a) Topic: School Subjects

b) 3 tables:
Maths (me)
Physics (Richi)
CS (David)

C) For me:
Maths tables:
Student: (StudentID:TEXT, SFirstName:TEXT, SSurname:TEXT, Gender:TEXT, CourseID:TEXT FOREIGN KEY: CourseID REFERENCING: Courses, PRIMARY KEY StudentID)

Teacher: (TeacherID:TEXT, TFirstName:TEXT, TSurname:TEXT, Gender:TEXT, TTitle:TEXT, PRIMARY KEY: TeacherID)

Courses: (CourseID:TEXT, TeacherID:TEXT, Title:TEXT, Credits:INT, YearGroup:INT, FOREIGN KEY: TeacherID REFERENCING: Teacher, PRIMARY KEY: CourseID)

StudentsEnrolled(StudentID:TEXT, FirstName:TEXT, Surname:TEXT, Gender:TEXT, CourseID:TEXT FOREIGN KEY: CourseID REFERENCING: Courses, PRIMARY KEY StudentID, ON DELETE CASCADE, CourseID CANNOT BE NULL)

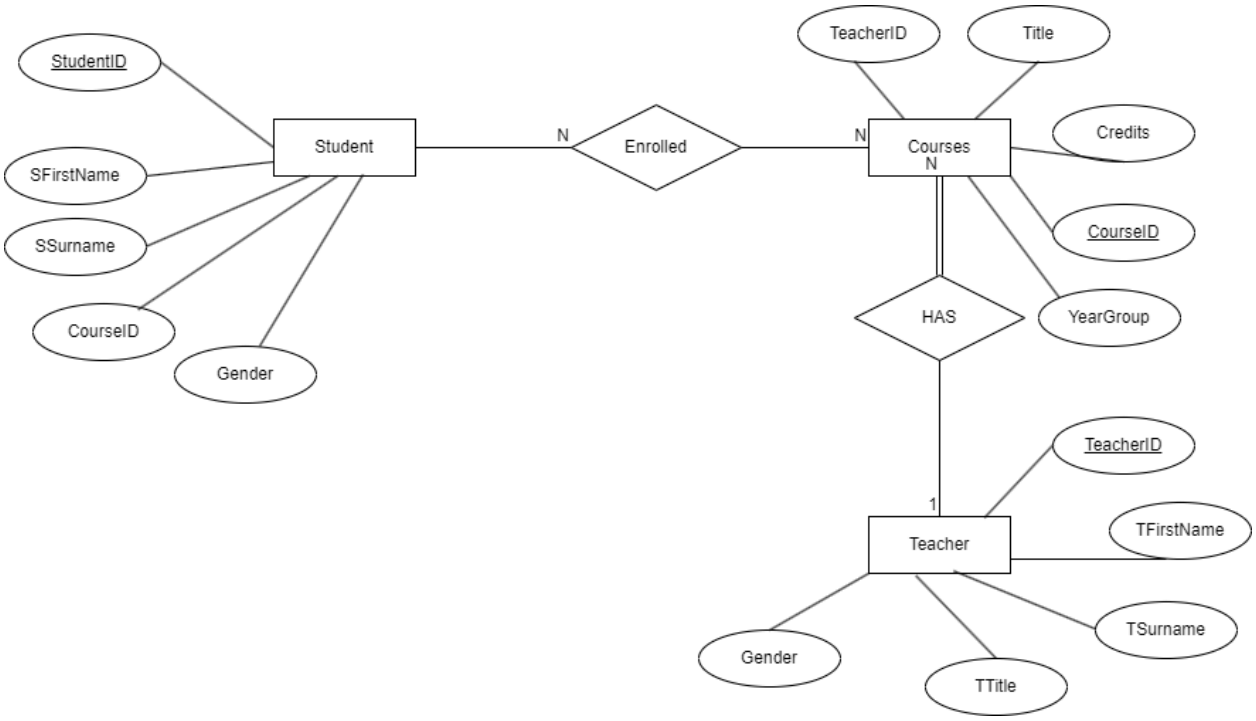
CoursesHas(CourseID:TEXT, TeacherID:TEXT, Title:TEXT, Credits:INT, YearGroup:INT, FOREIGN KEY: TeacherID REFERENCING: Teacher, PRIMARY KEY: CourseID, ON DELETE CASCADE, TeacherID CANNOT BE NULL)

Relationships:
HAS (Teacher and Courses)
Enrolled (Student and Courses)

D) The entity domain is Maths, The entities are “Student”, “Teacher” and “Courses” The HAS relationship ensures that every course must have a corresponding teacher teaching it, ensuring a course cannot exist without a teacher, courses has foreign key TeacherID , expressing the multiplicity that a teacher must teach at least 1 course. “Enrolled” relationship connects “Student” and “Courses" ensures every student enrolled in a course must exist, ensuring that StudentID exists within the student table. Student has foreign key CourseID referencing the Course table. Courses has foreign key TeacherID referencing the Teacher table

C)

Entity Relationship Diagram



D)

Normalisation

Student is in 3NF as it is in 2nd NF and has no transitive dependencies for non-prime attributes

Teacher is in 3NF as it is in 2nd NF and has no transitive dependencies for non-prime attributes.

Courses are in 2NF as it has transitive dependencies, YearGroup depends on title and CourseID so it depends on the primary key however also the title, therefore to normalise this you'd need a separate table relating YearGroup and title.

e)

DDL

(CREATE TABLE IF NOT EXISTS Teacher (
TeacherID INT NOT NULL,

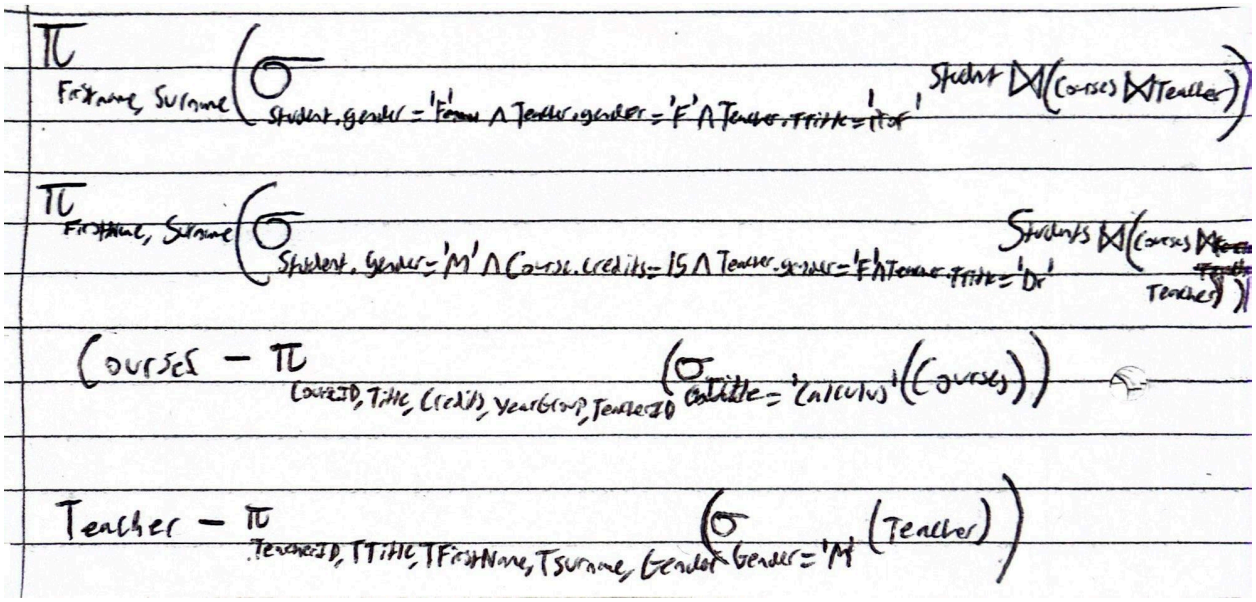
```
        TTitle VARCHAR(20),
        TFirstName VARCHAR(255),
        TSurname VARCHAR(255),
        Gender VARCHAR(20),
        PRIMARY KEY (TeacherID))

);


(CREATE TABLE IF NOT EXISTS Courses(
        CourseID INT NOT NULL,
        Title VARCHAR(255),
        Credits INT,
        YearGroup INT,
        TeacherID INT NOT NULL,
        PRIMARY KEY(CourseID),
        FOREIGN KEY (TeacherID) REFERENCES Teacher(TeacherID) ON DELETE
RESTRICT)
                );

(CREATE TABLE IF NOT EXISTS Student (
        StudentID INT NOT NULL,
        SFirstName VARCHAR(255),
        SSurname VARCHAR(255),
        Gender VARCHAR(20),
        CourseID INT NOT NULL,
        PRIMARY KEY (StudentID),
        FOREIGN KEY (CourseID) REFERENCES Courses(CourseID) ON DELETE
RESTRICT)"
                );
```

f)



g)

DML statements

QUERY 1

Select Male Students Doing a course worth 15 credits taught by a female teacher with the title Dr

```
SELECT s.SFirstName, s.SSurname
FROM Student s
JOIN Courses c ON s.CourseID = c.CourseID
JOIN Teacher t ON c.TeacherID = t.TeacherID
WHERE s.Gender = ‘M’
AND c.Credits = 15
AND t.Gender = ‘F’
AND t.Title = ‘Dr’
```

```
GROUP BY s.StudentID, s.SFirstName, s.SSurname
HAVING COUNT(DISTINCT T.TeacherID) = 1;
```

Query 2

Select Female students taught by one female teacher with the title Prof

```
SELECT s.FirstName, s,Surname
FROM Student AS s
INNER JOIN Courses C ON S.CourseID = C.CourseID
INNER JOIN Teacher T ON C.TeacherID = T.TeacherID
WHERE S.Gender = ‘F’ AND T.Gender = ‘F’ AND T.Title = ‘Prof’
GROUP BY S.FirstName, S.Surname
HAVING COUNT(DISTINCT T.TeacherID) = 1;
```

Query 3

Delete Teachers whose gender is male from teacher

```
DELETE FROM Teacher WHERE Gender = ‘M’
```

This is rejected because of the referential integrity in the Courses that has ON DELETE RESTRICT clause

Query 4

Delete Course “Calculus” from courses

```
DELETE FROM Courses WHERE Title = ‘Calculus’
```

Due to the referential integrity in the Student table for the foreign key CourseID on delete restrict, the row will not be deleted and will return an error.