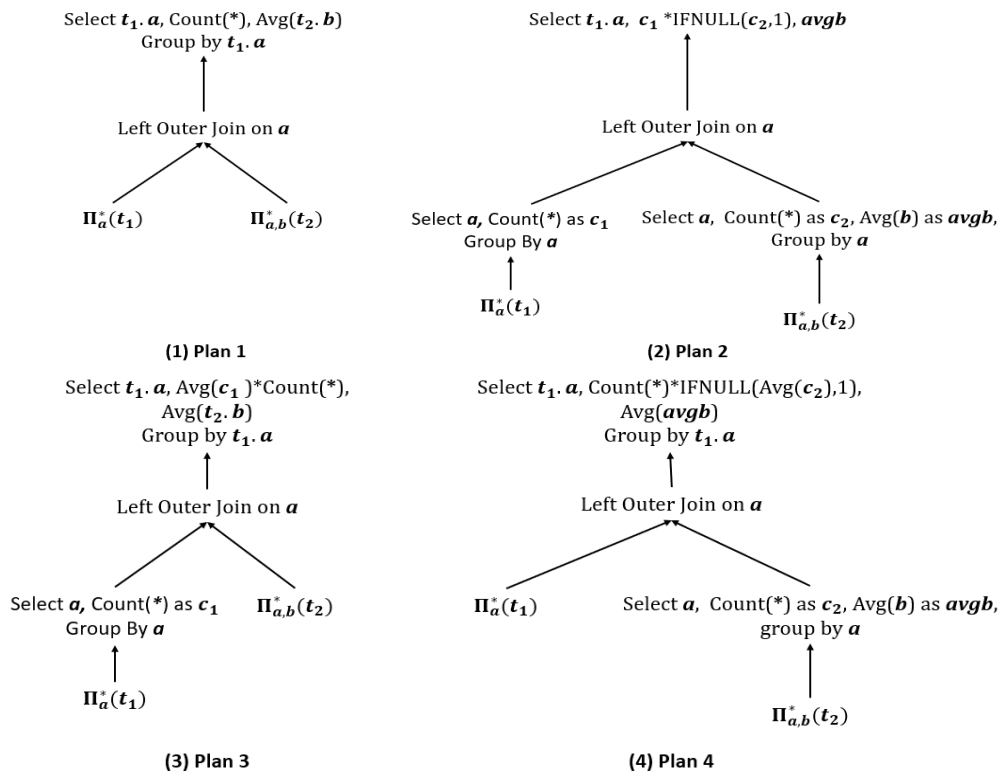


1. Query Plans

由启发式算法可知，投影应该尽早进行，在这个query中，需要投影的列一是需要group的 $t1.a$ 和 $t2.a$ ，二是在group完成后需要参与aggregate计算的 $t2.b$ 。故投影部分就不再在以下查询执行计划中讨论了。该query按语义转换过来可得到下图中的plan 1，由于这个query只涉及到两张表，所以group by操作符是可以无条件下滤的，分别可得到下图中的plan 2、3、4。

$\Pi_{attr\{ \}}^*(r)$ -- Projection without duplicate elimination



group by通过减少数据冗余（用一行记录代表一个group的信息），从而优化查询。但特殊的，如果 a 是 $t1$ 的主键(执行group by a 后每个group里只有一行记录)，group by是没有优化效果的，plan 3就退化成了plan 1。如果 a 是 $t2$ 的主键，plan 4就退化成了plan 1。但由于group by primary key基本没什么成本，所以即使退化，对整体的执行性能几乎不会有损失。对这个query来说，group by操作越早执行越好，故plan 2是最佳的逻辑执行方案。

2. Join和Group方案选择

join的方案可选择scan join、sort-based merge join、hash join，而group的方案可选择hash-based和sort-based。按照数据是否有序是否重复，可将 $t1.a$ 的数据分布分为以下四种类型：

1. 有序且不重复。此时对 $t1.a$ 提前执行group by操作无优化作用，但由于有序，故采用sort-based group by操作带来的额外开销很小。
2. 有序且重复。此时对 $t1.a$ 提前执行group by操作能起到明显的去冗余的优化效果。且由于有序，提前采用sort-based group by操作的成本开销也很小。
3. 无序且不重复。此时对 $t1.a$ 提前执行group by操作起不到明显的优化效果。在无序的情况下，提前执行group by操作还需要一定的开销，反而可能会降低query执行的性能。

4. **无序且重复**。此时对**t1.a**提前执行group by操作可以起到明显的去冗余的优化效果，虽然在无序的情况下，提前执行group by操作需要一定的开销，但可以通过利用group by的结果用到后面的left outer join操作中去。由于**t1.a**既参与了group by操作，又参与了join操作，最好group by执行的方式和join执行的方式一致。比如**t2.a**是无序的，那**t1.a**在提前执行group by操作时，可采用hash的方式，得到的hash表可直接用于和**t2.a**的left outer join。如果**t2.a**是有序的，那**t1.a**在提前执行group by操作时，可采用sort based的方式，在后面与**t2.a**left outer join的时候，可直接采用merge join的方式。

下图是对**t1.a**和**t2.a**在不同数据分布下的一些简单考虑：

	t1 (有序-不重复)	t1 (有序-重复)	t1 (无序-不重复)	t1 (无序-重复)
t2 (有序-不重复)	group-by下滤无优化 join和group采用sort-based	先对t1执行group有优化 join和group采用sort-based	group-by下滤无优化 join和group采用sort-based	先对t1执行group有优化 join和group采用sort-based
t2 (有序-重复)	先对t2执行group有优势 join和group采用sort-based	先对t1和t2分别执行group join和group采用sort-based	先对t2执行group有优化 先对t2 hash group, 再scan join	先对t1和t2分别执行group t1 t2都sort group, 在sort join
t2 (无序-不重复)	group-by下滤无优化 先hash join了后再sort-based group	先对t1执行group有优化 先对t1 group, 再hash join t2	group-by下滤无优化 先hash join再hash group	先对t1执行group有优化 t1 hash group再hash join t2
t2 (无序-重复)	先对t2执行group有优势 先t2 sort-based group再sort join 再sort group	先对t1和t2执行group t1 sort group, t2 sort group, 后merge join	先对t2执行group有优化 先对t2 sort group然后再sort join 然后再sort group	先对t1和t2分别执行group t1 t2分别hash group, 再hash join