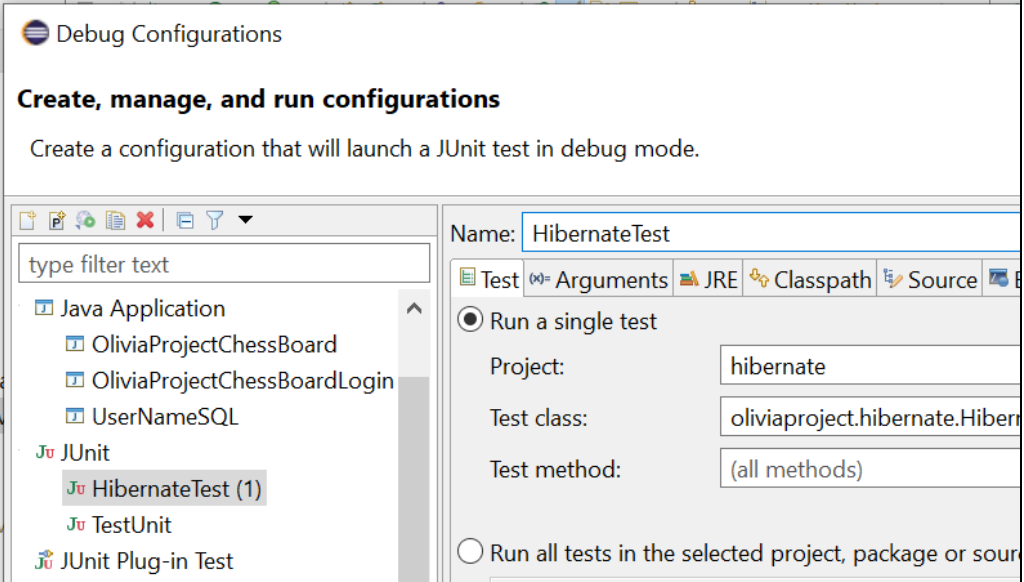


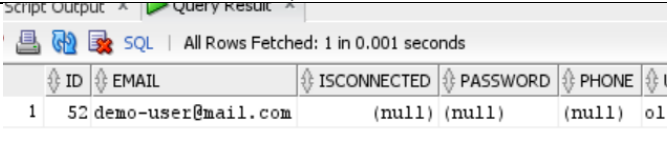
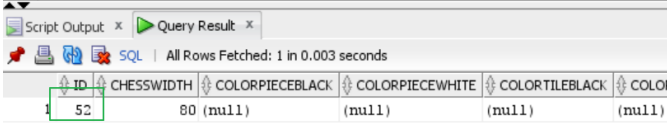
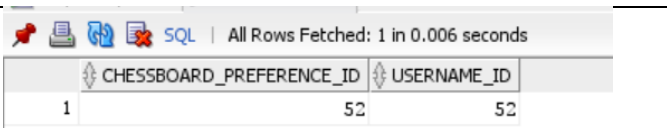
OneToOne with foreign key on owner's table

| N | Steps | Description |
|---|-----------------------------------|--|
| 0 | | |
| 1 | Database | I used an oracle database and provided the configuration on the file in src>main>resources> hibernate.cfg.xml |
| 2 | Dependencie | You need to set in your PATH C:\tools\gradle-7.5.1 More: https://docs.gradle.org/7.5.1/release-notes.html |
| 3 | Java | C:\Program Files\Java\jdk-18.0.2 This is because I played with hibernate-core and hibernate-common-annotations // https://mavenlibs.com/maven/pom/org.hibernate/hibernate-core implementation('org.hibernate:hibernate-core:6.1.2.Final') // https://mvnrepository.com/artifact/org.hibernate.common/hibernate-commons-annotations |
| 4 | Eclipse project | Run gradle.bat eclipse. The dependencies should get sorted, except for ojdbc8.jar oracle library that is referred in build.gradle: <code>implementation(files("C:\\thirdparty\\ojdbc8.jar"))</code> |
| 5 | Launch the junit test via eclipse | Using 1 run configuration: oliviaproject.hibernate.foreignkey.HibernateForeignKeyTest  |

| | | <div><div>ps-code-analysis - hibernate/src/test/java/oliviaproject/hibernate/HibernateTest.java - Eclipse SDK</div><div>le Edit Source Refactor Navigate Search Project Run Window Help</div><div><div>Package Explorer</div><div>Type Hierarchy</div><div>JUnit ×</div></div><div>ished after 7.498 seconds</div><div>uns: 1/1Errors: 0Failures: 0</div><div>oliviaproject.hibernate.HibernateTest [Runner: JUnit 4] (7.467 s)</div><div><div>HibernateTest.java</div><div>15</div><div>16private</div><div>17</div><div>18</div><div>19public</div><div>20try</div><div>21</div><div>22</div><div>23</div><div>24</div><div>25</div></div></div> | | | | | | | | | | | | | | | | | | | | | | | | |
|----|------------------------------|---|-----------------|----------------|-------------|-------------------------|-------|----------|-------------------------|---|-----------------------|--------|--------|--------|--------|----|----|-----------|-----------------|-----------------|----------------|---|----|-----------|--------|--------|
| 6 | Result | <div><div><div>select * from foreignkeyusername;</div><div>select * from foreignkeychessboardpreference;</div></div><div>The foreign key is on username</div><div><div>Script Output ×Query Result ×</div><div>SQL All Rows Fetched: 1 in 0.008 seconds</div><table><tr><th>ID</th><th>EMAIL</th><th>ISCONNECTED</th><th>PASSWORD</th><th>PHONE</th><th>USERNAME</th><th>CHESBOARD_PREFERENCE_ID</th></tr><tr><td>1</td><td>52 demo-user@mail.com</td><td>(null)</td><td>(null)</td><td>(null)</td><td>olivia</td><td>52</td></tr></table></div><div><div>select * from foreignkeychessboardpreference;</div><div><div>Script Output ×Query Result ×</div><div>SQL All Rows Fetched: 1 in 0.003 seconds</div><table><tr><th>ID</th><th>CHESWIDTH</th><th>COLORPIECEBLACK</th><th>COLORPIECEWHITE</th><th>COLORTILEBLACK</th></tr><tr><td>1</td><td>52</td><td>80 (null)</td><td>(null)</td><td>(null)</td></tr></table></div></div></div> | ID | EMAIL | ISCONNECTED | PASSWORD | PHONE | USERNAME | CHESBOARD_PREFERENCE_ID | 1 | 52 demo-user@mail.com | (null) | (null) | (null) | olivia | 52 | ID | CHESWIDTH | COLORPIECEBLACK | COLORPIECEWHITE | COLORTILEBLACK | 1 | 52 | 80 (null) | (null) | (null) |
| ID | EMAIL | ISCONNECTED | PASSWORD | PHONE | USERNAME | CHESBOARD_PREFERENCE_ID | | | | | | | | | | | | | | | | | | | | |
| 1 | 52 demo-user@mail.com | (null) | (null) | (null) | olivia | 52 | | | | | | | | | | | | | | | | | | | | |
| ID | CHESWIDTH | COLORPIECEBLACK | COLORPIECEWHITE | COLORTILEBLACK | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 52 | 80 (null) | (null) | (null) | | | | | | | | | | | | | | | | | | | | | | |
| 7 | You may need to reinitialize | <div>As I check for unicity of username.</div> <div>drop table foreignkeyusername cascade constraints; drop table foreignkeychessboardpreference cascade constraints;</div> <div>see database-schema.sql</div> | | | | | | | | | | | | | | | | | | | | | | | | |

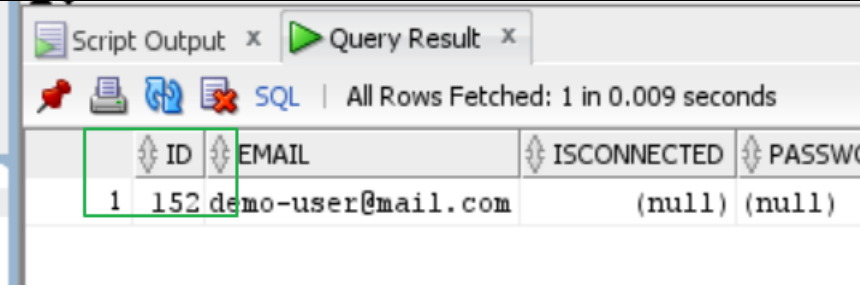
OneToOne with Jointable





| | | |
|--------|----------|--|
| N 0 | Steps | Description |
| 1 | Run with | oliviaproject.hibernate.jointable.HibernateJointableTest |

| | | |
|---|---|--|
| 2 | You may need to reinitialize | Reinitialisation with @OneToOne jointable: drop table jointableusername cascade constraints; drop table jointablechessboardpreference cascade constraints; drop table USERNAME_CHESSBOARDPREFERENCE cascade constraints; see database-schema.sql |
| 3 | jointableusername |  |
| 4 | jointablechessboardpreference |  |
| 5 | JOINTABLE_USERNAME_CHESSBOARDPREFERENCE |  So the join table makes the links between the 2 other tables. |

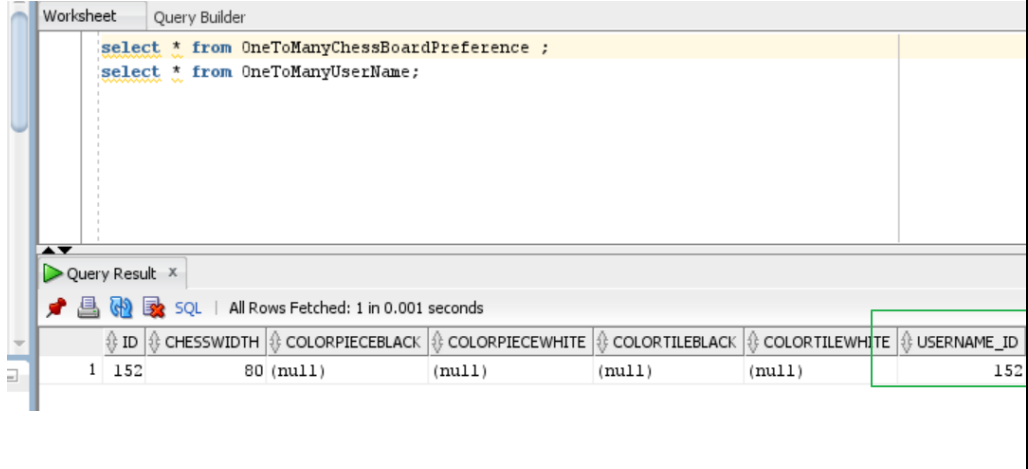
OneToOne Using a Shared Primary Key

The third technique is something new that uses a shared key in both table

| N | Steps | Description |
|---|---------------------|--|
| 0 | | |
| 1 | Debug configuration | HibernateSharedKeyTest |
| 2 | Reinitialization | drop table sharedkeyusername cascade constraints; drop table sharedkeychessboardpreference cascade constraints; |
| 3 | sharedkeyusername |  |

| 4 | sharedkeychessboard preference | <div><div>Script Output x</div><div>Query Result x</div></div> <div><div></div><div>SQL All Rows Fetched: 1 in 0.002 seconds</div><table><tr><th>ID</th><th>CHESSWIDTH</th><th>COLORPIECEBLACK</th><th>COLORPIECEWHITE</th><th>COLORPIECEBLACK</th></tr><tr><td>1 152</td><td>80</td><td>(null)</td><td>(null)</td><td>(null)</td></tr></table></div> | ID | CHESSWIDTH | COLORPIECEBLACK | COLORPIECEWHITE | COLORPIECEBLACK | 1 152 | 80 | (null) | (null) | (null) |
|-------|-----------------------------------|---|-----------------|-----------------|-----------------|-----------------|-----------------|-------|----|--------|--------|--------|
| ID | CHESSWIDTH | COLORPIECEBLACK | COLORPIECEWHITE | COLORPIECEBLACK | | | | | | | | |
| 1 152 | 80 | (null) | (null) | (null) | | | | | | | | |

OneToMany with foreign key

| N 0 | Steps | Description | | | | | | | | | | | | | | |
|-------|--------------|---|-----------------|----------------|-----------------|-----------------|----------------|----------------|-------------|-------|----|--------|--------|--------|--------|-----|
| 1 | Run with | HibernateOneToManyForeignKeyTest | | | | | | | | | | | | | | |
| 2 | Reinitialize | drop table OneToManyChessBoardPreference cascade constraints; drop table OneToManyUserName cascade constraints; | | | | | | | | | | | | | | |
| 3 | Result | <p>The foreign key is on the slave table.</p>  <p>Worksheet Query Builder</p> <pre>select * from OneToManyChessBoardPreference ; select * from OneToManyUserName;</pre> <p>Query Result x</p> <p>SQL All Rows Fetched: 1 in 0.001 seconds</p> <table><tr><th>ID</th><th>CHESSWIDTH</th><th>COLORPIECEBLACK</th><th>COLORPIECEWHITE</th><th>COLORTILEBLACK</th><th>COLORTILEWHITE</th><th>USERNAME_ID</th></tr><tr><td>1 152</td><td>80</td><td>(null)</td><td>(null)</td><td>(null)</td><td>(null)</td><td>152</td></tr></table> | ID | CHESSWIDTH | COLORPIECEBLACK | COLORPIECEWHITE | COLORTILEBLACK | COLORTILEWHITE | USERNAME_ID | 1 152 | 80 | (null) | (null) | (null) | (null) | 152 |
| ID | CHESSWIDTH | COLORPIECEBLACK | COLORPIECEWHITE | COLORTILEBLACK | COLORTILEWHITE | USERNAME_ID | | | | | | | | | | |
| 1 152 | 80 | (null) | (null) | (null) | (null) | 152 | | | | | | | | | | |

OneToMany with manytoone: best setup

| N0 | Steps | Description |
|----|--------------|--|
| 1 | Run with | HibernateOneToManyBiDirectionalTest |
| 2 | Reinitialize | drop table OneToManyBiDirChessBoardPreference cascade constraints; drop table OneToManyBiDirUserName cascade constraints; |
| 3 | Result | The foreign key is on the slave table. |

Usage of named queries

| N0 | Steps | Description |
|----|-------------|---|
| 1 | Test junit | HibernateQueryUserByNameAndPasswordTest |
| 2 | Explanation | <p>I added the namedquery in an annotation.</p> <pre>@Entity @org.hibernate.annotations.NamedQuery(name = "checkCredentials", query = "from NamedQueryUserName where userName = :userName and password =:password") public class NamedQueryUserName {</pre> |
| 3 | Junit | <pre>public static void testQueryUserByNamedQuery() { Session session=factory.openSession(); Query query = session.createNamedQuery("checkCredentials", NamedQueryUserName.class); query.setParameter("userName", "olivia"); query.setParameter("password", "olivia"); NamedQueryUserName user = (NamedQueryUserName) query.getSingleResult(); if(user!=null) { System.out.println("username and password are valid"); }else { System.out.println("username and password are not valid"); } session.close(); }</pre> |