SQL(Structured Query Language)：结构化查询语言， 它是关系型数据库的最重要的操作语言 。除数据库查询外，它具有其他功能，如可以定义数据结构、修改数据库中的数据，以及删除数据等。

**语句的分类：**

SQL命令一般分为DQL、 DML、 DDL、DCL和TCL。

DQL(Data Query Language)： 数据查询语句，主要用于数据查询，对应的命令是 SELECT。

DML(Data Manipulation Language)： 数据操纵语言， 主要用于插入、 更新、 删除数据， 分别对应的命令为INSERT、 UPDATE、 DELETE。

DDL(Data Definition Language)： 数据定义语言， 主要用于创建、 删除、修改表或索引等数据库对象，对应的命令为 CREATE、DROP、ALTER。

DCL：数据库控制语言，主要用于授予或撤销用户权限，对应命令为 GRANT、REVOKE。

TCL：事务控制语言，主要用于数据库的事务管理，对应命令为 BEGIN、SAVEPOINT、ROLLBACK、COMMIT。

**词法结构：**

一条SQL可以由多条SQL命令组成。 多条SQL命令之间由分号（;）分隔。

SQL命令由一系列记号组成， 这些记号可以由关键字、 标识符、 常量、 特殊的字符等组成。

**关键字**：SQL语言中具有特定意义的词。

**标识符**：标识表、列或者其他数据库对象的名字，随后可以通过标识符引用对象。 标识符必须以一个字母（a-z）或一个下划线（\_）开始。后续字符可以是字母，下划线()，数字(0-9)，或美元符号($)。

注意：根据SQL标准的字母规定，美元符号是不允许出现在标识符中的，因此他们的使用可能会降低应用的可移植性。

**常量**：字符串、位字符串和数字。

**特殊字符**：有一些特殊含义的操作符。

**美元符（$）**：跟在$后面的数字通常被用来表示在一个函数定义或一个预备语句中的位置参数。在其他上下文中该美元符号可以作为一个标识符或者一个美元引用字符串常量的一部分。

**圆括号（()）** ：用来分组表达式并且强制优先。在某些情况中，圆括号被要求作为一个特定 SQL 命令的固定语法的一部分。

**方括号（[]）**：用来选择一个数组中的元素。

**逗号（,）**：用在某些语法结构中来分割一个列表的元素。

**分号（;）**：结束一条 SQL 命令。它不能出现在一个命令中间的任何位置，除了在一个字符串常量中或者一个被引用的标识符中。

**冒号（:）**被用来从数组中选择“切片” 。在某些 SQL 的“方言”（例如嵌入式 SQL）中，冒号被用来作为变量名的前缀。

**星号（\*）**：被用在某些上下文中标记一个表的所有域或者组合值。当它被用作一个聚集函数的参数时，它还有一种特殊的含义，即该聚集不要求任何显式参数。

**句点（.）**被用在数字常量中，并且被用来分割模式、表和列名。

SQL命令中可以有注释， 这些注释在PostgreSQL中等同于空白。

--

/\* \*/

示例：

insert into t1 values(1,'aaa');

delete from t1 where id=1;

insert into t1 values(2,'bbb');

select \* from t1;

该SQL由3条命令组成。 在SQL中， 多行命令也可以写在一行中， 也可以写在多行中， 单条命令也可以占用多行。

**操作符优先级**

大部分操作符具有相同的优先并且是左结合的。操作符的优先级和结合性被硬写在解析器中。 如果您希望以不同于优先级规则所暗示的方式解析具有多个运算符的表达式，请添加括号。

|  |  |  |
| --- | --- | --- |
| 操作符/元素 | 结合性 | 描述 |
| . | 左 | 表/列名分隔符 |
| :: | 左 | PostgreSQL-风格的类型转换 |
| [ ] | 左 | 数组元素选择 |
| + - | 右 | 一元加、一元减 |
| ^ | 左 | 指数 |
| \* / % | 左 | 乘、除、模 |
| + - | 左 | 加、减 |
| （任意其他操作符） | 左 | 所有其他本地以及用户定义的操作符 |
| BETWEEN IN LIKE ILIKE SIMILAR |  | 范围包含、集合成员关系、字符串匹配 |
| < > = <= >= <> |  | 比较操作符 |
| IS NULL IS NOT NULL |  | IS TRUE、IS FALSE、IS NULL、IS DISTINCT FROM等 |
| NOT | 右 | 逻辑否定 |
| AND | 左 | 逻辑合取 |
| OR | 左 | 逻辑析取 |

**常见的relation**

Tablespace

Database

Schema

Table

Index

View

Constraint

Trigger

Event Trigger

Function

Procedure

**查看帮助**

\?

\h

pg\_ctl --help

**数据类型的分类**

|  |  |
| --- | --- |
| 分类名称 | 说明 |
| 数值类型 | 整数类型有2字节的smallint、4字节的int、8字节的bigint，也可以用int2、int4、int8来表示；精确类型的小数有numeric和decimal，numeric(m,n)也可以用decimal(m,n)表示；非精确类型的浮点小数有单精度的real和双精度的double precision；还有8字节的货币类型(money)；还有2字节的自增整数smallserial、4字节的自增整数serial、8字节的自增整数bigserial |
| 字符类型 | 有varchar(n)、char(n)、text 3种类型，varchar(n)也可以用char varying(n)来表示，varchar(n)最大可以存储1GB |
| 日期和时间类型 | 有 date、time、timestamp，而time和timestamp又根据是否包含时区分为两种类型，可以精确到秒以下，如毫秒 |
| 布尔类型 | PostgreSQL支持SQL标准的boolean数据类型，占用一个字节存储空间，表示 真/假 |
| 二进制数据类型 | 只有一种 bytea 类型 |
| 位串类型 | 有 bit 和 bit varying |
| 枚举类型 | 枚举类型是一个包含一系列有序静态值的集合的数据类型，相当于某些编程语言中的enum类型。使用枚举类型前需要先使用create type语句来创建该类型。 |
| 网络地址类型 | 有cidr、inet、macaddr 3种类型 |
| 数组类型 | 可以存储一个数组 |
| Range类型 | 范围类型，可以存储范围数据 |
| JSON/JSONB类型 | 可以存储JSON数据的类型 |
| 几何类型 | 包括点（point）、直线（line）、线段（lseg）、路径（path）、多边形（polygon）、圆（cycle）等类型 |
| XML类型 | 可以存储XML数据的类型 |
| 复合类型 | 可以把已有的简单类型组合成用户自定义的类型，如C语言中的结构体一样 |
| 文本搜索类型 | 有文本搜索查询 tsquery 和 文本搜索文档 tsvector |
| 伪类型 | 伪类型不能作为字段的数据类型，但是它可以用于声明函数的参数或者结果的类型。有any、anyarray、anyelement、cstring、internal、language\_handler、record、trigger、void、opaque等10种类型 |
| 其他类型 | 一些不易分类的类型都放在这里，如UUID类型，pg\_lsn类型，oid类型等 |

**数据类型的转换：**

select date '2000-1-1';

select CAST('2000-1-1' as date);

select '2000-1-1'::date;