

# CSC7072: Databases, fall 2015

Dr. Kim Bauters



relational model

# relation model

## relational model in detail

The diagram shows a table representing a relation. The table has four columns: id, name, dept\_name, and salary. The word 'instructor' is written in a box above the table, with an arrow pointing to the table header. The word 'unique table name' is written above the 'instructor' box. The word 'attribute' is written above the table, with four arrows pointing to each of the four columns: id, name, dept\_name, and salary.

| <u>id</u> | name       | dept_name  | salary |
|-----------|------------|------------|--------|
| 10101     | Srinivasan | Comp. Sci. | 65000  |
| 12121     | Wu         | Finance    | 90000  |
| 15151     | Mozart     | Music      | 40000  |
| 22222     | Einstein   | Physics    | 95000  |

each *named* attribute is:

- *atomic*: cannot be split in smaller data  
*for example*: it is not a set of telephone numbers
- taking its values from a *domain* – *which always contains null*  
*for example*: for gender we only allow M or F

# relation model

## relational model in detail

The diagram shows a table with four columns: id, name, dept\_name, and salary. The table is titled 'instructor'. Annotations include: 'unique table name' pointing to the table name, and 'attribute' pointing to each of the four columns. The table contains four rows of data.

| <u>id</u> | name       | dept_name  | salary |
|-----------|------------|------------|--------|
| 10101     | Srinivasan | Comp. Sci. | 65000  |
| 12121     | Wu         | Finance    | 90000  |
| 15151     | Mozart     | Music      | 40000  |
| 22222     | Einstein   | Physics    | 95000  |

changes rarely

a *relation schema* defines the attributes of our relation:

instructor(id, gender, name, salary)

specified by Data Definition Language (DDL)

# relation model

## relational model in detail



unique table name

attribute

| <u>id</u> | name       | dept_name  | salary |
|-----------|------------|------------|--------|
| 10101     | Srinivasan | Comp. Sci. | 65000  |
| 12121     | Wu         | Finance    | 90000  |
| 15151     | Mozart     | Music      | 40000  |
| 22222     | Einstein   | Physics    | 95000  |

tuple (or row / record)

changes constantly

a *relation instance* (i.e. the current values) is specified in a *table*  
each *tuple* describes a single *entity* of the type 'instructor'  
accessed/specified by Data Manipulation Language (DML)

# relation model

## relational model in detail

---

what is a database?

*consists of multiple relations (i.e. schema and its instance)*

information is broken up, each relation storing one part

- *instructor* stores information about instructors
- *teaches* stores information about who teaches what
- *course* store information on the course being offered

a bad design results in duplication/need for null values

course(name, prerequisite\_1, prerequisite\_2, teacher)

what if more than 2 prerequisites?

what if multiple instructors? over different years?

# relation model

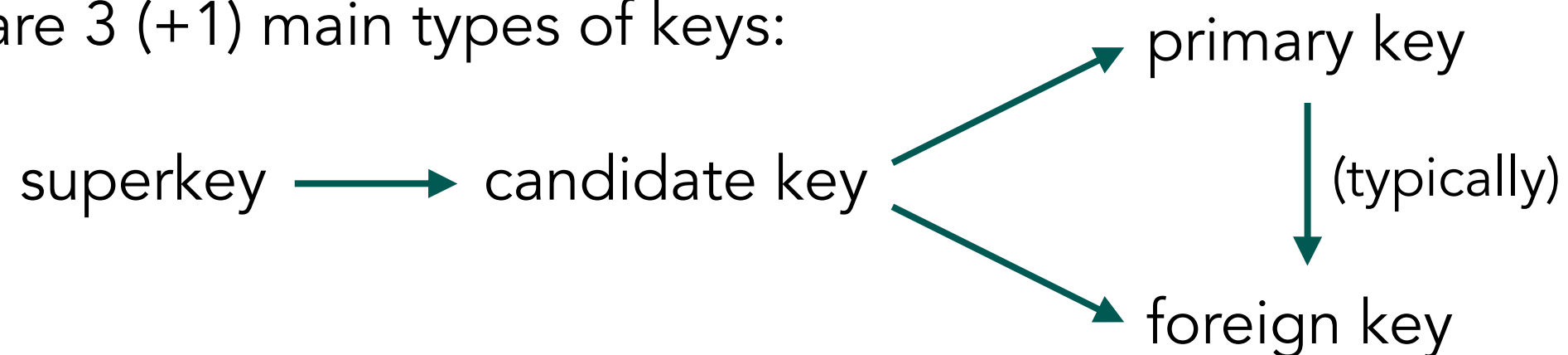
## keys: usage and types



keys allow to uniquely identify tuples

- 'key' component of any database!
- can consist of one or more attributes
  - simple key* uses one attribute,
  - composite key* uses more than one
- help to express connection between relations

there are 3 (+1) main types of keys:



# relation model

keys: usage and types

---

meet: the *superkey*

some (or all) of the attributes of a relation schema that are sufficient to uniquely identify any given tuple

→ may include extraneous attributes

| id    | name       | dept_name  | salary |
|-------|------------|------------|--------|
| 10101 | Srinivasan | Comp. Sci. | 65000  |
| 12121 | Wu         | Finance    | 90000  |
| 15151 | Mozart     | Music      | 40000  |
| 22222 | Einstein   | Physics    | 95000  |

# relation model

keys: usage and types

---

meet: the *superkey*

some (or all) of the attributes of a relation schema that are *sufficient* to uniquely identify any given tuple

| yes   |            | maybe      | maybe  |
|-------|------------|------------|--------|
| id    | name       | dept_name  | salary |
| 10101 | Srinivasan | Comp. Sci. | 65000  |
| 12121 | Wu         | Finance    | 90000  |
| 15151 | Mozart     | Music      | 40000  |
| 22222 | Einstein   | Physics    | 95000  |

yes



# relation model

keys: usage and types

---

the humble *candidate key*:

a *superkey* that is *minimal*, i.e. no excess attributes

| <i>no</i>  |            | <i>maybe</i> |        |
|------------|------------|--------------|--------|
| id         | name       | dept_name    | salary |
| 10101      | Srinivasan | Comp. Sci.   | 65000  |
| 12121      | Wu         | Finance      | 90000  |
| 15151      | Mozart     | Music        | 40000  |
| 22222      | Einstein   | Physics      | 95000  |
| <i>yes</i> |            |              |        |

# relation model

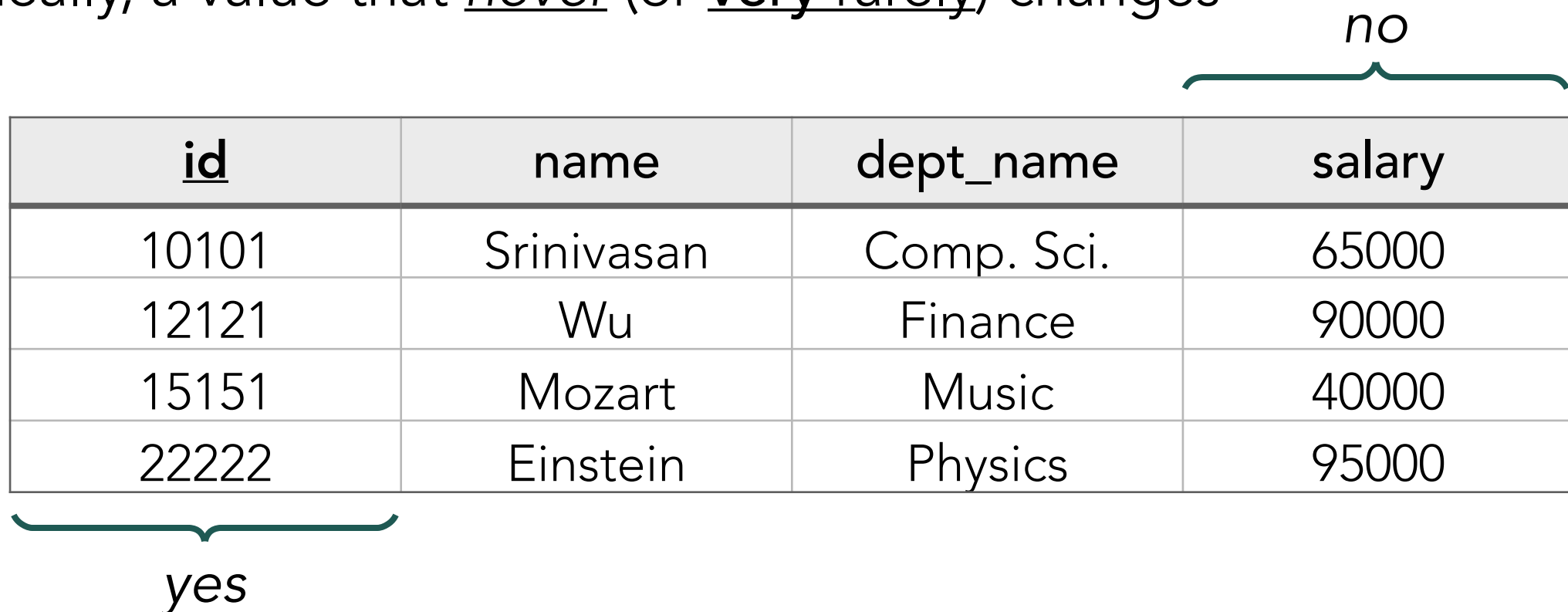
keys: usage and types

---

the all-important *primary key*:

a *candidate key* chosen as the *principal means* of identifying tuples in a relation: typically listed as first attribute

ideally, a value that never (or very rarely) changes



| <u>id</u> | name       | dept_name  | salary |
|-----------|------------|------------|--------|
| 10101     | Srinivasan | Comp. Sci. | 65000  |
| 12121     | Wu         | Finance    | 90000  |
| 15151     | Mozart     | Music      | 40000  |
| 22222     | Einstein   | Physics    | 95000  |

# relation model

keys: usage and types

the social *foreign key*:

an attribute that corresponds to the primary key of another relation and which is used to link tuples together

*instructor*

| <u>id</u> | name       | dept_name  | salary |
|-----------|------------|------------|--------|
| 10101     | Srinivasan | Comp. Sci. | 65000  |
| 1657      | Kim        | Comp. Sci. | 70000  |
| 15151     | Mozart     | Music      | 40000  |
| 22222     | Einstein   | Physics    | 95000  |

*referring relation*

*teaches*

| <u>instructor_id</u> | <u>course_id</u> | <u>semester</u> | <u>year</u> |
|----------------------|------------------|-----------------|-------------|
| 1657                 | CSC7072          | 1               | 2015        |
| 1657                 | CSC1023          | 2               | 2016        |
| ...                  | ...              | ...             | ...         |

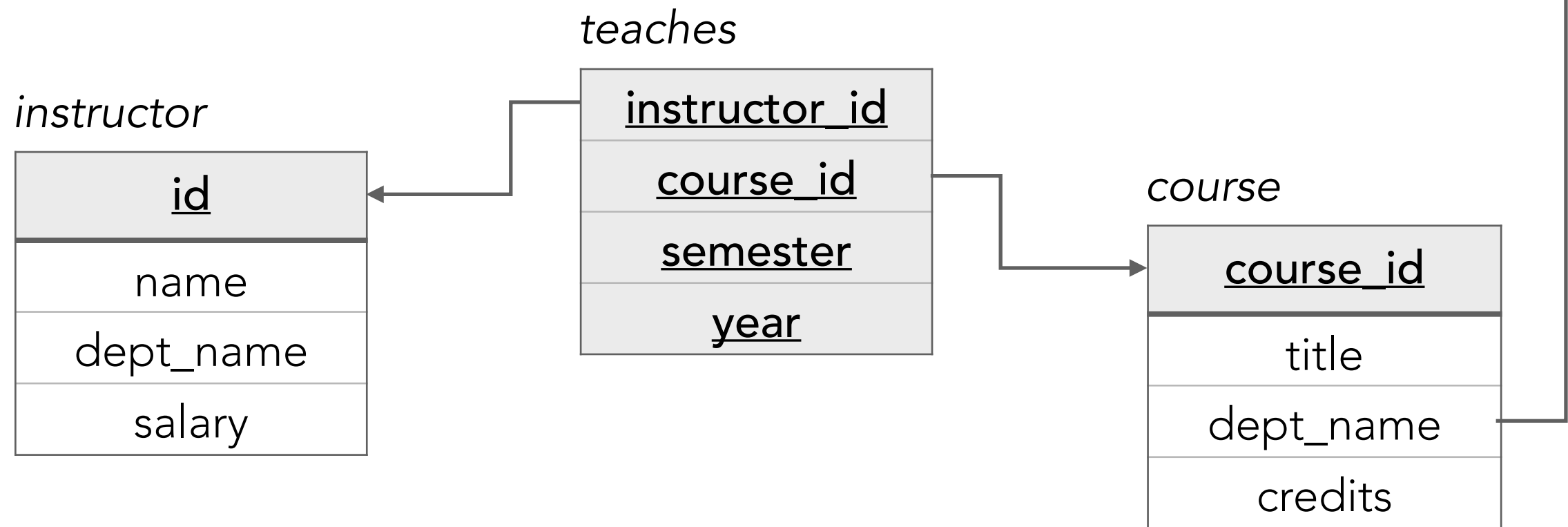
*referenced relation*

yes

# relation model

## schema diagrams

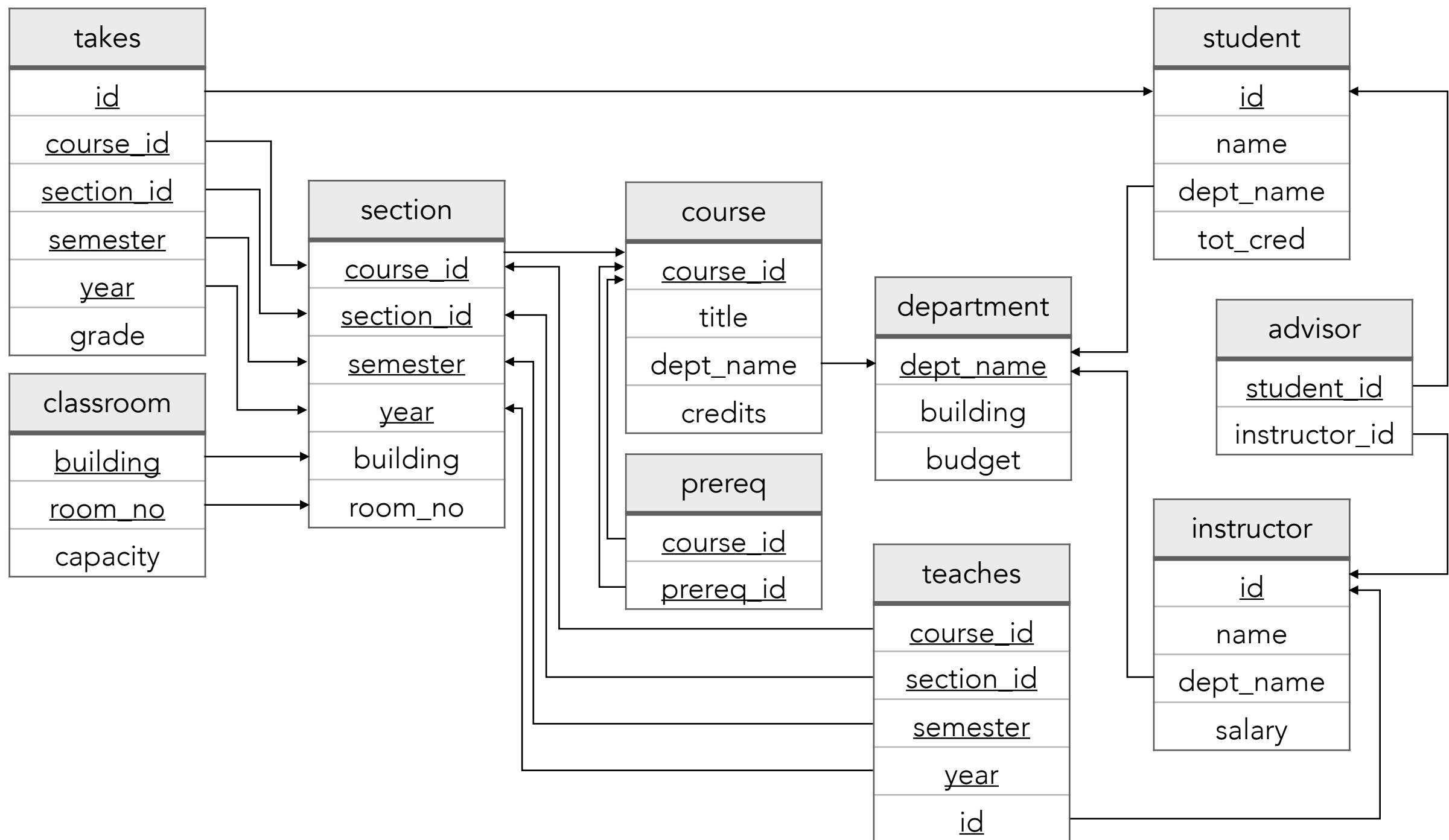
*schema diagram highlights the relations (and primary keys)*



*only values occurring in the primary key attribute of the referenced relation may occur in the foreign key attribute of the referencing relation*

# relation model

## schema diagram of our running example



# relation model

## relation query languages

---

a *relation query language* is used to retrieve info from a DB

SQL is a declarative relation query language

└ only express *what* we want to retrieve,  
not *how* to retrieve it (= procedural)

SQL is a so-called “*pure*” language based on relational algebra  
a *table* in databases closely resembles a *relation* in mathematics  
*hence the names!*

every operation in SQL:



- applies to a single or a pair of relations/tables (= input)
- gives a relation/table as response (= output)

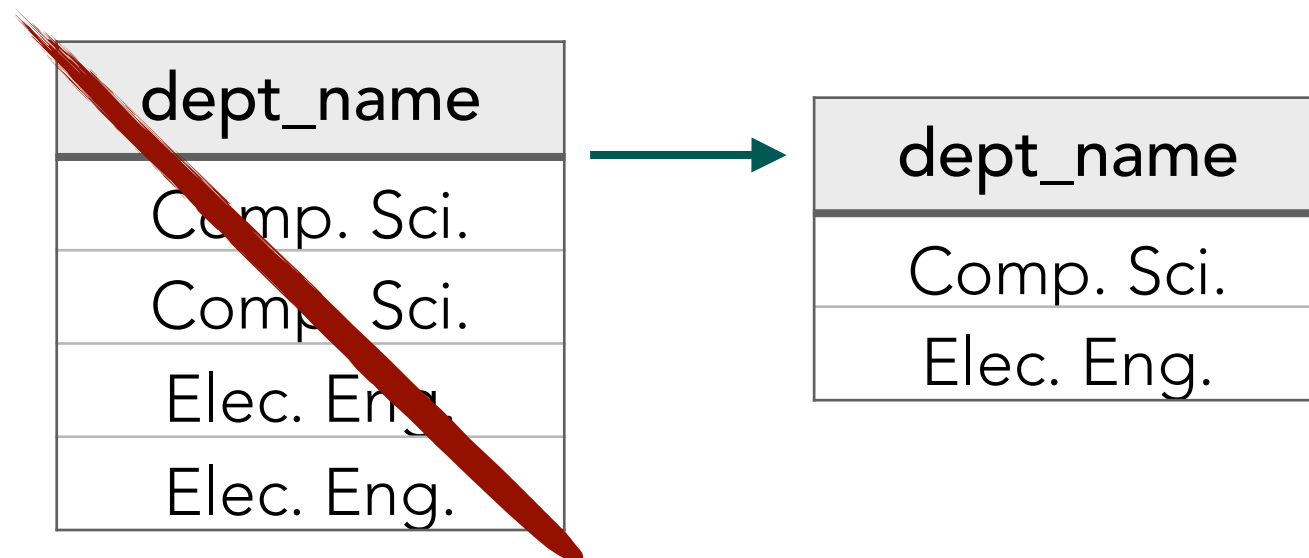
# relation model

## selection of attributes

*student*

| <u>id</u> | name     | dept_name  | tot_cred |
|-----------|----------|------------|----------|
| 12345     | Shankar  | Comp. Sci. | 32       |
| 54321     | Williams | Comp. Sci. | 32       |
| 76653     | Aoi      | Elec. Eng. | 60       |
| 98765     | Bourikas | Elec. Eng. | 98       |

*"select" dept\_name*



# relation model

## selection of attributes

---

*student*

| <u>id</u> | name     | dept_name  | tot_cred |
|-----------|----------|------------|----------|
| 12345     | Shankar  | Comp. Sci. | 32       |
| 54321     | Williams | Comp. Sci. | 32       |
| 76653     | Aoi      | Elec. Eng. | 60       |
| 98765     | Bourikas | Elec. Eng. | 98       |

*"select" dept\_name and tot\_cred*

| dept_name  | tot_cred |
|------------|----------|
| Comp. Sci. | 32       |
| Elec. Eng. | 60       |
| Elec. Eng. | 98       |



# relation model

## selection of tuples

---

*student*

| <u>id</u> | name     | dept_name  | tot_cred |
|-----------|----------|------------|----------|
| 12345     | Shankar  | Comp. Sci. | 32       |
| 54321     | Williams | Comp. Sci. | 32       |
| 76653     | Aoi      | Elec. Eng. | 60       |
| 98765     | Bourikas | Elec. Eng. | 98       |

*"select tuples" where tot\_cred >= 40*

| <u>id</u> | name     | dept_name  | tot_cred |
|-----------|----------|------------|----------|
| 76653     | Aoi      | Elec. Eng. | 60       |
| 98765     | Bourikas | Elec. Eng. | 98       |

# relation model

## Cartesian product

---

*instructor*

| <u>id</u> | name   | dept_name |
|-----------|--------|-----------|
| 3412      | Sophia | CS        |
| 0657      | Singh  | CS        |

*course*

| <u>course_id</u> | credits |
|------------------|---------|
| CSC7072          | 3       |
| CSC7075          | 2       |
| CSC7076          | 3       |

*Cartesian product* of instructor and course: instructor **x** course  
gives all possible combinations

| <u>id</u> | name   | dept_name | course_ID | credits |
|-----------|--------|-----------|-----------|---------|
| 3412      | Sophia | CS        | CSC7072   | 3       |
| 0657      | Singh  | CS        | CSC7075   | 2       |
| 3412      | Sophia | CS        | CSC7076   | 3       |
| 0657      | Singh  | CS        | CSC7072   | 3       |
| 3412      | Sophia | CS        | CSC7075   | 2       |
| 0657      | Singh  | CS        | CSC7076   | 3       |

# relation model

## union of two relations

---

*course\_in\_2014*

| <u>course_id</u> | credits |
|------------------|---------|
| CSC7072          | 3       |
| CSC7075          | 2       |
| CSC7074          | 4       |

*course\_in\_2015*

| <u>course_id</u> | credits |
|------------------|---------|
| CSC7072          | 3       |
| CSC7075          | 2       |
| CSC7076          | 3       |

*union* gives us elements from both relations without duplicates

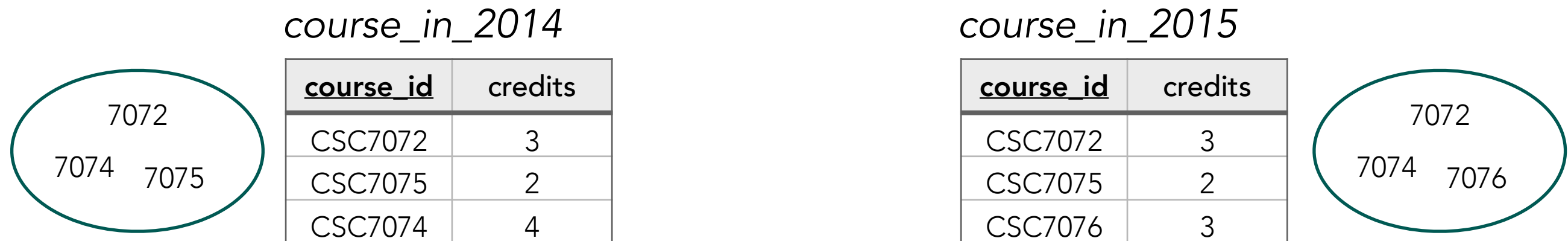
*course\_in\_2014* U *course\_in\_2015*

| <u>course_id</u> | credits |
|------------------|---------|
| CSC7072          | 3       |
| CSC7075          | 2       |
| CSC7074          | 4       |
| CSC7076          | 3       |

*which courses were taught in 2014 and/or 2015?*

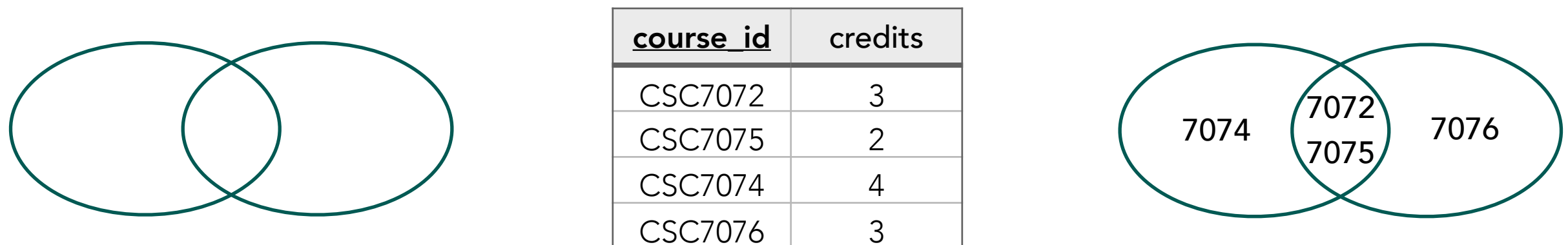
# relation model

## union of two relations



*union* gives us elements from both relations without duplicates

*course\_in\_2014* U *course\_in\_2015*



*which courses were thought in 2014 and/or 2015?*

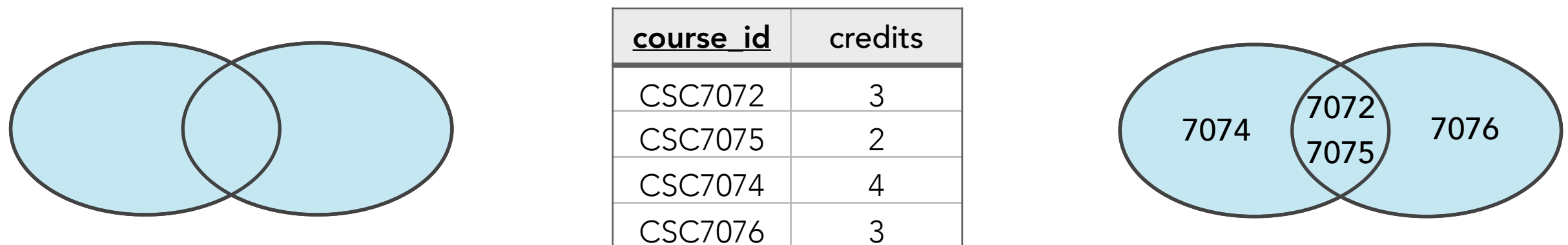
# relation model

## union of two relations



*union* gives us elements from both relations without duplicates

*course\_in\_2014* U *course\_in\_2015*



*which courses were thought in 2014 and/or 2015?*

# relation model

## intersection of two relations

---

*course\_in\_2014*

| <u>course_id</u> | credits |
|------------------|---------|
| CSC7072          | 3       |
| CSC7075          | 2       |
| CSC7074          | 4       |

*course\_in\_2015*

| <u>course_id</u> | credits |
|------------------|---------|
| CSC7072          | 3       |
| CSC7075          | 2       |
| CSC7076          | 3       |

*intersection* gives us elements found in both relations

$course\_in\_2014 \cap course\_in\_2015$

| <u>course_id</u> | credits |
|------------------|---------|
| CSC7072          | 3       |
| CSC7075          | 2       |

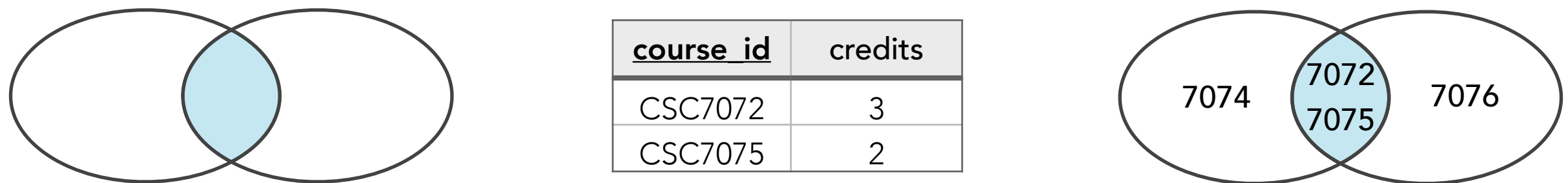
*which courses were taught both in 2014 and 2015?*

# relation model

## intersection of two relations



*intersection* gives us elements found in both relations  
 $course\_in\_2014 \cap course\_in\_2015$



*which courses were thought both in 2014 and 2015?*

# relation model

## difference of two relations

---

*course\_in\_2014*

| <u>course_id</u> | credits |
|------------------|---------|
| CSC7072          | 3       |
| CSC7075          | 2       |
| CSC7074          | 4       |

*course\_in\_2015*

| <u>course_id</u> | credits |
|------------------|---------|
| CSC7072          | 3       |
| CSC7075          | 2       |
| CSC7076          | 3       |

*difference* gives us elements in one but not in the other relation

*course\_in\_2014 / course\_in\_2015*

| <u>course_id</u> | credits |
|------------------|---------|
| CSC7074          | 4       |

*which courses were thought in 2014 but not in 2015?*

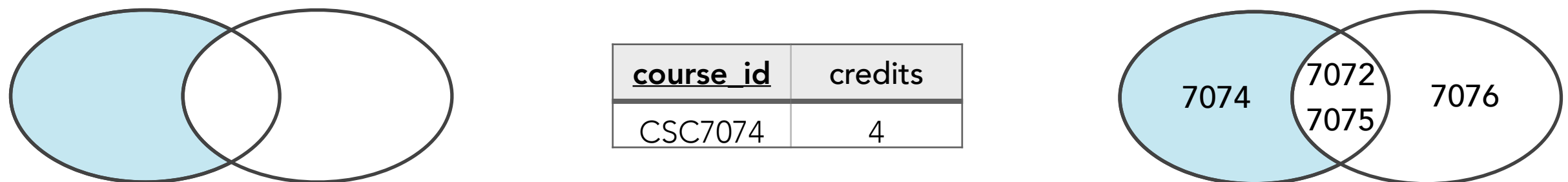


# relation model

## difference of two relations



*difference gives us elements in one but not in the other relation*  
*course\_in\_2014 / course\_in\_2015*



*which courses were thought in 2014 but not in 2015?*

# relation model

## natural join

when two relations have (some) attributes in common,  
combine the two relations based on these attributes

*instructor*

| <u>id</u> | name   | dept_name |
|-----------|--------|-----------|
| 3412      | Sophia | CS        |
| 0657      | Singh  | CS        |
| 1287      | Mary   | EE        |

*department*

| dept_name | building | budget |
|-----------|----------|--------|
| CS        | BCB      | 162    |
| PS        | DK       | 42     |
| EE        | ECIT     | 249    |

*natural join* of instructor and department is:

| <u>id</u> | name   | dept_name | building | budget |
|-----------|--------|-----------|----------|--------|
| 3412      | Sophia | CS        | BCB      | 162    |
| 0657      | Singh  | CS        | BCB      | 162    |
| 1287      | Mary   | EE        | ECIT     | 249    |