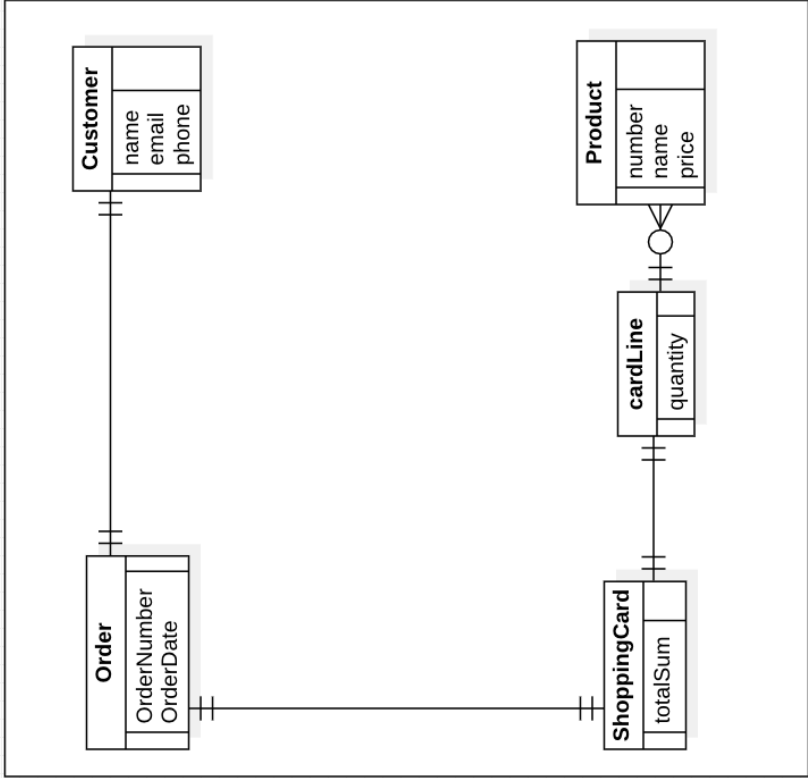


relational



Document

```
Order = {
  OrderNumber: 122435,
  OrderDate: 11/09/2021,
  customer: Customer {
    "name": "Frank Brown",
    "email": "frank@deloit.com",
    "phone": "2234554234"
  },
  shoppingCard: ShoppingCard {
    "cardLine": {
      "quantity": 2,
      products: [
        Product {
          "number": "A546",
          "name": "Iphone 12",
          "price": 980
        },
        Product {
          "number": "S333",
          "name": "Samsung Galaxy 12S",
          "price": 800
        }
      ]
    },
    "totalSum": 5160.00
  }
}
```

Cassandra

```
create table Order {
  OrderNumber int,
  OrderDate long,
  name text,
  email text,
  phone text,
  total int
  productNumber text,
  productName text,
  price int
  quantity int
  primaryKey((OrderNumber, OrderDate, total),
  name, email, phone)
}
```

Graph

```
Create (
:Order {
  OrderNumber: 122435,
  OrderDate: 11/09/2021
}
Create {
:Customer {
  name: "Frank Brown",
  email: "frank@deloit.com"
  phone: "2234554234"
}
Create {
:Product {
  "number": "A546",
  "name": "Iphone 12",
  "price": 980
}
:Product {
  "number": "S333",
  "name": "Samsung Galaxy 12S",
  "price": 800
}
}
Match (o: Order), (c: Customer), (p: Product)
where o.OrderNumber = "122435" and c.name= "Frank Brown" and
(p.number = "A546" or p.product = "A333")
CREATE (c) - [:shopped {totalSum : 5160}] -> (o),
CREATE (o) - [:includes] -> (p)
return o, c, p
```