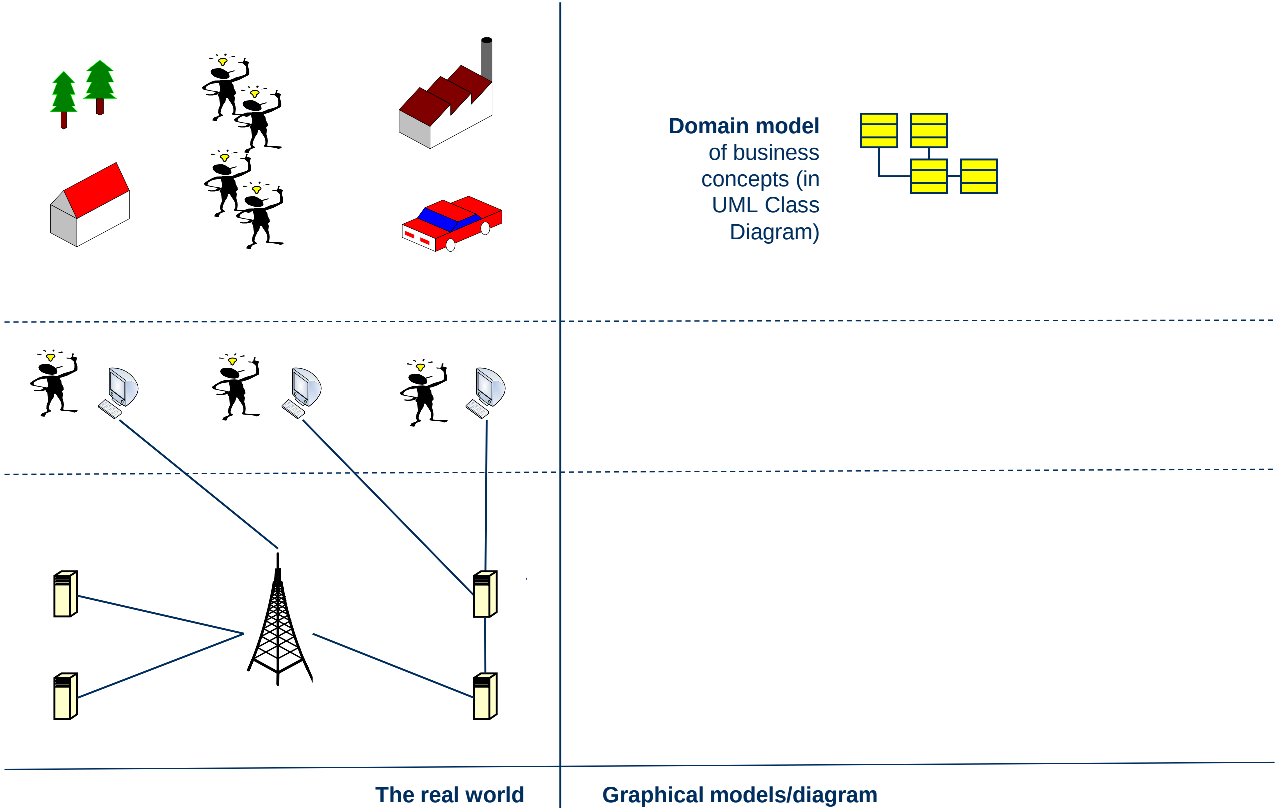
Stockholms universitet

Model Driven Method for Relational Database Design

Erik Perjons

Real World and Models

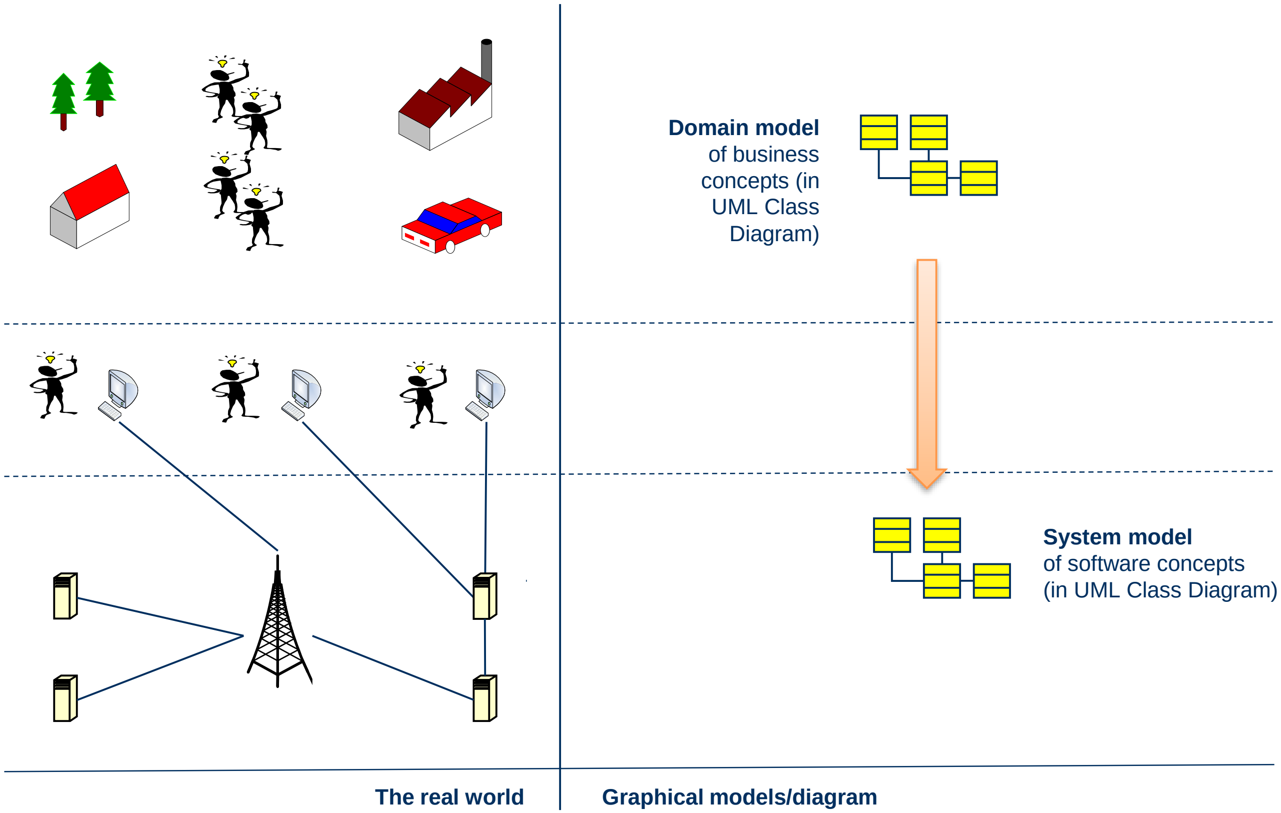


Domain model of business concepts (in UML Class Diagram)

The real world

Graphical models/diagram

Real World and Models



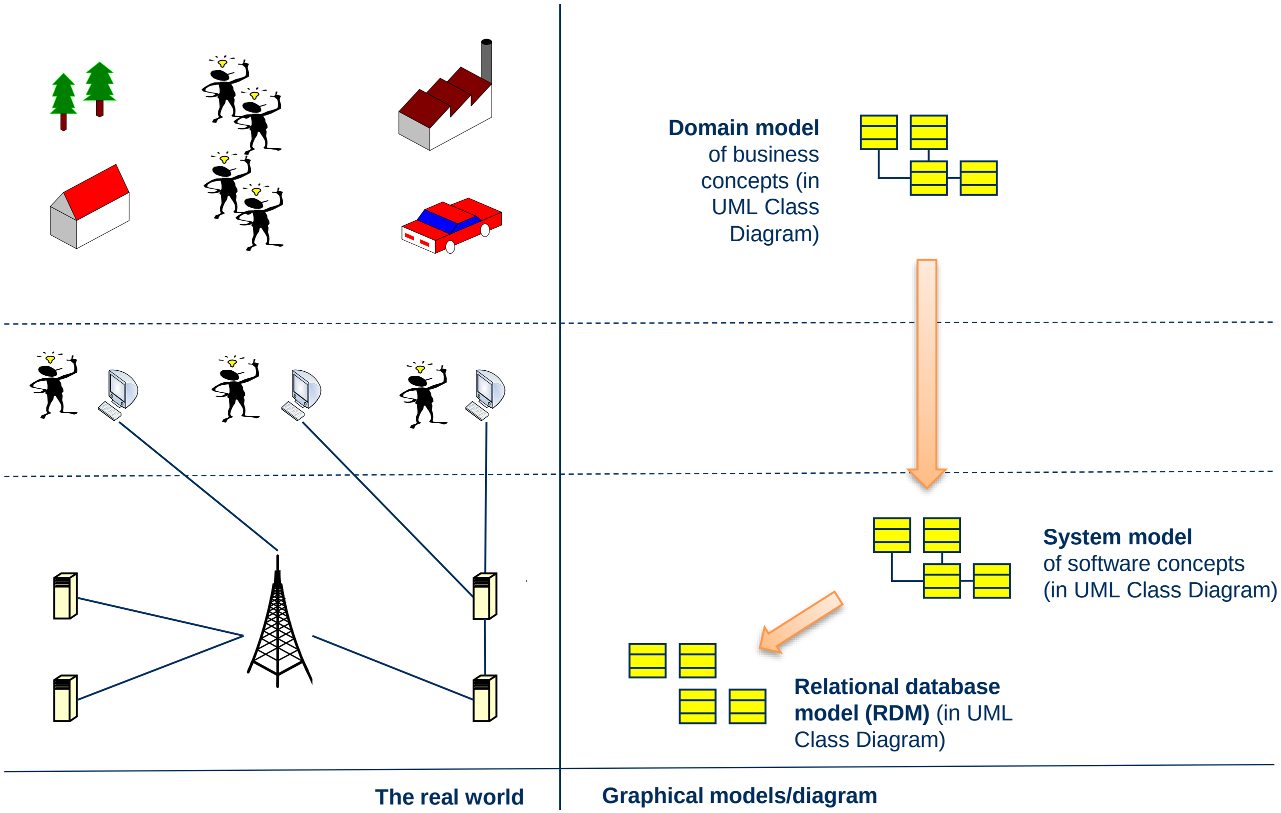
Domain model of business concepts (in UML Class Diagram)

System model of software concepts (in UML Class Diagram)

The real world

Graphical models/diagram

Real World and Models



Domain model of business concepts (in UML Class Diagram)

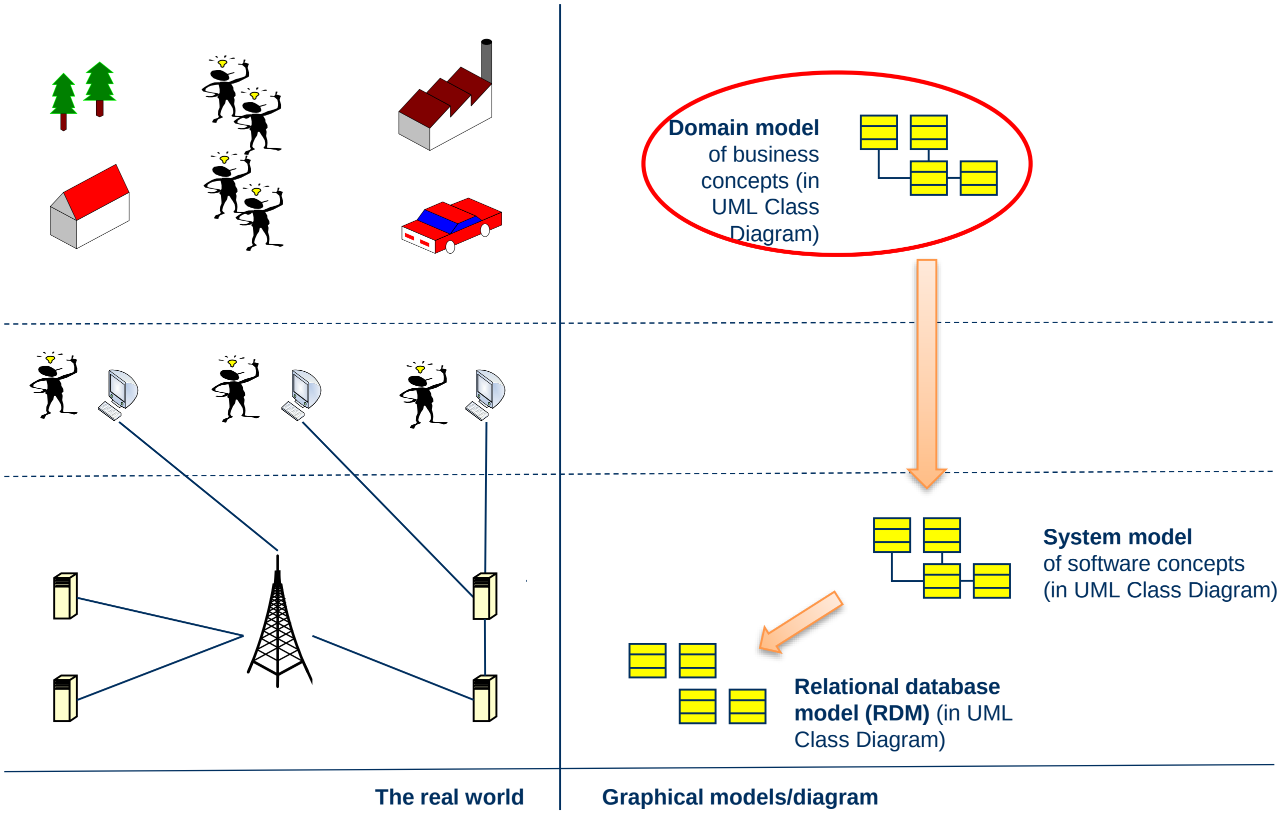
System model of software concepts (in UML Class Diagram)

Relational database model (RDM) (in UML Class Diagram)

The real world

Graphical models/diagram

Real World and Models



Domain model of business concepts (in UML Class Diagram)

System model of software concepts (in UML Class Diagram)

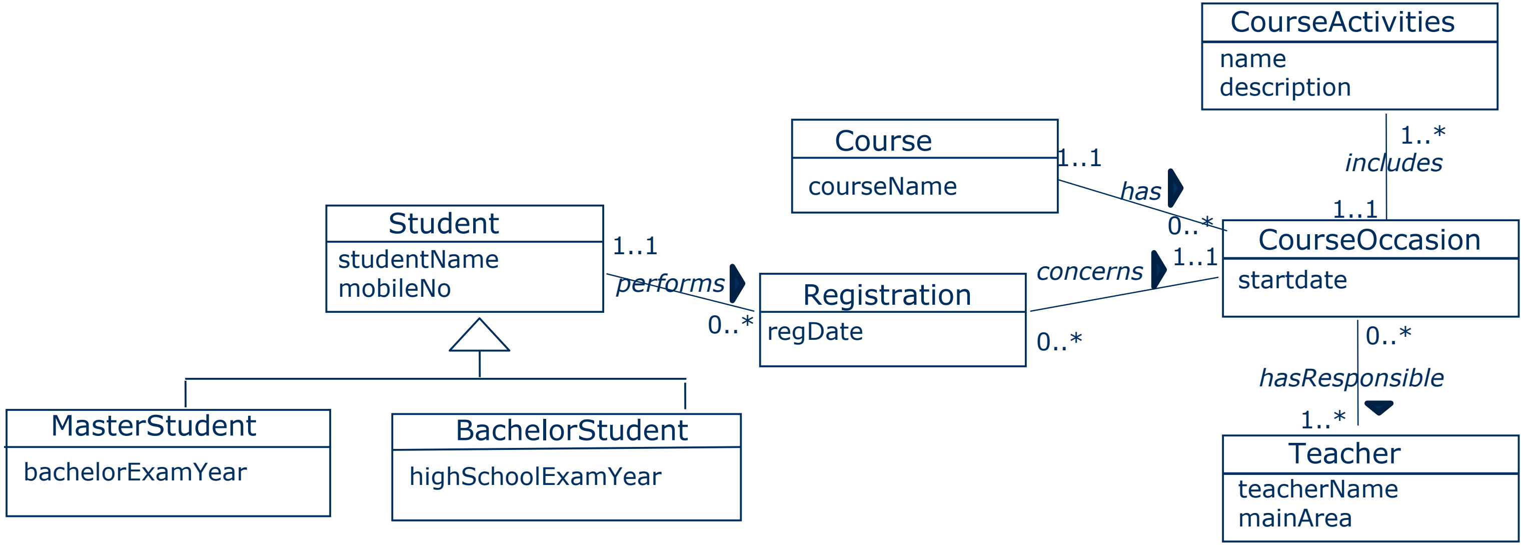
Relational database model (RDM) (in UML Class Diagram)

The real world

Graphical models/diagram

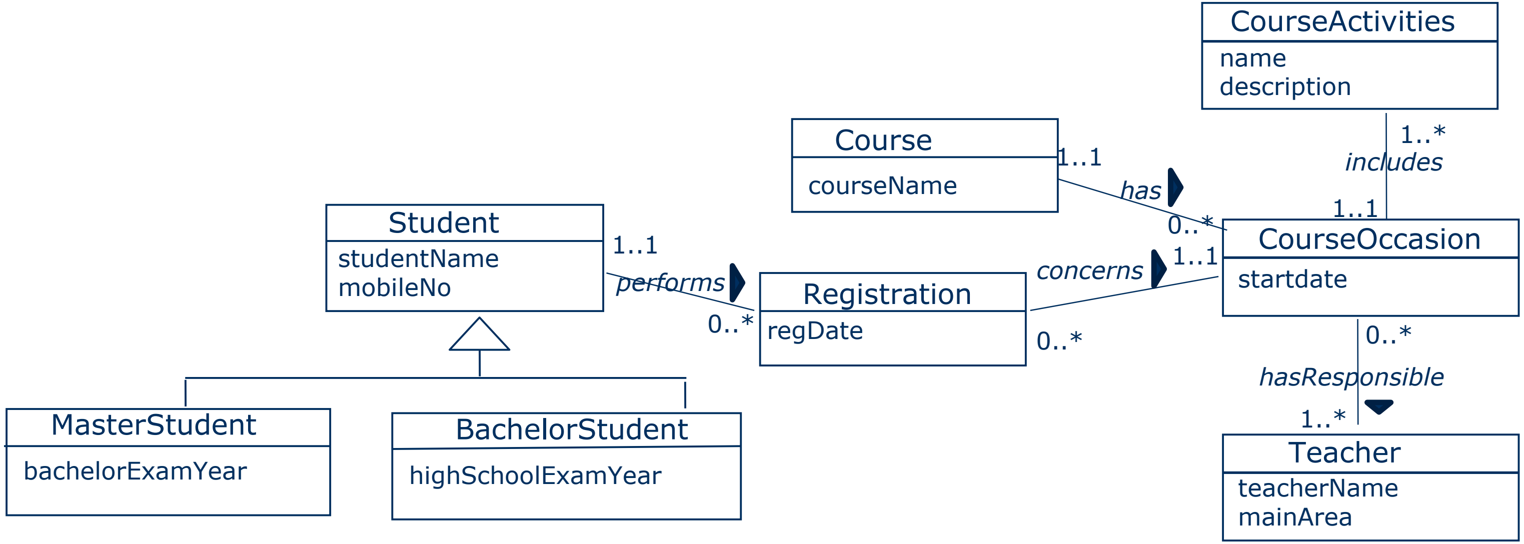
Domain model

- THE START: Concepts used in a business



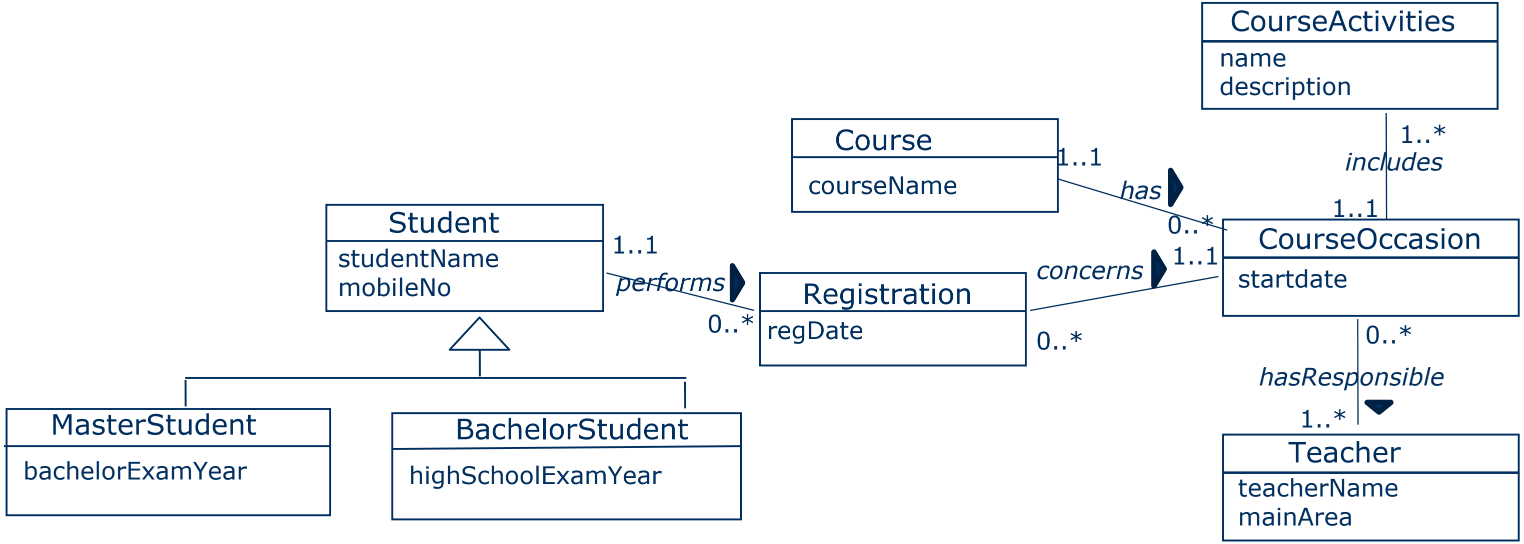
Domain model – Step 1

- TO DO: Add multiplicity for the attributes



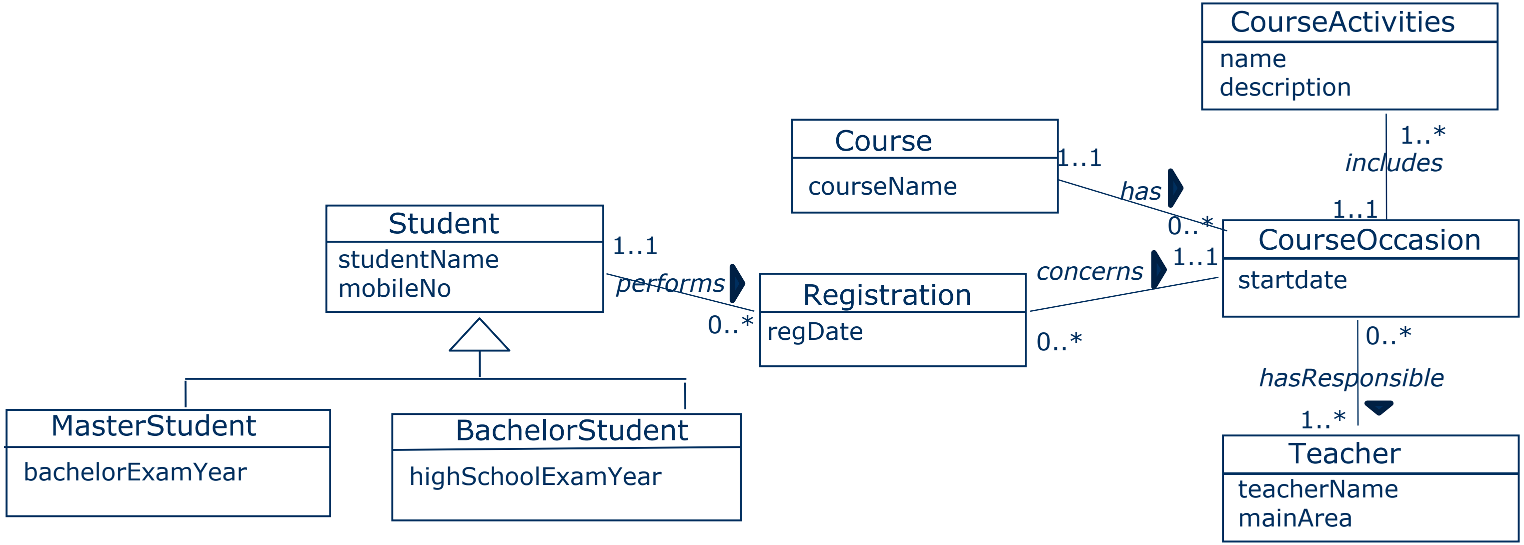
Domain model – Step 1

- TO DO: Add multiplicity for the attributes



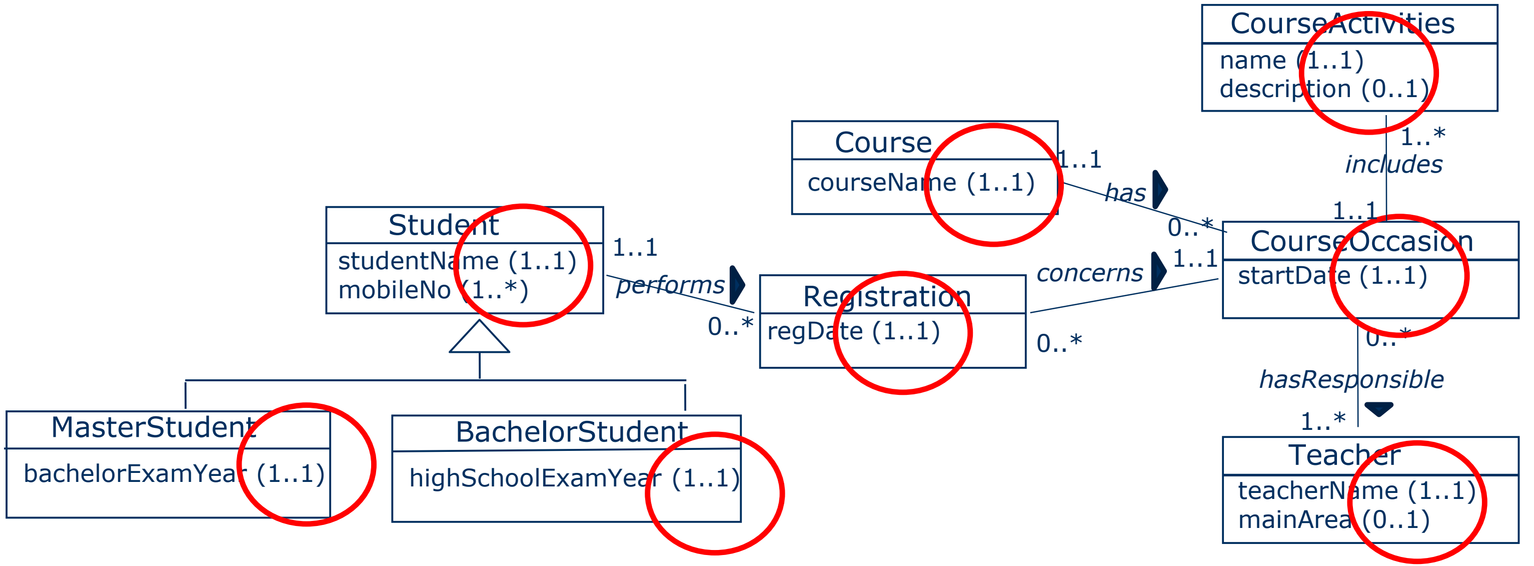
Domain model – Step 1

- TO DO: Add multiplicity for the attributes



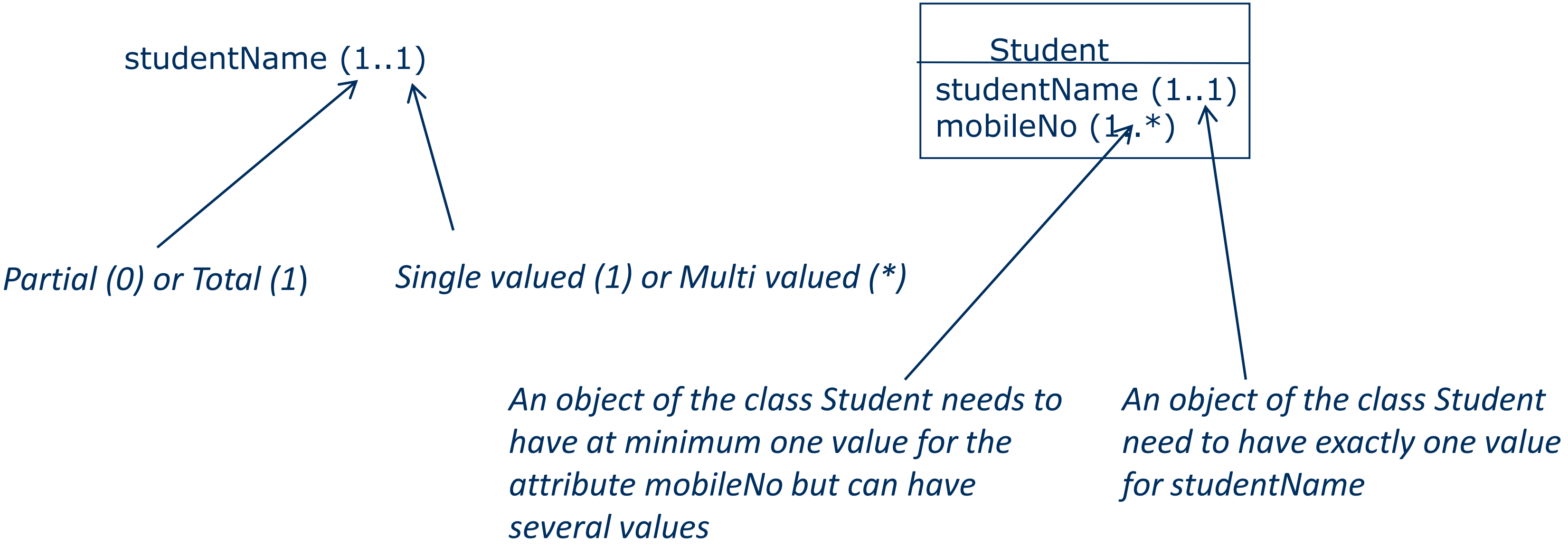
Domain model – Step 1

- DONE: Add multiplicity for the attributes



Towards a System Model- Step 1

- MORE ABOUT: Add multiplicity for the attributes



studentName (1..1)

Partial (0) or Total (1)

Single valued (1) or Multi valued (*)

Student

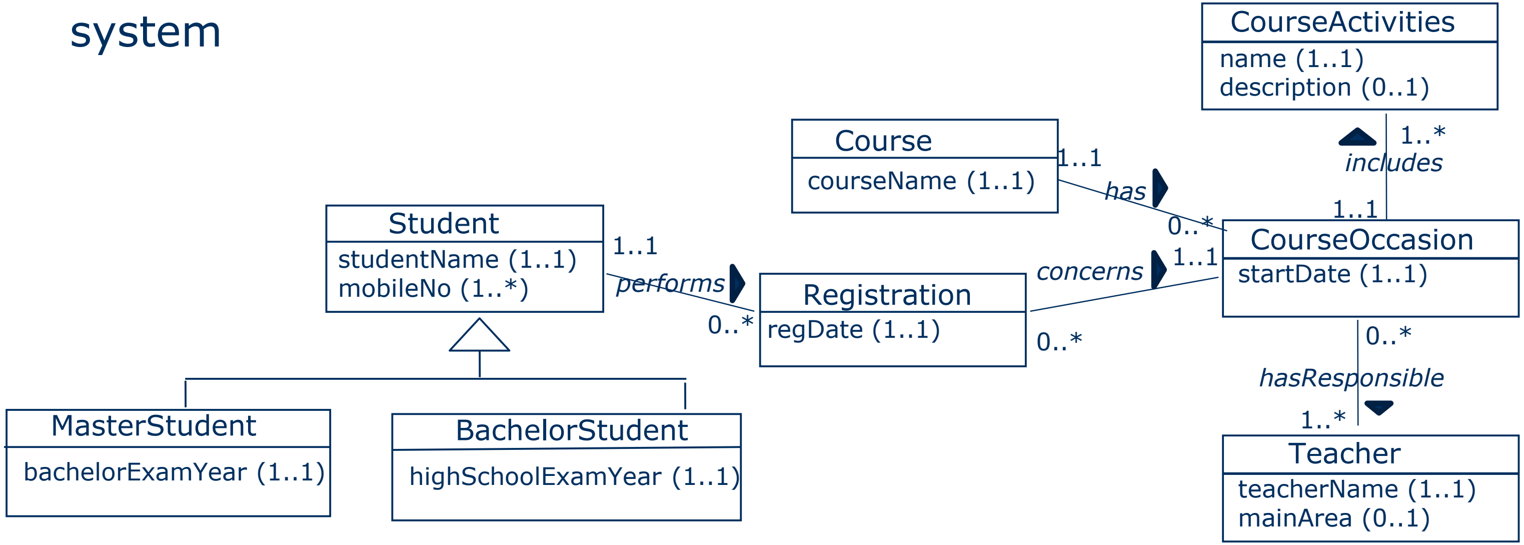
studentName (1..1)
mobileNo (1..*)

An object of the class Student needs to have at minimum one value for the attribute mobileNo but can have several values

An object of the class Student need to have exactly one value for studentName

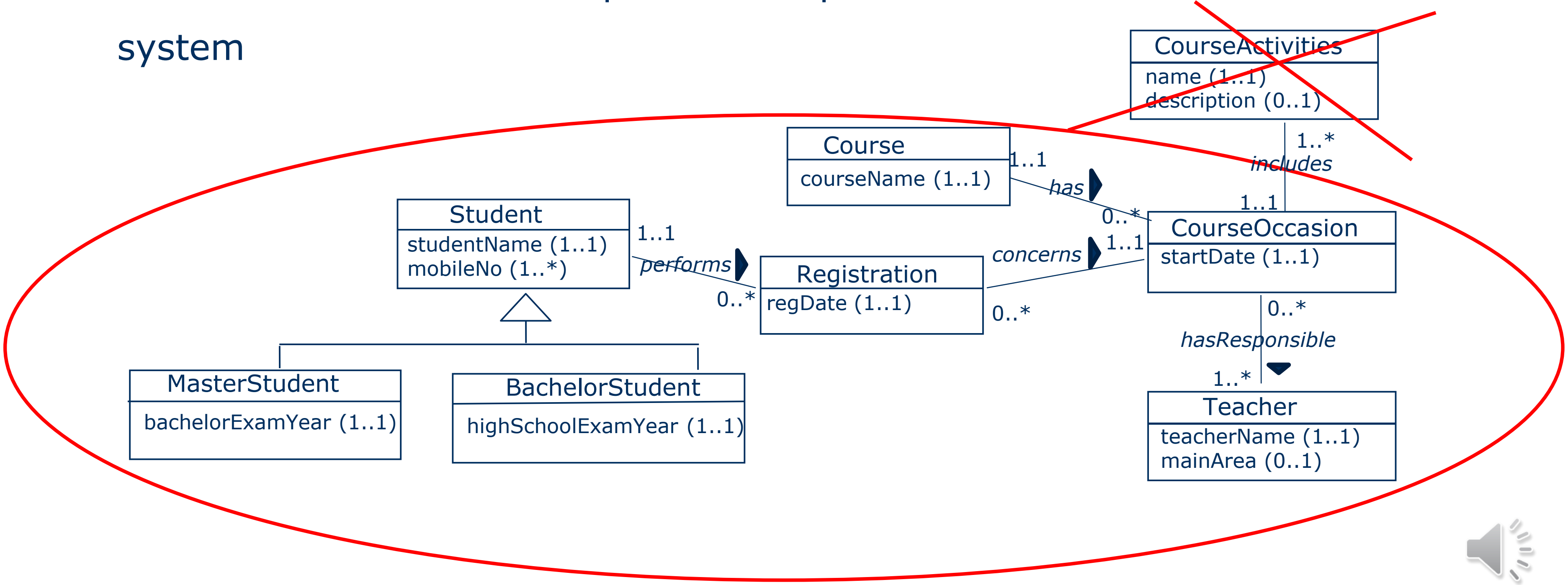
Domain model – Step 2

- TO DO: Decide which concepts to be implemented in a system

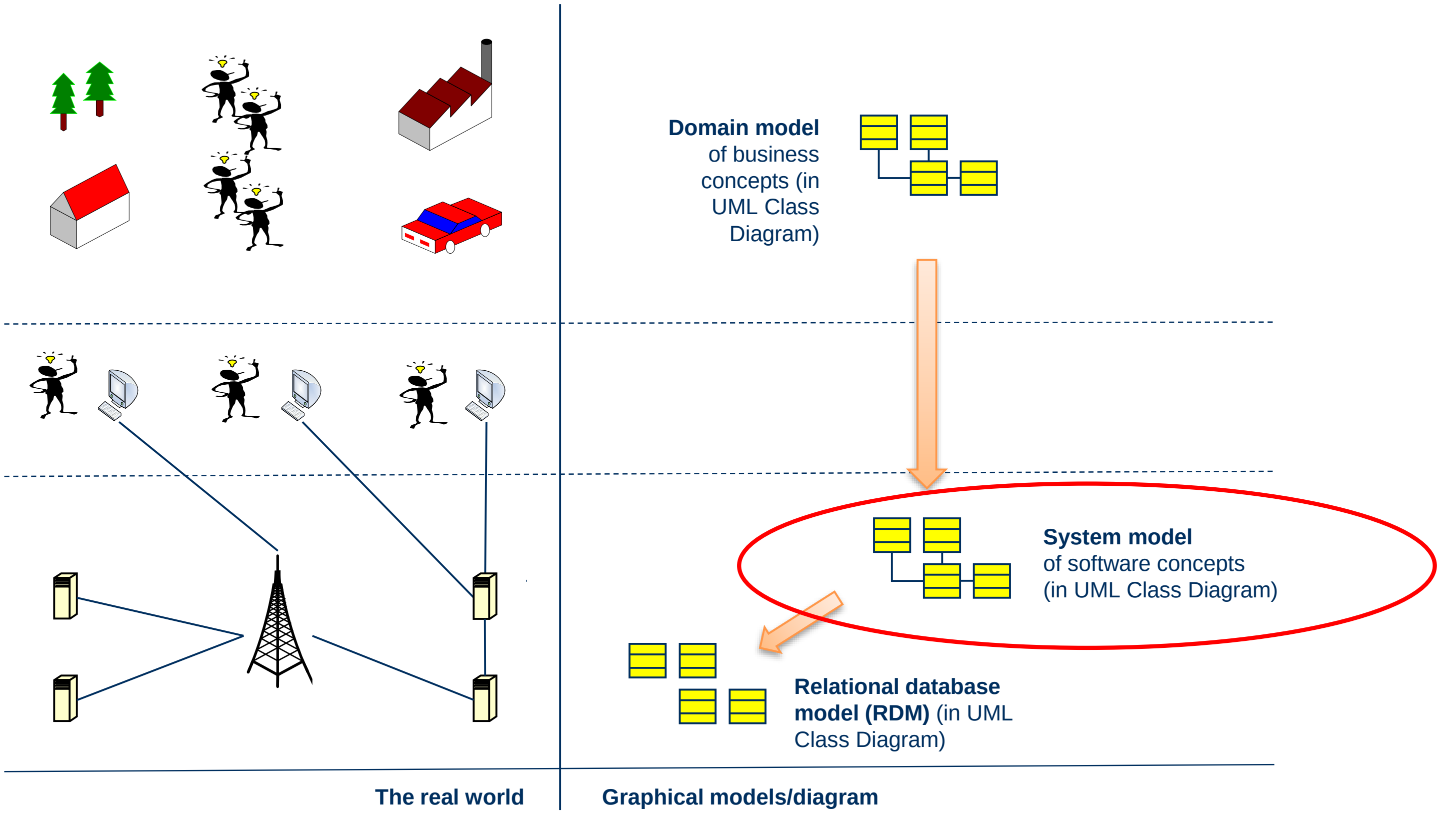


Towards a System Model- Step 2

- DONE: Decide which concepts to be implemented in a system



Real World and Models



Domain model of business concepts (in UML Class Diagram)

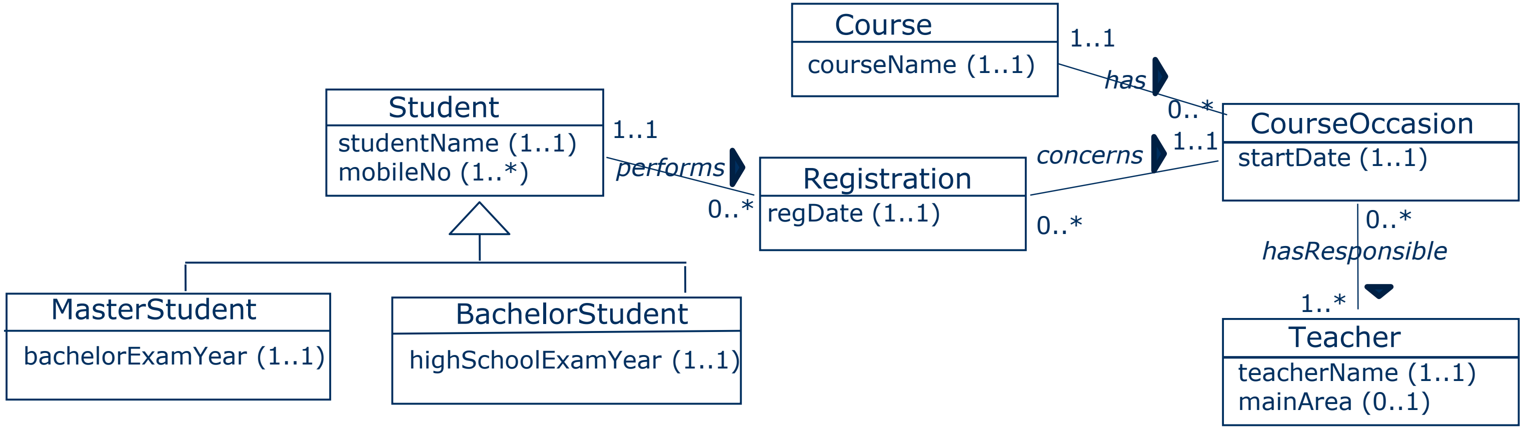
System model of software concepts (in UML Class Diagram)

Relational database model (RDM) (in UML Class Diagram)

The real world

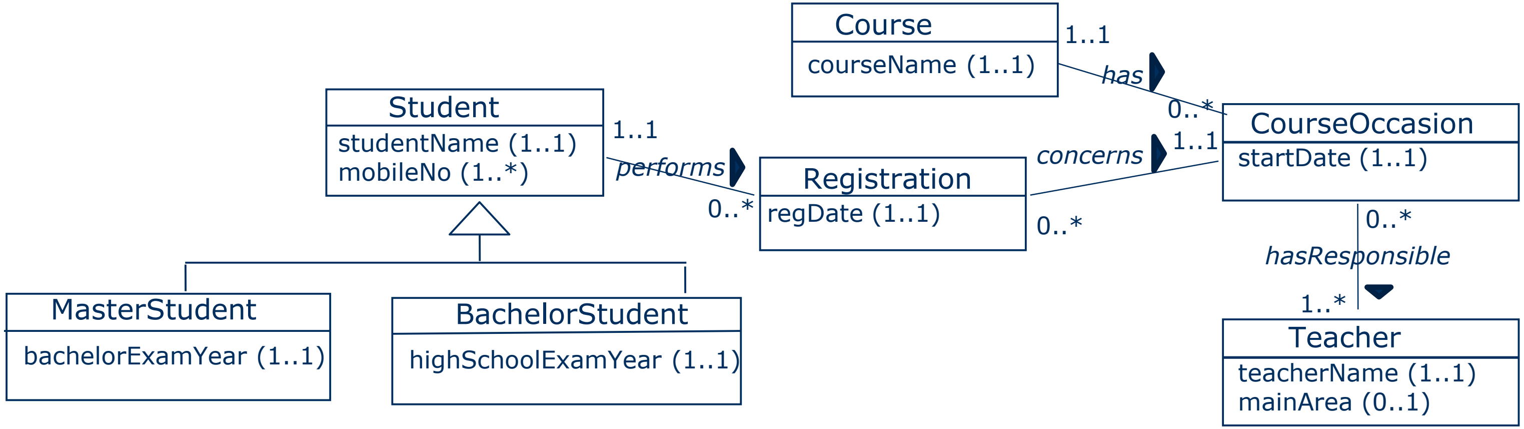
Graphical models/diagram

System Model



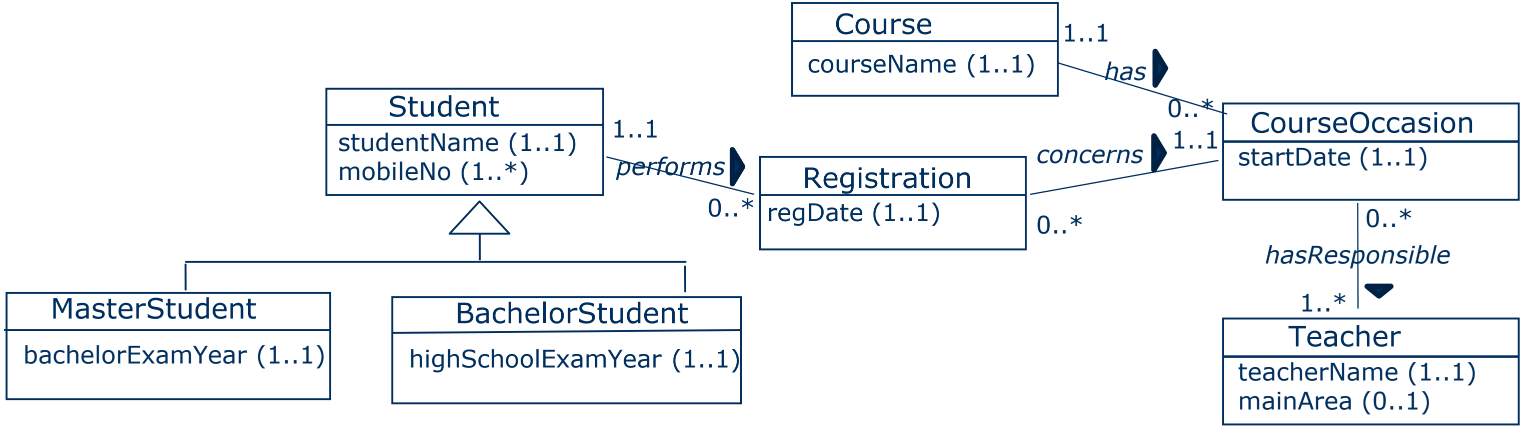
System Model

- A **System Model** is a model that contains the concepts and relationships that you want to base your IT system on – that is, the concepts and relationships that you want to implement



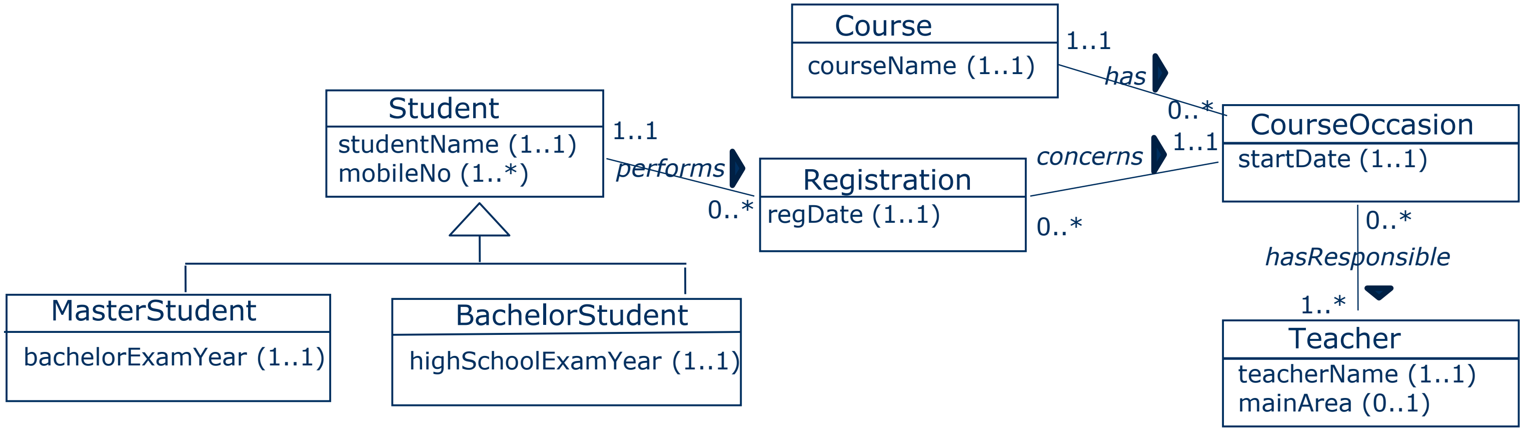
System Model

- A **System Model** is independent on technology. It can be implemented as a relational database model/schema or as an application in Java or C++



System Model

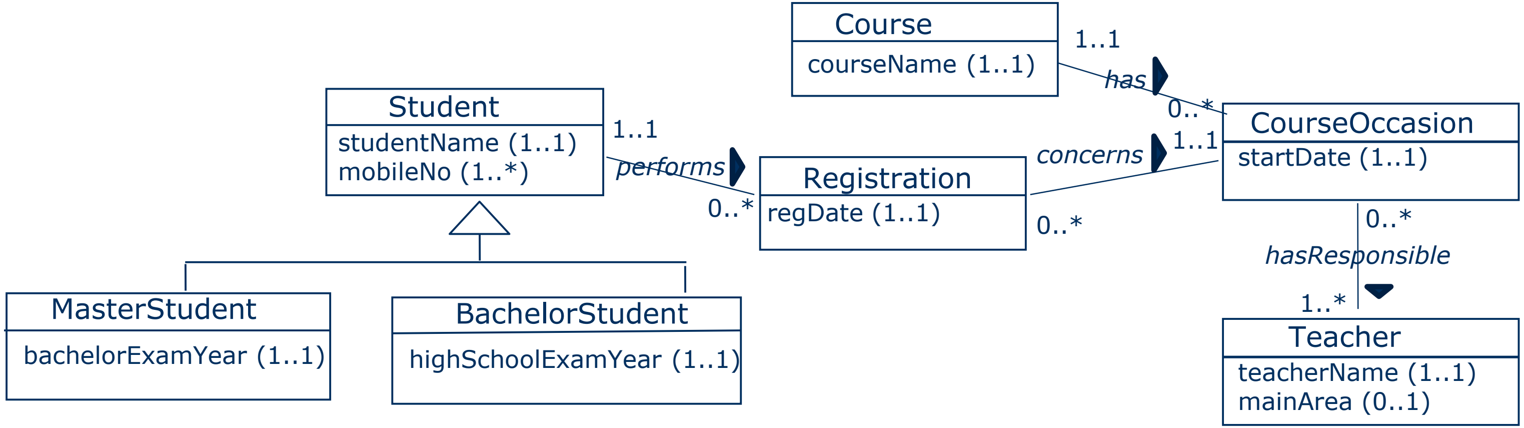
- In this presentation we will implement the **System Model** as a **Relational Database Model** – and suggest a number of steps to do this transformation



System Model

- In this presentation we will implement the **System Model** as **Relational Database Model** – and suggest a number of steps to do this transformation

Note, this can be done in several ways, and we will present one way to do it



Course
courseName (1..1)

1..1
has
0..*

CourseOccasion
startDate (1..1)

Student
studentName (1..1)
mobileNo (1..*)

1..1
performs
0..*

Registration
regDate (1..1)

concerns
1..1
0..*

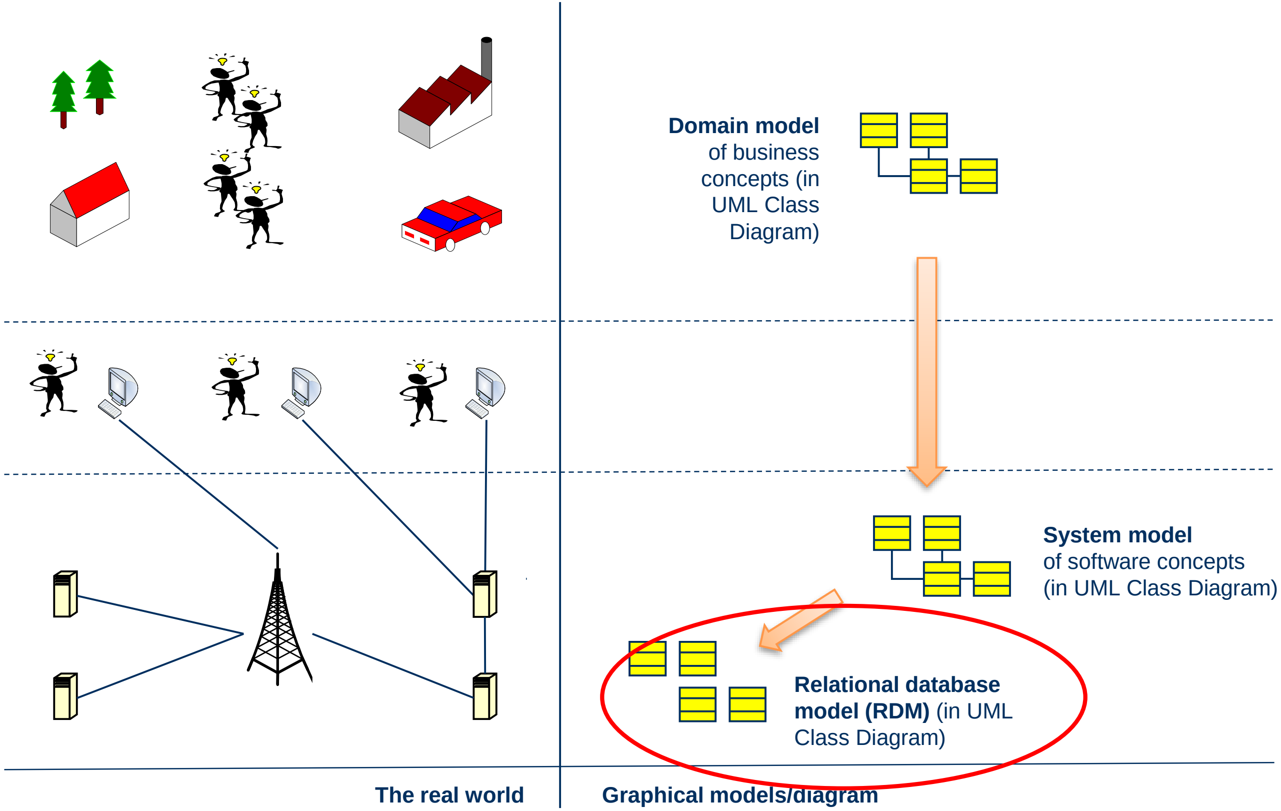
0..*
hasResponsible
1..*

Teacher
teacherName (1..1)
mainArea (0..1)

MasterStudent
bachelorExamYear (1..1)

BachelorStudent
highSchoolExamYear (1..1)

Real World and Models



Domain model of business concepts (in UML Class Diagram)

System model of software concepts (in UML Class Diagram)

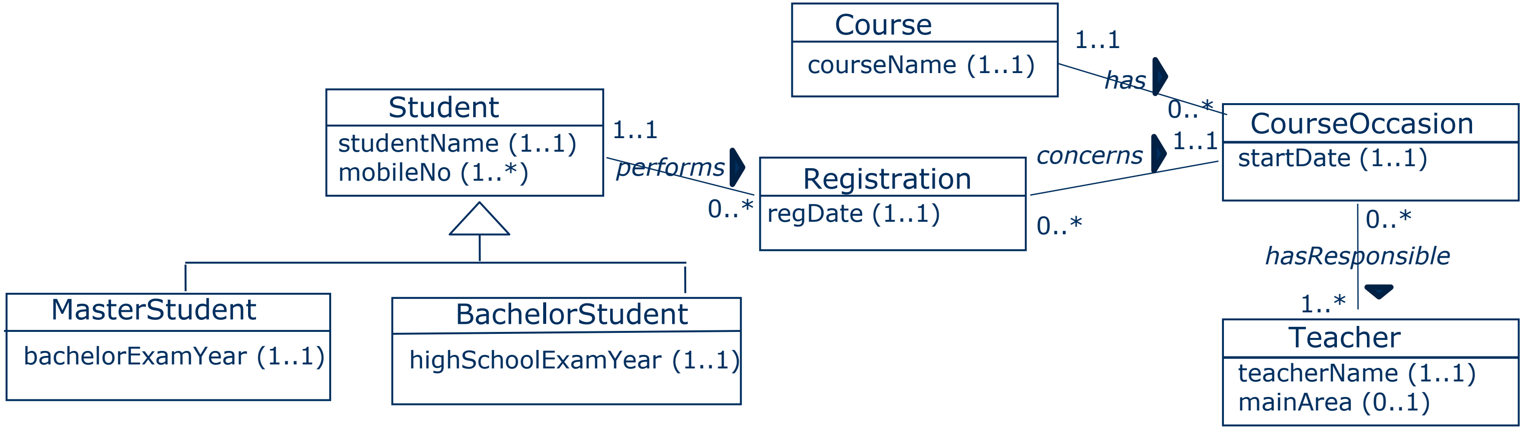
Relational database model (RDM) (in UML Class Diagram)

The real world

Graphical models/diagram

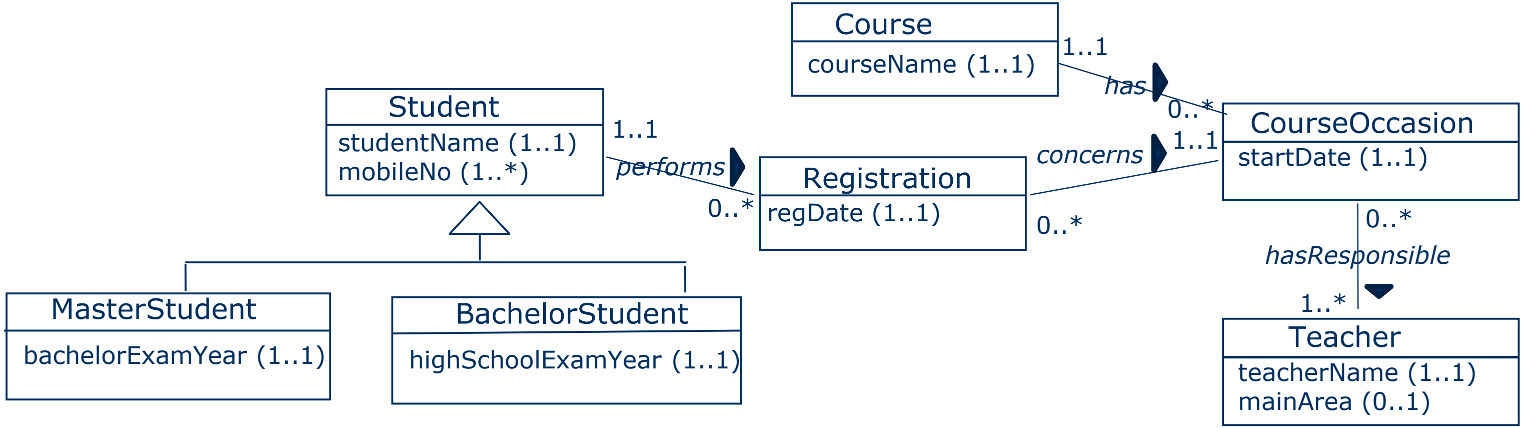
Towards a RDM

- THE START: We will transform the System Model to a Relational Database Model (RDM) in a number of steps



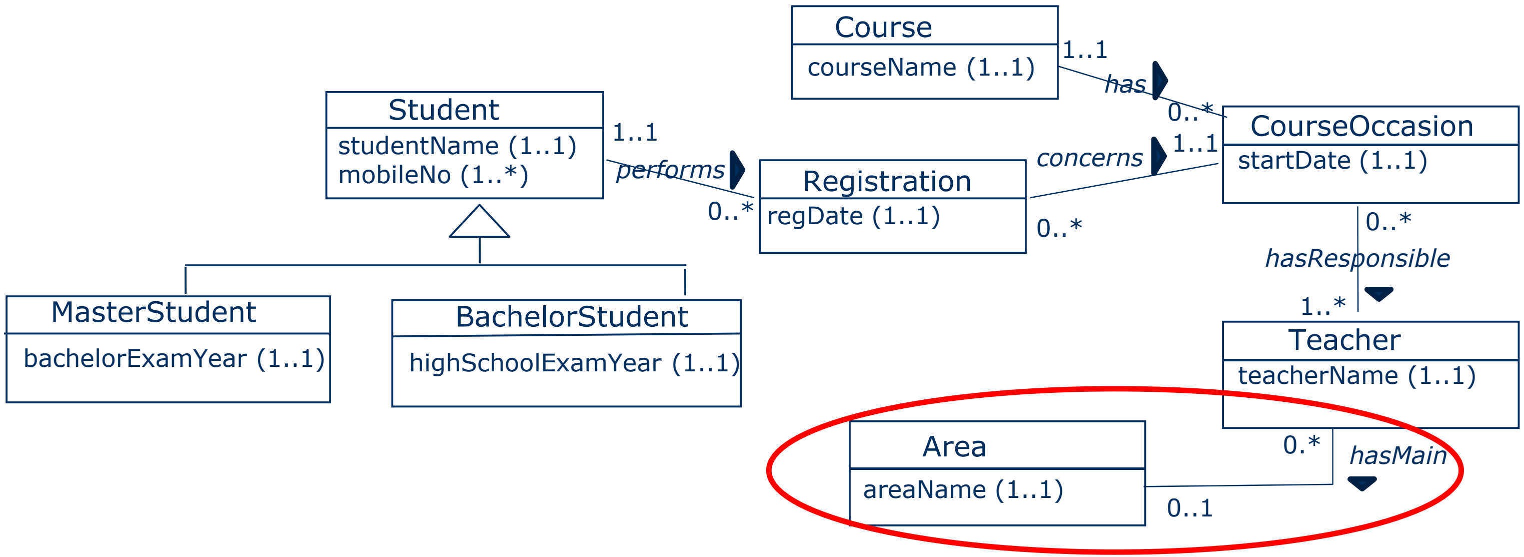
Towards a RDM - Step 1

- TO DO: Start check the multiplicity of the attributes. Create new classes if some of the attributes have the multiplicity 0..1 or 0..*. Why? In order to not accept NULL



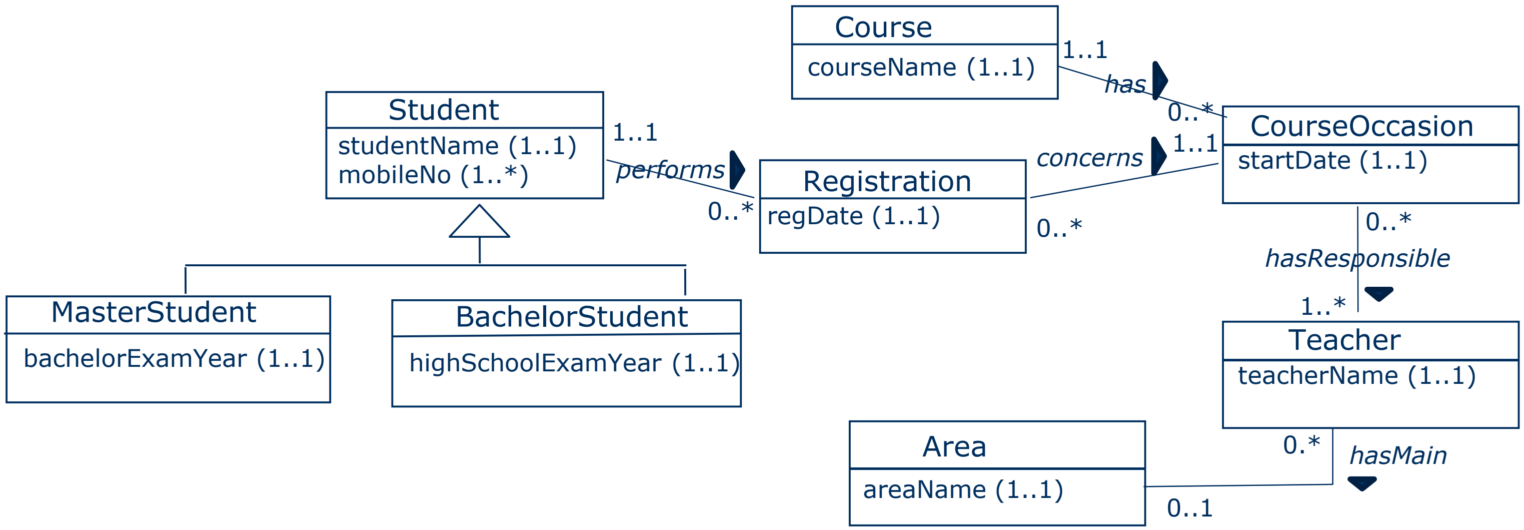
Towards a RDM - Step 1

- DONE: Start check the multiplicity of the attributes. Create new classes if some of the attributes have the multiplicity 0..1 or 0..*. Why? In order to not accept NULL



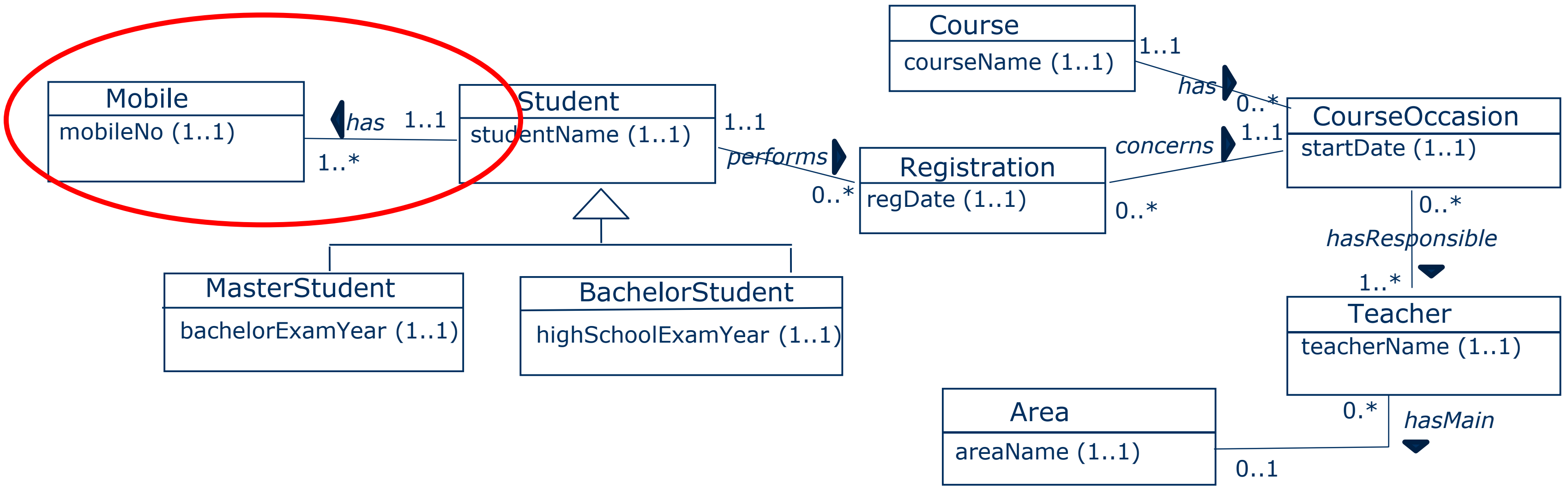
Towards a RDM - Step 2

- TO DO: Create new classes if some of the attribute have the multiplicity 1..* or 0..*. Why? In order to not accept multi values (see 1NF)



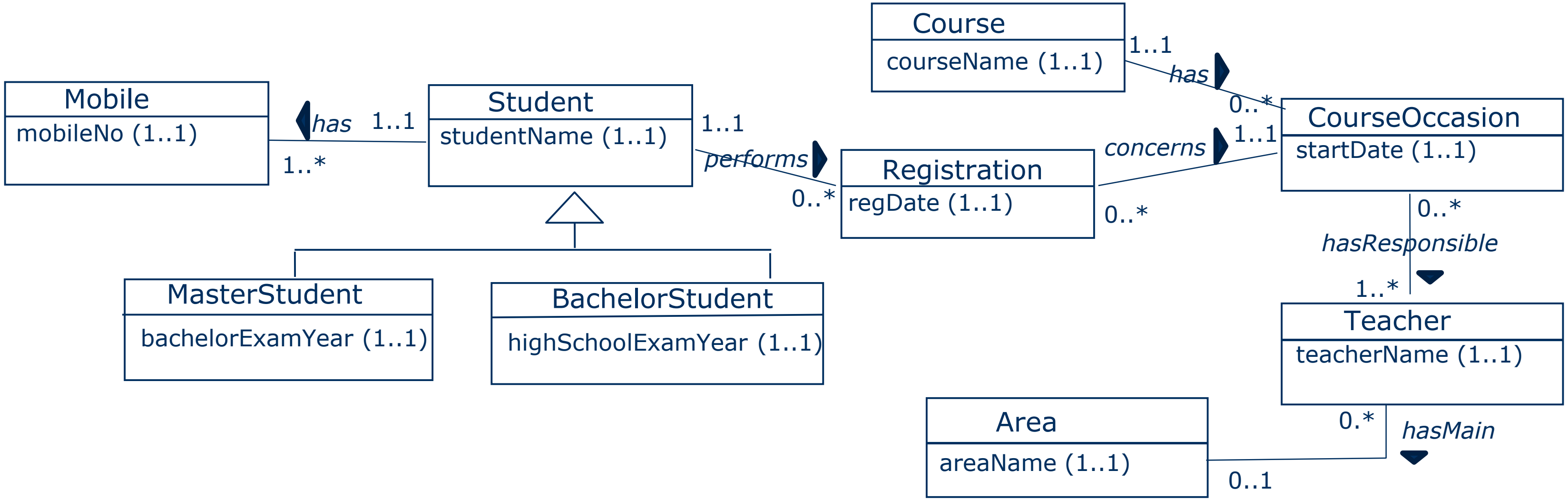
Towards a RDM - Step 2

- DONE: Create new classes if some of the attribute have the multiplicity 1..* or 0..*. Why: In order to not accept multi values (see 1NF)



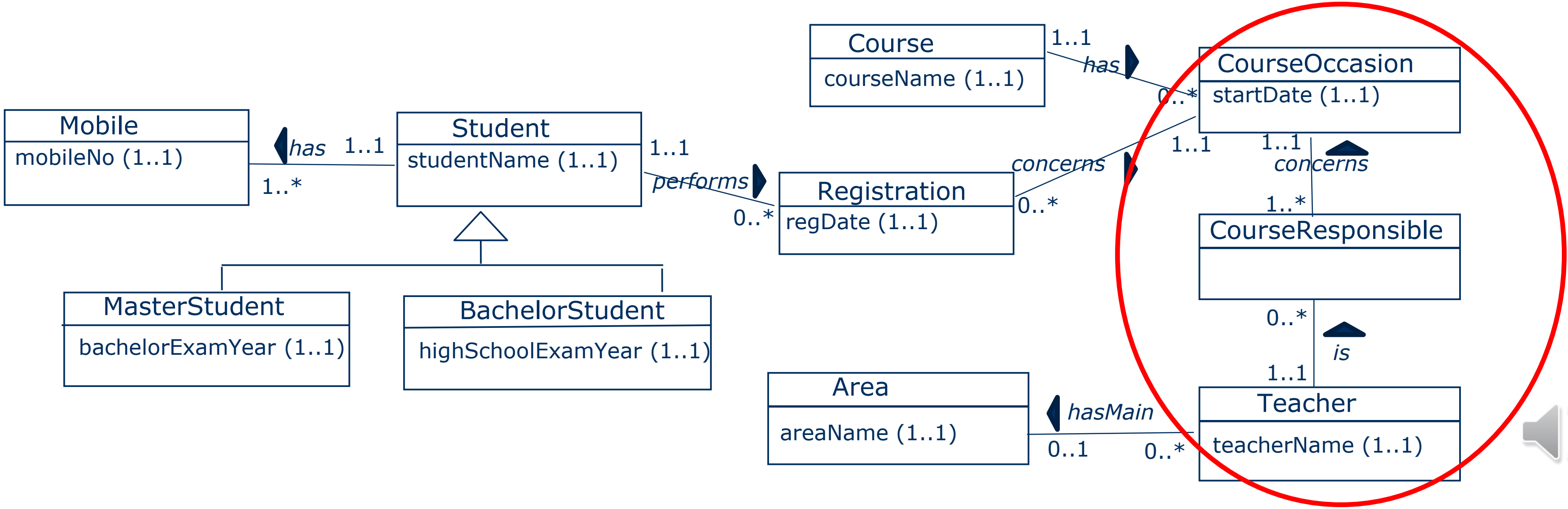
Towards a RDM - Step 3

- TO DO: Create new classes if some associations have the multiplicity 1..*/0..* or 0..*/0..* on both sides. Why? The relational database technology requires that



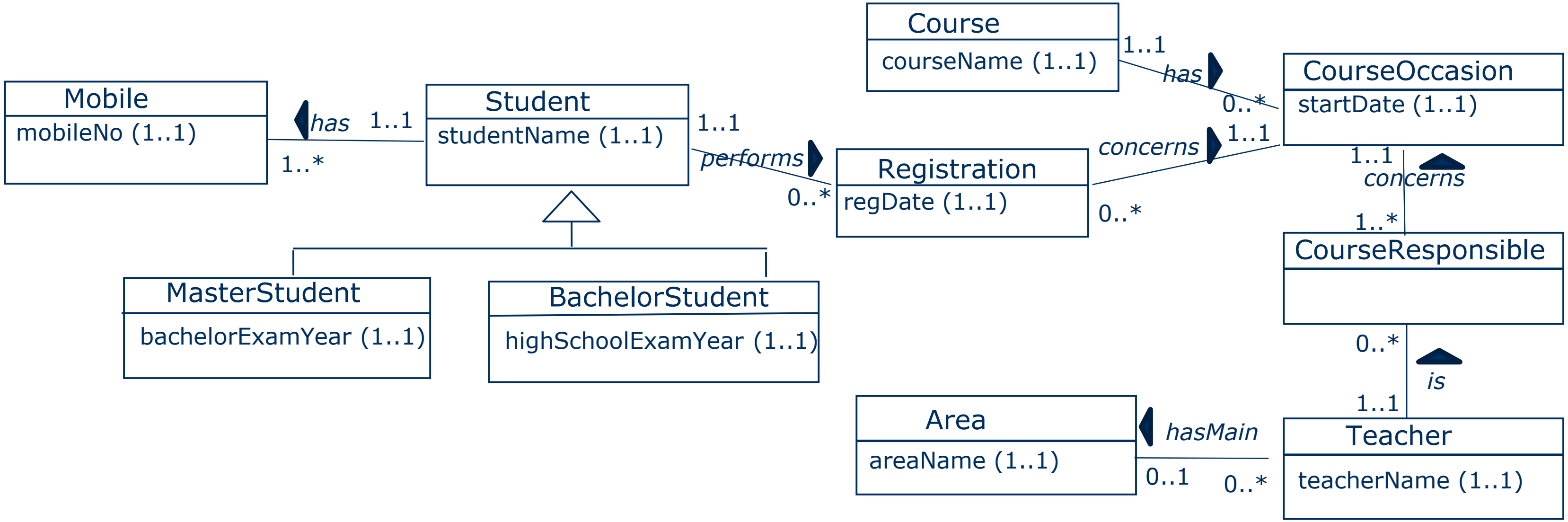
Towards a RDM - Step 3

- DONE: Create new classes if some associations have the multiplicity 1..*/0..* or 0..*/0..* on both sides. Why? The relational database technology requires that



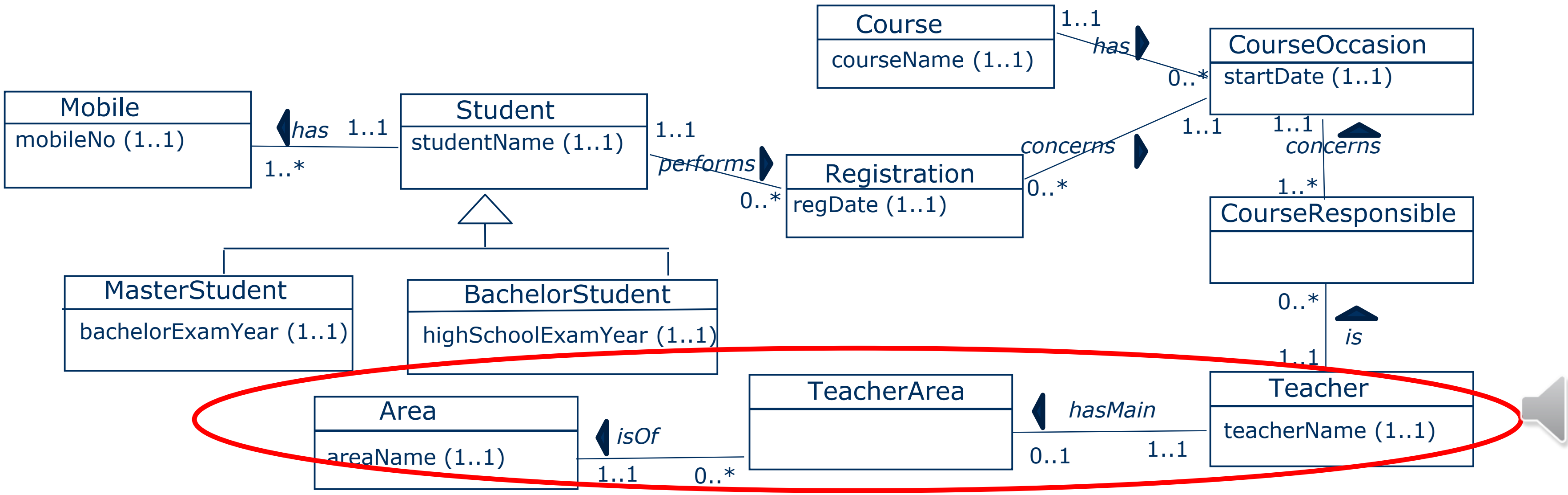
Towards a RDM - Step 4

- TO DO: Create new classes if some associations still have the multiplicity that start with 0 on both sides, for example 0..1 and 0..*. Why? To avoid NULL



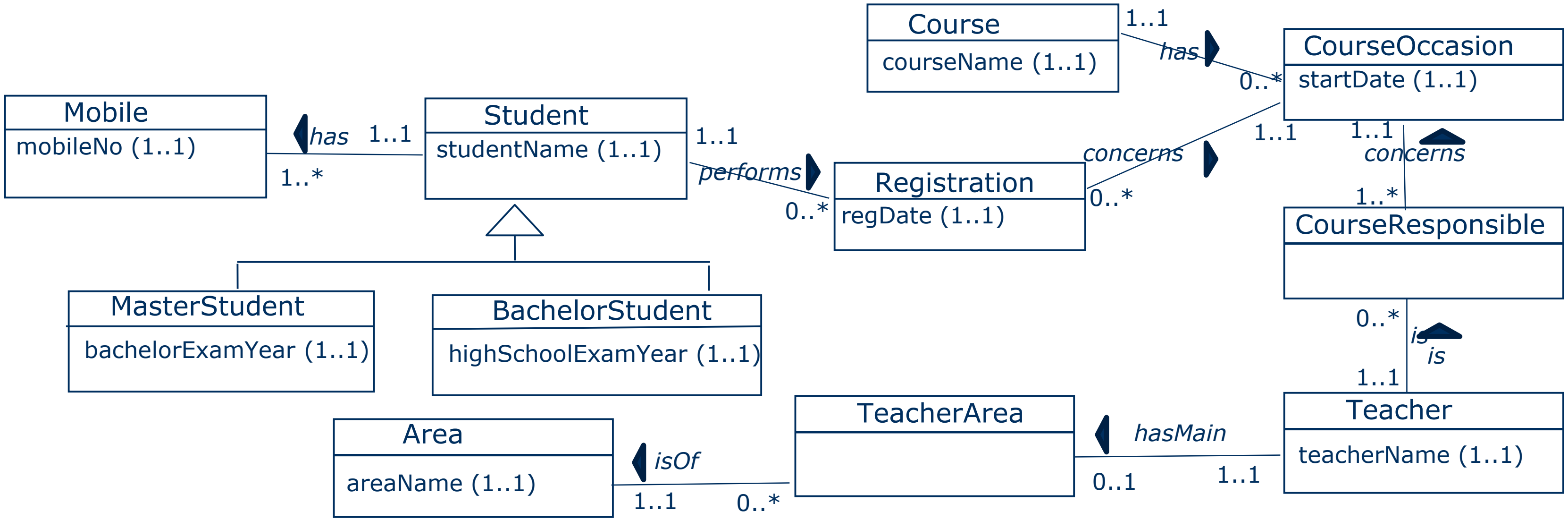
Towards a RDM - Step 4

- TO DO: Create new classes if some associations have the multiplicity that start with 0 on both sides, for example 0..1 and 0..*. Why? To avoid NULL



Towards a RDM - Step 5

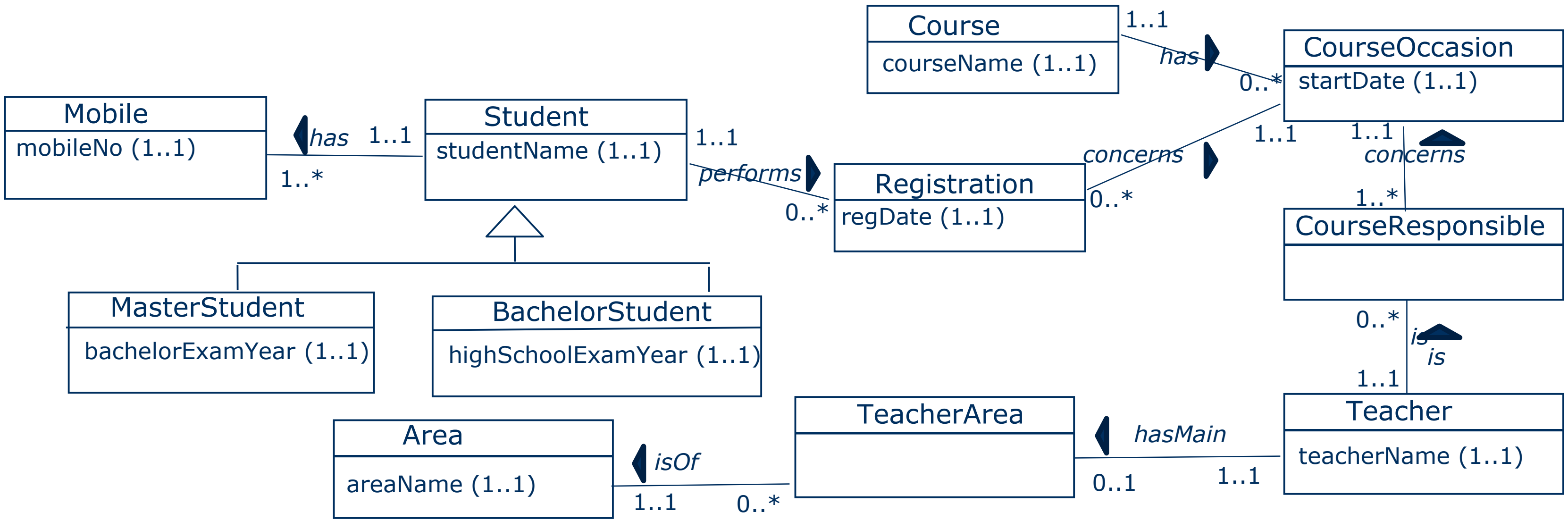
- TO DO: Decide identifier in each class – that is, decide primary key (PK). Why? The relational database technology requires PK



Towards a RDM – Step 5

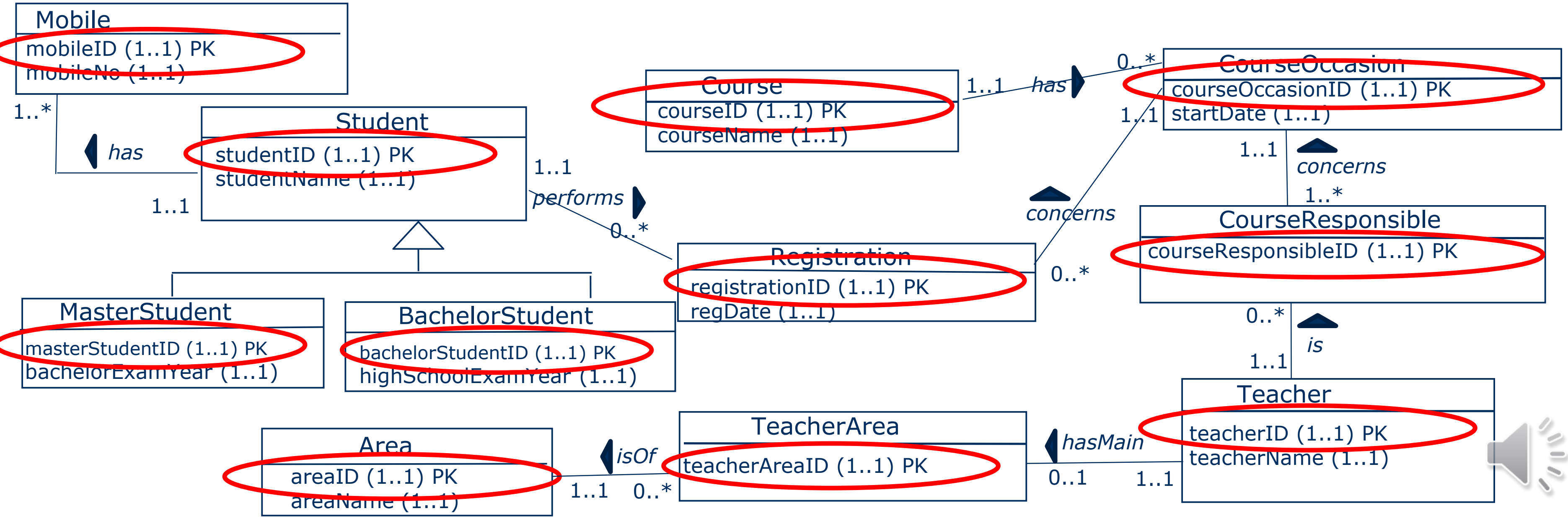
- TO DO: Decide identifier in each class – that is, decide primary key (PK). How? By adding a surrogate key (SK) as the PK in each class as *a first of two steps* to replace associations

Note, this can be done in several ways, and we will present one way to do it



Towards a RDM - Step 5

- DONE: Decide identifier in each class – that is, decide primary key (PK) – by adding a surrogate key (SK) as the PK in each class as a *first of two steps* to replace associations



Mobile
mobileID (1..1) PK
mobileNo (1..1)

Student
studentID (1..1) PK
studentName (1..1)

MasterStudent
masterStudentID (1..1) PK
bachelorExamYear (1..1)

BachelorStudent
bachelorStudentID (1..1) PK
highSchoolExamYear (1..1)

Course
courseID (1..1) PK
courseName (1..1)

CourseOccasion
courseOccasionID (1..1) PK
startDate (1..1)

Registration
registrationID (1..1) PK
regDate (1..1)

CourseResponsible
courseResponsibleID (1..1) PK

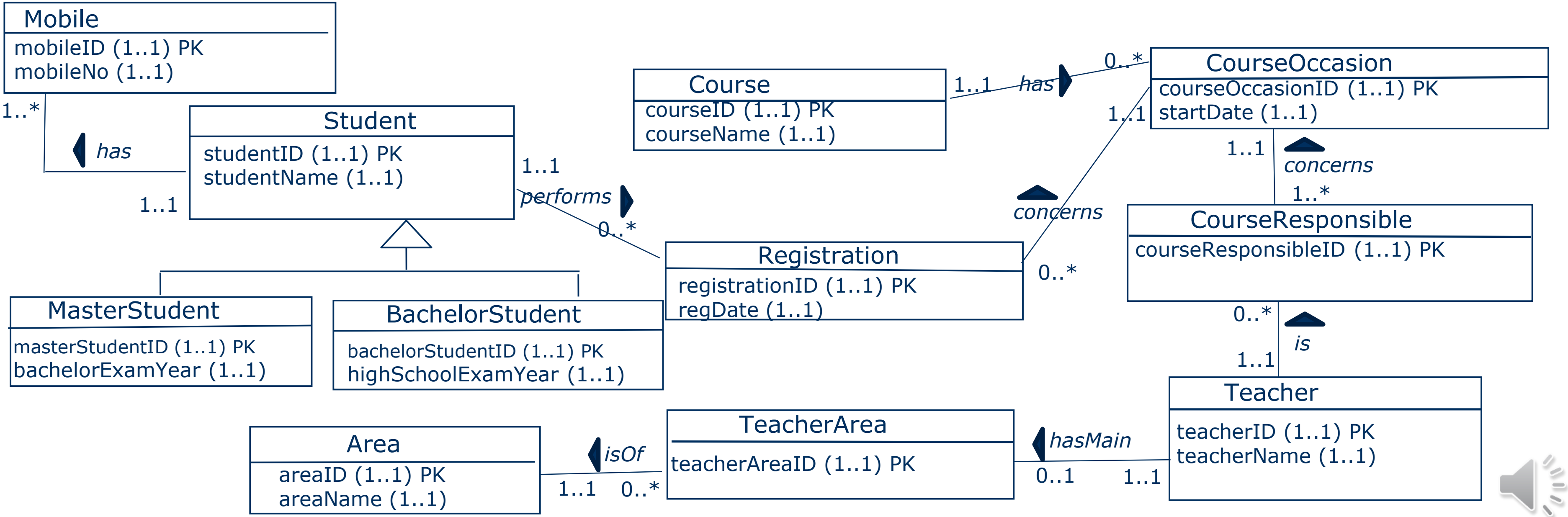
Teacher
teacherID (1..1) PK
teacherName (1..1)

TeacherArea
teacherAreaID (1..1) PK

Area
areaID (1..1) PK
areaName (1..1)

Towards a RDM - Step 6

- TO DO: Add foreign keys (FK) to match the PK as the *second of two steps* to replace and represent associations. Why? The relational database technology requires FK



Mobile
mobileID (1..1) PK
mobileNo (1..1)

Student
studentID (1..1) PK
studentName (1..1)

MasterStudent
masterStudentID (1..1) PK
bachelorExamYear (1..1)

BachelorStudent
bachelorStudentID (1..1) PK
highSchoolExamYear (1..1)

Course
courseID (1..1) PK
courseName (1..1)

CourseOccasion
courseOccasionID (1..1) PK
startDate (1..1)

Registration
registrationID (1..1) PK
regDate (1..1)

CourseResponsible
courseResponsibleID (1..1) PK

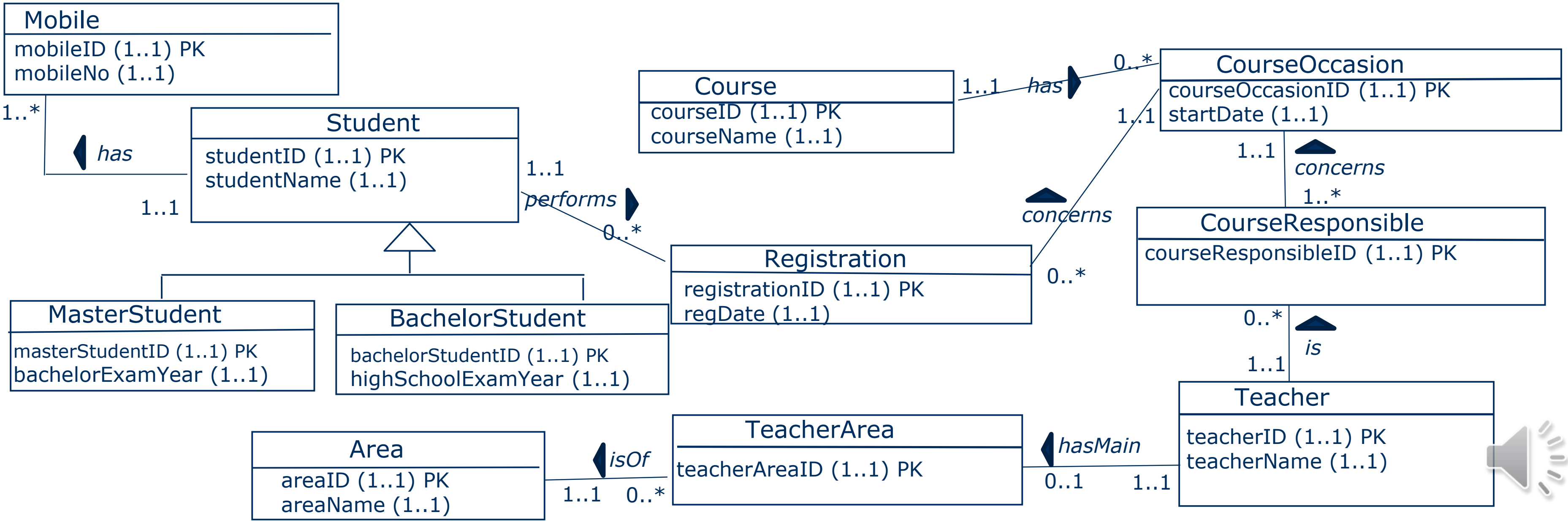
Teacher
teacherID (1..1) PK
teacherName (1..1)

TeacherArea
teacherAreaID (1..1) PK

Area
areaID (1..1) PK
areaName (1..1)

Towards a RDM - Step 6

- TO DO: Add foreign keys (FK) to match the PK, as the second of two steps to replace and represent associations – **on the side of the association that has 0..* or 1..***



Mobile
mobileID (1..1) PK
mobileNo (1..1)

Student
studentID (1..1) PK
studentName (1..1)

MasterStudent
masterStudentID (1..1) PK
bachelorExamYear (1..1)

BachelorStudent
bachelorStudentID (1..1) PK
highSchoolExamYear (1..1)

Course
courseID (1..1) PK
courseName (1..1)

CourseOccasion
courseOccasionID (1..1) PK
startDate (1..1)

Registration
registrationID (1..1) PK
regDate (1..1)

CourseResponsible
courseResponsibleID (1..1) PK

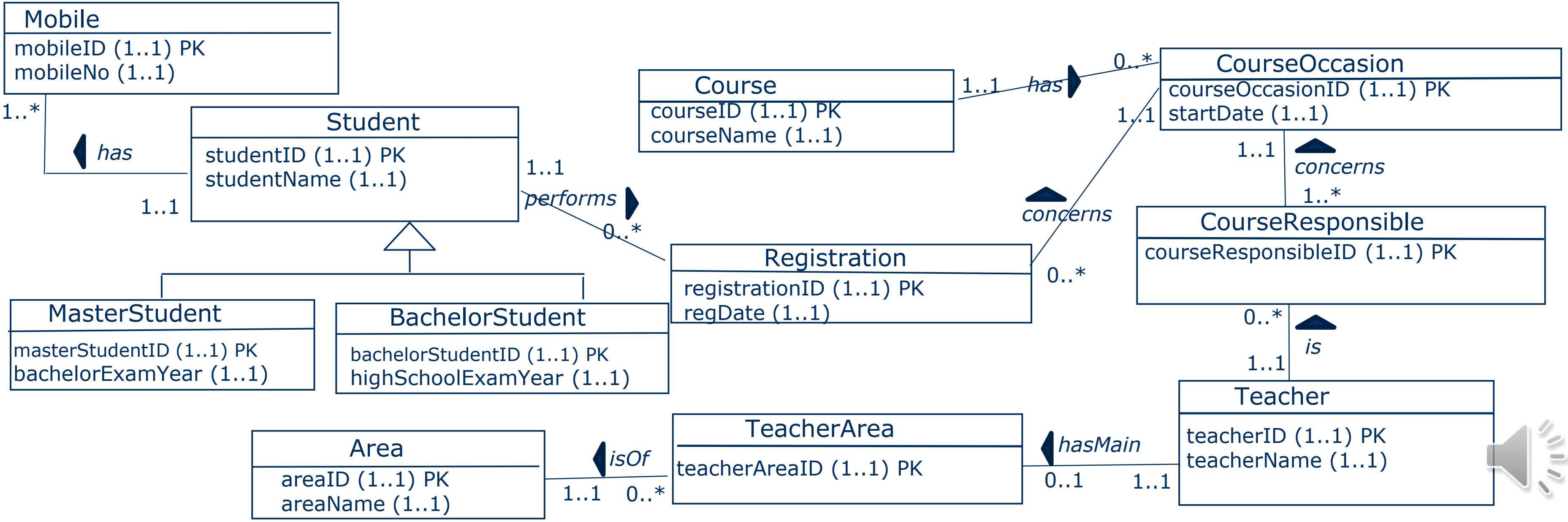
Teacher
teacherID (1..1) PK
teacherName (1..1)

TeacherArea
teacherAreaID (1..1) PK

Area
areaID (1..1) PK
areaName (1..1)

Towards a RDM - Step 6

- TO DO: Add foreign keys (FK) to match the PK, as the second of two steps to replace and represent associations – **on the side of the association that has 0..1 if the other side is 1..1**



Mobile
mobileID (1..1) PK
mobileNo (1..1)

Student
studentID (1..1) PK
studentName (1..1)

MasterStudent
masterStudentID (1..1) PK
bachelorExamYear (1..1)

BachelorStudent
bachelorStudentID (1..1) PK
highSchoolExamYear (1..1)

Course
courseID (1..1) PK
courseName (1..1)

CourseOccasion
courseOccasionID (1..1) PK
startDate (1..1)

Registration
registrationID (1..1) PK
regDate (1..1)

CourseResponsible
courseResponsibleID (1..1) PK

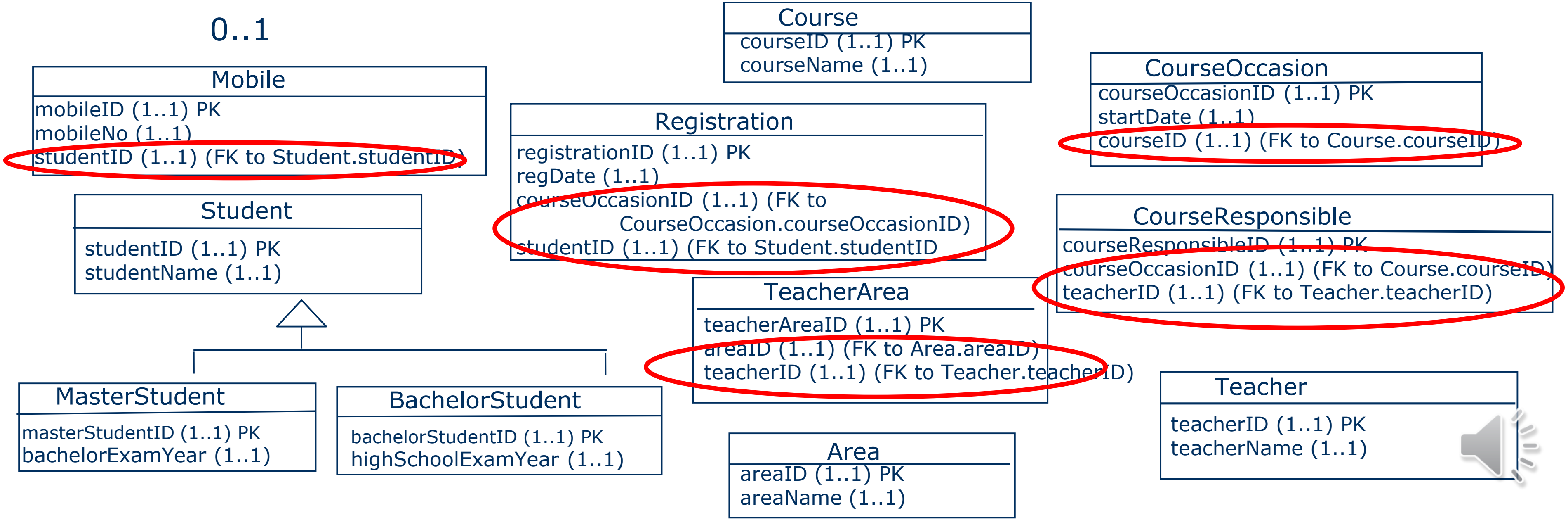
Teacher
teacherID (1..1) PK
teacherName (1..1)

TeacherArea
teacherAreaID (1..1) PK

Area
areaID (1..1) PK
areaName (1..1)

Towards a RDM - Step 6

- DONE: Add foreign keys (FK) to match the PK, as the second of two steps to replace and represent associations – on the side of the association that has 0..* or 1..* or 0..1



Course
courseID (1..1) PK
courseName (1..1)

CourseOccasion
courseOccasionID (1..1) PK
startDate (1..1)
courseID (1..1) (FK to Course.courseID)

Mobile
mobileID (1..1) PK
mobileNo (1..1)
studentID (1..1) (FK to Student.studentID)

Registration
registrationID (1..1) PK
regDate (1..1)
courseOccasionID (1..1) (FK to CourseOccasion.courseOccasionID)
studentID (1..1) (FK to Student.studentID

Student
studentID (1..1) PK
studentName (1..1)

CourseResponsible
courseResponsibleID (1..1) PK
courseOccasionID (1..1) (FK to Course.courseID)
teacherID (1..1) (FK to Teacher.teacherID)

TeacherArea
teacherAreaID (1..1) PK
areaID (1..1) (FK to Area.areaID)
teacherID (1..1) (FK to Teacher.teacherID)

MasterStudent
masterStudentID (1..1) PK
bachelorExamYear (1..1)

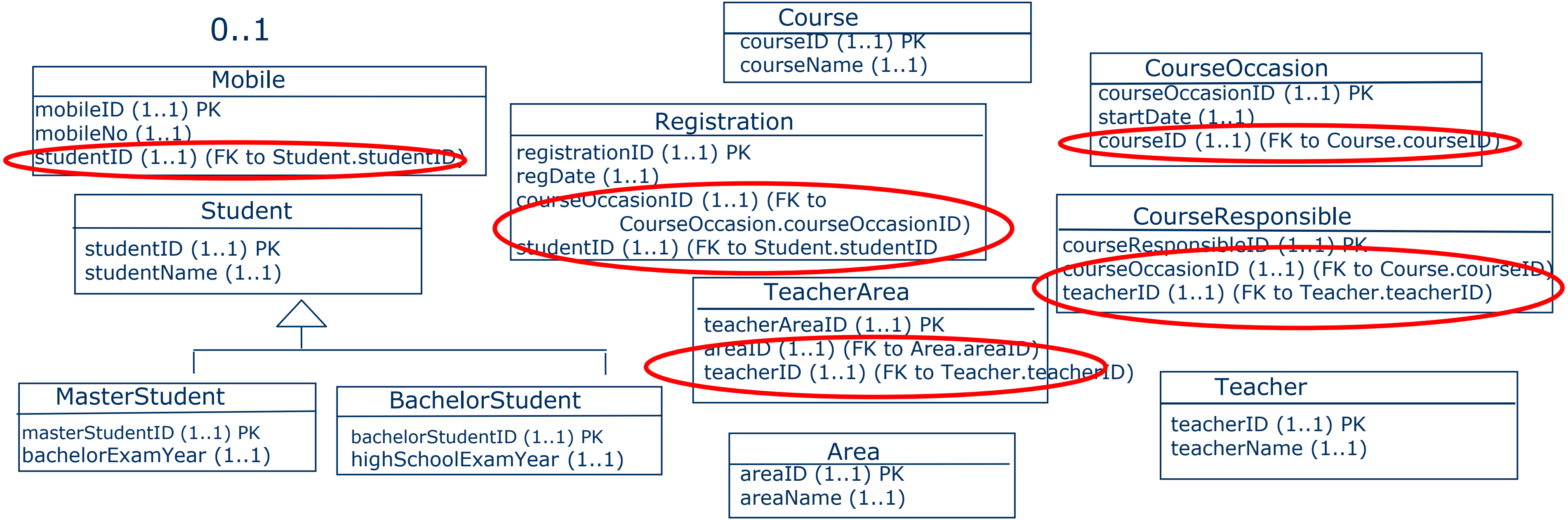
BachelorStudent
bachelorStudentID (1..1) PK
highSchoolExamYear (1..1)

Teacher
teacherID (1..1) PK
teacherName (1..1)

Area
areaID (1..1) PK
areaName (1..1)

Towards a RDM - Step 6

- DONE: Add foreign keys (FK) to match the PK, as the second of two steps to replace and represent associations – on the side of the association that has 0..* or 1..* or 0..1



Course
courseID (1..1) PK
courseName (1..1)

CourseOccasion
courseOccasionID (1..1) PK
startDate (1..1)
courseID (1..1) (FK to Course.courseID)

Mobile
mobileID (1..1) PK
mobileNo (1..1)
studentID (1..1) (FK to Student.studentID)

Registration
registrationID (1..1) PK
regDate (1..1)
courseOccasionID (1..1) (FK to CourseOccasion.courseOccasionID)
studentID (1..1) (FK to Student.studentID

Student
studentID (1..1) PK
studentName (1..1)

CourseResponsible
courseResponsibleID (1..1) PK
courseOccasionID (1..1) (FK to Course.courseID)
teacherID (1..1) (FK to Teacher.teacherID)

TeacherArea
teacherAreaID (1..1) PK
areaID (1..1) (FK to Area.areaID)
teacherID (1..1) (FK to Teacher.teacherID)

MasterStudent
masterStudentID (1..1) PK
bachelorExamYear (1..1)

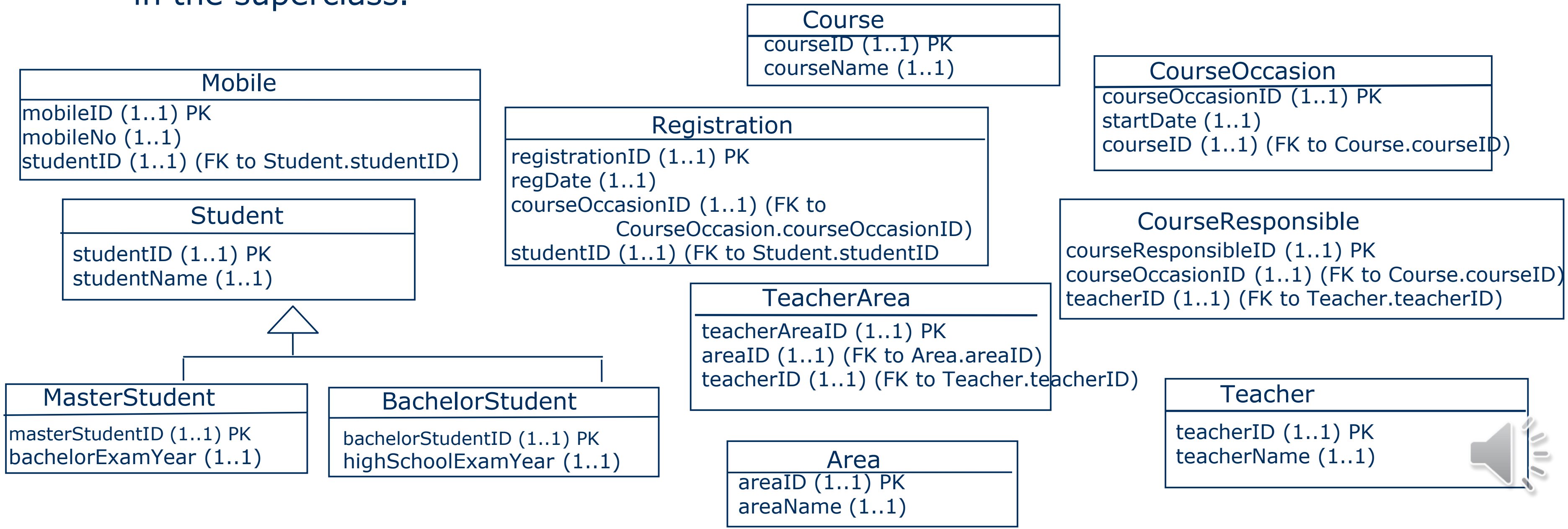
BachelorStudent
bachelorStudentID (1..1) PK
highSchoolExamYear (1..1)

Teacher
teacherID (1..1) PK
teacherName (1..1)

Area
areaID (1..1) PK
areaName (1..1)

Towards a RDM - Step 7

- TO DO: We also need to manage the generalization/specialization relationship. We do that by letting the primary keys in subclasses be foreign keys to the primary key in the superclass.



Course
courseID (1..1) PK
courseName (1..1)

Mobile
mobileID (1..1) PK
mobileNo (1..1)
studentID (1..1) (FK to Student.studentID)

Registration
registrationID (1..1) PK
regDate (1..1)
courseOccasionID (1..1) (FK to CourseOccasion.courseOccasionID)
studentID (1..1) (FK to Student.studentID

CourseOccasion
courseOccasionID (1..1) PK
startDate (1..1)
courseID (1..1) (FK to Course.courseID)

Student
studentID (1..1) PK
studentName (1..1)

CourseResponsible
courseResponsibleID (1..1) PK
courseOccasionID (1..1) (FK to Course.courseID)
teacherID (1..1) (FK to Teacher.teacherID)

TeacherArea
teacherAreaID (1..1) PK
areaID (1..1) (FK to Area.areaID)
teacherID (1..1) (FK to Teacher.teacherID)

MasterStudent
masterStudentID (1..1) PK
bachelorExamYear (1..1)

BachelorStudent
bachelorStudentID (1..1) PK
highSchoolExamYear (1..1)

Teacher
teacherID (1..1) PK
teacherName (1..1)

Area
areaID (1..1) PK
areaName (1..1)

Towards a RDM - Step 7

- DONE: Let the primary keys in subclasses be foreign keys to the primary key in the superclass. Why? One way to transfer generalization/specialization towards a relational database model

Mobile
mobileID (1..1) PK
mobileNo (1..1)
studentID (1..1) (FK to Student.studentID)

Student
studentID (1..1) PK
studentName (1..1)

MasterStudent
masterStudentID (1..1) PK (FK to Student.studentID)
bachelorExamYear (1..1)

BachelorStudent
bachelorStudentID (1..1) PK (FK to Student.studentID)
highSchoolExamYear (1..1)

Course
courseID (1..1) PK
courseName (1..1)

Registration
registrationID (1..1) PK
regDate (1..1)
courseOccasionID (1..1) (FK to
CourseOccasion.courseOccasionID)
studentID (1..1) (FK to Student.studentID

TeacherArea
teacherAreaID (1..1) PK
areaID (FK to Area.areaID)
teacherID (1..1) (FK to Teacher.teacherID)

Area
areaID (1..1) PK
areaName (1..1)

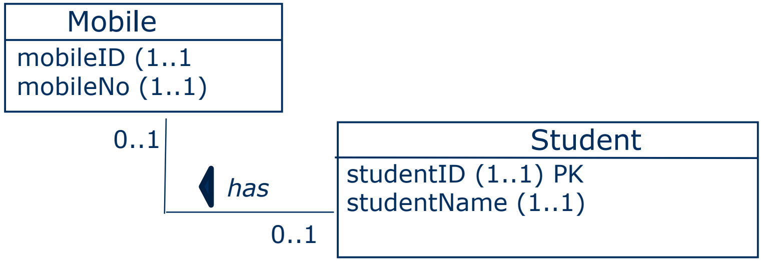
CourseOccasion
courseOccasionID (1..1) PK
startDate (1..1)
courseID (1..1) (FK to Course.courseID)

CourseResponsible
courseResponsibleID (1..1) PK
courseOccasionID (1..1) (FK to Course.courseID)
teacherID (1..1) (FK to Teacher.teacherID)

Teacher
teacherID (1..1) PK
teacherName (1..1)

Towards a RDM - Step 8

- TO DO: If there is an accociation with the multiplicity 0..1 on both side – add a new table



Towards a RDM – Step 8

- DONE: If there is an accociation with the multiplicity 0..1 on both side – add a new table

Note, not used in our case, therefore not added to the case

Mobile

mobileID (1..1) PK
mobileNo (1..1)

StudentHasMobile

studentHasMobileID (1..1) PK
studentID (1..1) (FK to Student.studentID
mobileID (1..1) (FK to Mobile.mobileID)

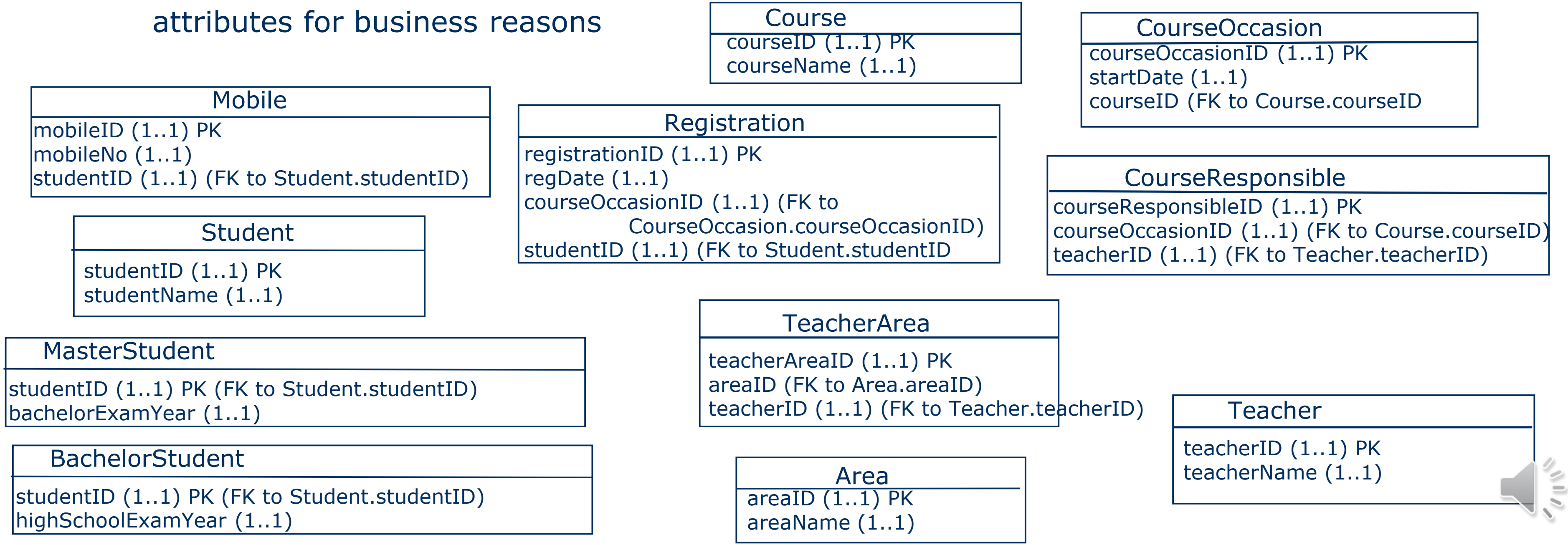
Student

studentID (1..1) PK
studentName (1..1)

Towards a RDM – Step 9

- TO DO: Identify Alternative Keys (AK), that also can uniquely identify objects of a class.

  Why? For data quality reason. We may need to enforce the use of unique values for these attributes for business reasons



Mobile
mobileID (1..1) PK
mobileNo (1..1)
studentID (1..1) (FK to Student.studentID)

Student
studentID (1..1) PK
studentName (1..1)

MasterStudent
studentID (1..1) PK (FK to Student.studentID)
bachelorExamYear (1..1)

BachelorStudent
studentID (1..1) PK (FK to Student.studentID)
highSchoolExamYear (1..1)

Course
courseID (1..1) PK
courseName (1..1)

Registration
registrationID (1..1) PK
regDate (1..1)
courseOccasionID (1..1) (FK to CourseOccasion.courseOccasionID)
studentID (1..1) (FK to Student.studentID

TeacherArea
teacherAreaID (1..1) PK
areaID (FK to Area.areaID)
teacherID (1..1) (FK to Teacher.teacherID)

Area
areaID (1..1) PK
areaName (1..1)

CourseOccasion
courseOccasionID (1..1) PK
startDate (1..1)
courseID (FK to Course.courseID

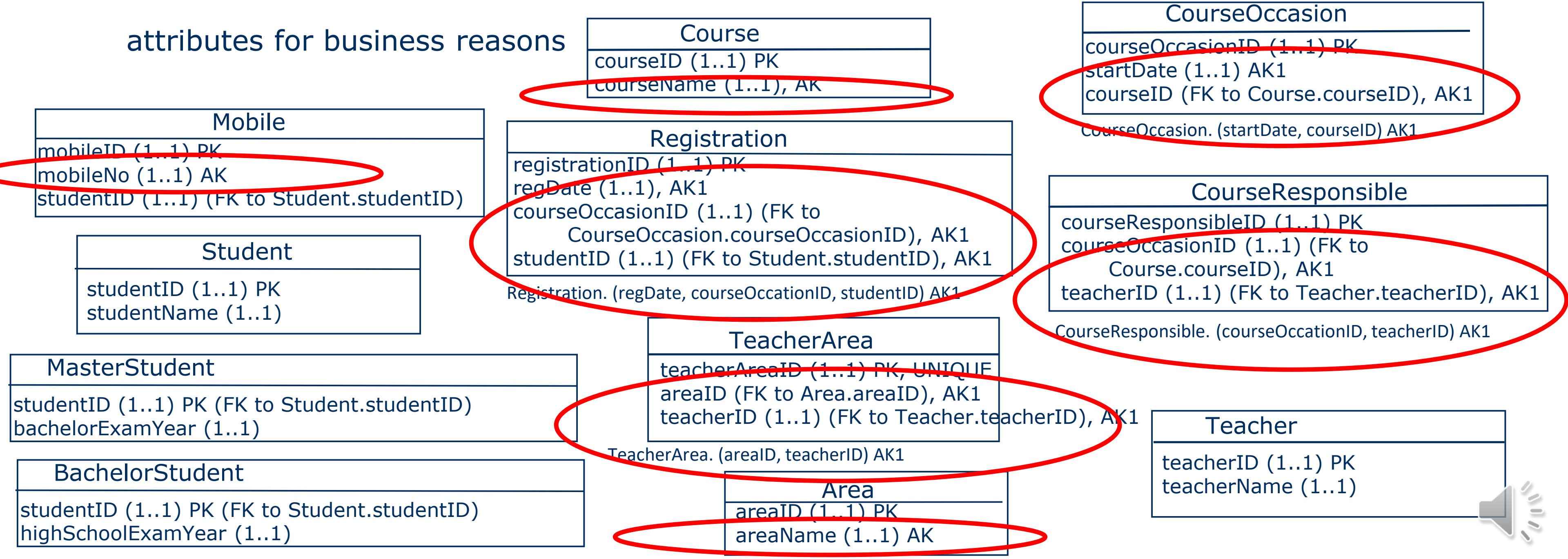
CourseResponsible
courseResponsibleID (1..1) PK
courseOccasionID (1..1) (FK to Course.courseID)
teacherID (1..1) (FK to Teacher.teacherID)

Teacher
teacherID (1..1) PK
teacherName (1..1)

Towards a RDM – Step 9

- DONE: Identify Alternative Keys (AK), that also can uniquely identify objects of a class.

  Why? For data quality reason. We may need to enforce the use of unique values for these attributes for business reasons



Course
courseID (1..1) PK
courseName (1..1), AK

CourseOccasion
courseOccasionID (1..1) PK
startDate (1..1) AK1
courseID (FK to Course.courseID), AK1

CourseOccasion. (startDate, courseID) AK1

Mobile
mobileID (1..1) PK
mobileNo (1..1) AK
studentID (1..1) (FK to Student.studentID)

Registration
registrationID (1..1) PK
regDate (1..1), AK1
courseOccasionID (1..1) (FK to CourseOccasion.courseOccasionID), AK1
studentID (1..1) (FK to Student.studentID), AK1

Registration. (regDate, courseOccationID, studentID) AK1

CourseResponsible
courseResponsibleID (1..1) PK
courseOccasionID (1..1) (FK to Course.courseID), AK1
teacherID (1..1) (FK to Teacher.teacherID), AK1

CourseResponsible. (courseOccationID, teacherID) AK1

Student
studentID (1..1) PK
studentName (1..1)

MasterStudent
studentID (1..1) PK (FK to Student.studentID)
bachelorExamYear (1..1)

TeacherArea
teacherAreaID (1..1) PK, UNIQUE
areaID (FK to Area.areaID), AK1
teacherID (1..1) (FK to Teacher.teacherID), AK1

TeacherArea. (areaID, teacherID) AK1

Teacher
teacherID (1..1) PK
teacherName (1..1)

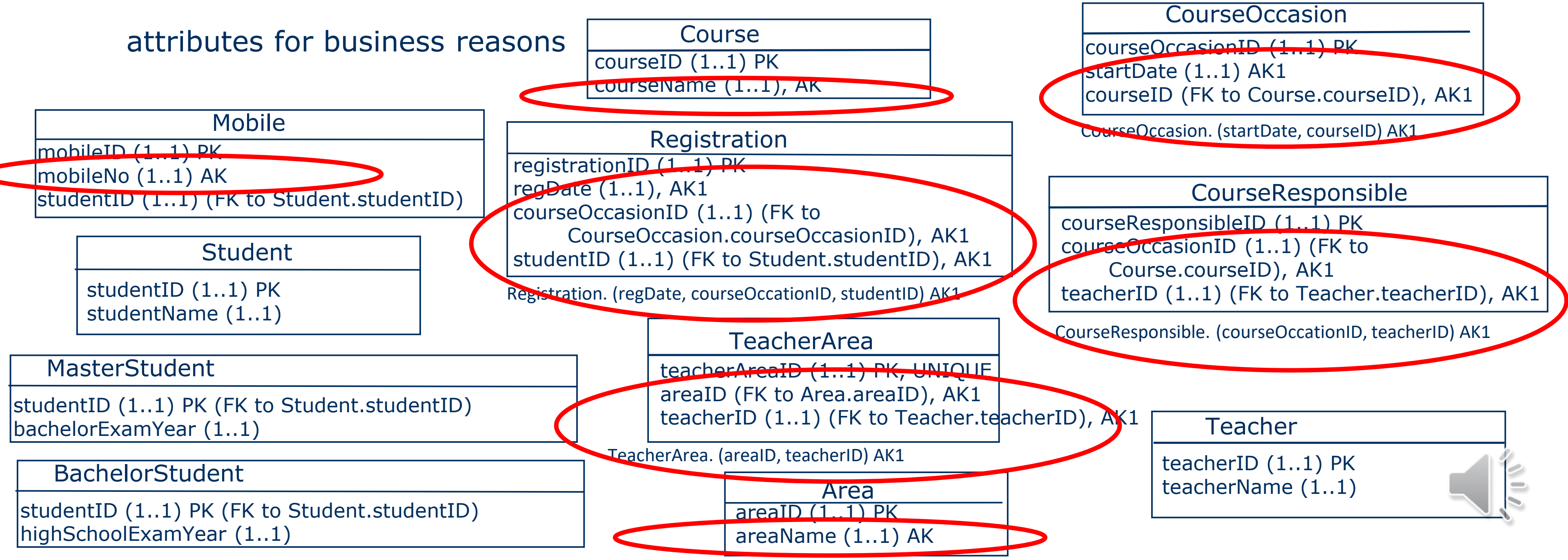
BachelorStudent
studentID (1..1) PK (FK to Student.studentID)
highSchoolExamYear (1..1)

Area
areaID (1..1) PK
areaName (1..1) AK

Towards a RDM – Step 9

- DONE: Identify Alternative Keys (AK), that also can uniquely identify objects of a class.

  Why? For data quality reason. We may need to enforce the use of unique values for these attributes for business reasons



Course
courseID (1..1) PK
courseName (1..1), AK

CourseOccasion
courseOccasionID (1..1) PK
startDate (1..1) AK1
courseID (FK to Course.courseID), AK1

CourseOccasion. (startDate, courseID) AK1

Mobile
mobileID (1..1) PK
mobileNo (1..1) AK
studentID (1..1) (FK to Student.studentID)

Registration
registrationID (1..1) PK
regDate (1..1), AK1
courseOccasionID (1..1) (FK to CourseOccasion.courseOccasionID), AK1
studentID (1..1) (FK to Student.studentID), AK1

Registration. (regDate, courseOccationID, studentID) AK1

Student
studentID (1..1) PK
studentName (1..1)

CourseResponsible
courseResponsibleID (1..1) PK
courseOccasionID (1..1) (FK to Course.courseID), AK1
teacherID (1..1) (FK to Teacher.teacherID), AK1

CourseResponsible. (courseOccationID, teacherID) AK1

MasterStudent
studentID (1..1) PK (FK to Student.studentID)
bachelorExamYear (1..1)

TeacherArea
teacherAreaID (1..1) PK, UNIQUE
areaID (FK to Area.areaID), AK1
teacherID (1..1) (FK to Teacher.teacherID), AK1

TeacherArea. (areaID, teacherID) AK1

Teacher
teacherID (1..1) PK
teacherName (1..1)

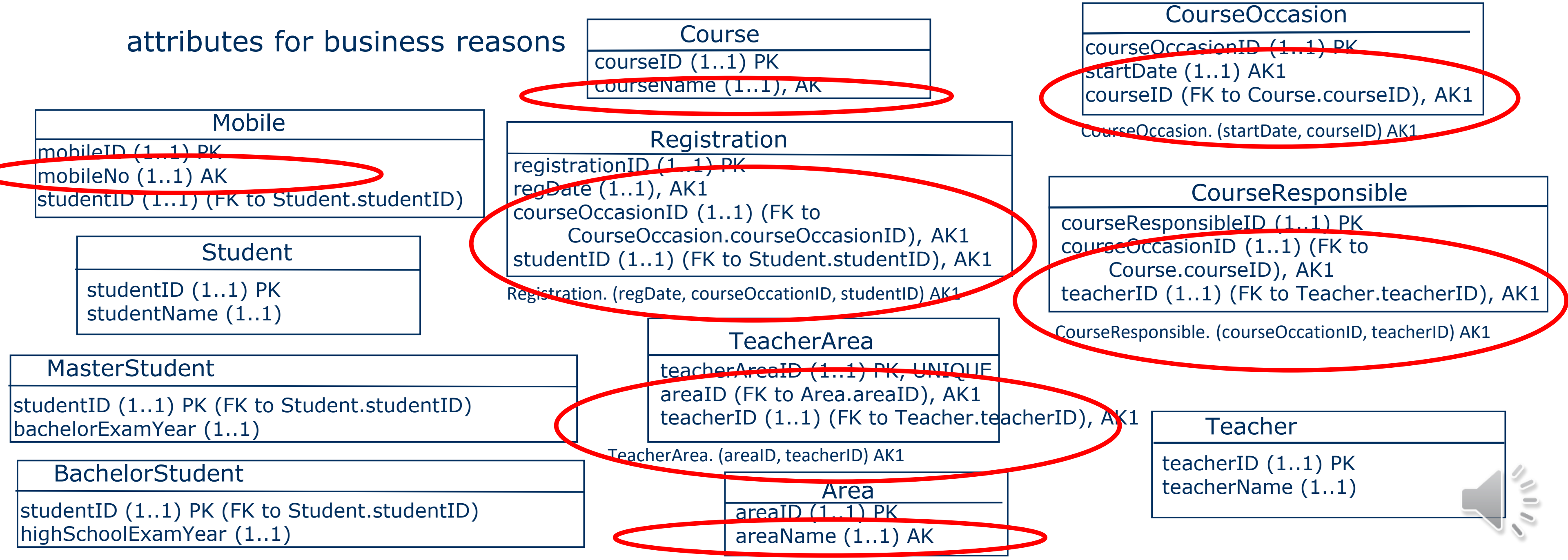
BachelorStudent
studentID (1..1) PK (FK to Student.studentID)
highSchoolExamYear (1..1)

Area
areaID (1..1) PK
areaName (1..1) AK

Towards a RDM – Step 9

- DONE: Identify Alternative Keys (AK), that also can uniquely identify objects of a class.

  Why? For data quality reason. We may need to enforce the use of unique values for these attributes for business reasons



Course
courseID (1..1) PK
courseName (1..1), AK

CourseOccasion
courseOccasionID (1..1) PK
startDate (1..1) AK1
courseID (FK to Course.courseID), AK1

CourseOccasion. (startDate, courseID) AK1

Mobile
mobileID (1..1) PK
mobileNo (1..1) AK
studentID (1..1) (FK to Student.studentID)

Registration
registrationID (1..1) PK
regDate (1..1), AK1
courseOccasionID (1..1) (FK to CourseOccasion.courseOccasionID), AK1
studentID (1..1) (FK to Student.studentID), AK1

Registration. (regDate, courseOccationID, studentID) AK1

CourseResponsible
courseResponsibleID (1..1) PK
courseOccasionID (1..1) (FK to Course.courseID), AK1
teacherID (1..1) (FK to Teacher.teacherID), AK1

CourseResponsible. (courseOccationID, teacherID) AK1

Student
studentID (1..1) PK
studentName (1..1)

MasterStudent
studentID (1..1) PK (FK to Student.studentID)
bachelorExamYear (1..1)

TeacherArea
teacherAreaID (1..1) PK, UNIQUE
areaID (FK to Area.areaID), AK1
teacherID (1..1) (FK to Teacher.teacherID), AK1

TeacherArea. (areaID, teacherID) AK1

Teacher
teacherID (1..1) PK
teacherName (1..1)

BachelorStudent
studentID (1..1) PK (FK to Student.studentID)
highSchoolExamYear (1..1)

Area
areaID (1..1) PK
areaName (1..1) AK

Towards a RDM – Step 9

- DONE: Identify Alternative Keys (AK), that also can uniquely identify objects of a class.

  Why? For data quality reason. We may need to enforce the use of unique values for these attributes for business reasons

Course
courseID (1..1) PK
courseName (1..1)

Course. courseName AK

CourseOccasion
courseOccasionID (1..1) PK
startDate (1..1)
courseID (FK to Course.courseID)

Mobile
mobileID (1..1) PK
mobileNo (1..1)
studentID (1..1) (FK to Student.studentID)

Mobile. mobileNo AK

Registration
registrationID (1..1) PK
regDate (1..1),
courseOccasionID (1..1) (FK to CourseOccasion.courseOccasionID)
studentID (1..1) (FK to Student.studentID)

Registration. (regDate, courseOcctionID, studentID) AK

CourseResponsible
courseResponsibleID (1..1) PK
courseOccasionID (1..1) (FK to Course.courseID)
teacherID (1..1) (FK to Teacher.teacherID)

CourseResponsible. (courseOcctionID, teacherID) AK

Student
studentID (1..1) PK
studentName (1..1)

MasterStudent
studentID (1..1) PK (FK to Student.studentID)
bachelorExamYear (1..1)

TeacherArea
teacherAreaID (1..1) PK
areaID (FK to Area.areaID)
teacherID (1..1) (FK to Teacher.teacherID)

TeacherArea. (areaID, teacherID) AK

Teacher
teacherID (1..1) PK
teacherName (1..1)

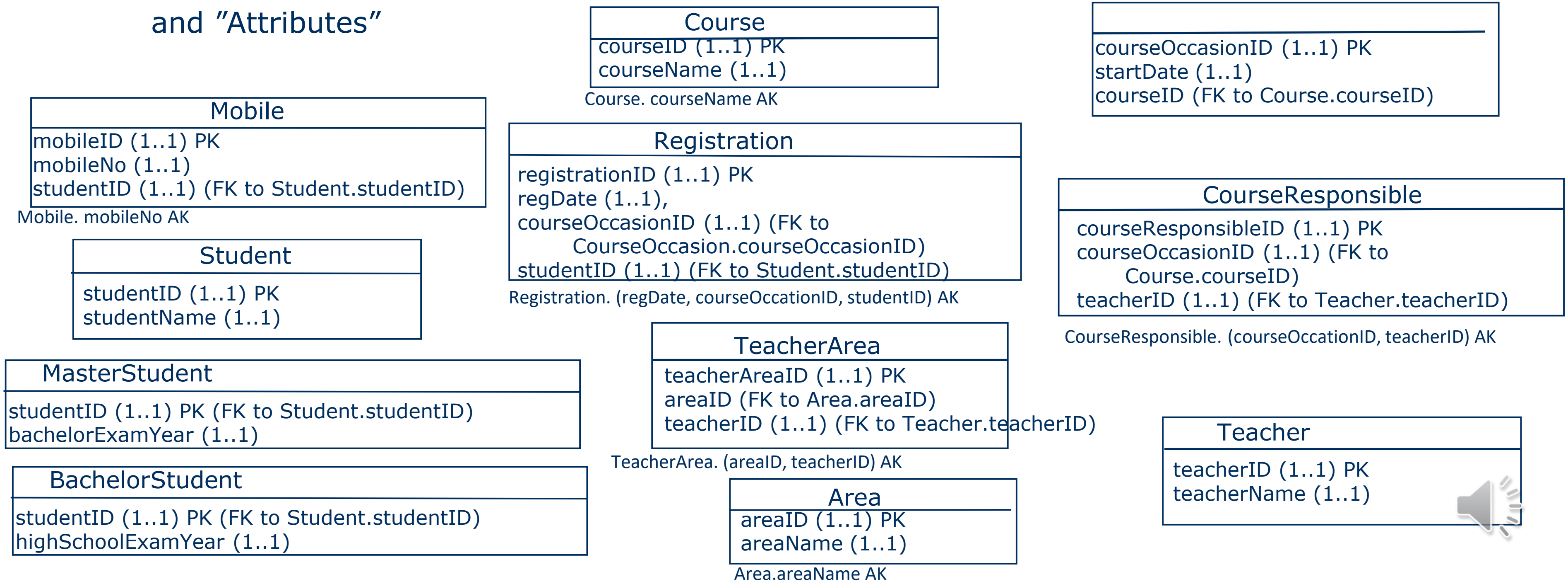
BachelorStudent
studentID (1..1) PK (FK to Student.studentID)
highSchoolExamYear (1..1)

Area
areaID (1..1) PK
areaName (1..1)

Area.areaName AK

Towards a RDM – Step 10

- TO DO: Transform the ”Classes” to ”Tables” (or ”Relations”) and ”Attributes” to ”Columns” (or ”Table definitions”) - that is, use the terms ”Tables” and ”Columns” instead ot ”Classes” and ”Attributes”



Course
courseID (1..1) PK
courseName (1..1)
Course. courseName AK

courseOccasionID (1..1) PK
startDate (1..1)
courseID (FK to Course.courseID)

Mobile
mobileID (1..1) PK
mobileNo (1..1)
studentID (1..1) (FK to Student.studentID)
Mobile. mobileNo AK

Registration
registrationID (1..1) PK
regDate (1..1),
courseOccasionID (1..1) (FK to CourseOccasion.courseOccasionID)
studentID (1..1) (FK to Student.studentID)
Registration. (regDate, courseOcctionID, studentID) AK

CourseResponsible
courseResponsibleID (1..1) PK
courseOccasionID (1..1) (FK to Course.courseID)
teacherID (1..1) (FK to Teacher.teacherID)
CourseResponsible. (courseOcctionID, teacherID) AK

Student
studentID (1..1) PK
studentName (1..1)

MasterStudent
studentID (1..1) PK (FK to Student.studentID)
bachelorExamYear (1..1)

TeacherArea
teacherAreaID (1..1) PK
areaID (FK to Area.areaID)
teacherID (1..1) (FK to Teacher.teacherID)
TeacherArea. (areaID, teacherID) AK

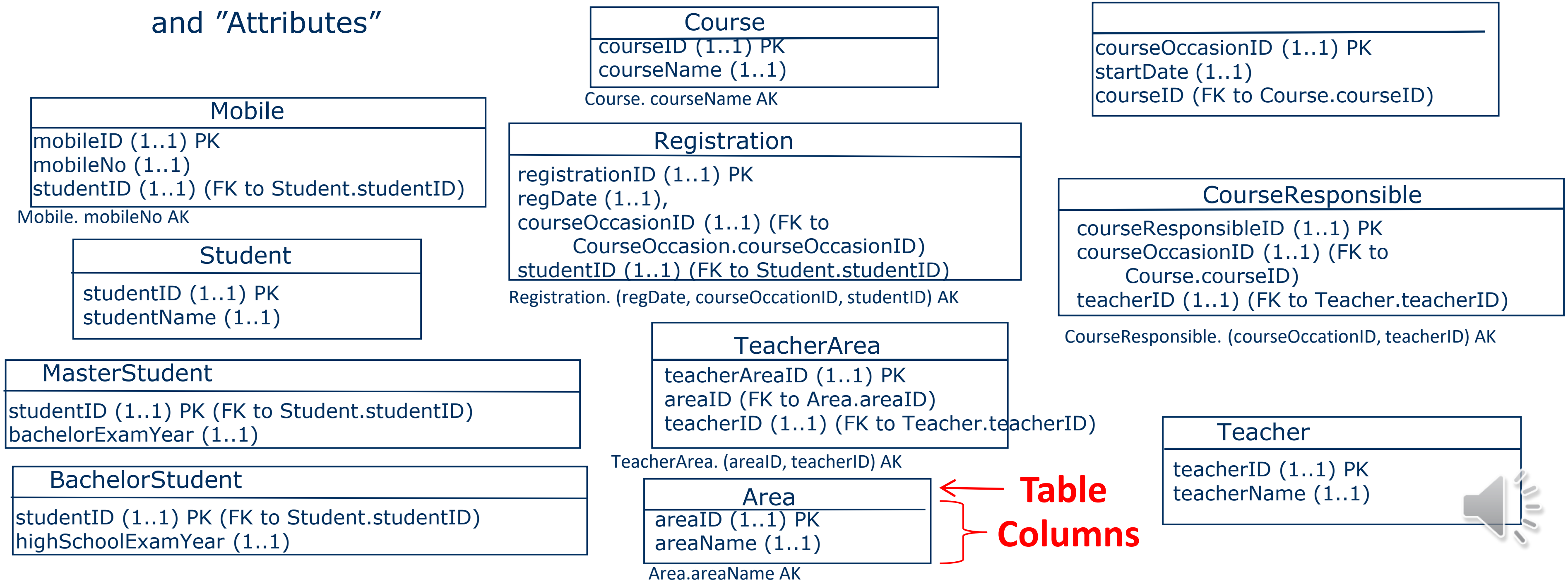
BachelorStudent
studentID (1..1) PK (FK to Student.studentID)
highSchoolExamYear (1..1)

Area
areaID (1..1) PK
areaName (1..1)
Area.areaName AK

Teacher
teacherID (1..1) PK
teacherName (1..1)

Towards a RDM – Step 10

- TO DO: Transform the ”Classes” to ”Tables” (or ”Relations”) and ”Attributes” to ”Columns” (or ”Table definitions”) - that is, use the terms ”Tables” and ”Columns” instead ot ”Classes” and ”Attributes”



Course
courseID (1..1) PK
courseName (1..1)
Course. courseName AK

courseOccasionID (1..1) PK
startDate (1..1)
courseID (FK to Course.courseID)

Mobile
mobileID (1..1) PK
mobileNo (1..1)
studentID (1..1) (FK to Student.studentID)
Mobile. mobileNo AK

Registration
registrationID (1..1) PK
regDate (1..1),
courseOccasionID (1..1) (FK to CourseOccasion.courseOccasionID)
studentID (1..1) (FK to Student.studentID)
Registration. (regDate, courseOcctionID, studentID) AK

CourseResponsible
courseResponsibleID (1..1) PK
courseOccasionID (1..1) (FK to Course.courseID)
teacherID (1..1) (FK to Teacher.teacherID)
CourseResponsible. (courseOcctionID, teacherID) AK

Student
studentID (1..1) PK
studentName (1..1)

MasterStudent
studentID (1..1) PK (FK to Student.studentID)
bachelorExamYear (1..1)

TeacherArea
teacherAreaID (1..1) PK
areaID (FK to Area.areaID)
teacherID (1..1) (FK to Teacher.teacherID)
TeacherArea. (areaID, teacherID) AK

Teacher
teacherID (1..1) PK
teacherName (1..1)

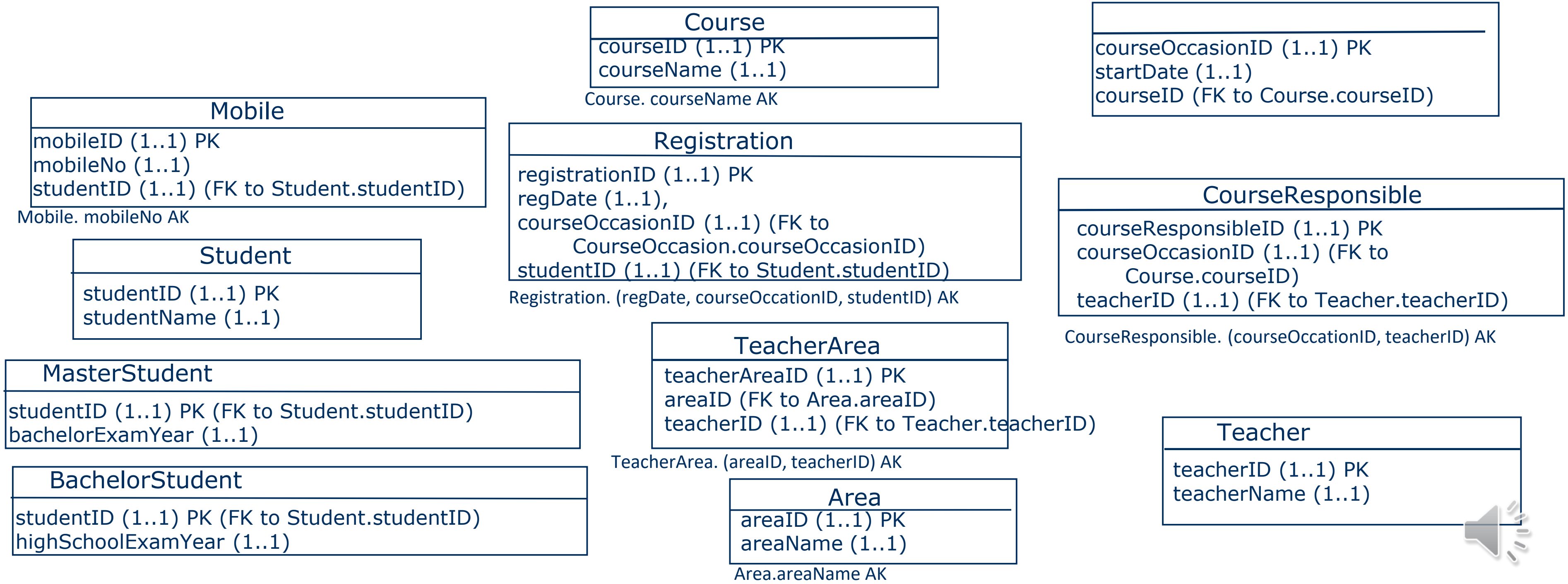
BachelorStudent
studentID (1..1) PK (FK to Student.studentID)
highSchoolExamYear (1..1)

Area
areaID (1..1) PK
areaName (1..1)
Area.areaName AK

Table
Columns

Relational Database Model

- FINALLY: We have the Relational Database Model



Course
courseID (1..1) PK
courseName (1..1)
Course. courseName AK

courseOccasionID (1..1) PK
startDate (1..1)
courseID (FK to Course.courseID)

Mobile
mobileID (1..1) PK
mobileNo (1..1)
studentID (1..1) (FK to Student.studentID)
Mobile. mobileNo AK

Registration
registrationID (1..1) PK
regDate (1..1),
courseOccasionID (1..1) (FK to CourseOccasion.courseOccasionID)
studentID (1..1) (FK to Student.studentID)
Registration. (regDate, courseOcctionID, studentID) AK

CourseResponsible
courseResponsibleID (1..1) PK
courseOccasionID (1..1) (FK to Course.courseID)
teacherID (1..1) (FK to Teacher.teacherID)
CourseResponsible. (courseOcctionID, teacherID) AK

Student
studentID (1..1) PK
studentName (1..1)

MasterStudent
studentID (1..1) PK (FK to Student.studentID)
bachelorExamYear (1..1)

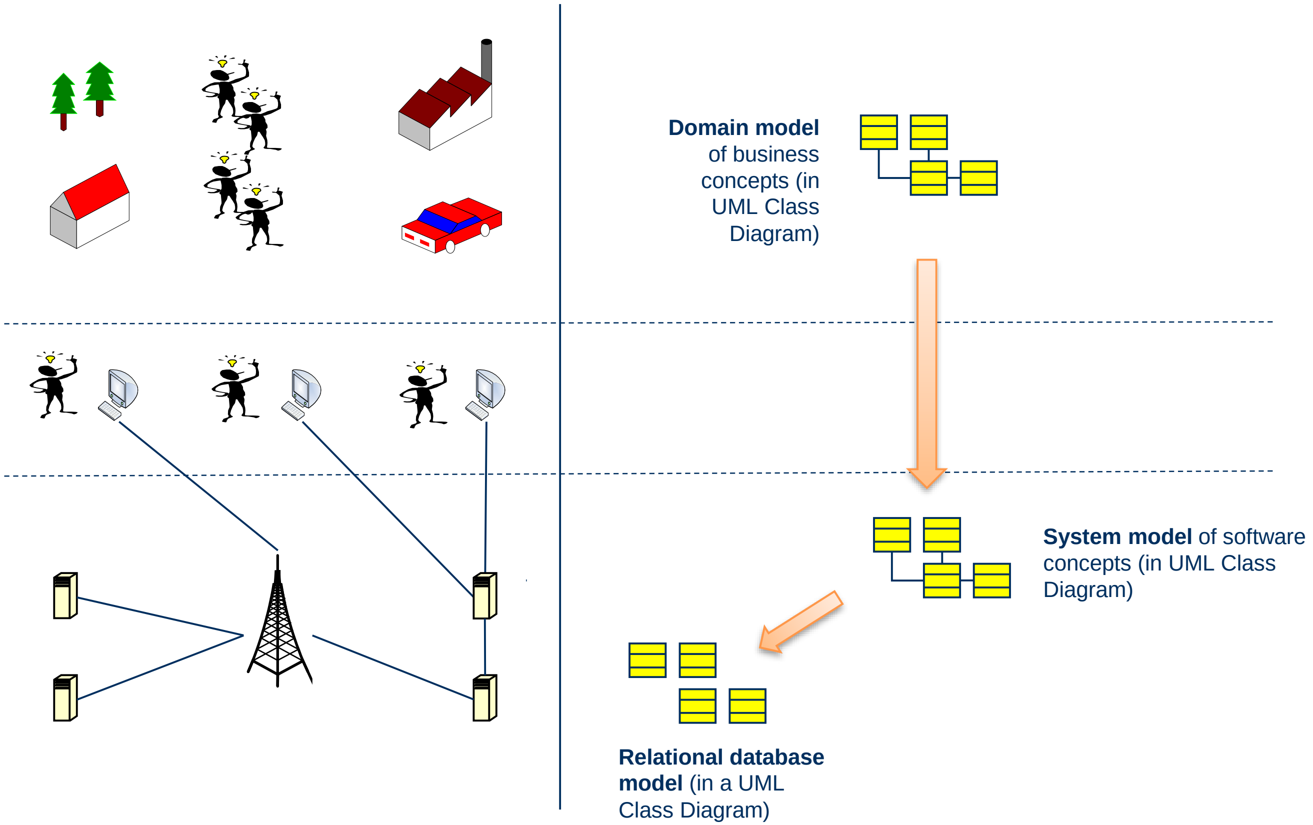
TeacherArea
teacherAreaID (1..1) PK
areaID (FK to Area.areaID)
teacherID (1..1) (FK to Teacher.teacherID)
TeacherArea. (areaID, teacherID) AK

BachelorStudent
studentID (1..1) PK (FK to Student.studentID)
highSchoolExamYear (1..1)

Teacher
teacherID (1..1) PK
teacherName (1..1)

Area
areaID (1..1) PK
areaName (1..1)
Area.areaName AK

A Relational Database Model

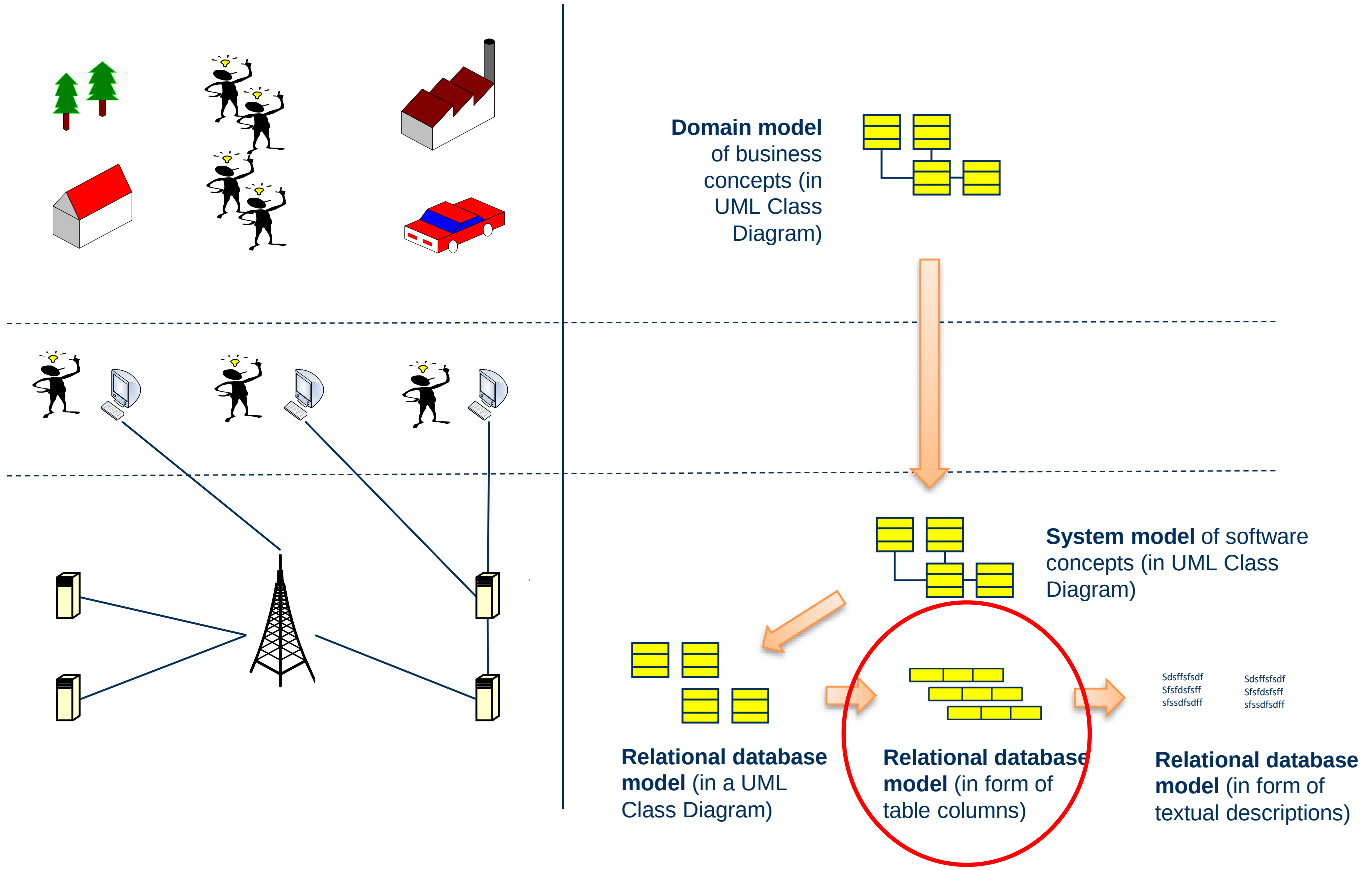


Domain model of business concepts (in UML Class Diagram)

System model of software concepts (in UML Class Diagram)

Relational database model (in a UML Class Diagram)

Different forms of RDMs



Domain model of business concepts (in UML Class Diagram)

System model of software concepts (in UML Class Diagram)

Relational database model (in a UML Class Diagram)

Relational database model (in form of table columns)

Relational database model (in form of textual descriptions)

Towards a RDM in form of table columns

Student

studentID (1..1) PK
studentName (1..1)

Registration

registrationID (1..1) PK
regDate (1..1)
courseOccasionID (1..1) (FK to
CourseOccasion.courseOccasionID)
studentID (1..1) (FK to Student.studentID)

Registration. (regDate, courseOccationID, studentID) AK

Course

courseID (1..1) PK
courseName (1..1)

CourseOccasion

courseOccasionID (1..1) PK
startDate (1..1)
courseID (FK to Course.courseID),

CourseOccasion. (startDate, courseID) AK

Towards a RDM in form of table columns

Student
studentID (1..1) PK
studentName (1..1)

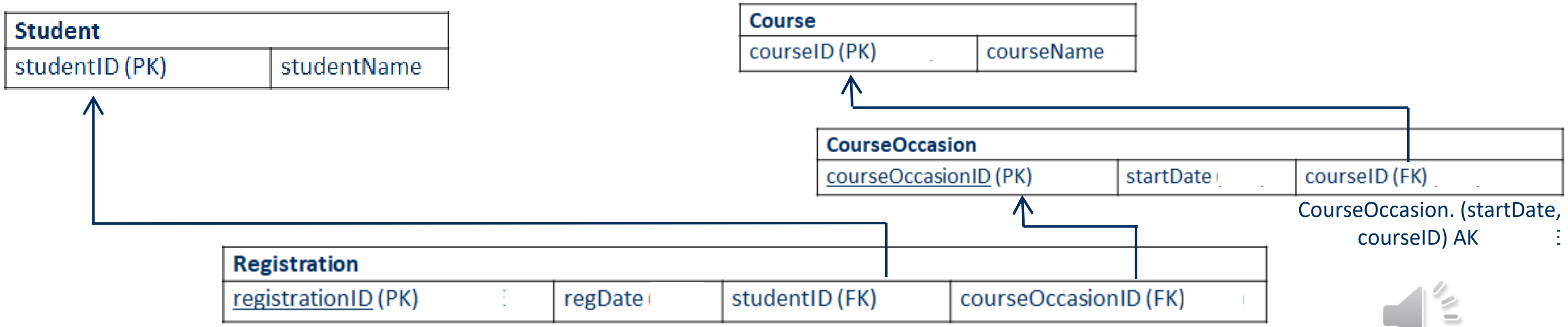
Registration
registrationID (1..1) PK
regDate (1..1)
courseOccasionID (1..1) (FK to
CourseOccasion.courseOccasionID)
studentID (1..1) (FK to Student.studentID)

Registration. (regDate, courseOccationID, studentID) AK

Course
courseID (1..1) PK
courseName (1..1)

CourseOccasion
courseOccasionID (1..1) PK
startDate (1..1)
courseID (FK to Course.courseID)

CourseOccasion. (startDate, courseID) AK



Student	
studentID (PK)	studentName

Course	
courseID (PK)	courseName

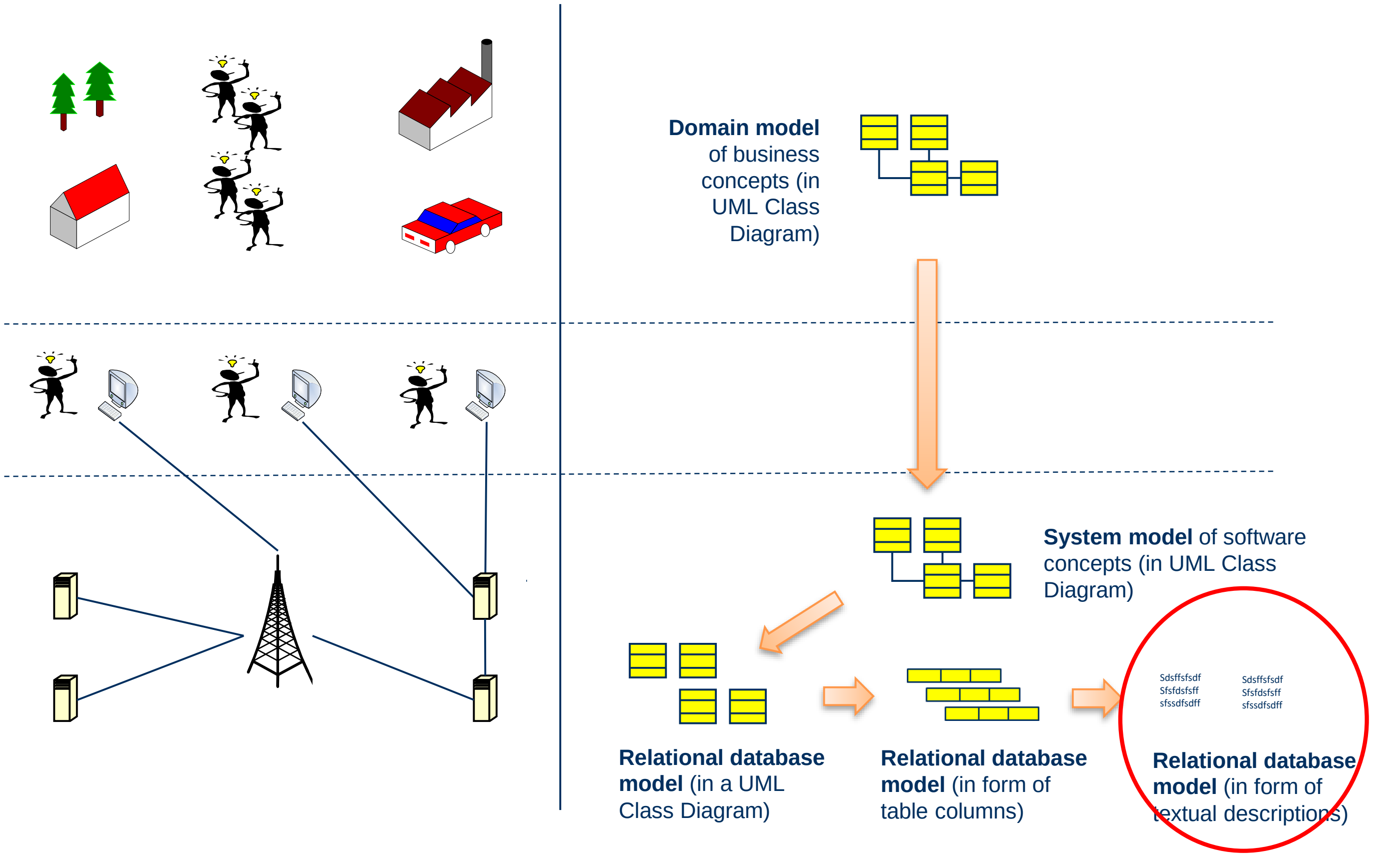
CourseOccasion		
courseOccasionID (PK)	startDate	courseID (FK)

CourseOccasion. (startDate, courseID) AK

Registration			
registrationID (PK)	regDate	studentID (FK)	courseOccasionID (FK)

Registration. (regDate, courseOccationID, studentID) AK

Different forms of RDMs



Domain model of business concepts (in UML Class Diagram)

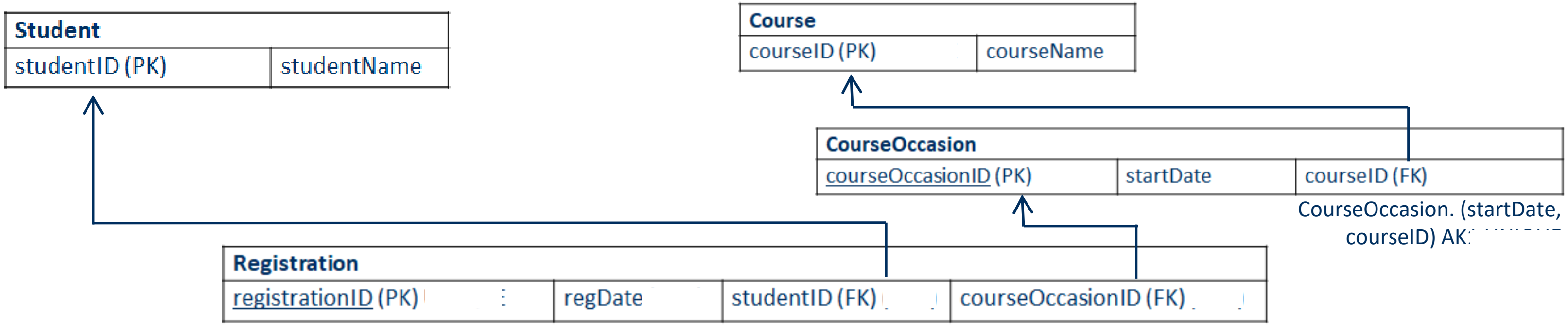
System model of software concepts (in UML Class Diagram)

Relational database model (in a UML Class Diagram)

Relational database model (in form of table columns)

Relational database model (in form of textual descriptions)

Towards a RDM in form of textual descriptions



Student	
studentID (PK)	studentName

Course	
courseID (PK)	courseName

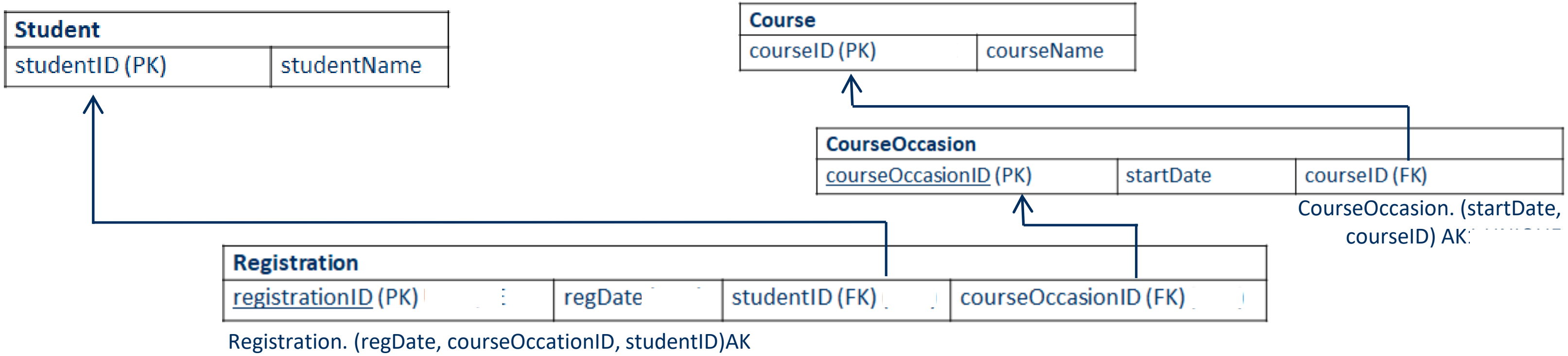
CourseOccasion		
courseOccasionID (PK)	startDate	courseID (FK)

CourseOccasion. (startDate, courseID) AK:

Registration			
registrationID (PK)	regDate	studentID (FK)	courseOccasionID (FK)

Registration. (regDate, courseOccationID, studentID)AK

Towards a RDM in form of textual descriptions



Student	
studentID (PK)	studentName

Course	
courseID (PK)	courseName

CourseOccasion		
courseOccasionID (PK)	startDate	courseID (FK)

CourseOccasion. (startDate, courseID) AK:

Registration			
registrationID (PK)	regDate	studentID (FK)	courseOccasionID (FK)

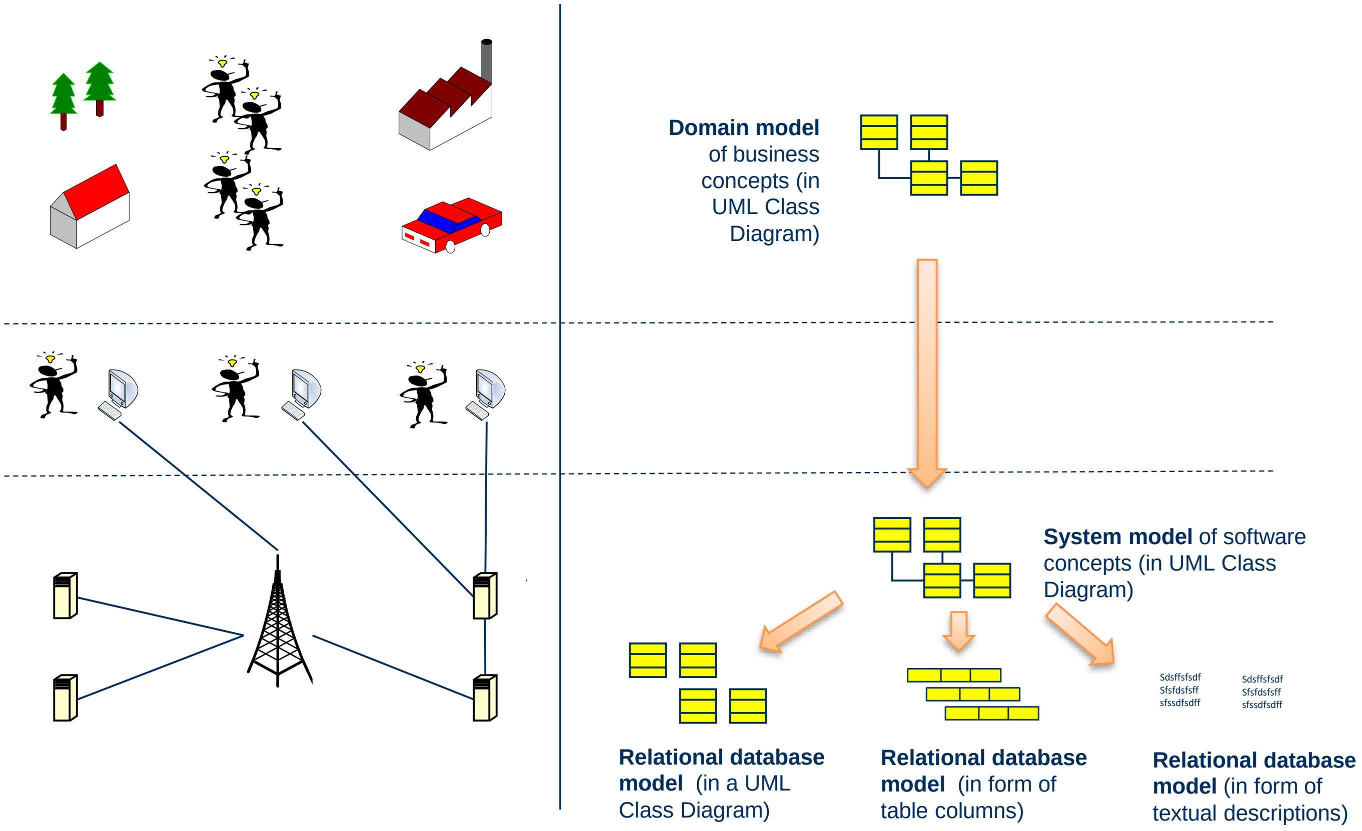
Registration. (regDate, courseOccationID, studentID)AK

Tables:
Student (studentID, studentName)
Course (courseID, courseName)
CourseOccasion (courseOccasionID, startDate, courseID)
Registration (registrationID, RegDate, studentID, courseOccasionID)

FKs
Registration.studentID is FK towards Student.studentID
Registration.courseOccasionID is FK towards CourseOccasion.courseOccasionID
CourseOccasion.CourseID is FK towards Course.ourseID

AKs
Registration. (regDate, courseOccationID, studentID) AK
CourseOccasion. (startDate, courseID) AK

Different forms of RDMs



Domain model of business concepts (in UML Class Diagram)

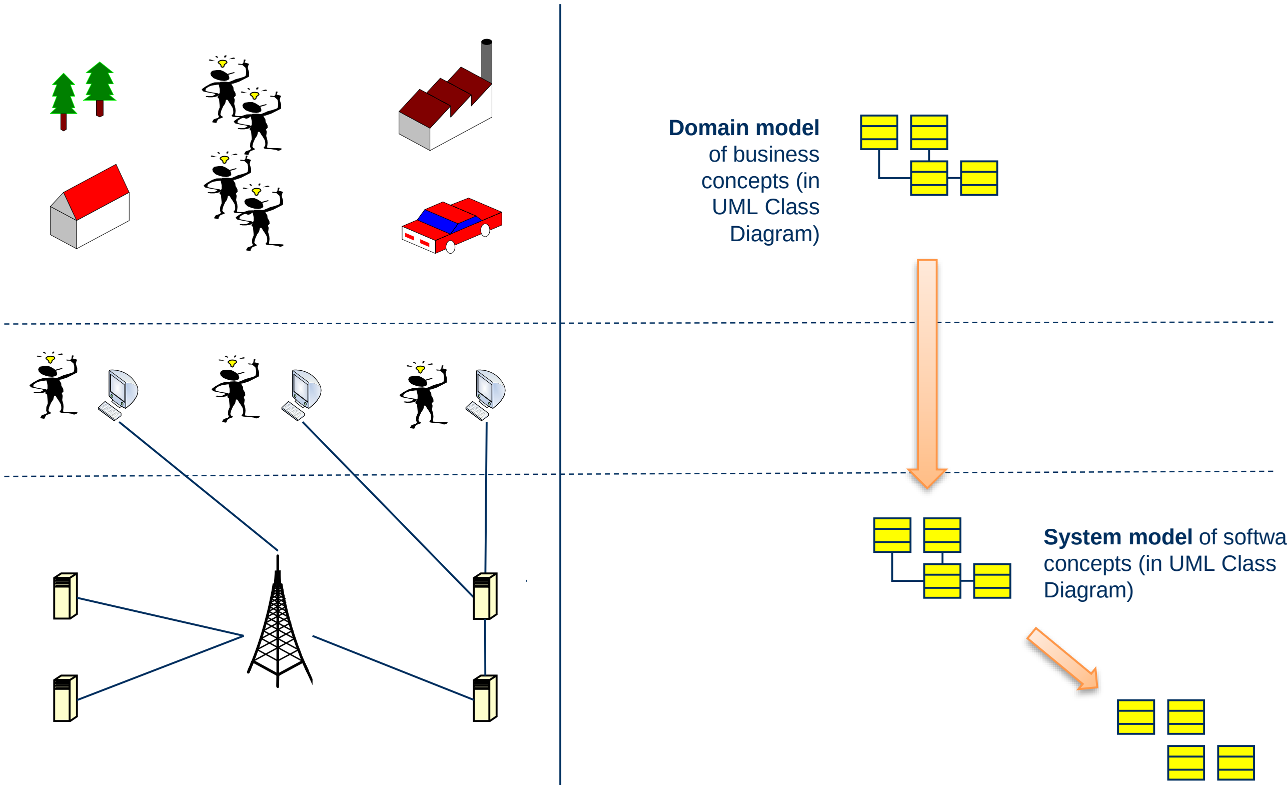
System model of software concepts (in UML Class Diagram)

Relational database model (in a UML Class Diagram)

Relational database model (in form of table columns)

Relational database model (in form of textual descriptions)

Towards Model Driven Development

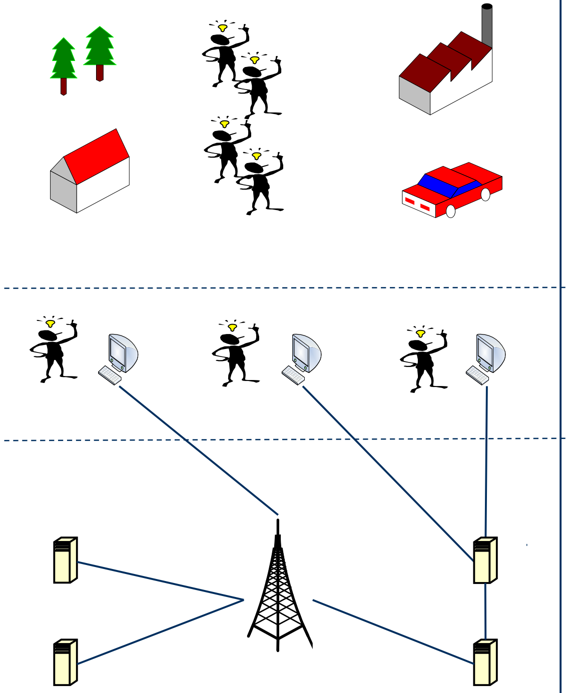


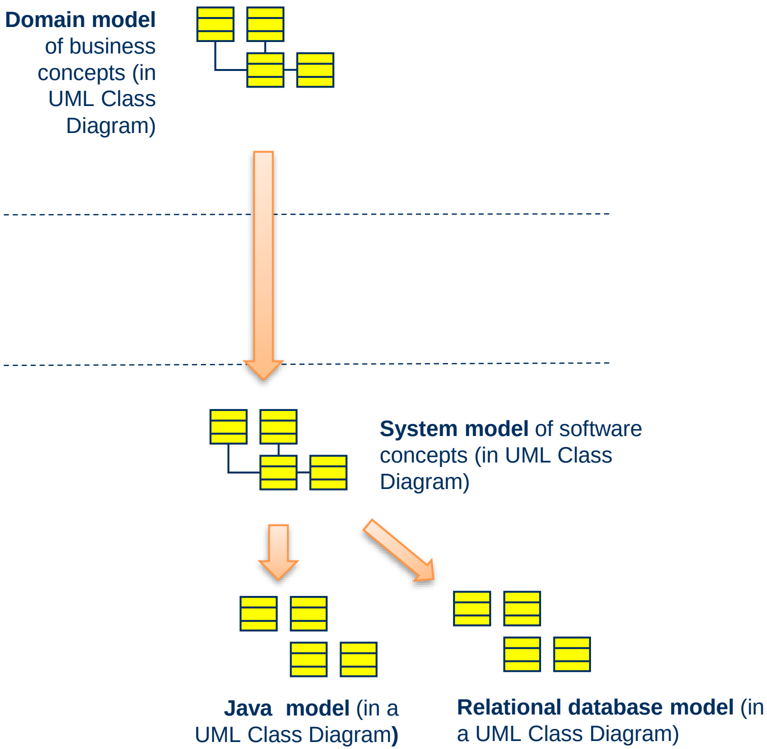
Domain model of business concepts (in UML Class Diagram)

System model of software concepts (in UML Class Diagram)

Relational database model (in a UML Class Diagram)

Towards Model Driven Development





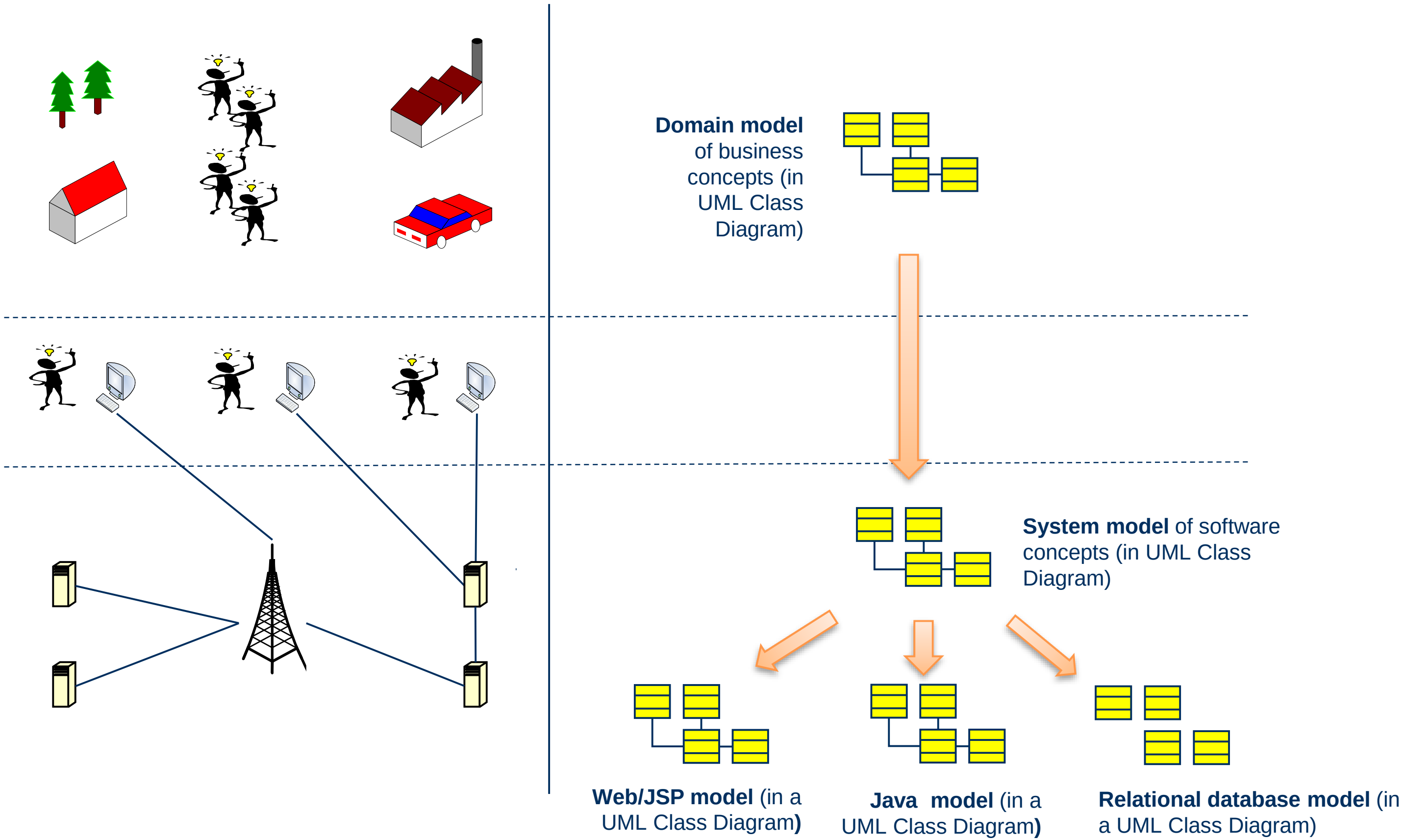
Domain model of business concepts (in UML Class Diagram)

System model of software concepts (in UML Class Diagram)

Java model (in a UML Class Diagram)

Relational database model (in a UML Class Diagram)

Towards Model Driven Development



Domain model of business concepts (in UML Class Diagram)

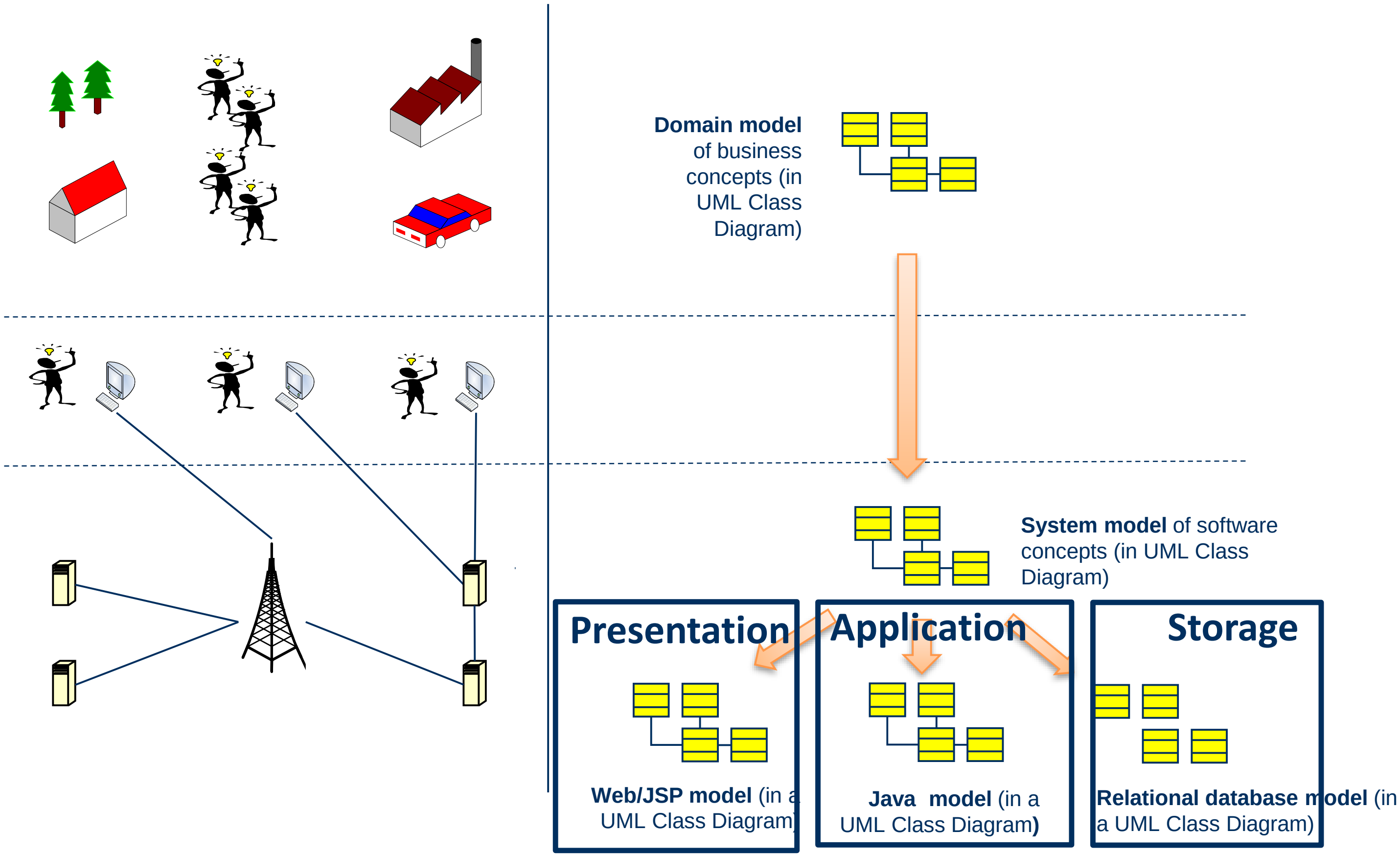
System model of software concepts (in UML Class Diagram)

Web/JSP model (in a UML Class Diagram**)**

Java model (in a UML Class Diagram**)**

Relational database model (in a UML Class Diagram)

Towards Model Driven Development



Domain model of business concepts (in UML Class Diagram)

System model of software concepts (in UML Class Diagram)

Presentation

Web/JSP model (in a UML Class Diagram)

Application

Java model (in a UML Class Diagram)

Storage

Relational database model (in a UML Class Diagram)