

```
ALTER TABLE account
ADD amount_in_account float;
```

```
create or replace function update_amount_function()
returns trigger
language plpgsql
as $$
declare
    f record;
Begin
    Savepoint first_savepoint;
```

```
for f in select account_number, amount_in_account from Account
loop
    If (f.account_number = new.sender_id) then
        if(f.amount_in_account >= new.amount) then
            update account
            set amount_in_account = amount_in_account - new.amount;
```

```
        update account
        set amount_in_account = amount_in_account + new.amount
        where account_number = new.receiver_id ;
        Return new;
    else
        Raise notice 'insufficient funds';
        Return null;
    END IF;
END IF;
END LOOP;
```

```
end;
$$;
```

```
Create trigger update_transaction_trigger
After insert
On transaction
For each row
Execute procedure update_amount_function() ;
```

```
Create table deposit
(
deposit_id serial not null,
account_number int not null,
amount float NULL,
Primary key (deposit_id),
FOREIGN KEY ( account_number) references Account (account_number)

);
```

```
Create or replace function deposit_trigger_function()
Returns trigger
Language plpgsql
As $$
Begin
update account
set amount_in_account = amount_in_account + new.amount
where account_number = new.account_number ;
Return new;
End;
$$;
```

```
Create trigger deposit_trigger
After insert
On deposit
For each row
Execute procedure deposit_trigger_function();
```

```
Create table withdrawal
(
withdrawal_id serial not null,
account_number int not null,
withdrawal_amount float NULL,
Primary key (withdrawal_id),
FOREIGN KEY ( account_number) references Account (account_number)

);
```

```
create or replace function withdrawl_trigger_function()
```

```
returns trigger  
language plpgsql  
as $$
```

```
declare
```

```
    f1 record;
```

```
Begin
```

```
  
for f1 in select account_number, amount_in_account from Account  
loop
```

```
    If (f1.account_number = new.account_number) then
```

```
        if(f1.amount_in_account >= new.withdrawal_amount) then
```

```
            update account
```

```
            set amount_in_account = amount_in_account - new.withdrawal_amount;
```

```
            Return new;
```

```
        else
```

```
            Raise notice 'insufficient funds';
```

```
            Return null;
```

```
        END IF;
```

```
    END IF;
```

```
END LOOP;
```

```
end;
```

```
$$;
```

```
  
Create trigger withdrawal_trigger
```

```
After insert
```

```
On withdrawal
```

```
For each row
```

```
Execute procedure withdrawal_trigger_function()
```

```
;
```