

Task 3 (3 marks)

An objective of this task is to interpret a query processing plan created by a query optimizer and to draw a syntax tree of a query processing plan

Consider the following fragment of query processing plan.

Id	Operation	Name	Rows	Bytes	TempSpc	Cost (%CPU)	Time
0	SELECT STATEMENT		284K	89M		12926 (1)	00:00:01
* 1	HASH JOIN		284K	89M	7200K	12926 (1)	00:00:01
2	TABLE ACCESS FULL	CUSTOMER	41861	6704K		390 (1)	00:00:01
* 3	HASH JOIN ANTI		284K	44M	41M	9808 (1)	00:00:01
* 4	TABLE ACCESS FULL	ORDERS	285K	37M		2698 (1)	00:00:01
* 5	INDEX FAST FULL SCAN	LINEITEM_PKEY	1943K	48M		1571 (1)	00:00:01

Predicate Information (identified by operation id):

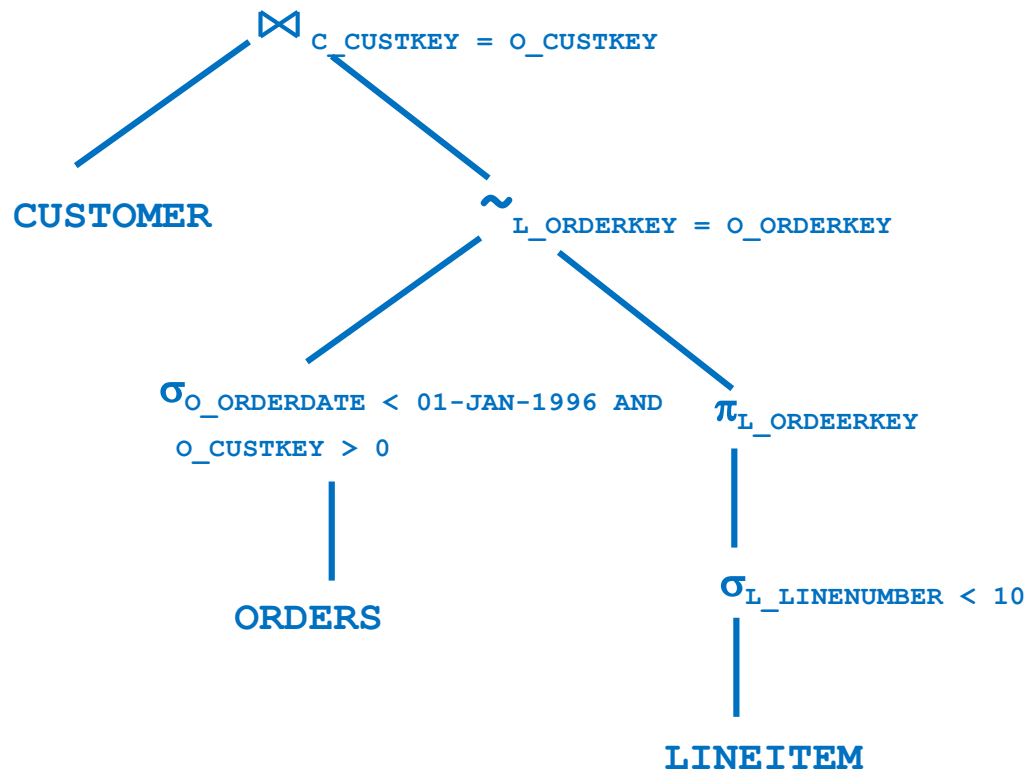
- 1 - access("O_CUSTKEY"="C_CUSTKEY")
- 3 - access("ORDERS"."O_ORDERKEY"="L_ORDERKEY")
- 4 - filter("ORDERS"."O_ORDERDATE"<TO_DATE(' 1996-01-01 00:00:00', 'yyyy-mm-dd hh24:mi:ss') AND "O_CUSTKEY">=0)
- 5 - filter("L_LINENUMBER"<10)

Find and draw a syntax tree of the query processing plan listed above. To draw a syntax tree, use the relational algebra operations explained during the lecture classes. Assume that the operations HASH JOIN and HASH JOIN ANTI used in a query processing plan is the same as the operations of join and antijoin in the relational algebra. Please remember, that you must create a syntax tree with the relational algebra operations explained to you during the lecture classes and NOT with the implementations of such operations by Oracle database system. Save a drawing of a syntax tree in a file `solution3.pdf`.

Deliverables

A file `solution3.pdf` with a drawing of syntax tree of the given query processing plan. A syntax tree must use the relational algebra operations explained to you during the lecture classes. You are allowed to use any line drawing tool to draw a syntax tree. A scanned/photographed copy of a neat hand drawing is also acceptable.

Solution



End of sample solution