



Het SQL Leerboek

RICK F. VAN DER LANS



Het SQL Leerboek

Zevende herziene druk

25 jaar aan de top

Rick F. van der Lans



Meer informatie over deze en andere uitgaven kunt u verkrijgen bij:

Sdu Klantenservice

Postbus 20014

2500 EA Den Haag

tel.: (070) 378 98 80

www.sdu.nl/service

© 2012 Sdu Uitgevers, Den Haag

Academic Service is een imprint van Sdu Uitgevers bv.

1e druk 1986

2e druk 1987

3e druk 1988

4e druk 1992

5e herziene druk 1998

5e verbeterde druk 2004

6e druk, april 2006

7e herziene druk december 2011

Zetwerk: Redactie bureau Ron Heijer, Markelo

Omslag: Carlito's Design, Amsterdam

ISBN: 978 90 395 2655 2

NUR: 123 / 995

Alle rechten voorbehouden. Alle intellectuele eigendomsrechten, zoals auteurs- en databankrechten, ten aanzien van deze uitgave worden uitdrukkelijk voorbehouden. Deze rechten berusten bij Sdu Uitgevers bv en de auteur.

Behoudens de in of krachtens de Auteurswet gestelde uitzonderingen, mag niets uit deze uitgave worden verveelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand of openbaar gemaakt in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopieën, opnamen of enige andere manier, zonder voorafgaande schriftelijke toestemming van de uitgever.

Voor zover het maken van reprografische verveelvoudigingen uit deze uitgave is toegestaan op grond van artikel 16 h Auteurswet, dient men de daarvoor wettelijk verschuldigde vergoedingen te voldoen aan de Stichting Reprorecht (Postbus 3051, 2130 KB Hoofddorp, www.reprorecht.nl). Voor het overnemen van gedeelte(n) uit deze uitgave in bloemlezingen, readers en andere compilatiewerken (artikel 16 Auteurswet) dient men zich te wenden tot de Stichting PRO (Stichting Publicatie- en Reproductierechten Organisatie, Postbus 3060, 2130 KB Hoofddorp, www.cedar.nl/pro). Voor het overnemen van een gedeelte van deze uitgave ten behoeve van commerciële doeleinden dient men zich te wenden tot de uitgever.

Hoewel aan de totstandkoming van deze uitgave de uiterste zorg is besteed, kan voor de afwezigheid van eventuele (druk) fouten en onvolledigheden niet worden ingestaan en aanvaardt de auteur(s), redacteur(en) en uitgever deswege geen aansprakelijkheid voor de gevolgen van eventueel voorkomende fouten en onvolledigheden.

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 publisher's prior consent.

While every effort has been made to ensure the reliability of the information presented in this publication, Sdu Uitgevers neither guarantees the accuracy of the data contained herein nor accepts responsibility for errors or omissions or their consequences.

Inhoud

Over de auteur	xvii
Voorwoord	xix
Deel I Inleiding	1
1 Inleiding SQL	3
1.1 Inleiding	3
1.2 Database, databaseserver en databasetaal	3
1.3 Het relationele model	5
1.3.1 Tabel, kolom en rij	6
1.3.2 Null-waarde	7
1.3.3 Integriteitsregel	7
1.3.4 Primaire sleutel	8
1.3.5 Kandidaatsleutel	8
1.3.6 Alternatieve sleutel	9
1.3.7 Refererende sleutel	9
1.4 Wat is SQL?	10
1.5 Verschillende categorieën SQL-programma's	14
1.6 De geschiedenis van SQL	19
1.7 Van monolithisch via client/server naar internet	21
1.8 Transactiedatabases en datawarehouses	24
1.9 Standaardisatie van SQL	26
1.10 De markt van SQL-databaseservers	29
1.11 Welk SQL-dialect?	31
1.12 De structuur van het boek	32
2 De voorbeelddatabase van een tennisvereniging	33
2.1 Inleiding	33
2.2 Beschrijving van de tennisvereniging	33
2.3 De inhoud van de tabellen	36
2.4 Integriteitsregels	39
3 Installeren van de software	41
3.1 Inleiding	41
3.2 Downloaden van MySQL	41
3.3 Installeren van MySQL	41
3.4 Installeren van WinSQL	41
3.5 Ophalen van SQL-instructies van de website	42
3.6 Afsluiting	42

4	SQL in vogelvlucht	43
4.1	Inleiding	43
4.2	Inloggen op de MySQL-databaseserver	43
4.3	Creëren van nieuwe SQL-gebruikers	45
4.4	Creëren van databases	46
4.5	Selecteren van de courante database	47
4.6	Creëren van tabellen	48
4.7	Vullen van tabellen met gegevens	50
4.8	Raadplegen van tabellen	51
4.9	Wijzigen en verwijderen van rijen	53
4.10	Versnellen van het raadplegen met indexen	55
4.11	Views	56
4.12	Gebruikers en gegevensbeveiliging	57
4.13	Verwijderen van databaseobjecten	58
4.14	Systeemparemeters	59
4.15	Groepen SQL-instructies	60
4.16	De catalogustabellen	61
4.17	Definities van SQL-instructies	65
	Deel II Raadplegen en muteren van gegevens	67
5	SELECT-instructie: basisbegrippen	69
5.1	Inleiding	69
5.2	Constanten en hun datatypes	69
5.2.1	De integer-constante	70
5.2.2	De decimal-constante	71
5.2.3	Float-, real- en double-constanten	71
5.2.4	De alfanumerieke constante	72
5.2.5	De datumconstante	73
5.2.6	De tijdconstante	73
5.2.7	De timestamp-constante	74
5.2.8	De boolean-constante	75
5.2.9	De hexadecimal-constante	75
5.3	De expressie	76
5.4	Toekennen van namen aan resultaatkolommen	79
5.5	De kolomspecificatie	81
5.6	De gebruikersvariabele en de SET-instructie	82
5.7	De systeemvariabele	84
5.8	De case-expressie	85
5.9	De scalaire expressie tussen haakjes	90
5.10	De scalaire functie	91
5.11	Casting van expressies	94
5.12	De null-waarde als expressie	97
5.13	De samengestelde scalaire expressie	98
5.13.1	De samengestelde numerieke expressie	
5.13.2	De samengestelde alfanumerieke expressie	101
5.13.3	De samengestelde datumexpressie	103
5.13.4	De samengestelde tijdexpressie	106
5.13.5	De samengestelde timestamp-expressie	108

5.14	De aggregatiefunctie en de scalaire subquery	109
5.15	De rijexpressie	110
5.16	De tabellexpressie	111
6	SELECT-instructie, tabellexpressie en subquery	113
6.1	Inleiding	113
6.2	De definitie van de SELECT-instructie	113
6.3	Verwerking van de componenten in een select-blok	116
6.4	Mogelijke vormen van een tabellexpressie	122
6.5	Wat is een SELECT-instructie?	125
6.6	Wat is een subquery?	125
7	SELECT-instructie: de FROM-component	131
7.1	Inleiding	131
7.2	Tabelspecificaties in de FROM-component	131
7.3	Nogmaals de kolomspecificatie	133
7.4	Meerdere tabelspecificaties in de FROM-component	134
7.5	Pseudoniemen voor tabelnamen	138
7.6	Diverse voorbeelden van joins	139
7.7	Verplicht gebruik van pseudoniemen	142
7.8	Expliciete joins in de FROM-component	144
7.9	Outer-joins	147
7.9.1	De left-outer-join	147
7.9.2	De right-outer-join	151
7.9.3	De full-outer-join	151
7.10	Extra condities in de join-conditie	154
7.11	De cross-join	156
7.12	De union-join en de natural-join	157
7.13	Equi-joins en theta-joins	158
7.14	Join-condities vervangen door USING	158
7.15	De FROM-component met tabellexpressies	159
8	SELECT-instructie: De WHERE-component	167
8.1	Inleiding	167
8.2	Conditie met een vergelijkingsoperator	168
8.3	Vergelijkingsoperatoren met subquery's	174
8.4	Vergelijkingsoperatoren met gecorreleerde subquery's	178
8.5	Conditie koppelen met AND, OR en NOT	180
8.6	De IN-operator met expressielijst	183
8.7	De IN-operator met subquery	188
8.8	De BETWEEN-operator	197
8.9	De LIKE-operator	200
8.10	De REGEXP-operator	202
8.11	De MATCH-operator	210
8.12	De IS NULL-operator	220
8.13	De EXISTS-operator	222
8.14	De ALL- en ANY-operatoren	225
8.15	Reikwijdte van kolommen bij subquery's	233

8.16	Extra voorbeelden met gecorreleerde subquery's	236
8.17	Conditie met een ontkenning	242
9	SELECT-instructie: SELECT-component en aggregatiefuncties	245
9.1	Inleiding	245
9.2	Opvragen van alle kolommen (*)	245
9.3	Expressies in de SELECT-component	246
9.4	Verwijderen van identieke rijen met DISTINCT	247
9.5	Wanneer zijn twee rijen gelijk?	250
9.6	Een inleiding tot aggregatiefuncties	252
9.7	COUNT-functie	254
9.8	MAX- en MIN-functies	258
9.9	De SUM- en AVG-functie	263
9.10	De VARIANCE- en STDDEV-functies	266
9.11	De OLAP-functies	268
9.11.1	De ROW_NUMBER-, RANK- en DENSE_RANK-functies	269
9.11.2	Partitioneren van volgnummers	277
9.11.3	De OLAP-aggregatiefuncties	281
10	SELECT-instructie: de GROUP BY-component	287
10.1	Inleiding	287
10.2	Groeperen op één kolom	288
10.3	Groeperen op twee of meer kolommen	291
10.4	Groeperen op expressies	293
10.5	Groeperen van null-waarden	295
10.6	Algemene regels voor de GROUP BY-component	296
10.7	Complexe voorbeelden met GROUP BY	298
10.8	Groeperen met WITH ROLLUP	303
10.9	Groeperen met WITH CUBE	305
10.10	Grouping sets	306
10.11	Groeperen met ROLLUP en CUBE	310
10.12	Combineren van grouping sets	315
11	SELECT-instructie: de HAVING-component	317
11.1	Inleiding	317
11.2	Voorbeelden van de HAVING-component	318
11.3	Algemene regel voor de HAVING-component	320
12	SELECT-instructie: de ORDER BY-component	323
12.1	Inleiding	323
12.2	Sorteren op kolomnamen	323
12.3	Sorteren op expressies	325
12.4	Sorteren met behulp van volgnummers	327
12.5	Oplopend en aflopend sorteren	329
12.6	Sorteren van null-waarden	331

13	SELECT-instructie: de LIMIT-component	333
13.1	Inleiding	333
13.2	Geef de top ...	335
13.3	Subquery's met een LIMIT-component	338
13.4	Limit met een offset	341
14	Combineren van tabelexpressies	343
14.1	Inleiding	343
14.2	Koppelen met UNION	343
14.3	Regels voor het gebruik van UNION	346
14.4	Koppelen met INTERSECT	349
14.5	Koppelen met EXCEPT	353
14.6	Behouden van dubbele rijen met ALL	356
14.7	Set-operatoren en de null-waarde	357
14.8	Meerdere set-operatoren combineren	357
14.9	Set-operatoren en de theorie	359
15	SELECT-instructie: de WITH-component	361
15.1	Inleiding	361
15.2	Simpele voorbeelden van de WITH-component	362
15.3	Recursieve SELECT-instructies	367
15.4	Recursief doorlopen van een hiërarchische structuur	372
15.5	Het voorkomen van een eindeloze lus	378
15.6	Het recursief doorlopen van een netwerkstructuur	382
15.7	Het vluchtschema	385
16	Muteren van tabellen	389
16.1	Inleiding	389
16.2	Invoeren van nieuwe rijen	389
16.3	Een tabel vullen met rijen van een andere tabel	391
16.4	Wijzigen van waarden in rijen	393
16.5	Verwijderen van rijen uit een tabel	397
17	Werken met XML-documenten	399
17.1	XML in een notendop	399
17.2	Opslaan van XML-documenten	400
17.3	Zoeken van XML-documenten	403
17.4	Zoeken met behulp van posities	409
17.5	De uitgebreide notatie van XPath	411
17.6	XPath-expressies met condities	413
17.7	Wijzigen van XML-documenten	414
	Deel III Creëren van databaseobjecten	417
18	Creëren van tabellen	419
18.1	Inleiding	419
18.2	Opzetten van nieuwe tabellen	419

18.3	Datatypes van kolommen	422
18.3.1	De integer-datatypes	422
18.3.2	De decimal-datatypes	423
18.3.3	De float-datatypes	423
18.3.4	De alfanumerieke datatypes	425
18.3.5	De temporal datatypes	427
18.3.6	De blob-datatypes	427
18.4	Creëren van tijdelijke tabellen	428
18.5	Kopiëren van tabellen	430
18.6	Naamgeving van tabellen en kolommen	434
18.7	Kolom-opties: default en comment	435
18.8	Afgeleide kolommen	437
18.9	Tabellen en de catalogus	438
19	Integriteitsregels	441
19.1	Inleiding	441
19.2	Primaire sleutels	442
19.3	Alternatieve sleutels	445
19.4	Refererende sleutels	447
19.5	De refererende actie	452
19.6	Check-integriteitsregels	454
19.7	Namen toekennen aan integriteitsregels	456
19.8	Verwijderen van integriteitsregels	457
19.9	Integriteitsregels en de catalogus	457
20	Karaktersets en collations	459
20.1	Inleiding	459
20.2	Beschikbare karaktersets en collations	460
20.3	Toekennen van karaktersets aan kolommen	462
20.4	Toekennen van collations aan kolommen	464
20.5	Expressies met karaktersets en collations	465
20.6	Sorteren en groeperen met collations	468
20.7	De coercibility van expressies	469
20.8	Gerelateerde systeemvariabelen	471
20.9	Karaktersets en de catalogus	472
21	Veranderen en verwijderen van tabellen	473
21.1	Inleiding	473
21.2	Verwijderen van gehele tabellen	474
21.3	Hernoemen van tabellen	474
21.4	Wijzigen van tabeleigenschappen	475
21.5	Toevoegen van kolommen	476
21.6	Wijzigen van kolommen	477
21.7	Verwijderen van kolommen	479
21.8	Toevoegen van integriteitsregels	480
21.9	Verwijderen van integriteitsregels	481

22	Ontwerpen van tabellen	483
22.1	Inleiding	483
22.2	Welke tabellen en welke kolommen?	483
22.3	Toevoegen van redundante gegevens	488
22.4	Kiezen van een datatype voor een kolom	491
22.5	Wanneer NOT NULL?	492
23	Gebruik van indexen	493
23.1	Inleiding	493
23.2	Rijen, tabellen en bestanden	493
23.3	De werking van een index	495
23.4	De verwerkingsstappen van een SELECT-instructie	499
23.5	Creëren van indexen	503
23.6	Verwijderen van indexen	505
23.7	Indexen en primaire sleutels	506
23.8	De grote SPELERS_XXL-tabel	506
23.9	Op welke kolom moet een index staan?	508
23.9.1	Een unieke index op kandidaatsleutels	508
23.9.2	Een index op refererende sleutels	508
23.9.3	Een index op kolommen waarop geselecteerd wordt	509
23.9.4	Een index op een combinatie van kolommen	511
23.9.5	Een index op kolommen waarop gesorteerd wordt	512
23.10	Speciale indexvormen	513
23.10.1	De multi-tabelindex	513
23.10.2	De virtuele-kolomindex	513
23.10.3	De selectieve index	514
23.10.4	De hashindex	514
23.10.5	De bitmapindex	516
23.11	Indexen en de catalogus	517
24	Views	519
24.1	Inleiding	519
24.2	Creëren van views	519
24.3	De kolomnamen van views	522
24.4	Muteren van views, de WITH CHECK OPTION	523
24.5	Verwijderen van views	525
24.6	Views en de catalogus	525
24.7	Beperkingen van views bij het raadplegen	527
24.8	Beperkingen bij het muteren	528
24.9	Verwerken van instructies op views	529
24.10	Toepassingen van views	531
24.10.1	Vereenvoudigen van routinematige instructies	531
24.10.2	Reorganisatie van tabellen	532
24.10.3	Stapsgewijs opzetten van SELECT-instructies	534
24.10.4	Specificeren van integriteitsregels	535
24.10.5	Gegevensbeveiliging	535

25	Databases	537
25.1	Inleiding	537
25.2	Databases en de catalogus	537
25.3	Creëren van databases	538
25.4	Wijzigen van databases	539
25.5	Verwijderen van databases	540
26	Gebruikers en gegevensbeveiliging	541
26.1	Inleiding	541
26.2	Invoeren en verwijderen van gebruikers	542
26.3	Wijzigen van wachtwoorden	544
26.4	Verstrekken van tabel- en kolombevoegdheden	544
26.5	Verstrekken van databasebevoegdheden	547
26.6	Verstrekken van gebruikersbevoegdheden	549
26.7	Doorgeven van bevoegdheden met de WITH GRANT OPTION	551
26.8	Werken met rollen	552
26.9	Registreren van bevoegdheden in de catalogus	554
26.10	Intrekken van bevoegdheden	556
26.11	Beveiliging van en met views	559
27	Sequences	561
27.1	Inleiding	561
27.2	Waarom hebben we sequences nodig?	561
27.3	Opties van de sequences	564
27.4	De sequence-expressie in detail	568
27.5	Opvragen van het laatst gegenereerde nummer	570
27.6	Wijzigen en verwijderen van sequences	572
27.7	Bevoegdheden voor sequences	573
28	Schema's	575
28.1	Wat is een schema?	575
28.2	Creëren van een schema	576
28.3	Verwijderen van een schema	577
28.4	Schema versus SQL-gebruiker	577
	Deel IV Procedurele databaseobjecten	579
29	Stored procedures	581
29.1	Inleiding	581
29.2	Een voorbeeld van een stored procedure	581
29.3	De parameters van een stored procedure	584
29.4	De body van een stored procedure	585
29.5	Lokale variabelen	587
29.6	De SET-instructie	589
29.7	Flow-control-instructies	589
29.8	Aanroepen van stored procedures	596
29.9	Opvragen van gegevens met SELECT INTO	599
29.10	Foutboodschappen, handlers en conditions	603

29.11	Gegevens ophalen met een cursor	607
29.12	Stored procedures en de catalogus	612
29.13	Verwijderen van stored procedures	613
29.14	Compileren en hercompileren	614
29.15	Beveiliging met stored procedures	615
29.16	Voordelen van stored procedures	616
30	Stored functions	617
30.1	Inleiding	617
30.2	Voorbeelden van stored functions	619
30.3	Verwijderen van stored functions	623
31	Triggers	625
31.1	Inleiding	625
31.2	Een voorbeeld van een trigger	626
31.3	Complexere voorbeelden	630
31.4	Triggers als integriteitsregels	636
31.5	Verwijderen van triggers	638
31.6	Verschillen tussen SQL-producten	639
	Deel V Programmeren met SQL	641
32	Inleiding embedded SQL	643
32.1	Inleiding	643
32.2	De pseudo programmeertaal	643
32.3	DDL- en DCL-instructies en embedded SQL	643
32.4	Verwerken van programma's	645
32.5	Gebruik van hostvariabelen in SQL-instructies	648
32.6	De SQLCODE-hostvariabele	649
32.7	Executeerbare en niet-executeerbare SQL-instructies	651
32.8	De WHENEVER-instructie	651
32.9	Inloggen op SQL	654
32.10	SELECT-instructies met één rij	655
32.11	Null-waarden en de null-indicator	658
32.12	Cursors voor het raadplegen van veel rijen	659
32.13	De richting van het doorlopen van een cursor	664
32.14	Verwerken van cursors	665
32.15	De FOR-component	667
32.16	Verwijderen van rijen via cursors	669
32.17	Voorbeeld van een C-programma	670
33	Transacties en multi-usergebruik	673
33.1	Inleiding	673
33.2	Wat is een transactie?	673
33.3	Starten van transacties	678
33.4	Embedded SQL en transacties	679
33.5	Savepoints	680
33.6	Stored procedures en transacties	681

33.7	Problemen met multi-usergebruik	682
33.7.1	Dirty read ofwel uncommitted read	682
33.7.2	Nonrepeatable read ofwel nonreproducible read	683
33.7.3	Phantom read	684
33.7.4	Lost update	684
33.8	Locking	685
33.9	Deadlocks	686
33.10	Granulariteit van het blokkeren	687
33.11	De LOCK TABLE- en UNLOCK TABLE-instructies	687
33.12	Het isolation level	688
34	Optimaliseren van instructies	691
34.1	Inleiding	691
34.2	Vermijd de OR-operator	691
34.3	Vermijd onnodig gebruik van de UNION-operator	693
34.4	Vermijd de NOT-operator	694
34.5	Isoleer kolommen in condities	694
34.6	Gebruik de BETWEEN-operator	695
34.7	Vermijd bepaalde vormen van de LIKE-operator	695
34.8	Voeg bij joins redundante condities toe	696
34.9	Vermijd de HAVING-component	696
34.10	Maak de SELECT-component zo smal mogelijk	697
34.11	Vermijd DISTINCT	697
34.12	Gebruik de ALL-optie bij set-operatoren	697
34.13	Geef de voorkeur aan outer-joins boven UNION-operatoren	698
34.14	Vermijd datatype-conversies	699
34.15	De grootste tabel als laatste	699
34.16	Vermijd de ANY- en ALL-operatoren	699
34.17	De toekomst van de optimiser	701
35	De toekomst van SQL	703
	Bijlage A Referenties	705
	Register	707

Over de auteur

Rick F. van der Lans is auteur van verscheidene boeken over SQL. Naast dit SQL Leerboek dat in diverse talen vertaald is, waaronder Engels, Duits, Chinees en Italiaans, heeft hij SQL-boeken geschreven voor producten als MySQL, Oracle, SQLite, Ingres en Pervasive PSQL.



Hij is onafhankelijk adviseur, auteur en docent gespecialiseerd in databasetechnologie, datawarehousing en applicatie-integratie. Hij is oprichter en directeur van R20/Consultancy. Door de jaren heen heeft hij veel organisaties geadviseerd op het gebied van IT-architecturen.

Als spreker op conferenties en seminars wordt hij internationaal gerespecteerd. Al meer dan vijftientig jaar geeft hij over de gehele wereld lezingen, waaronder in de meeste Europese landen, Noord- en Zuid-Amerika en Australië. Hij is voorzitter van de jaarlijkse European Data Warehouse and Business Intelligence Conference. Hij schrijft een column voor Database Magazine en voor het vooraanstaande Beye-Network.com. Zeven jaar lang was hij lid van de Nederlandse ISO-commissie verantwoordelijk voor ISO SQL-standaard.

Rick kan via de volgende kanalen bereikt worden:

E-mail:	rick@r20.nl
Twitter:	http://twitter.com/Rick__vanderlans
LinkedIn:	http://www.linkedin.com/pub/rick-van-der-lans/9/207/223

Voorwoord

Inleiding

SQL was, is en blijft voorlopig de databasetaal van relationele databaseservers zoals IBM DB2, Ingres, Microsoft SQL Server, MySQL, Oracle, Pervasive PSQL, SQLite, Sybase Adaptive Server en nog tientallen andere. De taal is ontworpen voor het benaderen en bewerken van databasegegevens. Dit boek bevat een volledige en grondige beschrijving van SQL (Structured Query Language). Het is opgezet als een leerboek, een soort doe-boek. Na het lezen van dit boek kent u alle instructies, mogelijkheden en eigenaardigheden van SQL en bent u in staat haar effectief en efficiënt toe te passen. De afkorting SQL wordt door sommigen uitgesproken als EsQueEl en door anderen als SieKwol.

SQL kent een kleine maar zeer krachtige verzameling instructies gericht op het bewerken, beheren en beveiligen van databasegegevens. Deze kracht heeft geleid tot een enorme populariteit. Bestonden er aan het begin van de jaren tachtig nog een twintigtal SQL-databaseservers, nu is dit aantal minimaal verviervoudigd. Bijna elke databaseserver ondersteunt SQL of een dialect van deze taal. SQL-producten zijn momenteel verkrijgbaar voor elk soort computer, of dat nu een kleine mobiele telefoon, een pc of een grote server is, en voor elk besturingssysteem, inclusief Microsoft Windows en veel UNIX-varianten. Al sinds 1987 bestaat er een officiële, internationale standaard voor SQL. SQL is duidelijk uitgegroeid tot, zoals Michael Stonebraker, een autoriteit op het gebied van databases, het ooit verwoordde: *intergalactic dataspeak*.

Onderwerpen

Dit boek is volledig gewijd aan SQL. Alle aspecten van de taal worden grondig en kritisch besproken, waaronder:

- raadplegen van gegevens (joins, functies en subquery's)
- muteren van gegevens
- opzetten van tabellen en views
- specificeren van primaire, refererende sleutels en andere integriteitsregels
- gebruik van indexen
- gegevensbeveiliging
- ontwikkelen van stored procedures en triggers
- programmaontwikkeling met embedded SQL
- werken met transacties
- optimaliseren van instructies
- de catalogus

Welk SQL-dialect?

Er zijn tegenwoordig veel SQL-producten beschikbaar. Al deze implementaties van SQL lijken veel op elkaar, maar vertonen helaas toch verschillen. Sommige ondersteunen niet alle SQL-instructies en andere bieden niet al de mogelijkheden van een specifieke SQL-instructie. In sommige gevallen kunnen identieke instructies bij verschillende producten zelfs uiteenlopende resultaten opleveren.

De vraag is dan: welk SQL-dialect wordt er in dit boek beschreven? Om het boek zo praktisch mogelijk te maken, worden de SQL-instructies en -mogelijkheden beschreven die

door de meeste dominante SQL-producten ondersteund worden. Dit verhoogt de praktische waarde van dit boek. Na het lezen van dit boek moet u in staat zijn om met elk willekeurig SQL-product aan de slag te kunnen. Met andere woorden, dit boek is dus gericht op algemeen SQL: SQL zoals geïmplementeerd in de meeste SQL-producten, en niet zozeer op DB2, Oracle of MySQL en zelfs niet op de internationale standaarden voor SQL.

Voor wie is dit boek bestemd?

Het SQL Leerboek is bestemd voor iedereen die de volledige kracht van SQL op een effectieve en efficiënte manier in de praktijk wil toepassen en het is daarom zeer geschikt voor de volgende groepen personen:

- **Studenten** in het hoger onderwijs, zowel in het hoger beroepsonderwijs als aan universiteiten en hogescholen.
- **Ontwikkelaars** die programma's ontwikkelen of gaan ontwikkelen met behulp van een SQL-product.
- **Ontwerpers, analisten en adviseurs** die direct of indirect met SQL of een andere relationele databasetaal te maken krijgen en willen weten wat de mogelijkheden en onmogelijkheden zijn.
- **Thuisstuderenden** die geïnteresseerd zijn in SQL in het bijzonder of in relationele databasetalen in het algemeen.
- **Gebruikers** die de bevoegdheid hebben de database van het bedrijf of de instelling waar ze werken via SQL te raadplegen.

Een praktisch boek

Het boek moet voornamelijk gezien worden als een *leerboek* in de zin van een doe-boek, en niet zozeer als een naslagwerk. Het bevat daarom veel voorbeelden en opgaven (met antwoorden). Slaat u de opgaven niet over, want uit ervaring is gebleken dat u de taal het best en het snelst leert door veel te oefenen, en dus door veel opgaven te maken.

Met opzet zijn in het boek veel Engelse termen niet vertaald. Het merendeel van de handleidingen van de diverse SQL-producten is namelijk alleen in het Engels beschikbaar. Vertalen van al deze termen in dit boek zou de handleidingen ontoegankelijker maken. Als een Engelse term echter voor de eerste keer genoemd wordt, is de Nederlandse vertaling erbij vermeld.

Oefenen met MySQL

Een van de bekendste SQL-databaseservers is een product genaamd *MySQL*. We gaan er in dit boek van uit dat u MySQL gebruikt als u de voorbeelden naspeelt en de opgaven wilt uitwerken. Er is gekozen voor MySQL vanwege de populariteit van het product en omdat het SQL-dialect van MySQL qua functionaliteit zeer uitgebreid is en veel overeenkomsten vertoont met de internationale standaard voor SQL.

Ik raad u uiteraard aan MySQL te installeren en zoveel mogelijk oefeningen met behulp van MySQL na te spelen. Het uitvoeren van SQL-instructies en het bekijken van de resultaten, blijft nog steeds een ideale manier om deze krachtige taal onder de knie te krijgen.

Waar MySQL gedownload kan worden en hoe het geïnstalleerd moet worden, staat beschreven op de website die voor dit boek opgezet is (zie de volgende sectie).

De website behorende bij het boek

Op de website www.r20.nl zijn allerlei zaken te vinden die gerelateerd zijn aan dit boek. Ten eerste staan er de volgende documenten die alle gratis gedownload kunnen worden:

- **Installatiegids:** Dit document bevat beschrijvingen die helpen bij het installeren van de benodigde software en voor het opbouwen van de voorbeelddatabase.
- **Syntaxis van SQL:** Dit document bevat definities van alle SQL-instructies zoals beschreven in dit boek. In voorgaande edities was dit een bijlage in het boek, maar door het als PDF-document beschikbaar te stellen, is het eenvoudiger om bepaalde definities te zoeken. Dit is gedaan naar aanleiding van vragen van lezers.
- **Scalaire functies:** Dit document bevat beschrijvingen van alle scalaire functies. Ook dit document was bij vorige edities een bijlage, maar om dezelfde reden als voor de syntaxis is besloten deze als PDF-document beschikbaar te maken waardoor de beschrijvingen van functies eenvoudiger en sneller gevonden kunnen worden.
- **Objectrelationele concepten in SQL:** Dit document bevat twee hoofdstukken uit de vorige editie. Omdat dit deel van SQL uiteindelijk niet zo populair is geworden, is besloten deze uit het boek te verwijderen en daarmee ruimte te maken voor beschrijvingen van nieuwe, succesvollere onderdelen van SQL.
- **Introductie tot de verzamelingenleer en de logica:** SQL is gebaseerd op het relationele model dat op zich weer gebaseerd is op de verzamelingenleer en logica. Voor diegenen die niet zo bekend zijn met deze materie, is een document op de website geplaatst waar kort en bondig de basis van beide uitgelegd wordt.
- **De relationele algebra en calculus:** Beide zijn onderdelen van het relationele model en geven aan wat de operatoren van een relationele databasetaal zijn. In feite hebben de relationele algebra en calculus model gestaan voor SQL. Dit hoofdstuk is toegevoegd om te kunnen aangeven waarom SQL bepaalde operatoren ondersteunt.

Ten tweede, als u door dit boek bladert, zult u zien dat het vol staat met SQL-instructies. Soms zijn het voorbeelden, soms antwoorden op opgaven. Als u MySQL geïnstalleerd hebt, kunt u deze instructies naspelen om te kijken of ze werken en wat precies het effect is. Hiervoor kunt u op spartaanse wijze alle instructies opnieuw intikken, maar u kunt het uzelf ook gemakkelijk maken door alle instructies vanaf de website te downloaden.

Uiteraard zal de website ook voor andere zaken gebruikt worden:

- Als een storende fout in het boek gevonden wordt, zal de correctie op de website vermeld worden.
- Opmerkingen van lezers die voor anderen interessant zouden kunnen zijn, worden op de website geplaatst.

Houd deze website dus in de gaten.

Vereiste voorkennis

Enige algemene kennis van programmeertalen en databaseservers wordt verondersteld.

De historie van dit boek

Het was 1984 en de databasewereld was in de ban van een revolutie. SQL was aan haar triomftocht begonnen. Leveranciers als IBM en Oracle hadden de commerciële versies van hun SQL-databaseservers beschikbaar en hun marketingmachines draaiden op volle toeren. De markt reageerde positief op deze overname door de eerste generatie SQL-databaseservers. Veel organisaties besloten om een dergelijke databaseserver aan te schaffen en hun bestaande producten uit te faseren.

Mijn toenmalige werkgever had besloten zich ook in het strijdgewoel te mengen. Ook hij wilde geld verdienen aan deze nieuwe databasetaal en besloot SQL-cursussen te organiseren. Omwille van mijn achtergrondkennis kreeg ik deze taak toegewezen. Dat SQL zo succesvol zou worden en dat mijn instemming om de cursussen te gaan geven voor mijzelf zoveel persoonlijke en zakelijke consequenties zou hebben, had ik nooit kunnen vermoeden.

Na SQL aandachtig bestudeerd te hebben, kon ik beginnen met het ontwikkelen van het cursusmateriaal. Na de cursus twee jaar met plezier gedoceerd te hebben, ontstond bij mij het idee een boek over deze databasetaal te schrijven. Het zou een boek moeten worden dat geheel gewijd zou zijn aan deze taal met haar vele mogelijkheden en eigenaardigheden.

Na veel ploeteren verscheen in 1986 de eerste Nederlandstalige editie met de titel *Het SQL Leerboek*. Dit boek was amper klaar of ik werd gevraagd een Engelstalige editie te schrijven. Deze verscheen in 1987 en was het eerste boek in die taal dat volledig aan SQL gewijd was. Hierna volgden nog een Duitse en een Italiaanse editie. Er was duidelijk behoefte aan informatie over SQL. Iedereen wilde met SQL beginnen, maar de informatie was schaars.

Omdat SQL nog jong was, ontwikkelde zij zich snel. Instructies werden toegevoegd, uitgebreid en verbeterd. Nieuwe implementaties kwamen op de markt, nieuwe toepassingsgebieden werden gevonden en er verschenen nieuwe versies van de SQL-standaard. Dus al snel moest er een nieuwe editie van het boek geschreven worden. En er volgden er nog meer. Het boek dat u in handen hebt, is alweer de zevende editie van de Nederlandstalige versie. En het zal waarschijnlijk niet de laatste editie zijn, want SQL heeft de revolutie in de databasewereld glorieus gewonnen en aan de horizon is geen competitie in zicht.

Voorwoord bij de zevende editie, ofwel het vijftienvigjarig jubileum

Deze zevende editie van *Het SQL Leerboek* is een herziene editie. De volgende vier geheel nieuwe hoofdstukken zijn toegevoegd:

- Opslaan, zoeken en bewerken van XML-documenten
- Zoeken in tekst
- Werken met recursieve SQL-instructies
- Definiëren van zogenaamde sequences om unieke nummers te genereren

Enkele hoofdstukken zijn enigszins bijgewerkt, omdat SQL zelf blijft veranderen. Drie hoofdstukken en twee bijlagen zijn uit het boek gehaald en op de website geplaatst. De website bevat ook enkele documenten met nieuw materiaal.

Toen ik in 1985 begon met het schrijven van dit boek, dat vijftienvig jaar geleden verscheen, heb ik mij niet gerealiseerd dat dit boek zo'n lang leven zou hebben. Mijn doelstelling was toen een helder boek te schrijven waarin SQL gedoceerd werd. Ik wist dat er behoefte was aan een dergelijk boek, maar dat het vervolgens mijn gehele IT-carrière zou meegaan, dat had ik absoluut niet verwacht.

In de periode dat ik de eerste editie aan het schrijven was, bestond het internet nog niet (laat staan social media networks), en Java en C# waren nog niet eens uitgevonden. Voor sommigen is het misschien ondenkbaar, maar ook de mobiele telefoon bestond nog niet en de iPad en eReader al helemaal niet. Wat is er eigenlijk in die vijftienvig jaar veel veranderd. Veel nieuwe technologieën zijn beschikbaar gekomen en er zijn er ook veel verdwenen. SQL daarentegen staat nog steeds als een rots in de branding. De meeste orga-

nisaties zouden ook niet meer zonder SQL kunnen, want ze hebben veel van hun belangrijke gegevens in SQL-databases opgeslagen.

Maar ik realiseerde mij dat allemaal niet in de jaren tachtig. De vraag die ik mij wel steeds vaker stel, is of SQL er over tien jaar nog wel zal zijn, of over twintig jaar? Gaat deze taal mijzelf overleven? We gaan het allemaal de komende jaren meemaken. En ik ben reuze benieuwd of er over enkele jaren weer een volgende editie van dit boek noodzakelijk is.

En ten slotte ...

Het schrijven van dit boek was geen soloproject. Veel mensen hebben bijgedragen aan deze of vorige edities van dit boek. Daarom wil ik dit voorwoord gebruiken om hen te bedanken voor hun hulp, bijdragen, ideeën, inhoudelijk commentaar, mentale steun en geduld.

Hoe vaak een schrijver zijn eigen werk ook naleest, redacteuren blijven onontbeerlijk. Een schrijver leest niet wat hij heeft geschreven, maar leest wat hij dacht dat hij zou schrijven. Wat dat betreft lijkt schrijven op programmeren. Ik ben daarom de volgende personen zeer dankbaar voor het plaatsen van kritische opmerkingen en het geven van zeer nuttige adviezen: Klaas Brant, Marc van Cappellen, Ian Cargill, Corine Cools, Richard van Dijk, Rose Endres, Andrea Gray, Ed Jedeloo, Josien van der Laan, Oda van der Lans, Deborah Leendertse, Arjen Lentz, Onno de Maar, Andrea Maurino, Sandor Nieuwenhuijs, Henk Schreij, Dave Slayton, Aad Speksnijder, Nok van Veen, John Vicherek en David van der Waaij. Ze hebben allemaal dit manuscript gelezen, of een deel ervan, of het manuscript van een vorige editie, een vertaling ervan of een aangepaste versie.

Speciaal zou ik Wim Frederiks en Roland Bouman willen bedanken voor het minutieus doornemen van dit boek. Beide hebben met engelengeduld elke bladzijde gelezen en mij gewezen op fouten en inconsistenties. Ik ben ze daar zeer dankbaar voor.

Tevens wil ik de duizenden cursisten over de hele wereld bedanken die ik in al die jaren SQL heb gedoceerd. Hun opmerkingen en aanbevelingen zijn voor dit boek van onschatbare waarde geweest. Een groot aantal lezers van de vorige edities van mijn boek heeft gehoor gegeven aan mijn oproep om commentaar en suggesties toe te sturen. Ook hen wil ik hartelijk danken voor de moeite die zij genomen hebben dit allemaal duidelijk op papier te zetten.

Ongeacht de toewijding en inzet van de eerder genoemde personen, en ongeacht hoe groot hun bijdrage was, één persoon was onmisbaar voor dit project: mijn echtgenote en 'persoonlijk vertaler', Diane Cools. We hebben nu al aan meer dan tien boeken samengewerkt. Zij was het die bij de eerste edities zeer veel tik- en correctiewerk verzorgd heeft. Hier ben ik haar nog steeds dankbaar voor, want werken met Wordstar versie 1 op een PC/XT zonder harde schijf leek toen een luxe. En na al die jaren is het nog steeds geweldig om met haar samen te werken. Zoals altijd, Diane, bedankt!

Zoals ik al jaren gewend ben, heb ik weer veel hulp gekregen van mijn uitgever, van wie het commentaar en de ideeën weer uitermate nuttig waren. Het is altijd een genot om met professionals te werken.

Ten slotte wil ik de lezers vragen commentaar, meningen, ideeën en suggesties betreffende de inhoud van dit boek op te sturen naar sql@r20.nl onder vermelding van 'Het SQL Leerboek.' Bij voorbaat hartelijk dank voor uw medewerking.

Rick F. van der Lans

Den Haag, november 2011

Deel I

Inleiding

SQL is een compacte en krachtige taal voor het werken met databases. Ondanks die compactheid is ze niet eenvoudig in enkele hoofdstukken te beschrijven; daarmee zouden we de taal onrecht doen. De leveranciers van SQL-producten hebben het nog eens ingewikkelder gemaakt door niet allemaal hetzelfde SQL-dialect te implementeren. Om de uitleg gestructureerd aan te pakken, beginnen we dit boek met een aantal inleidende hoofdstukken die samen het eerste deel vormen.

Hoofdstuk 1 beschrijft de essentie, achtergrond, historie en toepassingsgebieden van SQL. Tevens wordt een aantal concepten uit het relationele model (de theorie achter SQL) toegelicht.

- Dit boek staat vol met voorbeelden en oefeningen. Om te voorkomen dat u zich bij elk voorbeeld opnieuw in een andere situatie moet verplaatsen, is besloten om bij de meeste dezelfde database te gebruiken. Deze database vormt de basis voor een administratie van een tennisvereniging. Hoofdstuk 2 beschrijft de structuur van deze database. Bestudeer deze aandachtig voordat u opgaven maakt.
- Het is, zoals reeds vermeld, sterk aan te raden dat u met MySQL de opgaven uitwerkt. Hiertoe moet u wel de software en de voorbeelddatabase installeren. De instructies hiervoor staan op de website; zie ook hoofdstuk 3
- Dit deel eindigt met hoofdstuk 4 waarin alle belangrijke SQL-instructies de revue passeren. Na het lezen hiervan hebt u een globaal beeld van wat SQL als taal te bieden heeft en hebt u een gevoel van wat er in dit boek behandeld wordt.

Hoofdstuk 1

Inleiding SQL

1.1 Inleiding

Dit hoofdstuk beschrijft globaal wat SQL is, de achtergrond en historie van SQL en haar toepassingsgebieden. Tevens worden fundamentele onderwerpen als database en databaseserver beschreven. SQL is gebaseerd op theorieën van het *relationele model*. Als u SQL wilt leren gebruiken, is enige kennis over dit model onontbeerlijk. Daarom wordt in paragraaf 1.3 het relationele model beschreven. In paragraaf 1.4 wordt in het kort beschreven wat SQL is, wat men met de taal kan doen en waarin de taal verschilt van andere talen (zoals Java, C#, Visual Basic of Python). Paragraaf 1.6 beschrijft de geschiedenis van SQL. Al is SQL een moderne taal, ze kent toch al een lange geschiedenis die rond 1972 begonnen is. Momenteel is ze in veel producten geïmplementeerd en bezit een monopoliepositie in de wereld van databasetalen. Paragraaf 1.9 beschrijft de belangrijkste, actuele standaarden voor SQL. In paragraaf 1.10 geven we een kort overzicht van de belangrijkste SQL-producten.

We sluiten dit eerste hoofdstuk af met een beschrijving van de structuur van het boek. Elk hoofdstuk wordt in een paar zinnen samengevat.

1.2 Database, databaseserver en databasetaal

SQL (Structured Query Language) is een databasetaal voor het formuleren van opdrachten die door een databaseserver verwerkt worden. Deze zin bevat drie belangrijke begrippen: *database*, *databaseserver* en *databasetaal*. We beginnen met een toelichting van deze termen.

Wat is een *database*? Wij hanteren in dit boek een definitie afgeleid van die van Chris J. Date uit [DATE03]:

A database consists of some collection of persistent data, that is used by the applications systems of some given enterprise, and that is managed by a database management system.

Kaartenbakbestanden worden dus niet gezien als databases. Maar wel bijvoorbeeld de grote, geautomatiseerde bestanden van banken, verzekeringsmaatschappijen, telefoonmaatschappijen en organisaties als de Rijksdienst voor het Wegverkeer. Deze databases bevatten gegevens over adressen, saldo's, kentekennummers, gewicht van voertuigen, enzovoorts. Ongetwijfeld beschikt het bedrijf waarvoor u werkt over een eigen computer en worden daarmee gegevens over bijvoorbeeld uw salaris geregistreerd.

Gegevens in een database worden pas waardevol wanneer er iets mee gedaan wordt. Volgens de definitie worden gegevens in de database beheerd door een afzonderlijk programmatuursysteem. Dit systeem wordt een *databaseserver* ofwel *database management systeem* (*dbms*) genoemd. Met behulp van een databaseserver kunnen databasegegevens verwerkt worden. MySQL is zo'n databaseserver. Zonder databaseserver is het onmogelijk gegevens in de database te bekijken, te wijzigen of verouderde gegevens te verwijderen. Alleen de databaseserver weet waar en hoe de gegevens opgeslagen zijn. In [ELMA06] wordt door R. Elmasri het begrip databaseserver als volgt gedefinieerd:

A databaseserver is a collection of programs that enables users to create and maintain a database.

Een databaseserver zal nooit uit zichzelf de gegevens in een database wijzigen of verwijderen; iemand zal de databaseserver daartoe de opdracht moeten geven. Voorbeelden van opdrachten die een gebruiker aan de databaseserver zou kunnen geven zijn: 'verwijder alle gegevens over het voertuig met kentekennummer 'DR-12-DP', of 'geef de namen van alle bedrijven die facturen van afgelopen maart nog niet betaald hebben.' Gebruikers communiceren echter niet direct met een databaseserver. Er is altijd een programma nodig waarmee gebruikers aan de databaseserver opdrachten kunnen geven. Tussen de gebruiker en de databaseserver bevindt zich dus altijd een programma. Hierover meer in paragraaf 1.4.

De definitie van de term database bevat tevens het woord *persistent*. Hiermee wordt bedoeld dat gegevens in een database permanent opgeslagen liggen totdat ze expliciet veranderd of verwijderd worden. Als u nieuwe gegevens in een database opslaat en de databaseserver heeft gemeld dat dit gelukt is, kunt u ervan op aan dat de gegevens er morgen nog zijn (ook al wordt de machine uitgezet). Dit in tegenstelling tot gegevens die we opslaan in het interne geheugen van een computer. Als de computer wordt uitgezet, zijn die gegevens voor altijd verdwenen. Zij zijn dus niet persistent.

Opdrachten aan een databaseserver worden geformuleerd in speciale talen. Dergelijke talen worden *databasetalen* genoemd. Opdrachten, ook wel instructies genoemd, die volgens de regels van de databasetaal geformuleerd worden, worden door gebruikers met behulp van speciale programma's ingevoerd en door de databaseserver verwerkt. Elke databaseserver, van welke leverancier dan ook, bezit een databasetaal. Sommige systemen hebben zelfs meer dan één databasetaal. Tussen al die talen bestaan verschillen waardoor ze in groepen te verdelen zijn. Een van deze groepen wordt gevormd door de *relationele databasetalen*. SQL is zo'n relationele databasetaal.

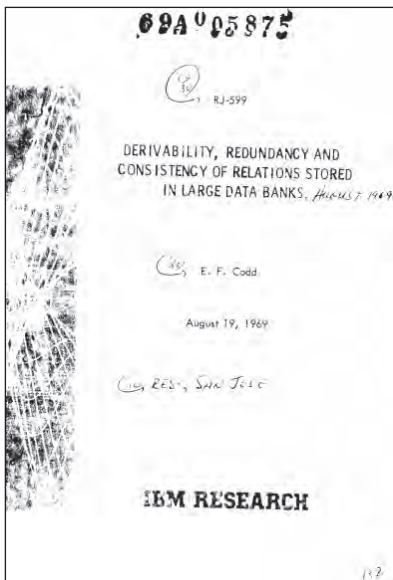
Hoe slaat een databaseserver de gegevens op? Een databaseserver gebruikt geen ladekast of hangmappen voor het registreren van gegevens, maar opslagmedia die door computers te benaderen zijn, zoals magnetische en optische schijven, tapes en solid state disks. De wijze waarop een databaseserver gegevens op deze media opslaat, is zeer complex en technisch. Wij zullen u daar in dit boek niet mee lastig vallen. Dit hoeft ook eigenlijk niet, want een van de belangrijkste taken van een databaseserver is het bieden van *gegevensonafhankelijkheid*. Hiermee wordt bedoeld dat gebruikers niet hoeven te weten hoe en waar gegevens opgeslagen worden. Voor hen is een database één groot reservoir waarin alle gegevens opgeslagen zijn. De wijze van opslag is volledig onafhankelijk van de databasetaal die gebruikt wordt. Dit alles is enigszins te vergelijken met het inleveren van bagage bij het inchecken van een vlucht. Wij bemoeien ons niet met waar en hoe de vliegtuigmaatschappij onze koffers opslaat. Het enige waarin we geïnteresseerd zijn, is dat we ze bij aankomst weer kunnen ophalen.

Een andere belangrijke taak van een databaseserver is het handhaven van de *integriteit* van de databasegegevens. Hiermee wordt ten eerste bedoeld dat de databaseserver moet zorgen dat databasegegevens altijd voldoen aan de regels die in de werkelijkheid gelden. Als bijvoorbeeld een werknemer maar voor één afdeling tegelijkertijd mag werken, mag nergens in de database geregistreerd kunnen worden dat een bepaalde werknemer voor twee of meer afdelingen werkt. Ten tweede wordt met integriteit bedoeld dat twee verschillende databasegegevens elkaar niet tegenspreken (dit wordt ook wel *consistentie* van

gegevens genoemd). Ergens staat bijvoorbeeld geregistreerd dat de heer Jansen geboren is op 4 augustus 1964 en op een andere plaats wordt aangegeven dat de geboortedatum van de heer Jansen 14 december 1946 is. Deze twee gegevens zijn inconsistent. Elke database-server kent instructies waarmee *integriteitsregels* gespecificeerd kunnen worden. Eenmaal ingevoerd, zal de databaseserver zorgen voor het handhaven van deze regels.

1.3 Het relationele model

SQL is gebaseerd op een formele en wiskundige theorie. Deze theorie, die uit een aantal begrippen en definities bestaat, heet het *relationele model*. Het relationele model is in 1970 geïntroduceerd door E. F. Codd, toen nog in dienst van IBM. Hij deed dit in de haast legendarische artikelen: 'Derivability, Redundancy and Consistency of Relations Stored in Large Data Banks' (zie [Codd69] en figuur 1.1) en 'A Relational Model of Data for Large Shared Data Banks' (zie [Codd70]). Dit relationele model vormt een theoretische basis voor databasetalen. Het model bestaat uit een klein aantal eenvoudige concepten voor het registreren van gegevens in een database, samen met een aantal operatoren om de gegevens te bewerken. Al deze concepten en operatoren zijn voornamelijk ontleend aan de *verzamelingenleer* en de *predikatenlogica*. Later, in 1979, heeft Codd ideeën gepresenteerd voor een verbeterde versie van het model; zie [Codd79] en [Codd90].



Figuur 1.1 Een afdruk van de voorkant van Codd's onderzoeksartikel waarin hij het relationele model introduceerde

Het relationele model heeft als voorbeeld gediend bij de ontwikkeling van diverse databasetalen, waaronder QUEL (zie [Ston86]), SQUARE (zie [Boyc73a]) en, natuurlijk, SQL. Deze databasetalen zijn gebaseerd op de concepten en ideeën van dat relationele model en worden daarom ook wel *relationele databasetalen* genoemd. SQL is daar een duidelijk voorbeeld van. In de rest van deze paragraaf beschrijven we globaal de volgende termen uit het relationele model, die we veelvuldig in het boek zullen gebruiken. Bij elke term staat de Engelse benaming tussen haakjes.

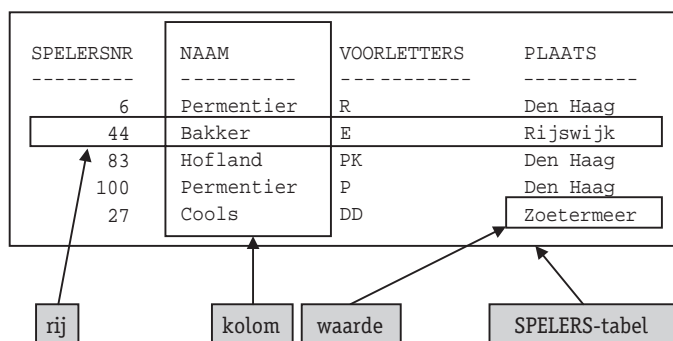
- tabel (table)
- kolom (column)
- rij (row)
- null-waarde
- integriteitsregel (integrity rule of constraint)
- primaire sleutel (primary key)
- kandidaatsleutel (candidate key)
- alternatieve sleutel (alternate key)
- refererende sleutel (foreign key of referential key)

Let wel, dit is slechts een deel van de termen uit het relationele model. In deel III worden de meeste gedetailleerd behandeld. Voor meer uitvoerige beschrijvingen verwijzen we naar [COD90] en [DATE03].

1.3.1 Tabel, kolom en rij

Gegevens in een relationele database kunnen we maar op één manier opslaan: in *tabellen*. De officiële naam voor tabel is eigenlijk *relatie* (relation). Van deze naam stamt ook de term het *relationele* model af. We hebben echter voor de term tabel gekozen, omdat dit de gangbare term binnen SQL is.

Informeel gesteld is een tabel een verzameling *rijen*, waarbij elke rij weer uit een verzameling *waarden* bestaat. Alle rijen in een bepaalde tabel hebben hetzelfde aantal waarden. In figuur 1.2 staat een voorbeeld van een tabel genaamd de SPELERS-tabel. Deze tabel bevat gegevens over vijf spelers die allen leden van een tennisvereniging zijn.



Figuur 1.2 De concepten waarde, rij, kolom en tabel

Deze SPELERS-tabel bevat vijf *rijen*, voor elke speler een. Een rij met waarden kunnen we beschouwen als een verzameling bij elkaar horende gegevens. Bijvoorbeeld, in deze tabel bevat de eerste rij de waarden 6, Permentier, R en Den Haag. Deze gegevens vertellen ons dat er een speler bestaat met nummer 6, dat hij als achternaam Permentier en als voorletter R heeft en dat hij woonachtig is in de plaats Den Haag.

SPELERSNR, NAAM, VOORLETTERS en PLAATS zijn de namen van de *kolommen* in de tabel. De SPELERSNR-kolom bevat de waarden 6, 44, 83, 100 en 27. Deze verzameling waarden wordt ook wel de *populatie* van de SPELERSNR-kolom genoemd. Elke rij heeft een waarde voor elke kolom. Dus, in de eerste rij is er een waarde voor de SPELERSNR-kolom en een waarde voor de NAAM-kolom, enzovoorts.

Een tabel heeft twee speciale eigenschappen:

- De kruising van een rij en een kolom kan maar één waarde bevatten, één *atomaire waarde*. Een *atomaire waarde* is een ondeelbaar geheel. De databaseserver kan een dergelijke waarde alleen in haar totaliteit behandelen.
- De rijen in een tabel hebben geen specifieke volgorde. Men kan dus niet spreken van de eerste rij, de laatste drie of de volgende rij. De inhoud van een tabel moet werkelijk als een *verzameling* rijen in de ware zin van het woord beschouwd worden.

1.3.2 Null-waarde

Kolommen worden gevuld met *atomaire waarden*. Een dergelijke waarde is bijvoorbeeld een getal, een woord of een datum. Een speciale waarde is de *null-waarde*. De *null-waarde* is te vergelijken met ‘waarde onbekend’ of ‘waarde niet aanwezig.’ Neem figuur 1.2 nogmaals als voorbeeld. Als we de woonplaats van speler 27 niet zouden weten, dan zouden we de PLAATS-kolom in de rij behorende bij speler 27 met een *null-waarde* kunnen vullen.

Een *null-waarde* is niet hetzelfde als het getal nul of een rij spaties. Het moet geïnterpreteerd worden als een waarde die ontbreekt. Een *null-waarde* heeft de eigenschap dat zij nooit gelijk is aan een andere *null-waarde*. Twee *null-waarden* zijn dus niet gelijk aan elkaar, maar ook niet ongelijk. Als we namelijk wel zouden weten dat twee *null-waarden* gelijk of ongelijk zijn, zouden we ‘iets’ van die *null-waarden* afweten. We zouden dan niet meer kunnen zeggen dat de twee waarden (totaal) onbekend zijn. We komen hier later uitvoerig op terug.

Eigenlijk is de term *null-waarde* niet geheel correct. Eigenlijk zouden we de kortere term *null* moeten gebruiken. Het is namelijk geen waarde, het is eerder een gat in een tabel of de waarde is afwezig. Toch hanteren we deze term in dit boek, omdat hij ook in de verschillende standaarden en producten gebruikt wordt.

1.3.3 Integriteitsregel

In de eerste paragraaf van dit hoofdstuk hebben we gesproken over de integriteit van de databasegegevens, de gegevens die in de tabellen liggen opgeslagen. De inhoud van een tabel behoort meestal aan bepaalde regels te voldoen, de zogenaamde *integriteitsregels* (constraints). Twee voorbeelden van *integriteitsregels* zijn: het spelersnummer van een speler mag niet negatief zijn en twee spelers mogen niet hetzelfde spelersnummer hebben. *Integriteitsregels* zijn vergelijkbaar met verkeersborden. Ook zij geven aan wat wel en wat niet mag.

Integriteitsregels behoren door een relationele databaseserver gecontroleerd te worden. Elke keer nadat de inhoud van een tabel gemuteerd is, behoort de databaseserver te controleren of de nieuwe gegevens aan de geldende *integriteitsregels* voldoen. Dit is duidelijk een taak van de databaseserver. De *integriteitsregels* moeten dan wel eerst gespecificeerd worden, zodat ze bij de databaseserver bekend zijn.

Integriteitsregels kunnen allerlei vormen hebben. Sommige komen in de praktijk zo vaak voor dat ze speciale namen hebben gekregen, zoals *primaire sleutel*, *kandidaatsleutel*, *alternatieve sleutel* en *refererende sleutel*. Ook hier gaat de analogie met verkeersborden op. Voor vaak voorkomende verkeersregels zijn ooit speciale symbolen bedacht die vervolgens op een verkeersbord zijn geplaatst. Die verkeersborden hebben namen gekregen, zoals *voorrangsbord* of *stopbord*. We zullen de benoemde *integriteitsregels* in de volgende paragrafen toelichten.



Figuur 1.3 Integriteitsregels zijn de verkeersborden van een database

1.3.4 Primaire sleutel

De *primaire sleutel* van een tabel is een kolom (of een combinatie van een aantal kolommen) die gebruikt kan worden als unieke identificatie voor de rijen in die tabel. Met andere woorden: twee verschillende rijen in een tabel mogen nooit in de primaire sleutel dezelfde waarde hebben en in elke rij moet de primaire sleutel altijd één waarde hebben. De SPELERSNR-kolom in de bovenstaande SPELERS-tabel kunnen we beschouwen als de primaire sleutel van die tabel. Twee spelers mogen dus nooit hetzelfde nummer hebben en er zal nooit een speler mogen zijn zonder spelersnummer. Het laatste wil zeggen dat null-waarden in een primaire sleutel niet toegestaan zijn.

Primaire sleutels komen overal voor. Bijvoorbeeld, de tabel waarin een bank gegevens over bankrekeningen vastlegt, zal de kolom bankrekeningnummer als primaire sleutel hebben. En als we een tabel bouwen waarin verschillende auto's geregistreerd worden, zal nummerbord de primaire sleutel kunnen zijn.



Figuur 1.4 Nummerbord als mogelijke primaire sleutel

1.3.5 Kandidaatsleutel

Sommige tabellen bezitten meer dan één kolom (of combinatie van kolommen) die aangewezen zou kunnen worden als primaire sleutel. Zij bezitten allemaal de uniciteitseigenschap van een primaire sleutel en tevens zijn ook bij hen null-waarden niet toegestaan. Deze kolommen (of combinaties van kolommen) worden *kandidaatsleutels* genoemd. Slechts één daarvan wordt echter aangewezen als de primaire sleutel. Een tabel heeft dus altijd minimaal één kandidaatsleutel.

Stel dat we in de SPELERS-tabel ook paspoortnummers registreren, dan zal die kolom een kandidaatsleutel zijn, want paspoortnummers zijn uniek. Twee spelers kunnen namelijk nooit hetzelfde paspoortnummer hebben. Deze kolom zouden we dan ook als primaire sleutel kunnen aanwijzen.

Opmerking

De term kandidaatsleutel komt niet voor in de taal SQL. Omdat het concept onderdeel van het relationele model is, zullen we het desondanks in dit boek wel gebruiken.

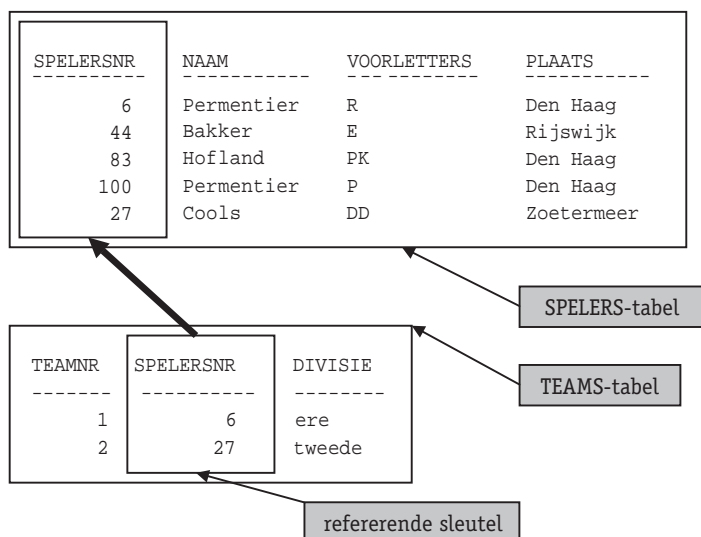
1.3.6 Alternatieve sleutel

Een kandidaatsleutel die niet tot primaire sleutel van een tabel gekozen is, wordt een *alternatieve sleutel* genoemd. Men kan dus stellen dat voor een tabel nul of meer alternatieve sleutels gedefinieerd kunnen zijn. Het begrip kandidaatsleutel is daarmee een verzamelnaam voor alle primaire en alternatieve sleutels. Als elke speler een paspoort zou moeten hebben en als we die ook in de SPELERS-tabel zouden registreren, dan zou de kolom PASPOORTNR een alternatieve sleutel zijn geweest.

1.3.7 Refererende sleutel

Een *refererende sleutel* is een kolom (of combinatie van kolommen) in een tabel waarvan de populatie een deelverzameling is van de populatie van de primaire sleutel van een tabel (dit hoeft niet een andere tabel te zijn). Andere benamingen voor refererende sleutel zijn: verwijzende sleutel, vreemde sleutel, foreign key of referential key.

Veronderstel dat er naast de SPELERS-tabel ook een TEAMS-tabel bestaat; zie figuur 1.5. De TEAMNR-kolom is de primaire sleutel van deze tabel. De SPELERSNR-kolom stelt de aanvoerder voor van een team. Dit moet wel een bestaand spelersnummer zijn, een die voorkomt in de SPELERS-tabel. De populatie van deze kolom is een deelverzameling van de populatie van de SPELERSNR-kolom in de SPELERS-tabel. SPELERSNR in de TEAMS-tabel wordt een refererende sleutel genoemd.



Figuur 1.5 De refererende sleutel

U ziet meteen hoe we ‘koppelingen’ tussen twee tabellen realiseren. We doen dit door in de TEAMS-tabel ook de SPELERSNR-kolom op te nemen. Hiermee hebben we een koppeling met de SPELERSNR-kolom van de SPELERS-tabel gelegd.

1.4 Wat is SQL?

Zoals vermeld is SQL een *relationele databasetaal*. De taal bevat onder andere instructies voor het invoeren, wijzigen, verwijderen, raadplegen en beveiligen van gegevens. Hieronder staan enkele instructies die met SQL te formuleren zijn:

- Voer de adresgegevens van een nieuwe werknemer in.
- Verwijder alle voorraadgegevens behorend bij product ABC.
- Geef het adres van werknemer Jansen.
- Geef het aantal verkochte schoenen per regio en per maand.
- Hoeveel producten zijn in de afgelopen drie maanden in Londen verkocht?
- Zorg dat de heer Jansen niet de salarisgegevens kan inzien.

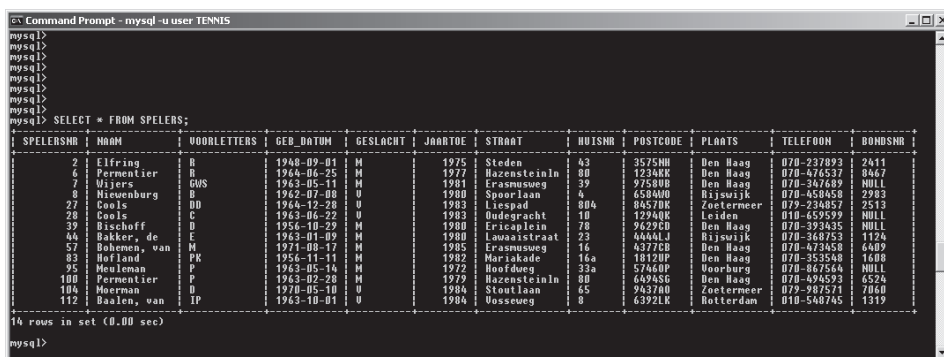
De taal is reeds door veel leveranciers als databasetaal voor hun databaseserver geïmplementeerd. IBM, Microsoft en Oracle zijn allemaal leveranciers van SQL-producten. Het is dus niet zo dat SQL de naam is van een bepaald product dat door één leverancier op de markt gebracht wordt. Al is SQL geen databaseserver, toch zullen we voor de eenvoud in dit boek onder SQL zowel de taal als de databaseserver verstaan. Alleen op die plaatsen waar dat nodig is zal een onderscheid gemaakt worden.

We noemen SQL een relationele databasetaal, omdat zij met gegevens omgaat volgens de regels gedefinieerd in het relationele model. (Eigenlijk moeten we stellen dat de theorie en SQL op bepaalde punten verschillen, maar hiervoor verwijzen we naar [CODD90].) En omdat SQL een relationele databasetaal is, is zij lange tijd geschaard geweest onder de declaratieve ofwel niet-procedurele databasetalen. Met *declaratief* en *niet-procedureel* wordt bedoeld dat gebruikers met behulp van instructies slechts hoeven op te geven met welke gegevens zij willen werken. Zij hoeven niet aan te geven *hoe* die gegevens een voor een benaderd moeten worden. Bekende talen als C, C++, Java, PHP, Ruby en Visual Basic zijn voorbeelden van procedurele talen. In deze context betekent de term *procedureel* dat alle stappen die een programma moet uitvoeren om het gewenste resultaat te bereiken, allemaal expliciet geprogrammeerd moeten worden. Vanuit dit perspectief categoriseren we de objectgeoriënteerde programmeertalen ook als procedurele talen.

We kunnen SQL al lang niet meer een puur declaratieve taal noemen. Sinds het begin van de jaren negentig van de vorige eeuw zijn door veel leveranciers procedurele extensies aan SQL toegevoegd. Hiermee worden procedurele databaseobjecten als *triggers* en *stored procedures* gecreëerd; zie deel IV. Aan SQL zijn ook klassieke instructies als IF-THEN-ELSE en WHILE-DO toegevoegd. Al zijn de meest bekende SQL-instructies nog steeds niet-procedureel van aard, SQL is veranderd in een hybride taal bestaande uit procedurele- en niet-procedurele instructies. Ook MySQL is uitgebreid met deze procedurele databaseobjecten.

SQL is op twee manieren te gebruiken. Ten eerste kan de taal *interactief* gebruikt worden: een SQL-instructie wordt dan door een gebruiker ter plekke ingevoerd en direct door de databaseserver verwerkt. Het resultaat is ook direct zichtbaar. Interactief SQL is voornamelijk bedoeld voor de ontwikkelaars van programma's en voor die eindgebruikers die zelf rapporten willen creëren.

De producten die interactief SQL ondersteunen, kunnen we in twee groepen verdelen: die met een wat ouderwetse, terminalachtige interface en die met een moderne grafische interface. Bij MySQL wordt een terminalachtig product meegeleverd dat dezelfde naam als de databaseserver draagt: `mysql`. Figuur 1.6 laat zien hoe dit programma eruitziet. Eerst staat er een SQL-instructie (`SELECT * FROM SPELERS`) met daaronder in tabelvorm het resultaat.



```

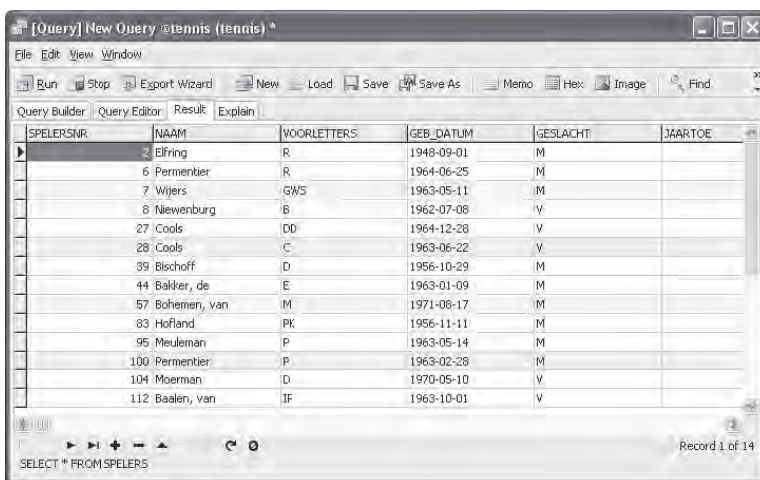
C:\ Command Prompt - mysql-u user TENNIS
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> SELECT * FROM SPELERS;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| SPELERSNR | NAAM          | VOORLETTERS | GEB_DATUM | GESLACHT | JAARTOE | STRAAT      | HUISNR | POSTCODE | PLAATS   | TELEFOON | BOMHSNR |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 2 | Elfring       | R           | 1948-09-01 | M        | 1975    | Steden      | 43      | 3575NH   | Ben Haag | 070-237893 | 2411 |
| 6 | Permentier   | R           | 1964-06-25 | M        | 1977    | Hazensteinlaan | 80      | 1234KE   | Ben Haag | 070-474537 | 8467 |
| 7 | Wijers       | GWS        | 1963-05-11 | M        | 1981    | Erasmusweg  | 39      | 97580B   | Ben Haag | 070-347489 | NULL |
| 8 | Nieuwenburg | B          | 1962-07-08 | V        | 1980    | Spoorlaan   | 4        | 65840D   | Bissuijk | 070-458450 | 2983 |
| 27 | Cools        | DD         | 1964-12-28 | V        | 1983    | Liespad     | 804      | 94570E   | Zoetermeer | 071-220457 | 2513 |
| 28 | Cools        | C          | 1963-06-22 | V        | 1983    | Oudegracht  | 10       | 12940K   | Leiden   | 010-659599 | NULL |
| 39 | Bischoff     | D          | 1956-10-29 | M        | 1980    | Ericaplein  | 78       | 9629CD   | Ben Haag | 070-393435 | NULL |
| 44 | Bakker, de   | E          | 1963-01-09 | M        | 1980    | Louwistraat | 23       | 4004LI   | Bissuijk | 070-368752 | 1124 |
| 57 | Bohemen, van | M          | 1971-08-17 | M        | 1985    | Erasmusweg  | 16       | 4377CB   | Ben Haag | 070-473458 | 6489 |
| 83 | Hofland     | PK         | 1956-11-11 | M        | 1982    | Mariakade   | 16a      | 18120P   | Ben Haag | 070-353548 | 1688 |
| 95 | Meuleman    | P          | 1963-05-14 | M        | 1972    | Boefdoeng   | 32a      | 57460P   | Voorburg | 070-867544 | NULL |
| 100 | Permentier  | P          | 1963-02-28 | M        | 1979    | Hazensteinlaan | 80       | 6494SG   | Ben Haag | 070-494593 | 6524 |
| 104 | Moerman     | D          | 1970-05-10 | V        | 1984    | Stoutlaan   | 65       | 94378D   | Zoetermeer | 071-987571 | 7860 |
| 112 | Baalen, van | IF         | 1963-10-01 | V        | 1984    | Vosseweg    | 8        | 6392LE   | Rotterdam | 010-548745 | 1319 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
14 rows in set (0.00 sec)

mysql>

```

Figuur 1.6 Een voorbeeld van het opvraagprogramma genaamd **mysql** waarmee SQL-instructies interactief gespecificeerd kunnen worden

Maar er zijn ook producten beschikbaar voor interactief gebruik met een meer grafische interface, zoals MySQL Query Browser, SQLyog van Webyog, phpMyAdmin, Navicat van PremiumSoft (zie figuur 1.7) en WinSQL van Synametrics (zie figuur 1.8).



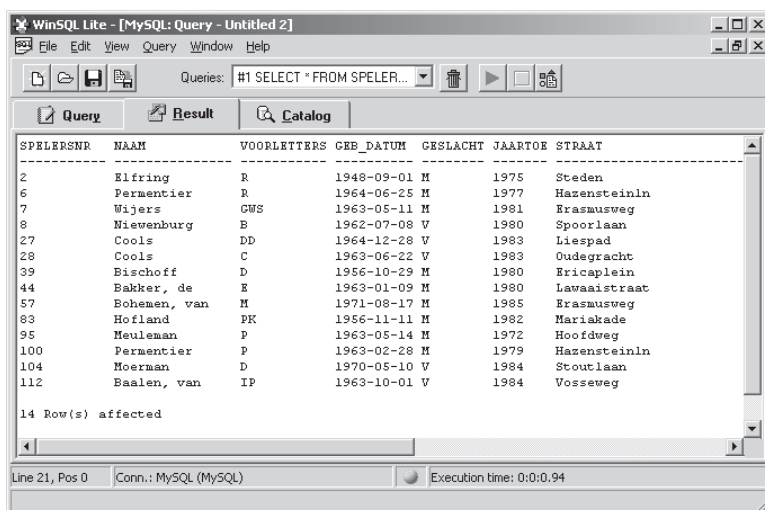
```

[Query] New Query @tennis (tennis) ^
File Edit View Window
Run Stop Export Wizard New Load Save Save As Memo Hex Image Find
Query Builder Query Editor Result Explain
+-----+-----+-----+-----+-----+
| SPELERSNR | NAAM          | VOORLETTERS | GEB_DATUM | GESLACHT | JAARTOE |
+-----+-----+-----+-----+-----+
| 2 | Elfring       | R           | 1948-09-01 | M        | 1975    |
| 6 | Permentier   | R           | 1964-06-25 | M        | 1977    |
| 7 | Wijers       | GWS        | 1963-05-11 | M        | 1981    |
| 8 | Nieuwenburg | B          | 1962-07-08 | V        | 1980    |
| 27 | Cools        | DD         | 1964-12-28 | V        | 1983    |
| 28 | Cools        | C          | 1963-06-22 | V        | 1983    |
| 39 | Bischoff     | D          | 1956-10-29 | M        | 1980    |
| 44 | Bakker, de   | E          | 1963-01-09 | M        | 1980    |
| 57 | Bohemen, van | M          | 1971-08-17 | M        | 1985    |
| 83 | Hofland     | PK         | 1956-11-11 | M        | 1982    |
| 95 | Meuleman    | P          | 1963-05-14 | M        | 1972    |
| 100 | Permentier  | P          | 1963-02-28 | M        | 1979    |
| 104 | Moerman     | D          | 1970-05-10 | V        | 1984    |
| 112 | Baalen, van | IF         | 1963-10-01 | V        | 1984    |
+-----+-----+-----+-----+-----+
Record 1 of 14

SELECT * FROM SPELERS

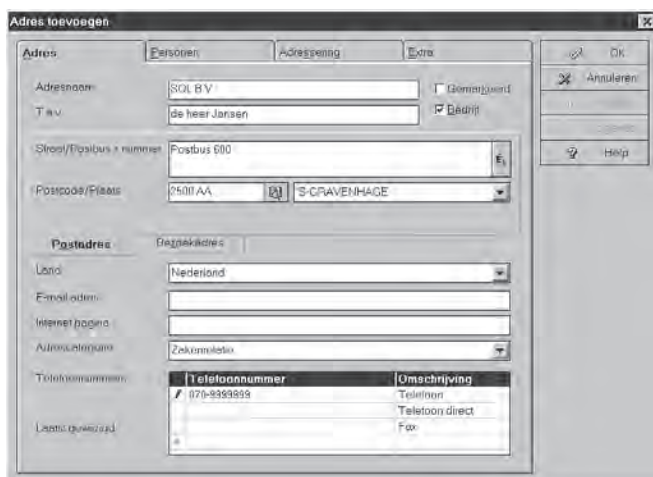
```

Figuur 1.7 Een voorbeeld van het opvraagprogramma Navicat



Figuur 1.8 Een voorbeeld van het opvraagprogramma WinSQL

De tweede gebruikswijze wordt *voorgeprogrammeerd SQL* genoemd. SQL-instructies zijn dan opgenomen in een programma dat in een andere programmeertaal geschreven is. Resultaten van SQL-instructies zijn in dit geval voor de gebruiker niet direct zichtbaar, maar worden door het *omhullende* programma bewerkt. Voorgeprogrammeerd SQL wordt voornamelijk gebruikt in programma's die voor eindgebruikers ontwikkeld zijn. Deze eindgebruikers behoeven dan geen SQL-instructies te leren, maar krijgen bijvoorbeeld eenvoudige menu's en schermen te zien. Voorbeelden hiervan zijn programma's om klantgegevens bij te houden en programma's ten behoeve van voorraadbeheer. Figuur 1.9 bevat een voorbeeld van een scherm met velden waarmee de gebruiker adresgegevens kan intikken zonder dat hij SQL hoeft te kennen. Het programma dat 'achter' dit scherm draait, is geprogrammeerd om bepaalde SQL-instructies aan de databaseserver door te geven. Het programma gebruikt dus zelf wel SQL-instructies om de ingetikte gegevens in de database te krijgen.

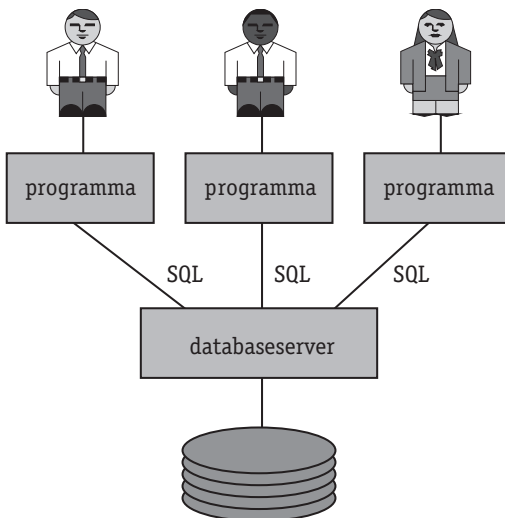


Figuur 1.9 SQL is in veel programma's afgeschermd, gebruikers zien alleen invoervelden

In de beginjaren van SQL bestond er maar één methode van voorgeprogrammeerd SQL, *embedded SQL* geheten. In de jaren negentig zijn daar andere methoden bijgekomen. De belangrijkste wordt *call level interface SQL* (CLI SQL) genoemd. Er bestaan verschillende CLI-varianten, zoals ODBC (Open Database Connectivity) en JDBC (Java Database Connectivity). In dit boek beschrijven we de belangrijkste. De verschillende methoden van voorgeprogrammeerd SQL worden in de Engelstalige literatuur *binding styles* genoemd.

De instructies en mogelijkheden van interactief en voorgeprogrammeerd SQL zijn nagenoeg gelijk. Hiermee bedoelen we dat de meeste instructies die interactief ingevoerd en verwerkt kunnen worden, ook in een SQL-programma opgenomen kunnen zijn. Wel is voorgeprogrammeerd SQL uitgebreid met enkele instructies die uitsluitend toegevoegd zijn om de koppeling tussen SQL-instructies en niet-SQL-instructies mogelijk te maken. Wij zullen in dit boek voornamelijk uitgaan van interactief SQL. Voorgeprogrammeerd SQL wordt in een apart deel van dit boek behandeld.

Bij de interactieve en voorgeprogrammeerde verwerking van SQL-instructies staan drie componenten centraal: de gebruiker, het programma en de databaseserver; zie figuur 1.10. De databaseserver is verantwoordelijk voor het opslaan en benaderen van gegevens op schijf. Hier heeft het programma en zeker de gebruiker niets mee te maken. De databaseserver verwerkt de SQL-instructies die door het programma aangeleverd worden. Op een of andere wijze kunnen het programma en de databaseserver SQL-instructies uitwisselen. Het resultaat van een SQL-instructie wordt weer aan de gebruiker teruggegeven.



Figuur 1.10 Bij de verwerking van SQL staan de gebruiker, het programma en de databaseserver centraal

Opmerking

SQL wordt dus als een relationele databasetaal geclassificeerd. SQL voldoet echter niet aan alle regels van het relationele model. Bijvoorbeeld, het relationele model staat geen tabellen toe met dubbele rijen, SQL wel. Een ander voorbeeld is dat elke tabel volgens het relationele model een primaire sleutel moet hebben. Dit geldt echter niet voor SQL, want we kunnen wel degelijk tabellen zonder primaire sleutels creëren. Kortom, SQL is geen exacte implementatie van het relationele model. Om deze reden vinden sommige

experts dat we SQL geen relationele databasetaal mogen noemen. En voor deze mening is veel te zeggen. In dit boek zullen we echter SQL toch als relationele databasetaal classificeren om in lijn te blijven met de handleidingen en andere literatuur over SQL.

1.5 Verschillende categorieën SQL-programma's

SQL wordt in een breed scala aan programma's gebruikt. Als SQL toegepast wordt, is het belangrijk om in te zien welk type programma ontwikkeld wordt. Voert een programma bijvoorbeeld veel eenvoudige SQL-instructies uit of juist enkele zeer complexe? Dit kan invloed hebben op hoe SQL-instructies het meest efficiënt geformuleerd kunnen worden, welke instructies worden gekozen en hoe SQL wordt ingezet. Om de discussie hierover te vereenvoudigen, introduceren we hier een hiërarchische indeling van programma's waar we in de andere hoofdstukken, indien relevant, naar zullen verwijzen.

1 programma met voorgeprogrammeerd SQL

- a invoerprogramma
 - i online-invoerprogramma
 - ii batch-invoerprogramma
- b batch-rapportageprogramma

2 programma met interactief SQL

- a opvraagprogramma
 - i direct SQL
 - ii Query-By-Example
 - iii natuurlijke taal
- b business intelligence programma
 - i OLAP-programma
 - ii statistisch programma
 - iii KPI-programma
 - iv visualisatie-programma

De eerste onderverdeling heeft te maken met het al dan niet voorgeprogrammeerd zijn van SQL-instructies. Bij de producten die interactief SQL ondersteunen is dat niet het geval. Hier bepaalt de gebruiker, direct of indirect, welke SQL-instructies gecreëerd worden.

De programma's die voorgeprogrammeerd zijn, worden onderverdeeld in invoer- (1.a) en batch-rapportageprogramma's (1.b). Van de *invoerprogramma's* bestaan twee varianten: online (1.a.i) en batch (1.a.ii). Een online-invoerprogramma, bijvoorbeeld, kan geschreven zijn in Java of Visual Basic met SQL en staat toe dat gegevens aan een database toegevoegd worden. Karakteristiek worden dergelijke programma's door veel gebruikers tegelijkertijd gebruikt en de SQL-instructies die voorgeprogrammeerd zijn, zijn relatief eenvoudig. Batch-invoerprogramma's lezen bestanden met nieuwe gegevens in en voegen die aan de bestaande databasegegevens toe. Deze programma's draaien meestal op geregelde tijden en voeren veel werk uit. Ook hier zijn de SQL-instructies relatief eenvoudig.

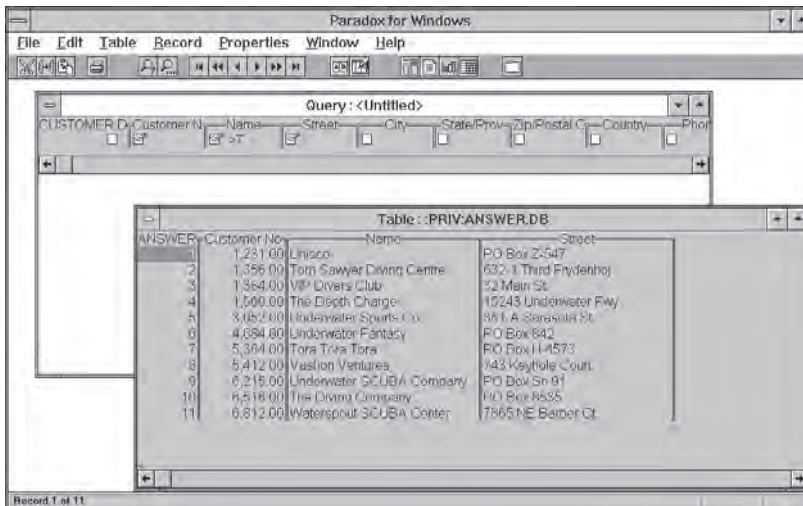
Een batch-rapportageprogramma (1.b) genereert rapporten. Bijvoorbeeld, op elke zondag wordt een rapport aangemaakt met de totale verkoopcijfers uitgesplitst naar regio en dit rapport wordt maandagochtend met de interne post afgeleverd op de kamer van een manager of naar zijn e-mailadres gestuurd. Veel bedrijven gebruiken dergelijke program-

ma's. Een batch-rapportageprogramma bevat gewoonlijk zeer weinig, maar wel complexe instructies.

De markt voor programma's waarbij gebruikers interactief met SQL werken, is wat uitgebreider. Bij de eerste subcategorie, de opvraagprogramma's (2.a), moet de gebruiker bekend zijn met de relationele concepten. Om vragen te stellen werkt hij onder andere met tabellen, kolommen, rijen, primaire sleutels en refererende sleutels.

Binnen de categorie opvraagprogramma's zijn weer drie subcategorieën te identificeren. Bij de eerste (2.a.i) moeten SQL-instructies direct door de gebruiker ingetikt worden. Deze moet dus volledig op de hoogte zijn van de grammatica van SQL. De reeds genoemde programma's MySQL Query Browser, Navicat en WinSQL zijn voorbeelden van dergelijke opvraagprogramma's; zie figuren 1.7 en 1.8. Bij figuur 1.8 staat in het midden van het scherm een SQL-instructie met eronder het resultaat.

Subcategorie 2.a.ii, genaamd *Query-By-Example* (QBE), is in de jaren zeventig ontwikkeld door Moshé Zloof; zie [ZLOO77]. Het was bedoeld als relationele databasetaal en op een bepaalde manier als alternatief voor SQL. Uiteindelijk is de taal model gaan staan voor een verzameling producten met een bepaalde interfacestijl. Gebruikers van QBE behoeven de syntaxis van SQL niet te kennen, want SQL-instructies worden gegenereerd. Simpelweg, gebruikers tekenen tabellen en vullen vervolgens hun condities en specificaties in die tabellen in. Iemand heeft QBE wel eens gedefinieerd als een grafische versie van SQL. Dit is niet een geheel correcte bewering, maar geeft wel ongeveer aan wat QBE is. Figuur 1.11 toont hoe een QBE-vraag eruitziet.



Figuur 1.11 Een voorbeeld van Query-By-Example

Enkele leveranciers hebben getracht gebruikers de kracht en flexibiliteit van SQL te bieden zonder dat ze die taal hoefden te leren (2.a.iii). Dit is geïmplementeerd door er een natuurlijke taalinterface bovenop te plaatsen. Gebruikers kunnen hiermee hun vragen gewoonweg in het Nederlands of Engels formuleren, die vervolgens naar SQL vertaald worden. De markt voor dergelijke producten is altijd klein geweest, maar het blijft nog steeds een boeiende categorie opvraagprogramma's.

Nogmaals, bij al deze opvraagprogramma's moet de gebruiker de principes van relationele databases wel begrijpen. Voor sommige gebruikers is dat te technisch. Zij die geen technische achtergrond hebben, maar experts zijn op het gebied van marketing, logistiek of verkoop, willen toch de databasegegevens benaderen. Hiervoor zijn de *business intelligence*-programma's (2.b) ontworpen. Hier wordt een dikke laag software boven op SQL geplaatst. Van SQL en relationele concepten is dan niets meer zichtbaar.

Zeer populair binnen de business intelligence-programma's is de categorie OLAP-programma's (2.b.i). OLAP staat voor *OnLine Analytical Processing* en is een term geïntroduceerd door E. F. Codd. Het zijn producten ontworpen voor gebruikers die bijvoorbeeld verkoop-, marketing- of productiecijfers van verscheidene kanten willen bekijken en op verschillende niveaus van detail willen bestuderen.

Gebruikers van OLAP-programma's zien niet de bekende relationele interface, dus geen 'platte' tabellen of SQL, maar een zogenaamde *multidimensionale interface*. Gegevens zijn logisch gegroepeerd binnen *kubussen*. Kubussen bestaan uit dimensies, zoals regio, product en tijd. Binnen een dimensie kunnen hiërarchieën van elementen worden opgebouwd. Het element Amsterdam bijvoorbeeld behoort tot Nederland en Nederland behoort tot Noord-Europa. Alle drie behoren tot de dimensie regio. Helaas moeten we concluderen dat de leveranciers allemaal hun eigen terminologie hebben. Array, model, variabele en multidimensionale tabel zijn bijvoorbeeld allemaal alternatieve namen voor kubus.

Het valt buiten de context van dit boek om een gedetailleerd beeld van OLAP te schetsen. Om toch een idee te geven is een eenvoudig voorbeeld opgenomen. Figuur 1.12 bevat een aantal verkoopcijfers per regio. Er zijn drie verkoopregio's te zien: Boston, Portland en Concord. Wat ook te zien is, is dat in de vierde kolom het verkoopcijfer van Boston tussen haakjes staat. Dit impliceert dat dit getal te laag is. Dat klopt ook in deze situatie, want Boston had voor 128.549 moeten verkopen (zie de tweede kolom van de tabel), maar er is slechts voor 91.734 verkocht (zie de eerste kolom). Dit is een duidelijke achterstand op de

	Sales (\$)	Plan Sales (\$)	Sales (Units)	Sales-Plan	Sales-Plan <Rank>	Sales-Plan %Variation	LY Sales (\$)	LY Plan Sales (\$)	LY Sales-Plan	LY Sales-Plan %Variation
Boston	91,734	128,549	1,118	(36,815)	1	-29%	89,865	81,044	8,821	11%
Portland	89,305	85,866	1,094	3,440	2	4%	82,291	80,295	1,996	2%
Concord	92,821	87,906	1,124	4,918	3	6%	88,987	81,114	7,873	10%

Figuur 1.12 Drie verkoopregio's met hun respectievelijke verkoopcijfers

geplande verkoop. Managers willen waarschijnlijk nu weten wat hiervan de reden is. Zij willen meer gedetailleerde cijfers zien. Hiervoor hoeft uitsluitend op het woord Boston geklikt te worden om de cijfers die in figuur 1.13 staan tevoorschijn te halen. Hier zijn de verkoopcijfers in Boston uitgesplitst naar product. Duidelijk is nu dat het niet de gehele verkoop is die tegenvalt, maar alleen die voor boormachines (power drills). Uiteraard is dit niet het einde van het verhaal en weet de gebruiker nog steeds niet wat er aan de hand is, maar hopelijk is voor u de kracht van OLAP duidelijk. Zonder SQL te hoeven leren, kan de gebruiker met gegevens spelen en dat is zeer waardevol voor hen. Voor een meer uitvoerige beschrijving van OLAP verwijzen we naar [THOM02].

The screenshot shows a software window titled "DSS Agent: True Relational OLAP" with a menu bar (File, Edit, View, Tools, Window, Help) and a toolbar. The main display area contains a table titled "Boston" with the following data:

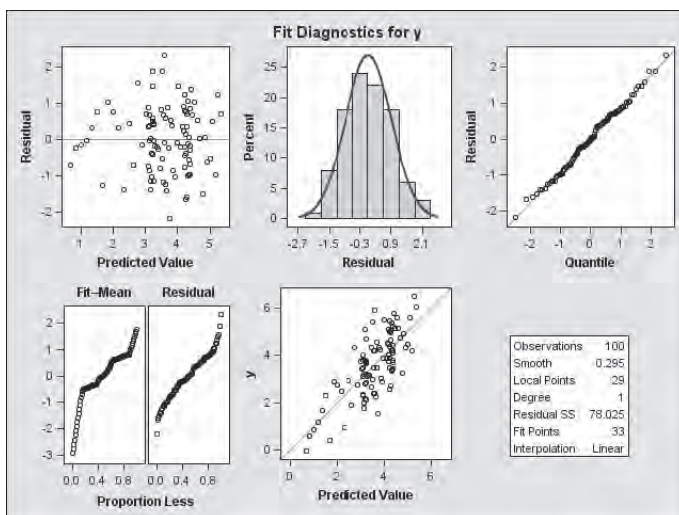
		Sales (\$)	Plan Sales (\$)	Sales-Plan	Sales (Units)	Avg Inventory	Turnover (Days)	Sell Through
Power Tools	Power Drill (3/8")	36,263	75,251	(38,968)	330	35	5	100%
	Skill Saw	16,281	15,438	843	181	62	19	80%
	Electric Sander	15,584	14,785	799	312	71	11	70%
	Cordless Drill	23,585	23,074	511	295	60	12	73%
Hand Tools	Handi Screwdriver Set	10,932	10,480	452	365	67	9	91%
	Ratchet Kit (74 Piece)	18,787	14,990	3,797	264	65	13	40%
	Adjustable Wrench Set	14,182	13,310	872	355	68	10	85%
	Hammer (28oz.)	8,928	8,073	2,855	273	75	13	92%
Electrical	Romex Wire (3 Strand)	8,184	7,769	415	341	71	10	67%
	Wall Switches (White)	7,584	7,251	333	316	67	11	84%
	Outlets (White)	7,032	6,531	501	293	63	12	43%
	Outlet Box (Single)	8,544	8,225	319	356	68	10	91%
Lawn Products	Lawn Sprinkler	7,744	5,979	1,765	261	70	13	78%
	Garden Hose (75')	8,688	8,293	395	362	69	10	91%
	Lawn Mower (3/4 hp)	6,672	6,375	297	278	68	13	88%
	Delux Leaf Rake	8,640	8,282	358	360	70	9	78%

The bottom of the window features a navigation toolbar with buttons: Back, Fwd, Home, History, Tools, Help, and Exit.

Figuur 1.13 De verkoopcijfers van de verkoopregio Boston uitgesplitst naar product

Categorie 2.b.ii, de statistische programma's, bestaan waarschijnlijk het langst. Deze producten bieden hun eigen talen om gegevens in databases te benaderen en uiteraard om statistische analyses uit te voeren; zie figuur 1.14. Deze categorie bevat ook de *datamining*-programma's. OLAP-programma's maken het mogelijk dat gebruikers op eenvoudige wijze gegevens, opgeslagen in de database, van diverse kanten kunnen bekijken. Ze presenteren meestal gegevens of gesommeerde gegevens die in de database liggen opgeslagen. Daarentegen zullen datamining-programma's nooit als resultaat een gegeven of een totaal presenteren. Hun kracht ligt in het ontdekken van trends en patronen in de gegevens. Ze trachten bijvoorbeeld te achterhalen of bepaalde producten vaak samen gekocht worden, wat de dominante karakteristieken zijn van klanten die levensverzekeringen afsluiten of wat de karakteristieken zijn van een goed lopend product in een grote stad. Voor een uitgebreide introductie tot datamining verwijzen we naar [LARO04].

Uiteraard moeten ook deze programma's gegevens benaderen om de trends te ontdekken. Over het algemeen hebben ze hiervoor hun eigen specialistische en zeer krachtige talen. Achter de schermen worden dan, indien relevant, hun eigen instructies vertaald naar SQL-instructies.



Figuur 1.14 Het resultaat van een statistische analyse gemaakt met SAS-software

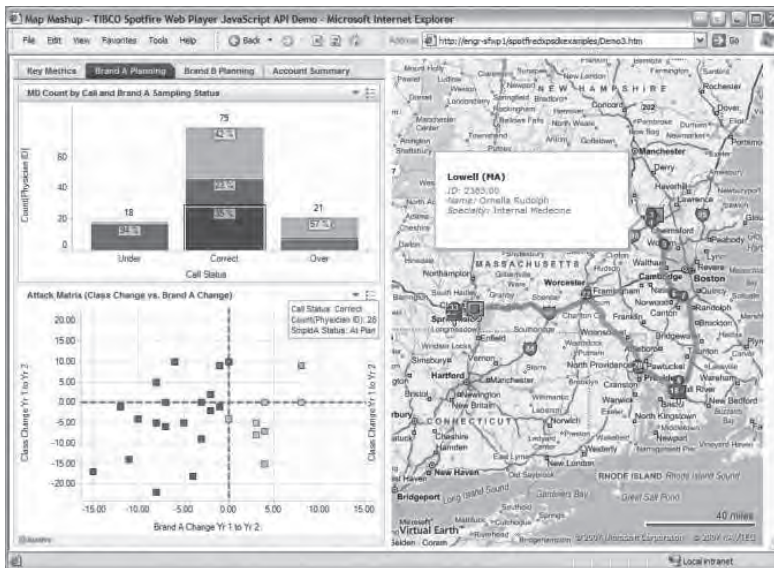
Een KPI (Kritische Prestatie Indicator) is een variabele om prestaties van organisaties te analyseren. Het is een instrument voor het management om inzage te hebben in bepaalde bedrijfsprocessen. In één oogopslag zien ze of alles goed gaat of dat er ingegrepen moet worden. Met subcategorie 2.b.iii, de KPI-programma's, worden deze variabelen op een simpele wijze als een dashboard gepresenteerd; zie figuur 1.15. Meestal kunnen deze programma's ook zelf een signaal afgeven waarmee ze de gebruikers waarschuwen dat er iets aan de hand is.



Figuur 1.15 Een voorbeeld van een KPI-programma

Visualisatie-programma's (2.b.iv) zijn zeer rijk aan mogelijkheden om gegevens op verschillende wijzen te tonen. De gegevens kunnen als tabellen, grafieken en geografisch gepresenteerd worden en soms kunnen ze ook in een figuur (foto, tekening) geplaatst worden.

De doelstelling is dat de gebruikers inzage in de gegevens krijgen. De meeste van deze programma's staan toe dat gebruikers zeer dynamisch met de gegevens kunnen werken en dat ze eenvoudig gecombineerd kunnen worden; zie figuur 1.16.



Figuur 1.16 Een visualisatie-programma; met toestemming van Tibco Corporation

De complexiteit van de SQL-instructies die door statistische-, OLAP-, KPI- en visualisatie-programma's gegenereerd worden, kan hoog zijn. Dit betekent dat databaseservers veel werk moeten verrichten om deze instructies te verwerken.

Ongetwijfeld zullen er in de toekomst nog veel categorieën programma's bijkomen, maar momenteel zijn dit de meest gebruikte.

Ten slotte, hoe een programma de databaseserver kan vinden, dus hoe ze instructies en resultaten onderling kunnen uitwisselen, wordt in dit boek niet in detail behandeld. We lichten in deel V een tipje van de sluier op. Voorlopig kunt u ervan uitgaan dat als een programma een databaseserver wil benaderen, een speciaal stukje code met het programma meegelinkt moet worden. Dit stukje code wordt *middleware* genoemd en is (waarschijnlijk) ontwikkeld door de leverancier van de databaseserver. Het is te vergelijken met een loodsmans die aan boord komt om het schip naar de goede aanlegplaats te leiden.

1.6 De geschiedenis van SQL

De geschiedenis van SQL begint begin jaren zeventig in de onderzoekslaboratoria van IBM. De lange en bewogen historie die daar op volgt is te verdelen in vier decennia.

1970 – 1980: Het begin van de geschiedenis van SQL is nauw verweven met de geschiedenis van een IBM-project, genaamd *System R*. In dit project werd een experimentele relationele databaseserver ontwikkeld die dezelfde naam droeg als het project: *System R*. Dit systeem werd gebouwd in het onderzoekslaboratorium van IBM in San Jose, Californië. Het project moest aantonen dat de positieve gebruikersgemakken van het relationele model geïmplementeerd konden worden in een systeem dat voldeed aan alle eisen van een moderne databaseserver.

Een probleem dat in het System R-project opgelost moest worden, was dat er nog geen relationele databasetalen bestonden. Het was dus belangrijk dat er een ontwikkeld werd. Dit was de geboorte van de taal genaamd *Sequel*. De eerste publicaties over deze taal werden geschreven door de ontwerpers R.F. Boyce en D.D. Chamberlin; zie [BOYC73a] en [CHAM76]. Later werd de taal herdoopt tot SQL, omdat de naam Sequel conflicteerde met een bestaand trademark. (De naam SQL wordt echter nog vaak uitgesproken als 'siekwol'.)



Figuur 1.17 Don Chamberlin, een van de ontwerpers van SQL

Het System R-project werd in drie fasen uitgevoerd. In de eerste fase, fase nul (van 1974 tot 1975), werd slechts een deel van SQL geïmplementeerd. De join (voor het koppelen van gegevens uit verschillende tabellen) werd bijvoorbeeld nog niet geïmplementeerd en van het systeem werd alleen een single-userversie gebouwd. De bedoeling van deze fase was om te kijken of het implementeren van een dergelijk systeem eigenlijk wel mogelijk was. Deze fase werd met succes afgerond; zie [ASTR80].

In 1976 werd begonnen met fase 1. Alle programmacode, geschreven voor fase nul, werd opzij gelegd en men begon opnieuw. Fase 1 omvatte het totale systeem. Dit betekende onder andere dat de multiuser-mogelijkheid en de join opgenomen werden. De ontwikkeling van fase 1 vond plaats tussen 1976 en 1977.

In de laatste fase werd System R geëvalueerd. Het systeem werd op diverse plaatsen binnen IBM en bij een aantal grote IBM-klienten geïnstalleerd. De evaluatie vond plaats in 1978 en 1979 en deze resultaten zijn onder andere in [CHAM80] beschreven. In het jaar 1979 werd het System R-project afgesloten.

1980 – 1990: De in deze drie fasen opgedane kennis en ontwikkelde technologie werden gebruikt om SQL/DS te bouwen. SQL/DS was IBM's eerste relationele databaseserver die commercieel beschikbaar kwam. In 1981 kwam het product voor het besturingssysteem DOS/VSE op de markt. Later, in 1983, verscheen de VM/CMS-versie. In datzelfde jaar werd tevens DB2 aangekondigd. DB2 is momenteel beschikbaar voor veel besturingssystemen.

Over de ontwikkeling van System R heeft IBM veel gepubliceerd. Dit gebeurde overigens in een tijd dat op conferenties en seminars veel gesproken werd over relationele databaseservers. Het was niet verwonderlijk dat ook andere leveranciers begonnen met het bouwen van relationele systemen. Sommige daarvan, waaronder Oracle, implementeerden SQL als databasetaal. Ondertussen zijn er veel SQL-producten bijgekomen, wat ertoe geleid heeft dat SQL nu beschikbaar is op alle mogelijke systemen, van klein tot groot. Ook zijn bestaande databaseservers later met SQL uitgebreid.

Tijdens dit decennium heeft er ook een strijd rondom SQL plaatsgevonden. De nieuwe SQL-producten en hun jonge leveranciers moesten een aandeel veroveren in een markt die op dat moment volledig in handen was van niet-SQL-databaseservers, zoals IMS, IDMS en Total. Bovendien moesten ze het tegen elkaar opnemen. In deze jaren ontwikkelden de producten zich ook van veelbelovende nieuwkomers tot volwaardige databaseservers met uitgebreide mogelijkheden en capaciteiten. Werden de producten eerst voornamelijk ingezet voor rapportage, uiteindelijk waren ze ook krachtig genoeg om invoerprogramma's te ondersteunen. In deze periode werd ook de eerste versie van de SQL-standaard gepubliceerd hetgeen tot enige homogeniteit leidde.

1990 – 2000: In de jaren negentig hadden SQL-producten als DB2, Informix, Oracle, Progress en Sybase een dominante positie in de databasemarkt veroverd. Veel organisaties migreerden van een niet-SQL-databaseserver naar SQL en zij die nog geen databaseserver in gebruik hadden, begonnen direct met SQL. In deze periode is de hegemonie van SQL opgebouwd. Dit succes had ook veel te maken met de opkomst van goedkope en kleine servers, de client/server-architectuur en besturingssystemen als Unix en Windows. SQL-producten waren ook veel eerder op de nieuwe machines beschikbaar.

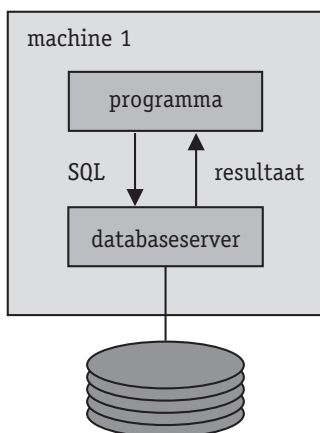
2000 – 2010: In de eerste tien jaren van de 21^{ste} eeuw hebben de SQL-producten hun markt alleen maar vergroot. Op diverse fronten zijn de producten uitgebreid om te kunnen voldoen aan de nieuwe wensen van organisaties op de gebieden performance, schaalbaarheid en betrouwbaarheid. De meeste zijn nu uitgebreid met mogelijkheden om XML-documenten op te slaan en te bewerken, en met mogelijkheden om in ongestructureerde tekst te zoeken, enzovoorts. Maar wat misschien wel het opvallendst aan deze periode is, is dat er veel nieuwe SQL-producten op de markt zijn verschenen. Naast de commerciële producten, zijn er nu ook *open source* SQL-databaseservers, waaronder Firebird, MySQL, PostgreSQL en SQLite. Tevens zijn er speciale SQL-producten op de markt gekomen die speciaal voor business intelligence-programma's ontwikkeld zijn. Dit worden ook wel *analytische databaseservers* genoemd. Deze producten zijn intern anders opgebouwd. De gehele interne architectuur is gericht op het snel verwerken van complexe vragen.

Kortom, de ontwikkeling van SQL staat niet stil. De taal is uitgebreid, de producten zijn op alle fronten verbeterd en hun markt is vergroot. SQL is klaar voor het komende decennium.

1.7 Van monolithisch via client/server naar internet

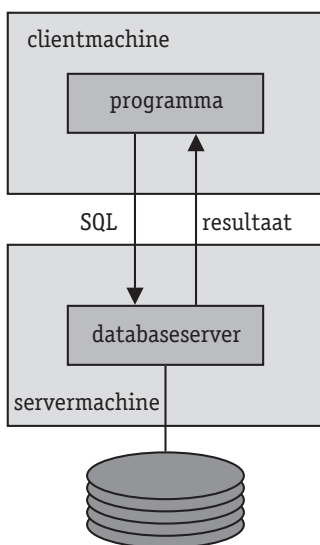
Tot aan het einde van de jaren tachtig van de vorige eeuw bestond er maar één architectuur waarin een SQL-databaseserver kon worden gebruikt: de *monolithische architectuur*. Bij een monolithische architectuur draait alles op een en dezelfde machine. Deze machine kan een groot mainframe zijn, een midrange-machine, een pc of zelfs een mobiele telefoon. Tegenwoordig bestaan er veel meer architecturen, waarvan client/server en internet zeer populair zijn.

De monolithische architectuur bestaat nog steeds, zie figuur 1.18. Bij deze architectuur draaien het programma en de databaseserver op dezelfde machine. Zoals beschreven in paragraaf 1.4 geeft het programma SQL-instructies aan de databaseserver door, die ze vervolgens verwerkt en de resultaten aan het programma teruggeeft. Ten slotte worden de resultaten aan de gebruikers getoond. Omdat het programma en de databaseserver beide op dezelfde machine draaien, vindt de communicatie via zeer snelle interne lijnen plaats. In feite hebben we hier te maken met twee processen die intern communiceren.



Figuur 1.18 De monolitische architectuur

Door de komst van goedkopere en snelle kleine machines is in de jaren negentig van de vorige eeuw de *client/server-architectuur* geïntroduceerd. Hiervan bestaan weer diverse subvormen. We zullen deze hier niet allemaal behandelen, maar alleen de populairste. Bij deze vorm van client/server draait het programma op een andere machine dan de databaseserver; zie figuur 1.9. De machine waarop het programma draait, wordt de *clientmachine* genoemd en de andere de *servermachine*. We zeggen dan dat er gewerkt wordt met een *remote databaseserver*. Onderlinge communicatie verloopt meestal via een lokaal netwerk (LAN) en incidenteel via een WAN (Wide Area Network). Het zou kunnen zijn dat een gebruiker in Parijs een programma op zijn of haar pc start en gegevens uit een database ophaalt die zich op een servermachine in Sydney bevindt. De communicatie verloopt dan waarschijnlijk via een satellietverbinding.

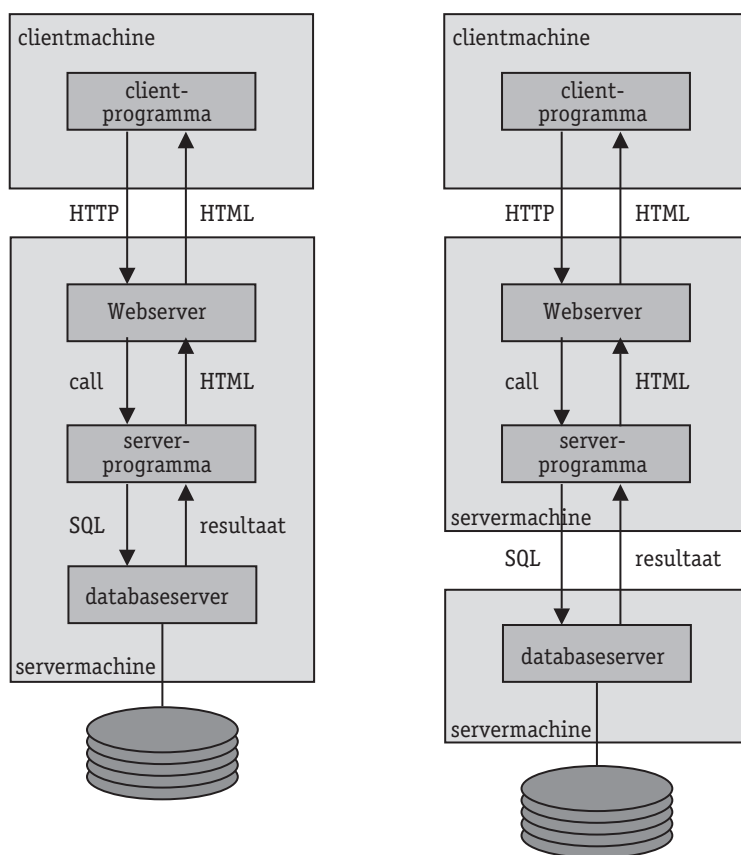


Figuur 1.19 De client/server-architectuur

De derde architectuur is de meeste recente, namelijk de *internetarchitectuur*. De essentie van deze architectuur is dat het programma dat in de client/server-architectuur op de clientmachine draait, in twee delen opgebroken is; zie het linkerdeel van figuur 1.10. Het deel dat zich bezighoudt met de gebruiker, ofwel de userinterface, draait op de clientmachine. Het deel dat met de databaseserver communiceert, ook wel de *programmalogica* genoemd, draait op de servermachine. We noemen deze twee delen in dit boek respectievelijk het client- en het serverprogramma.

In het *clientprogramma* staan waarschijnlijk geen SQL-instructies, maar instructies om code aan te roepen die in het serverprogramma opgenomen zijn. Voor het clientprogramma worden vaak talen gebruikt als HTML, JavaScript en VBScript. De aanroep gaat via internet of een intranet naar de servermachine. Hierbij wordt meestal gebruikgemaakt van het bekende *HTTP* (HyperText Transport Protocol). De aanroep komt binnen bij een *webserver*, die fungeert als een switchboard operator. De webserver weet welke aanroep naar welk serverprogramma gestuurd moet worden.

Vervolgens komt de aanroep bij het serverprogramma aan. Het serverprogramma vuurt dan de benodigde SQL-instructies op de databaseserver af. Veel serverprogramma's draaien onder de controle van Java-applicatieservers, zoals JBoss en WebSphere van IBM.



Figuur 1.20 De internetarchitectuur

De resultaten van de SQL-instructies worden weer door de databaseserver teruggegeven. Het serverprogramma vertaalt dit SQL-resultaat op een of andere wijze naar een HTML-pagina en geeft die pagina door aan de webserver. En de webserver weet, als switchboard operator, naar welk clientprogramma het HTML-antwoord teruggestuurd moet worden.

In het rechterdeel van figuur 1.20 staat een variant van de internetarchitectuur. Hierbij zijn ook het serverprogramma en de databaseserver op verschillende servermachines geplaatst.

Voor de programmeur, die verantwoordelijk is voor het bedenken en schrijven van het programma en de SQL-instructies, is het feit dat de databaseserver en de database remote zijn, volledig transparant (dat wil zeggen: onzichtbaar). Maar het is niet irrelevant, want voor sommige taaltechnische of efficiëntieaspecten van SQL is het wel degelijk belangrijk om te weten of het programma in een monolitische, client/server- dan wel internetarchitectuur ingezet wordt. We gaan in dit boek uit van eerstgenoemde, maar zullen, wanneer dat nodig is, bespreken wat de invloed van client/server of internet kan zijn.

1.8 Transactiedatabases en datawarehouses

Gegevens in een database kunnen voor zeer uiteenlopende doeleinden gebruikt worden. De eerste databases waren voornamelijk ontworpen voor het opslaan van *operationele gegevens*. We geven hier twee voorbeelden van. Banken houden bijvoorbeeld bij welke bankrekeninghouders er zijn, waar ze wonen, wat hun saldo is, en natuurlijk wordt van elke overboeking geregistreerd wanneer die plaatsvond, wat het bedrag was en welke twee bankrekeningnummers het betrof. De afschriften die we periodiek van de banken krijgen, zijn waarschijnlijk verslagen (misschien gegenereerd met SQL) van dergelijke overboekingen. Maar ook luchtvaartmaatschappijen hebben databases opgebouwd die door de jaren heen met zeer veel operationele gegevens gevuld zijn. Zij verzamelen bijvoorbeeld informatie over welke klant wanneer met welke vlucht waarnaartoe is gevlogen. Ook dit zijn zeer veel operationele gegevens.

Databases met operationele gegevens worden ontwikkeld om de gegevens die vrijkomen bij productieprocessen, te registreren. Die gegevens kunnen gebruikt worden om bijvoorbeeld de voortgang te rapporteren en te controleren, en eventueel om het productieproces te verbeteren of te versnellen. Stelt u zich eens voor dat alle overboekingen bij de bank nog met de hand verwerkt zouden worden en uw bankrekeninggegevens in één groot boek zouden worden bijgehouden. Hoe lang zou het wel niet duren voordat uw overboeking inderdaad gerealiseerd zou zijn? Gezien de grootte van banken is dit tegenwoordig niet meer te doen. Databases zijn hier onmisbaar.

We noemen databases die voornamelijk ontworpen en geïmplementeerd zijn om operationele gegevens vast te leggen *transactie*-, *operationele* of *productie*databases. Navenant worden de bijbehorende programma's *transactie*-, *operationele* of *productie*programma's genoemd.

Later zijn databases ook voor andere doeleinden ingezet. Gegevens werden steeds meer gebruikt om overzichtsrapporten te maken. Voorbeelden hiervan zijn: hoeveel reizigers hebben wij de afgelopen maanden vervoerd van Londen naar Parijs? Of geef per regio het aantal verkochte producten in het lopende jaar. Gebruikers ontvangen deze rapporten bijvoorbeeld elke maandagochtend op hun bureau of via e-mail. U zult in dit boek zien dat SQL zeer veel mogelijkheden biedt om rapporten te creëren. Initieel werden deze rap-

porten periodiek gecreëerd en op momenten dat het de verwerking van de transactieprogramma's niet verstoort, bijvoorbeeld op zondag of midden in de nacht.

Recentelijker gingen gebruikers, door de komst van de pc, hogere eisen aan rapportage stellen. Ten eerste groeide de vraag naar online rapporten. Online rapporten worden gecreëerd op het moment dat er door de gebruiker om gevraagd wordt. Ten tweede ontstond de behoefte om zelf nieuwe rapporten te creëren. Om de verstoring op de transactiedatabases tot een minimum te beperken, worden hier aparte databases voor gebouwd. Deze databases worden dan periodiek met gegevens uit de transactiedatabases gevuld en worden voornamelijk voor online rapportage ingezet. Degelijke databases worden *data-warehouses* genoemd.

Bill Inmon definieert een datawarehouse als volgt (zie tevens [GILL96]):

Een datawarehouse is een onderwerpgeoriënteerde, geïntegreerde, statische en tijdgebonden verzameling van gegevens ter ondersteuning van beslissingen op managementniveau.

In deze definitie komen vier belangrijke termen voor. Met *onderwerpgeoriënteerd* wordt bijvoorbeeld bedoeld dat alle klantgegevens bij elkaar liggen opgeslagen en dat productgegevens bij elkaar staan. Het tegenoverstelde is *programmeergeoriënteerd*. Hierbij bevat een database gegevens die relevant zijn voor een bepaald programma. Het kan dan gebeuren dat klantgegevens over twee of meer databases verspreid zijn. Dit zou rapportage zeer complex maken, want dan moeten de gegevens voor een rapport uit meerdere databases gehaald worden.

Kortweg, de term *geïntegreerd* heeft te maken met het consistent coderen van gegevens, zodat ze geïntegreerd opgehaald en gecombineerd kunnen worden.

Een datawarehouse is een *statische* database. Wanneer een database voornamelijk voor het genereren van rapporten gebruikt wordt, dan willen de gebruikers ongetwijfeld dat de inhoud niet continu verandert. Stelt u zich eens voor dat twee gebruikers een vergadering moeten bijwonen en beiden daarvoor eerst de database raadplegen om te kijken wat de omzetcijfers in een bepaalde regio waren. Stelt u zich tevens voor dat tussen de vragen van de twee gebruikers een tijdsduur van tien minuten ligt en in die tien minuten wordt de database gewijzigd. De gebruikers zullen afwijkende gegevens naar de vergadering meenemen. Om dit probleem te voorkomen, wordt een datawarehouse periodiek bijgewerkt. Ieder weekeinde of elke avond worden nieuwe gegevens toegevoegd.

De *tijdgebondenheid* van een datawarehouse is ook een belangrijk aspect. Normaliter wordt getracht transactiedatabases zo klein mogelijk te houden. Want hoe kleiner de database, hoe beter dat voor de verwerkingssnelheid van SQL is. Een veelgebruikte manier om databases klein te houden, is door oude gegevens te verwijderen. Die oude gegevens kunnen dan op magneetband of dvd geplaatst worden, zodat ze later nog eens gebruikt kunnen worden. Gebruikers van datawarehouses verwachten juist veel historische gegevens. Ze willen bijvoorbeeld uitzoeken of de verkoop van boottickets naar Londen de afgelopen tien jaar veranderd is. Of ze willen weten wat de invloed van het weer op de verkoop van bier is, en willen daarvoor gegevens van de laatste vijf jaar gebruiken. Het effect is dat veel historische gegevens opgenomen moeten worden en dat ze bijna allemaal tijdgebonden zijn. Omwille hiervan zijn datawarehouses groot tot zeer groot. Een datawarehouse van 1 terabyte of meer is niet ongewoon.

Opmerking

Bij het ontwerpen van een database moet van tevoren bepaald worden hoe deze gebruikt gaat worden. Wordt het een transactiedatabase of een datawarehouse? In dit boek zullen we op die plaatsen waar dat relevant is, dit onderscheid maken.

1.9 Standaardisatie van SQL

Zoals reeds vermeld, heeft elke SQL-databaseserver een eigen dialect. Al deze dialecten lijken wel op elkaar, maar zijn niet honderd procent identiek. Ze verschillen ten eerste in de instructies die ze ondersteunen (sommige producten kennen wat meer SQL-instructies dan andere), ten tweede kunnen de mogelijkheden van instructies uiteenlopen en ten derde ondersteunen twee producten soms dezelfde instructie, maar wordt die instructie door hen anders verwerkt.

Om de verschillen tussen de vele databaseservers van diverse leveranciers te beperken, werd al vroegtijdig besloten een standaard voor SQL te definiëren. De achterliggende gedachte was dat als de databaseservers te veel uit elkaar zouden groeien, de acceptatie van SQL minder groot zou zijn. Een standaard zou ervoor zorgen dat een programma met SQL-instructies eenvoudiger over te zetten zou zijn van de ene naar de andere database-server.

Rond 1983 is de ISO (International Standardization Organization) in samenwerking met het ANSI (American National Standards Institute) begonnen met de ontwikkeling van een standaard voor SQL. ISO is de meest vooraanstaande, internationaal georiënteerde organisatie voor standaardisatie en heeft als doel het bevorderen van internationale, regionale en nationale standaarden. Veel landen kennen lokale vertegenwoordigingen van ISO, het ANSI is de Amerikaanse tak ervan.

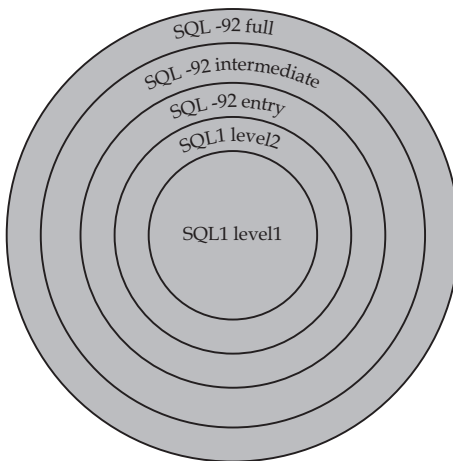
Na veel vergaderingen en enkele valse starts verscheen in 1986 de eerste ANSI-versie van de SQL-standaard. Deze is beschreven in het document ANSI X3.135-1986 'Database Language SQL'. Deze SQL-86-standaard wordt informeel SQL1 genoemd. Een jaar later werd de ISO-versie uitgebracht met de naam ISO 9075-1987 'Database Language SQL'; zie [ISO87]. Dit document is ontwikkeld onder de verantwoordelijkheid van Technical Committee TC97. Het taakgebied van TC97 is omschreven als Computing and Information Processing. Subcommissie SC21 heeft de standaard laten ontwikkelen. De ISO- en ANSI-varianten van SQL1 ofwel SQL-86 zijn identiek.

SQL1 is opgebouwd uit twee niveaus. Niveau 2 beslaat het gehele document. Niveau 1 is een deelverzameling van niveau 2, wat wil zeggen dat niet alle specificaties van SQL1 tot niveau 1 behoren. Als een leverancier meldt dat zijn database-server aan deze standaard voldoet, dan moet tevens het ondersteunde niveau worden opgegeven. Het opdelen in niveaus was gedaan om ondersteuning voor en acceptatie van SQL1 te versnellen. Leveranciers konden de standaard daardoor gefaseerd ondersteunen, eerst niveau 1 en vervolgens doorgroeien naar niveau 2.

De SQL1-standaard is op het gebied van integriteit zeer matig. Om die reden is zij in 1989 uitgebreid met onder andere primaire en refererende sleutels. Deze versie van de SQL-standaard wordt SQL-89 genoemd. Het bijbehorende ISO-document heet heel toepasselijk ISO 9075:1989 'Database Language SQL with Integrity Enhancements'. De ANSI-versie kwam tegelijkertijd uit.

Direct na het uitkomen van SQL1 in 1987 is begonnen met de ontwikkeling van een nieuwe versie van de standaard; zie [ISO92]. Deze geplande opvolger van SQL-89 had als projectnaam SQL2. Er was bij aanvang nog geen publicatiedatum bekend, dus vandaar de simpele naam. In feite werd er tegelijk aan SQL-89 en SQL2 gewerkt. Uiteindelijk verscheen de opvolger in 1992, die de toenmalige standaard (SQL-89) verving. De nieuwe standaard SQL-92 is een uitbreiding op de SQL1-standaard. Veel nieuwe instructies en mogelijkheden van bestaande instructies zijn toegevoegd. Een complete beschrijving van SQL-92 is te vinden in [DATE97].

Net als SQL1 kent SQL-92 niveaus (*levels*). Deze hebben echter geen nummers maar namen: *entry*, *intermediate* en *full*. Full SQL is de volledige standaard. Intermediate SQL is qua functionaliteit een deelverzameling van full SQL en entry SQL is daar weer een deelverzameling van. Entry SQL kan ruwweg vergeleken worden met SQL1 niveau 2 uitgebreid met enkele specificaties. Alle niveaus samen kunnen als uienringen worden voorgesteld; zie figuur 1.21. Hierbij stelt een ring een bepaalde hoeveelheid functionaliteit voor. En hoe groter de ring, hoe meer functionaliteit binnen dat niveau gedefinieerd is. Als een ring binnen een andere ring valt, betekent dit dat het gaat om een deelverzameling aan functionaliteit. Op het moment van schrijven ondersteunen de meeste SQL-producten full SQL-92.



Figuur 1.21 De verschillende niveaus van SQL1 en SQL-92 voorgesteld als ringen

Sinds de publicatie van SQL-92 zijn er diverse extra documenten verschenen waarmee de mogelijkheden van de taal uitgebreid werden. In 1995 werd *SQL/CLI* (Call Level Interface) gepubliceerd. Deze werd later hernoemd tot CLI-95. Het jaar daarop verscheen *SQL/PSM* (Persistent Stored Modules) ofwel PSM-96. De meest recente toevoeging, PSM-96, beschrijft een hoeveelheid functionaliteit om zogenaamde stored procedures te creëren. Dit concept komt in hoofdstuk 29 uitgebreid aan bod. Twee jaar na PSM-96 volgde *SQL/OLB* (Object Language Bindings) ofwel OLB-98. Dit document beschrijft hoe SQL-instructies binnen de programmeertaal Java opgenomen moeten worden.

Nog voordat SQL-92 klaar was, werd reeds met de opvolger begonnen. Deze had uiteraard als projectnaam SQL3. In 1999 verscheen deze standaard onder de naam SQL:1999. Om meer in lijn te zijn met de namen van andere ISO-standaarden werd gekozen voor een dubbele punt en niet een streepje zoals bij de vorige versies. En vanwege de problema-

tiel rond het jaar 2000 werd besloten om 1999 niet af te korten tot 99. Zie [GULU99], [MELT01] en [MELT03] voor uitvoerige beschrijvingen van deze standaard.

Toen SQL:1999 verscheen bestond het uit vijf delen: SQL/Framework, SQL/Foundation, SQL/CLI, SQL/PSM en SQL/Bindings. Later zijn daar onder andere aan toegevoegd SQL/OLAP, SQL/MED (Management of External Data), SQL/OLB, SQL/Schemata en SQL/JRT (Routines and Types using the Java Programming Language) en SQL/XML (XML-Related Specifications). De huidige SQL-standaard van ISO bestaat dus uit een reeks documenten die allemaal met de ISO-code 9075 beginnen. Bijvoorbeeld, de volledige aanduiding van het SQL/Framework is ISO/IEC 9075-1:2003.

Naast de 9075 documenten bestaat er nog een groep documenten die zich op SQL richt. Deze groep documenten wordt meestal aangeduid met de term SQL/MM. Deze afkorting staat voor SQL Multimedia en Application Packages. Al deze documenten dragen de ISO-code 13249. SQL/MM bestaat uit vijf delen. SQL/MM deel 1 is het SQL/MM Framework, deel 2 richt zich op tekstretrieval (het werken met teksten), deel 3 is gewijd aan ruimtelijke toepassingen, deel 4 aan still images (zoals foto's) en deel 5 op datamining (het zoeken naar trends en patronen in de gegevens).

Dus vanaf SQL:1999 is de SQL-standaard als het ware in aparte documenten verdeeld. Twee daarvan vormen de basis: SQL/Framework en SQL/Foundation. De andere documenten, zoals SQL/MED, SQL/MM en SQL/JRT, kunnen als aparte, op zichzelf staande documenten gezien worden die SQL/Framework en SQL/Foundation als fundament hebben. Het voordeel van deze aanpak is dat van die extra documenten onafhankelijk van elkaar nieuwe versies uitgebracht kunnen worden. Het is niet meer noodzakelijk dat van alle functionaliteit tegelijkertijd nieuwe documenten gepubliceerd worden. Bijvoorbeeld, als het noodzakelijk is om de mogelijkheden voor language bindings uit te breiden, kan gewoonweg een nieuwe versie van SQL/Bindings uitgebracht worden.

Dit documentsgewijs uitgeven blijkt ook als we kijken naar wat de recente versies van de verschillende documenten zijn. Bijvoorbeeld, in 2008 zijn de meest recente versies van SQL/Framework en SQL/Foundation verschenen. Maar voor SQL/MM Still Images is dat 2003 en voor SQL/PSM is dat 2008. In de literatuur wordt wel eens naar de gehele groep documenten die in 2003 is verschenen gerefereerd met de afkorting SQL:2003 en naar die uit 2008 met SQL:2008

De belangrijkste technologie die ooit afgeleid is van het werk van de Open SQL Access Group en dus van SQL/CLI, is ODBC (Open DataBase Connectivity) van Microsoft.

Er is dus al veel tijd en geld gestoken in het standaardiseren van SQL. Maar is een standaard eigenlijk wel belangrijk? Als antwoord op deze vraag geven we enkele praktische voordelen als alle databaseservers exact dezelfde gestandaardiseerde databasetaal zouden ondersteunen:

- Verhoogde overdraagbaarheid: een programma kan voor de ene databaseserver ontwikkeld worden en zonder veel veranderingen draaien op de andere.
- Verhoogde uitwisselbaarheid: databaseservers kunnen onderling communiceren, omdat ze dezelfde taal spreken. Het wordt tevens eenvoudiger om programma's verschillende databases te laten benaderen.
- Verminderde opleidingskosten: programmeurs kunnen sneller van databaseserver veranderen, omdat de taal hetzelfde blijft. Het is niet nodig een nieuwe databasetaal te leren.

- Verlengde levensduur: talen die gestandaardiseerd zijn, hebben de neiging langer te blijven bestaan, en dus ook de programma's die erin geschreven zijn. COBOL is hier een prachtig voorbeeld van.

MySQL ondersteunt een redelijk deel van de SQL-92- en SQL:1999-standaarden. Zeker vanaf versie 5 is het op dit gebied fors uitgebreid. De laatste jaren heeft de MySQL-organisatie de doelstelling om het product zoveel mogelijk conform de standaard te implementeren. Of met andere woorden, als de MySQL-organisatie iets nieuws wil toevoegen aan MySQL en in de standaard staat hiervoor iets beschreven, dan conformeert ze zich aan die standaard.

1.10 De markt van SQL-databaseservers

De taal SQL is al in zeer veel producten op een of andere wijze geïmplementeerd. SQL-databaseservers zijn verkrijgbaar voor alle besturingssystemen en op alle soorten machines, van de kleinste mobiele telefoon tot de grootste multiprocessormachine. In tabel 1.1 zijn van een aantal leveranciers de namen van hun SQL-producten opgenomen. In dit boek zullen we op enkele plaatsen naar deze producten verwijzen. Voor gedetailleerde informatie verwijzen we naar de leveranciers zelf. Elk van deze leveranciers heeft een website waar meer informatie beschikbaar is.

Leverancier	SQL-Producten
Action Corporation	Ingres, VectorWise
Apache Software Foundation	Apache Derby
Birdstep Technology	Birdstep RDM Server
Borland	InterBase
Centura Software	SQLBase
Cincom	Supra Server SQL
Computer Associates	CA-Datcom, CA-IDMS
Daffodil Software	Daffodil DB
EMC ²	Greenplum Database
Empress Software	Empress RDBMS
Enea	Polyhedra DBMS
EXASol	EXASolution
Faircom	c-treeSQL
FileMaker	FileMaker
Firebird	Firebird
FirstSQL	FirstSQL/J
Frontbase	Frontbase
H2 Database Engine	H2 Database Engine

Tabel 1.1 Overzicht van bekende SQL-databaseservers en hun leveranciers

Vervolg →

vervolg tabel 1.1

Leverancier	SQL-Producten
Hewlett Packard	Vertica Analytics System
Hughes Technologies	Mini SQL (alias mSQL)
HSQldb	HSQldb (alias Hypersonic SQL)
IBM	DB2, Netezza, Informix, Red Brick Warehouse, Cloudscape JDBMS, SolidDB, UniData (voorheen van Ardent Software)
InfoBright	InfoBright
InstantDB	InstantDB
InterSystems	Caché
Kognitio	WX2
Korea Computer Communications	UniSQL
LucidDB	LucidDB
Machine Independent Software Corporation	CQL++
McKoi	McKoi SQL Database
Micro Focus	XDB
Microsoft	Microsoft SQL Server, Microsoft Access
MonetDB	MonetDB/SQL
MySQL (Oracle)	MySQL
Nexus Database Systems	NexusDB
Ocelot Computer Services	Ocelot
Oracle	Oracle 11g, Oracle Rdb, TimesTen
ParAccel	ParAccel Analytic Database
Pervasive Software	PSQL
PointBase	PointBase Embedded, PointBase Server
PostgreSQL	PostgreSQL
Progress Software	Progress
QuadBase	QuadBase-SQL
Sand Technology	CDBMS
SAP	Sybase Adaptive Server, Sybase Adaptive Server Anywhere, Sybase Adaptive Server IQ, Advantage Database Server (voorheen van Extended Systems), SapDB
Siemens	SESAM/SQL-Server, UDS/SQL
Software AG	Adabas D Server
SQLite	SQLite

Tabel 1.1 Overzicht van bekende SQL-databaseservers en hun leveranciers

Vervolg →

vervolg tabel 1.1

Leverancier	SQL-Producten
StreamBase Systems	StreamBase
Teradata	Teradata en Aster database
ThinkSQL	ThinkSQL RDBMS
TigerLogic	D3
Tigris	Axion
TinySQL	TinySQL
Unify Corporation	Unify Data Server
Upright Database Technology	Mimer SQL Engine

Tabel 1.1 Overzicht van bekende SQL-databaseservers en hun leveranciers

1.11 Welk SQL-dialect?

De vorige paragraaf bevat een lijst met databaseservers die SQL ondersteunen. Zoals reeds in het voorwoord aangegeven, lijken al deze implementaties van SQL veel op elkaar, maar vertonen ze helaas ook verschillen. Ook de internationale SQL-standaarden kunnen als dialecten gezien worden, want niemand heeft deze momenteel volledig geïmplementeerd. Er bestaan dus verschillende *SQL-dialecten*.

De vraag die u dan misschien hebt is: Welk SQL-dialect wordt in dit boek beschreven? Deze vraag is niet eenvoudig te beantwoorden. We hebben namelijk niet voor het dialect van één specifiek product gekozen, omdat dit boek probeert SQL zo algemeen mogelijk te beschrijven. We hebben ook niet voor de dialecten van de SQL1-, SQL-92- of SQL:2003-standaard gekozen, omdat de eerste te 'klein' is en de laatste nog door niemand ondersteund wordt. Er is zelfs niet voor het dialect van MySQL gekozen (het product waar we in dit boek van uitgaan). Het ligt iets complexer.

Om de praktische waarde van dit boek te verhogen, worden primair de instructies en mogelijkheden beschreven die door alle dominante SQL-producten ondersteund worden. Dit maakt het boek algemeen toepasbaar. Na het lezen van dit boek moet u in staat zijn om met elk willekeurig SQL-product te kunnen werken. Dit boek is dus gericht op algemeen SQL, ofwel SQL zoals geïmplementeerd in de meeste SQL-producten.

Hoe zit dat nu met MySQL? MySQL wordt in dit boek aangeraden, maar betekent dit dat alle voorbeelden en opdrachten in dit boek met MySQL uitgevoerd kunnen worden? Helaas is het antwoord daarop nee! Er is voor MySQL gekozen, omdat het populair is, omdat het gemakkelijk te installeren is, en omdat het beschikbaar is voor de meeste moderne besturingssystemen, maar vooral omdat het SQL-dialect heel rijk is. De meeste, maar niet alle, voorbeelden en opdrachten in dit boek kunnen met dit krachtige product uitgevoerd worden.

Op diverse plaatsen wordt met het onderstaande koffersymbool aangegeven of bepaalde instructies tussen SQL-producten overdraagbaar zijn en of deze door MySQL ondersteund worden.



Overdraagbaarheid

Hier staat een advies of opmerking over de overdraagbaarheid van een SQL-instructie.

1.12 De structuur van het boek

We sluiten dit hoofdstuk af met een beschrijving van de structuur van dit boek. Vanwege het grote aantal hoofdstukken is het boek in delen opgesplitst.

Het eerste deel bestaat uit een aantal inleidende onderwerpen waarvan dit hoofdstuk er een is. Hoofdstuk 2 bevat een uitvoerige beschrijving van een database die bij de meeste voorbeelden en opgaven wordt gebruikt. Deze database vormt de basis voor een wedstrijd-administratie van een tennisvereniging. Hoofdstuk 4 geeft een algemeen overzicht van SQL. Na het lezen van dit hoofdstuk hebt u een globaal beeld van de mogelijkheden van deze taal en een goed beeld van wat u in dit boek te wachten staat.

Deel twee is volledig gericht op het raadplegen en muteren van tabellen. Ze is grotendeels aan de zogenaamde SELECT-instructie gewijd. Met veel voorbeelden worden alle mogelijkheden toegelicht. Er wordt in dit boek veel aandacht aan deze SELECT-instructie geschonken, omdat het de instructie is die u in de praktijk het meest zult gebruiken en omdat een groot aantal andere instructies hierop gebaseerd is. Het laatste hoofdstuk in dit deel beschrijft hoe bestaande databasegegevens gewijzigd en/of verwijderd kunnen worden en hoe nieuwe rijen aan tabellen toegevoegd kunnen worden.

Deel drie behandelt het creëren van *databaseobjecten*. De term databaseobject is de verzamelnaam voor alle objecten waaruit een database is opgebouwd. De objecten tabel, primaire, alternatieve en refererende sleutel, index en view worden onder andere behandeld. Dit deel behandelt tevens het beveiligen van gegevens.

Deel vier beschrijft stored procedures, stored functions en triggers. Stored procedures en stored functions zijn brokken SQL-code die in de database opgeslagen liggen en vanuit programma's aangeroepen kunnen worden. Triggers zijn ook brokken code, maar deze worden door de databaseserver zelf aangeroepen, bijvoorbeeld om controles uit te voeren of om gegevens automatisch te laten bijwerken.

Deel vijf gaat in op het programmeren met SQL. Met behulp van embedded SQL wordt uitgelegd hoe men met SQL plus een programmeertaal programma's kan ontwikkelen. Embedded SQL is een vorm van programmeren waarbij SQL-instructies op een speciale manier binnen programma's opgenomen worden. Deze programma's kunnen geschreven zijn in talen als C, COBOL of Pascal. In dit deel worden ook de volgende begrippen toegelicht: transactie, savepoint, terugdraaien van transacties, isolation level en repeatable read. En omdat bij het programmeren performance een belangrijk aspect is, is een hoofdstuk gericht op hoe, door het herformuleren van een SQL-instructie, de verwerkingstijd ervan verkort kan worden.

Register

Symbolen

+ -operator 103

||-operator 101

A

ABS-functie 56

actieve databaseserver 625

ADDTIME-functie 106

afgeleide kolom 437

afgeleide tabel 56, 519

aflopend sorteren 329

AFTER 628

aggregatiefunctie 109, 252

aggregatieniveau van een resultaat 289

aggregeren van gegevens 287

alfanumeriek datatype 425

maximale lengte 426

variabele lengte 426

vaste lengte 426

alfanumerieke constante 72

alfanumerieke expressie

samengestelde 101

algemene join 158

algemene tabelexpressie 363

alias 138

ALL-databasebevoegdheid 548

ALL in de SELECT-component 250

ALL-operator 225, 699

ALL-tabelbevoegdheid 544

ALTER-databasebevoegdheid 547

ALTER DATABASE-instructie 539

alternatieve sleutel 9, 445

in de voorbeelddatabase 39

ALTER ROUTINE-databasebevoegdheid 548

ALTER-sequence-bevoegdheid 573

ALTER SEQUENCE-instructie 572

ALTER-tabelbevoegdheid 544

ALTER TABLE-instructie 474, 475

CONVERT 475

ALTER USER-instructie 544

American National Standards Institute 26

analytische databaseserver 21

AND-operator 180

ANSI 26

ANSI_QUOTES-instelling 434

ANY-operator 225, 699

applicable null-waarde 50

architectuur

client/server 22

internet 23

monolitische 21

array 16

AS 138

ASC(ending) 329

ASCII-karakterset 329, 459

assertion 627

associatieve wet 360

asterisk 245

atomaire waarde 7

attribuut van XML 399

autocommit 673

AUTOCOMMIT-systeemvariabele 674

AVG-functie 109, 263

B

basisbegrippen 69

basistabel 519

batch-invoerprogramma 14

batch-rapportageprogramma 14

BEFORE 628

BEGIN DECLARE SECTION-instructie 649

benaderende waarde 423

beperkingen van views 527

bereik van integer-datatype 422

berekenen van subtotaal 347

berekenen van totaal 347

bestand 493

BESTUURSLEDEN-tabel 36

BETWEEN-operator 197, 695

beveiligen van gegevens 57, 535, 541

beveiliging 615

beveiliging van en met views 559

bevoegdheid 45, 57

BIGINT-datatype 422

bills-of-material problem 372

binder 647

binding style 13

bitmapindex 516

blob-datatype 427
 body van een stored procedure 585
 BOETES-tabel 36
 bondsnummer 33
 boolean-constante 75
 boolean search 216
 boolean search-operator 217
 boom 495
 bouwer van databaseobject 575
 Boyce-Codd-normaalvorm 484
 Boyce R.F. 20
 browsing 495
 B-tree index 499
 buffer 686
 business intelligence 16

C

C 670
 C# 3
 CACHE-sequence-optie 567
 CALL-instructie 582, 596
 Call Level Interface 13, 27
 cardinaliteit van de kolom 509
 cardinaliteit van de tabel 509
 cartesisch product van tabellen 135, 145, 156
 CASCADE refererende actie 452
 case-expressie 85
 CASE-instructie 593
 cast-expressie 95
 casting 95
 catalogus 61
 catalogustabel 61
 catalogusview
 COLUMN__AUTHS 554
 COLUMNS 390, 438
 COLUMNS__IN__INDEX 517
 INDEXES 517
 TABLE__AUTHS 554, 555
 TABLES 438
 USER__AUTHS 84
 USERS 554
 VIEWS 525
 CHARACTER-datatype 427
 character encoding 459
 CHARACTER__SET__CLIENT 471
 CHARACTER__SET__CONNECTION 471
 CHARACTER__SET__DATABASE 471
 CHARACTER__SET__DIR 471
 CHARACTER__SET__RESULTS 471

CHARACTER__SET__SERVER 471
 CHARACTER__SET__SYSTEM 471
 CHARACTER VARYING-datatype 427
 CHAR-datatype 425
 CHARSET-functie 467
 CHAR VARYING-datatype 427
 check-integriteitsregel 454
 CLI 13
 clientmachine 22
 clientprogramma 23
 client/server-architectuur 22
 CLOSE CURSOR-instructie 610
 CLOSE-instructie 662
 COALESCE-functie 92
 Codd, E. F. 16, 17, 50, 701
 code character set 459
 code page 459
 coderingsformaat 459
 coercibility 469
 COERCIBILITY-functie 470
 collating sequence 460
 collation 172, 329, 459, 460
 coercibility 469
 groeperen van rijen 468
 sorteren van waarden 468
 toekennen aan kolom 464
 COLLATION__CONNECTION 471
 COLLATION__DATABASE 471
 COLLATION-functie 467
 COLLATION__SERVER 471
 COLUMN__AUTHS-catalogusview 554
 COLUMNS-catalogusview 390, 438
 COLUMNS__IN__INDEX-catalogusview 517
 combineren van grouping sets 315
 COMMENT-kolom-optie 436
 COMMIT-instructie 675
 common table expression 363
 commutatieve wet 359
 compiler 645
 componenten van de SELECT-instructie 114
 composite primary key 443
 compound statement 586
 computed columns 437
 CONCAT-functie 92
 concurrency 685
 conditie 167, 168, 605
 evaluatie 182
 conditie met ontkenning 242
 conjuncte populaties 146

connectie 46
 CONNECT-instructie 654
 consistentie van gegevens 5, 441
 constante 69
 alfanumerieke 72
 boolean 75
 datatype 69
 datum 73
 decimal 71
 drijvende komma 71
 float 71
 hexadecimal 75
 integer 70
 temporal 73
 tijd 73
 timestamp 74
 constraint 7, 441
 CONTINUE-handler 604
 correctheid van gegevens 441
 COUNT-functie 109, 254
 courante database 47, 48
 CREATE-databasebevoegdheid 547
 CREATE DATABASE-instructie 47, 537, 538
 CREATE INDEX-instructie 55, 503
 CREATE PROCEDURE-instructie 582
 CREATE ROLE-instructie 552
 CREATE ROUTINE-databasebevoegdheid 548
 CREATE SCHEMA-instructie 576
 CREATE SEQUENCE-instructie 562
 CREATE TABLE-instructie 48, 419
 CREATE TEMPORARY TABLES-databasebevoegdheid 547
 CREATE TRIGGER-instructie 626
 CREATE USER-gebruikersbevoegdheid 550
 CREATE USER-instructie 45, 542
 CREATE VIEW-databasebevoegdheid 547
 CREATE VIEW-instructie 56, 519
 creëren van databases 46
 creëren van indexen 55, 503
 creëren van rollen 552
 creëren van SQL-gebruikers 542
 creëren van tabellen 48, 419
 creëren van views 519
 cross referential integrity 480
 CUBE 313
 cumulatieve waarden 301
 CURRENT__DATE 84
 CURRENT__TIME 84
 CURRENT__TIMESTAMP 84

CURRENT__USER 84
 cursor 607, 659
 insensitive 666
 sensitive 666
 cursor stability isolation level 689
 CYCLE-sequence-optie 566

D

database 3, 46, 575
 courante 47
 creëren 46
 DATABASE__AUTHS-catalogusview 555
 databasebevoegdheid 541, 547
 ALL 548
 ALTER 547
 ALTER ROUTINE 548
 CREATE 547
 CREATE ROUTINE 548
 CREATE TEMPORARY TABLES 547
 CREATE VIEW 547
 DELETE 547
 DROP 547
 EXECUTE ROUTINE 548
 INDEX 547
 INSERT 547
 LOCK TABLES 548
 REFERENCES 547
 SELECT 547
 UPDATE 547
 database management systeem 3
 databaseobject 32, 58, 417
 procedureel 579
 database procedure 581
 Database Request Module 647
 databaseserver 3
 databasestructuur 483
 databasetaal 3, 4
 Data Control Language 60
 Data Definition Language 60
 Data Manipulation Language 60
 datamining 17
 datatype 69
 kiezen voor een kolom 491
 van een kolom 420, 422
 van een rijexpressie 111
 datawarehouse 25
 Date, Chris J. 3
 DATE-datatype 427
 DATEDIFF-functie 94

- DATE-functie 92
- datum
 - rekenen met 103
- datumconstante 73
 - bereik 73
- datumexpressie
 - samengestelde 103
- DAYNAME-functie 93
- DAYOFYEAR-functie 93
- DB2 20
- dbms 3
- DBRM 647
- DCL-instructie 60
 - embedded SQL 643
- DDL-instructie 60
 - embedded SQL 643
- deadlock 686
- DEC-datatype 423
- decimal-constante 71
- decimal-datatype 423
- declaratieve taal 10
- DECLARE CONDITION-instructie 605
- DECLARE CURSOR-instructie 608, 660
 - FOR-component 667
- DECLARE HANDLER-instructie 603
- DECLARE VARIABLE-instructie 587
- deelverzameling 146
- DEFAULT 436
- DEFAULT-functie 436
- default-karakterset
 - wijzigen 475
- DEFAULT-kolom-optie 435
- default-waarde 435
- definiëren van gebruikersvariabelen 83
- definities van SQL-instructies 65
- DELETE-databasebevoegdheid 547
- DELETE-instructie 54, 397, 669
- DELETE-tabelbevoegdheid 544
- denormaliseren 489
- DENSE _ RANK-functie 275
- DESC(ending) 329
- determinant 484
- dirty read 682
- dirty read isolation level 689
- DISCONNECT-instructie 655
- disjuncte populaties 146
- DISTINCT 140, 247, 697
- DISTINCT (GROUP BY) 297
- distributed van waarden 509
- distributed wet 360
- DML-instructie 60
- doelgroep van boek xx
- doorsnede 349
- double precision float-datatype 424
- downloaden van MySQL 41
- downloaden van SQL-instructies xxi
- drijvende komma-constante 71
- DROP-databasebevoegdheid 547
- DROP DATABASE-instructie 58, 540
- DROP FUNCTION-instructie 623
- DROP INDEX-instructie 58, 505
- DROP-instructie 58
- DROP PROCEDURE-instructie 613
- DROP ROLE-instructie 553
- DROP SCHEMA-instructie 577
- DROP SEQUENCE-instructie 572
- DROP TABLE-instructie 58, 474
- DROP TRIGGER-instructie 638
- DROP USER-instructie 543
- DROP VIEW-instructie 58, 525

E

- EBCDIC
 - karakterset 459
- EBCDIC-karakterset 329
- ECA-rule 627
- eenvoudige vergelijking 168
- eerste integriteitsregel 445
- eigenaar 133, 575
- eigenaar van tabel 133
- element van XML 399
- Elmasri R. 3
- embedded SQL
 - algemene regels 643
 - DCL-instructie 643
 - DDL-instructie 643
 - hostvariabele 648
 - mutatie-instructie 644
- encoding scheme 459
- END DECLARE SECTION-instructie 649
- END-EXEC 643
- enkelvoudige expressie 77
- entiteit integriteitsregel 445
- entry SQL 27
- equi-join 158
- escape-teken 202
- evaluatie van een conditie 182
- EXCEPT ALL-operator 356, 697

EXCEPT-operator 353
 exclusive lock 686
 EXCLUSIVE lock-type 688
 EXEC SQL 643
 executeerbare instructie 651
 EXECUTE PROCEDURE-instructie 582
 EXECUTE ROUTINE-databasebevoegdheid 548
 EXISTS-operator 222
 expliciete casting 95
 expliciete join 144
 expressie 76
 datatype 94
 datumexpressie 103
 PREVIOUS VALUE FOR 570
 rij 77, 110
 samengestelde 98
 sequence 562
 tabel 77, 111
 waarde 77
 expressielijst 184
 EXTRACTVALUE-functie 403

F

FALSE 75
 FETCH ABSOLUTE 664
 FETCH CURSOR-instructie 609
 FETCH-instructie 661, 662
 FETCH RELATIVE 664
 fibonaccireeks 591
 file 493
 float-constante 71
 float-datatype 423
 lengte 424
 FLOAT-datatype 423
 floating point 71
 FOR-component 667
 foreign key 9
 FOREIGN KEY 447
 forward-chaining rule 627
 full SQL 27
 fulltext index 212
 functie
 ABS 56
 ADDTIME 106
 AVG 263
 CHARSET 467
 COALESCE 92
 COERCIBILITY 470
 COLLATION 467

CONCAT 92
 COUNT 254
 DATEDIFF 94
 DAYNAME 93
 DAYOFYEAR 93
 DEFAULT 436
 DENSE __RANK 275
 EXTRACTVALUE 403
 LEFT 92
 MAX 258
 MIN 258
 MOD 92
 MONTHNAME 93
 nesten 92
 OLAP 268
 RANK 275
 ROUND 92
 ROW __NUMBER 269
 scalaire 91
 set 109
 SQRT 92
 STDDEV 267
 SUBSTR 205
 SUM 263
 UCASE 91
 UPDATEXML 414
 VAR 266
 VARIANCE 266
 YEAR 91
 functies van de catalogus 61

G

gastheertaal 641
 gebruiker 45
 gebruiker creëren 542
 gebruikersbevoegdheid 541, 549
 CREATE USER 550
 gebruikersnaam 43, 542
 gebruikersvariabele 82
 geclusterde index 497
 gecorrleerde subquery 178, 234, 236
 gegevens
 operationele 24
 gegevensbeveiliging 57, 483, 535, 541
 gegevensonafhankelijkheid 4
 gegevenspagina 494
 gehele getal 422
 geïndexeerde zoekmethode 495
 geïntegreerde gegevens 25

- gelijke populaties 146
- gemiddelde 263
- genereren van unieke nummers 561
- geoptimaliseerde strategie 499
- gerefereerde tabel 448
- geschiedenis van SQL 19
- graad van een rijexpressie 110
- graad van een tabelexpressie 112
- GRANT-instructie 46, 57, 542, 573, 615
- granulariteit van blokkeren 687
- greater-than-join 158
- Gregoriaanse kalender 73, 103, 108
- groeperen
 - collations 468
- groeperen met CUBE 313
- groeperen met ROLLUP 311
- groeperen op één kolom 288
- groeperen op expressies 293
- groeperen op meerdere kolommen 291
- groeperen van null-waarden 295
- groepering 288
- groepfunctie 252
- GROUP BY-component 287
 - WITH ROLLUP 303, 305
- grouping sets 306
- grouping sets combineren 315

H

- haakjes bij scalaire expressies 90
- handler 604
- hashindex 514
- HAVING-component 317, 696
- helpfunctie 61
- hexadecimal-constante 75
- historie van boek xxi
- horizontale deelverzameling 245
- horizontale vergelijking 251
- host language 641
- hostvariabele 643
 - embedded SQL 648
 - SQLCODE 649, 657, 662
- HTML 23
- HTTP 23

I

- IBM 19, 20
- idempotentie wet 360
- identieke rijen 250
- IF-instructie 590

- impliciete casting 95
- impliciete join 144
- inapplicable null-waarde 50
- IN BOOLEAN MODE 216
- INCLUDE-instructie 650
- inconsistent read 683
- INCREMENT BY-sequence-optie 565
- index 55
 - fulltext 212
 - geclusterd 497
- INDEX-databasebevoegdheid 547
- INDEXES-catalogusview 517
- INDEX-tabelbevoegdheid 544
- indexvorm
 - bitmap 516
 - B-tree 499
 - hash 514
 - multi-tabel 513
 - selectief 514
 - virtuele kolomindex 513
- informatiedragende namen 434
- INFORMATION_SCHEMA
 - CHARACTER_SETS-tabel 461
 - COLLATIONS-tabel 462
- initialiseren van gebruikersvariabelen 83
- inloggen 43
- inloggen op SQL 654
- Inmon, Bill 25
- inner identity 360
- inner-join 144
- innerselect 125
- IN-operator 183
- IN-operator met subquery 188
- INSENSITIVE 666
- insensitive cursor 666
- INSERT-databasebevoegdheid 547
- INSERT-instructie 50, 389
 - algemene regels 392
 - DEFAULT 436
- INSERT-tabelbevoegdheid 544
- installeren van MySQL 41
- INSTEAD OF 629
- INT-datatype 423
- integer-constante 70
- integer-datatype 422
 - bereik 422
- INTEGER-datatype 422
- integere database 441

- ul style="list-style-type: none; padding-left: 0;">
- integriteitsregel 7, 8, 441, 535
 - in de voorbeelddatabase 39
 - toevoegen 480
 - verwijderen 481
 - verwijderen van 457
- integriteit van gegevens 4
- integrity rule 441
- interactief SQL 10, 641
- intergalactic dataspeak xix
- intermediate SQL 27
- International Standardization Organization 26
- internetarchitectuur 23
- INTERSECT ALL-operator 356, 697
- intersectie 349
- INTERSECT-operator 349
- interval 103, 106
- INTERVAL 103, 104
- intervaleenheid 104
- intervallengte 104
- INTO-component 599, 655, 662
 - FETCH CURSOR-instructie 609
- intrekken van bevoegdheden 556
- invoeren van rijen 389
- invoeren van SQL-gebruikers 542
- invoerparameter van een stored procedure 584
- invoerprogramma 14
- IS NULL-operator 220
- ISO 26
- isolation level 688
 - cursor stability 689
 - dirty read 689
 - read committed 689
 - repeatable read 688
 - serializable 688
- isoleren van kolommen 694
- ITERATE-instructie 595
- J
- Java 3, 703
 - join 140
 - USING 158
 - join-conditie 140
 - join-index 513
 - join-kolom 140
 - join-type
 - equi-join 158
 - greater-than-join 158
 - inner-join 144
 - left-outer-join 147, 148
 - less-than-join 158
 - natural-join 157
 - non-equi-join 158
 - theta-join 158
 - union-join 157
- K
- kandidaatsleutel 8, 484, 508
 - karakterset 172, 329, 426, 459
 - karakterset toekennen aan kolom 462
 - kettingpagina 516
 - knooppunt in een indexboom 495
 - kolom 6
 - afgeleid 437
 - datatype 420, 422
 - kwalificeren 81
 - naamgeving 434
 - toekennen van collation 464
 - toekennen van karakterset 462
 - toevoegen 476
 - verwijderen 479
 - virtueel 437
 - wijzigen van datatype 478
 - wijzigen van eigenschappen 477
 - wijzigen van naam 478
 - kolombevoegdheid 541, 546
 - kolomdefinitie 420
 - kolom-integriteitsregel 421, 454
 - kolomkop 79
 - kolom-optie
 - COMMENT 436
 - DEFAULT 435
 - kolomspecificatie 81, 133
 - kolom-subquery 127, 190
 - kolomtoekenning 394
 - kopiëren van rijen 391
 - kopiëren van tabellen 430
 - koppelen van condities 180
 - KPI 18
 - KPI-programma 18
 - Kritische Prestatie Indicator 18
 - kwalificeren van kolommen 81, 82, 133
 - kwalificeren van tabellen 132
- L
- label 586
 - leafpage 495
 - LEAVE-instructie 594
 - LEFT-functie 92

- left-outer-join 147, 148
- lengte van float-datatype 424
- less-than-join 158
- LIKE-operator 200, 695
- LIMIT-component 333
 - offset 341
 - subquery 338
- link-editor 645
- literal 70
- literatuur 705
- load-module 645
- locking 685
- LOCK TABLE-instructie 687
- LOCK TABLES-databasebevoegdheid 548
- lock-type 688
 - EXCLUSIVE 688
 - SHARE 688
- logische operator 180
- lokale variabele 587
- LONG VARCHAR-datatype 425
- LOOP-instructie 595
- lost update 684

- M
- masker 200
- MATCH-operator 210
 - relevantiewaarde 214
 - zoekvorm 210
- materialisatie van views 530
- MAX-functie 109, 258
- maximale lengte van alfanumeriek datatype 426
- MAXVALUE-sequence-optie 566
- mediaan 275
- menselijke gebruiker 45
- middleware 19
- MIN-functie 109, 258
- minimaliteitsregel 445
- MINUS-operator 355
- MINVALUE-sequence-optie 566
- MOD-functie 92
- monolitische architectuur 21
- MONTHNAME-functie 93
- multidimensionale interface 16
- multi-tabelindex 513
- multi-user 673
- MUTATIES-tabel 626
- muten van tabellen 389
- muten van views 523

- MySQL xx
 - downloaden 41
 - installeren 41
- MySQL Query Browser 11
- MySQL website 41

- N
- naamgeving
 - van kolom 434
 - van tabel 434
- natural join 157
- natural language search 212
- natural language search with query expansion 219
- natuurlijke-join 157
- Navicat 11
- nesten van functies 92
- nesten van subquery's 126
- NEW 629
- niet-executeerbare instructie 651
- niet-procedurele taal 10
- NOCYCLE-sequence-optie 566
- NOMAXVALUE-sequence-optie 566
- NOMINVALUE-sequence-optie 566
- non-equi-join 158, 302
- nonrepeatable read 683
- nonreproducible read 683
- NOORDER-sequence-optie 567
- NOT FOUND-handler 605
- NOT NULL 49
- NOT NULL-integriteitsregel 441
- NOT NULL-optie 492
- NOT-operator 180, 694
- null-indicator 658
- null-specificatie 420
- null-waarde 7, 97, 420, 658
- null-waarde en set-operatoren 357
- NUM-datatype 423
- NUMERIC-datatype 423
- numerieke expressie
 - samengestelde 99
- nummeringfuncties 269

- O
- objectmodule 645
- ODBC 28
- offset van de LIMIT-component 341
- OLAP 16, 268
- OLAP-functie 268

OLD 630
 ON DELETE 452
 ONDERDELEN-tabel 372
 onderwerpgeoriënteerd 25
 ongedaan maken van instructies 673
 OnLine Analytical Processing 16
 online-invoerprogramma 14
 ontkenning 242
 ontwerpen van een databasestructuur 483
 ON UPDATE 452
 opdracht 4
 OPEN CURSOR-instructie 609
 Open DataBase Connectivity 28
 OPEN-instructie 661
 open source 21
 Open SQL Access Group 28
 operationeel programma 24
 operationele database 24
 operationele gegevens 24
 operator
 set 123
 wiskundige 99
 oplopend sorteren 329
 optimaliseren van instructies 493, 499
 optimiser 500
 opzetten van tabellen 419
 Oracle 20
 ORDER BY-component 323
 ORDER-sequence-optie 567
 OR-operator 180, 691
 outer-join 698

P

pagina 494, 687
 ketting 516
 parameters van een scalaire functie 91
 parameter van een stored procedure 584
 partitie-specificatie 277
 partitionering 277
 patroon 200
 permanente tabellen 428
 persistent 4
 Persistent Stored Modules 27
 phantom read 684
 phpMyAdmin 11
 PIPES _AS_ CONCAT-variabele 101
 PL/SQL 579
 populatiestandaarddeviatie 266

populatie van een kolom 6
 populatievariantie 266
 positioned delete 669
 positioned update 667
 precisie 71
 van decimal-datatype 423
 precompiler 646
 precompileren van programma's 646
 predikaat 167
 predikatenlogica 5
 PREVIOUS VALUE FOR 570
 primaire sleutel 8, 421, 442, 484, 508
 in de voorbeelddatabase 39
 samengesteld 443
 primary key
 composite 443
 PRIMARY KEY 421, 443
 prioriteiten bij het rekenen 100
 procedurebody 582
 procedureel databaseobject 579
 procedurele instructie 60

 CALL 596
 CASE 593
 IF 590
 ITERATE 595
 LEAVE 594
 LOOP 595
 REPEAT 594
 WHILE 593
 procedurele taal 10
 productiedatabase 24
 productieprogramma 24
 production rule 627
 programma 45
 programmable databaseserver 581
 programmalogica 23
 proleptisch 103
 pseudoniem 79, 126, 138
 pseudo programmeertaal 643
 PUBLIC 546
 Python 3

Q

QBE 15
 QUEL 5
 Query-By-Example 15
 query expansion 219
 query optimisation 499

R

- raadplegen van de catalogusviews 64
- raadplegen van tabellen 51
- RANK-functie 275
- rapportageprogramma
 - batch 14
- read committed isolation level 689
- read lock 686
- read uncommitted isolation level 689
- recreatiespelers 33
- recursief aanroepen van stored procedures 597
- recursieve SELECT-instructie 367
- redundante conditie 696
- redundantie 488
- REFERENCES 447
- REFERENCES-databasebevoegdheid 547
- REFERENCES-tabelbevoegdheid 544
- referential key 9
- referenties 705
- refererende actie 452
- refererende integriteitsregel 447
- refererende sleutel 9, 447, 508
 - in de voorbeelddatabase 39
- refererende tabel 448
- REGEXP-operator 202
- regular expression 202
- reikwijdte van een kolom 233
- rekenkundig gemiddelde 263
- relatie 6
- relatie-operator 168
- relaties tussen join-kolommen 146
- relationele databaseserver 7
- relationele databasetaal 4, 5
- relationele model 3, 5
- relevantiewaarde 214
- remote databaseserver 22
- RENAME TABLE-instructie 474
- reorganisatie van een index 499
- reorganisatie van een tabel 532
- repeatable read isolation level 688
- repeating group 485
- REPEAT-instructie 594
- RESTRICT refererende actie 452
- RETURNS-specificatie 617
- richting 664
- richtlijnen naamgeving 434
- rij 6
- rijen gelijk 250
- rijexpressie 77, 110, 390

- rij-identificatie 495

- rij-subquery 127

- rijwaarde 77

- rol 552

- ROLLBACK-instructie 675, 680

- ROLLUP 303, 305, 311

- root 43, 45

- root van een indexboom 495

- ROUND-functie 92

- routinematige instructies 531

- ROUTINES-catalogustabel 612

- ROW_NUMBER-functie 269

- rule 627

- run-time 647

S

- samengestelde alfanumerieke expressie 101

- samengestelde datumexpressie 103

- samengestelde expressie 78, 98

- samengestelde index 499

- samengestelde instructie 586

- samengestelde numerieke expressie 99

- samengestelde primaire sleutels 443

- samengestelde sleutels 195

- samengestelde tabelexpressie 343

- samengestelde tijdexpressie 106

- samengestelde timestamp-expressie 108

- samenvoegen van kolommen 487

- savepoint 680

- SAVEPOINT-instructie 680

- scalaire expressie 77

- scalaire expressie tussen haakjes 90

- scalaire functie 91

- scalaire subquery 109, 127, 174

- scalaire waarde 77

- scanning 495

- schaal 71

- van decimal-datatype 423

- schema 575

- schrikkeljaar 103, 183

- SCROLL 664

- scroll-cursor 664

- searched delete 669

- searched update 667

- secondefractie 74

- select-blok 114, 233

- kopdeel 114

- staartdeel 114

- SELECT-component 245

- SELECT-databasebevoegdheid 547
- selectieve index 514
- SELECT-instructie 51, 69
 - componenten 114
 - recursief 367
 - verwerking 116
- SELECT INTO-instructie 599
- SELECT-INTO-instructie 656
- SELECT-tabelbevoegdheid 544
- self referencing table 450
- self referential integrity 450
- sensitive cursor 666
- Sequel 20
- sequence 561
- sequence-bevoegdheid
 - ALTER 573
 - USAGE 573
- sequence-expressie 562
- sequence-optie
 - CACHE 567
 - CYCLE 566
 - INCREMENT BY 565
 - MAXVALUE 566
 - MINVALUE 566
 - NOCYCLE 566
 - NOMAXVALUE 566
 - NOMINVALUE 566
 - NOORDER 567
 - ORDER 567
 - START WITH 564
- sequentiële zoekmethode 495
- serializable isolation level 688
- serializability 686
- servermachine 22
- SET AUTOCOMMIT-instructie 674
- set-functie 252
- SET-instructie 59, 83, 589
- SET NULL refererende actie 452, 453
- set-operator 123, 343
 - EXCEPT 353
 - EXCEPT ALL 356
 - INTERSECT 349
 - INTERSECT ALL 356
 - MINUS 355
 - UNION 343
 - UNION ALL 356
- set-operatoren combineren 357
- set-operatoren en de theorie 359
- set-operatoren en null-waarde 357
- SET TRANSACTION-instructie 690
- share lock 686
- SHARE lock-type 688
- single precision float-datatype 424
- single precision floating point 71
- single-user 673
- sleutel
 - alternatieve 9
 - kandidaat 8
 - primaire 8
 - refererende 9
 - verwijzende 9
 - vreemde 9
- SMALLINT-datatype 422
- SOME-operator 225
- sorteervolgorde 459
- sorteren
 - collations 468
- sorteren met volgnummers 327
- sorteren op expressies 325
- sorteren op kolomnamen 323
- sorteren van null-waarden 331
- sorteren van rijen 323
- sortering 323
- speciaal register 84
- SPELERS-tabel 35
- SPELERS_XXL-tabel 506, 701
- spreiding 266
- spreidingsmaatstaf 267
- SQL 10, 11
 - geschiedenis 19
- SQL\
 - 1999-standaard 28
 - 2003-standaard 28
 - 2008-standaard 28
- SQL1-standaard 26
- SQL2-standaard 27
- SQL3-standaard 27
- SQL-86-standaard 26
- SQL-89-standaard 26
- SQL-92-standaard 27
 - niveaus 27
- SQL/Bindings 28
- SQLCA 650
- SQL/CLI 27, 28
- SQLCODE-hostvariabele 649, 657, 662
- SQL-dialect xix, 31
- SQL/DS 20
- SQL-error-code 603

- SQL-exception-handler 605
- SQL/Foundation 28
- SQL/Framework 28
- SQL-gebruiker 45
- SQL-gebruiker creëren 542
- SQL-instructie
 - ALTER DATABASE 539
 - ALTER SEQUENCE 572
 - ALTER TABLE 474, 475
 - ALTER USER 544
 - BEGIN DECLARE SECTION 649
 - CALL 582
 - CLOSE 662
 - CLOSE CURSOR 610
 - COMMIT 675
 - CONNECT 654
 - CREATE DATABASE 47, 537, 538
 - CREATE INDEX 55, 503
 - CREATE PROCEDURE 582
 - CREATE ROLE 552
 - CREATE SCHEMA 576
 - CREATE SEQUENCE 562
 - CREATE TABLE 48, 419
 - CREATE TRIGGER 626
 - CREATE USER 45, 542
 - CREATE VIEW 56, 519
 - DECLARE CONDITION 605
 - DECLARE CURSOR 608, 660
 - DECLARE HANDLER 603
 - DECLARE VARIABLE 587
 - DELETE 54, 397, 669
 - DISCONNECT 655
 - DROP 58
 - DROP DATABASE 58, 540
 - DROP FUNCTION 623
 - DROP INDEX 58, 505
 - DROP PROCEDURE 613
 - DROP ROLE 553
 - DROP SCHEMA 577
 - DROP SEQUENCE 572
 - DROP TABLE 58, 474
 - DROP TRIGGER 638
 - DROP USER 543
 - DROP VIEW 58, 525
 - END DECLARE SECTION 649
 - EXECUTE PROCEDURE 582
 - FETCH 661, 662
 - FETCH CURSOR 609
 - GRANT 46, 57, 542, 573, 615
 - INCLUDE 650
 - INSERT 50, 389
 - LOCK TABLE 687
 - OPEN 661
 - OPEN CURSOR 609
 - RENAME TABLE 474
 - ROLLBACK 675, 680
 - samengesteld 586
 - SAVEPOINT 680
 - SELECT 51, 69
 - SELECT INTO 599, 656
 - SET 59, 83, 589
 - SET AUTOCOMMIT 674
 - SET TRANSACTION 690
 - START TRANSACTION 678
 - UPDATE 54, 393
 - USE 47
 - WHENEVER 651
- SQL/JRT 28
- SQL/MED 28
- SQL/MM 28
- SQL__MODE systeemparameter 434
- SQL__MODE-systeemparameter 101
- SQL__MODE-systeemvariabele
 - ANSI__QUOTES 434
- SQL/OLAP 28
- SQL/OLB 27
- SQL/PSM 27, 28
- SQL/Schemata 28
- SQLSTATE 603
- SQLWARNING-handler 605
- SQL/XML 28
- SQLyog 11
- SQRT-functie 92
- SQUARE 5
- standaarddeviatie 266, 267
- stappenplan voor installatie 41
- stapsgewijs opzetten van SELECT's 534
- START TRANSACTION-instructie 678
- START WITH-sequence-optie 564
- statische eigenschap van datawarehouses 25
- statistische functie 252
- STDDEV-functie 109, 267
- Stonebraker, Michael xix
- stopwoord 213
- stored function
 - verwijderen van 623

- stored procedure 10
 - body 585
 - parameter 584
 - recursief aanroepen 597
 - transactie 681
 - verwijderen van 613
- stored procedure 581
- string data type 425
- Structured Query Language 10, 11
- structuur van het boek 32
- subquery 125, 174
 - scalaire 109
- subquery in de FROM-component 126
- subquery met LIMIT-component 338
- subselect 125
- substitutie van views 529
- SUBSTR-functie 205
- subtotaal 347
- SUM-functie 109, 263
- systeemparameter 59
 - SQL__MODE 101
- systeemtabel 61
- systeemvariabele 59, 84
 - CHARACTER__SET__CLIENT 471
 - CHARACTER__SET__CONNECTION 471
 - CHARACTER__SET__DATABASE 471
 - CHARACTER__SET__DIR 471
 - CHARACTER__SET__RESULTS 471
 - CHARACTER__SET__SERVER 471
 - CHARACTER__SET__SYSTEM 471
 - COLLATION__CONNECTION 471
 - COLLATION__DATABASE 471
 - COLLATION__SERVER 471
- System R 19
- T
- tabel 6
 - afgeleide 519
 - basis 519
 - creëren 48, 419
 - kopiëren 430
 - naamgeving 434
 - opzetten 48, 419
 - pseudoniem 138
 - verwijderen 474
- tabelbevoegdheid 541, 544
 - ALL 544
 - ALTER 544
 - DELETE 544
 - INDEX 544
 - INSERT 544
 - REFERENCES 544
 - SELECT 544
 - UPDATE 544
- tabel-element 420
- tabelexpressie 77, 111, 113
 - algemene 363
- tabelexpressie zonder GROUP BY 245
- tabel-integriteitsregel 455
- tabeloptie 463
- tabelreferentie 131
- tabelschema 420
- tabelspecificatie 131
 - volgorde 137
- tabelstructuur wijzigen 473
- tabel-subquery 127, 159
- tabel-waarde 77
- TABLE__AUTHS-catalogusview 554
- table expression
 - common 363
- TABLES-catalogusview 438
- tag van XML 399
- TEAMS-tabel 36
- tekenset 459
- temporal-constante 73
- temporal datatype 427
- temporary table 428
- tennisvereniging 33
- theta-join 158
- tijd
 - rekenen met 106
- tijdconstante 73
 - bereik 74
- tijdelijke tabel 428
- tijdexpressie
 - samengestelde 106
- tijdgebondenheid 25
- tijdsduur 103
- tijdsinterval 103, 106
- TIME-datatype 427
- timestamp
 - rekenen met 108
- timestamp-constante 74
 - bereik 75
- TIMESTAMP-datatype 427
- timestamp-expressie
 - samengestelde 108
- toevoegen van databases 538

toevoegen van integriteitsregels 480
 toevoegen van kolommen 476
 totaal 347
 transactie 673
 stored procedure 681
 transactiedatabase 24
 transactieprogramma 24
 Transact-SQL 579
 trigger 10, 625
 trigger-actie 626, 628
 trigger-conditie 626, 632
 trigger-event 626
 triggering instructie 628
 triggering tabel 628
 trigger-moment 626
 TRUE 75
 tussenresultaat 118
 subquery 190

 U
 UCASE-functie 91
 uitvoerparameter van een stored procedure 584
 uncommitted read 682
 uniciteitsregel 444
 Unicode 329, 460
 unieke index 56
 unieke nummers 561
 UNION ALL-operator 356, 697
 union-compatibel 346, 359
 union-join 157
 UNION-operator 123, 343, 693
 UNIQUE 421, 445
 unit of work 673
 UPDATE-databasebevoegdheid 547
 UPDATE-instructie 54, 393
 UPDATE-tabelbevoegdheid 544
 UPDATEXML-functie 414
 USAGE-sequence-bevoegdheid 573
 USE-instructie 47
 USER__AUTHS-catalogusview 84, 555
 user-defined variables 82
 USERS-catalogusview 554
 user variable 82
 USING 158
 UTF-8 460
 UTF-16 460
 UTF-32 460

V
 VALUES-component 390
 VARCHAR-datatype 425
 VAR-functie 266
 VARIANCE-functie 109, 266
 variantie 266
 vereniging 343
 vergelijkingsoperator 168
 vermenigvuldigen van tabellen 135
 verschil 353
 versnellen van het raadplegen 55
 verstrekken van databasebevoegdheden 547
 verstrekken van gebruikersbevoegdheden 549
 verstrekken van kolombevoegdheden 546
 verstrekken van sequence-bevoegdheden 573
 verstrekken van tabelbevoegdheden 541, 544
 verticale deelverzameling 245
 verticale vergelijking 251, 295
 verwerken van views 529
 verwerkingsstrategie 499
 verwerkingstijd 483, 493
 verwijderen van databaseobjecten 58
 verwijderen van databases 540
 verwijderen van dubbele rijen in resultaat 247
 verwijderen van gebruikers 543
 verwijderen van indexen 505
 verwijderen van integriteitsregels 457, 481
 verwijderen van kolommen 479
 verwijderen van rijen 54, 397
 verwijderen van stored functions 623
 verwijderen van stored procedures 613
 verwijderen van tabellen 474
 verwijderen van triggers 638
 verwijderen van views 525
 verwijzende sleutel 9
 verzamelingenleer 5
 view 56, 519
 kolomnamen 522
 materialisatie 530
 muteren 523
 substitutie 529
 toepassingen 531
 verwerken 529
 viewformule 519
 VIEWS-catalogusview 525
 virtuele kolom 437
 virtuele-kolomindex 513
 virtuele tabel 56

- Visual Basic 3
- visualisatie-programma 18
- volgorde van componenten 115
- voorbeelddatabase
 - algemene beschrijving 33
 - inhoud 36
 - integriteitsregels 39
 - schematische weergave 39
- voorgeprogrammeerd SQL 641
- voorkennis xxi
- vreemde sleutel 9
- W
- W3C 400
- waarde 7, 8
- waarheidstabel 180
- wachtwoord 43, 542, 544
- webserver 23
- website behorende bij Het SQL leerboek xxi
- website van het boek 41
- website van MySQL 41
- WEDSTRIJDEN-tabel 36
- wedstrijdspelers 33
- werken van rollen 552
- wetten
 - associatieve 360
 - commutatief 359
 - distributieve 360
 - idempotentie 360
- when-definitie 85
- WHENEVER-instructie 651
- WHERE-component 167
- WHILE-instructie 593
- wijzigen datatype 478
- wijzigen naam van een tabel 474
- wijzigen van databases 539
- wijzigen van datatype 478
- wijzigen van default-karakterset 475
- wijzigen van de tabelstructuur 473
- wijzigen van kolommen 477
- wijzigen van kolomnamen 478
- wijzigen van rijen 54
- wijzigen van waarden 393
- wijzigen van wachtwoord 544
- wijzigen van XML-documenten 414
- WinSQL 11
- wiskundige operator 99
- WITH CASCADED CHECK OPTION 524
- WITH CHECK OPTION 524
 - CASCADED 524
- WITH-component 361
- WITH GRANT OPTION 551
- WITH RECOMPILE 614
- WITH ROLLUP 303, 305
- wortel van een indexboom 495
- write lock 686
- www.mysql.com 41
- www.r20.nl xxi, 41
- X
- XML 399, 703
- XML-document
 - wijzigen 414
- XPath 403
- XQuery 703
- Y
- YEAR-functie 91
- Z
- zoeken op de null-waarde 220
- zoekmethode 495
- zoekvorm 210
 - boolean search 216
 - natural language search 212
 - natural language search with query expansion 219



SQL was, is en blijft de taal voor de databasewereld. Alle bekende databaservers, waaronder MySQL, Microsoft SQL Server, IBM's DB2 en Oracle, werken met SQL. Dit al meer dan 25 jaar populaire leerboek maakt door middel van honderden voorbeelden de lezer stap voor stap vertrouwd met alle aspecten van SQL. Aan bod komen het opzetten en raadplegen van tabellen, muteren van gegevens, aanmaken van indexen, optimaliseren van instructies, opzetten van en werken met views, beveiligen van gegevens, en applicatieontwikkeling.

Deze zevende druk is geheel herzien en geactualiseerd.

Vier geheel nieuwe hoofdstukken zijn toegevoegd:

- Opslaan, zoeken en bewerken van XML-documenten
- Zoeken in tekst
- Werken met recursieve SQL-instructies
- Definiëren van zogenaamde sequences om unieke nummers te genereren

Online ondersteuning

Op de website www.r2o.nl zijn allerlei zaken te vinden die gerelateerd zijn aan dit boek. Naast een installatiegids, alle code uit het boek en de antwoorden op de opgaven is onder andere aanvullend materiaal beschikbaar over

- Syntaxis van SQL
- Scalaire functies
- Objectrelationele concepten in SQL
- Introductie tot de verzamelingenleer en de logica
- Relationale algebra en calculus

Rick F. van der Lans is onafhankelijk adviseur, docent en auteur, gespecialiseerd in SQL, databasetechnologie, datawarehousing en applicatie-integratie. Hij is vele jaren lid geweest van de Nederlandse commissie verantwoordelijk voor de ISO SQL-standaard. Hij is auteur van diverse boeken waarvan enkele in verscheidene talen uitgegeven worden. Daarnaast is hij voorzitter van de jaarlijkse European Data Warehouse and Business Intelligence Conference.

.

ISBN 978 90 395 2655 2

NUR 123 / 995



9 789039 526552

www.academicservice.nl