

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Колледж экономики и информатики



УТВЕРЖДАЮ
Директор КЭИ УлГТУ
О.П. Каширина
2016г.

Методические указания
для студентов по выполнению
курсового проекта

МДК.02.02 Технология разработки и защиты баз данных
по специальности 09.02.03 Программирование в компьютерных системах

Ульяновск
2016

1 ОСНОВНЫЕ ПОЛОЖЕНИЯ

Основными целями курсового проектирования являются:

- закрепление знаний и умений, полученных при изучении дисциплины,
- выработка навыков организации баз данных и создания приложений для работы с базами данных.
- подготовка к последующему выполнению дипломного проекта.

В основу методических указаний положены стандарты единой системы конструкторской и технологической документации (ЕСКД и ЕСТД) единой системы подготовки производства (ЕСТПП) единой системы стандартов СЭВ.

Пояснительная записка должна удовлетворять требованиям ГОСТ 2.105-79 «Общие требования к текстовым документам» и ГОСТ 2.106-68 «текстовые документы».

Пояснительная записка курсового проекта пишется разборчиво и аккуратно без сокращений слов, за исключением общепринятых сокращений, а так же установленных в стандартах. Текст пояснительной записки выполняется на листах писчей бумаги формата А4 (297*210мм) ГОСТ 2301-68 по формам 5 и 5а ГОСТ 2.106-68. Пояснительная записка должна иметь титульный лист, бланк задания на курсовое проектирование, содержание курсового проекта, список использованной литературы.

При защите курсового проекта руководитель должен проверить качество работы, ее соответствие заданию, усвоение учащимся основного учебного материала.

1.1 Тематика курсовых проектов.

Тематика курсовых проектов должна отражать реальные конкретные задачи.

При составлении заданий необходимо исходить из примерно одинаковой сложности задания на каждого учащегося.

Задания выполняются индивидуально, предусматривают разработку базы данных: выбор и построение модели, выявление атрибутов, их взаимосвязей и ограничений, определение требований к создаваемому программному приложению; реализацию базы данных: выбор и обоснование СУБД, построение словаря данных, заполнение базы данных значениями, создание приложения базы данных; оценку работы базы данных.

Возможно определение системы управления базами данных (СУБД) в задании или самостоятельный выбор СУБД студентом в ходе выполнения курсового проекта. Примерные темы: Проектирование и реализация базы данных в СУБД ACCESS. Проектирование и реализация базы данных в DELPHI. Проектирование и реализация базы данных.

1.2 Содержание курсового проекта: Курсовой проект состоит из пояснительной записки объемом около 25-30 страниц, графической части и дискет с реализованной базой данных и программным приложением. Пояснительная записка составляется следующим образом:

- титульный лист;
- задание на курсовое проектирование;
- содержание с указанием страниц разделов;
- текстовая часть;
- приложения с распечатками программных модулей и прочими
- список используемой литературы

Графическая часть должна состоять из чертежей:

- структура базы данных
- структура таблицы базы данных.

2 ОБЩИЕ ТРЕБОВАНИЯ К ВЫПОЛНЕНИЮ КУРСОВОГО ПРОЕКТА:

2.1 Требования к выполнению пояснительной записки

Пояснительная записка должна содержать следующие разделы:

- 1 Введение.
- 2 Предварительное планирование и проверка осуществимости.
- 3 Постановка задачи (определение требований)
- 4 Разработка базы данных
- 5 Реализация базы данных:
 - 1.1 Выбор и обоснование СУБД
 - 1.2 Построение словаря данных
 - 1.3 Заполнение базы данных значениями
 - 1.4 Создание приложения
- 6 Оценка работы и поддержка базы данных.

Для обеспечения правил выполнения и оформления пояснительной записки следует выполнять требования ГОСТ 2.105-79 «Общие требования к текстовым документам» и ГОСТ 2.106-68 «Текстовые документы». Содержание пояснительной записки следует подразделять на основные части: разделы, подразделы, пункты и подпункты. Каждая часть должна иметь порядковый номер. Нумерация производится арабскими цифрами в пределах документа. Номер каждой составной части должен включать в себя номер соответствующих составных частей, более высоких ступеней. Номера раздела состоят из номера раздела и подраздела, разделенных точкой. Наименование разделов и подразделов должно быть арабским и соответствовать содержанию документа и записываться в виде заголовков. Переносы слов и их сокращения в заголовках не допускаются, и точку в конце заголовка не ставят. Если заголовки состоят из 2-х и более предложений, их разделяют точкой. Каждый раздел пояснительной записки начинают с новой страницы. Цифры, указывающие номера пунктов не должны выступать за границы абзаца. Для разделов и подразделов расстояние между последней строкой и последующим заголовком должно быть примерно 15мм.

Терминология и определения должны быть едиными и соответствовать общепринятым в научно-технической литературе. Условное буквенное обозначение должно соответствовать государственным Стандартам.

Значение символов и числовых коэффициентов должны быть приведены под формулой. Формулы нумеруются арабскими цифрами в пределах раздела и порядкового номера формулы, разделенных точкой.

Например:

$$B=P+O \quad (2.1.)$$

Все размещаемые иллюстрации, если их более одной, нумеруются арабскими цифрами по разделам. Номер иллюстрации состоит из порядкового номера раздела и порядкового номера иллюстрации, разделенных точкой, например: 1.1.

Ссылки на ранее упомянутые иллюстрации дают с сокращением слова «см» смотри, например: см. рис.2.1.

Цифровой материал необходимо оформлять в виде таблиц без графы № по полям.

При использовании справочной литературы даются ссылки на использованные источники с указанием в квадратных скобках порядкового номера источника, помещенного в конце пояснительной записки.

2.2 Требования к выполнению графической части курсового проекта.

Графическая часть выполняется в полном соответствии с действующими стандартами ЕСКД. Выполняется на листах любых форматов, установленных ГОСТ 2.301-68. Основная надпись ГОСТ 2.104-68 Общий объем графической части 1 лист формата А1.

3 МЕТОДИКА ВЫПОЛНЕНИЯ РАЗДЕЛОВ КУРСОВОГО ПРОЕКТА.

Основные вопросы, касающиеся курсового проектирования, разрабатываются в процессе изучения курса «Базы данных» при выполнении практических работ. Далее дается только краткая характеристика разделов пояснительной записки и краткие теоретические сведения по разработке базы данных.

1 Раздел. Введение.

Во введении следует дать краткий анализ проведенной работы:

- 1.1 необходимость применения баз данных в предприятиях и организациях любых форм собственности и разрабатываемой базы данных для конкретного предприятия или организации,
- 1.2 описание реализованной базы данных
- 1.3 описание программного приложения для базы данных
- 1.4 структура файлов базы, находящихся на дискете, порядок установки базы данных и программного приложения на компьютер пользователя

2 Раздел. Предварительное планирование и проверка осуществимости.

В ходе предварительного планирования проводится разработка стратегического плана базы данных: в каком виде на данный момент осуществляется работа по рассматриваемой задаче на предприятии или организации, существует ли уже какой-то вариант разрабатываемой базы данных или прикладных программ, используются ли другие базы данных, какие из них могут быть связаны с разрабатываемой, какие новые приложения находятся в процессе создания. Эта информация должна быть использована для установления связей между разрабатываемой базой данных и другими приложениями. Она также помогает определить будущие требования к системе и получить от системы БД экономическую выгоду.

В ходе проверки осуществимости определяется:

- технологическая осуществимость, т.е. определение доступности оборудования и программного обеспечения, необходимых для работы базы данных.
- операционная осуществимость, т.е. выяснение наличия персонала, способного работать с базой данных.
- экономическая целесообразность: как скоро можно ожидать выгоду, каковы издержки и риски при реализации системы базы данных, как система базы данных может помочь в реализации долгосрочных планов организации.

3 Раздел. Постановка задачи.

Постановка задачи или определение требований к системе базы данных включает в общем случае выбор целей базы данных, выяснение информационных потребностей различных отделов и служб, требований к оборудованию и программному обеспечению, разработка плана поэтапного создания системы, включающего выбор исходных приложений.

В ходе курсовой работы при постановке задачи следует определить содержащиеся в базе сведения о том, как она должна использоваться и какую информацию пользователь будет получать в процессе ее эксплуатации. В результате устанавливаются атрибуты, которые должны содержаться в отношении базы данных и уточненные требования к программному приложению. В пояснительной записке следует перечислить атрибуты и их краткие характеристики, описать какие операции с данными предусматриваются в программном приложении (ввод новых данных, корректирование или удаление существующих, поля сортировок и условия выборок из базы, форма отчета).

4 Раздел. Разработка базы данных.

Разработка базы данных проводится по методу Сущность-связь, который включает этапы:

- 1 выявление сущностей и связей между ними
- 2 построение ER-диаграмм с учетом всех сущностей и их связей
- 3 формирование набора предварительных отношений с указанием предполагаемого первичного ключа для каждого отношения и использованием ER-диаграмм
- 4 добавление неключевых атрибутов в отношение
- 5 приведение предварительных отношений к нормальной форме Бойса-Кодда
- 6 пересмотр ER-диаграмм при невозможности приведения к нормальной форме Бойса-Кодда или невозможности логического обоснования нахождения некоторых атрибутов в предварительных отношениях.

МЕТОД СУЩНОСТЬ-СВЯЗЬ

→ Метод сущность-связь называют также методом ER-диаграмм (Essence-сущность, Relation-связь).

Основными понятиями метода являются следующие:

- сущность
- атрибут сущности
- ключ сущности
- связь между сущностями
- степень связи
- класс принадлежности экземпляров сущности
- ER-диаграммы

Сущность представляет собой множество вещей одного типа, информация о которых хранится в базе данных. Экземпляры (элементы) сущности отличаются друг от друга и однозначно идентифицируются. Названиями сущностей, как правило, являются существительные, например: Преподаватель, Группа.

Атрибут представляет собой свойство сущности. Так, атрибутами сущности Преподаватель может быть Фамилия, Стаж работы, Оклад и т.д.

Ключ сущности – атрибут или набор атрибутов, используемый для идентификации экземпляра сущности.

Связь двух и более сущностей предполагает зависимость между атрибутами этих сущностей. Название связи обычно представляются глаголами. Например: Преподаватель **Ведет** Предмет .

Следует иметь в виду, что в процессе проектирования могут быть получены несколько вариантов одной БД. Так, два разных проектировщика, рассматривая одну и ту же проблему, могут получить различные наборы сущностей и связей. При этом оба варианта могут быть рабочими и выбор лучшего из них будет результатом личных предпочтений.

С целью повышения наглядности и удобства проектирования для представления сущностей и связей между ними используются ER-диаграммы.

Например, на рис.1 приведена ER-диаграмма сущностей Преподаватель и Дисциплина и связи между ними Ведет Дисциплину



Рис.1 Пример ER-диаграммы

На начальном этапе проектирования БД выделяют атрибуты, составляющие ключ сущностей.

На основе анализа ER_диаграмм формируют отношения проектируемой БД. При этом учитывается степень связи сущностей и класс их принадлежности.

Степень связи является характеристикой связи между сущностями, которая может быть типа: 1:1, 1:M, M:1, M:M.

Связь 1:1 имеет место в том случае, когда одному экземпляру первой сущности соответствует один экземпляр другой.

Связь 1:M имеет место, если одному экземпляру первой сущности соответствуют один или несколько экземпляров второй.

Связь M:1 имеет место, если нескольким экземплярам первой сущности соответствует один экземпляр другой.

Связь M:M имеет место, если нескольким экземплярам первой сущности соответствуют несколько экземпляров другой.

Класс принадлежности может быть обязательным и необязательным. Класс принадлежности является обязательным, если все экземпляры этой сущности обязательно участвуют в рассматриваемой связи, в противном случае класс принадлежности является необязательным.

Правила формирования отношений.

Правила формирования отношений основываются на учете степени связи между сущностями и класса принадлежности экземпляров сущностей.

Рассмотрим 6 правил.

Формирование отношений для связи 1:1

Правило 1. Если имеется тип связи 1:1 и класс принадлежности обеих сущностей обязательный, то формируется одно отношение. Первичным ключом может быть ключ любой из сущностей.

Правило 2. Если тип связи 1:1 и класс принадлежности одной сущности обязательный, другой - необязательный, то под каждую из сущностей формируется по отношению с первичными ключами, являющимися ключами соответствующих сущностей. Далее к отношению, сущность которого имеет обязательный класс принадлежности, добавляется в качестве атрибута ключ сущности с необязательным классом.

Правило 3. Если степень связи 1:1 и класс принадлежности обеих сущностей является необязательным, то используются три отношения. Два отношения соответствуют связываемым сущностям, ключи которых являются первичными ключами в этих отношениях. Третье отношение является связным между первыми двумя, поэтому его ключ объединяет ключевые атрибуты связываемых отношений.

Формирование отношений для связи 1:M

Правило 4. Если тип связи 1:M (или M:1) и класс принадлежности M-связной сущности обязательный, то достаточно формирование двух отношений (по одному на каждую сущность). При этом первичными ключами этих отношений являются ключи их сущностей. Кроме этого, ключ 1-связной сущности добавляется как атрибут (внешний ключ) в отношение, соответствующее M-связной сущности.

Правило 5. Если тип связи 1:M (M:1) и класс принадлежности M-связной сущности является необязательным, то необходимо формирование трех отношений. Два отношения соответствуют связываемым сущностям, ключи которых являются первичными в этих отношениях. Третье отношение является связным между первыми двумя. (его ключ объединяет ключевые атрибуты связываемых отношений)

Формирование отношений для связи M:M.

Правило 6. Если тип связи M:M, то независимо от класса принадлежности сущностей формируются три отношения. Два отношения соответствуют связываемым сущностям и их ключи являются первичными ключами этих отношений. Третье отношение является связным между первыми двумя, а его ключ объединяет ключевые атрибуты связываемых отношений.

Нормализация отношений.

Нормализация – процесс приведения реляционных таблиц к стандартному виду. При этом БД избавляется от избыточного дублирования данных и аномалий.

Избыточность данных – повторение данных в БД. *Аномалиями* называют такую ситуацию в БД,

Для определения нормальных форм дадим понятия:

Функциональная зависимость - значение атрибута в кортеже(строке) однозначно определяет значение другого атрибута в кортеже (каждому значению А, соответствует в точности одно значение В). Транзитивная зависимость, если для атрибутов А,В и С выполняется условие: А → В, и В → С, но обратная зависимость отсутствует.

Детерминант- атрибут в левой части функциональной зависимости, определяют значения других атрибутов кортежа(строки)

Чтобы предотвратить появление или избавиться от аномалий используется метод нормальных форм, заключающийся в разбиении таблицы на несколько таблиц

1 нормальная форма. Отношение находится в 1 нормальной форме, если значения в таблице являются простыми (имеют единственное значение). Исходное отношение строится таким образом, чтобы оно было в 1-ой нормальной форме.

2 нормальная форма . Отношение находится во 2-ой нормальной форме, если оно находится в 1-ой нормальной форме, и каждый неключевой атрибут функционально полно зависит от первичного ключа - составного (никакие неключевые атрибуты не являются функционально зависимыми от части ключа).

Для устранения частичной зависимости и перевода отношения в 2-ю нормальную форму необходимо, используя операцию проекции, разложить его на несколько отношений следующим образом:

- построить проекцию без атрибутов, находящихся в частичной зависимости от первичного ключа
- построить проекции на части составного первичного ключа и атрибуты, зависящие от этих частей.

3 нормальная форма. Отношение находится в 3-ей нормальной форме, если оно находится во 2-ой нормальной форме и все неключевые атрибуты взаимно независимы. (каждый неключевой атрибут нетранзитивно зависит от первичного ключа).

Транзитивные зависимости также порождают избыточное дублирование информации в отношении. Они устраняются использованием операции проекции на атрибуты, являющиеся причиной транзитивных зависимостей.

На практике построение 3 нормальной формы является достаточным и процесс нормализации таблиц на этом заканчивается.

Однако, если в отношении имеется зависимость атрибутов составного первичного ключа от неключевых атрибутов, то необходимо перейти к усиленной нормальной форме или нормальной

Пример проектирования БД учебной части.

БД учебной части по заданию содержит следующие сведения:

ФИО – фамилия и инициалы преподавателя (возможность совпадения фамилии и инициалов у преподавателей исключена)

Разряд – преподавательский разряд

Оклад – оклад преподавателя по разряду

ПрКом – предметная комиссия, в которой работает преподаватель

Стаж – преподавательский стаж

ДСтаж – доплата за стаж работы

Предмет – название предмета, который ведет преподаватель

Группа – номер группы, в которой преподаватель проводит занятия.

Кол_час – количество часов на проводимый в группе предмет

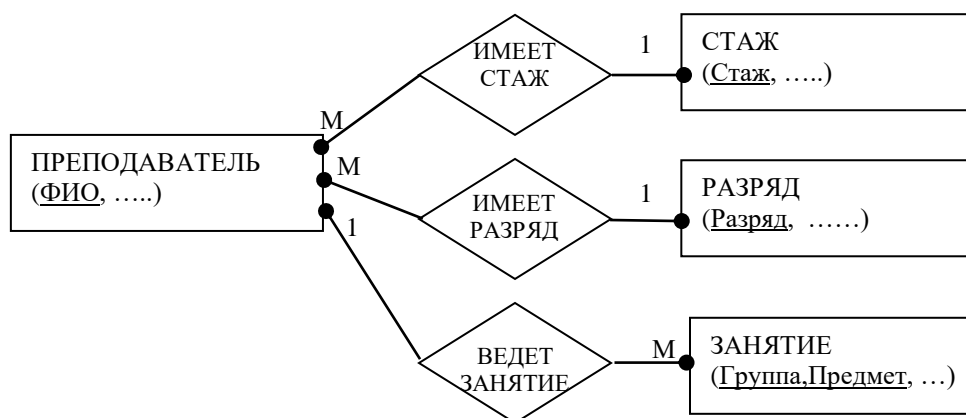
1 этап проектирования – выделение сущностей и связей между ними:

| Тип сущности | Сущности | Ключ |
|---------------|-----------------|------|
| ПРЕПОДАВАТЕЛЬ | ФИО | |
| ЗАНЯТИЕ | Группа, Предмет | |
| РАЗРЯД | Разряд | |
| СТАЖ | Стаж | |

Связи между сущностями:

| Тип связи | Типы сущностей |
|---------------|------------------------|
| ИМЕЕТ_РАЗРЯД | ПРЕПОДАВАТЕЛЬ, РАЗРЯД |
| ВЕДЕТ_ЗАНЯТИЕ | ПРЕПОДАВАТЕЛЬ, ЗАНЯТИЕ |
| ИМЕЕТ_СТАЖ | ПРЕПОДАВАТЕЛЬ, СТАЖ |

2 этап построение ER-диаграмм:



Виды связей:

Связь ИМЕЕТ_РАЗРЯД является связью типа М:1, т.к. одинаковый разряд могут иметь несколько преподавателей. Сущность ПРЕПОДАВАТЕЛЬ имеет обязательный класс принадлежности, т.е. каждый преподаватель имеет свой разряд. Сущность РАЗРЯД имеет необязательный класс принадлежности, т.е. возможны такие значения разряда, которые не имеет ни один преподаватель.

Связь ИМЕЕТ_СТАЖ является связью М:1, т.к. одинаковый стаж могут иметь разные преподаватели. Сущность ПРЕПОДАВАТЕЛЬ имеет обязательный класс принадлежности, т.е. каждый преподаватель имеет свой стаж педагогической работы. Сущность СТАЖ имеет необязательный класс принадлежности, т.е. возможны такие значения стажа, которые не имеет ни один преподаватель.

Связь ВЕДЕТ_ЗАНЯТИЕ имеет тип 1:М, т.к. преподаватель может вести несколько занятий, а занятие может проводиться только одним преподавателем. Обе сущности имеют обязательный класс принадлежности (предполагаем, что нет преподавателей, не проводящих занятия и нет занятий, которые не обеспечены преподавателями).

3 этап проектирования – формирование набора предварительных отношений с указанием предполагаемого первичного ключа для каждого отношения на основе анализа диаграммы:

На основе анализа диаграммы и с помощью шести правил формируем набор предварительных отношений:

Связь ИМЕЕТ_РАЗРЯД удовлетворяет условиям правила 4, в соответствии с которым получаем два отношения:

- 1.ПРЕПОДАВАТЕЛЬ (ФИО, Разряд, ...) – добавился внешний ключевой атрибут Разряд
- 2.РАЗРЯД (Разряд, ...)

Связь ИМЕЕТ_СТАЖ удовлетворяет условиям правила 4, получаем два отношения:

- 1.ПРЕПОДАВАТЕЛЬ(ФИО, Стаж,...) – добавился внешний ключевой атрибут Стаж
- 2.СТАЖ (Стаж,...)

Связь ВЕДЕТ_ЗАНЯТИЕ удовлетворяет условиям правила 4, получаем два отношения:

- 1.ПРЕПОДАВАТЕЛЬ(ФИО, ...)
- 2.ЗАНЯТИЕ(Группа, Предмет, ФИО) - добавился внешний ключевой атрибут ФИО

4 этап проектирования - добавление неключевых атрибутов:

ПРЕПОДАВАТЕЛЬ(ФИО, Разряд, Стаж, ПрКом)

РАЗРЯД(Разряд, Оклад)

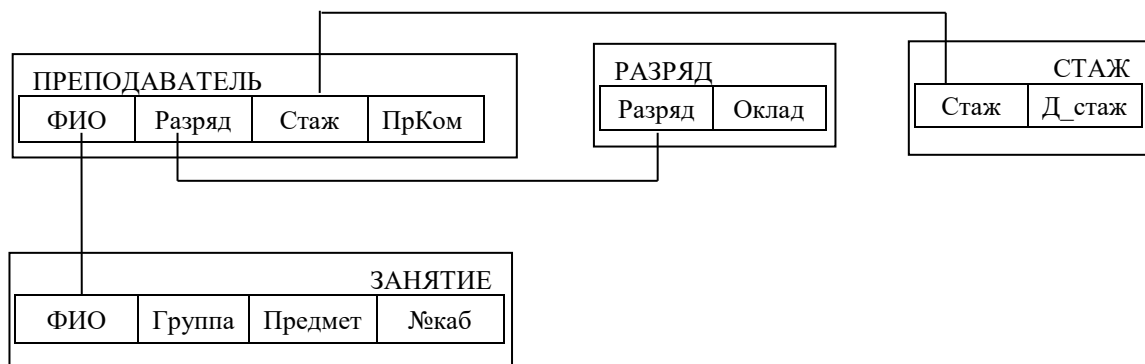
СТАЖ (Стаж, Д_стаж)

ЗАНЯТИЕ(Группа, Предмет, ФИО, №каб)

5 этап - проверяем определенные отношения на соответствие нормальной форме Бойса-Кодда.

После определения отношений проверяем их на соответствие требованиям нормальной формы

Бойса-Кодда. Полученная схема имеет вид:



5 Раздел. Реализация базы данных.

5.1 Выбор и обоснование СУБД.

Если в задании не указана конкретная СУБД (система управления базами данных), например, Access, студент должен сам подобрать подходящую, консультируясь с преподавателем. Это могут быть СУБД: Paradox, FoxPro, dBase, Access и прочие. При реализации в Access база данных и приложение создается в среде этой СУБД, в остальных случаях приложение разрабатывается в среде Delphi.

5.2 Построение словаря данных.

Словарь данных включает в себя информацию о структурах данных, взаимосвязях файлов БД друг с другом, типах данных и форматах их представления, принадлежности данных пользователям, кодах защиты и разграничения доступа и т.п.

В СУБД Access функции словаря данных выполняются непосредственно в ней, в Delphi реализуется с помощью утилиты Database Desktop.

Основные правила создания баз данных и таблиц в Access или в среде Delphi и типы данных основных СУБД можно узнать из методических пособий по лабораторно-практическим работам по курсу «Базы данных» или из литературы.

В пояснительной записке следует кратко описать создание базы данных и входящих в нее таблиц. Структура таблиц дается в виде таблицы.

Например, таблица ПРЕПОДАВАТЕЛЬ, которая реализована в СУБД PARADOX:

имя таблицы: PREP.DB

| <i>№ поля</i> | <i>Наименование поля</i> | <i>Имя поля</i> | <i>Тип поля</i> | <i>Размер</i> | <i>Ключевое поле</i> | <i>Свойства</i> |
|---------------|--------------------------|-----------------|-----------------|---------------|----------------------|---------------------------|
| 1 | Код преподавателя | Id_pr | A | 3 | * | required field |
| 2 | ФИО | FIO | A | 50 | | required field |
| 3 | Разряд | Razr | N | | | min value=8, max value=15 |
| 4 | Стаж | Stag | N | | | |
| 5 | Предметная комиссия | Pr_kom | A | 4 | | |

Аналогично для всех таблиц БД.

5.3 Заполнение базы данных значениями.

Для наглядности и возможности оценки работы базы данных и приложения следует ввести 5-10 строк значений в таблицы БД. Заполненные данные даются в пояснительной записке.

Например, заполнение значениями таблицы PREP.DB:

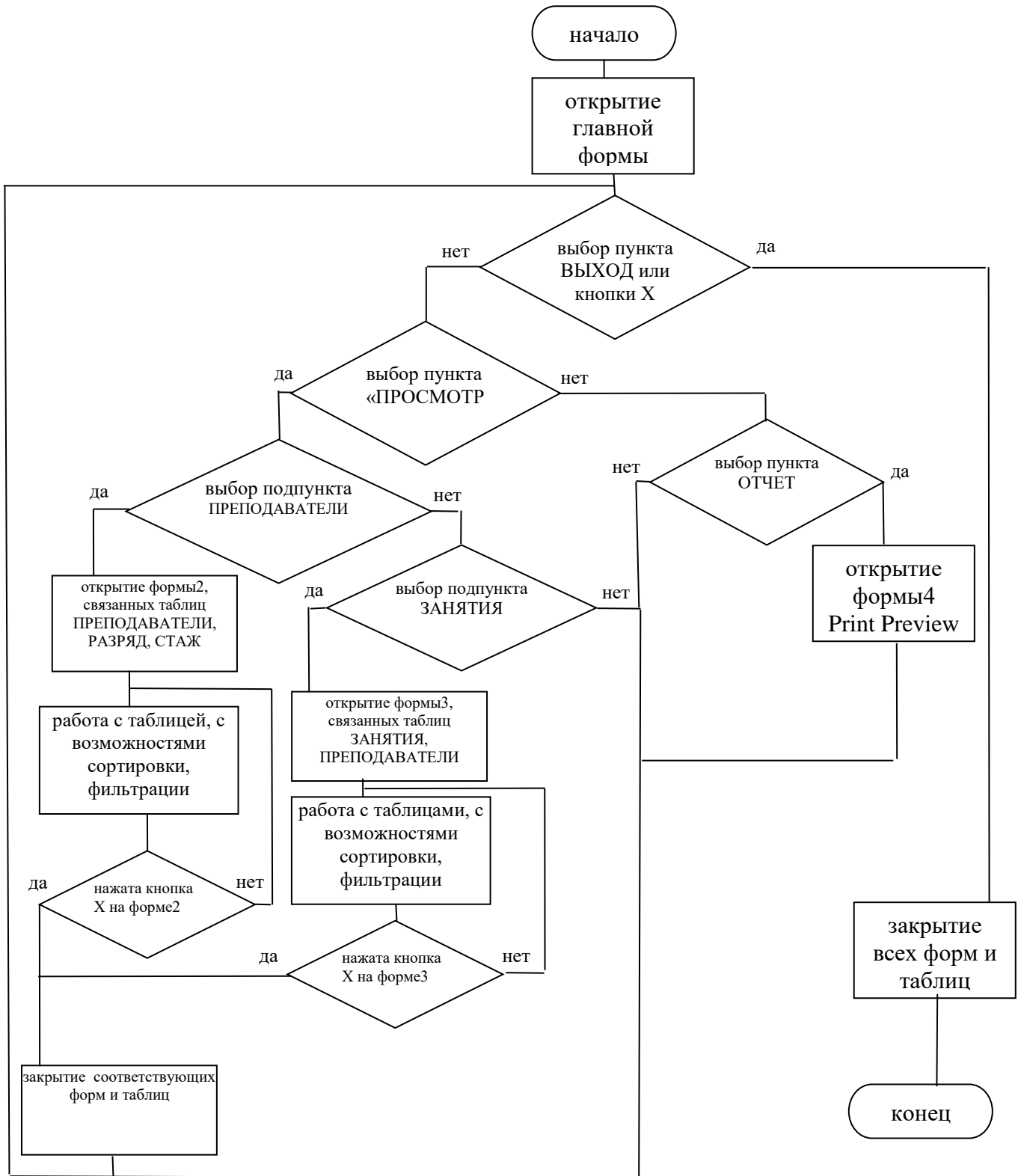
| <i>Id_pr</i> | <i>FIO</i> | <i>Razr</i> | <i>Stag</i> | <i>Pr-kom</i> |
|--------------|---------------|-------------|-------------|---------------|
| 001 | Иванов В.П. | 9 | 0 | 2202 |
| 002 | Петров С.П. | 10 | 5 | 2201 |
| 003 | Федорова К.Л. | 15 | 25 | 2913 |
| 004 | Еремеева А.Л. | 13 | 16 | 2913 |
| 005 | Китаева Н.Н. | 13 | 20 | 2201 |
| 006 | Клюев С.П. | 12 | 10 | 1201 |

Аналогично для всех таблиц БД.

5.4 Создание приложения.

Разрабатывается алгоритм программы, согласно требованиям, уточненным на этапе постановки задачи. Блок-схема алгоритма дается в пояснительной записке и выполняется в графической части.

Например, примерный алгоритм для приложения к БД «Учебная часть»



На основе алгоритма создается приложение. Приложение представляет собой программу или комплекс программ, обеспечивающих автоматизацию обработки информации для базы данных. Приложения могут создаваться в среде и вне среды СУБД. Приложения, разработанные в среде СУБД часто называют приложениями СУБД, а приложения, разработанные вне СУБД – внешними приложениями.

Приложение должно иметь удобный пользовательский интерфейс, обязательно включающий кнопки и/или меню, отчеты, обеспечивать надежную работу с базой данных.

Для приложений, созданных в Delphi в пояснительной записке следует дать тексты программных модулей и файла описания форм.

Так как представление структуры базы данных, экранных форм, запросов, отчетных форм в Microsoft Access обеспечивается внутренней реализацией СУБД, то представление полного программного кода базы данных, написанной на Microsoft Access является невозможным, в связи с отсутствием такового. Однако, при создании экранных форм для присвоения различных функций элементам управления - кнопкам - используется внутренний язык СУБД - Access Basic. Процедуры, написанные на этом языке, представляются в пояснительной записке

6 Раздел. Оценка работы и поддержка базы данных.

После завершения реализации работа системы оценивается с точки зрения обеспечения выполнения предъявленных требований. В случае необходимости вносятся изменения и поправки.