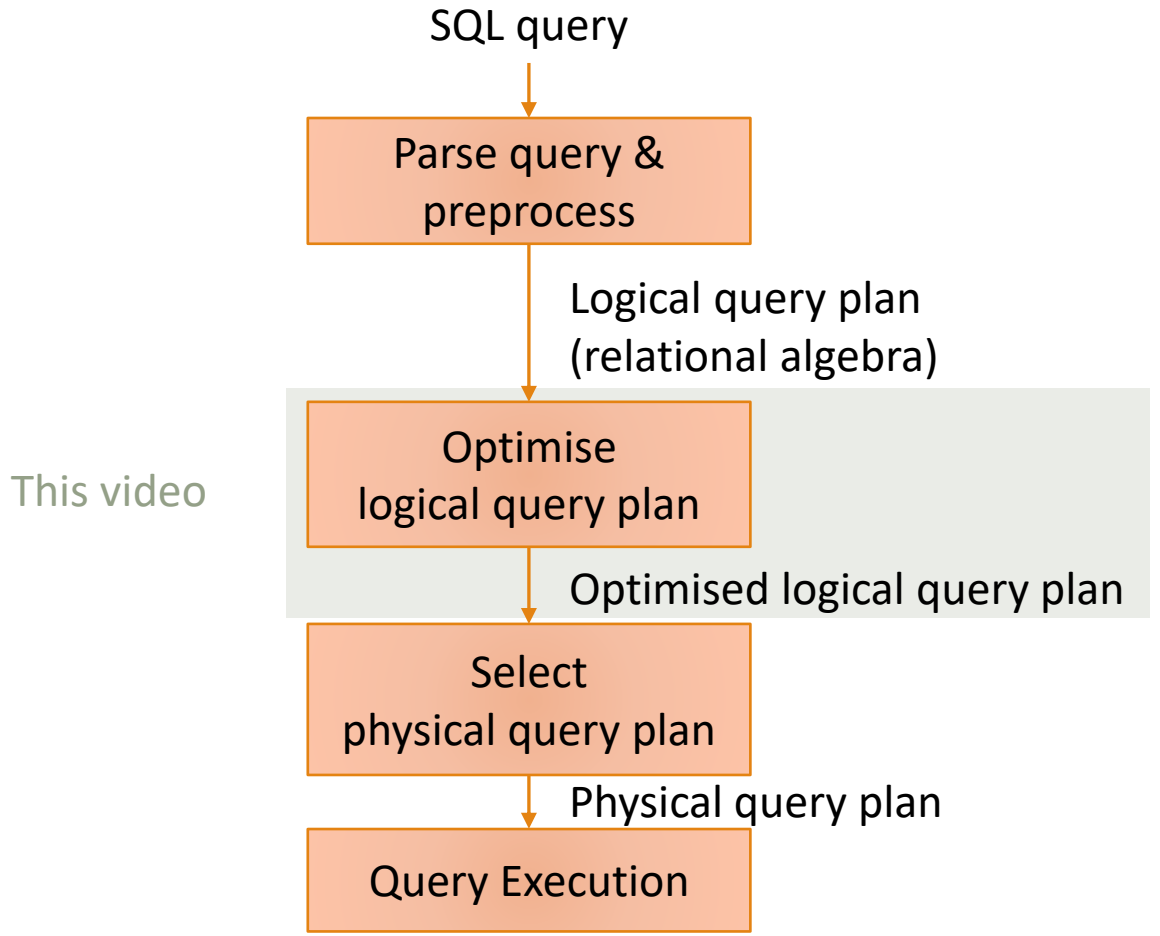
Optimising query plans

Overview of this video

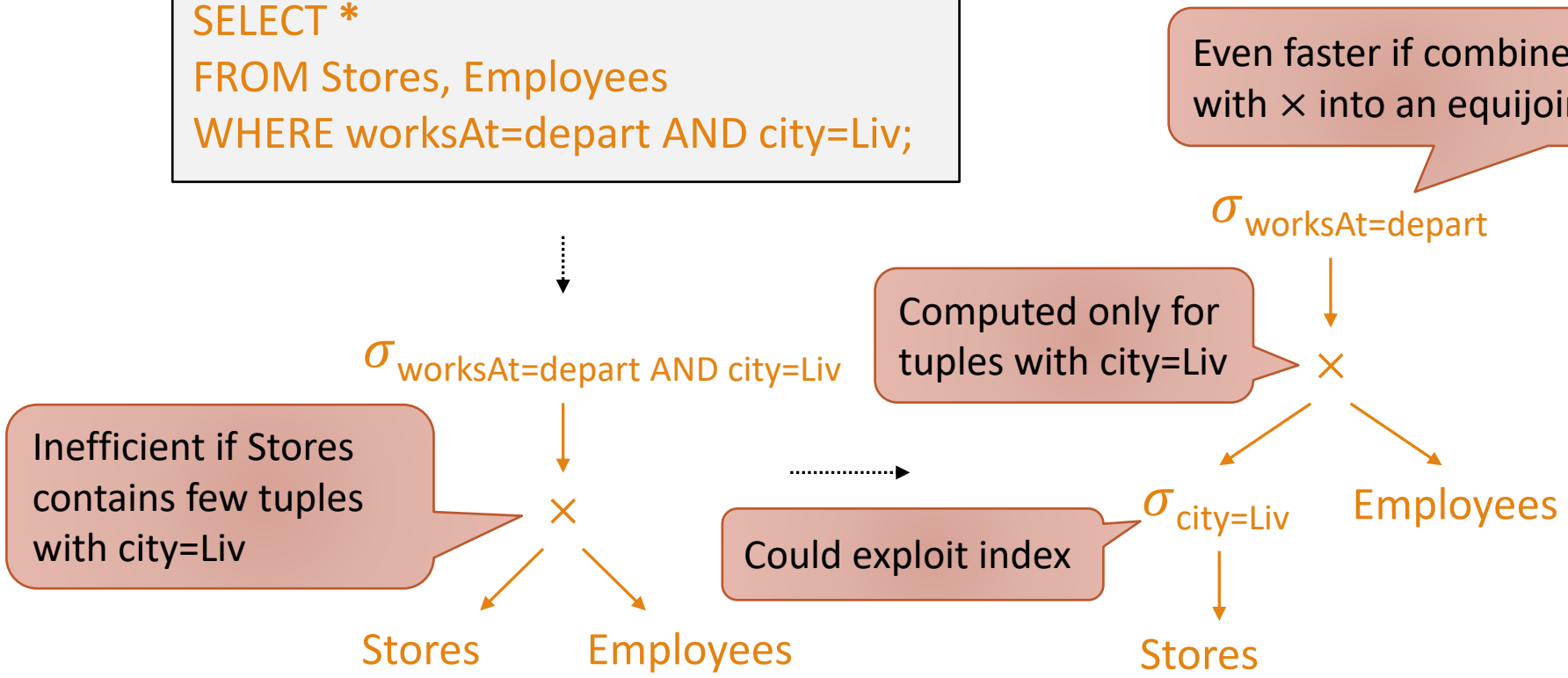


How to optimise the initial query plan?

Why Optimise Query Plans?

The initial query plan might not be the optimal one

```
SELECT *
FROM Stores, Employees
WHERE worksAt=depart AND city=Liv;
```



How to Optimise?

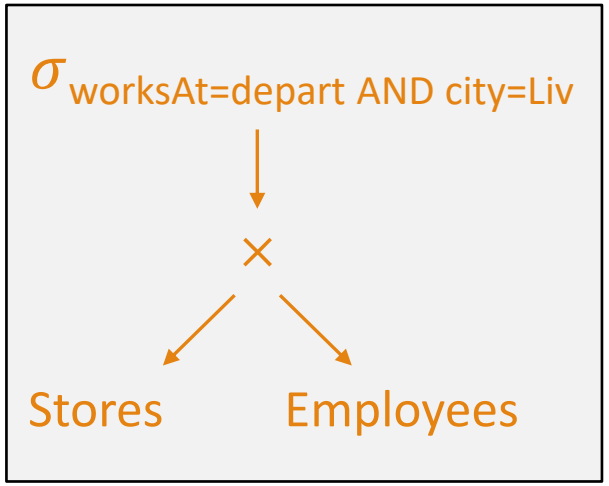
Evaluation of query plans:

- Bottom-up
- Efficiency depends on size of intermediate results

Rewrite the initial query plan so that intermediate results will be smaller

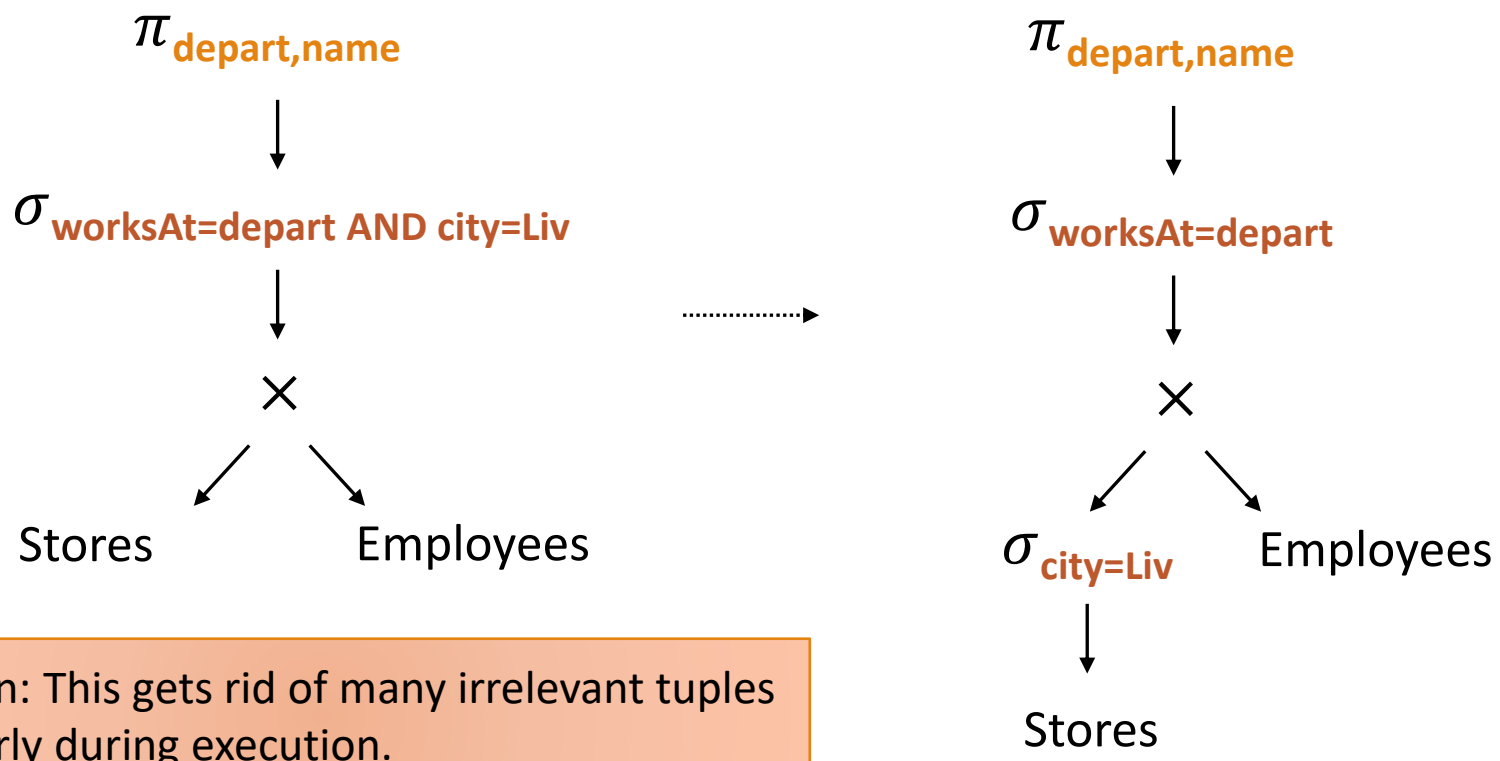
Based on equivalence laws of relational algebra

- Many laws (see references for this lecture on Vital)
- Examples:
  - $\sigma_{A=a \text{ AND } B=b}(R) = \sigma_{A=a}(\sigma_{B=b}(R))$
  - $\sigma_{A=a}(\mathbf{R} \times S) = \sigma_{A=a}(\mathbf{R}) \times S$ — If A is on R
  - $\sigma_{A=B}(R \times S) = R \bowtie_{A=B} S$ — If A is on R and B is on S



Heuristics

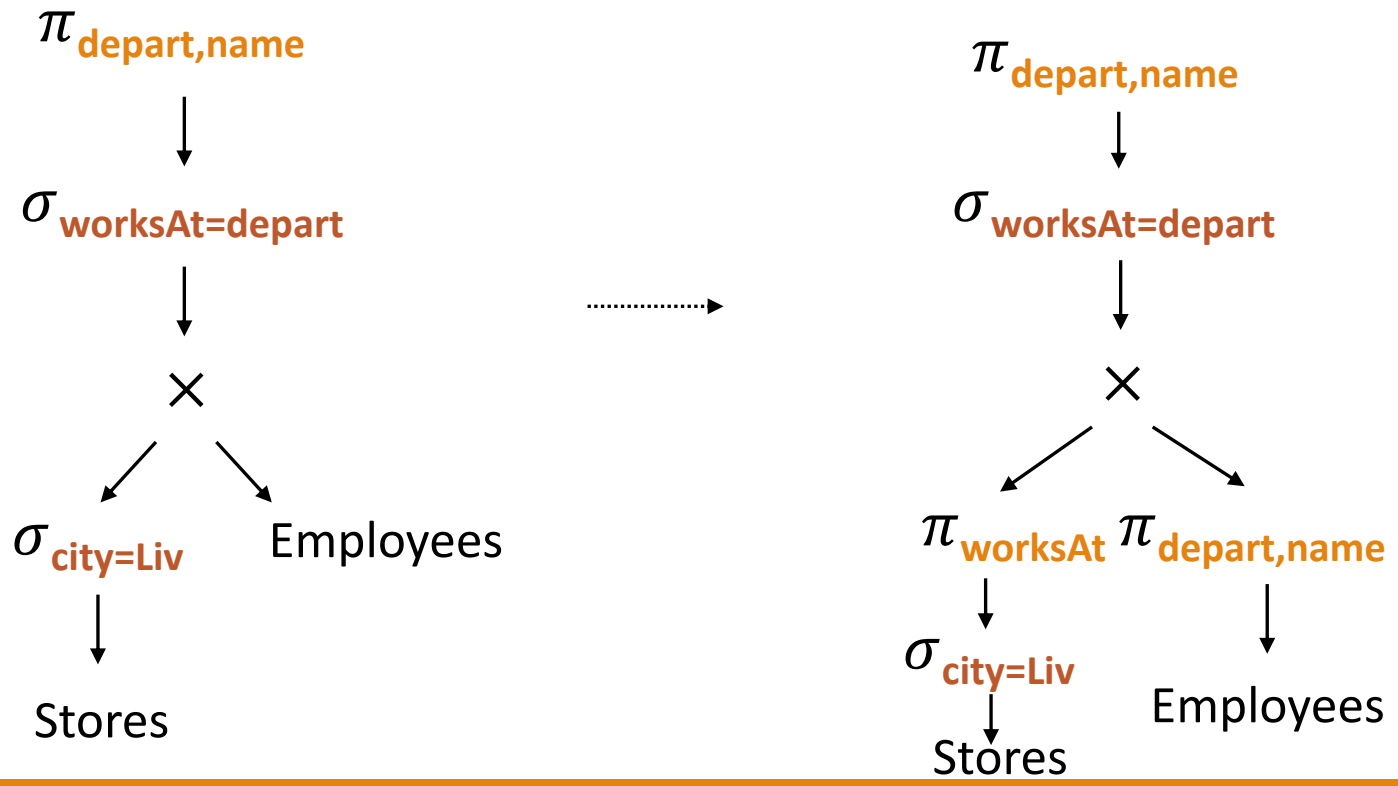
#1: Push selections as far down the tree as possible



Intuition: This gets rid of many irrelevant tuples very early during execution.

Heuristics

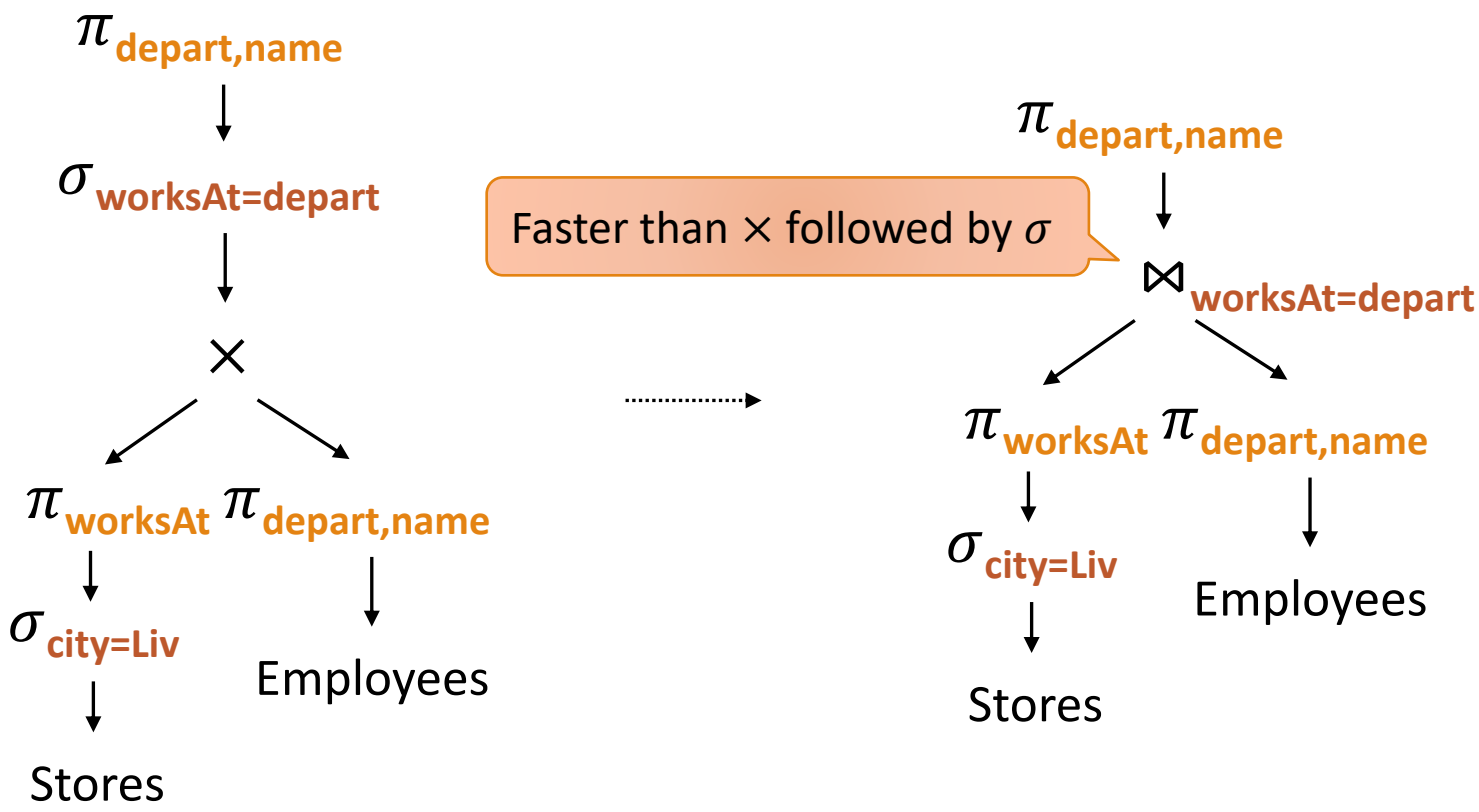
#2: "Push" projections as far down the tree as possible, or insert projections where appropriate



Many more heuristics…

Heuristics

#3: If possible, introduce equijoins for × followed by σ



Summary

Optimisation of query plans at the logical level

- Started with initial query plan
- Applied equivalence laws of relational algebra to optimise the plan