Refresher: ER-modeling, logical relational model, dependencies

Toon Calders
toon.calders@ulb.ac.be

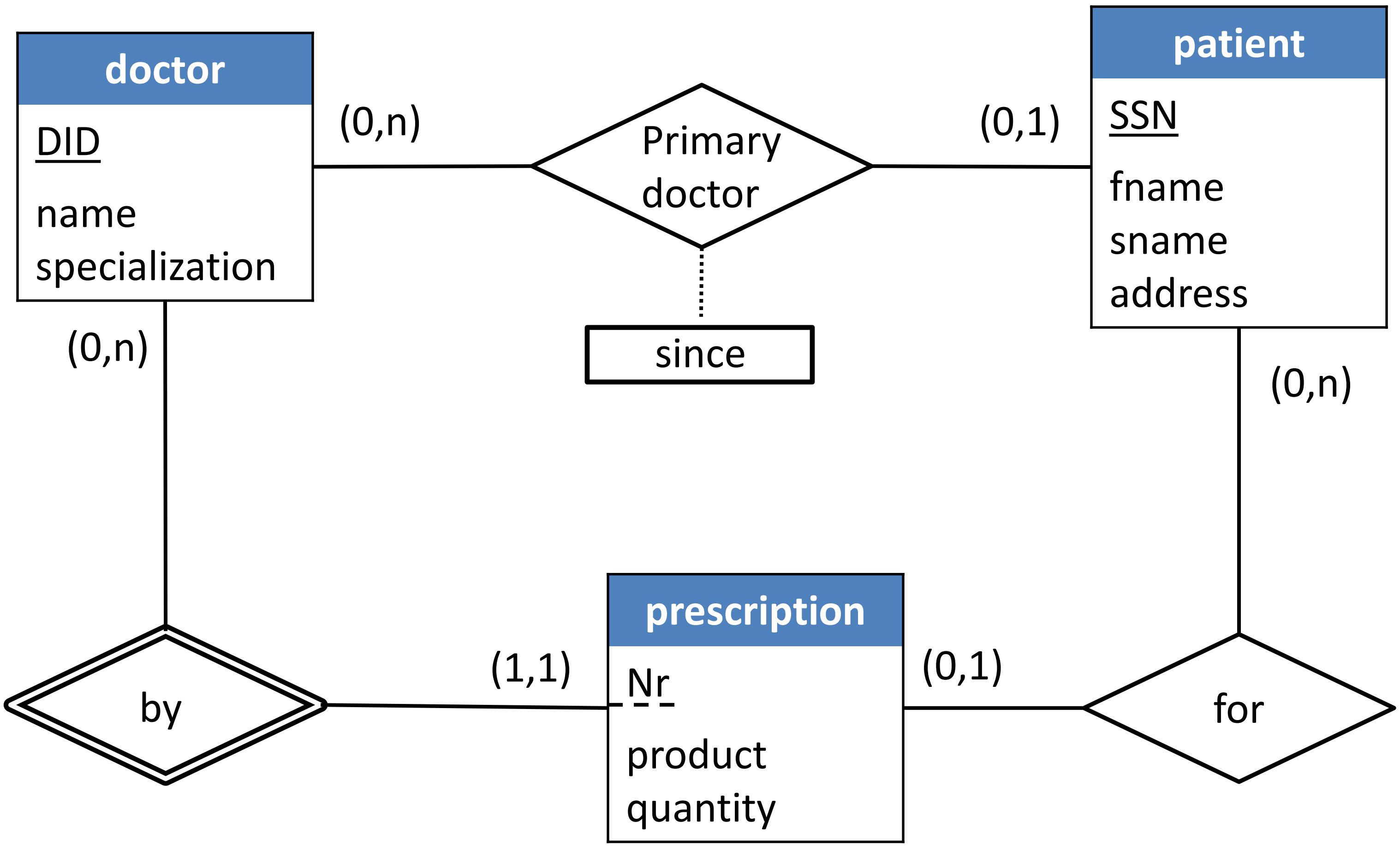
ULB
ECOLE POLYTECHNIQUE DE BRUXELLES

Different Levels

- Conceptual level: ER-diagrams
- Logical level: Relations, attributes, schemas, primary keys, foreign key dependencies
- Physical level: Storage model, partitions, indices, triggers, …

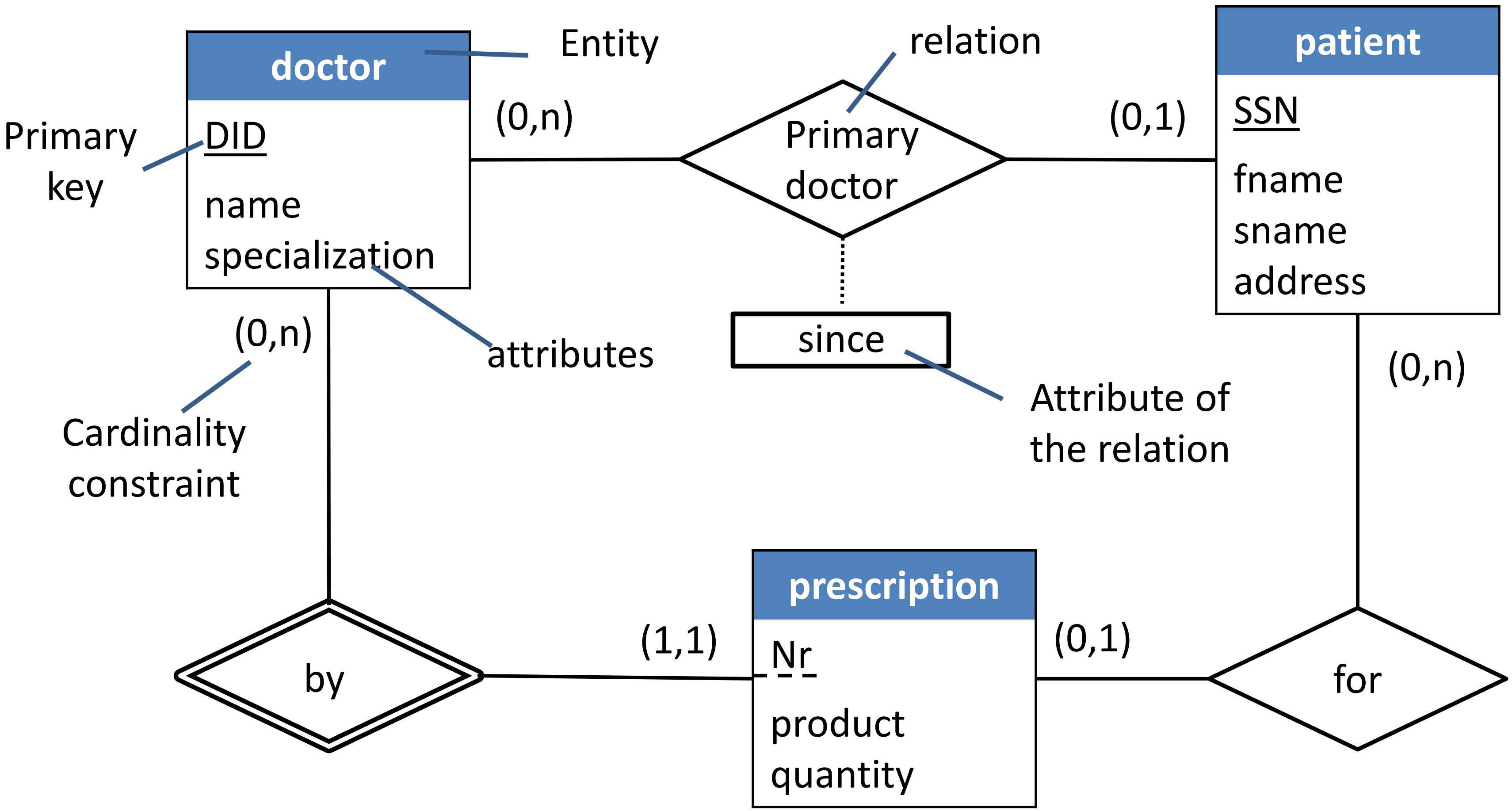
Conceptual Level: ER-Diagram

- Expresses entities and relations between them



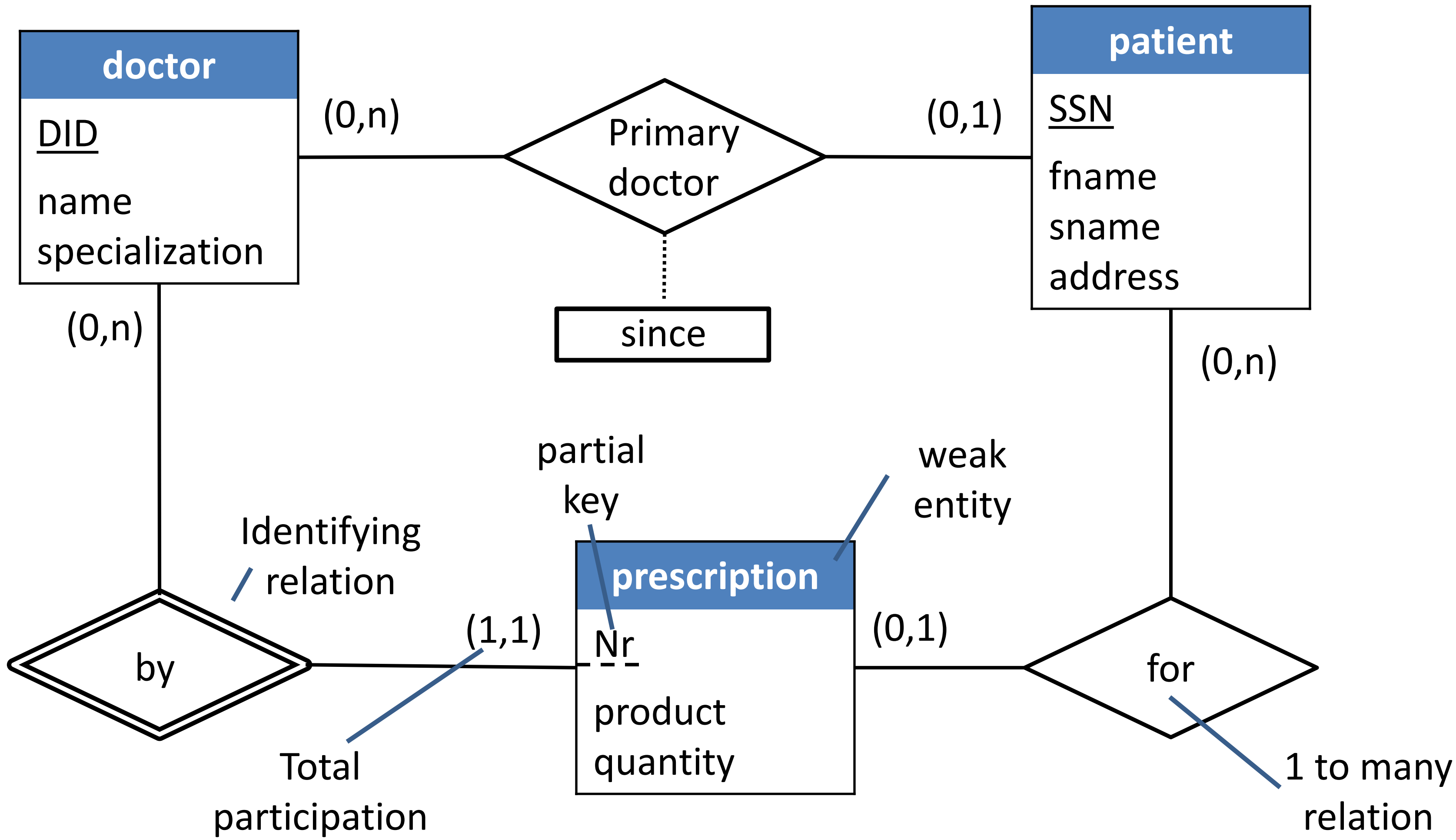
Conceptual Level: ER-Diagram

- Expresses entities and relations between them

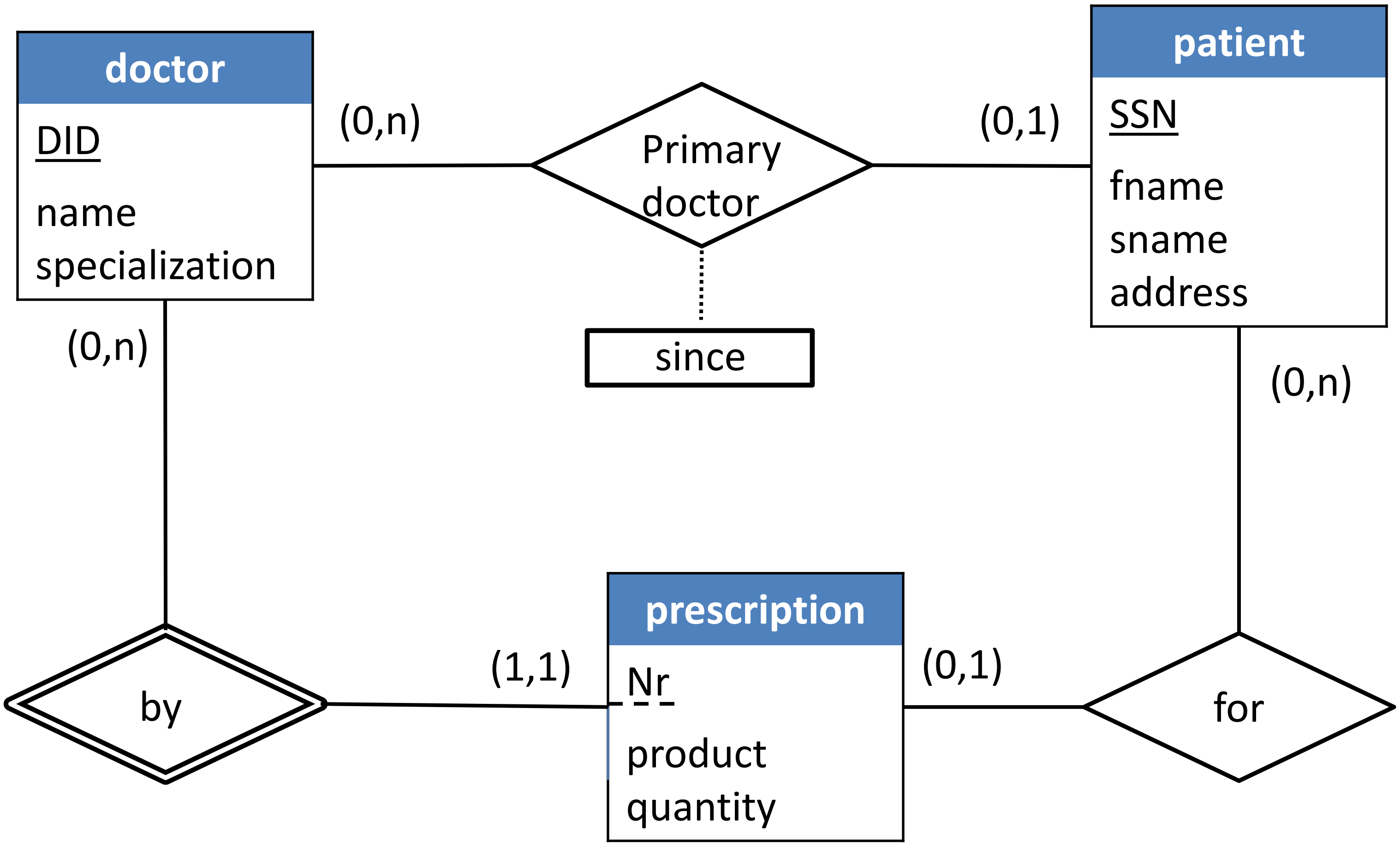


Conceptual Level: ER-Diagram

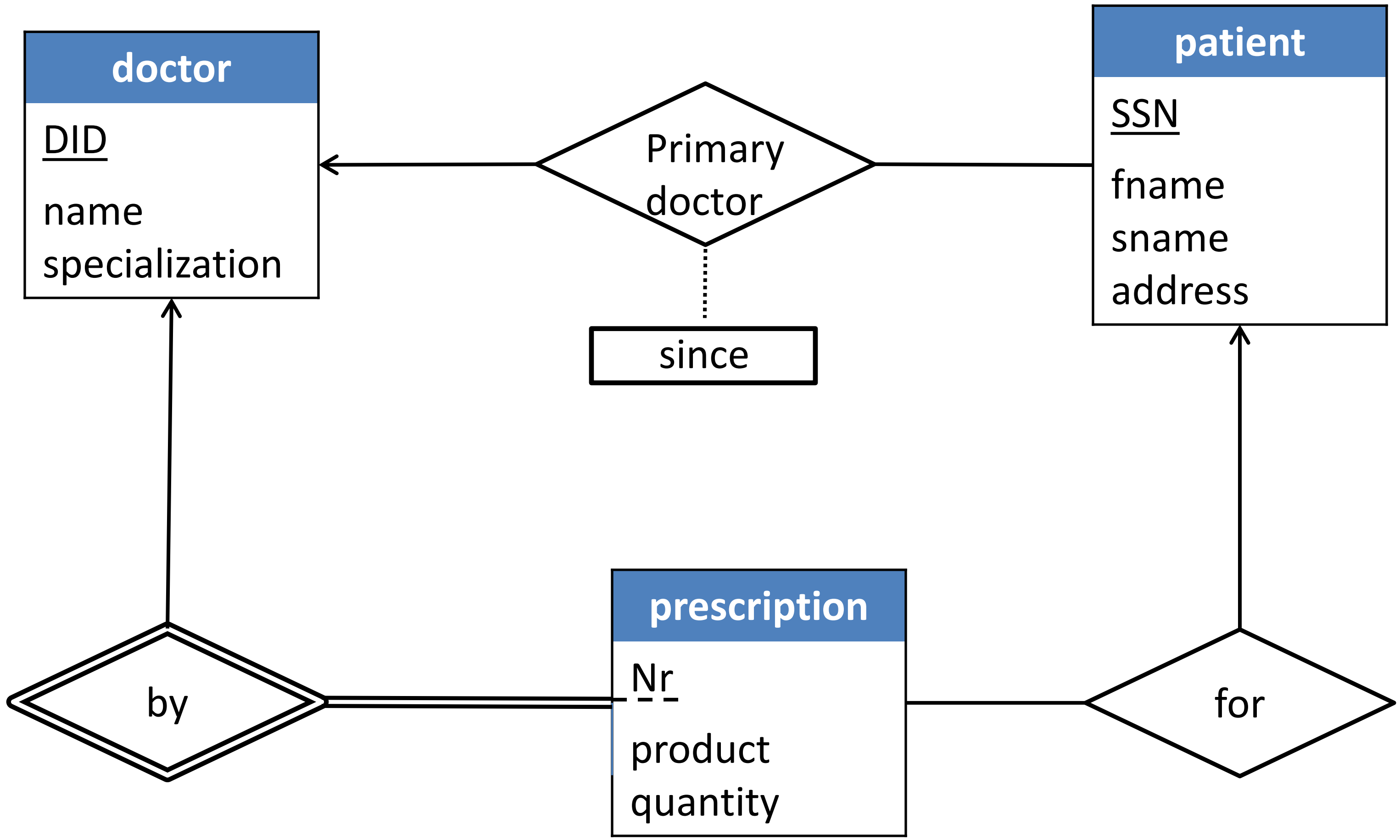
- Expresses entities and relations between them



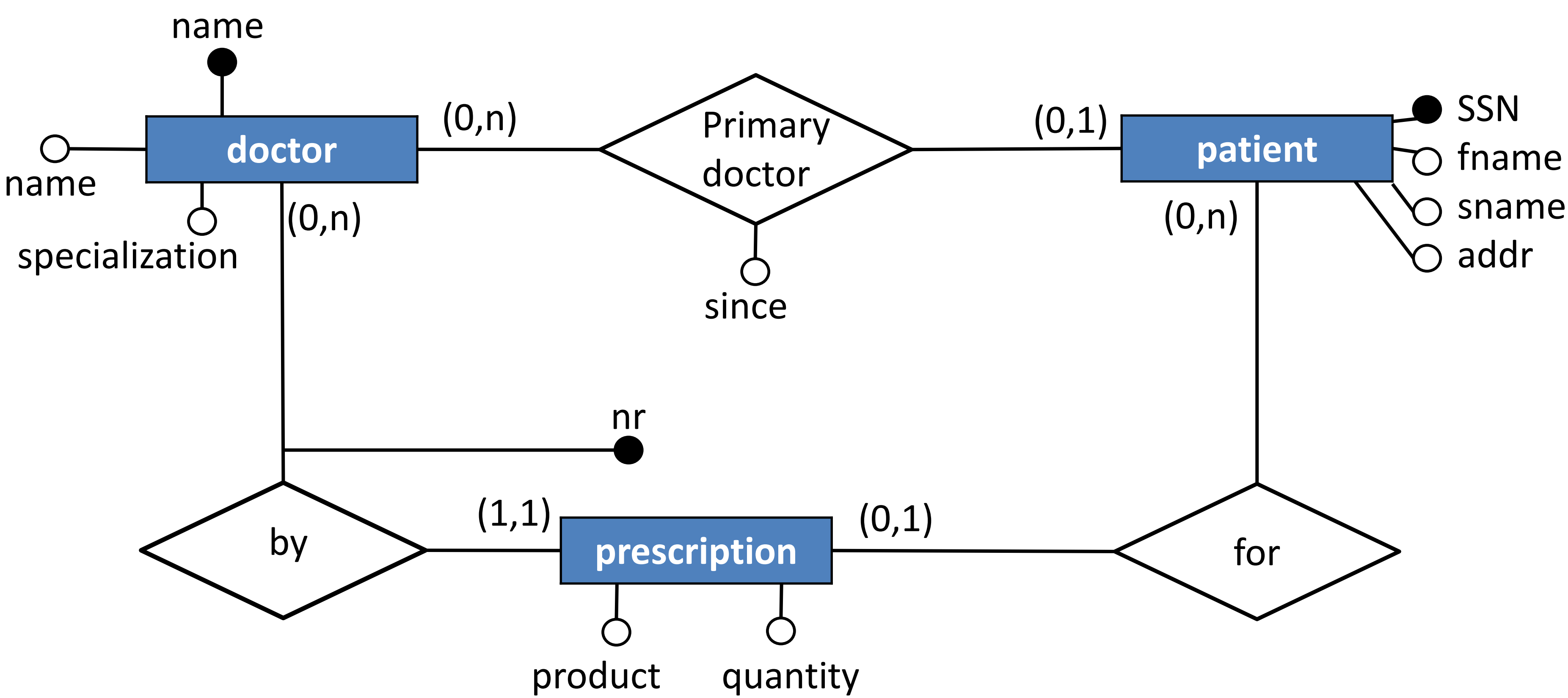
Different Notations



Different Notations



Different Notations



Exercise: ER Modeling

Construct an E-R diagram for a car insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents. Each insurance policy covers one or more cars, and has one or more premium payments associated with it. Each payment is for a particular period of time, and has associated due date, and the date the payment was received.

Logical Model: Relational

- A schema is a set of attributes
- Domain of attribute A: dom(A)
- Tuple t over schema S: mapping from S to values; for all $A \in S$, $t(A) \in dom(A)$
- Relation R(S): (finite) set of tuples over S
- Database D is a set of relations

Relational Database: Example

Courses

Code	Name
2ID45	Advanced Databases
2ID05	Databases I

Offerings

Code	Semester	Lecturer
2ID45	Spring 2011	Calders
2ID05	Spring 2011	Fletcher

Follows

Student	Code	Semester	Grade
Phil	2ID45	Spring 2011	A+
Mary	2ID45	Spring 2011	C
John	2ID45	Spring 2011	B-
Paul	2ID05	Spring 2011	C

Relational Model: Keys

- Set of attributes $K \subseteq S$ is a *superkey* for relation $R(S)$ if: for every legal instance r of R, and for every $t_1, t_2 \in r$, $t_1(K)=t_2(K)$, then $t_1 = t_2$
- K is a *candidate key* if K is a superkey and K is minimal (no strict subset is a key)
- In the logical model, we chose one of the candidate keys as the *primary key*.

Keys: Example

Student	Code	Name	Semester	Lecturer	Grade
Phil	2ID45	Advanced Databases	Spring 2011	Calders	A+
Mary	2ID45	Advanced Databases	Spring 2011	Calders	C
John	2ID45	Advanced Databases	Spring 2011	Calders	B-
Paul	2ID05	Databases I	Spring 2011	Fletcher	C

- Some superkeys:
  - {Student, Code, Name, Semester}
  - {Student, Code, Semester, Lecturer}
- Only candidate Key: {Student, Code, Semester}

Functional Dependencies

- $X, Y \subseteq S$ Functional dependency $X \rightarrow Y$ holds in relation $R(S)$ if: for every legal instance r of R, and for every $t_1, t_2 \in r$, if $t_1(X)=t_2(X)$, then $t_1(Y)=t_2(Y)$

- "$X \rightarrow Y$ holds" is equivalent to "if we project the relation on XY, X is a superkey."

Example: Functional Dependencies

Student	Code	Name	Semester	Lecturer	Grade
Phil	2ID45	Advanced Databases	Spring 2011	Calders	A+
Mary	2ID45	Advanced Databases	Spring 2011	Calders	C
John	2ID45	Advanced Databases	Spring 2011	Calders	B-
Paul	2ID05	Databases I	Spring 2011	Fletcher	C

- Code → Name
- Code, Semester → Lecturer
- Student, Code, Semester → Grade

Foreign Key Dependencies

(Also called “inclusion dependencies”)

- Two relations R(S) and T(U);
  T has primary key PK.
- $F \subseteq S$ is a foreign key into T if:
  for every legal instance r of R and t of T,
  $\Pi_F\, r \subseteq \Pi_{PK}\, t$

(Alternative: for every $u \in r$, there exists a $v \in t$ such that u(F)=v(PK))

Foreign Keys: Example

Courses

Code	Name
2ID45	Advanced Databases
2ID05	Databases I

Follows

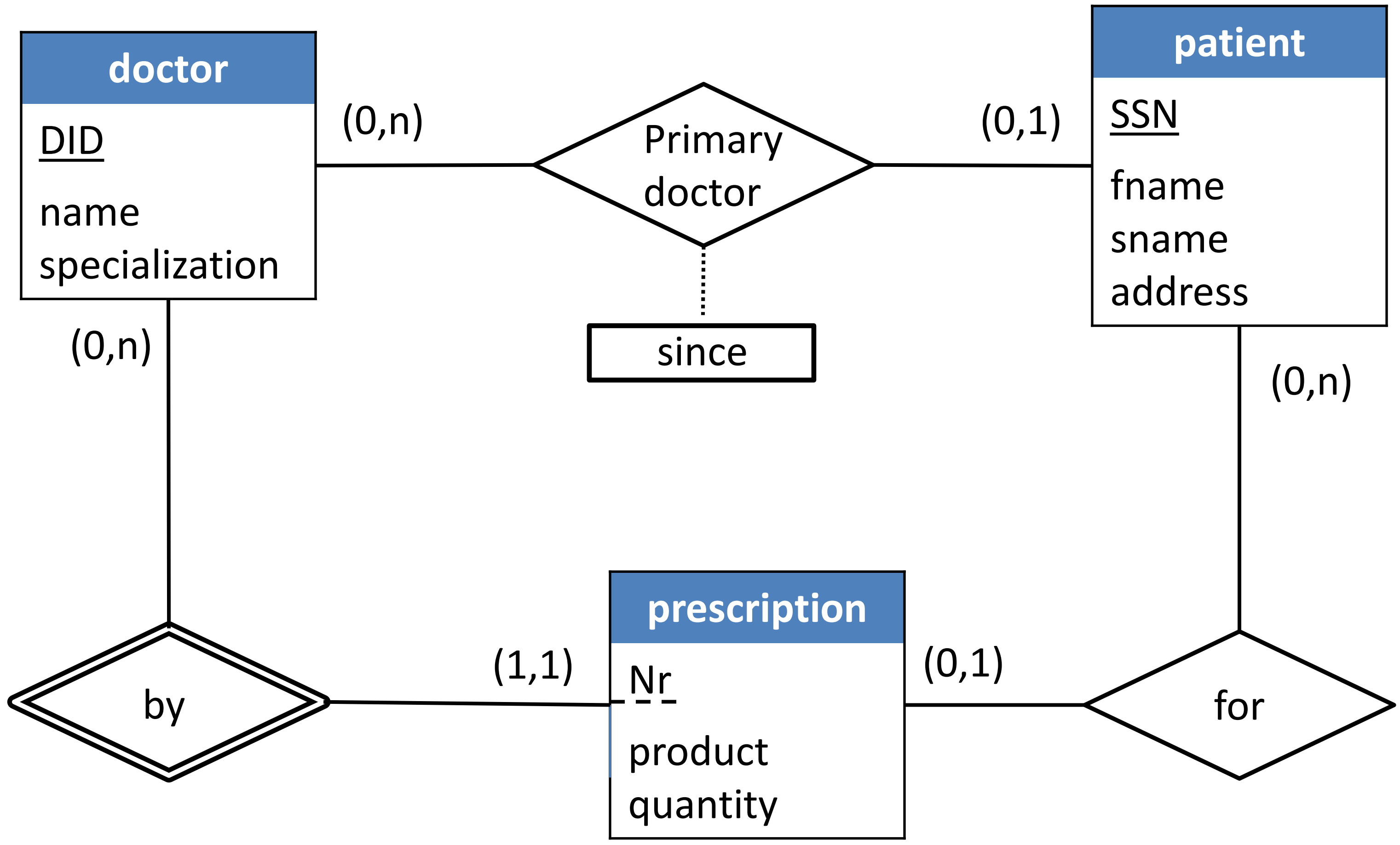
Student	Code	Semester	Grade
Phil	2ID45	Spring 2011	A+
Mary	2ID45	Spring 2011	C
John	2ID45	Spring 2011	B-
Paul	2ID05	Spring 2011	C

Offerings

Code	Semester	Lecturer
2ID45	Spring 2011	Calders
2ID05	Spring 2011	Fletcher

- Offerings(Code) $\subseteq$ Courses(Code)
- Follows(Code,Sem) $\subseteq$ Offerings(Code,Sem)

Translate to the relational model



Database Normalization

- Relation R is in *Boyce-Codd Normal Form* if: for every functional dependency $X \rightarrow Y$ that holds in R, either
  - $Y \subseteq X$, or
  - X is a superkey
- Idea: avoid redundancies, which may lead to inconsistencies

Example: Non-BCNF

Student	Code	Name	Semester	Lecturer	Grade
Phil	2ID45	Advanced Databases	Spring 2011	Calders	A+
Mary	2ID45	Advanced Databases	Spring 2011	Calders	C
John	2ID45	Advanced Databases	Spring 2011	Calders	B-
Paul	2ID05	Databases I	Spring 2011	Fletcher	C

- Some violating FDs:
  - Code → Name
  - Code, Semester → Lecturer

Example: BCNF

Courses

Code	Name
2ID45	Advanced Databases
2ID05	Databases I

Follows

Student	Code	Semester	Grade
Phil	2ID45	Spring 2011	A+
Mary	2ID45	Spring 2011	C
John	2ID45	Spring 2011	B-
Paul	2ID05	Spring 2011	C

Offerings

Code	Semester	Lecturer
2ID45	Spring 2011	Calders
2ID05	Spring 2011	Fletcher

OLTP Systems

- Relations in principle normalized
  - Avoid update inconsistencies
- Primary keys and foreign key dependencies are automatically checked
- Transaction management ensures ACID

Exercise: ER Modeling

Design an ER schema for keeping track of information about vote taken in the U.S. House of Representatives during the current two-year congressional session. The database needs to keep track of each U.S. STATEs Name (e.g., Texas, New York, Connecticut) and include the Region of the state (whose domain is Northeast, Midwest, Southeast, Southwest, West). Each congress person in the House of Representatives is described by his or her Name, plus the District represented, the start date when the congress person was first elected, and the political Party to which he or she belongs (whose domain is Republican, Democrat, Independent, Other)). The database keeps track of each BILL (i.e., proposed law), including the name of the bill, the date of the vote on the bill, whether the bill passed or failed (whose domain is Yes, No, and the Sponsor (the congress person(s) who sponsored-that is, proposed-the bill). The database keeps track of how each congress person voted on each bill (domain of Vote attribute is Yes, No, Abstain, Absent).