Database Systems

(CS 355 / CE 373)

Dr. Umer Tariq
Assistant Professor,
Dhanani School of Science & Engineering,
Habib University

Acknowledgements

- Many slides have been borrowed from the official lecture slides accompanying the textbook:

  Database System Concepts, (2019), Seventh Edition,
  Avi Silberschatz, Henry F. Korth, S. Sudarshan
  McGraw-Hill, ISBN 9780078022159

  The original lecture slides are available at:

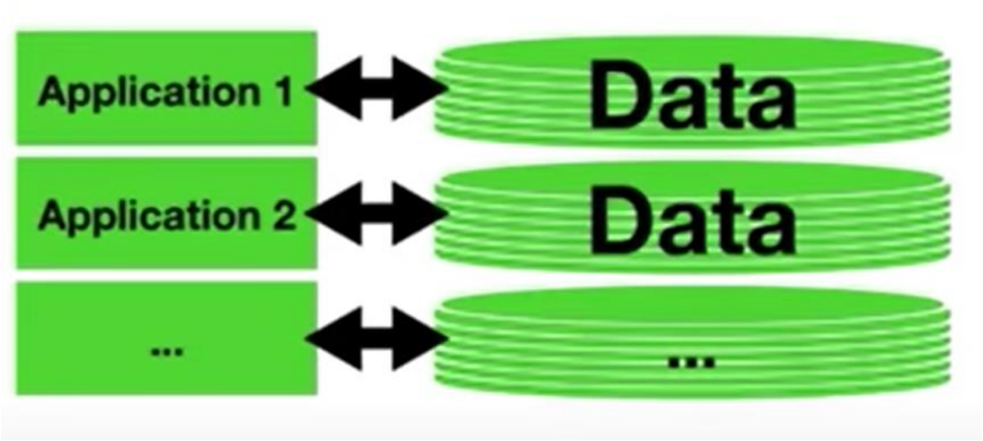
  https://www.db-book.com/

- Some of the slides have been borrowed from the lectures by Dr. Immanuel Trummer (Cornell University). Available at: (www.itrummer.org)

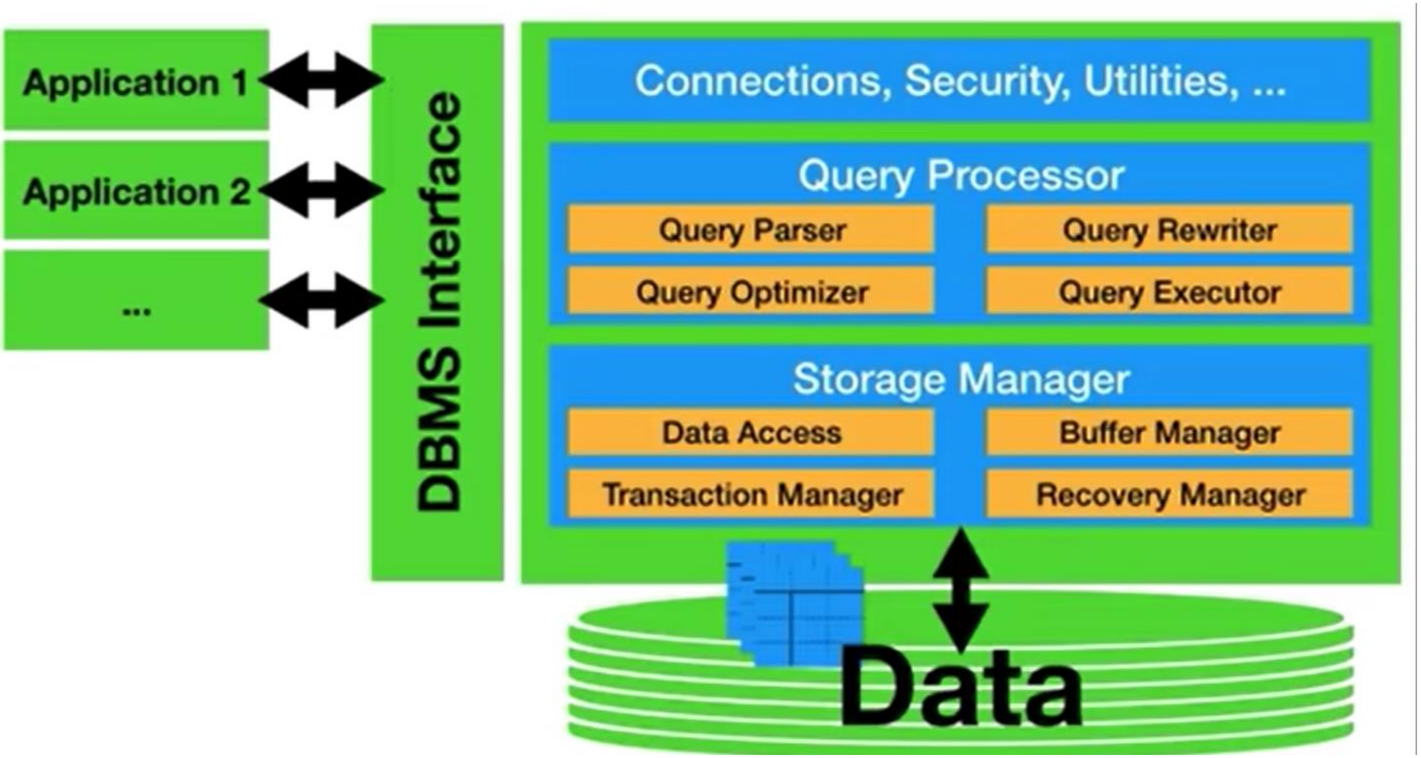
Outline: Week 3

- Database Design Process
- Entity-Relationship Model
- Entity-Relationship Diagrams
- Mapping Cardinalities
- Converting E-R Diagrams to Relational Schemas

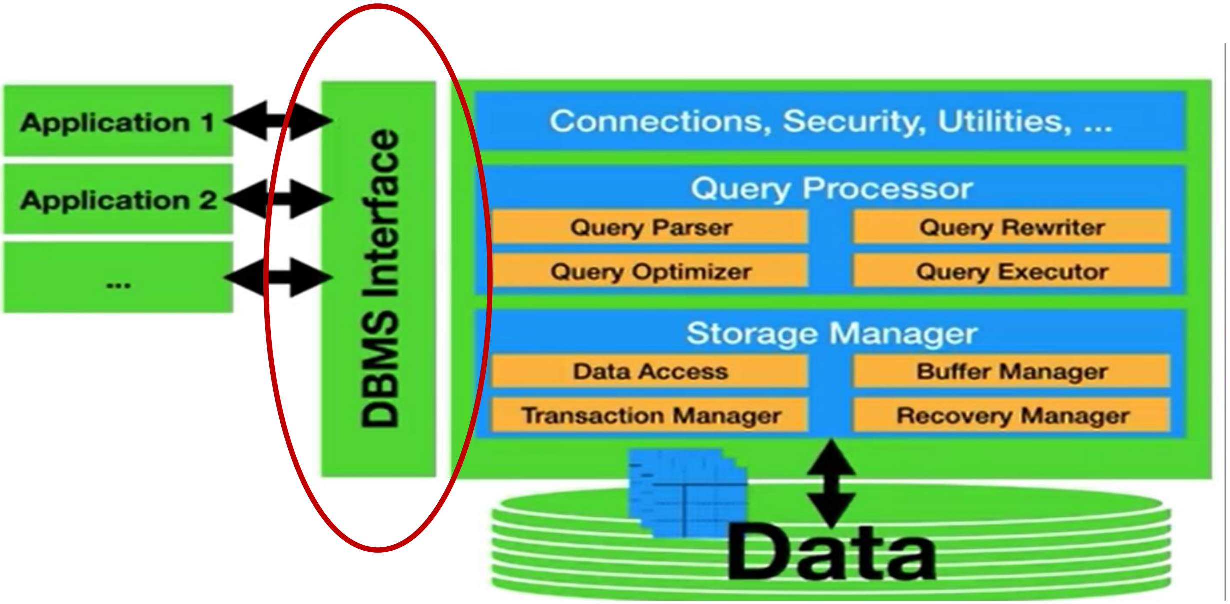
File-Based Approach



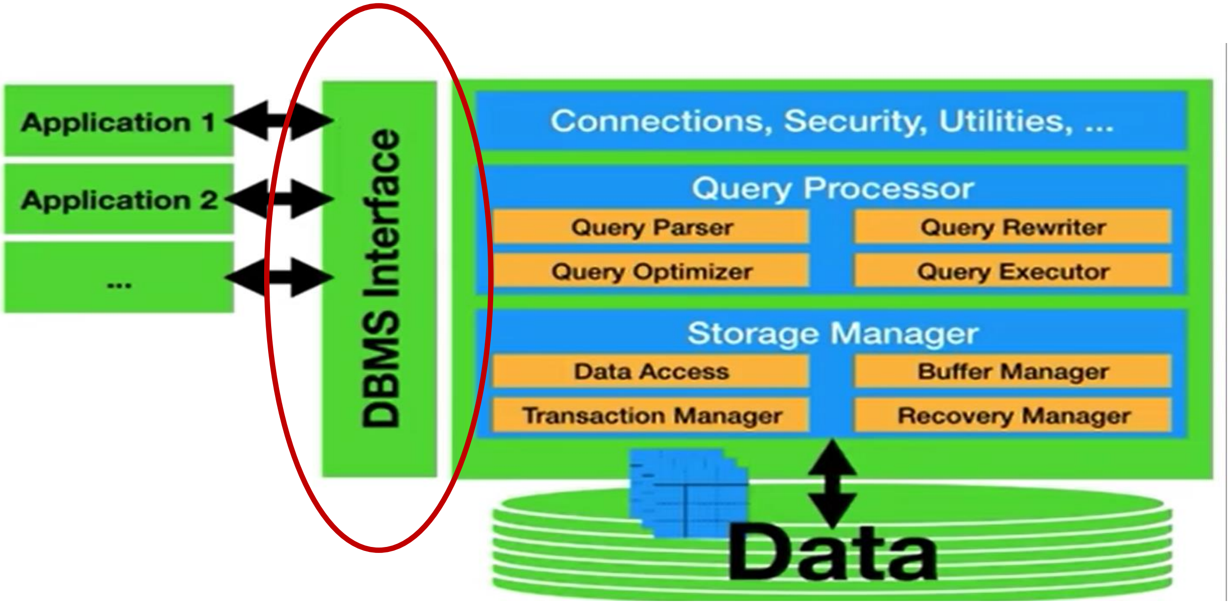
Database Management System (DBMS)



What should be the DBMS Interface?



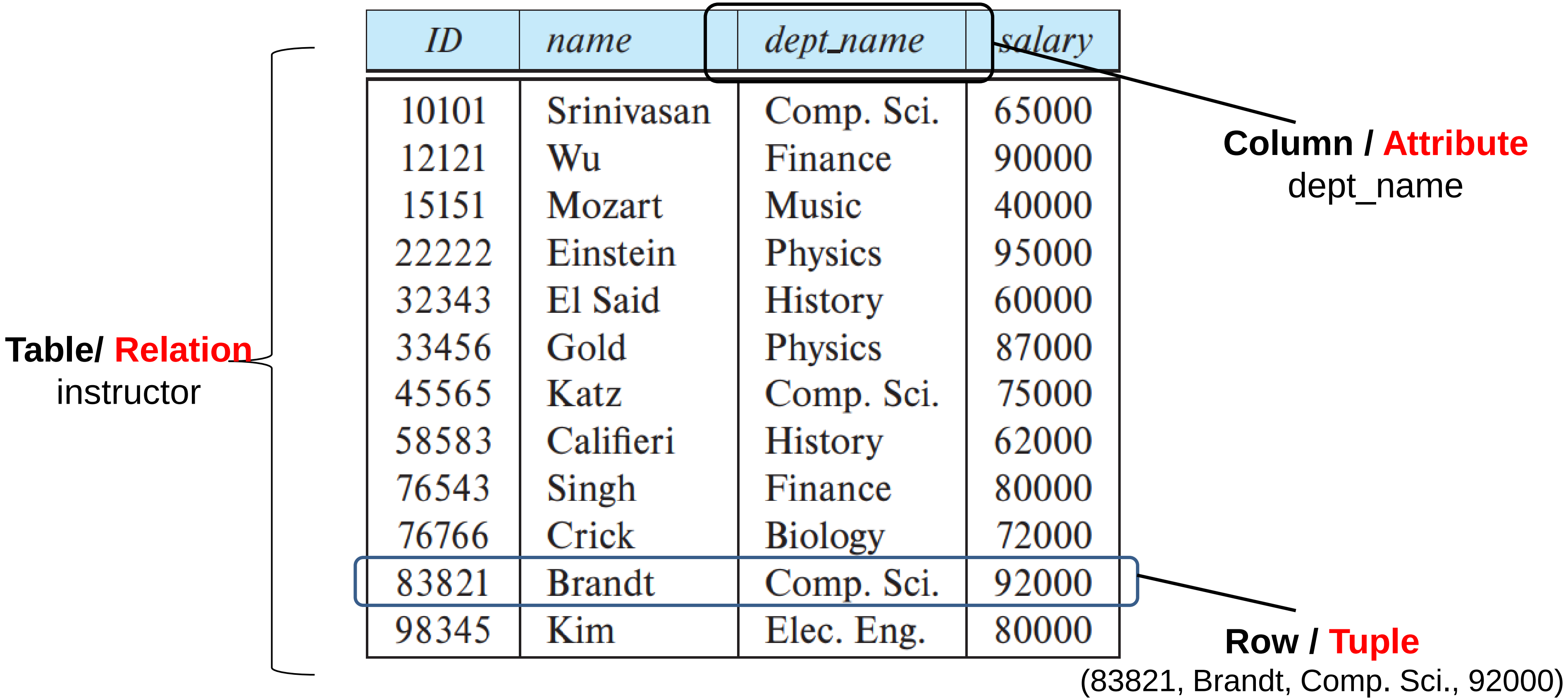
What should be the DBMS Interface?



- Data Model
  - A collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints.

The Relational Model

- The relational model uses a collection of tables to represent both data and the relationships among those data.



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

Table/ Relation
instructor

Column / Attribute
dept_name

Row / Tuple
(83821, Brandt, Comp. Sci., 92000)

Figure 2.1 The *instructor* relation.

Database Application Design

- The task of creating a database application is a complex one, involving:
  - design of the database schema
  - design of the programs that access and update the data
  - design of a security scheme to control access to data
  - design of the user interface of the application

- The needs of the users play a central role in the design process.

Database Application **Design**

- The task of creating a database application is a complex one, involving:
  - **design of the database schema**
  - design of the programs that access and update the data
  - design of a security scheme to control access to data
  - design of the use interface of the application

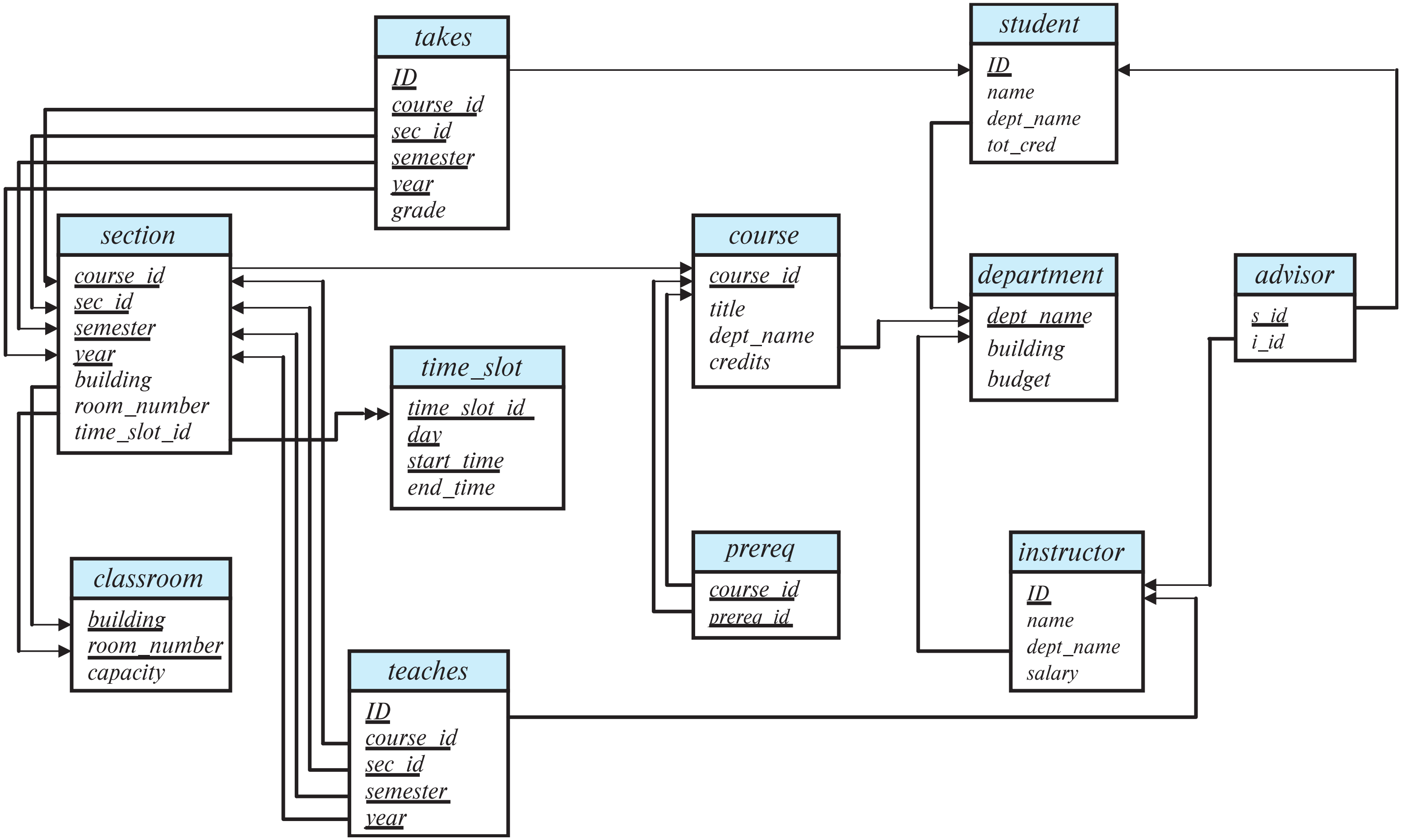
- The needs of the users play a central role in the design process.

Schema Design 101

- Identify Tables
- Identify Columns/Attributes associated with each table
- Identify Primary Keys
- Identify relationships among tables through Foreign Keys

Schema Diagram

- Used to depict a database schema along with primary key and foreign key dependencies



Let’s Focus on the Database Design Process

- What if you are asked to design a database for managing information at
  - Habib University
  - Aga Khan University
  - Aga Khan Hospital

- How would you go about it?

Database Design Process: Small Application vs Real-World Application

- Small Application
  - An application designer may understand the application requirements
  - An application designer may decide directly the relations to be created, their attributes, and constraints

- Real-world Application
  - Highly complex
  - Often no one person understands the complete data needs of an application
  - The database designer must interact with the users of the application to understand the needs of the application
  - The database designer must represent those needs in a high-level fashion that can be understood by the users
  - The database designer must translate those requirements into lower levels of the design.

Database Design Process: Small Application vs Real-World Application

- Small Application
  - An application designer may understand the application requirements
  - An application designer may decide directly the relations to be created, their attributes, and constraints

- Real-world Application
  - Highly complex
  - Often no one person understands the complete data needs of an application
  - The database designer must interact with the users of the application to understand the needs of the application
  - The database designer must represent those needs in a high-level fashion that can be understood by the users
  - The database designer must translate those requirements into lower levels of the design.

The Need for A Higher-Level Data Model than Relations

Entity-Relationship Diagrams

- Real-world Application
  - Highly complex
  - Often no one person understands the complete data needs of an application
  - The database designer must interact with the users of the application to understand the needs of the application
  - The database designer must represent those needs in a high-level fashion that can be understood by the users
  - The database designer must translate those requirements into lower levels of the design.

HOW??

The Need for A Higher-Level Data Model than Relations

Entity-Relationship Diagrams
(A high-level data model that allows the database designer to specify the database structure and evaluate/validate whether it meets the user requirements)

- Real-world Application
  - Highly complex
  - Often no one person understands the complete data needs of an application
  - The database designer must interact with the users of the application to understand the needs of the application
  - The database designer must represent those needs in a high-level fashion that can be understood by the users
  - The database designer must translate those requirements into lower levels of the design.

HOW??

Entity-Relationship Diagram: Informal Introduction

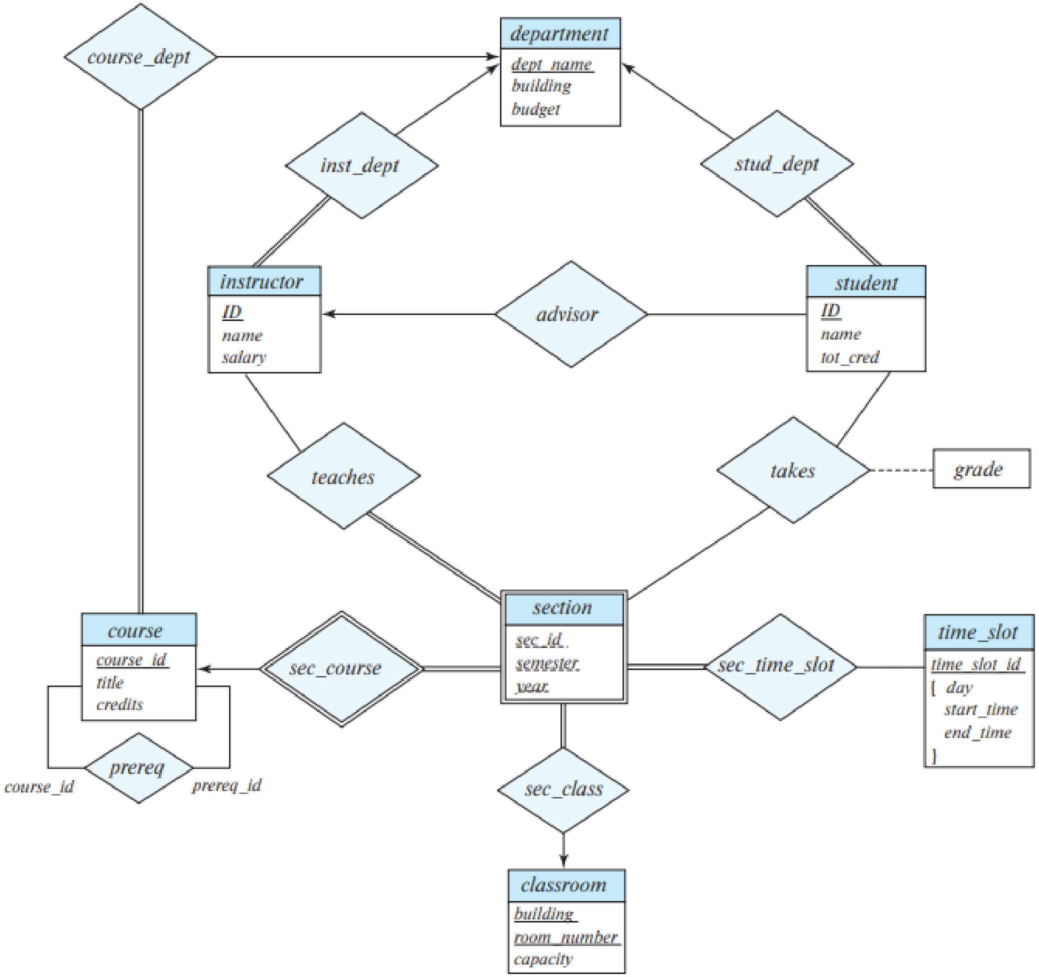


Figure 6.15 E-R diagram for a university enterprise.

Scenario ERD

Entity-Relationship Diagram vs Relational Schema Diagram

1) Translated in 2) TRANSLATION 3)

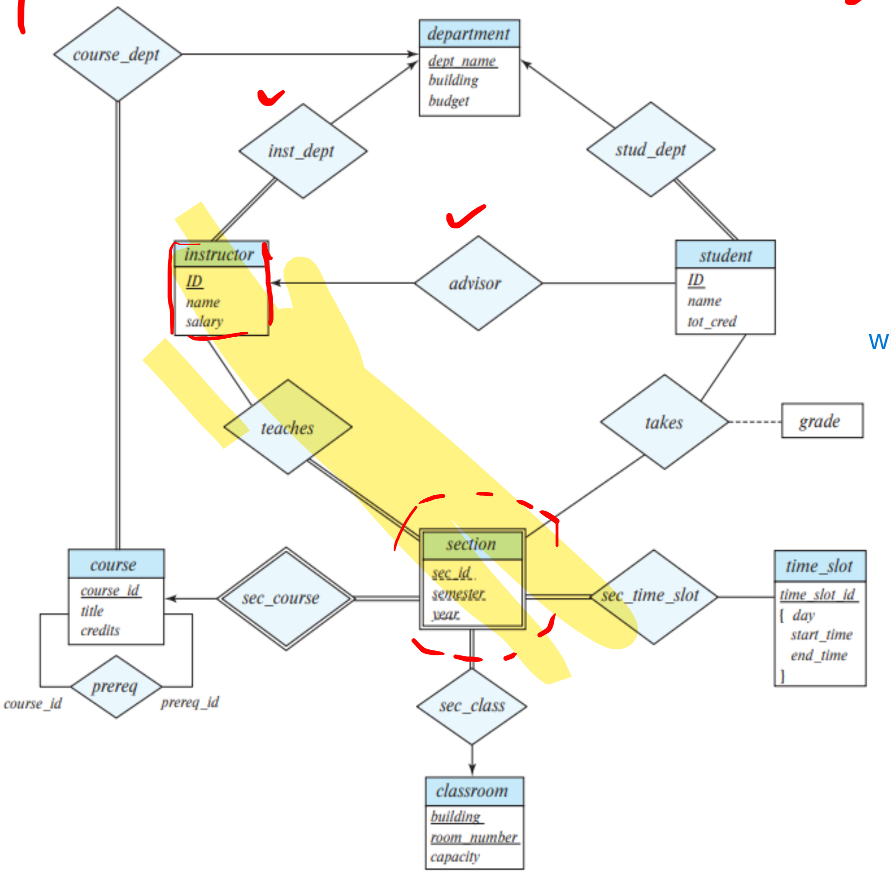
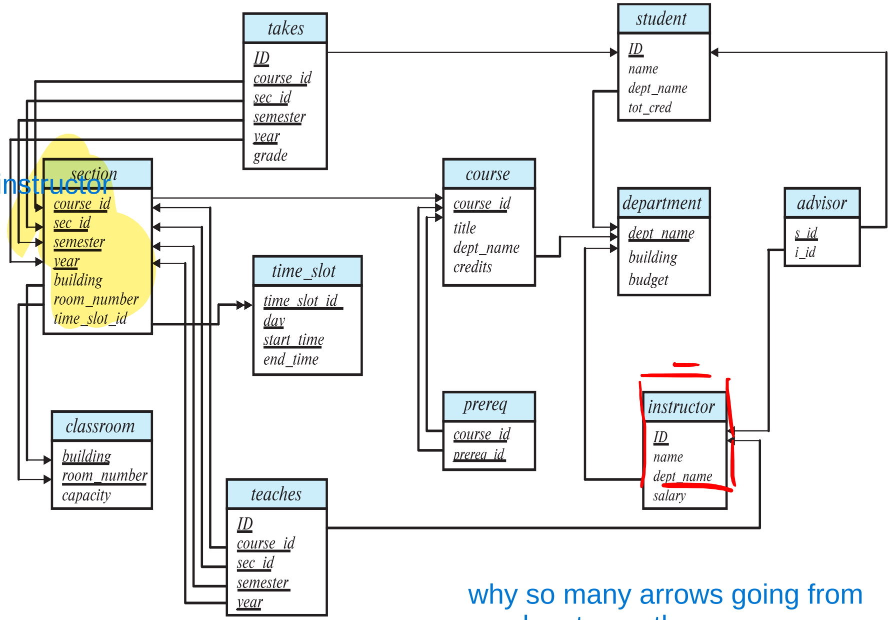


Figure 6.15 ~~E-R diagram~~ for a university enterprise.



where instructor

why so many arrows going from one box to another

Relational Schema

Database Design Process

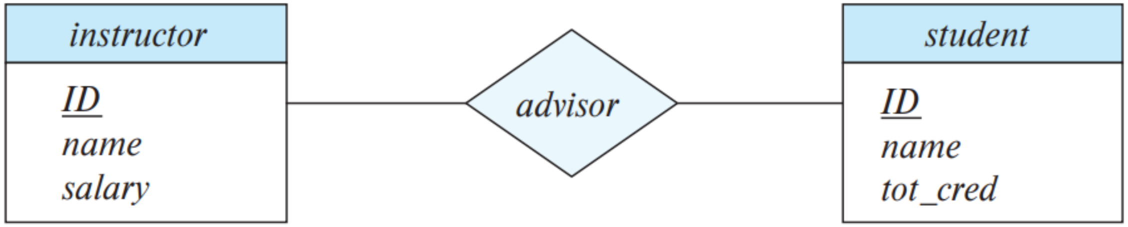
- 1) Specification of User Requirements
  - Involves extensive interaction with the users
- 2) Conceptual Design
  - User requirements are translated into a conceptual schema of the database (such as an **E-R Diagram**)
- Specification of Functional Requirements
  - With the help of users, describe the kind of operations (modifying, searching, retrieving, updating) that will be performed on the data
  - At this stage, the designer can review the conceptual schema to ensure that it meets functional requirements
- Logical Design
  - The designer maps the high-level conceptual schema (such as an **E-R Diagram**) into the implementation data model (such as the **relational data model**)
- Physical Design
  - The designer can specify the physical features of the database (such as the form of file organization) to optimize the performance of the database.

Entity-Relationship (E-R) Model

- E-R Model
  - was developed to facilitate database design
  - models the real-world enterprise as a collection of *entities* and their *relationships*

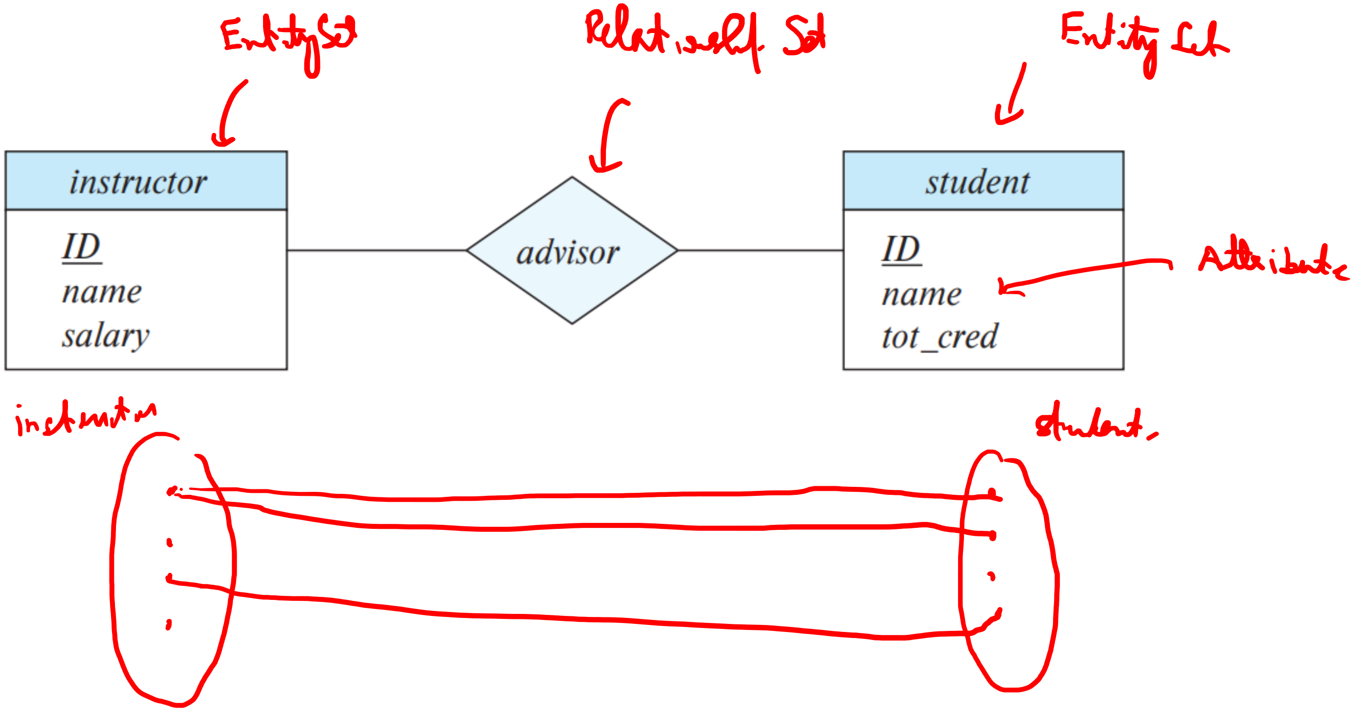
- E-R Model helps in mapping the entities and interaction of real-world enterprises onto the database schema by providing an intermediate abstraction that can be used for effective communication between DB Designer and DB Application User

- The E-R model also has an associated diagrammatic representation
  - **E-R Diagram**



E-R Model: Basic Concepts

- E-R Model employs three basic concepts
  - Entity Sets
  - Relationship Sets
  - Attributes



Entity Set

Relationship Set

Entity Set

instructor

ID
name
salary

advisor

student

ID
name
tot_cred

Attribute

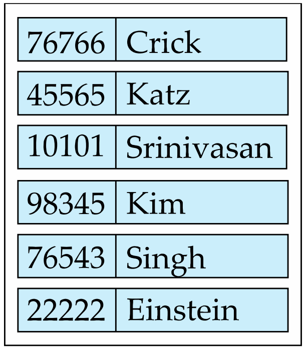
instructor

student

Entity Set

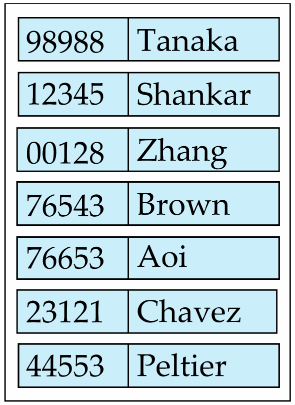
- An ***entity*** is an "object" or a "thing" that exists in the real-world and is distinguishable from all other objects.
  - Examples: a specific instructor, a specific student, a specific course
  - An entity has a set of properties, known as *attributes*.

- An ***entity set*** is a set of entities of the same type that share the same properties.
  - Examples: Set of all instructors, set of all students, set of all courses



76766	Crick
45565	Katz
10101	Srinivasan
98345	Kim
76543	Singh
22222	Einstein

instructor



98988	Tanaka
12345	Shankar
00128	Zhang
76543	Brown
76653	Aoi
23121	Chavez
44553	Peltier

student

Entity Set: Diagrammatic Notation

- An entity (set) is represented as a rectangle in an E-R diagram
  - Attributes are listed inside the rectangle,
  - Primary keys are underlined.

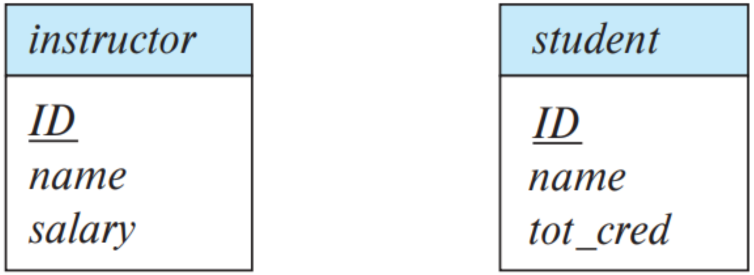
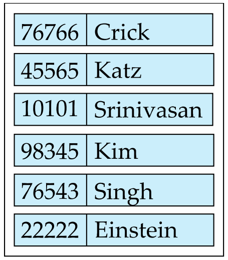
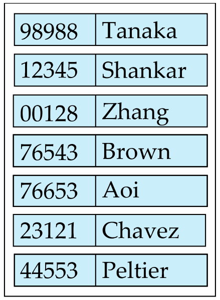


Figure 6.1 E-R diagram showing entity sets *instructor* and *student*.



76766	Crick
45565	Katz
10101	Srinivasan
98345	Kim
76543	Singh
22222	Einstein

instructor



98988	Tanaka
12345	Shankar
00128	Zhang
76543	Brown
76653	Aoi
23121	Chavez
44553	Peltier

student

Relationship Set

- A ***relationship*** is an association among several *entities*.
- A ***relationship set*** is a set of relationships of the same type.

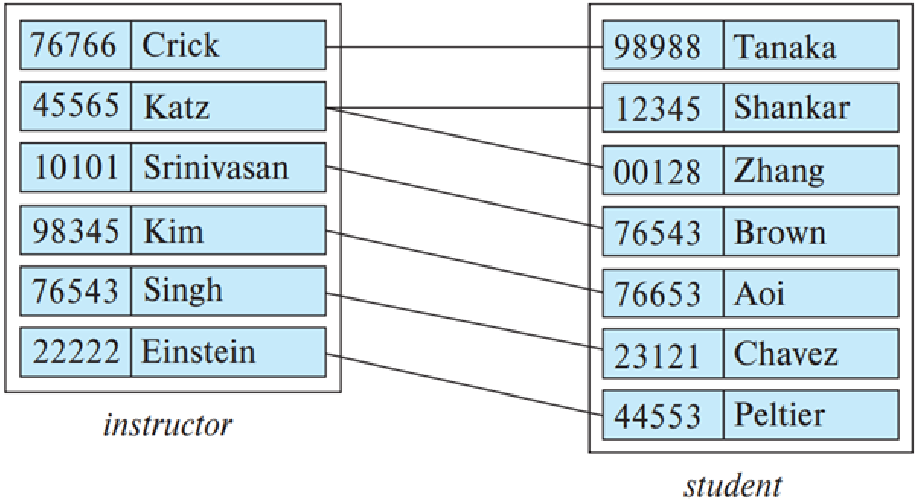


Figure 6.2 Relationship set *advisor* (only some attributes of *instructor* and *student* are shown).

Relationship Set: Diagrammatic Notation

- A relationship set is represented in an E-R diagram by a diamond, which is linked via lines to a number of different entity sets (rectangles).

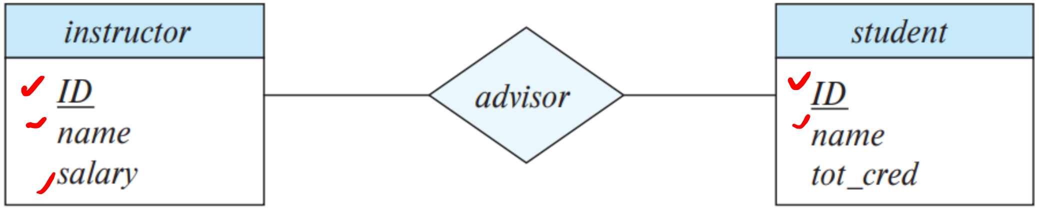


Figure 6.3 E-R diagram showing relationship set *advisor*.

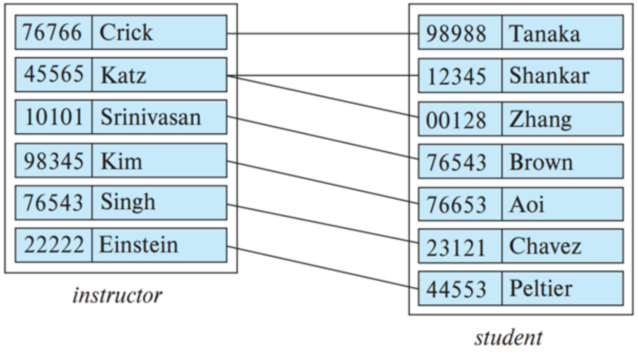
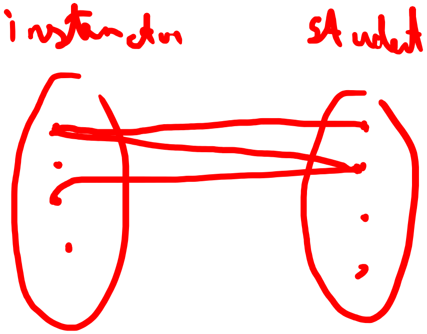


Figure 6.2 Relationship set *advisor* (only some attributes of *instructor* and *student* are shown).



Mapping Cardinalities

- **Mapping cardinalities**, or **cardinality ratios**, express the number of entities to which another entity can be associated via a relationship set.

- For a binary relationship R between entity sets A and B, the mapping cardinality must be one of the following:
  - One-to-one
  - One-to-many
  - Many-to-one
  - Many-to-many

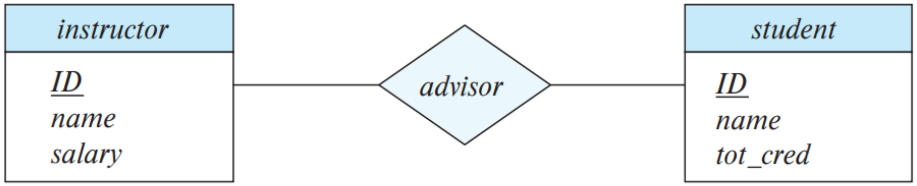
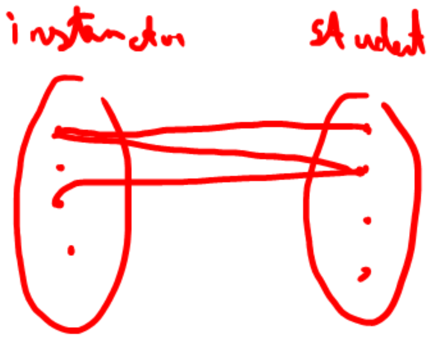
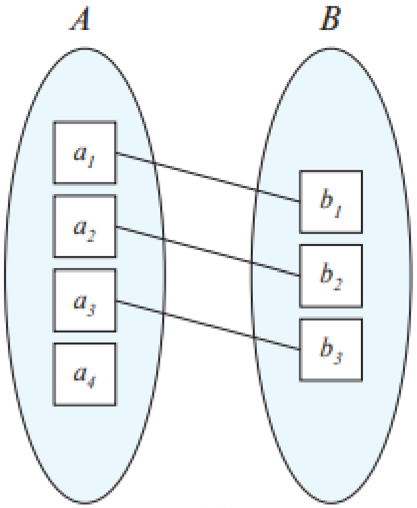


Figure 6.3 E-R diagram showing relationship set *advisor*.



Mapping Cardinalities: One-to-One

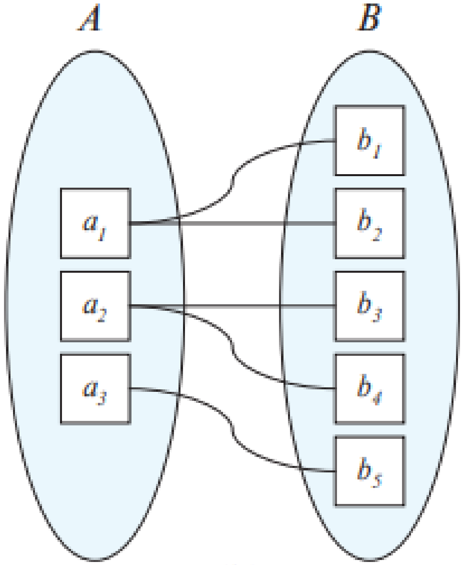
- An entity in A is associated with at most one entity in B, and an entity in B is associated with at most one entity in A.



(a)

Mapping Cardinalities: One-to-Many

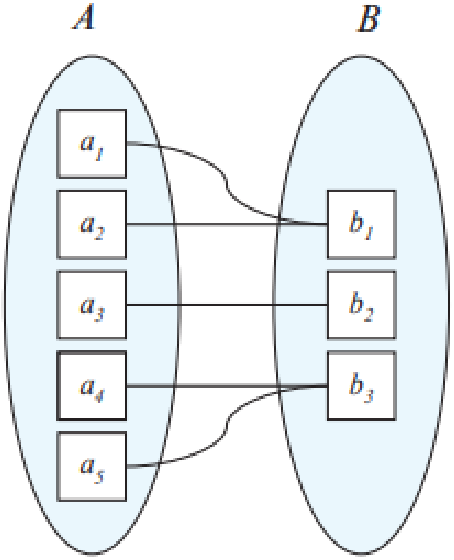
- An entity in A is associated with any number (zero or more) of entities in B. An entity in B, however, can be associated with at most one entity in A.



(b)

Mapping Cardinalities: Many-to-One

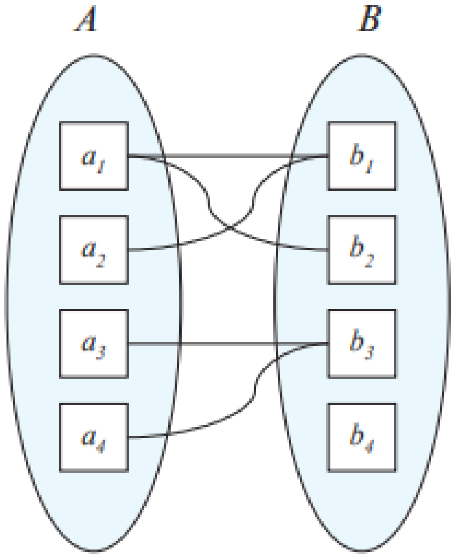
- An entity in A is associated with at most one entity in B. An entity in B, however, can be associated with any number (zero or more) of entities in A.



(a)

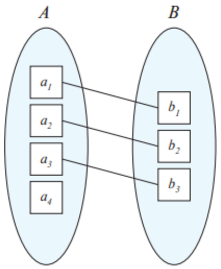
Mapping Cardinalities: Many-to-Many

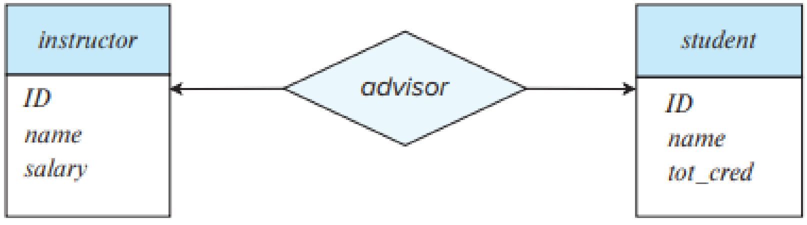
- An entity in A is associated with any number (zero or more) of entities in B, and an entity in B is associated with any number (zero or more) of entities in A.



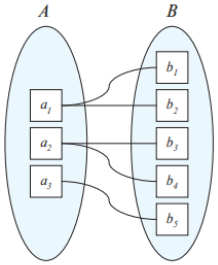
(b)

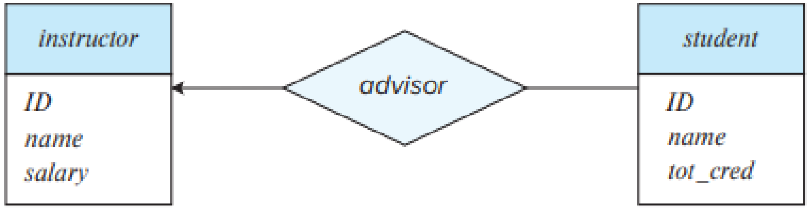
Mapping Cardinalities: Diagrammatic Notation



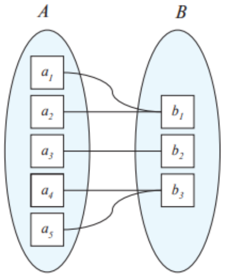


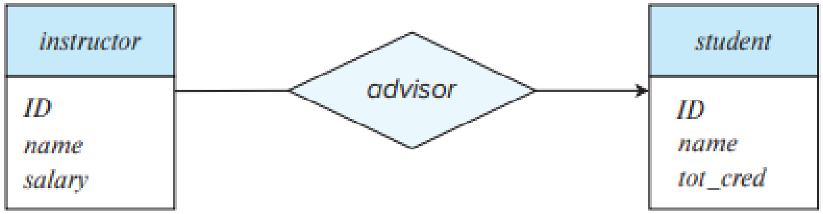
(a) One-to-one



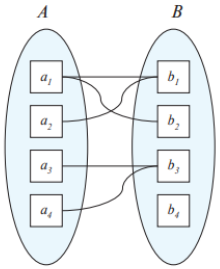


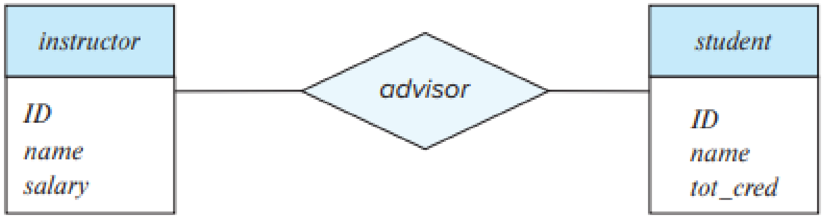
(b) One-to-many





(c) Many-to-one

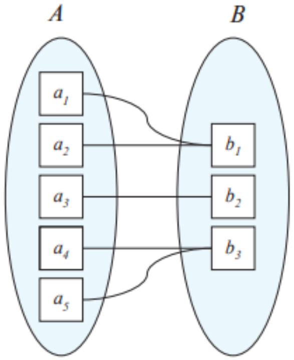




(d) Many-to-many

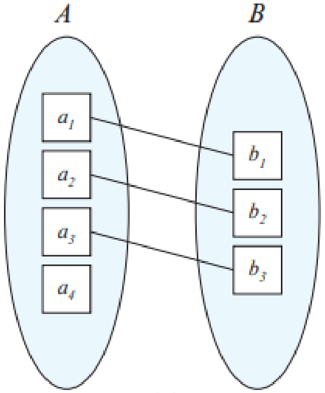
Total Participation vs Partial Participation

- Total Participation
  - The participation of an entity set E in a relationship set R is said to be total if every entity in E must participate in at least one relationship in R.



(a)

- Partial Participation
  - If it is possible that some entities in E do not participate in relationships in R, the participation of entity set E in relationship R is said to be partial.



(a)

Total vs Partial Participation: Diagrammatic Notation

- Total Participation is represented through the use of double lines in E-R Diagram

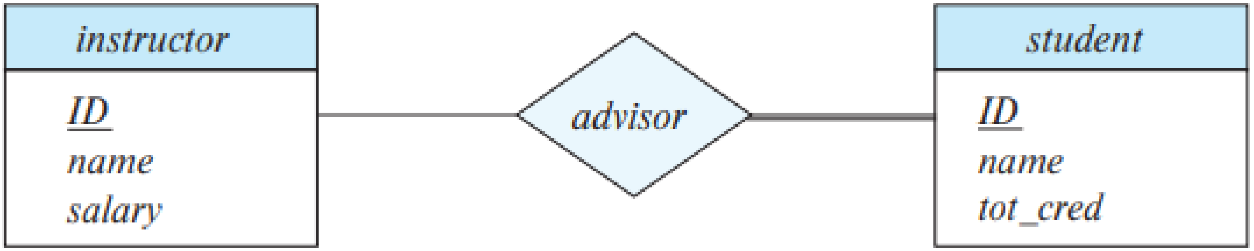
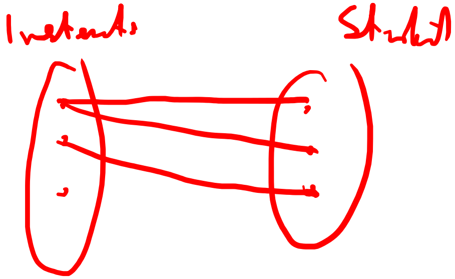


Figure 6.12 E-R diagram showing total participation.



Capturing More Complex Constraints on Participation

- E-R diagrams provide ways to indicate more complex constraints on the number of times each entity (of an entity set) may participate in a relationship.

- A line may have an associated minimum and maximum cardinality, shown as l..h, where l is the minimum and h is the maximum cardinality.

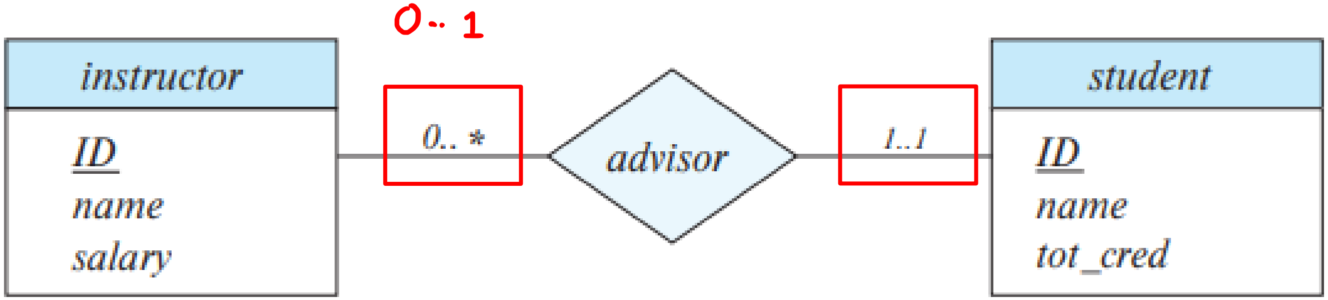
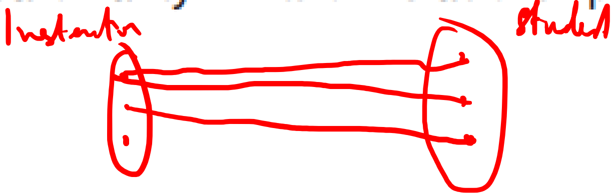


Figure 6.13 Cardinality limits on relationship sets.



Capturing More Complex Constraints on Participation

- A minimum value of 1 indicates that each entity in the entity set occurs in at least one relationship in that relationship set.
- A maximum value of 1 indicates that each entity can participate in at most one relationship in the relationship set.
- A maximum value ∗ indicates that each entity can participate in unlimited relationships in the relationship set.

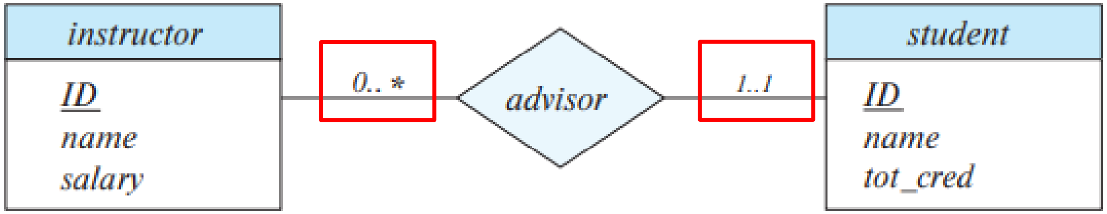
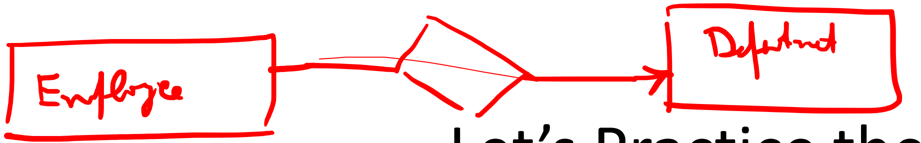
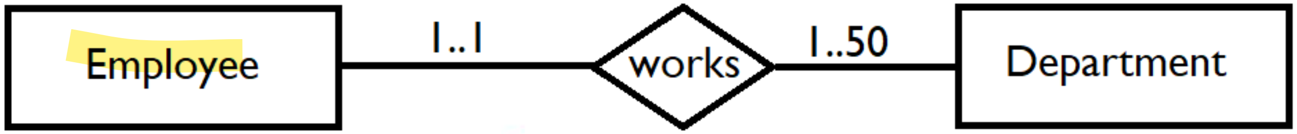


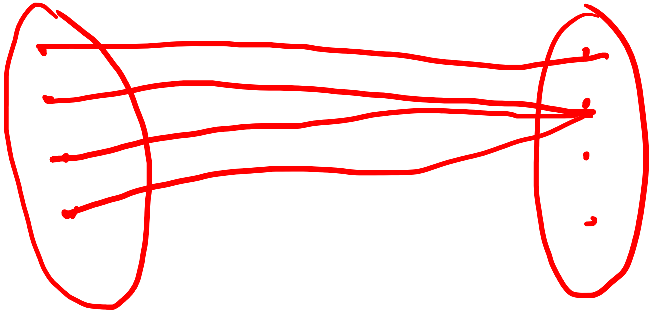
Figure 6.13 Cardinality limits on relationship sets.



Let’s Practice the Notation?


Employee
1..1
works
1..50
Department

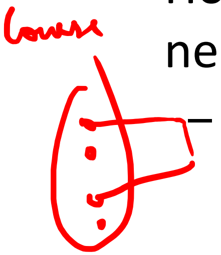
- Can you interpret this?



Many to One

Role of An Entity in a Relationship

- The function that an entity plays in a relationship is called that entity's ***role***.
- Since entity sets participating in a relationship set are generally distinct, roles are implicit and are not usually specified.
- However, specification of roles is useful when the meaning of a relationship needs clarification.
  - Example: When the entity sets of a relationship are not distinct.



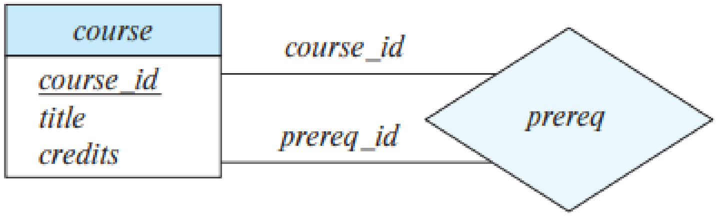


Figure 6.4 E-R diagram with role indicators.

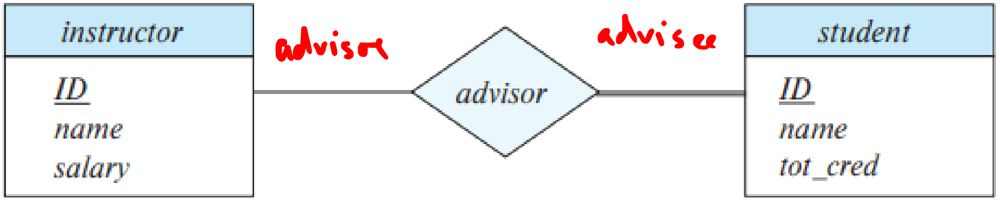
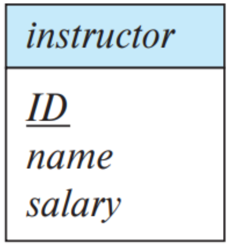


Figure 6.12 E-R diagram showing total participation.

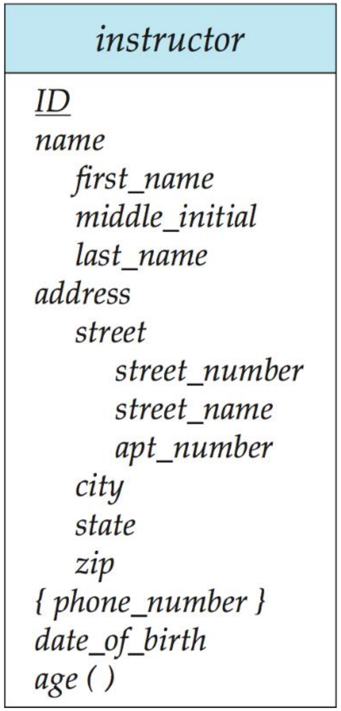
Attributes of an Entity Set: Simple vs Composite

- Simple attributes are not divided into sub-parts.
- Composite attributes can be divided into sub-parts.
  - Example: attribute *name* can be structured as a composite attribute, consisting of *first_name*, *middle_initial*, and *last_name*



instructor

ID
name
salary



instructor

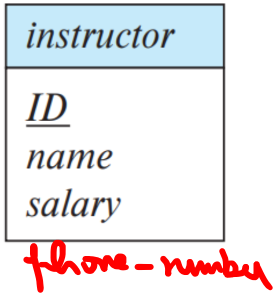
ID
name
  first_name
  middle_initial
  last_name
address
  street
    street_number
    street_name
    apt_number
  city
  state
  zip
{ phone_number }
date_of_birth
age ( )

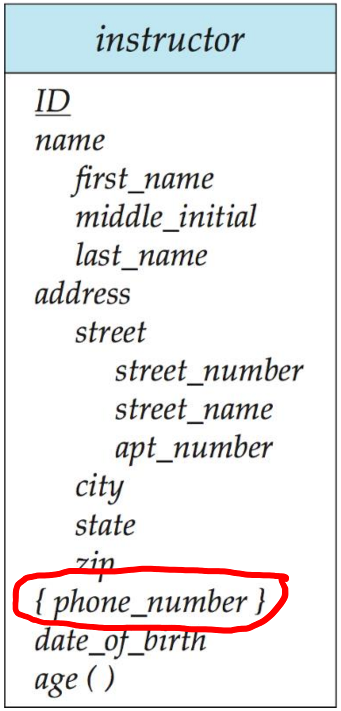
'ID' is a 'simple' attribute

'name' is a composite attribute

Attributes of an Entity Set: Single-Valued vs Multiple-Valued

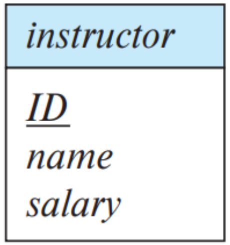
- Single valued attributes have a single value for a particular entity. For example, Student ID.
- Multivalued attributes may have zero, one or several values. For example, phone number attribute for an instructor.





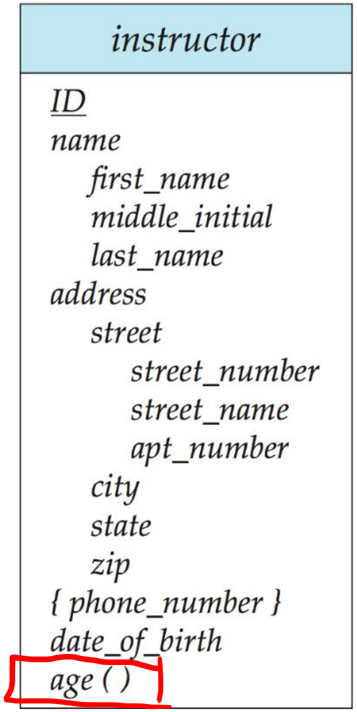
Attributes of an Entity Set: Derived Attributes

- Derived attribute: An attribute whose value can be derived from the values of other related attributes and entities.
  - Example: If the instructor entity set has both *date_of_birth* and *age* attributes, the *age* can be calculated from date_of_birth and the current date. Thus, *age* is a derived attribute.



instructor

- *ID*
- *name*
- *salary*



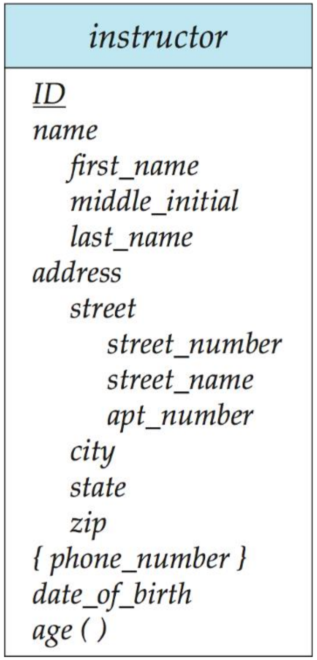
instructor

- *ID*
- *name*
  - *first_name*
  - *middle_initial*
  - *last_name*
- *address*
  - *street*
    - *street_number*
    - *street_name*
    - *apt_number*
  - *city*
  - *state*
  - *zip*
- *{ phone_number }*
- *date_of_birth*
- *age ( )*

Attributes of an Entity Set: Null Value

- An attribute may have a **null** value for an entity, indicating one of the three conditions:
  - Not Applicable: the value does not exist for the entity
  - Missing: the value does exist, but we do not have that information
  - Not known: We do not know whether or not the value actually exists.

- Consider the following situations:
  - An instructor has no middle name
  - A null value for the first_name attribute of the instructor.
  - A null value for the apt_number



instructor

- *ID*
- *name*
  - *first_name*
  - *middle_initial*
  - *last_name*
- *address*
  - *street*
    - *street_number*
    - *street_name*
    - *apt_number*
  - *city*
  - *state*
  - *zip*
- *{ phone_number }*
- *date_of_birth*
- *age ( )*

Attributes of a Relationship Set

- A relationship set may also have attributes, called **descriptive attributes**.
- A relationship set may have multiple descriptive attributes

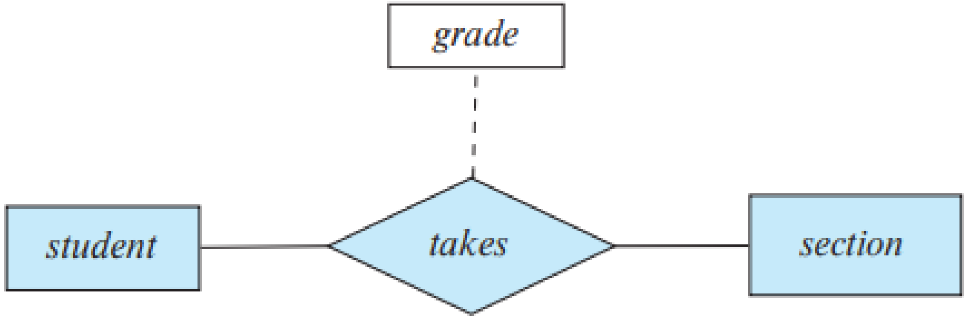
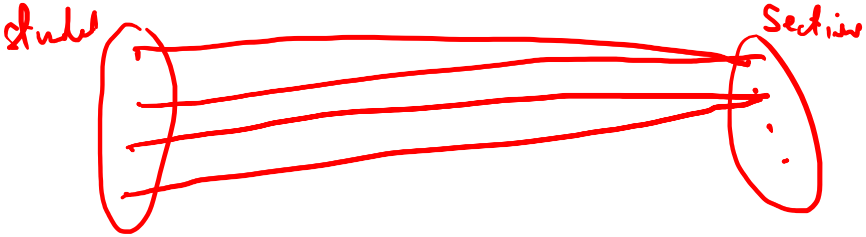


Figure 6.5 E-R diagram with an attribute attached to a relationship set.



Attributes of a Relationship Set

- A relationship set may also have attributes, called **descriptive attributes**.
- A relationship set may have multiple descriptive attributes

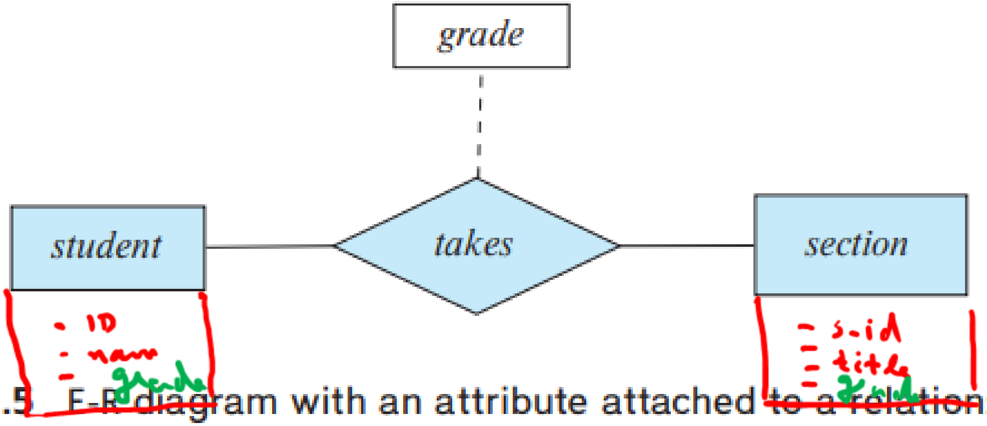
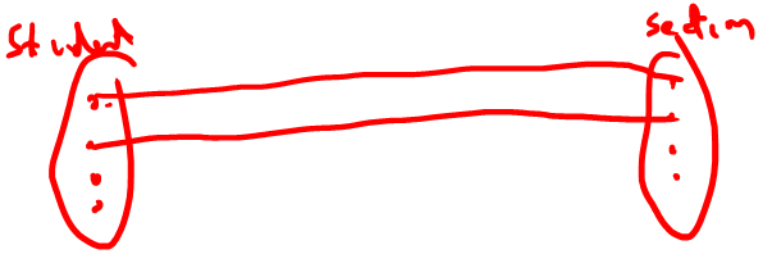


Figure 6.5 E-R diagram with an attribute attached to a relationship set.



Entity-Relationship Model: Concept of Keys

- The concepts of superkey, candidate key, and primary key are applicable to the entity sets (similar to the way they were defined for relation schemas)

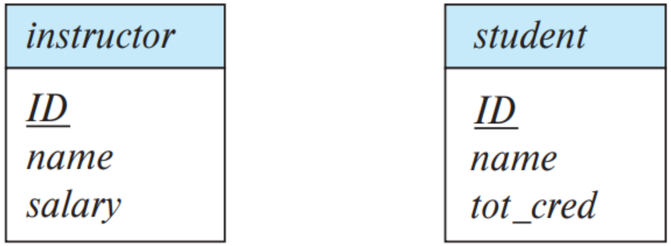


Figure 6.1 E-R diagram showing entity sets *instructor* and *student*.

Weak Entity Set

- A **weak entity set** is one whose existence is dependent on another entity set, called its identifying entity set
  - Example: Can a section exist without a course

- Instead of associating a primary key with a weak entity, we use the primary key of the identifying entity, along with extra attributes, called discriminator attributes to uniquely identify a weak entity.

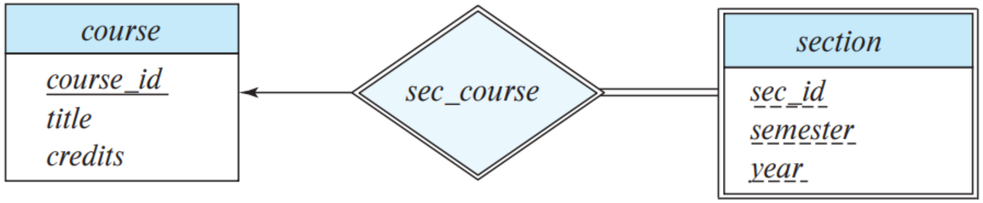


Figure 6.14 E-R diagram with a weak entity set.

E-R Diagrams: Alternative Notations

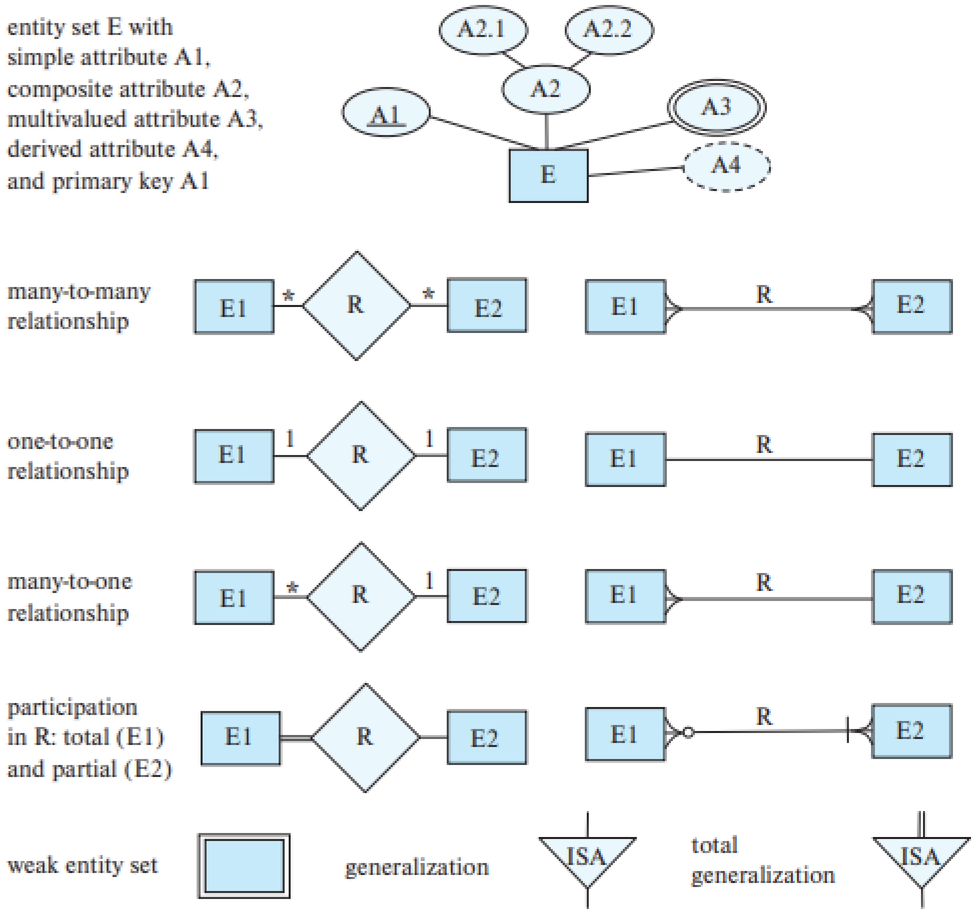


Figure 6.27 Alternative E-R notations.

E-R Design Issues: Use of Entity Sets vs Attributes

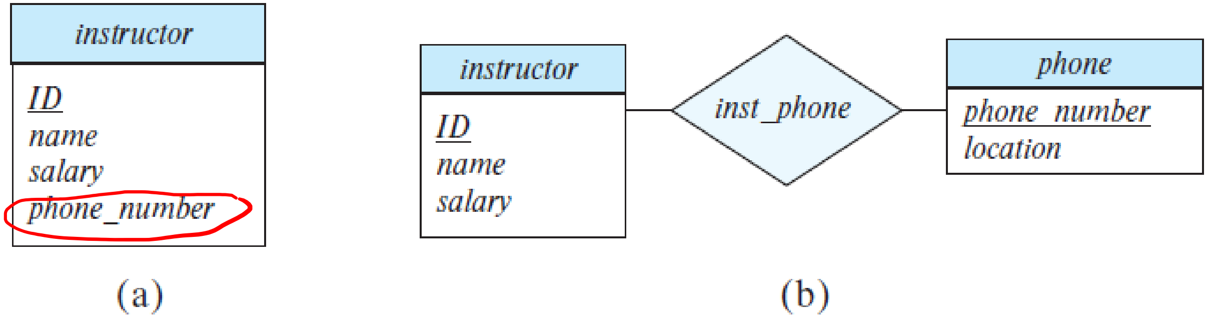
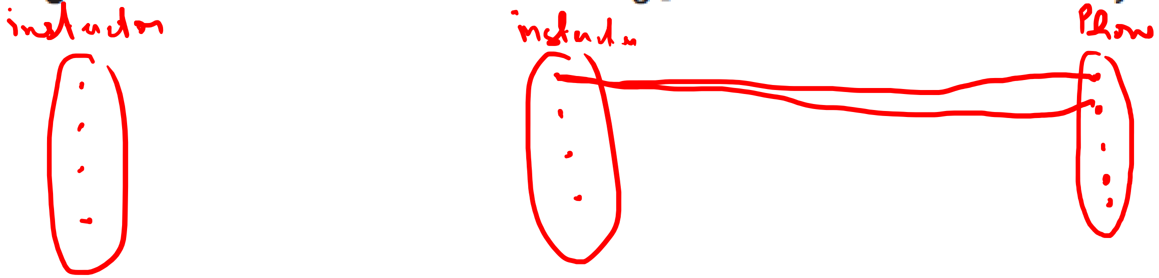


Figure 6.23 Alternatives for adding *phone* to the *instructor* entity set.



E-R Design Issues: Use of Entity Sets vs Relationship Sets

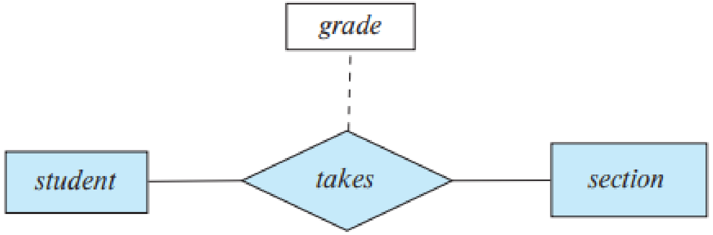


Figure 6.5 E-R diagram with an attribute attached to a relationship set.

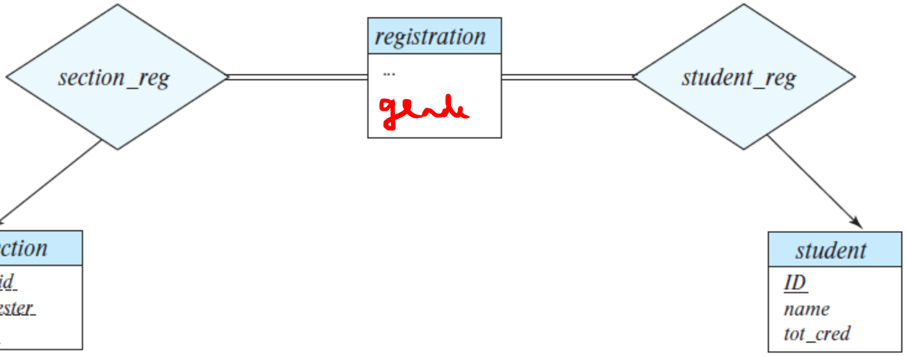
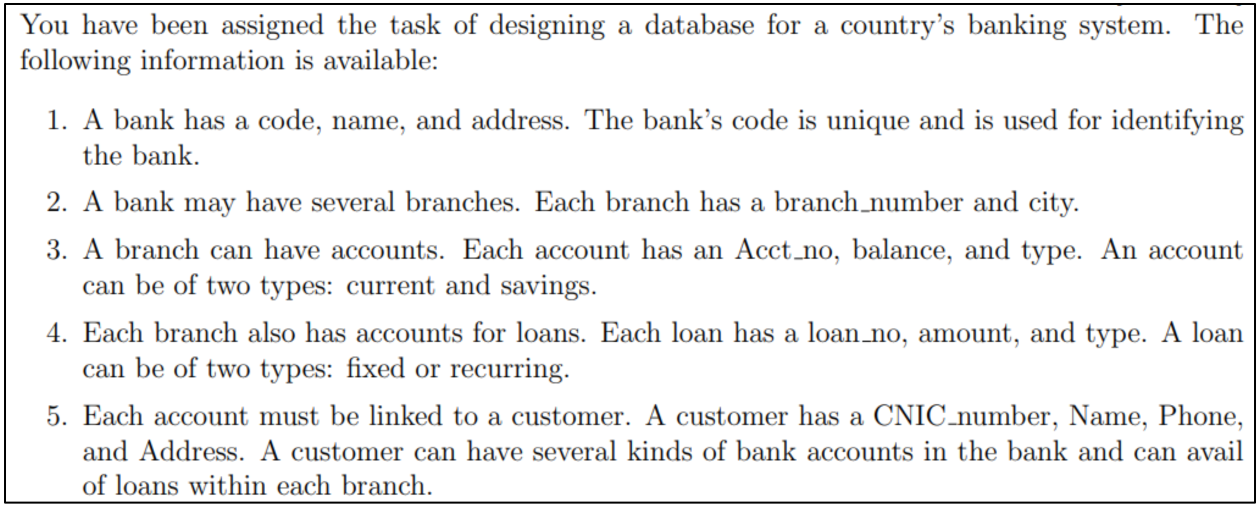


Figure 6.24 Replacement of *takes* by *registration* and two relationship sets.

Developing E-R Diagram: Example



You have been assigned the task of designing a database for a country's banking system. The following information is available:

1. A bank has a code, name, and address. The bank's code is unique and is used for identifying the bank.
2. A bank may have several branches. Each branch has a branch_number and city.
3. A branch can have accounts. Each account has an Acct_no, balance, and type. An account can be of two types: current and savings.
4. Each branch also has accounts for loans. Each loan has a loan_no, amount, and type. A loan can be of two types: fixed or recurring.
5. Each account must be linked to a customer. A customer has a CNIC_number, Name, Phone, and Address. A customer can have several kinds of bank accounts in the bank and can avail of loans within each branch.

Developing E-R Diagrams: Example

a) Identify Entity Sets

b) Identify Relationship Sets
(and their Cardinalities)

c) Attributes

1)

2) Bank

3) Branch

4) Loan Account

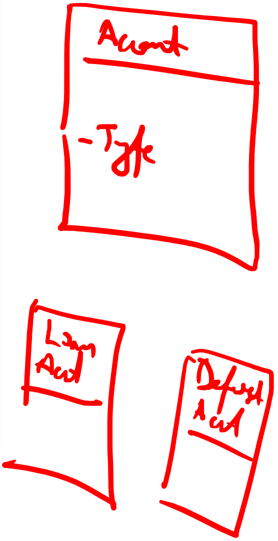
Deposit Acount

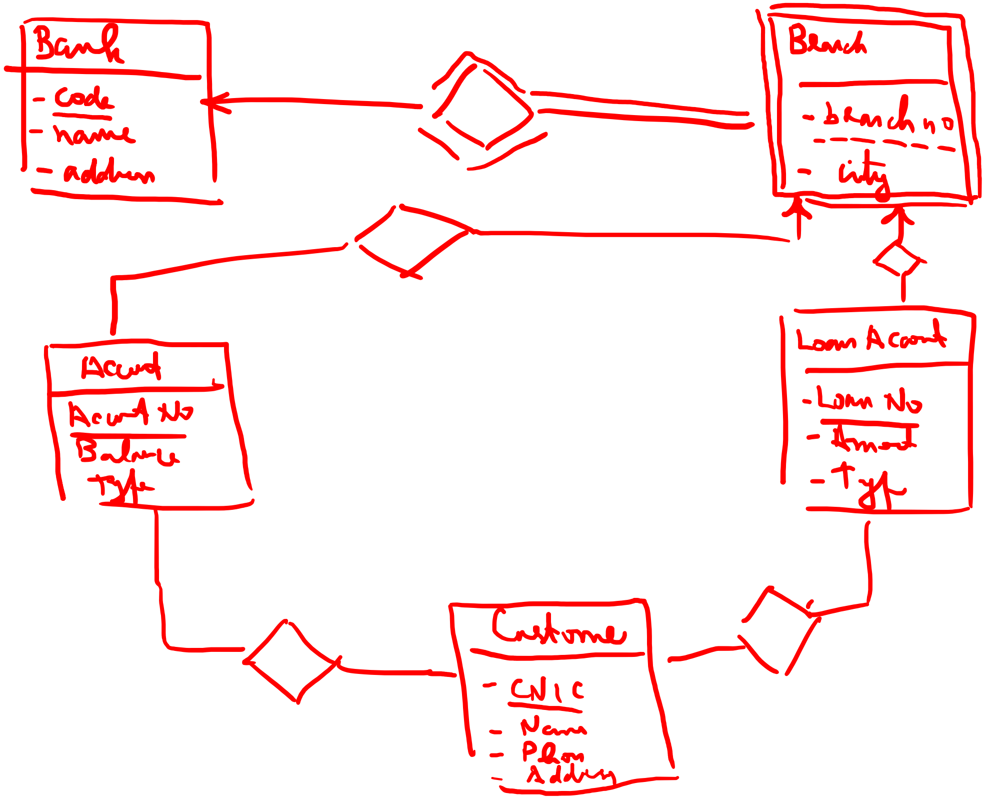
5) Customer

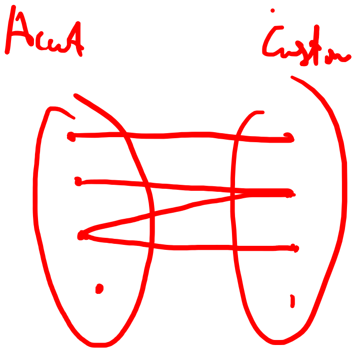
Entity Sets ??

You have been assigned the task of designing a database for a country's banking system. The following information is available:

1. A bank has a code, name, and address. The bank's code is unique and is used for identifying the bank.
2. A bank may have several branches. Each branch has a branch_number and city.
3. A branch can have accounts. Each account has an Acct_no, balance, and type. An account can be of two types: ~~current~~ and savings.
4. Each branch also has accounts for loans. Each loan has a loan_no, amount, and type. A loan can be of two types: ~~fixed~~ or recurring.
5. Each account must be linked to a customer. A customer has a CNIC_number, Name, Phone, and Address. A customer can have several kinds of bank accounts in the bank and can avail of loans within each branch.





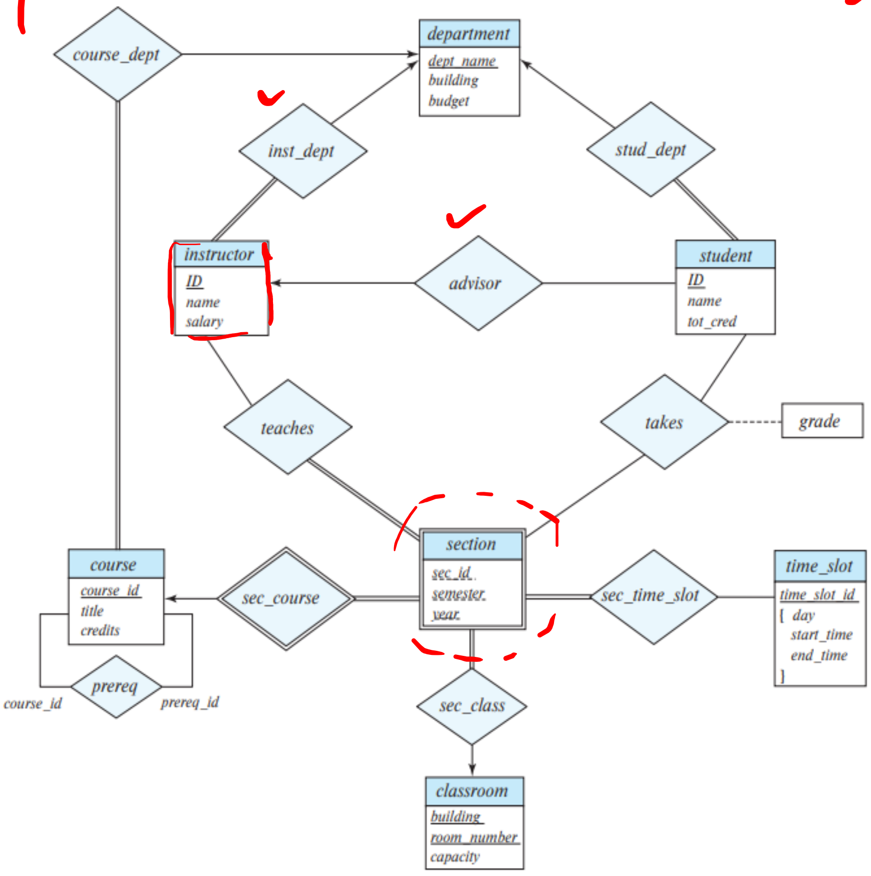


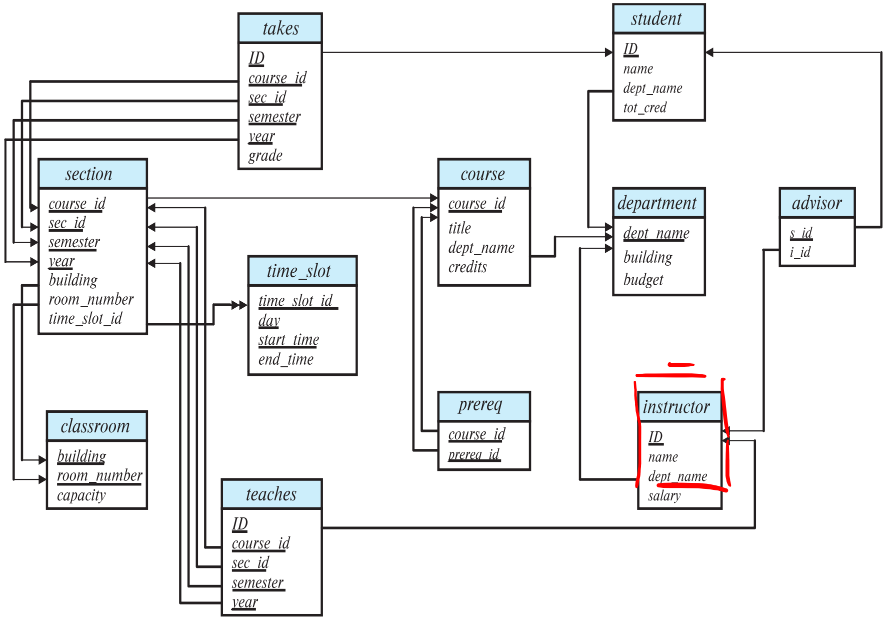
Scenario ERD

Entity-Relationship Diagram vs Relational Schema Diagram

1) Translated in 2)

TRANSLATION 3)





Relational Schema

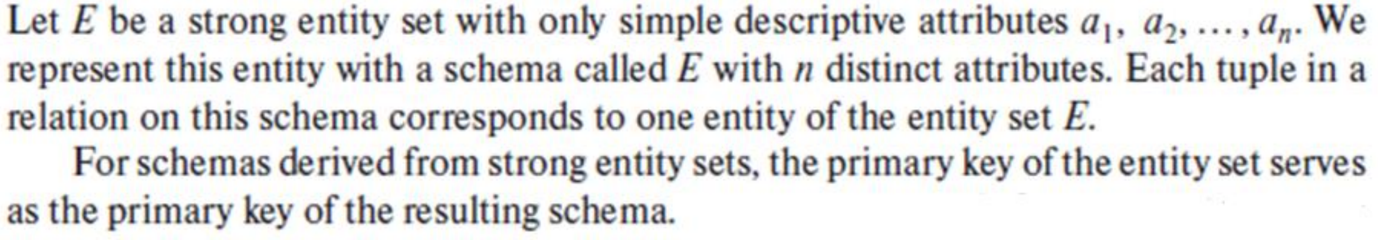
Figure 6.15 ~~E-R diagram~~ for a university enterprise.

Converting E-R Diagrams to Relational Schemas: Some Rules

- For each entity set and for each relationship set, a unique relation schema is created to which we assign the name of the corresponding entity set or relationship set.

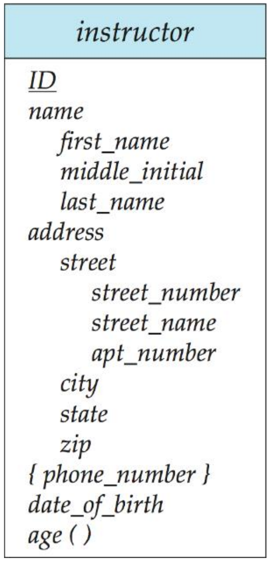
Converting E-R Diagrams to Relational Schemas: Some Rules

- For each entity set and for each relationship set, a unique relation schema is created to which we assign the name of the corresponding entity set or relationship set.
- Handling Strong Entity Sets with Simple Attributes:



Let E be a strong entity set with only simple descriptive attributes $a_1, a_2, \ldots, a_n$. We represent this entity with a schema called E with n distinct attributes. Each tuple in a relation on this schema corresponds to one entity of the entity set E.

For schemas derived from strong entity sets, the primary key of the entity set serves as the primary key of the resulting schema.



ID	Phonenumber
1	(0332 ... , 0313 ...)

ID	Ph1	Ph2	Ph3
1	033-...	null	null

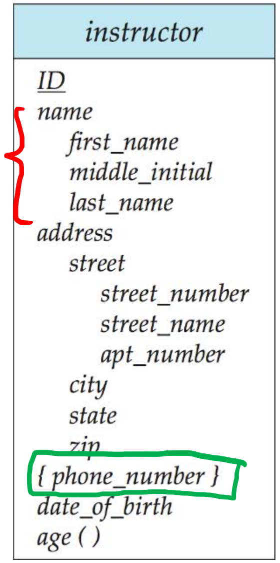
Converting E-R Diagrams to Relational Schemas: Some Rules

100

- For each entity set and for each relationship set, a unique relation schema is created to which we assign the name of the corresponding entity set or relationship set.

- Handling Strong Entity Sets with Composite Attributes:
  - We handle composite attributes by creating a separate attribute for each of the component attribute.
  - We do not create a separate attribute for the component attribute itself.

Entity Set


instructor

ID
name
first_name
middle_initial
last_name
address
street
street_number
street_name
apt_number
city
state
zip
{ phone_number }
date_of_birth
age ( )

instructor

ID	~~name~~

instructor
ID first_name middle_initial last_name phone_number1

phone_number2
phone_number3

instructor

ID	
1	
2	
3	

Phonenumbers

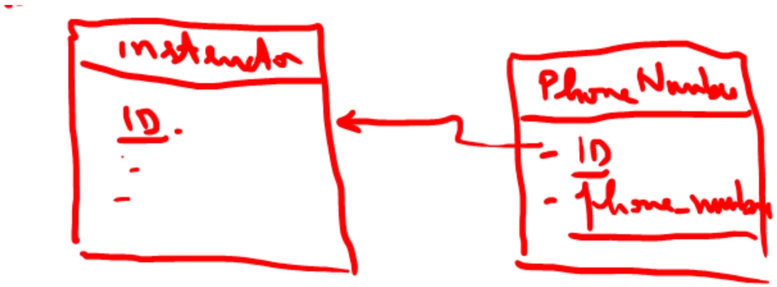
ID	Phone-number
1	0333
2	0310
2	0313

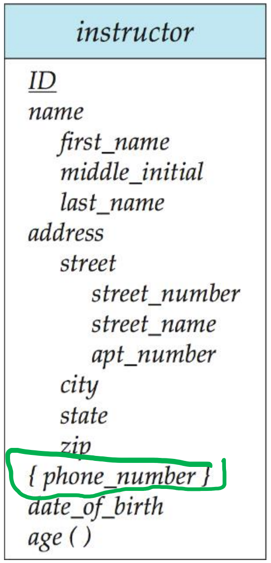
Converting E-R Diagrams to Relational Schemas: Some Rules

- For each entity set and for each relationship set, a unique relation schema is created to which we assign the name of the corresponding entity set or relationship set.

- Handling Strong Entity Sets with Multi-valued Attributes:

For a multivalued attribute M, we create a relation schema R with an attribute A that corresponds to M and attributes corresponding to the primary key of the entity set or relationship set of which M is an attribute.

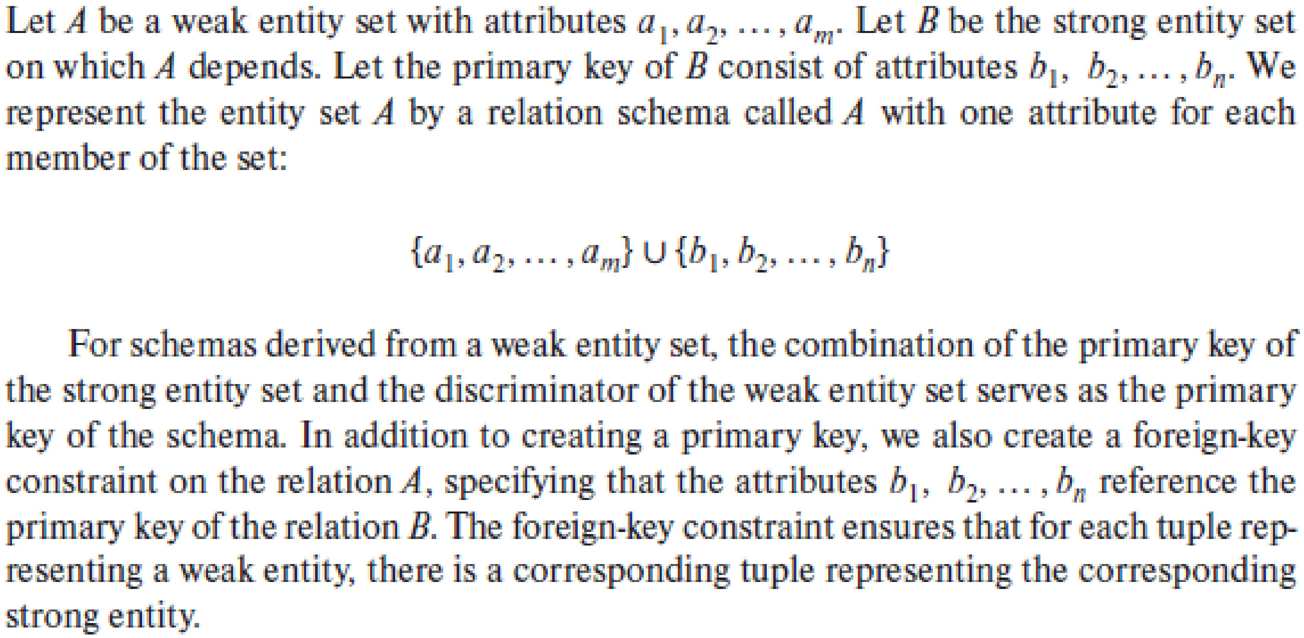




Converting E-R Diagrams to Relational Schemas: Some Rules

- For each entity set and for each relationship set, a unique relation schema is created to which we assign the name of the corresponding entity set or relationship set.

- Handling Weak Entity Sets:



Let A be a weak entity set with attributes $a_1, a_2, \ldots, a_m$. Let B be the strong entity set on which A depends. Let the primary key of B consist of attributes $b_1,\ b_2, \ldots, b_n$. We represent the entity set A by a relation schema called A with one attribute for each member of the set:

$$\{a_1, a_2, \ldots, a_m\} \cup \{b_1, b_2, \ldots, b_n\}$$

For schemas derived from a weak entity set, the combination of the primary key of the strong entity set and the discriminator of the weak entity set serves as the primary key of the schema. In addition to creating a primary key, we also create a foreign-key constraint on the relation A, specifying that the attributes $b_1,\ b_2, \ldots, b_n$ reference the primary key of the relation B. The foreign-key constraint ensures that for each tuple representing a weak entity, there is a corresponding tuple representing the corresponding strong entity.

Converting E-R Diagrams to Relational Schemas: Some Rules

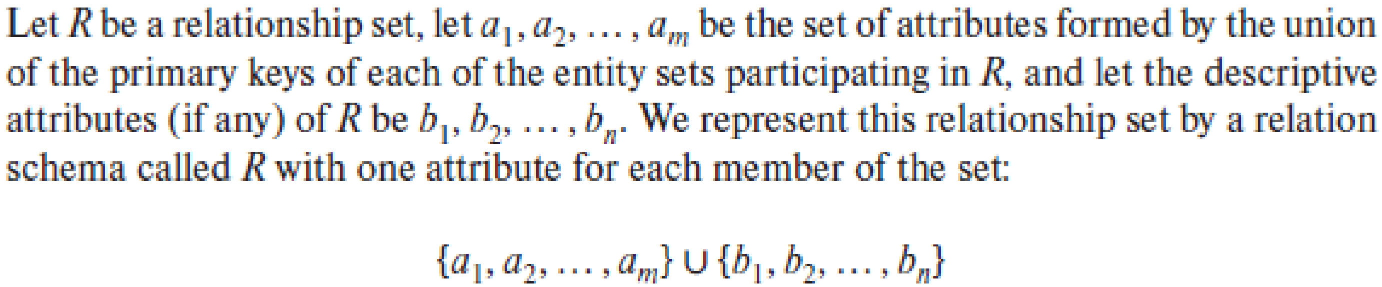
- For each entity set and for each relationship set, a unique relation schema is created to which we assign the name of the corresponding entity set or relationship set.

- Handling Weak Entity Sets:

Converting E-R Diagrams to Relational Schemas: Some Rules

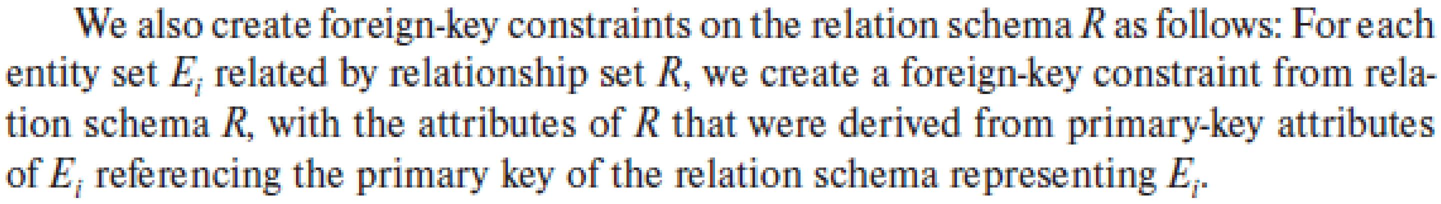
- For each entity set and for each relationship set, a unique relation schema is created to which we assign the name of the corresponding entity set or relationship set.

- Handling Relationship Sets:



Let R be a relationship set, let $a_1, a_2, \ldots, a_m$ be the set of attributes formed by the union of the primary keys of each of the entity sets participating in R, and let the descriptive attributes (if any) of R be $b_1, b_2, \ldots, b_n$. We represent this relationship set by a relation schema called R with one attribute for each member of the set:

$$\{a_1, a_2, \ldots, a_m\} \cup \{b_1, b_2, \ldots, b_n\}$$



We also create foreign-key constraints on the relation schema R as follows: For each entity set E_i related by relationship set R, we create a foreign-key constraint from relation schema R, with the attributes of R that were derived from primary-key attributes of E_i referencing the primary key of the relation schema representing E_i.

Converting E-R Diagrams to Relational Schemas: Some Rules

- For each entity set and for each relationship set, a unique relation schema is created to which we assign the name of the corresponding entity set or relationship set.

- Handling Relationship Sets:

Converting E-R Diagrams to Relational Schemas: Some Rules

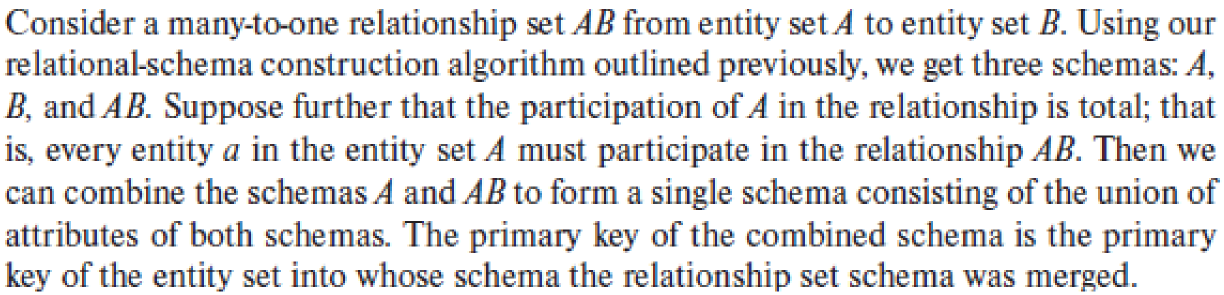
- After creating a unique relation schema for each entity set and each relationship set, a few rules can be applied to merge some relations into each other

- One-to-One Relationships:
  - The relation schema for the relationship set can be combined with the schemas for either of the entity sets.

Converting E-R Diagrams to Relational Schemas: Some Rules

- After creating a unique relation schema for each entity set and each relationship set, a few rules can be applied to merge some relations into each other

- Many-to-One Relationships:



Consider a many-to-one relationship set *AB* from entity set *A* to entity set *B*. Using our relational-schema construction algorithm outlined previously, we get three schemas: *A*, *B*, and *AB*. Suppose further that the participation of *A* in the relationship is total; that is, every entity *a* in the entity set *A* must participate in the relationship *AB*. Then we can combine the schemas *A* and *AB* to form a single schema consisting of the union of attributes of both schemas. The primary key of the combined schema is the primary key of the entity set into whose schema the relationship set schema was merged.

Converting E-R Diagrams to Relational Schemas: Some Rules

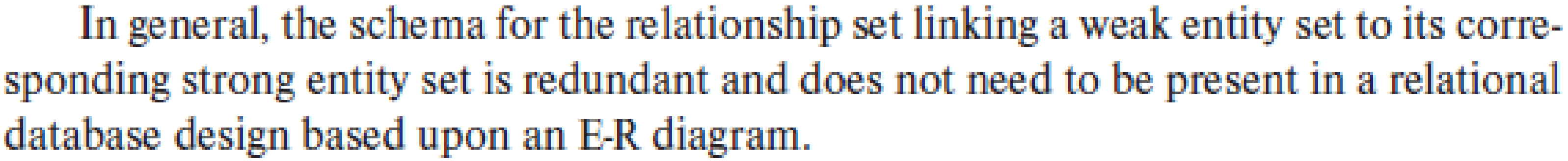
- After creating a unique relation schema for each entity set and each relationship set, a few rules can be applied to merge some relations into each other

- Many-to-Many Relationships:
  - The relation schemas corresponding to many-to-many relationship sets are not merged.

Converting E-R Diagrams to Relational Schemas: Some Rules

- After creating a unique relation schema for each entity set and each relationship set, a few rules can be applied to merge some relations into each other

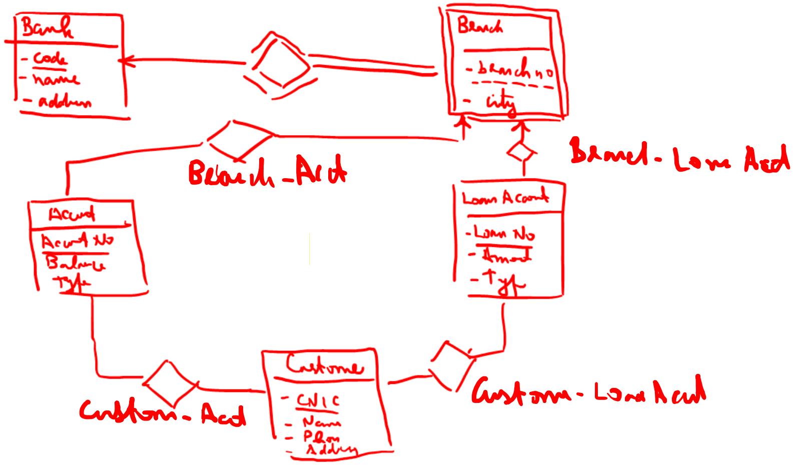
- Relationship Sets Linking Weak Entity Set to Its Corresponding Strong Entity Set:



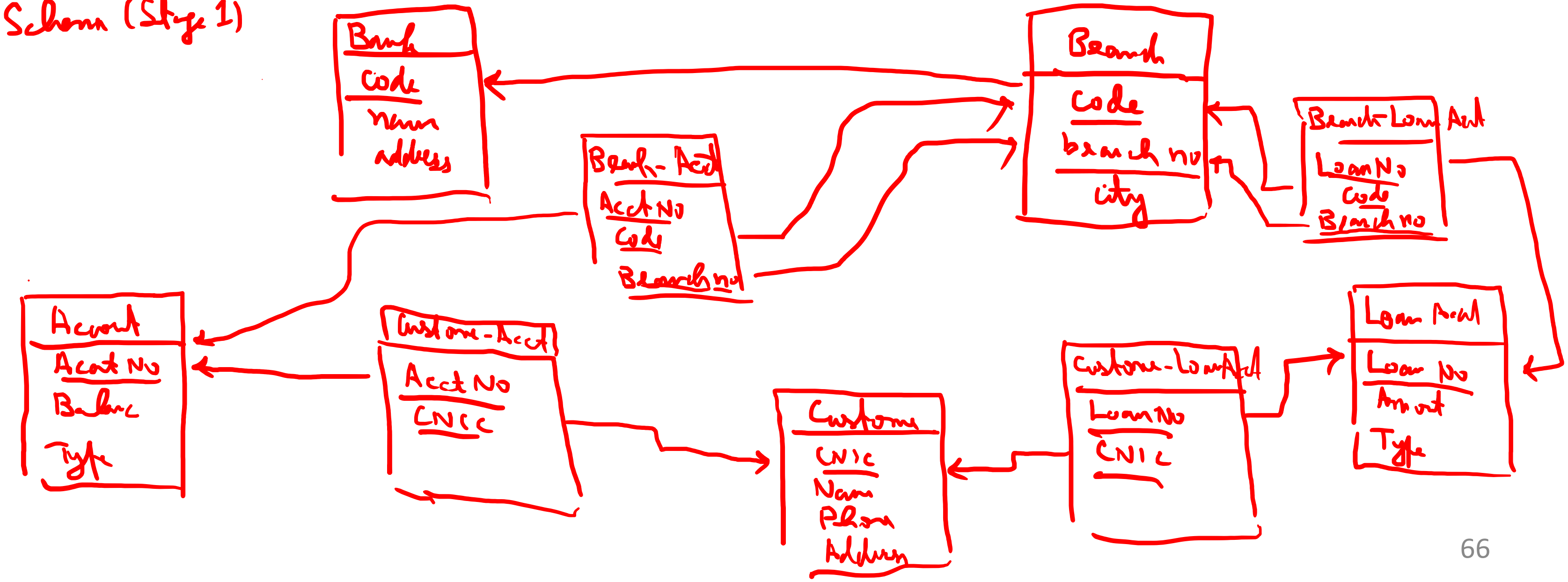
In general, the schema for the relationship set linking a weak entity set to its corresponding strong entity set is redundant and does not need to be present in a relational database design based upon an E-R diagram.

Converting E-R Diagrams to Relational Schemas: Example

ERD

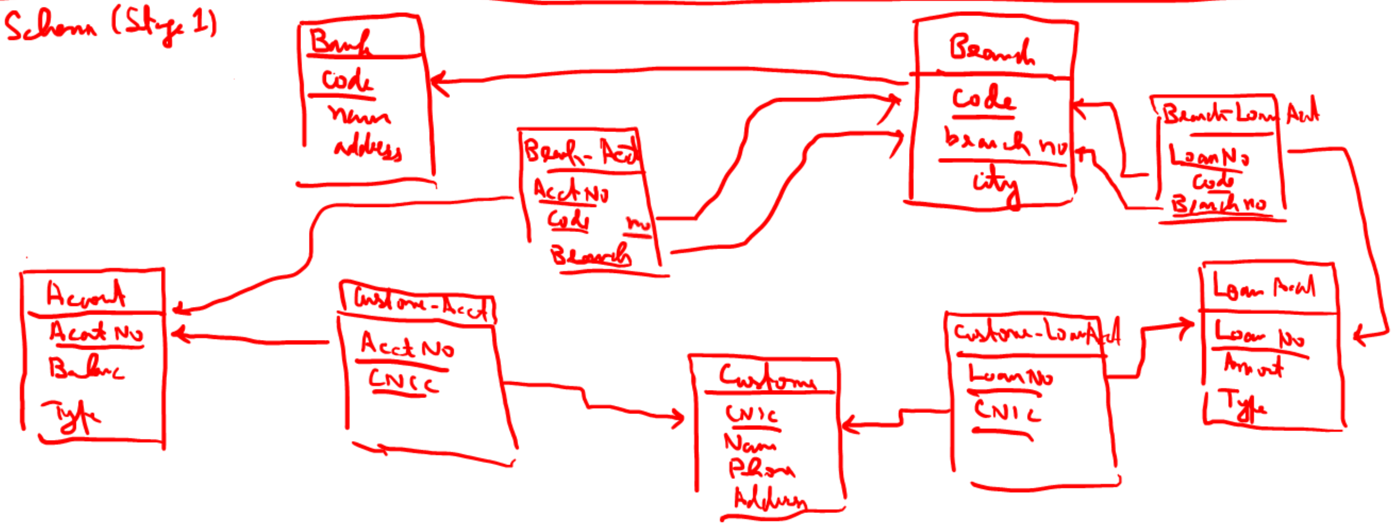


Relational Schema (Stage 1)

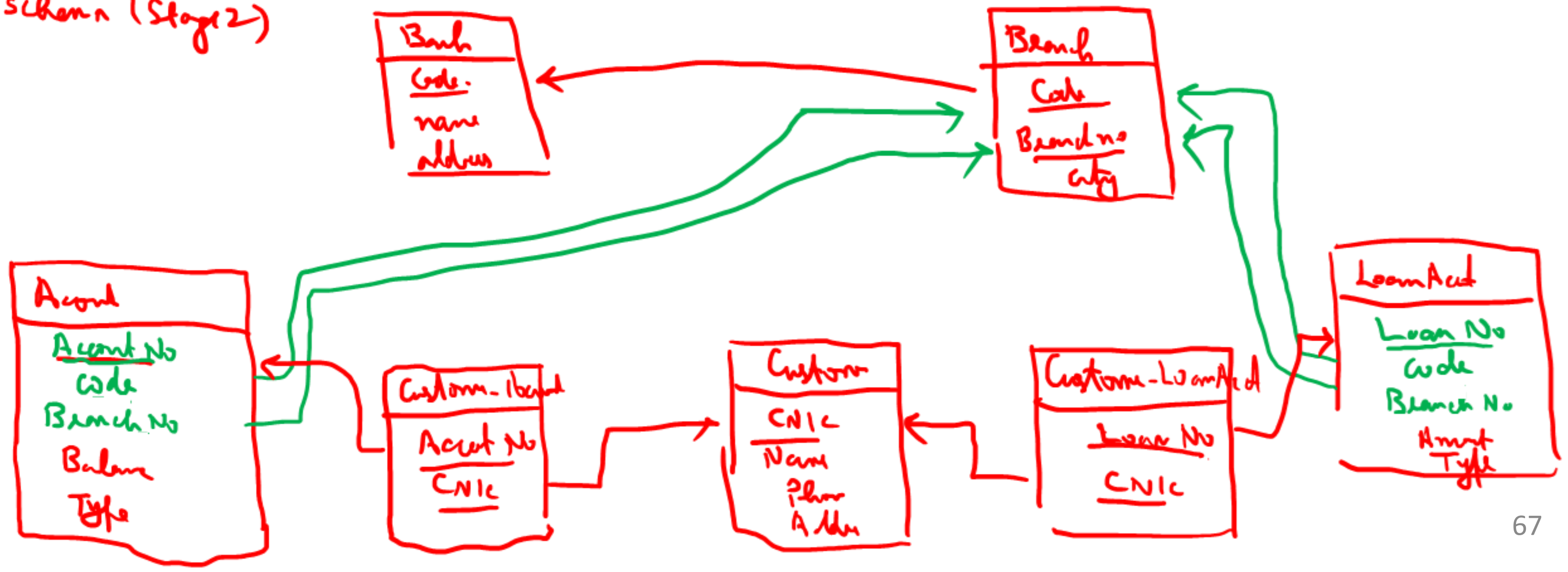


Converting E-R Diagrams to Relational Schemas: Example

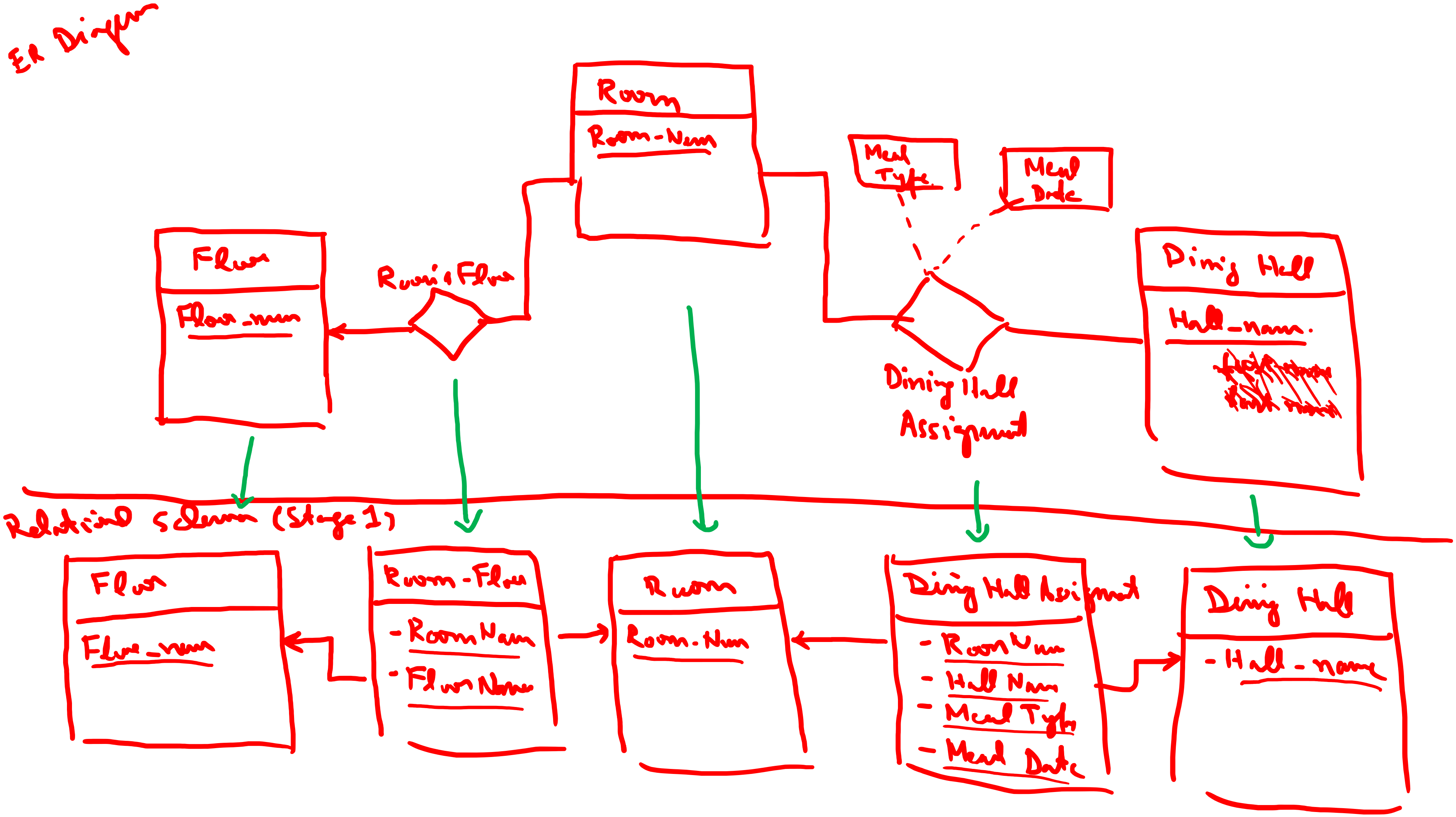
Relational Schema (Stage 1)



Relational schema (Stage 2)

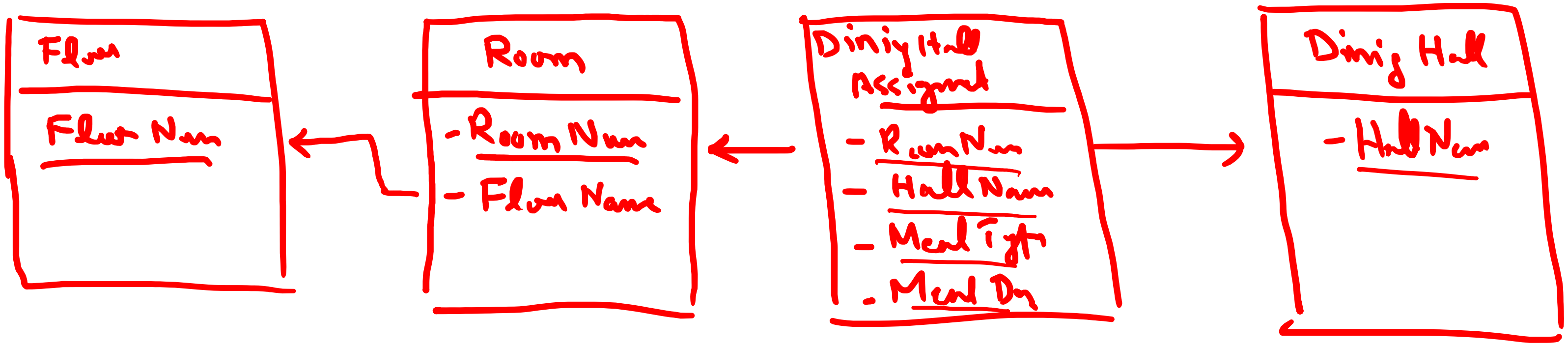


Converting E-R Diagrams to Relational Schemas: Example



Converting E-R Diagrams to Relational Schemas: Example

Relational Schema (Stage 2)



Floor
Floor Num
Room
- Room Num
- Floor Num
Dining Hall Assignment
- Room Num
- Hall Name
- Meal Type
- Meal Day
Dining Hall
- Hall Name