

№1

--a

create function increment(val integer) returns integer as \$\$

begin

return val + 1;

end; \$\$

language plpgsql;

select increment(20);

--b

create function addition(val integer, vak integer) returns integer as \$\$

begin

return val + vak;

end; \$\$

language plpgsql;

select addition(20, 30);

--c

create function check\_divisible(val integer) returns boolean as \$\$

begin

if val % 2 = 0 then return true;

else return false;

end if;

end; \$\$

language plpgsql;

select check\_divisible(30);

--d

create function check\_password(pass varchar) returns boolean as \$\$

begin

if LENGTH(pass) >= 11 then return true;

else return false;

end if;

end; \$\$

language plpgsql;

select check\_password(Omr\_076);

--e

```
create function calculator(val integer, out square_root integer, out square integer) as $$  
    begin  
        square_root := pow(val, 2);  
        square := val * val;  
    end; $$  
language plpgsql;
```

Nº2

--a

```
create function cur() return trigger as $$  
    begin  
        raise notice '%', now();  
        return new;  
    end;  
    $$  
language plpgsql;
```

```
create trigger cur_t before insert on table_1 for each row excute procedure current();
```

-- b

```
create function age() return trigger as $$  
    begin  
        raise notice '%', age(now(), new.t);  
        return new;  
    end; $$  
language plpgsql;
```

```
create trigger age_t before insert on table_2 for each row execute procedure age();
```

--c

```
create function adds() return trigger as $$  
    begin  
        new.cost = new.cost * 1.12;  
        return new;  
    end; $$
```

```
language plpgsql;
```

```
create trigger adds_t before insert on table_3 for each row execute procedure adds();
```

```
--d
```

```
create function stop_deletion() return trigger as $$
```

```
begin
```

```
    raise exception "Deletion is not allowed";
```

```
end; $$
```

```
language plpgsql;
```

```
create trigger stop_d before delete on table_4 execute procedure stop_deletion();
```

```
--e
```

```
create function launches_another() return trigger as $$
```

```
begin
```

```
    raise notice '%', check_password(new.s);
```

```
    raise notice '%', calculator(new.a);
```

```
end; $$
```

```
language plpgsql;
```

```
create trigger launches_t before insert on table_5 for each row execute procedure launchess_another();
```

```
create table work(id int, name varchar, date_of_birth date, age int, inout salary numeric, workexpirence  
int, out discount numeric);
```

№3

```
create table t5 (id int primary key ,name varchar(20), date_of_birth date, age int, salary  
int,workexperience int , discount int );
```

```
insert into t5 values (1,'kkk','1999-12-12',12,5000, 24, 1000);
```

```
insert into t5 values (4,'kkfsdg','1999-12-12',10,500000, 1, 2000);
```

```
insert into t5 values (5,'aehyrjk','1999-12-12',1,6000, 10, 3000);
```

```
insert into t5 values (2,'trnk','1999-12-12',2,50, 5, 4000);
```

```
insert into t5 values (3,'opkep','1999-12-12',122,3000, 2, 5000);
```

```
insert into t t5 values (6,'GWUEY','1999-12-12',136,1000, 36, 6000);
```

```
select *from t5;
```

```
begin;
```

```
    UPDATE t5 SET salary=salary*1.1*(workexperience/2);
```

```
    UPDATE t5 SET discount=discount*1.01 WHERE workexperience>5;
```

end;

BEGIN;

UPDATE t t5 SET salary=salary\*1.15 WHERE age>=40;

UPDATE t5 SET salary=salary\*1.15 WHERE workexperience>=8;

UPDATE t5 SET discount=discount\*1.2 WHERE workexperience>=8;

end;