Aggregates

Aggregate

A cluster of domain objects that can be treated as a single unit

Boundaries

Transaction

Distribution

Concurrency

Aggregates are consistency islands

Moloka‘i

Maui

Lāna‘i

Kaho‘olawe

1000

Aggregates are always internally consistent.

- invariants apply at every transaction commit

Aggregates are “eventually consistent” with each other.

- asynchronous updates propagate through system

Guidelines for first-pass aggregate boundaries

Meaningful conceptual whole

“Delete” rule of thumb

Ultimate criterion is *usefulness*

5 Rules of Thumb for Aggregate Design

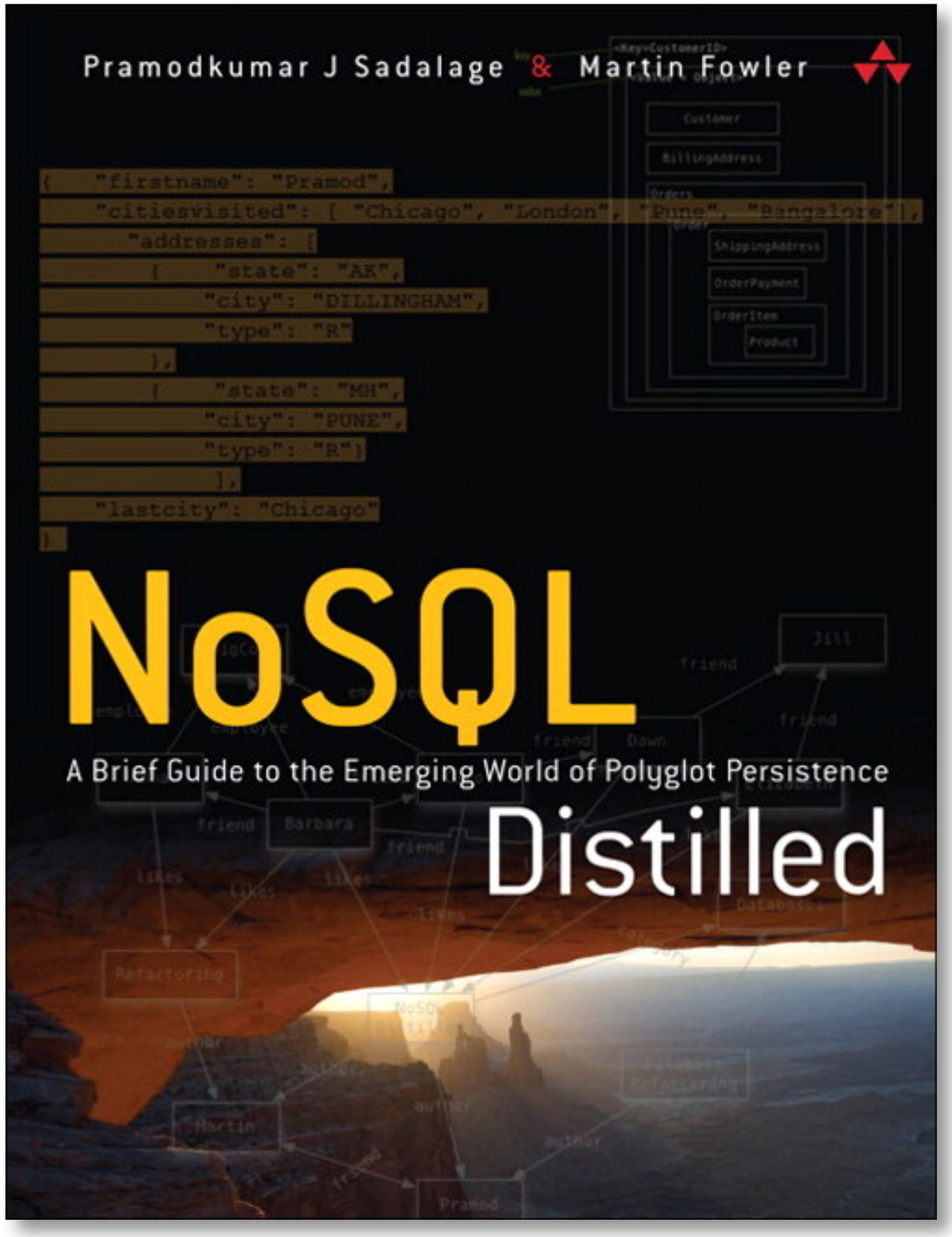
Model true invariants within aggregate boundaries

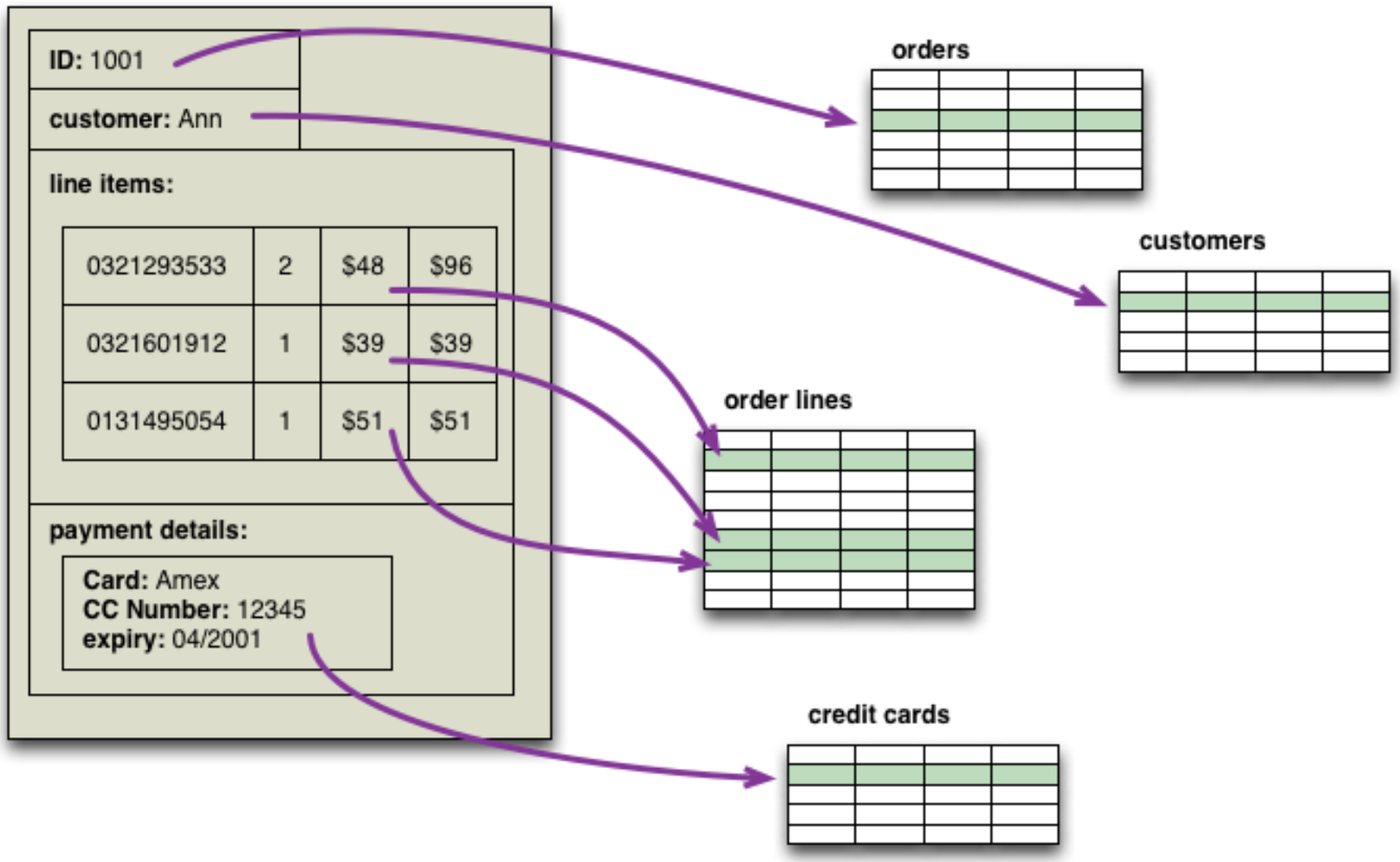
Only hold references to aggregate roots

Use eventual consistency across aggregate boundaries

Design small aggregates

Reference other aggregates by identity

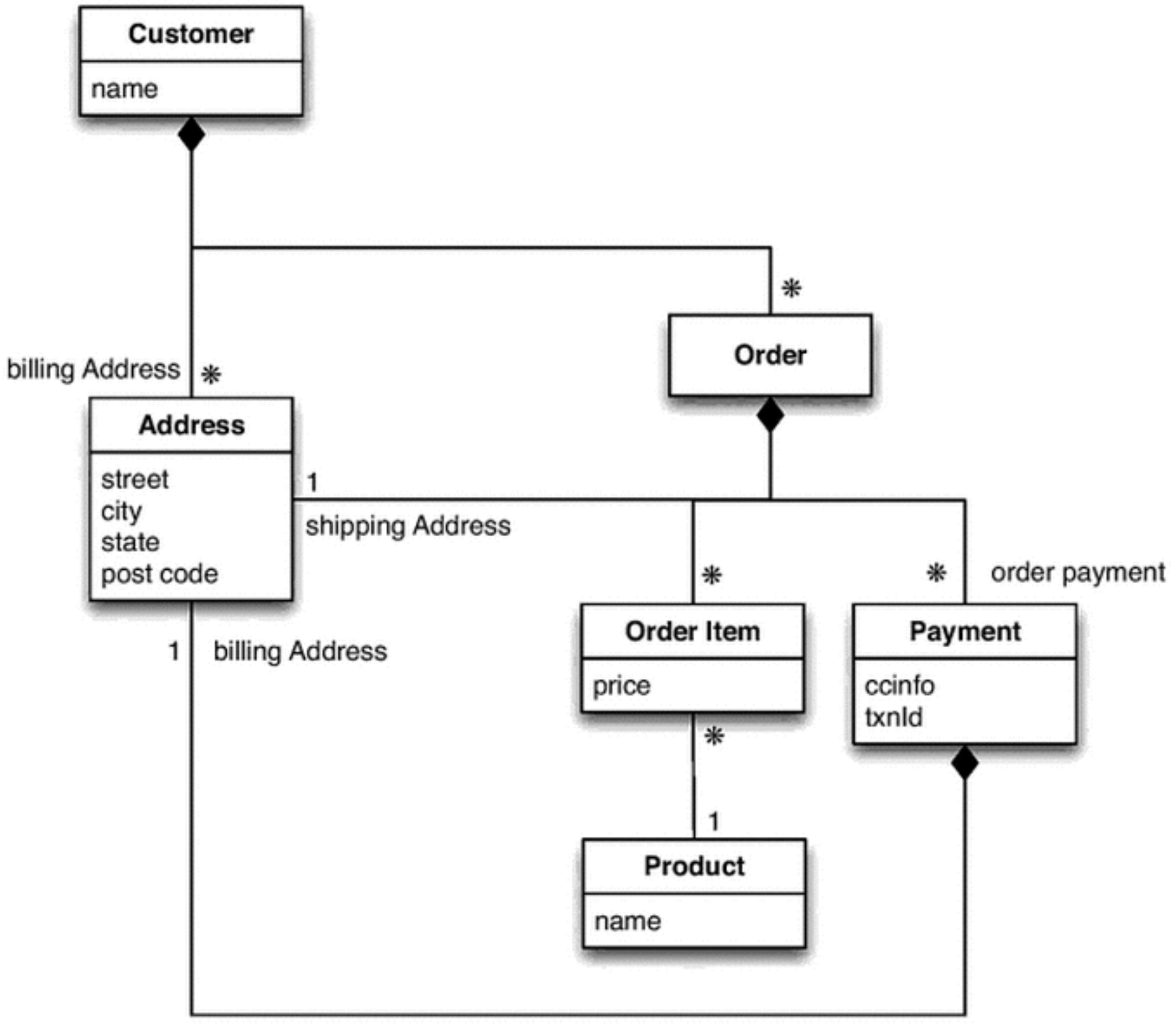

Pramodkumar J Sadalage & Martin Fowler
{ "firstname": "Pramod",
"citiesvisited": ["Chicago", "London", "Pune", "Bangalore"],
"addresses": [
{ "state": "AR",
"city": "DILLINGHAM",
"type": "R"
},
{ "state": "MH",
"city": "PUNE",
"type": "R"}
],
"lastcity": "Chicago"
}
NoSQL
A Brief Guide to the Emerging World of Polyglot Persistence
Distilled

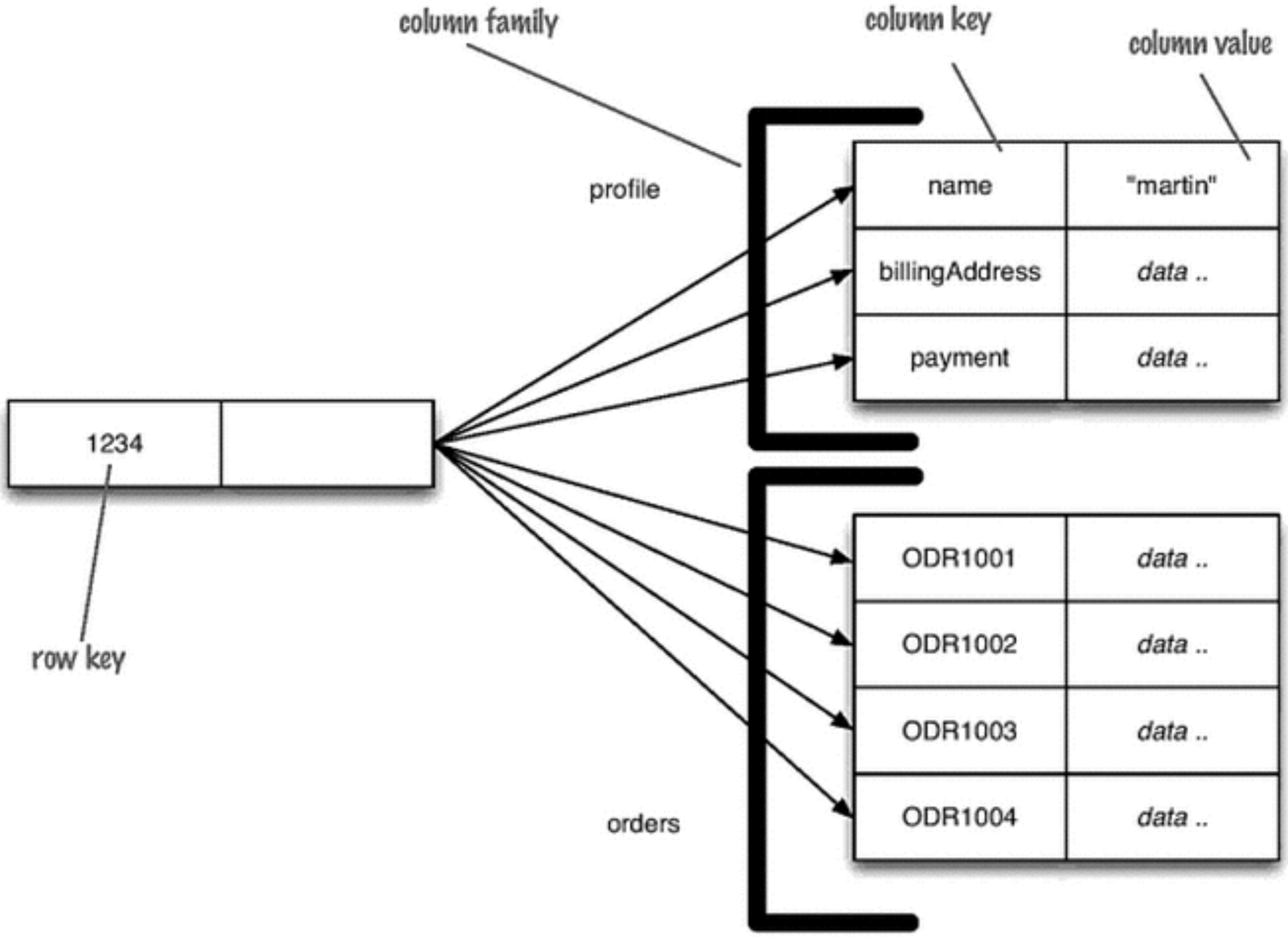

ID: 1001
customer: Ann
line items:
0321293533
2
$48
$96
0321601912
1
$39
$39
0131495054
1
$51
$51
payment details:
Card: Amex
CC Number: 12345
expiry: 04/2001
orders
customers
order lines
credit cards

Aggregates are the basic element of transfer of data storage

- you request to load or save *whole aggregates.*

```
{
    orderid: '1001',
    customerid: '232322',
    customername: 'Ann Rogers',
    shippingaddress: {
        street: '1343 Broadway Street',
        city: 'Denver',
        zipcode: '80210',
        state: 'CO'
    }
    lineitems: [
        {id: '032193533', count: '2', cost: '$48', amount: '$96'},
        {id: '0321601912', count: '1', cost: '$39', amount: '$39'},
        {id: '013495054', count: '1', cost: '$51', amount: '$51'}
    ]
    paymentdetails : {
        card: 'Amex',
        ccnumber: '12345',
        expiry: '04/2001'
    }
}
```


Customer
name
billing Address
*
*
Order
Address
street
city
state
post code
1
shipping Address
1
billing Address
*
*
order payment
Order Item
price
Payment
ccinfo
txnId
*
1
Product
name


column family
column key
column value
profile
name
"martin"
billingAddress
data ..
payment
data ..
1234
row key
ODR1001
data ..
ODR1002
data ..
ODR1003
data ..
ODR1004
data ..
orders

An aggregate is a collection of data that we interact with as a unit.

Aggregates form the boundaries for ACID operations with the database.

Key-value, document, and column-family databases can all be seen as forms of aggregate-oriented database.

Aggregates make it easier for the database to manage data storage over clusters.

Aggregate-oriented databases work best

when most data interaction is done with the same aggregate;

aggregate-ignorant databases are better

when interactions use data organized in many different formations.