Database Administration and Management

Members:

- **Daud Mustafa F2021105159**
- **Zainab Zahid F2022105062**
- **Muhammad Umair F2021105152**
- **Zunnorain F2021105230**
- **Fatima Shoukat Ali F2021105217**

Project Title: Student Information System

Submitted To: Sir Raybal Akhtar .

Section: Y3.

Date: 22/01/2024.

Department of Sst.

1. DBA Policy, Standards, Procedure for Database Security and Database Quality Management:

i. Database Security:

Developing effective security policies to protect sensitive information in databases that are up to industrial standards and regulations.

Define access controls for users to gain data confidentiality and integrity.

Scheduled review and auditing of database security measures, so that any vulnerabilities can be addressed beforehand.

ii. Database Quality Management:

Create data quality standards and procedures so that the accuracy and consistency of Information can be stored in the database.

Prevent data inconsistencies and error via data validation rules.

Work with stakeholders to define and enforce data quality standards that will live out the database life cycle.

2. DBA Roles and Responsibilities (ERD):

i. Student Table:

Overseeing the data integrity, indexing, and optimization to ensure that data stored in the table is up to defined quality standards.

ii. Course Table:

Optimize the structure and performance of the Course table by implementing data quality measures.

iii. Enrollment Table:

Prioritize the maintenance of data integrity and optimizing queries, and collaborate with the stakeholders to address any errors related to enrollment data.

iv. Fee Table:

Ensure accurate recording and processing of fee-related information. Furthermore, implement measures to protect sensitive fee data.

v. Assessment Table:

Optimize the structure of assessment table for efficient data retrieval.

vi. Relationships:

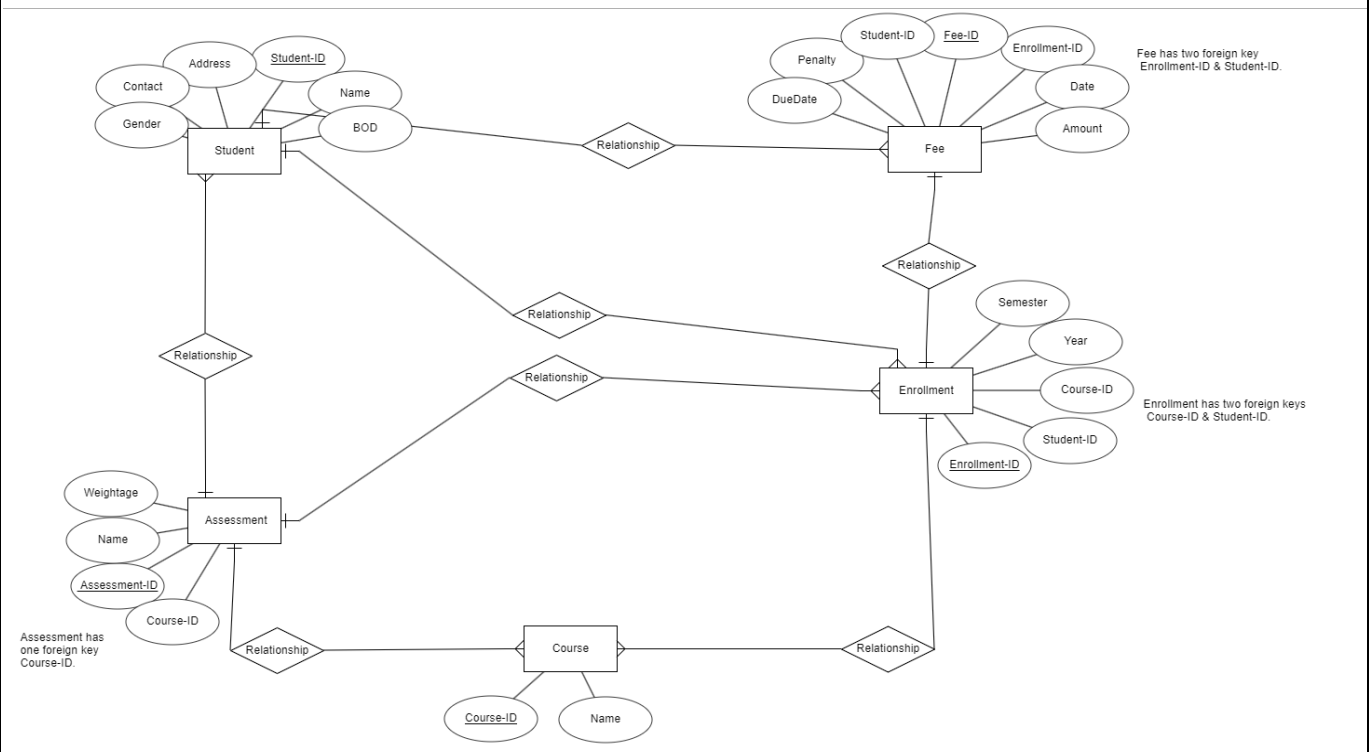
Manage the one-to-many relationship between student and enrollment, student and fee, course and Enrollment, Course and Assessment.

Oversee the Many-to-Many relationship between Enrollment and Assessment, ensuring complete association and disassociation of records.

Conclusion

The DBA will play an essential role in maintaining the security, quality, integrity, and overall health of the database. However, this doesn't exempt him from the specific responsibilities related to each table and the defined relationships with ERD.

ERD



PROJECT SCHEMA

```sql
--schema

CREATE TABLE Student (
    StudentID INT PRIMARY KEY,
    Name VARCHAR(100),
    Address VARCHAR(255),
    Contact VARCHAR(15),
    Gender CHAR(1),
    DateOfBirth DATE
);

drop table Student;

CREATE TABLE Course (
    CourseID INT PRIMARY KEY,
    Name VARCHAR(100)
);

drop table Course;

CREATE TABLE Enrollment (
    EnrollmentID INT PRIMARY KEY,
    StudentID INT,
    FOREIGN KEY (StudentID) REFERENCES Student(StudentID),
    CourseID INT,
    Semester VARCHAR(10),
    Years INT,
    FOREIGN KEY (CourseID) REFERENCES Course(CourseID)
);

drop table Enrollment;

drop table Fee;
CREATE TABLE Fee (
    FeeID INT PRIMARY KEY,
    StudentID INT,
    EnrollmentID INT,
    Amount DECIMAL(10, 2),
    Date DATE,
    DueDate DATE,
    Penalty DECIMAL(10, 2),
    FOREIGN KEY (StudentID) REFERENCES Student(StudentID),
    FOREIGN KEY (EnrollmentID) REFERENCES Enrollment(EnrollmentID)
);

drop table Assessment;

CREATE TABLE Assessment (
    AssessmentID INT PRIMARY KEY,
    CourseID INT,
    Name VARCHAR(100),
    Weightage INT,
    FOREIGN KEY (CourseID) REFERENCES Course(CourseID)
);

-- Inserting into student table--
```

```sql
INSERT INTO Student (StudentID, Name, Address, Contact, Gender, DateOfBirth)
VALUES
('2021', 'John Doe', '123 Main St, Lahore', '+923001234567', 'M', '2000-01-01'),
('2022', 'Jane Smith', '456 Elm St, Karachi', '+923009876543', 'F', '2000-02-02'),
('2023', 'Ahmed Khan', '789 Pine St, Islamabad', '+923004321098', 'M', '2000-03-03'),
('2024', 'Sara Ali', '321 Oak St, Multan', '+923007654321', 'F', '2000-04-04'),
('2025', 'Muhammad Ali', '654 Maple St, Faisalabad', '+923002109876', 'M', '2000-05-05');
select * from Student ;
```

33 %

Results | Messages

	StudentID	Name	Address	Contact	Gender	DateOfBirth
1	2022	Jane Smith	456 Elm St, Karachi	+923009876543	F	2000-02-02
2	2023	Ahmed Khan	789 Pine St, Islamabad	+923004321098	M	2000-03-03
3	2024	Sara Ali	321 Oak St, Multan	+923007654321	F	2000-04-04
4	2025	Muhammad Ali	654 Maple St, Faisalabad	+923002109876	M	2000-05-05

```sql
-- Inserting into Course table
INSERT INTO Course (CourseID, Name)
VALUES
(1, 'Computer Science'),
(2, 'Physics'),
(3, 'Chemistry'),
(4, 'Biology'),
(5, 'Mathematics');
select * from Course;
```

33 %

Results | Messages

	CourseID	Name
1	1	Computer Science
2	2	Physics
3	3	Chemistry
4	4	Biology
5	5	Mathematics

```sql
-- Inserting into Enrollment table with soft delete flag
INSERT INTO Enrollment (EnrollmentID, StudentID, CourseID, Semester, Years)
VALUES
(1, 2021, 1, 'Fall', 2024),
(2, 2022, 2, 'Spring', 2024),
(3, 2023, 3, 'Fall', 2024),
(4, 2024, 4, 'Spring', 2024),
(5, 2025, 5, 'Fall', 2024);
select * from Enrollment;
```

Results | Messages

	EnrollmentID	StudentID	CourseID	Semester	Years
1	1	2021	1	Fall	2024
2	2	2022	2	Spring	2024
3	3	2023	3	Fall	2024
4	4	2024	4	Spring	2024
5	5	2025	5	Fall	2024

```sql
-- Inserting into Fee table
INSERT INTO Fee (FeeID, StudentID, EnrollmentID, Amount, Date, DueDate, Penalty)
VALUES
(1, '2021', 1, 10000.00, '2024-01-01', '2024-02-01', 1000.00),
(2, '2022', 2, 50000.00, '2024-02-02', '2024-03-02', 5000.00),
(3, '2023', 3, 30000.00, '2024-03-03', '2024-04-03', 3000.00),
(4, '2024', 4, 80000.00, '2024-04-04', '2024-05-04', 6000.00),
(5, '2025', 5, 50000.00, '2024-05-05', '2024-06-05', 5000.00);
select * from Fee;
```

Results | Messages

	FeeID	StudentID	EnrollmentID	Amount	Date	DueDate	Penalty
1	1	2021	1	10000.00	2024-01-01	2024-02-01	1000.00
2	2	2022	2	50000.00	2024-02-02	2024-03-02	5000.00
3	3	2023	3	30000.00	2024-03-03	2024-04-03	3000.00
4	4	2024	4	80000.00	2024-04-04	2024-05-04	6000.00
5	5	2025	5	50000.00	2024-05-05	2024-06-05	5000.00

```sql
-- Inserting into Assessment table
INSERT INTO Assessment (AssessmentID, CourseID, Name, Weightage)
VALUES
(101, 1, 'Midterm Exam', 50),
(102, 2, 'Final Exam', 50),
(103, 3, 'Project', 30),
(104, 4, 'Quiz', 20),
(105, 5, 'Assignment', 10);

select * from Assessment;
```

Results | Messages

	AssessmentID	CourseID	Name	Weightage
1	101	1	Midterm Exam	50
2	102	2	Final Exam	50
3	103	3	Project	30
4	104	4	Quiz	20
5	105	5	Assignment	10

Store Procedures

```
/* creating stored procedures that will show
enrolled course and student id */
create procedure colldata
as
begin
select
enrollment.studentid,Student.Name,Enrollment.CourseID,Course.Name
from Enrollment
inner join
student on Enrollment.StudentID=student.StudentID
inner join
course on enrollment.CourseID=Course.CourseID
end
```

```
Commands completed successfully.
Completion time: 2024-01-20T01:10:02.6442601-08:00
```

```
exec colldata
```

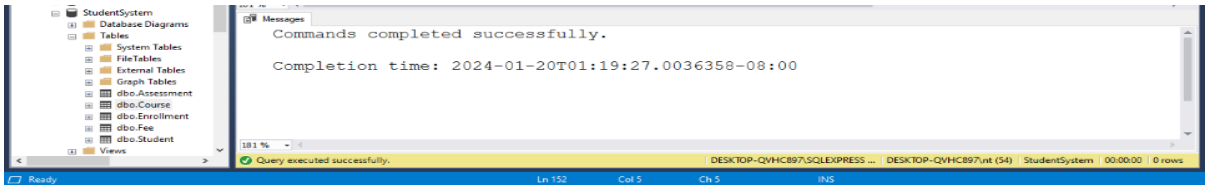
Results | Messages

	studentid	Name	CourseID	Name
1	2021	John Doe	1	Computer Science
2	2022	Jane Smith	2	Physics
3	2023	Ahmed Khan	3	Chemistry
4	2024	Sara Ali	4	Biology
5	2025	Muhamma...	5	Mathematics

Query executed successfully. DESKTOP-9NLUTM5\SQLEXPRESS ...

```
/*Creating stored procedure that will show the courses
with student id can find the course through giving
specific student ID*/
```

```sql
CREATE PROCEDURE EnrolledCourses
    @StudentID INT
AS
BEGIN
    SELECT
        Enrollment.StudentID,
        Student.Name AS StudentName,
        Enrollment.CourseID,
        Course.Name AS CourseName
    FROM
        Enrollment
    INNER JOIN
        Student ON Enrollment.StudentID = Student.StudentID
    INNER JOIN
        Course ON Enrollment.CourseID = Course.CourseID
    WHERE
        Enrollment.StudentID = @StudentID;
END;
```

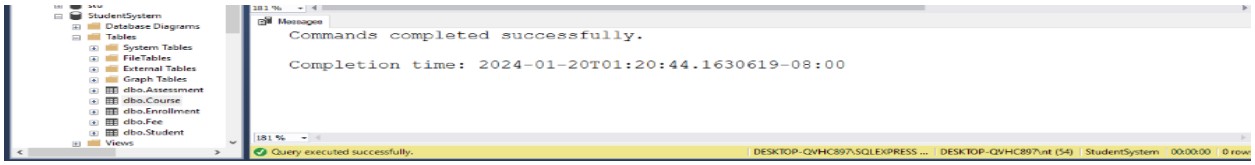



```sql
EXEC EnrolledCourses @StudentID = 2023;
```

Results | Messages

	StudentID	StudentName	CourseID	CourseName
1	2023	Ahmed Khan	3	Chemistry

```sql
/*all Data of enrollment table*/
create procedure ALLDATA
as
begin
select *from Enrollment
end
```



```
exec ALLDATA
```

Results | Messages

	EnrollmentID	StudentID	CourseID	Semester	Years
1	1	2021	1	Fall	2024
2	2	2022	2	Spring	2024
3	3	2023	3	Fall	2024
4	4	2024	4	Spring	2024
5	5	2025	5	Fall	2024

Views

```
create view studentData
as
select Student.StudentID,Student.Name,Enrollment.CourseID,Enrollment.Semester from

Student join Enrollment on Student.StudentID=Enrollment.StudentID
```

133 %

Messages

```
Commands completed successfully.

Completion time: 2024-01-20T01:23:10.8332738-08:00
```

133 %

Query executed successfully. | DESKTOP-F1C39TI\SQLEXPRESS ... | DESKTOP-F1C39TI\AI haf... | StudentSystem | 00:00:00 | 0

```
select * from studentData
```

133 %

Results | Messages

	StudentID	Name	CourseID	Semester	Year
1	2021	John Doe	1	Fall	2024
2	2022	Jane Smith	2	Spring	2024
3	2023	Ahmed Khan	3	Fall	2024
4	2024	Sara Ali	4	Spring	2024
5	2025	Muhammad Ali	5	Fall	2024

Query executed successfully. | DESKTOP-F1C39TI\SQLEXPRESS ... | DESKTOP-F1C39TI\AI haf... | StudentSystem | 00:00:00 | 5 rows

```
alter view studentData
as
select Student.StudentID,Student.Name,Enrollment.CourseID,Enrollment.Semester, Enrollment.Year
from
Student join Enrollment on Student.StudentID=Enrollment.StudentID
```

Messages

```
Commands completed successfully.

Completion time: 2024-01-20T01:24:16.0603931-08:00
```

133 %

Query executed successfully. | DESKTOP-F1C39TI\SQLEXPRESS ... | DESKTOP-F1C39TI\AI haf... | StudentSystem | 00:00:00 | 0 row

```
select * from studentData
```

133 %

Results | Messages

	StudentID	Name	CourseID	Semester	Year
1	2021	John Doe	1	Fall	2024
2	2022	Jane Smith	2	Spring	2024
3	2023	Ahmed Khan	3	Fall	2024
4	2024	Sara Ali	4	Spring	2024
5	2025	Muhammad Ali	5	Fall	2024

Query executed successfully. | DESKTOP-F1C39TI\SQLEXPRESS ... | DESKTOP-F1C39TI\AI haf... | StudentSystem | 00:00:00 | 5 rows

```
create view CO_AS
with schemabinding
as
select Assessment.Weightage from
dbo.Assessment join dbo.Course on Assessment.CourseID=Course.CourseID
```

133 %

Messages

```
Commands completed successfully.

Completion time: 2024-01-20T01:28:44.4779583-08:00
```

133 %

Query executed successfully. | DESKTOP-F1C39TI\SQLEXPRESS ... | DESKTOP-F1C39TI\AI haf... | StudentSystem | 00:00:00 | 0 rows

```
select * from CO_AS
```

133 %

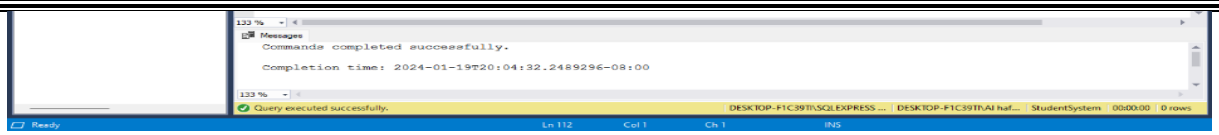
Results | Messages

	Weightage
1	50
2	50
3	30
4	20
5	10

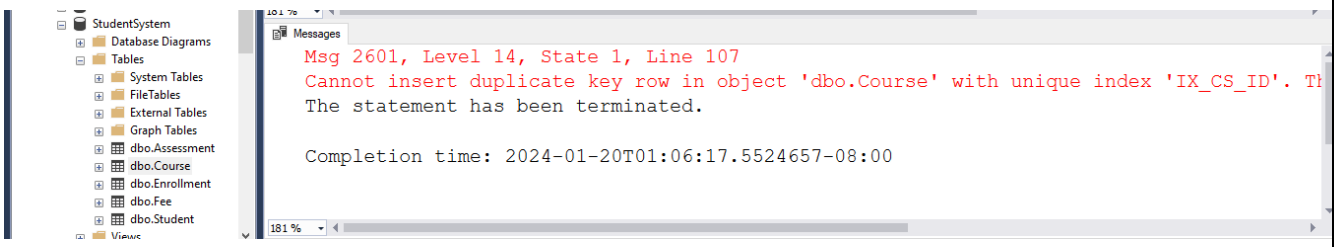
Query executed successfully. | DESKTOP-F1C39TI\SQLEXPRESS ... | DESKTOP-F1C39TI\AI haf... | StudentSystem | 00:00:00 | 5 rows

Indexes

```
create nonclustered index IX_tblEnrollment_Semester
on Enrollment(Semester)
```

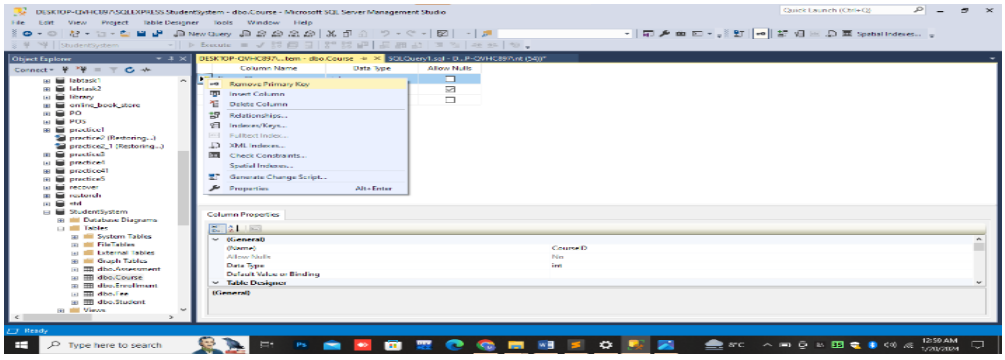



```
/*Finding the id of COURSE using clustered index*/
create clustered index IX_Course_ID
on Course(CourseID);
```

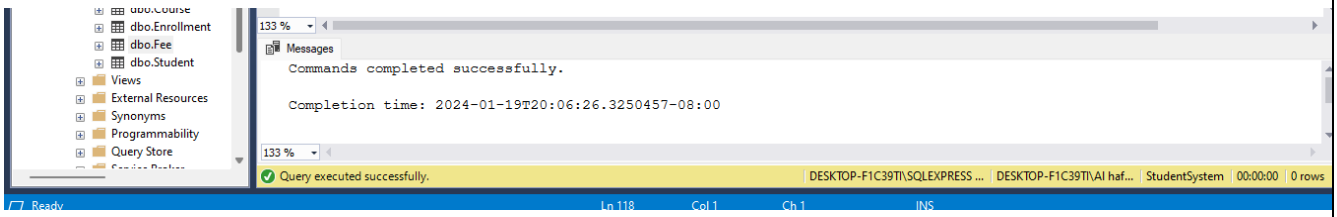


Removing The Error

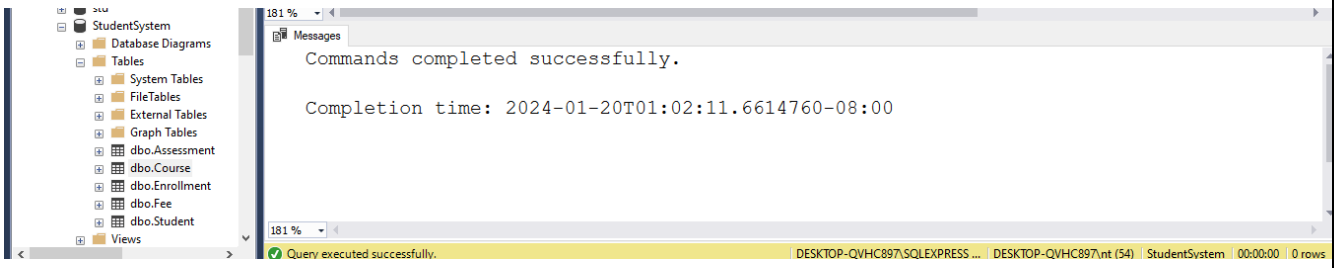
```
/*Finding the id of COURSE using clustered index*/
create clustered index IX_Course_ID
on Course(CourseID);
```



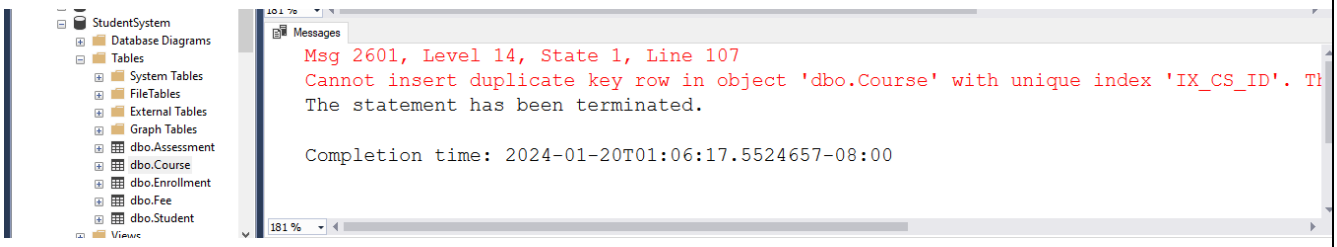
```
create unique index UIX_tblStudent_StudentID_Name
on Student(StudentID,Name)
```



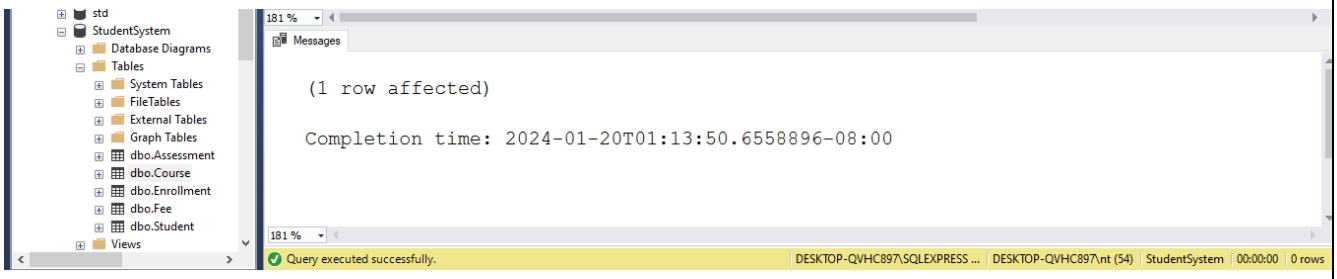
```sql
/* Creating unique index */

create unique nonclustered index IX_CS_ID
on Course( CourseID);
```




```sql
INSERT INTO Course (CourseID, Name)
VALUES
(3, 'Chemistry')
```




```sql
INSERT INTO Course (CourseID, Name)
VALUES
(7, 'Chemistry')
```



DML TRIGGERS

WITH THE CONCEPT OF SOFT DELETE

```
-- Usage of DML Triggers (as per your project idea) for maintaining the integrity of the
--information on the database.

drop trigger studentDuplicateID;

CREATE TRIGGER studentDuplicateID
ON Student
INSTEAD OF INSERT
AS
BEGIN
    DECLARE @existingCount INT;

    SELECT @existingCount = COUNT(*)
    FROM Student
    WHERE StudentID IN (SELECT StudentID FROM INSERTED);

    IF @existingCount > 1
    BEGIN
        THROW 50000, 'Duplicate StudentID. Cannot insert.', 1;
    END
END;


-- to check the trigger activity Inserting some duplicate data into student table
INSERT INTO Student (StudentID, Name, Address, Contact, Gender, DateOfBirth)
VALUES
('2021', 'New Student', '789 New St, Karachi', '+923005678901', 'F', '2000-07-07');

select * from Student;
```

```
%
Messages
Msg 50000, Level 16, State 1, Procedure studentDuplicateID, Line 15 [Batch Start Line 47]
Duplicate StudentID. Cannot insert.

Completion time: 2024-01-20T18:35:58.0818153+05:00
```

```
drop trigger StudentAuditTrail;
--student audit detials
CREATE TRIGGER StudentAuditTrail
ON Student
AFTER INSERT, UPDATE, DELETE
AS
BEGIN
    INSERT INTO StudentAudit (ActionType, StudentID, Name, ModifiedDate)
```

```sql
    SELECT
        CASE
            WHEN EXISTS (SELECT * FROM INSERTED) AND EXISTS (SELECT * FROM DELETED) THEN 'Update'
            WHEN EXISTS (SELECT * FROM INSERTED) THEN 'Insert'
            WHEN EXISTS (SELECT * FROM DELETED) THEN 'Delete'
        END,
        COALESCE(i.StudentID, d.StudentID),
        COALESCE(i.Name, d.Name),
        GETDATE()
    FROM INSERTED i
    FULL OUTER JOIN DELETED d ON i.StudentID = d.StudentID;
END;

create table StudentAudit(
 ActionType varchar(255),
 StudentID int ,
 FOREIGN KEY (StudentID) REFERENCES Student(StudentID),
 Name varchar(255),
 ModifiedDate varchar(255)
);

drop table StudentAudit;

select * from StudentAudit;

-- Insert a new student
INSERT INTO Student (StudentID, Name, Address, Contact, Gender, DateOfBirth)
VALUES ('2026', 'New Student', '123 Test St', '+1234567890', 'M', '2000-01-01');
```

Results | Messages

ActionType	StudentID	Name	ModifiedDate
Update	2026	Updated Student	Jan 20 2024 7:05PM

```sql
-- Update an existing student
UPDATE Student
SET Name = 'Updated Student'
WHERE StudentID = '2026';

-- Delete a student
DELETE FROM Student
WHERE StudentID = '2026';

--soft delete


CREATE TABLE softDeleteStudent (
    StudentID INT PRIMARY KEY,
    Name VARCHAR(100),
    Address VARCHAR(255),
    Contact VARCHAR(15),
    Gender CHAR(1),
    DateOfBirth DATE
);

-- Create a trigger for soft delete
CREATE TRIGGER SoftDeleteStudentTrigger
```

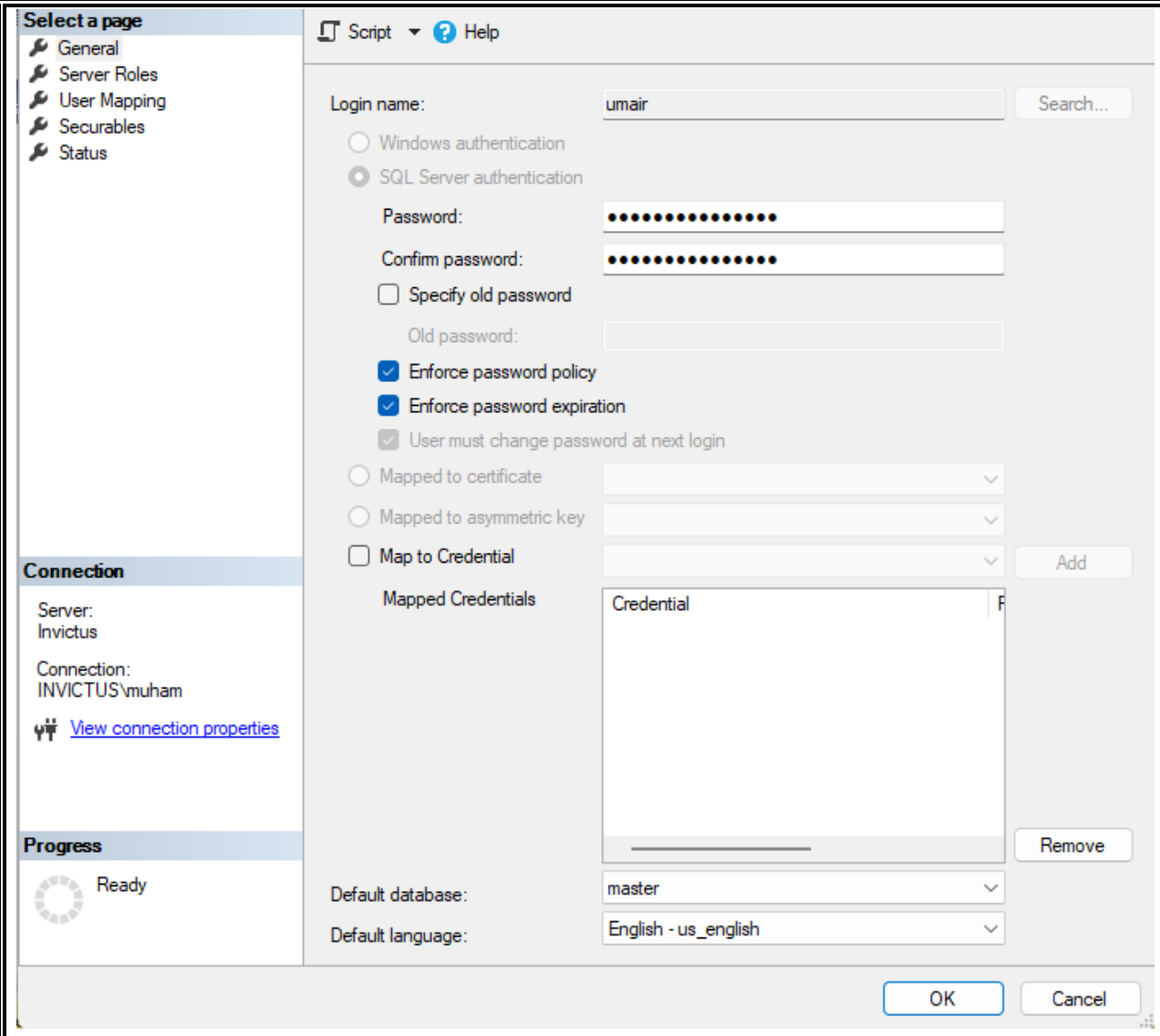
```sql
ON Student
INSTEAD OF DELETE
AS
BEGIN
    -- Insert the deleted records into the clone table
    INSERT INTO softDeleteStudent
    SELECT * FROM deleted;

    DELETE FROM Student
    WHERE StudentID IN (SELECT StudentID FROM deleted);
END;
-- Delete a record from the Student table (soft delete)
DELETE FROM Student
WHERE StudentID = '2021';
select * from softDeleteStudent;
```

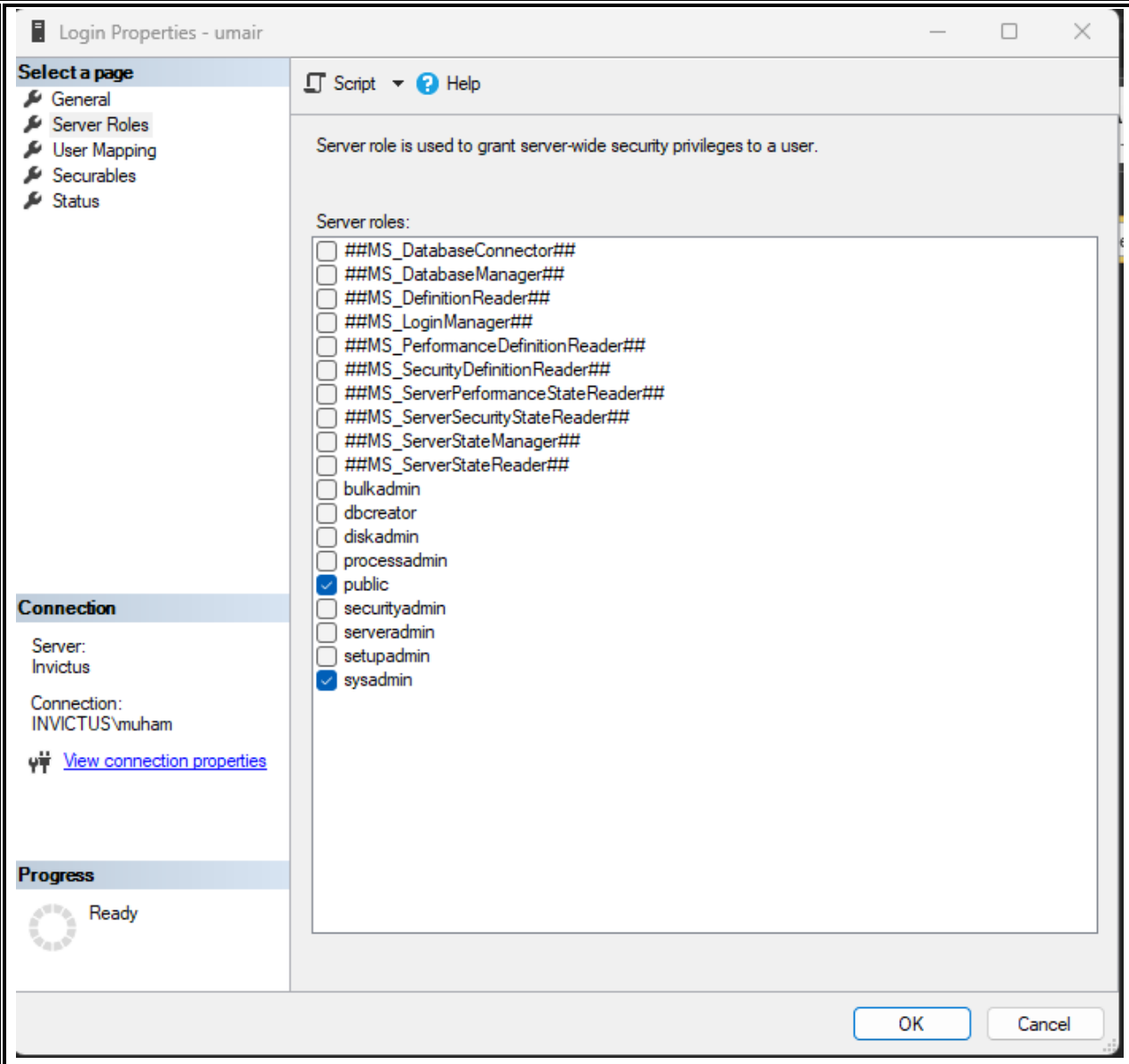
%

Results | Messages

StudentID	Name	Address	Contact	Gender	DateOfBirth
2021	John Doe	123 Main St, Lahore	+923001234567	M	2000-01-01

User Creation:


Select a page
General
Server Roles
User Mapping
Securables
Status
Script
Help
Login name:
umair
Search...
Windows authentication
SQL Server authentication
Password:
Confirm password:
Specify old password
Old password:
Enforce password policy
Enforce password expiration
User must change password at next login
Mapped to certificate
Mapped to asymmetric key
Map to Credential
Add
Mapped Credentials
Credential
Remove
Connection
Server:
Invictus
Connection:
INVICTUS\muham
View connection properties
Progress
Ready
Default database:
master
Default language:
English - us_english
OK
Cancel


Login Properties - umair
Select a page
General
Server Roles
User Mapping
Securables
Status
Script
Help
Server role is used to grant server-wide security privileges to a user.
Server roles:
##MS_DatabaseConnector##
##MS_DatabaseManager##
##MS_DefinitionReader##
##MS_LoginManager##
##MS_PerformanceDefinitionReader##
##MS_SecurityDefinitionReader##
##MS_ServerPerformanceStateReader##
##MS_ServerSecurityStateReader##
##MS_ServerStateManager##
##MS_ServerStateReader##
bulkadmin
dbcreator
diskadmin
processadmin
public
securityadmin
serveradmin
setupadmin
sysadmin
Connection
Server:
Invictus
Connection:
INVICTUS\muham
View connection properties
Progress
Ready
OK
Cancel

Login Properties - umair

Select a page

- General
- Server Roles
- User Mapping
- Securables
- Status

Script Help

Settings

Permission to connect to database engine:

- (•) Grant
- ( ) Deny

Login:

- (•) Enabled
- ( ) Disabled

Status

SQL Server authentication:

- [ ] Login is locked out

Connection

Server:
Invictus

Connection:
INVICTUS\muham

View connection properties

Progress

Ready

OK Cancel

Login Properties - umair

Select a page

- General
- Server Roles
- User Mapping
- Securables
- Status

Script Help

Users mapped to this login:

Map	Database	User	Default Schema
☐	f2021105152		
☐	f2021105162		
☑	finalproject	umair	
☐	master		
☐	model		
☐	msdb		
☐	task7		
☐	tempdb		

☐ Guest account enabled for: finalproject

Database role membership for: finalproject

- ☐ db_accessadmin
- ☐ db_backupoperator
- ☐ db_datareader
- ☐ db_datawriter
- ☐ db_ddladmin
- ☐ db_denydatareader
- ☐ db_denydatawriter
- ☐ db_owner
- ☐ db_securityadmin
- ☑ public

Connection

Server:
Invictus

Connection:
INVICTUS\muham

View connection properties

Progress

Ready

OK Cancel

-- In case queries--

```
--To retrieves the list of students enrolled in a particular course--
CREATE PROCEDURE GetEnrolledStudentsInCourse
@CourseID INT
AS
BEGIN
   SELECT Student.StudentID, Student.Name
   FROM Student
   INNER JOIN Enrollment ON Student.StudentID = Enrollment.StudentID
   WHERE Enrollment.CourseID = @CourseID;
END;
```

121 %

Messages

```
Commands completed successfully.

Completion time: 2024-01-22T20:34:54.7930171+05:00
```

```
EXEC GetEnrolledStudentsInCourse @CourseID = 1;
```

121 %

Results | Messages

	StudentID	Name
1	2021	John Doe

```
--A view to combines information from the Course and Enrollment tables--
CREATE VIEW CourseData
AS
SELECT
   Course.CourseID,
   Course.Name,
   Enrollment.StudentID,
   Student.Name AS StudentName,
   Enrollment.Semester
FROM
```

```
    Course
JOIN
    Enrollment ON Course.CourseID = Enrollment.CourseID
JOIN
    Student ON Enrollment.StudentID = Student.StudentID;
SELECT * FROM CourseData;
```

121 %

Results | Messages

	CourseID	Name	StudentID	StudentName	Semester
1	1	Computer Science	2021	John Doe	Fall
2	2	Physics	2022	Jane Smith	Spring
3	3	Chemistry	2023	Ahmed Khan	Fall
4	4	Biology	2024	Sara Ali	Spring
5	5	Mathematics	2025	Muhammad Ali	Fall