NORMAL FORMS

Dr. Jose Annunziato

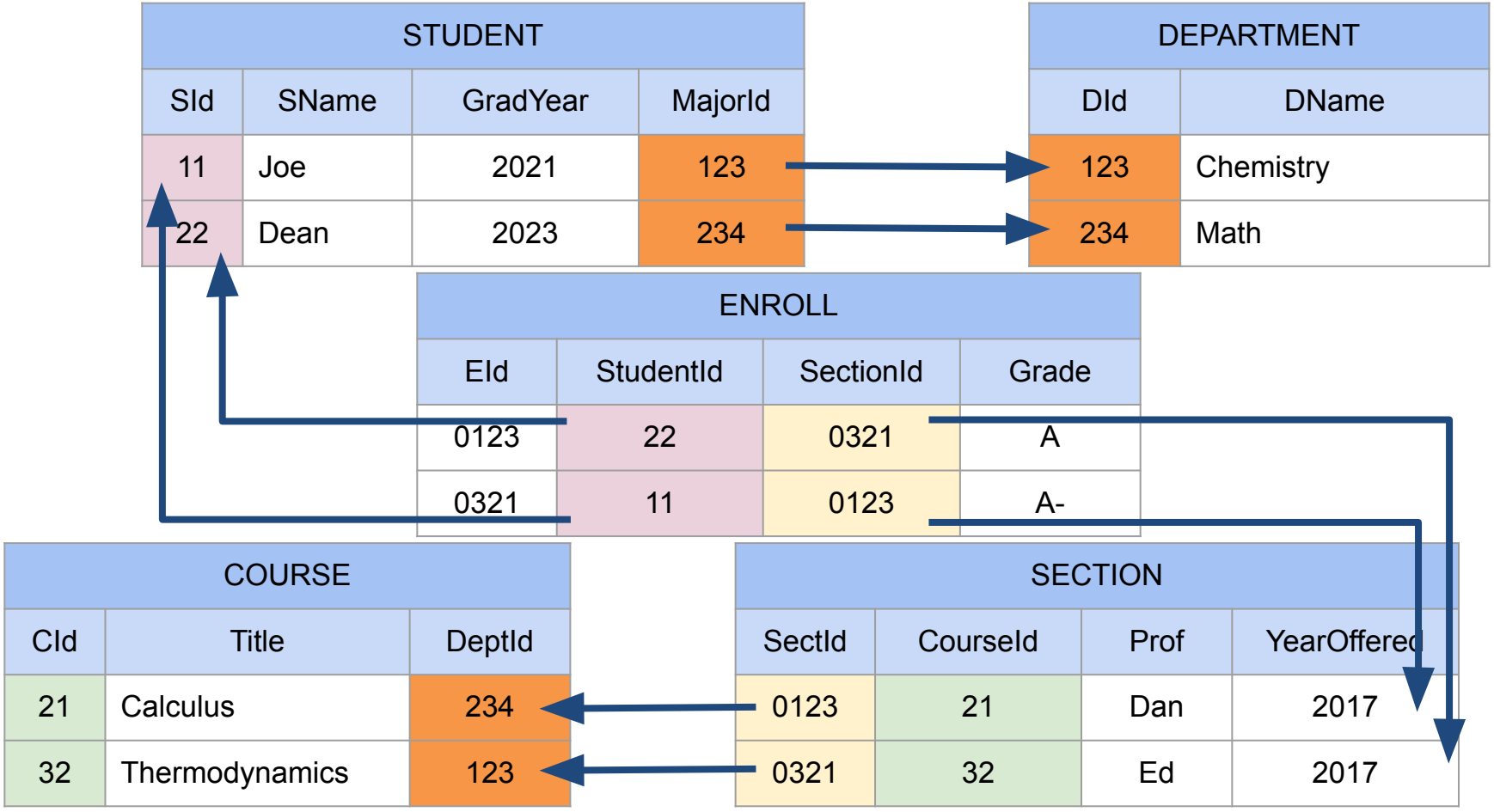
Normalizing Data

- Why not just put it all in one table?

REDUNDANCY!
INCONSISTENCY RISK!

ONE_BIG_TABLE							
Id	SName	GradYear	DName	Title	Prof	YearOffered	Grade
1	Joe	2021	Chemistry	Calculus	Dan	2017	A
2	Alice	2017	Chemistry	Calculus	Dan	2017	A-
3	Bob	2020	Chemistry	Calculus	Dan	2017	A
4	Dan	2018	Chemistry	Calculus	Dan	2017	B+
5	Dean	2023	Math	Thermodynamics	Ed	2017	A-
6	Chuck	2017	Math	Thermodynamics	Ed	2017	B+
7	Ed	2020	Math	Thermodynamics	Ed	2017	A
8	Frank	2018	Math	Thermodynamics	Ed	2017	A-

Schema Tables



STUDENT

SId	SName	GradYear	MajorId
11	Joe	2021	123
22	Dean	2023	234

DEPARTMENT

DId	DName
123	Chemistry
234	Math

ENROLL

EId	StudentId	SectionId	Grade
0123	22	0321	A
0321	11	0123	A-

COURSE

CId	Title	DeptId
21	Calculus	234
32	Thermodynamics	123

SECTION

SectId	CourseId	Prof	YearOffered
0123	21	Dan	2017
0321	32	Ed	2017

Normalization

- Organize fields and tables ***minimizing redundancy*** & ***dependency***

- Usually ***split*** large tables ***into several*** smaller ***tables***

- And ***define relations*** between them

Normalization

- Objective is ***decouple*** data so ***updates*** (insertions, deletions, & updates) are to ***individual tables***

- Changes propagate through relations

First Normal Form (1NF)

- A table is in first normal form if it contains **no repeating fields** / columns
- Repeating fields
  - **waste disk space**
  - **less flexible**
  - more **difficult to search**
- ***1NF – data must depend on the key***

First Normal Form (1NF)

- Consider the following ***Customer*** table

CID	FirstName	Surname	Telephone
123	Alice	Wonderland	555-123-2345
456	**Billy**	**Squire**	**555-234-3455**
789	Daniel	Craig	555-345-5433

First Normal Form Example (1/3)

- Now consider adding additional phones
- We can put them all in a single field Telephone**s**

CID	FirstName	Surname	Telephones
123	Alice	Wonderland	555-123-2345
456	**Billy**	**Squire**	**555-123-3455,555-234-3455**
789	Daniel	Craig	555-345-5433

First Normal Form Example (1/3)

- Or separate columns: Telephone1, 2. But what about 3, 4, …?

CID	FirstName	Surname	Telephone1	Telephone2
123	Alice	Wonderland	555-123-2345	
456	**Billy**	**Squire**	**555-123-3455**	**555-234-3455**
789	Daniel	Craig	555-345-5433	

First Normal Form Example (2/3)

- Or we can have separate records

CID	FirstName	Surname	Telephone
123	Alice	Wonderland	555-123-2345
456	**Billy**	**Squire**	**555-123-3455**
457	**Billy**	**Squire**	**555-234-3455**
789	Daniel	Craig	555-345-5433

- But this introduces ***redundancy*** → inconsistency

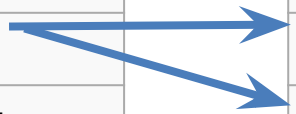
First Normal Form Example (3/3)

- Remove redundant row

CID	FirstName	Surname	Telephone
123	Robert	Ingram	555-861-2025
456	**Jane**	**Wright**	**555-403-1659**
457	**Jane**	**Wright**	**555-776-4100**
789	Maria	Fernandez	555-808-9633

- Revised ***Customer*** table and new ***Telephones*** table

CID	FirstName	Surname
123	Robert	Ingram
456	**Jane**	**Wright**
789	Maria	Fernandez



CID	Telephone
123	555-861-2025
456	**555-403-1659**
456	**555-776-4100**
789	555-808-9633

2NF – The Whole Key

- Second normal form refactors tables that have columns that do not **depend on whole *primary key***
- Consider the following table

CID	*Semester*	#Seats	CourseName
CS1500	2009-1	100	Programming
CS1500	2009-2	150	Programming
CS5200	2009-1	200	Databases
CS5200	2010-1	150	Databases
CS5610	2009-2	120	Web Design

- **CourseName** column does not **depend on whole PK**, only on one of the keys: **CID**

Achieving 2NF

- We achieve 2NF using the same technique as 1NF

CID	***Semester***	#Seats
CS1500	2009-1	100
CS1500	2009-2	100
CS5200	2009-1	200
CS5200	2010-1	150
CS5610	2009-2	120

- New Course table referencing table above

CID	CourseName
CS1500	Programming
CS5200	Databases
CS5610	Web Design

Another 2NF Example (1/2)

- Consider the following Skills table

Employee	**Skill**	**Current Location**
Alice	QA	Cambridge
Alice	CSS	Cambridge
Alice	HTML	Cambridge
Bob	jQuery	Newton
Charly	Java	Newton
Charly	C#	Newton
Dan	jQuery	Newton

- {***Employee, Skill***} is PK, but ***Location*** depends only on ***Employee***, not ***Skill*** → redundancy

Another 2NF Example (2/2)

- Split into several tables

Employee	Skill
Alice	QA
Alice	CSS
Alice	HTML
Bob	jQuery
Charly	Java
Charly	C#
Dan	jQuery

Employee	Current Location
Alice	Cambridge
Bob	Newton
Charly	Newton
Dan	Newton

But 2NF May Not be Enough

- This 2NF table could have problems

Course	Year	Prof	Prof Date of Birth
Web Dev	2012	Alice Wonder	26 November 1996
Intro to DB	2012	Bob Marley	6 February 1945
C++	2011	Charly Garcia	23 October 1951
Web Dev	2013	Bob Marley	6 February 1945
Intro to DB	2013	Alice Wonder	26 November 1996

But 2NF May Not be Enough

- This 2NF table could have problems

Course	Year	Prof	Prof Date of Birth
Web Dev	2012	Alice Wonder	26 November 1865
Intro to DB	2012	Bob Marley	6 February 1945
C++	2011	Charly Garcia	23 October 1951
Web Dev	2013	Bob Marley	6 February 1945
Intro to DB	2013	Alice Wonder	26 November 1865

- {***Course, Year***} is PK and other fields depend on them → 2NF

But 2NF May Not be Enough

- This 2NF table could have problems

Course	Year	Prof	Prof Date of Birth
Web Dev	2012	Alice Wonder	26 November 1865
Intro to DB	2012	Bob Marley	6 February 1945
C++	2011	Charly Garcia	23 October 1951
Web Dev	2013	Bob Marley	6 February 1945
Intro to DB	2013	Alice Wonder	26 November 1865

- {***Course, Year***} is PK and other fields depend on them → 2NF
- But, we can mistakenly change Prof's birth date

But 2NF May Not be Enough

- This 2NF table could have problems

Course	Year	Prof	Prof Date of Birth
Web Dev	2012	Alice Wonder	26 November 1865
Intro to DB	2012	Bob Marley	6 February 1945
C++	2011	Charly Garcia	23 October 1951
Web Dev	2013	Bob Marley	6 February 1945
Intro to DB	2013	Alice Wonder	26 November 1865

- {***Course, Year***} is PK and other fields depend on them → 2NF
- But, we can mistakenly change Prof's birth date
- Fix with 3NF

3NF – Nothing But the Key

- Every non-key attribute must provide a fact about ***the key, the whole key, and nothing but the key***

Course	Year	Prof	Prof Date of Birth
Web Dev	2012	**Alice Wonder**	**26 November 1865**
Intro to DB	2012	**Bob Marley**	**6 February 1945**
C++	2011	**Charly Garcia**	**23 October 1951**
Web Dev	2013	**Bob Marley**	**6 February 1945**
Intro to DB	2013	**Alice Wonder**	**26 November 1865**

- To fix, again, split tables into separate related tables

Achieving 3NF

- Removing birth date from table:

Course	Year	Prof
Web Dev	2012	**Alice Wonder**
Intro to DB	2012	**Bob Marley**
C++	2011	**Charly Garcia**
Web Dev	2013	**Bob Marley**
Intro to DB	2013	**Alice Wonder**

- And moving it to a dedicated table

Prof	Prof Date of Birth
Alice Wonder	**26 November 1865**
Bob Marley	**6 February 1945**
Charly Garcia	**23 October 1951**