

FIT9132 Introduction to Databases

Week 5 Tutorial Activities

NORMALISATION

FIT Database Teaching Team

Complete the week 5 activities:

[5.1 Normalisation -- Class Discussion](#)

[5.1.1 Data Anomalies](#)

[5.1.2 Representing a Form as a Relation](#)

[5.1.3 The Normalisation Process](#)

[5.2 Multiple Forms Normalisation -- Part 1](#)

[5.3 Multiple Forms Normalisation -- Part 2](#)

[5.4 Additional Normalisation Exercise](#)

FIT9132 2021 S2

FIT9132 Introduction to Databases

Author: FIT Database Teaching Team

License: Copyright © Monash University, unless otherwise stated. All Rights Reserved.

COPYRIGHT WARNING

Warning

This material is protected by copyright. For use within Monash University only. NOT FOR RESALE.

Do not remove this notice.

Important

Remember, before starting any tutorial activity which involves working with files, first use SQL Developer to pull from the FIT GitLab server so as to ensure your local and server files are in sync.

Learning Objectives

- understand the definition of data anomalies
- understand the definition of the different types of dependencies in normalisation (full, partial, transitive)
- understand the steps taken during normalisation starting from unnormalised form (UNF) to third normal form (3NF)
- be able to draw a dependency diagram in 1NF, 2NF and 3NF
- be able to perform a normalisation process to 3NF given a description of database requirement

5.1 Normalisation -- Class Discussion

In relational databases, all relations must be valid and free of anomalies which may occur when inserting, updating or deleting data. Normalisation is a systematic series of steps that is used for creating valid relations, i.e. each relation meets the required properties of the relational model. The relational model makes use of primary keys (PK) and foreign keys (FK) to create logical connections between collections of data (relations, or in implementation tables). The process of normalisation and the terms primary key and foreign key thus **ONLY** have relevance when discussing the relational model or its implementation in an RDBMS.

5.1.1 Data Anomalies

Consider the following form:

dentist_no	dentist_name	patient_no	patient_name	appointment		surgeryroom_no
				date	time	
D1011	Tony Smith	P100	Gillian White	12-Sep-19	10:00	S15
D1011	Tony Smith	P105	Jill Bell	12-Sep-19	12:00	S15
D1024	Helen Pearson	P108	Ian MacKay	12-Sep-19	10:00	S10
D1024	Helen Pearson	P108	Ian MacKay	14-Sep-19	14:00	S10
D1032	Robin Plevin	P105	Jill Bell	14-Sep-19	16:30	S15
D1032	Robin Plevin	P110	John Walker	15-Sep-19	18:00	S13

This form shows the details about dental surgery appointments. Discuss possible insert, update and delete anomalies based on the above form.

5.1.2 Representing a Form as a Relation

The normalisation process starts with identifying what the supplied form represents, the attributes on the form and the possible existence of any repeating groups. The starting set of attributes is called UNF (UnNormalized Form).

The form shown in section 5.1.21 above shows multiple instances of the details about dental surgery appointment (APPOINTMENT for short), as such the UNF is represented as:

APPOINTMENT(dentist_no, dentist_name, patient_no, patient_name, app_datetime, surgeryroom_no)

To complete the normalisation, you take the UNF as the starting point and follow the normalisation steps through to at least 3NF.

5.1.3 The Normalisation Process

Legend: Primary Key is underlined.

When completing normalisation you **must not add any new attributes**, for example, surrogate keys (these are added to the final logical model post normalisation). Also during normalisation you **must not exclude any attributes** displayed on the provided form/s - we wish to capture the full semantics of the form/s. Decisions about making an attribute derived or not need to be based on transaction loads and more advanced statistics which are not available at this stage of the design process. Such decisions are made after the logical model is implemented as part of the fine-tuning of the database.

You should complete your normalisations using either Google Docs or MS Word. If you are using Google Docs be sure to download your work as a MS Word file and save it in your working directory. Regularly push your work onto FITGitLab server to maintain the development history.

UNF

APPOINTMENT(dentist_no, dentist_name, patient_no, patient_name, app_datetime, surgeryroom_no)

1NF

APPOINTMENT(dentist_no, dentist_name, patient_no, patient_name, app_datetime, surgeryroom_no)

*note that there are 3 candidate keys: (dentist_no, app_datetime), (patient_no, app_datetime) and (surgeryroom_no, app_datetime) and (dentist_no, app_datetime) is picked as PK

Partial dependencies:

dentist_no → dentist_name

patient_no → patient_name

*note that we use general definition, partial dependency is based on PK and all candidate keys

2NF

APPOINTMENT(dentist_no, patient_no, app_datetime, surgeryroom_no)

DENTIST(dentist_no, dentist_name)

PATIENT(patient_no, patient_name)

Transitive dependencies:

No transitive dependency

3NF

There is no transitive dependency, the 3NF is the same as the 2NF. Note that you are **required** to show all forms, even if they are the same as a previous form.

APPOINTMENT(dentist_no, patient_no, app_datetime, surgeryroom_no)

DENTIST(dentist_no, dentist_name)

PATIENT(patient_no, patient_name)

Full Dependencies:

dentist_no, app_datetime → patient_no, surgeryroom_no

dentist_no → dentist_name

patient_no → patient_name

5.2 Multiple Forms Normalisation -- Part 1

Normalise the following forms describing a student/units system to 3NF. Remember you must show UNF, 1NF, 2NF, 3NF and amalgamate

Note:

- To simplify a normalisation process that involves multiple forms, you should perform the normalisation one form at a time until all relations are in 3NF. Once you have done this process for all forms, consolidate the relations from the different forms (called attribute synthesis) by:
 - grouping together all relations with the same primary key, i.e representing the same entity.
- choose a single name for synonyms. For example, **mentor** is the same as **lecturer**.

Normally we include all attributes in the form/report, but the report date in following reports is not an attribute that should be included in the normalisation process since it only represents the date when the report was printed.

UNITS CURRENTLY APPROVED			REPORT DATE: 22/03/2020
Unit Number	Unit Name	Unit Description	Unit Value
FIT9131	Programming Foundations	Introduction to programming	6
FIT9132	Introduction to Databases	Database Fundamentals	6
FIT9134	Computer Architecture and Operating Systems	Fundamentals of computer systems and the computing environment	6
FIT9135	Data Communications	Fundamentals of data and computer communications	6

* Unit value may be either 3, 6 or 12 points

LECTURER DETAILS		REPORT DATE: 22/03/2020
LECTURER'S NUMBER: 10234 LECTURER'S NAME: GUISEPPE BLOGGS LECTURER'S OFFICE No.: 169 LECTURER'S PHONE No.: 99037111		
UNIT ADVISER FOR:		
UNIT NUMBER	UNIT NAME	
FIT9131	Programming Foundations	
FIT9134	Computer Architecture and Operating Systems	

* A given unit may have several advisers

* Some lecturers share offices, although each has their own phone

STUDENT DETAILS			REPORT DATE: 22/03/2020
STUDENT No.: 12345678 STUDENT NAME: Poindexter Jones STUDENT ADDRESS: 23 Wide Road, Caulfield, 3162 COURSE ENROLLED: MIT MODE OF STUDY: On-Campus MENTOR NUMBER: 10234 MENTOR NAME: Guiseppe Bloggs			
ACADEMIC RECORD:			
UNIT NUMBER	UNIT NAME	YEAR / SEMESTER	GRADE
FIT9131	Programming Foundations	2017/2	N
FIT9131	Programming Foundations	2018/1	D
FIT9132	Introduction to Databases	2018/1	D

* Grade may have the value N, P, C, D or HD

* Mode of Study must be On-campus (O) or Distance Education (D)

In order to add a student, the lecturer who advises that student must already exist in the database. No lecturer who advises any students may be deleted from the database. If the lecturer number of a lecturer is changed, then the number would be changed for each student advised by that lecturer.

5.3 Multiple Forms Normalisation -- Part 2

Normalise the following forms to 3NF. Remember you must show UNF, 1NF, 2NF, 3NF and amalgamate (consolidate/attribute synthesis). To simplify a normalisation process that involves multiple forms, you should perform the normalisation one form at a time until all relations are in 3NF. Once you have done this process for all forms, consolidate the relations from the different forms (called attribute synthesis) by grouping together all relations with the same primary key, i.e representing the same entity.

When working with a potentially composite attribute, you should consider how it is depicted on the form/report. If the attribute is depicted as non decomposed, then you should treat it as a simple attribute for the purposes of normalisation. *However*, this decision must be informed by any further information that you have about the scenario you are modelling. For example your client has informed you that they record a tenants given name and family name, whereas a form you are normalising has an entry such as:

Tenant Name: Johanston Karkov

In such a situation, this output is simply the manner which has been chosen to display the name on the form. As part of normalising such an attribute it must be decomposed into given name and family name since you were informed that this is the way it is recorded.

The agency records normal property maintenance. Maintenance costs are charged to the property owner. There is only a single maintenance for each property at a particular date and time. Below is a sample of property maintenance report prepared for the property owner:

PROPERTY MAINTENANCE REPORT		
Property No: 1965		
Property Address: Unit 1 100-102 Platypus St, Oakleigh, VIC, 3166		
Owner No: 9321		
Owner:		
Given Name: Lilian		
Family Name: Potter		
Owner Address: 14 Puffin Rd, Wantirna, VIC, 3152		
MAINTENANCE RECORD:		
DateTime	Description	Cost
2019-02-01 9:00	Roof leak	\$854
2019-02-10 10:00	Lawn repair	\$230
2019-02-10 14:00	Minor crack repair	\$428

The agency also provides two samples of simplified property tenant ledgers (tenant payment history):

Note that the two property tenant ledgers shown are two examples of the same form, only one normalisation process that covers both samples is required.

PROPERTY TENANT LEDGER (Sample 1)				
Property No: 1965 Property Address: Unit 1 100-102 Platypus St, Oakleigh, VIC, 3166 Lease Start Date: 2017-01-14 Weekly Rental Rate: \$425 Bond: \$1,842 Tenant No: 512736 Tenant Name: Johanston Karkov				
PAYMENT RECORD:				
Number	Date	Type	Amount	Paid By
80010	2017-01-10	Bond	\$1,842	bank cheque
80024	2017-01-10	Rental	\$1,842	direct deposit
80501	2017-02-10	Rental	\$1,842	bpay
80612	2017-02-28	Damage	\$523	credit card
80615	2017-02-28	Damage	\$312	credit card
80892	2017-03-12	Rental	\$1,842	bpay
81073	2017-04-13	Rental	\$1,842	credit card

PROPERTY TENANT LEDGER (Sample 2)				
Property No: 1965 Property Address: Unit 1 100-102 Platypus St, Oakleigh, VIC, 3166 Lease Start Date: 2019-05-05 Weekly Rental Rate: \$475 Bond: \$2,059 Tenant No: 623124 Tenant Name: Kiky Longbottom				
PAYMENT RECORD:				
Number	Date	Type	Amount	Paid By
101829	2019-05-01	Bond	\$2,059	bank cheque
101934	2019-05-03	Rental	\$2,059	credit card
102104	2019-06-04	Rental	\$2,059	credit card

5.4 Additional Normalisation Exercise

A travel agency manages flight bookings for its clients. For each booking, the agency creates a booking itinerary. The following rules are applied to the itineraries:

- A booking is made for a single person. Each booking is identified by a unique booking number. A person may have one or more bookings with the travel agency.
- A booking may involve one or many flights. Each flight is identified by the flight number. A flight number is unique for a given flight route and time of the day. The flight number will be re-used *across different days* to represent the same scheduled trip. However, airlines use a different flight number to represent each scheduled trip on the same day. For example, flight number QF47 represents a flight which departs from Sydney and flies to Melbourne at 7:30 PM. Flight QF47 may, for example, be scheduled to fly on each day of the week.
- The duration of the flight is calculated based on the difference between the departure date/time and the arrival date/time.

SAMPLE ITINERARIES

Three sample itineraries are supplied below:

Booking Number: QFTR23

Client no: 123

Client Name: W.H Red

Flight Number	: QF9
Depart date	: 12-July-2018
Depart time	: 10 AM
Depart Airport code	: MEL
Depart Airport name	: Melbourne
Arrival date	: 12-July-2018
Arrival time	: 7:50 AM
Arrival Airport code	: LAX
Arrival Airport name	: Los Angeles
Duration	: 14 hr 50 mins

Flight Number	: AA1
Depart date	: 12-July-2018
Depart time	: 10 AM
Depart Airport code	: LAX
Depart Airport name	: Los Angeles
Arrival date	: 12-July-2018
Arrival time	: 11:05 AM
Arrival Airport code	: SAN
Arrival Airport name	: San Diego
Duration	: 1 hr 5 mins

Booking Number: AXYT12

Client no: 345

Client Name: B.B Brown

Flight Number	: QF47
Depart date	: 21-Aug-2018
Depart time	: 7:30 PM
Depart Airport code	: SYD
Depart Airport name	: Sydney
Arrival date	: 21-Aug-2018
Arrival time	: 8:50 PM
Arrival Airport code	: MEL
Arrival Airport name	: Melbourne
Duration	: 1 hr 20 mins

Booking Number: QFAB12

Client no: 123

Client Name: W.H Red

Flight Number:	: QF47
Depart date	: 22-Aug-2018
Depart time	: 7:30 PM
Depart Airport code	: SYD
Depart Airport name	: Sydney
Arrival date	: 22-Aug-2018
Arrival time	: 8:50 PM
Arrival Airport code	: MEL
Arrival Airport name	: Melbourne
Duration	: 1 hr 20 mins

1. Represent this booking data in UNF. Note that the three itineraries shown are three examples of the same form, only one UNF is required that would cover all three examples.
2. Convert this UNF to the first normal form (1NF) and show all dependencies via dependency diagrams. Continue the normalisation through to the third normal form (3NF). Clearly write the relations in each step from the unnormalised form (UNF) to the third normal form (3NF). You may consolidate the relations after arriving at 3NF (if necessary).

Important

After you have completed your current lab activities, at the end of each session remember to add, commit and push any changes you have made to the FIT GitLab server.

You need to get into the habit of establishing this as a standard FIT9132 workflow - Use SQL Developer to pull at the start of your working session, work on the activities you wish to/are able to complete during this session, add files (stage)/commit changes and then push the changes back to the FIT GitLab server.

Remember you should also regularly use the Web UI (login to the web interface of the server) to check that your files are correctly being pushed. Note that you must not modify (add, update, and delete) the files in the server using the Web UI.