

# Бази от данни

## Релационна алгебра

доц. д-р Димитър Димитров

# Въведение (1)

- Какво е алгебра?
- Какви алгебри познава всеки от нас?

# Въведение (2)

- Какви алгебри познава всеки от нас?
  - Елементарна алгебра
    - vs. аритметика
  - Булева алгебра
  - Матрична алгебра
    - Напр.  $n \times n$  матрици върху  $R$

# Алгебра

- Операнди
  - Променливи или стойности
- Операции
  - Позволяват създаването на нови стойности от операндите
  - Конструирание на изрази чрез прилагане на операциите върху операнди или други изрази

# Релационна алгебра

- Операнди
  - Променливи, които представят релации
  - Константи – крайни релации
- Операции
  - 1.Обединение, сечение, разлика
  - 2.Премахващи части от дадена релация
  - 3.Комбиниращи кортежите на две релации
  - 4.И др.

# Мотивация (1)

- Защо учим релационна алгебра?
- Предлага прости, но мощни начини да конструираме нови релации от дадени съществуващи
  - Можем да конструираме сложни изрази
- $\Rightarrow$  може да се използва за заявки върху релационни БД
  - Ще наричаме изразите *заявки*
- Теоретична основа на SQL

# Мотивация (2)

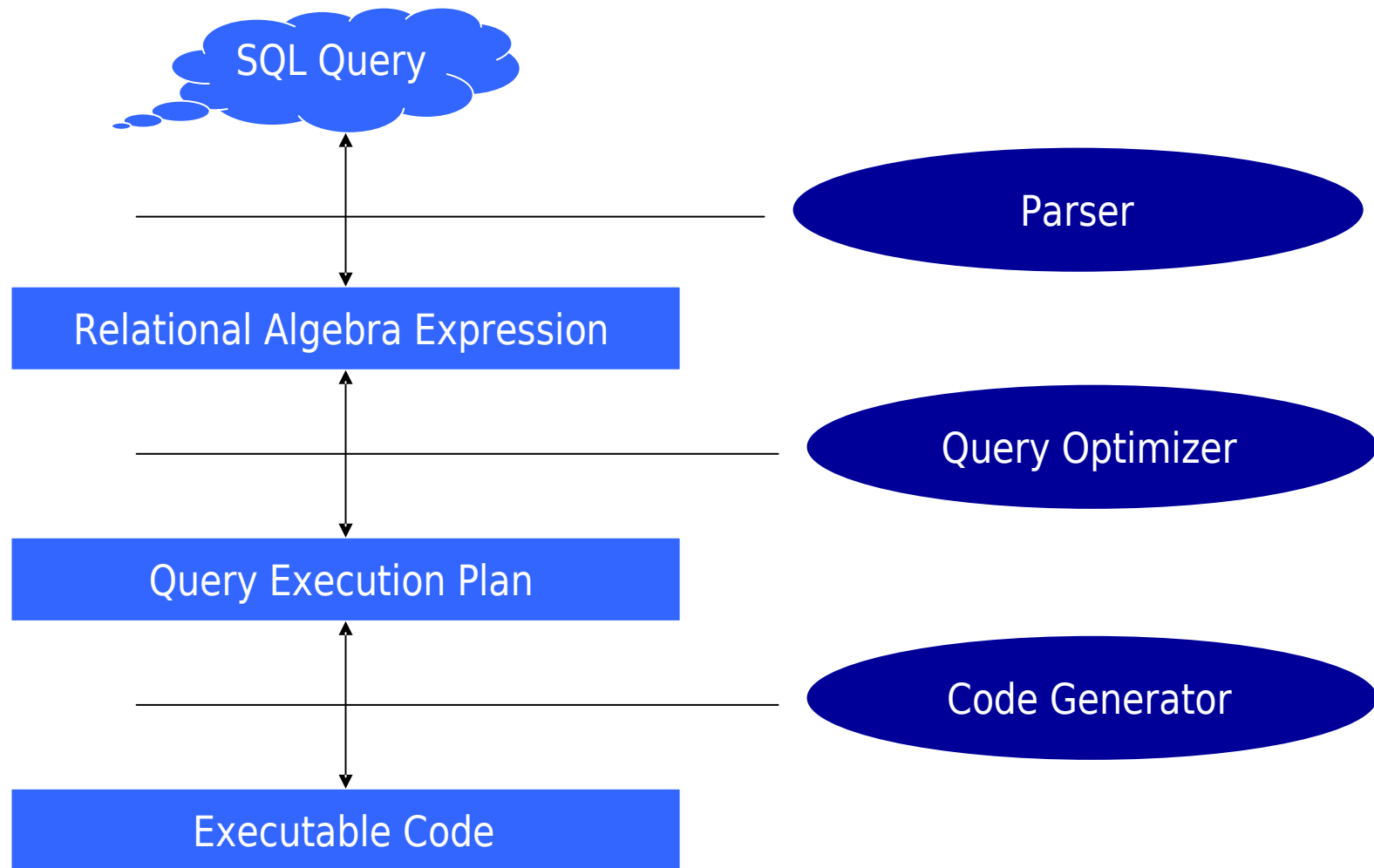
- Не е цялостна по Тюринг (Turing complete)
  - Недостатъци
  - Предимства
  - SQL цялостен ли е по Тюринг?

# Мотивация (3)

- Междинен език за изпълнение на заявките
- Използва се от СУБД
- Изразите могат да бъдат неефективни
- Множество от правила за обработка на алгебрични изрази
- Изразите се преобразуват в други, по-ефективни



# Изпълнение на заявки



- Ядро на релационната алгебра (РА)
  - Релациите са множества от кортежи
- Разширена РА
  - Мултимножества

# Основни класове операции

## 1. Операции над множества

- Обединение
- Сечение
- Разлика

## 2. Операции, премахващи части от дадена релация

- Проекция
- Селекция

## 3. Операции, комбиниращи кортежите на две релации

- Декартово произведение

## 4. Преименуване

# Припомняне на познати термини

- Комутативност
  - $a + b = b + a$
- Асоциативност
  - Мястото на скобите няма значение
  - $a + (b + c) = (a + b) + c$
  - $a * (b * c) \neq (a * b) * c$

# Операции върху множества (1)

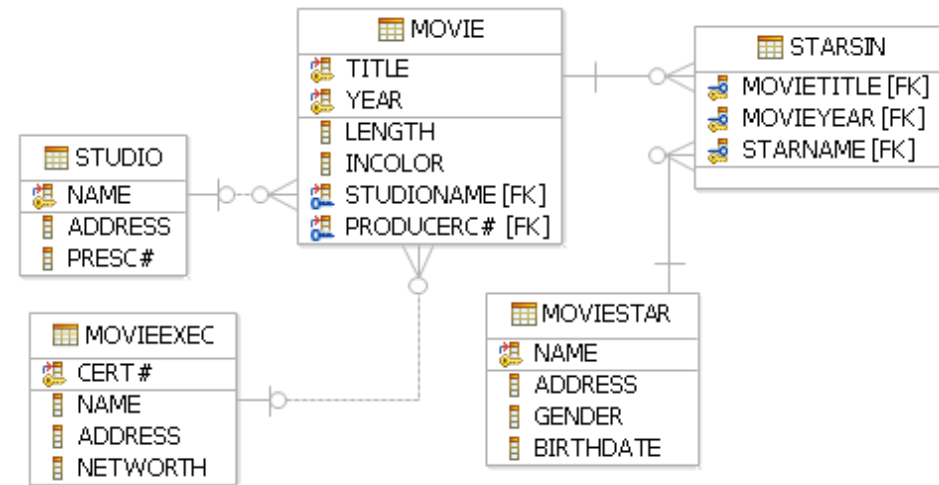
- Множества от какво? :)
- Обединение ( $\cup$ )
- Сечение ( $\cap$ )
- Разлика ( $-$ )
- Бинарни операции ли са?
- Кой са комутативни?
- Кой са асоциативни?
- Може ли да ги прилагаме върху произволни две релации?

# Операции върху множества (2)

- $R(A_1, \dots, A_n)$  и  $S(B_1, \dots, B_m)$ :
  - $n=m$  (еднаква степен/арност)
  - За всяко  $i = 1, \dots, n$ :  $\text{dom}(A_i) = \text{dom}(B_i)$
- Съответни оператори в SQL?

# Схема на примерна БД

- Studio (name, address, presC#)
- MovieExec (cert#, name, address, networth)
- Movie (title, year, length, inColor, studioName, producerC#)
- MovieStar (name, address, gender, birthdate)
- StarsIn (movieTitle, movieYear, starName)



# Операции върху множества – пример (1)

<i>R</i>			
<i>name</i>	<i>address</i>	<i>gender</i>	<i>birthdate</i>
Carrie Fisher	123 Maple St., Hollywood	F	9/9/99
Mark Hamil	456 Oak Rd., Brentwood	M	8/8/88

<i>S</i>			
<i>name</i>	<i>address</i>	<i>gender</i>	<i>birthdate</i>
Carrie Fisher	123 Maple St., Hollywood	F	9/9/99
Harrison Ford	789 Palm Dr., Beverly Hills	M	7/7/77



# Операции върху множества – пример (2)

$R \cap S$	<i>name</i>	<i>address</i>	<i>gender</i>	<i>birthdate</i>
	Carrie Fisher	123 Maple St., Hollywood	F	9/9/99

$R \cup S$	<i>name</i>	<i>address</i>	<i>gender</i>	<i>birthdate</i>
	Carrie Fisher	123 Maple St., Hollywood	F	9/9/99
	Harrison Ford	789 Palm Dr., Beverly Hills	M	7/7/77
	Mark Hamil	456 Oak Rd., Brentwood	M	8/8/88

$R - S$	<i>name</i>	<i>address</i>	<i>gender</i>	<i>birthdate</i>
	Mark Hamil	456 Oak Rd., Brentwood	M	8/8/88

# Проекция

- Унарна операция
- $\pi_L(R)$ , където  $L=A_1, \dots, A_n$  е списък от атрибути
- Създава нова релация
  - С какви атрибути?
  - С какви кортежи?
- Хоризонтална рестрикция
- SQL: SELECT DISTINCT

# Пример

- $\pi_{\text{title, year}}(\text{Movie})=?$
- $\pi_{\text{filmType}}(\text{Movie})=?$

<i>title</i>	<i>year</i>	<i>length</i>	<i>filmType</i>	<i>studioName</i>
Star Wars	1977	124	color	Fox
Mighty Ducks	1991	104	color	Disney
Wayne's World	1992	95	color	Fox

# Селекция

- Унарна операция
- $\sigma_C(R)$
- Създава нова релация със същата схема и тези кортежи на  $R$ , които удовлетворяват дадено условие  $C$ 
  - Условие върху атрибутите на  $R$
- Вертикална рестрикция
- Съответни оператори в SQL?

# Пример

- $\sigma_{\text{length} \geq 120 \text{ AND filmType} = \text{'color'}}(\text{Movie})$
- Съответна SQL заявка:  
SELECT \*  
FROM Movie  
WHERE length >= 120  
AND filmType='color';

<b>Movies</b>	<i>title</i>	<i>year</i>	<i>length</i>	<i>filmType</i>	<i>studioName</i>
	Star Wars	1977	124	color	Fox
	Mighty Ducks	1991	104	color	Disney
	Wayne's World	1992	95	color	Paramount

# Допълнителен материал: условие при селекция (1)

- Според някои автори е от вида  $a\theta b$ , където:
  - $a$  е име на атрибут на  $R$
  - $b$  е име на атрибут на  $R$  или константа
  - $\theta \in \{<, >, =, \leq, \geq, \neq\}$  - бинарна операция
- Трудно изразяване на по-сложни условия с конюнкция, дизюнкция и отрицание
  - Как?

# Допълнителен материал: условие при селекция (2)

- Изразяване на по-сложни условия:

$$\sigma_{\varphi \wedge \psi}(R) = \sigma_{\varphi}(R) \cap \sigma_{\psi}(R)$$

$$\sigma_{\varphi \vee \psi}(R) = \sigma_{\varphi}(R) \cup \sigma_{\psi}(R)$$

$$\sigma_{\neg \varphi}(R) = R - \sigma_{\varphi}(R)$$

- Образуват ли базис  $\wedge$ ,  $\vee$  и  $\neg$ ?

# Декартово произведение (1)

- Бинарна
- Комутативна
- Асоциативна
- $R \times S$
- Обща дефиниция (несвързана с РА):  
$$A \times B = \{(a, b) \mid a \in A \text{ and } b \in B\}$$
- РА: двойката  $(t, u)$  е кортеж, образуван от конкатенацията на двата кортежа  $t$  и  $u$



# Декартово произведение (2)

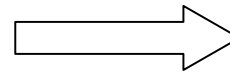
- Схемата на  $R \times S$  се състои от всички атрибути на  $R$  и на  $S$
- Ако  $R$  и  $S$  имат едноименен атрибут  $A$ , първо се преименува съответно на  $R.A$  и  $S.A$ 
  - Едва сега може да се каже, че схемата на  $R \times S$  е обединение на двете схеми
- Брой атрибути на схемата на  $R \times S = ?$
- Брой кортежи в  $R \times S = ?$
- Съответни оператори в SQL?

# Пример

R		
	A	B
	1	2
	3	4

x

S			
	B	C	D
	2	5	6
	4	7	8
	9	10	11

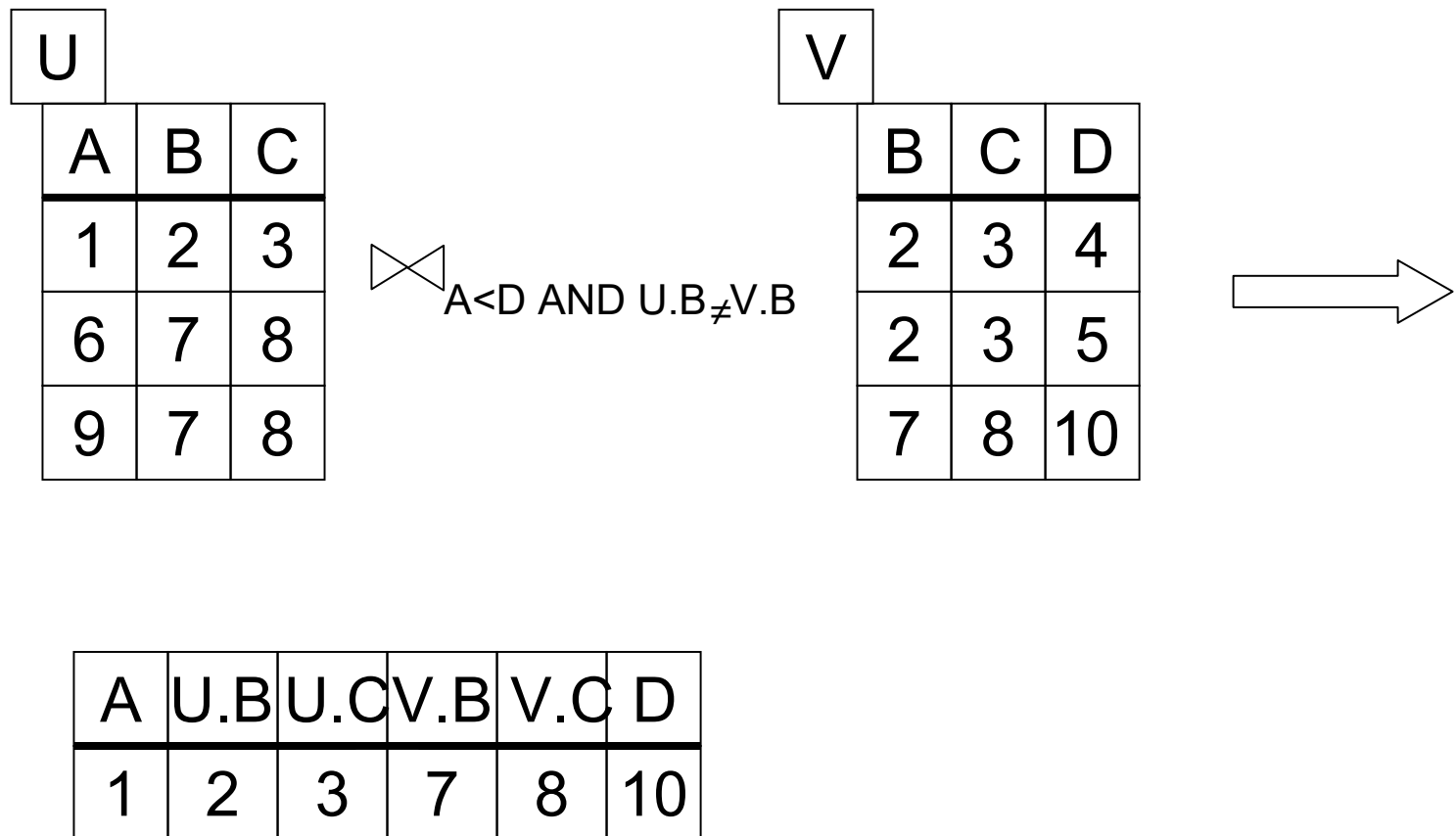


A	R.B	S.B	C	D
1	2	2	5	6
1	2	4	7	8
1	2	9	10	11
3	4	2	5	6
3	4	4	7	8
3	4	9	10	11

# Съединения

- Тета-съединение (Theta join)
- Бинарна операция
- $R \bowtie_c S = \sigma_c(R \times S)$ 
  - Декартово произведение, последвано от селекция
  - Схема – същата като на  $R \times S$

# Пример



# Видове съединения

- Тета-съединение
- Еквисъединение (Equijoin)
  - Условието включва само съвпадение по атрибутите
- Естествено съединение (Natural join)
  - $R \bowtie S$
  - Еквисъединение по всички двойки едноименни атрибути в R и S
  - Отстраняване на повтарящите се атрибути
  - Примери – от лекцията за нормализация

# Преименуване

- Унарна операция
- $\rho_{R2(A1, \dots, A_n)}(R1)$
- Дава нова схема на релацията  $R1 - R2(A_1, \dots, A_n)$ 
  - Със същите кортежи
- $\rho_s(R)$  – промяна само на името на релацията
- Алтернативни нотации -  $\rho_{a/b}(R)$  и др.
- SQL – AS
- Примери

# Основни операции

- Обединение
- Разлика
- Селекция
- Проекция
- Декартово произведение
- Преименуване
- Всички други операции могат да бъдат представени чрез тях
  - Ще ги наричаме допълнителни
  - Напр. сечение

# Представяне на допълнителни операции

- $R \cap S = R - (R - S)$
- $R \bowtie_c S = \sigma_c(R \times S)$
- $R(A_1, \dots, A_n) \bowtie S(B_1, \dots, B_m) =$   
 $\pi_{\{A_1, \dots, A_n\} \cup \{B_1, \dots, B_m\}}(\sigma_c(R \times S))$ , където  
 $C = \bigwedge \{R.D = S.D \mid D \in \{A_1, \dots, A_n\} \cap \{B_1, \dots, B_m\}\}$
- В SQL има ли значение дали условие е в ON или WHERE при INNER JOIN?



# Алгебрични изрази

- Пример от училище:
- $3+(5+a)*2$

# Приоритет на операциите в РА

- Приоритет от висок към нисък:
  - Унарни операции – селекция, проекция, преименуване
  - Декартово произведение и съединение
  - Сечение
  - Обединение и разлика
- Използване на скоби, където е необходимо

# РА – примерен израз

- Movies (title, year, length, filmType, studioName)
- Имената и годините на всички филми на студио Fox, които са дълги поне 100 минути
- $\pi_{\text{title, year}}(\sigma_{\text{length} \geq 100 \text{ AND studioName} = \text{'Fox'}}(\text{Movie}))$

ИЛИ

- $\pi_{\text{title, year}}(\sigma_{\text{length} \geq 100}(\text{Movie})) \cap \pi_{\text{title, year}}(\sigma_{\text{studioName} = \text{'Fox'}}(\text{Movie}))$

# Линейна нотация

- Вместо дълъг израз можем да запишем последователност от присвоявания
  - Подобно на код на C++/Java/JS/...
- Имена на временни релации
- Имплицитно преименуване на атрибути чрез схемите на новите релации
  - Пример:  $S(a,b) := \sigma_c(R)$
  - $:=$  или  $\leftarrow$

# Линейна нотация – пример

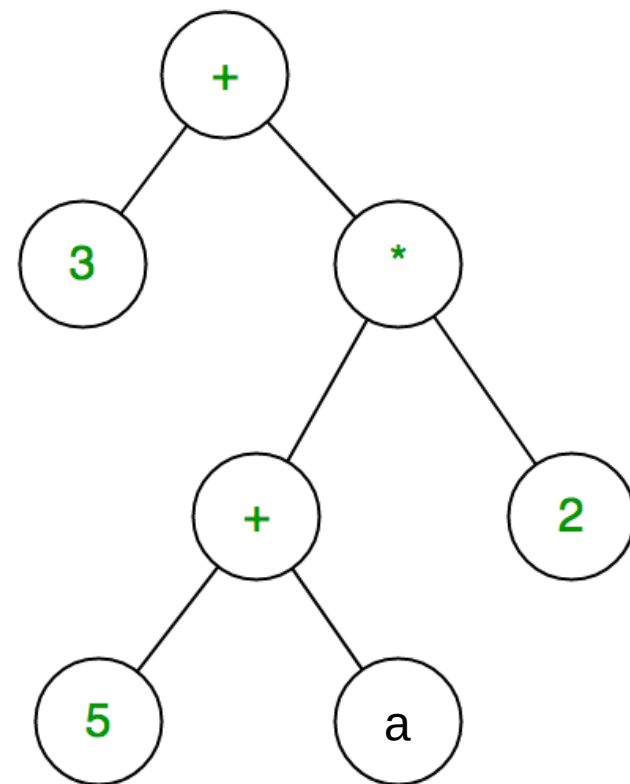
- Имената и годините на всички филми на студио Fox, които са дълги поне 100 минути
- $\text{LongMovies} := \sigma_{\text{length} \geq 100}(\text{Movie})$   
 $\text{FoxMovies} := \sigma_{\text{studioName} = \text{'Fox'}}(\text{Movie})$   
 $\text{LongFoxMovies} := \text{LongMovies} \cap \text{FoxMovies}$   
 $\text{Result} := \pi_{\text{title}, \text{year}}(\text{LongFoxMovies})$

# Представяне на изрази чрез дървета

- Пример от университета:

$$3+(5+a)*2$$

- Ползи
  - Лесно за разбиране
    - Приоритети, няма скоби, ...
  - Лесно за модификация
  - Удобно за алгоритми за оптимизация

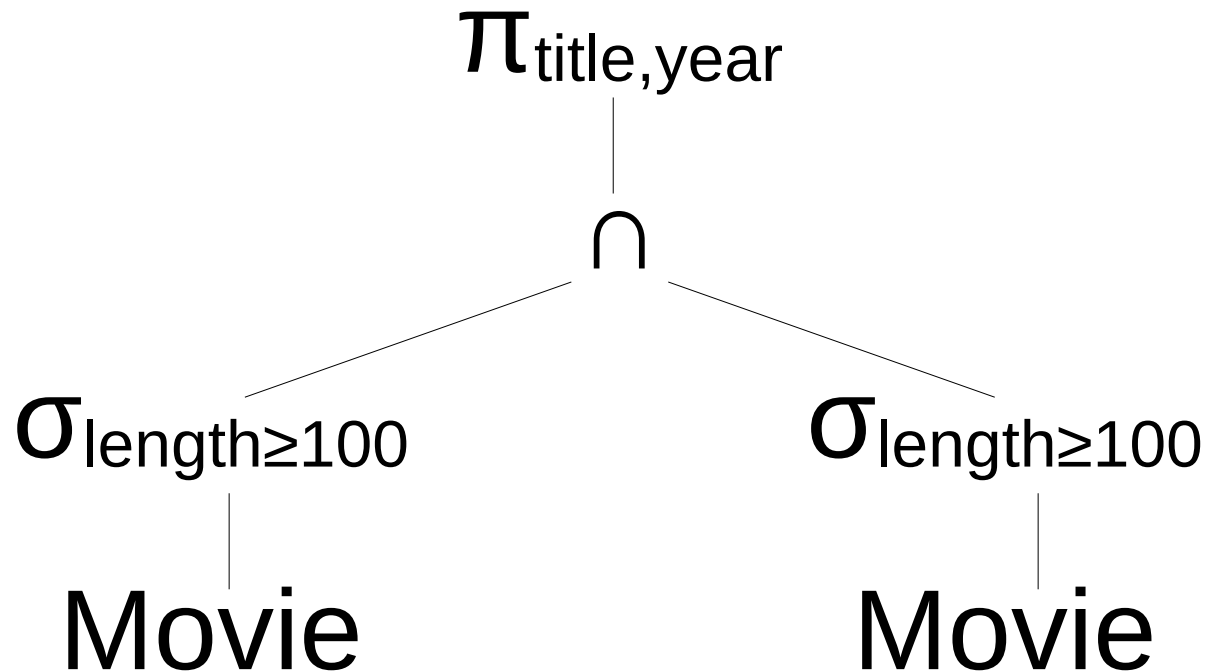


# Представяне на изрази от РА чрез дървета

- Листа (операнди) – релации
- Вътрешни възли – операции, приложени към техните наследници

# Пример

- $\pi_{\text{title,year}}(\sigma_{\text{length} \geq 100}(\text{Movie})) \cap \pi_{\text{title,year}}(\sigma_{\text{studioName} = \text{'Fox'}}(\text{Movie}))$





- Помощен инструмент за преобразуване SQL → PA и визуализация на дървета:

<https://nireas.iee.ihu.gr/relax/calc.htm>

# Допълнителен материал: Оптимизиране на заявки

- SQL – декларативен език
  - Какво, а не как
- Еквивалентни преобразувания, водещи до по-ефективни за изчисление изрази
  - По-бързи заявки

# Допълнителен материал: ОПТИМИЗИРАНЕ НА ЗАЯВКИ

- Как би могло да стане?
- Еднакви поддървета може да се изчисляват само веднъж
- По-важно: намаляване на средния размер на междинните релации
- Генериране на няколко алтернативи и избиране на достатъчно добра
- Перфектната оптимизация би отнела много време, затова – компромисни решения

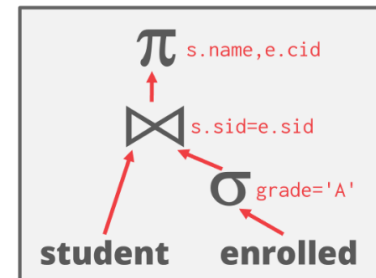
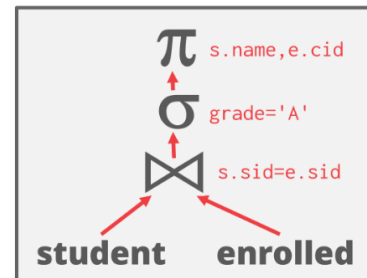
# Допълнителен материал: Оптимизиране на заявки

- Стратегии
  - Статични правила / евристики
    - Откриване на шаблони
    - Не се гледат екземпляри на релациите
  - Оптимизация, базирана на цена
    - Гледат се и данни
    - Избира се по-добър измежду няколко плана
  - AI (машинно обучение) – защо не?

# Допълнителен материал: ОПТИМИЗИРАНЕ НА ЗАЯВКИ

- Predicate Pushdown
- Селекциите да се извършват по-близо до източника на данни
- Намаляване на размера на междинните релации
- Как би изглеждала заявката, ако трябваше ние да укажем филтриране преди JOIN?

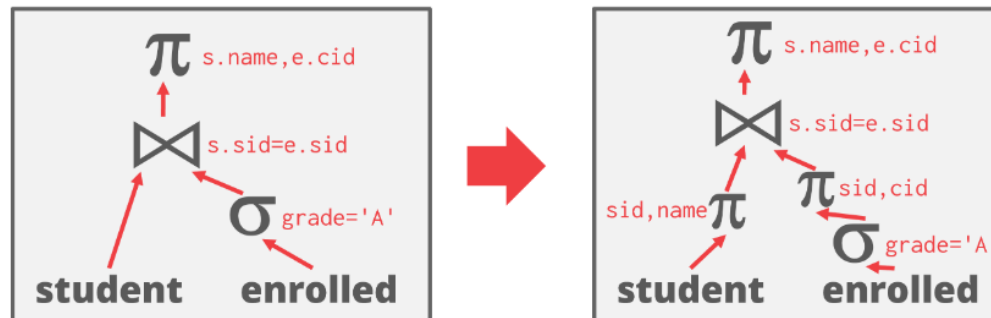
```
SELECT s.name, e.cid  
FROM student AS s, enrolled AS e  
WHERE s.sid = e.sid  
AND e.grade = 'A'
```



# Допълнителен материал: ОПТИМИЗИРАНЕ НА ЗАЯВКИ

- Аналогично: Projection Pushdown
- Използване само на необходимите атрибути

```
SELECT s.name, e.cid  
FROM student AS s, enrolled AS e  
WHERE s.sid = e.sid  
AND e.grade = 'A'
```



# Допълнителен материал: ОПТИМИЗИРАНЕ НА ЗАЯВКИ

- Декорелация

```
SELECT name FROM sailors AS S
WHERE EXISTS (
  SELECT * FROM reserves AS R
  WHERE S.sid = R.sid
  AND R.day = '2018-10-15'
)
```



```
SELECT name
FROM sailors AS S, reserves AS R
WHERE S.sid = R.sid
AND R.day = '2018-10-15'
```

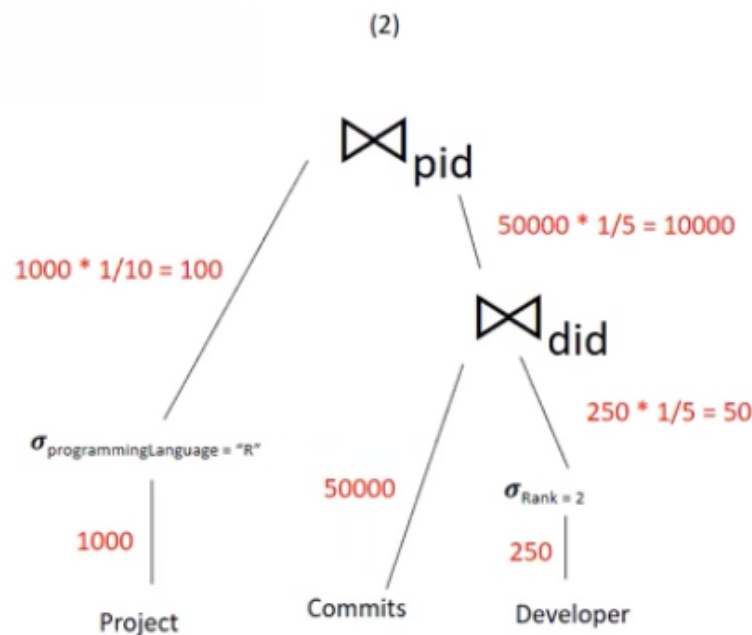
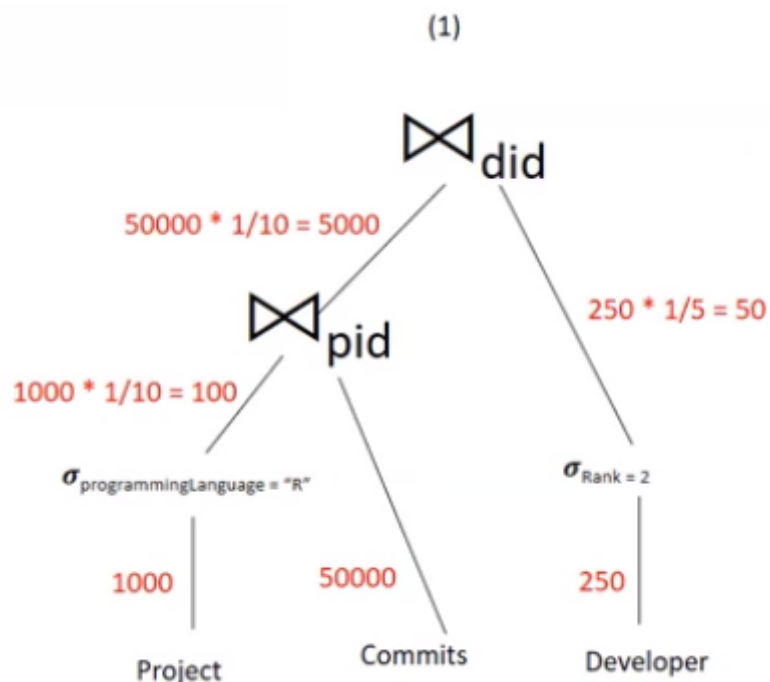
# Допълнителен материал: ОПТИМИЗИРАНЕ НА ЗАЯВКИ

- Пренареждане на съединенията
  - Няма значение в какъв ред пишем INNER JOIN операции в една заявка
  - СУБД съхранява статистики – брой различни стойности в колона и т.н.
  - По тях може да се пресметне груба оценка колко кортежа биха съдържаха междинните релации



# Допълнителен материал: ОПТИМИЗИРАНЕ НА ЗАЯВКИ

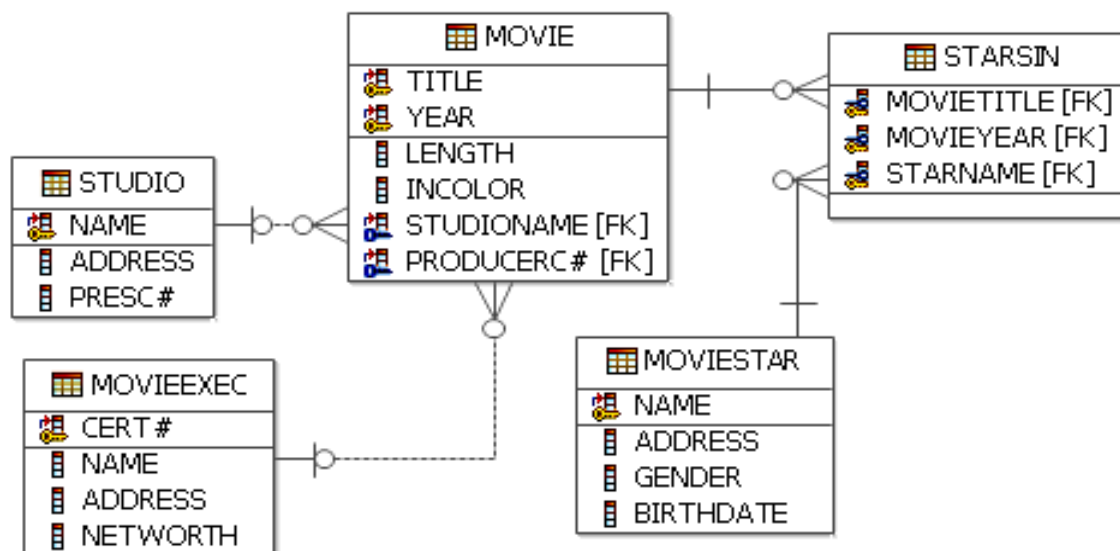
- Пример: 1000 проекта, 10 различни програмни езика и т.н.



- $100 + 5000 + 50 = 5150$      $100 + 50 + 10000 = 10150$   
очаквани междинни резултати

# Упражнения

- Още заявки върху Movies



# Въпроси?