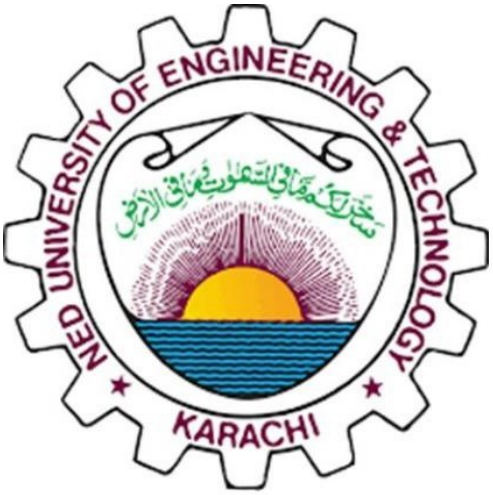
NED University of Engineering & Technology



"ACADEMIC MANAGEMENT SYSTEM"

GROUP MEMBERS:

***MUHAIB SHAMSHER** (CR-029)*

***S. GHUFRAN RAZA** (CR-028)*

***HASNAIN RAZA KHAN** (CR-034)*

Instructor: *Ms. Nudrat Naved*

Database Management System

(CT-261)

Department Of Computer Science & Information Technology

(Specialization In Cyber Security)

1)DESCRIPTION:

Project Description: Academic Management System

This database project aims to create a comprehensive Student Enrollment and Course Management System for an educational institution. The system manages students, courses, instructors, departments, enrollments, grades, and course details. Here's an overview of its components:

- Students Table: Stores student information including ID, name, date of birth, gender, enrollment date, email, and phone number.

- Courses Details Table: Contains details about courses such as course ID, name, code, description, and the department it belongs to.

- Departments Table: Lists different academic departments within the institution, each identified by a unique department ID and named accordingly.

- Instructors Table: Stores details of instructors who teach courses, including their ID, name, contact information (email and phone), hire date, and their associated department.

- Course Instructors Table: Links courses to instructors by storing pairs of course IDs and instructor IDs, establishing which instructors teach which courses.

- Enrollments Table: Manages student enrollments into courses by storing enrollment IDs along with student IDs, course IDs, and enrollment dates.

- Grades Table: Tracks grades awarded to students for enrolled courses, linked to enrollment IDs and storing the actual grade.

The system supports complex queries and operations such as:

- Advanced Queries: Retrieving detailed information about students, courses, instructors, departments, and their relationships.

- Data Manipulation: Updating instructor details, modifying course descriptions, and handling changes in student enrollment statuses.

- Data Integrity: Ensuring data consistency through proper use of foreign key constraints and transaction management.

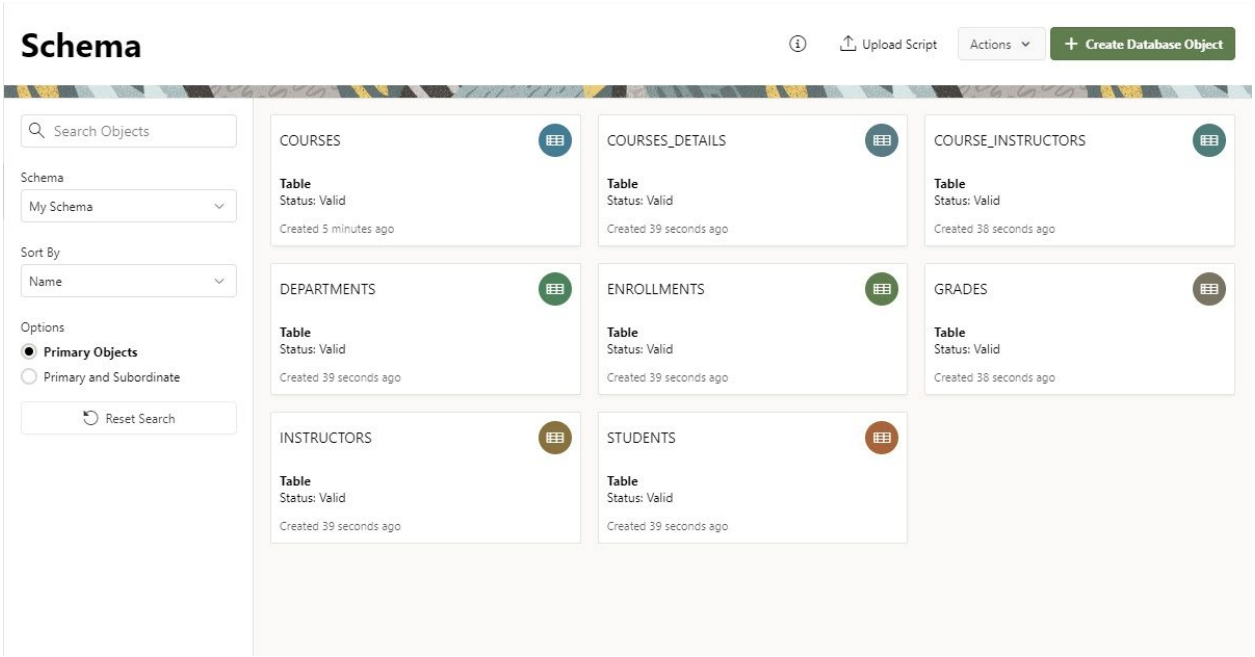
- Reporting: Generating reports on student performance, course popularity, instructor workloads, and departmental statistics.

This Student Enrollment and Course Management System enhances administrative efficiency, improves data accessibility, and supports informed decision-making within the educational institution.

2) Relational Schema:

Students	(**student_id** , first_name, last_name, date_of_birth, gender, enrollment_date, email, phone)
Courses_details	(**course_id** , course_name, course_code, course_description, department_id)
Courses	(**course_code**, credits, semester_offered)
Instructors	(**instructor_id**, first_name, last_name, email, phone, hire_date, department_id)
Departments	(**department_id** , department_name)
Enrollments	(**enrollment_id** , student_id , course_id, enrollment_date)
Grades	(**grade_id** , enrollment_id, grade)
Course_Instructors	(**course_id** , **instructor_id**)

3) Schema:



Queries:

1) **To insert the student data in Students table**.

```
INSERT INTO Students (student_id, first_name, last_name, date_of_birth, gender, enrollment_date, email, phone)
VALUES
(1, 'Jane', 'Doe', '1995-03-15', 'F', '2023-09-01', 'jane.doe@example.com', '9876543210'),
(2, 'John', 'Brown', '1996-05-10', 'M', '2023-08-02', 'john.brown@example.com', '3456789012'),
(3, 'Hamid', 'Ali', '1997-11-12', 'M', '2023-05-28', 'hamid.ali@example.com', '5678901234'),
(5, 'Emily', 'Robinson', '1996-06-15', 'F', '2023-09-02', 'emily.robinson@example.com', '5678901234'),
(6, 'Frank', 'Patel', '1999-03-20', 'M', '2023-09-03', 'frank.patel@example.com', '6789012345'),
(7, 'Henry', 'Anderson', '1997-09-22', 'F', '2023-09-04', 'henry.anderson@example.com', '8901234567'),
(8, 'Isabella', 'Garcia', '1996-03-08', 'M', '2023-09-05', 'isabella.garcia@example.com', '9012345678'),
(9, 'Jack', 'Thompson', '1998-07-14', 'F', '2023-09-06', 'jack.thompson@example.com', '1230123012'),
(10, 'Katherine', 'Harris', '1995-01-25', 'M', '2023-09-07', 'katherine.harris@example.com', '2341234123'),
(11, 'Liam', 'Martin', '1997-12-03', 'F', '2023-09-08', 'liam.martin@example.com', '3452345234'),
(12, 'Mia', 'Rodriguez', '1996-05-20', 'M', '2023-09-09', 'mia.rodriguez@example.com', '4563456345'),
(13, 'Noah', 'Young', '1999-04-18', 'F', '2023-09-10', 'noah.young@example.com', '5674567456'),
(14, 'Olivia', 'Lee', '1994-08-07', 'M', '2023-09-11', 'olivia.lee@example.com', '6785678567');
```

STUDENT_ID	FIRST_NAME	LAST_NAME	DATE_OF_BIRTH	GENDER	ENROLLMENT_DATE	EMAIL	PHONE
2	John	Brown	10-MAY-96	M	02-AUG-23	john.brown@example.com	3456789012
1	Jane	Doe	15-MAR-95	F	01-SEP-23	jane.doe@example.com	9876543210
3	Hamid	Ali	12-NOV-97	M	28-MAY-23	hamid.ali@example.com	5678901234
5	Emily	Robinson	15-JUN-96	F	02-SEP-23	emily.robinson@example.com	5678901234
6	Frank	Patel	20-MAR-99	M	03-SEP-23	frank.patel@example.com	6789012345
7	Henry	Anderson	22-SEP-97	F	04-SEP-23	henry.anderson@example.com	8901234567
8	Isabella	Garcia	08-MAR-96	M	05-SEP-23	isabella.garcia@example.com	9012345678
9	Jack	Thompson	14-JUL-98	F	06-SEP-23	jack.thompson@example.com	1230123012
10	Katherine	Harris	25-JAN-95	M	07-SEP-23	katherine.harris@example.com	2341234123
11	Liam	Martin	03-DEC-97	F	08-SEP-23	liam.martin@example.com	3452345234
12	Mia	Rodriguez	20-MAY-96	M	09-SEP-23	mia.rodriguez@example.com	4563456345
13	Noah	Young	18-APR-99	F	10-SEP-23	noah.young@example.com	5674567456
14	Olivia	Lee	07-AUG-94	M	11-SEP-23	olivia.lee@example.com	6785678567

2) Delete query scenario involving multiple tables.

```
DELETE FROM grades
WHERE enrollment_id IN (
   SELECT enrollment_id
   FROM enrollments
   WHERE student_id = 1
);

DELETE FROM enrollments
WHERE student_id = 1;

DELETE FROM students
WHERE student_id = 1;
```

```
10  DELETE FROM grades
11  WHERE enrollment_id IN (
12      SELECT enrollment_id
13      FROM enrollments
14      WHERE student_id = 1
15  );
16
17  DELETE FROM enrollments
18  WHERE student_id = 1;
19
20  DELETE FROM students
21  WHERE student_id = 1;
22
```

```
2 row(s) deleted.

2 row(s) deleted.

1 row(s) deleted.
```

3) Update the department name and instructor details for a specific course.

```
UPDATE Courses_Details cd
SET cd.course_description = 'Advanced Biology with Lab'
WHERE cd.course_id = 101;

UPDATE Instructors i
SET i.email = 'updated.email@example.com'
WHERE i.instructor_id = (
   SELECT ci.instructor_id
   FROM Course_Instructors ci
   WHERE ci.course_id = 101
   AND ci.instructor_id = 22101
);

UPDATE Departments d
SET d.department_name = 'Biological Sciences'
WHERE d.department_id = (
   SELECT cd.department_id
   FROM Courses_Details cd
   WHERE cd.course_id = 101
);

UPDATE Course_Instructors ci
SET ci.instructor_id = 22102
WHERE ci.course_id = 101
AND ci.instructor_id = 22101;
```

4) Students with Multiple Enrollments. (Join & Aggregate function)

```
SELECT s.student_id, s.first_name, s.last_name, COUNT(e.enrollment_id) AS enrollments_count
FROM students s
JOIN enrollments e
ON s.student_id = e.student_id
GROUP BY s.student_id, s.first_name, s.last_name
HAVING COUNT(e.enrollment_id) > 1;
```

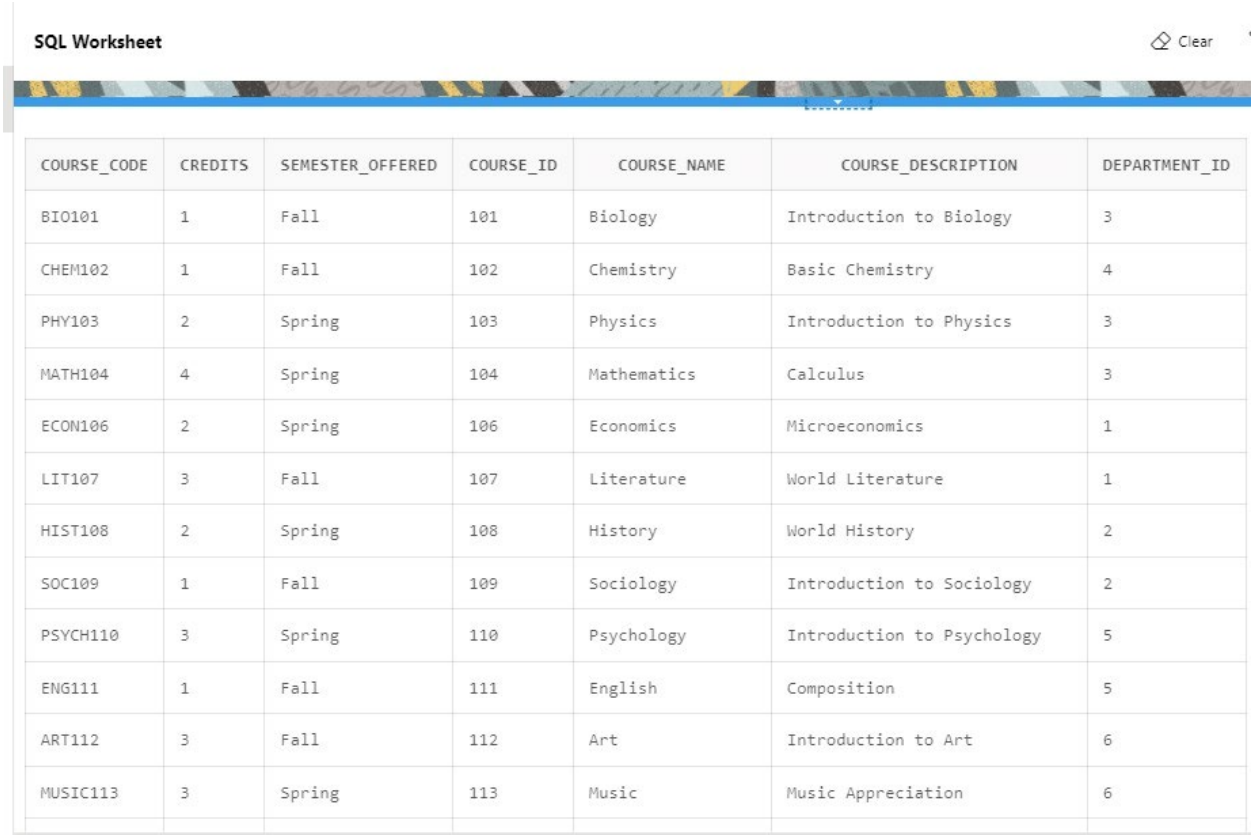
STUDENT_ID	FIRST_NAME	LAST_NAME	ENROLLMENTS_COUNT
1	Jane	Doe	2
2	John	Brown	2
3	Hamid	Ali	3

Download CSV

3 rows selected.

5) To retrieve information from both courses and courses_details tables using a join query. (Inner join)

```
SELECT c.course_code, c.credits, c.semester_offered, cd.course_id, cd.course_name, cd.course_description, cd.department_id
FROM courses c
INNER JOIN courses_details cd
ON c.course_code = cd.course_code;
```



COURSE_CODE	CREDITS	SEMESTER_OFFERED	COURSE_ID	COURSE_NAME	COURSE_DESCRIPTION	DEPARTMENT_ID
BIO101	1	Fall	101	Biology	Introduction to Biology	3
CHEM102	1	Fall	102	Chemistry	Basic Chemistry	4
PHY103	2	Spring	103	Physics	Introduction to Physics	3
MATH104	4	Spring	104	Mathematics	Calculus	3
ECON106	2	Spring	106	Economics	Microeconomics	1
LIT107	3	Fall	107	Literature	World Literature	1
HIST108	2	Spring	108	History	World History	2
SOC109	1	Fall	109	Sociology	Introduction to Sociology	2
PSYCH110	3	Spring	110	Psychology	Introduction to Psychology	5
ENG111	1	Fall	111	English	Composition	5
ART112	3	Fall	112	Art	Introduction to Art	6
MUSIC113	3	Spring	113	Music	Music Appreciation	6

6) **Retrieve Student Enrollment Details with Course Information**.

```
SELECT s.student_id, s.first_name, s.last_name, s.enrollment_date AS student_enrollment_date, e.enrollment_id,
e.enrollment_date AS enrollment_date, cd.course_id, cd.course_name, cd.course_code, cd.department_id

FROM students

JOIN enrollments e

ON s.student_id = e.student_id

JOIN courses_details cd

ON e.course_id = cd.course_id;
```

STUDENT_ID	FIRST_NAME	LAST_NAME	STUDENT_ENROLLMENT_DATE	ENROLLMENT_ID	ENROLLMENT_DATE	COURSE_ID	COURSE_NAME	COURSE_CODE	DEPARTMENT_ID
1	Jane	Doe	01-SEP-23	1	01-SEP-23	101	Biology	BIO101	3
2	John	Brown	02-AUG-23	4	04-AUG-23	101	Biology	BIO101	3
3	Hamid	Ali	28-MAY-23	5	28-MAY-23	101	Biology	BIO101	3
1	Jane	Doe	01-SEP-23	2	02-SEP-23	102	Chemistry	CHEM102	4
2	John	Brown	02-AUG-23	3	02-AUG-23	103	Physics	PHY103	3
3	Hamid	Ali	28-MAY-23	7	25-MAY-23	103	Physics	PHY103	3
3	Hamid	Ali	28-MAY-23	6	27-MAY-23	104	Mathematics	MATH104	3
5	Emily	Robinson	02-SEP-23	9	02-SEP-23	106	Economics	ECON106	1
6	Frank	Patel	03-SEP-23	10	03-SEP-23	107	Literature	LIT107	1
7	Henry	Anderson	04-SEP-23	11	04-SEP-23	108	History	HIST108	2
8	Isabella	Garcia	05-SEP-23	12	05-SEP-23	109	Sociology	SOC109	2
9	Jack	Thompson	06-SEP-23	13	06-SEP-23	110	Psychology	PSYCH110	5

7) Filtering Courses by Semester Offered and Department Name.

```
SELECT cd.course_id, cd.course_name, cd.course_code, cd.course_description,  c.credits, c.semester_offered,
d.department_name
FROM courses_details cd
JOIN courses c
ON cd.course_code = c.course_code
JOIN departments d
ON cd.department_id = d.department_id
WHERE c.semester_offered = 'Spring' AND d.department_name = 'Science';
```

COURSE_ID	COURSE_NAME	COURSE_CODE	COURSE_DESCRIPTION	CREDITS	SEMESTER_OFFERED	DEPARTMENT_NAME
103	Physics	PHY103	Introduction to Physics	2	Spring	Science
104	Mathematics	MATH104	Calculus	4	Spring	Science

Download CSV

2 rows selected.

8) Subquery to Find Courses Taught by Each Instructor.

```
SELECT
  i.instructor_id,
  i.first_name || ' ' || i.last_name AS instructor_name,
  (
        SELECT LISTAGG(cd.course_name, ', ')
         WITHIN GROUP (ORDER BY cd.course_name)
         FROM courses_details cd
         JOIN course_instructors ci
        ON cd.course_id = ci.course_id
        WHERE ci.instructor_id = i.instructor_id
) AS courses_taught
FROM instructors i;
```

INSTRUCTOR_ID	INSTRUCTOR_NAME	COURSES_TAUGHT
22101	Alice Smith	Biology
22102	Bob Johnson	Chemistry
22103	Carol Williams	Physics
22104	David Brown	Mathematics
22106	Zara Montgomery	Economics
22107	Finnegan Everest	Literature
22108	Octavius Hawthorn	History
22109	Juniper Sinclair	Sociology
22110	Orion Hawthorne	Psychology
22111	Seraphina Delgado	English
22112	Cassius Kensington	Art
22113	Elara Winslow	Music

9) Subquery to Count Courses per Department.

```
SELECT
   d.department_id,
   d.department_name,
   (
            SELECT COUNT(*)
            FROM courses_details cd
            WHERE cd.department_id = d.department_id
    ) AS num_courses
FROM departments d;
```

DEPARTMENT_ID	DEPARTMENT_NAME	NUM_COURSES
1	Technology	2
2	Humanities	2
3	Science	3
4	Engineering	1
5	Social Sciences	2
6	Arts	2
7	Politics	2
8	Sociology	2
9	Business	2
10	Communication	1

Download CSV

10 rows selected.