

```
table_item->ref_id_改为索引表的tid。这样这个屋里计划就只查索引表,不查原表了。
 如果有表别名,这里的
 table_item>table_id_是表别名的tid。如果没有表别名,则table_item>table_id_是索
 引表的tid。
 3. 根据表达式数组project_array里面的表达式生成原表的行描述,将该行描述存到操作符
 rpc_scan里面。
 4. sql_read_strategy根据rowkey和filter确定该表是使用scan方法或是get方法来获取数据
table_rpc_scan_op->init(sql_context_, &hint);
🧖 6. table_scan_op->add_filter(filter);//add前要把filter里面存的列的tid改成索引表的tid,否
/ 则会报错。如果该表有别名,则不需要改filter的列的tid
     |---> ObTableRpcScan::add_filter()
           |---> rpc_scan_.add_filter(expr)
               |---> read_param_->add_filter()
          |---> select_get_filter_.add_filter();
 7. table_scan_op->add_output_column(output_expr); //add前要把output_expr里面存的列
  的tid改成索引表的tid,否则会报错。如果该表有别名,则不需要改filter的列的tid
      |---> ObTableRpcScan::add_output_column(expr);
           |---> rpc_scan_.add_output_column(expr);
```

CREATE_PHY_OPERRATOR(table_rpc_scan_op, ObTableRpcScan, physical_plan, err_stat);

2. table_rpc_scan_op->set_table(table_item->table_id_, table_item->ref_id_); //将

```
|---> read_param_->add_output_column(expr);
                |---> cur_row_desc_.add_column_desc();
1. CREATE PHY OPERRATOR(table rpc scan op, ObTableRpcScan, physical plan, err stat);
2. table_rpc_scan_op->set_table(table_item->table_id_,index_tid_without_storing); //这里把tableid改为索引表的tid,因为第一次scan的是索引表。
3. table_rpc_scan_op->set_main_tid(table_item->ref_id_);//这里把原表的tid先存下来,第二次get原表的时候会用到。
4. table_rpc_scan_op->set_is_use_index_without_storing();//表明该操作符实现的是回表的索引,有两次查询。
5. table_rpc_scan_op->set_main_rowkey_info(main_table_schema->get_rowkey_info()); //把原表的主键信息存下来,第二次get时会用到。
6. sql_read_strategy根据rowkey和filter确定该表是使用scan方法或是get方法来获取数据
table_rpc_scan_op->init(sql_context_, &hint);
8. table_scan_op->add_main_filter(filter); //把where条件中的表达式先存下来,第二次get的时候会用到。
9. 判断where条件中的每个表达式,如果该表达式里面的列的cid在索引表中,则修改下该表达式的tid,然后调用下面的table_scan_op->add_filter(filter);把表达
式存起来,以便第一次scan的时候用
 table_scan_op->add_filter(filter);
    |---> ObTableRpcScan::add_filter()
           |---> rpc_scan_.add_filter(expr)
                |---> read_param_->add_filter()
          |---> select_get_filter_.add_filter();
10. table_scan_op->add_main_output_column(output_expr); //将原sql的输出信息存下来,第二次get原表时返回数据需要用到。
11. 获得索引表的主键信息,把主键的所有列封装成表达式,通过table_scan_op->add_output_column(output_expr)存起来,作为第一次scan索引表的输出列。
 table_scan_op->add_output_column(output_expr);
    |---> ObTableRpcScan::add_output_column(expr);
           |---> rpc_scan_.add_output_column(expr);
                |---> read_param_->add_output_column(expr);
                |---> cur_row_desc_.add_column_desc();
12. 遍历where条件中的每一个表达式,如果该表达式的列的cid在索引表的非主键中,通过table_scan_op->add_output_column(output_expr)存起来,作为第一次
scan索引表的输出列。如果该表达式的列的cid不在索引表中,调用table_scan_op->add_index_filter(filter)把该表达式存起来。第二次get原表的时候会用做过滤。
```