



HOME

This tool will take the DDL that you enter to create your tables, including your foreign key constraints, and create a Hibernate project with all the Entities, and Sample Application Drivers for each Entity.

1. **There are NO refunds!**

2. This is a tool only, and meant to be informative and educational only. There are usually no syntax errors, but due to the multitude of possible inputs, we can't guarantee no syntax errors.

3. Click here to see [sample](#) DDL code that you'd enter to create your project.


4. You will have working examples of how to create, update, query and delete entities.

5. First you must pay \$20 to run this code-generator and then you can input your DDL code to generate your hibernate JPA project.

I'm happy to answer questions/concerns. Please use the "[Contact US](#)" page for best results.




 **PayPal CREDIT**

 Debit or Credit Card

Powered by  **PayPal**

Log in to your PayPal account - Google Chrome

sandbox.paypal.com/checkoutnow?sessionID=90362e65e5_mdu6mdy6...



Pay with PayPal

With a PayPal account, you're eligible for free return shipping, Purchase Protection, and more.

 Stay logged in for faster purchases [?](#)

Log In

[Having trouble logging in?](#)

or

Pay with Debit or Credit Card

Quality Software Engineering, Consulting (LLC)

n key
ion Drivers for

sually no syntax
ors.

es.
DL code to

results.



Welcome to the Hibernate Code Generator

Here you will enter your create table DDL, including your foreign key constraints to create a complete and functional JAVA JPA layer.

1. THANK YOU FOR YOUR PAYMENT.

Your transaction has been completed, and a receipt for your purchase has been emailed to you. Log into your PayPal account to view transaction details.

2. Make sure you enter DDL (create table statements) for MYSQL.

3. To help ensure success, make sure DDL loads successfully in your own instance of MYSQL Version 5.7.30

4. Make sure the lines are semicolon delimited.

5. Only use int PRIMARY KEY and with auto_increment. It can generate code without the "Auto_increment", but you may have to make some modifications.

6. Click here to see [sample](#) DDL code.

7. I'm happy to answer questions/concerns. Please use the "[Contact US](#)" page for best results.

Enter you DDL here:

SUBMIT

CLEAR

Welcome to the Hibernate Code Generator

Here you will enter your create table DDL, including your foreign key constraints to create a complete and functional JAVA JPA layer.

1. THANK YOU FOR YOUR PAYMENT.

Your transaction has been completed, and a receipt for your purchase has been emailed to you. Log into your PayPal account to view transaction details.

2. Make sure you enter DDL (create table statements) for MYSQL.

3. To help ensure success, make sure DDL loads successfully in your own instance of MYSQL Version 5.7.30

4. Make sure the lines are semicolon delimited.

5. Only use int PRIMARY KEY and with auto_increment. It can generate code without the "Auto_increment", but you may have to make some modifications.

6. Click here to see [sample](#) DDL code.

7. I'm happy to answer questions/concerns. Please use the "[Contact US](#)" page for best results.

Enter you DDL here:

```
CREATE TABLE PERSONS (  
  ID      INT NOT NULL AUTO_INCREMENT,  
  LASTNAME  VARCHAR(255),  
  FIRSTNAME VARCHAR(255),  
  DOB     DATE,  
  PRIMARY KEY (ID)  
);  
CREATE TABLE CARS (  
  ID      INT NOT NULL AUTO_INCREMENT,  
  MODEL  VARCHAR(255),  
  MAKE   VARCHAR(255),  
  LMTS   TIMESTAMP ,  
  PRIMARY KEY (ID)  
);  
CREATE TABLE OWNERSHIP (  
  ID      INT NOT NULL AUTO_INCREMENT,  
  P_ID    INT,  
  C_ID    INT,  
  PURCHASEDT DATE,  
  PRIMARY KEY (ID)  
);
```

SUBMIT

CLEAR

Mon Jun 01 05:34:06 UTC 2020

RESULTS:

DDL command	STATUS
create table persons (id int not null auto_increment, lastname varchar(255), firstname varchar(255), dob date, primary key (id))	OK
create table cars (id int not null auto_increment, model varchar(255), make varchar(255), lmts timestamp , primary key (id))	OK
create table ownership (id int not null auto_increment, p_id int, c_id int, purchasedt date, primary key (id))	OK
alter table ownership add constraint fk_pid foreign key (p_id) references persons (id)	OK
alter table ownership add constraint fk_cid foreign key (c_id) references cars (id)	OK

[CLICK HERE](#)

to download your project

INSTRUCTIONS TO INSTALL IN YOUR ECLIPSE IDE:

1) copy to target dir

2) UNZIP it - must have "zip" command

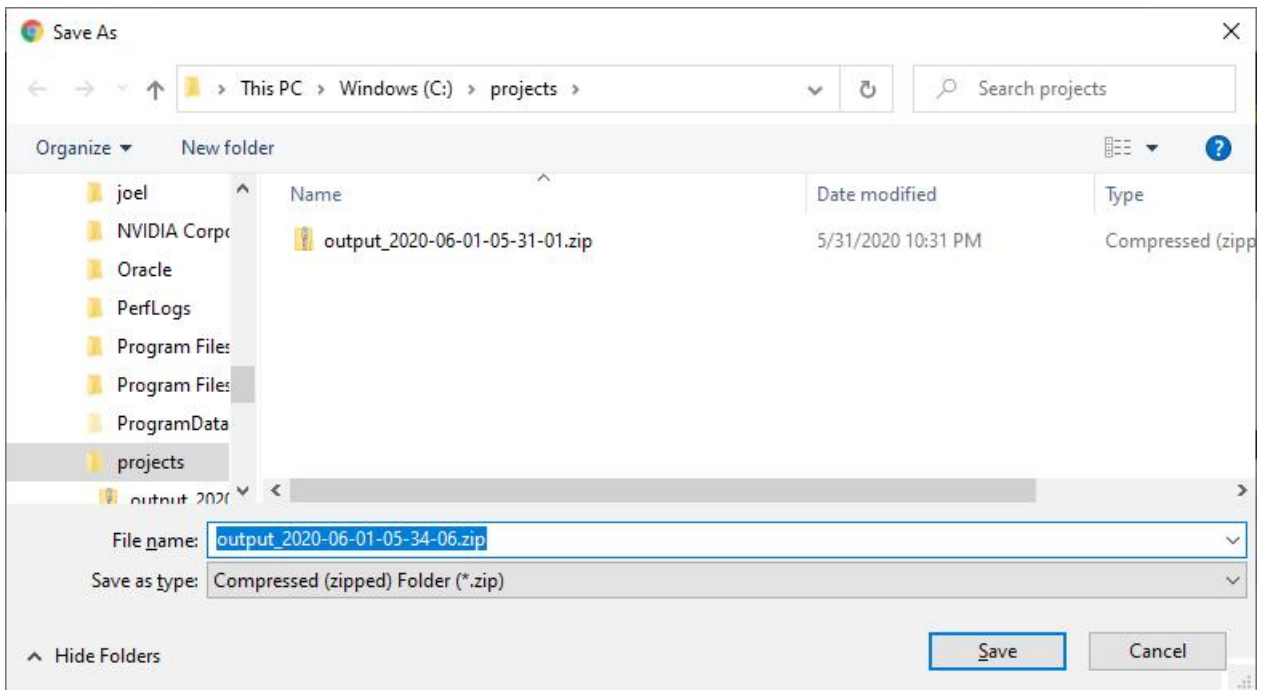
3) in eclipse : File->Import; type "maven" in filter; select "Existing Maven Project" select "output" folder ,
ie. c:\projects\output_2020-05-31-17-04-51\output

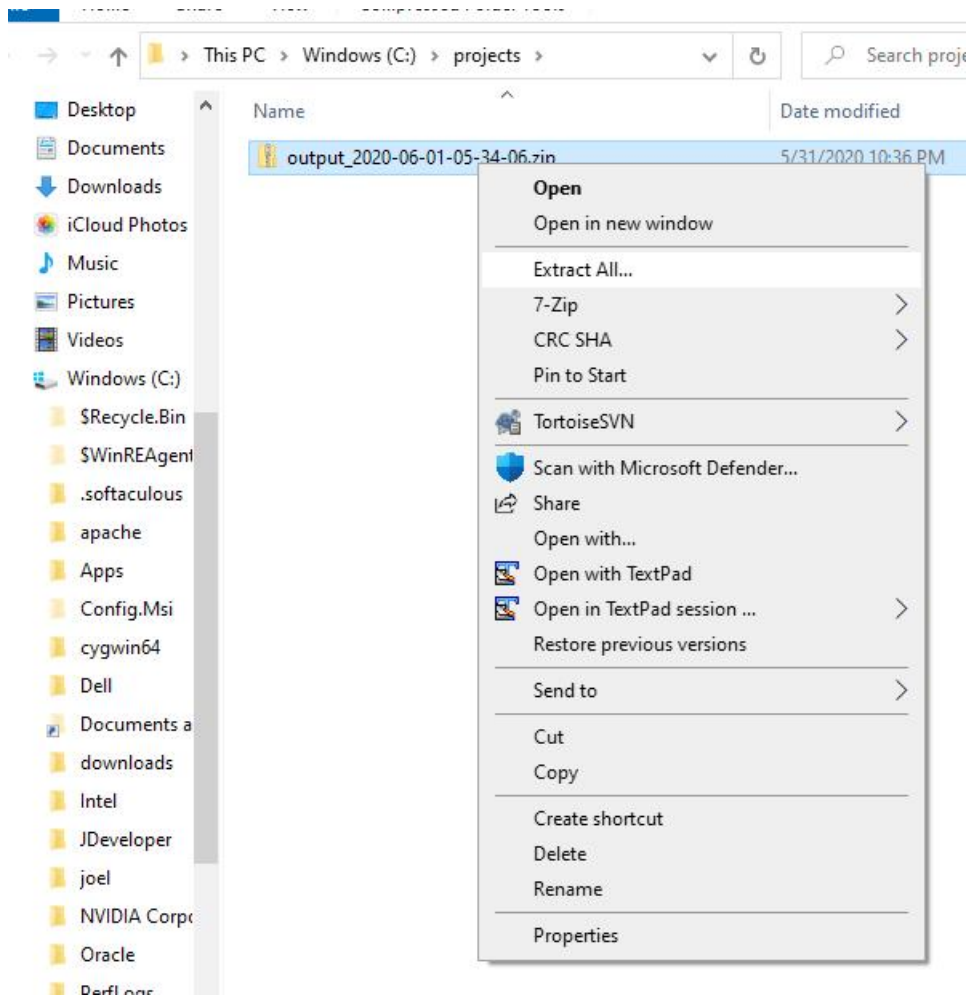
4) select the "pom.xml"

5) if you've done this before, then the "rsigen" will be conflicting name with the pom.xml get , you need to type "rsigen2" (or other #) in the "Name template" (under Advanced section)


6) open project; and navigate to your "generated" package. There you will see "Entities" and a sample Application command line driver.

7) under the /rsigen/src/main/resources directory you'll see hibernate.cfg.xml - you need to modify this to setup your MYSQL connectivity in order to run your Application. 8) you can right-click one of the App.java files and select "run" or "debug" and it will run against your MYSQL database. (Make sure your tables are installed)







←  Extract Compressed (Zipped) Folders

Select a Destination and Extract Files

Files will be extracted to this folder:

C:\projects\output_2020-06-01-05-34-06

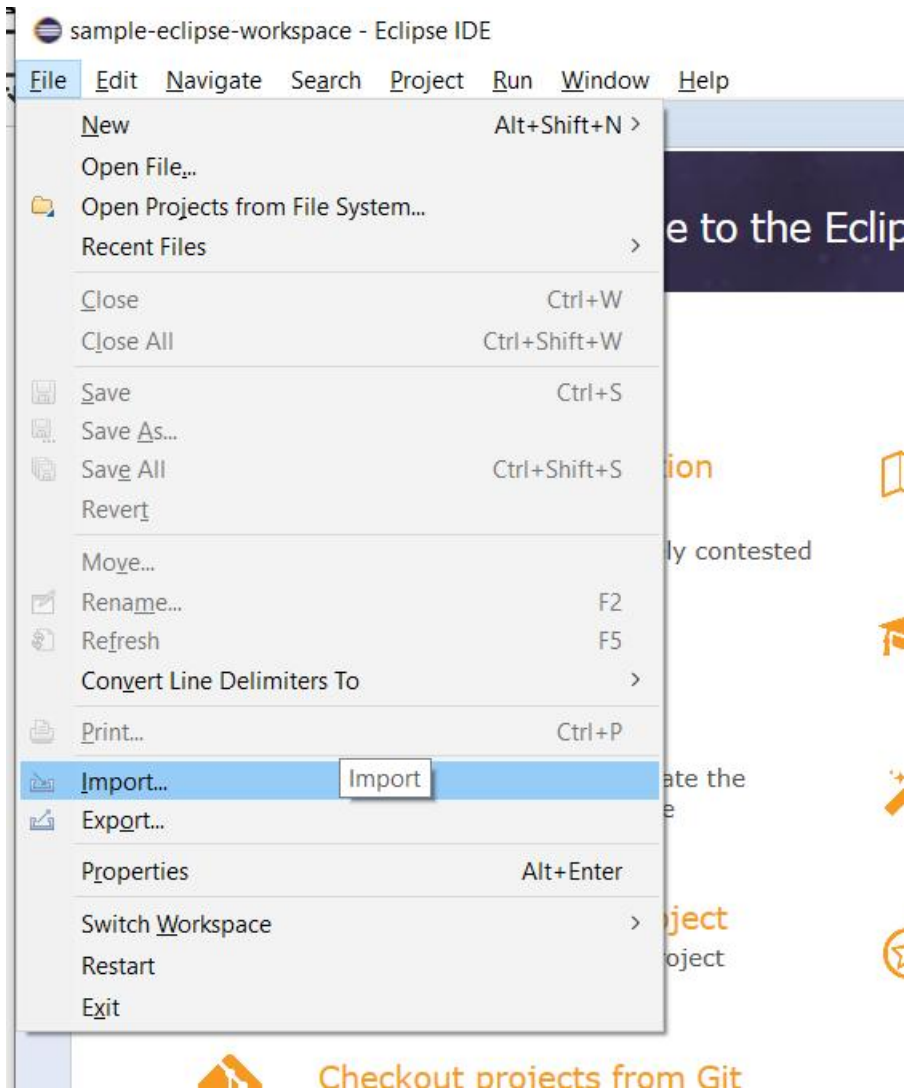
Browse...

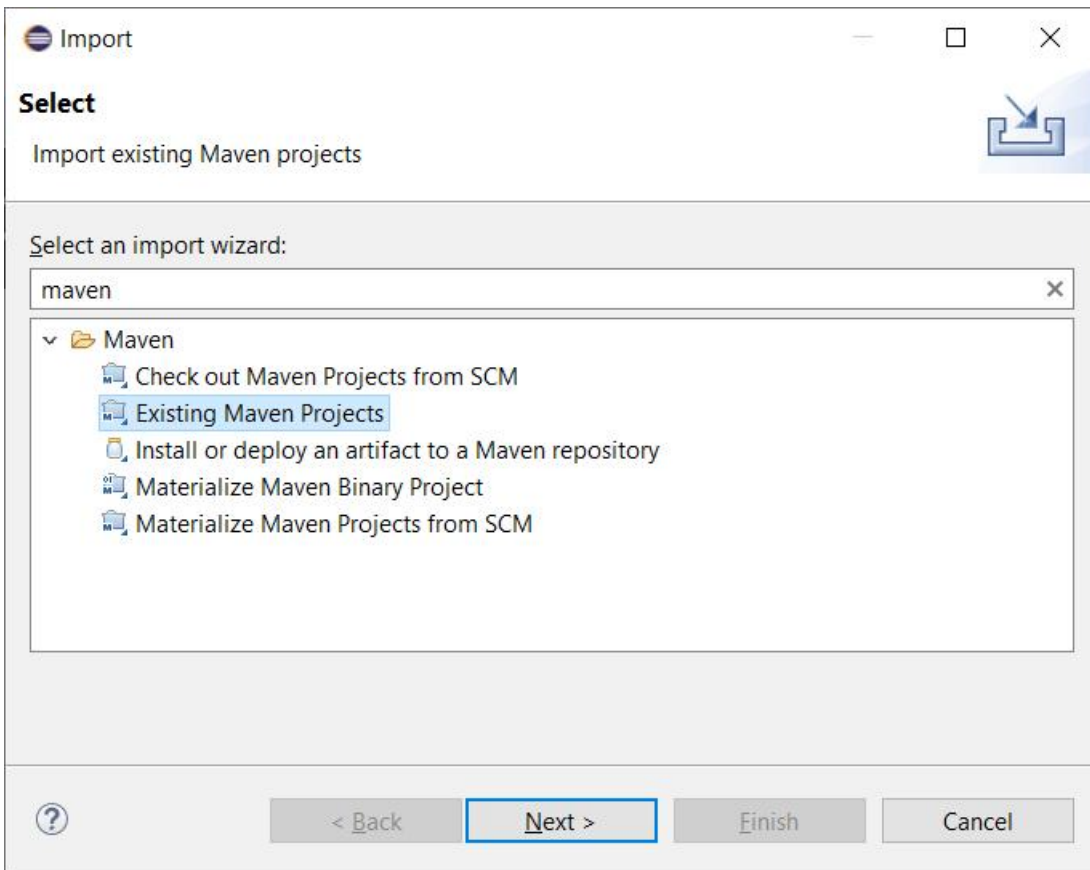
Show extracted files when complete

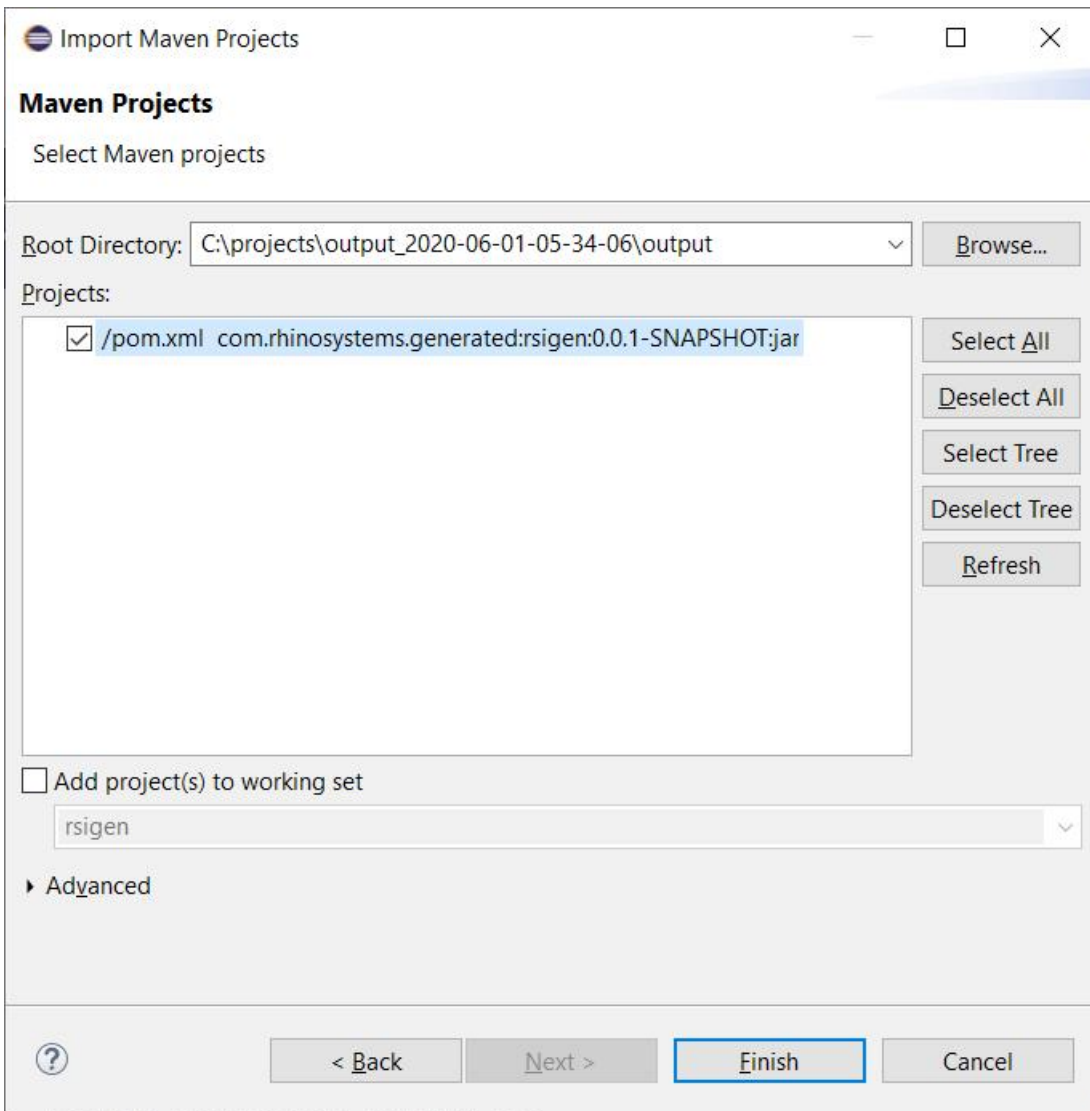
Extract

Cancel

 Defl...







sample-eclipse-workspace - rsigen/src/main/java/generated/Cars.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer Ownership.java Cars.java

```
15 @Table(name = "cars")
16 public class Cars {
17
18     @Id
19     @GeneratedValue(strategy = GenerationType.AUTO)
20     private Integer id;
21
22     @Column(name = "model")
23     private String model;
24
25     @Column(name = "make")
26     private String make;
27
28     @Column(name = "lmts")
29     private java.sql.Timestamp lmts;
30
31     public Integer getId() {
32         return id;
33     }
34
35     public void setId(Integer p) {
36         id = p;
37     }
38
39     public String getModel() {
40         return model;
41     }
42 }
```

rsigen

- src/main/java
 - generated
 - Cars.java
 - Cars
 - CarsApp.java
 - ObjectUtil.java
 - Ownership.java
 - OwnershipApp.java
 - Persons.java
 - PersonsApp.java
- src/main/resources
 - hibernate.cfg.xml
 - JRE System Library [JavaSE-1.7]
 - Maven Dependencies
 - src
 - target
 - pom.xml

sample-eclipse-workspace - rsigen/src/main/resources/hibernate.cfg.xml - Eclipse IDE

File Edit Source Navigate Search Project Run Window Help


















Package Explorer

- rsigen
 - src/main/java
 - generated
 - Cars.java
 - Cars
 - CarsApp.java
 - ObjectUtil.java
 - OwnershipApp.java
 - Persons.java
 - PersonsApp.java
 - src/main/resources
 - hibernate.cfg.xml
 - JRE System Library [JavaSE-1.7]
 - Maven Dependencies
 - src
 - target
 - pom.xml

Ownership.java Cars.java hibernate.cfg.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <!DOCTYPE hibernate-configuration SYSTEM
3 "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
4
5 <hibernate-configuration>
6   <session-factory>
7     <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</>
8     <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</>
9     <property name="hibernate.connection.url">jdbc:mysql://localhost/mydb</>
10    <property name="hibernate.connection.username">admin</property>
11    <property name="hibernate.connection.password">mypassword</property>
12    <property name="hibernate.hbm2ddl.auto">update</property>
13    <property name="show_sql">>true</property>
14    <mapping class="generated.Persons" />
15    <mapping class="generated.Cars" />
16    <mapping class="generated.Ownership" />
17
18  </session-factory>
19 </hibernate-configuration>
```

Session Factory Security Source

- ▼  rsigen
 - ▼  src/main/java
 - ▼  generated
 - >  Cars.java
 - ▼  CarsApp.java
 - ▼  CarsApp
 -  configuration
 -  dbg
 -  session
 -  sessionFactory
 -  create(Cars) : Integer
 -  delete(Integer) : void
 -  deleteAll() : void
 -  findById(Integer) : Cars
 -  getSession() : Session
 -  getSessionFactory() : SessionFactory
 -  main(String[]) : void
 -  read() : List<Cars>
 -  sampleData(int) : Cars
 -  update(Cars) : void

The image shows an IDE interface with a Package Explorer on the left and a code editor on the right. A context menu is open over the 'generated' folder in the Package Explorer. The menu items include: New, Open (F3), Open With, Open Type Hierarchy (F4), Open Call Hierarchy (Ctrl+Alt+H), Show In (Alt+Shift+W), Copy (Ctrl+C), Copy Qualified Name, Paste (Ctrl+V), Delete, Remove from Context (Ctrl+Alt+Shift+Down), Build Path, Source (Alt+Shift+S), Refactor (Alt+Shift+T), Import..., Export..., References, Declarations, Refresh (F5), Assign Working Sets..., Toggle Class Load Breakpoint, Coverage As, Run As, Debug As, Profile As, and Validate. The 'Run As' option is highlighted, and a sub-menu is visible with '1 Java Application' (Alt+Shift+X, J) and 'Run Configurations...'. The code editor shows an XML file named 'hibernate.cfg.xml' with the following content:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <!DOCTYPE hibernate-configuration PUBLIC
3 "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd"
4 [] [] []
5 <hibernate-configuration>
6   <session-factory>
7     <property name="hibernate.connection.driver_class">org.h2.Driver</property>
8     <property name="hibernate.connection.url">jdbc:h2:mem://test</property>
9     <property name="hibernate.connection.username">sa</property>
10    <property name="hibernate.connection.password"></property>
11    <property name="hibernate.dialect">org.hibernate.dialect.H2Dialect</property>
12    <property name="hibernate.show_sql">true</property>
13    <property name="hibernate.format_sql">true</property>
14    <mapping class="generate"></mapping>
15    <mapping class="generate"></mapping>
16    <mapping class="generate"></mapping>
17  </session-factory>
18 </hibernate-configuration>
```

```

<terminated> CarsApp [Java Application] C:\Program Files\Java\jre1.8.0_161\bin\javaw.exe (May 31, 2020, 10:55:23 PM - 10:55:26 PM)
Successfully created Cars: [{lmts:2020-05-31 22:55:26.473},{model:str_1590990926473_2},{id:5},{make:
fld:id,type=java.lang.Integer
SKIP ID fld:id,type=java.lang.Integer
fld:model,type=java.lang.String
fld:make,type=java.lang.String
fld:lmts,type=java.sql.Timestamp
Hibernate: insert into cars (lmts, make, model) values (?, ?, ?)
Successfully created Cars: [{lmts:2020-05-31 22:55:26.481},{model:str_1590990926481_3},{id:6},{make:
=====READ =====
Hibernate: select cars0_.id as id1_0_, cars0_.lmts as lmts2_0_, cars0_.make as make3_0_, cars0_.mode
Found 3 Carss
Cars: [{lmts:2020-05-31 22:55:24.0},{model:str_1590990924106_1},{id:4},{make:str_1590990924106_1}]
Cars: [{lmts:2020-05-31 22:55:26.0},{model:str_1590990926473_2},{id:5},{make:str_1590990926473_2}]
Cars: [{lmts:2020-05-31 22:55:26.0},{model:str_1590990926481_3},{id:6},{make:str_1590990926481_3}]
=====UPDATE =====
Successfully updated Cars: [{lmts:2020-05-31 22:55:24.106},{model:str_1590990924106_1},{id:4},{make:
=====READ =====
Hibernate: select cars0_.id as id1_0_, cars0_.lmts as lmts2_0_, cars0_.make as make3_0_, cars0_.mode
Found 3 Carss
Cars: [{lmts:2020-05-31 22:55:24.0},{model:str_1590990924106_1},{id:4},{make:str_1590990924106_1}]
Cars: [{lmts:2020-05-31 22:55:26.0},{model:str_1590990926473_2},{id:5},{make:str_1590990926473_2}]
Cars: [{lmts:2020-05-31 22:55:26.0},{model:str_1590990926481_3},{id:6},{make:str_1590990926481_3}]
=====DELETE =====
Hibernate: select cars0_.id as id1_0_0_, cars0_.lmts as lmts2_0_0_, cars0_.make as make3_0_0_, cars0
Hibernate: delete from cars where id=?
Successfully deleted Cars: [{lmts:2020-05-31 22:55:26.0},{model:str_1590990926473_2},{id:5},{make:st
=====READ =====
Hibernate: select cars0_.id as id1_0_, cars0_.lmts as lmts2_0_, cars0_.make as make3_0_, cars0_.mode
Found 2 Carss
Cars: [{lmts:2020-05-31 22:55:24.0},{model:str_1590990924106_1},{id:4},{make:str_1590990924106_1}]
Cars: [{lmts:2020-05-31 22:55:26.0},{model:str_1590990926481_3},{id:6},{make:str_1590990926481_3}]
=====DELETE ALL =====
Hibernate: delete from cars
Successfully deleted all employees.
=====READ =====
Hibernate: select cars0_.id as id1_0_, cars0_.lmts as lmts2_0_, cars0_.make as make3_0_, cars0_.mode
Found 0 Carss

```