--1)  
--a  
create or replace function *inc*(val integer)  
returns integer as $$  
begin  
return val + 1;  
end; $$  
language plpgsql;  
  
select *inc*(2);  
--b  
create or replace function *mm*(a integer, b integer)  
returns integer as $$  
begin  
return a+b;  
end; $$  
language plpgsql;  
  
select *mm*(4,7);  
--c  
create or replace function *even*(a int)  
 returns boolean as  
$$  
begin  
 if a % 2 = 0 then return true;  
 else return false;  
 end if;  
end;  
$$  
language plpgsql;  
  
select *even*(4);  
select *even*(5);  
--d  
create or replace function *validity*(s varchar)  
 returns boolean as  
$$  
begin  
 if *LENGTH*(s) > 7 then return true;  
 else return false;  
 end if;  
end;  
$$  
language plpgsql;  
  
select *validity*('qwerty123');  
--e  
create or replace function *pow*(a numeric, out square numeric, out cube numeric) as  
$$  
begin  
 square = a \* a;  
 cube = square \* a;  
end;  
$$  
language plpgsql;  
  
select *pow*(2);  
  
--2)  
create table table1(  
  
);  
create table table2(  
  
);  
create table table3(  
  
);  
create table table4(  
  
);  
create table table5(  
  
);  
--a  
create function *current*()  
 returns trigger as  
$$  
begin  
 raise notice '%',*now*();  
 return new;  
end;  
$$  
language plpgsql;  
  
create trigger current\_t before insert on table1 for each row execute procedure *current*();  
--b  
create function *age*()  
 returns trigger as  
$$  
begin  
 raise notice '%', *age*(*now*(),new.t);  
 return new;  
end;  
$$  
language plpgsql;  
  
create trigger age\_t before insert on table2 for each row execute procedure *age*();  
--c  
create function *tax*()  
 returns trigger as  
$$  
begin  
 new.cost = new.cost \* 1.12;  
 return new;  
end;  
$$  
language plpgsql;  
  
create trigger tax\_t before insert on table3 for each row execute procedure *tax*();  
--d  
create function *stop*()  
 returns trigger as  
$$  
begin  
 raise exception 'Deletion is not allowed';  
end;  
$$  
language plpgsql;  
  
create trigger stop\_t before delete on table4 execute procedure *stop*();  
--e  
create function *call*()  
 returns trigger as  
$$  
begin  
 raise notice '%', *validity*(new.s);  
 raise notice '%', *pow*(new.a);  
 return new;  
end;  
$$  
language plpgsql;  
  
create trigger call\_t before insert on table5 for each row execute procedure *call*();  
  
--3)  
create table company(id int,  
name varchar,  
date\_of\_birth date,  
age int, salary numeric,  
workexperience int,  
discount numeric  
);  
--a  
create or replace function *increasing*(id int, name varchar, date\_of\_birth date, age int, inout salary numeric, workexperience int, out discount numeric)  
as $$  
declare  
 count int;  
begin  
 discount = 10;  
 count = workexperience/2;  
 for step in 1..count loop  
 salary = salary \* 1.1;  
 end loop;  
 count = workexperience/5;  
 for step in 1..count loop  
 discount = discount \* 1.01;  
 end loop;  
 insert into work values(id, name, date\_of\_birth, age, salary, workexperience, discount);  
end;  
$$  
language plpgsql;  
  
select \* from *increasing*(1, 'Ospan', '1988-03-06', 34, 1000, 6);  
--b  
create or replace function *inc2*(id int, name varchar, date\_of\_birth date, age int, inout salary numeric, workexperience int, out discount numeric)  
as $$  
declare  
 count int;  
begin  
 if age >= 40 then salary = salary \* 1.15;  
 end if;  
 discount = 10;  
 count = workexperience/2;  
 for step in 1..count loop  
 salary = salary \* 1.1;  
 end loop;  
 count = workexperience/5;  
 for step in 1..count loop  
 discount = discount \* 1.01;  
 end loop;  
 if workexperience > 8 then salary = salary \* 1.15;  
 end if;  
 if workexperience > 8 then discount = 20;  
 end if;  
 insert into work values(id, name, date\_of\_birth, age, salary, workexperience, discount);  
end;  
$$  
language plpgsql;  
  
select \* from *inc2*(2, 'Simon', '1976-03-06', 46, 10000,16);