COP-5725
MIDTERM REVIEW

hapters 1 – 5, 19

M. Amanda Crick
(Uses slides from Fernando Farfan and Eduardo
J. Ruiz

Chapter 1: Overview of DBMSs

Concepts:

- DBMS
- Relational Model
- Levels of Abstraction
- Data Independence

Exercise 1.1

Problem

- Why would you choose a database system instead of simply storing data in operating system files? When would it make sense *not to use a database system?*

Exercise 1.1

Solution

- *Data independence and efficient access.*
  - Physical, logical independence
  - Efficient storage and data retrieval
- *Reduced application development time.*
  - Data storage aspect of application already written and debugged; only need to write application code
- *Data integrity and security.*
  - Database prevents changes that violate integrity constraints. Views and authorization mechanism.

Exercise 1.1

Solution

- *Data administration.*
  - Maintenance and data administration made easier.
- *Concurrent access and crash recovery*
  - Transactions prevent two conflicting operations from being carried out concurrently.
  - Keeps a log of changes to data, so that the system can recover from a crash.

Exercise 1.4

Problem

- Explain the difference between external, internal, and conceptual schemas. How are these different schema layers related to the concepts of logical and physical data independence?

Exercise 1.4

Solution

- *External schemas:*
  - Allow data access to be customized at the level of individual users or groups of users using different ***VIEWS*** of the same conceptual schema.
  - Views are not stored in DBMS but they generated on-demand.
- *Conceptual (logical) schemas:*
  - Describes all the data in terms of the data model. In a relational DBMS, it describes all relations stored.
  - While there are several views for a given database, there is exactly one conceptual schema

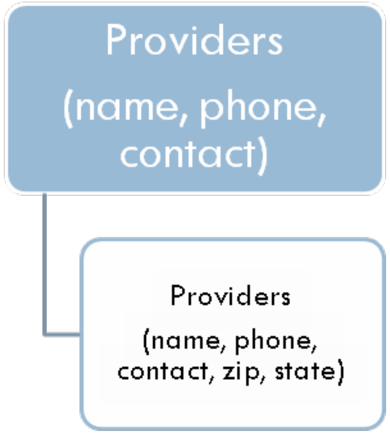
Exercise 1.4

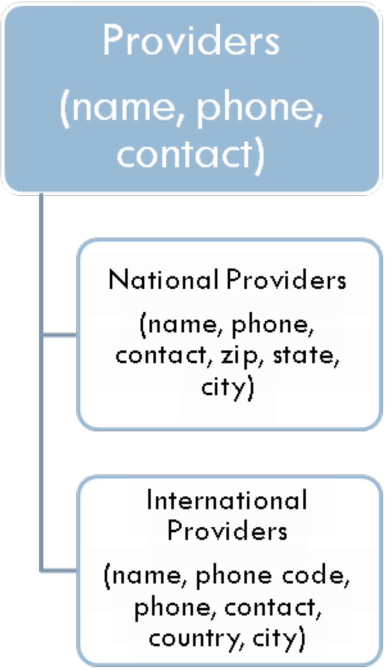
Solution

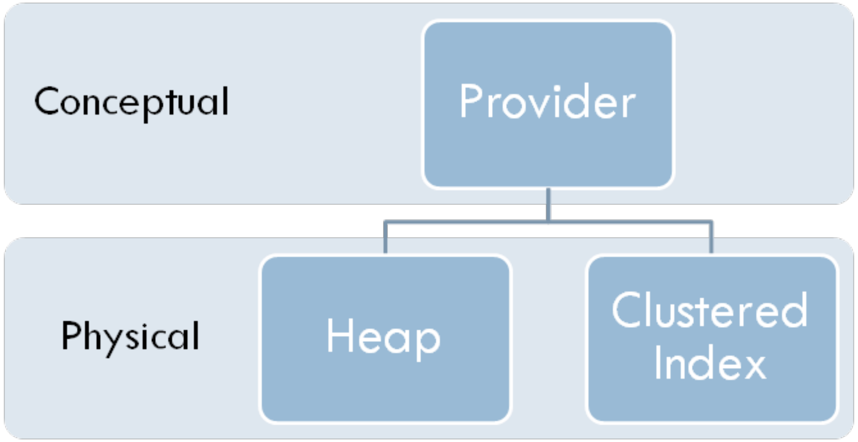
- *Internal (physical) schemas:*
  - Describes how the relations described in the conceptual schema are actually stored on disk (or other physical media).

Exercise 1.4

Solution







Exercise 1.4

Solution

- The *logical schema* protects outside programs and users from changes to the database relational schema.
- The *physical schema* protects programs and users from changes to the way database files are stored.

Chapter 2: Database Design

Concepts:

- Domain
- Attribute
- Entity (Set)
- Relationship(Set)
- Primary Key
- Participation Constraint
- Key Constraint
- Aggregation
- Overlap Constraint
- Descriptive Attribute
- Roles
- One-to-Many
- Many-to-May
- Weak Entity Set
- Identifying

Exercise 2.2

Problem

- A university database contains information about professors (identified by social security number, or SSN) and courses (identified by courseid). Professors teach courses; each of the following situations concerns the Teaches relationship set. For each situation, draw an ER diagram that describes it (assuming no further constraints hold). Draw an ER diagram that captures this information.

Exercise 2.2

Problem

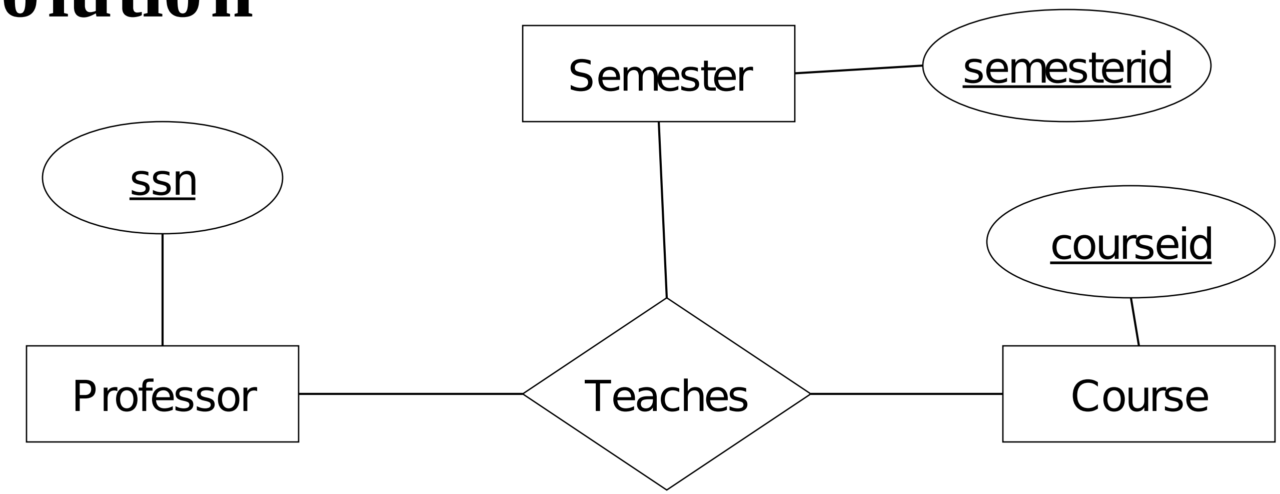
- A university database contains information about professors (identified by social security number, or SSN) and courses (identified by courseid). Professors teach courses; each of the following situations concerns the Teaches relationship set. For each situation, draw an ER diagram that describes it (assuming no further constraints hold). Draw an ER diagram that captures this information.

Exercise 2.2

Problem

1. Professors can teach the same course in several semesters, and each offering must be recorded.

Solution

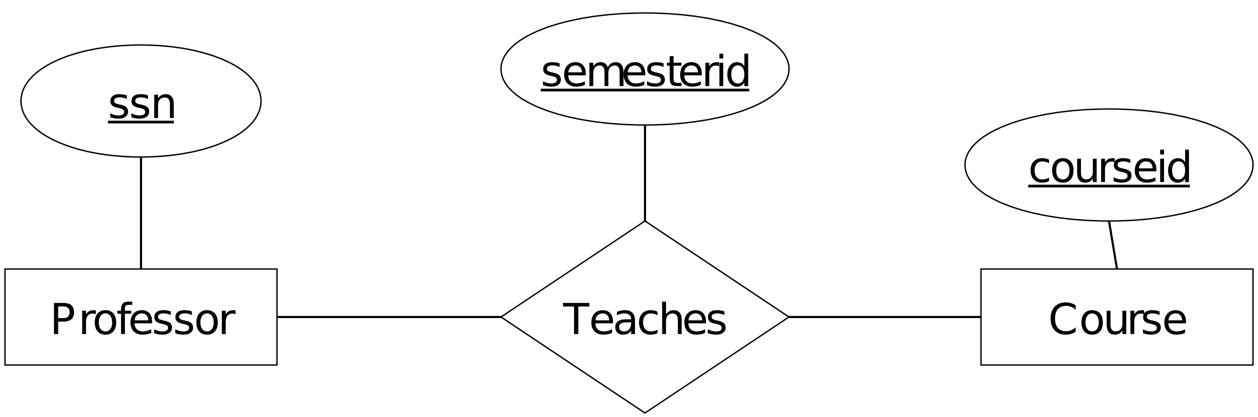


Exercise 2.2

Problem

2. Professors can teach the same course in several semesters, and only the most recent such offering needs to be recorded. (Assume this condition applies in all subsequent questions.)

Solution

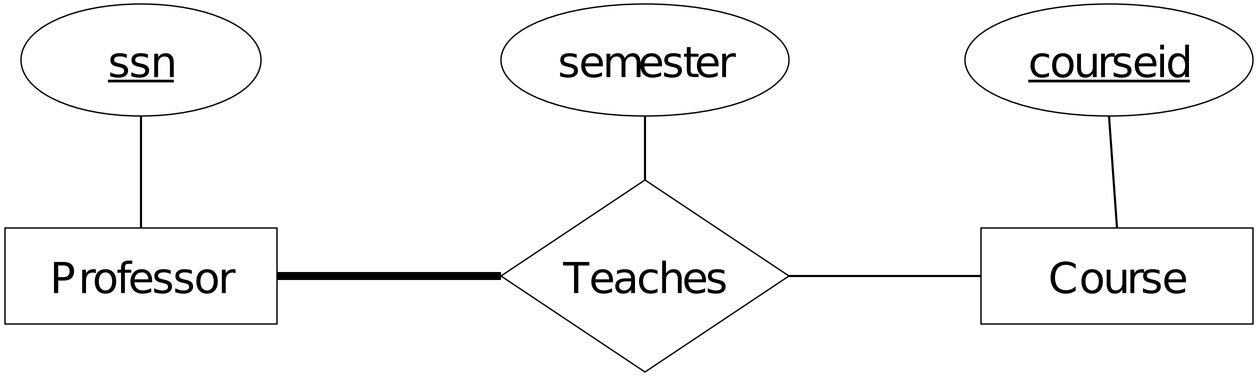


Exercise 2.2

Problem

3. Every professor must teach some course.

Solution

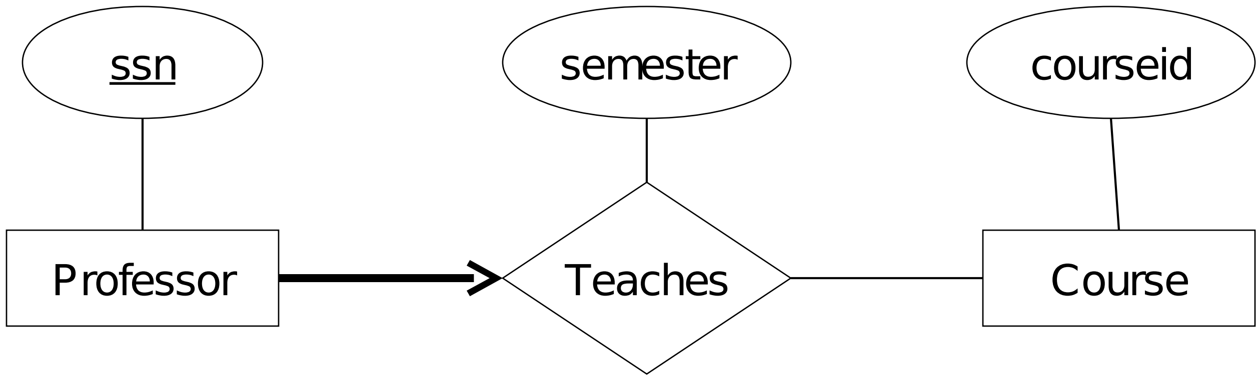


Exercise 2.2

Problem

4. Every professor teaches exactly one course (no more, no less).

Solution

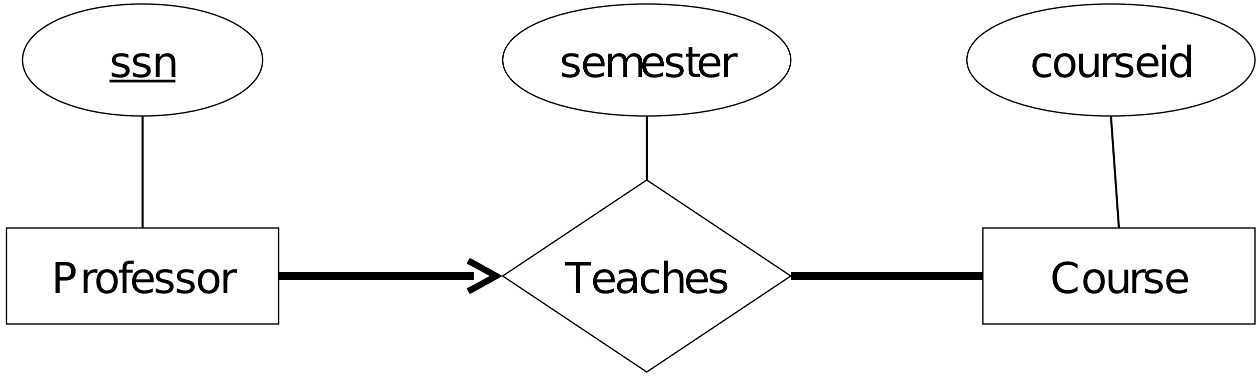


Exercise 2.2

Problem

5. Every professor teaches exactly one course (no more, no less), and every course must be taught by some professor.

Solution



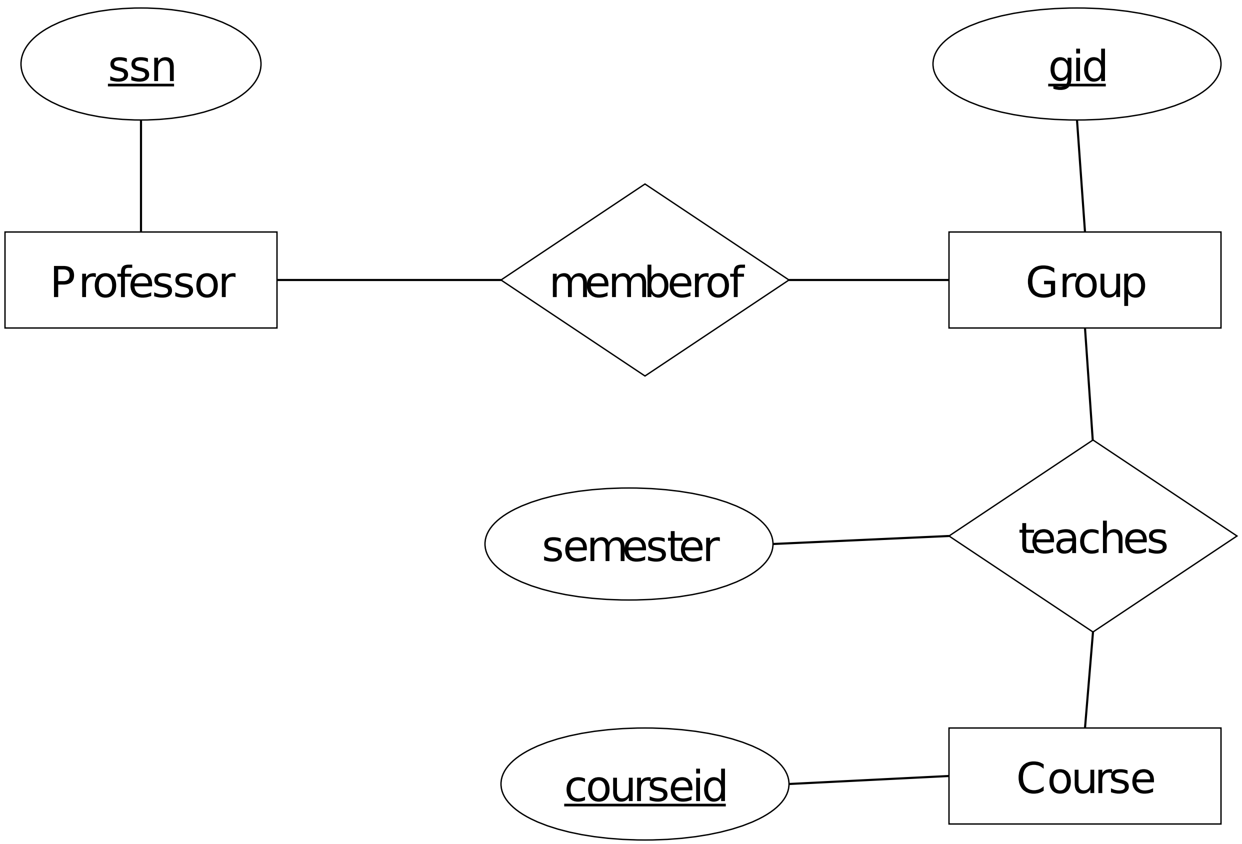
Exercise 2.2

Problem

6. Now suppose that certain courses can be taught by a team of professors jointly, but it is possible that no one professor in a team can teach the course. Model this situation, introducing additional entity sets and relationship sets if necessary.

Exercise 2.2

Solution



Chapter 3: Relational Model

Concepts:

- Table/Relation
- Relation Schema
- Attributes/Domain
- Relation Instance
- Tuple/Records
- Cardinality
- DDL
- Primary Key
- Superkey
- Candidate Key
- Foreign Key

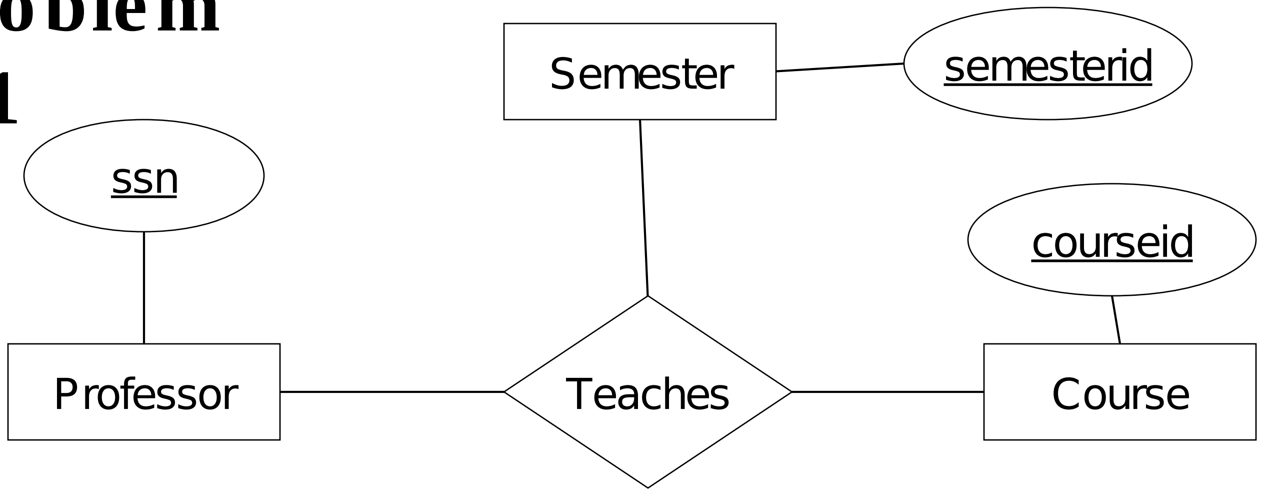
Exercise 3.12

Problem

- Consider the scenario from Exercise 2.2, where you designed an ER diagram for a university database. Write SQL statements to create the corresponding relations and capture as many of the constraints as possible. If you cannot capture some constraints, explain why.

Exercise 3.12

Problem 1



Exercise 3.12

Solution to (1)

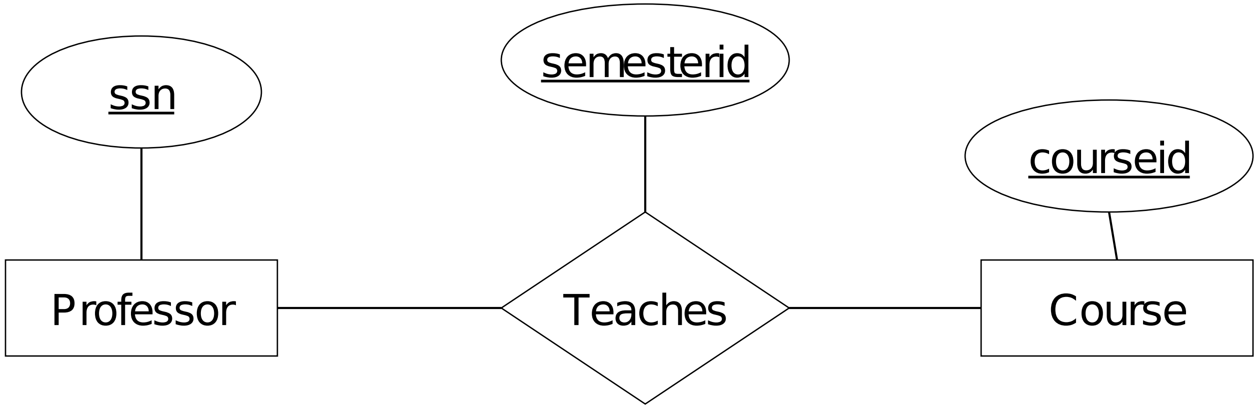
```
CREATE TABLE Teaches
( ssn CHAR(10),
  courseId INTEGER,
  semester CHAR(10),
  PRIMARY KEY (ssn, courseId, semester),
  FOREIGN KEY (ssn) REFERENCES
Professor,
  FOREIGN KEY (courseId) REFERENCES
Course )
  FOREIGN KEY (semester) REFERENCES
Semester)
```

Since all of the entity table can be created similarly, the definition for Course is given below.

```
CREATE TABLE Course ( courseId INTEGER,
                      PRIMARY KEY (courseId) )
```

Exercise 3.12

Problem 2



Exercise 3.12

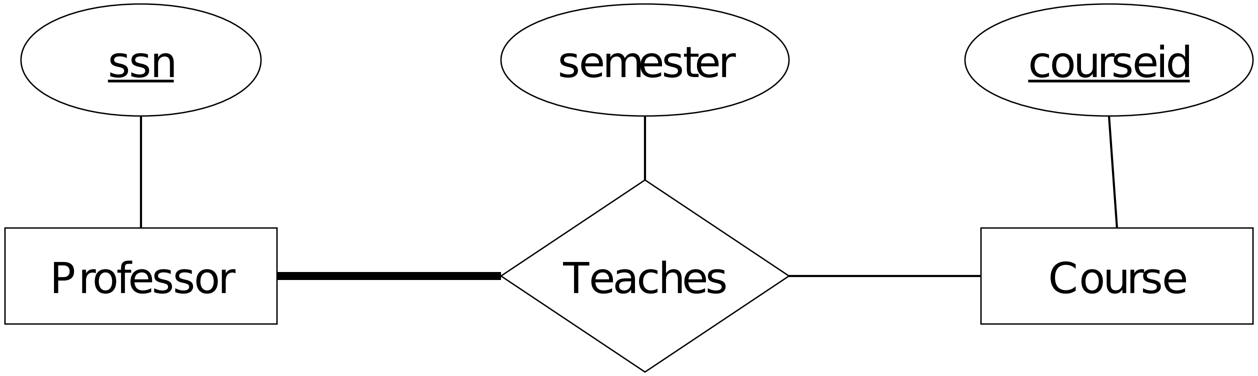
Solution to (2)

```
CREATE TABLE Teaches
 ( ssn CHAR(10),
   courseId INTEGER,
   semester CHAR(10),
   PRIMARY KEY (ssn, courseId),
   FOREIGN KEY (ssn) REFERENCES Professor,
   FOREIGN KEY (courseId) REFERENCES Course )
```

Professor and Course can be created as they were in the solution to (1).

Exercise 3.12

Problem 3



Exercise 3.12

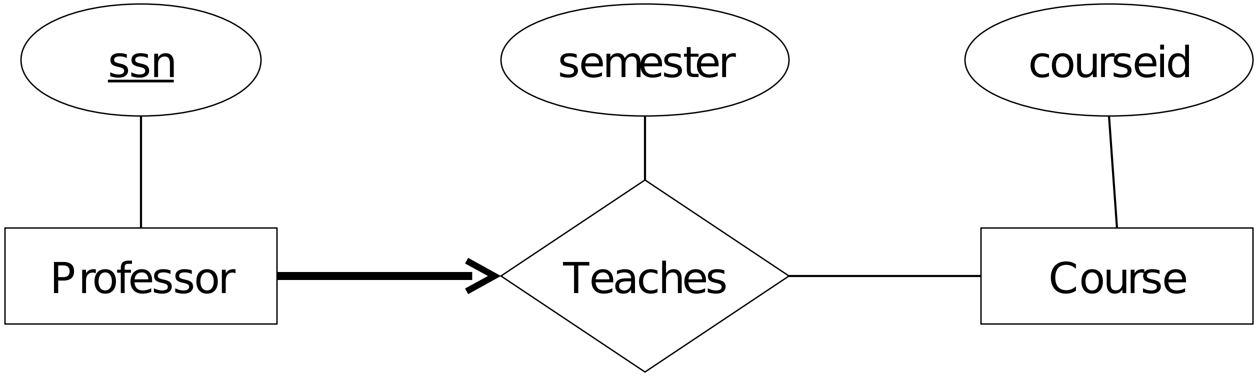
Solution to (3)

The answer to (2) is the closest answer that can be expressed for this section.

Without using assertions or check constraints, the total participation constraint between Professor and Teaches cannot be expressed.

Exercise 3.12

Problem 4



Exercise 3.12

Solution to (4)

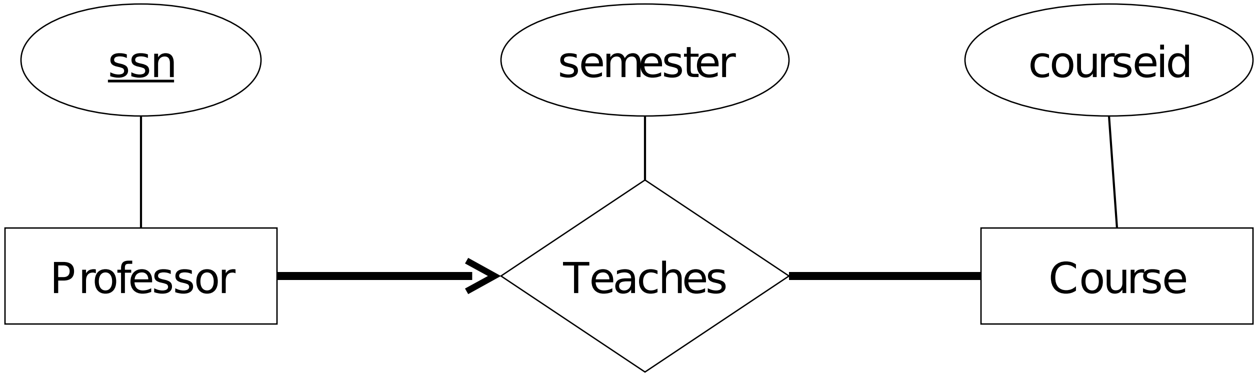
```
CREATE TABLE Professor_ teaches
( ssn CHAR(10),
  courseId INTEGER,
  semester CHAR(10),
  PRIMARY KEY (ssn),
  FOREIGN KEY (courseId)
  REFERENCES Course )

CREATE TABLE Course ( courseId INTEGER,
                      PRIMARY KEY (courseId) )
```

Since Professor and Teacher have been combined into one table, a separate table is not needed for Professor.

Exercise 3.12

Problem 5



Exercise 3.12

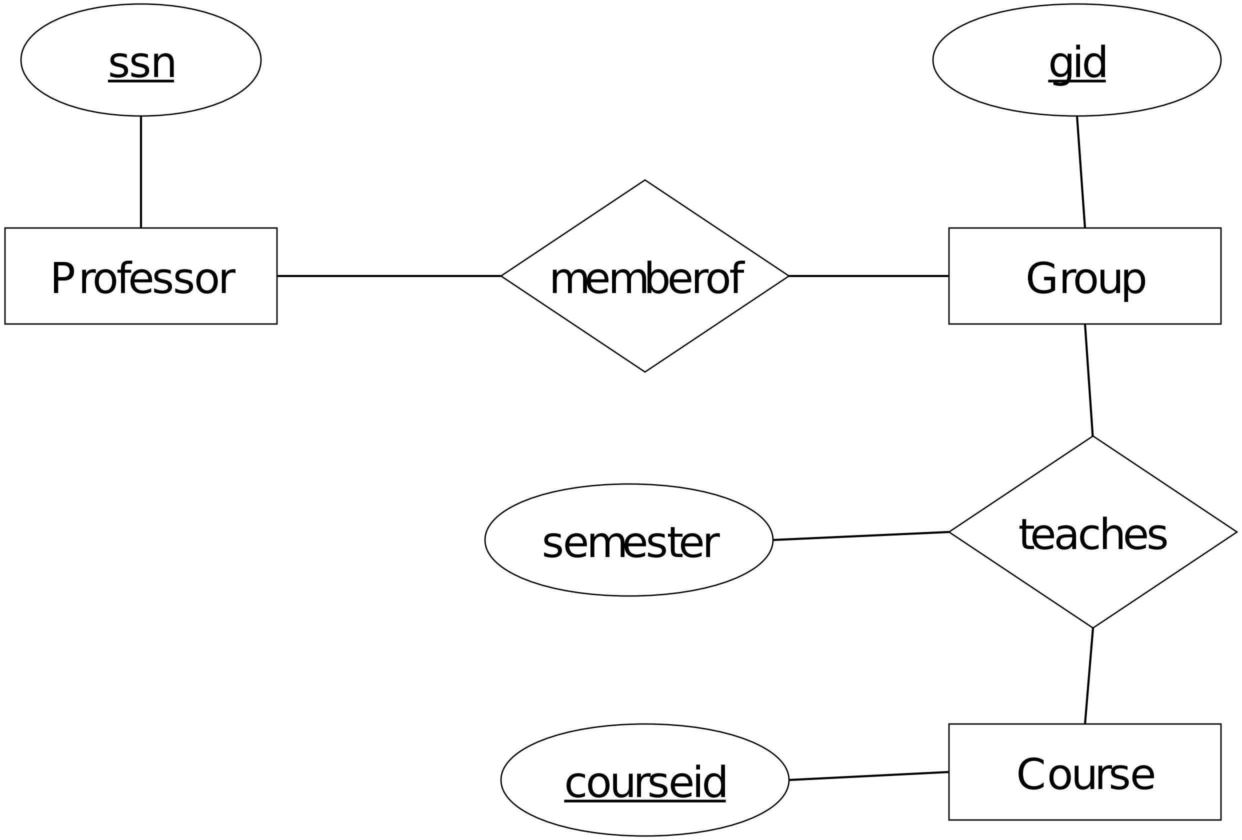
Solution to (5)

```
CREATE TABLE Professor_teaches
( ssn CHAR(10),
  courseId INTEGER,
  semester CHAR(10),
  PRIMARY KEY (ssn),
  FOREIGN KEY (courseId)
  REFERENCES Course )
```

Since the course table has only one attribute and total participation, it is combined with the Professor_teaches table.

Exercise 3.12

Solution



Exercise 3.12

Solution to (6)

```
CREATE TABLE Teaches
( gid INTEGER,
  courseId INTEGER,
  semester CHAR(10),
  PRIMARY KEY (gid, courseId),
  FOREIGN KEY (gid) REFERENCES Group,
  FOREIGN KEY (courseId) REFERENCES Course )

CREATE TABLE MemberOf
( ssn CHAR(10),
  gid INTEGER,
  PRIMARY KEY (ssn, gid),
  FOREIGN KEY (ssn) REFERENCES Professor,
  FOREIGN KEY (gid) REFERENCES Group )
```

Exercise 3.12

Solution to (6)

```
CREATE TABLE Course ( courseId INTEGER,
                     PRIMARY KEY (courseId) )

CREATE TABLE Group ( gid INTEGER,
                     PRIMARY KEY (gid) )

CREATE TABLE Professor ( ssn CHAR(10),
                     PRIMARY KEY (ssn) )
```

Chapter 4: Relational Algebra and Calculus

Concepts

- Selection
- Projection
- Join

Exercise 4.2

Problem

- Given two relations $R1$ and $R2$, where $R1$ contains N1 tuples, $R2$ contains N2 tuples, and N2 $>$ N1 $>$ 0, give the min and max possible sizes for the resulting relational algebra expressions:

Exercise 4.2

Solution

Expression	Assumption	Min	Max
$R1 \cup R2$	$R1$ and $R2$ are union-compatible	N2	N1 + N2
$R1 \cap R2$	$R1$ and $R2$ are union-compatible	0	N1
$R1 - R2$	$R1$ and $R2$ are union-compatible	0	N1
$R1 \times R2$		N1 * N2	N1 * N2
$\sigma_{a=5}(R1)$	$R1$ has an attribute named a	0	N1
$\pi_a(R1)$	$R1$ has attribute a, N1>0	1	N1

Exercise 4.4

Problem

- Consider the Supplier-Parts-Catalog schema. State what the following queries compute:

Suppliers(<u>*sid:* `integer`</u>, *sname:* `string`, *address:* `string`)
Parts(<u>*pid:* `integer`</u>, *pname:* `string`, *color:* `string`)
Catalog(<u>*sid:* `integer`, *pid:* `integer`</u>, *cost:* `real`)

Exercise 4.4

Suppliers(<u>*sid:* integer</u>, *sname:* string, *address:* string)
Parts(<u>*pid:* integer</u>, *pname:* string, *color:* string)
Catalog(<u>*sid:* integer, *pid:* integer</u>, *cost:* real)

Problem

1. Find the Supplier names of the suppliers who supply a red part that costs less than 100 dollars.

Solution

1. $\pi_{sname}(\pi_{sid}((\sigma_{color='red'}Parts) \bowtie (\sigma_{cost<100}Catalog)) \bowtie Suppliers)$

Exercise 4.4

Suppliers(*sid*: integer, *sname*: string, *address*: string)
Parts(*pid*: integer, *pname*: string, *color*: string)
Catalog(*sid*: integer, *pid*: integer, *cost*: real)

Problem

2. This Relational Algebra statement does not return anything because of the sequence of projection operators. Once the sid is projected, it is the only field in the set. Therefore, projecting on sname will not return anything.

Solution

2. $\pi_{sname}(\pi_{sid}((\sigma_{color='red'} Parts) \bowtie (\sigma_{cost<100} Catalog) \bowtie Suppliers))$

Exercise 4.4

Suppliers(<u>*sid:* `integer`</u>, *sname:* `string`, *address:* `string`)
Parts(<u>*pid:* `integer`</u>, *pname:* `string`, *color:* `string`)
Catalog(<u>*sid:* `integer`, *pid:* `integer`</u>, *cost:* `real`)

Problem

3. Find the Supplier names of the suppliers who supply a red part that costs less than 100 dollars and a green part that costs less than 100 dollars.

Solution

3. $(\pi_{sname}((\sigma_{color='red'}Parts) \bowtie (\sigma_{cost<100}Catalog) \bowtie Suppliers)) \cap$
$(\pi_{sname}((\sigma_{color='green'}Parts) \bowtie (\sigma_{cost<100}Catalog) \bowtie Suppliers))$

Exercise 4.4

Suppliers(<u>*sid:* integer</u>, *sname:* string, *address:* string)
Parts(<u>*pid:* integer</u>, *pname:* string, *color:* string)
Catalog(<u>*sid:* integer, *pid:* integer</u>, *cost:* real)

Problem

4. Find the Supplier ids of the suppliers who supply a red part that costs less than 100 dollars and a green part that costs less than 100 dollars.

Solution

4. $(\pi_{sid}((\sigma_{color='red'} Parts) \bowtie (\sigma_{cost<100} Catalog) \bowtie Suppliers)) \cap$
$(\pi_{sid}((\sigma_{color='green'} Parts) \bowtie (\sigma_{cost<100} Catalog) \bowtie Suppliers))$

Exercise 4.4

Suppliers(<u>*sid:* integer</u>, *sname:* string, *address:* string)
Parts(<u>*pid:* integer</u>, *pname:* string, *color:* string)
Catalog(<u>*sid:* integer, *pid:* integer</u>, *cost:* real)

Problem

5. Find the Supplier names of the suppliers who supply a red part that costs less than 100 dollars and a green part that costs less than 100 dollars.

Solution

5. $\pi_{sname}((\pi_{sid,sname}((\sigma_{color='red'}Parts) \bowtie (\sigma_{cost<100}Catalog) \bowtie Suppliers)) \cap (\pi_{sid,sname}((\sigma_{color='green'}Parts) \bowtie (\sigma_{cost<100}Catalog) \bowtie Suppliers)))$

Chapter 5: SQL, Null Values, Views

Concepts:

- DML
- DDL
- Query
- Nested Query
- Aggregation

Exercise 5.2

Problem

- Consider the following relational schema:
  - Suppliers(*<u>sid: integer</u>, sname: string, address: string)*
  - Parts(*<u>pid: integer</u>, pname: string, color: string)*
  - Catalog(*<u>sid: integer, pid: integer</u>, cost: real)*
- The Catalog relation lists the prices charged for parts by Suppliers. Write the following queries in SQL:

Exercise 5.2

Problem

Suppliers(*<u>sid: integer</u>, sname: string, address: string)*

Parts(*<u>pid: integer</u>, pname: string, color: string)*

Catalog(*<u>sid: integer, pid: integer</u>, cost: real)*

10. For every supplier that only supplies green parts, print the name of the supplier and the total number of parts that she supplies.

Exercise 5.2

Solution for (10)

```
SELECT S.sname, COUNT(*) as PartCount
FROM Suppliers S, Parts P, Catalog C
WHERE P.pid = C.pid AND C.sid = S.sid
GROUP BY S.sname, S.sid
HAVING EVERY (P.color='Green')
```

Exercise 5.2

Problem

Suppliers(*sid: integer*, *sname: string, address: string*)

Parts(*pid: integer*, *pname: string, color: string*)

Catalog(*sid: integer, pid: integer*, *cost: real*)

11. For every supplier that supplies a green part and a red part, print the name and price of the most expensive part that she supplies.

Exercise 5.2

Solution for (11)

```
SELECT S.sname, MAX(C.cost) as MaxCost
FROM Suppliers S, Parts P, Catalog C
WHERE P.pid = C.pid AND C.sid = S.sid
GROUP BY S.sname, S.sid
HAVING ANY ( P.color='green' ) AND ANY ( P.color = 'red' )
```

Exercise 5.4

Problem

- Consider the following relational schema. An employee can work in more than one department; the *pct_time field of the Works relation shows the percentage* of time that a given employee works in a given department.
  - Emp(*<u>eid: integer</u>, ename: string, age: integer, salary: real)*
  - Works(*<u>eid: integer, did: integer</u>, pct_time: integer)*
  - Dept(*<u>did: integer</u>, dname: string, budget: real, managerid: integer)*
- Write the following queries in SQL:

Exercise 5.4

Problem

Emp(*eid: integer*, *ename: string, age: integer, salary: real*)

Works(*eid: integer, did: integer*, *pct_time: integer*)

Dept(*did: integer*, *dname: string, budget: real, managerid: integer*)

6. If a manager manages more than one department, he or she *controls the sum of all* the budgets for those departments. Find the *managerids of managers who control* more than $5 million.

Exercise 5.4

Solution for (6)

```
SELECT D.managerid
FROM Dept D
WHERE 5000000 < (SELECT SUM (D2.budget)
                         FROM Dept D2
                         WHERE D2.managerid = D.managerid )
```

Exercise 5.4

Problem

Emp(*<u>eid: integer</u>, ename: string, age: integer, salary: real*)

Works(*<u>eid: integer, did: integer</u>, pct_time: integer*)

Dept(*<u>did: integer</u>, dname: string, budget: real, managerid: integer*)

7. Find the *managerids of managers who control the largest amounts.*

Exercise 5.4

Solution for (7)

```
SELECT DISTINCT tempD.managerid
FROM (SELECT DISTINCT D.managerid,
           SUM (D.budget) AS tempBudget
           FROM Dept D
           GROUP BY D.managerid ) AS tempD
WHERE tempD.tempBudget = (SELECT MAX (tempD.tempBudget)
                                   FROM tempD)
```

Chapter 19: Normal Forms

Concepts:

- Redundancy
- Functional Dependency
- BCNF
- 3NF

Exercise 19.2

Problem

- Consider a relation R *with five attributes ABCDE. You are given the* following dependencies: $A \rightarrow B$, $BC \rightarrow E$, *and* $ED \rightarrow A$.

1. List all keys for R

Solution

- CDE, ACD, BCD

Exercise 19.2

$A \rightarrow B, BC \rightarrow E, and\ ED \rightarrow A.$

Problem

2. Is R in 3NF?

Solution

- R is in 3NF because B, E and A are all parts of keys.

Exercise 19.2

$A \rightarrow B, BC \rightarrow E, and\ ED \rightarrow A.$

Problem

3. Is R in BCNF?

Solution

- R is not in BCNF because none of A, BC and ED contain a key.

Exercise 19.8

Problem 1

- Consider the attribute set $R = ABCDEGH$ and $the\ FD\ set\ F =$
  - $\{AB \rightarrow C,$
  - $AC \rightarrow B,$
  - $AD \rightarrow E,$
  - $B \rightarrow D,$
  - $BC \rightarrow A,$
  - $E \rightarrow G\}.$

Exercise 19.8

Problem 1

- For each of the following attribute sets, do the following:
  - (i) Compute the set of dependencies that hold over the set and write down a minimal cover.
  - (ii) Name the strongest normal form that is not violated by the relation containing these attributes.
  - (iii) Decompose it into a collection of BCNF relations if it is not in BCNF.

Exercise 19.2

$F = \{AB \rightarrow C, AC \rightarrow B, AD \rightarrow E, B \rightarrow D, BC \rightarrow A, E \rightarrow G\}$.

Problem

a) ABC

Solution

i. *R1 = ABC: The FD's are AB → C, AC → B, BC → A.*

ii. This is already a minimal cover.

iii. This is in BCNF since *AB, AC and BC are candidate keys for R1. (In* fact, these are all the candidate keys for *R1).*

Exercise 19.2

$F = \{AB \rightarrow C, AC \rightarrow B, AD \rightarrow E, B \rightarrow D, BC \rightarrow A, E \rightarrow G\}.$

Problem

b) ABCD

Solution

i. $R2 = ABCD$: *The FD's are* $AB \rightarrow C, AC \rightarrow B, B \rightarrow D, BC \rightarrow A.$

ii. This is already a minimal cover.

iii. The keys are: *AB, AC, BC. R2 is not in BCNF or even 2NF because of* the FD, $B \rightarrow D$ *(B is a proper subset of a key!) However, it is in 1NF.* Decompose as in: *ABC, BD. This is a BCNF decomposition*

This is the end of the lecture!

I hope you enjoyed it.