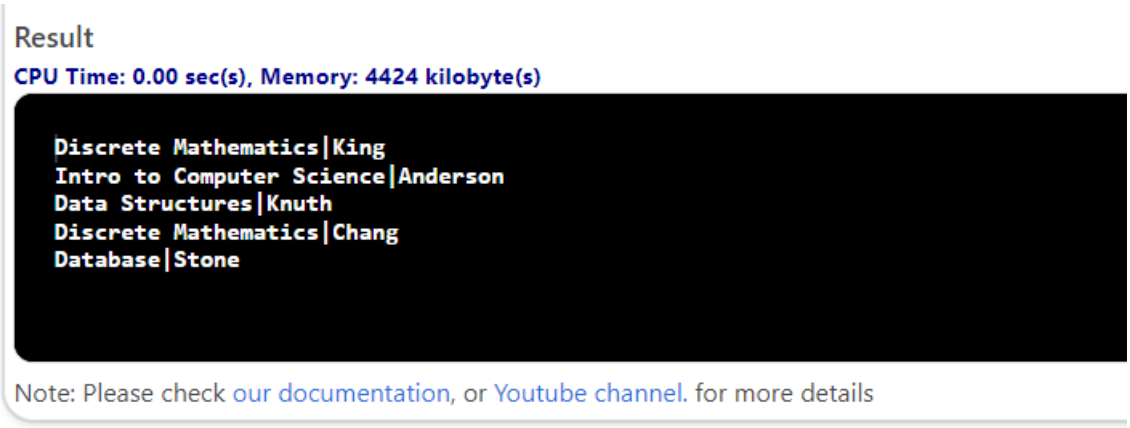
TASK 1:

```
CREATE TABLE STUDENT (
   Name VARCHAR(50),
   Student_number INT PRIMARY KEY,
   Class VARCHAR(50),
   Major VARCHAR(50)
);

CREATE TABLE COURSE (
   Course_name VARCHAR(50),
   Course_number VARCHAR(50) PRIMARY KEY,
   Credit_hours INT,
   Department VARCHAR(50)
);

CREATE TABLE SECTION (
   Section_identifier INT PRIMARY KEY,
   Course_number VARCHAR(50),
   Semester VARCHAR(50),
   Year INT,
   Instructor VARCHAR(50),
   FOREIGN KEY (Course_number) REFERENCES COURSE(Course_number)
);

CREATE TABLE GRADE_REPORT (
   Student_number INT,
   Section_identifier INT,
   Grade VARCHAR(5),
   PRIMARY KEY (Student_number, Section_identifier),
   FOREIGN KEY (Student_number) REFERENCES
STUDENT(Student_number),
   FOREIGN KEY (Section_identifier) REFERENCES Section(Section_identifier)
);

CREATE TABLE PREREQUISITE (
   Course_number VARCHAR(50),
   Prerequisite_number VARCHAR(50),
   PRIMARY KEY (Course_number, Prerequisite_number),
   FOREIGN KEY (Course_number) REFERENCES COURSE(Course_number),
   FOREIGN KEY (Prerequisite_number) REFERENCES
COURSE(Course_number)
);

# add data
```

```
-- Insert data into STUDENT table
INSERT INTO STUDENT (Name, Student_number, Class, Major) VALUES
('Smith', 17, '1', 'CS'),
('Brown', 8, '2', 'CS');

INSERT INTO COURSE (Course_name, Course_number, Credit_hours,
Department) VALUES
('Intro to Computer Science', 'CS1310', 4, 'CS'),
('Data Structures', 'CS3320', 4, 'CS'),
('Discrete Mathematics', 'MATH2410', 3, 'MATH'),
('Database', 'CS3380', 3, 'CS')
;

INSERT INTO SECTION (Section_identifier, Course_number, Semester, Year,
Instructor) VALUES
(85, 'MATH2410', 'Fall', 07, 'King'),
(92, 'CS1310', 'Fall', 07, 'Anderson'),
(102, 'CS3320', 'Spring', 08, 'Knuth'),
(112, 'MATH2410', 'Fall', 08, 'Chang'),
(119, 'CS1310', 'Fall', 08, 'Anderson'),
(135, 'CS3380', 'Fall', 08, 'Stone')
;

INSERT INTO GRADE_REPORT (Student_number, Section_identifier, Grade)
VALUES
(17, 112, 'B'),
(17, 119, 'C'),
(8, 85, 'A'),
(8, 92, 'A'),
(8, 102, 'B'),
(8, 135, 'A')
;

INSERT INTO PREREQUISITE (Course_number, Prerequisite_number)
VALUES
('CS3380', 'CS3320'),
('CS3380', 'MATH2410'),
('CS3320', 'CS1310')
;

# a
SELECT DISTINCT Course_name, Instructor
FROM COURSE JOIN SECTION
ON COURSE.Course_number = SECTION.Course_number;
```



b

```
SELECT SECTION.Section_identifier, Student_number,
SECTION.Course_number, Semester, Year
FROM COURSE JOIN SECTION
ON COURSE.Course_number = SECTION.Course_number
JOIN GRADE_REPORT
ON SECTION.Section_identifier = GRADE_REPORT.Section_identifier;
```

c

```
SELECT Name, STUDENT.Student_number, Major,
COUNT(GRADE_REPORT.section_identifier) AS Number_of_sections
FROM STUDENT JOIN GRADE_REPORT
ON STUDENT.Student_number = GRADE_REPORT.Student_number
GROUP BY Name, STUDENT.Student_number, Major
```

HAVING COUNT(Section_identifier) > 2;

TASK 2:

There are several steps to convert the **Entity Relation Diagram** to **Relational Schema**. First, we must look for strong entities, primary key (PK) constraints, and weak entities.

Strong Entities

- Bank Entity

Code (PK)	Name	Address

- Loan Entity

Loan Number (PK)	Amount	Type

- Account Entity

Account Number (PK)	Balance	Type

- Customer Entity

SSN (PK)	Name	Address	Phone

Weak Entities

- Bank_Branch Entity

Branch Number (PK)	Address

Now, we have to look for the relationships between the entities, **1:1**, **1:N**, **M:N**, and **Multi-Valued Attributes**. Since there are no 1:1 and Multi-Valued Attributes, so we only convert 1:N and M:N binary relationships into relations.

Conversion of 1:N

As Bank and Bank_Branch have a 1:N relation, add the Primary Key (PK) of the Bank entity into the Bank_Branch entity, which is Foreign Key(FK) for Bank_Branch. A similar goes for Bank_Branch and Account entities as well as for Bank_Branch and Loan entities

- Bank_Branch Entity

Branch Number (PK)	Address	Code (FK)

- Account Entity

Account Number (PK)	Balance	Type	Branch Number (FK)

- Loan Entity

Loan Number (PK)	Amount	Type	Branch Number (FK)

Conversion of M:N

Converting M:N Relationship means creating a new Entity with the relationship name and adding primary keys of both entities along with descriptive attributes if had**.**

- Customers_Account Entity

Account Number (FK refers from Account)	Customer SSN (SSN refers from Customer)

- Customers_loan Entity

Loan Number (FK refers from Loan)	Customer SSN (SSN refers from Customer)

Final Relational Schema

- Bank Entity

Code (PK)	Name	Address

- Bank_Branch Entity

Branch Number (PK)	Address	Code (FK)

- Account Entity

Account Number (PK)	Balance	Type	Branch Number (FK)

- Loan Entity

Loan Number (PK)	Amount	Type	Branch Number (FK)

- Customer Entity

SSN (PK)	Name	Address	Phone

- Customers_Account Entity

Account Number (FK refers from Account)	Customer SSN (SSN refers from Customer)

- Customers_loan Entity

Loan Number (FK refers from Loan)	Customer SSN (SSN refers from Customer)