

Entity-relationship diagrams (DDD)

The Advanced Higher course requires candidates to describe, exemplify and implement entity-relationship diagrams with three or more entities, indicating:

- ◆ entity name
- ◆ entity type (strong, weak)
- ◆ attributes
- ◆ relationship participation (mandatory, optional)
- ◆ name of relationship
- ◆ cardinality

Candidates also need to be able to identify relationship participation from an entity-occurrence diagram.

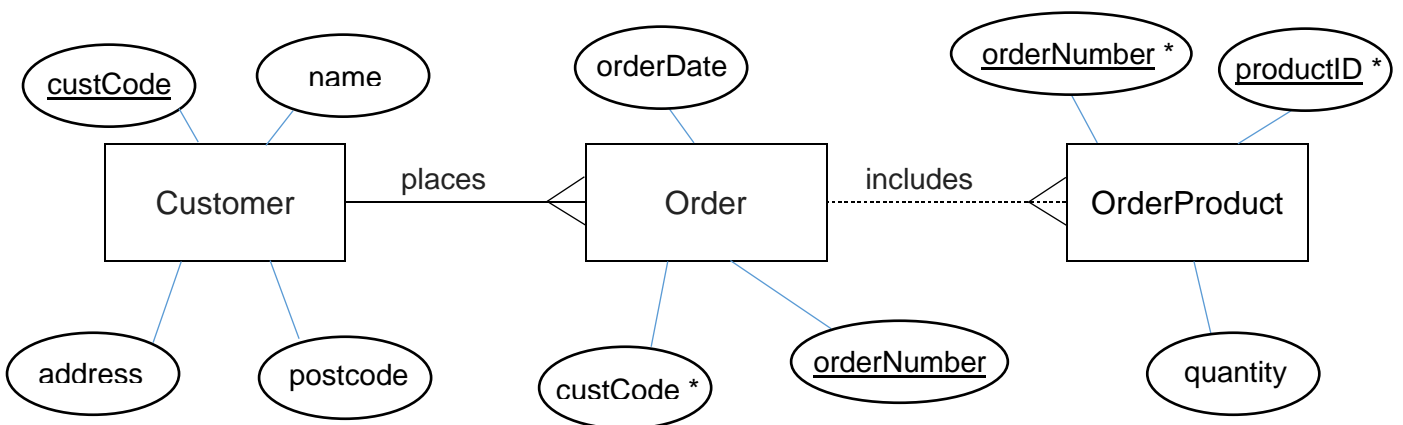
Entity type

A **strong entity** is one whose existence does not depend on the existence of any other entity in the same database. The primary key of a strong entity uniquely identifies each occurrence within the entity.

A **weak entity** is one that depends on one or more strong entities for its existence. For this reason, strong entities are sometimes referred to as owner entities. A weak entity cannot be used independently because its existence depends on one or more owner entities.

The primary key of a weak entity is formed, in part, using the primary key of its owner entity(ies). The presence of a weak entity is indicated by using a dotted line to represent the relationship between the weak entity and its owner entity. The weak entity itself is indicated by using optionality.

Consider the (incomplete) entity-relationship diagram shown below. This illustrates three of the entities that form part of an online ordering system.



In this situation, the Customer and Order entities both have a primary key that uniquely identifies individual occurrences in each entity. For this reason, Customer and Order are **strong entities**.

The primary key of the OrderProduct entity is a compound key that is formed using the primary key of the Order entity. This means that OrderProduct is a **weak entity**. The dotted line is used to represent the weak relationship between the Order and OrderProduct entities.

Relationship participation

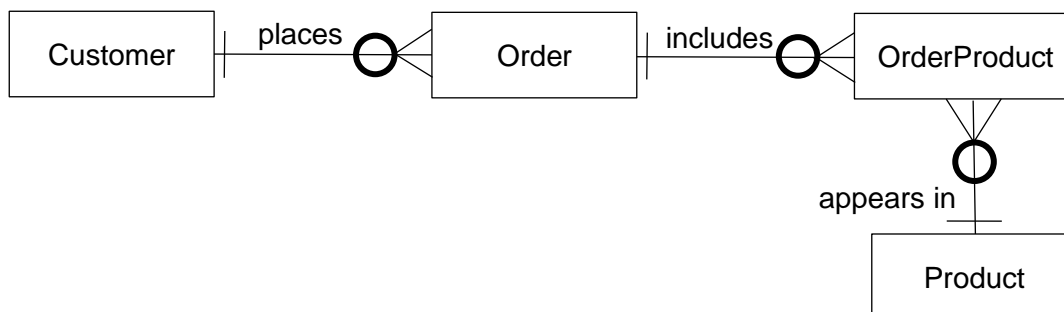
Participation refers to the nature of the relationship between entities. Participation can be either mandatory or optional.

Mandatory participation describes a relationship where at least one occurrence of an entity must exist before any occurrences can be added to its associated entity. The mandatory side of any relationship is indicated by using a vertical line



Optional participation describes a relationship between two entities where it is possible to add occurrences of one entity without the need to have existing occurrences in the associated entity. The optional side of a relationship is indicated by using a bold circle ○

Participation has been added to the entity-relationship diagram introduced earlier and is shown below. For completeness, the Product entity has also been added to show all four entities that form the online ordering system.



Customer and Order are both strong entities, as each has its own uniquely identifying primary key.

The Customer entity is linked to the Order entity using the **places** relationship. Each customer in the Customer entity can place many orders but it is also possible for details of a customer to be stored without them placing any orders.

As each order in the Order entity must always have one set of corresponding customer details in the Customer entity, it is not possible to add a new set of details to the Order entity without first having added details of the relevant customer to the Customer entity.

The Order entity is linked to the weak OrderProduct entity using the **includes** relationship. The entity-relationship diagram shows that a new order can be created without a pre-existing, corresponding occurrence in the OrderProduct entity. Once it has been added to the Order entity, the order can be linked to several occurrences within the OrderProduct entity; it is also possible for an order to have no corresponding OrderProduct occurrences.

As OrderProduct is a weak entity, it is not possible to add an OrderProduct occurrence without first having an existing, corresponding occurrence in the Order entity.

The Product entity is linked to OrderProduct entity using the **appears in** relationship. As Product is a strong entity with its own uniquely identifying primary key, new product details can be added without the need to have any corresponding occurrences in the OrderProduct entity. The entity-relationship diagram shows that each product can appear in many individual OrderProduct occurrences, but it is possible that a product is never ordered.

As OrderProduct is a weak entity, it is not possible to add an OrderProduct occurrence without first having an existing, corresponding occurrence in the Product entity.

Example

A travel agency uses a relational database to store details on a booking system.

It stores details of Scottish holiday resorts, hotels in each resort, customers and their bookings. These details are arranged in four separate entities.

The attributes stored in each entity are shown below.

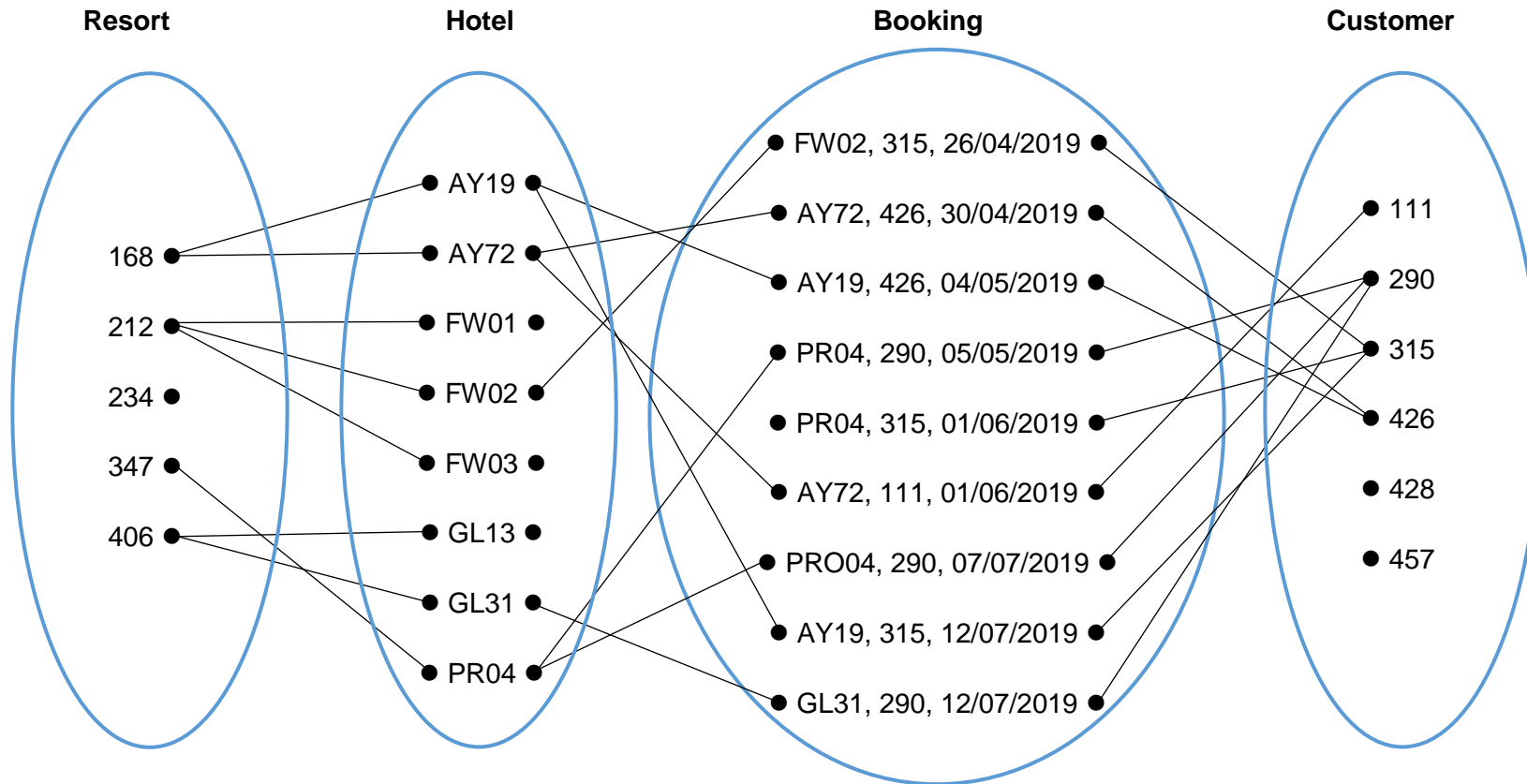
Resort	Hotel	Customer	Booking
<u>resortID</u> resortName resortType	<u>hotelRef</u> hotelName resortID * starRating seasonStartDate mealPlan checkInTime pricePersonNight	<u>customerNo</u> firstname surname address town postcode	<u>hotelRef</u> * <u>customerNo</u> * <u>startDate</u> numberOfNights numberInParty

Strong and weak entities

From the list of attributes, we can see that Resort, Hotel and Customer are all **strong entities** while Booking is a **weak entity**.

Relationship participation

An entity-occurrence diagram indicating the relationships between the entities is shown below.



Using an entity-occurrence diagram helps to clarify the nature of each relationship.

The entity-occurrence diagram for the travel agency booking system makes it clear that:

- ◆ Resort: Hotel is a 1: M relationship:
 - Resort has **optional** participation in this relationship:
 - an individual resort may have one or more hotels
 - it is possible that a resort has no hotels at all
 - Hotel has **mandatory** participation in this relationship:
 - every hotel is located in exactly one resort

- ◆ Hotel: Booking is a 1: M relationship:
 - Hotel has **optional** participation in this relationship:
 - an individual hotel may feature in one or more bookings
 - it is possible for a hotel to have no bookings
 - Booking has **mandatory** participation in this relationship:
 - each booking must be associated with exactly one hotel

- ◆ Booking to Customer is a 1: M relationship:
 - Booking has **mandatory** participation in this relationship:
 - each booking is made by exactly one customer
 - Customer has **optional** participation:
 - an individual customer may make one or more bookings
 - it is possible that some customers never make a booking (for example, details of customers on the mailing list will be stored in the database, even though they have never made any bookings)

Entity-relationship diagram

The complete entity-relationship diagram that represents the relationships between the entities and relationships in the travel agency booking system is shown below.

