Effects of Join Order

\(|A| = |B| = |C| = 1,000\)

Plan 1:

\[
\text{sel}_{C \times B} := \frac{|C \times B|}{|C| \times |B|} = \frac{100,000}{1,000 \times 1,000} = 0.1
\]

Plan 2:

\[
\text{sel}_{A \times B} := \frac{|A \times C|}{|A| \times |C|} = \frac{100}{1,000 \times 1,000} = 0.0001
\]

Plan 1: Top-level join has to process 1,000 + 100,000 tuples.

Plan 2: Top-level join has to process 100 + 1,000 tuples.
Effects of a Scan Access Path

\[ \prod_{text} \]
\[ \forall_{A.id=B.dz} \]
\[ \prod_{text, id} B \]
\[ \sigma_{A.name='Hugo'} \]

\[ A \]

scan on A

\[ \prod_{text} \]
\[ \forall_{A.id=B.dz} \]
\[ \prod_{text, id} B \]
\[ \sigma_{A.name='Hugo'} \]

\[ A \]
Effects of an Index Access Path

\[ \prod_{\text{title}} \\exists_{A.id=B.dz} \prod_{\text{title, id}} B \sigma_{A.name='Hugo'} \]

index on A.name

\[ \prod_{\text{title}} \\exists_{A.id=B.dz} \prod_{\text{title, id}} B \sigma_{A.name='Hugo'} \]
Which Index Access Path Exactly?

clustered index on A.name

\[ \prod_{\text{title}} \left( A.\text{id}=B.dz \right) \prod_{\text{title, id}} B \sigma_{A.\text{name}='Hugo'} \]

unclustered index on A.name

\[ \prod_{\text{title}} \left( A.\text{id}=B.dz \right) \prod_{\text{title, id}} B \sigma_{A.\text{name}='Hugo'} \]

covering index on A.name, A.title, A.id

\[ \prod_{\text{title}} \left( A.\text{id}=B.dz \right) \prod_{\text{title, id}} B \sigma_{A.\text{name}='Hugo'} \rightarrow \text{index-only path} \]
Estimating Index Access Costs

**clustered index on A.name**

\[ \Pi_{\text{title}, \text{id}} \sigma_{A \text{.name}=\text{‘Hugo’}} \]

\[ \downarrow \text{seqeval} \]

expected costs:
- one random I/O to fetch leaf,
- one random I/O in store, then ISAM

**unclustered index on A.name**

\[ \Pi_{\text{title}, \text{id}} \sigma_{A \text{.name}=\text{‘Hugo’}} \]

expected costs:
- as clustered index,
- plus random I/O to store depending on selectivity

**covering index on A.name, A.title, A.id**

\[ \Pi_{\text{title}, \text{id}} \sigma_{A \text{.name}=\text{‘Hugo’}} \]

expected costs:
- one random I/O to fetch leaf,
- then ISAM
- no need to go to store
Estimating Scan vs Index Access Costs

scan on A

expected costs:
one random I/O to go to start of A,
then sequential read

unclustered index on A.name

expected costs:
as clustered index,
plus random I/O to store depending on selectivity