The diagram on page two includes one *one-to-many* relationship and two *many-to-many* relationships that are broken down through the use of two linking tables. Conversely, a participant may book a place on a number of guided walks, but they can attend only one walk at a given time. Walks take place on set dates at different times and any number of guides can lead the same walk across different time slots.

A *Walk* table records every walk on offer with a primary key walk\_id. This helps keep track of walks that are not part of the company’s current programme. A walk may appear in many programmes, but a programme can include multiple instances of a walk. A *Walkprogramme* table with a unique walkprogramme\_id attribute stores this information. The *one-to-many* relationship integrity between the tables is guaranteed by the addition of a foreign key walkprogramme\_walk in the *Walkprogramme* table (the ‘many’ side of the relationship). It references the primary key walk\_id in the *Walk* table (the ‘one’ side of the relationship). The duration of a walk is then considered so that dates and times are allotted for every instance of a walk.

A separate *Guide* entity with a unique guide\_id attribute holds the details of guides. It can be used to retrieve the details of those who are not leading a walk. A guide in charge of a walk should be contacted in case of emergencies so the guide\_phone field is set to NOT NULL. Since a number of guides can lead any walk, a list of those who do lead walks would be handy. That list could also link the *Walkprogramme* and *Guide* tables.

The linking *Guidedwalk* entity participates totally in the *many-to-one* relationship it has with each table, while a *many-to-many* relationship is afforded for the *Walkprogramme* and *Guide* tables. The linking entity is identified by the foreign keys walkprogramme\_id (reference to the programme identificators in the *Walkprogramme* table) and guide\_id (reference to the primary key of the *Guide* table), both generating a so-called composite key.

Each participant is uniquely identified by a participant\_id in the *Participant* entity. Participants pay either a full fee of £8.00 or a concessionary fee of £6.00 so their payment status needs to be known. The participant\_payment column confirms this information. A participant may place a number of bookings, while a booking should be associated with only one participant.

However, bookings have a meaning in the context of a participant attending a walk programme, therefore a *Booking* table could be a linking entity for the *Walkprogramme* and *Participant* tables. It adopts the participant\_id and walkprogramme\_id fields of the latter, both combining to form a composite primary key. This ensures that each participant is only booked once on a given walk.