Fundamentals of Query Optimization

CPE 444/742 - Juggapong Natwichai

Outline

- Motivation
- Query processing
- Heuristic query optimization
- Cost-based query optimization

Motivation

- Database Schema
  - Student(ID, Name, Major, Gender)
  - Course(Code, Title, Instructor, Textbook)
  - Enrolment(StID, CrsID, Grade)

Motivation (cont.)

ID	Name	Major	Gender
001	Ponjet	CPE	Male
002	Patis	CPE	Male
003	Bantueng	EE	Male
004	Pruet	ME	Male
005	Tasinee	CPE	Female

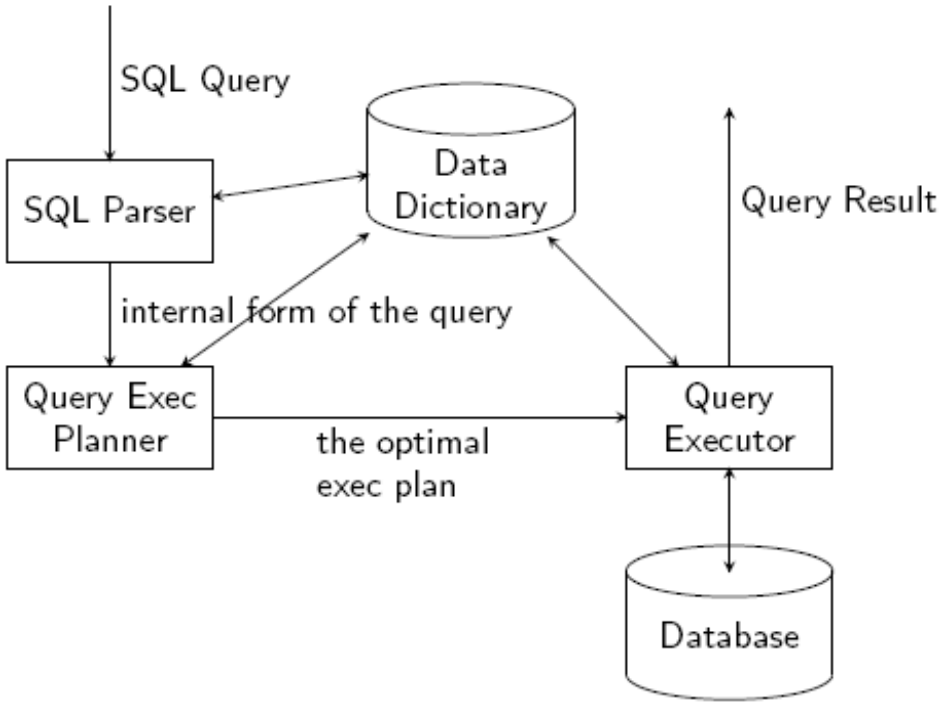
StID	CrsID	Grade
001	101	A
002	101	B
003	101	A
004	101	B
002	203	A
005	101	A

Code	Title	Instructor	Textbook
101	Basic Computer	Arnan	Introduction to Computer
102	Programming Language	Sutasinee	C Programming
203	OOP	Narathip	OOP using Java 6

Motivation (cont.)

- Query in SQL:
  - SELECT title, major
    FROM student, course, enrolment
    WHERE enrolment.StID = student.ID
    AND enrolment.CrsID = course.Code
    AND enrolment.Grade = 'A'

Motivation (cont.)



From Fundamentals of Database Systems, 5th Edition by Ramez Elmasri and Shamkant B. Navathe, Addison-Wesley 2007

Motivation (cont.)

- Query in RA:

$$\pi_{Title,Major}(\sigma_{Grade=A}(student \otimes enrolment \otimes course))$$

Motivation (cont.)

- Query in RA version 2:

$$\pi_{Title,Major}(\pi_{Title,StID}(course \otimes \sigma_{Grade=A} enrolment) \otimes student)$$

Motivation (cont.)

- Query in RA version 3:

$$\pi_{Title,Major}(\pi_{CrsID,Major}(student \otimes \sigma_{Grade=A} enrolment) \otimes course)$$

Motivation (cont.)

- Optimization problem with complexity as the cost.
- Existing approaches
  - Heuristic query optimization
  - Cost-based query optimization
- How the "cost" be determined?

Query Processing

- Select processing
- Project processing
- Join processing

Query Processing (cont.)

- Select processing
  - Approaches:
    - Linear search
    - Binary search on data files
    - Binary search on index files
      - Point query
      - Range query
    - Search by tree-structure index
  - O(n), not so slow.

Query Processing (cont.)

- Project processing
  - Two-steps processing
    - Sort
    - Remove duplication
  - O(n log n)
  - Comparing with select:

N	Projection (N+N log N)	Selection (N)
1,000	4,000	1,000
10,000	50,000	10,000
100,000	600,000	100,000
1,000,000	7,000,000	1,000,000

Query Processing (cont.)

- Join processing
  - Nested-loop join
  - Single-loop join
  - Sort-merge join

Query Processing (cont.)

- Join processing – sort-merge join (1)

A=

Read	Join Attribute A
x	1
	3
	4
	5

B=

Read	Join Attribute B
x	2
	3
	5
	6

t(A)<t(B)

Move A

Query Processing (cont.)

- Join processing – sort-merge join (2)

A=

Read	Join Attribute A
	1
x	3
	4
	5

B=

Read	Join Attribute B
x	2
	3
	5
	6

t(A)>t(B)

Move B

Query Processing (cont.)

- Join processing – sort-merge join (3)

A=

Read	Join Attribute A
	1
x	3
	4
	5

B=

Read	Join Attribute B
	2
x	3
	5
	6

t(A)=t(B)

Move B

Get result

Query Processing (cont.)

- Join processing – sort-merge join (4)

A=

Read	Join Attribute A
	1
x	3
	4
	5

B=

Read	Join Attribute B
	2
	3
x	5
	6

t(A)<t(B)

Move A

Query Processing (cont.)

- Join processing – sort-merge join (5)

A=

Read	Join Attribute A
	1
	3
x	4
	5

B=

Read	Join Attribute B
	2
	3
x	5
	6

t(A)<t(B)

Move A

Query Processing (cont.)

- Join processing – sort-merge join (6)

A=

Read	Join Attribute A
	1
	3
	4
x	5

B=

Read	Join Attribute B
	2
	3
x	5
	6

t(A)=t(B)

Move B

Get result

Query Processing (cont.)

- Join processing – sort-merge join (6)

A=

Read	Join Attribute A
	1
	3
	4
x	5

B=

Read	Join Attribute B
	2
	3
	5
x	6

t(A)<t(B)

Done

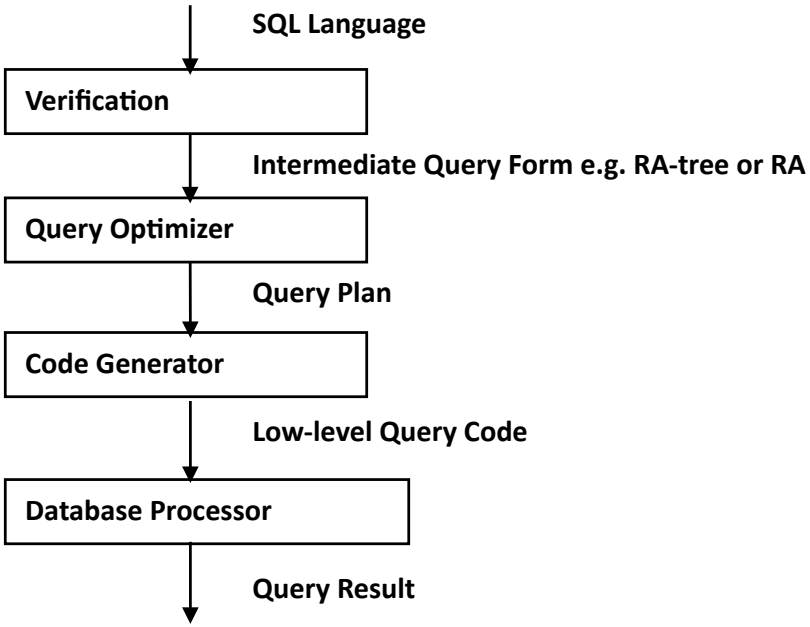
Query Processing (cont.)

- Complexity in summary

N	Sort-merge join 2*(N+N log N)	Project (N+N log N)	Selection (N)	Cartesian Product (N^2)
1,000	8,000	4,000	1,000	1,000,000
10,000	100,000	50,000	10,000	100,000,000
100,000	1,200,000	600,000	100,000	10,000,000,000
1,000,000	14,000,000	7,000,000	1,000,000	1,000,000,000,000

Heuristic Query Optimization

- Query processing sequence



Heuristic Query Optimization (cont.)

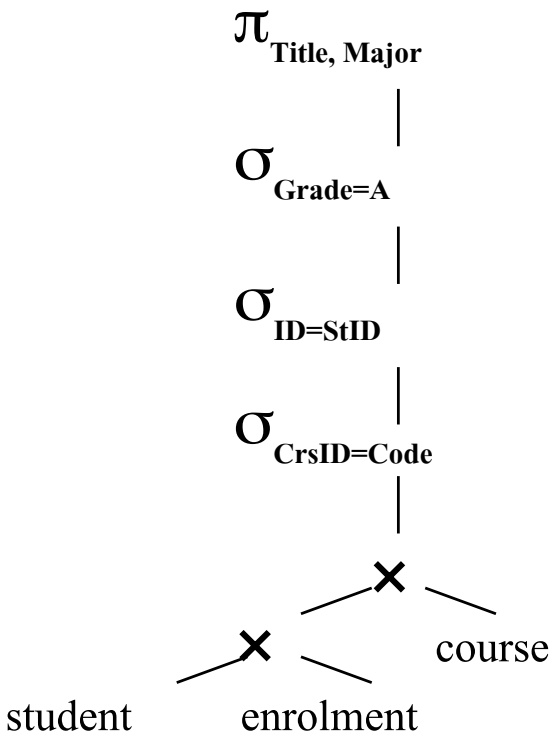
- Approach: re-sequence RA operations
- Selectivity property
  - High: large attribute domain
  - Low: small attribute domain
- Select operation
  - Prioritize high selectivity
- Project operation
  - Prioritize low selectivity
- Typically, the DB query processing engines will consult the system catalog e.g. FK-PK relationship or real statistics are to be utilized.

Heuristic Query Optimization (cont.)

- Example:
  - SELECT title, program
    FROM student, course, enrolment
    WHERE enrolment.StID = student.ID
    AND enrolment.CrsID = course.Code
    AND enrolment.Grade = 'A'

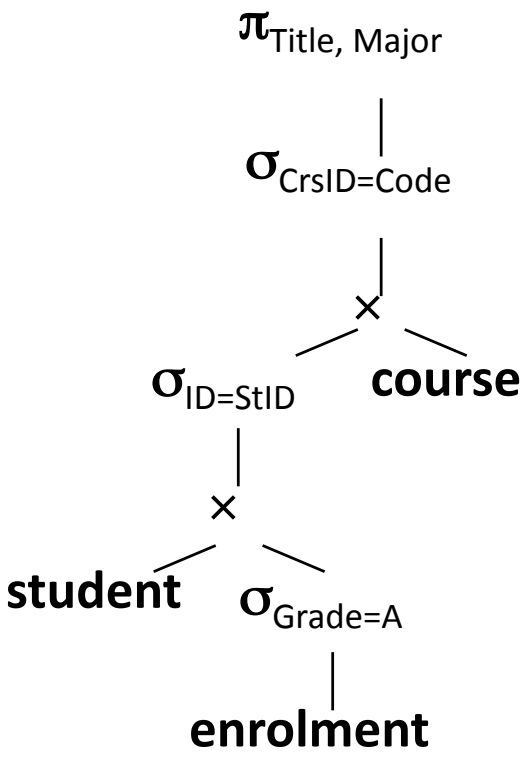
$$\pi_{Title,Major}(\sigma_{Grade=A}(student \otimes enrolment \otimes course))$$

Heuristic Query Optimization (cont.)



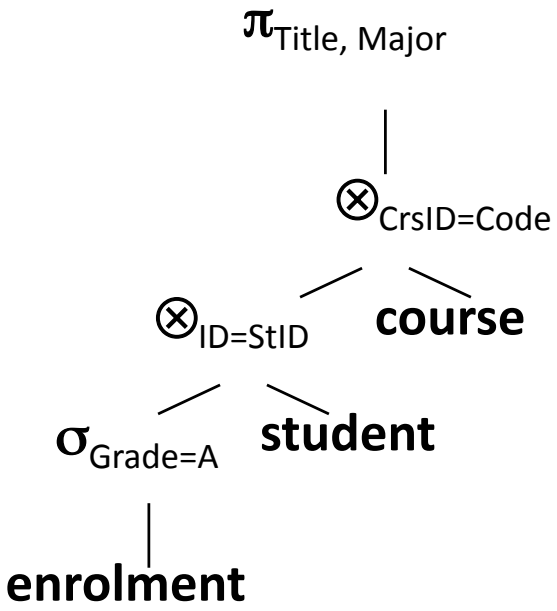
Heuristic Query Optimization (cont.)

- 1st Heuristic: Push down selection ranked by the most restriction.



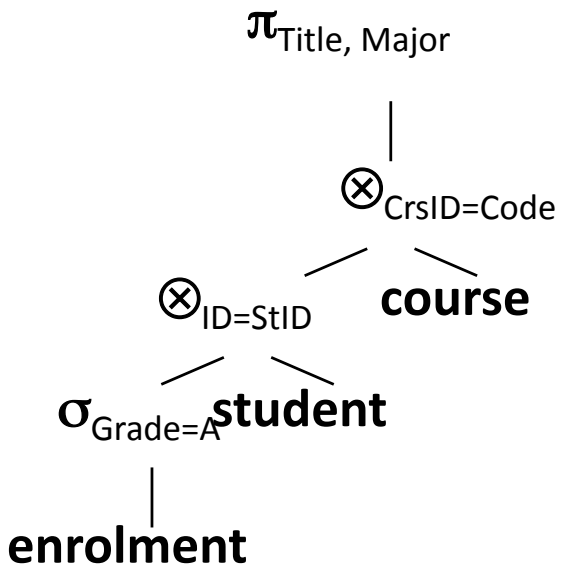
Heuristic Query Optimization (cont.)

- 2nd Heuristic: Turn Cartesian product into joins
  - Note that the selection on enrolment is to be processed first.



Heuristic Query Optimization (cont.)

- 3rd Heuristic: Do most restrictive joins first



Heuristic Query Optimization (cont.)

- 4th Heuristic: Push down projection