



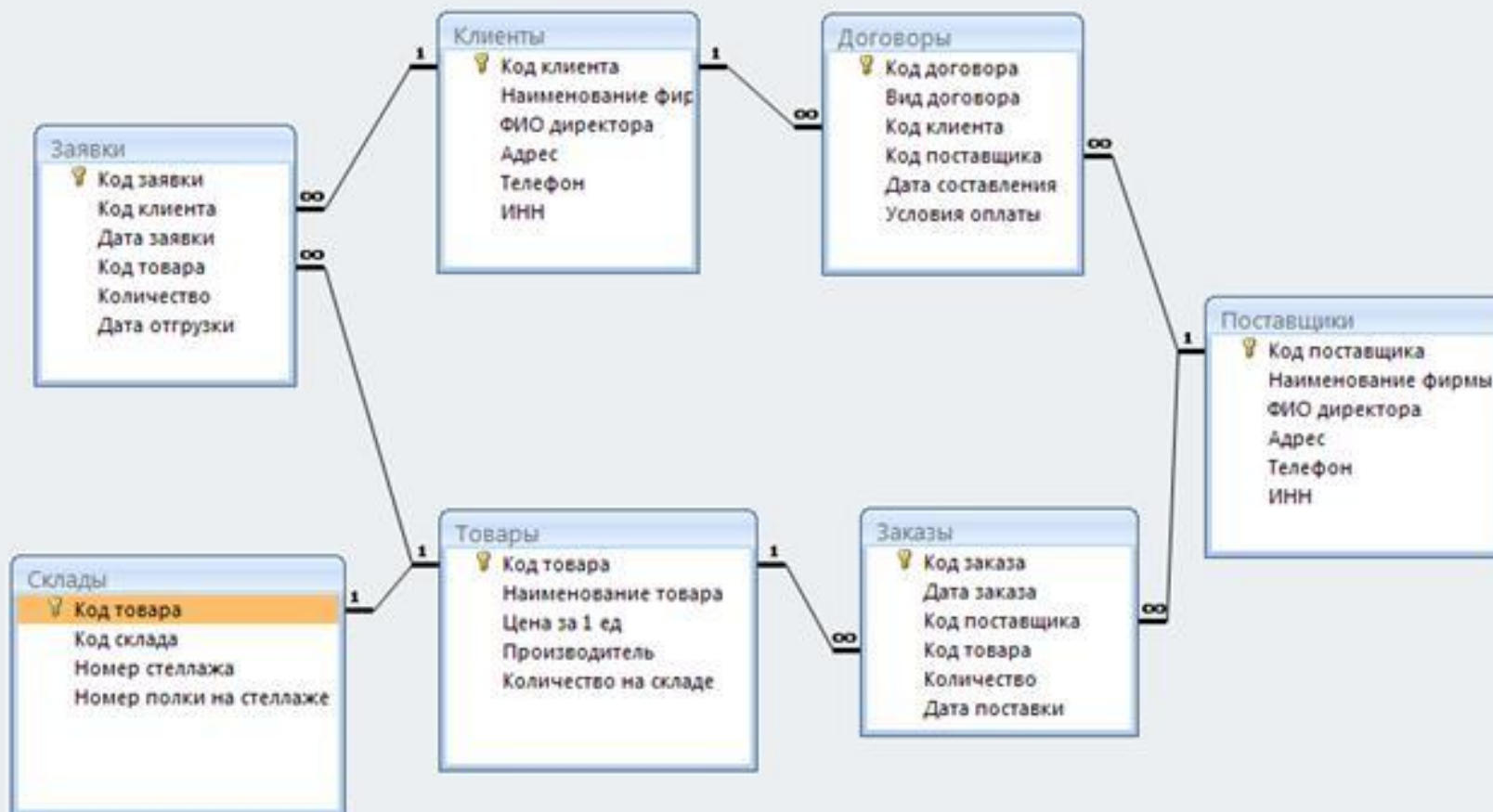
# ТЕМА УРОКА: МОДЕЛИ ПРЕДСТАВЛЕНИЯ ДАННЫХ

# Понятие о моделях данных. Пример

---

- **Модель данных** - это совокупность взаимосвязанных структур данных и операций над этими структурами. Она позволяет пользователям трактовать данные как информацию (сведения, содержащие не только данные, но и связи между ними).
- С помощью модели данных могут быть представленные объекты предметной области и взаимосвязи между ними.

# НАПРИМЕР



# Виды моделей

---

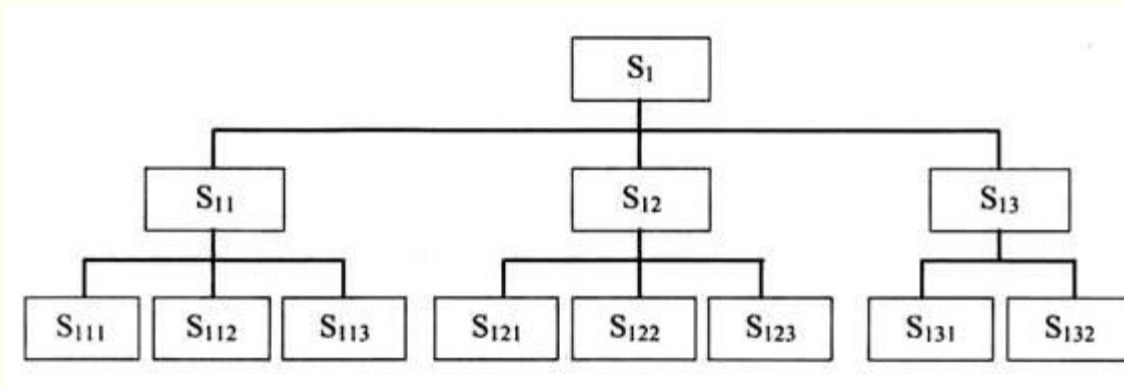
Модели данных делятся на:

1. Иерархическую
2. Сетевую
3. Реляционную
4. Объектно-ориентированную
5. Объектно-реляционную

# Иерархическая модель

---

- **Иерархические модели СУБД** имеют **древовидную** структуру. При этом каждому узлу структуры соответствует один **сегмент**, представляющий собой поименованный линейный кортеж **полей данных**. Каждому сегменту соответствует **один ВХОДНОЙ** и **несколько ВЫХОДНЫХ** сегментов.



---

---

В иерархической модели данных между записями двух типов выполняется групповое отношение . Владелец группового отношения – родительская запись, а члены группового отношения – дочерние записи.

Корневая запись каждого дерева обязательно должна содержать ключ с уникальным значением. Ключи некоторых записей должны иметь уникальное значение только в рамках группового отношения. Каждая запись идентифицируется полным ключом, под которым понимается совокупность ключей всех записей от корневой до искомой по иерархическому пути.

Иерархическая модель данных позволяет непосредственно реализовывать связи со степенями один - ко - многим (1:N) и один - ко - одному (1:1).

# Основные операции

---

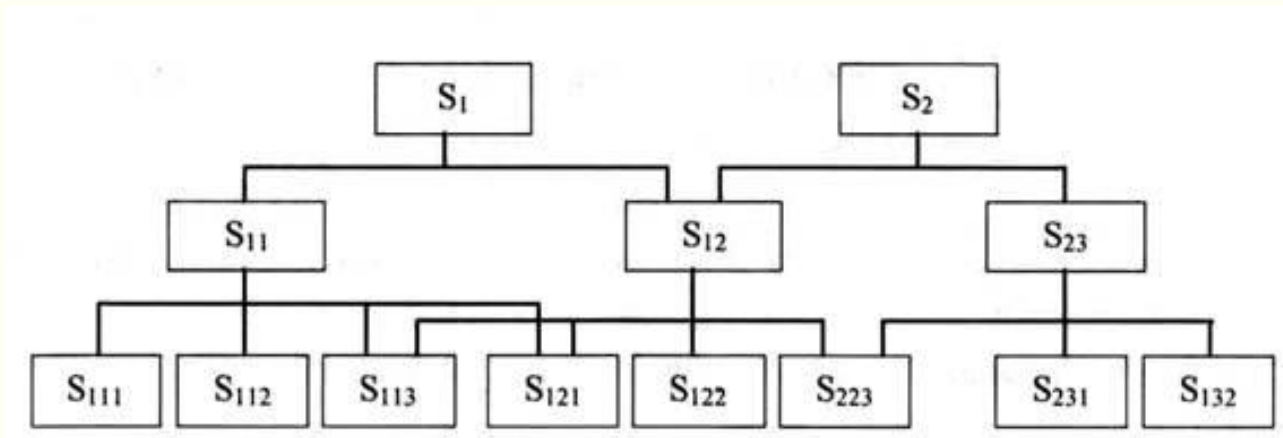
---

- добавлять в базу данных новую запись с обязательным формированием значения ключа для корневой записи
- изменять значения данных, причем ключевые значения данных не должны подвергаться изменениям
- удалять некоторую запись и все подчиненные ей записи
- извлекать ключевую запись по ключевому значению
- извлекать следующую запись (следующая запись извлекается в порядке левостороннего/правостороннего обхода дерева).

# Сетевая модель

---

- **Сетевая модель СУБД** во многом подобна иерархической. **Отличие** заключается в том, что если в иерархической модели для каждого сегмента записи допускается только один входной сегмент при N выходных, то в сетевой модели для сегментов **допускается несколько входных сегментов** наряду с возможностью наличия сегментов без входов с точки зрения иерархической структуры.





---

---

**Сетевая** модель означает представление данных в виде произвольного графа. Достоинством сетевой и иерархической моделей данных является возможность их эффективной реализации по показателям затрат памяти и оперативности. Недостатком сетевой модели данных является высокая сложность и жесткость схемы БД, построенной на ее основе.

Операции над типами записей в сетевой модели данных можно разделить на две основные группы:

- 1. операции с записями в наборе*
- 2 . операции навигации*

# *Операции с записями в наборе*

---

- нахождение конкретной записи в наборе
- создание новой записи
- уничтожение записи
- модифицирование записи.

# *Операции навигации*

---

- переход от предка к потомку по некоторой связи
- переход к следующему потомку в некоторой связи
- переход от потомка к предку по некоторой связи
- включение в связь
- исключение из связи
- перестановка в другую связь

# Реляционная модель

---

- **Реляционной называется СУБД**, в которой средства управления БД поддерживают реляционную модель данных. В такой модели общая структура данных (отношение) может быть представлена в виде таблицы, в которой каждая строка значений (кортеж) соответствует логической записи, а заголовки столбцов являются названиями полей(элементов) записи.
- Данная модель позволяет определять:
  - операции по запоминанию и поиску данных;
  - ограничения, связанные с обеспечением целостности данных.
- Операции запоминания и поиска делятся на две группы:
  - - операции на **множествах** (объединение, пересечение, разность, произведение);
  - - **реляционные** операции (выбрать, спроецировать, соединить, разделить).

# Пример



# Описание

---

---

- Таблица состоит из строк и столбцов и имеет имя, уникальное внутри базы данных, которое именуется, как отношение .

- Строка таблицы носит название кортежа , а столбец – атрибута .

- Количество кортежей называется кардинальным числом , а количество атрибутов – степенью отношения .

Первичный ключ является уникальным идентификатором и представляет собой такой столбец или комбинацию столбцов, что в любой момент времени не существует двух строк, содержащих одинаковое значение в этом столбце или комбинации столбцов.

---

---

Каждый столбец таблицы имеет имя, которое записывается в верхней части таблицы. Оно должно быть **уникальным** в таблице, однако различные таблицы могут иметь столбцы с одинаковыми именами. **Любая таблица должна иметь, по крайней мере, один столбец.** Порядок следования столбцов в таблице определяется порядком следования их имен при ее создании. В отличие от столбцов, строки не имеют имен, порядок их следования в таблице не определен, а количество логически не ограничено.

*Отношения обладают следующими важными свойствами:*

- в них нет одинаковых кортежей
- кортежи не упорядочены сверху вниз
- атрибуты не упорядочены слева на право
- все значения атомарны, т.е. отношения нормализованы.

# Отношения между таблицами

---

Существуют четыре типа отношений между таблицами: **один к одному, один ко многим, много к одному, много ко многим.**

- **Отношение один к одному** означает, что каждая запись в **одной** таблице соответствует **только одной** записи в другой таблице.
- **Отношение один ко многим** означает, что **одна** запись из первой таблицы может быть связана **более чем с одной** записью из другой таблицы
- И 2 другие противоположны данным отношениям.



# Объектно-ориентированная модель

---

---

- **Объектно-ориентированная модель** изначально строилась с учетом ее эволюции и расширения.
- Данная модель появились в начале 90-ых и довольно скоро зарекомендовала себя в ряде важных областей, таких как САПР (система автоматизированного проектирования), промышленность программного обеспечения, финансовая сфера, медицина, телекоммуникации, мультимедиа, управляющие информационные системы. Именно в тех областях требовалось найти адекватные средства хранения больших объемов разнообразных данных.

---

---

Принципиальное отличие реляционных и объектно ориентированных БД заключается в следующем:

- объектно-ориентированная модель данных оказывается более адекватной моделью предметной области реального мира
- объекты вторых можно хранить и использовать непосредственно, не раскладывая их по таблицам
- типы данных определяются разработчиком и не ограничены набором predetermined типов.

# Пример

БИБЛИОТЕКА		
свойство	тип	значение
район	string	Невский
АБОНЕНТ	class	
КАТАЛОГ	class	
ВЫДАЧА	class	
билет	abs	
номер	abs	

ВЫДАЧА		
билет	string	00015
номер	string	02867
дата	string	9 01 97

АБОНЕНТ		
билет	string	00015
имя	string	Васильев
адрес	string	Мира, 3
телефон	string	2461288

КАТАЛОГ		
isbn	string	3 217 00628 5
удк	string	6S1306
название	string	Базы данных на Паскале
автор	string	Дж Ульман
КНИГА	class	

КНИГА		
номер	string	02694
стеллаж	string	7
издание	string	1

# Объектно-реляционная модель

---

- В объектно-реляционных СУБД (ОРСУБД) используется базовая реляционная модель данных. ОРСУБД являются постепенным развитием предшествующих им реляционным СУБД.
- Основная идея объектно-реляционного подхода - это допущение использовать в качестве атрибутов не только простые, атомарные типы данных, но и **составные, абстрактные типы данных**, что противоречит классической концепции реляционных СУБД.

- 
- 
- Их возникновение объясняется тем, что реляционные базы данных хорошо работают со встроенными типами данных и гораздо хуже — с пользовательскими, нестандартными. Когда появляется новый важный тип данных, приходится либо включать его поддержку в СУБД, либо заставлять программиста самостоятельно управлять данными в приложении.
  - Объектно-реляционная СУБД позволяет загружать код, предназначенный для обработки "нетипичных" данных. Таким образом, база данных сохраняет свою табличную структуру, но способ обработки некоторых полей таблиц определяется извне, т.е. программистом.