

КАФЕДРА ИНЖЕНЕРНОЙ КИБЕРНЕТИКИ

ЗАДАНИЕ
на курсовую работу
по учебной дисциплине

«Специальные главы по базам данных»

бакалавриат по направлению 01.03.04 прикладная математика

VIII семестр, 2024÷2025 уч. год

СОДЕРЖАНИЕ

Список используемых в тексте основных сокращений	3
Введение.....	4
1. Общие положения.....	5
1.1. Краткое описание проблематики	5
1.2. Выбор предметной области и средств разработки	9
1.3. Защита курсовой работы.....	11
2. Рекомендуемые направления.....	12
2.1. Перечень возможных направлений для выбора темы курсовой работы.....	12
2.2. Направление (тема), предложенная учащимся	13
3. Требования к архитектуре (модели) данных.....	14
3.1. Общие требования к модели данных проекта	14
3.2. Общие требования к реляционной модели данных в проекте	16
3.3. Общие требования к ролевому доступу	17
4. Основные требования к программному обеспечению для взаимодействия с базами (хранилищами) данных	18
4.1. Требования к основной функциональности программного обеспечения	18
4.2. Требования к дополнительной функциональности программного обеспечения	19
4.3. Основные требования и рекомендации к графическому интерфейсу пользователя	20
5. Общие требования к структуре и содержанию отчета.....	25
5.1. Обязательные разделы отчета	25
5.2. Дополнительные разделы отчета	31
6. Требования к оформлению отчета	32
6.1. Параметры страниц	32
6.2. Форматирование текста.....	32
7. Требования к электронной презентации	34
ПРИЛОЖЕНИЕ – Титульный лист отчета.....	36

Список используемых в тексте основных сокращений

БД	- база данных
КР	- курсовая работа
ИС	- информационная система.
ИТ	- информационные технологии
ПО	- программное обеспечение
ОМД	- общая модель данных
РСУБД	- реляционная система управления базами данных
СУБД	- система управления базами данных
GUI	- Graphical User Interface (графический интерфейс пользователя)
IDE	- Integrated Development Environment

Введение

Настоящий документ содержит **задание** и основные требования к структуре, содержанию и оформлению отчетных материалов по курсовой работе (**КР**), выполняемой учащимися бакалавриата по направлению подготовки 01.03.04 прикладная математика в процессе изучения учебной дисциплины «**Специальные главы баз данных**» в семестре обучения, предусмотренном учебным планом.

Основная цель курсовой работы по учебной дисциплине «**Специальные главы по базам данных**» – разработка на уровне демонстрационного прототипа алгоритмического и программного обеспечения, решающего конкретную задачу обработки и (или) анализа информации, функциональная структура которого в обязательном порядке включает

- **мультимodelьную подсистему хранения информация**¹ для выбранной предметной области, реализованную на базе **не менее трёх классов (видов, типов) моделей данных** и с использованием **не менее двух различных СУБД** (мультимodelьных или мономodelьных);
- действующий (работоспособный) прототип программы для взаимодействия с БД (клиентского приложения(-ий)) для **ввода, редактирования, удаления и отображения** (администрирования – при необходимости) всей информации, хранящейся в базах данных, а также для управления тематическими запросами (не менее 10 ед.²) и отображения результатов их выполнения.

Выполнение курсовой работы может осуществляться в команде из 3-х человек. Допускается выполнение курсовой работы в **парах** либо **индивидуально**, но с обязательным согласованием с преподавателем.

В случае выполнения курсовой работы в составе команды с преподавателем обязательно согласовывается разделение функциональных обязанностей каждого учащегося в команде.

¹ В настоящее время также используется термин *многовариантное хранение данных*, соответствующий англоязычному термину «**polyglot persistence**»

² Количество и содержание тематических (предметных) запросов к содержимому баз данных согласовывается с преподавателем

1. Общие положения

1.1. Краткое описание проблематики

В современную эпоху, одним из признаков которой является лавинообразный рост информации, разработчики программных систем самых разнообразных типов и назначения постоянно сталкиваются с необходимостью решать проблему, связанную с такой доставляющей неудобство характеристикой современных информационных потоков, как разнообразие структур и форм данных, подлежащих долговременному хранению в базах данных (БД). Данные, которые необходимо обрабатывать, анализировать и хранить, имеют самую разную структуру. Они могут быть в форме временных рядов, в форме документов различных форматов (текстовые файлы (контейнеры), электронные таблицы, контент интернет-ресурсов), в форме статических и динамических графических изображений, в форме видео и аудио файлов, в объектной форме и т.д. и т.п.

«Ключевой особенностью рынка систем управления базами данных является лавинообразное возрастание требований к работе с данными. При среднем росте объема собираемых данных на 25-30% в год, количество операций, совершаемых с данными, по нашим оценкам более чем удваивается ежегодно.

*Если раньше универсальным способом решения задач, связанных с производительностью, было просто добавление новых серверов, то теперь количество необходимого «железа» становится настолько велико, что это уже экономически невыгодно, и чтобы соответствовать все более возрастающим клиентским требованиям и оставаться при этом экономически эффективными, **компании начинают искать решения, основанные на качестве софта, а не на «железе»** ...»¹*

Как следствие – необходимо соблюдать ряд новых требований, среди которых наиболее важным является такое, что: современные СУБД должны на новом уровне качества поддерживать различные «техники» (модели, подходы, методы) разделения информации и изначально быть готовыми превращаться в высокоэффективные и «эластичные» распределенные СУБД.

¹ Источник цитирования – сайт аналитической компании SH Capital LLC (<http://dbms.sh.capital/>)

Всё это приводит к тому, что для организации надежного долговременного хранения данных с таким разнообразием форм разработчикам приходится создавать целые информационные комплексы, отличающиеся следующей особенностью. В рамках этих информационных комплексов, нередко та их часть (подсистема), которая отвечает за хранение данных и решения различных задач по их обработке, формируется на базе нескольких различных СУБД, каждая из которых поддерживает свою модель данных.

Это привело к появлению такого направления в технологиях баз данных, как разработка и использование **мультимодельных** (или **многовариантных**) подсистем хранения информации (СУБД, СХД, Data Warehouse и проч.) в различных информационных системах и комплексах. Известный IT-эксперт Мартин Фаулер¹ в одной из своих книг² ввёл специальный термин для таких ситуаций – «**polyglot persistence**», который на русский язык часто переводится как **многовариантное хранение данных** и который устоялся и прочно закрепился в IT-лексиконе.

До сих пор во многих информационных системах, как правило, используют один единый «движок» БД (СУБД). Он одновременно используется для хранения и обработки данных самых разных по содержанию и по исходной форме представления в предметной области:

- всех данных из предметной области (документы, информация о структурах сетевого типа, мультимедийная информация и проч.),
- служебных данных, например, сведения о транзакциях, данные управления сеансами, журналы изменений данных и обращений к БД и проч.;
- для других потребностей хранения и обработки, например, генерация отчетов, обеспечение данными BI-инструментов, пополнение хранилищ данных и т.д.

Однако использование одного единого «движка» для данных (в виде одной конкретной СУБД) для всех указанных потребностей в настоящее время приводит к

¹ Мартин Фаулер (англ. Martin Fowler) — ведущий IT-эксперт и автор многих книг по архитектуре ПО, объектно-ориентированному анализу и разработке, экстремальному программированию и другим аспектам современно software development.

² *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence* / Pramod J. Sadalage, Martin Fowler, Addison-Wesley Professional, –2012

неэффективным решениям. Возникает целый спектр архисерьёзных проблем, оказывающих взаимное влияние друг на друга и связанных с попыткой «уложить в прокрустово ложе» одной модели данных (чаще всего это реляционная модель) информацию самого широкого спектра типов и форм.

Во-первых, это сложность управления, администрирования и в целом поддержание «боеготового» состояния такого «хозяйства».

Во-вторых, сложность управления транзакциями в соответствии с требованиями ACID в такой *de facto* гетерогенной системе.

В третьих, сложность решения задач по масштабированию, репликации и шардингу существующих систем хранения данных (СУБД, СХД, хранилища данных).

В четвертых, усложнение поиска решения для задач аналитической обработки хранящейся информации.

В пятых, экономическая неэффективность и постоянно растущие финансовые затраты на поддержание в работоспособном состоянии существующего программного комплекса, построенного в рамках мономодельной парадигмы для системы хранения данных (СУБД).

Указанный ряд проблем (которыми, кстати, ситуация не ограничивается) в современных условиях не может быть легко или экономически эффективно решен с помощью одного типа технологии баз данных. Несмотря на то, что значительная по объему часть базовой информации является традиционной для хранения в классических реляционных СУБД (например, информация об обрабатываемых СУБД транзакциях), много другой информации в исходной форме не является реляционной. И в этом случае для хранения данных уже требуется, по крайней мере, не менее двух типов «механизмов» (моделей, подходов), что «автоматически» подталкивает разработчиков и архитекторов таких систем переходить «на рельсы» парадигмы *polyglot persistence*.

Всё вышеперечисленное, привело к тому, что в настоящее время в технологиях баз данных и в практике их применения происходят радикальные изменения. Среди этих изменений необходимо выделить следующие два процесса.

1) В последнее десятилетие наблюдается рост вариативности используемых технологий хранения данных. Одновременное использование нескольких типов СУБД при создании единой подсистемы хранения информации значительно возросло по сравнению с традиционной стратегией, построенной на мономодельной парадигме (в основном – реляционной), когда все информационные «яйца складываются в одну корзину». Возник специальный термин – **polyglot persistence** – означающий, что в системе долговременного хранения информации используется **многовариантное (или мультимодельное) решение** для хранения данных, опирающееся, в том числе, на их специализацию и форму представления в предметной области.

Базы данных, построенные в парадигме *polyglot persistence*, используются, когда необходимо решить сложную задачу, разбив ее на сегменты и применив различные модели баз данных для решения различных задач, а также когда необходимо использовать различные технологии для обработки различных потребностей в хранении данных.

2) Лидеры рынка СУБД (и часть других СУБД) инкапсулируют «под одной продуктово-технологической крышей» (или на одной технологической платформе) одновременно вместе с реляционной технологией и другие технологии хранения данных, базирующихся, в первую очередь, на наиболее популярных NoSQL подходах (документная модель, графовая/сетевая модель, модель «ключ-значение» и проч.). Такие СУБД уже позиционируются (и фактически являются) уже не как «чистые» РСУБД (Relational DBMS), а уже как СУБД, поддерживающие более одной моделей данных или **мультимодельными СУБД** (*многовариантными*; multimodel DBMS). В таких **мультимодельных СУБД** наряду с традиционной реляционной моделью можно использовать и другие подходы (модели) для хранения данных в форме, более приближенной к той, в какой они существуют в предметной области.

Таким образом, в настоящее время технологии баз данных находятся в переходной стадии своего существования. Эта стадия развития технологий баз данных является трендом сегодняшнего дня и ближайшего времени. Она характеризуется процессом миграции из существовавшего десятилетия состояния доминирования мономодельной парадигмы хранения данных на базе реляционного подхода в состо-

яние, в котором «первую скрипку» будут играть программные решения, предназначенные для долговременного хранения информации и создаваемые на основе мультимодельной (многовариантной) парадигмы.

1.2. Выбор предметной области и средств разработки

Основные направления, рекомендуемые учащимся для определения темы курсовой работы по дисциплине «**Специальные главы баз данных**» представлены далее (см. п.2.1). Предметная область, тема, основные планируемые результаты курсовой работы и ключевые параметры создаваемого программного обеспечения выбираются и предлагаются учащимся самостоятельно с обязательным последующим их согласованием с преподавателем. При необходимости учащийся может разрабатывать дополнительное программное обеспечение, если потребность в нем определяется общей постановкой задачи и целевым назначением программной системы.

При выполнении **КР** учащемуся предоставляется право свободного выбора:

- целевой программно-аппаратной платформы, под управлением которой будет функционировать разрабатываемое в **КР** программная система;
- моделей данных и **СУБД** для организации хранения информации в разрабатываемой программной системе;
- основного архитектурного решения (концепции) по реализации в **КР** программного проекта, например, архитектура «клиент-сервер» (в том числе многоуровневая), трёхзвенная архитектура (англ. three-tier), однорагновая сеть (peer-to-peer) и т.п.;
- языка (языков) программирования для создания программного обеспечения при выполнении **КР**;
- единой среды разработки программного обеспечения (IDE – комплекс программных средств для создания **ПО**);
- программных средств для логического (инфологического) и/или физического моделирования данных;
- ORM-библиотеки;

- иного дополнительного программного обеспечения, необходимого для выполнения разработки программного обеспечения **КР**, как-то: библиотеки; фреймворки; комплексы технологического программного обеспечения для обеспечения взаимодействия (middleware) между различными приложениями, системам, компонентами и т.п.
- современных систем/платформ искусственного интеллекта (интеллектуальные интернет-сервисы), обладающих возможностями для решения задач в различных областях интеллектуальной обработки, анализа информации и прогнозирования, и предоставляющих открытый (бесплатный) доступ к своей функциональности через интернет (при необходимости).

ВНИМАНИЕ! При использовании выбранных для выполнения **КР** программных средств учащийся обязан соблюдать все требования, связанные с защитой прав интеллектуальной собственности, которые установлены их владельцами и (или) авторами (разработчиками).

Выбор указанных средств должен быть согласован с преподавателем, ведущим учебную дисциплину.

Как было сказано выше, учащемуся в **КР** предоставляется право выбора моделей данных и **СУБД**. При выполнении **КР** можно использовать, как «чистые» реляционные СУБД, мультимодельные **СУБД**, а также системы, не поддерживающие реляционную модель данных (**СУБД** из класса **NoSQL**). В случае использования в **КР** реляционной модели данных (и соответственно реляционной СУБД), единственным технологическим ограничением является то, что выбранная СУБД с поддержкой реляционной модели должна поддерживать стандарт языка SQL не ниже уровня SQL:2003.

Разработанные в курсовой работе БД должны быть заполнены некоторым количеством содержательной и взаимосвязанной информации (примерами). Для иллюстрации работоспособности приложения достаточно внести в основные таблицы базы данных 7 ÷ 10 содержательно значимых записей, т.е. имеющих содержательный смысл в контексте выбранной предметной области.

1.3. Защита курсовой работы

Каждая курсовая работа проходит процедуру **защиты**. При проведении защиты КР учащийся (команда учащихся) демонстрирует:

- электронную презентацию (при командной работе);
- работоспособность созданного программного обеспечения и безошибочность реализации всего заявленного функционала;
- подготовленный отчет по КР.

В процессе и (или) после демонстрации преподаватель задает учащемуся (членам команды) вопросы по различным аспектам выполнения КР и полученным в ходе её выполнения результатам. В случае наличия существенных замечаний и (или) выявления ошибок в работе созданного программного обеспечения; тексте отчета и (или) презентации защита КР прекращается до полного исправления выявленных ошибок и устранения сделанных замечаний.

После того как преподавателем будет окончательно принято созданное в курсовой работе программное обеспечение, результаты его работы и отчетные материалы, учащийся обязан сдать¹ преподавателю следующие обязательные **отчетные материалы по курсовой работе**:

- ♦ оформленный отчет по курсовой работе – индивидуальный для каждого учащегося;
- ♦ при командной работе – общая для команды презентация по результатам курсовой работы (не менее 10 слайдов);
- ♦ все файлы программного проекта.

Форма и режим сдачи отчетных материалов объявляются преподавателем отдельно.

¹ По решению преподавателя отчетные материалы могут либо сдаваться в распечатанном виде (на твердом носителе) либо загружаются учащимся в электронном виде в LMS CANVAS

2. Рекомендуемые направления

Выбор предметной области, направления и темы курсовой работы по дисциплине «**Специальные главы баз данных**» осуществляется учащимся в срок, установленный преподавателем.

2.1. Перечень возможных направлений для выбора темы курсовой работы

Учащимся предлагаются следующие возможные направления для разработки для определения предметно-содержательной ориентации темы **КР**:

- программная система для учета кадров организации и траекторий развития их профессиональных компетенций (демонстрационный прототип программной системы класса Human Capital Management);
- программная система для учета, организации и планирования работы и хранения видеоконтента для телестудии, осуществляющей вещание в интернет (уровень: демонстрационный прототип);
- программная система закрытой (например, корпоративной) социальной сети (уровень: демонстрационный прототип);
- программная система для ведения учета сети предприятий общественного питания (рестораны, кафе, столовые и т.п.) от закупок продуктов и составления плана поставок и до количества положительных отзывов клиента (уровень: демонстрационный прототип);
- программная система для учета, организации и планирования занятий специализированного образовательного учреждения в сфере дополнительного образования и (или) специализированного досуга, например, музыкальная школа, спортивный центр (школа, секция, каток, корт, бассейн, фитнес-центр и т.п.), стрельбище (тир), автодром и т.п. (уровень: демонстрационный прототип);
- программная система управления частными (например, домашними) коллекциями (книги; произведения изобразительного искусства и антиквариата; монеты; продукция филателии и т.п.);

- программная система для учета и управления библиографической информацией, позволяющая хранить, совместно создавать и редактировать научно-исследовательские работы (публикации) с подключением к международной социальной сети учёных (демонстрационный прототип-аналог таких систем, как Mendeley, Citavi, EndNote® и иных аналогов);
- программная система регистрации в реальном времени событий и хранения мультимедийной информации для комплексных систем безопасности (уровень: демонстрационный прототип);
- обучающая система в той части, которая отвечает за хранение образовательного материала и автоматизацию некоторого множества функций и процессов при изучении пользователем конкретной темы (предмета, дисциплины, и т.п.) (уровень: демонстрационный прототип).

2.2. Направление (тема), предложенная учащимся

Команда учащихся (либо отдельный учащийся) может предложить свою тему для курсовой работы, которая должна быть обязательно утверждена преподавателем (вместе с основными параметрами работы).

3. Требования к архитектуре (модели) данных

3.1. Общие требования к модели данных проекта

Основным обязательным элементом функциональной структуры программного проекта, который должен реализовать в КР учащийся (команда учащихся), является – **мультимодельная подсистема хранения информации** для выбранной предметной области, реализованная на базе **не менее чем трёх классов моделей данных** и с использованием **не менее чем двух различных СУБД**.

В ходе выполнения КР учащийся (команда учащихся) разрабатывает **общую модель данных** программной системы (ОМД) для организации надежного длительного хранения информации в создаваемой системе. Такая **ОМД** должна включать в себя **не менее трёх** информационных логических (тематических) кластеров, каждый из которых создается на базе выбранного класса моделей данных.

В курсовой работе **ОМД** образуется по следующему принципу (схеме):

Общая модель данных системы = Модель данных 1 «+»

Модель данных 2 «+»

Модель данных 3 [«+»]

Модель данных 4 [+ ...]

Например (для трёхмодельной **ОМД**), возможные сочетания различных классов моделей данных могут быть такими:

Общая модель данных системы = Реляционная модель +

Документно-ориентированная модель +

Модель «key-value»

Общая модель данных системы = Реляционная модель + Графовая модель +

Модель «key-value»

**Общая модель данных системы = Графовая модель +
Модель «key-value» +
Документно-ориентированная модель**

**Общая модель данных системы = Объектно-реляционная модель +
Графовая модель +
Документно-ориентированная модель**

**Общая модель данных системы = Документно-ориентированная модель +
Модель «ключ-значение» +
Модель пространственных данных
(spatial)**

Выбор конкретных классов моделей данных для реализации **ОМД** осуществляется учащимся **самостоятельно**, исходя из целевой направленности разрабатываемой в **КР** программной системы и особенностей предметной области.

Архитектура (структура) ОМД по всем аспектам согласовывается учащимся с преподавателем в обязательном порядке.

3.2. Общие требования к реляционной модели данных в проекте¹

В случае использования реляционной модели данных команда учащихся должна соблюдать следующие требования.

1) Количество таблиц в реляционном информационном кластере (составляющей **ОМД**) определяется учащимся самостоятельно с учетом численного состава команды (см. табл.), при этом среднее число полей в одной таблице: **5 ед.**

Число участников команды	Количество таблиц:
1	Не менее 10
2	Не менее 15
3	Не менее 18

2) Все уникальные идентификаторы (первичные ключи) являются **суррогатными**. Применение **естественных первичных ключей** возможно только в качестве исключения и должно быть чётко аргументировано. Разрабатываемое программное обеспечение должно реализовывать автоматическую (без участия пользователя) установку значений уникальных идентификаторов (значений первичных ключей) у записей во всех таблицах.

3) Выбор типов данных для полей и ограничений, накладываемых на поля и таблицы, осуществляется учащимся самостоятельно.

4) Реакция на команды обновления.

– **Удаление** значения (delete) любого **родительского ключа**, в случае если существуют внешние ссылки на него – **запрещены** (restrict).

– **Изменение** существующих значений (update) любого **родительского ключа** – для всех ссылающихся на него внешних ключей должно быть автоматически проведено каскадное обновление.

5) Состав и назначение иных классов метаобъектов базы данных, таких как – хранимые процедуры, триггеры, представления (view), исключения, генераторы,

¹ Требования, изложенные в данном разделе, необходимо соблюдать только при использовании *реляционной модели данных*

домены и др., определяется учащимся самостоятельно по своему усмотрению (если иное не установлено настоящим заданием).

3.3. Общие требования к ролевому доступу

В КР при взаимодействии пользователей посредством клиентских приложений с БД **должен быть реализован механизм ролевого доступа**.

При выполнении **КР** в обязательном порядке реализуются следующие роли:

- роль пользователя с полными правами (привилегиями) на выполнение всех возможных действий (добавление/вставка данных, извлечение, модификация и удаление данных) с любыми объектами БД (роль администратора БД);
- роль пользователя, имеющего права (привилегии) только на чтение данных (возможно не всех);
- роль пользователя с комбинированными правами доступа, для которой все объекты БД разделяются на три подмножества:
 - первое – включает в себя объекты, к которым любая форма доступа (привилегии) запрещена;
 - второе – включает в себя объекты, доступ к которым разрешен только для чтения данных;
 - третье – включает в себя объекты, к которым у пользователя есть полный доступ.

Для роли с комбинированными правами конкретный состав объектов БД в каждом из указанных множеств определяется разработчиками исходя из архитектуры БД и процессов, которые должен выполнять соответствующий пользователь.

4. Основные требования к программному обеспечению для взаимодействия с базами (хранилищами) данных

Обязательным элементом архитектуры разрабатываемого программного проекта, который должен быть реализован в КР, является – **действующий (работоспособный) прототип программной системы для взаимодействия с БД** (клиентское приложение) реализованного на выбранном учащимся языке программирования. Клиентское приложение может быть не единичным. В зависимости от количества, содержания, размерности и логики процессов обработки информации в выбранной предметной области, учащийся может разработать столько отдельных самостоятельных приложений для взаимодействия с БД, сколько необходимо.

Основная задача клиентского приложения – дать пользователю средство для выполнения следующих операций с БД: **ввод данных, удаления данных из БД, извлечение (выборка) и модификация существующих данных, а также для управления тематическими запросами (не менее 10 ед.) к данным, хранящимся в БД и отображения результатов их выполнения.**

Клиентское приложение **должно располагать графическим пользовательским интерфейсом** (англ. graphical user interface, GUI) – т.е. реализованной системой запрограммированных средств для взаимодействия пользователя с БД, основанной на представлении всех доступных пользователю системных объектов и функций в виде графических компонентов экрана (окон, значков, меню, кнопок, разнообразных списков и т. п.).

При разработке программного обеспечения для взаимодействия с БД (клиентского приложения) следует реализовать пользовательский интерфейс в части ввода данных таким образом, чтобы **минимизировать ошибки ввода данных в БД и дублирования сохраняемой в БД информации** в том числе.

4.1. Требования к основной функциональности программного обеспечения

В программном обеспечении для взаимодействия с БД (**клиентское приложение**) должны быть реализованы следующие основные функции:

- **ввод** информации в любые объекты **БД**, созданные для долговременного хранения информации (например, таблицы в **РСУБД**);
- **редактирование** информации, содержащейся в любых объектах **БД** в соответствие с логикой процессов обработки информации, которая определяется общим назначением разрабатываемого в **КР** программного проекта и свойствами предметной области;
- **удаления** информации, содержащейся в любых объектах **БД** в соответствие с логикой процессов обработки информации, которая определяется общим назначением разрабатываемого в **КР** программного проекта и свойствами предметной области;
- **извлечения** (выборка) всей информации, содержащейся в **БД** в соответствие с правами (привилегиями) пользователя и текущей ролью активного пользователя;
- **управление транзакциями**, в рамках которых осуществляются операции ввода, удаления и изменения данных;
- **управление** тематическими (предметно-содержательными) запросами (не менее 10 ед.) и отображения результатов их выполнения;
- **управление** ролевым доступом к объектам **БД** (см. п.3.3).

4.2. Требования к дополнительной функциональности программного обеспечения

В случае необходимости и в зависимости от количества, содержания, размерности и логики процессов обработки информации в выбранной предметной области, учащийся может реализовать другие дополнительные функции в программном обеспечении для взаимодействия с **БД** (клиентское приложение).

В качестве таких дополнительных функций можно указать, например, администрирование **БД** и (или) всей программной системы в целом; взаимодействие со сторонним программным обеспечением (локально или через сеть); генерация отчетов; аналитическая обработка хранящейся в **БД** информации и др.

4.3. Основные требования и рекомендации к графическому интерфейсу пользователя

Язык интерфейса в программном обеспечении для взаимодействия с БД (клиентского приложения) – **русский**.

Программное обеспечение для взаимодействия с БД (клиентское приложение) должно предоставлять удобный, интуитивно понятный графический пользовательский интерфейс, который должен обеспечивать:

- визуальное отображение показываемой пользователю информации в комфортной для пользователя цвето-шрифтовой системе;
- быструю и удобную навигацию на экранных формах по отображаемым и активным визуальным компонентам управления;
- масштабирование отображаемой информации в зависимости от активного экранного разрешения режима отображения информации на мониторе (экране мобильного устройства) пользователя и других настроек экрана пользователя (гарнитуры и размер шрифтов и др.);
- визуальное структурирование информации в соответствии с её предметным (тематическим) содержанием и (или) назначением;
- соответствие всех экранных форм и визуальных компонент управления выбранному единому стилю оформления и форматирования.

Дизайн всех экранных форм (или их аналогов) **клиентского приложения** и расположенных на нем видимых визуальных компонент управления должен быть унифицирован и оформлен, исходя из одного стиля, единого для всей программы. Применяемые шрифты должны быть хорошо читаемыми, с комфортной для пользователя комбинацией размера, яркости и контрастности.

Если **клиентское приложение** разрабатывается для использования на мобильном устройстве, то дизайн каждого из экранов, с которыми работает пользователь, должен быть разработан под оба положения экрана мобильных устройств: вертикальное и горизонтальное, а также быть адаптивным и автоматически подстраиваться (регулироваться) под различные разрешения экранов мобильных устройств. В за-

висимости от выбранной целевой программной или программно-аппаратной платформы, под управлением которой будет функционировать клиентское приложение, разрешается использовать дизайн экранов в соответствии со стандартами проектирования интерфейсов компаний Apple и Google.

Фон экранных форм должен быть однородным, в случае, если используется текстурирование, то выбранная текстура также должна быть однородной и не создавать помех и затруднений для визуального считывания пользователем информации, отображаемой на экранной форме.

Программное обеспечение для взаимодействия с БД (клиентское приложение) должно:

– обеспечивать пользователя возможностями (средствами) для минимизации ошибок при **вводе** данных в БД:

а) непосредственный («ручной») ввод в **БД** данных должен иметь место только там, где ввести информацию иным способом нельзя;

б) ввод информации на экранных формах везде, где это возможно, должен осуществляться путем выбора вводимых данных из четко определенного множества значений, указываемого в визуальных элементах управления, например, списочного типа (выпадающий список, пролистываемый список, календарь и т.д. и т.п.);

в) ввод информации, требующий соблюдение известных форматов ввода данных, выполнять с использованием *масок ввода*¹ данных;

– обеспечивать пользователя функциональными возможностями (средствами) для минимизации ошибок при выполнении операций **изменения** данных и **удаления** данных из БД:

а) всегда, когда это возможно, перед стартом операции изменения или удаления данных пользователю должно показываться программное окно (экранная форма) с сообщением или предупреждением, содержащим

¹ **Маска ввода** — это строка символов, указывающая формат допустимых значений входных данных. **Маски ввода** можно использовать в полях таблиц или запросов, а также в элементах управления форм и отчетов. **Маску ввода** рекомендуется использовать, когда вводимые значения должны иметь единый формат, например для полей с номерами телефонов из десяти цифр.

требование дополнительно подтвердить выполнение соответствующей операции изменения или удаления данных;

б) везде где это возможно использовать всплывающие подсказки (hints) с информацией, поясняющей назначение элементов пользовательского интерфейса;

– обеспечивать соблюдение следующих основных требований к интерфейсу пользователя:

а) наличие названий экранных форм (экранов), содержательно отражающих соответствующий этап работы с данными и (или) работы клиентского приложения;

б) наличие программных меню на каждой экранной форме, отражающих только те функции и задачи в активной странице (экране), которые ему доступны, при этом:

- все пункты меню должны иметь контекстную помощь в виде всплывающих подсказок (hints) и, по возможности, сопровождаться пиктограммами (графическими изображениями), поясняющими назначения данного пункта меню;
- разделы и пункты меню, которые дают доступ к функциям управления, должны быть постоянно доступны пользователю с любой страницы (экрана, экранной формы) с минимальным поэтапным предварительным переходом.

в) для числовых типов данных:

- запрещены: предшествующий необязательный знак «+» и лидирующие нули (если иное не является обязательным из-за особенностей самих данных, например, температура окружающей среды; полный телефонный номер в международном формате и т.п.);
- для вещественных чисел – число цифр после десятичной точки должно быть чётным числом, например, две, четыре или шесть цифр (если иное не требуется, исходя из содержательной особенностей и свойств самих данных);

г) для данных типа «Дата» и «Время» обязательно заполнение в формате без указания временного пояса: «Дата» – <ДД-ММ-ГГГГ>; «Время» – <ЧЧ:ММ> или < ЧЧ:ММ:СС>;

д) для визуальных (видимых) управляющих элементов управления интерфейса пользователя (controls):

- позиционирование осуществляется только в активной (видимой) области экрана;
- в случае необходимости, при размещении вне видимой области экрана визуальных (видимых) управляющих элементов интерфейса управления должна быть возможность перемещения (сдвига, прокрутки) видимой области экрана;
- для всех допускающих прокрутку (сдвиг) областей экрана сам факт возможности прокрутки и метод прокрутки должен быть интуитивно понятен;
- расположение элементов управления должно осуществляться с учетом рационального использования доступного экранного пространства и антропометрической совместимости: пространственная компоновка форм и размерные характеристики визуальных элементов управления должны обеспечивать быстрый доступ к ним и комфортную работу с ними;

е) при табличном отображении данных на экранных формах:

- название полей (колонок, столбцов) таблицы должны быть на русском языке и они не должны повторять непосредственные названия полей тех таблиц БД, информация из которых отображается;
- запрещается выводить значения уникальных идентификаторов (первичных ключей) и значений родительских ключей для полей таблиц, организующих связи между таблицами в БД (при использовании в КР реляционной модели данных);

– обеспечивать невозможность изменения пользователем «вручную» значений уникальных идентификаторов записей во всех таблицах (при использовании в **КР** реляционной модели данных);

5. Общие требования к структуре и содержанию отчета

Основные итоговые сведения, касающиеся выполненной курсовой работы и полученных в ней результатов, учащийся оформляет в форме отчета по КР, рекомендуемый объем которого – 20÷35 стр. без учета титульного листа и приложений.

Язык отчета по НИР – русский.

Отчет оформляется в соответствии с требованиями, изложенными в разделах № 3, 4.

Отчет по КР должен включать в себя нижеперечисленные обязательные разделы (главы), а также может содержать дополнительные (необязательные) разделы, комплексно раскрывающие содержание и уровень выполненных разработок, а также качество полученных результатов.

5.1. Обязательные разделы отчета

Отчет по курсовой работе **должен** включать в себя следующие разделы.

- 1) **Титульный лист**¹.
- 2) **Содержание отчета** (оглавление с указанием страниц начала разделов).
- 3) **Список используемых основных сокращений**.
- 4) **Введение**

Во введение указываются:

- полное название темы курсовой работы;
- основной результат, полученный в ходе выполнения **КР**: основная функциональность, наиболее значимые характеристики (применяемые модели данных – в обязательном порядке), особенности использования и свойства;
- общее (краткое) описание и основные характеристики той информации, для хранения и обработки которой создана соответствующая система;

¹ Распространяется преподавателем отдельно

- характер выполнения **КР** – индивидуальный или командный (проектный) и, в случае командной работы, полный перечень участников команды с указанием учебной группы и роли каждого участника.

5) Постановка задачи

Рекомендуемый объем – 1 ÷ 1½ стр.

В данном разделе учащийся должен, используя научно-технический стиль изложения, содержательно дать точную формулировку условий задачи КР с однозначно-понятным выделением этапов ее выполнения. Содержание данного раздела должно включать информацию, из которой однозначно понятно основное целевое назначение разрабатываемой программной системы и состоящей из описания:

- предметной области;
- назначения разрабатываемой программной системы, включая перечень основных задач в контексте предметной области (функционал), которые решает разработанное учащимся (командой учащихся) алгоритмическое и программное обеспечение;
- подробное развернутое описание того, какую информацию и в какой форме предлагается хранить и обрабатывать, а также примеры сохраняемых данных в базе;
- исходных данных для созданной программной системы (с примерами);
- дополнительных аспектов задачи (при наличии таковых).

6) Используемые средства разработки и системные требования.

В данном разделе приводятся следующие сведения:

- а) характеристики целевой программно-аппаратной платформы, под управлением которой должно функционировать разработанная в **КР** программная система, а также отдельные системные требования для подсистемы хранения информации (СУБД; сервер баз данных) и созданного программного обеспечения для взаимодействия с базой данных;

б) описание основного архитектурного решения, на принципах которого в **КР** был реализован весь программный проект в целом;

в) краткое описание языковых, инструментальных и иных средств ИТ, использованных для создания приложения:

- выбранные модели данных и конкретные СУБД для организации хранения информации в разработанной программной системе;
- язык (языки) программирования;
- программные средства для логического (инфологического) и/или физического моделирования данных;
- системы и средства разработки приложения/БД – IDE, фреймворки, библиотеки, платформы; специализированные сервисы; SDK, API и т.п.;
- ORM-библиотеки;
- системы/платформы искусственного интеллекта (интеллектуальные интернет-сервисы), предоставляющие открытый (бесплатный) доступ к своей функциональности через Интернет (при необходимости, т.е. если были использованы в **КР**).

7) Модель данных проекта

В данном пункте приводится полное и подробное описание разработанной **ОМД**, реализованной с использованием функциональных возможностей выбранных СУБД или (и) клиентского приложения.

а) графическими изображениями (диаграммы, графические модели (например, это ER-диаграмма(-ы)), и т.п.),

б) поясняющим сопроводительным текстом (объем 1 ÷ 1½ стр.), в котором даются краткие пояснения, раскрывающие наиболее важные моменты и особенности структуры модели.

в) В случае использования в КР реляционной модели, в отчете соответствующий [реляционный] логический кластер **ОМД** дополнительно представляется описанием созданных объектов БД в следующем формате.

в.1) Таблицы базы данных (для реляционных СУБД).

Вначале приводится общий список таблиц, созданных учащимся в БД.

№	Название таблицы	Назначение

После общего списка учащийся приводит краткое описание структуры каждой таблицы в следующей форме:

Название таблицы				
№	Название поля	Тип данных	Ключ (PK/FK)	Назначение

Необходимость использования, конкретный состав и назначение иных классов метаобъектов БД, как-то: хранимые процедуры, триггеры, домены, представления (VIEW), исключения, генераторы и др., определяется учащимся самостоятельно по своему усмотрению (если иное не установлено настоящим заданием).

В случае если учащийся разработал и использовал в созданной им БД указанные классы метаобъектов БД, то он обязан привести их описание по установленной форме (см. далее.).

в.2) Представления (VIEW) базы данных (для реляционных СУБД).

Общий список VIEW приводится с указанием следующих аспектов:

- Назначение (решаемая задача);
- Текст SQL запроса, формирующего VIEW.

в.3) Информация об иных использованных метаобъектах БД.

В данном пункте приводятся краткие сведения о прочих метаобъектах, созданных учащимся в БД, в форме таблицы со следующей структурой:

Список хранимых процедур (триггеров; доменов, индексов пользователя; др. объекты)

№	Название объекта	Назначение объекта	Примечание

Список метаобъектов каждого класса приводится в своей отдельной таблице.

8) **Организация ролевого доступа к данным.**

В данном пункте приводится описание созданных ролей в базе данных для организации различных типов доступа к данным:

- название роли;
- назначение;
- система привилегий, закрепленная за ролью.

9) **Тематические запросы к данным**

В данном пункте необходимо привести те запросы, которые решают в приложении содержательные задачи (в контексте предметной области), отличные от простого ввода/вывода информации и изменения данных в базе. Например, запросы, представляющие статистическую информацию, формирующие обобщающие отчеты, осуществляющие обработку регулярных выражений и т.п.

Описание таких запросов должно включать в себя краткий поясняющий текст (назначение) и собственно текст реализующей его команды (например, текст SQL команды).

10) **Научный аппарат (при наличии)**

Данный раздел отчета оформляется в случае, если в работе программного обеспечения задействован (реализован) какой-либо научный аппарат (направления, модели, методы, алгоритмы и проч.), связанный с направлениями (разделами) прикладной математики, машинного обучения, искусственного интеллекта и т.д. для обработки и (или) анализа хранящейся в БД информации.

Учащийся, используя научно-технический стиль изложения, должен содержательно описать то, какой конкретно научный аппарат, для каких задач и каким образом был использован в КР.

11) Программное обеспечение для взаимодействия с базой данных (клиентские приложения)

В данном пункте приводится описание разработанного учащимся программного обеспечения на основании следующей общей схемы изложения.

а) Общие сведения. Здесь описываются назначение программного обеспечения в целом, все его функциональные возможности, специфические особенности программной системы.

б) Список классов разработанных для работы приложения (в случае использования ООП и при наличии таковых).

в) Основные экранные формы, иллюстрирующие работу системы.

Описание экранных форм включает в себя скриншот формы (снимок) и краткий сопроводительный текст, поясняющий назначение формы и представленной на ней информации, а также особенности ее функционирования (если таковые имеются).

Рекомендуется в данном пункте привести описание не более 5 (пяти) экранных форм. Остальные экранные формы (при наличии) выносятся в приложениях к отчету по КР.

Важно! Скриншоты (снимки) экранных форм должны быть сделаны только в процессе работы клиентского приложения. Вставка скриншотов (снимков) экранных форм, сделанных в среде разработки/проектирования (IDE) не допускается.

12) Выводы.

Объем – не менее $\frac{3}{4}$ стр., но не более $1\frac{1}{2}$ стр.

В выводах дается краткое описание исследований и разработок, выполненных в ходе выполнения КР с обязательным анализом полученных результатов.

13) Список использованных источников (при наличии)

14) Приложения.

В качестве приложений отчета по КР **должны** обязательно присутствовать:

- пример программного кода клиентского приложения (не более 7 стр. по выбору учащегося) на основном языке программирования, использованного в разработке;
- скриншоты экранных форм, которые не были представлены в основном тексте отчета;
- полный общий SQL-скрипт (SQL-код) созданной БД, отражающий реальную физическую модель данных в контексте использованной СУБД (в случае использования РСУБД);
- блок-схемы алгоритмов (по желанию учащегося);

Любую другую информацию, которая, по мнению учащегося, является значимой и/или необходимой для иллюстрации проведенной работы и полученных результатов, также можно представить в качестве приложений к отчету по КР. Например, скриншоты экранных форм, не показанных в основной части отчета (при наличии таковых).

Необходимо иметь в виду, что если учащийся приводит какую-либо информацию в качестве дополнительных сведений, то она рассматривается и оценивается наравне с информацией, содержащейся в обязательных разделах отчета.

5.2. Дополнительные разделы отчета

Дополнительные разделы отчета не являются обязательными и оформляются учащимся только при необходимости в них. Однако в случае их присутствия в отчете по КР, их содержание рассматривается и оценивается наравне с информацией, содержащейся в обязательных разделах отчета. В качестве дополнительных разделов отчета могут выступать, например, описание отдельных алгоритмов, отличных от типовых действий по вводу/выводу/модификации/обработки информации; описание формул расчета значений каких-либо показателей, свойств, характеристик и т.п. для объектов предметной области, информация по которым обрабатывается и (или) анализируется программным обеспечением и т.д.

6. Требования к оформлению отчета

Текст отчета по КР, диаграммы (схемы) и иной иллюстративный материал должны быть отредактированы, а также отформатированы единым образом.

Текст отчета должен быть проверен на наличие и не должен содержать орфографических и синтаксических ошибок.

Сокращения слов, за исключением общепринятых, не допускаются.

Распечатанный отчет сдается только в сброшюрованном (сшитом) виде.

6.1. Параметры страниц

- Размер листа – формат «А4».
- Поля: левое: 25 мм, правое: 10 мм, верх и низ: 15 мм.
- Печать текста на листе – односторонняя.

6.2. Форматирование текста

- Шрифты:
 - ♦ основной текст – Times New Roman Cyr, 14 пт.;
 - ♦ названия глав (разделов) отчета – Times New Roman Cyr, 16 пт., набор прописными буквами с полужирным выделением (bold) и выравниванием по центру страницы;
 - ♦ заголовки пунктов и подпунктов – Times New Roman Cyr, 14 пт. с полужирным выделением (bold);
 - ♦ подписи рисунков и таблиц – Times New Roman Cyr, 11 пт.
- Абзацы:
 - ♦ первая строка – отступ: 1.25 см;
 - ♦ межстрочный интервал – 1.5;
 - ♦ выравнивание абзаца – по ширине;
 - ♦ интервал после абзаца обычного текста – 6 пт.
 - ♦ интервал после названий глав (разделов) отчета – 18 пт.

- ♦ интервал до и после заголовков пунктов и подпунктов – 12 пт.
- Нумерация страниц: внизу по центру; титульный лист является первой страницей отчета и не нумеруется.
- Содержание, выводы (заключение) и список использованных источников не нумеруются.
- Каждая глава (раздел), а также содержание, введение, выводы, список использованных источников и приложения начинаются с новой страницы.
- Нумерация рисунков и таблиц – сквозная целочисленная.
- Все таблицы и рисунки должны иметь название.
- Нумерация приложений прописными русскими буквами, например, «ПРИЛОЖЕНИЕ А», «ПРИЛОЖЕНИЕ Б» и т.д. с последующим указанием название каждого приложения;
- При изображении блок-схем алгоритмов необходимо придерживаться требований и правил стандарта ГОСТ 19.701-90 (ИСО 5807-85) «Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения».
- Все графики и графические изображения, отражающие какие-либо функциональные зависимости, должны содержать подписи по всем обозначенным осям (абсцисс, ординат, аппликат), включающие в себя: название; единицу измерения соответствующей величины; числовые отметки на оси, говорящие о масштабе отображаемой величины.
- Если рисунок или таблица размещается на странице (страницах), для которой задана «альбомная» ориентация листа, то их размещение должно быть таким, чтобы правильное расположение объекта относительно читателя достигалось поворотом страницы из нормального положения документа на 90° по часовой стрелке.
- Размещаемые в отчете рисунки не должны вызывать затруднений для рассмотрения, а текстовая информация на рисунках должна быть читабельной.
- Титульный лист отчета оформляется в соответствии с **приложением А**.

7. Требования к электронной презентации

Электронная презентация по выполненной **КР** является обязательными отчетными материалом только при выполнении **КР** командой из двух или более учащихся. Она создаётся для освещения и иллюстрации наиболее важных аспектов работы, выполненных разработок и полученных результатов.

Форма выступления – доклад.

Время сообщения: 5÷7 мин.

Рекомендуемый объем электронной презентации: 7÷10 слайдов.

Каждый слайд презентации, кроме титульного, должен иметь номер.

В электронной презентации необходимо отразить следующие аспекты выполненной **КР**:

– титульный слайд, на котором должны быть представлены:

- ♦ официальный логотип НИТУ МИСиС,
- ♦ официальный логотип и название кафедры,
- ♦ название учебной дисциплины;
- ♦ фамилии и инициалы учащихся – членов команды разработчиков с указанием номеров студенческих групп,
- ♦ название темы **КР**.

– предметная область **КР**;

– формулировка содержательной постановки задачи на разработку программной системы;

– информация о выбранных и использованных технологиях и моделях баз данных, языках программирования и прочих методов и средств **ИТ**;

– сведения о проведенных разработках;

– подробная информация об основном результате **КР** с иллюстрациями его работоспособности;

- структура и примеры исходных данных (если есть) и те их особенности и свойства, которые были значимы (влияли) в ходе разработки;
- дополнительный материал (при необходимости и реальном наличии в КР; например, использованный научный аппарат, методы и модели машинного обучения, искусственного интеллекта, математического моделирования и т.п.);
- выводы.

Цветовая схема слайдов, шрифтовой набор и компоновка представляемых на слайдах информационных объектов (текст, рисунки, диаграммы, таблицы, формулы и проч.) не должны препятствовать быстрому и точному восприятию изображенной на них информации.

Институт ИТКН

Кафедра инженерной кибернетики

Направление подготовки: 01.03.04 прикладная математика

Квалификация (степень): **бакалавр**

КУРСОВАЯ РАБОТА

на тему:

...название темы курсовой работы...

учебная дисциплина

«Специальные главы по базам данных»

VIII семестр, 2024 – 2025 уч. год

Учащийся: *Фамилия И.О. (ак. группа),*

Фамилия И.О. (ак. группа),

Преподаватель: доц., к.т.н. А.С. Кожаринов

Оценка: _____

Дата защиты: _____

Москва 2025
