OXFORD APPLIED MATHEMATICS AND COMPUTING SCIENCE SERIES C.1874

## A Pascal Database Book

## JULIAN ULLMANN



## CARTLA

# BIBLIOTHEQUE DU CERIST

# BIBLIOTHEQUE DU CERIST





General Editors J. N. Buxton, R. F. Churchhouse, and A. B. Tayler

## OXFORD APPLIED MATHEMATICS AND COMPUTING SCIENCE SERIES

- I. Anderson: A First Course in Combinatorial Mathematics
- D. W. Jordan and P. Smith: Nonlinear Ordinary Differential Equations
- D. S. Jones: Elementary Information Theory
- B. Carré: Graphs and Networks
- A. J. Davies: The Finite Element Method
- W. E. Williams: Partial Differential Equations
- R. G. Garside: The Architecture of Digital Computers
- J. C. Newby: Mathematics for the Biological Sciences
- G. D. Smith: Numerical Solution of Partial Differential Equations (Third Edition)
- J. R. Ullmann: A Pascal Database Book
- S. Barnett and R. G. Cameron: Introduction to Mathematical Control Theory (Second Edition)
- A. B. Tayler: Mathematical Models in Applied Mechanics



## A Pascal database book

· Ł.,

CLARENDON PRESS · OXFORD 1985

Oxford University Press, Walton Street, Oxford OX2 6DP. Oxford New York Toronto Delhi Bombay Calcutta Madras Karachi Kuala Lumpur Singapore Hong Kong Tokyo Nairobi Dar es Salaam Cape Town Melbourne Auckland and associated companies in Beirut Berlin Ibadan Nicosia

Oxford is a trade mark of Oxford University Press

Published in the United States by Oxford University Press, New York

© Julian Ullmann, 1985

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission of Oxford University Press

This book is sold subject to the condition that it shall not, by way of trade or otherwise, be lent, re-sold, hired out or otherwise circulated without the publisher's prior consent in any form of binding or cover other than that in which it is published and without a similar condition including this condition being imposed on the subsequent purchaser

British Library Cataloguing in Publication Data

Ullmann, Julian R. A Pascal database book – (Oxford applied mathematics and computing science series) 1. Data base management 2. File organization (Computer science) 3. PASCAL (Computer program language) I. Title 001.64'42 QA76.9.D3 ISBN 0-19-859643-X ISBN 0-19-859642-1 (Pbk.) Library of Congress Cataloging in Publication Data Ullmann, Julian R. (Julian Richard), 1936-A Pascal database book (Oxford applied mathematics and computing science series) Includes index. 1. Data base management. 2. PASCAL (Computer program language) 1. Title. II. Series. QA76.9.D3U45 85-4775 001.64'2 ISBN 0-19-859643-X ISBN 0-19-859642-1 (Pbk.)

Typeset by Latimer Trend & Company Ltd, Plymouth L Printed in Great Britain at the University Press, Oxford by David Stanford, Printer to the University

## Preface

Database technology is of immense importance in practical computing. Indeed, for many students it will prove to be one of the areas of computer science that is most relevant vocationally.

This book has been written primarily for students at universities and polytechnics who use Pascal as their principal programming language. One of the main aims of this book is to teach database technology so that its relation to a Pascal cultural framework can be clearly seen, thus improving the coherence of a student's overall view of computer science and avoiding unhelpful but traditional compartmentalization.

Another main aim of this book is to enable database technology to be taught earlier in the computer science curriculum than has been usual hitherto. We have found that concepts such as fifth normal form and CODASYL data structures are within the grasp of good first-year students. Reasons for teaching database technology early are as follows.

(a) Near the beginning of a computer science course it is particularly important to introduce practical applications of computing, and the database field provides plenty of opportunities for this.

(b) Database technology includes some intellectually appealing concepts, for example in normalization theory. This may help to motivate further study of computer science.

(c) Full details of database management systems such as CODASYL systems are apt to be complicated, and are perhaps not the kind of material that an undergraduate should be expected to learn. By teaching earlier we feel more free to concentrate on principles rather than non-academic although vocationally important details.

(d) Early in a computer science curriculum a database course can serve as a useful vehicle for programming practice, particularly with files, records, pointers, and uses of indirection.

(e) An early database course also serves to introduce important general features of computer science, for example the distinction between higher and lower levels of programming. (f) If the teaching of database technology is delayed until the final stages of the curriculum this may preclude subsequent teaching of database architectures.

Yet another aim of this book is to introduce rudiments of data processing in the same context as database technology, rather than in a separate educational compartment. For example Chapter 8 introduces check digits in the same context as database integrity constraints. Chapter 5 introduces batch processing as well as Btrees: B-trees are used later to implement database access paths.

An introduction to database technology is surely most convincing if built on at least an elementary grasp of practical uses of files. For this reason Chapter 1 concentrates simply on typical data and its uses. At the same time, Chapter 1 is intended to provide a meaningful initial introduction to files, records, fields, and keys.

A relational view of a database is at a higher level than a view in which access paths are visible. A purely relational view should be taught first to consolidate understanding of uses of files, and to show that a relational view can be self-contained. A relational view is purer and simpler than a lower-level view, and there is much to be said for introducing simple ideas before elaborate ones. After an introduction to the relational level the student is better equipped to appreciate the access paths that are visible at a lower level, and to see how these can support relational technology. The common but strategically mistaken practice of teaching file organization before relational technology is a reflection of the history of development of the field, rather than being the outcome of a deliberate search for the most effective way to teach the subject.

For the elementary purposes of this book we have chosen to introduce just one relational query language and one lower-level language which is actually an extension of Pascal. At each level we prefer to provide a serviceable introduction to one language, rather than a frustratingly superficial introduction to many.

From the cornucopia of relational query languages we have chosen relational algebra because this is helpful for normalization theory in Chapter 3. Furthermore, the names of operators of relational algebra are very useful in Chapters 6 and 7. We introduce relational algebra as a stand-alone query language which is not embedded in a host language.

At a lower level it is educationally desirable for students to see how database access path programming relates to standard Pascal programming. For this purpose Chapters 4 and 5 introduce a minimal extension to Pascal, including data manipulation language procedures such as INSERT, FIND, and REMOVE; thereby we hope to help bridge any culture gap that may have existed between Pascal programming and commercial data manipulation language programming. This extension of Pascal is deliberately at a lower level than other extensions that endow Pascal with relational query language capabilities in order to increase the power of the tools available to the programmer.

Experience suggests that we should systematically build upon that which the student understands, even if this means proceeding from the particular to the general, having first consolidated understanding of the particular by means of specific detailed examples. This is why this book starts with detail and ends in Chapter 8 with a more abstract introduction to database administration. Students would not find Chapter 8 so palatable if the book started with it, although this would be a fairly traditional starting point.

Chapter 9 is really an appendix. It is an informal introductory outline of standard Pascal data structuring facilities, and is designed to be read before Chapter 4 if required.

Many textbooks on Pascal have identifiers in lower case, or in mixed case as in SalesOrderHeader. The usual convention in the data-processing and database literature is to have identifiers in upper case. In this book we should like to follow the convention that has been established with the programming language ADA, but because standard Pascal does not allow underbars in identifiers, as in SALES\_ORDER\_HEADER, we close up spaces, as in SALESORDERHEADER, in programs and expressions. Otherwise we leave spaces within identifiers, as in SALES ORDER HEADER, for readability. This inconsistency appears to be the most palatable of the available options.

Thanks are due to Siobhan North, Hugh Lafferty, Jim McGregor, and Laurence Atkinson for helpful comments on various parts of the manuscript. The Pascal enhancement introduced in this text has been fully implemented (on a Prime 750 under PRIMOS) by Siobhan North (see A Pascal database management system, *Journal* of Pascal, Ada and Modula 2, Vol. 3, No. 6 (1984), pp 15–22).

Sheffield December 1984

J.R.U.

# BIBLIOTHEQUE DU CERIST

## Contents

## 1. AN INTRODUCTION TO DATA PROCESSING

1.1.	An introduction to data that is processed	1
1.2.	Order entry	2
1.3.	Shipping and invoicing	8
1.4.	Purchase orders from Typico to suppliers	13
1.5.	Aspects of accounting	16
1.6.	Accounts payable	19
1.7.	The general ledger	20
1.8.	Employees file and payroll	24
1.9.	Exercises	28

### 2. RELATIONAL ALGEBRA

2.1.	Introduction	30
2.2.	Music files	30
2.3.	Operators and operands	34
2.4.	Projection	35
2.5.	Selection	36
2.6.	Very simple queries	37
2.7.	Join	38
2.8.	Exercises	43
2.9.	Union	44
2.10.	Intersection	45
2.11.	Difference	46
2.12.	Division	47
2.13.	Product	49
2.14.	Exercises	51
2.15.	Query optimization	52
2.16.	Numerical capabilities of query languages	53
2.17.	Query language capabilities other than retrieval	53

### 3. RELATIONAL NORMALIZATION

3.1.	Allocation of fields to files	54
3.2.	Non-loss decomposition	54
3.3.	Duplication	55
3.4.	Dangling records	59
3.5.	Normalization	62

Contents
----------

	3.6. 3.7. 3.8. 3.9. 3.10. 3.11.	Functional dependence Normalization based on functional dependence Exercises Fourth normal form Entities and attributes Exercises	67 70 74 75 77 87
4.	PEF	SISTENT PROGRAMMING	
	4.1. 4.2. 4.3. 4.4. 4.5. 4.6.	Higher and lower levels of programming Pascal database declaration Programs that refer to a database declaration Persistent pointers Further Pascal database facilities Exercises	93 94 95 99 107 108
5.	AN	INTRODUCTION TO FILE ORGANIZATION	
	5.1. 5.2. 5.3. 5.4. 5.5. 5.6. 5.7. 5.8.	Organization for access Sorted order terminology Simple sequential files Hit rate and efficiency of access Address calculation Indexed sequential files File maintenance B-trees	<ol> <li>111</li> <li>111</li> <li>113</li> <li>121</li> <li>122</li> <li>135</li> <li>149</li> <li>149</li> </ol>
6.	імі	PLEMENTATION OF SELECTION OPERATIONS	
	<ul> <li>6.1.</li> <li>6.2.</li> <li>6.3.</li> <li>6.4.</li> <li>6.5.</li> <li>6.6.</li> <li>6.7.</li> </ul>	Introduction Categories of indexes Implementation of access to members Implementation of access to owners Selection operations on more than one key value Indirection Exercises	169 170 174 186 191 194 198
7.	IMI	PLEMENTATION OF JOINS	
	7.1. 7.2. 7.3. 7.4. 7.5. 7.6. 7.7. 7.8.	Hierarchical interfile indexes Two-way interfile indexes Exercises CODASYL DBTG TOTAL ADABAS Implementation of joins using secondary indexes Exercises	203 208 229 231 241 244 251 252

x

## Contents

## 8. ADMINISTRATIVE ISSUES

8.1.	Integrity control	255
8.2.	Recovery from failure	259
8.3.	Concurrency control	262
8.4.	Access control	268
8.5.	Views and subschemas	273
8.6.	Data independence	276
8.7.	Database administrator	276
8.8.	Data dictionary	277

## 9. APPENDIX: PASCAL FILES, RECORDS, AND POINTERS

9.1.	Introduction	280
9.2.	Files of text	281
9.3.	Type declarations	285
9.4.	Pascal record types and variables	292
9.5.	Binary files	300
9.6.	File buffers	302
9.7.	Exercises	306
9.8.	Pointers	309
9.9.	Exercises	325
INDEX		329