CS307 Assignment 3

> Deadline: May 17th

Q1. Implement two triggers after `update` and `delete`

```sql
CREATE TABLE Post (
    PostID SERIAL PRIMARY KEY,
    Title VARCHAR(255) NOT NULL,
    Author VARCHAR(255) NOT NULL,
    Content TEXT NOT NULL
);
CREATE TABLE PostRevision (
    RevisionID SERIAL PRIMARY KEY,
    PostID INT NOT NULL,
    RevisionDate TIMESTAMP NOT NULL,
    OldVersionContent TEXT NOT NULL,
    NewVersionContent TEXT NOT NULL
);
CREATE TABLE DeleteLog (
    DeleteID SERIAL PRIMARY KEY,
    PostID INT NOT NULL,
    DeleteTime TIMESTAMP NOT NULL
);
```

给定以上三张数据表，请完成以下两个 PostgreSQL Trigger 的实现：

1. 当 Post 表中的 Content 列更新时，自动在 PostRevision 表中添加一行记录，记录更新的PostID，当前时间作为更新时间，以及更新后的内容。
2. 当 Post 表中的某一行被删除时，自动删除其在 PostRevision 表中的所有相关修订记录，并在 DeleteLog 表中添加一行记录，记录删除时间和被删除的PostID。

Given the three data tables above, please implement the following two PostgreSQL Triggers:

1. When the Content column in the Post table is updated, automatically add a row to the

`PostRevision` table, recording the updated `PostID`, the `current time` as the revision time, and the `updated content`.
2. When a row in the `Post` table is deleted, automatically delete all related revision records according to `PostID` in the `PostRevision` table and add a row to the `deleteLog` table, recording to `deletion time` and the `deleted PostID`.

Format

```
CREATE OR REPLACE FUNCTION add_post_revision() RETURNS TRIGGER AS $$
-- here is your procedure
$$ LANGUAGE plpgsql;

CREATE TRIGGER post_content_update
AFTER UPDATE ON Post
FOR EACH ROW
EXECUTE FUNCTION add_post_revision();

CREATE OR REPLACE FUNCTION delete_post() RETURNS TRIGGER AS $$
-- here is your procedure
$$ LANGUAGE plpgsql;

CREATE TRIGGER post_delete
AFTER DELETE ON Post
FOR EACH ROW
EXECUTE FUNCTION delete_post();
```

Q2. Implement a trigger before `insert`

税收是国家收入的重要来源，在中国的税收体系中每个纳税人都有唯一的识别号码TIN(tax-identification number)，我们在本题中将设计一个触发器，用于检查插入数据表中的的TIN是否合法，阻止不合法的数据插入数据库中，同时为插入的合法TIN填充类别信息。

Taxation is an important source of national income. In China's tax system, each taxpayer has a unique tax identification number (TIN). In this assignment, we will design a trigger to check the validity of the TINs inserted into a data table, prevent illegal data from being inserted into the database, and add category information for valid TINs.

首先请了解一下需要插入数据的表结构，以及一些作为参考的合法数据。

First, please familiarize yourself with the table structure for the data to be inserted, as well as some valid data provided as a reference.

```sql
create table if not exists tax_identification_number (
    tin_id serial not null
    constraint tin_pkey primary key,
    tin varchar(20) not null
    constraint tin_uq unique,
    nation varchar(10) not null,
    format varchar(100),
    type varchar(100) not null
);
```

	tin_id	tin	nation	format	type
1	4	130631190002140071	cn	cn id	person
2	5	C30631190002140071	cn	cn passport	person
3	6	W30631190002140071	cn	fg passport	person
4	7	H30631190002140071	cn	hk	person
5	8	M30631190002140071	cn	mo	person
6	9	T30631190002140071	cn	tw	person

我们需要处理的纳税对象分为以下六类: `中国身份证`, `中国护照`, `外国护照`, `香港纳税身份`, `澳门纳税身份`, `台湾纳税身份`。

We need to deal with six types of TINs: `cn id`, `cn passport`, `fg passport`, `hk` , `mo`, `tw`.

- `cn id`

  该信息是中国居民身份证真实格式的简化版。TIN必须是精确的18个字符，这18个字符应该都是数字（最后一位可以是**大写X**，表示"10"）。由于本次作业所有测试信息均为虚构，所以我们**只关心该id序列是否通过我们提供的校验规则**。

  This information is a simplified version of the real format of China's resident identity card. TIN must be exactly 18 characters long, and all 18 characters should be digits (the last character can be **uppercase X**, representing "10"). Since all test information for this homework is fictitious, we **only care if the ID sequence passes our provided validation rules**.

  > **Formula**: (Notes: i from 1)
  >
  > $S = \sum_{i=1}^{17} w_i \times d_i$
  >
  > $Checksum = (12 - (S \bmod 11)) \bmod 11$
  >
  > $w_i = 2^{18-i} \bmod 11$
  >
  > d_i is the i^{th} position of digit of ID number, from left to right.

  > **example**:
  >
  > ID: 130631190002140071
  >
  > w_i array: $7, 9, 10, 5, 8, 4, 2, 1, 6, 3, 7, 9, 10, 5, 8, 4, 2$
  >
  > d_i array: $1, 3, 0, 6, 3, 1, 1, 9, 0, 0, 0, 2, 1, 4, 0, 0, 7$
  >
  > $S = \sum_{i=1}^{17} w_i \times d_i = 165$
  >
  > $Checksum = (12 - (165 \bmod 11)) \bmod 11 = 1$

> $Checksum = 1$, which is the last digit of the ID, so this ID is valid

- `cn passport`

  TIN必须是精确的18个字符，以大写字母`C`开头，后面跟着任意17个数字。

  TIN must be exact 18 characters, starting with a capital letter `C` followed by any 17 digits.

  > **example:** C30631190002140071

- `fg passport`

  TIN必须是精确的18个字符，以大写字母`W`开头，后面跟着任意17个数字。

  TIN must be exact 18 characters, starting with a capital letter `W` followed by any 17 digits.

  > **example:** W30631190002140071

- `hk`

  TIN必须是精确的18个字符，以大写字母`H`开头，后面跟着任意17个数字。

  TIN must be exact 18 characters, starting with a capital letter `H` followed by any 17 digits.

  > **example:** H30631190002140071

- `mo`

  TIN必须是精确的18个字符，以大写字母`M`开头，后面跟着任意17个数字。

  TIN must be exact 18 characters, starting with a capital letter `M` followed by any 17 digits.

  > **example:** M30631190002140071

- `tw`

  TIN必须是精确的18个字符，以大写字母`T`开头，后面跟着任意17个数字。

  TIN must be exact 18 characters, starting with a capital letter `T` followed by any 17 digits.

  > **example:** T30631190002140071

Hint

你可以学习如何使用正则表达式来使你的代码更加简洁和优雅。正则表达式是一种广泛使用的技术，用于指定字符串模式匹配。

You can learn how to use Regular Expressions to make your codes more concise and decent. Regex is a widely-used technique to match string patterns.

Format

```
create or replace function tin_check()
returns trigger
as
$$
-- here is your procedure
$$ language plpgsql;
create trigger tin_trigger
    before insert
    on tax_identification_number
    for each row
    execute procedure tin_check();
```

Test

这里提供一些合法数据供参考，插入后表中数据应和上面提供的图例保持一致。

Here are some valid data for your reference. After insertion, the data in the table should match the example provided above.

```
insert into tax_identification_number (tin) values ('130631190002140071');
insert into tax_identification_number (tin) values ('C30631190002140071');
insert into tax_identification_number (tin) values ('W30631190002140071');
insert into tax_identification_number (tin) values ('H30631190002140071');
insert into tax_identification_number (tin) values ('M30631190002140071');
insert into tax_identification_number (tin) values ('T30631190002140071');
```