

Typos

The original content of this wiki page has been copied from the [PhoneBook tutorial](#). It might or might not be in sync with it at the time you are reading this.

If you get the specification for the property at the other side of a relation wrong, the compiler will not warn you, but `dbschema.exe` will complain with this error message:

```
Execution aborted. Exception stack:
Remotion.Data.DomainObjects.Mapping.MappingException: Opposite relation
property 'PhoneNumber' could not be found on type
'PhoneBook.Domain.Person'.
Declaring type: PhoneBook.Domain.PhoneNumber, property: Person
```

This happens if you write, for example, "PhoneNumber" instead of "PhoneNumbers" in the `DBBidirectionalAttribute` parameter for the opposite property:

```
// Wrong: "PhoneNumber" instead of "PhoneNumbers" -- typo goes
// undetected by compiler
[DBBidirectionalRelation("PhoneNumber")]
public virtual Person Person { get; set; }
// -> Missing DBBidirectionalRelation attribute error
```

Mind the SQL!

If you get SQL-exceptions, you might frown upon re-motion and think that it is a bug in re-motion (or re-store, to be more precise). After all, re-store is supposed to hide all those SQL details from you, right? So if there is an SQL-problem, it can't possibly be your problem, right? Almost. The spot where you DO work with SQL is the `SortExpression` parameter in a `DBBidirectionalRelation` attribute. This is an SQL-fragment (for an `ORDER BY` clause) and is copied into the schema on an as-is basis. If you misspell a property name, the database will throw an exception at you as soon as it tries to evaluate that expression. So if you get an "RdbmsProviderException was unhandled/Error while executing SQL command", check the spelling of your `SortExpression` (the same thing applies to the `ContainsKey` parameter, explained in 1:1 bidirectional relationships).