

CSCI317 Database Performance Tuning

Query Processing Plans

Dr Janusz R. Getta

School of Computing and Information Technology -
University of Wollongong

Query Processing Plans

Outline

Syntax tree

Left/right deep syntax tree

EXPLAIN PLAN statement of SQL

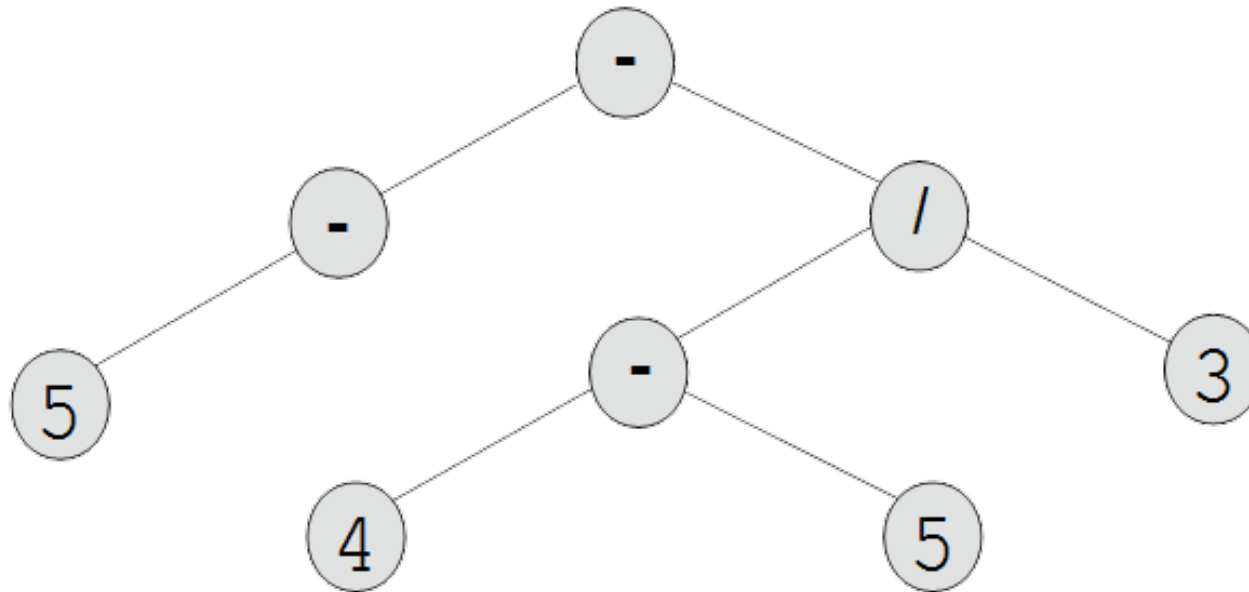
Interpretation of processing plans

Syntax tree

Syntax tree is a two dimensional visualization of an expression that consists of operations and arguments

$(-5 - (4 - 5))/3$

Arithmetic expression



Query Processing Plans

Outline

Syntax tree

Left/right deep syntax tree

EXPLAIN PLAN statement of SQL

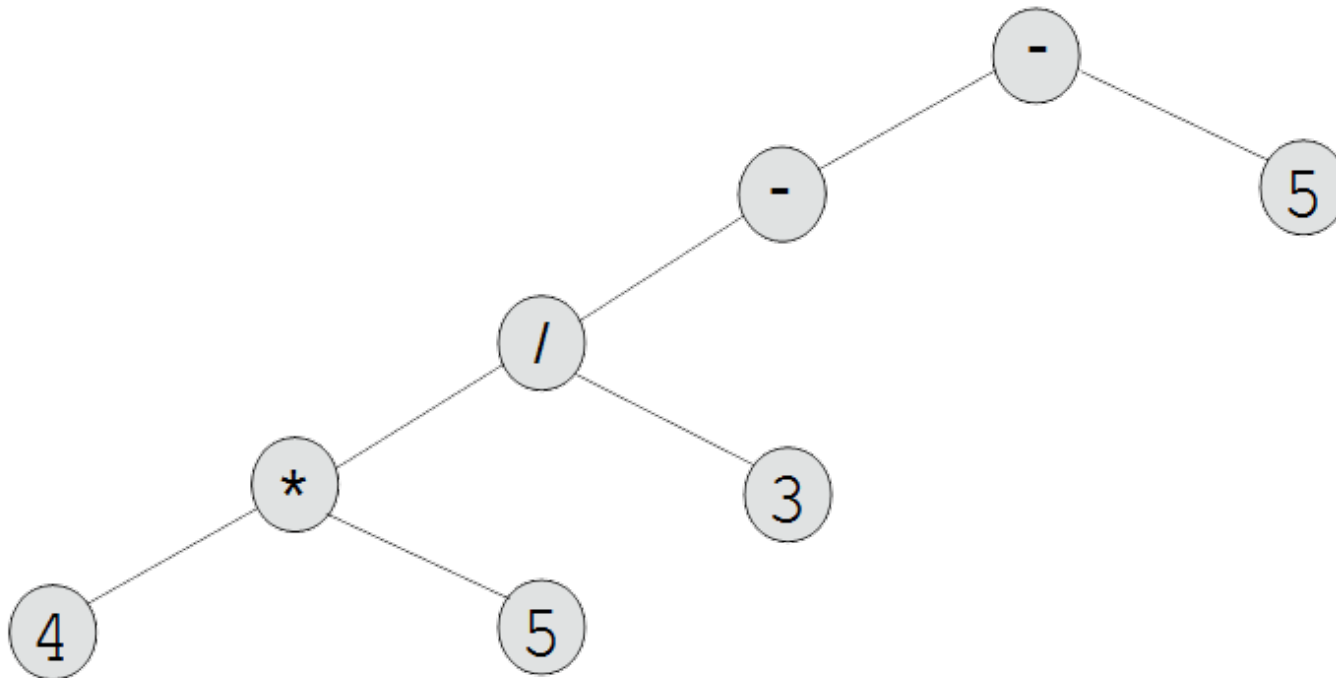
Interpretation of processing plans

Left/right deep syntax tree

Left deep syntax tree is a syntax tree such that the right argument of each operation is an atomic value

$(-((4 * 5) / 3)) - 5$

Left deep arithmetic expression

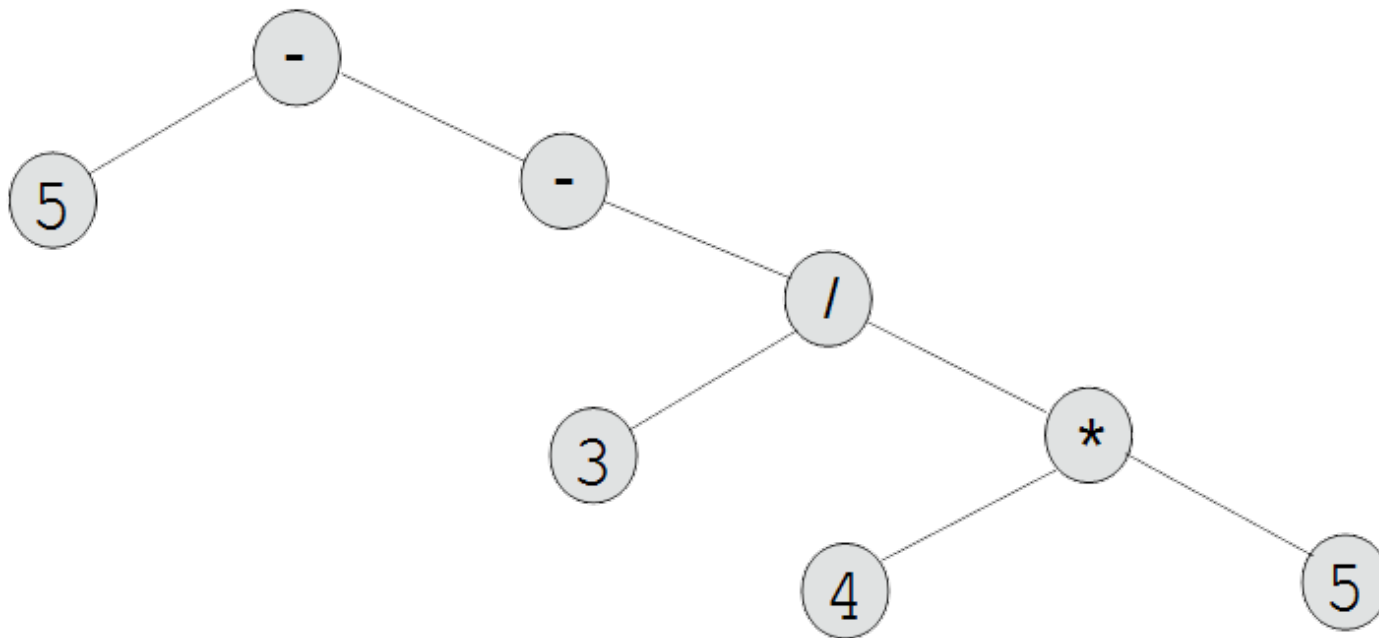


Left/right deep syntax tree

Right deep syntax tree is a syntax tree such that the left argument of each operation is an **atomic value**

5 - (- (3 / (4 * 5)))

Right deep arithmetic expression

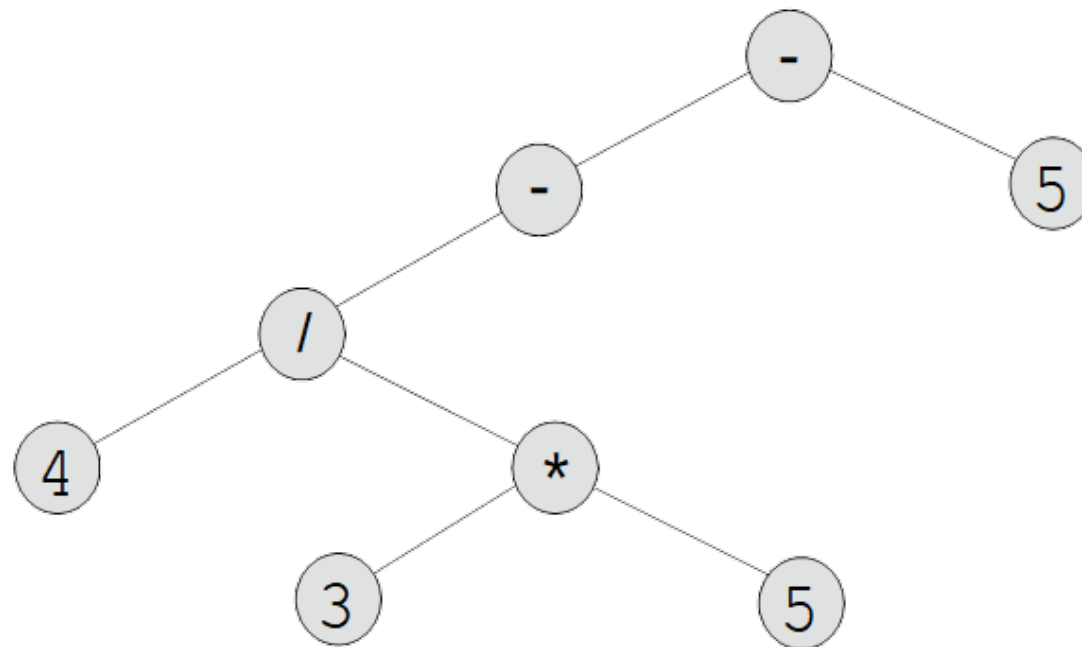


Left/right deep syntax tree

Left/right deep syntax tree is a syntax tree such that the right/left argument of each operation is an atomic value

$(-(4 / (3 * 5))) - 5$

Left/right deep arithmetic expression



Query Processing Plans

Outline

Syntax tree

Left/right deep syntax tree

EXPLAIN PLAN statement of SQL

Interpretation of processing plans

EXPLAIN PLAN statement of SQL

EXPLAIN PLAN statement of SQL lists a query processing plan created by a query optimizer for a given SQL statement

```
EXPLAIN PLAN FOR  
SELECT *  
FROM NATION;
```

EXPLAIN PLAN statement

```
SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY);  
or  
@showplan
```

Displaying processing plan

PLAN_TABLE_OUTPUT

A fragment of processing plan

Plan hash value: 79620726

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		25	2475	9 (0)	00:00:01
1	TABLE ACCESS FULL	NATION	25	2475	9 (0)	00:00:01

Query Processing Plans

Outline

Syntax tree

Left/right deep syntax tree

EXPLAIN PLAN statement of SQL

Interpretation of processing plans

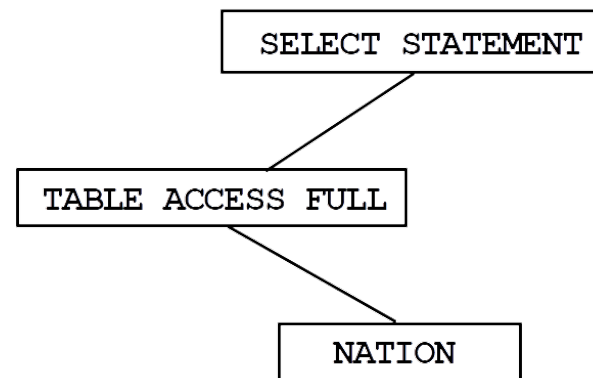
Interpretation of processing plan

A fragment of processing plan

PLAN_TABLE_OUTPUT

Plan hash value: 79620726

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		25	2475	9 (0)	00:00:01
1	TABLE ACCESS FULL	NATION	25	2475	9 (0)	00:00:01



EXPLAIN PLAN statement of SQL

```
EXPLAIN PLAN FOR
SELECT *
FROM LINEITEM
WHERE L_ORDERKEY = 7;
```

EXPLAIN PLAN statement

Processing plan

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		4	500	4 (0)	00:00:01
1	TABLE ACCESS BY INDEX ROWID BATCHED	LINEITEM	4	500	4 (0)	00:00:01
* 2	INDEX RANGE SCAN	LINEITEM_PKEY	4		3 (0)	00:00:01

Predicate Information (identified by operation id):

PLAN_TABLE_OUTPUT

2 - access("L_ORDERKEY"=7)

Interpretation of processing plan

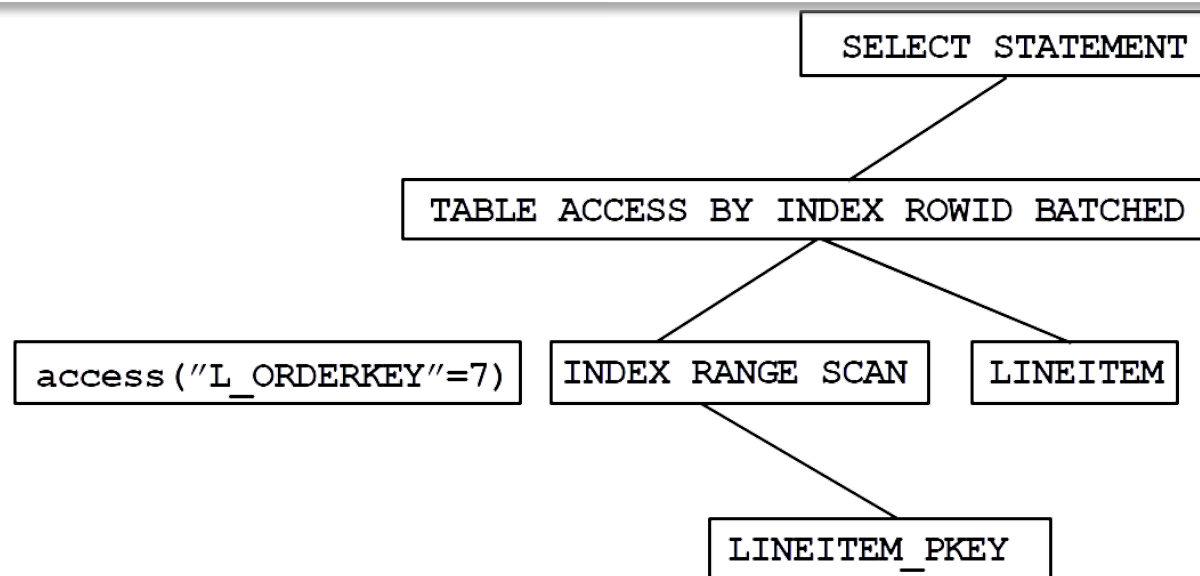
Processing plan

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		4	500	4 (0)	00:00:01
1	TABLE ACCESS BY INDEX ROWID BATCHED	LINEITEM	4	500	4 (0)	00:00:01
* 2	INDEX RANGE SCAN	LINEITEM_PKEY	4		3 (0)	00:00:01

Predicate Information (identified by operation id):

PLAN_TABLE_OUTPUT

2 - access("L_ORDERKEY"=7)



EXPLAIN PLAN statement of SQL

```
EXPLAIN PLAN FOR
SELECT *
FROM NATION JOIN REGION
      ON N_REGIONKEY = R_REGIONKEY;
```

EXPLAIN PLAN statement

Processing plan

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		25	4900	11 (0)	00:00:01
1	MERGE JOIN		25	4900	11 (0)	00:00:01
2	TABLE ACCESS BY INDEX ROWID	REGION	5	485	2 (0)	00:00:01
3	INDEX FULL SCAN	REGION_PKEY	5		1 (0)	00:00:01
* 4	SORT JOIN		25	2475	9 (0)	00:00:01
* 5	TABLE ACCESS FULL	NATION	25	2475	9 (0)	00:00:01

Predicate Information (identified by operation id):

```
4 - access("N_REGIONKEY"="R_REGIONKEY")
    filter("N_REGIONKEY"="R_REGIONKEY")
5 - filter("N_REGIONKEY">=0)
```

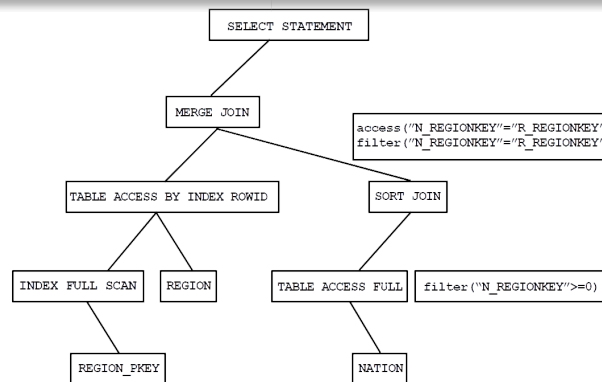
Interpretation of processing plan

Processing plan

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		25	4900	11 (0)	00:00:01
1	MERGE JOIN		25	4900	11 (0)	00:00:01
2	TABLE ACCESS BY INDEX ROWID	REGION	5	485	2 (0)	00:00:01
3	INDEX FULL SCAN	REGION_PKEY	5		1 (0)	00:00:01
* 4	SORT JOIN		25	2475	9 (0)	00:00:01
* 5	TABLE ACCESS FULL	NATION	25	2475	9 (0)	00:00:01

Predicate Information (identified by operation id):

4 - access("N_REGIONKEY"="R_REGIONKEY")
 filter("N_REGIONKEY"="R_REGIONKEY")
 5 - filter("N_REGIONKEY">=0)



EXPLAIN PLAN statement of SQL

```
EXPLAIN PLAN FOR
SELECT *
FROM LINEITEM JOIN ORDERS
ON L_ORDERKEY = O_ORDERKEY;
```

EXPLAIN PLAN statement

Processing plan

Id	Operation	Name	Rows	Bytes	TempSpc	Cost (%CPU)	Time
0	SELECT STATEMENT		1776K	396M		24979 (1)	00:00:01
* 1	HASH JOIN		1776K	396M	51M	24979 (1)	00:00:01
2	TABLE ACCESS FULL	ORDERS	450K	46M		1950 (1)	00:00:01
3	TABLE ACCESS FULL	LINEITEM	1800K	214M		8788 (1)	00:00:01

Predicate Information (identified by operation id):

1 - access("L_ORDERKEY"="O_ORDERKEY")

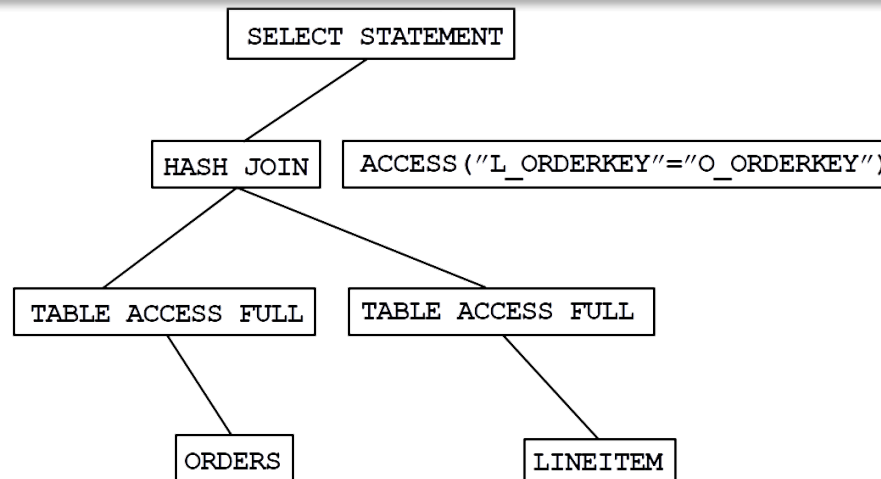
Interpretation of a plan

Processing plan

Id	Operation	Name	Rows	Bytes	TempSpc	Cost (%CPU)	Time
0	SELECT STATEMENT		1776K	396M		24979 (1)	00:00:01
* 1	HASH JOIN		1776K	396M	51M	24979 (1)	00:00:01
2	TABLE ACCESS FULL	ORDERS	450K	46M		1950 (1)	00:00:01
3	TABLE ACCESS FULL	LINEITEM	1800K	214M		8788 (1)	00:00:01

Predicate Information (identified by operation id):

1 - access("L_ORDERKEY"="O_ORDERKEY")



References

[Cookbook, How to find SQL processing plans and how to use hints ?](#)

[SQL Tuning Guide, Part III Query Execution Plans](#)

L. Nossov, H. Ernst, V. Chupis, Formal SQL Tuning for Oracle Databases, Springer, 2016 (Available from UOW Library)

Oracle® Database SQL Tuning Guide 19c

G. Harrison Oracle Performance Survival Guide, Prentice Hall, 2010