

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ НОВЫЙ УНИВЕРСИТЕТ»
(АНО ВО «РОСНОУ»)

Институт информационных систем и инженерно-компьютерных технологий

Проектирование информационных систем
методические указания
к курсовому проекту

Введение

Методические указания предназначены для студентов третьего и четвертого курсов специальностей «Прикладная информатика (в экономике)» и «Информационные системы» всех форм обучения. Они помогают выполнить и оформить документацию к курсовому проекту.

Курсовой проект имеет целью научить студентов самостоятельно применять полученные знания для комплексного решения конкретных теоретических и практических задач, привить навыки самостоятельного проведения научных исследований.

Курсовой проект по дисциплине «Проектирование информационных систем» предназначен для обучения студентов разработке законченных модулей информационных систем, начиная с описания предметной области выбранного объекта и заканчивая реализованными базой данных и необходимыми пользовательскими интерфейсами.

За время работы над курсовым проектом студент получает практические навыки ведения проекта и оформления сопутствующей документации, умения создавать и анализировать модели баз данных, использования структурного метода проектирования, работы в специализированном CASE-средстве Erwin [1].

Темы курсовых проектов разрабатываются преподавателями в соответствии с основным содержанием учебной дисциплины и согласовываются со студентом по его желанию, рассматриваются и утверждаются на заседании кафедры.

Для выбора темы курсового проекта по дисциплине «Проектирование информационных систем» необходимо определить область деятельности, которая наиболее хорошо знакома и интересна студенту. Тема курсового проекта является индивидуальной, т.е. если несколько студентов выбирают одну и ту же область деятельности, то для каждого из них должны быть указаны разные функции этой области деятельности. Примеры тем курсового проекта приведены в Приложении 1.

На курсовой проект кафедра разрабатывает задание. Для руководства работой студентов назначаются руководители из числа преподавателей кафедры, ведущих данную дисциплину.

Заданием на курсовой проект является проектирование модуля информационной системы по выбранным функциям определенной области деятельности с помощью метода ER-диаграмм и CASE-средства ERwin.

Курсовой проект выполняется студентами в часы самостоятельной работы. По выполнении курсового проекта, после его проверки преподавателем, составляется письменная рецензия.

Проект считается выполненным, если

- создана база данных в соответствии с требованиями метода ER-диаграмм (на электронном носителе);
- спроектированы и реализованы пользовательские интерфейсы, в том числе и Web-интерфейсы (на электронном носителе);
- подготовлена и оформлена в виде пояснительной записки вся необходимая документация.

Курсовой проект выносится на открытую защиту перед преподавателем (комиссией). В ходе защиты студент демонстрирует и доказывает работоспособность проекта и его экономическую привлекательность. По результатам его защиты студенту выставляется оценка.

При получении неудовлетворительной оценки они выполняют работу по новой теме или перерабатывают прежнюю в сроки, устанавливаемые деканом факультета.

Содержание пояснительной записки к курсовому проекту.

Введение.

1. Описание предметной области.
2. Проектирование базы данных.
 - 2.1. Этап концептуального проектирования.
 - 2.1.1. Описание сущностей.
 - 2.1.2. Описание связей.
 - 2.1.3. Концептуальная модель данных.
 - 2.2. Этап логического проектирования.
 - 2.2.1. ER-диаграмма в среде ERwin.
 - 2.2.2. Анализ ER-диаграммы.
 - 2.2.3. Окончательная ER-диаграмма.
 - 2.3 Этап физического проектирования.
 - 2.3.1. Генерация базы данных
 - 2.3.2. Схема данных в среде выбранной СУБД.
3. Проектирование пользовательских интерфейсов.
 - 3.1 Список требуемых транзакций.
 - 3.2 Анализ транзакций на этапе логического проектирования.
 - 3.3 Документация на пользовательские интерфейсы.
 - 3.3.1. Постановка задачи.
 - 3.3.2. Исходные данные.
 - 3.5.2.1. Переданные из БД.
 - 3.5.2.2. Введенные вручную.
 - 3.5.2.3. Справочные константы.
 - 3.3.3. Алгоритм решения.
 - 3.3.4. Макет интерфейса.
 - 3.4. Реализация транзакций средствами выбранной СУБД.
 - 3.5. Анализ транзакций на этапе физического проектирования.
4. Заключение.

Комментарии к содержанию пояснительной записки.

Введение.

Во введении студент в произвольной форме описывает выбранный объект, анализирует основные функции его деятельности, выделяет те функции и тех пользователей, для которых будет проектироваться информационная система. Желательно также пояснить причину выбора конкретного объекта и функций.

1. Описание предметной области.

В произвольной форме на языке деловой прозы необходимо подробно изложить, как и кем выполняются заявленные функции, какие бизнес-правила существуют на выбранном объекте, относящиеся к выполнению выбранных функций, какие входные документы используются, какие выходные документы формируются и куда направляются. В описании предметной области приводятся примеры всех документов, которые используются для реализации выбранных функций.

2. Проектирование базы данных.

Для проектирования базы данных используется одна из нотаций структурного метода – метод ER-диаграмм:

- стандарт Чена - для построения концептуальной модели данных
- стандарт IDEF1X - для CASE –средства ERwin.

2.1. Этап концептуального проектирования.

Задачей этапа концептуального проектирования БД является создание формализованного описания данных на основе описания предметной области – концептуальной модели данных (КМД). Создание КМД позволит автоматизировать процесс проектирования, давая возможность использовать различные CASE – средства.

2.1.1. Описание сущностей.

На этом шаге необходимо из описания предметной области выделить и описать все сущности.

Сущность – объект или концепция, которая характеризуется на данном предприятии как имеющая определенное существование.

Другими словами, из описания предметной области выделяются все существительные и устойчивые словосочетания. Далее необходимо определить, для каких из выделенных существительных ответ будет положительным на следующие вопросы:

- важно ли это существительное для выполнения заявленной функции данного объекта?
- имеет ли данное существительное дополнительное описание, которое требуется знать для реализации заявленной функции данного объекта?

2.1.1. Описание связей.

На этом шаге необходимо найти все связи, существующие на проектируемом объекте и имеющие отношение к выделенной функции между описанными в таблице № 2.1 сущностями. Для этого заполняется таблица № 2.2.

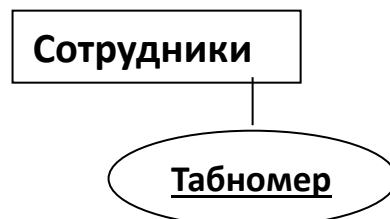
Связь – осмысленная ассоциация между разными сущностями.

2.1.2. ER-диаграмма.

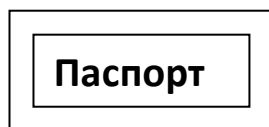
Используя данные таблиц № 2.1 и №2.2, создается концептуальная модель данных с использованием метода ER-диаграмм.

Для создания ER-диаграммы Чена:

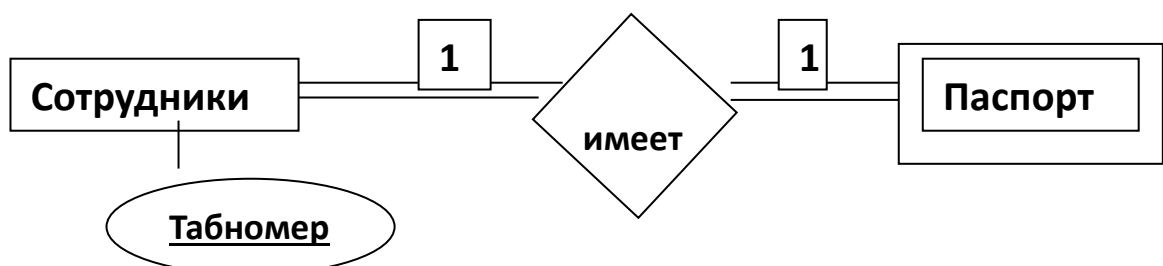
- сильные сущности отображаются прямоугольниками с именем сущности внутри прямоугольника с обязательным указанием первичного ключа, который находится в овале, связанном с прямоугольником. Имя первичного ключа в овале подчеркивается для отличия первичного ключа от обычного атрибута. Например,



- слабые сущности отображаются двойными прямоугольниками с именем сущности внутри внутреннего прямоугольника без указания первичного ключа. Например,



- связь отображается ромбом с именем связи внутри. Ромб линиями соединяется с прямоугольниками связанных сущностей. Одна линия – частичная степень участия, две линии – полная степень участия. Над каждой линией ставится показатель кардинальности. Например,



2.2. Этап логического проектирования.

Задачей этапа логического проектирования БД является создание такой логической модели данных, которая, с одной стороны, соответствует описанию предметной области, а с другой, отвечает всем требованиям целевой СУБД, которая выбирается (или используется как существующая) на этом этапе проектирования.

2.2.1. ER-диаграмма в среде ERwin.

Для построения ER-диаграммы в среде ERwin (стандарт IDEF1X) используют данные из таблиц № 2.1 и 2.2 или КМД, построенную на этапе концептуального проектирования.

2.2.2. Анализ ER- диаграммы.

Целью данного анализа является преобразование полученной ранее модели в соответствии с требованиями реализации существующих типов СУБД. В строгом понимании эти действия не являются элементами логического проектирования, но эта процедура заставляет разработчика более тщательно обдумать смысл каждого элемента данных. На этом этапе необходимо проанализировать следующие «нежелательные», с точки зрения многих СУБД, элементы.

- Многозначные атрибуты – меняются на сущность с именем многозначного атрибута и связь с показателем кардинальности «1 x M». См. рис. 2.13 – атрибут «телефон» сущности «Клиент» заменен на сущность «Телефон». Обратите внимание, что, в сущности, «Клиент» такого атрибута уже нет.
- Производные атрибуты – удаляются из логической модели с обязательным указанием всех производных атрибутов в таблице №2.1.
- Рекурсивные связи – возможно требуют добавления сущности или сущности-подкласса и связи с показателем кардинальности «1 x M».
- Связи с показателем кардинальности «1 x 1» - требуют дополнительного анализа, действительно ли это две разные сущности или возможно объединение в одну сущность.
- Избыточная связь – связь, соединяющая две сущности, соединенные друг с другом набором других связей и не несущая дополнительных данных. Обычно на этом этапе удаляется до 80% избыточных связей. На рис. 2.13 видно, что связь «заклучают» заменена на связи «менеджер-договор» и «оператор-договор» как несущие дополнительные данные.
- Связи с показателем кардинальности «M x N» - анализируются на наличие собственных атрибутов.

Все проведенные изменения обязательно фиксируются. Измененная ER- диаграмма является результатом данного этапа проектирования и считается окончательной ER- диаграммой. Например, ER- диаграмма на этом этапе может принимать вид, как на рис. 2.12.

2.3. Этап физического проектирования.

Этап физического проектирования всегда тесно связан с особенностями конкретной выбранной СУБД.

2.3.1. Генерация базы данных.

На этапе физического проектирования в ER-диаграмме для всех атрибутов уточняются все типы данных, чтобы убедиться в их применении в выбранной среде реализации. Для этого в CASE-средстве Erwin достаточно выбрать физический этап проектирования в пиктографическом меню (см. рис. 2.14). Все имеющиеся связи с показателем кардинальности «М x N» раскрываются в ассоциативные таблицы. Чтобы получить ассоциативную таблицу, необходимо поставить курсор на связь с показателем кардинальности «М x N» и нажать на правую кнопку мыши, выбрать из всплывающего меню строчку "Create Associative Table" и последовательно нажимать «ОК» во всех диалоговых окнах.

3.Проектирование пользовательских интерфейсов.

Целью данного этапа является проектирование интерфейса пользователя и прикладных программ, предназначенных для работы с базой данных в рамках его должностных инструкций.

Необходимо помнить, что в жизненном цикле информационных систем этот этап выполняется параллельно с этапом проектирования базы данных с постоянным обменом информации.

Необходимо убедиться, что все заявленные требования пользователей будут выполняться и поддерживаться создаваемой базой данных.

3.1. Список требований пользователей.

На данном этапе необходимо зафиксировать всех пользователей будущей информационной системы и выписать их функциональные требования в рамках заявленной к проектированию функции. Пользователи, имеющие разные должностные инструкции, но выполняющие одинаковые задачи (или разностью в выполнении задач можно пренебречь) по реализации заявленной к проектированию функции, называются **типом пользователя**.

Функциональные требования типа пользователя необходимо проанализировать и записать таким образом, чтобы они представляли специализацию транзакций. **Транзакция**

– одно или несколько неделимых действий над базой данных, выполняемых одним типом пользователей. Например:

- Менеджер:
 1. Список всех операторов.
 2. Перечень всех договоров конкретного менеджера за конкретный месяц.
 3. Поиск информации об операторе по его ФИО.
- Оператор:
 1. Ввод нового договора.
 2. Поиск объекта по цене.
 3. Ввод нового клиента.
 4. Поиск всех договоров конкретного клиента.

Список типов пользователей, а тем более список транзакций для каждого типа пользователя, должен соответствовать той функции или функциям, которые были заявлены для проектирования информационной системы. Например, если заявлена функция продаж, возможны следующие транзакции:

- Продавец:
 1. Поиск товара по названию и цене «от» - «до»
 2. Поиск товара по марке-производителю.
 3. Формирование чека.
 4. Список всех своих чеков за период.

Невозможны в этом случае транзакции:

- Продавец:
 1. Ввод нового товара – относится к описанию функции поставок.
 2. Список всех чеков по фамилии конкретного сотрудника - так как данная транзакция не входит в перечень должностных инструкций продавца, а относится к деятельности менеджера отдела продаж.

Для включения в специализацию транзакций при выполнении курсового проекта следует избегать транзакций типа Delete (удаление) и типа Update (обновление).

3.2. Анализ транзакций на этапе логического проектирования.

Целью анализа транзакций является проверка того факта, что база данных, модель которой получена на этапе логического проектирования поддерживает необходимыми данными все заявленные транзакции пользователей.

Для этого мы используем модель данных, полученную на этапе логического проектирования в результате анализа, и попытаемся выполнить каждую транзакцию

вручную. Если это окажется возможным для всех транзакций, заявленных в списке транзакций, то можно считать, что данная логическая модель успешно проверена. Если же выполнить вручную некоторую из транзакций окажется невозможно, значит в логической модели данных присутствует ошибка, которую необходимо удалить. Вероятнее всего, в модели отсутствуют необходимая сущность, атрибут или связь.

Для примера рассмотрим анализ всех транзакций, заявленных менеджером.

1. Список всех операторов.

Для выполнения данной транзакции необходимо найти значение первичного ключа – номера данного менеджера среди значений данного атрибута сущности «Менеджеры». Если такой номер не будет найден, необходимо вывести сообщение о том, что такого менеджера не существует. Если будет найден, далее необходимо найти связь с сущностью «Операторы». Такая связь есть через сущность «Персонал», но эта связь не позволяет определить, кто из персонала подчиняется данному менеджеру. Данную транзакцию выполнить нельзя. Необходимо добавить связь «Менеджеры» - «управляют» - «Операторы», показатель кардинальности которой «1 x M». В этом случае по этой связи мы найдем данные обо всех операторах в сущности «Операторы», у которых значение атрибута «Таб_ном_мен» равен заявленному.

2. Перечень всех договоров конкретного менеджера за конкретный месяц.

Для выполнения данной транзакции необходимо найти значение первичного ключа – номера данного менеджера в сущности «Персонал», далее найти значения всех атрибутов в сущности «Договор», где значение атрибута «менеджер» совпадает с заданным и значение атрибута «Дата_закл» попадает в границы заданного месяца. Данная транзакция может быть выполнена.

3. Поиск информации об операторе по его ФИО.

Для выполнения данной транзакции необходимо найти заданное значение атрибута «ФИО» в сущности «Персонал». Если такого значения нет, необходимо вывести сообщение об ошибке, если таких значений одно или больше, а это возможно, так как атрибут «ФИО» не является первичным ключом, и для всех этих значений необходимо найти значения первичных ключей «Таб_ном» и всех остальных атрибутов, так как они содержат данные об искомых операторах, и далее найти значения всех атрибутов в сущности-подклассе «Операторы», где значения первичных ключей совпадают с найденными. Данная транзакция может быть выполнена.

В курсовой проект достаточно включить только визуальное отображение анализа не более 15 транзакций, наиболее важных для заявленной к проектированию

функции, то есть тех транзакций, которые поддерживают работу пользовательского интерфейса (ПИ). Для этого необходимо сначала выполнить проект макета ПИ.

3.3. Документация на пользовательские интерфейсы.

Для каждого заявленного типа пользователя приводится список основных должностных инструкций. Далее создается макет ПИ для каждого типа пользователя, позволяющий в удобной и понятной пользователю форме реализовать эти функции. В курсовой проект достаточно включить пользовательский интерфейс только для одного типа пользователя, в рамки деятельности которого входит реализация заданной к проектированию функции. ПИ должен позволять реализовывать все должностные инструкции пользователя, и только их.

В рамках курсового проекта разрабатывается диалоговый пользовательский интерфейс, содержащий сценарий деятельности конкретного пользователя.

Документация на пользовательские интерфейсы содержит следующие разделы:

3.3.1. Постановка задачи.

При постановке задачи необходимо указать какой тип пользователя будет работать и какие действия будет реализовывать в данном ПИ. Например:

ПИ обеспечивает деятельность оператора по заключению договора на аренду жилья:

- Поиск или ввод клиента;
- Поиск объектов жилья по требованию клиента:
 - ✓ По ближайшей станции метро и/или цене «от»-«до»;
 - ✓ По количеству комнат и/или метражу жилой площади;
 - ✓ По наличию телефона и/или мебели;
- Оформление договора.

3.3.2. Исходные данные.

Исходные для ПИ данные делятся на:

3.3.2.1. Переданные из БД.

Обычно в ПИ оформляются как поля ввода (текстовые поля). Необходимо перечислить все поля ПИ, которые содержат данные из БД и указать из какой таблицы (поле) они берутся. Например:

- Поле «Полный адрес объекта» берется из таблицы Объект(Адрес)
- Поля ФИО, дата рождения, прописка клиента – таблица Клиент(ФИО, Д_рожд, адрес)
- Поля Паспортные данные – таблица Паспорт(номер, серия, кто, когда)
- Поле Стоимость – таблица Объект(ст-ть)

- Количество месяцев.

3.3.2.2. Введенные вручную.

Обычно в ПИ оформляются как поля ввода (текстовые поля). Необходимо перечислить все поля ПИ, которые вводятся вручную. Например:

- ФИО клиента
- Номер и серия паспорта
- Цена «от»
- Цена «до»
- Начало аренды
- Конец аренды.

3.3.2.3. Справочные константы.

Обычно в ПИ оформляются как метки, поля ввода (текстовые поля) с уже введенными данными, поля со списком выбора, метки с «переключателем». Необходимо перечислить все поля ПИ, которые содержат справочную информацию и ее источник. Например:

- ближайшая станция метро – список всех станций метрополитена.

3.3.3. Алгоритм решения.

Если в ПИ проводятся какие-либо вычисления, необходимо пояснить их алгоритм либо в виде формул, либо в виде блок-схемы, либо в виде текста с пояснениями. Например:

$$\begin{aligned} \text{стоимость по договору} &= \text{стоимость} * \text{количество месяцев} \\ \text{количество месяцев} &= \text{конец аренды} - \text{начало аренды.} \end{aligned}$$

Необходимо заметить, что в правой части формулы должны содержаться только данные из пункта 3.3.2, в противном случае их необходимо пояснить следующей формулой.

3.3.4. Макет интерфейса.

Если ПИ имеет только одну экранную форму – представлять нужно только ее, если несколько – представлять нужно все формы с указанием условий переходов и возвратов от одной формы к другой. Например:

Рис. 3.2. Макет пользовательского интерфейса.

Оператор. Работа с клиентами.

Выберите ФИО оператора Дата

Поиск клиента

Введите номер паспорта и серию паспорта **Найти**

Результат поиска

Номер клиента телефон Адрес

Добавить клиента **Просмотреть список договоров клиента**

	Номер договора	Дата заключения	Срок	Стоимость	Номер объекта	Адрес
*						

Поиск квартиры по любому набору параметров.

по ближайшей станции метро

цене от руб. до

Количество комнат

Количество метров жилой площади кв.м.

тип дома **Найти** Количество найденных квартир -

	номер объекта	адрес	кол-во комнат	тип дома	стоимость руб./мес	Общая пл	Жилая пл.
*							

Заключение договора.

Дата начала аренды: . . 2010 г.

Срок мес.

Стоимость в мес.

Номер договора

Заключить договор **Распечатать договор**

Количество договоров за день : **на сумму :** руб.

3.3.5. Перечень всех управляющих элементов макета.

Необходимо перечислить все управляющие элементы, которые используются в макете ПИ и зафиксировать действия, которые будут выполняться при использовании этих элементов. Удобнее всего это сделать в табличной форме. Например:

Таблица №3.1 Описание управляющих элементов.

Номер управляющего элемента	Имя элемента	Какие действия выполняются
ComboBox1		Позиционирование конкретного оператора
Button1	Найти	Поиск данных о клиенте по номеру и серии паспорта
Button2	Добавить клиента	Добавление данных о новом клиенте
Button3	Просмотреть список договоров клиента	Просмотр всех договоров найденного клиента
ComboBox2		Выбор станции метро из всего списка станций

ComboBox3		Выбор количества комнат из возможного в компании списка (1,2,3,4)
ComboBox4		Выбор типа дома из возможного в компании списка.
Buttom4	Найти	Поиск варианта квартиры по одному или любому набору представленных параметров.
ComboBox5		Выбор месяца
Buttom5	Заклучить договор	Добавление данных нового договора. В результате добавления в окне «Номер договора» автоматически появится следующий номер договора. В окне «Количество договоров за день» происходит увеличение на 1. В окне «на сумму» происходит увеличение на сумму добавленного договора.
Buttom6	Распечатать договор	Происходит распечатка шаблона договора, хранящегося в MSWord с добавлениями полей таблицы «Договор»

3.3.6. Программный код.

Макет ПИ реализуется на любом объектно-ориентированном языке программирования. Программный продукт сохраняется на любом носителе, который прилагается к пояснительной записке. В пояснительной записке указывается имя файла, где записан программный код. Распечатка программного кода реализации ПИ на любом объектно-ориентированном языке программирования добавляется к документации.

В рамках работы над курсовым проектом полная реализация ПИ не является обязательным этапом и оговаривается преподавателем. Возможна либо частичная реализация, либо полное отсутствие реализации.

3.4. Реализация транзакций средствами выбранной СУБД.

В специализацию транзакций необходимо добавить все транзакции, поддерживающие ПИ, если они еще не включены. Часть этих транзакций можно реализовать внутренними средствами СУБД.

В курсовой проект включаются 10 реализованных средствами СУБД транзакций. Описание реализованных транзакций делается в виде таблицы (см. таблицу №3.2)

Таблица №3.2. Описание реализованных в СУБД MS Access транзакций.

Имя или номер транзакции по спецификации транзакции	Форма реализации	Имя реализации.
T12. Выбор сотрудников с должностью оператор	Запрос	Операторы

T27. Список договоров, содержащий информацию об арендованных объектах	Сохраняемый запрос	Список_договоров
---	--------------------	------------------

В первой колонке не обязательно указывать саму транзакцию, как приведено в таблице. Достаточно указать ее номер по спецификации.

3.5. Анализ транзакций на этапе физического проектирования.

Целью анализа является определение функциональных характеристик транзакций, которые будут выполняться в базе данных, и выделение наиболее важных из них.

Для того чтобы разрабатываемый физический проект базы данных обладал требуемым уровнем эффективности, необходимо получить максимум сведений о тех транзакциях, которые будут выполняться в проектируемой базе данных. Нам потребуются как качественные, так и количественные характеристики. Для каждой транзакции необходимо знать следующее:

- ожидаемая частота выполнения транзакций;
- отношения и атрибуты, к которым потребуется иметь доступ при выполнении транзакции, а также тип этого доступа (R – выборка, I – вставка, U – обновление, D – удаление);
- ограничения, устанавливаемые на время выполнения транзакций.

Во многих случаях проанализировать все транзакции просто невозможно, поэтому необходимо выбрать из них самые «важные». В схеме, на которой проводился анализ транзакций на этапе логического проектирования (см. рис. 3.1), надо установить, какие из отношений наиболее интенсивно используются при выполнении транзакций. Для этого необходимо посчитать количество входящих в каждую сущность стрелочек. Например, в нашем случае получается следующее:

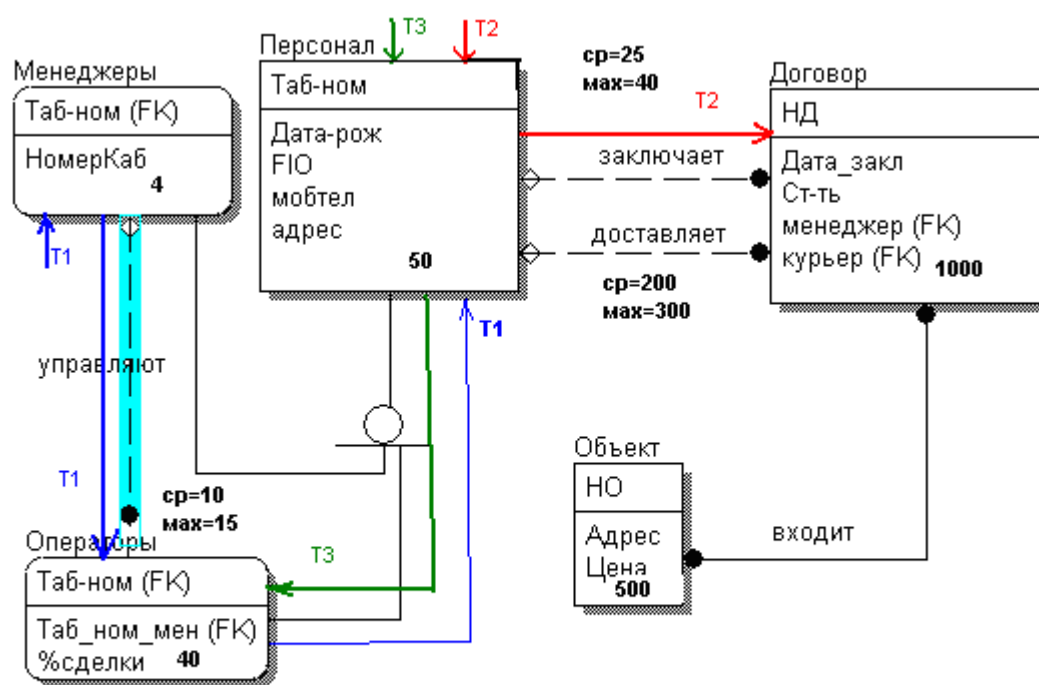
Сущности	Количество входящих стрелочек
Персонал	3
Менеджеры	1
Операторы	2
Договор	1

Так как в нашем примере анализировалось небольшое количество транзакций, то количество входящих стрелочек для всех сущностей примерно одинаково. Обычно 2-3 сущности имеют значительно большее количество, чем все остальные. На этапе

физического проектирования целесообразно анализировать только те транзакции, которые включают обращения к отношениям с большим количеством входящих стрелочек. В нашем случае это сущности «Персонал» и «Операторы», через которые проходят все заявленные в нашем примере транзакции. Обычно остается для анализа на физическом этапе проектирования около 80% всех транзакций.

Далее необходимо указать ожидаемое количество строк в отношениях, а также среднюю и максимальную кратности каждой связи (см. рис. №3.3). Например, ожидается, что персонал компании составит 50 человек, 4 из которых менеджеры, 40 операторов. Компания владеет 500 объектами жилья, на которые заключается 1000 договоров.

Рис. № 3.3. Указание ожидаемой размерности отношений и связи.



При анализе каждой из транзакций очень важно знать не только среднее и максимальное количество ее вызовов в час, но и иметь сведения о тех днях недели и часах суток, когда она обычно выполняется, включая и данные о времени пиковой нагрузки. Например, частота вызова некоторых транзакций может удерживаться на некотором уровне постоянно, но все же она имеет четко выраженный пик нагрузки в последний четверг месяца с 14-00 до 16-00, вызванный подготовкой отчетов. Другие транзакции вообще могут выполняться только в определенные моменты времени, например - по понедельникам с 9-00 до 10-00, что также является пиком нагрузки.

Когда выполнение потока транзакций требует частого доступа к определенным отношениям, очень большое значение приобретают выбранные для них схемы выполнения. Если выполнение этих транзакций не зависит друг от друга, риск возникновения проблем с производительностью уменьшается. Однако если схемы их выполнения конфликтуют, возможные проблемы могут быть частично устранены посредством тщательного изучения транзакций, что позволит найти способ их модификации с целью повышения их производительности. В нашем случае необходимо рассмотреть транзакции, проходящие через оператора и персонал. Результат анализа заносят в таблицу 3.3.

Таблица № 3.3. Результаты анализа.

Транзакция	Актив-ность	День недели	Время суток	Частота вызовов в месяц
Т1. Список всех операторов	Пиковая	-	-	-
	Средняя	Понедельник	9-00 – 12-00	1
От отношения	К отношению	Атрибуты	Тип доступа	Частота вызовов в месяц
-	Менеджер	Таб-ном	R(E)	1
Менеджер	Оператор	Таб-ном-мен Таб-ном	R(E)*	10 - 15
Оператор	Персонал	Таб-ном *	R(E) R	
Транзакция	Актив-ность	День недели	Время суток	Частота вызовов в месяц
Т2. Перечень всех договоров конкретного менеджера за конкретный месяц.	Пиковая	Последний четверг месяца	9-00 – 12-00	4
	Средняя	-	-	-
От отношения	К отношению	Атрибуты	Тип доступа	Частота вызовов в месяц
-	Персонал	Таб-ном	R(E)	4
Персонал	Договор	Таб-ном *	R(E)* R	800 - 1200
Транзакция	Актив-ность	День недели	Время суток	Частота вызовов в месяц
Т3. Поиск информации об операторе по его ФИО	Пиковая	Последний четверг месяца	15-00 - 16-00	40
	Средняя	Понедельник-пятница	Случайным образом	20
От отношения	К отношению	Атрибуты	Тип доступа	Частота вызовов в месяц
-	Персонал	ФИО *	R(E) R	60

4. Заключение.

В заключении к курсовому проекту необходимо в краткой форме подвести выводы по проделанной работе, которые могут включать:

- Мнение студента о проделанной работе;
- Отношение к изученному материалу;
- Оценку средств и методов проектирования.

Пример оформления пояснительной записки к курсовому проекту

Автономная некоммерческая организация высшего образования
«Российский новый университет»

Институт информационных систем и инженерно-компьютерных технологий

Курсовая работа/курсовой проект

по дисциплине: «Проектирование информационных систем»

на тему:

студента (ки) _____ курса
_____ формы обучения
направления подготовки
_____ направленности (профиля)
_____.
_____.

(фамилия, имя, отчество)

Руководитель:

Работа допущена к защите «___» апреля
2021 г.

Оценка _____

Регистрационный номер _____ от «___» _____ 2021 г.

г. Москва, 2021 г.

