GEORGE MASON UNIVERSITY | Computational and Data Sciences

CSI 695: Scientific Databases

Fall Term 2017

Lecture 3: From Data to Data Management

Lectures: Prof. Dr. Matthias Renz

Exercises: TBA

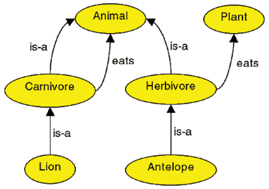
Outline

- Data, Metadata, Relationships and Ontologies
- Introduction to Data Modeling (E/R Diagram)
- The Relational Database Model
- Normalization
- Introduction to SQL (DDL, DML)

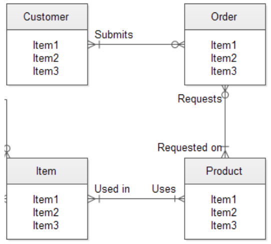
Data, Metadata, Relationships and Ontologies

In the following some definitions of basic data related incredients required to build a database in a data management system

- **Data** = Complex data entities from observations, experiments, simulations, models, and higher order assemblies.
- **Metadata** = Subset of data, and are "data that provides information about other data“
  [http://www.merriam-webster.com/dictionary/metadata]
- **Ontologies** = Systematic description of a given phenomenon.
- **Relationships** = Conceptual description of the association between entities of different types of data.
  Examples:
  Relationships between entities of data on different abstraction level (e.g. Ontologies)



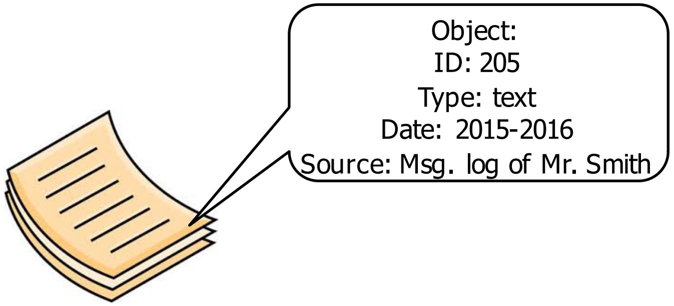
  Relationships between entities of data organized in different files/relations/collections



- A **relationship**, in the context of **databases**, is a situation that exists between two **relational database** tables when one table has a foreign key that references the primary key of the other table. **Relationships** allow **relational databases** to split and store data in different tables, while linking disparate data items.

Data, Metadata, Relationships and Ontologies

- What is Metadata?
  - Definition:
    Metadata is "data that provides information about other data“
    [http://www.merriam-webster.com/dictionary/metadata]
  - Supposed to give some abstract information about the data, rather the content.
  - BUT: Not just numbers, dates, times, sources … associated with the data.



- Why metadata matters??? Daniel John Sobieski
  - You call a phone sex service at 2 o‘clock in the morning and spoke for 17 minutes. But they „don‘t know“ what you are talking about.
  - They know you spoke with an HIV testing service, your doctor, and then your health insurance company that same hour. „Nobody knows“ what was discussed.
  - That afternoon you called a suicide prevention hotline from the Golden Gate Bridge. The topic of the call remains „a secret“.

Data, Metadata, Relationships and Ontologies

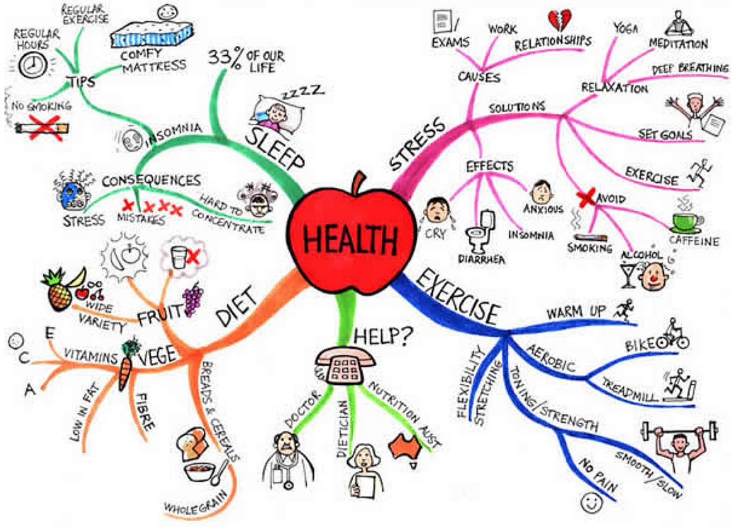
- Dublin Core (DC) Metadata
  - DC is an internationally approved standard.
  - DC metadata represent a minimal set.
  - Reference: http://dublincore.org/
  - DC contains 15 elements:
    - Title
    - Creator
    - Subject
    - Description
    - Publisher
    - Contributor
    - Date
    - Type
    - Format
    - Identifier
    - Source
    - Language
    - Relation
    - Coverage
    - Rights

Data, Metadata, Relationships and Ontologies

- Scientific databases require metadata to accompany the data:
  - To express the content of the data files.
  - To indicate the origin of the data (what instrument was used, when, under what conditions, analyzed with what data processing package, which version, by whom, …). This is called Data Provenance.
  - To report the structure of the data file to data analysis packages; for data integration within applications; for data-sharing and reuse.
  - To express data quality and associated measurement errors.
  - To express the context in which the data may be used (e.g., model input, model output, remote sensing, microarray analysis, astronomy).
  - To express the semantic meaning of the data (e.g., weather forecast, gene sequence, galaxy database, high-energy particle cross-sections, census counts, hydrodynamic simulation results, chemical reaction rates, gene expression map, …) – these are expressed in Ontologies.

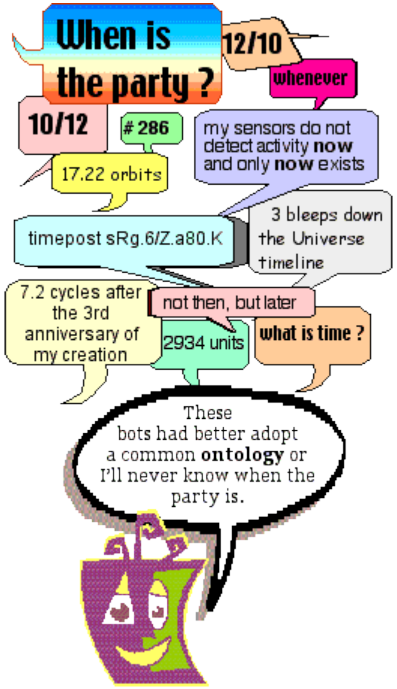
Data, Metadata, Relationships and Ontologies

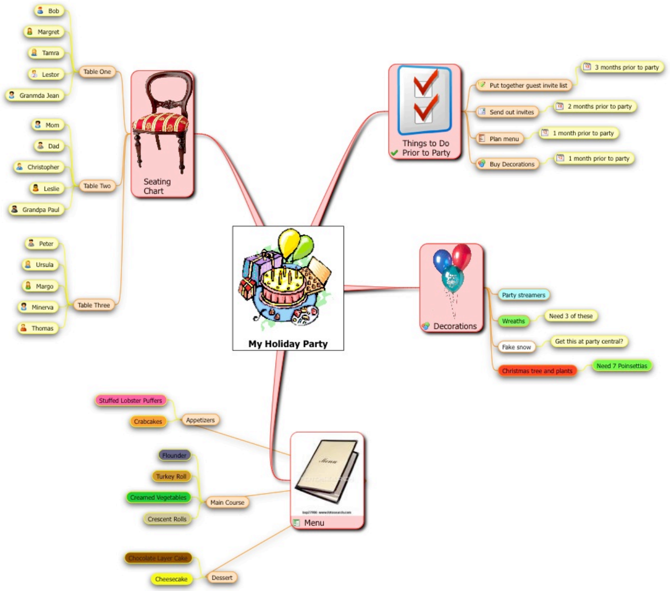
- Ontologies
  - Definition:
    "Systematic description of a given phenomenon." → describes the Concept
  - It often includes a controlled vocabulary and relationships, captures nuances in meaning and enables knowledge sharing and reuse.
  - An Ontology is a "formal explicit specification of a shared conceptualization." - T. Gruber



Data, Metadata, Relationships and Ontologies

- Ontologies
  - Why using Ontologies?


When is the party ?
12/10
whenever
10/12
# 286
my sensors do not detect activity now and only now exists
17.22 orbits
3 bleeps down the Universe timeline
timepost sRg.6/Z.a80.K
7.2 cycles after the 3rd anniversary of my creation
not then, but later
2934 units
what is time ?
These bots had better adopt a common ontology or I'll never know when the party is.

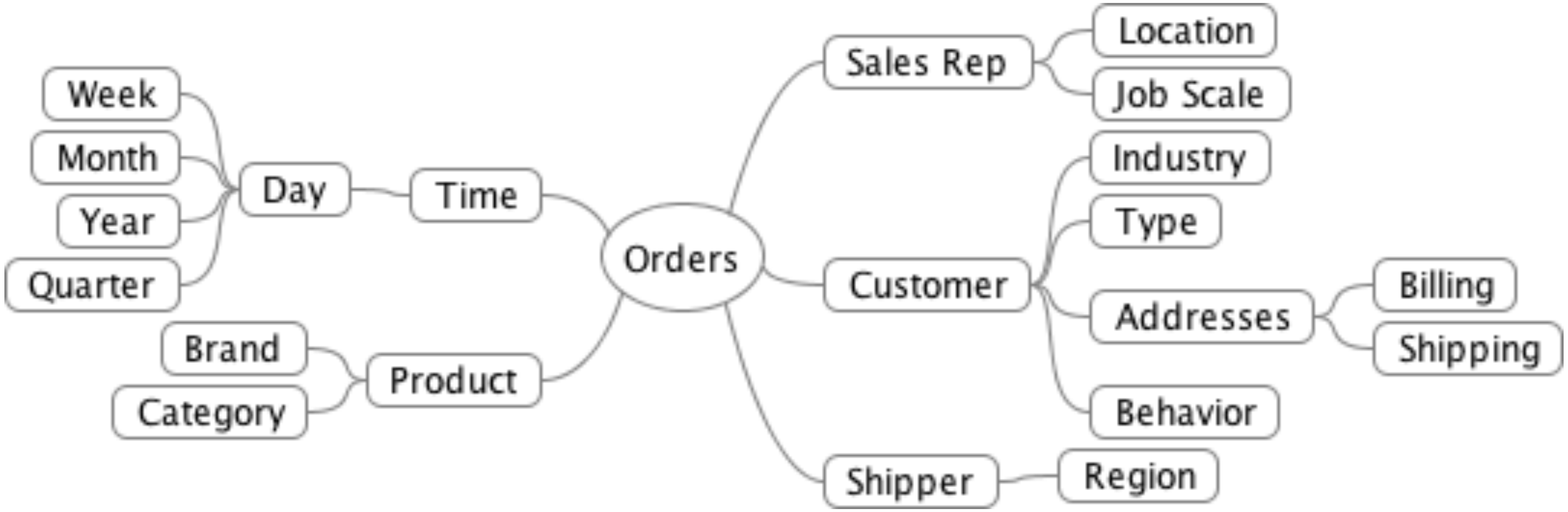


Outline

- Data, Metadata, Relationships and Ontologies
- Introduction to Data Modeling (E/R Diagram)
- The Relational Database Model
- Normalization
- Introduction to SQL (DDL, DML)

Introduction to Data Modeling

- **Data Modeling** begins with having the right concepts about your data
  - **Concept mapping** is like **Mind-mapping**, which is connected to Ontology specification.
  - **Concepts are** “information about the domain” and “information about the data” **= Metadata!**



Introduction to Data Modeling

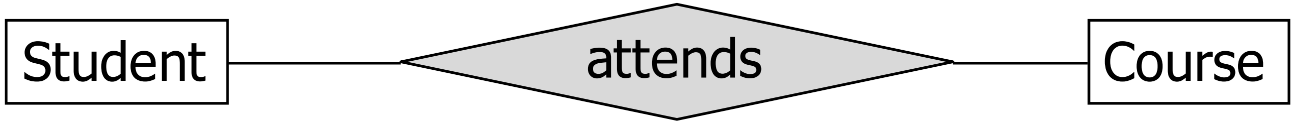
- Start modeling with “10 Questions”:
  - This is an expression that is used to represent the concept that the scientific end-users of a new database have. The “10 Questions” are several specific questions (or database queries) that they expect to be answered after the database is built.
  - For example, for an academic enrollment database:
    - How many students are enrolled in Chemistry courses?
    - How many students are enrolled in Chemistry 101?
    - How many sections of Chemistry 101 are offered?
    - Who is teaching Chemistry 101?
    - On what days is Chemistry 101 taught?
    - List all Chemistry courses.
    - List all Chemistry professors.
    - List all students in Chemistry 101 Section 001.
    - How many Chemistry 101 students are enrolled in the Lab?
    - What is the maximum enrollment in any of the Chemistry Labs?

Introduction to Data Modeling

- There are Entities: students, courses, instructors.
- Entities have Attributes: names and G# for students; course names (CDS 302), class times, and classrooms for courses; names and department affiliations for instructors.
- Attributes have Values: G0123456, CDS 302, West Building 1007, SPACS, Kirk Borne.
- Values have Constraints: determined by the university's people, places, and things
- Entities have Relationships: students take courses; instructors teach courses; courses are assigned to classrooms.

Introduction to Data Modeling

- The Entity/Relationship (E/R) Model
  - Generell Task:
    Find a formal description (Model) for a part of the real world to be modelled.
  - Intermediate Step:
    - Description by natural language (specifications)
      Example: *In a database, all students should be stored in association with the courses they are registered for.*
    - Description by abstract graphical illustration:

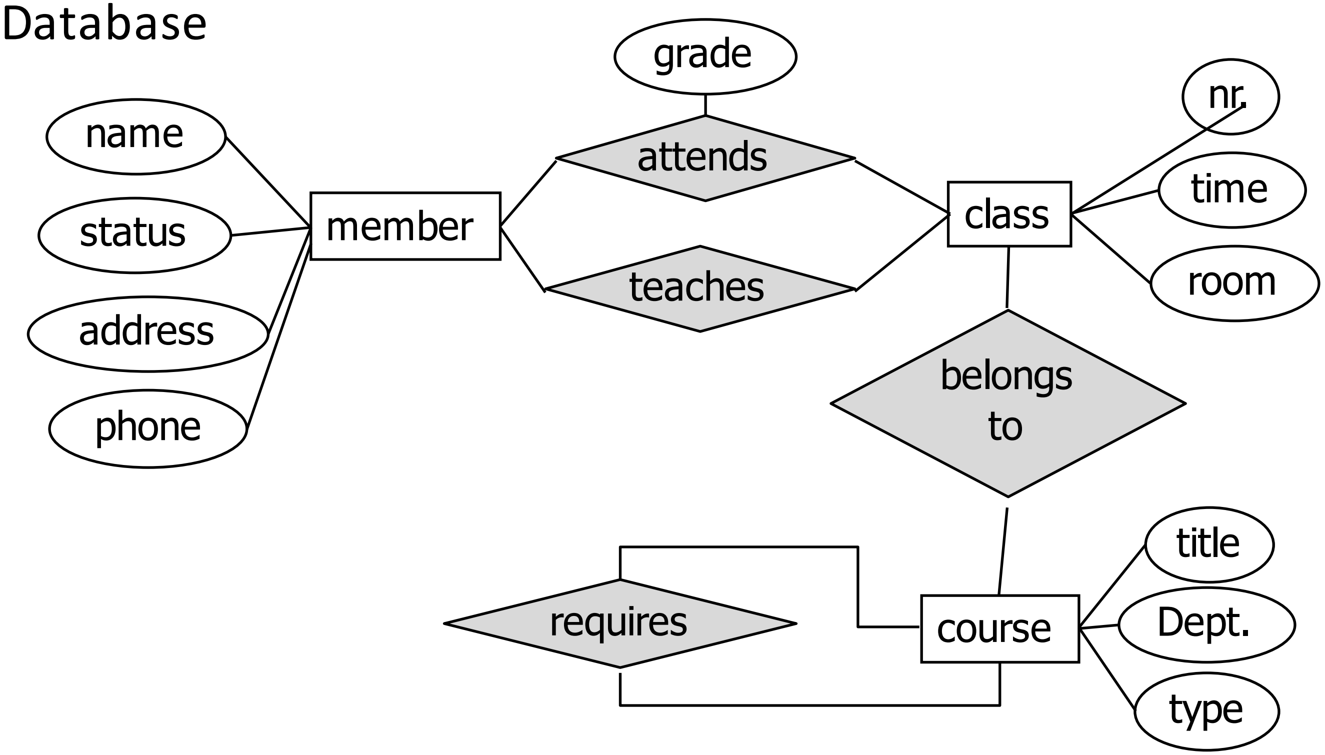


Introduction to Data Modeling

- The Entity/Relationship (E/R) Model ...
  - is used to build a conceptional scheme of an excerpt (detail) of the real world.
  - is graphically illustrated by the E/R-diagram (ERD).
  - is an abstract (non-machine readable) model.
  - disregards any efficiency issues.
  - helps to identify an appropriate database schema.
    - Simple rules for the transformation into a database schema.
    - Efficiency issues have to be taken into account for the transformation (Normalization!!!).

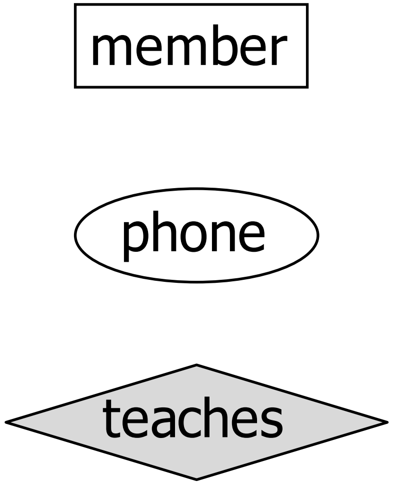
Introduction to Data Modeling

- The Entity/Relationship (E/R) Model ... an Example:
  - College Database



Introduction to Data Modeling

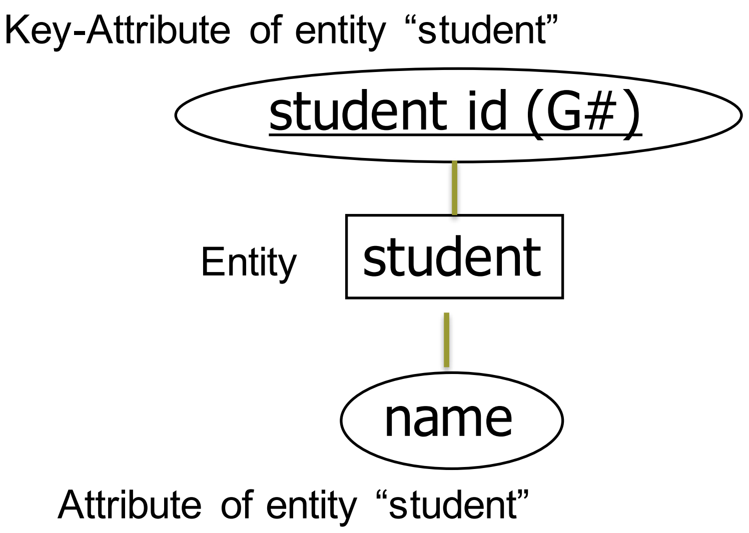
- Elements of an E/R-Model:
  - **Entities**
    (a.k.a entity sets)
    type of object
  - **Attribute**
    components of an object
  - **Relationship**
    between entities



Challenge: find proper entities, attributes and relationships

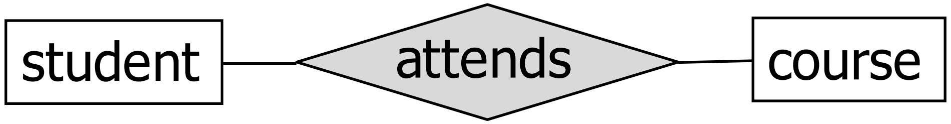
Introduction to Data Modeling

- Elements of an E/R-Model:
  - Entities
    - Objects, types, „beings“
    - Objects of the real world, distinguishable
    - Examples: human, house, course, …
  - Attribute
    - Describe entities by characteristic properties
    - (usually) simple data types, incl. INT, STRING
    - Examples: Color, weight, name, title, …
    - Usually only relevant attributes specified
  - Key-Attribute(s):
    Attributes that uniquely identify entities (primary key) are underlined.
    (key attributes will be introduced later on (relational model))


Key-Attribute of entity “student”
student id (G#)
Entity
student
name
Attribute of entity “student”

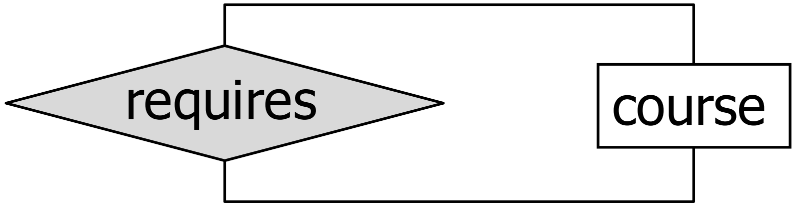
Introduction to Data Modeling

- Elements of an E/R-Model:
  - Relationships
    - Show relationships between entities
    - Example: student attends course



Instances:
attends(Jim,algorithmic II)
attends(Lin,physics I)
attends(han,geography I)

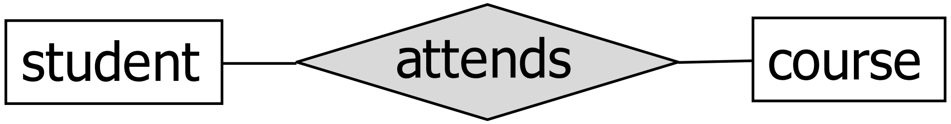
    - Relationship between an entity to itself possible (relationship between instances associated with an entity)



Instances:
requires(AlgII,AlgI)

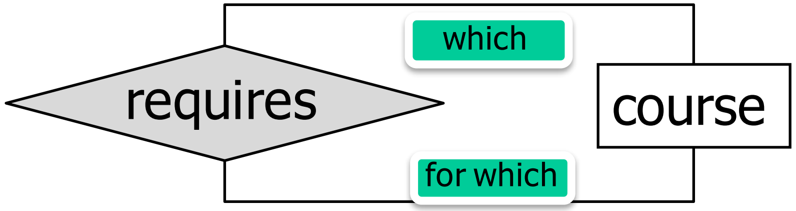
Introduction to Data Modeling

- Elements of an E/R-Model:
  - Relationships
    - Show relationships between entities
    - Example: student attends course



Instances:
attends(Jim,algorithmic II)
attends(Lin,physics I)
attends(han,geography I)

- Relationship between an entity to itself possible (relationship between instances associated with an entity)

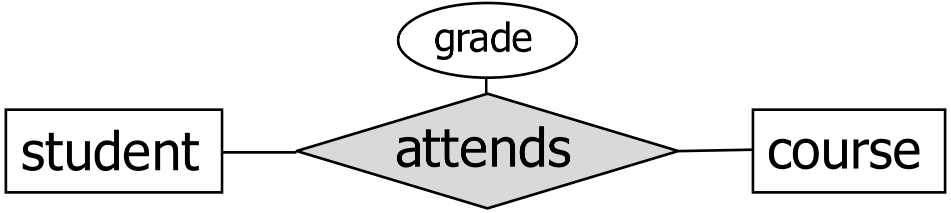


Instances:
requires(AlgII,AlgI)

attending entities can have different roles

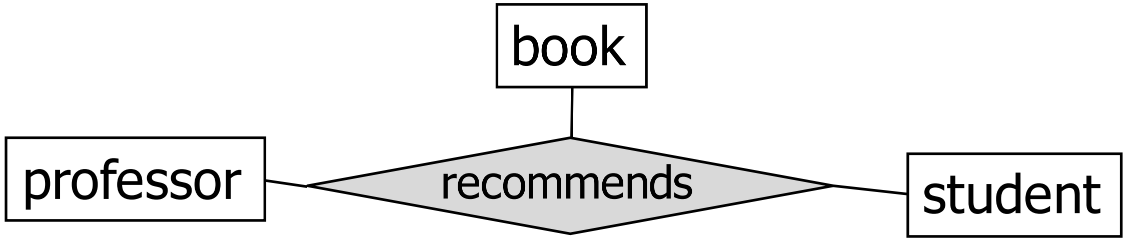
Introduction to Data Modeling

- Elements of an E/R-Model:
  - Relationships
    - Relationships may have their own attributes.
    - Example: student attends course with grade A.



Instances:
attends(Jim,algorithmic II,C)
attends(Lin,physics I,A)
attends(han,geography I,B)

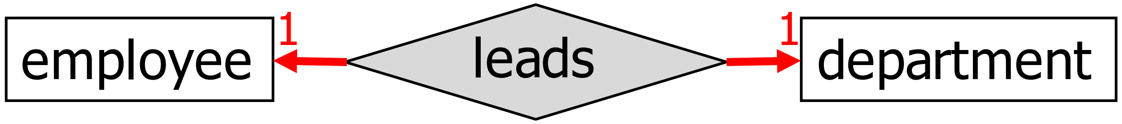
    - Relationships between multiple entities.



Instances:
recommends(Prof. Ali,Lin,Data Model I)

Introduction to Data Modeling

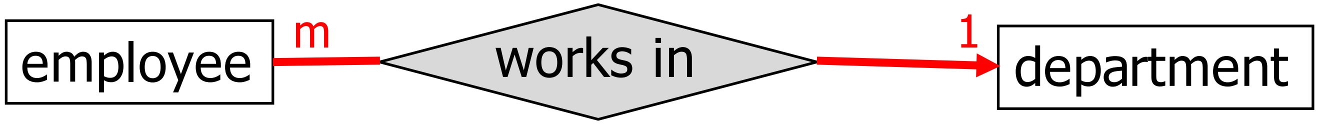
- Elements of an E/R-Model:
  - Functionality of Relationships
    - **1:1 (one-to-one) relationship:**



- Characteristic: each instance (object) from the left entity belongs to at most one instance of the right entity and vice versa.
- Graphical notation: arrow indicates relationship to at most one instance of the entity the arrow directs to.
- Example: Each employee can lead at most one department (right arrow) and each department can be lead by at most one employee (left arrow)

Introduction to Data Modeling

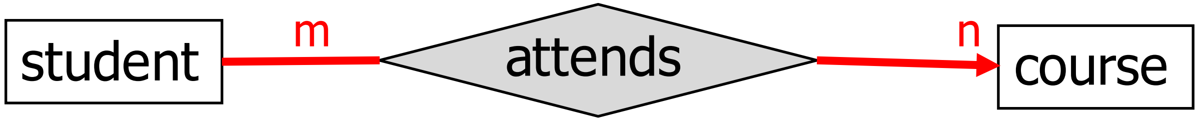
- Elements of an E/R-Model:
  - Functionality of Relationships
    - **m:1 (many-to-one) relationship:**



    - Characteristic: each instance (object) from the left entity belongs to at most one instance of the right entity, but each instance of the right entity may belong to many instances of the left one.
    - Graphical notation: „many“-side of the relationship without an arrow.
    - Example: Each employee works in at most one department (right arrow) but each department can be assigned to many employees working in it.

Introduction to Data Modeling

- Elements of an E/R-Model:
  - Functionality of Relationships
    - **m:n (many-to-many) relationship:**

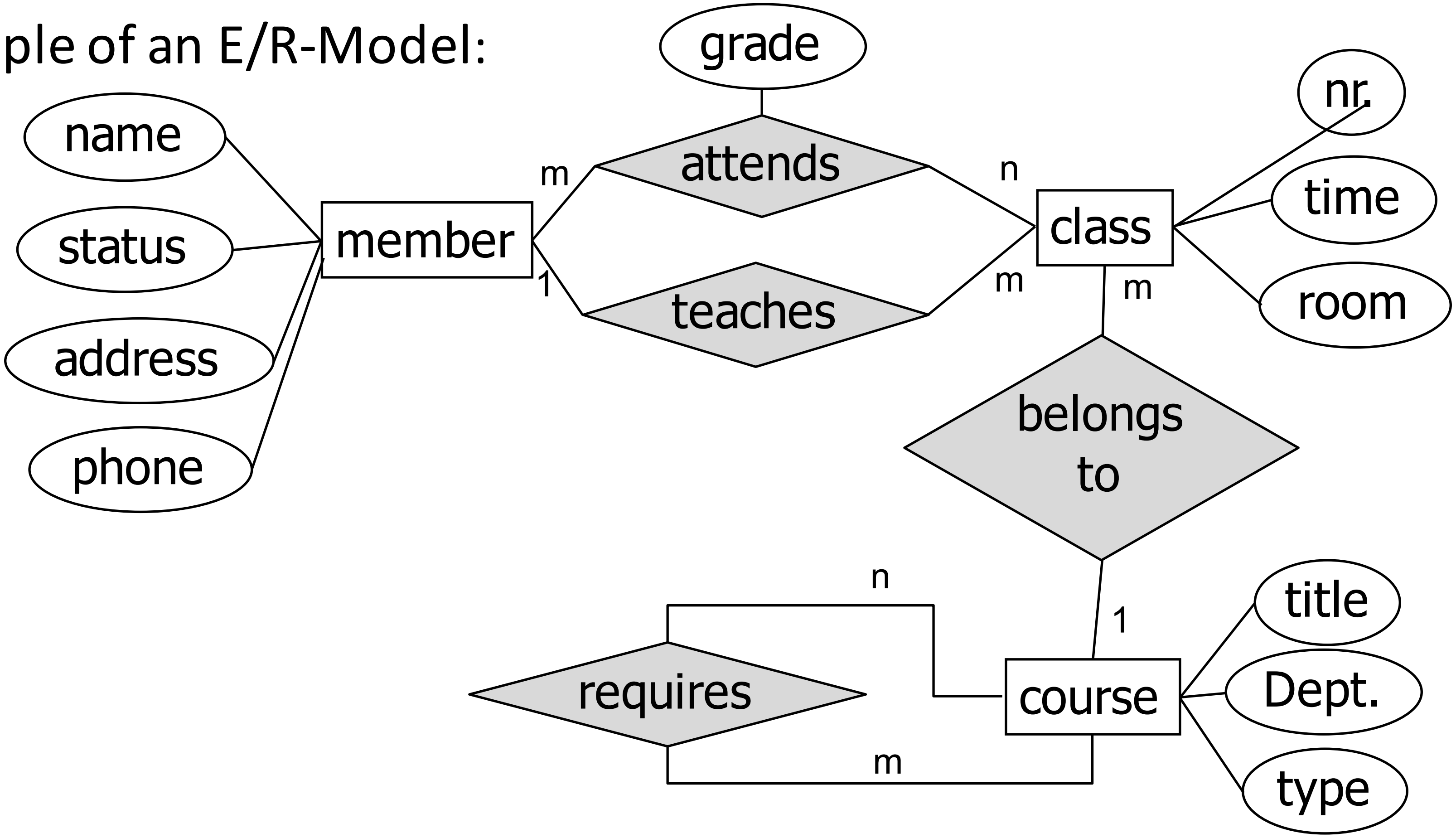


Instances:
attends(Jim,algorithms)
attends(Lin,algorithms)
attends(han,math)
attends(Jim,math)

- Characteristic: each instance (object) from the left entity belongs to many instances of the right entity, and each instance of the right entity may belong to many instances of the left one. (i.e. no restrictions for the relationship)
- Graphical notation: both „many“-sides without an arrow.
- Example: Each student attends one or more courses, and each course can be attended by many students.

Introduction to Data Modeling

- Example of an E/R-Model:



Outline

- Data, Metadata, Relationships and Ontologies
- Introduction to Data Modeling (E/R Diagram)
- The Relational Database Model
- Normalization
- Introduction to SQL (DDL, DML)

The Relational Database Model

- The relational database model ...
  - uses the fact that information can be easily represented in form of tables (relations).
  - abstracts from the internal organization of the data.
    → concept driven data organization instead of organization based on the underlying memory system
    →data organization independent of the system
- Why is this important?
  - It has been introduced by Edgar F. Codd, 1970.
    A relational model of data for large shared data banks. Comm. of the ACM 13.06.1970
  - It is the principle of many commercial and open-source database systems

ORACLE

IBM DB2 / Informix

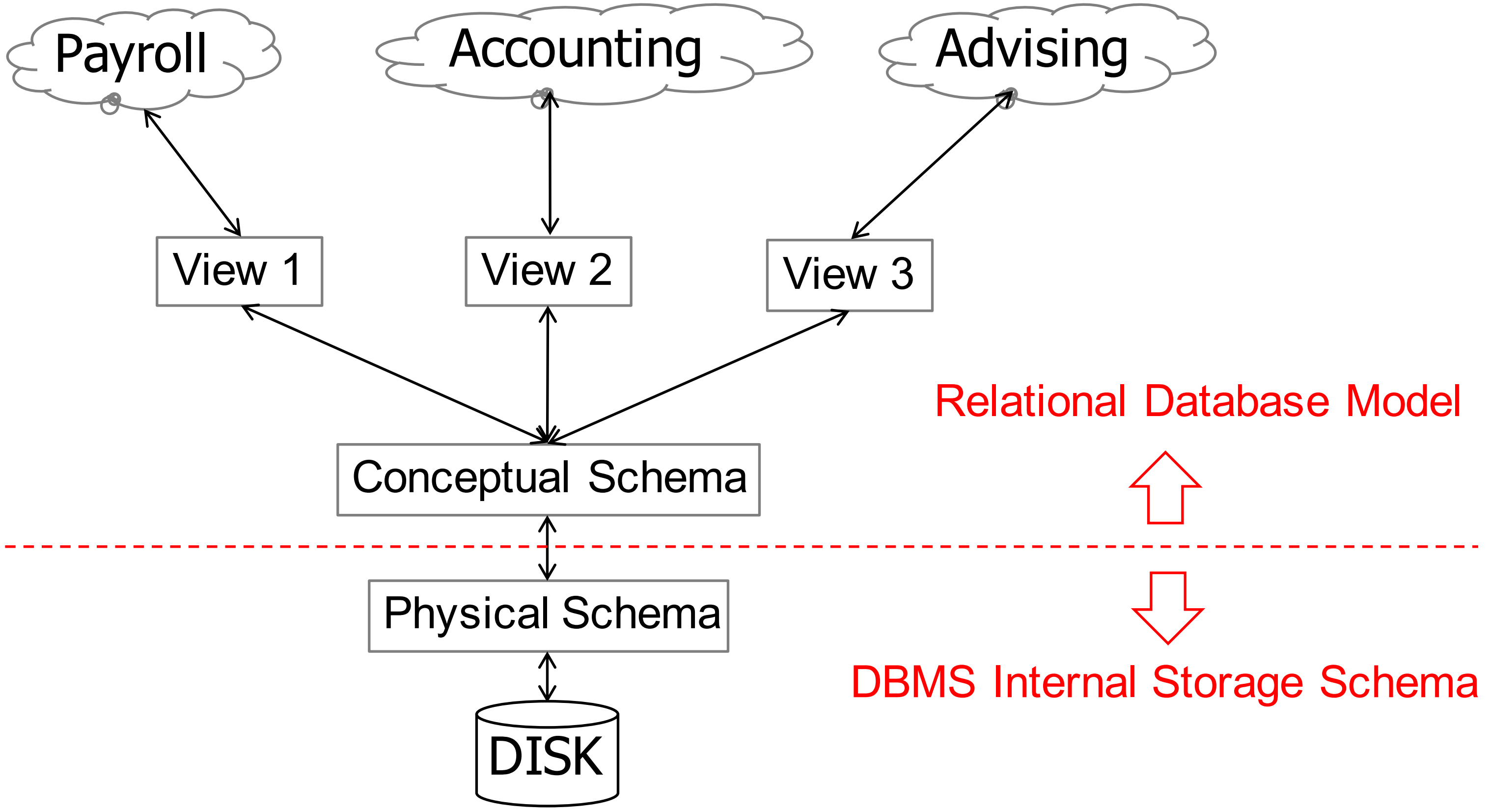
MySQL

Microsoft SQL Server

PostgreSQL

The Relational Database Model

- Recap - Levels of data independence:



The Relational Database Model

- Definition: Domain
  - logically associated (finite or infinite) set of values
  - Example:
    - D1 = Integer
    - D2 = String
    - D3 = Date

      Infinite domain

    - D4 = {red,green,blue,yellow}
    - D5 = {1,2,3}

      Finite domain

The Relational Database Model

- Definition (mathematical): Relation R
  - is subset of the cartesian product of k domains D1, D2, ..., Dk

$$R \subseteq D_1 \times D_2 \times \ldots \times D_k$$

  - Examples (k=2):
    - D1 = {1, 2, 3}, D2 = {a, b}
    - R1 = {} (empty set)
    - R2 = {(1,a),(2,b)}
    - R3 = {(1,a),(2,a),(3,a)}
    - R4 = D1 × D2 = {(1,a),(1,b),(2,a),(2,b),(3,a),(3,b)}
  - The number of tuples (entities) in a relation *R* is called cardinality of *R*, denoted by |*R*|.

The Relational Database Model

- The domains in a relation can be thought as columns in a table and are called attributes.
- For $R \subseteq D_1 \times \ldots \times D_k$, k is called the degree of R.
- The elements of a relation are called tuples:
  (1,a),(2,a),(3,a) are 3 tuples of degree k = 2
- A relation is a set of tuples, i.e. the order of tuples is irrelevant.
  {(0,a),(1,b)} = {(1,b),(0,a)}
- BUT, the order of attributes within a tuple does matter!!!
  {(0,a),(1,b)} ≠ {(a,0),(b,1)}

The Relational Database Model

- Alternative definition in Databasesystems:
  - A relation R is an instantiation of a relational schema.
  - Schema – structural description of relations in a database
It includes
    - the name of the relation
    - the attributes of the relation and
    - the types of these attributes

    It builds the header of a table (relation)
  - Instance – actual contents of a relation at a given point in time
    - Note: Null-values are important in relational DBS.
      meaning: value is not known yet.

Student

ID	Name	Credit	Pic
143	Amy	17	☺
539	Bob	28	☹
342	Tim	null	😐

The Relational Database Model

- Alternative definition in Databasesystems (cont.):
  - A relation is specified by a schema (in the ordered relational schema model):
    - k-tuple of domains (attributes)
    - Attributes referenced according to their position within the tuple
    - Attributes may have an attribute name in addition

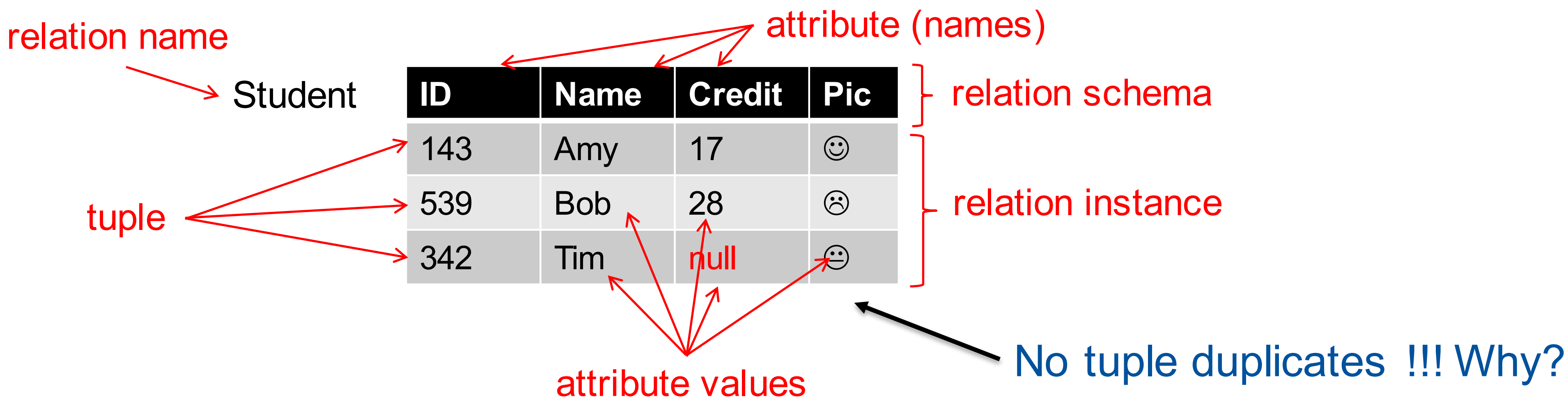
    Relation: R = (A1:D1,…,Ak:Dk), where A1 = 1st attribute and D1 = domain of A1
  - Example:
    - Schema:
      R = (ID:INT, Name:STRING, Credit:INT, Pic:JPG)
    - Instance: r = {(143,Amy,17,☺),(539,Bob,28,☹),…}

Student

ID	Name	Credit	Pic
143	Amy	17	☺
539	Bob	28	☹
342	Tim	null	😐

The Relational Database Model

- Terms & Definitions:
  - Relation Instance: Instance of a (relational) schema
  - Database schema: Set of (relational) schemas
  - Database: Set of relations (relation instances)



relation name

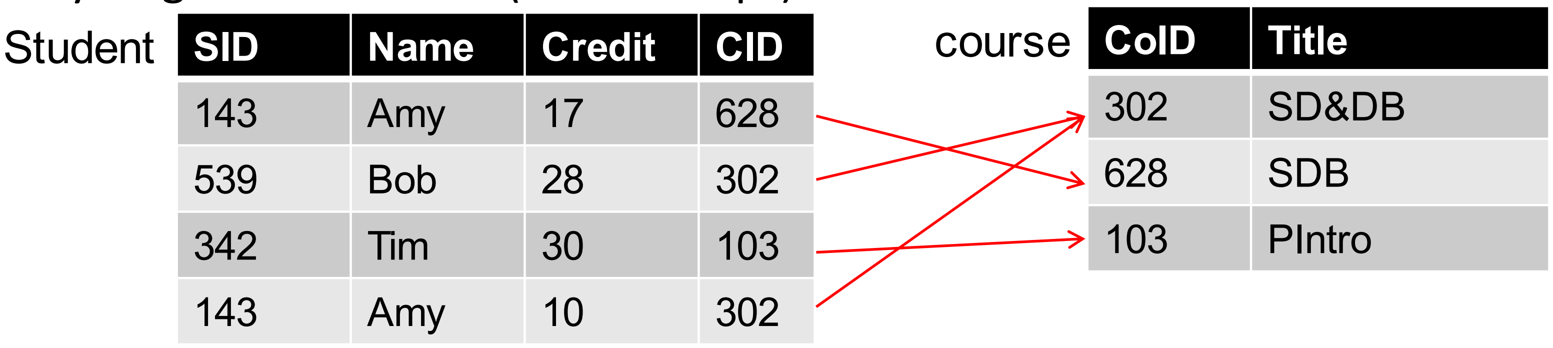
Student

ID	Name	Credit	Pic
143	Amy	17	☺
539	Bob	28	☹
342	Tim	null	😐

No tuple duplicates !!! Why?

The Relational Database Model

- Keys:
  - Tuples have to be unique (uniquely identified).
  - Why? E.g. for references (relationships):



Student

SID	Name	Credit	CID
143	Amy	17	628
539	Bob	28	302
342	Tim	30	103
143	Amy	10	302

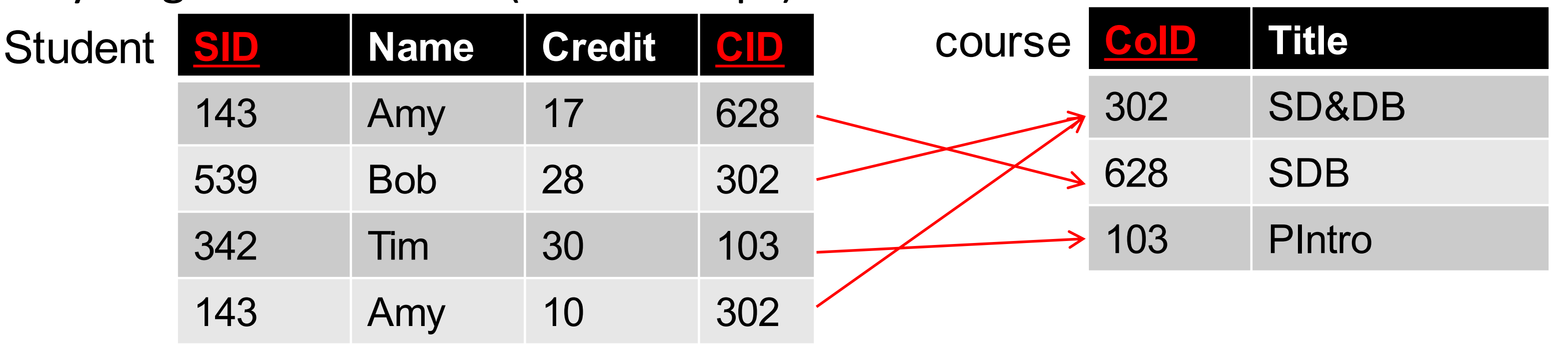
course

CoID	Title
302	SD&DB
628	SDB
103	PIntro

  - Object reference in Java: Block address in memory
  - Object reference in relational db model: reference to tuples with attribute values (keys)

The Relational Database Model

- Keys:
  - Tuples have to be unique (uniquely identified).
  - Why? E.g. for references (relationships):



Student

SID	Name	Credit	CID
143	Amy	17	628
539	Bob	28	302
342	Tim	30	103
143	Amy	10	302

course

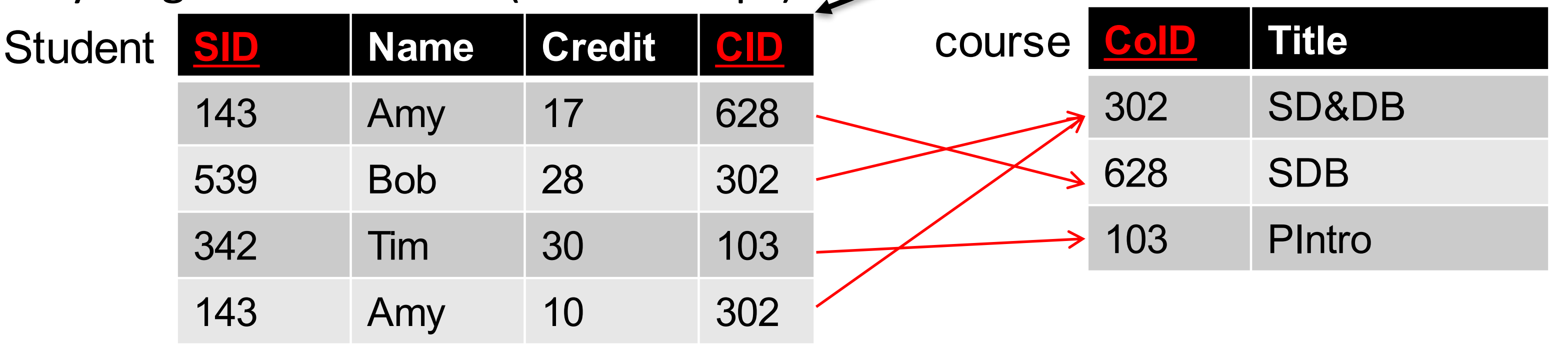
ColD	Title
302	SD&DB
628	SDB
103	PIntro

- Object reference in Java/C++/…: Block address in memory
- Object reference in relational db model: reference to tuples with attribute values (keys)
- One ore more attributes are marked as key (called primary key)
- Key attribute(s) are underlined

The Relational Database Model

- Keys:
  - Tuples have to be unique (uniquely identified).
  - Why? E.g. for references (relationships):

Attributes that reference tuples of another relation (table) are called foreign key(s), e.g. CID



Student

SID	Name	Credit	CID
143	Amy	17	628
539	Bob	28	302
342	Tim	30	103
143	Amy	10	302

course

CoID	Title
302	SD&DB
628	SDB
103	PIntro

- Object reference in Java/C++/…: Block address in memory
- Object reference in relational db model: reference to tuples with attribute values (keys)
- One ore more attributes are marked as key (called primary key)
- Key attribute(s) are underlined

The Relational Database Model

- Keys (formal definition):
  - A subset S of the attributes of a relation schema R ($S \subseteq R$) is called key, iff the following holds:
  - Uniqueness:
    No instance of R contains two tuples that are equal in the values of all attributes in S.
  - Minimality:
    There does not exist any real subset $T \subset S$ ($T \neq S$), that already fulfills the uniqueness property.
- Example:
  What is the key here?
  and Why?

Student Credits	SCID	SID	Name	Credit	CID
(t_1=)	1	143	Amy	17	628
(t_2=)	2	539	Bob	28	103
(t_3=)	3	342	Amy	17	103

The Relational Database Model

- Key Example (cont.): What is the key here? and Why?

Student Credits	SCID	SID	Name	Credit	CID	Term
(t_1=)	1	143	Amy	17	628	Fall 2015
(t_2=)	2	539	Bob	28	103	Fall 2015
(t_3=)	3	342	Amy	17	103	Spring 2016

- “Name”, “Credit”, “CID” and “Term” are no keys - have duplicates
- “SID” not a key, though does not have duplicates in “Student Credits”, but is logically not unique, since a student can get credits from different courses.
- “SCID“ is a key (candidate)
- {SCID,SID} is not a key: violates minimality!
- {SID,CID,Term} is a key (candidate)

The Relational Database Model

- Notes on Keys:
  - „Minimality“ does not mean key with the smallest number of attributes!!!
  - If there are multiple different keys, they are called key candidates.
  - One!!! key candidate has to be selected as primary key.
  - Tuples in a relation R are referenced by the primary key of R.
  - Attribute(s) that reference tuples in another relation R (by means of R’s primary key) is(are) called foreign key.
  - A relation has one primary key but can have multiple foreign keys.

The Relational Database Model

- From E/R Model to Relational DB Model
  - The E/R model can be transformed to the relational database model (schema) by simple rules.
    - E/R Diagram → Relational schema
    - Entity → Relation
    - Entity attributes → Attributes of the corresponding relation
    - Entity keys → Primary keys of the relation
    - Relationships → Additional attributes or relation, depending on the functionality of the relationship