Relational logic design

Translation in relational model: entities and many to many relationships

DBMG

Translation in relational model

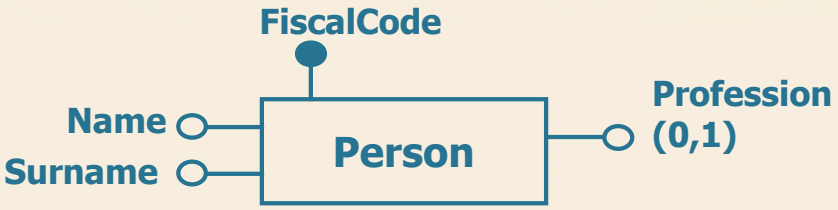
- It is executed on restructured ER schema
  - Without hierarchies, multivalued attributes and compounds
- Transformations
  - To each entity correspond a table with the same attributes
  - For what regards relations we need to consider the maximum cardinality

Relational logic design

Translation in relational model: entity

DBMG

Entity translation



Person(<u>FiscalCode</u>, Name, Surname, Profession*)

- Primary key underlined
- Optional attributes indicated by "*"

Relational logic design

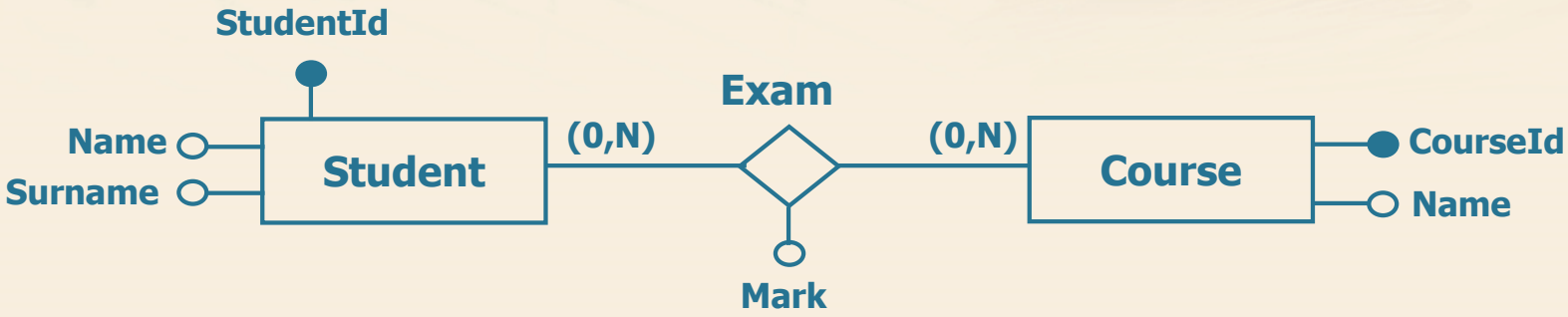
Translation in relational model: many to many relationship

DBMG

Many to many binary relationships

- Each many to many relationship corresponds a table
  - The primary key is the combination of identifiers of the two linked entities
  - It is possible to rename the attributes of the table that corresponds to the relationship (needed in case of recursive relations)

Many to many binary relationship



Student(StudentId, Name, Surname)

Course(CourseId, Name)

Exam(StudentId, CourseId, Mark)

Relational logic design

Translation in relational model: one to many relationship

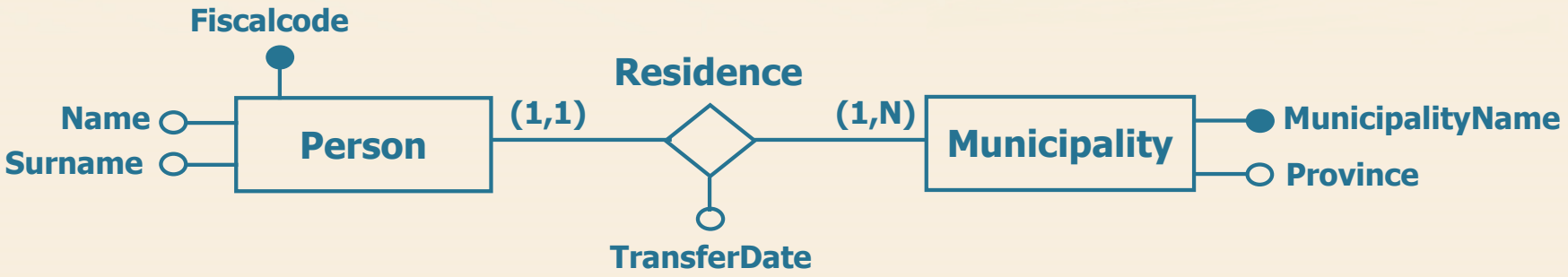
DBMG

One to many binary relationship

- There are two translation modes
  - Through attributes
  - Through a new table

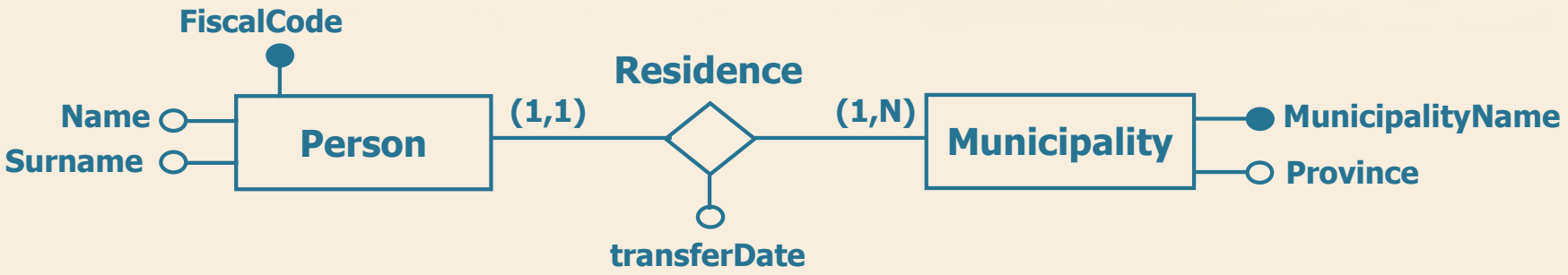
One to many binary relationship

- Mandatory participation from the “One” side



One to many binary relationship: entity

- Mandatory participation from the "One" side

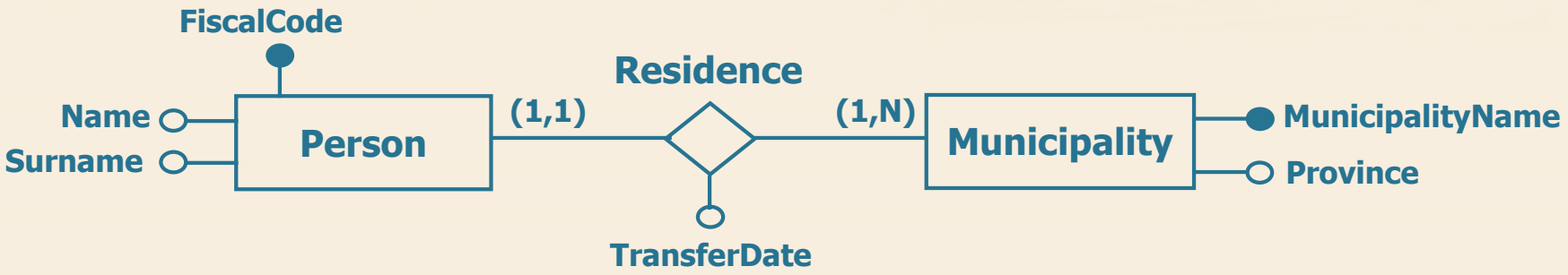


Person(FiscalCode, Name, Surname)

Municipality(MunicipalityName, Province)

One to many binary relationship: relationship

- Mandatory participation from the "One" side

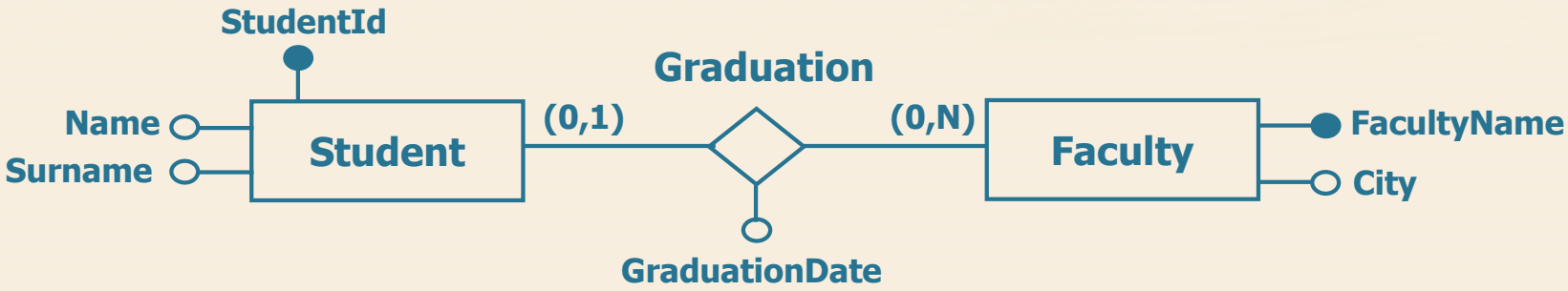


Persona(FiscalCode, Name, Surname, *MunicipalityName, TransferDate*)

Municipality(MunicipalityName, Province)

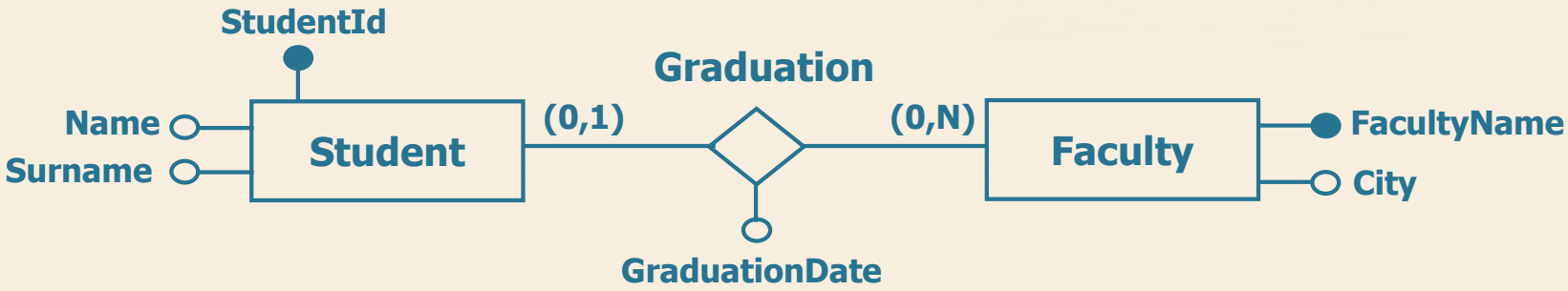
One to many binary relationship

- Optional participation from the "One" side



One to many binary relation: alternative n.1

- Optional participation from the "One" side



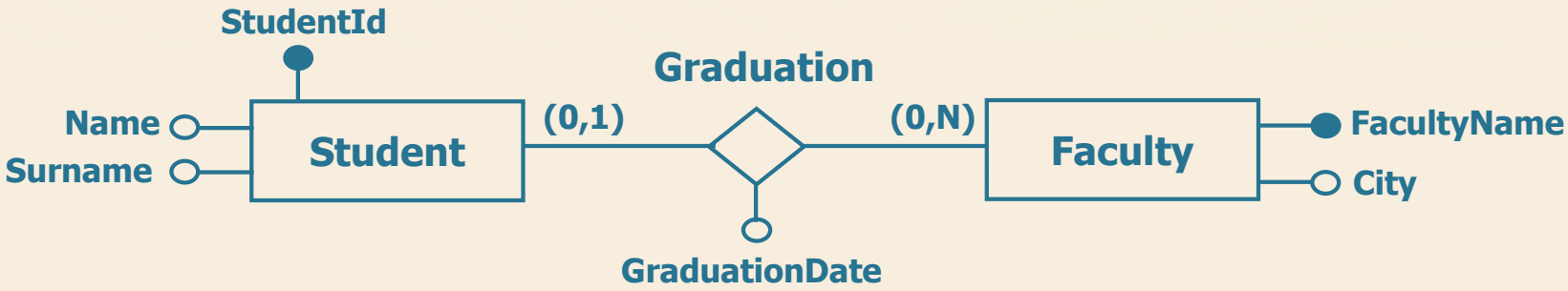
Student(StudentiId, Name, Surname)

Faculty(FacultyName, City)

Graduation(StudentId, FacultyName, GraduationDate)

One to many binary relation: alternative n.2

- Optional participation from the "One" side



Student(StudentId, Name, Surname, *FacultyName**, *GraduationDate**)

Faculty(FacultyName, City)

Relational logic design

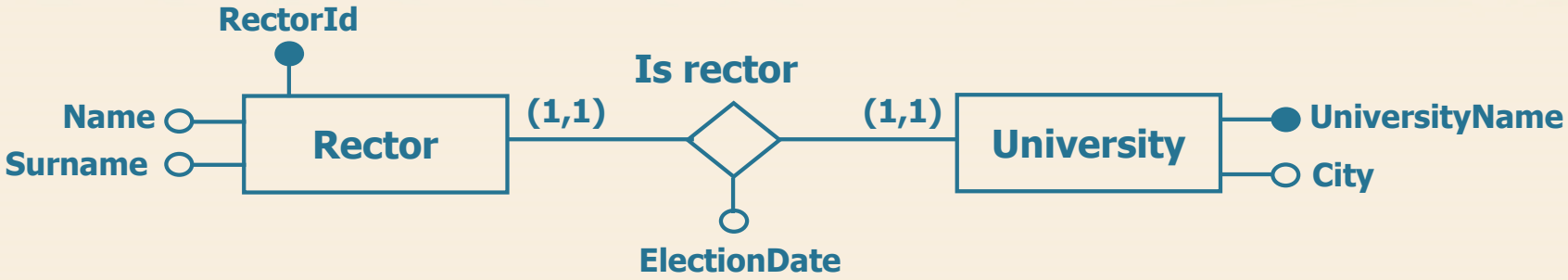
Translation in relational model: one to one relationship

One to one binary relationship

- There are more possible translations
  - It depends on minimum cardinality value

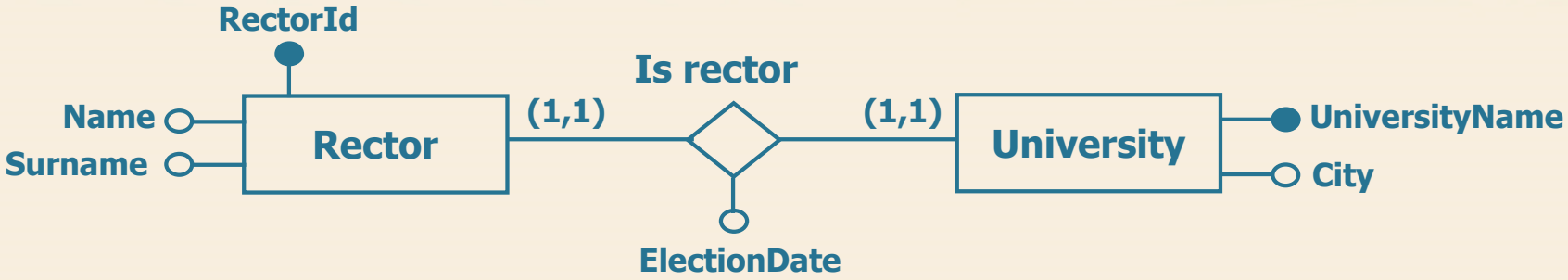
One to one binary relationship: case 1

- Mandatory participation from both sides



One to one binary relation: alternative n.1

- Mandatory participation from both sides

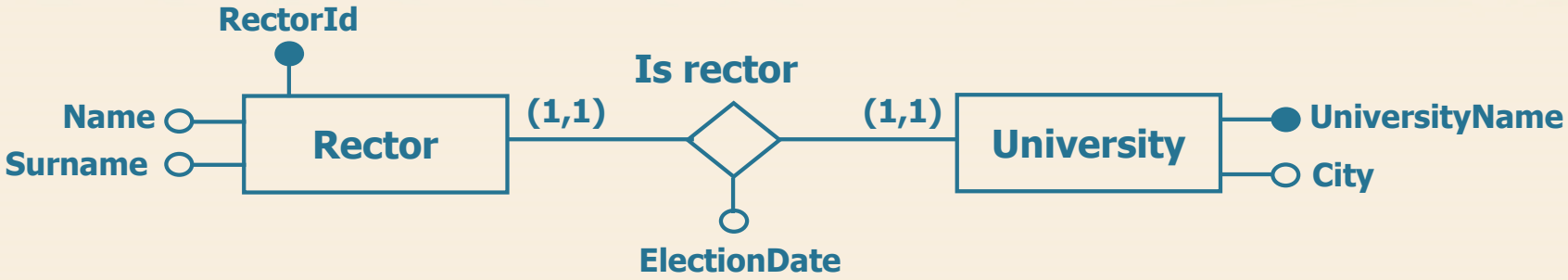


Rector(<u>RectorId</u>, Name, Surname)

University(<u>UniversityName</u>, City)

One to one binary relation: alternative n.1

- Mandatory participation from both sides

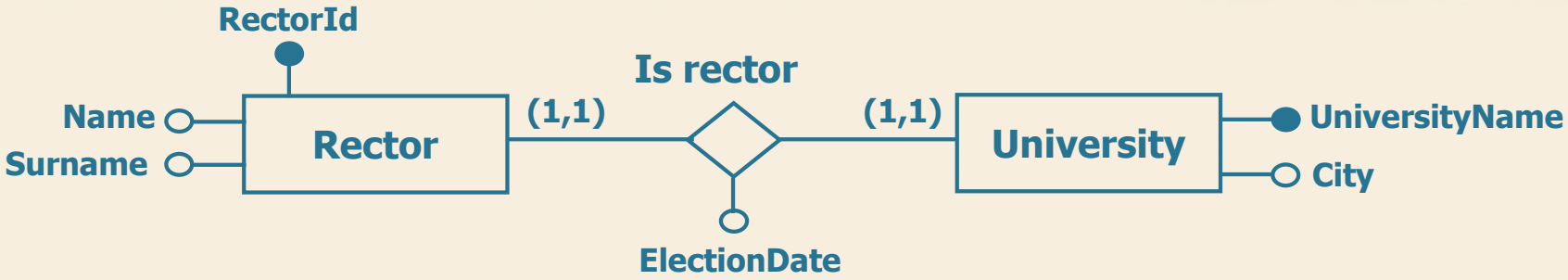


Rector(RectorId, Name, Surname, *UniversityName, ElectionDate*)

University(UniversityName, City)

One to one binary relation: alternative n.2

- Mandatory participation from both sides

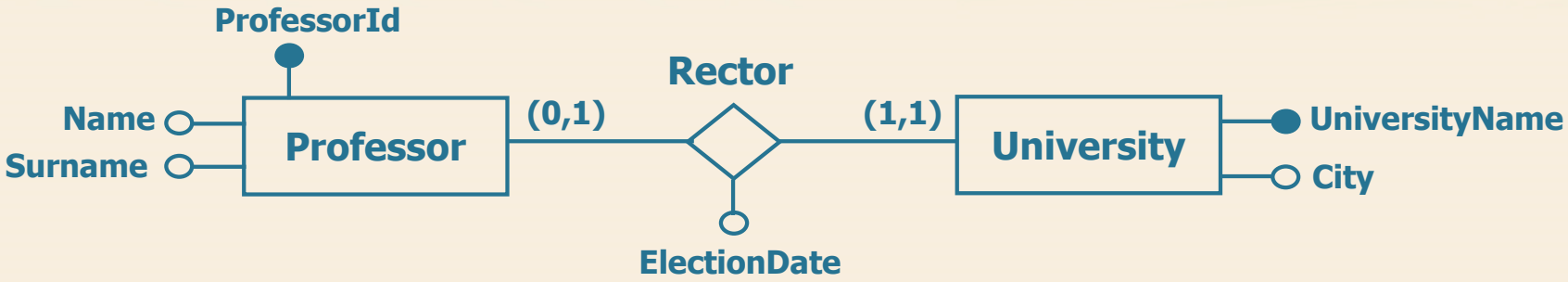


Rector(<u>RectorId</u>, Name, Surname)

University(<u>UniversityName</u>, City, *RectorId, ElectionDate*)

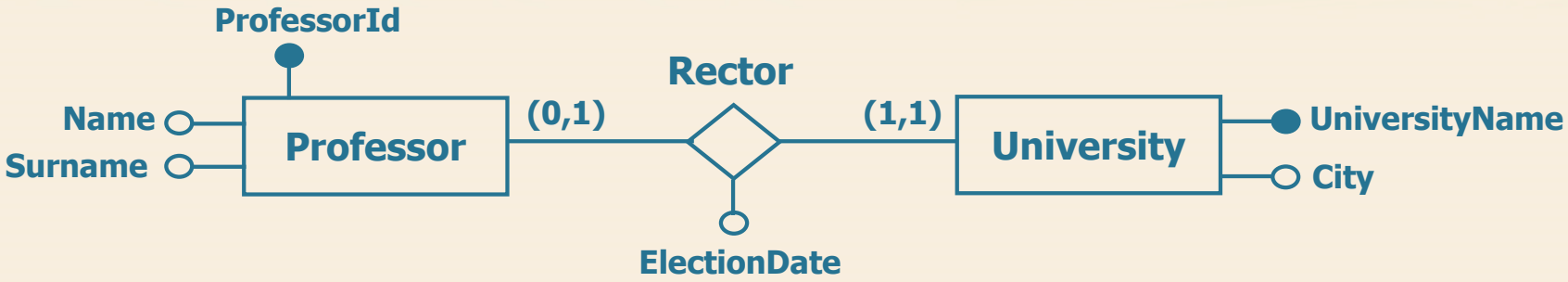
One to one binary relationship: case 2

- Optional participation from one side



One to one binary relationship: entity

- Optional participation from one side

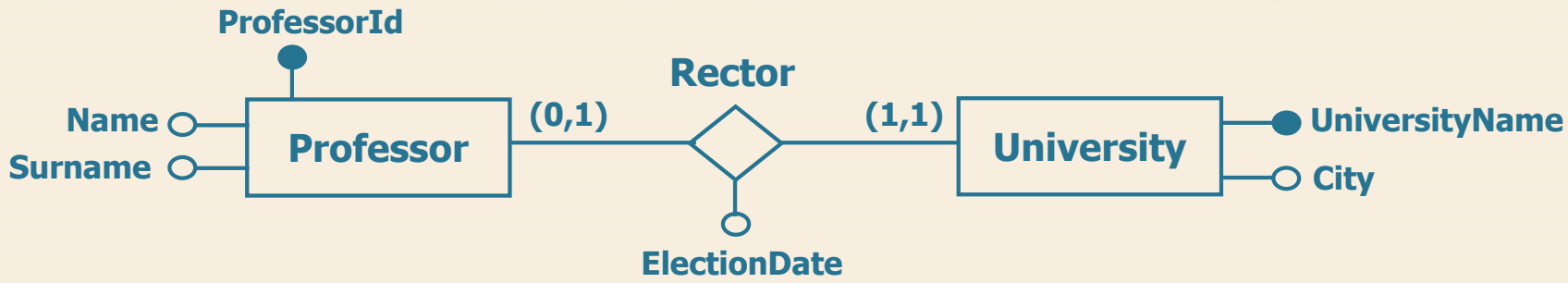


Professor(ProfessorId, Name, Surname)

University(UniversityName, City)

One to one binary relationship

- Optional participation from one side

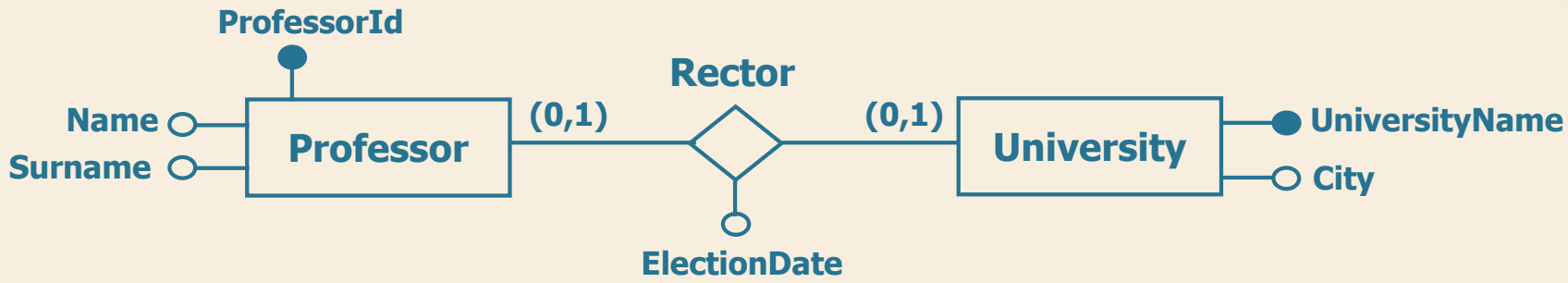


Professor(<u>ProfessorId</u>, Name, Surname)

University(<u>UniversityName</u>, City, *ProfessorId, ElectionDate*)

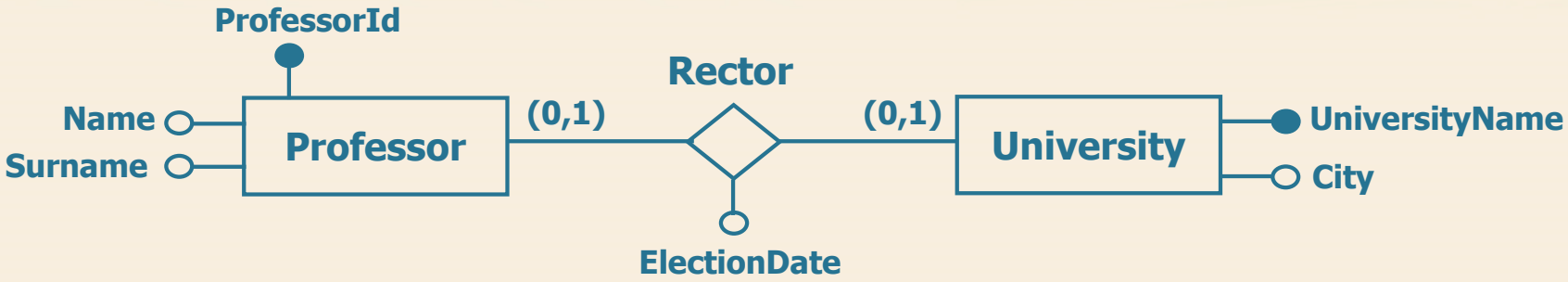
One to one binary relationship: case 3

- Optional participation from both sides



One to one binary relationship: alternative n.1

- Optional partecipation from both sides

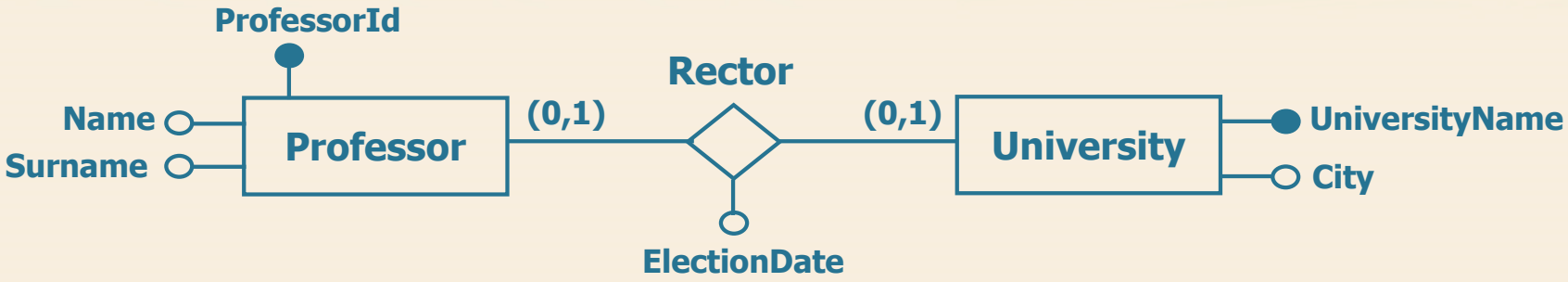


Professor(ProfessorId, Name, Surname)

University(UniversityName, City)

One to one binary relationship: alternative n.1

- Optional partecipation from both sides



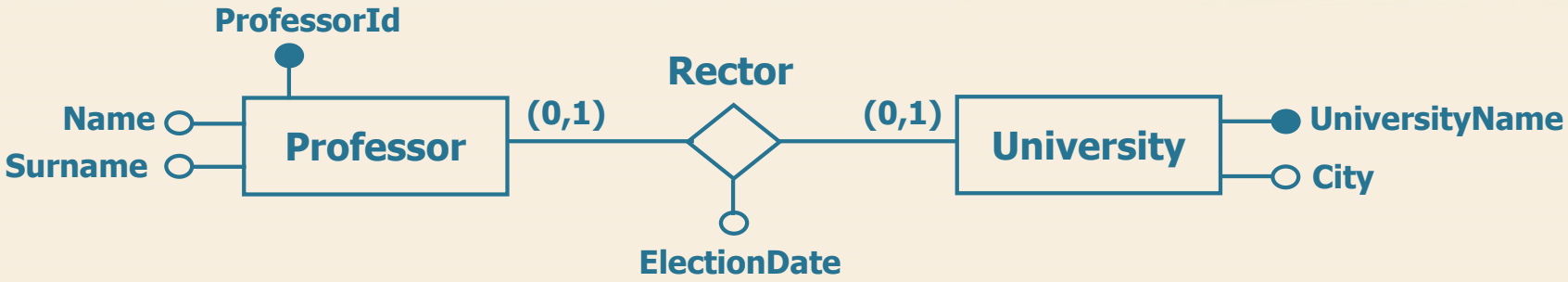
Professor(ProfessorId, Name, Surname)

University(UniversityName, City)

Rector(ProfessorId, UniversityName, ElectionDate)

One to one binary relationship:alternative n.2

- Optional participation from both sides



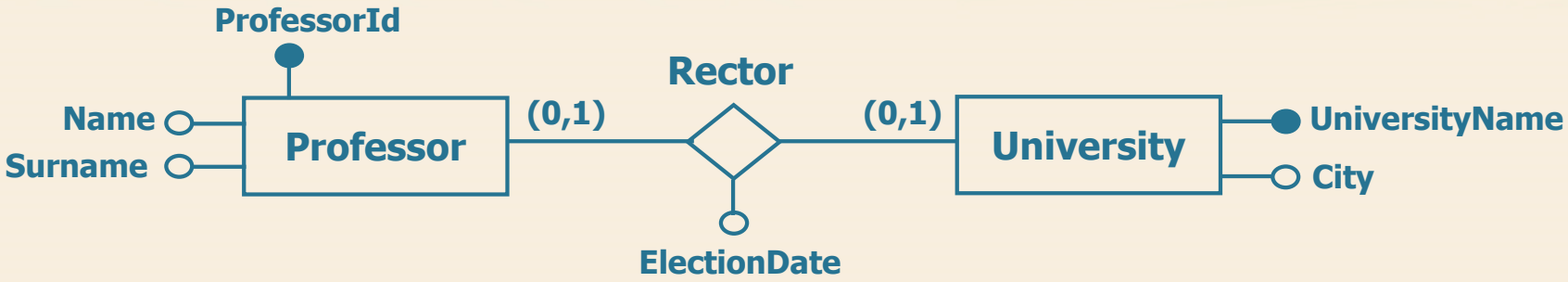
Professor(ProfessorId, Name, Surname)

University(UniversityName, City)

Rector(ProfessorId, UniversityName, ElectionDate)

One to one binary relationship:alternative n.3

- Optional participation from both sides

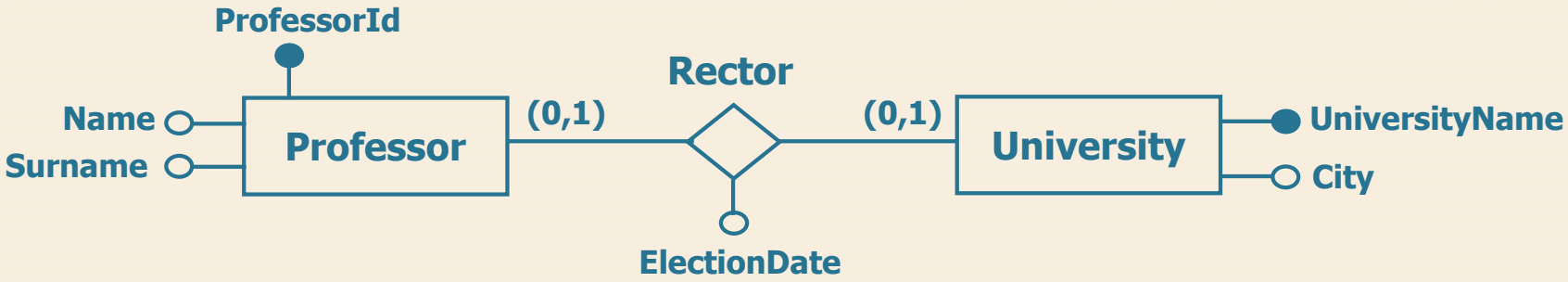


Professor(ProfessorId, Name, Surname)

University(UniversityName, City)

One to one binary relationship:alternative n.3

- Optional participation from both sides



Professor(<u>ProfessorId</u>, Name, Surname)

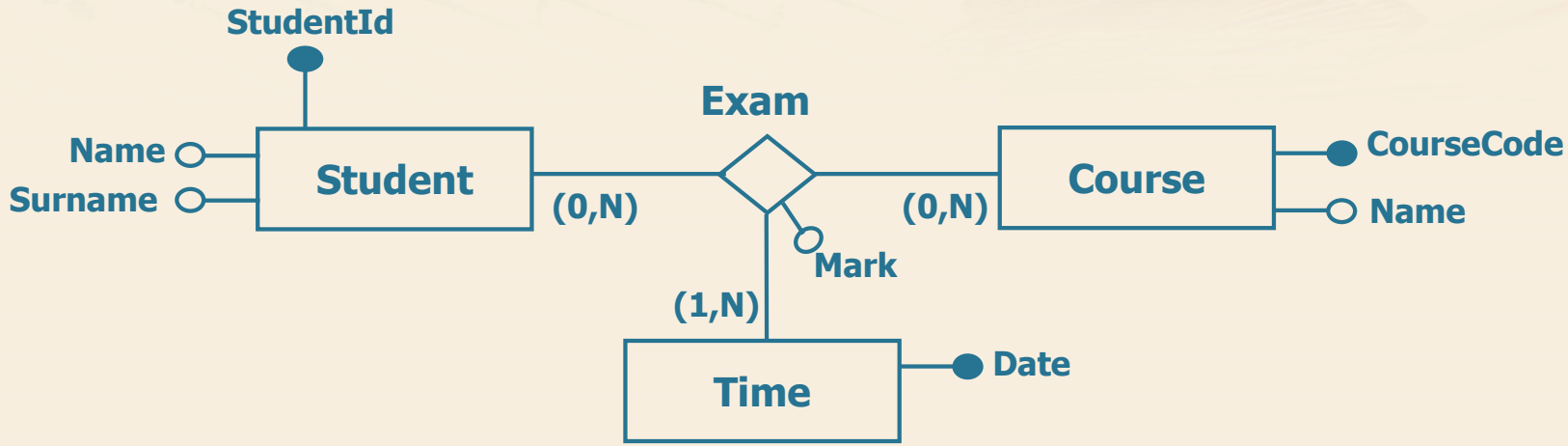
University(<u>UniversityName</u>, City, *ProfessorId** , *ElectionDate**)

Relational logic design

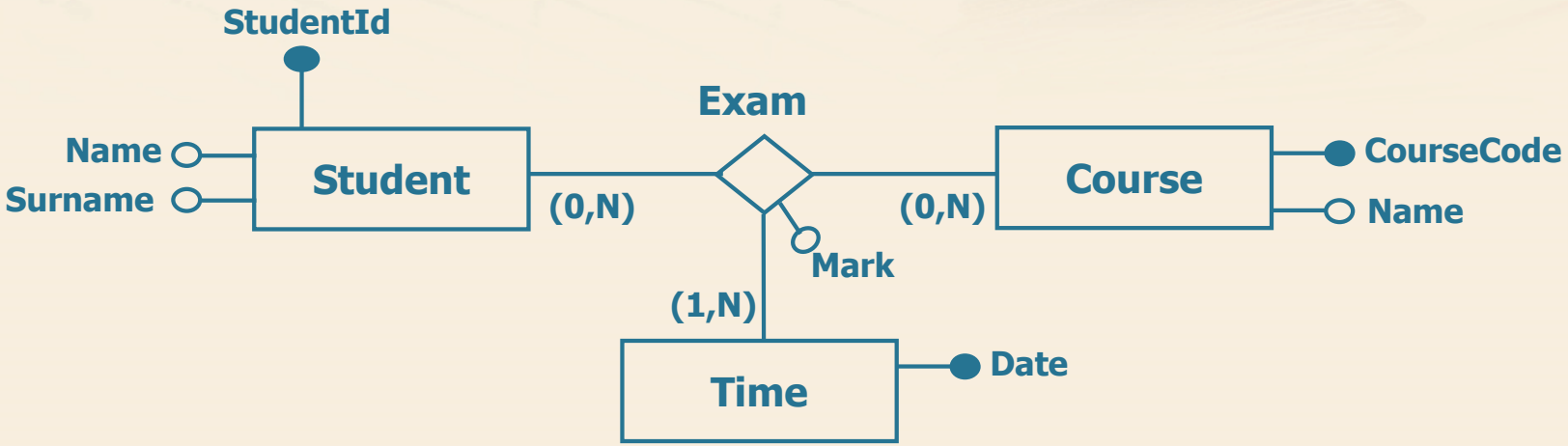
Translation in relational model: ternary relationship

DBMG

Ternary relationship

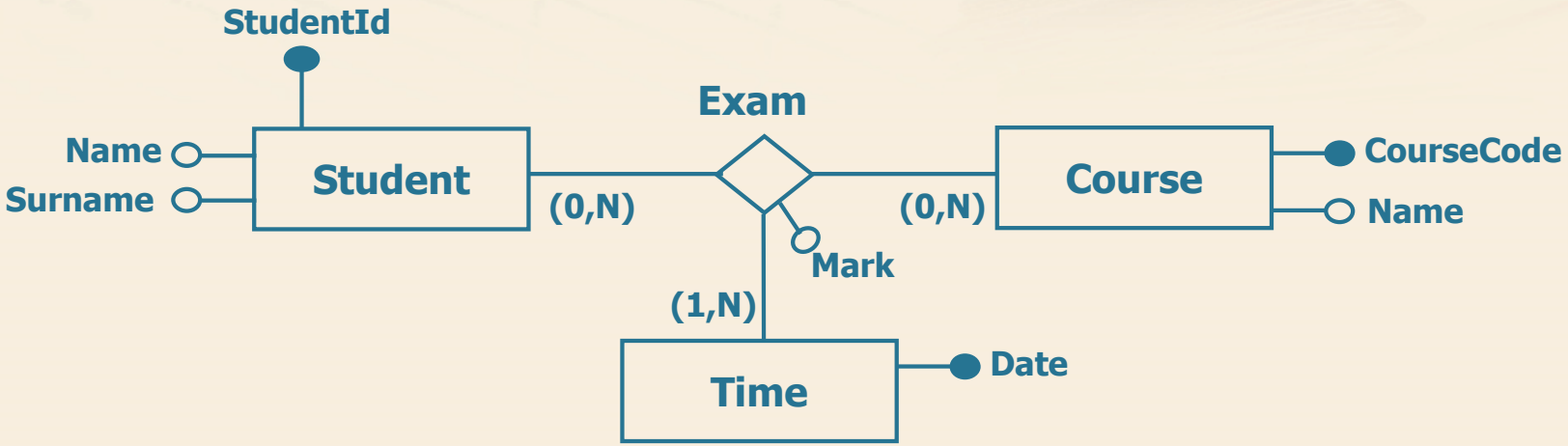


Ternary relationship: entity



Student(<u>StudentId</u>, Name, Surname)
Course(<u>CourseCode</u>, Name)
Time(<u>Date</u>)

Ternary relationship: relationship



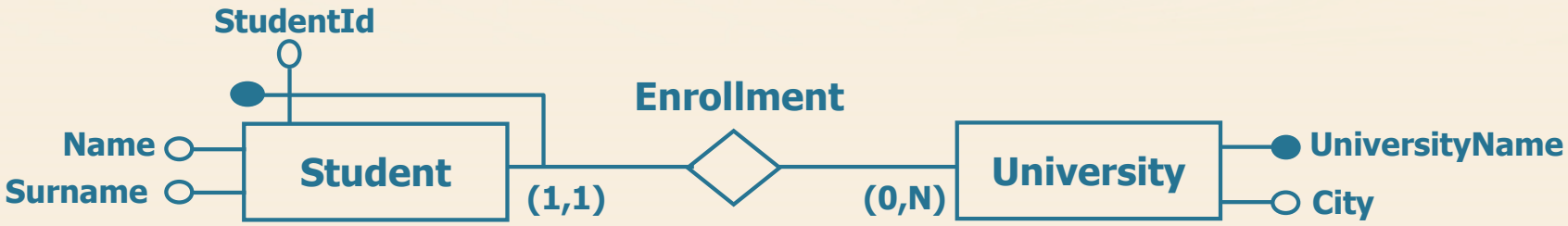
Student(<u>StudentId</u>, Name, Surname)
Course(<u>CourseCode</u>, Name)
Time(<u>Date</u>)
Exam(<u>StudentId</u>, <u>CourseCode</u>, <u>Date</u>, Mark)

Relational logic design

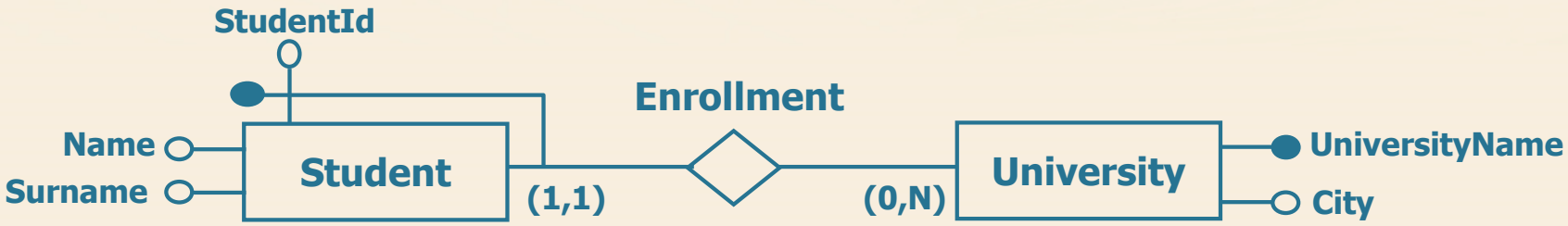
Translation in relational model: entity with external identificator

DBMG

Entity with external identificator



Entity with external identificator



University(<u>UniversityName</u>, City)
Student(<u>*StudentId*</u>, <u>*UniversityName*</u>, Name, Surname)

- The relation is represented together with the identifier

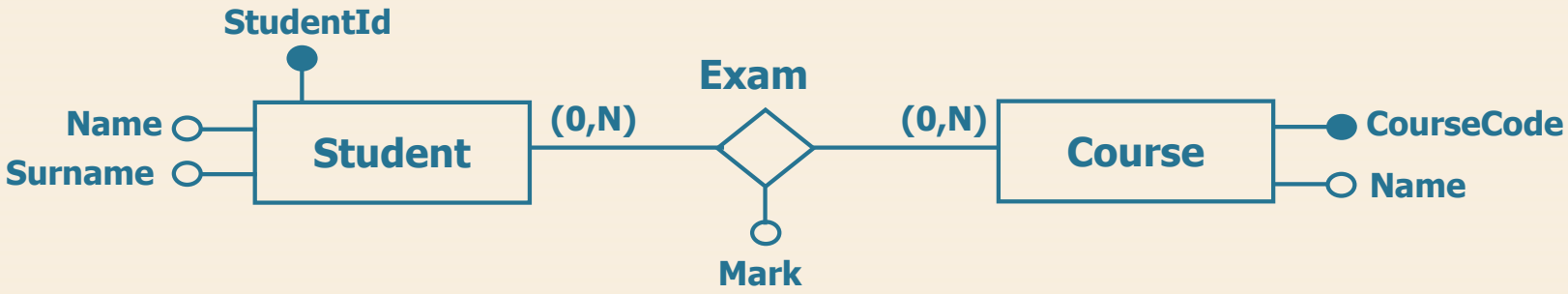
Relational logic design

Constraints of referential integrity

DBMG

Constraints of referential integrity

- Relationships represent referential constraints



Referential integrity: exam relationship

- Involved tables

Student(StudentId, Name, Surname)

Course(CourseCode, Name)

Exam(*StudentId*, CourseCode, Mark)

- Constraints of referential integrity

Exam(StudentId) REFERENCES Student(StudentId)

Referential integrity: exam relationship

- Involved tables

Student(StudentId, Name, Surname)

Course(CourseCode, Name)

Exam(StudentId, *CourseCode*, Mark)

- Constraints of referential integrity

Exam(StudentId) REFERENCES Student(StudentId)

Exam(CourseCode) REFERENCES Course(CourseCode)