



Welcome to Student Management

Team members:
Ahmed Mohamed Abdelaziz
Kareem Mohamed Badr
Abdelrahman Hamada

Description:

- ▶ Our project is about keep tracking for students' registration and attendance for classes.
- ▶ We have 3 main classes are :

1- Section

2- Subject

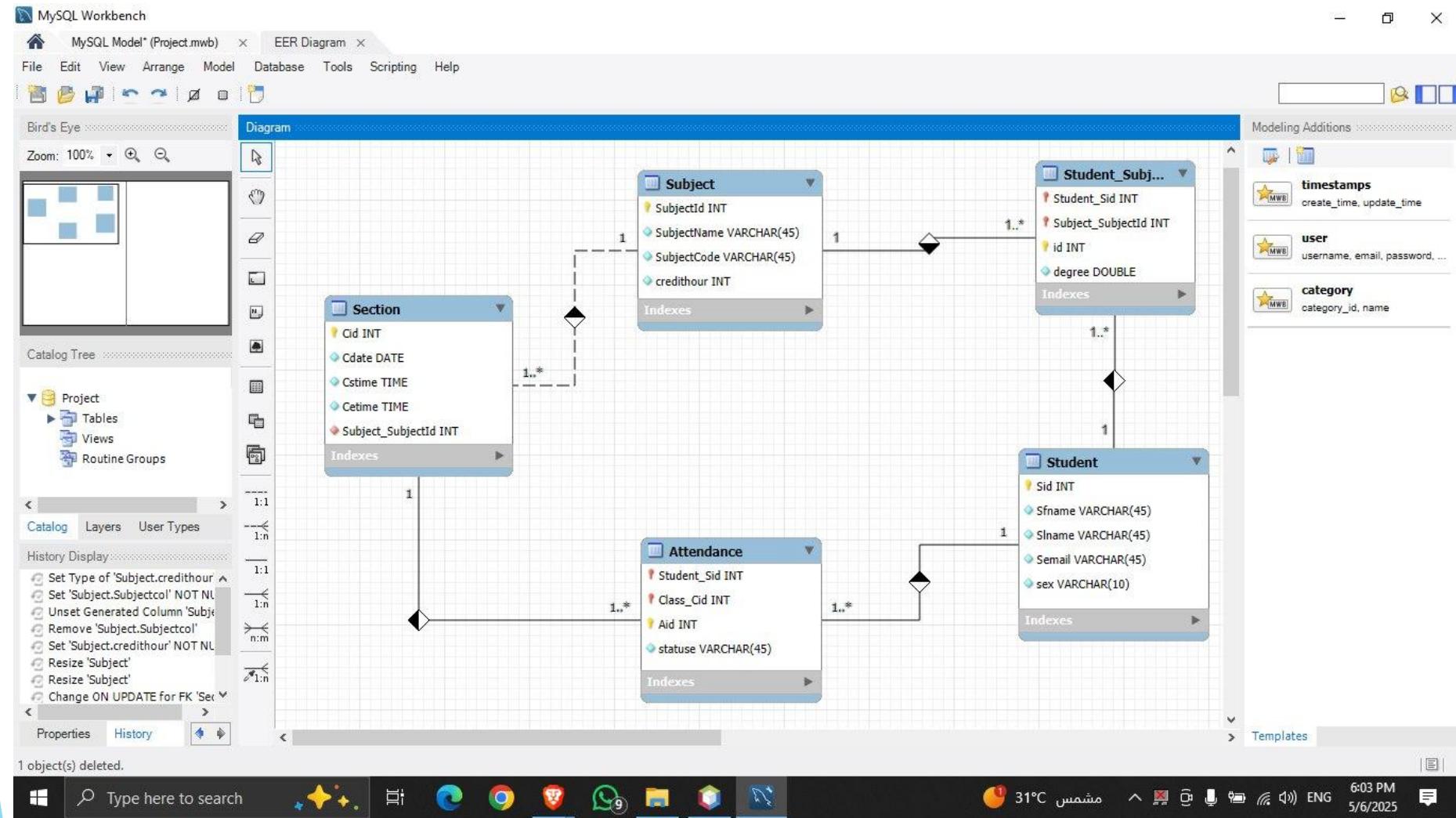
3-Student

And we have 2 many to many relationships are:

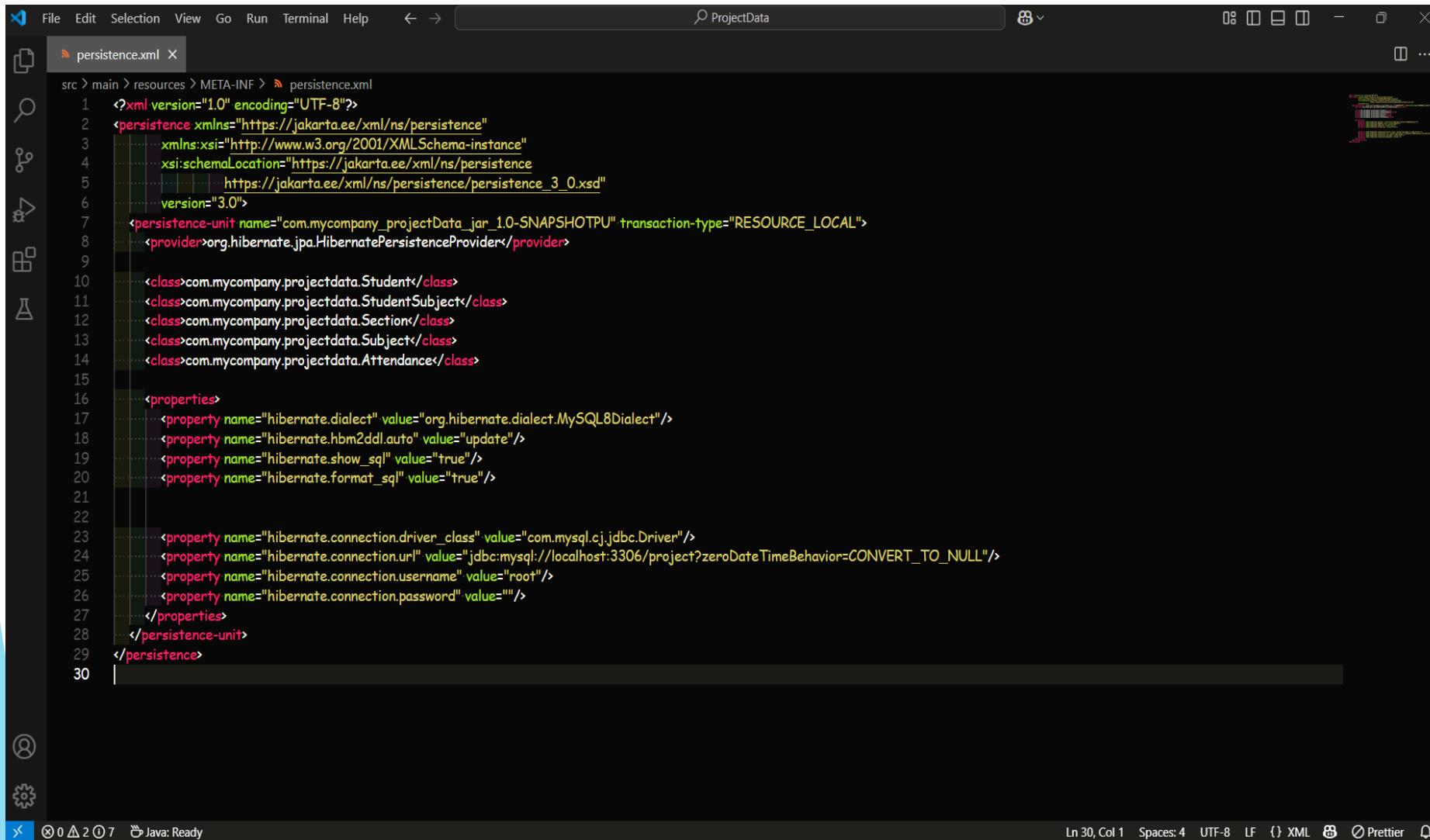
1-Attendance

2-Student_Subject

EER Diagram:



The Xml file



```
File Edit Selection View Go Run Terminal Help ← → ProjectData persistence.xml src > main > resources > META-INF > persistence.xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <persistence xmlns="https://jakarta.ee/xml/ns/persistence"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4      xsi:schemaLocation="https://jakarta.ee/xml/ns/persistence
5          https://jakarta.ee/xml/ns/persistence/persistence_3_0.xsd"
6      version="3.0">
7      <persistence-unit name="com.mycompany_projectData_jar_1.0-SNAPSHOTPU" transaction-type="RESOURCE_LOCAL">
8          <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>
9
10         <class>com.mycompany.projectdata.Student</class>
11         <class>com.mycompany.projectdata.StudentSubject</class>
12         <class>com.mycompany.projectdata.Section</class>
13         <class>com.mycompany.projectdata.Subject</class>
14         <class>com.mycompany.projectdata.Attendance</class>
15
16         <properties>
17             <property name="hibernate.dialect" value="org.hibernate.dialect.MySQL8Dialect"/>
18             <property name="hibernate.hbm2ddl.auto" value="update"/>
19             <property name="hibernate.show_sql" value="true"/>
20             <property name="hibernate.format_sql" value="true"/>
21
22
23             <property name="hibernate.connection.driver_class" value="com.mysql.cj.jdbc.Driver"/>
24             <property name="hibernate.connection.url" value="jdbc:mysql://localhost:3306/project?zeroDateTimeBehavior=CONVERT_TO_NULL"/>
25             <property name="hibernate.connection.username" value="root"/>
26             <property name="hibernate.connection.password" value="" />
27         </properties>
28     </persistence-unit>
29 </persistence>
30 |
```

Ln 30, Col 1 Spaces: 4 UTF-8 LF {} XML Prettier

Student Table before insert

ID	FirstName	Last Name	Email	Sex
4	mesh	zizo	meshzizo@gmail.com	Male
5	Mohamed	Hassan	mohamed.hassan@example.com	Male
6	keko	badr	badr@gmail.com	Male
7	abdo	hamada	abdo@gmail.com	Male

[Back](#) [Insert](#) [Update](#) [Delete](#)

Student ID:

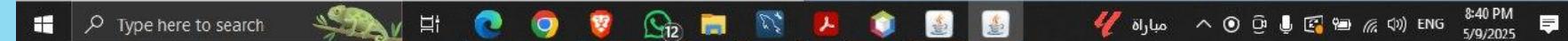
First Name:

Last Name:

Email:

Sex:

Confirm Insert



Student Table after insert

Student Management

ID	First Name	Last Name	Email	Sex
1	Rodina	Mohamed	Roro@gmail.com	Female
4	mesh	zizo	meshzizo@gmail.com	Male
5	Mohamed	Hassan	mohamed.hassan@example.com	Male
6	keko	badr	badr@gmail.com	Male
7	abdo	hamada	abdo@gmail.com	Male

[Back](#) [Insert](#) [Update](#) [Delete](#)

Student ID:

First Name:

Last Name:

Email:

Sex:

[Confirm Insert](#)



Type here to search

8:40 PM 5/9/2025

Student table before update

Student Management

ID	First Name	Last Name	Email	Sex
1	Rodina	Mohamed	Roro@gmail.com	Female
4	mesh	zizo	meshzizo@gmail.com	Male
5	Mohamed	Hassan	mohamed.hassan@example.com	Male
6	keko	badr	badr@gmail.com	Male
7	abdo	hamada	abdo@gmail.com	Male

Back Insert Update Delete

Student ID: 4

First Name: mesh

Last Name: Ahmed

Email: meshzizo@gmail.com

Sex: Male

Confirm Update

Type here to search   ENG 8:41 PM 5/9/2025

Student table after update

Student Management

ID	First Name	Last Name	Email	Sex
1	Rodina	Mohamed	Roro@gmail.com	Female
4	mesh	Ahmed	meshzizo@gmail.com	Male
5	Mohamed	Hassan	mohamed.hassan@example.com	Male
6	keko	badr	badr@gmail.com	Male
7	abdo	hamada	abdo@gmail.com	Male

Back Insert Update Delete

Student ID:

First Name:

Last Name:

Email:

Sex:

Confirm Update



Type here to search                    8:41 PM 5/9/2025

Student table before delete

Student Management

ID	First Name	Last Name	Email	Sex
1	Rodina	Mohamed	Roro@gmail.com	Female
4	mesh	Ahmed	meshzizo@gmail.com	Male
5	Mohamed	Hassan	mohamed.hassan@example.com	Male
6	keko	badr	badr@gmail.com	Male
7	abdo	hamada	abdo@gmail.com	Male

Back Insert Update Delete

Student ID: 5

Confirm Delete

Type here to search                      ENG 8:41 PM 5/9/2025

Student table after delete

Student Management

ID	First Name	Last Name	Email	Sex
1	Rodina	Mohamed	Roro@gmail.com	Female
4	mesh	Ahmed	meshzizo@gmail.com	Male
6	keko	badr	badr@gmail.com	Male
7	abdo	hamada	abdo@gmail.com	Male

Back Insert Update Delete

Student ID: 5

Confirm Delete

Type here to search

Windows icon

System tray icons: battery, signal, volume, clock (8:41 PM), date (5/9/2025), and a message bubble.

Section table before insert

Section Management

ID	Date	Start Time	End Time	Subject ID
204	2025-05-10	12:00:00	13:00:00	101
207	2025-05-10	22:00:00	00:00:00	207

[Back](#) [Insert](#) [Update](#) [Delete](#)

Section ID:

Section Date (yyyy-MM-dd):

Start Time (HH:mm:ss):

End Time (HH:mm:ss):

Subject ID:

Confirm Insert



Type here to search              32°C      ENG 9:19 PM 5/9/2025 

Section table after insert

Section Management

ID	Date	Start Time	End Time	Subject ID
204	2025-05-10	12:00:00	13:00:00	101
207	2025-05-10	22:00:00	00:00:00	207
301	2025-05-15	09:00:00	12:00:00	102

[Back](#) [Insert](#) [Update](#) [Delete](#)

Section ID:

Section Date (yyyy-MM-dd):

Start Time (HH:mm:ss):

End Time (HH:mm:ss):

Subject ID:

Confirm Insert

USD/ILS 10,82- 9:27 PM 5/9/2025

Section table before update

Section Management

ID	Date	Start Time	End Time	Subject ID
204	2025-05-10	12:00:00	13:00:00	101
207	2025-05-10	22:00:00	00:00:00	207
301	2025-05-15	09:00:00	12:00:00	102

[Back](#) [Insert](#) [Update](#) [Delete](#)

Section ID:

Section Date (yyyy-MM-dd):

Start Time (HH:mm:ss):

End Time (HH:mm:ss):

Subject ID:

Confirm Update

Type here to search                          USD/ILS 10,82- 9:28 PM 5/9/2025 ENG

Section table after update

Section Management

ID	Date	Start Time	End Time	Subject ID
204	2025-05-10	12:00:00	13:00:00	101
207	2025-05-10	22:00:00	00:00:00	207
301	2028-12-01	13:00:00	15:00:00	102

Back Insert Update Delete

Section ID:

Section Date (yyyy-MM-dd):

Start Time (HH:mm:ss):

End Time (HH:mm:ss):

Subject ID:

Confirm Update

USD/ILS 10,82- 9:28 PM 5/9/2025

Section table before delete

Section Management

ID	Date	Start Time	End Time	Subject ID
204	2025-05-10	12:00:00	13:00:00	101
207	2025-05-10	22:00:00	00:00:00	207
301	2028-12-01	13:00:00	15:00:00	102

Back Insert Update Delete

Section ID:

301

Confirm Delete

Type here to search              USD/ILS 0,82-          ENG 9:28 PM 5/9/2025

Section table after delete

Section Management

ID	Date	Start Time	End Time	Subject ID
204	2025-05-10	12:00:00	13:00:00	101
207	2025-05-10	22:00:00	00:00:00	207

Back Insert Update Delete

Section ID:

Confirm Delete

USD/ILS 0,82- 9:28 PM
5/9/2025

Subject table before insert

Subject Management

ID	Name	Code	Credit Hour
101	Database	COMP101	3
102	Computer Networks	CN102	3
103	Operating Systems	OS103	4
207	Java	COMP207	3

Back Insert Update Delete

Subject ID:

Subject Name:

Subject Code:

Credit Hour:

Confirm Insert

32°C ٩:١٢ PM ENG 5/9/2025

Subject table after insert

Subject Management

ID	Name	Code	Credit Hour
101	Database	COMP101	3
102	Computer Networks	CN102	3
103	Operating Systems	OS103	4
207	Java	COMP207	3
210	Network	COMP210	3

[Back](#) [Insert](#) [Update](#) [Delete](#)

Subject ID:

Subject Name:

Subject Code:

Credit Hour:

[Confirm Insert](#)

Type here to search             32°C     ENG 9:13 PM 5/9/2025

Subject table before update

Subject Management

ID	Name	Code	CreditHour
101	Database	COMP101	3
102	Computer Networks	CN102	3
103	Operating Systems	OS103	4
207	Java	COMP207	3
210	Network	COMP210	3

[Back](#) [Insert](#) [Update](#) [Delete](#)

Subject ID:

Subject Name:

Subject Code:

Credit Hour:

Confirm Update



Subject table after update

Subject Management

ID	Name	Code	CreditHour
101	Database	COMP101	3
102	Computer Networks	CN102	3
103	Operating System	Comp103	3
207	Java	COMP207	3
210	Network	COMP210	3

[Back](#) [Insert](#) [Update](#) [Delete](#)

Subject ID:

103

Subject Name:

Operating System

Subject Code:

Comp103

Credit Hour:

3

Confirm Update



Subject table before delete

Subject Management

ID	Name	Code	Credit Hour
101	Database	COMP101	3
102	Computer Networks	CN102	3
103	Operating System	Comp103	3
207	Java	COMP207	3
210	Network	COMP210	3

Back Insert Update Delete

Subject ID: 210

Confirm Delete



Type here to search

32°C 9:14 PM 5/9/2025

Subject table after delete

Subject Management

ID	Name	Code	Credit Hour
101	Database	COMP101	3
102	Computer Networks	CN102	3
103	Operating System	Comp103	3
207	Java	COMP207	3

Back Insert Update Delete

Subject ID:

Confirm Delete

Type here to search              32°C     ENG 9:14 PM 5/9/2025 

Attendance table before insert

Attendance Management

Student ID	Section ID	AID	Status
------------	------------	-----	--------

Buttons: Insert, Update, Delete, Back

Form Fields:

Student ID:	1
Section ID:	204
AID:	1
Status:	ok

Confirm Insert:

9:33 PM 5/9/2025

Attendance table after insert

Attendance Management

	Student ID	Section ID	AID	Status
1	204	1	ok	

Insert **Update** **Delete** **Back**

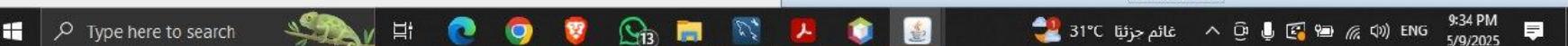
Student ID:

Section ID:

AID:

Status:

Confirm Insert



Type here to search

31°C ٩:٣٤ PM
5/9/2025

Attendance table before update

Attendance Management

Student ID	Section ID	AID	Status
1	204	1	ok

Buttons: Insert, Update, Delete, Back

Form Fields:

- Student ID: 1
- Section ID: 204
- AID: 1
- Status: absent

Confirm Update

Type here to search                       31°C 9:35 PM 5/9/2025

Attendance table after update

Attendance Management

	Student ID	Section ID	AID	Status
1	204	1		absent

[Insert](#) [Update](#) [Delete](#) [Back](#)

Student ID:

Section ID:

AID:

Status:

[Confirm Update](#)

Type here to search                      31°C 9:35 PM 5/9/2025

Attendance table before delete

Attendance Management

	Student ID	Section ID	AID	Status
1	204		1	absent
4	204		2	ok
6	204		3	absent
6	207		3	ok
7	204		4	absent

[Insert](#) [Update](#) [Delete](#) [Back](#)

Student ID:

6

Section ID:

207

AID:

3

Confirm Delete



Type here to search



ملاحة

ENG

Attendance table after delete

Attendance Management

	Student ID	Section ID	AID	Status
1	204	1		absent
4	204	2		ok
6	204	3		absent
7	204	4		absent

[Insert](#) [Update](#) [Delete](#) [Back](#)

Student ID:

Section ID:

AID:

[Confirm Delete](#)

Type here to search                  ENG 9:38 PM 5/9/2025

Student_Subject table before insert

Student-Subject Management

Student ID	Subject ID	ID	Degree
4	207	100	200.00
6	101	11	120.00
7	101	12	120.00

[Back](#) [Insert](#) [Update](#) [Delete](#)

Student ID:

Subject ID:

ID:

Degree:

Confirm Insert

Type here to search

31°C ٣١°C ٢٠٢٥ ENG 9:30 PM 5/9/2025

Student_Subject table after insert

Student-Subject Management

Student ID	Subject ID	ID	Degree
1	103	13	200.00
4	207	100	200.00
6	101	11	120.00
7	101	12	120.00

[Back](#) [Insert](#) [Update](#) [Delete](#)

Student ID:

Subject ID:

ID:

Degree:

Confirm Insert

Type here to search                    ENG 9:30 PM 5/9/2025

Student_Subject table before update

Student-Subject Management

	Student ID	Subject ID	ID	Degree
1	103	13	200.00	
4	207	100	200.00	
6	101	11	120.00	
7	101	12	120.00	

Back Insert Update Delete

Student ID:

Subject ID:

ID:

Degree:

Confirm Update

Type here to search

31°C ٢٤ جمادى الآخرة ١٤٢٥ ٩:٣٠ PM ENG 5/9/2025

Student_Subject table after update

Student-Subject Management

	Student ID	Subject ID	ID	Degree
1	103	13	103.00	
4	207	100	200.00	
6	101	11	120.00	
7	101	12	120.00	

Back Insert Update Delete

Student ID:

Subject ID:

ID:

Degree:

Type here to search                    ENG 9:31 PM 5/9/2025

Confirm Update

Student_Subject table before delete

Student-Subject Management

	Student ID	Subject ID	ID	Degree
1	103	13	103.00	
4	207	100	200.00	
6	101	11	120.00	
7	101	12	120.00	

[Back](#) [Insert](#) [Update](#) [Delete](#)

Student ID:

Subject ID:

ID:

Confirm Delete



Type here to search                    ENG 9:32 PM 5/9/2025

Student_Subject table after delete

Student-Subject Management

	Student ID	Subject ID	ID	Degree
1	103	13	103.00	
4	207	100	200.00	
6	101	11	120.00	

[Back](#) [Insert](#) [Update](#) [Delete](#)

Student ID:

Subject ID:

ID:

[Confirm Delete](#)

Type here to search    31°C  عالمي ENG 9:32 PM 5/9/2025

Insert function in Student

```
13
14
15 public void insertStudent(Integer sid, String sfname, String slname, String semail, String sex){
16     EntityManager em = emf.createEntityManager();
17     try{
18         em.getTransaction().begin();
19         Student student = new Student(sid, sfname, slname, semail, sex);
20         em.persist(student);
21         em.getTransaction().commit();
22         System.out.println("Student inserted successfully.");
23     } catch (Exception e){
24         if (em.getTransaction().isActive()){
25             em.getTransaction().rollback();
26         }
27         e.printStackTrace();
28     } finally{
29         em.close();
30     }
31 }
32 }
```

update function in Student

```
public void updateStudent(Integer sid, String sfname, String slname, String semail, String sex) {
    EntityManager em = emf.createEntityManager();
    try {
        em.getTransaction().begin();
        Student student = em.find(Student.class, sid);
        if (student != null) {
            student.setSfname(sfname);
            student.setSlname(slname);
            student.setSemail(semail);
            student.setSex(sex);
            em.merge(student);
            System.out.println("Student updated successfully.");
        } else {
            System.out.println("Student not found.");
        }
        em.getTransaction().commit();
    } catch (Exception e) {
        if (em.getTransaction().isActive()){
            em.getTransaction().rollback();
        }
        e.printStackTrace();
    } finally {
        em.close();
    }
}
```

delete function in Student

```
58
59     public void deleteStudent(Integer studentId) {
60         EntityManager em = emf.createEntityManager();
61         try {
62             em.getTransaction().begin();
63             Student student = em.find(Student.class, studentId);
64             if (student != null) {
65                 em.remove(student);
66                 System.out.println("Student deleted successfully.");
67             } else {
68                 System.out.println("Student not found.");
69             }
70             em.getTransaction().commit();
71         } catch (Exception e) {
72             if (em.getTransaction().isActive()) {
73                 em.getTransaction().rollback();
74             }
75             e.printStackTrace();
76         } finally {
77             em.close();
78         }
79     }
80 }
81 }
```

Insert function in Section

```
public void insertSection(Integer cid, Date cdate, Date cstime, Date cetime, Integer subjectId) {
    EntityManager em = emf.createEntityManager();
    try{
        em.getTransaction().begin();

        Subject subject = em.find(Subject.class, subjectId);
        if(subject == null){
            System.out.println("Subject not found.");
            return;
        }

        Section section = new Section(cid, cdate, cstime, cetime);
        section.setSubjectSubjectId(subject);

        em.persist(section);
        em.getTransaction().commit();
        System.out.println("Section inserted successfully.");
    } catch (Exception e){
        if(em.getTransaction().isActive()){
            em.getTransaction().rollback();
        }
        e.printStackTrace();
    } finally{
        em.close();
    }
}
```

Update function in Section

```
public void updateSection(Integer cid, Date cdate, Date cstime, Date cetime, Integer subjectId){  
    EntityManager em = emf.createEntityManager();  
    try{  
        em.getTransaction().begin();  
        Section section = em.find(Section.class, cid);  
        if(section != null){  
  
            Subject subject = em.find(Subject.class, subjectId);  
            if(subject != null){  
                section.setSubjectSubjectId(subject);  
            }  
            section.setCdate(cdate);  
            section.setCstime(cstime);  
            section.setCetime(cetime);  
            em.merge(section);  
            em.getTransaction().commit();  
            System.out.println("Section updated successfully.");  
        } else {  
            System.out.println("Section not found.");  
        }  
    } catch (Exception e){  
        if(em.getTransaction().isActive()){  
            em.getTransaction().rollback();  
        }  
        e.printStackTrace();  
    } finally{  
        em.close();  
    }  
}
```

delete function in Section

```
public void deleteSection(Integer cid){  
    EntityManager em = emf.createEntityManager();  
    try{  
        em.getTransaction().begin();  
        Section section = em.find(Section.class, cid);  
        if(section != null){  
            em.remove(section);  
            em.getTransaction().commit();  
            System.out.println("Section deleted successfully.");  
        } else {  
            System.out.println("Section not found.");  
        }  
    } catch (Exception e){  
        if(em.getTransaction().isActive()){  
            em.getTransaction().rollback();  
        }  
        e.printStackTrace();  
    } finally{  
        em.close();  
    }  
}
```

Insert function in Subject

```
'

public void insertSubject(Integer subjectId, String subjectName, String subjectCode, int credithour) {
    EntityManager em = emf.createEntityManager();
    try{
        em.getTransaction().begin();
        Subject subject = new Subject(subjectId, subjectName, subjectCode, credithour);
        em.persist(subject);
        em.getTransaction().commit();
        System.out.println("Subject inserted successfully.");
    } catch (Exception e){
        if (em.getTransaction().isActive()){
            em.getTransaction().rollback();
        }
        e.printStackTrace();
    } finally {
        em.close();
    }
}
```

update function in Subject

```
}

public void updateSubject(Integer subjectId, String subjectName, String subjectCode, int credithour){
    EntityManager em = emf.createEntityManager();
    try{
        em.getTransaction().begin();
        Subject subject = em.find(Subject.class, subjectId);
        if(subject != null){
            subject.setSubjectName(subjectName);
            subject.setSubjectCode(subjectCode);
            subject.setCredithour(credithour);
            em.merge(subject);
            System.out.println("Subject updated successfully.");
        } else {
            System.out.println("Subject not found.");
        }
        em.getTransaction().commit();
    } catch (Exception e){
        if(em.getTransaction().isActive()){
            em.getTransaction().rollback();
        }
        e.printStackTrace();
    } finally{
        em.close();
    }
}
```

delete function in Subject

```
}

public void deleteSubject(Integer subjectId){
    EntityManager em = emf.createEntityManager();
    try{
        em.getTransaction().begin();
        Subject subject = em.find(Subject.class, subjectId);
        if (subject != null) {
            em.remove(subject);
            System.out.println("Subject deleted successfully.");
        } else {
            System.out.println("Subject not found.");
        }
        em.getTransaction().commit();
    } catch (Exception e){
        if (em.getTransaction().isActive()){
            em.getTransaction().rollback();
        }
        e.printStackTrace();
    } finally{
        em.close();
    }
}
```

Insert function in Student_Subject

```
}

public void insertStudentSubject(int studentSid, int subjectSubjectId, int id, BigDecimal degree){
    EntityManager em = emf.createEntityManager();
    try {
        em.getTransaction().begin();
        StudentSubjectPK studentSubjectPK = new StudentSubjectPK(studentSid, subjectSubjectId, id);
        StudentSubject studentSubject = new StudentSubject(studentSubjectPK, degree);
        em.persist(studentSubject);
        em.getTransaction().commit();
        System.out.println("Student-Subject inserted successfully.");
    } catch (Exception e) {
        if (em.getTransaction().isActive()) {
            em.getTransaction().rollback();
        }
        e.printStackTrace();
    } finally {
        em.close();
    }
}
```

update function in Student_Subject

```
}

public void updateStudentSubject(int studentSid, int subjectSubjectId, int id, BigDecimal degree) {
    EntityManager em = emf.createEntityManager();
    try{
        em.getTransaction().begin();
        StudentSubjectPK studentSubjectPK = new StudentSubjectPK(studentSid, subjectSubjectId, id);
        StudentSubject studentSubject = em.find(StudentSubject.class, studentSubjectPK);
        if (studentSubject != null){
            studentSubject.setDegree(degree);
            em.merge(studentSubject);
            System.out.println("Student-Subject updated successfully.");
        } else {
            System.out.println("Student-Subject not found.");
        }
        em.getTransaction().commit();
    } catch (Exception e){
        if (em.getTransaction().isActive()){
            em.getTransaction().rollback();
        }
        e.printStackTrace();
    } finally{
        em.close();
    }
}
```

Delete function in Student_Subject

```
}

public void deleteStudentSubject(int studentSid, int subjectSubjectId, int id) {
    EntityManager em = emf.createEntityManager();
    try {
        em.getTransaction().begin();
        StudentSubjectPK studentSubjectPK = new StudentSubjectPK(studentSid, subjectSubjectId, id);
        StudentSubject studentSubject = em.find(StudentSubject.class, studentSubjectPK);
        if (studentSubject != null) {
            em.remove(studentSubject);
            System.out.println("Student-Subject deleted successfully.");
        } else {
            System.out.println("Student-Subject not found.");
        }
        em.getTransaction().commit();
    } catch (Exception e) {
        if (em.getTransaction().isActive()) {
            em.getTransaction().rollback();
        }
        e.printStackTrace();
    } finally {
        em.close();
    }
}
```

Insert function in Attendance

```
}

public void insertAttendance(int studentSid, int sectionCid, int aid, String status){
    EntityManager em = emf.createEntityManager();
    try{
        em.getTransaction().begin();
        AttendancePK attendancePK = new AttendancePK(studentSid, sectionCid, aid);
        Attendance attendance = new Attendance(attendancePK, status);
        em.persist(attendance);
        em.getTransaction().commit();
        System.out.println("Attendance inserted successfully.");
    } catch (Exception e){
        if (em.getTransaction().isActive()){
            em.getTransaction().rollback();
        }
        e.printStackTrace();
    } finally{
        em.close();
    }
}
```

update function in Attendance

```
public void updateAttendance(int studentSid, int sectionCid, int aid, String status){  
    EntityManager em = emf.createEntityManager();  
    try{  
        em.getTransaction().begin();  
        AttendancePK attendancePK = new AttendancePK(studentSid, sectionCid, aid);  
        Attendance attendance = em.find(Attendance.class, attendancePK);  
        if (attendance != null){  
            attendance.setStatus(status);  
            em.merge(attendance);  
            System.out.println("Attendance updated successfully.");  
        } else {  
            System.out.println("Attendance not found.");  
        }  
        em.getTransaction().commit();  
    } catch (Exception e){  
        if (em.getTransaction().isActive()){  
            em.getTransaction().rollback();  
        }  
        e.printStackTrace();  
    } finally {  
        em.close();  
    }  
}
```

Delete function in Attendance

```
public void deleteAttendance(int studentSid, int sectionCid, int aid) {
    EntityManager em = emf.createEntityManager();
    try {
        em.getTransaction().begin();
        AttendancePK attendancePK = new AttendancePK(studentSid, sectionCid, aid);
        Attendance attendance = em.find(Attendance.class, attendancePK);
        if (attendance != null) {
            em.remove(attendance);
            System.out.println("Attendance deleted successfully.");
        } else {
            System.out.println("Attendance not found.");
        }
        em.getTransaction().commit();
    } catch (Exception e) {
        if (em.getTransaction().isActive()) {
            em.getTransaction().rollback();
        }
        e.printStackTrace();
    } finally {
        em.close();
    }
}
```

Queries

The screenshot shows the Apache NetBeans IDE 25 interface with a dark theme. The main window displays a Java file named `variousQuery.java`. The code implements several methods to query a database:

```
//List<Student> students = Q.getStudentsByAttendanceRange(startDate, endDate);  
43 //List<Student> students = Q.getStudentsBySubjectId(101);  
44 //List<Student> students = Q.getStudentsBySubjectIdAndSectionId(101, 204);  
45 List<Student> students = Q.getAllStudents();  
46 for(Student s : students)  
47 {  
48     System.out.println(s);  
49 }  
50 System.out.println("=====\n");  
51 students = Q.getMaleStudents();  
52 for(Student s : students)  
53 {  
54     System.out.println(s);  
55 }  
56  
57 }  
58 }  
59 }
```

The IDE's navigation pane on the left shows the project structure with files like `ProjectData.java` and `variousQuery.java`. The bottom status bar shows the current file is `main`, the encoding is `INS Unix (LF)`, and the system time is `10:11 PM 5/9/2025`.

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help ProjectData - Apache NetBeans IDE 25 Search (Ctrl+I) 424.9/542.0MB

ProjectData.java x variousQuery.java x

Source History

```
48     System.out.println(s);
49 }
50 System.out.println("=====\n");
51 students = Q.getMaleStudents();
52 for(Student s : students)
53 {
54     System.out.println(s);
55 }
```

Projects Files Services Navigator

com.mycompany.projectdata.ProjectData > main >

Output - Run (ProjectData) x

```
s1_0.sex=?  
com.mycompany.projectdata.Student[ sid=4 fname=mesh lname=Ahmed email=meshzizo@gmail.com sex=Male ]  
com.mycompany.projectdata.Student[ sid=6 fname=keko lname=badr email=badr@gmail.com sex=Male ]  
com.mycompany.projectdata.Student[ sid=7 fname=abdo lname=hamada email=abdo@gmail.com sex=Male ]  
-----  
BUILD SUCCESS  
-----  
Total time: 53.949 s
```

Type here to search

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help ProjectData - Apache NetBeans IDE 25 Search (Ctrl+I) <default config> 423.0/542.0MB

ProjectData.java x variousQuery.java x

Source History

```
48     System.out.println(s);
49 }
50 System.out.println("=====\\n");
51 students = Q.getMaleStudents();
52 for(Student s : students)
53 {
54     System.out.println(s);
55 }
56
57
58 }
```

com.mycompany.projectdata.ProjectData > main >

Output - Run (ProjectData) x

```
from
student s1_0
com.mycompany.projectdata.Student[ sid=1 fname=Rodina lname=Mohamed email=Roro@gmail.com sex=Female ]
com.mycompany.projectdata.Student[ sid=4 fname=mesh lname=Ahmed email=meshzizo@gmail.com sex=Male ]
com.mycompany.projectdata.Student[ sid=6 fname=keko lname=badr email=badr@gmail.com sex=Male ]
com.mycompany.projectdata.Student[ sid=7 fname=abdo lname=hamada email=abdo@gmail.com sex=Male ]
```

Type here to search 10:10 PM 5/9/2025

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help ProjectData - Apache NetBeans IDE 2.5 Search (Ctrl+I) 337 / 556.0MB

ProjectData.java x variousQuery.java

Source History

```
56     List<Subject> subjects = Q.getAllSubjects();
57     for (Subject s : subjects) {
58         System.out.println(s);
59     }
59     System.out.println("=====");
60     Subject sub = Q.getSubjectByCode("COMP207");
61     System.out.println(sub);
62
63 }
64
65 }
```

com.mycompany.projectdata.ProjectData > main >

Output - Run (ProjectData) x

```
subject s1_0
com.mycompany.projectdata.Student[ subjectId=101      name=Database    code=COMP101    Nohours=3 ]
com.mycompany.projectdata.Student[ subjectId=102      name=Computer Networks   code=CN102    Nohours=3 ]
com.mycompany.projectdata.Student[ subjectId=103      name=Operating System   code=Comp103    Nohours=3 ]
com.mycompany.projectdata.Student[ subjectId=207      name=Java        code=COMP207    Nohours=3 ]
=====
Hibernate:
    select
```

Type here to search

The screenshot shows the Apache NetBeans IDE 2.5 interface. The main window displays two Java files: `ProjectData.java` and `variousQuery.java`. The `variousQuery.java` file contains the following code:

```
56     List<Subject> subjects = Q.getAllSubjects();
57     for (Subject s : subjects) {
58         System.out.println(s);
59     }
59     System.out.println("=====");
60     Subject sub = Q.getSubjectByCode("COMP207");
61     System.out.println(sub);
62
63 }
64
65 }
```

The `Output - Run (ProjectData)` window shows the execution results:

```
from
subject s1_0
where
    s1_0.SubjectCode=?[{"subjectId":207,"name":"Java","code":"COMP207","Nohours":3}]
com.mycompany.projectdata.Student[ subjectId=207 name=Java code=COMP207 Nohours=3 ]
BUILD SUCCESS
```

The system tray at the bottom right indicates a temperature of 30°C, battery level at 60:67, and the date/time as 10:17 PM on 5/9/2025.

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help ProjectData - Apache NetBeans IDE 25 Search (Ctrl+I) 303.3/556.0MB

ProjectData.java X variousQuery.java X

Source History

```
//  
//  
System.out.println("=====");  
  
students = Q.getStudentsByAttendanceTimeRange(startTime, endTime);  
for (Student s : students) {  
    System.out.println(s);  
}  
}  
}  
}
```

com.mycompany.projectdata.ProjectData > main >

Output - Run (ProjectData) x

```
section ss  
    ON ss.Cid = a.Section_Cid  
    AND ss.Ctime BETWEEN ? AND ?  
com.mycompany.projectdata.Student[ sid=4      fname=mesh      lname=Ahmed      email=meshzizo@gmail.com      sex=Male ]  
  
BUILD SUCCESS  
  
Total time: 36.209 s
```

Type here to search

30°C صافي عالي 10:25 PM ENG 5/9/2025

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help ProjectData - Apache NetBeans IDE 25 Search (Ctrl+I) 3964/556.0MB

ProjectData.java x variousQuery.java

Source History

```
59     //}
60     //System.out.println("=====");
61     //Subject sub = Q.getSubjectByCode("COMP207");
62     //System.out.println(sub);
63
64     students = Q.getStudentsBySubjectId(101);
65     for (Student s : students) {
66         System.out.println(s);
67     }
68     //System.out.println("=====");
```

com.mycompany.projectdata.ProjectData > main >

Output - Run (ProjectData) x

```
-->
      student s
      JOIN
      student_subject ss
      ON s.Sid = ss.Student_Sid
      WHERE
      ss.Subject_SubjectId = ?
com.mycompany.projectdata.Student[ sid=6          fname=keko          lname=badr          email=badr@gmail.com          sex=Male ]
```

Type here to search 29°C 10:36 PM 5/9/2025

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help ProjectData - Apache NetBeans IDE 25 Search (Ctrl+I) 3028/556.0MB

ProjectData.java x variousQuery.java x

Source History

```
61     //Subject sub = Q.getSubjectByCode("COMP207");
62     //System.out.println(sub);
63
64     students = Q.getStudentsBySubjectIdAndsectionId(101, 204);
65     for (Student s : students) {
66         System.out.println(s);
67     }
68     //System.out.println("=====");
69
70     //students = Q.getStudentsByAttendanceTimeRange(startTime, endTime);
71     //for (Student s : students) {
```

com.mycompany.projectdata.ProjectData > main >

Output - Run (ProjectData) x

```
section ss
  ON ss.Cid = ?
  AND ss.Subject_SubjectId = ?
com.mycompany.projectdata.Student[ sid=1      fname=Rodina      lname=Mohamed      email=Roro@gmail.com      sex=Female ]
com.mycompany.projectdata.Student[ sid=4      fname=mesh      lname=Ahmed      email=meshzizo@gmail.com      sex=Male ]
com.mycompany.projectdata.Student[ sid=6      fname=keko      lname=badr      email=badr@gmail.com      sex=Male ]
com.mycompany.projectdata.Student[ sid=7      fname=abdo      lname=hamada      email=abdo@gmail.com      sex=Male ]
```

Type here to search 10:39 PM 5/9/2025

USD/ILS 0,71- ENG