



С. А. Жданов, М. Л. Соболева, А. С. Алфимова

Информационные системы

МЕДИАТЕЛЕВИЗНОЕ
ПРОМЕТЕЙ



УДК 004.03(075.8)

ББК 32.973.202

Ж 242

Научный редактор: академик РАН, академик РАО, доктор физико-математических наук, профессор *В.Л. Матросов*

Рецензенты: первый проректор Московского института открытого образования, доктор педагогических наук, профессор *С.Д. Каракозов*; профессор кафедры теоретической информатики и дискретной математики Московского педагогического государственного университета, доктор физико-математических наук, профессор *В.А. Горелик*

Ж 42 Информационные системы: учебник для студ. учреждений высш. образования / С.А. Жданов, М.Л. Соболева, А.С. Алфимова. — М.: ООО «Прометей», 2015. — 302 с.

Учебник создан в соответствии с Федеральным государственным образовательным стандартом по направлению подготовки 050100. «Педагогическое образование» профиль «Информатика» (квалификация «бакалавр»).

Рассматриваются вопросы, связанные с основными понятиями теории информационных систем, этапами и методологиями структурного анализа и проектирования и объектно-ориентированного подхода к разработке, проектированию и реализации информационных систем; методами хранения и доступа к данным.

Затрагиваются вопросы основ языка структурированных запросов SQL, архитектуры информационных систем, технологии «клиент-сервер», возможностей и особенностей различных систем баз данных, а также перспектив развития систем управления базами данных.

Для студентов учреждений высшего образования.

ISBN 978-5-9906-2644-7

© Издательство «Прометей», 2015

ОГЛАВЛЕНИЕ

Предисловие	5
1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ УЧЕБНОЙ ДИСЦИПЛИНЫ «ИНФОРМАЦИОННЫЕ СИСТЕМЫ»	12
<i>Лекция 1.</i> Основные понятия теории информационных систем. Введение в теорию баз данных	12
<i>Лекция 2.</i> Проектирование информационных систем. Инфологическое проектирование	17
<i>Лекция 3.</i> Даталогическое проектирование	21
<i>Лекция 4.</i> CASE-средства. Структурный подход к разработке и проектированию информационных систем	23
<i>Лекция 5.</i> CASE-средства. Объектно-ориентированный подход к разработке и проектированию информационных систем	31
<i>Лекция 6.</i> Язык структурированных запросов SQL	35
<i>Лекция 7.</i> Элементы теории транзакций	48
<i>Лекция 8.</i> Система управления базами данных Microsoft Access	57
<i>Лекция 9.</i> Система управления базами данных OpenOffice Base	67
<i>Лекция 10.</i> Архитектура информационных систем	125
<i>Лекция 11.</i> Перспективы развития СУБД. OLAP-технологии	127
2. СТРУКТУРНОЕ ПРЕДСТАВЛЕНИЕ ПОНЯТИЙНОГО АППАРАТА УЧЕБНОЙ ДИСЦИПЛИНЫ «ИНФОРМАЦИОННЫЕ СИСТЕМЫ»	136
3. ПРАКТИЧЕСКАЯ ЧАСТЬ УЧЕБНОЙ ДИСЦИПЛИНЫ «ИНФОРМАЦИОННЫЕ СИСТЕМЫ»	180
<i>Практическая работа № 1.</i> Создание базы данных «Студенты и задания» (СУБД MS Access)	180
<i>Практическая работа № 2.</i> Создание базы данных «Студенты» (СУБД MS Access)	190

<i>Практическая работа № 3.</i> Создание базы данных «Видеотека» (СУБД MS Access)	198
<i>Практическая работа № 4.</i> Создание базы данных «Библиотека» (СУБД MS Access)	203
<i>Практическая работа № 5.</i> Создание базы данных «Европа» (СУБД MS Access)	213
<i>Практическая работа № 6.</i> Создание главной кнопочной формы базы данных «Авторы и книги» (СУБД MS Access).	219
<i>Практическая работа № 7.</i> Создание функциональной модели с использованием методологии IDEF0 с помощью CASE-средств	232
<i>Практическая работа № 8.</i> Создание концептуальной модели (модели «сущность-связь») с использованием методологии IDEF1X с помощью CASE-средств	237
<i>Практическая работа № 9.</i> Проектирование и построение функциональной и концептуальной моделей выбранной предметной области (индивидуальный проект)	240
<i>Практическая работа № 10.</i> Создание базы данных «Домашние животные» (СУБД MySQL)	242
<i>Практическая работа № 11.</i> Создание базы данных «Продукты» (СУБД MySQL)	256
<i>Практическая работа № 12.</i> Создание базы данных «Экзамены» (СУБД MySQL)	259

4. ПРИМЕРНЫЕ КОНТРОЛЬНЫЕ РАБОТЫ ПО ДИСЦИПЛИНЕ «ИНФОРМАЦИОННЫЕ СИСТЕМЫ»	265
Требования к проектному заданию по индивидуальной тематике	265
Пример отчета проектного задания по индивидуальной тематике по дисциплине «Информационные системы»	268
Примерные вопросы контрольного проверочного теста по дисциплине «Информационные системы»	288
Пояснительная записка к контрольному проверочному тесту по дисциплине «Информационные системы»	295
Словарь терминов	296
Список литературы	298

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ УЧЕБНОЙ ДИСЦИПЛИНЫ «ИНФОРМАЦИОННЫЕ СИСТЕМЫ»

Лекция 1.

Основные понятия теории информационных систем. Введение в теорию баз данных

Цель лекции: сформировать общее представление о теории информационных систем и раскрыть основные понятия данной теории.

Задачи:

1. Рассмотреть историю развития информационных систем.
2. Определить основные понятия теории информационных систем.
3. Проанализировать различные классификации информационных систем.
4. Сформировать понятия классической теории баз данных.

Существует две точки зрения на этапы развития информационных систем:

1. Точка зрения отечественных ученых (схема 3).
2. Точка зрения зарубежных ученых (схема 4).

В настоящее время большинство ученых в области информационных систем едины во мнении, что понятие «информационная система» определяется через такие понятия как «информация» и «система».

Информация – это совокупность фактов, явлений, событий, представляющих интерес, подлежащих регистрации и обработке [39].

Система – это объект, способный осуществлять хранение, обработку и передачу информации [39].

Информационная система – это совокупность:

1) функциональных процессов и связанных с ними информационных, специфичных в конкретной предметной области;

2) средств, способов и методов, направленных на создание, сбор, хранение, анализ, обработку и передачу информации, существенно зависящих от специфики применения области;

3) совокупность процессов по управлению решением функциональных задач, а также информационными, материальными и денежными потоками предметной области.

Существуют несколько классификаций информационных систем (схема 5), в основе которых лежат следующие критерии:

- цель функционирования (схема 6);
- характер процесса преобразования данных (схема 7);
- характерные функции управления данными (схема 8);
- сферы применения (схема 9).

Любая информационная система состоит из следующих компонентов (схема 10):

1. **База данных** – это ядро информационной системы, состоит из совокупности данных, которые имеют следующие свойства:

1) интегрированность, направленная на решение общих задач одной предметной области;

2) модельность, структурированность отражающие некоторую часть реального мира;

3) независимость описания данных от прикладных программ [6] (схемы 11-16).

2. **Система управления базами данных (СУБД)** – это комплекс языков и программ, позволяющий создать базу данных и управлять ее функционированием [39] (схемы 24-29).

3. **Администратор базы данных** – это специалист или группа специалистов, занятых обслуживанием базы данных. Координирует процессы сбора информации, проектирования и эксплуатации базы данных, обеспечивает защиту и целостность данных. Учитывает текущие и перспективные информационные потребности пользователя [24].

4. **Словарь данных** – это специальная программа базы данных, которая хранит единообразную и централизованную информацию обо всех ресурсах данных базы данных [36].

5. Вычислительная система – это персональные компьютеры и ноутбуки, соединенные каналами связи в вычислительную сеть [24].

6. Обслуживающий персонал – это пользователи информационной системы, от качественно выполняемых функциональных задач которых зависит четкость и правильность работы всей информационной системы (схемы 30-32).

Ядром любой информационной системы является база данных, поэтому рассмотрим более подробно это понятие.

Существует две наиболее распространённых **классификации баз данных**:

1. по функциональному назначению (схема 13);

2. по отношению к использованию технических средств (схема 14).

Любая база данных имеет модель данных.

Модель данных – интегрированный набор понятий для описания данных, связей между ними и ограничений, накладываемых на данные в некоторой организации [18] (схемы 16-19).

В классической теории баз данных рассматриваются три модели данных, основанных на записях (схема 19):

Реляционная модель – логическая модель данных, основанная на понятии математических отношений. Данные и связи представлены в виде таблиц, каждая из которых имеет столбцы с уникальными именами (схема 20);

Сетевая модель – логическая модель данных, основанная на представлении данных сетевыми структурами типов записей и связанных отношениями мощности один-к-одному или один-ко-многим (рис.1.1.1).

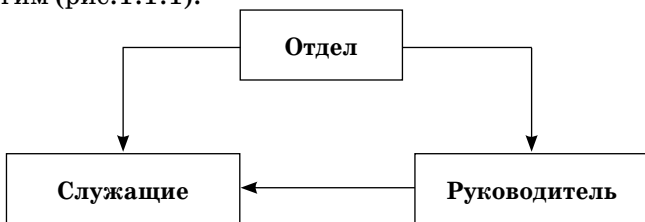


Рис.1.1.1. Пример схемы сетевой модели

Иерархическая модель – логическая модель данных в виде древовидной структуры, представляющая собой совокупность

элементов, расположенных в порядке иерархической упорядоченности, которые образуют перевернутое дерево (граф) (рис.1.1.2).

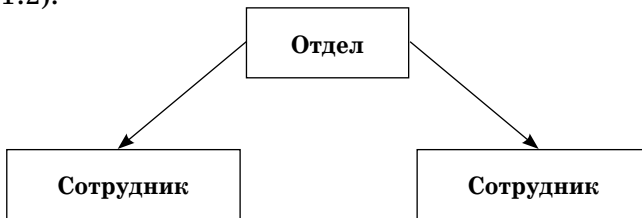


Рис.1.1.2 (а). Пример схемы иерархической модели данных «Отдел–Сотрудники»

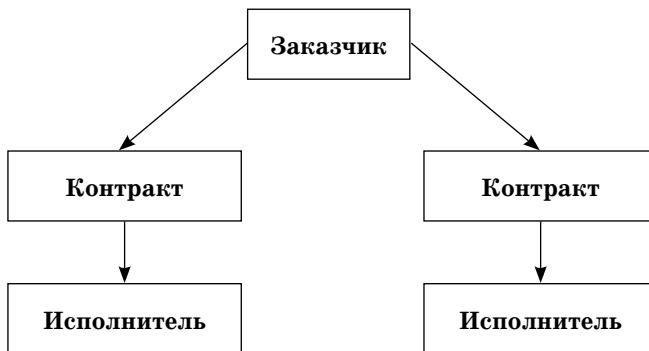


Рис.1.1.2 (б). Пример схемы иерархической модели данных «Заказчик–Контракт–Исполнитель»

В теории баз данных существует такое понятие как архитектура баз данных – это уровни представления данных (схема 15):

- уровень внешнего представления данных;
- уровень концептуального представления данных;
- уровень внутреннего (физического) представления данных.

Любую базу данных можно создать с помощью языка баз данных (схема 21). Языками баз данных являются язык структурированных запросов (SQL – Structured Query Language) (схема 22) и язык запросов по образцу (QBE – Query by Example) (схема 23).

Контрольные вопросы

1. Какие понятия являются базовыми для понятия «информационная система»?
2. Какие понятия являются компонентами информационной системы?
3. Какое понятие является ядром информационной системы? Из каких элементов состоит база данных?
4. Перечислите основные понятия реляционной модели данных? На каком понятии данная модель основана?
5. Сколько уровней в традиционной архитектуре базы данных?
6. Какие языки баз данных рассматриваются в теории базы данных?

Лекция 2.

Проектирование информационных систем. Инфологическое проектирование

Цель лекции: сформировать общее представление о процессе проектирования, как об одном из этапов жизненного цикла информационных систем. Структурировать и систематизировать знания об этапах проектирования информационных систем и процессе инфологического (концептуального) проектирования.

Задачи:

1. Определить понятие жизненного цикла информационных систем и сформировать общее представление об этапах жизненного цикла.
2. Рассмотреть этап проектирования жизненного цикла и его составляющие части: этап инфологического (концептуального) проектирования, этап даталогического проектирования, физическое проектирование, этап опытной эксплуатации.
3. Проанализировать процесс инфологического проектирования информационных систем.
4. Рассмотреть основные понятия этапов концептуального проектирования.

Жизненный цикл информационных систем определяется как период времени, который начинается с момента принятия решения о необходимости создания информационной системы и заканчивается в момент его полного изъятия из эксплуатации (IEEE Std 610.12 – 19990. IEEE Standart Glossary of Engineering Terminology).

Жизненный цикл информационных систем включает в себя несколько **этапов** (схема 33):

- предпроектное исследование предметной области;
- проектирование;

- тестирование;
- ввод в действие;
- эксплуатация и сопровождение;
- снятие с эксплуатации.

Самым важным этапом жизненного цикла информационных систем является этап проектирования, который тоже состоит из нескольких этапов (схема 35):

- этап инфологического (концептуального) проектирования;
- этап даталогического проектирования;
- этап физического проектирования;
- этап опытной эксплуатации.

Рассмотрим этап инфологического проектирования. Данный этап содержит пять подэтапов, которые необходимо выполнять последовательно при проектировании информационных систем (схема 36).

1. Процесс идентификации сущностей, основным понятием которого является «сущность».

Сущность (Entity) – реальный либо воображаемый объект, имеющий существенное значение для рассматриваемой предметной области [5].

Пример: сущность – ГОРОД, экземпляр сущности – МОСКВА.

2. Процесс определения атрибутов сущности, основным понятием которого является «атрибут».

Атрибут (Attribute) – любая характеристика сущности, значимая для рассматриваемой предметной области и предназначенная для квалификации, идентификации, классификации, количественной характеристики или выражения свойств, ассоциированных с множеством реальных или абстрактных объектов [5].

Пример: атрибут – ЦВЕТ можно отнести ко многим сущностям – СОБАКА, АВТОМОБИЛЬ, ДЫМ и т.д.

Экземпляр атрибута – это определенная характеристика отдельного элемента множества [5].

Пример: сущность – СОБАКА, атрибут этой сущности – ПОРОДА, экземпляр атрибута – СЕТТЕР.

3. Процесс установления связи между сущностями, основным понятием которого является «связь».

Связь (Relationship) – поименованная ассоциация между двумя сущностями, значимая для рассматриваемой предметной области [5].

4. Процесс нормализации модели, основными понятиями которого являются «нормализация», 1-я нормальная форма (НФ), 2-я НФ, 3-я НФ, НФ Бойса-Кодда, 4-я НФ, 5-я НФ.

Нормализация модели – это обратимый пошаговый процесс декомпозиции (разбиения) исходных отношений на более простые, при котором устраняются нежелательные зависимости (5).

Нормализация – процесс последовательного приведения модели проектируемой информационной системы к 1-й НФ, ко 2-й НФ, к 3-й НФ, к НФ Бойса-Кодда, к 4-й НФ и к 5-й НФ (схема 37).

Схема приведения 1-й НФ ко 2-й НФ представлена на рис. 1.2.1.

Пример приведен на рис. 1.2.2.

Схема приведения 2-й НФ к 3-й НФ представлена на рис. 1.2.3.

Пример приведен на рис. 1.2.4.

A*	} Ключевые	C	} Неключевые атрибуты, которые
B*		Д	
	атрибуты		

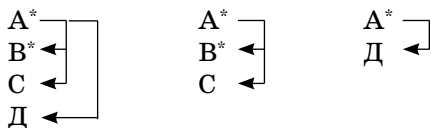


Рис.1.2.1. Схема приведения 1-й НФ ко 2-й НФ



Рис.1.2.2. Пример приведения 1-й НФ ко 2-й НФ

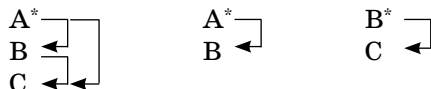


Рис.1.2.3. Схема приведения 2-й НФ к 3-й НФ

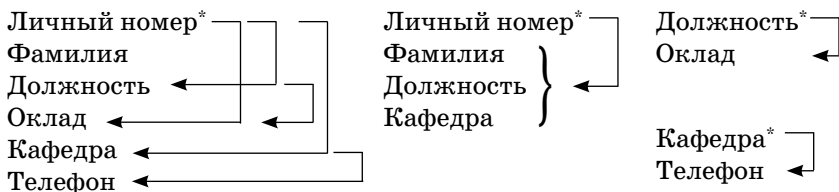


Рис.1.2.4. Пример приведения 2-й НФ к 3-й НФ

5. Процесс минимизации числа сущностей, основными правилами которого являются следующие:

1) если все атрибуты рассматриваемой сущности присутствуют в какой-либо другой, то рассматриваемая сущность является избыточной и должна быть ликвидирована;

2) если имеется несколько сущностей с одинаковыми первичными ключами, то необходимо эти сущности объединить;

3) если некоторая сущность-связь является всеми возможными проекциями одной сущности-связи, то сущности проекции могут объединить в одну сущность-связь.

Контрольные вопросы

1. Что такое жизненный цикл информационной системы?
2. Какие этапы проходит любая информационная система в своем жизненном цикле?
3. Из каких процессов состоит этап проектирования жизненного цикла информационной системы?
4. Какие этапы включает в себя инфологическое (концептуальное) проектирование?
5. Как определяют понятия «сущность», «атрибут», «связь»?
6. Что такое процесс нормализации модели проектируемой информационной системы? Чем отличается 1-я НФ от НФ Бойс-Кодда?
7. В чем заключается этап минимизации числа сущностей при проектировании информационной системы?

Лекция 3.

Даталогическое проектирование

Цель лекции: сформировать представление о даталогическом этапе проектирования информационных систем.

Задачи:

1. Проанализировать результаты построения информационной системы, полученные после этапа инфологического (концептуального) проектирования, и определить дополнительные требования к проектируемой информационной системе с учетом выбранного для ее написания программного средства.

2. Рассмотреть этапы даталогического проектирования после выбора инструментального средства.

3. Определить результаты даталогического проектирования.

Результатами инфологического проектирования и входными данными даталогического проектирования являются:

1) инфологическая (концептуальная) модель предметной области (СУБД-независимая схема);

2) характеристики одной или нескольких СУБД;

3) вычислительные средства – ограничения на конфигурацию и объем аппаратного и программного обеспечения;

4) количественная оценка эксплуатационных характеристик:

- спецификации требований целостности;
- спецификации требований безопасности;
- спецификации требований восстанавливаемости;
- спецификации требований ограничений на время отклика;
- спецификации требований прогноза объема роста и изменений структуры и архитектуры базы данных.

Дополнительные требования к проектируемой информационной системе на этапе выбора инструментального средства даталогического проектирования:

1) количественная оценка объема и частоты выполнения приложений:

- правила поддержания взаимной непротиворечивости элементов данных;
- правила устранения непротиворечивости данных, а также ограничения на дублирование и обновление данных.

2) требования к прикладным программам.

Процесс даталогического проектирования после выбора программно-инструментального средства состоит из следующих шагов:

- 1) определение локальных информационных структур;
- 2) формирование СУБД-ориентированной схемы;
- 3) оценка характеристик полученных СУБД-схем;
- 4) усовершенствование схемы с целью повышения ее эффективности.

В результате даталогического проектирования получают:

- СУБД-ориентированную схему с учетом спецификаций выбранной СУБД, ее структуры и архитектуры;
- пакет прикладных программ и руководство по их применению и проектированию;
- руководство для группы сопровождения базы данных, администратора и обслуживающего персонала.

Контрольные вопросы

1. В чем заключается этап даталогического проектирования?

2. Что необходимо разработчику информационной системы иметь перед началом процесса даталогического проектирования?

3. На какие шаги подразделяют этап даталогического проектирования после выбора инструментального средства разработки и реализации информационной системы?

Лекция 4.

CASE-средства. Структурный подход к разработке и проектированию информационных систем

Цель лекции: сформировать знания о структурном подходе к разработке и проектированию информационных систем при использовании CASE-средств.

Задачи:

1. Определить понятия CASE-технологии и CASE-средства.
2. Рассмотреть базовые и основные принципы структурного анализа и проектирования информационных систем.
3. Обозначить основные методологии структурного анализа и структурного проектирования информационных систем, проанализировать методологию SADT.
4. Проанализировать основные модели (диаграммы) структурного подхода к проектированию информационных систем.

CASE в переводе с английского расшифровывается как Computer Aided Software/System Engineering – проектирование программного обеспечения и программных систем с помощью компьютера.

CASE-технология – совокупность методов проектирования информационных систем, а также набор инструментальных средств, позволяющих в наглядной форме моделировать предметную область, анализировать эту модель на всех стадиях разработки и сопровождения информационных систем и разрабатывать приложения в соответствии с информационными потребностями пользователей [5].

CASE-средства – это инструментарий, позволяющий автоматизировать процесс проектирования и разработки информационных систем [5].

На сегодняшний день существуют три основных подхода к разработке и проектированию информационных систем (схема 38):

1. Структурный анализ и структурное проектирование (Structured Analysis and Structured Design – SA/SD) (схема 39).
2. Объектно-ориентированный подход (схема 44).
3. Интегрированный подход (схема 48).

В основу структурного анализа и структурного проектирования положен принцип функциональной декомпозиции, при которой структура системы описывается в терминах иерархии ее отдельных функций и передачи информации между отдельными функциональными элементами [5].

Существуют два **базовых принципа** структурного подхода (схема 40):

- 1) «разделяй и властвуй» – принцип деления (декомпозиции) на небольшие подсистемы, каждая из которых разрабатывается независимо от других;
- 2) принцип иерархической упорядоченности – принцип организации составных частей в иерархические древовидные структуры с добавлением новых деталей на каждом уровне.

Основными принципами данного подхода являются следующие (схема 41):

- 1) принцип абстрагирования;
- 2) принцип формализации;
- 3) принцип упрятывания;
- 4) принцип концептуальной общности;
- 5) принцип полноты;
- 6) принцип непротиворечивости;
- 7) принцип логической независимости;
- 8) принцип независимости данных;
- 9) принцип структурирования данных;
- 10) принцип доступа конечного пользователя.

Основные **методологии** структурного анализа и проектирования (схема 42): SADT, IDEF0, IDEF1X, IDEF3.

В контексте данной лекции рассматривается методология SADT, все остальные являются ее составными частями.

Методология SADT была разработана в 1973 году Дугласом Россом в SoftTech корпорации. Она успешно применялась в военных, промышленных и коммерческих организациях для решения разнообразных задач.

Методология SADT – это совокупность правил и процедур, предназначенных для построения функциональной модели (схемы) предметной области разрабатываемой информационной системы.

Функциональная модель (схема) отображает функциональную структуру объекта, то есть производимые им действия и связи между этими действиями [5].

Основными рабочими элементами функциональной модели (схемы) являются диаграммы, которые иерархически упорядочены (чем ниже уровень диаграммы, тем более она детализирована), а также блоки и дуги, входящие в состав диаграмм. Блоками обозначают функции проектируемой системы, а дугами – связи между ними. На одной диаграмме размещают от 3-х до 6-и блоков в виде ступенчатых схем. Дуги маркируются текстом на естественном языке. Каждая дуга блока имеет строго определенное значение, которое зависит от ее местоположения по отношению к блоку (рис.1.4.1).

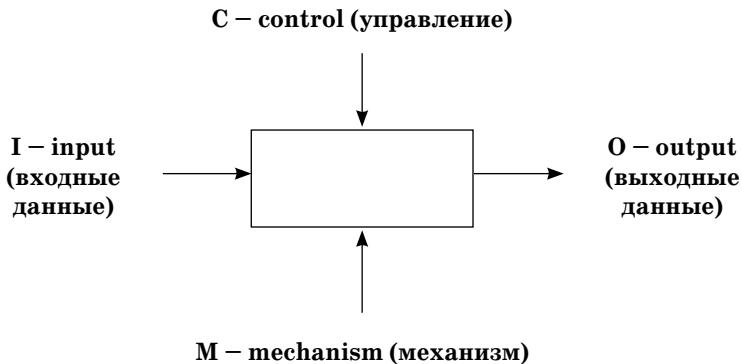


Рис.1.4.1. Кодировка внешних связей системы

ICOM (input, control, output, mechanism) – отражает определенные принципы функционирования системы. Входные данные преобразуются в выходные, управление ограничивает или предписывает условия выполнения, механизмы описывают, за счет чего выполняются преобразования [15, 16].

Основными **моделями (диаграммами)** структурного анализа и структурного проектирования (схема 43) являются DFD-модель (диаграмма), ERD-модель (диаграмма) и STD-модель (диаграмма).

DFD-модель (диаграмма) – (Data Flow Diagrams) – диаграмма потоков данных – является средством для моделирования функциональных требований к проектируемой системе. Требования представляют в виде иерархии функциональных процессов, связанных потоками данных. Основная задача DFD-модели (диаграммы) показать преобразования входных данных в выходные каждого процесса и обозначить отношения между процессами [5].

Для построения модели (диаграммы) потоков данных используют два метода Йордана и Гейна-Сэрсона. Методы Йордана и Гейна-Сэрсона используют графическое представление модели (диаграммы) DFD, которое имеет незначительные отличия в каждом методе.


Рассмотрим DFD-модель в рамках метода Йордана.

Основные символы:

1. **Поток данных** – это механизм для моделирования передачи информации из одной части системы в другую.

Обозначается: имя потока

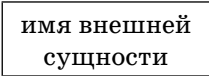
2. **Процесс** – это функция преобразования входных данных в выходные.

Обозначается: 

3. **Хранилище** – это данные, которые сохраняются в памяти компьютера между процессами.

Обозначается: имя хранилища

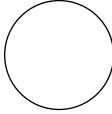
4. **Внешняя сущность** – это источник или приемник системных данных.

Обозначается: 

Для обеспечения декомпозиции данных в DFD-модели (диаграмме) существуют следующие типы объектов:

1. Групповой узел – служит для разъединения или соединения потоков.

Обозначается:



2. Узел-предок – необходим для детализации объектов проектируемой системы.

Обозначается:



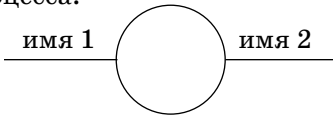
3. Неиспользованный узел – служит для обозначения элементов, не входящих в узел данного потока.

Обозначается:



4. Узел изменения имени – необходим при изменении имени какого-либо процесса.

Обозначается:



5. Текст – может находиться в любом месте DFD-модели (диаграммы).

Обозначается: естественным языком.

Рекомендации по построению диаграммы потоков данных:

- на диаграмме должно быть изображено от 3-х до 6-и процессов;
- декомпозиция потоков данных должна осуществляться параллельно с декомпозицией процессов;
- имена процессов и потоков должны быть ясными, отражающими их сущность.

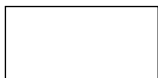
ERD-модель (диаграмма) – (Entity-Relationship Diagrams) – диаграмма «сущность-связь» – является концептуальной схемой базы данных проектируемой информационной системы концептуального уровня представления данных.

Впервые модель «сущность-связь» была введена Питером Ченом в 1976 году. Её базовыми понятиями являются сущность, атрибут, связь (определения данных понятий были рассмотрены в лекции 2), а также ключ.

Виды сущностей:

1) **независимая сущность**, представляет собой независимые данные, присутствие которых необходимо в проектируемой системе. При этом связи с другими сущностями могут как существовать, так и отсутствовать;

Обозначается:



2) **зависимая сущность**, представляет собой зависимые данные от других сущностей. При этом связи с другими сущностями всегда существуют.

Обозначается:



3) **ассоциированная сущность**, представляет собой данные, которые ассоциируются связями с двумя или более сущностями.

Выделяют следующие **типы связей (отношений)**: неограниченная, или обязательная; ограниченная, или необязательная, существенно-ограниченная.

1. **Неограниченная, или обязательная, связь (отношение)** существует до тех пор, пока существуют относящиеся к ней сущности.

2. **Ограниченная, или необязательная, связь (отношение)** является условной между сущностями.

3. **Существенно-ограниченная связь (отношение)** существует только при взаимосвязанных и взаимозависимых сущностях.

Связи (отношения) подразделяют на следующие виды:

- один к одному – ($1 \leq 1$);
- один ко многим – ($1 \leq \infty$);
- многие к одному – ($\infty \leq 1$);
- многие ко многим – ($\infty \leq \infty$).

Ключ – это элемент данных, который позволяет уникально определять отдельные экземпляры некоторого типа сущности [18].

Различают следующие типы ключей:

1) **первичный ключ** (ПК – primary key) – это один или несколько атрибутов, однозначно определяющих каждый экземпляр сущности;

2) **альтернативный ключ** (АК – alternative key) – каждая сущность должна обладать хотя бы одним первичным ключом, при существовании нескольких возможных первичных ключей один из них обозначается как первичный, а остальные – как альтернативные;

3) **внешний ключ** (FK – foreign key) – между двумя сущностями имеется специфическое отношение связи, атрибуты, входящие в первичный ключ общей или родительской сущности, наследуются в качестве атрибутов сущностью потомком. Данные наследуемые атрибуты и являются внешними ключами сущности потомка.

Пример ERD-модели представлен на рис. 1.4.2.

СУЩНОСТЬ СТУДЕНТ

код студента (ПК)
 фамилия
 имя
 отчество
 факультет
 курс
 номер группы

СУЩНОСТЬ

ЗАЧЕТНАЯ КНИЖКА

номер зачетной книжки (ПК)
 код студента (FK)
 название предмета
 оценка за предмет

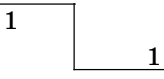


Рис.1.4.2. Пример ERD-модели

STD-модель (диаграмма) – (State Transactions Diagrams) – диаграммы переходов состояний являются средством для моделирования систем реального времени. Основные элементы модели: состояние (объекта или системы) и переход из одного состояния в другое [5].

В настоящее время существует огромное количество различных CASE-средств. Примеры CASE-средств приведены в таблице 1.4.1.