1.  
 a  
 CREATE FUNCTION *increment*(a int)  
 RETURNS int AS $$  
 BEGIN  
 RETURN a+1;  
 END;  
 $$ LANGUAGE PLPGSQL;  
 SELECT *increment*(20);  
b  
 CREATE FUNCTION *sigma*(a int, b int)  
 RETURNS int AS $$  
 BEGIN  
 RETURN a+b;  
 END;  
 $$ LANGUAGE PLPGSQL;  
 SELECT *sigma*(20,30);  
c  
 CREATE FUNCTION *even*(a int)  
 RETURNS bool AS $$  
 BEGIN  
 RETURN a%2=0;  
 END;  
 $$ LANGUAGE PLPGSQL;  
 SELECT *even*(20);  
d  
 CREATE FUNCTION *pass*(s text)  
 RETURNS bool AS $$  
 BEGIN  
 RETURN (*char\_length*(s)>=8)and((*strpos*(s,'!')>0)or(*strpos*(s,'@')>0)or(*strpos*(s,'#')>0)or(*strpos*(s,'$')>0)or(*strpos*(s,'%')>0));  
 END;  
 $$ LANGUAGE PLPGSQL;  
 SELECT *pass*('kawabanga%');  
e  
 CREATE FUNCTION *L\_S*(s TEXT,  
 OUT len int, OUT symb boolean)  
 AS $$  
 BEGIN  
 len:=*char\_length*(s);  
 symb=(*strpos*(s,'!')>0)or(*strpos*(s,'@')>0)or(*strpos*(s,'#')>0)or(*strpos*(s,'$')>0)or(*strpos*(s,'%')>0);  
  
 end;  
 $$ LANGUAGE plpgsql;  
 SELECT *L\_S*('pj#');  
  
  
2  
a  
 create TABLE t(id int, tekst text);  
 CREATE TABLE out( id int generated always as identity , time timestamp(6));  
 CREATE FUNCTION *timest*()  
 RETURNS TRIGGER AS $$  
 BEGIN  
 INSERT INTO out(time) VALUES (*now*());  
 return new;  
 end;  
 $$ LANGUAGE plpgsql;  
 CREATE TRIGGER changes  
 BEFORE INSERT  
 ON t  
 FOR EACH STATEMENT  
 EXECUTE PROCEDURE *timest*();  
SELECT \* FROM out;  
INSERT INTO t(tekst) VALUES ('HTGFKUTFKU');  
  
b  
 CREATE TABLE t2 (id int generated always as identity , birthdate timestamp);  
 CREATE TABLE out2( id int generated always as identity, years int);  
 CREATE FUNCTION *getage*()  
 RETURNS TRIGGER  
 AS $$  
 BEGIN  
 INSERT INTO out2(years) VALUES (*extract*(year from *now*())-*extract*(year from new.birthdate));  
 return new;  
 end;  
 $$ LANGUAGE plpgsql;  
  
 CREATE TRIGGER yearst  
 BEFORE INSERT  
 ON t2  
 FOR EACH ROW  
 EXECUTE PROCEDURE *getage*();  
 SELECT \* FROM out2;  
 INSERT INTO t2(birthdate) VALUES (*now*());  
  
  
c  
 CREATE TABLE t3 (id int generated always as identity , price int);  
 CREATE TABLE out3( id int generated always as identity, price int);  
 CREATE FUNCTION *increase*()  
 RETURNS TRIGGER  
 AS $$  
 BEGIN  
 new.price:=new.price\*1.12;  
 return new;  
 end;  
 $$ LANGUAGE plpgsql;  
drop function *increase*();  
 CREATE TRIGGER incprice  
 BEFORE INSERT  
 ON t3  
 FOR EACH ROW  
 EXECUTE PROCEDURE *increase*();  
drop trigger incprice on t3;  
 SELECT \* FROM t3;  
 INSERT INTO t3(price) VALUES (100);  
  
d  
 CREATE FUNCTION *dont*()  
 RETURNS TRIGGER  
 AS $$  
 BEGIN  
 raise exception using message = 'S 167', detail = 'D 167', hint = 'H 167', errcode = 'P3333';  
  
 end;  
 $$ LANGUAGE plpgsql;  
drop function *dont*;  
 CREATE TRIGGER dontdel  
 BEFORE delete  
 ON t3  
 FOR EACH ROW  
 EXECUTE PROCEDURE *dont*();  
 SELECT \* FROM t3;  
 INSERT INTO t3(price) VALUES (120);  
 DELETE FROM t3;  
  
  
  
e  
 CREATE FUNCTION *pass*(s text)  
 RETURNS bool AS $$  
 BEGIN  
 RETURN (*char\_length*(s)>=8)and((*strpos*(s,'!')>0)or(*strpos*(s,'@')>0)or(*strpos*(s,'#')>0)or(*strpos*(s,'$')>0)or(*strpos*(s,'%')>0));  
 END;  
 $$ LANGUAGE PLPGSQL;  
  
  
 CREATE FUNCTION *L\_S*(s TEXT,  
 OUT len int, OUT symb boolean)  
 AS $$  
 BEGIN  
 len:=*char\_length*(s);  
 symb=(*strpos*(s,'!')>0)or(*strpos*(s,'@')>0)or(*strpos*(s,'#')>0)or(*strpos*(s,'$')>0)or(*strpos*(s,'%')>0);  
  
 end;  
 $$ LANGUAGE plpgsql;  
  
  
 CREATE FUNCTION *doit*()  
 RETURNS TRIGGER  
 AS $$  
 BEGIN  
 raise notice 'pass=% L\_S=%',*pass*(new.tekst),*L\_S*(new.tekst);  
 return new;  
 end;  
 $$ LANGUAGE plpgsql;  
DROP FUNCTION *doit*;  
 CREATE TRIGGER dontdel  
 BEFORE insert  
 ON t  
 FOR EACH ROW  
 EXECUTE PROCEDURE *doit*();  
 drop trigger dontdel on t;  
 SELECT \* FROM out;  
INSERT INTO t(tekst) VALUES ('HTGFKUTFKU');  
  
3  
  
CREATE TABLE t16 (id int primary key ,name varchar(20), date\_of\_birth date, age int, salary int,workexperience int , discount int );  
insert into t16 values (1,'kkk','1999-12-12',12,5000, 24, 1000);  
insert into t16 values (4,'kkfsdg','1999-12-12',10,500000, 1, 2000);  
insert into t16 values (5,'aehyrjk','1999-12-12',1,6000, 10, 3000);  
insert into t16 values (2,'trnk','1999-12-12',2,50, 5, 4000);  
insert into t16 values (3,'opkep','1999-12-12',122,3000, 2, 5000);  
insert into t16 values (6,'GWUEY','1999-12-12',136,1000, 36, 6000);  
  
SELECT \*FROM t16;  
BEGIN;  
 UPDATE t16 SET salary=salary\*1.1\*(workexperience/2);  
 UPDATE t16 SET discount=discount\*1.01 WHERE workexperience>5;  
end;  
  
BEGIN;  
 UPDATE t16 SET salary=salary\*1.15 WHERE age>=40;  
 UPDATE t16 SET salary=salary\*1.15 WHERE workexperience>=8;  
 UPDATE t16 SET discount=discount\*1.2 WHERE workexperience>=8;  
end;