Constraints.

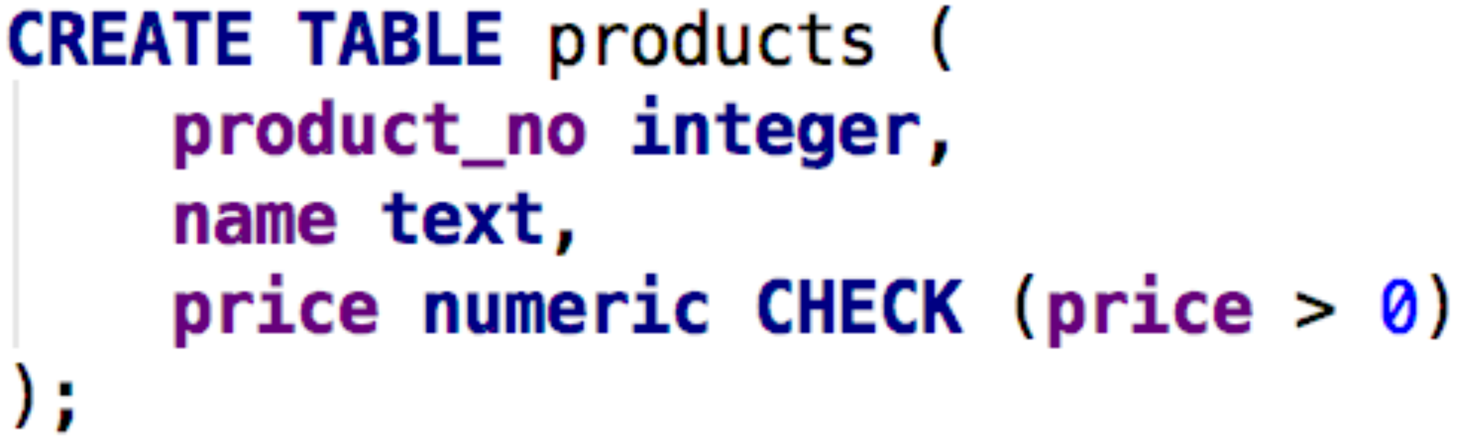
Overview

- Check constraints
- Not-Null constraints
- Unique constraints
- Primary keys
- Foreign keys

CHECK

- A check constraint is the most generic constraint type.

- It allows you to specify that the value in a certain column must satisfy a Boolean (truth-value) expression.



```
CREATE TABLE products (
    product_no integer,
    name text,
    price numeric CHECK (price > 0)
);
```

CHECK

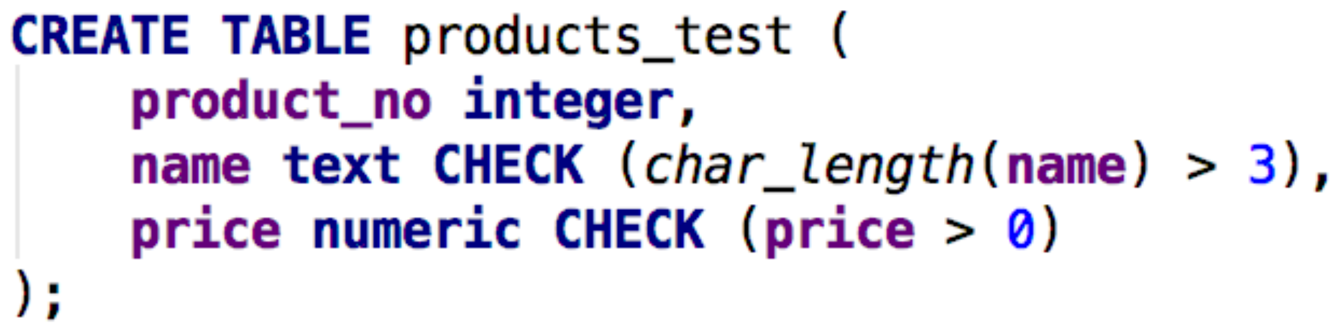
- Constraint definition comes after the data type, just like default value definitions.

- Default values and constraints can be listed in any order.

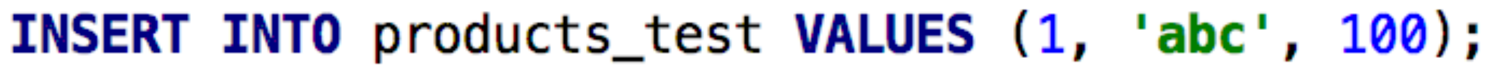
- A check constraint consists of the key word CHECK followed by an expression in parentheses.

- The check constraint expression should involve the column thus constrained, otherwise the constraint would not make too much sense.

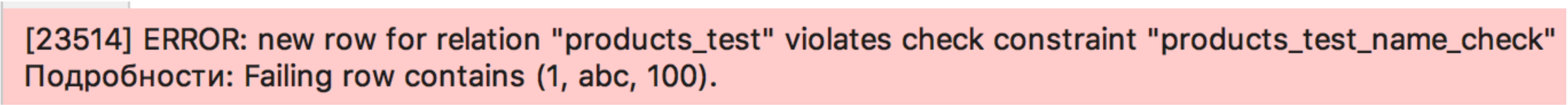
CHECK



```sql
CREATE TABLE products_test (
    product_no integer,
    name text CHECK (char_length(name) > 3),
    price numeric CHECK (price > 0)
);
```

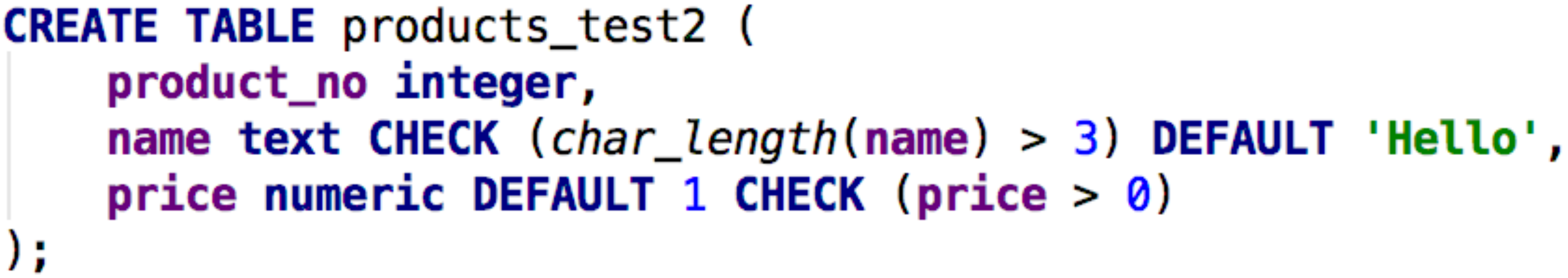


```sql
INSERT INTO products_test VALUES (1, 'abc', 100);
```



```
[23514] ERROR: new row for relation "products_test" violates check constraint "products_test_name_check"
Подробности: Failing row contains (1, abc, 100).
```

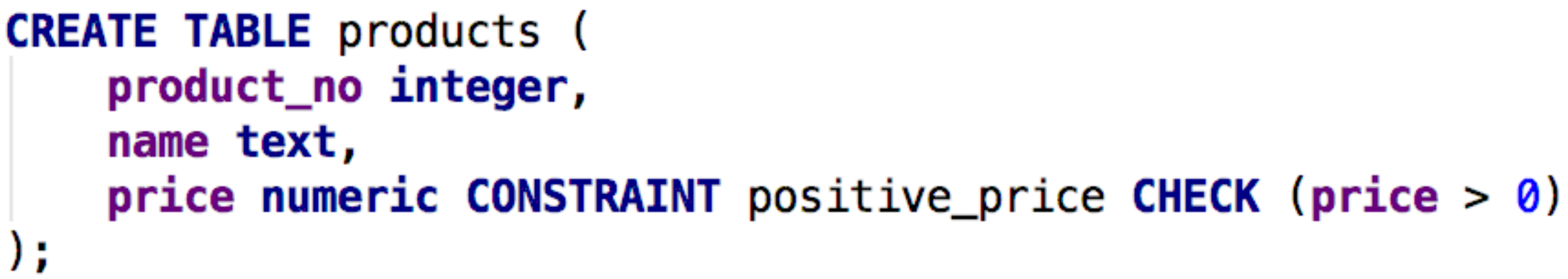
CHECK



```sql
CREATE TABLE products_test2 (
    product_no integer,
    name text CHECK (char_length(name) > 3) DEFAULT 'Hello',
    price numeric DEFAULT 1 CHECK (price > 0)
);
```

CHECK

- You can also give the constraint a separate name.
- This clarifies error messages and allows you to refer to the constraint when you need to change it.
- The syntax is:



```
CREATE TABLE products (
    product_no integer,
    name text,
    price numeric CONSTRAINT positive_price CHECK (price > 0)
);
```

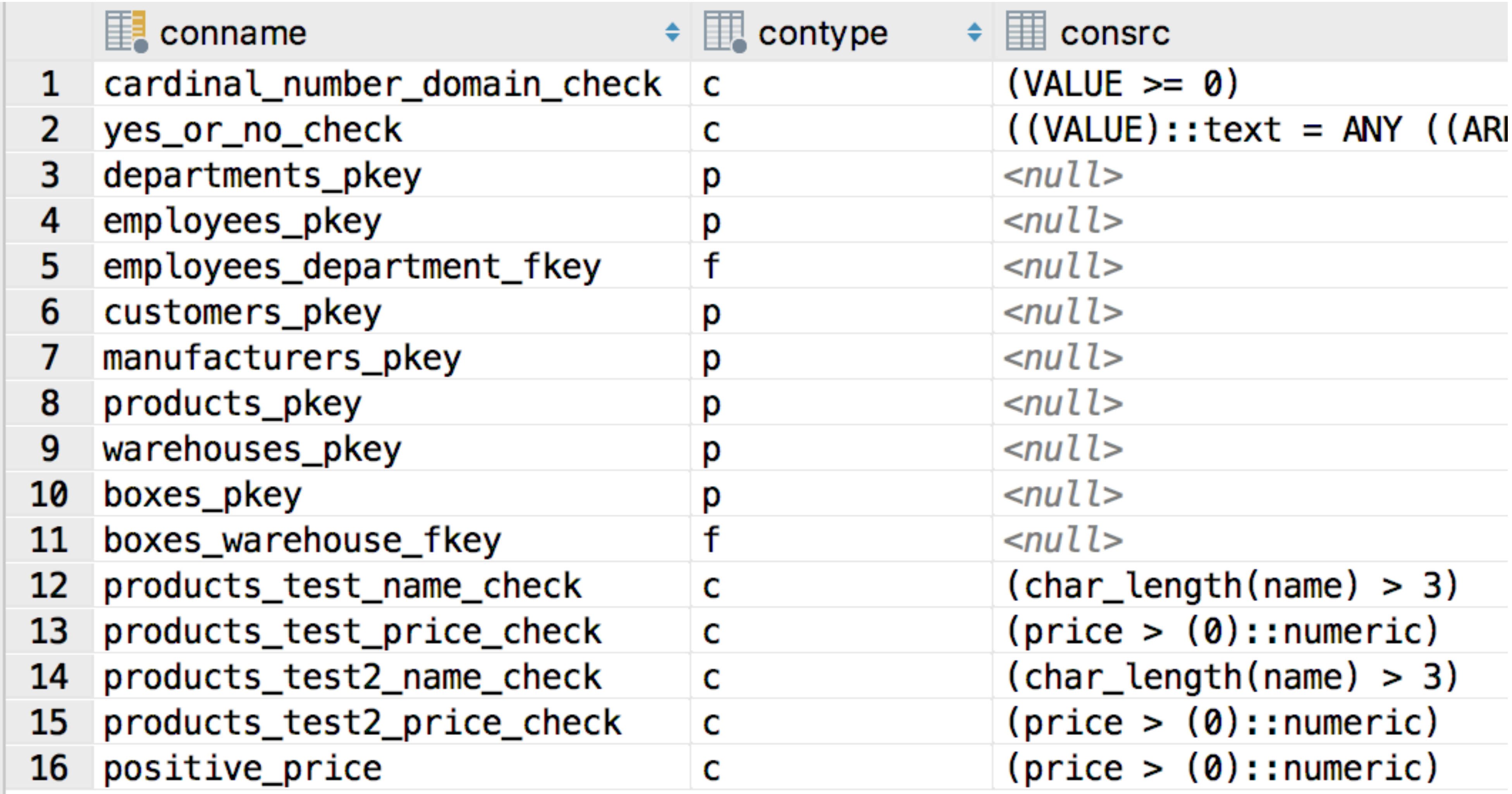
CHECK

- To specify a named constraint, use the key word CONSTRAINT followed by an identifier followed by the constraint definition.

- If you don't specify a constraint name in this way, the system chooses a name for you.

CHECK

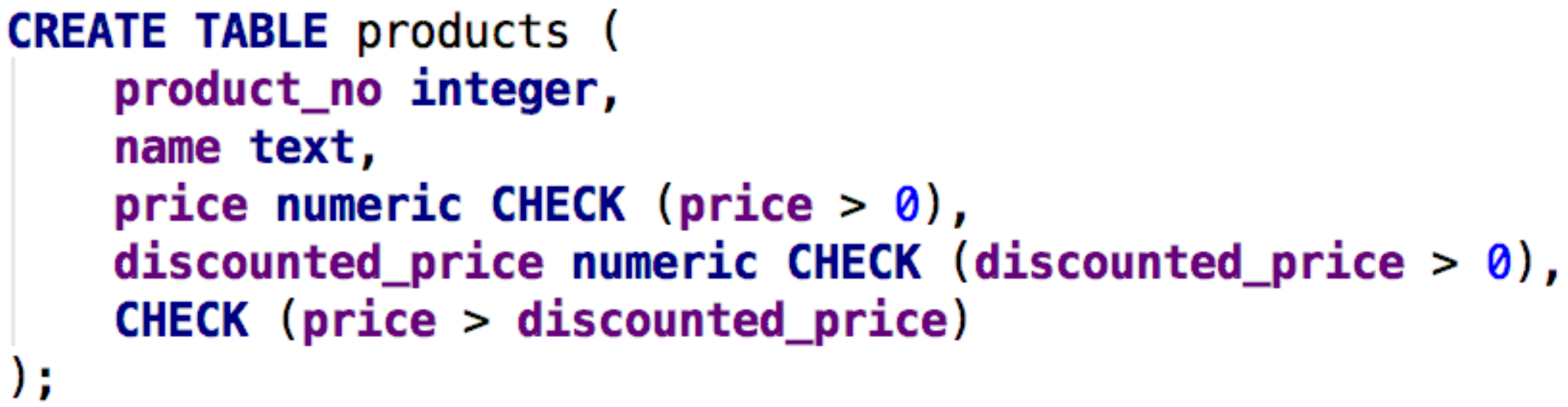
```
SELECT conname, contype, consrc FROM pg_constraint;
```



	conname	contype	consrc
1	cardinal_number_domain_check	c	(VALUE >= 0)
2	yes_or_no_check	c	((VALUE)::text = ANY ((AR
3	departments_pkey	p	<null>
4	employees_pkey	p	<null>
5	employees_department_fkey	f	<null>
6	customers_pkey	p	<null>
7	manufacturers_pkey	p	<null>
8	products_pkey	p	<null>
9	warehouses_pkey	p	<null>
10	boxes_pkey	p	<null>
11	boxes_warehouse_fkey	f	<null>
12	products_test_name_check	c	(char_length(name) > 3)
13	products_test_price_check	c	(price > (0)::numeric)
14	products_test2_name_check	c	(char_length(name) > 3)
15	products_test2_price_check	c	(price > (0)::numeric)
16	positive_price	c	(price > (0)::numeric)

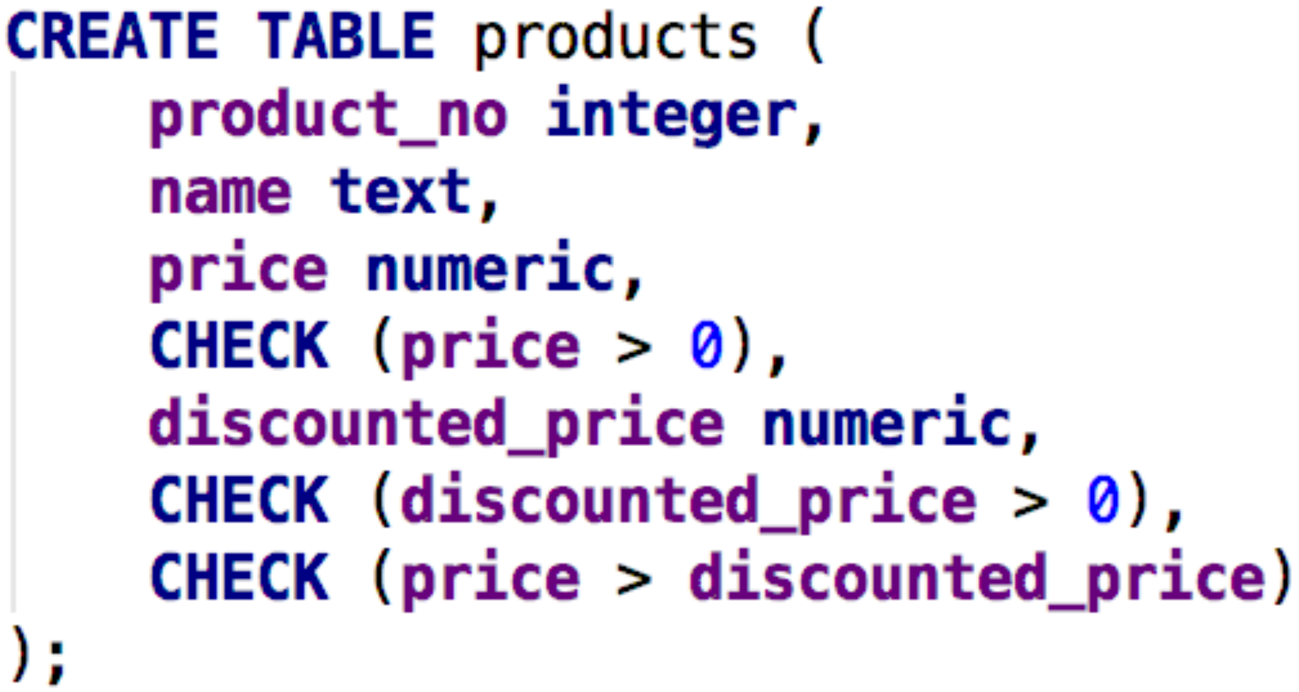
CHECK

- A check constraint can also refer to several columns.
- Say you store a regular price and a discounted price, and you want to ensure that the discounted price is lower than the regular price:



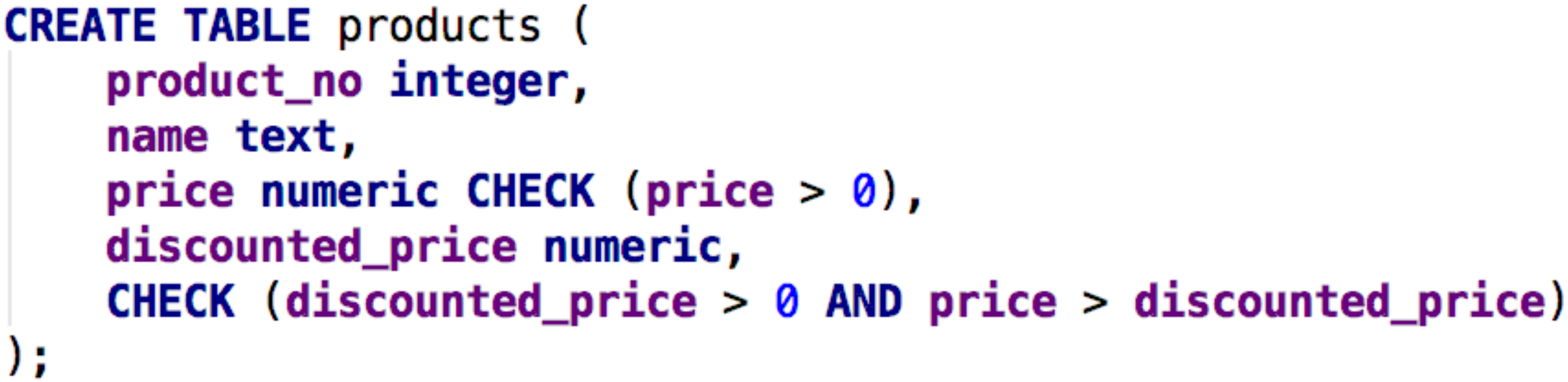
```
CREATE TABLE products (
    product_no integer,
    name text,
    price numeric CHECK (price > 0),
    discounted_price numeric CHECK (discounted_price > 0),
    CHECK (price > discounted_price)
);
```

CHECK



```
CREATE TABLE products (
    product_no integer,
    name text,
    price numeric,
    CHECK (price > 0),
    discounted_price numeric,
    CHECK (discounted_price > 0),
    CHECK (price > discounted_price)
);
```

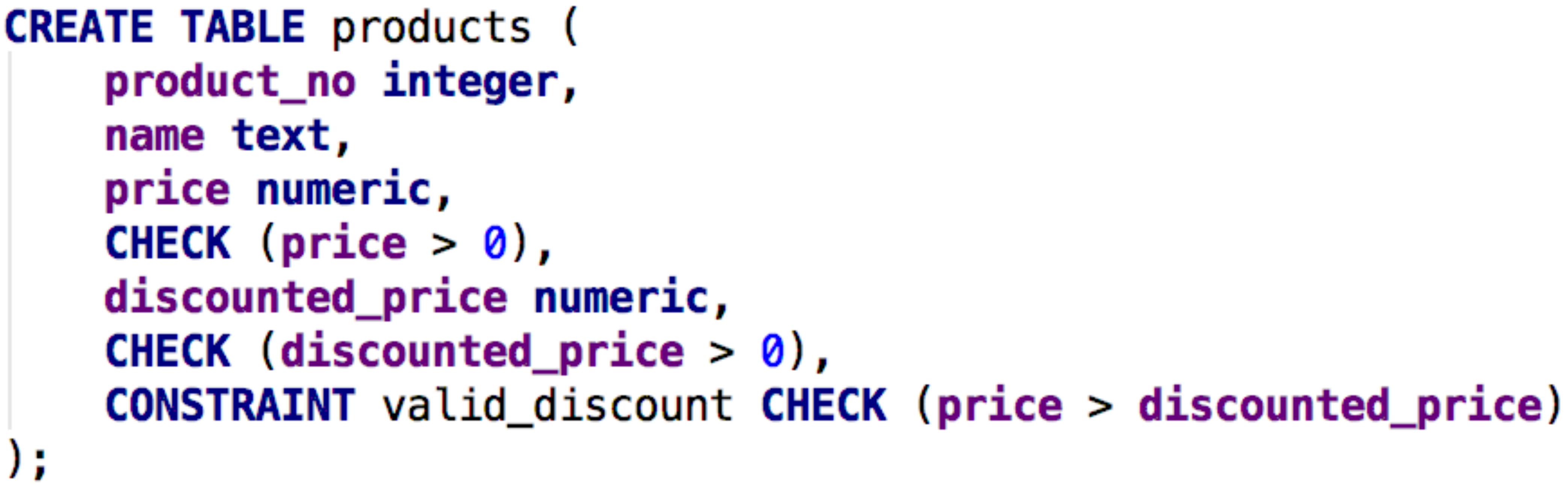
CHECK



```sql
CREATE TABLE products (
    product_no integer,
    name text,
    price numeric CHECK (price > 0),
    discounted_price numeric,
    CHECK (discounted_price > 0 AND price > discounted_price)
);
```

CHECK

- Names can be assigned to table constraints in the same way as column constraints:



```
CREATE TABLE products (
    product_no integer,
    name text,
    price numeric,
    CHECK (price > 0),
    discounted_price numeric,
    CHECK (discounted_price > 0),
    CONSTRAINT valid_discount CHECK (price > discounted_price)
);
```

Questions?