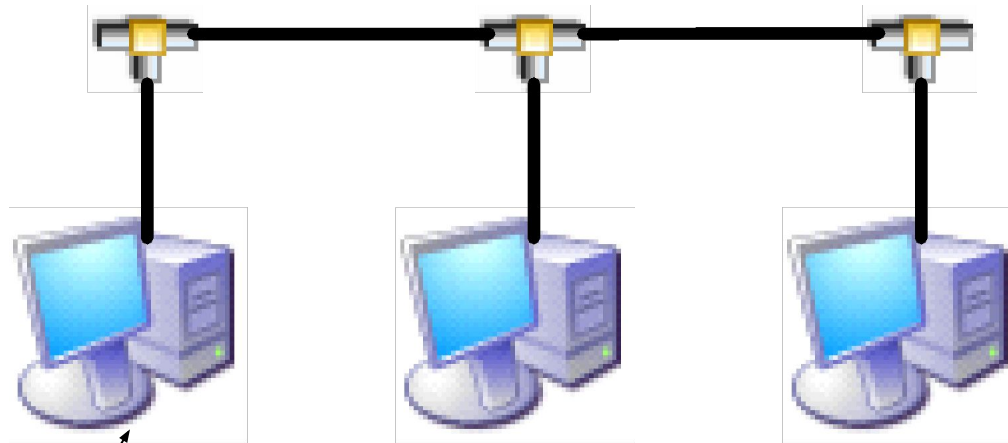


Логическая и физическая структура БД

Технология физического хранения БД

1 вариант

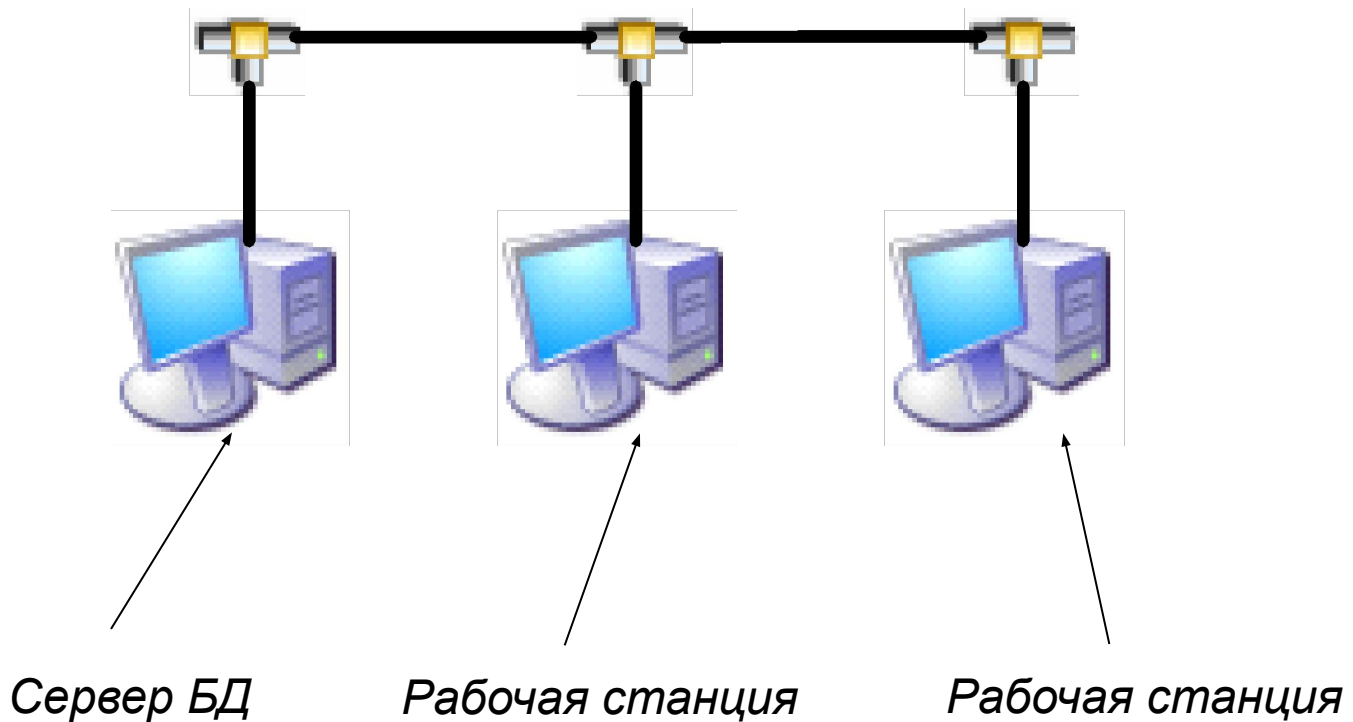


Сервер БД

Логическая и физическая структура БД

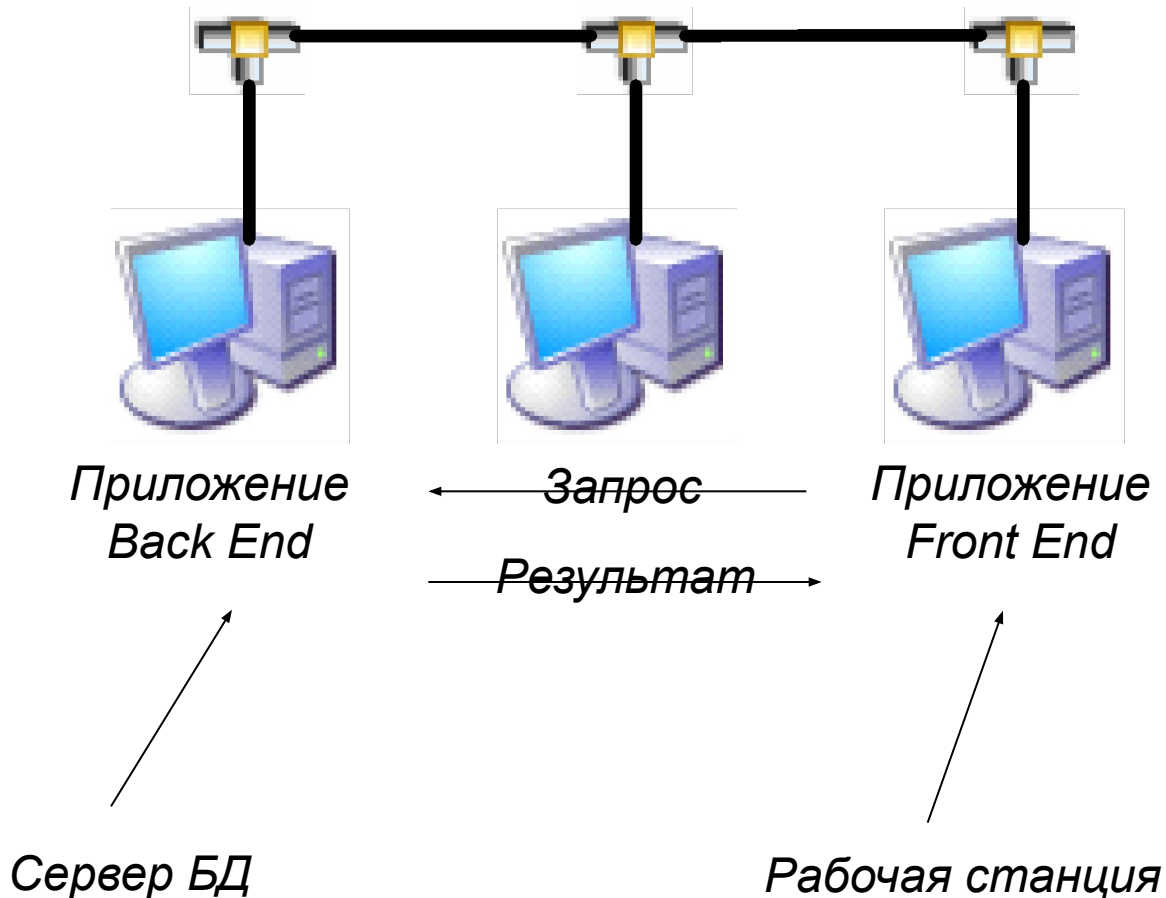
Технология физического хранения БД

2 вариант



Логическая и физическая структура БД

Технология клиент-сервер



Логическая и физическая структура БД

Технология клиент-сервер

1. При технологии клиент-сервер клиентское приложение (front-end) формирует запрос к серверу БД (back-end), на котором выполняются все команды. Результаты команд посылаются затем клиенту для использования и просмотра.
2. Visual FoxPro, Visual Basic и Access обеспечивают средства для создания клиентских частей в приложениях клиент-сервер, которые сочетают мощность, скорость, графический интерфейс, продвинутое средства построения запросов и отчетов.
3. MS SQL Server является на настоящий Момент одним из наиболее мощных серверов БД.

Логическая и физическая структура БД

Технология клиент-сервер

1. Важнейшим этапом в построении приложения клиент-сервер является установка связи клиентского приложения с источником данных, находящимся на сервере БД. В настоящий момент различные средства разработки используют несколько технологий обеспечения доступа к данным. Общепризнанным стандартом является технология ODBC

Открытый доступ к данным — Open Database Connectivity (ODBC) — это общее определение языка и набор протоколов. ODBC позволяет клиентскому приложению, написанному, например, на Access или Visual FoxPro, работать с командами и функциями, поддерживаемыми сервером.

Логическая и физическая структура БД

Технология клиент-сервер

1. В качестве сервера может выступать любой сервер БД, имеющий драйвер ODBC (MS SQL Server, Oracle и т. д.), или даже настольная база данных, ведь часто может возникнуть необходимость в совместной обработке данных, хранящихся в формате Paradox, приложениями, написанными и на FoxPro и на Delphi. ODBC находится как бы посередине между приложением, использующем данные, и самими данными, хранящимися в формате, которые мы не можем обработать напрямую в приложении, и используется как средство коммуникации между двумя front-end и back-end сторонами.

Основные принципы проектирования БД

Этапы проектирования БД

1. Определение классов объектов (сущностей).
2. Определение взаимосвязей между сущностями.
3. Задание первичных и альтернативных ключей.
Определение атрибутов (полей)
4. Приведение к требуемому уровню нормализации.
5. Физическое описание модели.

ИНФОРМАЦИОННЫЕ СИСТЕМЫ

(Нормализация отношений)

Нормализация отношений - это процесс построения оптимальной структуры таблиц и связей в реляционной БД (процесс уменьшения избыточности информации).

В процессе нормализации данные группируются в таблицы, представляющие классы объектов и их взаимодействие.

ИНФОРМАЦИОННЫЕ СИСТЕМЫ (Нормализация отношений)

Цели, которые преследуются при построении наиболее эффективной структуры данных:

- обеспечить быстрый доступ к данным;
- исключить ненужное повторение данных, которое может являться причиной ошибок при вводе, а также привести к нерациональному использованию дискового пространства;
- обеспечить целостность данных, т.о. чтобы при изменении одних объектов автоматически происходило соответствующее изменение связанных с ними объектов

Теория нормализации отношений работает с 5 нормальными формами таблиц. Каждая последующая форма должна отвечать требованиям предыдущих плюс некоторые дополнительные требования.

ИНФОРМАЦИОННЫЕ СИСТЕМЫ

(Первая нормальная форма (1НФ))

Таблица, находящаяся в первой нормальной форме должна отвечать следующим требованиям:

1. таблица не должна иметь повторяющихся записей.
2. в таблице должны отсутствовать повторяющиеся группы полей.

ИНФОРМАЦИОННЫЕ СИСТЕМЫ

(Первая нормальная форма (1НФ))

Для приведения к 1НФ можно использовать следующий алгоритм:

1. Определить поле, которое можно назначить первичным ключом. Если такого поля нет, то добавить новое уникальное ключевое поле.
2. Определить группы повторяющихся полей.
3. Вынести группы повторяющихся полей в отдельные таблицы, в основной таблице остается одно поле для организации связи между таблицами.
4. Назначить первичные ключи в новых таблицах. (В качестве ключевых полей можно использовать поля таблицы или добавить новое поле. Если ключевое поле имеет большой размер, предпочтительней добавлять новое поле.)
5. Определить тип отношения между таблицами.

ИНФОРМАЦИОННЫЕ СИСТЕМЫ

(Первая нормальная форма (1НФ))

Пусть дана однотабличная БД

Заказы

Дата заказа
Фамилия и инициалы клиента
Телефон клиента
Номер счета
Наименование товара
Цена
Фирма производитель
Индекс фирмы производителя
Адрес фирмы производителя
Количество товара
Отметка об оплате

Поскольку один клиент может в течении одного дня сделать несколько заказов, таблица может содержать одинаковые строки. На один и тот же товар может поступить несколько заказов, и наконец, фирма производитель может поставлять несколько товаров. Таблица может содержать следующие данные:

ИНФОРМАЦИОННЫЕ СИСТЕМЫ (Первая нормальная форма (1НФ))

Дата заказа	Фамилия и инициалы клиента	Телефон клиента	Номер счета	Наименование товара
01/03/00	Семенова А.Н.	23-13-87	2343200000345550099	Алебастр
01/03/00	Семенова А.Н.	23-13-87	2343200000345550099	Алебастр
02/03/00	Скоробогатов И.П.	88-76-80	4560000874567558856	Лак
04/03/00	Кузнецов А.В.	22-11-43	2349885746580009585	Краска

Цена	Фирма производитель	Индекс фирмы производителя	Адрес фирмы производителя	Кол-во товара	Отметка об оплате
24.00	«Невбытхим»	456344	Володарский 43	4	<input checked="" type="checkbox"/>
24.00	«Невбытхим»	456344	Володарский 43	2	<input type="checkbox"/>
35.70	«Невбытхим»	456344	Володарский 43	5	<input type="checkbox"/>
43.00	«Росхим»	857805	Бергарина 23/3	10	<input checked="" type="checkbox"/>

Приведем БД к 1НФ.

ИНФОРМАЦИОННЫЕ СИСТЕМЫ (Первая нормальная форма (1НФ))

Воспользуемся алгоритмом для приведения к 1НФ :

1. Добавим в таблицу поле Номер заказа, что позволит однозначно идентифицировать каждый из заказов.
2. Таблица содержит три группы повторяющихся полей:

Поля характеризующие клиента:

Фамилия и инициалы клиента

Телефон клиента,

Номер счета

Вынесем их в отдельную таблицу **Клиенты**

ИНФОРМАЦИОННЫЕ СИСТЕМЫ

(Первая нормальная форма (1НФ))

Продолжение алгоритма для приведения к 1НФ :

Поля характеризующие Товар:

Наименование товара

Цена

Вынесем их в отдельную таблицу **Товары**

Поля характеризующие Производителя:

Фирма производитель

Индекс фирмы производителя

Адрес фирмы производителя

Вынесем их в отдельную таблицу **Производители**

ИНФОРМАЦИОННЫЕ СИСТЕМЫ

(Вторая нормальная форма (2НФ))

Таблица, находящаяся во второй нормальной форме должна отвечать всем требованиям 1НФ, а также любое неключевое поле однозначно идентифицируется полным набором ключевых полей

2НФ применяется к таблицам, которые имеют **составной** ключ.

Частичнозависимое поле поле, зависящее только от части ключа

ИНФОРМАЦИОННЫЕ СИСТЕМЫ

(Вторая нормальная форма (2НФ))

Пример (таблица «проекты»)

Код проекта
Название
Основная цель
Продолжительность
Код руководителя проекта
Фамилия руководителя
Телефон руководителя

Пусть таблица имеет составной ключ, который формируется по двум полям: Код проекта, Код руководителя проекта.

Поля Название, Основная цель, Продолжительность, Фамилия руководителя Телефон руководителя являются частичнозависимыми. Т.к. Название, Основная цель, Продолжительность зависят только от Кода проекта, но не зависят от Кода руководителя, т.е. однозначно идентифицируется частью ключа, а не полным набором ключевых полей. Аналогично поля Фамилия руководителя и Телефон руководителя зависят только от Кода руководителя проекта.

ИНФОРМАЦИОННЫЕ СИСТЕМЫ (Вторая нормальная форма (2НФ))

Алгоритм для приведения ко 2НФ :

1. вынести все частичнозависимые поля в отдельную таблицу;
2. определить ключевые поля;
3. установить отношения между таблицами.

Проекты
Код проекта
Название
Основная цель
Продолжительность
Код руководителя проекта

Руководители
Код руководителя
Фамилия руководителя
Телефон руководителя

ИНФОРМАЦИОННЫЕ СИСТЕМЫ

(Третья нормальная форма (3НФ))

Таблица, находящаяся в третьей нормальной форме должна отвечать всем требованиям 2НФ, а также ни одно из неключевых полей не идентифицируется при помощи другого неключевого поля.

Другими словами в таблице **нет полей**, которые **не зависят** от ключа.

Пример

Заказы

Код заказа	← первичный ключ
Фамилия покупателя	
Дата продажи	
Код менеджера	
Фамилия менеджера	

Таблица не находится в 3НФ, т.к. неключевое поле Фамилия менеджера зависит от другого неключевого поля Код менеджера.

ИНФОРМАЦИОННЫЕ СИСТЕМЫ

(Третья нормальная форма (3НФ))

Алгоритм для приведения к 3НФ :

Для приведения к 3НФ необходимо вынести поле Фамилия менеджера в отдельную таблицу

