CSE 562
Database Systems

Query Processing: Overview

Some slides are based or modified from originals by
Database Systems: The Complete Book,
Pearson Prentice Hall 2nd Edition
©2008 Garcia-Molina, Ullman, and Widom

cse@buffalo

Outline – Query Optimization

- Overview
- Relational algebra level
  - Algebraic Transformations
- Detailed query plan level
  - Estimate Costs
    - Estimating size of results
    - Estimating # of IOs
  - Generate and compare plans

Query Processing

- The query processor turns user queries and data modification commands into a query plan – a sequence of operations (or algorithms) on the database
  - from high level queries to low level commands
- Decisions taken by the query processor:
  - Which of the algebraically equivalent forms of a query will lead to the most efficient algorithm?
  - For each algebraic operator, what algorithm should we use to run the operator?
  - How should the operators pass data from one to the other? (e.g., main memory buffers, disk buffers)

Example

```
SELECT R.B, S.D
FROM R, S
WHERE R.A = 'c' AND S.E = 2 AND R.C = S.C
```

Example (cont.)

R	A	B	C
	a	1	10
	b	1	20
	c	2	10
	d	2	35
	e	3	45

S	C	D	E
	10	x	2
	20	y	2
	30	z	2
	40	x	1
	50	y	3

Answer

B	D
2	x

Query Processing

How do we execute a query eventually?

One idea

- Scan relations
- Do Cartesian product
- Select tuples
- Do projection

Example (cont.)

R x S	R.A	R.B	R.C	S.C	S.D	S.E
	a	1	10	10	x	2
	a	1	10	20	y	2
			⋮			
Bingo! Got one...	c	2	10	10	x	2
			⋮			

Example (cont.)

Relational algebra can be enhanced to describe plans

Example: **Plan I**

1. Scan R
2. For each tuple r of R, scan S
3. For each (r, s), where s in S select and project on the fly

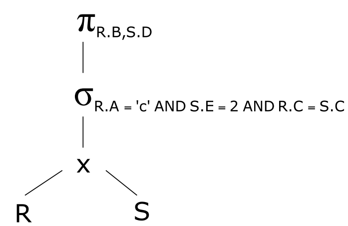
$$\pi^{FLY}_{R.B,S.D}$$
$$\sigma^{FLY}_{R.A = 'c' \text{ AND } S.E = 2 \text{ AND } R.C = S.C}$$
$$\times$$
$$R^{SCAN} \qquad S^{SCAN}$$

OR: $\pi^{FLY}_{B,D}[\sigma^{FLY}_{R.A = 'c' \text{ AND } S.E = 2 \text{ AND } R.C = S.C}(R^{SCAN} \times S^{SCAN})]$

Example (cont.)

FLY and SCAN are the defaults

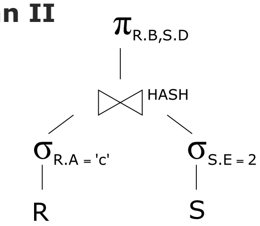
Example: **Plan I**

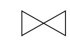


Example (cont.)

Another Idea

Example: **Plan II**





natural join

1. Scan R and S
2. Perform on the fly selections
3. Do hash join
4. Project

Example (cont.)

R

A	B	C
a	1	10
b	1	20
c	2	10
d	2	35
e	3	45

$\sigma_{R.A = 'c'}$

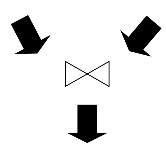
A	B	C
c	2	10

$\sigma_{S.E = 2}$

C	D	E
10	x	2
20	y	2
30	z	2

S

C	D	E
10	x	2
20	y	2
30	z	2
40	x	1
50	y	3



Example (cont.)

Example: **Plan III**

Use R.A and S.C Indexes

(1) Use R.A index to select R tuples with R.A = "c"

(2) For each R.C value found, use S.C index to find matching join tuples

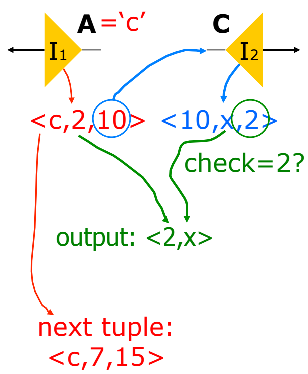
(3) Eliminate join tuples for which S.E ≠ 2

(4) Project B,D attributes

Example (cont.)

R

A	B	C
a	1	10
b	1	20
c	2	10
d	2	35
e	3	45

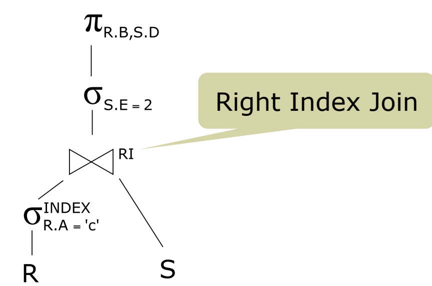


S

C	D	E
10	x	2
20	y	2
30	z	2
40	x	1
50	y	3

Example (cont.)

Algebraic Form of Plan



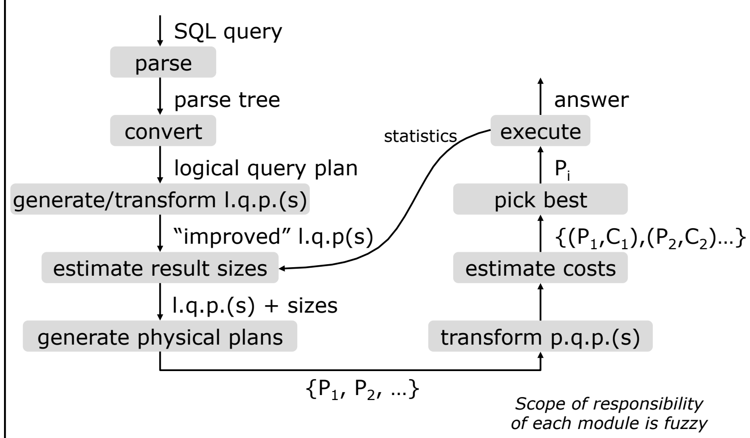
From Query To Optimal Plan

- Complex process
- Algebra-based logical and physical plans
- Transformations
- Evaluation of multiple alternatives

Issues in Processing and Optimization

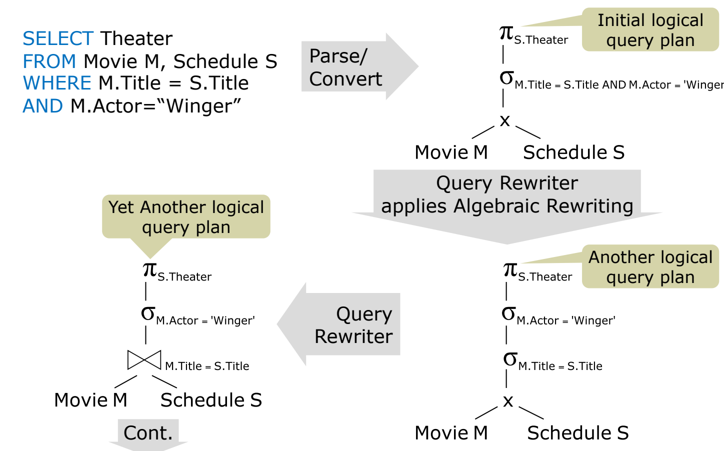
- Generate Plans
  - Employ efficient execution primitives for computing relational algebra operations
  - Systematically transform expressions to achieve more efficient combinations of operators
- Estimate Cost of Generated Plans
  - Statistics
- "Smart" Search of the Space of Possible Plans
  - always do the "good" transformations (relational algebra optimization)
  - prune the space (e.g., System R)
- Often the above steps are mixed

Architecture

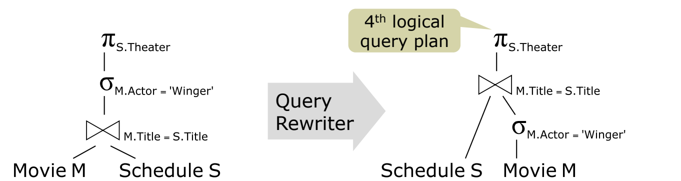


Scope of responsibility of each module is fuzzy

Example Journey of a Query

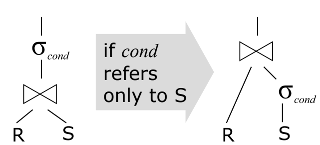


Example Journey of a Query (cont.)

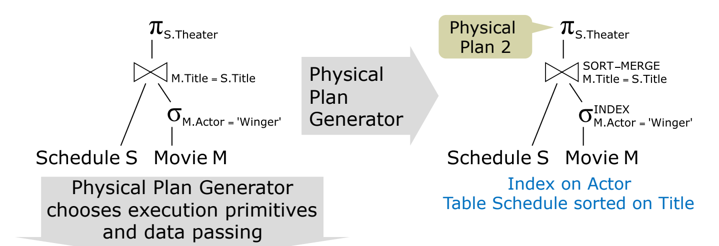


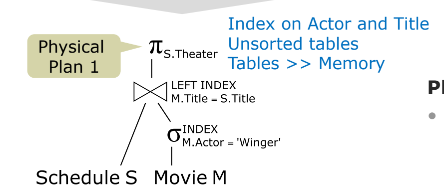
Summary of Logical Plan Generator

- 4 logical query plans created
- algebraic rewritings were used for producing the candidate logical query plans
- the last one is the winner (at least, cannot be a big loser)
- in general, multiple logical plans may "win" eventually



Example Journey of a Query (cont.)





Physical Plan Generator

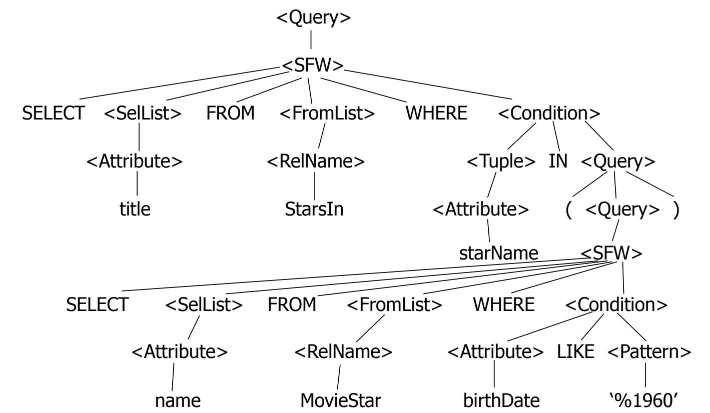
- More than one plans may be generated by choosing different primitives

Example: Nested SQL Query

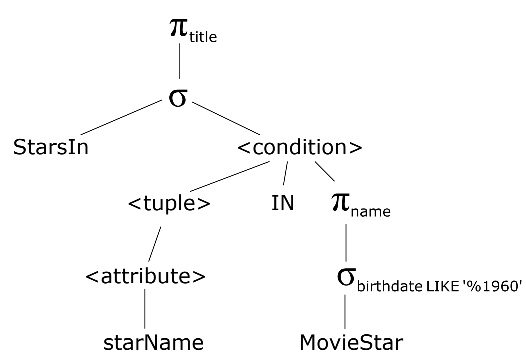
```
SELECT title
FROM StarsIn
WHERE starName IN
        (SELECT name
         FROM MovieStar
         WHERE birthdate LIKE '%1960')
```

Find the movies with stars born in 1960

Example: Parse Tree

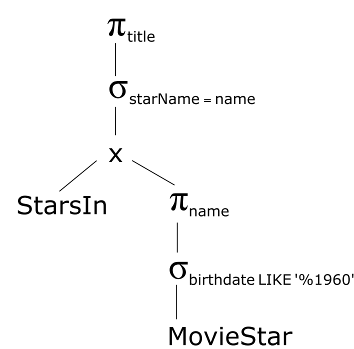


Example: Generating Relational Algebra



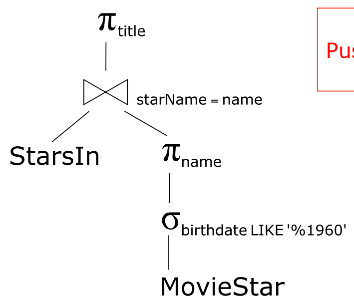
- An expression using a two-argument σ, midway between a parse tree and relational algebra

Example: Logical Query Plan



- May consider "IN" elimination as a transformation in the query rewriter or may consider it a task of the converter

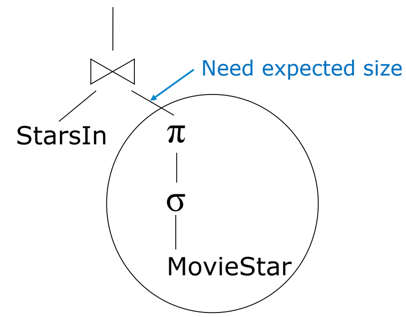
Example: Improved Logical Query Plan



<u>Question</u>
Push project to StarsIn?

Example: Estimate Result Sizes

Result sizes are important for selecting physical plans



Example: One Physical Plan