Relational Query Processing

Walid Aref

Spring 2020

1

1

How Queries Get Processed

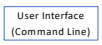
- Start from an SQL query:

2

2

How Queries Get Processed

- Start from an SQL query:
  - Select * from A, B, C Where
    - A.a1 = B.b1 and B.b2 = C.c2 and
    - A.a0 = "123"

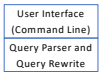


3

3

How Queries Get Processed

- Start from an SQL query:
  - Select * from A, B, C Where
    - A.a1 = B.b1 and B.b2 = C.c2 and
    - A.a0 = "123"
- Transform this query to its equivalent relational algebra expression:

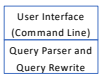


4

4

How Queries Get Processed

- Start from an SQL query:
  - Select * from A, B, C Where
    - A.a1 = B.b1 and B.b2 = C.c2 and
    - A.a0 = "123"
- Transform this query to an equivalent relational algebra expression:
  - $\sigma_{A.a0 = "123"}(A \bowtie_{A.a1 = B.b1} B \bowtie_{B.b2 = C.c2} C)$

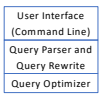


5

5

How Queries Get Processed

- Start from an SQL query:
  - Select * from A, B, C Where
    - A.a1 = B.b1 and B.b2 = C.c2 and
    - A.a0 = "123"
- Transform this query to an equivalent relational algebra expression:
  - $\sigma_{A.a0 = "123"}(A \bowtie_{A.a1 = B.b1} B \bowtie_{B.b2 = C.c2} C)$
- Optimize this expression:



6

6

How Queries Get Processed

- Start from an SQL query:
  - Select * from A, B, C Where
    - A.a1 = B.b1 and B.b2 = C.c2 and
    - A.a0 = "123"
- Transform this query to an equivalent relational algebra expression:
  - $\sigma_{A.a0 = "123"}(A \bowtie_{A.a1 = B.b1} B \bowtie_{B.b2 = C.c2} C)$
- Optimize this expression
  - $(\sigma_{A.a0 = "123"}(A) \bowtie_{A.a1 = B.b1} B) \bowtie_{B.b2 = C.c2} C)$

User Interface (Command Line)
Query Parser and Query Rewrite
Query Optimizer

7

7

How Queries Get Processed

- Start from an SQL query:
  - Select * from A, B, C Where
    - A.a1 = B.b1 and B.b2 = C.c2 and
    - A.a0 = "123"
- Transform this query to an equivalent relational algebra expression:
  - $\sigma_{A.a0 = "123"}(A \bowtie_{A.a1 = B.b1} B \bowtie_{B.b2 = C.c2} C)$
- Optimize this expression:
  - $(\sigma_{A.a0 = "123"}(A) \bowtie_{A.a1 = B.b1} B) \bowtie_{B.b2 = C.c2} C)$
- Generate an equivalent query evaluation plan:

User Interface (Command Line)
Query Parser and Query Rewrite
Query Optimizer

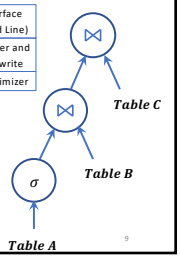
8

8

How Queries Get Processed

- Start from an SQL query:
  - Select * from A, B, C Where
    - A.a1 = B.b1 and B.b2 = C.c2 and
    - A.a0 = "123"
- Transform this query to an equivalent relational algebra expression:
  - $\sigma_{A.a0 = "123"}(A \bowtie_{A.a1 = B.b1} B \bowtie_{B.b2 = C.c2} C)$
- Optimize this expression:
  - $(\sigma_{A.a0 = "123"}(A) \bowtie_{A.a1 = B.b1} B) \bowtie_{B.b2 = C.c2} C)$
- Generate an equivalent query evaluation plan:

User Interface (Command Line)
Query Parser and Query Rewrite
Query Optimizer

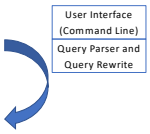


9

9

How Queries Get Processed

- Start from an SQL query:
  - Select * from A, B, C Where
    - A.a1 = B.b1 and B.b2 = C.c2 and
    - A.a0 = "123"
- Transform this query to an equivalent relational algebra expression:
  - $\sigma_{A.a0 = "123"}(A \bowtie_{A.a1 = B.b1} B \bowtie_{B.b2 = C.c2} C)$
- Question:
  - Is the above transformation always possible?



?

10

10

Conceptual Evaluation of SQL Queries into Relational Algebra Expressions

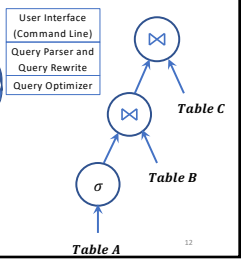
- Given an SQL query:
  - Select $attr_1, attr_2, \ldots, attr_n$
    - From $Table_1, Table_2, \ldots, Table_m$
    - Where $Pred_1$ and $Pred_2$ and . . . and $Pred_k$
- The equivalent Relational Algebra expression:
  - $\pi_{attr1, attr2, \ldots, attrn}(\sigma_{(Pred1 \wedge Pred2 \wedge \ldots \wedge Predk)}(Table_1 \times Table_2 \times \ldots \times Table_m))$

11

11

How Queries Get Processed

- Start from an SQL query:
  - Select * from A, B, C Where
    - A.a1 = B.b1 and B.b2 = C.c2 and
    - A.a0 = "123"
- Transform this query to an equivalent relational algebra expression:
  - $\sigma_{A.a0 = "123"}(A \bowtie_{A.a1 = B.b1} B \bowtie_{B.b2 = C.c2} C)$
- Optimize this expression:
  - $(\sigma_{A.a0 = "123"}(A) \bowtie_{A.a1 = B.b1} B) \bowtie_{B.b2 = C.c2} C)$
- Generate an equivalent query evaluation plan:

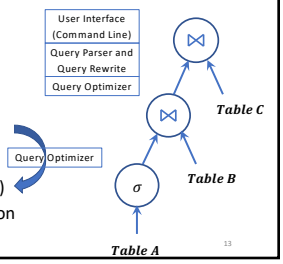


12

12

How Queries Get Processed

- Start from an SQL query:
  - Select * from A, B, C Where
    - A.a1 = B.b1 and B.b2 = C.c2 and
    - A.a0 = "123"
- Transform this query to an equivalent relational algebra expression:
  - $\sigma_{A.a0 = "123"}(A \bowtie_{A.a1 = B.b1} B \bowtie_{B.b2 = C.c2} C)$
- Optimize this expression:
  - $((\sigma_{A.a0 = "123"}(A) \bowtie_{A.a1 = B.b1} B) \bowtie_{B.b2 = C.c2} C)$
- Generate an equivalent query evaluation plan:

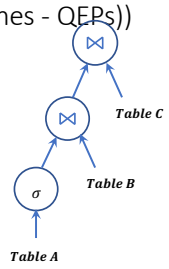


13

13

Query Evaluation Trees (or Query Evaluation Plans (Or Query Evaluation Pipelines - QEPs))

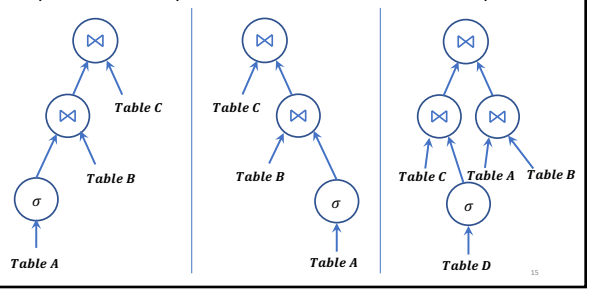
- Select * from A, B, C Where
  - A.a1 = B.b1 and B.b2 = C.c2 and
  - A.a0 = "123"
- Query Parser -> Query Rewrite -> Query Optimizer -> Query Evaluation Pipeline – Query Processor
- Questions:
  - How does a query evaluation pipeline work?
  - Why is it called "pipeline"?
  - What are the various shapes of query evaluation pipelines?
  - Does the shape of a QEP matter for query processing purposes?



14

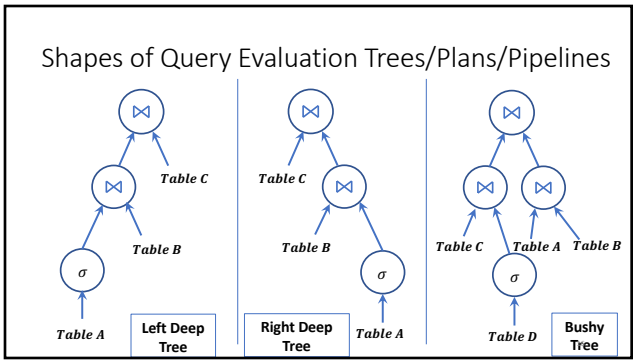
14

Shapes of Query Evaluation Trees/Plans/Pipelines



15

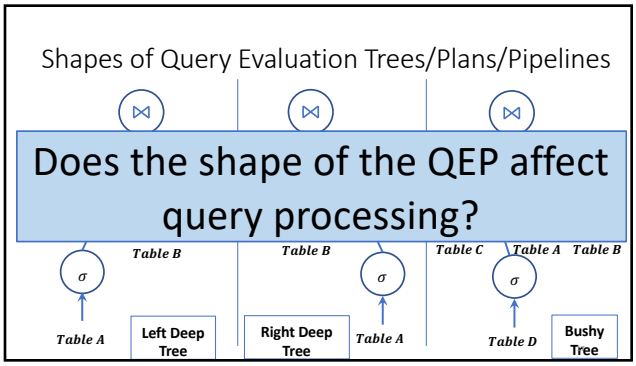
15



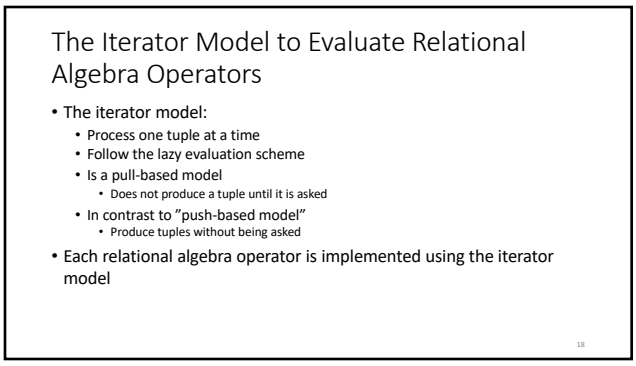
16

assume right-deep tree has a temporary table and temporary table has p tuples. Then the cost of IO is

c*p(c joins with temp table)+ p(save temporary table to disk) + ab(iterate all a,b tuples to construct temporary table)



17



18

How to Code an Operator in the Iterator Model?

- Each operator has the following interface methods:
  - Open():
    - Open the operator
    - Initialize the operator and its internal state variables
  - GetNext():
    - Start from the current state of the operator
    - Execute the operator from its current state until is able to retrieve the next tuple
    - Report the next tuple
    - Advance and store the current state for the next execution
    - If no tuples are available for return (e.g., reached end-of-table), report done/null/end-of-table
  - Close():
    - Close the operator
  - Other auxiliary operators:
    - HasNext(), Reset() or ReOpen(), among others

19

19

Physical Operators and their Realizations as Iterators

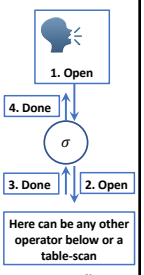
- The Table Scan Operator:
  - Open():
    - Open the table
    - Set the cursor at the beginning of the table
  - GetNext():
    - Get the next tuple from the table
    - Advance the cursor to the next tuple
    - If no tuple exists, then return null or end-of-table
    - Else return the tuple
  - Close():
    - Close the underlying table
  - Reset():
    - Reset the cursor to point to the beginning of the table

20

20

Physical Operators and their Realizations as Iterators: The Select Operator

- The Select Operator. $\sigma_P(\ldots)$:
  - Upon construction at compile time, Predicate P of the select is passed to the operator to be embedded in it (or passed as a Hint. More on this later).
  - Open():
    - Issue an Open() operator to the operator under it (could be a table scan operator or any other operator underneath

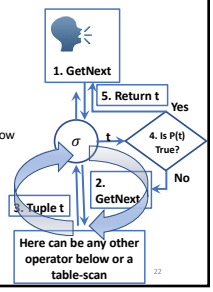


21

21

Physical Operators and their Realizations as Iterators: The Select Operator

- The Select Operator:
  - GetNext():
    - Try to getNext() from the operator below the select - - >(t)
    - Check P on t
    - If true, return t and exit
    - Else repeat the above until a tuple is found or operator below reports Done.
  - Close():
    - Call Close() for the operator below the select
    - Return done

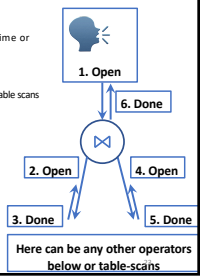


22

22

Physical Operators and their Realizations as Iterators: The Join Operator

- The Join Operator:
- The join predicate will be plugged into the inside of the operator at compile time or will be passed as a hint (more on this issue later).
- State = left_tuple and right_tuple
- Open():
  - Issue an Open() operator to both the left and right operators under the join (could be table scans operator or any other operators underneath)
  - Set left_tuple and right_tuple to be null
- GetNext():
  - If left_tuple = null, then Get the next tuple from the left operator underneath
  - While left operator did not reach end_of_table do
    - If right_tuple = null, then Get the next tuple from the right operator underneath
    - While Join_predicate is not true and not end_of_table of right operator do
      - right_tuple <- right_operator.GetNext();
    - If join_predicate is true then return left_tuple join right tuple
    - Else
      - Reset right operator
      - Left-tuple <- left_operator.GetNext()
- Close():
  - Call Close() for both the left and right operators below the join
  - Return done



23

23

Putting the Operators Together to Construct a Query Evaluation Pipeline

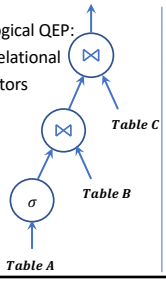
- Connect the operators together to form a QEP.
- Each operator does NOT know the operator underneath it
  - And it does not need to know.
- Each operator follows the iterator model protocol (open, getnext, close).
- Thus, operators can communicate successfully and move tuples around to answer the query.

24

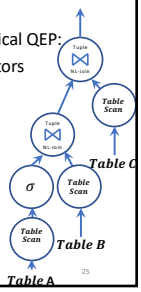
24

Transforming a Logical QEP to a Physical QEP

- An Example Logical QEP:
- Uses Logical Relational Algebra operators



- Corresponding Physical QEP:
- Uses Physical operators

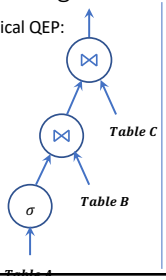


25

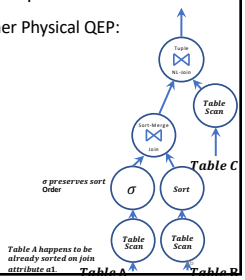
25

Transforming a Logical QEP to a Physical QEP: Using Sort-Merge Joins + Sort Operators

An Example Logical QEP:



Another Physical QEP:



26

Physical Operators

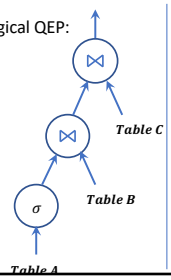
- Table-Scan
- Table-Scan with select predicate as a hint
- Index-Scan (Useful for index-only plans)
- Index–Scan with select predicate as a hint
- Temp (Create a temporary table out of the incoming tuples from the lower-level operator)
- Create-index (Create index on the fly)
- Table-Copy (Make copy of table)
- Page-oriented NL-join, Block NL-join, Indexed NL-join, Hash join, Sort-Merge Join
- Sort
- Bulk-load table from file
- Insert, delete, update
- Collect/update statistics
- Check some integrity constraint
- …

27

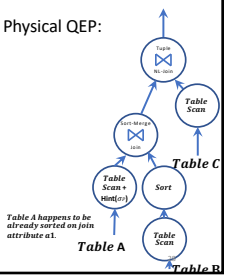
27

Transforming a Logical QEP to a Physical QEP (3): Using Table-Scans with Hints

An Example Logical QEP:



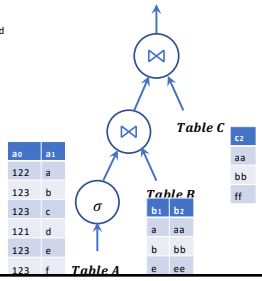
Another Physical QEP:



28

Example Execution of Left-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"



a0	a1
122	a
123	b
123	c
121	d
123	e
123	f

b1	b2
a	aa
b	bb
e	ee

c2
aa
bb
ff

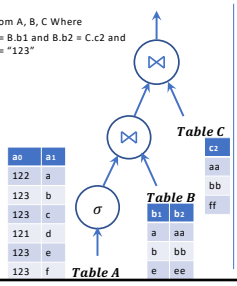
- Open
- GetNext
- GetNext
- GetNext
- Close

29

29

Example Execution of Left-Deep QEP

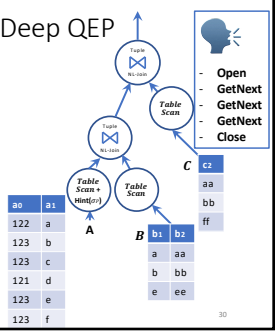
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"



a0	a1
122	a
123	b
123	c
121	d
123	e
123	f

b1	b2
a	aa
b	bb
e	ee

c2
aa
bb
ff



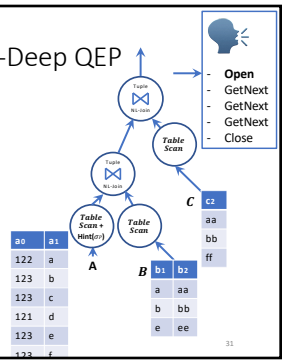
- Open
- GetNext
- GetNext
- GetNext
- Close

30

30

Example Execution of Left-Deep QEP

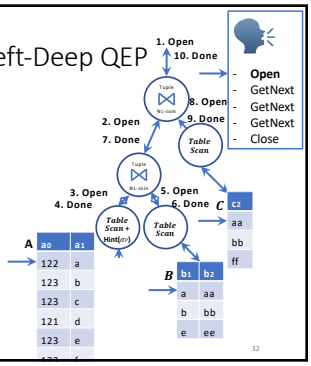
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"



31

Example Execution of Left-Deep QEP

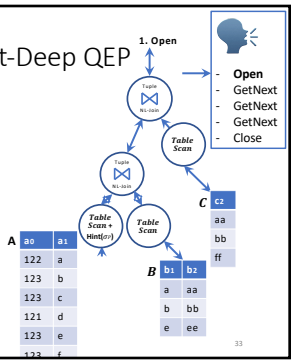
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"



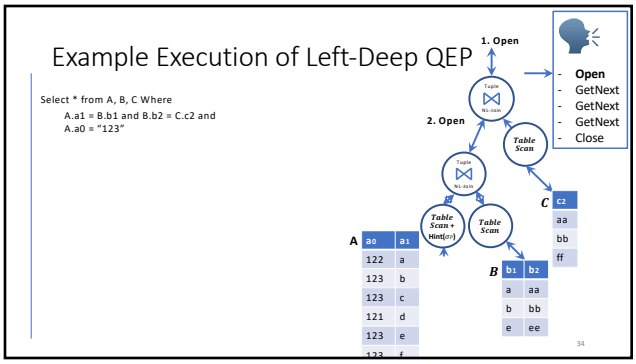
32

Example Execution of Left-Deep QEP

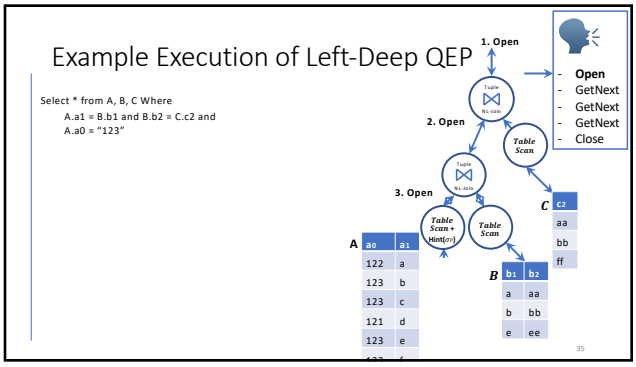
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"



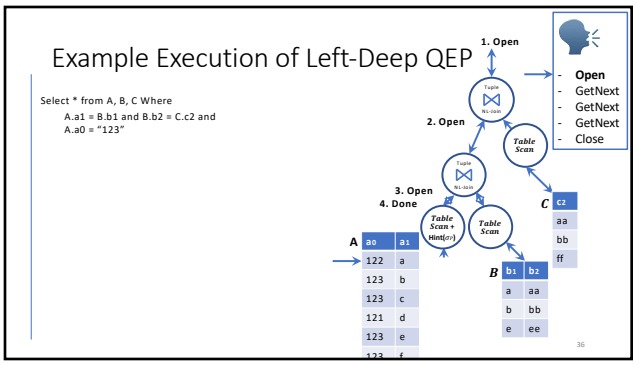
33


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
1. Open
2. Open
Table Scan
Table Scan + Hint(σ)
- Open
- GetNext
- GetNext
- GetNext
- Close
A
a0 a1
122 a
123 b
123 c
121 d
123 e
B
b1 b2
a aa
b bb
e ee
C
c2
aa
bb
ff
34

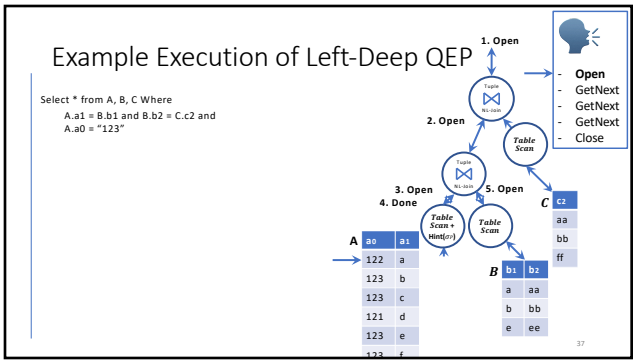
34


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
1. Open
2. Open
3. Open
Table Scan
Table Scan + Hint(σ)
- Open
- GetNext
- GetNext
- GetNext
- Close
A
a0 a1
122 a
123 b
123 c
121 d
123 e
B
b1 b2
a aa
b bb
e ee
C
c2
aa
bb
ff
35

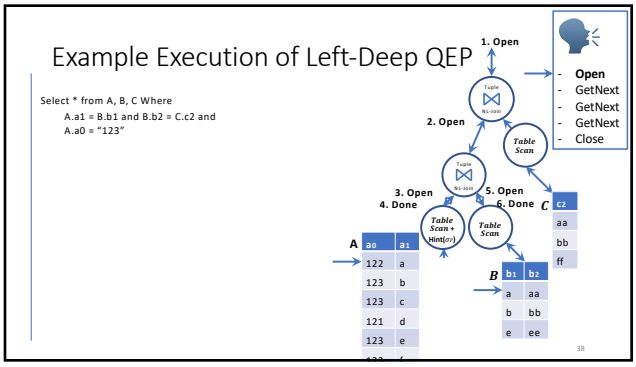
35


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
1. Open
2. Open
3. Open
4. Done
Table Scan
Table Scan + Hint(σ)
- Open
- GetNext
- GetNext
- GetNext
- Close
A
a0 a1
122 a
123 b
123 c
121 d
123 e
B
b1 b2
a aa
b bb
e ee
C
c2
aa
bb
ff
36

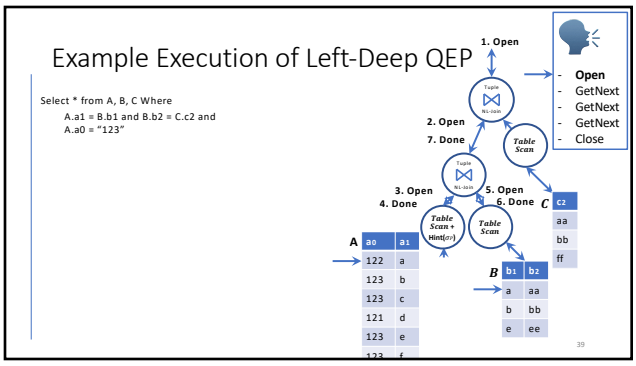
36


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
1. Open
2. Open
3. Open
4. Done
5. Open
- Open
- GetNext
- GetNext
- GetNext
- Close
Table Scan
Table Scan + Hint(a1)
A
a0 a1
122 a
123 b
123 c
121 d
123 e
B
b1 b2
a aa
b bb
e ee
C
c2
aa
bb
ff
37

37


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
1. Open
2. Open
3. Open
4. Done
5. Open
6. Done
- Open
- GetNext
- GetNext
- GetNext
- Close
Table Scan
Table Scan + Hint(a1)
A
a0 a1
122 a
123 b
123 c
121 d
123 e
B
b1 b2
a aa
b bb
e ee
C
c2
aa
bb
ff
38

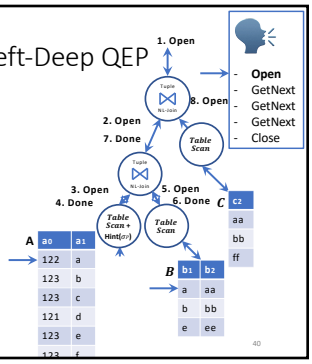
38


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
1. Open
2. Open
7. Done
3. Open
4. Done
5. Open
6. Done
- Open
- GetNext
- GetNext
- GetNext
- Close
Table Scan
Table Scan + Hint(a1)
A
a0 a1
122 a
123 b
123 c
121 d
123 e
B
b1 b2
a aa
b bb
e ee
C
c2
aa
bb
ff
39

39

Example Execution of Left-Deep QEP

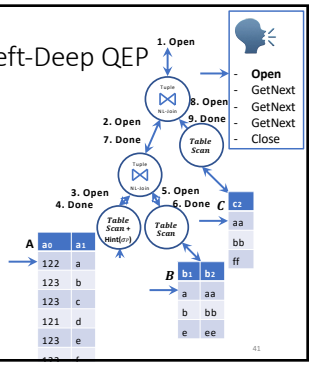
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"



40

Example Execution of Left-Deep QEP

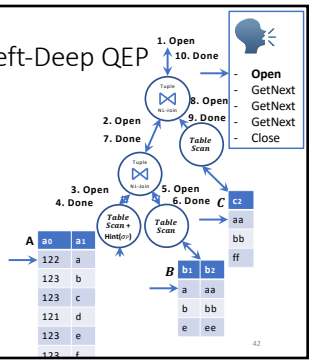
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"



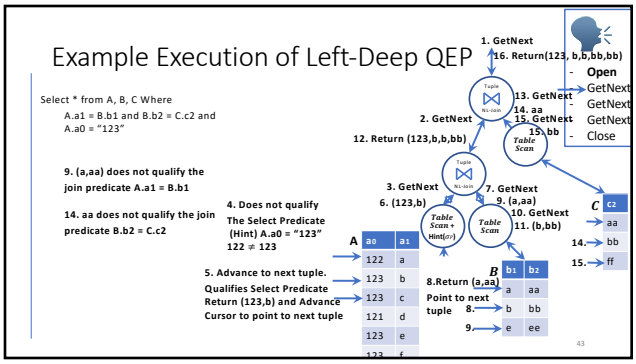
41

Example Execution of Left-Deep QEP

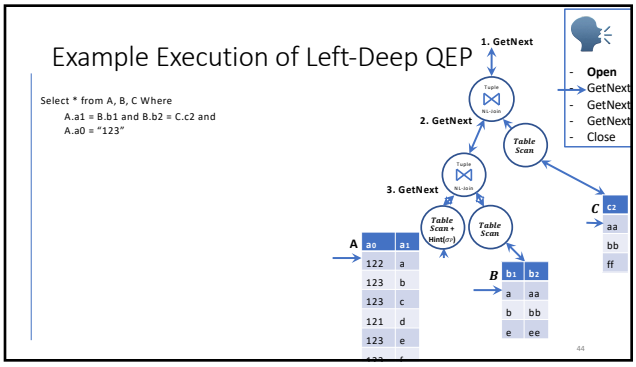
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"



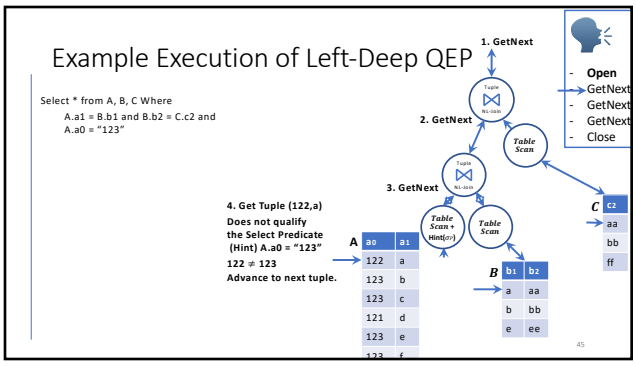
42


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
9. (a,aa) does not qualify the join predicate A.a1 = B.b1
14. aa does not qualify the join predicate B.b2 = C.c2
4. Does not qualify The Select Predicate (Hint) A.a0 = "123" 122 ≠ 123
5. Advance to next tuple. Qualifies Select Predicate Return (123,b) and Advance Cursor to point to next tuple
8.Return (a,aa) Point to next tuple
1. GetNext
16. Return(123, b,b,bb,bb)
2. GetNext
13. GetNext
14. aa
15. GetNext
15. bb
12. Return (123,b,b,bb)
3. GetNext
6. (123,b)
7. GetNext
9. (a,aa)
10. GetNext
11. (b,bb)
14.
15.
Open
GetNext
GetNext
GetNext
Close
Table Scan
Table Scan + Hint(a0)
A
B
C
43

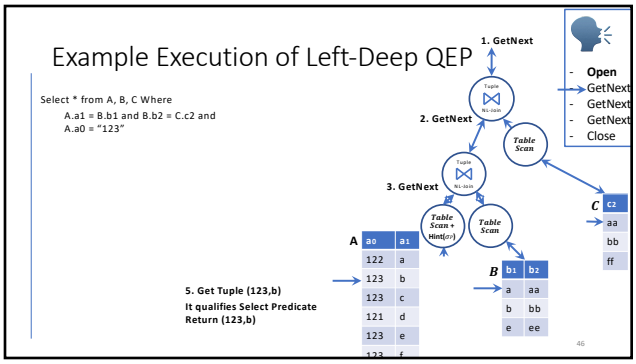
43


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
1. GetNext
2. GetNext
3. GetNext
Open
GetNext
GetNext
GetNext
Close
Table Scan
Table Scan + Hint(a0)
A
B
C
44

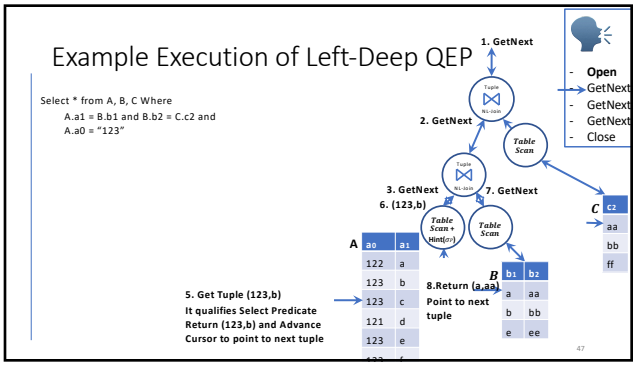
44


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
4. Get Tuple (122,a)
Does not qualify the Select Predicate (Hint) A.a0 = "123"
122 ≠ 123
Advance to next tuple.
1. GetNext
2. GetNext
3. GetNext
Open
GetNext
GetNext
GetNext
Close
Table Scan
Table Scan + Hint(a0)
A
B
C
45

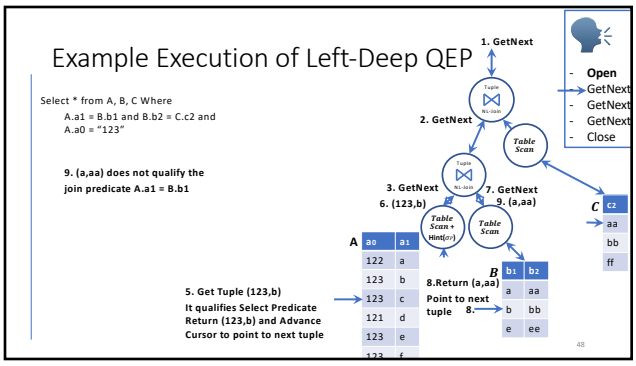
45


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
1. GetNext
2. GetNext
3. GetNext
Table Scan
Table Scan + Hint(a1)
- Open
- GetNext
- GetNext
- GetNext
- Close
A
a0 a1
122 a
123 b
123 c
121 d
123 e
B
b1 b2
a aa
b bb
e ee
C
c2
aa
bb
ff
5. Get Tuple (123,b)
It qualifies Select Predicate
Return (123,b)
46

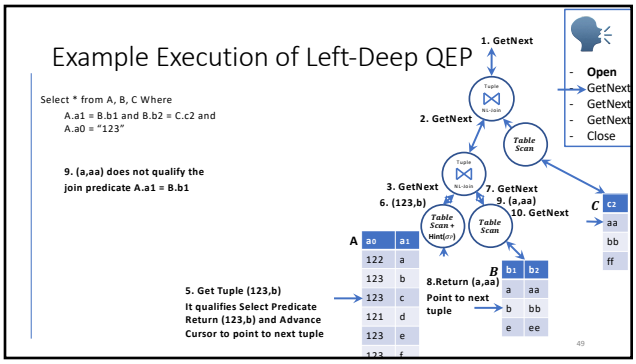
46


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
1. GetNext
2. GetNext
3. GetNext
6. (123,b)
7. GetNext
Table Scan
Table Scan + Hint(a1)
- Open
- GetNext
- GetNext
- GetNext
- Close
A
a0 a1
122 a
123 b
123 c
121 d
123 e
B
b1 b2
a aa
b bb
e ee
C
c2
aa
bb
ff
5. Get Tuple (123,b)
It qualifies Select Predicate
Return (123,b) and Advance
Cursor to point to next tuple
8.Return (a,aa)
Point to next
tuple
47

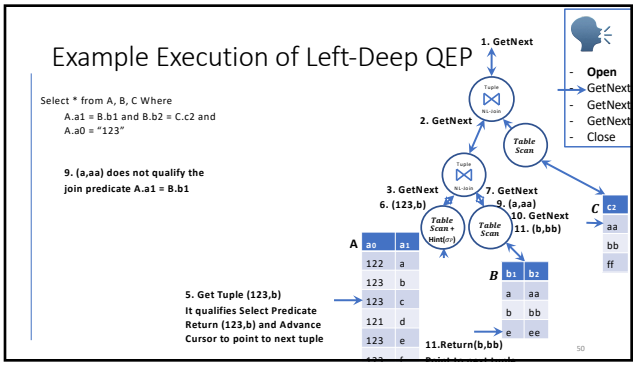
47


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
9. (a,aa) does not qualify the
join predicate A.a1 = B.b1
1. GetNext
2. GetNext
3. GetNext
6. (123,b)
7. GetNext
9. (a,aa)
Table Scan
Table Scan + Hint(a1)
- Open
- GetNext
- GetNext
- GetNext
- Close
A
a0 a1
122 a
123 b
123 c
121 d
123 e
B
b1 b2
a aa
b bb
e ee
C
c2
aa
bb
ff
5. Get Tuple (123,b)
It qualifies Select Predicate
Return (123,b) and Advance
Cursor to point to next tuple
8.Return (a,aa)
Point to next
tuple
8.
48

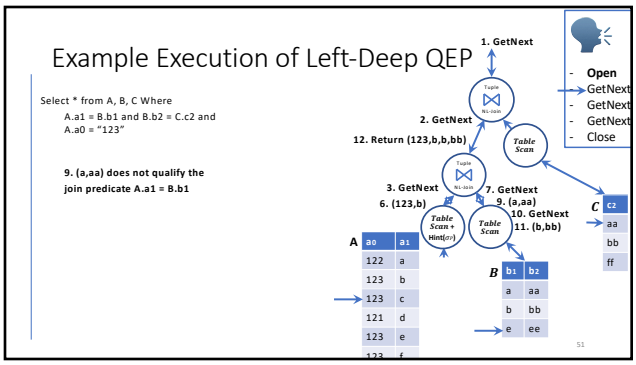
48


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
9. (a,aa) does not qualify the join predicate A.a1 = B.b1
1. GetNext
2. GetNext
3. GetNext
6. (123,b)
7. GetNext
9. (a,aa)
10. GetNext
Table Scan
Table Scan + Hint(a1)
Tuple
NL-Join
- Open
- GetNext
- GetNext
- GetNext
- Close
A
a0 a1
122 a
123 b
123 c
121 d
123 e
B
b1 b2
a aa
b bb
e ee
C
c2
aa
bb
ff
5. Get Tuple (123,b)
It qualifies Select Predicate
Return (123,b) and Advance
Cursor to point to next tuple
8.Return (a,aa)
Point to next tuple
49

49


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
9. (a,aa) does not qualify the join predicate A.a1 = B.b1
1. GetNext
2. GetNext
3. GetNext
6. (123,b)
7. GetNext
9. (a,aa)
10. GetNext
11. (b,bb)
Table Scan
Table Scan + Hint(a1)
Tuple
NL-Join
- Open
- GetNext
- GetNext
- GetNext
- Close
A
a0 a1
122 a
123 b
123 c
121 d
123 e
B
b1 b2
a aa
b bb
e ee
C
c2
aa
bb
ff
5. Get Tuple (123,b)
It qualifies Select Predicate
Return (123,b) and Advance
Cursor to point to next tuple
11.Return(b,bb)
50

50

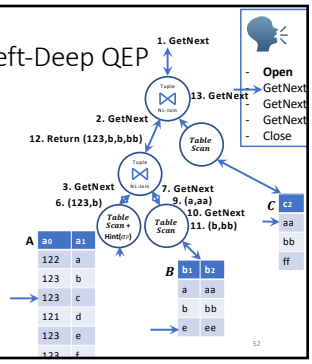

Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
9. (a,aa) does not qualify the join predicate A.a1 = B.b1
1. GetNext
2. GetNext
12. Return (123,b,b,bb)
3. GetNext
6. (123,b)
7. GetNext
9. (a,aa)
10. GetNext
11. (b,bb)
Table Scan
Table Scan + Hint(a1)
Tuple
NL-Join
- Open
- GetNext
- GetNext
- GetNext
- Close
A
a0 a1
122 a
123 b
123 c
121 d
123 e
B
b1 b2
a aa
b bb
e ee
C
c2
aa
bb
ff
51

51

Example Execution of Left-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

9. (a,aa) does not qualify the join predicate A.a1 = B.b1



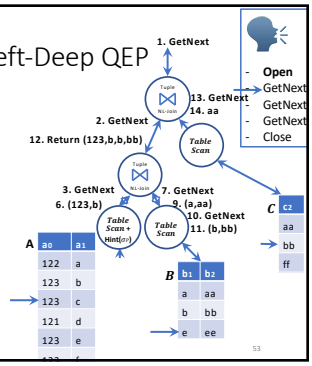
52

Example Execution of Left-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

9. (a,aa) does not qualify the join predicate A.a1 = B.b1

14. aa does not qualify the join predicate B.b2 = C.c2



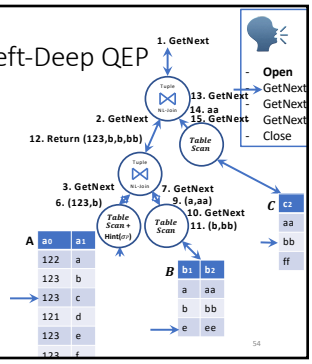
53

Example Execution of Left-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

9. (a,aa) does not qualify the join predicate A.a1 = B.b1

14. aa does not qualify the join predicate B.b2 = C.c2



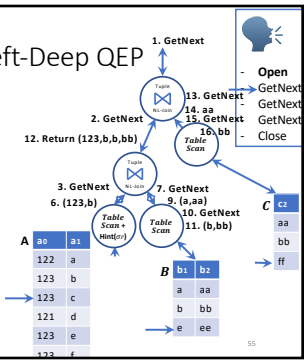
54

Example Execution of Left-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

9. (a,aa) does not qualify the join predicate A.a1 = B.b1

14. aa does not qualify the join predicate B.b2 = C.c2



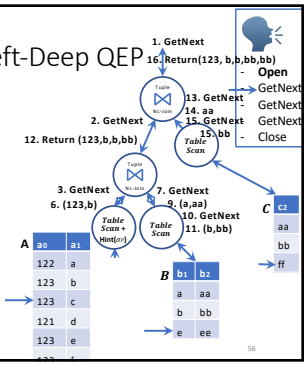
55

Example Execution of Left-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

9. (a,aa) does not qualify the join predicate A.a1 = B.b1

14. aa does not qualify the join predicate B.b2 = C.c2

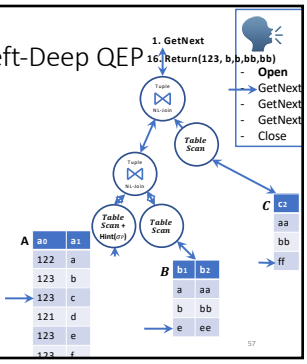


56

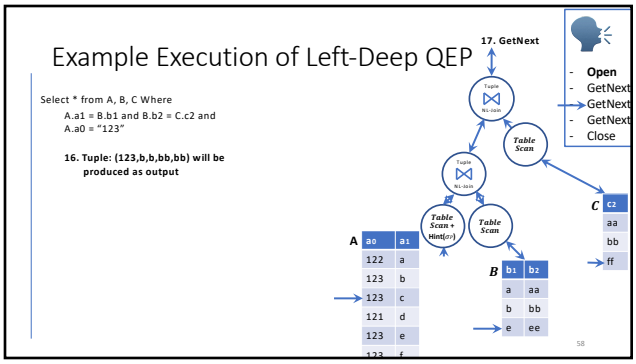
Example Execution of Left-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

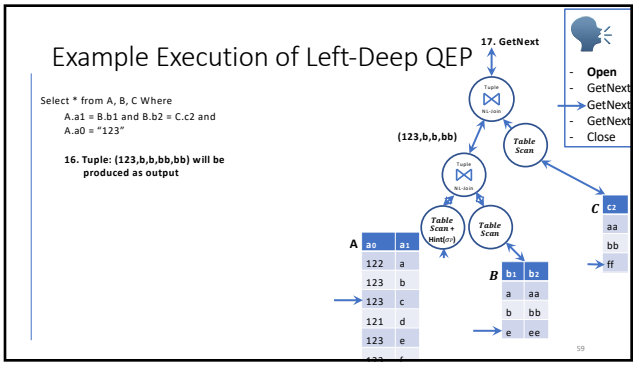
16. Tuple: (123,b,b,bb,bb) will be produced as output



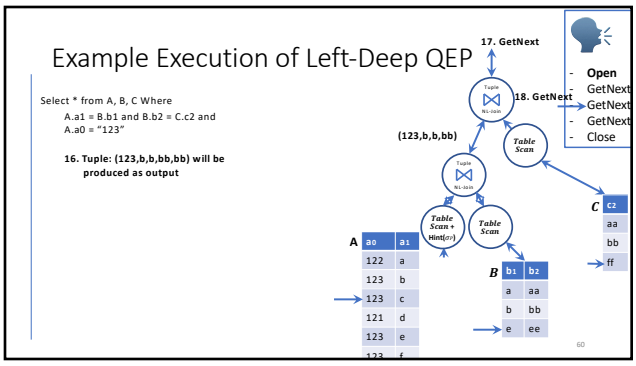
57


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
16. Tuple: (123,b,b,bb,bb) will be produced as output
17. GetNext
- Open
- GetNext
- GetNext
- GetNext
- Close
Table Scan
Table Scan + Hint(a1)
A a0 a1 122 a 123 b 123 c 121 d 123 e
B b1 b2 a aa b bb e ee
C c2 aa bb ff
58

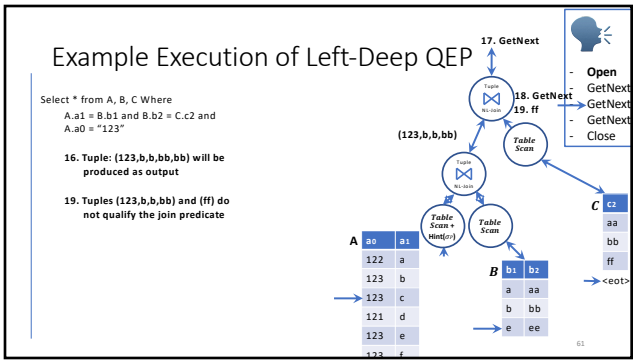
58


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
16. Tuple: (123,b,b,bb,bb) will be produced as output
17. GetNext
(123,b,b,bb)
- Open
- GetNext
- GetNext
- GetNext
- Close
Table Scan
Table Scan + Hint(a1)
A a0 a1 122 a 123 b 123 c 121 d 123 e
B b1 b2 a aa b bb e ee
C c2 aa bb ff
59

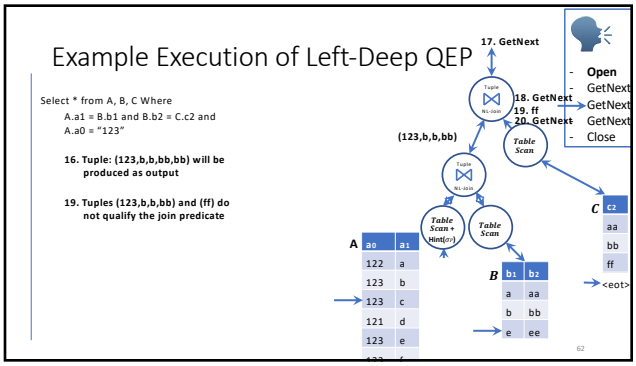
59


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
16. Tuple: (123,b,b,bb,bb) will be produced as output
17. GetNext
18. GetNext
(123,b,b,bb)
- Open
- GetNext
- GetNext
- GetNext
- Close
Table Scan
Table Scan + Hint(a1)
A a0 a1 122 a 123 b 123 c 121 d 123 e
B b1 b2 a aa b bb e ee
C c2 aa bb ff
60

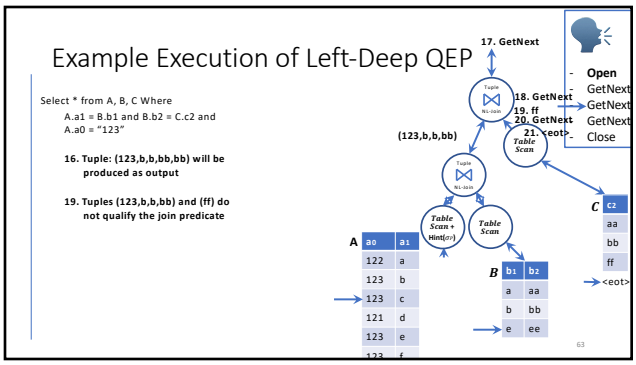
60


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
16. Tuple: (123,b,b,bb,bb) will be produced as output
19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate
17. GetNext
18. GetNext
19. ff
(123,b,b,bb)
Table Scan
Table Scan + Hint(cr)
Table Scan
A
a0 a1
122 a
123 b
123 c
121 d
123 e
B
b1 b2
a aa
b bb
e ee
C
c2
aa
bb
ff
<eot>
- Open
- GetNext
- GetNext
- GetNext
- Close
61

61


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
16. Tuple: (123,b,b,bb,bb) will be produced as output
19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate
17. GetNext
18. GetNext
19. ff
20. GetNext
(123,b,b,bb)
Table Scan
Table Scan + Hint(cr)
Table Scan
A
a0 a1
122 a
123 b
123 c
121 d
123 e
B
b1 b2
a aa
b bb
e ee
C
c2
aa
bb
ff
<eot>
- Open
- GetNext
- GetNext
- GetNext
- Close
62

62


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
16. Tuple: (123,b,b,bb,bb) will be produced as output
19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate
17. GetNext
18. GetNext
19. ff
20. GetNext
21. <eot>
(123,b,b,bb)
Table Scan
Table Scan + Hint(cr)
Table Scan
A
a0 a1
122 a
123 b
123 c
121 d
123 e
B
b1 b2
a aa
b bb
e ee
C
c2
aa
bb
ff
<eot>
- Open
- GetNext
- GetNext
- GetNext
- Close
63

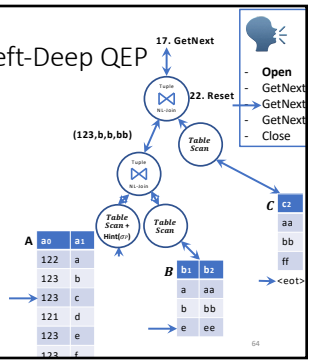
63

Example Execution of Left-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

16. Tuple: (123,b,b,bb,bb) will be produced as output

19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate



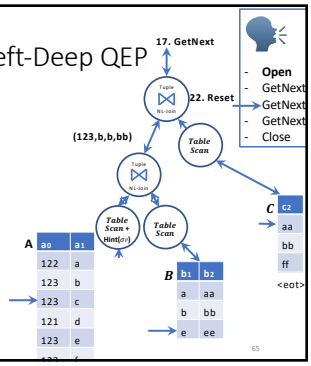
64

Example Execution of Left-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

16. Tuple: (123,b,b,bb,bb) will be produced as output

19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate



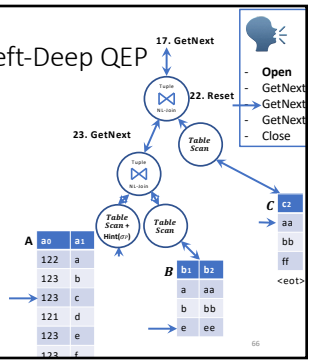
65

Example Execution of Left-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

16. Tuple: (123,b,b,bb,bb) will be produced as output

19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate



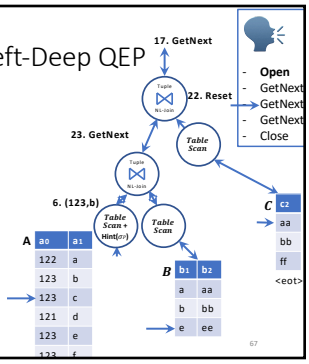
66

Example Execution of Left-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

16. Tuple: (123,b,b,bb,bb) will be produced as output

19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate



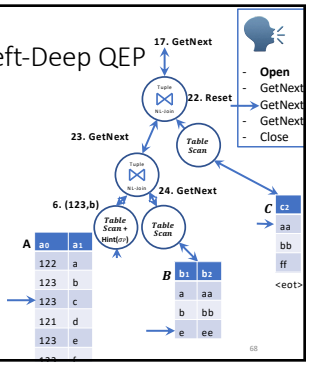
67

Example Execution of Left-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

16. Tuple: (123,b,b,bb,bb) will be produced as output

19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate



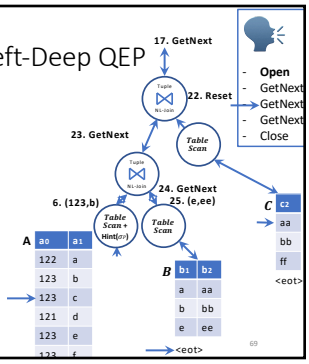
68

Example Execution of Left-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

16. Tuple: (123,b,b,bb,bb) will be produced as output

19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate



69

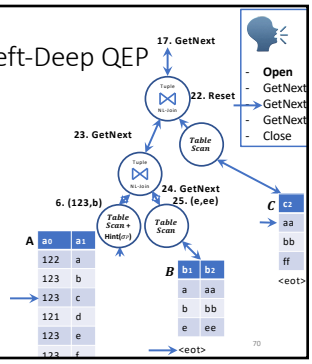
Example Execution of Left-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

16. Tuple: (123,b,b,bb,bb) will be produced as output

19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate

25. Tuples (123,b) and (e,ee) do not qualify the join predicate



70

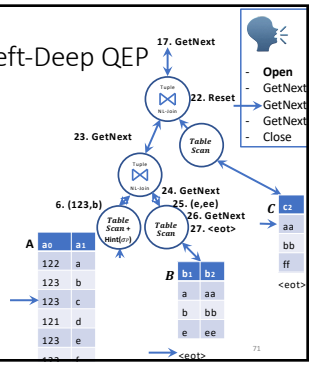
Example Execution of Left-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

16. Tuple: (123,b,b,bb,bb) will be produced as output

19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate

25. Tuples (123,b) and (e,ee) do not qualify the join predicate



71

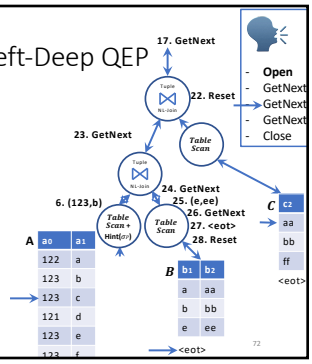
Example Execution of Left-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

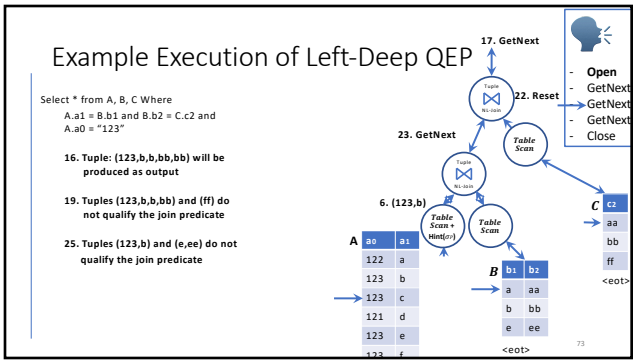
16. Tuple: (123,b,b,bb,bb) will be produced as output

19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate

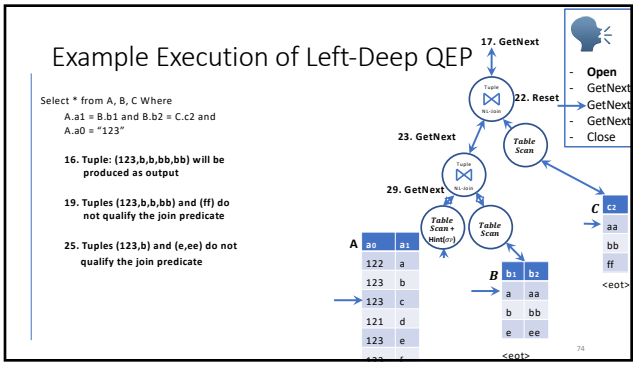
25. Tuples (123,b) and (e,ee) do not qualify the join predicate



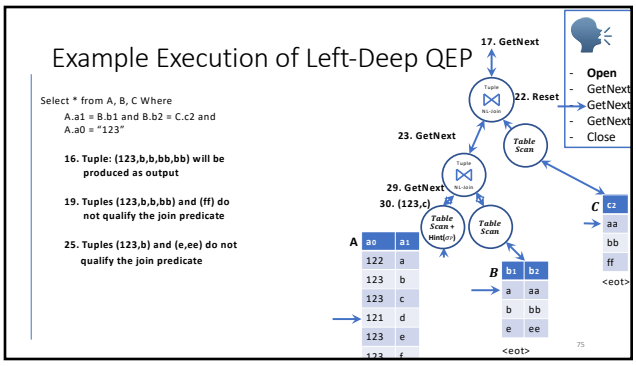
72


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
16. Tuple: (123,b,b,bb,bb) will be produced as output
19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate
25. Tuples (123,b) and (e,ee) do not qualify the join predicate
17. GetNext
22. Reset
23. GetNext
6. (123,b)
Open
GetNext
GetNext
GetNext
Close
Table Scan
Table Scan + Hint(a1)
A
B
C
<eot>
73

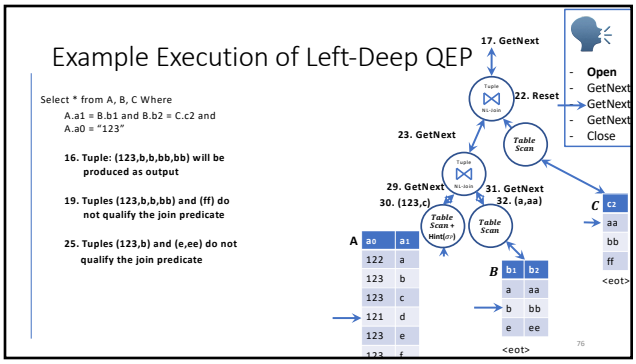
73


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
16. Tuple: (123,b,b,bb,bb) will be produced as output
19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate
25. Tuples (123,b) and (e,ee) do not qualify the join predicate
17. GetNext
22. Reset
23. GetNext
29. GetNext
Open
GetNext
GetNext
GetNext
Close
Table Scan
Table Scan + Hint(a1)
A
B
C
<eot>
74

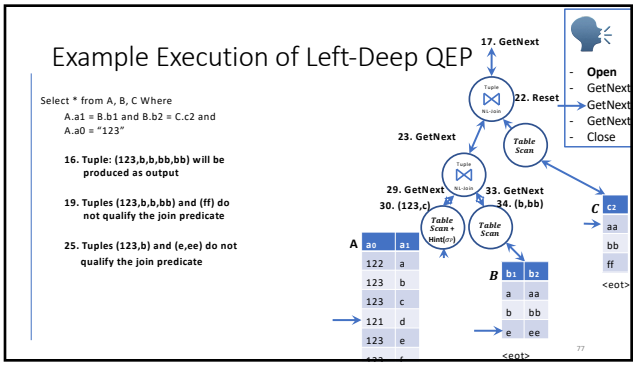
74


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
16. Tuple: (123,b,b,bb,bb) will be produced as output
19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate
25. Tuples (123,b) and (e,ee) do not qualify the join predicate
17. GetNext
22. Reset
23. GetNext
29. GetNext
30. (123,c)
Open
GetNext
GetNext
GetNext
Close
Table Scan
Table Scan + Hint(a1)
A
B
C
<eot>
75

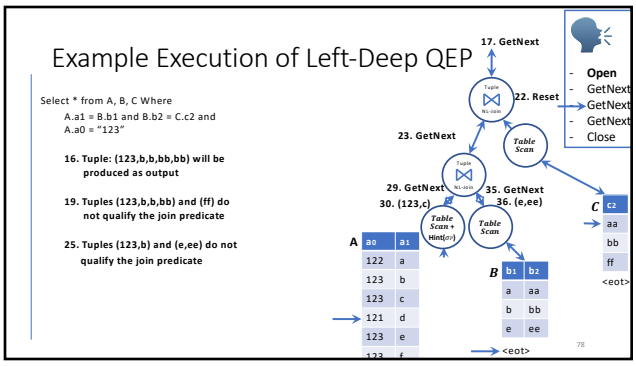
75


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
16. Tuple: (123,b,b,bb,bb) will be produced as output
19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate
25. Tuples (123,b) and (e,ee) do not qualify the join predicate
17. GetNext
22. Reset
23. GetNext
29. GetNext
30. (123,c)
31. GetNext
32. (a,aa)
Table Scan
Table Scan + Hint(idx)
A
B
C
- Open
- GetNext
- GetNext
- GetNext
- Close
<eot>
76

76


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
16. Tuple: (123,b,b,bb,bb) will be produced as output
19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate
25. Tuples (123,b) and (e,ee) do not qualify the join predicate
17. GetNext
22. Reset
23. GetNext
29. GetNext
30. (123,c)
33. GetNext
34. (b,bb)
Table Scan
Table Scan + Hint(idx)
A
B
C
- Open
- GetNext
- GetNext
- GetNext
- Close
<eot>
77

77


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
16. Tuple: (123,b,b,bb,bb) will be produced as output
19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate
25. Tuples (123,b) and (e,ee) do not qualify the join predicate
17. GetNext
22. Reset
23. GetNext
29. GetNext
30. (123,c)
35. GetNext
36. (e,ee)
Table Scan
Table Scan + Hint(idx)
A
B
C
- Open
- GetNext
- GetNext
- GetNext
- Close
<eot>
78

78

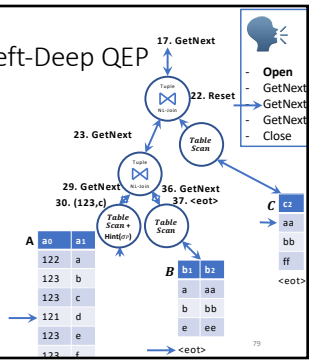
Example Execution of Left-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

16. Tuple: (123,b,b,bb,bb) will be produced as output

19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate

25. Tuples (123,b) and (e,ee) do not qualify the join predicate



79

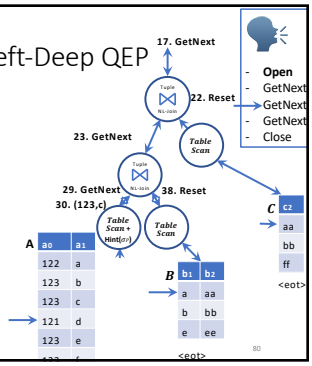
Example Execution of Left-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

16. Tuple: (123,b,b,bb,bb) will be produced as output

19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate

25. Tuples (123,b) and (e,ee) do not qualify the join predicate



80

Example Execution of Left-Deep QEP

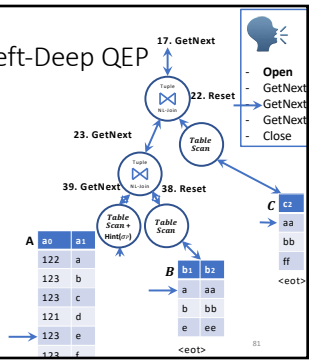
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

16. Tuple: (123,b,b,bb,bb) will be produced as output

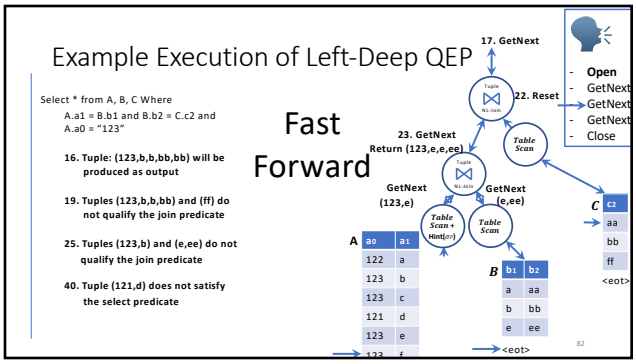
19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate

25. Tuples (123,b) and (e,ee) do not qualify the join predicate

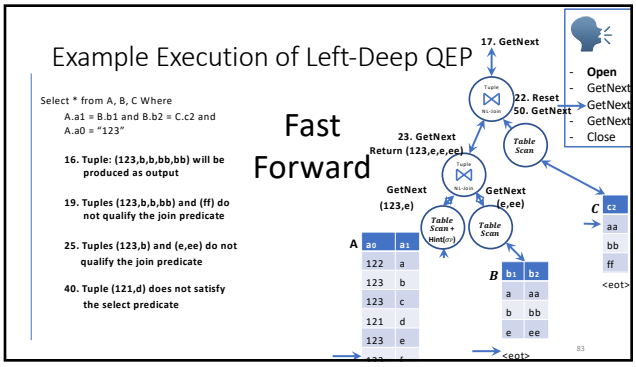
40. Tuple (121,d) does not satisfy the select predicate



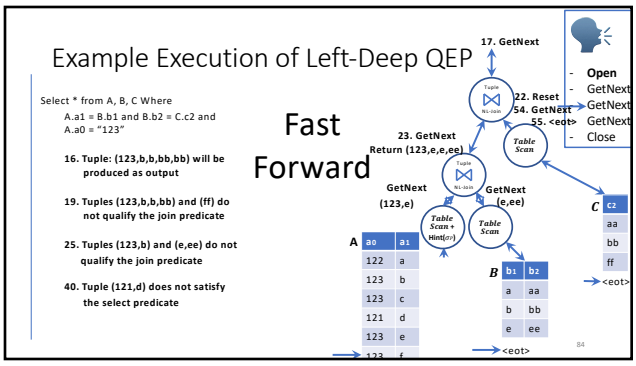
81


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
16. Tuple: (123,b,b,bb,bb) will be produced as output
19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate
25. Tuples (123,b) and (e,ee) do not qualify the join predicate
40. Tuple (121,d) does not satisfy the select predicate
Fast Forward
17. GetNext
22. Reset
23. GetNext
Return (123,e,e,ee)
GetNext (123,e)
GetNext (e,ee)
Table Scan
Table Scan + Hint(a1)
- Open
- GetNext
- GetNext
- GetNext
- Close
A
a0 a1
122 a
123 b
123 c
121 d
123 e
B
b1 b2
a aa
b bb
e ee
C
c2
aa
bb
ff
<eot>
82

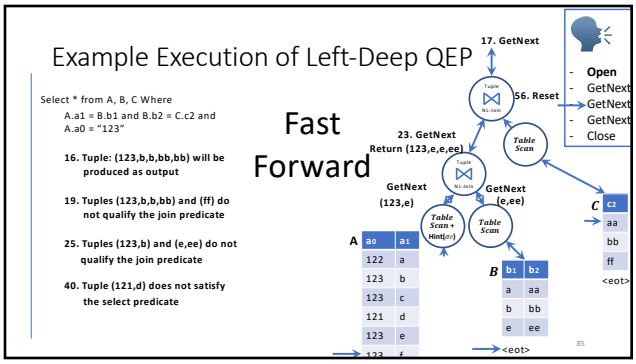
82


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
16. Tuple: (123,b,b,bb,bb) will be produced as output
19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate
25. Tuples (123,b) and (e,ee) do not qualify the join predicate
40. Tuple (121,d) does not satisfy the select predicate
Fast Forward
17. GetNext
22. Reset
50. GetNext
23. GetNext
Return (123,e,e,ee)
GetNext (123,e)
GetNext (e,ee)
Table Scan
Table Scan + Hint(a1)
- Open
- GetNext
- GetNext
- GetNext
- Close
A
a0 a1
122 a
123 b
123 c
121 d
123 e
B
b1 b2
a aa
b bb
e ee
C
c2
aa
bb
ff
<eot>
83

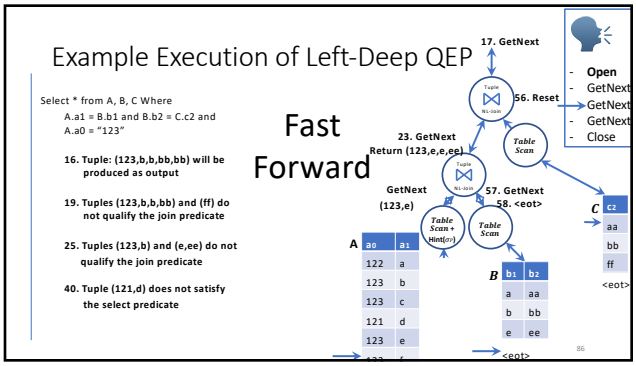
83


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
16. Tuple: (123,b,b,bb,bb) will be produced as output
19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate
25. Tuples (123,b) and (e,ee) do not qualify the join predicate
40. Tuple (121,d) does not satisfy the select predicate
Fast Forward
17. GetNext
22. Reset
54. GetNext
55. <eot>
23. GetNext
Return (123,e,e,ee)
GetNext (123,e)
GetNext (e,ee)
Table Scan
Table Scan + Hint(a1)
- Open
- GetNext
- GetNext
- GetNext
- Close
A
a0 a1
122 a
123 b
123 c
121 d
123 e
B
b1 b2
a aa
b bb
e ee
C
c2
aa
bb
ff
<eot>
84

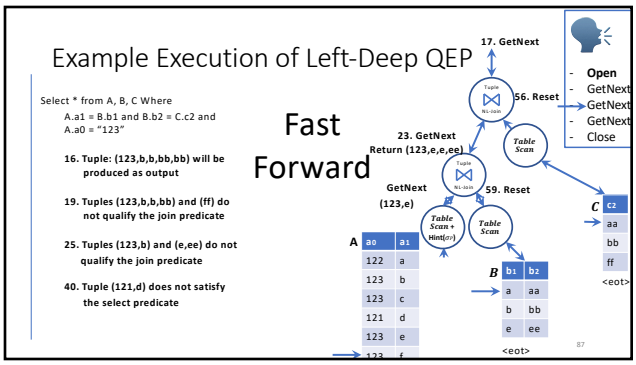
84


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
16. Tuple: (123,b,b,bb,bb) will be produced as output
19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate
25. Tuples (123,b) and (e,ee) do not qualify the join predicate
40. Tuple (121,d) does not satisfy the select predicate
Fast Forward
17. GetNext
56. Reset
23. GetNext
Return (123,e,e,ee)
GetNext
(123,e)
GetNext
(e,ee)
Table Scan
Table Scan + Hint(a1)
- Open
- GetNext
- GetNext
- GetNext
- Close
A
a0
a1
122 a
123 b
123 c
121 d
123 e
B
b1
b2
a aa
b bb
e ee
C
c2
aa
bb
ff
<eot>
85

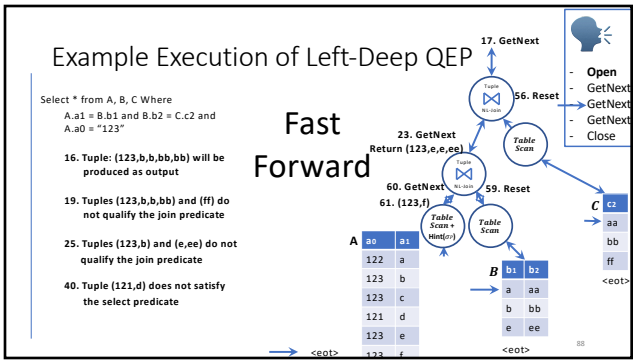
85


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
16. Tuple: (123,b,b,bb,bb) will be produced as output
19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate
25. Tuples (123,b) and (e,ee) do not qualify the join predicate
40. Tuple (121,d) does not satisfy the select predicate
Fast Forward
17. GetNext
56. Reset
23. GetNext
Return (123,e,e,ee)
GetNext
(123,e)
57. GetNext
58. <eot>
Table Scan
Table Scan + Hint(a1)
- Open
- GetNext
- GetNext
- GetNext
- Close
A
a0
a1
122 a
123 b
123 c
121 d
123 e
B
b1
b2
a aa
b bb
e ee
C
c2
aa
bb
ff
<eot>
86

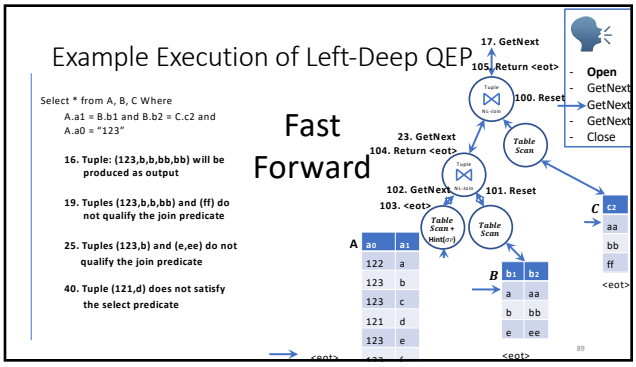
86


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
16. Tuple: (123,b,b,bb,bb) will be produced as output
19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate
25. Tuples (123,b) and (e,ee) do not qualify the join predicate
40. Tuple (121,d) does not satisfy the select predicate
Fast Forward
17. GetNext
56. Reset
23. GetNext
Return (123,e,e,ee)
GetNext
(123,e)
59. Reset
Table Scan
Table Scan + Hint(a1)
- Open
- GetNext
- GetNext
- GetNext
- Close
A
a0
a1
122 a
123 b
123 c
121 d
123 e
B
b1
b2
a aa
b bb
e ee
C
c2
aa
bb
ff
<eot>
87

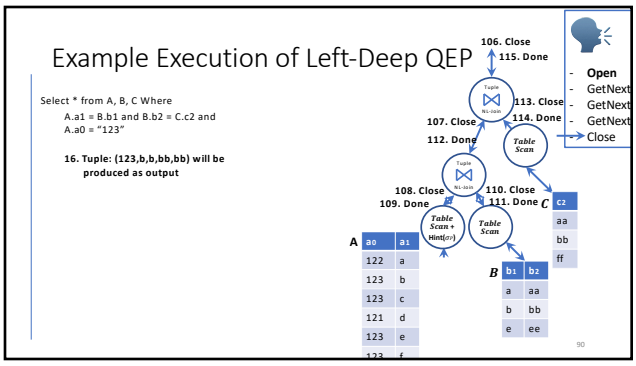
87


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
16. Tuple: (123,b,b,bb,bb) will be produced as output
19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate
25. Tuples (123,b) and (e,ee) do not qualify the join predicate
40. Tuple (121,d) does not satisfy the select predicate
Fast Forward
17. GetNext
56. Reset
23. GetNext
Return (123,e,e,ee)
60. GetNext
61. (123,f)
59. Reset
Table Scan
Table Scan + Hint(cr)
Open
GetNext
GetNext
GetNext
Close
A
a0
a1
122 a
123 b
123 c
121 d
123 e
B
b1
b2
a aa
b bb
e ee
C
c2
aa
bb
ff
<eot>
88

88


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
16. Tuple: (123,b,b,bb,bb) will be produced as output
19. Tuples (123,b,b,bb) and (ff) do not qualify the join predicate
25. Tuples (123,b) and (e,ee) do not qualify the join predicate
40. Tuple (121,d) does not satisfy the select predicate
Fast Forward
17. GetNext
105. Return <eot>
100. Reset
23. GetNext
104. Return <eot>
102. GetNext
103. <eot>
101. Reset
Table Scan
Table Scan + Hint(cr)
Open
GetNext
GetNext
GetNext
Close
A
a0
a1
122 a
123 b
123 c
121 d
123 e
B
b1
b2
a aa
b bb
e ee
C
c2
aa
bb
ff
<eot>
89

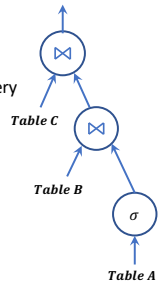
89


Example Execution of Left-Deep QEP
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"
16. Tuple: (123,b,b,bb,bb) will be produced as output
106. Close
115. Done
113. Close
114. Done
107. Close
112. Done
108. Close
109. Done
110. Close
111. Done
Table Scan
Table Scan + Hint(cr)
Open
GetNext
GetNext
GetNext
Close
A
a0
a1
122 a
123 b
123 c
121 d
123 e
B
b1
b2
a aa
b bb
e ee
C
c2
aa
bb
ff
90

90

Right-Deep QEPs

- Does a Right-deep QEP differ in terms of query processing w.r.t. Left-deep QEPs?

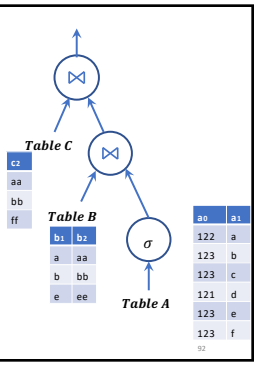


91

91

Example Execution of Right-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"



92

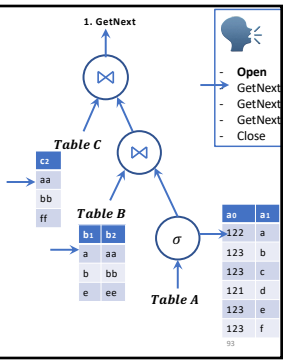
92

Example Execution of Right-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

- Assume we executed the Open method all the way

- Now, we execute GetNext

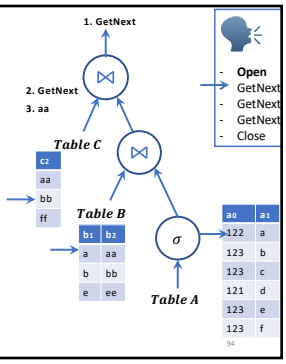


93

93

Example Execution of Right-Deep QEP

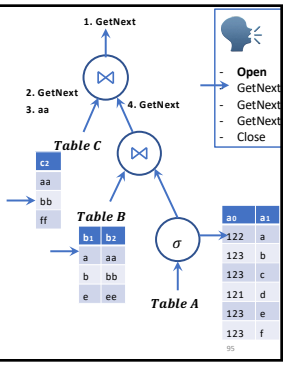
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"



94

Example Execution of Right-Deep QEP

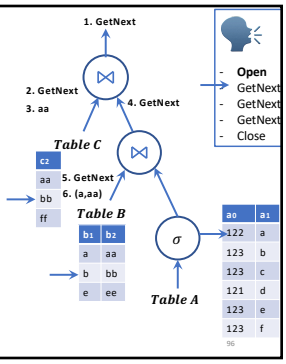
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"



95

Example Execution of Right-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

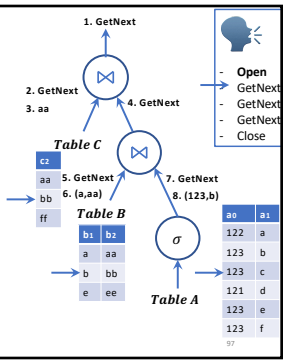


96

Example Execution of Right-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

Tuple (122,a) does not qualify the select predicate

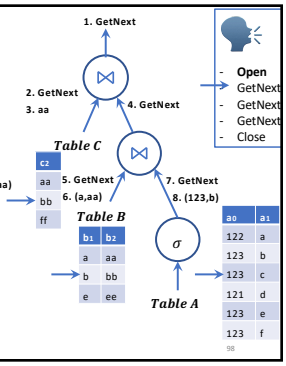


97

Example Execution of Right-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

7.1. Tuple (122,a) does not qualify the select predicate
9. All other tuples in Table A will not join with tuple (a,aa)



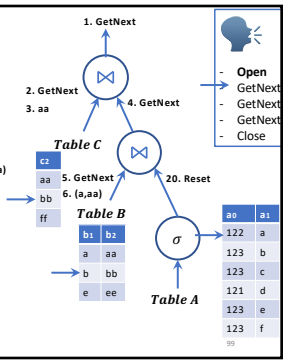
98

Example Execution of Right-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

- All other tuples in Table A will not join with tuple (a,aa)

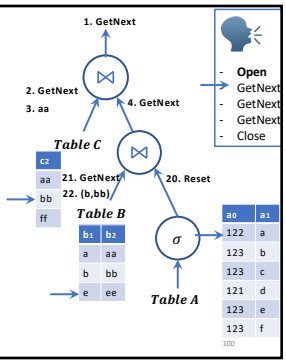
- Fast Forward



99

Example Execution of Right-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"



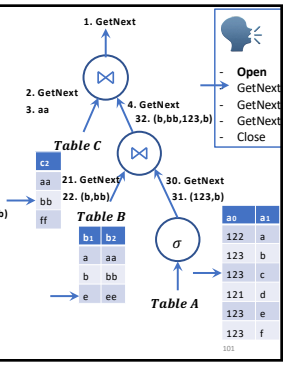
100

Example Execution of Right-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

33. Tuples (aa) and (b,bb,123,b) do not satisfy the join predicate

Notice that Tuple (b,bb,123,b) would join with Tuple (bb) in Table C that is yet to be processed, but we will throw away Tuple (b,bb,123,b) and move on.



101

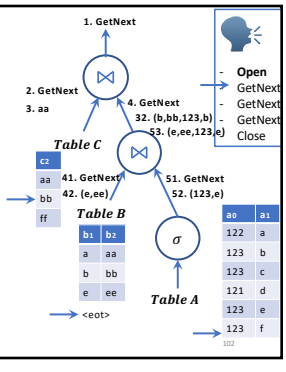
Example Execution of Right-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

Fast Forward

54. Tuples (aa) and (e,ee,123,e) do not satisfy the join predicate

55. Fast Forward



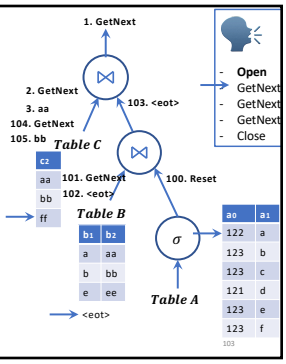
102

Example Execution of Right-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

106. Now, the join with Tables B and A will need to be reevaluated for every new tuple of C

107. Fast Forward



103

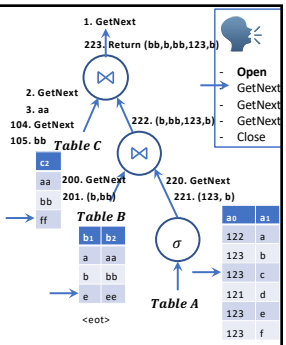
Example Execution of Right-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

222. Tuples (b,bb) and (123,b) satisfy the join predicate for the second time (Recall Step 32)

223. Tuples (bb) and (b,bb,123,b) satisfy the join predicate and is produced as output

224. Fast Forward



104

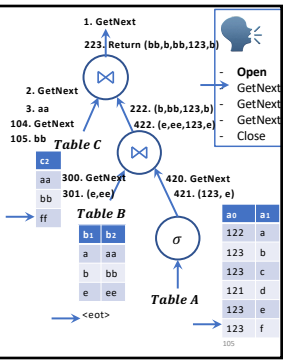
Example Execution of Right-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

423. Tuple (e,ee,123,e) got reevaluated for the second time (Recall Step 53)

424. Tuple (bb) and (e,ee,123,e) do not satisfy the join predicate

425. Fast Forward to the end as no more tuples will qualify

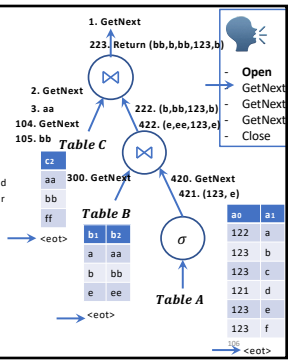


105

Example Execution of Right-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

Contrast the two snapshots of execution in this Slide and the snapshot in Slide 102 (listed also in the next slide for simplicity



106

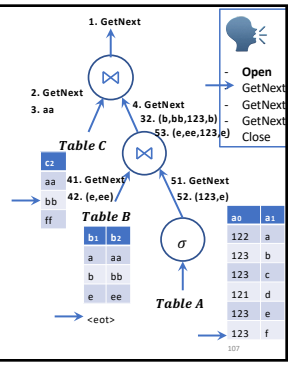
Example Execution of Right-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

Fast Forward

54. Tuples (aa) and (e,ee,123,e) do not satisfy the join predicate

55. Fast Forward



107

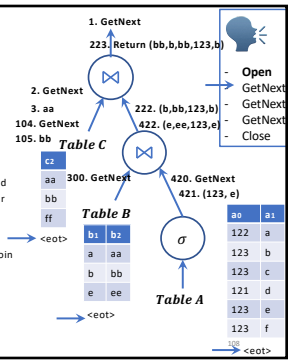
Example Execution of Right-Deep QEP

Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"

Contrast the two snapshots of execution in this Slide and the snapshot in Slide 102 (listed also in the next slide for simplicity

Notice that the join of Tables A and B was performed twice, and of course in each time, produced the same join result: Tuples: **(b,bb,123,b) and e,ee,123,e)**

Can you suggest an optimization here?

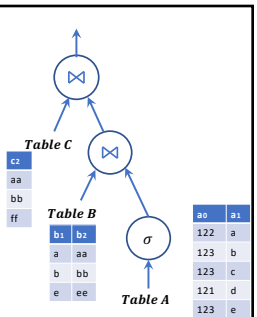


108

Optimizing Right-deep QEPs

- Will add a temp operator on top of each join operator:
  - Perform the once and save its results
  - Next time, instead of re-evaluating it, access the temp table

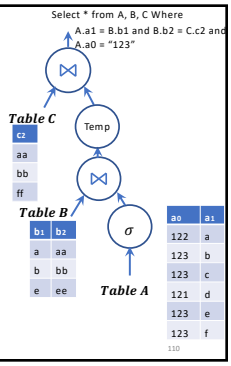
Select * from A, B, C Where
A.a1 = B.b1 and B.b2 = C.c2 and
A.a0 = "123"



109

Optimizing Right-deep QEPs

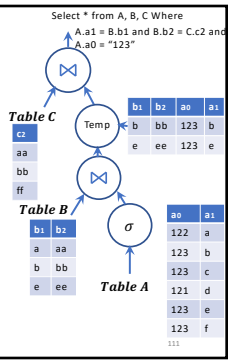
- Note the addition of the Temp operator above the join of B and A
- Temp will create a temporary table that is the result of the subtree below it
  - Temp = B $\bowtie_{b.b1 = A.a1}$ ($\sigma_{A.a0 = \text{"123"}}$(A))
- The temporary table is created the first time the Temp operator is invoked



110

Optimizing Right-deep QEPs

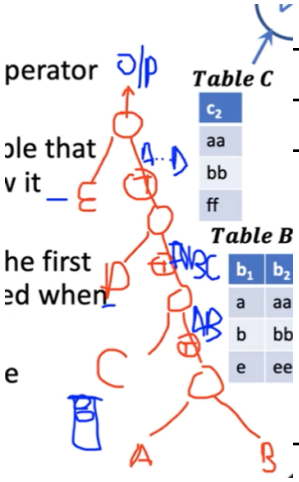
- Note the addition of the Temp operator above the join of B and A
- Temp will create a temporary table that is the result of the subtree below it
  - Temp = B $\bowtie_{b.b1 = A.a1}$ ($\sigma_{A.a0 = \text{"123"}}$(A))
- The temporary table is created the first time the Temp operator is invoked when processing Tuple (aa) of Table C
- For the remaining tuples in C, the temporary table is accessed



111

How to optimize right deep QEP
By ~~adding a temporary table(result of subtree below it) above a join op~~erator.
So we don't need to iterate to get every join permutation.

i.e. If there is a query that E joins D joins C joins A and B.
We ~~add a temporary table above each join operator~~



Comparison between Left- and Right-deep QEPs

- Left-deep Trees are the more favorable
- Allow for non-blocking operators
- Do not need to write temporary tables on disk
- Do not have to read the temporary tables from disk over and over for every tuple of the outer tables
- Make the query optimization job easier as we do not have to search over all possible trees that can evaluate the query
- But still exponential as we need to know the order of the joins (which join to perform first).

112

112

However, temporary table is blocking (get all results from subtree until done) and need space to store temporary results