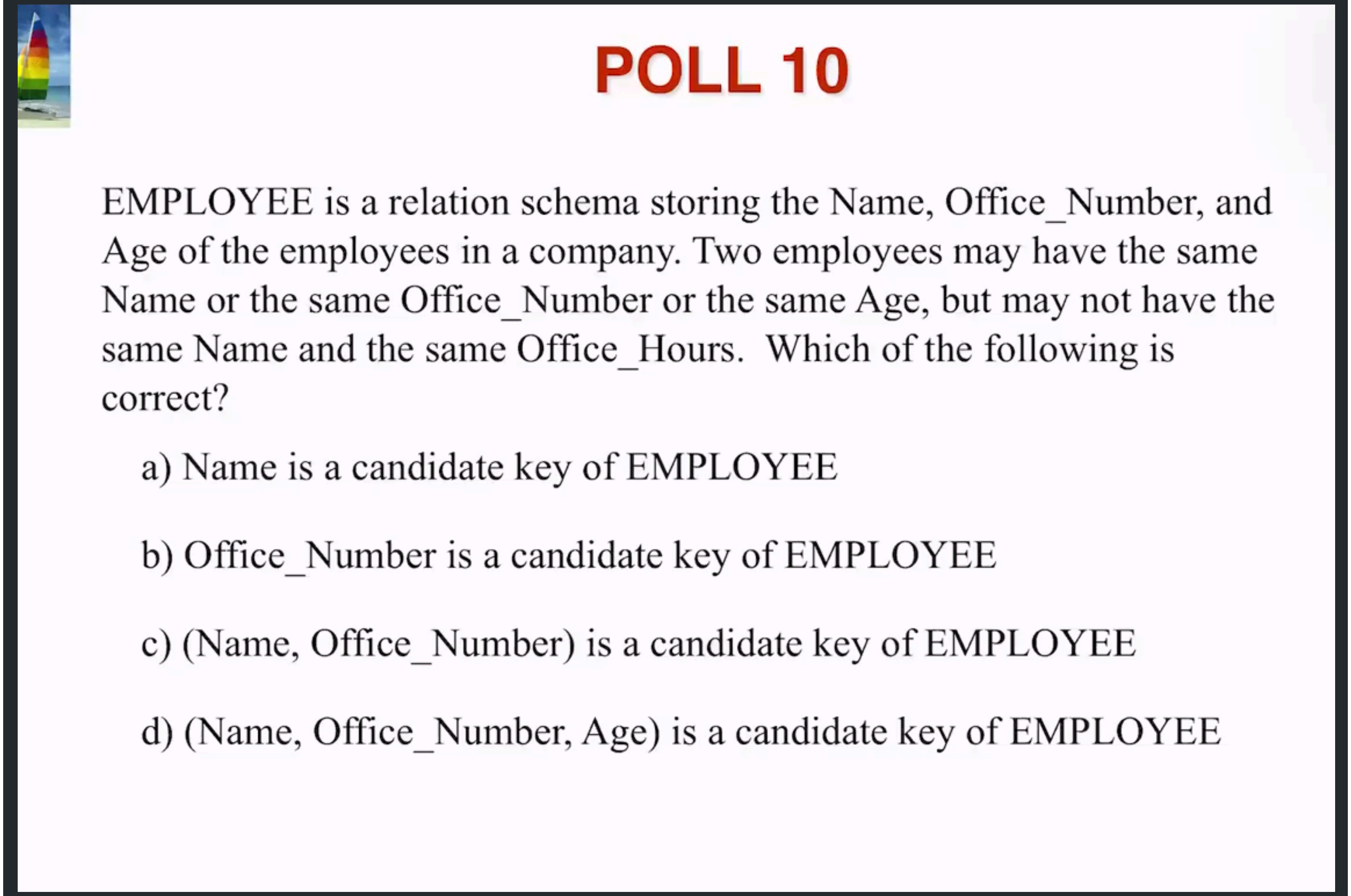
Topic 3 Polls



POLL 10

EMPLOYEE is a relation schema storing the Name, Office_Number, and Age of the employees in a company. Two employees may have the same Name or the same Office_Number or the same Age, but may not have the same Name and the same Office_Hours. Which of the following is correct?

a) Name is a candidate key of EMPLOYEE

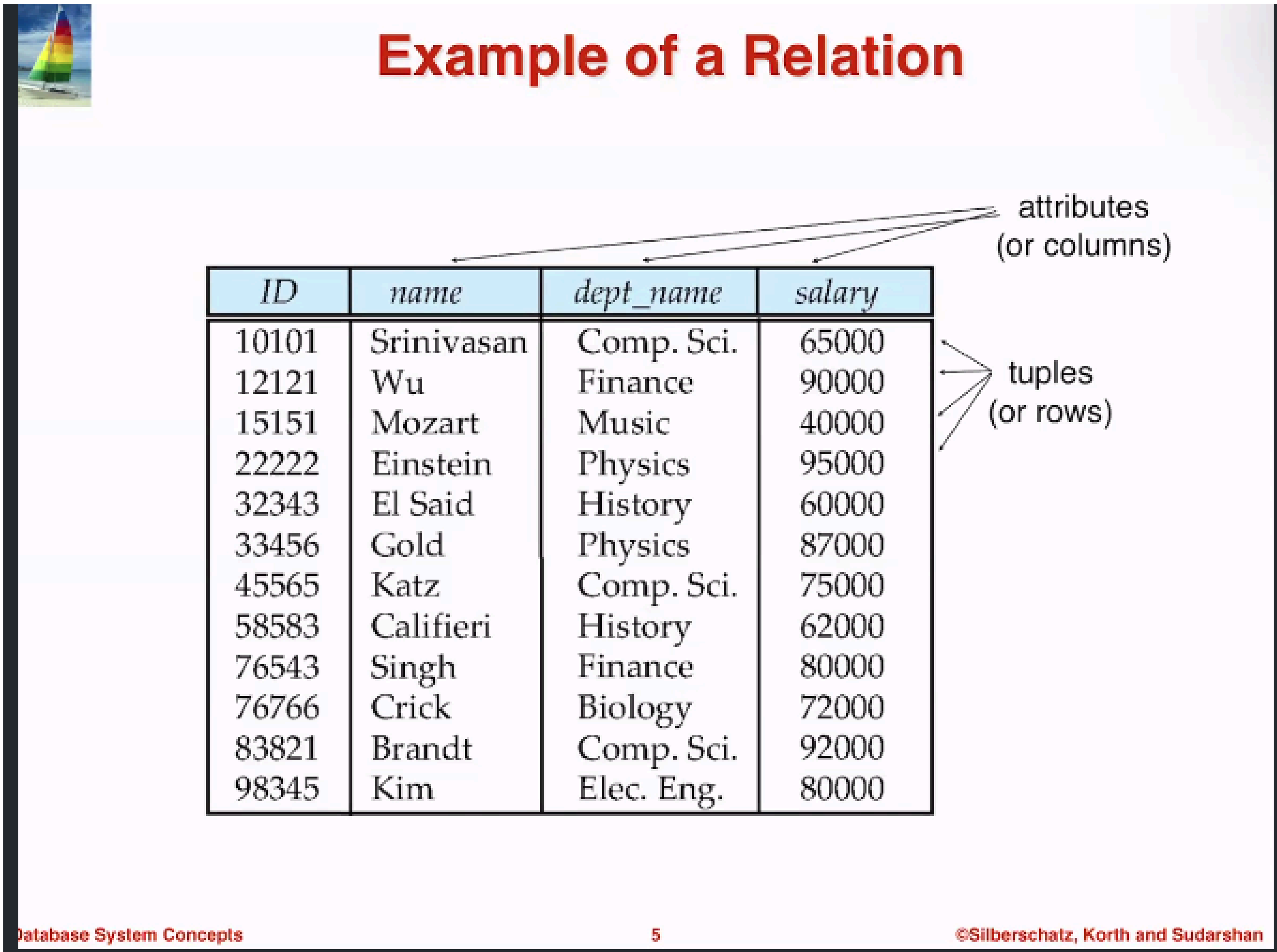
b) Office_Number is a candidate key of EMPLOYEE

c) (Name, Office_Number) is a candidate key of EMPLOYEE

d) (Name, Office_Number, Age) is a candidate key of EMPLOYEE

Lecture 1

Topic 3: Relational Data Model



Example of a Relation

attributes (or columns)

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

tuples (or rows)

Database System Concepts 5 ©Silberschatz, Korth and Sudarshan

This table illustrates the relational data model, where each row is a tuple of its columns, `(ID, name, dept_name, salary)`. A limitation of the relational model is that it doesn't readily depict relationships between tables, making it less expressive than an ER diagram.

Attribute Types

- **Domain**: The set of permitted values for each attribute, also present in the ER model.
- **Atomic**: An atomic attribute cannot be divided further; most attributes must be atomic.
- **Null**: Null is a special value and is always part of every domain, which can complicate operations like joining and selecting.

Relation Schema and Instances

A **relation** is defined as: R(A_1, A_2, ..., A_n), where A_1, A_2, ..., A_n are attributes of an entity set.

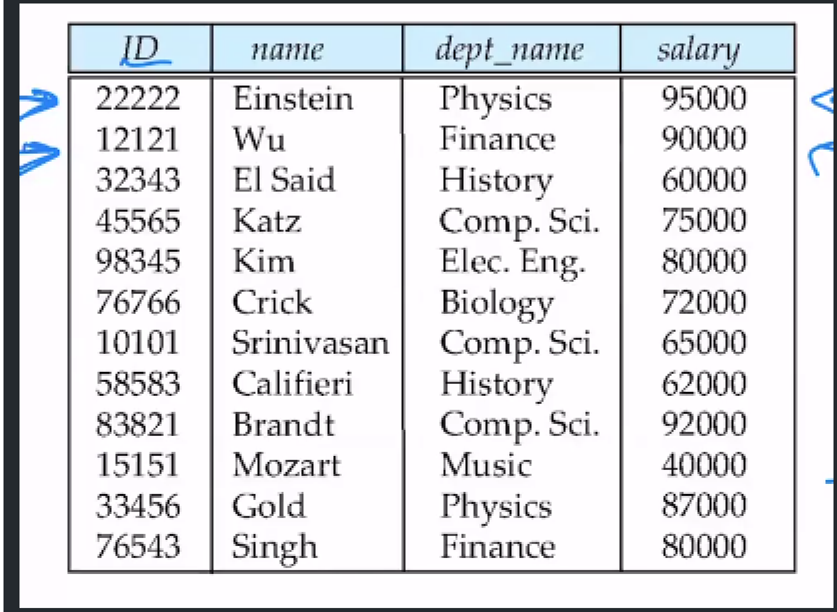
Example: `student(ID, name, major, GPA)`, where `student` is R and its attributes are `ID`, `name`, `major`, and `GPA`.

Relation instance: A snapshot in time or an instance of a relation.

- **Example**: For the `student` relation, instances may include `student(0, Parker, Comp Sci, 5.0)` and `student(1, Jorge, Graphic Design, 7.0)`.

Properties

- Tuples are formally unordered. When a tuple is inserted, the DBMS determines its order.



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

This example shows that there is no clear ordering of the rows (tuples) in the table.

Keys

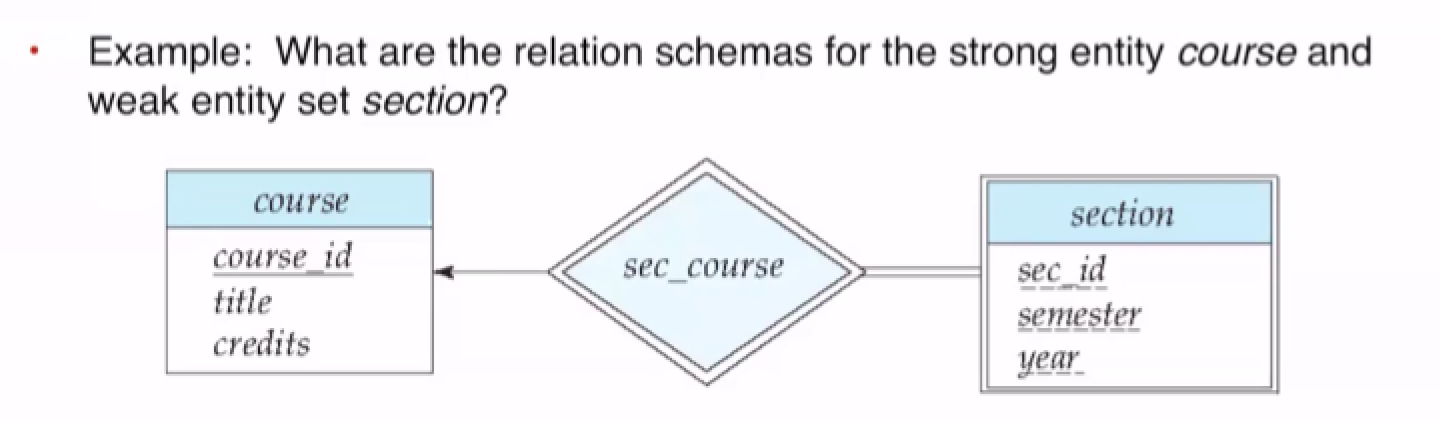
This section was reviewed from Topic 2.

Converting ER Diagrams to Relation Schemas

To convert an ER diagram to a relational model, we create a set of relation schemas, which can be implemented as tables in a DBMS.

Converting Entity Sets with Simple Attributes

- A strong entity set becomes a schema with identical attributes.
- A weak entity set includes its attributes plus the primary key of the strong identifying entity set.



- Example: What are the relation schemas for the strong entity *course* and weak entity set *section*?

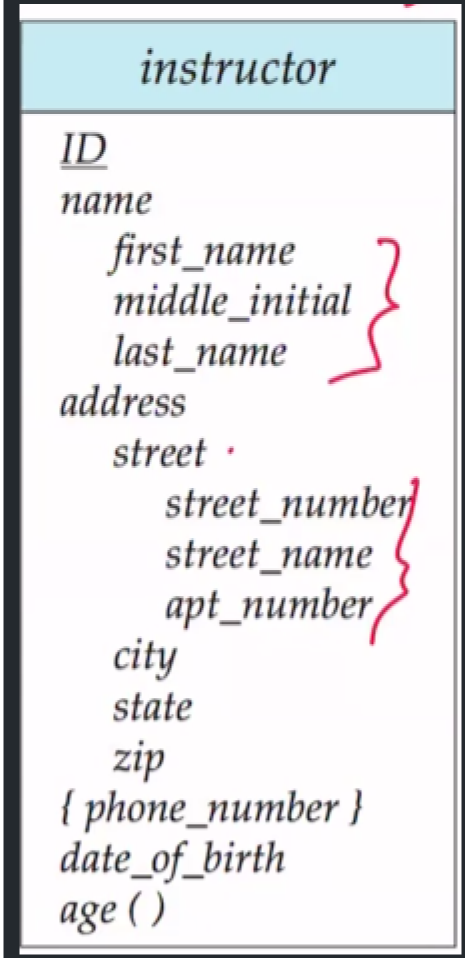
Answers:

1. For `course`: course(course_id, title, credits)
2. For `section`: As a weak entity set, its schema is section(course_id, sec_id, semester, year), with no dotted line and underlined discriminators.

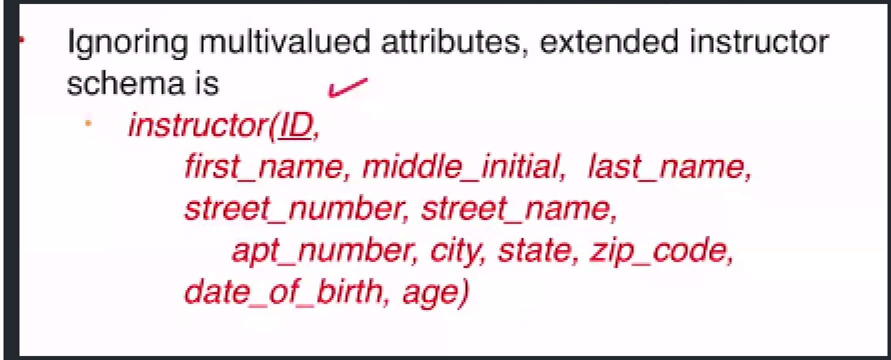
Composite and Multivalued Attributes

Multivalued attributes require multiple relation schemas, unlike simple attributes, which convert directly.

Example



Relation schema without considering multivalued attributes



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)*

- The `name` attribute is omitted, and only attributes of the composite attribute are stored. The same applies to `street`.
- In the relational model, we lose the connection between sub-attributes and their composite attributes, which is clear in the ER model.
- To resolve ambiguity with similarly named sub-attributes, prefixes are added: instructor(ID, name_first_name, name_middle_name, name_last_name, ...).

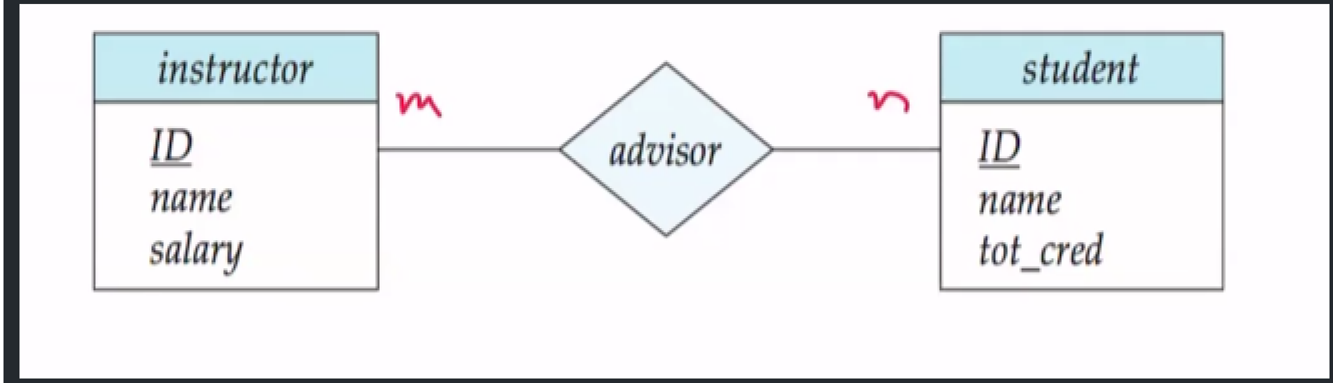
Relation schema considering multivalued attributes

A second schema/tuple is created for multivalued attributes. The ID associates phone numbers with instructors, and both are keys since one instructor can have multiple phone numbers.

Answer: phone_number(ID, phone_number)

Converting Relationship Sets

Many to Many

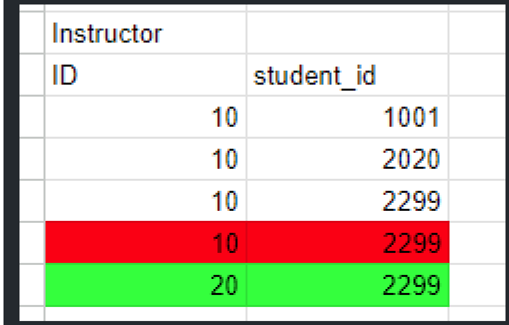


- Convert one entity set at a time, followed by the relationship.

Answer: instructor(ID, name, salary), student(ID, name, total_cred), advisor(instructor_ID, student_ID).

- Both keys are necessary due to the many-to-many relationship, which requires unique identification through two primary keys.

An instructor can advise multiple students, and a student can have multiple advisors, but the same instructor cannot advise the same student more than once.

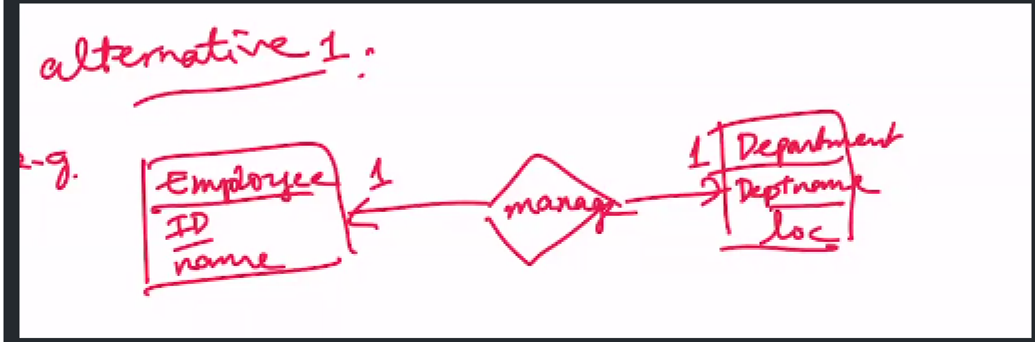


Instructor	
ID	student_id
10	1001
10	2020
10	2299
10	2299
20	2299

This is depicted in this table.
Red means it is not permitted with the relational model, green means it would be okay.

One to One

Example 1: Normal 1-to-1



Employee has an ID and name.
Department has a dept_name and location (loc).
Employees have a one to one relatonship with department, manage.

Answer:

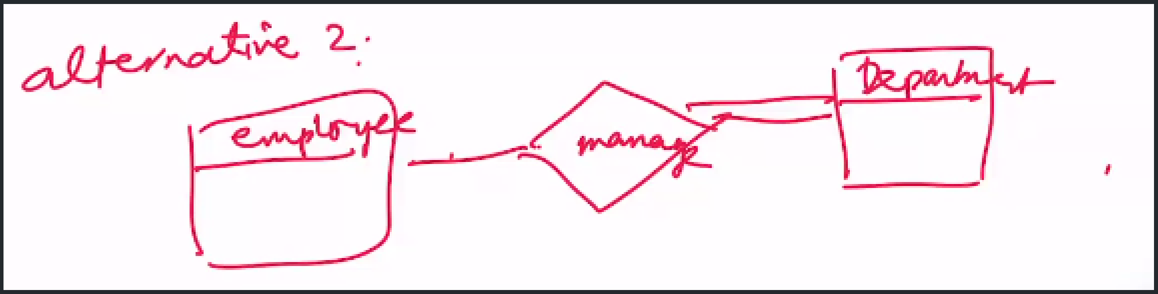
employee(ID, name)
department(dept_name, loc)

1. Create a new relation schema containing both primary keys of employee and department.
2. Choose only one of these keys to use as the primary key for the schema.

manage(ID, dept_name)
manage(ID, dept_name)

Both solutions accomplish the task, however the best solution is the one with the shorter key. So, manage(ID, dept_name) is the best solution.

Example 2: Total Participation (One Entity Set)



From topic 2, total participation means that at *every* department must be managed by *at least one* employee. This is indicated by the double line.

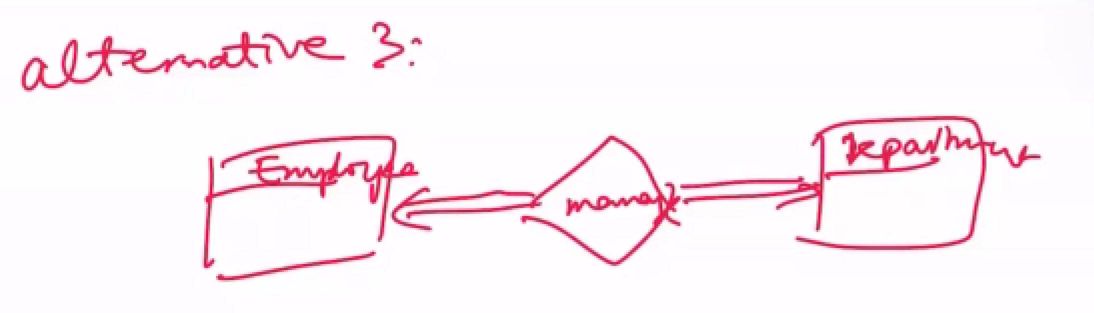
1. If only one of the entity sets have total participation,
2. Create 2 relation schemas, one for each entity set

3. Then include the relation in the entity set with the total relation.

Answer:

employee(ID, name)
department(dept_name, loc, employee_ID)

Example 3: Total Participation (Both Entity Sets)



1. If both the entity sets have total participation,
2. Create 1 relation schema, for either of the entity sets
3. Then include the relation in the entity set with the total relation.

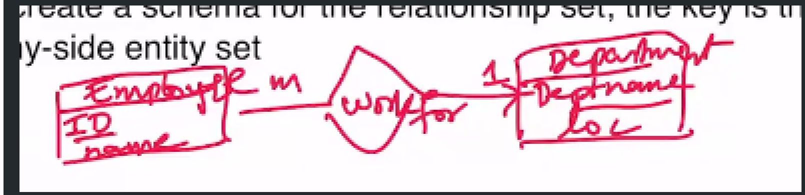
Solution:

employee_department(ID, name, dept_name, loc)
employee_department(ID, name, dept_name, loc)

The better of the two solutions is employee_department(ID, name, dept_name, loc), because `ID` is shorter than `dept_name`.

One-to-Many and Many-to-One

Example 1



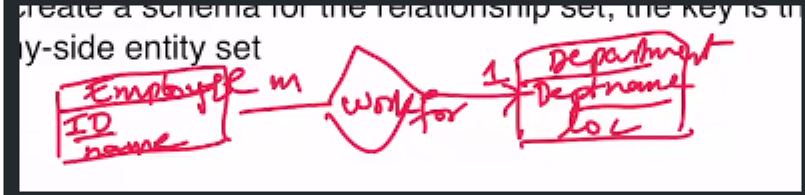
1. Create a relation schema for the relationship
2. Put the key of the many-side entity set into that relation schema

Solution:

employee(ID, name)
department(department_name, loc)

works_for(ID, department_name)

Example 2



reate a schema for the relationship set, the key is th

y-side entity set

Don't create an extra relation schema.

1. Create the relation schema for both entity sets
2. Store the relationship in the many-side entity set.

employee(ID, name, dept_name)
department(department_name, loc)

Converting Relationship-Sets that have Attributes