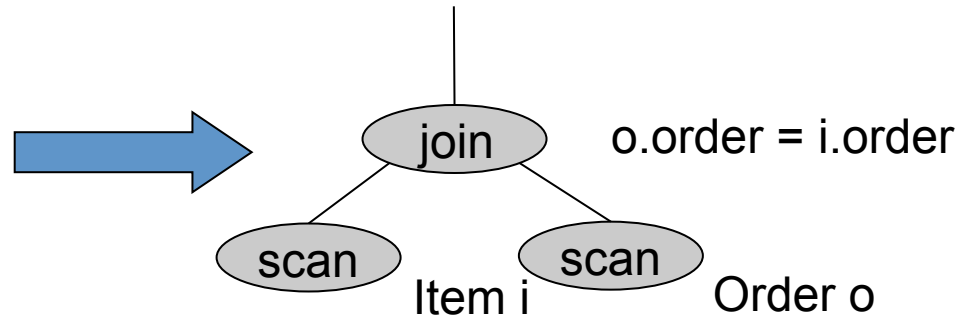


# Same logical expression, different physical algorithms

```
SELECT *  
  FROM Order o, Item i  
 WHERE o.order = i.order
```



*Which is faster?*

Option 1

```
for each record i in Item:  
  for each record o in Order:  
    if o.order = i.order:  
      return (r,s)
```

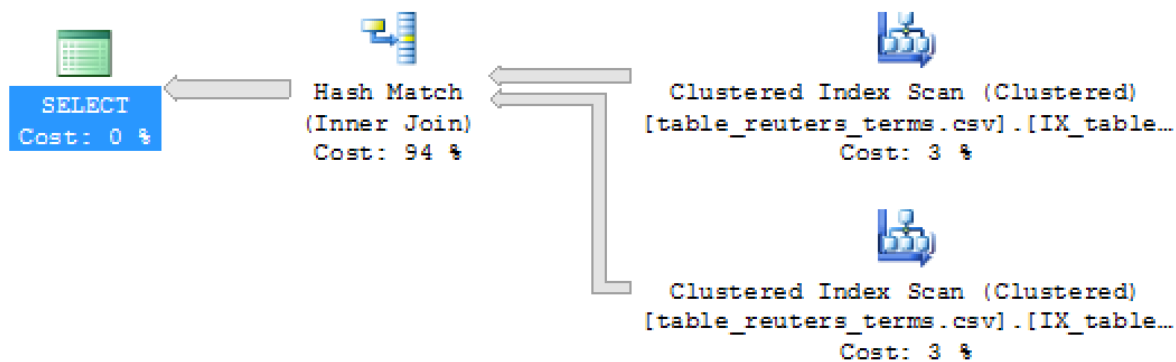
Option 2

```
for each record i in Item:  
  insert into hashtable  
  
for each record o in Order:  
  lookup corresponding records in hashtable  
  return matching pairs
```

# Exposing the Algebra: Microsoft SQL Server

```
1 select a.term_id, b.term_id
2 from [billhowe].[reuters] a, [billhowe].[reuters] b
3 where a.doc_id = b.doc_id
4      and a.term_id != b.term_id
```

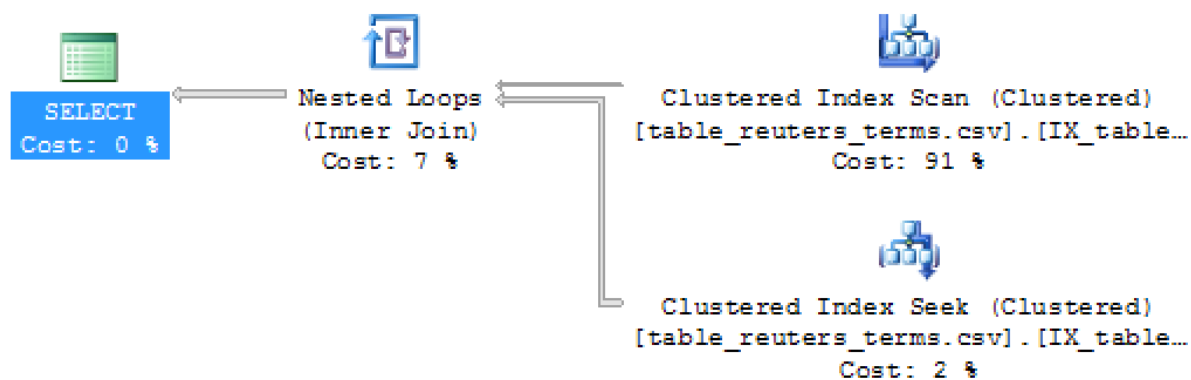
*EXPLAIN*



# Exposing the Algebra: Microsoft SQL Server

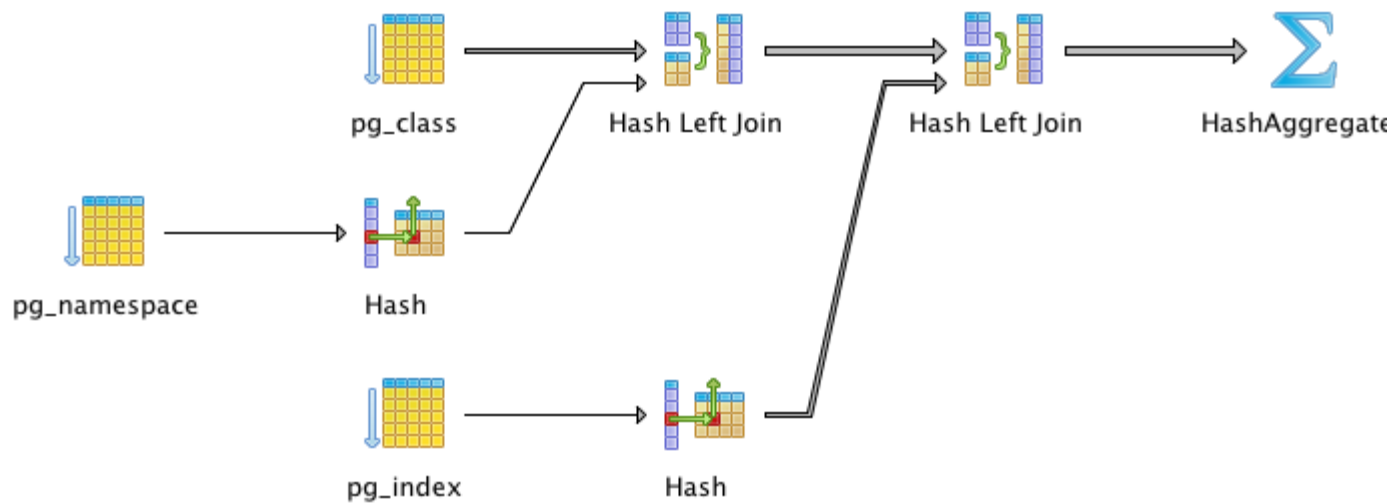
```
1 select a.term_id, b.term_id
2 from [billhowe].[reuters] a, [billhowe].[reuters] b
3 where a.doc_id = b.doc_id
4       and a.term_id != b.term_id
5       and a.term_id = 'parliament'
```

*EXPLAIN*



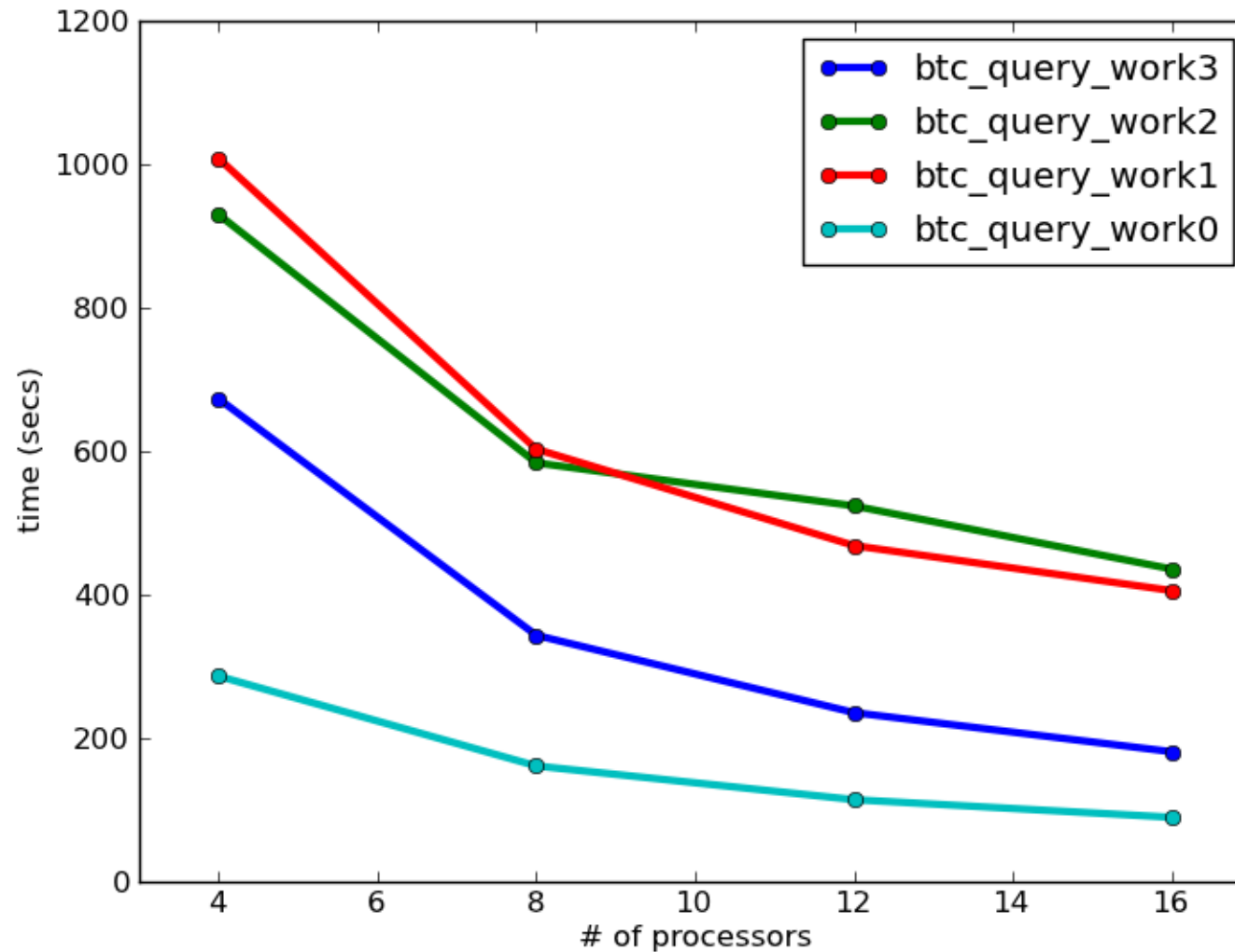
# Exposing the Algebra: PostgreSQL

EXPLAIN SELECT ....



*screenshot from pgAdmin3*

# Algebraic Optimization Matters



*BTC 2010 Dataset*

*3B quads*

*623 GB processed*

# Picasso Query Plan Diagrams

