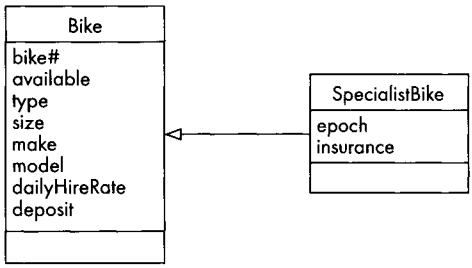

Figure 9.18 *Inheritance relationship between the Bike and SpecialistBike classes*

Inheritance. Figure 9.18 shows the inheritance relationship between the Bike and SpecialistBike classes in the Wheels system.

There are three different ways of implementing an inheritance relationship, and each situation has to be considered carefully in order to select the most suitable approach. This example from the Wheels system is a very simple relationship; there is only one subclass and it only adds two new attributes, so in this case we would implement the relationship in a single table as shown in Figure 9.19. The bottom record in the table is for a specialist bike and so fills all the fields in the table. The disadvantage of implementing an inheritance relationship in this way is that it may result in a lot of null values, particularly if there are many fields that are only relevant to the subclass, or (as here) when there are comparatively few instances of the subclass.

The decision as to how to implement an inheritance relationship is more complicated in the case where there are two or more subclasses. It is possible to create tables for the subclasses only; this option is suitable in the case where the superclass is abstract (i.e. there are no actual instances of it) or where very few of the attributes are shared and appear in the superclass. The disadvantage is that the

Bike

Bike No.	Available	Type	Size	Make	Model	Epoch	Insur-ance	Daily hire rate	Deposit
249	On hire	mountain	woman's	Scott	Atlantic Trail			£80.00	£50.00
250	Available	tourer	man's	Raleigh	Pioneer			£9.00	£60.00
251	On hire	mountain	woman's	Scott	Atlantic Trail			£8.00	£50.00
252	On hire	tourer	man's	Dawes	Galaxy			£8.00	£50.00
253	Available	mountain	child's	Raleigh	Chopper			£5.00	£25.00
254	Available	tandem	man's	Sunbeam	Voyager	1930s	£15.00	£20.00	£100.00

Figure 9.19 *Inheritance relationship between the Bike and SpecialistBike classes implemented as a single table*