---

marp: true theme: default class: invert size: 16:9 paginate: true footer: 國立陽明交通大學電子與光子學士學位學程 headingDivider: 1 style: | section::after { content: attr(data-

marpit-pagination) '/' attr(data-marpit-pagination-total); }

.columns { display: grid; grid-template-columns: repeat(2, minmax(0, 1fr)); gap: 1rem; } .columns img { width: 50%; } .middle-grid { display: grid; grid-template-columns: repeat(2, minmax(0, 1fr)); gap: 1rem; } .middle-grid img { width: 75%; } .grid { display: grid; grid-template-columns: 1fr 1fr; gap: 10px; } .grid img { width: 100%; } .red-text { color: red; }

.blue-text { color: blue;
}

.brown-text { color: brown;
}

.small-text { font-size: 0.75rem; }

Chapter 2: Data Models

- Data modeling is to build data models, which is the first step in the database design journey, serving as a bridge between real-world objects and the computer database.
- One big problem of database design is that designers, programmers, and end users see data in different ways, which introduce misunderstanding and increase communication cost.
- Database designers must obtain a precise description (data model) of the data's nature and environments within the organization to reduce communication efforts.

Data Modeling and Data Models

- Data modeling refers to the process of creating a specific data model for a determined problem domain (mini-world).
- Data modeling is an iterative, progressive process.
- A data model is a relatively simple representation of more complex real-world objects
    - Entity (table)
    - Attribute (column)

    - Relationship (linkage between tables)
    - Constrain

The Importance of Data Models

- Data models are a communication tool
    - End users know the business rule running in real world
    - Developers develop aps to manage data and transform data into information
    - People view data in different ways
        - Managers want a universal view of data
        - Staffs need details of data
- A good database system environment requires an overall database design based on an appropriate data model
- No appropriate data model, no good database system environment

Data Model Basic Building Blocks

- An entity is a person, thing or event about which data will be collected and stored
- An attribute is a characteristic (property) of an entity
- A relationship describes an association among entities
    - One-to-many (1:M or 1..*): PAINTER paints PAINTING
    - Many-to-many (M:N or *..*): EMPLOYEE learn SKILL
    - One-to-one (1:1 or 1..1): EMPLOYEE manage STORE
- A constraint is a restriction placed on the data to help data integrity
    - An employee's salary must have values between 6,000 and 350,000
    - Each class must have one and only one teacher

Business Rules

- A **business rule** is a brief, precise, and unambiguous description of a policy, procedure, or principle within a specific organization

    - made from company managers, policy makers, department managers, and written procedures
    - used to define entities, attributes, relationships, and constraints
- Example: A customer may generate many invoices' may be translated into data model
    - Customer and invoice are objects of interest and should be represented by respective entities
    - There is a **'generate'** relationship between customer and invoice
    - The generate relationship is one-to-many (1:M)

Naming Conventions

- Names should be descriptive and familiar to the users
- A good naming convention can
    - Make less confusion and reduce errors
    - Promote code consistently and readability
- Follow organization practice or develop at the start of project by considering
    - Should table name and column name be singular or plural? (student or students)
    - Should prefix tables or columns? (name or prod_name)
    - Should use capital letters for naming? (cap_cap, capCap or CapCap)
    - Which terminology should be selected? (user, person or people)

Supplement - Naming Conventions

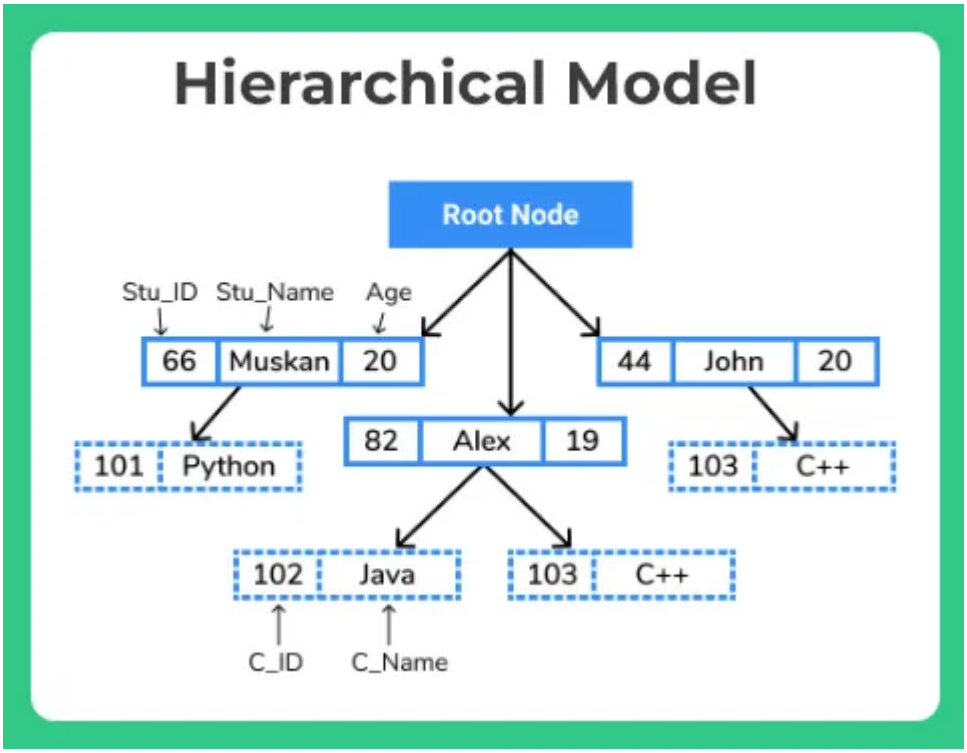
Udemy Video of naming conventions Naming conventions of MySQL

The Evolution of Data Models

Data models represent a lot of thought as to what a database is, what it should do, the types of structures that it should employ, and the technology that would be used to implement these structures  bg right:55% w:700 data models evolution

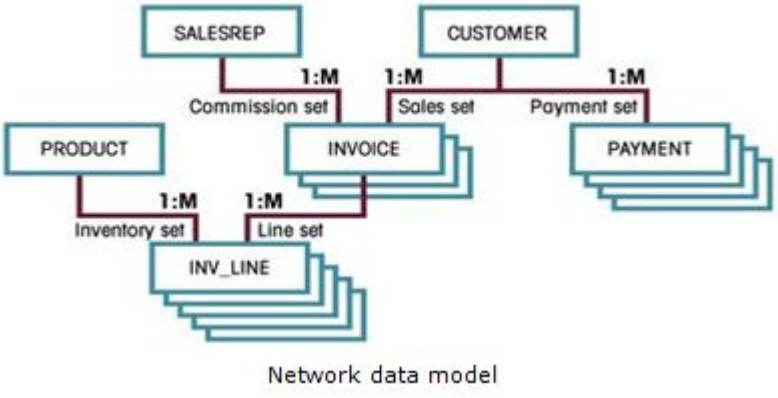
Hierarchical Models

- The hierarchical model organizes the data into a tree structure which consist of a single root node where each record is having a parent record and many child records and expands like a tree



Network Models

- In the network model, the user perceives the network database as a collection of records in 1:M relationships. However, unlike the hierarchical model, the network model allows a record to have more than one parent.



Network data model

Database Concepts Inherited from Network Model

- The **schema** is the conceptual and structural definition of a whole database. Once you claim the schema of a database, it must

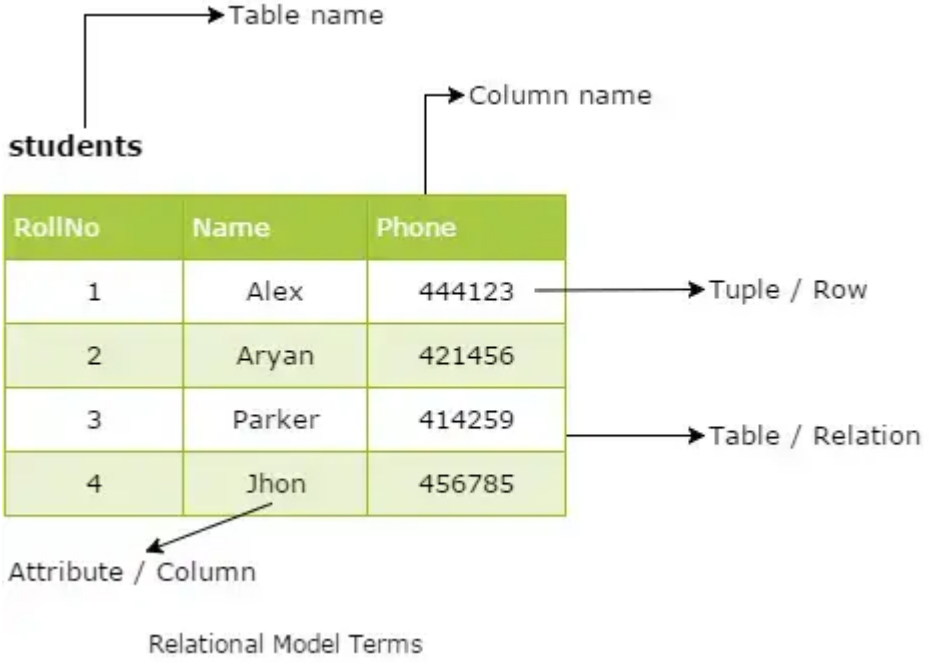
now no longer be modified often because it will distort the data organization inside the Database.

- The **data manipulation language (DML)** defines the way to insert, read, update, delete data in database
- A **schema data definition language (DDL)** enables the DBA to define the schema components (create, drop, alter table, create index or trigger)

Relational Model

The relational model's foundation is a mathematical concept known as a relation, which is introduced by Edgar F. Codd in 1969.

- Relation: a table with columns and rows.
- Attribute: a named column of a relation.
- Domain: the set of allowable values for one or more attributes.
- Tuple: a row of a relation



Relational Model Terms

Relational Diagram


Linking relational tables


A relational diagram

Supplement of Relational Model

- Relation schema
- Relational database schema
- Degree of a relation
- Cardinality of a relation
- Relation state (or relation instance) Relational model terminology

Entity Relationship Model

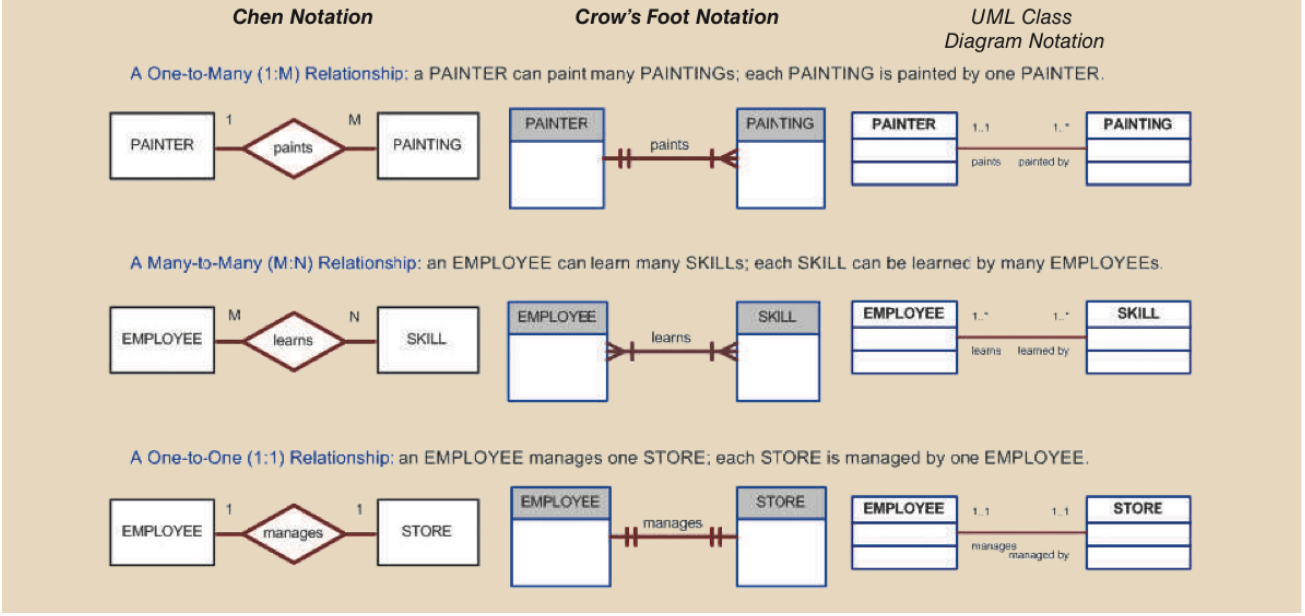
- Although the relational model was a vast improvement over the hierarchical and network models, it still lacked the features that would make it an effective database design tool.
- Database designers prefer to use a graphical tool in which entities and their relationships are pictured.
- The **entity relationship model (ERM)** using *graphical representations* to model database components has become a widely accepted standard for data modeling.
- The relational data model and ERM combined to provide the foundation for tightly structured database design

Entity Relationship Model Notation

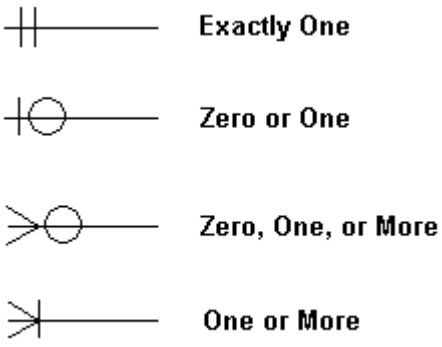
- Entity – an entity is represented in the ERD by a rectangle (entity box)
- Attributes – each entity consists of a set of attributes that describes particular characteristics of the entity
- Relationships – relationships describe associations among data


bg right:50% w:600 ER model

FIGURE 2.3 THE ER MODEL NOTATIONS



Crow's Foot Notations



Relational Model vs Entity Relationship Model

ER Model first, then converted into Relational Model for DBMS implementation.

Aspect	ER Model	Relational Model
Used For	Conceptual database design	Logical database implementation
Representation	ER Diagram (graphical)	Tables (relational schema)
Elements	Entities, Attributes, Relationships	Tables, Attributes, Tuples

Aspect	ER Model	Relational Model
Constraints	Cardinality in ER Diagram	Primary/Foreign keys, SQL constraints
Conversion	Converted to Relational Model	Implemented in DBMS

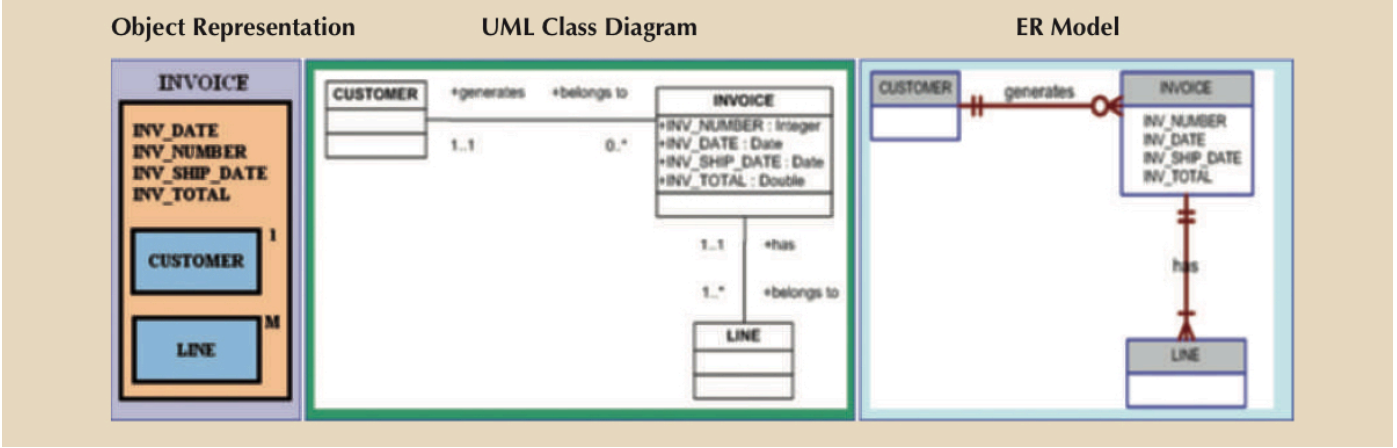
Object-Oriented Model

In the object-oriented data model (OODM), both data and its relationship are contained in a single structure known as an object

- **Object**: an abstraction of a real-world entity
- **Attributes**: describe the properties of an object
- **Method**: represents a real-world action
- **Class**: a collection of similar objects with shared structure and behavior
- **Inheritance**: an object within the *class hierarchy* to inherit the attributes and methods of the classes above it. (class EMPLOYEE and CUSTOMER can be created as subclasses inherit from the class PERSON)
- OODM are typically depicted using Unified Modeling Language (UML) class diagrams

OODM Diagram

FIGURE 2.4 A COMPARISON OF THE OO, UML, AND ER MODELS



ERDM and O/R DBMS

- The extended relational data model (ERDM) adds many of the OO model's features within the simpler relational database structure
- A DBMS based on the ERDM is an object/relational database management system (O/R DBMS)

Comparison of RDBMS, OODBMS and O/R DBMS

Feature	RDBMS	OODBMS	O/R DBMS
Tables & SQL	✅ Yes	❌ No	✅ Yes
Objects & Classes	❌ No	✅ Yes	✅ Yes
Inheritance	❌ No	✅ Yes	✅ Yes
Encapsulation (Methods in DB)	❌ No	✅ Yes	✅ Yes
Complex Data Types	❌ No	✅ Yes	✅ Yes

Products of O/R DBMS

- PostgreSQL (most commonly used O/R DBMS)
- Oracle Database (with Object-Relational Features)
- Microsoft SQL Server (limited Object-Relational capabilities)

Emerging Data Models: Big Data and NoSQL

- **Big Data** refers to a movement to find new and better ways to manage large amounts which DBMS can not manage
- Big Data characteristics (3 Vs) : volume, velocity, and variety
- Most frequently used Big Data technologies
    - Hadoop: an **ecosystem** provides a collection of softwares to operate big data
    - Hadoop Distributed File System (HDFS) is a fault-tolerant **file storage** system
    - MapReduce is an distributed **computational framework**

- NoSQL database is a large-scale distributed **database system** that stores unstructured and semi-structured data in efficient ways

NoSQL Databases

- Schemaless
- Horizontal scalability
- Distributed data store
- Lower cost
- Non-relational
- Handle large volume of data

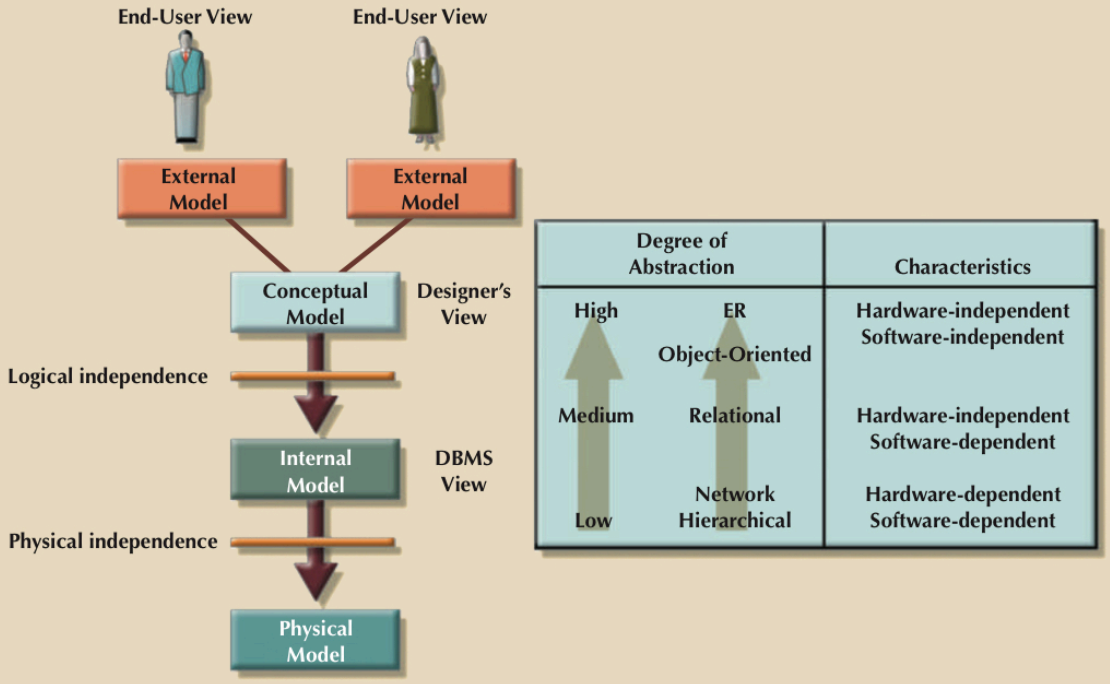
RDBMS (Relational DBMS) vs NoSQL

RDBMS	NoSQL
Structured data with a rigid schema	Unstructured, Semi-structured data with a flexible schema
Storage in rows and columns	Data are stored in Key/Value pairs database, Columnar database, Document database, Graph Database.
Scale up	Scale out
SQL server, Oracle, mySQL	MongoDB, HBase, Cassandra
SQL language	Solution-specific method

Degrees of Data Abstraction

- External, Conceptual, Internal and Physical levels

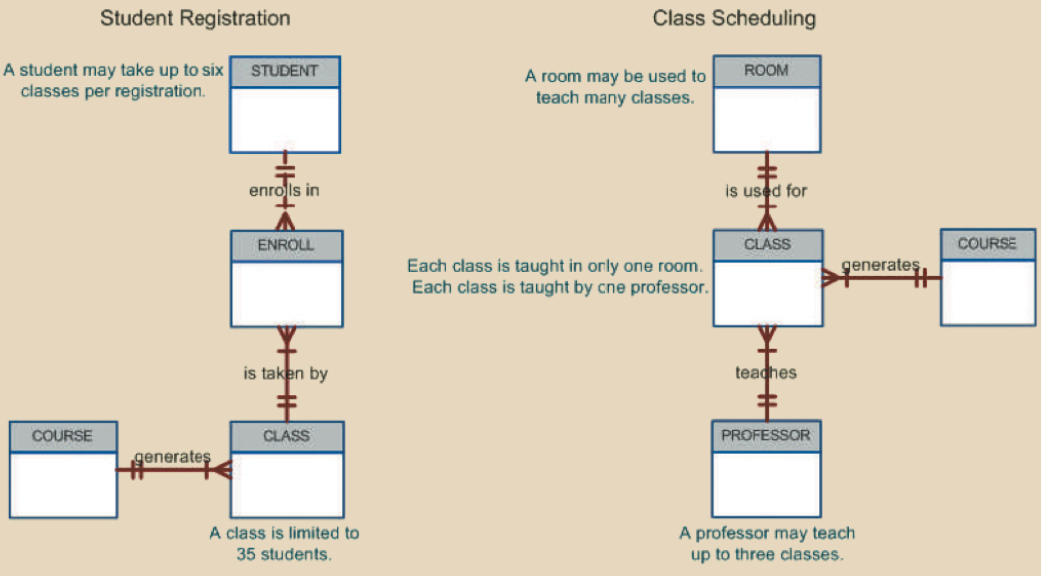
FIGURE 2.6 DATA ABSTRACTION LEVELS



External Model

- The end users' view of the data environment
- Use **ER diagrams** to represent the external model
- The external views represent **subsets** of the database
  - Easy to scope and communicate specific data required to support targeted end users
  - Ensure **security** constraints in the database design

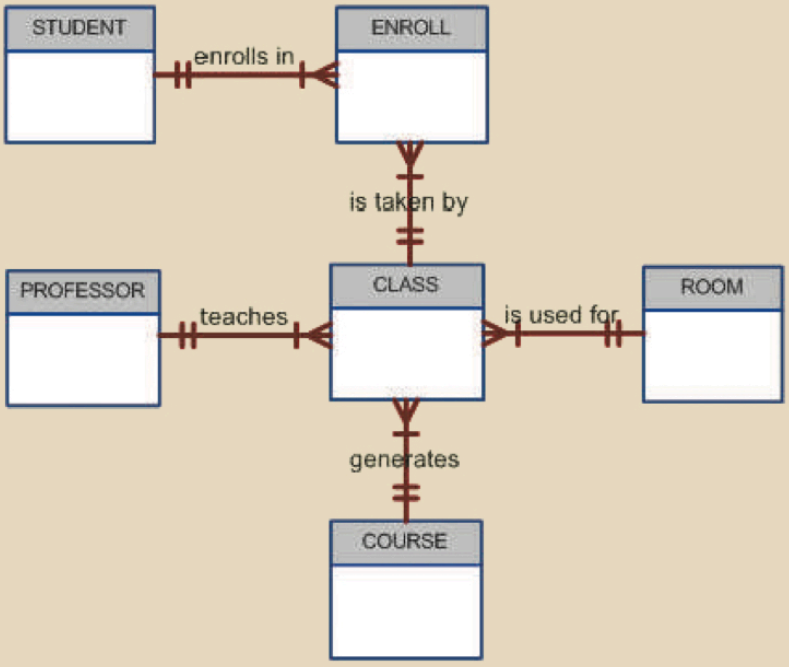
FIGURE 2.7 EXTERNAL MODELS FOR TINY COLLEGE



Conceptual Model

- A **global view** of the entire database by the entire organization
- Use **ER diagrams** to represent the conceptual model
- Identify and high-level describe main data objects
- Independent of both software and hardware
- The term **conceptual design** refers to creating a conceptual data model by ER diagrams

FIGURE 2.8 A CONCEPTUAL MODEL FOR TINY COLLEGE

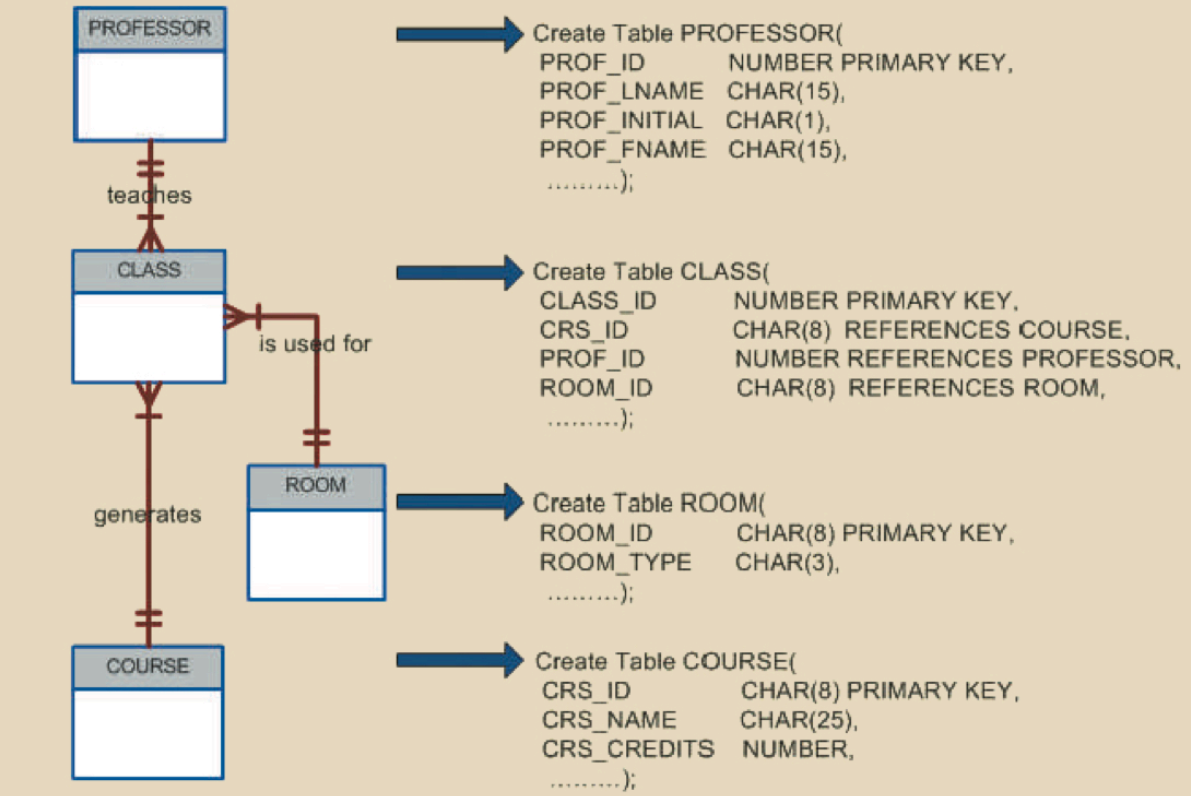


Internal Model

- Use the database constructs of the chosen DBMS to match the conceptual model's characteristics and constraints to build the internal model
- The term **logical design** refers to creating a logical data model by a set of SQL statements

- Software dependent and hardware independent

FIGURE 2.9 INTERNAL MODEL FOR TINY COLLEGE



Physical Model

- Operates at the lowest level of abstraction, describing which physical storage device the data is saved and how to access the data
- The term **physical design** refers to define data storage organization, security control, performance measure
- Both software and hardware dependent

Levels of Data Abstraction

TABLE 2.4

LEVELS OF DATA ABSTRACTION

MODEL	DEGREE OF ABSTRACTION	FOCUS	INDEPENDENT OF
External	High	End-user views	Hardware and software
Conceptual	↕	Global view of data (database model independent)	Hardware and software
Internal	↕	Specific database model	Hardware
Physical	Low	Storage and access methods	Neither hardware nor software

Review Questions

- Why data models are important?
- What are the data model basic building blocks
- How have the major data models evolved
- Explain NoSQL characteristics
- What are the four levels of data abstraction

Homework #A

資料庫課程作業(A)