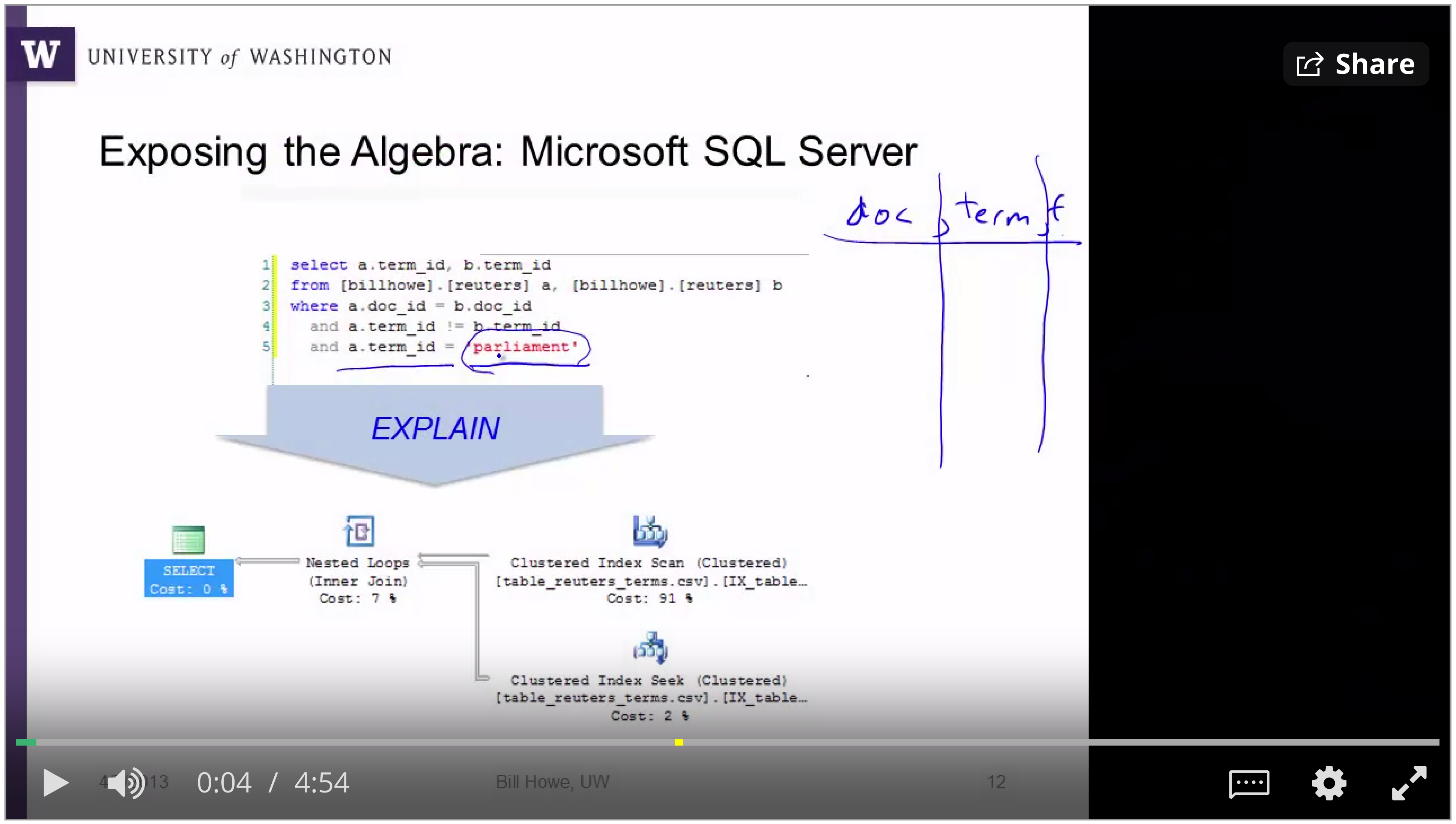
Optimization: Choosing Physical Plans



W UNIVERSITY of WASHINGTON

Exposing the Algebra: Microsoft SQL Server

```
select a.term_id, b.term_id
from [billhowe].[reuters] a, [billhowe].[reuters] b
where a.doc_id = b.doc_id
  and a.term_id != b.term_id
  and a.term_id = 'parliament'
```

doc
term
f
EXPLAIN
SELECT
Cost: 0 %
Nested Loops
(Inner Join)
Cost: 7 %
Clustered Index Scan (Clustered)
[table_reuters_terms.csv].[IX_table…
Cost: 91 %
Clustered Index Seek (Clustered)
[table_reuters_terms.csv].[IX_table…
Cost: 2 %
Bill Howe, UW
12
Share
0:04 / 4:54

Save Note Discuss Download Share

English Help Us Translate

0:00 [MUSIC] Now, what I want you to notice though is that when I explain this query, I get a different physical plan. The logical plan looks the same. It's still got scan, scan, and a join, but the algorithm to compute the join has changed. And now it's this thing called nested loops. And that nested loop corresponds exactly to this pseudocode here. That's why they they call it nested loops, the outer loop and the inner loop, so exactly the same thing. And so it chose to do this nested loops plan, even though we argued that it was an n-squared algorithm, and it probably wouldn't be chosen very often. So why was it in this case? So if you think about it, the one of the sides of this join is only dealing with those terms or occurrences of the term parliament in a document, which is a