DATABASE SYSTEMS

Database design

Bence Molnár

2022.03.01.

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék

Agenda

- Workflow of design
- Specification
- Keys
- E/R diagram
- Relational scheme diagram
- Challenges of relational databases
- Redundancy, anomalies
- Normalizing, decomposition

Database design

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék

Design

- Designing is about to outline a concept in the future, the way to get there and describing the circumstances. Generally multiple solutions exist, thus designing is a sort of decision-making.
- Design should be before the activities, decision is not automatically processed.
- Problems cannot be separated to independent and trivial parts.
- This means that designing is a sequence of decisions.
- Decisions made based on earlier experiences, regulations, standards, examples.
- Relaying on natural sciences: mathematics, physics, statistics and performing abstractions

Abstraction

- This process is about to eliminate the irrelevant properties of things (objects and concepts) and focusing on relevant attributes.
- Three most important abstraction:
  - ***Entities*** *(objects, things, concepts)*
  - ***Attributes***
  - ***Relationships***

World → Table

Problems to solve:

- Managing book borrows at a library.
- Track monthly costs.
- Stock information in a storage.
- Backtracking my works based on time and projects.
- Tracking a bridge elevation changes effected by different loads.

Therefore we create catalogs (tables). Question is how to organize data (structure).

The abstraction process is called data modeling.

A POSSIBLE WORKFLOW OF DATABASE DESIGN

- Collecting goals and requirements (specification)
- Entity-Relationship (E/R) diagram
- Relational database schema
- Queries, transactions and stored procedures
- Building database (Realization)
- Upload data
- Fine tuning and testing

Data modelling

World → E/R diagram → Relational database schema → Realization

Specification

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék

Specification

- Defining objectives
  - *Customer and user needs*
  - *Technical circumstances and restrictions*
- Clarify goals: what kind a forms, reports are needed and what questions should be answered by the database.
- What is the input and output data (functional questions)?
- Selecting methodology
- As software systems are generally limited in view of available methods. Selecting a system determines the methodology as well.

Case study

- Task: A bridge monitoring system will be built to observe the bridge elevation changes and data stored in a DB.
- Goals: Send notification to authorities if the predefined security threshold is exceeded. Store all measurements in DB to make backtracking available.

wind direction
wind speed

triaxial accelerometers

amplifier

central unit

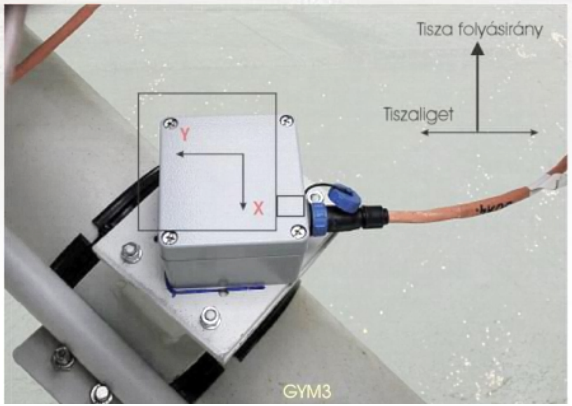
3x6 channel

2 channel

acc3 acc2 acc1

acc4 acc5 acc6

computer ➔



Case study

Input

- Sensor data (raw measurements)
- Security thresholds (limits) determined by experts
- Classifying thresholds and corresponding notification levels

Output

- Report of current state of the bridge
- Notifications
- Backtracking past events

E/R diagram

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék

Entity-Relationship diagram

Entity-Relationship (E/R)

Abstract description

Parts:

- **Entity**: subject of modeling, a well separated part of the world
- **Attribute** (properties): Data/information related to the entity
- **Relationship**: connections of entities

Entity

Marking: rectangles

Bridge

Sensor

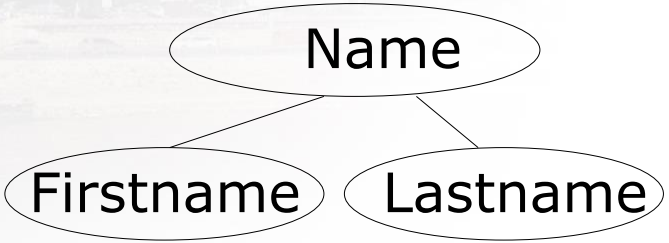
Measure.

Attribute

Marking: ovals

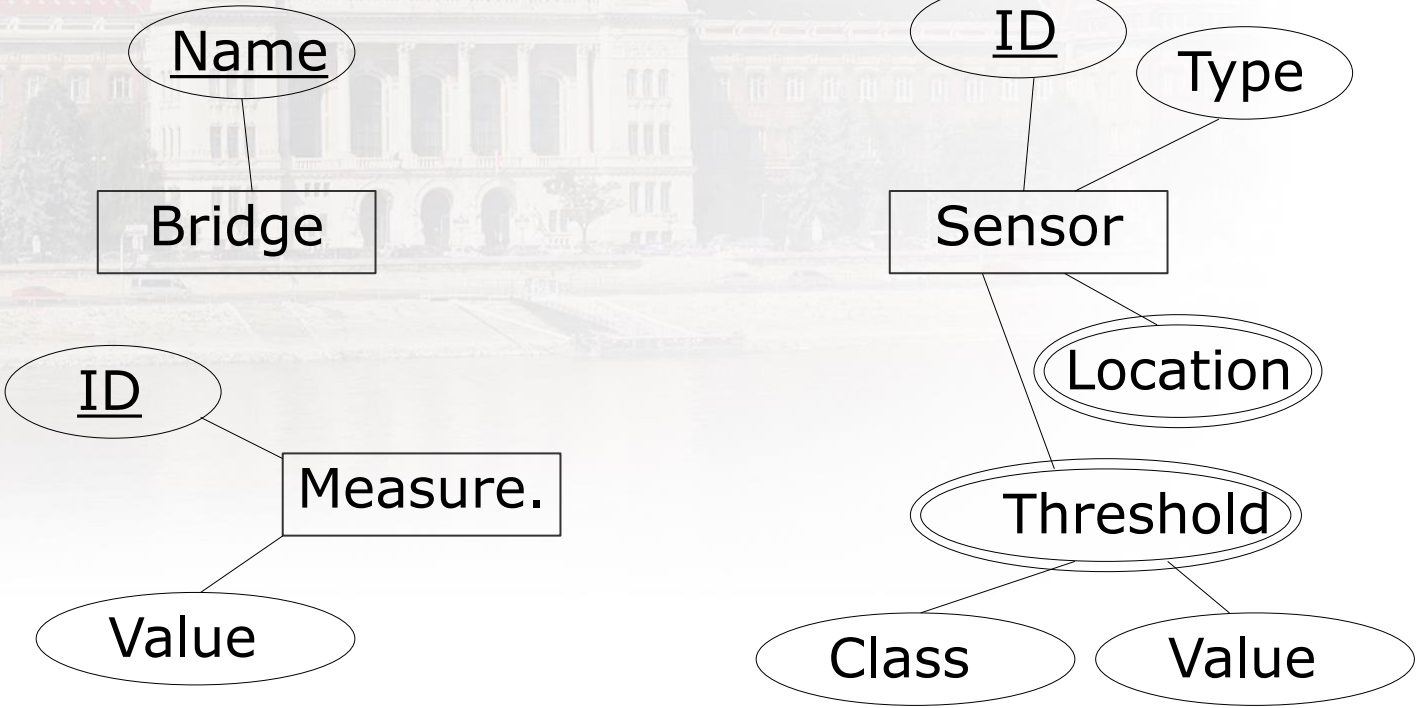
- Simple

- Complex: attribute can be divided to multiple attributes.



- Multi-value attribute: set or list

ATTRIBUTE


Name
Bridge
ID
Measure.
Value
ID
Type
Sensor
Location
Threshold
Class
Value

Relationship (1:1)

Marking: diamonds

Connections of entities.

- One-one (1:1): A head connects to one and only on neck and vice versa.

Relationship (1:N)

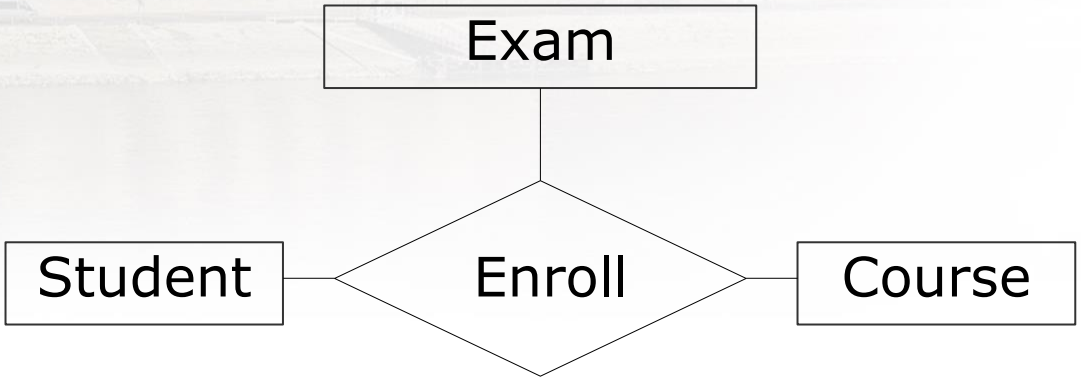
- *One-many relationship (1:N): A dad may have multiple kids, however a kid has only one dad.*

Relationship (N:M)

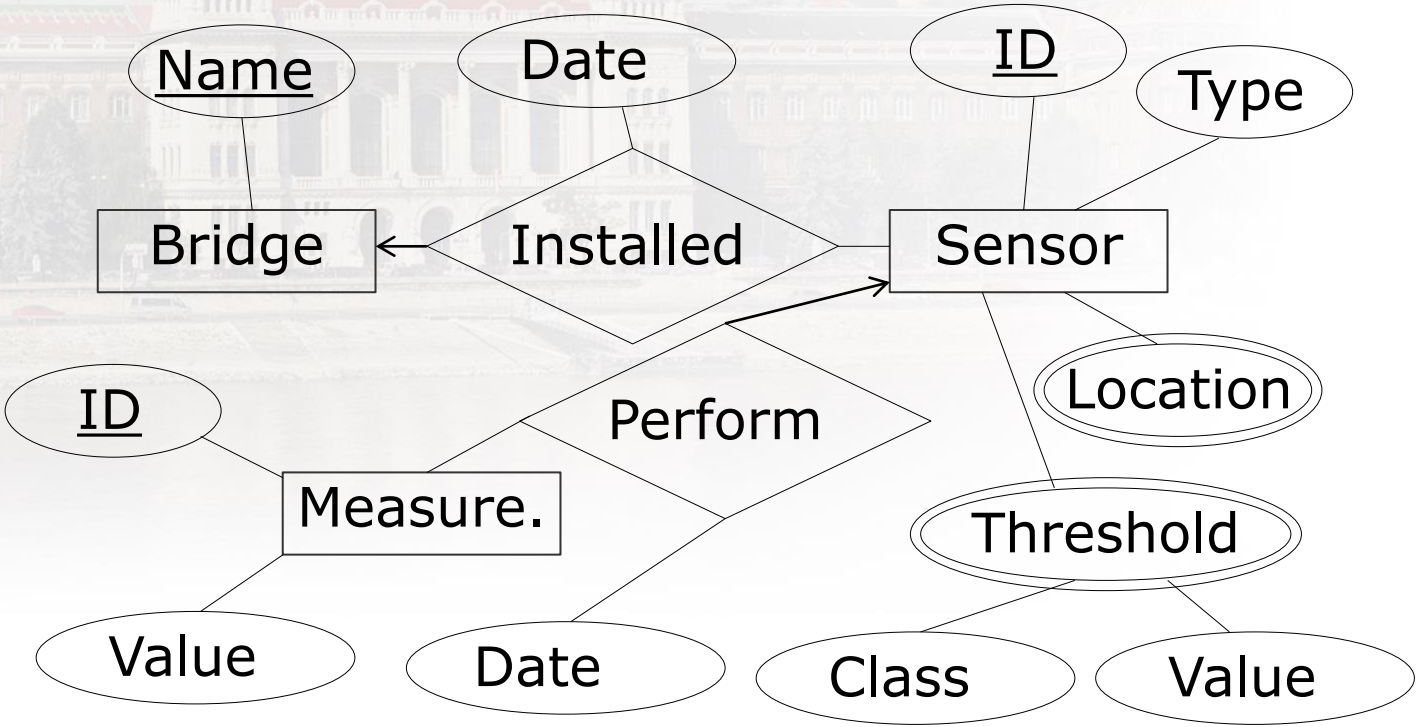
- *Many-many relationship (N:M): A person may have multiple house and a house may have multiple owners.*

Multiway relationship

Multiway relationship: A student may attend to multiple exam of a single class. Can be replaced by multiple two-way relationships.



Relationships and attributes



Tools to draw E/R diagram

- *Manual drawing -> photo*
- *Web browser based editors*
- *Microsoft PowerPoint*
- *LibreOffice Impress*
- *yEd (recommended, Palette → Entity Relationship)*
- *Dia*

Candidate key, superkey

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék

SUPERKEY

Set of attributes which holds that there are no two distinct tuples (rows) that have the same values for the attributes in this set

Name	ID	Age
John Doe	123	18
John Doe	124	18

SK1={{ID}, {ID, name}, {ID, age}, {ID, name, age}}

SK2=SK1 U {{age}, {age, name}}

Name	ID	Age
John Doe	123	18
John Doe	124	**19**

Candidate key

A minimal set of attributes necessary to identify a tuple

Name	ID	Age
John Doe	123	18
John Doe	124	18

K1={{ID}}

K2={{ID}, {age}}

Name	ID	Age
John Doe	123	18
John Doe	124	**19**

Candidate key – example

Name	**ID**	**Class**	**Grade**
John Doe	123	Math	5
John Doe	123	**Biology**	5
John Doe	**124**	Math	5

SK={{ID, class}, {name, ID, class}, …}

K={ID, class}

Compound key

Further examples

Profile(Name: String, Profile: Real, Moment of inertia: Real, Price: Integer)

Sphere(X: Real, Y: Real, Z: Real, R: Real)

Road(Name: String, Class: Integer)

Realty(ParcelNumber: String, Owner: String, Area: Real, Price: Integer)

Sometimes candidate key is missing!

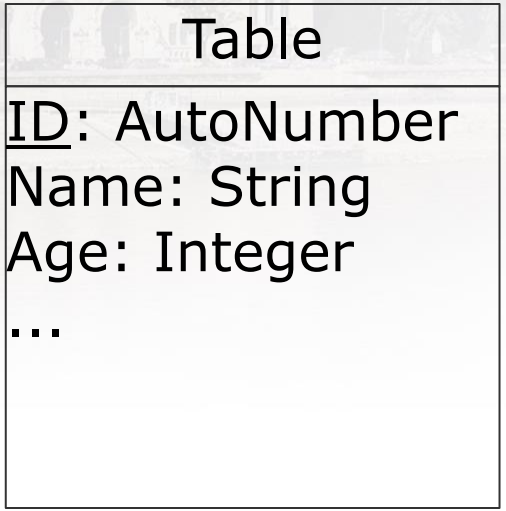
Relational database schema

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék

Relational database schema

- Relational schema: Table(ID: AutoNumber, Name: String, Age: Integer)
- Relational database schema:



KEYS IN RELATIONAL SCHEMA

So far we discussed keys on given data (realization)

However it can be specified in schema to ensure an attribute to be a key.

It's marked in the scheme as follows:

```
Grades(ID: AutoNumber, Name: String, Grade: Integer)
```

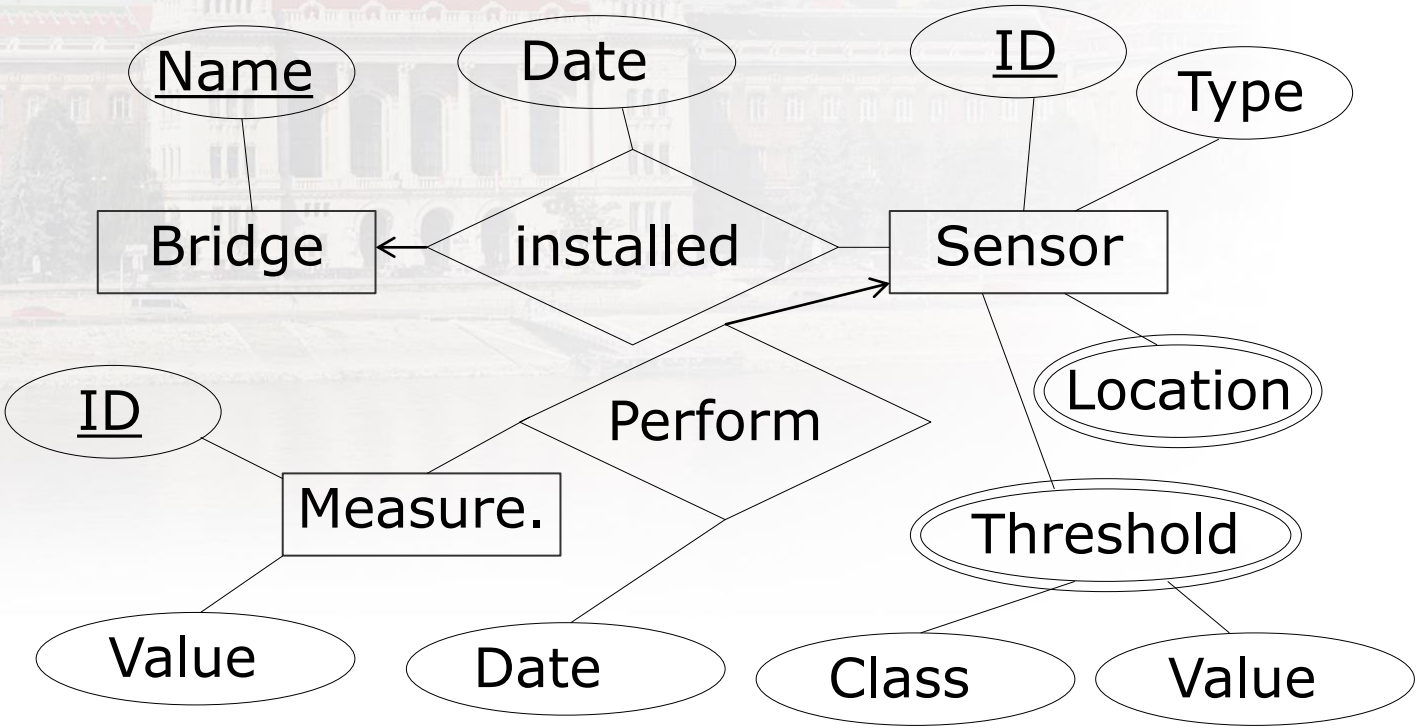
Primary key

- In schema we enforce the primary key field
- Marking: underscore on field name
- Primary Key
- All other fields functionally depends on this field

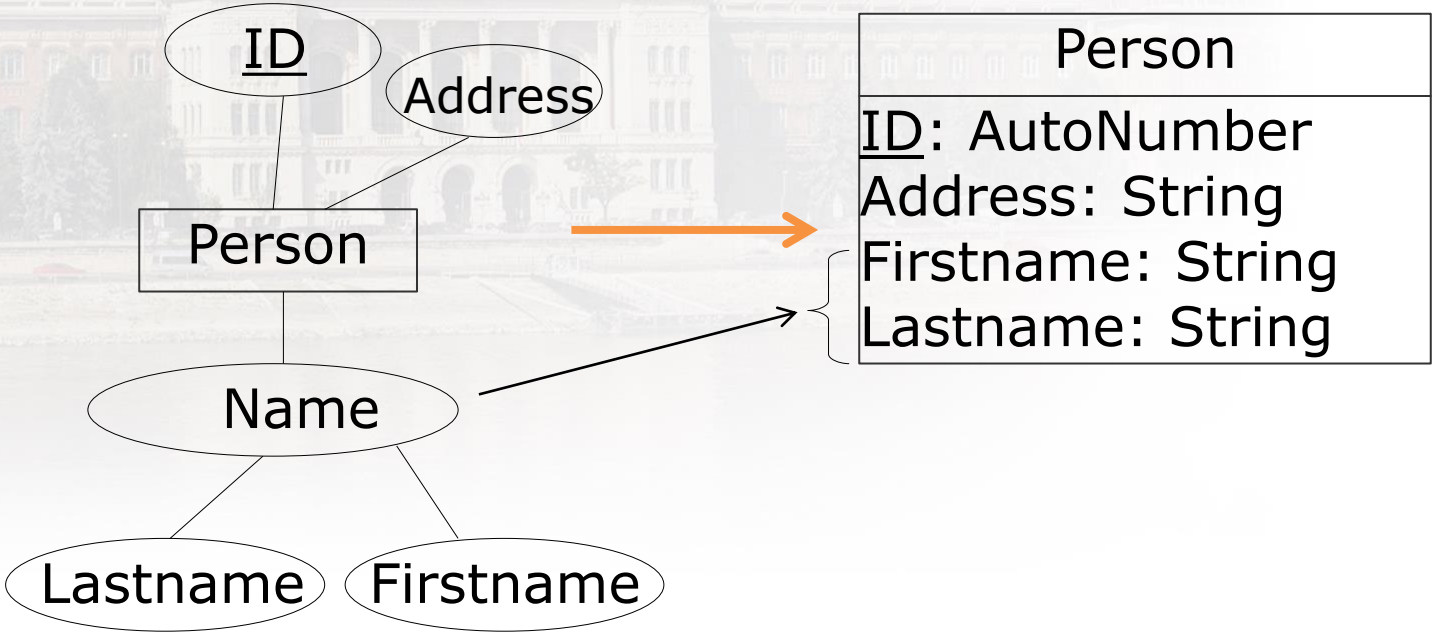
Foreign key

- Refers to an other table's primary key
- Marking: dashed underscore/(FK)
- This is generally not unique in this table, other field does not depend on this field.

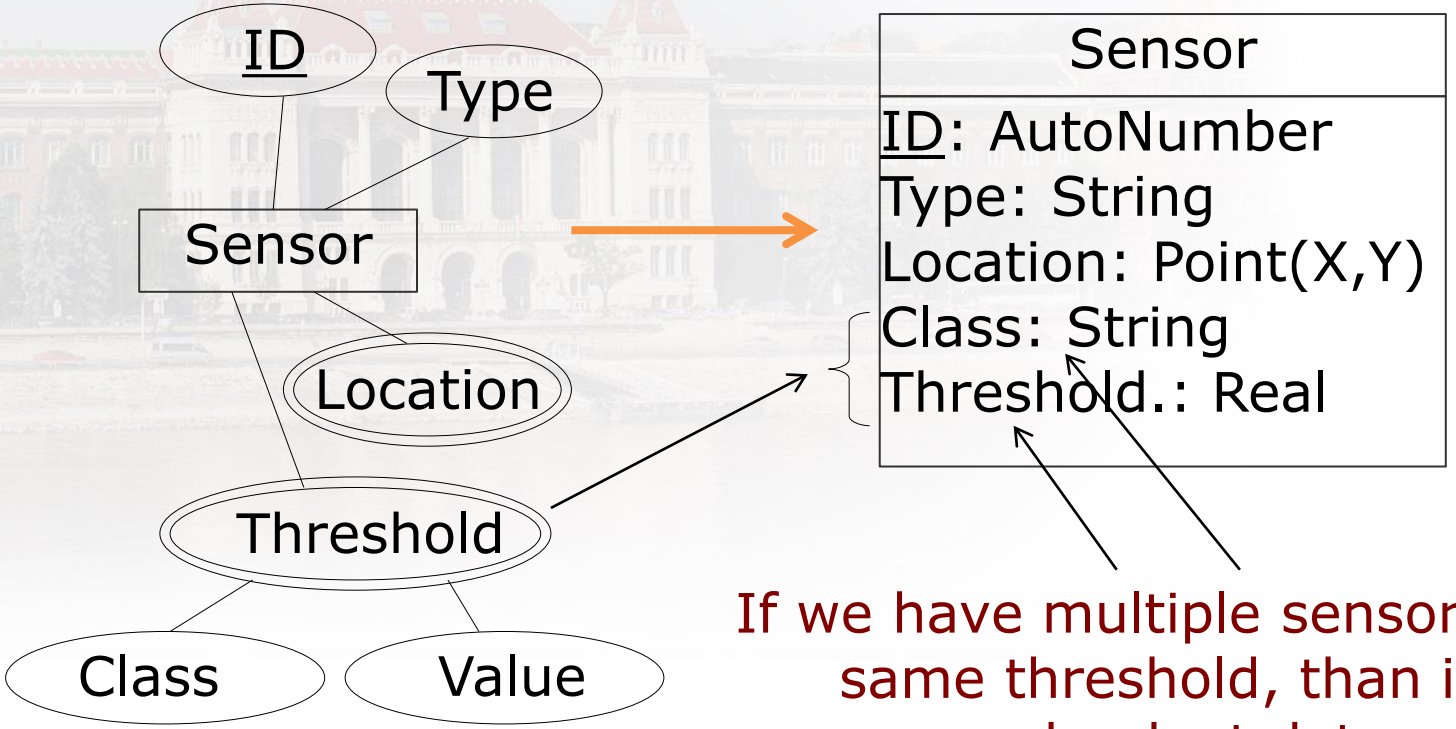
SOURCE: E/R DIAGRAM



ATTRIBUTE TRANSFORMATION

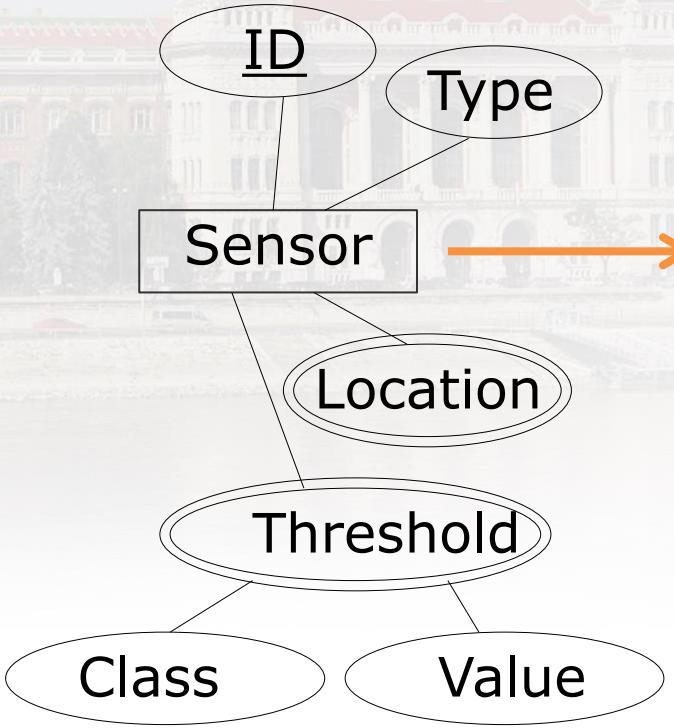


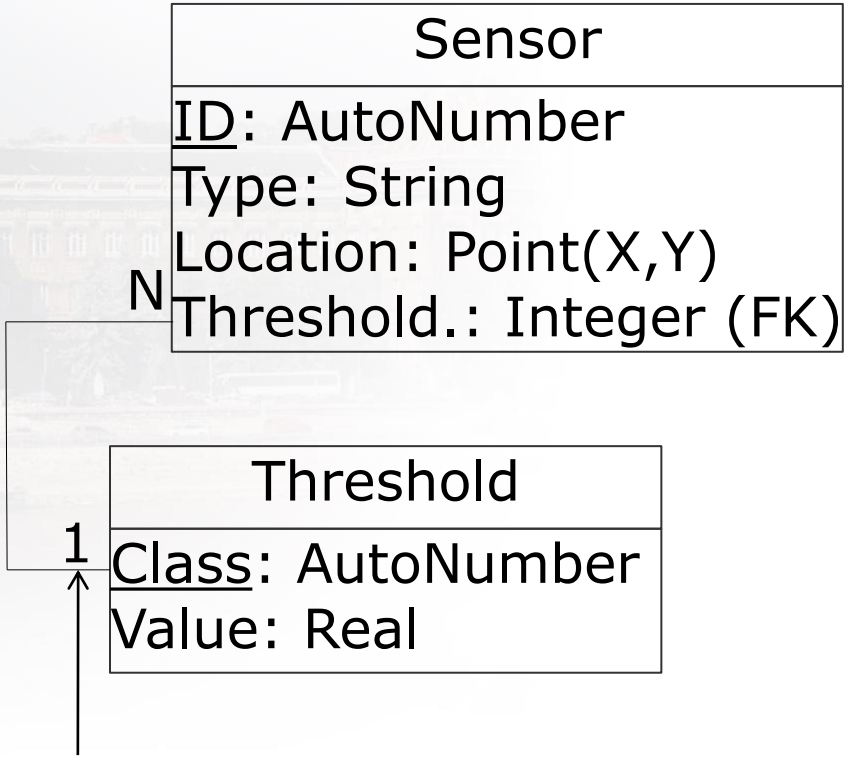
ATTRIBUTE TRANSFORMATION



If we have multiple sensors with same threshold, than it's redundant data

ATTRIBUTE TRANSFORMATION

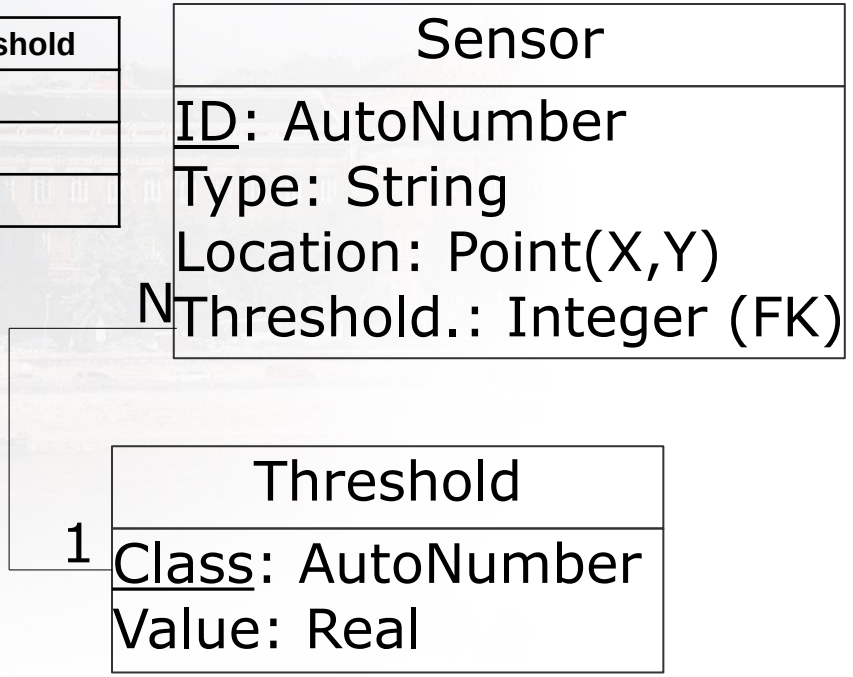




ATTRIBUTE TRANSFORMATION

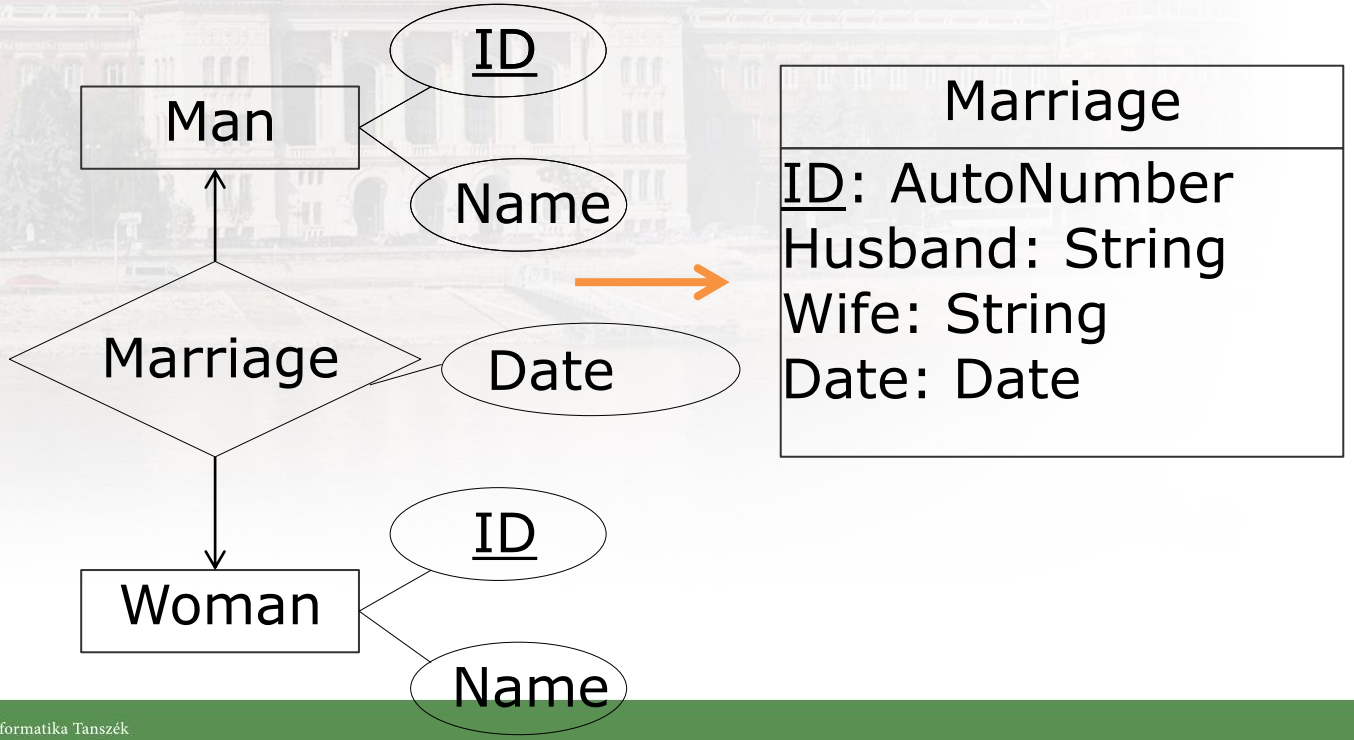
ID	Type	Location	Threshold
1	Strain gauge	P(0,0)	1
2	Strain gauge	O(0,100)	1
3	Accelerometer	P(100,0)	2

Class	Value
1	100
2	200
3	300

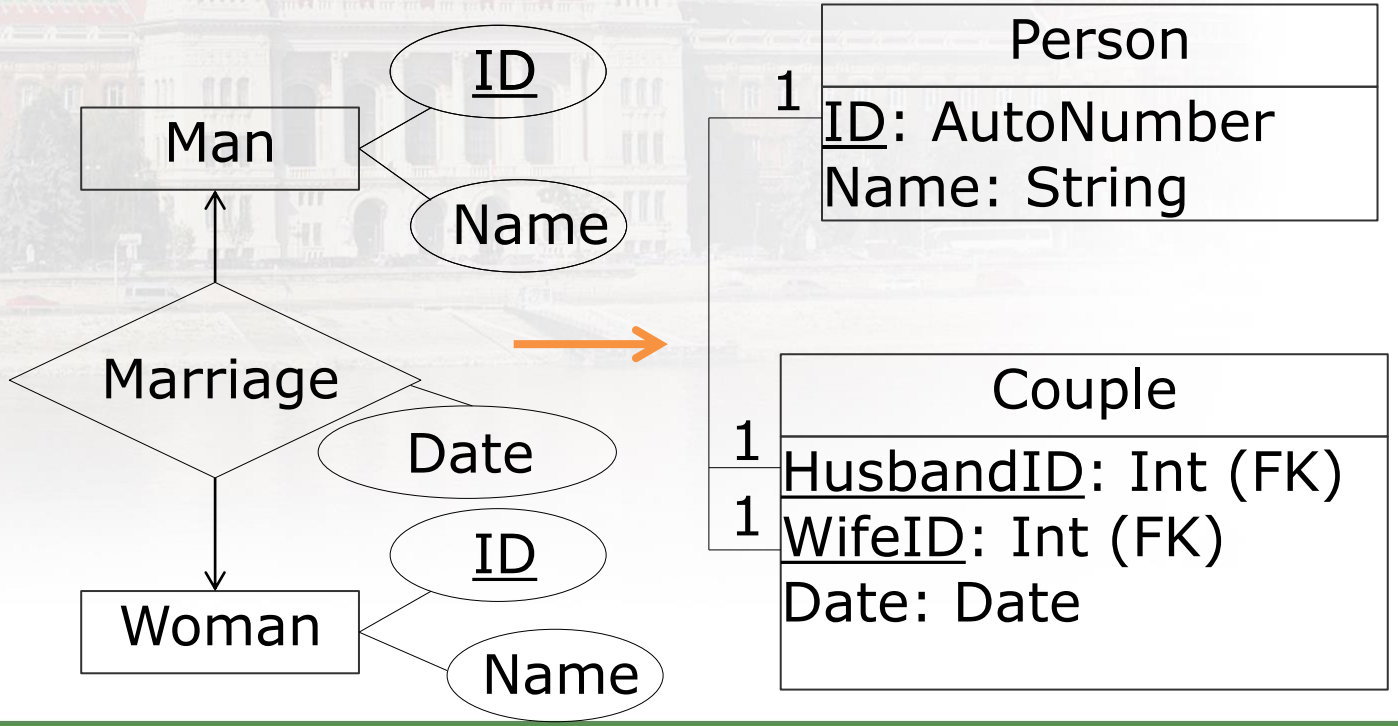


Note: Multiplicity 1 is always on the primary key side!

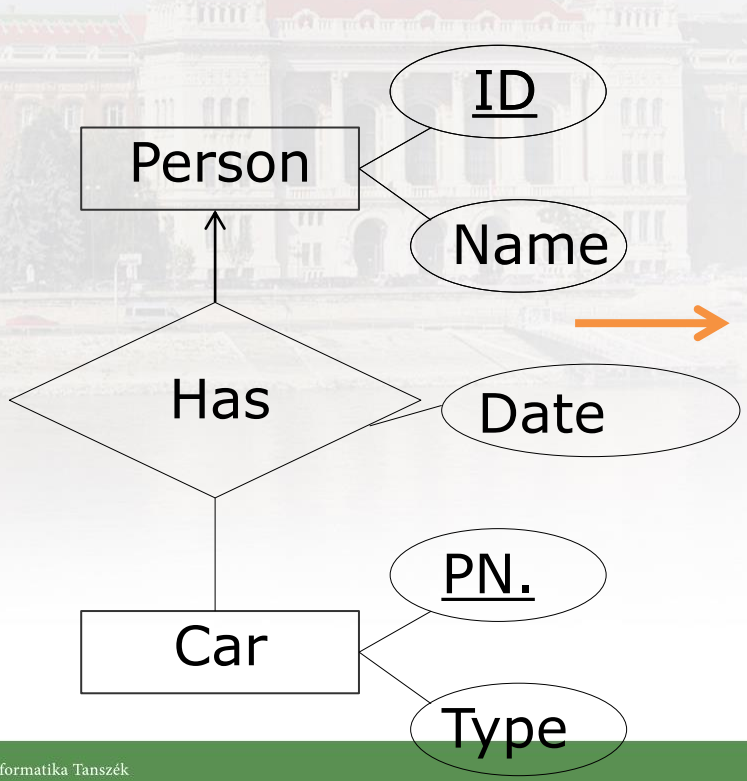
Relationship transformation (1:1)



Relationship transformation (1:1)



Relationship transformation (1:N)



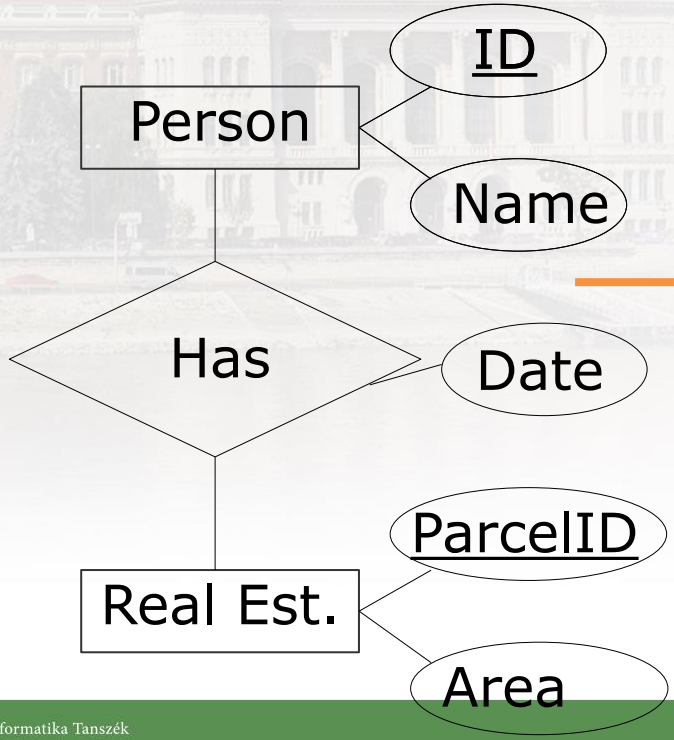
Person
ID: AutoNumber
Name: String

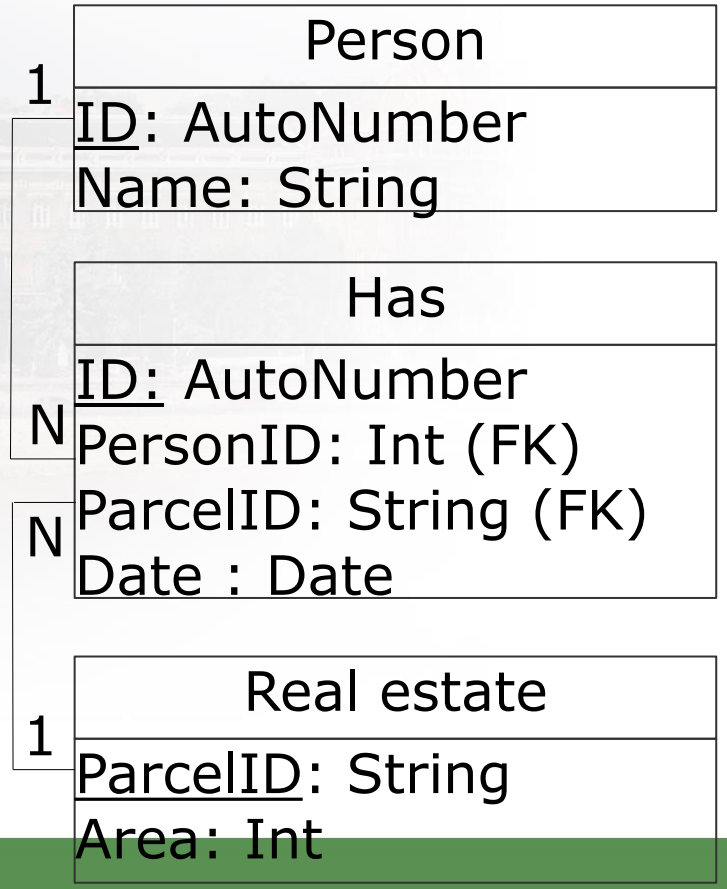
1

N

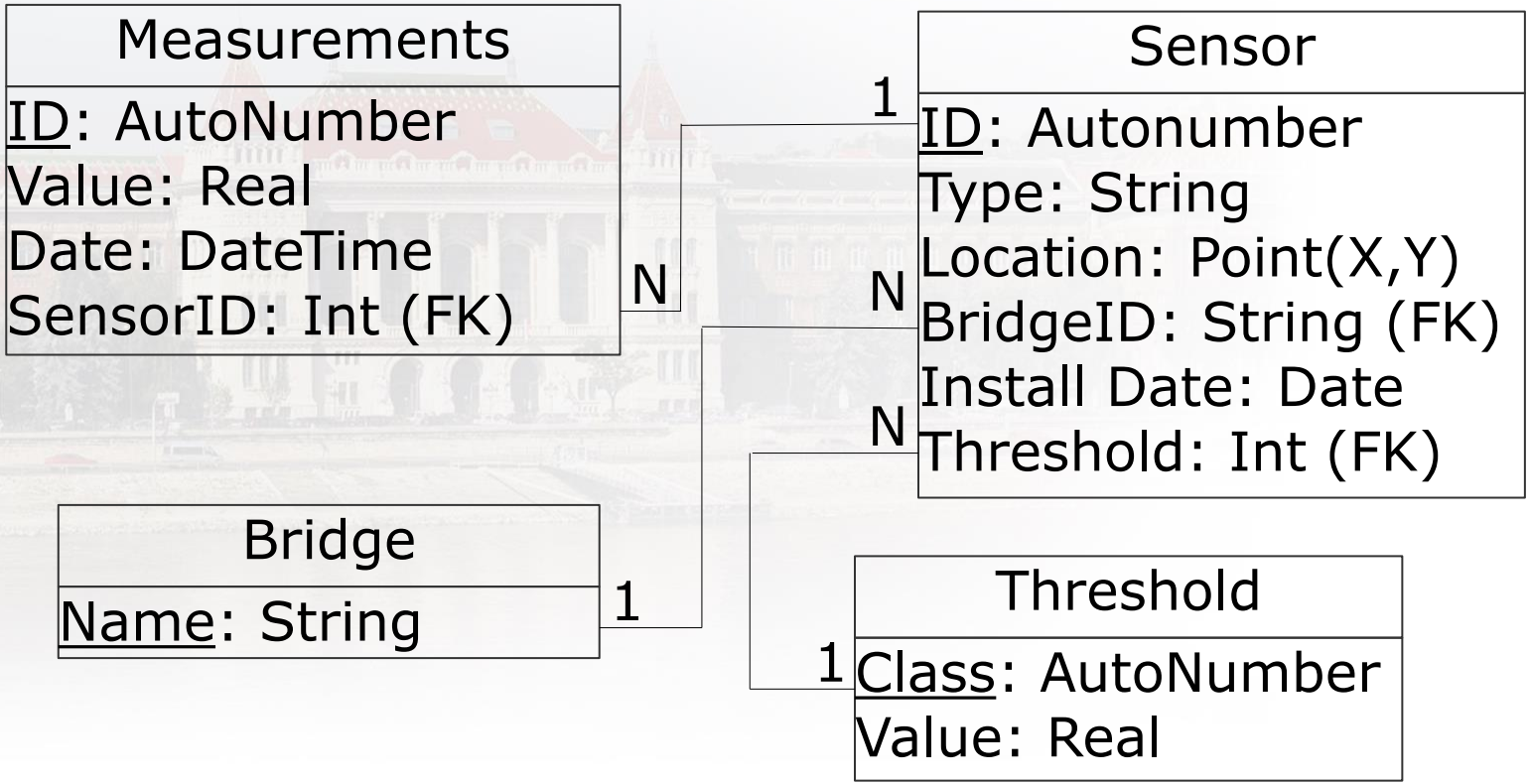
Car
PlateNumber: String
Type: String
OwnerID: Int (FK)
Date: Date

Relationship transformation (N:M)





Case study


Measurements
ID: AutoNumber
Value: Real
Date: DateTime
SensorID: Int (FK)
Sensor
ID: Autonumber
Type: String
Location: Point(X,Y)
BridgeID: String (FK)
Install Date: Date
Threshold: Int (FK)
Bridge
Name: String
Threshold
Class: AutoNumber
Value: Real
1
N
N
1
N
1

Case study

Measurements			
ID	**Value**	**Date**	**SensorID**
1	10	2009.02.10 09:10:01	1
2	11	2009.02.10 09:10:03	1
3	15	2009.02.10 09:10:06	1
4	10	2009.02.10 09:10:01	2

Sensor					
ID	**Type**	**Location**	**BridgeID**	**InstallDate**	**Threshold**
1	Strain gauge	P(100,100)	Chain	2009.01.10	1
2	Accelerometer	P(0,0)	Chain	2009.01.10	2

Híd
ID
Chain

Threshold	
Class	**Value**
1	13
2	15

Challenges of relational databases

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék

Bad table

Invalid value

(no candidate key)

Name	Firstname	Age	ID	SSN
John Doe	John	23	1	101
Alba Flores	Alba	-1	2	102
John Doe	John	23	1	101

redundancy

(if there were a key, these fields are functional dependent)

Bad table

Duplicate row

inconsistency

Name	Firstname	Age	ID	SSN
John Doe	John	23	1	101
Alba Flores	Alisa	-1	2	102
John Doe	John	23	1	101

We cannot find a person based on lastname

KEY POINTS

- Enforce consistency: DB shouldn't contain contradictory information
- Enforce data validity: Data value should be a part of an exact set of values.
- Minimizing/eliminate redundancy:
  - *Multiple storage of same data*
  - *Difficult to enforce data consistency*
- Avoid duplicated tuples (if needed/possible)
- Avoid duplicated data storage (if needed/possible)

Solutions

- Enforce consistency: normalization
- Enforce data validity: selection of proper data type
- Minimizing/eliminating redundancy: normalization
- Avoid duplicated tuples: key definition
- Avoid multiple data storage: careful designing, normalization

Redundancy, anomalies

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék

Redundancy

Redundancy is the source of following anomalies

- Update anomalies
- Insertion anomalies
- Deletion anomalies

Update anomaly

Name	Age	Item
John Doe	23	Chocolate
John Doe	23	Beer
Alba Flores	21	Beer

Once I going to update John Does name I have to update it in multiple tuples!

Insertion anomaly

Name	Age	Item
John Doe	23	Chocolate
John Doe	**22**	Beer
Alba Flores	21	Beer

Age was mistyped (typo), unable to achieve reliable information.

Deletion anomaly

Name	Age	Item
John Doe	23	Chocolate
John Doe	23	Beer
Alba Flores	21	Beer

If all rows corresponding to John Doe is deleted, his age information lost and no more available!

Avoiding redundancy

- To avoid redundancy, table structure must be normalized during DB design.
- During normalization we achieve different normalization levels and **normal form**s.
- To understand normal forms, first functional dependencies should be introduced.

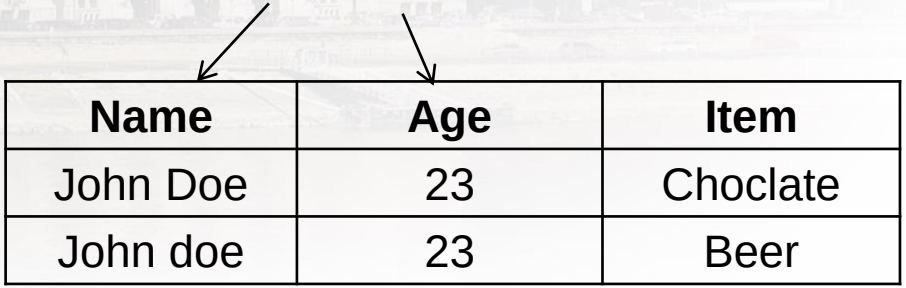
Functional dependencies

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék

Functional dependencies

If two tuples of R agree on some attributes (A), then they must also agree on other attribute, B. B functionally depends on A.



Name	Age	Item
John Doe	23	Choclate
John doe	23	Beer

Name → Age

Syntax

Functional dependency – Candidate key

- Note: functional dependency is a generalization of candidate key:
- If one or more attribute functionally defines **all** other **attributes**, than those attributes are superkey.
- If this set is minimal, than it's a candidate key.

Functional dependency - Example

Give the set of functional dependencies!

- Employees(IDNr, Name, Salary, Position)
  - *IDNr. → Name, Salary, Position*

Functional dependency - Example

Give the set of functional dependencies!

- Employees(IDNr, Name, Salary, Position)
  - *IDNr. → Name, Salary, Position*
  - *Position → Salary (?)*

Functional dependency - Example

Give the set of functional dependencies!

- Employees(IDNr, Name, Salary, Position)
  - *IDNr. → Name, Salary, Position*
  - *Position → Salary (?)*
- Results(ClassName, NeptunID, Name, Grade)

Functional dependency - Example

Give the set of functional dependencies!

- Employees(IDNr, Name, Salary, Position)
  - *IDNr. → Name, Salary, Position*
  - *Position → Salary (?)*
- Results(ClassName, NeptunID, Name, Grade)
  - *ClassName, NeptunID → Grade*

Additional terms

- *Full functional dependency: all non-key attributes are fully functionally dependent on the primary key*
- *Transitive dependency: in a database is an indirect relationship between attributes in the same table that causes a functional dependency. (All non-key attributes functionally depend on key, but there is one or more non-key attribute that functionally depends on a non-key attribute)*

Normal forms

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék

First normal form (1NF)

- No rows may be duplicated (has candidate key)
  - *Has a key and all other attribute functionally depends on that.*
  - *Key can be all attribute of the relation*
- Column number and order are the same for all tuples
- Each column may contain only one attribute
  - *All attribute have single value*
  - *It's not trivial → question of DB design*

Second normal form (2NF)

- Definition of 2NF:
  - *fulfilled to 1NF +*
  - *All not candidate key fully functional depending on candidate key and not depends on any subset of candidate key*
  - *Full functional dependency: not functionally dependent on any proper subset of attributes*
- If candidate key is a single attribute, it's OK

Decomposition (2NF)

Neptun	ID.	TestNr	Result
ABCD1	100AA	1	4
ABCD2	101AA	1	5
ABCD1	100AA	2	3

Neptun	ID
ABCD1	100AA
ABCD2	101AA

Neptun	TestNr	Result
ABCD1	1	4
ABCD2	1	5
ABCD1	2	3

Third normal form (3NF)

- Definition of 3NF:
  - *Fulfilled to 2NF +*
  - *All attribute that excluded from candidate key full functionally depends on candidate key and only on that. There is no transitional dependency.*

Decomposition (3NF)

TransactionD	Item	Price
1	Chocolate	100
2	Beer	300
3	Chocolate	100
4	Chocolate	100
5	Beer	300

TransactionID	Item
1	Chocolate
2	Beer
3	Chocolate
4	Chocolate
5	Beer

Product	Price
Chocolate	100
Beer	300

Higher level of normalizations

- In practice 3NF is generally satisfactory
- 3NF ensures in most cases redundancy free DB
- BCNF, 4NF, 5NF
- Cons of higher levels:
  - *Fragmented table schema*
  - *Complicated queries*
  - *Reduces the efficiency of queries*

In practice

To increase efficiency normalization is not strictly performed in some cases.

Conclusion

- Workflow of design
- Specification
- Keys
- E/R diagram
- Relational scheme diagram
- Challenges of relational databases
- Redundancy, anomalies
- Normalizing, decomposition

Thank you for your attention!

Questions?

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék