|  |
| --- |
| create function increment(n integer) |
|  | returns integer |
|  | language plpgsql |
|  | as |
|  | $$ |
|  | begin |
|  | return n+1; |
|  | end |
|  | $$; |
|  |  |
|  | select increment(5); |
|  |  |
|  | --b |
|  | create function summa(x integer, y integer) |
|  | returns integer |
|  | language plpgsql |
|  | as |
|  | $$ |
|  | begin |
|  | return x+y; |
|  | end; |
|  | $$; |
|  |  |
|  | select summa(2, 3); |
|  |  |
|  | --c |
|  | create or replace function is\_even(n integer) |
|  | returns boolean |
|  | language plpgsql |
|  | as |
|  | $$ |
|  | begin |
|  | if n%2 = 0 then |
|  | return true; |
|  | else |
|  | return false; |
|  | end if; |
|  | end; |
|  | $$; |
|  |  |
|  | select is\_even(12); |
|  |  |
|  |  |
|  | --d |
|  | create or replace function is\_valid(x varchar(20)) |
|  | returns boolean |
|  | language plpgsql |
|  | as |
|  | $$ |
|  | begin |
|  | if x ~ '^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)(?=.\*[@$!%\*?&])[A-Za-z\d@$!%\*?&]{8,}$' then |
|  | return true; |
|  | else |
|  | return false; |
|  | end if; |
|  | end |
|  | $$; |
|  |  |
|  | select is\_valid('Qucherme21\*'); |
|  |  |
|  | --e |
|  | create or replace function two\_output(x varchar(30)) |
|  | returns record |
|  | language plpgsql |
|  | as |
|  | $$ |
|  | declare y record; |
|  | begin |
|  | select split\_part(x, ' ', 1) f, |
|  | split\_part(x, ' ', 2) s2 |
|  | into y; |
|  | return y; |
|  | end; |
|  | $$; |
|  |  |
|  | select two\_output('BAGa Maga'); |
|  |  |
|  |  |
|  |  |
|  | --task 2. |
|  |  |
|  | --task 3. In Postgres, the main functional difference between a function and a |
|  | -- stored procedure is that a function returns a result, but a stored procedure |
|  | -- does not. This is because the intent of the stored procedure is to perform |
|  | -- some action and then complete it, which will then return control to the caller. |
|  |  |
|  |  |
|  | --task4 |
|  |  |
|  | create table workers( |
|  | id int primary key, |
|  | name varchar(30), |
|  | birthday date, |
|  | age integer, |
|  | salary integer, |
|  | experience\_in\_years integer, |
|  | discount\_percent int |
|  | ); |
|  |  |
|  | insert into workers values (1,'Stalin', '1955-01-02' , 66, 250000, 30, 0); |
|  |  |
|  | -- a |
|  | create procedure a() |
|  | language plpgsql |
|  | as |
|  |  |
|  | $$ |
|  | declare |
|  | years int; |
|  | begin |
|  | update workers set salary = salary + (salary\*(experience\_in\_years \* 0.1)) where experience\_in\_years>1; |
|  | update workers set discount\_percent = 10 where experience\_in\_years >= 5; |
|  | update workers set discount\_percent = discount\_percent + 1 where experience\_in\_years > 5 and experience\_in\_years / 5 >0; |
|  |  |
|  | end; |
|  | $$; |
|  |  |
|  | call a() |
|  |  |
|  | --b |
|  |  |
|  | create procedure b() |
|  | language plpgsql |
|  | as |
|  | $$ |
|  | begin |
|  | update workers set salary = salary + (salary\*0.15) where age > 40; |
|  | update workers set salary = salary + (salary\*0.15) where experience\_in\_years > 8; |
|  | update workers set discount\_percent = discount\_percent + 20 where experience\_in\_years > 8; |
|  | end; |
|  | $$; |
|  |  |
|  | call b(); |
|  |  |
|  |  |
|  | --task 5 |
|  |  |
|  | with recursive recommenders(recommender, member) as ( |
|  | select recommendedby, memid |
|  | from members |
|  | union all |
|  | select mems.recommendedby, recs.member |
|  | from recommenders recs |
|  | inner join members mems |
|  | on mems.memid = recs.recommender |
|  | ) |
|  | select recs.member member, recs.recommender, mems.firstname, mems.surname |
|  | from recommenders recs |
|  | inner join members mems |
|  | on recs.recommender = mems.memid |
|  | where recs.member = 22 or recs.member = 12 |
|  | order by recs.member asc, recs.recommender desc |