Chapter 3
Data modelling and databases

Outline

- Data model
- Data modeling
- E/R model
- Ralational data model
- Data modeling process

Data model

- A collection of conceptual tools for describing data, data relationships, data semantics and consistency constraint.
  - A conceptual representation of data structures
  - Visually represents the nature of data, business rules governing the data, and how it will be organized in the database.

Data modeling

- Involving professional data modelers working closely with business stakeholders, as well as potential users of the information system.
  - Process of creating a data model for an information system by applying certain formal techniques.
  - Defining and analyze data requirements needed to support the business processes.
- **Objectives**
  - To make sure all data objects required by a database are completely and accurately represented
  - The blueprint for creating a physical implementation of a database (commonly)

Data modeling requirements

- What is the information/data domain that you are modeling?
- What are the queries that you want to do?
  - e.g., "Find out the most wanted products? How many sales today?"
- What software do you want (have) to use?
- How do you want to share the data?

Data Model Terms

- **Entity** – a class of real-world objects having common attributes (e.g., sites, variables, methods).
- **Attribute** – A characteristic or property of an entity (site name, latitude, longitude)
- **Relationship** – an association between two or more entities
- **Cardinality** – the number of entities on either end of a relationship (one-to-one, one-to-many, many-to-many, etc.)

Examples

- Consider:
  - What is the “entity”?
  - What are the “attributes” of the entity?

Examples

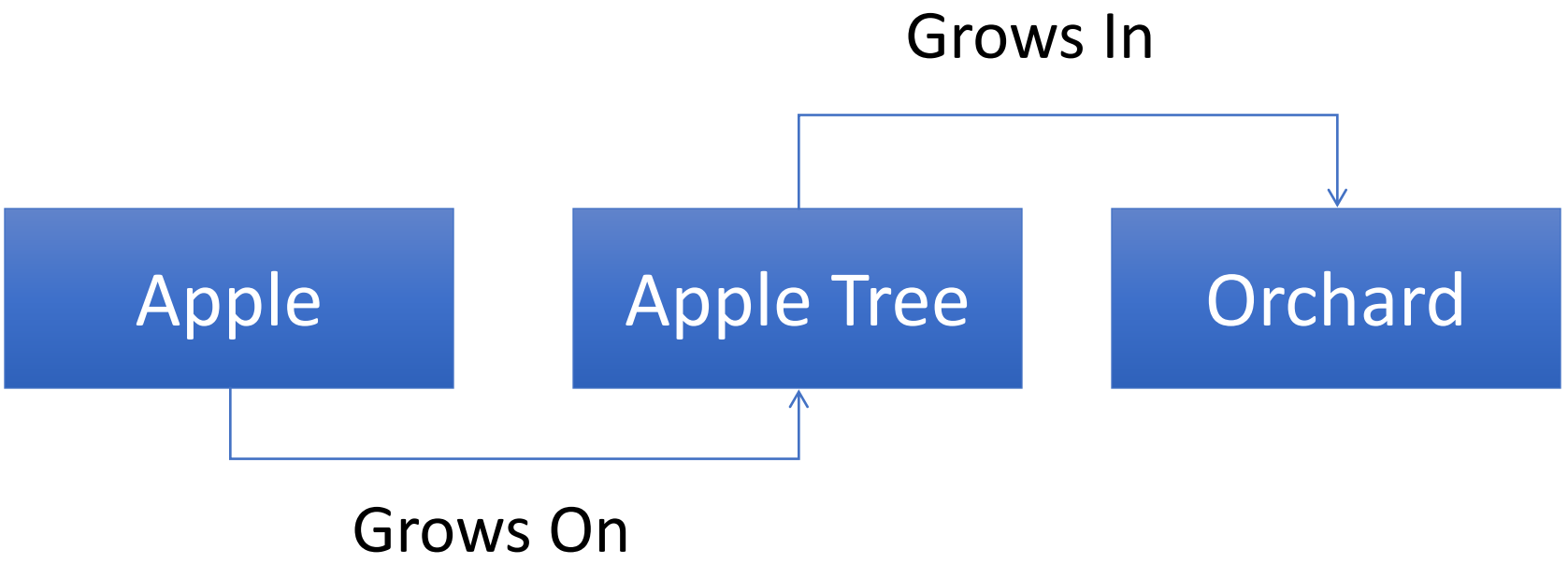
- What is the entity?
- What are the attributes?

Examples

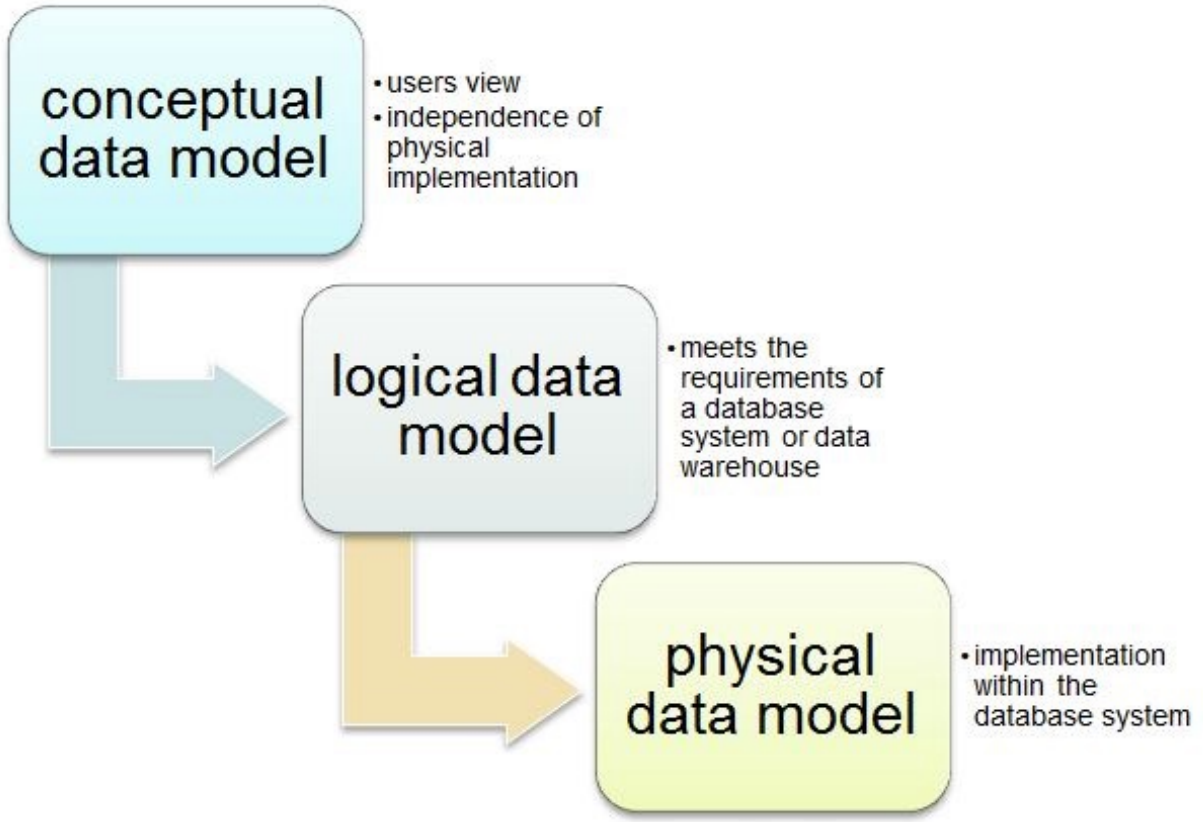
- What is the entity?
- What are the attributes?

Examples

- What are the relationships?



Different levels of data models



Different levels of data models

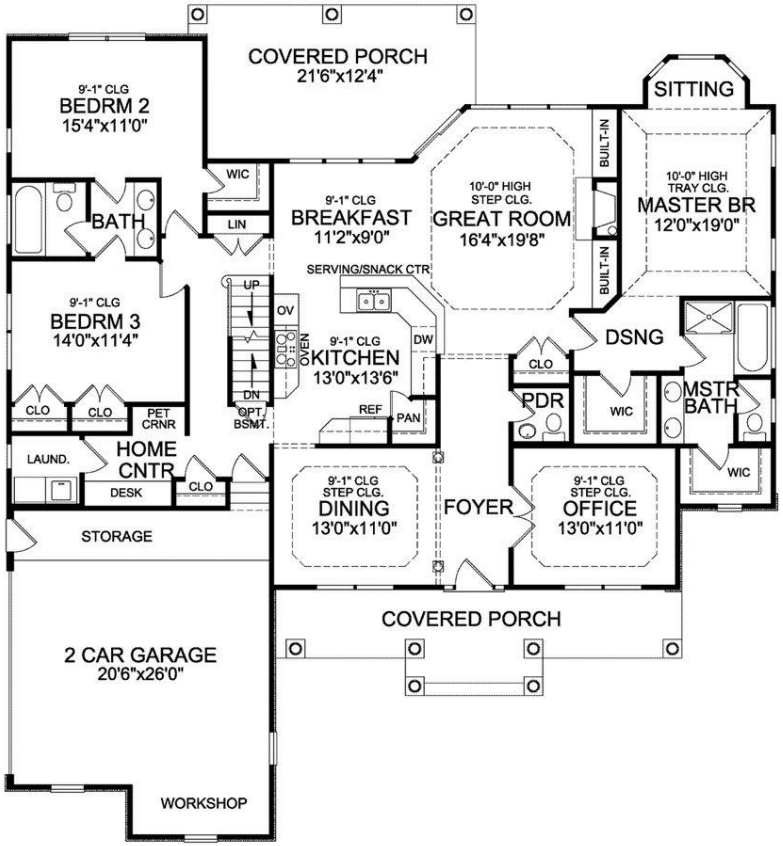
- **Conceptual:** describes WHAT the system contains
  - High-level description of the data domain
  - Does not constrain how that description is mapped to an actual implementation in software
- **Logical:** describes HOW the system will be implemented, regardless of the DBMS
  - Technology independent
  - Contains more detail than the Conceptual Data Model
- **Physical:** describes HOW the system will be implemented using a specific DBMS

Conceptual data models

- WHAT the system contains.

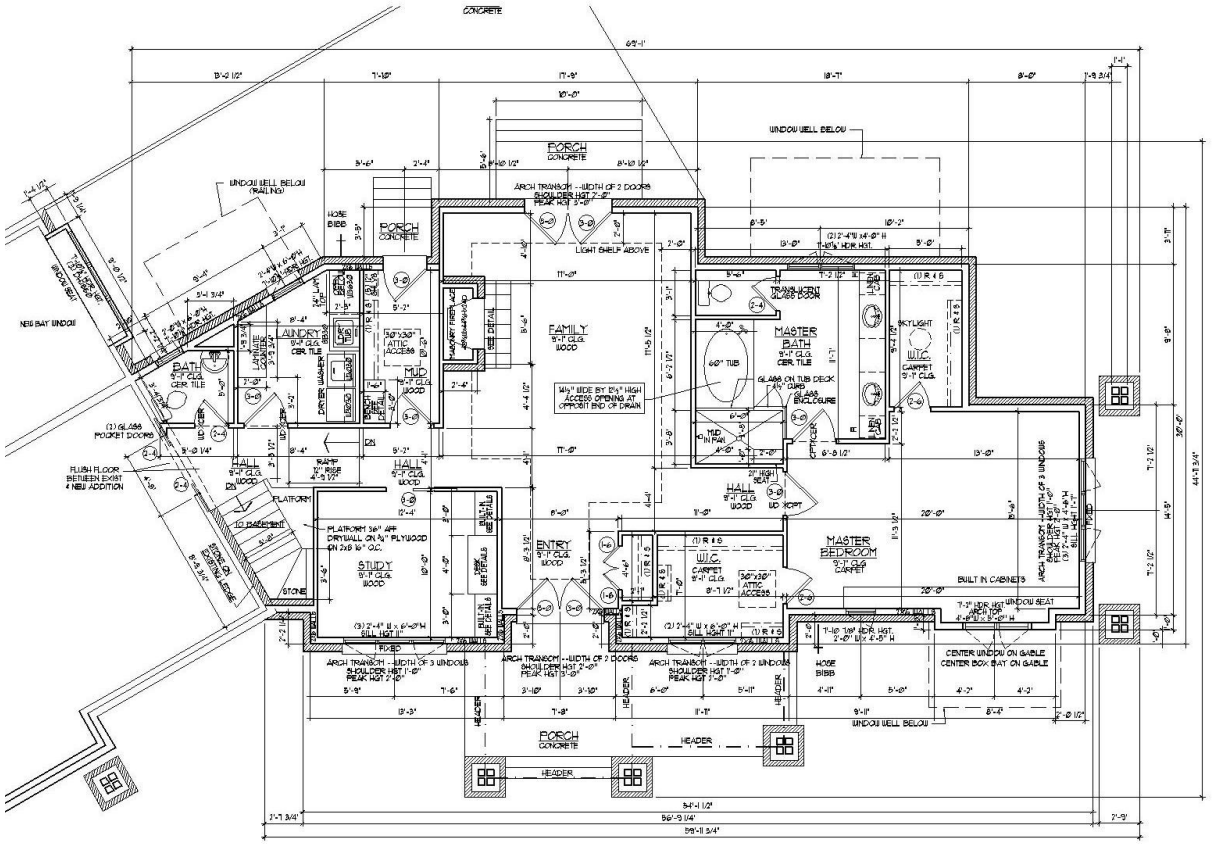
Logical data models

- HOW the system will be implemented, regardless of the DBMS



Physical data models

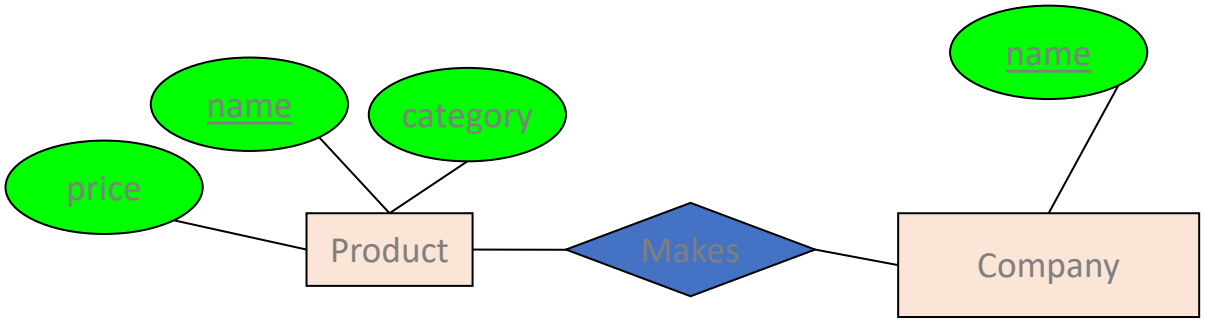
- HOW the system will be implemented using a specific DBMS



Entity – Relationship model

E/R data model

- E/R is a visual syntax for DB design which is precise enough for technical points, but abstracted enough for non-technical people

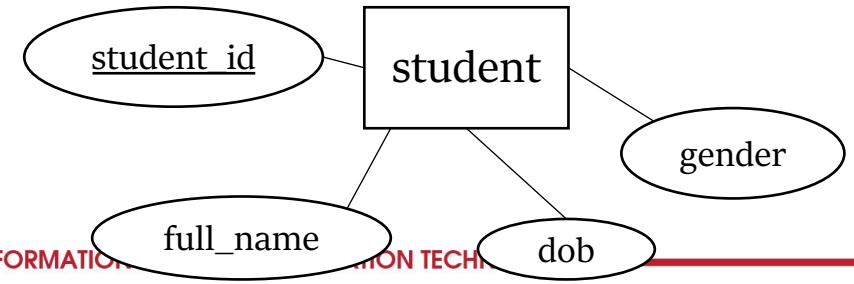


Entity and Entity sets

- Entity
  - is a thing in the real world with an independent existence.
  - An entity may be an object with a physical existence (a particular person, car, house, or employee) or it may be an object with a conceptual existence (a company, a job, or a university course).
- Entity sets
  - a collection of similar entities forms an entity set.
- In ERD, rectangular boxes represent for entity sets

Attributes

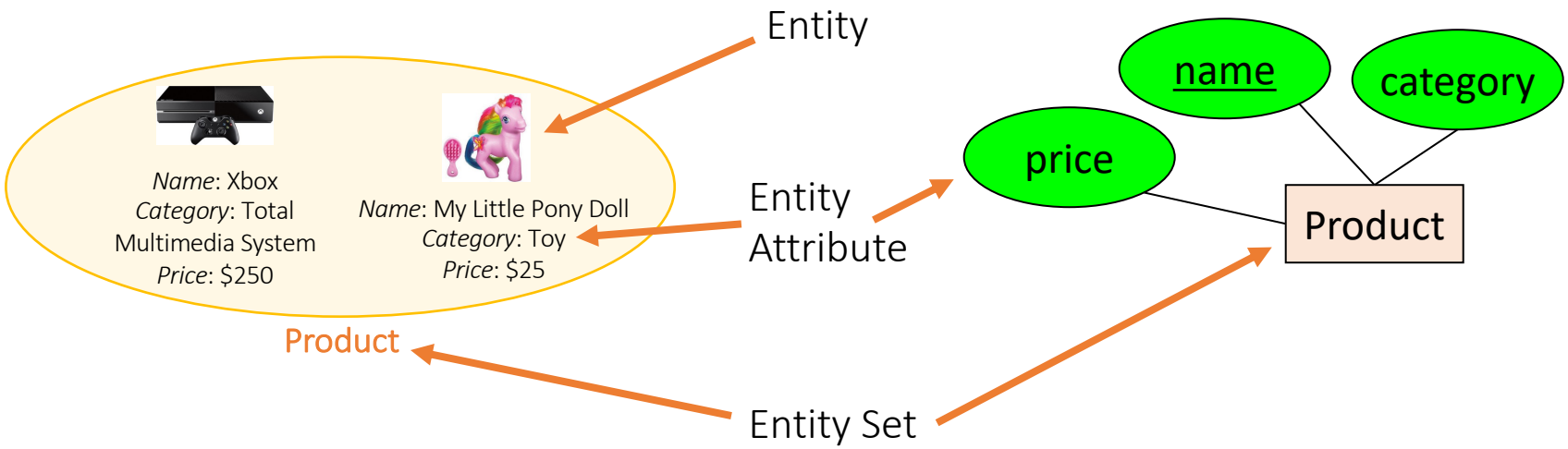
- Entity sets have associated attributes, which are properties of the entities in that set.
  - For instance, each entity "student" has some properties such as student_id, first_name, last_name, dob, gender, address, and so on.
  - In ERD, ovals represent for attributes
- Value domain of an attribute
  - Each simple attribute of an entity type is associated with a value set (or domain of values).
  - For example: domain(gender) = {male, female}; domain(dob) = {date}; domain(last_name) = {char(30)}.



Example: Entities, Entity sets, and attributes

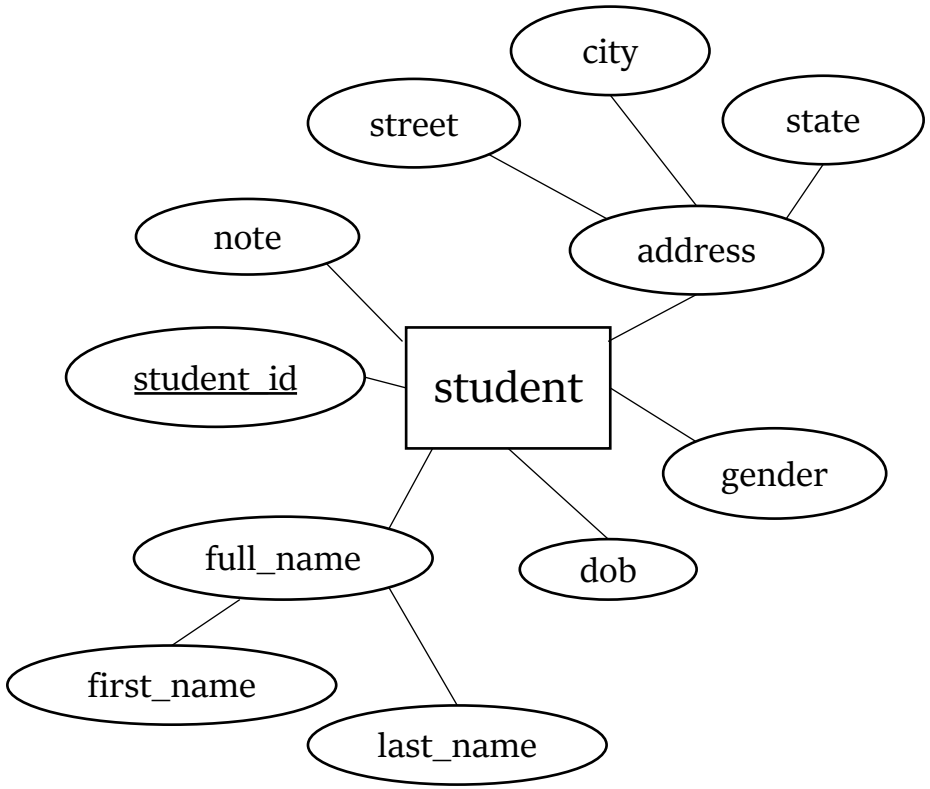
Example:

Entities are **not** explicitly represented in E/R diagrams!



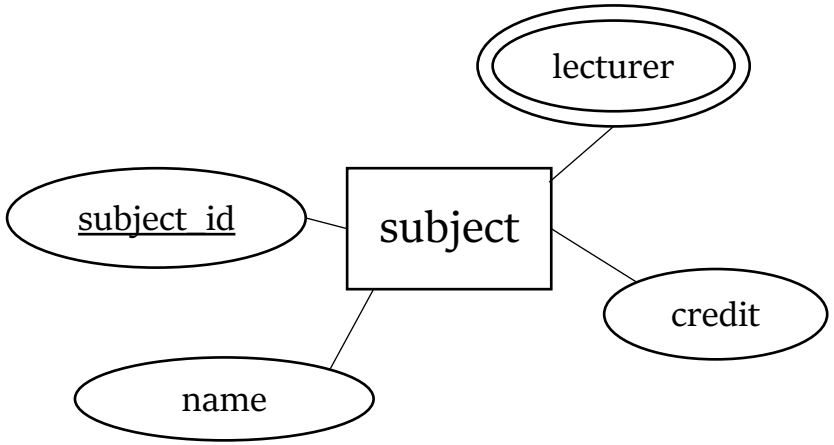
Attribute types

- Simple/atomic attributes: Attributes that are not divisible.
- Composite attributes: attributes can be divided into smaller subparts, which represent more basic attributes with independent meanings.



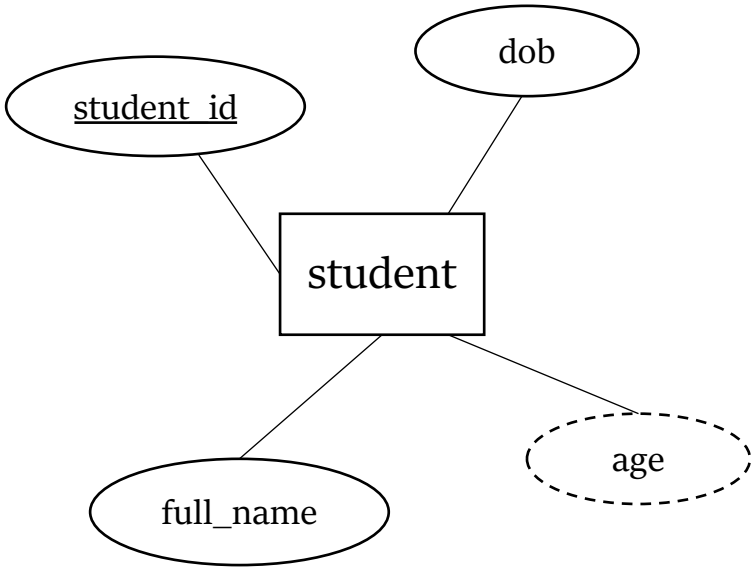
Attribute types

- Single-valued attributes: have a single value for a particular entity
- Multi-valued attributes: can have different numbers of values



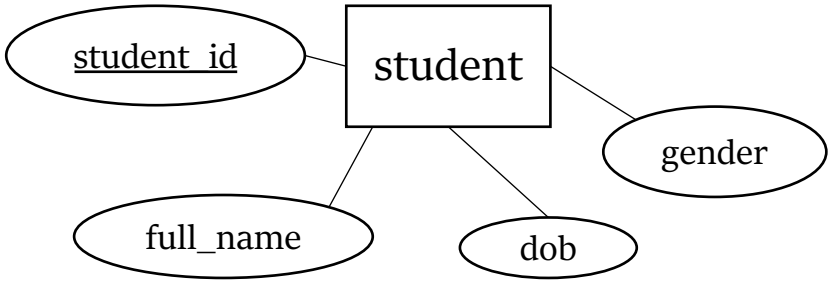
Attribute types

- Stored attributes vs. Derived attributes: age attribute is called a derived attribute and is said to be derivable from the dob attribute, which is called a stored attribute.



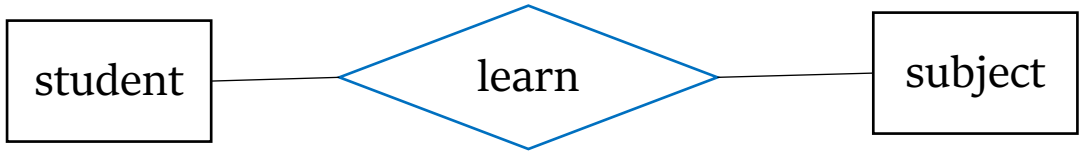
Keys

- A *key* is a **minimal** set of attributes that uniquely identifies an entity.
  - One or more attributes whose values are distinct for each individual entity in the entity set
- Each entity can have some keys (candidate keys). We choose one of them to be primary key.
- In ER diagrammatic notation, each key attribute has its name underlined inside the oval.

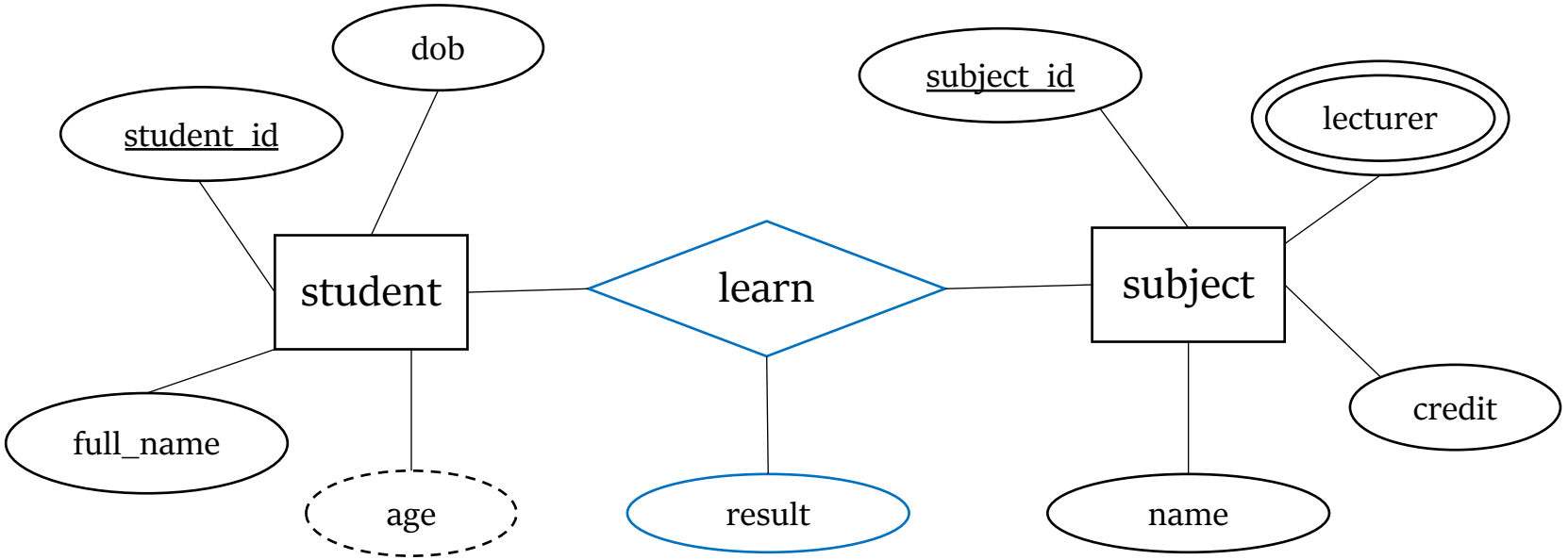


Relationships

- Relationships are connections among two or more entity sets.
- In ER diagrams, relationship types are displayed as diamond-shaped boxes
  - which are connected by straight lines to the rectangular boxes representing the participating entity types.
  - The relationship name is displayed in the diamond-shaped box.

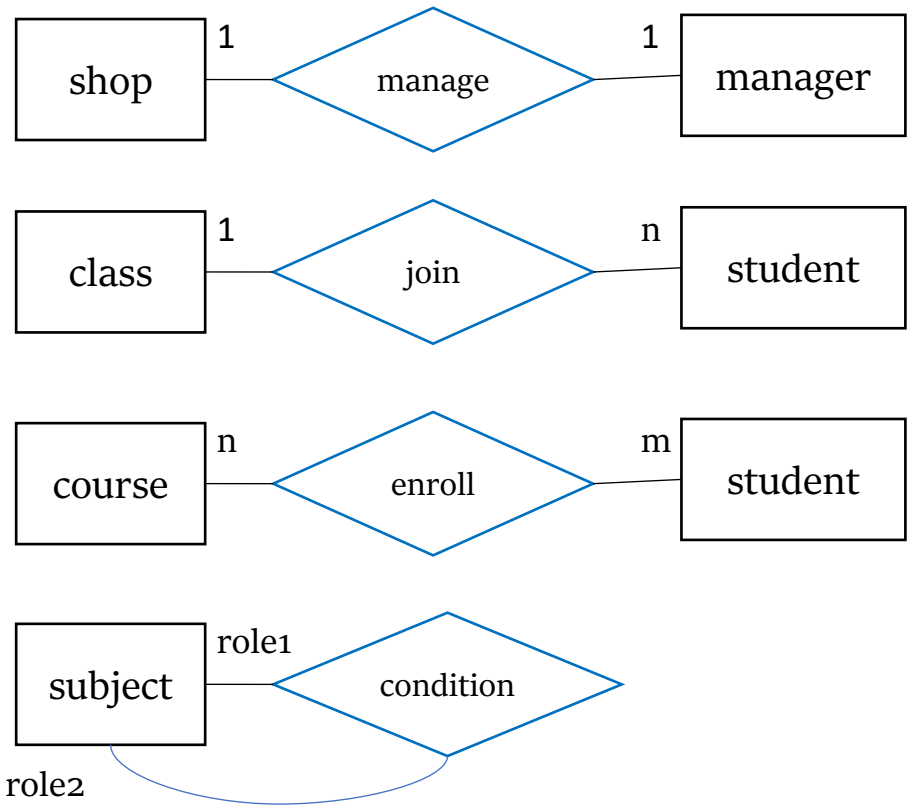


Relationship attributes



Relationship types

- 1 – 1
- 1 – n
- n – m
- recursive



How to create an ERD

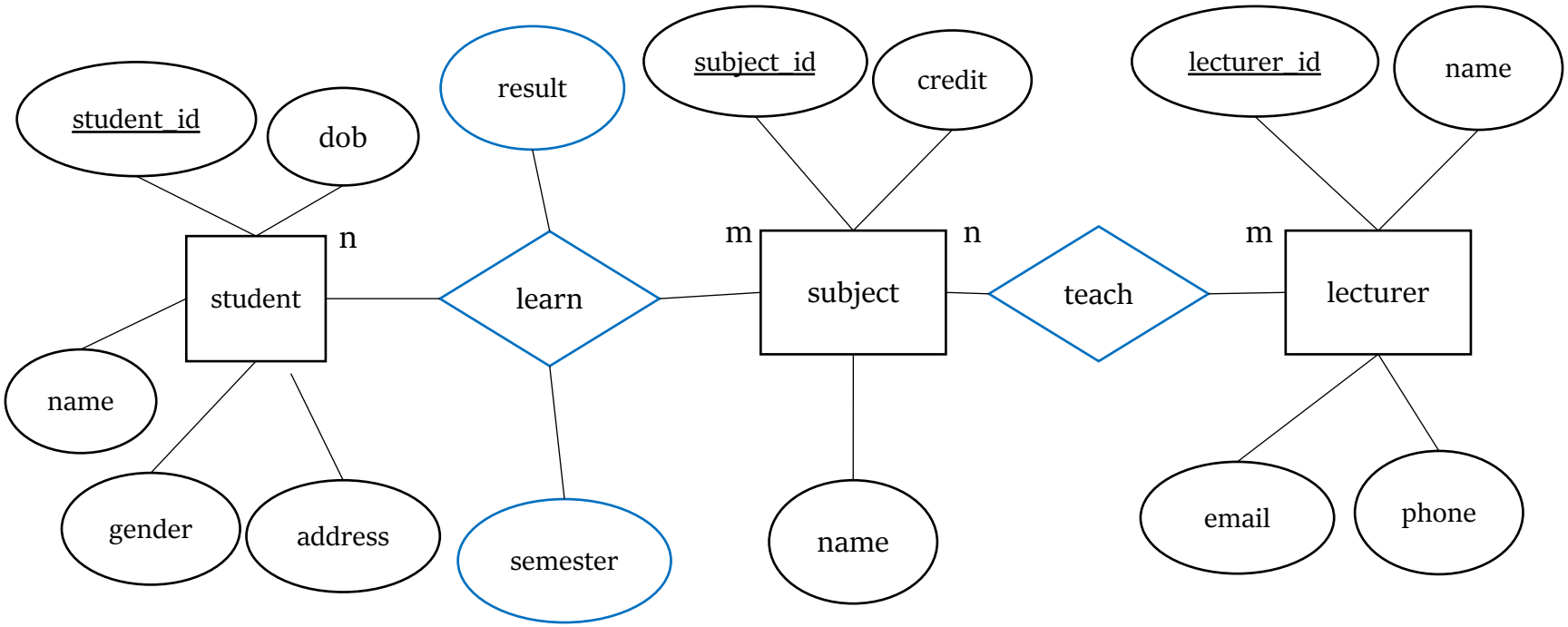
- Step 1: Identify all entity sets
  - Notice concepts, nouns
- Step 2: Identify all relationships among entity sets
  - Notice verbs
  - Type and degree of relationships

An example

- Read carefully the following scenario:
  - The information about students includes student identification (uniquely identify each student), name, gender, date of birth and address.
  - During the education time at school, students must study a lot of subjects. A subject can be learnt by students. A subject should be contained information such as subject identification, name and credit.
  - A lecturer can teach some subjects, and a subject can be taught by a group of lecturers. The information about lecturers should include lecturer identification, name, phone, email.
  - Students learn subjects at some semester, and their results should be stored.

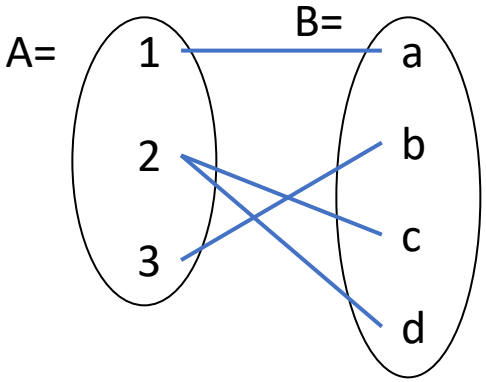
An example

- We can draw this ER diagram

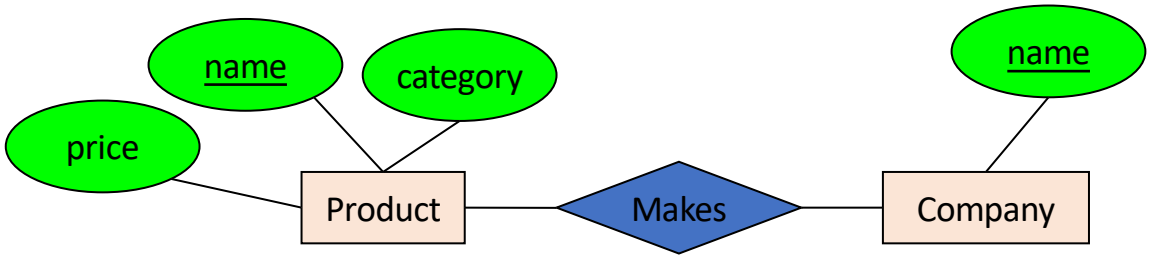


What is a Relationship?

- ***A mathematical definition:***
  - Let A, B be sets
    - *A={1,2,3}, B={a,b,c,d},*
  - A x B (the ***cross-product***) is the set of all pairs (a,b)
    - *A × B = {(1,a), (1,b), (1,c), (1,d), (2,a), (2,b), (2,c), (2,d), (3,a), (3,b), (3,c), (3,d)}*
  - **We define a relationship to be a subset of A x B**
    - *R = {(1,a), (2,c), (2,d), (3,b)}*



What is a Relationship?



A relationship between **entity sets P and C** is a *subset of all possible pairs of entities in P and C*, with tuples uniquely identified by *P and C's keys*

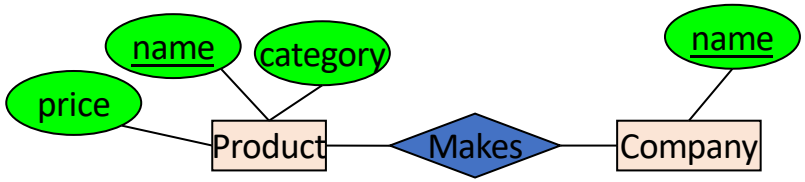
What is a Relationship?

Company

name
GizmoWorks
GadgetCorp

Product

name	category	price
Gizmo	Electronics	$9.99
GizmoLite	Electronics	$7.50
Gadget	Toys	$5.50



A relationship between **entity sets P and C** is a ***subset of all possible pairs of entities in P and C***, with tuples uniquely identified by ***P and C's keys***

What is a Relationship?

Company

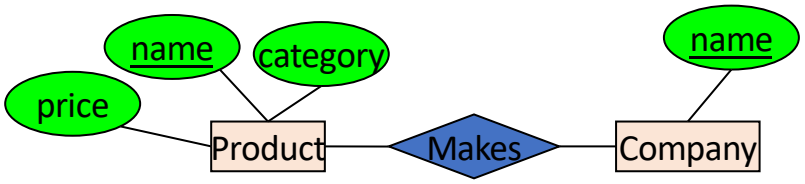
name
GizmoWorks
GadgetCorp

Product

name	category	price
Gizmo	Electronics	$9.99
GizmoLite	Electronics	$7.50
Gadget	Toys	$5.50

Company C × Product P

C.name	P.name	P.category	P.price
GizmoWorks	Gizmo	Electronics	$9.99
GizmoWorks	GizmoLite	Electronics	$7.50
GizmoWorks	Gadget	Toys	$5.50
GadgetCorp	Gizmo	Electronics	$9.99
GadgetCorp	GizmoLite	Electronics	$7.50
GadgetCorp	Gadget	Toys	$5.50



A relationship between **entity sets P and C** is a *subset of all possible pairs of entities in P and C*, with tuples uniquely identified by *P and C's keys*

What is a Relationship?

Company

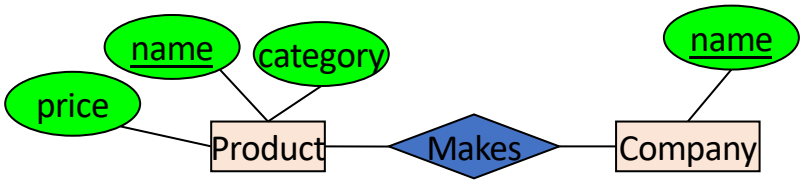
name
GizmoWorks
GadgetCorp

Product

name	category	price
Gizmo	Electronics	$9.99
GizmoLite	Electronics	$7.50
Gadget	Toys	$5.50

Company C × Product P

C.name	P.name	P.category	P.price
GizmoWorks	Gizmo	Electronics	$9.99
GizmoWorks	GizmoLite	Electronics	$7.50
GizmoWorks	Gadget	Toys	$5.50
GadgetCorp	Gizmo	Electronics	$9.99
GadgetCorp	GizmoLite	Electronics	$7.50
GadgetCorp	Gadget	Toys	$5.50



Makes

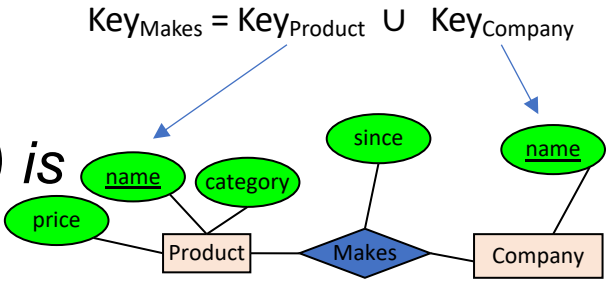
C.name	P.name
GizmoWorks	Gizmo
GizmoWorks	GizmoLite
GadgetCorp	Gadget

A relationship between **entity sets P and C** is a ***subset of all possible pairs of entities in P and C***, with tuples uniquely identified by ***P and C's keys***

What is a Relationship?

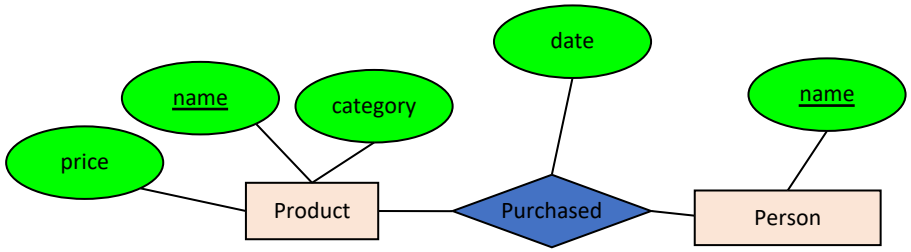
- There can only be **one relationship for every unique combination of entities**
- This also means that **the relationship is uniquely determined by the keys of its entities**
- *Example: the "key" for Makes (to right) is {Product.name, Company.name}*

This follows from our mathematical definition of a relationship- it's a SET!



Decision: Relationship vs. Entity?

- **Q:** What does this say?

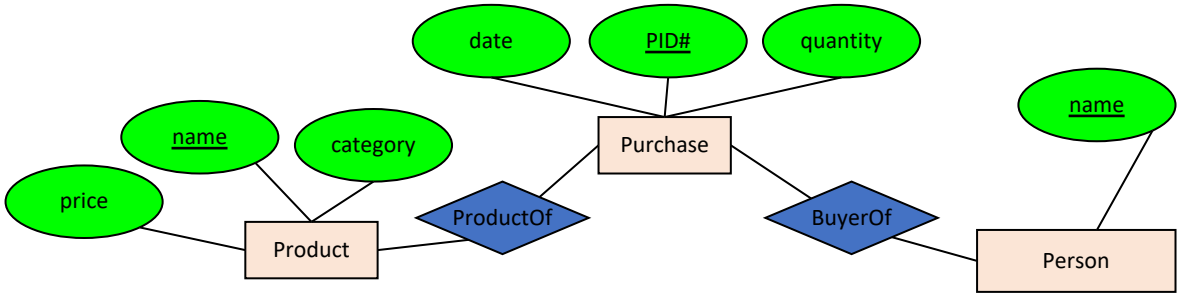


- **A:** A person can only buy a specific product once (on one date)

Modeling something as a relationship makes it unique; what if not appropriate?

Decision: Relationship vs. Entity?

- What about this way?



- *Now we can have multiple purchases per product, person pair!*

We can always use **a new entity** instead of a relationship. For example, to permit multiple instances of each entity combination!

Relational data model

Relational Data Model

- introduced by C F Codd in 1970
- A logical view of data
- Enables programmer to view data logically rather than physically
- Simple and uniform data model
- Based on a firm mathematical foundation (set theory)
- Main aspects:
  - Data structure: n-ary two dimensional
    - Attributes, domain value
    - Tuples
    - Tables/relations
  - Integrity constraints
    - entity integrity
    - referential integrity

Student

Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

Example: University Database

Student

Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

Takes

SID	SNO
1108	21
1108	23
1108	29
8507	23
8507	29

Enrol

SID	Course
3936	101
1108	113
8507	101

Course

No	Name	Dept
113	BCS	CSCE
101	MCS	CSCE

Subject

No	Name	Dept
21	Systems	CSCE
23	Database	CSCE
29	VB	CSCE
18	Algebra	Maths

Attribute

- Definition
  - designated by a meaningful name
  - also called field or column
  - denoted A
  - ✓Example
    - Id, name, suburb, dept, …

- Domain
  - define the original sets of data values used
  - denoted Domain(A)
  - √ Example
    - text, number, boolean, date/time, memo
    - Id: text(10)
    - Name: text(30)
    - birthday: date
    - …

Relation

- Definition
  - Defined as a set of attribute
  - also called table
  - Denoted R(A1,A2, … An)

    $R(A1,A2, \ldots An) \subseteq Dom(A1) \times \ldots \times Dom(An)$
  - ✓ Example
    - STUDENT(Id , Name, Suburb)
    - SUBJECT (No, Name, Dept)
- A relation is composed of tuples

Tuple

- Definition
  - defined as a set of attribute value
  - also called record or row
  - Denoted t(a1,a2, … an)
  - t(a1,a2, … an) ∈ Dom(A1) x … x Dom(An)
  - ✓ Example

(1108, Robert, Kew)

(3936, Glen, Bundoora)

- Each tuple must have a primary key which can uniquely identified the tuple.

Key

- Definition
  - A set of attribute in a relation
  - Used to identify each tuple
  - Given R(A1,A2, … An), $K \subseteq \{Ai\}$,
    K is key if $\forall t1, t2 \in R, \exists Ai \in K: t1.K \neq t2.K$
  - ✓Example
    - STUDENT(**Id** , Name, Suburb)
    - TAKE(**SID, SNO**)
- Remark:
  - If $K \subseteq \{Ai\}$ is key and $K \subseteq K' \subseteq \{Ai\}$
    → K' is super key
  - candidate key
    - a superkey without redundant attributes

Primary key

- Definition
  - a “smallest” key, i.e. with single attribute or smallest number of attributes allowing identify a unique tuple
  - A candidate key chosen to be the main key for the relation

- Entity constraint:
  - No attribute in the primary key can be NULL

✓Example
  - STUDENT(**Id** , Name, Suburb)
  - SUBJECT (**No**, Name, Dept)

Foreign key

- Specified between two relations and maintain the correspondence between tuples in these relations
  - also called referential integrity

- A set of attributes FK in a relation R1 is foreign key if
  - The attributes in FK correspond to the attributes in the primary key of another relation R2
  - The value for FK in each tuple of R1 either occur as values of primary key of a tuple in R2 or is entirely NULL

Example: University Database

Student

Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

Takes

SID	SNO
1108	21
1108	23
1108	29
8507	23
8507	29

Enrol

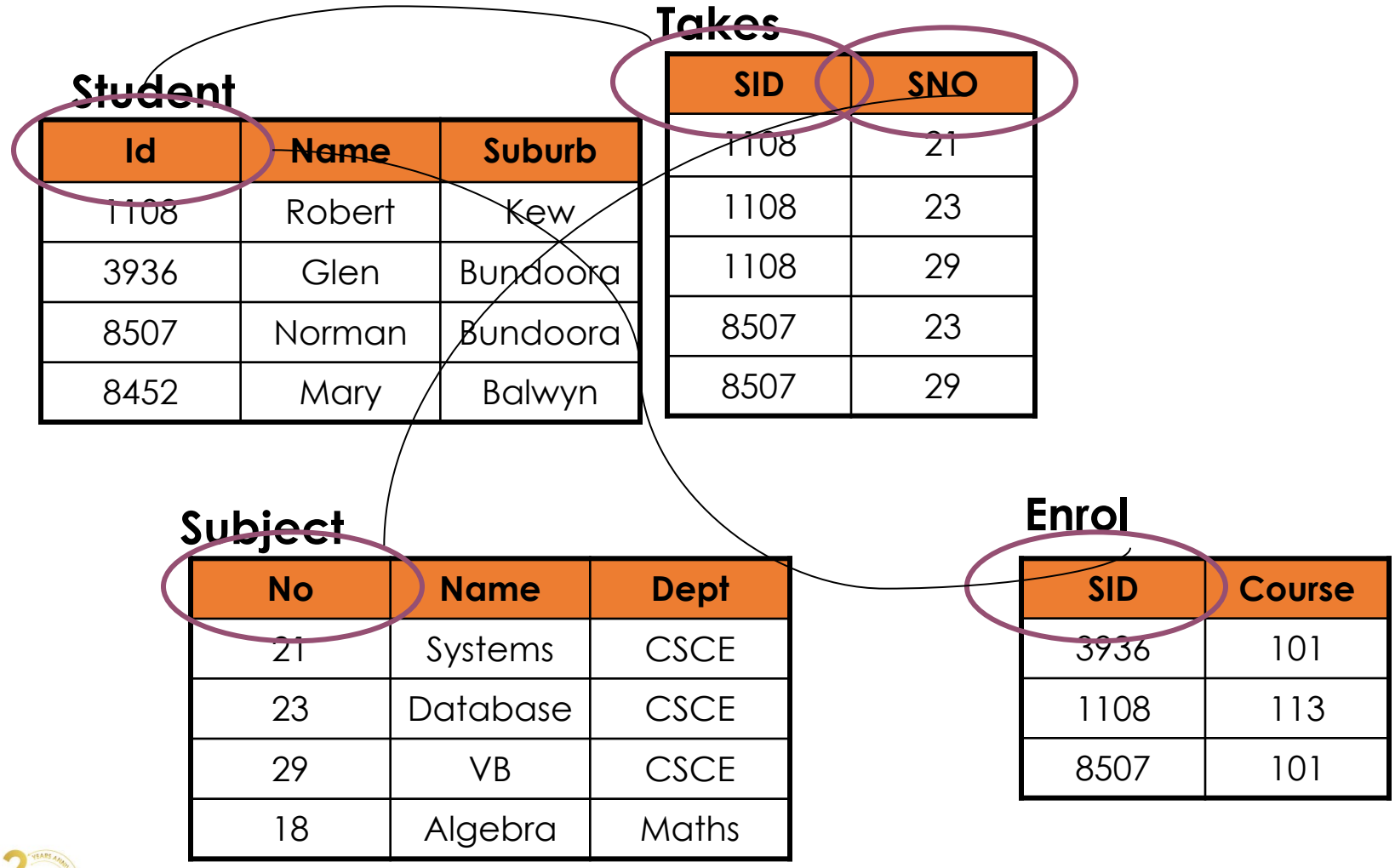
SID	Course
3936	101
1108	113
8507	101

Course

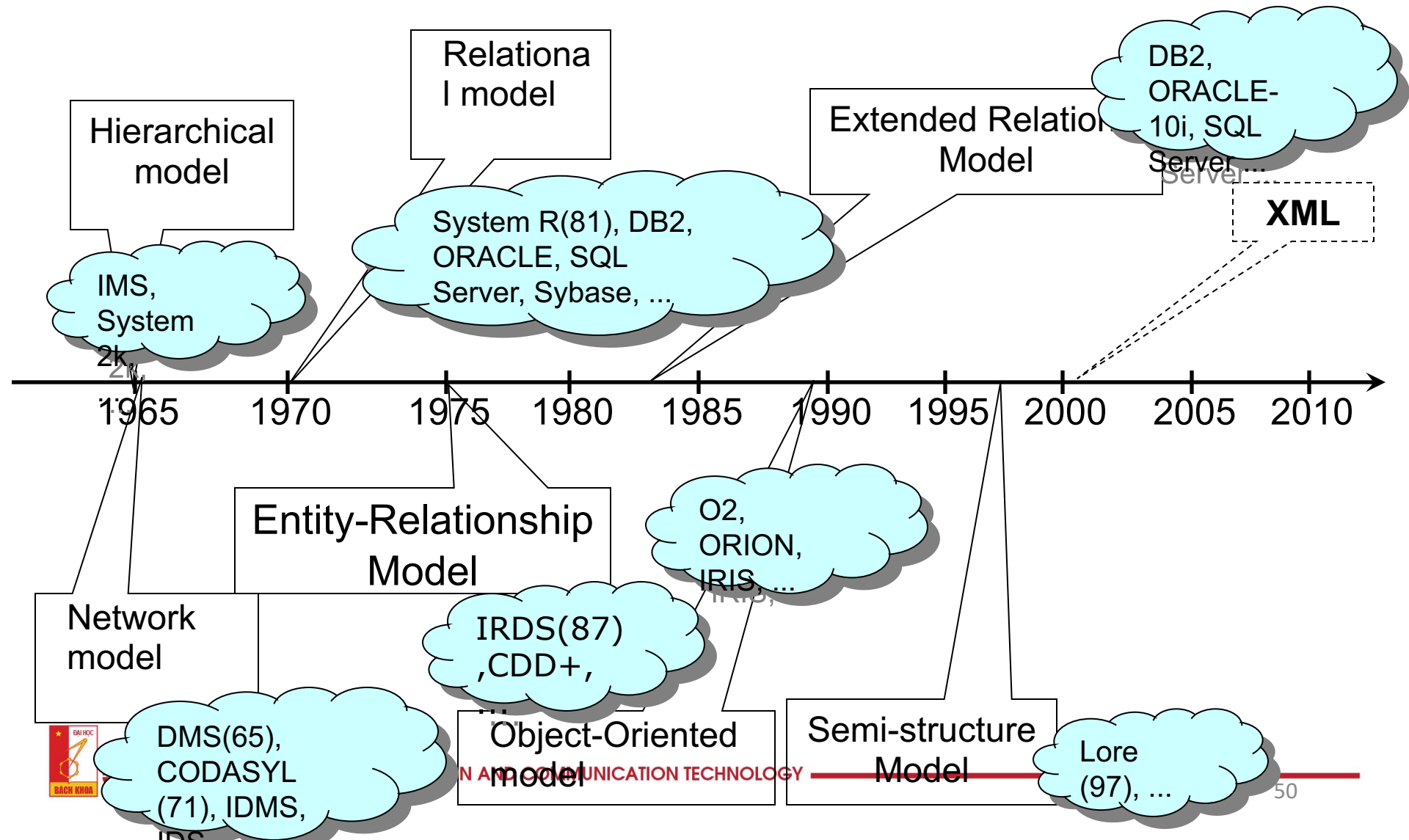
No	Name	Dept
113	BCS	CSCE
101	MCS	CSCE

Subject

No	Name	Dept
21	Systems	CSCE
23	Database	CSCE
29	VB	CSCE
18	Algebra	Maths

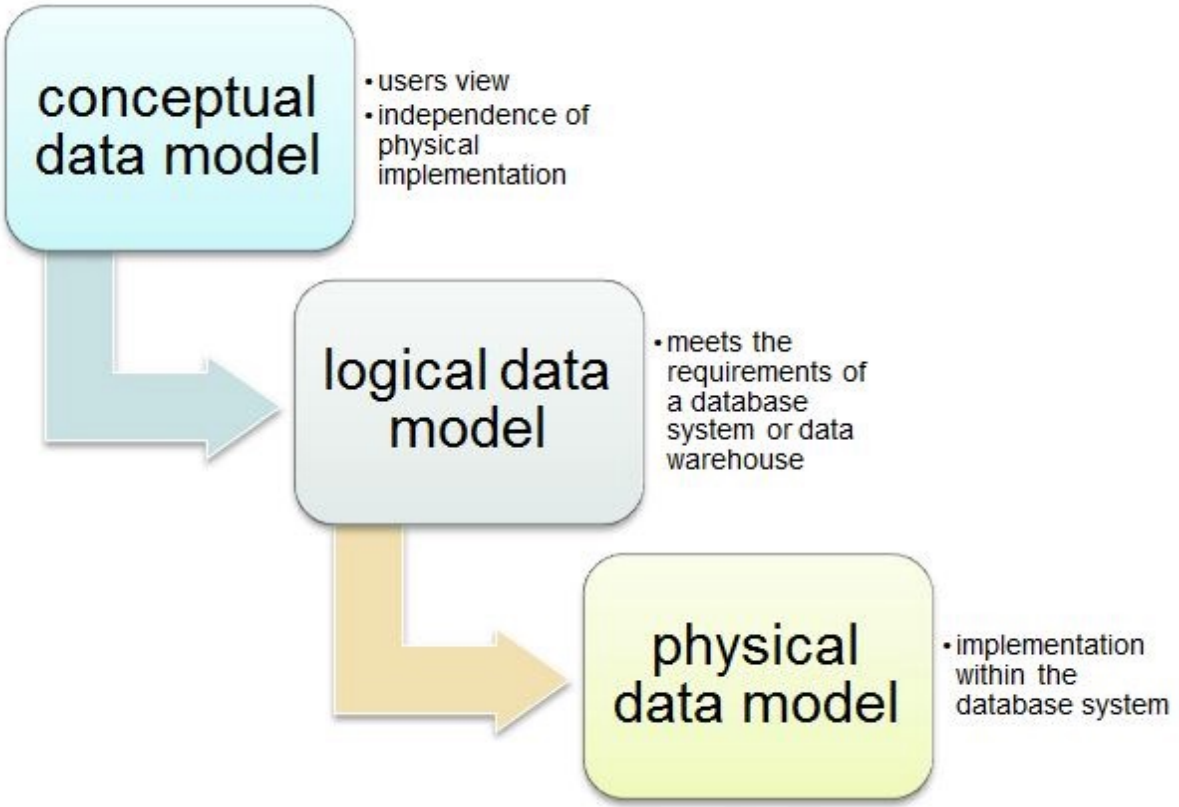

Student
Id
Name
Suburb
1108
Robert
Kew
3936
Glen
Bundoora
8507
Norman
Bundoora
8452
Mary
Balwyn
Takes
SID
SNO
1108
21
1108
23
1108
29
8507
23
8507
29
Subject
No
Name
Dept
21
Systems
CSCE
23
Database
CSCE
29
VB
CSCE
18
Algebra
Maths
Enrol
SID
Course
3936
101
1108
113
8507
101

A Brief History



Building Models Top Down vs. Bottom Up

- Top-Down
- Bottom-Up
- Using a Hybrid Approach – Middle Out



Top-down

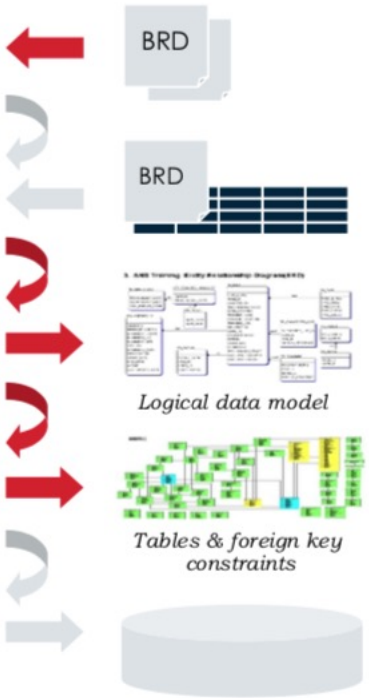
Step 1: Speak with the business representatives. Document the key business requirements and agree on high-level scope.

Step 2: Create detailed business requirement specification document with subscriber data requirement, business process and business rules

Step 3: Understand and document the business major attributes and definitions from business subject matter experts. Create logical data model.

Step 4: Verify the logical data model with the stakeholders and create physical data model

Step 5: Implement using the created physical model



© CHRISTOPHER BRADLEY (CDMP FELLOW)

Conclusion

- In this lesson you have learnt
  - Data model
  - Data modeling
  - E/R model
  - Ralational data model
  - Data modeling process

ĐẠI HỌC BÁCH KHOA

25 YEARS ANNIVERSARY SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attention!!!

soict.hust.edu.vn/ fb.com/groups/soict

CONCEPTUAL	LOGICAL	PHYSICAL
Defines the scope, audience, context for information	Defines key business concepts and their definitions	Represents core business rules and data relationships at a detailed level
Main purpose is for communication and agreement of scope and context	Main purpose is for communication and agreement of definitions and business logic	Provides enough detail for subsequent first cut physical design
Relationships optional. If shown, represent hierarchy.	Many-to-Many relationships OK	Many-to-Many relationships resolved
Cardinality not shown	Cardinality shown	Cardinality shown
No attributes shown	Attributes are optional. If shown, can be composite attributes to convey business meaning.	Attributes required and all attributes are atomic. Primary and foreign keys defined.
Not normalized (Relational models)	Not normalized (Relational models)	Fully normalized (Relational models)
Subject names should represent high-level data subjects or functional areas of the business	Concept names should use business terminology	Entity names may be more abstract
Subjects link to 1-M HDMs	Many concepts are supertypes, although subtypes may be shown for clarity	Supertypes all broken out to include sub-types
'One pager'	*Should* be a 'one pager'	May be larger than one page
Business-driven	Cross-functional & more senior people involved in HDM process with fewer IT.	Multiple smaller groups of specialists and IT folks involved in LDM process.
Informal notation	'Looser' notation required – some format construct needed, but ultimate goal is to be understood by a business user	Formal notation required
< 20 objects	< 100 objects	> 100 objects

© CHRISTOPHER BRADLEY (CDMP FELLOW)