

Wirtschaftsinformatik 2

LE 06 – Relationales Modell (Teil 3)

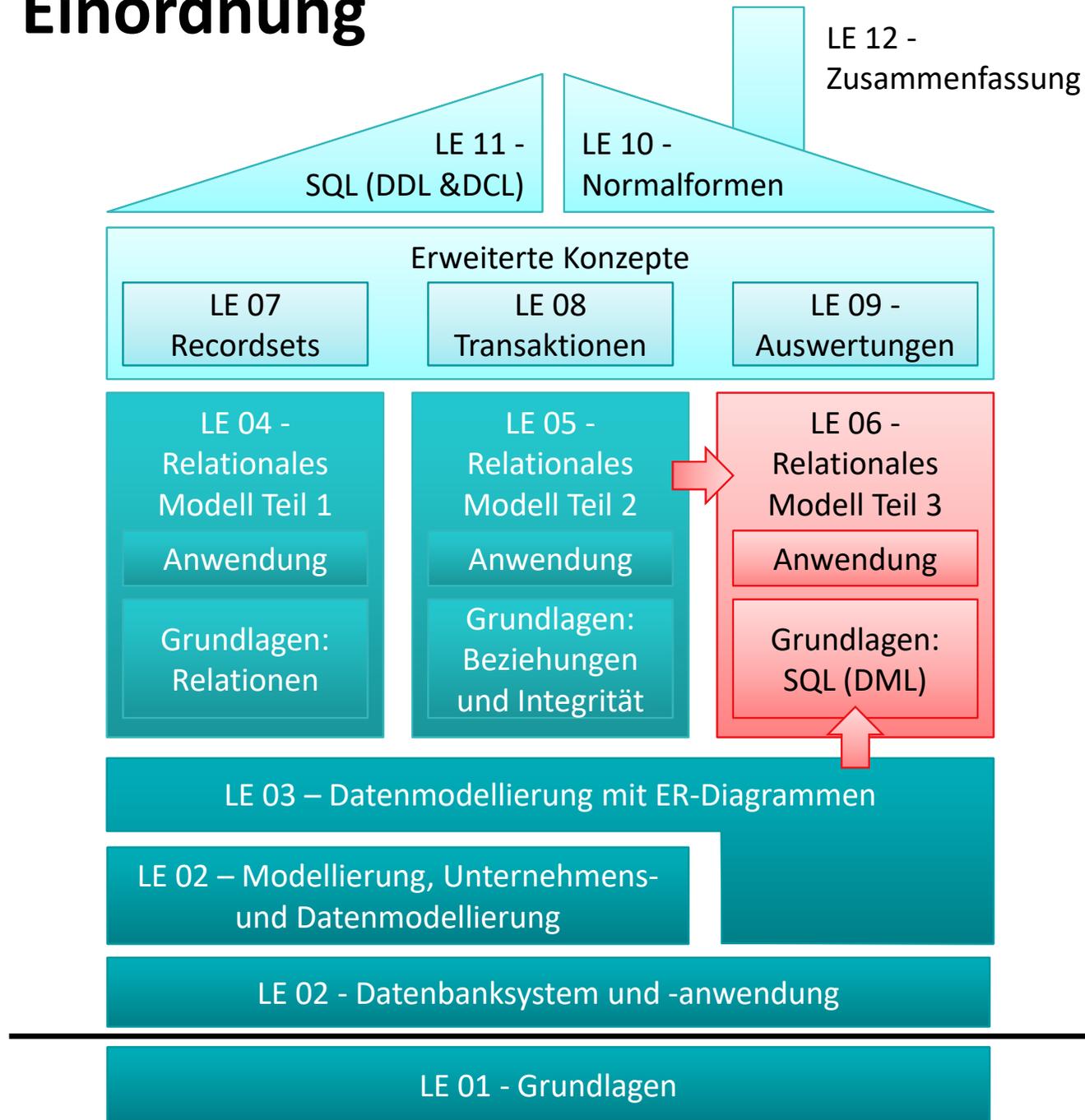
SQL

Prof. Dr. Thomas Off

<http://www.ThomasOff.de/lehre/beuth/wi2>

- Relationale Algebra und ihre wichtigsten Operationen kennenlernen
- Grundbestandteile der Datenbanksprache SQL kennenlernen
- SQL-Anweisungen der Data Manipulation Language (DML) nutzen
 - Abfrage von Daten mit SQL zur Umsetzung von Operationen der Relationalen Algebra
 - Hinzufügen, Löschen und Ändern von Daten mit SQL
- Praktische Anwendung in MS Access üben
 - Verwendung der verschiedenen SQL-Anweisungen in verschiedenen Abfragearten von MS Access
 - Integration von SQL-Anweisungen in eigene Funktionen, Prozeduren und Formulare

Einordnung



Inhalt

Ziel und Einordnung

Rückblick

- Beziehungen im Relationen Modell
- Fremdschlüssel und Referenzielle Integrität

Relationale Algebra und ihre Operationen

SQL kennenlernen

- Bestandteile
- Sprachumfang zum Auswählen, Einfügen, Ändern und Löschen
- Zusammenfassung

SQL in MS Access anwenden

- Formulare und SQL-Auswahlabfragen
- Vordefinierte VBA-Funktionen zum Einfügen, Ändern und Löschen mit SQL
- Zusammenhang und Abgrenzung grafischer Abfragen von SQL-Abfragen
- Zusammenfassung

Ausblick

Rückblick

Beziehungen werden über Fremdschlüssel hergestellt

- ER-Modell



- Relationen

Mitarbeiter	<u>Nr</u>	Name	VName	Verkäufe	<u>MaNr</u>	<u>PrdNr</u>	Produkte	<u>Nr</u>	Name
	123	Huber	Mike		123	23456		23456	KB6314
	234	Mittag	Michael		123	34567		34567	AT1224
	345	Albers	Heidi		345	34567		45678	AT1218
					345	45678		56789	MB4711

Es gibt keinen Fremdschlüssel (ungleich "leer"), dessen Wert im zugehörigen Primärschlüssel nicht existiert (2. Integritätsregel, Referenzielle Integrität)

- was beim Entwurf der Relationen und
- beim Anlegen von Daten bedacht werden muss.

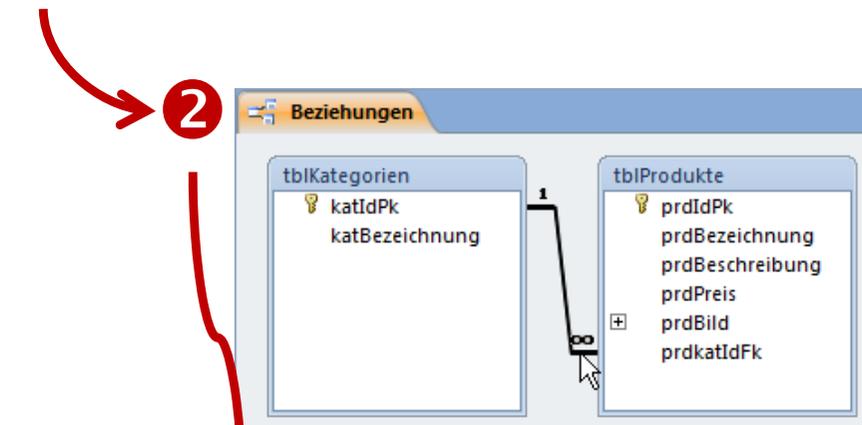
Rückblick

Umsetzung von Beziehungen in MS Access

- Schritt 1: Beziehungen zwischen vorhandenen Relationen werden über zusätzliche Spalten für Fremdschlüssel umgesetzt
- Schritt 2: Anlegen der Beziehung in der Beziehungsansicht, Festlegen der Integritätsbedingungen
- Schritt 3: Erfassen von Daten in der Reihenfolge in der die Beziehung dies erfordert

1

tblProdukte			
Feldname	Felddatentyp	Beschreibung	
prdidPk	AutoWert		
prdBezeichnung	Text	Bezeichnung des Produktes	
prdBeschreibung	Text	Ausführliche Beschreibung der F	
prdPreis	Währung	Verkaufspreis des Produktes	
prdBild	Anlage	Produktfoto	
prdkatIdFk	Zahl	Fremdschlüssel auf Kategorien	



3

tblProdukte		tblKategorien	
katIdPk	katBezeichr	Zum Hinzufügen klicken	
	Garten		
	2 Haushalt		
	(Neu)		

4

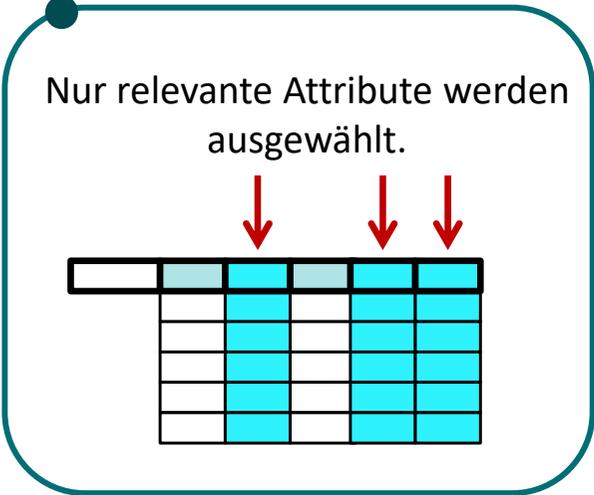
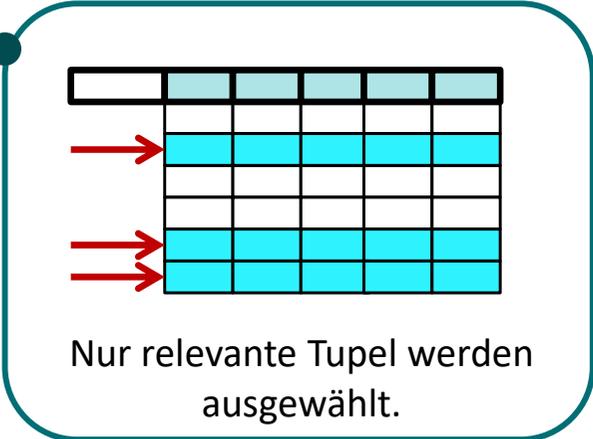
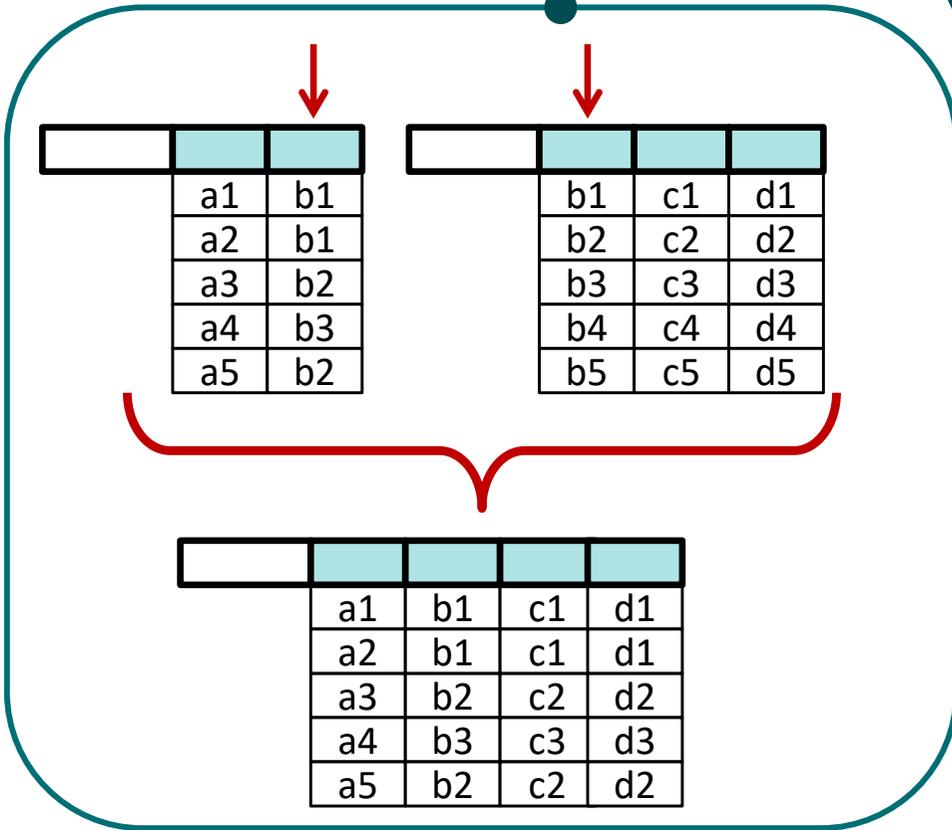
tblProdukte		tblKategorien				
prdidPk	prdBezeichr	prdBeschrei	prdPreis		prdkatIdFk	Zu
	1 Produkt ABC	5 kg für ca. 30 c	23,45 €	0(1)	1	
	2 Produkt DEF	10 kg für ca. 60	45,67 €	0(1)	2	
	3 Produkt GHJ	20 kg für ca. 12	56,78 €	0(1)	2	
*	(Neu)			0(0)		

Rückblick

Selektion (Restriktion)

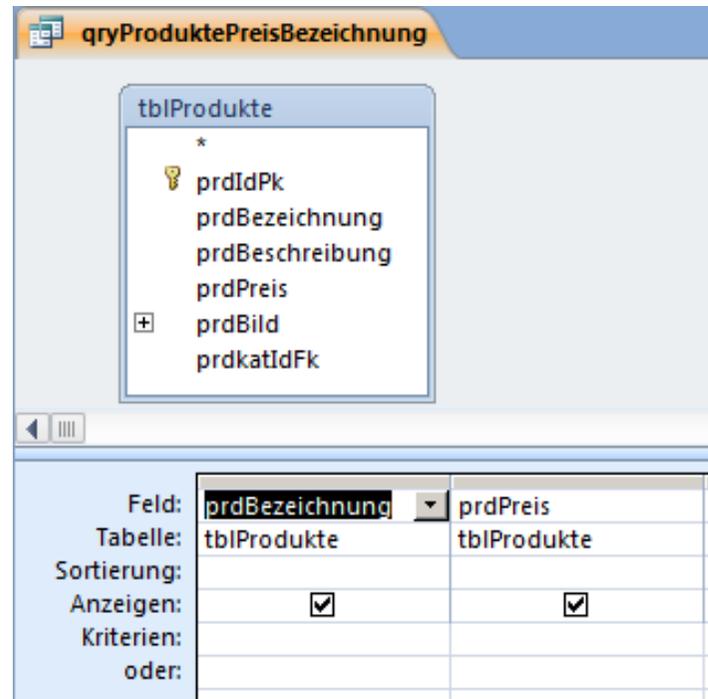
Projektion

Join



Auswählen von Tupeln über grafische Abfragen in Access

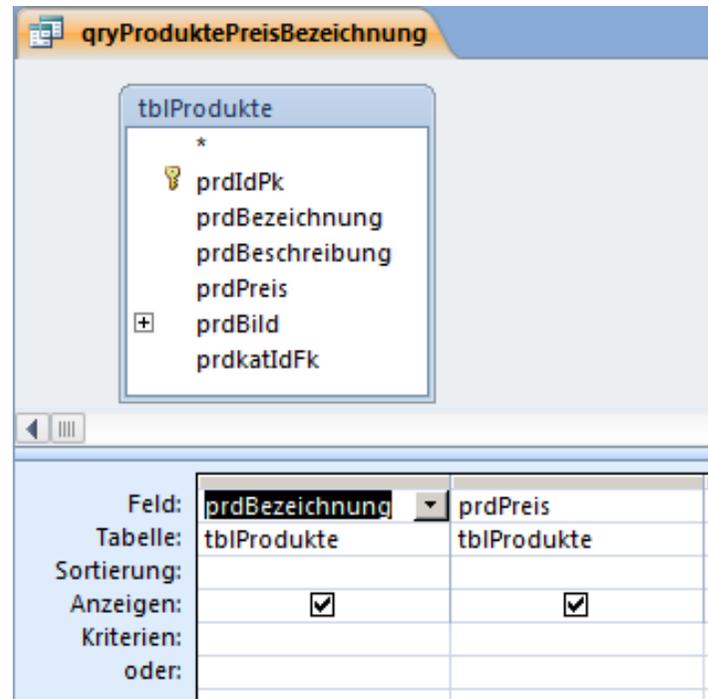
- ...
- ...
- ...



prdBezeichr	prdPreis
Produkt ABC	23,45 €
Produkt DEF	45,67 €
Produkt GHIJ	56,78 €
*	

Auswählen von Tupeln über grafische Abfragen in Access

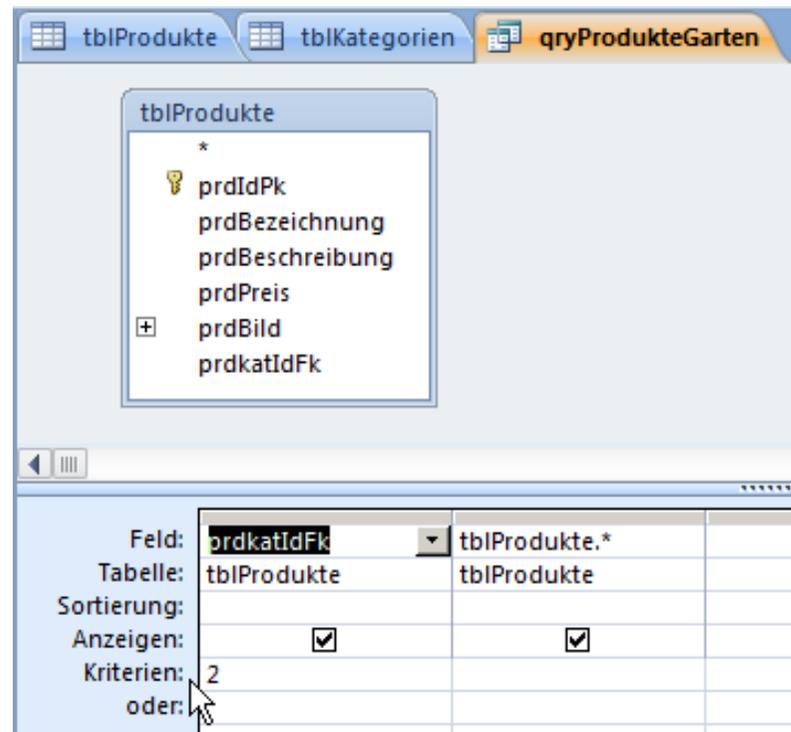
- ...
- Projektion
- ...



prdBezeichr	prdPreis
Produkt ABC	23,45 €
Produkt DEF	45,67 €
Produkt GHIJ	56,78 €
*	

Auswählen von Tupeln über grafische Abfragen in Access

- ...
- Projektion
- ...

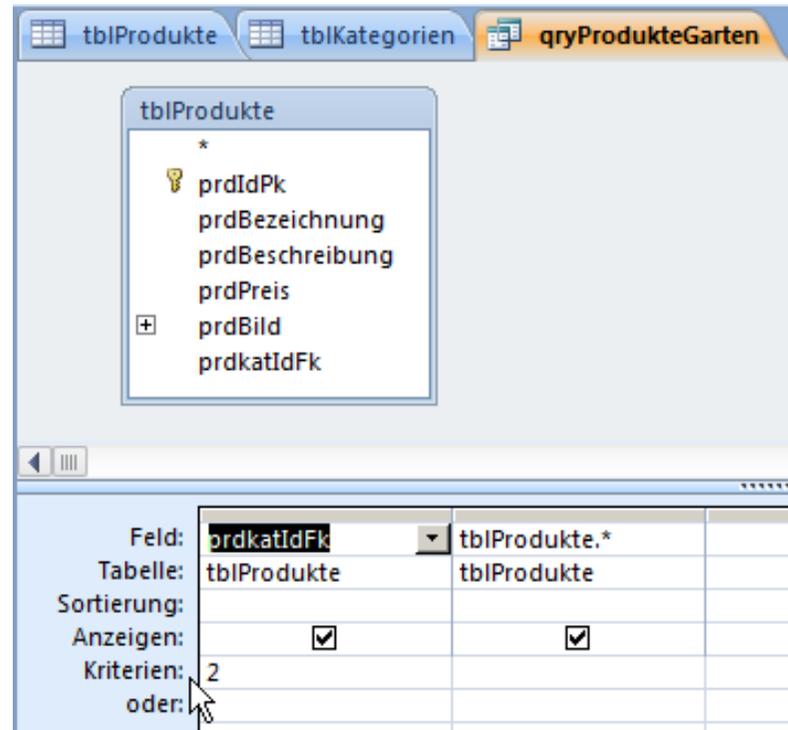


tblProdukte	prdkatIdFk	prdBezeichr	prdBeschrei	prdPreis	Join Icon
2	2	Produkt DEF	10 kg für ca. 60	45,67 €	⌵(1)
2	3	Produkt GHIJ	20 kg für ca. 12	56,78 €	⌵(1)
*	(Neu)				⌵(0)

Rückblick

Auswählen von Tupeln über grafische Abfragen in Access

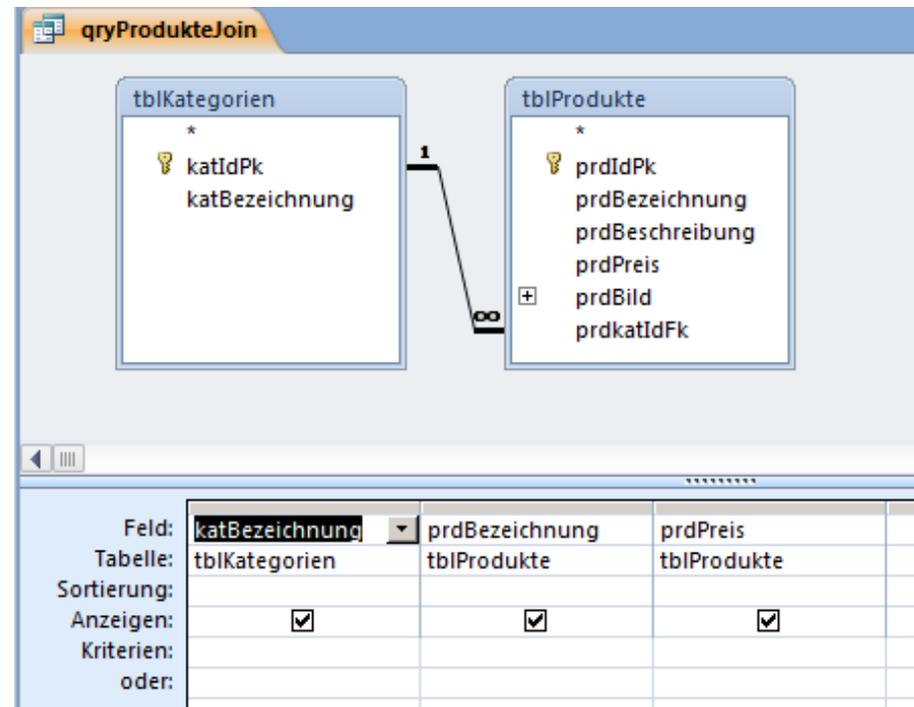
- Restriktion
- Projektion
- ...



tblProdukte	prdIdPk	prdBezeichr	prdBeschrei	prdPreis	
	2	Produkt DEF	10 kg für ca. 60	45,67 €	🔗(1)
	2	3 Produkt GHIJ	20 kg für ca. 12	56,78 €	🔗(1)
*	(Neu)				🔗(0)

Auswählen von Tupeln über grafische Abfragen in Access

- Restriktion
- Projektion
- Join



katBezeichr	prdBezeichr	prdPreis
Garten	Produkt ABC	23,45 €
Haushalt	Produkt DEF	45,67 €
Haushalt	Produkt GHIJ	56,78 €
*		

Rückblick

Hinweis:

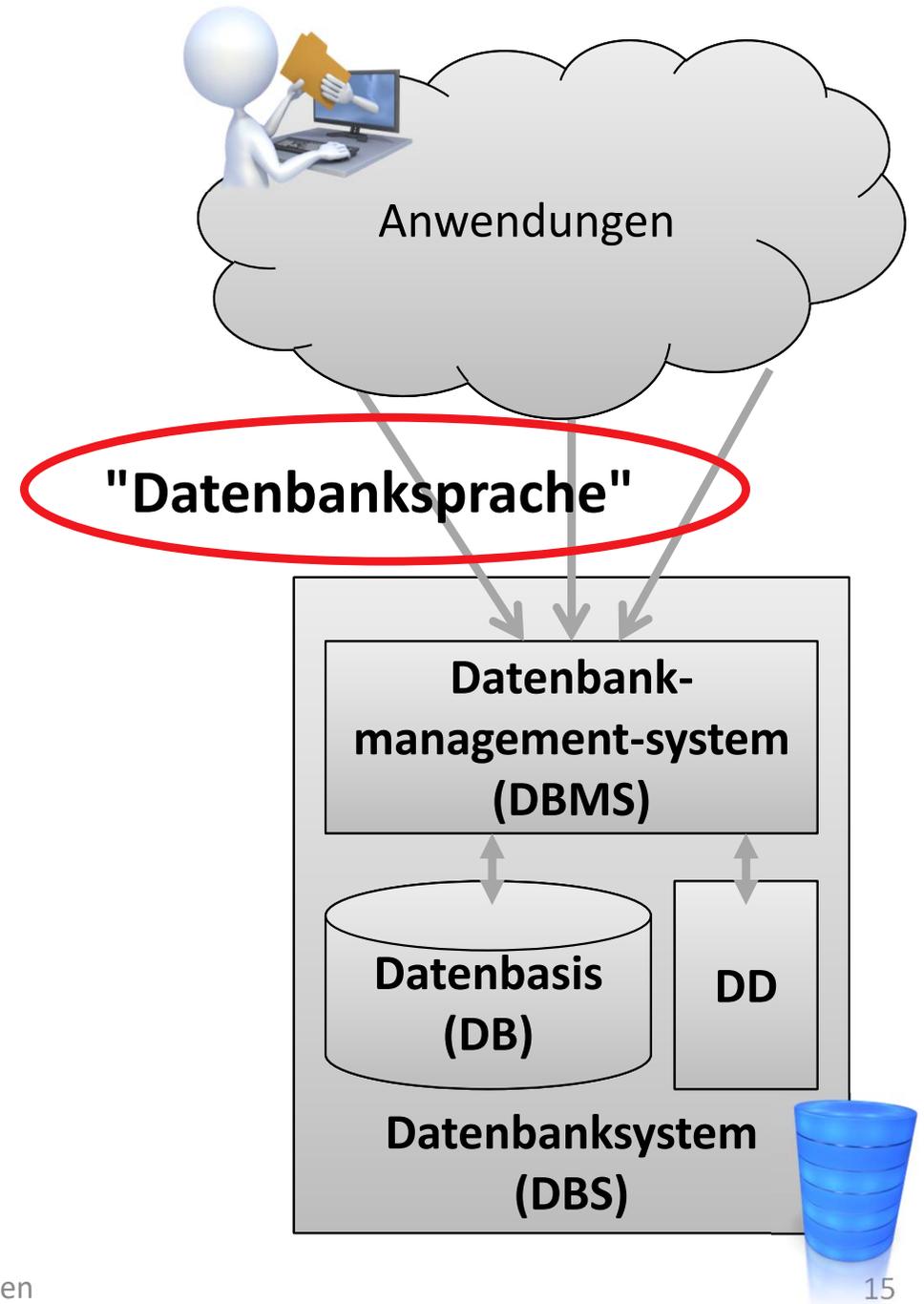
Wenn LE online, dann auch hier schon SQL-Ansicht zeigen!

Rückblick

Komponenten eines Datenbanksystems

- Datenbasis (syn. Datenbank, DB)
- Data Dictionary (DD)
- Datenbankmanagementsystem (DBMS)
- Datenbanksprache

Datenbanksystem besteht aus DBMS + DD + mind. einer DB + Datenbanksprache



Rückblick

Inhalt

Ziel und Einordnung

Rückblick

- Beziehungen im Relationen Modell
- Fremdschlüssel und Referenzielle Integrität

Relationale Algebra und ihre Operationen

SQL kennenlernen

- Bestandteile
- Sprachumfang zum Auswählen, Einfügen, Ändern und Löschen
- Zusammenfassung

SQL in MS Access anwenden

- Formulare und SQL-Auswahlabfragen
- Vordefinierte VBA-Funktionen zum Einfügen, Ändern und Löschen mit SQL
- Zusammenhang und Abgrenzung grafischer Abfragen von SQL-Abfragen
- Zusammenfassung

Ausblick

Relationale Algebra

Relationale Algebra

- ist Bestandteil des Relationalen Modells
- definiert eine Menge von Operationen, mit denen Relationen verknüpft und abgefragt werden können

Operationen der relationalen Algebra

- werden immer auf Relationen angewandt (Ausgangsrelationen)
- erzeugen als Ergebnis eine neue Relation (Ergebnisrelation)
 - ist (im Gegensatz zu Ausgangsrelationen) nicht gespeichert, d.h. nur vorübergehend verfügbar und
 - kann durch erneute Ausführung der Operation wieder erzeugt werden
- können nacheinander (geschachtelt) angewandt werden

Operationen der Relationalen Algebra

Mengenoperationen

- Vereinigung
- Schnitt
- Differenz

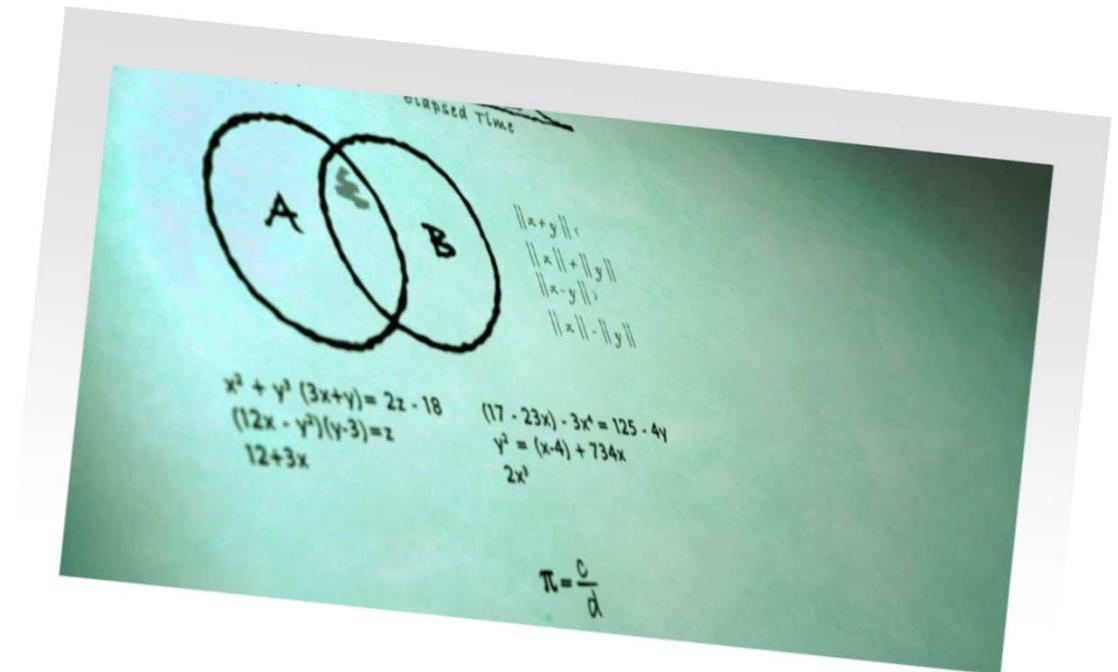
Projektion

Selektion/Restriktion

Kartesisches Produkt

Verbund (Join)

...



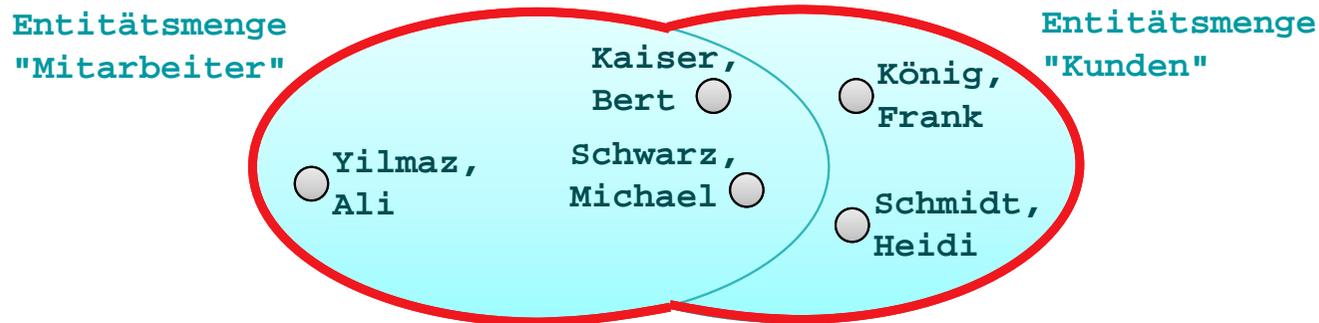
Vereinigung

- Ausgangsrelationen müssen die gleichen Attribute mit passenden Datentypen haben (vereinigungskompatibel)
- Ergebnisrelation
 - hat alle Attribute der Ausgangsrelation und
 - umfasst alle Tupel der Ausgangsrelationen, wobei gleiche Tupel nur einmal enthalten sind
- ...

Mengenoperationen

Vereinigung

- ...
- Beispiel: Sowohl Kunden als auch Mitarbeiter sollen eine Weihnachtskarte bekommen. Die Vereinigungsmenge ist zu bilden.



Ausgangsrelationen

Mitarb	Name	VName
	Kaiser	Bert
	Schwarz	Michael
	Yilmaz	Ali

Kunden	Name	VName
	Kaiser	Bert
	Schwarz	Michael
	König	Frank
	Schmidt	Heidi



Ergebnisrelation

Name	VName
Kaiser	Bert
Schwarz	Michael
Yilmaz	Ali
König	Frank
Schmidt	Heidi

Mengenoperationen

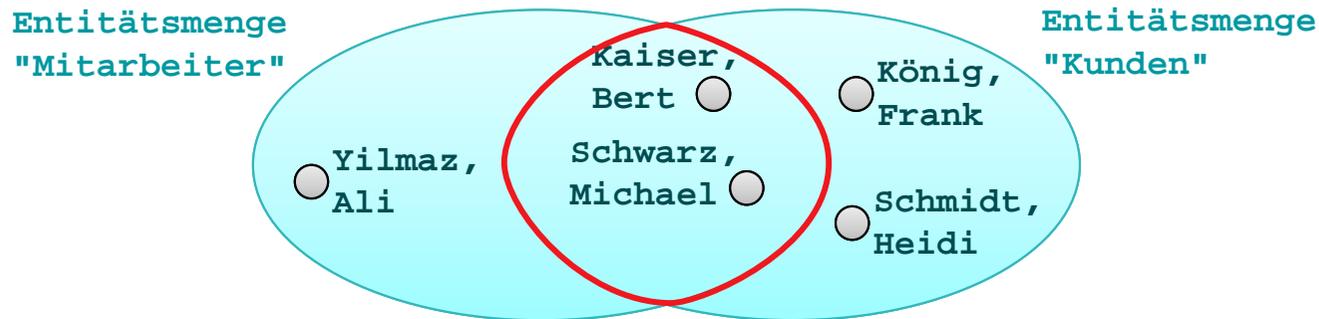
Schnitt

- Ausgangsrelationen müssen die gleichen Attribute mit passenden Datentypen haben (vereinigungskompatibel)
- Ergebnisrelation
 - hat alle Attribute der Ausgangsrelationen,
 - umfasst alle Tupel, die in beiden Relationen enthalten sind
- ...

Mengenoperationen

Schnitt

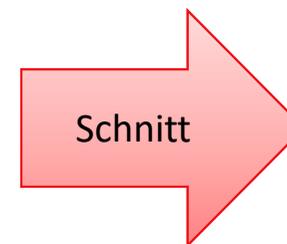
- ...
- Beispiel: Mitarbeiter, die auch Kunden sind, bekommen 10% Rabatt auf ihre Einkäufe am Jahresende ausbezahlt. Die Schnittmenge ist zu bilden.



Ausgangsrelationen

Mitarb	Name	VName
	Kaiser	Bert
	Schwarz	Michael
	Yilmaz	Ali

Kunden	Name	VName
	Kaiser	Bert
	Schwarz	Michael
	König	Frank
	Schmidt	Heidi



Ergebnisrelation

Name	VName
Kaiser	Bert
Schwarz	Michael

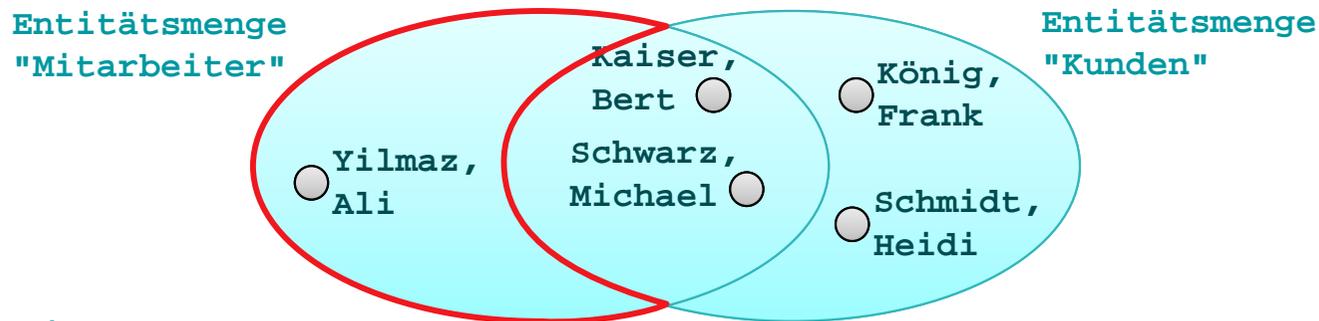
Differenz

- Ausgangsrelationen müssen die gleichen Attribute mit passenden Datentypen haben (vereinigungskompatibel)
- Ergebnisrelation
 - hat alle Attribute der Ausgangsrelationen,
 - umfasst alle Tupel, die in der ersten Relationen enthalten sind und in der zweiten Relation nicht enthalten sind
- ...

Mengenoperationen

Differenz

- ...
- Beispiel 1
 - An alle Mitarbeiter, die noch keine Kunden sind, soll eine Information über den Mitarbeiterrabatt versandt werden. Die Differenz von Mitarbeitern ohne Kunden ist zu bilden.



Ausgangsrelationen

Mitarb	Name	VName
	Kaiser	Bert
	Schwarz	Michael
	Yilmaz	Ali

Kunden	Name	VName
	Kaiser	Bert
	Schwarz	Michael
	König	Frank
	Schmidt	Heidi



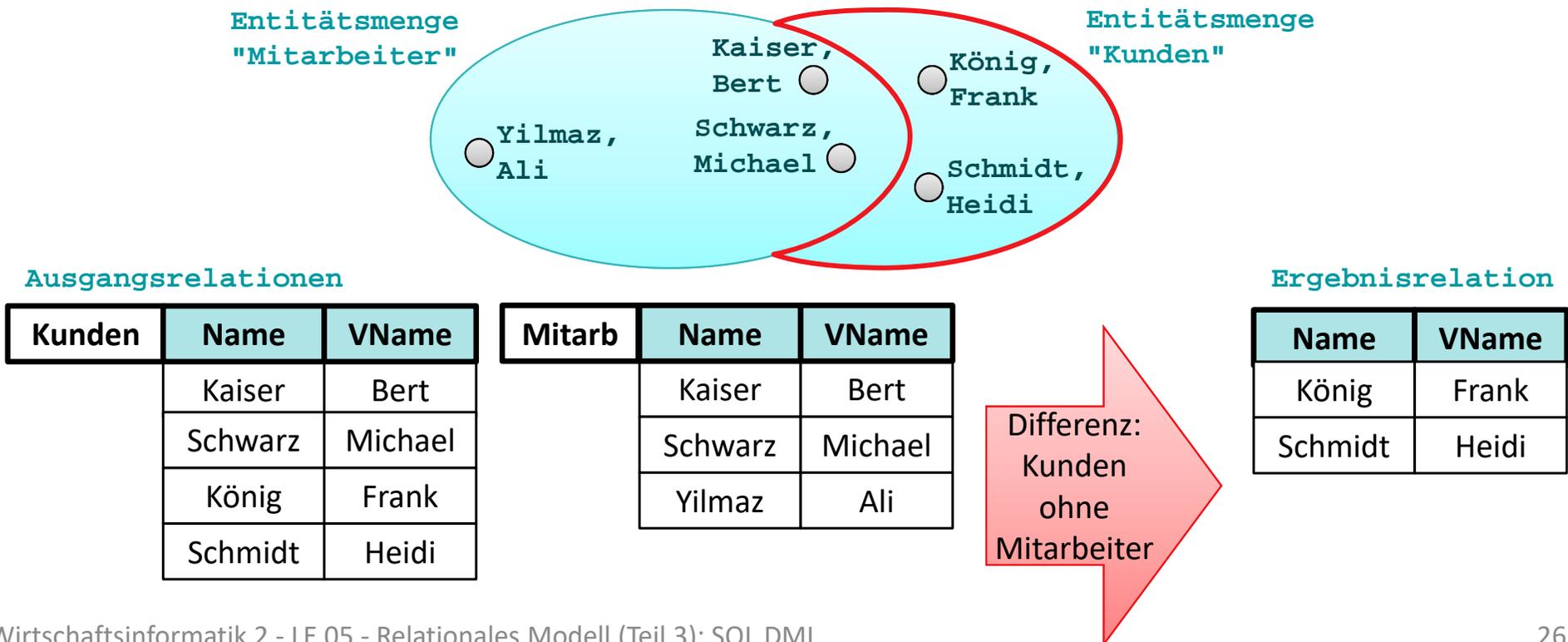
Ergebnisrelation

Name	VName
Yilmaz	Ali

Mengenoperationen

Differenz

- ...
- Beispiel 2
 - Für Kunden gibt es einen Weihnachtsrabatt, der nicht für Mitarbeiter gewährt wird. Die Differenz von Kunden ohne Mitarbeiter ist zu bilden.



Projektion und Selektion (Restriktion)

Projektion

- bestimmt Attribute der Ausgangsrelation, die in der Ergebnisrelation enthalten sein sollen
- Reihenfolge der Attribute in der Ergebnisrelation festgelegt
- entstehen durch Wegfallen von Attributen doppelte Tupel, so werden diese per Definition aus Ergebnisrelation entfernt
- Beispiel: Ort, Name (in dieser Reihenfolge) in Ergebnisrelation

Ausgangsrelationen

Kunden	<u>ID</u>	Name	VName	Plz	Ort	StrasseNr
	9	Kaiser	Bert	12345	Berlin	Bergstr. 8
	8	Schwarz	Michael	12345	Berlin	Schloßstr. 7
	7	Kaiser	Frank	12360	Berlin	Turmstr. 23
	6	Schmidt	Heidi	14482	Potsdam	Badstr. 9

Projektion
(Ort, Name)



Ergebnisrelation

Ort	Name
Berlin	Kaiser
Berlin	Schwarz
Potsdam	Schmidt

Projektion und Selektion (Restriktion)

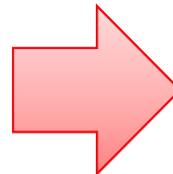
Selektion

- bestimmt die Tupel der Ausgangsrelation, die in der Ergebnisrelation enthalten sein sollen
- Formulierung einer Bedingung, die Attributwerte in Tupeln der Ausgangsrelation erfüllen müssen
- Ergebnisrelation umfasst nur solche Tupel, deren Attributwerte die Bedingung erfüllen
- Beispiel: Alle Kunden aus Berlin.

Ausgangsrelationen

Kunden	ID	Name	VName	Ort
	9	Kaiser	Bert	Berlin
	8	Schwarz	Michael	Berlin
	7	König	Frank	Berlin
	6	Schmidt	Heidi	Potsdam

Selektion
(Ort = Berlin)



Ergebnisrelation

ID	Name	VName	Ort
9	Kaiser	Bert	Berlin
8	Schwarz	Michael	Berlin
7	König	Frank	Berlin

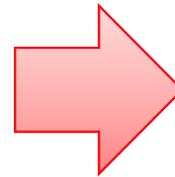
Kartesisches Produkt

Ergebnisrelation hat alle Attribute der Ausgangsrelationen und umfasst alle möglichen Kombinationen von Tupeln der Ausgangsrelationen

Ausgangsrelationen

Namen	Name	VNamen	VName
	Kaiser		Thomas
	Schwarz		Mike
	König		Susi
	Schmidt		

Kartesisches
Produkt



Ergebnisrelationen

Name	VName
Kaiser	Thomas
Kaiser	Mike
Kaiser	Susi
Schwarz	Thomas
Schwarz	Mike
Schwarz	Susi
König	Thomas
König	Mike
König	Susi
Schmidt	Thomas
Schmidt	Mike
Schmidt	Susi

Verbund (Join)

Allgemeine Form (Theta Join)

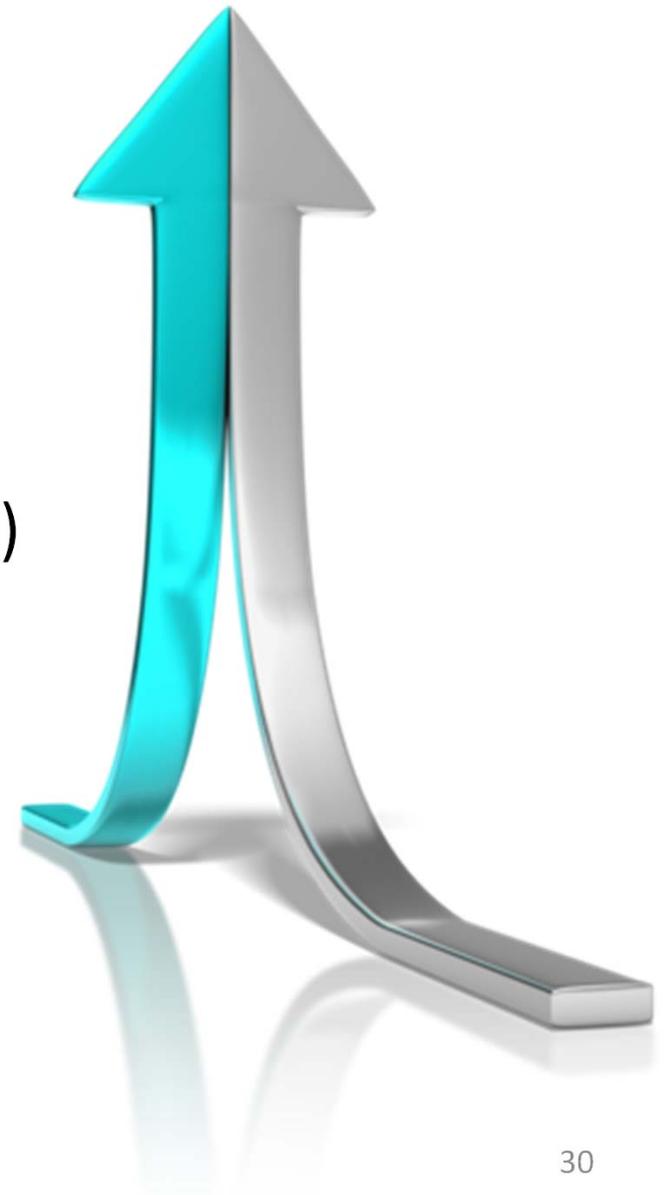
Gleichverbund (Equi Join)

Natürlicher Verbund (Natural Join)

Innerer Verbund (Inner Join)

Äußerer Verbund (Outer Join)

- Linker äußerer Verbund (Left Outer Join)
- Rechter äußerer Verbund (Right Outer Join)



Verbund (Join)

Allgemeine Form (Theta Join)

- Verbindung von Ausgangsrelationen anhand der Werte ausgewählter Attribute
- Formulierung einer Bedingung, die diese Attributwerte erfüllen müssen

Verbund (Join)

Gleichverbund (Equi Join)

- Verbindung von Ausgangsrelationen anhand gleicher Werte der ausgewählten Attribute
- Nur wenn die Werte der ausgewählten Attribute in den Tupeln gleich sind, ist das durch eine Verbindung entstehende neue Tupel in der Ergebnisrelation enthalten
- Häufig wird in der praktischen Anwendung zusätzlich verlangt, dass die am Vergleich beteiligten Attribute gleiche Bezeichnungen haben müssen
- ...

Verbund (Join)

Gleichverbund (Equi Join)

– ...

– Beispiel: Produkte.LID = Lieferanten.LID

Ausgangsrelationen

Produkte	<u>PID</u>	Name	<u>LID</u>
	123	Multi AB	987
	234	Flexi 123	987
	345	Mega+	876
	456	Super XL	

Lieferanten	<u>LID</u>	Firma	Ort
	987	Müller AG	Berlin
	876	Meier GmbH	Potsdam
	765	Bach&Sohn	Cottbus

Ergebnisrelation

PID	Name	LID	LID	Firma	Ort
123	Multi AB	987	987	Müller AG	Berlin
234	Flexi 123	987	987	Müller AG	Berlin
345	Mega+	876	876	Meier GmbH	Potsdam

Verbund (Join)

Natürlicher Verbund (Natural Join)

- Verbindung von Ausgangsrelationen anhand gleicher Werte ausgewählter Attribute (analog zum Gleichverbund)
- die Attribute, deren Werte verglichen wurden, sind nur einmal in der Ergebnisrelation enthalten
- Sehr häufig wird in der praktischen Anwendung zusätzlich verlangt, dass die am Vergleich beteiligten Attribute gleiche Bezeichnungen haben müssen
- ...

Verbund (Join)

Natürlicher Verbund (Natural Join)

– ...

– Beispiel: Beispiel: Produkte.LID = Lieferanten.LID

Ausgangsrelationen

Produkte	<u>PID</u>	Name	<u>LID</u>
	123	Multi AB	987
	234	Flexi 123	987
	345	Mega+	876
	456	Super XL	

Lieferanten	<u>LID</u>	Firma	Ort
	987	Müller AG	Berlin
	876	Meier GmbH	Potsdam
	765	Bach&Sohn	Cottbus

Ergebnisrelation

PID	Name	LID	Firma	Ort
123	Multi AB	987	Müller AG	Berlin
234	Flexi 123	987	Müller AG	Berlin
345	Mega+	876	Meier GmbH	Potsdam

Verbund (Join)

Innerer Verbund (Inner Join)

- Verbindung von Ausgangsrelationen anhand der Werte ausgewählter Attribute
- Formulierung einer Bedingung, die diese Attributwerte erfüllen müssen
- Nur wenn die Werte der ausgewählten Attribute in den verglichenen Tupeln die Bedingung erfüllen, ist das durch eine Verbindung entstehende neue Tupel in der Ergebnisrelation enthalten
- Hinweis: Im Folgenden prüft die Bedingung auf Gleichheit. Es muss aber im Inner Join nicht immer Gleichheit sein! Deshalb erscheinen beide in der Bedingung geprüften Spalten in der Ergebnisrelation.

Verbund (Join)

Innerer Verbund (Inner Join)

– ...

– Beispiel: Produkte.LID = Lieferanten.ID

Ausgangsrelationen

Produkte	<u>PID</u>	Name	<u>LID</u>
	123	Multi AB	987
	234	Flexi 123	987
	345	Mega+	876
	456	Super XL	

Lieferanten	<u>ID</u>	Firma	Ort
	987	Müller AG	Berlin
	876	Meier GmbH	Potsdam
	765	Bach&Sohn	Cottbus

Ergebnisrelation

PID	Name	LID	ID	Firma	Ort
123	Multi AB	987	987	Müller AG	Berlin
234	Flexi 123	987	987	Müller AG	Berlin
345	Mega+	876	876	Meier GmbH	Potsdam

Verbund (Join)

Äußerer Verbund (Outer Join)

- Verbindung von Ausgangsrelationen anhand der Werte ausgewählter Attribute
- Formulierung einer Bedingung, die diese Attributwerte erfüllen müssen
- Im Ergebnis sind mindestens die Tupel einer der beteiligten Relation enthalten und wurden ergänzt, um
 - Attributwerte aus der anderen Relation, wenn die Attributwerte der verglichenen Tupel die Bedingung erfüllen
 - NULL-Werte (leere Attributwerte), wenn die Attributwerte der verglichenen Tupel die Bedingung nicht erfüllen
- Es werden der linke und der rechte äußere Verbund unterschieden (nächste Folie).

Verbund (Join)

Linker äußerer Verbund (Left Outer Join)

- Mindestens die Tupel der linken Relation im Ergebnis vorhanden; wenn sie keinen Partner haben, dann mit NULL
- Beispiel: Produkte.LID = Lieferanten.ID

Ausgangsrelationen

Produkte	<u>PID</u>	Name	<u>LID</u>
	123	Multi AB	987
	234	Flexi 123	987
	345	Mega+	876
	456	Super XL	

Lieferanten	<u>ID</u>	Firma	Ort
	987	Müller AG	Berlin
	876	Meier GmbH	Potsdam
	765	Bach&Sohn	Cottbus
	654	Berg GmbH	Berlin

Ergebnisrelation

PID	Name	LID	ID	Firma	Ort
123	Multi AB	987	987	Müller AG	Berlin
234	Flexi 123	987	987	Müller AG	Berlin
345	Mega+	876	876	Meier GmbH	Potsdam
456	Super XL				

Mindestens
alle linken
Tupel



Verbund (Join)

Rechter äußerer Verbund (Right Outer Join)

- Mindestens die Tupel der rechten Relation im Ergebnis vorhanden; wenn sie keinen Partner haben, dann mit NULL
- Beispiel: Produkte.LID = Lieferanten.ID

Ausgangsrelationen

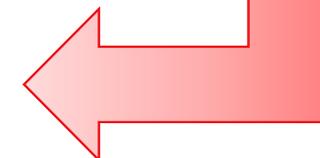
Produkte	<u>PID</u>	Name	<u>LID</u>
	123	Multi AB	987
	234	Flexi 123	987
	345	Mega+	876
	456	Super XL	

Lieferanten	<u>ID</u>	Firma	Ort
	987	Müller AG	Berlin
	876	Meier GmbH	Potsdam
	765	Bach&Sohn	Cottbus
	654	Berg GmbH	Berlin

Ergebnisrelation

PID	Name	LID	ID	Firma	Ort
123	Multi AB	987	987	Müller AG	Berlin
234	Flexi 123	987	987	Müller AG	Berlin
345	Mega+	876	876	Meier GmbH	Potsdam
			765	Bach&Sohn	Cottbus
			654	Berg GmbH	Berlin

**Mindestens
alle rechten
Tupel**



Verbund (Join)

Hinweise

- Die vorherigen Darstellungen zeigten immer
 - zwei Tabellen, der Verbund ist aber analog auch mit mehreren Tabellen möglich (indem er zunächst für die ersten zwei Tabellen durchgeführt wird und das Ergebnis mit der nächsten Tabelle verbunden wird usw.)
 - die Verbindung anhand einer Bedingung für Werte von zwei Attributen, mehrere Bedingungen und mehrere Attribute sind auch möglich
- Es gibt weitere Arten des Verbunds, die hier nicht relevant sind
 - Auto-/Self Join einer Relation mit sich selbst (z.B. Person mit ihrem Ehepartner, der ebenfalls als Person gespeichert ist)
 - Semi Join: Natürlicher Verbund ohne die zusätzlichen Attribute der rechten Relation
 - ...

Verbund (Join)

Allgemeine Form (Theta Join)

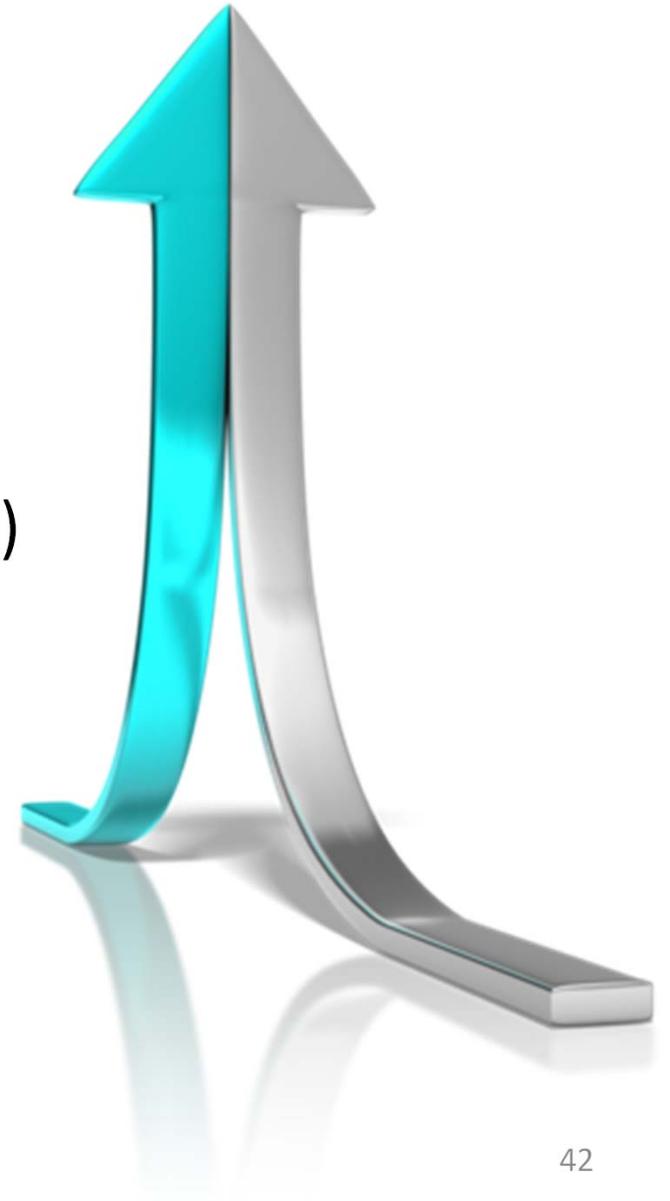
Gleichverbund (Equi Join)

Natürlicher Verbund (Natural Join)

Innerer Verbund (Inner Join)

Äußerer Verbund (Outer Join)

- Linker äußerer Verbund (Left Outer Join)
- Rechter äußerer Verbund (Right Outer Join)



Operationen der Relationalen Algebra

Mengenoperationen

- Vereinigung
- Schnitt
- Differenz

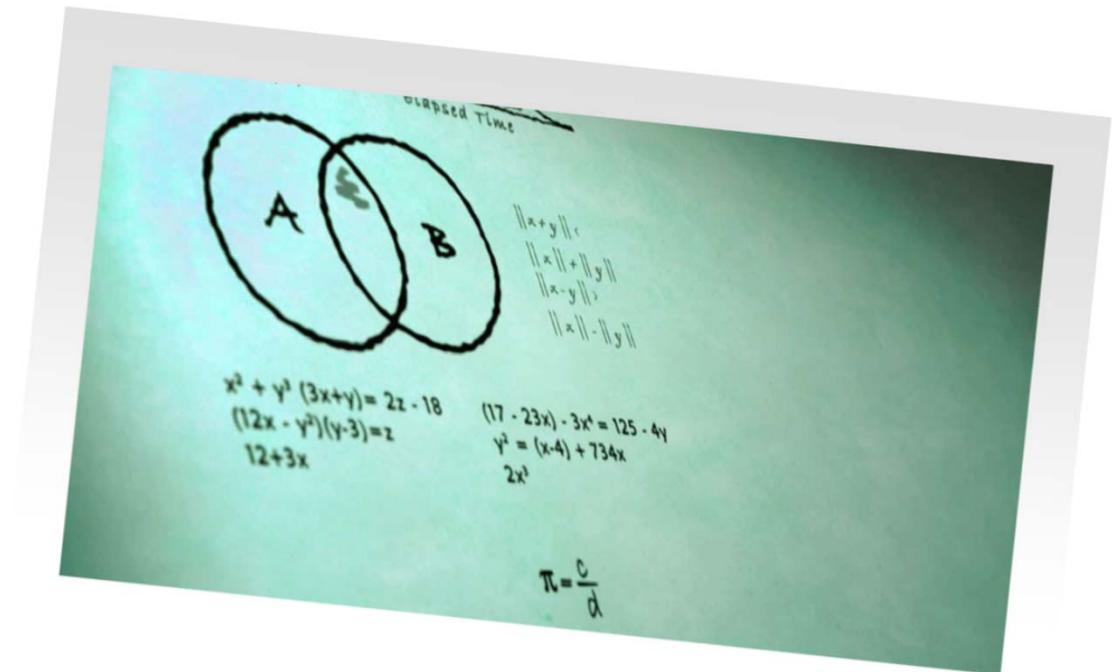
Projektion

Selektion/Restriktion

Kartesisches Produkt

Verbund (Join)

...



Inhalt

Ziel und Einordnung

Rückblick

- Beziehungen im Relationen Modell
- Fremdschlüssel und Referenzielle Integrität

Relationale Algebra und ihre Operationen

SQL kennenlernen

- Bestandteile
- Sprachumfang zum Auswählen, Einfügen, Ändern und Löschen
- Zusammenfassung

SQL in MS Access anwenden

- Formulare und SQL-Auswahlabfragen
- Vordefinierte VBA-Funktionen zum Einfügen, Ändern und Löschen mit SQL
- Zusammenhang und Abgrenzung grafischer Abfragen von SQL-Abfragen
- Zusammenfassung

Ausblick

Bestandteile des SQL-Sprachumfangs

Data Manipulation Language (DML): dient zur Abfrage, zum Hinzufügen, zur Veränderung und zum Löschen von Daten

- SELECT
- INSERT
- UPDATE
- DELETE

Data Definition Language (DDL): dient zum Erzeugen, Verändern und Löschen der Strukturen, die für die Speicherung der Daten benutzt werden (z.B. Tabellen, Spalten)

Data Control Language (DCL): dient zum Einrichten, Festlegen und Entziehen von Zugriffsrechten für Benutzer und Gruppen auf den Strukturen und auf Aktionsmöglichkeiten der Datenbank

Auswählen

- vorhandene Tupel werden gelesen
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel bereitgestellt werden

Einfügen

- neue Tupel werden zur Relation hinzugefügt

Ändern

- vorhandene Tupel werden geändert
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel zu ändern sind

Löschen

- vorhandene Tupel werden gelöscht
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel zu ändern sind

Auswählen von Tupeln mit SQL

**SQL-Syntax ähnlich der natürlichen Sprache (in Englisch),
d.h. es kann bspw. formuliert werden**

- "Wähle alles aus Tabelle *Kunden*."
- "Wähle *Name, Vorname* aus Tabelle *Kunden*." (Projektion)
- "Wähle alle aus der Tabelle *Kunden* bei denen der *Ort 'Berlin'* ist." (Selektion)
- "Wähle *Gesamtpreis* der *Bestellung* und *Name* sowie *Vorname* des *Kunden*, der die Bestellung bestellt hat." (Join)

Auswählen von Tupeln mit SQL

Allgemeine Abfrage in SQL (Bsp. 1):

- Deutsch: "Wähle alle Spalten aus Tabelle *Kunden*."
- Englisch: "Select all Columns from table *Kunden*."
- SQL: **SELECT * FROM *kunden*;**

Ausgangsrelation

Kunden	<u>KnNr</u>	Name	Vorname	Ort
	123	Albers	Willi	Aachen
	234	Boehrs	Vera	Berlin
	345	Dinkel	Ulrike	Berlin
	456	Dinkels	Thomas	Berlin
	567	Esser	Thomas	Dessau

SELECT * FROM Kunden;



Ergebnis

KnNr	Name	Vorname	Ort
123	Albers	Willi	Aachen
234	Boehrs	Vera	Berlin
345	Dinkel	Ulrike	Berlin
456	Dinkel	Thomas	Berlin
567	Esser	Thomas	Dessau

Auswählen von Tupeln mit SQL

Projektion in SQL (Bsp. 2):

- Deutsch: "Wähle *Name, Vorname* aus Tabelle *Kunden*."
- Englisch: "Select *Name, Vorname* from table *Kunden*."
- SQL: **SELECT name, vorname FROM kunden;**

Ausgangsrelation

Kunden	<u>KndNr</u>	Name	Vorname	Ort
	123	Albers	Willi	Aachen
	234	Boehrs	Vera	Berlin
	345	Dinkel	Ulrike	Berlin
	456	Dinkels	Thomas	Berlin
	567	Esser	Thomas	Dessau

↑
↑
Projektion

```
SELECT Name, Vorname FROM Kunden;
```

Ergebnis

Name	Vorname
Albers	Willi
Boehrs	Vera
Dinkel	Ulrike
Dinkels	Thomas
Esser	Thomas

Auswählen von Tupeln mit SQL

Selektion in SQL (Bsp. 3):

- Deutsch: "Wähle alle Spalte aus der Tabelle *Kunden* bei denen der Ort 'Berlin' ist."
- Englisch: "Select all columns from table *Kunden* where Ort is 'Berlin'."
- SQL: **SELECT * FROM kunden
WHERE ort='Berlin';**

Ausgangsrelation

Kunden	<u>KndNr</u>	Name	Vorname	Ort
	123	Albers	Willi	Aachen
	234	Boehrs	Vera	Berlin
	345	Dinkel	Ulrike	Berlin
	456	Dinkels	Thomas	Berlin
	567	Esser	Thomas	Dessau

```
SELECT * FROM Kunden WHERE Ort='Berlin';
```

Selektion

Ergebnis

KndNr	Name	Vorname	Ort
234	Boehrs	Vera	Berlin
345	Dinkel	Ulrike	Berlin
456	Dinkels	Thomas	Berlin

Auswählen von Tupeln mit SQL

Hinweis:

Wenn LE online, dann hier analog zu Folien 101 bis 105 die SQL-Ansicht einführen und folgende Beispiele zeigen

- Ermittle alle Bestellungen vom 11.10.2013
- Ermittle alle Bestellungen, des Kunden Yilmaz
- Ermittle alle Bestellungen, mit einem Preis von mind. 200 EUR
- Ermittle alle Bestellungen, die bereits geliefert wurden
- Ermittle alle Bestellungen, die keine Lieferadresse haben

Anschließend Übungen Ü6.1 Teil 1 inkl. Fortsetzung

Auswählen von Tupeln mit SQL

Innerer Verbund (Inner Join) in SQL (Bsp. 5):

- *Name* sowie *Vorname* des Kunden, der eine Bestellung bestellt hat und den *Gesamtpreis* der Bestellung.
- SQL (Variante 1):

```
SELECT kunden.name, kunden.vorname, bestellungen.gesamt
FROM kunden, bestellungen
WHERE kunden.kndnr = bestellungen.kndnr;
```

Ausgangsrelationen

Kunden	<u>KndNr</u>	Name	Vorname	Ort
	123	Albers	Willi	Aachen
	234	Boehrs	Vera	Berlin
	345	Dinkel	Ulrike	Berlin
	456	Dinkels	Thomas	Berlin
	567	Esser	Thomas	Dessau

Join

Bestellungen	<u>BstID</u>	<u>KndNr</u>	Gesamt	Datum
	987	456	80€	1.1.2012
	876	345	320€	7.3.2011
	765	234	120€	1.8.2011

```
SELECT
kunden.name,
kunden.vorname
bestellungen.gesamt
FROM
bestellungen, kunden
WHERE
kunden.kndnr =
bestellungen.kndnr;
```

Ergebnis

Name	Vorname	Gesamt
Boehrs	Vera	120€
Dinkel	Ulrike	320€
Dinkels	Thomas	80€

Auswählen von Tupeln mit SQL

Innerer Verbund (Inner Join) in SQL (Bsp. 5):

- *Name* sowie *Vorname* des Kunden, der eine Bestellung bestellt hat und den *Gesamtpreis* der Bestellung.
- SQL (Variante 2):

```
SELECT kunden.name, kunden.vorname, bestellungen.gesamt  
FROM kunden INNER JOIN bestellungen  
ON kunden.kndnr = bestellungen.kndnr;
```

Ausgangsrelationen

Kunden	<u>KndNr</u>	Name	Vorname	Ort
	123	Albers	Willi	Aachen
	234	Boehrs	Vera	Berlin
	345	Dinkel	Ulrike	Berlin
	456	Dinkels	Thomas	Berlin
	567	Esser	Thomas	Dessau

Join

Bestellungen	<u>BstID</u>	<u>KndNr</u>	Gesamt	Datum
	987	456	80€	1.1.2012
	876	345	320€	7.3.2011
	765	234	120€	1.8.2011

```
SELECT  
kunden.name,  
kunden.vorname,  
bestellungen.gesamt  
FROM kunden  
INNER JOIN  
bestellungen  
ON kunden.kndnr =  
bestellungen.kndnr;
```

Ergebnis

Name	Vorname	Gesamt
Boehrs	Vera	120€
Dinkel	Ulrike	320€
Dinkels	Thomas	80€

Auswählen von Tupeln mit SQL

Linker äußerer Verbund (Left Outer Join) in SQL (Bsp. 5):

- Alle Lieferadressen mit Straße, Ort und ggf. vorhandene Bestellungen mit ID, Datum
- SQL:

```
SELECT LAdr.StraßeNr, LAdr.PlzOrt, Bestell.BstId, Bestell.Datum  
FROM LAdr LEFT JOIN Bestell  
ON LAdr.AdrNr = Bestell.AdrNr;
```

Ausgangsrelationen

LAdr	AdrNr	StraßeNr	PlzOrt
	123	Bergstr 7	14476 Potsdam
	234	Badstr 1a	12345 Berlin
	345	Schloßstr 8	12456 Berlin

Bestell	BstID	AdrNr	Liefern	Gesamt	Datum
	987	234	J	80€	1.1.2012
	876		N	320€	7.3.2011
	765	345	J	120€	1.8.2011

Linke
Tabelle

Ergebnisrelation

StraßeNr	PlzOrt	BstID	Datum
Bergstr 7	14476 Potsdam		
Badstr 1a	12345 Berlin	987	1.1.2012
Schloßstr 8	12456 Berlin	765	1.8.2011

Auswählen von Tupeln mit SQL

Rechter äußerer Verbund (Right Outer Join) in SQL (Bsp. 5):

- Alle *Bestellungen* mit *ID*, *Datum* und ggf. vorhandene *Lieferadressen* mit *Straße*, *Ort*
- SQL:

```
SELECT LAdr.StraßeNr, LAdr.PlzOrt, Bestell.BstID, Bestell.Datum  
FROM LAdr RIGHT JOIN Bestell  
ON LAdr.AdrNr = Bestell.AdrNr;
```

Rechte Tabelle

Ausgangsrelationen

LAdr	AdrNr	StraßeNr	PlzOrt
	123	Bergstr 7	14476 Potsdam
	234	Badstr 1a	12345 Berlin
	345	Schloßstr 8	12456 Berlin

Bestell	BstID	AdrNr	Liefern	Gesamt	Datum
	987	234	J	80€	1.1.2012
	876		N	320€	7.3.2011
	765	345	J	120€	1.8.2011

Ergebnisrelation

StraßeNr	PlzOrt	BstID	Datum
Badstr 1a	12345 Berlin	987	1.1.2012
		876	7.3.2011
Schloßstr 8	12456 Berlin	765	1.8.2011

Auswählen von Tupeln mit SQL

Hinweis

Wenn LE online, dann hier Beispiele zeigen

- Kunden (Name, Vorname) und Bestellungen (Datum, Preis)
- Kunden (Name, Vorname) mit und ohne Bestellung sowie die ggf. vorhandenen Bestellungen (Datum, Preis)
- Bestellungen (Datum, Preis) mit und ohne Kunden, sowie die ggf. vorhandenen Kunden mit (Name, Vorname)
- Warum liefert die letzte Abfrage das gleiche Ergebnis wie die erste?

Anschließend weiter mit Übung Ü6.1 Teil2

Auswählen von Tupeln mit SQL

Mengenoperationen

- Vereinigung
- Differenz
- Schnitt

Kartesisches Produkt

Auswählen von Tupeln mit SQL

Mengenoperationen

- Vereinigung (UNION)
 - Mengen A und B werden in einer Menge vereinigt
 - Voraussetzung: Gleiche Spalten (und Datentypen)
- Differenz
- Schnitt

Mitarbeiter	Name	Vorname	Kunden	Name	Vorname
	Wurst	Hans		Albers	Willi
	Wurst	Ida		Boehrs	Vera
	Vetter	Jan		Dinkel	Ulrike

```
SELECT * FROM mitarbeiter
UNION
SELECT * FROM kunden;
```

Name	Vorname
Albers	Willi
Boehrs	Vera
Dinkel	Ulrike
Wurst	Hans
Wurst	Ida
Vetter	Jan

Auswählen von Tupeln mit SQL

Mengenoperationen

- Vereinigung (UNION)
- Differenz (EXCEPT)
 - Eliminieren der Tupel in A, die auch in B sind
 - Voraussetzung: Gleiche Spalten (und Datentypen)
- Schnitt

Kartesisches Produkt

Personen	Name	Vorname
	Albers	Willi
	Boehrs	Vera
	Dinkel	Ulrike
	Wurst	Hans
	Wurst	Ida
	Vetter	Jan

Kunden	Name	Vorname
	Albers	Willi
	Boehrs	Vera
	Dinkel	Ulrike

```
SELECT * FROM personen
EXCEPT
SELECT * FROM kunden;
```

Name	Vorname
Wurst	Hans
Wurst	Ida
Vetter	Jan

Auswählen von Tupeln mit SQL

Mengenoperationen

- Vereinigung (UNION)
- Differenz (EXCEPT)
- Schnitt (INTERSECT)
 - Gemeinsame Tupel in A und B
 - Voraussetzung: Gleiche Spalten (und Datentypen)

Kunden	Name	Vorname	Mitarbeiter	Name	Vorname
	Wurst	Ida		Albers	Willi
	Vetter	Jan		Boehrs	Vera
	Dinkel	Ulrike		Dinkel	Ulrike
	Wurst	Hans		Wurst	Hans

```
SELECT * FROM kunden
INTERSECT
SELECT * FROM mitarbeiter;
```

Name	Vorname
Dinkel	Ulrike
Wurst	Hans

Auswählen von Tupeln mit SQL

Mengenoperationen

- Vereinigung (UNION)
- Differenz (EXCEPT)
- Schnitt (INTERSECT)

Kartesisches Produkt

- Kombination jedes Elements von Menge A mit jedem Element von B

```
SELECT *
FROM R1, R2;
```

R1	S1	S2
	A	B
	B	C
	C	D
	D	E



R2	S3	S4
	1	2
	2	3
	3	4

S1	S2	S3	S4
A	B	1	2
A	B	2	3
A	B	3	4
B	C	1	2
B	C	2	3
B	C	3	4
C	D	1	2
C	D	2	3
C	D	3	4
D	E	1	2
D	E	2	3
D	E	3	4

Auswählen von Tupeln mit SQL

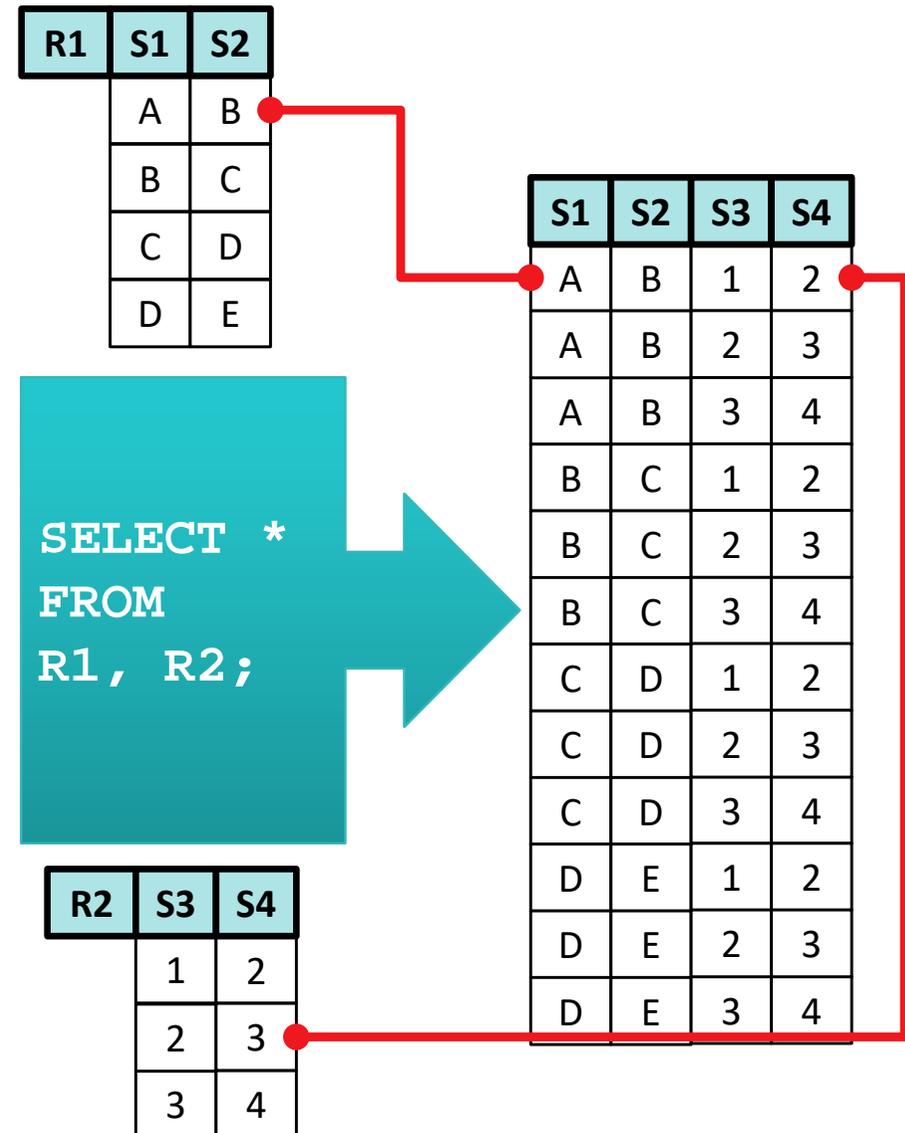
Mengenoperationen

- Vereinigung (UNION)
- Differenz (EXCEPT)
- Schnitt (INTERSECT)

Kartesisches Produkt

- Kombination jedes Elements von Menge A mit jedem Element von B

```
SELECT *
FROM R1, R2;
```



Auswählen von Tupeln mit SQL

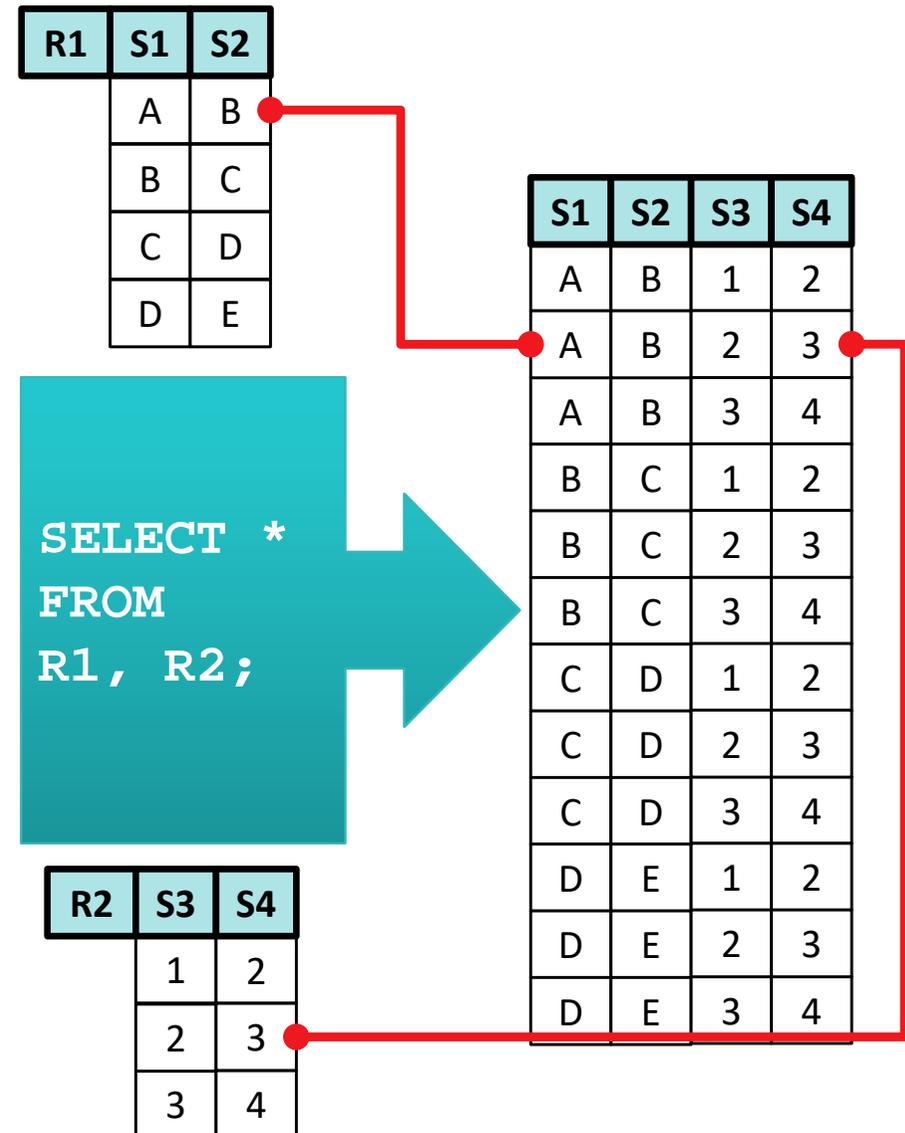
Mengenoperationen

- Vereinigung (UNION)
- Differenz (EXCEPT)
- Schnitt (INTERSECT)

Kartesisches Produkt

- Kombination jedes Elements von Menge A mit jedem Element von B

```
SELECT *
FROM R1, R2;
```



Auswählen von Tupeln mit SQL

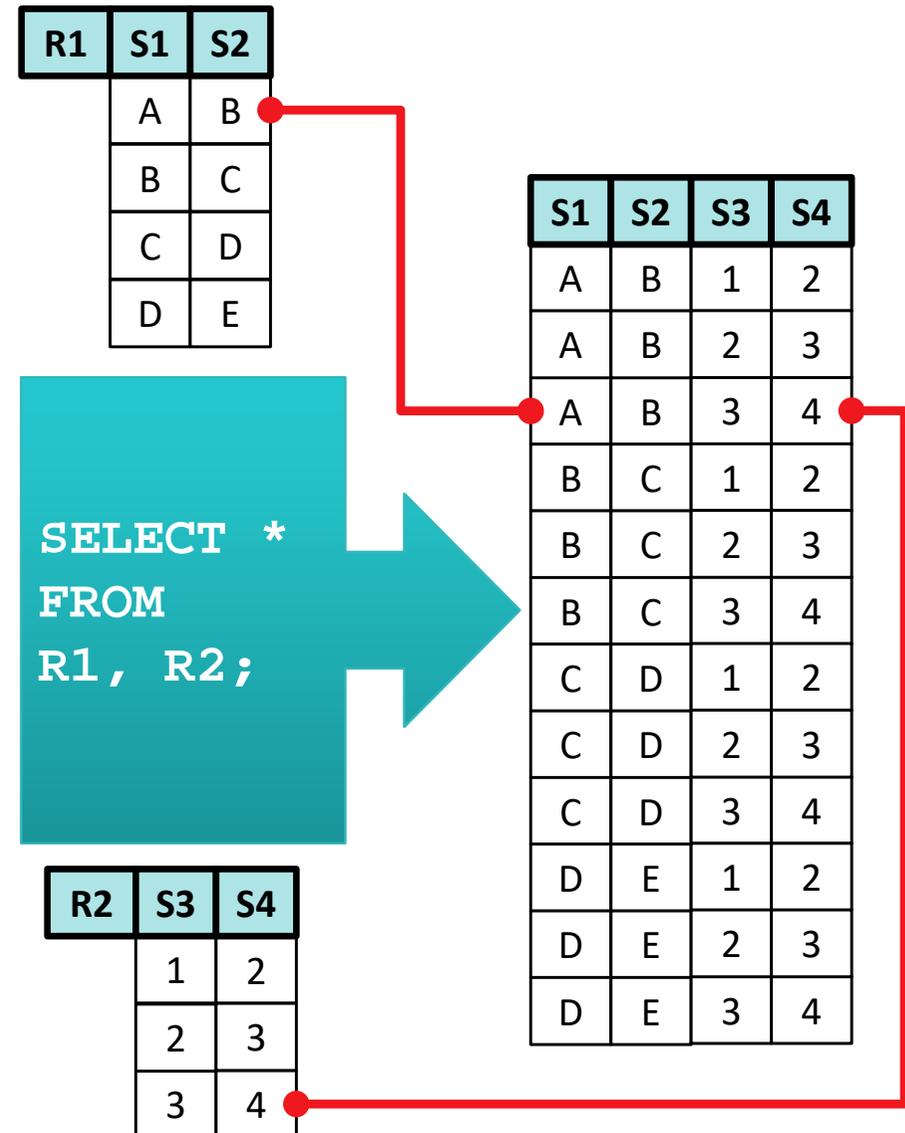
Mengenoperationen

- Vereinigung (UNION)
- Differenz (EXCEPT)
- Schnitt (INTERSECT)

Kartesisches Produkt

- Kombination jedes Elements von Menge A mit jedem Element von B

```
SELECT *
FROM R1, R2;
```



Auswählen von Tupeln mit SQL

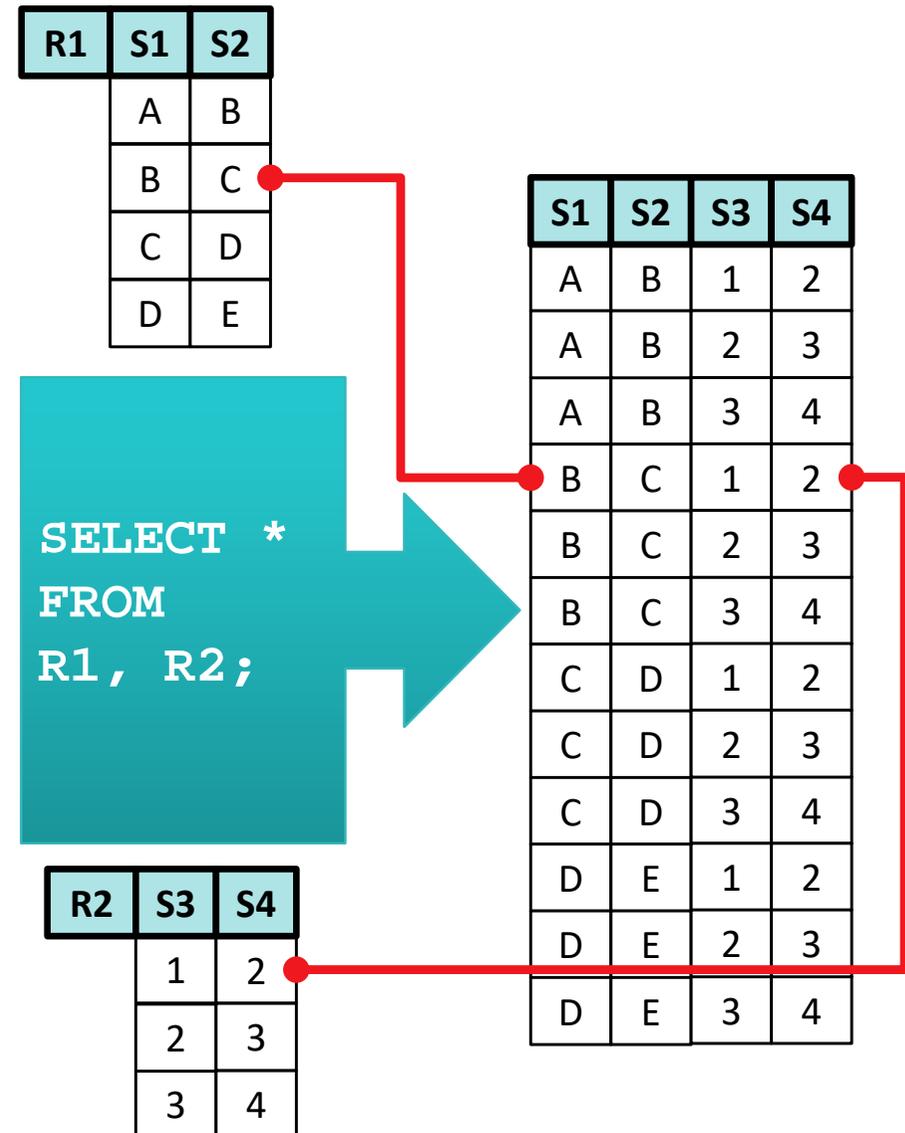
Mengenoperationen

- Vereinigung (UNION)
- Differenz (EXCEPT)
- Schnitt (INTERSECT)

Kartesisches Produkt

- Kombination jedes Elements von Menge A mit jedem Element von B

```
SELECT *
FROM R1, R2;
```



Auswählen von Tupeln mit SQL

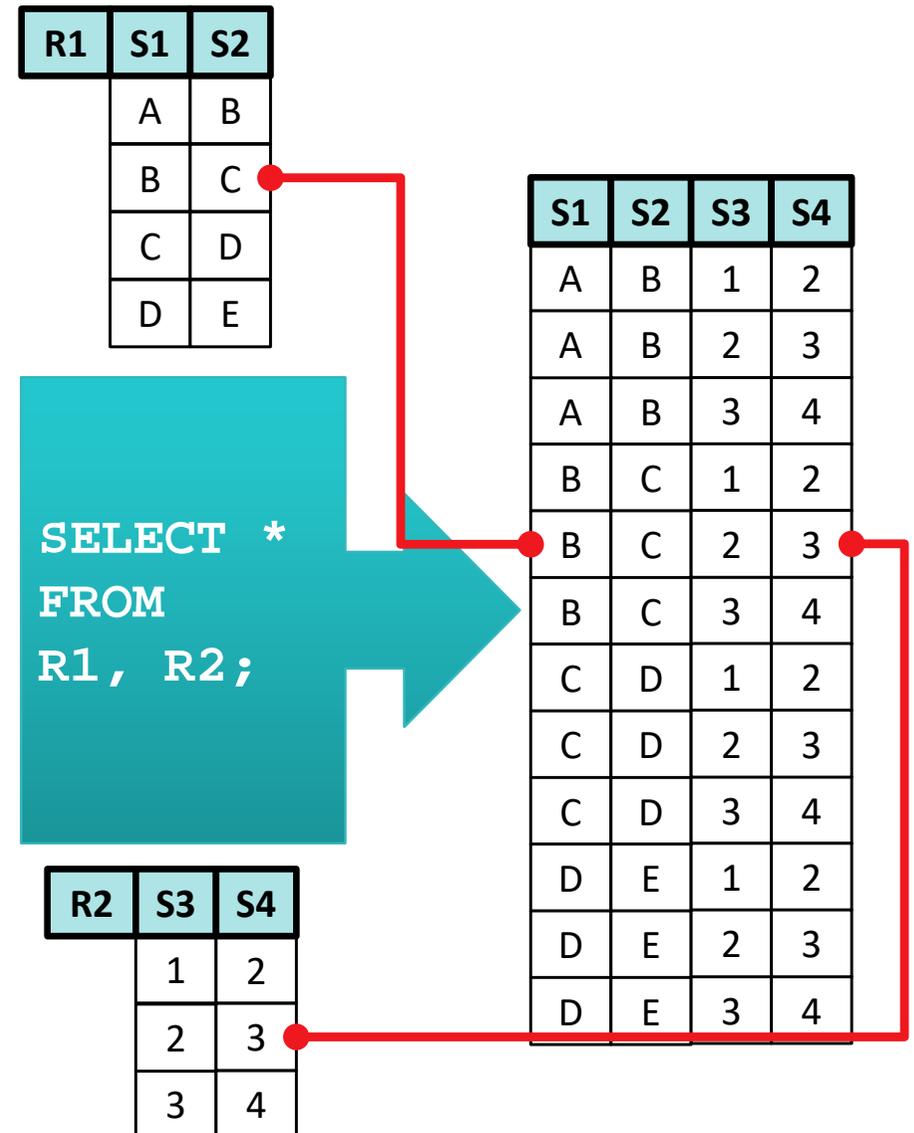
Mengenoperationen

- Vereinigung (UNION)
- Differenz (EXCEPT)
- Schnitt (INTERSECT)

Kartesisches Produkt

- Kombination jedes Elements von Menge A mit jedem Element von B

```
SELECT *
FROM R1, R2;
```



Auswählen von Tupeln mit SQL

Mengenoperationen

- Vereinigung (UNION)
- Differenz (EXCEPT)
- Schnitt (INTERSECT)

Kartesisches Produkt

- Kombination jedes Elements von Menge A mit jedem Element von B

```
SELECT *
FROM R1, R2;
```

R1	S1	S2
	A	B
	B	C
	C	D
	D	E

```
SELECT *
FROM
R1, R2;
```

R2	S3	S4
	1	2
	2	3
	3	4

S1	S2	S3	S4
A	B	1	2
A	B	2	3
A	B	3	4
B	C	1	2
B	C	2	3
B	C	3	4
C	D	1	2
C	D	2	3
C	D	3	4
D	E	1	2
D	E	2	3
D	E	3	4

Auswählen von Tupeln mit SQL

Mengenoperationen

- Vereinigung (UNION)
- Differenz (EXCEPT)
- Schnitt (INTERSECT)

Kartesisches Produkt

- Kombination jedes Elements von Menge A mit jedem Element von B

```
SELECT *  
FROM R1, R2;
```

R1	S1	S2
	A	B
	B	C
	C	D
	D	



R2	S3	S4
	1	2
	2	3
	3	4

S1	S2	S3	S4
A	B	1	2
A	B	2	3
A	B	3	4
B	C	1	2
B	C	2	3
B	C	3	4
C	D	1	2
C	D	2	3
C	D	3	4
D	E	1	2
D	E	2	3
D	E	3	4

Auswählen von Tupeln mit SQL

Mengenoperationen

- Vereinigung (UNION)
- Differenz (EXCEPT)
- Schnitt (INTERSECT)

Kartesisches Produkt

- Kombination jedes Elements von Menge A mit jedem Element von B

```
SELECT *
FROM R1, R2;
```

R1	S1	S2
	A	B
	B	C
	C	D
	D	E

```
SELECT *
FROM
R1, R2;
```

R2	S3	S4
	1	2
	2	3
	3	4

S1	S2	S3	S4
A	B	1	2
A	B	2	3
A	B	3	4
B	C	1	2
B	C	2	3
B	C	3	4
C	D	1	2
C	D	2	3
C	D	3	4
D	E	1	2
D	E	2	3
D	E	3	4

Auswählen von Tupeln mit SQL

Mengenoperationen

- Vereinigung
- Differenz
- Schnitt

Kartesisches Produkt

SELECT ... WHERE <Bedingung>

- Bedingungen mit NOT und verknüpft mit AND und OR möglich
- Bedingungen können Prüfung umfassen
 - gleich =
 - ungleich <> bzw. !=
 - größer als > bzw. kleiner als <
 - größer gleich >= bzw. kleiner gleich <=
 - nicht größer als !> bzw. nicht kleiner als !<
- Beispiele
 - **SELECT name FROM kunden
WHERE plz >= 12000 AND plz <= 15000
AND name='Thomas' ;**

SELECT ... WHERE <Bedingung>, weitere Operatoren für die Bedingung

– IN-Operator

- `SELECT * FROM kunden
WHERE name IN ('Wurst', 'Vetter', 'Dinkel')`

– BETWEEN-Operator

- `SELECT * FROM bestellungen
WHERE gesamtpreis BETWEEN 500 AND 1000;`

– LIKE-Operator

- `SELECT * FROM kunden
WHERE name LIKE 'Dink';`

– NULL-Operator

- `SELECT * FROM kunden
WHERE email IS NOT NULL;`

Alias für Tabellen verwenden

```
SELECT a.*, u.* FROM artikel AS a, umsatz AS u
```

Sortieren der Ergebnismenge (absteigend, engl. descending)

```
SELECT b.gesamt, b.datum FROM bestellungen as b  
ORDER BY b.gesamt DESC;
```

Berechnungen in Abfragen und Sortierung (aufsteigend, engl. ascending)

```
SELECT einzelpreis, stueckzahl, einzelpreis *  
stueckzahl AS gesamt  
FROM bestellpositionen  
ORDER BY gesamt ASC;
```

Entfernen doppelter Einträge (z.B. gleiche Names-Kombinationen bei Kunden)

```
SELECT DISTINCT name, vorname FROM kunden;
```

Überblick über die SELECT-Anweisung¹⁾

```
SELECT [DISTINCT]
    <Spaltenname>
    <Berechnung>
    <Konstante>
    <einer der obigen Ausdrücke> As Alias
    [, weitere der obigen Ausdrücke]
FROM <Ausdruck, der Tabelle zurückgibt> As Alias
[WHERE ...]
[GROUP BY ...]
[HAVING ...]
[UNION [ALL]]
[Weitere SELECT-Anweisung, welche dieselbe Zahl
von
    Spalten und Datentypen liefert]
[ORDER BY [Order-By-Ausdruck] ASC | DESC]]
[, weitere Sortierungen]
```

¹⁾ nach [1]

Auswählen

- vorhandene Tupel werden gelesen
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel bereitgestellt werden

Einfügen

- neue Tupel werden zur Relation hinzugefügt

Ändern

- vorhandene Tupel werden geändert
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel zu ändern sind

Löschen

- vorhandene Tupel werden gelöscht
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel zu ändern sind

Auswählen

- vorhandene Tupel werden gelesen
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel bereitgestellt werden

Einfügen

- neue Tupel werden zur Relation hinzugefügt

Ändern

- vorhandene Tupel werden geändert
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel zu ändern sind

Löschen

- vorhandene Tupel werden gelöscht
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel zu ändern sind

Einfügen von Tupeln mit SQL

**SQL-Syntax ähnlich der natürlichen Sprache (in Englisch),
d.h. es kann bspw. formuliert werden**

- "Füge in Tabelle Kunden die Werte 345, 'Simon', 'Jakob' ein."
- "Füge in Tabelle Kunden für die Spalte Vorname, Name, Kundennummer die Werte 'Simon', 'Jakob', 345 ein."

Einfügen von Tupeln mit SQL

Einfügen in vorgegebener Spaltenreihenfolge (Bsp. 1):

- Deutsch: "Füge in Tabelle Kunden die Werte 345, 'Simon', 'Jakob' ein."
- Englisch: "Insert into table *Kunden* values 345, 'Simon', 'Jakob'."
- SQL: **INSERT INTO kunden
VALUES (345, 'Simon', 'Jakob');**

Ausgangsrelation

Kunden	<u>KndNr</u>	Name	Vorname
	123	Albers	Willi
	234	Boehrs	Vera

```
INSERT INTO kunden  
VALUES (345, 'Simon', 'Jakob');
```

Ergebnisrelation

Kunden	<u>KndNr</u>	Name	Vorname
	123	Albers	Willi
	234	Boehrs	Vera
	345	Simon	Jakob

Einfügen von Tupeln mit SQL

Einfügen in abweichender Spaltenreihenfolge (Bsp. 1):

- Deutsch: "Füge in Tabelle Kunden für die Spalte Vorname, Name, Kundennummer die Werte 'Simon', 'Jakob', 345."
- Englisch: "Insert into table *Kunden* in columns Vorname, Name, Kundennummer values 'Simon', 'Jakob', 345."
- SQL:

```
INSERT INTO kunden(Vorname, Name, KndNr)
VALUES ('Simon', 'Jakob', 345);
```

Ausgangsrelation

Kunden	<u>KndNr</u>	Name	Vorname
	123	Albers	Willi
	234	Boehrs	Vera

```
INSERT INTO kunden(Vorname, Name, KndNr)
VALUES ('Simon', 'Jakob', 345);
```

Ergebnisrelation

Kunden	<u>KndNr</u>	Name	Vorname
	123	Albers	Willi
	234	Boehrs	Vera
	345	Jakob	Simon

Weiterführendes SQL zum Einfügen von Tuplen ~~BHT~~

Wenn Primärschlüssel auf Relation (als AutoWert) definiert, muss beim INSERT kein Wert für den Schlüssel angegeben werden

```
INSERT INTO personen (vorname, name)
VALUES ('Barbara', 'Beispiel');
```

Kombination von INSERT INTO mit dem Ergebnis einer Auswahl mittels SELECT

```
INSERT INTO personen (vorname, name)
SELECT vorname, name FROM kunden
WHERE ort='Berlin';
```

Einfügen von Tupeln mit SQL

Hinweis

Wenn LE online, dann hier Beispiele zeigen

- Einfügen einer neuen Bestellung für einen vorhandenen Kunden
- Einfügen einer neuen Bestellung für einen neuen Kunden

Anschließend weiter mit Übung Ü6.1 (Teil 3 inkl. Fortsetzung)

Überblick über die INSERT-Anweisung¹⁾

Entweder

```
INSERT INTO
    <Tabellenname>( <Spaltenname> [, weitere Spalte] )
VALUES
    ( <Wert für die erste Spalte> [, weitere Werte] )
```

Oder

```
INSERT INTO
    <Tabellenname>( <Spaltenname> [, weitere Spalte] )
SELECT <erste Spalte> [, weitere Ausgaben]
    [FROM ...]
    [WHERE ...]
    [GROUP BY ...]
    [HAVING ...]
```

¹⁾ siehe [1]

Auswählen

- vorhandene Tupel werden gelesen
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel bereitgestellt werden

Einfügen

- neue Tupel werden zur Relation hinzugefügt

Ändern

- vorhandene Tupel werden geändert
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel zu ändern sind

Löschen

- vorhandene Tupel werden gelöscht
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel zu ändern sind

Auswählen

- vorhandene Tupel werden gelesen
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel bereitgestellt werden

Einfügen

- neue Tupel werden zur Relation hinzugefügt

Ändern

- vorhandene Tupel werden geändert
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel zu ändern sind

Löschen

- vorhandene Tupel werden gelöscht
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel zu ändern sind

Ändern von Tupeln mit SQL

**SQL-Syntax ähnlich der natürlichen Sprache (in Englisch),
d.h. es kann bspw. formuliert werden**

- "Ändere in der Tabelle Kunden den Wert der Spalte Name in 'Albers', wo die Kundennummer '234' ist."

Ändern von Tupeln mit SQL

Änderungen (Bsp. 1):

- Deutsch: "Ändere in der Tabelle *Kunden* den Wert der Spalte *Name* in 'Albers', wo die *Kundennummer* '234' ist."
- Englisch: "Update in table *Kunden* value of column *Name* to 'Albers' where *Kundennummer* is '234'."
- SQL: **UPDATE kunden SET name='Albers' WHERE kndnr=234;**

Ausgangsrelation

Kunden	<u>KndNr</u>	Name	Vorname
	123	Albers	Willi
	234	Boehrs	Vera
	345	Muster	Michael

```
UPDATE kunden SET name='Albers'  
WHERE kndnr=234;
```



Ergebnisrelation

Kunden	<u>KndNr</u>	Name	Vorname
	123	Albers	Willi
	234	Albers	Vera
	345	Muster	Michael

Überblick über die UPDATE-Anweisung¹⁾

UPDATE <Tabelle>

SET <Name einer Spalte> = <Ausdruck aus Spalten,
Konstanten, Funktionen>

[, weitere Spaltennamen = Ausdruck]

WHERE <Bedingung>

1) siehe [1]

Ändern von Tupeln mit SQL

Hinweis

Wenn LE online, dann hier Beispiele zeigen:

- Die Bestellung mit dem Primärschlüsselwert 6 ist geliefert worden
- Erhöhung des Preises alle Bestellungen, die geliefert werden um 5 EUR (für Lieferkosten)

Anschließend weiter mit Übung Ü6.1 (Teil 4 inkl. Fortsetzung)

Auswählen

- vorhandene Tupel werden gelesen
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel bereitgestellt werden

Einfügen

- neue Tupel werden zur Relation hinzugefügt

Ändern

- vorhandene Tupel werden geändert
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel zu ändern sind

Löschen

- vorhandene Tupel werden gelöscht
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel zu ändern sind

Auswählen

- vorhandene Tupel werden gelesen
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel bereitgestellt werden

Einfügen

- neue Tupel werden zur Relation hinzugefügt

Ändern

- vorhandene Tupel werden geändert
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel zu ändern sind

Löschen

- vorhandene Tupel werden gelöscht
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel zu ändern sind

Löschen von Tupeln mit SQL

**SQL-Syntax ähnlich der natürlichen Sprache (in Englisch),
d.h. es kann bspw. formuliert werden**

- "Lösche alles aus der Tabelle Kunden."
- "Lösche aus der Tabelle Kunden wo der Name 'Albers' ist."

Ändern von Tupeln mit SQL

Alle Zeilen löschen (Bsp. 1):

- Deutsch: "Lösche alles aus der Tabelle *Kunden*."
- Englisch: "Delete all from table *Kunden*."
- SQL: **DELETE FROM kunden;**

Ausgangsrelation

Kunden	<u>KndNr</u>	Name	Vorname
	123	Albers	Willi
	234	Boehrs	Vera
	345	Muster	Michael

DELETE FROM kunden;



Ergebnisrelation

Kunden	<u>KndNr</u>	Name	Vorname
--------	--------------	------	---------

Ändern von Tupeln mit SQL

Alle Zeilen, die einer Bedingung genügen löschen (Bsp. 2):

- Deutsch: "Lösche aus der Tabelle *Kunden* wo der *Name* 'Albers' ist."
- Englisch: "Delete from table *Kunden* where *Name* is 'Albers'."
- SQL:
DELETE FROM kunden WHERE name='Albers' ;

Ausgangsrelation

Kunden	<u>KndNr</u>	Name	Vorname
	123	Albers	Willi
	234	Albers	Vera
	345	Muster	Michael

```
DELETE FROM kunden  
WHERE name='Albers' ;
```

Ergebnisrelation

Kunden	<u>KndNr</u>	Name	Vorname
	345	Muster	Michael

Überblick über die DELETE-Anweisung¹⁾

```
DELETE FROM <Tabelle>  
[WHERE <Bedingung>];
```

1) siehe [1]

Ändern von Tupeln mit SQL

Hinweis

Wenn LE online, dann hier Beispiele

- Löschen einer Bestellung
- Löschen eines Kunden
- Löschen aller Bestellungen eines Kunden

Anschließend weiter mit Übung Ü6.1 Teil 5 inkl. Fortsetzung

Dann weiter bei Folie Nr. 100 und dann weiter bei Nr. 121

Auswählen

- vorhandene Tupel werden gelesen
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel bereitgestellt werden

Einfügen

- neue Tupel werden zur Relation hinzugefügt

Ändern

- vorhandene Tupel werden geändert
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel zu ändern sind

Löschen

- vorhandene Tupel werden gelöscht
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel zu ändern sind

Auswählen

- vorhandene Tupel werden gelesen
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel bereitgestellt werden

Einfügen

- neue Tupel werden zur Relation hinzugefügt

Ändern

- vorhandene Tupel werden geändert
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel zu ändern sind

Löschen

- vorhandene Tupel werden gelöscht
- ggf. nach bestimmten Kriterien festgelegt, welche Tupel zu ändern sind

Inhalt

Ziel und Einordnung

Rückblick

- Beziehungen im Relationen Modell
- Fremdschlüssel
- Integritätsregeln
- Operationen auf Relationen und Tupeln

Grundlagen des Relationalen Datenmodell (Teil 3) - SQL

- Bestandteile
- Sprachumfang zum Auswählen, Einfügen, Ändern und Löschen
- Zusammenfassung

Arbeiten mit dem Relationalen Modell (Teil 3) - SQL in MS Access

- Formulare und SQL-Auswahlabfragen
- Vordefinierte VBA-Funktionen zum Einfügen, Ändern und Löschen mit SQL
- Zusammenhang und Abgrenzung grafischer Abfragen von SQL-Abfragen
- Zusammenfassung

Ausblick

Zusammenfassung

SQL (Structured Query Language) als Datenbanksprache

Bestandteile des SQL-Sprachumfangs

- Data Manipulation Language (DML): dient zur Abfrage, zum Hinzufügen, zur Veränderung und zum Löschen von Daten
 - SELECT
 - UPDATE
 - INSERT
 - DELETE
- Data Definition Language (DDL): dient zum Erzeugen, Verändern und Löschen der Strukturen, die für die Speicherung der Daten benutzt werden (z.B. Tabellen, Spalten)
- Data Control Language (DCL): dient zum Einrichten, Festlegen und Entziehen von Zugriffsrechten für Benutzer und Gruppen auf den Strukturen und auf Aktionsmöglichkeiten der Datenbank

Thema von LE 11
Thema von LE 11

Inhalt

Ziel und Einordnung

Rückblick

- Beziehungen im Relationen Modell
- Fremdschlüssel
- Integritätsregeln
- Operationen auf Relationen und Tupeln

Grundlagen des Relationalen Datenmodell (Teil 3) - SQL

- Bestandteile
- Sprachumfang zum Auswählen, Einfügen, Ändern und Löschen
- Zusammenfassung

Arbeiten mit dem Relationalen Modell (Teil 3) - SQL in MS Access

- Formulare und SQL-Auswahlabfragen
- Vordefinierte VBA-Funktionen zum Einfügen, Ändern und Löschen mit SQL
- Zusammenhang und Abgrenzung grafischer Abfragen von SQL-Abfragen
- Zusammenfassung

Ausblick

SQL in MS Access

SQL in der SQL-Ansicht von grafischen "Abfragen" zum Auswählen, Einfügen, Ändern und Löschen

SQL SELECT-Abfragen als Datenquelle für Formulare

Vordefinierte VBA-Funktionen zum Einfügen, Ändern und Löschen mit SQL

SQL in MS Access

SQL in der SQL-Ansicht von grafischen "Abfragen" zum Auswählen, Einfügen, Ändern und Löschen

- SELECT
- INSERT
- UPDATE
- DELETE

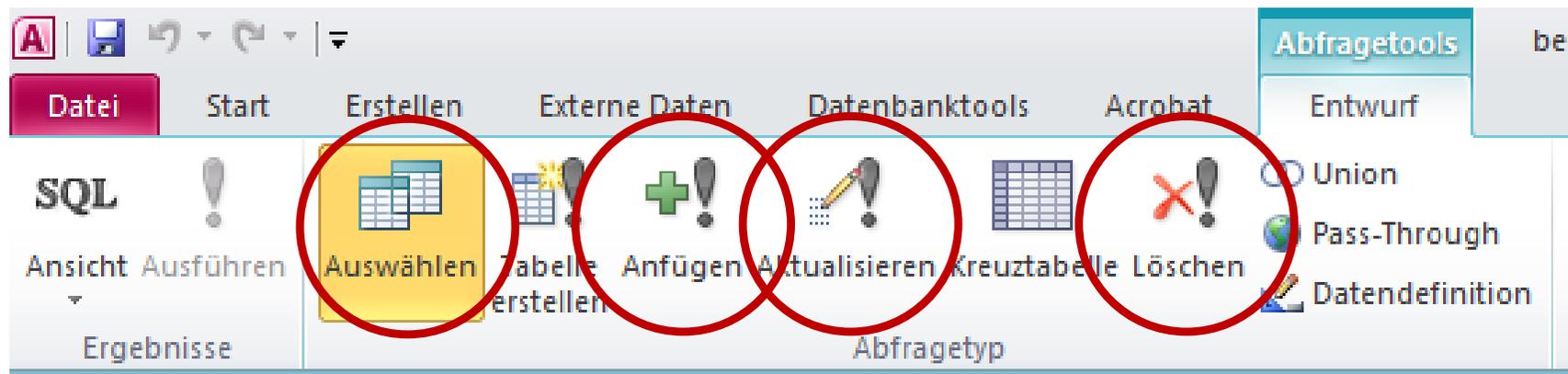
SQL SELECT-Abfragen als Datenquelle für Formulare

Vordefinierte VBA-Funktionen zum Einfügen, Ändern und Löschen mit SQL

Grafische Abfragen und SQL-Abfragen

MS Access stellt dem Sprachumfang von SQL vergleichbare grafische Abfragen bereit

- Auswahlabfrage für Restriktion, Projektion und Join
 - entspricht: SELECT
 - vgl. letzte LE
- Anfügeabfrage: entspricht INSERT
- Aktualisierungsabfrage: entspricht UPDATE
- Löschartabfrage: entspricht DELETE



SQL-Ansicht grafischer Abfragen

Jede Abfrage kann in mehreren Ansichten dargestellt werden

– Entwurfsansicht

The screenshot shows a database query designer interface. At the top, a tab labeled 'qryProdukteGarten' is visible. Below it, a table 'tblProdukte' is shown with the following fields: prdIdPk (marked with a key icon), prdBezeichnung, prdPreis, and prdkatIdFk. Below the table, a query grid is displayed with the following columns: 'Feld:', 'Tabelle:', 'Sortierung:', 'Anzeigen:', 'Kriterien:', and 'oder:'. The grid contains the following data:

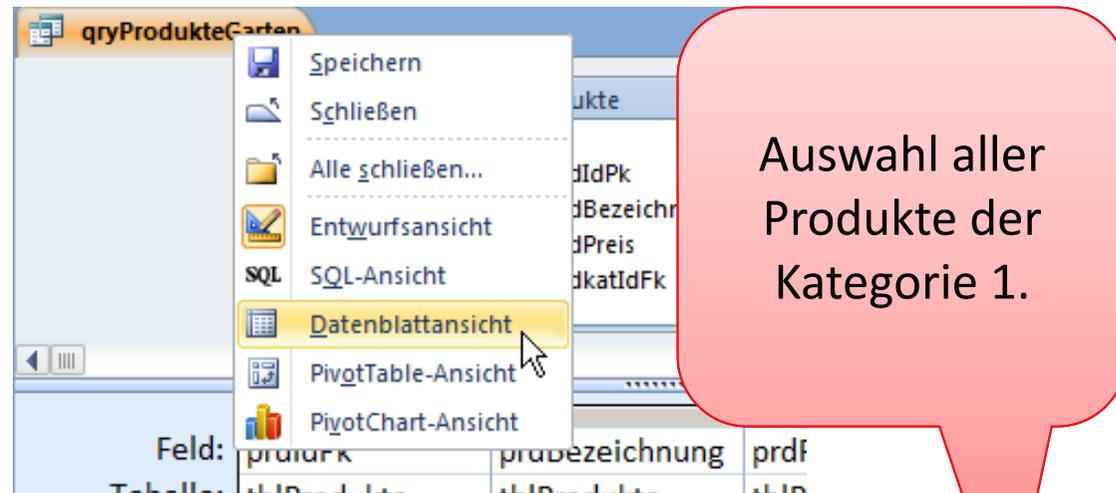
Feld:	prdIdPk	prdBezeichnung	prdPreis	prdkatIdFk
Tabelle:	tblProdukte	tblProdukte	tblProdukte	tblProdukte
Sortierung:				
Anzeigen:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kriterien:				1
oder:				

A red callout bubble points to the 'tblProdukte' table in the query grid, containing the text: 'Auswahl aller Produkte der Kategorie 1.'

SQL-Ansicht grafischer Abfragen

Jede Abfrage kann in mehreren Ansicht dargestellt werden

- Entwurfsansicht
- Datenblattansicht

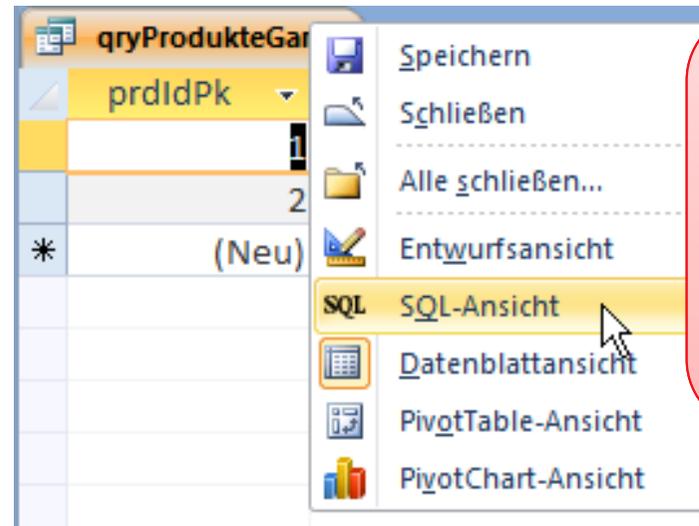


qryProdukteGarten				
	prdlIdPk	prdbezeichnung	prdpPreis	prdkatIdFk
	1	Harke "Boris"	5,99 €	1
	2	Rasenbesen "Thomas"	6,99 €	1
*	(Neu)			

SQL-Ansicht grafischer Abfragen

Jede Abfrage kann in mehreren Ansicht dargestellt werden

- Entwurfsansicht
- Datenblattansicht
- SQL-Ansicht



Auswahl aller
Produkte der
Kategorie 1.

qryProdukteGarten

```
SELECT tblProdukte.prdIdPk, tblProdukte.prdBezeichnung,  
tblProdukte.prdPreis, tblProdukte.prdkatIdFk  
FROM tblProdukte  
WHERE (((tblProdukte.prdkatIdFk)=1));
```

SQL-Ansicht grafischer Abfragen

SQL-Ansicht steht auch für Anfüge-, Aktualisierungs- und Löschabfragen zur Verfügung

Beispiel

- Aktualisierung der Kategorienbezeichnung "Haushalt" in "Haushalt/Balkon"

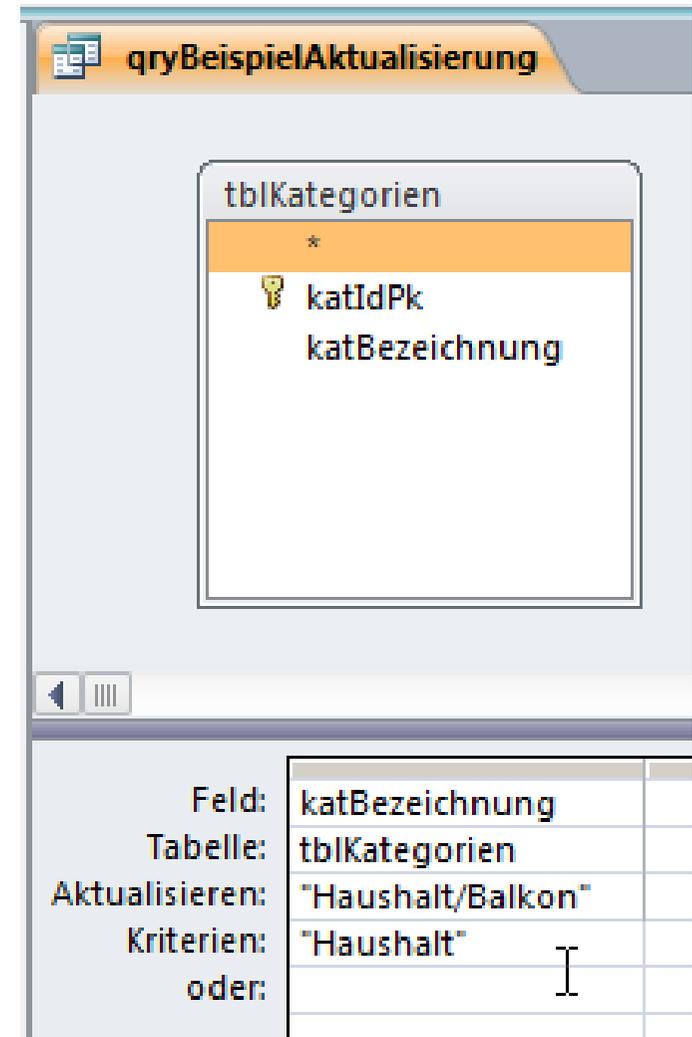
	katIdPk	katBezeichn	Zum Hinzu
+	1	Garten	
+	2	Haushalt	
*	(Neu)		

SQL-Ansicht grafischer Abfragen

SQL-Ansicht steht auch für Anfüge-, Aktualisierungs- und Löschartfragen zur Verfügung

Beispiel

- Aktualisierung der Kategorienbezeichnung "Haushalt" in "Haushalt/Balkon"



SQL-Ansicht grafischer Abfragen

SQL-Ansicht steht auch für Anfüge-, Aktualisierungs- und Löschanfragen zur Verfügung

Beispiel

- Aktualisierung der Kategorienbezeichnung "Haushalt" in

The screenshot shows a query editor window titled 'qryBeispielAktualisierung'. The SQL query is: `UPDATE tblKategorien SET tblKategorien.katBezeichnung = "Haushalt/Balkon" WHERE (((tblKategorien.katBezeichnung)="Haushalt"));`. Red annotations are as follows: 1 points to the table name 'tblKategorien' in the SET clause; 2 points to the field name 'katBezeichnung' in both the SET and WHERE clauses; 3 points to the new value 'Haushalt/Balkon' in the SET clause; 4 points to the old value 'Haushalt' in the WHERE clause. A graphical view of the query is shown below, with a table 'tblKategorien' containing a row with an asterisk. A legend at the bottom right maps the numbers to the graphical elements: 1 (Table: tblKategorien), 2 (Field: katBezeichnung), 3 (Aktualisieren: "Haushalt/Balkon"), and 4 (Kriterien: "Haushalt").

SQL-Ansicht grafischer Abfragen

SQL-Ansicht steht auch für Anfüge-, Aktualisierungs- und Löschartfragen zur Verfügung

Beispiel

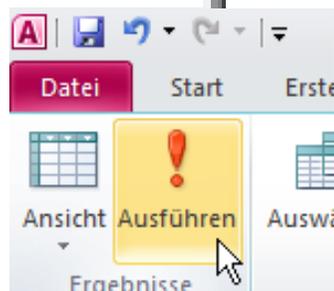
- Aktualisierung der Kategorienbezeichnung "Haushalt" in "Haushalt/Balkon"

The screenshot shows a database management tool interface. At the top, a tab labeled 'qryBeispielAktualisierung' is active. Below it, a table structure for 'tblKategorien' is displayed, showing a primary key 'katIdPk' and a field 'katBezeichnung'. Below the table structure, a query editor window shows the following SQL statement:

```
UPDATE tblKategorien SET tblKategorien.katBezeichnung = "Haushalt/Balkon"  
WHERE (((tblKategorien.katBezeichnung)="Haushalt"));
```

Below the query editor, a table structure is shown with the following fields:

Feld:	katBezeichnung
Tabelle:	tblKategorien
Aktualisieren:	"Haushalt/Balkon"
Kriterien:	"Haushalt"
oder:	



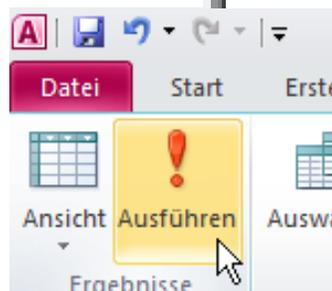
SQL-Ansicht grafischer Abfragen

SQL-Ansicht steht auch für Anfüge-, Aktualisierungs- und Löschartfragen zur Verfügung

Beispiel

- Aktualisierung der Kategorienbezeichnung "Haushalt" in "Haushalt/Balkon"

The screenshot shows a database management tool interface. At the top, a window titled 'qryBeispielAktualisierung' displays the structure of the 'tblKategorien' table, which has a primary key 'katIdPk' and a field 'katBezeichnung'. Below this, a larger window of the same title shows the SQL query: `UPDATE tblKategorien SET tblKategorien.katBezeichnung = "Haushalt/Balkon" WHERE (((tblKategorien.katBezeichnung)="Haushalt"));`. A cursor is positioned at the end of the query. To the right, a table shows the execution parameters: 'Feld: katBezeichnung', 'Tabelle: tblKategorien', 'Aktualisieren: "Haushalt/Balkon"', and 'Kriterien: "Haushalt"'. A cursor is also visible in the 'Kriterien' field.

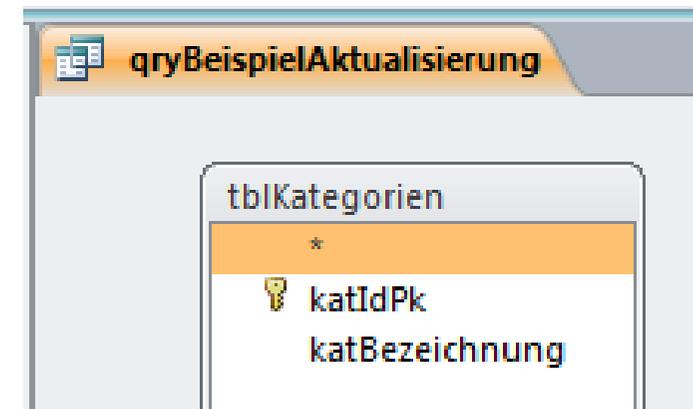


SQL-Ansicht grafischer Abfragen

SQL-Ansicht steht auch für Anfüge-, Aktualisierungs- und Löschartfragen zur Verfügung

Beispiel

- Aktualisierung der Kategorienbezeichnung "Haushalt" in "Haushalt/Balkon"



```
UPDATE tbIKategorien SET tbIKategorien.katBezeichnung = "Haushalt/Balkon"  
WHERE (((tbIKategorien.katBezeichnung)="Haushalt");
```

	katIdPk	katBezeichnung	Zum
+	1	Garten	
+	2	Haushalt/Balkon	
*	(Neu)		

Field: `katBezeichnung`
Tabelle: `tbIKategorien`
Aktualisieren: `"Haushalt/Balkon"`
Kriterien: `"Haushalt"`
oder:

SQL-Ansicht grafischer Abfragen

Verbund (Join) in der Entwurfs- und SQL-Ansicht von Auswahlabfragen

The image shows a database design tool interface. On the left, two tables are displayed: **tblLieferAdressen** and **tblBestellungen**. **tblLieferAdressen** has fields: **adrIdPK** (primary key), **adrStrasseNr**, **adrPlzOrt**, and **adrkndIdFk**. **tblBestellungen** has fields: **bstIdPk** (primary key), **bstDatum**, **bstLieferung**, **bstkndIdFk**, **bstGesamt**, and **bstadrIdFk**. A relationship line connects the two tables, with a '1' at the **tblLieferAdressen** end and an '∞' at the **tblBestellungen** end. A red arrow points from the relationship line to the **Verknüpfungseigenschaften** dialog box.

The **Verknüpfungseigenschaften** dialog box shows the following configuration:

- Linker Tabellename: **tblLieferAdressen**
- Rechter Tabellename: **tblBestellungen**
- Linker Spaltenname: **adrIdPK**
- Rechter Spaltenname: **bstadrIdFk**
- Join Type: **1: Beinhaltet nur die Datensätze, bei denen die Inhalte der verknüpften Felder beider Tabellen gleich sind.** (Selected)
- Option 2: **2: Beinhaltet ALLE Datensätze aus 'tblLieferAdressen' und nur die Datensätze aus 'tblBestellungen', die die verknüpften Felder beider Tabellen gleich sind.**
- Option 3: **3: Beinhaltet ALLE Datensätze aus 'tblLieferAdressen' und nur die Datensätze aus 'tblBestellungen', die die verknüpften Felder beider Tabellen gleich sind.**

A red callout box labeled **Inner Join** points to the selected join type option.

Below the dialog box is a table showing the fields selected for the query:

Feld:	bstIdPk	bstDatum	adrStrasseNr	adrPlzOrt			
Sortierung:							
Anzeigen:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Kriterien:							
oder:							

SQL-Ansicht grafischer Abfragen

Verbund (Join) in der Entwurfs- und SQL-Ansicht von Auswahlabfragen

The image shows a database design tool interface. On the left, two tables are displayed: **tblLieferAdressen** (primary key: adrIdPK) and **tblBestellungen** (primary key: bstIdPk). A relationship line connects them, with a '1' on the left and an 'n' on the right, and an arrow pointing to the 'n' side. A red arrow points from this relationship to the 'Verknüpfungseigenschaften' dialog box.

The dialog box 'Verknüpfungseigenschaften' has the following settings:

- Linker Tabellename: **tblLieferAdressen**
- Rechter Tabellename: **tblBestellungen**
- Linker Spaltenname: **adrIdPK**
- Rechter Spaltenname: **bstadrIdFk**
- Join Type: 2: Beinhaltet ALLE Datensätze aus 'tblLieferAdressen' und nur die Datensätze aus 'tblBestellungen', bei denen die Inhalte der verknüpften Felder beider Tabellen gleich sind.

A red callout box labeled 'Left Outer Join' points to the selected join type option.

Feld:	bstIdPk	bstDatum	adrStrasseNr	adrPlzOrt		
Tabelle:	tblBestellungen	tblBestellungen	tblLieferAdresser	tblLieferAdresser		
Sortierung:						
Anzeigen:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Kriterien:						
oder:						

SQL-Ansicht grafischer Abfragen

Verbund (Join) in der Entwurfs- und SQL-Ansicht von Auswahlabfragen

The image shows a database design tool interface. On the left, two tables are displayed: **tblLieferAdressen** (primary key: adrIdPK) and **tblBestellungen** (primary key: bstIdPk). A relationship line connects them, with a '1' on the left and an '∞' on the right. A red arrow points from the '∞' side of the relationship to the 'Verknüpfungseigenschaften' dialog box.

The **Verknüpfungseigenschaften** dialog box shows the following configuration:

- Linker Tabellennamen: **tblLieferAdressen**
- Rechter Tabellennamen: **tblBestellungen**
- Linker Spaltenname: **adrIdPK**
- Rechter Spaltenname: **bstadrIdFk**
- Selected option: **3: Beinhaltet ALLE Datensätze aus 'tblBestellungen' und nur die Datensätze aus 'tblLieferAdressen', bei denen die Inhalte der verknüpften Felder beider Tabellen gleich sind.**

Below the dialog box is a query grid:

Feld:	bstIdPk	bstDatum	adrStrasseNr	adrPlzOrt		
Tabelle:	tblBestellungen	tblBestellungen	tblLieferAdresser	tblLieferAdre.		
Sortierung:						
Anzeigen:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Kriterien:						
oder:						

A red callout box with the text **Right Outer Join** points to the selected option in the dialog box.

SQL-Ansicht grafischer Abfragen

Verzicht auf grafische Abfragen, jetzt kommt SQL!

Generelle Vorgehensweise

1. Erstellen eines neuen Abfrageentwurfs
2. Schließen des Dialogs zum Hinzufügen von Tabellen
3. Sofortiges Wechseln in SQL-Ansicht
4. Erstellen der SQL-Anweisung bzw. SQL-Abfrage
5. Speichern als Abfrage (Präfix: qry)
6. Ausführen der Abfrage
 - Klick auf Symbol für Ausführen (bei SQL-Anweisungen)
 - Wechsel in Datenblattansicht (bei SQL-Abfragen)



D06.03

- Erstellen von drei SQL-Anweisungen
 1. zum Hinzufügen eines neuen Benutzers
 2. zum Ändern des Passworts des hinzugefügten Benutzers
 3. zum Löschen des geänderten Benutzers
- Vorgehensweise
 - Erstellen eines neuen Abfrageentwurfs
 - Schließen des Dialogs zum Hinzufügen von Tabellen
 - Sofortiges Wechseln in SQL-Ansicht
 - Speichern als Abfrage
 - Ausführen der Abfrage
- Warum kann die Anweisung Nr. 1 nicht mehrfach hintereinander ausgeführt werden?



SQL in MS Access

SQL in der SQL-Ansicht von grafischen "Abfragen" zum Auswählen, Einfügen, Ändern und Löschen

- SELECT
- INSERT
- UPDATE
- DELETE

SQL SELECT-Abfragen als Datenquelle für Formulare

Vordefinierte VBA-Funktionen zum Einfügen, Ändern und Löschen mit SQL

SQL in MS Access

SQL in der SQL-Ansicht von grafischen "Abfragen" zum Auswählen, Einfügen, Ändern und Löschen

SQL SELECT-Abfragen als Datenquelle für Formulare

Vordefinierte VBA-Funktionen zum Einfügen, Ändern und Löschen mit SQL

SQL in MS Access

SQL in der SQL-Ansicht von grafischen "Abfragen" zum Auswählen, Einfügen, Ändern und Löschen

SQL-Abfragen als Datenquelle für Formulare

– nur SELECT

Vordefinierte VBA-Funktionen zum Einfügen, Ändern und Löschen mit SQL

SQL-Abfragen als Datenquelle für Formulare

Schritt 1

- SQL-Abfrage überlegen

Schritt 2

- Formular erstellen
- SQL-Abfrage als Datensatzquelle eintragen

The screenshot displays the Microsoft Access interface. On the left, the 'frmProdukteGarten' form is visible, featuring a title bar and a main area with the text 'Produkte rund um den Garten'. The form is divided into sections: 'Formularkopf', 'Detailbereich', and 'Formularfuß'. The 'Detailbereich' contains three text boxes labeled 'prdIdPk', 'prdBezeichnung', and 'prdBeschre'. On the right, the 'Eigenschaftenblatt' (Properties Sheet) is open, showing the 'Auswahltyp: Formular' and 'Formular' dropdown. The 'Datensatzquelle' property is set to 'SELECT * FROM tblProdukte WHERE prdkatIdFk=2;', which is circled in red. Other properties include 'Recordsettyp' (Dynaset), 'Standardwerte abrufen' (Ja), 'Filter' (Beim Laden filtern: Nein, Sortiert nach: Beim Laden sortieren: Ja, Auf Nachverarbeitung warten: Nein), 'Daten eingeben' (Nein), 'Anfügen zulassen' (Ja), 'Löschen zulassen' (Ja), 'Bearbeitungen zulassen' (Ja), 'Filter zulassen' (Ja), and 'Datensätze sperren' (Keine Sperrungen).

Schritt 1

- SQL-Abfrage überlegen

Schritt 2

- Formular erstellen
- SQL-Abfrage als Datensatzquelle eintragen
oder
SQL-Abfrage per Programmierung als Datensatzquelle festlegen

```
' Datensatzquelle als Eigenschaft des aktuellen Formulars
```

```
' kann so verändert werden
```

```
Me.RecordSource = "SELECT * FROM tblProdukte " & _  
                  "WHERE prdkatIdFk=" & bytAktKategorie & ";"
```

```
' Aktualisierung des Formulars (Neuladen) erforderlich
```

```
Me.Requery
```

D06.01 (Teil 1)

- Vorhanden ist das Formular eines Produktkatalogs, dass
 - bisher alle Produkte zeigt und
 - die Datensatzquelle tblProdukte verwendet
- Es soll zukünftig nur die Produkte der Kategorie 4 (Lebensmittel) zeigen
- Wie muss die SQL-Abfrage lauten?
- Ändern der Datensatzquelle, um die SQL-Abfrage zu verwenden
- Prüfen des Formularinhalts (nur noch Lebensmittel)
- ...

D06.01 (Teil 2)

- ...
- Formular im Kopfbereich um zwei Schaltflächen "Alle Produkte" und "Getränke" erweitern
- zwei SQL-Abfragen überlegen
 - alle Produkte auswählen
 - nur Getränke (Produktkategorie 4) auswählen
- Ereignisprozedur für Klicken anlegen
 - Variable für SQL-Abfrage deklarieren (String)
 - Variable mit SQL-Abfrage initialisieren
 - Datensatzquelle des Formulars die SQL-Anweisung zuweisen
- Formular erproben: Umschalten zwischen allen Produkten und nur Getränken möglich

SQL in MS Access

SQL in der SQL-Ansicht von grafischen "Abfragen" zum Auswählen, Einfügen, Ändern und Löschen

- SELECT
- INSERT
- UPDATE
- DELETE

SQL SELECT-Abfragen als Datenquelle für Formulare

Vordefinierte VBA-Funktionen zum Einfügen, Ändern und Löschen mit SQL

SQL in MS Access

SQL-Abfragen als Datenquelle für Formulare

Vordefinierte VBA-Funktionen zum Einfügen, Ändern und Löschen mit SQL

Zusammenhang und Abgrenzung grafischer Abfragen von SQL-Abfragen

SQL in MS Access

SQL in der SQL-Ansicht von grafischen "Abfragen" zum Auswählen, Einfügen, Ändern und Löschen

SQL-Abfragen als Datenquelle für Formulare

Vordefinierte VBA-Funktionen zum Einfügen, Ändern und Löschen mit SQL

- INSERT
- UPDATE
- DELETE
- kein SELECT!

Vordefinierte VBA-Funktionen für SQL

Einfügen, Ändern und Löschen mit SQL durch Nutzung von

- ' Standardfunktionen aufrufen, als Parameter
- ' String mit SQL-Abfrage übergeben

```
Call CurrentDb.Execute("<SQL-Anweisung>")
```

```
Call DoCmd.RunSQL("<SQL-Anweisung>")
```

Um Auswirkungen in einem Formular sichtbar zu machen

- ' Aktualisierung des Formulars (Neuladen)
- ' erforderlich

```
Call Me.Requery
```

Beispiel

- Löschen des aktuellen Warenkorb-Eintrags über neue Schaltfläche
- ID des aktuellen Eintrags kann mit Me.<FeldMitPrimärschlüssel> abgefragt werden
- Wird in SQL-Anweisung "eingebaut" und Anweisung wird ausgeführt
- Neuladen des Formulars notwendig

WarenkorbID	Bezeichnung	Anzahl	Preis	Gesamtpreis	WkbZeitpunkt	knIdPk
6	Pflanzdünger	2	34,56 €	69,12 €	19.04.2012	1
7	Universaldünger	3	45,67 €	137,01 €	30.05.2012	1

```
Option Compare Database  
Option Explicit
```

```
Private Sub btnLoeschen_Click()  
    CurrentDb.Execute ("DELETE FROM tblWarenkoerbe WHERE wkbIdPk =" & Me.txtWkbIdPk)  
    Me.Requery  
End Sub
```

Hinweis: Nur zur einfachen Demonstration wird hier SQL direkt aus der Oberfläche ausgeführt. Eine saubere Trennung in Schichten ist empfehlenswert.

D06.02

- Schaltfläche zum Löschen von Produkten im Formular des Warenkorbs einfügen
- Bei Klicken soll der in der aktuellen Zeile des Formulars dargestellte Kunde gelöscht werden
 - SQL-Abfrage überlegen und
 - in Ereignisprozedur ausführen

SQL in MS Access

SQL in der SQL-Ansicht von grafischen "Abfragen" zum Auswählen, Einfügen, Ändern und Löschen

SQL-Abfragen als Datenquelle für Formulare

Vordefinierte VBA-Funktionen zum Einfügen, Ändern und Löschen mit SQL

- INSERT
- UPDATE
- DELETE
- kein SELECT!

SQL in MS Access

SQL-Abfragen als Datenquelle für Formulare

Vordefinierte VBA-Funktionen zum Einfügen, Ändern und Löschen mit SQL

Zusammenhang und Abgrenzung grafischer Abfragen von SQL-Abfragen

Inhalt

Ziel und Einordnung

Rückblick

- Beziehungen im Relationen Modell
- Fremdschlüssel
- Integritätsregeln
- Operationen auf Relationen und Tupeln

Grundlagen des Relationalen Datenmodell (Teil 3) - SQL

- Bestandteile
- Sprachumfang zum Auswählen, Einfügen, Ändern und Löschen
- Zusammenfassung

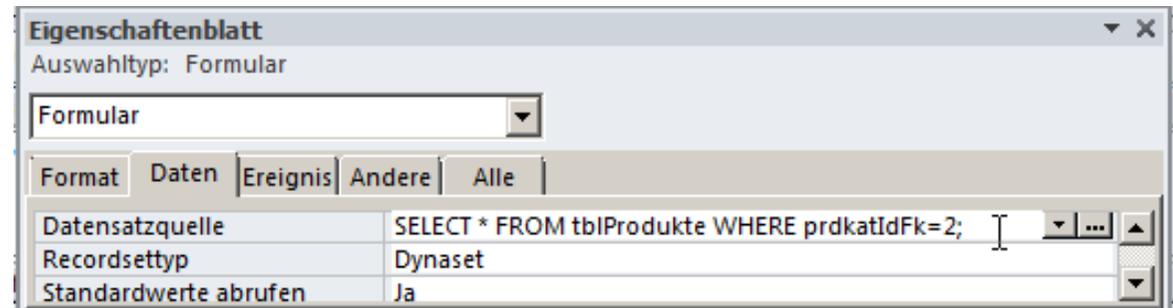
Arbeiten mit dem Relationalen Modell (Teil 3) - SQL in MS Access

- Formulare und SQL-Auswahlabfragen
- Vordefinierte VBA-Funktionen zum Einfügen, Ändern und Löschen mit SQL
- Zusammenhang und Abgrenzung grafischer Abfragen von SQL-Abfragen
- Zusammenfassung

Ausblick

Zusammenfassung

SQL-Abfragen können als Datenquelle für Formulare verwendet werden



Einfügen, Ändern und Löschen mit SQL mit vordefinierter VBA-Funktionen

' Generelle Syntax

```
Call CurrentDB.Execute(<strSQL-Anweisung>)
```

Grafische Abfragen und SQL

- werden von MS Access in SQL übersetzt
- SQL-Ansicht zeigt das generierte SQL
- Ausführung SQL ist direkt über den entsprechenden Abfragetyp in der SQL-Ansicht möglich

Inhalt

Ziel und Einordnung

Rückblick

- Beziehungen im Relationen Modell
- Fremdschlüssel
- Integritätsregeln
- Operationen auf Relationen und Tupeln

Grundlagen des Relationalen Datenmodell (Teil 3) - SQL

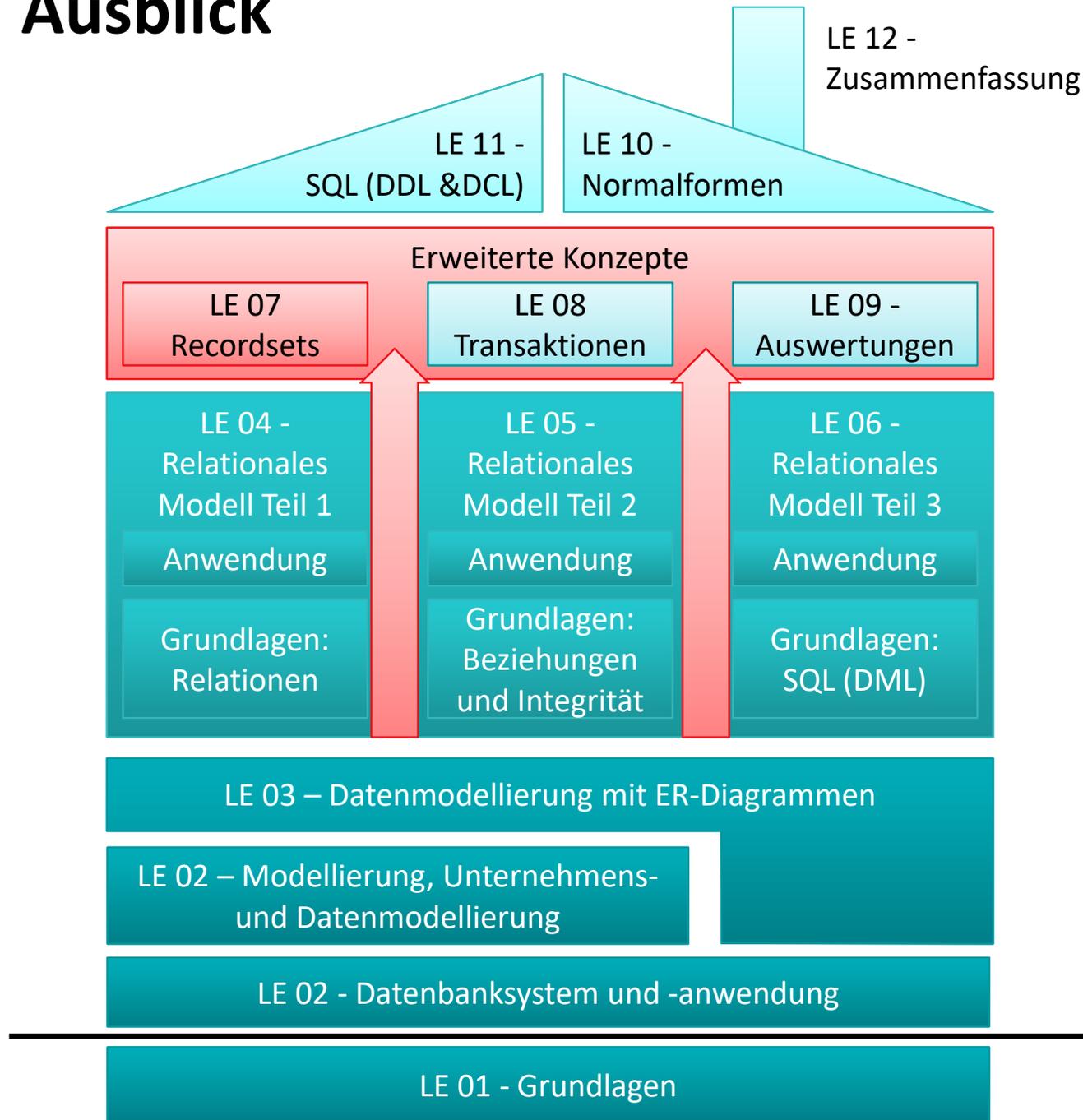
- Bestandteile
- Sprachumfang zum Auswählen, Einfügen, Ändern und Löschen
- Zusammenfassung

Arbeiten mit dem Relationalen Modell (Teil 3) - SQL in MS Access

- Formulare und SQL-Auswahlabfragen
- Vordefinierte VBA-Funktionen zum Einfügen, Ändern und Löschen mit SQL
- Zusammenhang und Abgrenzung grafischer Abfragen von SQL-Abfragen
- Zusammenfassung

Ausblick

Ausblick



Quellen

Literatur

- [1] Jürgen Auer: Sql-Tutorial. <http://www.sql-und-xml.de/sql-tutorial/>

Abbildungen

- [1] Computer History Museum: Hall of Fellows, Donald Chamberlin;
<http://www.computerhistory.org/fellowawards/hall/bios/DonaldChamberlin/>
- [2] Homepage of Don Chamberlin;
<http://www.almaden.ibm.com/cs/people/chamberlin/>

Wirtschaftsinformatik 2

LE 06 – Relationales Modell (Teil 3)

SQL

Prof. Dr. Thomas Off

<http://www.ThomasOff.de/lehre/beuth/wi2>