

CREATING REDSHIFT USER DEFINED FUNCTION (UDF) IN PYTHON



You can create user defined functions in [Amazon Redshift](#) in [Python](#). If you use Amazon Lambda containerless virtual machines, then you can use additional languages. (Using [AWS Lambda](#), you will have some usage costs. But, unless you're budget-conscious, that's no reason not to use it.)

You can add third-party libraries. In the case of Python, you could use Pandas NumPy, for example.

UDF example

Let's walk through a simple example. This is a scalar function, meaning it returns a single value.

First create a table:

```
create table orders(  
  customernumber integer,  
  ordernumber integer,  
  orderdate date,  
  quantity smallint ,  
  discount decimal(3,2) ,  
  price decimal(8,2),  
  primary key(customernumber, ordernumber));
```

Then add one record to it.

```
insert into orders(customernumber ,ordernumber ,orderdate ,quantity
,discount, price )
values(123, 456, '2020-10-20', 100, 0, 30)
```

Then create a function. Notice the odd language name **plpythonu**. That is for historical reasons as it's the name [PostgreSQL](#) uses for their Python Procedural Language. Of course, Redshift is not PostgreSQL.

The function format is basically **functionName(arguments ...)**

Then give it a return type:

```
create function revenue (price float, quantity float)
returns float
stable
as $$
return price * quantity
$$ language plpythonu;
```

Now run that function over the price and quantity columns in the orders table.

```
select price, quantity, revenue(price, quantity)
from orders
```

Here are the results:

```
30.00,100,3000.0
```

Additional resources

For more tutorials like this, explore these resources:

- [BMC Machine Learning & Big Data Blog](#)
- [How To Load Data to Amazon Redshift from S3](#)
- [How To Copy JSON Data to an Amazon Redshift Table](#)
- [Writing SQL Statements in Amazon Redshift](#)
- [AWS Guide](#)
- [Snowflake Guide](#)