Challenge 11: Customer's orders

Friday, April 7, 2017 12:11 PM

Step 1

We've created a database for customers and their orders. Not all of the customers have made orders, however. Come up with a query that lists the name and email of every customer followed by the item and price of orders they've made. Use a LEFT OUTER JOIN so that a customer is listed even if they've made no orders, and don't add any ORDER BY.

```
CREATE TABLE customers (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    name TEXT,
    email TEXT);

INSERT INTO customers (name, email) VALUES ("Doctor Who", "doctorwho@timelords.com");
INSERT INTO customers (name, email) VALUES ("Harry Potter", "harry@potter.com");
INSERT INTO customers (name, email) VALUES ("Captain Awesome", "captain@awesome.com");

CREATE TABLE orders (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    customer_id INTEGER,
    item TEXT,
    price REAL);

INSERT INTO orders (customer_id, item, price)
    VALUES (1, "Sonic Screwdriver", 1000.00);
INSERT INTO orders (customer_id, item, price)
    VALUES (2, "High Quality Broomstick", 40.00);
INSERT INTO orders (customer_id, item, price)
    VALUES (1, "TARDIS", 1000000.00);

SELECT customers.name, customers.email, orders.item, orders.price FROM customers
LEFT OUTER JOIN orders ON customers.id = orders.customer_id
```

DATABASE SCHEMA

customers 3 rows
id (PK)INTEGER
nameTEXT
emailTEXT

orders 3 rows
id (PK)INTEGER
customer_idINTEGER
itemTEXT
priceREAL

RESULTS

name	email	item	price
Doctor Who	doctorwho@timelords.com	Sonic Screwdriver	1000
Doctor Who	doctorwho@timelords.com	TARDIS	1000000
Harry Potter	harry@potter.com	High Quality Broomstick	40
Captain Awesome	captain@awesome.com	NULL	NULL

Step 2

Now, create another query that will result in one row per each customer, with their name, email, and total amount of money they've spent on orders. Sort the rows according to the total money spent, from the most spent to the least spent.

(Tip: You should always GROUP BY on the column that is most likely to be unique in a row.)

```
SELECT customers.name, customers.email, SUM(orders.price) as "Total Amount" FROM customers
LEFT OUTER JOIN orders ON customers.id = orders.customer_id
GROUP BY customers.name
ORDER BY "Total Amount" DESC;
```

Step 3