## Initial mapping

ISBN (id, number, itemNo)  
Primary Key id  
Alternate Key number  
Foreign Key itemNo references Book(itemNo)  
ON UPDATE CASCADE, ON DELETE CASCADE

**Postcode**(suburb, state, country, postcode)

**Primary Key** suburb, state, country

**Address**(addressId, unit, streetNumber, streetName, suburb, state, country)

**Primary Key** addressId

**Foreign Key** suburb **references** Postcode(suburb)

**Foreign Key** state **references** Postcode(state)

**Foreign Key** country **references** Postcode(country)

**Campus**(campusId, name, addressId)

**Primary Key** campusId

**Foreign Key** addressId **references** Address(addressId)

ON UPDATE NO ACTION, ON DELETE NO ACTION

**Facility**(facilityId, name, campusId)

**Primary Key** facilityId

**Foreign Key** campusId **references** Campus(campusId)

ON UPDATE NO ACTION, ON DELETE NO ACTION

**Room**(roomNumber, facilityId, capacity, type)

**Primary Key** roomNumber, facilityId

**Foreign Key** facilityId **references** Facility(facilityId)

ON UPDATE NO ACTION, ON DELETE NO ACTION

**Person**(personId, firstName, lastName, contactNumber, qualification, highestCurrentEducation, isStudent, isStaff, addressId)

**Primary Key** personId

**Foreign Key** addressId **references** Address(addressId)

ON UPDATE NO ACTION, ON DELETE SET NULL

**Department**(departmentId, name, description, contactNumber, subDepartmentOf)

**Primary Key** departmentId

**Alternate Key** name

**DepartmentAssignment**(personId, departmentId, startDate, endDate, role)

**Primary Key** personId, departmentId, startDate

**Foreign Key** departmentId **references** Department(departmentId)

ON UPDATE NO ACTION, ON DELETE NO ACTION

**Foreign Key** personId **references** Person(personId)

ON UPDATE CASCADE, ON DELETE CASCADE

**AcademicProgramme**(academicProgrammeId, name, programTotalCredits, level, certificationAcheived)

**Primary Key** academicProgrammeId

**Alternate Key** name

**MajorMinor**(code, name, description, totalCredits)

**Primary Key** code

**Foreign Key (**academicProgrammeId) **references** academicProgramme(academicProgrammeId)

ON UPDATE NO ACTION, ON DELETE CASCADE

MajorMinorConditions(conditionId, code, condition)

**Primary Key conditionId**

**Foreign Key** code **references** MajorMinor(code)

**ProgramConvenor**(personId, academicProgrammeId, startDate, endDate)

**Primary Key** personId, academicProgrammeId, startDate

**Foreign Key** personId **references** Person(personId)

**Foreign Key** academicProgrammeId **references** academicProgrammeId(academicProgramme)

ON UPDATE NO ACTION, ON DELETE NO ACTION

**Course**(courseId, name, numberOfcredits, description)

**Primary Key** courseId

**Alternate Key** name

**Prerequisite**(prerequisite, courseId)

**Primary Key** prerequisite, courseId

**Foreign Key** prerequisite **references** Course(courseId)

ON UPDATE NO ACTION, ON DELETE NO ACTION

**Foreign Key** courseId **references** Course(courseId)

ON UPDATE NO ACTION, ON DELETE CASCADE

**CourseAcademicProgrammeAssignment**(courseId, academicProgrammeId , startDate, endDate, type)

**Primary Key** courseId, academicProgrammeId, startDate

**Foreign Key** courseId **references** Course(courseId)

ON UPDATE NO ACTION, ON DELETE CASCADE

**Foreign Key** academicProgrammeId **references** AcademicProgramme(academicProgrammeId)

ON UPDATE NO ACTION, ON DELETE CASCADE

**TimePeriod**(timePeriodId, year, isSemester, semesterTrimesterNumber)

**Primary Key** timePeriodId

**CourseOffering**(courseId, timePeriodId, campusId, courseCoordinator)

**Primary Key** courseId, timePeriodId, campusId

**Foreign Key** courseId **references** Course(courseId)

ON UPDATE NO ACTION, ON DELETE CASCADE

**Foreign Key** campusId **references** Campus(campusId)

ON UPDATE NO ACTION, ON DELETE CASCADE

**Foreign Key** timePeriodId **references** TimePeriod(timePeriodId)

ON UPDATE NO ACTION, ON DELETE CASCADE

**Foreign Key** courseCoordinator **references** Person(personId)

ON UPDATE NO ACTION, ON DELETE CASCADE

**TimeSlot**(timeSlotId, day, startTime, endTime, roomNumer, personId, courseId, timePeriodId, campusId)

**Primary Key** timeSlotId

**Foreign Key** roomNumber **references** Room(roomNumber)

ON UPDATE NO ACTION, ON DELETE NO ACTION

**Foreign Key** personId **references** Person(personId)

ON UPDATE NO ACTION, ON DELETE NO ACTION

**Foreign Key** courseId **references** CourseOffering(courseId)

ON UPDATE NO ACTION, ON DELETE NO ACTION

**Foreign Key** timePeriodId **references** CourseOffering(timePeriodId)

ON UPDATE NO ACTION, ON DELETE NO ACTION

**Foreign Key** campusId **references** CourseOffering(campusId)

ON UPDATE NO ACTION, ON DELETE NO ACTION

**StudentEnrolment**(academicProgrammeId, personId, timePeriodId, enrolmentDate, completionDate)

**Primary Key** academicProgramId, personId, timePeriodId

**Foreign Key** timePeriodId **references** TimePeriod(timePeriodId)

ON UPDATE NO ACTION, ON DELETE CASCADE

**Foreign Key** personId **references** Person(personId)

ON UPDATE NO ACTION, ON DELETE CASCADE

**Foreign Key** academicProgrammeId **references** AcademicProgramme(academicProgrammeId)

ON UPDATE NO ACTION, ON DELETE CASCADE

**StudentRegistersInCourse**(personId, courseId, timePeriodId, finalMark, finalGrade)

**Primary Key** personId, courseId, timePeriodId

**Foreign Key** personId **references** Person(personId)

ON UPDATE NO ACTION, ON DELETE CASCADE

**Foreign Key** courseId **references** CourseOffering(courseId)

ON UPDATE NO ACTION, ON DELETE CASCADE

**Foreign Key** timePeriodId **references** TimePeriodOffering(timePeriodId)

ON UPDATE NO ACTION, ON DELETE CASCADE

## Recursive Relationships

Subdepartments can be represented by having a subDepartmentOf attribute in the Department table

Prerequisites can be represented by having a separate table with the course that acts as a prerequisite and the course receiving the prerequisite.

## Normalisation

INF

MajorMinor wasn’t in 1NF because the conditions attribute could be multivalued. To make this atomic a separate table called MajorMinorCondition with the MajorMinor code as a FK and conditionId as PK.

2NF

Every relation is in 2NF because every non-candidate is fully functionally dependent on any candidate key.

3NF

The table is in 3NF because there are no non-candidate keys that are transitively dependent on candidate keys.

BCNF

After making Postcode its own table every relation is in BCNF.