

	20.000 Entry	100.000 Entry
Oracle Insert	28.45sn	142.70sn
Oracle Select	50.83sn	427.99sn
Hazelcast Put	5.512sn	22.52sn
Hazelcast Get	3.139sn	13.652sn

Oracle Results

```

13 public static void main(String[] args) throws InterruptedException {
14
15     // ORACLE INSERT
16     try
17     {
18         Class.forName("oracle.jdbc.driver.OracleDriver");
19         Connection connection = DriverManager.getConnection("jdbc:oracle:thin:@localhost:2130:GoldenDB", "berkay2", "berkay2");
20         Statement statement = connection.createStatement();
21
22         Random random = new Random();
23         long startTime = System.nanoTime();
24
25         for(int i = 0; i < 20000; i++)
26         {
27             statement.executeUpdate("INSERT INTO TEST VALUES('"+i+"','"+random.nextInt(20000)+"')");
28         }
29         long endTime = System.nanoTime() - startTime;
30         double second = (double)endTime / 1000000000.0;
31         System.out.print("Oracle Insert Time = "+second);
32     }
33     catch(Exception e)
34     {
35         System.out.print(e);
36     }
37
38
39
40
41 }

```

Markers Properties Servers Data Source Explorer Snippets Console Progress Coverage

<terminated> TestOracleInsert [Java Application] C:\Program Files\Java\jdk1.8.0_291\bin\javaw.exe (1 Agu 2021 23:26:44 - 23:27:15)

Oracle Insert Time = 28.451786999

```

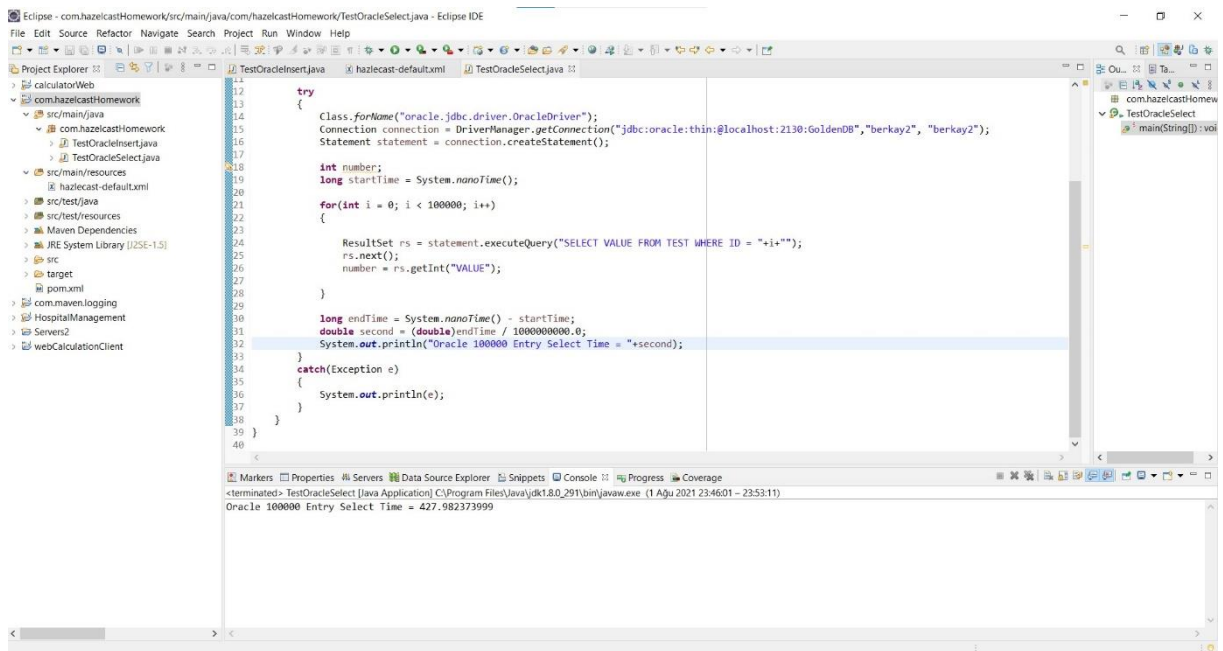
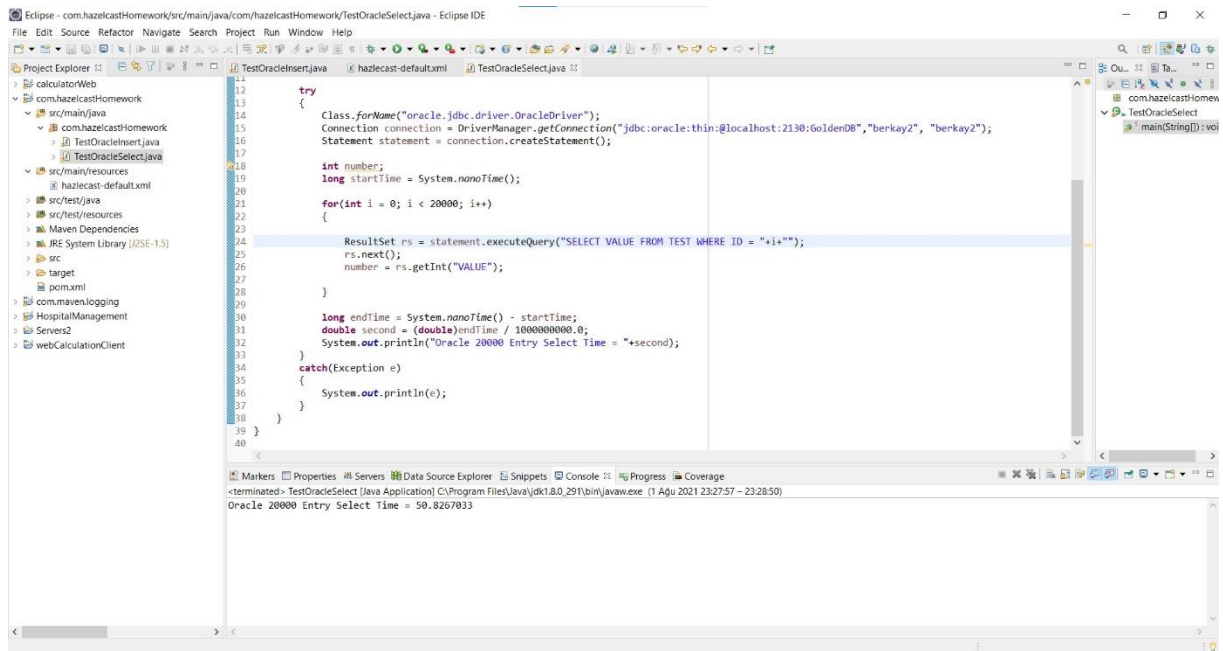
13 public static void main(String[] args) throws InterruptedException {
14
15     // ORACLE INSERT
16     try
17     {
18         Class.forName("oracle.jdbc.driver.OracleDriver");
19         Connection connection = DriverManager.getConnection("jdbc:oracle:thin:@localhost:2130:GoldenDB", "berkay2", "berkay2");
20         Statement statement = connection.createStatement();
21
22         Random random = new Random();
23         long startTime = System.nanoTime();
24
25         for(int i = 0; i < 100000; i++)
26         {
27             statement.executeUpdate("INSERT INTO TEST VALUES('"+i+"','"+random.nextInt(100000)+"')");
28         }
29         long endTime = System.nanoTime() - startTime;
30         double second = (double)endTime / 1000000000.0;
31         System.out.print("Oracle Insert Time = "+second);
32     }
33     catch(Exception e)
34     {
35         System.out.print(e);
36     }
37
38
39
40
41 }

```

Markers Properties Servers Data Source Explorer Snippets Console Progress Coverage

<terminated> TestOracleInsert [Java Application] C:\Program Files\Java\jdk1.8.0_291\bin\javaw.exe (1 Agu 2021 23:31:49 - 23:34:14)

Oracle Insert Time = 142.7036694



Hazelcast Results

The screenshot shows the Eclipse IDE with the `TestHazelcastPut.java` file open. The code defines a `TestHazelcastPut` class with a `main` method that inserts 20,000 random entries into a Hazelcast map. The console output shows the Hazelcast cluster starting and the insertion time.

```
1 package com.hazelcastHomework;
2
3 import java.util.Random;
4
5 import com.hazelcast.core.Hazelcast;
6 import com.hazelcast.core.HazelcastInstance;
7 import com.hazelcast.map.IMap;
8
9 public class TestHazelcastPut {
10
11     public static void main(String[] args) {
12         HazelcastInstance instance = Hazelcast.newHazelcastInstance();
13         IMap<Object, Object> map = instance.getMap("mapTest");
14
15         Random random = new Random();
16         long startTime = System.nanoTime();
17         for (int i = 0; i < 20000; i++) {
18             map.put(i, random.nextInt(20000));
19         }
20         long endTime = System.nanoTime() - startTime;
21         double seconds = (double)endTime / 1000000000.0;
22         System.out.println("Hazelcast 20000 Entry Insert Time = "+seconds);
23     }
24 }
25
26
27
```

Console Output:

```
Members {size:2, ver:2} [
  Member [192.168.1.33]:5701 - a76c5007-3769-458f-a267-bbfad7ad698
  Member [192.168.1.33]:5702 - db087322-0302-4064-a7d9-90146a0fe365 this
]
Aug 02, 2021 12:18:21 AM com.hazelcast.core.LifecycleService
INFO: [192.168.1.33]:5702 [dev] [4.2] [192.168.1.33]:5702 is STARTED
Hazelcast 20000 Entry Insert Time = 5.511314499
```

The screenshot shows the Eclipse IDE with the `TestHazelcastPut.java` file open. The code defines a `TestHazelcastPut` class with a `main` method that inserts 100,000 random entries into a Hazelcast map. The console output shows the Hazelcast cluster starting and the insertion time.

```
1 package com.hazelcastHomework;
2
3 import java.util.Random;
4
5 public class TestHazelcastPut {
6
7     public static void main(String[] args) {
8         HazelcastInstance instance = Hazelcast.newHazelcastInstance();
9         IMap<Object, Object> map = instance.getMap("mapTest");
10
11         Random random = new Random();
12         long startTime = System.nanoTime();
13         for (int i = 0; i < 100000; i++) {
14             map.put(i, random.nextInt(100000));
15         }
16         long endTime = System.nanoTime() - startTime;
17         double seconds = (double)endTime / 1000000000.0;
18         System.out.println("Hazelcast 100000 Entry Insert Time = "+seconds);
19     }
20 }
21
22
23
```

Console Output:

```
Members {size:2, ver:2} [
  Member [192.168.1.33]:5701 - 4fb46976-1198-4707-866f-955414536f1b
  Member [192.168.1.33]:5702 - 100e5f26-bcc2-4638-8bbb-f6bed220bc6 this
]
Aug 02, 2021 12:33:41 AM com.hazelcast.core.LifecycleService
INFO: [192.168.1.33]:5702 [dev] [4.2] [192.168.1.33]:5702 is STARTED
Hazelcast 100000 Entry Insert Time = 22.519760699
```

