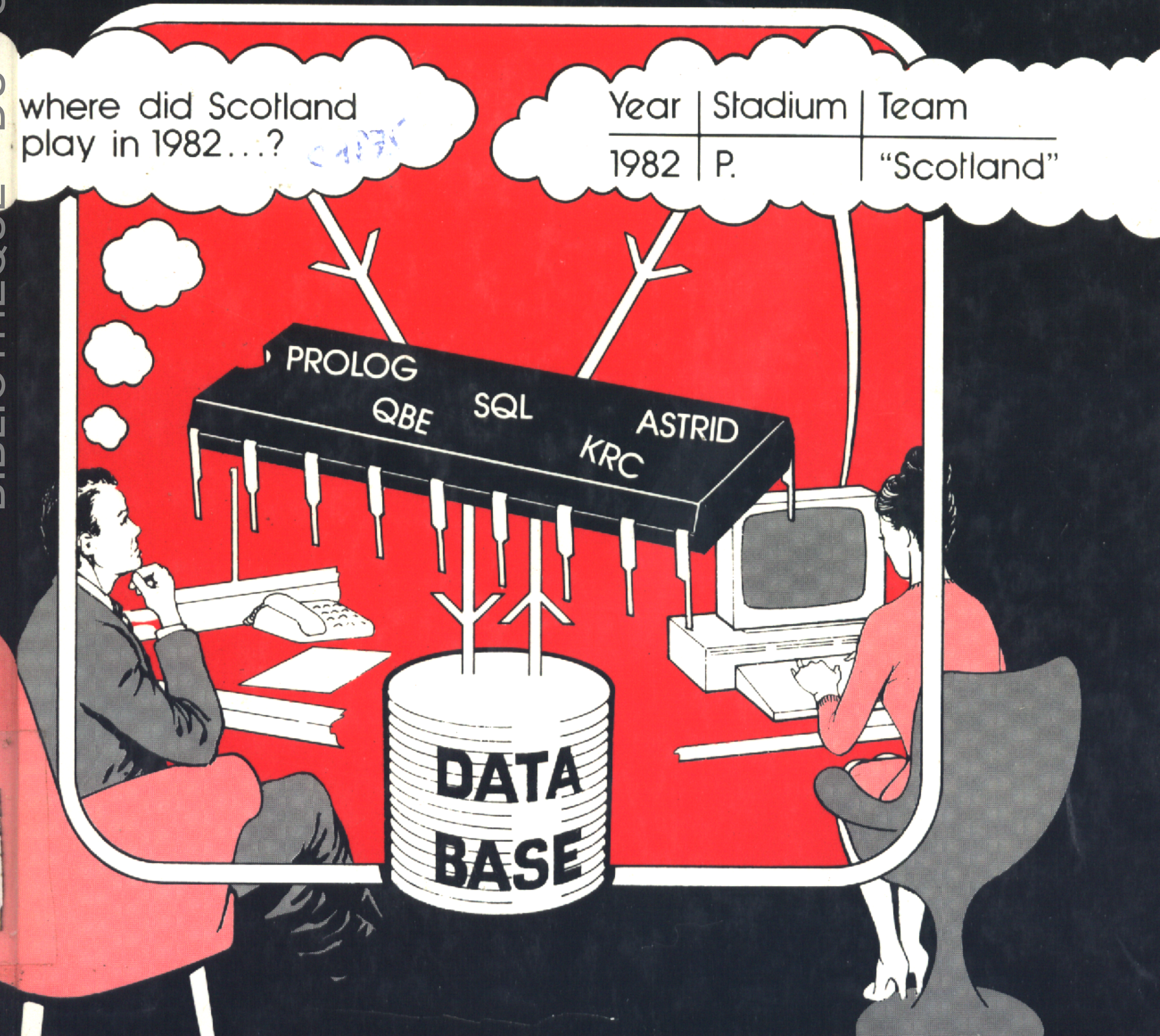
Ellis Horwood Series
COMPUTERS AND THEIR APPLICATIONS 29

Logic, Algebra and Databases

Peter Gray



Year	Stadium	Team
1982	P.	"Scotland"

BIBLIOTHEQUE DU CERIST

C 1275

LOGIC, ALGEBRA AND DATABASES

BIBLIOTHEQUE DU CERIST

BIBLIOTHEQUE DU CERIST

ELLIS HORWOOD SERIES IN COMPUTERS AND THEIR APPLICATIONS

Series Editor: Brian Meek, Director of the Computer Unit, Queen Elizabeth College, University of London

Atherton, R.	Structured Programming with COMAL
Berry, R.E.	Programming Language Translation
Brailsford, D.F. and Walker, A.N.	Introductory ALGOL 68 Programming
Bull, G.M.	The Dartmouth Time Sharing System
Burns, A.	New Information Technology
Burns, A.	The Microchip: Appropriate or Inappropriate Technology
Cope, T.	Computing using BASIC: An Interactive Approach
Dahlstrand, I.	Software Portability and Standards
Davie, F.J.T. and Morrison, R.	Recursive Descent Compiling
Deasington, R.J.	A Practical Guide to Computer Communications and Networking Second Edition
Deasington, R.J.	X.25 Explained
Ennals, R.	Logic Programmers and Logic Programming
Fossum, E.	Computerization of Working Life
Gray, P.M.D.	Logic, Algebra and Databases
Harland, D.M.	Polymorphic Programming Languages
Hill, I.D. and Meek, B.L.	Programming Language Standardisation
Hutchison, D.	Fundamentals of Computer Logic
McKenzie, J. and Lewis, R.	Interactive Computer Graphics in Science Teaching
Matthews, J.	FORTH
Meek B.L. and Fairthorne, S.	Using Computers
Meek, B.L., Heath, P. and Rushby, N.	Guide to Good Programming Practice, 2nd Edition
Millington, D.	Systems Analysis and Design for Computer Application
Moore, L.	Foundations of Programming with PASCAL
Pemberton, S. and Daniels, M.	PASCAL Implementation
Pesaran, H.M. and Slater, L.J.	Dynamic Regression: Theory and Algorithms
Sharp, J.A.	Data Flow Computing
Smith, I.C.H.	Microcomputers in Education
Spath, H.	Cluster Analysis Algorithms
Spath, H.	Cluster Dissection Algorithms
Stratford-Collins, M.J.	ADA: A Programmer's Conversion Course
Teskey, F.N.	Principles of Text Processing
Thimbleby, H.	Principles of User Interface Design
Turner, S.J.	An Introduction to Compiler Design
Young, S.J.	An Introduction to ADA
Young, S.J.	Real Time Languages

ELLIS HORWOOD BOOKS IN COMPUTING

Atherton, R.	Structured Programming with BBC BASIC
Barrett, T.P. and Colwill, S.	Winning Games on the Commodore 64
Barrett, T.P. and Jones, A.J.	Winning Games on the VIC-20
Christensen, B.	Beginning COMAL
Cole, D.G.J.	Getting Started on the ORIC-1
Cotton, K. et al.	Information Technology and the New Generation
Ennals, R.	Beginning micro-PROLOG
Goodyear, P.	LOGO: A Guide to Learning through Programming
Jones, A.J. and Carpenter, G.	Mastering the Commodore 64
Jones, A.J., Coley, E.A. and Cole, D.G.J.	Mastering the VIC-20
Matthews, T. and Smith, P.	Winning Games on the ZX Spectrum
Moore, L.	Mastering the ZX Spectrum
Narayanan, A.	Beginning LISP
Simon and Matthews, J.	Mastering the Electron

مركز البحث في [illegible]

LOGIC, ALGEBRA AND DATABASES

PETER M. D. GRAY, MA, D. Phil, MBCS
Senior Lecturer
Department of Computing Science, King's College
University of Aberdeen, Scotland

ELLIS HORWOOD LIMITED
Publishers · Chichester
Halsted Press: a division of
JOHN WILEY & SONS
New York · Chichester · Brisbane · Toronto

BIBLIOTHEQUE DU CERIST

First published in 1984
and Reprinted in 1985 by

ELLIS HORWOOD LIMITED
Market Cross House, Cooper Street, Chichester, West Sussex, PO19 1EB, England

The publisher's colophon is reproduced from James Gillison's drawing of the ancient Market Cross, Chichester.

Distributors:

Australia, New Zealand, South-east Asia:
Jacaranda-Wiley Ltd., Jacaranda Press,
JOHN WILEY & SONS INC.,
GPO Box 859, Brisbane, Queensland 4001, Australia

Canada:
JOHN WILEY & SONS CANADA LIMITED
22 Worcester Road, Rexdale, Ontario, Canada.

Europe, Africa:
JOHN WILEY & SONS LIMITED
Baffins Lane, Chichester, West Sussex, England.

North and South America and the rest of the world:
Halsted Press: a division of
JOHN WILEY & SONS
605 Third Avenue, New York, NY 10158, USA

© 1984 P.M.D. Gray/Ellis Horwood Limited

British Library Cataloguing in Publication Data
Gray, Peter M.D.
Logic, algebra and databases. –
(Ellis Horwood series in computers and their applications)
1. Data base management 2. Logic, symbolic and mathematical
I. Title
001.64'42 QA76.9.D3

Library of Congress Card No. 84-12854

ISBN 0-85312-709-3 (Ellis Horwood Limited – Library Edn.)
ISBN 0-85312-803-0 (Ellis Horwood Limited – Student Edn.)
ISBN 0-470-20103-7 (Halsted Press – Library Edn.)
ISBN 0-470-20259-9 (Halsted Press – Student Edn.)

Typeset by Ellis Horwood Limited
Printed in Great Britain by R.J. Acford, Chichester

4968

COPYRIGHT NOTICE –
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 permission of Ellis Horwood Limited, Market Cross House, Cooper Street, Chichester, West Sussex, England.

BIBLIOTHEQUE DU CERIST

BIBLIOTHEQUE DU CERIST

Table of Contents

Author's Preface 13

Acknowledgements 15

CHAPTER 1 The Propositional Calculus
1.1 Introduction 17
1.2 Truth and falsehood of propositions – two-valued logic 18
1.3 Compound propositions 18
1.4 Truth tables for basic operations 19
1.5 Theorems from Boolean Algebra 20
1.6 De Morgan's laws 22
1.7 The distribution law – propositions in normal form 22
1.8 The implies operator 24
1.9 Rules of inference: chain rule and modus ponens 25
1.10 Chains of inference 26
1.11 Tautologies and instantiation 27
1.12 Logical equivalences and substitution 28
1.13 Proof by adopting a premise 29
1.14 Reductio ad absurdum 30
1.15 Proof by resolution 31
1.16 Advantages of resolution 33

CHAPTER 2 The Predicate Calculus
2.1 Introduction – parametrised propositions 34
2.2 The universal quantifier ∀ 35

2.3 The existential quantifier ∃ 36
2.4 Hints on writing calculus statements 36
2.5 Combinations of quantifiers 37
  2.5.1 Renaming of bound variables 38
  2.5.2 Mixed quantifiers 39
2.6 Formal syntax 39
2.7 Semantic constraints of the calculus 40
2.8 A set-theoretic model 40
2.9 Negating quantifiers 42
2.10 Examples of quantified statements 42
2.11 Quantified statements about data 43
2.12 Equivalence of different formulae 44
2.13 Conversion to clausal form 44
2.14 The resolution principle 46
2.15 Using resolution for question answering 46
2.16 Heuristics for proof 48
2.17 Substitution and unification 49
2.18 Summary 50

BIBLIOTHEQUE DU CERIST

CHAPTER 3 Lambda Expressions and List Processing

3.1 Introduction 51
3.2 Functions viewed as mappings between sets 52
  3.2.1 Ordered pairs 53
  3.2.2 Functions of multiple arguments 55
  3.2.3 The Cartesian product 56
3.3 Composition of functions: the compound statement 56
3.4 Restriction and union of functions: the 'IF' statement 58
  3.4.1 Use of the restrict operator 59
3.5 Forms of applicative syntax: the Lambda notation 60
3.6 Writing recursive definitions in lambda notation 62
  3.6.1 Downgoing and upgoing recursion 63
3.7 Recursive list processing 63
  3.7.1 Sequential access by recursion 64
  3.7.2 List construction 64
  3.7.3 Implementation 65
  3.7.4 Representing lists in Pascal 66
3.8 Styles of recursive functions 67
  3.8.1 Hints for writing recursive definitions 67
  3.8.2 Tree recursion 68
  3.8.3 How to manage without assignment and 'goto' 69
3.9 Some set and list processing functions in lambda notation 70
  3.9.1 Member 70
  3.9.2 Intersection 71
3.10 Summary 72

BIBLIOTHEQUE DU CERIST

CHAPTER 4 Representing Programs by Clauses: Prolog

4.1 History of logic programming 73
4.2 Elements of the Prolog language 74
 4.2.1 Execution of a Prolog query 74
 4.2.2 Constructing alternative solutions 75
 4.2.3 Prolog syntax 76
4.3 Evaluation viewed as goal-directed depth-first search 76
4.4 Example queries on a Prolog database 78
4.5 Axiomatisation of list processing operations 80
4.6 The mechanism for generating alternative answers 81
4.7 Functions as record constructors 83
 4.7.1 Problems of using selector functions in Prolog 84
 4.7.2 Prolog syntax for lists 84
 4.7.3 General record structures 85
 4.7.4 Type checking and unification 86
 4.7.5 Tree structures 86
4.8 Resolution related to depth-first search 87
 4.8.1 Horn clauses 87
 4.8.2 Evaluation in Prolog viewed as theorem proving 88
4.9 Special predicates in Prolog 89
 4.9.1 Evaluable predicates 89
 4.9.2 Use of meta-level predicates 89
4.10 Converting from lambda notation to Prolog 91
4.11 Some basic set and list processing procedures 92
4.12 Database and set operations in Prolog 94
 4.12.1 Use of 'assert' and 'retract' 94
 4.12.2 Constructing sets of answers 95
4.13 Sumary 96

CHAPTER 5 Representing Programs in Functional Notation

5.1 Functional programming 97
5.2 Functional syntax: KRC 98
5.3 Functions that control iteration: maplist, lit 100
5.4 Function-producing functions 101
 5.4.1 Partial application 102
 5.4.2 Curried functions 102
5.5 The combinators of Curry and Turner 103
5.6 Functions that generate sets 105
5.7 Lazy evaluation 106
5.8 Functions that operate on streams: examples from FQL 108
 5.8.1 Use of generators and function composition 108
 5.8.2 The tupling operator 109
5.9 Function summary 109
 5.9.1 Database functions 110

5.10 Generalised types for functions: examples from HOPE 111
5.10.1 Abstract data types in HOPE 112
5.10.2 Types in FQL 113
5.11 Summary 113

CHAPTER 6 The Relational Model

6.1 Introduction 115
6.2 General features of relations in a relational database 116
6.2.1 A relation considered as a file 116
6.2.2 Keys 116
6.2.3 Value-based matching 118
6.2.4 Normal form 118
6.2.5 Similarity to tables 119
6.2.6 Relation definition — summary 119
6.3 The binary relational model 119
6.4 The entity relationship model 121
6.5 Normalisation 122
6.6 Generalised set operations on two relations 124
6.6.1 Relation composition by union 124
6.6.2 Relation composition by join 124
6.6.3 Definition of join and equi-join 126
6.7 Differences between relations and files in Pascal and Cobol 127
6.7.1 Domain information 127
6.7.2 Group fields 127
6.7.3 Repeating groups 127
6.7.4 Variant records 129
6.7.5 Use of pointers in Pascal 129
6.7.6 Null values 130
6.8 Sub-types and type hierarchies 131
6.8.1 Component hierarchy 132
6.8.2 Sub-type hierarchy 133
6.9 Summary 134

CHAPTER 7 Calculus-Based Languages

7.1 Introduction 136
7.2 The QUEL language of INGRES as a tuple-oriented calculus 136
7.3 Syntax of QUEL 137
7.3.1 Two-variable queries 138
7.3.2 Problems of negation and quantification 139
7.4 SQL and other languages 140
7.4.1 Simple queries in SQL 141
7.4.2 Quantified queries in SQL 142
7.4.3 Two variables in the same relation 143
7.4.4 Syntax of SQL with nested sub-queries 143
7.4.5 UNION and SET DIFFERENCE in SQL 144

BIBLIOTHEQUE DU CERIST

7.5 QBE: query-by-example 145
7.5.1 QBE as a domain-oriented calculus 146
7.5.2 Correspondence between QBE and Prolog 147
7.6 Examples from QBE compared to SQL 147
7.7 Use of output relations, snapshots and views 149
7.7.1 Snapshots and views 149
7.8 The GROUP_BY operation in SQL and QBE 150
7.9 Summary 154

CHAPTER 8 Relational Algebra: an Applicative Language

8.1 Introduction 155
8.2 The four basic algebraic operations 156
8.2.1 Selection 156
8.2.2 Projection 156
8.2.3 Union 158
8.2.4 Join 158
8.2.5 Degree and cardinality 159
8.3 Correspondence with data processing operations on serial files 160
8.4 Combining the basic operations 161
8.5 Comparison of algebra and calculus 162
8.5.1 Relational algebra as an applicative language 163
8.6 The ASTRID language: syntax for a generalised algebra 163
8.6.1 Cardinal (two-relation) operators 163
8.6.2 Modifier (single-relation) operators 164
8.7 Use of Join with rename 164
8.7.1 The theta-join operation of Codd 166
8.7.2 Traversing trees in relational algebra 166
8.8 Example queries in ASTRID compared to QBE 167
8.9 The extend operation and derived fields 169
8.10 The GROUP_BY operation as an extended projection 170
8.11 The generalised difference operation and universal quantification 173
8.12 Forming complex multi-line queries 174
8.13 Summary 175

CHAPTER 9 Query Transformation in Algebra and Calculus

9.1 Introduction 176
9.1.1 Motives for query transformation 176
9.1.2 Advantages of query translation for users 177
9.1.3 Outline of chapter 177
9.2 Conversion from relational algebra to calculus 178
9.2.1 Correspondence of algebraic operations and set expressions 178
9.3 Optimising algebraic expressions — re-ordering operations 180
9.3.1 Moving selections through joins 180
9.3.2 Combining multiple selections 181
9.3.3 Combining selections and projections 182
9.3.4 Moving projections through joins 182

BIBLIOTHEQUE DU CERIST

9.4 Optimising union and difference operations 183
- 9.4.1 Moving selections 183
- 9.4.2 Moving projections 184
- 9.4.3 Combinations of selections and projections 184
- 9.4.4 Implementation of transformations 185

9.5 Transforming expressions involving GROUP_BY 185
- 9.5.1 General equivalences with 'any' and 'all' 186

9.6 The divide operation and its equivalences 187
- 9.6.1 Equivalence with GROUP_BY 187
- 9.6.2 Equivalent to double difference 189

9.7 Other useful equivalences from set theory 189

9.8 The PRTV optimisation algorithm 190

9.9 Translation from QBE to relational algebra 190
- 9.9.1 Selections on a single relation 191
- 9.9.2 Equi-joins and self-joins 191
- 9.9.3 Extension and GROUP_BY 192
- 9.9.4 General algorithm 192

9.10 Codd's algorithm for converting relational calculus to algebra 194
- 9.10.1 WFF in Alpha 194
- 9.10.2 Conversion of range-separable WFFs 195

9.11 Improved version of Codd's algorithm using GROUP_BY 196
- 9.11.1 Validity of the algorithm 197
- 9.11.2 An example of the algorithm 198

9.12 Summary 198

CHAPTER 10 The Functional Data Model

10.1 Basic notions — entities and functions 201
- 10.1.1 Classes of functions 202
- 10.1.2 Corresponding with binary relations 203
- 10.1.3 Many—many relations 203
- 10.1.4 One-to-one relations 204
- 10.1.5 Sub-types 205
- 10.1.6 Schema description 205

10.2 Function composition in the functional model 205
- 10.2.1 Similarity to join of binary relations 207

10.3 Comparisons of syntax for data manipulation 208
- 10.3.1 Adaplex 208
- 10.3.2 Daplex 208
- 10.3.3 Syntax for updates in Adaplex 209

10.4 Representing a functional schema by relations 210

10.5 Representing relations by a Daplex schema 211

10.6 Relationship to abstract data types 212

10.7 Summary — key ideas 212
- 10.7.1 An abstract basis for CODASYL 212
- 10.7.2 Use of references not values 213

BIBLIOTHEQUE DU CERIST

10.7.3 Multi-valued function composition and join 213
10.7.4 Databases are for stored functions 213
10.7.5 Making schemas extensible 213

CHAPTER 11 The CODASYL DBTG Database

11.1 Introduction 215
11.2 Representing sets by chains and pointers 216
11.2.1 CODASYL sets 217
11.2.2 Placement modes 218
11.3 A CODASYL schema and its Bachman diagram 218
11.3.1 The conceptual schema 219
11.3.2 Declaring a hierarchical (one-to-many) relationship 219
11.3.3 Declaring a net (many—many) relation 220
11.3.4 Declaring a generalised hierarchy 221
11.3.5 A cyclic hierarchy 221
11.4 Other details in the set description 223
11.4.1 Automatic and manual insertion 223
11.4.2 Insertion and retention class 223
11.4.3 Ordering within sets 224
11.4.4 Pointer array mode 225
11.4.5 Summary of set entry 225
11.5 The ANSI/SPARC proposals: external views and storage schemas 226
11.5.1 The sub-schema 226
11.5.2 The storage schema 227
11.5.3 The three-level schema 227
11.6 CODASYL DML operations 228
11.6.1 Use of variables in work areas and registers 228
11.7 The forms of the FIND command 230
11.8 Other DML commands 232
11.9 Summary 233

CHAPTER 12 Functional Query Languages for CODASYL Databases

12.1 Implementing the FQL language on a CODASYL database 235
12.1.1 An alternative to currency registers 235
12.1.2 Use of DML operations by an FQL interpreter 237
12.1.3 Summary 237
12.2 Implementing Daplex functions on CODASYL 237
12.2.1 Daplex composed functions 238
12.3 Representing a Daplex schema by CODASYL sets 239
12.3.1 Sub-types 239
12.4 Forming a relational view of a basic CODASYL schema 240
12.4.1 The anchor record approach 241
12.4.2 Zaniolo's method 241
12.4.3 Reiner and Rosenthal's method 243
12.4.4 The ASTRID Mapping 243
12.4.5 Mapping features in existing CODASYL databases 245

BIBLIOTHEQUE DU CERIST

12.5 Implementing relational algebra operations on CODASYL 246
- 12.5.1 Join methods for routines for perceived records 247

12.6 The ASTRID system of code modification 248
- 12.6.1 Traversals 248
- 12.6.2 Combination and modification of traversals 249
- 12.6.3 Modification by selection 249
- 12.6.4 Modification by extension 250
- 12.6.5 Modification by projection 250
- 12.6.6 Combination by JOIN 251
- 12.6.7 General conditions for join by overlap 252
- 12.6.8 Extensions to join algorithm 253

12.7 Problems of updates and their mapping 253
- 12.7.1 Simple update by modify 253
- 12.7.2 Problems of combined deletion and insertion 254
- 12.7.3 Temporary inconsistency 254
- 12.7.4 Problems of updating views 255

12.8 Integrity constraints 256
- 12.8.1 Constraints in Daplex and QBE 257
- 12.8.2 Problems of implementation 258

12.9 Future developments 259
- 12.9.1 Distributed databases 259
- 12.9.2 Meta data 260
- 12.9.3 Knowledge bases 260
- 12.9.4 Interpretations of logic 260

12.10 Conclusion 261

Appendix I **Relations for the World Cup database** 262
Appendix II **ASTRID relational algebra syntax** 269
Appendix III **Bachman diagram and IDS-II schema DDL for the World Cup database** 272
Appendix IV **Syntax for SQL** 277
Appendix V **Daplex syntax** 279

References 282

Index 289

BIBLIOTHEQUE DU CERIST