Introduction to Operating Systems and SQL for Data Science

Lecture 7 – Relational/tabular model

Previous lecture

- Database management system (DBMS)
- Operations on DBMS
- DBMS environment
- ACID - transactions
- Operation types in DBMS (DDL/DML)

Data model

Terms and symbols are used to define the structure of a database.

The structure includes:

- Data types.
- Relationships between the data.
- Constraints that applied to the data and contexts.
- Basic operations on the data.

Sometimes the model is depicted in the form of text, and sometimes in the form of a diagram.

Reichman University

Data model categories

- **Physical models**: the database is defined in "physical" terms of organizing and processing files. Define how the data is stored, how to search, and how to update the data. (E.g., the technique of handling collisions, indexes, etc.)

- **Logical models**: The database is defined in a non-physical terms; closer to the world of users. (Such as: what are the files, records, fields, keys, types of connections between the files)

- **Conceptual models**: The database is defined in very general terms, not "physical," but as they are in reality and understood by most users. (E.g., entities; attributes; relationships). They are used to plan the database schema. The model is presented in the form of a diagram.

Reichman University

Physical models

- Uses computer / physical terms that describe how data is stored in the information system.

  For example, File management systems (FMS) apply techniques such as linked lists, B-Tree, Hashing, and Inverted Files.

- The definition of the database and the handling of the data are done using a procedural programming language or a similar language of the system.

- They are considered "unfriendly" models and outdated.

Reichman University

Logical (applied) models

- Defines the data structure and handle data at a level of abstraction "higher" than the physical level.

- Most of the DBMS are related to the logical models. The top logical models are:
  - Hierarchical
  - Network
  - Relational

  Some say that those are physical models

- The data is defined in terms of file names (tables), records, fields, keys, sets.

- The system includes a language for defining the data (DDL) and a language(s) for handling the data (DML).

Reichman University

Conceptual models

- Uses general terms on the data, as they are understood by most users, rather than computing terms.

- Use terms such as: entities, attributes and relationships.
  - **Entity** - represents a thing or object from the real world. Examples: student, employee, course. There are tangible and intangible entities.

  - **Attribute** - describes an entity. Examples: ID number, name, date of birth. There are features of different types.

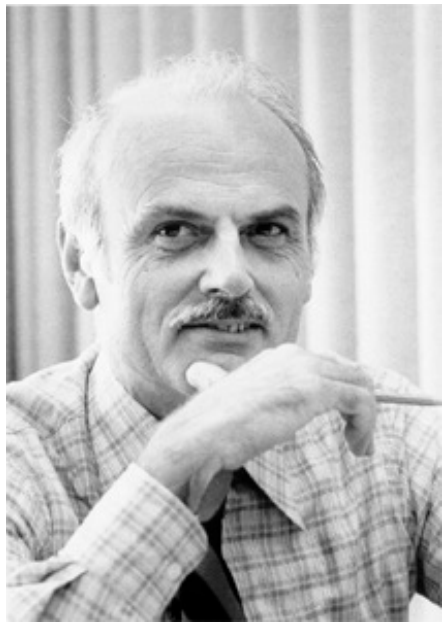
  - **Relationship** - between two or more entities that have a relationship. For example a connection between a student and the department in which he is studying. There are different types of relationships between entities.
- Typically these are models for designing the database schema (DDL) but not for DBMS implementation.

Reichman University

Relational data model

Relational Data Model

- This is a logical data model with a mathematical basis which was developed by Codd in 1972.
- Most DBMS use it.
- Usually DMBS that use it often called – RDBMS.



Edgar "Ted" Codd

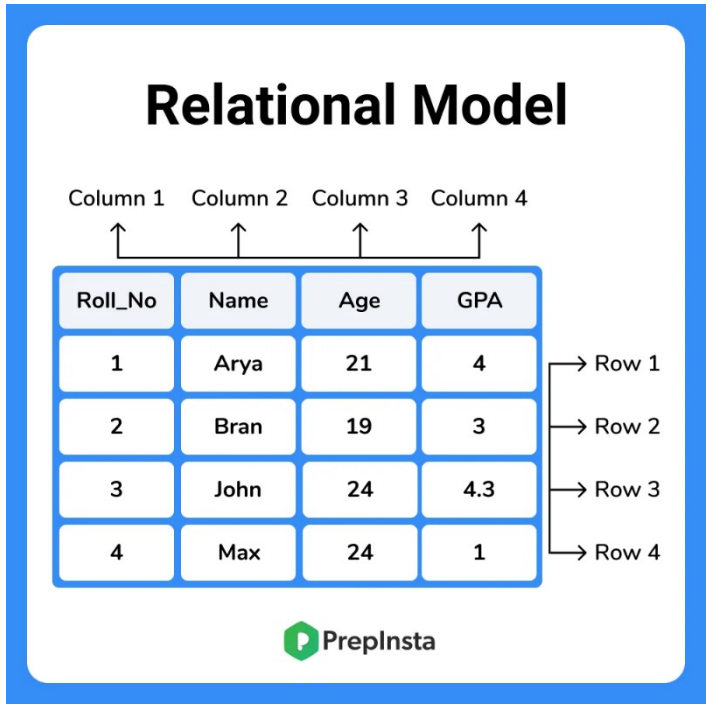
3 Relational Data Model Examples

Relational Database Design — What does it mean?

Reichman University

Relation

- The basic component of the tabular/relational model is a two-dimensional table called a Table or Relation.



https://prepinsta.com/dbms/relational-data-model/

Reichman University

Relational components

Key

Attribute names

ID	Name	GPA
S4	Alice	4.0
S2	Bob	3.5
s1	Charlie	3.8

Tuples

Attribute values

Reichman University

Tuple

Tuple is a specific row in a table. Marked as:

$t_i = <t_{i,1}, t_{i,2}, \ldots, t_{i,n}>$

For example:

$t_1 = <s4, Alice, 4.0>$

Reichman University

Scheme definition

- Each table has a name, in this case: Students
- The table has Attributes.
- The name of the table and its attributes together define the schema of the relation or table (Relation Schema), and the marking of this is:
  - R (a1,…., An)

Example in this case:

Students (FirstName, LastName, **<u>ID</u>**, YearOfBirth, City)

The key is marked by an underline.

Reichman University

Attribute value type

- We will define each attribute in the table its type and domain of values (Domain).

- Common types:
  - Integer
  - Decimal
  - Character
  - Boolean
  - VARCHAR (String)

Reichman University

Example for defining attributes

Age = {Integer; Age>0 and Age<120}

First_Name = {Character (20)}

Reichman University

Keys

Keys

Key - one (or more) fields which allows the distinction between records and the unambiguous identification of each record in relation.

We distinguish between the different keys:

Super Key - A subset of fields (non-minimal) that identify a record.

Candidate Key - A subset of fields (minimum) that identify a record

Primary Key - One of the candidate keys.

Alternate Key - The other candidate keys.

Foreign Key - A field (fields) that refers to the key (Primary Key) of another relation (base relation). That is: the field value exists in the base relational key. (foreign key name does not have to be the same as the corresponding primary key name but must be from the same domain.)

A key should be minimal; meaning we can’t eliminate fields. A key must contain a value (not "empty") - not null.

Various Types of Key in Relational DBMS

Reichman University

Integrity constraints

They were designed to ensure data integrity within and between relationships.

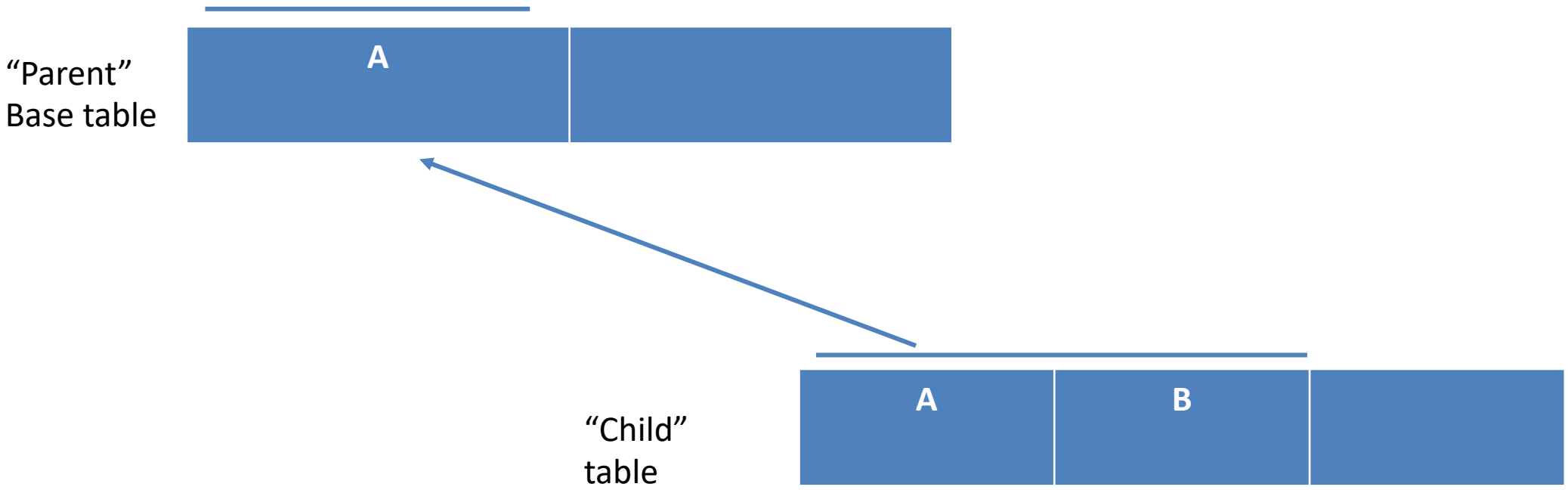
- **Key constraint** – a relation has a key.
- **Entity Integrity constraint** - The Primary Key must contain a value (not null).
- **Referential Integrity constraint** - Each value of a "foreign" key in a table must have the same value in its base table key or Null.

Foreign Key and Primary Key are defined in the same domain. Foreign Key values are a subset of the Primary Key values.

Reichman University

Three options for a Foreign key

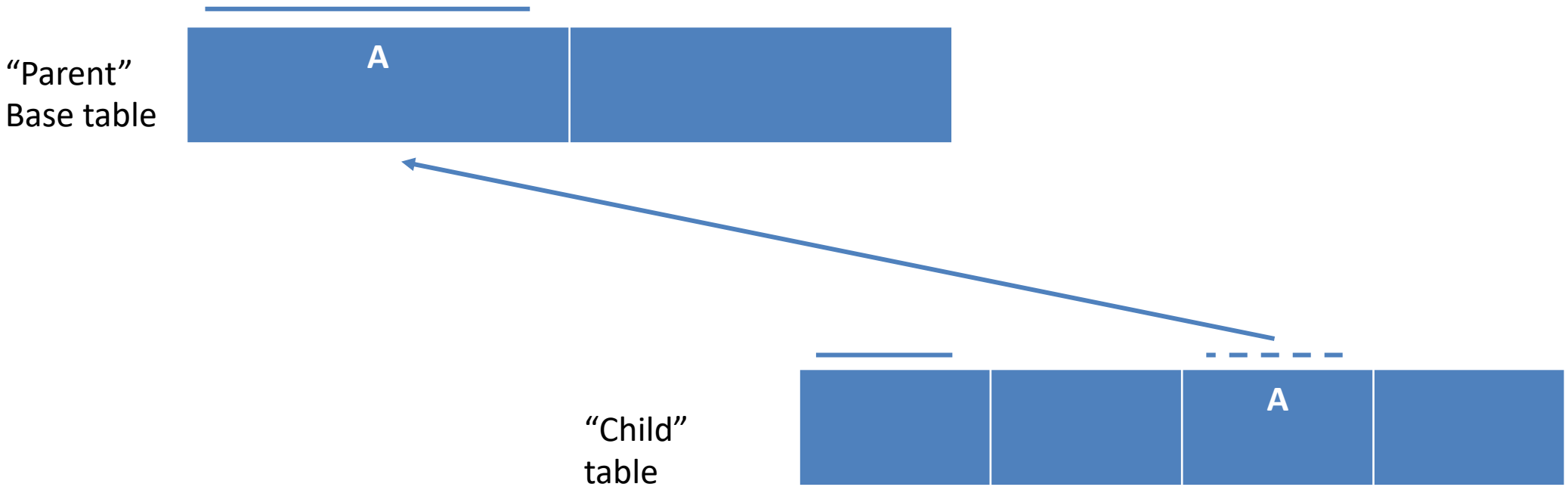
A) A foreign key can be part of a table key



Reichman University

Three options for a Foreign key

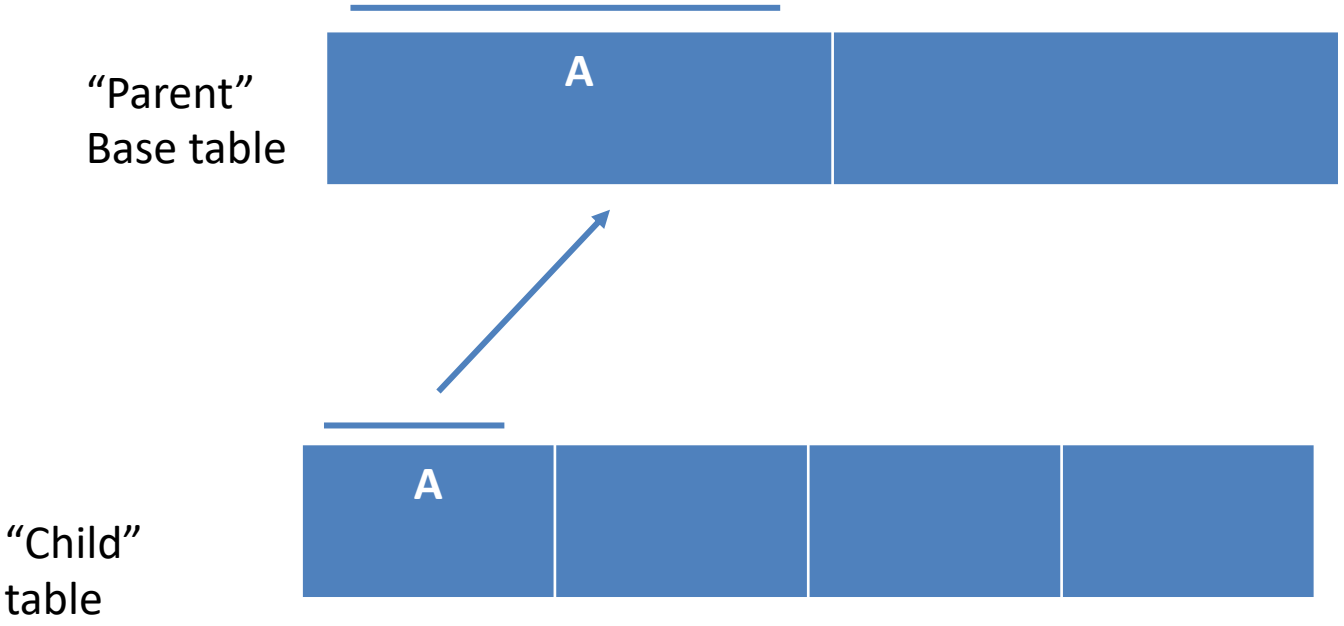
B) A foreign key can be a field that is not part of a table key



Reichman University

Three options for a Foreign key

C) A foreign key can itself be a table key:



This is a 1:1 relation between the "Parent" table and "Child" table

Reichman University

Foreign key: add a record

- If you want to add an entry to any table, its key value must be new. That is, a new record and not a duplicate – regardless of foreign keys.

- If you want to add an entry to a ‘child' table, its foreign-key value must be present in its 'parent' (base) table or be null (provided it is not part of the key)

- For example:

Departments (dept-name, location, emps_count)

Employees (emp-ID, emp-name, department)

Reichman University

Foreign key: delete a record

- If we want to delete a record in the ‘Parent’ (base) table, and its has matching records in the ‘Child’ table (they have the same value in "Foreign Key") raises the problem of completeness.
- If we delete a department what happened to all the department employees

- Possible solutions - by defining a suitable constraints:
  - Restricted - Do not allow delete actions as long as there are records in the ‘Child’ tables.
  - Nullify - Allow delete actions and reset the "foreign key" values - provided it is not part of a master key in the table.
  - Cascade - enable delete actions + deletes of corresponding ‘Child’ records.
  - Set default - Allow cancellation and set a "default" value to a "foreign key". The desired option is defined for the FK field in the ‘Child’ table.

Reichman University

Foreign key: update a record

- If you want to change the value of a primary key in the ‘Parent’ (base) table (a **rare** operation) - a similar Referential Integrity problem is created.

- Possible solutions:
  - Restricted - Do not allow key change before canceling/changing the FK in the corresponding records in the ‘Child’ table.
  - Nullify - Allow change, but reset the foreign-key values in the child records, only provided it is not part of a primary key in the same table.
  - Cascade - Enable changing the base record + ‘Child’ records of the same value in a foreign key field.
  - Set default - Allow the key to be changed and set a "default" value for the FK.

Reichman University

Other types of constraints

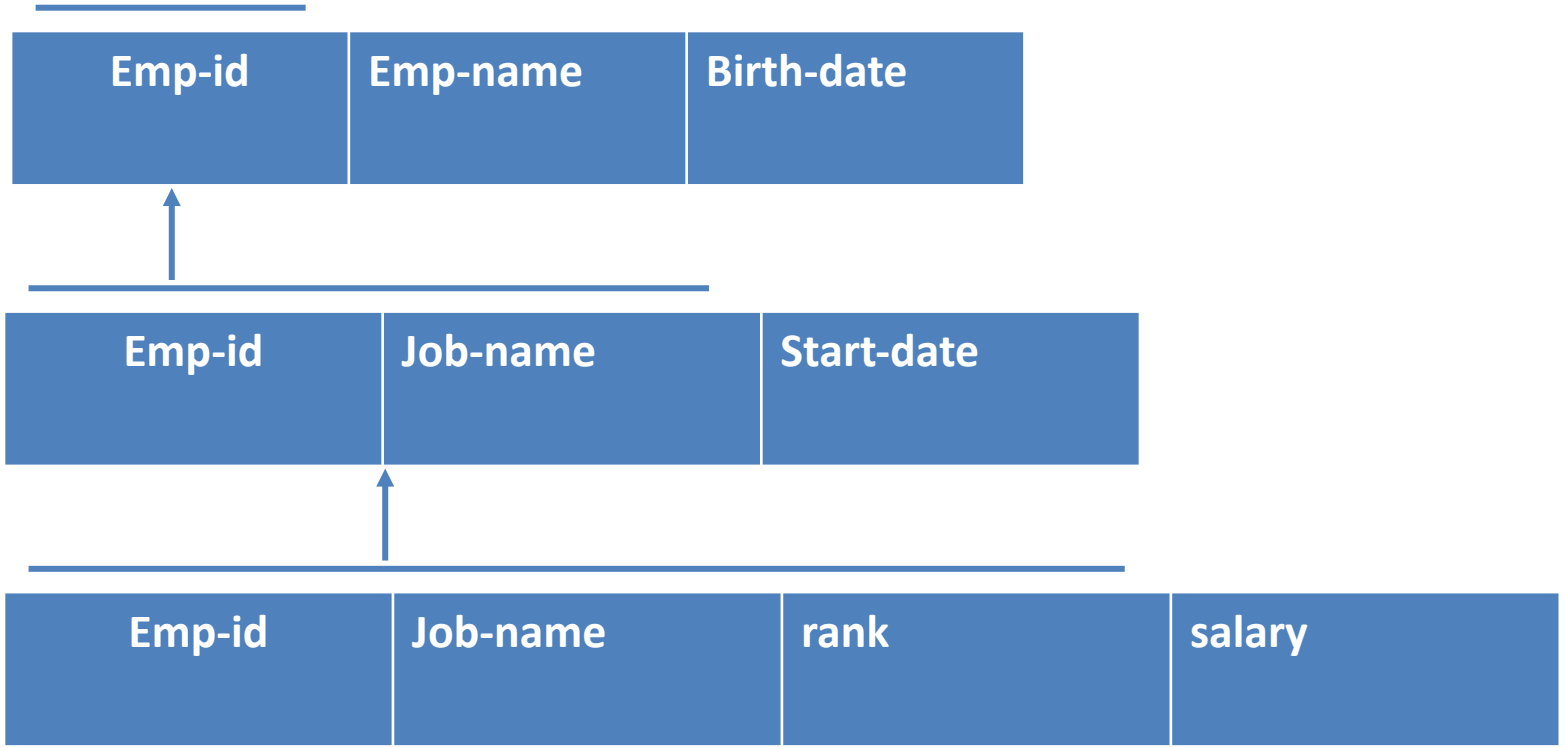
- Semantic integrity constraints
- Constraints on field values. Such as, a field value in one record cannot be larger/smaller than the value of the same field in another record. A specific field value cannot be larger/smaller than a specific size.
- Such constraints can be enforced through implementation plans or Constraint Specification Language commands; Such commands are also called: assertions or triggers.
- Due to time constraints, we will not learn triggers. But there are widely used. For example, if we want to do something when we add a new entity we can use triggers.

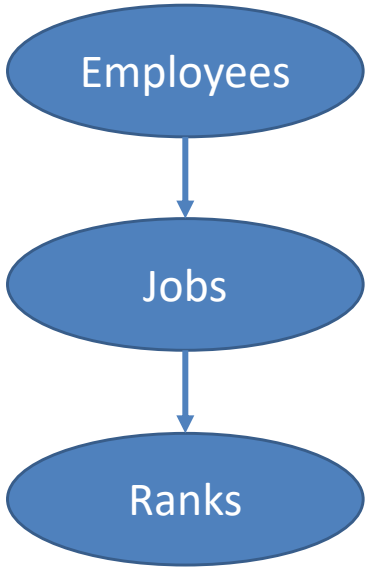
Reichman University

Data structure representation in RDBMS

Tree structure – n:1

We can describe this structure by:



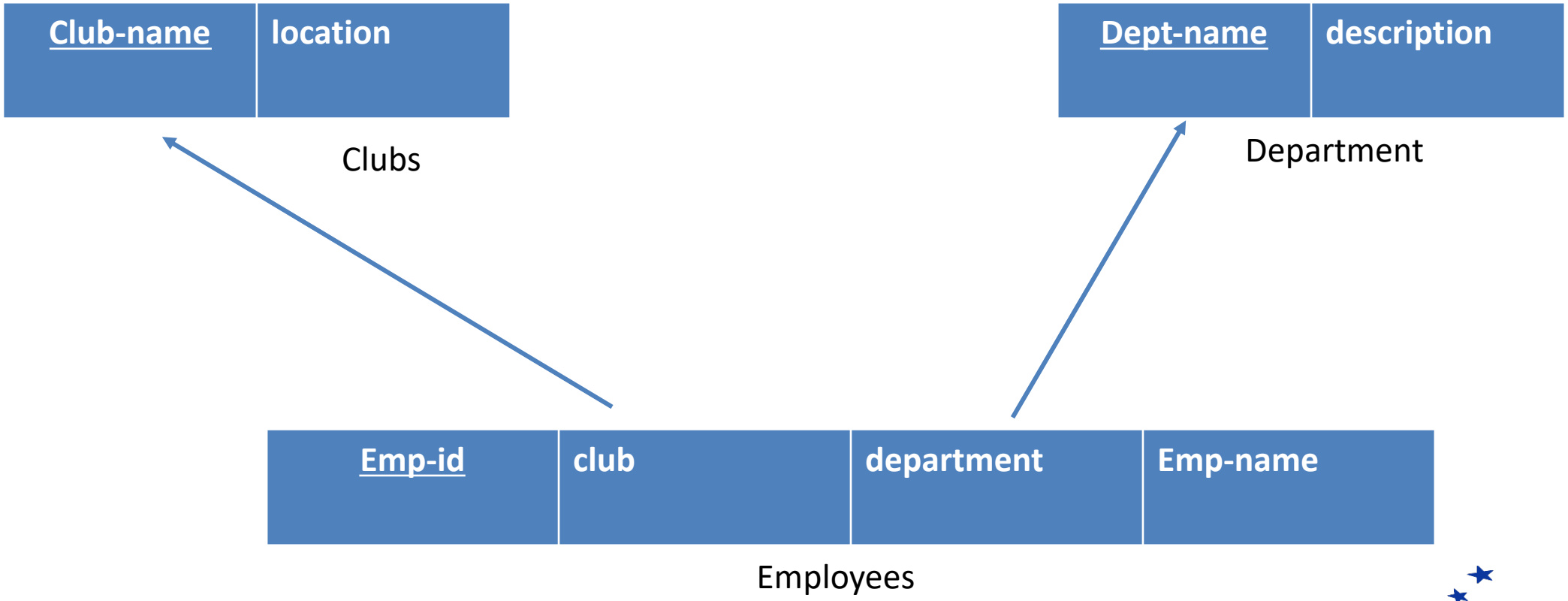


Reichman University

Simple network

An employee belongs to one club and is subject to one department:

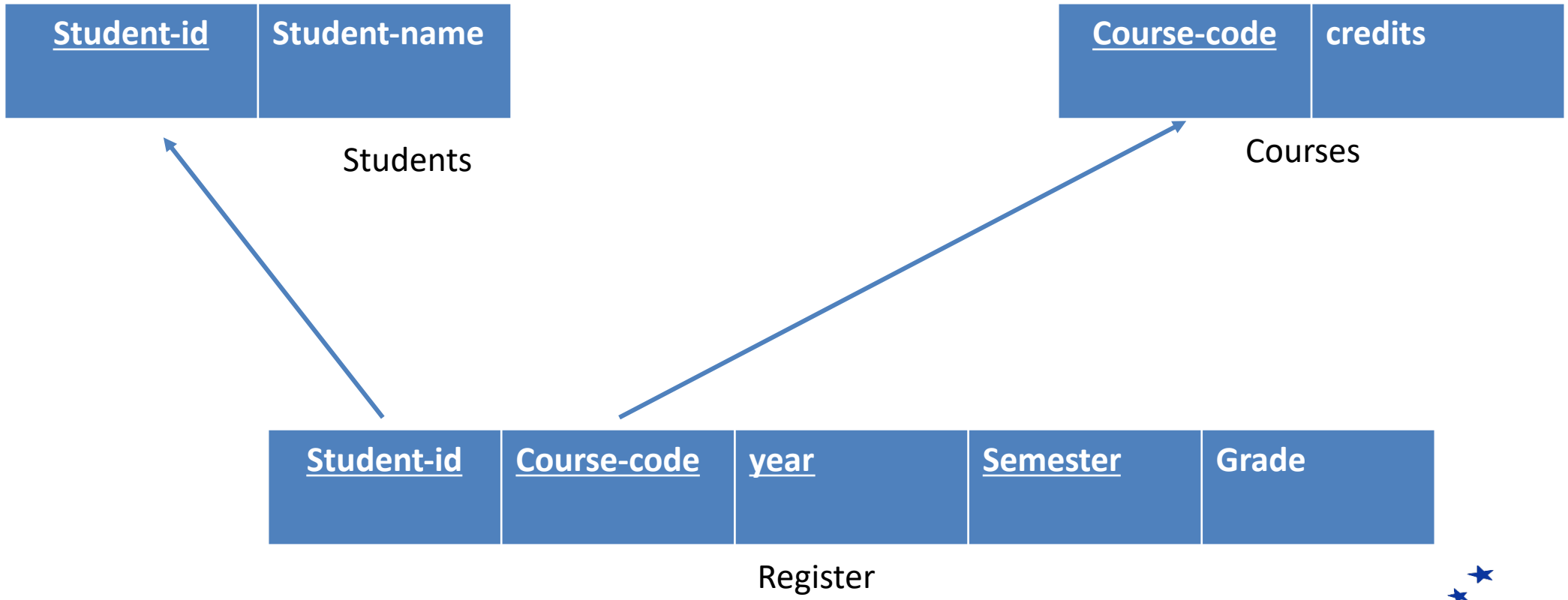
two type connections of n:1 connection.



Reichman University

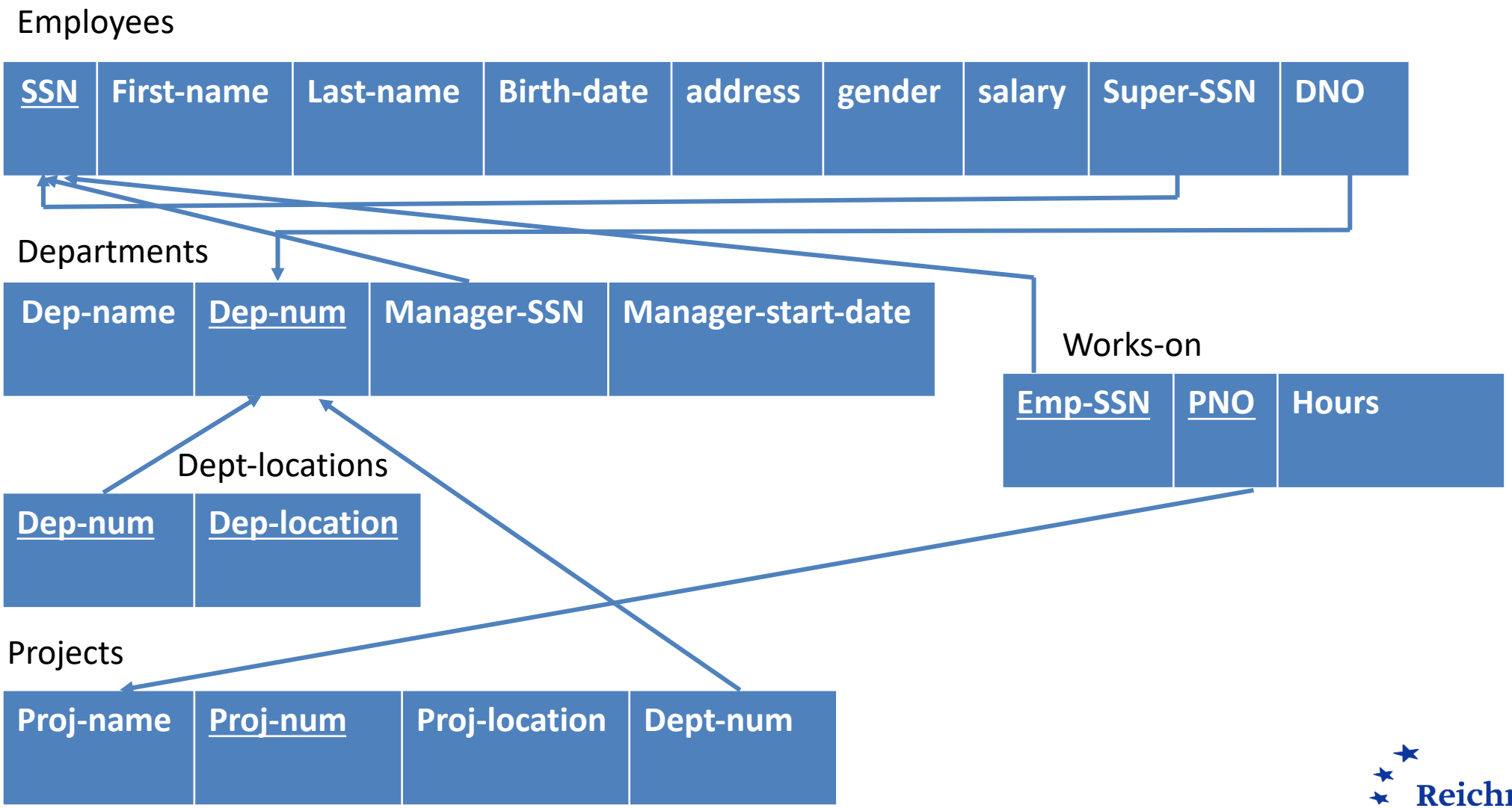
Complex network

m:n connection between students and courses – represented by “connections relation” (register)



Reichman University

Example of database scheme



Reichman University