1. 数据说明

我们采用如下三个表：customer表，nation表region表。

属性的数据类型

Integer (-2,147,483,646 , 2,147,483,647)

Fixed text, size N

Variable text, size <= MAX\_LENGTH

Date YYYY-MM-DD

表的定义

|  |  |  |
| --- | --- | --- |
| Column Name | Datatype | Requirements |
| custkey | Integer | Primary Key |
| cusname | variable text, size 25 |  |
| address | variable text, size 40 |  |
| nationkey | Integer | Foreign Key to NATION\_nationkey |
| phone | fixed text, size 15 |  |
| acctbal | fixed text, size 15 |  |
| mktsegment | fixed text, size 10 |  |
| cuscomment | variable text, size 117 |  |

CUSTOMER Table Layout

|  |  |  |
| --- | --- | --- |
| Column Name | Datatype | Requirements |
| nationkey | Integer | 25, Primary Key |
| naname | fixed text, size 25 |  |
| regionkey | Integer | Foreign Key to REGION\_regionkey |
| nacomment | variable text, size 152 |  |

NATION Table Layout

|  |  |  |
| --- | --- | --- |
| Column Name | Datatype | Requirements |
| regionkey | Integer | 5, Primary Key |
| regionname | fixed text, size 25 |  |
| regioncomment | variable text, size 152 |  |

REGION Table Layout

1. 实现的功能

支持的SQL语句

* 新建表：CREATE TABLE
  + 列属性：整型，字符型，日期型
* 删除表：DROP TABLE
* 插入行：INSERT
* 删除行：DELETE
* 查询表：SELECT

1. sql语法

<Query>::=<SFW>

<Query>::=SELECT <attribute\_list> FROM <table\_list> WHERE <where\_list>

<attribute\_list>::=<attribute[Array]>

<table\_list>::=<table[Array]>

<where\_list>::=<where\_list> AND <where\_list>

<where\_list>::=<where\_list> OR <where\_list>

<where\_list>::=<attribute> compare\_opt <condition\_const>

compare\_opt = > < >= <= like

ORDER BY

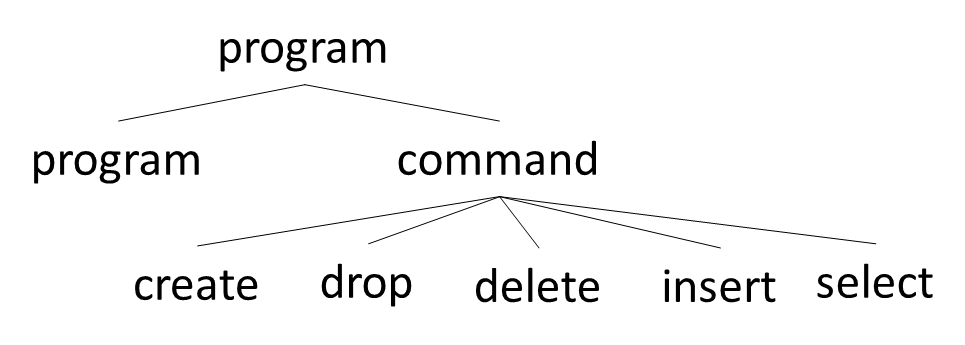
注1：小写为中间变量，大写为词法分析得到的终结token

注2：select语句要以分号作为结束

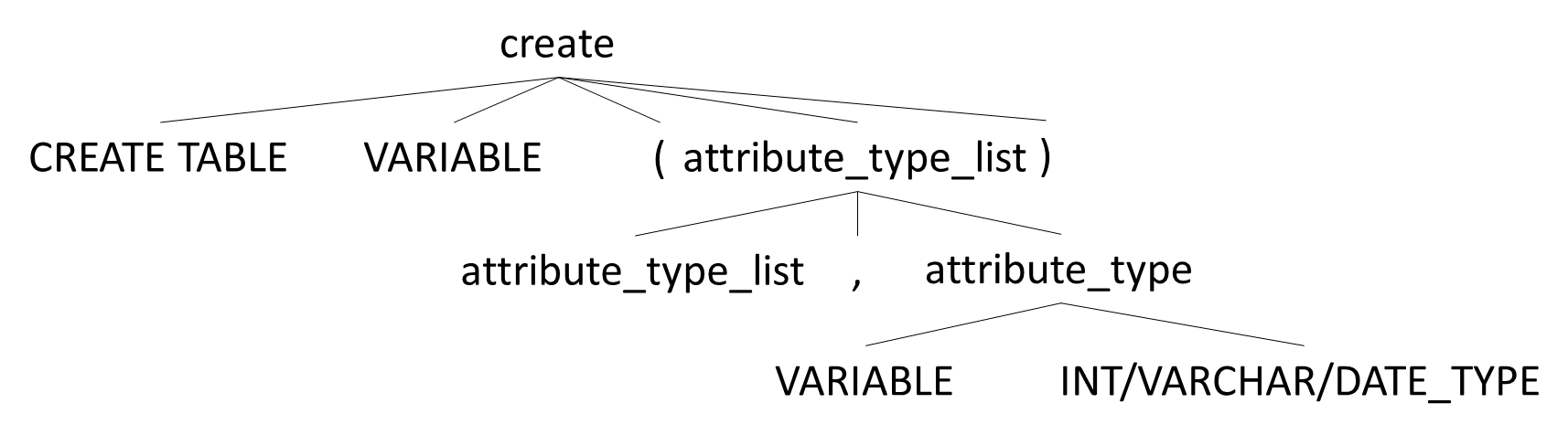
1. （几个能运行的）query说明 举例 画语法树，查询计划

语法树形式

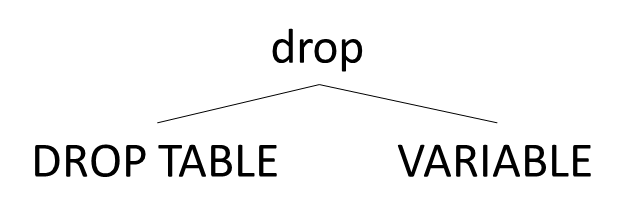
1. 顶层语法树：用于连续执行多条SQL语句



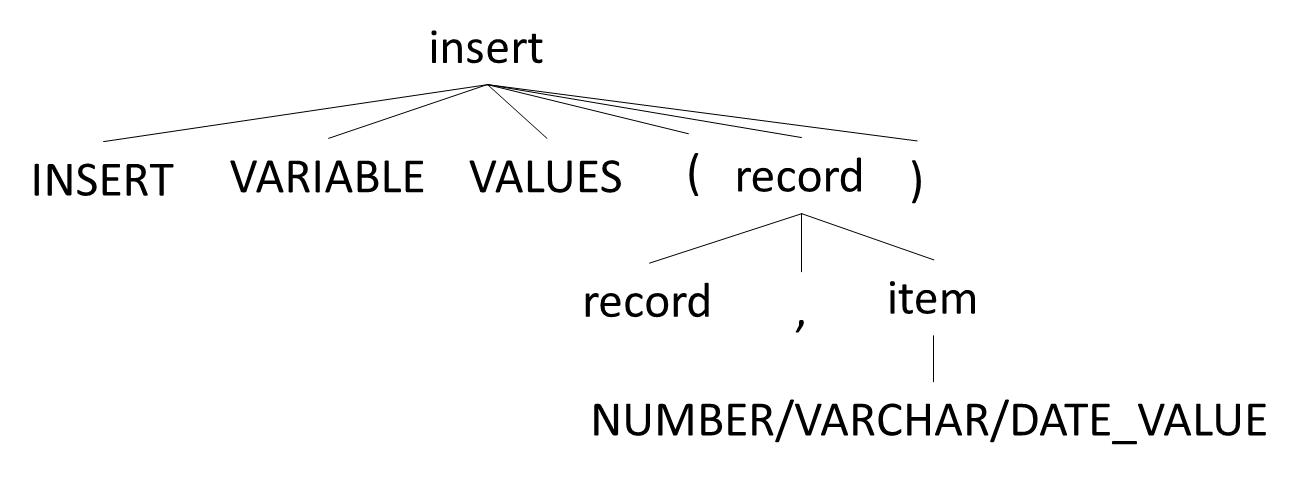
1. CREATE语句：



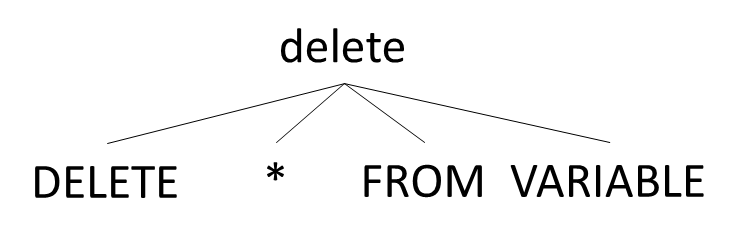
1. DROP语句：

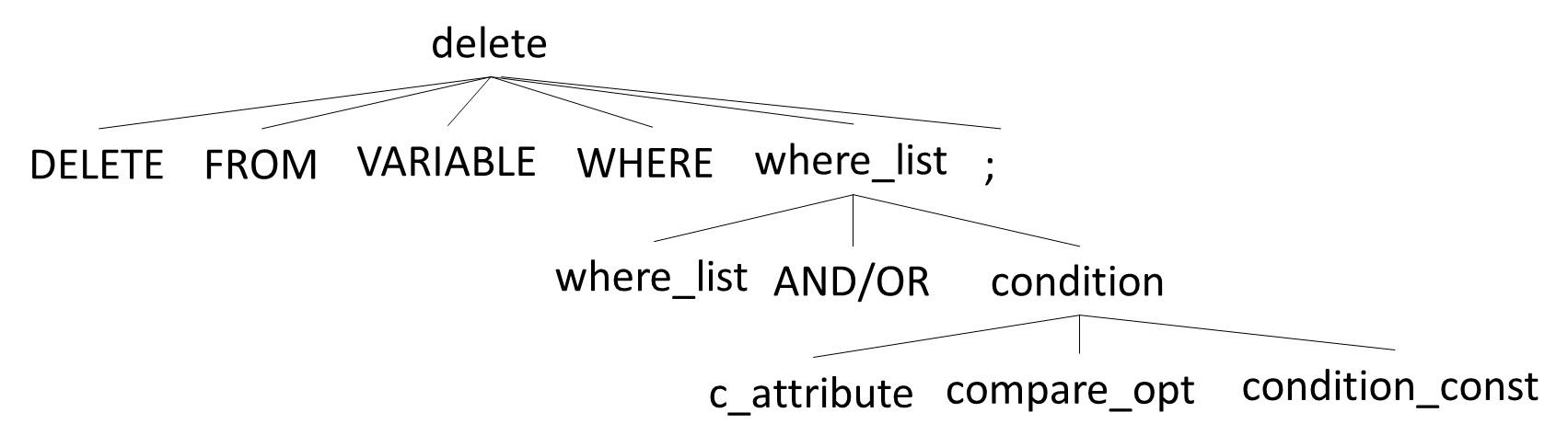


1. INSERT语句：



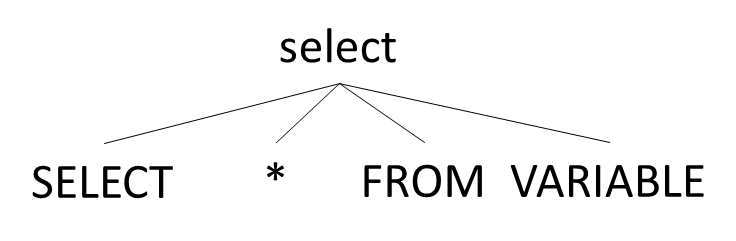
1. DELETE语句：

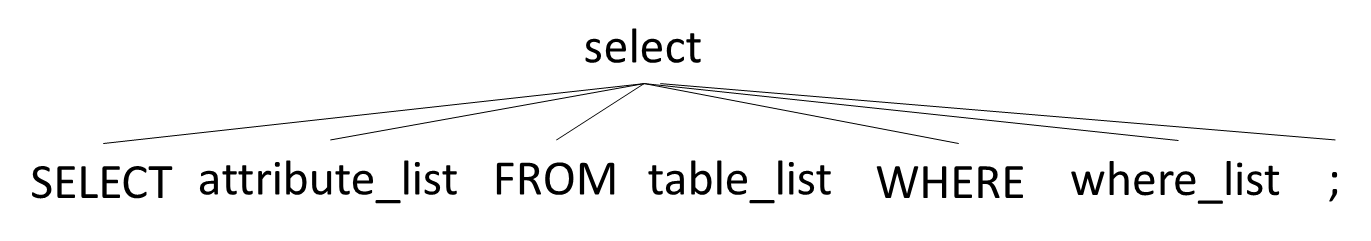




1. SELECT语句：

注：其中where语句与delete中类似，不赘述。





样例query

1. create table

CREATE TABLE student (sid int, sname varchar(10), birth date)

2. drop table

DROP student

3. insert

INSERT INTO student VALUES (2015, "cc", 1993-12-11)

4. delete

DELETE \* FROM student

DELETE FROM student WHERE sid=2015;

5. select

SELECT cusname, address FROM customer WHERE nationkey=13 AND custkey=2;

1. 查询优化
   1. 基于启发式规则的逻辑优化

编译时进行优化

<1>选择下推

在投影操作之前，将选择操作下推到每个表，使得表的大小显著降低，再对选择后的数据进行连接等后续操作。

<2>投影下推

我们在查询分析时，查询数据字典，得到每一个表在后期操作所全部需要的属性（包括条件where语句中出现的属性和select的属性），对每个表都在这些属性上做出投影操作。这样可以降低表的大小。

* 1. 基于代价的物理优化

查询执行时进行连接操作算法的选择

<1> 如果至少一个表有索引，就选择index join 连接方式，索引的表作为右表

<2> 如果其中一个表数据量很大，就选择hash join 连接方式，小表作左表，大表作右表

<3> 如果数据量不是特别大，并且两表数据量差不多，就选择sort merge join

查询执行时进行filter操作算法的选择？排序算法的选择？

1. 数据字典
2. 分模块
   1. 存储管理
   2. 索引
   3. cursor datadictionary
   4. create insert delete drop
   5. select: tablescan indexscan SPJ
   6. 查询执行：流水线，push

数据结构

函数调用图

模块之间的接口

主要内部函数

每人按如下格式写自己的函数：

＊＊＊＊＊＊＊＊＊

函数名，功能，参数含义，返回值

＊＊＊＊＊＊＊＊＊