



# Datenbanken 1

## Benutzersichten (Views)

---

## 1 Benutzersichten (Views)

---

2 Die Check-Option

3 Bedingt-änderbare Views

4 Nicht-änderbare Views

5 Grenzen von Views

## Ausführbarkeit:

Die Sicht definiert den Ausschnitt der Datenbank, die das Anwendungsprogramm zur Erfüllung seiner Logik benötigt (so **viele Daten** wie nötig).

## Datenschutz:

Es wird nur der Ausschnitt der Datenbank bereitgestellt, den das Programm sehen und verarbeiten darf (so **wenig Daten** wie möglich).

## Vorteile:

- Gesamtstruktur ist ohne Folgen änderbar
- Neue Sichten sind jederzeit definierbar

Benutzersichten (Views) sind in dem DBMS gespeicherte Datenbankabfragen.

Lageraushilfe



View Kundeliste

Kundennummer	Anrede	Vorname	Nachname
SELECT Kundennummer, Anrede, Vorname, Nachname FROM Kunde			

Tabelle Kunde

Kundennummer	Anrede	Vorname	Nachname	Bankleitzahl	Konto
2310	Frau	Lise	Meitner	44350013	9987342
4812	...	...	...	...	...

Sachbearbeiter  
Buchhaltung



View Kontoverbindungen

Kundennummer	Bankleitzahl	Konto
SELECT Kundennummer, Bankleitzahl, Konto FROM Kunde		

Durch eine **Benutzersicht (View)** wird eine SQL-Anfrage als Datenbankobjekt gespeichert. Die Auswertung der gespeicherten Anfrage erfolgt bei jedem Aufruf der Benutzersicht neu. Bei der Viewdefinition entsprechen die Spaltentypen denen der Attribute der Basistabelle. Umbenennung der Attribute über die Attributliste sind möglich.

## Syntax

```
CREATE View <Viewname> [(Attributliste)]  
AS  
    <Select-Anweisung>  
[WITH CHECK OPTION]
```

## Beispiel

```
CREATE View KundenAusDortmund  
    (KNr,      KName,      Vorname, Anrede)  
AS  
    SELECT Kundenummer, Nachname, Vorname, Anrede  
    FROM Kunde  
    WHERE Ort = 'Dortmund'
```

Attribute des Views  
Attribute der Tabelle

Benutzersichten sind gespeicherte Datenbankabfragen (ohne Sortierung).

Definition

```
CREATE View KundenAusDortmund(KNr, Nachname, Vorname, Anrede)
AS
    SELECT Kundennummer, Nachname, Vorname, Anrede
    FROM Kunde
    WHERE Ort = 'Dortmund'
```

Löschen

```
DROP View KundenAusDortmund
```

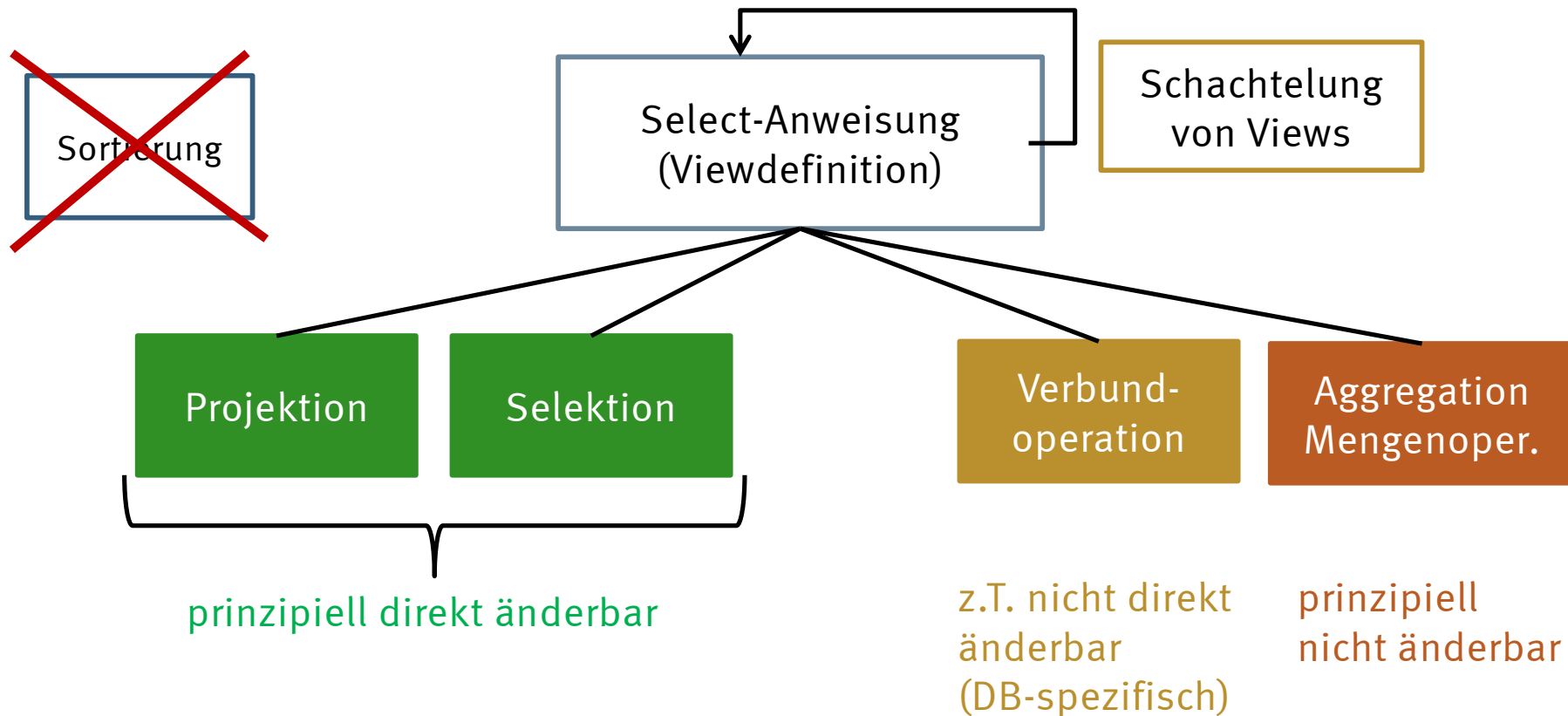
Auswertung:

```
SELECT * FROM KundenAusDortmund
```

Tabelle  
Kunde

Kundennummer	Nachname	Vorname	Anrede	Geburtsdatum	Ort
2310	Meitner	Lise	Frau	17.11.1878	Stockholm
7562	Einstein	Albert	Herr	14.03.1879	Princeton
8523	Dekanat	Informatik	NULL	NULL	Dortmund

Sicht



# Änderungen auf einem einfachen Views

Änderungen auf dem View sind möglich, wenn Tupel eindeutig identifizierbar sind.  
Speziell gilt:

- INSERT
  - Bei fehlenden Werte wird NULL bzw. Standardwert ergänzt
  - Nicht möglich bei fehlendem NOT-NULL Werten ohne Default-Wert
- UPDATE
  - Änderbar sind nur die im View sichtbaren Attribute der Basistabelle
- DELETE
  - Entfernung des gesamten Tupels

Basistabelle

Kunde

<u>Kunden- nummer</u>	Nachname	Vorname	Anrede	Geburts- datum	Ort
2310	Meitner	Lise	Frau	17.11.1878	Berlin
7562	Einstein	Albert	Herr	14.03.1879	Princeton
8523	Dekanat	Informatik	NULL	NULL	Dortmund

Sicht



1 Benutzersichten (Views)

---

2 Die Check-Option

---

3 Bedingt-änderbare Views

4 Nicht-änderbare Views

5 Grenzen von Views

Wird das Skript ausgeführt?

<https://vote.fh.do/W6I3>

- (1) 

```
CREATE View KundenAusDortmund  
AS  
    SELECT Kundennummer, Nachname, Ort FROM Kunde  
    WHERE Ort='Dortmund'
```
- (2) 

```
INSERT INTO KundenAusDortmund VALUES(8525, 'Gross','Bochum')
```
- (3) 

```
DELETE FROM KundenAusDortmund WHERE Kundennummer=8525
```
- (4) 

```
INSERT INTO KundenAusDortmund VALUES(8525, 'Gross','Dortmund')
```

Die WHERE-Bedingung der Benutzersicht wurde beim Insert nicht geprüft!

- (1) CREATE View KundenAusDortmund  
AS  
SELECT Kundennummer, Nachname, Ort FROM Kunde  
WHERE Ort='Dortmund'

KundenAusDortmund

KUNDENUMMER	NACHNAME	ORT
8523	Dekanat Informatik ...	Dortmund ...
8524	Meier ...	Dortmund ...

- (2) INSERT INTO KundenAusDortmund VALUES(8525, 'Gross','Bochum')

1 Zeile eingefügt.

KundenAusDortmund

KUNDENUMMER	NACHNAME	ORT
8523	Dekanat Informatik ...	Dortmund ...
8524	Meier ...	Dortmund ...



Wo ist der neue Kunde Gross?

## Tabelle Kunde:

	⚡ KUNDENUMMER	⚡ ANREDE	⚡ NACHNAME		⚡ VORNAME		⚡ GEBURTSDATUM	⚡ ORT
1	2310	Frau	Meitner	...	Lise	...	17.11.78	Berlin
2	7562	Herr	Einstein	...	Albert	...	14.03.79	Princeton
3	8365	Frau	Curie	...	Marie	...	07.11.67	Paris
4	8523	(null)	Dekanat Informati...	...	FH Dortmund	...	(null)	Dortmund
5	8524	Herr	Meier	...	Max	...	24.12.87	Dortmund
6	8525	(null)	Gross	...	(null)	...	(null)	Bochum

(3) DELETE FROM KundenAusDortmund WHERE Kundennummer=8525

0 Zeilen gelöscht.

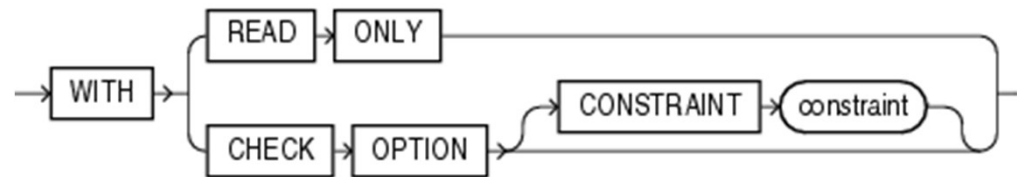
(4) INSERT INTO KundenAusDortmund VALUES(8525, 'Gross','Dortmund')



Weshalb wurde der Kunde Gross nicht gelöscht?

Weshalb wird der Kunde Gross nicht eingefügt?

Bei Änderungen auf einem View wird die WHERE-Bedingung nur dann geprüft, wenn die CHECK-Option vorhanden ist. ist.  
Bei Oracle erfolgt die Prüfung nur für das jeweilige View **kaskadierend**, d.h. die CHECK-Klausel gilt auch für die eingeschachtelten Views.



```
CREATE View <Viewname> [(Attributliste)]  
AS  
    <Select-Anweisung>  
[WITH CHECK OPTION]
```

```
CREATE View KundenAusDortmund2  
(Kundennummer, Name, Ort)  
AS  
    SELECT Kundennummer, Nachname, Ort  
    FROM KUNDE  
    WHERE Ort='Dortmund'  
WITH CHECK OPTION
```

Aufgrund der CHECK OPTION wird die WHERE-Bedingung der Benutzersicht beim Insert geprüft!

- (1) 

```
CREATE View KundenAusDortmund
AS
    SELECT Kundenummer, Nachname, Ort FROM Kunde
    WHERE Ort='Dortmund'
    WITH CHECK OPTION
```
- (2) 

```
INSERT INTO KundenAusDortmund VALUES(8525, 'Gross','Bochum')
```

SQL-Fehler: ORA-01402: Verletzung der WHERE-Klausel einer View  
WITH CHECK OPTION  
01402. 00000 - "view WITH CHECK OPTION where-clause violation"



Gegeben sind folgende Benutzersichten:

```
CREATE View WeiblicheKundenAusDortmund
AS
    SELECT Kundennummer, Nachname, Ort, Anrede
    FROM KundeWeiblich
    WHERE Ort='Dortmund'
```

```
CREATE View KundeWeiblich
AS
    SELECT Kundennummer, Nachname, Ort, Anrede
    FROM KUNDE
    WHERE Anrede='Frau'
WITH CHECK OPTION
```

Was bewirken folgende Änderungsoperationen?

```
INSERT INTO WeiblicheKundenAusDortmund
VALUES (1, 'Test', 'Dortmund', 'Herr');
INSERT INTO WeiblicheKundenAusDortmund
VALUES (2, 'Test', 'Bochum', 'Frau') ;
```

1 Benutzersichten (Views)

2 Die Check-Option

---

3 **Bedingt-änderbare Views**

---

4 Nicht-änderbare Views

5 Grenzen von Views



# Beispiel – View mit Verbundoperation

Bei Verbundoperationen müssen die betroffenen Tupel eindeutig identifiziert werden können.

## Definition

```
CREATE VIEW FirmenkundeWarenkorb
AS SELECT Kundennummer, Artikelnummer, Anzahl
FROM Warenkorb NATURAL JOIN Kunde
WHERE Anrede IS NULL
```

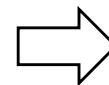
## Abfrage

```
SELECT * FROM FirmenkundenWarenkorb
WHERE Kundennummer=8523
```

## Änderung

```
Update FirmenkundenWarenkorb
SET Anzahl = 3
WHERE Kundennummer=8523      Eindeutige Identifizierung
AND Artikelnummer=4811      des Tupels
```

	KUNDENUMMER	ARTIKELNUMMER	ANZAHL
1	8523	4811	1
2	8523	4812	1
3	8523	4820	1



	KUNDENUMMER	ARTIKELNUMMER	ANZAHL
1	8523	4811	3
2	8523	4812	1
3	8523	4820	1

Bei Verbundoperationen müssen **alle** Schlüssel in der Benutzersicht enthalten und eindeutig sein, damit die betroffenen Tupel eindeutig identifiziert werden können.

Definition:

```
CREATE VIEW Artikelbestand
AS
SELECT Artikelnummer, Artikelname, Lagernummer, Lagerbestand
FROM Artikel a JOIN Lager l ON a.Artikelnummer=l.ANummer
```

Abfrage:

```
SELECT * FROM Artikelbestand
```

ARTIKELNUMMER	ARTIKELNAME	LAGERNUMMER	LAGERBESTAND
4820	Datenbank-Skript	27527	3
4820	Datenbank-Skript	27528	5

Änderung:

```
UPDATE Artikelbestand
SET Artikelname='Datenbanken'
WHERE Artikelnummer=4820
```



SQL-Fehler:

ORA-01779: cannot modify a column which maps to a non key-preserved table

→ (Oracle 11g) Fehler, wenn ein Fremdschlüssel nicht unique (=non key-preserved) ist  
(Ein Artikel kann mehrere Lagerplätze besitzen)

ORA-30926: Stabile Zeilengruppe in den Quelltabellen kann nicht eingelesen werden  
→ Oracle 12g: Fehler bei Duplikaten (Ein Artikel hat mehrere Lagerplätze)

# Vorsicht Falle!

Eine Definition von Benutzersichten ohne Primärschlüssel sowie Änderungen darauf sind prinzipiell möglich.

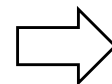
Definition

```
CREATE View Lager_Standort AS
SELECT ANummer, Lagerbestand, Standort
FROM Lager
```

Änderung

```
UPDATE Lager_Standort
SET Lagerbestand=4
WHERE Artikelnummer=4820 Standort fehlt hier!
```

	ARTIKELNUMMER	LAGERBESTAND	STANDORT
1	4811	2	INF
2	4812	18	INF
3	4813	0	FAN
4	4814	1	MED
5	4815	3	MED
6	4816	5	MED
7	4820	3	INF
8	4820	5	MED



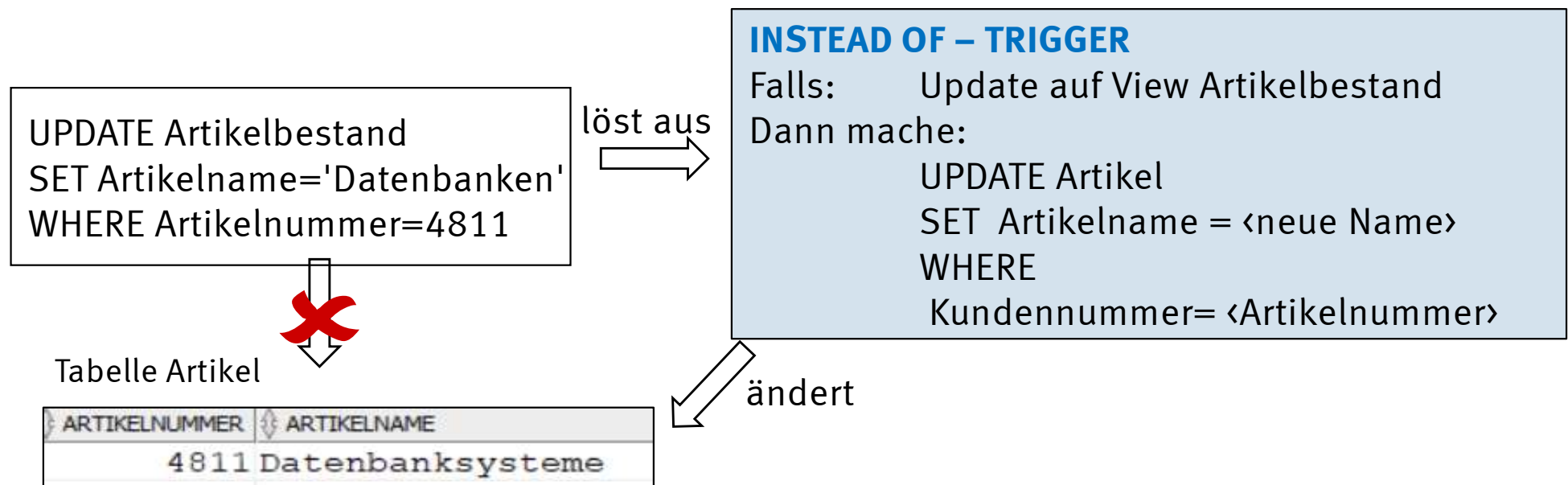
	ARTIKELNUMMER	LAGERBESTAND	STANDORT
1	4811	2	INF
2	4812	18	INF
3	4813	0	FAN
4	4814	1	MED
5	4815	3	MED
6	4816	5	MED
7	4820	4	INF
8	4820	4	MED

Um eindeutig identifizierbare Tupel einer nicht direkt änderbaren Benutzersicht zu ändern, können abhängig vom DBMS Datenbankprogramme erstellt werden, welche die Änderungen auf der Basistabelle durchführen.

Und nun?



Indirekte Änderbarkeit durch Datenbankprogramme  
(ORACLE, IBM DB/2)



- 1 Benutzersichten (Views)
- 2 Die Check-Option
- 3 Bedingt-änderbare Views

---

- 4 Nicht-änderbare Views

---

- 5 Grenzen von Views

# Beispiel – View mit Mengenoperationen

Eine Benutzersicht, bei der Tupel zusammengefasst werden (können), ist nicht änderbar, da dann einzelne Tupel nicht mehr eindeutig identifizierbar sind.

Definition

```
CREATE VIEW KundenAusDortmund
AS
    SELECT Kundenummer, Nachname, Ort
    FROM Kunde
    WHERE Ort='Dortmund' AND Anrede='Frau'

UNION

    SELECT Kundenummer, Nachname, Ort
    FROM Kunde
    WHERE Ort='Dortmund' AND Anrede='Herr'
```

Änderung

```
Update KundenAusdortmund
SET Nachname='Meier'
WHERE Kundenummer=8524
```



SQL-Fehler: ORA-01732: data manipulation operation not legal on this view  
01732. 00000 - "data manipulation operation not legal on this view"

# Beispiel – View mit Aggregationsfunktion

Eine Benutzersicht, welches eine Aggregationsfunktion oder eine Gruppierung verwendet, ist nicht änderbar, da **mehrere** Tupeln zusammengeführt werden.

## Definition

```
CREATE VIEW Artikelanzahl
AS
SELECT Kundennummer, Nachname, COUNT(Anzahl)Anzahl
FROM Warenkorb NATURAL JOIN Kunde
GROUP BY Kundennummer, Nachname
```

## Abfrage

```
SELECT * FROM Artikelanzahl
WHERE Kundennummer=2310
```

KUNDENNUMMER	NACHNAME	ANZAHL
8523	Dekanat Informatik	3

## Änderung

```
UPDATE Artikelanzahl
SET Nachname = 'Dekanat FB4'
WHERE Kundennummer=8523
```



SQL-Fehler: ORA-01732: data manipulation operation not legal on this view  
01732. 00000 - "data manipulation operation not legal on this view"

Prinzipiell ist eine Änderung über die Benutzersicht möglich, wenn

- Tupel der Benutzersicht eindeutig auf Tupel der Basisrelation abzubilden sind. In der Regel ist das der Fall, wenn der (Primär-) Schlüssel vollständig in der Benutzersicht enthalten ist.

Keine Änderung auf dem View ist möglich, bei:

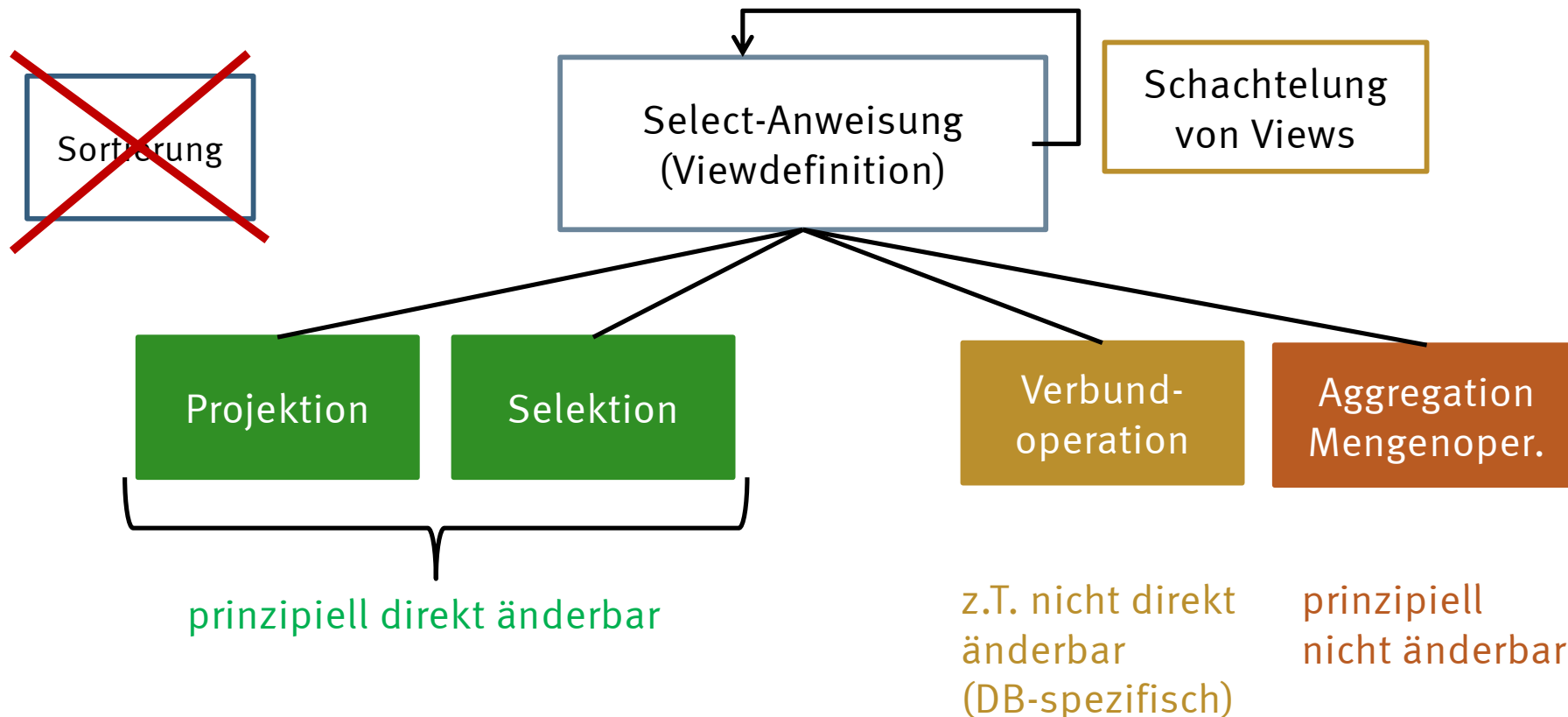
- fehlenden NOT NULL Spalten
- Verwendung von DISTINCT
- Verwendung von Aggregatfunktionen (COUNT, SUM, AVG, MIN und MAX)
- Verwendung von Mengenoperatoren (UNION, INTERSECT und MINUS)
- Gruppierung durch GROUP BY und HAVING
- Ein Unterabfrage in der SELECT-Klausel

Die Änderbarkeit von Views, die Unterabfragen (Subqueries) beinhalten, hängt von dem verwendeten DBMS ab.



Die Änderbarkeit auf einem Views hängt ab von

- der Änderungsoperation
- der Definition des Views (Änderbarkeitskriterien)
- Vorhandensein der Check-Klausel
- dem benutzten relationalen DBMS



Views können ineinander verschachtelt werden. Die Änderbarkeit hängt von der Änderbarkeit der verschachtelten Benutzersichten ab. Entscheidend ist, ob die Tupel der resultierenden Benutzersicht eindeutig identifiziert werden können.

Beispiel: Liefere die Anzahl aller Firmenkunden

```
CREATE View AnzahlFirmenKundenAusDortmund (Anzahl)  
AS
```

```
SELECT count(*)  
FROM KundenAusDortmund  
WHERE Anrede IS NULL
```



```
CREATE View KundenAusDortmund  
AS
```

```
SELECT Kundenummer, Nachname, Vorname, Anrede  
FROM Kunde  
WHERE Ort = 'Dortmund'
```

Sind die Benutzersichten änderbar, bedingt änderbar oder nicht änderbar?

<https://vote.fh.do/W6I3>

```
CREATE View  
AS  
SELECT  
FROM  
WHERE
```

```
CREATE View  
AS  
SELECT  
FROM
```



d, Standort)

- 1 Benutzersichten (Views)
- 2 Die Check-Option
- 3 Bedingt-änderbare Views
- 4 Nicht-änderbare Views

---

- 5 Grenzen von Views

---

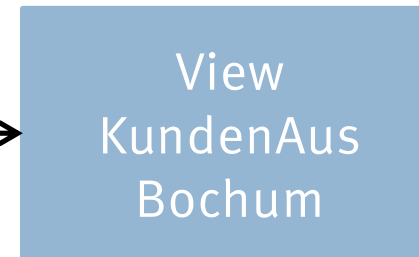
## Vorteile

- **Verfeinerte Rechtevergabe**, bei der die Benutzer nicht mehr direkt auf die Basistabellen zugreifen (Datenschutz)
- Definition **stabiler Schnittstellen**, die im Data Dictionary gespeichert werden (Logische Datenunabhängigkeit)
- Definition **angepasster Datenstrukturen** und Datenmengen
  - Verdichtung von Informationen mit Hilfe von Aggregationsfunktionen
  - Kapselung der Verteilung (Fragmentierung) von Daten über viele Tabellen, d.h. eine Benutzersicht mit Daten aus mehreren verbundenen Tabellen
  - Benutzerfreundlichkeit
- Nachteile
  - Nicht alle Views sind **änderbar**
  - Hoher Aufwand, um bei bedingt änderbaren Benutzersichten die **Änderbarkeit** über **Datenbankprogramme** zu erreichen
  - Schlechte **Wartbarkeit** von verschachtelten Benutzersichten, da Abhängigkeiten aufgelöst werden müssen
  - **Performanceverluste**, da bei Abfragen die Benutzersichten erst aufgelöst werden müssen

# Auch Views haben Grenzen ...



Wohnortwechsel



```
CREATE View KundenAusDortmund(KNr, Nachname, Vorname, Anrede)
AS
SELECT Kundennummer, Nachname, Vorname, Anrede
FROM Kunde
WHERE Ort = 'Dortmund'
```

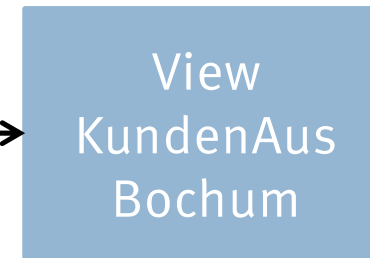
Kunden- nummer	Nachname	Vorname	Anrede	Geburts- datum	Ort
1	Meitner	Lise	Frau	17.11.1878	Stockholm
2	Einstein	Albert	Herr	14.03.1879	Princeton
8523	Dekanat	Informatik	NULL	NULL	Dortmund
8524	Meier	Max	Herr	24.12.1987	<del>Dortmund</del>

Bochum

Eine Änderung des Ortes auf Bochum ist den Sachbearbeiter für Dortmund nicht möglich. Weshalb?



Wohnortwechsel



Kunden- nummer	Nachname	Vorname	Anrede	Geburts- datum	Ort
8524	Meier	Max	Herr	24.12.1987	<del>Dortmund</del>

nicht  
änderbar

änderbar

Rechte

eingeschränkt  
änderbar

nicht  
sichtbar

Bochum

→ Integritätsregeln

## Rechte

-Vergabe von Privilegien für den Zugriff auf ein Schema, eine Tabelle, Attribute einer Tabelle

## Views

- Benutzer greifen nicht direkt auf die Basistabelle zu und sehen nur einen Ausschnitt der Tabelle.
- Einschränkung der Bearbeitungsrechte auf View- oder Attributebene

## Datenbankprogramme

- Änderungsoperationen werden durch Datenbankprogramme vor dem Anwender gekapselt. Die Anwender rufen nur die Datenbankprogramme auf.



## Fragen zur Selbstreflexion

1. Was ist ein View und wie wird es definiert?
2. Wann ist ein Insert- oder Updateoperation auf einem View möglich?
3. Was bewirkt die CHECK-Option mit und ohne Kaskadierung?
4. Wann ist ein View änderbar oder bedingt änderbar?
5. Wie wirkt sich die Verschachtelung von Views auf die Änderbarkeit von Attributen aus?



**Vielen Dank  
für Ihre Aufmerksamkeit**