

Query Optimization

CPS 116

Introduction to Database Systems

Announcements (November 18)

◆ Homework #4 due in one week!

Query optimization

◆ One logical plan → “best” physical plan

◆ Questions

- How to enumerate possible plans
- How to estimate costs
- How to pick the “best” one

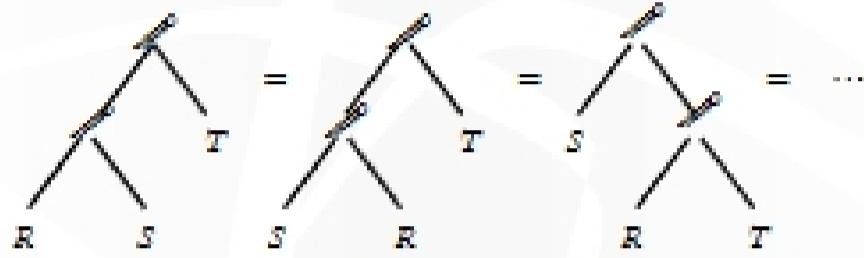
◆ Often the goal is not getting the optimum plan, but instead avoiding the horrible ones

Any of these will do

1 second 1 minute 1 hour

Plan enumeration in relational algebra ⁴

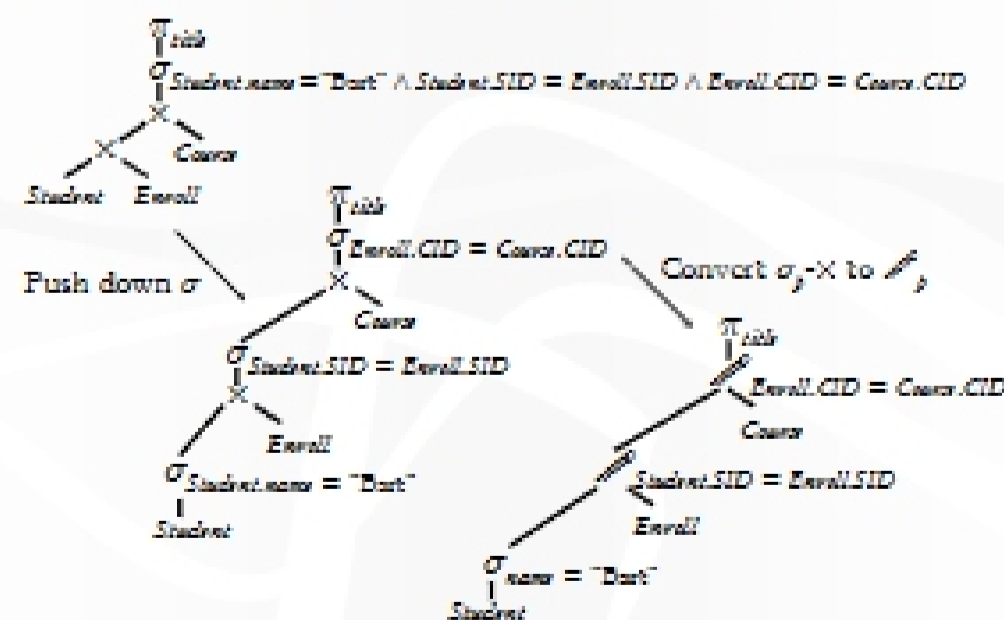
- ◆ Apply relational algebra equivalences
- ◆ Join reordering: \times and \bowtie are associative and commutative (except column ordering, but that is unimportant)



More relational algebra equivalences ⁵

- ◆ Convert $\sigma_p \times$ to/from \bowtie_p : $\sigma_p(R \times S) = R \bowtie_p S$
- ◆ Merge/split σ 's: $\sigma_{p1}(\sigma_{p2} R) = \sigma_{p1 \wedge p2} R$
- ◆ Merge/split π 's: $\pi_{L1}(\pi_{L2} R) = \pi_{L1} R$, where $L1 \subseteq L2$
- ◆ Push down/pull up σ :
 - $\sigma_{p \wedge p'}(R \bowtie_p S) = (\sigma_{p'} R) \bowtie_{p \wedge p'} (\sigma_p S)$, where
 - p is a predicate involving only R columns
 - p' is a predicate involving only S columns
 - p and p' are predicates involving both R and S columns
- ◆ Push down π : $\pi_L(\sigma_p R) = \pi_L(\sigma_p(\pi_{L'} R))$, where
 - L' is the set of columns referenced by p that are not in L
- ◆ Many more (seemingly trivial) equivalences...
 - Can be systematically used to transform a plan to new ones

Relational query rewrite example ⁶



Heuristics-based query optimization ⁷

- ❖ Start with a logical plan
- ❖ Push selections/projections down as much as possible
 - Why?
 - Why not?
- ❖ Join smaller relations first, and avoid cross product
 - Why?
 - Why not?
- ❖ Convert the transformed logical plan to a physical plan (by choosing appropriate physical operators)

SQL query rewrite ⁸

- ❖ More complicated—subqueries and views divide a query into nested “blocks”
 - Processing each block separately forces particular join methods and join order
 - Even if the plan is optimal for each block, it may not be optimal for the entire query
- ❖ Unnest query: convert subqueries/views to joins
- ❖ We can just deal with select-project-join queries
 - Where the clean rules of relational algebra apply

SQL query rewrite example ⁹

- ❖ `SELECT name
FROM Student
WHERE SID = ANY (SELECT SID FROM Enroll);`
- ❖ `SELECT name
FROM Student, Enroll
WHERE Student.SID = Enroll.SID;`
- ❖ `SELECT name
FROM (SELECT DISTINCT Student.SID, name
FROM Student, Enroll
WHERE Student.SID = Enroll.SID);`
