# delete的解析与执行

# delete语句的解析

1. parse\_sql\_stmt\_delete 函数参数是一个解析器，返回一个sql\_stmt\_delete结构体，结构体中包含一个char\* tablename用来保存表名，还有一个SRA\_T类型用来保存逻辑查询计划树。
2. 首先是解析 delete 词，使用matchToken()函数进行解析
3. 其次解析from 关键词，同样是使用matchToken()函数进行解析
4. 接下来解析表名，使用new\_id\_name() 为一个字符串指针开辟空间（包含了字符串的最大长度）,把表名赋值给结构体的字段
5. 需要注意的是matchToken()函数在解析关键词成功之后，会将parser中的token替换为下一个Token,而匹配表名的时候我们没有使用matchtoken函数，因此需要手动获取下一个token，token=parseEatAndNextToken(parser);
6. 接下来匹配where关键词，如果匹配到就使用parseExpressionRD函数解析where 后面的表达式，否则就不再进行解析
7. 使用TableReference\_make函数利用tablename创建一个TableReference\_t结构体，之后使用SRATable()函数创建一个表示全部元组的集合，接下来如果where子句不为空的话，就就利用SRATable创建一个SRASelect类型
8. 为sql\_stmt\_delete的各部分赋值，并返回

# delete语句的执行

plan\_execute\_delete 是delete语句的物理执行函数。只要将sql\_stmt\_delete中的逻辑查询计划树转换成物理查询计划树，然后执行就可以了。生成物理查询计划树的函数是physical\_scan\_generate 。

1. 使用 physical\_scan\_generate 函数传入sql\_stmt\_delete->where ，tx事务，生成物理查询计划树
2. 使用scan->beforeFirst(scan) 初始化读取位置到记录的第一条的位置
3. 使用循环while(scan->next(scan)) 遍历满足条件的记录，使用deleterec进行记录的删除
4. 关系物理计划scan->close(scan)
5. 返回成功DONGMENDB\_OK(0)



