

Spring Professional Exam Tutorial v5.0

Question 07

Question 07 - How does the JdbcTemplate support generic queries? How does it return objects and lists/maps of objects?

Jdbc Template supports generic queries with following methods:

- ▶ `queryForObject` - returns single object, expects query to return only one record, if this requirement is not matched `IncorrectResultSizeDataAccessException` will be thrown
- ▶ `queryForList` - returns list of objects of declared type, expects query to return results with only one column, otherwise `IncorrectResultSetColumnCountException` will be thrown
- ▶ `queryForMap` - returns map for single row with keys representing column names and values representing database record value, expects query to return only one record, if this requirement is not matched `IncorrectResultSizeDataAccessException` will be thrown
- ▶ `queryForRowSet` - returns `SqlRowSet` object that contains metadata information (like column names) and allows to read results data and iterate through records

All of the methods above have many versions, allowing you to specify not only query itself, but also parameters to the query and customer row mapper if required.

Question 07 - How does the JdbcTemplate support generic queries? How does it return objects and lists/maps of objects?

Jdbc Template returns objects, lists/map by using following:

- ▶ **objects** - `queryForObject` - `SingleColumnRowMapper` for generic types and `RowMapper` for custom types
- ▶ **lists** - `queryForList` - `SingleColumnRowMapper` for generic types
- ▶ **maps** - `queryForMap` - `ColumnMapRowMapper` for any query

