Relational model

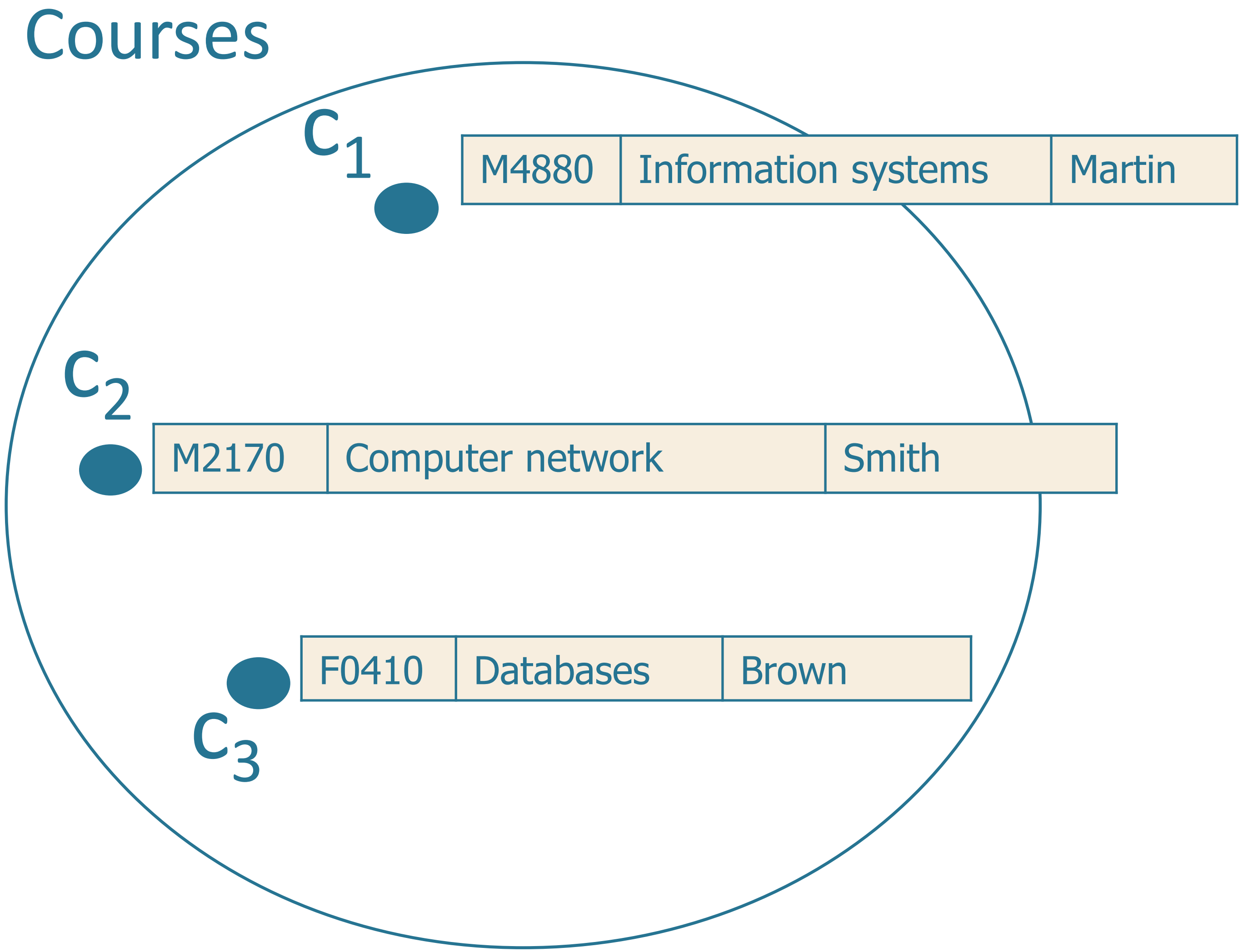
Relational model

- Introduction
- Definitions
- References between relations
- Incomplete information
- Integrity constraints
- Primary key
- Tuple constraint and domain constraint
- Referential integrity constraint

Introduction

Relational model

Intuition



Courses

c_1

M4880	Information systems	Martin

c_2

M2170	Computer network	Smith

c_3

F0410	Databases	Brown

Relational model

- Proposed by E. F. Codd in 1970 to support higher abstract levels compared to the previous models
  - data independence
- Made available in commercial DBMSs in 1981,
  - today it is the main model exploited in commercial DBMSs
- Based on the mathematical concept of *relation*
  - each relation is represented in an informal way by means of a table

Example

Courses

Code	Name	TeacherID
M2170	Information systems	D101
M4880	Computer Networks	D102
F0410	Databases	D321

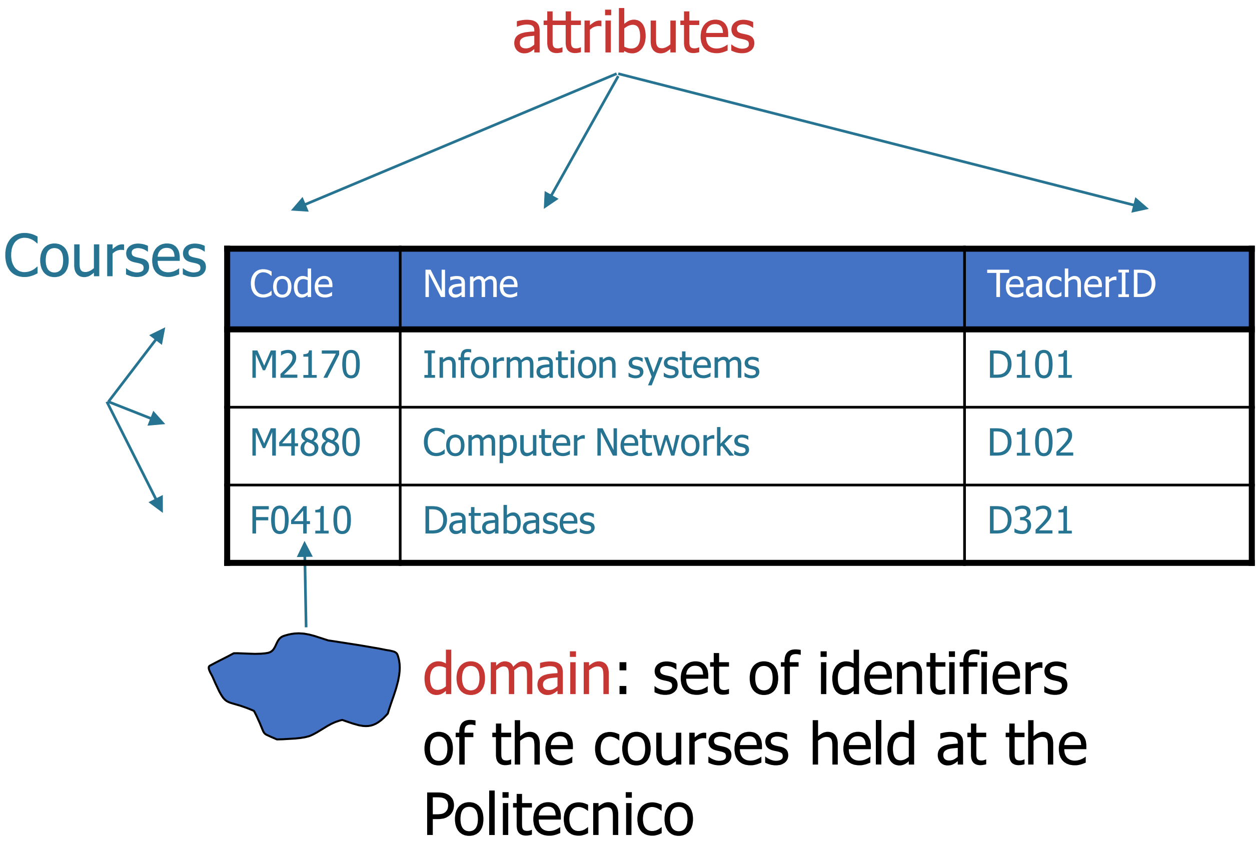
Teachers

ID	Name	Department	Phone#
D101	Green	Computer Engeneering	123456
D102	White	Telecommunications	636363
D321	Black	Computer Engeneering	414243

Definitions

Relational model

Definitions



attributes

Courses

Code	Name	TeacherID
M2170	Information systems	D101
M4880	Computer Networks	D102
F0410	Databases	D321

domain: set of identifiers of the courses held at the Politecnico

Attribute
- Name of a table column

Domain
- The set of values that can be assumed by an attribute

N-tuple (or tuple)
- Table row

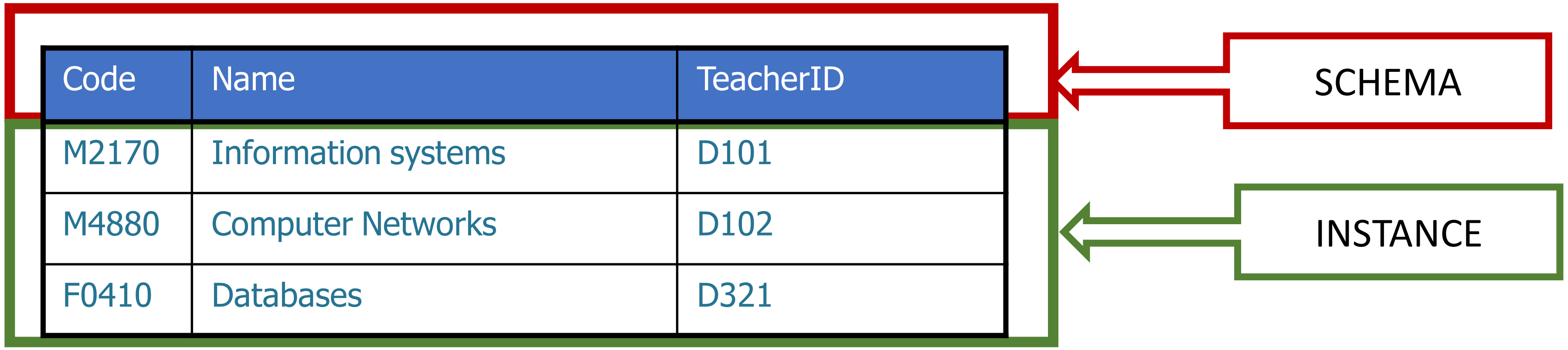
Cardinality
- Number of n-tuples in a relation

Degree
- Number of attributes in a relation

Definitions

- *Schema:* describes the structure of the data
  - virtually invariant over time
  - is represented by the header of each table (table name and column names)
- *Instance:* consists of the contents of each table, i.e., the actual values of the data
  - variable over time, even very quickly
  - is represented by the rows of the tables

Courses



Code	Name	TeacherID
M2170	Information systems	D101
M4880	Computer Networks	D102
F0410	Databases	D321

SCHEMA

INSTANCE

Properties

- Tuples (rows) *are not* ordered
- Tuples are *distinct* among them (there are no duplicated rows)
- Attributes are not ordered
  - it is not possible to identify an attribute by means of its position

References between relations

Relational model

References between relations

- The relational model is *value-based*
  - references between data in different relations are represented by means of values of the domains
- Advantages:
  - independent of the physical structure
  - only information that is relevant from the application point of view is stored
  - data can be easily transferred across different systems
  - differently from pointers, the link is not oriented

Value-Based Reference: Example

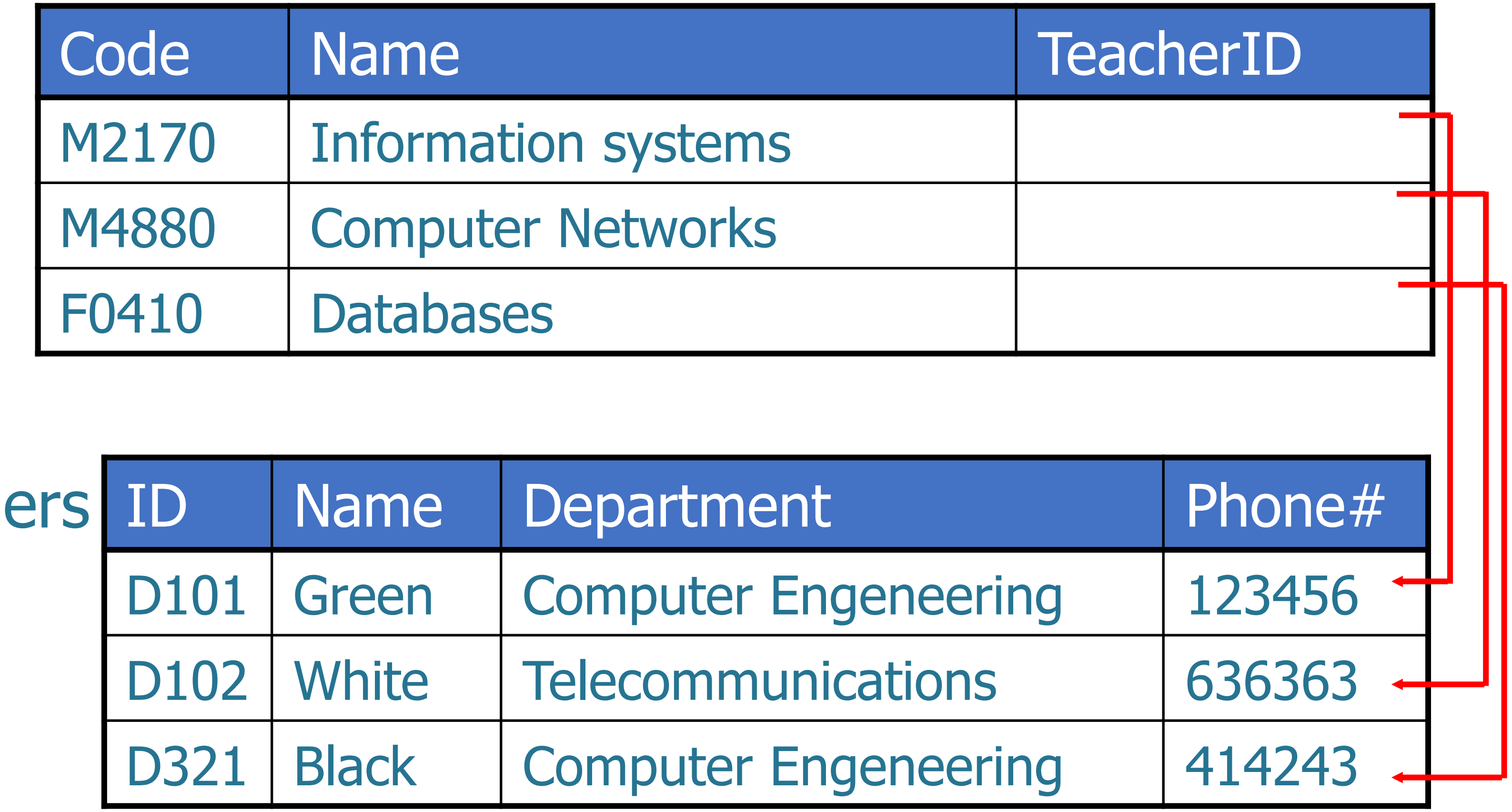
Courses

Code	Name	TeacherID
M2170	Information systems	D101
M4880	Computer Networks	D102
F0410	Databases	D321

Teachers

ID	Name	Department	Phone#
D101	Green	Computer Engeneering	123456
D102	White	Telecommunications	636363
D321	Black	Computer Engeneering	414243

Pointer-based reference: Example



Courses

Code	Name	TeacherID
M2170	Information systems	
M4880	Computer Networks	
F0410	Databases	

Teachers

ID	Name	Department	Phone#
D101	Green	Computer Engeneering	123456
D102	White	Telecommunications	636363
D321	Black	Computer Engeneering	414243

Null values

Relational model

Incomplete information

- Some information could be not available for all the tuples in the relation
- Example:

- **Student (StudentID, Surname, BirthDate, Phone#, DegreeYear)**
  - *the phone number could be (temporarily?) unknown*
  - *for students who have not yet graduated, year of degree is not defined*

Null values

- To represent lack of information we could use a special value belonging to the domain (0, empty string, 999, …)
    - an “unused” value is required (example: DegreeYear=0, Phone#=?)
    - “unused” values could become meaningful (Phone#= 999999)
    - it is necessary to deal separately with “special” values in different applications
- This representation is not adequate

Null value

- Special value named *null value* (NULL)
  - it is not a value of the domain
  - it denotes both the absence of a domain value and a value not defined
  - it must be used with caution (example: StudentID=NULL?)
- Notation: attributes that can have a null value (NULL) are often highlighted with superscript * in the relation schema

ID	Name	Department	Phone#*
D101	Green	Computer Engeneering	123456
D102	White	Telecommunications	NULL
D321	Black	Computer Engeneering	414243

Primary keys

Relational model

Unique identification for tuples

Students

StudentID	Name	Surname	BirthDate	EnrollementYear
64655	Mike	Red	4/8/1978	1998
81999	Paul	White	4/8/1978	1999
75222	Marco	Red	8/3/1979	1998

- There are no pairs of students with the same value for the StudentID
  - the StudentID uniquely identifies students
- There are no pairs of students with the same values for personal data
  - name, surname and birth date uniquely identify students

Key

- A *key* is an attribute set that uniquely identifies tuples in a relation
  - it is a property of the relational schema
- Formal definition: a set K of attributes is a key in a relation r if
  - the relation r does not contain a pair of distinct tuples with the same values for K (uniqueness)
  - K is minimal (there exists no subset K’ of K that is still unique)

Example

- The attribute

  {StudentID}

  is unique and minimal, thus it is a key

- The attribute set

  {Name, Surname, BirthDate}

  is unique and minimal (none of its subsets is unique), thus it is a key

Superkey

- A set K of attributes is a key in a relation r if
  - The relation r does not contain a pair of distinct tuples with the same values for K (uniqueness)
  - K is minimal (there are not proper subsets of K still unique)
- If only the first property is satisfied, K is a *superkey* of r

Examples

- The attribute set

  {StudentID,Name}

  is unique, but not minimal (the StudentID is unique), thus the attribute set is a superkey, but it is *not* a key

- The attribute set

  {BirthDate, EnrollementYear}

  is unique and minimal: is it a general property?

Primary key

- If a key can assume the NULL value, it cannot be a key (the uniqueness property is lost)
    - it is mandatory to avoid the NULL values in the keys
- Solution:
    - A reference key, which does not allow null values, is defined and set as *primary key*
    - The other keys (candidate keys) can assume null values
    - References between data in different relations are defined by means of the primary key

Primary key

- Notation: The attributes that make up the primary key are often highlighted by underlining in the relation schema

ID	Name	Department	Phone#*
D101	Green	Computer Engeneering	123456
D102	White	Telecommunications	NULL
D321	Black	Computer Engeneering	414243

Integrity constraints

Relational model

Integrity constraints

- Integrity constraint: property that must be satisfied by all correct instances of the database
- Types of constraints:
  - Intra-relational constraints
    - defined on the attributes of a single relation (e.g.: uniqueness constraints, domain constraints, and tuple constraints)
  - Inter-relational constraints
    - defined on multiple relation at the same time (e.g.: referential integrity constraints)

Uniqueness Constraints: Example

Courses

Code	Name	TeacherID
M2170	Information systems	D101
~~F0410~~	Computer Networks	D102
~~F0410~~	Databases	D321

Teachers

ID	Name	Department	Phone#
D101	Green	Computer Engineering	123456
D102	White	Telecommunications	636363
D321	Black	Computer Engineering	414243

Domain Constraints

- Domain Constraint
  - expresses conditions on the value assumed by a single attribute of a tuple
    - it can be a Boolean expression (and, or, not) of simple predicates
  - example: Grade > 0 and Grade 30

Exam

CourseID	StudentID	Grade
M2170	S1234	23
M4880	S4321	28
F0410	s4321	~~40~~

Tuple constraint

- Tuple constraint
  - expresses conditions on the values of each tuple, indipendently of other tuples
    - it can correlate many attributes
    - it can be a Boolean expression (and, or, not) of simple predicates (e.g., comparison between attributes, between an attribute and a constant)
  - examples :
    - *Price = Cost + TaxPerc*Cost*
    - *CumLaude = True* if *Grade = 30*

Exam

CourseID	StudentID	Grade	CumLaude
M2170	S1234	23	False
M4880	S4321	30	True
F0410	s4321	~~26~~	~~True~~

Referential Integrity Constraint: Example

- Information in different relations is related through common values of one or more attributes
  - the TeacherID attribute in the Courses relation (referencing relation) refers to the StudentID attribute in the Teacher relation (referenced relation)
  - the values assumed by the TeacherID attribute in the Courses relation can only be values taken by the ID attribute, the primary key of the Teacher relation
  - the TeacherID attribute in the Courses relation is the foreign key of Courses

Foreign key

Courses (referencing relation)

Code	Name	TeacherID
M2170	Information systems	D101
M4880	Computer Networks	D102
F0410	Databases	D321

Teacher (referenced relation)

ID	Name	Department	Phone#
D101	Green	Computer Engineering	123456
D102	White	Telecommunications	636363
D321	Black	Computer Engineering	414243

Referential integrity constraint

- Given two relations:
  - R (referenced/external relation)
  - S, that refers R through a set X of attributes (referencing/internal relation)

  values on a set X of attributes in a relation S can be exclusively values for the primary key of the relation R
- The set X of attributes in S represents its foreign key
- Referential integrity constraints are imposed in order to guarantee that the values in the referencing relation refer to actual values in the referenced relation (the relational model is value-based)

Referential Integrity Constraint: Example

Teacher
(referenced relation)

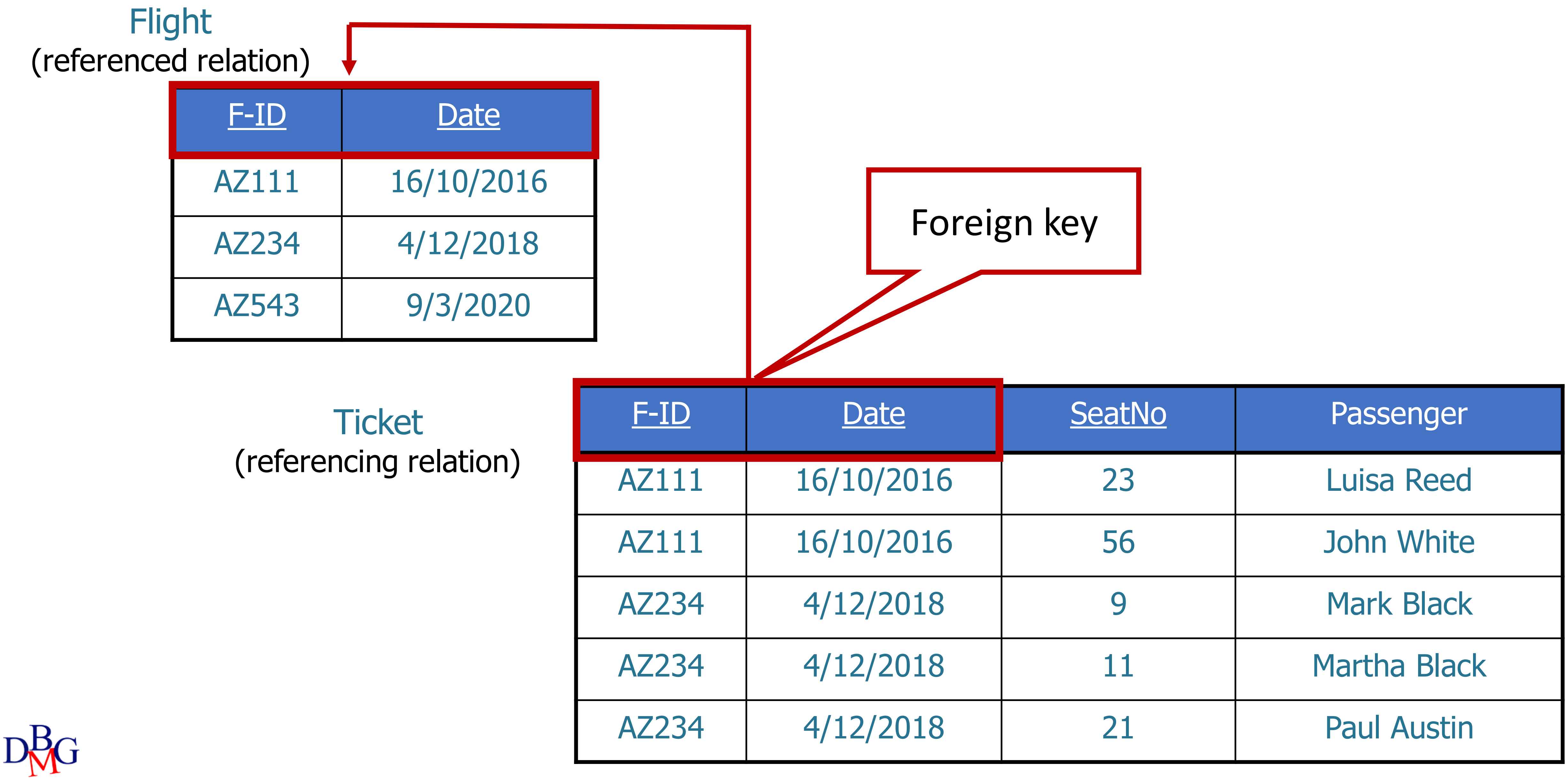
ID	Name	Department	Phone#
D101	Green	Computer Engineering	123456
D102	White	Telecommunications	636363
D321	Black	Computer Engineering	414243

Foreign key

Courses (referencing relation)

Code	Name	TeacherID
M2170	Information systems	D101
M4880	Computer Networks	D102
F0410	Databases	D101

Referential Integrity Constraint: Example



Flight
(referenced relation)

F-ID	Date
AZ111	16/10/2016
AZ234	4/12/2018
AZ543	9/3/2020

Foreign key

Ticket
(referencing relation)

F-ID	Date	SeatNo	Passenger
AZ111	16/10/2016	23	Luisa Reed
AZ111	16/10/2016	56	John White
AZ234	4/12/2018	9	Mark Black
AZ234	4/12/2018	11	Martha Black
AZ234	4/12/2018	21	Paul Austin

Referential Integrity Constraint: Example

Flight
(referenced relation)

F-ID	Date
AZ111	16/10/2016
AZ234	4/12/2018
AZ543	9/3/2020

Foreign key

Ticket
(referencing relation)

F-ID	Date	SeatNo	Persona
AZ111	16/10/2016	23	Luisa Reed
AZ111	~~16/11/2016~~	56	John White
AZ234	4/12/2018	9	Mark Black
AZ234	4/12/2018	11	Martha Black
AZ234	4/12/2018	21	Paul Austin

Referential Integrity Constraint: Example

Flight
(referenced relation)

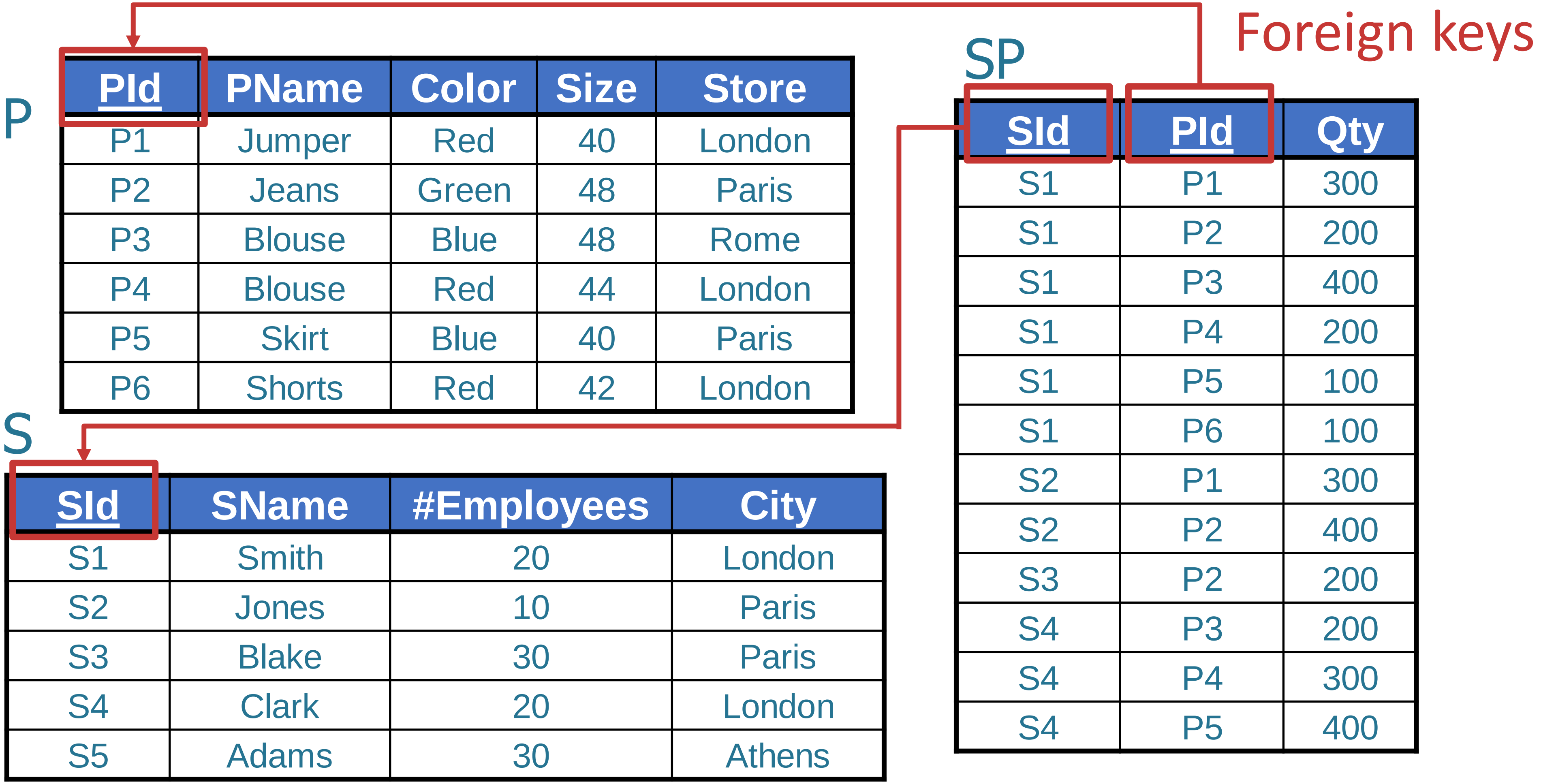
F-ID	Date
AZ111	16/10/2016
AZ234	4/12/2018
AZ543	9/3/2020

Foreign key

Ticket
(referencing relation)

F-ID	Date	SeatNo	Passenger
AZ111	16/10/2016	23	Luisa Reed
AZ111	16/10/2016	56	John White
AZ234	4/12/2018	9	Mark Black
AZ234	4/12/2018	11	Martha Black
~~AZ543~~	~~4/12/2018~~	21	Paul Austin

Referential Integrity Constraint: Example



Foreign keys

P

PId	PName	Color	Size	Store
P1	Jumper	Red	40	London
P2	Jeans	Green	48	Paris
P3	Blouse	Blue	48	Rome
P4	Blouse	Red	44	London
P5	Skirt	Blue	40	Paris
P6	Shorts	Red	42	London

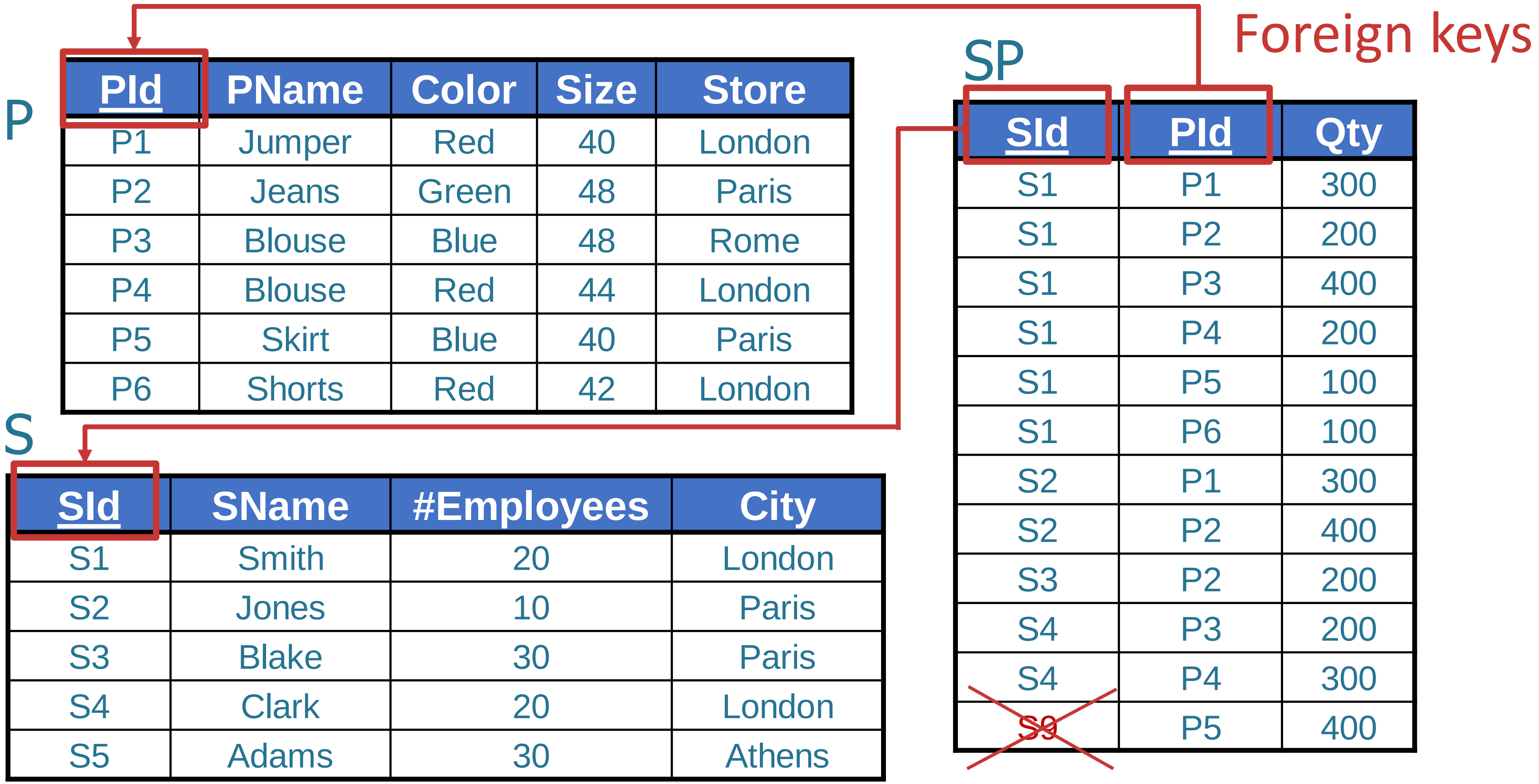
S

SId	SName	#Employees	City
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

SP

SId	PId	Qty
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P3	200
S4	P4	300
S4	P5	400

Referential Integrity Constraint: Example



Foreign keys

P

PId	PName	Color	Size	Store
P1	Jumper	Red	40	London
P2	Jeans	Green	48	Paris
P3	Blouse	Blue	48	Rome
P4	Blouse	Red	44	London
P5	Skirt	Blue	40	Paris
P6	Shorts	Red	42	London

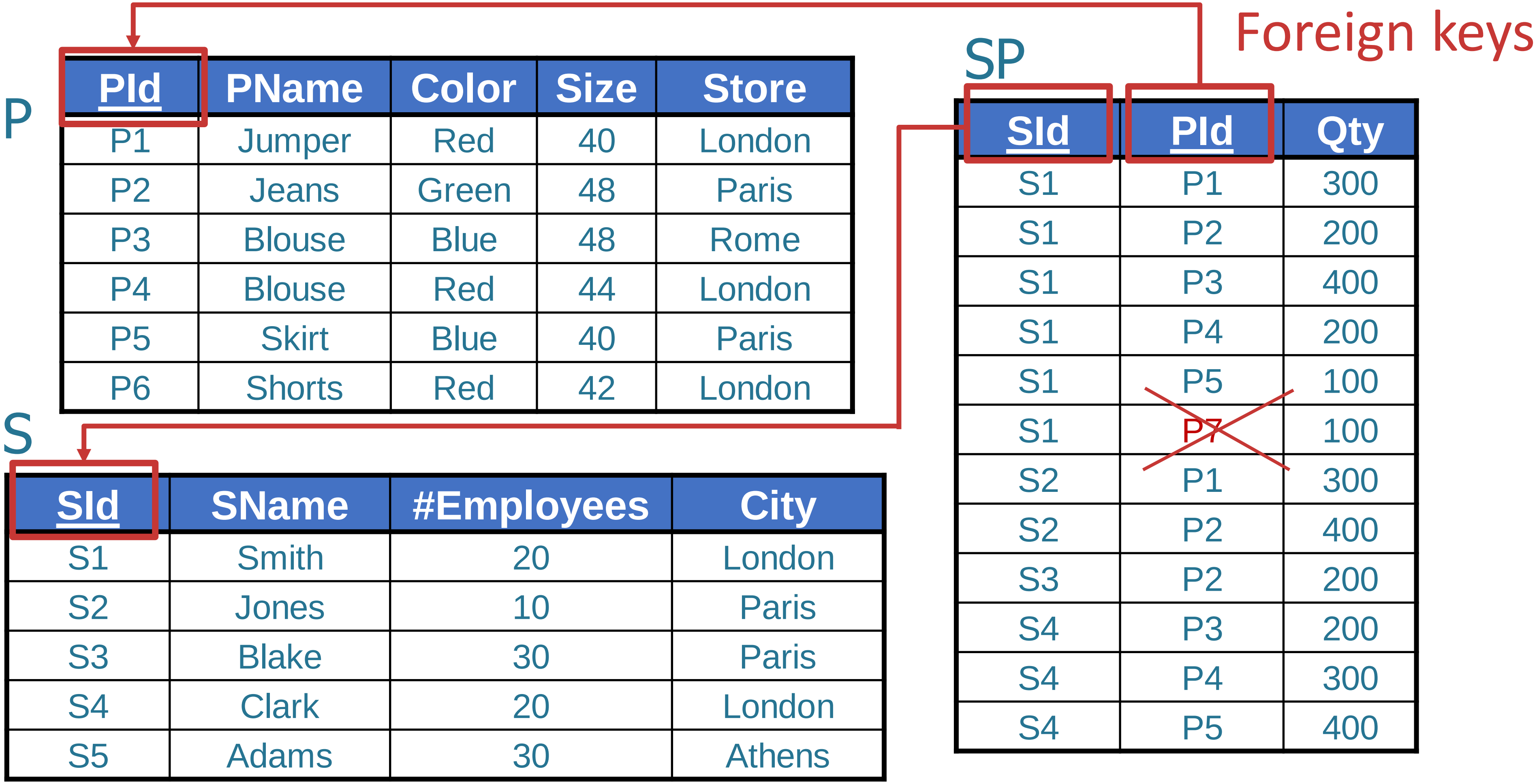
S

SId	SName	#Employees	City
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

SP

SId	PId	Qty
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P3	200
S4	P4	300
~~S9~~	P5	400

Referential Integrity Constraint: Example



Foreign keys

P

PId	PName	Color	Size	Store
P1	Jumper	Red	40	London
P2	Jeans	Green	48	Paris
P3	Blouse	Blue	48	Rome
P4	Blouse	Red	44	London
P5	Skirt	Blue	40	Paris
P6	Shorts	Red	42	London

S

SId	SName	#Employees	City
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

SP

SId	PId	Qty
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	~~P7~~	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P3	200
S4	P4	300
S4	P5	400