链滴

postgre PREPARE 函数

作者：SummerWind

原文链接：https://ld246.com/article/1576755985874

来源网站：链滴

<h2 id="名称">名称</h2>
PREPARE -- 创建一个预备语句
<h2 id="语法">语法</h2>
PREPARE name [(datatype [, ...])] AS statement
/p>
<h2 id="描述">描述</h2>
PREPARE 创建一个预备语句。一个预备语句是服务器端的对象，可以用于优化性能。在执行 PR
PARE 语句的时候，指定的查询被分析、重写、规划。当随后发出 EXECUTE 语句的时候，预备语句
只需要执行了。因此，分析、重写、规划阶段都只执行一次，而不是每次都要执行一次。
预备语句可以接受参数：在它执行的时候替换到查询中的数值。可以在一个预备语句里按照位置
引用参数，比如 $1, $2 等。可以指定一个相应的参数数据类型列表。如果一个参数的数据类型没有
指定或声明为 unknown ，那么其类型将根据该参数所使用的实际上下文环境进行推测(如果有可能的
)。当执行该语句的时候，将在 EXECUTE 语句中为这些参数指定实际值。参见 <a href="https://ld2
6.com/forward?goto=https%3A%2F%2Fwww.yiibai.com%2Fmanual%2Fpostgresql%2Fsql-exe
ute.html" target="_blank" rel="nofollow ugc">EXECUTE 获取更多信息。</p

预备语句只是在当前数据库会话的过程中存在。如果客户端退出，那么预备语句就会被遗忘，因
必须在被重新使用之前重新创建。这也意味着一个预备语句不能被多个数据库客户端同时使用；但是
每个客户端可以创建它们自己的预备语句来使用。预备语句可以用 <a href="https://ld246.com/for
ard?goto=https%3A%2F%2Fwww.yiibai.com%2Fmanual%2Fpostgresql%2Fsql-deallocate.html
target="_blank" rel="nofollow ugc">DEALLOCATE 命令手工清除。
如果一个会话准备用于执行大量类似的查询，那么预备语句可以获得最大限度的性能优势。如果
询非常复杂，需要复杂的规划或者重写，那么性能差距将更加明显。比如，如果查询设计许多表的连
，或者有多种规则要求应用。如果查询的规划和重写相对简单，而执行起来开销相当大，那么预备语
的性能优势就不那么明显。
<h2 id="参数">参数</h2>
name
给予这个特定的预备语句任意名字。它必须在一个会话中是唯一的，并且用于执行或者删除一个
备语句。
datatype
预备语句的某个参数的数据类型。如果某个参数的数据类型未指定或指定为 unknown ，那么将
据该参数使用的上下文环境进行推断。可以使用 $1, $2 等等在预备语句内部引用这个参数。
statement
SELECT, INSERT, UPDATE, DELETE, VALUES 语句之一
<h2 id="注意">注意</h2>
在一些情况下，PostgreSQL 为一个预备语句生成的查询规划可能还不如按照普通方法提交并执
的查询生成的规划好。这是因为该查询在被规划的时候(也是优化器判断最优查询规划的时候)，在查
中声明的任何参数的实际数值都还不可见。PostgreSQL 在表中收集数据分布的统计，而且可以利用
询中的常量来猜测执行查询的可能结果。因为这些数据在规划的时候还是未知，所以，得到的规划可
很差劲。使用 <a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.yiibai.com%2
manual%2Fpostgresql%2Fsql-explain.html" target="_blank" rel="nofollow ugc">EXPLA
N 查看 PostgreSQL 为预备语句选取的查询计划。
有关查询规划和 PostgreSQL 为查询优化的目的收集统计的更多信息，参阅 <a href="https://ld
46.com/forward?goto=https%3A%2F%2Fwww.yiibai.com%2Fmanual%2Fpostgresql%2Fsql-an
lyze.html" target="_blank" rel="nofollow ugc">ANALYZE 文档。
可以通过查询 <a href="https://ld246.com/forward?goto=https%3A%2F%2Fwww.yiibai.co
%2Fmanual%2Fpostgresql%2Fview-pg-prepared-statements.html" target="_blank" rel="nofol
ow ugc">pg_prepared_statements 系统试图获得某个会话中所有可用的预备语句
<h2 id="例子">例子</h2>
为一个 INSERT 语句创建一个预备语句然后执行它：
PREPARE fooplan (int, text, bool, numeric) AS
INSERT INTO foo VALUES($1, $2, $3, $4);
EXECUTE fooplan(1, 'Hunter Valley', 't', 200.00);
为一个 SELECT 语句创建一个预备语句然后执行它：

PREPARE usrrptplan (int) AS
SELECT * FROM users u, logs l WHERE u.usrid=$1 AND u.usrid=l.usrid
AND l.date = $2;
EXECUTE usrrptplan(1, current_date);
注意，第二个参数的数据类型并未指定。所以将从上下文环境推测 $2 的类型。
<h2 id="兼容性">兼容性</h2>
SQL 标准包含一个 PREPARE 语句，但是它只用于嵌入式 SQL 。PostgreSQL 实现的 PREPARE 语句的语法也略有不同。
<h2 id="又见">又见</h2>
DEALLOCATE, EXECUTE
个人例子：
<pre><code class="highlight-chroma">create table a (
a_1 int4,
a_2 text,
a_3 boolean,
a_4 numeric
)

PREPARE aplan (in, text, bool, numeric) AS
    INSERT INTO aiw.a VALUES($1, $2, $3, $4);
EXECUTE aplan(1, Hunter Valley', 't', 200.00);

PREPARE aselect (nt) AS
    SELECT * FROM aigw.a WHERE a.a_1=$1;
EXECUTE aselect(1;
</code></pre>
可以在函数中使用如下语句 执行 sql
<pre><code class="highlight-chroma">执行转换存储过程
-- 拼接SQL
v_excute_sql := concat('select * from phis61.', v_rec_task.zx2phisprocname, '($1);');
select row_to_json v_rec_task) into v_jsonb_uploadtask;

-- 执行SQL
EXECUTE v_excute sql INTO v_rec USING v_jsonb_uploadtask;
</code></pre>
v_的变量为 Function declare 声明的变量