University of Salford MANCHESTER

Database Systems

CRN 32741, UMC G400 10045

Dr Bryant

2023/2024

Preface

This booklet was prepared using LaTeX 2_ε on 15th September, 2023 .

0.1 Bookmarks

This booklet includes bookmarks for chapters, sections and subsections. If you are using the Adobe Acrobat Reader, then these may be displayed in the Navigation Pane on the left hand side. You can control whether the bookmarks are displayed by selecting the option from the menu at: View → Show/Hide → navigation panes.

0.2 Hypertext Links

This booklet contains hypertext links for web addresses, citations, page references, section references, table references and figure references. The links are displayed in colour. The colour depends on the the type of link.

magenta (deep purplish red) for web addresses.

cyan (light blue) for bibliographical citations in text.

red for all the other types of links.

The magenta hypertext links for web addresses will only work if the security settings of the PDF reader you are using permit it to display a web page using a web browser.

Note that Adobe Acrobat Reader includes a back button which you may find very useful after following one of these links. You can control whether the back button is displayed on the tool bar by selecting the option from the menu at: View → Show/Hide → Toolbar Items → Show Page Navigation Tools → Previous View

0.3 The Inclusive Student Experience Project

This booklet has been made inclusive and accessible as follows.

- Headings are consistently positioned to one side and can be clearly differentiated from the body of the text.
- Paragraph text is consistent in size and style.
- Neither capitals or the size or colour of text has been used to emphasise a point.
- The default font size is 12 points.
- A sans serif font has been used.
- Images have only been used when they are necessary or enhance understanding of written content.
- All tables have a legend.

Contents

List of Tables

Part I

Introduction

Chapter 1

Module Handbook

This module will share fundamental knowledge of database management systems, their design, implementation and applications. It will develop your knowledge and understanding of the underlying principles of relational database management system, and how to implement and maintain an efficient database system.

1.1 Programmes Taking the Module

- BSc (Hons) Computer Networks
- BSc (Hons) Computer Science
- BSc (Hons) Computer Science with Cyber Security
- BSc (Hons) Software Engineering

1.2 Aims of the Module

- To provide you with fundamental knowledge of database management systems, their design, implementation and their applications.
- To develop your knowledge and understanding of the underlying principles of Relational Database Management System.
- To demonstrate database trilingualism in: the basic algebraic operations, a standard query language and English.
- To build up your ability to learn DBMS advanced features.
- To build up your ability to implement and maintain an efficient database system using emerging trends.

1.3 Learning Outcomes

Upon successful completion of the module, you will be able to:

- Describe the basic concepts of relational database design.
- Describe, create, populate and maintain a database and provide controlled access to it.
- Obtain information from a database using a standard query language.
- Describe the process of database query processing and evaluation.
- Explain the concepts of transaction management and concurrency control.
- Discuss database security and recovery.

1.4 Textbooks and Teaching Materials

Lecture slides, datasets and the exercise booklet containing the tutorial and workshop will be made available on Blackboard. For more information about Blackboard, see Section 2.5 on page 10.

The remainder of this section concerns text books. Specific details of recommended reading are given at the end of nearly every lecture. The recommendations are listed on a slide near the end of each presentation in the booklet of lecture slides. You may be recommended to read from the following books:

- (Connolly & Begg, 2004),
- (Connolly & Begg, 2014),
- (Silberschatz et al., 2019) and
- (Donahoo & Speegle, 2005).

Generally speaking, it is not normally necessary for a student to purchase the core text books for this module because they are available in electronic or/an paper format from the library.

1.5 Software Used on this Module

The software used on the module is called Microsoft (MS) Access, which you may access in the following ways.

Computers provided by the University of Salford MS Access is installed on the Windows operating system on the computers in the laboratories on campus.

Your Own Computer You can install Office 365 on your own personal device by logging in to https://www.office.com/ and clicking "Install". Please note that this option is not available to students with macOS because MS Access is not natively supported in macOS.

Chapter 3 will introduce you to MS Access.

1.6 Provisional Schedule

Tables 1.1 and 1.2 show the provisional schedule for the classes.

Week	Lectures and Tutorials			Workshop
	1^{st} Hour	2^{nd} Hour	3^{rd} Hour	
T1.1	Organisation	Introduction		Chapter 3
T1.2	Queries In Relational Databases	Chapter 4		Chapter 5
T1.3	More SQL	Chapter 6		Chapter 7
T1.4	Data Description Language	Chapter 8		Chapter 9
T1.5	Conceptual Modelling	Chapter 10	Chapter 11	
T1.6	Logical (Relational) Modelling	Chapter 12	Chapter 13	
T1.7	Normalisation	Chapter 14	C/w Briefing	
T1.8	SQL Subqueries	Chapter 15		Chapter 16
T1.9	Chapter 17	Chapter 18	C/w Queries	
T1.10	Relational Algebra	Chapter 19		Chapter 20
T1.11	Enhanced modelling	Chapter 21	Chapter 22	

Table 1.1: Provisional schedule for Semester One.

1.7 Surgeries

If you want to have a one-to-one conversation with Dr Bryant then please attend one of his surgeries. Please note that this is a drop-in service. In other words, there is no need to make an appointment. The time and location of his next surgery is shown on Blackboard's calendar, which you find at:
Blackboard
$\longrightarrow$ Database Systems
$\longrightarrow$ Calendar tab (at the top on the left-hand-side.)
$\longrightarrow$ Click on the button "Month".

Week	Lectures and Tutorials			Workshop
	1^{st} Hour	2^{nd} Hour	3^{rd} Hour	
T2.1 (18)	Organisation	Relational Algebra	Optimisation	
T2.2 (19)	Statistics&Indexes	Chapter 23		Chapter 24
T2.3 (20)	Transactions	Concurrency	Chapter 25	
T2.4 (21)	Concurrency Issues	Chapter 26		Chapter 27
T2.5 (22)	Precedence Graphs		Chapter 28	
T2.6 (23)	Locks	Chapter 29		Chapter 30
T2.7 (24)	Deadlock	Chapter 31	Recovery	
T2.8 (25)	Recovery (Part 2)	Chapter 32		Chapter 33
T2.9 (26)	Security (Part 1)	Chapter 34	C/w Briefing	
T2.10 (27)	Chapter 35	Chapter 36		Catch-up
Easter				
T2.11 (31)	Security (Part 2)	Chapter 37	C/w Queries	

Table 1.2: Provisional schedule for Semester Two.

1.8 Recording of Teaching Sessions

Students are prohibited from **video** recording classes. In other words, you are not allowed to do this. Students who wish to **audio** record teaching sessions must comply with the rules set out here. These rules apply to all students of the University. These rules apply to any teaching session including lectures, seminars, tutorials (group or one to one), discussion groups, supervision sessions, laboratory work, fieldwork or other learning activity.

1.9 Assessment Methodology

Each level of an undergraduate degree programme consists of 120 credits. Each level is divided into a number of modules. This module is worth 20 credits.

Assessment of your performance in the module will be by the following method:

- During Semester One, you will be assessed by coursework on material covered during Semester One. This coursework will contribute 50% of your overall module mark. The provisional dates are:

  Issue date: Tuesday 7^{th} November 2023, Week T1.8

  Submission date: 4pm Wednesday 6^{th} December 2023, Week T1.12.

- During Semester Two, you will be assessed by coursework on material covered during Semester Two. This coursework will contribute 50% of your overall module mark. The provisional dates are:

Issue date: Tuesday 12^{th} March 2024, Week 26 (T2.9)

Submission date: 4pm Wednesday 17^{th} April 2024, Week 31 (T2.11).

To pass the module you must achieve a mark of 40% or greater overall.

Chapter 2

Where to Get Help

2.1 Aim of the Exercise

Dr Bryant and your laboratory tutor can answer your questions regarding databases. The purpose of this exercise is to check that you know where to get help on other matters.

2.2 Exercise

- Check that you have a Student ID card and that it grants you access to the laboratory where your workshop is taking place. If not, take remedial action (see Section 2.3).
- Make sure you can log into the computers in the laboratory. If not, take remedial action (see Section 2.4).
- Verify that you have access to the part of Blackboard for this module. If not, take remedial action (see Section 2.5).
- If you are planning to use the electronic copies of the text books then check that you can access them (see Section 2.6).
- Set up an email signature that contains the information listed in Section 2.11.1.
- If necessary, spend some time making yourself familiar with the route to the rooms where your classes for this module are taking place (see Section 2.7).

2.3 Student ID Cards

You may need a Student ID to enter the laboratory where your workshop is taking place. If your card will not grant you access to your laboratory then please contact the School

Office (see Section 2.11.5). If your card is lost or damaged then please purchase a replacement card via the web shop at http://shop.salford.ac.uk/ by clicking on:
Product Catalogue $\longrightarrow$ Student Administration $\longrightarrow$ Replacement Student ID Card
Please note that Dr Bryant has no involvement with the issue and maintenance of these cards. Consequently he cannot rectify any problems with them.

2.4 Computers in the Laboratory

You will need a User name and password to use the computers in the laboratory. If you have registered as a student at this University then your should already have these. Please raise any problems with your account, email, the operating system, the laboratory hardware or printers with Digital IT. Digital IT aim to resolve all cases which are logged and then communicate the resolution to you.

Please note that Dr Bryant is not part of Digital IT. Dr Bryant has no involvement with the set up and maintenance of the laboratory hardware, operating system, printers or User accounts. Consequently he cannot rectify any problems with them. For obvious security reasons, Dr Bryant does not know what your account password is.

2.4.1 How do I report problems to Digital IT?

When reporting issues to DigitalIT you need to supply the following information:

- your contact details,
- a description of the problem, and
- the date and time when the problem occurred.

When reporting issues with a computer provided by the university you also need to supply the asset tag, which is usually displayed on the side or back of the computer, rather than the monitor/screen.

It is important to request a case-reference number. You are strongly advised to keep a record of both your communication with Digital IT and your case-reference number. The contact details of Digital IT are given in Section 2.11.3.

2.5 Blackboard

The university's chosen standard virtual learning environment is called Blackboard. Electronic copies of some of the teaching materials will be made available on Blackboard. Dr Bryant may use Blackboard's facilities for communicating with students such as announcements and email.

Where is the best place to seek help with Blackboard? This depends upon what your problem is.

You find Blackboard confusing. If you don't know how to use Blackboard then help is available at https://www.salford.ac.uk/library/know-how/blackboard.

You cannot log into Blackboard Please contact Digital IT (see Section 2.11.3).

You cannot find this module on Blackboard. If you cannot see this module on Blackboard then this may be because there is a problem with your registration. Please ask Digital IT (see Section 2.11.3) whether your module enrolment details are correct.

Blackboard is broken. Please contact Digital IT (see Section 2.11.3).

You cannot find the teaching materials. If you can see this module on Blackboard but you cannot find the teaching materials for this module then please ask your laboratory tutor to help you.

Dr Bryant is not part of the support team responsible for maintaining Blackboard. Dr Bryant cannot fix Blackboard if it is not working properly.

2.6 Library Resources

Copies of the books listed in Section 1.4 are available from the library. Some books are available in electronic format. To access an e-Book off campus you should expect to have to login. If you need help then please contact the library (see Section 2.11.4).

Please note that Dr Bryant is not a librarian. Dr Bryant has no involvement with the maintenance of the library and its resources. Consequently he cannot rectify any problems with them. For obvious security reasons, Dr Bryant does not know your password.

If you plan to purchase your own copy of a textbook then please note that some books are available from the Inspire on-line shop at http://www.salford-inspire.co.uk.

2.7 Directions to the Classes

Your timetable should show the rooms where your classes for this module are taking place. If necessary, spend some time making yourself familiar with the route to these rooms. (As there is only a ten minute gap between consecutive classes, you may not have time to discover the direct routes if you leave it until the last minute.) Unfortunately Dr Bryant does not have time to answer emails requesting directions to rooms or buildings on campus. You can find a map at: http://www.salford.ac.uk/about-us/travel
If you find yourself lost in a building then a good place to ask for help is the reception.

It is your responsibility to make sure you get up early enough to allow sufficient time for you to travel to your first class of the day. Alarm clocks are available from the Inspire on-line shop at http://www.salford-inspire.co.uk

2.8 What should I do if I am ill?

Do not enter the classroom if you are ill, especially if you believe that you may have an infection or contagion. If you are seeking authorised absence due to illness then you should send an email message to SEE-Engage@salford.ac.uk and copy in your Programme Leader.

2.9 How can I catch up if I miss a class?

It is your responsibility to keep up-to-date. If you fall behind then you should endeavour to catch up. Tables 1.1 and 1.2 show the provisional schedule for the classes. This tells you what need to do if you miss a class. Unless Dr Bryant tells you otherwise, you may assume that we are adhering to this schedule. You must keep up-to-date with the tutorials and workshop; otherwise you will find it difficult to perform well in the assessments.

2.10 Disabled Students

Disability & Learner Support (see Section 2.11.6) exists to support disabled students throughout their studies at the university. Disabled students are kindly invited to discuss their support requirements for *this particular module* with Dr Bryant should they so wish to do so. The best time to do this is during one of his surgeries (see Section 1.7 on page 5). Advice on how to change the appearance of Access to suit your special needs can be found in Section 3.10 on page 21.

2.11 Contact Details

2.11.1 Guidance on How to Contact Staff

Face-to-face If you decide to visit any of the people mentioned in this section in person then remember to take your ID card with you.

Email When composing a message to send to any of the email addresses given in this section:

- use your university email account;
- write sentences of English which are lucid, grammatically correct and complete;
- strive to avoid spelling mistakes;
- use an appropriate salutation. (An example of an appropriate salutation would be “Dear Dr Bryant” and an example of an inappropriate one would be “Hey”.)
- Include the following information at the end your message.
  - Your name.
  - Your student roll number. (Student roll numbers usually comprise the @ symbol followed by eight digits.)
  - Your User ID. (User IDs usually comprise three characters followed by three digits.)
  - Your programme (see Section 1.1).
  - The level at which you are studying, namely, Level 4.

MS Teams If you are new to MS Teams then please note that you can find an introduction to MS Teams on the link here to linkedin.com. Please remember to have your microphone switched on if you want to be heard and to have your webcam switched on if you want to be seen. Please check that you can get your webcam and microphone to work with MS Teams before you try to contact a member of staff. A very helpful video that tells you how to do this can be found on the link here to linkedin.com. If you do not have a microphone then please note that headsets are available from the Inspire on-line shop at http://www.salford-inspire.co.uk. A headset includes headphones and a microphone.

2.11.2 Contact Details of Dr Bryant

Face-to-face on campus: Attend his surgery (see Section 1.7 on page 5).

Email: c.h.bryant@salford.ac.uk

Virtual face-to-face: MS Teams

If you just want to send Dr Bryant an written message, then do not use MS Teams or Blackboard to do this. Instead, please send it to his email address.

2.11.3 Contact Details of Digital IT

Face-to-face on campus: click here

Email: Digital-ITServicedesk@salford.ac.uk

Telephone: 0161 295 2444

Web: https://testlivesalfordac.sharepoint.com/sites/Uos_Students/SitePages/Digital-IT.aspx

2.11.4 Contact Details of The Library

Face-to-face on campus: Clifford Whitworth Building, Peel Park Campus.

Email: library-SEE@salford.ac.uk

Web: http://www.salford.ac.uk/library

2.11.5 Contact Details of The School Office

Face-to-face on campus: Room 01.07, First Floor, SoSEE building, Peel Park Campus.

Email: SEESchoolEnquiries@salford.ac.uk

Telephone: 0161 295 5338

Web click here

2.11.6 Contact Details of Disability & Learner Support

Face-to-face on campus: University House, Peel Park Campus.

Email: disability@salford.ac.uk

Telephone: 0161 295 0023 (option 1, option 2)

Web: https://www.salford.ac.uk/askus/topics/disability-inclusion-service

Part II

Semester One

Chapter 3

Laboratory Exercise: Introduction to Microsoft Access

3.1 Aim

A Database Management System (DBMS) is a software system that enables users to define, create, and maintain the database and which provides controlled access to this database. During this module you will use a relational DBMS called Microsoft Access Database. For the sake of brevity, this software will be referred to simply as Access throughout the rest of this document. The aim of this laboratory exercise is for you to learn how to view the tables in a Access database and run SQL queries in Access.

3.2 Opening a Sample Access Database

1. Download the file “NWind.accdb” from Blackboard. This file contains a database called NWind. Make a note of where you have downloaded it to.

2. Open Access.

3. Click on “Open”. Navigate to wherever you stored the NWind database in step 1.

4. Once the database is opened, you may encounter a security warning at the top of the application. If so, click on options and select “Enable content”.

5. If you click on Database Tools and select Relationships you can see that the database has 8 tables and a large number of attributes. You can see which attributes are in which tables and how the tables are linked to each other.

3.3 Viewing Tables in an Access Database

- To examine the data in a table, simply double click on the name of the table. This allows you to look at a table in what Access calls “Datasheet View”.

- To examine the design of a table, right click on the name of the table and select design view from the pull-down menu. Note there are some subtle differences between the names of the attributes as they appear in the “Datasheet View” and how they appear in the “Design View”. In subsequent laboratory exercises you will write SQL statements to query this database. When you do so, you must use the attribute names as they appear the “Design View”.

- Notice that the “Design View” uses the key symbol. *Primary keys* uniquely identify each row (or tuple or record) of a relation. What is the primary key of the table (i.e., relation) called Shippers?

- An attribute that is linked to the primary key of another table is known as a *foreign key*. How can you use Access to see the foreign keys in the database? Recall from the previous section that if you click on Database Tools and select Relationships you can see how the tables are linked to each other.

3.4 SQL

There are many relational DBMSs. Fortunately they use a common language called SQL. SQL is correctly pronounced S–Q–L; however many people also pronounce it *see-kwel*.

To try to ensure that all relational DBMSs speak exactly the same language, a standard form of SQL has been agreed upon by organisations such as International Organisation for Standards (ISO) and American National Standards Institute (ANSI). As far as possible, this module will focus on standard SQL.

However you should be aware that each DBMS does things slightly differently; most of them do not follow the standard exactly. There are many reasons why a DBMS may deviate from the standard, relating to things such as performance, legacy or marketing.

3.5 Proprietor Specific Aspects of Access

Please note that this module is not a course on Access. Hence the focus of the exercises will not be on proprietor specific aspects of Access. The main motivation for using Access is to allow you to observe the effect of executing SQL queries on a computer. As far as possible, the laboratory exercises will focus on standard SQL.

3.6 Running SQL Queries in Access

The aim of this section is for you to learn how to run SQL queries in Access.

1. To begin writing queries we need to click on “Create” on the top menu and then select “Query Design” from the right hand side of the toolbar.

2. You are going to type in SQL queries. Consequently you do not need the design GUI. So close the window called “Show Table” and then right click in the top half of the query1 window. From the menu that pops up, select SQL View. A window should open.

3. Type in the following query.

   - SELECT *
     FROM Customers;

4. After typing the query, you can execute it by clicking the Run icon.

3.7 Examples of SQL Queries

The aim of this section is for you to gain experience of running SQL queries to retrieve specific information from a relation database. **Execute** the following list of SQL queries. Each query is first expressed in English (to aid your understanding) and then is expressed in SQL. It is the SQL statements, i.e., the text after each bullet point, that you have to enter into the SQL View window mentioned near the end of the last section.

1. Display information about all customers in the database.

   - SELECT *
     FROM Customers;

2. Display information about customers who are from Brazil.

   - SELECT *
     FROM Customers
     WHERE Country = “Brazil”;

3. Display company name and contact number of customers who are from Germany.

   - SELECT CompanyName, Phone
     FROM Customers
     WHERE Country = “Germany”;

Notice that all of the SQL statements end with a semi colon(;). Also note that the attribute names used in the queries are those given in the “Design View”. Recall from Section 3.3 on page 18 that there are some subtle differences between the names of the attributes as they appear in the “Datasheet View” and how they appear in the “Design View”.

3.8 Formatting SQL queries

3.8.1 White space is not significant

Note that in the examples given in Section 3.7, each SQL statement was broken into separate lines, one for each part of the SQL statement. This is considered good practice. You are recommended to do this because it makes it easier for other people to read your SQL queries. This is a convention designed to help humans, not the computer. SQL is just as happy if an entire SQL statement is written on one line. SQL also allows multiple spaces, tabs, or other white space characters to separate elements in the query.

3.8.2 Are SQL statements case sensitive?

Note that in the examples given in Section 3.7 on page 19, SQL keywords were written in upper case. Table names and column names were not written in upper case. This is considered good practice. You are recommended to do this because it makes it easier for other people to read your SQL queries. This is a convention designed to help humans, not the computer. SQL keywords, table names and columns are, in fact, case insensitive, although character data may be sensitive to case when it is stored and compared.

3.9 Saving Your SQL Queries

Create a file in OneDrive called ‘DatabaseSystemsWorkshop.txt’. Copy your queries from the MS Access interface, paste them into this file and save it to OneDrive before you log off.

3.9.1 OneDrive

OneDrive is a place where you can store files which you can access from any computer in any of the CS&SE laboratories at any time during the rest of the academic year. This is a useful place to keep a copy of SQL queries that you write. For more information:

1. go to https://www.salford.ac.uk/skills/it-skills/microsoft-office-courses;

2. click on the box “ Learn about O365 & OneDrive”;
3. under “Get Started with Office 365”, click on “Store and sync files”; and
4. watch the video.

3.9.2 What to do when the network traffic is high

You access OneDrive across a network because it is not physically located in the laboratory. Unfortunately the network connection may be very slow during busy periods. When this happens, to avoid delays, you are recommended to store your work on the C: drive of the computer in front of you for the duration of each class. At the end of each class, you should copy your work to OneDrive.

3.10 Customise the Appearance of MS Access to Suit your Needs

If you want change the appearance of MS Access to suit your special needs then you can do this as follows.

1. Open Access.
2. Select the “File” pane on the left side.
3. Click on “Options” and make selections to suit your needs.
4. E.g., to change the size of font in which data is displayed, go to “Datasheet”.
5. E.g., to change the size of font in which code is displayed, go to “Object Designs”.

Chapter 4

Tutorial: SQL Queries

4.1 Aim

A query is a question that you want to ask about the data in a database. In relational databases, querying can be done using the SQL SELECT query. The aim of this chapter is for you to gain experience of devising SQL queries.

4.2 Types of Joins

Suppose that a database contains the relations shown in Tables 4.1, 4.2 and 4.3.

part-number	name	description
1000	pin	plain pin
1001	stud	threaded pin
1002	ring	fibre ring

Table 4.1: A relation called parts.

order-number	project-id	part-number	quantity
100	p2	1000	450
200	p3	1002	234
300	p4	1000	987

Table 4.2: A relation called orders.

Write down the result of:

1. the equijoin of the parts table with the orders table on part-number;
2. the left outer join of the parts table with the orders table on part-number;

project-id	title
p1	Nemesis
p2	Ranger
p3	AVSC

Table 4.3: A relation called projects.

3. the equijoin of the projects table with the orders table on project-id;
4. the right outer join of the projects table with the orders table on project-id.

4.3 Pillaging-Pirates

Sea pirates commit robbery or hijacking aboard a ship. Piracy occasionally hits the news headlines due to the activities of contemporary pirates operating off the coast of Africa. The database used in this chapter contains data on historical pirates. stored in the relations shown in Tables 4.4, 4.5, 4.6 and 4.7.

p-code	p-name	legs	eyes	origin
p1	Blue Peter	2	2	British
p2	One-Eyed Jack	2	1	American
p3	Black Jack	1	2	Spanish
p4	Blind Pierre	2	0	French

Table 4.4: A relation called pirates.

boat-code	boat-name	boat-type	origin
b1	Saucy Sue	Barque	British
b2	Hispaniola	Spanish Galleon	Spanish
b3	Narcissus	Clipper	American
b4	Don Fanucci	Merchantman	Italian
b5	Reine de la Mer	Merchantman	French
b6	Conquistador	Spanish Galleon	Spanish
b7	Ubiquitous	Man O' War	British
b8	Chrysos	Argosy	Spanish

Table 4.5: A relation called boats.

1. Write down SQL expressions that will implement each of the following single table queries.

   (a) List full details of all pirates.

(b) List full details of all booty sorted by value.

(c) List the names and types of all Spanish boats.

(d) List full details of all pirates whose names start with “B”.

(e) List the names of all pirates with less than two eyes or two legs or both.

booty-code	booty-name	value
bt1	Oriental Spices	10000
bt2	Silver Plate	50000
bt3	Gold Sovereigns	80000
bt4	Gold Doubloons	55000
bt5	Bags of Turnip	100
bt6	Chests of Tea	10000
bt7	Precious Gems	100000
bt8	Nothing	0
bt9	Weevil-Ridden Biscuits	10
bt10	Barrels of Pemmican	2000
bt11	Fresh Fruit	10000
bt12	Rotten Fruit	100

Table 4.6: A relation called booty.

p-code	boat-code	booty-code	date
p1	b1	bt5	1735-07-06
p1	b1	bt8	1735-07-07
p2	b3	bt1	1745-05-08
p2	b3	bt2	1745-05-08
p2	b3	bt6	1745-05-08
p3	b5	bt2	1750-08-08
p3	b5	bt3	1750-08-08
p4	b5	bt8	1750-08-09
p4	b2	bt4	1743-06-30
p4	b2	bt7	1743-06-30
p4	b5	bt3	1744-09-10
p4	b6	bt2	1745-05-12

Table 4.7: A relation called pillage.

2. In each of the following decide which tables would have to be joined in order to retrieve the information required and then write down SQL expressions that implement the query.

   (a) List the names of pirates who have pillaged booty of type bt3.

   (b) List the names of pirates who have pillaged “Gold Doubloon”.

4.3.1 Challenge Questions

1. List the names of the boats pillaged by either a French or Spanish pirate.
2. List the names of the boats pillaged by either a French or Spanish pirate for either Silver Plate or Gold Doubloons.

4.3.2 Glossary

Argosy - a large abundantly laden merchant ship.

Barque - a sailing ship having three or more masts arranged in a particular way.

Clipper - a fast sailing ship.

Doubloon - a former Spanish gold coin.

Galleon - a large sailing ship having three or more masts arranged in a particular way.

Man O' War - a war ship.

Pemmican - an early convenience food made of dried venison or buffalo meat that was pounded to a powder, mixed with cranberry paste, formed into a cake and left to dry in the sun. Pemmican was a favorite of pirates because it lasted a long time.

Chapter 5

Laboratory Exercises on SQL Queries

5.1 Aim

A query is a question that you want to ask about the data in a database. In relational databases, querying can be done using the SQL SELECT query. The aim of this laboratory exercise is for you to gain experience of designing and then *executing* SQL queries.

5.2 Dealing with Duplicate Values

You just have to read this section. Do **not** try to execute the queries in this section using Access. You will execute some queries later when you use do the exercise in Section 5.3. Suppose a database includes the relation shown in Table 5.1. Executing the SQL statement

```
SELECT boat-type
FROM boats;
```

would result in the output shown in Table 5.2. We could eliminate duplicate rows from the result if we added the SQL reserved word DISTINCT as follows.

```
SELECT DISTINCT boat-type
FROM boats;
```

Executing this SQL statement would result in the output shown in Table 5.3.

boat-code	boat-name	boat-type	origin
b1	Saucy Sue	Barque	British
b2	Hispaniola	Spanish Galleon	Spanish
b3	Narcissus	Clipper	American
b4	Don Fanucci	Merchantman	Italian
b5	Reine de la Mer	Merchantman	French
b6	Conquistador	Spanish Galleon	Spanish
b7	Ubiquitous	Man O' War	British
b8	Chrysos	Argosy	Spanish

Table 5.1: A relation called boats.

boat-type
Barque
Spanish Galleon
Clipper
Merchantman
Merchantman
Spanish Galleon
Man O' War
Argosy

Table 5.2: Output from an SQL query.

boat-type
Barque
Spanish Galleon
Clipper
Merchantman
Man O' War
Argosy

Table 5.3: Output from an SQL query that removes duplicates.

5.3 Exercise

The aim of this section is for you to gain experience of running SQL queries which you have written yourself. Open Access and load the NWind.accdb database. (If you cannot recall how to do this then refer to Section 3.2 on page 17.) **Execute** SQL statements which answer the following queries.

5.3.1 Single-Table Queries

1. Display names and quantities of products in the database. Sort the results in order of increasing quantity.
2. Display the names and addresses of companies who are customers from France.
3. Display the contact names of customers who are from UK and whose contact title is “Sales Representative”.

5.3.2 Multi-Table Queries

1. List the names of the products which are supplied by a supplier in Manchester.
2. List the names and telephone numbers of shippers who have shipped orders to Bern.

5.3.3 Challenge Question

There are just five orders in which both the customer and employee come from the same city. These five orders involve just two cities. Write an SQL query which lists the names of these cities.

Chapter 6

Tutorial: More SQL

6.1 Aim

The aim of this chapter is for you to:

1. learn more about SQL data types;
2. gain experience of writing SQL statements that add, delete and modify data in a relational database; and
3. learn more about aggregate functions.

6.2 Data Types

You just have to read this section. You will use the knowledge you acquire from this section later when you use do the exercise in Section 6.3.

Every column of a table (i.e., attribute of a relation) has a declared data type which specifies what kind of data the column may contain. There are five categories of SQL data types.

6.2.1 Character String

Attributes such as names and addresses are often represented by strings of characters. The most common SQL data types for strings of characters are:

CHARACTER(L) A fixed-length character string containing exactly L characters. If the string contains fewer characters, then the remaining characters contain padding characters. The padding characters are usually spaces. CHARACTER is often abbreviated to **CHAR**.

CHARACTER VARYING(L) A variable-length character string that may hold up to L characters. Only the specified number of characters are stored, so there is never any padding. CHARACTER VARYING is often abbreviated to **VARCHAR**.

6.2.2 Numeric

Attributes such as age and salary are usually represented by numeric data types. The most common SQL numeric data types are:

INTEGER A signed whole number. The range of possible values depends upon which DBMS is being used. INTEGER may be abbreviated as **INT**.

NUMERIC(P,S) A signed, fixed-point number where P (precision) specifies the total number of digits in the number and S (scale) specifies the number of digits to the right of the decimal place. E.g., NUMERIC(5,2) specifies a type ranging from -999.99 to 999.99.

A database should not allow you to use a number with an absolute value that is too large for the data type. E.g., it should not allow you to put -1000 into a NUMERIC(5,2) field. If you use a number with too many digits to the right of the decimal point, the DBMS may either truncate or round the value. So if you put 0.0023 in a NUMERIC(5,2) field, the value stored in the database will be 0.00. A number with fewer digits than the limit of the data type will be stored without change. E.g., 7.5 fits in a NUMERIC(5,2) field.

6.2.3 Temporal

DATE YYYY-MM-DD A date represented by four digits for the year (1–9999), two digits for the month (1–12) and two digits for the date (1–31). Use this type when you do not care about the time of an event, e.g., birthday.

TIME HH:MM:SS A time represented by two digits for the hour, two digits for the minute, two digits for the second, plus optional fractional digits. Use this type when you do not care about the date, e.g., time a cafe opens to serves lunch.

TIMESTAMP YYYY-MM-DD HH:MM:SS Contains the date and time. Use this type when you need to record the date and time of an event, e.g., the time an order is placed.

INTERVAL Refers to a period of time, e.g., a warranty period. May have a value such as 90 days. Could more accurately be termed a span.

6.2.4 Large Objects

SQL binary types are designed to store sequences of binary digits. Binary types are commonly used for photographs, sounds and movies. One such type is:

BINARY LARGE OBJECT(L) A large, variable-length binary string that may hold up to L bytes. BINARY LARGE OBJECT is often abbreviated to **BLOB**.

6.2.5 Boolean

The name Boolean commemorates George Boole (1815-1864) who first placed the study of logic on a sound mathematical basis. We use logic to reason about truth. Sometimes we just want to record (in a database) whether something is true or not. SQL has a data type which allows us to do this:

BOOLEAN A type which can only have one of three values: true, false and unknown.

6.2.6 NULL

Relational databases provide a special value, called NULL, which indicates that a field's value is unknown. Note that a database value of NULL is not the same as a space or zero. Unless explicitly forbidden, NULL is a valid value for any data type. We will study NULL later in the module.

6.3 Exercise on Data Types

1. Suggest a data type for storing each of the following.
   (a) Street
   (b) Bank holidays.
   (c) The outcome of tossing a coin.
   (d) The outcome of throwing a dice.
   (e) Numbers that can range from -9854 to 9860.78.
   (f) Screen dumps.
2. What is the difference between the literal 'Bryant' stored as CHAR(10) and stored as VARCHAR(10)?
3. Can an attribute of type INTEGER have the value NULL?

6.4 Exercise on Data Types and Keys

Suppose a library needs to store data in the following relations.

book The relation book is to store the following information about the books held in a library: ISBN, title, author.

member The relation member is to store the following information about the members of a library: membership number, name, age, address and telephone number.

loan The relation loan is needed to store information about the borrowing of books from the library. Each row must contain data specifying the book borrowed, the borrower, the date borrowed and the date returned. The relation has two foreign key attributes, one linked to the relation book, and one to the relation members.

Note the library only has one copy of each book.

For each of the relations listed above, invent an appropriate list of attribute names. Underline primary keys. Highlight foreign keys using an asterisk, i.e., *. Suggest a suitable data type for each attribute.

6.5 Exercise on Adding, Deleting and Modifying Data

In Chapter 4 you devised SQL queries to retrieve specific information from relational databases. The aim of this exercise is for you to gain experience of writing SQL statements that add, delete and modify data in a relational database. The database used in this exercises is the one about pirates described in Section 4.3 on page 24.

1. (a) Append the following tuple to the pirates table.
   - “p5”,“Pegleg”,1,2,“Spain”

   (b) Append the following tuple to the boats table.
   - “b9”,“Swift”,“British”

   Note that no data about the type of boat is to be added to the database.

   (c) Append the following tuple to the booty table.
   - “bt13”,“Flour”,8
2. Change the name of booty “bt8” from “nothing” to “Infected animals”.
3. Remove b7 from the boats table.

6.6 Aggregating Results

The aim of this section is for you to learn more about aggregate functions.

6.6.1 Exercise

The database used in this exercises is the one about pirates described in Section 4.3 on page 24. Write SQL statements for the following.

1. Calculate the total value of all the booty in the booty table.
2. List each type of boat, together with the number of boats of that type.

6.6.2 Aggregate Functions: NULL and DISTINCT

Table 6.1 shows, for each aggregate function, whether NULLs are ignored and whether DISTINCT is meaningful.

Name of SQL Function	Returns	Data type accepted	Are NULLs ignored?	Is DISTINCT meaningful?
AVG()	Average value	Numeric	Yes	Yes
MAX()	Largest value	Any	Yes	No
MIN()	Smallest value	Any	Yes	No
SUM()	Sum of values	Numeric	Yes	Yes
COUNT()	Number of non-null values	Any	Yes	Yes
COUNT(*)	Number of rows	Any	No	Illegal

Table 6.1: Aggregate Functions.

6.6.3 Challenge Questions

1. Suppose we have a column with values 5, 7, 9, and NULL. What value will the SQL function AVG(X) have?
2. If X is a primary key, what is COUNT(X)? How does that value compare to COUNT(*) over the same table?
3. If the result of the query is the same, which is better, eliminating rows with a WHERE clause or eliminating rows with a HAVING clause?

Chapter 7

Laboratory Exercise: More SQL

7.1 Aim

The aim is for you to gain more experience of running SQL queries.

7.2 Naming Result Table Columns

You just have to read this section. Do **not** try to execute the queries in this section using Access. You will execute some queries in the next section.

We can think of the result of the execution of a SQL query as generating a completely new table which usually only exists long enough to output the results. By default, the names of the columns in the table of results are the same as the names of the attributes in the original table. However you can override this default behaviour by specifying an attribute alias using the SQL reserved word AS. Consider the following example. Suppose a database includes the relation shown in Table 7.1. Executing the SQL statement
SELECT p-name AS nickname
FROM pirates;
would result in the output shown in Table 7.2.

p-code	p-name	legs	eyes	origin
p1	Blue Peter	2	2	British
p2	One-Eyed Jack	2	1	American
p3	Black Jack	1	2	Spanish
p4	Blind Pierre	2	0	French

Table 7.1: A relation called pirates.

nickname
Blue Peter
One-Eyed Jack
Black Jack
Blind Pierre

Table 7.2: Output of a SQL query that gives a column an alias.

7.3 More Examples of SQL Queries

Load the NWind.accdb database into Access. (If you cannot recall how to do this then refer to Section 3.2 on page 17.) **Execute** the following list of SQL queries. Each query is first expressed in English (to aid your understanding) and then is expressed in SQL.

1. For each country, display total number of customers from the country, together with the name of the country.
   SELECT Country, COUNT(*) AS “Number of Customers”
   FROM Customers
   GROUP BY Country;

2. Display the Product Id, Product Name, Products Quantity Per Unit and Unit Price of individual products that are not discontinued. Sort the results by ascending alphabetical order on the name of the products.
   SELECT ProductID, ProductName, QuantityPerUnit, UnitPrice
   FROM products
   WHERE Discontinued=No
   ORDER BY ProductName;

7.4 Exercise

The aim of this exercise is for you to gain more experience of running SQL queries which you have written yourself. **Execute** SQL statements which answer the following queries.

1. List the unit price of the cheapest product.
2. What is the total value of all the products?
3. How many products have been discontinued?
4. What is the average unit price of the products which have been discontinued?
5. Display the name of every supplier and the total number of products supplied by that company.
6. Display total number of orders for each employee.

Chapter 8

Tutorial: Data Description Language (DDL)

8.1 Aim

In Chapter 6 you leaned how to add, delete and modify data in a relational database. This chapter gives you a chance to practise these skills on another database. However the principal aim of this chapter is for you to gain experience of writing SQL statements which create, modify and destroy *tables* in a relational database.

8.2 The Three Parts of SQL

Data Manipulation Language (DML) Used to retrieve and modify data. The SQL commands SELECT, INSERT, UPDATE and DELETE belong to this part of SQL.

Data Description Language (DDL) Used to define the structure of the data. You will use this part of SQL in this chapter and the laboratory exercise in Chapter 9.

Data Control Language (DCL) Used to restrict access to data by certain users which is useful for database security.

8.3 Exercise on Creating, Modifying and Destroying Tables

1. Write SQL queries that create tables for the following relations.
   - doctors(doctor-id, doctor-name, telephone-number)
   - treats(treat-id, doctor-id, patient-id, drug, treatment-date)

- patients(patient-id, title, first-name, surname, address)

The attribute doctor-id in the treats relation is a foreign key to doctor-id in the doctors relation. Likewise, the attribute patient-id in the treats relation is a foreign key to patient-id in the patients relation.

Your queries should impose the following constraints.

- Every telephone number is unique.
- Every patient must have both a first and surname.

2. (a) Append the following tuples to the doctors table.
   - “D001”,“Dr Smith”,“0161–456–1234”
   - “D002”,“Dr Miller”,“0161–555–4321”

   (b) Append the following tuples to the patients table.
   - “P001”,“Mr”,“Joe”,“Bloggs”,“Bury”
   - “P002”,“Mrs”,“Mary”,“Green”

   Note that no data about the address of Mrs Green is to be added to the database.

   (c) Append the following tuple to the treats table.
   - “T001”,“D001”,“P001”,2008–09–08

   Note that no drug was given.

3. Write a statement in SQL which modifies the patients table to include a telephone number.
4. Change the address of patient “P001” from “Bury” to “Rochdale”.
5. Remove Dr Miller from the doctors table.
6. For each drug, display the number of times it was given.
7. Display those drugs that were given more than once.
8. Remove all the tables from the database.

Chapter 9

Laboratory Exercise on Data Description Language (DDL)

9.1 Aim

The aim of this laboratory exercise is for you to gain experience of *executing* SQL statements which create a relational database.

9.2 The Exercise

1. You are *not* going to use the NWind database, so if you have already opened it then please close it now.

2. You will need to open a *new* database. Select the relevant icon from the Access menu bar. (You may need to add this icon to the menu bar first.)

3. Recall that to begin writing SQL statements we need to click on “Create” on the top menu and then select “Query Design” from the right hand side of the toolbar.

4. You are going to type in SQL statements. Consequently you do not need the design GUI or the Tables Object. So close windows associated with them. Select SQL View.

9.2.1 Create the Database Tables

Suppose that a company requires a database to store data about its employees and departments, such as that shown in Section 9.2.2. Full details of the company’s requirements are given below. In this exercise, your task is to create the database for the company by executing SQL statements.

The Department Table

In the SQL view window, type in a SQL statement which creates a table that can store tuples about the various departments that employees work in. This table should have the following attributes:

depno the primary key of the department table.

dname the name of each department.

loc the location of each department.

The Employee Table

Open another SQL view window and type in a statement of SQL which creates an employee table that can store tuples about employees. This table should have the following attributes.

empno the primary key of the employee table.

ename the name of the employee.

job the job the employee does.

sal the salary of the employee.

comm the annual commission paid to the employee.

depno the number of the department in which an employee is based. It is a foreign key, linking the employee details stored in the employee table with the details of departments in which employees work, which are stored in the department table.

Note that you cannot create your employee table before your department table as employee.depno references department.depno.

The data types in Access do not conform exactly to the SQL standard. Nevertheless during this exercise you should be able to use the data types CHAR, INTEGER, and NUMERIC as they are described in this booklet. One exception is that you may have difficultly declaring the precision and scale of a NUMERIC data type. If so, you may be able to avoid this problem by using NUMERIC rather than NUMERIC(P,S). (If necessary, study the Access help system which has information on its data types.)

9.2.2 Populate the Database with Data

Populate the tables with data shown in Tables 9.1 and 9.2 using the SQL INSERT command. Note that you cannot populate your employee table before your department table as employee.depno references department.depno. The DBMS won't let you add a value in the employee table for a depno that does not already exist in the department table.

depno	dname	loc
10	Accounting	New York
20	Research	Dallas
30	Sales	Chicago
40	Operations	Boston

Table 9.1: Data for the relation called department.

empno	ename	job	sal	comm	depno
7369	Smith	Clerk	800		20
7499	Allen	Secretary	1600	300	30
7521	Ward	Secretary	1250	500	30
7782	Clark	Manager	2450		10

Table 9.2: Data for the relation called employee.

9.2.3 Retrieve Data from the Database

Execute SQL queries for the following.

1. List the names of all employees.
2. List the names of all secretaries.
3. Calculate the sum of all the salaries of secretaries.
4. Find the average commission, counting only those employees who receive a commission.
5. What is the salary paid to the lowest paid employee?

9.2.4 Make Changes to the Database

1. Add an attribute for telephone number to the department table.
2. Increase the salary of the employee whose number is 7369 to 900.
3. Remove the employee whose number is 7369 from the database.

Chapter 10

Tutorial: Entity Relationship Modelling

10.1 Aim

The aim of this chapter is for you to gain experience of the first step in the process of designing a database. This first step involves developing a conceptual data model. One form of data conceptual model is the Entity-Relationship (ER) model. This contains entities, attributes, relationships and constraints. An Entity-Relationship model can be represented graphically using the Crow's Foot notation.

10.2 Crow's Foot Notation

An entity relationship (ER) diagram is as a graphical representation of the entities and their relationships. There are many different notations for ER diagrams. One of these is the 'Crow's Foot' notation which is denoted in:

- Appendix A.2 of (Connolly & Begg, 2004);
- Appendix C.2 of (Connolly & Begg, 2014);
- (Barker, 1989).

Many organisations still use the Crow's Foot notation, especially for legacy systems which are vital to these organisations. Whenever you are asked to draw an E-R diagram in this chapter, you must use the Crow's Foot notation.

10.3 Exercise

1. Draw an initial E-R diagram for each of the following. Your diagrams need only include entities and relationships.

(a) Projects makes orders.

(b) Suppliers supply parts.

(c) Staff work in departments.

(d) Managers manage projects.

(e) Teams have players and managers manage teams.

2. Given the additional information listed below, add the functionality of the relationships and the membership classes of the entities to your initial E-R diagrams.

(a) A project may make several orders but each order is made by just one project.

(b) A supplier can supply any number of parts and each part may be supplied by many different suppliers.

(c) Each staff member must work in exactly one department but each department will have many staff.

(d) Each manager manages one project and each project has one manager.

(e) Each team has one manager and many players. Each player is on one team but a manager can manage more than one team.

10.4 Challenge Question

Develop a complete E-R model for the organisation described below.

An organisation uses a number of items of equipment to produce goods. Each item is at one *location*, of one *type* and has a *description*.

Each fault on an item is identified by a unique *fault-id*. A record is kept of both the time at which a fault is reported (*time-reported*) and the time at which the repair of the fault is complete (*time-fixed*). Each fault involves just one item.

One or more persons are assigned to a fault and work on it until it is fixed. Each person records how much time they spent fixing the fault. Each person has a *person-id*, a surname, a first-name and any number of qualifications.

A part may be used to fix more than one fault. Any number of different parts may be used to fix a fault. For each part, the quantity used to fix each fault is recored as *qty-used*.

Each part is identified by a *part-id* and has a *weight* and a *cost*. Each part can have any number of colours.

Chapter 11

Tutorial: Automatic Teller Machines

1. The police in Salford use a database system to store data on automatic teller machines (ATMs). The database comprises three relations: bank, branch and ATM. A sample of the data in the database is shown in Tables 11.1, 11.2 and 11.3.

   (a) Write down SQL statements for the following:

      i. Generate a list of the ages of ATMs whose security level is normal. The list should be in descending order.

      ii. List the name of each bank which has a branch whose address starts with the word Eccles. The list should not include any duplicates.

      iii. Calculate the average age of the ATMs with low or normal levels of security. Name the results column “Average age”.

      iv. Remove the tuple from the relation ATM whose identifier is atm04.

      v. Add a row for an ATM, whose atmID is atm05, whose security level is high and which is located at branch T1.

      vi. Add an attribute called telephone_number to the relation Branch.

      vii. Set the address of every branch to the value Salford.

   (b) Write down a SQL statement to join the relations shown in the table as follows.

   Bank LEFT OUTER JOIN Branch

   (c) Show the results of applying the join in part (b).

bankID	name
b1	Barclays
b2	NatWest
b3	Co-op
b4	Yorkshire

Table 11.1: A relation called bank.

branchID	address	sort-code	**bankID***
T1	Eccles Old Rd	13-19-72	b1
T2	Ordsall Lane	54-37-50	b2
T3	Eccles New Rd	34-77-56	b3
T4	Frederick Rd	34-77-99	b3

Table 11.2: A relation called branch.

atmID	security-level	age	**branchID***
atm01	normal	13	T4
atm02	normal	9	T3
atm03	high	3	T3
atm04	low	25	T2

Table 11.3: A relation called atm.

Chapter 12

Tutorial: Transforming an E-R Model to a Logical (Relational) Model

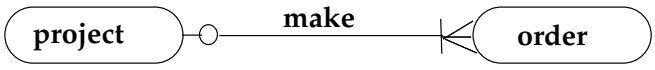
12.1 Aim

Chapter 10 concerned the first step in the process of designing a database, i.e., conceptual modelling. The form of conceptual data model used was the Entity-Relationship model. The aim of this chapter is for you to gain experience of the second step, i.e., developing a Logical model. A Logical model models the real world situation using structures appropriate to the type of database. For a relational database, these are relations (tables), foreign keys, attribute constraints, table constraints etc.

12.2 Exercise

Transform the following E-R models into relational models.

1. **E-R diagram**



Entities

project(proj#, name, start-date, end-date)
order(order#, date, quantity)

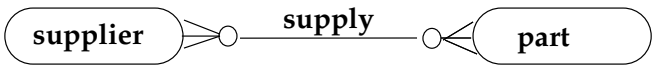
Relationships

make [1:M][o:m]

Constraints/Assumptions

The end date of a project must be after its start date.

2. **E-R diagram**



Entities

supplier(supplier#, name, tel-no)
part(part#, name, price)

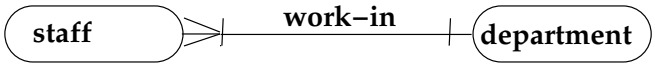
Relationships

supply [M:M][o:o]

Constraints/Assumptions

none

3. **E-R diagram**



Entities

staff(staff-id, name, address, phone-no)
department(dept-id, name, location)

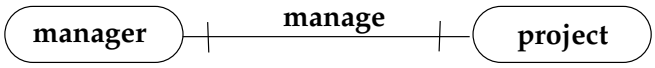
Relationships

work-in [M:1][m:m]

Constraints/Assumptions

none

4. **E-R diagram**



Entities

manager(man-id, name, address, tel-no)
project(proj-id, name, start-date, end-date)

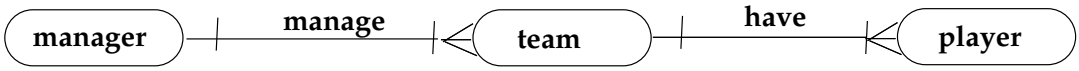
Relationships

manage [1:1][m:m]

Constraints/Assumptions

The end date of a project must be after its start date.

5. **E-R diagram**



Entities

manager(man-id, name, address, tel-no)
team(team-name, date-founded, address)
player(player-id, name, address, tel-no)

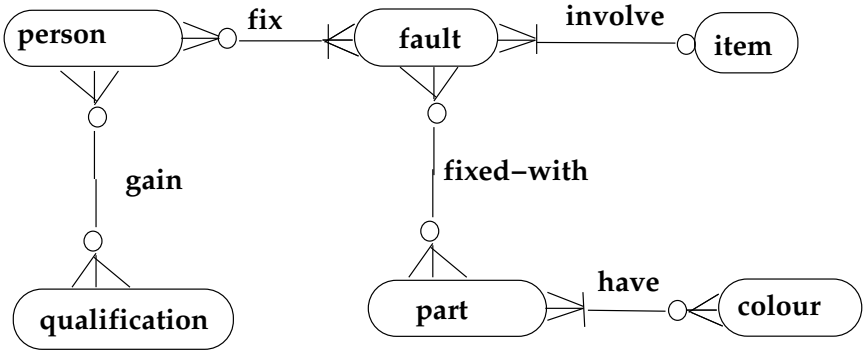
Relationships

manage [1:M][m:m]
have [1:M][m:m]

12.3 Challenge Question

Transform the following E-R model into a relational model.

E-R diagram



Entities

item(item#, location, type, description)
fault(fault-id, time-reported, time-fixed)
person(person-id, surname, first-name)
qualification(qual-code, qual-description)
part(part-id, weight, max-dimension)
colour(col-code, col-description)

Relationships

gain(person-id, qual-code, date)
have [M:M][m:o]
involve [M:1][m:o]
fix(person-id, fault-id, time-spent)
fixed-with(fault-id, part-id, qty-used)

Constraints/Assumptions

None

Chapter 13

Tutorial: Case Study

13.1 Aim

Design a database system which satisfies the requirements listed in Section 13.3 and which, if implemented, would be suitable for storing the data shown in Tables 13.1 and 13.2.

1. Draw an E-R diagram drawn using the **Crow's foot notation**.
2. Transform your E-R model to a logical model.
3. Check that your logical model is normalised.
4. Write down your final logical model.
5. Explain why your logical model is in normal form.

You are not expected to complete all of the steps in one hour. It may take some of you a long time to draw the E-R diagram because you have very limited experience of drawing them. You should complete the remaining steps of the design process in your own time. Carrying out these steps on the same problem will help you to learn how the different steps are related to each other.

13.2 Background

Trade unions are organisations that represent people at work. Their purpose is to protect and improve people's pay and conditions of employment. They also campaign for laws and policies which will benefit working people. Trade unions exist because an individual worker has very little power to influence decisions that are made about his or her job. By joining together with other workers, there is more chance of having a voice and influence.

The University and College Union (UCU) is the largest trade union for academics, lecturers, trainers, researchers and academic-related staff working in further and higher education throughout the UK.

If you are curious to learn more about trade unions then you may wish to visit the People's History Museum which is not far from the university. It is a two minute walk from Salford Central railway station. Admission is free to all. For more information, see: http://www.phm.org.uk/.

13.3 The Database

UCU requires a database system to record details of its members.

For the purposes of membership administration, UCU is divided into geographic regions. Each region is assigned a unique code and has a name. Each region has an office located in a city in the region. Each region includes a number of branches of the union.

When people join UCU they are assigned a unique membership number called `mem_no`. Data on the name and age (in years) of members are stored in the database, together with the type of membership for each member.

Most members belong to a branch of UCU based at the educational institution where they work. A member cannot belong to more than one branch. Branches are uniquely identified by a unique number called `branch_no`. Each branch is based at just one educational institution and lies within one region only.

Some members do not belong to a branch of the union because they work at the union's HQ in London or at a regional office.

Some members play a particular role in the union such as president. Each role is performed at either national, regional or branch level. The names of roles are not unique. E.g., UCU has a National President but some branches have a Branch President.

Branch		Regional Office		
Number	Institution	Code	Name	Location
nw113	University of Salford	nw	north west	Manchester
nw081	University of Manchester	nw	north west	Manchester
nw021	Manchester College	nw	north west	Manchester
ne187	University of York	ne	north east	Leeds

Table 13.1: Data on branches and regional offices.

Member				Branch	Role		
Number	Name	Age	Type	Number	RoleID	Level	Role
m01	tom	24	full-time	nw081			
m02	dick	45	part-time	nw081			
m03	harry	27	full-time	nw021			
m04	sue	26	full-time	nw021			
m05	kate	66	retired	ne187			
m06	jo	43	full-time		r01	national	general secretary
m07	parveen	59	full-time		r02	national	president
m08	martin	44	full-time		r03	regional	region officer
m09	helen	35	full-time	nw113	r04	branch	president
m10	umran	50	part-time	nw113	r05	branch	vice-president

Table 13.2: Data on members.

Chapter 14

Tutorial: Normalisation

14.1 Aim

The aim of this chapter is for you to gain experience of identifying faults in the design of tables and restructuring your tables to remove the faults. In other words, you are going to learn how ensure that your tables are normalised.

14.2 Exercise: Identifying Design Faults

1. Is Table 14.1 in 1NF? Is it in 2NF? Is it in 3NF? Justify your answers.

StudentID	Module	CW Mark	Exam Mark
u010123	CRN003	75	65
u010999	CRN003	65	50
u010123	CRN007	65	40
u010999	CRN007	44	65

Table 14.1: Data on exam and coursework marks.

2. Is Table 14.2 in 1NF? Is it in 2NF? Is it in 3NF? Justify your answers.

StudentID	Module	CW Mark	CW Outcome
u010123	CRN003	75	Pass
u010999	CRN003	65	Pass
u010123	CRN007	65	Pass
u010999	CRN007	44	Fail

Table 14.2: Data on coursework marks with pass and fail outcomes.

14.3 Exercise: Restructuring Tables

1. Convert Table 14.3 to 1^{st} normal form.

country	saint	cities
England	George	Manchester, Leeds, Newcastle
Wales	David	Swansea, Cardiff
Scotland	Andrew	Glasgow, Edinburgh

Table 14.3: Data on patron saints of countries.

2. Convert Table 14.4 to 2^{nd} normal form.

module	student-ID	stud-name	grade
Databases	S001	Tom	A
Databases	S002	Dick	B
Databases	S003	Harry	B
Research Methods	S001	Tom	A
Research Methods	S003	Harry	B
Java	S001	Tom	C
Java	S002	Dick	C
Java	S003	Harry	C

Table 14.4: Data on modules, students and grades.

3. Examine Table 14.5 to see if it contains redundant data, i.e., the same fact stored more than once. Suggest how the redundancy might be removed by splitting the table into two relations with a primary key/foreign key link.

project-manager-ID	project-manager-name	part	quantity
p1	Jill	hammer	7
p2	Henry	hammer	11
p2	Henry	saw	20
p1	Jill	drill	2
p1	Jill	hammer	11

Table 14.5: Data on projects and parts.

Chapter 15

Tutorial: SQL Subqueries

15.1 Aim

We can embed a SELECT statement within another SELECT statement. As one is inside the other, we distinguish between the two by referring to them as:

1. the inner SELECT statement,
2. the outer SELECT statement.

The aim of this chapter is for you to gain experience of devising SQL subqueries to retrieve specific information from a tiny relational database.

15.2 The Exercise

The following questions refer to the Pillaging-Pirates database shown in Section 4.3 on page 24.

Formulate SQL expressions that implement the following queries. *Do not be confused if you encounter a sense of déjà-vu: you should already have written similar queries for this data previously* This time all of your expressions must include subqueries. None of them should involve joins.

1. List the names of pirates who have pillaged booty of type bt3.
2. List the names of pirates who have pillaged any kind of fruit.
3. List the names of booty whose value is greater than the average value of all the booty.

Chapter 16

Laboratory Exercise on SQL Subqueries

16.1 Aim

The aim of this laboratory exercise is for you to gain experience of designing and then *executing* SQL subqueries to retrieve specific information from a relational database.

16.2 The Exercise

- Open Access and load the NWind.accdb database (see Section 3.2 on page 17).
- Type in SQL statements which answer the following queries.
- Do **not** use the design GUI.
- All of your expressions must include **sub**queries.

1. List the names of the products whose number of units on order is greater than the average number of units in stock.
2. List the names of the products which have a value for UnitsInStock that is less than the average number of units in stock. Your query should also list the value for the UnitsInStock for each of these products. The results must be sorted into ascending order on UnitsInStock.
3. List names of the products whose number of units in stock is less than the average number of units in stock and whose number of units on order is less than the average number of units on order.

Chapter 17

Tutorial: Data Modelling

1. (a) Using the Crow's foot notation, draw an Entity-Relationship diagram for the following specification. You diagram must show both the functionalities and membership classes of relationships.

   A database system is required to store data on students' attendance at classes at the University of Salford. The database needs to keep a record of which students attend each module during each week of the semester.

   A student is registered to study on just one programme. Programmes without any students are not stored in the database. A programme is comprised of modules. Every module is taught on one (and only one) programme.

   The database must be ready for the start of each semester. Therefore the database has to be able to store the details of modules and students before any data on students' attendance at classes is entered.

   (b) Transform the conceptual data model shown below into a logical (relational) model.

   (c) Convert Table 17.1 to Second Normal Form.

   (d) Write down a definition of Third Normal Form.

OrderNo	PartNo	Description	Quantity	UnitPrice	Cost
1	P1	Screw	3	6p	18p
1	P2	Nut	4	7p	28p
1	P3	Bolt	1	10p	10p
2	P1	Screw	3	6p	18p
2	P3	Bolt	4	10p	40p

Table 17.1: Data on orders and parts.

Chapter 18

Tutorial: Mobile Phones and Energy Generation

1. (a) A mobile phone company requires you to implement a database with the following two relations:

   customer(cust-number, name, age, nationality)

   handset(handsetID, size, cust-number*)

   where the attribute cust-number is nine characters long, the attribute handsetID is six characters and both of the attributes age and size are integers.

   The database must comply with the following constraints.

   - A customer must have a name.
   - Unless otherwise specified, the nationality of a customer is assumed to be British.

   Write SQL commands which create the two relations. (You are not required to populate the relations with data.)

   (b) The United Kingdom is facing an energy crisis. Suppose that the government has set up a database of fuels and countries that can supply them. A sample of the data in the database is shown in Tables 18.1, 18.2 and 18.3.

   i. Define a SQL subquery which lists the names of countries which supply carbon neutral fuels. Do not use joins.

   ii. Define a SQL join query which also lists names of countries which supply carbon neutral fuels. Do not use subqueries.

   iii. If the amount of data stored in the tables was huge, would it be better to use your answer to question (i) or your answer to question (ii). Briefly justify your answer.

countryID	name	continent
C001	Russia	Europe
C002	Germany	Europe
C003	Morocco	Africa
C004	China	Asia

Table 18.1: A relation called country.

countryID	fuelID	capacity
C001	F100	382
C001	F200	452
C002	F100	134
C003	F400	3
C004	F100	644
C004	F300	980

Table 18.2: A relation called supplies.

fuelID	fuel-name	carbon-neutral
F100	Coal	No
F200	Gas	No
F300	Uranium	Yes
F400	Solar	Yes

Table 18.3: A relation called fuel.

Chapter 19

Tutorial: Relational Algebra

19.1 Aim

Relational databases have a long history by computing standards. The relational data model was proposed in a seminal paper (Codd, 1970). Relational Algebra (Codd, 1971) is a (high-level) procedural language which be used to tell the DBMS how to build a new relation from one or more relations in the database. The aim of this chapter is for you to gain experience of writing expressions of relational algebra. The following questions refer to the Pillaging-Pirates database shown in Section 4.3 on page 24.

19.2 Exercise

Express each of the following in both (a) relational algebra and (b) SQL.

1. List the names and types of all boats.
2. List the names of all pirates.
3. List the names of all pirates who have less than two eyes.
4. List the names of all pirates who have less than two eyes or less than two legs or both.
5. List all the places that either pirates or boats or both originate from.
6. List all the places where boats but not pirates originate from.

Chapter 20

Laboratory Exercises on Relational Algebra

20.1 Aim

Relational Algebra is a (high-level) procedural language. It can be used to tell the DBMS how to build a new relation from one or more relations in the database. The aim of this laboratory exercise is for you to gain experience of *executing* set operations in SQL queries.

20.2 Example Used in the Lecture

Suppose the staff of a company comprises the employees and the managers shown in Tables 20.1 and 20.2.

- Implement the two tables using the SQL CREATE TABLE command.
- Write SQL statements that populate the two tables with the data shown.
- Write a single SQL query which returns all the staff from both of the tables.

empNo	eName	sal
7782	clark	2450
7839	king	5000

Table 20.1: A relation called employee

empNo	eName	sal
8034	smith	7000
8044	yao	6000

Table 20.2: A relation called manager.

20.3 NWind Exercise

Load the NWind.accdb database into Access (see Section 3.2 on page 17). For each of the following, type in a single SQL statement.

1. List all company names and cities from both the Suppliers and Customers tables.
2. $\Pi_{\text{CompanyName, City}}(\text{Suppliers})$
3. $\Pi_{\text{CompanyName, City}}(\sigma_{\text{Country=uk}}(\text{Suppliers}))$
4. $\Pi_{\text{Country}}(\text{Suppliers}) \cup \Pi_{\text{ShipCountry}}(\text{Orders})$

Chapter 21

Tutorial: Enhanced Entity Relationship (EER) Modelling

21.1 Aim

Chapter 10 concerned basic Entity-Relationship (E-R) models. The aim of this chapter is for you to gain experience of developing more advanced E-R models.

21.2 Additional Features of ER Modelling

21.2.1 Exercise

Draw an initial E-R diagram for each of the following. Your diagrams need only include entities and relationships. Omit the functionalities and membership classes.

1. A member of staff matriculates a student at a university;
2. A surgeon and an anaesthetist operate on a patient in an operating theatre;
3. A member of staff is supervised by another member of staff.

21.3 Enhanced Entity Relationship (EER) modelling

21.3.1 Exercise

1. Develop a complete E-R model for the banking system described below. Make what you see as reasonable choices for membership classes, functionalities and the attributes. List any constraints and assumptions that must be applied to the system.

The database will store data on the balance of each account and the date it was opened. It will also store the name, address and telephone number of each customer. One or more customers may jointly own any account. Each customer has one current account which has an overdraft limit which cannot be exceeded. A customer may have zero, one or more deposit accounts. The balance on a deposit account is not allowed to fall below zero. Each deposit account has a maximum amount which can be withdrawn in any one transaction. Records are kept of the deposits and withdrawals made by customers, including the date, time and amount of money.

2. Transform your E-R model into relational model.

Chapter 22

Tutorial: Team Management

1. Relational algebra is part of the mathematical basis for relational databases.
   (a) The relational operators are classed as either unary or binary. Explain the distinction between unary and binary operators.
   (b) Write down the name of the FIVE fundamental operations of relational algebra and give a brief description of each one.
   (c) Consider the Staff relation shown in Table 22.1. Express each of the following in relational algebra.
      i. SELECT *
      FROM staff
      WHERE salary $>$ 20000;
      ii. SELECT DISTINCT name, salary
      FROM staff;
      iii. SELECT DISTINCT name
      FROM staff
      WHERE salary $<$ 10000;
   (d) Consider the teamManager relation shown in Table 22.2. With reference to both the teamManager relation and the staff relation (see Table 22.1), express the query "List the staff numbers of all staff who are not managers." as follows:
      i. Write a single statement of relational algebra.
      ii. Write a single statement of SQL.

staffID	name	DOB	salary	branchNo
S1	Tom	1970-12-03	30000	B005
S2	Sarah	1980-12-01	12000	B003
S3	Harry	1975-02-09	18000	B003
S4	Sophie	1981-05-04	9000	B007
S5	Louise	1974-07-04	24000	B003
S6	Laura	1979-11-07	9000	B005

Table 22.1: A relation called staff.

staffID	team
S2	personnel
S4	payroll
S5	accounts
S6	estates

Table 22.2: A relation called teamManager.

Part III

Semester Two

Chapter 23

Tutorial: Query Optimisation

23.1 Aim

When the relational model was first launched commercially, one of the major criticisms was that the performance of the queries was inadequate. Since then, a significant amount of research has been devoted to developing highly efficient algorithms for processing queries. The aim of this tutorial is for you to gain some insight into how a DBMS executes queries efficiently by drawing query trees and optimising them using heuristics.

23.2 Exercise

1. What is a query tree? Explain what the different types of nodes represent.
2. (a) Write a relational algebra expression for the following SQL query.

```
SELECT lecName, schedule
FROM lecturer, module, enroll, student
WHERE lastName="Burns"
AND firstName="Edward"
AND module.moduleNumber=enrol.moduleNumber
AND lecturer.lecID=module.lectID
AND student.stuID=enrol.stuID;
```

   (b) Draw a query tree for this SQL query.
3. Why do we use relation algebra for query optimisation? Why can we not just use SQL?
4. List the heuristics (rules of thumb) that optimisers apply to reduce the cost of optimisation.
5. (a) Draw a **near optimal** query tree for the following SQL query.


```
SELECT sailors.name
FROM sailors, reservations
WHERE reservations.sID=sailors.ID
AND reservations.bID=100
AND sailors.rating=7;
```

(b) Write a relational algebra expressions for this tree.

6. (a) Draw a **near optimal** query tree for the following SQL query.

```
SELECT employee.name, employee.address
FROM employee, department
WHERE department.name="Research"
AND department.deptID=employee.deptID;
```

(b) Write a relational algebra expressions for this tree.

Chapter 24

Laboratory Exercises on Theta Joins

24.1 Aim

The aim of this laboratory exercise is for you to gain experience of *executing* SQL queries which retrieve the same data from a database as some theta joins do.

24.2 Definition of Theta Join

$R \bowtie_F S$ defines a relation that contains tuples satisfying F from the Cartesian product of two relations R and S. In other words, it takes the Cartesian product of two relations R and S and then selects the rows which satisfy the condition F. Expressing this mathematically: $R \bowtie_F S = \sigma_F(R \times S)$

24.3 NWind Exercise

Open Access and load the NWind.accdb database. (If you cannot recall how to do this then refer to Section 3.2 on page 17.)

1. Write SQL queries which count the number of tuples in each of the following relations.
   (a) Products
   (b) Suppliers
2. How many suppliers does each product have? Hint: click on Database Tools and select Relationships.
3. Write SQL queries which count the number of tuples in each of the following algebraic statements. Provide an explanation for the counts.

(a) The relation created by Products $\times$ Suppliers.

(b) The relation created by
Products $\bowtie_{\text{Products.SupplierID=Suppliers.SupplierID}}$ Suppliers.

4. For each of the following, write a description in English of the data it will retrieve and execute a single SQL statement which retrieves this data from the database.

(a) $\Pi_{\text{ProductName}}($
$\sigma_{\text{City=Manchester}}(\sigma_{\text{Products.SupplierID=Suppliers.SupplierID}}($
Products $\times$ Suppliers
$)))$

(b) $\Pi_{\text{ProductName}}(\sigma_{\text{City=Manchester}}($
Products $\bowtie_{\text{Products.SupplierID=Suppliers.SupplierID}}$ Suppliers
$))$

(c) $\Pi_{\text{Employees.City}}(\sigma_{\text{Employees.city=Customers.City}}($
Employees $\bowtie_{\text{Employees.EmployeeID=Orders.EmployeeID}}$ (
Orders $\bowtie_{\text{Orders.CustomerID=Customers.CustomerID}}$ Customers
$)))$

Chapter 25

Tutorial: Transactions and Concurrency

25.1 Aim

Online Transaction Processing Systems need real-time support for querying and updating of databases by one or more concurrent users. The aim of this tutorial is help you understand why this is so.

25.2 Exercise

1. A major objective in developing a database is to enable many users to access shared data concurrently. Explain why concurrent access is relatively easy if all users are only reading data. Illustrate your answer using the example of an on-line railway timetable.

2. Explain the following statement. “A transaction is a logical unit of work.”

3. The DBMS does not guarantee that the semantic meaning of the transaction truly represents the real-world event. Who does have responsibility for the semantic meaning? What are the possible consequences of this limitation? Illustrate your answer using a transaction that is intended to transfer money from one bank account to another.

4. Name the four properties of database transactions which make up the acronym ACID. Describe each of the ACID properties.

5. Explain why the DBMS alone cannot be expected to uphold the responsibility for the consistency property. Illustrate your answer using a transaction that is intended to transfer money from one bank account to another.

6. Why do database systems support concurrent execution of transactions, in spite of the extra programming effort needed to ensure that concurrent execution does not cause any problems?

7. Justify the following statement. “Executing transactions concurrently is more important when data must be fetched from secondary storage or when transactions are long, and is less important when data is in primary storage and transactions are very short.”

Chapter 26

Tutorial: Concurrency Control

26.1 Aim

The aim of this tutorial is for you to learn why a lack of concurrency control causes problems for real-world applications of databases.

26.2 Exercise

1. (a) What is meant by the term Concurrency Control?
   (b) Why is Concurrency Control needed?
2. (a) A major objective in developing a database is to enable many users to access shared data concurrently. Explain why concurrent access needs to be carefully managed when two or more users are accessing the database simultaneously and at least one is updating data.
   (b) Illustrate your answer using the example of an on-line railway seat reservation system. Assume that the railway seat reservation system must ensure that a particular seat on a particular train journey is not reserved by more than one passenger.
3. Which component of the DBMS takes responsibility for the isolation property of database transactions?
4. Name the three most common concurrent transaction execution problems.
5. Under what circumstances do two operations in a schedule conflict?
6. Identify all of the pairs of operations which conflict in each of the schedules shown in Table 26.1.

Schedule (a)

Time	T1	T2	T3
21:02	Read(x)		
21:03	Write(x)		
21:05		Read(x)	
21:06		Write(x)	
21:07			Read(x)
21:08	Commit		
21:12		Commit	
21:13			Commit

Schedule (b)

Time	T1	T2
03:09		Read(y)
03:11	Write(y)	
03:12		Write(y)
03:17	Read(y)	
03:18	Commit	Commit

Schedule (c)

Time	T1	T2
07:15	Read(x)	
07:19	Write(x)	
07:20		Write(x)
07:25		Read(y)
07:26	Read(y)	
07:27	Write(z)	
07:28	Commit	
07:29		Commit

Schedule (d)

Time	T1	T2
19:53	Read(x)	
19:56		Read(x)
19:57	Write(x)	
19:59		Write(x)
20:00	Write(y)	
20:05	Commit	
20:09		Commit

Table 26.1: Four schedules that contain operations which conflict.

Chapter 27

Laboratory Exercise: Application of Joins

27.1 Aim

The aim of this laboratory exercise is for you to gain more experience of creating, populating and querying a database.

27.2 Scenario

Suppose that you work for a company that imports spare parts for cars. The company have asked you to create a database about cars and parts, populate it with data and use it to help the company find answers to questions about the data.

27.3 Create the Database Relations

1. You are *not* going to use the NWind database, so if you have already opened it then please close it now.

2. You will need to open a *new* database. Select the relevant icon from the Access menu bar. (You may need to add this icon to the menu bar first.)

3. Recall that to begin writing SQL statements we need to click on “Create” on the top menu and then select “Query Design” from the right hand side of the toolbar.

4. You are going to type in SQL statements. Consequently you do not need the design GUI or the Tables Object. So close windows associated with them. Select SQL View.

5. Your first task is to create the following relations by executing SQL CREATE TABLE statements.
   car(carId, carMake, carModel, carPrice)
   part(partId, partName, partPrice)
   carPart(carId*, partId*)
   When selecting data types, bear in mind that these relations will be used to store the tuples listed in Tables 27.1, 27.2 and 27.3. After you have created the relations, check that they have the referential integrity constraints by clicking on Database Tools and selecting Relationships.

carId	carMake	carModel	carPrice
1	Ford	Focus	18000
2	Toyota	Avensis	20000
3	BMW	640d	60000
4	Porsche	Boxster 718	42000

Table 27.1: A sample of the data for the relation called car.

27.4 Populate the Database with Data

Populate the relations with data shown in Tables 27.1, 27.2 and 27.3 using the SQL INSERT command. Note that you cannot populate your carPart relation before your car relation because carPart.carID references car.carId. Note that you cannot populate your carPart relation before your part relation because carPart.partid references part.partId.

partId	partName	partPrice
1	part1	54
2	part2	664
3	part3	224
4	part4	11
5	part5	187
6	part6	22
7	part7	998
8	part8	115
9	part9	23
10	part10	55

Table 27.2: A sample of the data for the relation called part.

27.5 Queries Involving Joins

Write a single SQL query for each of the following.

1. List each partId and partName, together with the corresponding values of carMake, carModel and carPrice. Order the list by partId.
2. List partIds, together with the average price of the cars that can use the part.
3. Find the model names of the cars that use the most expensive part. Your query should return the Ford Focus and Porsche Boxster 718, both of which use the most expensive part, i.e., part7 at £998.

carId	partId
1	3
1	8
1	5
1	1
1	7
2	3
2	4
2	2
2	8
3	9
3	5
3	3
3	1
4	7
4	9
4	5
4	6
4	4
4	2

Table 27.3: A sample of the data for the relation called carPart.

Chapter 28

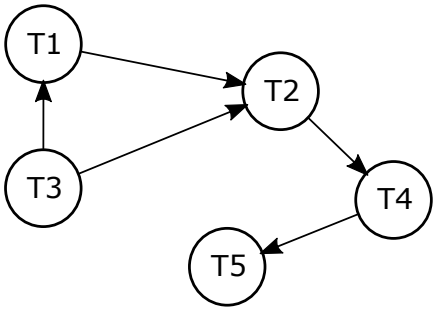
Tutorial: Precedence Graphs

28.1 Aim

The aim of this tutorial is for you to gain experience of determining whether a non-serial schedule will allow transactions to execute concurrently without interfering with one another, and thereby produce a database state that could be produced by a serial execution.

28.2 Exercise

1. (a) What is a serial schedule?
   (b) Why is a serial schedule considered correct?
   (c) What is a serializable schedule?
   (d) Why is a serializable schedule considered correct?
2. Consider the following precedence graph for a non-serial schedule consisting of transactions. Is the corresponding schedule conflict serializable? Justify your answer.



3. For each of the schedules shown in Table 28.1, draw a precedence graph, determine whether it is conflict serializable and justify your answer.

Schedule (a)

T1	T2	T3
Read(x)		
Write(x)		
	Read(x)	
	Write(x)	
		Read(x)
Commit		
	Commit	
		Commit

Schedule (b)

T1	T2
	Read(y)
Write(y)	
	Write(y)
Read(y)	
Commit	Commit

Schedule (c)

T1	T2
Read(x)	
Write(x)	
	Write(x)
	Read(y)
Read(y)	
Write(z)	
Commit	
	Commit

Schedule (d)

T1	T2
Read(x)	
	Read(x)
Write(x)	
	Write(x)
Write(y)	
Commit	
	Commit

Table 28.1: Four schedules.

Chapter 29

Tutorial: Two-Phase Locking

29.1 Aim

Recall that the objective of concurrency control is to find non-serial schedules that allow transactions to execute concurrently without interfering with one another, and thereby produce a database state that could be produced by a serial execution. The aim of this tutorial is for you to understand to what extent the two-phase locking protocol achieves this objective.

29.2 Exercise

1. (a) What are locks used for?
   (b) Briefly describe how locks are implemented.
   (c) What is an exclusive lock?
   (d) Under what circumstances is an exclusive lock granted?
2. State whether each of the following is true or false. If it is false then briefly explain why.
   (a) A transaction in which all steps must be completed successfully or none of them will be completed is called a durable transaction.
   (b) Although two transactions may be correct in themselves, interleaving of operations may produce an incorrect result.
   (c) Two-phase locking guarantees that transactions are serializable.
   (d) Two-phase locking prevents deadlock.
3. Suppose the 2PL protocol is to be applied to the schedule listed in Table 29.1 that concerns the balance of a building society account.

(a) Following the same interleaving as far as possible, write down the resulting revised schedule. Your revised schedule must explicitly include details of the locking.

(b) State whether the 2PL protocol solves the concurrency problem.

Time	Transaction 1	Transaction 2	balance
t1	begin transaction		100
t2	read(balance)		100
t3	balance:=balance-60		100
t4	write(balance)	begin transaction	40
t5	⋮	read(balance)	40
t6	abort and rollback	balance:=balance-10	100
t7		write(balance)	30
t8		commit	30

Table 29.1: Schedule involving two transactions. The final column shows the balance of an account as recorded on secondary storage.

Chapter 30

Laboratory Exercise: Groups and Joins

30.1 Aim

The aim of this laboratory exercise is for you to gain further experience of devising SQL queries involving joins and grouping.

30.2 NWind Exercise

- Open Access and load the NWind.accdb database. (If you cannot recall how to do this then refer to Section 3.2 on page 17.)
- Click on Database Tools and select Relationships and study the structure of the database.
- Recall that to begin writing SQL statements we need to click on “Create” on the top menu and then select “Query Design” from the right hand side of the toolbar.
- You are going to type in SQL statements. Consequently you do not need the design GUI or the Tables Object. So close windows associated with them. Select SQL View.
- Write and execute SQL queries which do the following.

1. List the last name of each employee, together the number of customers that have placed orders with this employee. Your query should produce the results shown in Table 30.1(a).
2. List the last name of each employee, together with the number of customers in Germany that have placed orders with this employee. Your query should produce the results shown in Table 30.1(b).

3. List the contact name of each customer who places orders with more than twenty employees, together the number of orders which the customer placed. Your query should produce the results shown in Table 30.2(a).

4. List the contact name of each customer who places orders with more than four employees in the UK, together the number of orders which the customer placed. Your query should produce the results shown in Table 30.2(b).

Last Name	NumberOfCustomers
Buchanan	42
Callahan	104
Davolio	123
Dodsworth	43
Fuller	96
King	72
Leverling	127
Peacock	156
Suyama	67

(a) Customers in all countries.

Last Name	NumberOfCustomers
Buchanan	4
Callahan	17
Davolio	19
Dodsworth	9
Fuller	14
King	6
Leverling	19
Peacock	25
Suyama	9

(b) Only customers in Germany.

Table 30.1: List of employees and the number of customers that placed orders with this employee.

Contact Name	NumberOfOrders
Horst Kloss	28
Jose Pavarotti	31
Roland Mendel	30

(a) With more than 20 employees.

Contact Name	NumberOfOrders
Annette Roulet	6
Carlos Hernandez	5
Jose Pavarotti	11
Jose Pedro Freyre	5
Lucia Carvalho	7
Maria Larsson	5
Pascale Cartrain	5
Patricia McKenna	8
Roland Mendel	9

(b) With more than 4 UK employees.

Table 30.2: List of customers who placed orders.

Chapter 31

Tutorial: Wait For Graphs

31.1 Aim

Basic two-phase locking does not prevent deadlock. The aim of this tutorial is for you to gain experience of determining when deadlock may occur.

31.2 Exercise

1. (a) When does deadlock occur?
   (b) How does the DBMS usually detect that deadlock has occurred?
   (c) What is the only way to break deadlock once it has occurred?
2. Draw the wait-for graph (WFG) graph for the schedule shown in Table 31.1 and determine whether there is potential for deadlock. Explain your answer.

Time	T1	T2	T3
1		Begin transaction	
2		Read(Z)	
3	Begin transaction	Read(X)	
4	Write lock(X)	Write lock(Z)	
5	X=25*X	Z=X*Y	
6	Write(X)	Write(Z)	Begin transaction
7	⋮	⋮	Read(Z)
8			Write lock(Z)
9			WAIT
10			WAIT

Table 31.1: A schedule involving three concurrent transactions.

3. Table 31.2 lists the locking information for some transactions. Draw a WFG. Is there potential for deadlock? Explain your answer.

Transaction	Data items locked by transaction	Data items transaction is waiting to lock
T1	C	A
T2	A	
T3	F	D,E
T4	D	B,C
T5	B,E	C

Table 31.2: Locking information on five transactions involving six data items.

4. Table 31.3 lists the locking information for some transactions. Draw a WFG. Is there potential for deadlock? Explain your answer.

Transaction	Data items locked by transaction	Data items transaction is waiting to lock
T1	E	C
T2	B	E
T3	A	B
T4	D	A
T5	C	B,D

Table 31.3: Locking information on five transactions involving five data items.

Chapter 32

Tutorial: Recovery

32.1 Aim

Database recovery is the process of restoring the database to a correct state after a failure. The aim of this tutorial is develop your understanding of two protocols for recovery.

32.2 Exercise

1. State whether each of the following is true or false. If it is false then briefly explain why.
   (a) The REDO operation updates the database with new values (after-image) that are stored in the log.
   (b) In case of the deferred update protocol, updates are not written to the database until after a transaction has reached its COMMIT point.
   (c) In case of the immediate update protocol, all updates to the database are applied immediately as they occur and a record of all changes is kept in the transaction log.
   (d) The deferred update protocol is also known as the UNDO/REDO method.
   (e) Shadow paging is a technique where transaction logs are not required.
2. The DBMS maintains a log file containing transaction records that identify the start and end of transactions and the before and after images of the write operations. Consider the log files shown in Tables 32.1 and 32.2. For each one, explain what the DBMS recovery system would do, and why, if there was a failure at the time shown below each table and the DBMS was using:
   (a) the deferred update protocol;
   (b) immediate update protocol.

Transaction ID	Time	Operation	Object	Before	After
T1	10:10	Start			
T1	10:11	Update	X	200	350
T2	10:13	Start			
T2	10:14	Insert	Y		150
T2	10:16	Delete	Z	15	
T3	10:18	Start			
T1	10:19	Update	A	23	200
T1	10:21	Commit			
T2	10:23	Insert	C		234
T3	10:25	Delete	G	653	
T2	10:27	Commit			
T4	10:29	Start			
T3	10:31	Commit			
T4	10:35	Update	C	234	500

Table 32.1: Log file for Monday. Assume that there is a failure at 10:24.

Transaction ID	Time	Operation	Object	Before	After
T1	9:10	Start			
T2	9:01	Start			
T1	9:02	Read	X		
T1	9:03	Read	Y		
T2	9:04	Read	X		
T3	9:05	Start			
T2	9:06	Read	Z		
T3	9:07	Write	C	55	70
T1	9:08	Write	J	89	16
T1	9:11	Commit			
T4	9:12	Start			
T2	9:13	Read	R		
	9:15	Checkpoint			
T2	9:16	Write	L	34	44
T4	9:19	Write	P	55	3
T3	9:20	Commit			

Table 32.2: Log file for Tuesday. Assume that there is a failure at 9:21.

3. For each of the log files shown in Tables 32.3 and 32.4, categorise each transaction as safe, needs undoing or needs redoing. Assume that:
   - there was a failure at 10:36 on both Wednesday and Thursday;
   - the DBMS was using immediate update protocol.

Transaction ID	Time	Operation	Object	Before	After
T1	10:10	Start			
T1	10:11	Write	X	200	350
T2	10:13	Start			
T2	10:14	Read	Y		150
T2	10:16	Delete	Z	15	
T3	10:18	Start			
T1	10:19	Write	A	23	200
T1	10:21	Commit			
T2	10:23	Insert	C		234
T3	10:24	Delete	G	653	
T2	10:27	Commit			
T4	10:29	Start			
T3	10:31	Commit			
T4	10:35	Update	C	234	500

Table 32.3: Log file for Wednesday.

Transaction ID	Time	Operation	Object	Before	After
T1	10:10	Start			
T1	10:11	Write	X	200	350
T2	10:13	Start			
T2	10:14	Read	Y		150
T2	10:16	Delete	Z	15	
T3	10:18	Start			
T1	10:19	Write	A	23	200
T1	10:21	Commit			
T2	10:22	Insert	C		234
	10:23	Checkpoint			
T3	10:24	Delete	G	653	
T2	10:27	Commit			
T4	10:29	Start			
T3	10:31	Commit			
T4	10:35	Update	C	234	500

Table 32.4: Log file for Thursday.

Chapter 33

Laboratory Exercise: Consolidation

33.1 Aim

The aim of this laboratory exercise is for you to gain further experience of devising SQL queries involving pattern matching, sorting, aggregate functions, joins and set operators.

33.2 NWind Exercise

- Open Access and load the NWind.accdb database.
- Click on Database Tools and select Relationships and study the structure of the database.
- Write and execute SQL queries which do the following.

1. List the company names of customers that have a contact name which includes the string “liz”. Your query should return the results shown in Table 33.1(a).

Company Name
Bottom-Dollar Markets
Consolidated Holdings
The Big Cheese

(a) Includes “liz”.

Product Name
Alice Mutton
Chef Anton’s Gumbo Mix
Northwoods Cranberry Sauce
Perth Pasties
Thringer Rostbratwurst

(b) Less than ten.

Country
Singapore
Spain
Sweden
Switzerland

(c)

Table 33.1: Results of executing SQL queries.

2. List the names of the products (sorted in alphabetic order) for which less than ten are in stock and less than ten are on order. Your query should return the results shown in Table 33.1(b).

3. What is the total cost of all the unit prices of the products supplied by companies based in the UK? Your query should return 159.70.

4. List all the countries that start with the letter S in which a supplier or a customer or both are based. Your query should return the results shown in Table 33.1(c).

5. List all the countries in which customers, but no employees, are based. Your query should return the results shown in Table 33.2. Unfortunately, MS Access does not support the SQL reserved word EXCEPT, so you will have to devise an alternative solution.

Country
Argentina
Austria
Belgium
Brazil
Canada
Denmark
Finland
France
Germany
Ireland
Italy
Mexico
Norway
Poland
Portugal
Spain
Sweden
Switzerland
Venezuela

Table 33.2: Countries with customers but no employees.

Chapter 34

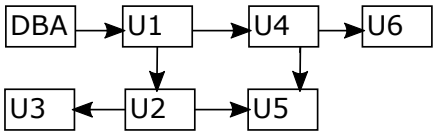
Tutorial: Security

34.1 Aim

The aim of this tutorial is for you to gain experience of writing SQL statements that control which Users have access to which parts of a database system.

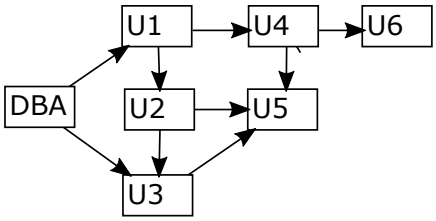
34.2 Exercise

1. Define the following terms in the context of database security.
   (a) authentication;
   (b) authorisation;
   (c) roles within SQL.
2. Explain what the following parts of an authorization grant graph represent.
   (a) The nodes.
   (b) The root node.
   (c) An edge.
3. Consider the following grant authorisation graph.



What would be the result of U1 revoking authorisation from U4?

4. Consider the following grant authorisation graph



What would be the result of DBA revoking authorisation from U1?

5. Suppose that a database includes the following relations.

   - Student(sNo, sName, sAddress, sLevel, sMno, sPerformance)
   - Module(mNo, mTitle, mLevel, mLecturer)

   (a) The Users of the database are:

   Alfred who is authorised to do anything with the relations;

   Christian who can see and change all student details, but cannot register new students and cannot delete students;

   Robin who has the same authorisation as Christian; and

   George who has the same authorisation as same as Robin.

   Write SQL statements which implement the above.

   (b) There is another User, Jessica, who can only see final year students and final year modules. Jessica is not authorised to do anything else. Write SQL statements that implement this. Hint: you can use a view to achieve this!

   (c) Now suppose that George has now moved to another university and, therefore, should no longer have access to the database. Write SQL statements to implement this.

Chapter 35

Tutorial: Query Optimisation: Revision

35.1 Aim

The aim of this tutorial is for you to gain further insight into how a DBMS executes queries efficiently.

35.2 Exercise

1. How are query processing and query optimisation related?
2. Suppose that a relation r(rID, rName, rDescription) contains 5,000 tuples. Each tuple is comprised of a header (24 bytes) and three attributes: rID (6 bytes), rName (10 bytes), and rDescription (100 bytes). The size of each disc block is 1024 bytes and the size of the header of each disc block size 24 bytes.
   (a) How many blocks would be required to store the whole of this relation?
   (b) How many blocks would be required to store a projection resulting in the elimination of the attribute rDescription?
   (c) Explain why such a projection could be used to optimise a query whose results do not include data for the attribute rDescription.
3. (a) Write a relational algebra expressions for the following SQL query.

```
SELECT lecturers.name, students.name, courses.name
FROM lecturers, students, assessors, courses
WHERE lecturers.lid=assessors.lid
AND students.sid=assessors.sid
AND students.cid=courses.cid
AND courses.cid="CS";
```

   (b) Draw a query tree for this SQL query.

4. List the heuristics (rules of thumb) that optimisers apply to reduce the cost of optimisation.

5. (a) Draw a **near optimal** query tree for the following SQL query.

```
SELECT hotels.name
FROM hotels, ratings
WHERE hotels.rid=ratings.rid
AND hotels.hid=10
AND ratings.rating>7;
```

   (b) Write a relational algebra expressions for this tree.

Chapter 36

Tutorial: Precedence Graphs: Revision

36.1 Aim

The aim of this tutorial is for you to gain further experience of determining whether a non-serial schedule will allow transactions to execute concurrently without interfering with one another, and thereby produce a database state that could be produced by a serial execution.

36.2 Exercise

1. State whether each of the following is true or false. If it is false then briefly explain why.

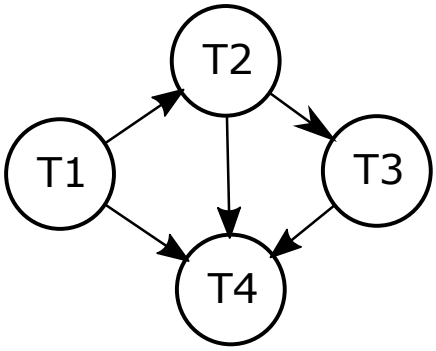
   (a) A transaction in which all steps must be completed successfully or none of them will be completed is called a durable transaction.

   (b) Two-phased locking can be used to ensure that transactions are serializable.

   (c) Although two transactions may be correct in themselves, interleaving of operations may produce an incorrect result.

   (d) Transactions should be written to the log before they are applied to the database itself.

2. Consider the following precedence graph for a non-serial schedule consisting of four transactions. Is the corresponding schedule conflict serializable? Justify your answer. If it is conflict-serializable give a corresponding serial schedule.



3. For each of the schedules shown in Table 36.1, draw a precedence graph, determine whether it is conflict serializable and justify your answer.

Schedule (a)

T1	T2	T3
	Read(y)	
		Read(y)
Write(x)		
		Write(y)
	Read(x)	
Read(y)		
		Write(y)

Schedule (b)

T1	T2	T3
		Write(y)
Read(y)		
	Read(x)	
Read(x)		
	Write(y)	
		Read(y)

Table 36.1: Two schedules

Chapter 37

Tutorial: Recovery: Revision

37.1 Aim

The aim of this tutorial is to further develop your understanding of recovery.

37.2 Exercise

1. List four facilities that a DBMS should provide to help recover from failures. State the purpose of each facility.

2. For the log file shown in Table 37.1 categorise each transaction as safe, needs to be undone, needs to be redone, or can be ignored. Assume that there was a failure at 14:28 and the DBMS was using the **deferred** update protocol.

Transaction ID	Time	Operation	Object	Before	After
T1	14:11	Start			
T2	14:13	Start			
T3	14:14	Start			
T1	14:16	Write	B	0	14
T2	14:18	Write	D	25	59
T1	14:19	Write	B	14	35
T2	14:21	Commit			
T3	14:23	Write	D	59	19
	14:24	Checkpoint			
T3	14:25	Write	D	19	23
T1	14:27	Commit			

Table 37.1: Log file for Friday.

3. For which of the ACID properties of transactions is the DBMS recovery subsystem responsible?

4. For the log file shown in Table 37.2, categorise each transaction as safe, needs to be undone, needs to be redone, or can be ignored. Assume that there was a failure at 19:35 and the DBMS was using the **immediate** update protocol.

Transaction ID	Time	Operation	Object	Before	After
T2	19:11	Start			
T3	19:13	Start			
T3	19:14	Write	C	0	24
T3	19:15	Commit			
T1	19:16	Start			
T1	19:18	Write	B	0	32
T2	19:19	Write	D	13	9
T2	19:21	Write	D	9	5
	19:23	Checkpoint			
T4	19:24	Start			
T1	19:27	Commit			
T5	19:29	Start			
T4	19:31	Write	D	5	12
T4	19:32	Commit			
T5	19:34	Write	E	0	4

Table 37.2: Log file for Saturday.

Bibliography

Barker, R. (1989). *CASE method : entity relationship modelling*. Number ISBN: 9780201416961. Addison-Wesley, first edition.

Codd, E. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, 13(6), 377–387.

Codd, E. (1971). A data base sublanguage founded on the relation calculus. In *Proceedings of the 1971 ACM SIGFIDET (Now SIGMOD) Workshop on Data Description, Access and Control* (pp. 35–68). San Diego, California: ACM New York, NY, USA.

Connolly, T. & Begg, C. (2004). *Database Solutions: A Step-by-step Guide to Building Databases*. Number ISBN: 0321173503. Pearson Education, second edition.

Connolly, T. & Begg, C. (2014). *Database Systems: A Practical Approach to Design, Implementation and Management: Global Edition*. Number ISBN: 9781292061184. Pearson, sixth edition.

Donahoo, M. & Speegle, G. (2005). *SQL: practical guide for developers*. Number ISBN: 9780122205316. Morgan Kaufmann, first edition.

Silberschatz, A., Korth, H., & Sudarshan, S. (2019). *Database Systems Concepts*. Number ISBN: 9781260084504. McGraw-Hill, seventh edition.

