

物理算子及火山模型

由朱悦铭于2025年设计

代码框架贡献者：张子阳

在数据库系统的**基础场景**中，即上个星期的练习题中，用户可通过简单的指令（如 `SELECT * FROM t;`）直接获取全表数据，或通过条件筛选（如 `SELECT name, age FROM t WHERE gpa > 3.0;`）提取特定结果。这类操作看似直观，但其底层实现需面对以下挑战：

1. 功能泛化性：

数据库需支持**任意复杂度的查询组合**（如多层过滤、连接、聚合、排序等），单一的代码逻辑无法覆盖所有场景。

2. 执行效率与资源管理：

若直接通过硬编码实现所有功能，会导致代码臃肿、内存占用不可控（例如全表加载）、难以优化局部逻辑。

3. 模块化与可扩展性：

新增指令类型（如Count、Delete）等，需避免对核心引擎的侵入式修改。

为解决上述问题，火山模型提出**物理算子抽象化**的思想，后续我们会依次介绍物理算子与火山模型的概念。

一、物理算子

在数据库执行引擎中，**复合运算的实现遵循单一职责原则**：将每个独立操作抽象为**物理算子**（Physical Operator），每个算子仅完成特定功能（如扫描、过滤、投影），并通过统一接口向上层返回处理后的元组（`Tuple`）。

示例解析：

```
SELECT name, age FROM t WHERE gpa > 3.0;
```

该查询可分解为三个物理算子的协同执行：

1. ScanOperator（表扫描算子）

- 职责：从存储引擎逐行读取表 `t` 的原始数据。

2. FilterOperator（过滤算子）

- 职责：应用谓词 `gpa > 3.0` 筛选符合条件的元组。

3. ProjectOperator（投影算子）

- 职责：裁剪字段，仅保留 `name` 和 `age` 列。

物理算子的接口标准化

为实现算子的可组合性，所有物理算子需遵循统一的迭代器接口。以Java语言为例，其核心方法定义如下：

```
public interface PhysicalOperator {  
    // 初始化算子（如打开文件、初始化子算子）  
    void Begin() throws DBException;  
  
    // 检查是否还有数据  
    boolean hasNext() throws DBException;  
  
    // 处理下一条数据（核心逻辑）  
    void Next() throws DBException;  
  
    // 获取当前元组  
    Tuple Current();  
  
    // 清理资源（如关闭文件）  
    void Close();  
  
    // 定义输出结果的元数据（列名、类型）  
    ArrayList<ColumnMeta> outputSchema();  
}
```

这里要主要强调两个数据类型：

1. Tuple:

即元组，是物理算子间传递数据信息的核心数据结构，其本质是一行数据的抽象表示。也是一个二进制数据向编程语言数据类型转换的媒介。一般来说，多个数据库管理系统（DBMS）中的火山模型结构，例如：MySQL, Vastbase, PostgreSQL等都是通过返回Tuple向上传递数据。

2. ColumnMeta:

记录要返回列的集合。如果投影为全表，那么返回该表中所有的列，如果投影为表的一部分列，那么返回的ColumnMeta集合也要对应修改。

二、火山模型

火山模型（Volcano Model）是数据库查询执行中的一种经典架构模式，也叫做**迭代器模式**（Iterator Model）逐条处理数据。是指将SQL语句拆分成不同的物理算子，并组合这些物理算子的调用顺序，形成自顶向下的层级结构。例如：投影 -> 过滤 -> 连接 -> 扫描等。上层算子通过，next()或 hasNext() 驱动下层算子获取数据，下层算子通过current() 返回Tuple 类型给上层算子。

例如下面的SQL语句：

```
select t.age, t.name from t where t.gpa > 3.0;
```

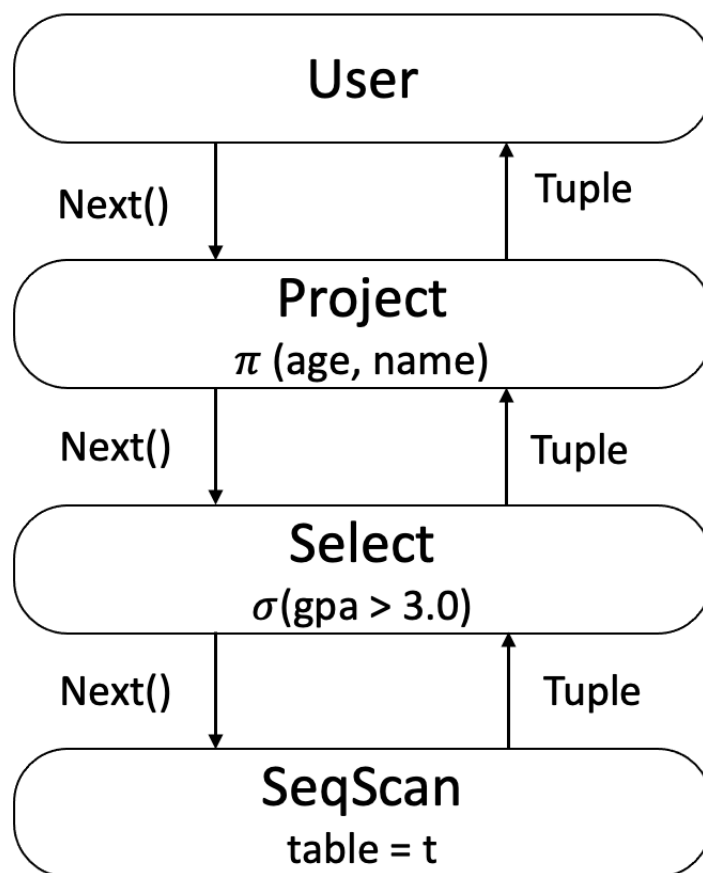
构建的逻辑算子为：

```
ProjectOperator(selectItems=[t.age, t.name])
└─ LogicalFilterOperator(condition=t.gpa > 3.0)
    └─ TableScanOperator(table=t)
```

构建的物理算子计划：

```
ProjectOperator(selectItems=[t.age, t.name])
└─ FilterOperator(condition=t.gpa > 3.0)
    └─ SeqScanOperator(table=t)
```

其火山模型对应的执行关系如下图所示：



练习

对于Delete语句，创建相关物理算子以及搭建火山模型，最后并实现。要求对于Delete语句要考虑到两种形式：

- `delete from t where t.id = 3;`
- `delete from t;`

设计通用的火山模型，要适配以上两种模式，并实现。

