Query Optimization

1. Conjunction of SELECT operations → cascade of selects
   $\sigma_{c1} AND \sigma_{c2} AND \cdots$
   $= \sigma_{c1}(\sigma_{c2}(\sigma_{c3}(\cdots)))$

2. Commutativity of SELECT
   $\sigma_{c1}(\sigma_{c2}(R)) = \sigma_{c2}(\sigma_{c1}(R))$

3. Cascade of Projections operations: only last projection step counts
   $\pi_x(\pi_y(\pi_z(\cdots))) = \pi_x(\cdots)$

4. Commutativity of σ, π

5. $\bowtie$ and X are commutative

6. Commutativity of σ with $\bowtie$ and X when σ condition applies to a single table involved in the $\bowtie$ or X
   $\sigma_{A[x]=\cdots} = (A \times B) = (\sigma_{A[x]=\cdots}(A) \times B)$

7. Commutativity of π with $\bowtie$ and X when join (or X) only involved attributes in projection

8. Associativity of $\bowtie, X$

Heuristic Algorithm for Optimizing Query Tree

1. Break up any conjunction of select conditions into a cascade

2.

Transactions

- Problem: multiple users accessing/modifying db at the same time
- Basic model:
  - Read an object
  - Write an object (e.g. blocks, rows, tables)

Concurrency Problems

1. Lost updates

2. Dirty read

Improve Model:

- Begin transaction
- Read/write
- Commit
- Rollback
- Goal: transaction treated as atomic unit of work (”all or nothing”)
- Successful commit →