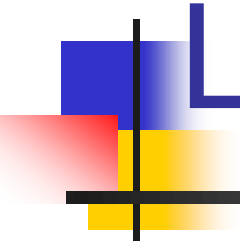


# Lecture 2

## Relational Database Model

# Data Modeling

- Data models are representations of complex real-world data structures
- Iterative and progressive process of creating a specific data model for a determined problem
- Abstraction for the creation of a good database
- Facilitate communication between the designer, applications programmer, and end user

# Data Models

- Entity: Unique and distinct object used to collect and store data
- Attribute: Characteristic of an entity
- Relationship: Describes the association between entities
  - One-to-many (1:M)
  - Many-to-many (M:N)
  - One-to-one (1:1)
- Constraint: Set of rules placed on the data to ensure data integrity

# Business Rules

- Description of policies, procedures, or principles within a specific organization
- Enables the defining of basic building blocks
- Describes the main and distinguishing characteristics of the data
- Ensures accurate data models
- Must be in writing and kept up to date

# Discovering Business Rules

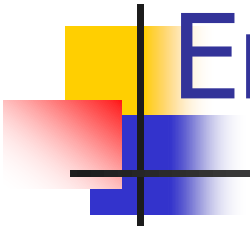
- Sources of business rules:
  - Company managers
  - Policy makers
  - Department managers
  - Written documentation
    - Procedures
    - Standards
    - Operations manuals
  - Direct interviews with end users

# Translating Business Rules

- Nouns translate into entities
- Verbs translate into relationships among entities
- Relationships are bidirectional
- Questions to identify the relationship type:
  - How many instances of B are related to one instance of A?
  - How many instances of A are related to one instance of B?

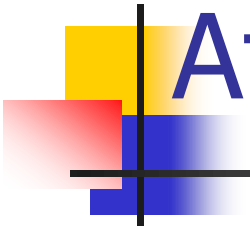
# Entities

- Person, place, or thing for which data can be collected and stored
- Relation is an entity (table) in a relational model
- Consists of a matrix of intersecting tuples (rows) and attributes (columns)



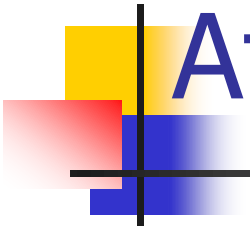
# Entities

- An instance or occurrence is a tuple (row) of real data in an entity (table)
- Each entity name must be distinct within a database
- A tuple should be distinct with no duplicates



# Attributes

- Piece of data to be stored about each entity instance
- Descriptive property or characteristic of the entity
- An attribute name must be distinct within an entity

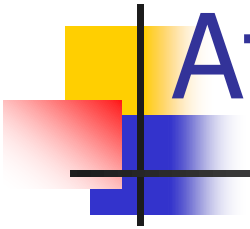


# Attributes

- Compound attribute
  - Attribute that can be subdivided to yield additional attributes
  - Example:

    Address

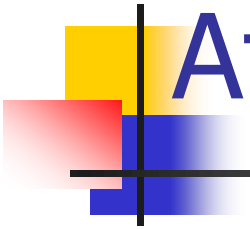
    (Street, City, Province)



# Attributes

- Single-valued attribute
  - Attribute that has only a single value
  - Cannot be subdivided
  - Example:

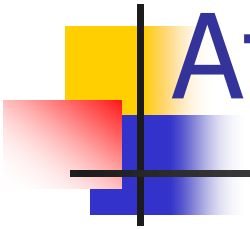
    Birth_Date



# Attributes

- Multi-valued attribute
  - Attribute that has many values
  - Example:

    Student_Contacts



# Attributes

- Derived attribute
  - Attribute whose value is calculated from other attributes using an algorithm
  - Often not stored in the database
  - Example:

    Annual = Monthly * 12

# Constraints

- Data Type
  - Type of data that can be stored in the attribute
  - Example:  Character, Numeric, Date, Binary
- Attribute Domain
  - Allowable attribute values
  - Example:  (M/F, Y/N, 1200-1499)
- Default Value
  - Value that will be recorded if not specified
  - Example:  Invoice date defaults to today's date

# Constraints

- NULL / NOT NULL
  - Specifies if attribute is required (NOT NULL) or not required (NULL)
  - Example:  Birth_Date NOT NULL
- UNIQUE
  - No duplicate value exist for the attribute in the entity
  - Example: SIN

# Characteristics of a Relational Table

TABLE 3.1

| CHARACTERISTICS OF A RELATIONAL TABLE | |
|---|---|
| 1 | A table is perceived as a two-dimensional structure composed of rows and columns. |
| 2 | Each table row (tuple) represents a single entity occurrence within the entity set. |
| 3 | Each table column represents an attribute, and each column has a distinct name. |
| 4 | Each intersection of a row and column represents a single data value. |
| 5 | All values in a column must conform to the same data format. |
| 6 | Each column has a specific range of values known as the attribute domain. |
| 7 | The order of the rows and columns is immaterial to the DBMS. |
| 8 | Each table must have an attribute or combination of attributes that uniquely identifies each row. |

# Schema and Subschema

- Schema
  - Conceptual organization of the entire database as viewed by the database administrator
- Subschema
  - Portion of the database seen by application programs which produce the desired information using the data contained within the database

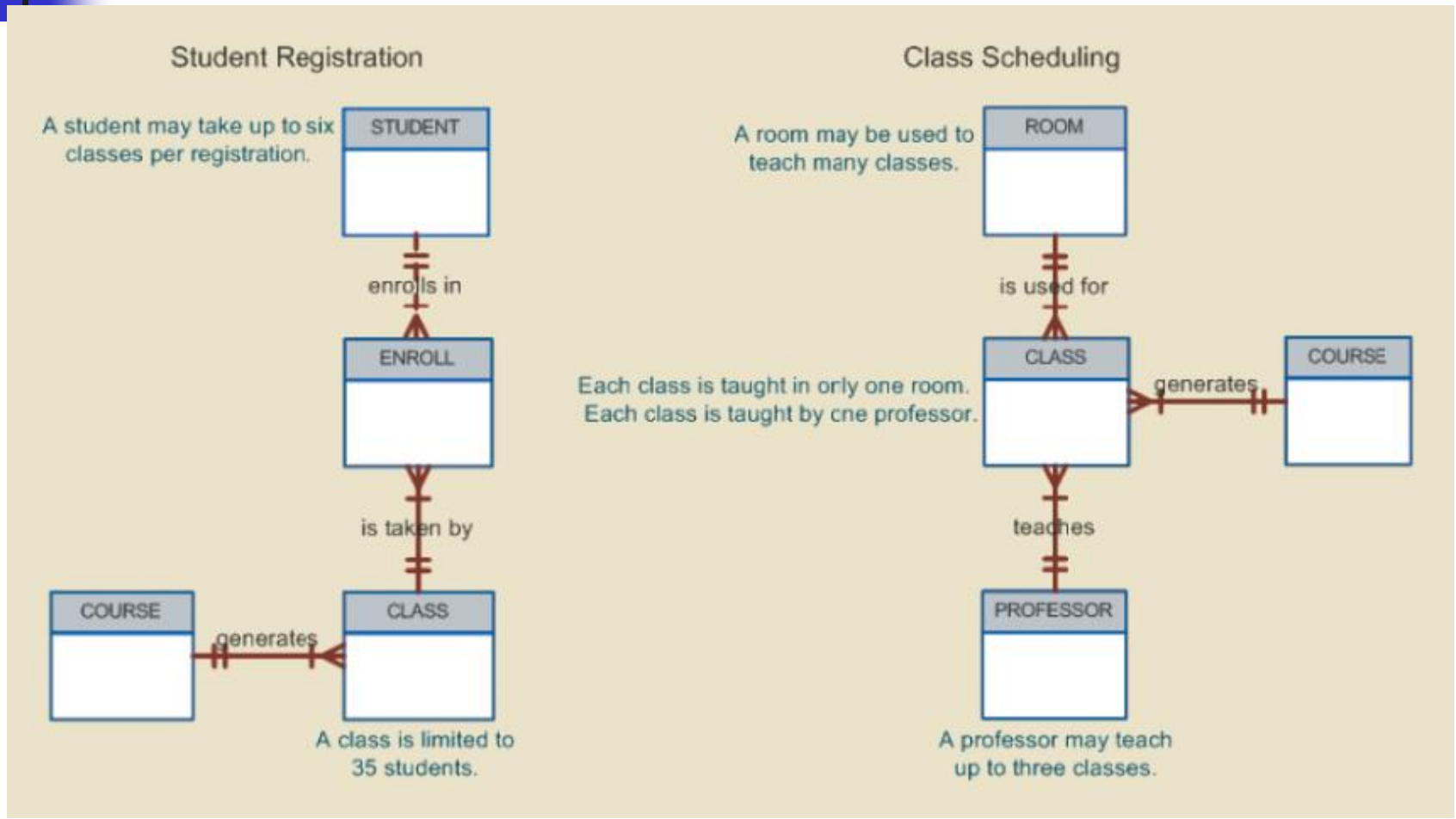
# Level Architecture

- External model represents the end user's view of the data environment
- Conceptual model describes what data is stored and its relationships for the entire database
- Internal model represents the database as seen by the DBMS
- Physical model describes how the data is stored physically

# External Model

- End user's view of the data environment
- Identify specific data required to support each business unit's operations
- External schema
  - Specific representation of an external view

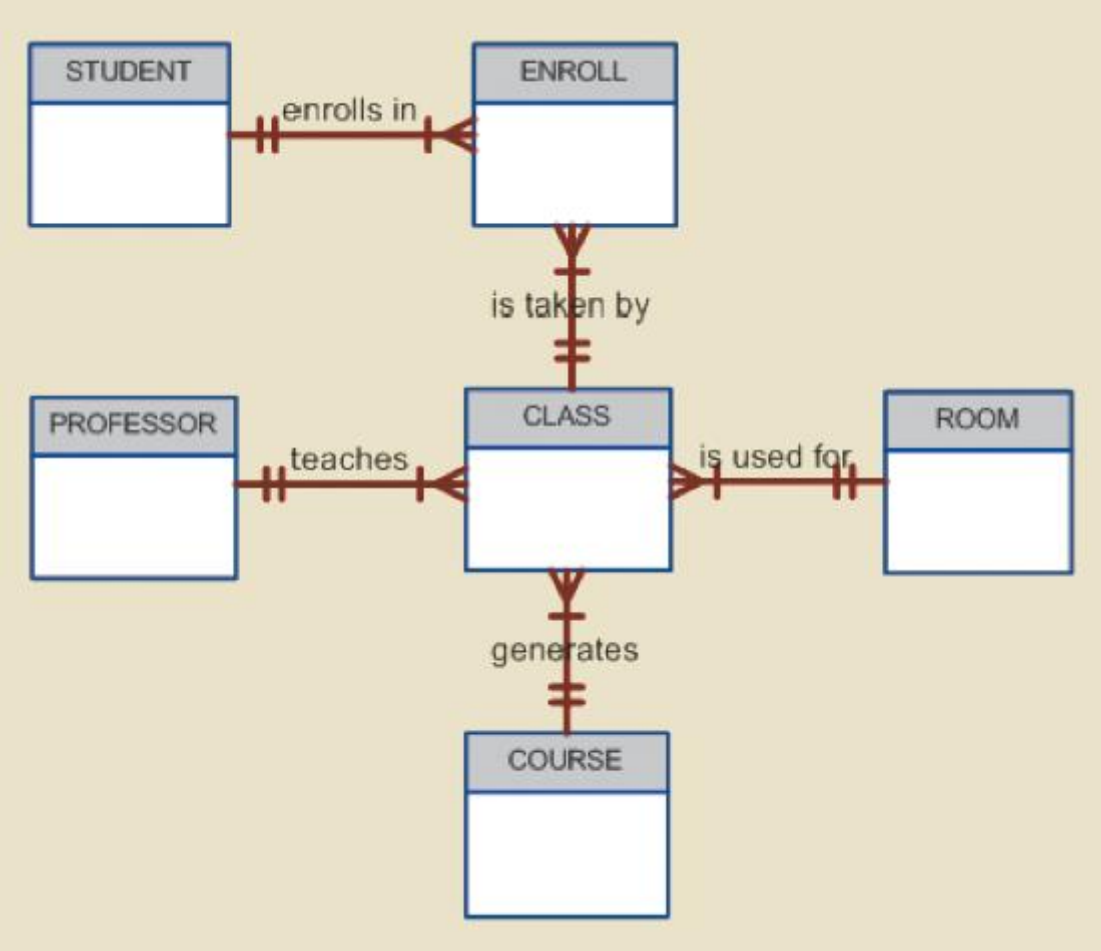
# External Models for Tiny College



# Conceptual Model

- Represents a global view of the entire database by the entire organization
- Conceptual schema
  - All external views integrated into a single global view of the data in the enterprise
- Software and hardware independence
- Logical design
  - Task of creating a conceptual data model

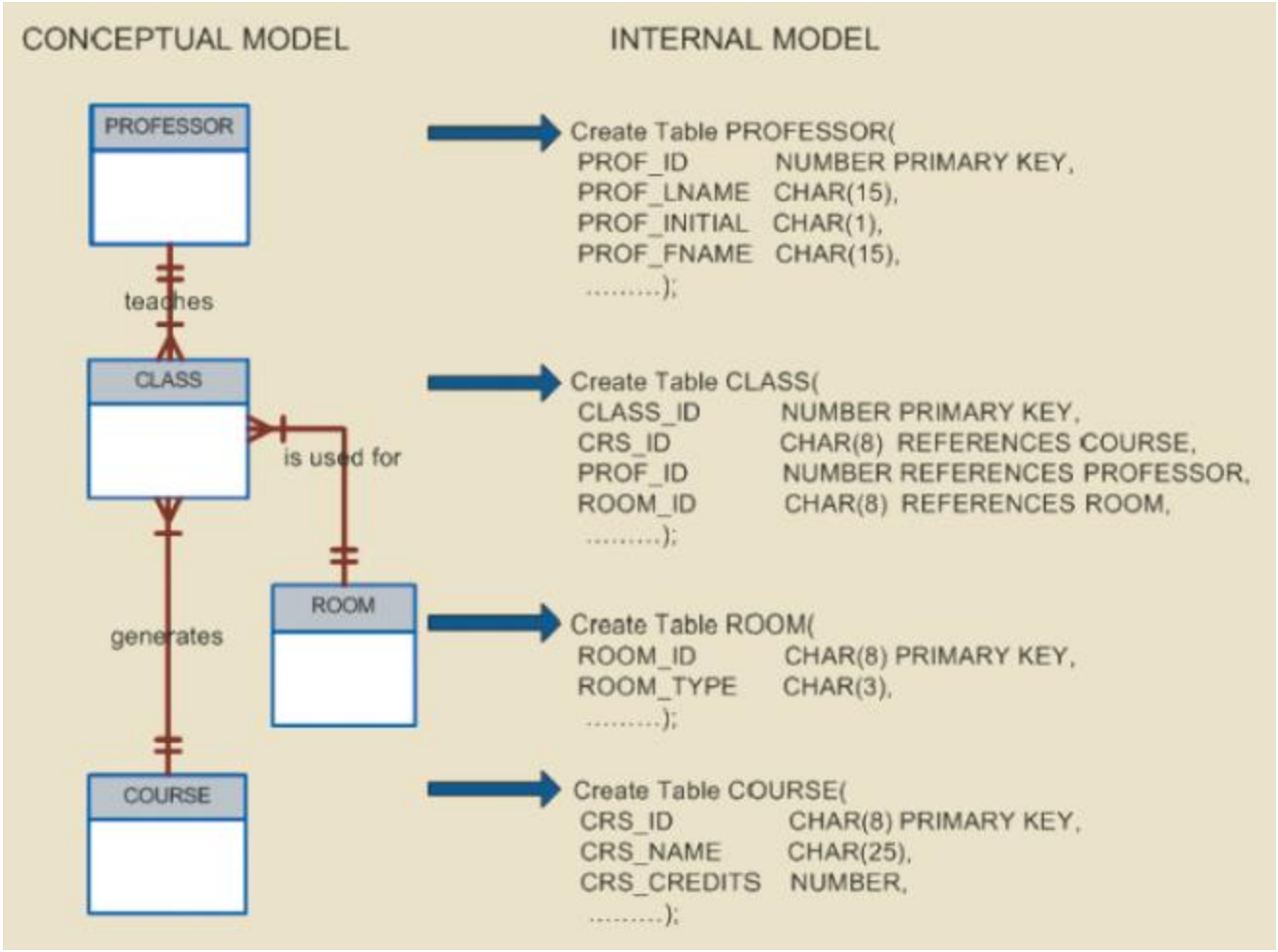
# Conceptual Model For Tiny College



# Internal Model

- Representation of the database as seen by the DBMS mapping conceptual model to the DBMS
- Internal schema
  - Specific representation of an internal model
- Software dependent and hardware independent
- Logical independence
  - Change internal model without affecting conceptual model

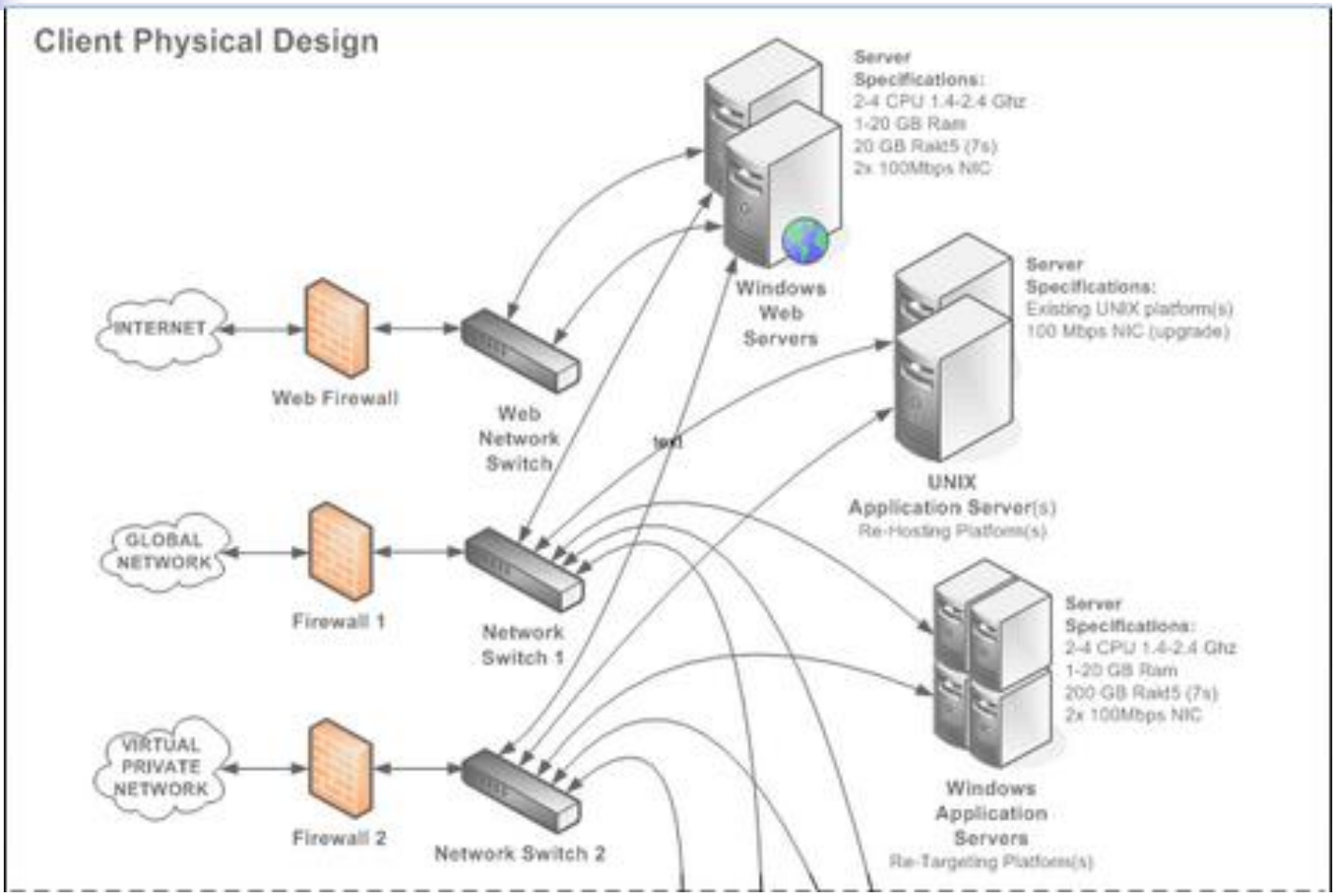
# Internal Model for Tiny College



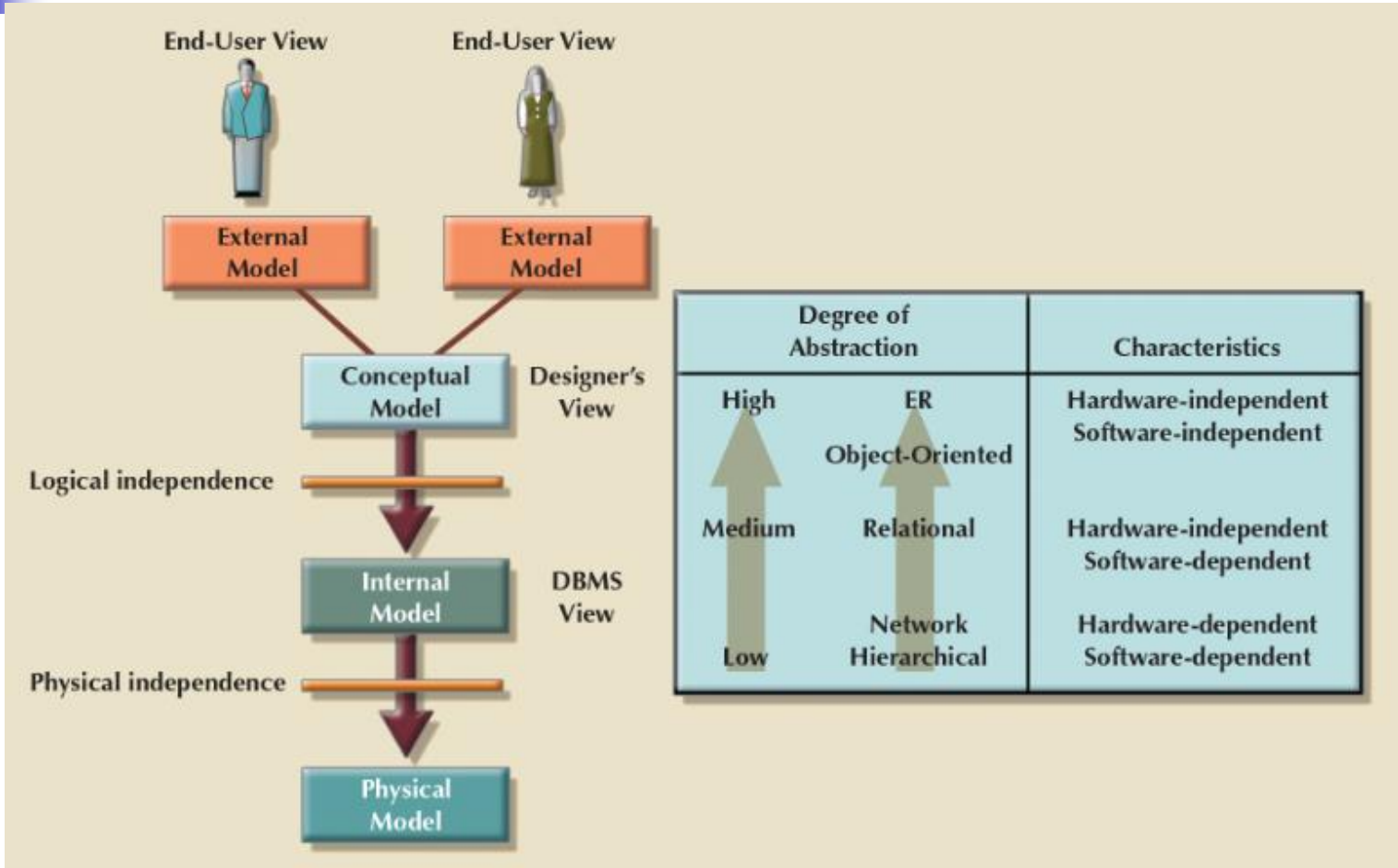
# Physical Model

- Operates at lowest level of abstraction
- Describes the way data is saved on storage media
- Requires the definition of physical storage and data access methods
- Physical independence
  - Changes in physical model do not affect internal model

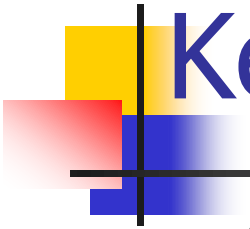
# Physical Model



# Data Abstraction Levels

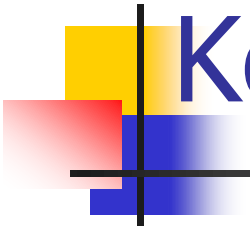



| Degree of Abstraction | | Characteristics |
|---|---|---|
| High | ER<br>Object-Oriented | Hardware-independent<br>Software-independent |
| Medium | Relational | Hardware-independent<br>Software-dependent |
| Low | Network<br>Hierarchical | Hardware-dependent<br>Software-dependent |



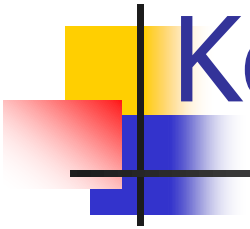
# Keys

- Consists of one or more attributes that uniquely identify a tuple in an entity
- Ensures each tuple in an entity is uniquely identifiable
- Establishes relationships among the entities
- Ensures integrity of the data



# Keys

- Candidate key
- Primary key
- Composite key
- Secondary key
- Foreign key



# Keys

- Candidate key
  - Attributes that may be selected to become the primary identifier of instances of an entity
  - Example: Emp_Id and SIN
- Primary key
  - Attribute selected to uniquely identify each entity instance
  - Cannot contain null entries (absence of any data value)
  - Example: Emp_Id

# Keys

- Composite key
  - Composed of more than one attribute to uniquely identify an instance of an entity
- Secondary key
  - Any attribute or combination of attributes used strictly for data retrieval purposes
- Foreign key
  - Attribute in one entity whose value either match the primary key in another entity or is null

# Example - Simple Relational Database

**Table name: PRODUCT**
**Primary key: PROD_CODE**
**Foreign key: VEND_CODE**

**Database name: Ch03_SaleCo**

| PROD_CODE | PROD_DESCRIPT | PROD_PRICE | PROD_ON_HAND | VEND_CODE |
|---|---|---|---|---|
| 001278-AB | Claw hammer | 12.95 | 23 | 232 |
| 123-21UUY | Houselite chain saw, 16-in. bar | 189.99 | 4 | 235 |
| QER-34256 | Sledge hammer, 16-lb. head | 18.63 | 6 | 231 |
| SRE-657UG | Rat-tail file | 2.99 | 15 | 232 |
| ZZX/3245Q | Steel tape, 12-ft. length | 6.79 | 8 | 235 |

**link**

**Table name: VENDOR**
**Primary key: VEND_CODE**
**Foreign key: none**

| VEND_CODE | VEND_CONTACT | VEND_AREACODE | VEND_PHONE |
|---|---|---|---|
| 230 | Shelly K. Smithson | 608 | 555-1234 |
| 231 | James Johnson | 615 | 123-4536 |
| 232 | Annelise Crystall | 608 | 224-2134 |
| 233 | Candice Wallace | 904 | 342-6567 |
| 234 | Arthur Jones | 615 | 123-3324 |
| 235 | Henry Ortozo | 615 | 899-3425 |

# Integrity Rules

- Domain Constraints
  - Values are restricted by the domain of the attribute
    - Range of permissible values for an attribute
- Entity Integrity
  - All values in the primary keys must be unique
    - No part of the primary key may be null
- Referential Integrity
  - Foreign key matches a primary key value in an entity to which it is related or is a null entry

# Database Management System

- Structured Query Language (SQL) is the most commonly used query language
- Uses Data Definition Language (DDL) to define the database and database objects
- Uses Data Manipulation Language (DML) to retrieve, insert, update, and delete data stored in the database
- Uses Data Control Language (DCL) to control access to data stored in a database

# Relational Model

- Developed by E. F. Codd (IBM) in 1970
- Based on mathematical theory
- Introduces normalization concept
- Set-oriented using relations / tables
- Matrix consisting of the intersection of row and column
- In 1985, Codd published a list of 12 rules to define a relational database system

# Relational Algebra

- Defines a theoretical way of manipulating a table contents using relational operators
- Helps in the understanding of the Data Manipulation Language (DML)
- Structured Query Language (SQL) is based on relational algebra

# Property of Closure

- Operations that work on one or more relations produces a new relation without changing the original relation
- Result of an algebraic operation on a SQL command produces another temporary relation

# Relational Operators

- Select
- Project
- Union
- Intersect
- Difference
- Product
- Join

# Select

- Yields a horizontal subset of a table

**Original table**

| P_CODE | P_DESCRIPT | PRICE |
|---|---|---|
| 123456 | Flashlight | 5.26 |
| 123457 | Lamp | 25.15 |
| 123458 | Box Fan | 10.99 |
| 213345 | 9v battery | 1.92 |
| 254467 | 100W bulb | 1.47 |
| 311452 | Powerdrill | 34.99 |

**SELECT ALL yields**

**New table**

| P_CODE | P_DESCRIPT | PRICE |
|---|---|---|
| 123456 | Flashlight | 5.26 |
| 123457 | Lamp | 25.15 |
| 123458 | Box Fan | 10.99 |
| 213345 | 9v battery | 1.92 |
| 254467 | 100W bulb | 1.47 |
| 311452 | Powerdrill | 34.99 |

**SELECT only PRICE less than $2.00 yields**

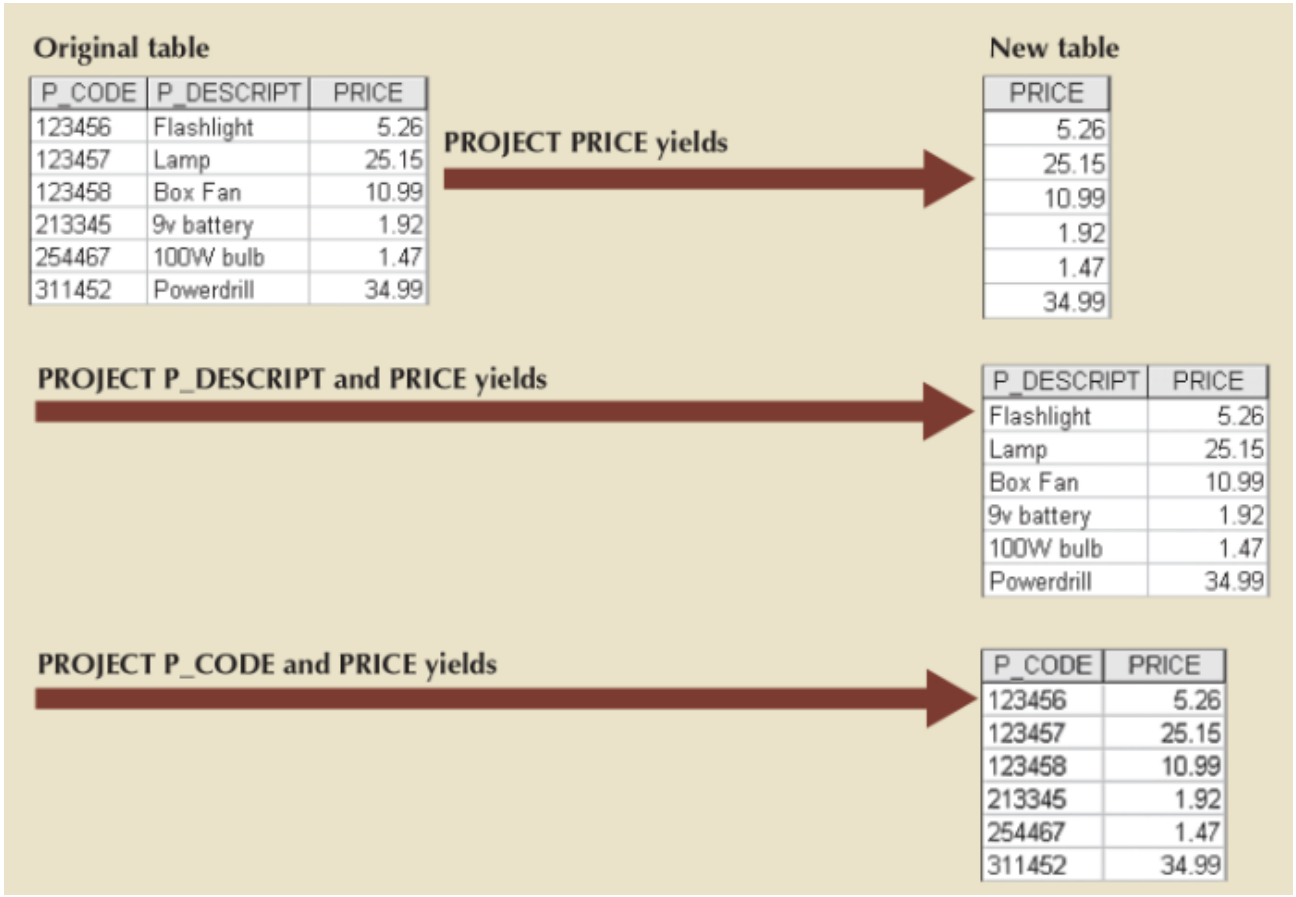
| P_CODE | P_DESCRIPT | PRICE |
|---|---|---|
| 213345 | 9v battery | 1.92 |
| 254467 | 100W bulb | 1.47 |

**SELECT only P_CODE = 311452 yields**

| P_CODE | P_DESCRIPT | PRICE |
|---|---|---|
| 311452 | Powerdrill | 34.99 |

# Project

- Yields a vertical subset of a table



# Union

- Combines all the rows from two tables excluding duplicate rows
- Tables must be union-compatible

| P_CODE | P_DESCRIPT | PRICE |
|---|---|---|
| 123456 | Flashlight | 5.26 |
| 123457 | Lamp | 25.15 |
| 123458 | Box Fan | 10.99 |
| 213345 | 9v battery | 1.92 |
| 254467 | 100W bulb | 1.47 |
| 311452 | Powerdrill | 34.99 |

UNION

| P_CODE | P_DESCRIPT | PRICE |
|---|---|---|
| 345678 | Microwave | 160.00 |
| 345679 | Dishwasher | 500.00 |
| 123458 | Box Fan | 10.99 |

yields

| P_CODE | P_DESCRIPT | PRICE |
|---|---|---|
| 123456 | Flashlight | 5.26 |
| 123457 | Lamp | 25.15 |
| 123458 | Box Fan | 10.99 |
| 213345 | 9v battery | 1.92 |
| 254467 | 100W bulb | 1.47 |
| 311452 | Powerdrill | 34.99 |
| 345678 | Microwave | 160 |
| 345679 | Dishwasher | 500 |

# Intersect

- Yields only the rows that appear in both tables
- Tables must be union-compatible

| STU_FNAME | STU_LNAME |
|---|---|
| George | Jones |
| Jane | Smith |
| Peter | Robinson |
| Franklin | Johnson |
| Martin | Lopez |

**INTERSECT**

| EMP_FNAME | EMP_LNAME |
|---|---|
| Franklin | Lopez |
| William | Turner |
| Franklin | Johnson |
| Susan | Rogers |

**yields**

| STU_FNAME | STU_LNAME |
|---|---|
| Franklin | Johnson |

# Difference

- Yields all the rows in one relation not found in the other relation
- Relations must be union-compatible

| STU_FNAME | STU_LNAME |
|---|---|
| George | Jones |
| Jane | Smith |
| Peter | Robinson |
| Franklin | Johnson |
| Martin | Lopez |

**DIFFERENCE**

| EMP_FNAME | EMP_LNAME |
|---|---|
| Franklin | Lopez |
| William | Turner |
| Franklin | Johnson |
| Susan | Rogers |

**yields**

| STU_FNAME | STU_LNAME |
|---|---|
| George | Jones |
| Jane | Smith |
| Peter | Robinson |
| Martin | Lopez |

# Product

- Yields all possible pairs of rows from two relations
- Also known as Cartesian Product

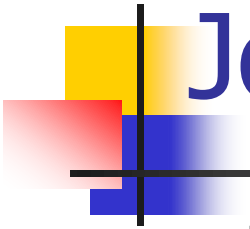
| P_CODE | P_DESCRIPT | PRICE |
|---|---|---|
| 123456 | Flashlight | 5.26 |
| 123457 | Lamp | 25.15 |
| 123458 | Box Fan | 10.99 |
| 213345 | 9v battery | 1.92 |
| 254467 | 100W bulb | 1.47 |
| 311452 | Powerdrill | 34.99 |

**PRODUCT**

| STORE | AISLE | SHELF |
|---|---|---|
| 23 | W | 5 |
| 24 | K | 9 |
| 25 | Z | 6 |

**yields**

| P_CODE | P_DESCRIPT | PRICE | STORE | AISLE | SHELF |
|---|---|---|---|---|---|
| 123456 | Flashlight | 5.26 | 23 | W | 5 |
| 123456 | Flashlight | 5.26 | 24 | K | 9 |
| 123456 | Flashlight | 5.26 | 25 | Z | 6 |
| 123457 | Lamp | 25.15 | 23 | W | 5 |
| 123457 | Lamp | 25.15 | 24 | K | 9 |
| 123457 | Lamp | 25.15 | 25 | Z | 6 |
| 123458 | Box Fan | 10.99 | 23 | W | 5 |
| 123458 | Box Fan | 10.99 | 24 | K | 9 |
| 123458 | Box Fan | 10.99 | 25 | Z | 6 |
| 213345 | 9v battery | 1.92 | 23 | W | 5 |
| 213345 | 9v battery | 1.92 | 24 | K | 9 |
| 213345 | 9v battery | 1.92 | 25 | Z | 6 |
| 311452 | Powerdrill | 34.99 | 23 | W | 5 |
| 311452 | Powerdrill | 34.99 | 24 | K | 9 |
| 311452 | Powerdrill | 34.99 | 25 | Z | 6 |
| 254467 | 100W bulb | 1.47 | 23 | W | 5 |
| 254467 | 100W bulb | 1.47 | 24 | K | 9 |
| 254467 | 100W bulb | 1.47 | 25 | Z | 6 |



# Joins

- Essential operation in relational algebra
- Builds a relation from two specified relations on the basis of matching a common attribute over the same domain
- Natural Join
  - Joins tables by selecting only rows with common values in their common attributes
- Outer Join
  - Matched pairs are retained and unmatched values in the other table are left null

# Inner Join

- Returns matched rows only from the tables that are being joined

**Table name: CUSTOMER**

| CUS_CODE | CUS_LNAME | CUS_ZIP | AGENT_CODE |
|---|---|---|---|
| 1132445 | Walker | 32145 | 231 |
| 1217782 | Adares | 32145 | 125 |
| 1312243 | Rakowski | 34129 | 167 |
| 1321242 | Rodriguez | 37134 | 125 |
| 1542311 | Smithson | 37134 | 421 |
| 1657399 | Vanloo | 32145 | 231 |

**Table name: AGENT**

| AGENT_CODE | AGENT_PHONE |
|---|---|
| 125 | 6152439887 |
| 167 | 6153426778 |
| 231 | 6152431124 |
| 333 | 9041234445 |

| CUS_CODE | CUS_LNAME | CUS_ZIP | CUSTOMER.AGENT_CODE | AGENT.AGENT_CODE | AGENT_PHONE |
|---|---|---|---|---|---|
| 1217782 | Adares | 32145 | 125 | 125 | 6152439887 |
| 1321242 | Rodriguez | 37134 | 125 | 125 | 6152439887 |
| 1312243 | Rakowski | 34129 | 167 | 167 | 6153426778 |
| 1132445 | Walker | 32145 | 231 | 231 | 6152431124 |
| 1657399 | Vanloo | 32145 | 231 | 231 | 6152431124 |

# Left Outer Join

- Yields all rows in the first table including those that do not have a matching value in the second table

**Table name: CUSTOMER**

| CUS_CODE | CUS_LNAME | CUS_ZIP | AGENT_CODE |
|---|---|---|---|
| 1132445 | Walker | 32145 | 231 |
| 1217782 | Adares | 32145 | 125 |
| 1312243 | Rakowski | 34129 | 167 |
| 1321242 | Rodriguez | 37134 | 125 |
| 1542311 | Smithson | 37134 | 421 |
| 1657399 | Vanloo | 32145 | 231 |

**Table name: AGENT**

| AGENT_CODE | AGENT_PHONE |
|---|---|
| 125 | 6152439887 |
| 167 | 6153426778 |
| 231 | 6152431124 |
| 333 | 9041234445 |

| CUS_CODE | CUS_LNAME | CUS_ZIP | CUSTOMER.AGENT_CODE | AGENT.AGENT_CODE | AGENT_PHONE |
|---|---|---|---|---|---|
| 1217782 | Adares | 32145 | 125 | 125 | 6152439887 |
| 1321242 | Rodriguez | 37134 | 125 | 125 | 6152439887 |
| 1312243 | Rakowski | 34129 | 167 | 167 | 6153426778 |
| 1132445 | Walker | 32145 | 231 | 231 | 6152431124 |
| 1657399 | Vanloo | 32145 | 231 | 231 | 6152431124 |
| 1542311 | Smithson | 37134 | 421 | | |

# Right Outer Join

- Yields all rows in the second relation including those that do not have a matching value in the first relation

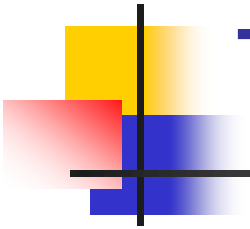
**Table name: CUSTOMER**

| CUS_CODE | CUS_LNAME | CUS_ZIP | AGENT_CODE |
|---|---|---|---|
| 1132445 | Walker | 32145 | 231 |
| 1217782 | Adares | 32145 | 125 |
| 1312243 | Rakowski | 34129 | 167 |
| 1321242 | Rodriguez | 37134 | 125 |
| 1542311 | Smithson | 37134 | 421 |
| 1657399 | Vanloo | 32145 | 231 |

**Table name: AGENT**

| AGENT_CODE | AGENT_PHONE |
|---|---|
| 125 | 6152439887 |
| 167 | 6153426778 |
| 231 | 6152431124 |
| 333 | 9041234445 |

| CUS_CODE | CUS_LNAME | CUS_ZIP | CUSTOMER.AGENT_CODE | AGENT.AGENT_CODE | AGENT_PHONE |
|---|---|---|---|---|---|
| 1217782 | Adares | 32145 | 125 | 125 | 6152439887 |
| 1321242 | Rodriguez | 37134 | 125 | 125 | 6152439887 |
| 1312243 | Rakowski | 34129 | 167 | 167 | 6153426778 |
| 1132445 | Walker | 32145 | 231 | 231 | 6152431124 |
| 1657399 | Vanloo | 32145 | 231 | 231 | 6152431124 |
| | | | | 333 | 9041234445 |



# To Do

- Assignments:
  - Exercise for Week 2
  - Review Questions for Week 2
  - Discussion Topic for Week 2
- Reading:
  - Review Chapters 2 and 3
  - Read Chapter 4