


SQL

Datamodellering 2008


1



Wat is SQL?

- SQL is een standaard interactieve bevroings- en definitietaal voor relationele DBMSen
- SQL is een set-based, declaratieve query-taal (dus niet procedureel!)
- DB-leveranciers → proprietary extensions

2/26



Geschiedenis SQL

- Jaren '70: IBM
Donald D. Chamberlin & Raymond F. Boyce
→ SEQUEL (voor Sytem R)
- 1986: SQL-86: ANSI-standaard
- 1989: SQL-89: kleinere uitbreidingen
- 1992: SQL-2: belangrijke herziening
- 1990: SQL-3: triggers, recursieve queries
- 2003: SQL2003: windows, XML-features
- 2006: SQL2006: integratie met XML, XQuery

3/26

Wat kun je doen met SQL?

- Data definition
 - definiëren van datastructuur en relaties
- Data retrieval.
 - bevrogen van de database
- Data manipulation.
 - toevoegen, verwijderen en wijzigen van gegevens
- Access control
 - autorisatie van gebruikers
- Data sharing
 - coördineren van gelijktijdig gebruik van data door verschillende gebruikers
- Data integrity
 - regels definiëren om de data integriteit van de database te bewaken

4/26

SQL als verbindende schakel

The diagram illustrates SQL as a central connecting link. At the top, five application boxes are shown: 'Firma Facility' (with a computer icon), 'Report Writer' (with a document icon and binary code), 'Interactive Query Tool' (with a monitor icon), 'Application Program' (with a programming icon), and 'Database Front-end' (with a server icon). Each application box has an arrow labeled 'SQL' pointing down to a central horizontal cylinder labeled 'Database Engine'. Below the 'Database Engine', two arrows labeled 'SQL' point to a 'Database' (represented by a cylinder) and a 'Database Gateway' (represented by a server rack). The 'Database Gateway' has an arrow labeled 'to other DBMS brands' pointing downwards.

5/26

Onderdelen SQL

- Data Manipulation Language (DML)
 - SELECT - extracts data from a database
 - UPDATE - updates data in a database
 - DELETE - deletes data from a database
 - INSERT INTO - inserts new data
- Data Definition Language (DDL)
 - CREATE DATABASE - creates a new database
 - ALTER DATABASE - modifies a database
 - CREATE TABLE - creates a new table
 - ALTER TABLE - modifies a table
 - DROP TABLE - deletes a table
 - CREATE INDEX - creates an index (search key)
 - DROP INDEX - deletes an index

6/26

Voorbeeld DDL

- `CREATE TABLE Employee (
Employee_id INT,
Name VARCHAR (50),
Byear INT NOT NULL,
PRIMARY KEY (Employee_id)
);`

7/26

SQL in DMO

- In DMO maar een beperkt deel van SQL (DML): alleen de queries (SELECT)
- Basisbewerkingen op relationele model:
 - selectie
 - projectie
 - join
 - unionDeze leveren altijd weer een tabel op!
- Let goed op, hoe deze bewerkingen in SQL worden "vertaald"

8/26

Projectie

- Projection
 - operatie op een relationele tabel waarbij **kolommen** worden geselecteerd, en die samen worden opgeleverd als een nieuwe tabel.
 - SQL
 - `SELECT Name, Byear FROM Employee`
 - N.B. projectie gaat meestal samen met selectie

9/26

Selectie

- Selection
 - operatie op een relationele tabel waarbij **rijen** (records) worden geselecteerd die aan een bepaald criterium voldoen, en die samen worden opgeleverd als een nieuwe tabel.
 - SQL
 - `SELECT * FROM Employee WHERE Byear > 1950`
 - Syntax:
 - `SELECT kolom(men) FROM tabel WHERE conditie ORDER BY kolom(men) ASC|DESC`

10/26

Condities en operatoren

Operator	Description
=	Equal
<>	Not equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
BETWEEN	Between an inclusive range
LIKE	Search for a pattern
IN	Return a specific set



11/26

BETWEEN

- `SELECT * FROM Persons WHERE LastName BETWEEN 'Hansen' AND 'Pettersen'`
- Implementaties verschillen (wel of niet inclusief)

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

12/26

LIKE

- Alle personen in een stad waarvan de naam begint met "Sa":
- SELECT * FROM Persons WHERE City LIKE 'Sa%'**
- % = wildcard (willekeurige letters)

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

13/26

IN (set)

- Alleen Hansen en Pettersen
- SELECT * FROM Persons WHERE LastName IN ('Hansen', 'Pettersen')**

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

14/26

Duplicaten verwijderen

- SELECT DISTINCT City FROM Persons**
- Resultaat: 1x Sandnes en 1x Stavanger

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

15/26

Join

- Join is een "zijdelingse" verbinding van tabellen, waarbij eveneens een nieuwe tabel ontstaat
- Combineren van data uit
 - verschillende tabellen onderling verbonden door PK-FK
 - dezelfde tabel: self-referencing → self-join
- Speciale joins als bij een 1:N-relatie aan de N-kant records ontbreken → inner / outer join

16/26

Inner join

- Alleen matching records worden opgeleverd in de nieuwe tabel
- Als =-operator wordt gebruikt: equi-join
- SQL
 - ```
SELECT T1.*, T2.Product
FROM T1, T2
WHERE T1.ID = T2.Cid
```
  - of vanaf SQL92:
 

```
SELECT T1.*, T2.Product
FROM T1
JOIN T2
ON T1.ID = T2.Cid
```

 (mag ook nog een where-clause hebben)

17/26

---

---

---

---

---

---

---

---

## Outer join

- Alle records (van de linker en/of rechter tabel) worden opgeleverd in de nieuwe tabel → NULL-waarden waar matching niet mogelijk is
  - left outer join
  - right outer join
  - full outer join
- SQL
  - ```
SELECT T1.*, T2.Product
FROM T1
LEFT JOIN T2
ON T1.ID = T2.Cid
```

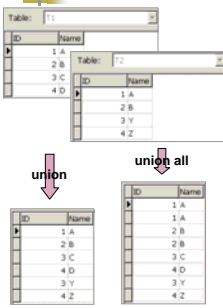
18/26

Union - 1

- Union is een "kop-staart" verbinding van tabellen, waarbij eveneens een nieuwe tabel ontstaat
- Union combineert de resultaten van twee select-queries, waarbij duplicaat-records automatisch worden verwijderd (tenzij **union all** wordt gebruikt)
- Tabellen moeten wel overeenkomen in datastructuur!

19/26

Union - 2



- Gegeven
 - tabel T1 en tabel T2
- Gevraagd
 - de totale verzameling T1 plus T2
- SQL
 - `SELECT * FROM T1
UNION
SELECT * FROM T2
ORDER BY Name`
 - `SELECT * FROM T1
UNION all
SELECT * FROM T2
ORDER BY Name`

20/26

Aggregatie

- Functies → geaggregeerde waarde (telling, som, etc.)

Function	Description
AVG(column)	Returns the average value of a column
COUNT(column)	Returns the number of rows (without a NULL value) of a column
COUNT(*)	Returns the number of selected rows
MAX(column)	Returns the highest value of a column
MIN(column)	Returns the lowest value of a column
SUM(column)	Returns the total sum of a column

Er zijn er meer!

21/26

GROUP BY

- Aggregatie **per groep**
- `SELECT Company, SUM(Amount) FROM Sales GROUP BY Company`
- Resultaat:
 - W3Schools 12600
 - IBM 4500

Company	Amount
W3Schools	5500
IBM	4500
W3Schools	7100

22/26

HAVING

- HAVING: **conditie** bij aggregatie
- `SELECT Company, SUM(Amount) FROM Sales GROUP BY Company HAVING SUM(Amount)>10000`
- Resultaat: W3Schools 12600

Company	Amount
W3Schools	5500
IBM	4500
W3Schools	7100

23/26

Views

- VIEW is een virtuele tabel gebaseerd op het resultaat van een SELECT statement
- Syntax:


```
CREATE VIEW view_naam AS
SELECT kolom(men)
FROM tabel(len) WHERE condition
```
- Gebruik: als "hulptabel", b.v. in nieuwe query

24/26

Oefening 1

Queries:

- 1) Iedereen met voor-naam, achter-naam die *niet* woont in 'Payson'
- 2) Iedereen ouder dan 40
- 3) Iedereen met 'ay' in de achternaam

empinfo					
first	last	id	age	city	state
John	Jones	99980	45	Payson	Arizona
Mary	Jones	99982	25	Payson	Arizona
Eric	Edwards	88232	32	San Diego	California
Mary	Edwards	88233	32	Phoenix	Arizona
Ginger	Howell	98002	42	Cottonwood	Arizona
John	Smith	92001	23	Gila Bend	Arizona
Gus	Gray	22322	35	Bagdad	Arizona
Mary	May	32326	52	Tucson	Arizona
Erica	Williams	32327	60	Show Low	Arizona
Leroy	Brown	32380	22	Pinetop	Arizona
Elroy	Cleaver	32382	22	Globe	Arizona

25/26

Oefening 2

Query:

Alle klanten (voor-, achternaam, plaats) in: Arizona, Washington, Colorado en Hawaii

customers					
customer	firstname	lastname	city	state	
10101	John	Gray	Lynden	Washington	
10298	Leroy	Brown	Pinetop	Arizona	
10299	Elroy	Keller	Snoqualmie	Washington	
10315	Lisa	Jones	Oshkosh	Wisconsin	
10325	Ginger	Schultz	Pocatello	Idaho	
10329	Kelly	Mendoza	Kailua	Hawaii	
10330	Shawn	Dalton	Cannon Beach	Oregon	
10338	Michael	Howell	Tillamook	Oregon	
10339	Anthony	Sanchez	Winslow	Arizona	
10408	Elroy	Cleaver	Globe	Arizona	
10410	Mary Ann	Howell	Charleston	Colorado	
10413	Donald	Davids	Gila Bend	Arizona	

26/26