THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

4b. Normalization

CSCI 2541 Database Systems & Team Projects

Gabe

Slides adapted from Prof. Bhagi Narahari; and Silberschatz, Korth, Sudarshan, and Wood

Announcements?

Last time…

SQL DDL & DML

Entity Relationship Model

Normalization

- Bad Schemas
- Normal Forms
- Functional Dependencies

this time…

Good Schemas

The ER model can help us design a logical DB structure that matches our business goals

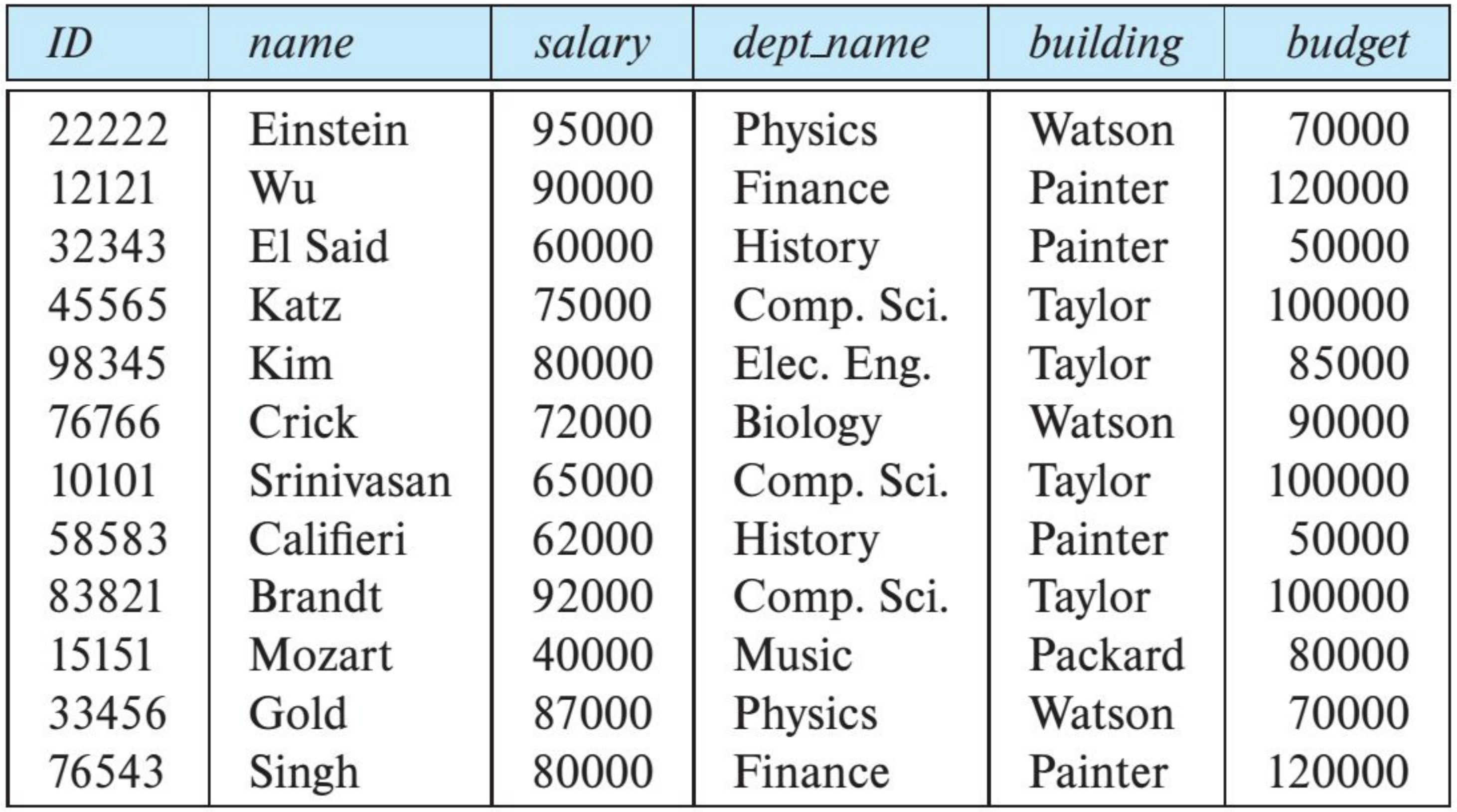
The conceptual schema must be translated into a logical (SQL) schema

How do we judge if a SQL schema is well designed?

Bad Schemas

Let’s track professors and their department

- We will put all the info together in one table so we don’t have to worry about joining stuff!



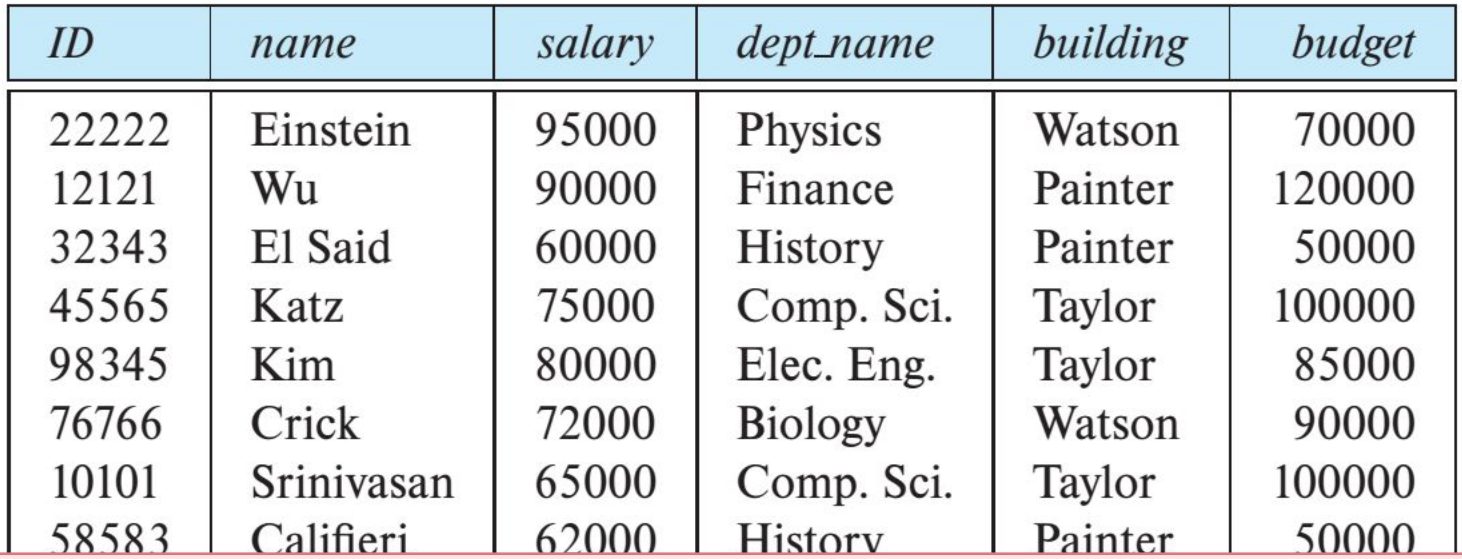
ID	*name*	*salary*	*dept_name*	*building*	*budget*
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan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

Why is this a bad idea?

Bad Schemas

Let’s track professors and their department

- We will put all the info together in one table so we don’t have to worry about joining stuff!



ID	*name*	*salary*	*dept_name*	*building*	*budget*
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000

Update Anomalies: need to modify all repetitive rows

Insertion Anomalies: Need to use NULL if we add a department with no instructors

Deletion Anomalies: Removing all instructors loses information about the department

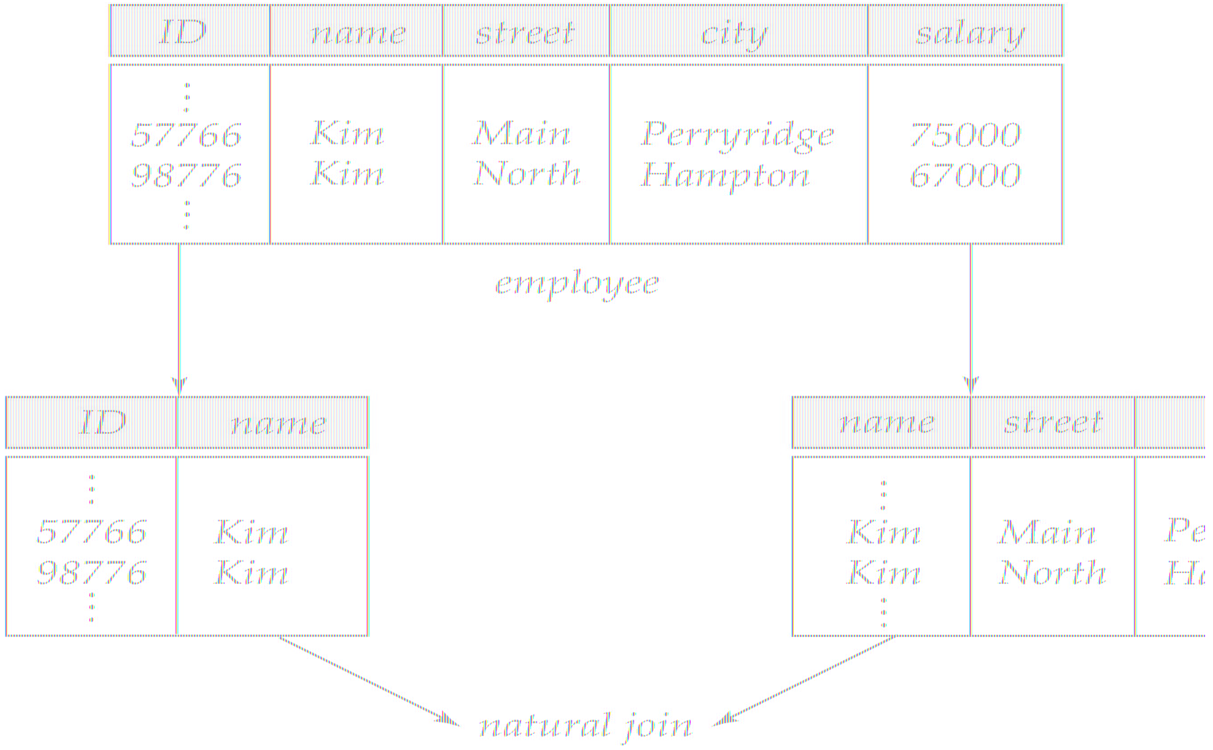
Splitting Tables

Decomposing into separate tables helps resolve this… but there are multiple ways to split tables

- Not all decompositions are good!

Let's split our table into two parts, and use Name attribute to connect them

- Good idea?



ID	name	street	city	salary
⋮				
57766	Kim	Main	Perryridge	75000
98776	Kim	North	Hampton	67000
⋮				

employee

ID	name
⋮	
57766	Kim
98776	Kim
⋮	

name	street	
⋮		
Kim	Main	Pe
Kim	North	H
⋮		

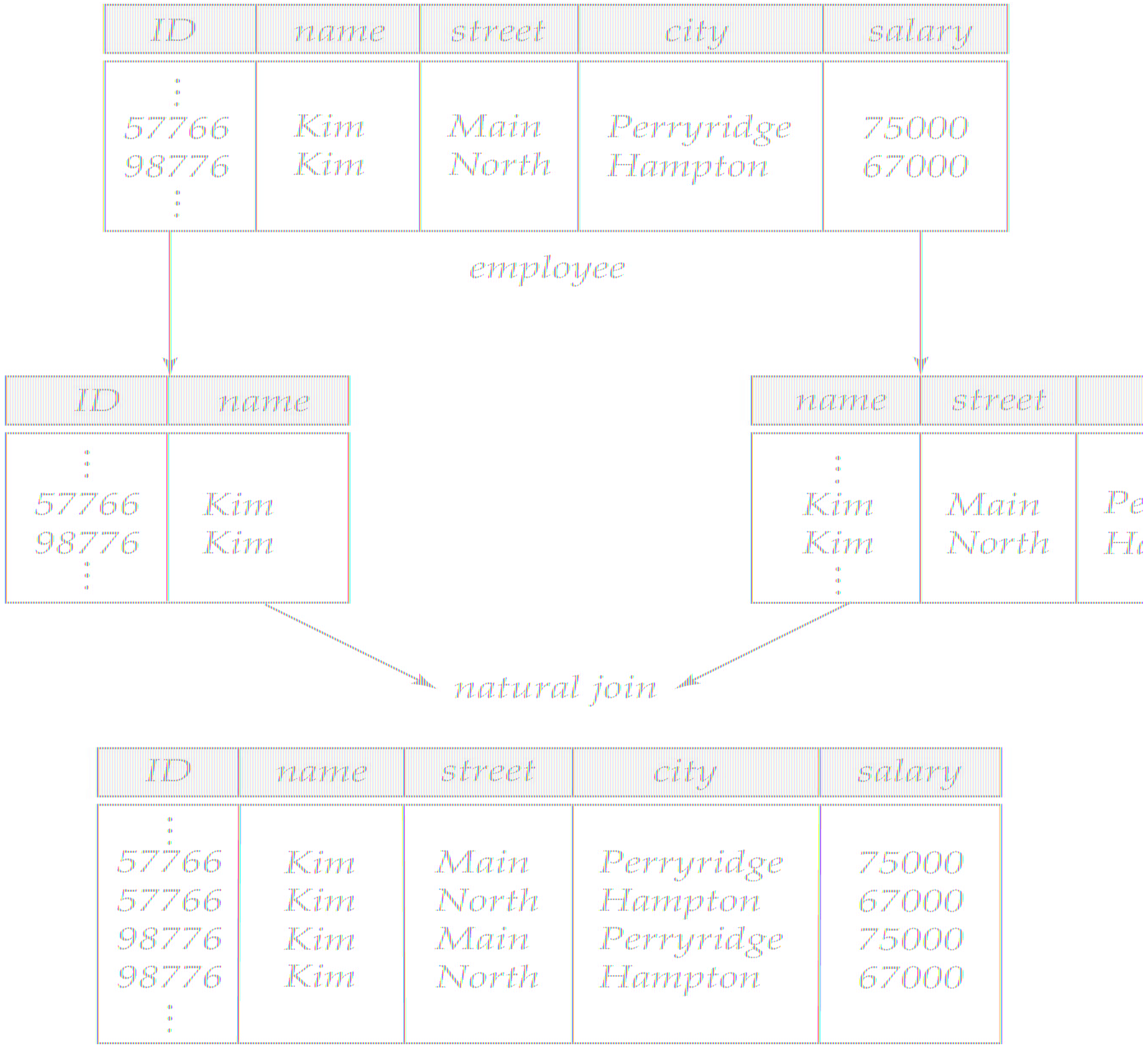
What happens if I join these tables "ON name = name"?

Splitting Tables

Decomposing into separate tables helps resolve this… but there are multiple ways to split tables

- Not all decompositions are good!

A **Lossy Decomposition** results in us losing data or getting incorrect data if we try to merge back using a join



ID	name	street	city	salary
⋮				
57766	Kim	Main	Perryridge	75000
98776	Kim	North	Hampton	67000
⋮				

employee

ID	name
⋮	
57766	Kim
98776	Kim
⋮	

name	street	
⋮		
Kim	Main	Pe
Kim	North	H
⋮		

natural join

ID	name	street	city	salary
⋮				
57766	Kim	Main	Perryridge	75000
57766	Kim	North	Hampton	67000
98776	Kim	Main	Perryridge	75000
98776	Kim	North	Hampton	67000
⋮				

What is Normalization?

1. Tests to see how “good” a schema is

2. Normalization algorithms to decompose relations into smaller relations that contain less redundancy
   - This decomposition requires that **no information is lost** and **reconstruction** of the original relations from the smaller relations must be possible.

Normalization should be done when you design your schema and anytime you update it

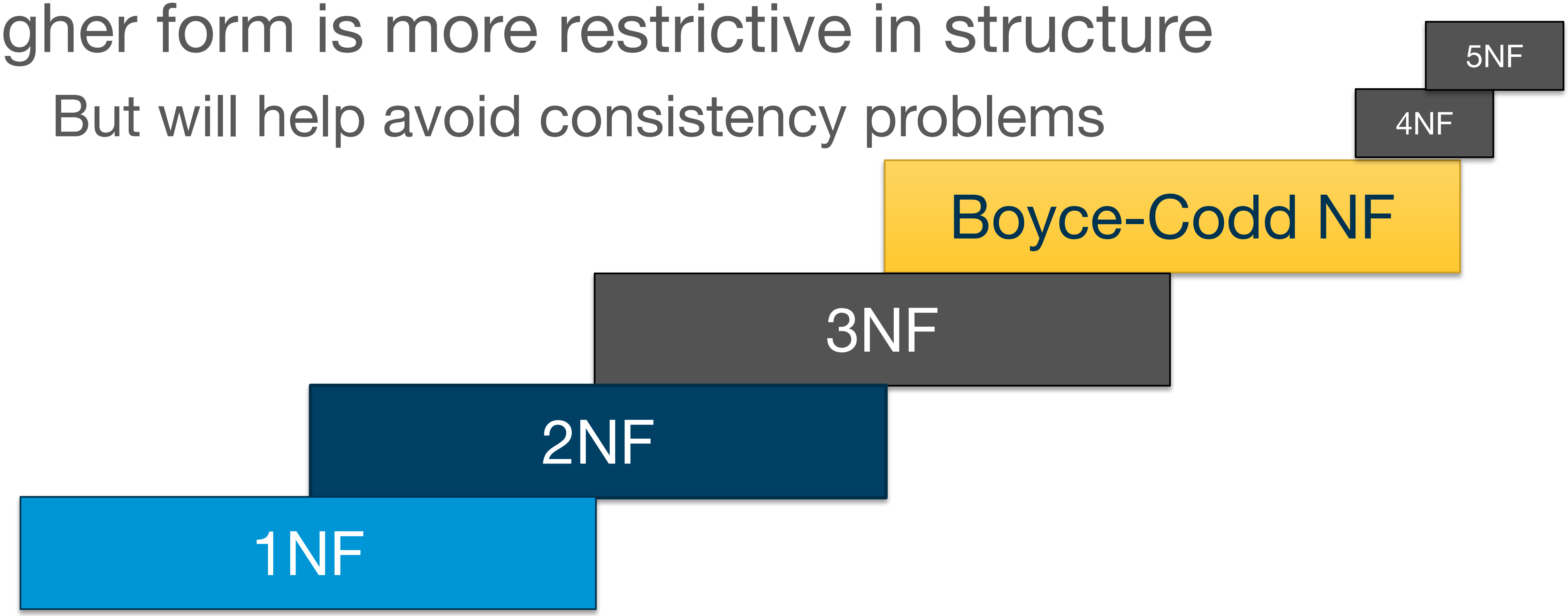
Normal Forms

Normal forms give us a hierarchy of rules

- No normalization - unconstrained, messy data
- First Normal Form - removes some redundancy
- Second Normal From - removes more redundancy… etc

Higher form is more restrictive in structure

- But will help avoid consistency problems



First Normal Form (1NF)

Attributes should be atomic and tables should have no repeating groups

Each field only has one value

No columns repeat the same "type" of information

No duplicate rows in the table; order doesn't matter

1NF Examples

Attributes should be atomic and tables should have no repeating groups

Do these violate 1NF?

Customer ID	First Name	Surname	Telephone Number
123	Pooja	Singh	555-861-2025, 192-122-1111
456	San	Zhang	(555) 403-1659 Ext. 53; 182-929-2929
789	John	Doe	555-808-9633

Customer ID	First Name	Surname	TNumber1	TNumber2
123	Pooja	Singh	555-861-2025	192-122-1111
456	San	Zhang	(555) 403-1659 Ext. 53	182-929-2929
789	John	Doe	555-808-9633	

Examples from https://en.wikipedia.org/wiki/First_normal_form

1NF Examples

Attributes should be atomic and tables should have no repeating groups

Do these violate 1NF?

Customer ID	First Name	Surname	Telephone Number
123	Pooja	Singh	555-861-2025, 192-122-1111
456	San	Zhang	(555) 403-1659 Ext. 53; 182-929-2929
789	John	Doe	555-808-9633

Customer ID	First Name	Surname	TNumber1	TNumber2
123	Pooja	Singh	555-861-2025	192-122-1111
456	San	Zhang	(555) 403-1659 Ext. 53	182-929-2929
789	John	Doe	555-808-9633	

Both are bad!

Examples from https://en.wikipedia.org/wiki/First_normal_form

1NF Split or Flatten

Attributes should be atomic and tables should have no repeating groups

Possible solutions

Customer ID	First Name	Surname	Telephone Number
123	Pooja	Singh	555-861-2025
123	Pooja	Singh	192-122-1111
456	San	Zhang	182-929-2929
456	San	Zhang	(555) 403-1659 Ext. 53
789	John	Doe	555-808-9633

OR

Customer ID	First Name	Surname
123	Pooja	Singh
456	San	Zhang
789	John	Doe

Customer ID	Telephone Number
123	555-861-2025
123	192-122-1111
456	(555) 403-1659 Ext. 53
456	182-929-2929
789	555-808-9633

Examples from https://en.wikipedia.org/wiki/First_normal_form

1NF Violations

Generally easy to detect:

1. Check for Column names with a number (telephone1, telephone2, course1, course2, etc)

2. Make sure that order of rows doesn’t matter

3. Have a primary key to enforce uniqueness across rows

Second Normal Form (2NF)

No value in a table should be dependent on only **part** of a key that uniquely identifies a row

It must be in 1NF and…

We should **not** be able to derive the value of a column based on only **a part of a Candidate Keys**

- Must hold for all Candidate Keys if there are multiple

Reminder: Key types

Superkey of R:

- A (**possibly larger than necessary**) set of attributes that is sufficient to uniquely identify each tuple in r(R)

Candidate Key of R: A “minimal” superkey

- A **minimal set** of attributes to denote uniqueness!
- A Candidate Key is a Superkey but opposite may not be true

Primary Key: A specific Candidate Key chosen to represent a relation/table

2NF Examples

No value in a table should be dependent on only part of a key that uniquely identifies a row

Does this violate 2NF?

Customer ID	First Name	Surname	Telephone Number
123	Pooja	Singh	555-861-2025
123	Pooja	Singh	192-122-1111
456	San	Zhang	182-929-2929
456	San	Zhang	(555) 403-1659 Ext. 53
789	John	Zhang	555-808-9633

2NF Examples

No value in a table should be dependent on only part of a key that uniquely identifies a row

Does this violate 2NF?

Customer ID	First Name	Surname	Telephone Number
123	Pooja	Singh	555-861-2025
123	Pooja	Singh	192-122-1111
456	San	Zhang	182-929-2929
456	San	Zhang	(555) 403-1659 Ext. 53
789	John	Zhang	555-808-9633

Yes!

- Our Key is (Customer ID, Telephone Number), but from Customer ID alone we could uniquely identify the name
- We could make func(CustomerID) -> (First Name, Surname)

In general, better to use the splitting method for 1NF

2NF vs 1NF

Why do we care??

1NF

Customer ID	First Name	Surname	Telephone Number
123	Pooja	Singh	555-861-2025
123	Pooja	Singh	192-122-1111
456	San	Zhang	182-929-2929
456	San	Zhang	(555) 403-1659 Ext. 53
789	John	Zhang	555-808-9633

VS

2NF

Customer ID	First Name	Surname
123	Pooja	Singh
456	San	Zhang
789	John	Zhang

Customer ID	Telephone Number
123	555-861-2025
123	192-122-1111
456	(555) 403-1659 Ext. 53
456	182-929-2929
789	555-808-9633

2NF vs 1NF

Redundant data can lead to inconsistencies if it is only partially updated!

1NF

Customer ID	**First Name**	**Surname**	**Telephone Number**
123	Pooja	Singh	555-861-2025
123	Pooja	Sing	192-122-1111
456	San	Zhang	182-929-2929
456	San	Zhang	(555) 403-1659 Ext. 53
789	John	Zhang	555-808-9633

VS

2NF

Customer ID	**First Name**	**Surname**
123	Pooja	Sing
456	San	Zhang
789	John	Zhang

Customer ID	**Telephone Number**
123	555-861-2025
123	192-122-1111
456	(555) 403-1659 Ext. 53
456	182-929-2929
789	555-808-9633

More 2NF Examples

Manufacturer	Model	Price	Manufacturer country
Forte	X-Prime	50	Italy
Forte	Ultraclean	50	Italy
Dent-o-Fresh	EZbrush	65	USA
Brushmaster	SuperBrush	34	USA
Kobayashi	ST-60	22	Japan
Hoch	Toothmaster	18	Germany
Hoch	X-Prime	50	Germany

More 2NF Examples

This avoids **Update Anomalies**

- Previously we would have had to scan all tuples if a manufacturer moved to a different country to ensure consistency

Manufacturer	Model	Price
Forte	X-Prime	45
Forte	Ultraclean	50
Dent-o-Fresh	EZbrush	65
Brushmaster	SuperBrush	34
Kobayashi	ST-60	22
Hoch	Toothmaster	18
Hoch	X-Prime	22

Manufacturer	Country
Forte	Italy
Dent-o-Fresh	USA
Brushmaster	USA
Kobayashi	Japan
Hoch	Germany

Third Normal Form (3NF)

No value should be able to be derived based on another non-key field

It must be in 2NF and…

all **non-prime attributes** depend only on the **candidate keys** and do not have a **transitive dependency** on another key

3NF Intuition

No value should be able to be derived based on another non-key field

What is the redundant information in this table?

Customer ID	**First Name**	**Surname**	**Birthday**	**Age**	**Fav Color**
123	Pooja	Singh	1/4/1984	37	Blue
456	San	Zhang	3/15/2001	19	Blue
789	John	Zhang	11/12/2006	14	Buff

3NF Intuition

No value should be able to be derived based on another non-key field

What is the redundant information in this table?

Customer ID	**First Name**	**Surname**	**Birthday**	~~**Age**~~	**Fav Color**
123	Pooja	Singh	1/4/1984	~~37~~	Blue
456	San	Zhang	3/15/2001	~~19~~	Blue
789	John	Zhang	11/12/2006	~~14~~	Buff

If we know Birthday, we can calculate Age -> there is an obvious dependency between them! Can remove Age.

3NF Intuition

No value should be able to be derived based on another non-key field

What is the redundant information in this table?

Tournament	Year	Winner	Winner's Birthplace
Indiana Invitational	1998	Al Fredrickson	Ohio
Cleveland Open	1999	Bob Albertson	New York
Des Moines Masters	1999	Al Fredrickson	Ohio
Indiana Invitational	1999	Chip Masterson	Kentucky

3NF Intuition

No value should be able to be derived based on another non-key field

What is the redundant information in this table?

Tournament	Year	Winner	Winner's Birthplace
Indiana Invitational	1998	Al Fredrickson	Ohio
Cleveland Open	1999	Bob Albertson	New York
Des Moines Masters	1999	Al Fredrickson	~~Ohio~~ New Jersey
Indiana Invitational	1999	Chip Masterson	Kentucky

Updates can miss redundant information!

The {Winner's Birthplace} attribute can be determined based on Winner, which is not a Candidate Key for the table. Need to split!

3NF Intuition

No value should be able to be derived based on another non-key field

What is the redundant information in this table?

Tournament	Year	Winner
Indiana Invitational	1998	Al Fredrickson
Cleveland Open	1999	Bob Albertson
Des Moines Masters	1999	Al Fredrickson
Indiana Invitational	1999	Chip Masterson

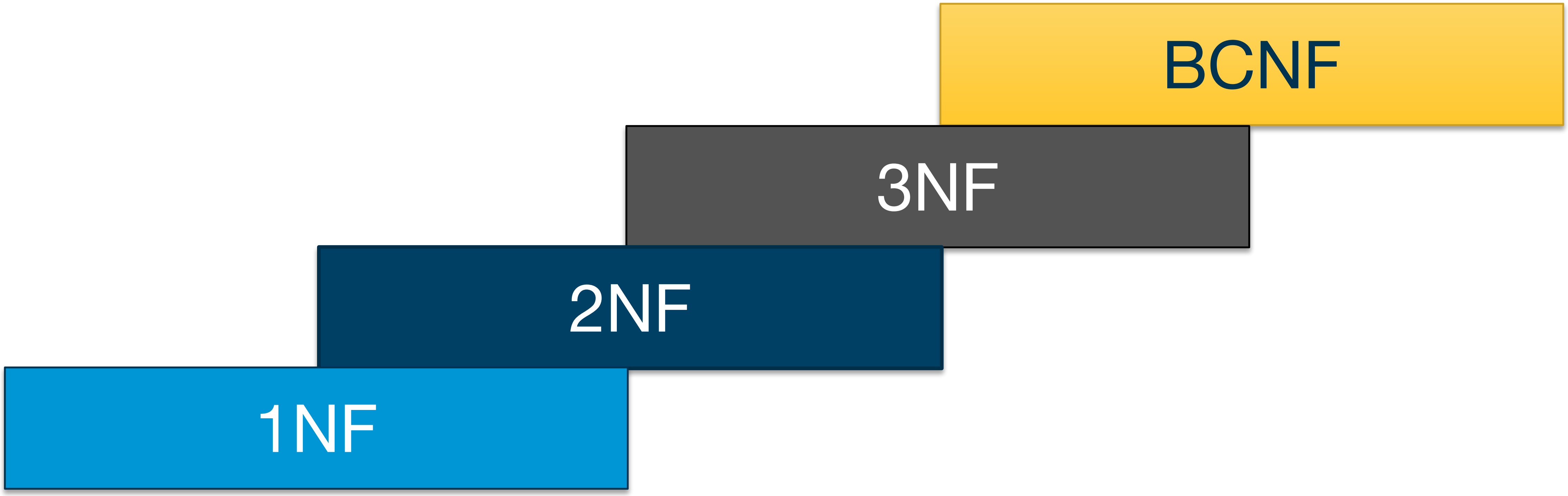
Winner	Winner's Birthplace
Bob Albertson	New York
Al Fredrickson	New Jersey
Chip Masterson	Kentucky

The {Winner’s Birthplace} attribute can be determined based on Winner, which is not a Candidate Key for the table. Need to split!

Normal Form Redundancy

1NF and 2NF - eliminate redundancy **across** rows

3NF, BCNF - also eliminate redundancy **within** rows


BCNF
3NF
2NF
1NF