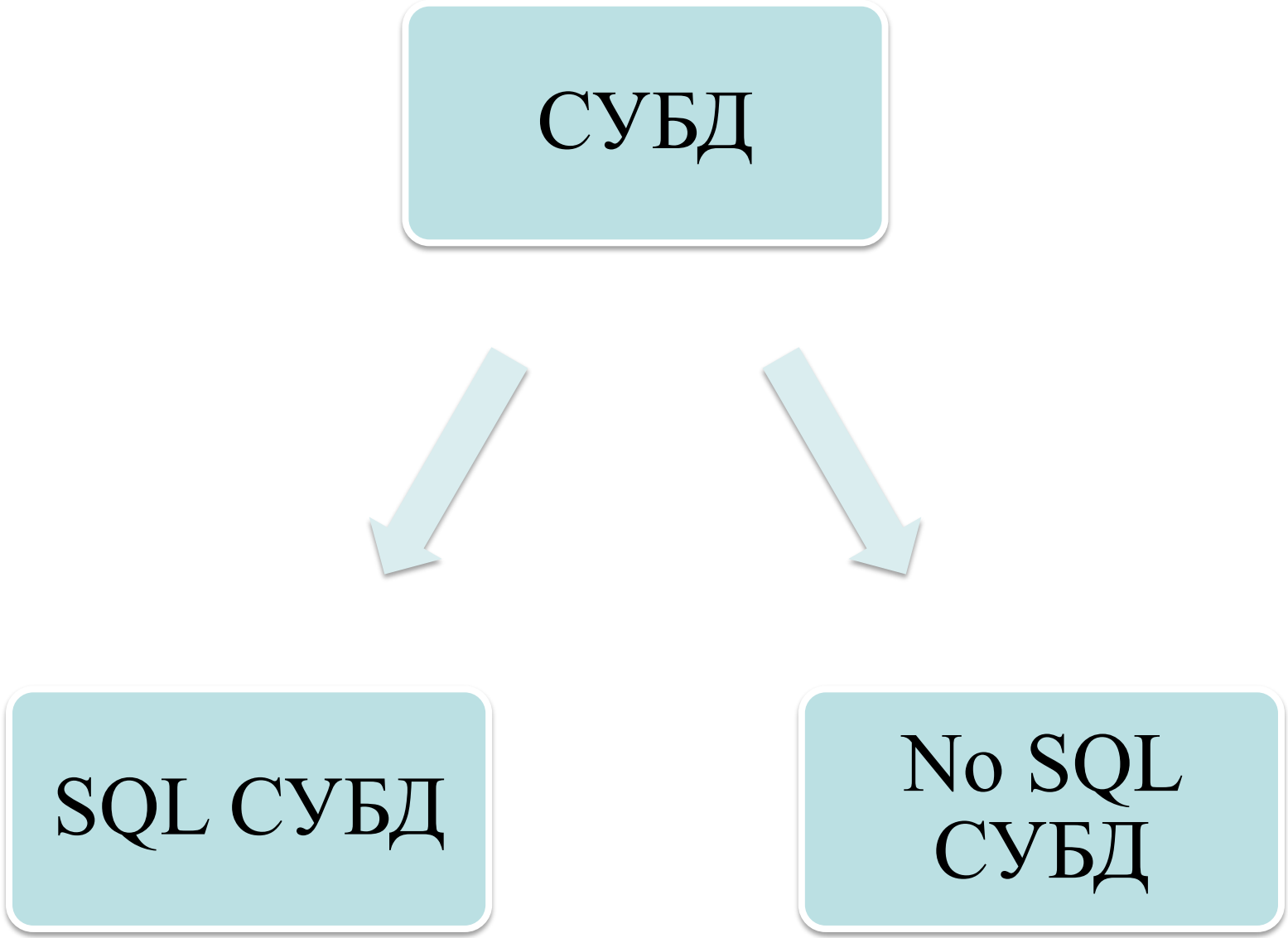
Методы и системы обработки больших данных

1. Введение

Современные типы систем управления базами данных (СУБД)

Тип СУБД	Специфика применения	Примеры
Реляционные (SQL)	Нужна транзакционность; высокая нормализация; большая доля операций на вставку	Oracle, MySQL, Microsoft SQL Server, PostgreSQL
Ключ-значение	Задачи кэширования и брокеры сообщений	Redis, Memcached
Документные	Для хранения объектов в одной сущности, но с разной структурой; хранение структур на основе JSON	CouchDB, MongoDB, Amazon DocumentDB
Графовые	Задачи подобные социальным сетям; системы оценок и рекомендаций	Neo4j, Amazon Neptune, InfiniteGraph, InfoGrid
Колоночные	Хранилища данных; выборки со сложными аналитическими вычислениями; количество строк в таблице превышает сотни миллионов	Vertica, ClickHouse, Google BigTable, Sybase \ SAP IQ, InfoBright, Cassandra

Классификация СУБД



Статистика

- Практически все разработчики современных приложений, предусматривающих связь с системами баз данных, ориентируются на SQL СУБД
- По данным аналитиков реляционные (SQL) СУБД используются в абсолютном большинстве крупных проектов по разработке информационных систем
- По результатам исследований компании IDC всего около 7 % составляют проекты, в которых используются No SQL СУБД

Реляционная (SQL) СУБД

- База данных представляется в виде совокупности плоских (двумерных) таблиц-отношений с высокой степенью нормализации
- Нормализация: каждой сущности соответствует отдельная таблица; каждое свойство сущности представлено в виде отдельного атрибута (столбца таблицы)
- Связи между таблицами реализуются на логическом уровне с помощью первичных и внешних ключей
- Стандартным языком манипулирования данными является SQL

История SQL

- В 1969 году английский специалист по информатике Эдгар Ф. Кодд разработал реляционную модель, в которой все данные представлены в терминах кортежей, сгруппированных в отношения
- В 1975 IBM создала первую реляционную СУБД System R и язык манипулирования данными SEQUEL (Structured English QUEry Language)
- В 1980 по юридическим соображениям язык SEQUEL был переименован в SQL (Structured Query Language)
- В 1986 г. вышел первый стандарт языка SQL
- В 2016 г. вышел девятый стандарт языка SQL

Эволюция SQL

Год	Стандарт	Изменения
1986	SQL-86	Первый вариант стандарта, принятый институтом ANSI и одобренный ISO в 1987 году.
1989	SQL-89	Немного доработанный вариант предыдущего стандарта.
1992	SQL-92	Значительные изменения (ISO 9075); уровень *Entry Level* стандарта SQL-92 был принят как стандарт FIPS 127-2
1999	SQL:1999	Добавлена поддержка регулярных выражений, рекурсивных запросов, поддержка триггеров, базовые процедурные расширения, нескалярные типы данных и некоторые объектно-ориентированные возможности.
2003	SQL:2003	Введены расширения для работы с XML-данными, оконные функции (применяемые для работы с OLAP-базами данных), генераторы последовательностей и основанные на них типы данных.
2006	SQL:2006	Функциональность работы с XML-данными значительно расширена. Появилась возможность совместно использовать в запросах SQL и Xquery.
2008	SQL:2008	Улучшены возможности оконных функций, устранены некоторые неоднозначности стандарта SQL:2003.
2011	SQL:2011	Реализована поддержка хронологических баз данных (PERIOD FOR), поддержка конструкции FETCH.
2016	SQL:2016	Защита на уровне строк, полиморфные табличные функции, JSON.

Эволюция СУБД

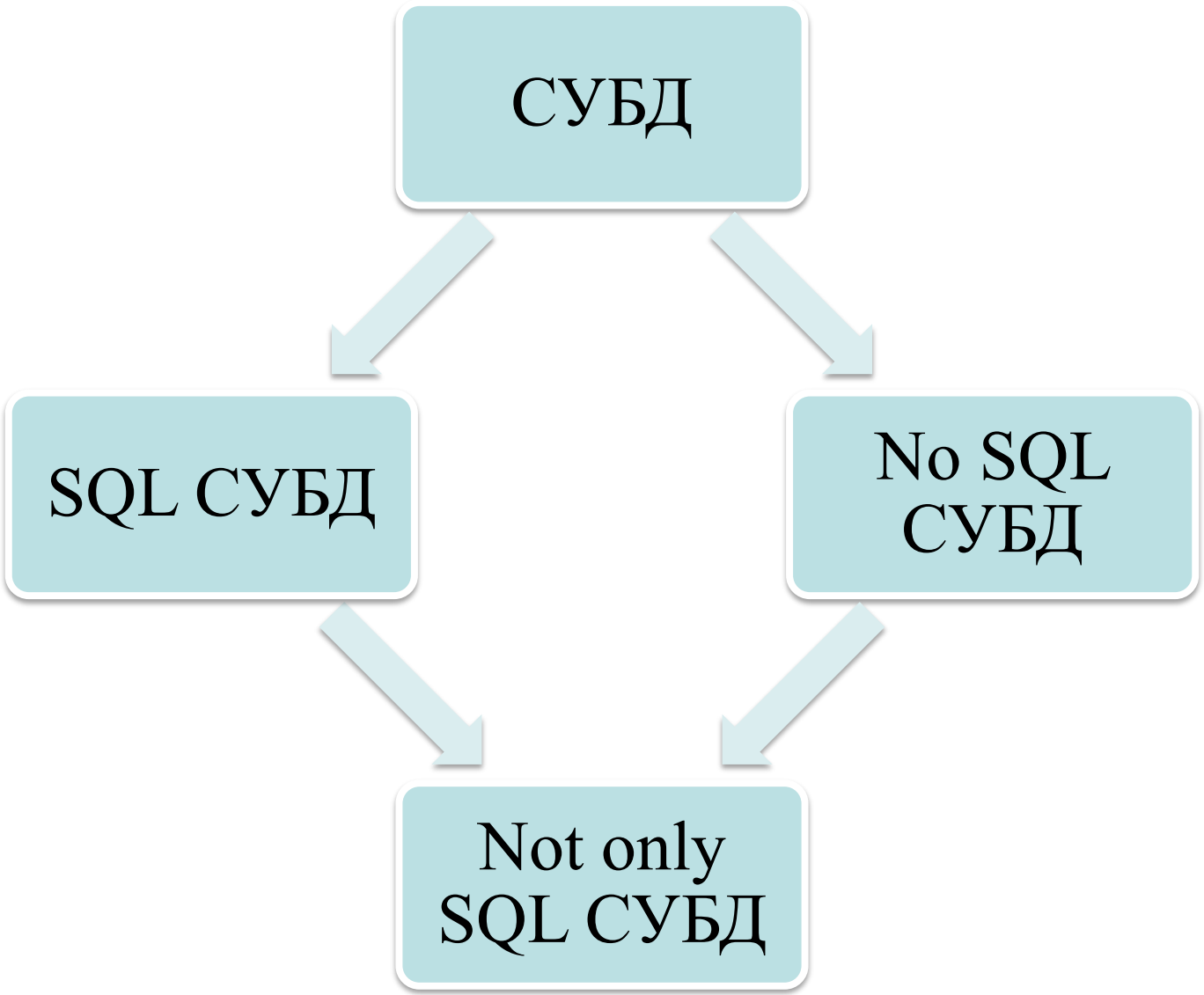


Схема базы данных «Поставки»

Отношения (таблицы)	Семантика
S(IDS, NameS, CityS, Rating, Deposit)	Информация о поставщиках
P(IDP, NameP, CityP, Color)	Информация о деталях
SP(IDS, IDP, Amount, Price)	Поставки деталей

S (Suppliers)

Отношение S содержит информацию о поставщиках

Атрибут	Семантика
IDS*	Код поставщика
NameS	Имя поставщика
CityS	Город регистрации поставщика
Rating	Рейтинг поставщика
Deposit	Страховой депозит

* Первичный ключ

P (Parts)

Отношение P содержит информацию о деталях

Атрибут	Семантика
IDP*	Код детали
NameP	Ия детали
CityP	Город производства детали
Color	Цвет детали

* Первичный ключ

SP (Supplies of Parts)

Отношение SP содержит информацию о поставках деталей

Атрибут	Семантика
IDS*	Код поставщика
IDP*	Код детали
Amount	Количество штук в поставке
Price	Цена за штуку

* Первичный ключ

Пример базы данных

IDS	NameS	CityS	Rating	Deposit
1	Petrov	Moscow	100	1000
2	Smith	New-York	60	2000
3	Sidorov	Moscow	80	1000
4	Abramov	Perm	50	500

S (Suppliers)

IDS	IDP	Amount	Price
1	1	2000	100
1	2	3000	100
2	3	2500	200
4	4	500	150

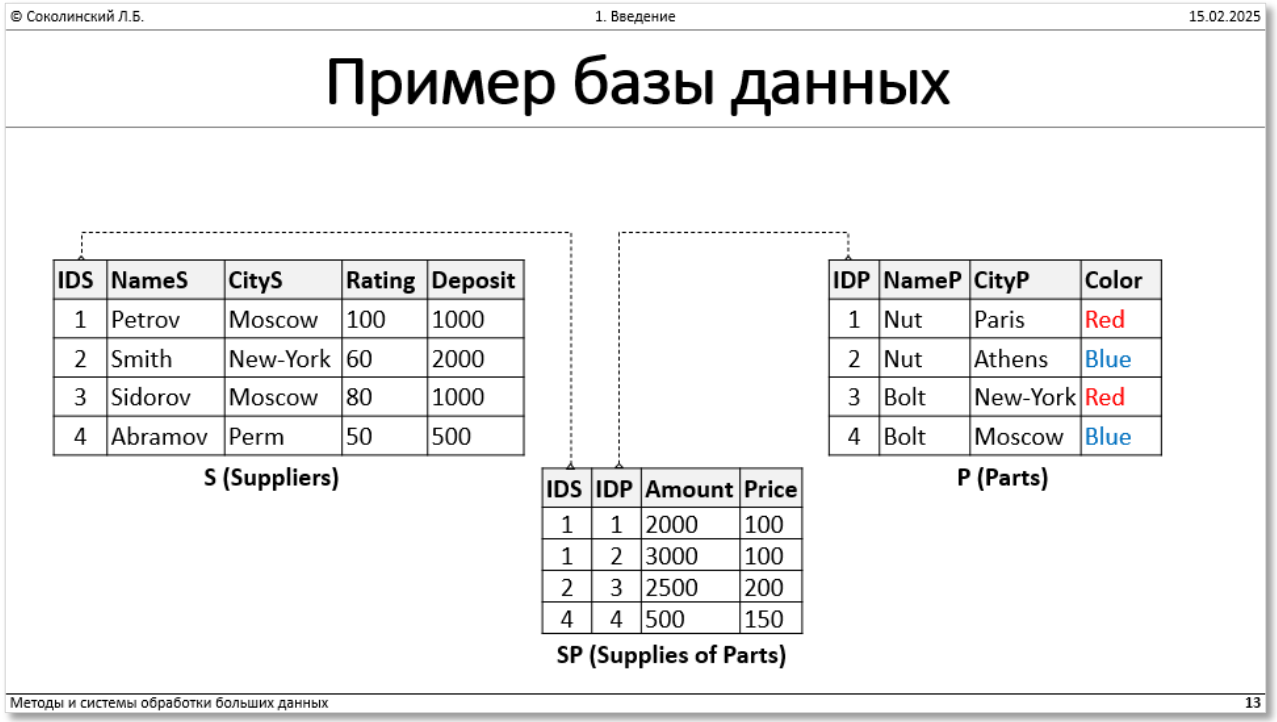
SP (Supplies of Parts)

IDP	NameP	CityP	Color
1	Nut	Paris	Red
2	Nut	Athens	Blue
3	Bolt	New-York	Red
4	Bolt	Moscow	Blue

P (Parts)

Пример запроса на языке SQL

Имена поставщиков, поставляющих хотя бы одну красную деталь.



IDS	NameS	CityS	Rating	Deposit
1	Petrov	Moscow	100	1000
2	Smith	New-York	60	2000
3	Sidorov	Moscow	80	1000
4	Abramov	Perm	50	500

S (Suppliers)

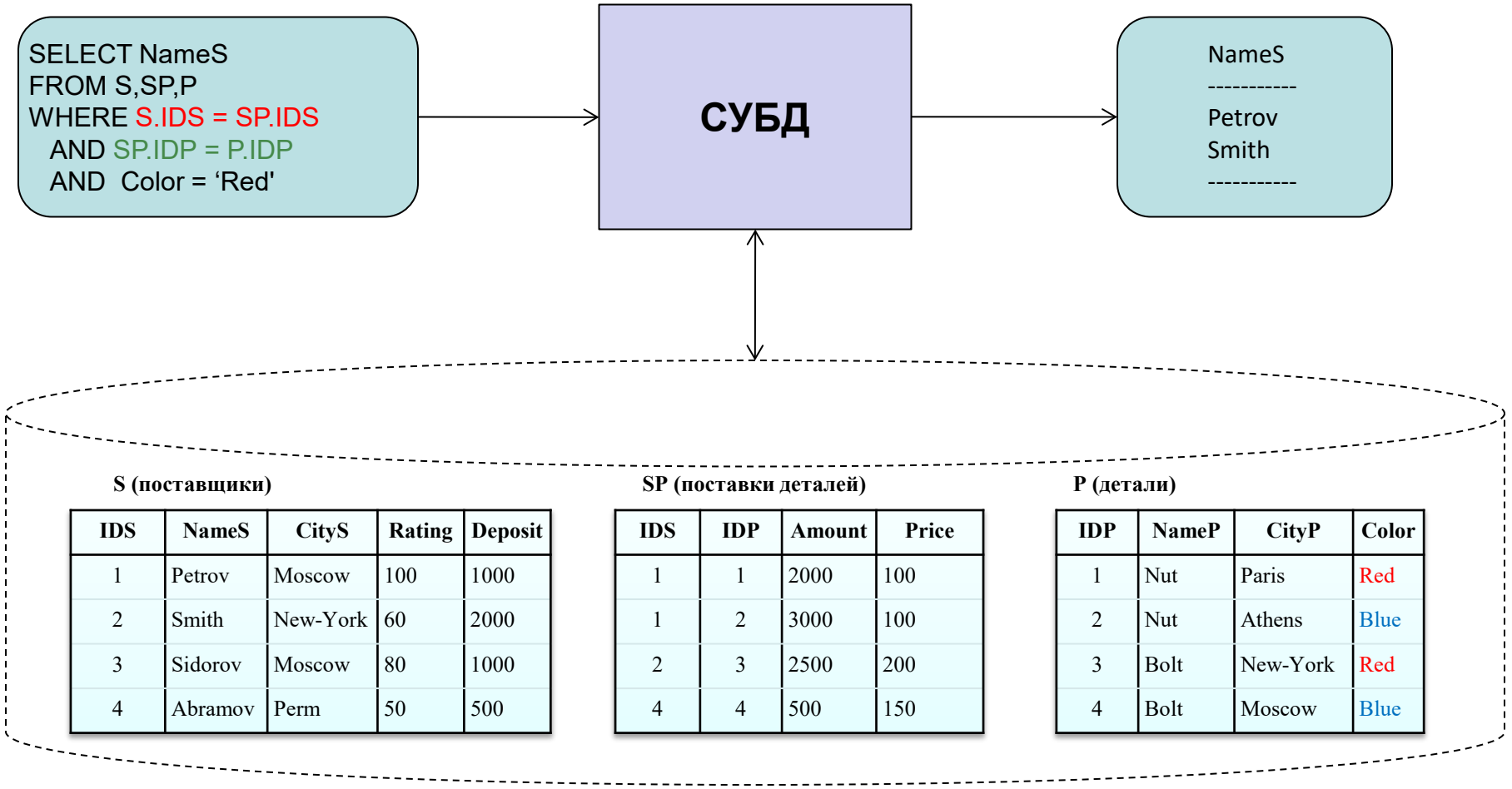
IDP	NameP	CityP	Color
1	Nut	Paris	Red
2	Nut	Athens	Blue
3	Bolt	New-York	Red
4	Bolt	Moscow	Blue

P (Parts)

IDS	IDP	Amount	Price
1	1	2000	100
1	2	3000	100
2	3	2500	200
4	4	500	150

SP (Supplies of Parts)

Как СУБД получает ответ?



S (поставщики)

IDS	NameS	CityS	Rating	Deposit
1	Petrov	Moscow	100	1000
2	Smith	New-York	60	2000
3	Sidorov	Moscow	80	1000
4	Abramov	Perm	50	500

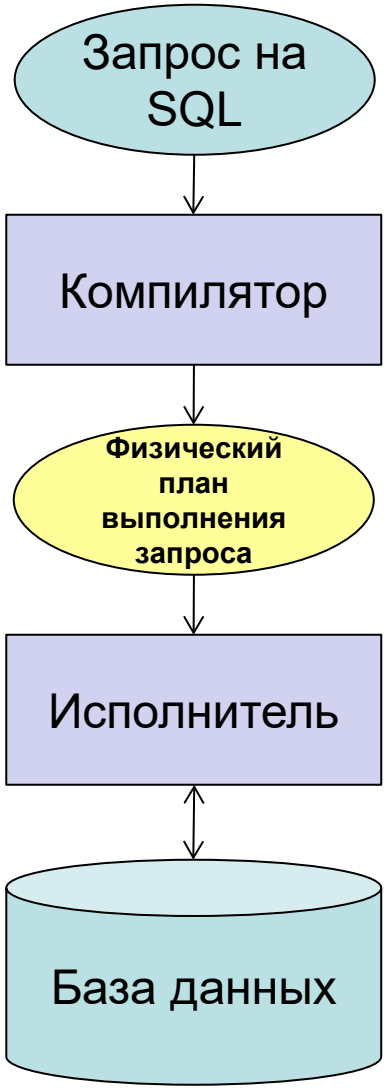
SP (поставки деталей)

IDS	IDP	Amount	Price
1	1	2000	100
1	2	3000	100
2	3	2500	200
4	4	500	150

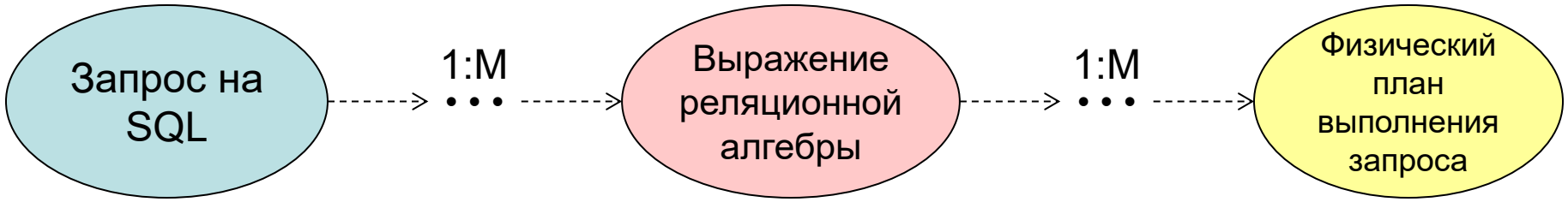
P (детали)

IDP	NameP	CityP	Color
1	Nut	Paris	Red
2	Nut	Athens	Blue
3	Bolt	New-York	Red
4	Bolt	Moscow	Blue

Схема обработки запроса



На пути к плану выполнения запроса



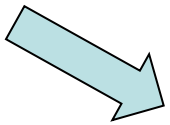
Основные операции реляционной алгебры

Операция	Название	Семантика
$\pi_{A,C}(R)$	Проекция	Вертикальная селекция (столбцов).
$\sigma_{\theta}(R)$	Выборка	Горизонтальная селекция (строк) по условию θ.
$R \times S$	Прямое произведение	Сочетание пар «каждый с каждым».
$R \bowtie S$	Естественное соединение	Соединение пар по общим атрибутам.
$R \underset{\theta}{\bowtie} S$	Тета-соединение	Соединение пар по условию θ.
$\delta(R)$	Удаление дубликатов	Мультимножество кортежей R преобразуется в множество кортежей
$\gamma_{B, AVG(C) \rightarrow X}(R)$	Группировка	Результирующая таблица имеет схему (B^*,X). Для всех записей из R, имеющих одинаковое значение B, вычисляется среднее значение C и помещается в столбец X.
$\gamma_{SUM(B)}(R)$	Агрегирование	Вычисляет сумму всех значений в столбце B.

Проекция (вырезание столбцов)

R

A*	B	C
1	20	100
2	40	300
3	20	100
4	10	300



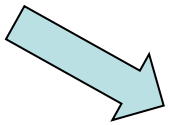
$\pi_{A,C}(R)$

A*	C
1	100
2	300
3	100
4	300

Выборка (вырезание строк)

R

A*	B	C
1	20	100
2	40	300
3	20	100
4	10	300



$\sigma_{C<150}(R)$

A*	B	C
1	20	100
3	20	100

Прямое произведение

R

A*	B	C
1	20	100
2	40	300
3	20	100
4	10	300

S

D*	A#	E
1	3	0.2
2	1	0.5
3	1	0.5

R×S

R.A*	B	C	D*	S.A	E
1	20	100	1	3	0.2
1	20	100	2	1	0.5
1	20	100	3	1	0.5
2	40	300	1	3	0.2
2	40	300	2	1	0.5
2	40	300	3	1	0.5
3	20	100	1	3	0.2
3	20	100	2	1	0.5
3	20	100	3	1	0.5
4	10	300	1	3	0.2
4	10	300	2	1	0.5
4	10	300	3	1	0.5

Тета-соединение $R \underset{\theta}{\bowtie} S$

R

A*	B	C
1	20	600
2	40	300
3	20	250
4	10	300

S

D*	A#	E
1	3	0.2
2	1	0.7
3	1	0.5

$$R \underset{R.A = S.A \wedge C>E*1000}{\bowtie} S = \sigma_{R.A = S.A \wedge C>E*1000}(R \times S)$$

R.A*	B	C	D*	S.A	E
1	20	600	3	1	0.5
3	20	250	1	3	0.2

Естественное соединение $R \bowtie S$

R

A*	B	C
1	20	100
2	40	300
3	20	100
4	10	300

S

D*	A#	E
1	**3**	0.2
2	**1**	0.5
3	**1**	0.5

$$R \bowtie S = \pi_{*\setminus S.A}\left(R \underset{R.A=S.A}{\bowtie} S\right)$$

R⋈S

A*	B	C	D*	E
1	20	100	2	0.5
1	20	100	3	0.5
3	20	100	1	0.2

$R \underset{R.A=S.A}{\bowtie} S$

R.A*	B	C	D*	S.A	E
1	20	100	2	**1**	0.5
1	20	100	3	**1**	0.5
3	20	100	1	**3**	0.2

Возникновение дубликатов

R

A*	B	C
1	20	100
2	40	300
3	20	100
4	10	300

$\pi_{B,C}(R)$

B	C
20	100
40	300
20	100
10	300

Удаление дубликатов

R

B	C
20	100
40	300
20	100
10	300

δ(R)

B*	C
20	100
40	300
10	300

Группировка с вычислением среднего значения в группе

R

A*	B	C
1	**20**	400
2	**40**	300
3	**20**	150
4	**40**	500
5	**20**	50

A*	B	C
1	**20**	400
3	**20**	150
5	**20**	50
2	**40**	300
4	**40**	500

A*	B	C	X=AVG(C)
1	**20**	400	200
3	**20**	150	
5	**20**	50	
2	**40**	300	400
4	**40**	500	

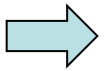
$\gamma_{B,\ AVG(C)\rightarrow X}(R)$

B*	X
20	200
40	400

Агрегирование

R

A*	B	C
1	2	400
2	4	300
3	2	150
4	4	500
5	2	50



$\gamma_{SUM(B*C)}(R)$

4400

Агрегирующие функции

Функция	Семантика
SUM	Сумма значений
AVG	Среднее значение
MIN	Минимальное значение
MAX	Максимальное значение
COUNT	Количество кортежей

Операции реляционной алгебры в SQL

Операция	SQL
$\pi_{A,C}(R)$	**SELECT** A,C **FROM** R
$\sigma_{\theta}(R)$	**SELECT** * **FROM** R **WHERE** θ
$R \times S$	R **CROSS JOIN** S
$R \bowtie S$	R **NATURAL JOIN** S
$R \underset{\theta}{\bowtie} S$	R **JOIN** S **ON** θ
$\delta(R)$	**SELECT DISTINCT** * **FROM** R
$\gamma_{B,AVG(C)\rightarrow X}(R)$	**SELECT** B, AVG(C) **AS** X **FROM** R **GROUP BY** B

Построение реляционного выражения

```
/* Имена поставщиков, поставляющих
хотя бы одну красную деталь */
SELECT NameS
FROM S,SP,P
WHERE S.IDS = SP.IDS
   AND SP.IDP = P.IDP
   AND  Color = 'Red'
```

Построение реляционного выражения

$$\pi_{NameS}(\sigma_{S.IDS=SP.IDS\ \&\ SP.IDP=P.IDP\ \&\ Color='Red'}(\ S\times(SP\times P)))))$$

/* Имена поставщиков, поставляющих хотя бы одну красную деталь */

```
SELECT NameS
FROM S,SP,P
WHERE S.IDS = SP.IDS
   AND SP.IDP = P.IDP
   AND  Color = 'Red'
```

Построение реляционного выражения

$$\pi_{NameS}(\sigma_{S.IDS=SP.IDS\ \&\ SP.IDP=P.IDP\ \&\ Color='Red'}(\ S\times(SP\times P)))))$$

$$\pi_{NameS}(\sigma_{Color='Red'}(S\bowtie(SP\bowtie P)))$$

/* Имена поставщиков, поставляющих хотя бы одну красную деталь */

```
SELECT NameS
FROM S,SP,P
WHERE S.IDS = SP.IDS
   AND SP.IDP = P.IDP
   AND  Color = 'Red'
```

Построение реляционного выражения

$$\pi_{NameS}(\sigma_{S.IDS=SP.IDS\ \&\ SP.IDP=P.IDP\ \&\ Color='Red'}(\ S\times(SP\times P)))))$$

$$\pi_{NameS}(\sigma_{Color='Red'}(S\bowtie(SP\bowtie P)))$$

$$\pi_{NameS}(\sigma_{Color='Red'}((S\bowtie SP)\bowtie P))$$

/* Имена поставщиков, поставляющих хотя бы одну красную деталь */

```
SELECT NameS
FROM S,SP,P
WHERE S.IDS = SP.IDS
   AND SP.IDP = P.IDP
   AND  Color = 'Red'
```

Построение реляционного выражения

$\pi_{NameS}(\sigma_{S.IDS=SP.IDS\ \&\ SP.IDP=P.IDP\ \&\ Color='Red'}(\ S\times(SP\times P)))))$

$\pi_{NameS}(\sigma_{Color='Red'}(S\bowtie(SP\bowtie P)))$

$\pi_{NameS}(\sigma_{Color='Red'}((S\bowtie SP)\bowtie P))$

$\pi_{NameS}((S\bowtie SP)\bowtie\sigma_{Color='Red'}(P))$

/* Имена поставщиков, поставляющих хотя бы одну красную деталь */

```
SELECT NameS
FROM S,SP,P
WHERE S.IDS = SP.IDS
   AND SP.IDP = P.IDP
   AND  Color = 'Red'
```

Построение реляционного выражения

/* Имена поставщиков, поставляющих хотя бы одну красную деталь */

```
SELECT NameS
FROM S,SP,P
WHERE S.IDS = SP.IDS
  AND SP.IDP = P.IDP
  AND  Color = 'Red'
```

$$\pi_{NameS}(\sigma_{S.IDS=SP.IDS\ \&\ SP.IDP=P.IDP\ \&\ Color='Red'}(S\times(SP\times P)))$$

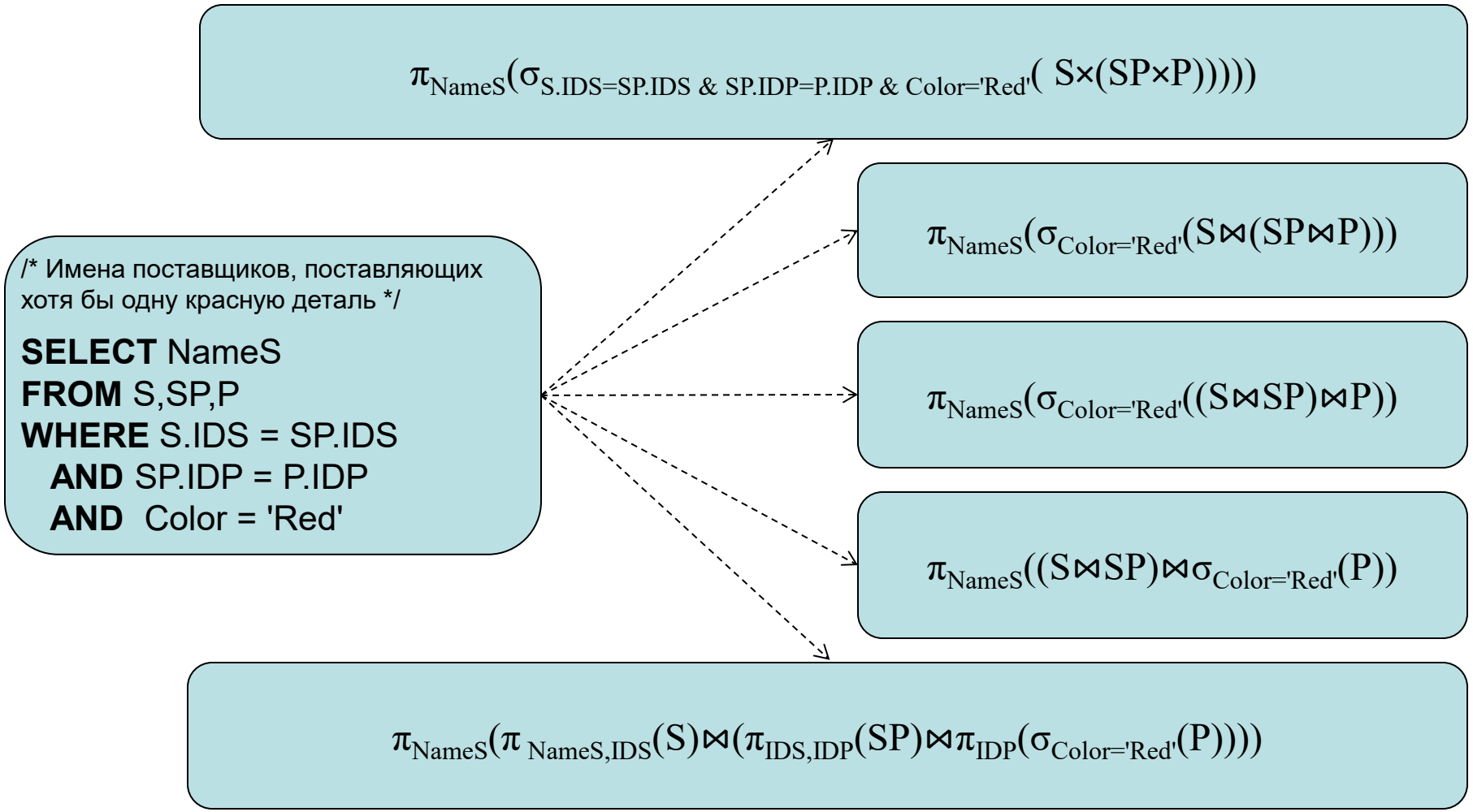
$$\pi_{NameS}(\sigma_{Color='Red'}(S\bowtie(SP\bowtie P)))$$

$$\pi_{NameS}(\sigma_{Color='Red'}((S\bowtie SP)\bowtie P))$$

$$\pi_{NameS}((S\bowtie SP)\bowtie\sigma_{Color='Red'}(P))$$

$$\pi_{NameS}(\pi_{NameS,IDS}(S)\bowtie(\pi_{IDS,IDP}(SP)\bowtie\pi_{IDP}(\sigma_{Color='Red'}(P))))$$

Построение реляционного выражения



Построение реляционного выражения

$\pi_{NameS}(\sigma_{S.IDS=SP.IDS\ \&\ SP.IDP=P.IDP\ \&\ Color='Red'}(\ S\times(SP\times P)))))$

/* Имена поставщиков, поставляющих хотя бы одну красную деталь */

```
SELECT NameS
FROM S,SP,P
WHERE S.IDS = SP.IDS
  AND SP.IDP = P.IDP
  AND  Color = 'Red'
```

$\pi_{NameS}(\sigma_{Color='Red'}(S\bowtie(SP\bowtie P)))$

$\pi_{NameS}(\sigma_{Color='Red'}((S\bowtie SP)\bowtie P))$

$\pi_{NameS}((S\bowtie SP)\bowtie\sigma_{Color='Red'}(P))$

$\pi_{NameS}(\pi_{NameS,IDS}(S)\bowtie(\pi_{IDS,IDP}(SP)\bowtie\pi_{IDP}(\sigma_{Color='Red'}(P))))$

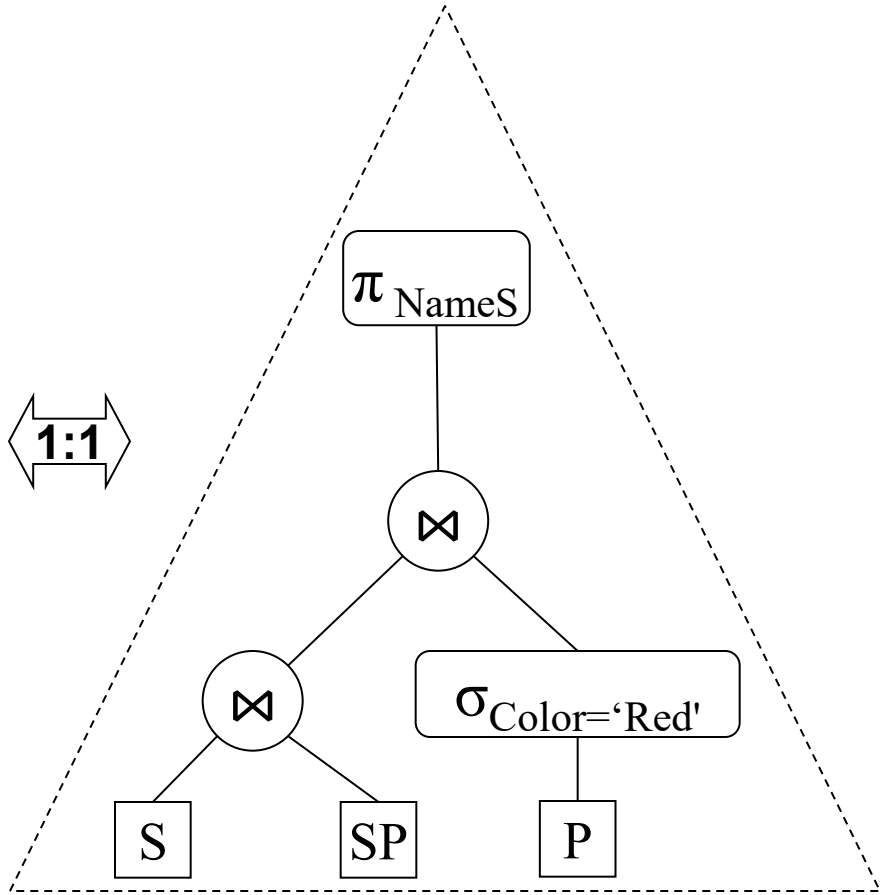
Какой вариант лучший?

Почему? Как построить? Как выбрать? • • •

Представление реляционного выражения в виде логического плана

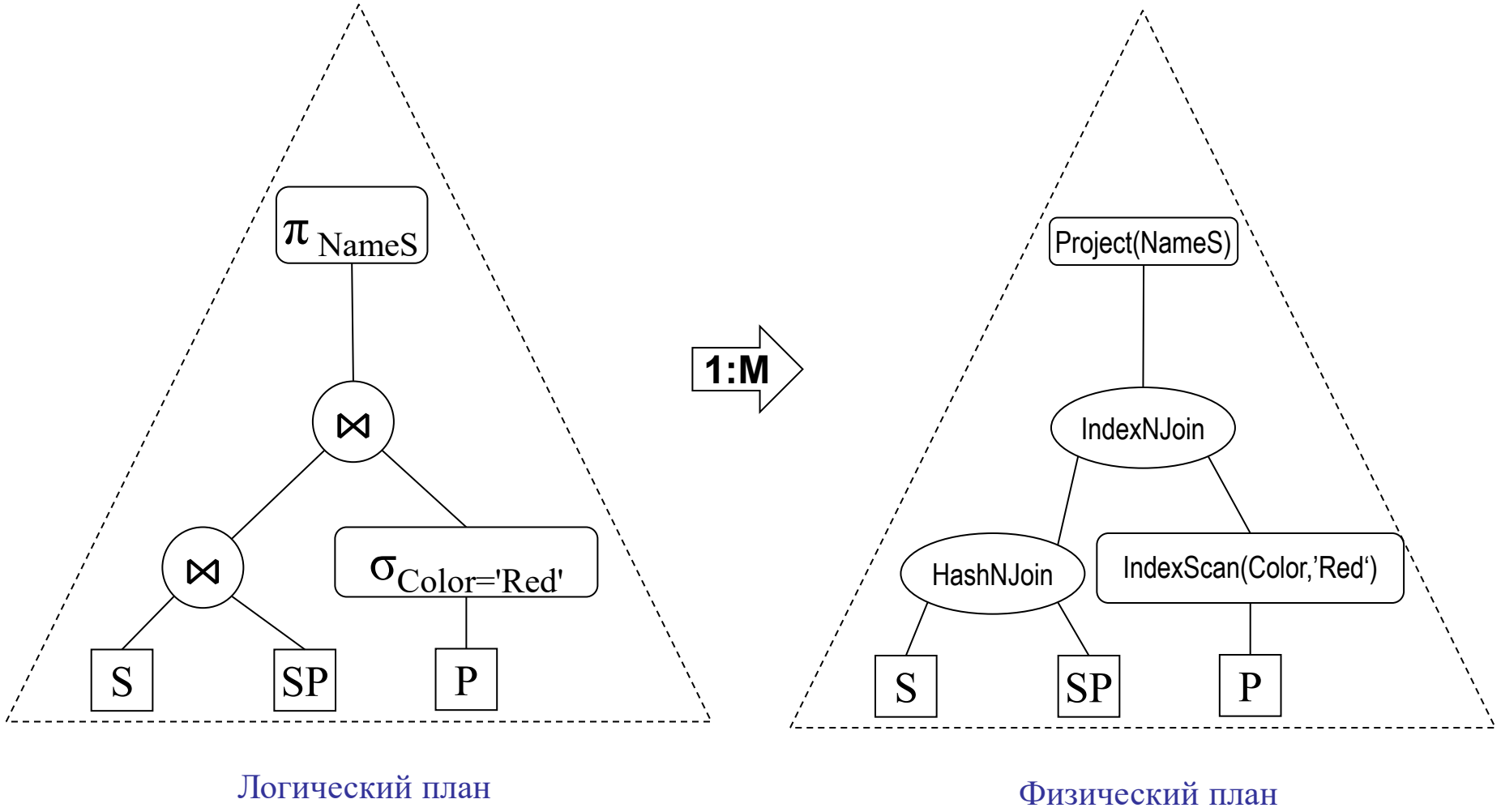
$$\pi_{NameS}((S \bowtie SP) \bowtie \sigma_{Color='Red'}(P))$$

Выражение реляционной алгебры



Логический план

Преобразование логического плана в физический



Логический план

Физический план

Конец лекции 1