Contents

1 Introduction **1**

1.1 Database Management Systems 1

1.1.1 On-line Transaction Processing (OLTP) 2

1.1.2 Analytic Query Processing 2

1.1.3 Column Stores 3

1.1.4 MonetDB/X100 and Vectorwise 3

1.2 Research Questions 4

1.3 Thesis Outline and Contributions 5

2 Hardware overview **7**

2.1 Introduction 7

2.2 Latency and Bandwidth 8

2.3 Data Representation 8

2.4 CPU Architecture 9

2.4.1 Instructions and Clock Cycles 10

2.4.2 Pipelining 11

2.4.3 Super-scalar Execution 11

2.4.4 Hazards 11

2.4.5 IPC and ILP 14

2.4.6 Out-of-Order Execution 14

2.4.7 Role of Compilers 15

2.4.8 Coding Techniques 16

2.4.9 Advanced Architectures 17

2.4.10 Further CPU Trends 19

2.5 Hierarchical Memories 24

2.5.1 The “Memory Wall” 24

2.5.2 Hardware Caches 27

2.5.3 Virtual Memory 29

2.5.4 Memory Access in Multi-processors 30

2.6 Disk technology 33

2.6.1 Hard Disk Drives (HDD) 33

2.6.2 Solid State Drives (SSD) 35

2.6.3 RAID 39

2.7 Conclusion 40

3 Relational Database Management Systems 43
3.1 Relational Model . . . 43
3.2 Relational Algebra . . . 44
3.2.1 Overview . . . 44
3.2.2 Extended Operations . . . 46
3.2.3 Database Manipulation . . . 46
3.2.4 Structured Query Language (SQL) . . . 47
3.3 Storage Management . . . 48
3.3.1 Storage Models . . . 48
3.3.2 Buffer Manager . . . 51
3.3.3 Indexing . . . 52
3.4 Query Processing . . . 57
3.4.1 Overview . . . 57
3.4.2 Operator Selection . . . 58
3.5 Transaction Management . . . 60
3.5.1 ACID . . . 61
3.5.2 Concurrency Control . . . 62
3.6 Application Domains . . . 64

4 Vectorwise: a DBMS on Modern Hardware 65
4.1 Introduction . . . 65
4.2 Motivation . . . 66
4.3 Vectorized Execution . . . 67
4.4 Storage Management . . . 70
4.4.1 DSM . . . 70
4.4.2 Compression . . . 70
4.4.3 Buffer Management . . . 72
4.4.4 Indexing . . . 75
4.5 Transactional Updates . . . 79
4.6 Vectorwise . . . 80
4.6.1 SQL Front-end . . . 80
4.6.2 Parallelism . . . 82
4.6.3 Research Topics . . . 83
4.7 Related Work . . . 84

5 Column Compression 87
5.1 Introduction . . . 87
5.1.1 Contributions . . . 87
5.1.2 Outline . . . 89
5.2 Related Work . . . 89
5.3 Super-scalar Compression . . . 91
5.3.1 PFOR, PFOR-DELTA and PDICT . . . 92
5.3.2 Disk Storage . . . 92
5.3.3 Decompression . . . 93
5.3.4 Compression . . . 96
5.3.5 Fine-grained Access . . . 97
5.3.6 Compulsory Exceptions . . . 99
5.3.7 RAM-RAM vs. RAM-Cache Decompression . . . 99
5.3.8 Choosing Compression Schemes . . . 100
5.4 TPC-H Experiments . . . 101

5.5 Inverted File Compression 105
5.6 Conclusions and Future Work 106

6 Positional Updates 107
6.1 Introduction 107
6.1.1 Differential Updates 107
6.1.2 Positional Updates (PDT) 108
6.1.3 Outline 109
6.2 Terminology 109
6.2.1 Ordered Tables 109
6.2.2 Ordering vs. Clustering 110
6.2.3 Positional Updates 110
6.2.4 Differential Structures 110
6.2.5 Checkpointing 111
6.2.6 Stacking 111
6.2.7 RID vs. SID 111
6.3 PDT by Example 112
6.3.1 Inserting Tuples 112
6.3.2 Storing Tuple Data 113
6.3.3 Modifying Attribute Values 114
6.3.4 Deleting Tuples 115
6.3.5 RID $\Leftrightarrow$ SID 115
6.3.6 Value-based Delta Trees (VDTs) 117
6.3.7 Merging: PDT vs VDT 117
6.4 PDT in Detail 118
6.4.1 Properties 118
6.4.2 Design Decisions 118
6.4.3 Implementation Details 119
6.4.4 Lookup by SID and RID 120
6.4.5 MergeScan 121
6.4.6 Adding Updates 122
6.4.7 Disrespecting Deletes 127
6.5 Transaction Processing 128
6.5.1 Propagate 130
6.5.2 Overlapping Transactions 130
6.5.3 Serialize 131
6.5.4 Commit 131
6.5.5 Example 134
6.6 Logging and Checkpointing 135
6.7 Experimental Evaluation 137
6.7.1 Benchmark Setup 137
6.7.2 Update Micro Benchmarks 137
6.7.3 MergeScan Micro Benchmarks 138
6.7.4 TPC-H Benchmarks 140
6.8 Related Work 142
6.9 Related Systems 143
6.9.1 Microsoft SQL Server 143
6.9.2 Vertica 145
6.9.3 SAP HANA 148
6.10 Summary 150

7 Index Maintenance **151**
7.1 Introduction . . . 151
7.1.1 Outline . . . 151
7.2 Abbreviations and Terminology . . . 151
7.3 MinMax Maintenance . . . 153
7.3.1 Update Handling . . . 153
7.3.2 Example . . . 154
7.3.3 Range Scans . . . 156
7.4 Join Index Maintenance . . . 157
7.4.1 An Updateable Join Index Representation . . . 157
7.4.2 Join Index Updates . . . 158
7.4.3 Join Index Summary (JIS) . . . 159
7.4.4 Range Propagation . . . 162
7.5 Update Operators . . . 163
7.5.1 Insert . . . 163
7.5.2 Delete . . . 165
7.5.3 Modify . . . 166
7.6 Concurrency Issues . . . 166
7.6.1 Computing FRID for Child-PDT Inserts . . . 167
7.6.2 Deferred Index Maintenance . . . 168
7.6.3 Serializing Child-PDT Inserts . . . 170
7.7 Experiments and Optimizations . . . 174
7.7.1 Setup . . . 174
7.7.2 Explanation of Graphs . . . 175
7.7.3 Non-Clustered Baseline . . . 176
7.7.4 Clustered Table Layout . . . 180
7.7.5 Throughput and Total Times . . . 184
7.8 Summary and Conclusions . . . 185

8 Conclusion **187**
8.1 Vectorwise . . . 187
8.2 Light-weight Column Compression . . . 188
8.2.1 Contributions . . . 188
8.2.2 Research Impact . . . 189
8.2.3 Future Work . . . 190
8.3 Positional Differential Updates . . . 190
8.3.1 Contributions . . . 191
8.3.2 Discussion and Future Work . . . 192

A Information Retrieval using Vectorwise **197**
A.1 Introduction . . . 197
A.1.1 Contributions . . . 197
A.1.2 Outline . . . 198
A.2 TREC-TB Setup . . . 198
A.2.1 Overview . . . 198
A.2.2 Indexing . . . 198
A.3 Querying . . . 199
A.3.1 Keyword Search Using Relational Algebra . . . 199
A.3.2 Performance Optimizations . . . 201
A.3.3 Distributed Execution . . . 202

A.4 TREC-TB Results 203
A.5 Related work 205
A.6 Conclusion 206

Bibliography **206**

Summary / Samenvatting **221**