**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')

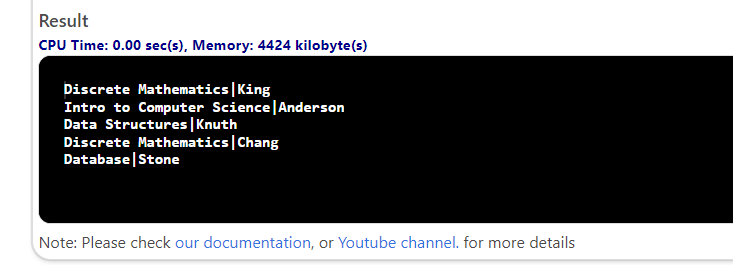
;

# 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) |