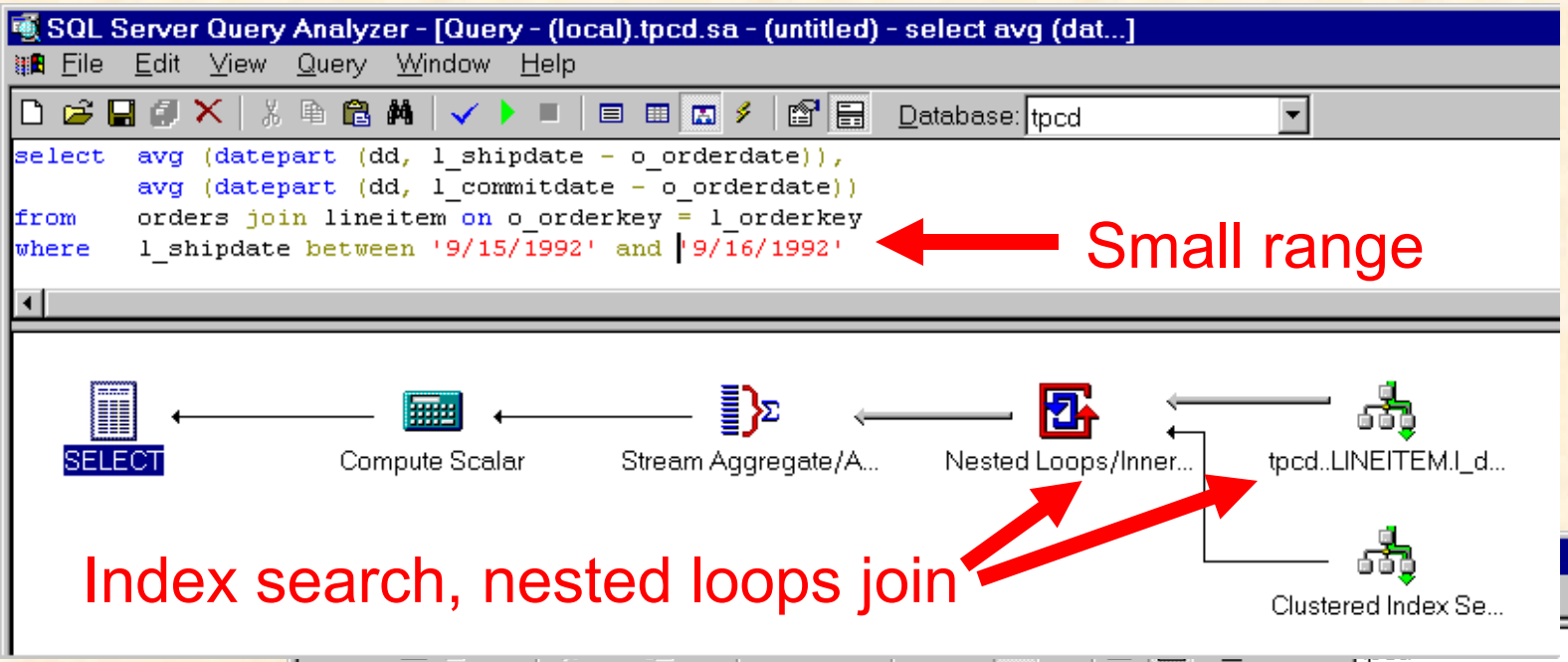
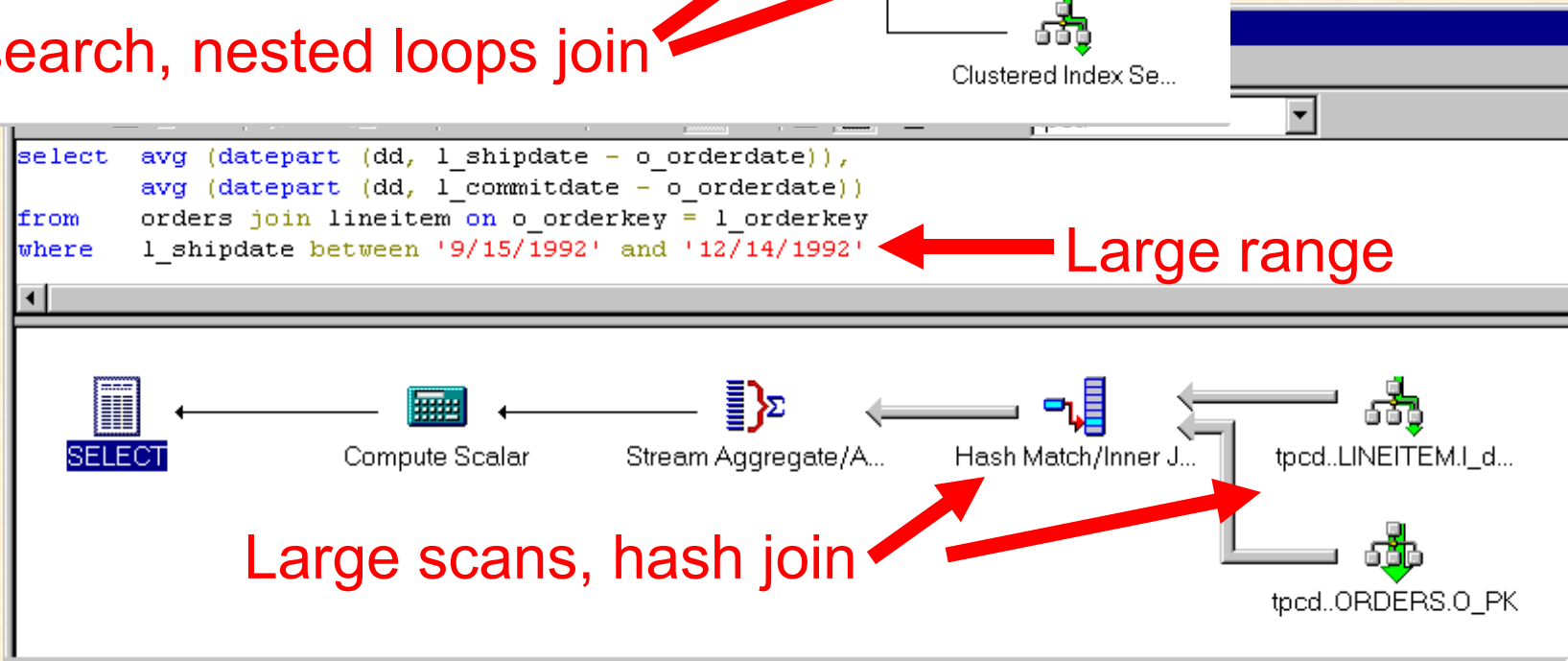
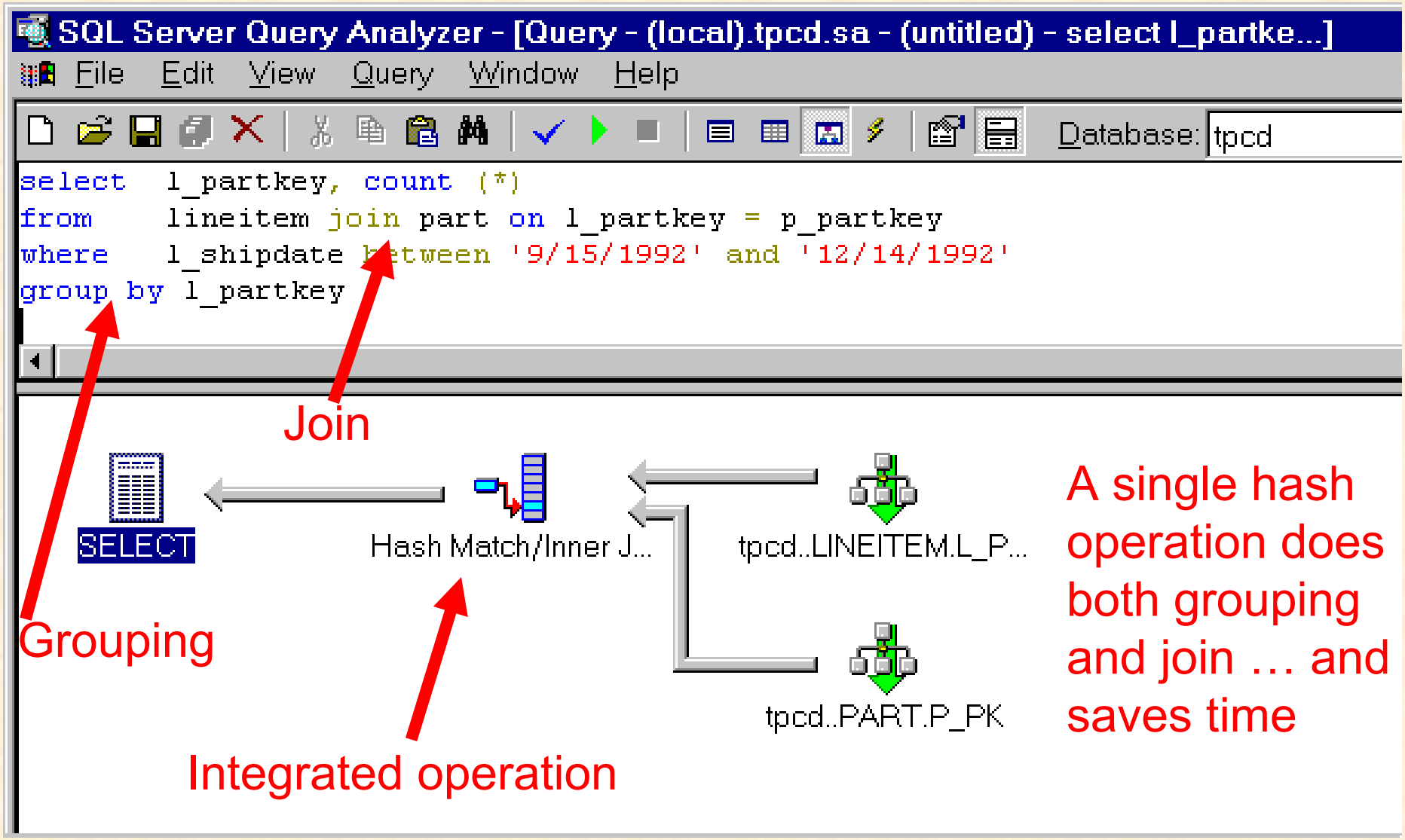
Need for cost-based optimization

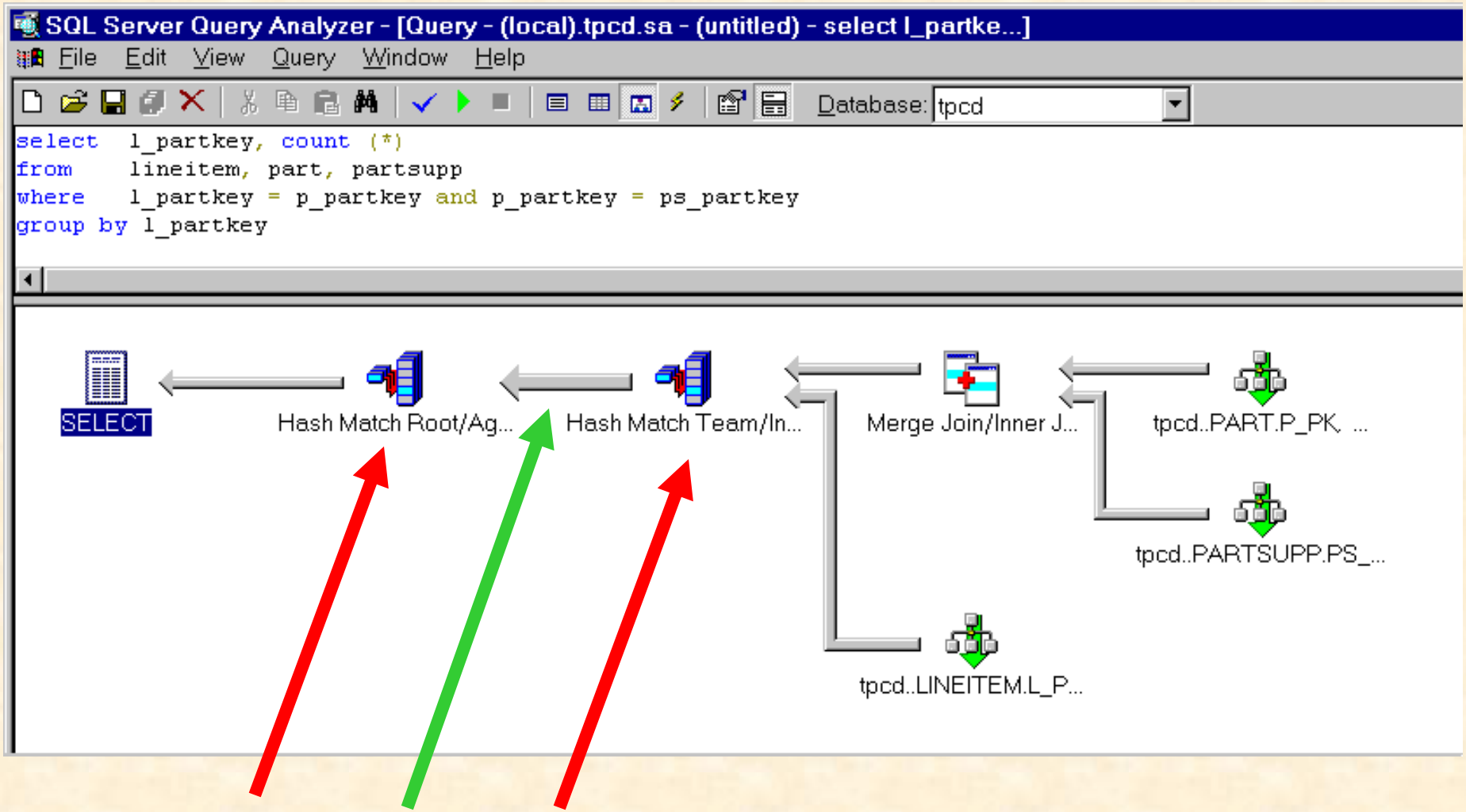




Integrated hashing operation

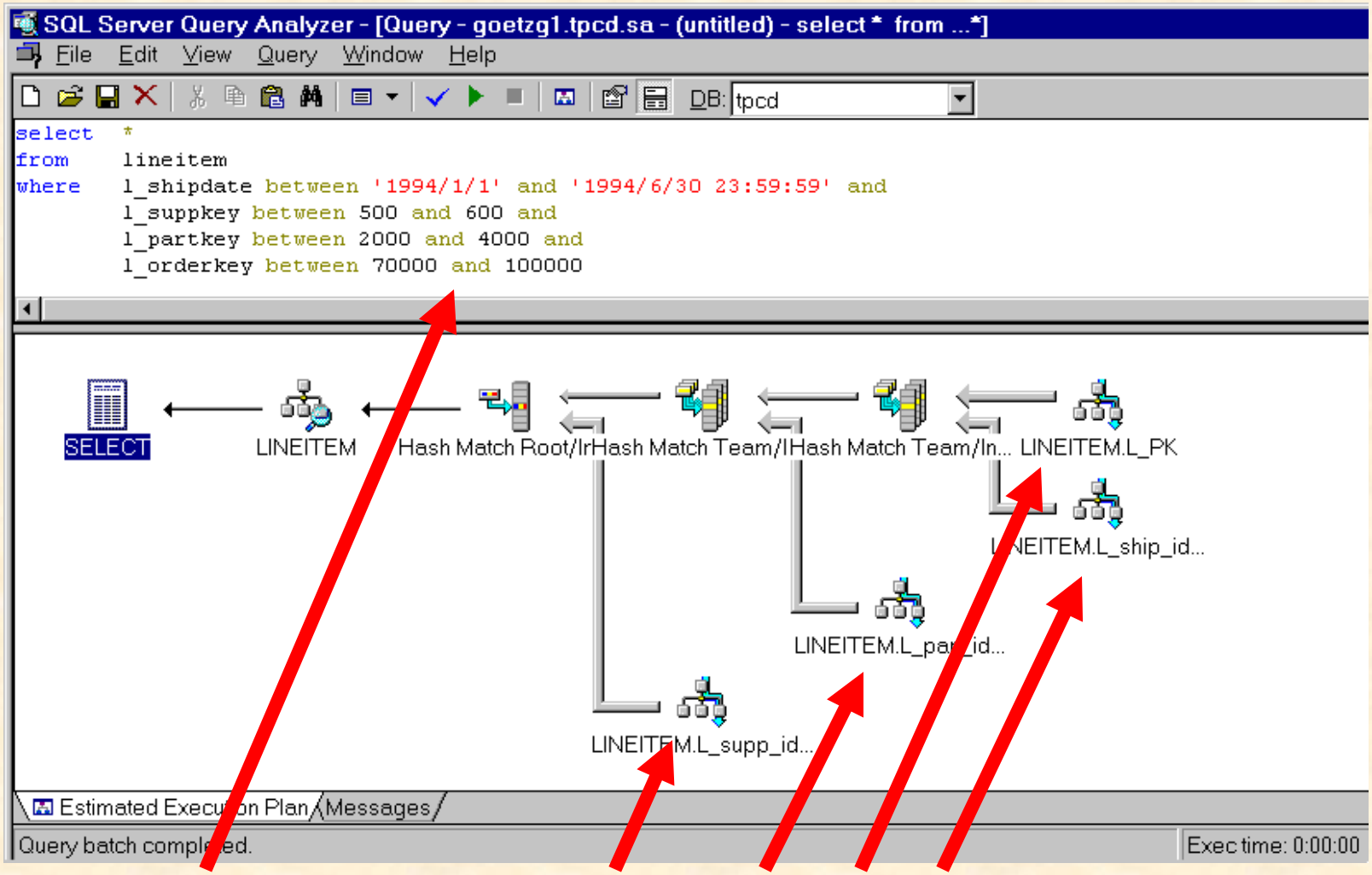


Hash team root & member



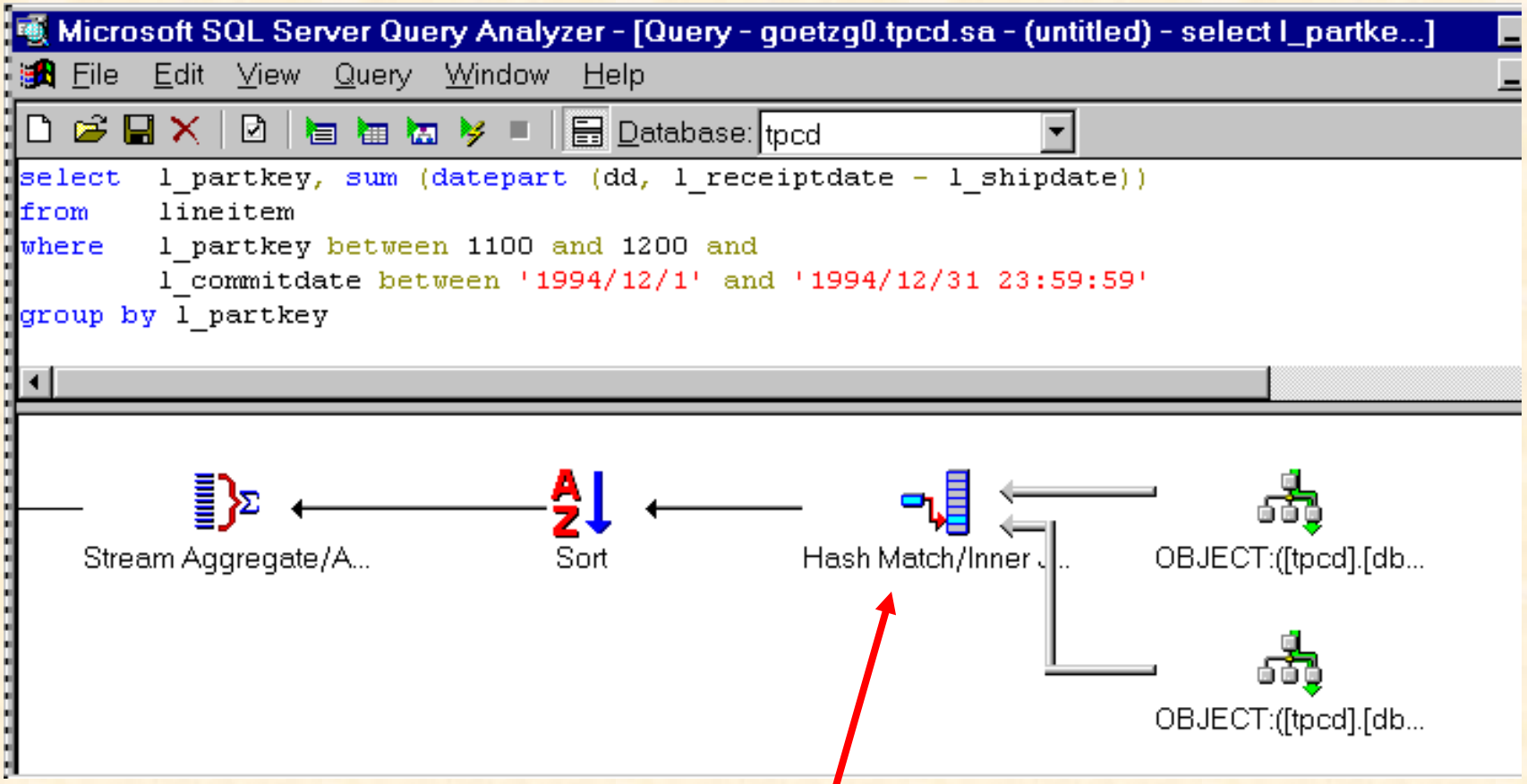
No overflow files or I/O costs for intermediate result due to team

Index intersection



One table, four predicates, four indexes exploited

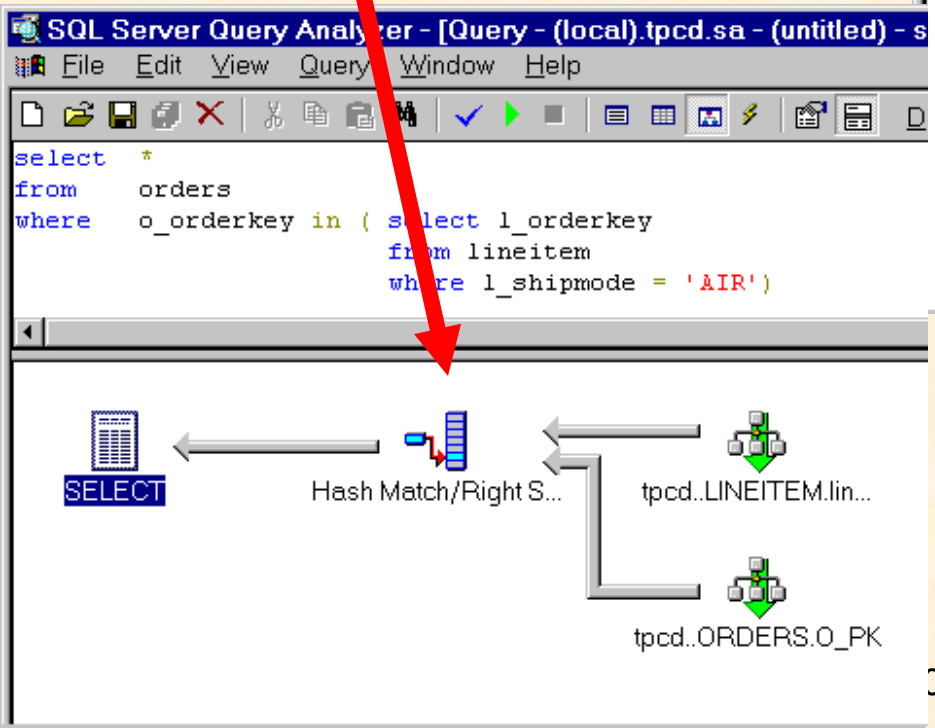
Multiple indexes covering a query

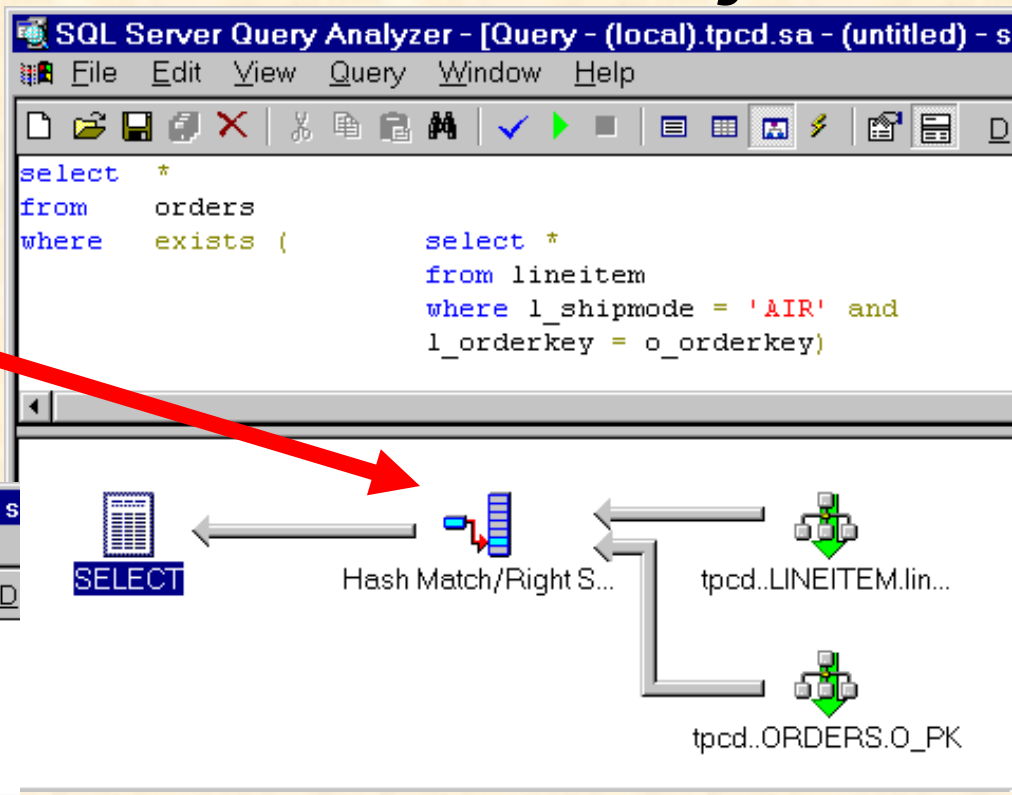


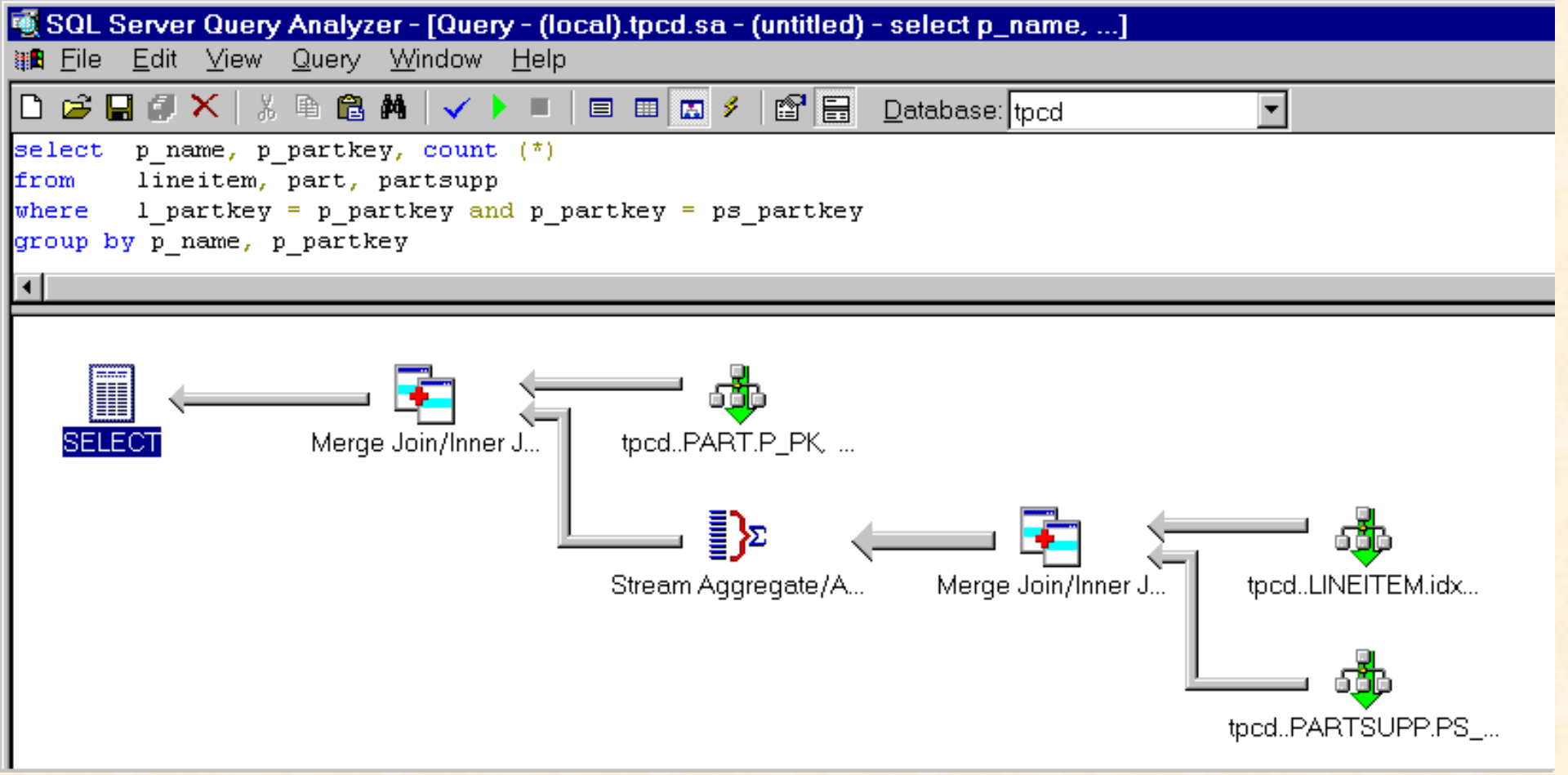
After joining two indexes of one table, all required columns are present – expensive record fetching is avoided

Nested query becomes semi-join

Exploit join algorithms designed for large inputs

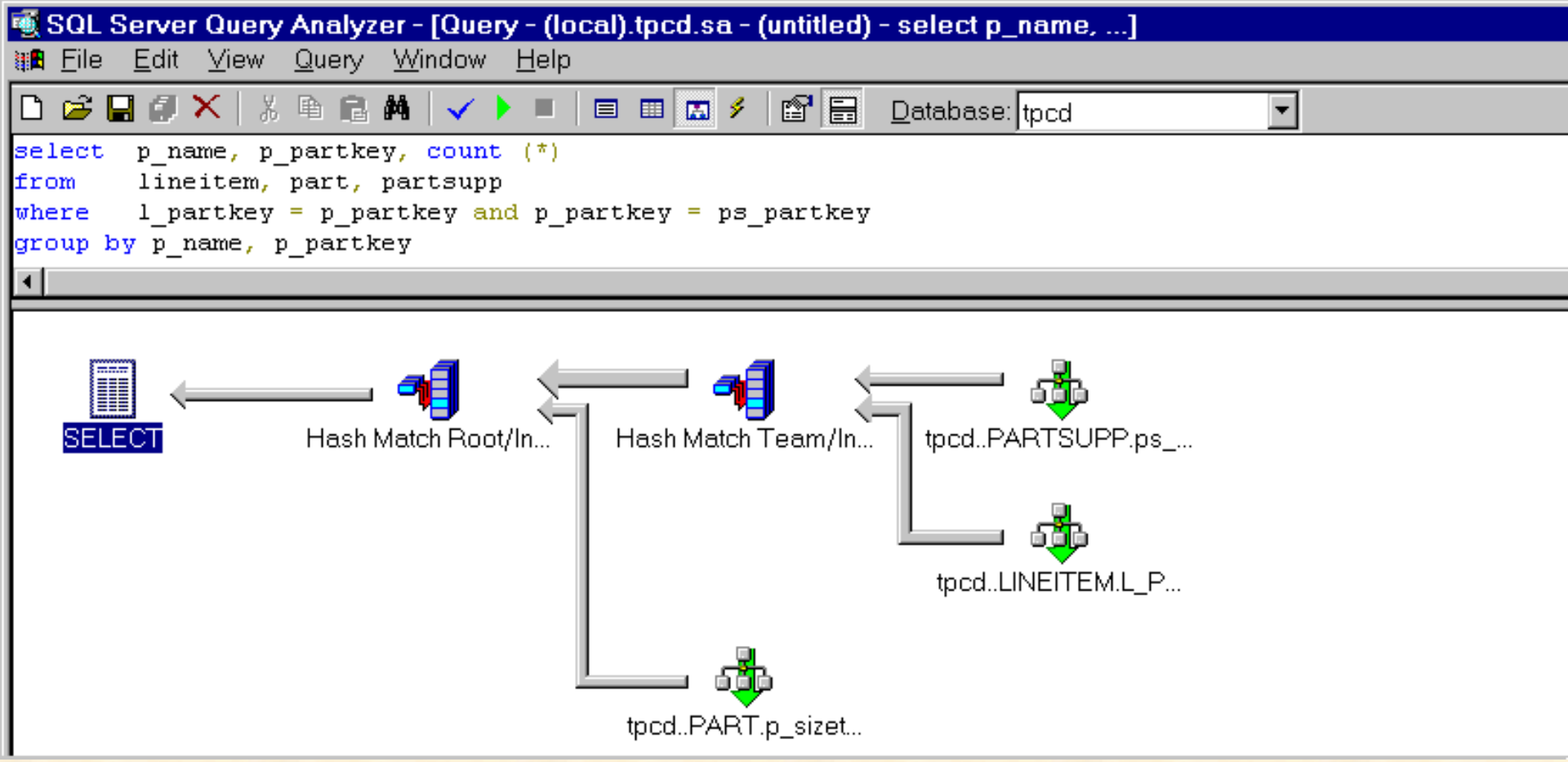






Multiple optimization techniques are needed to find this plan

- Join clause inferred between line item & part supply
- Group-by list reduced by functional dependencies
- Grouping (on alternative column) pushed down through join
- “Interesting orderings” between scans, joins, grouping

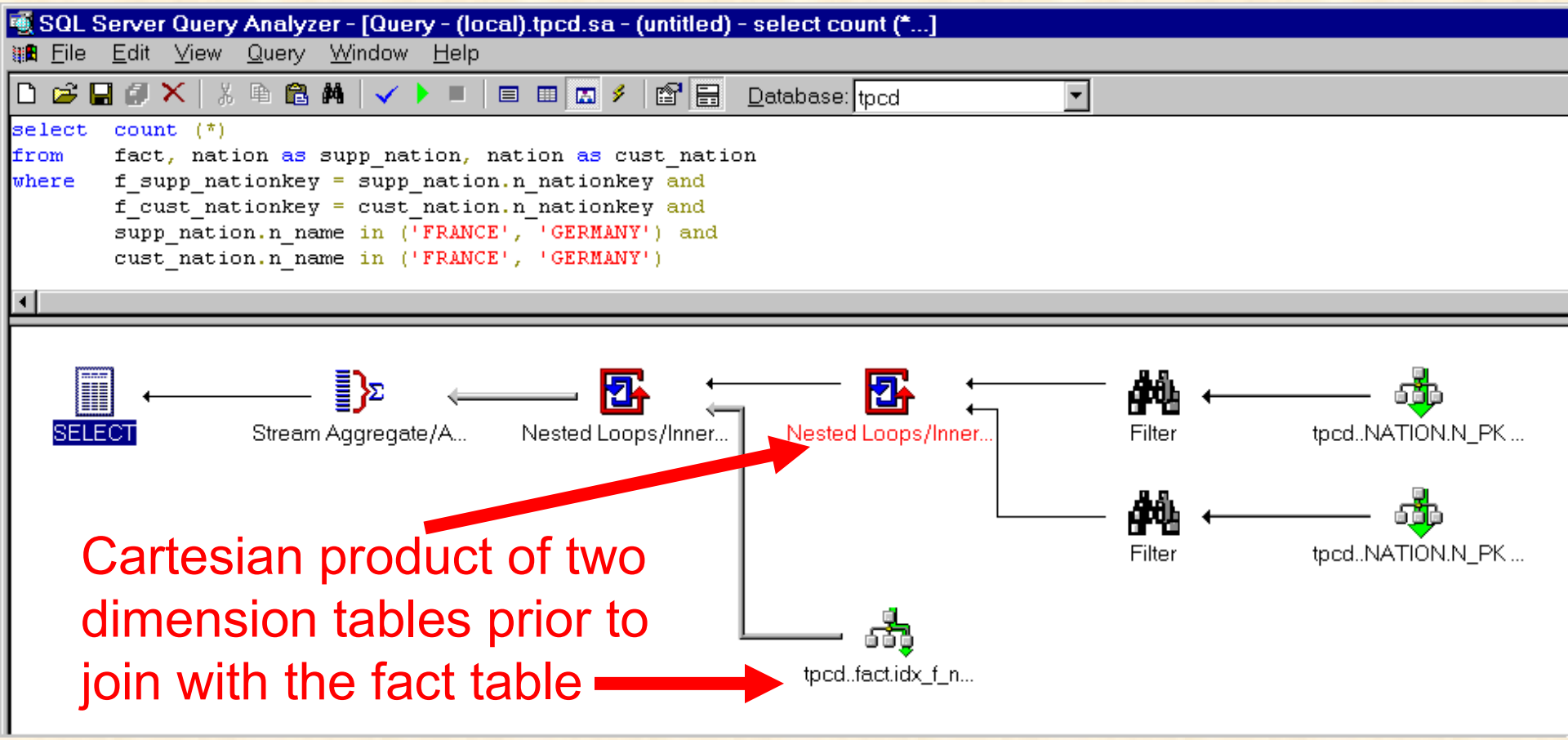


Multiple optimization techniques in a hash-based plan

Same as previous example, *plus*

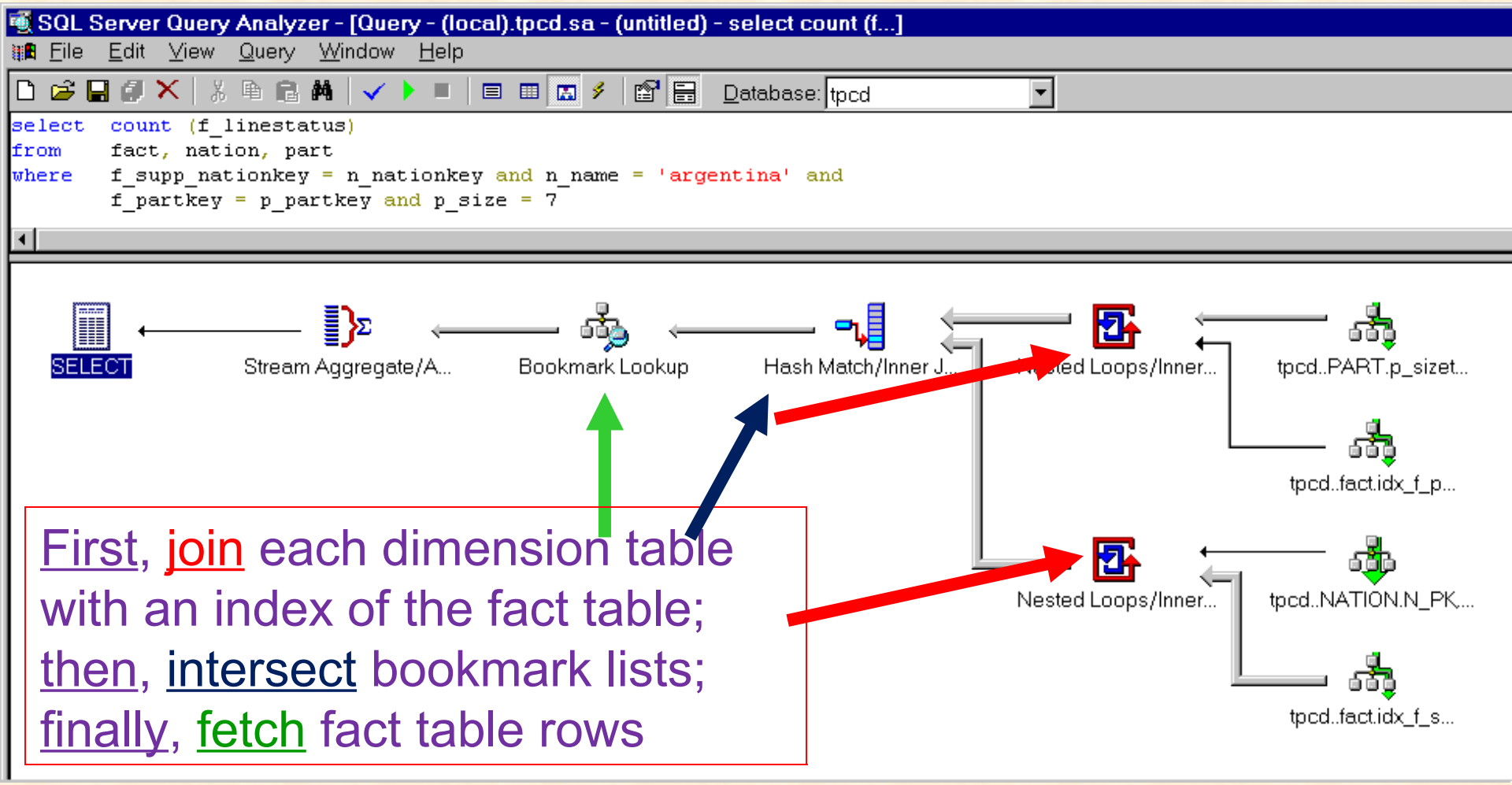
- Integrated hash operation …
- … within a hash team
- Disk-order scans

Star joins: Cartesian products



Let the large fact table participate in fewer joins – reduce star join cost without any query plan hints!

Star joins: semi-join reduction



All star join technologies now also in SQL Server!