Databases

Functional Dependencies and Database Normalization

João R. Campos

Bachelor in Informatics Engineering
Department of Informatics Engineering
University of Coimbra
2024/2025

From Previous Lesson(s)…

- Conceptual-design, logical-design, physical-design
- Entity-relationship (E-R) data model
- Entity and entity set
- Relationship and relationship set
- Cardinality and participation
- E-R diagram
- E-R diagram to relational schemas

Outline

- Functional Dependencies
  - Functional Dependency Diagrams
  - Inference Rules
  - Minimum Coverage
- Normal Forms
  - Redundant and Duplicated Data
  - 1st Normal Form (1NF), 2nd Normal Form (2NF), 3rd Normal Form (3NF)
  - Boyce-Codd Normal Form (BCNF)
- Decomposition using Functional Dependencies

Register your presence at UCStudent

What is the Problem?

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

Lots of Redundancy!

Source: A. Silberschatz, H. F. Korth and S. Sudarshan, "Database System Concepts", McGraw-Hill Education, Seventh Edition, 2019.

Design Alternatives

- There may be several design alternatives for the same problem
- We must ensure that we avoid two major pitfalls:
  - Redundancy: a bad design may result in repeated information
  - Incompleteness: a bad design may make certain aspects of the enterprise difficult or impossible to model
- Avoiding bad designs is not enough!
  - There may be a set of good designs from which we must choose

Database Design Approaches

- Entity-Relationship Model (top-down):
  - Entities and entity sets
  - Relationships and relationship sets
  - Cardinalities and participation
  - Relational schema

Large Projects

Studied in previous lessons

- Normalization Theory (bottom-up):
  - Universal relation and primary key
  - Functional dependencies analysis
  - Normal forms
  - Relational schema

Small Projects

This lesson!

Recommended: E-R + Normalization

- Design the Entity-Relationship (E-R) diagram
  - Entities and entity sets
  - Relationships and relationship sets
  - Cardinalities and participation
  - Obtain a first relational schema

- Improve the schema using functional dependencies and normalization
  - Identify the relations that need improvement or further refinement
  - Identify functional dependencies
  - Apply normal forms
  - Obtain the final relational schema!

Databases

Functional Dependency
Functional Dependency Diagrams
Inference Rules
Minimum Coverage

FUNCTIONAL DEPENDENCIES

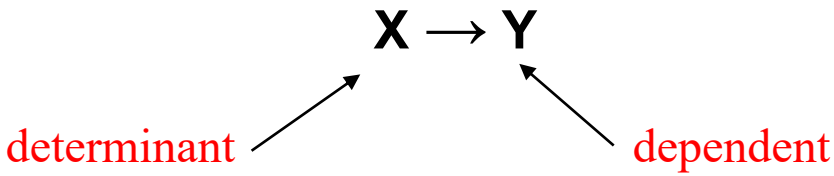
Functional Dependency

- Given a relation R and sets of attributes X, Y ⊆ R…
- … X is said to functionally determine Y if and only if each X value in R is associated with precisely one Y value in R
- … Y is functionally dependent on X, if for every valid instance of X, that value of X uniquely determines the value of Y

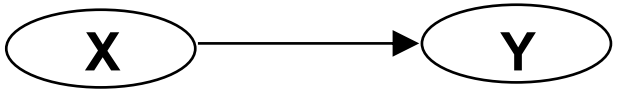
  written as: X → Y
- For example:
  - ID → PERSON NAME
  - DEP. NAME → BUILDING
  - COURSE ID → TITLE
  - PERSON NAME → ID ?

Functional Dependency Diagrams

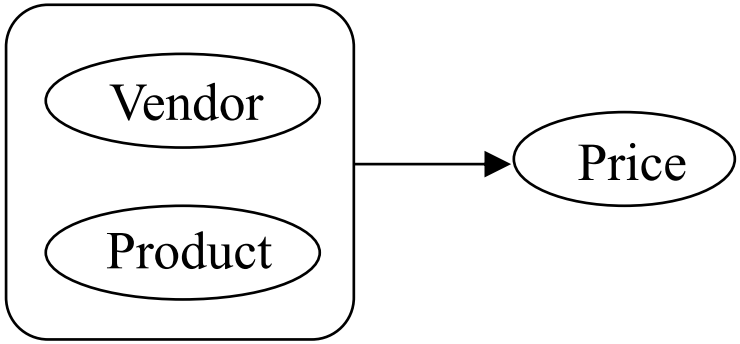
Symbolic Notation



Diagram



{Vendor, Product} ⟶ Price



Identifying Functional Dependencies

- Observing a subset of rows is not enough to accurately identify functional dependencies
  - We need to analyze the intrinsic properties of the attributes!

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

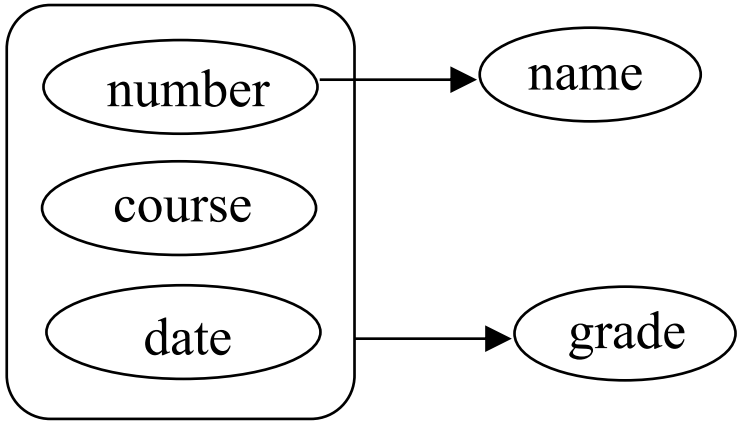
- Attribute *name* is functionally dependent on *ID*!
- *ID* is functionally dependent on *name*?
- Attribute *dept_name* determines *building*?

DEMO #1

- Assume the following relation (*exams*):

name	*number*	*course*	*date*	*grade*
António Silva	1234343	Databases	12/03/2020	12
António Silva	1234343	Analysis	12/03/2020	14
Joana Antunes	1275432	Algebra	13/03/2020	10
Joana Antunes	1275432	Databases	13/03/2020	11

- TODO: Identify functional dependencies…
  - Is the name unique?
  - What refers the date to?
  - Grade depends on what?
  - …



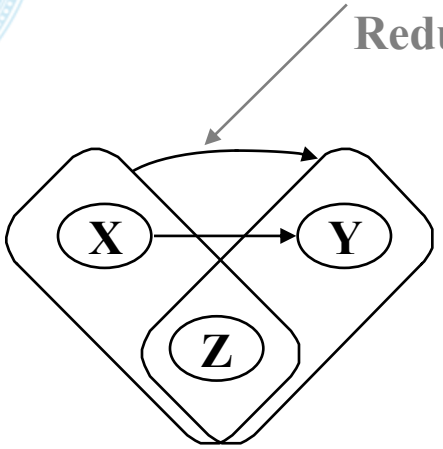
Inference Rules

- Rules to reduce or change a set of Functional Dependencies (FDs) into another set of equivalent FDs

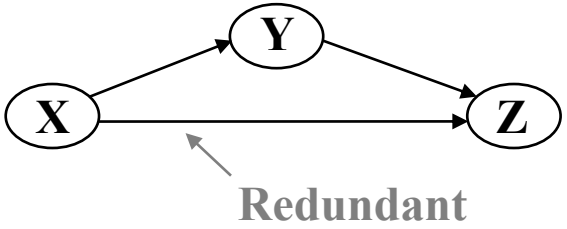
- Rules:
  - Reflexive: if Y is a subset of X, then X determines Y
  - Augmentation: if X determines Y, then XZ determines YZ for any Z
    - Also called as a partial dependency!
  - Transitive: if X determines Y and Y determines Z, then X also determines Z
  - Union: if X determines Y and X determines Z, then X also determines Y and Z
  - Decomposition: if X determines Y and Z, then X determines Y and X determines Z separately
  - Pseudo transitive: if X determines Y and YZ determines W, then XZ determines W

Inference Rules: Cases

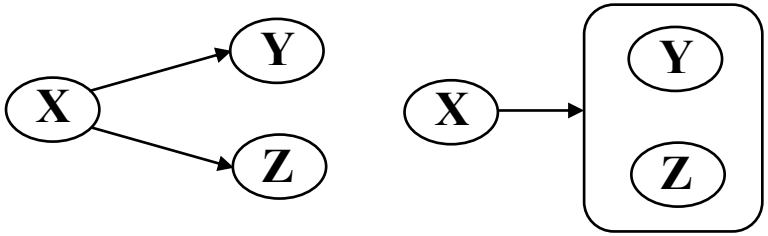
Augmentation



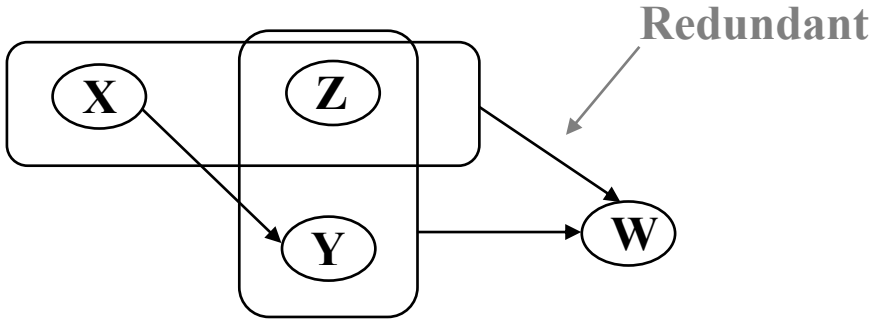
Transitive



Union / Decomposition

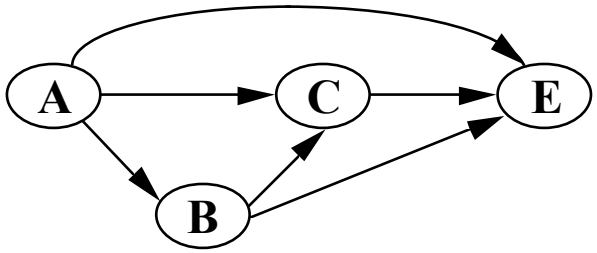


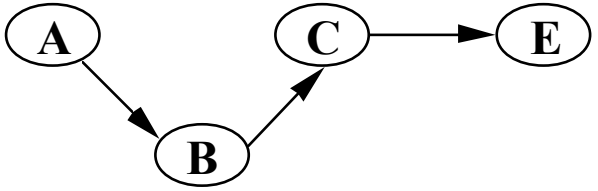
Pseudo Transitive



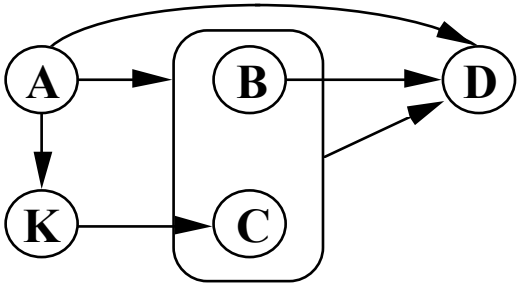
Minimal Cover

- Set of FDs obtained by removing all the redundant FDs
- May depend on the order by which the redundant FDs are removed
  - Thus, the same set of FDs may lead to different minimal covers

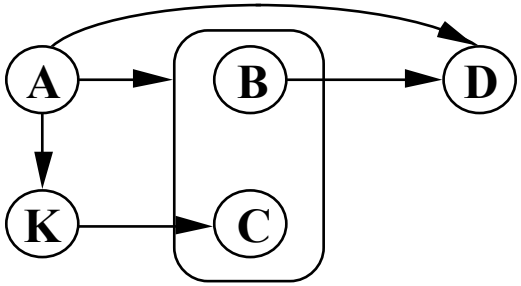




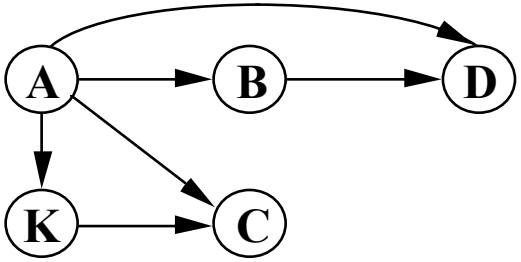
Minimal Cover: Example



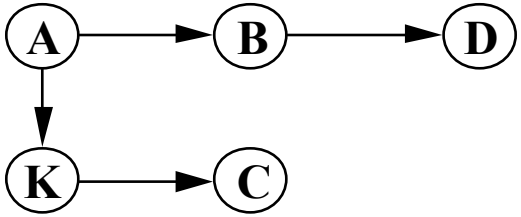
1. Original set of FDs



2. B,C → D is removed
   (augmentation)



3. A → B,C is replaced by
   A → B & A → C
   (decomposition)



4. A → C and A → D are removed
   (transitive)

Databases

Redundant and Duplicated Data
First Normal Form (1NF)
Second Normal Form (2NF)
Third Normal Form (3NF)
Boyce-Codd Normal Form (BCNF)

NORMAL FORMS

Redundant Data

- Data that can be removed without losing information

emp_sup

nemp	*supervisor*	*phone_sup*
125	António Silva	2397223433
138	Joana Antunes	2397275432
193	António Silva	2397223433
200	Joana Antunes	2397275432

Where is the redundancy?

- Is it better this way?

emp_sup

nemp	*supervisor*	*phone_sup*
125	António Silva	2397223433
138	Joana Antunes	2397275432
193	António Silva	
200	Joana Antunes	

What is the problem?

Duplicated Data

- We need to distinguish between duplicated and redundant data

emp_sup

nemp	*supervisor*
125	António Silva
138	Joana Antunes
193	António Silva
200	Joana Antunes

Duplicated but not Redundant data

- Redundant data can be removed without losing info, which is not the case for duplicated data

emp_sup

nemp	*supervisor*
125	António Silva
138	Joana Antunes
193	
200	

What is the problem?

How to Avoid Redundant Data?

emp_sup

nemp	*supervisor*	*phone_sup*
125	António Silva	2397223433
138	Joana Antunes	2397275432
193	António Silva	2397223433
200	Joana Antunes	2397275432

Table with redundant data

- The solution is to divide the table in two (lossless decomposition):

emp_sup

nemp	*supervisor*
125	António Silva
138	Joana Antunes
193	António Silva
200	Joana Antunes

supervisor

name_sup	*phone_sup*
António Silva	2397223433
Joana Antunes	2397275432

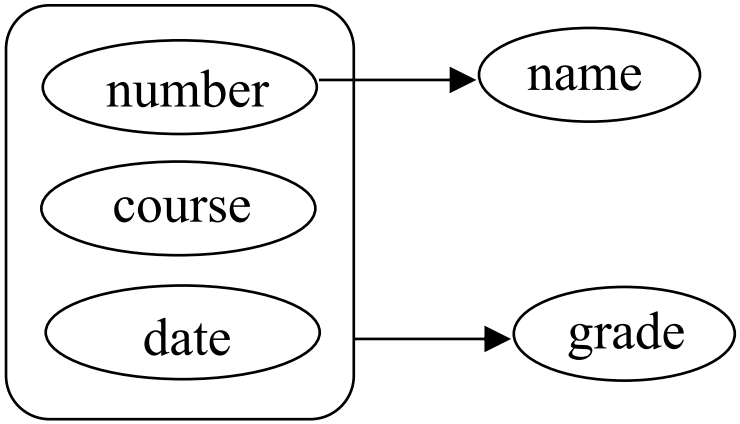
Duplicated data, but no redundant data!

DEMO #2

- Assume the following relation (*exams*):

name	*number*	*course*	*date*	*grade*
António Silva	1234343	Databases	12/03/2020	12
António Silva	1234343	Analysis	12/03/2020	14
Joana Antunes	1275432	Algebra	13/03/2020	10
Joana Antunes	1275432	Databases	13/03/2020	11

- TODO:
  - Identify redundant data
  - Decompose the table to avoid redundant data
    - student {<u>number</u>, name}
    - eval {<u>number</u>, <u>course</u>, <u>date</u>, grade}



Normalization

- Process to identify the correct structure for each table in the database
  - If a table is more normalized, then there are less redundant data
- Normalizing a table consists in dividing it into two or more tables by following a well-defined procedure
- Normal forms: consist of several successive degrees of normalization of the tables

Normal Forms

- Main normal forms:
  - First Normal Form (1NF)
  - Second Normal Form (2NF)
  - Third Normal Form (3NF)
  - Boyce-Codd Normal Form (BCNF)

- If a table is in a more advanced normal form, then it is also in the previous normal forms
- The goal is to have tables with no redundant data (or a minimum of redundant data)
  - Are there cases where having redundant data is a good solution?

First Normal Fo

number	advisor	office	course	grade
1022	Ferreira	A23	Physics	15
1022	Ferreira	A23	Chemistry	12
1022	Ferreira	A23	History	16
4123	Altino	B12	Portuguese	12
4123	Altino	B12	English	14
4123	Altino	B12	Philosophy	11

- A relation is in the 1NF if:
  - The domain of the attributes consist only of atomic values
  - There is no more than one attribute describing a given characteristic
- Is this in the 1NF?

student

number	*advisor*	*office*	*courses*	*grades*
1022	Ferreira	A23	Physics, Chemestry, History	15, 12, 16
4123	Altino	B12	Portuguese, English, Philosophy	12, 14, 11

- What about this one?

student

number	*advisor*	*office*	*course1*	*course2*	*course3*	*grade1*	*grade2*	*grade3*
1022	Ferreira	A23	Physics	Chemistry	History	15	12	16
4123	Altino	B12	Portuguese	English	Philosophy	12	14	11

Why the 1NF is NOT Enough?

- If a relation in the 1NF can hold all the information in the database, why decompose it in multiple tables?

- A relation in the 1NF contains redundant data
- Some operations (insert, update and delete) may lead to integrity violations
- Three types of data anomalies:
  - Insert anomalies: inability to add data to the database due to the absence of other data
  - Update anomalies: data inconsistency that results from data redundancy and a partial update
  - Delete anomalies: unintended loss of data due to deletion of other data

Data Anomalies: Examples

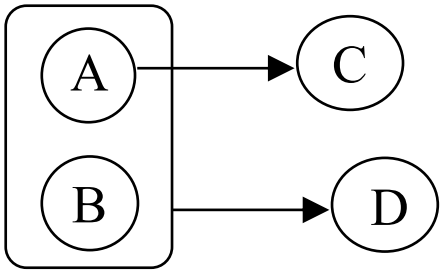
ID	*name*	*salary*	*dept_name*	*building*	*budget*
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan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

Source: A. Silberschatz, H. F. Korth and S. Sudarshan, "Database System Concepts", McGraw-Hill Education, Seventh Edition, 2019.

- Insert anomalies
  - Insert a person without a department
  - Insert a new department without people
- Update anomalies
  - Update the budget of the *Music* department?
  - What about the *Physics* department?
- Delete anomalies
  - Delete student *15151*

Second Normal Form (2NF)

- A relation is in the 2NF if:
  - It is in 1NF
  - Every non-prime attribute (attribute that is not a part of any candidate key) is dependent on the whole of every candidate key



- To assess if a relation is in the 2NF, we should ask:
  - What is the key of the relation?
    - If the key is a single attribute, then it is in the 2NF
  - If the key includes more than one attribute, then we ask: are there attributes that are non-prime attributes and that depend on part of the key?
    - If not, then it is in the 2NF!

Not in the 2NF!

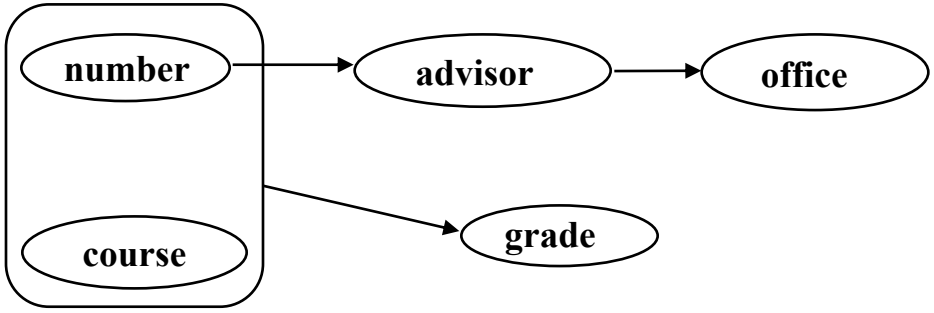
Second Normal Form (2NF): Example

student

number	*advisor*	*office*	*course*	*grade*
1022	Ferreira	A23	Physics	15
1022	Ferreira	A23	Chemistry	12
1022	Ferreira	A23	History	16
4123	Altino	B12	Portuguese	12
4123	Altino	B12	English	14
4123	Altino	B12	Philosophy	11
7239	Ferreira	A23	Chemistry	16

- What is the key of the relation?
  - *{number,course}*
- Are there attributes that are non-prime attributes and that depend on part of the key?

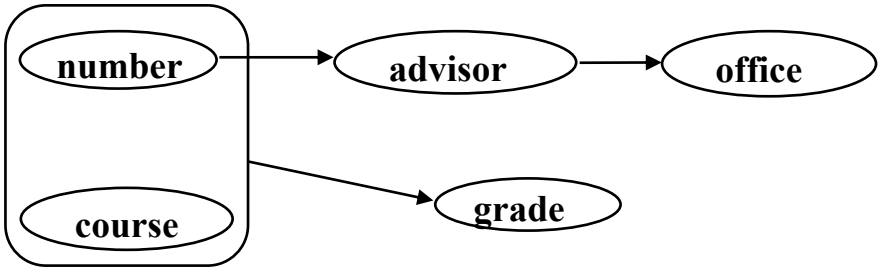
The relation is not in the 2NF!



How to make it to the 2NF?

student

number	*advisor*	*office*	*course*	*grade*
1022	Ferreira	A23	Physics	15
1022	Ferreira	A23	Chemistry	12
1022	Ferreira	A23	History	16
4123	Altino	B12	Portuguese	12
4123	Altino	B12	English	14
4123	Altino	B12	Philosophy	11
7239	Ferreira	A23	Chemistry	16



student_advisor

number	*advisor*	*office*
1022	Ferreira	A23
4123	Altino	B12
7239	Ferreira	A23

student_course

number	*course*	*grade*
1022	Physics	15
1022	Chemistry	12
1022	History	16
4123	Portuguese	12
4123	English	14
4123	Philosophy	11
7239	Chemistry	16

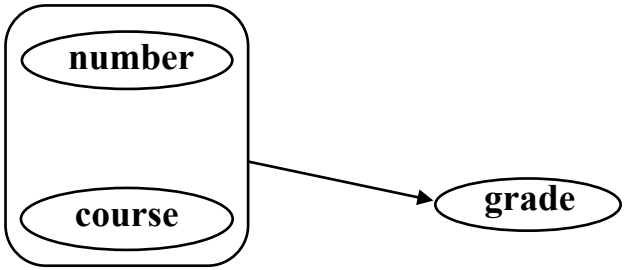
Third Normal Form (3NF)

- A relation is in the 3NF if:
  - It is in 2NF
  - All non-prime attributes depend only on the candidate key (i.e. there are no dependencies on other non-prime attributes)

student_course

number	*course*	*grade*
1022	Physics	15
1022	Chemistry	12
1022	History	16
4123	Portuguese	12
4123	English	14
4123	Philosophy	11
7239	Chemistry	16

Is this in the 3NF?

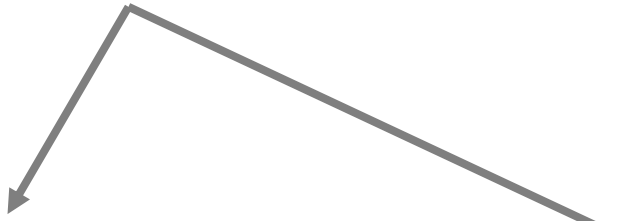


Third Normal Form (3NF): Example

student_advisor

number	*advisor*	*office*
1022	Ferreira	A23
4123	Altino	B12
7239	Ferreira	A23

Is this in the 3NF?



student_advisor

number	*advisor*
1022	Ferreira
4123	Altino
7239	Ferreira

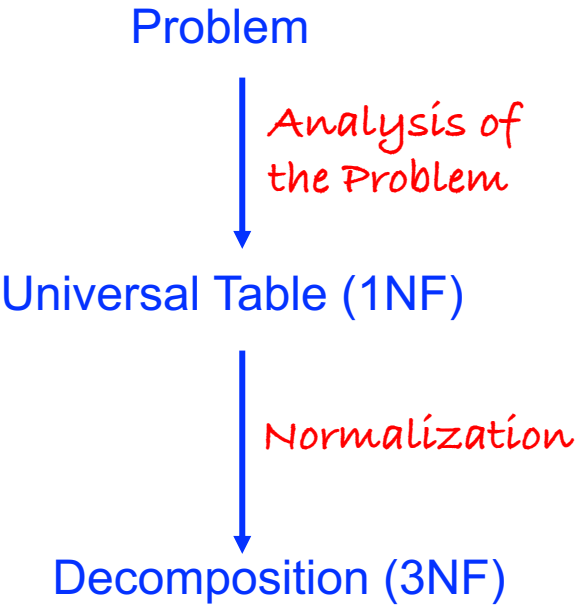
advisor_office

advisor	*office*
Ferreira	A23
Altino	B12

In Summary…

student

number	*advisor*	*office*	*course*	*grade*
1022	Ferreira	A23	Physics	15
1022	Ferreira	A23	Chemistry	12
1022	Ferreira	A23	History	16
4123	Altino	B12	Portuguese	12
4123	Altino	B12	English	14
4123	Altino	B12	Philosophy	11
7239	Ferreira	A23	Chemistry	16



student_course

number	*course*	*grade*
1022	Physics	15
1022	Chemistry	12
1022	History	16
4123	Portuguese	12
4123	English	14
4123	Philosophy	11
7239	Chemistry	16

student_advisor

number	*advisor*
1022	Ferreira
4123	Altino
7239	Ferreira

advisor_office

advisor	*office*
Ferreira	A23
Altino	B12

DEMO #3

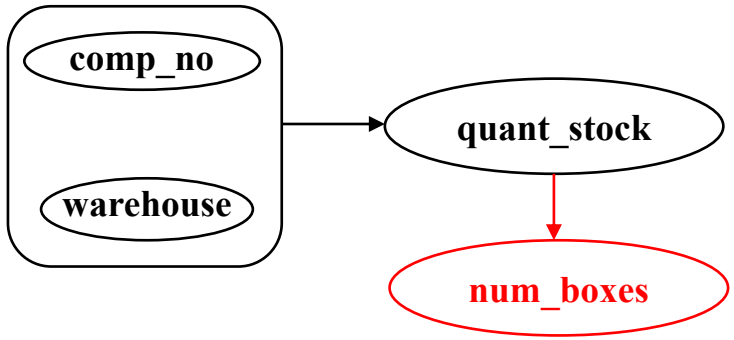
- Assume the following relation:

components_warehouse

comp_no	*warehouse*	*quant_stock*	*num_boxes*
T232	Lisbon	467	47
T232	Porto	319	32
T232	Leiria	121	13
H995	Aveiro	578	58
H995	Lisbon	227	23

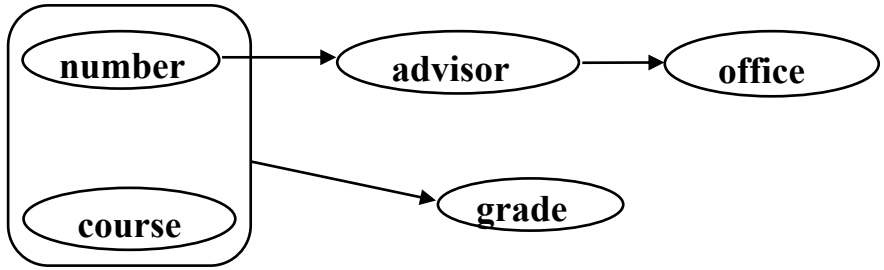
Assumption: each box holds 10 components

- TODO:
  - First Normal Form?
  - Key?
  - Second Normal Form?
  - Third Normal Form?



Boyce-Codd Normal Form (BCNF)

- A relation is in the BCNF if:
  - Every determinant attribute (or set of attributes) is a candidate key
- BCNF corresponds to a higher normalization than 3NF



Universal Relation
Functional Dependencies Diagram
Decomposition

DECOMPOSITION USING FUNCTIONAL DEPENDENCIES

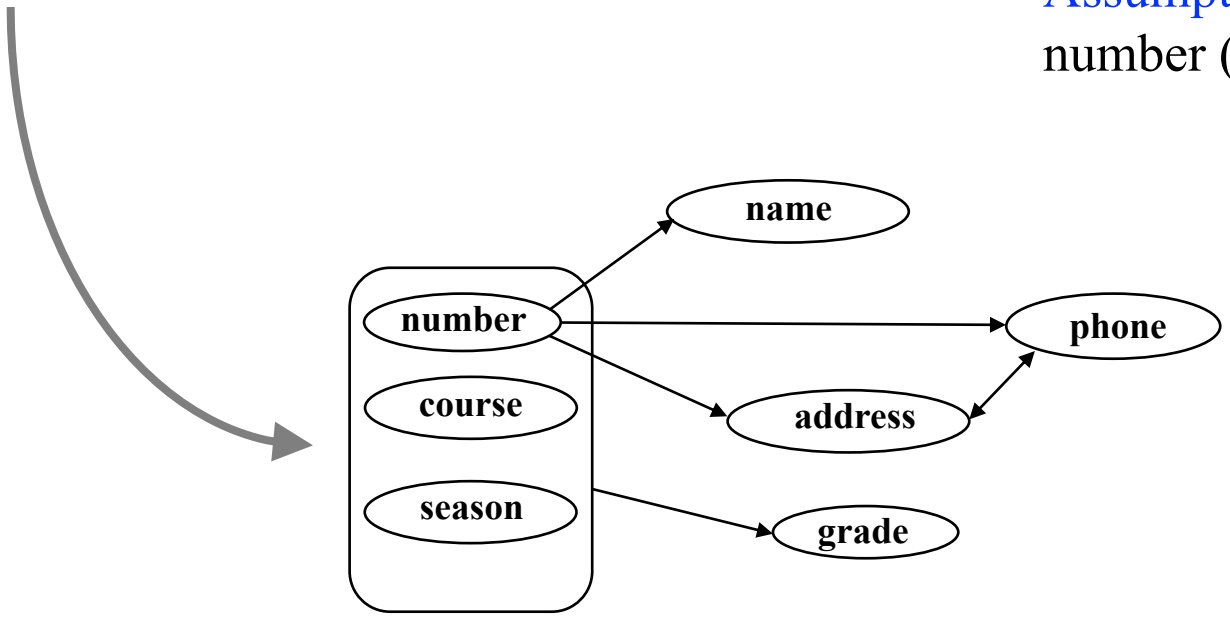
FDs And Normalization

Universal Relation

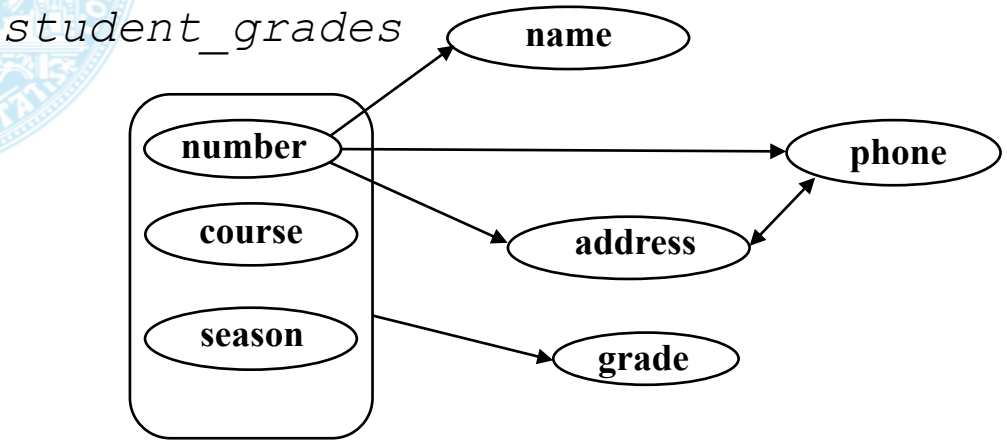
student_grades

number	*name*	*address*	*phone*	*course*	*season*	*grade*
3215	Rui Silveira	Dias da Silva, 159	239711228	MA122	2S2021	12

Assumption: there is a single phone number (landline) for an address



Is the Relation in the BCNF?



- Candidate Keys:

  {number, course, season}

- Determinants:

  {number, course, season}

  {number}

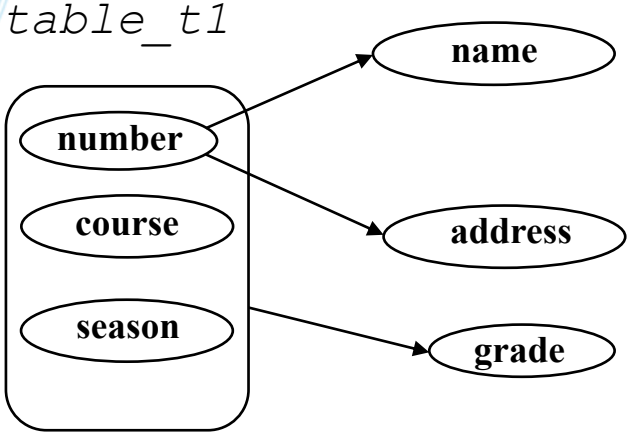
  {address}

  {phone}

- Not in the BCNF because there are determinants that are not candidate keys
  - We need to start decomposing!

Is the Relation in the BCNF?

table_t1



- Candidate Keys:

  {number, course, season}

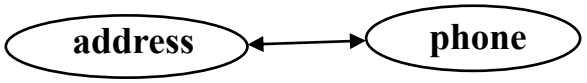
- Determinants:

  {number, course, season}

  {number}

Not in the BCNF!

address_phone



- Candidate Keys:

  {address}

  {phone}

- Determinants:

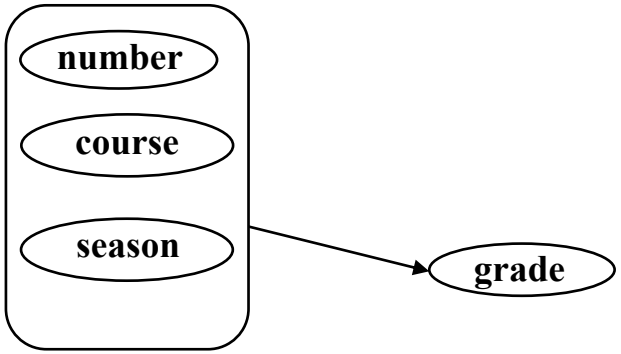
  {address}

  {phone}

This is in the BCNF!

Is the Relation in the BCNF?

grades



- Candidate Keys:

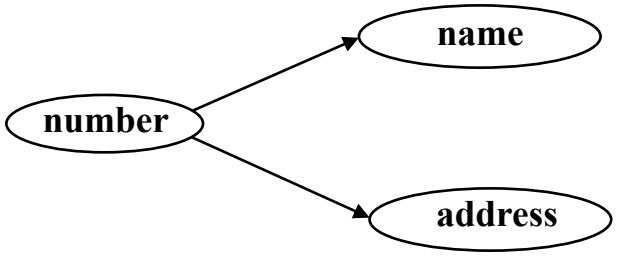
  {number, course, season}

- Determinants:

  {number, course, season}

This is in the BCNF!

students



- Candidate Keys:

  {number}

- Determinants:

  {number}

This is in the BCNF!

This Leads To…

number	*name*	*address*	*phone*	*course*	*season*	*grade*
3215	Rui Silveira	Dias da Silva, 159	239711228	MA122	2S2021	12

- The initial table is divided into three tables

  address_phone(address, phone)

  students(number, name, address)

  grades(number, course, season, grade)

`address_phone`

address	*phone*
Dias da Silva, 159	239711228
Av. Brasil, 21, 1º	239722119
T. Valadim,19	329712345

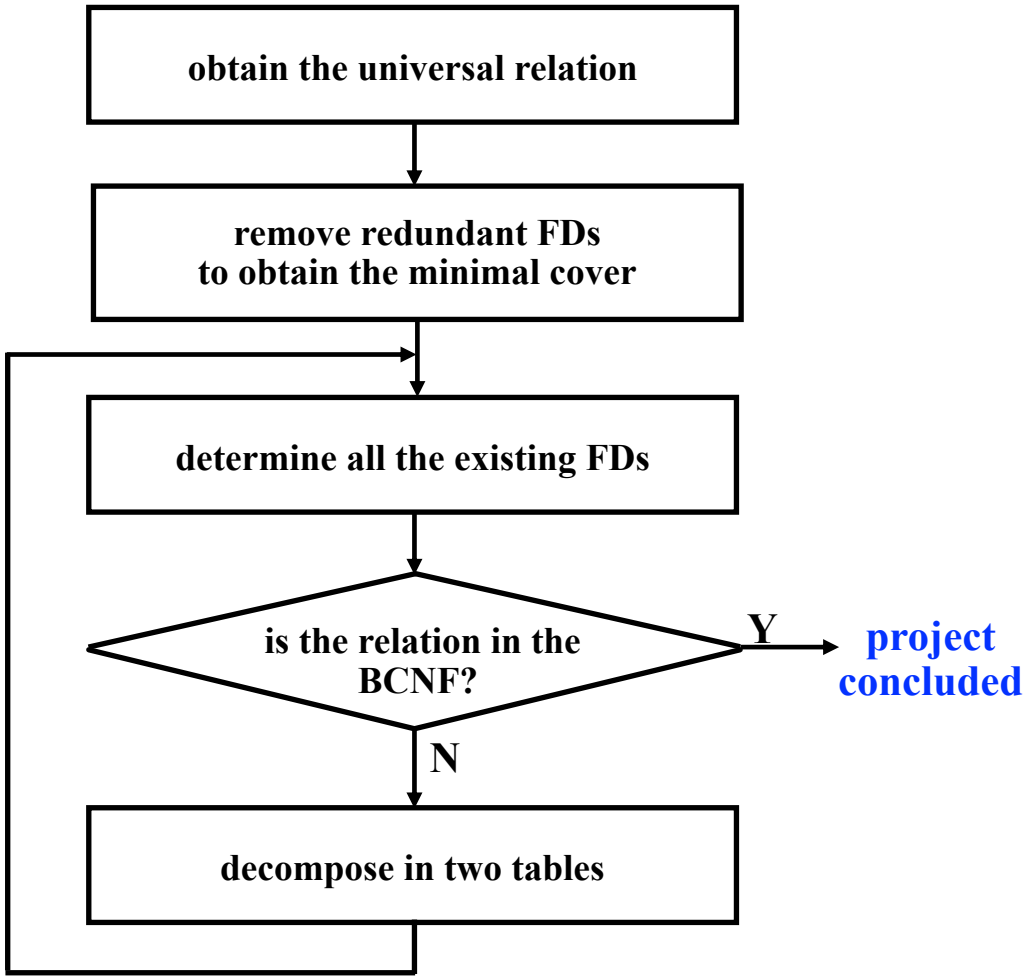
`students`

number	*name*	*address*
3215	Rui Silveira	Dias da Silva, 159
3462	João Rodrigues	Av. Brasil, 21, 1º
3567	Luís Almeida	T. Valadim,19
4756	Ana Meneses	Dias da Silva, 159

`grades`

number	*course*	*season*	*grade*
3215	MA122	2S2020	12
3215	FI120	2S2020	15
3215	AR230	1S2021	17
3215	MA122	1S2021	18
3462	MA122	2S2020	11
3462	CO123	1S2021	14
3462	AR220	1S2021	12
3567	FI239	2S2020	11
3567	SA171	1S2021	15
3567	RG141	1S2021	13
4756	EG389	2S2019	12

Overall Strategy



Take-Away(s)

- Functional Dependencies
- Redundant and Duplicated Data
- First Normal Form (1NF)
- Second Normal Form (2NF)
- Third Normal Form (3NF)
- Boyce-Codd Normal Form (BCNF)
- Functional Dependencies can be used to support the normalization process

Next Lesson(s)

- Database Transactions and Properties
- Concurrency Control
- Locking for Concurrency Control
- Transaction Isolation
- Database Recovery

Q&A

Databases

Functional Dependencies and Database Normalization

João R. Campos

Bachelor in Informatics Engineering
Department of Informatics Engineering
University of Coimbra
2023/2024