Task 2

Let's take a look at the runtimes of the 12 different combinations of storage layout + execution model. The runtimes corresponds to the 10 queries executed in a row (no warmup or micro-queries) in milliseconds :

	NSM	**DSM**	**PAX**
Tuple at a time	991	870	927
Operator at a time	1547	1048	1369
Block at a time	1190	942	938
Late operator at a time	2144	1623	1774

The fastest combination is DSM storage + Volcano-style tuple at a time execution, while the slowest is NSM + late materialization operator at a time.

Storage layouts analysis : On average, the DSM storage layout gives the best results while the NSM layout is the worst. It is interesting to note that the cases where the difference is the most notable is when NSM is combined with an operator at a time model, which results in a significant difference when compared to DSM. This makes sense since transforming data organized by rows to be suitable for the operator at a time model requires extra work.
When combined with the 2 other execution models however (tuple/block at a time), the difference between the 3 storage layouts is of a way smaller magnitude, with DSM and PAX having almost similar runtimes.

Execution model analysis : On average, the early tuple at a time model gives the best results while the late operator at a time model consistently produces the worst results. However, it should be noted that this heavily depends on implementation details. My approach sometimes reuses the logic from the early materialization procedures, which means that it sometimes needs to materialize the data from the IDs in the middle of the process, thus requiring more time. It should be possible to have better runtimes in the late materialization model with a different approach.