Normalization exercises

Francesco Di Giacomo, Ahmad Omar

Note: For each of the following exercises complete the following tasks:

- 1. (Complete after lesson 1) Find what normal form each of the following tables satisfies. Motivate the answer according the definition of normal forms seen in class.
- 2. (Complete after lesson 2) Apply the normalization algorithms seen in class to each table. Use intermediate refinements, i.e. if the table is in 1NF first normalize in 2NF and, if necessary, in BCNF.

Exercise 1 - Lockers The table is in no normal form, since it contains a multi-value attribute. Note that, after putting the sub-attributes of the multi-value attribute, there exists the dependency $locker_id \rightarrow key_num, size$

1NF:

locker					
teacher_id	locker_id	name	surname	key_num	size

2NF:

The dependency created at the previous step breaks 2NF because the left side is part of a key and the right side is a non-key attribute (remember that you can use the decomposition rule to create two separate dependencies, each one with a single attribute).

teacher				
$\underline{\text{teacher_id}}$	<u>locker_id</u>	name	surname	

locker				
<u>locker_id</u>	key_num	size		

In teacher the attribute locker_id is a foreign key to locker.

BCNF

The table is already in BCNF since all dependencies have the left argument which is a superkey.

Exercise 2 - Library The table is in no normal form since it contains a multi-value attribute.

1NF:

With this transformation we introduce the dependency author, title, date \rightarrow

library						
$\underline{\operatorname{card_num}}$	book_author	book_title	$\underline{\text{book_date}}$	name	surname	$return_date$

return_date

2NF:

Because of the previous dependency, the table is not in 2NF (the left side is part of a key and the right side is a non-key attribute)

member					
card_num	book_author	book_title	book_date	name	surname

		book		
Ì	book_author	book_title	$\underline{\text{date}}$	return_date

In member the attributes { book_title,book_date,return_date } are a foreign key to book.

At this point we still have the dependency date \rightarrow return_date which breaks the 2NF, so the table book must be decomposed again. The final structure in 2NF is the following:

member					
card_num	book_author	book_title	book_date	name	surname

book				
book_author	book_title	date		

	dates
date	$return_date$

In member the attribute date is foreign key to dates

BCNF:

All the tables at this point are in BCNF because all the dependencies have a superkey as left argument.

Exercise 3 - Books 1NF:

The table is already in 1NF because all the attributes are atomic.

2NF:

Both dependencies have the left argument that is part of a key and the right argument that is a non-key attribute, thus they break 2NF

In the table book the attribute _author is a foreign key to book_author and title is a foreign key to description.



boo	k_author
$\underline{\text{author}}$	$author_bdate$

description				
$\underline{\text{title}}$	genre	page	section	

BCNF:

All the tables are already in BCNF since the left argument of all the dependencies are superkeys.

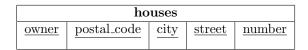
Exercise 4 - Houses The table is in no normal form because it contains a multi-value attribute.

1NF:

houses							
owner	postal_code	city	$\underline{\text{street}}$	$\underline{\mathrm{number}}$	price	size	account

2NF:

Both dependencies break the 2NF because the left argument is part of a key and the right argument is a non-key attribute. Indeed using the decomposition rule we can get, from the first one, the three dependencies postal_code \rightarrow city,street,number, postal_code \rightarrow pricepostal_code \rightarrow size. Note that the dependency postal_code \rightarrow city,street,number (which is equivalent to postal_code \rightarrow address) does not break the 2NF because the right arguments are key attributes.



house_info				
postal_code	price	size		

$owner_info$			
$\underline{\text{owner}}$	owner_account		

In houses the attribute owner is a foreign key to owner_info and postal_code is a foreign key to house_info

BCNF

All tables are already in BCNF because the left argument of all the dependencies is a superkey.

Exercise 5 - Port 1NF:

The table is already in 1NF because all attributes are atomic

2NF:

The dependencies $ship_name \rightarrow docked_at$, country, weight, class and captain $\rightarrow cpt_license$ break the 2NF because the left argument is part of a key and the right argument is a non-key attribute (again use the decomposition rule). Note that the dependency $docked_at \rightarrow country$ does not break 2NF because the left argument is not part of any key (the 2NF only considers dependencies where the left side is part of a key)



		ship		
ship_name	weight	class	$docked_at$	country

license			
captain	cpt_license		

In ship _captain the attribute ship_name is foreign key to ship and captain is a foreign key to license.

BCNF:

The dependency $docked_at \rightarrow country$ breaks BCNF because the left argument is not a superkey. The table ship must be decomposed in BCNF.

ship				
ship_name	weight	class	$docked_at$	

docking		
docke_at	country	

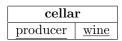
In the table ship the attribute docked_at is a foreign key to docking.

Exercise 6 - Cellar 1NF:

The table is in 1NF because it contains only atomic attributes

2NF:

The dependencies producer \rightarrow country,location and wine \rightarrow bottling_date,price/l,grape_variety break the 2NF since the left side is part of a key and the right side is a non-key attribute. The other dependencies do not break 2NF because they do not have a left argument that is part of a key



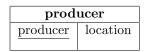
producer		
producer	country	location

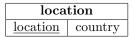
	w	rine	
wine	bottling_date	price/l	grape_variety

In cellar the attribute producer is a foreign key to the table producer and the attribute wine is a foreign key to the table wine.

BCNF:

The dependencies location \rightarrow country grape_variety \rightarrow price/1 break BCNF because their left argument is not a superkey. The tables wine and producer must be normalized in BCNF.





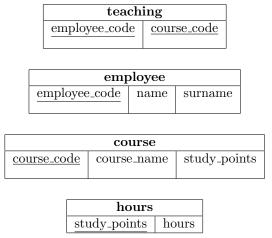
	\mathbf{wine}	
$\underline{\text{wine}}$	bottling_date	grape_variety

grape	
grape_variety	price/l

In the table wine the attribute grape_variety is foreign key to grape and in the table producer the attribute location is foreign key to the table location.

Note: The following exercises are analogous to the previous exercises so only the final solution can be found

Exercise 7 - Courses In teaching the attribute employee_code is a foreign key to employee and the attribute course_code is a foreign key to course. In the table course the attribute study_points is a foreign key to hours.



Exercise 8 - Flights In the table flying the attribute flight_code is a foreign key to flight, and the attribute captain_code is a foreign key to captain. In the table flight the attributes { departure,arrival } are a foreign key to duration.

fying

	fl	ight_code	cap	otain_code	
	flight				
flight_co	<u>de</u>	plane_model departure		arrival	
captain					
	captain_code captain_name]	

duration				
departure <u>arrival</u> flight_time				

Exercise 9 - Buildings In materials the attribute building_type_code is a foreign key to building and material is a foreign key to the table material. In the table building the attributes { length,width,height } are a foreign key to taxes.

${ m building_materials}$				
building_code	$\underline{\text{component_type}}$	material		

material				
material	aximum_pressure	specific_weight		

building					
building_code	building_length	building_width	building_height		

taxes					
building_length	building_width	building_height	tax_rate		