

Course Summary

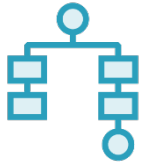


Gerald Britton

IT SOLUTIONS DESIGNER, DATABASE SPECIALIST

@GeraldBritton www.linkedin.com/in/geraldbritton

Business Use Cases



Encapsulate business logic



Enforce semantic relationships



Validate and transform data



Control access to sensitive data



From Queries to Functions

SELECT ..
FROM ..

CREATE VIEW ...

CREATE
FUNCTION ...
(arguments)

First Function

```
1 CREATE OR REPLACE FUNCTION cust_order(customerid INT)
2     RETURNS TABLE (
3         firstname VARCHAR,
4         lastname VARCHAR,
5         orderid INT,
6         netamount NUMERIC)
7 AS $$
8     SELECT c.firstname,
9            c.lastname,
10           o.orderid,
11           o.netamount
12     FROM customers c
13     JOIN orders o ON c.customerid = o.customerid
14     WHERE c.customerid = cust_order.customerid;
15 $$ LANGUAGE SQL;
```

PL/pgSQL program structure

```
[ <<label>> ]  
[ DECLARE  
    declarations ]  
BEGIN  
    statements  
END [ label ];
```

Using variables

Executing queries

Control structures

Iteration

Cursors

Dynamic queries

Polymorphic Functions

Ad Hoc

Subtype

Parametric

Variadic

Defaults

Securing Your Functions



Principle of least permission



Double abstraction

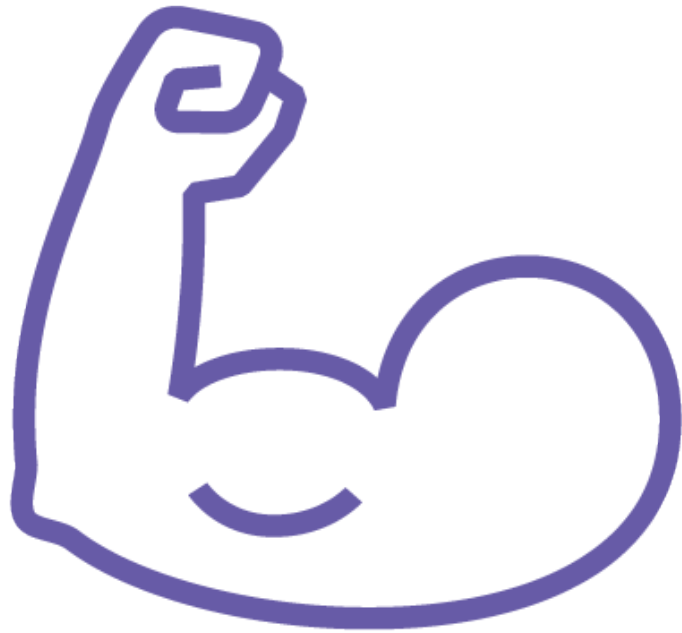


Using function attributes



Security using schemas





Thinking defensively

False assumptions

Environment settings

Database objects

Handling errors

Course Summary



Gerald Britton

IT SOLUTIONS DESIGNER, DATABASE SPECIALIST

@GeraldBritton www.linkedin.com/in/geraldbritton