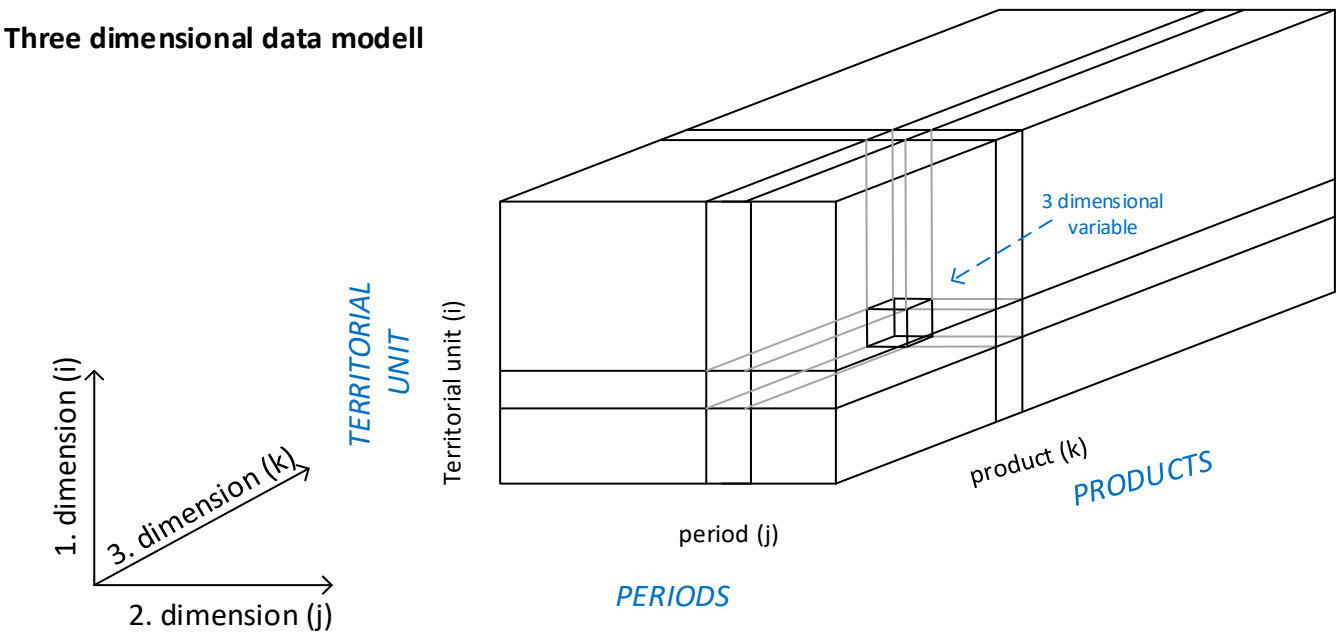
Relational data model, database planning

Types of data model

Relational data model [emphasis on the attributes] – today almost dominant. Way of storage: data tables. One table contains attributes of one type of entities. The data model consists of tables related to each other.

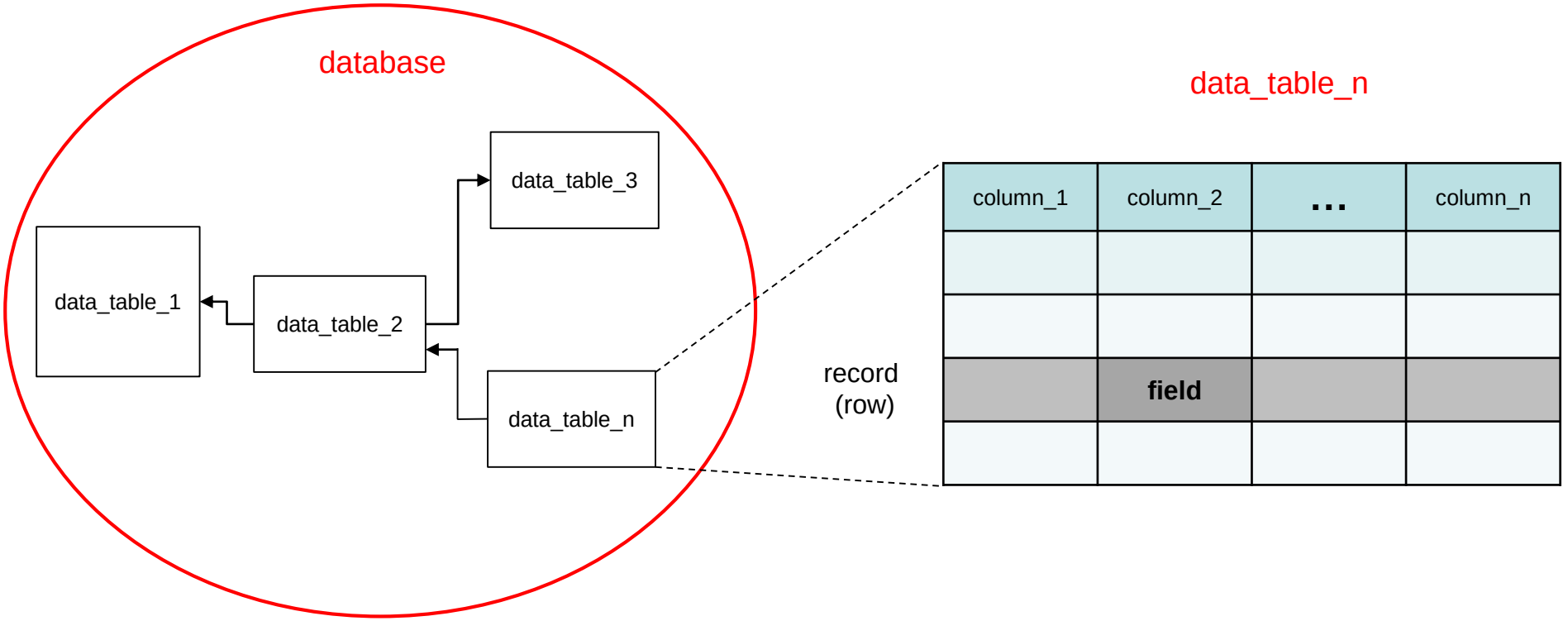
Multidimensional data model – it has been developed from relational data model; it will be spread in the future. The data model focuses on the type of processes and quick analyses (method of **slicing**). The data elements are stored in multidimensional variables. Oracle company develops this kind of software.



Relational data model

Concepts applied in the relational data model

database term	datamodel term
database	connected data tables
data_table	entity type
column	attribute
record (row)	entity
data element (field)	1 attribute of 1 entity



1. definition of structure of data table - specification of columns:
name, type, length, constrains regarding data elements, error message

identifying, descriptive attributes

key: column(s) for unambiguous identification of entities (records)

simple key — e.g. number of the identity card

multiple key — e.g. name, address, date of birth

minimal key (it is wanted) — e.g. previous one

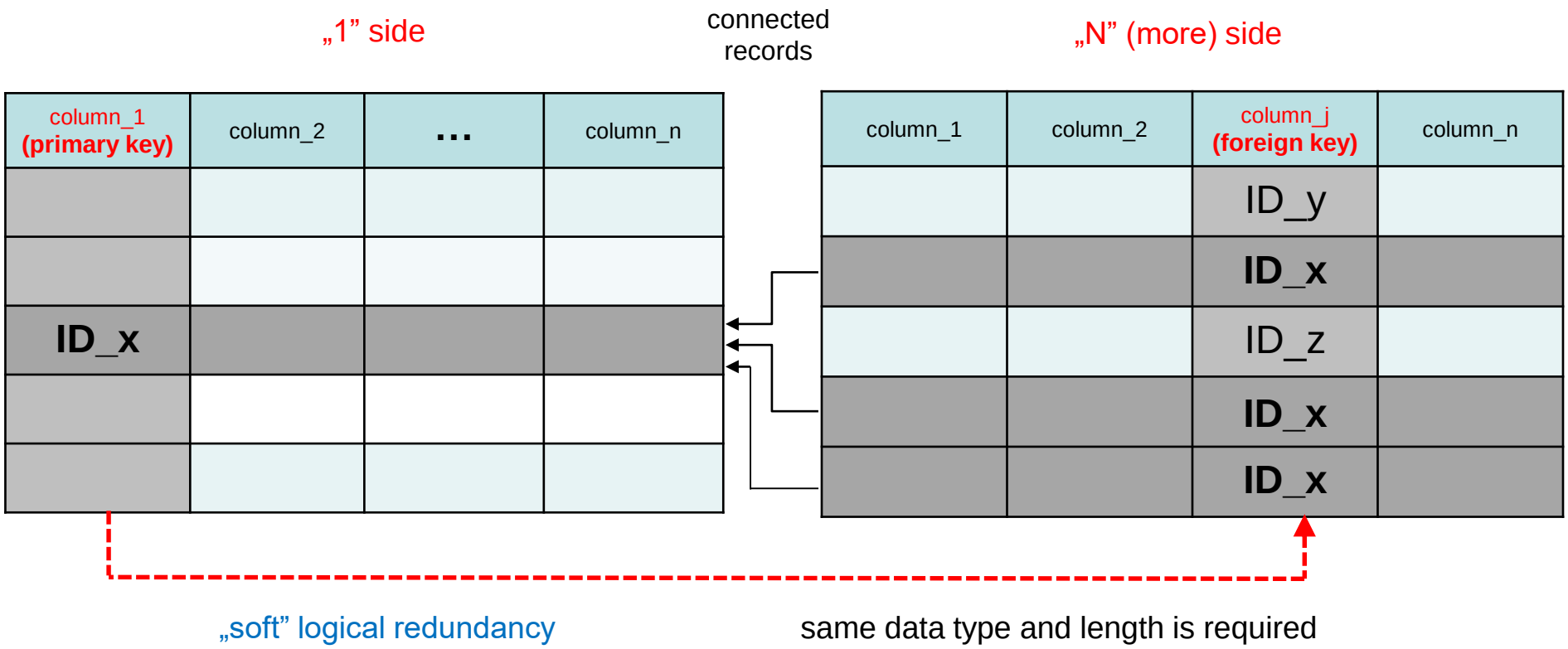
super key (more than enough attributes, it should be avoided) — e.g. name, address, date of birth, number of driving license

primary key – the key, that used at the certain process
1. NOT NULL (obligatory filling in)
2. UNIQUE (values must not be repeated in this column)

foreign key – its aim: to define *relation* between two tables; the relation is realized through the values (data elements). It is not key in the given table (in the connected table; ∞ side), but is primary key in another table (in the connector table; 1 side).

primary attribute – participates at least in one key
secondary attribute – does not participate in any key

2. create connections between data tables (1:N connection type)



META DATA: „Data referring to data”. It describes storage characteristics and structure of “useful” datasets. Range of metadata depends on database management system.

For example, in the case of relational databases, the names of the tables, the names of the attributes and the types of data regarding a certain attribute belong to set of metadata

Normalisation of databases

Main question: When is the structure of the database **appropriate?**

Normalization of data tables (operations to disaggregate tables): our aim is to store data with as few redundancy as possible.

To create relations soft logical redundancy is necessary. Anomalies caused by redundancy.

Definition of normal forms

1 NF (first normal form):

- number and sequence of the columns are the same in each row,
- there is not two alike rows; unique key belongs to each row, and all the other fields have functional dependency on it,
- one and only value belongs to every attribute.

Functional dependency: if in one column (column1) at two records (entities) there are the same values, then in another column (column2) the values belonging to these records (entities) are also identical.

Example: register of employees

Are there any functional dependencies between the attributes of **company name (column 1) and registered office address (column 2)** if yes, in which direction?

ID	company name (column 1)	registered office address (column 2)
1	Tudás Bt.	Egyetem u. 3.
2	Szorgalom Kft.	Iskola tér 5.
3	Tudás Bt.	Egyetem u. 3.
4	Kitartás Rt.	Egyetem u. 3.
5	Szorgalom Kft.	Iskola tér 5.
6	Kitartás Rt.	Egyetem u. 3.

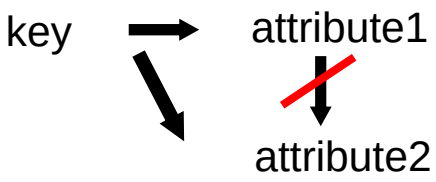
column 1 → **column2**

column2 ↛ **column1**

2 NF (second normal form): if the data table is in first normal form and the non-key fields have entire functional dependency on the multiple primary key and not only on its some parts [We do examinations only in the case of multiple primary keys].

Solution: the fields that have functional dependency on not the entire multiple primary key are relocated into a new table, where the partial key will be the primary key.

3 NF (third normal form): if the data table is in second normal form and dependency originated only from the primary key. There is not dependency among the secondary attributes, namely there is not transitive dependency in the table.



Solution: the secondary attributes depending on each other are relocated into a new table, where the attribute1 is designated to primary key.

More normal forms are used too.