

WIEDERHOLUNG

FUNKTIONALE ABHÄNGIGKEITEN

Sei V eine Attributmenge, $X, Y \subseteq V$ und $r \in \text{Rel}(V)$

Dann ist Y funktional abhängig von X ,
notiert als $X \rightarrow Y$, wenn und nur wenn jeder
 X -Wert von r genau einen eindeutigen Y -Wert
in r bestimmt.

FINDEN SIE DIE FUNKTIONALEN ABHANGIGKEITEN.

-----	-----	-----	-----
a	b	c	d
-----	-----	-----	-----
a1	b1	c1	d1
-----	-----	-----	-----
a1	b1	c2	d2
-----	-----	-----	-----
a2	b1	c1	d3
-----	-----	-----	-----
a2	b1	c3	d4
-----	-----	-----	-----

WELCHE DIESER ZEILEN DURFEN EINGEFUGT WERDEN?

	- - - - -		- - - - -		- - - - -		- - - - -	
	a		b		c		d	
	- - - - -		- - - - -		- - - - -		- - - - -	
	a1		b1		c1		d1	
	- - - - -		- - - - -		- - - - -		- - - - -	
	a1		b1		c2		d2	
	- - - - -		- - - - -		- - - - -		- - - - -	
	a2		b1		c1		d3	
	- - - - -		- - - - -		- - - - -		- - - - -	
	a2		b1		c3		d4	
	- - - - -		- - - - -		- - - - -		- - - - -	

1. a5 b6 c7 d8

2. a2 b2 c1 d8

3. a3 b1 c4 d3

4. a1 b1 c2 d5

NORMALFORMEN

1NF

2NF

3NF

A

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11
1	A	AA	AQ	AD	AM	AP	AL, AT	4, 7	2, 4	36
2	B	BB	AS	AB	AM	AP	AL, AK	3, 5	4, 2	22
3	C	CC	AW	AC	AN	AQ	AK, AT	1, 1	16, 32	48
4	D	DD	AR	AD	AM	AP	AK	6	16	96
5	E	EE	AT	AF	AN	AQ	AL, AK	6, 1	8, 16	64

LOS GEHT'S

\o/

ANFRAGESPRACHEN

KRITERIEN FÜR ANFRAGESPRACHEN

– AD-HOC-FORMULIERUNG

– DESKRIPTIVITÄT

– MENGENORIENTIERUNG

– ABGESCHLOSSENHEIT

– ADÄQUATHEIT

– ORTHOGONALITÄT

– OPTIMIERBARKEIT

– EFFIZIENZ

– SICHERHEIT

– VOLLSTÄNDIGKEIT

ANFRAGEALGEB RA VS ANFRAGEKALKÜ L

ANFRAGEALGEBRA

RELATIONALE ALGEBRA

PROJEKTION π

Person

- - - - -	- - - - -
Nr	Name
- - - - -	- - - - -
1201	Lovelace
- - - - -	- - - - -
1202	Descartes
- - - - -	- - - - -
1203	Euler
- - - - -	- - - - -

$\pi[\text{Name}](\text{Pers})$

- - - - -
Name
- - - - -
Lovelace
- - - - -
Descartes
- - - - -
Euler
- - - - -

SELEKTION σ

Person

- - - - -	- - - - -
Nr	Name
- - - - -	- - - - -
1201	Lovelace
- - - - -	- - - - -
1202	Descartes
- - - - -	- - - - -

$\sigma[\text{Nr} < 1203](\text{Pers})$

- - - - -
Nr
- - - - -
1201
- - - - -
1202
- - - - -

KARTESISCHES PRODUKT \times

Person

Nr	Name
1201	Lovelace
1202	Descartes
1203	Euler

Buch

Id	Preis	Titel
302	13.99	Hamlet
306	14.99	Faust
310	44.99	Die Verwandlung

Person	x	Buch			
Nr	Name	Id	Preis	Titel	
1201	Lovelace	302	13.99	Hamlet	
1201	Lovelace	306	14.99	Faust	
1201	Lovelace	310	44.99	Die Verwandlung	
1202	Descartes	302	13.99	Hamlet	
1202	Descartes	306	14.99	Faust	
1202	Descartes	310	44.99	Die Verwandlung	
1203	Euler	302	13.99	Hamlet	
1203	Euler	306	14.99	Faust	
1203	Euler	310	44.99	Die Verwandlung	

VERBUND (JOIN) 

Person

PNr	Name
1201	Lovelace
1202	Descartes
1203	Euler

Buch

Id	PNr	Preis	Titel
302	1201	13.99	Hamlet
306	1202	14.99	Faust
310	1202	44.99	Die Verwandlung

Person ⌘ Buch

Id	PNr	Preis	Titel	Name
302	1201	13.99	Hamlet	Lovelace
306	1202	14.99	Faust	Descartes
310	1202	44.99	Die Verwandlung	Descartes

UMBENENNUNG β

Person

- - - - -	- - - - -
Nr	Name
- - - - -	- - - - -
1201	Lovelace
- - - - -	- - - - -
1202	Descartes
- - - - -	- - - - -
1203	Euler
- - - - -	- - - - -

β [Personalnummer<-Nr](Pers)

- - - - -	- - - - -
Personalnummer	Name
- - - - -	- - - - -
1201	Lovelace
- - - - -	- - - - -
1202	Descartes
- - - - -	- - - - -
1203	Euler
- - - - -	- - - - -

KARDINALITÄT

KARDINALITÄT EINER PROJEKTION

KARDINALITÄT EINER SELEKTION

KARDINALITÄT EINES KARTESISCHEN PRODUKT

KARDINALITÄT EINES JOINS

ANFRAGEKALKÜL

42 > Preis

Einkaufspreis < Verkaufspreis

SQL


```
CREATE TABLE PERSONAL (  
    PNR INT PRIMARY KEY,  
    VORNAME VARCHAR(20),  
    NACHNAME VARCHAR(20),  
    GEH_STUFE VARCHAR(5),  
    ABT_NR VARCHAR(5),  
    KRANKENKASSE VARCHAR(3)  
);
```

```
CREATE TABLE MASCHINE (  
    MNR INT PRIMARY KEY,  
    NAME VARCHAR(20),  
    PNR INT,  
    ANSCH_DATUM DATE,  
    NEUWERT INT,  
    ZEITWERT INT,  
    FOREIGN KEY (PNR) REFERENCES PERSONAL(PNR)  
);
```

```
SQL> describe MASCHINE;
```

Name	Null?	Type
-----	-----	-----
MNR	NOT NULL	NUMBER(38)
NAME		VARCHAR2(20)
PNR		NUMBER(38)
ANSCH_DATUM		DATE
NEUWERT		NUMBER(38)
ZEITWERT		NUMBER(38)

```
insert into PERSONAL values (123, 'Ada', 'Lovelace', 'it5', 'abt42', 'dek');  
insert into MASCHINE values (1, 'Bohrmaschine', 123, '01-feb-14', 30000, 15000);
```

```
SQL> select * from Personal;
```

PNR	VORNAME	NACHNAME	GEH_S	ABT_N	KRA
123	Ada	Lovelace	it5	abt42	dek

```
SQL> select * from MASCHINE;
```

MNR	NAME	PNR	ANSCH_DAT	NEUWERT	ZEITWERT
1	Bohrmaschine	123	01-FEB-14	30000	15000

```
SELECT [ALL|DISTINCT] ausdruck
FROM {tabellenname|viewname}[aliasname]
    [{tabellenname|viewname}[ aliasname], ...]
[WHERE suchbedingungen]
[GROUP BY nicht aggregierender ausdruck
    [,nicht aggregierender ausdruck]...]
[HAVING suchbedingungen]
[ORDER BY {spaltenname | nummer in spaltenliste} [{ASC|DESC}]
    [, {spaltenname|nummer in spaltenliste} [{ASC|DESC}]]...]
```

-----	-----	-----	-----	-----	-----
MNR	NAME	PNR	ANSCH_DATUM	NEUWERT	ZEITWERT
-----	-----	-----	-----	-----	-----
1	Bohrmaschine	123	01-feb-14	30000	15000
-----	-----	-----	-----	-----	-----
2	Bohrmaschine	123	01-jul-07	30000	18000
-----	-----	-----	-----	-----	-----
3	Fraesmaschine	124	04-jan-13	40000	10000
-----	-----	-----	-----	-----	-----
11	Hobelmaschine	127	15-jan-07	29000	19000
-----	-----	-----	-----	-----	-----
12	Drehbank	126	01-aug-14	31000	21000
-----	-----	-----	-----	-----	-----
14	Hobelmaschine	123	01-nov-13	32000	22000
-----	-----	-----	-----	-----	-----
16	Drehbank	134	25-nov-06	32000	23000
-----	-----	-----	-----	-----	-----
17	Bohrmaschine	127	01-feb-08	31000	25000
-----	-----	-----	-----	-----	-----

**GESUCHT SIND DIE PERSONALNUMMERN DERJENIGEN MITARBEITER,
DIE AN MEHR ALS EINER MASCHINE ARBEITEN, VON DENEN JEDE
EINEN GRÖßEREN ZEITWERT ALS 17.000.- € HAT.
ERGEBNIS IST ABSTEIGEND SORTIERT AUSZUGEBEN.**


```
SELECT PNR
FROM MASCHINE
WHERE ZEITWERT > 17000
GROUP BY PNR
HAVING COUNT(ZEITWERT) > 1
ORDER BY PNR DESC;
```

FROM-KOMPONENTE

FROM MASCHINE

WHERE-KOMPONENTE

WHERE ZEITWERT > 17000

**GROUP BY -
KOMPONENTE**

GROUP BY PNR

- - - - -	- - - - -	- - - - -	- - - - -	- - - - -	- - - - -	- - - - -
MNR	NAME	PNR	ANSCH_DATUM	NEUWERT	ZEITWERT	
- - - - -	- - - - -	- - - - -	- - - - -	- - - - -	- - - - -	- - - - -
2	Bohrmaschine	123	01-jul-07	30000	18000	
14	Hobelmaschine		01-nov-13	32000	22000	
- - - - -	- - - - -	- - - - -	- - - - -	- - - - -	- - - - -	- - - - -
11	Hobelmaschine	127	15-jan-07	29000	19000	
17	Bohrmaschine		01-feb-08	31000	25000	
- - - - -	- - - - -	- - - - -	- - - - -	- - - - -	- - - - -	- - - - -
12	Drehbank	126	01-aug-14	31000	21000	
- - - - -	- - - - -	- - - - -	- - - - -	- - - - -	- - - - -	- - - - -
16	Drehbank	134	25-nov-06	32000	23000	
- - - - -	- - - - -	- - - - -	- - - - -	- - - - -	- - - - -	- - - - -

HAVING-KOMPONENTE

HAVING COUNT(ZEITWERT) > 1

MNR	NAME	PNR	ANSCH_DATUM	NEUWERT	ZEITWERT
2	Bohrmaschine	123	01-jul-07	30000	18000
14	Hobelmaschine		01-nov-13	32000	22000
11	Hobelmaschine	127	15-jan-07	29000	19000
17	Bohrmaschine		01-feb-08	31000	25000

SELECT-KOMPONENTE

SELECT PNR

	- - - - -	
	PNR	
	- - - - -	
	123	
	- - - - -	
	127	
	- - - - -	

ORDER BY-KOMPONENTE

ORDER BY PNR DESC

	- - - - -	
	PNR	
	- - - - -	
	127	
	- - - - -	
	123	
	- - - - -	


```
SELECT Vorname, Nachname  
FROM Personen  
WHERE postleitzahl = '38855'
```

```
SELECT *  
FROM Personen, Bücher  
WHERE Personen.Name = Bücher.Name
```

SELECT

```
SELECT DISTINCT Name  
FROM Person
```


SELECT Buch.ISBN
FROM Buch, Bestellung

FROM

```
SELECT *  
FROM Buch, Prof
```

```
SELECT *  
FROM Buch CROSS JOIN Prof
```

```
SELECT *  
FROM Buch quelle, Buch ziel
```

WHERE

EINFACHER VERGLEICH

OPERATOR (=, <=, >=, <>, <, >)

```
SELECT NAME, VORNAME  
FROM PERSONAL  
WHERE PNR > 200;
```

```
SELECT MNR, NAME  
FROM MASCHINE  
WHERE ZEITWERT+10000<NEUWERT;
```

BEDINGUNGEN MIT AND.
OR. NOT

```
SELECT NAME, VORNAME  
FROM PERSONAL  
WHERE GEH_STUFE='it3' AND KRANKENKASSE='AOK';
```

```
SELECT NAME, VORNAME  
FROM PERSONAL  
WHERE NOT KRANKENKASSE = 'AOK';
```


IN-OPERATOR

A IN (B,C,D)

$\Leftrightarrow (A=B) \text{ OR } (A=C) \text{ OR } (A=D)$

A NOT IN (B,C,D)

$\Leftrightarrow \text{NOT}(A \text{ IN } (B,C,D))$

$\Leftrightarrow (A<>B) \text{ AND } (A<>C) \text{ AND } (A<>D)$

```
SELECT NAME, VORNAME, KRANKENKASSE  
FROM PERSONAL  
WHERE KRANKENKASSE IN ( 'AOK', 'KKH', 'TKK' );
```

BETWEEN-OPERATOR

A BETWEEN B AND C
 $\Leftrightarrow (A \geq B) \text{ AND } (A \leq C)$

```
SELECT MNR, NAME, ZEITWERT FROM MASCHINE  
WHERE ZEITWERT  
BETWEEN 20000 AND 24000;
```

LIKE-OPERATOR

**% STEHT FÜR KEIN, EIN ODER MEHRERE ZEICHEN.
_ STEHT FÜR GENAU EIN ZEICHEN.**


```
SELECT MNR, NAME  
FROM MASCHINE  
WHERE NAME LIKE '%maschine'
```

```
SELECT MNR, NAME  
FROM MASCHINE  
WHERE NAME LIKE '_r%'
```

```
SELECT MNR, NAME  
FROM MASCHINE  
WHERE NAME LIKE 'Drehbank'
```

NULL-OPERATOR

```
SELECT MNR, NAME  
FROM MASCHINE  
WHERE TEL IS NULL
```

BEISPIEL

1. LEGEN SIE EINE TABELLE AN, IN DER SIE PERSONEN SPEICHERN KÖNNEN (DENKEN SIE AN PRIMÄRSCHLÜSSEL)

- > VORNAME**
- > NACHNAME**
- > HERKUNFTSLAND**

2. LEGEN SIE EINE TABELLE AN, DIE EREIGNISSE SPEICHERN KANN (DENKEN SIE AN PRIMÄRSCHLÜSSEL)

- > NAME**
- > JAHR**
- > PERSON**

3. FÜGEN SIE FOLGENDE EINTRÄGE IHREN TABELLEN HINZU:

- > ENTDECKUNG DES FREQUENZSPRUNGVERFAHREN, 1942, HEDY LAMARR, GEB. IN ÖSTERREICH-UNGARN**
- > ENTDECKUNG DER RÖNTGENSTRAHLUNG, 1895, MARIE CURIE GEB. IN POLEN**
- > ERSTE PROGRAMMIERERIN, 1842, ADA LOVELACE GEB. IN GROßBRITANIEN**
- > ERFINDUNG DES SPANNING TREE PROTOCOL, 1990, RADIA PERLMAN GEB. IN USA**

4. FINDEN SIE DIE SELECT-ANWEISUNGEN FÜR FOLGENDE FRAGEN:

- > ORDNEN SIE DIE EREIGNISSE NACH JAHR ABSTEIGEND**
- > GEBEN SIE DIE VORNAMEN DER PERSONEN AUS, DIE AN
EREIGNISSE VOR 1900 BETEILIGT WAREN**
- > GEBEN SIE ALLE HERKUNFTSLÄNDER AUS**



**DAS WAR'S FÜR
HEUTE
#scnr**