



# Лекция №4

## Постреляционные модели данных

- » *Постреляционная модель* представляет собой расширенную реляционную модель, снимающую ограничение неделимости данных.
- » Модель допускает *многозначные поля* – поля, значения которых состоят из подзначений.





## Накладные

№ накладной	Покупатель
0373	8723
8374	8232
7364	8723

## Накладные-товары

№ накладной	Товар	Количество
0373	Сыр	3
0373	Рыба	2
8374	Лимонад	1
8374	Сок	6
8374	Печенье	2
7364	Йогурт	1

## Накладные

№ накладной	Покупатель	Товар	Количество
0373	8723	Сыр	3
		Рыба	2
8374	8232	Лимонад	1
		Сок	6
		Печенье	2
7364	8723	Йогурт	1



» По сравнению с реляционной моделью в постреляционной модели данные хранятся более эффективно, а при обработке не требуется выполнять операцию соединения данных из двух таблиц.





## *Достоинства*

» возможность представления совокупности связанных реляционных таблиц одной постреляционной таблицей. Это обеспечивает высокую наглядность представления информации и повышение эффективности ее обработки.

## *Недостатки*

» сложность решения проблемы обеспечения целостности и непротиворечивости хранимых данных.



# *Многомерная модель данных*

Многомерные СУБД являются узкоспециализированными СУБД, предназначенными для интерактивной аналитической обработки информации.

Основные понятия, используемые в этих СУБД:

- » агрегируемость ,
- » историчность
- » прогнозируемость.





- » **Агрегируемость** данных означает рассмотрение информации на различных уровнях ее обобщения. В ИС степень детальности представления информации для пользователя зависит от его уровня: аналитик, пользователь, управляющий, руководитель.
- » **Историчность** данных предполагает обеспечение высокого уровня статичности собственно данных и их взаимосвязей, а также обязательность привязки данных ко времени.
- » **Прогнозируемость** данных подразумевает задание функций прогнозирования и применение их к различным временным интервалам .



## » Реляционное и многомерное представление данных

Модель	Месяц	Объем		Модель	Июнь	Июль	Август
Жигули	июнь	12		Жигули	12	24	5
Жигули	июль	24		Москвич	2	18	No
Жигули	август	5		Волга	No	19	No
Москвич	июнь	2					
Москвич	июль	18					
Волга	июль	19					➤



## Основные понятия многомерных моделей данных:

» *Измерение* – это множество однотипных данных, образующих одну из граней гиперкуба. Играют роль индексов, служащих для идентификации конкретных значений в ячейках.

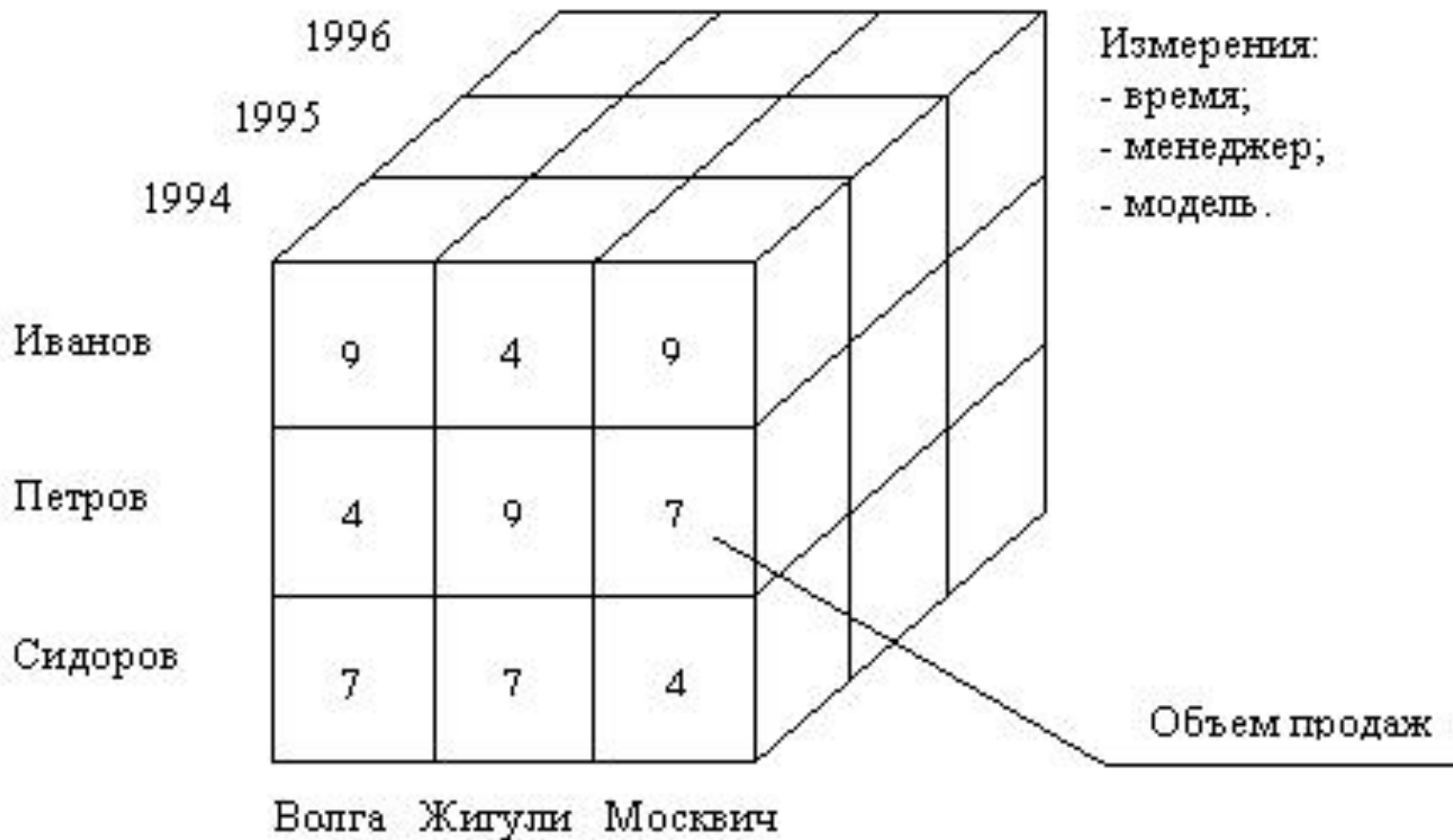
» *Ячейка* – это поле, значение которого однозначно определяется фиксированным набором измерений.

- ✓ Тип поля определен как цифровой.
- ✓ В зависимости от того, как формируются значения некоторой ячейки, она может быть переменной либо формулой.



В примере каждое значение ячейки *Объем продаж* однозначно определяется комбинацией временного измерения *Месяц продаж* и модели автомобиля.

Пример трехмерной модели данных.





В существующих многомерных СУБД используются две основных схемы организации данных:

»В *поликубической* схеме предполагается, что в БД может быть определено несколько гиперкубов с различной размерностью и с различными измерениями в качестве граней.

»В случае *гиперкубической* схемы предполагается, что все ячейки определяются одним и тем же набором измерений. Это означает, что при наличии нескольких гиперкубов в БД, все они имеют одинаковую размерность и совпадающие измерения.



» *Достоинство* - удобство и эффективность аналитической обработки больших объемов данных, связанных со временем.

» *Недостатком* является ее громоздкость для простейших задач обычной оперативной обработки информации.

»





# *Объектно-ориентированная модель*

- » При представлении данных имеется возможность идентифицировать отдельные записи базы данных.
- »
- » Между записями и функциями их обработки устанавливаются взаимосвязи



Структура графически представима в виде дерева, узлами которого являются **объекты**.

**Свойства** объектов описываются некоторым стандартным типом или типом, конструируемым пользователем (определяется как `class` ).

Значение свойства типа `class` есть объект, являющийся экземпляром соответствующего класса.

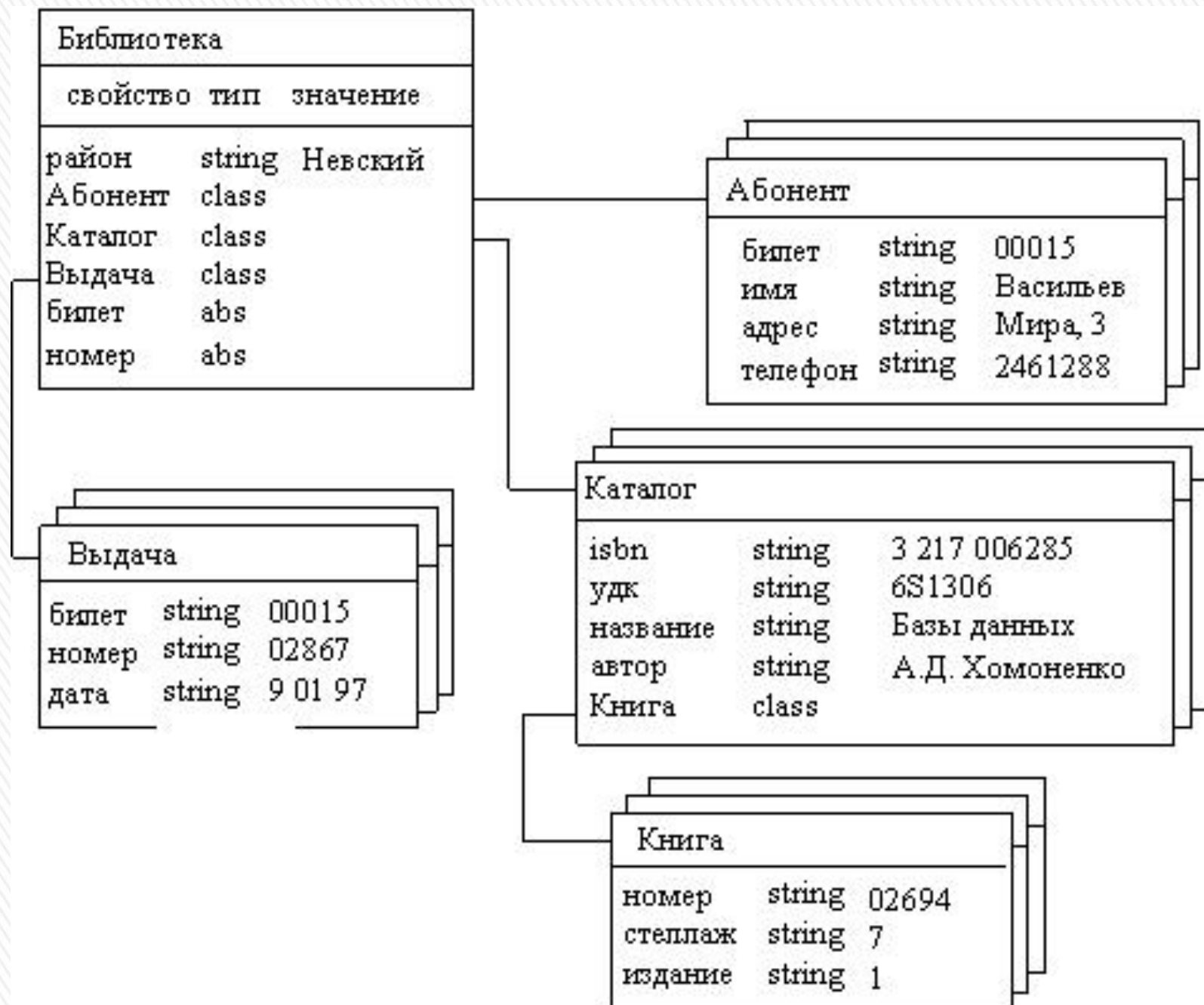
Каждый объект-экземпляр класса считается потомком объекта, в котором он определен как свойство.

Объект-экземпляр класса принадлежит своему классу и имеет одного родителя.





## » Логическая структура БД библиотечного дела



Здесь объект типа *Библиотека* является родительским для объектов-экземпляров классов *Абонент* , *Каталог* и *Выдача* .

Различные объекты типа *Книга* а могут иметь одного или разных родителей.

Объекты типа *Книга* , имеющие одного и того же родителя, должны различаться, по крайней мере, инвентарным номером (уникален для каждого экземпляра книги), но имеют одинаковые значения свойств *isbn* , *удк* , *название* и *автор* .





Для выполнения действий над данными в рассматриваемой модели БД применяются логические операции

**Инкапсуляция** ограничивает область видимости имени свойства пределами того объекта, в котором оно определено. Так, если в объект типа *Каталог* добавить свойство, задающее телефон автора книги и имеющее название *телефон*, то мы получим одноименные свойства у объектов *Абонент* и *Каталог*. Смысл такого свойства будет определяться тем объектом, в который оно инкапсулировано.

**Наследование**, наоборот, распространяет область видимости свойства на всех потомков объекта. Так, всем объектам типа *Книга*, являющимся потомками объекта типа *Каталог*, можно приписать свойства объекта-родителя: *isbn*, *удк*, *название* и *автор*.



*Полиморфизм* в объектно-ориентированных языках программирования означает способность одного и того же программного кода работать с разнотипными данными.

Применительно к рассматриваемому примеру полиморфизм означает, что объекты класса *Книга*, имеющие разных родителей из класса *Каталог*, могут иметь разный набор свойств. Следовательно, программы работы с объектами класса *Книга* могут содержать полиморфный код.





- » Основным *достоинством* является возможность отображения информации о сложных взаимосвязях объектов. Объектно-ориентированная модель данных позволяет идентифицировать отдельную запись базы данных и определять функции их обработки.
- » *Недостатками* объектно-ориентированной модели являются высокая понятийная сложность, неудобство обработки данных и низкая скорость выполнения запросов.

