

BTD210- Assignment 2

Learning outcomes

Upon successful completion of this assignment, you will have demonstrated the abilities to:

1. Convert tables into 3NF/BCNF and infer the associated ERD
2. Implement the relational database model by creating tables and constraints on attributes.
3. Enter data into the database.
4. Query the database to retrieve information.

Please work in **groups** to complete this assignment. This assignment is **worth 10%** of the total course grade and will be evaluated through your written submission, as well as the assignment demo. During the assignment demo, group members are randomly selected to present the answers to each of the assignment questions. Group members who are not present during the assignment demo will lose the demo mark (up to 30% of assignment mark).

Submission

Please submit the submission file(s) through Blackboard. Only one person must submit for the group and only the last submission will be marked.

- Assignment SQL file: Asg2.SQL
- Assignment document: Asg2.docx
- ERD.pdf if needed

Assignment overview

In this assignment, you will design and implement a database for the BSA program. This database will store information about courses in the program. Samples of course outlines are attached (Table 1 samples).

Some information to help you design the database follows:

- 1- Each course is identified by a course code (for example BTD210).
- 2- Each course has one or more course learning outcomes (CLOs).
- 3- Each course can meet one or more of the 8 program learning outcomes (PLOs).
- 4- One or more of the 12 engineering graduate attributes (GAs) are taught in each course.
- 5- Each course can have zero to many pre-requisites, which must be courses offered in previous semesters.
- 6- Each course can have zero to many textbooks. It is possible to have the same textbook in multiple courses.
- 7- Each course is developed by one of the Professors. For example:

Table 2: Professor developing courses.

| Course code | Developed by |
|-------------|-----------------|
| MEC110 | Jacky Lau |
| SES250 | Jacky Lau |
| SEM305 | Allan Randall |
| SEH300 | Kifah Al-Ansari |
| BTD210 | Vida Movahedi |

- 8- Each academic term, e.g., Fall 2024 (code: 2247) zero to many of the program courses are offered, possibly in multiple sections. For example, BTD210NAA_2247 and BTD210NCC_2247 are two sections of the BTD210 course offered in Fall 2024.
- 9- Each course section (e.g. BTD210NAA_2247) can have one or more professors. Each professor can teach zero to many course sections in each academic term. For example, these courses were offered:

Table 3: Courses offered.

| Section code | Professor |
|----------------|-----------------------|
| SES250NAA_2244 | Arif Obaid; Jacky Lau |
| SEP200NAA_2244 | Mufleh Al-Shatnawi |
| MEC110NAA_2244 | Jacky Lau; Ali Sanaee |

| | |
|----------------|-----------------------------|
| MEC110NAA_2247 | Jacky Lau; Arif Obaid |
| SEM305NAA_2247 | Allan Randall |
| SEH300NAA_2247 | Kifah Al-Ansari; Arif Obaid |
| BTD210NBB_2247 | Vida Movahedi |

Part I. Database Normalization and Entity Relationship Model (40%)

- 1- For each of the given tables, convert to 3NF, *showing all steps* (from 1NF to 3NF). Use the arrow notation.

1NF:

(Course_code->

Course_name, Semester, Course_description, Contact_Hours_week, ISBN, Name, Author, Publisher, Method_of_instruction_ID, Method_of_instruction, Prerequisite_ID, Prerequisite)

CLO_ID-> Learning_Outcomes

Graduate_ID-> Graduate_attributes

PLO_ID->PLO

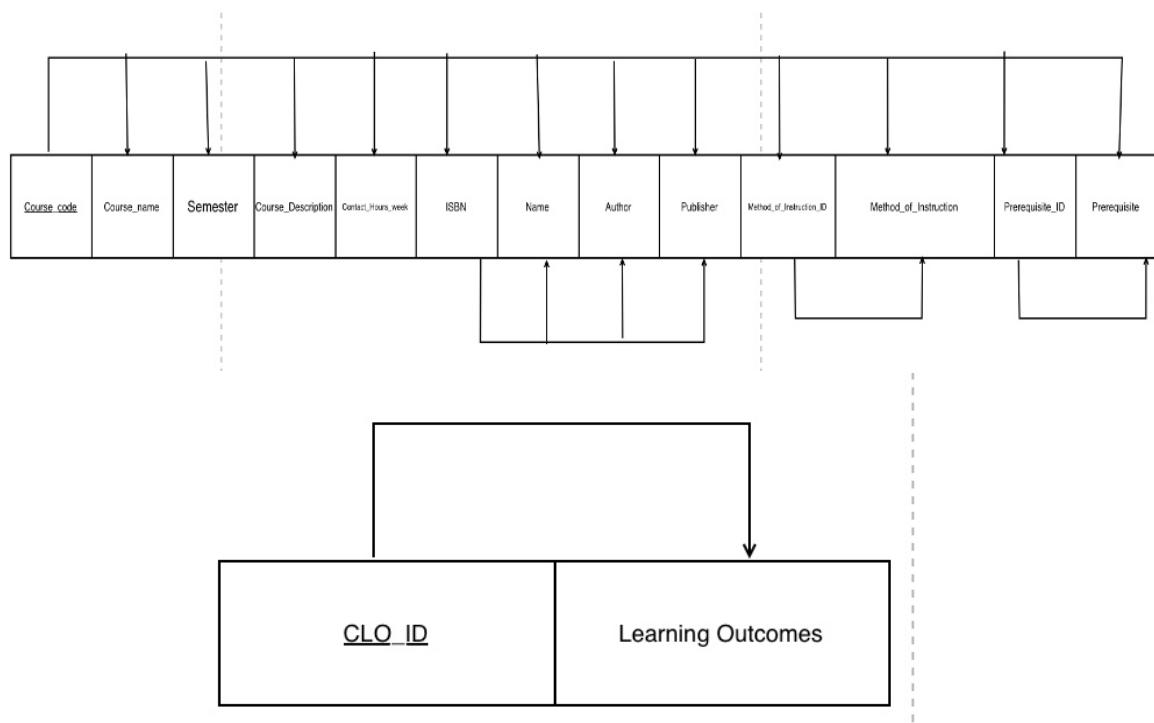
Partial Dependency - None

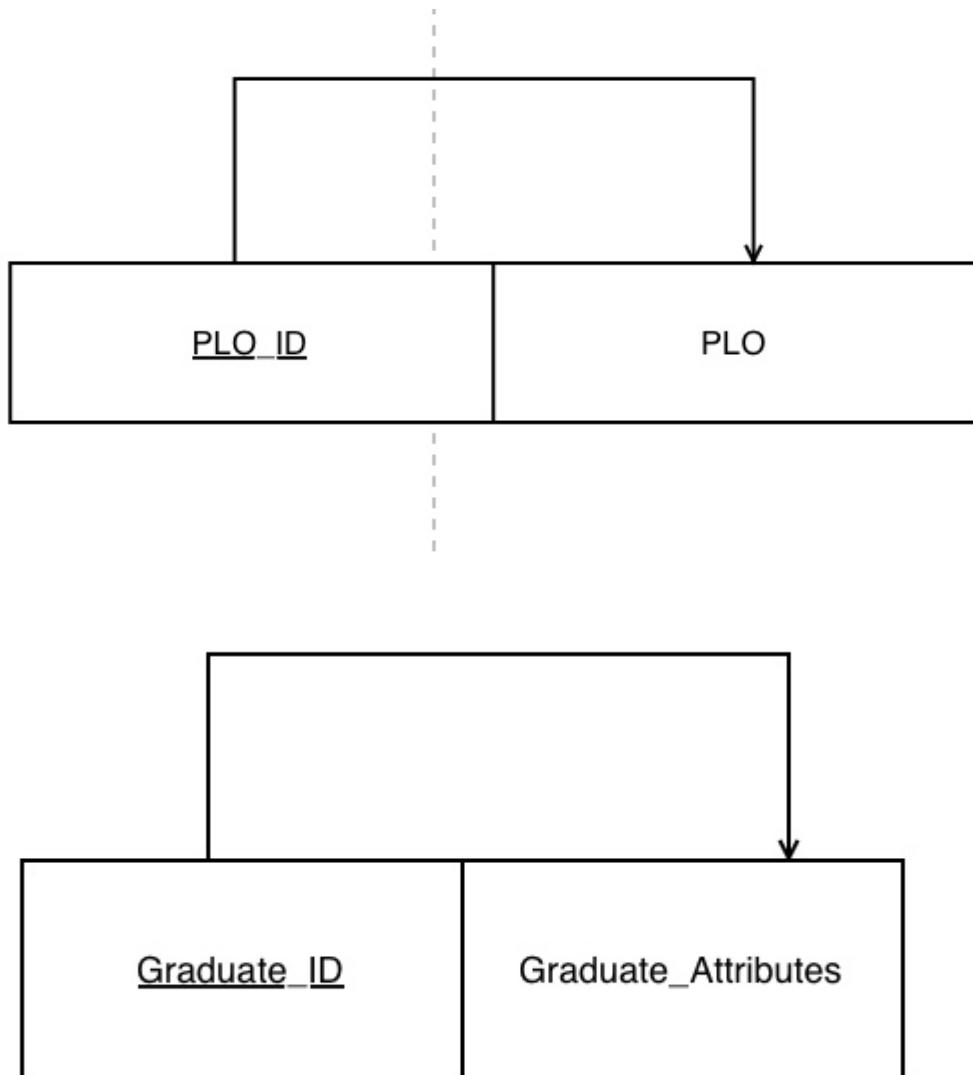
Transitive Dependency

ISBN-> Name, Author, Publisher

Method_of_instruction_ID-> Method_of_instruction

Prerequisite_ID->Prerequisite





2NF:

SAME AS 1NF AS THERE ARE NO PARTIAL DEPENDENCY

3NF:

Course_code→Course_name,Semester,Course_description,Contact_Hours_week,ISBN,
,Method_of_instruction_ID ,Prerequisite_ID)

ISBN→ Name,Author,Publisher

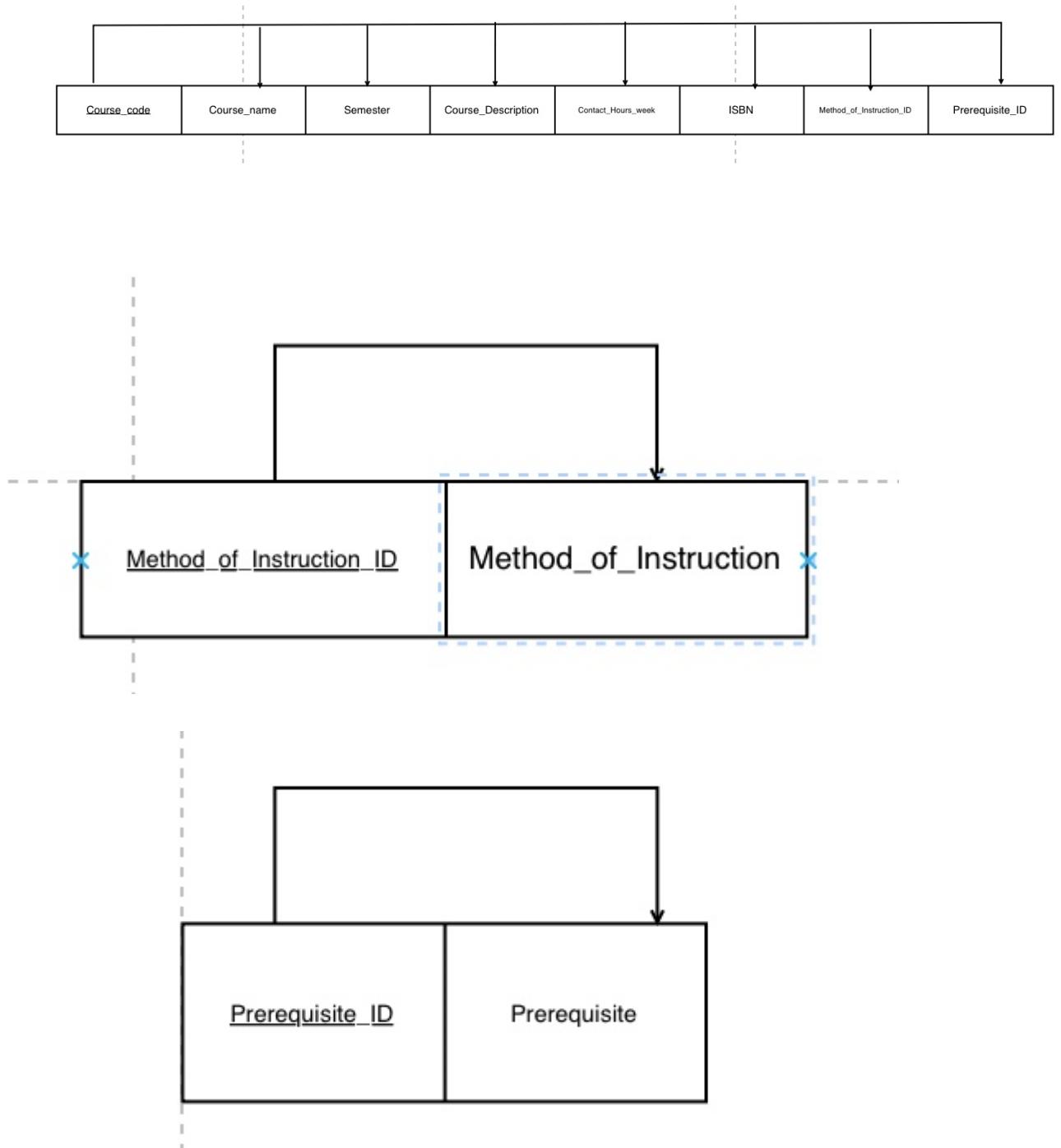
Method_of_instruction_ID→ Method_of_instruction

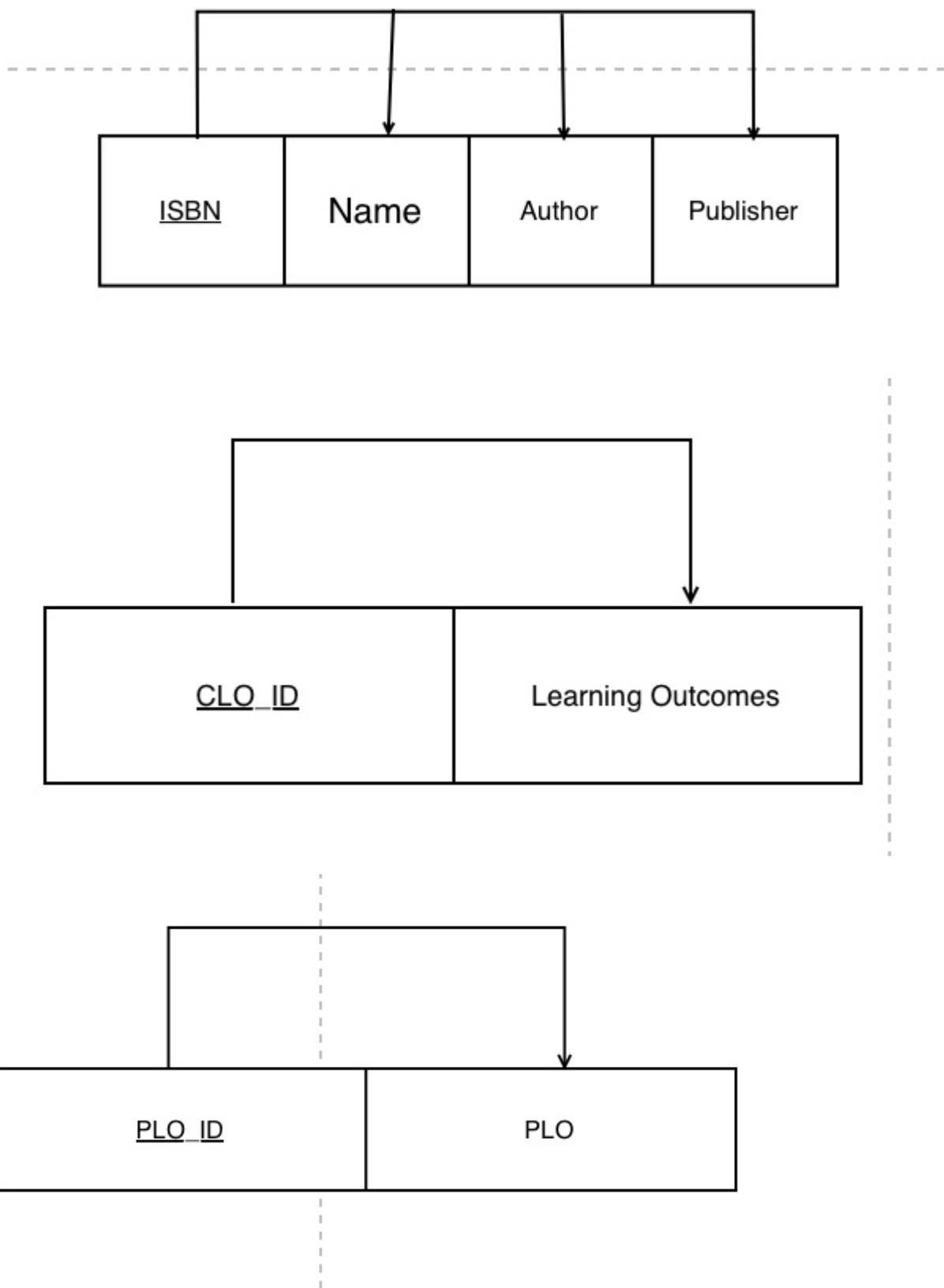
Prerequisite_ID→Prerequisite

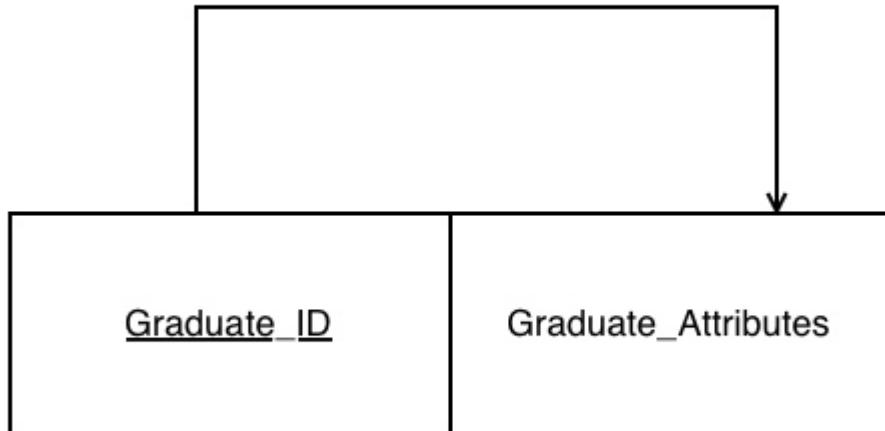
CLO_ID→ Learning_Outcomes

Graduate_ID→ Graduate_attributes

PLO_ID→PLO

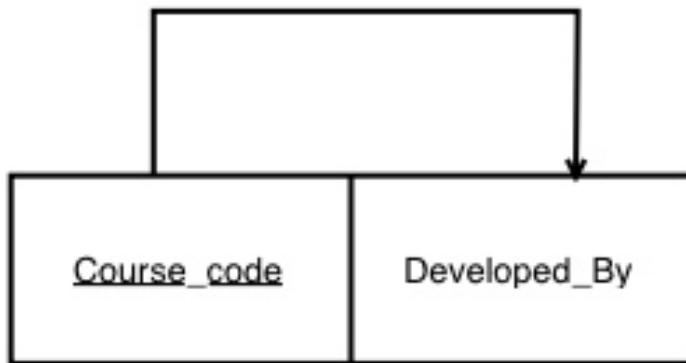






Normals Forms for Second Table: Professor Developing Courses

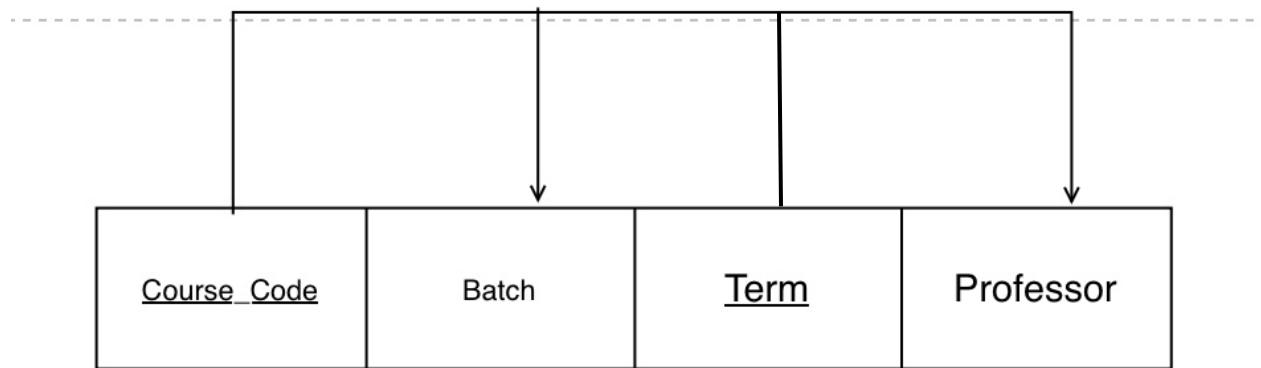
1NF , 2NF AND 3NF (All will be same)



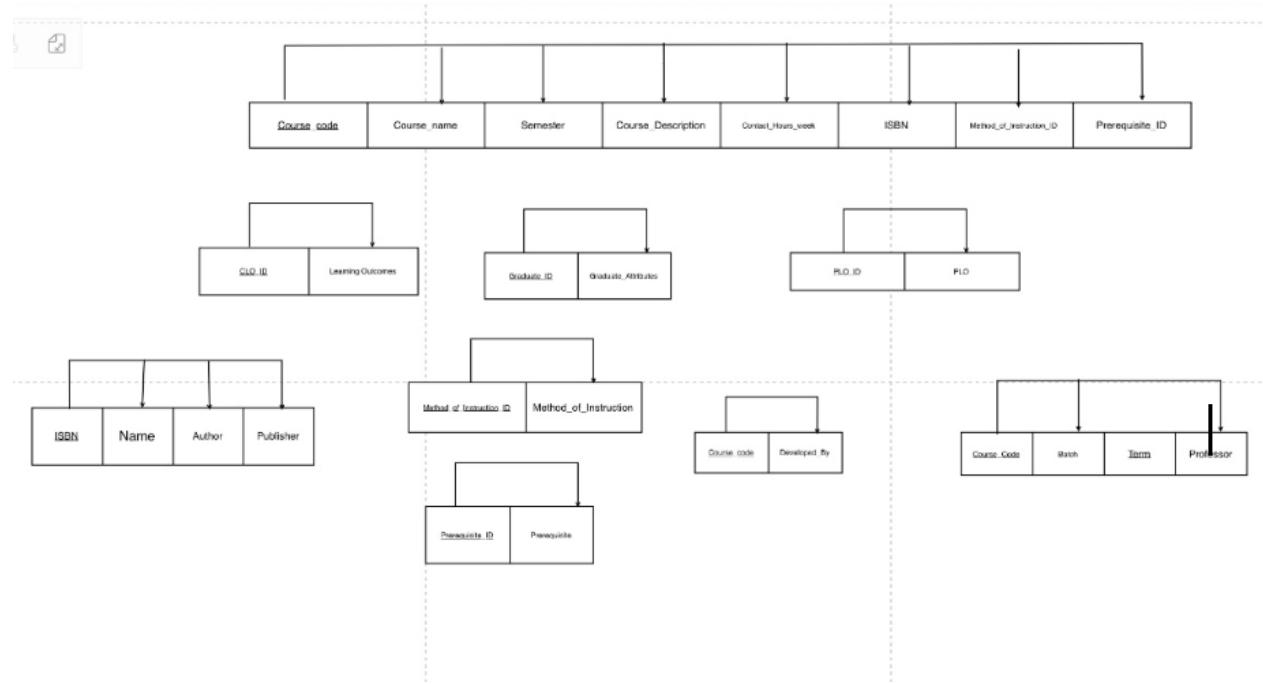
Normal Forms for Third Table:

Section code is divided into three parts

1NF, 2NF AND 3NF (All will be same)



- 2- Combine the tables obtained at the end of the normalization process into one logical design. Apply changes needed based on Table 6.7 (slides 22 to 25 in slides_08_Normalization_2) and explain



Combining all the tables at the end of the normalization process creates a unified, logical design that ensures each table retains its unique purpose while contributing to the overall structure. For instance, the **Course table** manages course-specific details like course codes, names, descriptions, and prerequisites, while maintaining relationships to textbooks, CLOs, and PLOs. The **Book table** provides a centralized repository for ISBNs, authors, publishers, and related course material, avoiding redundancy. The **Prerequisite table** isolates dependencies between courses, ensuring clarity in course progression.

Similarly, the **Method of Instruction table** encapsulates details of teaching methods (lecture, lab) with a clear association to specific courses. The **Professor table** records which instructors develop or teach courses, maintaining accountability and traceability. The **CLO, PLO, and Graduate Attributes tables** focus on educational outcomes and align seamlessly with relevant courses, supporting curriculum analysis and assessment. Also DevelopedBy table giving the developerNames of the course .

Combining these tables logically organizes data by separating concerns, reducing redundancy, and ensuring consistency through normalized relationships. This structure reflects real-world scenarios where each entity has a distinct role but interacts with others in a cohesive and efficient manner. It is robust, scalable, and capable of handling complex queries while preserving data integrity and accuracy.

- 3- Draw the *ERD* using Microsoft Visio.

Note that your design must not have any many-to-many relationships or multi-valued attributes. PK and FK must be clearly stated. Strong and weak relationships must be correctly shown. Required attributes must be in **BOLD**. Export the ERD to .jpg and insert in your assignment document file (or attach a separate pdf). **ZERO marks for Part I and II, if ERD is not clear.**

- 4- Design: Is this design suitable for implementation in the school? What are some limitations or considerations? Would you consider denormalization? Would you consider more atomicity? Or less atomicity? More granularity? Less granularity? Please explain in detail.

Strengths

1. Normalization: The design is highly normalized, reducing redundancy and ensuring data consistency.
2. Clear Relationships: Entities like Course, Book, PLO, Graduate Attributes, and their respective bridges provide clear relational mapping.
3. Granularity: Fine granularity in tables like CLO, Graduate_Bridge, and Book_Bridge allows for detailed data representation.
4. Scalability: This design supports scalability as new entities or relationships can be added without major redesign.

Considerations and Limitations:

1. Complexity:
 - The extensive normalization might make querying complex, requiring multiple joins, which could impact performance for large datasets.
 - For example, finding all books associated with a professor teaching a specific course may involve multiple joins (Book_Bridge, Course, Section_Bridge, Professor).
2. Atomicity:
 - While high atomicity reduces redundancy, it can also make the system harder to manage. For example, if textbooks are frequently updated across multiple courses, the Book_Bridge table might become cumbersome.
3. Denormalization:
 - Consider partial denormalization for frequently accessed data. For example, maintaining a summary table for courses, professors, and books might improve query performance.

- Denormalizing some bridge tables (e.g., merging Graduate_Bridge and PLO_Bridge) could simplify certain queries.

4. Granularity:

- Current granularity is high, which is great for detailed tracking. However, it might be excessive for less critical data like Method_of_Instruction if it isn't central to operational queries.
- More granularity might be needed in some areas, such as Professor if you want to track additional attributes like department or contact info.

5. Usability:

- High normalization requires proficient database administrators and developers who can handle the system's complexity.
- The design assumes a high level of data integrity, so maintaining this integrity requires robust data entry and validation processes.

Recommendations

1. Performance Optimizations:

- Create indexed views or materialized views for frequently accessed queries.
- Use indexing on foreign keys and frequently queried columns to enhance performance.

2. Atomicity Adjustments:

- Less atomicity might be beneficial for less critical entities like Method_of_Instruction by embedding this information into the Course table.

3. Granularity Review:

- Maintain current granularity where detailed analysis is needed (e.g., CLO tracking).
- Reduce granularity in areas that do not require detailed tracking, like Term.

4. Denormalization:

- Consider denormalizing bridges that connect highly interrelated entities with many-to-many relationships (e.g., Graduate_Bridge and PLO_Bridge).

5. Scalability for New Features:

- Include space for extending the schema to handle additional features like online materials, course evaluations, or student performance tracking.

5- How many tables are in your ERD?

There are a total of **14 tables**

Part II. Database IMPLEMENTATION (20%)

In your SQL file, write the SQL statements to:

- 1- Create a database named BTD210_Asg2_YourGroupNo.
- 2- Create the tables of your relational database model. **These must match your design in Part I.** Specify the PK, FK, and other necessary constraints. Include all statements in your SQL file.
- 3- Enter the data from the samples shown in the given tables. Include all *insert* queries in your SQL file. Paste screenshots of table contents.

Next page→

BTD210 - Database Design Principles

The screenshot shows the Object Explorer on the left and a query window on the right. The query window contains the following SQL code:

```
INSERT INTO Graduate_attributes (Graduate_ID, Graduate_attributes)
VALUES
(1, 'Knowledge Base'),
(2, 'Problem Analysis'),
(3, 'Investigation'),
(4, 'Design'),
(5, 'Use of Engineering Tools'),
(6, 'Individual and Team Work'),
(7, 'Communication Skills'),
(8, 'Professionalism'),
(9, 'Impact on Society and the Environment'),
(10, 'Ethics and Equity'),
(11, 'Economics and Project Management'),
(12, 'Life-long Learning');
```

The Messages pane at the bottom shows the results of the query execution:

```
(12 rows affected)

Completion time: 2024-12-04T10:45:26.0497834-05:00
```

A yellow bar at the bottom indicates "Query executed successfully."

The screenshot shows the Object Explorer on the left and a query window on the right. The query window contains the following SQL code:

```
INSERT INTO PLO (PLO_ID, PLO)
VALUES
(1, 'Apply mathematics, natural sciences, and engineering fundamentals to solve engineering problems.'),
(2, 'Create software engineering solutions that satisfy technical and business requirements.'),
(3, 'Design an optimal solution using artificial intelligence, data mining, and machine learning tools for complex and open-ended problems.'),
(4, 'Employ interpersonal, teambuilding, and leadership skills to solve problems independently and in diverse teams.'),
(5, 'Communicate complex engineering problems and solutions to fellow software engineers and designers as well as non-technical audiences.'),
(6, 'Act ethically and responsibly with public welfare and environmental protection as a guiding professional practice.'),
(7, 'Plan and manage the scope, cost, timing, and quality of the project for success as defined by the project stakeholders.'),
(8, 'Utilize investigative practices and self-awareness techniques to identify and pursue lifelong learning opportunities within their field of study and more broadly.');
```

The Messages pane at the bottom shows the results of the query execution:

```
(8 rows affected)

Completion time: 2024-12-04T10:46:34.0771104-05:00
```

A yellow bar at the bottom indicates "Query executed successfully."

BTD210 - Database Design Principles

The screenshot shows the Object Explorer on the left, connected to the local SQL Server 15.0.2000 instance. The database selected is BTD210_Asg2_5. The SQL Query window on the right contains the following SQL code:

```
SQLQuery1.sql - (lo...10_Asg2_5 (sa (52))*
INSERT INTO Professor (Professor_ID, Professor)
VALUES
(1, 'Arif Obaid'),
(2, 'Jacky Lau'),
(3, 'Mufleh Al-Shatnawi'),
(4, 'Ali Sanaee'),
(5, 'Allan Randall'),
(6, 'Kifah Al-Ansari'),
(7, 'Vida Movahedi');
```

The Messages pane shows the output:

```
(7 rows affected)
Completion time: 2024-12-04T10:47:20.1713198-08:00
```

The status bar at the bottom indicates "Query executed successfully."

The screenshot shows the Object Explorer on the left, connected to the local SQL Server 15.0.2000 instance. The database selected is BTD210_Asg2_5. The SQL Query window on the right contains the following SQL code:

```
SQLQuery1.sql - (lo...10_Asg2_5 (sa (52))*
INSERT INTO Course (Course_code, Course_Name, Semester, Course_Description, Contact_Hours_Week)
VALUES
('SEM305', 'Discrete Mathematics', 1,
'The mathematics of modern computer science is built almost entirely on discrete math. Students are introduced to discrete structures in order to formulate abstract concepts that can be manipulated by computers.', 56),
('MEC110', 'Mechanics', 1,
'This course introduces the subject of statics. The study of particles and rigid bodies in equilibrium. Students study, and solve problems involving, the equilibrium of particles and rigid bodies.', 84),
('SES250', 'Electromagnetics', 2,
'Students are introduced to electrostatics, magnetism and circuit theory with an emphasis on circuit and machine design. Electromagnetics is the foundation of all student work in engineering.', 84),
('BTD210', 'Database Design Principles', 3,
'This course introduces the principles of relational database design and use. Students learn a methodology for relational database design that uses Entity Relationship (ER) modeling.', 56),
('SEH300', 'Digital and Analog Circuits', 3,
'An introduction to the basic concepts of electricity, magnetism, electric circuits, and basic combinational and sequential digital circuits. Students develop an understanding of how these concepts are used in the design of electronic systems.', 56);
```

The Messages pane shows the output:

```
(5 rows affected)
Completion time: 2024-12-04T10:52:31.9129953-06:00
```

The status bar at the bottom indicates "(local)\SQLEXPRESS (15.0 RTM) sa (52) BTD210_Asg2_5 00:00:00 0 rows".

BTD210 - Database Design Principles

The screenshot shows the Object Explorer on the left with the database 'BTD210_Asg2_5' selected. The SQL Query window on the right contains the following SQL code:

```
SQLQuery1.sql - (local)\Asg2_5 (sa (52)) * 100 % 100 %
INSERT INTO CLO(CLO_ID, Course_code, Learning_Outcomes)
VALUES
(1, 'SEM305', 'Compare logical constructs and proofs to verify mathematical statements.'),
(2, 'SEM305', 'Arrange sets as a building block for the types of objects considered in discrete mathematics.'),
(3, 'SEM305', 'Construct matrices for mathematical transformations of physical systems.'),
(4, 'SEM305', 'Classify algorithms according to growth to minimize computing time.'),
(5, 'SEM305', 'Apply the principles of induction for mathematical proofs.'),
(6, 'SEM305', 'Assemble graphs to show the relationships between objects.'),
(7, 'SEM305', 'Construct trees to model computer algorithms.')
(8, 'SEM305', 'Design logic circuits using the principles of Boolean algebra.'),
(9, 'MEC110', 'Calculate the resultant of a number of concurrent forces in two or three dimensions.'),
(10, 'MEC110', 'Calculate the magnitude and direction of a force required to keep a given force system in equilibrium.'),
(11, 'MEC110', 'Draw a free-body diagram of a particle acted on by forces and use the diagram as an aid to calculate the magnitudes and directions of the unknown force(s).'),
(12, 'MEC110', 'Use the concepts of moments and couples to calculate the single force and the single couple which is equivalent to a system of coplanar couples and/or a system of parallel forces acting on a rigid body about a point of rotation.'),
(13, 'MEC110', 'Define a rigid body and use free body diagrams to aid in solving for the unknown forces and/or couples required to maintain a two-dimensional rigid body in equilibrium about a point of rotation.'),
(14, 'MEC110', 'Calculate the coordinates of the centroids of plane areas and of the centers of gravity of homogeneous plates having uniform thickness. '),
(15, 'MEC110', 'Calculate the magnitudes of the unknown forces acting in some or all of the members of a truss, machine, or frame which is acted on by a number of external forces and which is held in equilibrium by a number of reaction forces at supports and joints.'),
(16, 'MEC110', 'Explain dry friction and use the equations of dry friction to solve the problems of statics equilibrium.'),
(17, 'SES250', 'Examine electrostatic induction in machines for commercial applications.'),
(18, 'SES250', 'Explore the role of electrical resistance in controlling current and voltage in hardware applications.'),
(19, 'SES250', 'Apply magnetic induction to electric motors.'),
(20, 'SES250', 'Connect the interaction between electric and magnetic fields in simple circuits.'),
(21, 'SES250', 'Predict component current and voltage for various circuit configurations.'),
(22, 'SES250', 'Design simple AC and DC circuits for electrical power transfer.'),
(23, 'BTD210', 'Compose SQL to retrieve data from databases.'),
(24, 'BTD210', 'Compose SQL to create and modify tables in databases.'),
(25, 'BTD210', 'Prepare a physical relational database schema for specific business applications.'),
(26, 'BTD210', 'Prepare a logical relational database schema for specific business applications.'),
(27, 'BTD210', 'Compose an Entity Relationship Diagram for specific business applications.')
```

The message pane shows '(99 rows affected)' and the completion time is '2024-12-04T10:56:18.7032638-05:00'. The status bar at the bottom indicates '(local)\SQLEXPRESS (15.0 RTM) sa (52) BTD210_Asg2_5 00:00:00 0 rows'.

The screenshot shows the Object Explorer on the left with the database 'BTD210_Asg2_5' selected. The SQL Query window on the right contains the following SQL code:

```
SQLQuery1.sql - (local)\Asg2_5 (sa (52)) * 100 % 100 %
INSERT INTO PLO_Bridge (PLO_Bridge_ID, PLO_ID, Course_code)
VALUES
(1, 1, 'SEM305'),
(2, 3, 'SEM305'),
(3, 1, 'MEC110'),
(4, 1, 'SES250'),
(5, 2, 'BTD210'),
(6, 3, 'BTD210'),
(7, 1, 'SEH300');
```

The message pane shows '(7 rows affected)' and the completion time is '2024-12-04T11:02:03.5737814-05:00'. The status bar at the bottom indicates '(local)\SQLEXPRESS (15.0 RTM) sa (52) BTD210_Asg2_5 00:00:00 0 rows'.

BTD210 - Database Design Principles

The screenshot shows the Object Explorer on the left and the SQL Query Editor on the right. The Object Explorer lists several database objects including Constraints, Triggers, Indexes, Statistics, and tables like dbo.DevelopedBy, dbo.Graduate_attributes, dbo.Graduate_Bridge, dbo.Method_of_Instruction, dbo.PLO, and dbo.PLO_Bridge. The SQL Query Editor contains the following SQL code:

```
INSERT INTO Method_of_Instruction (Method_of_Instruction_ID, Method_of_Instruction, Course_code)
VALUES
(1, 'Lecture', 'SEM305'),
(2, 'Laboratory', 'SEM305'),
(3, 'Lecture', 'MEC110'),
(4, 'Lecture', 'SES250'),
(5, 'Laboratory', 'SES250'),
(6, 'Lecture', 'BTD210'),
(7, 'Laboratory', 'BTD210'),
(8, 'Lecture', 'SEH300'),
(9, 'Laboratory', 'SEH300');
```

The Messages pane at the bottom shows the results of the query execution:

```
(9 rows affected)  
Completion time: 2024-12-04T11:07:52.5238824-05:00
```

A yellow bar at the bottom indicates "Query executed successfully."

The screenshot shows the Object Explorer on the left and the SQL Query Editor on the right. The Object Explorer lists the same set of database objects as the previous screenshot. The SQL Query Editor contains the following SQL code:

```
INSERT INTO Book (BOOK_ID, ISBN, Textbook_name, Author, Publisher)
VALUES
(1, '978-1260091991', 'Discrete Mathematics and Its Applications (8th Edition)', 'Kenneth H. Rosen', 'McGraw-Hill Education'),
(2, '978-0070799233', 'Vector Mechanics for Engineers: Statics and Dynamics (5th Edition)', 'Ferdinand P. Beer and E. Russell Johnston Jr.', 'McGraw-Hill'),
(3, '978-1107014022', 'Electricity and Magnetism', 'Edward M. Purcell and David J. Morin', 'Cambridge University Press'),
(4, '978130562748', 'Database Systems: Design, Implementation, & Management (12th Edition)', 'Carlos Coronel and Steven Morris', 'Course Technology'),
(5, NULL, NULL, NULL, NULL);
```

The Messages pane at the bottom shows the results of the query execution:

```
(5 rows affected)  
Completion time: 2024-12-04T11:11:22.0013701-08:00
```

A yellow bar at the bottom indicates "Query executed successfully."

BTD210 - Database Design Principles

The screenshot shows the Object Explorer on the left with the database structure. The SQL Query window on the right contains the following SQL code:

```
INSERT INTO DevelopedBy (DevelopedBy_ID, DeveloperName, Course_code)
VALUES
(1, 'Jacky Lau', 'MEC110'),
(2, 'Jacky Lau', 'SES250'),
(3, 'Allan Randall', 'SEM305'),
(4, 'Kifah Al-Ansari', 'SEH300'),
(5, 'Vida Movahedi', 'BTD210');
```

The Messages pane shows the results: "5 rows affected". The status bar at the bottom indicates "Query executed successfully.".

The screenshot shows the Object Explorer on the left with the database structure. The SQL Query window on the right contains the following SQL code:

```
INSERT INTO Prerequisite (Prerequisite_ID, Prerequisite, Course_code)
VALUES
(1, 'None', 'SEM305'),
(2, 'None', 'MEC110'),
(3, 'MEC110', 'SES250'),
(4, 'None', 'BTD210'),
(5, 'SES250', 'SEH300'),
(6, 'SEM305', 'SEH300');
```

The Messages pane shows the results: "6 rows affected". The status bar at the bottom indicates "Query executed successfully.".

BTD210 - Database Design Principles

The screenshot shows the SSMS interface with the Object Explorer on the left and a query window on the right. The query window contains the following SQL code:

```
INSERT INTO Term (Term_Code, Batch, Term)
VALUES
(1, 'NAA_', 2244),
(2, 'NAA_', 2247),
(3, 'NBB_', 2247);
```

The status bar at the bottom indicates "Query executed successfully." and provides connection information: (local)\SQLEXPRESS (15.0 RTM) | sa (55) | BTD210_Asg2_5 | 00:00:00 | 0 rows.

The screenshot shows the SSMS interface with the Object Explorer on the left and a query window on the right. The query window contains the following SQL code:

```
INSERT INTO Book_Bridge (Book_Bridge_ID, BOOK_ID, Course_code)
VALUES
(1, 1, 'SEM305'),
(2, 2, 'MEC110'),
(3, 3, 'SES250'),
(4, 4, 'BTD210'),
(5, 5, 'SEH300');
```

The status bar at the bottom indicates "Query executed successfully." and provides connection information: (local)\SQLEXPRESS (15.0 RTM) | sa (55) | BTD210_Asg2_5 | 00:00:00 | 0 rows.

BTD210 - Database Design Principles

The screenshot shows the Object Explorer on the left with the database structure. In the center, a query window titled "SQLQuery1.sql" is open, displaying an INSERT statement into the "Graduate_Bridge" table. The query inserts 10 rows of data. Below the query, the "Messages" pane shows "(10 rows affected)" and the completion time. At the bottom, a green status bar indicates "Query executed successfully".

```
Object Explorer
Connect ▾ SQLQuery1.sql - (local)\SQLEXPRESS (sa (55)) * ×
SQLQuery1.sql - (local)\SQLEXPRESS (sa (55)) * ×
INSERT INTO Graduate_Bridge (Graduate_Bridge_ID, Course_code, Graduate_ID)
VALUES
(1, 'SEM305', 1),
(2, 'MEC110', 1),
(3, 'SES250', 1),
(4, 'BTD210', 1),
(5, 'BTD210', 2),
(6, 'BTD210', 4),
(7, 'BTD210', 5),
(8, 'SEH300', 1),
(9, 'SEH300', 2),
(10, 'SEH300', 5);

100 % Messages
(10 rows affected)
Completion time: 2024-12-04T18:08:30.4415156-05:00

100 % Query executed successfully. (local)\SQLEXPRESS (15.0 RTM) sa (55) BTD210.Asg2_5 00:00:00 0 rows
Ln 277 Col 19 Ch 19 INS
```

The screenshot shows the Object Explorer on the left with the database structure. In the center, a query window titled "SQLQuery1.sql" is open, displaying an INSERT statement into the "Section_Bridge" table. The query inserts 11 rows of data. Below the query, the "Messages" pane shows "(11 rows affected)" and the completion time. At the bottom, a green status bar indicates "Query executed successfully".

```
Object Explorer
Connect ▾ SQLQuery1.sql - (local)\SQLEXPRESS (sa (55)) * ×
SQLQuery1.sql - (local)\SQLEXPRESS (sa (55)) * ×
INSERT INTO Section_Bridge (Section_Bridge_ID, Course_code, Professor_ID, Term_Code)
VALUES
(1, 'SES250', 1, 1), -- SES250 - Arif Obaid for NAA_ 2244
(2, 'SES250', 2, 1), -- SES250 - Jacky Lau for NAA_ 2244
(3, 'SEP200', 3, 1), -- SEP200 - Mufleh Al-Shatnawi for NAA_ 2244
(4, 'MEC110', 2, 1), -- MEC110 - Jacky Lau for NAA_ 2244
(5, 'MEC110', 4, 1), -- MEC110 - Ali Sanaee for NAA_ 2244
(6, 'MEC110', 2, 2), -- MEC110 - Jacky Lau for NAA_ 2247
(7, 'MEC110', 1, 2), -- MEC110 - Arif Obaid for NAA_ 2247
(8, 'BTD210', 7, 3), -- BTD210 - Vida Movahedi for NBB_ 2247
(9, 'SEH300', 6, 2), -- SEH300 - Kifah Al-Ansari for NAA_ 2247
(10, 'SEH300', 1, 2), -- SEH300 - Arif Obaid for NAA_ 2247
(11, 'SEH300', 5, 2); -- SEM305 - Allan Randall for NBB_ 2247

100 % Messages
(11 rows affected)
Completion time: 2024-12-04T18:15:17.3902565-05:00

100 % Query executed successfully. (local)\SQLEXPRESS (15.0 RTM) sa (55) BTD210.Asg2_5 00:00:00 0 rows
Ln 289 Col 61 Ch 61 INS
```

How many tables are in your database? **This must match the number in ERD** (question 5).
We have a total of 14 tables.

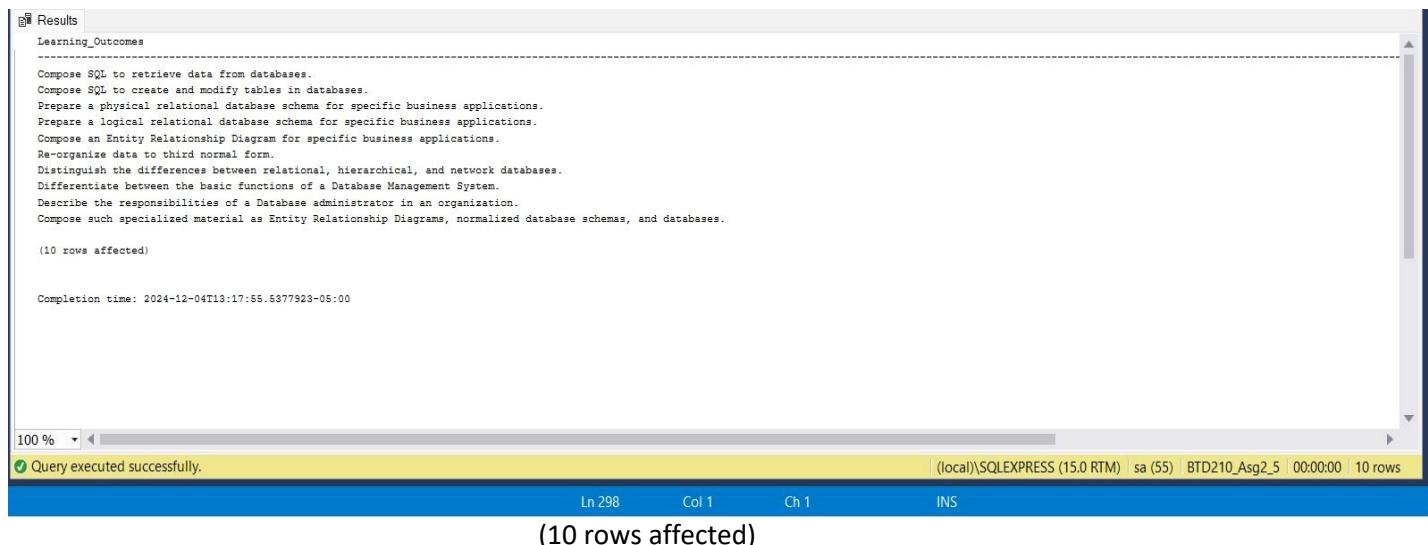
Part III. Data Manipulation (40%)

For the following questions, include in your assignment document file,

- (a) The SQL command (both in the SQL file and the document file)
- (b) The output in text format,
- (c) The number of rows affected.

Q1. List the CLOs for BTD210.

```
SELECT Learning_Outcomes  
FROM CLO  
WHERE Course_code = 'BTD210';
```



Results

| Learning_Outcomes |
|--|
| Compose SQL to retrieve data from databases. |
| Compose SQL to create and modify tables in databases. |
| Prepare a physical relational database schema for specific business applications. |
| Prepare a logical relational database schema for specific business applications. |
| Compose an Entity Relationship Diagram for specific business applications. |
| Re-organize data to third normal form. |
| Distinguish the differences between relational, hierarchical, and network databases. |
| Differentiate between the basic functions of a Database Management System. |
| Describe the responsibilities of a Database administrator in an organization. |
| Compose such specialized material as Entity Relationship Diagrams, normalized database schemas, and databases. |

(10 rows affected)

Completion time: 2024-12-04T13:17:55.5377923-05:00

100 %

Query executed successfully.

Ln 298 Col 1 Ch 1 INS

(10 rows affected)

Q2. List all PLOs and the number of courses that teach each PLO.

```
SELECT COUNT(PLO_Bridge.Course_code) AS Course_Count , PLO.PLO  
FROM PLO  
LEFT JOIN PLO_Bridge ON PLO.PLO_ID = PLO_Bridge.PLO_ID  
GROUP BY PLO.PLO;
```

BTD210 - Database Design Principles

```
Results
Course_Count_PLO
-----
0   Act ethically and responsibly with public welfare and environmental protection as a guiding professional practice.
1   Apply mathematics, natural sciences, and engineering fundamentals to solve engineering problems.
0   Communicate complex engineering problems and solutions to fellow software engineers and designers as well as non-technical audiences.
1   Create software engineering solutions that satisfy technical and business requirements.
2   Design an optimal solution using artificial intelligence, data mining, and machine learning tools for complex and open-ended problems.
0   Employ interpersonal, teambuilding, and leadership skills to solve problems independently and in diverse teams.
0   Plan and manage the scope, cost, timing, and quality of the project for success as defined by the project stakeholders.
0   Utilize investigative practices and self-awareness techniques to identify and pursue lifelong learning opportunities within their field of study and more broadly.

Warning: Null value is eliminated by an aggregate or other SET operation.

(8 rows affected)

Completion time: 2024-12-04T13:22:07.7240447-05:00
```

(8 rows affected)

Q3. List publishers for all textbooks of all courses.

```
SELECT Book.Publisher
FROM Book
JOIN Book_Bridge ON Book.BOOK_ID = Book_Bridge.BOOK_ID;
```

```
Results
Publisher
-----
McGraw-Hill Education
McGraw-Hill
Cambridge University Press
Course Technology
NULL

(5 rows affected)

Completion time: 2024-12-04T13:25:33.0054736-05:00
```

(5 rows affected)

Q4. For all courses offered in 2247, list the course code, title, the name of the professor who developed the course, and all professors who are teaching the course. (When there are multiple professors teaching the course, each should show on a separate row.)

Q5. List the course code, course title and the textbook titles for courses that teach about 'circuit's (mentioned in their description or CLOs).

BTD210 - Database Design Principles

SELECT

c.Course_code,
c.Course_Name,
b.Textbook_name

FROM

Course c

JOIN

Book_Bridge bb ON c.Course_code = bb.Course_code

JOIN

Book b ON bb.BOOK_ID = b.BOOK_ID

LEFT JOIN

CLO clo ON c.Course_code = clo.Course_code

WHERE

(c.Course_description LIKE '%circuit%' OR clo.Learning_Outcomes LIKE '%circuit%');

The screenshot shows a database query results window. The results table has three columns: Course_code, Course_Name, and Textbook_name. The data is as follows:

| Course_code | Course_Name | Textbook_name |
|-------------|-----------------------------|---|
| SEM305 | Discrete Mathematics | Discrete Mathematics and Its Applications (8th Edition) |
| SES250 | Electromagnetics | Electricity and Magnetism |
| SES250 | Electromagnetics | Electricity and Magnetism |
| SES250 | Electromagnetics | Electricity and Magnetism |
| SES250 | Electromagnetics | Electricity and Magnetism |
| SES250 | Electromagnetics | Electricity and Magnetism |
| SES250 | Electromagnetics | Electricity and Magnetism |
| SEM300 | Digital and Analog Circuits | NULL |
| SEM300 | Digital and Analog Circuits | NULL |
| SEM300 | Digital and Analog Circuits | NULL |
| SEM300 | Digital and Analog Circuits | NULL |
| SEM300 | Digital and Analog Circuits | NULL |
| SEM300 | Digital and Analog Circuits | NULL |
| SEM300 | Digital and Analog Circuits | NULL |

(14 rows affected)

Completion time: 2024-12-04T18:48:30.6613253-06:00

(14 rows affected)

Q6.Create a view that lists all course codes, course titles, and their pre-reqs (code and title).

```
CREATE VIEW CoursePreReqs AS  
SELECT c.Course_code, c.Course_Name, p.Prerequisite_ID,p.Prerequisite  
FROM Course c  
LEFT JOIN Prerequisite p ON c.Course_code = p.Course_code;
```

```
SELECT *  
FROM CoursePreReqs;
```

| Results | | | |
|-------------|-----------------------------|-----------------|--------------|
| Course_code | Course_Name | Prerequisite_ID | Prerequisite |
| BTD210 | Database Design Principles | 4 | None |
| MEC110 | Mechanics | 2 | None |
| SEH300 | Digital and Analog Circuits | 5 | SES250 |
| SEH300 | Digital and Analog Circuits | 6 | SEM305 |
| SEM305 | Discrete Mathematics | 1 | None |
| SEP200 | Object-Oriented Programming | NULL | NULL |
| SES250 | Electromagnetics | 3 | MEC110 |

(7 rows affected)

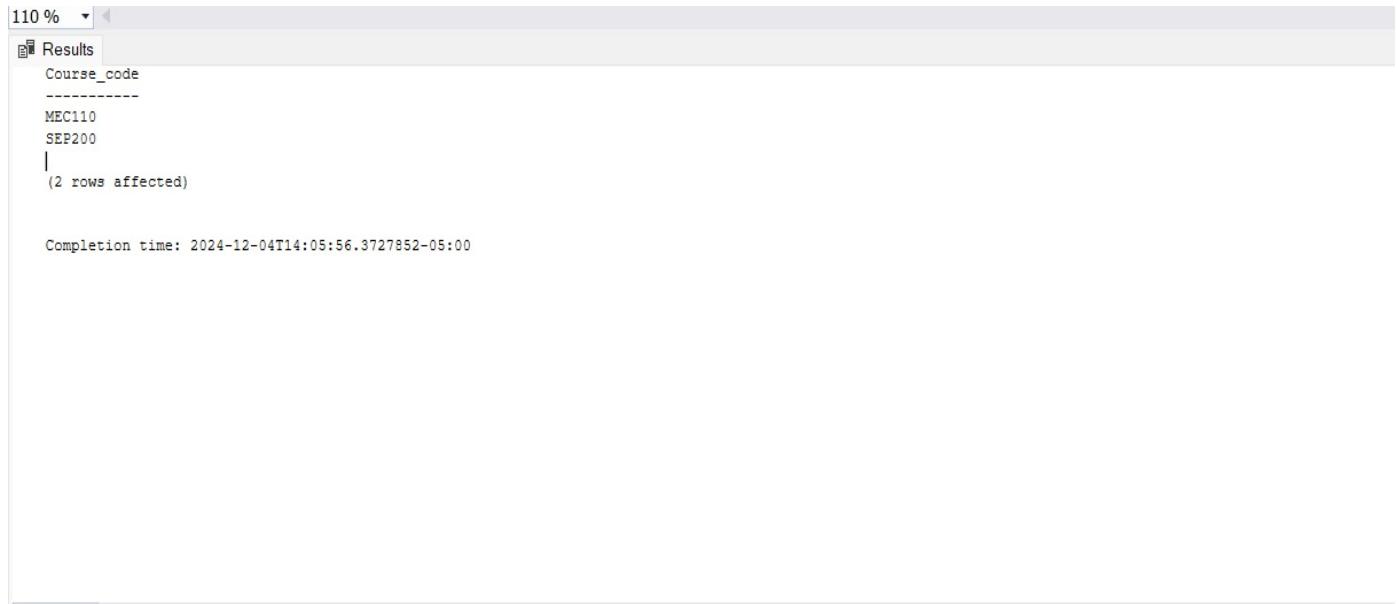
Completion time: 2024-12-04T13:58:36.3923367-05:00

100 %

(7 rows affected)

Q7. List all courses that do NOT have a laboratory component in their method of instruction.

```
SELECT Course_code
FROM Course c
WHERE NOT EXISTS (
    SELECT 1
    FROM Method_of_Instruction m
    WHERE m.Course_code = c.Course_code
    AND m.Method_of_Instruction LIKE '%Laboratory%'
);
```



The screenshot shows the SQL Server Management Studio interface. The top window displays the query results:

| Course_code |
|-------------|
| MEC110 |
| SEP200 |

Below the results, it says "(2 rows affected)". The bottom window shows the status bar with "Query executed successfully." and "(local)\SQLEXPRESS (15.0)".

Q8.What is the total number of hours of all courses in the program? Consider only the samples entered.

```
SELECT SUM(Contact_Hours_Week) AS Total_Hours  
FROM Course;
```

The screenshot shows a software interface for running SQL queries. At the top, there is a dropdown menu set to '110 %'. Below it, a toolbar has a 'Results' button selected. The main area displays the query results:

```
Results  
Total_Hours  
-----  
392  
(1 row affected)
```

At the bottom of the results pane, the completion time is shown as: Completion time: 2024-12-04T14:10:24.0803623-05:00

Q9.What is the total number of hours of all courses offered in Fall 2024?

```
SELECT  
    SUM(c.Contact_Hours_Week) AS Total_Hours  
FROM  
    Course c  
JOIN  
    Section_Bridge sb ON c.Course_code = sb.Course_code  
JOIN  
    Term t ON sb.Term_Code = t.Term_Code  
WHERE
```

```
t.Term = 2247;
```



110 % ▾

Results

| Total_Hours |
|-------------|
| 392 |

(1 row affected)

Completion time: 2024-12-04T14:18:49.3656715-05:00

110 % ▾

Query executed successfully.

Q10. Which GA is taught in the highest number of courses? Show this GA and the number of courses it is taught in.

SELECT

```
ga.Graduate_attributes AS GA,  
COUNT(DISTINCT gb.Course_code) AS NumberOfCourses
```

FROM

```
Graduate_attributes ga
```

JOIN

```
Graduate_Bridge gb ON ga.Graduate_ID = gb.Graduate_ID
```

GROUP BY

```
ga.Graduate_attributes
```

HAVING

```
COUNT(DISTINCT gb.Course_code) = (  
    SELECT MAX(CourseCount)
```

```
FROM (
    SELECT COUNT(DISTINCT gb.Course_code) AS CourseCount
    FROM Graduate_Bridge gb
    JOIN Graduate_attributes ga ON gb.Graduate_ID = ga.Graduate_ID
    GROUP BY ga.Graduate_attributes
) AS Counts
);
```

| Results | |
|------------------|-----------------|
| GA | NumberOfCourses |
| Knowledge Base | 5 |
| (1 row affected) | |

Completion time: 2024-12-04T14:23:24.9299255-05:00

Q11. List all GAs and the highest semester they are taught in.

```
SELECT
    ga.Graduate_attributes AS GA,
    MAX(t.Term) AS HighestSemester
FROM
    Graduate_Bridge gb
JOIN
    Graduate_attributes ga ON gb.Graduate_ID = ga.Graduate_ID
JOIN
```

Section_Bridge sb ON gb.Course_code = sb.Course_code

JOIN

Term t ON sb.Term_Code = t.Term_Code

GROUP BY

ga.Graduate_attributes;

```
Results
GA           HighestSemester
-----
Design       2247
Knowledge Base 2247
Problem Analysis 2247
Use of Engineering Tool 2247

(4 rows affected)

Completion time: 2024-12-04T16:28:57.8531381-05:00

121 %
```

Q12.What is the minimum, maximum and average number of CLOs in courses.

SELECT

 MIN(CLOCOUNT) AS MinCLOs,

 MAX(CLOCOUNT) AS MaxCLOs,

 AVG(CLOCOUNT) AS AvgCLOs

FROM (

 SELECT

 Course_code,

 COUNT(CLO_ID) AS CLOCOUNT

 FROM

 CLO

 GROUP BY

 Course_code

) AS CLOCOUNTS;

The screenshot shows a database query results window. The results table has three columns: MinCLOs, MaxCLOs, and AvgCLOs. The data row shows values 6, 10, and 7 respectively. Below the table, a message indicates '(1 row affected)'. At the bottom of the window, a status bar shows '121 %' and a success message 'Query executed successfully.' The status bar also displays 'Ln 388' and 'Col 1'.

| MinCLOs | MaxCLOs | AvgCLOs |
|---------|---------|---------|
| 6 | 10 | 7 |

(1 row affected)

Completion time: 2024-12-04T16:25:28.6787385-05:00

121 %

Query executed successfully.

Ln 388 Col 1

Q13.Create a view named 'DesignInCLO' which lists course codes and the number of times the word 'design' is mentioned in the CLOs of each course (let's call this column nDesign).

```
CREATE VIEW DesignInCLO AS
```

```
SELECT
```

```
    Course_code,
```

```
    COUNT(*) AS nDesign
```

```
FROM
```

```
    CLO
```

```
WHERE
```

```
    Learning_Outcomes LIKE '%design%'
```

```
GROUP BY
```

```
    Course_code;
```

The screenshot shows a database query editor window. The main pane displays the SQL code for creating a view:

```
CREATE VIEW DesignInCLO AS
SELECT
    Course_code,
    COUNT(*) AS nDesign
FROM
    CLO
WHERE
    Learning_Outcomes LIKE '%design%'
GROUP BY
    Course_code;
```

Below the code, a message bar indicates the command was completed successfully:

Commands completed successfully.

Completion time: 2024-12-04T14:32:57, 5431494-nhj0t.

A green status bar at the bottom of the window also shows the message:

Query executed successfully.

The screenshot shows a SQL query window with the following content:

```
SELECT *  
FROM DesignInCLO;
```

Results:

| Course_code | nDesign |
|-------------|---------|
| SEH300 | 2 |
| SEM305 | 1 |
| SES250 | 1 |

(3 rows affected)

Completion time: 2024-12-04T14:33:01.9340101-05:00

At the bottom, a message bar indicates: "Query executed successfully."

Q14. List course code, nDesign and whether the course is checked off for 'design' in GAs for all courses that nDesign is more than zero. Use the DesignInCLO view created in the previous question.

SELECT

 dic.Course_code,

 dic.nDesign,

CASE

 WHEN COUNT(ga.Graduate_ID) > 0 THEN 'Yes'

 ELSE 'No'

 END AS CheckedForDesign

FROM

 DesignInCLO dic

LEFT JOIN

Graduate_Bridge gb ON dic.Course_code = gb.Course_code

LEFT JOIN

Graduate_attributes ga ON gb.Graduate_ID = ga.Graduate_ID AND ga.Graduate_attributes LIKE '%design%'

WHERE

dic.nDesign > 0

GROUP BY

dic.Course_code, dic.nDesign;

| Results | | |
|---|---------|------------------|
| Course_code | nDesign | CheckedForDesign |
| SEH300 | 2 | No |
| SEM305 | 1 | No |
| SES250 | 1 | No |
| Warning: Null value is eliminated by an aggregate or other SET operation. | | |
| (3 rows affected) | | |

121 %

Query executed successfully.

Q15. Calculate the percentage of CLOs per course that are at 'APPLY' level in Bloom's taxonomy. For the purpose of this assignment, A CLO is at 'APPLY' level if it uses contains one or more of these verbs: (Apply, Use, Implement, Demonstrate, Interpret, Execute, Solve, Calculate).

WITH QUESTIONFIF AS (

SELECT

Course_code,

COUNT(*) AS Apply_CLO_Count,

(SELECT COUNT(*) FROM CLO lo2 WHERE lo2.Course_code = lo.Course_code) AS Total_CLO_Count,

CONVERT(DECIMAL(5,2), COUNT(*) * 100.0 /

(SELECT COUNT(*) FROM CLO lo2 WHERE lo2.Course_code = lo.Course_code)) AS Applied_Percentage

FROM CLO lo

WHERE

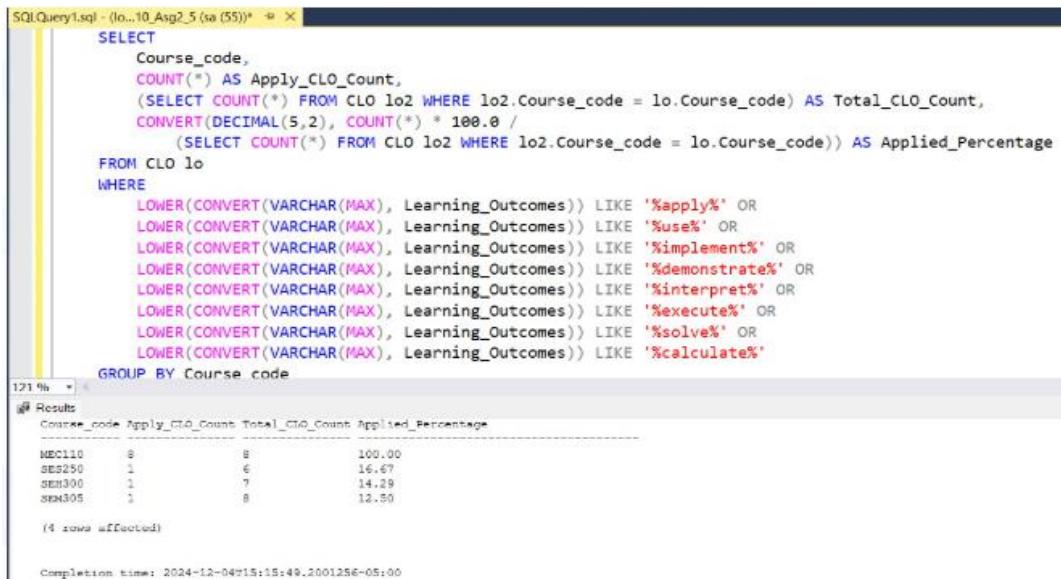
BTD210 - Database Design Principles

```
LOWER(CONVERT(VARCHAR(MAX), Learning_Outcomes)) LIKE '%apply%' OR
LOWER(CONVERT(VARCHAR(MAX), Learning_Outcomes)) LIKE '%use%' OR
LOWER(CONVERT(VARCHAR(MAX), Learning_Outcomes)) LIKE '%implement%' OR
LOWER(CONVERT(VARCHAR(MAX), Learning_Outcomes)) LIKE '%demonstrate%' OR
LOWER(CONVERT(VARCHAR(MAX), Learning_Outcomes)) LIKE '%interpret%' OR
LOWER(CONVERT(VARCHAR(MAX), Learning_Outcomes)) LIKE '%execute%' OR
LOWER(CONVERT(VARCHAR(MAX), Learning_Outcomes)) LIKE '%solve%' OR
LOWER(CONVERT(VARCHAR(MAX), Learning_Outcomes)) LIKE '%calculate%'

GROUP BY Course_code

)

SELECT
    Course_code,
    Apply_CLO_Count,
    Total_CLO_Count,
    Applied_Percentage
FROM QUESTIONFIF
ORDER BY Applied_Percentage DESC;
```



The screenshot shows the SQL Query window with the following content:

```
SQLQuery1.sql - (lo...10_Asg2_5 (sa (55))*) 121 %
```

```
SELECT
    Course_code,
    COUNT(*) AS Apply_CLO_Count,
    (SELECT COUNT(*) FROM CLO lo2 WHERE lo2.Course_code = lo.Course_code) AS Total_CLO_Count,
    CONVERT(DECIMAL(5,2), COUNT(*) * 100.0 /
        (SELECT COUNT(*) FROM CLO lo2 WHERE lo2.Course_code = lo.Course_code)) AS Applied_Percentage
FROM CLO lo
WHERE
    LOWER(CONVERT(VARCHAR(MAX), Learning_Outcomes)) LIKE '%apply%' OR
    LOWER(CONVERT(VARCHAR(MAX), Learning_Outcomes)) LIKE '%use%' OR
    LOWER(CONVERT(VARCHAR(MAX), Learning_Outcomes)) LIKE '%implement%' OR
    LOWER(CONVERT(VARCHAR(MAX), Learning_Outcomes)) LIKE '%demonstrate%' OR
    LOWER(CONVERT(VARCHAR(MAX), Learning_Outcomes)) LIKE '%interpret%' OR
    LOWER(CONVERT(VARCHAR(MAX), Learning_Outcomes)) LIKE '%execute%' OR
    LOWER(CONVERT(VARCHAR(MAX), Learning_Outcomes)) LIKE '%solve%' OR
    LOWER(CONVERT(VARCHAR(MAX), Learning_Outcomes)) LIKE '%calculate%'

GROUP BY Course_code
```

Results

| Course_code | Apply_CLO_Count | Total_CLO_Count | Applied_Percentage |
|-------------|-----------------|-----------------|--------------------|
| MEC110 | 8 | 8 | 100.00 |
| SES250 | 1 | 6 | 16.67 |
| SEM300 | 1 | 7 | 14.29 |
| SEM305 | 1 | 8 | 12.50 |

(4 rows affected)

Completion time: 2024-12-04 15:15:49.2001256-05:00

Part IV: Group Work

- 1- Add this declaration on the top of your file.

We, Dhruv Chotalia , Rudra Patel , Navdisha Bhakri declare that the attached assignment is our own work in accordance with the Seneca Academic Policy. No part of this assignment has been copied manually or electronically from any other source (including web sites) **or distributed to other students.**

- 2- Specify what each member has done towards the completion of this work:

| | Name | Task(s) |
|---|-----------------|-----------|
| 1 | Dhruv Chotalia | All tasks |
| 2 | Navdisha Bhakri | All tasks |
| 3 | Rudra Patel | All tasks |