Topic 3-Part 1: Relational Model (Chapters 2 and 6)

Database System Concepts

Topic 3: Contents

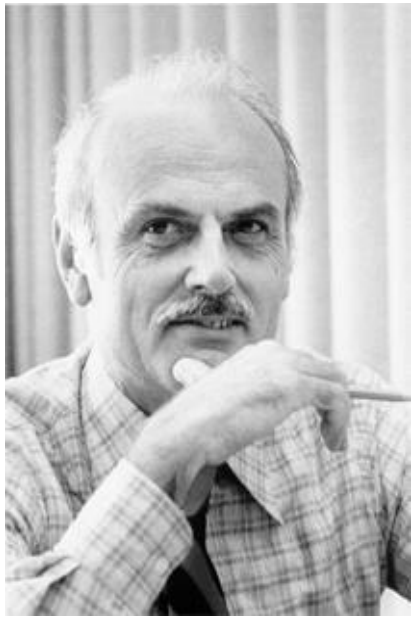
- Part 1 (Chapters 2 and 6)
  - Relational Model
  - Converting an ER diagram to relation schemas
- Part 2 (Chapters 3, 4, 5)
  - SQL

Topic 3: Part 1

- Relational Data Model
- Converting an ER Diagram to Relation Schemas

Relational Data Model - Introduction

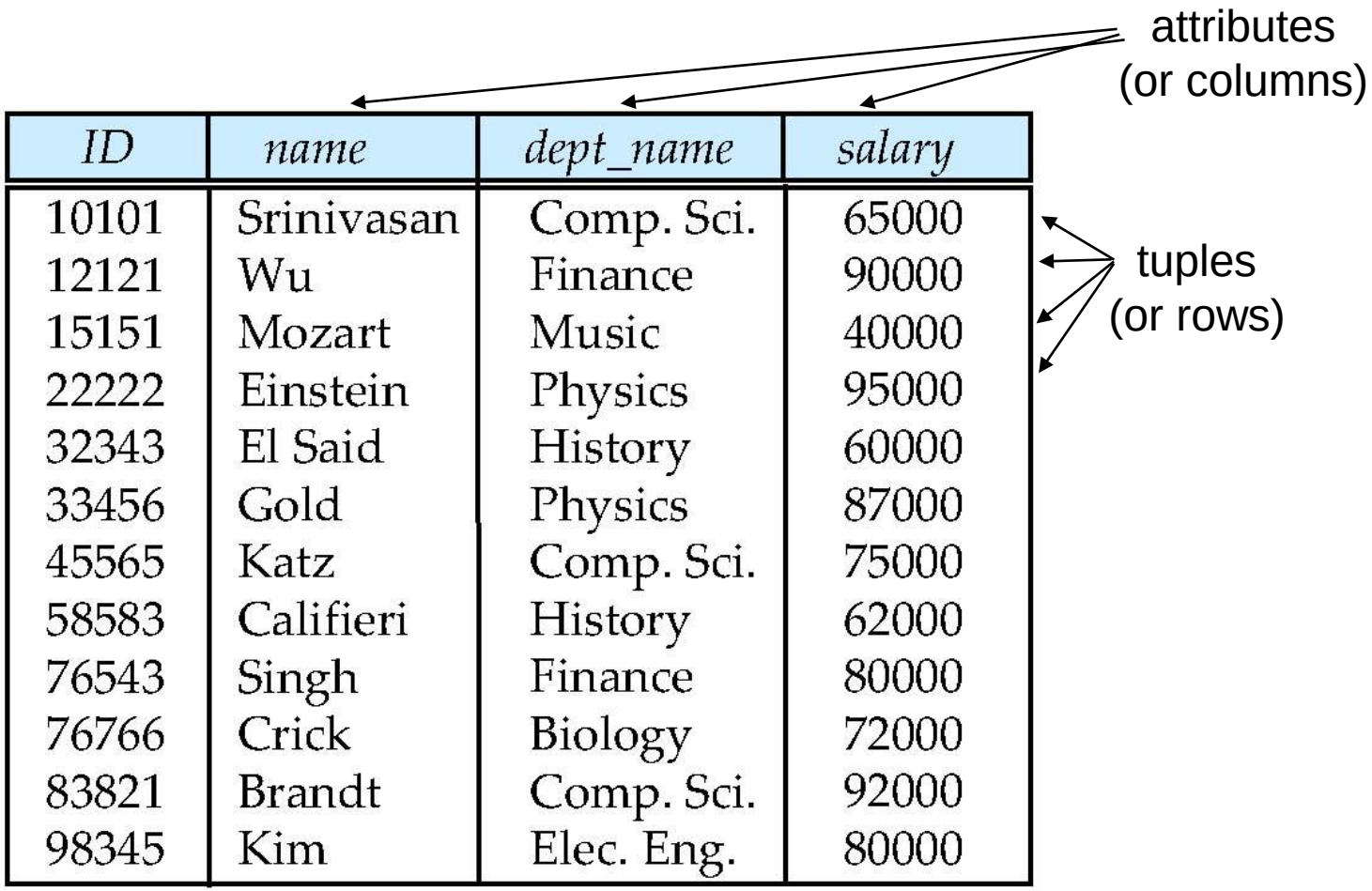
- Developed in 1969 (before the ER model) by Edgar Codd at IBM



(http://en.wikipedia.org/wiki/Edgar_F._Codd)

- A relational database consists of a collection of relations (tables).

Example of a Relation



ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Attribute Types

- The set of allowed values for each attribute is called the **domain** of the attribute
- Attribute values are (normally) required to be **atomic**; that is, indivisible
- The special value ***null*** is a member of every domain
- The null value causes complications in the definition of many operations

Relation Schema and Instance

- $A_1, A_2, \ldots, A_n$ are *attributes*

- $R\ (A_1, A_2, \ldots, A_n\ )$ is a *relation schema named R*

  Example:

  instructor (*ID, name, dept_name, salary*)

- Formally, given sets $D_1, D_2, \ldots. D_n$ a **relation *r*** is a subset of

  D_1 x D_2 x … x D_n

  Thus, a relation is a set of *n*-tuples $(a_1, a_2, \ldots, a_n)$ where each $a_i \in D_i$

- The current values (**relation instance**) of a relation are specified by a table
- An element ***t*** of ***r*** is a *tuple*, represented by a *row* in a table

Relations are Unordered

- Order of tuples is irrelevant (tuples may be stored in an arbitrary order)
- Example: *instructor* relation with unordered tuples

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

Keys

- Let K $\subseteq$ R
- *K* is a **superkey** of *R* if values for *K* are sufficient to identify a unique tuple of each possible relation *r(R)*
  - Example: {*ID*} and {ID,name} are both superkeys of *instructor.*
- Superkey *K* is a **candidate key** if *K* is minimal

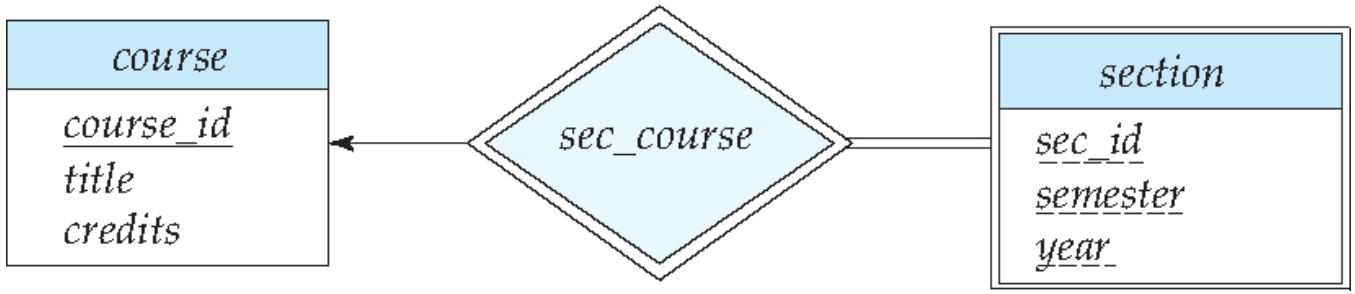
  Example: {*ID*} is a candidate key for *Instructor*
- One of the candidate keys is selected to be the **primary key**.
  - which one?
- **Foreign key** constraint: Value in one relation must appear in another
  - **Referencing** relation
  - **Referenced** relation

Converting an ER diagram to Relation Schemas

- Entity sets and relationship sets can be expressed uniformly as *relation schemas* that represent the contents of the database.
- A database which conforms to an E-R diagram can be represented by a collection of relation schemas.
- A relational database design produces a collection of relation schemas
- For each entity set and relationship set there is a unique schema that is assigned the name of the corresponding entity set or relationship set.
- Each schema has a number of columns (generally corresponding to attributes), which have unique names.
- Format of a relation schema R of n attributes:
  - R (A1, A2,…, An)
  - Underline attributes that form the primary key

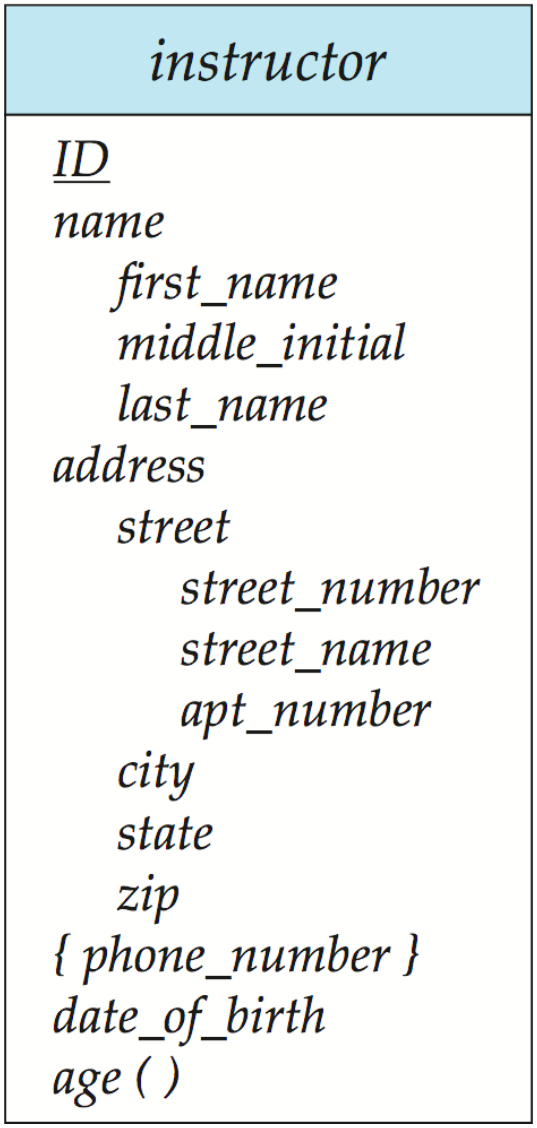
Converting Entity Sets With Simple Attributes

- A strong entity set reduces to a schema with the same attributes
- A weak entity set becomes a relation schema that includes the primary key of the identifying strong entity set
- Example: What are the relation schemas for the strong entity *course* and weak entity set *section*?



Answer:

Composite and Multivalued Attributes



- Composite attributes are flattened out by creating a separate attribute for each component attribute
  - Example: given entity set *instructor* with composite attribute *name* with component attributes *first_name* and *last_name* the schema corresponding to the entity set has two attributes *name_first_name* and *name_last_name*
    - *Prefix omitted if there is no ambiguity*
- Ignoring multivalued attributes, extended instructor schema is
  - *instructor(ID, first_name, middle_initial, last_name, street_number, street_name, apt_number, city, state, zip_code, date_of_birth, age)*

Composite and Multivalued Attributes (cont.)

- A multivalued attribute *M* of an entity *E* is represented by a separate schema *EM*
  - Schema *EM* has attributes corresponding to the primary key of *E* and an attribute corresponding to multivalued attribute *M*
  - Example: Multivalued attribute *phone_number* of *instructor* is represented by a schema:

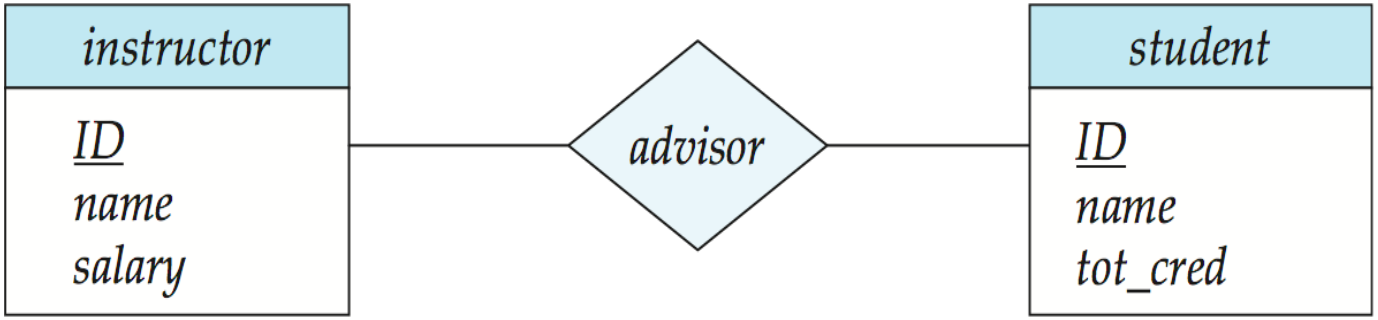
Answer:

  - Each value of the multivalued attribute maps to a separate tuple of the relation on schema *EM*
    - For example, an *instructor* entity with primary key 22222 and phone numbers 456-7890 and 123-4567 maps to two tuples:
      (22222, 456-7890) and (22222, 123-4567)

Converting Relationship Sets That Have No Attributes

Converting Many-to-Many Relationship Sets

- A many-to-many relationship set is represented as a schema with attributes for the primary keys of the two participating entity sets
- Example: what is the relation schema for relationship set *advisor*?



Answer:

Converting One-To-One Relationship Sets

- Alternative 1: create a relation schema for the relationship set; the key can be the key of any of the participating entity sets
  - Example:
- Alternative 2: if one of the participating entity sets has a total participation, we can create 2 relations, one for each entity set, and include the relationship in the relation schema of the entity set with total participation (to avoid null values)
  - Example:
- Alternative 3: if both entity sets have total participation, we can create one schema to represent both entity sets and the relationship set.
  - Example:

Representing One-to-Many or Many-to-One Relationship Sets

- Alternative 1: create a schema for the relationship set; the key is the key of the Many-side entity set
  - Example:

- Alternative 2: put the relationship in the relation schema of the Many-side entity set
  - Example:

Converting Relationship Sets That Have Attributes

- The attributes of the relationship set are included in the relation schema where the relationship is represented
- Whether an attribute of the relationship set is part of the key depends on the semantic
- Example:

Converting The Relationship Set Linking A Weak Entity Set To Its Identifying Strong Entity Set

- No separate relation schema is needed
- Redundant info
- Why?

Converting Generalization/Specialization

- Disjoint:
  - Total
  - Partial
- Example:

Converting Aggregation

- Example:

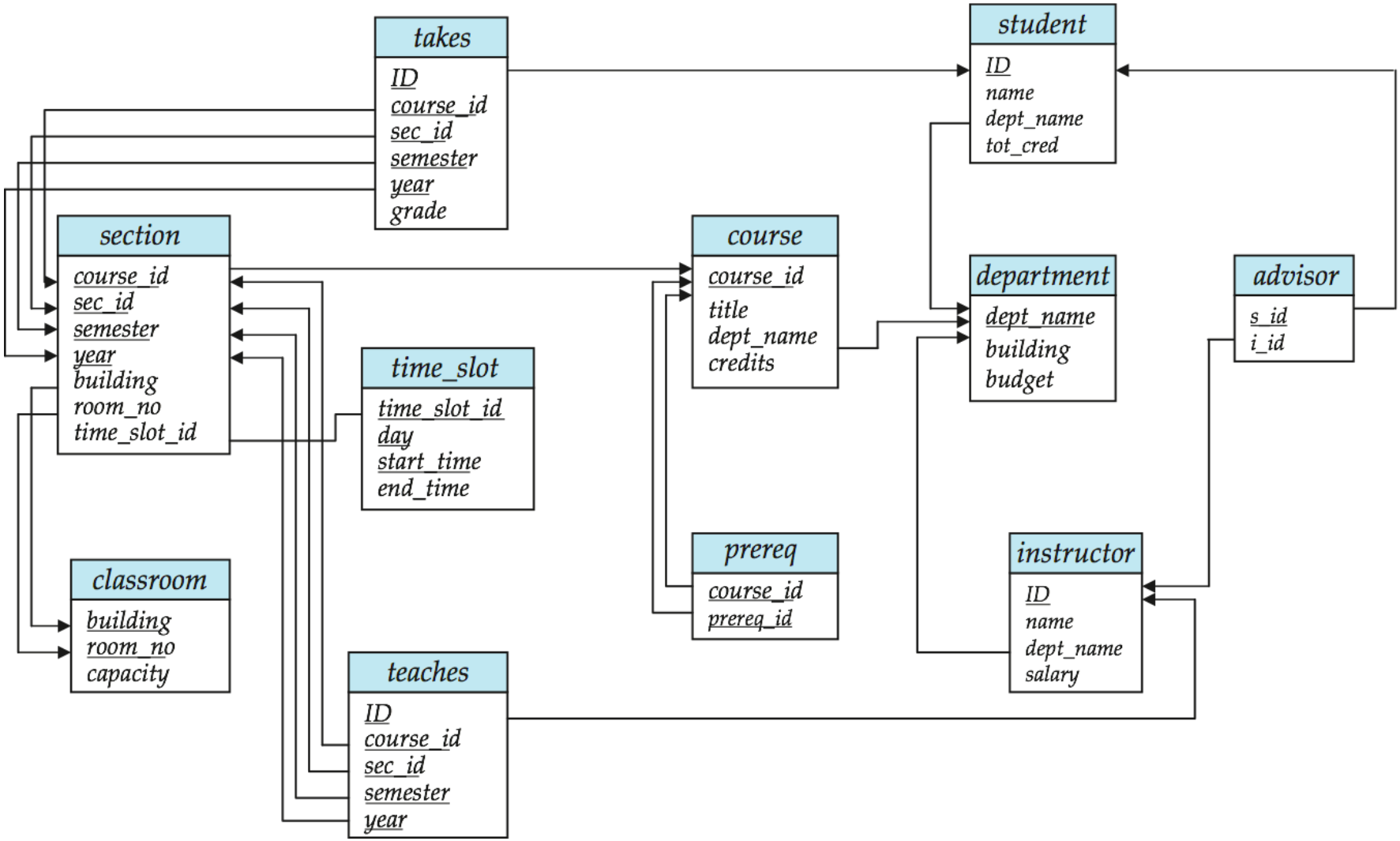
Converting Role Indicator (Recursive Relationship)

- Example:

Schema Diagram

- A relational database schema, along with primary key and foreign key dependencies, can be depicted by a schema diagram.
- A schema diagram:
  - Each relation appears as a box
    - The relation name appears at top of the box
    - The attribute names are inside the box
    - The primary key is underlined
  - Foreign key dependencies appear as arrows from the foreign key attributes of the referencing relation to the primary key of the referenced relation.

Schema Diagram for University Database



End of Topic 3 – Part 1

Database System Concepts