

# Query Optimization

## Query Tree

Consider the following schema and query:

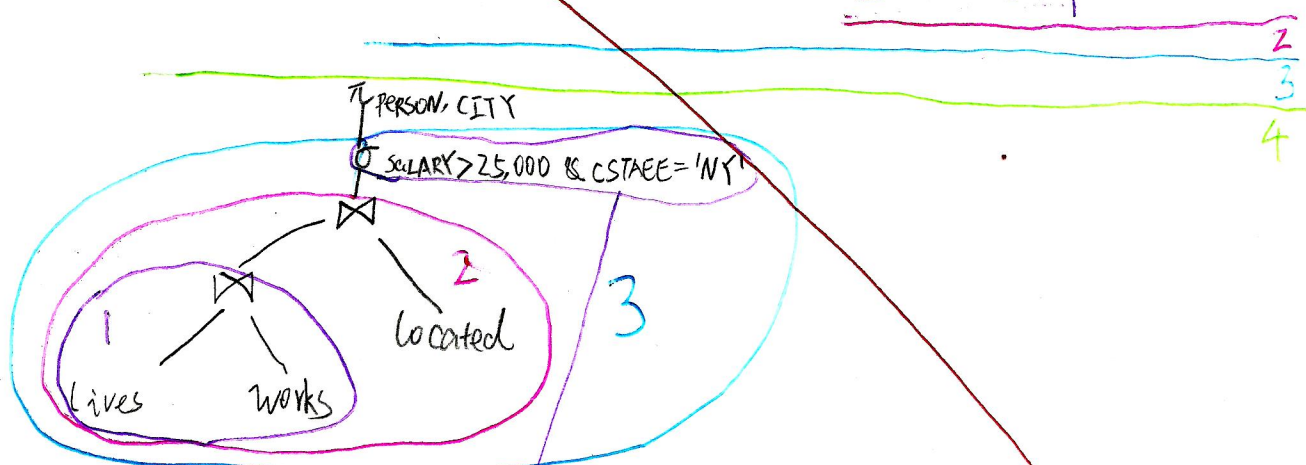
lives (PERSON, STREET, CITY, STATE, ZIP)

works (PERSON, COMPANY, SALARY, POSITION)

located (COMPANY, CCITY, CSTATE, CZIP)

Find the name and city of all people who earn more than 25,000 and work for a company located in the state of NY.

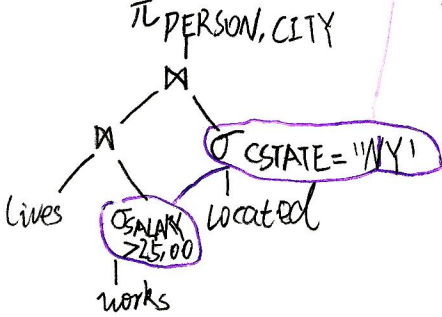
$\pi_{\text{PERSON, CITY}} \sigma_{\text{SALARY} > 25,000 \wedge \text{CSTATE} = \text{'NY'}} ((\text{lives} \bowtie \text{works}) \bowtie \text{located})$



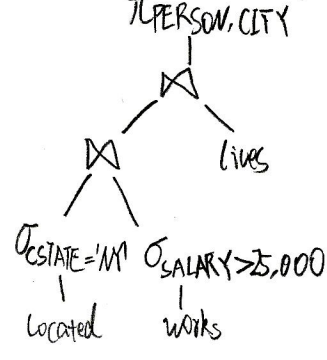
## HEURISTIC ALGEBRAIC OPTIMIZATION ALGORITHM

1. Cascade selection (rule 1)
2. Move selection as far down the tree as possible (rules 2, 4, 6, 10)
3. Rearrange leaf node to get smaller intermediate relations (rule 9)
4. Combine a  $\times$  with  $\sigma$  to yield  $\bowtie$ , if possible
5. Cascade projections and push down the tree (rules 3, 4, 7, 11)
6. Identify common subexpressions.

T<sub>1</sub>: Cascade selections and push down tree



T<sub>2</sub>: Using commutativity & associativity properties of join rearrange join order.



T<sub>3</sub>: Introduce projections

