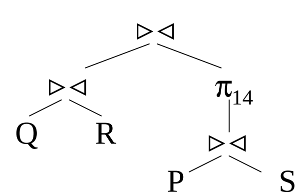
Query Processing

- The query processor turns user queries and data modification commands into a sequence of operations (or algorithm) 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 operations pass data from one to the other? (eg, main memory buffers, disk buffers)

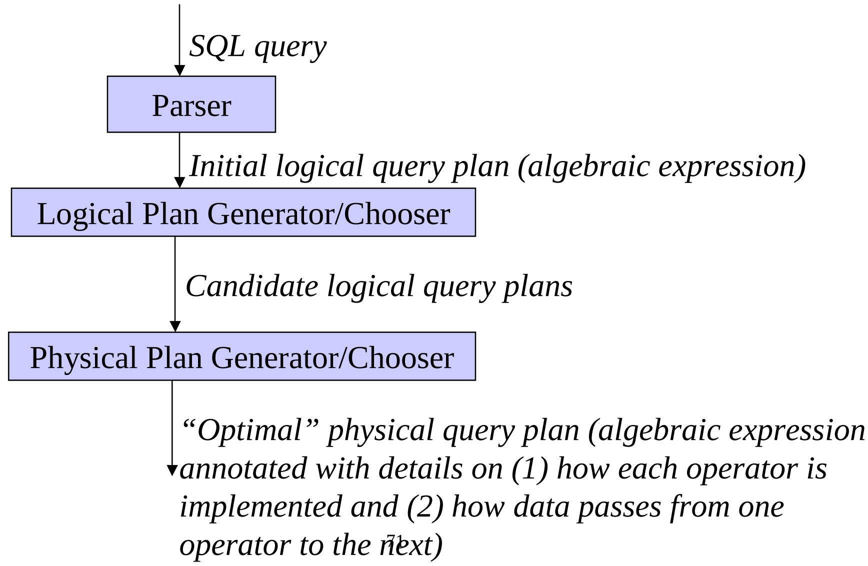
69

Journey of a Query

FO(SQL)	$\exists z(P(xz) \wedge Q(zy)) \wedge \ldots$
Relational Algebra	$\pi_{13}(P \bowtie Q) \bowtie \ldots$
Query Rewriting	$\pi_{14}(P \bowtie S) \bowtie Q \bowtie R$
Query Execution Plan	
Execution	
	Physical Level

70

Rough Architecture of the Query Processor



Query Rewriting

- Static Optimization
- Simplification Rules:

R: AB S: BC

$R \bowtie R = R$

$R - R = \emptyset$

$\emptyset \bowtie R = \emptyset$

$\sigma_{A=a}(R \bowtie S) = \sigma_{A=a}(R) \bowtie S$

$\pi_{AB}(R \bowtie S) = R \bowtie \pi_{B}(S)$

$\pi_{A}(\pi_{AB}(T)) = \pi_{A}(T)$

etc.

Query Evaluation Plan

- Example: For given algebra query, order of evaluation
- Factors:
  - Statistical info on data
    (# tuples, selectivity, etc.)
    e.g. $p \bowtie q \bowtie r$

    1000 tuples 200 tuples 3 tuples

  - Common Subexpressions

Implementation of Individual Operations: $\bowtie$, σ, ...

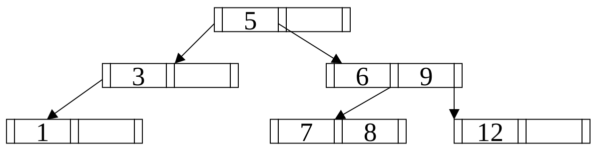
- Related to access method and file organization
- Unordered files: $\bowtie$ takes $O(n^2)$
- Files ordered by some key field
  - Access by index: $\bowtie$ takes O(nlog n)

- Ordered file (sequential) by name

Name	Sal	Name	Sal	...
Aaron		Abbott		

- e.g. R: AB S: BC; index on B
  - R $\bowtie$ S takes O(nlog n)

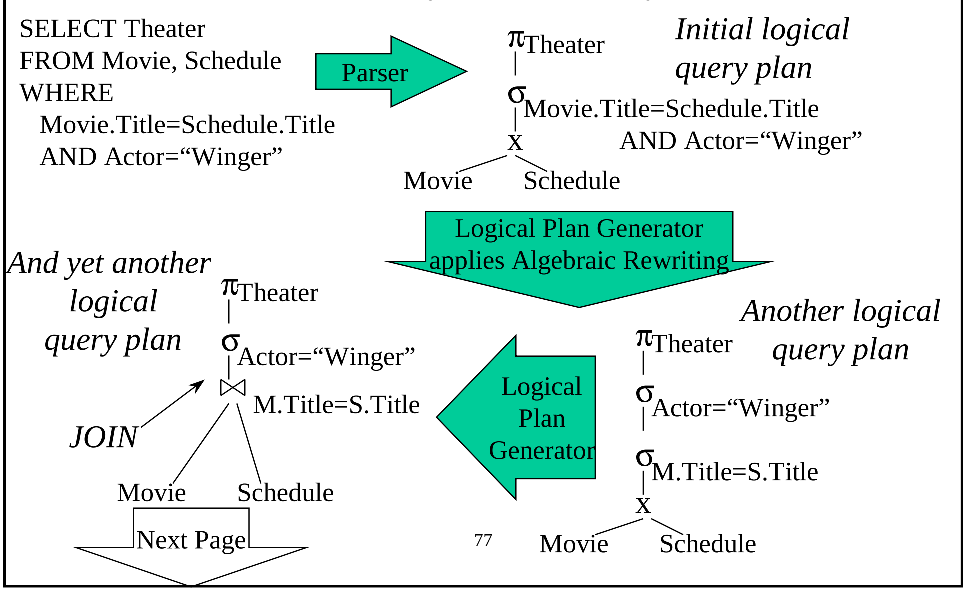
- Index: auxiliary file providing fast access to physical location of record with given key value
- Search tree



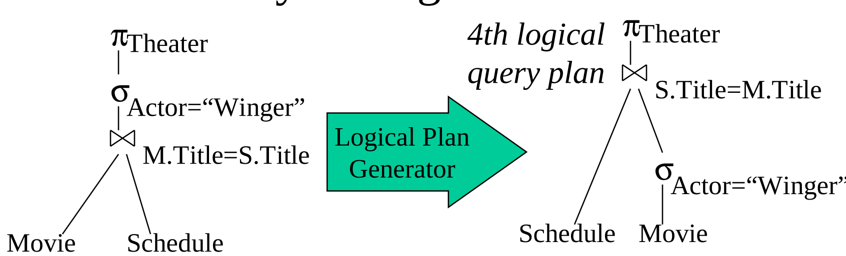
- If tree is balanced, access time is $\log_p(n)$ for p-ary trees
- Special search trees: B-trees, B^+-trees
  - Always balanced

- Note: SQL allows specification of indexes
  - e.g. CREATE INDEX(NAME_INDEX)
    ON EMPLOYEE(NAME)
- Note: can also use to declare a key:
  - CREATE UNIQUE INDEX(No_INDEX)
    ON EMPLOYEE(No)
  - No: Social Security Number

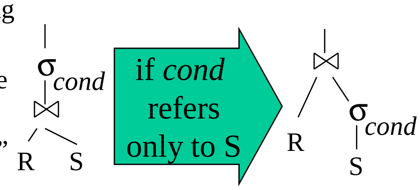
Query Processing and Optimization: The Journey of a Query



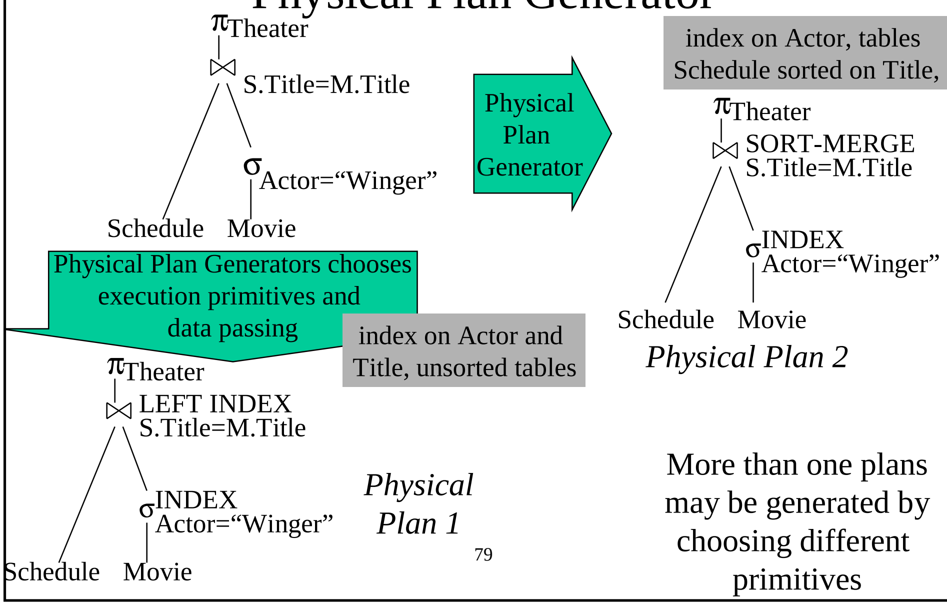
The Journey of a Query cont'd: Summary of Logical Plan Generator



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



The Journey of a Query Continues at the Physical Plan Generator



79

Issues in Query Processing and Optimization

- Generate Plans
  - systematically transform expressions
  - employ execution primitives for computing relational algebra operations
- 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

80

Exact Minimization of # of Joins

Possible for large set of queries:

- Basic, unnested SQL queries
- One-stage QBE queries
- Relational algebra with π, σ, $\bowtie$
- Relational calculus with $\exists$, $=$, $\wedge$

- Basic idea:

$$
\begin{array}{ccc}
\text{query} & & \text{simplified query} \\
\downarrow & & \uparrow \\
\text{pattern} & \xrightarrow[\text{minimize}]{} & \text{pattern} \\
\text{(like QBE)} & &
\end{array}
$$

81

Minimization for a simple case

- Relational algebra with:
  - one input relation
  - π, $\bowtie$
  - σ_{cond} where cond is A = const

$\downarrow$

rspj query

- First step: write query in QBE style, using a pattern $\rightarrow$ tableau
  - e.g. tableau for $\pi_{AB}(R) \bowtie \pi_{BC}(R)$, where R: ABC, is:

A	B	C	
a	b	c_1	pattern
a_1	b	c	
a	b	c	answer

82

Tableaux

- Tableau over R: <s, t> where
  - s is the set of rows over R, with variables or constants
  - t is the “answer” row

  t is over a subset of att(R)

  t(A) is a variable occurring in the A column of S, or a constant

- Terminology
  - variables in t: distinguished
    - denoted a, b, c
  - other variables: nondistinguished
    - denoted a_1 a_2 … b_1 b_2 … c_1 c_2

83

Example

- Directors who are also actors

- QBE:

movie	title	dir	actor
		-d	
			-d

answer	dir
I.	-d

- Tableau:

	title	dir	actor	
S	t_1	d	a_1	
	t_2	d_1	d	
T		d		answer row

84

Example

R: ABC

q: $\pi_{AC}(\pi_{AB}(R) \bowtie \pi_{BC}(\sigma_{A=5}(\pi_{AB}(R)) \bowtie \pi_{AC}(R)))$

Relational calculus:

$\exists b_2[\exists c_1(R(ab_2c_1)) \wedge \exists a_1(\exists c_2(R(a_1b_2c_2)) \wedge a_1=5 \wedge \exists b_1(R(a_1b_1c)))]$

Prenex form (move $\exists$ to left):

$\exists a_1b_1b_2c_1c_2[R(ab_2c_1) \wedge R(a_1b_2c_2) \wedge R(a_1b_1c) \wedge a_1=5]$

85

Example, continued

Tableau:

A	B	C
a	b_2	c_1
5	b_2	c_2
5	b_1	c
a		c

Note: # rows in tableau = 1 + # joins

Note: like QBE query

R	A	B	C
	-a	$-b_2$	
	5	$-b_2$	
	5		-c

answer	A	C
I.	-a	-c

86

Minimizing Tableau

- <s, t>: tableau
- Definition
  - Mapping f on variables is a homomorphism on <s, t> iff:
    - h(t) = t
    - h(c) = c   if c is constant
    - h(s) $\subseteq$ s
- Fact
  - A row r $\in$ S is redundant iff for some homomorphism
    - f of <s, t>, r $\notin$ f(s)
  - Note: <f(s), t> equivalent to <s, t>

87

Example

- Tableau <s, t>

A	B	C	
a	b_1	c_1	
a_1	b	c_1	
a	b_2	c_2	s
a_2	b_2	c	
a_2	b_1	c	
a	b	c	t

- Let f: $c_2 \to c_1$
  $b_2 \to b_1$
  $a_1 \to a_1$
  $a_2 \to a_2$

88

Example, continued

- f(<s, t>):

A	B	C
a	b_1	c_1
a_2	b_1	c
a_1	b	c_1
a	b	c

No redundant rows: MINIMAL

- Fact: all minimal equivalent tableaux are isomorphic!

89

Example

R: ABC

q: $\pi_{AB}(\sigma_{B=5}(R)) \bowtie \pi_{BC}(\pi_{AB}(R) \bowtie \pi_{AC}(\sigma_{B=5}(R)))$

Tableau:

A	B	C
a	5	c_1
a_1	5	c_2
a_1	5	c
a	5	c

Minimization

$a_1 \rightarrow a_1$

$c_2 \rightarrow c_1$

$c_2 \rightarrow c$

A	B	C
a	5	c_1
a_1	5	c
a	5	c

Minimized query

$\sigma_{B=5}(\pi_{AB}(R) \bowtie \pi_{BC}(R))$

90

Functional Dependencies

- Dependencies: statements about properties of valid data
  - e.g.: “Every student is a person”
    - inclusion dependency
  - “Each employee works in more than one department”
    - NAME $\rightarrow$ DEPARTMENT
    - functional dependency
- Use of dependencies:
  - check data integrity
  - query optimization
  - schema design $\rightarrow$ “normal forms”

91

Functional Dependencies

- Functional dependency over R:
  - expression $x \rightarrow y$ where $x, y \subseteq att(R)$
- A relation r over R satisfies $x \rightarrow y$
  - iff whenever two tuples in y agree on x, they also agree on y

e.g.

SCHEDULE	THEATER	TITLE
	odeon	tango
	forum	delicatessen

Satisfies THEATER $\rightarrow$ TITLE

SCHEDULE	THEATER	TITLE
	odeon	tango
	forum	delicatessen
	forum	l’amant

Violates THEATER $\rightarrow$ TITLE, satisfies TITLE $\rightarrow$ THEATER

92

Using FDs in Query Optimization

- Example: R: ABC with $B \rightarrow C$
  - query $\pi_{AB}(R) \bowtie \pi_{BC}(R)$
- Fact: if R satisfies $B \rightarrow C$ then
  - $\pi_{AB}(R) \bowtie \pi_{BC}(R) = R$
  - Why: tableau of query is

A	B	C
a	b	c_1
a_1	b	c
a	b	c

  - Note: if <a, b, c> $\in$ R and $<a_1, b, c> \in$ R then $c_1 = c$ since R satisfies $B \rightarrow C$
  - So, tableau is equivalent to

A	B	C
a	b	c
a	b	c

$\equiv$ R

93

- In general: can simplify tableau <s, t> over R if R satisfies a set F of FDs.
- Algorithm: The Chase
  - Input: tableau <s, t>, set F of FDs
  - Output: tableau $CHASE_F$<s, t> on all relations satisfying F
- Note: assume without loss of generality that FDs in F are of the form $X \rightarrow A$ where A is one attribute

94

The Chase

- Repeat until no change
  - For each $X \rightarrow A$ in F do
    - For each t_1, t_2 in s such that $t_1(x) = t_2(x)$, $t_1(A) \neq t_2(A)$ do
      - if $t_1(A)$, $t_2(A)$ are nondistinguished then replace one by the other in s
      - if $t_1(A)$ distinguished, $t_2(A)$ nondistinguished then replace $t_2(A)$ by $t_1(A)$ in s
      - if $t_1(A)$ is constant, $t_2(A)$ is variable then replace $t_2(A)$ by $t_1(A)$ in s
      - if $t_1(A)$ is constant, $t_2(A)$ is constant then STOP and output $\varnothing$

95

Optimization of RSPJ Queries with FDs

- q over R, set of FDs F over R
  - build tableau <s, t> of q
  - compute $CHASE_F$<s, t>
  - minimize $CHASE_F$<s, t>
  - construct rspj query from minimal tableau

96

Example

– R: ABC F = {B → A}

– $q = \pi_{BC}(\sigma_{A=5}(R)) \bowtie \pi_{AB}(R)$

<s, t>:

A	B	C
5	b	c
a	b	c_1
a	b	c

CHASE<s, t>:

A	B	C
5	b	c
5	b	c_1
5	b	c

MIN:

A	B	C
5	b	c
5	b	c

RSPJ: $\sigma_{A=5}(R)$

97

Example

– R: ABC F = {B → A}, $q = \pi_{BC}(\sigma_{A=5}(R)) \bowtie \pi_{AB}(\sigma_{A=6}(R))$

<s, t>:

A	B	C
5	b	c
6	b	c_1
6	b	c

CHASE<s, t>: ∅

QUERY: ∅

98

Example

– R: ABC F = {A → B}, $q = \pi_{AB}(R) \bowtie \pi_A(\sigma_{B=5}(R)) \bowtie \pi_{AB}(\pi_{AC}(R) \bowtie \pi_{BC}(R))$

<s, t>:

A	B	C
a	b	c_1
a	b_1	c_2
a_1	b	c_2
a	5	c_3
a	b	

CHASE<s, t>:

A	B	C
a	5	c_1
a	5	c_2
a_1	5	c_2
a	5	c_3
a	5	

MIN:

A	B	C
a	5	c_1
a	5	

RSPJ: $\pi_{AB}(\sigma_{B=5}(R))$

99