

Laboratory 4

Normalization

References

Lecture Notes, Elmasri and Navathe, 2017: Chapter 14

Exercise 1

- a) Consider the **Patient History** table below that shows all details of each patient's attendance at a clinic:

Patient History

patientNo	name	address	date	time	drNo	drName	visitCode	description
P0001	Johnson	12 Sturt St, Balwyn	12/11/2017	9:00am	D0003	Kelly	VC034	Allergy Desensitisation Injection
P0002	Singh	5 Willow Ave, Box Hill	14/11/2017	9:00am	D0003	Kelly	VC015	Suspected Glandular Fever – tests ordered
P0003	Hatzis	18 High St, Ringwood	14/11/2017	9:30am	D0001	Able	VC006	Stomach Complaint
P0001	Johnson	12 Sturt St, Balwyn	19/11/2017	2:00pm	D0003	Kelly	VC034	Allergy Desensitisation Injection
P0004	Ong	16 Plum St, Bulleen	23/11/2017	4:30pm	D0003	Kelly	VC034	Allergy Desensitisation Injection
P0005	Jacobson	2 Apple Tce, Balwyn	25/11/2017	10:00am	D0001	Able	VC098	Flu Vaccination
P0001	Johnson	12 Sturt St, Balwyn	26/11/2017	11:30am	D0003	Kelly	VC034	Allergy Desensitisation Injection
P0003	Hatzis	18 High St, Ringwood	03/12/2017	10:00am	D0002	Jones	VC098	Flu Vaccination
P0001	Johnson	12 Sturt St, Balwyn	03/12/2017	3:00pm	D0002	Jones	VC034	Allergy Desensitisation Injection

The following dependencies apply:

patientNo → name

patientNo → address

patientNo, date, time → drNo

patientNo, date, time → visitCode

patientNo, date, time → description

drNo → drName

(drNo => doctor number)

(visitCode => code to describe the consultation)

(description => concise description of the consultation)

(drName => doctor name)

Transform the above **Patient History** relation into BCNF table/s.

UNF -No repeating group

1NF – remove repeating groups (no repeating groups), identify primary keys

HISTORY (patientNo, name, address, date, time, drNo, drName, visitCode, description)

2NF – remove any partial dependencies

PATIENT (patientNo, name, address)

HISTORY (patientNo, date, time, drNo, drName, visitCode, description)

3NF – remove any transitive dependencies

PATIENT (patientNo, name, address)

HISTORY (patientNo, date, time, *drNo*, visitCode, description)

DOCTOR (drNo, drName)

FINAL TABLES

PATIENT (patientNo, name, address)

HISTORY (patientNo, date, time, *drNo*, visitCode, description)

DOCTOR (drNo, drName)

- b) Consider the **Supervision** table below that shows a list of students and their principle supervisor.

supervisorNo	supervisorName	studentNumber	studentName	Topic
SV001	Beresford	S0001	Patel	Process Modelling
SV002	Roulton	S0002	Wong	Data Warehousing
SV003	Zheng	S0003	Nguyen	Network Engineering
SV001	Beresford	S0004	Richards	Software Metrics

Transform the above **Supervision** relation into BCNF table/s.

UNF

STUDENT (supervisorNo, supervisorName (studentNumber, studentName, topic))

1NF

One student can only have one principle supervisor and each supervisor can supervise more than one student so there is a 1:M relationship. Therefore the primary key from the non-repeating group become a foreign key in the repeating group.

STUDENT_SUPERVISOR (supervisorNo, supervisorName)

STUDENT (studentNumber, studentName, topic, *supervisorNo*)

The above relation is already in 3NF

Exercise 2 – Practice with comparison between ER-Model and Normalization

The manager of the *Big Spender Dining Club* wishes to create a database to keep track of the club's dinner invitations which are mailed to members. He/she wants to be able to plan meals and keep track of who attends dinners more efficiently. Currently the manager uses the following two forms:

Big Spender Dinning Club					
Member Name:		Peter Wright			
Membership Number:		1234			
Address:		2 Plenty Road			
City:		Bundoora			
Postcode:		3083			
Invitation Number	Invitation Date	Acceptance Date	Dinner Date	Number of Guests	Dinner Code
25	01/07/2017	10/07/2017	14/07/2017	2	23
13	02/08/2017	04/08/2017	14/08/2017	1	24
56	01/09/2017		14/09/2017		25
25	03/10/2017	10/10/2017	17/10/2017	3	26

- Invitation Number: each invitation mailed to a member contains a unique invitation number for a given dinner
- Invitation Date – the date on which the invitation is mailed out – invitations for a given dinner are mailed out over a period of one week
- Acceptance Date – the date on which the member accepts/rejects
- Dinner Date – the date the dinner is to be held
- Number of Guests – the number of people attending the dinner, excluding the member
- Dinner Code – a unique code for the dinner being offered – mains and desserts are mixed and matched to build dinners. A given main or dessert may be served at many different dinners

Big Spender Dinning Club		
Dinner Code: 24 Date Dinner Held: 14/8/2017 Dinner Description: Roast Dinner		
Main Course		
<i>Main Code</i>	<i>Main Description</i>	<i>Number Served</i>
M0004	Roast Chicken	30
M0005	Roast Lamb	5
M0006	Roast Beef	10
Dessert Course		
<i>Dessert Code</i>	<i>Dessert Description</i>	<i>Number Served</i>
D0001	Bread and Butter Pudding	4
D0002	Sticky Date Pudding	20
D0003	Plum Pudding	10

Normalize the above two forms to arrive at a suitable database design for the *Big Spender Dinning Club*, clearly show the stages UNF, 1NF, 2NF, 3NF and BCNF for **each** form. Make sure you clearly show the final tables.¹

Form 1:

UNF

MEMBER (membershipNo, memberName, address, city, postcode (inviteNo, inviteDate, acceptDate, dinnerDate, noOfGuests, dinnerCode))

1NF

A member can attend many dinners and a dinner can be attended by many members. Therefore the primary key of the non-repeating relation is combined with the key of the repeating relation.

MEMBER (membershipNo, memberName, address, city, postcode)

MEMBER-DINNER (membershipNo, dinnerCode, inviteNo, inviteDate, acceptDate, dinnerDate, noOfGuests)

2NF

There is a partial dependency between the attribute dinnerDate and the key of MEMBER-DINNER (i.e. dinnerDate is **only** dependent on dinnerCode). Therefore a new relation is created with dinnerCode as the key and dinnerDate is removed from MEMBER-DINNER into this new relation.

¹ Adapted from Rob P. and Coronel C. *Database Systems – Design, Implementation, & Management*, Thomson Learning, 2000.

MEMBER (membershipNo, memberName, address, city, postcode)
 MEMBER-DINNER (membershipNo, dinnerCode, inviteNo, inviteDate, acceptDate,
 noOfGuests)
 DINNER (dinnerCode, dinnerDate)

3NF

MEMBER (membershipNo, memberName, address, city, postcode)
 - 2NF due to postcode → city, but leave in 2NF for performance reasons
 MEMBER-DINNER (membershipNo, dinnerCode, inviteNo, inviteDate, acceptDate,
 noOfGuests)
 DINNER (dinnerCode, dinnerDate)

Form 2:

UNF

DINNER (dinnerCode, dinnerDate, dinnerDesc (mainCode, mainDesc,
 mainNumberServed) (desCode, desDesc, desNumberServed))

1NF

DINNER (dinnerCode, dinnerDate, dinnerDesc)
 DINNER-MAIN (dinnerCode, mainCode, mainDesc, mainNumberServed)
 DINNER-DESSERT (dinnerCode, desCode, desDesc, desNumberServed)

2NF

DINNER (dinnerCode, dinnerDate, dinnerDesc)
 DINNER-MAIN (dinnerCode, mainCode, mainNumberServed)
 DINNER-DESSERT (dinnerCode, desCode, desNumberServed)
 MAIN (mainCode, mainDesc)
 DESSERT (desCode, desDesc)

3NF

DINNER (dinnerCode, dinnerDate, dinnerDesc)
 DINNER-MAIN (dinnerCode, mainCode, mainNumberServed)
 DINNER-DESSERT (dinnerCode, desCode, desNumberServed)
 MAIN (mainCode, mainDesc)
 DESSERT (desCode, desDesc)

FINAL TABLES

MEMBER (membershipNo, memberName, address, city, postcode)
 MEMBER-DINNER (membershipNo, dinnerCode, inviteNo, inviteDate, acceptDate,
 noOfGuests)
 DINNER (dinnerCode, dinnerDate, dinnerDesc)
 DINNER-MAIN (dinnerCode, mainCode, mainNumber)
 DINNER-DESSERT (dinnerCode, desCode, desNumber)
 MAIN (mainCode, mainDesc)
 DESSERT (desCode, desDesc)