



| class 1 | class 2 | relationship why |
|----------|--------------|--|
| Order | EBook | Aggregation (1..*) An Order contains one or more EBook instances. Aggregation is used because EBooks exist independently of any particular Order—they are part of the store catalog, not unique to individual orders. |
| Customer | ShoppingCart | Aggregation (1 to 0..1) A Customer can have one ShoppingCart, but they can also exist without one (e.g., if they haven't added any items to the cart yet). This relationship reflects that customers may or may not use the shopping cart, but it is directly associated with the customer. |
| Invoice | Order | Composition (1 to 1) An Invoice is directly dependent on an Order, as each order generates a single invoice. Composition is appropriate here since an Invoice cannot exist without its associated Order. |
| Order | Discount | Association (1 to 0..*) Order is associated with Discount because orders may apply one or more discount types, such as loyalty or bulk purchase discounts. However, Discount is an independent class, not strictly owned by Order. |
| Customer | Invoice | Association (1 to 0..*) Each Customer may receive multiple Invoice instances over time, depending on the number of orders they place. Association is suitable here because invoices are generated for customer orders but are not exclusively bound to a customer. |
| Customer | Order | Association (1 to 0..*) A Customer may place multiple Order instances. Association reflects that while orders are made by customers, they do not possess exclusive ownership over them, allowing each order to stand independently. |