

# LE 6 / 7a – IBIS – Datenbanken

## Relationenalgebra und einfache Abfragen in SQL

Angelehnt an Skript „Datenmodellierung und Datenmanagement“, THM  
Mitwirkende Autoren: Prof. Dr. Guckert; Timo Péus, Dr. Thomas Farrenkopf,  
Melanie Vanderpuye, Prof. Dr. Grüne (2017)

Prof. Dr. Markus Grüne, FB03, Wirtschaftsinformatik

# Inhalt

- Was ist Relationenalgebra
- Was ist SQL?
- Wie werden Abfragen in SQL formuliert?
- Wie sind die Ergebnisse zu interpretieren?

# Relationenalgebra

Suchanfragen an die Relationen einer Datenbank können in der so genannten **relationalen Algebra** formuliert werden.

Hierfür stehen verschiedene Operatoren zur Verfügung, die auf Relationen angewendet jeweils wieder Relationen als Ergebnis liefern.

Es gibt zwei Typen von Operatoren:

- Mengenoperatoren (Durchschnitt, Vereinigung,...)
- spezielle Operatoren für Relationen (Projektion,...)

## Wiederholung „Relation“

$A_1, \dots, A_n$  seien Attribute mit den Domänen  $D_i = \text{dom}(A_i)$ .

$R(A_1, \dots, A_n) \subseteq D_1 \times D_2 \times \dots \times D_n$  heißt eine Relation von Grad  $n$ .

Ein **Tupel** aus  $R$  hat dann die Form  $(r_1, \dots, r_n) \in R$  mit  $r_i \in D_i$ .

Eine Relation ist (also) eine Teilmenge eines kartesischen Produkts.  
Die relationalen Operatoren sind auf der Menge aller Tupel definiert.

$R$  meint im Folgenden die Relation.

# Operatoren der Relationenalgebra

## **Projektion $\pi$**

Auswahl von Attributen

## **Selektion $\sigma$**

Auswahl von Tupeln

## **Vereinigung (Union) $\cup$**

Ergebnismenge enthält alle Elemente der verknüpften Mengen.

## **Durchschnitt (Intersection) $\cap$**

## **Differenz (Difference) $\setminus$**

Zeilen der einen Menge, die in der anderen nicht enthalten sind.

## **Umbenennen $\rightarrow$**

Attribut (oder Relation) erhält einen neuen Namen.

## **Verbund (Join) $\bowtie$**

Führt mehrere Tabellen zu einem Ergebnis zusammen. Umkehrung der Normalisierung (folgen im nächsten Kapitel).

# Beispielrelation *Lehrer*

	PNr	Titel	Nachname	Vorname	Gebdatum	SchulEintritt	Stufe
▶	1	Dr.	Schmidt	Erika	1953-07-01	1978-08-01	Studiendirektorin
	2	NULL	Schön	Helmut	1948-01-07	1975-03-01	OberStudienrat
	3	NULL	Glensmann	Jürgen	1968-06-06	1993-05-01	Studienrat
	4	NULL	Derwall	Jupp	1958-03-02	1984-06-01	Studienrat
	5	NULL	Nerz	Ottillie	1962-05-06	1990-05-06	Stduienrat
	6	NULL	Lukas	Laura	1979-01-09	2004-08-17	Studienrätin
	7	NULL	Meier	Horst	1959-04-04	1989-07-04	Oberstudienrat
	8	NULL	Müller	Gerd	1975-02-02	2004-09-14	Studienrat
	9	Dr.	Bauer	Renate	1965-05-01	1991-04-02	Studienrdirektor
	10	NULL	Hummel	Heinz	1969-02-02	1994-01-01	Studienrat
	12	NULL	Herberger	Sepp	1962-04-01	1987-08-01	Oberstudienrat
	13	NULL	Hörbiger	Christine	1963-05-01	1987-08-01	Oberstudienrätin

Zustand

Schema

lehrer - Table										
		Table Name: lehrer				Schema: schule				
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default	
PNr	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'	
Titel	CHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL	
Nachname	CHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL	
Vorname	CHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL	
Gebdatum	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL	
SchulEintritt	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL	
Stufe	CHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL	

# Beispielrelation Schüler

	SNr	Nachname	Vorname	GebDatum	SchulEintritt	KNr
▶	1	Rottenberg	Silke	1991-01-25	2001-08-01	8
	2	Angerer	Nadine	1997-11-10	2007-08-01	2
	3	Krahn	Annik	1994-07-01	2004-08-01	5
	4	Stegemann	Kerstin	1996-09-29	2006-08-01	3
	5	Zietz	Jennifer	1992-09-14	2002-08-01	7
	6	Gunther	Sarah	1992-01-25	2002-08-01	7
	7	Minnert	Sandra	1992-04-07	2002-08-01	7
	8	Hingst	Ariane	1998-07-25	2008-08-01	1
	9	Carlson	Britta	1997-03-03	2007-08-01	2
	10	Lingor	Renate	1994-10-11	2004-08-01	5
	11	Okoyino_d...	Celia	1997-06-27	2007-08-01	2
	12	Garefrekes	Kerstin	1998-09-04	2008-08-01	1
	13	Wunderlich	Pia	1994-01-26	2004-08-01	5
	14	Grings	Inka	1997-10-31	2007-08-01	2
	15	Smisek	Sandra	1996-07-03	2006-08-01	3
	16	Prinz	Birgit	1996-10-25	2006-08-01	3
	17	Mittan	Ania	1994-05-16	2004-08-01	5

Zustand

Schema

schueler - Table									
Table Name: <input type="text" value="schueler"/>		Schema: <b>schule</b>							
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
SNr	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Nachname	CHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
Vorname	CHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
GebDatum	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
SchulEintritt	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
KNr	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'

# Projektion

Eine Projektion stellt nur bestimmte Attribute einer Relation dar.  $\pi_{\langle \text{Attributliste} \rangle}(R)$

Schreibweise/Symbol:  $\pi_{\text{Nachname,Vorname}}(\text{Lehrer})$

	nachname	vorname
►	Schmidt	Erika
	Schön	Helmut
	Gliensmann	Jürgen
	Derwall	Jupp
	Nerz	Otilie
	Lukas	Laura
	Meier	Horst
	Müller	Gerd
	Bauer	Renate
	Hummel	Heinz



# Selektion

Die Selektion liefert eine Teilmenge der Tupel der Relation. Schreibweise/Symbol:

$$\sigma_{\langle \text{Filterbedingung} \rangle}(R)$$

$$\sigma_{\text{Vorname}=\text{"Erika"}}(\text{Lehrer})$$

	PNr	Titel	Nachname	Vorname	Gebdatum	SchulEintritt	Stufe
▶	1	Dr.	Schmidt	Erika	1953-07-01	1978-08-01	Studiendirektorin

# Vereinigung, Durchschnitt und Differenz

**Vereinigung** (union) , **Durchschnitt** (intersection) und **Differenz** (difference) entsprechen den bekannten Verknüpfungen auf Mengen. Die Relationen müssen dafür kompatibel sein: Gleicher Grad und gleiche Domänen der Attribute.

$$\pi_{\text{Nachname,Vorname}}(\text{Lehrer}) \cup \pi_{\text{Nachname,Vorname}}(\text{Schüler})$$

	nachname	vorname
►	Schmidt	Erika
	Schön	Helmut
	Gliensmann	Jürgen
	Derwall	Jupp
	Nerz	Ottile
	Lukas	Laura
	Meier	Horst
	Müller	Gerd
	Bauer	Renate
	Hummel	Heinz

	nachname	vorname
►	Rottenberg	Silke
	Angerer	Nadine
	Krahn	Annik
	Stegemann	Kerstin
	Zietz	Jennifer
	Gunther	Sarah
	Minnert	Sandra
	Hinnst	Ariane

	nachname	vorname
►	Schmidt	Erika
	Schön	Helmut
	Gliensmann	Jürgen
	Derwall	Jupp
	Nerz	Ottile
	Lukas	Laura
	Meier	Horst
	Müller	Gerd
	Bauer	Renate
►	Rottenberg	Silke
	Angerer	Nadine
	Krahn	Annik
	Stegemann	Kerstin
	Zietz	Jennifer
	Gunther	Sarah
	Minnert	Sandra
	Hinnst	Ariane

# Ausdrücke in der Algebra

Die Operatoren können hintereinander ausgeführt werden.  
Dies ist möglich, da die Ergebnisse ja jeweils wieder Relationen sind.

$\pi_{\text{Nachname.Vorname}}(\sigma_{\text{Vorname}=\text{"Erika"}}(\text{Lehrer}))$

	nachname	vorname
▶	Schmidt	Erika

$\sigma_{\text{Vorname}=\text{"Erika"}}(\pi_{\text{Nachname, Vorname}}(\text{Lehrer}))$

?

Komplexe algebraische Ausdrücke können mit Hilfe des Operators für das Umbenennen vereinfacht werden.

Schreibweise / Symbol:

$S \leftarrow R$

Neuer Name für Relation

oder

$S_{attr_1\_neu, \dots, attr_n\_neu} \leftarrow R_{attr_1\_alt, \dots, attr_n\_alt}$

Neuer Name für Relation und Attribute

Diese Relation heißt jetzt S

	nname	vname
►	Schmidt	Erika
	Schön	Helmut
	Gliensmann	Jürgen
	Derwall	Jupp
	Nerz	Ottile
	Lukas	Laura
	Meier	Horst
	Müller	Gerd

$S_{nname, vname} \leftarrow \pi_{nachname, vorname} (\text{Lehrer})$

	nachname	vorname
►	Schmidt	Erika
	Schön	Helmut
	Gliensmann	Jürgen
	Derwall	Jupp
	Nerz	Ottile
	Lukas	Laura
	Meier	Horst
	Müller	Gerd

# SQL als Standardabfragesprache

SQL ist standardisiert (DIN und ISO).

Datenbanksysteme am Markt entsprechen mehr oder weniger den Richtlinien. DML und DDL sind weitgehend einheitlich.

Starke Abweichungen zwischen verschiedenen Produkten

- bei den Spaltenfunktionen und im Bereich der DCL.
- Umsetzung von Prozeduren (Abläufe von Anweisungen, gespeichert unter einem Namen, zu behandeln wie ein eigener Befehl).

# SQL

Alle Ausdrücke der Relationenalgebra können auch in SQL formuliert und ausgeführt werden.

Lesende Datenbankabfragen werden mit *Select*-Statements durchgeführt:

- **Selektion** = Welche Sätze möchte ich sehen?  
(Auswahl von Zeilen)
- **Projektion** = Welche Attribute möchte ich sehen?  
(Auswahl von Attributen)

# Einfaches Select

SELECT [<Liste von Spalten>, \*]

FROM <tabelle>

WHERE <boolscher Ausdruck auf Spalten>

$\pi$   
R  
 $\sigma$

```
SELECT *  
FROM LEHRER;  
SELECT NACHNAME  
FROM LEHRER  
WHERE SCHULEINTRITT>"1997-11-01";
```

Für die WHERE-Klausel Spalten stehen typabhängige Vergleichsoperatoren bereit. Vergleiche können mit boolschen Operatoren verknüpft werden.

Column Name	Datatype	PK	NN
PNr	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Titel	CHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>
Nachname	CHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>
Vorname	CHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>
Gebdatum	DATE	<input type="checkbox"/>	<input type="checkbox"/>
SchulEintritt	DATE	<input type="checkbox"/>	<input type="checkbox"/>
Stufe	CHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>

**Alle Spalten, keine Einschränkung**

`SELECT * FROM lehrer;`

Result Grid							
Filter Rows:							
Edit:							
Export/Import:							
	PNr	Titel	Nachname	Vorname	Gebdatum	SchulEintritt	Stufe
▶	1	Dr.	Schmidt	Erika	1953-07-01	1978-08-01	Studiendirektorin
	2	NULL	Schön	Helmut	1948-01-07	1975-03-01	OberStudienrat
	3	NULL	Gliensmann	Jürgen	1968-06-06	1993-05-01	Studienrat
	4	NULL	Derwall	Jupp	1958-03-02	1984-06-01	Studienrat
	5	NULL	Nerz	Ottile	1962-05-06	1990-05-06	Stduienrat
	6	NULL	Lukas	Laura	1979-01-09	2004-08-17	Studienrätin
	7	NULL	Meier	Horst	1959-04-04	1989-07-04	Oberstudienrat
	8	NULL	Müller	Gerd	1975-02-02	2004-09-14	Studienrat
	9	Dr.	Bauer	Renate	1965-05-01	1991-04-02	Studienrdirektor
	10	NULL	Hummel	Heinz	1969-02-02	1994-01-01	Studienrat
	12	NULL	Herberger	Sepp	1962-04-01	1987-08-01	Oberstudienrat
	13	NULL	Hörbiger	Christine	1963-05-01	1987-08-01	Oberstudienrätin
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL



Column Name	Datatype	PK	NN
PNr	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Titel	CHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>
Nachname	CHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>
Vorname	CHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>
Gebdatum	DATE	<input type="checkbox"/>	<input type="checkbox"/>
SchulEintritt	DATE	<input type="checkbox"/>	<input type="checkbox"/>
Stufe	CHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>



## Projektion auf 2 Spalten, Keine Einschränkung

`SELECT Nachname, Gebdatum FROM lehrer;`



Result Grid		Filter Rows:
	Nachname	Gebdatum
▶	Schmidt	1953-07-01
	Schön	1948-01-07
	Gliensmann	1968-06-06
	Derwall	1958-03-02
	Nerz	1962-05-06
	Lukas	1979-01-09
	Meier	1959-04-04
	Müller	1975-02-02
	Bauer	1965-05-01
	Hummel	1969-02-02
	Herberger	1962-04-01
	Hörbiger	1963-05-01

Column Name	Datatype	PK	NN
PNr	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Titel	CHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>
Nachname	CHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>
Vorname	CHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>
Gebdatum	DATE	<input type="checkbox"/>	<input type="checkbox"/>
SchulEintritt	DATE	<input type="checkbox"/>	<input type="checkbox"/>
Stufe	CHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>



## Projektion auf 2 Spalten, mit Selektion

```
SELECT Nachname, Gebdatum FROM lehrer
WHERE Stufe = 'Studienrat' or Stufe = 'Studienrätin';
```




Result Grid			Filter Rows:
	Nachname	Gebdatum	
▶	Gliensmann	1968-06-06	
	Derwall	1958-03-02	
	Lukas	1979-01-09	
	Müller	1975-02-02	
	Hummel	1969-02-02	

# Sortieren

- Die Resultate einer SQL-Abfrage sind Mengen und besitzen daher keine festgelegte Sortierreihenfolge.
- Mit Hilfe eines Sortierzusatzes ORDER BY kann die Ergebnismenge sortiert werden.


```
SELECT *  
FROM lehrer  
ORDER BY nachname;
```

```
SELECT vorname, nachname  
FROM lehrer  
ORDER BY 2 ASC;
```



Sortiere nach 2. Spalte der Ergebnismenge

```
SELECT DISTINCT vorname  
FROM schueler  
ORDER BY 1 DESC
```



Keine Duplikate, also eine echte Menge

ASC=Ascending  
DESC=Descending

# Spaltenfunktionen

- In einer SQL-Abfrage können auf die Spalten so genannten Spaltenfunktionen angewendet werden.
- Beispiel:  
Extraktion des Jahres bei einer Spalte mit Wertebereich Date.

```
SELECT NACHNAME  
FROM LEHRER  
WHERE YEAR(SCHULEINTRITT) > 1970;
```

# Liste von Spaltenfunktionen (MySQL)

## **Verknüpfung von Spalten mit typgerechten Operatoren**

z.B. `spalte1+spalte2`, `spalte1/365`

## **Numerische Operationen**

z.B. `truncate(n,d)`

## **Datumsoperationen** DAY, MONTH, YEAR, ADDDATE,

z.B. `YEAR(GEBDATUM)`, `DATEDIFF(datum1,datum2)`

## **Zeichenketten:** Substring, Konkatination,...

z.B. `SUBSTR(Nachname,1,1)`

# Beispiel

## Kurz vor der Pension...

```
SELECT nachname, YEAR(NOW()) - YEAR(gebdatum)
FROM lehrer
WHERE YEAR(NOW()) - YEAR(gebdatum) > 50;
```

## Initialen

```
SELECT
CONCAT(SUBSTR(vorname, 1, 1), '.', SUBSTR(nachname, 1, 1))
FROM lehrer;
```

Für *NOW()* können wir auch die Konstante *CURRENT\_TIMESTAMP* verwenden!  
Für das aktuelle Datum stehen *CURDATE()* und *CURRENT\_DATE* zur Verfügung.

# Bedingungen

## Vergleichsoperatoren

=, <, >, <=, >=, <>

## Boolsche Operatoren

AND, OR

## Bereichsprüfungen

<wert> BETWEEN <wert> AND <wert>

## Mustervergleiche

<wert> like "MUSTER%", <wert> like "MUSTER\_"

## Elementprüfung

<wert> in (<wert>, <wert>, ..., <wert>)

<wert> ist Attribut oder Konstante  
% Wildcard (entspricht: .\*)  
\_ Wildcard (entspricht: .)  
Können mit \ maskiert werden.

## Beispiel

```
select * from lehrer  
  where nachname like "M%";
```

```
select * from lehrer  
  where nachname like "M_ier";
```

```
select * from lehrer  
  where year(gebdatum) between 1960 and 1967;
```

```
select * from lehrer  
  where year(gebdatum) in (1960,1961,1962);
```



# Behandlung von NULL

- Der Wert **NULL** steht für einen undefinierten Wert. Er kann daher nicht mit anderen Werten verglichen werden.
- Für die Selektion muss daher die **IS NULL-Klausel** genutzt werden.

Alle Lehrer, die keinen akademischen Titel haben

```
SELECT nachname  
FROM lehrer  
WHERE titel = NULL;
```

```
SELECT nachname  
FROM lehrer  
WHERE titel = IS NULL;
```

**Negation: IS NOT NULL**

Achtung: hier gibt es keinen Fehler, sondern eine fälschlicherweise leere Ergebnismenge

# Mengenoperationen

Die Vereinigung zweier Ergebnismengen erfolgt mit dem **UNION-Operator**. Die Ergebnisse müssen die gleiche Struktur (d.h. die gleichen Spalten) besitzen!

Für Differenz und Durchschnitt gibt es keine SQL-Anweisung! Sie können mit Hilfe so genannter **Subselects** realisiert werden (kommt später).

## Personenliste

```
SELECT nachname, vorname, gebdatum  
FROM lehrer  
UNION  
SELECT nachname, vorname, gebdatum  
FROM schueler;
```

Das ist eine Menge! Sie enthält keine Duplikate, die aus der Vereinigung resultieren!

# Rename

- Auch SQL erlaubt das Umbenennen von Attributen.
- Hierzu wird in der Projektion einfach hinter dem Attributnamen das Schlüsselwort **AS** gefolgt vom neuen Namen angegeben.

```
SELECT nachname as nname  
FROM lehrer;
```

# Lernziel / Fragen

Umgang mit der Relationenalgebra sowie einfache SQL-Abfragen.

Was ist die Relationenalgebra?

Was ist SQL?

Wie werden Abfragen formuliert?

Wie sind deren Ergebnisse zu interpretieren?