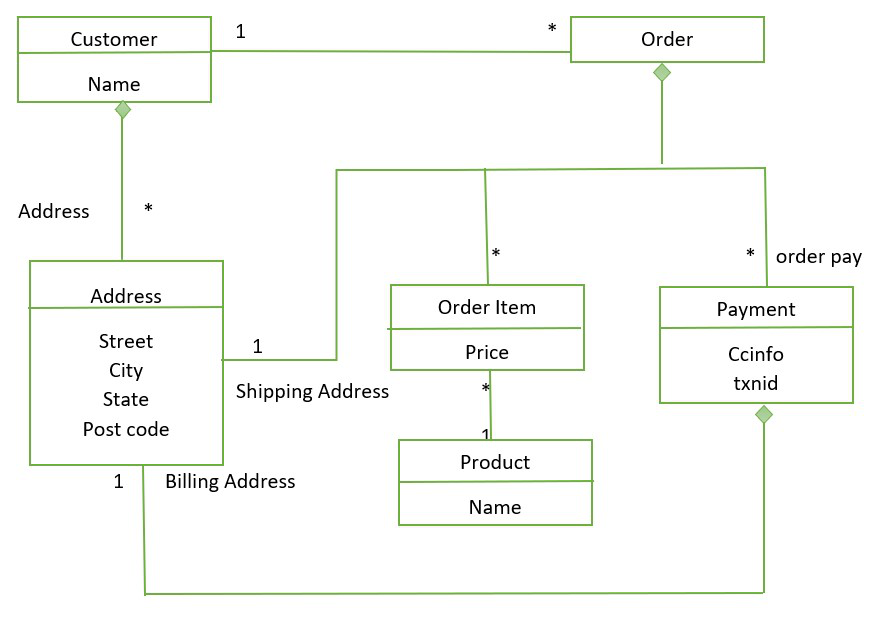
### **Aggregate Data Models:**

The term aggregate means a collection of objects that we use to treat as a unit. An aggregate is a collection of data that we interact with as a unit. These units of data or aggregates form the boundaries for ACID operation.

**Example of Aggregate Data Model:**

Here in the diagram have two Aggregate:

* Customer and Orders link between them represent an aggregate.
* The diamond shows how data fit into the aggregate structure.
* Customer contains a list of billing address
* Payment also contains the billing address
* The address appears three times and it is copied each time
* The domain is fit where we don’t want to change shipping and billing address.

**Advantage:**

* It can be used as a primary data source for online applications.
* Easy Replication.
* No single point Failure.
* It provides fast performance and horizontal Scalability.
* It can handle Structured semi-structured and unstructured data with equal effort.

**Disadvantage:**

* No standard rules.
* Limited query capabilities.
* Doesn’t work well with relational data.
* Not so popular in the enterprise.
* When the value of data increases it is difficult to maintain unique values.

The aggregate-Oriented database is the NoSQL database which does not support ACID transactions and they sacrifice one of the ACID properties. Aggregate orientation operations are different compared to relational database operations. We can perform OLAP operations on the  Aggregate-Oriented database. Aggregate – Oriented databases are classified into four major data models. They are as follows:

* Key-value
* Document
* Column family
* Graph-based

**1.key-value Data Model:**

* A key-value store is a simple and highly performant database model that stores data as a collection of key-value pairs.
* Each key is unique, and it maps directly to a specific value, which can be a string, number, JSON object, or even a complex data structure.
* The primary use case is to quickly retrieve the value using its corresponding key.
  1. **Document Data Model:**
* Document databases store data in the form of documents, typically using JSON, BSON, XML, or similar formats.
* Each document contains key-value pairs but can also have nested structures, making them more complex than key-value stores.

 This model is best suited for hierarchical data, where the document itself can represent a rich structure.

* 1. **Column family Data Model:**

 Column-family stores, sometimes called wide-column stores, organize data into rows and columns, but unlike traditional relational databases, columns are grouped into families, where each family can have a variable number of columns.

 This model is designed for applications where querying a subset of data is common, and it can efficiently store sparse datasets.

**4.Graph Data Model:**

 A graph database is designed to store and navigate relationships between data, representing the data as nodes (entities) and edges (relationships).

 This model excels in scenarios where relationships are as important as the data itself, allowing efficient traversal and querying of relationships between entities.