

## **EXPERIMENT -3**

# **Working with HQL - Sorting, Pagination & Aggregates**

Hibernate is an Object Relational Mapping (ORM) framework that maps Java classes to database tables.

Hibernate Query Language (HQL) is similar to SQL but operates on **Java objects instead of database tables**.

HQL supports:

- **Sorting** using ORDER BY
- **Pagination** using setFirstResult() and setMaxResults()
- **Aggregate functions** such as COUNT, AVG, MAX

### **Database Table**

**Table Name:** student

<b>Field</b>	<b>Type</b>
id	int (Primary Key)
name	varchar
marks	int

### **Program Code**

#### **1. Student Entity Class**

```
package com.example.entity;  
import jakarta.persistence.*;
```

```
@Entity
@Table(name = "student")
public class Student {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;
    private String name;
    private int marks;
    public Student() {}
    public Student(String name, int marks) {
        this.name = name;
        this.marks = marks;
    }
    public int getId() { return id; }
    public void setId(int id) { this.id = id; }
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
    public int getMarks() { return marks; }
    public void setMarks(int marks) { this.marks = marks; }
    @Override
    public String toString() {
        return "Student{id=" + id + ", name='" + name + "', marks=" + marks + "}";
    }
}
```

## 2. StudentDAO Class (HQL Operations)

```
package com.example.dao;
import java.util.List;
import org.hibernate.Session;
import org.hibernate.Transaction;
import org.hibernate.query.Query;
import com.example.entity.Student;
import com.example.util.HibernateUtil;
public class StudentDAO {
    public void saveStudent(Student student) {
        Session session =
        HibernateUtil.getSessionFactory().openSession();
        Transaction tx = session.beginTransaction();
        session.save(student);
        tx.commit();
        session.close();
    }
    public List<Student> getStudentsSortedByMarks() {
        Session session =
        HibernateUtil.getSessionFactory().openSession();
        String hql = "FROM Student ORDER BY marks DESC";
        List<Student> list = session.createQuery(hql,
        Student.class).list();
        session.close();
    }
}
```

```
        return list;
    }

    public List<Student> getStudentsWithPagination(int start,
int limit) {
    Session session =
HibernateUtil.getSessionFactory().openSession();

    Query<Student> query = session.createQuery("FROM
Student", Student.class);

    query.setFirstResult(start);
    query.setMaxResults(limit);
    List<Student> list = query.list();
    session.close();
    return list;
}

public long getStudentCount() {
    Session session =
HibernateUtil.getSessionFactory().openSession();

    Query<Long> query = session.createQuery(
        "SELECT COUNT(*) FROM Student", Long.class);
    long count = query.uniqueResult();
    session.close();
    return count;
}

public double getAverageMarks() {
    Session session =
HibernateUtil.getSessionFactory().openSession();
```

```

        Query<Double> query = session.createQuery(
            "SELECT AVG(marks) FROM Student",
            Double.class);

        double avg = query.uniqueResult();
        session.close();
        return avg;
    }

    public int getMaxMarks() {
        Session session =
        HibernateUtil.getSessionFactory().openSession();

        Query<Integer> query = session.createQuery(
            "SELECT MAX(marks) FROM Student",
            Integer.class);

        int max = query.uniqueResult();
        session.close();
        return max;
    }
}

```

### **3. Main Application Class**

```

package com.example;

import java.util.List;

import com.example.dao.StudentDAO;
import com.example.entity.Student;

public class MainApp {

    public static void main(String[] args) {
        StudentDAO dao = new StudentDAO();

```

```

        dao.saveStudent(new Student("Karthik", 100));
        dao.saveStudent(new Student("Vikram", 85));
        dao.saveStudent(new Student("Anjali", 90));
        dao.saveStudent(new Student("Ravi", 75));
        System.out.println("Students Sorted by Marks:");
        List<Student> sorted = dao.getStudentsSortedByMarks();
        sorted.forEach(System.out::println);
        System.out.println("\nPagination (First 2 Records):");
        dao.getStudentsWithPagination(0, 2)
            .forEach(System.out::println);
        System.out.println("\nAggregate Functions:");
        System.out.println("Total Students: " +
        dao.getStudentCount());
        System.out.println("Average Marks: " +
        dao.getAverageMarks());
        System.out.println("Maximum Marks: " +
        dao.getMaxMarks());
    }
}

```

#### **4. Hibernate Configuration File**

```

<hibernate-configuration>
    <session-factory>
        <property name="hibernate.connection.driver_class">
            com.mysql.cj.jdbc.Driver
        </property>
        <property name="hibernate.connection.url">

```

```
    jdbc:mysql://localhost:3306/experiment
</property>
<property
name="hibernate.connection.username">root</property>
<property
name="hibernate.connection.password">1313</property>
<property name="hibernate.dialect">
    org.hibernate.dialect.MySQL8Dialect
</property>
<property
name="hibernate.hbm2ddl.auto">create</property>
<property name="show_sql">true</property>
<property name="format_sql">true</property>
<mapping class="com.example.entity.Student"/>
</session-factory>
</hibernate-configuration>
```

## OUTPUT:

The screenshot shows the VS Code interface with the following details:

- Explorer:** Shows the project structure under the `MAIN` folder, including `src/main/java/com/example`, `dao`, `entity`, `MainApp.java`, and `resources`. A `pom.xml` file is also listed.
- Maven Terminal:** The active tab is the `TERMINAL`. The command `mvn clean pack` is being run, and the output is displayed:

```
PS C:\Users\JAINI VIKRAM\OneDrive\Pictures\Documents - Copy\2nd Year\2nd year Second Sem\FSAD\main> mvn clean pack
[INFO] Scanning for projects...
[INFO] [INFO] -----[ com.example:hibernate-demo ]-----
[INFO] [INFO] Building hibernate-demo 1.0
[INFO] [INFO] from pom.xml
[INFO] [INFO] -----[ jar ]-----
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-shade-plugin/3.5.0/maven-shade-plugin-3.5.0.pom
[INFO] Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-shade-plugin/3.5.0/maven-shade-plugin-3.5.0.pom (12 kB at 27 kB/s)
[INFO] [INFO] --- clean:3.2.0:clean (default-clean) @ hibernate-demo ---
[INFO] [INFO] Deleting C:\Users\JAINI VIKRAM\OneDrive\Pictures\Documents - Copy\2nd Year\2nd year Second Sem\FSAD\main\target\*.class
[INFO] [INFO] [INFO] --- resources:3.3.1:resources (default-resources) @ hibernate-demo ---
[INFO] [INFO] Copying 1 resource from src/main/resources to target\classes
[INFO] [INFO] --- compiler:3.13.0:compile (default-compile) @ hibernate-demo ---
[INFO] [INFO] Recompiling the module because of changed source code.
[INFO] [INFO] Compiling 4 source files with javac [debug target 17] to target\classes
[WARNING] [INFO] Location of system modules is not set in conjunction with -source 17
[INFO] [INFO] Not setting the location of system modules may lead to class files that cannot run on JDK 17
[INFO] [INFO] --release 17 is recommended instead of -source 17 -target 17 because it sets the location of system modules automatically
[INFO] [INFO] /C:/Users/JAINI VIKRAM/OneDrive/Pictures/Documents - Copy\2nd Year\2nd year Second Sem\FSAD\main\src\main\j
```
- Problems:** 24 problems are listed.
- Output:** 0 messages.
- Debug Console:** 0 messages.
- Terminal:** The active tab, showing the Maven build process.
- Ports:** 0 messages.
- PowerShell:** A sidebar with two entries: `powershell` and `powershell`.

```

PS C:\Users\JAINI VIKRAM\OneDrive\Pictures\Documents - Copy\2nd Year\2nd year Second Sem\FSAD\main> java -jar targ
et/hibernate-demo-1.0.jar
Dec 25, 2025 9:36:42 AM org.hibernate.Version logVersion
INFO: HHH000412: Hibernate ORM core version [WORKING]
Dec 25, 2025 9:36:42 AM org.hibernate.boot.jaxb.internal.stax.LocalXmlResourceResolver resolveEntity
WARN: HHH90000012: Recognized obsolete hibernate namespace http://hibernate.sourceforge.net/hibernate-configuration. Use namespace http://www.hibernate.org/dtd/hibernate-configuration instead. Support for obsolete DTD/XSD namespaces may be removed at any time.
Dec 25, 2025 9:36:43 AM org.hibernate.cache.internal.RegionFactoryInitiator initiateService
INFO: HHH000026: Second-level cache disabled
Dec 25, 2025 9:36:43 AM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl configu
re
WARN: HHH10001002: Using built-in connection pool (not intended for production use)
Dec 25, 2025 9:36:43 AM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCr
eator
INFO: HHH10001005: Loaded JDBC driver class: com.mysql.cj.jdbc.Driver
Dec 25, 2025 9:36:43 AM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCr
eator
INFO: HHH10001012: Connecting with JDBC URL [jdbc:mysql://localhost:3306/experiment?useSSL=false&serverTimezone=UT
C&allowPublicKeyRetrieval=true]
Dec 25, 2025 9:36:43 AM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCr
eator
INFO: HHH10001001: Connection properties: {user=root, password=****}
Dec 25, 2025 9:36:43 AM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCr
eator
INFO: HHH10001003: Autocommit mode: false
Dec 25, 2025 9:36:43 AM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledC
onnections <init>

```

TERMINAL

```

[WARNING] mvn dependency:tree -Ddetail=true and the above output.
[WARNING] See https://maven.apache.org/plugins/maven-shade-plugin/
[INFO] Replacing original artifact with shaded artifact.
[INFO] Replacing C:\Users\JAINI VIKRAM\OneDrive\Pictures\Documents - Copy\2nd Year\2nd year Second Sem\FSAD\main\t
arget\hibernate-demo-1.0.jar with C:\Users\JAINI VIKRAM\OneDrive\Pictures\Documents - Copy\2nd Year\2nd year Secon
d Sem\FSAD\main\target\hibernate-demo-1.0-shaded.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 7.194 s
[INFO] Finished at: 2025-12-25T09:36:08+05:30
[INFO] -----
PS C:\Users\JAINI VIKRAM\OneDrive\Pictures\Documents - Copy\2nd Year\2nd year Second Sem\FSAD\main> dir target

Directory: C:\Users\JAINI VIKRAM\OneDrive\Pictures\Documents - Copy\2nd Year\2nd year Second
Sem\FSAD\main\target

Mode LastWriteTime Length Name
---- ----- ---- -
d--- 25-12-2025 09:36 classes
d--- 25-12-2025 09:36 generated-sources
dar-- 25-12-2025 09:36 maven-archiver
d--- 25-12-2025 09:36 maven-status
-a-- 25-12-2025 09:36 22069530 hibernate-demo-1.0.jar
-a-- 25-12-2025 09:36 6513 original-hibernate-demo-1.0.jar

```

```

e
INFO: HHH000489: No JTA platform available (set 'hibernate.transaction.jta.platform' to enable JTA platform integrat
ion)
Hibernate:
    drop table if exists student
Dec 25, 2025 9:36:44 AM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl
getIsolatedConnection
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvir
onmentInitiator$ConnectionProviderJdbcConnectionAccess@760487aa] for (non-JTA) DDL execution was not in auto-commi
t mode; the Connection 'local transaction' will be committed and the Connection will be set into auto-commit mode.
Hibernate:
    create table student (
        id integer not null auto_increment,
        marks integer not null,
        name varchar(255),
        primary key (id)
        name varchar(255),
        primary key (id)
    ) engine=InnoDB
Dec 25, 2025 9:36:44 AM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl
getIsolatedConnection
    name varchar(255),
    primary key (id)
) engine=InnoDB
Dec 25, 2025 9:36:44 AM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl
getIsolatedConnection
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvir
name varchar(255),

```

```
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProviderJdbcConnectionAccess@106b014e] for (non-JTA) DDL execution was not in auto-commit mode; the Connection 'local transaction' will be committed and the Connection will be set into auto-commit mode.
Hibernate:
    insert
    into
        student
        (marks, name)
    onmentInitiator$ConnectionProviderJdbcConnectionAccess@106b014e] for (non-JTA) DDL execution was not in auto-commit mode; the Connection 'local transaction' will be committed and the Connection will be set into auto-commit mode.
Hibernate:
    insert
    into
        student
        (marks, name)
Hibernate:
    insert
    into
        student
        (marks, name)
values
    (?, ?)
into
    student
    (marks, name)
values
    (?, ?)
values
    (?, ?)
    (?, ?)
Hibernate:
    insert
    into
        student
        (marks, name)
values
    (?, ?)
Student saved successfully!
PS C:\Users\JAINI VIKRAM\OneDrive\Pictures\Documents - Copy\2nd Year\2nd year Second Sem\FSAD\main> 
```