

Carl-Benz-Schule Koblenz	Dynamische Webseiten	A. Hahn
	Thema: <b>SQL-Befehle</b>	Version: 2.0

## Operationen der Relationenalgebra in SQL

SQL (Structured Query Language) ist eine Programmiersprache der 4. Generation und **die** Sprache zum Aufbau, zur Verwaltung und zur Abfrage von relationalen Datenbanken. Sie wurde von IBM im Rahmen eines Forschungsprojektes entwickelt und 1987 international standardisiert. (Fast) alle Datenbanksysteme benutzen und arbeiten mit dieser Sprache.

## Einteilung der SQL-Befehle

SQL-Befehle werden in drei Klassen unterteilt:

- **DDL – Data Definition Language**  
Befehle zur Definition von Tabellen und anderer Datenstrukturen
- **DCL – Data Control Language**  
Befehle zur Kontrolle der Zugriffsberechtigungen
- **DML – Data Manipulation Language**  
Befehle zur Datenmanipulation und Datenabfrage

DDL	DCL	DML
create table (Tabellen erzeugen)	grant (Zugriffsrechte gewähren)	select (Tabellen abfragen)
alter table (Aufbau von Tabellen ändern)	revoke (zugriffsrechte entziehen)	delete (Zeilen einer Tabelle löschen)
drop table (Tabellen löschen)		
create index (Index für Tabellen anlegen)		insert (Zeilen in eine Tabelle einfügen)
create view (Erzeugen einer virtuellen Tabelle)		update (Daten in einer Tabelle verändern)
rename (Tabellen, Spalten,... Umbenennen)		

Carl-Benz-Schule Koblenz	Dynamische Webseiten	A. Hahn
	Thema: <b>SQL-Befehle</b>	Version: 2.0

## DML-Befehl: SELECT

Die SQL-Abfrage erfolgt mit dem Befehl SELECT unter Angabe von bis zu sechs Komponenten. Die allgemeine Syntax hat die Gestalt:

```
SELECT [ALL | DISTINCT] {spalten | *}
FROM tabelle [alias] [tabelle [alias]] ...
[WHERE {bedingung | unterabfrage}]
[GROUP BY spalten [HAVING {bedingung
|unterabfrage}]]
[ORDER BY spalten [ASC | DESC]...];
```

Die schwierige Syntax lässt sich wie folgt verstehen:

Klausel	Erläuterung
<b>SELECT</b> [ <b>DISTINCT</b> ]	<b>Wähle</b> die Werte aus der/den Spalte(n) [ <b>mehrfache Datensätze nur einmal</b> ]...
<b>FROM</b>	... <b>aus</b> der Tabelle bzw. den Tabellen ...
<b>WHERE</b>	... <b>wobei</b> die Bedingung(en) erfüllt sein soll(en) ...
<b>GROUP BY</b>	... und <b>gruppieren</b> die Ausgabe von allen Zeilen mit gleichem Attributwert zu einer einzigen ...
<b>HAVING</b>	... <b>wobei darin</b> folgende zusätzliche Bedingung(en) gelten müssen/muss ...
<b>ORDER BY</b> [ <b>ASC/DESC</b> ]	... und <b>sortiere</b> nach den Spalten [ <b>auf- bzw. absteigend</b> ].

Die SQL-Befehle sind maximal 256-Zeichen lang und müssen mit Semikolon abgeschlossen werden. Attributbezeichner, die Leerzeichen oder Satzzeichen enthalten müssen in eckigen Klammern gesetzt werden.

## Selektion in SQL

Aus der Tabelle **Schueler** sollen alle Zeilen selektiert werden, in denen der Name "Müller" steht. Die Umsetzung in SQL lautet:

```
SELECT *
FROM Schueler
WHERE Name = 'Müller';
```

SNr	Vorname	Name
4711	Paul	Müller
0815	Erich	Schmidt
7472	Sven	Lehmann
1234	Olaf	Müller
2313	Jürgen	Paulsen

→

SNr	Vorname	Name
4711	Paul	Müller
1234	Olaf	Müller

Carl-Benz-Schule Koblenz	Dynamische Webseiten	A. Hahn
	Thema: <b>SQL-Befehle</b>	Version: 2.0

Die WHERE-Klausel liefert also die Selektion. Um zu zeigen, dass alle Spalten in der Ausgabetabelle angezeigt werden sollen, wird ein Joker – das Prozent-Symbol % – verwendet. (MS Access: Das Stern-Symbol)

Nun sollen aus der Tabelle **Schueler** sollen alle Zeilen selektiert werden, in denen der Name "Müller" steht und deren Vorname mit "O" *beginnt*. Die Umsetzung in SQL lautet:

```
SELECT *
FROM Schueler
WHERE Name = 'Müller'
AND Vorname
LIKE 'O%';
```

SNr	Vorname	Name	→	SNr	Vorname	Name
4711	Paul	Müller		1234	Olaf	Müller
0815	Erich	Schmidt				
7472	Sven	Lehmann				
1234	Olaf	Müller				
2313	Jürgen	Paulsen				

Bedingungen lassen sich mit **AND**, **OR** und **NOT** verknüpfen.

Operator	Erklärung
=	Attributwert gleich einem anderen Attributwert oder einer Konstanten
< <= > >=	Attributwert soll kleiner, größer, kleiner gleich oder größer gleich einem anderen Attributwert oder einer Konstanten sein.
<>	Attributwert ist ungleich einem anderen Attributwert oder einer Konstanten.
BETWEEN	Attributwert zwischen zwei Grenzen: ... WHERE Geburtsjahr BETWEEN 1960 and 1970
IN	Attributwert in einer Menge enthalten: ... WHERE Geburtsjahr IN (1960,1961,1962)
LIKE	Suche nach Zeichenketten anhand von Ähnlichkeitsoperatoren: % : Platzhalter für beliebige Zeichenkette (in MS Access: *) _ : Platzhalter für ein Zeichen (in MS Access: ?) Es wird zwischen Groß- und Kleinschreibung unterschieden.
IS NULL	IS NULL oder IS NOT NULL zur Selektion nicht definierter Attributwerte

### Hier einige Beispiele für Platzhalter:

- **'A\_Z'**: Alle Zeichenketten die mit einem 'A' beginnen, worauf ein weiteres Zeichen folgt, und mit einem 'Z' enden. 'ABZ' und 'AZZ' würden beispielsweise diese Bedingung erfüllen, 'AKKZ' hingegen nicht (wegen der zwei Zeichen zwischen A und Z).
- **'ABC%'**: Alle Zeichenketten, die mit 'ABC' beginnen. Sowohl 'ABCD' als auch 'ABCABC' würden zum Beispiel diese Bedingung erfüllen.
- **'%XYZ'**: Alle Zeichenketten, die auf 'XYZ' enden. So würden beispielsweise sowohl 'WXYZ' als auch 'ZZXYZ' diese Bedingung erfüllen.
- **'%AN%'**: Alle Zeichenketten, die an irgendeiner Stelle das Muster 'AN' enthalten.

## Projektion in SQL

Aus der Tabelle **Schueler** sollen alle Spalten mit dem Attribut "Name" projiziert werden.  
Die Umsetzung in SQL lautet:

```
SELECT Name  
FROM Schueler;
```

<u>SNr</u>	Vorname	Name
4711	Paul	Müller
0815	Erich	Schmidt
7472	Sven	Lehmann
1234	Olaf	Müller
2313	Jürgen	Paulsen

→

Name
Müller
Schmidt
Lehmann
Müller
Paulsen

Im Gegensatz zur Projektion laut Relationenalgebra zeigt SQL alle Mehrfacheinträge an. Um dies zu unterbinden, muss dem SELECT-Befehl das Attribut **DISTINCT** hinzugefügt werden. Die Umsetzung in SQL lautet:

```
SELECT DISTINCT Name  
FROM Schueler;
```

<u>SNr</u>	Vorname	Name
4711	Paul	Müller
0815	Erich	Schmidt
7472	Sven	Lehmann
1234	Olaf	Müller
2313	Jürgen	Paulsen

→

Name
Müller
Schmidt
Lehmann
Paulsen

Die SELECT-Klausel liefert also die Projektion.

## Hintereinanderausführung von Projektion und Selektion

Aus der Tabelle **Schueler** sollen die Vornamen aller Schüler angezeigt werden, deren Nachname Müller ist. Die Umsetzung in SQL lautet:

```
SELECT Vorname  
FROM Schueler  
WHERE Name='Müller' ;
```

<u>SNr</u>	Vorname	Name
4711	Paul	Müller
0815	Erich	Schmidt
7472	Sven	Lehmann
1234	Olaf	Müller
2313	Jürgen	Paulsen

→

Vorname
Paul
Olaf

Die Abarbeitung des SQL-Befehls erfolgt also immer von innen nach außen, d. h. zuerst die Selektion, dann die Projektion.

Carl-Benz-Schule Koblenz	Dynamische Webseiten	A. Hahn
	Thema: <b>SQL-Befehle</b>	Version: 2.0

## Joins in SQL

Es gibt verschiedene Arten von JOINS (also Verknüpfungen mehrerer Tabellen).

**Lassen Sie sich hierzu die PPT von Hr. Hahn zeigen**

Aus den Tabellen **Schueler** und **Kurse** soll eine Tabelle gebildet werden.

Das gemeinsame Attribut lautet **SNr**.

Die Umsetzung in SQL lautet:

```
SELECT *
FROM Schueler
INNER JOIN Kurse
ON Schueler.SNr = Kurse.SNr
ORDER BY Kurse.Fehlstunden
DESC ;
```

Schueler			Kurse			
SNr	Vorname	Name	SNr	KNr	Fehlstunden	Punkte
4711	Paul	Müller	0815	03	0	12
0815	Erich	Schmidt	4711	03	12	03
7472	Sven	Lehmann	1234	23	3	14
1234	Olaf	Müller	0815	23	0	10
2313	Jürgen	Paulsen				

→

Schueler.SNr	Vorname	Name	Kurse.SNr	KNr	Fehlstunden	Punkte
4711	Paul	Müller	4711	03	12	03
1234	Olaf	Müller	1234	23	3	14
0815	Erich	Schmidt	0815	03	0	12
0815	Erich	Schmidt	0815	23	0	10

Das Ergebnis ist ein INNER JOIN, wobei die entstandene Tabelle nach den Fehlstunden absteigend sortiert ist. Ein solches Ergebnis wird aber i. d. R. nicht gewünscht. Es genügen oft nur einige Spalten des Verbundes.

Aus den Tabellen **Schueler** und **Kurse** sollen alle Schülernamen mit ihren Fehlstunden aufgelistet werden. Der Befehl lautet:

```
SELECT Schueler.Vorname, Schueler.Name, Kurse.Fehlstunden
FROM Schueler
INNER JOIN Kurse ON Schueler.SNr = Kurse.SNr
ORDER BY Kurse.Fehlstunden
DESC ;
```

Vorname	Name	Fehlstunden
Paul	Müller	12
Olaf	Müller	3
Erich	Schmidt	0
Erich	Schmidt	0

Der letzte Eintrag erscheint doppelt. Dies lässt sich durch Benutzung des Attributes **DISTINCT** verhindern