Question #1: Query Plan Tree

In order to answer this question, we divided this pdf in 4 sections. The first one contains the query we are working on. The second one contains our drawing of the query plan. The third one explains each step of the plan.

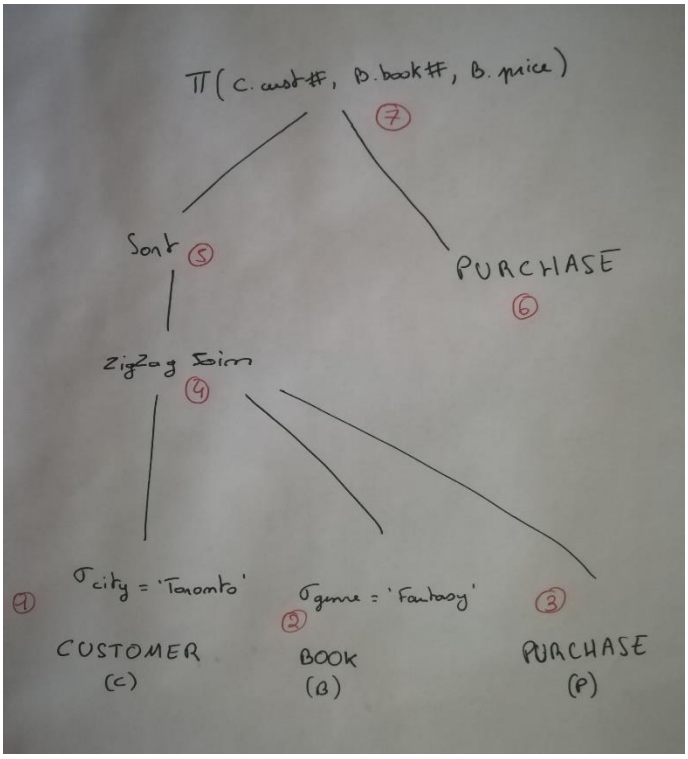
For reference we saved the actual db2 plan for this query in the file one.plan.

I. Query:

```
select C.cust#, B.book#, B.price
from stl.Customer C, stl.Purchase P, stl.Book B
where C.cust# = P.cust#
  and P.book# = B.book#
  and B.genre = 'Fantasy'
  and C.city = 'Toronto'
  and when between '11/01/2013' and '12/31/2013';
```

In order to avoid ambiguities, we are going to explain this query. This query returns a table with 3 columns customer_did, book_id, and book_price. Basically, a list of all the purchased books by customer with the buying price. The book must be of the Fantasy genre, the customer must live in Toronto and must have purchased the book between '11/01/2013' and '12/31/2013'.

II. Query Plan Tree



This tree is a simplified version of the real query plan. As requested, it is in the style of the course's textbook.

Each major operation is tagged with a number in a red circle. In the next section we will explain what this plan is doing more concretely. Although from this version we can already get a good idea.

III. Operations Analysis

1. CUSTUMER TABLE SCAN: This operation is very simple and straightforward; a table scan is performed since there are no available indexes for this table. When retrieving rows, the planner filters the table only choosing the customers that live in Toronto.
2. BOOK TABLE SCAN: Similar to the previous operation; a table scan is performed for the same reasons, also filtering the table by only selecting books that belong to the Fantasy genre.
3. PURCHASE INDEX SCAN: Since the PURCHASE table is so big the query planner decides to scan the clustered index on cust#, book# and when so that we only have the purchased books from '11/01/2013' to '12/31/2013'.
4. ZIGZAG JOIN[1]: Now that we have our tree tables filtered, we are joining them using the ZigZag join operation.
5. TABLE SORTING: We perform a sort on the table by ascending order of custumer_id as specified by db2.
6. PURCHASE TABLE SCAN: A simple table scan to retrieve all rows from this table.
7. JOIN: For this last join operation we want to filter the customers that haven't bought books between '11/01/2013' and '12/31/2013'.
8. RETURN: Not showing in the picture but after the join our final table is returned.

[1] Zigzag join: which joins a fact table and two or more dimension tables in a star schema. The columns joining dimension tables with the fact table must be unique. Queries that use an all-probes list-prefetch plan for the fact table access will include two separate ZZJOIN operators. One of those two ZZJOIN operators represents a back-join between the fact table and the dimension tables.