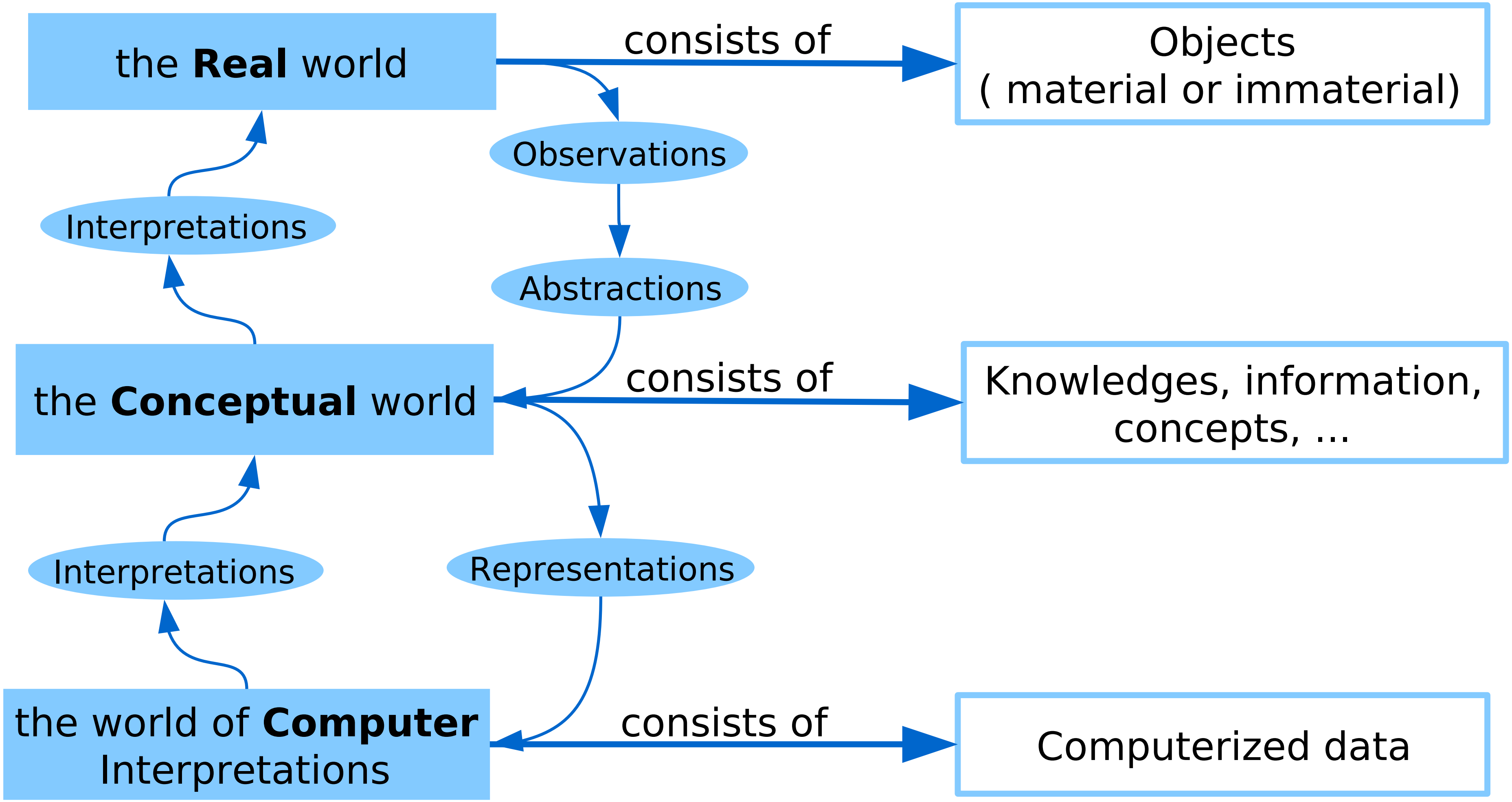
Introduction
to
DATABASES

1. Introduction to Databases and DBMS
DATA AND DATABASES

The three Worlds



the **Real** world

consists of

Objects
(material or immaterial)

Observations

Abstractions

Interpretations

the **Conceptual** world

consists of

Knowledges, information, concepts, …

Representations

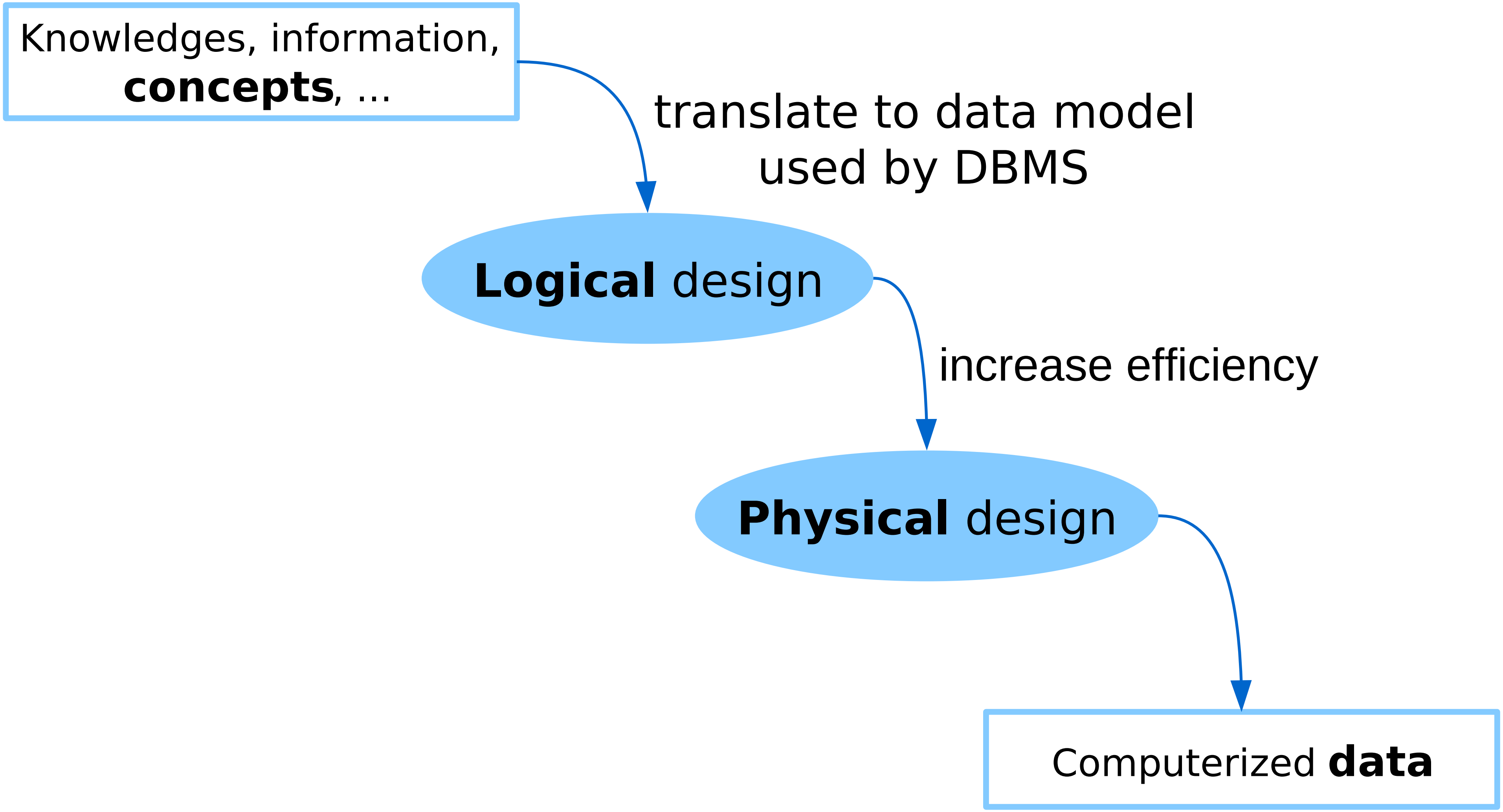
Interpretations

the world of **Computer** Interpretations

consists of

Computerized data

1. Introduction to Databases and DBMS
DATA AND DATABASES



Knowledges, information, **concepts**, ...

translate to data model used by DBMS

Logical design

increase efficiency

Physical design

Computerized **data**

1. Introduction to Databases and DBMS

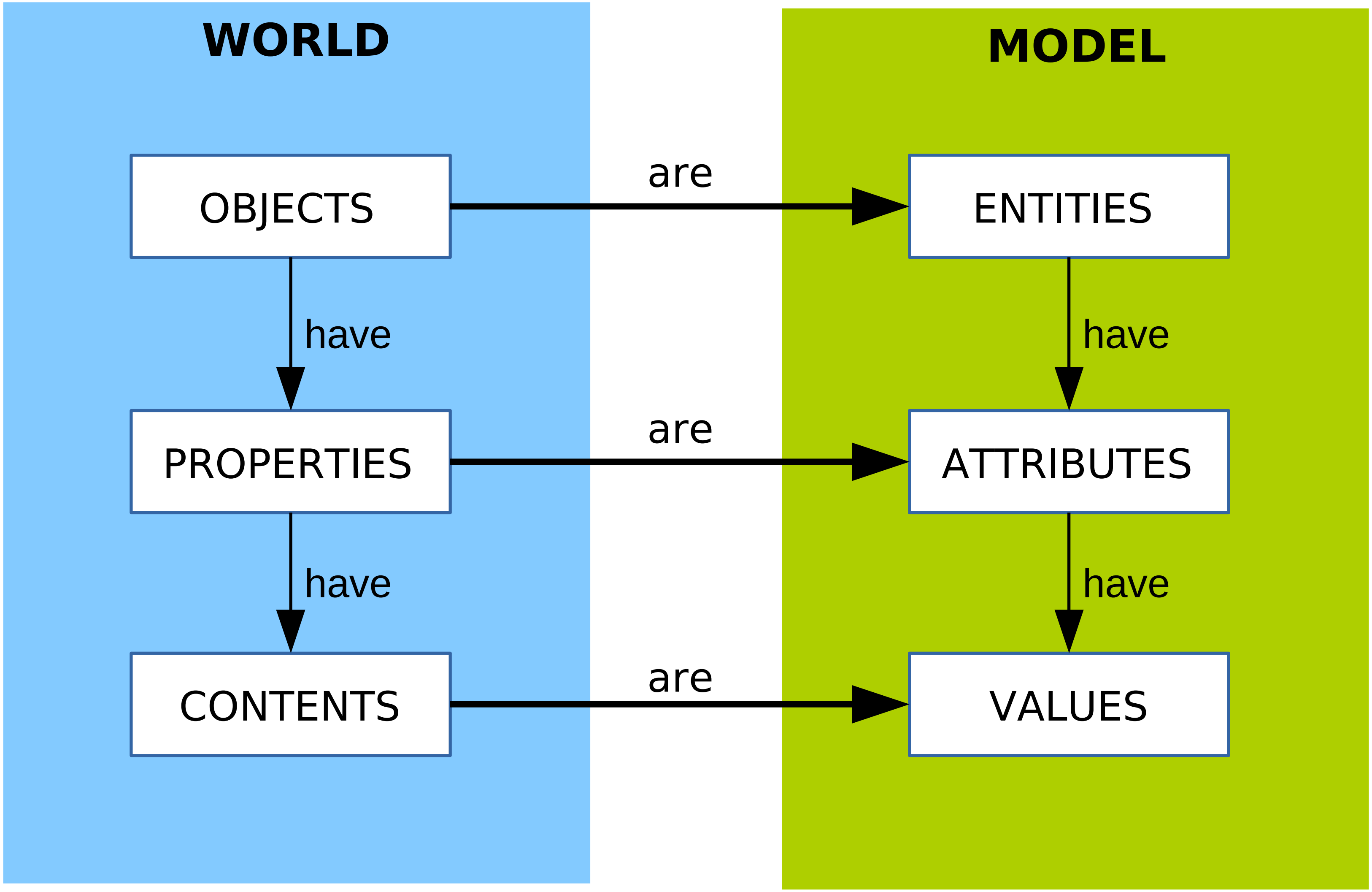
DATA AND DATABASES

From the same **data model** we can get different physical designs (with different levels of efficiency) depending on:

- Decisions about designs
- technology used (files, relational DB, distributed DB,..)

In any case, the computer expert is responsible for the correct representations and efficient implementations.

1. Introduction to Databases and DBMS
DATA AND DATABASES



IMPORTANT: If we only know two of these three elements, we are not going to have a real information

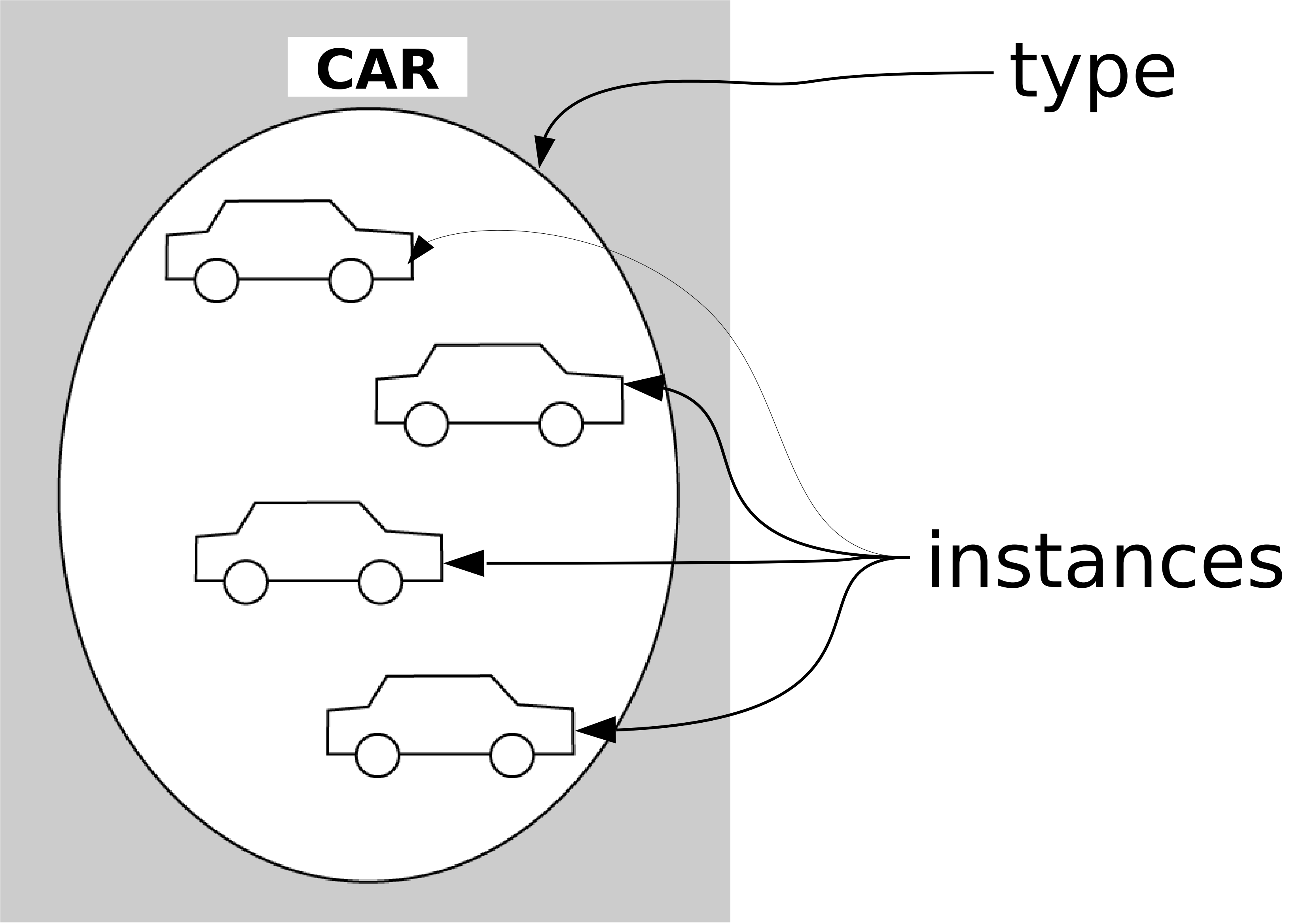
1. Introduction to Databases and DBMS
DATA AND DATABASES

- **ENTITY TYPE**: A generic type, an abstraction, a class of things. Ex: car, film, book, ...

- **ENTITY INSTANCE**: specific object of the real world, such as a car, that we can distinguish from other cars, thanks to some property (as might be the value of Registration attribute).

1. Introduction to Databases and DBMS

DATA AND DATABASES



Data type vs Domain

- A data type defines
  - ✓ a set of **values**,
  - ✓ with common characteristics → **compatible**,
  - ✓ a set of permissible **operations**

- A domain is a subset of possible values of a data type (defining the **range of values** supported)

1. Introduction to Databases and DBMS
DATA AND DATABASES

The expression **null** indicates no value associated with a given attribute of an entity instance. This empty value is:

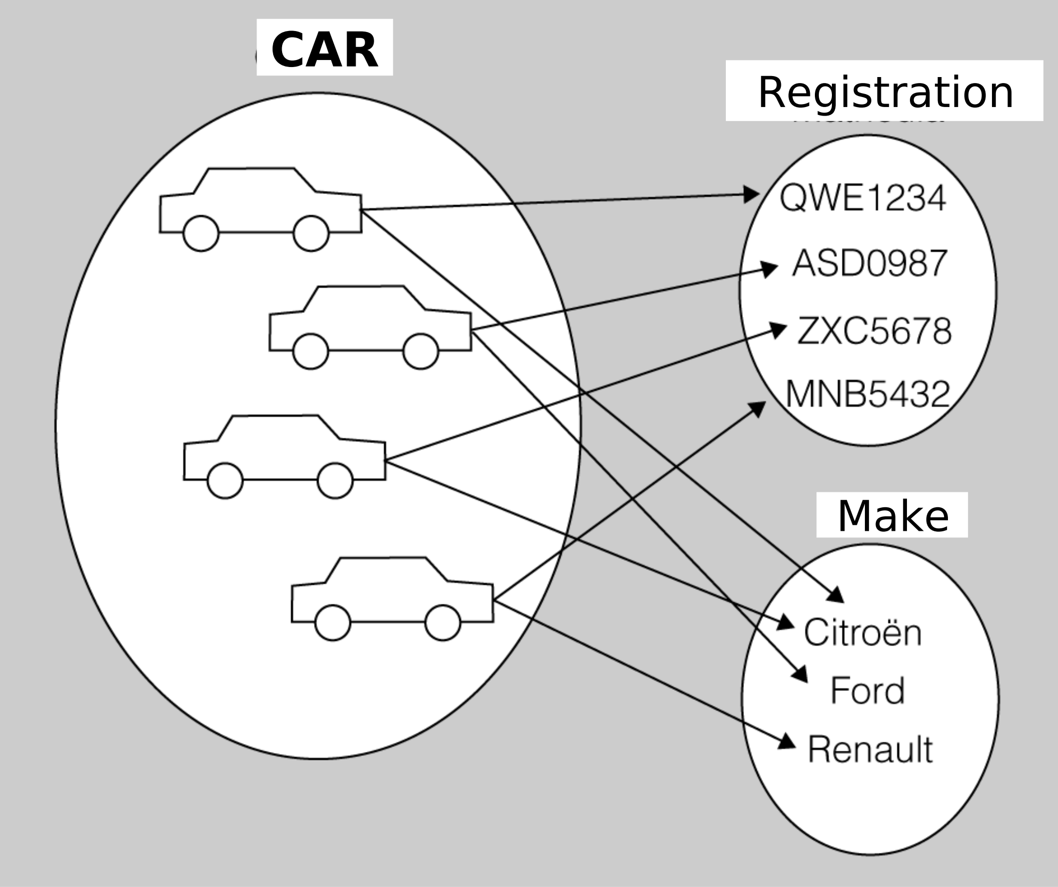
- unknown (or not exists)
- different from zero
- different from blank space

1. Introduction to Databases and DBMS
DATA AND DATABASES

- An attribute **identifier** allows us to distinguish each entity instance from the rest, because its value is unique and not repeated among all entities instance.
- An attribute identifier or a set of attributes that identify unequivocally the instances of an entity are called **keys**.
- By definition, attributes that make up a key can never admit a null.

1. Introduction to Databases and DBMS
DATA AND DATABASES

A graphical representation, not computerized, of entity CARS, which consists of two attributes: Registration and Make.



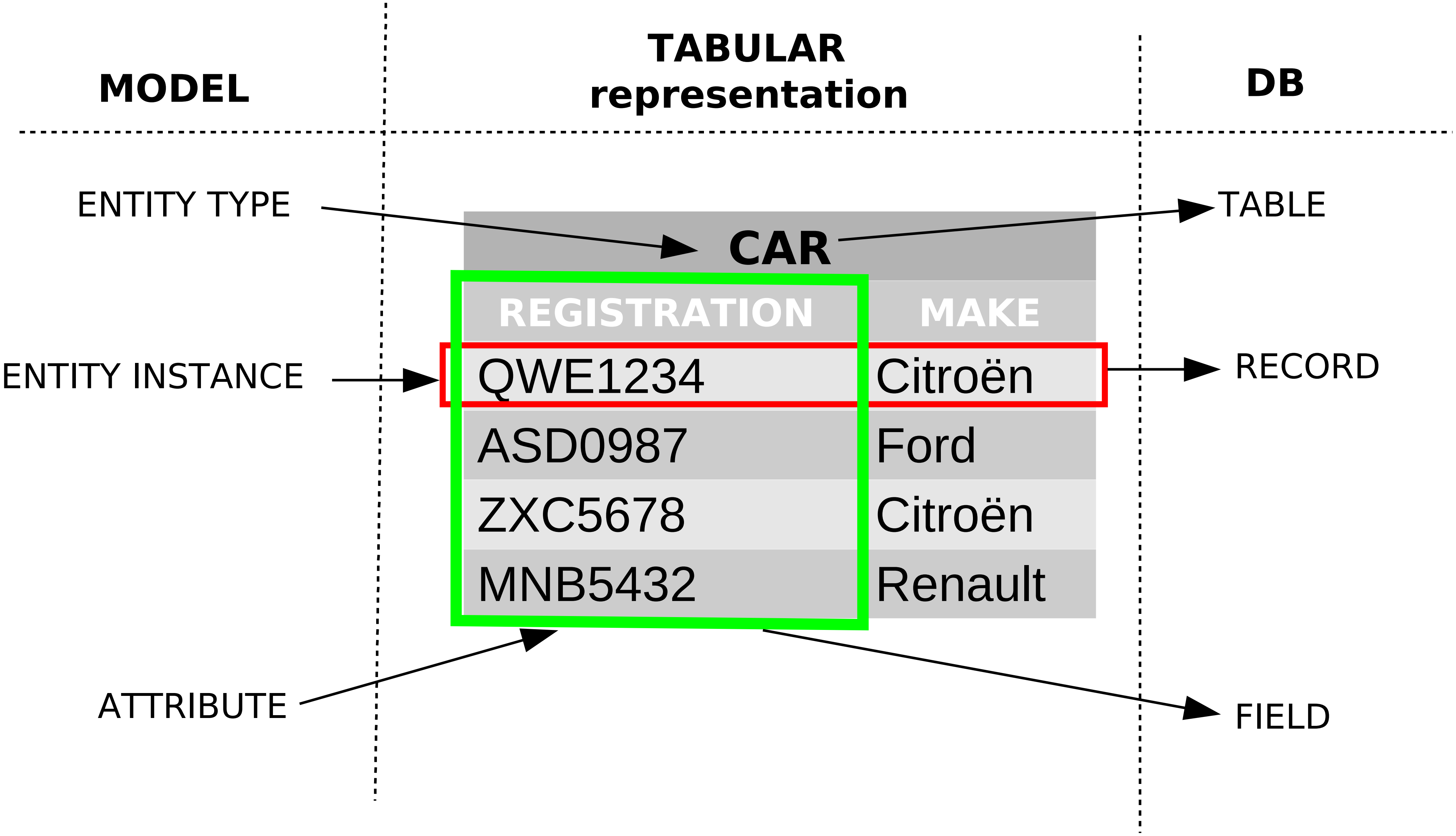
1. Introduction to Databases and DBMS

DATA AND DATABASES

The representation more frequent in the field of DB is the **tabular** representation (as a table or list), where files are structured in records and fields.

- Each row represents an entity instance.
- Each column represents an attribute.
- Each cell (ie each intersection of a row and a column) stores the value that has the attribute of the corresponding instance.

1. Introduction to Databases and DBMS
DATA AND DATABASES



CAR

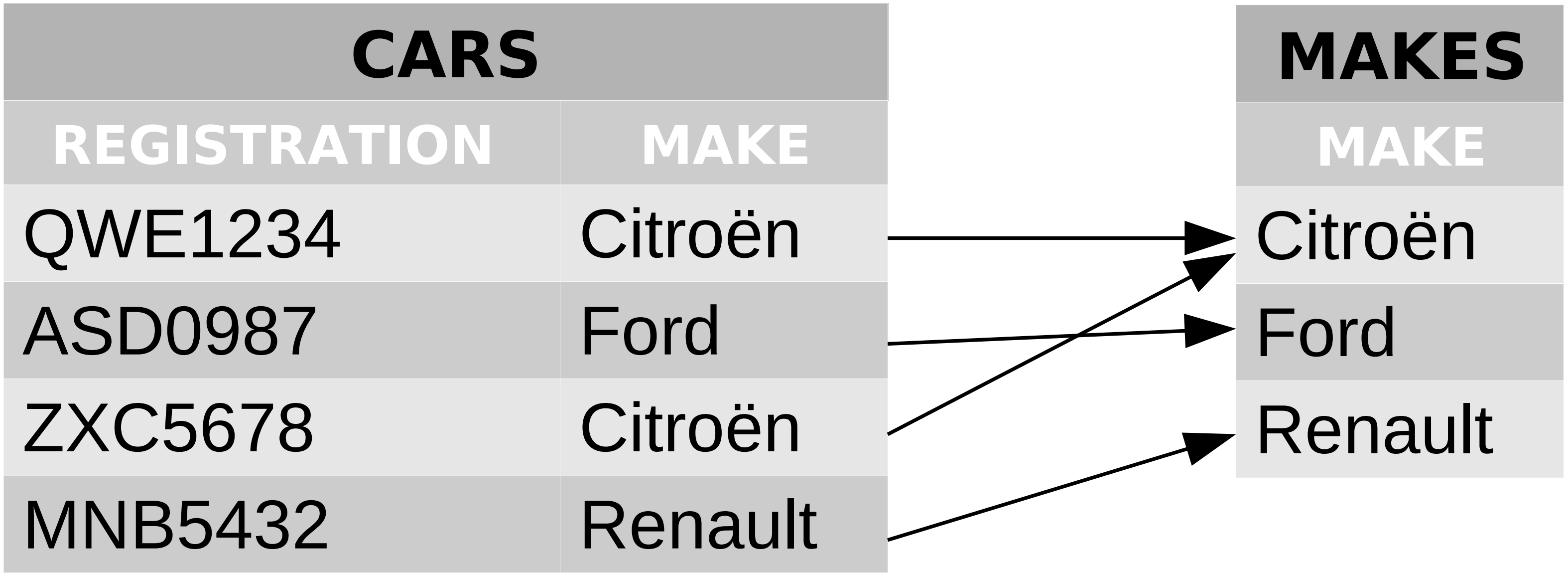
REGISTRATION	MAKE
QWE1234	Citroën
ASD0987	Ford
ZXC5678	Citroën
MNB5432	Renault

1. Introduction to Databases and DBMS
DATA AND DATABASES

- Normally when we represent the conceptual world, we will find more than ONE entity. And these entities, tables, or files will not be unconnected objects, they must be **interrelated**.

- The relationships among records of two (or more) tables are made using fields of the same data type that store the same values.

1. Introduction to Databases and DBMS
DATA AND DATABASES



CARS	
REGISTRATION	**MAKE**
QWE1234	Citroën
ASD0987	Ford
ZXC5678	Citroën
MNB5432	Renault

MAKES
MAKE
Citroën
Ford
Renault

PROBLEMS: Updating these relationships is no so easy
Consistency of data is complex

SOLUTION: **DB & DBMS**

1. Introduction to Databases and DBMS
CONCEPTS ABOUT FILES AND DATABASES

	FILES	DATABASES
Type entities	Only one entity	Many entities
Relationships	None	Relations are possible
Redundancies	Specific file for each application	All applications work with the same db
Inconsistencies	The same data don't match in different files	No redundancies → no inconsistencies
Data collection	Every time through a new program	You can get data directly from work environment and from programs
Isolation data	Data are scattered and isolated in different files	All data are in the same db, interconnected

1. Introduction to Databases and DBMS

CONCEPTS ABOUT FILES AND DATABASES

	FILES	DATABASES
Data integrity	The programs must implement all restrictions on the data	DB directly implements restrictions on the data.
Atomicity	Very difficult to implement	Transactions
Concurrent access	Very easy to get inconsistent data	Data locking
Security	Acces to ALL data of a file	Users can have different profiles with VIEWS and ACCES RIGHTS on some specific data of databases

1. Introduction to Databases and DBMS
CONCEPTS ABOUT FILES AND DATABASES

When should I use files?

- low volume of data
- only one instance entity

Where?

- Application configuration files
- Systems configuration files
- Files of events (logs)