|  |  |
| --- | --- |
|  | **Database Management Systems**  **BSCS-4**  **Department of Computer Science**  **Bahria University, Lahore Campus** |

**Assignment: [2]**

Date: Week 7, 4th November 2023

Name: affan ahmad\_\_

Roll No: \_03-134221-003\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| **Evaluation of CLO** | **Question Number** | **Marks** | **Obtained Marks** |
| **CLO: Apply different database model to design conceptual, logical or physical database.** |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| **Total Marks** | | **20** |  |

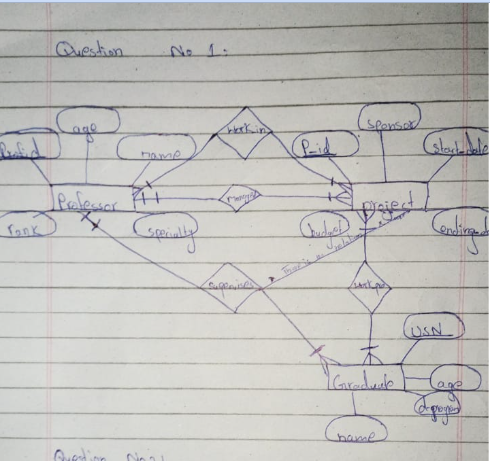
**Question 1**

Professors have a PROFID, a name, an age, a rank, and a research specialty.

Projects have a project number, a sponsor name (e.g. UGC/AICTE/...), and a budget.

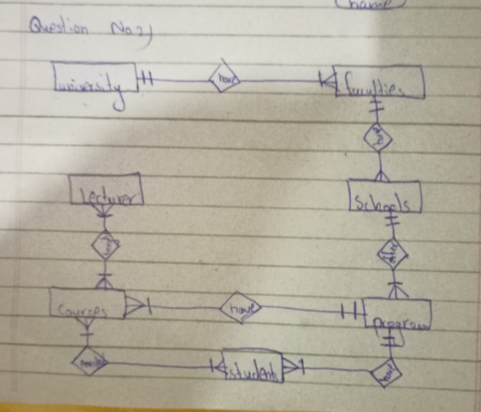
Graduate students have an ID, a name, an age, and a degree program (e.g. MCA/ MPhil/BE/ME).

Each project is managed exactly by one professor (known as the project's principal investigator). Each project is worked on by one or more professors (known as the project's co-investigators). Professors can manage/work on multiple projects. Each project is worked on by one or more graduate students (known as the project's research assistants). Graduate students can work on multiple projects. Each professor can supervise many students. A student who is working on a project can be supervised by only one professor. Draw an ERD of the above scenario.



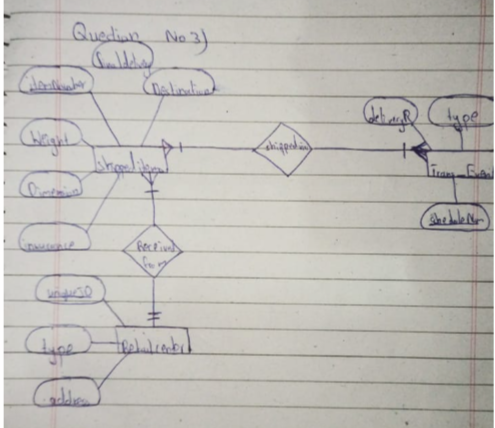
**Question 2**

A University contains many Faculties. The Faculties in turn are divided into several Schools. Each School offers numerous programs and each program contains many courses. Lecturers can teach many different courses and even the same course numerous times. Courses can also be taught by many lecturers. A student is enrolled in only one program but a program can contain many students. Students can be enrolled in many courses at the same time and the courses have many students enrolled.



**Question 3**

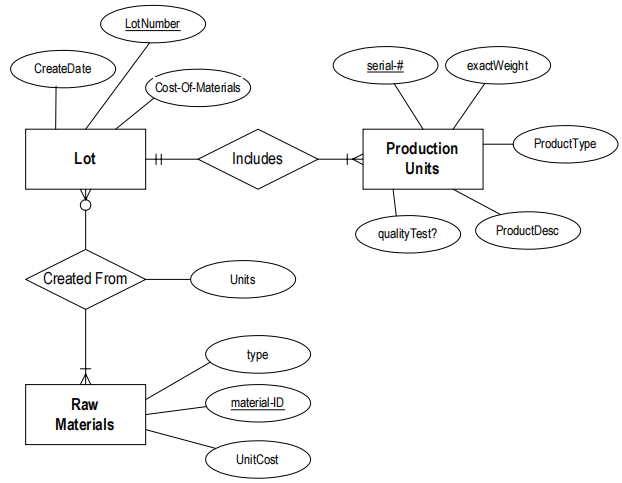
UPS prides itself on having up-to-date information on the processing and current location of each shipped item. To do this, UPS relies on a company-wide information system. Shipped items are the heart of the UPS product tracking information system. Shipped items can be characterized by item number (unique), weight, dimensions, insurance amount, destination, and final delivery date. Shipped items are received into the UPS system at a single retail center. Retail centers are characterized by their type, uniqueID, and address. Shipped items make their way to their destination via one or more standard UPS transportation events (i.e., flights, truck deliveries). These transportation events are characterized by a unique scheduleNumber, a type (e.g, flight, truck), and a deliveryRoute. Please create an Entity Relationship diagram that captures this information about the UPS system. Be certain to indicate identifiers and cardinality constraints.



**Question 4**

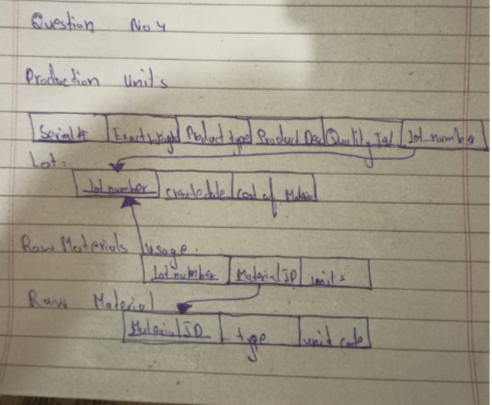
**CREATING A RELATIONAL DATABASE SCHEMA.**

Production tracking is important in many manufacturing environments (e.g., the pharmaceuticals industry, children’s toys, etc.). The following ER diagram captures important information in the tracking of production. Specifically, the ER diagram captures relationships between production lots (or batches), individual production units, and raw materials.



**Note: The following questions refer back to the above ER diagram.**

1. Please convert the ER diagram into a relational database schema. Be certain to indicate primary keys and referential integrity constraints.
2. Please identify an attribute in the above ER diagram that might represent a composite attribute, and explain why/how it might represent a composite attribute.
3. Please identify an attribute in the ER diagram that could represent a derived attribute and explain why/how it might represent a derived attribute.
4. The ER diagram/relational database schema contains several instances of data redundancy. Please identify one instance where a data redundancy issue exists.



**Bahria University, Lahore Campus**

Department of Computer Sciences

**Assignment: [1]**

**(Fall 2023)**

|  |  |  |
| --- | --- | --- |
| Course: | **Database Management System Lab** | Date: |
| Course Code: | CSL 220 | Max Marks: 40 |
| Faculty’s Name: |  | Lab Engineer: |

Name: AFFAN AHMAD \_\_\_ Enroll No: \_03-134221-003\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Instructions for the Case Study:**

Perform the given queries

Suitable tuples have to be entered so that queries may be executed correctly

**Scenario**

A university database contains information about students, courses, and registrations. The database schema consists of the following tables:

**1. Students:**

- student\_id (Primary Key)

- student\_name

- student\_age

- student\_major

**2. Courses:**

- course\_id (Primary Key)

- course\_name

- course\_department

- course\_credits

**3. Registrations:**

- registration\_id (Primary Key)

- student\_id (Foreign Key referencing Students)

- course\_id (Foreign Key referencing Courses)

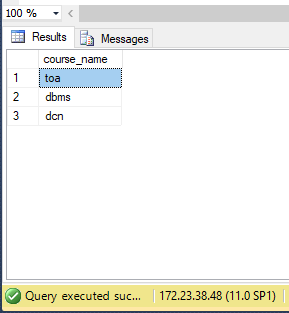
- registration\_date

**Assume suitable data exists in these tables.**

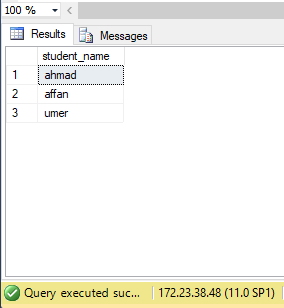
**Task 1:**

1. Write SQL queries to create the database named "UniversityDB" and create the three tables (Students, Courses, Registrations) with the appropriate attributes.
2. create database UniversityDB
3. create table Students128
4. (
5. student\_id int Primary Key,
6. student\_name varchar,
7. student\_age int ,
8. student\_major varchar,
9. );
10. create table courses128
11. (
12. course\_id int Primary Key,
13. course\_name varchar,
14. course\_department varchar ,
15. course\_credits varchar
16. );
17. create table Registrations128
18. (
19. registration\_id int Primary Key,
20. student\_id int Foreign Key references Students128,
21. course\_id int Foreign Key references Courses128,
22. registration\_date int
23. );
24. Display the names of all courses offered by the university.

select course\_name from courses129



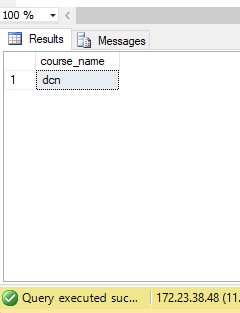
1. List all students who are majoring in "Computer Science."
2. select student\_name from Students127
3. where student\_major = 'computer science';



1. Show the course names that belong to the "Science" department.

select course\_name from courses127

where course\_department = 'science';



1. Retrieve the names of students who have registered for the course with ID 101.

SELECT S.student\_name

FROM Students S, Registrations R

WHERE S.student\_id = R.student\_id

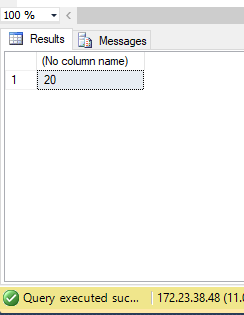
AND R.course\_id = 101;

1. Display the total number of registrations in the database.

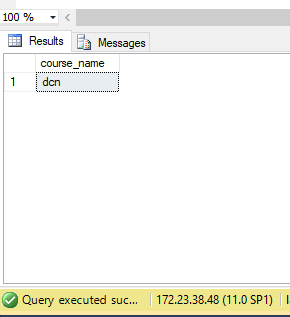
**Task 2:**

1. Write an SQL query to calculate the average age of all students.

select avg(student\_age) from Students127



1. List the course names that have more than 3 credits.
2. select course\_name from courses126
3. where course\_credits>3 ;



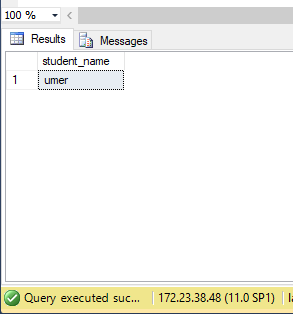
1. Show the names of students who are majoring in "Mathematics" and are older than 20 years.

SELECT S.student\_name

FROM Students S

WHERE S.student\_major = 'Mathematics'

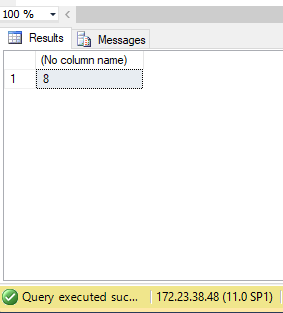
AND S.student\_age > 20;



1. Display the student names who have not registered for any course.
2. List the course names along with the count of students registered for each course.
3. Retrieve the student names who registered for courses in the year 2022.

**Task 3:**

1. Calculate the sum of credits for all courses.
2. SELECT SUM(course\_credits) FROM courses126



1. Show the student names along with their major and the department of the courses they have registered for.
2. Display the course names with the earliest registration date.
3. List the student names who registered for the same course multiple times.

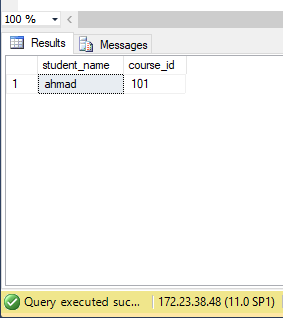
SELECT S.student\_name, R.course\_id

FROM Students S, Registrations R

WHERE S.student\_id = R.student\_id

GROUP BY S.student\_name, R.course\_id

HAVING COUNT(R.registration\_id) > 1;



1. Retrieve the course names that have the highest number of registrations.

SELECT C.course\_name

FROM Courses C

WHERE C.course\_id = (

SELECT course\_id

FROM Registrations

GROUP BY course\_id

ORDER BY COUNT(\*) DESC

LIMIT 1

);

1. Eliminate the "Registrations" table from the database.

DROP TABLE Registrations122

**Lab Grading Sheet**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Max Marks** | **Obtained Marks** | **Comments(*if any*)** |
| 1. | 10 |  |  |
| 2. | 10 |  |  |
| 3. | 10 |  |  |
| 4. | 10 |  |  |
| **Total** | **40** |  | **Signature** |

* **Note : Attempt all tasks and get them checked by your Lab. Instructor**