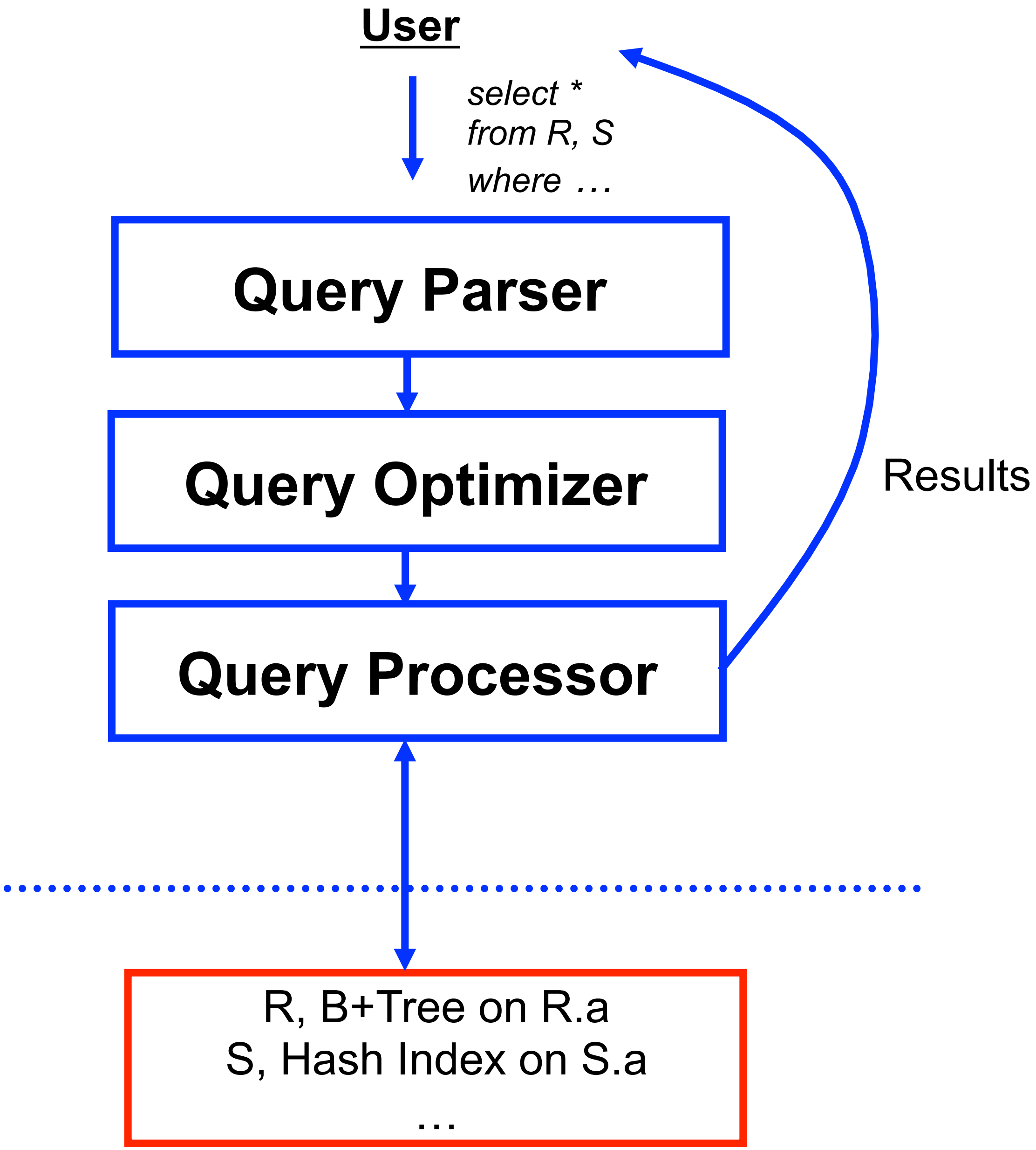
CMSC424: Database Design

Instructor: Amol Deshpande
amol@cs.umd.edu

Overview



User

*select **
from R, S
where …

Query Parser

Query Optimizer

Query Processor

Results

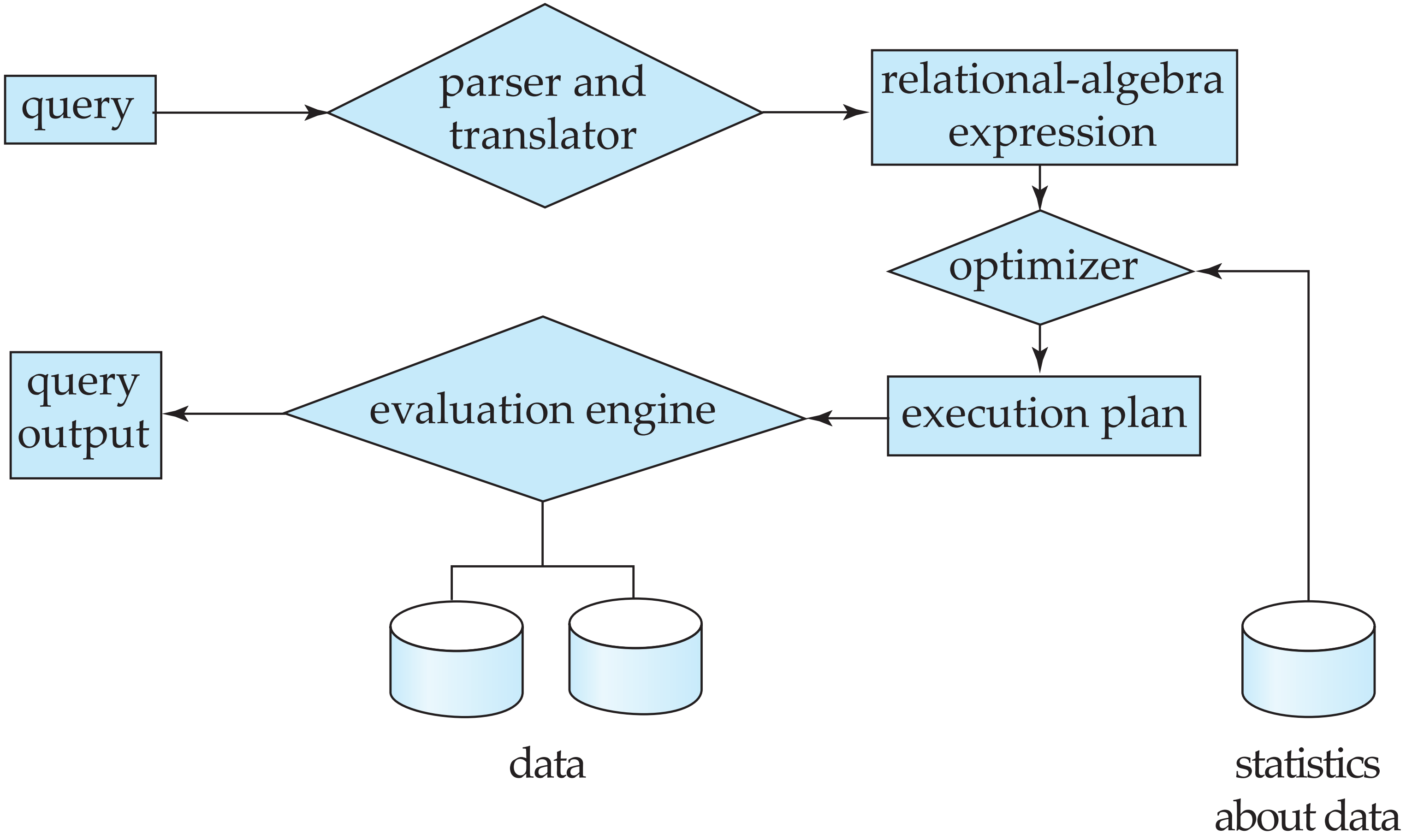
R, B+Tree on R.a
S, Hash Index on S.a
…

Resolve the references,
Syntax errors etc.
Converts the query to an internal format
relational algebra like

Find the *best* way to evaluate the query
- Which index to use ?
- What join method to use ?
- …

Read the data from the files
Do the query processing
- *joins, selections, aggregates*
- …

Overview


query
parser and translator
relational-algebra expression
optimizer
execution plan
evaluation engine
query output
data
statistics about data

“Cost”

- Complicated to compute
  - Too many factors
  - Total resource consumptions vs response time
- We will focus on disk:
  - Number of I/Os not sufficient by itself
    - Number of seeks matters a lot… why ?
  - t_T – time to transfer one block
  - t_S – time for one seek
  - Cost for *b* block transfers plus *S* seeks

$$b * t_T + S * t_S$$

  - Measured in *seconds*

Query Processing

- Overview
- Selection operation
- Join operators
- Sorting
- Other operators
- Putting it all together…

Selection Operation

- select * from person where SSN = “123”
- Option 1: Sequential Scan
  - Read the relation start to end and look for “123”
    - Can always be used (not true for the other options)
  - Cost ?
    - *Let* b_r *= Number of relation blocks*
    - Then:
      - 1 seek and br block transfers
    - So:
      - $t_S + b_r * t_T$ *sec*
    - *Improvements:*
      - *If SSN is a key, then can stop when found*
        - *So on average,* $b_r/2$ *blocks accessed*

Selection Operation

- select * from person where SSN = “123”
- Option 2 : Binary Search:
  - Pre-condition:
    - *The relation is sorted on SSN*
    - *Selection condition is an equality*
      - E.g. can’t apply to “*Name like ‘%424%’*”
  - Do binary search
    - Cost of finding the *first* tuple that matches
      - $\lceil \log_2(b_r) \rceil * (t_T + t_S)$
      - All I/Os are random, so need a seek for all
        - The last few are closeby, but we ignore such small effects
  - Not quite: What if 10000 tuples match the condition ?
    - Incurs additional cost

Selection Operation

- select * from person where SSN = “123”
- Option 3 : Use Index
  - Pre-condition:
    - *An appropriate index must exist*
  - Use the index
    - Find the first leaf page that contains the search key
    - Retrieve all the tuples that match by following the pointers
      - If primary index, the relation is sorted by the search key
        - Go to the relation and read blocks sequentially
      - If secondary index, must follow all pointers using the index

Selection w/ B+-Tree Indexes

	cost of finding the first leaf	cost of retrieving the tuples
primary index, candidate key, equality	$h_i * (t_T + t_S)$	$1 * (t_T + t_S)$
primary index, not a key, equality	$h_i * (t_T + t_S)$	$1 * (t_T + t_S) + (b – 1) * t_T$ *Note: primary == sorted* *b = number of pages that contain the matches*
secondary index, candidate key, equality	$h_i * (t_T + t_S)$	$1 * (t_T + t_S)$
secondary index, not a key, equality	$h_i * (t_T + t_S)$	$n * (t_T + t_S)$ *n = number of records that match* This can be bad

h_i = height of the index