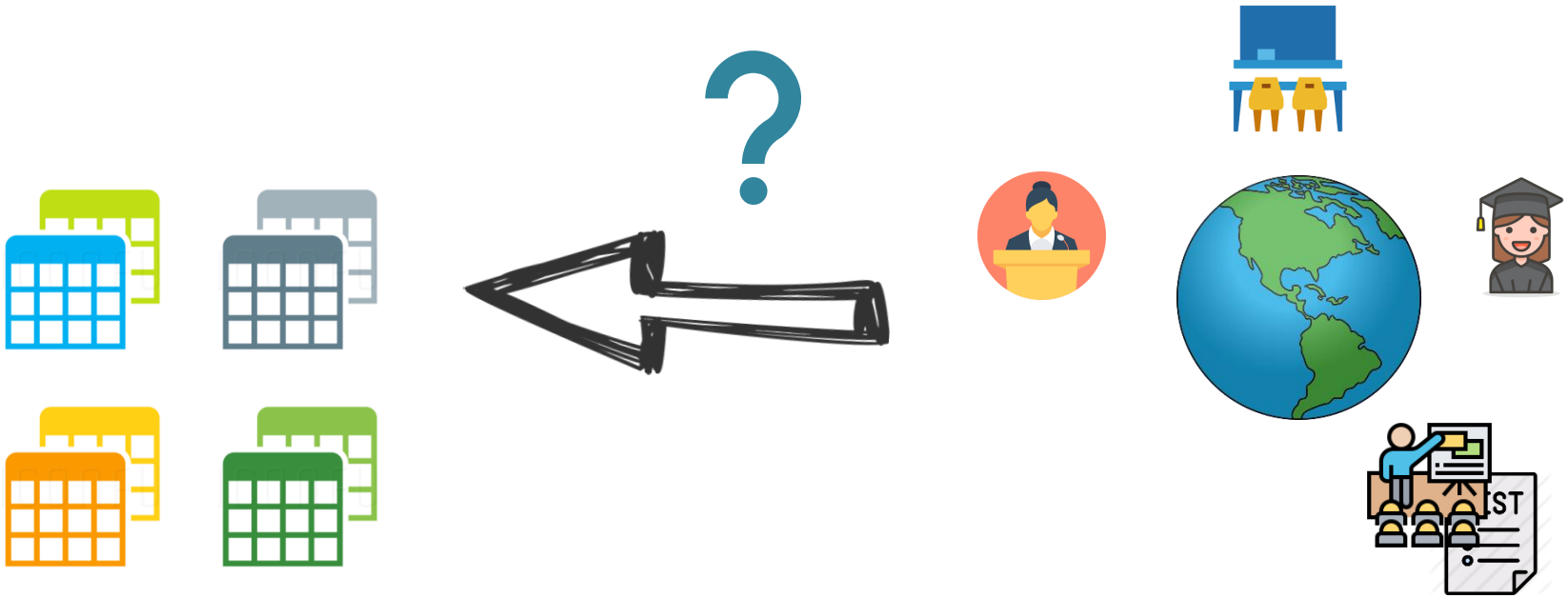
Normalization

Amos Azaria, Netanel Chkroun

Designing a Database



Normalization

- *Database normalization* is the process of structuring a database, usually a *relational database*, in accordance with a series of so-called *normal forms* in order to reduce data *redundancy* and improve data *integrity*.
- It was first proposed by *Edgar F. Codd* as part of his relational model.

https://en.wikipedia.org/wiki/Database_normalization

Dependencies

- An attribute (or set of attributes), B, is said to be dependent of another attribute (or set of attributes), A, if there exists a relation (function) such that A → B.
- In other words, if given A, it is not possible for an entry to have two different values for B, we say that A→B.
- For example, A = student ID, B=student first name.

  student ID -> student first name

- This dependency is also called functional dependency (B is functionally dependent of A).

Dependencies

- Obviously, for every B such that $B \subseteq A$, we have that $A \rightarrow B$.
  - E.g.: A = stFirstName, stLastName. B = stFirstName
  - F(stFirstName, stLastName) = stFirstName

A	B	Dependency?
{Street, HouseNum, City, State}	Zip code	$A \rightarrow B$
Day of week	Date = {Day, Month, Year}	$B \rightarrow A$
First Name	Last Name	None
{University, Department}	DepartmentHeadId	$A \rightarrow B$ and $B \rightarrow A$

Keys

- Candidate key: A minimal set of attributes that determines an entry. That is, all other attributes are dependent on the key.
- E.g.:
  - Student id in student table.
  - Course table: {id} or {name, year, semester}.

Minimal set: removal of any attribute from the set, will no longer determine the entry.

Courses

id	name	lecturer	year	semster
10	Introduction to intro.	Knows Nothing	2020	1
20	Calculus	Tamar Ezra	2021	1
30	Algebra	Shay Mann	2022	1
35	Calculus	Adel Smith	2022	1
40	Advanced Program...	David Gol	2022	2

Students

id	age	gender	degree	firstName	lastName
111	21	1	1	Chaya	Glass
444	23	0	1	Moti	Cohen
222	28	1	3	Tal	Negev
333	24	0	1	Gadi	Golan

Keys (cont.)

- Is studentName a key?
  - No (there may be multiple students with the same name)
- What would be a key for the grades table?
  - StudentId + courseId

Students

id	age	gender	degree	firstName	lastName
111	21	1	1	Chaya	Glass
444	23	0	1	Moti	Cohen
222	28	1	3	Tal	Negev
333	24	0	1	Gadi	Golan

Grades

courseId	studentId	grade	passed
20	111	43	0
20	222	85	1
30	111	90	1
30	444	95	1
40	222	67	1
40	333	40	0

Keys (cont.)

- A single table can have more than one set of keys (both being minimal), e.g.:
  - R(university, department, depHeadId)
    - {depHeadId}
    - {university, department}

Assuming every department has a single head, and a person can be a department head of a single department in a single university.

Prime / Non-Prime

- Prime attributes are attributes that are part of some candidate-key.
- Similarly, non-prime attributes are attributes that are not part of any candidate-key.

Prine / Non Prime (cont.)

- E.g.:

R(university, department, depHeadId)

  - Candidate keys are:
    - {depHeadId}
    - {university, department}
  - Prime ? Non prime?

Course table:

  - Candidate keys are:
    - {id}
    - {name, year, semester}
  - Prime? Non prime?

Courses

id	name	lecturer	year	semster
10	Introduction to intro.	Knows Nothing	2020	1
20	Calculus	Tamar Ezra	2021	1
30	Algebra	Shay Mann	2022	1
35	Calculus	Adel Smith	2022	1
40	Advanced Program...	David Gol	2022	2

Super-Key

- **Any** set of attributes that determines an entry.
  - E.g. the whole set of attributes.
- Same as candidate key, just without the minimal requirement.

Candidate kay is called also-
'Minimal super key'

Normalization

- What is the problem with the following relation?

StudentId	StudentFirst	StudentLast	Courses
542	Yossi	Agasi	4244, 3423, 6734
956	Tamar	Atiya	4244, 5437
754	Gabbi	Matar	4325, 6543, 564
327	Shay	Shalom	5324

Multiple values for a single attribute. How can we get all students in 3423?

Normalization

Heavy redundancy.
What happens when we update student's address? And what if we delete all grades of a student?

- And with this one?

StudentId	StudentFirst	StudentLast	Address	CourseId	Grade
542	Yossi	Agasi	Harambam 45, Ariel	4244	87
542	Yossi	Agasi	Harambam 45, Ariel	3423	65
956	Tamar	Atiya	Hadekel 12, Herzeliya	4244	86
542	Yossi	Agasi	Harambam 45, Ariel	6734	80

Normalization

- *Database normalization* is the process of structuring a database, usually a relational database, in accordance with a series of so-called *normal forms* in order to reduce data redundancy and improve data integrity.
- It was first proposed by *Edgar F. Codd* as part of his relational model.

https://en.wikipedia.org/wiki/Database_normalization

1NF (=Normalized Form)

- Every attribute must hold a single atomic value (searchability)

StudentId	StudentFirst	StudentLast	Courses
542	Yossi	Agasi	4244, 3423, 6734
956	Tamar	Atiya	4244, 5437
754	Gabbi	Matar	4325, 6543, 564
327	Shay	Shalom	5324

StudentId	StudentFirst	StudentLast	Courses
542	Yossi	Agasi	4244
956	Tamar	Atiya	4244
754	Gabbi	Matar	4325
327	Shay	Shalom	5324
542	Yossi	Agasi	3423
542	Yossi	Agasi	6734
956	Tamar	Atiya	5437
754	Gabbi	Matar	6543
754	Gabbi	Matar	564

2NF

- Table must be in 1NF
- Non-prime attributes do not depend on a strict(proper) subset of a candidate key.

But StudentFirst, StudentLast and Address depend only on StudentId

What is the key?

StudentId+CourseId

StudentId	StudentFirst	StudentLast	Address	CourseId	Grade
542	Yossi	Agasi	Harambam 45, Ariel	4244	87
542	Yossi	Agasi	Harambam 45, Ariel	3423	65
956	Tamar	Atiya	Hadekel 12, Herzeliya	4244	86
542	Yossi	Agasi	Harambam 45, Ariel	6734	80

Fixing Table to Become 2NF

- In order to correct a relation that is not in 2NF, we split the information into 2 tables:

StudentId	StudentFirst	StudentLast	Address	CourseId	Grade
542	Yossi	Agasi	Harambam 45, Ariel	4244	87
542	Yossi	Agasi	Harambam 45, Ariel	3423	65
956	Tamar	Atiya	Hadekel 12, Herzeliya	4244	86
542	Yossi	Agasi	Harambam 45, Ariel	6734	80

StudentId	StudentFirst	StudentLast	Address
542	Yossi	Agasi	Harambam 45, Ariel
956	Tamar	Atiya	Hadekel 12, Herzeliya

StudentId	CourseId	Grade
542	4244	87
542	3423	65
956	4244	86
542	6734	80

Note that the new tables have 20 cells in total, while the original table had 24 cells. The new tables have 105 characters (combined) while the old table had 143.

2NF (cont.)

- Given: R(author, bookId, #pages)
- Each book can have one or more authors
- What is the candidate key?
  - {author, bookId}
- Is it in 2NF?
  - No:
    - bookId → #pages
    - {bookId} isn't a key
- How to fix?
  - Split to R1(author, bookId) and R2(bookId, #pages)

Authors Relation

Books Relation