

[Next](#) [Previous](#) [Index](#)

3. Basic Collection (one-to-many)

Definition:

A one-to-many reference is basically a collection. Here a first-rank class, A, holds a reference to a collection of another first-rank class, B.

Scenario:

We have two first-rank classes, Foo and Bar which are related to each other as follows:

```
Set Foo.getBars() // returns a collection of Bar instances
```

Hibernate Mapping:

In Hibernate, this could be mapped as follows:

```
<class name="Foo" table="foo">
    ...
    <set role="bars" table="bar">
        <key column="foo_id"/>
        <one-to-many class="Bar"/>
    </set>
</class>
```

NB: We will mostly use `<set>` type collections in these examples except where we are examining specific collection-type features. Please refer to the hibernate documentation to learn about the comprehensive set of collection types available.

Table Schema:

Foo
id

Bar
id foo_id

Now we have created an extra column in Bar's table which holds the FK to Foo. This allows Foo to be assigned a collection of Bars based on the value of the foo_id column in Bar.

Bidirectionality:

This relationship can be declared both ways, with Bar having getFoo(), by suitable code changes to Bar and the following schema change:

```
<class name="Bar" table="bar">
    ...
    <many-to-one name="foo" class="Foo" column="foo_id"/>
</class>
```

Now your Bars will know who their Foo is. NB: No extra columns are generated for the bidirectionality.

[Next](#) [Previous](#) [Index](#)