

## Лекции

Лекция 1. Информационные системы с базами данных. . . . .	19
Лекция 2. Предметная область базы данных и ее модели. . . . .	49
Лекция 3. Что такое проектирование баз данных . . . . .	74
Лекция 4. Реляционная модель данных . . . . .	96
Лекция 5. Функциональные зависимости и реляционные базы данных .	111
Лекция 6. Нормальные формы отношений. Создание логической модели реляционной базы данных . . . . .	129
Лекция 7. Методы проектирования логических моделей реляционных баз данных. Декомпозиция и синтез отношений. . . . .	145
Лекция 8. Введение в структурированный язык запросов – SQL . . . .	171
Лекция 9. Создание объектов для хранения данных. Работа с ограничениями . . . . .	200
Лекция 10. Создание физической модели базы данных. Учет влияния транзакций . . . . .	241
Лекция 11. Создание физической модели базы данных. Проектирование производительности. . . . .	265
Лекция 12. Разработка серверного кода . . . . .	294
Лекция 13. Создание физической модели базы данных: Подготовка скрипта для создания базы данных. . . . .	325
Лекция 14. Проектирование модулей приложений. . . . .	353
Лекция 15. Введение в оптимизацию запросов . . . . .	373
Лекция 16. Оптимизация обработки запросов. . . . .	400

## Содержание

Лекция 1. Информационные системы с базами данных. . . . .	19
Информация, данные, информационные системы . . . . .	19
<i>Информация как социальный ресурс . . . . .</i>	<i>19</i>
<i>Информация и данные . . . . .</i>	<i>20</i>
<i>Определение понятия информации . . . . .</i>	<i>21</i>
<i>Информационные системы . . . . .</i>	<i>22</i>
<i>Итерационная процедура построения информационных систем . . . . .</i>	<i>25</i>
Концепция баз данных. . . . .	26
<i>Основные подходы к обработке информации в автоматизированных информационных системах . . . . .</i>	<i>26</i>
<i>Концепция баз данных . . . . .</i>	<i>29</i>
<i>Системы управления базами данных . . . . .</i>	<i>31</i>
Модели данных. . . . .	31
<i>Понятие о модели данных . . . . .</i>	<i>31</i>
<i>Информационная модель данных. . . . .</i>	<i>33</i>
<i>Концепция трех схем. . . . .</i>	<i>33</i>
<i>Основные типы моделей и их эквивалентность . . . . .</i>	<i>34</i>
<i>Общие принципы классификации СУБД . . . . .</i>	<i>37</i>
Обзор основных моделей данных. . . . .	38
<i>Иерархическая модель . . . . .</i>	<i>38</i>
<i>Сетевая модель данных . . . . .</i>	<i>40</i>
Модели вычислений. . . . .	44
Лекция 2. Предметная область базы данных и ее модели. . . . .	49
Понятие предметной области . . . . .	50
Информационная модель предметной области базы данных . . . . .	54
<i>Сущности, атрибуты и идентификаторы (ключи) сущности, домены атрибутов. . . . .</i>	<i>54</i>
<i>Отношения, связи . . . . .</i>	<i>56</i>
<i>Подтипы и супертипы . . . . .</i>	<i>57</i>
Диаграммы «сущность–связь» . . . . .	58
<i>Документирование сущностей и атрибутов . . . . .</i>	<i>58</i>
<i>Документирование доменов . . . . .</i>	<i>59</i>

<i>Документирование отношений (связей)</i> . . . . .	59
<i>Документирование супертипов и подтипов</i> . . . . .	61
Функциональная модель предметной области базы данных . . . . .	62
<i>Понятие функциональной модели предметной области базы данных</i> . . . . .	62
<i>Бизнес-модель процессов (иерархия функций системы)</i> . . . . .	63
<i>Модель потока данных</i> . . . . .	65
<i>Модель жизненного цикла сущности</i> . . . . .	67
<i>Набор спецификаций функций системы (требования), описание функций системы через сущности и атрибуты, бизнес-правила</i> . . . . .	68
<i>Общесистемные требования и решения</i> . . . . .	68
Контроль качества результатов анализа предметной области . . . . .	70
Лекция 3. Что такое проектирование баз данных . . . . .	74
Введение . . . . .	75
Что такое проектирование базы данных . . . . .	75
Типовая бизнес-модель процесса проектирования базы данных . . . . .	78
Бизнес-модель процесса проектирования базы данных: сбор и анализ входных данных . . . . .	82
Бизнес-модель процесса проектирования реляционной базы данных: создание логической модели базы данных . . . . .	83
Бизнес-модель этапа проектирования – создание физической модели реляционной базы данных . . . . .	87
Бизнес-модель этапа проектирования – создание физической модели реляционной базы данных: учет влияния транзакций . . . . .	90
Краткое рассмотрение задач создания серверного кода и подготовки скрипта . . . . .	93
Лекция 4. Реляционная модель данных . . . . .	96
Понятие отношения . . . . .	96
Формы представления отношений . . . . .	99
Реляционные операции . . . . .	102
<i>Объединение отношений</i> . . . . .	103
<i>Пересечение отношений</i> . . . . .	103
<i>Разность отношений</i> . . . . .	104
<i>Декартово произведение отношений</i> . . . . .	105
<i>Проекция отношения</i> . . . . .	106

<i>Деление отношений</i> . . . . .	107
<i>Выбор из отношения</i> . . . . .	108
<i>Соединение отношений</i> . . . . .	109
Лекция 5. Функциональные зависимости и реляционные базы данных .	111
Понятие функциональной зависимости в данных . . . . .	111
Основные классы функциональных зависимостей. . . . .	115
Аксиомы вывода функциональных зависимостей . . . . .	118
Минимальные покрытия множеств функциональных зависимостей. . . . .	124
Лекция 6. Нормальные формы отношений. Создание логической модели реляционной базы данных . . . . .	129
Понятие о логической модели реляционной базы данных . . . . .	129
Нормализация отношений . . . . .	130
Первая нормальная форма. . . . .	134
Вторая нормальная форма. . . . .	136
Третья нормальная форма . . . . .	137
Нормальная форма Бойса-Кодда . . . . .	138
Четвертая нормальная форма . . . . .	140
Пятая нормальная форма. . . . .	142
Лекция 7. Методы проектирования логических моделей реляционных баз данных. Декомпозиция и синтез отношений. . . . .	145
Универсальное отношение. . . . .	145
Декомпозиция схем отношений, свойства соединения без потерь и сохранения ФЗ. . . . .	147
Методы проектирования на основе декомпозиции отношений .	151
<i>Понятие о методах декомпозиции отношений</i> . . . . .	151
<i>Алгоритм метода декомпозиции отношений</i> . . . . .	152
Методы проектирования на основе синтеза отношений . . . . .	153
<i>Некоторые проблемы метода декомпозиции</i> . . . . .	153
<i>Понятие о методах синтеза отношений</i> . . . . .	154
<i>Алгоритм метода синтеза отношений.</i> . . . . .	155
Создание логической модели реляционной базы данных методом декомпозиции: преобразование ER-диаграмм в отношения базы данных . . . . .	160
Пример преобразования ER-диаграмм в отношения базы данных .	167

Лекция 8. Введение в структурированный язык запросов – SQL . . . . .	171
Введение . . . . .	171
SQL – язык манипулирования данными в реляционной базе данных . . . . .	172
<i>SQL и его история</i> . . . . .	172
<i>Описание основных операторов SQL</i> . . . . .	173
<i>Встроенные функции SQL и их использование в запросах</i> . . . . .	176
Объекты реляционной базы данных. . . . .	184
<i>Иерархия объектов реляционной базы данных</i> . . . . .	184
<i>Основные объекты реляционной базы данных</i> . . . . .	186
<i>Правила определения имен</i> . . . . .	189
Домены и допустимые типы данных . . . . .	191
<i>Допустимые типы данных</i> . . . . .	191
<i>Константы, выражения, системные переменные</i> . . . . .	196
Приложение 1 к лекции 8 . . . . .	198
Лекция 9. Создание объектов для хранения данных.	
Работа с ограничениями . . . . .	200
Создание начальной внутренней схемы реляционной базы данных. . . . .	201
Создание таблиц . . . . .	201
<i>Определение базовых таблиц</i> . . . . .	202
<i>Добавление колонок в таблицы</i> . . . . .	203
<i>Определение типов данных для колонок</i> . . . . .	205
<i>Назначение первичных ключей таблицам</i> . . . . .	207
<i>Задание ограничений NOT NULL на значения колонок</i> . . . . .	210
Создание связывающих таблиц для ситуации разрешения отношений «многие-ко-многим» в логической модели базы данных . . . . .	212
Работа с ограничениями. Добавление ссылочной целостности . . . . .	217
<i>Ограничения и их использование в реляционной базе данных</i> . . . . .	217
<i>Добавление CHECK-ограничения в спецификацию колонки</i> . . . . .	218
<i>Использование опции DEFAULT</i> . . . . .	218
<i>Добавление NOT NULL ограничения в спецификацию колонки</i> . . . . .	218
<i>Добавление ограничения первичного ключа и внешнего ключа</i> . . . . .	220
<i>Добавление ограничения UNIQUE в спецификацию колонки</i> . . . . .	220
<i>Добавление, удаление и блокирование ограничений</i> . . . . .	220
Добавление ограничений ссылочной целостности . . . . .	222

<i>Ссылочная целостность</i> . . . . .	222
<i>Первичные и внешние ключи</i> . . . . .	223
<i>Отношение «родитель–потомок» между таблицами</i> . . . . .	225
<i>Создание таблиц с ограничениями ссылочной целостности</i> . . . . .	226
<i>Особенности манипулирования данными при ограничениях     ссылочной целостности</i> . . . . .	227
<i>Циклы зависимых таблиц</i> . . . . .	228
Создание представлений и синонимов. Понятие о внешней схеме . . . . .	230
<i>Представления или виртуальные таблицы</i> . . . . .	230
<i>Синонимы</i> . . . . .	236
<i>Понятие внешней схемы</i> . . . . .	237
<i>Создание первоначальной внешней схемы</i> . . . . .	238
Лекция 10. Создание физической модели базы данных.	
Учет влияния транзакций . . . . .	241
Введение . . . . .	241
Определение транзакций базы данных . . . . .	243
<i>Понимание типа приложений базы данных</i> . . . . .	244
<i>Спецификация транзакций</i> . . . . .	249
Денормализация . . . . .	249
<i>Понятие о денормализации</i> . . . . .	250
<i>Нисходящая денормализация</i> . . . . .	252
<i>Восходящая денормализация</i> . . . . .	253
<i>Внутритабличная денормализация</i> . . . . .	253
<i>Денормализация методом «разделяй и властвуй»</i> . . . . .	253
<i>Денормализация методом слияния таблиц</i> . . . . .	255
Методы реализации денормализации: Разбиение таблиц базы данных. . . . .	255
<i>Вертикальное разбиение длинных строк</i> . . . . .	256
<i>Длинные строки в таблицах хэширования</i> . . . . .	258
<i>Горизонтальное разбиение таблиц</i> . . . . .	260
<i>Разбиение таблиц и ссылочная целостность</i> . . . . .	262
Лекция 11. Создание физической модели базы данных.	
Проектирование производительности. . . . .	265
Введение . . . . .	266
Повышение производительности запросов: Индексы . . . . .	266
Индексирование . . . . .	266

<i>Индекс со структурой B-Tree</i> . . . . .	268
<i>Исключительно индексные таблицы</i> . . . . .	271
<i>Параметры индексирования</i> . . . . .	272
<i>О некоторых параметрах проектирования индексов</i> . . . . .	274
Повышение производительности запросов: Секционирование . . . . .	277
<i>Секционирование</i> . . . . .	277
<i>Секционирование по диапазону</i> . . . . .	278
<i>Хэш-секционирование</i> . . . . .	279
<i>Составное секционирование</i> . . . . .	280
<i>Секционирование индексов</i> . . . . .	281
<i>Секционирование представлений</i> . . . . .	284
Повышение производительности запросов: Кластеры . . . . .	286
Лекция 12. Разработка серверного кода . . . . .	294
Введение . . . . .	294
PL/SQL – процедурное расширение языка SQL . . . . .	296
<i>Структура программы на PL/SQL</i> . . . . .	296
<i>Константы, переменные и типы в PL/SQL</i> . . . . .	297
<i>Операторы управления выполнением программы PL/SQL</i> . . . . .	297
<i>Курсоры PL/SQL</i> . . . . .	300
<i>Обработка исключительных ситуаций в PL/SQL</i> . . . . .	301
Создание хранимых процедур и функций . . . . .	304
<i>Определение хранимых процедур и функций в PL/SQL</i> . . . . .	304
<i>Создание хранимых процедур и функций</i> . . . . .	308
<i>Особенности использования процедур и функций         в СУБД Oracle</i> . . . . .	310
Создание пакетов PL/SQL . . . . .	313
Создание триггеров PL/SQL . . . . .	317
Лекция 13. Создание физической модели базы данных:	
Подготовка скрипта для создания базы данных . . . . .	325
Введение . . . . .	325
Средства разграничения доступа в СУБД Oracle . . . . .	326
Создание пользователей и назначение им привилегий . . . . .	328
<i>Создание пользователей</i> . . . . .	328
<i>Назначение привилегий</i> . . . . .	329
Оценка размера базы данных . . . . .	333
<i>Формулы для оценки размера БД</i> . . . . .	333

<i>Пример расчета размера базы данных</i> . . . . .	337
Создание табличных пространств . . . . .	339
Проверка физической модели реляционной базы данных . . . . .	342
Подготовка скрипта создания физической базы данных . . . . .	349
Лекция 14. Проектирование модулей приложений . . . . .	353
Введение . . . . .	353
Анализ функциональной модели предметной области базы данных . . . . .	354
Определение функций . . . . .	356
Отображение функций в модули . . . . .	357
Системные модули . . . . .	362
Размещение логики обработки . . . . .	362
Общие принципы разработки спецификаций модулей . . . . .	364
Проектирование процесса тестирования модулей приложений . . . . .	368
Лекция 15. Введение в оптимизацию запросов . . . . .	373
Языки обработки данных и задача оптимизации обработки данных . . . . .	374
Процедурные языки обработки данных . . . . .	375
Декларативные языки обработки данных . . . . .	376
Оптимизация запросов . . . . .	377
<i>Синтаксическая оптимизация</i> . . . . .	379
<i>Оптимизация, основанная на правилах</i> . . . . .	381
<i>Оптимизация, основанная на вычислении стоимости</i> . . . . .	382
Последовательность шагов оптимизации запросов . . . . .	382
Обзор оптимизатора запросов . . . . .	382
Реляционные операции . . . . .	383
<i>Теоретико-множественные операции</i> . . . . .	384
<i>Специальные реляционные операторы</i> . . . . .	385
Физические операции . . . . .	390
<i>Сортировка и агрегация</i> . . . . .	391
<i>Операции доступа к диску</i> . . . . .	391
<i>Операции соединения</i> . . . . .	393
<i>Другие физические встроенные операции</i> . . . . .	396
Структура плана запроса . . . . .	397
<i>Дерево запроса</i> . . . . .	398
<i>Построение дерева запроса</i> . . . . .	398

<i>Преобразование логики предиката</i> . . . . .	399
Лекция 16. Оптимизация обработки запросов . . . . .	400
Реализация оптимизатора SQLBase . . . . .	400
<i>Статистика базы данных</i> . . . . .	400
<i>Статистика таблиц</i> . . . . .	401
<i>Статистика индексов</i> . . . . .	403
Фактор селективности . . . . .	404
<i>Фактор селективности в случае одного предиката</i> . . . . .	405
<i>Фактор селективности в случае нескольких предикатов</i> . . . . .	407
Использование оптимизатора для оптимизации выполнения запросов . . . . .	408
Анализ запросов с целью повышения скорости их выполнения . . . . .	418
Литература . . . . .	418

## Лекция 1. Информационные системы с базами данных

В лекции вводятся и объясняются основные понятия и термины, используемые при разработке и создании информационных систем с базами данных. Особое внимание уделено идеям и принципам, составляющим методологическую основу построения баз данных.

**Цель лекции:** изучив материал данной лекции, вы будете знать:

- что такое информация, данные, информационные системы;
- концепцию баз данных;
- основные модели данных;
- основные модели вычислений

и сможете идентифицировать архитектуру баз данных в информационных системах.

**Ключевые слова:** информация, данные, информационная система, аппаратное (hardware), программное (software), коммуникационное (netware), промежуточного слоя (middleware), лингвистическое и организационно-технологическое обеспечение, базы данных, система управления базой данных, модель данных, сетевая, иерархическая и реляционная модели, транзакция, модели вычислений, централизованные и распределенные вычисления, модель вычислений «файл-сервер», модель вычислений «клиент-сервер».

**Литература:** [1], [2], [10], [12], [21], [29], [34], [36], [40], [41], [47].

## Информация, данные, информационные системы

### Информация как социальный ресурс

За три последних десятилетия стало общепризнанным, что информация является не менее важным ресурсом человеческого общества, чем сырье, энергия и пища. Можно утверждать, что практически в любом виде человеческой деятельности *требуется удовлетворение информационных потребностей* в той или иной степени. Так, например, собираясь на улицу, мы всегда хотим получить информацию о погоде. Большинство из нас в том или ином виде ежедневно получают различную информацию из газет, радио, телепередач, Интернета. Не говоря уже об источниках профессиональной информации.

Развитие систем связи и коммуникаций привело к усложнению и дифференциации информационных процессов в человеческом обществе. Способность накапливать информацию и обеспечивать эффективный доступ к ней становится определяющим фактором не только развития человеческого общества, но и поддержания его жизнеспособности. Быстрый рост объемов информации, закрепленной на внешних по отношению к человеку носителях, привел к появлению новых общественных институтов (библиотеки, архивы, пресса, вычислительные центры и т. д.) и специальных систем (службы научно-технической информации, справочные службы, глобальные информационные компьютерные сети).

Развитие средств вычислительной техники и информационных технологий открыло новые возможности и способы хранения, представления и поиска информации, в частности, создание вычислительных систем, «доступных по требованию» — т.е. вычислительные ресурсы становятся таким же доступным ресурсом для потребления человеком, как электроэнергия, природный газ, вода.

Таким образом, резко возрастают требования к качеству и надежности проектирования систем для работы с информацией, представляемой в электронном виде.

## Информация и данные

Прежде чем перейти к обсуждению понятия информационной системы (ИС), попытаемся выяснить, что же понимается под словом **информация**. Ответить на этот вопрос и просто, и сложно: слово «информация» связано с широким кругом понятий, в том числе и определенных строго математически (информация по Шеннону, например).

Содержательная же сторона понятия «информация» очень многогранна и не имеет четких семантических границ. Однако всегда можно сказать, *что можно с ней делать*. Именно ответ на этот вопрос чаще всего и интересует как системных аналитиков и разработчиков (ИС), так и пользователей информации (ее основных потребителей).

С точки зрения как пользователей, так и разработчиков ИС, у информации есть одно важное свойство — она является единицей данных, подлежащих обработке. Обычно информация поступает потребителю именно в виде данных: таблиц, графиков, рисунков, фильмов, устных сообщений, которые фиксируют в себе информацию определенной структуры и типа. Таким образом, *данные выступают как способ представления информации в определенной, фиксированной форме, пригодной для обработки, хранения и передачи*. Хотя очень часто термины «информация» и «данные» выступают как синонимы (особенно в среде разработчиков ИС), следует

помнить об этом их существенном отличии. Именно в данных информация обретает интерпретацию в конкретной ИС.

При упоминании о «форме» представления информации следует сказать и еще об одном, «человеческом» свойстве информации – ее восприятию различными категориями людей. Данные могут быть сгруппированы совместно в документ. Документ может иметь или не иметь определенную внутреннюю структуру. Данные могут быть отображены на экране дисплея компьютера. Документы могут иметь аудио- или видеоформу. Разрабатывая ИС, никогда не следует забывать, для кого они (системы) создаются и кто будет их использовать (воспринимать информацию в них). Форма представления информации в ИС определяет не только ее «дружелюбие», но также и категории пользователей. ИС создаются для конкретных групп пользователей, т.е. они, как правило, проблемно-ориентированны.

### **Определение понятия информации**

Теперь мы готовы дать следующее определение информации в применении к ИС.

**Определение 1.** *Информация есть данные, которым придается некоторый смысл (интерпретация) в конкретной ситуации в рамках некоторой системы понятий. Информация представляется посредством кодирования данных и извлекается путем их декодирования и интерпретации.*

В этом определении фиксируется три основных преобразования **информации** и **данных** в процессе их обработки в ИС: *информация-данные, данные-данные, данные-информация.*

Рассмотрим пример с классической ИС – библиотекой. Книга поступает в библиотеку. На нее заводятся библиографические карточки (выполняется преобразование *информация-данные*). Библиографические карточки размещаются в каталогах в соответствии с внутренними библиотечными процедурами систематизации (выполняется преобразование «данные-данные»). Читатели работают с каталогами библиографических карточек в поисках нужных им книг (выполняется преобразование *данные-информация*). Аналогичным образом можно рассмотреть процесс продажи товара через склад и многие другие сферы человеческой деятельности.

Заметим, что вопрос о преобразовании *информация-информация*, которое также имеет прямое отношение к ИС, не фигурирует явно в данном рассмотрении. Это преобразование имеет прямое отношение к производству новых знаний. Производство новых знаний относится к разработке

систем искусственного интеллекта и не затрагивается непосредственно в данных лекциях.

На рис. 1.1 представлены две стороны определения понятия **информации**: функциональная и представительная. Первая в общих чертах определяет круг действий над информацией, а вторая – результат выполнения этих действий.



**Рис. 1.1.** Содержание термина «информация»

При разработке ИС важно отличать собственно генерацию информации и поддержку ее актуальности (соответствия текущему моменту) от процедур ее оформления для потребления пользователем.

## Информационные системы

Основной целью создания ИС является *удовлетворение информационных потребностей пользователей путем предоставления необходимой им информации на основе хранимых данных*. Потребность в информации как

таковой не исчерпывает понятия информационных потребностей. Обычно в понятие информационных потребностей включают определенные требования к качеству информационного обслуживания и поведению системы в целом (производительность, актуальность и надежность данных, ориентация на пользователя и ряд других, о чем мы поговорим позже).

**Определение 2.** *Под информационной системой понимается организационная совокупность технических и обеспечивающих средств, технологических процессов и кадров, реализующих функции сбора, обработки, хранения, поиска, выдачи и передачи информации.*

Необходимость повышения производительности труда в сфере информационной деятельности приводит к тому, что в качестве внешних средств хранения и быстрого доступа к информации чаще всего используются средства вычислительной техники (цифровой и аналоговой) на основе компьютеров. Современные ИС — сложные комплексы аппаратных и программных средств, технологии и персонала, которые еще называют *автоматизированными информационными системами*. Структурно ИС включают в себя **аппаратное (hardware), программное (software), коммуникационное (netware), промежуточного слоя (middleware), лингвистическое и организационно-технологическое обеспечение**.

**Аппаратное обеспечение** ИС включает в себя широкий набор средств вычислительной техники, средства передачи данных, а также целый ряд специальных технических устройств (устройства графического отображения информации, аудио- и видеоустройства, средства речевого ввода и т.д.). Аппаратное обеспечение является основой любой ИС.

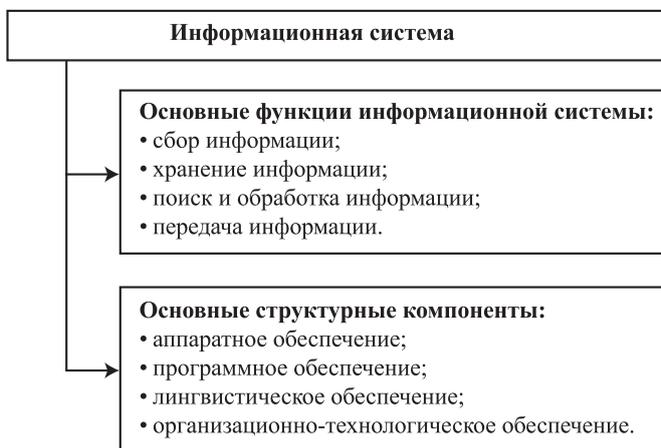
**Коммуникационное (сетевое) обеспечение** включает в себя комплекс аппаратных сетевых коммуникаций и программных средств поддержки коммуникаций в ИС. Оно имеет существенное значение при создании распределенных ИС и ИС на основе Интернета. При создании распределенных ИС огромную роль также играет **программное обеспечение промежуточного слоя**, состоящее из набора программных средств (служб и сервисов), которые управляют взаимодействием распределенных объектов в системе.

**Программное обеспечение** ИС обеспечивает реализацию функций ввода данных, их размещения на машиночитаемых носителях, модификации данных, доступ к данным, поддержку функционирования оборудования. Программное обеспечение можно разделить на системное (которое венчает процесс выбора аппаратно-программного решения, или платформы, как говорят в настоящее время) и пользовательское (которое применяется для решения задач удовлетворения потребностей пользователя в компьютерной среде, а именно, реализует бизнес-логику).

**Лингвистическое обеспечение** ИС предназначено для решения задач формализации смыслового содержания полнотекстовой и специальной информации для создания *поискового образа данных* (профиля). В классическом смысле обычно оно включает процедуры индексирования текстов, их классификацию и тематическую рубрикацию. Зачастую ИС, содержащие сложно-структурированную информацию, включают в себя тезаурусы терминов и понятий (средства поддержки метаданных). Сюда можно отнести и создание процессоров специализированных формальных языков конечных пользователей, например языков для манипулирования бухгалтерской информацией и т.д. Чаще всего работам по разработке лингвистического обеспечения не придается должного значения. Подобные упущения чаще всего ведут к неприятию пользователями самой системы и, как следствие, к ее закономерной гибели. Это относится в первую очередь к узко специализированным ИС.

По мере возрастания сложности и масштабов ИС важную роль начинает играть **организационно-технологическое обеспечение**, которое соединяет разнородные компоненты (аппаратуру, программы и персонал) в единую систему и обеспечивает процедуры ее управления и функционирования. Недооценка этой составляющей ИС чаще всего приводит к срыву сроков внедрения системы и вывода ее на производственные мощности.

На рис. 1.2 просуммированы в общих чертах функции ИС через ее основные структурные компоненты.



**Рис. 1.2.** Определение информационной системы

## Итерационная процедура построения информационных систем

В чем состоит преимущество ориентации на автоматизацию основных бизнес-процессов при автоматизации организации или предприятия? Традиционно и повсеместно используемым подходом (особенно на начальных этапах развития информационной инфраструктуры организации) является применение так называемого позадачного метода решения задач автоматизации, направленного на решение достаточно простых и понятных руководству задач.

Например, учет заказов, выписка счетов, подготовка документов. Конъюнктурное преимущество такого метода очевидно: достаточно быстро может быть получен результат, существование модной ныне ИТ-службы оправдано, внутренние инвестиции быстро вернулись. В принципе, с точки зрения системного анализа это является порочной практикой. Однако он существует, поскольку позволяет, с одной стороны, вроде бы не отставать от жизни (наличие ИС в организации зачастую является одним из определяющих факторов ее конкурентоспособности), а с другой — экономить денежные средства на автоматизации. Вышеуказанный подход позволяет использовать служащих с невысокой квалификацией. Рано или поздно это становится тормозом в развитии информационной инфраструктуры организации.

Низкая отдача уже существующей ИС организации на текущем этапе ее эксплуатации также становится тормозящим фактором. Изменение направлений бизнеса организации и ряд других факторов приводят к вопросу пересмотра отношения к ИС в организации, т.е. к извечному вопросу — переделать или начать с начала. Начать сначала всегда выгодней. Можно применить уже хорошо отработанные в информатике методики проектирования «сверху-вниз» или «снизу-вверх». Однако рано или поздно опять встанет вопрос о соответствии требованиям сегодняшнего дня.

Даже в тех случаях реализации ИС, которые одобряются системным анализом, *не удастся избежать переделки ИС, т.к. она, как органическая часть производственного процесса, должна следовать и отвечать стратегическому генеральному бизнес-плану развития организации.* Такой план всегда должен быть, если организация собирается долго жить в своем секторе рынка.

Разработчики ИС фактически всегда находятся в методике «из-середины» (middle of design). Есть некоторая основа (уже созданная или создаваемая), и вокруг нее следует развиваться в различных направлениях, не сильно ломая сложившиеся традиции. Таким образом, постулируется итерационный подход в разработке и создании информационных систем. И, как следует из вышесказанного, он определяется не желанием теоретиков ИС, а жизненной необходимостью.

Основным подходом в таких переделках (так же, как и при создании) ИС является концепция реинжиниринга, суть которого сводится к постоянному моделированию информационных процессов и данных организации и их отображения в существующей системе.

Что является основной особенностью технологии реализации итерационного подхода разработки и сопровождения ИС? Основная особенность реализации концепции разработки ИС, ориентированной на интегрированные процессы, — это наличие или отсутствие сборочного конвейера, поскольку необходимо собирать воедино многие технологические процессы обработки информации. При объединении технологических процессов обработки информации увеличивается скорость прохождения информации в системе, принятие решений на основе информационных потоков становится частью процесса обработки информации (и более состоятельной), уменьшается иерархия управленческих структур.

Следует также иметь в виду, что наличие в организации корпоративной ИС зачастую меняет представление пользователей о том, как информация должна циркулировать в организации, что является еще одним важным стимулом модернизации ИС. ИС — составляющая и несущая часть пирамиды обработки информации. Поэтому в процессе разработки и реализации ИС приходится перестраивать как бизнес-процессы, так и бизнес-правила и их взаимодействие, что и составляет основу реинжиниринга.

Для того чтобы ИС жила долго и ее эксплуатация приносила ощутимую выгоду, необходимо тщательно проектировать и ее архитектуру, и ее составные компоненты, в частности базы данных, о которых пойдет речь ниже.

## Концепция баз данных

Подавляющее большинство компьютеров, которые используются для аппаратного обеспечения создателями ИС, являются компьютерами фон Неймана. Основная идея, положенная в основу создания компьютера фон Неймана, состоит в том, что компьютер представляет собой вычислительную машину с загружаемым в его память кодом — программами и данными. В процессе своей работы такая машина интерпретирует код и различает программы (исполняемый код) и данные (неисполняемый код).

### **Основные подходы к обработке информации в автоматизированных информационных системах**

Одним из главных вопросов разработки программного обеспечения ИС (и программирования как самостоятельной дисциплины) является вопрос о *соотнесении программ и данных*, ибо решение этого вопроса, в конечном счете, определяет выбор алгоритмов обработки информации, ап-

[ . . . ]

пользуется язык описания данных CODASYL. Описание схемы БД в CODASYL состоит из четырех статей:

1. статья схемы: SCHEMA NAME IS Имя\_схемы;
2. статья областей: AREA NAME IS Имя\_области (файла);
3. статья записи: RECORD NAME IS Имя\_записи – способ выборки;
4. статья выбора: SET NAME IS Имя\_набора; способ включения экземпляров записей (устанавливает групповые отношения в БД).

#### Описание записи в сетевой модели

```

RECORD NAME IS Служащий;
Location mode IS Proc USING Табельный_Номер
DURLISATES ARE NOT ALLOWED;
WITHIN область-A AREA.
    02 Табельный_номер; PICTURE IS 9(5).
    02 ФИО; PICTURE IS A(40).
    02 Дата_Рождения.
        03 число; PICTURE IS 99.
        03 месяц; PICTURE IS A(3).
        03 год; PICTURE IS 99.
    02 Образование; OCCURS 3 TIMES.
        03 Учебное_заведение; PICTURE IS X(15).
        03 Год_окончания; PICTURE IS 9(4).
        03 Специальность; PICTURE IS 9(8).
    02 Должность.
        03 Наименование; PICTURE IS X(15).
        03 Оклад; PICTURE IS 9(9).
        03 Приказ; PICTURE IS X(6).
        03 Дата; PICTURE IS 99X99X9999.
  
```

**Рис. 1.9.** Описание записи в сетевой модели данных

Элементы данных сетевой модели допускают обработку следующими операциями, множество которых составляет язык манипулирования данными:

- **ЗАПОМНИТЬ** – заносит экземпляр записи в БД и включает его в существующее отношение;
- **ПРИСОЕДИНИТЬ** – связывает существующие записи в групповое отношение и определяет подчинение записей (родитель-потомок);
- **ПЕРЕКЛЮЧИТЬ** – связывает экземпляр подчиненной записи с другим экземпляром записи-родителя;
- **МОДИФИЦИРОВАТЬ** – изменяет значение полей в существующих записях БД, перед выполнением этой операции запись должна быть извлечена из БД;
- **НАЙТИ** – находит записи из БД согласно критерию поиска;

- УДАЛИТЬ – удаляет из БД ненужную запись;
- ОТДЕЛИТЬ – разрывает существующую связь между записями в групповом отношении;
- ПОЛУЧИТЬ – извлекает записи из БД.

В модели CODASYL существует набор дополнительных операций по обслуживанию БД, который здесь не рассматривается.

Очень часто к недостаткам сетевого подхода в БД относят как сложность самой модели данных, так и сложность освоения средств манипулирования данными в ней. Практически, при анализе ПО БД и программировании особенно тщательно приходится отлеживать цепочки связанных групповыми отношениями данных при операциях вставки, обновления и удаления. Однако действительный источник сложности сетевой модели данных состоит *в диапазоне предоставляемых моделью конструкций для представления информации и набора операции для манипулирования этими конструкциями.*

## Модели вычислений

Взгляд на использование компьютеров меняется в процессе их применения в различных сферах человеческого труда: большие вычислительные центры с мощными компьютерами, средние по мощности ЭВМ для автоматизации технологических процессов, персональные компьютеры, компьютеры, объединенные сетью коммуникаций. Неизменным остается требование пользователей к вычислительным ресурсам *для удовлетворения потребностей в информации* – время процессора (быстродействие), оперативная память, дисковое пространство и т.п. Проблема совместного использования ресурсов является одной из ключевых проблем решения любых прикладных задач на ЭВМ, в том числе и создания ИС. Решение этой проблемы приводит к разработке новых компьютерных технологий, которые являются сложным синтезом изменений в аппаратном и программном обеспечении. Основой таких модификаций как аппаратного, так и программного обеспечения являются **модели вычислений.**

Что принято понимать под моделью вычислений? Обычно *под моделью вычислений подразумевают совокупность аппаратно-программных средств, схему их взаимодействия между собой и пользователями*, т.е. постулируется ответ на вопросы, каким образом и какие вычислительные ресурсы используются в процессе выполнения вычислений. Поскольку понятие модели вычислений связано как и с аппаратным, так и с программным обеспечением, то нередко в качестве синонима слова *модель* используется слово архитектура. За всю историю развития вычислительной техники было предложено не так уж много моделей вычислений.

Централизованные вычисления:

- модель вычислений с использованием централизованной хост-ЭВМ;
- модель с автономными персональными вычислениями;

Распределенные вычисления:

- модель вычислений «файл-сервер»;
- модель вычислений «клиент-сервер»;
- модель «вычисление по требованию».

Исторически одной из первых моделей вычислений является модель с использованием централизованной хост-ЭВМ. В такой схеме вычислений пользователь получает доступ к вычислительным ресурсам ЭВМ через сеть неинтеллектуальных терминалов (т.е. терминалов, не обладающих никакими вычислительными возможностями). Центральный компьютер полностью отвечает за взаимодействие с пользователем и управление данными в многопользовательской среде.

Преимуществом такой модели вычислений является их централизация. Централизованные системы позволяют совместно использовать вычислительные ресурсы (диски, принтеры, оперативную память) с высокой эффективностью, а также обеспечивать высокую надежность и актуальность хранимых данных.

Самым большим недостатком такой схемы вычислений является линейная зависимость вычислительной мощности центральной ЭВМ от числа пользователей и, как следствие, высокая стоимость аппаратуры и программного обеспечения. Несмотря на устойчивую тенденцию снижения стоимости оборудования, такие системы по-прежнему остаются одними из дорогостоящих (отношение «цена/производительность» остается достаточно высокой).

В 80-е годы прошлого века появились персональные компьютеры и рабочие станции. Независимые друг от друга, предоставляющие вычислительные возможности, которые сопоставимы с большими ЭВМ, доступные по цене широкому кругу потребителей (отношение «цена/производительность» в данном случае гораздо ниже, чем при использовании больших ЭВМ). Персональные компьютеры положили конец централизованному подходу в обработке данных и обозначили переход к **распределенным вычислениям**.

Преимуществом такой модели вычислений является их автономность в использовании вычислительных ресурсов, т.е. централизованное использование компьютера, но на рабочем месте и независимо от других таких же компьютеров. В данном случае можно подобрать персональный компьютер адекватно решаемому кругу задач.

Однако у независимых персональных вычислений есть и свои проблемы. Эти проблемы порождают распределенность данных (невозможность

совместной работы с данными различных пользователей) по персональным компьютерам в случае, когда эти данные должны использоваться совместно в рамках одной организации. При этом выигрыш в отношении «цена/производительность» компенсируется потерями в производительности труда коллективов, работающих с распределенными таким образом данными.

Проблемы совместного использования данных, расположенных на персональных компьютерах, привели к разработке *концепции локальной вычислительной сети*, которая восстанавливает преимущества коллективных вычислений и сохраняет простоту использования персональных компьютеров. Наличие вычислительной сети компьютеров характерно для всех моделей распределенных вычислений.

**Модель вычислений «файл-сервер»** (или архитектура «файл-сервер») основывается на понятии сервера. Термин *сервер* имеет двойственный смысл. С одной стороны, сервер есть узел вычислительной сети (компьютер с сети), предназначенный для предоставления совместно используемых ресурсов и услуг, а с другой – программный компонент, предоставляющий общий функциональный сервис другим программным компонентам вычислительной сети.

Файловый сервер является обычно центральным узлом сети, на котором хранятся файлы коллективного пользования и который является также концентратором совместно используемых периферийных устройств (например, принтера или дискового накопителя большой емкости). Файловый сервер не принимает участия в обработке приложения. Он выполняет сетевой транспорт совместно используемых данных (часто пересылая файл целиком конечному пользователю).

Преимуществом такой модели является, несомненно, корпоративное использование территориально распределенных вычислительных ресурсов, имеющее одним из своих следствий создание глобальных вычислительных систем и новых технологий обмена информацией.

Однако у такой модели есть два крупных недостатка при разработке многопользовательских приложений. Интенсивный обмен данными (рост трафика сети) приводит к быстрому достижению ее пропускной способности и тем самым к снижению (из-за увеличения времени реакции приложения за счет времени ожидания) производительности многопользовательской системы.

Другая проблема – это обеспечение согласованности данных, т.е. одновременного разделения доступа к одним и тем же данным группой пользователей. Обычно файл блокируется для других пользователей, когда его начинает обрабатывать приложение. В случае, когда часть файла реплицируется на конечный узел для обработки, снижается актуализация данных, что может быть неприемлемо для систем оперативной обработки информации.

**Модель вычислений «клиент-сервер»** явилась следующим шагом в развитии распределенных вычислений, объединив в себе преимущества коллективных вычислений в сети компьютеров с доступом к совместно используемым данным и высокие характеристики производительности вычислений с центральной ЭВМ. Основными понятиями данной модели являются *сервер баз данных, клиентское приложение и сеть*.

Основное назначение *сервера баз данных* — оптимальное управление разделяемыми ресурсами на уровне данных для множества клиентов. На этом уровне достаточно эффективно решаются задачи обеспечения согласованности данных, их актуальности, защиты и целостности.

*Клиентское приложение* является частью системы, которая обеспечивает интерфейс приложения с серверов баз данных. Логика приложения может быть полностью реализована на клиентской части системы, а обработку данных забирает на себя сервер баз данных.

*Сеть и коммуникационное программное обеспечение* являются средствами передачи данных. Реализация этой компоненты модели обеспечивает прозрачность сервера баз данных по отношению к клиенту.

## Архитектура клиент-сервер

### Преимущества

- Разделение логики приложения (клиент) и обработки данных (сервер).
- Гибкость и адаптируемость архитектуры к изменениям в аппаратном и программном обеспечении.
- Масштабируемость архитектуры по отношению к изменению числа пользователей совместно используемых данных.
- Экономия средств при необходимом уровне отношения цена-производительность.

### Недостатки

- Экономия средств за счет стоимости аппаратуры достигается не всегда из-за высокой стоимости разрабатываемого программного обеспечения и стоимости переподготовки кадров.
- Можно не достигнуть ожидаемого увеличения производительности системы (и даже получить ее снижение) из-за несогласованного использования аппаратных и программных средств, созданных различными производителями.
- Существуют задачи, для решения которых данная модель может просто не подойти (например, расчет многомерных газодинамических течений при движении ракеты в атмосфере планеты).

**Рис. 1.10.** Преимущества и недостатки модели вычислений «клиент-сервер»

Несмотря на то, что модель вычислений «клиент-сервер» является высокопроизводительной распределенной моделью вычислений, она, помимо очевидных преимуществ, имеет присущие ей недостатки (рис. 1.10). Кроме того, другие модели вычислений также продолжают развиваться, обеспечивая приемлемые значения отношения «цена-производительность».

**Модель «вычисления по требованию»** или GRID является в настоящее время одной из перспективных распределенных моделей вычислений. Суть ее состоит в использовании вычислительных ресурсов, расположенных в локальной или глобальной вычислительной сети, аналогично тому, как мы в быту используем электричество, совершенно не отдавая себе отчета в том, с какой электростанции оно поступает к нам в дом.

В этой модели вычислений заявленные в сети GRID вычислительные ресурсы (компьютеры или кластеры ЭВМ) предоставляют свои свободные вычислительные ресурсы согласно правилам обслуживания заданий в очереди. Таким образом, находясь в России, вы можете запустить свою задачу на компьютере в Австралии, совершенно об этом не зная.

В этой лекции мы рассмотрели ряд основных понятий и терминов, которые потребуются проектировщику реляционных баз данных в процессе решения им своих профессиональных задач. В последующих лекциях мы последовательно и детально рассмотрим основные профессиональные задачи проектировщика реляционных баз данных.

[ . . . ]