Topic support guide

Cambridge International A Level

Cambridge International AS & A Level Information Technology

9626

For examination from 2017

Topic 9.2 Normalisation to third normal form (3NF)

Cambridge Advanced

CAMBRIDGE International Examinations

Version 1

A DIVISION OF CAMBRIDGE ASSESSMENT

Contents

Introduction

How to use this guide

The aim of this guide is to facilitate your teaching of Cambridge International AS & A Level Information Technology, syllabus topic 9.2 Normalisation to third normal form (3NF). This is part of topic 9 Database and file concepts. Normalisation can sometimes seem like a challenging topic to learn and a complex process to teach, but the guidance and activities in this resource are designed to help.

Section 1 lists some key terms used in this topic and their definitions. Section 2 Theory explains normalisation including the advantages and disadvantages, and can be used as a worksheet for your learners. Section 3 provides a worked example of the stages in taking a database from unnormalised form to third normal form. Section 4 indicates what your learners need to know, understand, or be able to do for this topic in the examination, and how they can prepare for it. Section 5 lists some further resources relevant to the topic for you or your learners to use. Section 6 provides activities that can be used to help learners understand normalisation:

- two examples of databases in different stages that require learners to normalise to third normal form
- a quiz to test learners on their knowledge of normalisation
- test questions for the topic of normalisation.

The software shown in screenshots, and the database files provided, are for Microsoft Access. This is the most common database package, but there are other suitable database packages available.

Learning objectives

Reading this guide should help you guide learners to cope with the following syllabus learning objectives:

- describe the characteristics of data in unnormalised form (0NF), first normal form (1NF), second normal form (2NF) and third normal form (3NF)
- discuss the advantages and disadvantages of normalisation
- normalise a database to 3NF

Prior knowledge

Before you begin teaching this topic:

- Make sure you understand the concepts of normalised and unnormalised data, first, second and third normal form before you begin teaching this unit
- Make sure you understand how to work through the stages of normalising a database.

1. Key terms

Word/phrase	Meaning
atomic	A field that is atomic contains only one item of data
composite key	A key made up of more than one field
database	A structure, either paper-based or electronic, to hold a set of data
data integrity	The level of accuracy and reliability of data
data redundancy	Data that is unnecessary (not needed)
duplicated data	Data that is repeated
field	One item of data or information about a person or thing in a database table
foreign key	A primary key that appears in another table in a database to create a link between the two tables
inconsistent data	Repeated entries of data in a database that do not match
normalisation	Organising data in a database to reduce data redundancy and increase data integrity
primary key	A unique identifier in a table that makes each record in the table unique
record	A collection of fields in a database table for one entry
relational database	A database consisting of more than one table where the tables are linked using key fields
unnormalised data	Data that is unnormalised may have repeated entries and fields that are not atomic
1NF	A database that is in 1NF will not have any repeating fields and all the fields in the database will be atomic
2NF	A database in 2NF will have the characteristics of 1NF. In addition, fields that are related will be separated into different tables, dependant on the primary key of the table.
3NF	A database in 3NF will have the characteristics of 2NF. In addition, any fields that are not directly related to each other are given further separation into tables that only contain fields that are directly related.

2. Theory

2.1 Introduction

There are a number of different methods that can be used to teach normalisation; the worked example in this guide shows one method. While it may be useful for learners to understand different methods, this can also confuse them. The method shown is a simple way to explain the process.

2.2 What is normalisation?

Normalisation is organising data in a database to reduce data redundancy and increase data integrity. When data is duplicated in a database it is known as data redundancy. Data redundancy can cause a problem as it can lead to inconsistencies in the data. This would reduce the integrity of the data. The purpose of normalisation is to reduce the amount of duplicated data and to simplify the structure of a database.

Example

A database stores data about people who buy books. Each time a person buys a book their name and address are entered into the database to record the purchase. This means that each time the person buys a book their data will be duplicated in the database.

Over time, the customers' data might be entered into the database many times. This can be very wasteful of storage space and it can also lead to inconsistencies, as the customer data may be entered slightly different each time. Also, if the person moves to a different address and buys another book, the same name will now be entered, but with a different address. The database now has inconsistent data, as the address fields will not match.

A database that allows duplicated entries of data is called an unnormalised database. This database can then be normalised to remove the redundant data.

2.3 Stages of normalisation

2.3.1 Unnormalised form (UNF or 0NF)

A database that is unnormalised has repeated entries of data. It may also have repeating fields and it does not have a primary key. An unnormalised database may also have data that is not atomic. This means that each field in the database could contain more than one piece of information. For example, the field 'Customer Name' in the database could contain the full name of the customer – both their first name and last name – in this one field.

2.3.2 First normal form (1NF)

A database that is in 1NF does not have any repeating fields and all the fields in the database are atomic. Each record in the database is unique. In order to create this the database has a unique identifier that is called a primary key.

2.3.3 Second normal form (2NF)

A database that is in 2NF has all of the characteristics of a database in 1NF. In addition, any data that is not dependent on the primary key of a table will be separated. This results in the creation of multiple tables that each have a primary key. The fields in each table are only related to the primary key of that table.

2.3.4 Third normal form (3NF)

A database that is in 3NF will have all of the characteristics of a database in 2NF. In addition, any fields that are not directly related to each other are separated into other tables. These tables contain only fields that are directly related, along with their primary key. In order to link these tables together to create relationships, a primary key may appear as a 'foreign key' in a linked table.

2.4 Advantages and disadvantages of normalisation

Advantages	**Disadvantages**
The storage capacity needed for the database may be smaller as there is no repeating data.	The process of normalising a database can be very complex and is a specialist skill.
The database is easy to search as there is less data to search through, due to the removal of repeated entries.	The process of searching the database may be slower. This is because a normalised database requires greater usage of a device's central processing unit (CPU), due to the links between tables.
If the data is amended in an entry this will be amended in all matching entries in the database, due to the links between the tables.	If data is amended in a normalised database then all matching entries are amended. This means that the previous data used is deleted and can no longer be seen in any entries.
The integrity of the data in the database is increased as there are no repeated entries of data that could cause it to be inconsistent.	

3. Worked example

This section works through the stages in taking a database from unnormalised form to third normal form.

An entertainment company books music bands and artists for concert dates in the UK. The company handles the bookings for multiple bands at multiple venues and it communicates with multiple agents to do this.

An example of a database for this company in unnormalised (0NF) form:

Artist ID	Artist Name	Venue ID	Concert Venue	Concert Date	Agent Name	Agent Email
RO01	Rock On	O21	O_2 Academy - Birmingham	01/06/2016	Alice Jones	ajones@welovemusic.com
SF01	Stage Fright	HA1	Hammersmith Apollo - London	01/06/2016	Bill Richards	billyrich@amazingartists.com
JS01	John Smith	O22	O_2 Academy - Leeds	10/06/2016	Henry Cooper	henry.cooper@me.com
RO01	Rock On	BC1	Barbican Centre - London	06/06/2016	Alice Jones	ajones@welovemusic.com
RO01	Rock On	O23	O_2 Academy - Liverpool	10/06/2016	Alice Jones	ajones@welovemusic.com
SF01	Stage Fright	RA1	The Royal Albert Hall - London	20/06/2016	Bill Richards	billyrich@amazingartists.com
JY01	Jimmy	RA1	The Royal Albert Hall - London	28/06/2016	Henry Cooper	henry.cooper@me.com
TD01	The Delights	O21	O_2 Academy - Birmingham	13/06/2016	Alice Jones	ajones@welovemusic.com
RO01	Rock On	HA1	Hammersmith Apollo - London	13/06/2016	Alice Jones	ajones@welovemusic.com
JY01	Jimmy	BC1	Barbican Centre - London	30/06/2016	Henry Cooper	henry.cooper@me.com

We can see from looking at this database that there are repeated entries for each artist. This means that there is more than one entry in the database for an artist. The fields ‘Concert Venue’ and ‘Agent Name’ are not atomic – that is, they have more than one item of data in the field. This means that the data in that field could be broken down further. For example, the agent’s first name and last name are both in the same field ‘Agent Name’. This could be broken down into two fields. Another feature of this database is that each record in the database does not have a unique identifier.

We can take this unnormalised database into 1NF by making sure that each field is atomic (only contains one item of data), and by giving each record a unique identifier. We do not need to remove any repeating fields in a record as none occur in this database. There are repeating entries but these are removed in a later stage of normalisation.

In 1NF our database looks like this:

Artist ID*	**Artist Name**	**Venue ID***	**Concert Venue**	**Location of Venue**	**Concert Date***	**Agent First Name**	**Agent Last Name**	**Agent Email**
RO01	Rock On	O21	O_2 Academy	Birmingham	01/06/2016	Alice	Jones	ajones@welovemusic.com
SF01	Stage Fright	HA1	Hammersmith Apollo	London	01/06/2016	Bill	Richards	billyrich@amazingartists.com
JS01	John Smith	O22	O_2 Academy	Leeds	10/06/2016	Henry	Cooper	henry.cooper@me.com
RO01	Rock On	BC1	Barbican Centre	London	06/06/2016	Alice	Jones	ajones@welovemusic.com
RO01	Rock On	O23	O_2 Academy	Liverpool	10/06/2016	Alice	Jones	ajones@welovemusic.com
SF01	Stage Fright	RA1	The Royal Albert Hall	London	20/06/2016	Bill	Richards	billyrich@amazingartists.com
JY01	Jimmy	RA1	The Royal Albert Hall	London	28/06/2016	Henry	Cooper	henry.cooper@me.com
TD01	The Delights	O21	O_2 Academy	Birmingham	13/06/2016	Alice	Jones	ajones@welovemusic.com
RO01	Rock On	HA1	Hammersmith Apollo	London	13/06/2016	Alice	Jones	ajones@welovemusic.com
JY01	Jimmy	BC1	Barbican Centre	London	30/06/2016	Henry	Cooper	henry.cooper@me.com

We have now made all the fields in the database atomic. We have done this by breaking down the venue name and location into two separate fields, and by breaking down the agent first and last name into two separate fields.

There is no single field in the database at present that is unique for each record. In order to create a unique identifier for each record we need to combine three fields together (they have been identified by *) to create a composite key. This composite key is a unique identifier for each record as only one artist will play at a particular venue on a given date.

The characteristics of 1NF are therefore:

- all fields in the database are atomic
- all repeating fields have been removed

- a unique identifier has been created for each record.

Now that we have taken our database to 1NF we can take it to 2NF. In 2NF our database looks like this:

ArtistDetails

Artist ID*	Artist Name	Agent First Name	Agent Last Name	Agent Email
RO01	Rock On	Alice	Jones	ajones@welovemusic.com
SF01	Stage Fright	Bill	Richards	billyrich@amazingartists.com
JS01	John Smith	Henry	Cooper	henry.cooper@me.com
JY01	Jimmy	Henry	Cooper	henry.cooper@me.com
TD01	The Delights	Alice	Jones	ajones@welovemusic.com

VenueDetails

Venue ID*	Concert Venue	Location of Venue
O21	O_2 Academy	Birmingham
HA1	Hammersmith Apollo	London
O22	O_2 Academy	Leeds
BC1	Barbican Centre	London
O23	O_2 Academy	Liverpool
RA1	The Royal Albert Hall	London

ArtistsBookings

Concert ID*	Artist ID	Venue ID	Concert Date
CON001	RO01	O21	01/06/2016
CON002	SF01	HA1	01/06/2016
CON003	JS01	O22	10/06/2016
CON004	RO01	BC1	06/06/2016
CON005	RO01	O23	10/06/2016
CON006	SF01	RA1	20/06/2016
CON007	JY01	RA1	28/06/2016
CON008	TD01	O21	13/06/2016
CON009	RO01	HA1	13/06/2016
CON010	JY01	BC1	30/06/2016

We have now separated our database into different tables. We have introduced a primary key to each database (marked with *). A primary key is a single field that acts as a unique identifier for

each record in the table. This means that the data entered into that field for each record must be different for every record. The data in each table is dependent on the primary key of that table:

- the **ArtistDetails** table contains details about each artist
- the **VenueDetails** table contains details about each venue
- the **ArtistsBookings** table contains details about each concert booking for each artist. An additional field **Concert ID** has been added to this table to create a primary key.

The fields that are in each of these tables are directly related to the primary key of that table.

The characteristics of 1NF are therefore:

- the database has the characteristics of 1NF
- related data has been separated into different tables
- each table has a primary key
- the fields in each table are dependent on the primary key.

Now we have taken our database to 2NF we can take it to 3NF. In 3NF our database looks like this:

ArtistDetails

Artist ID*	Artist Name	Agent ID
RO01	Rock On	AGAJ01
SF01	Stage Fright	AGBR01
JS01	John Smith	AGHC01
JY01	Jimmy	AGHC01
TD01	The Delights	AGAJ01

AgentDetails

Agent ID*	Agent First Name	Agent Last Name	Agent Email
AGAJ01	Alice	Jones	ajones@welovemusic.com
AGBR01	Bill	Richards	billyrich@amazingartists.com
AGHC01	Henry	Cooper	henry.cooper@me.com

VenueDetails

Venue ID*	Concert Venue	Location of Venue
O21	O_2 Academy	Birmingham
HA1	Hammersmith Apollo	London
O22	O_2 Academy	Leeds
BC1	Barbican Centre	London
O23	O_2 Academy	Liverpool
RA1	The Royal Albert Hall	London

ArtistsBookings

Concert ID*	Artist ID	Venue ID	Concert Date
CON001	RO01	O21	01/06/2016
CON002	SF01	HA1	01/06/2016
CON003	JS01	O22	10/06/2016
CON004	RO01	BC1	06/06/2016
CON005	RO01	O23	10/06/2016
CON006	SF01	RA1	20/06/2016
CON007	JY01	RA1	28/06/2016
CON008	TD01	O21	13/06/2016
CON009	RO01	HA1	13/06/2016
CON010	JY01	BC1	30/06/2016

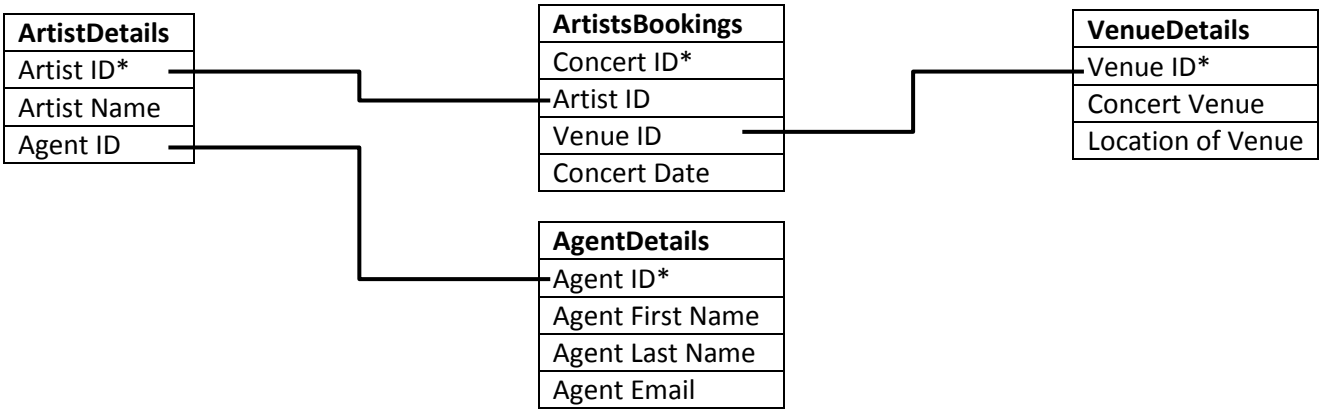
We have now separated any fields in the table that are not directly related to each other. In the **ArtistDetails** table, the fields relating to the artist were not directly related to the agent's details. We have therefore separated the agent details into another table. This table has also been given a primary key that is referenced in the artist's table.

The characteristics of 3NF are therefore:

- the database has the characteristics of 2NF
- fields have been further separated into tables that contain only fields that are directly related to each other.

By normalising our database we have removed any repeating fields and any repeating entries, and have separated any data that is not directly related to the primary key of the table or the other fields within that table.

We have taken our database through the three different stages of normalisation and can now create links between the tables to create a relational database:



Our database now allows bookings to be made for each artist at a concert venue. The bookings could be made using the **ArtistsBookings** table. The details for each artist and each venue are linked to the relevant table for those details. This is done through the use of their primary key appearing in this table as a foreign key.

A standard notation is used to represent the data structures of table in normalisation. The name of the table appears in capital letters and the field names that relate to it appear in brackets alongside, each one separated by a comma. The primary key of the table is underlined and any foreign keys are over-lined.

In this notation our data structure would be:

ARTISTDETAILS (Artist ID, Artist Name, Agent ID)

ARTISTSBOOKINGS (Concert ID, Artist ID, Venue ID, Concert Date)

VENUEDETAILS (Venue ID, Concert Venue, Location of Venue)

AGENTDETAILS (Agent ID, Agent First Name, Agent Last Name, Agent Email)

4. Exam preparation

Learners need to be able to recognise data in each given form and understand how to normalise data to third normal form. Learners are required to understand the theory of normalising a database to third normal form, as well having the ability to carry out a practical demonstration of this. Learners also need to understand the advantages and disadvantages of normalisation.

The most effective way to prepare learners for an assessment of this topic is to have them carry out the electronic practical examples. These require learners to normalise a database, in a given form, to third normal form. Learners should also practice the sample test questions to understand how normalisation can be tested in a theory examination.

5. Further resources

Useful websites

https://support.microsoft.com/en-us/kb/283878

http://databases.about.com/od/specificproducts/a/normalization.htm

http://www.teach-ict.com/as_a2_ict_new/ocr/AS_G061/315_database_concepts/normalisation/home_normalisation.html

6. Class and homework activities

6.1 Activities for consolidating this topic

The following activities are suggested to consolidate your students' learning after they have studied this topic:

6.1.1 Car hire firm

A car hire firm wants to store data about its cars and the customers that hire them. When a customer hires a car, the hire is given a Hire_ID. A customer can hire more than one car under the same Hire_ID. Each car can only be hired to one customer at a time. Each hire is made by a salesman.
The car hire database is currently in 1NF. Convert it to 3NF:
HIRE_CARS(Hire_ID, Hire_Date, Hire_Cost, Salesman_ID, Salesman_FirstName, Customer_ID, Customer_FirstName, Customer_LastName, Customer_Email, Car_Reg, Car_Make, Car_Model)
You can use the source file *Hire_Cars.csv* to demonstrate the normalisation process.

6.1 2. Veterinary surgery

A veterinary surgery treats a variety of pets. The surgery has several vets. Each pet has one owner, but an owner may have several pets. When a pet needs treatment, its owner makes an appointment. Each appointment has a time and date, a vet, a pet and an owner.
The unnormalised data for the surgery's database is as follows:
APPOINTMENTS(Appointment_ID, Appointment_Time, Appointment_Date, Vet_Name, Owner_ID, Owner_Name, Owner_Email, Pet_Name, Type_of_Pet)

Take this unnormalised data through to 3NF.
You can use the source file *Appointments.csv* to demonstrate the normalisation process.

6.1.3 Quiz

1. Normalisation is:
   a. Removing all necessary data from a database
   b. Organising a database to remove repeated entries and increase the accuracy of the data
   c. Putting fields from different tables into one big database

2. Which of the following statements fully describes data in 2NF:
   a. When any repeating fields have been removed and the table is given a primary key
   b. When all repeating entries of data are removed and the fields in each table are directly related to the primary key and no fields are present that are not related to each other
   c. When all the fields in each table are directly related to the primary key

3. An advantage of normalisation is:
   a. The storage space needed for a normalised database is likely to be smaller
   b. The process of normalising a database is very easy and doesn't require any skill
   c. It is possible to see previous details that have been changed, such as a customer's address, as we can look back at an older entry

4. A disadvantage of normalisation is:

a. The data loses its integrity as some of it is removed in the normalisation process
b. The process of searching the database may be slower due to a higher demand on the central processing unit (CPU)
c. Removing redundant data means that links cannot be created between tables

5. What is an atomic field?
   a. A field that contains multiple items of data
   b. A field that is repeated
   c. A field that contains only one item of data

6. What is a foreign key?
   a. A unique identifier in a database
   b. A primary key of one table that appears in another table
   c. A field that should not be in a table and needs removing

As well as these activities, the following resources are available for this topic:

- a crossword on normalisation
- normalisation dominoes.

6.2 End of topic test questions

1. Describe a database in first normal form (1NF) [4 marks]

2. Explain two advantages and one disadvantage of normalisation of a database [6 marks]

3. Explain why the following database is not in second normal form (2NF) [4 marks]

Customer First Name	Customer Last Name	Seat Number*	Artist	Date*	Time*
Jenny	Khan	C12	Rock On	13/06/2016	7:30pm
Billy	Thornton	F17	Rock On	13/06/2016	7:30pm
Deeraj	Smith	C12	Rock On	13/06/2016	2:30pm
Maria	Lopez	A20	Rock On	13/06/2016	7:30pm
Freddie	Pascal	B13	Rock On	13/06/2016	2:30pm

Cambridge International Examinations
1 Hills Road, Cambridge, CB1 2EU, United Kingdom
Tel: +44 (0)1223 553554 Fax: +44 (0)1223 553558
Email: info@cie.org.uk www.cie.org.uk