

ТЕМА 3

Взаимосвязи
в модели данных

При проектировании БД нам потребуется различать взаимосвязи:

↙
между
объектами

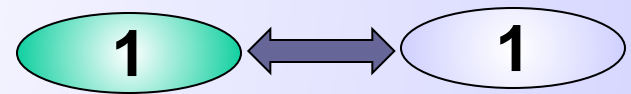
↓
между атрибутами
одного объекта

↘
и между атрибутами
различных объектов

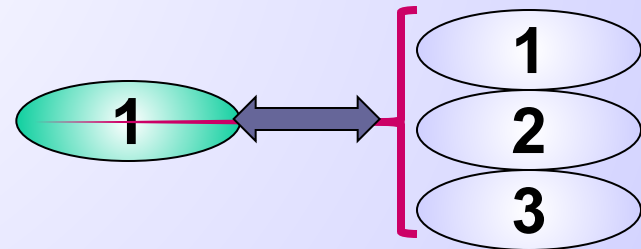
Взаимосвязь показывает взаимодействие
2-х множеств различных объектов.

Различают
взаимосвязи
типа:

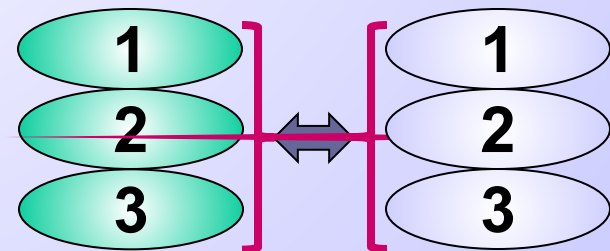
«ОДИН К ОДНОМУ»



«ОДИН КО МНОГИМ»



«МНОГИЕ КО МНОГИМ»

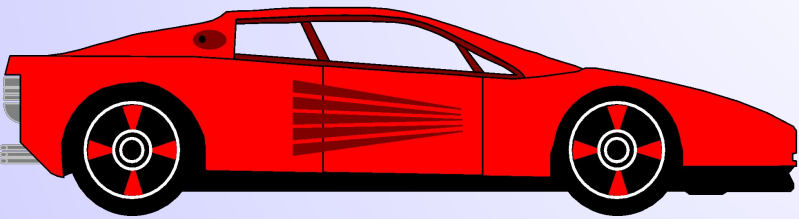


Общее описание предметной области...

Рассмотрим связи между объектами на примере автоматизации управления работой дилера по продаже легковых автомобилей.

Некоторые концептуальные требования

1. Если клиент производит заказ на покупку автомобиля впервые, то осуществляется первичная регистрация его данных и сведений о сделанном заказе.



2. Если клиент производит заказ повторно, осуществляется регистрация только данного заказа.

3. Вне зависимости от того, сколько раз данный клиент производил заказы, он имеет уникальный идентификационный номер (уникальный ключ клиента).



Информация о каждом клиенте включает:

- фамилию, имя, отчество
- адрес
- телефон
- факс
- наименование клиента
- признак юридического лица
- примечание

Можно выделить следующие объекты:

КЛИЕНТ

ЗАКАЗ

ПРОДАВЕЦ

МОДЕЛЬ АВТОМОБИЛЯ

Выделим атрибуты
объекта КЛИЕНТ:

«УНИКАЛЬНЫЙ КЛЮЧ»
«НАИМЕНОВАНИЕ КЛИЕНТА»
«АДРЕС КЛИЕНТА»

и т.д. (пока)



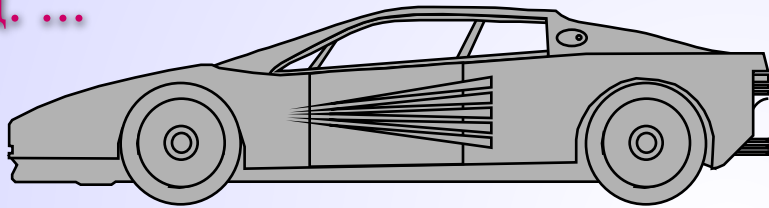
Атрибуты объекта

МОДЕЛЬ АВТОМОБИЛЯ:

«УНИКАЛЬНЫЙ КЛЮЧ МОДЕЛИ»

«НАИМЕНОВАНИЕ МОДЕЛИ»

и т.д. ...



Атрибуты объекта

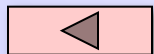
ПРОДАВЕЦ:

«УНИКАЛЬНЫЙ КЛЮЧ ПРОДАВЦА»

«ФАМИЛИЯ»

«ИМЯ ПРОДАВЦА»

«ОТЧЕСТВО ПРОДАВЦА»



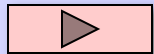
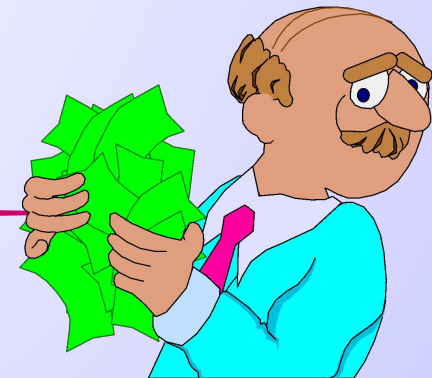
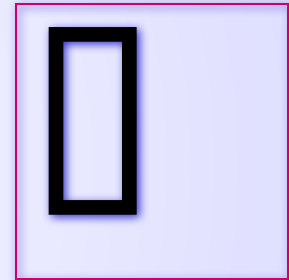
Атрибуты объекта

ЗАКАЗ:

«КЛЮЧ КЛИЕНТА»

«КЛЮЧ МОДЕЛИ»

«КЛЮЧ ПРОДАВЦА»



Взаимосвязь «один к одному»

(между 2-мя типами объектов)

- ✓ Допустим, что в определенный момент времени один клиент может сделать только один заказ...

Это
допущение
сделано
с
некоторой
натяжкой.

В этом случае между объектами КЛИЕНТ и ЗАКАЗ устанавливается взаимосвязь «один к одному».

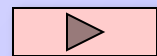
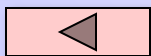


Взаимосвязь «один ко многим»

(между 2-мя типами объектов)

- ✓ В определенный момент

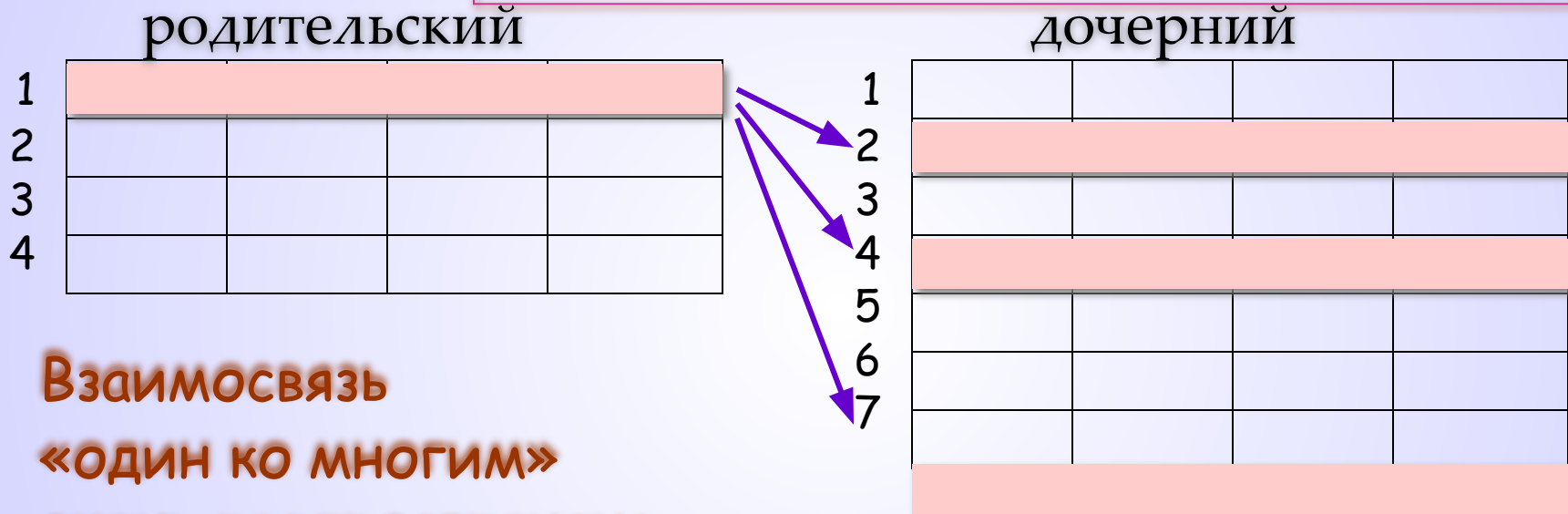
времени один клиент может стать обладателем нескольких автомобилей, при этом один автомобиль не может принадлежать нескольким клиентам одновременно.



Продолжение...

В этом случае:

одной записи первого объекта (родительского) будет соответствовать несколько записей второго объекта (дочернего или подчиненного).



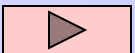
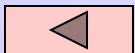
Взаимосвязь

«один ко многим»

очень распространена

при разработке реляционных баз данных.

В качестве родительского объекта часто выступает справочник, а в дочернем хранятся уникальные ключи для доступа к записям справочника.



В нашем примере:

в качестве справочника можно представить объект **КЛИЕНТ**, в котором хранятся сведения о всех клиентах.

При обращении к записи определенного клиента нам будет доступен список всех покупок, которые он сделал и сведения о которых хранятся в объекте **МОДЕЛЬ АВТОМОБИЛЯ**.

Объект
«КЛИЕНТ»

УНИКАЛЬНЫЙ КЛЮЧ	НАИМЕНОВАНИЕ КЛИЕНТА
1	Хитрая лиса
2	Злой волк

«Хитрая лиса»
приобрел
Mercedes, BMW и
Toyota Corolla

Объект «МОДЕЛЬ АВТОМОБИЛЯ»

УНИКАЛЬНЫЙ КЛЮЧ МОДЕЛИ	НАИМЕНОВАНИЕ МОДЕЛИ	КЛЮЧ КЛИЕНТА
7	MERCEDES 600	1
8	FERRARI	2
9	BMW	1
10	TOYOTA COROLLA	1



Как определить какой тип связи существует между объектами 2-х множеств?

Советы...

1. Из первого множества интересующих объектов нужно выделить один и проанализировать, с каким числом объектов другого множества он связан (взаимодействует).

Если этот объект взаимодействует со многими объектами другого множества, то это предпосылка к типу связи один-ко-многим.

2. С другой стороны, необходимо проанализировать ситуацию второго множества объектов, также выделив конкретного представителя этого множества.



Продолжение...

Если объект второго множества взаимодействует с одним конкретным объектом первого множества, то тогда делается ВЫВОД:

Тип этой связи - ОДИН-КО-МНОГИМ.

Если объект второго множества взаимодействует со многими объектами первого множества, то тогда делается ВЫВОД:

Тип этой связи - МНОГИЕ-КО-МНОГИМ.

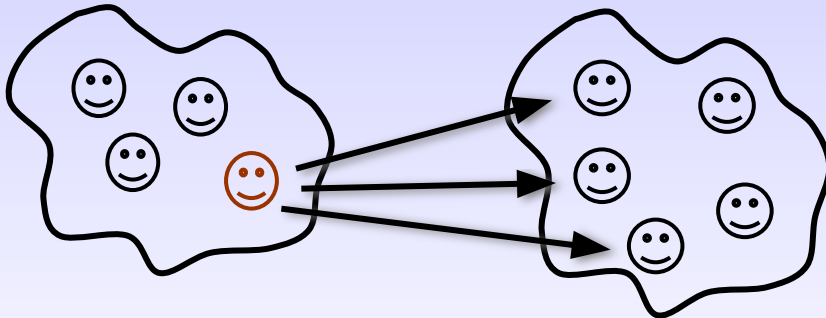
Если же объекты первого и второго множеств одинаково взаимодействуют друг с другом в одиночку, то делается ВЫВОД:

Тип этой связи - ОДИН-К-ОДНОМУ

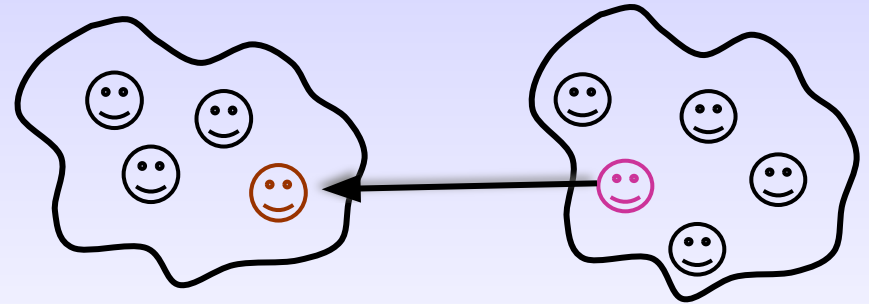


Продолжение...

Примеры

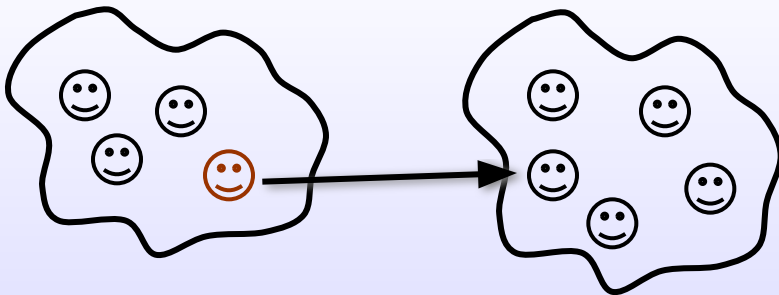


Один связан со многими

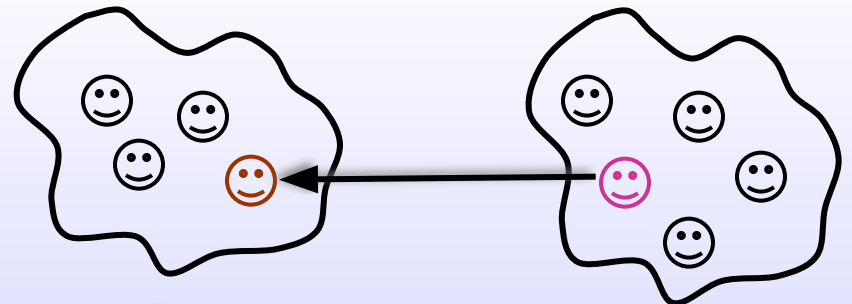


Один связан с одним

ТИП СВЯЗИ – ОДИН-КО-МНОГИМ



Один связан с одним



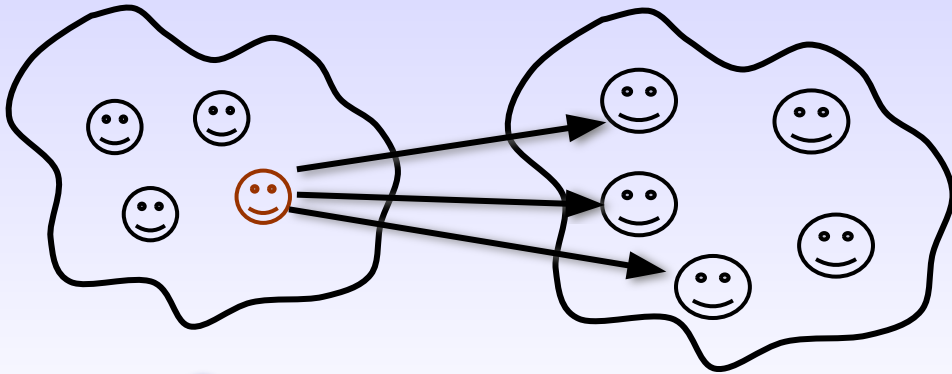
Один связан с одним

ТИП СВЯЗИ – ОДИН-К-ОДНОМУ

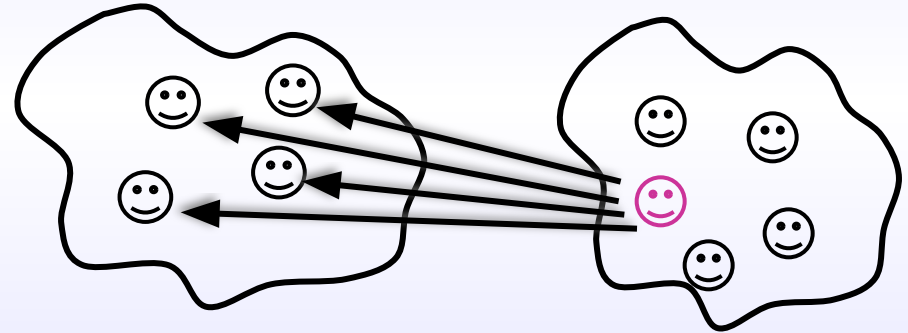


Продолжение...

Примеры



Один связан со
МНОГИМИ



Один связан со
МНОГИМИ

ТИП СВЯЗИ

– МНОГИЕ-КО-МНОГИМ



Замечания:

В случае, если в дочернем объекте (МОДЕЛЬ АВТОМОБИЛЯ) будут какие-то записи, для которых нет соответствующих записей в объекте КЛИЕНТ, то они называются ОДИНОКИМИ или ПОТЕРЯННЫМИ.

Для потерянных записей сведений о клиентах в таблице КЛИЕНТ мы не получим...

Это недопустимо!!!

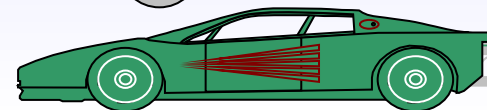
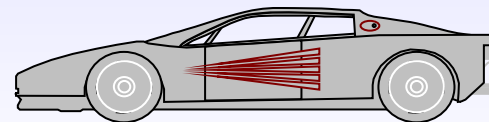
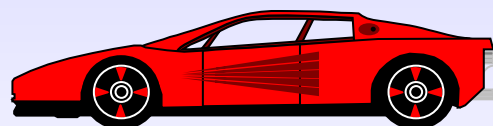
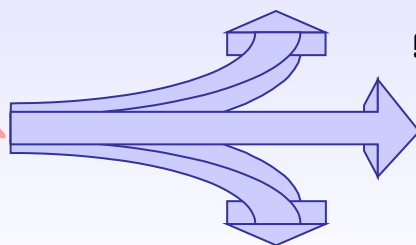
Таких ситуаций в таблицах
нужно избегать.



Взаимосвязь «многие ко многим»

(между 2-мя типами объектов)

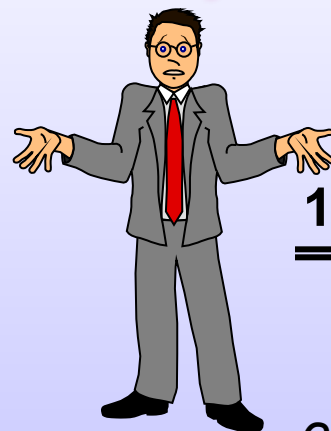
✓ В рассматриваемом примере продавец может обслужить нескольких клиентов (т.е. продать автомобиль).



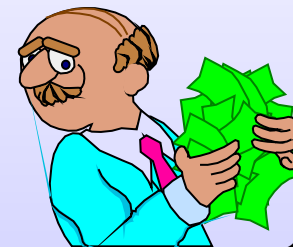
✓ С другой стороны, приобретая автомобиль в различное время, каждый клиент вполне может быть обслужен различными продавцами.



12¹⁵ час.



15³⁰ час.

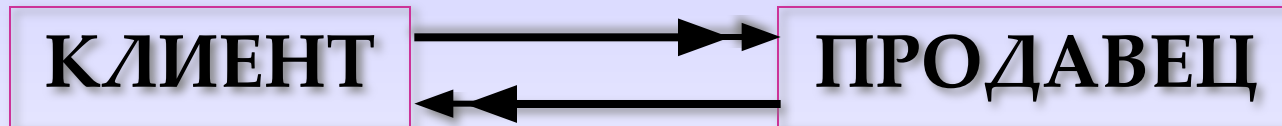


17¹⁰ час.

обслуживание...



Между объектами **КЛИЕНТ** и **ПРОДАВЕЦ** существует взаимосвязь «многие ко многим» .



НАПРИМЕР:

Объект «КЛИЕНТ»

УНИКАЛЬНЫЙ КЛЮЧ	НАИМЕНОВАНИЕ КЛИЕНТА
1	Хитрая лиса
2	Злой волк

Объект «ПРОДАВЕЦ»

КЛЮЧ ПРОДАВЦА	ИМЯ ПРОДАВЦА	КЛЮЧ КЛИЕНТА
1	КАРИНА	1
1	КАРИНА	2
2	ЛЕНА	1
2	ЛЕНА	2

При просмотре данных в таблице «ПРОДАВЕЦ» мы сможем узнать, какие продавцы обслуживали определенного клиента.



Однако в объекте «ПРОДАВЕЦ» придется завести несколько записей для каждого продавца.

Каждая запись будет соответствовать каждому обслуживанию клиента.

При таком подходе мы столкнемся с серьезными ПРОБЛЕМАМИ!

Например,

мы не сможем ввести в объекте ПРОДАВЕЦ уникальный ключ продавца, т.к. один продавец будет обслуживать нескольких клиентов.

Что делать в этой ситуации?



РЕШЕНИЕ

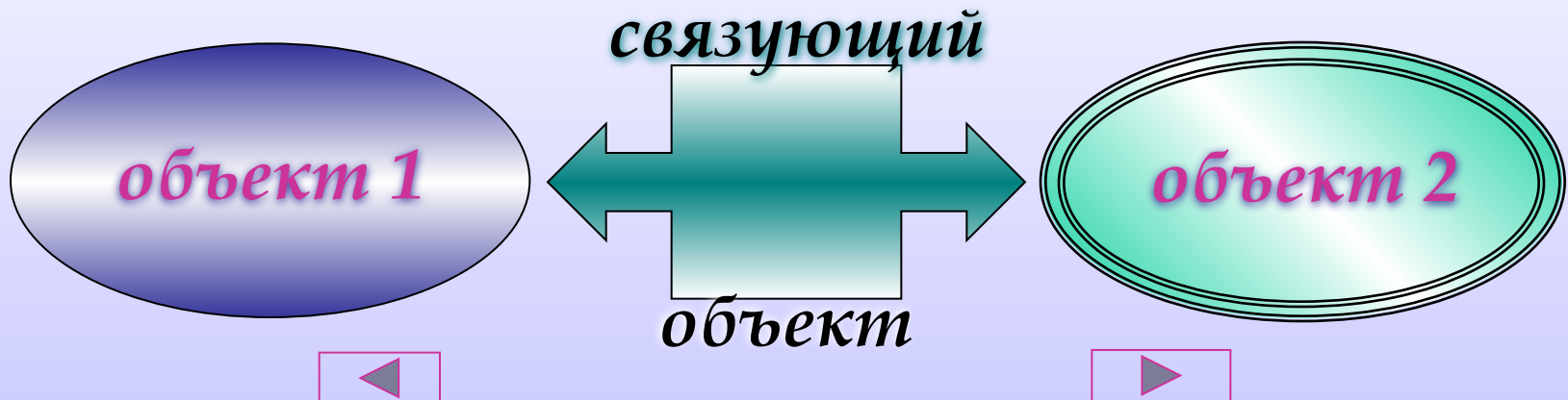
согласно теории

реляционных баз данных для хранения
взаимосвязи «многие ко многим»

требуется 3-и объекта:

по одному для каждой сущности и один для
хранения связей между ними
(промежуточный объект).

Промежуточный объект будет содержать
идентификаторы связанных объектов.



Преобразованные таблицы

Объект «КЛИЕНТ»

УНИКАЛЬНЫЙ КЛЮЧ	НАИМЕНОВАНИЕ КЛИЕНТА
1	Хитрая лиса
2	Злой волк

Связующий объект

КЛЮЧ КЛИЕНТА	КЛЮЧ ПРОДАВЦА
1	1
1	2
2	1
2	2

Объект «ПРОДАВЕЦ»

УНИКАЛЬНЫЙ КЛЮЧ ПРОДАВЦА	ИМЯ ПРОДАВЦА
1	Карина
2	Лена



Взаимосвязи между атрибутами

Наряду с взаимосвязями между объектами существуют взаимосвязи между атрибутами объектов.

Здесь различают те же
типы взаимосвязей

1.

1↔1

— Один к одному

— Один ко многим

— Многие ко многим

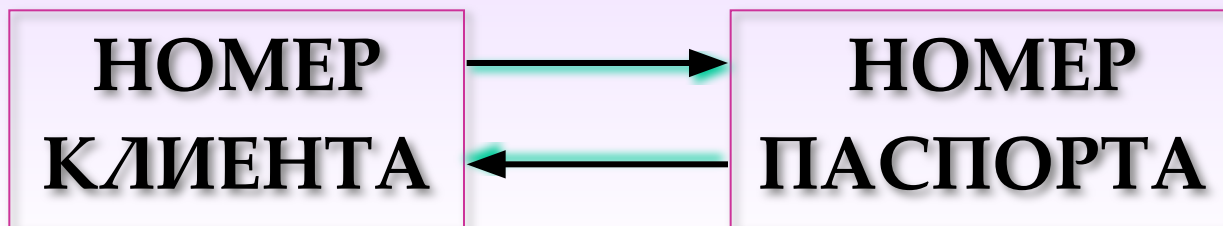
Допустим, клиент имеет ключ (номер), который является уникальным идентификатором.

Он не изменяется при поступлении заказов от данного клиента.



Если наряду с номером клиента в БД хранится другой уникальный идентификатор (например, номер паспорта), то между ними существует связь:

ОДИН-К-ОДНОМУ



2. 1 ⇔ ∞

Имя клиента и его номер существуют совместно.

Клиентов с одинаковыми именами может быть много, но все они имеют различные номера.

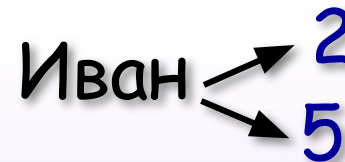
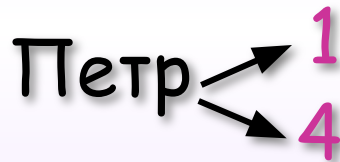
Каждому клиенту присваивается уникальный номер.

Это означает, что данному номеру соответствует только одно имя.



НАПРИМЕР:

ИМЯ	ФАМИЛИЯ	НОМЕР КЛИЕНТА
Петр	Снигирев	1
Иван	Львов	2
Олег	Медведев	3
Петр	Бердяев	4
Иван	Соколов	5



В итоге:

ИМЯ
КЛИЕНТА

НОМЕР
КЛИЕНТА




Имеем связь типа:

ОДИН-КО-МНОГИМ



3. 



 *Несколько клиентов с одинаковыми именами могли быть обслужены несколькими продавцами.*

Несколько продавцов с одинаковыми именами могли получить заказы от нескольких клиентов.



Между атрибутами
«ИМЯ КЛИЕНТА» и «ИМЯ ПРОДАВЦА»
существует взаимосвязь

МНОГИЕ-КО-МНОГИМ

ИМЯ
КЛИЕНТА

ИМЯ
ПРОДАВЦА



Типы логической модели данных

Существует три основных типа логической модели данных:

ИЕРАРХИЧЕСКАЯ

РЕЛЯЦИОННАЯ

СЕТЕВАЯ

Они отличаются друг от друга способами представления взаимосвязей между объектами.

Иерархическая и сетевая модели данных стали применяться в СУБД в начале 60-х годов. В начале 70-х годов была предложена реляционная модель данных.



Иерархическая модель

строится по принципу иерархии объектов

Один тип объекта является главным, а остальные, находящиеся на низших уровнях иерархии, - подчиненными.

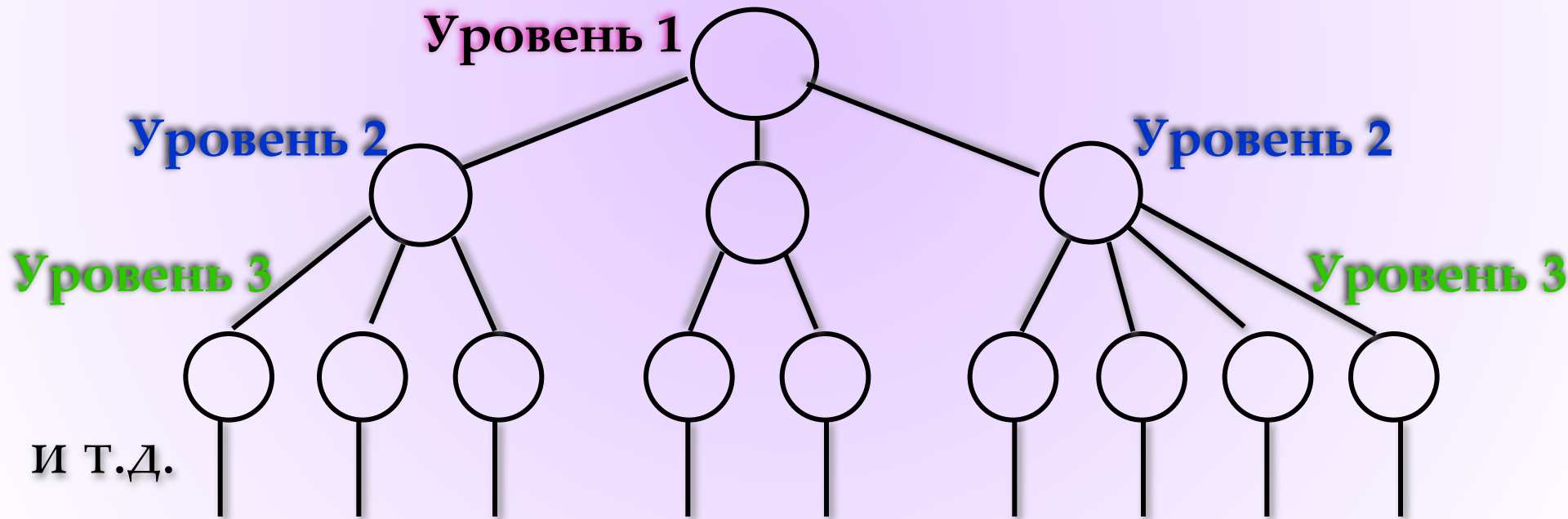
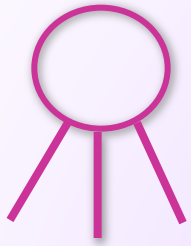


Схема иерархической модели



Между главным и подчиненными объектами

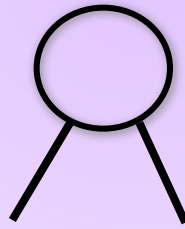
устанавливается связь «ОДИН КО МНОГИМ».



Аналогично, для каждого экземпляра главного объекта может быть несколько экземпляров подчиненных типов объектов.

Узлы и ветви образуют иерархическую древовидную структуру, которая предполагает несколько уровней подчинения.

Узел - совокупность атрибутов, описывающих объект.



Наивысший в иерархии узел называется корневым (это главный тип объекта).

Корневой узел находится на первом уровне иерархии, подчиненные - на втором, третьем и т.д. уровнях.

Примеры:

иерархическая модель применяется для организаций различного профиля: военных, административных, образовательных и т.д.

Сетевая модель

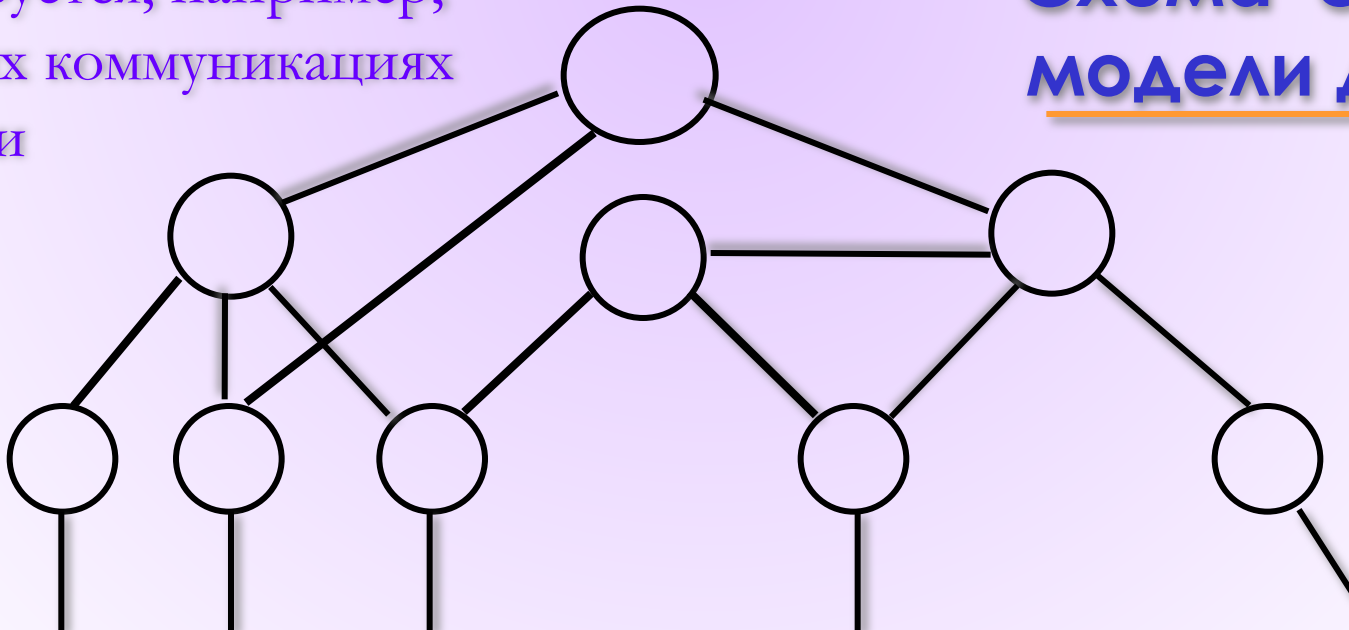
В сетевой модели данных понятия главного и подчиненного объектов несколько расширены.

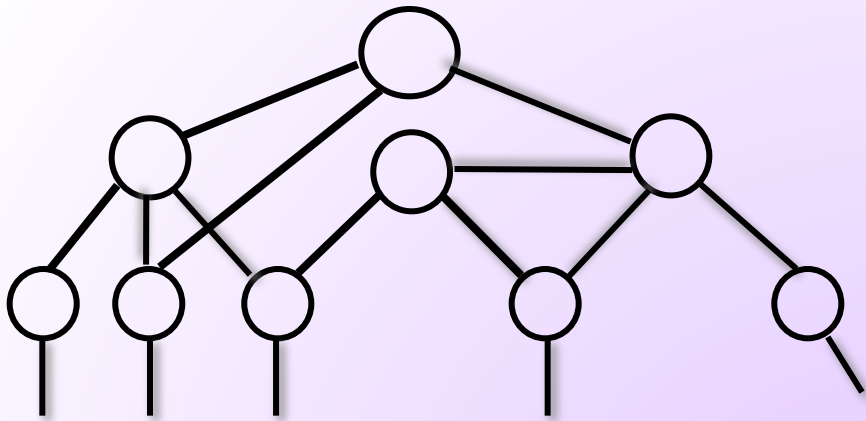
Любой объект может быть и главным и подчиненным.

Главный объект обозначается термином «владелец набора», а подчиненный - термином «член набора».

Используется, например, в сетевых коммуникациях передачи данных.

Схема сетевой модели данных





Один и тот же объект
в данной модели может
одновременно выступать:
и в роли владельца,
и в роли члена набора.

Это означает, что каждый объект может
участвовать в любом числе взаимосвязей.

Реляционная модель данных

В реляционной модели данных объекты и
взаимосвязи между ними представляются
с помощью таблиц.

Взаимосвязи также рассматриваются в
качестве объектов.



Схема реляционной модели

Таблица 1

Связь 2



Таблица 2

Связь 1



Таблица 3

Каждая таблица
представляет один объект.

В данной модели

таблица должна иметь **первичный ключ** (ключевой элемент) - поле или комбинацию полей, которые единственным образом идентифицируют каждую строку в таблице.

Благодаря своей простоте и естественности представления



реляционная модель получила наибольшее



распространение в СУБД для персональных компьютеров.

Тема

закончен

а

