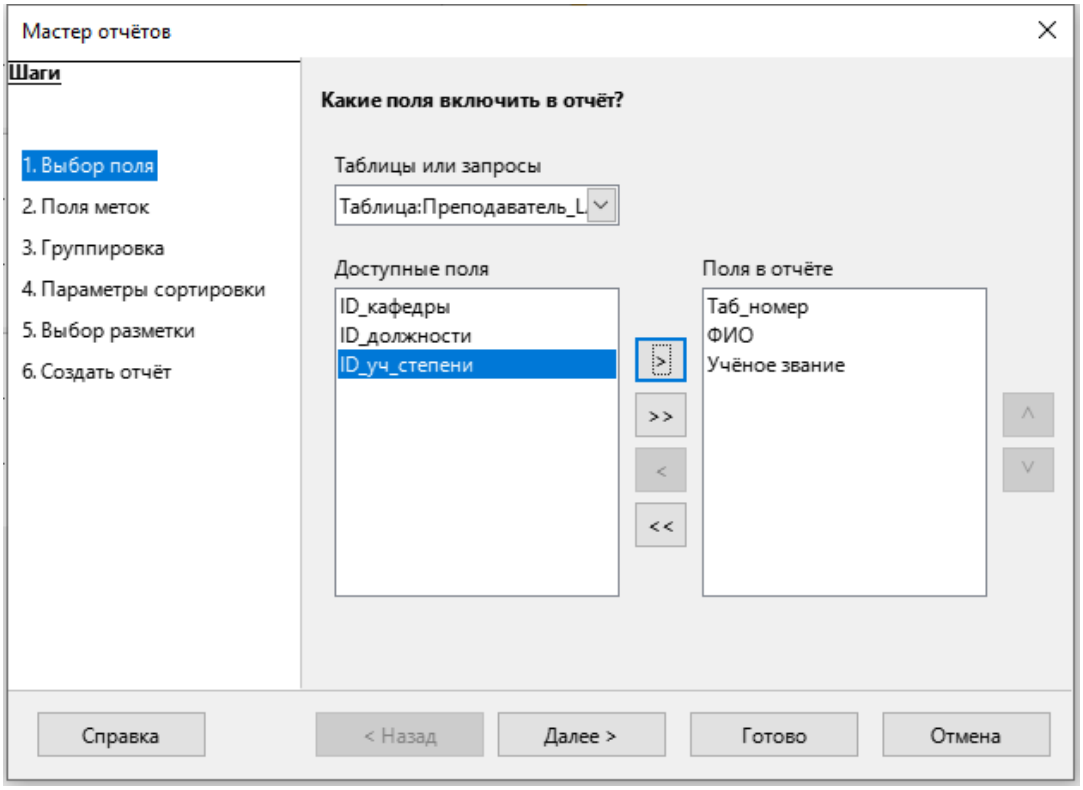


Рис 5.1 Первый шаг при создании простого отчета

1. Выбор поля. На первом шаге (Рис 5.1) сначала предлагается выбрать источник данных (Таблицы или запросы), а затем переместить необходимые поля в окно «Поля в отчете».

2. Поля меток. На втором шаге задаются заголовки столбцов (метки). Если имя поля используется «по умолчанию», то его редактировать не нужно (Рис 5.2)

3. Группировка. На третьем шаге данные отчета можно разбить на блоки-группы. В данном примере логически допустимым является группирование по полю «Ученое звание». Однако, пропустим этот шаг. (Студентам рекомендуется выполнить это упражнение самостоятельно.)

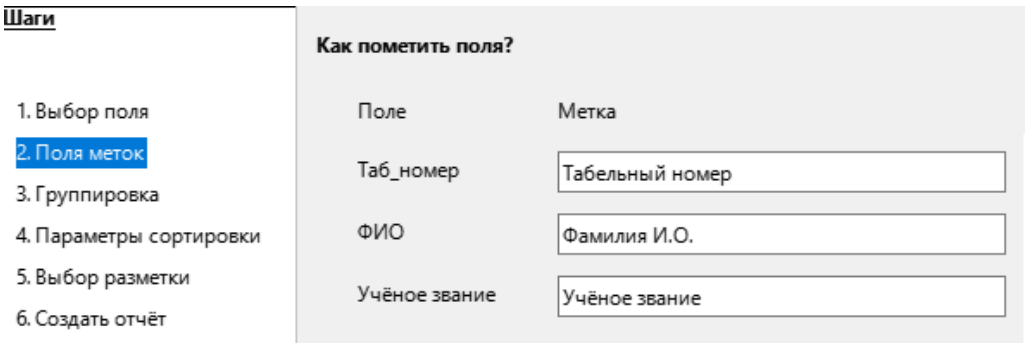


Рис 5.2 Редактирование оглавления отчета

4. Параметры сортировки. На четвертом шаге необходимо выбрать поля, по которым требуется отсортировать данные отчета. Поля можно сортировать по четырем уровням, на каждом – по возрастанию или по убыванию. Выберем сортировку по полю «Таб_номер», что фактически будет реализовывать неявное группирование сотрудников по кафедрам.

5. Выбор разметки. На пятом шаге необходимо выбрать вид отчета и ориентацию (альбомная или книжная). Несмотря на множественный выбор типа отчета в окне «Разметка данных», настоятельно рекомендуется оставлять табличный вид «по умолчанию».

6. Создать отчёт. На последнем шаге задаем тип отчета и имя файла, в котором будет сохранен отчет в формате OpenDocument (.odt) (Рис 5.3). Статический отчет зафиксирует в отчете те данные, которые были в базе данных на момент создания отчета, а динамический отчет будет отображать те данные, которые имелись в БД на момент запуска отчета на выполнение из меню «Отчеты».

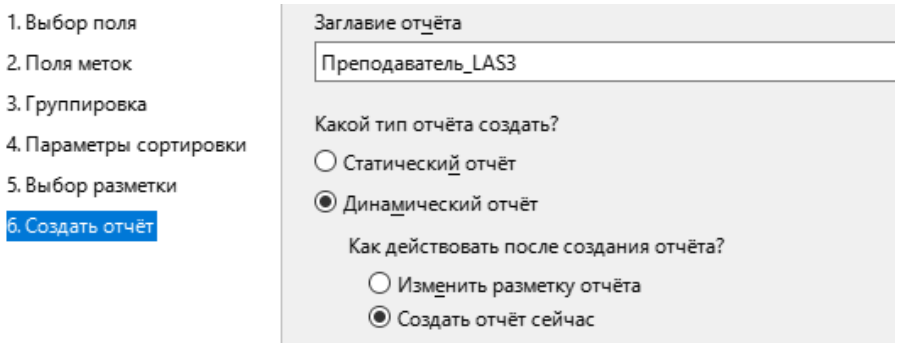


Рис 5.3 Оглавление файла и выбор типа отчета

Активная кнопка «Создать отчет сейчас» при нажатии «Готово» сразу открывает готовый отчет в текстовом процессоре LibreOffice Writer. Однако мастер, как правило, не позволяет создавать отчет. который сразу удовлетворяет требованиям задачи. Поэтому рекомендуется активировать кнопку «Изменить разметку отчета». Тогда при нажатии «Готово» открывается окно «Построитель отчетов» (Рис 5.4), в котором предварительно можно отформатировать отчет согласно требованиям оформления отчетов. Однако окончательное форматирование отчета рекомендуется выполнять непосредственно в текстовом процессоре.

В результате один из вариантов искомого отчета представлен на рисунке 5.5.

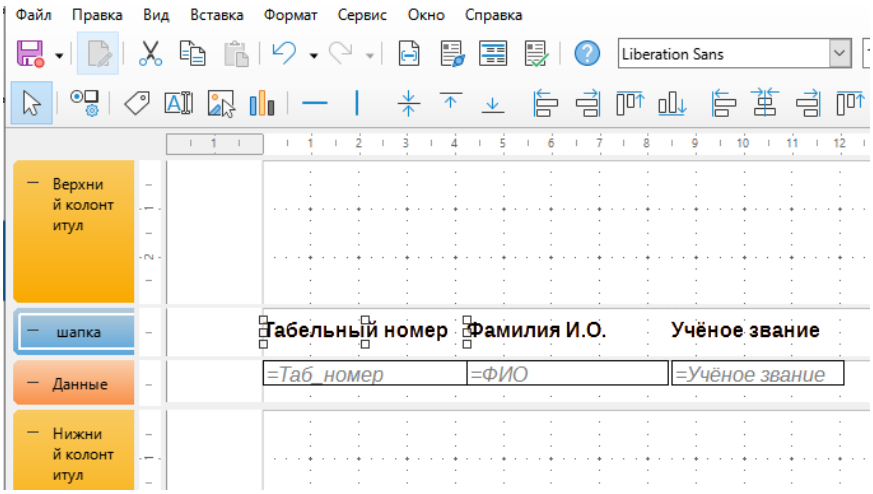


Рис 5.4 Окно редактирования отчетов «Построитель отчетов»

Табельный номер	Фамилия И.О.	Учёное звание
0101	Степанов А.Н	Профессор
0102	Сироченко В.П.	Доцент
0103	Луканов А.С.	Доцент
0105	Морозова Е.В.	-
0115	Бабич А.С	-
0205	Иванов Д.В.	Доцент
0211	Осипов М.Н.	Доцент
0808	Щепакина Е.А.	Профессор
0810	Воропаева Н.В.	Доцент
0812	Китаева Е.В.	Доцент
1105	Панов А.Н.	Профессор
1305	Асташкин С.В.	Профессор
1501	Степанова Л.В.	Доцент
1820	Салеев В.А.	Профессор

Рис 5.5 Отчет «Список преподавателей»

Внимание! После выполнения рассмотренных выше упражнений, в качестве задания лабораторной работы для ИС БД «Деканат» необходимо самостоятельно создать и ОТФОРМАТИРОВАТЬ 1–2 отчета на основе данных одной таблицы.

Однако, на практике, наибольший интерес представляют отчеты, созданные на основе данных нескольких таблиц. Очевидно, что подобные отчеты можно создавать только на основе запросов.

Например, отчёт, сформированный по запросу «Вывести список сотрудников всех кафедр, которые работают в должности «Доцент»» (Рис 4.9), формируется на основе данных трех таблиц.

Поэтому выполнив шаги мастера отчетов и отформатировав текст в текстовом процессоре LibreOffice Writer получим файл отчета, представленный на рисунке 5.6.

ФИО сотрудника кафедры	Должность	Кафедра
Сироченко В.П.	Доцент	Информатики и вычислительной математики
Луканов А.С.	Доцент	Информатики и вычислительной математики
Иванов Д.В.	Доцент	Безопасности информационных систем
Китаева Е.В.	Доцент	Дифференциальных уравнений и теории управления

Рис 5.6 Пример отчета, построенного на запросе

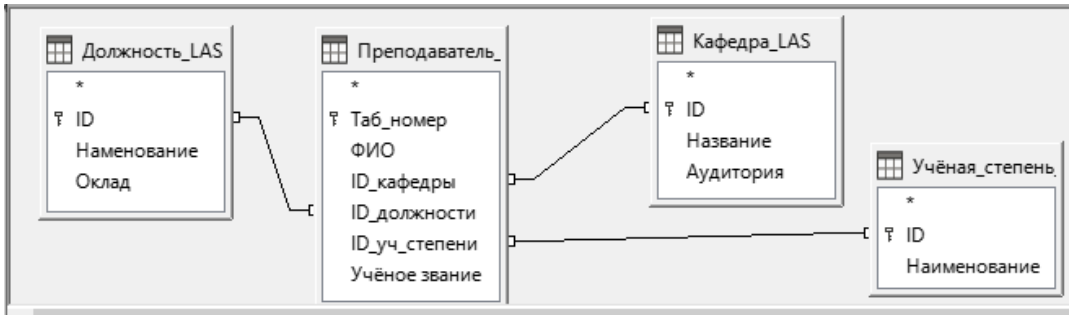
Внимание! После выполнения рассмотренных выше упражнений, в качестве задания лабораторной работы для ИС БД «Деканат» необходимо самостоятельно создать и ОТФОРМАТИРОВАТЬ 2–3 отчета на основе данных запросов, созданных в лабораторной работе № 3.

5.2 Создание отчетов с группировкой данных

Третий шаг мастера отчетов (Рис 5.1) позволяет настроить отчет по условию группирования.

Например, создадим отчёт для вывода расширенного списка преподавателей, сгруппированного по каждой кафедре.

Сначала необходимо создать запрос «Список преподавателей» с «раскодировкой» всех данных таблицы «Преподаватель» (Рис 5.7):



Поле	Название	ФИО	Наменование	Наименование	Учёное звание	
Псевдоним	Кафедра	ФИО преподавателя	Должность	Учёная степень		
Таблица	Кафедра_LAS	Преподаватель_LAS	Должность_LAS	Учёная_степень_L	Преподаватель_	
Сортировка						
Видимый	☑	☑	☑	☑	☑	☐

Рис 5.7 Запрос на расширенный список сотрудников кафедр

Тогда при создании отчета на шаге 3 для группировки выбираем поле «Кафедра» (Рис 5.8).

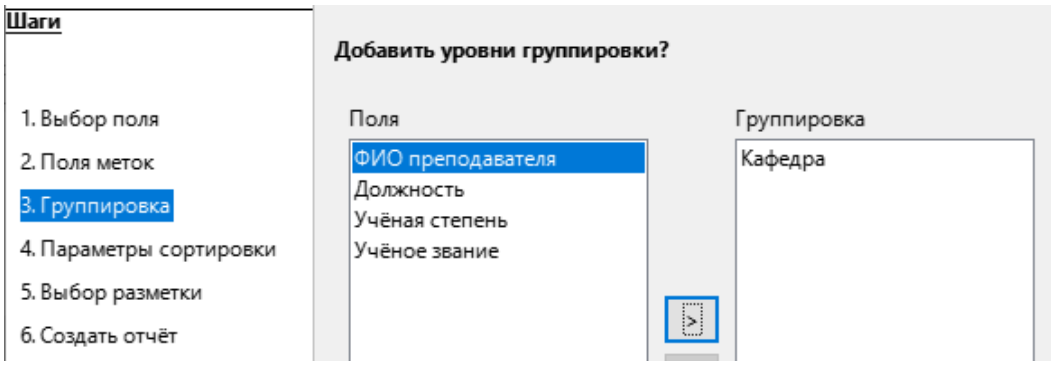


Рис 4.8 Группирование отчета по полю «Кафедра»

В результате фрагмент файла отчета после дополнительного форматирования будет иметь вид (Рис 5.9).

Кафедра	Дифференциальных уравнений и теории управления		
ФИО преподавателя	**Должность**	**Учёная степень**	**Учёное звание**
Воропаева Н.В.	Зам. декана	Кандидат физико-математических наук	Доцент
Китаева Е.В.	Доцент	Кандидат физико-математических наук	Доцент
Щепакина Е.А.	Заведующий кафедрой	Доктор физико-математических наук	Профессор

Кафедра	Информатики и вычислительной математики		
ФИО преподавателя	**Должность**	**Учёная степень**	**Учёное звание**
Бабич А.С.	Педагог-организатор	Без ученой степени	-
Луканов А.С.	Доцент	Кандидат физико-математических наук	Доцент
Морозова Е.В.	Старший преподаватель	Без ученой степени	-
Сироченко В.П.	Доцент	Кандидат физико-математических наук	Доцент
Степанов А.Н	Заведующий кафедрой	Доктор физико-математических наук	Профессор

Рис 5.9 Фрагмент отчета с группированием по полю «Кафедра»

Замечание 1. По условию лабораторной работы № 1 на каждой кафедре должны числиться не менее трёх сотрудников, поэтому и в отчете должно быть отражено не менее трёх сотрудников по каждой кафедре.

Внимание! После выполнения рассмотренных выше упражнений, в качестве задания лабораторной работы для ИС БД «Деканат» необходимо самостоятельно создать и ОТФОРМАТИРОВАТЬ 1-2 отчета с группированием.

5.3 Вопросы к собеседованию при сдаче лабораторной работы № 4

1. Почему отчеты являются важным элементом ИС БД?
2. К какому уровню архитектуры ANSI-SPARK относятся отчёты?
3. Как создать отчет в режиме дизайна?
4. Как создать отчет в режиме мастера?
5. Опишите первый шаг мастера отчётов – «Выбор поля».
6. Опишите второй шаг мастера отчётов – «Поле меток».
7. Опишите третий шаг мастера отчётов – «Группировка».
8. Опишите четвёртый шаг мастера отчётов – «Параметры сортировки».
9. Опишите пятый шаг мастера отчётов – «Выбор разметки».
10. Опишите шестой шаг мастера отчётов – «Создать отчёт».
11. Что лучше? Строить отчёты на основе данных таблиц или данных запроса?
12. Как можно изменить порядок сортировки значений того или иного поля в готовом отчете?
13. Когда нужно заполнять поля меток?
14. Приведите пример предметной области и укажите по какому параметру в ней возможно группирование.
15. Почему в качестве разметки рекомендуется использовать табличный тип отчёта?
16. Чем «статический» отчёт отличается от «динамического» отчёта?
17. Какие средства форматирования (редактирования) отчётов предоставляет LibreOffice Base?

6 ЛАБОРАТОРНАЯ РАБОТА «ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ»

6.1 Постановка задачи

Лабораторная работа № 5 предполагает самостоятельное проектирование реляционный база данных на основе **метода семантического моделирования** [1] для предложенной в каждом варианте искомой предметной области (ИПО) [3].

Варианты ИПО для студента определяются строго по журналу преподавателя.

Физическая реализация, разработанной студентом модели, допускается только после проверки модели и консультаций с преподавателем. Т. е. после «визирования» модели преподавателем.

В процессе создания таблиц-отношений, спроектированной базы данных, необходимо заполнять столбец «Описание» для описания домена, на котором определено данное поле – атрибут таблицы! Шаблон описания домена должен иметь имеет вид:

<Имя домена\Простой тип\Ограничения\Семантика>

В процессе заполнения таблиц данными, необходимо в «родительские» таблицы вводить не менее 5 строк, а «дочерние» – не менее 10.

После создания базы данных необходимо разработать элементы информационной системы: формы, запросы и отчёты. Причём, запросы и отчеты следует строить, исходя из глубокого понимания ИПО. Требования к формам, запросам и отчетам подробно рассмотрены в лабораторных работах № 2 - № 4.

При сдаче лабораторной работы студент должен продемонстрировать схему базы данных (и прокомментировать ее), показать, как можно внести новую информацию в базу данных и отредактировать уже имеющиеся записи, продемонстрировать работу форм, запросов и отчетов.

6.2 Варианты заданий [4]

1. **Разработайте ИС БД «Библиотека».** Она должна содержать данные о книгах, их авторах, ISBN, издательствах и времени издания книг. У книги может быть несколько авторов, автор же может написать несколько книг. В библиотеке может быть несколько экземпляров одной книги (с разными инвентарными номерами), при этом какие-то из них находятся «на стеллажах», а какие-то «на руках у читателей». ИС БД позволяет узнать, есть ли такая книга в библиотеке вообще и может ли она быть выдана в данный момент.

2. **Разработайте ИС БД «Аптека».** В аптеке продаются медикаменты, которые поступают от разных поставщиков. Один поставщик может поставлять несколько медикаментов (лекарственные препараты, предметы ухода за больными, предметы (средства) личной гигиены и т. п.). В то же время медикаменты одного наименования могут поставляться разными поставщиками. ИС БД позволяет «заказать» медикаменты у поставщика, «продать» медикаменты покупателю по рецепту.

3. **Разработайте ИС БД «Издательство».** Издательство работает с авторами, заказывая им книги по разным темам. При этом книги по одной теме могут быть заказаны нескольким авторам. В то же время один автор может работать над книгами сразу по нескольким темам. Книги состоят из глав. Завершив главу, автор «отсылает» ее в издательство. Когда книга готова более чем на 70%, издательство помещает ее в свой тематический план.

4. **Разработайте ИС БД «Магазин».** Магазин продает товары бытового назначения (род товаров определить самостоятельно). Покупатель может приобрести несколько товаров одновременно, при этом «выбивается» единый чек. Поскольку на проданный товар существует гарантия, необходимо обеспечить хранение сведений о товаре на протяжении всего гарантийного срока. В течение этого срока покупатель может обратиться в магазин с просьбой о замене некачественного товара. После проверки, действительно ли товар куплен в этом магазине (по номеру чека) и гарантия на него действует, товар должен быть заменен.

5. **Разработайте ИС БД «Подписка».** Почтовое отделение проводит подписку на газеты и журналы. Каждый подписчик может оформить подписку на несколько изданий одновременно. Разумеется, одно издание может быть выписано несколькими подписчиками. Будем считать, что подписка может быть оформлена на срок от 2 до 12 месяцев. ИС БД позволяет узнать, сколько времени осталось до окончания действия подписки (и, быть может, напомнить о том, что ее следует продлить). Среди подписных изданий могут появляться новые, а некоторые существующие могут прекратить выпуск. В последнем случае об этом следует известить подписчиков.

6. **Разработайте ИС БД «Фонотека».** В ней должны содержаться сведения об авторах и исполнителях музыкальных произведений, а также носителях, на которых они записаны (кассеты, диски различных форматов и т. п.). Одно и то же музыкальное произведение может быть записано в исполнении различных музыкантов. В то же время музыкант может исполнять различные произведения разных авторов. ИС БД позволяет узнать о наличии в фонотеке той или иной записи (поиск можно вести как по названию произведения, так и по фамилии автора или исполнителя), а также составить различные рейтинги (по числу обращений).

7. **Разработайте ИС БД «Аптечный склад».** На складе хранятся медикаменты, которые поступают от разных поставщиков и распределяются по сети аптек. Один поставщик может поставлять несколько медикаментов (лекарственные препараты, предметы ухода за больными, предметы (средства) личной гигиены и т. п.). В то же время медикаменты одного наименования могут поставляться разными поставщиками. ИС БД позволяет «заказать» медикаменты у поставщика, «отгрузить» медикаменты в аптеку (в том случае, если медикаменты в запрашиваемом количестве присутствует на складе).

8. **Разработайте ИС БД «Отдел кадров предприятия».** В ней должны храниться сведения о сотрудниках, их перемещениях по службе, изменениях в оплате труда (в том числе доплатах за стаж), об отпусках. Структурной единицей предприятия считается отдел. Сотрудник может работать только в одном отделе. Одновременно в отпуске может находиться не более 30 % списочного состава отдела, при этом «руководящий состав» должен уходить в отпуска строго по очереди (заместитель начальника отдела не может уйти в отпуск, пока из отпуска не вернется начальник). Сотрудник может переходить из одного отдела в другой, при этом стаж должен сохраняться. ИС БД позволяет узнать различные сведения о сотруднике, в том числе – может ли он уйти в отпуск в данные сроки.

9. **Разработайте ИС БД «Лыжный сезон».** В течение одного осенне-зимнего -весеннего периода одного года проводятся несколько соревнований по лыжам. Каждое соревнование (например, чемпионат России или чемпионат мира) проводятся в течение нескольких дней и состоят из множества лыжных дисциплин. Лыжные дисциплины для мужчин и женщин отличаются. ИС БД должна хранить и обрабатывать данные о лыжниках, их тренерах, результатах выступлений (место, финишное время). В идеале, ИС БД должна определять лучших лыжников по результатам всего сезона.

10. **Разработайте ИС БД «Хобби».** ИС БД должна содержать сведения о людях и их увлечениях. Каждый человек может иметь несколько увлечений. Несколько людей могут увлекаться одним и тем же. Список увлечений должен быть расширяемым. Следует хранить информацию об образовании и / или профессии каждого человека, его возрасте, степени увлеченности предметом (глубине знаний в этой области). ИС БД позволяет найти человека, который может наиболее точно ответить на вопрос из сферы интересов. Вопрос определяется сферами интереса человека.

11. **Разработайте ИС БД «Меню».** ИС БД должна содержать кулинарные рецепты различных блюд в разных категориях, например, «Закуска», «Горячее блюдо», «Десерт» и т. п. Рецепт каждого блюда приводится в расчете на одну порцию, указывается также длительность изготовления. ИС БД позволяет узнать, какие продукты и в каком количестве потребуются, согласно выбранному меню, на день (завтрак, обед, ужин) или на праздничный ужин, а также то, сколько времени потребуется на приготовление (возможно, временем приготовления, допустим, обеда можно считать время приготовления того блюда, у которого оно наибольшее).

12. **Разработайте ИС БД «Турагенство»**. ИС БД должна содержать данные турагенства и предлагаемые им услуги: страны, города, отели, условия проживания и проезда, экскурсионное обслуживание, сервис принимающей стороны, стоимость путевки. ИС БД должна формировать путёвку и предоставлять туристу данные о стране, экскурсиях, отелях и т.д.

13. **Разработайте ИС БД «Салон видеопроката»**. ИС БД должна содержать полные данные о фильмах: название, студия, жанр, год выпуска, режиссер, исполнители главных ролей, краткое содержание, субъективная оценка фильма. ИС БД должна отображать факт наличия фильма в видеотеке, оформление выдачи и возврата DVD или Blu-Ray диска. ИС БД, в идеале, должна формировать список фильмов с самым высоким рейтингом.

14. **Разработайте ИС БД «Ежедневник**. ИС БД должна содержать сведения о намечаемых мероприятиях – дате, времени и протяженности, места проведения и типа мероприятия. ИС БД должна «автоматически» напоминать о ближайших мероприятиях: по текущей дате и времени; «анализировать» возможные «накладки» – пересечение по дате и времени планируемых мероприятий. ИС БД должна планировать время на заданный период.

15. **Разработайте ИС БД «Бюро знакомств»**. ИС БД должна содержать данные о потенциальных женихах и невестах: ФИО, пол, увлечения, сведения о себе, требования к партнеру. ИС БД должна подбирать подмножества подходящих кандидатур, назначать даты свиданий и дату регистрации брака. ИС БД должна формировать отчет по заключенным бракам за заданный период.

16. **Разработайте ИС БД «Соревнования биатлонистов».** В течение одного осенне-зимнего -весеннего периода одного года проводятся несколько соревнований по биатлону. Каждое соревнование (например, чемпионат России или чемпионат мира) проводятся в течение нескольких дней и состоят из множества дисциплин биатлона. Дисциплины биатлона для мужчин и женщин отличаются. ИС БД должна хранить и обрабатывать данные о биатлонистах, их тренерах, результатах выступлений (место, число промахов, финишное время). В идеале, ИС БД должна определять самых «метких» биатлонистов по результатам стрельбы и самых «быстрых» – по результатам скорости на основании данных всего сезона.

17. **Разработайте ИС БД «Записная книжка».** ИС БД должна содержать анкетные данные родственников, друзей или сослуживцев по категориям. Данные включают в себя адреса,

телефоны, место работы или учебы, должность знакомых, коллег и родственников, характер знакомства, деловые качества и так далее. ИС БД должна «автоматически» формировать поздравления с днем рождения; упорядочение людей по категориям, интересам, возрасту; осуществлять поиск по произвольному шаблону.

18. **Разработайте ИС БД «Коммерческий банк»**. ИС БД должна содержать данные о коммерческом банке: название банка, адрес, статус (форма собственности) и банковские реквизиты (ИНН, номер счета, корреспондентский счёт, БИК, КБК). Аналогично, должны храниться данные клиентов банка (ФИО, паспортные данные, ИНН, номер счета). Банк предоставляет различные виды услуг (банковские карты, кредиты, депозиты и т. д.). Каждый клиент может заключать договора на несколько услуг. Договора по вкладам характеризуются размером начальной суммы, условиями вклада (капитализация, пополняемый, не пополняемый и т. д.) и процентной ставкой. ИС БД должна «подбирать» самый оптимальный вид вклада в зависимости от размера вклада и вычислять доход по вкладу.

19. **Разработайте ИС БД «Автосалон».** ИС БД должна содержать информацию о новых и подержанных автомобилях отечественных и зарубежных заводов: марка, год выпуска, технические характеристики, особенности исполнения, техническое состояние, запрашиваемая цена. База покупателей должна содержать контактные данные покупателя и требования к марке, техническим характеристикам и техническому состоянию. ИС БД должна «автоматически» подбирать варианты для покупателя и формировать заявки для поставщиков и перегонщиков (Автомобилевоз может за один рейс перевезти 7-8 автомобилей класса «В»).

20. **Разработайте ИС БД «Крылатые фразы».** ИС БД должна содержать информацию о пословицах, поговорках, афоризмах, каламбурах и других словесных курьезах. У крылатых фраз могут быть несколько авторов, а у одного автора могут быть несколько фраз. Если крылатая фраза пришла из иностранного языка, то у неё должен быть перевод на русский язык. ИС БД должна классифицировать крылатые фразы по авторам и источникам и осуществлять поиск по темам и ключевым словам.

21. **Разработайте ИС БД «Справочник астронома».** ИС БД должна содержать информацию о галактиках, звездных системах, созвездиях и о звёздах. Для каждой из зарегистрированных звезд известны: название, созвездие, видимая звездная величина, расстояние, координаты на небосклоне. ИС БД должна осуществлять поиск звезд заданного созвездия, самых ярких звезд, видимых звезд и созвездий в заданной точке земного шара в заданное время суток.

22. **Разработайте ИС БД «Ломбард»**. ИС БД должна содержать информацию о хранимых товарах и недвижимости: наименование товара; оценочная стоимость; сумма, выданная под залог; дата сдачи; срок хранения. База данных клиентов ломбарда должна содержать паспортные данные клиента, адрес проживания, телефоны контакта. ИС БД должна выполнять операции приема товара, возврата, оповещение залогодателя о приближении срока залога и продажи по истечении срока залога.

Список использованных источников

1. Кузнецов, С. Д. Введение в реляционные базы данных : учебное пособие / С. Д. Кузнецов. – 4-е изд. – Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2025. – 247 с. – ISBN 978-5-4497-0902-8. – Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROFобразование : [сайт]. — URL: https://profspo.ru/books/146337
2. Гущин, А.Н. Базы данных: учебник / А.Н. Гущин. - Москва : Директ-Медиа, 2014. – 266 с.: ил.,табл., схем. – ISBN 978-5-4458-5147-9
3. Лабораторный практикум по базам данных: метод. указания / М-во образования и науки Рос. Федерации, Самар. гос. ун-т ; [сост. М. С. Русакова, Е. В. Рогачева]. – Самара, 2015
4. LibreOffice. The Document Foundation: официальный сайт. – URL: https://ru.libreoffice.org/home (дата обращения 31.10.2025).

Учебное издание

Луканов Александр Сергеевич
Бабич Антон Сергеевич

БАЗЫ ДАННЫХ.
ОСНОВЫ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ

Учебно-методическое пособие

Редакционно-издательская обработка
издательства Самарского университета

Подписано в печать 30.07.2026. Формат 60×84 1/8.
Бумага офсетная. Печ. л. 5,75. Тираж 27 экз.
Заказ № . Арт.– З(УМП)/2026.

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИМЕНИ АКАДЕМИКА С. П. КОРОЛЕВА»
(САМАРСКИЙ УНИВЕРСИТЕТ)
443086, САМАРА, МОСКОВСКОЕ ШОССЕ, 34.

Издательство Самарского университета.
443086, Самара, Московское шоссе, 34.