create database lab7;  
--Sensei 57

--Task 4

create table task4(  
 id integer ,  
 name varchar(255),  
 age integer,  
 data\_of\_birth date,  
 salary integer,  
 work\_experince integer,  
 primary key (id)  
);  
select \* from task4;  
  
--Task 1  
--A  
create or replace function *get\_inc*(a int) returns int  
language plpgsql as  
$$  
begin  
 return a + 1;  
END;  
$$;  
  
select \*  
from *get\_inc*(1);  
  
--b)  
create or replace function *get\_sum*(a int, b int)  
returns int as  
$$  
begin  
 return a + b;  
END;  
$$  
language plpgsql;  
  
--C  
create or replace function *div\_two*(a int, b int)  
 returns bool as  
$$  
begin  
 return (a % 2 = 0 and b % 2 = 0);  
END;  
$$  
language plpgsql;  
  
select \* from *div\_two*(4, 12);  
  
--D  
create or replace function *check\_pass*(s varchar)  
 returns bool as $$  
begin  
 return s like 'S%' ;  
END;  
$$  
language plpgsql;  
  
select \* from *check\_pass*('some\_password');  
  
--E  
create or replace function *divides\_on\_3and4*(a int, out divides\_by\_3 bool, out divides\_by\_4 bool)  
as $$  
begin  
 divides\_by\_3 := a % 3 = 0;  
 divides\_by\_4 := a % 4 = 0;  
END;  
$$  
language plpgsql;  
  
select \* from *divides\_on\_3and4*(4);  
  
--TASK 2  
--A  
create function *insert\_timestamp*()  
 returns trigger  
 language plpgsql  
as $$  
begin  
 raise notice '%',*current\_timestamp*;  
 return new;  
end;  
$$;  
  
create or replace trigger time\_of\_action\_trigger  
 before insert on lab7  
 for each row  
execute function *insert\_timestamp*();  
  
insert into lab7 values(3, 'Mike', '1979-02-17 12:18:31.056275', 34, 55000, 10, 10);  
  
--B  
drop function *get\_age*();  
create function *get\_age*()  
returns trigger  
language plpgsql as $$  
begin  
raise notice '%', *age*(*current\_timestamp*, new.date\_of\_birth);  
return new;  
end;  
$$;  
  
drop trigger insert\_birthdate\_trigger on lab7;  
  
create or replace trigger insert\_birthdate\_trigger  
before insert on lab7  
for each row  
execute function *get\_age*();  
  
--C  
drop function *compute\_tax*();  
create function *compute\_tax*()  
returns trigger  
language plpgsql as $$ begin  
new.price = new.price \* 1.12;  
return new;  
end;  
$$;  
  
drop trigger insert\_item\_trigger on lab7;  
  
create trigger insert\_item\_trigger  
before insert on lab7  
for each row  
execute function *compute\_tax*();  
  
--D  
create function *prevent\_deletion*()  
returns trigger  
language plpgsql as $$  
begin  
raise exception 'Row is not deleted';  
end;  
$$;  
  
create trigger do\_not\_delete\_trigger  
before delete on lab7  
for each row  
execute function *prevent\_deletion*();  
  
--E  
create function *launch\_ll*()  
returns trigger  
language plpgsql as $$  
begin  
raise notice '%', *check\_pass*(new.name);  
raise notice '%', *divides\_on\_3and4*(new.age);  
end;  
$$;  
  
create trigger launch\_functions\_trigger  
before insert on lab7  
for each row  
execute function *launch\_ll*();  
  
insert into lab7(id, name, age) values(2, 'Sarah', 48);  
  
--TASK 3  
  
--A  
create function *incr\_sal*(inout name varchar, inout age int, inout salary int, in work\_experience int, out discount int)  
language plpgsql as $$  
declare  
 cnt int := work\_experience/2;  
 cnt2 int := work\_experience/5;  
begin  
 discount = 10;  
 for loop\_cnt in 1..cnt loop  
 salary = salary \* 1.1;  
 end loop;  
 for loop\_cnt2 in 1..cnt2 loop  
 discount = discount \* 1.01;  
 end loop;  
end;  
$$;  
  
select \* from *incr\_sal*('Anna', 45, 34000, 3);  
  
--B  
create or replace function *inc\_sal*(inout name varchar, inout age int, inout salary int, inout work\_experience int, out discount int)  
language plpgsql as $$  
begin  
 if age >= 40 then salary = salary \* 1.15;  
 end if;  
 if work\_experience > 8 then salary = salary \* 1.15;  
 end if;  
 if work\_experience > 8 then discount = 20;  
 end if;  
end;  
$$;  
  
--Task 4  
select \* from *inc\_sal*('ROMA', 80, 1230000, 5);