

实验十二 触发器

实验目的

通过实验使学生加深对数据完整性的理解，学会创建和使用触发器。

PostgreSQL触发器简明教程

PostgreSQL 触发器是一种特殊的函数，当指定的数据库事件发生时自动执行/调用。触发器可以在数据插入、更新或删除时自动运行一段指定的代码。

以下是创建 PostgreSQL 触发器的基本语法：

```
CREATE TRIGGER trigger_name [BEFORE|AFTER|INSTEAD OF] event_name ON table_name
FOR EACH ROW
EXECUTE FUNCTION function_name();
```

其中，`trigger_name` 是触发器的名称，`event_name` 是触发器要响应的事件，`table_name` 是触发器要绑定的表名，`function_name()` 是触发器要执行的函数。

以下是一些常见的 PostgreSQL 触发器事件：

- `INSERT`：在插入数据之前或之后触发。
- `UPDATE`：在更新数据之前或之后触发。
- `DELETE`：在删除数据之前或之后触发。

以下是一些常见的 PostgreSQL 触发器类型：

- `BEFORE`：在事件发生之前触发。
- `AFTER`：在事件发生之后触发。
- `INSTEAD OF`：代替事件发生。

以下是一个创建 PostgreSQL 触发器的例子：

```
CREATE TRIGGER example_trigger
AFTER INSERT ON example_table
FOR EACH ROW
EXECUTE FUNCTION example_function();
```

这个触发器会在 `example_table` 表中插入数据之后触发，并执行 `example_function()` 函数。

`INSTEAD OF` 触发器可以用于实现不可更新视图的增删改。当在一个视图上执行插入、更新或删除操作时，如果该视图不可更新，则会触发 `INSTEAD OF` 触发器，而不是执行默认的插入、更新或删除操作。

以下是一个 `INSTEAD OF` 触发器的例子，它可以用于实现不可更新视图的插入操作：

```

CREATE OR REPLACE FUNCTION insert_into_view() RETURNS TRIGGER AS $$
BEGIN
    INSERT INTO table_name (column1, column2, ...)
    VALUES (NEW.column1, NEW.column2, ...);
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER insert_into_view_trigger
INSTEAD OF INSERT ON view_name
FOR EACH ROW
EXECUTE FUNCTION insert_into_view();

```

这个触发器会在插入操作发生时触发，将插入的数据插入到表中。如果视图不可更新，则会触发这个触发器，而不是执行默认的插入操作。

类似地，可以创建 INSTEAD OF 触发器来实现不可更新视图的更新和删除操作。这些触发器的实现方式与上面的插入触发器类似，只需要将触发器的类型从 INSERT 改为 UPDATE 或 DELETE 即可。

触发器可以用于实现各种功能，例如：

- 检查约束。
- 数据审计。
- 数据同步。
- 数据加密。
- 数据备份。

课内实验

1. 为worker表（参照实验十一）建立触发器T1,当插入或是更新表中数据时，保证所操作的记录的sage 值大于0。（参考代码如下：）

```

CREATE OR REPLACE FUNCTION check_sage() RETURNS TRIGGER AS $$
BEGIN
    IF NEW.Sage <= 0 THEN
        RAISE EXCEPTION 'Sage must be greater than 0';
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER check_sage_trigger
BEFORE INSERT OR UPDATE ON worker
FOR EACH ROW
EXECUTE FUNCTION check_sage();

```

2. 为 worker 表建立触发器T2, 禁止删除编号为00001的CEO。

```

CREATE OR REPLACE FUNCTION forbid_delete_ceo() RETURNS TRIGGER AS $$
BEGIN
    IF OLD.Number = '00001' THEN
        RAISE EXCEPTION 'CEO cannot be deleted';
    END IF;
    RETURN OLD;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER forbid_delete_ceo_trigger
BEFORE DELETE ON worker
FOR EACH ROW
EXECUTE FUNCTION forbid_delete_ceo();

```

3. worker 表中的人员的编号是不可改变的，创建触发器 T3 实现更新中编号的不可改变性。
4. 演示违反 T1 触发器的约束的插入操作。
5. 演示违反 T1 触发器的约束的更新操作。
6. 演示违反 T2 触发器的约束的插入操作。
7. 演示违反 T2 触发器的约束的更新操作。
8. 演示 INSTEAD OF 触发器在不可更新视图上的运用。

自我实践

1. 建立一个在 worker 表上的触发器 T4, 要求插入记录的 sage 值必须比表中已记录的最大 sage 值大。
2. 建立一个在 worker 表上的触发器 T5, 要求当更新一个记录的时候，表中记录的 sage 值要比老记录的 sage 值大，因为一般工资级别只能升不能降。

实验报告提交要求

实验报告（课内实验+自我实践）至少包括以下内容：①实验目的；②实验环境；③实验内容、步骤、结果和实验过程中出现的问题；④实验过程中主要使用的SQL语句。

注意：本学期理论课和实验课作业（练习）迟交政策：截止日期后两周内补交分数打九折，期末考试前补交分数打八折。