Query Tree

查询处理过程

在 PostgreSQL 中，一条 SQL 语句从接收到执行一共需要经过上图的五个主要步骤

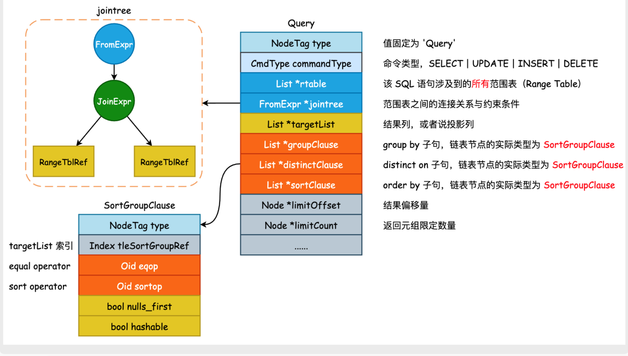
子系统

1. 解析器，根据 SQL 语句生成一棵语法解析树（parse tree），这个阶段可以类比于编译过程中的词法分析和语法分析
2. 分析器，根据语法解析树进行语义分析，生成一棵查询树（query tree）。在这个阶段会判断 SQL 语句中的表、列是否存在
3. 重写器，按照规则系统存在的规则对查询树进行重写，主要是逻辑重写
4. 计划器，基于查询树生成一棵执行效率最高的计划树（plan tree），基于代价（cost）的优化就发生在这个阶段，同时也是查询处理最为复杂的阶段
5. 执行器，按照计划树中的顺序访问表和索引，执行相应查询，从硬盘或者是内存中取出相应的数据

计划器通常也被称之为查询优化器，这东西可以说是 RDBMS 的精髓所在。而优化器的输入是一棵查询树，并且该查询树在整个查询优化的过程中会被频繁使用，所以理解查询树的结构非常有必要

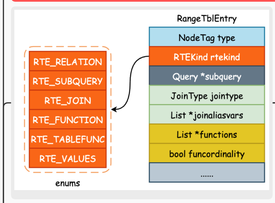
基本组成

在 PostgreSQL 中，一棵查询树由 Query 结构体表示，其大致组成如下图所示



rtable

rtable 是 range table 的缩写，它不仅仅包含位于硬盘中的实体表（relation），还会包括视图、子查询等一切可以作为数据来源的对象。rtable 是一个链表，记录了查询语句中 FROM 子句后面指出的所有范围表，rtable 由结构体 RangeTblEntry 表示

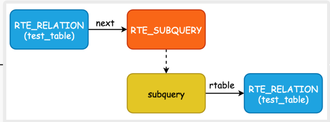


如左图所示，RangeTblEntry 更像是一个"大杂烩"，所有类型的范围表均使用同一个结构体表示，那么不同类型的范围表所使用到的字段也不尽相同

例如对于 RTE_SUBQUERY 来说，会使用到 subquery 字段；对于 RTE_JOIN 来说，会使用到 jointype 和 joinaliasvars 字段；对于 RTE_FUNCTION 来说，会使用到 function 和 funcordinality 字段

Example-1

SQL: select * from test_table where (i, tz) in (select distinct on (i) i, tz from test_table) order by i;

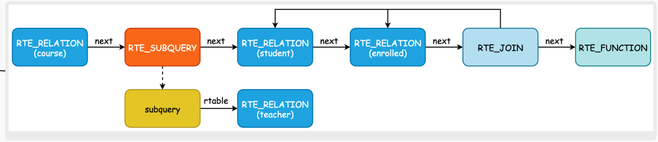


上述 SQL 语句包含了一个子查询（select distinct on .....），因此 rtable 链表的最后一个 RangeTblEntry 的 rtekind 为 RTE_SUBQUERY

而对于子查询来说，它内部没有更多的子查询了，因此其 rtable 只有一个节点，且类型为 RTE_REALTION

Example-2

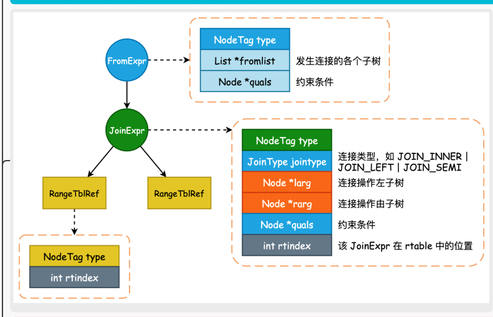
select * from course, (select * from teacher) as t, student left join enrolled on student.sid = enrolled.sid, generate_series(1,10) as func(i);



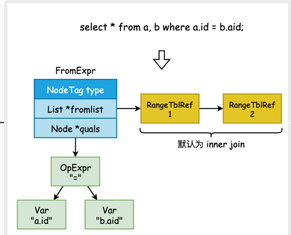
虽然 rtable 是一个 List，也就是一个链表，但实际上可以把它当做是一个静态数组：链表大小和元素位置一经确认将不会发生改变，因此在 jointree 中还会使用 index 来索引 RangeTblEntry

jointree

jointree 则表示范围表之间的连接关系和约束条件，一共有 3 种类型的节点：FromExpr、JoinExpr 和 RangeTblRef。如上述 Query 结构图所示，其中 FromExpr 为 jointree 的顶层节点，JoinExpr 为非叶子节点，RangeTblRef 则是叶子节点



对于 FromExpr 来说，存储了连接和的来条件，这既包含了基本约束条件，如 age >= 25、user_id = '10001' 等，同时也包含了连接（join）信息。基本约束条件则保存在 quals 中，通常是一个链表；连接信息则保存在 fromlist 中，该链表中的节点默认是 inner join 的关系



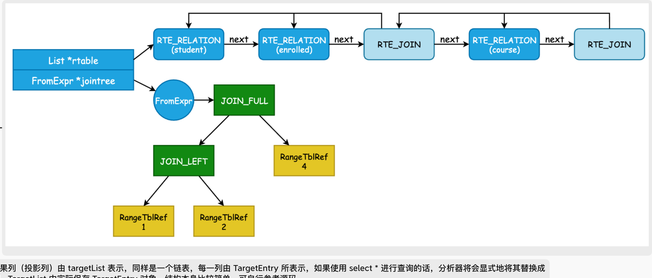
如左图所示，当我们未显式指定连接类型时，a、b 两张表将会作为链表中的两个节点挂接到 fromlist 上，表示内连接关系，而连接的约束条件则保存在 quals 中

在 JoinExpr 中存在 larg 和 rarg 这两个左、右孩子指针，因此作为非叶子节点挂接 JoinExpr 和 RangeTblExpr

RangeTblRef 是 jointree 的叶子节点，本质上其实就是对 RangeTableEntry 的引用，rtindex 就是 RangeTblEntry 在 rtable 链表中的索引位置

Example-1

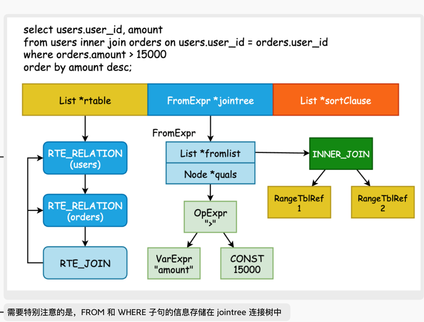
select * from student left join enrolled on student.student_id = enrolled.student_id full join course on enrolled.course_id = course.course_id;



targetList

查询语句的结果列（投影列）由 targetList 表示，同样是一个链表，每一列由 TargetEntry 所表示，如果使用 select * 进行查询的话，分析器将会显式地将其替换成所有具体的列。TargetList 中实际保存 TargetEntry 对象，结构本身比较简单，可自行参考源码

小结



需要特别注意的是，FROM 和 WHERE 子句的信息存储在 jointree 连接树中