



Module 57

Partha Pratim  
Das

Objectives &  
Outline

Query  
Optimization

Equivalent  
Expressions

Evaluation Plan  
Cost

Transformation of  
Relational  
Expressions

Equivalence Rules

Example

Plan Generation

Module Summary

# Database Management Systems

## Module 57: Query Processing and Optimization/2: Optimization

Partha Pratim Das

Department of Computer Science and Engineering  
Indian Institute of Technology, Kharagpur

*ppd@cse.iitkgp.ac.in*



## Module 57

Partha Pratim  
Das

### Objectives & Outline

Query  
Optimization

Equivalent  
Expressions

Evaluation Plan  
Cost

Transformation of  
Relational  
Expressions

Equivalence Rules

Example

Plan Generation

Module Summary

- Understood the overall flow for Query Processing and defined the Measures of Query Cost
- Studied the algorithms for processing Selection Operations, Sorting, Join Operations and a few Other Operations



## Module 57

Partha Pratim  
Das

### Objectives & Outline

Query  
Optimization

Equivalent  
Expressions

Evaluation Plan

Cost

Transformation of  
Relational  
Expressions

Equivalence Rules

Example

Plan Generation

Module Summary

- To understand the basic issues for optimizing queries
- To understand how transformation of Relational Expressions can create alternates for optimization



## Module 57

Partha Pratim  
Das

### Objectives & Outline

Query  
Optimization

Equivalent  
Expressions

Evaluation Plan  
Cost

Transformation of  
Relational  
Expressions

Equivalence Rules

Example

Plan Generation

Module Summary

- Introduction to Query Optimization
- Transformation of Relational Expressions



## Module 57

Partha Pratim  
Das

Objectives &  
Outline

**Query  
Optimization**

Equivalent  
Expressions

Evaluation Plan

Cost

Transformation of  
Relational  
Expressions

Equivalence Rules

Example

Plan Generation

Module Summary

# Introduction to Query Optimization



# Query Optimization

## Module 57

Partha Pratim  
Das

Objectives &  
Outline

Query  
Optimization

Equivalent  
Expressions

Evaluation Plan  
Cost

Transformation of  
Relational  
Expressions

Equivalence Rules

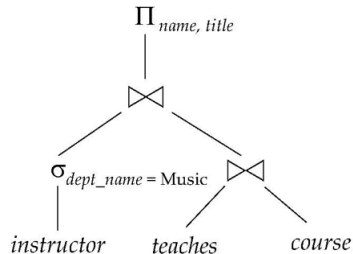
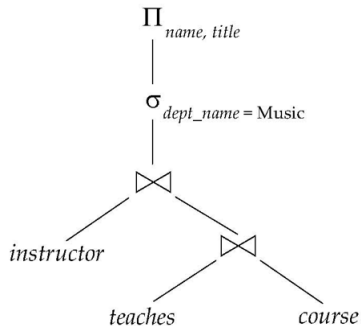
Example

Plan Generation

Module Summary

- Alternative ways of evaluating a given query
  - Equivalent expressions
  - Different algorithms for each operation

*course(course id, title, dept name, credits)*  
*instructor(ID, name, dept name, salary)*  
*teaches(ID, course id, sec id, semester, year)*





# Query Optimization (2)

## Module 57

Partha Pratim  
DasObjectives &  
OutlineQuery  
OptimizationEquivalent  
ExpressionsEvaluation Plan  
CostTransformation of  
Relational  
Expressions

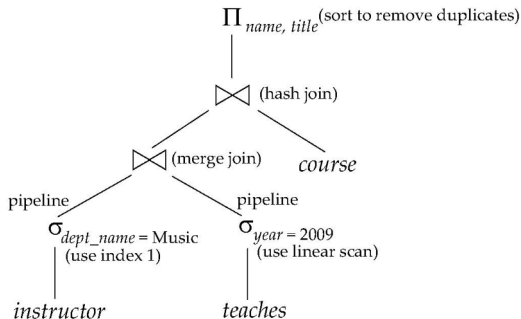
Equivalence Rules

Example

Plan Generation

Module Summary

- An **evaluation plan** defines exactly what algorithm is used for each operation, and how the execution of the operations is coordinated

$$\Pi_{name, title}(\sigma_{dept\_name = \text{"Music"} \wedge year = 2009} (instructor \bowtie teaches \bowtie course))$$


$$\begin{aligned} &course(course\ id, title, dept\ name, credits) \\ &instructor(\underline{ID}, name, dept\ name, salary) \\ &teaches(\underline{ID}, \underline{course\ id}, \underline{sec\ id}, \underline{semester}, year) \end{aligned}$$



# Query Optimization (3)

## Module 57

Partha Pratim  
Das

Objectives &  
Outline

Query  
Optimization

Equivalent  
Expressions

Evaluation Plan

Cost

Transformation of  
Relational  
Expressions

Equivalence Rules

Example

Plan Generation

Module Summary

- Cost difference between evaluation plans for a query can be enormous
  - For example, seconds vs. days in some cases
- Steps in **cost-based query optimization**
  - a) Generate logically equivalent expressions using **equivalence rules**
  - b) Annotate resultant expressions to get alternative query plans
  - c) Choose the cheapest plan based on **estimated cost**
- Estimation of plan cost based on:
  - Statistical information about relations.
    - ▷ Examples: number of tuples, number of distinct values for an attribute
  - Statistics estimation for intermediate results
    - ▷ to compute cost of complex expressions
  - Cost formulae for algorithms, computed using statistics





## Module 57

Partha Pratim  
Das

Objectives &  
Outline

Query  
Optimization

Equivalent  
Expressions

Evaluation Plan  
Cost

Transformation of  
Relational  
Expressions

Equivalence Rules

Example

Plan Generation

Module Summary

# Transformation of Relational Expressions



# Transformation of Relational Expressions

## Module 57

Partha Pratim  
Das

Objectives &  
Outline

Query  
Optimization

Equivalent  
Expressions

Evaluation Plan  
Cost

Transformation of  
Relational  
Expressions

Equivalence Rules

Example

Plan Generation

Module Summary

- Two relational algebra expressions are said to be **equivalent** if the two expressions generate the same set of tuples on every legal database instance
  - Note: order of tuples is irrelevant
  - We do not care if they generate different results on databases that violate integrity constraints
- In SQL, inputs and outputs are multisets of tuples
  - Two expressions in the multiset version of the relational algebra are said to be equivalent if the two expressions generate the same multiset of tuples on every legal database instance.
- An **equivalence rule** says that expressions of two forms are equivalent
  - Can replace expression of first form by second, or vice versa



# Equivalence Rules

## Module 57

Partha Pratim  
Das

Objectives &  
Outline

Query  
Optimization

Equivalent  
Expressions

Evaluation Plan  
Cost

Transformation of  
Relational  
Expressions

Equivalence Rules

Example

Plan Generation

Module Summary

- 1 Conjunctive selection operations can be deconstructed into a sequence of individual selections

$$\sigma_{\theta_1 \wedge \theta_2}(E) = \sigma_{\theta_1}(\sigma_{\theta_2}(E))$$

- 2 Selection operations are commutative

$$\sigma_{\theta_1}(\sigma_{\theta_2}(E)) = \sigma_{\theta_2}(\sigma_{\theta_1}(E))$$

- 3 Only the last in a sequence of projection operations is needed, the others can be omitted

$$\pi_{L_1}(\pi_{L_2}(\dots(\pi_{L_n}(E)))) = \pi_{L_1}(E)$$

- 4 Selections can be combined with Cartesian products and theta joins

$$\sigma_{\theta}(E_1 \bowtie E_2) = E_1 \bowtie_{\theta} E_2$$

$$\sigma_{\theta_1}(E_1 \bowtie_{\theta_2} E_2) = E_1 \bowtie_{\theta_1 \wedge \theta_2} E_2$$



# Equivalence Rules (2)

## Module 57

Partha Pratim  
Das

Objectives &  
Outline

Query  
Optimization

Equivalent  
Expressions

Evaluation Plan  
Cost

Transformation of  
Relational  
Expressions

Equivalence Rules

Example

Plan Generation

Module Summary

5 Theta-join operations (and natural joins) are commutative

$$E_1 \bowtie_{\theta} E_2 = E_2 \bowtie_{\theta} E_1$$

6 a. Natural join operations are associative:

$$(E_1 \bowtie E_2) \bowtie E_3 = E_1 \bowtie (E_2 \bowtie E_3)$$

b. Theta joins are associative in the following manner:

$$(E_1 \bowtie_{\theta_1} E_2) \bowtie_{\theta_2 \wedge \theta_3} E_3 = E_1 \bowtie_{\theta_1 \wedge \theta_3} (E_2 \bowtie_{\theta_2} E_3)$$

where  $\theta_2$  involves attributes from  $E_2$  and  $E_3$  only



# Equivalence Rules (3): Pictorial Depiction

## Module 57

Partha Pratim  
Das

Objectives &  
Outline

Query  
Optimization

Equivalent  
Expressions

Evaluation Plan  
Cost

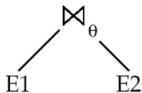
Transformation of  
Relational  
Expressions

Equivalence Rules

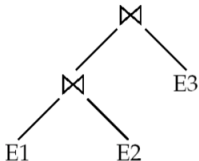
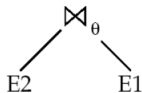
Example

Plan Generation

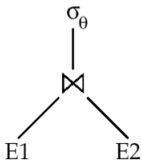
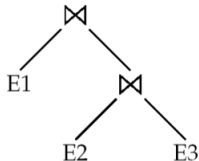
Module Summary



Rule 5

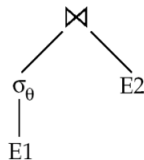


Rule 6a



Rule 7a

If  $\theta$  only has  
attributes from E1





# Equivalence Rules (4)

## Module 57

Partha Pratim  
Das

Objectives &  
Outline

Query  
Optimization

Equivalent  
Expressions

Evaluation Plan  
Cost

Transformation of  
Relational  
Expressions

Equivalence Rules

Example

Plan Generation

Module Summary

7 The selection operation distributes over the theta join operation under the following two conditions:

- When all the attributes in  $\theta_0$  involve only the attributes of one of the expressions ( $E_1$ ) being joined

$$\sigma_{\theta_0}(E_1 \bowtie_{\theta} E_2) = (\sigma_{\theta_0}(E_1)) \bowtie_{\theta} E_2$$

- When  $\theta_1$  involves only the attributes of  $E_1$  and  $\theta_2$  involves only the attributes of  $E_2$ .

$$\sigma_{\theta_1 \wedge \theta_2}(E_1 \bowtie_{\theta} E_2) = (\sigma_{\theta_1}(E_1)) \bowtie_{\theta} (\sigma_{\theta_2}(E_2))$$



# Equivalence Rules (5)

## Module 57

Partha Pratim  
Das

Objectives &  
Outline

Query  
Optimization

Equivalent  
Expressions

Evaluation Plan  
Cost

Transformation of  
Relational  
Expressions

Equivalence Rules

Example

Plan Generation

Module Summary

8 The projection operation distributes over the theta join operation as follows:

a. if  $\theta$  involves only attributes from  $L_1 \cup L_2$ :

$$\Pi_{L_1 \cup L_2}(E_1 \bowtie_{\theta} E_2) = \Pi_{L_1}(E_1) \bowtie_{\theta} \Pi_{L_2}(E_2)$$

b. Consider a join  $E_1 \bowtie_{\theta} E_2$

- Let  $L_1$  and  $L_2$  be sets of attributes from  $E_1$  and  $E_2$ , respectively
- Let  $L_3$  be attributes of  $E_1$  that are involved in join condition  $\theta$ , but are not in  $L_1 \cup L_2$ , and
- Let  $L_4$  be attributes of  $E_2$  that are involved in join condition  $\theta$ , but are not in  $L_1 \cup L_2$ .

$$\Pi_{L_1 \cup L_2}(E_1 \bowtie_{\theta} E_2) = \Pi_{L_1 \cup L_2}((\Pi_{L_1 \cup L_3}(E_1)) \bowtie_{\theta} (\Pi_{L_2 \cup L_4}(E_2)))$$



# Equivalence Rules (6)

## Module 57

Partha Pratim  
Das

Objectives &  
Outline

Query  
Optimization

Equivalent  
Expressions

Evaluation Plan  
Cost

Transformation of  
Relational  
Expressions

Equivalence Rules

Example

Plan Generation

Module Summary

9 The set operations union and intersection are commutative.

$$E_1 \cup E_2 = E_2 \cup E_1$$

$$E_1 \cap E_2 = E_2 \cap E_1$$

- (set difference is not commutative).

10 Set union and intersection are associative.

- $(E_1 \cup E_2) \cup E_3 = E_1 \cup (E_2 \cup E_3)$

- $(E_1 \cap E_2) \cap E_3 = E_1 \cap (E_2 \cap E_3)$

11 The selection operation distributes over  $\cup, \cap, -$

$$\sigma_{\theta}(E_1 - E_2) = \sigma_{\theta}(E_1) - \sigma_{\theta}(E_2)$$

and similarly for  $\cup$  and  $\cap$  in place of  $-$

Also:  $\sigma_{\theta}(E_1 - E_2) = \sigma_{\theta}(E_1) - E_2$

and similarly for  $\cap$  in place of  $-$ , but not for  $\cup$

12 The projection operation distributes over union

$$\pi_L(E_1 \cup E_2) = (\pi_L(E_1)) \cup (\pi_L(E_2))$$





# Exercise

## Module 57

Partha Pratim  
Das

Objectives &  
Outline

Query  
Optimization

Equivalent  
Expressions

Evaluation Plan  
Cost

Transformation of  
Relational  
Expressions

**Equivalence Rules**

Example

Plan Generation

Module Summary

- Create equivalence rules involving
  - The group by/aggregation operation
  - Left outer join operation



# Transformation Example: Pushing Selections

## Module 57

Partha Pratim  
DasObjectives &  
OutlineQuery  
OptimizationEquivalent  
ExpressionsEvaluation Plan  
CostTransformation of  
Relational  
Expressions

Equivalence Rules

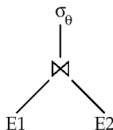
Example

Plan Generation

Module Summary

- Query: Find the names of all instructors in the Music department, along with the titles of the courses that they teach
  - $\pi_{name, title}(\sigma_{dept\_name = "Music"}(instructor \bowtie (teaches \bowtie \pi_{course\_id, title}(course))))$
- Transformation using rule 7a
  - $\pi_{name, title}((\sigma_{dept\_name = "Music"}(instructor)) \bowtie (teaches \bowtie \pi_{course\_id, title}(course)))$
- Performing the selection as early as possible reduces the size of the relation to be joined

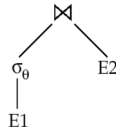
$course(course\_id, title, dept\ name, credits)$   
 $instructor(ID, name, dept\ name, salary)$   
 $teaches(ID, course\_id, sec\_id, semester, year)$



Rule 7a

↔

If  $\theta$  only has  
attributes from  $E1$





# Multiple Transformations

## Module 57

Partha Pratim  
DasObjectives &  
Outline

Query

Optimization

Equivalent

Expressions

Evaluation Plan

Cost

Transformation of

Relational

Expressions

Equivalence Rules

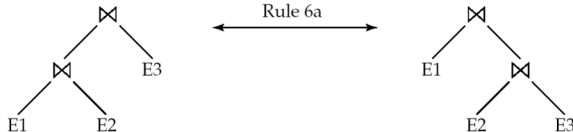
Example

Plan Generation

Module Summary

- Query: Find the names of all instructors in the Music department who have taught a course in 2009, along with the titles of the courses that they taught
  - $\pi_{name, title}(\sigma_{dept\_name="Music" \wedge year=2009}(instructor \bowtie (teaches \bowtie \pi_{course\_id, title}(course))))$
- Transformation using join associatively (Rule 6a):
  - $\pi_{name, title}(\sigma_{dept\_name="Music" \wedge year=2009}((instructor \bowtie teaches) \bowtie \pi_{course\_id, title}(course)))$
- Second form provides an opportunity to apply the “perform selections early” rule, resulting in the subexpression
  - $\sigma_{dept\_name="Music"}(instructor) \bowtie \sigma_{year=2009}(teaches)$

$course(course\_id, title, dept\_name, credits)$   
 $instructor(ID, name, dept\_name, salary)$   
 $teaches(ID, course\_id, sec\_id, semester, year)$





# Multiple Transformations (2)

## Module 57

Partha Pratim  
Das

Objectives &  
Outline

Query  
Optimization

Equivalent  
Expressions

Evaluation Plan  
Cost

Transformation of  
Relational  
Expressions

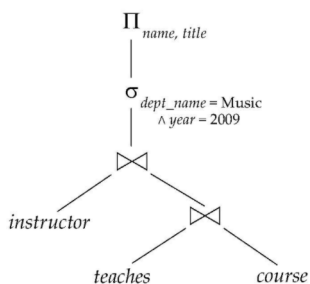
Equivalence Rules

Example

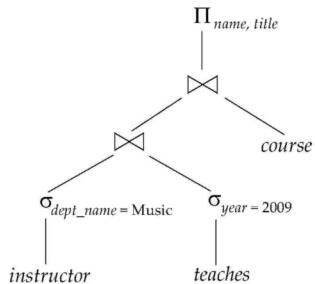
Plan Generation

Module Summary

*course*(course id, title, dept name, credits)  
*instructor*(ID, name, dept name, salary)  
*teaches*(ID, course id, sec id, semester, year)



(a) Initial expression tree



(b) Tree after multiple transformations



# Transformation Example: Pushing Projections

## Module 57

Partha Pratim  
DasObjectives &  
OutlineQuery  
OptimizationEquivalent  
ExpressionsEvaluation Plan  
CostTransformation of  
Relational  
Expressions

Equivalence Rules

Example

Plan Generation

Module Summary

- Consider:

$$\pi_{name, title}((\sigma_{dept\_name="Music"}(instructor)) \bowtie (teaches \bowtie \pi_{course\_id, title}(course)))$$

- When we compute

$$\sigma_{dept\_name="Music"}(instructor \bowtie teaches)$$

we obtain a relation whose schema is:

$$(ID, name, dept\_name, salary, course\_id, sec\_id, semester, year)$$

- Push projections using equivalence rules 8a and 8b; eliminate unneeded attributes from intermediate results to get:

$$\pi_{name, title}(\pi_{name, course\_id}(\sigma_{dept\_name="Music"}(instructor) \bowtie teaches)) \bowtie \pi_{course\_id, title}(course)$$

- Performing the projection as early as possible reduces the size of the relation to be joined

$course(course\_id, title, dept\_name, credits)$ $instructor(ID, name, dept\_name, salary)$ $teaches(ID, course\_id, sec\_id, semester, year)$
--

$$\begin{aligned} \Pi_{L_1 \cup L_2}(E_1 \bowtie_{\theta} E_2) &= \Pi_{L_1}(E_1) \bowtie_{\theta} \Pi_{L_2}(E_2) \\ \Pi_{L_1 \cup L_2}(E_1 \bowtie_{\theta} E_2) &= \Pi_{L_1 \cup L_2}((\Pi_{L_1 \cup L_3}(E_1)) \bowtie_{\theta} (\Pi_{L_2 \cup L_4}(E_2))) \end{aligned}$$



# Join Ordering Example

## Module 57

Partha Pratim  
Das

Objectives &  
Outline

Query  
Optimization

Equivalent  
Expressions

Evaluation Plan  
Cost

Transformation of  
Relational  
Expressions

Equivalence Rules

**Example**

Plan Generation

Module Summary

- For all relations  $r_1$ ,  $r_2$ , and  $r_3$ ,  
 $(r_1 \bowtie r_2) \bowtie r_3 = r_1 \bowtie (r_2 \bowtie r_3)$   
(Join Associativity)
- If  $r_2 \bowtie r_3$  is quite large and  $r_1 \bowtie r_2$  is small, we choose  
 $(r_1 \bowtie r_2) \bowtie r_3$   
so that we compute and store a smaller temporary relation



# Join Ordering Example (2)

## Module 57

Partha Pratim  
Das

Objectives &  
Outline

Query  
Optimization

Equivalent  
Expressions

Evaluation Plan  
Cost

Transformation of  
Relational  
Expressions

Equivalence Rules

Example

Plan Generation

Module Summary

- Consider the expression
$$\pi_{name, title}(\sigma_{dept\_name="Music"}(instructor) \bowtie teaches) \bowtie \pi_{course\_id, title}(course)$$
- Could compute  $teaches \bowtie \pi_{course\_id, title}(course)$  first, and join result with  $\sigma_{dept\_name="Music"}(instructor)$  but the result of the first join is likely to be a large relation
- Only a small fraction of the university's instructors are likely to be from the Music department
  - it is better to compute
$$\sigma_{dept\_name="Music"}(instructor) \bowtie (teaches)$$
 first

$course(course\ id, title, dept\ name, credits)$ $instructor(ID, name, dept\ name, salary)$ $teaches(ID, course\ id, sec\ id, semester, year)$
--



# Enumeration of Equivalent Expressions

## Module 57

Partha Pratim  
Das

Objectives &  
Outline

Query  
Optimization

Equivalent  
Expressions

Evaluation Plan  
Cost

Transformation of  
Relational  
Expressions

Equivalence Rules

Example

Plan Generation

Module Summary

- Query optimizers use equivalence rules to **systematically** generate expressions equivalent to the given expression
- Can generate all equivalent expressions as follows:
  - Repeat
    - ▷ apply all applicable equivalence rules on every subexpression of every equivalent expression found so far
    - ▷ add newly generated expressions to the set of equivalent expressionsUntil no new equivalent expressions are generated above
- The above approach is very expensive in space and time
  - Two approaches
    - ▷ Optimized plan generation based on transformation rules
    - ▷ Special case approach for queries with only selections, projections and joins





# Implementing Transformation Based Optimization

## Module 57

Partha Pratim  
Das

Objectives &  
Outline

Query  
Optimization

Equivalent  
Expressions

Evaluation Plan  
Cost

Transformation of  
Relational  
Expressions

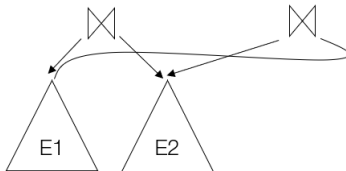
Equivalence Rules

Example

Plan Generation

Module Summary

- Space requirements reduced by sharing common sub-expressions:
  - when E1 is generated from E2 by an equivalence rule, usually only the top level of the two are different, subtrees below are the same and can be shared using pointers
    - ▷ E.g. when applying join commutativity



- Same sub-expression may get generated multiple times
    - ▷ Detect duplicate sub-expressions and share one copy
- Time requirements are reduced by not generating all expressions
  - Dynamic programming



# Module Summary

## Module 57

Partha Pratim  
Das

Objectives &  
Outline

Query  
Optimization

Equivalent  
Expressions

Evaluation Plan  
Cost

Transformation of  
Relational  
Expressions

Equivalence Rules

Example

Plan Generation

Module Summary

- Understood the basic issues for optimizing queries
- For every relational expression, usually there are a number of equivalent expressions that can be created by simple transformations
- Final execution plan can be created by choose the estimated least cost expression from the alternates

**Slides used in this presentation are borrowed from <http://db-book.com/> with kind permission of the authors.**

**Edited and new slides are marked with “PPD”.**