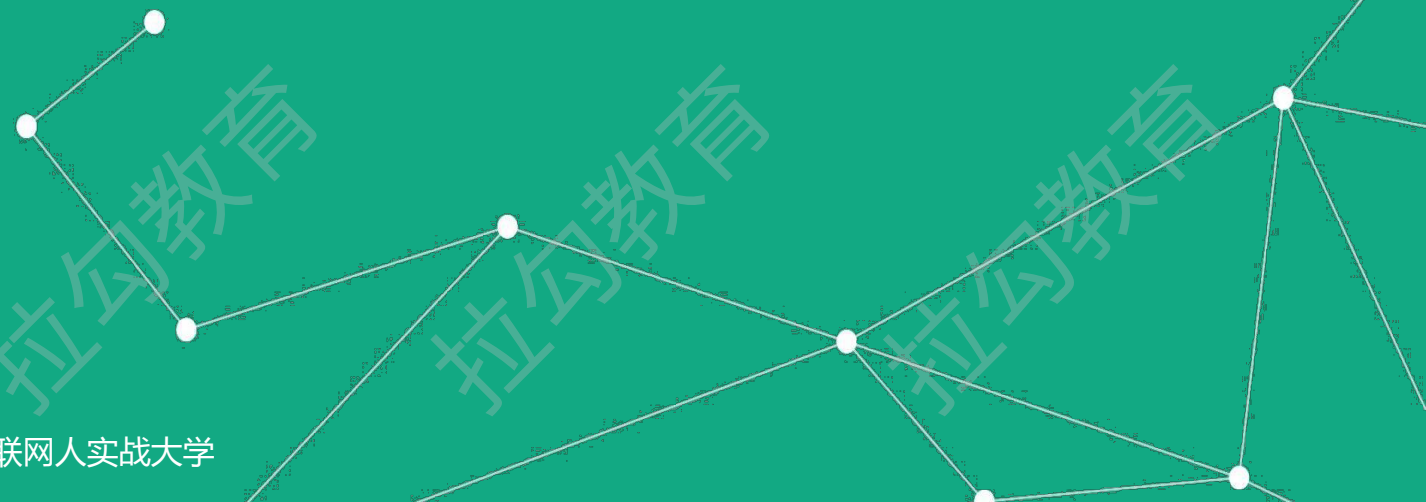
课时 10

树在Amazon云服务中的应用

1. 什么是SQL
2. 为什么要优化SQL的执行
3. SQL语法树是个什么树
4. 树的序列化和S-expression



什么是SQL

SQL（Structured Query Language）在硅谷科技公司一般读作 sequel

本来被称作 SEQUEL（Structured English Query Language）

但后来因为 SEQUEL 作为商标已经被注册了，只能改名为 SQL

什么是SQL

SQL 被用来管理和查询关系型数据库

比如：

```
SELECT name FROM table;
```

ID	name
1	Alex
2	David
3	Patrick

为什么要优化SQL的执行

- SQL 最常用的用途就是查询数据
- 关系型数据库在执行 SQL 语句时必须要考虑效率

需要多长时间完成这条 SQL 查询？

需要多少计算资源，包括 CPU、磁盘、电能去执行这条 SQL 语句？

为什么要优化SQL的执行

关系型数据库具体怎样执行 SQL 语句我们称之为执行计划（Execution Plan）

```
SELECT *
FROM customers c, orders o
WHERE c.id = o.cust_id AND c.creation_timestamp < 1579069516;
```

SQL 引擎在执行它的时候是有很多纠结的，纠结什么呢？

执行计划

为什么要优化SQL的执行

以下列举几个执行计划的例子：

- Creation_timestamp 小于这样一个过滤条件应该在 JOIN 语句前执行，还是在 JOIN 语句之后执行？
- 对于 JOIN 语句应该使用 merge join，还是 hash join，还是用建立的 index 使用 nested loop join 呢？
- 如果是 hash join 或者 nested loop join 的话，应该是先遍历 cutomers 表然后查找对应的 orders 还是先遍历 orders 表再查找对应的 customers 呢？
- 如果对于 creation_timestamp 建立有二级索引（Secondary Index）的话，是应该用二级索引查找 creation_timestamp 还是用主索引查找 ID 比较快？

为什么要优化SQL的执行

这些问题的答案都是相互依赖的，也都是需要交叉考虑的

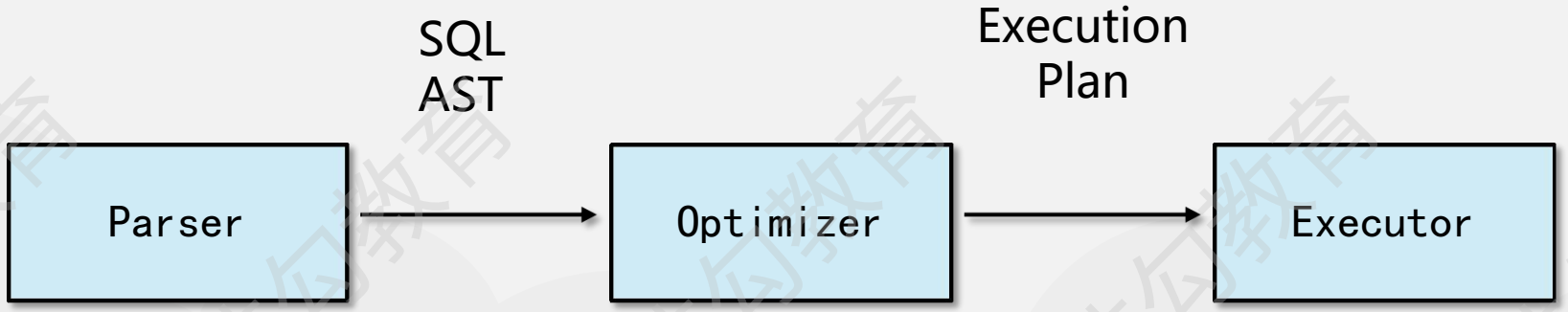
最优的执行计划还取决于：

1. 我们的表有多少行
2. 各种物理操作的相对速度
3. 不同数据的存储位置和数据值的分布等

为什么要优化SQL的执行

- 首先是 SQL 解析器（Parser），它负责把用户输入的 SQL 解析成 SQL 语法树（AST）
- 后面的 SQL 优化器（Optimizer）接受前面解析的原生语法树，对它进行优化重写语法树和执行计划

  一般优化器不仅仅会看语法树，还需要结合特定的用户数据库配置，数据实际分布进行优化
- 最后面的 SQL 执行器（Executor）才会去真正的执行优化重写的 SQL 语法树



SQL语法树是个什么树

AST（Abstract Syntax Tree）即语法树，在计算机科学中是用树来表达源代码的一种方式

```
SELECT name FROM table;
```

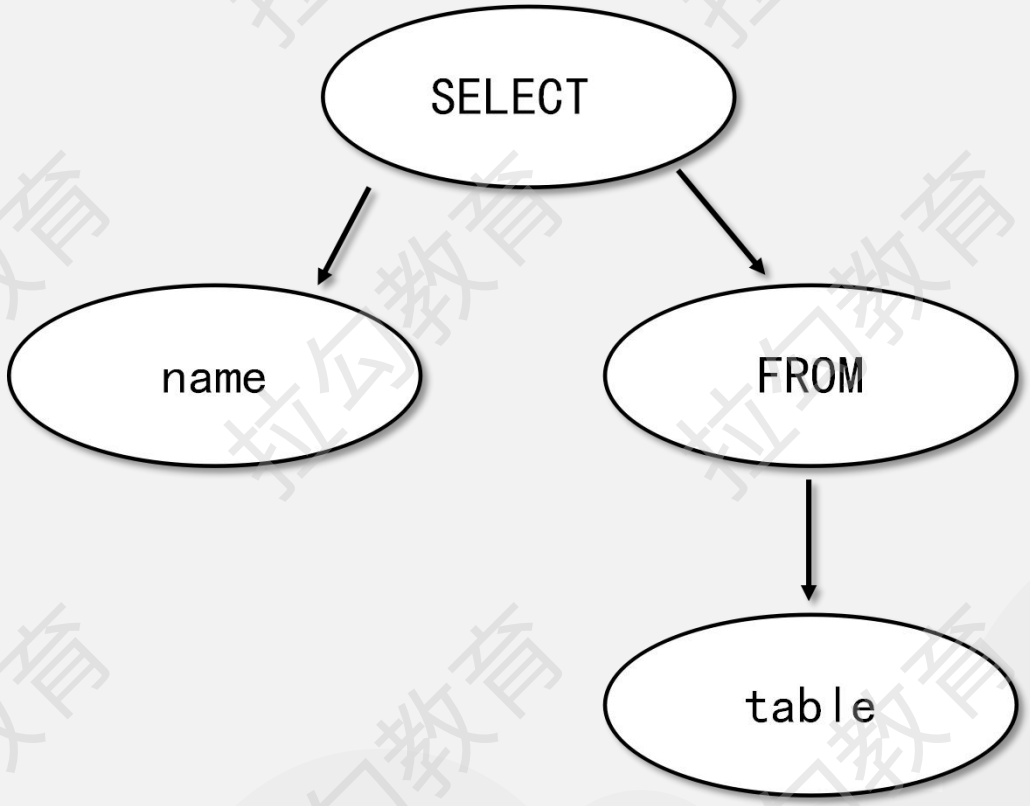
ID	name
1	Alex
2	David
3	Patrick

SQL语法树是个什么树

用 AST 表达 SQL 语句时

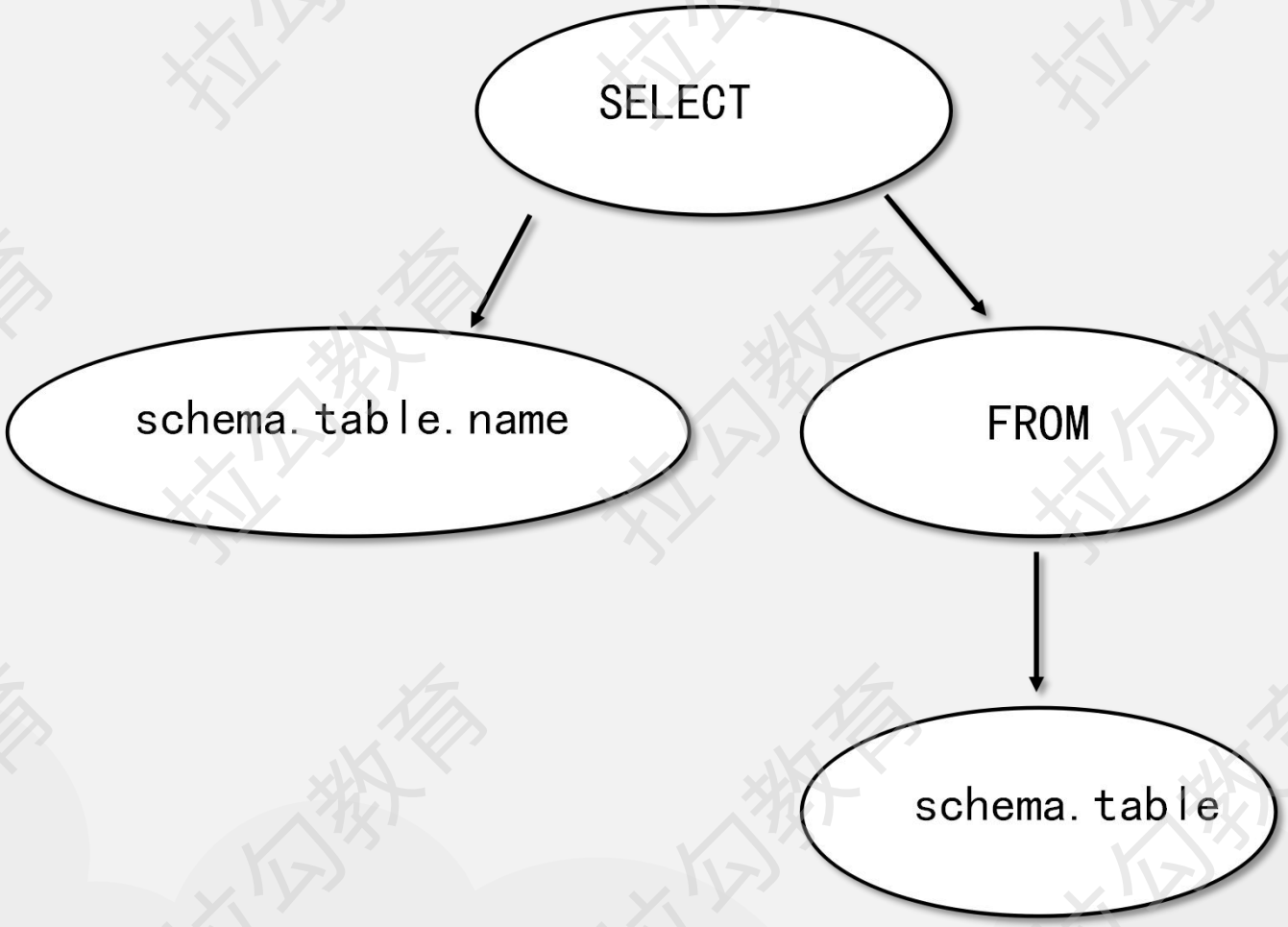
SQL 操作符永远是子树的根

而树的叶子则是比如这里的 name 或者 table



- SQL 语句的操作就是 SELECT 和 FROM
  他们都是子树的根节点
- from 和 name 在语义上是不同的含义
  from 是操作，而 name 是一张表中的列名称

SQL语法树是个什么树



树的序列化和S-expression

后续的优化器和执行器中我们是怎样传递这棵树的呢？

在 Amazon 分布式数据库中，往往是好几个不同的服务器组合而成，在不同的服务器 RPC 之间传递树，我们需要把树序列化成可以在网络中传输解析的格式

S-expression

树的序列化和S-expression

S-expression

有时候也简称为 sexp

最初是由 Lisp 语言发明并被人们广泛使用

常见的 S-expression 被定义为：

- 一个不可分的元素（atom）
- 或者是 “(x y)”，左括号，x，y，右括号的形式

  其中 x 和 y 都是 S-expression

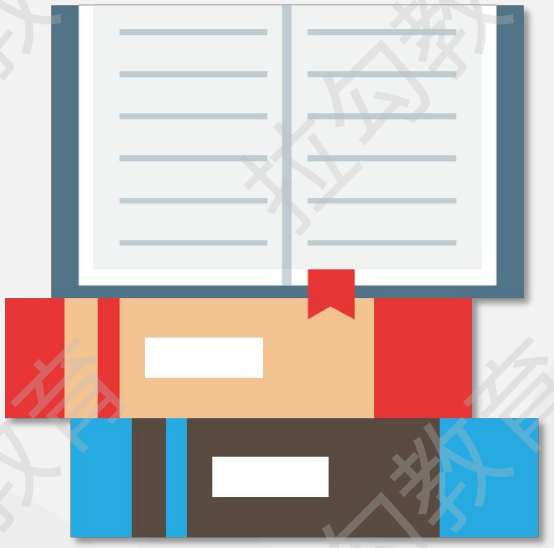
树的序列化和S-expression

```
(SELECT schema.table.name (FROM schema.table))
```

我们需要有一个嵌套的括号：

括号的第一个 token 被定义成这个括号表达的子树的根

每一个嵌套的括号就是一个子树



S-expression 的反序列化：把每一个括号展开成一个子树就可以

总结

1. 树在 Amazon 这样的超大型分布式数据库系统中如何应用
2. 为什么超大型分布式数据库需要对 SQL 引擎进行优化
3. 分析了怎么解析 SQL 语法树
4. 怎样用树表达一个 SQL 语句
5. 为什么树的表达能给我们处理 SQL 语句带来便利
6. S-expression 怎样序列化表达通用的树