Relational Models

Slides are adopted from Aarne Ranta @ Chalmers University

Relational Data Model

- ***Relation***:

  $R \subseteq D_1 \times ... \times D_n$

  $D_1, D_2, ..., D_n$ are domains

  Example: AddressBook $\subseteq$ *string x string x integer*

- ***Tuple***: $t \in R$

  Example: *t* = ("Mickey Mouse", "Main Street", 4711)

- ***Schema***: associates labels to domains

  Example:

  AddrBook: {[*Name: string, Address: string, <u>Tel#:integer</u>*]}

Relational Data Model

AddrBook		
Name	**Street**	**<u>Tel#</u>**
Mickey Mouse	Main Street	4711
Minnie Mouse	Broadway	94725
Donald Duck	Broadway	95672
…	…	…

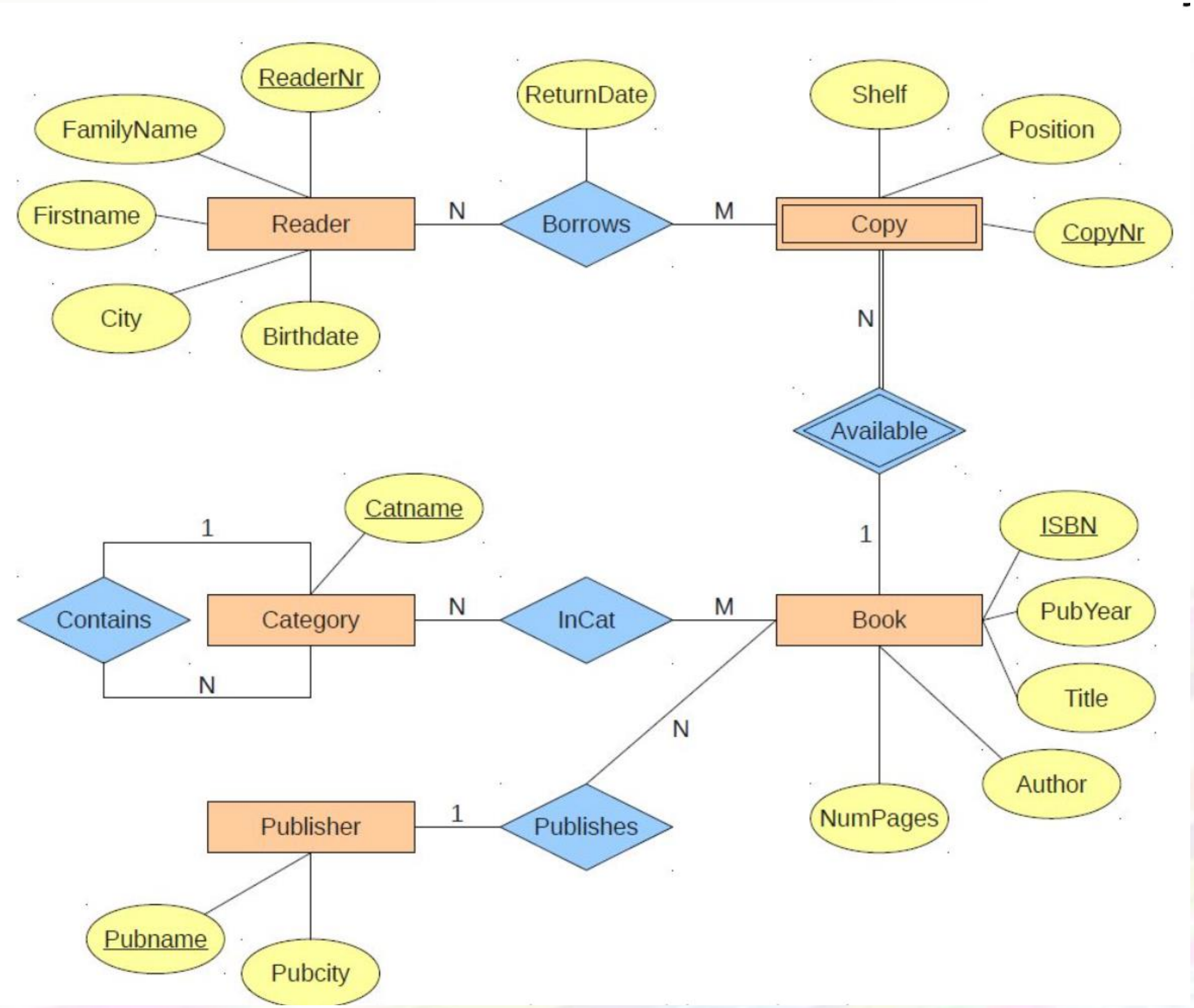
- Key: minimal set of attributes that identify each tuple uniquely
  - E.g., {<u>Tel#</u>}
- Primary Key: (marked in schema by underlining)
  - select one key
  - use primary key for references

Exercise1: Quiz

- Why does every relation in a relational schema have at least one key?

- The tuples of a relation have to be differentiated
- Therefore, the set of all the attributes of a relation must uniquely identify every tuple
- However, most of the time, a key consisting of only one or a few attributes is enough
- A key is defined as:
  - A set of attributes that allow to uniquely identify a tuple
  - A set of attributes from which no further elements can be removed

Exercise2: Library System



Tasks to Complete

- Task 1: Entities to relations:
  - Reader (ReaderNr, FamilyName, FirstName, City, Birthday) (1)
  - Book (ISBN, Title, Author, NumPages, PubYear, PubName) (2)
  - Publisher (Pubname, Pubcity) (3)
  - Category (Catname) (4)
  - Copy (ISBN, CopyNr, Shelf, Position) (5)

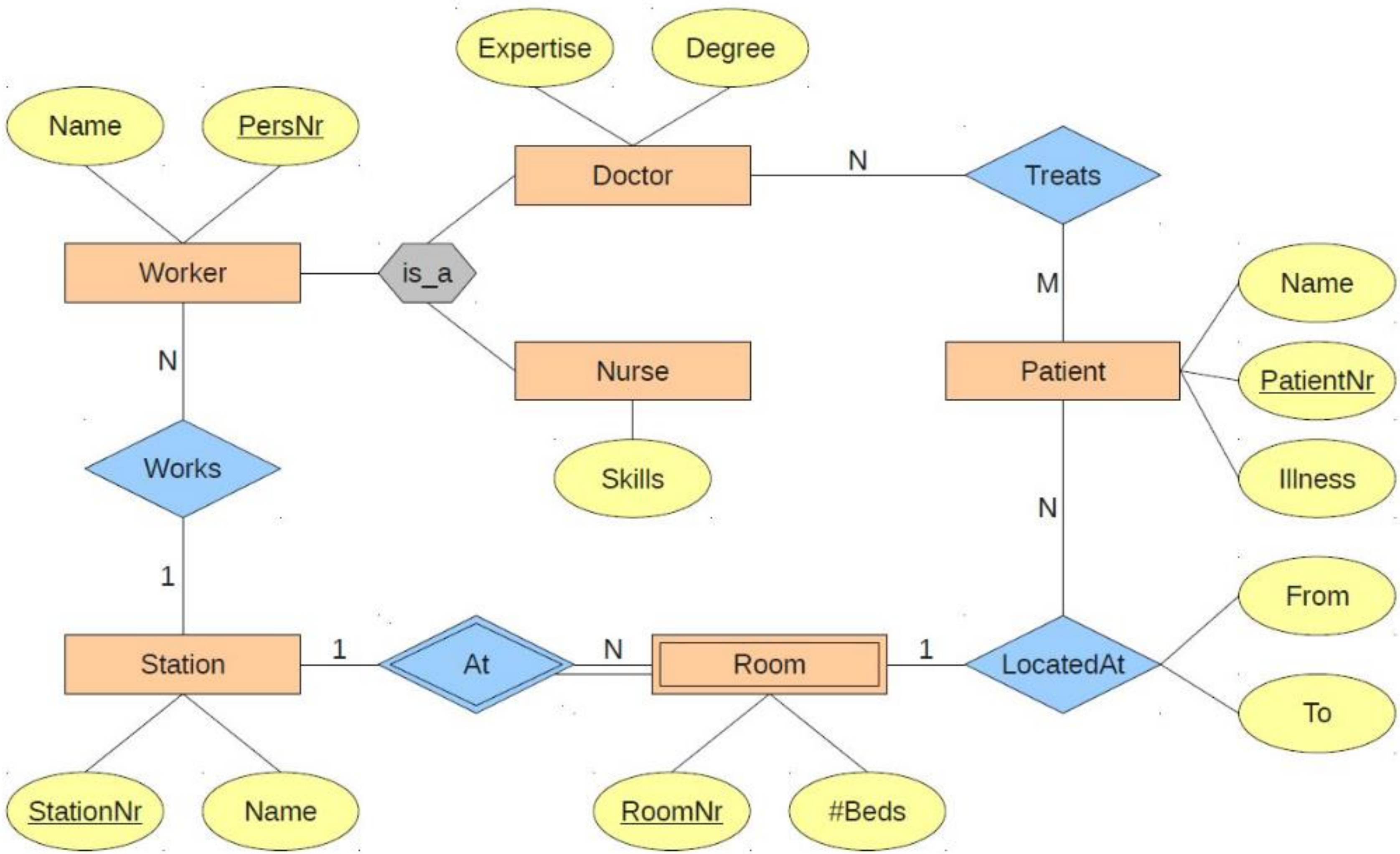
- Task 2: Relationships to relations:
  - Borrows (ReaderNr, ISBN, CopyNr, ReturnDate) (6)
  - Available (ISBN, CopyNr) (7)
  - Contains (CatName, ContainedIn) (8)
  - InCat (ISBN, CatName) (9)
  - Publishes (ISBN, Pubname) (10)

- Task 3: Combine relations that are of type 1:1 or 1:N
  - (7)->(5), (8)->(4), (10)->(2)

Tasks to Complete…

- Task 4: Complete Solution
  - Reader (ReaderNr, FamilyName, Firstname, City, Birthday)
  - Book (ISBN, Title, Author, NumPages, PubYear, PubName)
  - Publisher (Pubname, Pubcity)
  - Category (Catname, ContainedIn)
  - Copy (ISBN, CopyNr, Shelf, Position)
  - Borrows (ReaderNr, ISBN, CopyNr, ReturnDate)
  - InCat (ISBN, CatName)

Exercise3: Hospital



ER Translations: Relations

- Straightforward (Entities with 1:N Relationships)
  - Station (<u>StationNr</u>, Name)
  - Worker (<u>PersNr</u>, Name, StationNr)
  - Room (<u>StationNr</u>, <u>RoomNr</u>, NumBeds)
  - Patient (<u>PatientNr</u>, Name, Illness, RoomNr, StationNr, From, To)
- Additional Relation for N:M Relationship
  - Treats (<u>PatientNr</u>, <u>PersNr</u>)
- Generalization inherits attributes
  - Doctor (<u>PersNr</u>, Name, StationNr, Expertise, Degree)
  - Nurse (<u>PersNr</u>, Name, StationNr, Skills)

Relational Algebra

- What is Algebra?
  - It consist of Operators and Atomic Operands
  - E.g., (X+Y) * Z
- Relational Algebra
  - It is another example of an algebra
  - Relational Operators used with Relations as Operands
- Operation classification
  - Set Operations
  - Remove parts of relation
  - Combine tuples of relations
  - Modification, e.g.,Renaming

Relational Algebra…

- Relational algebra is a formally defined algebra. This means:
  - We can prove, whether two algebraic expressions are equivalent.
- SQL is based on relational algebra.
- We can prove that two SQL queries are equivalent.
- The DB can rewrite a SQL query and prove its correctness.

Operators

Operator	Name
σ	Selection
π	Projection
$\times$	Cartesian Product
$\bowtie$	Join
ρ	Rename
$-$	Set Minus
$\div$	Relational Division
$\cup$	Union
$\cap$	Intersection
$\ltimes$	Semi-Join (Left)
$\rtimes$	Semi-Join (Right)
⟕	Left outer Join
⟖	Right outer Join
⟗	Full outer Join

- A minimal set of operators are:

  - Projection
  - Selection
  - Cartesian Product
  - Union
  - Set Minus
  - Rename

Exercise 4

- Consider the following relational schema:

  Reader (RDNR, Surname, Firstname, City, Birthday)
  Book (ISBN, Title, Author, NoPages, PubYear, PublisherName)
  Publisher (PublisherName, PublisherCity)
  Category (CategoryName, BelongsTo)
  Copy (ISBN, CopyNumber, Shelf, Position)
  Loan (ReaderNr, ISBN, Copy, ReturnDate)
  BookCategory (ISBN, CategoryName)

- Formulate the following queries in relational algebra:

  Which are the last names of the readers in Kazan?

  $$\Pi_{\text{surname}} (\sigma_{\text{City=Kazan}}(\text{Reader}))$$

  Which Books (Author, Title) are from publishers in Kazan, Moscow, or St. Petersburg?

$$\Pi_{\text{author,title}} (\text{Book} \bowtie (\sigma_{\text{City=KazanORCity=MoscowORCity=St.Petersburg}}(\text{Publisher})))$$

Exercise 5 (1/3)

- Consider the following relational schema:

  Cities (Name, State)
  Stations (Name, NoPlatforms, CityName, State)
  Itinerary (ItNr, Length, StartStation, DestinationStation)
  Connections (FromStation, ToStation, ItNr, Departure, Arrival)

- Suppose that the relation “Connections” already contains the transitive closure.
- For example, if there is a train from Kazan via Moscow and St. Petersburg to Vladivostok, then there exists a relation tuple for Kazan -> Moscow, Kazan->St. Petersburg and Kazan->Vladivostok.
- Formulate the following queries in relational algebra:

Exercise 5 (2/3)

- Cities (Name, State)
  Stations (Name, NoPlatforms, CityName, State)
  Itinerary (ItNr, Length, StartStation, DestinationStation)
  Connections (FromStation, ToStation, ItNr, Departure, Arrival)
- **Find all the direct connections from Kazan to Vladivostok**

$$\rho_{\text{FromName<-Name}}(\Pi_{\text{Name}}(\sigma_{\text{CityName=Kazan}}(\text{Stations})))$$
$$\bowtie_{\text{FromName=FromStation}} \text{Connections}$$
$$\bowtie_{\text{ToName=ToStation}}$$
$$(\rho_{\text{ToName<-Name}}(\Pi_{\text{Name}}(\sigma_{\text{CityName=Vladivostok}}(\text{Stations}))))$$

Exercise 5 (3/3)

- Cities (<u>Name</u>, <u>State</u>)
  Stations (<u>Name</u>, NoPlatforms, CityName, State)
  Itinerary (<u>ItNr</u>, Length, StartStation, DestinationStation)
  Connections (<u>FromStation</u>, <u>ToStation</u>, <u>ItNr</u>, Departure, Arrival)

- **Find all the single-transfer connections from Kazan to Yekaterinburg. The transfer station can be any of the stations but the connecting trains should run on the same day. (You can use a function DAY() on the attributes Departure and Arrival in order to determine the day)**

$$\rho_{\text{FromName<-Name}(}\Pi_{\text{Name}}(\sigma_{\text{CityName=Kazan}}(\text{Stations}))))$$
$$\bowtie_{\text{FromName=c1.FromStation}}\ \rho_{\text{c1}}(\text{Connections})$$
$$\bowtie_{\text{c1.ToStation=c2.FromStation}\wedge\text{c1.Arrival}<\text{c2.Departure}}$$
$$\wedge\text{DAY(c1.Arrival)=DAY(c2.Departure)}\ \wedge\text{c1.ItNr}<>\text{c2.ItNr}$$
$$\rho_{\text{c2}}(\text{Connections})$$
$$\bowtie_{\text{ToName=ToStation}}$$
$$(\rho_{\text{ToName<-Name}(}\Pi_{\text{Name}}(\sigma_{\text{CityName=Yekaterinburg}}(\text{Stations}))))$$

Professor			
PersID	Name	Level	Room
2125	Sokrates	FP	226
2126	Russel	FP	232
2127	Kopernikus	AP	310
2133	Popper	AP	52
2134	Augustinus	AP	309
2136	Curie	FP	36
2137	Kant	FP	7

Student		
StuID	Name	Semester
24002	Xenokrates	18
25403	Jonas	12
26120	Fichte	10
26830	Aristoxenos	8
27550	Schopenhauer	6
28106	Carnap	3
29120	Theophrastos	2
29555	Feuerbach	2

Lecture			
LecID	Title	CP	PersID
5001	Grundzüge	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Mäeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126
5259	Der Wiener Kreis	2	2133
5022	Glaube und Wissen	2	2134
4630	Die 3 Kritiken	4	2137

requires	
Prerequisite	Follow-up
5001	5041
5001	5043
5001	5049
5041	5216
5043	5052
5041	5052
5052	5259

attends	
StuID	LecID
26120	5001
27550	5001
27550	4052
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049
29555	5022
25403	5022

Researcher			
PersID	Name	Area	Supervisor
3002	Platon	Ideenlehre	2125
3003	Aristoteles	Syllogistik	2125
3004	Wittgenstein	Sprachtheorie	2126
3005	Rhetikus	Planetenbewegung	2127
3006	Newton	Keplersche Gesetze	2127
3007	Spinoza	Gott und Natur	2126

tests			
StuID	LecID	PersID	Grade
28106	5001	2126	1
25403	5041	2125	2
27550	4630	2137	2

Relational Calculus

Queries have the following form:

$\{t \mid P(t)\}$

with t a variable, P(t) a predicate.

Examples:

- All full professors

  $\{p \mid p \in \text{Professor} \wedge p.\text{Level} = \text{'FP'}\}$

- Students who attend at least one lecture of Curie

  $\{s \mid s \in \text{Student}$

  $\wedge \exists a \in \text{attends}(s.\text{StuID}=a.\text{StuID}$

  $\wedge \exists l \in \text{Lecture}(a.\text{LecID}=l.\text{LecID}$

  $\wedge \exists p \in \text{Professor}(p.\text{PersID}=l.\text{PersID}$

  $\wedge p.\text{Name} = \text{'Curie'})))\}$

- Who attends all lectures with 4 CP?

$$\{s \mid s \in \text{Student} \land \forall l \in \text{Lecture}\ (l.CP{=}4 \Rightarrow \exists a \in \text{attends}(a.LecID{=}l.LecID \land a.StuID{=}\ s.StuID))\}$$

- There are two variants of relational calculus:

tuple relational calculus (as in examples above, tuple vars)

domain relational calculus (variables iterate over domains)

Tuple Relational Calculus

Atoms

- s | R
  s is a tuple variable, R is a name of a relation
- s.A ϕ t.B or s. A ϕ c
  s and t tuple variables, A and B attribute names
  ϕ a comparison (i.e., =, ≠, ≤, ...)

c is a constant (i.e., 25)

Formulas

- All atoms are legal formulas
- If P is a formula, then
  $\neg$P and (P) are also formulas
- If P_1 and P_2 are formulas, then
  $P_1 \wedge P_2$, $P_1 \vee P_2$ and $P_1 \Rightarrow P_2$ are also formulas
- If P(t) is a formula with a free variable t, then

  $\forall t \in R(P(t))$ and $\exists\ t \in R(P(t))$ are also formulas

Safety

- Restrict formulas to queries with finite answers

  Semantic not syntactic property!

- Example: The following expression is not safe

$$\{n \mid \neg (n \in Professor)\}$$

- Definition of safety
  - result must be subset of the "domain of the formula"
  - "domain of the formula"
    - All constants used in the formula
    - All domains of relations used in the formula

Domain Relational Calculus

An expression has the following form

$$\{[v_1, v_2, ..., v_n] \mid P(v_1, ..., v_n)\}$$

each $v_1, ..., v_n$ is either a domain variable or a constant.

P is a formula.

Example: StuID and Name of all students tested by Curie:

$$\{[l, n] \mid \exists s\,([l, n, s] \in Student \wedge \exists v, p, g\,([l, v, p, g] \in tests \wedge \exists a, r, b([p, a, r, b] \in Professor \wedge a = \text{'Curie'})))\}$$

Safety in the DRC

- Defined in same way as for tuple relational calculus
- Example: The following expression is not safe

  $\{[p,n,r,o] \mid \neg ([p,n,r,o] \in \text{Professor})\}$

- (see text book for exact definition of safety in DRC)

Codd`s Theorem

The three languages

1. relational algebra,
2. tuple relational calculus (safe expressions only)
3. domain relational calculus (safe expressions only)

are **equivalent**

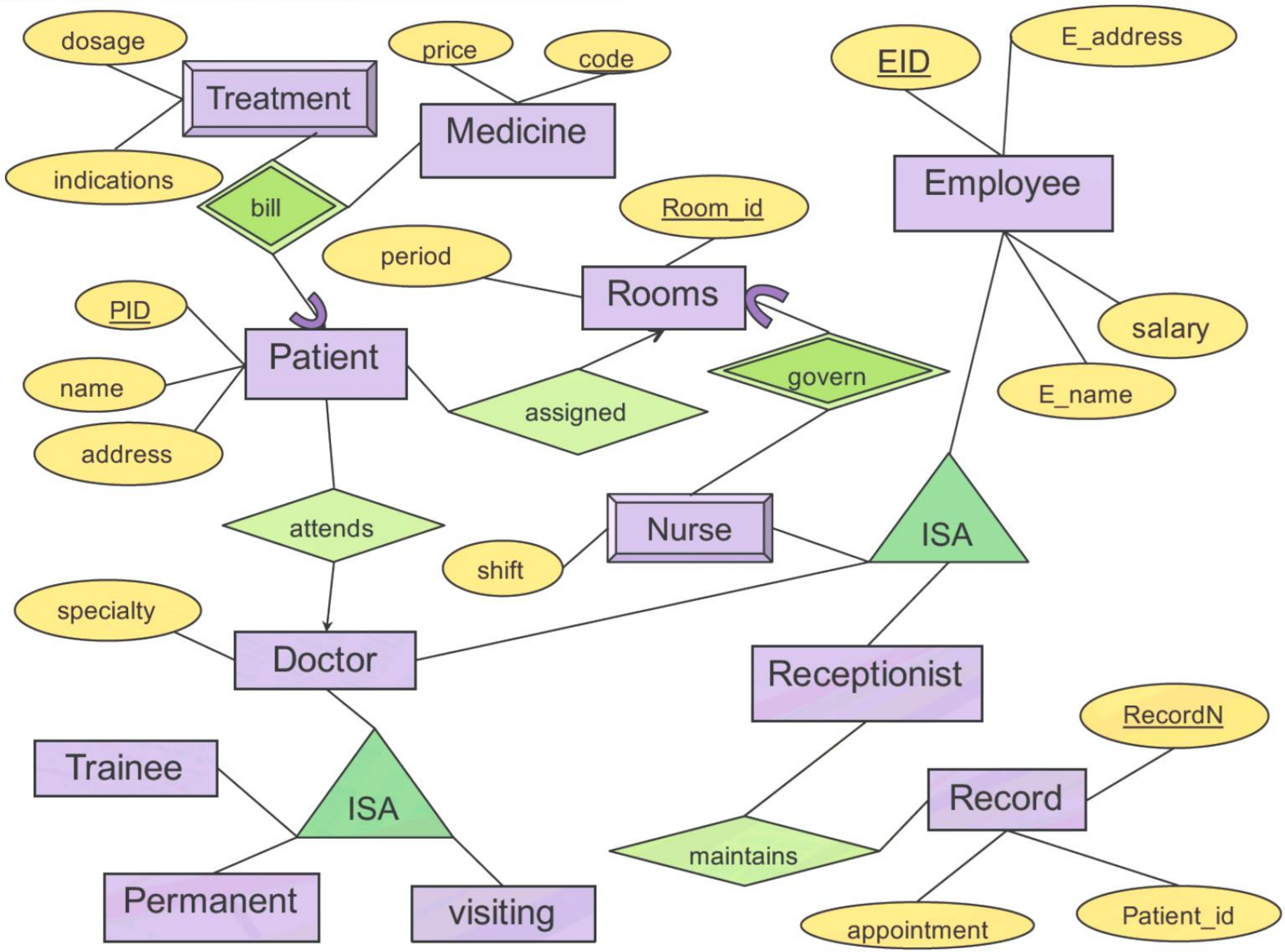
Impact of Codd`s theorem

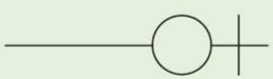
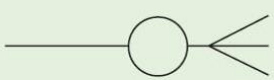
- SQL is based on the relational calculus
- SQL implementation is based on relational algebra
- Codd`s theorem shows that SQL implementation is correct and complete.

Assignment

1. Translate the given ER diagram to relational data model.

2. Formulate following queries in relational algebra.
   a) Find all employees who are taking care of patients in room 107.
   b) Find all the nurses that Dr.Alex is not working with.
   c) Find all employees who has more salary than at least one doctor.
   d) Find all the rooms that has at least one patient.


dosage
Treatment
price
code
Medicine
EID
E_address
Employee
indications
bill
Room_id
period
Rooms
PID
Patient
salary
govern
name
assigned
E_name
address
attends
Nurse
ISA
shift
specialty
Doctor
Receptionist
RecordN
Trainee
ISA
Record
maintains
Permanent
visiting
appointment
Patient_id

Symbol	Meaning	
	One—Mandatory	Exactly one
	Many—Mandatory	At least one
	One—Optional	At most one
	Many—Optional	From 0 to many

3. Formulate the queries in relational algebra and **compute the results**.

Flights (<u>flight#</u>, from, to, distance)

Aircraft (<u>aid</u>, aname, range)

Certified (<u>eid</u>, <u>aid</u>)

Employees (<u>eid</u>, ename, salary)

flight = {(112, Moscow, Kazan, 600), (111, Kazan, Moscow, 600),
(100, Kazan, Istanbul, 1300), (300, Kazan, St. Petersburg, 1000)}
aircraft = {(B747, a, 1000), (B997, b, 1200), (B1100, c, 3000),
(B970, d, 800)}
certified = {(1, B747), (1, B997), (2, B747), (3, B1100), (3, B747), (4, B747),
(4, B970), (4, B1100)}
employees= {(1, Mike, $50,000), (2, Andrew, $80,000),
(3, Alex, $60,000),(4,Sam, $40,000),(5, Dmitry, $150,000)}

a) Find an employee with the third highest salary.
b) Find employees who can fly the flight100.
c) Find flights that are certified only by employees with salary more than $70,000.

4. Given relations R and S, compute range of the number of resulting tuples for $R \bowtie S$, where $|R|=m$ and $|S|=n$.

5. Determine if $\Pi_N(R-S)$ and $\Pi_N(R)-\Pi_N(S)$ are equivalent. If not, give an example.

6. Consider relations R(A,B,C) and S(C,D). Determine the equivalence of given expressions and show reasons.

a) $R \bowtie S$ and $\sigma_{R.C=S.C}(R \times S)$
b) $\Pi_C(R \bowtie S)$ and $\Pi_C(R) \bowtie \Pi_C(S)$