# Chapter 2: Intro to Relational Model

**Database System Concepts, 6th Ed.**

# Data Models

- A collection of tools/constructs for describing
  - Data
  - Data relationships
  - Data constraints

- Relational model
- Object-based data models (Object-oriented and Object-relational)
- Other older models:
  - Network model
  - Hierarchical model

# Basic Unit of the Relational Model

## Example of a Relation/Table:

attributes (or columns)

| *ID* | *name* | *dept_name* | *salary* |
|---|---|---|---|
| 10101 | Srinivasan | Comp. Sci. | 65000 |
| 12121 | Wu | Finance | 90000 |
| 15151 | Mozart | Music | 40000 |
| 22222 | Einstein | Physics | 95000 |
| 32343 | El Said | History | 60000 |
| 33456 | Gold | Physics | 87000 |
| 45565 | Katz | Comp. Sci. | 75000 |
| 58583 | Califieri | History | 62000 |
| 76543 | Singh | Finance | 80000 |
| 76766 | Crick | Biology | 72000 |
| 83821 | Brandt | Comp. Sci. | 92000 |
| 98345 | Kim | Elec. Eng. | 80000 |

tuples (or rows)

| course_id | title | dept_name | credits |
|---|---|---|---|
| BIO-101 | Intro. to Biology | Biology | 4 |
| BIO-301 | Genetics | Biology | 4 |
| BIO-399 | Computational Biology | Biology | 3 |
| CS-101 | Intro. to Computer Science | Comp. Sci. | 4 |
| CS-190 | Game Design | Comp. Sci. | 4 |
| CS-315 | Robotics | Comp. Sci. | 3 |
| CS-319 | Image Processing | Comp. Sci. | 3 |
| CS-347 | Database System Concepts | Comp. Sci. | 3 |
| EE-181 | Intro. to Digital Systems | Elec. Eng. | 3 |
| FIN-201 | Investment Banking | Finance | 3 |
| HIS-351 | World History | History | 3 |
| MU-199 | Music Video Production | Music | 3 |
| PHY-101 | Physical Principles | Physics | 4 |

| dept_name | building | budget |
|---|---|---|
| Biology | Watson | 90000 |
| Comp. Sci. | Taylor | 100000 |
| Elec. Eng. | Taylor | 85000 |
| Finance | Painter | 120000 |
| History | Painter | 50000 |
| Music | Packard | 80000 |
| Physics | Watson | 70000 |

| course_id | prereq_id |
|---|---|
| BIO-301 | BIO-101 |
| BIO-399 | BIO-101 |
| CS-190 | CS-101 |
| CS-315 | CS-101 |
| CS-319 | CS-101 |
| CS-347 | CS-101 |
| EE-181 | PHY-101 |

| course_id | sec_id | semester | year | building | room_number | time_slot_id |
|---|---|---|---|---|---|---|
| BIO-101 | 1 | Summer | 2009 | Painter | 514 | B |
| BIO-301 | 1 | Summer | 2010 | Painter | 514 | A |
| CS-101 | 1 | Fall | 2009 | Packard | 101 | H |
| CS-101 | 1 | Spring | 2010 | Packard | 101 | F |
| CS-190 | 1 | Spring | 2009 | Taylor | 3128 | E |
| CS-190 | 2 | Spring | 2009 | Taylor | 3128 | A |
| CS-315 | 1 | Spring | 2010 | Watson | 120 | D |
| CS-319 | 1 | Spring | 2010 | Watson | 100 | B |
| CS-319 | 2 | Spring | 2010 | Taylor | 3128 | C |
| CS-347 | 1 | Fall | 2009 | Taylor | 3128 | A |
| EE-181 | 1 | Spring | 2009 | Taylor | 3128 | C |
| FIN-201 | 1 | Spring | 2010 | Packard | 101 | B |
| HIS-351 | 1 | Spring | 2010 | Painter | 514 | C |
| MU-199 | 1 | Spring | 2010 | Packard | 101 | D |
| PHY-101 | 1 | Fall | 2009 | Watson | 100 | A |

| ID | course_id | sec_id | semester | year |
|---|---|---|---|---|
| 10101 | CS-101 | 1 | Fall | 2009 |
| 10101 | CS-315 | 1 | Spring | 2010 |
| 10101 | CS-347 | 1 | Fall | 2009 |
| 12121 | FIN-201 | 1 | Spring | 2010 |
| 15151 | MU-199 | 1 | Spring | 2010 |
| 22222 | PHY-101 | 1 | Fall | 2009 |
| 32343 | HIS-351 | 1 | Spring | 2010 |
| 45565 | CS-101 | 1 | Spring | 2010 |
| 45565 | CS-319 | 1 | Spring | 2010 |
| 76766 | BIO-101 | 1 | Summer | 2009 |
| 76766 | BIO-301 | 1 | Summer | 2010 |
| 83821 | CS-190 | 1 | Spring | 2009 |
| 83821 | CS-190 | 2 | Spring | 2009 |
| 83821 | CS-319 | 2 | Spring | 2010 |
| 98345 | EE-181 | 1 | Spring | 2009 |

# Table/Relation

- In general, a **row** in a table represents a relationship among a set of values.
- In the relational model the term **relation** is used to refer to **a table**, while the term **tuple** is used to refer to **a row**.
- Similarly, the term **attribute** refers to a **column of a table**.
- The term **relation instance** is used to refer to a specific instance of a relation, i.e., containing a specific set of rows.

# Attribute Types

- The set of permitted values for each attribute is called the **domain** of the attribute.
- Attribute values are (normally) required to be **atomic**; that is, indivisible
- The special value ***null*** is a member of every domain
- The null value causes complications in the definition of many operations

# Relation Schema and Instance

- The term **relation schema** is consists of attribute set and their domains
- whereas the **relation instance** corresponds to the values which the variable takes.

- A *relation schema syntax:* $\boldsymbol{R = (A_1 :D1, A_2 :D2, \ldots, A_n :Dn)}$

  Example:

*instructor* = (*ID int, name String, dept_name String, salary float*)

A database consists of multiple relations

# Relations are Unordered

- Order of tuples is irrelevant (tuples may be stored in an arbitrary order)
- Example: *instructor* relation with unordered tuples

| *ID* | *name* | *dept_name* | *salary* |
|---|---|---|---|
| 22222 | Einstein | Physics | 95000 |
| 12121 | Wu | Finance | 90000 |
| 32343 | El Said | History | 60000 |
| 45565 | Katz | Comp. Sci. | 75000 |
| 98345 | Kim | Elec. Eng. | 80000 |
| 76766 | Crick | Biology | 72000 |
| 10101 | Srinivasan | Comp. Sci. | 65000 |
| 58583 | Califieri | History | 62000 |
| 83821 | Brandt | Comp. Sci. | 92000 |
| 15151 | Mozart | Music | 40000 |
| 33456 | Gold | Physics | 87000 |
| 76543 | Singh | Finance | 80000 |

# Keys

- Candidate key
- Primary Key
- Super key  (Non minimal Key)
- Foreign Key

# Candidate and Primary Key

- The attributes ID and name together can distinguish instructor tuples, their combination, {ID, name} does not form a candidate key, since the attribute {ID} alone is a candidate key.
- One of the candidate keys is selected to be the **primary key**.
  - which one?
- Any two individual tuples in the relation are prohibited from having the same value on the key attributes at the same time.

- The primary key should be chosen such that its attribute values are **never, or very rarely changed and it should not have null value.**

- It is customary to list the primary key attributes of a relation schema before the other attributes.
- Primary key attributes are also underlined

# Foreign Key

- A relation, say *r*1, may include among its attributes the primary key of another relation, say *r*2. This attribute is called a **foreign key** from *r*1, referencing *r*2.

- **Referencing relation – r1**
- **Referenced relation – r2**

- For example, the attribute *dept_name* in *instructor* is a foreign key from *instructor*, referencing *department*.

- (c_id, sec_id, sem, year) together becomes the foreign key for teaches relation because it is the primary key in section relation.

# Example of a Database (with a Conceptual Data Model)

- **Mini-world for the example:**
  - Part of a UNIVERSITY environment.
- **Some mini-world *entities*:**
  - STUDENTs
  - COURSEs
  - SECTIONs (of COURSEs)
  - (academic) DEPARTMENTs
  - INSTRUCTORs

# Example of a Database (with a Conceptual Data Model)

- **Some mini-world *relationships*:**
  - SECTIONs *are of specific* COURSEs
  - STUDENTs *take* SECTIONs
  - COURSEs *have prerequisite* COURSEs
  - INSTRUCTORs *teach* SECTIONs
  - COURSEs *are offered by* DEPARTMENTs
  - STUDENTs *major in* DEPARTMENTs

- Note: The above entities and relationships are typically expressed in a conceptual data model, such as the ENTITY-RELATIONSHIP data model (see Chapters 3, 4)

# Example of a simple database

**COURSE**

| Course_name | Course_number | Credit_hours | Department |
|---|---|---|---|
| Intro to Computer Science | CS1310 | 4 | CS |
| Data Structures | CS3320 | 4 | CS |
| Discrete Mathematics | MATH2410 | 3 | MATH |
| Database | CS3380 | 3 | CS |

**SECTION**

| Section_identifier | Course_number | Semester | Year | Instructor |
|---|---|---|---|---|
| 85 | MATH2410 | Fall | 04 | King |
| 92 | CS1310 | Fall | 04 | Anderson |
| 102 | CS3320 | Spring | 05 | Knuth |
| 112 | MATH2410 | Fall | 05 | Chang |
| 119 | CS1310 | Fall | 05 | Anderson |
| 135 | CS3380 | Fall | 05 | Stone |

**GRADE_REPORT**

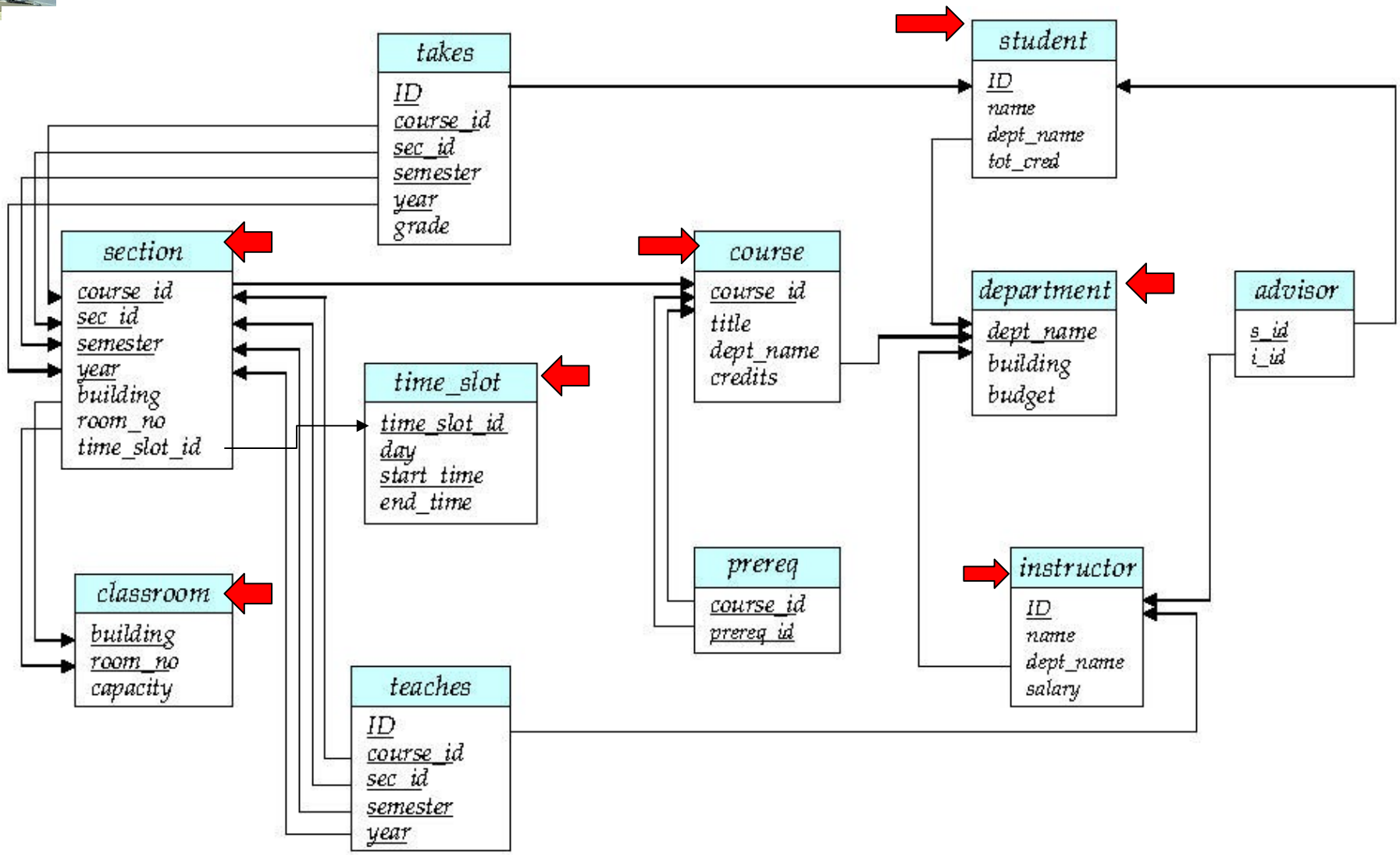
| Student_number | Section_identifier | Grade |
|---|---|---|
| 17 | 112 | B |
| 17 | 119 | C |
| 8 | 85 | A |
| 8 | 92 | A |
| 8 | 102 | B |
| 8 | 135 | A |

**PREREQUISITE**

| Course_number | Prerequisite_number |
|---|---|
| CS3380 | CS3320 |
| CS3380 | MATH2410 |
| CS3320 | CS1310 |

**Figure 1.2**
A database that stores student and course information.

# Schema Diagram for University Database



# Schema

- *classroom*(*building*, *room_number*, *capacity*)
- *department*(*dept_name*, *building*, *budget*)
- *course*(*course_id*, *title*, *dept_name*, *credits*)
- *instructor*(*ID*, *name*, *dept_name*, *salary*)
- *section*(*course_id*, *sec_id*, *semester*, *year*, *building*, *room_number*, *time_slot_id*)
- *teaches*(*ID*, *course_id*, *sec_id*, *semester*, *year*)
- *student*(*SID*, *name*, *dept_name*, *tot_cred*)
- *takes*(*SID*, *course_id*, *sec_id*, *semester*, *year*, *grade*)
- *advisor*(*sID*, *iID*)
- *Time_slot*(*time_slot_id*, *day*, *start_time*, *end_time*)
- *prereq*(*course_id*, *prereq_id*)

# Relational Query Languages

- A **query language** is a language in which a user requests information from the database.
- **Procedural:** The user instructs the system to perform a sequence of operations on the database to compute the desired result.

# Relational Operations

- The result of a query is a new relation that is a subset of the original relation

- Example, if we select tuples from the *instructor* relation satisfying the predicate "*salary* is greater than $85000"

| *ID* | *name* | *dept_name* | *salary* |
|---|---|---|---|
| 12121 | Wu | Finance | 90000 |
| 22222 | Einstein | Physics | 95000 |
| 33456 | Gold | Physics | 87000 |
| 83821 | Brandt | Comp. Sci. | 92000 |

# Selection of tuples

- Relation r

| A | B | C | D |
|---|---|---|---|
| α | α | 1 | 7 |
| α | β | 5 | 7 |
| β | β | 12 | 3 |
| β | β | 23 | 10 |

- Select tuples with A=B and D > 5
  - $\sigma_{A=B \text{ and } D > 5}(r)$

| A | B | C | D |
|---|---|---|---|
| α | α | 1 | 7 |
| β | β | 23 | 10 |

# Selection of Columns (Attributes)

- Relation *r*:

| A | B | C |
|---|---|---|
| α | 10 | 1 |
| α | 20 | 1 |
| β | 30 | 1 |
| β | 40 | 2 |

- Select A and C
  - Projection
  - $\Pi_{A, C}$ (r)

| A | C |
|---|---|
| α | 1 |
| α | 1 |
| β | 1 |
| β | 2 |

=

| A | C |
|---|---|
| α | 1 |
| β | 1 |
| β | 2 |

# Joining two relations – Cartesian Product

- Relations *r, s*:

| A | B |
|---|---|
| α | 1 |
| β | 2 |

*r*

| C | D | E |
|---|---|---|
| α | 10 | a |
| β | 10 | a |
| β | 20 | b |
| γ | 10 | b |

*s*

- *r* x *s*:

| A | B | C | D | E |
|---|---|---|---|---|
| α | 1 | α | 10 | a |
| α | 1 | β | 10 | a |
| α | 1 | β | 20 | b |
| α | 1 | γ | 10 | b |
| β | 2 | α | 10 | a |
| β | 2 | β | 10 | a |
| β | 2 | β | 20 | b |
| β | 2 | γ | 10 | b |

# Union of two relations

- Relations *r, s:*

| A | B |
|---|---|
| α | 1 |
| α | 2 |
| β | 1 |

*r*

| A | B |
|---|---|
| α | 2 |
| β | 3 |

*s*

- r ∪ s:

| A | B |
|---|---|
| α | 1 |
| α | 2 |
| β | 1 |
| β | 3 |

# Set difference of two relations

- Relations *r*, *s*:

| A | B |
|---|---|
| α | 1 |
| α | 2 |
| β | 1 |

*r*

| A | B |
|---|---|
| α | 2 |
| β | 3 |

*s*

- *r* – *s*:

| A | B |
|---|---|
| α | 1 |
| β | 1 |

# Set Intersection of two relations

- Relation *r, s*:

| *A* | *B* |
|---|---|
| α | 1 |
| α | 2 |
| β | 1 |

*r*

| *A* | *B* |
|---|---|
| α | 2 |
| β | 3 |

*s*

- $r \cap s$

| *A* | *B* |
|---|---|
| α | 2 |

# Joining two relations – Natural Join- Example

- The natural join operation on two relations matches tuples whose values are the same on **all attribute names that are common to both relations**.

- Relations r, s:

| A | B | C | D |
|---|---|---|---|
| α | 1 | α | a |
| β | 2 | γ | a |
| γ | 4 | β | b |
| α | 1 | γ | a |
| δ | 2 | β | b |

*r*

| B | D | E |
|---|---|---|
| 1 | a | α |
| 3 | a | β |
| 1 | a | γ |
| 2 | b | δ |
| 3 | b | ε |

*s*

- Natural Join
  - r ⋈ s

| A | B | C | D | E |
|---|---|---|---|---|
| α | 1 | α | a | α |
| α | 1 | α | a | γ |
| α | 1 | γ | a | α |
| α | 1 | γ | a | γ |
| δ | 2 | β | b | δ |

# Figure 2.06

| *course_id* | *sec_id* | *semester* | *year* | *building* | *room_number* | *time_slot_id* |
|---|---|---|---|---|---|---|
| BIO-101 | 1 | Summer | 2009 | Painter | 514 | B |
| BIO-301 | 1 | Summer | 2010 | Painter | 514 | A |
| CS-101 | 1 | Fall | 2009 | Packard | 101 | H |
| CS-101 | 1 | Spring | 2010 | Packard | 101 | F |
| CS-190 | 1 | Spring | 2009 | Taylor | 3128 | E |
| CS-190 | 2 | Spring | 2009 | Taylor | 3128 | A |
| CS-315 | 1 | Spring | 2010 | Watson | 120 | D |
| CS-319 | 1 | Spring | 2010 | Watson | 100 | B |
| CS-319 | 2 | Spring | 2010 | Taylor | 3128 | C |
| CS-347 | 1 | Fall | 2009 | Taylor | 3128 | A |
| EE-181 | 1 | Spring | 2009 | Taylor | 3128 | C |
| FIN-201 | 1 | Spring | 2010 | Packard | 101 | B |
| HIS-351 | 1 | Spring | 2010 | Painter | 514 | C |
| MU-199 | 1 | Spring | 2010 | Packard | 101 | D |
| PHY-101 | 1 | Fall | 2009 | Watson | 100 | A |

# Figure in-2.1

| Symbol (Name) | Example of Use |
|---|---|
| σ (Selection) | $\sigma_{\text{salary}>=85000}(instructor)$ |
| | Return rows of the input relation that satisfy the predicate. |
| Π (Projection) | $\Pi_{ID,\ salary}(instructor)$ |
| | Output specified attributes from all rows of the input relation. Remove duplicate tuples from the output. |
| ⋈ (Natural Join) | $instructor \bowtie department$ |
| | Output pairs of rows from the two input relations that have the same value on all attributes that have the same name. |
| × (Cartesian Product) | $instructor \times department$ |
| | Output all pairs of rows from the two input relations (regardless of whether or not they have the same values on common attributes) |
| ∪ (Union) | $\Pi_{name}(instructor) \cup \Pi_{name}(student)$ |
| | Output the union of tuples from the two input relations. |

# End of Chapter 2

# Figure 2.01

| *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 |

# Figure 2.02

| *course_id* | *title* | *dept_name* | *credits* |
|---|---|---|---|
| BIO-101 | Intro. to Biology | Biology | 4 |
| BIO-301 | Genetics | Biology | 4 |
| BIO-399 | Computational Biology | Biology | 3 |
| CS-101 | Intro. to Computer Science | Comp. Sci. | 4 |
| CS-190 | Game Design | Comp. Sci. | 4 |
| CS-315 | Robotics | Comp. Sci. | 3 |
| CS-319 | Image Processing | Comp. Sci. | 3 |
| CS-347 | Database System Concepts | Comp. Sci. | 3 |
| EE-181 | Intro. to Digital Systems | Elec. Eng. | 3 |
| FIN-201 | Investment Banking | Finance | 3 |
| HIS-351 | World History | History | 3 |
| MU-199 | Music Video Production | Music | 3 |
| PHY-101 | Physical Principles | Physics | 4 |

# Figure 2.03

| *course_id* | *prereq_id* |
|---|---|
| BIO-301 | BIO-101 |
| BIO-399 | BIO-101 |
| CS-190 | CS-101 |
| CS-315 | CS-101 |
| CS-319 | CS-101 |
| CS-347 | CS-101 |
| EE-181 | PHY-101 |

# Figure 2.04

| *ID* | *name* | *dept_name* | *salary* |
|---|---|---|---|
| 22222 | Einstein | Physics | 95000 |
| 12121 | Wu | Finance | 90000 |
| 32343 | El Said | History | 60000 |
| 45565 | Katz | Comp. Sci. | 75000 |
| 98345 | Kim | Elec. Eng. | 80000 |
| 76766 | Crick | Biology | 72000 |
| 10101 | Srinivasan | Comp. Sci. | 65000 |
| 58583 | Califieri | History | 62000 |
| 83821 | Brandt | Comp. Sci. | 92000 |
| 15151 | Mozart | Music | 40000 |
| 33456 | Gold | Physics | 87000 |
| 76543 | Singh | Finance | 80000 |

# Figure 2.05

| *dept_name* | *building* | *budget* |
|---|---|---|
| Biology | Watson | 90000 |
| Comp. Sci. | Taylor | 100000 |
| Elec. Eng. | Taylor | 85000 |
| Finance | Painter | 120000 |
| History | Painter | 50000 |
| Music | Packard | 80000 |
| Physics | Watson | 70000 |

# Figure 2.07

| *ID* | *course_id* | *sec_id* | *semester* | *year* |
|---|---|---|---|---|
| 10101 | CS-101 | 1 | Fall | 2009 |
| 10101 | CS-315 | 1 | Spring | 2010 |
| 10101 | CS-347 | 1 | Fall | 2009 |
| 12121 | FIN-201 | 1 | Spring | 2010 |
| 15151 | MU-199 | 1 | Spring | 2010 |
| 22222 | PHY-101 | 1 | Fall | 2009 |
| 32343 | HIS-351 | 1 | Spring | 2010 |
| 45565 | CS-101 | 1 | Spring | 2010 |
| 45565 | CS-319 | 1 | Spring | 2010 |
| 76766 | BIO-101 | 1 | Summer | 2009 |
| 76766 | BIO-301 | 1 | Summer | 2010 |
| 83821 | CS-190 | 1 | Spring | 2009 |
| 83821 | CS-190 | 2 | Spring | 2009 |
| 83821 | CS-319 | 2 | Spring | 2010 |
| 98345 | EE-181 | 1 | Spring | 2009 |

# Figure 2.10

| *ID* | *name* | *dept_name* | *salary* |
|---|---|---|---|
| 12121 | Wu | Finance | 90000 |
| 22222 | Einstein | Physics | 95000 |
| 33456 | Gold | Physics | 87000 |
| 83821 | Brandt | Comp. Sci. | 92000 |

# Figure 2.11

| *ID* | *salary* |
|---|---|
| 10101 | 65000 |
| 12121 | 90000 |
| 15151 | 40000 |
| 22222 | 95000 |
| 32343 | 60000 |
| 33456 | 87000 |
| 45565 | 75000 |
| 58583 | 62000 |
| 76543 | 80000 |
| 76766 | 72000 |
| 83821 | 92000 |
| 98345 | 80000 |

# Figure 2.12

| *ID* | *name* | *salary* | *dept_name* | *building* | *budget* |
|---|---|---|---|---|---|
| 10101 | Srinivasan | 65000 | Comp. Sci. | Taylor | 100000 |
| 12121 | Wu | 90000 | Finance | Painter | 120000 |
| 15151 | Mozart | 40000 | Music | Packard | 80000 |
| 22222 | Einstein | 95000 | Physics | Watson | 70000 |
| 32343 | El Said | 60000 | History | Painter | 50000 |
| 33456 | Gold | 87000 | Physics | Watson | 70000 |
| 45565 | Katz | 75000 | Comp. Sci. | Taylor | 100000 |
| 58583 | Califieri | 62000 | History | Painter | 50000 |
| 76543 | Singh | 80000 | Finance | Painter | 120000 |
| 76766 | Crick | 72000 | Biology | Watson | 90000 |
| 83821 | Brandt | 92000 | Comp. Sci. | Taylor | 100000 |
| 98345 | Kim | 80000 | Elec. Eng. | Taylor | 85000 |

# Figure 2.13

| *ID* | *salary* |
|---|---|
| 12121 | 90000 |
| 22222 | 95000 |
| 33456 | 87000 |
| 83821 | 92000 |