UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Information Storage and Management I

Dr. Alejandro Arbelaez

Normalization of Database Tables

Data Redundancy

- Data redundancy occurs when the same piece of data is stored in two or more separate places and is a common occurrence in many businesses.
- Data redundancy in database means that some data fields are repeated in the database.
- This data repetition may occur either if a field is repeated in two or more tables or if the field is repeated within the table

Data Integrity

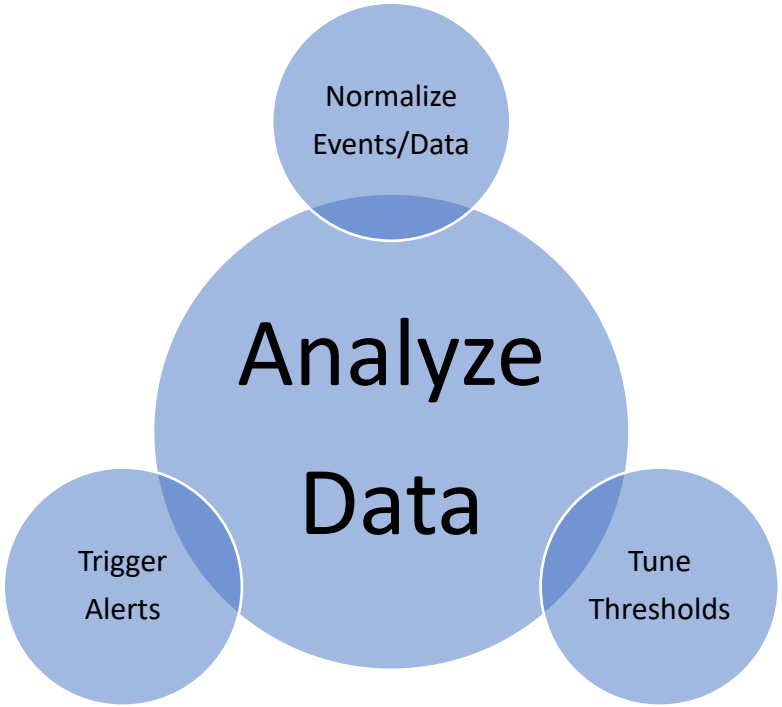
- **Data integrity** refers to maintaining and assuring the accuracy and consistency of data over its entire duration life-cycle and is a critical aspect to the design, implementation and usage of any system which stores, process, or retrieves data
- Any unintended changes to a data as a result of a storage, retrieval or processing operation, including malicious intent, unexpected hardware failure, and human error, is failure of **data integrity**

Data Integrity

- **Physical integrity:** deal with challenges associated with correctly storing and fetching the data itself
- **Logical integrity:** concerned with the correctness or rationality of a piece of data, given a particular context. This includes topics such as referential integrity and entity integrity in a relational database

Data Integrity

- MySQL allows you to add several **constraints** to a table. The primary goal of most constraints is **data integrity**. In order words, their purpose is to improve the validity and consistency of your data



Disadvantages Of Data Redundancy

- Increases the size of the database unnecessarily.
- Causes data inconsistency.
- Decreases efficiency of database.
- May cause data corruption.

Database Tables and Normalization

- Normalization is a process for assigning attributes to entities. It reduces data redundancies and helps eliminate the data anomalies.
- Probably most valuable as a way of evaluating and correcting DB design
- Normalization works through a series of stages called normal forms:
  - First normal form (1NF)
  - Second normal form (2NF)
  - Third normal form (3NF)
  - Fourth normal form (4NF)

Database Tables and Normalization

- Case of a Construction Company
  - Building project -- Project number, Name, Employees assigned to the project.

Project number
Project, name

Employees assigned to the project

Database Tables and Normalization

- Case of a Construction Company
  - Building project -- Project number, Name, Employees assigned to the project.
  - Employee -- Employee number, Name, Job classification

Employee number: 1234
Name: Homer Simpson
Job classification: Nuclear Safety Inspector

Database Tables and Normalization

- Case of a Construction Company
  - Building project -- Project number, Name, Employees assigned to the project.
  - Employee -- Employee number, Name, Job classification
  - The company charges its clients by billing the hours spent on each project. The hourly billing rate is dependent on the employee's position.

Database Tables and Normalization

- Case of a Construction Company
  - Building project -- Project number, Name, Employees assigned to the project.
  - Employee -- Employee number, Name, Job classification
  - The company charges its clients by billing the hours spent on each project. The hourly billing rate is dependent on the employee's position.
  - **Periodically, a report is generated.**

Reports

PROJ. NUM.	PROJECT NAME	EMPLOYEE NUMBER	EMPLOYEE NAME	JOB CLASS.	CHG/ HOUR	HOURS BILLED	TOTAL CHARGE
15	Evergreen	103	June E. Arbough	Elec. Engineer	$84.50	23.8	$2,011.10
		101	John G. News	Database Designer	$105.00	19.4	$2,037.00
		105	Alice K. Johnson *	Database Designer	$105.00	35.7	$3,748.50
		106	William Smithfield	Programmer	$35.75	12.6	$450.45
		102	David H. Senior	Systems Analyst	$96.75	23.8	$2,302.65
				Subtotal			**$10,549.70**
18	Amber	114	Annelise Jones	Applications Designer	$48.10	24.6	$1,183.26
	Wave	118	James J. Frommer	General Support	$18.36	45.3	$831.71
		104	Anne K. Ramoras *	Systems Analyst	$96.75	32.4	$3,134.70
		112	Darlene M. Smithson	DSS Analyst	$45.95	44.0	$2,021.80
				Subtotal			**$7,171.47**
22		105	Alice K. Johnson	Database Designer	$105.00	64.7	$6,793.50
		104	Anne K. Ramoras	Systems Analyst	$96.75	48.4	$4,682.70
		113	Delbert K. Joenbrood*	Applications Designer	$48.10	23.6	$1,135.16
		111	Geoff B. Wabash	Clerical Support	$26.87	22.0	$591.14
		106	William Smithfield	Programmer	$35.75	12.8	$457.60
				Subtotal			**$13,660.10**
25		107	Maria D. Alonzo	Programmer	$35.75	24.6	$879.45
		115	Travis B. Bawangi	Systems Analyst	$96.75	45.8	$4,431.15
		101	John G. News *	Database Designer	$105.00	56.3	$5,911.50
		114	Annelise Jones	Applications Designer	$48.10	33.1	$1,592.11
		108	Ralph B. Washington	Systems Analyst	$96.75	23.6	$2,283.30
		118	James J. Frommer	General Support	$18.36	30.5	$559.98
		112	Darlene M. Smithson	DSS Analyst	$45.95	41.4	$1,902.33
				Subtotal			**$17,559.82**
				Total			**48,941.09**

Note: * indicates project leader

A Table Whose Structure Matches the Report Format

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	$84.50	23.8
		101	John G. News	Database Designer	$105.00	19.4
		105	Alice K. Johnson *	Database Designer	$105.00	35.7
		106	William Smithfield	Programmer	$35.75	12.6
		102	David H. Senior	Systems Analyst	$96.75	23.8
18	Amber Wave	114	Annelise Jones	Applications Designer	$48.10	24.6
		118	James J. Frommer	General Support	$18.36	45.3
		104	Anne K. Ramoras *	Systems Analyst	$96.75	32.4
		112	Darlene M. Smithson	DSS Analyst	$45.95	44.0
22	Rolling Tide	105	Alice K. Johnson	Database Designer	$105.00	64.7
		104	Anne K. Ramoras	Systems Analyst	$96.75	48.4
		113	Delbert K. Joenbrood *	Applications Designer	$48.10	23.6
		111	Geoff B. Wabash	Clerical Support	$26.87	22.0
		106	William Smithfield	Programmer	$35.75	12.8
25	Starflight	107	Maria D. Alonzo	Programmer	$35.75	24.6
		115	Travis B. Bawangi	Systems Analyst	$96.75	45.8
		101	John G. News *	Database Designer	$105.00	56.3
		114	Annelise Jones	Applications Designer	$48.10	33.1
		108	Ralph B. Washington	Systems Analyst	$96.75	23.6
		118	James J. Frommer	General Support	$18.36	30.5
		112	Darlene M. Smithson	DSS Analyst	$45.95	41.4

A Table Whose Structure Matches the Report Format

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	$84.50	23.8
		101	John G. News	Database Designer	$105.00	19.4
		105	Alice K. Johnson *	Database Designer	$105.00	35.7
		106	William Smithfield	Programmer	$35.75	12.6
		102	David H. Senior	Systems Analyst	$96.75	23.8
18	Amber Wave	114	Annelise Jones	Applications Designer	$48.10	24.6
		118	James J. Frommer	General Support	$18.36	45.3
		104	Anne K. Ramoras *	Systems Analyst	$96.75	32.4
		112	Darlene M. Smithson	DSS Analyst	$45.95	44.0
22	Rolling Tide	105	Alice K. Johnson	Database Designer	$105.00	64.7
		104	Anne K. Ramoras	Systems Analyst	$96.75	48.4
		113	Delbert K. Joenbrood *	Applications Designer	$48.10	23.6
		111	Geoff B. Wabash	Clerical Support	$26.87	22.0
		106	William Smithfield	Programmer	$35.75	12.8
25	Starflight	107	Maria D. Alonzo	Programmer	$35.75	24.6
		115	Travis B. Bawangi	Systems Analyst	$96.75	45.8
		101	John G. News *	Database Designer	$105.00	56.3
		114	Annelise Jones	Applications Designer	$48.10	33.1
		108	Ralph B. Washington	Systems Analyst	$96.75	23.6
		118	James J. Frommer	General Support	$18.36	30.5
		112	Darlene M. Smithson	DSS Analyst	$45.95	41.4

Not ideal

- There is repetition of information
- Need to use null values
- Redundancy problems

Redundancy Problems

- **Update Anomalies**
  - If the project number changes, we need to make sure that we change some entries in the table

Redundancy Problems

- **Update Anomalies**
  - If the project number changes, we need to make sure that we change some entries in the table
- **Insert Anomalies**
  - May not be possible to add a new project without employees, unless we have employees in the project

Redundancy Problems

- **Update Anomalies**
  - If the project number changes, we need to make sure that we change some entries in the table
- **Insert Anomalies**
  - May not be possible to add a new project without employees, unless we have employees in the project
- **Delete Anomalies**
  - If all employees that work in a project are deleted, then we lose the project number

Unnormalized Form (UNF)

- A table that contains one or more repeating groups.
- To create an unnormalized table:
  - Transform the data from the information source (e.g. form) into table format with columns and rows.

University Example
Unnormalized Form (UNF)

student(sid, cid, room, grade, name, address)

sid	cid	room	grade	name	address
123	15-445	GHC 6115	A	Andy	Pittsburgh
456	15-721	GHC 8102	B	Tupac	Los Angeles
789	15-445	GHC 6115	A	Obama	Chicago
012	15-445	GHC 6115	C	Weka Flocka	Atlanta
789	15-721	GHC8102	A	Obama	Chicago

University Example
Unnormalized Form (UNF)

student(sid, cid, room, grade, name, address)

sid	cid	room	grade	name	address
123	15-445	GHC 6115	A	Andy	Pittsburgh
456	15-721	GHC 8102	B	Tupac	Los Angeles
789	15-445	GHC 6115	A	Obama	Chicago
012	15-445	GHC 6115	C	Weka Flocka	Atlanta
789	15-721	GHC8102	A	Obama	Chicago

University Example
Unnormalized Form (UNF)

student(sid, cid, room, grade, name, address)

sid	cid	room	grade	name	address
123	15-445	GHC 6115	A	Andy	Pittsburgh
456	15-721	GHC 8102	B	Tupac	Los Angeles
789	15-445	GHC 6115	A	Obama	Chicago
012	15-445	GHC 6115	C	Weka Flocka	Atlanta
789	15-721	GHC8102	A	Obama	Chicago

University Example
Unnormalized Form (UNF)

student(sid, cid, room, grade, name, address)

sid	cid	room	grade	name	address
123	15-445	GHC 6115	A	Andy	Pittsburgh
456	15-721	GHC 8102	B	Tupac	Los Angeles
789	15-445	GHC 6115	A	Obama	Chicago
012	15-445	GHC 6115	C	Weka Flocka	Atlanta
789	15-721	GHC8102	A	Obama	Chicago

Update Anomalies
If the room number changes, we need to make sure that we change all student records

University Example
Unnormalized Form (UNF)

student(<u>sid</u>, <u>cid</u>, room, grade, name, address)

sid	cid	room	grade	name	address
123	15-445	GHC 6115	A	Andy	Pittsburgh
456	15-721	GHC 8102	B	Tupac	Los Angeles
789	15-445	GHC 6115	A	Obama	Chicago
012	15-445	GHC 6115	C	Weka Flocka	Atlanta
789	15-721	GHC8102	A	Obama	Chicago

Update Anomalies
If the room number changes, we need to make sure that we change all student records

Insert Anomalies
May not be possible to add a student unless they are enrolled in a course

University Example
Unnormalized Form (UNF)

student(sid, cid, room, grade, name, address)

sid	cid	room	grade	name	address
123	15-445	GHC 6115	A	Andy	Pittsburgh
456	15-721	GHC 8102	B	Tupac	Los Angeles
789	15-445	GHC 6115	A	Obama	Chicago
012	15-445	GHC 6115	C	Weka Flocka	Atlanta
789	15-721	GHC8102	A	Obama	Chicago

Update Anomalies
If the room number changes, we need to make sure that we change all student records

Insert Anomalies
May not be possible to add a student unless they are enrolled in a course

Delete Anomalies
If all students enrolled in a course are deleted, then we lose the room number

Database Tables and Normalization

- Problems with the design based on report Handout
  - Just doesn't fit in a Relational DB – not a table
  - The student number is intended to be part of a primary key, but it contains nulls.
  - The table displays data **redundancies**.
  - The table entries invite data **inconsistencies**.
  - The data redundancies yield the following anomalies:
    - Update anomalies.
    - Addition anomalies.
    - Deletion anomalies.

University Database

Why this decomposition is better and how to find it

Why this decomposition is better?

student(<u>sid</u>, name, address)

sid	name	address
123	Andy	Pittsburgh
456	Tupac	Los Angeles
789	Obama	Chicago
012	Weka Flocka	Atlanta

rooms(<u>cid</u>, room)

cid	room
15-415	GHC 615
15-721	GHC 8102

Course(<u>sid</u>, <u>cid</u>, name, address)

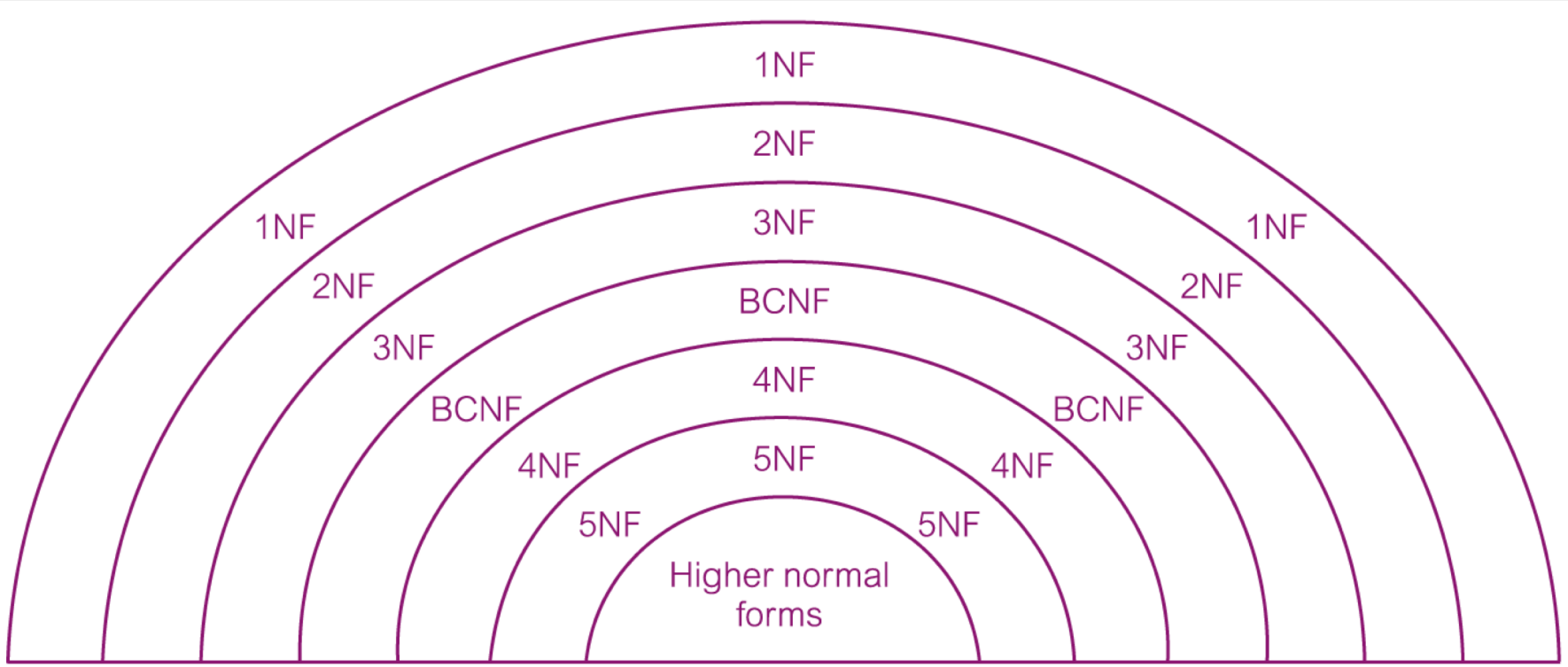
sid	cid	grade
123	15-415	A
456	15-721	B
789	15-415	A
012	15-415	C
789	15-721	A

The Normalization Process (continued)

NORMAL FORM	CHARACTERISTIC
First normal form (1NF)	Table format; no repeating groups and PK identified
Second normal form (2NF)	1NF and no partial dependencies
Third normal form (3NF)	2NF and no transitive dependencies
Boyce-Codd normal form (BCNF)	Every determinant is a candidate key (special case of 3NF)
Fourth normal form (4NF)	3NF and no independent multivalued dependencies

The Process of Normalization

As normalization proceeds, the relations become progressively more restricted (stronger) in format and also less vulnerable to update anomalies.



Functional Dependencies

- A **functional dependency** (FD) is a form of a constraint and part of a relation's schema to define a valid instance

- Definition: **A** → **B**
- The value of **A** functionally defines the value of **B**

> We say an attribute, B, has a ***functional dependency*** on another attribute, A, if for any two records, which have the same value for A, then the values for B in these two records must be the same. We illustrate this as:

Functional Dependencies

Formal Definition:

- **X → Y** $\Rightarrow(t_i[x] = t_j[x] \Rightarrow t_i[y] = t_j[y])$
- If any two tuples (t_i, t_j) agree on the X attribute, then they must agree on the Y attribute too

Functional Dependencies

Forma Definition:

- **X → Y** $\Rightarrow(t_i[x] = t_j[x] \Rightarrow t_i[y] = t_j[y])$
- If any two tuples (t_i, t_j) agree on the X attribute, then they must agree on the Y attribute too

sid	cid	room	grade	name	address
123	15-445	GHC 6115	A	Andy	Pittsburgh
456	15-721	GHC 8102	B	Tupac	Los Angeles
789	15-445	GHC 6115	A	Obama	Chicago
012	15-445	GHC 6115	C	Weka Flocka	Atlanta
789	15-721	GHC8102	A	Obama	Chicago

X Y

cid → room ?

Functional Dependencies

Forma Definition:

- **X → Y** $\Rightarrow(t_i[x] = t_j[x] \Rightarrow t_i[y] = t_j[y])$
- If any two tuples (t_i, t_j) agree on the X attribute, then they must agree on the Y attribute too

sid	cid	room	grade	name	address
123	15-445	GHC 6115	A	Andy	Pittsburgh
456	15-721	GHC 8102	B	Tupac	Los Angeles
789	15-445	GHC 6115	A	Obama	Chicago
012	15-445	GHC 6115	C	Weka Flocka	Atlanta
789	15-721	GHC8102	A	Obama	Chicago

X Y

cid → room ☺

Functional Dependencies

Forma Definition:

- **X → Y** $\Rightarrow(t_i[x] = t_j[x] \Rightarrow t_i[y] = t_j[y])$
- If any two tuples (t_i, t_j) agree on the X attribute, then they must agree on the Y attribute too

sid	cid	room	grade	name	address
123	15-445	GHC 6115	A	Andy	Pittsburgh
456	15-721	GHC 8102	B	Tupac	Los Angeles
789	15-445	GHC 6115	A	Obama	Chicago
012	15-445	GHC 6115	C	Weka Flocka	Atlanta
789	15-721	GHC8102	A	Obama	Chicago
		X	Y		

cid → grade?

Functional Dependencies

Forma Definition:

- **X → Y** $\Rightarrow(t_i[x] = t_j[x] \Rightarrow t_i[y] = t_j[y])$
- If any two tuples (t_i, t_j) agree on the X attribute, then they must agree on the Y attribute too

sid	cid	room	grade	name	address
123	15-445	GHC 6115	A	Andy	Pittsburgh
456	15-721	GHC 8102	B	Tupac	Los Angeles
789	15-445	GHC 6115	A	Obama	Chicago
012	15-445	GHC 6115	C	Weka Flocka	Atlanta
789	15-721	GHC8102	A	Obama	Chicago
		X	Y		

cid → grade ☹

Not a valid FD

Functional Dependencies

Forma Definition:

- **X → Y** $\Rightarrow(t_i[x] = t_j[x] \Rightarrow t_i[y] = t_j[y])$
- If any two tuples (t_i, t_j) agree on the X attribute, then they must agree on the Y attribute too

R1(<u>sid</u>, name, address)

sid	name	address
123	Andy	Pittsburgh
456	Tupac	Los Angeles
789	Obama	Chicago
012	Weka Flocka	Atlanta

X Y

sid → name ☺

Functional Dependencies

FD is a constraint that allows instances for which the FD holds

You can check if a FD is violated by an instance, but you cannot prove that an FD is a part of the schema using an instance

R1(sid, name, address)

sid	name	address
123	Andy	Pittsburgh
456	Tupac	Los Angeles
789	Obama	Chicago
012	Weka Flocka	Atlanta
555	Andy	Providence

What about
name → address ?

Functional Dependencies

FD is a constraint that allows instances for which the FD holds

You can check if a FD is violated by an instance, but you cannot prove that an FD is a part of the schema using an instance

R1(sid, name, address)

sid	name	address
123	Andy	Pittsburgh
456	Tupac	Los Angeles
789	Obama	Chicago
012	Weka Flocka	Atlanta
555	Andy	Providence

What about
name → address ? **Nope**

Functional Dependencies

- Two FDs A→B and A→C can be written in shorthand as A→BC

- But AB →C is not the same as the two FDs A→C and B→C
  - Be really careful with this

FDs

- Given this table let's compute FDs

student(sid, cid, room, grade, name, address)

sid	cid	room	grade	name	address
123	15-445	GHC 6115	A	Andy	Pittsburgh
456	15-721	GHC 8102	B	Tupac	Los Angeles
789	15-445	GHC 6115	C	Obama	Chicago
012	15-445	GHC 6115	A	Weka Flocka	Atlanta

FDs
?

FDs

- Given this table let's compute FDs

student(sid, cid, room, grade, name, address)

sid	cid	room	grade	name	address
123	15-445	GHC 6115	A	Andy	Pittsburgh
456	15-721	GHC 8102	B	Tupac	Los Angeles
789	15-445	GHC 6115	C	Obama	Chicago
012	15-445	GHC 6115	A	Weka Flocka	Atlanta

FDs
sid → name

FDs

- Given this table let's compute FDs

student(sid, cid, room, grade, name, address)

sid	cid	room	grade	name	address
123	15-445	GHC 6115	A	Andy	Pittsburgh
456	15-721	GHC 8102	B	Tupac	Los Angeles
789	15-445	GHC 6115	C	Obama	Chicago
012	15-445	GHC 6115	A	Weka Flocka	Atlanta

FDs
sid → name
sid → address

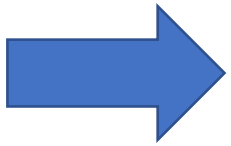
FDs

- Given this table let's compute some FDs

student(sid, cid, room, grade, name, address)

sid	cid	room	grade	name	address
123	15-445	GHC 6115	A	Andy	Pittsburgh
456	15-721	GHC 8102	B	Tupac	Los Angeles
789	15-445	GHC 6115	C	Obama	Chicago
012	15-445	GHC 6115	A	Weka Flocka	Atlanta

FDs
sid → name
sid → address



FDs
sid → name, address

FDs

- Given this table let's compute some FDs

student(sid, cid, room, grade, name, address)

sid	cid	room	grade	name	address
123	15-445	GHC 6115	A	Andy	Pittsburgh
456	15-721	GHC 8102	B	Tupac	Los Angeles
789	15-445	GHC 6115	C	Obama	Chicago
012	15-445	GHC 6115	A	Weka Flocka	Atlanta

FDs
sid → name, address
cid → room

FDs

- Given this table let's compute some FDs

student(sid, cid, room, grade, name, address)

sid	cid	room	grade	name	address
123	15-445	GHC 6115	A	Andy	Pittsburgh
456	15-721	GHC 8102	B	Tupac	Los Angeles
789	15-445	GHC 6115	C	Obama	Chicago
012	15-445	GHC 6115	A	Weka Flocka	Atlanta

FDs
sid → name, address
cid → room
cid → grade?

FDs

- Given this table let's compute some FDs

student(sid, cid, room, grade, name, address)

sid	cid	room	grade	name	address
123	15-445	GHC 6115	A	Andy	Pittsburgh
456	15-721	GHC 8102	B	Tupac	Los Angeles
789	15-445	GHC 6115	C	Obama	Chicago
012	15-445	GHC 6115	A	Weka Flocka	Atlanta

FDs
sid → name, address
cid → room
cid → grade? Is not a valid FD

FDs

- Given this table let's compute some FDs

student(<u>sid</u>, <u>cid</u>, room, grade, name, address)

sid	cid	room	grade	name	address
123	15-445	GHC 6115	A	Andy	Pittsburgh
456	15-721	GHC 8102	B	Tupac	Los Angeles
789	15-445	GHC 6115	C	Obama	Chicago
012	15-445	GHC 6115	A	Weka Flocka	Atlanta

FDs
sid → name, address
cid → room
cid → grade? is not a valid FD

But
sid, cid → grade
Is a valid FD

Keys

- **Superkey**: SK, a subset of attributes of a relation satisfying uniqueness, that is, no two tuples have the same combination of values for these attributes
- **Candidate key**: K, a super key SK, satisfying minimality, that is, no component of K can be eliminated without destroying the uniqueness property
- **Primary key**: PK, the selected candidate key K
- **Prime attributes**: attributes that constitute the primary key (or candidate key)

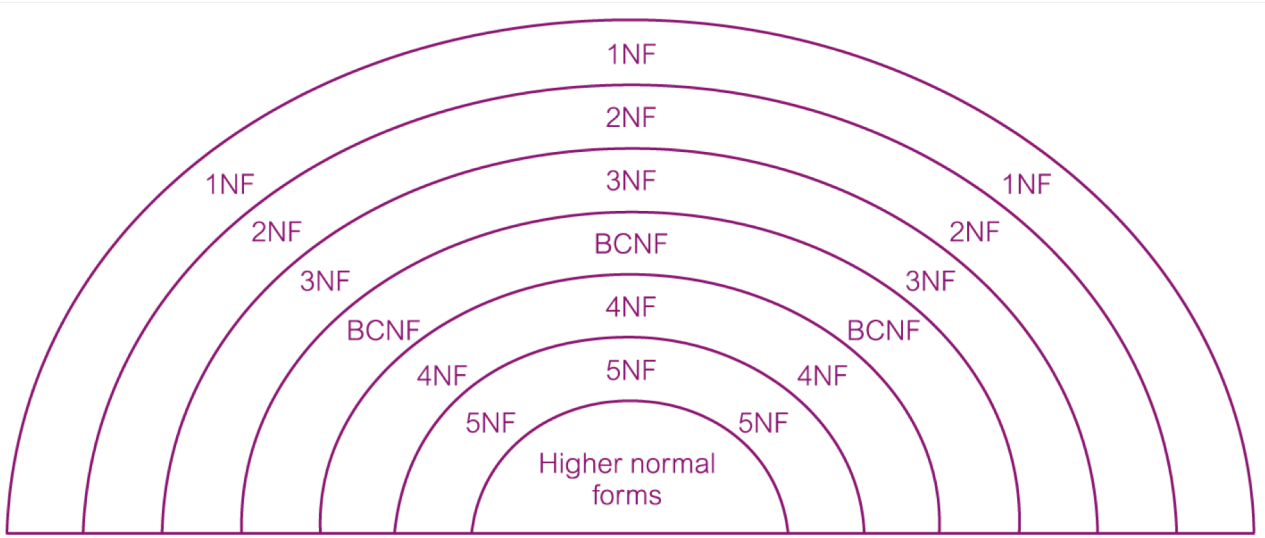
Keys

student(<u>sid</u>, <u>cid</u>, room, grade, name, address)

sid	cid	room	grade	name	address
123	15-445	GHC 6115	A	Andy	Pittsburgh
456	15-721	GHC 8102	B	Tupac	Los Angeles
789	15-445	GHC 6115	C	Obama	Chicago
012	15-445	GHC 6115	A	Weka Flocka	Atlanta

- Superkey: {sid} and {sid, cid}
- Candidate key: {sid}
- Non-prime attributes: cid, room, grade, name, address

Normal Forms



First Normal Form (1NF)

- All types must be atomic
- This means that table entries should be single values, not sets or composite objects (or **repeating groups**)

Example

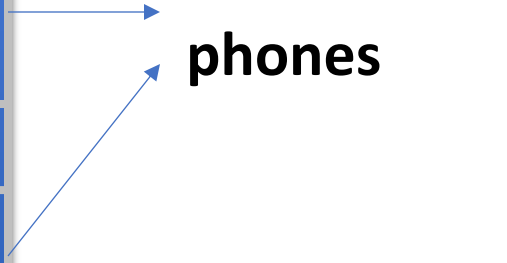
Suppose a company wants to store the names and contact details of its employees. It creates a table that looks like this:

emp_id	emp_name	emp_address	emp_mobile
101	Herschel	New Delhi	8912312390
102	Jon	Kanpur	8812121212, 9900012222
103	Ron	Chennai	7778881212
104	Lester	Bangalore	9990000123, 8123450987

Example

Suppose a company wants to store the names and contact details of its employees. It creates a table that looks like this:

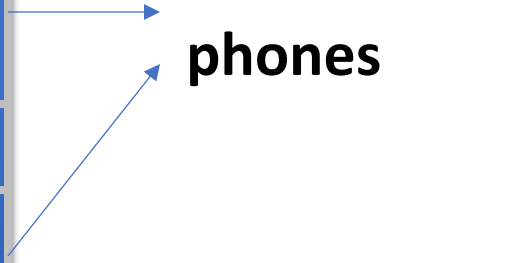
emp_id	emp_name	emp_address	emp_mobile
101	Herschel	New Delhi	8912312390
102	Jon	Kanpur	8812121212, 9900012222
103	Ron	Chennai	7778881212
104	Lester	Bangalore	9990000123, 8123450987



Example

This table is not in 1NF → Each attribute must have atomic (single) values

emp_id	emp_name	emp_address	emp_mobile
101	Herschel	New Delhi	8912312390
102	Jon	Kanpur	8812121212, 9900012222
103	Ron	Chennai	7778881212
104	Lester	Bangalore	9990000123, 8123450987



What about this solution?

emp_id	emp_name	emp_address	emp_mobile1	emp_mobile2
101	Herschel	New Delhi	8912312390	
102	Jon	Kanpur	8812121212	9900012222
103	Ron	Chennai	7778881212	
104	Lester	Bangalore	9990000123	8123450987

What about this solution?

but then….

What about this solution?

emp_id	emp_name	emp_address	emp_mobile1	emp_mobile2	…	emp_mobileN
101	Herschel	New Delhi				
102	Jon	Kanpur				
103	Ron	Chennai				
104	Lester	Bangalore				

What about this solution? ☹

Repeating group: the two telephone number columns still form a "repeating group": they repeat what is conceptually the same attribute, namely a telephone number.

emp_id	emp_name	emp_address	emp_mobile1	emp_mobile2	…	emp_mobileN
101	Herschel	New Delhi				
102	Jon	Kanpur				
103	Ron	Chennai				
104	Lester	Bangalore				

What about this solution? Not a valid 1NF ☹

Repeating group: the two telephone number columns still form a "repeating group": they repeat what is conceptually the same attribute, namely a telephone number.

emp_id	emp_name	emp_address	emp_mobile1	emp_mobile2	…	emp_mobileN
101	Herschel	New Delhi				
102	Jon	Kanpur				
103	Ron	Chennai				
104	Lester	Bangalore				

What about this solution? -- Valid 1NF ☺

- We split the strings we used to hold our telephone number information into "atomic" (i.e. indivisible) entities Single phone numbers.
- We ensure no row contains more than one phone number.

emp_id	emp_name	emp_address	emp_mobile
101	Herschel	New Delhi	8912312390
102	Jon	Kanpur	8812121212
102	Jon	Kanpur	9900012222
103	Ron	Chennai	7778881212
104	Lester	Bangalore	9990000123
104	Lester	Bangalore	8123450987

University Example -- UNF to 1NF

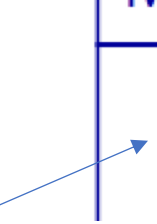
Unnormalised

Module	Dept	Lecturer	Texts
M1	D1	L1	T1, T2
M2	D1	L1	T1, T3
M3	D1	L2	T4
M4	D2	L3	T1, T5
M5	D2	L4	T6

University Example -- UNF to 1NF

Unnormalised

Module	Dept	Lecturer	Texts
M1	D1	L1	T1, T2
M2	D1	L1	T1, T3
M3	D1	L2	T4
M4	D2	L3	T1, T5
M5	D2	L4	T6



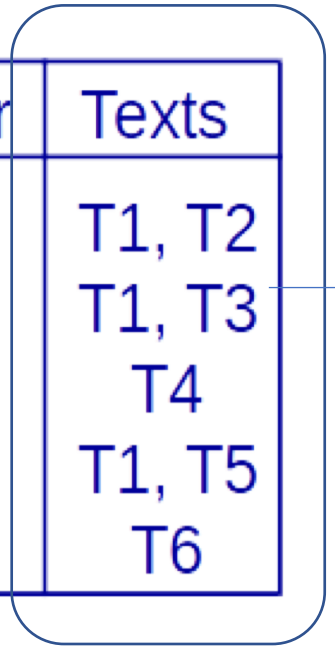
1NF

Module	Dept	Lecturer	Text
M1	D1	L1	T1
M1	D1	L1	T2

University Example -- UNF to 1NF

Unnormalised

Module	Dept	Lecturer	Texts
M1	D1	L1	T1, T2
M2	D1	L1	T1, T3
M3	D1	L2	T4
M4	D2	L3	T1, T5
M5	D2	L4	T6



1NF

Module	Dept	Lecturer	Text
M1	D1	L1	T1
M1	D1	L1	T2
M2	D1	L1	T1
M2	D1	L1	T3

University Example -- UNF to 1NF

Unnormalised

Module	Dept	Lecturer	Texts
M1	D1	L1	T1, T2
M2	D1	L1	T1, T3
M3	D1	L2	T4
M4	D2	L3	T1, T5
M5	D2	L4	T6

1NF

Module	Dept	Lecturer	Text
M1	D1	L1	T1
M1	D1	L1	T2
M2	D1	L1	T1
M2	D1	L1	T3
M3	D1	L2	T4
M4	D2	L3	T1
M4	D2	L3	T5
M5	D2	L4	T6

Problems in 1NF

1NF

Module	Dept	Lecturer	Text
M1	D1	L1	T1
M1	D1	L1	T2
M2	D1	L1	T1
M2	D1	L1	T3
M3	D1	L2	T4
M4	D2	L3	T1
M4	D2	L3	T5
M5	D2	L4	T6

INSERT anomalies:

- Can't add a module with no texts

UPDATE anomalies:

- To change lecturer for M1, we have to change two rows

DELETE anomalies:

- If we remove M3, we remove L2 as well

Example - University Marks (UNF)

C = Course; Cn = C1, C2 etc; S = Student number, Sn = S1,S2. etc. Lname = Lecturer name.

Data Format 1a

C1	Database		B.Treacy	B223l
	S100	J.Bloggs 1, Kenley, Cork		40
	S102	J.Murphy 3 Parchment Sq,Cork		45
C2	Modelling		M.Davin	F2.1
	S100	J.Bloggs 1, Kenley, Cork		60
	S102	J.Murphy 3 Parchment Sq,Cork		65
C3	Maths		A.Tobin	B190
	S104	F. O'Shea 5 Wilton Rd		80

Data Format 1b

S100	J.Bloggs	1, Kenley, Cork			
	C1	Database	B.Treacy	B223l	40
	C2	Modelling	M.Davin	F2.1	60
S102	J.Murphy	3 Parchment Sq,Cork			
	C1	Database	B.Treacy	B223l	45
	C2	Modelling	M.Davin	F2.1	65
S104	F.O'Shea	5 Wilton Rd			
	C3	Maths	A.Tobin	B190	80

Example - College Marks (1NF)

S#	Name	Address	C#	Title	Lname	Room#	Mark
S100	J.Bloggs	1, Kenley, Cork	C1	Database	B.Treacy	B223L	40
S102	J.Murphy	3 Parchment Sq,Cork	C1	Database	B. Treacy	B223L	45
S100	J.Bloggs	1, Kenley, Cork	C2	Modelling	M. Davin	F2.1	60
S102	J.Murphy	3 Parchment Sq, Cork	C2	Modelling	M. Davin	F2.1	65
S104	F.OShea	5 Wilton Rd, Cork	C3	Maths	A.Tobin	B190	80

Identify Functional Dependencies (given this table):

Example - College Marks (1NF)

S#	Name	Address	C#	Title	Lname	Room#	Mark
S100	J.Bloggs	1, Kenley, Cork	C1	Database	B.Treacy	B223L	40
S102	J.Murphy	3 Parchment Sq,Cork	C1	Database	B. Treacy	B223L	45
S100	J.Bloggs	1, Kenley, Cork	C2	Modelling	M. Davin	F2.1	60
S102	J.Murphy	3 Parchment Sq, Cork	C2	Modelling	M. Davin	F2.1	65
S104	F.OShea	5 Wilton Rd, Cork	C3	Maths	A.Tobin	B190	80

Identify Functional Dependencies (given this table):

{S#} → {Name, Address}

{C#} → {Title,Lname,Room#}

{S#,C#} → {Mark}

S# → Lname ?

Example - College Marks (1NF)

S#	Name	Address	C#	Title	Lname	Room#	Mark
S100	J.Bloggs	1, Kenley, Cork	C1	Database	B.Treacy	B223L	40
S102	J.Murphy	3 Parchment Sq,Cork	C1	Database	B. Treacy	B223L	45
S100	J.Bloggs	1, Kenley, Cork	C2	Modelling	M. Davin	F2.1	60
S102	J.Murphy	3 Parchment Sq, Cork	C2	Modelling	M. Davin	F2.1	65
S104	F.OShea	5 Wilton Rd, Cork	C3	Maths	A.Tobin	B190	80

Identify Functional Dependencies (given this table):

{S#} → {Name, Address}
{C#} → {Title,Lname,Room#}
{S#,C#} → {Mark}

S# → Lname is not a valid FD
Tuple-1 S# 100 → Lname B.Treacy
Tuple-3 S# 100 → Lname M. Davin

Example - College Marks (1NF)

S#	Name	Address	C#	Title	Lname	Room#	Mark
S100	J.Bloggs	1, Kenley, Cork	C1	Database	B.Treacy	B223L	40
S102	J.Murphy	3 Parchment Sq,Cork	C1	Database	B. Treacy	B223L	45
S100	J.Bloggs	1, Kenley, Cork	C2	Modelling	M. Davin	F2.1	60
S102	J.Murphy	3 Parchment Sq, Cork	C2	Modelling	M. Davin	F2.1	65
S104	F.OShea	5 Wilton Rd, Cork	C3	Maths	A.Tobin	B190	80

Identify Functional Dependencies (given this table):

{S#} → {Name, Address}

{C#} → {Title,Lname,Room#}

{S#,C#} → {Mark}

What is the primary key? (1NF Solution)

Example - College Marks (1NF)

S#	Name	Address	C#	Title	Lname	Room#	Mark
S100	J.Bloggs	1, Kenley, Cork	C1	Database	B.Treacy	B223L	40
S102	J.Murphy	3 Parchment Sq,Cork	C1	Database	B. Treacy	B223L	45
S100	J.Bloggs	1, Kenley, Cork	C2	Modelling	M. Davin	F2.1	60
S102	J.Murphy	3 Parchment Sq, Cork	C2	Modelling	M. Davin	F2.1	65
S104	F.OShea	5 Wilton Rd, Cork	C3	Maths	A.Tobin	B190	80

Identify Functional Dependencies (given this table):

{S#} → {Name, Address}

{C#} → {Title,Lname,Room#}

{S#,C#} → {Mark}

What is the primary key? (1NF Solution)

S#, C#, Name Address, Title, Lname, Room#, Mark

Example - College Marks (1NF)

S#	Name	Address	C#	Title	Lname	Room#	Mark
S100	J.Bloggs	1, Kenley, Cork	C1	Database	B.Treacy	B223L	40
S102	J.Murphy	3 Parchment Sq,Cork	C1	Database	B. Treacy	B223L	45
S100	J.Bloggs	1, Kenley, Cork	C2	Modelling	M. Davin	F2.1	60
S102	J.Murphy	3 Parchment Sq, Cork	C2	Modelling	M. Davin	F2.1	65
S104	F.OShea	5 Wilton Rd, Cork	C3	Maths	A.Tobin	B190	80

1NF Solution

S#, C#, Name, Address,Title, Lname, Room#, Mark

If the PK is (S#, C#), can you identify Partial dependencies?

Example - College Marks (1NF)

S#	Name	Address	C#	Title	Lname	Room#	Mark
S100	J.Bloggs	1, Kenley, Cork	C1	Database	B.Treacy	B223L	40
S102	J.Murphy	3 Parchment Sq,Cork	C1	Database	B. Treacy	B223L	45
S100	J.Bloggs	1, Kenley, Cork	C2	Modelling	M. Davin	F2.1	60
S102	J.Murphy	3 Parchment Sq, Cork	C2	Modelling	M. Davin	F2.1	65
S104	F.OShea	5 Wilton Rd, Cork	C3	Maths	A.Tobin	B190	80

1NF Solution

S#, C#, Name, Address,Title, Lname, Room#, Mark

If the PK is (S#, C#), can you identify Partial dependencies?

Partial Dependency occurs when a non-prime attribute is functionally dependent on part of a candidate key

Example - College Marks (1NF)

S#	Name	Address	C#	Title	Lname	Room#	Mark
S100	J.Bloggs	1, Kenley, Cork	C1	Database	B.Treacy	B223L	40
S102	J.Murphy	3 Parchment Sq,Cork	C1	Database	B. Treacy	B223L	45
S100	J.Bloggs	1, Kenley, Cork	C2	Modelling	M. Davin	F2.1	60
S102	J.Murphy	3 Parchment Sq, Cork	C2	Modelling	M. Davin	F2.1	65
S104	F.OShea	5 Wilton Rd, Cork	C3	Maths	A.Tobin	B190	80

1NF Solution

S#, C#, Name, Address,Title, Lname, Room#, Mark

If the PK is (S#, C#), can you identify Partial dependencies?

Name, Address, C#,
Title, Lname, Room#, Mark

Partial Dependency occurs when a non-prime attribute is functionally dependent on part of a candidate key

Example - College Marks (1NF)

S#	Name	Address	C#	Title	Lname	Room#	Mark
S100	J.Bloggs	1, Kenley, Cork	C1	Database	B.Treacy	B223L	40
S102	J.Murphy	3 Parchment Sq,Cork	C1	Database	B. Treacy	B223L	45
S100	J.Bloggs	1, Kenley, Cork	C2	Modelling	M. Davin	F2.1	60
S102	J.Murphy	3 Parchment Sq, Cork	C2	Modelling	M. Davin	F2.1	65
S104	F.OShea	5 Wilton Rd, Cork	C3	Maths	A.Tobin	B190	80

1NF Solution

S#, C#, Name, Address,Title, Lname, Room#, Mark

If the PK is (S#, C#), can you identify Partial dependencies?

S# → Name ?

Name, Address, C#,
Title, Lname, Room#, Mark

Partial Dependency occurs when a non-prime attribute is functionally dependent on part of a candidate key

Example - College Marks (1NF)

S#	Name	Address	C#	Title	Lname	Room#	Mark
S100	J.Bloggs	1, Kenley, Cork	C1	Database	B.Treacy	B223L	40
S102	J.Murphy	3 Parchment Sq,Cork	C1	Database	B. Treacy	B223L	45
S100	J.Bloggs	1, Kenley, Cork	C2	Modelling	M. Davin	F2.1	60
S102	J.Murphy	3 Parchment Sq, Cork	C2	Modelling	M. Davin	F2.1	65
S104	F.OShea	5 Wilton Rd, Cork	C3	Maths	A.Tobin	B190	80

1NF Solution

S#, C#, Name, Address,Title, Lname, Room#, Mark

If the PK is (S#, C#), can you identify Partial dependencies?

S# → Name ✓

Name, Address, C#,
Title, Lname, Room#, Mark

Partial Dependency occurs when a non-prime attribute is functionally dependent on part of a candidate key

Example - College Marks (1NF)

S#	Name	Address	C#	Title	Lname	Room#	Mark
S100	J.Bloggs	1, Kenley, Cork	C1	Database	B.Treacy	B223L	40
S102	J.Murphy	3 Parchment Sq,Cork	C1	Database	B. Treacy	B223L	45
S100	J.Bloggs	1, Kenley, Cork	C2	Modelling	M. Davin	F2.1	60
S102	J.Murphy	3 Parchment Sq, Cork	C2	Modelling	M. Davin	F2.1	65
S104	F.OShea	5 Wilton Rd, Cork	C3	Maths	A.Tobin	B190	80

1NF Solution

S#, C#, Name, Address,Title, Lname, Room#, Mark

If the PK is (S#, C#), can you identify Partial dependencies?

{S#} → {Name, Address}

{C#} →{Title, Lname, Room#}

Example - College Marks (1NF)

S#	Name	Address	C#	Title	Lname	Room#	Mark
S100	J.Bloggs	1, Kenley, Cork	C1	Database	B.Treacy	B223L	40
S102	J.Murphy	3 Parchment Sq,Cork	C1	Database	B. Treacy	B223L	45
S100	J.Bloggs	1, Kenley, Cork	C2	Modelling	M. Davin	F2.1	60
S102	J.Murphy	3 Parchment Sq, Cork	C2	Modelling	M. Davin	F2.1	65
S104	F.OShea	5 Wilton Rd, Cork	C3	Maths	A.Tobin	B190	80

Can you Identify Insert, Update, Delete anomalies?

Example - College Marks (1NF)

S#	Name	Address	C#	Title	Lname	Room#	Mark
S100	J.Bloggs	1, Kenley, Cork	C1	Database	B.Treacy	B223L	40
S102	J.Murphy	3 Parchment Sq,Cork	C1	Database	B. Treacy	B223L	45
S100	J.Bloggs	1, Kenley, Cork	C2	Modelling	M. Davin	F2.1	60
S102	J.Murphy	3 Parchment Sq, Cork	C2	Modelling	M. Davin	F2.1	65
S104	F.OShea	5 Wilton Rd, Cork	C3	Maths	A.Tobin	B190	80

Can you Identify Insert, Update, Delete anomalies?

INSERT anomalies:

- Can't add courses until a student enrolls on a course

UPDATE anomalies:

- To change C2 title, we have to change two rows

DELETE anomalies:

- If we remove S104, we remove the Maths course information