

# **БАЗЫ ДАННЫХ**

## **часть II**

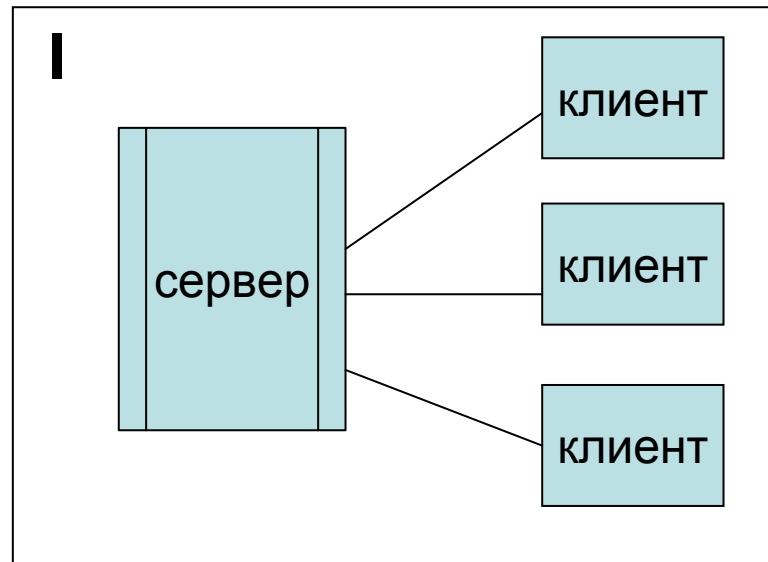
Распределенные и  
параллельные системы  
управления базами данных

# Распределенные и параллельные СУБД

**Возможные архитектуры баз данных**

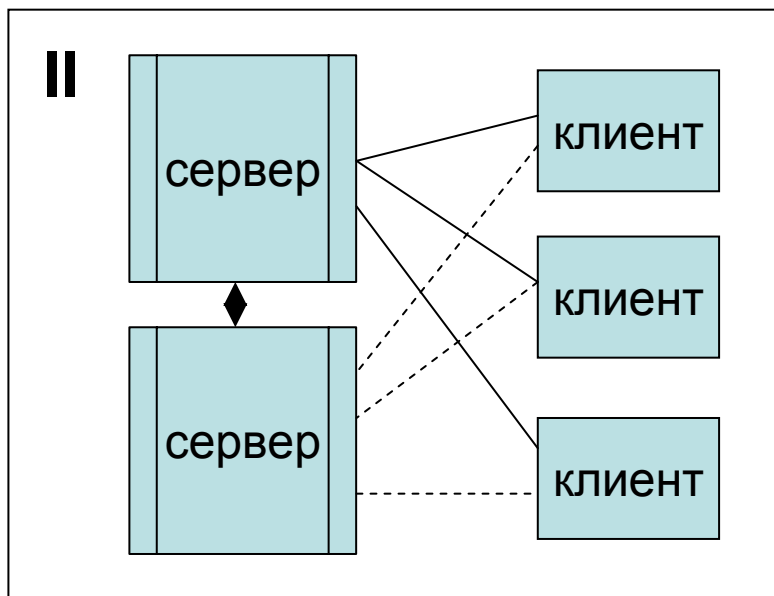
# Распределенные и параллельные СУБД

Наиболее популярна в настоящее время архитектура **клиент-сервер**, когда множество машин-клиентов осуществляют доступ к одному серверу баз данных.



# Распределенные и параллельные СУБД

Архитектура типа *много-клиентов/много-серверов*, когда база данных размещена на множестве серверов, которым, для того чтобы вычислить результат пользовательского запроса или выполнить транзакцию, необходимо взаимодействовать друг с другом.

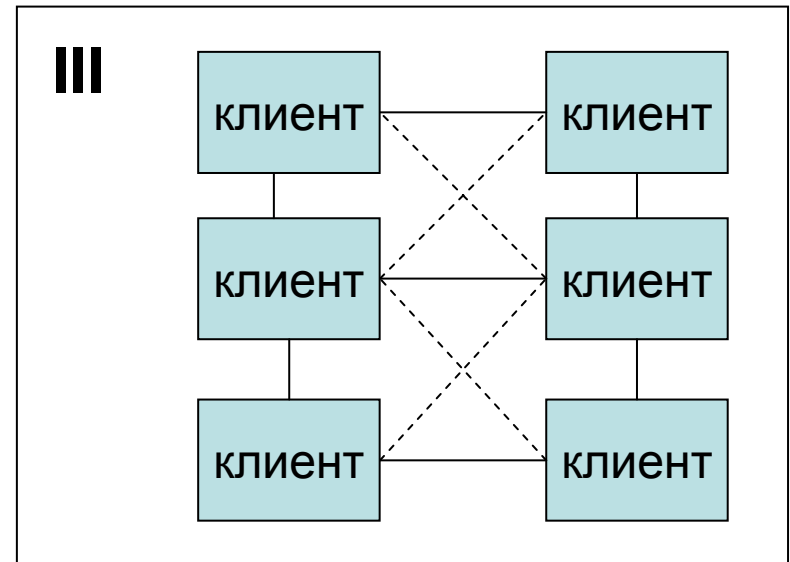


Каждая клиентская машина имеет свой "домашний" сервер; ему она направляет пользовательские запросы. Взаимодействие серверов друг с другом прозрачно для пользователей.

# Распределенные и параллельные СУБД

В истинно распределенной СУБД клиентские и серверные машины не различаются. В идеале каждый узел может выступать и как клиент, и как сервер.

Такие архитектуры, тип которых определяют как *равный-к-равному*, требуют сложных протоколов управления данными, распределенными по множеству узлов.



# Распределенные и параллельные СУБД

Архитектуры параллельных систем варьируются между двумя крайними точками, называемыми **архитектура без разделяемых ресурсов** и **архитектура с разделяемой памятью**. Промежуточную позицию занимает архитектура с разделяемыми дисками.

# Распределенные и параллельные СУБД

В случае **неразделения ресурсов** каждый процессор имеет эксклюзивный доступ к собственной оперативной памяти и к набору дисков.

Архитектуры без разделяемых ресурсов обладают тремя важнейшими **преимуществами**: низкие затраты, расширяемость, высокая доступность

Наиболее существенные **проблемы** - сложность реализации и (потенциальные) трудности соблюдения баланса загрузки.

# Распределенные и параллельные СУБД

Подход, основанный на разделении памяти, заключается в том, что каждый процессор посредством быстрых линий связи (высокоскоростных шин или кросс-панельных коммутаторов) соединен со всеми модулями памяти и дисковыми устройствами.

**Сильные стороны** - простота и хороший баланс загрузки.

Наиболее существенные **проблемы** - стоимость, ограниченная расширяемость, невысокая надежность.



# Распределенные и параллельные СУБД

В системах с разделяемыми дисками каждый процессор имеет доступ к любому дисковому устройству посредством специальных соединений и эксклюзивный доступ к своей собственной оперативной памяти.

**Преимущества:** низкие затраты, расширяемость, хороший баланс загрузки, высокая доступность, простота миграции с однопроцессорных систем.

**Трудности:** сложность системы, потенциальные проблемы производительности.

## **Обработка и оптимизация запросов**

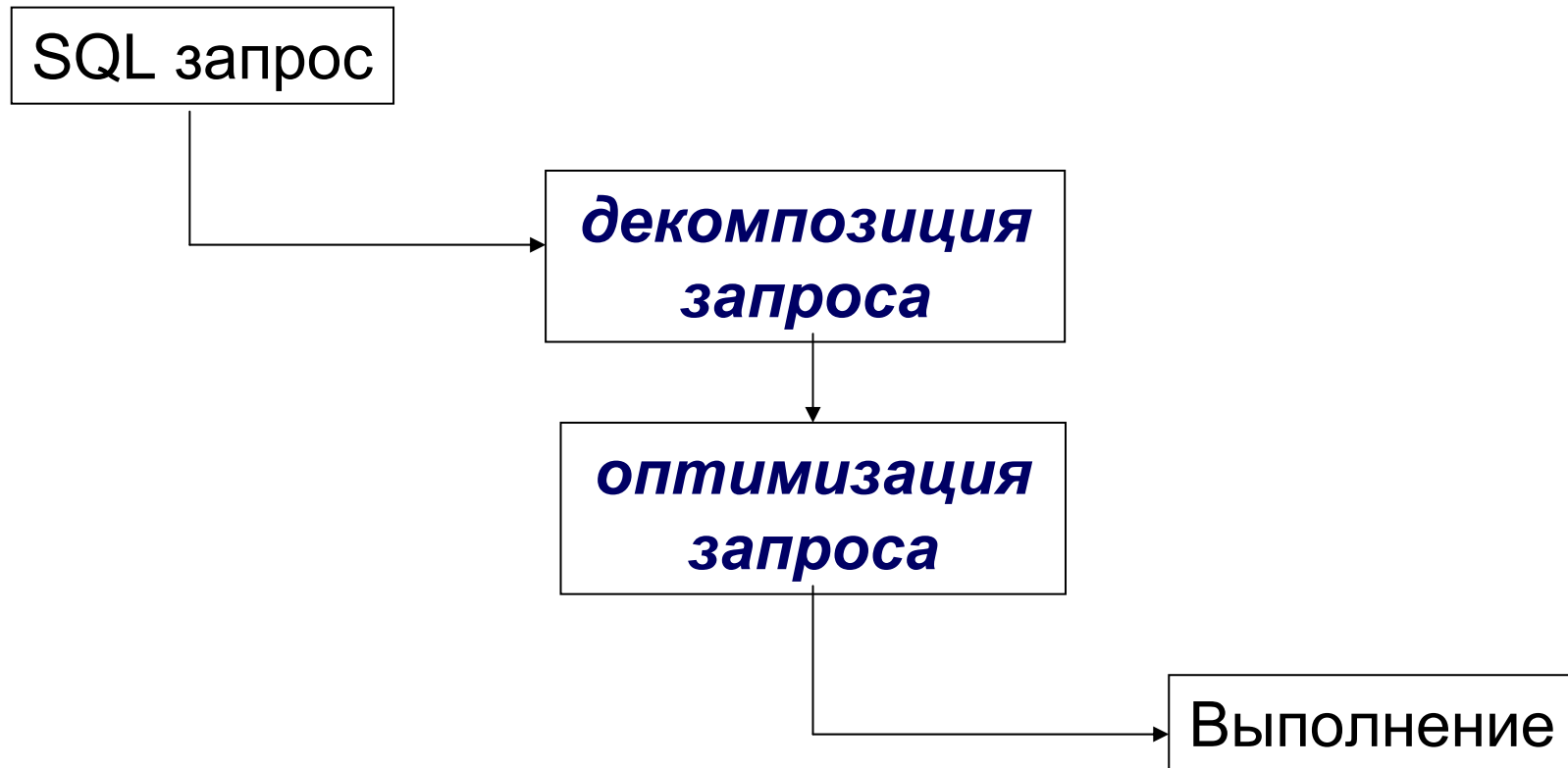
# Распределенные и параллельные СУБД

**Обработка запроса** - это процесс трансляции декларативного определения запроса в операции манипулирования данными низкого уровня. Стандартным языком запросов, поддерживаемым современными СУБД, является SQL.

**Оптимизация запроса** - это процедура выбора "наилучшей" стратегии для реализации запроса из множества альтернатив.

# Распределенные и параллельные СУБД

Для централизованной СУБД весь процесс состоит обычно из двух шагов:



# Распределенные и параллельные СУБД

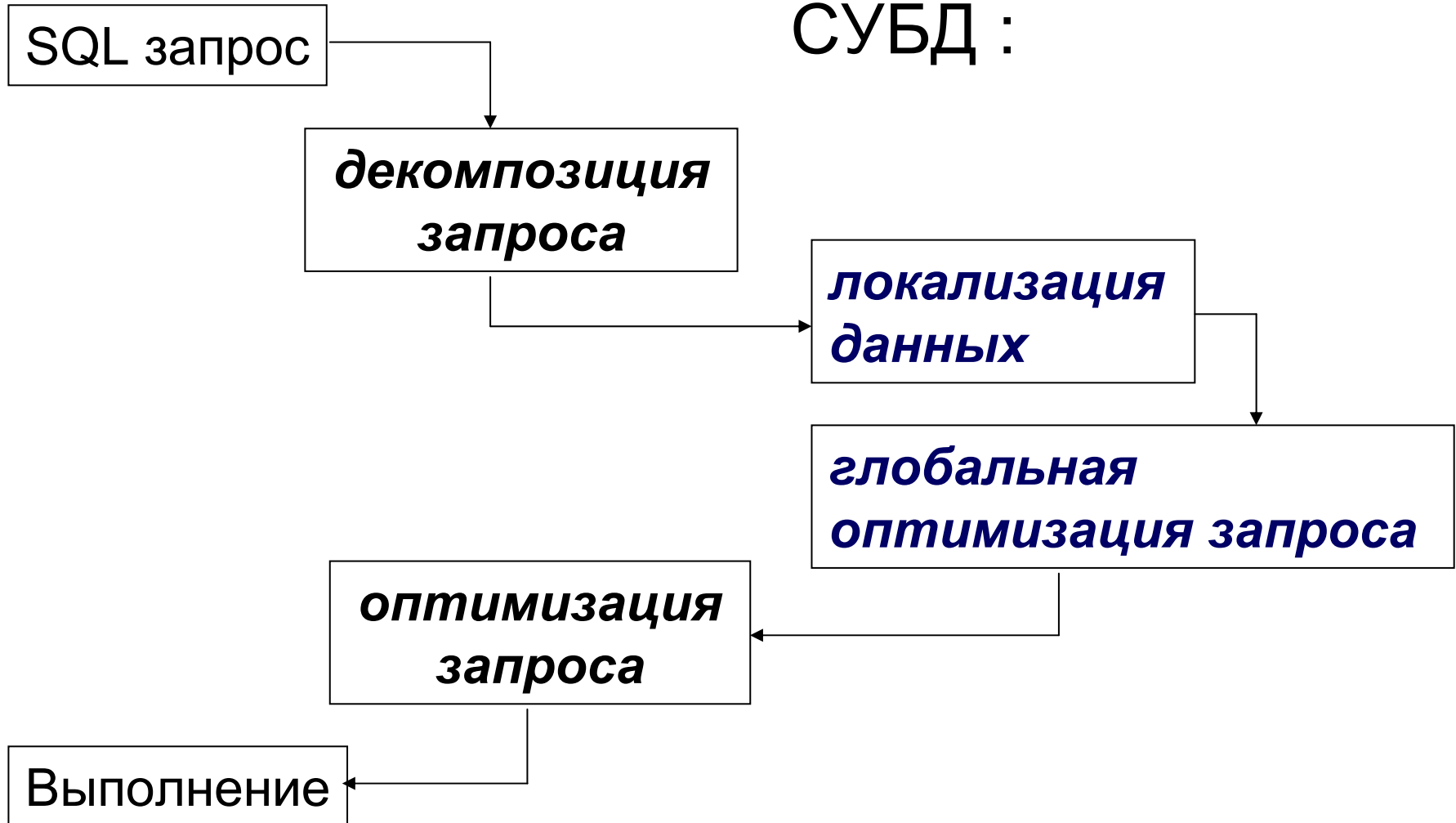
***Декомпозиция запроса*** - это трансляция его с языка SQL в выражение реляционной алгебры. В ходе декомпозиции запрос подвергается семантическому анализу; при этом некорректные запросы отвергаются, а корректные упрощаются. Упрощенный запрос преобразуется в алгебраическую форму.

# Распределенные и параллельные СУБД

"Качество" алгебраического выражения определяется исходя из объема затрат, необходимых для его вычисления. SQL-запрос транслируется в какое-нибудь выражение, а затем, применяя правила эквивалентных алгебраических преобразований, получают из него другие алгебраические преобразования, пока не будет найдено "наилучшее". При поиске "наилучшего" выражения используется функция стоимости, в соответствии с которой вычисляется сумма затрат, необходимых для выполнения запроса. Этот процесс и называется ***оптимизацией запросов***.

# Распределенные и параллельные СУБД

В распределенной СУБД :



# Распределенные и параллельные СУБД

Исходной информацией для локализации данных служит алгебраическое выражение, полученное на этапе декомпозиции запроса. В этом выражении фигурируют глобальные отношения без учета их фрагментации или распределения. Сущность данного шага заключается в том, чтобы локализовать участвующие в запросе данные, используя информацию об их распределении. При этом выявляются фрагменты, реально участвующие в запросе, а запрос преобразуется к форме, где операции применяются уже не к глобальным отношениям, а к фрагментам.



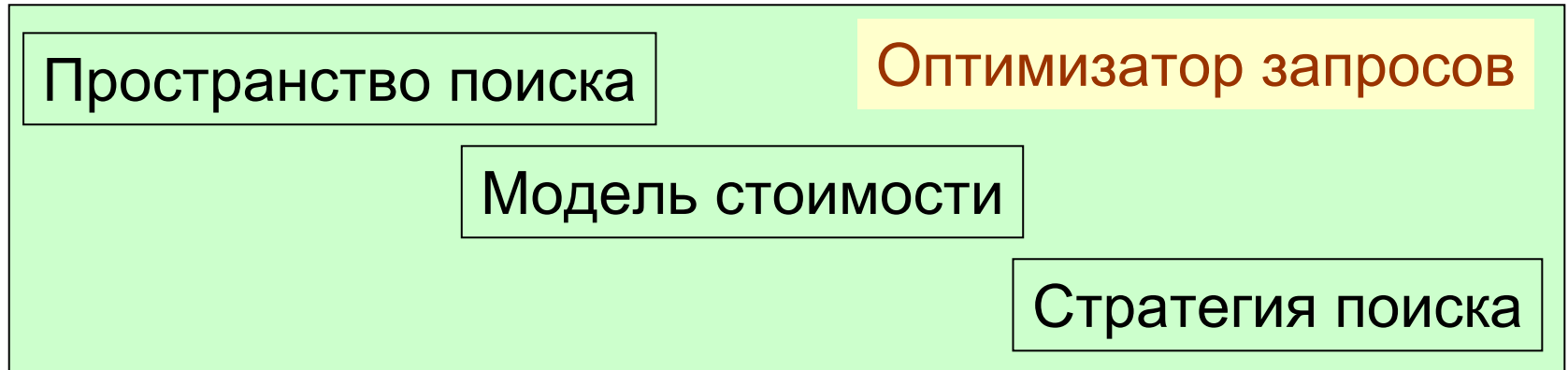
# Распределенные и параллельные СУБД

Распределенные отношения реконструируются путем применения инверсии правил фрагментации. Это называется **программой локализации**. Таким образом, на этапе локализации данных запрос заменяется программой локализации; фрагментный запрос затем упрощается и реструктурируется, пока не будет получено "хорошее" выражение. Как и в шаге декомпозиции, окончательный "хороший" фрагментный запрос может быть еще далек от оптимального; данный процесс лишь исключает "плохие" алгебраические выражения.

# Распределенные и параллельные СУБД

Цель **глобальной оптимизации** - найти стратегию выполнения запроса, близкую к оптимальной. Стратегию выполнения распределенного запроса можно выразить в терминах *операций реляционной алгебры* и *коммуникационных примитивов* (операций "послать"/"получить"), описывающих пересылки данных между узлами. Однако проведенные оптимизации на предыдущих этапах не зависели от характеристик фрагментов; еще не учтены коммуникационные операции. Путем перестановок операций в рамках фрагментного запроса можно получить множество эквивалентных планов его выполнения. Оптимизация запроса заключается в

# Распределенные и параллельные СУБД



**Пространство поиска** - это множество альтернативных планов выполнения исходного запроса.

**Модель стоимости** - это способ оценить стоимость реализации заданного плана.

**Стратегия поиска** - это способ обхода пространства поиска и выбора наилучшего плана.

# Распределенные и параллельные СУБД

В распределенной среде **функция стоимости**, часто определяемая в единицах времени, оценивает затраты вычислительных ресурсов, таких как дисковое пространство, число обменов с дисками, время CPU, коммуникации и т. д. Обычно это некоторая взвешенная сумма затрат ввода-вывода, CPU и коммуникаций.

В распределенных СУБД применяется упрощенный подход, когда в качестве наиболее значимых рассматриваются лишь коммуникационные затраты.

# Распределенные и параллельные СУБД

Внутриоперационный параллелизм достигается за счет выполнения операции сразу на нескольких узлах многопроцессорной машины.

Множество узлов, на которых хранится отношение, называется **домашним множеством**.

**Домашним множеством узлов операции** называется множество узлов, на которых она выполняется; оно должно совпадать с домашними множествами узлов ее операндов, для того чтобы операция имела доступ к своим исходным данным.

# Распределенные и параллельные СУБД

*Межоперационный параллелизм* имеет место, когда одновременно выполняются две или более операции, независимые или связанные общим потоком данных. Термином **поток данных** мы обозначаем форму параллелизма, реализуемую методами *конвейерной обработки*.

При **независимом** параллелизме операции выполняются одновременно или в произвольном порядке. Независимый параллелизм возможен, только если операции не содержат в качестве операндов общих данных.

**ВОПРОСЫ ?**

## **СОВЕТ ДНЯ:**

**Если с утра уже третий раз не охота вставать – значит пришла среда!**

**Если в это утро еще и не хочется есть – значит опаздываешь на пару!**

**Позаботься о допуске во вторник!**