

## CPSC 2221 Group Project

### Milestone 3

**Table:** StudentEnroll

**Description:** Student entity merged with relationship of enrollment in department

**Table attribute:**

StudentID, PK  
StudentPassword  
StudentName  
StudentAddress  
EnrolledInDept, FK(Department(Code))

**Functional Dependency:**

StudentID -> ALL (PK)

**Table:** Department

**Description:** Department of the faculty

**Table attribute:**

Code, PK  
Name

**Functional Dependency:**

Code -> ALL (PK)  
Name -> ALL (CK)

**Table:** CourseTeachBy

**Description:** Course Entity merged with relationship teach by instructor

**Table Attribute:**

CourseID, PK  
DepartmentCode, PK, FK(Department(code))  
Room  
Time  
InstructorID, FK(Instructor(ID))

**Functional Dependency:**

CourseID, DepartmentCode -> ALL (PK)

**Table:** CourseBelongsDept

**Description:** Relationship of Course Belongs to Department

**Table Attribute:**

DepartmentCode, FK(Department(Code)), PK  
CourseID, FK(Course(ID), PK  
DepartmentName, FK(Department(Name))

**Functional Dependency:**

DepartmentCode, CourseID -> ALL (PK)  
DepartmentName, CourseID -> ALL (CK)  
DepartmentCode -> DepartmentName  
DepartmentName -> DepartmentCode

**Table:** Waitlist

**Description:** relationship for student waiting in the waitlist of a course

**Table Attribute:**

DepartmentCode, FK(Department(Code))  
CourseID, FK(Course(ID))  
PositionInLine  
StudentID, FK(Student(ID))

**Functional Dependency:**

DeptCode, CourseID, PositionInLine -> StudentID (PK)  
DeptCode, CourseID, StudentID -> PositionInLine (CK)

**Table:** StudentTakeCourse

**Description:** Student take course relationship

**Table Attribute:**

StudentID, FK(Student(ID))  
DepartmentCode, FK(Department(Code))  
CourseID, FK(Course(ID))

**Function Dependency:**

N/A

**Table:** InvoicePaidBy

**Description:** Invoice entity merged with relationship paid by

**Table Attribute:**

StudentID, PK, FK(Student(ID))  
TransactionID, PK  
PaymentMethod  
RegisteredCredit  
Amount

**Functional Dependency:**

StudentID, TransactionID -> ALL (PK)  
RegisteredCredit -> Amount

**Table:** Prerequisite

**Description:** Self Referencing relationship of Course

**Table Attribute**

DepartmentCode, FK(Department(Code))

CourseID

PreDeptCode, FK(Department(Code))

PreCourseID

**Functional Dependency:**

N/A

**Table:** Instructor

**Description:** Instructor entity

**Table Attribute:**

ID, PK

Name

**Functional Dependency:**

ID -> Name (PK)

**Table:** InstructorWithPHD

**Description:** subclass for instructor having PHD

**Table Attribute:**

Inherited attribute from superclass

Institution

**Functional Dependency:**

ID -> Institution

**Table:** InstructorWithNoPHD

**Description:** subclass for instructor do not have PHD

**Table Attribute:**

Inherited attribute from superclass

YearsOfExperience

**Functional Dependency:**

ID -> YearsOfExperience

## Normalization

**Table:** StudentEnroll

**Description:** Student entity merged with relationship of enrollment in department

**Table attribute:**

StudentID, PK  
StudentPassword  
StudentName  
StudentAddress  
EnrolledInDept, FK(Department(Code))

**Functional Dependency:**

StudentID -> ALL (PK)

**This table is already in BCNF**

**Table:** Department

**Description:** Department of the faculty

**Table attribute:**

Code, PK  
Name

**Functional Dependency:**

Code -> ALL (PK)  
Name -> ALL (CK)

**This table is already in BCNF**

**Table:** CourseTeachBy

**Description:** Course Entity merged with relationship teach by instructor

**Table Attribute:**

CourseID, PK  
DepartmentCode, PK, FK(Department(code))  
Room  
Time  
InstructorID, FK(Instructor(ID))

**Functional Dependency:**

CourseID, DepartmentCode -> ALL (PK)

**This table is already in BCNF**

**Table:** CourseBelongsDept

**Description:** Relationship of Course Belongs to Department

**Table Attribute:**

DepartmentCode, FK(Department(Code)), PK

CourseID, FK(Course(ID), PK

DepartmentName, FK(Department(Name))

**Functional Dependency:**

DepartmentCode, CourseID -> ALL (PK)

DepartmentName, CourseID -> ALL (CK)

DepartmentCode -> DepartmentName

DepartmentName -> DepartmentCode

**This table is already in 3NF**

**However this table is redundant since CourseTeachBy table capture all the information in this table, it is not necessary to keep this table, therefore this table will be abundant**

**Table:** Waitlist

**Description:** relationship for student waiting in the waitlist of a course

**Table Attribute:**

DepartmentCode, PK, FK(Department(Code))

CourseID, PK, FK(Course(ID))

PositionInLine, PK

StudentID, FK(Student(ID))

**Functional Dependency:**

DeptCode, CourseID, PositionInLine -> StudentID (PK)

DeptCode, CourseID, StudentID -> PositionInLine (CK)

**This table is already in BCNF**

**Table:** StudentTakeCourse

**Description:** Student take course relationship

**Table Attribute:**

StudentID, FK(Student(ID))

DepartmentCode, FK(Department(Code))

CourseID, FK(Course(ID))

**Function Dependency:**

N/A

**This table is already in BCNF**

**Table:** InvoicePaidBy

**Description:** Invoice entity merged with relationship paid by

**Table Attribute:**

StudentID, PK, FK(Student(ID))

TransactionID, PK

PaymentMethod

RegisteredCredit

Amount

**Functional Dependency:**

StudentID, TransactionID -> ALL (PK)

RegisteredCredit -> Amount

RegisteredCredit -> Amount violate BCNF, decompose

Table1(RegisteredCredit, Amount)

Table2(StudentID, TransactionID, PaymentMethod, RegisteredCredit)

**Now Table is BCNF**

**Table:** Prerequisite

**Description:** Self Referencing relationship of Course

**Table Attribute**

DepartmentCode, FK(Department(Code))

CourseID

PreDeptCode, FK(Department(Code))

PreCourseID

**Functional Dependency:**

N/A

**This table is already in BCNF**

**Table:** Instructor

**Description:** Instructor entity

**Table Attribute:**

ID, PK

Name

**Functional Dependency:**

ID -> Name (PK)

**This table is already in BCNF**

**Table:** InstructorWithPHD

**Description:** subclass for instructor having PHD

**Table Attribute:**

Inherited attribute from superclass

Institution

**Functional Dependency:**

ID -> Institution

**This table is already in BCNF**

**Table:** InstructorWithNoPHD

**Description:** subclass for instructor do not have PHD

**Table Attribute:**

Inherited attribute from superclass

YearsOfExperience

**Functional Dependency:**

ID -> YearsOfExperience

**This table is already in BCNF**

## SQL DDL to create all Tables

```
Create TABLE Department(  
    Code CHAR(20),  
    Name CHAR(20),  
    PRIMARY KEY(Code));
```

```
Create TABLE StudentEnroll(  
    ID INTEGER,  
    Password CHAR(20),  
    Name CHAR(20),  
    Address CHAR(30),  
    EnrolledInDept CHAR(20),  
    PRIMARY KEY(ID)  
    FOREIGN KEY(EnrolledInDept) REFERENCES Department(Code));
```

```
Create TABLE Instructor(  
    ID INTEGER,  
    Name CHAR(20),  
    PRIMARY KEY(ID));
```

```
Create TABLE InstructorWithPHD(  
    ID INTEGER,  
    Name CHAR(20),  
    Institution CHAR(20),  
    PRIMARY KEY(ID));
```

```
Create TABLE InstructorWithNoPHD(  
    ID INTEGER,  
    Name CHAR(20),  
    YearsOfExperience INTEGER,  
    PRIMARY KEY(ID));
```

```
Create TABLE CourseTeachBy(  
    CourseID INTEGER,  
    DepartmentCode CHAR(20),  
    Room CHAR(20),  
    Time CHAR(30),  
    InstructorID INTEGER,  
    PRIMARY KEY(CourseID, DepartmentCode)  
    FOREIGN KEY(DepartmentCode) REFERENCES Department(Code)  
    FOREIGN KEY(InstructorID) REFERENCES Instructor(ID));
```



```
Create TABLE Waitlist(  
    CourseID INTEGER,  
    DepartmentCode CHAR(20),  
    PositionInLine INTEGER,  
    StudentID INTEGER,  
    PRIMARY KEY(CourseID, DepartmentCode, PositionInLine)  
    FOREIGN KEY(DepartmentCode) REFERENCES Department(Code)  
    FOREIGN KEY(CourseID) REFERENCES CourseTeachBy(CourseID)  
    FOREIGN KEY(StudentID) REFERENCES StudentEnroll(ID));
```

```
Create TABLE StudentTakeCourse(  
    CourseID INTEGER,  
    DepartmentCode CHAR(20),  
    StudentID INTEGER,  
    FOREIGN KEY(DepartmentCode) REFERENCES Department(Code)  
    FOREIGN KEY(CourseID) REFERENCES CourseTeachBy(CourseID)  
    FOREIGN KEY(StudentID) REFERENCES StudentEnroll(ID));
```

```
Create TABLE CreditPrice(  
    RegisteredCredit INTEGER,  
    Amount REAL);
```

```
Create TABLE InvoicePaidBy(  
    StudentID INTEGER,  
    TransactionID INTEGER  
    PaymentMethod CHAR(20),  
    RegisteredCredit INTEGER,  
    PRIMARY KEY(StudentID, TransactionID)  
    FOREIGN KEY(StudentID) REFERENCES StudentEnroll(ID)  
    FOREIGN KEY(RegisteredCredit) REFERENCES CreditPrice(RegisteredCredit));
```

```
Create TABLE Prerequisite(  
    DepartmentCode CHAR(20),  
    CourseID INTEGER  
    PreDeptCode CHAR(20),  
    PreCourseID INTEGER,  
    FOREIGN KEY(DepartmentCode) REFERENCES Department(Code)  
    FOREIGN KEY(PreDeptCode) REFERENCES Department(Code)  
    FOREIGN KEY(CourseID) REFERENCES CourseTeachBy(CourseID)  
    FOREIGN KEY(PreCourseID) REFERENCES CourseTeachBy(CourseID));
```

## Populating Tables

```
INSERT INTO Department(Code, Name)
VALUES('CPSC', 'Computer Science');
INSERT INTO Department(Code, Name)
VALUES('MATH', 'Mathematics');
INSERT INTO Department(Code, Name)
VALUES('PHYS', 'Physiology');
INSERT INTO Department(Code, Name)
VALUES('COMM', 'Communication');
INSERT INTO Department(Code, Name)
VALUES('CHEM', 'Chemistry');
```

```
INSERT INTO StudentEnroll(ID, Name, Address, EnrolledInDept)
VALUES(1001, 'John', '1002 Some Street, That City', 'CPSC' );
INSERT INTO StudentEnroll(ID, Name, Address, EnrolledInDept)
VALUES(1002, 'Mary', '1022 This Street, That City', 'MATH' );
INSERT INTO StudentEnroll(ID, Name, Address, EnrolledInDept)
VALUES(1003, 'Jimmy', '1202 Owell Street, Some City', 'PHYS' );
INSERT INTO StudentEnroll(ID, Name, Address, EnrolledInDept)
VALUES(1004, 'Tony', '1402 Man Street, Woman City', 'COMM' );
INSERT INTO StudentEnroll(ID, Name, Address, EnrolledInDept)
VALUES(1005, 'Jerry', '1052 IdontCare Street, Whocares City', 'CHEM' );
```

```
INSERT INTO Instructor (ID, Name)
VALUES(5001, 'Jay');
INSERT INTO Instructor (ID, Name)
VALUES(5002, 'Choy');
INSERT INTO Instructor (ID, Name)
VALUES(5003, 'Joe');
INSERT INTO Instructor (ID, Name)
VALUES(5004, 'Henry');
INSERT INTO Instructor (ID, Name)
VALUES(5005, 'Poter');
INSERT INTO Instructor (ID, Name)
VALUES(5006, 'Harry');
INSERT INTO Instructor (ID, Name)
VALUES(5007, 'Ryan');
INSERT INTO Instructor (ID, Name)
VALUES(5008, 'Romeo');
INSERT INTO Instructor (ID, Name)
VALUES(5009, 'Juliet');
INSERT INTO Instructor (ID, Name)
VALUES(5010, 'Michal');
```

```
INSERT INTO InstructorWithPHD (ID, Name, Institution)
VALUES(5001, 'Jay', 'SomeWhat University');
INSERT INTO InstructorWithPHD (ID, Name, Institution)
VALUES(5002, 'Choy', 'ALOHA College');
INSERT INTO InstructorWithPHD (ID, Name, Institution)
VALUES(5003, 'Joe', 'O Well University');
INSERT INTO InstructorWithPHD (ID, Name, Institution)
VALUES(5004, 'Henry', 'Made Up College);
INSERT INTO InstructorWithPHD (ID, Name, Institution)
VALUES(5005, 'Poter', 'Very Good University');
```

```
INSERT INTO InstructorWithNoPHD (ID, Name, YearsOfExperience)
VALUES(5006, 'Harry', 5);
INSERT INTO InstructorWithNoPHD (ID, Name, YearsOfExperience)
VALUES(5007, 'Ryan', 3);
INSERT INTO InstructorWithNoPHD (ID, Name, YearsOfExperience)
VALUES(5008, 'Romeo', 7);
INSERT INTO InstructorWithNoPHD (ID, Name, YearsOfExperience)
VALUES(5009, 'Juliet', 12);
INSERT INTO InstructorWithNoPHD (ID, Name, YearsOfExperience)
VALUES(5010, 'Michal', 22);
```

```
INSERT INTO CourseTeachBy (CourseID, DeparmentCode, Room, Time, InstructorID)
VALUES(2300, 'CPSC', 'A343', 'Monday 5.30-7.30PM', 5005);
INSERT INTO CourseTeachBy (CourseID, DeparmentCode, Room, Time, InstructorID)
VALUES(2133, 'PHYS', 'A244', 'Tuesday 5.30-7.30PM', 5003);
INSERT INTO CourseTeachBy (CourseID, DeparmentCode, Room, Time, InstructorID)
VALUES(2560, 'CHEM', 'B163', 'Friday 2.30-3.30PM', 5001);
INSERT INTO CourseTeachBy (CourseID, DeparmentCode, Room, Time, InstructorID)
VALUES(2730, 'COMM', 'C234', 'Wednesday 8.30-10.30AM', 5007);
INSERT INTO CourseTeachBy (CourseID, DeparmentCode, Room, Time, InstructorID)
VALUES(2300, 'MATH', 'T555', 'Monday 5.30-7.30PM', 5004);
```

```
INSERT INTO Waitlist (CourseID, DeparmentCode, PositionInLine, StudentID)
VALUES(2300, 'MATH', 1, 1001);
INSERT INTO Waitlist (CourseID, DeparmentCode, PositionInLine, StudentID)
VALUES(2300, 'MATH', 2, 1003);
INSERT INTO Waitlist (CourseID, DeparmentCode, PositionInLine, StudentID)
VALUES(2730 'COMM', 1, 1002);
INSERT INTO Waitlist (CourseID, DeparmentCode, PositionInLine, StudentID)
VALUES(2300, 'CPSC', 1, 1002);
INSERT INTO Waitlist (CourseID, DeparmentCode, PositionInLine, StudentID)
VALUES(2300, 'CPSC', 2, 1005);
```

```
INSERT INTO StudentTakeCourse (CourseID, DepartmentCode, StudentID)
VALUES(2300, 'CPSC', 1001);
INSERT INTO StudentTakeCourse (CourseID, DepartmentCode, StudentID)
VALUES(2300, 'CPSC', 1003);
INSERT INTO StudentTakeCourse (CourseID, DepartmentCode, StudentID)
VALUES(2300, 'CPSC', 1001);
INSERT INTO StudentTakeCourse (CourseID, DepartmentCode, StudentID)
VALUES(2300, 'MATH', 1004);
INSERT INTO StudentTakeCourse (CourseID, DepartmentCode, StudentID)
VALUES(2300, 'MATH', 1005);
```

```
INSERT INTO CreditPrice (RegisteredCredit, Amount)
VALUES(3, 120.66);
INSERT INTO CreditPrice (RegisteredCredit, Amount)
VALUES(6, 236.00);
INSERT INTO CreditPrice (RegisteredCredit, Amount)
VALUES(9, 349.89);
INSERT INTO CreditPrice (RegisteredCredit, Amount)
VALUES(12, 457.33);
INSERT INTO CreditPrice (RegisteredCredit, Amount)
VALUES(15, 566.12);
```

```
INSERT INTO InvoicePaidBy (StudentID, TransactionID, PaymentMethod, RegisteredCredit)
VALUES(1001, 62001, 'CASH', 6);
INSERT INTO InvoicePaidBy (StudentID, TransactionID, PaymentMethod, RegisteredCredit)
VALUES(1002, 62011, 'CREDIT CARD', 6);
INSERT INTO InvoicePaidBy (StudentID, TransactionID, PaymentMethod, RegisteredCredit)
VALUES(1003, 62101, 'MASTER CARD', 3);
INSERT INTO InvoicePaidBy (StudentID, TransactionID, PaymentMethod, RegisteredCredit)
VALUES(1003, 62111, 'DEBIT CARD', 12);
INSERT INTO InvoicePaidBy (StudentID, TransactionID, PaymentMethod, RegisteredCredit)
VALUES(1005, 62122, 'CHEQUE', 9);
```

```
INSERT INTO Prerequisite (DepartmentCode, CourseID, PreDeptCode, PreCourseID)
VALUES('CPSC', 2300, 'MATH', 2300);
INSERT INTO Prerequisite (DepartmentCode, CourseID, PreDeptCode, PreCourseID)
VALUES('CPSC', 2300, 'COMM', 2730);
INSERT INTO Prerequisite (DepartmentCode, CourseID, PreDeptCode, PreCourseID)
VALUES('PHYS', 2133, 'MATH', 2300);
INSERT INTO Prerequisite (DepartmentCode, CourseID, PreDeptCode, PreCourseID)
VALUES('CHEM', 2560, 'MATH', 2300);
INSERT INTO Prerequisite (DepartmentCode, CourseID, PreDeptCode, PreCourseID)
VALUES('PHYS', 2133, 'COMM', 2730);
```