



(12) 发明专利

(10) 授权公告号 CN 114003231 B

(45) 授权公告日 2022. 07. 26

(21) 申请号 202111146292.6

G06F 16/242 (2019.01)

(22) 申请日 2021.09.28

(56) 对比文件

(65) 同一申请的已公布的文献号

CN 113434533 A, 2021.09.24

申请公布号 CN 114003231 A

CN 113254581 A, 2021.08.13

(43) 申请公布日 2022.02.01

审查员 林方清

(73) 专利权人 厦门国际银行股份有限公司

地址 361000 福建省厦门市鹭江道8-10号

国际银行大厦1-6层

(72) 发明人 李龙权 郭晨皓 蒋天榕 胡涛

唐雁南

(74) 专利代理机构 厦门仕诚联合知识产权代理

事务所(普通合伙) 35227

专利代理师 蔡稷元

(51) Int. Cl.

G06F 8/41 (2018.01)

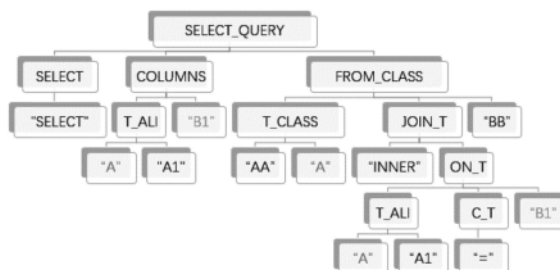
权利要求书2页 说明书6页 附图4页

(54) 发明名称

一种SQL语法解析树优化方法及系统

(57) 摘要

本发明公开了一种SQL语法解析树优化方法及系统,该系统包括:输入模块,用于输入单段SQL语句;语法解析器,用于解析单段SQL语句,得到AST语法解析树;提取模块,用于遍历AST语法解析树,将AST语法解析树与元数据库进行匹配,提取AST语法解析树中节点的信息;扩展模块,用于根据信息扩展AST语法解析树;形变模块,用于按照制定的约束条件,将AST语法解析树进行变形,得到形变后的AST语法解析树;保存模块,用于保存形变后的AST语法解析树。本发明补充完整SQL语法解析树,获得标准化SQL结构,可以获得原始SQL中无法获得的信息。



1. 一种SQL语法解析树优化系统,其特征在于,包括:
 - 输入模块,用于输入单段SQL语句;
 - 语法解析器,用于解析所述单段SQL语句,得到AST语法解析树;
 - 提取模块,用于遍历所述AST语法解析树,将所述AST语法解析树与元数据库进行匹配,提取所述AST语法解析树中节点的信息;
 - 扩展模块,用于根据所述信息扩展所述AST语法解析树;
 - 形变模块,用于按照制定的约束条件,将所述AST语法解析树进行形变,得到形变后的AST语法解析树;
 - 扩展和形变所述AST语法解析树的方法包括:
 - 针对节点属性为表的节点,扩充所述节点的第一子节点,所述第一子节点的节点属性为表中文名,值为该表对应中文名;
 - 针对节点属性为字段的节点,扩充所述节点的第二子节点,所述第二子节点的节点属性为字段中文名,值为该字段对应中文名;
 - 针对节点属性为字段的节点,扩充所述节点的第三子节点,所述第三子节点的节点属性为字段类型,值为该字段对应字段类型;
 - 针对节点属性为表的节点,如果没有节点属性为表别名的兄弟节点,扩展所述节点的第一兄弟节点,所述第一兄弟节点的节点属性为表别名,值为该表对应别名;
 - 针对节点属性为字段的节点,如果没有节点属性为字段别名的兄弟节点,扩展所述节点的第二兄弟节点,所述第二兄弟节点的节点属性为字段别名,值为该字段对应别名;
 - 针对节点属性为字段的节点,如果没有节点属性为归属表别名的兄弟节点,扩展所述节点的第三兄弟节点,所述第三兄弟节点的节点属性为归属表别名,值为该字段归属的表对应别名;
 - 保存模块,用于保存所述形变后的AST语法解析树。
2. 根据权利要求1所述的SQL语法解析树优化系统,其特征在于,所述输入模块还用于输入多段SQL语句或SQL脚本,以及将所述多段SQL语句或所述SQL脚本分割为多个所述单段SQL语句。
3. 根据权利要求2所述的SQL语法解析树优化系统,其特征在于,所述保存模块按照制定的XPath规则,将所述形变后的AST语法解析树按照XML格式进行封装并保存到数据库二维表中。
4. 根据权利要求3所述的SQL语法解析树优化系统,其特征在于,所述保存模块将所述单段SQL语句保存在所述数据库二维表中的一行的XML里,所述保存模块将所述多段SQL语句或所述SQL脚本根据分割成的多个所述单段SQL语句分开保存到多行的XML里。
5. 根据权利要求4所述的SQL语法解析树优化系统,其特征在于,还包括访问模块,用于使用所述XPath规则访问所述数据库二维表,获得业务信息。
6. 根据权利要求1所述的SQL语法解析树优化系统,其特征在于,进行树的后序遍历。
7. 根据权利要求1所述的SQL语法解析树优化系统,其特征在于,所述信息包括表中文名、表别名、字段中文名、字段类型以及表的归属关系。
8. 根据权利要求1所述的SQL语法解析树优化系统,其特征在于,当所述单段SQL语句包括至少一个层次的子查询时,将每个层次的节点属性为子查询的节点作为节点属性为表的

节点。

9. 一种SQL语法解析树优化方法,其特征在于,包括以下步骤:

输入单段SQL语句;

解析所述单段SQL语句,得到AST语法解析树;

遍历所述AST语法解析树,将所述AST语法解析树与元数据库进行匹配,提取所述AST语法解析树中节点的信息;

根据所述信息扩展所述AST语法解析树;

按照制定的约束条件,将所述AST语法解析树进行变形,得到形变后的AST语法解析树;

扩展和变形所述AST语法解析树的方法包括:针对节点属性为表的节点,扩充所述节点的第一子节点,所述第一子节点的节点属性为表中文名,值为该表对应中文名;针对节点属性为字段的节点,扩充所述节点的第二子节点,所述第二子节点的节点属性为字段中文名,值为该字段对应中文名;针对节点属性为字段的节点,扩充所述节点的第三子节点,所述第三子节点的节点属性为字段类型,值为该字段对应字段类型;针对节点属性为表的节点,如果没有节点属性为表别名的兄弟节点,扩展所述节点的第一兄弟节点,所述第一兄弟节点的节点属性为表别名,值为该表对应别名;针对节点属性为字段的节点,如果没有节点属性为字段别名的兄弟节点,扩展所述节点的第二兄弟节点,所述第二兄弟节点的节点属性为字段别名,值为该字段对应别名;针对节点属性为字段的节点,如果没有节点属性为归属表别名的兄弟节点,扩展所述节点的第三兄弟节点,所述第三兄弟节点的节点属性为归属表别名,值为该字段归属的表对应别名;

保存所述形变后的AST语法解析树。

一种SQL语法解析树优化方法及系统

技术领域

[0001] 本发明涉及大数据智能化技术领域,特别是一种SQL语法解析树优化方法及系统。

背景技术

[0002] SQL语言应用于各行各业,发挥着巨大的作用。相对于其他编程语言,SQL语言的语法结构非常简单,但是随着使用场景的复杂化,其最终表达形态也千变万化。SQL代码系统内部无法进行标识,其存储形式基本局限于文本,只能用于简单检索,无法进行结构化、继承使用、知识性学习以及向下传播。

[0003] 所有的数据库都会对提交的SQL代码进行解析,其目的是获取存储在数据库对应的数据。SQL代码本身也是一种源数据,其包含的知识点(业务口径、对应表和字段、限制条件、关联关系等)却无法有效得到提取跟继承应用。

[0004] 虽然目前市场上也有一些开源的SQL解析方案(采用ANTI等架构),可以实现SQL的解析并提取一些简单的信息(对应表和字段、限制条件),但是原始的SQL语法解析树存在不容易遍历使用,格式不统一,字段关系不明确等较多弊端,影响后续应用。

发明内容

[0005] 本发明为解决上述问题,提供了一种SQL语法解析树优化方法及系统,补充完整SQL语法解析树,获得标准化SQL结构,可以获得原始SQL中无法获得的信息。

[0006] 为实现上述目的,本发明采用的技术方案为:

[0007] 一种SQL语法解析树优化系统,包括:输入模块,用于输入单段SQL语句;语法解析器,用于解析所述单段SQL语句,得到AST语法解析树;提取模块,用于遍历所述AST语法解析树,将所述AST语法解析树与元数据库进行匹配,提取所述AST语法解析树中节点的信息;扩展模块,用于根据所述信息扩展所述AST语法解析树;形变模块,用于按照制定的约束条件,将所述AST语法解析树进行变形,得到形变后的AST语法解析树;保存模块,用于保存所述形变后的AST语法解析树。

[0008] 优选的,所述输入模块还用于输入多段SQL语句或SQL脚本,以及将所述多段SQL语句或所述SQL脚本分割为多个所述单段SQL语句。

[0009] 优选的,进行树的后序遍历。

[0010] 优选的,所述信息包括表中文名、表别名、字段中文名、字段类型以及表的归属关系。

[0011] 优选的,扩展和变形所述AST语法解析树的方法包括:针对节点属性为表的节点,扩充所述节点的第一子节点,所述第一子节点的节点属性为表中文名,值为该表对应中文名;针对节点属性为字段的节点,扩充所述节点的第二子节点,所述第二子节点的节点属性为字段中文名,值为该字段对应中文名;针对节点属性为字段的节点,扩充所述节点的第三子节点,所述第三子节点的节点属性为字段类型,值为该字段对应字段类型;针对节点属性为表的节点,如果没有节点属性为表别名的兄弟节点,扩展所述节点的第一兄弟节点,所述

第一兄弟节点的节点属性为表别名,值为该表对应别名;针对节点属性为字段的节点,如果没有节点属性为字段别名的兄弟节点,扩展所述节点的第二兄弟节点,所述第二兄弟节点的节点属性为字段别名,值为该字段对应别名;针对节点属性为字段的节点,如果没有节点属性为归属表别名的兄弟节点,扩展所述节点的第三兄弟节点,所述第三兄弟节点的节点属性为归属表别名,值为该字段归属的表对应别名。

[0012] 优选的,当所述单段SQL语句包括至少一个层次的子查询时,将每个层次的节点属性为子查询的节点作为节点属性为表的节点。

[0013] 优选的,所述保存模块按照制定的XPATH规则,将所述形变后的AST语法解析树按照XML格式进行封装并保存到数据库二维表中。

[0014] 优选的,所述保存模块将所述单段SQL语句保存在所述数据库二维表中的一行的XML里,所述保存模块将所述多段SQL语句或所述SQL脚本根据分割成的多个所述单段SQL语句分开保存到多行的XML里。

[0015] 优选的,所述系统还包括访问模块,用于使用所述XPATH规则访问所述数据库二维表,获得业务信息。

[0016] 基于同样的发明构思,本发明还提供了一种SQL语法解析树优化方法,包括以下步骤:输入单段SQL语句;解析所述单段SQL语句,得到AST语法解析树;遍历所述AST语法解析树,将所述AST语法解析树与元数据库进行匹配,提取所述AST语法解析树每个层次中节点属性为库、表、字段的节点的信息;根据所述信息扩展所述AST语法解析树;按照制定的约束条件,将所述AST语法解析树进行变形,得到形变后的AST语法解析树;保存所述形变后的AST语法解析树。

[0017] 本发明的有益效果是:

[0018] 1. 补充完整SQL语法解析树,获得标准化SQL结构,可以获得原始SQL中无法获得的信息;

[0019] 2. 将子查询建立的临时表视为数据库表扩展AST语法解析树,避免有效的业务信息在向上传递的过程中丢失;

[0020] 3. 使用XPATH规则存储形变后的AST语法解析树,可以快速访问XML获得需要的业务信息,方便后续应用。

附图说明

[0021] 图1为一种原始AST语法解析树的示例图;

[0022] 图2为一种形变后的AST语法解析树的示例图;

[0023] 图3为一种用于存储形变后的AST语法解析树的XML格式的大字段的示意图;

[0024] 图4为一种SQL语法解析树优化方法流程图。

具体实施方式

[0025] 为了使本发明所要解决的技术问题、技术方案及有益效果更加清楚、明白,以下结合具体实施例对本发明进行进一步详细说明。应当理解,此处所描述的具体实施例仅用以解释本发明,并不用于限定本发明。

[0026] SQL是一门ANSI的标准计算机结构化查询语言,用来访问和操作各类数据库系统。

SQL代码包含了丰富的业务信息,但是由于SQL代码的复杂性,SQL语言需要经过解析才能够让计算机进行识别,解析结果跟数据库高度绑定,在过去无法对这些业务信息进行有效的提取和使用。目前一些开源框架比如ANTLR4进行SQL解析的结果是一棵AST抽象语法树,虽然可以获取部分业务信息,但是由于是脱离数据库,存在信息丢失、使用不便等较大缺陷。

[0027] 实施例一

[0028] 本实施例公开了一种基于ANTLR4解析SQL后的AST抽象语法树优化系统,对SQL语句或SQL脚本进行解析,并扩展可能用到的各类信息到结构体中,然后根据规范对结构体进行变形,采用XML格式保存到数据库中,以方便后续查询和应用。

[0029] 本实施例所述的系统包括:

[0030] 输入模块,用于输入单段SQL语句、多段SQL语句或SQL脚本,以及将多段SQL语句或SQL脚本分割为多个单段SQL语句。

[0031] 由于ANTLR4解析器识别的是单段的SQL语句,如果输入的是多段的SQL语句或者完整的SQL脚本(包含多段SQL语句和一些控制性语法),需要使用标准分割符“;”对输入进行分割,识别其中每一个单段的SQL语句,并按SQL序号分别传递给解析器。

[0032] ANTLR4解析器,用于解析单段SQL语句,得到AST语法解析树,参照附图1。

[0033] 提取模块,用于遍历AST语法解析树,将AST语法解析树与数据库的元数据库进行匹配,提取元数据库中对应AST树中节点的信息。即结合数据库对AST树的各个根/叶子节点进行量化,该信息具体为每个层次中节点属性为库、表、字段的节点的值。在数据库中,一个字段仅能属于特定的一张表,一个表仅能属于特定的一个库,但是根据AST树中无法获得这个归属信息。在本实施例中,通过元数据库的匹配,就可以获得完整的SQL中每个层次使用的库、表、字段信息,例如用于下一步骤扩展AST树中进行补充的表中文名、字段中文名、字段类型等。

[0034] 后序遍历(LRD)是二叉树遍历的一种,在二叉树中,先左后右再根,即首先遍历左子树,然后遍历右子树,最后访问根结点。在本实施例中,针对量化后的解析树,使用AC算法进行树的后序遍历,可以更快地完成遍历,而且获得的数据层次清晰。

[0035] 扩展模块,用于根据提取的信息扩展AST语法解析树。

[0036] 扩展AST树的方法,以及下一步骤变形AST树的方法包括:

[0037] 针对节点属性为表的节点,扩充节点的第一子节点,第一子节点的节点属性为表中文名,值为该表对应中文名。

[0038] 针对节点属性为字段的节点,扩充节点的第二子节点,第二子节点的节点属性为字段中文名,值为该字段对应中文名。

[0039] 针对节点属性为字段的节点,扩充节点的第三子节点,第三子节点的节点属性为字段类型,值为该字段对应字段类型。

[0040] 针对节点属性为表的节点,如果没有节点属性为表别名的兄弟节点,扩展节点的第一兄弟节点,第一兄弟节点的节点属性为表别名,值为该表对应别名。如果已有这个兄弟节点,不额外扩展,直接复用。

[0041] 针对节点属性为字段的节点,如果没有节点属性为字段别名的兄弟节点,扩展节点的第二兄弟节点,第二兄弟节点的节点属性为字段别名,值为该字段对应别名。如果已有这个兄弟节点,不额外扩展,直接复用。

[0042] 针对节点属性为字段的节点,如果没有节点属性为归属表别名的兄弟节点,扩展节点的第三兄弟节点,第三兄弟节点的节点属性为归属表别名,值为该字段归属的表对应别名。如果已有这个兄弟节点,不额外扩展,直接复用。

[0043] 在本实施例中,到底是扩展兄弟节点还是子节点主要看需要。子节点用于补充属性,比如字段中文名、字段类型、表中文名,这些信息不属于AST树的整体框架。兄弟节点主要是保证AST树的结构一致性。

[0044] 形变模块,用于按照制定的约束条件,将AST语法解析树进行变形,得到形变后的AST语法解析树,参照附图2。

[0045] 为了后续结构体的方便使用,制定语法规则以保证AST树能够在语法规则约束下进行形变。

[0046] 该语法规则包括:

[0047] 规则1:所有的表必须有表别名。

[0048] 规则2:所有的数据库源表必须有表中文名(除非其来源对应的数据库源表无中文注释,则扩充的第一子节点的值为空)。

[0049] 规则3:子查询作为一种特殊的表,必须有表别名,但是可以没有表中文名。

[0050] 规则4:所有的字段必须有字段别名。

[0051] 规则5:所有的字段必须有字段类型。

[0052] 规则6:所有的字段必须有字段中文名(除非其来源对应的数据库源字段无中文注释)。

[0053] 规则7:所有的字段必须有归属的表别名,该表别名跟规则1保持一致。

[0054] 规则8:子查询作为一种特殊的表,满足规则4、5、6、7。

[0055] 规则9:所有类型为数据库、表、字段的值,设置为小写。

[0056] 规则10:所有类型为关键字的值,设置为大写。

[0057] 在本实施例中,扩展模块实现所有树叶节点信息补充,形变模块实现整个树结构按照设定的完整形态进行扩充。

[0058] 通过扩展模块,我们得到底层SQL扩展后的AST抽象语法树。由于SQL代码经常包含子查询,子查询是根据底层SQL形成的一个临时表,但是其在元数据库中并没有有效登记,无法获得表名、字段名等信息,会导致有效的业务信息在向上传递的过程中丢失。

[0059] 为了解决这个问题,当单段SQL语句包括至少一个层次的子查询时,将每个层次的节点属性为子查询的节点作为节点属性为表的节点。

[0060] 扩展和变形AST语法解析树的方法还包括以下步骤:

[0061] S1. 针对第一层次子查询,虚拟建立一张临时表,该临时表集成子查询的所有字段、字段中文名(步骤S2扩展得到)和字段类型(步骤S2扩展得到)。

[0062] S2. 将步骤S1建立的临时表视为数据库表,对AST树进行扩展/变形。

[0063] S3. 针对往上层次的所有子查询,递归执行步骤S1和S2,最终会获得一个经过扩充和形变后,符合语法规则的AST树。

[0064] 保存模块,用于保存形变后的AST语法解析树。

[0065] 将单段SQL语句变形为形变后的AST语法解析树的操作流程需要进行多次遍历,已达到不同的目的,包括:遍历第一次寻找需要与元数据库进行匹配的信息,第二次进行信息

扩展,第三次进行AST树的变形,第四次进行保存。这个就是语法解析树存在的技术缺陷,也是我们为何使用XPATH语法保存的原因。

[0066] 由于SQL解析树存在于内存中(JAVA类),无法有效存储,遍历比较困难,属性提取不直观,并不适宜后续应用。在本实施例中,根据所有SQL结构,建立一个用于存储SQL解析树的结构(SQL知识库结构),该结构具备可二维化存储、有灵活的遍历方式、可重新还原成SQL、可快速提取相关信息等能力,后续方便使用,并设计一个可以使用XPATH语法进行遍历的大字段,参照附图3,用于存储该结构。

[0067] 在本实施例中,保存模块按照制定的XPATH规则,将形变后的AST树按照XML格式进行封装并保存到数据库二维表中。

[0068] 该规则跟形变后的AST树严格对应,主要属性类型包括:

[0069] root根节点、ddlStatementddl语句、dmlStatementdml语句、SimpleSelect简单查询语句、QuerySpecification查询描述、SelectElements查询元素列表、FullColumnName字段、FullColumnCnName字段中文名、FullColumnType字段类型、FromClause from语句块、TableSources库名、TableSourceBase表来源、AtomTableItem原子表、TableName表名、TableCnName表中中文名、FullId完整Id、Uid唯一Id、SimpleId简单Id、OuterJoin外连接、TNode终结符、SubqueryTableItem子查询表、ComparisonOperator比较运算符、PredicateExpression判别表达式、BinaryComparasionPredicate二元比较判别、ExpressionAtomPredicate原子表达式判别、ConstantExpressionAtom常量表达式判别、Constant常量、DecimalLiteral数字等。

[0070] 保存模块将单段SQL语句保存在数据库二维表中的一行的XML里。

[0071] 保存模块将多段SQL语句或SQL脚本根据SQL序号,将分割成的多个单段SQL语句分开保存到多行的XML里。

[0072] 访问模块,用于使用XPATH规则访问数据库二维表,获得业务信息。

[0073] 访问模块可以使用XPATH语法根据需要访问XML,获得业务信息,进行后续包括SQL信息提取、SQL自动注释、SQL格式规范化、SQL智能推荐、SQL相似性判定等应用。举例如下:

[0074] 1. 获取SQL语句或SQL脚本中使用到的表(调度血缘关系):提取所有的TableSourceBaseTableName节点并且去重(原始AST树可以获得)。

[0075] 2. 获取SQL语句或SQL脚本中使用到的字段(影响性分析):提取所有的TableSourceBase.TableName.FullColumnName节点并且去重(原始AST树无法获得,必须经过扩充后才能够判定)。

[0076] 3. 提取SQL语句或SQL脚本中使用到的关联关系(主外键关系):提取ComparisonOperator值为=,并且等式左右两边的TableSourceBase.TableName.FullColumnName归属于不同的表(原始AST树无法获得,必须经过扩充后才能够判定)。

[0077] 4. 提取表常用限制条件:提取ComparisonOperator值为=,并且等式左右分别是TableSourceBase.TableName.FullColumnName跟Constant(原始AST语法解析树无法获得,必须经过扩充后才能够判定)。

[0078] 本实施例所述的系统补充完整SQL语法解析树,获得标准化SQL结构,从而可以获得原生SQL中无法获得的信息,包括但不限于表中文名、字段中文名、字段归属的表(可以是嵌套结构)、表别名(可以对隐性表进行扩充)。

[0079] 本实施例所述的系统可以实现所有的SQL代码(口径)的二维化存储,并且存储的SQL代码是经过扩充优化过,可以实现快速查找、关联查找、信息提取、SQL还原等多种功能。

[0080] 本实施例所述的系统可以整合所有的脚本、个人写的SQL在统一平台上,支撑扩展应用和后续业务口径的梳理。

[0081] 实施例二

[0082] 参照附图4,本实施例公开了一种基于ANTLR4解析SQL后的AST抽象语法树优化方法,包括以下步骤:

[0083] S1.输入SQL语句或者SQL脚本。

[0084] 其中,如果输入的是多段SQL语句或者完整的SQL脚本,需要根据标准分割符“;”将输入分割为多个单段SQL语句,并对单段SQL语句进行标序号,以便依次进行解析。

[0085] S2.解析单段SQL语句,得到AST语法解析树。

[0086] S3.遍历AST语法解析树,将AST语法解析树与元数据库进行匹配,提取AST语法解析树中节点的信息。

[0087] 后序遍历AST树,与数据库的元数据库进行匹配,利用语法正确性原则,基于确定的归属情况下,我们就可以获得整个AST树里面的信息,包括表中文名、字段中文名、字段类型,同时也可以标注出表别名信息,用于后面的步骤。

[0088] S4.根据信息扩展AST语法解析树。

[0089] 扩展步骤就是对AST树最底层节点进行信息补充,具体为根据需要扩展子节点或兄弟节点。其中,新增兄弟节点可能会导致原节点的下移,带一个父节点用于牵头。

[0090] S5.按照制定的约束条件,将AST语法解析树进行变形,得到形变后的AST语法解析树。

[0091] 变形就是根据预定的AST树的完整结构,新增缺失节点。外层的结构需要在里层的结构进行信息扩充后才能进行信息扩充。

[0092] 其中,如果SQL结构中包含包括至少一个层次的子查询时,将每个层次的节点属性为子查询的节点作为节点属性为表的节点,进行扩展和变形步骤。

[0093] S6.保存形变后的AST语法解析树。

[0094] 制定与形变后的AST树严格对应的XPath规则,按照该规则,将形变后的AST树按照XML格式进行保存。

[0095] S7.使用XPath语法根据需要访问XML获得业务信息。

[0096] 上述说明示出并描述了本发明的优选实施例,应当理解本发明并非局限于本文所披露的形式,不应看作是对其他实施例的排除,而可用于各种其他组合、修改和环境,并能够在本文发明构想范围内,通过上述教导或相关领域的技术或知识进行改动。而本领域人员所进行的改动和变化不脱离本发明的精神和范围,则都应在本发明所附权利要求的保护范围内。

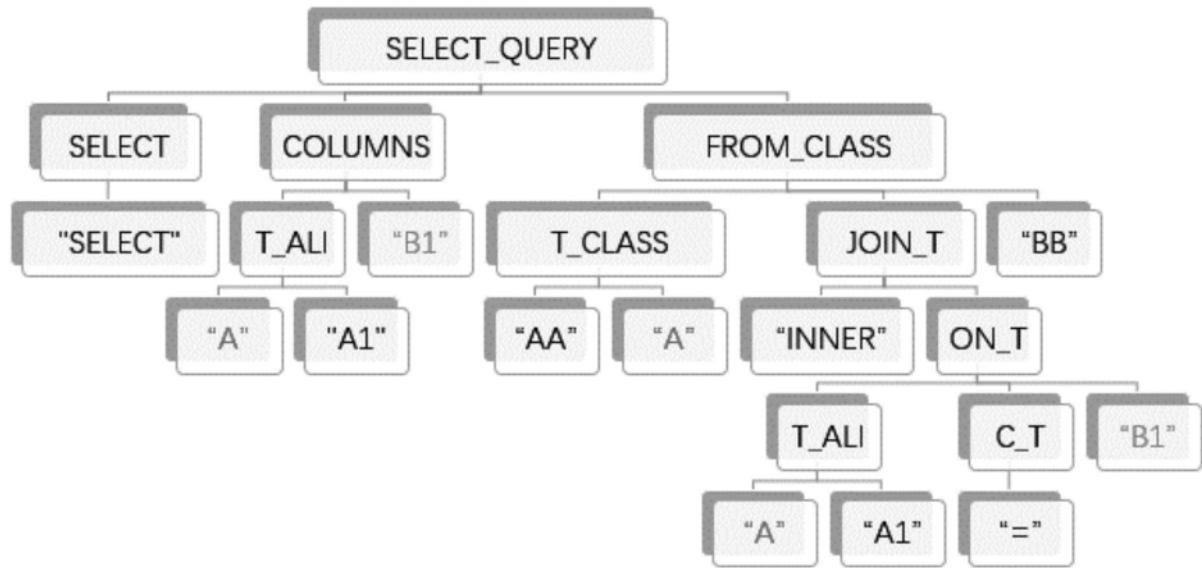


图1

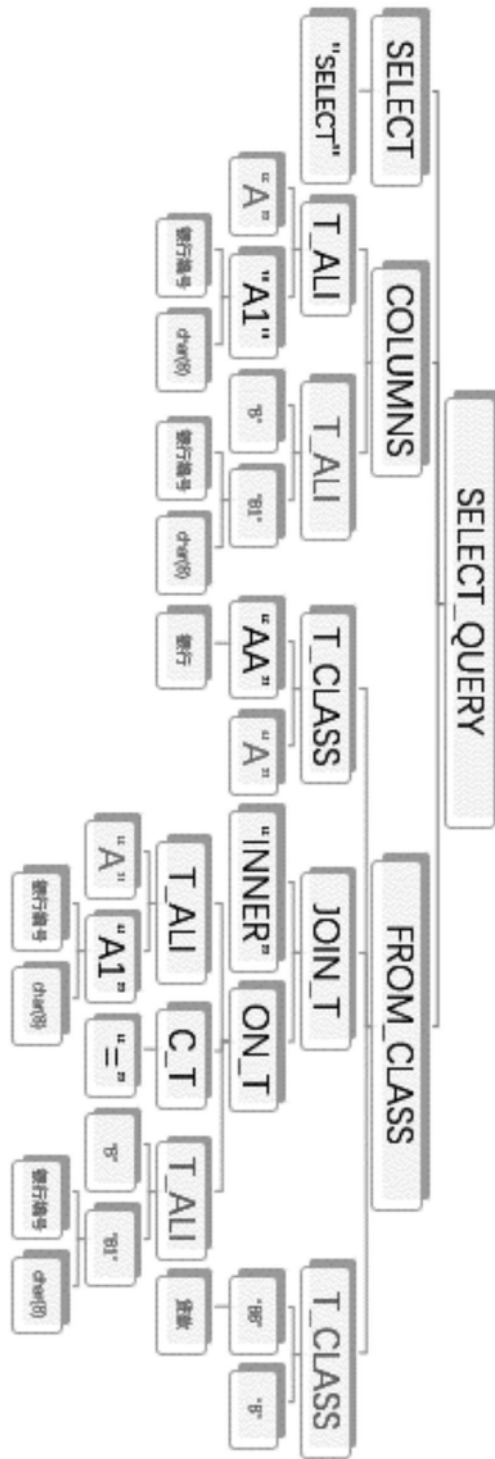


图2

```
<?xml version="1.0" encoding="UTF-8"?>
<SqlStatement>
  <DmlStatement>
    <SimpleSelect>
      <QuerySpecification>
        <TNode>SELECT</TNode>
        <SelectElements>
          <SelectColumnElement>
            <FullColumnName>
              <Uid>
                <SimpleId>
                  <TNode>NA</TNode>
                </SimpleId>
              </Uid>
              <DottedId>
                <TNode>.INTERNAL_KEY</TNode>
              </DottedId>
            </FullColumnName>
          </SelectColumnElement>
          <TNode>,</TNode>
          <SelectColumnElement>
            <FullColumnName>
              <Uid>
                <SimpleId>
                  <TNode>C</TNode>
                </SimpleId>
              </Uid>
              <DottedId>
                <TNode>.AIS1</TNode>
              </DottedId>
            </FullColumnName>
          </SelectColumnElement>
        </SelectElements>
        <FromClause>
          <TNode>FROM</TNode>
          <TableSources>
```

图3

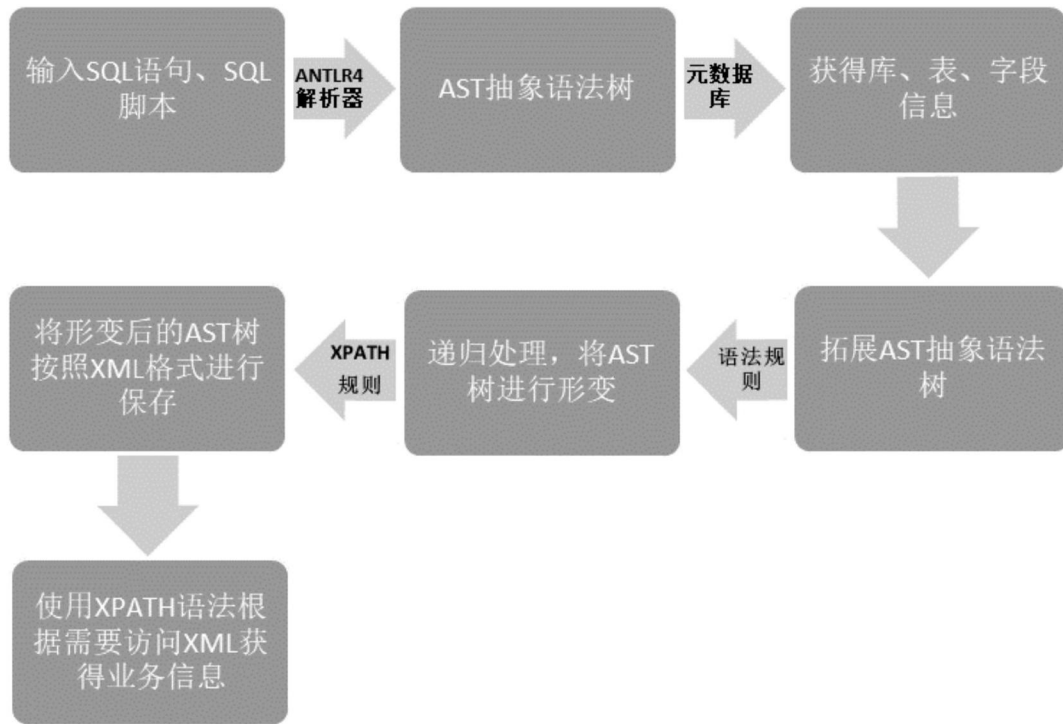


图4