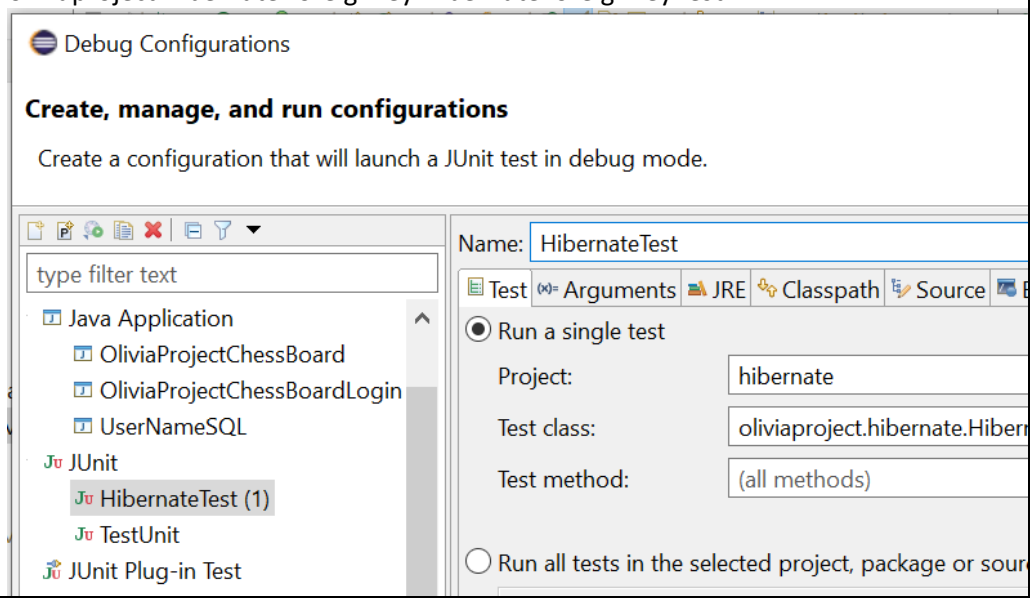
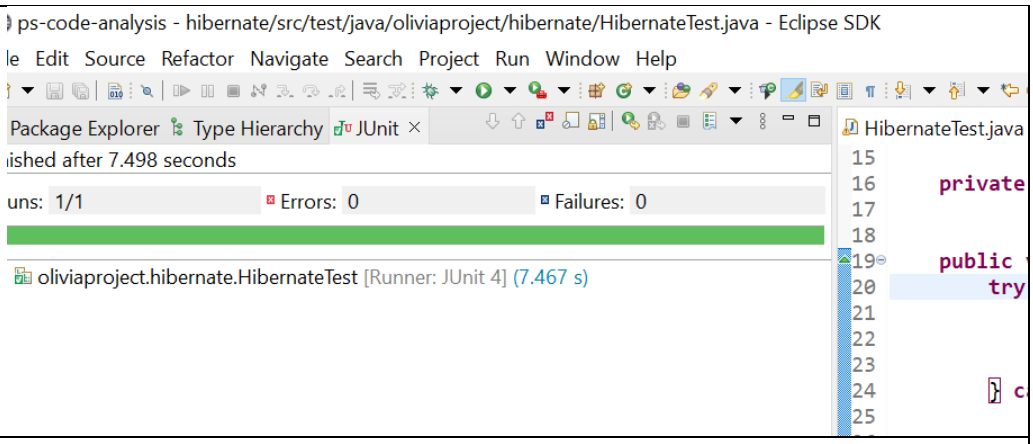
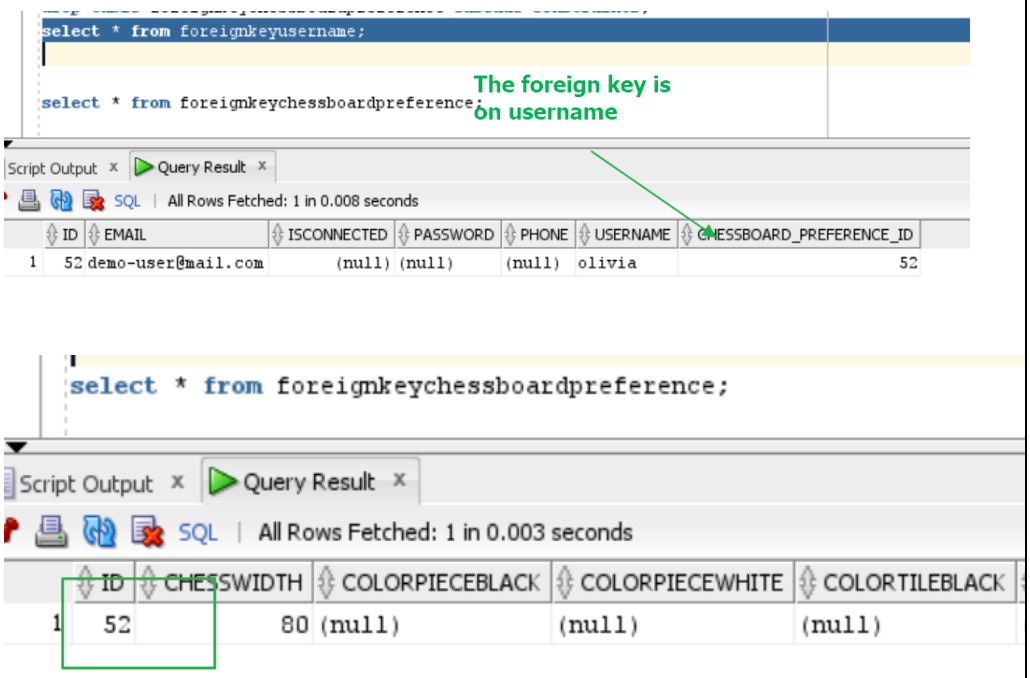
OneToOne with foreign key on owner's table

N0	Steps	Description
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: `implementation( files("C:\\thirdparty\\ojdbc8.jar"))`
5	Launch the junit test via eclipse	Using 1 run configuration: oliviaproject.hibernate.foreignkey.HibernateForeignKeyTest 

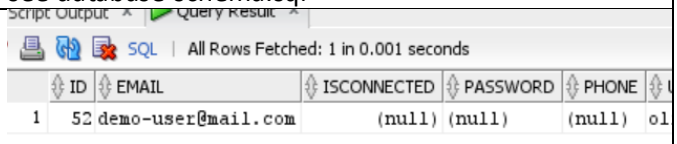
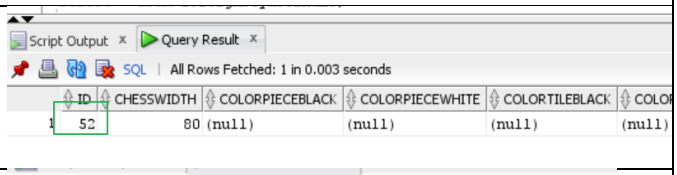
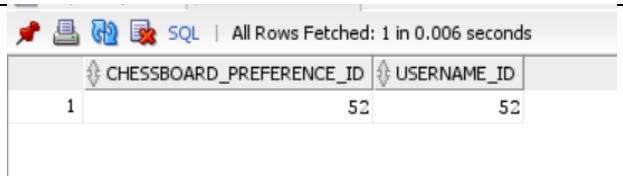
		
6	Result	

7	You may need to reinitialize	As I check for unicity of username.
		
		`drop table foreignkeyusername cascade constraints;`
		`drop table foreignkeychessboardpreference cascade constraints;`
		
		see database-schema.sql

OneToOne with Jointable

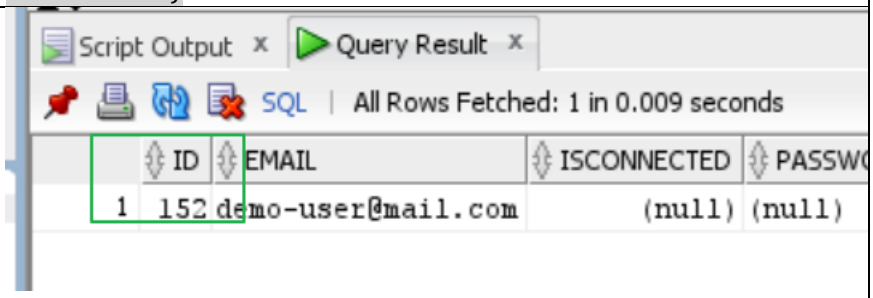
N 0	Steps	Description
1	Run with	`oliviaproject.hibernate.jointable.HibernateJoinTableTest`

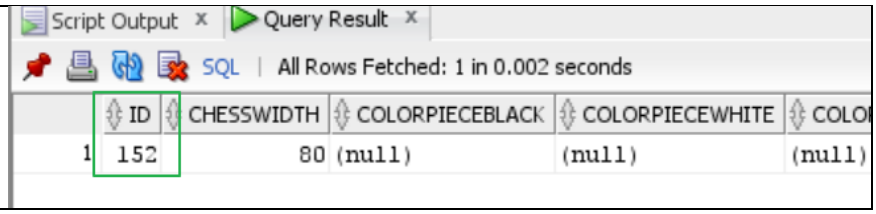
2	You may need to reinitialize	`Reinitialisation with @OneToOne jointable:`
		`drop table jointableusername cascade constraints;`
		`drop table jointablechessboardpreference cascade constraints;`
		`drop table USERNAME_CHESSBOARDPREFERENCE cascade constraints;`
		see database-schema.sql

3	jointableusername	
4	jointablechessboardpreference	
5	JOINTABLE_USERNAME_CHESSBOARDPREFERENCE	 So the join table makes the links between the 2 other tables.

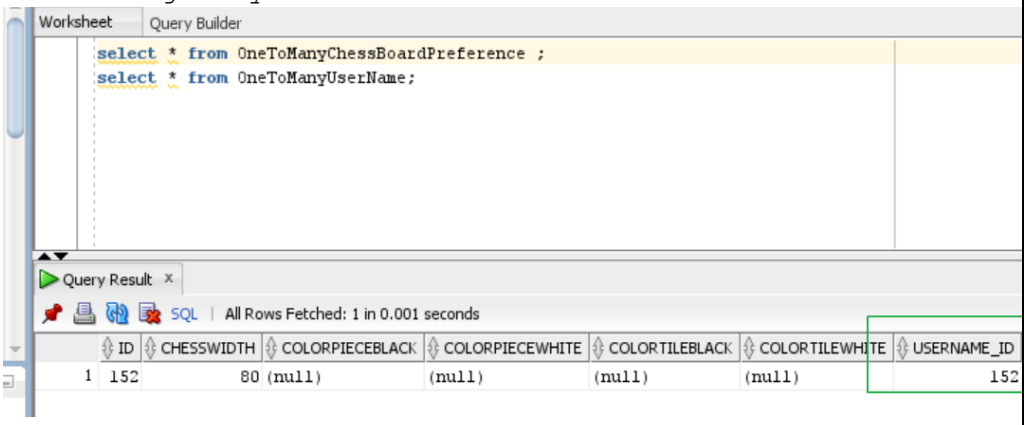
OneToOne Using a Shared Primary Key

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

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

4	sharedkeychessboard preference	

OneToMany with foreign key

N0	Steps	Description
1	Run with	HibernateOneToManyForeignKeyTest
2	Reinitialize	drop table OneToManyChessBoardPreference cascade constraints; drop table OneToManyUserName cascade constraints;
3	Result	The foreign key is on the slave table. 

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	I added the namedquery in an annotation.
		`@Entity`
		`@org.hibernate.annotations.NamedQuery(name = "checkCredentials",`
		`query = "from NamedQueryUserName where userName = :userName and password =:password")`
		`public class NamedQueryUserName {`

3	Junit	`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();`