Министерство науки и высшего образования Российской Федерации
ФГАОУ ВО «Севастопольский государственный университет»

Институт информационных технологий

Управление данными

Методические указания
по выполнению курсовой работы

для студентов всех форм обучения направления подготовки
09.03.02 «Информационные системы и технологии»
09.03.03 «Прикладная информатика»

Севастополь
2023

СОДЕРЖАНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ СОКРАЩЕНИЙ

ПрО – предметная область
БД – база данных
ERD – Entity-Relationship Diagrams (диаграмма «сущность-связь»)
IDEF – Integrated DEfinition Function)

1 ЗАДАНИЕ НА РГР

Осуществить проектирование (разработать логическую и физическую модель) и реализовать базу данных предметной области согласно варианту.

Вариант предметной области выбирается согласно номеру группы в потоке и порядковому номеру в журнале, таблицы вариантов приведены в Приложении А.

2 СТРУКТУРА ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ К РГР

Титульный лист (образец Приложение Б)
Содержание
Введение
1. Аналитическая часть
   1.1. Анализ предметной области (концептуальное моделирование)
   1.2. Постановка задачи
2. Разработка логической модели базы данных
   2.1. Построение диаграммы «сущность-связь» в нотации П.Чена.
   2.2. Построение модели основанной на ключах (Key Based model, KB)
   2.3. Построение полной атрибутивной модели в нотации IDEF1X (нормализация отношений до третьей или четвѐртой нормальной формы)
3. Разработка физической модели базы данных
   3.1. Выбор аппаратной и программной платформы для реализации БД
   3.2. Реализация базы данных
   3.3. Тестирование базы данных (создание и реализация запросов)
   3.4. Разграничение прав доступа
4. Разработка клиентского приложения
   4.1. Обоснование выбора языка программирования
   4.2. Разработка интерфейса пользователя
   4.3. Алгоритм работы каждого из модулей
   4.4. Тестирование работы приложения
Заключение
Список использованных источников
Приложения (структуры таблиц базы данных, тексты программ и т.д.)

3 ПОРЯДОК ВЫПОЛНЕНИЯ И СОДЕРЖАНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ К РГР

3.1 Аналитическая часть

База данных фактически представляет собой модель предметной области (ПрО). Значит, для создания БД надо сначала проанализировать ПрО и создать ее модель (этап носит название – инфологическое проектирование).

3.1.1 Анализ предметной области (концептуальное моделирование)

Основой для анализа предметной области служат документы, которые отражают ПрО, и информация, которую можно получить от специалистов этой предметной области в процессе общения с ними.

Для анализа берутся те документы, которые имеют отношение к решаемой задаче. Изучение документов позволяет выявить объекты (сущности ПрО) и атрибуты сущностей – данные, которые должны храниться в БД.

Из общения со специалистами необходимо извлечь сведения об особенностях ПрО, которые позволяют установить ограничения целостности, зависимости и связи между объектами (субъектами) предметной области. Также специалисты обладают знаниями о том, каковы алгоритмы обработки данных и какие задачи ставятся перед информационной системой.

3.1.2 Постановка задачи

Необходимо отразить следующие моменты:

– основные объекты предметной области, информация о которых должна храниться в базе данных;

– связи между объектами;

– группы пользователей разрабатываемой базы данных (например, работник, начальник отдела, администратор и т.п.), набор операций и выводимых данных для каждой группы;

– требования к функциональности системы с точки зрения каждой из групп пользователей.

Постановка задачи является основой для дальнейшего проектирования. Очевидно, что детальное описание задачи и формулирование требований существенно облегчает дальнейшую разработку базы данных.

3.2 Разработка логической модели базы данных

Логическая модель позволяет понять суть проектируемой системы, отражая логические взаимосвязи между сущностями.

Различают три уровня логической модели, отличающихся по глубине представления информации о данных:

– диаграмма сущность-связь (Entity Relationship Diagram, ERD);

– модель данных, основанная на ключах (Key Based model, KB);
– полная атрибутивная модель (Fully Attributed model, FA).

Диаграмма сущность-связь представляет собой модель данных верхнего уровня. Она включает сущности и взаимосвязи, отражающие основные бизнес-правила предметной области. Такая диаграмма не слишком детализирована, в нее включаются основные сущности и связи между ними, которые удовлетворяют основным требованиям, предъявляемым к ИС. Диаграмма сущность-связь может включать связи «многие-ко-многим» и не включать описание ключей.

Как правило, ERD используется для презентаций и обсуждения структуры данных с экспертами предметной области.

Модель данных, основанная на ключах – более подробное представление данных. Она включает описание всех сущностей и первичных ключей и предназначена для представления структуры данных и ключей, которые соответствуют предметной области.

Полная атрибутивная модель – наиболее детальное представление структуры данных: представляет данные в третьей нормальной форме и включает все сущности, атрибуты и связи (IDEF1X).

3.2.1 Построение диаграммы «сущность-связь» в нотации П.Чена

Диаграммы «сущность-связь» – ERD (Entity-Relationship Diagrams) (case-метод Баркера) являются наиболее распространенной методологией моделирования данных. С их помощью определяются важные для предметной области объекты (сущности), их свойства (атрибуты) и отношения друг с другом (связи). ERD непосредственно используются для проектирования реляционных баз данных.

Нотация ERD была впервые введена П.Ченом (Chen) и получила дальнейшее развитие в работах Баркера [1].

ERD-диаграмма, рис.1, позволяет рассмотреть систему целиком и выяснить требования, необходимые для ее разработки, касающиеся хранения информации.

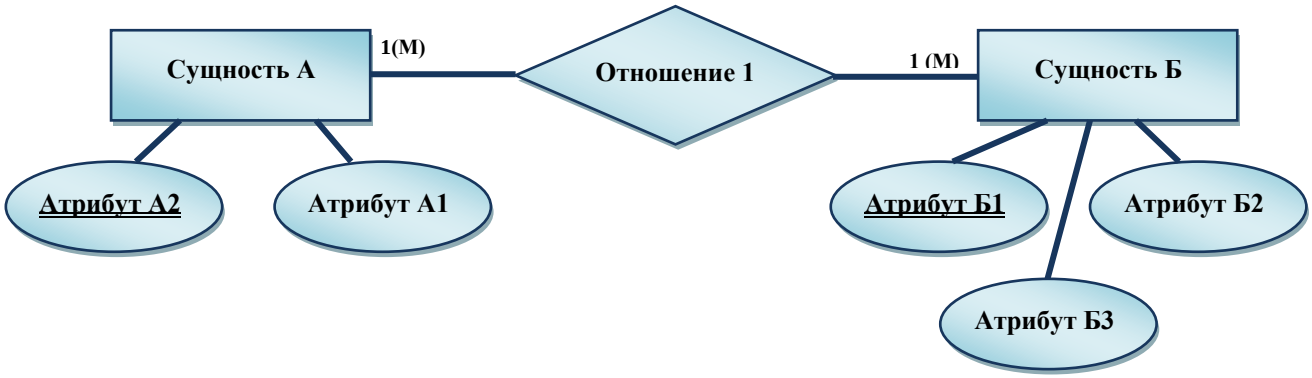


Рисунок 1 – Пример ER-диаграммы в нотации П.Чена

Существуют и другие нотации для представления ER-диаграмм, например нотация Мартина, нотация Баркера.

Базовыми понятиями ERD являются:

Сущность (Entity) – множество экземпляров реальных или абстрактных объектов (людей, событий, состояний, идей, предметов и др.), обладающих

общими атрибутами или характеристиками. Любой объект системы может быть представлен только одной сущностью, которая должна быть уникально идентифицирована. При этом имя сущности должно отражать тип или класс объекта, а не его конкретный экземпляр (например, ГРАЖДАНИН, а не ПЕТРОВ);

Связь (Relationship) – поименованная ассоциация между двумя сущностями, значимая для рассматриваемой предметной области. Связь – это ассоциация между сущностями, при которой каждый экземпляр одной сущности ассоциирован с произвольным (в том числе нулевым) количеством экземпляров второй сущности, и наоборот.

Каждая связь может иметь один из следующих типов:

– связь типа один-к-одному (1:1) означает, что один экземпляр первой сущности (левой) связан с одним экземпляром второй сущности (правой). Связь один-к-одному чаще всего свидетельствует о том, что на самом деле имеется всего одна сущность, неправильно разделенная на две;

– связь типа один-ко-многим означает, что один экземпляр первой сущности (левой) связан с несколькими экземплярами второй сущности (правой). Это наиболее часто используемый тип связи. Левая сущность (со стороны «один») называется родительской, правая (со стороны «много») – дочерней (1:***n***);

– связь типа много-ко-многим означает, что каждый экземпляр первой сущности может быть связан с несколькими экземплярами второй сущности, и каждый экземпляр второй сущности может быть связан с несколькими экземплярами первой сущности. Тип связи много-ко-многим является временным типом связи, допустимым на ранних этапах разработки модели. В дальнейшем этот тип связи должен быть заменен двумя связями типа один-ко-многим путем создания промежуточной сущности (***m***:***n***).

Каждая связь может иметь одну из двух модальностей связи:

– модальность «может» означает, что экземпляр одной сущности может быть связан с одним или несколькими экземплярами другой сущности, а может быть, и не связан ни с одним экземпляром (например, каждый пользователь может подать несколько заявок);

– модальность «должен» означает, что экземпляр одной сущности обязан быть связан не менее чем с одним экземпляром другой сущности (например, логин должен принадлежать только одному пользователю).

Связь может иметь разную модальность с разных концов.

Атрибут (Attribute) – любая характеристика сущности, значимая для рассматриваемой предметной области и предназначенная для квалификации, идентификации, классификации, количественной характеристики или выражения состояния сущности. Атрибут представляет тип характеристик или свойств, ассоциированных с множеством реальных или абстрактных объектов (людей, мест, событий, состояний, идей, предметов и т.д.). Экземпляр атрибута – это определенная характеристика отдельного элемента множества. Экземпляр атрибута определяется типом характеристики и ее значением, называемым значением атрибута. На диаграмме «сущность-связь» атрибуты ассоциируются с

конкретными сущностями. Таким образом, экземпляр сущности должен обладать единственным определенным значением для ассоциированного атрибута.

Атрибут или группа атрибутов, однозначно идентифицирующая экземпляр сущности – называется *первичным ключом (PrimaryKey)*.

При выборе первичного ключа можно внести в сущность дополнительный атрибут и сделать его ключом. Так, для определения первичного ключа часто используют уникальные номера, которые могут автоматически генерироваться системой при добавлении экземпляра сущности в базы данных.

На рис.2. представлены графические элементы ERD диаграмм в нотации П.Чена.

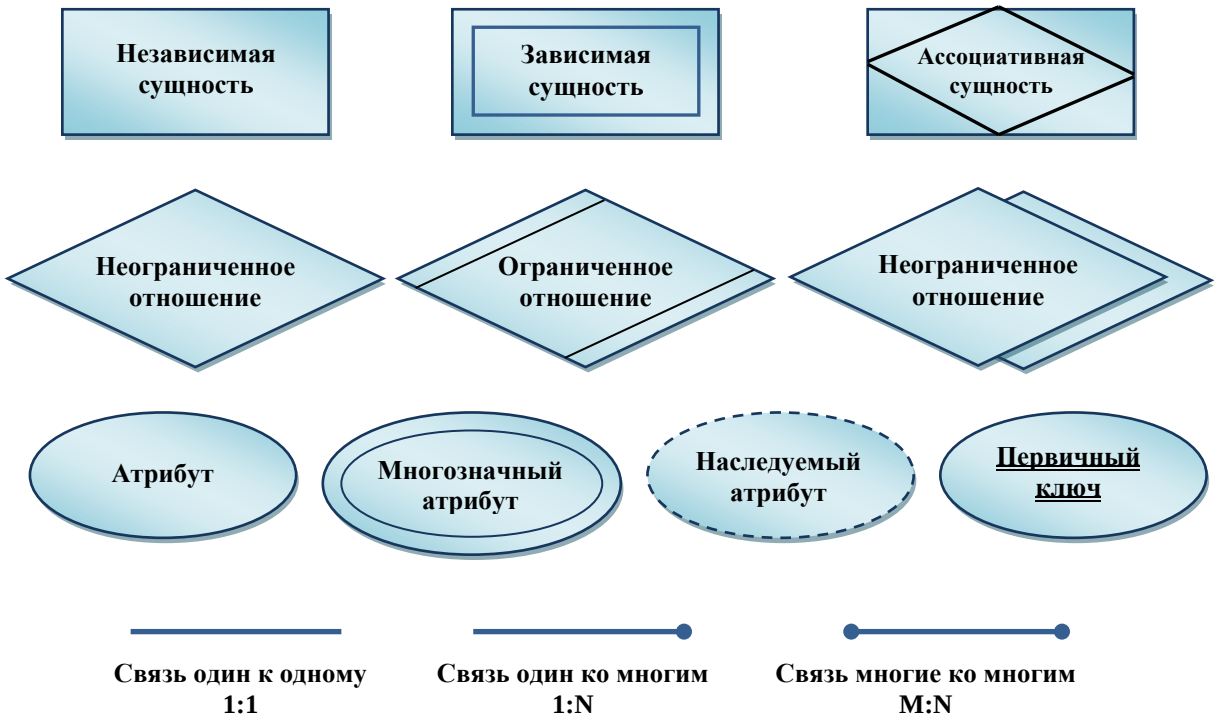


Рисунок 2 – Символы ERD, соответствующие сущностям, атрибутам, отношениям и связям в нотации П.Чена

Этапы построения ERD-диаграммы.

На *первом шаге* производится определение сущностей. Для построения ER-диаграммы «с нуля» производиться анализ предметной области и выделяются информационные объекты проектируемой системы, другими словами составляется список (пул) потенциальных сущностей (как правило, выделяются все существительные в текстовом описании предметной области).

На *втором шаге* необходимо описать каждый информационный объект набором характеристик (атрибутов), которые представляют важность с точки зрения выполняемых системой функций, то есть из списка потенциальных сущностей выделяются сущности, а остальное преобразуется в атрибуты сущностей.

На *третьем шаге* устанавливаются отношения и связи между сущностями по описанию предметной области на естественном языке. Определяются виды отношений и типы связей.

На *четвертом шаге* из списка атрибутов выделить атрибуты, способные однозначно идентифицировать экземпляры сущности, то есть определить первичные ключи.

На *пятом шаге* строится ER-диаграмма.

Правила и рекомендации для построения ERD-диаграммы

Каждая сущность должна обладать уникальным идентификатором. Каждый экземпляр сущности должен однозначно идентифицироваться и отличаться от всех других экземпляров данного типа сущности. Каждая сущность должна обладать некоторыми свойствами:

– иметь уникальное имя; к одному и тому же имени должна всегда применяться одна и та же интерпретация; одна и та же интерпретация не может применяться к различным именам, если только они не являются псевдонимами;

– иметь один или несколько атрибутов, которые либо принадлежат сущности, либо наследуются через связь;

– иметь один или несколько атрибутов, которые однозначно идентифицируют каждый экземпляр сущности *первичный ключ (PrimaryKey)*.

Каждая сущность может обладать любым количеством связей с другими сущностями модели.

3.2.2 Построение модели основанной на ключах (Key Based model, KB) и полной атрибутивной модели в нотации IDEF1X (нормализация отношений до третьей или четвѐртой нормальной формы)

Модель, основанная на ключах (Key Based model, KB) описывает основные структуры данных, которые охватывают обширные области ПрО. Все сущности и первичные ключи включены вместе с примерами атрибутов.

Основной целью модели, основанной на ключах, является широкий обзор структур данных и ключей, нужных для поддержки определенной области. Эта модель определяет контекст, в котором могут быть созданы подробные модели, пригодные для конкретного воплощения.

Модель показывает ту же область что и область ERD, но, вместе с тем, отображает больше деталей (см. рисунок 2).

Методология IDEF1 [2] разработанная Т.Рэмеем, основана на подходе П.Чена и позволяет построить модель данных, эквивалентную реляционной модели в третьей нормальной форме.

Методология IDEF1 позволяет на основе наглядных графических представлений моделировать информационные взаимосвязи и различия между:

– реальными объектами;

– физическими и абстрактными зависимостями, существующими среди реальных объектов;

– информацией о реальных объектах;

– структурой данных, используемой для приобретения, накопления и управления информацией.

IDEF1X – предназначена для построения концептуальной схемы логической структуры реляционной базы данных, которая была бы независимой от программной платформы её конечной реализации. По сравнению с IDEF1 она

дополнительно оперирует рядом понятий, правил и ограничений, такими как домены, представления, первичные, внешние и суррогатные ключи и другими, пришедшими из реляционной алгебры [3,4].

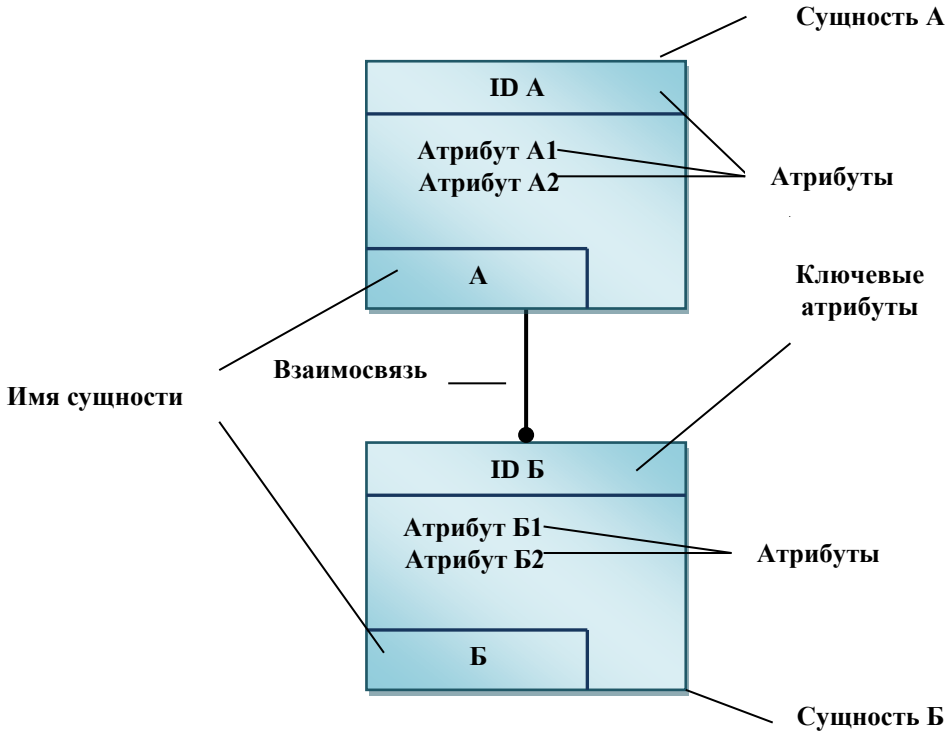


Рисунок 2 – Стандарт IDEF1

Использование метода IDEF1X наиболее целесообразно для построения логической структуры базы данных после того как все информационные ресурсы исследованы и решение о внедрении реляционной базы данных, как части корпоративной информационной системы, было принято.

Концепция и семантика IDEF1X.

Сущность в IDEF1X описывает собой совокупность или набор экземпляров похожих по свойствам, но однозначно отличаемых друг от друга по одному или нескольким признакам. Каждый экземпляр является реализацией сущности. Таким образом, сущность в IDEF1X описывает конкретный набор экземпляров реального мира, в отличие от сущности в IDEF1, которая представляет собой абстрактный набор информационных отображений реального мира.

Классификация сущностей в IDEF1X. При разработке модели, зачастую, приходится сталкиваться с сущностями, уникальность которых зависит от значений атрибута внешнего ключа. Для этих сущностей (для уникального определения каждой сущности) внешний ключ должен быть частью первичного ключа дочернего объекта.

Дочерняя сущность, уникальность которой зависит от атрибута внешнего ключа, называется зависимой сущностью.

Зависимые сущности далее классифицируются на сущности, которые не могут существовать без родительской сущности и сущности, которые не могут быть идентифицированы без использования ключа родителя (сущности, зависящие от идентификации).

Сущности, независящие при идентификации от других объектов в модели, называются независимыми сущностями.

Связи между сущностями в IDEF1X представляют собой ссылки, соединения и ассоциации между сущностями. Связи – это суть глаголы, которые показывают, как соотносятся сущности между собой.

Набор атрибутов, выбранных для идентификации уникальных экземпляров сущности – это первичный ключ.

Выбор первичного ключа для сущности является очень важным шагом, и требует большого внимания. В качестве первичных ключей могут быть использованы несколько атрибутов или групп атрибутов.

В IDEF1X концепция зависимых и независимых сущностей усиливается типом взаимосвязей между двумя сущностями. Если требуется, чтобы внешний ключ передавался в дочернюю сущность (и, в результате, создавал зависимую сущность), то необходимо создать идентифицирующую связь между родительской и дочерней сущностью.

Неидентифицирующие связи, являющиеся уникальными для IDEF1X, также связывают родительскую сущность с дочерней. Неидентифицирующие связи используются для отображения другого типа передачи атрибутов внешних ключей – передача в область данных дочерней сущности (под линией).

Основным *преимуществом IDEF1X*, по сравнению с другими многочисленными методами разработки реляционных баз данных, такими как ER является жесткая и строгая стандартизация моделирования. Установленные стандарты позволяют избежать различной трактовки построенной модели, которая, несомненно, является значительным недостатком ERD.

Итак, центральным понятием IDEF1 является понятие «сущность», рис.2.

Каждая сущность имеет своё имя и атрибуты.

Под связями в IDEF1 понимаются ссылки, соединения и ассоциации между сущностями.

Каждый объект разделяется горизонтальной линией на часть, в которой расположены ключевые поля и часть, где расположены не ключевые поля.

Ключевая область содержит первичный ключ для сущности. Первичный ключ – это набор атрибутов, выбранных для идентификации уникальных экземпляров сущности. Первичный ключ может включать в себя один или несколько атрибутов первичного ключа, составляющих уникальный идентификатор для каждой записи в сущности.Атрибуты первичного ключа располагаются над линией в ключевой области. Как следует из названия, не ключевой атрибут – это атрибут, который не был выбран ключевым. Не ключевые атрибуты располагаются под чертой в области данных.

При создании сущности в модели данных, одним из главных вопросов, на который нужно ответить, является: «Как можно идентифицировать уникальную запись?». Требуется уникальная идентификация каждой записи в сущности для того, чтобы правильно создать логическую модель данных.

В качестве первичных ключей могут быть использованы несколько атрибутов или наборов атрибутов. Атрибуты или группы атрибутов, которые могут быть выбраны первичными ключами, называются кандидатами в ключевые

атрибуты (потенциальные атрибуты). Кандидаты в ключи должны уникально идентифицировать каждую запись сущности. В соответствии с этим, ни одна из частей ключа не может быть NULL, не заполненной или отсутствующей. Пользователь, чаще всего, лучше, чем кто-либо другой находит кандидаты в ключи, так как лучше всех разбирается в бизнесе и данных, использующихся в его области деятельности.

Полная атрибутивная модель достигается нормализацией отношений до третьей или четвѐртой нормальной формы. Для проведения нормализации целесообразно записать информационную модель в виде отношений (по рис.1):

СущностьА (АтрибутА2, АтрибутА1)

СущностьБ (АтрибутБ1, АтрибутБ2, АтрибутБ3)

Целью нормализации базы данных является сокращение избыточности. Нормализация предполагает последовательное приведение схемы базы данных к так называемым нормальным формам, каждая последующая из которых предъявляет более строгие требования по сравнению с предыдущей.

Для описания нормальных форм требуются следующие определения:

– **функциональная зависимость** в отношении *R*: атрибут *Y* функционально зависит от атрибута *X* (*X* и *Y* могут быть составными) в том и только в том случае, если каждому значению *X* соответствует в точности одно значение *Y*: *R.X → R.Y*. Левая часть функциональной зависимости называется *детерминантом*.

– **полная (неприводимая) функциональная зависимость**: функциональная зависимость *R.X → R.Y* называется *полной*, если атрибут *Y* не зависит функционально от любого точного подмножества *X*.

– **транзитивная функциональная зависимость**: функциональная зависимость *R.X -> R.Y* называется *транзитивной*, если существует такой атрибут *Z*, что имеются функциональные зависимости *R.X→R.Z* и *R.Z → R.Y* и отсутствует функциональная зависимость *R.Z→R.X*.

– **многозначные зависимости** в отношении *R(A, B, C)* существует многозначная зависимость *R.A ->> R.B* в том и только в том случае, если множество значений *B*, соответствующее паре значений *A* и *C*, зависит только от *A* и не зависит от *C*.

Условия нахождения отношений в нормальных формах:

1НФ. Отношение *R* находится в первой нормальной форме, если выполняется условие атомарности значений атрибутов отношения.

2НФ. Отношение *R* находится во второй нормальной форме в том и только в том случае, когда находится в 1НФ, и каждый неключевой атрибут полностью зависит от первичного ключа.

3НФ. Отношение *R* находится в третьей нормальной форме в том и только в том случае, если находится в 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

В приведенных определениях 2НФ и 3НФ предполагается, что отношение имеет только один возможный ключ, который и является его первичным ключом.

Такое определение 3НФ оказывается несостоятельным при условии, что:

1.Отношение имеет два или более возможных ключа.

2.Эти потенциальные ключи являются составными.

3.Два или более потенциальных ключа перекрываются, т.е. имеют общие атрибуты.

Поэтому вводят понятие нормальной формы Бойса-Кодда, или усиленной 3НФ.

БКНФ. Отношение *R* находится в нормальной форме Бойса-Кодда в том и только в том случае, если каждый детерминант является возможным ключом.

Примечание: При нарушении одного из указанных выше условий 3НФ и БКНФ совпадают.

4НФ. Отношение *R* находится в четвертой нормальной форме в том и только в том случае, если в случае существования многозначной зависимости *A ->> B* все остальные атрибуты отношения *R* функционально зависят от *A*.

Теорема Фейджина. Отношение *R(A, B, C)* можно спроецировать без потерь в отношения *R1(A, B)* и *R2(A, C)* в том и только в том случае, когда существует многозначная зависимость *A ->> B | C*.

Пример. Дана реляционная схема, включающая следующие отношения:

Отдел(Н_Отд, Размер бюджета, Н_руководителя)

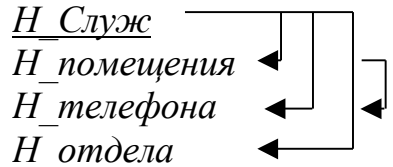
Служащий(Н_Служ, Н_помещения, Н_телефона, Н_отдела)

Помещение(Н_помещения, Площадь, Н_телефона)

1НФ. Все выделенные атрибуты атомарны, следовательно, схема находится в 1НФ.

2НФ. Все неключевые атрибуты зависят полностью от ключей, следовательно, схема находится в 2НФ.

3НФ.



Из зависимостей между атрибутами отношения *Служащий* видно, что имеется транзитивная зависимость:

Н_Служ →Н_помещения→Н_телефона.

Для устранения транзитивных зависимостей выполняется проекция, в результате получаются 2 отношения:

(Н_служащего, Н_помещения, Н_отдела)

(Н_помещения, Н_телефона)

Второе из них входит в состав отношения *Помещение*, поэтому его можно удалить. В результате получается схема:

Отдел(Н_Отд, Размер бюджета, Н_руководителя),

Служащий(Н_Служ, Н_помещения, Н_отдела),

Помещение(Н_помещения, Площадь, Н_телефона),

которая находится также в БКНФ и 4НФ.

БКНФ. Пусть есть отношение вида:

Служащие-Проекты(Н_служащего, Имя_служащего, Н_проекта, Работа_Служащего).

Предполагается, что служащий, работая над проектом с номером *Н_проекта*, выполняет работу *Работа_Служащего*.

Возможными ключами отношения являются следующие пары атрибутов:

Н_служащего, Н_проекта;

Имя_служащего, Н_проекта.

В отношении есть следующие функциональные зависимости:

Н_служащего→Имя_служащего;

Н_служащего→Н_проекта;

Имя_служащего→Н_служащего;

Имя_служащего→Н_проекта;

Н_служащего, Н_проекта →Работа_Служащего;

Имя_служащего, Н_проекта →Работа_Служащего.

Предполагается, что *Имя_служащего* так же уникально, как и *Н_служащего*. Видно, что отношение *Служащие-Проекты* находится в 3НФ.

Но тот факт, что имеются функциональные зависимости атрибутов отношения от атрибута, являющегося частью первичного ключа, приводит к аномалиям. Например, для того, чтобы изменить имя сотрудника с данным номером согласованным образом, потребуется модифицировать все кортежи, включающие его номер.

Если вынести связь *Н_служащего→Имя_Служащего* в отдельное отношение, то получится два отношения:

1.*Служащие(Н_Служащего, Имя_служащего)*.

Возможные ключи:

Н_Служащего,

Имя_служащего.

Зависимости:

Н_Служащего→Имя_служащего;

Имя_служащего→Н_Служащего.

2.*Служащие_Проекты(Н_служащего, Н_проекта, Работа_Служащего)*.

Возможный ключ: *Н_служащего, Н_проекта*.

Зависимости: *(Н_служащего, Н_проекта)→Работа_Служащего*.

Такая схема находится в БКНФ.

4НФ Пусть дано отношение:

Проекты(Н_проекта, Н_Служащего, Задание).

Отношение *Проекты* содержит номера проектов, для каждого проекта – список служащих, которые могут выполнять проект, и список заданий, предусматриваемых проектом. Служащие могут участвовать в нескольких проектах, и разные проекты могут включать одинаковые задания.

Каждый кортеж отношения связывает некоторый проект со служащим, участвующим в этом проекте, и заданием, который служащий выполняет в рамках данного проекта (предполагается, что любой служащий, участвующий в проекте, выполняет все задания, предусмотренные этим проектом).

Единственным возможным ключем отношения является комбинация атрибутов *(Н_проекта, Н_Служащего, Задание)*, и нет никаких других детерминантов. Т.е., отношение находится в БКНФ. Но при этом оно обладает

недостатками: если, например, некоторый сотрудник присоединяется к данному проекту, необходимо вставить в отношение *Проекты* столько кортежей, сколько заданий в нем предусмотрено.

Видно, что в отношении существуют следующие многозначные зависимости:

Н_проекта->>Н_служащего;

Н_проекта->>Задание.

Для приведения отношения к 4НФ многозначные зависимости выносятся в отдельные отношения. В результате получаем 2 отношения:

Проекты_Служащие(Н_проекта, Н_служащего);

Проекты_Задания(Н_проекта, Н_задания).

Оба отношения находятся в 4НФ.

Если все первичные ключи простые, то БД, находящаяся в 3НФ находится и в БКНФ и в 4НФ.

3.3 Разработка физической модели базы данных

Физическая модель отражает физические свойства проектируемой базы данных (типы данных, размер полей, индексы). Параметры физической информационной модели зависят от выбранной системы управления базами данных (СУБД).

3.3.1 Выбор аппаратной и программной платформы и реализация БД

Разработанную логическую схему базы данных реализуют с использованием одной из СУБД. Выбор СУБД выполняется на основе сравнительного анализа нескольких вариантов (если СУБД не обозначена в варианте). В пояснительной записке необходимо привести обоснование выбора СУБД.

Для реализации на конкретной СУБД на основе логической схемы строится физическая схема базы данных, в которой вместо названий атрибутов указываются реальные имена столбцов создаваемых таблиц и тип данных, используемый для каждого из столбцов. Тип данных для каждого атрибута (столбца таблицы) выбирается из числа поддерживаемых используемой СУБД, выбор обосновывается.

Полученная БД заполняется осмысленными тестовыми данными из соответствующей предметной области.

3.3.2 Тестирование базы данных (создание и реализация запросов)

Для проверки адекватности спроектированной базы данных необхдимо построить не менее 15 разноплановых запросов.

Тестирование базы данных включает следующие этапы:

– проверка правильности выполнения всех разработанных SQL-запросов к базе данных;

– тестирование ссылочной целостности базы данных, каскадирования операций удаления и модификации атрибутов, являющихся внешними ключами;

– тестирование разграничения прав доступа – проверка возможности выполнения пользователем всех операций, предусмотренных его правами, и невозможности выполнить действия, превышающие его права.

По каждому этапу тестирования составляются и выполняются тестовые запросы, результаты тестирования документируются и приводятся в пояснительной записке.

3.3.3 Разграничение прав доступа

Необходимо предусмотреть возможность работы с базой данных групп пользователей с различным уровнем доступа. Основываясь на постановке задачи, необходимо выделить основные роли пользователей БД, составить таблицу прав каждой из ролей. Затем написать SQL-запросы, реализующие назначение прав доступа к базе данных.

В большинстве СУБД такая возможность реализована на основе т.н. ролей. Роль – это группа пользователей базы данных, имеющих одинаковые права доступа. При соединении с базой данных каждый пользователь вводит логин, и ему предоставляются права доступа в соответствии с его ролью. При попытке выполнить непозволенные действия СУБД генерирует исключение.

Для создания роли используется оператор `CREATE ROLE`, например:

```
CREATE ROLE guests.
```

Для назначения привилегии и присвоения пользователю роли используется оператор `GRANT`, имеющий следующий формат (упрощенно):

```
GRANT <тип_привелегии>[, <тип_привелегии> ]|<имя_роли>
ON <объект_БД>
TO <имя_пользователя>|<имя_роли>
[WITH GRANT OPTION]
<тип_привелегии>=ALL|INSERT|UPDATE|DELETE|SELECT|EXECUTE
<объект_БД>=<имя_таблицы>[.<имя_столбца>]|<имя_процедуры>
```

Например, команда

```
GRANT SELECT ON Orders TO guests
```

разрешает пользователям, имеющим роль `guests`, выполнять операцию выборки из таблицы `Orders`.

Команда

```
GRANT guests TO ivanov,petrov;
```

присвоит роль `guests` пользователям `ivanov` и `petrov`.

Для того, чтобы забрать привилегии, используется оператор `REVOKE`:

```
REVOKE  <тип_привелегии>[, <тип_привелегии> ]
ON <объект_БД>
FROM <имя_пользователя>|<имя_роли>
```

Например,

```
REVOKE guests FROM ivanov
```

забирает у пользователя `ivanov` роль `guests`.

В случае, если в СУБД не реализован механизм ролей, права доступа каждому пользователю назначаются индивидуально.

3.4 Разработка клиентского приложения

Диалоговое приложение пользователя (прикладная программа) объединяет всю технологию обработки данных ПО, включая загрузку, ведение БД и решение всего комплекса задач.

Программа должна иметь «дружественный» интерфейс и содержать набор команд решения конкретных задач:

– добавление, удаление, изменение, просмотр данных;

– сортировка данных;

– выбор данных по запросам;

– формирование выходных документов (отчётов) с возможностью просмотра на экране и вывода на печать.

Интерфейс пользователя должен строиться на основе иерархических меню с использованием диалоговых окон, кнопок, «горячих клавиш» и т.п.

Приложение должно обеспечивать просмотр всех данных БД и выполнение всех необходимых запросов, удовлетворяющих требованиям технического задания (соответствующих описанию предметной области).

Разрабатываемое программное приложение должно позволять:

- заносить информацию в созданную базу данных;
- выполнять необходимые действия по модификации и удалению информации в базе данных, причем все операции по занесению, модификации и удалению данных должны выполняться в терминах предметной области, а не базы данных;
- поддерживать целостность базы данных, не допуская появления некорректных данных;
- выполнять все действия над базой данных в рамках транзакций;
- содержать достаточное количество данных, позволяющих показать результаты выполнения запросов;
- контролировать все вводимые данные.

Язык программирования выбирается студентом самостоятельно, в пояснительной записке приводится обоснование выбора. Рекомендуется приложение реализовать в виде двух подсистем: подсистемы администратора и подсистемы пользователя.

3.4.1 Подсистема пользователя

Данная подсистема предоставляет пользователю удобный интерфейс для просмотра необходимой информации, добавления и модификации данных под управлением выбранной СУБД. Необходимо разработать интуитивно понятный интерфейс, ориентированный на неподготовленного пользователя. Для выполнения наиболее часто используемых операций рекомендуется назначить т.н. «горячие клавиши».

Система должна предусматривать возможность работы пользователей с различным уровнем доступа к базе данных. При попытке пользователя выполнить

операцию, превышающую его права доступа, СУБД генерирует исключение, которое необходимо обрабатывать (например, блок try – catch в C++) и выводить на экран соответствующие сообщения. Поскольку подсистема рассчитана на неподготовленного пользователя, особое внимание рекомендуется обратить на обработку возможных ошибок и исключительных ситуаций.

3.4.2 Подсистема администратора

Подсистема администратора, кроме операций, реализованных в подсистеме пользователя, должна предоставлять возможности, связанные с администрированием базы данных: создание и удаление пользователей, просмотр и изменение информации о пользователях, управление правами пользователей, вывод статистики использования базы данных и т.п. Подсистему администратора рекомендуется реализовать в виде отдельного приложения.

3.4.3 Тестирование системы

Целью тестирования является проверка работоспособности системы, адекватность работы каждого из модулей. Учитывая сложность разрабатываемой системы, тестирование рекомендуется проводить в несколько этапов следующим образом:

1) Тестирование подсистем клиентского приложения. Проверяется работоспособность каждого модуля в отдельности до сборки системы.

2) Тестирование завершенной системы. Необходимо для выявления ошибок при взаимодействии между отдельными модулями.

4 ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ К РГР

Пояснительная записка оформляется согласно правилам оформления принятыми на кафедре, ГОСТам и ЕСКД.

Основные правила по оформлению отчетной документации:

Параметры страницы: А4 (21×29,7), ориентация – книжная (допускается использовать альбомную ориентацию страницы для выполнения схем и таблиц).

Поля: левое – 3, верхнее – 2, нижнее – 2, правое – 1.

Нумерация страницы – вверху, по центру. Нумерация ведется с титульного листа, номер на титульном листе не ставиться.

Шрифт Times New Roman, кегль 14, интервал – полуторный.

Заголовки разделов: абзацный отступ – 0, выравнивание по центру, шрифт – жирный, буквы прописные, нумерация – арабскими цифрами, точка в конце номера и названия раздела не ставиться.

Заголовки подразделов (допускается три уровня, например, 1.1., 1.1.1.): абзацный отступ – 1.25, выравнивание по ширине, шрифт – жирный, точка в конце названия подраздела не ставиться.

Основной текст: абзацный отступ – 1.25, выравнивание по ширине, шрифт – обычный.

Нумерация рисунков и таблиц – сквозная внутри раздела (например, в разделе 1 – рисунок 1.1, рисунок 1.2 и т.д., или таблица 1.1, таблица 1.2 и т.д.).

Рисунки помещаются после упоминания их в тексте и имеют подпись, размещаемую под рисунком без абзацного отступа и имеющую выравнивание по центру и точку на конце названия (например, Рисунок 1.1 – Название.).

Таблицы размещаются после ссылки на них в тексте. Название приводится над таблицей, без абзацного отступа с выравниванием по левому краю, без точки на конце названия (например, (Таблица 2.2 – Название).

Допускается выносить рисунки и таблицы в Приложения. В этом случае ссылка должна содержать номер приложения (например: рисунок А.1 приложения А или таблица Б.1 приложения Б).

Основная часть должна содержать ссылки на используемую литературу или информационные источники, список которых приводится после раздела Выводы и перед Приложениями. Ссылка заключается в квадратные скобки (например – [1], [5,7], [3-6].

Приложения обозначаются русскими заглавными буквами в порядке их следования (приложение А, приложение Б). Слово «Приложение…» выравнивается по центру без абзацного отступа и имеет жирный шрифт, прописные буквы. Название приложение располагается на следующей строке, без абзацного отступа, выравнивание по центру, шрифт – жирный, первая буква прописная, остальные – строчные.

5 ОРГАНИЗАЦИЯ ЗАЩИТЫ И КРИТЕРИИ ОЦЕНИВАНИЯ РГР

Защита РГР осуществляется в присутствии комиссии в состав которой входят: ведущий преподаватель и преподаватели, осуществляющие руководство РГР по дисциплине «Управление данными».

График защиты проектов доводится до студентов не позднее, чем за месяц до её начала.

К защите представляется пояснительная записка и электронная презентация, содержащая основные этапы выполнения расчетно-графической работы (постановка задачи; описание предметной области; этапы проектирования и разработки базы данных)

К пояснительной записке прилагается электронный носитель, содержащий папки с файлами базы данных, электронная презентация и пояснительная записка в формате .pdf или .doc.

Защита осуществляется в форме доклада. Регламент защиты проектов 5-6 минут. Для ответа на вопросы и замечания по курсовому проекту выделяется до 5 минут.

Степень усвоения теоретического материала оценивается по следующим критериям:

- ***оценка «отлично» выставляется, если:***
  - последовательно, четко, связно, обоснованно и безошибочно с использованием принятой терминологии изложен ход и результаты работы, выделены главные положения, ответ подтвержден конкретными примерами, фактами;
  - самостоятельно и аргументировано сделан анализ, обобщение, выводы, установлены межпредметные (на основе ранее приобретенных знаний) и внутрипредметные связи, творчески применены полученные знания в незнакомой ситуации;
  - самостоятельно и рационально используются справочные материалы, учебники, дополнительная литература, первоисточники; применяется систему условных обозначений при ведении записей, сопровождающих ответ; используются для доказательства выводы из наблюдений и опытов, ответ подтверждается конкретными примерами;
  - допускается не более одного недочета, который легко исправляется по требованию преподавателя.
- ***оценка «хорошо» ставится, если:***
  - представлено полное описание хода выполнения работы; допущены незначительные ошибки и недочеты при, определении понятий, неточности при использовании научных терминов или в выводах и обобщениях из наблюдений и опытов; материал излагает в определенной логической последовательности;
  - самостоятельно выделены главные положения в представляемых результатах; на основании фактов и примеров проведено обобщение, сделаны выводы, установлены внутрипредметные связи.

- допущены одна негрубая ошибку или не более двух недочетов, которые исправлены самостоятельно при требовании или при небольшой помощи преподавателя.

- ***оценка «удовлетворительно» ставится, если:***
  - показана недостаточная сформированность отдельных знаний и умений; выводы и обобщения аргументируются слабо, в них допускаются ошибки;
  - допущены ошибки и неточности в использовании научной терминологии, даются недостаточно четкие определения понятий; в качестве доказательства не используются выводы и обобщения из результатов работы, фактов, опытов или допущены ошибки при их изложении;
  - допускаются неполные ответы на вопросы преподавателя, с допущением одной – двух грубых ошибок.
- ***оценка «неудовлетворительно» ставится, если:***
  - не выполнено значительная или основная часть задания на курсовую работу;
  - при ответе (на один вопрос) допускается более двух грубых ошибок, которые не могут быть исправлены даже при помощи преподавателя;
  - не даются ответы ни на один их поставленных вопросов.

Оценка сформированности практических навыков проводится по следующим критериям

- ***оценка «отлично» ставится, если студент:***
  - творчески планирует выполнение работы;
  - самостоятельно и полностью использует знания программного материала;
  - правильно и аккуратно выполняет задание;
  - умеет пользоваться литературой и различными информационными источниками;
  - выполнил работу без ошибок и недочетов или допустил не более одного недочета
- ***оценка «хорошо» ставится, если студент:***
  - правильно планирует выполнение работы;
  - самостоятельно использует знания программного материала;
  - в основном правильно и аккуратно выполняет задание;
  - умеет пользоваться литературой и различными информационными источниками;
  - выполнил работу полностью, но допустил в ней: не более одной негрубой ошибки и одного недочета или не более двух недочетов.
- ***оценка «удовлетворительно» ставится, если студент:***
  - допускает ошибки при планировании выполнения работы;
  - не может самостоятельно использовать значительную часть знаний программного материала;
  - допускает ошибки и неаккуратно выполняет задание;

- затрудняется самостоятельно использовать литературу и информационные источники;
- правильно выполнил не менее половины работы или допустил:
  - не более двух грубых ошибок или не более одной грубой и одной негрубой ошибки и одного недочета;
  - не более двух- трех негрубых ошибок или одной негрубой ошибки и трех недочетов;
  - при отсутствии ошибок, но при наличии четырех-пяти недочетов.

- ***оценка «неудовлетворительно» ставится, если студент:***
  - не может правильно спланировать выполнение работы;
  - не может использовать знания программного материала;
  - допускает грубые ошибки и неаккуратно выполняет задание;
  - не может самостоятельно использовать литературу и информационные источники;
  - допустил число ошибок недочетов, превышающее норму, при которой может быть выставлена оценка «3»;
  - если правильно выполнил менее половины работы;
  - не приступил к выполнению работы;
  - правильно выполнил не более 10% всех заданий.

6 СПИСОК ЛИТЕРАТУРЫ И ИНФОРМАЦИОННЫХ РЕСУРСОВ

1. Barker R. CASE-Method. Entity-Relationship Modelling. Copyright Oracle Corporation UK Limited, Addison-Wesley Publishing Co., 1990.

2. IDEF1 Information Modeling Method. Описание. Режим доступа – http://www.idef.com/IDEF1.htm.

3. IDEF1X Data Modeling Method. Описание. Режим доступа – http://www.idef.com/IDEF1x.htm.

4. Integration Definition For Information Modeling (IDEF1X). Режим доступа – http://www.idef.ru/documents/Idef1x.pdf.

ПРИЛОЖЕНИЕ А
Варианты заданий к РГР

Вариант предметной области определяется по второй цифре номера группы и порядковому номеру в журнале группы из таблиц А.1- А.5.

Замечание: *студент вправе, предложить свой вариант предметной области и принять его к исполнению, после согласования и утверждения ведущим преподавателем*.

Таблица А.1 – Варианты предметных областей для группы №1

№ группы потока	№ в классном журнале	Предметная область
1	1.	База данных абонентского отдела интернет-провайдера
1	2.	База данных агентства по продаже авиабилетов
1	3.	База данных горноспасательной службы
1	4.	База данных деканата института (документооборот)
1	5.	База данных детского оздоровительного лагеря
1	6.	База данных диспетчерской службы радиотакси
1	7.	База данных минимаркета (движение товара)
1	8.	База данных мини-отеля
1	9.	База данных морского порта (организация пассажирских перевозок)
1	10.	База данных нотариальной конторы
1	11.	База данных оптового склада компьютерной техники
1	12.	База данных оптовый склад продуктов питания
1	13.	База данных отдела кадров университета
1	14.	База данных районной поликлиники (регистратура)
1	15.	База данных реставрационной мастерской
1	16.	База данных рыболовецкой артели
1	17.	База данных службы охраны труда
1	18.	База данных спортивного клуба
1	19.	База данных стоматологической поликлиники
1	20.	База данных строительной фирмы
1	21.	База данных транспортной компании (пассажирские перевозки внутригородские)
1	22.	База данных туристического оператора
1	23.	База данных фирмы по ремонту автомобилей (автосервис)
1	24.	База данных школы олимпийского резерва
1	25.	База данных фирмы по продаже программного обеспечения

Таблица А.2 – Варианты предметных областей для группы №2

№ группы потока	№ в классном журнале	Предметная область
2	1.	База данных агентства по экспресс-доставке
2	2.	База данных ассоциации фермерских хозяйств
2	3.	База данных аэропорта (организация авиаперевозок)
2	4.	База данных библиотеки научно-исследовательского института
2	5.	База данных ветеринарной клиники
2	6.	База данных кафедры института (документооборот)
2	7.	База данных коммерческого учебного центра
2	8.	База данных лечебного учреждения (стационар, планирование и учет работы медицинского персонала)
2	9.	База данных медицинской клиники (предоставление медицинских услуг)
2	10.	База данных медицинской страховой компании
2	11.	База данных морского порта (грузоперевозки)
2	12.	База данных оптового склада продуктов питания
2	13.	База данных предприятия по производству молочной продукции
2	14.	База данных проката спортивного инвентаря
2	15.	База данных санаторно-курортного комплекса
2	16.	База данных сети аптек
2	17.	База данных транспортной компании (железнодорожные перевозки грузов)
2	18.	База данных транспортной компании (междугородние пассажирские перевозки)
2	19.	База данных туристического бюро
2	20.	База данных туристического клуба
2	21.	База данных туристической базы
2	22.	База данных фирмы по продаже программного обеспечения
2	23.	База данных хранилища музейных ценностей
2	24.	База данных центра занятости (предоставление вакансий)
2	25.	Библиотека образовательного учреждения (университет)

Таблица А.3 – Варианты предметных областей для группы №3

№ группы потока	№ в классном журнале	Предметная область
3	1.	База данных библиотеки образовательного учреждения (школа)
3	2.	База данных адвокатской конторы
3	3.	База данных предприятия общепита
3	4.	База данных участкового врача
3	5.	База данных учета материальных ценностей
3	6.	База данных планового отдела производства кондитерских изделий
3	7.	База данных диспетчерской коммунальных аварийных служб
3	8.	База данных фермерского хозяйства (животноводство)
3	9.	База данных издательского дома
3	10.	База данных туристического агентства
3	11.	База данных фельдшерского пункта
3	12.	База данных комбината бытового обслуживания
3	13.	База данных сети автозаправочных станций
3	14.	База данных учета военнообязанных
3	15.	База данных агентства недвижимости
3	16.	База данных гостиничного комплекса
3	17.	База данных больницы. (лекарственное обеспечение)
3	18.	База данных поликлиники (планирование и учет работы медицинского персонала)
3	19.	База данных драматического театра
3	20.	База данных центра детского творчества
3	21.	База данных спортивной школы
3	22.	База данных минимаркета (движение товара)
3	23.	База данных диспетчерской службы радиотакси
3	24.	База данных службы охраны труда
3	25.	База данных показателей научно-исследовательской деятельности кафедры

Таблица А.4 – Варианты предметных областей для группы №4

№ группы потока	№ в классном журнале	Предметная область
4	1.	База данных почтового отделения связи
4	2.	База данных проектно-строительной компании
4	3.	База данных спортивной школы
4	4.	База данных для контроля выполнения нагрузки преподавателей ВУЗа
4	5.	База данных районной поликлиники (учет льготных лекарств)
4	6.	База данных букмекерской конторы
4	7.	База данных рыболовецкой артели
4	8.	База данных для контроля успеваемости школьников
4	9.	База данных для автозаправочной станции
4	10.	База данных отдела охраны памятников архитектуры
4	11.	База данных показателей научно-исследовательской деятельности кафедры
4	12.	База данных фирмы по прокату строительной техники
4	13.	База данных для управления работой компьютерных аудиторий учебного заведения
4	14.	База данных фирмы по ландшафтному дизайну
4	15.	База данных реставрационной мастерской
4	16.	База данных учета расчетов с клиентами в банке
4	17.	База данных лечебного учреждения (стационар, обслуживание пациента)
4	18.	База данных центра занятости (предоставление вакансий)
4	19.	База данных таксомоторного парка
4	20.	База данных абонентского отдела интернет-провайдера
4	21.	База данных детского оздоровительного лагеря
4	22.	База данных туристического клуба
4	23.	База данных ветеринарной клиники
4	24.	База данных горноспасательной службы
4	25.	База данных для учета контингента студентов ВУЗа

Таблица А.5 – Варианты предметных областей для группы №5

№ группы потока	№ в классном журнале	Предметная область
5	1.	База данных санитарно-эпидемиологической службы (медицинская статистика)
5	2.	База данных агропромышленного предприятия (мониторинг хода сельхозработ)
5	3.	База данных показателей научно-исследовательской деятельности кафедры
5	4.	База данных санитарно-эпидемиологической службы (мониторинг состояния прибрежных вод)
5	5.	База данных системы спутникового мониторинга транспорта
5	6.	База данных диспетчерской коммунальных аварийных служб
5	7.	База данных агропромышленного предприятия (инвентаризация земель)
5	8.	База данных кадастрового учет земель
5	9.	База данных расчетного центра услуг жилищно-коммунального хозяйства (ЖКХ)
5	10.	База данных отдела охраны памятников архитектуры
5	11.	База данных компании по перевозке опасных грузов
5	12.	База данных агропромышленного предприятия (мониторинг состояния сельхозземель)
5	13.	База данных ГИБДД (регистрация транспортных средств)
5	14.	База данных поисковых отрядов
5	15.	База данных системы мониторинга состояния атмосферного воздуха
5	16.	База данных агропромышленного предприятия (мониторинг состояния посевов)
5	17.	База данных археологической экспедиции
5	18.	База данных морского порта (лоцманская служба)
5	19.	База данных проектно-строительной компании
5	20.	База данных гостиничного комплекса
5	21.	База данных типографии
5	22.	База данных юридической консультации
5	23.	База данных кинологического клуба
5	24.	База данных пункта проката автомобилей
5	25.	База данных фирмы по ландшафтному дизайну