**User Defined Functions we can also call it as custom functions**

In Snowflake, UDFs allow us to extend SQL functionality. I’ve used SQL UDFs for masking sensitive data, JavaScript UDFs for business logic like tax calculations, and I’m aware of external UDFs to integrate with APIs. They help make queries reusable, cleaner, and more secure. we define udf as Secure or unsecure.

There are 2 types of functions, if udfs returns the single value we call it as scalar functions and its returns table or set of rows we call it as tabular functions. We can use these ufds in select, where having clauses.

show functions;

-- Scalar

create function fn\_area\_of\_triangle (base float, height float)

returns float

language SQL

as

$$

0.5\*base\*height

$$

;

select fn\_area\_of\_triangle (10,20) ; -- SELECT, WHERE, HAVING

show functions like 'fn%';

desc function fn\_area\_of\_triangle(float,float);

-- secure scalar function

create secure function fn\_area\_of\_triangle\_sec (base float, height float)

returns float

language SQL

as

$$

0.5\*base\*height

$$

;

select fn\_area\_of\_triangle\_sec (10,20) ; -- SELECT, WHERE, HAVING

show functions like 'fn%';

desc function fn\_area\_of\_triangle(float,float);

create role developer;

grant usage on database practice\_db to role developer;

grant usage on warehouse compute\_wh to role developer;

grant usage on schema snowflake to role developer;

grant usage on function PRACTICE\_DB.SNOWFLAKE.FN\_AREA\_OF\_TRIANGLE(float,float) to role developer;

grant usage on function PRACTICE\_DB.SNOWFLAKE.FN\_AREA\_OF\_TRIANGLE\_SEC(float,float) to role developer;

grant role developer to user DEVELOPER1;

show users;

use role developer;

---------------------------------------------------------------------------------------------

-- Tabular user defined function

show tables ;

drop table client ;

create table client

( id NUMBER(38,0), first\_name VARCHAR(16), last\_name VARCHAR(50), sex VARCHAR(1), ethinicity VARCHAR(30), ssn VARCHAR(15), street\_address VARCHAR(90),status VARCHAR(10)

);

insert into client values (111111, 'James', 'Schwartz', 'M', 'American','342-76-9087','5676 Washington Street','ACTIVE') ;

insert into client values (222222, 'Jessica', 'Escobar', 'F', 'Hispanic','456-93-5629','3234 WateringCan Drive','INACTIVE') ;

insert into client values (333333, 'Ben', 'Hardy', 'M', 'American','876-98-3245','6578 Historic Circle','INACTIVE') ;

insert into client values (444444, 'Anjali', 'Singh', 'F', 'Indian American','435-87-6532','8978 Autumn Day Drive','ACTIVE') ;

insert into client values (555555, 'Dean', 'Tracy', 'M', 'African','767-34-7656','2343 India Street','ACTIVE') ;

select \* from client ;

-- tabular function

create or replace function fn\_active\_client (p\_status varchar)

returns table (id number, first\_name varchar, last\_name varchar, sex varchar, ssn varchar, status varchar)

as

$$

select id, first\_name, last\_name, sex, ssn, status from client where status = p\_status

$$

;

select id, first\_name, last\_name, sex, ssn, status from table(fn\_active\_client ('ACTIVE')) ;