



BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

Wirtschaftsinformatik 2

LE 07 – Recordsets als Datenbankschnittstelle

Prof. Dr. Thomas Off

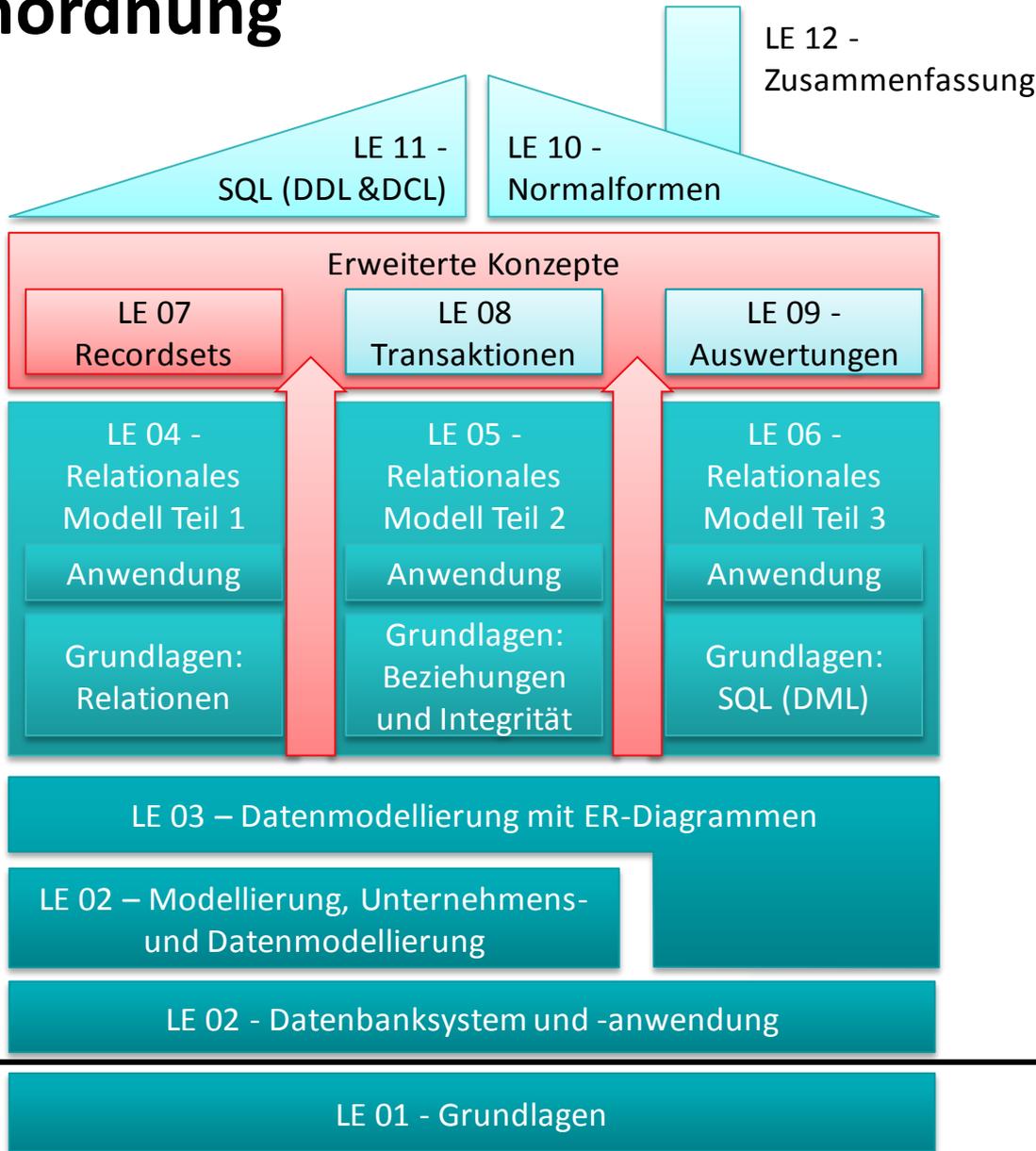
<http://www.ThomasOff.de/lehre/beuth/wi2>



Ziel

- Aufgaben der Datenbankschnittstelle kennenlernen
- Verstehen der Konzeptes eines Recordsets
- Arbeiten mit Recordsets und Anwendung in MS Access
 - Lesen von Daten mit Recordsets
 - Navigation in Recordsets
 - Suchen und Finden von Daten in Recordsets
 - Ändern und Hinzufügen von Daten mit Recordsets

Einordnung





Inhalt

Ziel und Einordnung

Rückblick

Ausgangssituation

Datenbankschnittstelle

Recordsets

- Was sind Recordsets
- Mit Recordsets Daten lesen
- Navigation im Recordset per Zeiger
- Suchen und Finden im Recordset
- Ändern und Hinzufügen im Recordset
- Kombination mit grafischen Abfragen und SQL

Weitere VBA-Funktionen

Fallstricke

Zusammenfassung und Ausblick



Inhalt

Ziel und Einordnung

Rückblick

Ausgangssituation

Datenbankschnittstelle

Recordsets

- Was sind Recordsets
- Mit Recordsets Daten lesen
- Navigation im Recordset per Zeiger
- Suchen und Finden im Recordset
- Ändern und Hinzufügen im Recordset
- Kombination mit grafischen Abfragen und SQL

Weitere VBA-Funktionen

Fallstricke

Zusammenfassung und Ausblick

Rückblick



Rückblick



Datensätze auswählen mit SQL

```
SELECT vorname, name 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

Rückblick



Datensätze auswählen mit SQL

```
SELECT vorname, name 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



**Restriktion/
Selektion**

Projektion

Rückblick



Datensätze auswählen mit SQL

```
SELECT vorname, name FROM kunden WHERE  
ort='Berlin';
```

Ausgangsrelation

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

```
SELECT vorname, name FROM kunden  
WHERE ort='Berlin';
```

Ergebnis



Rückblick



Datensätze auswählen mit SQL

```
SELECT vorname, name 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 vorname, name FROM kunden  
WHERE ort='Berlin';
```

Ergebnis

Vorname	Name
Vera	Boehrs
Ulrike	Dinkel
Thomas	Dinkels

Reihenfolge der Spalten, wie in
SELECT-Anweisung angegeben,
d.h. erst Vorname, dann Name!

Rückblick



Datensätze einfügen mit 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



Rückblick



Datensätze einfügen mit 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

Rückblick



Ändern von Datensätzen mit SQL

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

Ausgangsrelation

Kunden	KndNr	Name	Vorname
	123	Albers	Willi
	234	Boehrs	Vera
	345	Muster	Michael

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

Ergebnisrelation



Rückblick



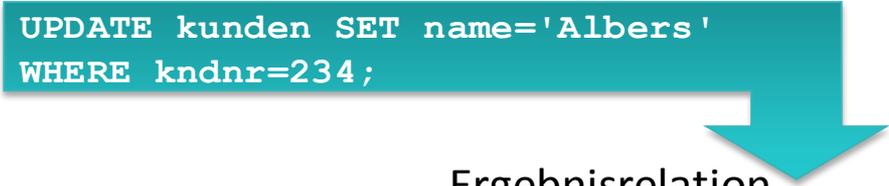
Ändern von Datensätzen mit SQL

```
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

Rückblick



Löschen von Datensätzen mit 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



Rückblick



Löschen von Datensätzen mit 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

Rückblick



Datensätze aus Tabellen mit INNER JOIN verbinden

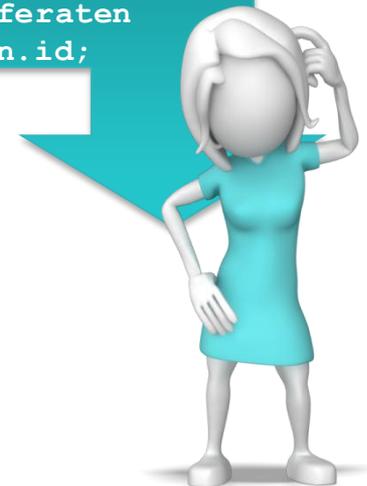
```
SELECT produkte.pid, produkte.name,  
       lieferanten.id, lieferaten.firma  
FROM produkte INNER JOIN lieferaten  
ON produkte.lid = lieferanten.id;
```

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

```
SELECT produkte.pid, produkte.name,  
       lieferanten.id, lieferaten.firma  
FROM produkte INNER JOIN lieferaten  
ON produkte.lid = lieferanten.id;
```



Rückblick



Datensätze aus Tabellen mit INNER JOIN verbinden

```
SELECT produkte.pid, produkte.name,  
       lieferanten.id, lieferaten.firma  
FROM produkte INNER JOIN lieferaten  
ON 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

```
SELECT produkte.pid, produkte.name,  
       lieferanten.id, lieferaten.firma  
FROM produkte INNER JOIN lieferaten  
ON produkte.lid = lieferanten.id;
```



Ergebnis

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

Rückblick



Datensätze aus Tabellen mit LEFT OUTER JOIN verbinden

```
SELECT produkte.pid, produkte.name,  
       lieferanten.id, lieferaten.firma  
FROM produkte LEFT JOIN lieferaten  
ON 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

```
SELECT produkte.pid, produkte.name,  
       lieferanten.id, lieferaten.firma  
FROM produkte LEFT JOIN lieferaten  
ON produkte.lid = lieferanten.id;
```



Rückblick



Datensätze aus Tabellen mit LEFT OUTER JOIN verbinden

```
SELECT produkte.pid, produkte.name,  
       lieferanten.id, lieferaten.firma  
FROM produkte LEFT JOIN lieferaten  
ON 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

```
SELECT produkte.pid, produkte.name,  
       lieferanten.id, lieferaten.firma  
FROM produkte LEFT JOIN lieferaten  
ON produkte.lid = lieferanten.id;
```

Ergebnis

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

Rückblick



Datensätze aus Tabellen mit RIGHT OUTER JOIN verbinden

```
SELECT produkte.pid, produkte.name,  
       lieferanten.id, lieferaten.firma  
FROM produkte RIGHT JOIN lieferaten  
ON 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

```
SELECT produkte.pid, produkte.name,  
       lieferanten.id, lieferaten.firma  
FROM produkte RIGHT JOIN lieferaten  
ON produkte.lid = lieferanten.id;
```



Rückblick



Datensätze aus Tabellen mit RIGHT OUTER JOIN verbinden

```
SELECT produkte.pid, produkte.name,  
       lieferanten.id, lieferaten.firma  
FROM produkte RIGHT JOIN lieferaten  
ON 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

```
SELECT produkte.pid, produkte.name,  
       lieferanten.id, lieferaten.firma  
FROM produkte RIGHT JOIN lieferaten  
ON produkte.lid = lieferanten.id;
```

Ergebnis

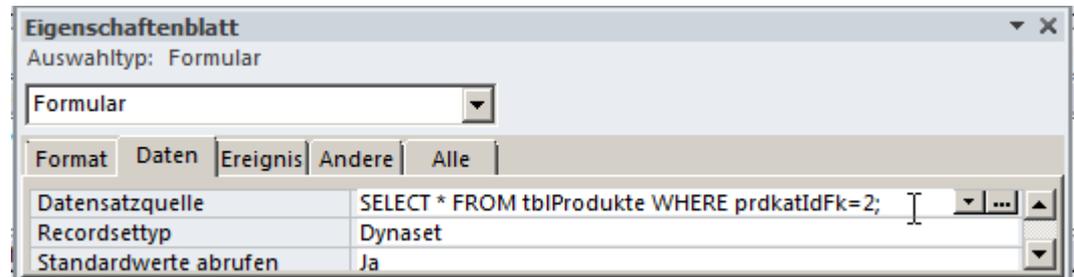
PID	Name	ID	Firma
123	Multi AB	987	Müller AG
234	Flexi 123	987	Müller AG
345	Mega+	876	Meier GmbH
		765	Bach&Sohn

Rückblick



SQL in MS Access

- SQL-Abfragen können als Datenquelle für Formulare verwendet werden
- Einfügen, Ändern und Löschen mit SQL mit vordefinierter VBA-Funktionen



```
CurrentDB.Execute("<SQL-Anweisung>")
```

- Grafische Abfragen und SQL
 - werden von MS Access in SQL übersetzt
 - SQL-Ansicht zeigt das generierte SQL
 - Ausführung von SQL direkt über SQL-Ansicht eines leeren Abfrageentwurfs oder Abfrage eines entsprechenden Abfragetyps möglich



Inhalt

Ziel und Einordnung

Rückblick

Ausgangssituation

Datenbankschnittstelle

Recordsets

- Was sind Recordsets
- Mit Recordsets Daten lesen
- Navigation im Recordset per Zeiger
- Suchen und Finden im Recordset
- Ändern und Hinzufügen im Recordset
- Kombination mit grafischen Abfragen und SQL

Weitere VBA-Funktionen

Fallstricke

Zusammenfassung und Ausblick





Inhalt

Ziel und Einordnung

Rückblick

Ausgangssituation

Datenbankschnittstelle

Recordsets

- Was sind Recordsets
- Mit Recordsets Daten lesen
- Navigation im Recordset per Zeiger
- Suchen und Finden im Recordset
- Ändern und Hinzufügen im Recordset
- Kombination mit grafischen Abfragen und SQL

Weitere VBA-Funktionen

Fallstricke

Zusammenfassung und Ausblick



Ausgangssituation

Benutzeroberfläche der Anwendung

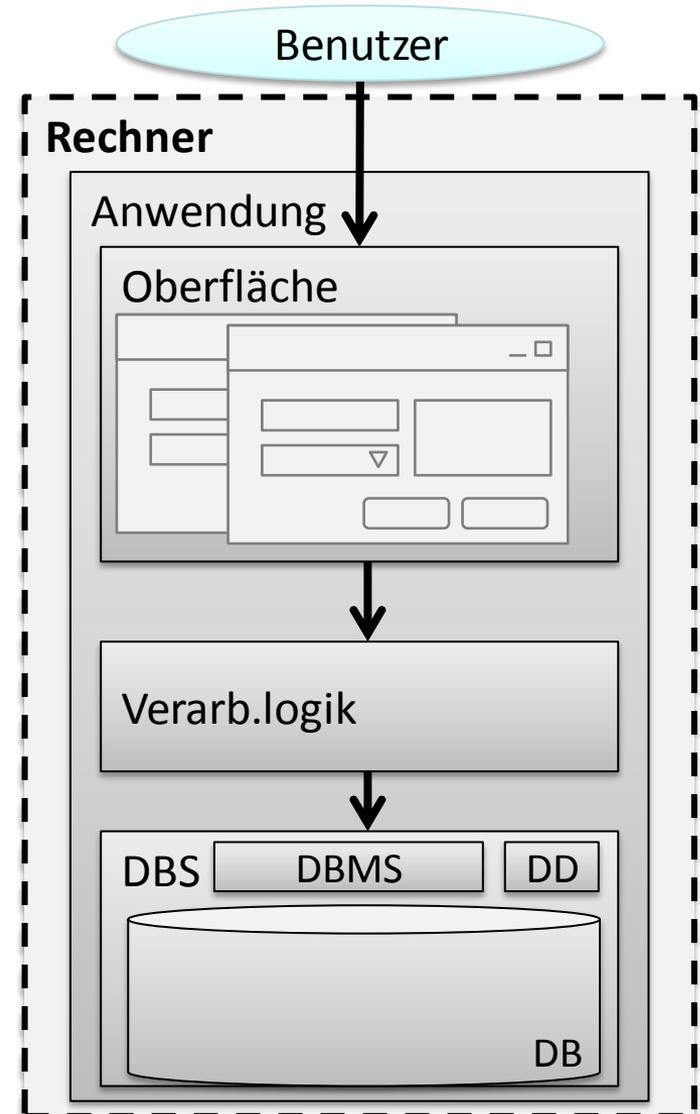
- Formulare mit Feldern, Schaltfläche usw.
- Nutzung der Verarbeitungslogik

Verarbeitungslogik

- Verarbeitung der Daten in der Oberfläche vor ihrer Speicherung
- Verarbeitung aus der DB geladener Daten vor ihrer Anzeige

Datenbank

- mit Tabellen (Relationen)
- und Daten (Tupeln)





Ausgangssituation

Benutzeroberfläche der Anwendung

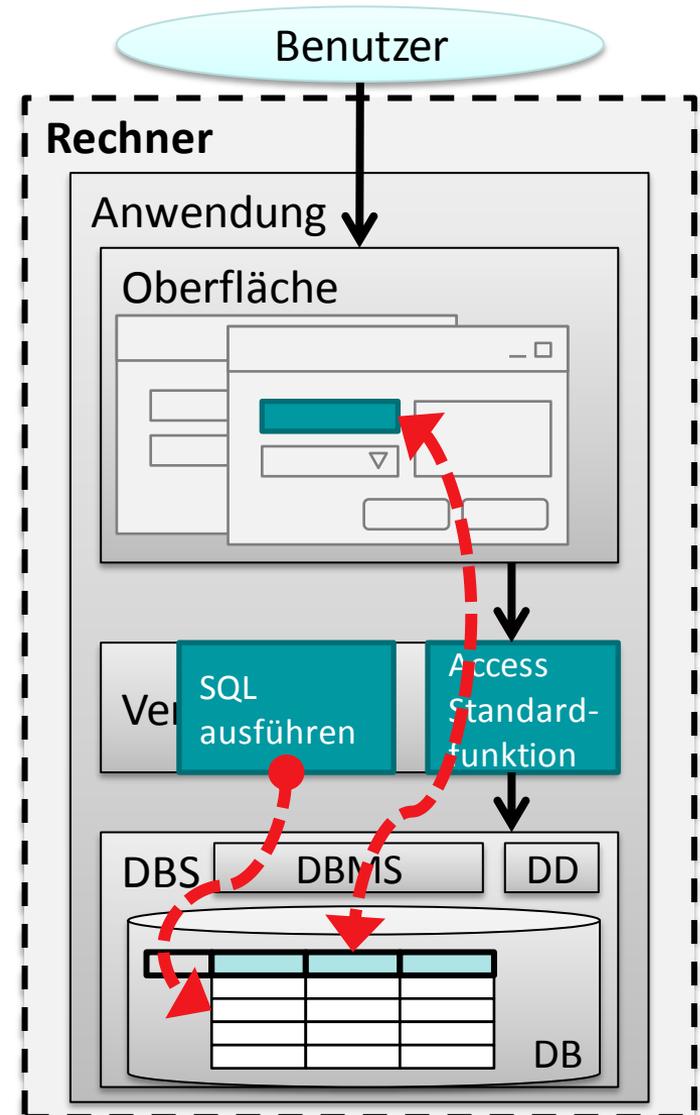
- Formulare mit Feldern, Schaltfläche usw.
- Nutzung der Verarbeitungslogik

Verarbeitungslogik

- bisher lediglich durch Nutzung der Standard-Access-Funktionen
 - zur Anzeige in Oberfläche
 - zum Ausführen von Abfragen/SQL

Datenbank

- mit Tabellen (Relationen)
- und Daten (Tupeln)



Toll!



Toll!



Aber...

- Geburtsdatum des Kunden lesen, sein Alter errechnen und in einer Variable speichern
- Geschlecht des Kunden lesen, passende Anrede "Frau" bzw. "Herr" mit dem Namen verbinden und in MessageBox ausgeben
- ...

Überhaupt irgendwas aus der Datenbank lesen, um es in einer Variable zu speichern.





Ausgangssituation

Benutzeroberfläche der Anwendung

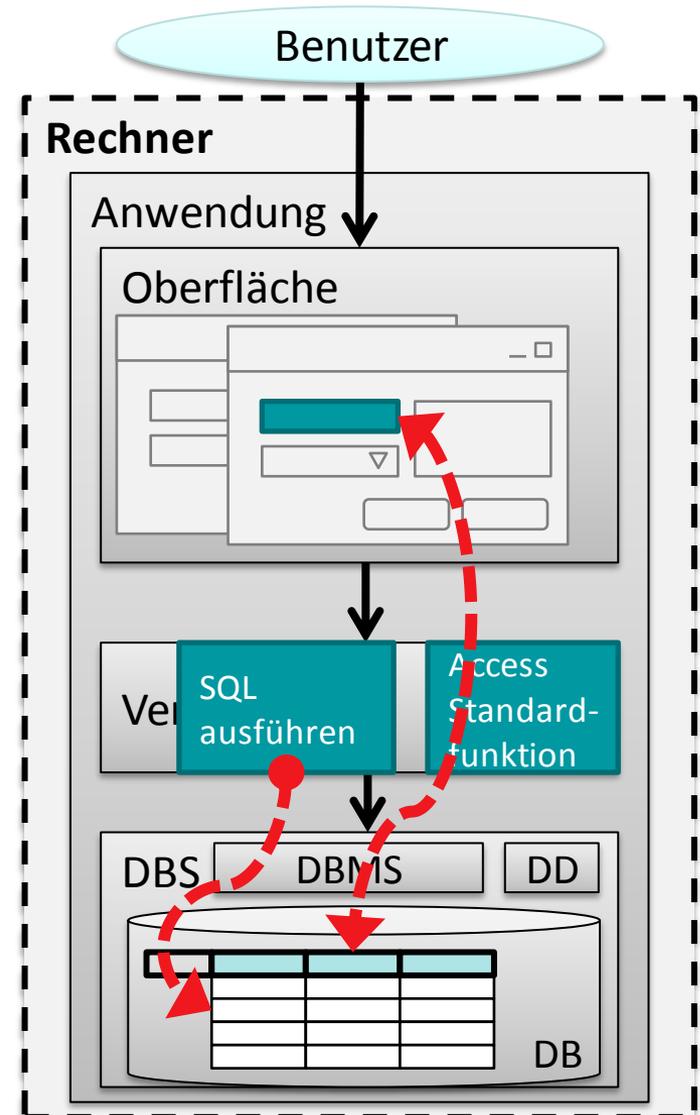
- Formulare mit Feldern, Schaltfläche usw.
- Nutzung der Verarbeitungslogik

Verarbeitungslogik

- bisher lediglich durch Nutzung der Standard-Access-Funktionen
 - zur Anzeige in Oberfläche
 - zum Ausführen von Abfragen/SQL

Datenbank

- mit Tabellen (Relationen)
- und Daten (Tupeln)





Ausgangssituation

Benutzeroberfläche der Anwendung

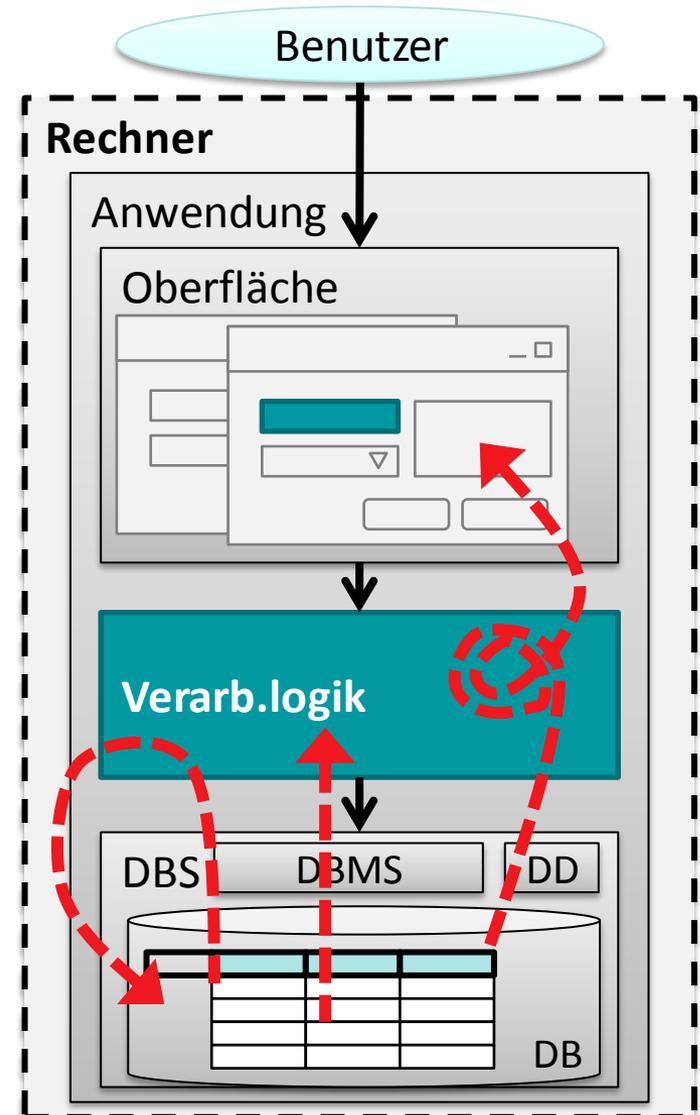
- Formulare mit Feldern, Schaltfläche usw.
- Nutzung der Verarbeitungslogik

Verarbeitungslogik

- jetzt Nutzung einer Möglichkeit direkt auf die Daten der zuzugreifen und
- Daten zu verarbeiten, ohne eine Verbindung mit der Oberfläche herstellen zu müssen

Datenbank

- mit Tabellen (Relationen)
- und Daten (Tupeln)





Ausgangssituation

Benutzeroberfläche der Anwendung

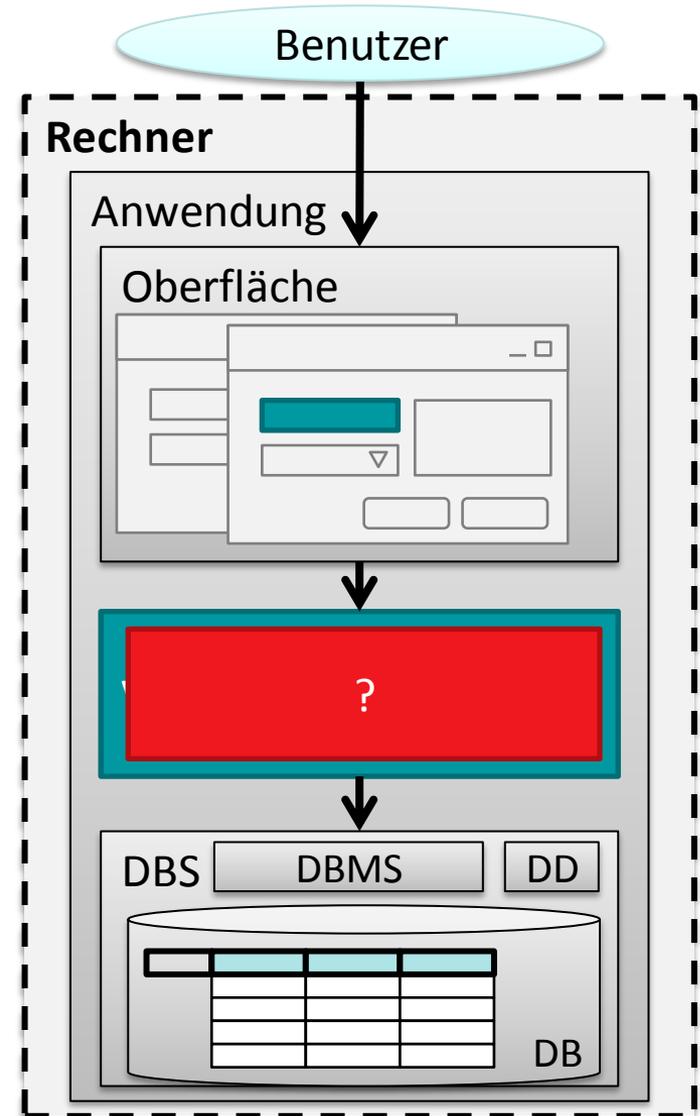
- Formulare mit Feldern, Schaltfläche usw.
- Nutzung der Verarbeitungslogik

Verarbeitungslogik

- jetzt Nutzung einer Möglichkeit direkt auf die Daten der zuzugreifen und
- Daten zu verarbeiten, ohne eine Verbindung mit der Oberfläche herstellen zu müssen

Datenbank

- mit Tabellen (Relationen)
- und Daten (Tupeln)





Inhalt

Ziel und Einordnung

Rückblick

Ausgangssituation

Datenbankschnittstelle

Recordsets

- Was sind Recordsets
- Mit Recordsets Daten lesen
- Navigation im Recordset per Zeiger
- Suchen und Finden im Recordset
- Ändern und Hinzufügen im Recordset
- Kombination mit grafischen Abfragen und SQL

Weitere VBA-Funktionen

Fallstricke

Zusammenfassung und Ausblick





Inhalt

Ziel und Einordnung

Rückblick

Ausgangssituation

Datenbankschnittstelle

Recordsets

- Was sind Recordsets
- Mit Recordsets Daten lesen
- Navigation im Recordset per Zeiger
- Suchen und Finden im Recordset
- Ändern und Hinzufügen im Recordset
- Kombination mit grafischen Abfragen und SQL

Weitere VBA-Funktionen

Fallstricke

Zusammenfassung und Ausblick

Programmierschnittstelle zur Datenbank



Programmiersprache

- dient der Programmentwicklung
- prozedurale Sprache, z.B. mit Variablen die einen Bereich des Arbeitsspeichers referenzieren und einen bestimmten Datentyp haben

Programmiersprache

```
Dim strName _  
    As String  
Let strName = ".."  
  
' ...
```

Programmierschnittstelle zur Datenbank



Datenbanksprache

- dient u.a. der Abfrage, Manipulation von Daten
- Konzepte relationaler Datenbanken: Tabellen, Datensätze, Spalten, Schlüssel

Programmiersprache

```
Dim strName _  
    As String  
Let strName = ".."  
  
' ...
```

Datenbanksprache

```
SELECT * FROM  
Kunden  
WHERE Stadt='Köln'
```

Programmierschnittstelle zur Datenbank



Um innerhalb einer Programmiersprache auf die Datenbank zugreifen zu können, ...

Programmiersprache

```
Dim strName _  
    As String  
Let strName = ".."  
  
' ...
```



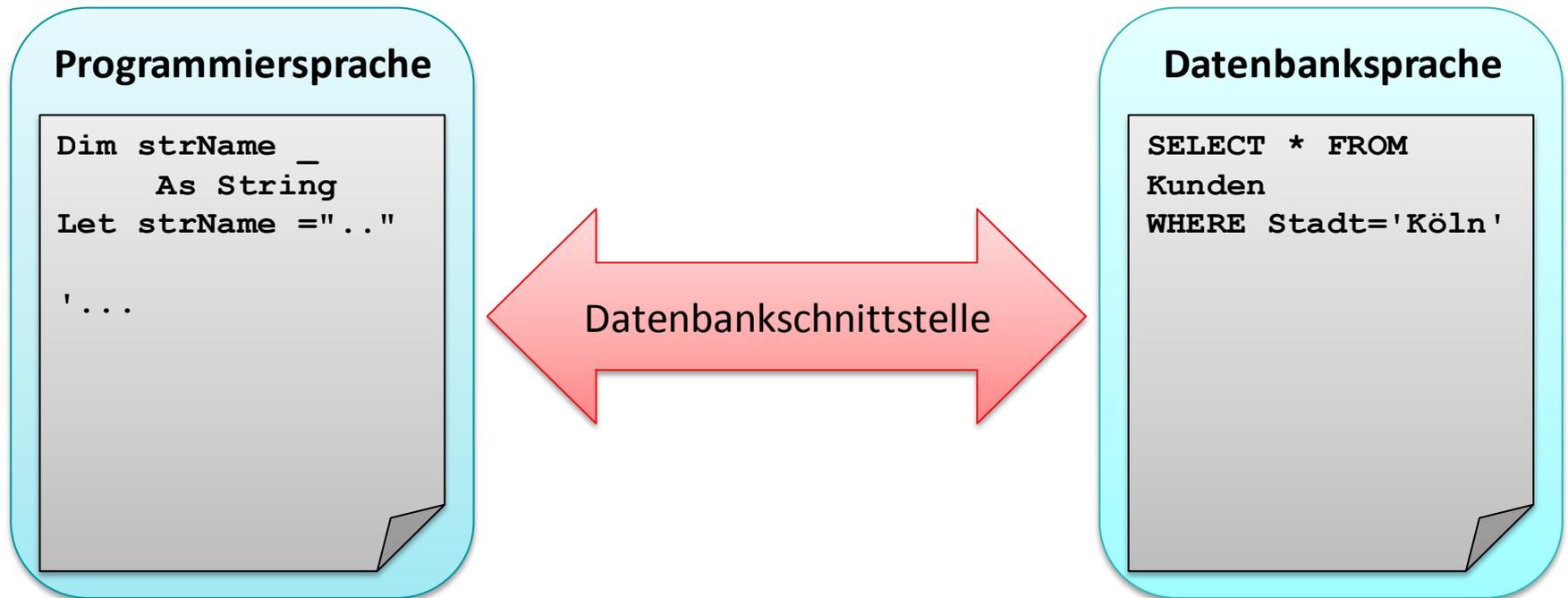
Datenbanksprache

```
SELECT * FROM  
Kunden  
WHERE Stadt='Köln'
```

Programmierschnittstelle zur Datenbank



Um innerhalb einer Programmiersprache auf die Datenbank zugreifen zu können, muss eine Verbindung zwischen Sprachelementen der Programmiersprache und Elementen der Datenbanksprache existieren:



Funktionen einer Datenbankschnittstelle



Funktionen einer Datenbankschnittstelle

- Übersetzen der Befehle einer Programmiersprache in Befehle der Datenbanksprache
- Übersetzen des Ergebnisses einer Datenbankabfrage in Elemente einer Programmiersprache
- Bereitstellen des Funktionsumfangs für
 - Abfragen von Datensätzen
 - Auswählen nach bestimmten Kriterien
 - Navigation über die gefundenen Datensätze
 - Einfügen von neuen Datensätzen
 - Ändern vorhandener Datensätze
 - Löschen vorhandener Datensätze

→ Wir lernen diese Grundkonzepte anhand von VBA-Recordsets kennen



Inhalt

Ziel und Einordnung

Rückblick

Ausgangssituation

Datenbankschnittstelle

Recordsets

- Was sind Recordsets
- Mit Recordsets Daten lesen
- Navigation im Recordset per Zeiger
- Suchen und Finden im Recordset
- Ändern und Hinzufügen im Recordset
- Kombination mit grafischen Abfragen und SQL

Weitere VBA-Funktionen

Fallstricke

Zusammenfassung und Ausblick



Was sind Recordsets?



Recordset

Was sind Recordsets?

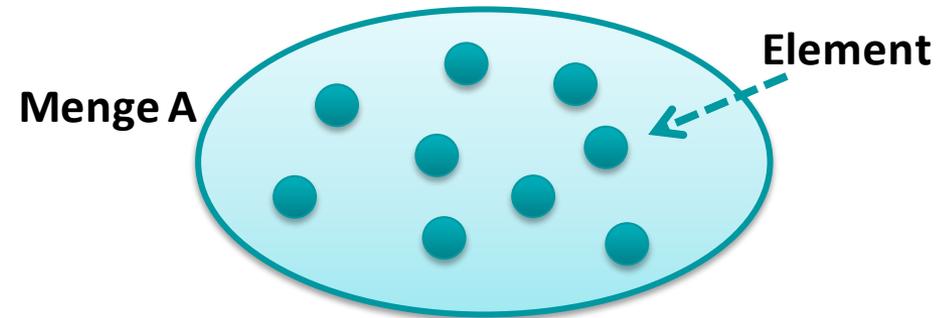


Recordset

Set = Menge

Menge

- fasst Elemente zusammen
- Sonderform „leere Menge“
und unendliche Menge



Was sind Recordsets?



Recordset

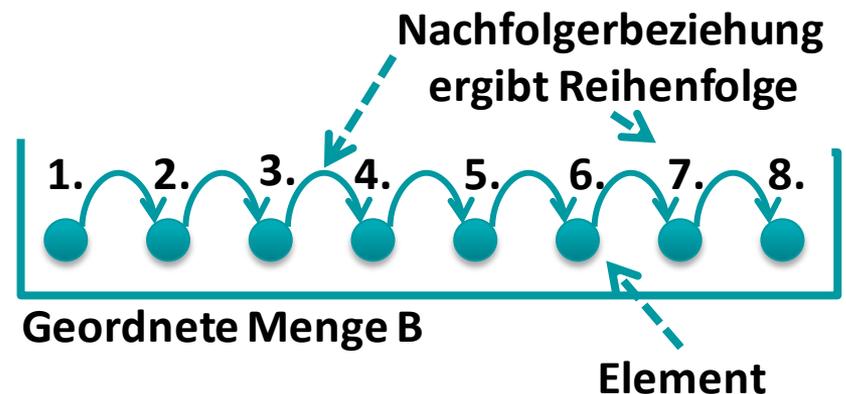
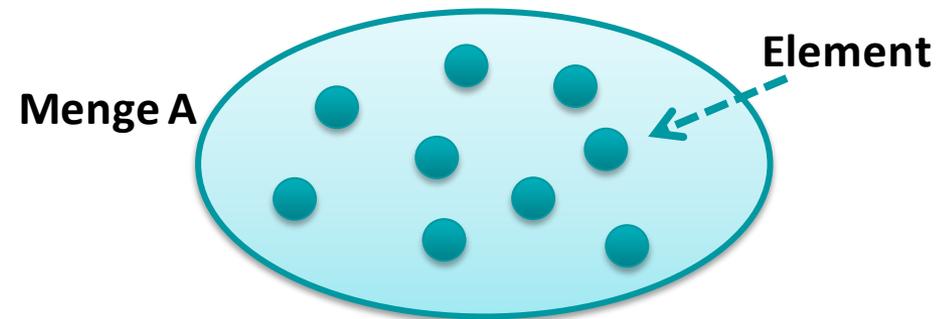
Set = Menge

Menge

- fasst Elemente zusammen
- Sonderform „leere Menge“ und unendliche Menge

Geordnete Menge

- fasst Elemente zusammen
- definiert eine „Reihenfolge“ zwischen Elementen (z.B. durch Nachfolgerbeziehung)
- entspricht mathematischer Ordnungsrelation



Was sind Recordsets?



Recordset

Record = Datensatz

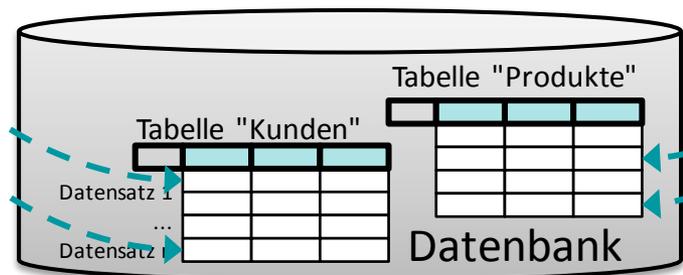
Datensatz

- repräsentiert etwas, z.B. eine Person, ein Produkt mit Eigenschaften
- fasst einzelne Datenfelder zusammen, z.B. Name, Vorname, Geburtsdatum
- wird als Zeile in Tabellen

Knd234	Müller	Mike	Köln
Knd345	Müller	Sabine	Bonn
Knd456	Schmidt	Susi	Berlin

Prd01	Multi XY	150 €
-------	----------	-------

Prd02	Flexi AC	75 €
-------	----------	------



Was sind Recordsets?



Recordset

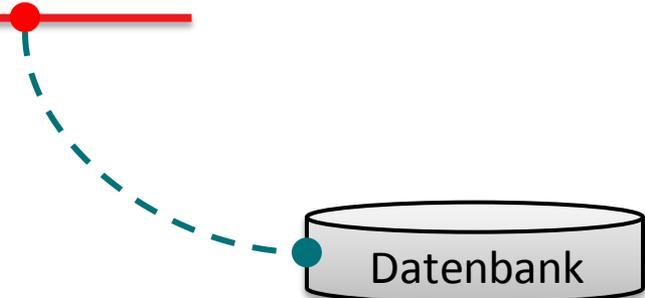
**Recordset = geordnete
Menge gleichartiger
Datensätze**

Beispiele:

- Recordset von Personen

Recordset von Personen

Knd234	Müller	Mike	Köln
Knd345	Müller	Sabine	Bonn
Knd456	Schmidt	Susi	Berlin
Knd678	Schmidt	Hans	Bremen
Knd789	Yilmaz	Ali	Berlin



Was sind Recordsets?



Recordset

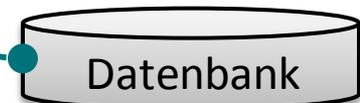
**Recordset = geordnete
Menge gleichartiger
Datensätze**

Beispiele:

- Recordset von Personen
- Recordset von Produkten

Recordset von Produkten

Prd01	Multi XY	150 €
Prd02	Flexi AC	75 €
Prd03	Hyper R	175 €
Prd04	C3PO	80 €



Was sind Recordsets?



Recordset

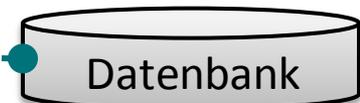
**Recordset = geordnete
Menge gleichartiger
Datensätze**

Beispiele:

- Recordset von Personen
- Recordset von Produkten
- Recordset von Kunde und Produkt

Recordset von Kunde und Produkt

Müller	Mike	Multi XY	150 €
Yilmaz	Ali	Flexi AC	75 €
Müller	Sabine	Hyper R	175 €
Schmidt	Susi	C3PO	80 €



Was sind Recordsets?



Recordset

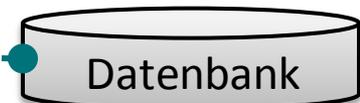
**Recordset = geordnete
Menge gleichartiger
Datensätze**

Kein Beispiel

- 2 Kunden und
2 Produkte
- Nicht gleichartig

Kein Recordset!

Müller	Mike	Köln
Yilmaz	Ali	Berlin
Multi XY	150 €	
Flexi AC	75 €	



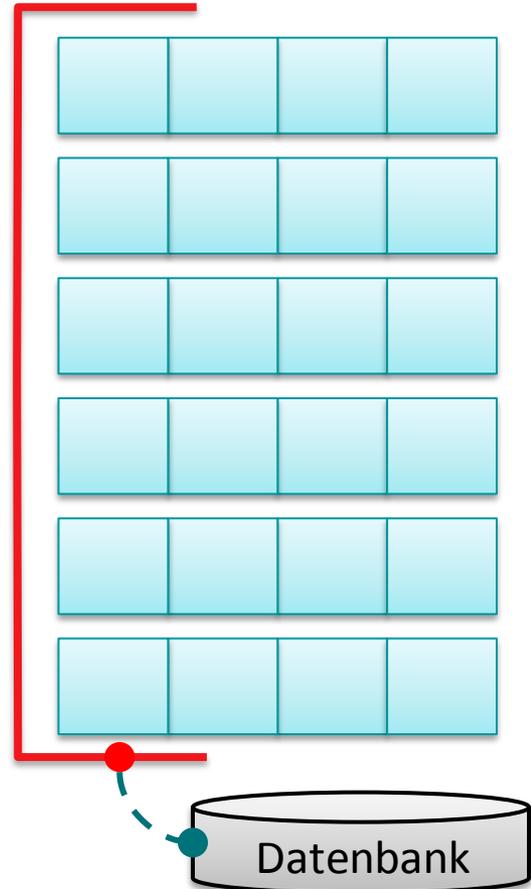
Was sind Recordsets?



Eigenschaften

Recordset

Recordset



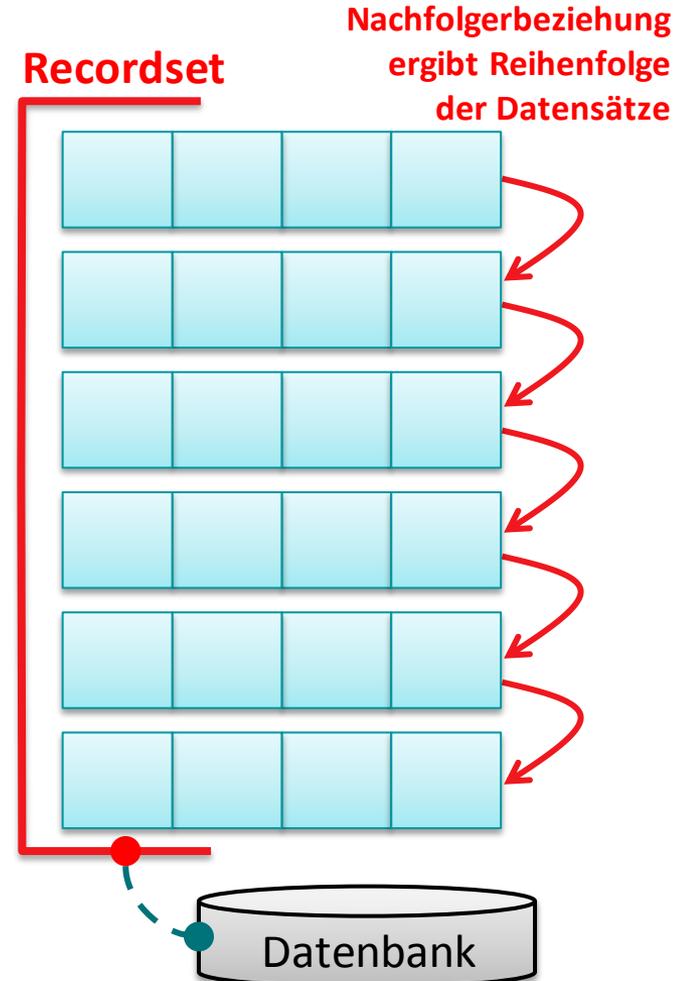
Was sind Recordsets?



Eigenschaften

- Geordnete Reihenfolge durch Nachfolger-Beziehung zw. Datensätzen

Recordset



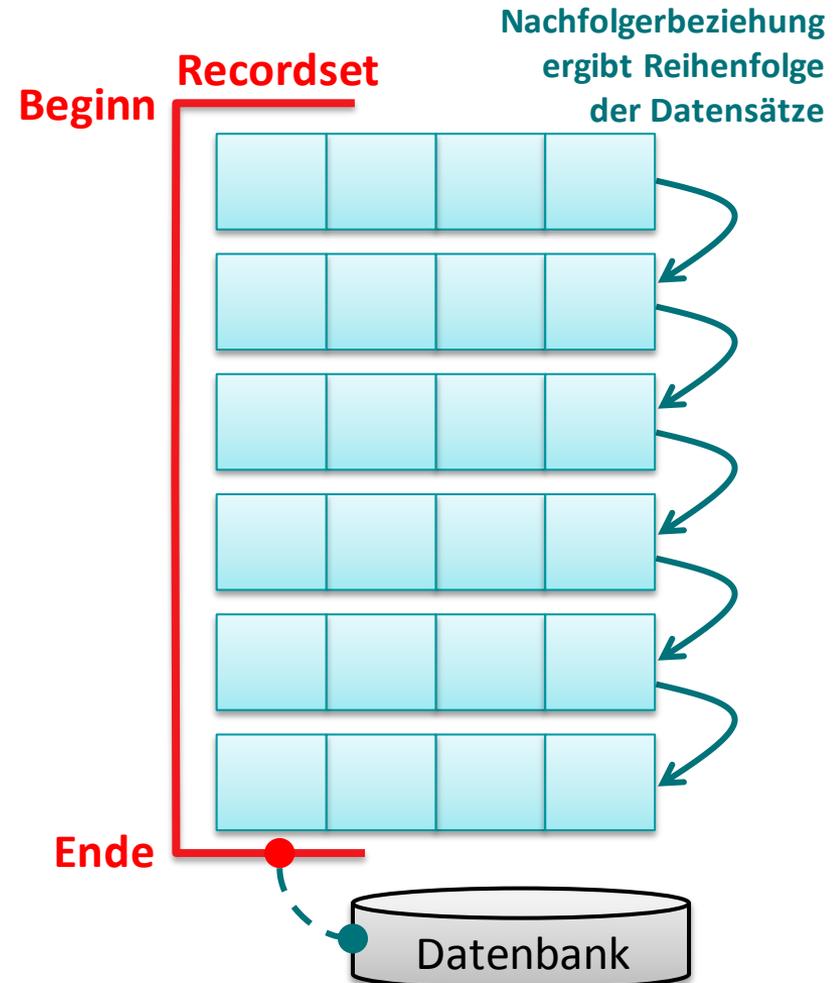


Was sind Recordsets?

Eigenschaften

- Geordnete Reihenfolge durch Nachfolger-Beziehung zw. Datensätzen
- Beginn: vor dem ersten Datensatz
- Ende: hinter dem letzten Datensatz

Recordset



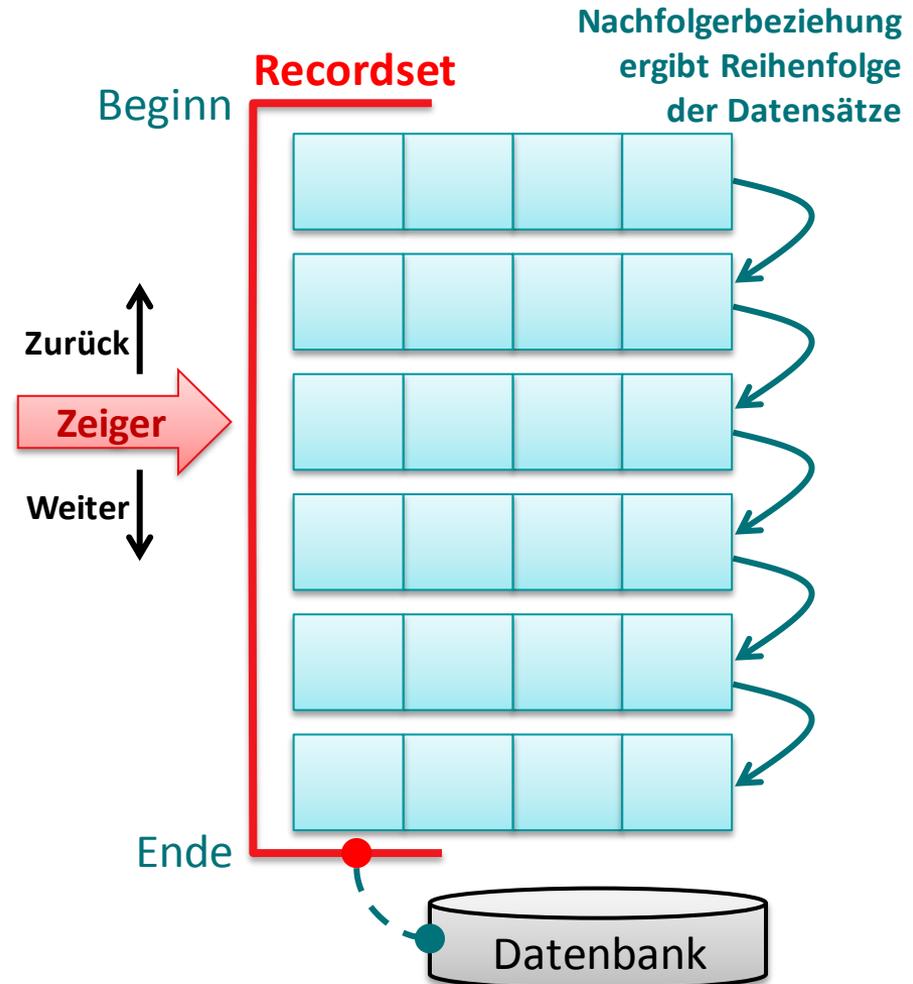


Was sind Recordsets?

Eigenschaften

- Geordnete Reihenfolge durch Nachfolger-Beziehung zw. Datensätzen
- Beginn: vor dem ersten Datensatz
- Ende: hinter dem letzten Datensatz
- Zeiger
 - der auf das aktuelle Element, den Beginn oder das Ende zeigt
 - zum nächsten Datensatz bewegt werden kann

Recordset



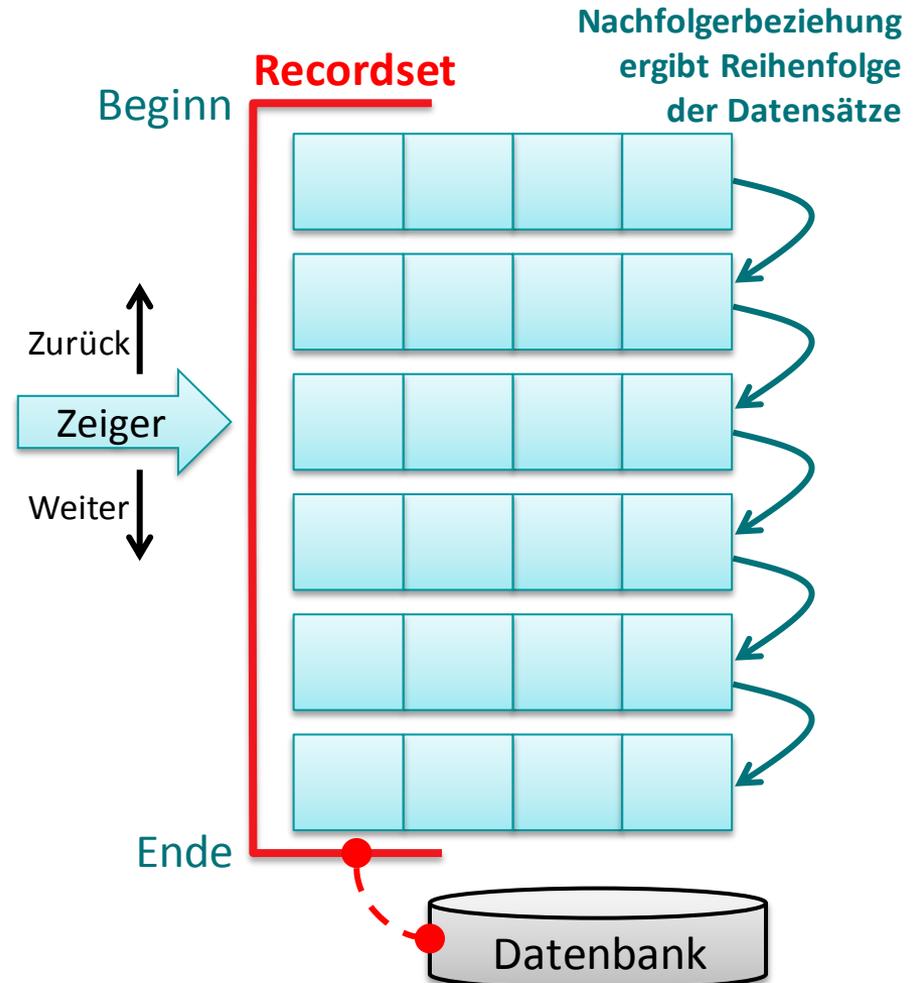


Was sind Recordsets?

Eigenschaften

- Geordnete Reihenfolge durch Nachfolger-Beziehung zw. Datensätzen
- Beginn: vor dem ersten Datensatz
- Ende: hinter dem letzten Datensatz
- Zeiger
 - der auf das aktuelle Element, den Beginn oder das Ende zeigt
 - zum nächsten Datensatz bewegt werden kann
- Beziehung zur Datenbank, aus der die Datensätze stammen

Recordset



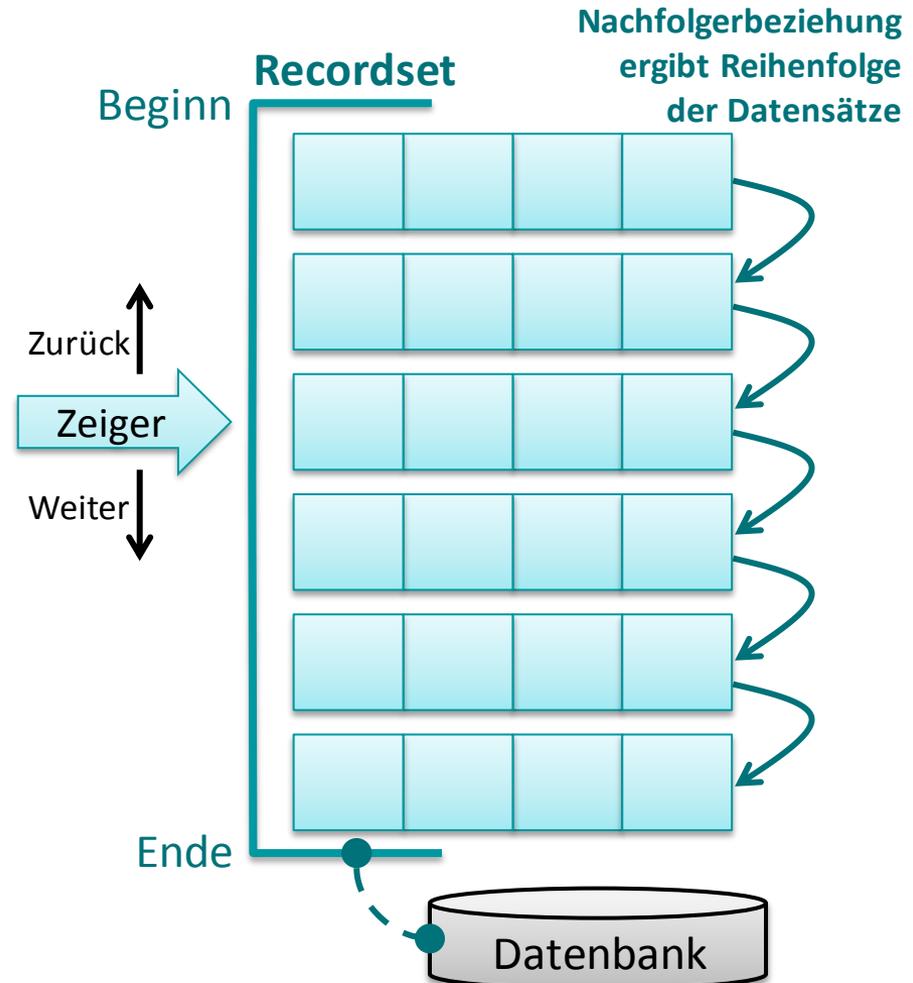


Was sind Recordsets?

Eigenschaften

- Geordnete Reihenfolge durch Nachfolger-Beziehung zw. Datensätzen
- Beginn: vor dem ersten Datensatz
- Ende: hinter dem letzten Datensatz
- Zeiger
 - der auf das aktuelle Element, den Beginn oder das Ende zeigt
 - zum nächsten Datensatz bewegt werden kann
- Beziehung zur Datenbank, aus der die Datensätze stammen

Recordset





Inhalt

Ziel und Einordnung

Rückblick

Ausgangssituation

Datenbankschnittstelle

Recordsets

- Was sind Recordsets
- Mit Recordsets Daten lesen
- Navigation im Recordset per Zeiger
- Suchen und Finden im Recordset
- Ändern und Hinzufügen im Recordset
- Kombination mit grafischen Abfragen und SQL

Weitere VBA-Funktionen

Fallstricke

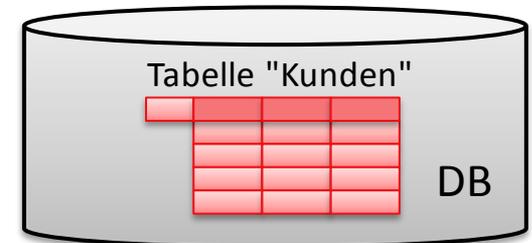
Zusammenfassung und Ausblick



Mit Recordsets Daten lesen



**Ziel: Aus Kunden den
Vornamen und Nachnamen
des ersten Kunden lesen**





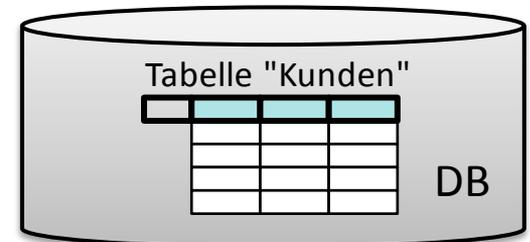
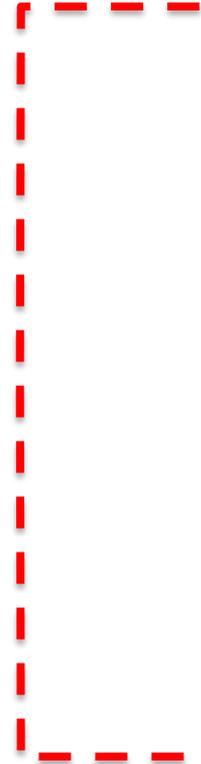
Mit Recordsets Daten lesen

**Ziel: Aus Kunden den
Vornamen und Nachnamen
des ersten Kunden lesen**

Ansatz

- Recordset deklarieren

Recordset





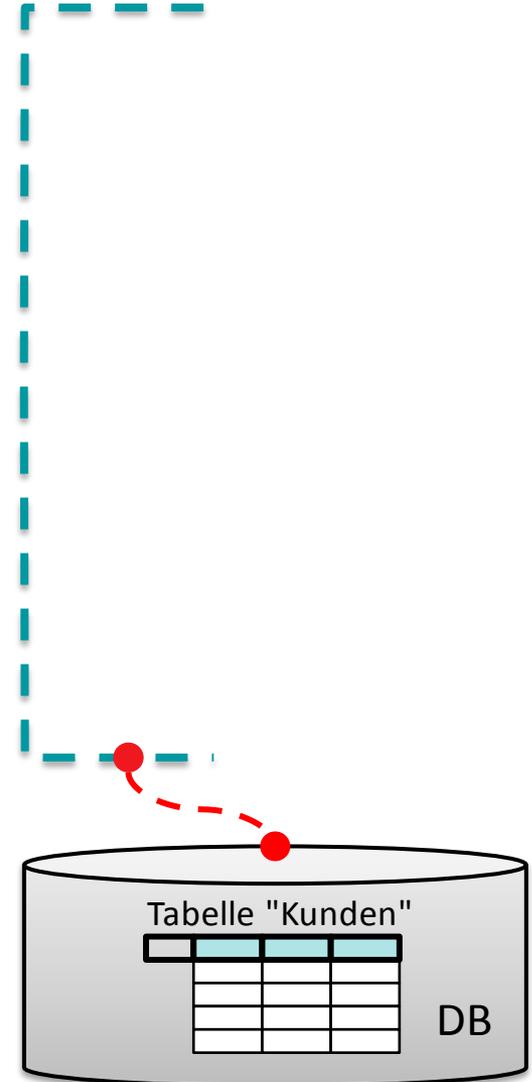
Mit Recordsets Daten lesen

**Ziel: Aus Kunden den
Vornamen und Nachnamen
des ersten Kunden lesen**

Ansatz

- Recordset deklarieren
- Verbindung zur Datenbank vorbereiten

Recordset



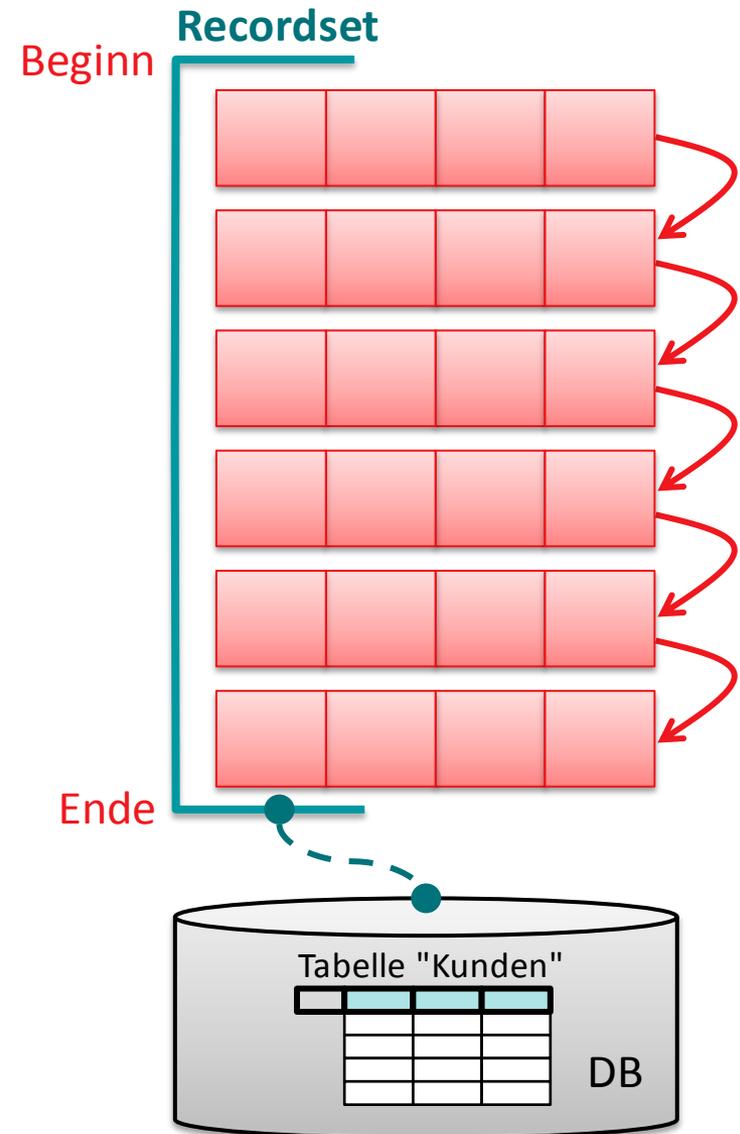
Mit Recordsets Daten lesen



Ziel: Aus Kunden den Vornamen und Nachnamen des ersten Kunden lesen

Ansatz

- Recordset deklarieren
- Verbindung zur Datenbank vorbereiten
- Recordset initialisieren und dadurch füllen



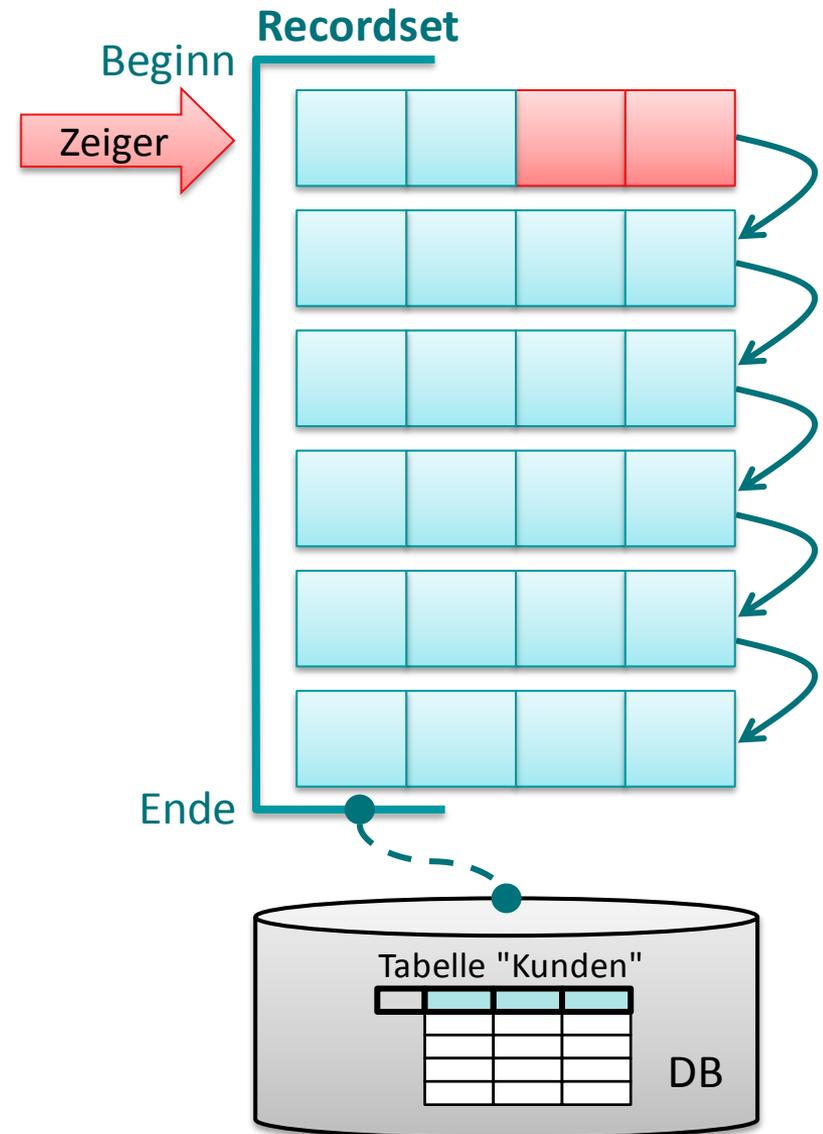
Mit Recordsets Daten lesen



Ziel: Aus Kunden den Vornamen und Nachnamen des ersten Kunden lesen

Ansatz

- Recordset deklarieren
- Verbindung zur Datenbank vorbereiten
- Recordset initialisieren und dadurch füllen
- per Zeiger (steht auf erstem Element) auf Felder zugreifen



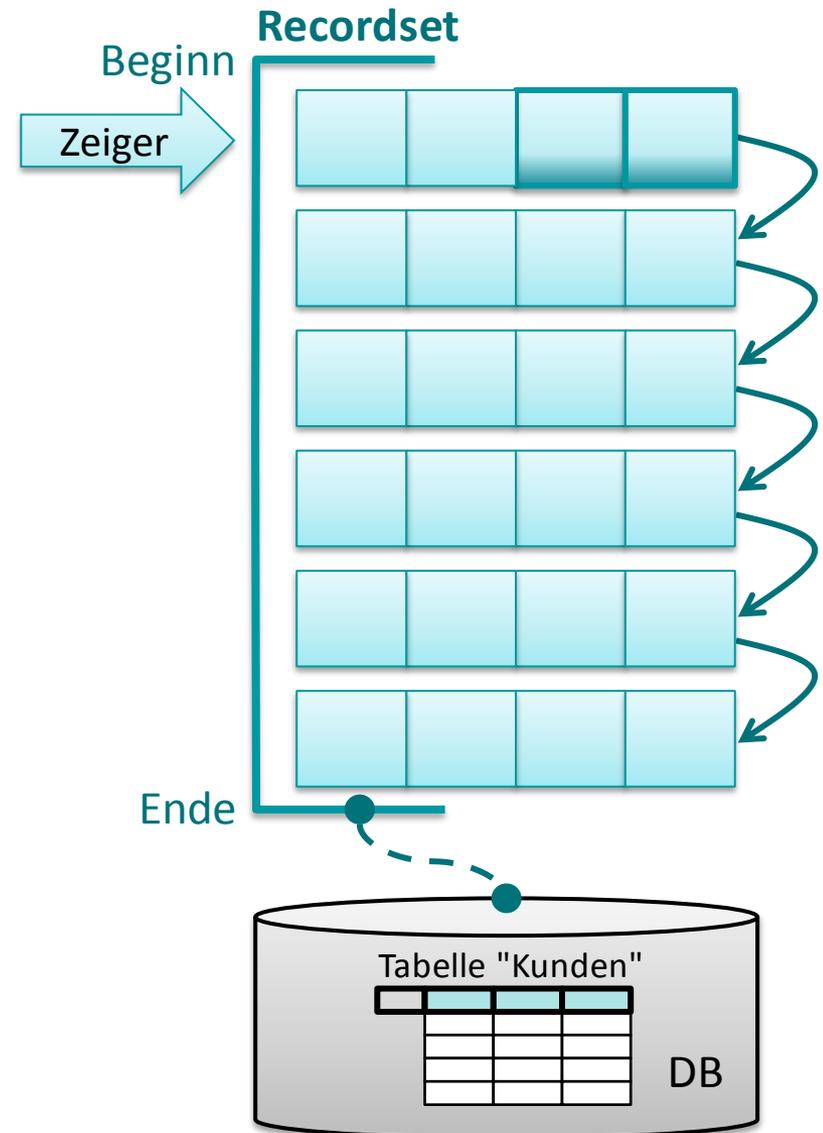
Mit Recordsets Daten lesen



Ziel: Aus Kunden den Vornamen und Nachnamen des ersten Kunden lesen

Ansatz

- Recordset deklarieren
- Verbindung zur Datenbank vorbereiten
- Recordset initialisieren und dadurch füllen
- per Zeiger (steht auf erstem Element) auf Felder zugreifen



Mit Recordsets Daten lesen

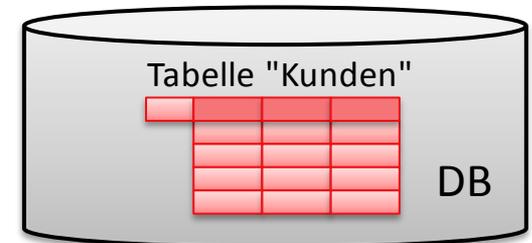


Modul

```
Option Compare Database
Option Explicit

Sub ausgebenKundeNameVorname ()

End Sub
```



Mit Recordsets Daten lesen



Modul

```
Option Compare Database
Option Explicit

Sub ausgebenKundeNameVorname ()

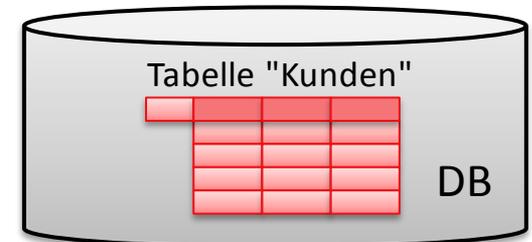
' Recordset deklarieren

' Verbindung zur DB vorbereiten

' Recordset initialisieren

' auf Felder zugreifen/ausgeben

End Sub
```



Mit Recordsets Daten lesen



Modul

```
Option Compare Database
Option Explicit

Sub ausgebenKundeNameVorname ()

' Recordset deklarieren
Dim rcsKnd As Recordset

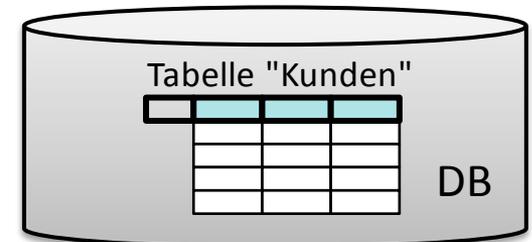
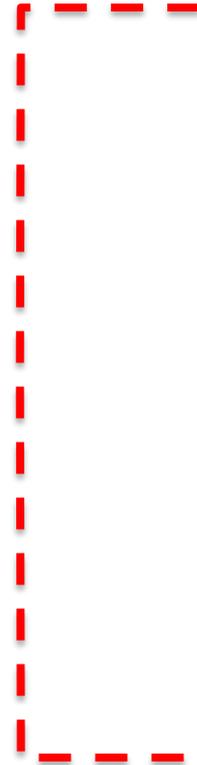
' Verbindung zur DB vorbereiten

' Recordset initialisieren

' auf Felder zugreifen/ausgeben

End Sub
```

Recordset



Mit Recordsets Daten lesen



Modul

```
Option Compare Database
Option Explicit

Sub ausgebenKundeNameVorname ()

' Recordset deklarieren
Dim rcsKnd As Recordset

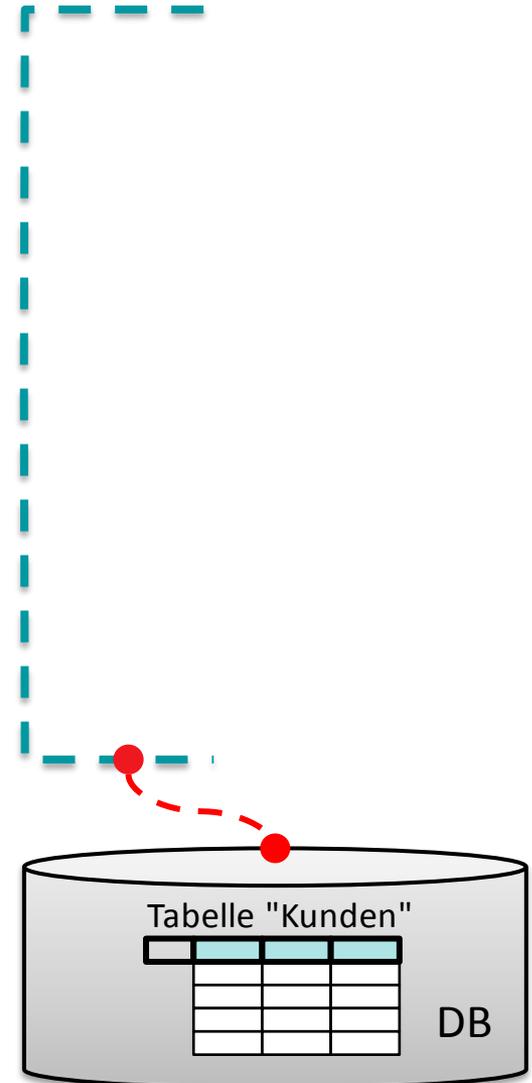
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb

' Recordset initialisieren

' auf Felder zugreifen/ausgeben

End Sub
```

Recordset



Mit Recordsets Daten lesen



Modul

```
Option Compare Database
Option Explicit

Sub ausgebenKundeNameVorname ()

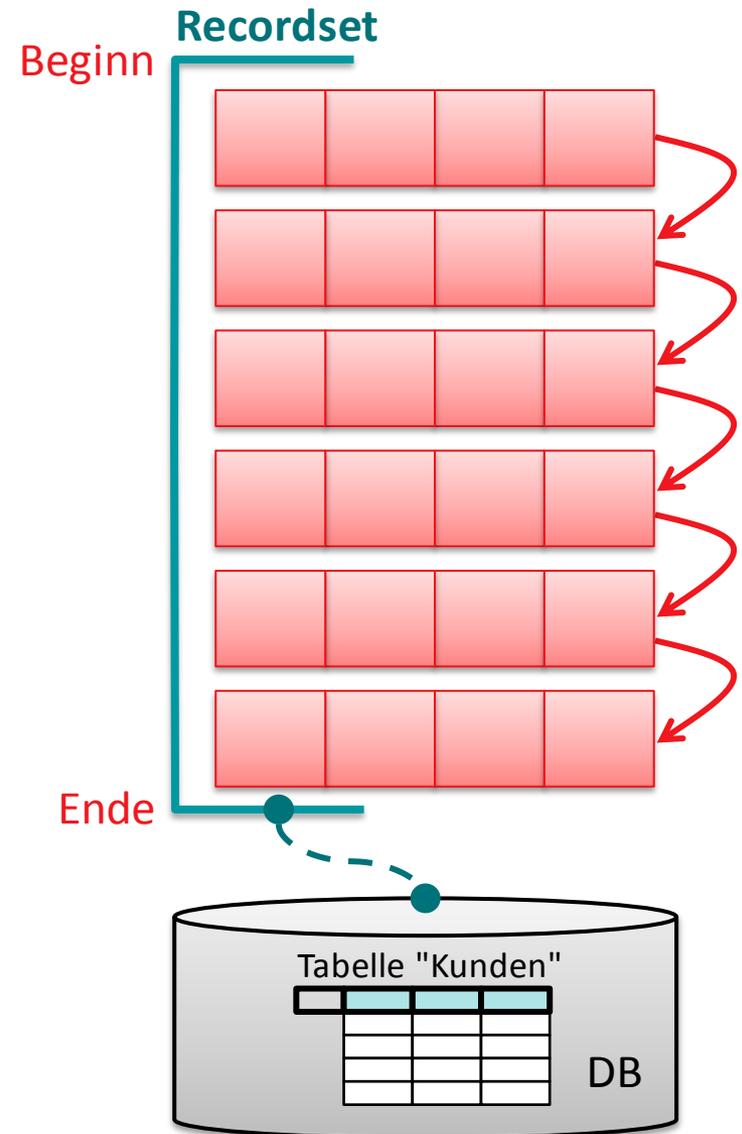
' Recordset deklarieren
Dim rcsKnd As Recordset

' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb

' Recordset initialisieren
Set rcsKnd=db.OpenRecordset ("tblKunden")

' auf Felder zugreifen/ausgeben

End Sub
```



Mit Recordsets Daten lesen



Modul

```
Option Compare Database
Option Explicit

Sub ausgebenKundeNameVorname ()

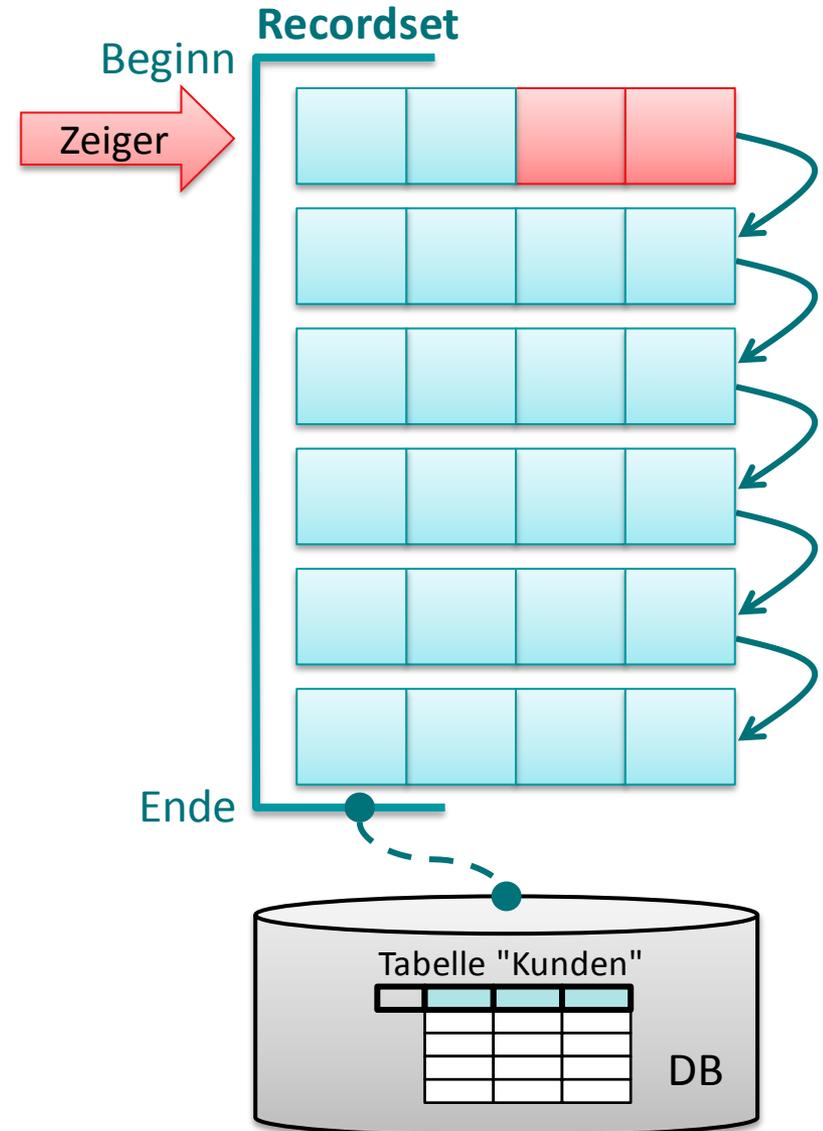
' Recordset deklarieren
Dim rcsKnd As Recordset

' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb

' Recordset initialisieren
Set rcsKnd=db.OpenRecordset ("tblKunden")

' auf Felder zugreifen/ausgeben
Debug.Print rcsKnd.Fields ("kndName")
Debug.Print rcsKnd.Fields ("kndVorname")

End Sub
```



Mit Recordsets Daten lesen



Modul

```
Option Compare Database
Option Explicit

Sub ausgebenKundeNameVorname ()

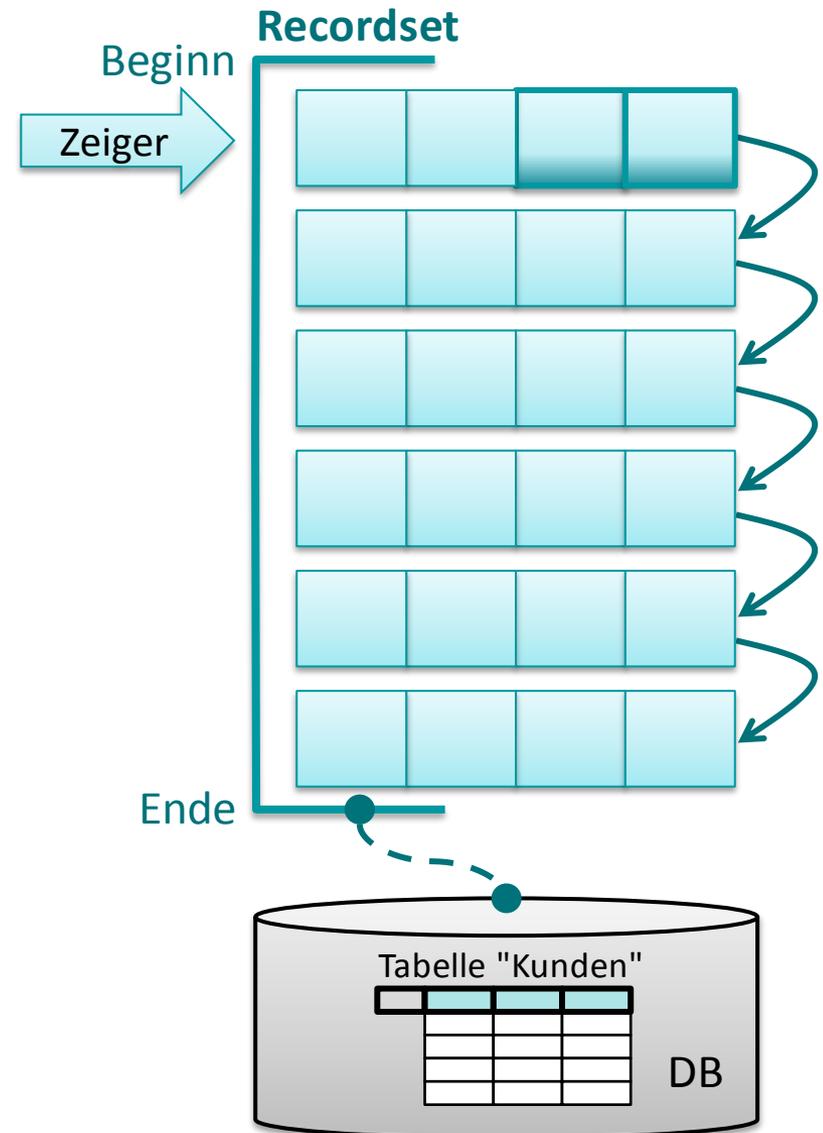
' Recordset deklarieren
Dim rcsKnd As Recordset

' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb

' Recordset initialisieren
Set rcsKnd=db.OpenRecordset ("tblKunden")

' auf Felder zugreifen/ausgeben
Debug.Print rcsKnd.Fields ("kndName")
Debug.Print rcsKnd.Fields ("kndVorname")

End Sub
```



Mit Recordsets Daten lesen



Modul

```
(Allgemein)
Option Compare Database
Option Explicit

Sub ausgebenKundeNameVorname ()

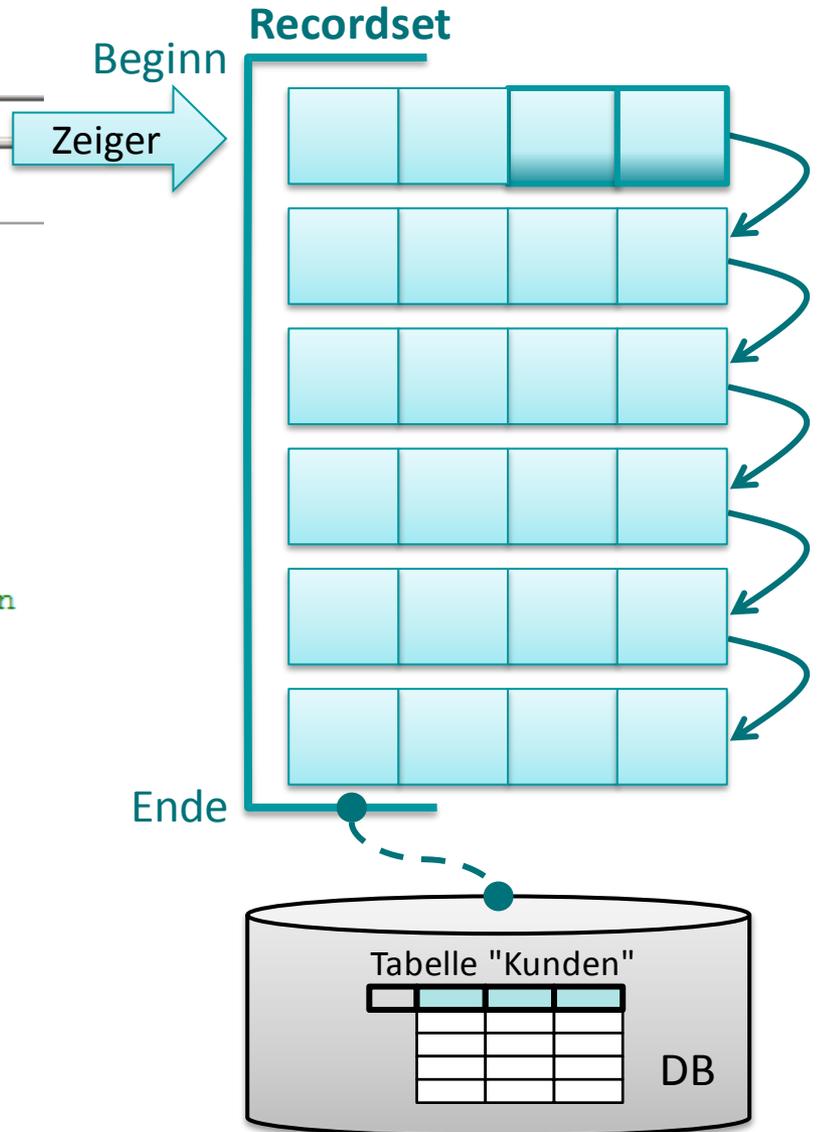
' Recordset deklarieren
Dim rcsKnd As Recordset

' Verbindung zur Datenbank vorbereiten
Dim db As Database
Set db = CurrentDb

' Recordset initialisieren und dadurch füllen
Set rcsKnd = db.OpenRecordset ("tblKunden")

' per Zeiger (steht auf erstem Element) auf
' Felder zugreifen und ausgeben
Debug.Print rcsKnd.Fields ("kndName")
Debug.Print rcsKnd.Fields ("kndVorname")

End Sub
```



Mit Recordsets Daten lesen



Modul

```
(Allgemein)

Option Compare Database
Option Explicit

Sub ausgebenKundeNameVorname ()

    ' Recordset deklarieren
    Dim rcsKnd As Recordset

    ' Verbindung zur Datenbank vorbereiten
    Dim db As Database
    Set db = CurrentDb

    ' Recordset initialisieren und dadurch füllen
    Set rcsKnd = db.OpenRecordset("tblKunden")

    ' per Zeiger (steht auf erstem Element) auf
    ' Felder zugreifen und ausgeben
    Debug.Print rcsKnd.Fields("kndName")
    Debug.Print rcsKnd.Fields("kndVorname")

End Sub
```



	kndIdPk	kndName	kndVorname
+	1	Meier	Thomas
+	2	Müller	Sven
+	3	Meier	Mike
+	4	Müller	Silke

Mit Recordsets Daten lesen



Modul

```
(Allgemein)

Option Compare Database
Option Explicit

Sub ausgebenKundeNameVorname ()

    ' Recordset deklarieren
    Dim rcsKnd As Recordset

    ' Verbindung zur Datenbank vorbereiten
    Dim db As Database
    Set db = CurrentDb

    ' Recordset initialisieren und dadurch füllen
    Set rcsKnd = db.OpenRecordset("tblKunden")

    ' per Zeiger (steht auf erstem Element) auf
    ' Felder zugreifen und ausgeben
    Debug.Print rcsKnd.Fields("kndName")
    Debug.Print rcsKnd.Fields("kndVorname")

End Sub
```



	kndIdPk	kndName	kndVorname
+	1	Meier	Thomas
+	2	Müller	Sven
+	3	Meier	Mike
+	4	Müller	Silke

Mit Recordsets Daten lesen

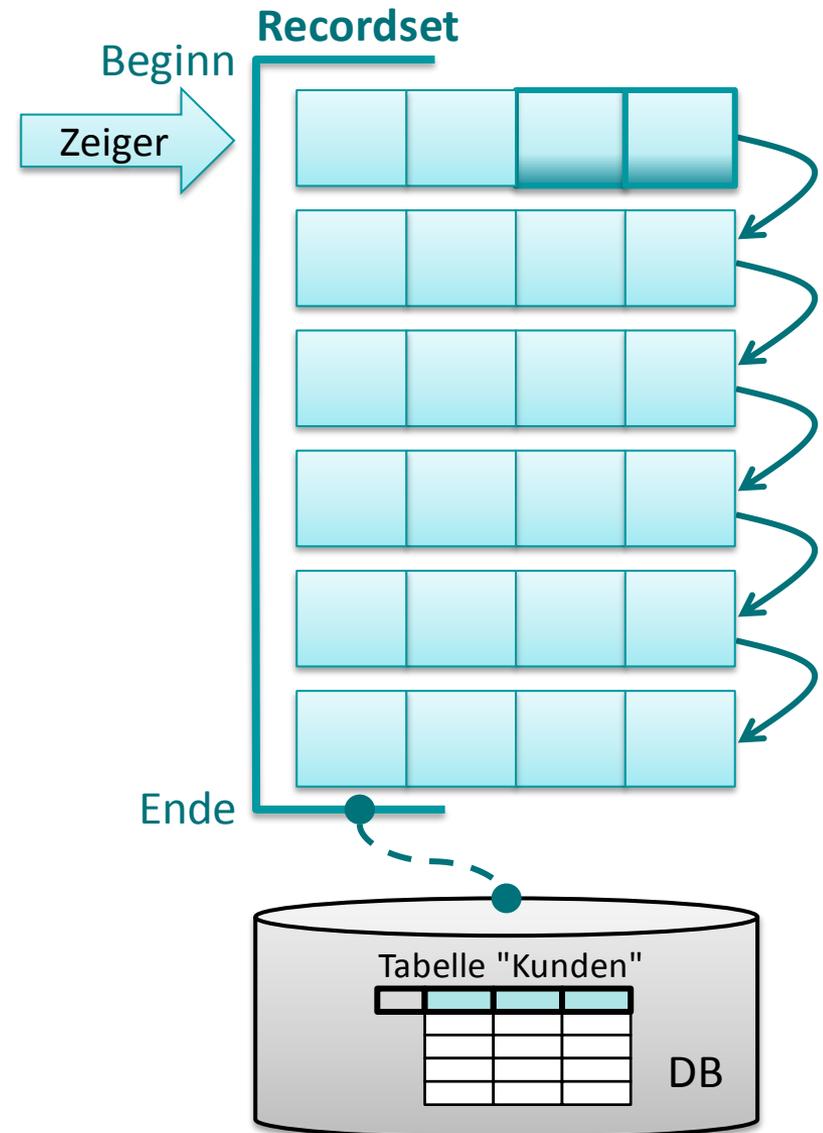


Vorgehensweise

- Recordset deklarieren
- Verbindung zur Datenbank vorbereiten
- Recordset initialisieren und dadurch füllen
- per Zeiger (steht auf erstem Element) auf Felder zugreifen

Zugriff auf Felder im Recordset

`<einRecordset>.Fields(<name>)`





Inhalt

Ziel und Einordnung

Rückblick

Ausgangssituation

Datenbankschnittstelle

Recordsets

- Was sind Recordsets
- Mit Recordsets Daten lesen
- Navigation im Recordset per Zeiger
- Suchen und Finden im Recordset
- Ändern und Hinzufügen im Recordset
- Kombination mit grafischen Abfragen und SQL

Weitere VBA-Funktionen

Fallstricke

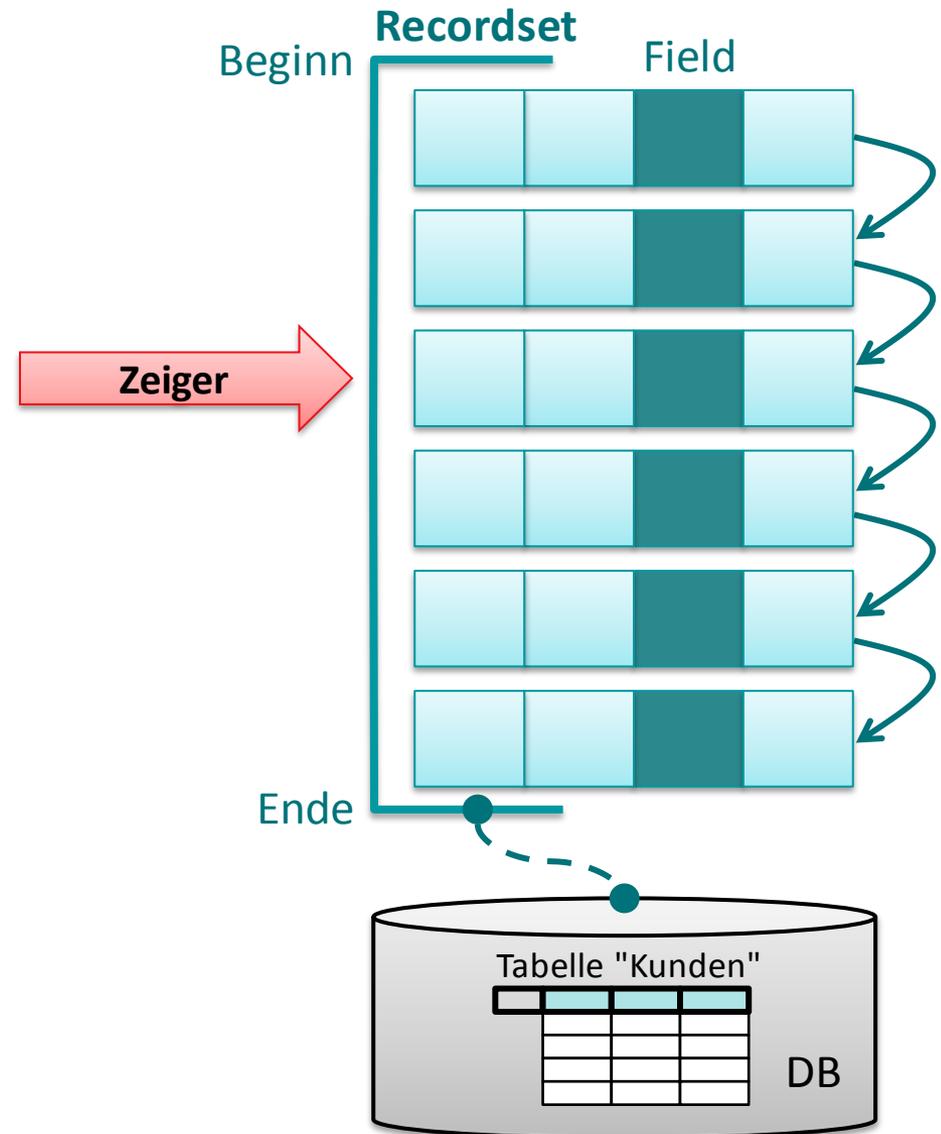
Zusammenfassung und Ausblick



Navigation im Recordset per Zeiger



Navigationsmöglichkeiten

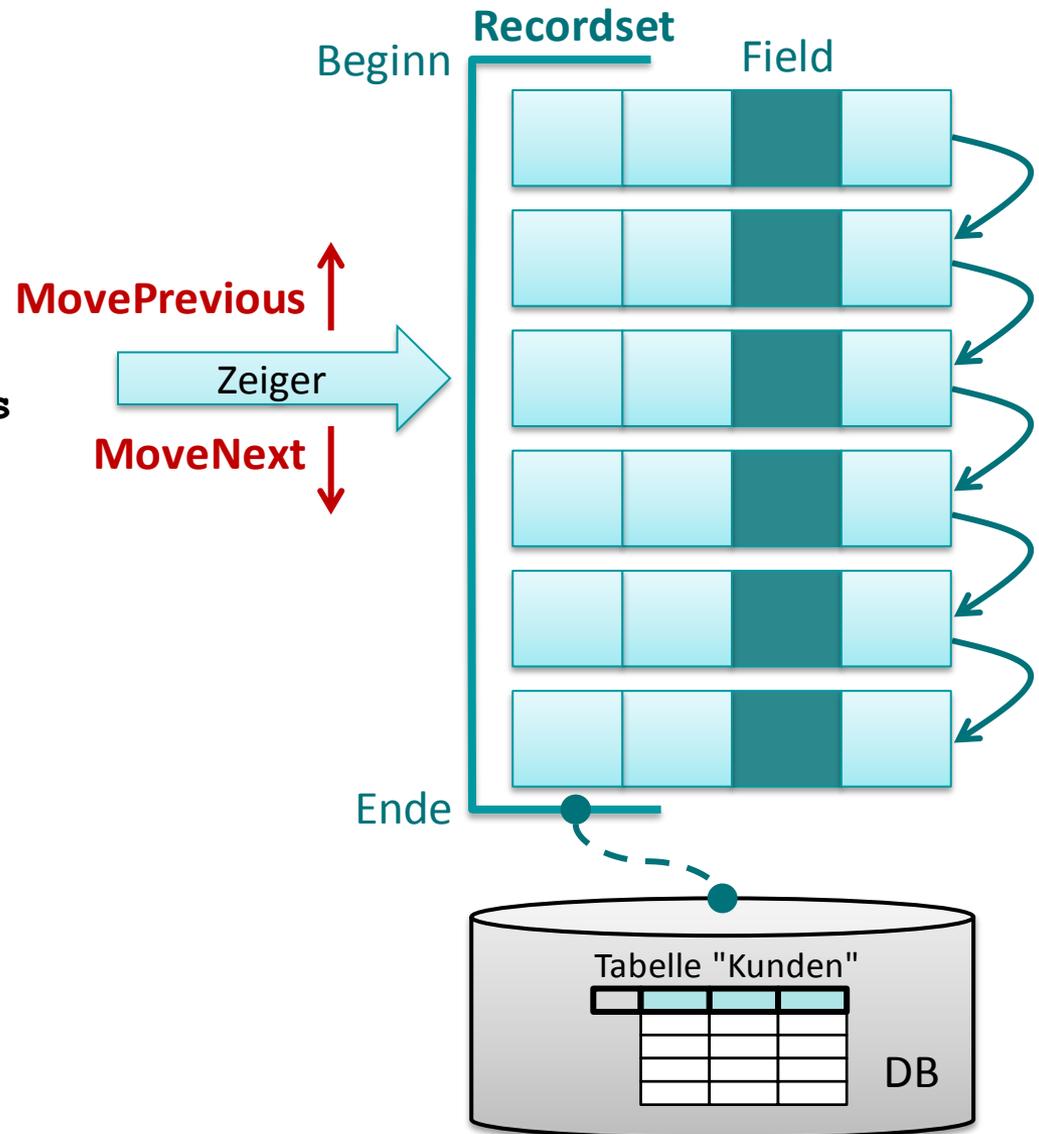


Navigation im Recordset per Zeiger



Navigationsmöglichkeiten

- zum nächsten Datensatz verschieben
`<einRecordset>.MoveNext`
- zum vorherigen Datensatz verschieben
`<einRcrdst>.MovePrevious`

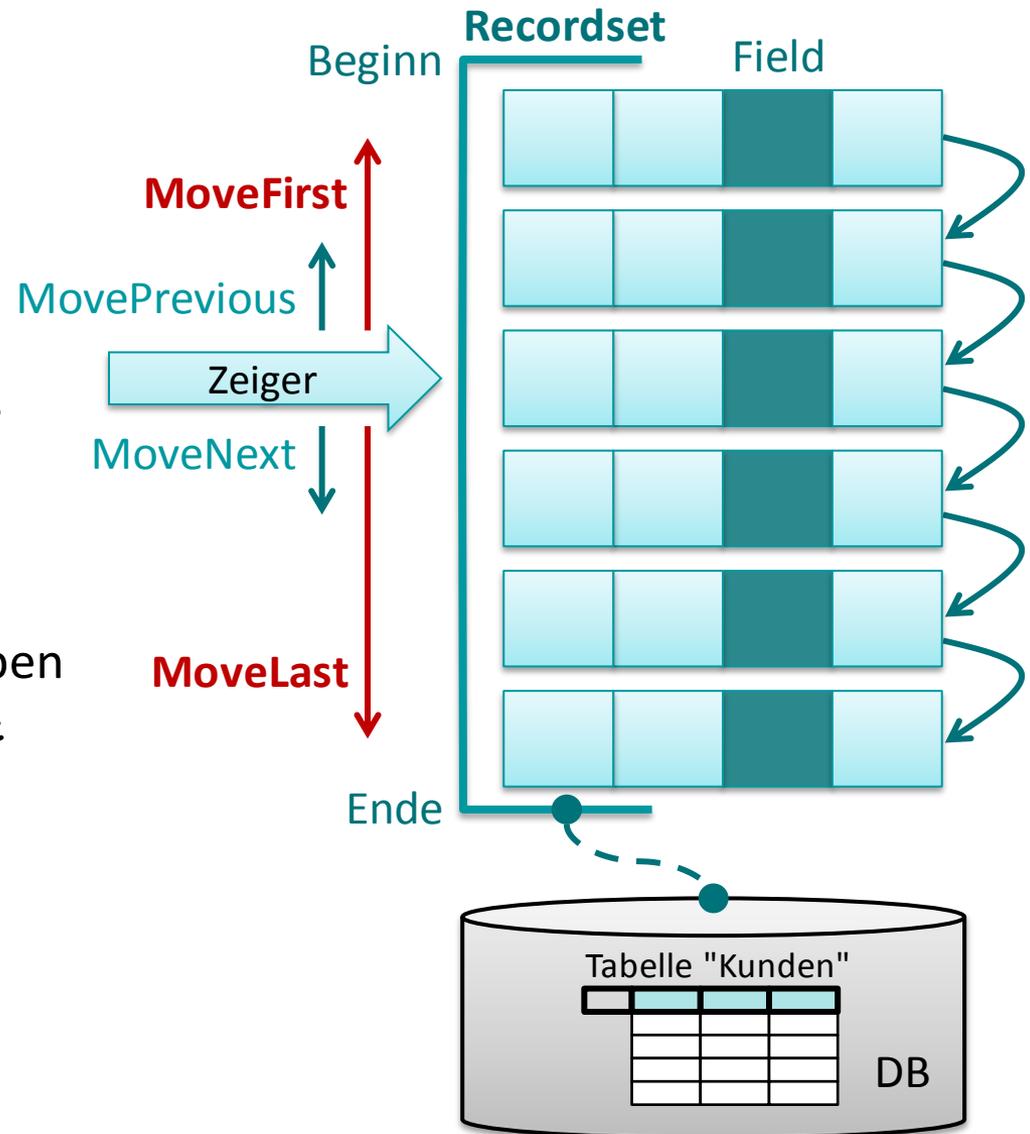


Navigation im Recordset per Zeiger



Navigationsmöglichkeiten

- zum nächsten Datensatz verschieben
`<einRecordset>.MoveNext`
- zum vorherigen Datensatz verschieben
`<einRcrdst>.MovePrevious`
- zum letzten Datensatz verschieben
`<einRecordset>.MoveLast`
- zum ersten Datensatz verschieben
`<einRecordset>.MoveFirst`





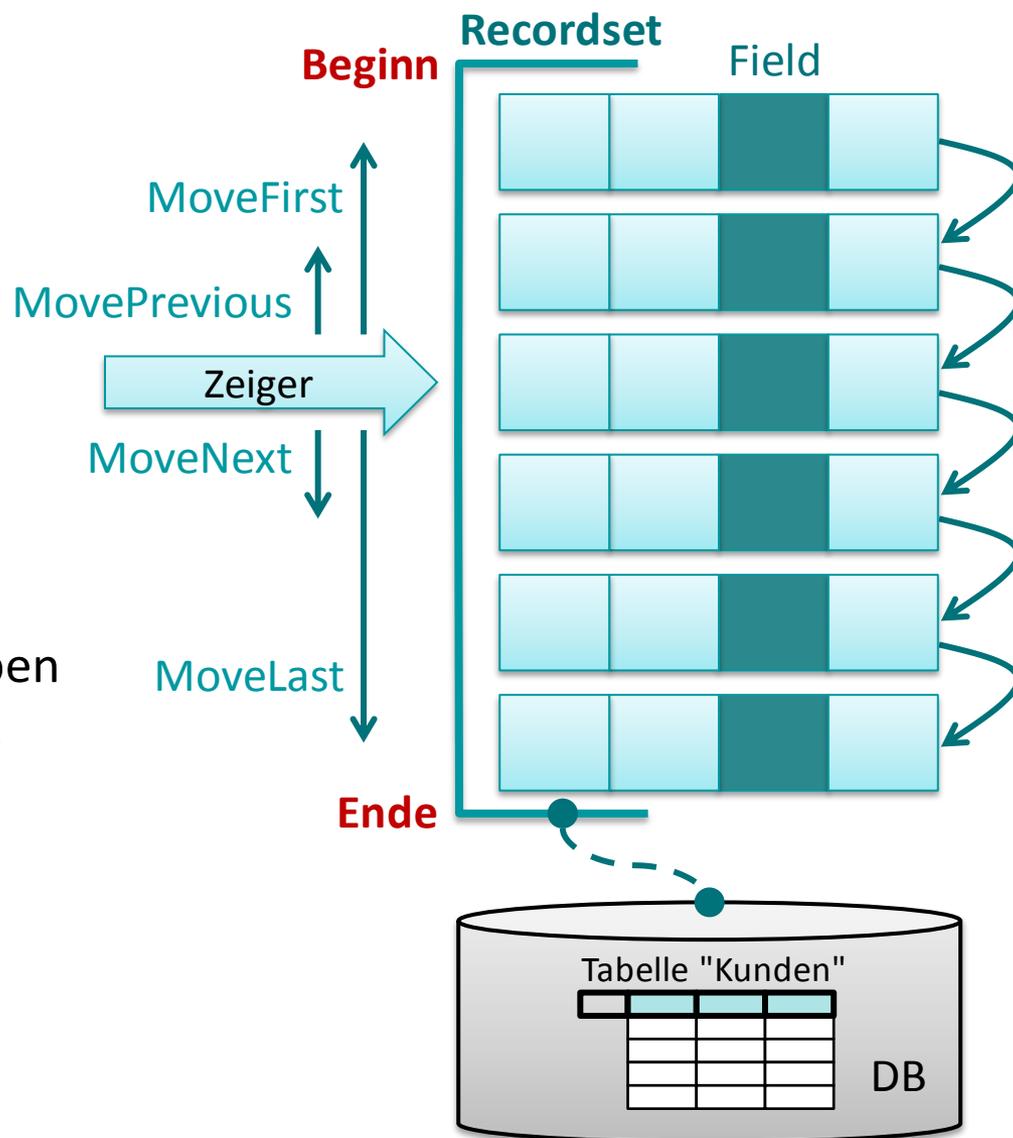
Navigation im Recordset per Zeiger

Navigationsmöglichkeiten

- zum nächsten Datensatz verschieben
`<einRecordset>.MoveNext`
- zum vorherigen Datensatz verschieben
`<einRcrdst>.MovePrevious`
- zum letzten Datensatz verschieben
`<einRecordset>.MoveLast`
- zum ersten Datensatz verschieben
`<einRecordset>.MoveFirst`

Positionsbestimmung, ob Zeiger am Beginn oder am Ende steht (End of File)

- `<einRecordset>.BOF`
- `<einRecordset>.EOF`



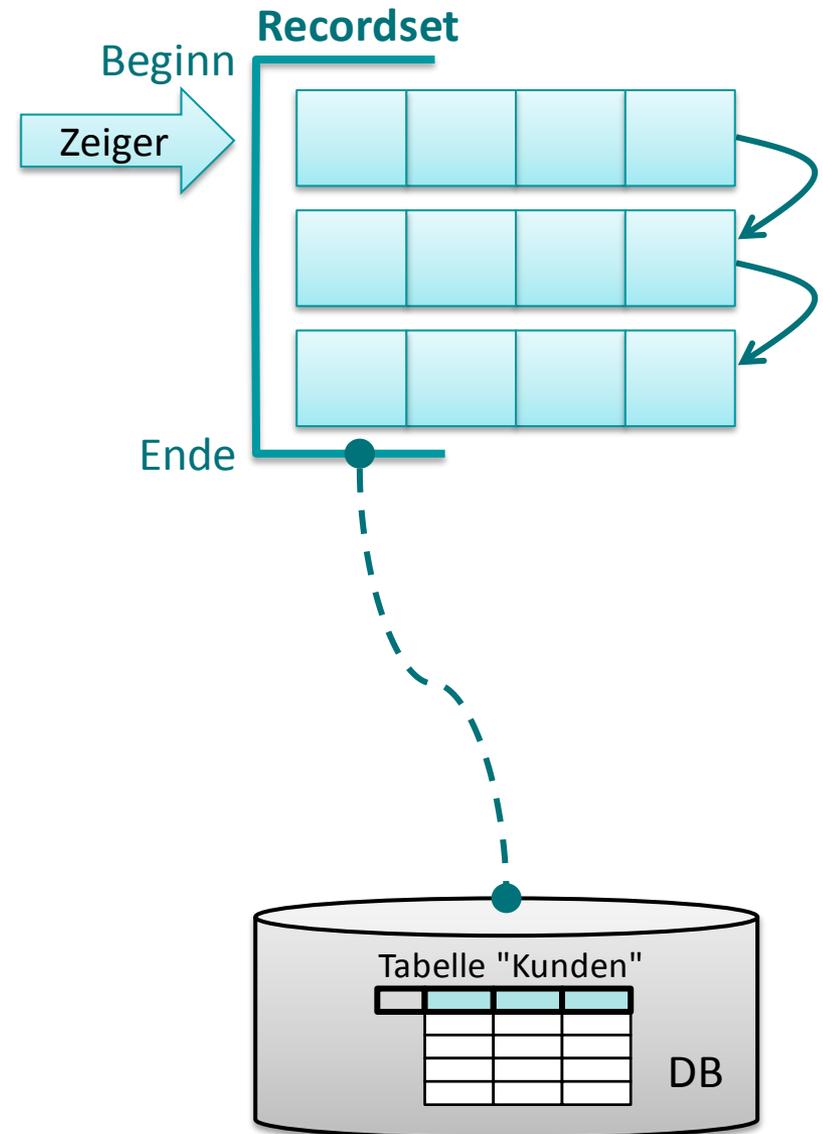
Navigation im Recordset per Zeiger



Ziel: Recordset aus Tabelle "tblKunden" laden und Name, Vorname aller Kunden im Direktbereich ausgeben

Ansatz

- Recordset befüllen und Zeiger steht auf erstem Datensatz oder wenn leer, dann auf Ende



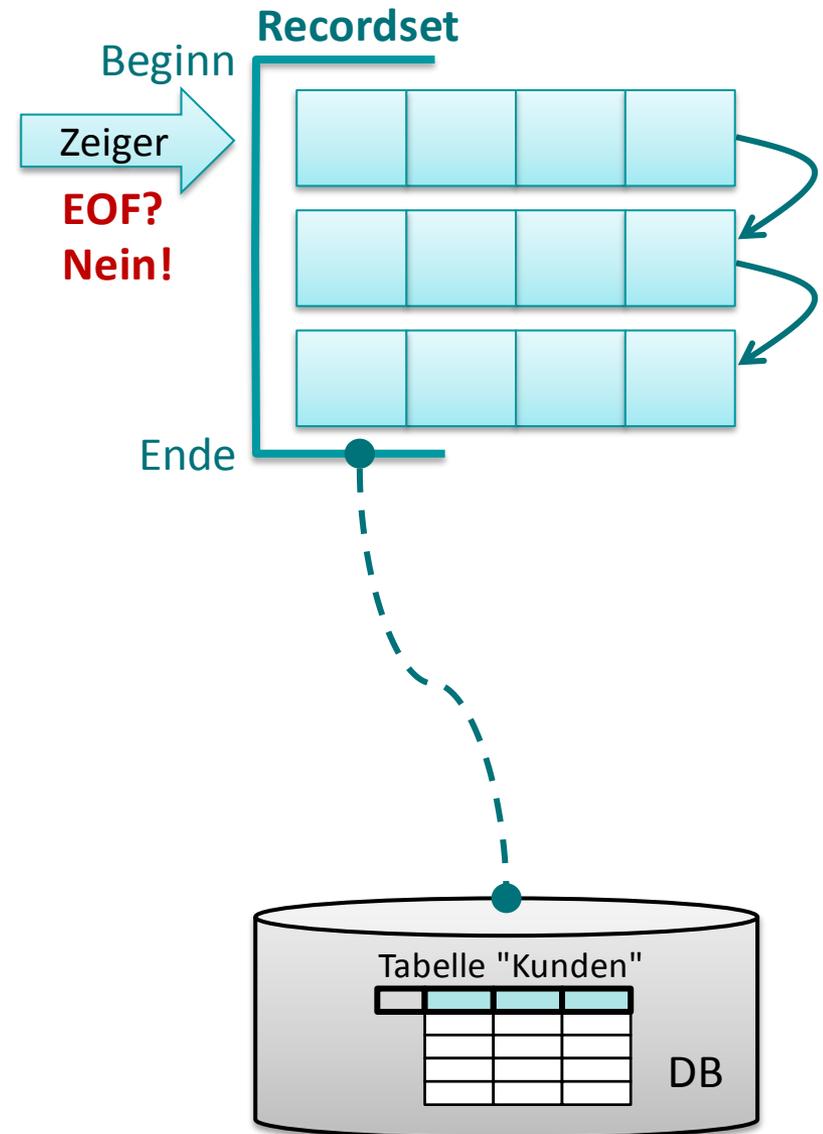


Navigation im Recordset per Zeiger

Ziel: Recordset aus Tabelle "tblKunden" laden und Name, Vorname aller Kunden im Direktbereich ausgeben

Ansatz

- Recordset befüllen und Zeiger steht auf erstem Datensatz oder wenn leer, dann auf Ende
- **Prüfen, ob Ende des Recordsets erreicht**



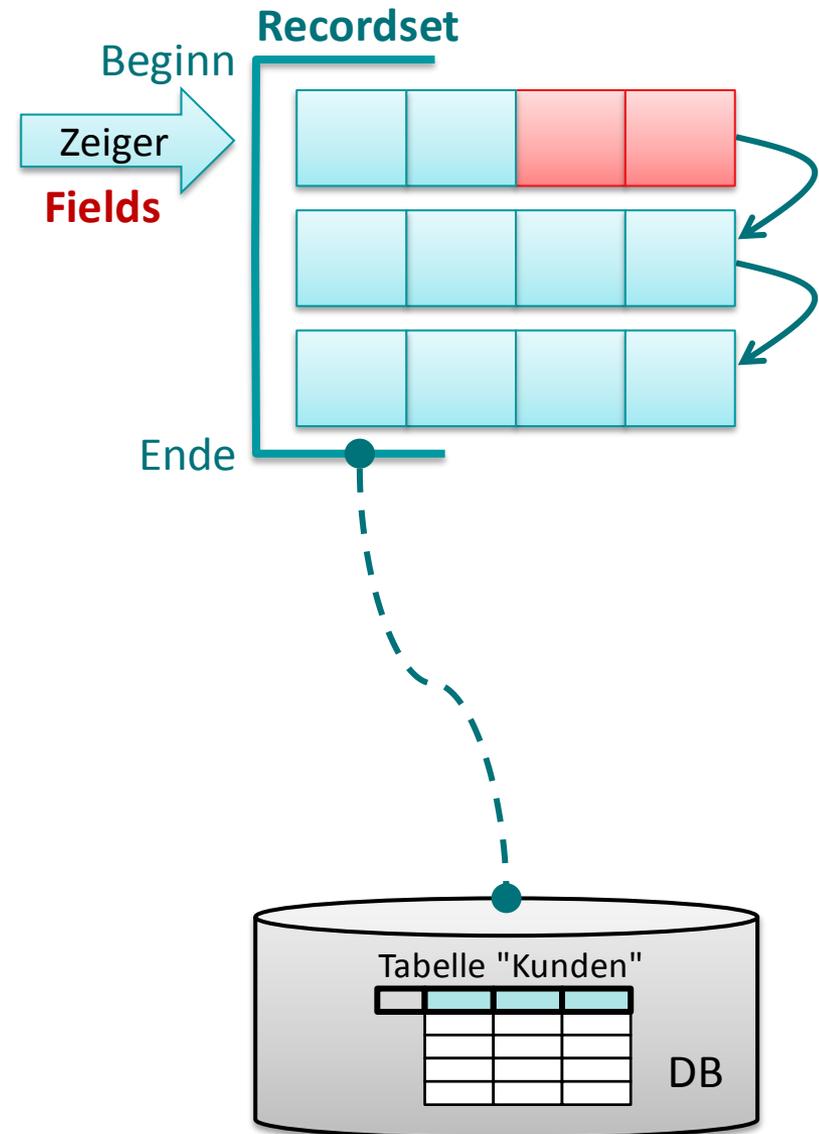


Navigation im Recordset per Zeiger

Ziel: Recordset aus Tabelle "tblKunden" laden und Name, Vorname aller Kunden im Direktbereich ausgeben

Ansatz

- Recordset befüllen und Zeiger steht auf erstem Datensatz oder wenn leer, dann auf Ende
- Prüfen, ob Ende des Recordsets erreicht
- **Wenn nicht Ende, dann Felder des Datensatzes ausgeben**



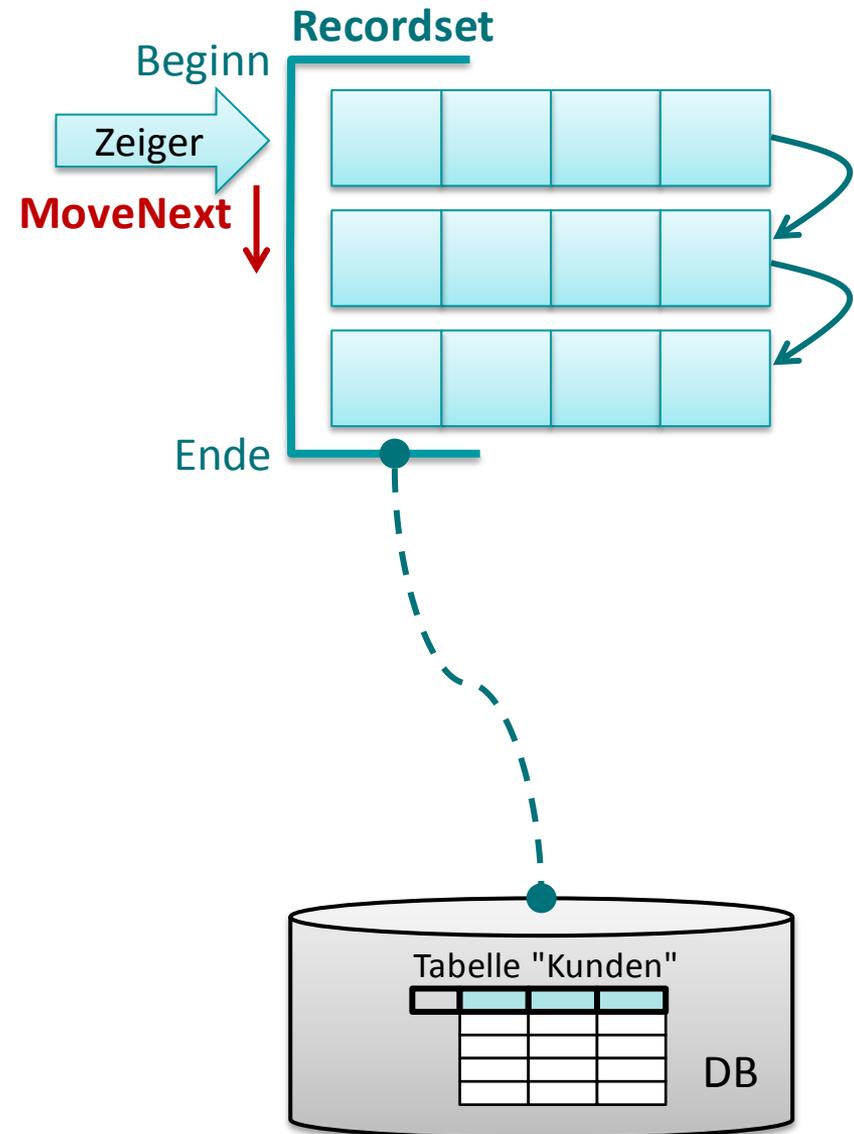


Navigation im Recordset per Zeiger

Ziel: Recordset aus Tabelle "tblKunden" laden und Name, Vorname aller Kunden im Direktbereich ausgeben

Ansatz

- Recordset befüllen und Zeiger steht auf erstem Datensatz oder wenn leer, dann auf Ende
- Prüfen, ob Ende des Recordsets erreicht
- Wenn nicht Ende, dann Felder des Datensatzes ausgeben
- **Zeiger zum nächsten Element verschieben**



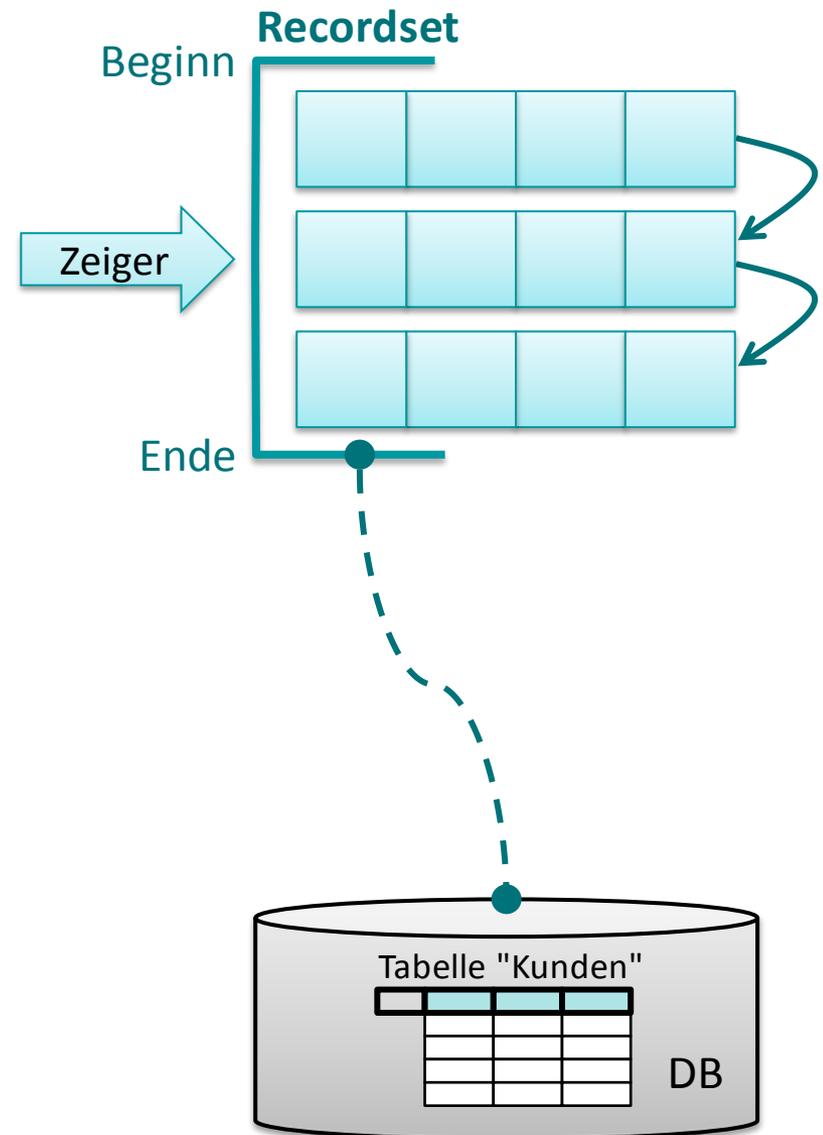


Navigation im Recordset per Zeiger

Ziel: Recordset aus Tabelle "tblKunden" laden und Name, Vorname aller Kunden im Direktbereich ausgeben

Ansatz

- Recordset befüllen und Zeiger steht auf erstem Datensatz oder wenn leer, dann auf Ende
- Prüfen, ob Ende des Recordsets erreicht
- Wenn nicht Ende, dann Felder des Datensatzes ausgeben
- Zeiger zum nächsten Element verschieben
- **Solange fortfahren, bis Ende erreicht**



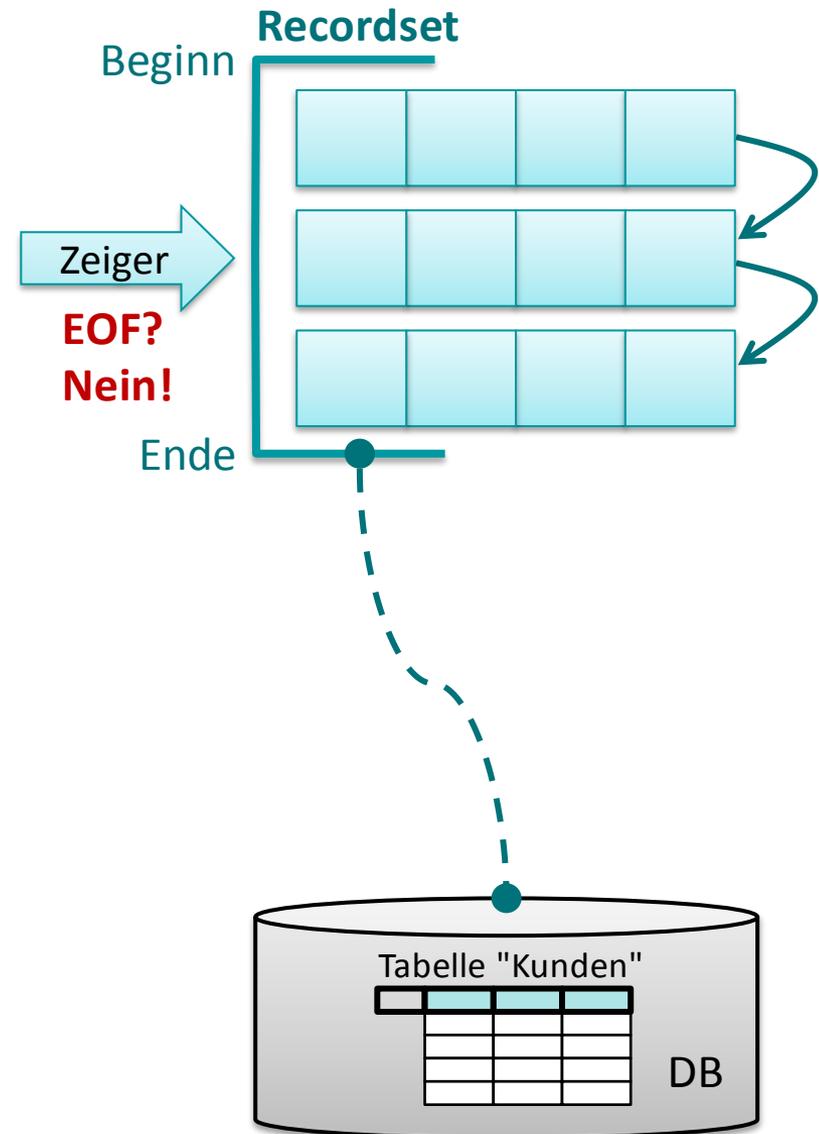


Navigation im Recordset per Zeiger

Ziel: Recordset aus Tabelle "tblKunden" laden und Name, Vorname aller Kunden im Direktbereich ausgeben

Ansatz

- Recordset befüllen und Zeiger steht auf erstem Datensatz oder wenn leer, dann auf Ende
- **Prüfen, ob Ende des Recordsets erreicht**
- Wenn nicht Ende, dann Felder des Datensatzes ausgeben
- Zeiger zum nächsten Element verschieben
- Solange fortfahren, bis Ende erreicht



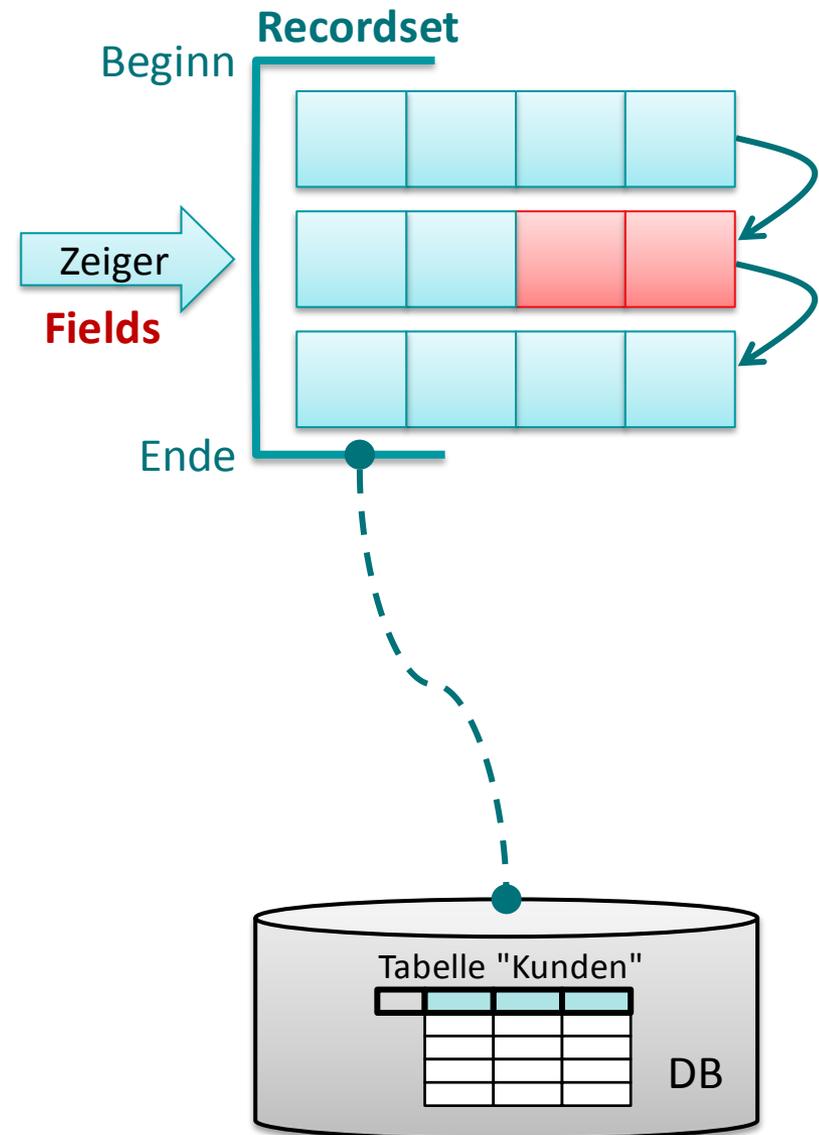


Navigation im Recordset per Zeiger

Ziel: Recordset aus Tabelle "tblKunden" laden und Name, Vorname aller Kunden im Direktbereich ausgeben

Ansatz

- Recordset befüllen und Zeiger steht auf erstem Datensatz oder wenn leer, dann auf Ende
- Prüfen, ob Ende des Recordsets erreicht
- **Wenn nicht Ende, dann Felder des Datensatzes ausgeben**
- Zeiger zum nächsten Element verschieben
- Solange fortfahren, bis Ende erreicht



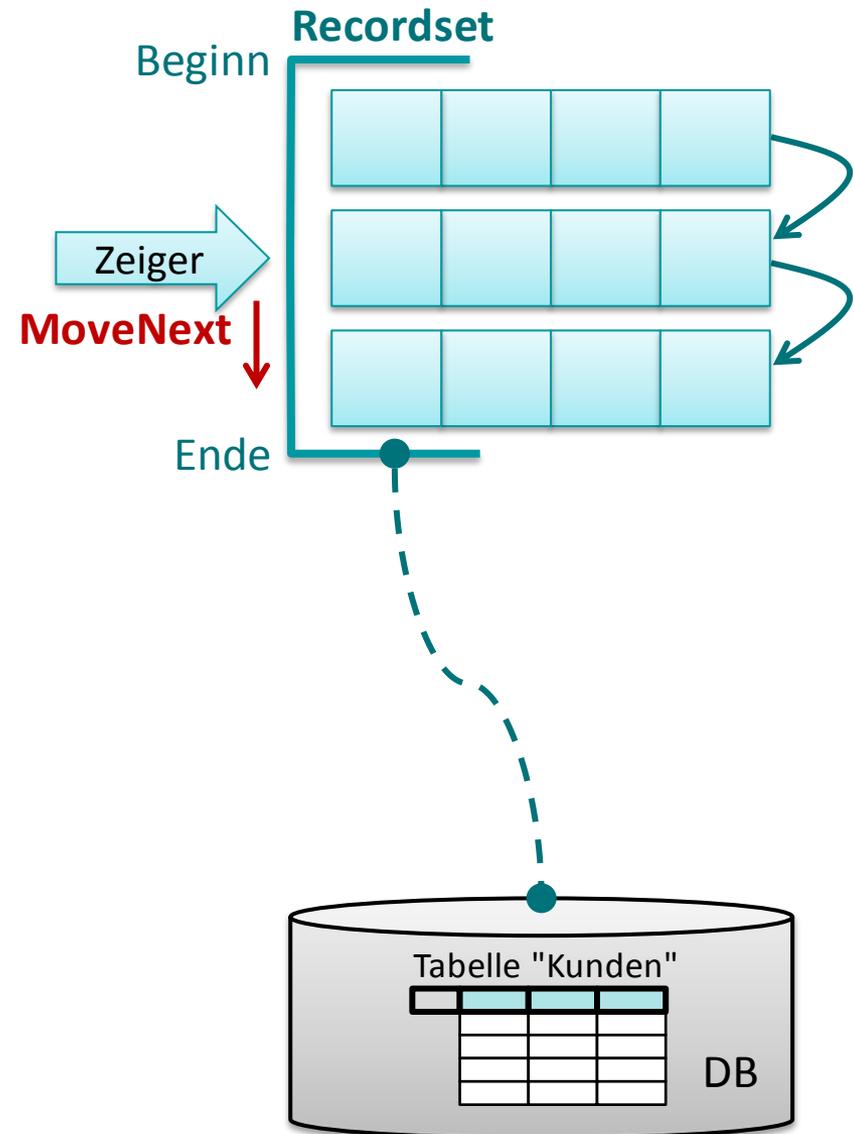


Navigation im Recordset per Zeiger

Ziel: Recordset aus Tabelle "tblKunden" laden und Name, Vorname aller Kunden im Direktbereich ausgeben

Ansatz

- Recordset befüllen und Zeiger steht auf erstem Datensatz oder wenn leer, dann auf Ende
- Prüfen, ob Ende des Recordsets erreicht
- Wenn nicht Ende, dann Felder des Datensatzes ausgeben
- **Zeiger zum nächsten Element verschieben**
- Solange fortfahren, bis Ende erreicht



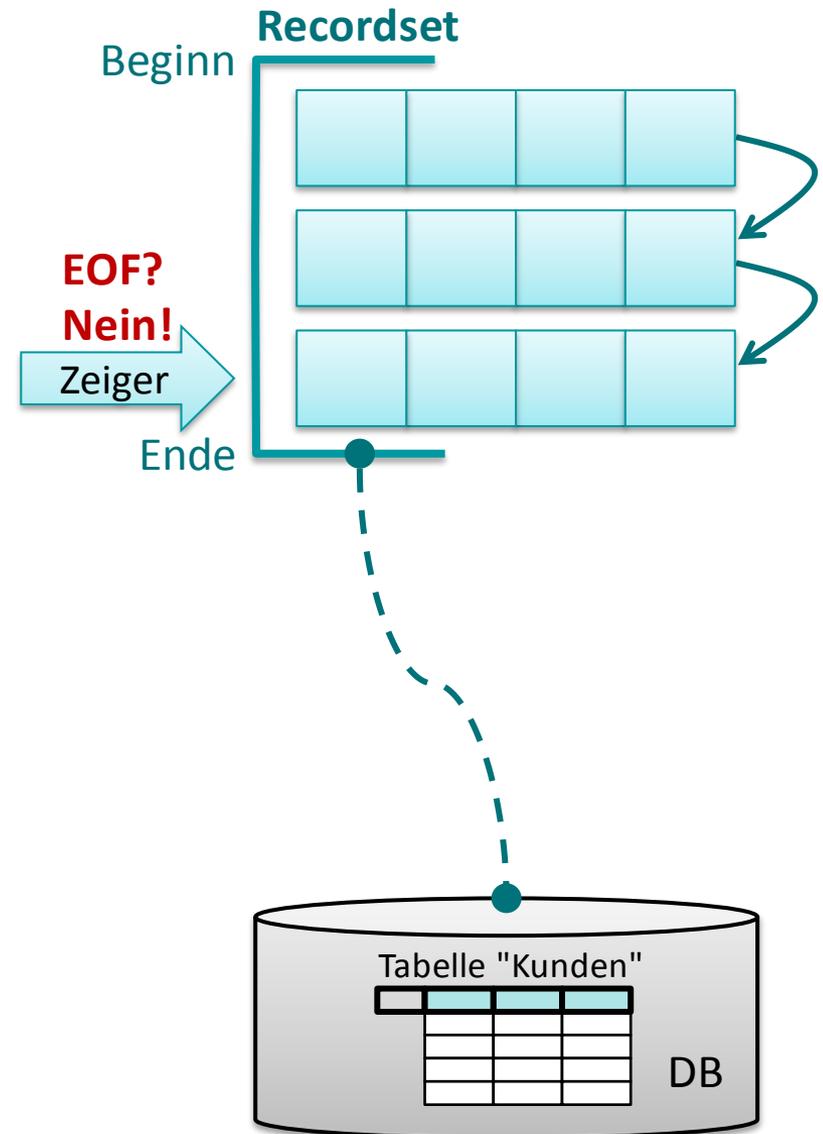


Navigation im Recordset per Zeiger

Ziel: Recordset aus Tabelle "tblKunden" laden und Name, Vorname aller Kunden im Direktbereich ausgeben

Ansatz

- Recordset befüllen und Zeiger steht auf erstem Datensatz oder wenn leer, dann auf Ende
- **Prüfen, ob Ende des Recordsets erreicht**
- Wenn nicht Ende, dann Felder des Datensatzes ausgeben
- Zeiger zum nächsten Element verschieben
- Solange fortfahren, bis Ende erreicht



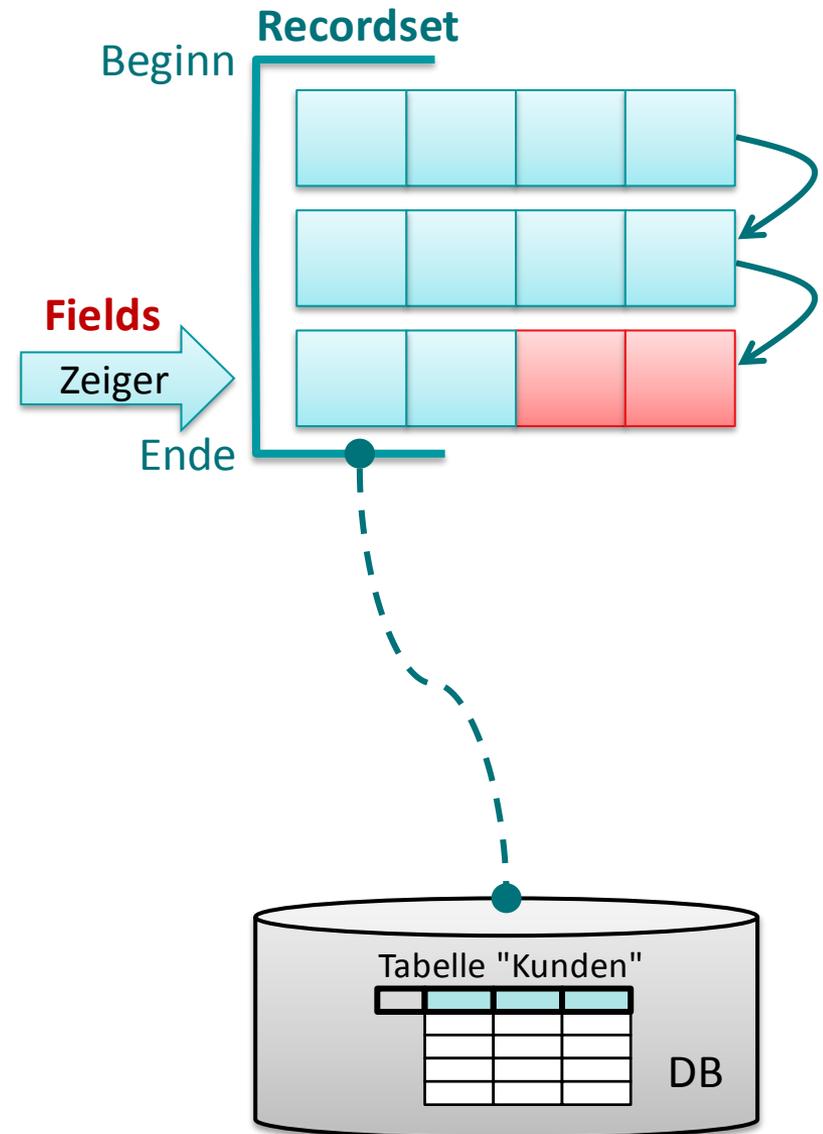


Navigation im Recordset per Zeiger

Ziel: Recordset aus Tabelle "tblKunden" laden und Name, Vorname aller Kunden im Direktbereich ausgeben

Ansatz

- Recordset befüllen und Zeiger steht auf erstem Datensatz oder wenn leer, dann auf Ende
- Prüfen, ob Ende des Recordsets erreicht
- **Wenn nicht Ende, dann Felder des Datensatzes ausgeben**
- Zeiger zum nächsten Element verschieben
- Solange fortfahren, bis Ende erreicht



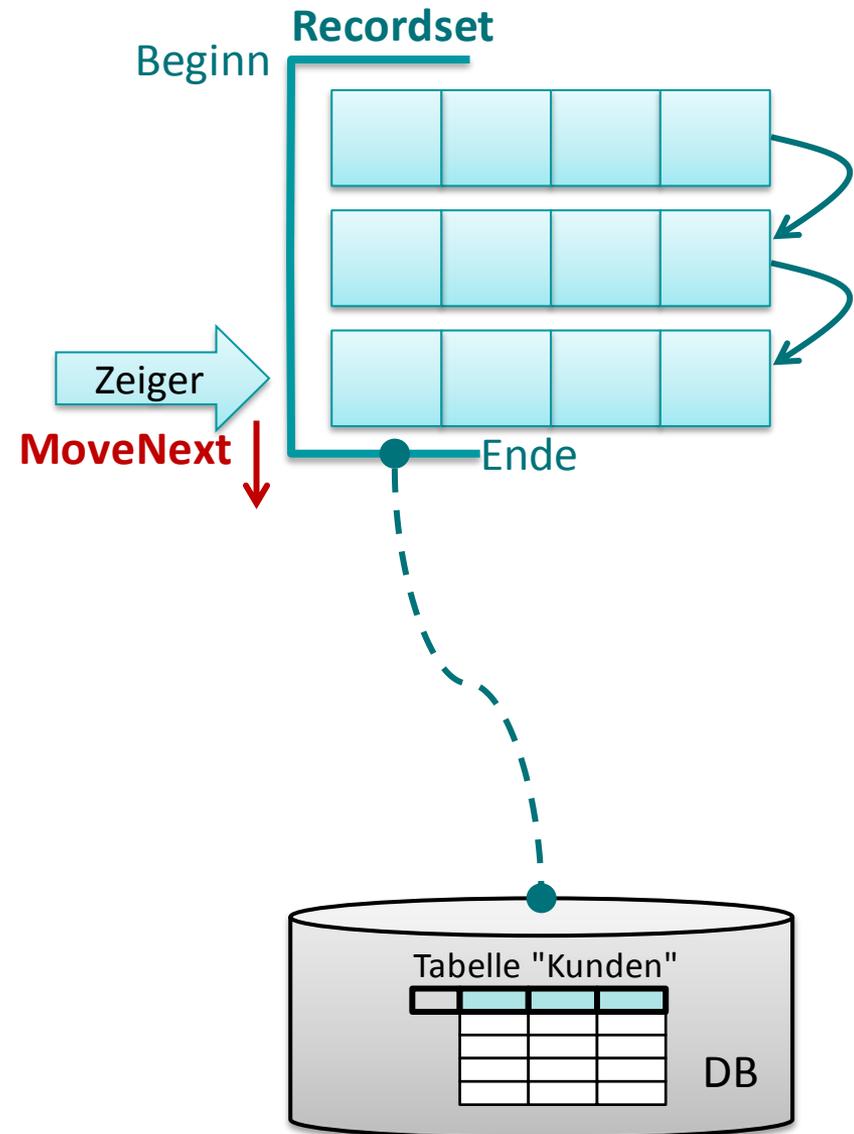


Navigation im Recordset per Zeiger

Ziel: Recordset aus Tabelle "tblKunden" laden und Name, Vorname aller Kunden im Direktbereich ausgeben

Ansatz

- Recordset befüllen und Zeiger steht auf erstem Datensatz oder wenn leer, dann auf Ende
- Prüfen, ob Ende des Recordsets erreicht
- Wenn nicht Ende, dann Felder des Datensatzes ausgeben
- **Zeiger zum nächsten Element verschieben**
- Solange fortfahren, bis Ende erreicht



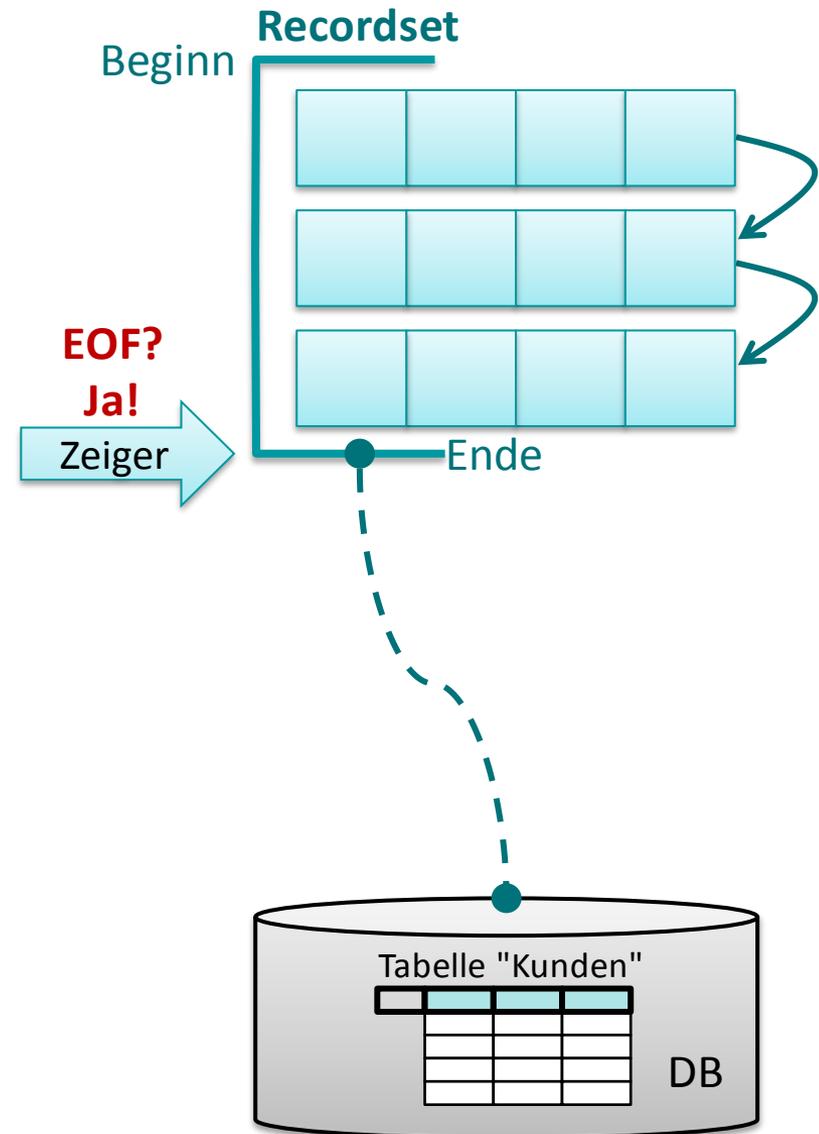


Navigation im Recordset per Zeiger

Ziel: Recordset aus Tabelle "tblKunden" laden und Name, Vorname aller Kunden im Direktbereich ausgeben

Ansatz

- Recordset befüllen und Zeiger steht auf erstem Datensatz oder wenn leer, dann auf Ende
- **Prüfen, ob Ende des Recordsets erreicht**
- Wenn nicht Ende, dann Felder des Datensatzes ausgeben
- Zeiger zum nächsten Element verschieben
- Solange fortfahren, bis Ende erreicht



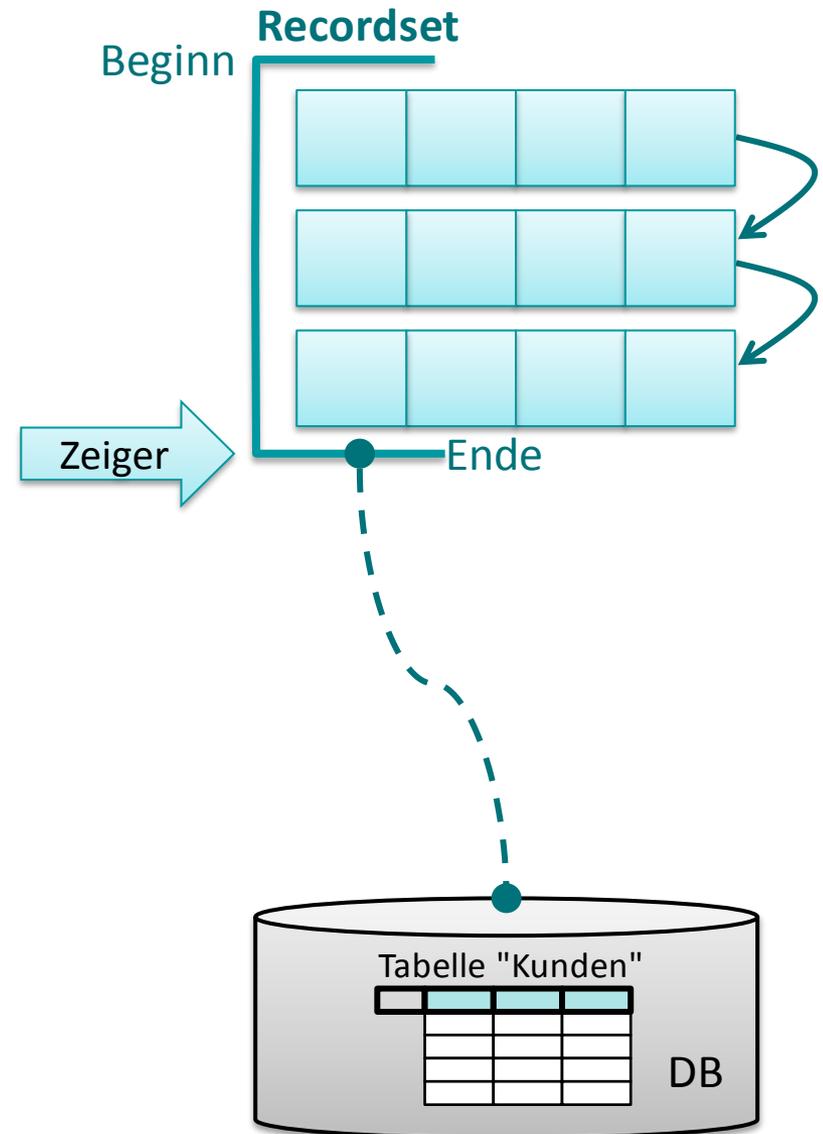
Navigation im Recordset per Zeiger



Ziel: Recordset aus Tabelle "tblKunden" laden und Name, Vorname aller Kunden im Direktbereich ausgeben

Ansatz

- Recordset befüllen und Zeiger steht auf erstem Datensatz oder wenn leer, dann auf Ende
- Prüfen, ob Ende des Recordsets erreicht
- Wenn nicht Ende, dann Felder des Datensatzes ausgeben
- Zeiger zum nächsten Element verschieben
- Solange fortfahren, bis Ende erreicht



Navigation im Recordset per Zeiger

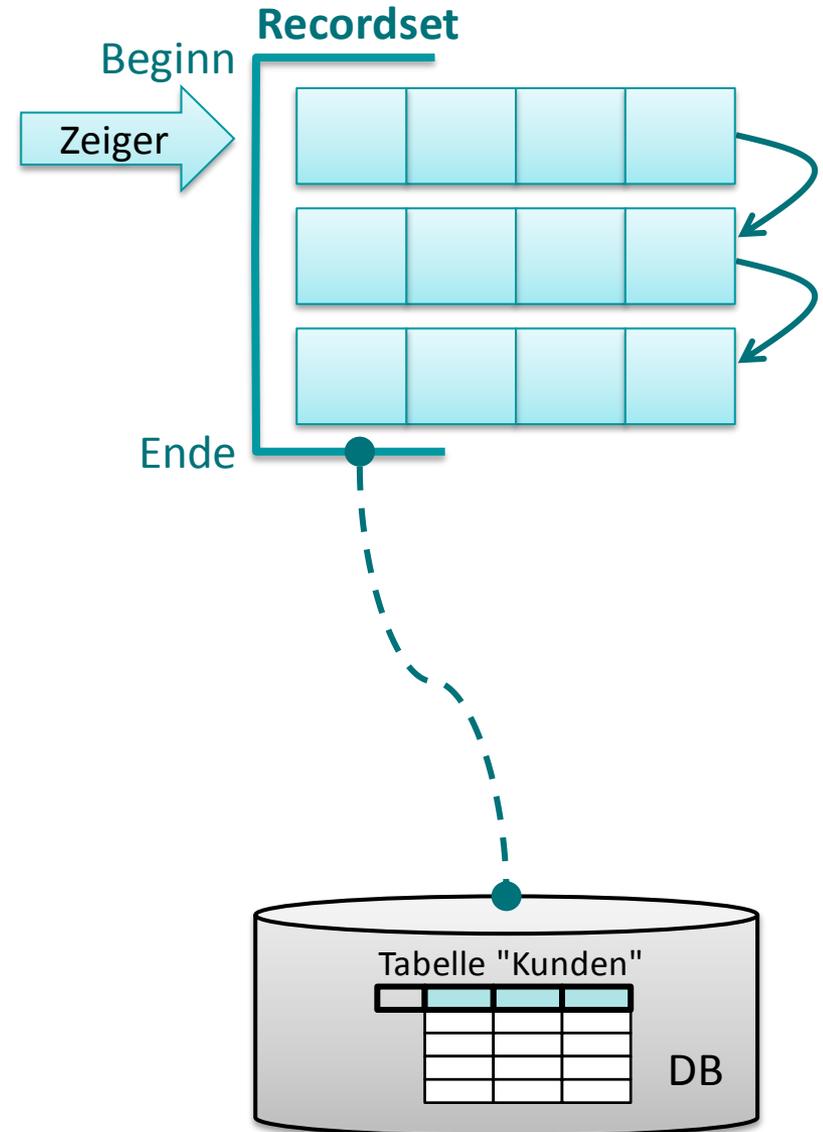


Modul

```
Option Compare Database
Option Explicit

Sub ausgebenAlleKunden()
' Recordset deklarieren
Dim rcsKnd As Recordset
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb
' Recordset initialisieren
Set rcsKnd=db.OpenRecordset("tblKunden")
```

End Sub



Navigation im Recordset per Zeiger



Modul

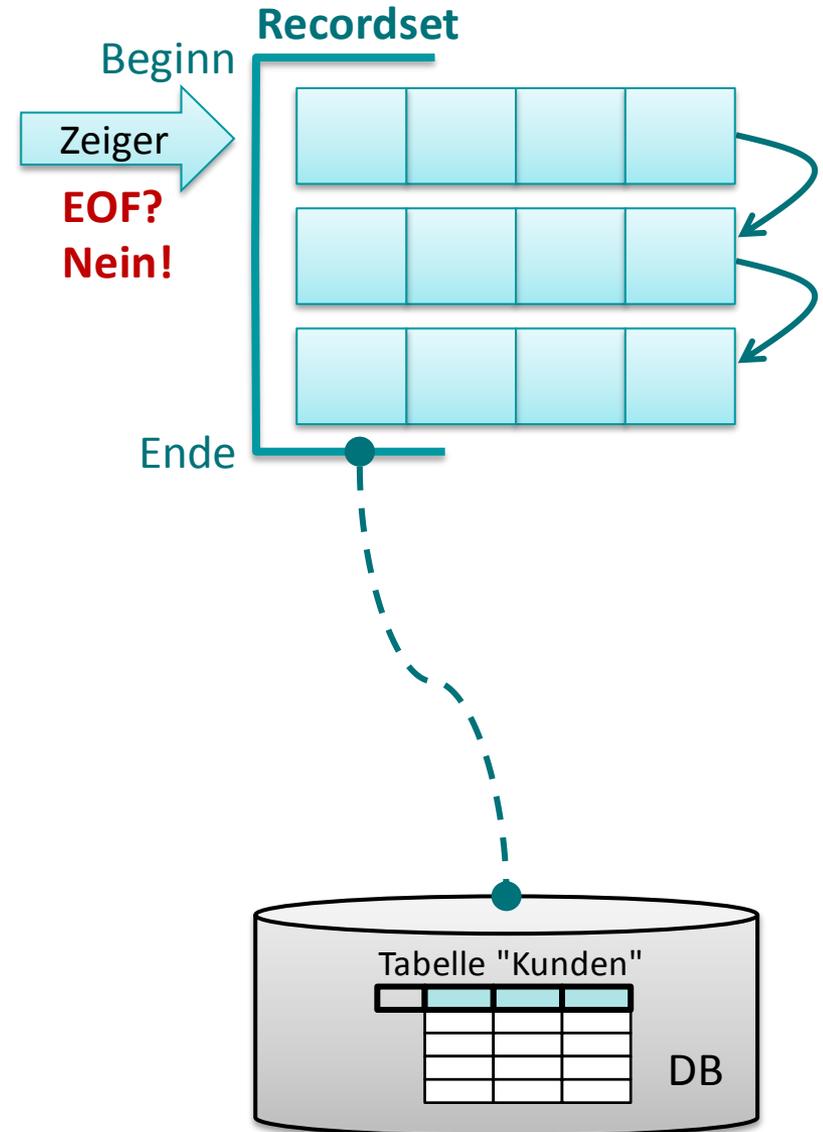
```
Option Compare Database
Option Explicit

Sub ausgebenAlleKunden()
' Recordset deklarieren
Dim rcsKnd As Recordset
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb
' Recordset initialisieren
Set rcsKnd=db.OpenRecordset("tblKunden")

'Schleife bis zum Ende des Recordset
Do Until rcsKnd.EOF

Loop

End Sub
```



Navigation im Recordset per Zeiger



Modul

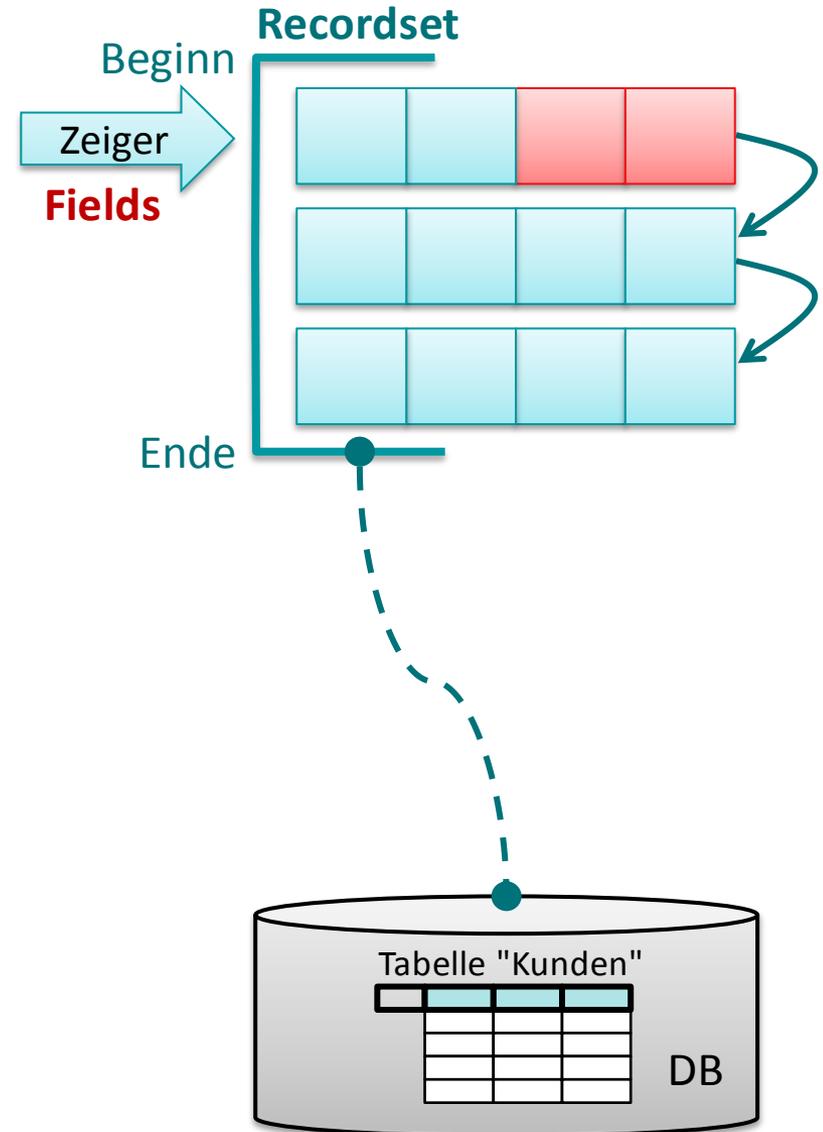
```
Option Compare Database
Option Explicit

Sub ausgebenAlleKunden()
' Recordset deklarieren
Dim rcsKnd As Recordset
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb
' Recordset initialisieren
Set rcsKnd=db.OpenRecordset("tblKunden")

'Schleife bis zum Ende des Recordset
Do Until rcsKnd.EOF
' Name, Vorname des aktuellen
' Elements ausgeben (Zeiger zeigt drauf)
Debug.Print rcsKnd.Fields("kndName") & _
  ", " & rcsKnd.Fields("kndVorname")

Loop

End Sub
```



Navigation im Recordset per Zeiger



Modul

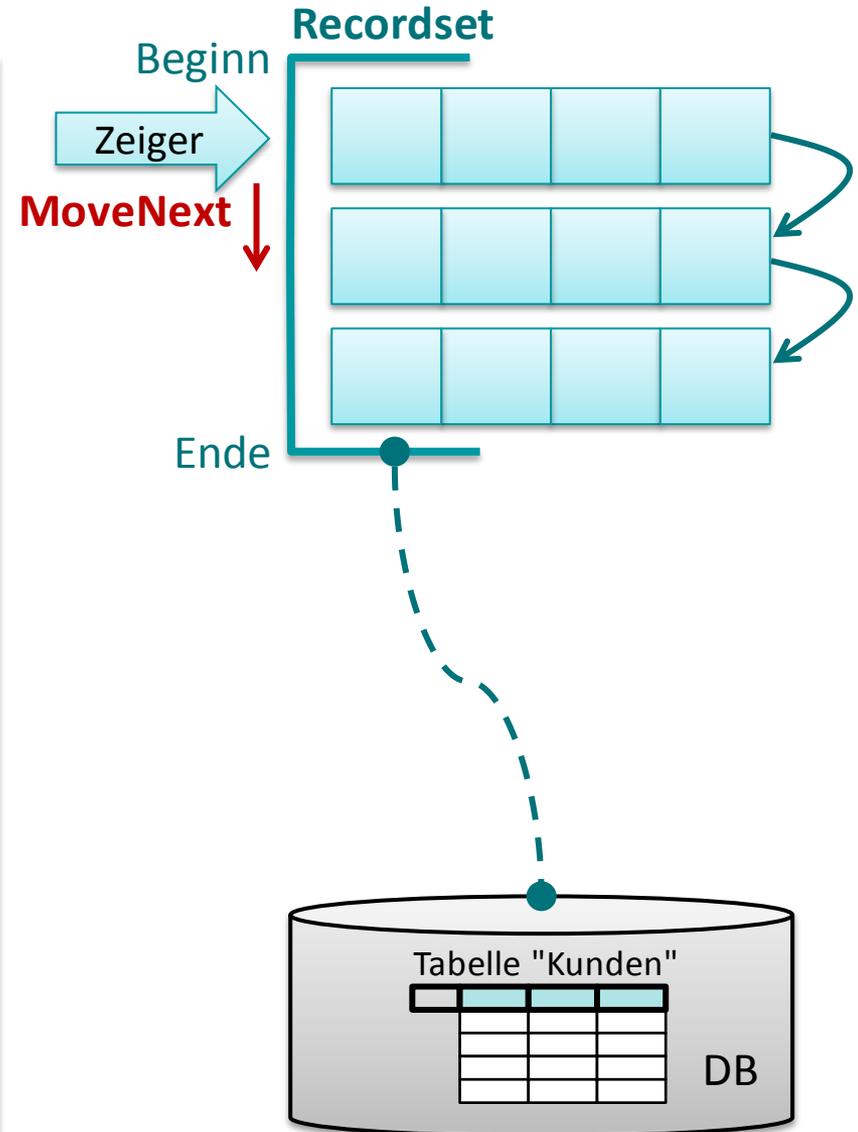
```
Option Compare Database
Option Explicit

Sub ausgebenAlleKunden()
' Recordset deklarieren
Dim rcsKnd As Recordset
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb
' Recordset initialisieren
Set rcsKnd=db.OpenRecordset("tblKunden")

'Schleife bis zum Ende des Recordset
Do Until rcsKnd.EOF
' Name, Vorname des aktuellen
' Elements ausgeben (Zeiger zeigt drauf)
Debug.Print rcsKnd.Fields("kndName") & _
  ", " & rcsKnd.Fields("kndVorname")
' Zeiger weiterrücken
rcsKnd.MoveNext

Loop

End Sub
```



Navigation im Recordset per Zeiger



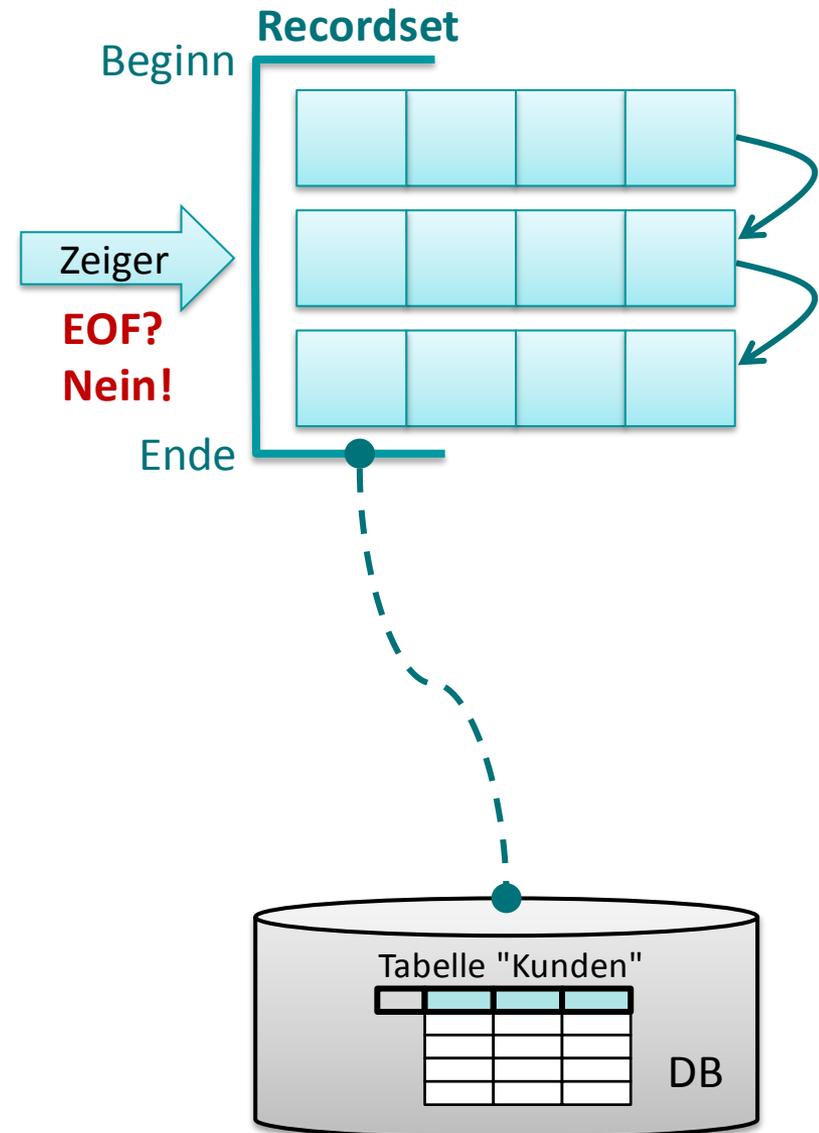
Modul

```
Option Compare Database
Option Explicit

Sub ausgebenAlleKunden()
' Recordset deklarieren
Dim rcsKnd As Recordset
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb
' Recordset initialisieren
Set rcsKnd=db.OpenRecordset("tblKunden")

'Schleife bis zum Ende des Recordset
Do Until rcsKnd.EOF
' Name, Vorname des aktuellen
' Elements ausgeben (Zeiger zeigt drauf)
Debug.Print rcsKnd.Fields("kndName") & _
", " & rcsKnd.Fields("kndVorname")
' Zeiger weiterrücken
rcsKnd.MoveNext
' Solange fortfahren, bis Ende erreicht
Loop

End Sub
```



Navigation im Recordset per Zeiger



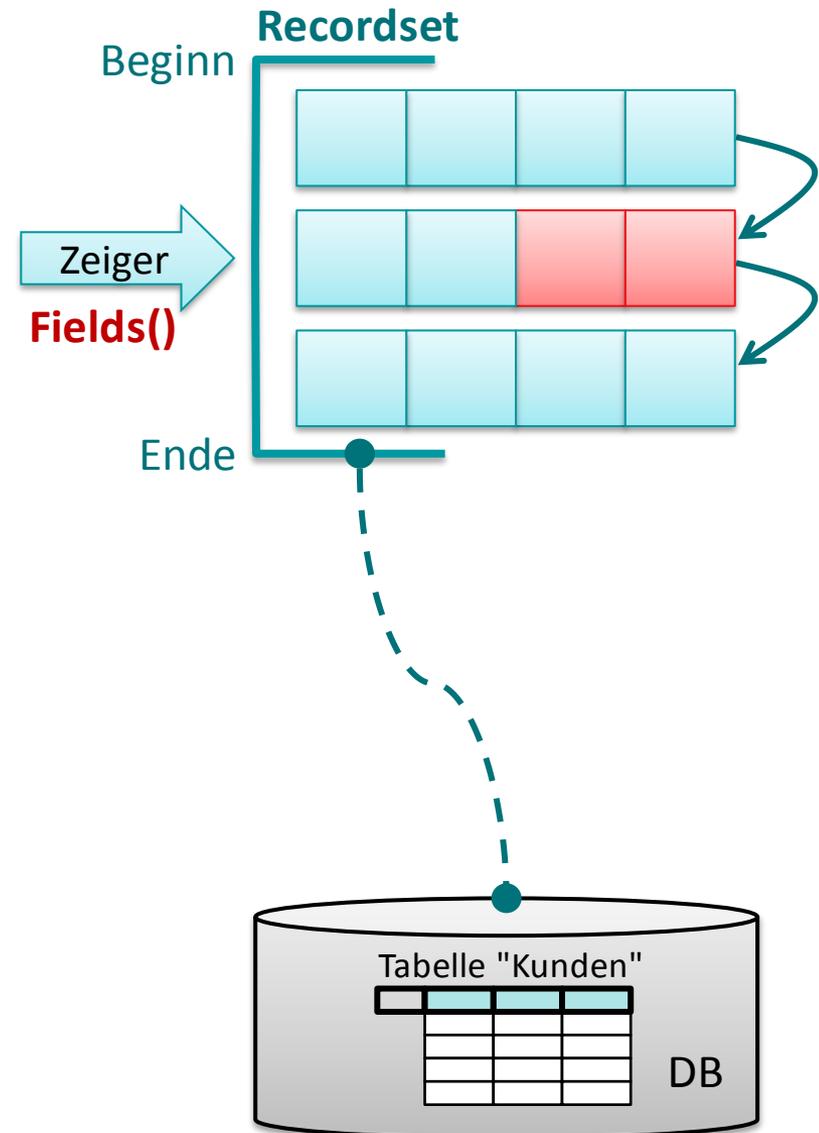
Modul

```
Option Compare Database
Option Explicit

Sub ausgebenAlleKunden()
' Recordset deklarieren
Dim rcsKnd As Recordset
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb
' Recordset initialisieren
Set rcsKnd=db.OpenRecordset("tblKunden")

'Schleife bis zum Ende des Recordset
Do Until rcsKnd.EOF
' Name, Vorname des aktuellen
' Elements ausgeben (Zeiger zeigt drauf)
Debug.Print rcsKnd.Fields("kndName") & _
", " & rcsKnd.Fields("kndVorname")
' Zeiger weiterrücken
rcsKnd.MoveNext
' Solange fortfahren, bis Ende erreicht
Loop

End Sub
```



Navigation im Recordset per Zeiger



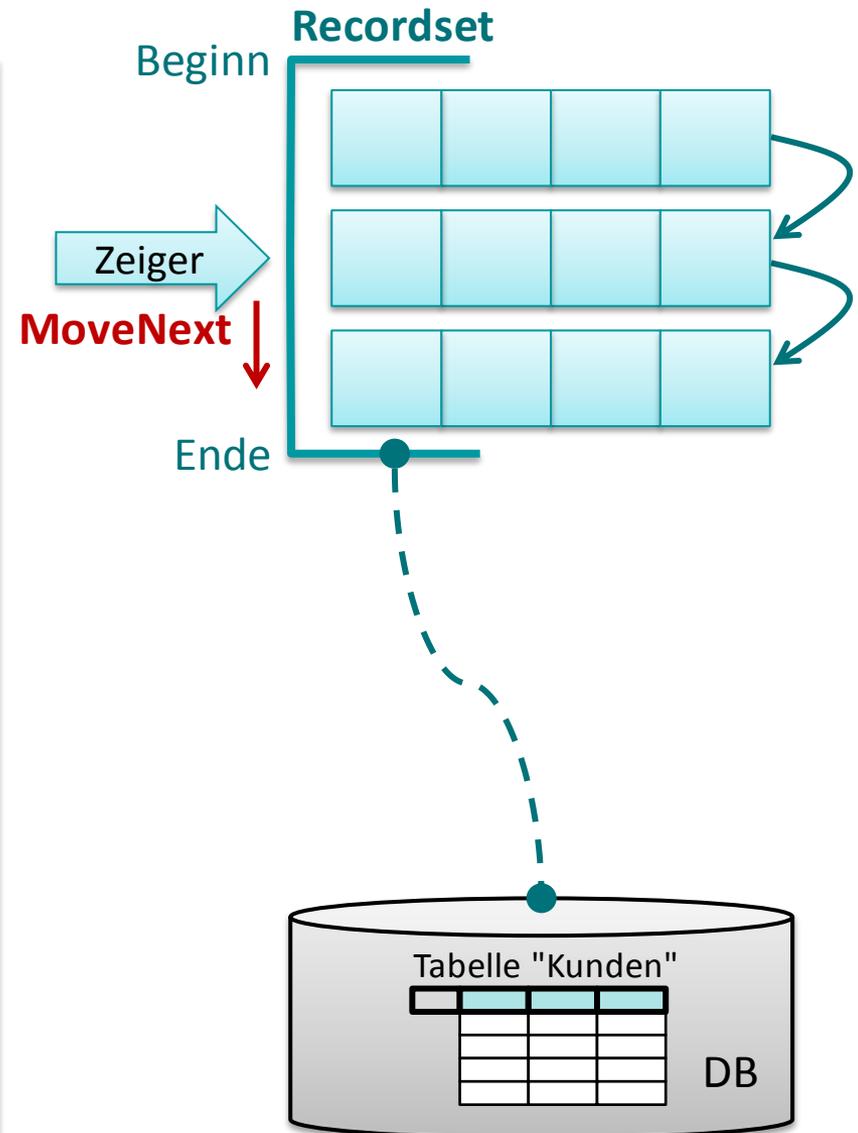
Modul

```
Option Compare Database
Option Explicit

Sub ausgebenAlleKunden()
' Recordset deklarieren
Dim rcsKnd As Recordset
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb
' Recordset initialisieren
Set rcsKnd=db.OpenRecordset("tblKunden")

'Schleife bis zum Ende des Recordset
Do Until rcsKnd.EOF
' Name, Vorname des aktuellen
' Elements ausgeben (Zeiger zeigt drauf)
Debug.Print rcsKnd.Fields("kndName") & _
  ", " & rcsKnd.Fields("kndVorname")
' Zeigeriterrücken
rcsKnd.MoveNext
' Solange fortfahren, bis Ende erreicht
Loop

End Sub
```



Navigation im Recordset per Zeiger



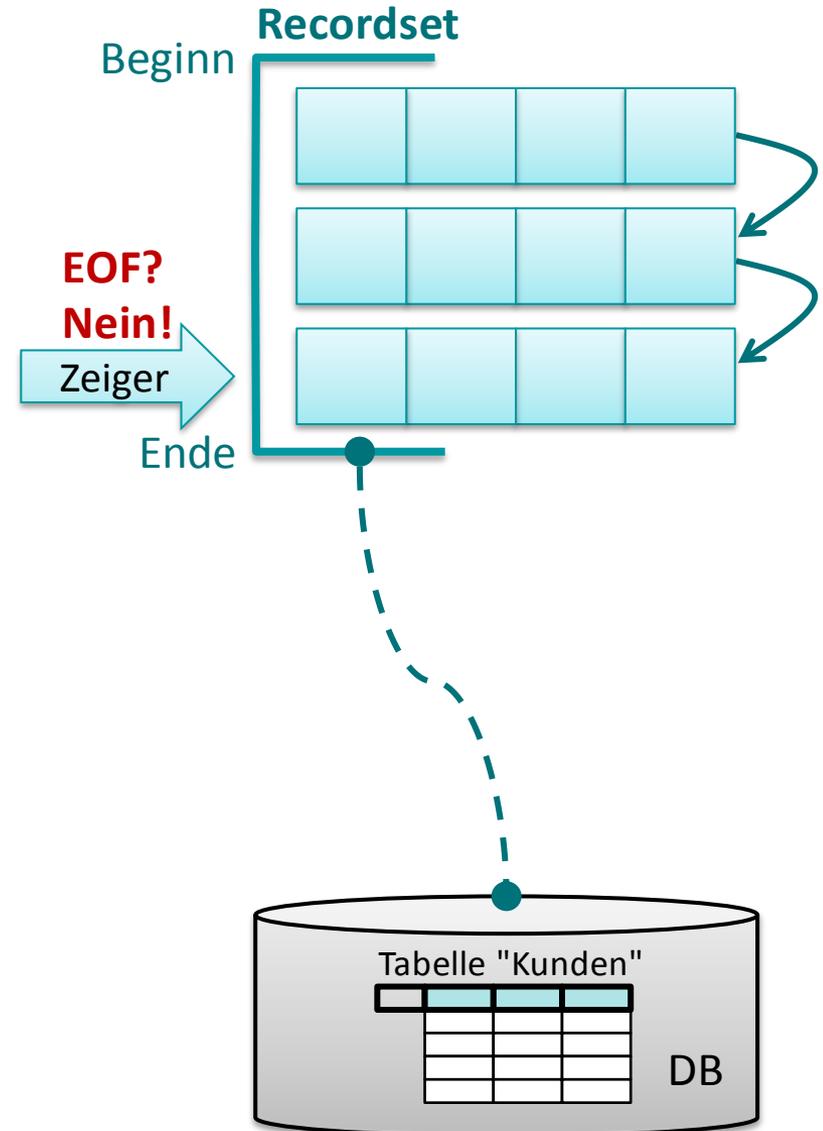
Modul

```
Option Compare Database
Option Explicit

Sub ausgebenAlleKunden()
' Recordset deklarieren
Dim rcsKnd As Recordset
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb
' Recordset initialisieren
Set rcsKnd=db.OpenRecordset("tblKunden")

'Schleife bis zum Ende des Recordset
Do Until rcsKnd.EOF
' Name, Vorname des aktuellen
' Elements ausgeben (Zeiger zeigt drauf)
Debug.Print rcsKnd.Fields("kndName") & _
  ", " & rcsKnd.Fields("kndVorname")
' Zeigeriterrücken
rcsKnd.MoveNext
' Solange fortfahren, bis Ende erreicht
Loop

End Sub
```



Navigation im Recordset per Zeiger



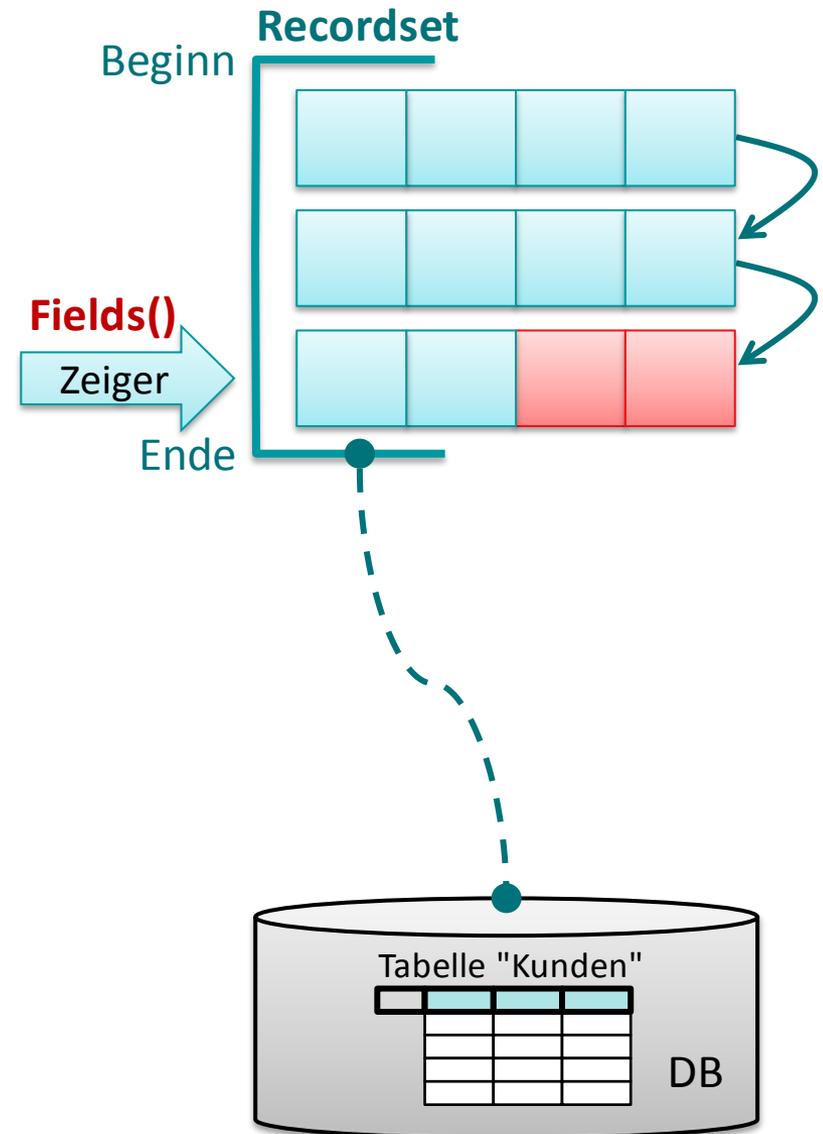
Modul

```
Option Compare Database
Option Explicit

Sub ausgebenAlleKunden()
' Recordset deklarieren
Dim rcsKnd As Recordset
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb
' Recordset initialisieren
Set rcsKnd=db.OpenRecordset("tblKunden")

'Schleife bis zum Ende des Recordset
Do Until rcsKnd.EOF
' Name, Vorname des aktuellen
' Elements ausgeben (Zeiger zeigt drauf)
Debug.Print rcsKnd.Fields("kndName") & _
", " & rcsKnd.Fields("kndVorname")
' Zeigeriterrücken
rcsKnd.MoveNext
' Solange fortfahren, bis Ende erreicht
Loop

End Sub
```



Navigation im Recordset per Zeiger



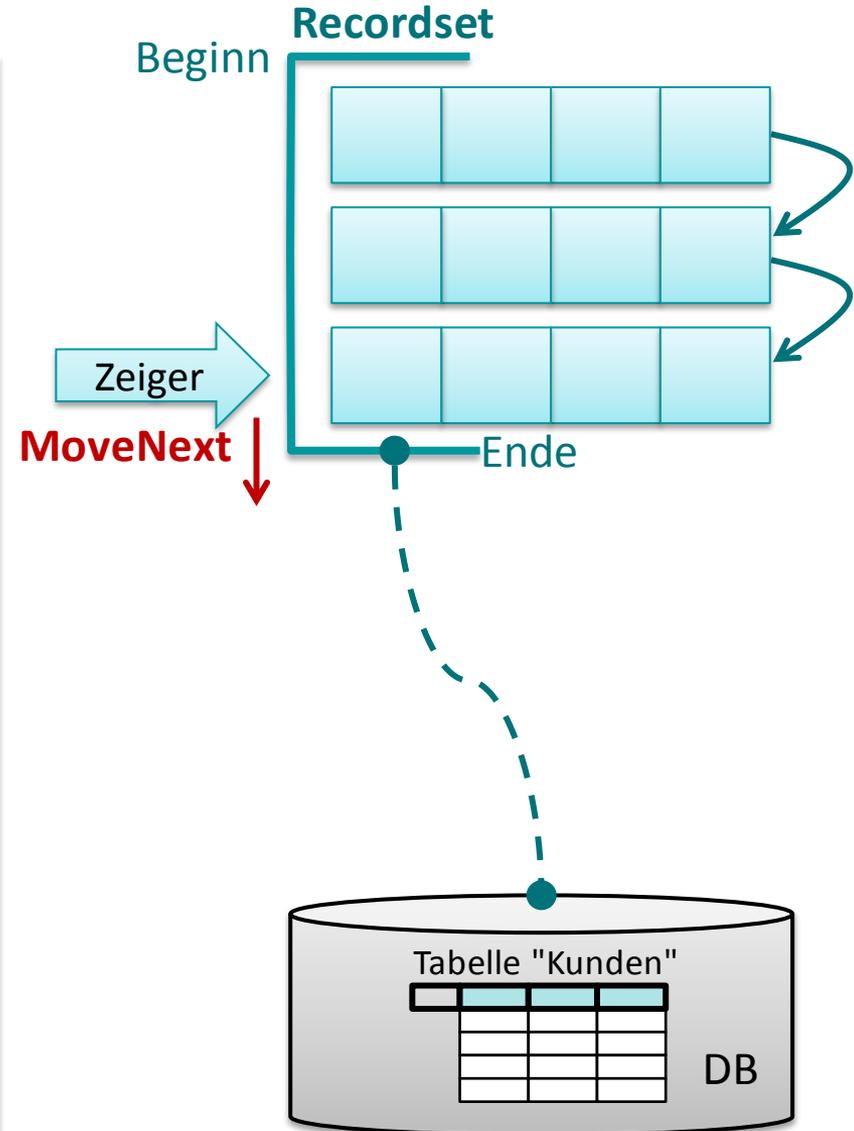
Modul

```
Option Compare Database
Option Explicit

Sub ausgebenAlleKunden()
' Recordset deklarieren
Dim rcsKnd As Recordset
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb
' Recordset initialisieren
Set rcsKnd=db.OpenRecordset("tblKunden")

'Schleife bis zum Ende des Recordset
Do Until rcsKnd.EOF
' Name, Vorname des aktuellen
' Elements ausgeben (Zeiger zeigt drauf)
Debug.Print rcsKnd.Fields("kndName") & _
", " & rcsKnd.Fields("kndVorname")
' Zeigeriterrücken
rcsKnd.MoveNext
' Solange fortfahren, bis Ende erreicht
Loop

End Sub
```



Navigation im Recordset per Zeiger



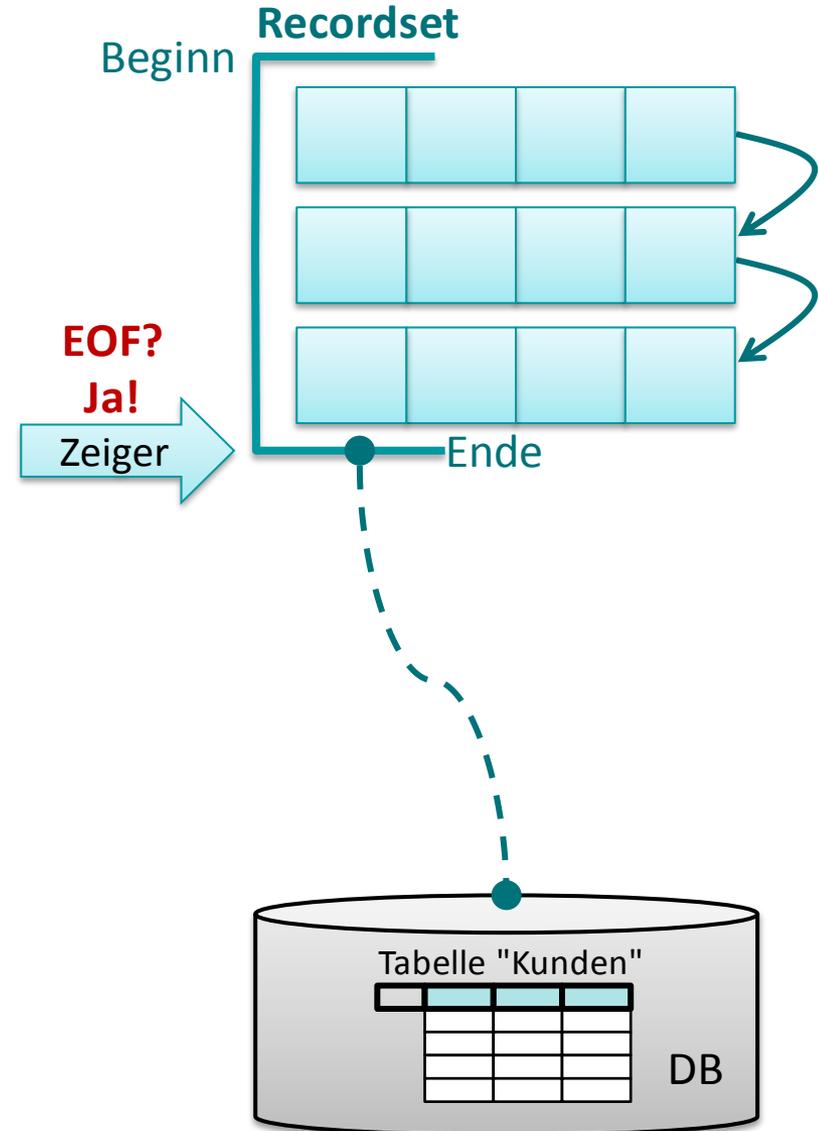
Modul

```
Option Compare Database
Option Explicit

Sub ausgebenAlleKunden()
' Recordset deklarieren
Dim rcsKnd As Recordset
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb
' Recordset initialisieren
Set rcsKnd=db.OpenRecordset("tblKunden")

'Schleife bis zum Ende des Recordset
Do Until rcsKnd.EOF
' Name, Vorname des aktuellen
' Elements ausgeben (Zeiger zeigt drauf)
Debug.Print rcsKnd.Fields("kndName") & _
  ", " & rcsKnd.Fields("kndVorname")
' Zeiger weiterrücken
rcsKnd.MoveNext
' Solange fortfahren, bis Ende erreicht
Loop

End Sub
```



Navigation im Recordset per Zeiger



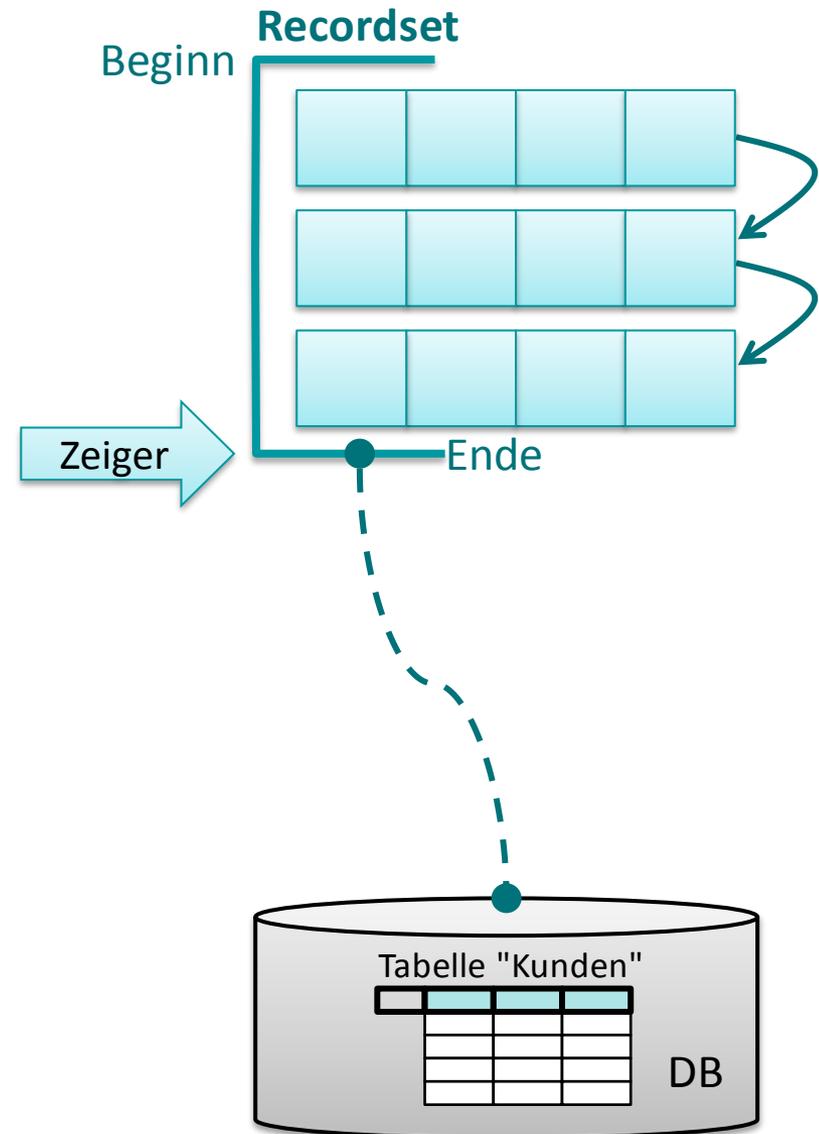
Modul

```
Option Compare Database
Option Explicit

Sub ausgebenAlleKunden()
' Recordset deklarieren
Dim rcsKnd As Recordset
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb
' Recordset initialisieren
Set rcsKnd=db.OpenRecordset("tblKunden")

'Schleife bis zum Ende des Recordset
Do Until rcsKnd.EOF
' Name, Vorname des aktuellen
' Elements ausgeben (Zeiger zeigt drauf)
Debug.Print rcsKnd.Fields("kndName") & _
", " & rcsKnd.Fields("kndVorname")
' Zeigeriterrücken
rcsKnd.MoveNext
' Solange fortfahren, bis Ende erreicht
Loop

End Sub
```



Navigation im Recordset per Zeiger



Modul

(Allgemein)

```
Option Compare Database
Option Explicit
```

```
Sub ausgebenAlleKunden()
' Recordset deklarieren
Dim rcsKnd As Recordset

' Verbindung zur Datenbank vorbereiten
Dim db As Database
Set db = CurrentDb

' Recordset initialisieren und dadurch füllen
Set rcsKnd = db.OpenRecordset("tblKunden")

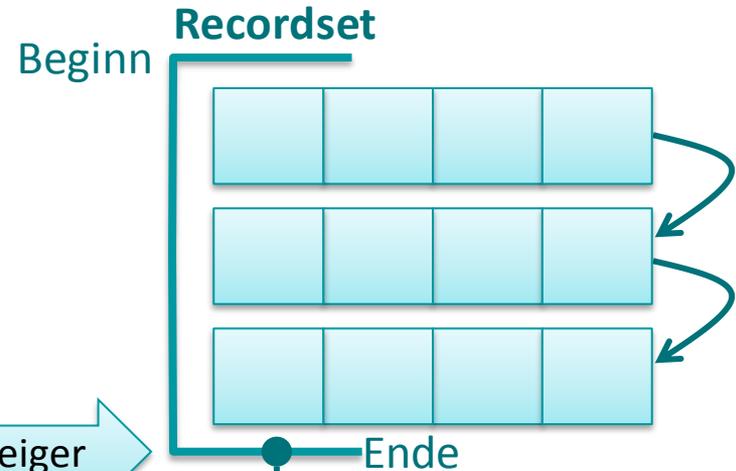
'Schleife bis zum Ende des Recordset
Do Until rcsKnd.EOF

' Name und Vorname des aktuellen Elements
' (auf das der Zeiger zeigt) ausgeben
Debug.Print rcsKnd.Fields("kndName") & _
  ", " & rcsKnd.Fields("kndVorname")

' Zeiger weiterrücken
rcsKnd.MoveNext

' Solange fortfahren, bis Ende erreicht
Loop

End Sub
```



Navigation im Recordset per Zeiger



Modul

(Allgemein)

```
Option Compare Database
Option Explicit
```

```
Sub ausgebenAlleKunden()
    ' Recordset deklarieren
    Dim rcsKnd As Recordset

    ' Verbindung zur Datenbank vorbereiten
    Dim db As Database
    Set db = CurrentDb

    ' Recordset initialisieren und dadurch füllen
    Set rcsKnd = db.OpenRecordset("tblKunden")

    'Schleife bis zum Ende des Recordset
    Do Until rcsKnd.EOF

        ' Name und Vorname des aktuellen Elements
        ' (auf das der Zeiger zeigt) ausgeben
        Debug.Print rcsKnd.Fields("kndName") & _
            ", " & rcsKnd.Fields("kndVorname")

        ' Zeiger weiterrücken
        rcsKnd.MoveNext

    ' Solange fortfahren, bis Ende erreicht
    Loop

End Sub
```



tblKunden			
	kndIdPk	kndName	kndVorname
+	1	Meier	Thomas
+	2	Müller	Sven
+	3	Meier	Mike
+	4	Müller	Silke

Navigation im Recordset per Zeiger



Modul

(Allgemein)

```
Option Compare Database
Option Explicit
```

```
Sub ausgebenAlleKunden()
' Recordset deklarieren
Dim rcsKnd As Recordset

' Verbindung zur Datenbank vorbereiten
Dim db As Database
Set db = CurrentDb

' Recordset initialisieren und dadurch füllen
Set rcsKnd = db.OpenRecordset("tblKunden")

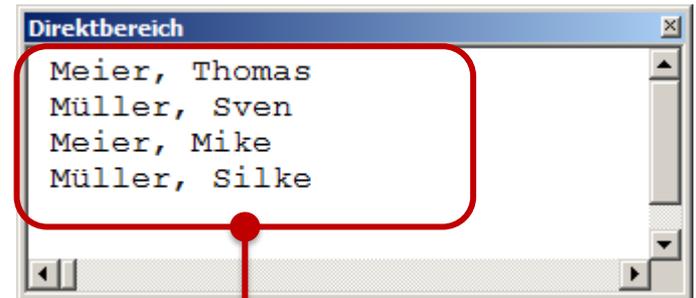
'Schleife bis zum Ende des Recordset
Do Until rcsKnd.EOF

' Name und Vorname des aktuellen Elements
' (auf das der Zeiger zeigt) ausgeben
Debug.Print rcsKnd.Fields("kndName") & _
  ", " & rcsKnd.Fields("kndVorname")

' Zeiger weiterrücken
rcsKnd.MoveNext

' Solange fortfahren, bis Ende erreicht
Loop

End Sub
```



tblKunden			
	kndIdPk	kndName	kndVorname
+	1	Meier	Thomas
+	2	Müller	Sven
+	3	Meier	Mike
+	4	Müller	Silke

Navigation im Recordset per Zeiger

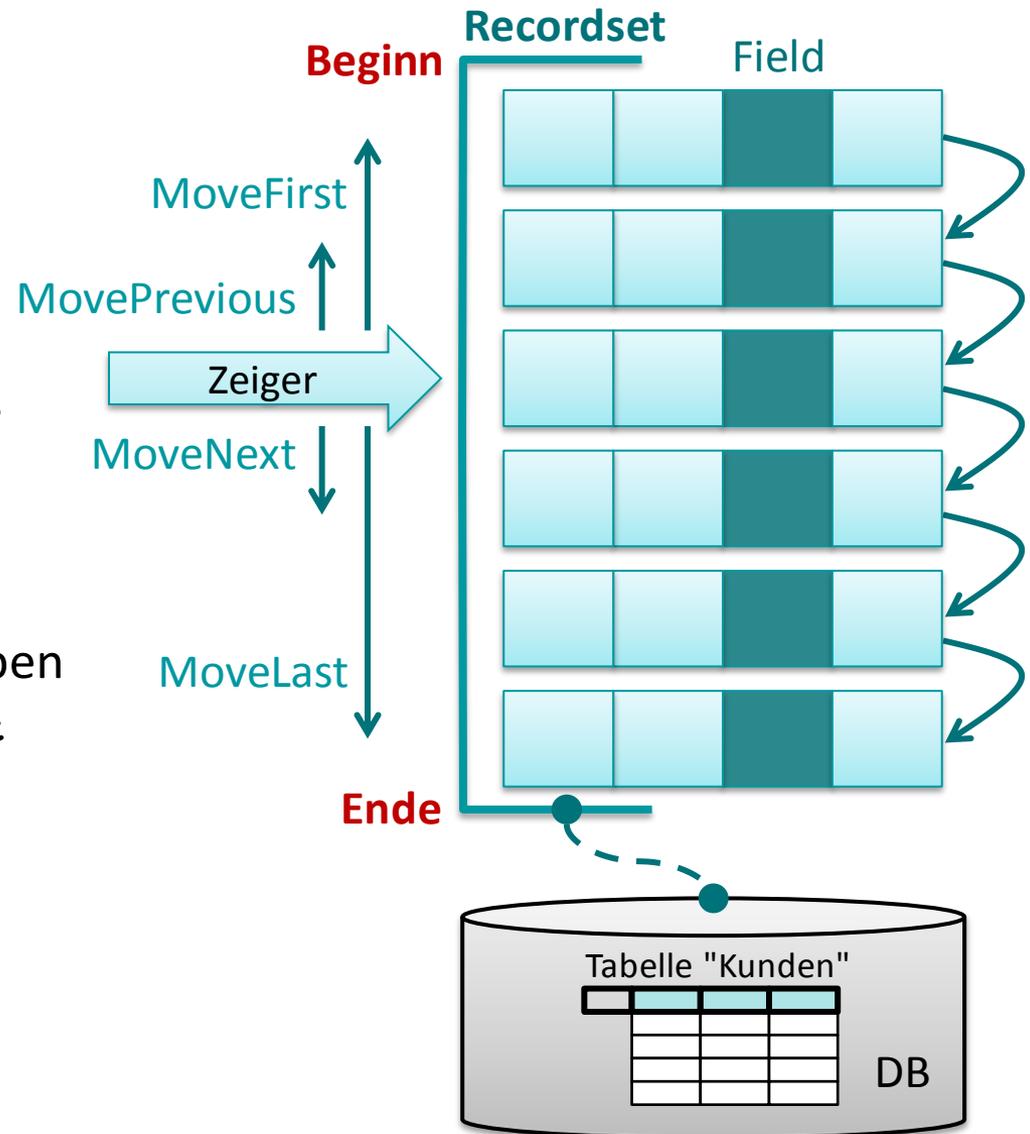


Navigationsmöglichkeiten

- zum nächsten Datensatz verschieben
`<einRecordset>.MoveNext`
- zum vorherigen Datensatz verschieben
`<einRcrdst>.MovePrevious`
- zum letzten Datensatz verschieben
`<einRecordset>.MoveLast`
- zum ersten Datensatz verschieben
`<einRecordset>.MoveFirst`

Positionsbestimmung, ob Zeiger am Beginn oder am Ende steht (End of File)

- `<einRecordset>.BOF`
- `<einRecordset>.EOF`



Ändern und Hinzufügen: Demo 07.01



D07.01: Verwenden Sie ein Recordset, um

- die Bezeichnung und den Preis aller Produkte im Direktbereich auszugeben
- Ausgangspunkt Tabelle "tblProdukte" mit Spalte "prdPreis"

tblProdukte	tblProduktKategorie		
prdIdPk	prdBezeichnung	prdBeschreibung	prdPreis
1	Blumenzwiebeln Sorte XY	Rote Blumen, sehr schön	0,59 €
2	Blumensamen, Sorte ABC	Gelbe Blumen, klein	0,99 €
3	Sonnenblumensamen, Sorte XXL	Große gelbe Blüten, sehr schön	1,99 €
4	Rasendünger Maxi	5kg, Lanzeitdünger	5,99 €
5	Rasendünger Mega	10kg, Sofortwirkung	9,99 €
6	Blumendünger "Sabine"	500 ml, Grün- und Zimmerpflanzen	0,99 €
7	Blumendünger "Dörte"	500 ml, Bio-Dünger	2,99 €
8	Rindenmulch fein	6 kg, feiner Mulch	2,49 €
9	Rindenmulch grob	6 kg, grober Mulch	2,69 €
10	Rasenbesen "Bert"	Markenprodukt, lange Haltbarkeit	5,99 €



Inhalt

Ziel und Einordnung

Rückblick

Ausgangssituation

Datenbankschnittstelle

Recordsets

- Was sind Recordsets
- Mit Recordsets Daten lesen
- Navigation im Recordset per Zeiger
- Suchen und Finden im Recordset
- Ändern und Hinzufügen im Recordset
- Kombination mit grafischen Abfragen und SQL

Weitere VBA-Funktionen

Fallstricke

Zusammenfassung und Ausblick



Suchen und Finden im Recordset



Recordset-Typen, die beim Befüllen angegeben werden können

- Dynaset: Änderungen am Datenbestand in der Datenbank spiegeln sich im Recordset wider

```
db.OpenRecordset("tblPersonen",  
RecordsetTypeEnum.dbOpenDynaset)
```

- Table: Keine Aktualisierung, hat Index; Keine Suche/Sortierung möglich

```
db.OpenRecordset("tblPersonen",  
RecordsetTypeEnum.dbOpenTable)
```

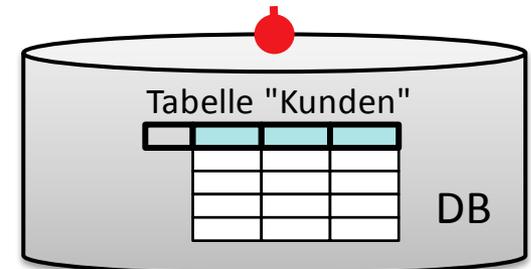
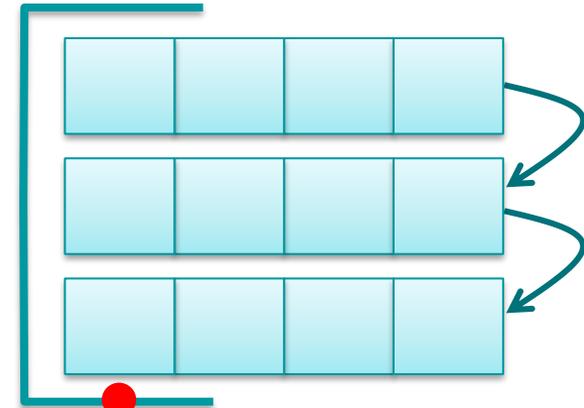
- ForwardOnly: Recordset kann nur einmal vorwärts durchlaufen werden; spart Speicherplatz

```
db.OpenRecordset("tblPersonen",  
RecordsetTypeEnum.dbOpenForwardOnly)
```

- Snapshot: Kopie der Daten zum Zeitpunkt des Befüllens

```
db.OpenRecordset("tblPersonen",  
RecordsetTypeEnum.dbOpenSnapshot)
```

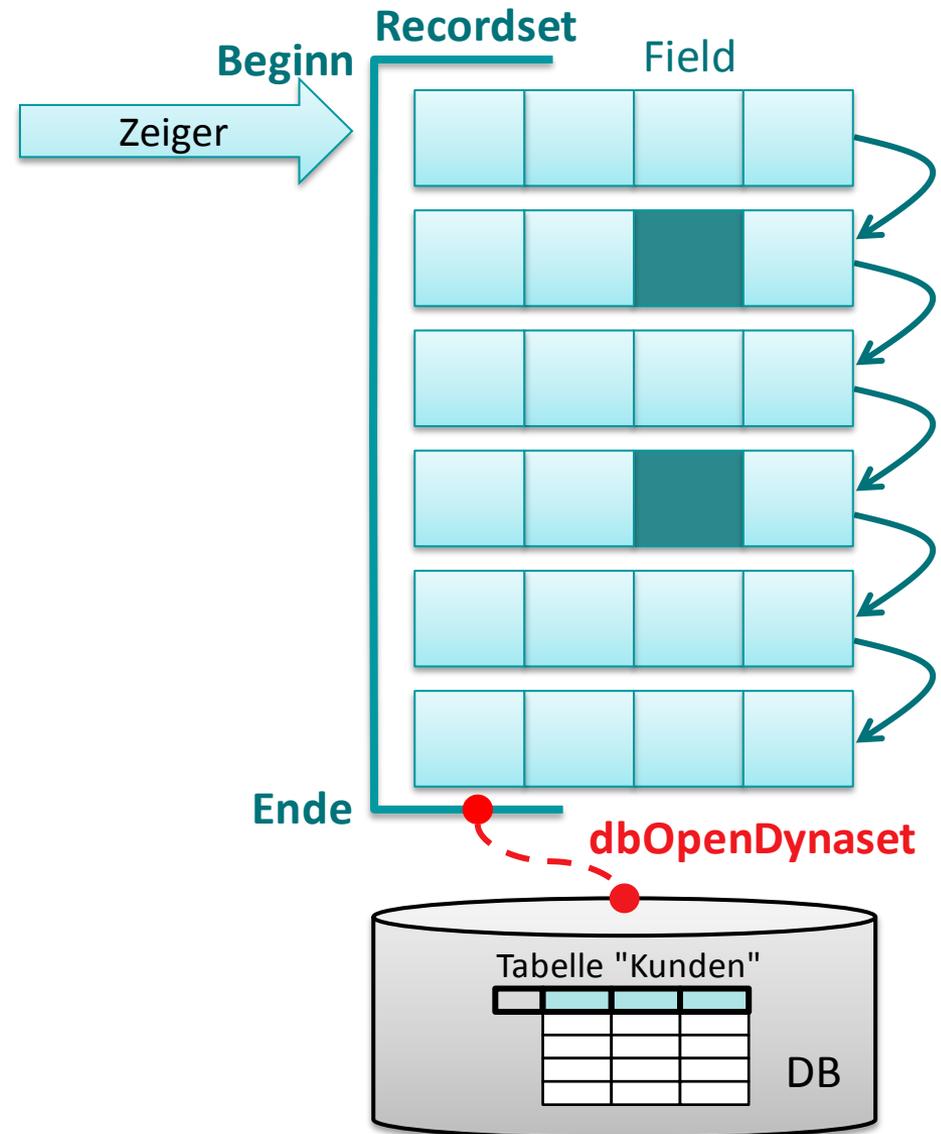
Recordset



Suchen und Finden im Recordset



Navigationsmöglichkeiten

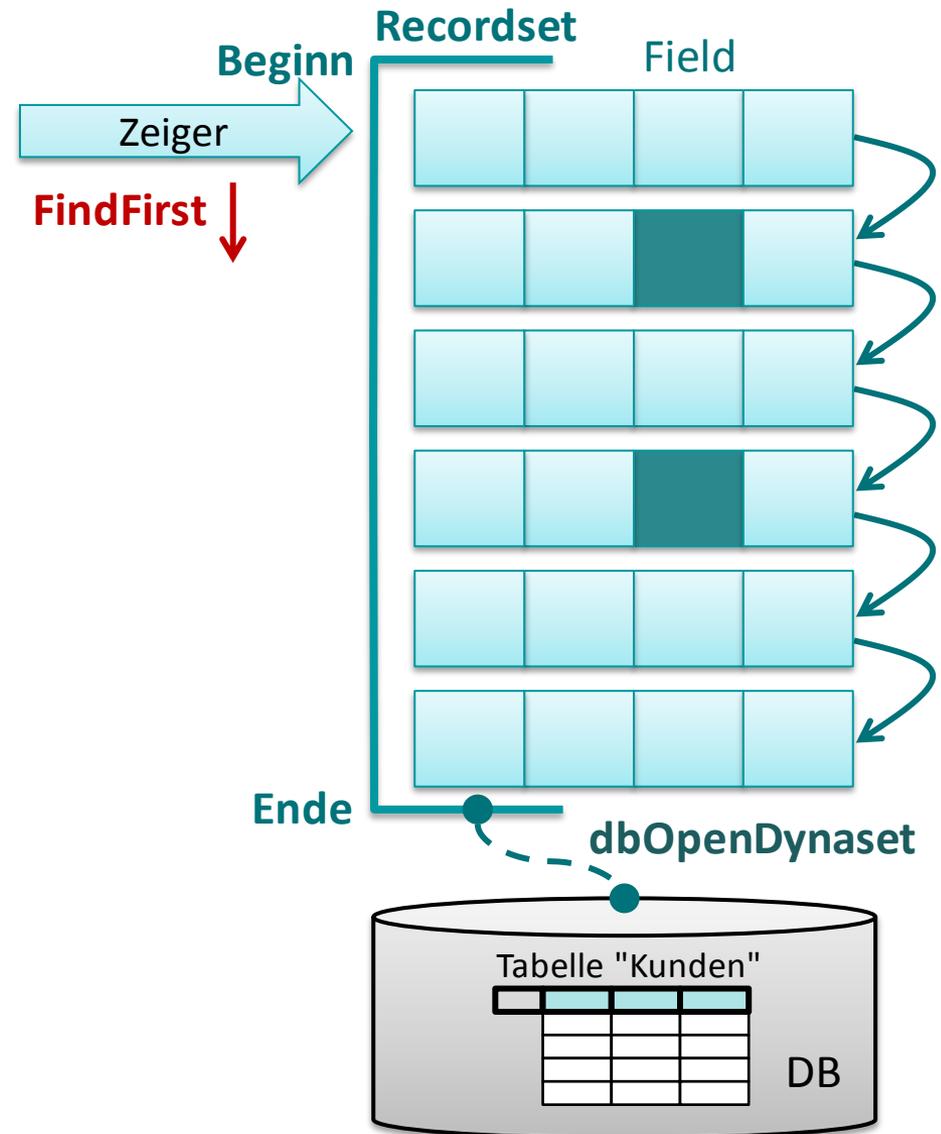


Suchen und Finden im Recordset



Navigationsmöglichkeiten

- ersten Datensatz finden, der ein bestimmtes Kriterium erfüllt
`<einRecordset>.FindFirst()`

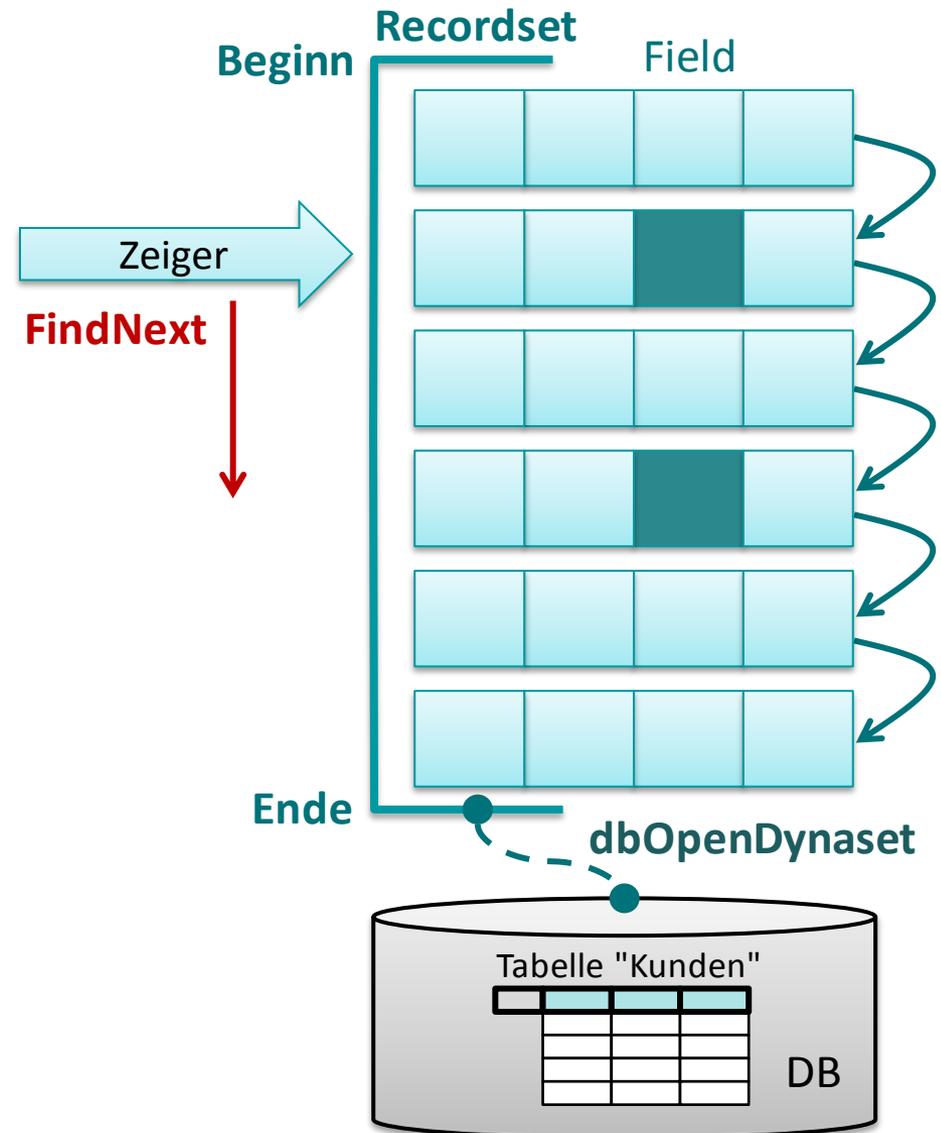


Suchen und Finden im Recordset



Navigationsmöglichkeiten

- ersten Datensatz finden, der ein bestimmtes Kriterium erfüllt
`<einRecordset>.FindFirst()`
- nächsten Datensatz finden, der ein bestimmtes Kriterium erfüllt
`<einRecordset>.FindNext()`

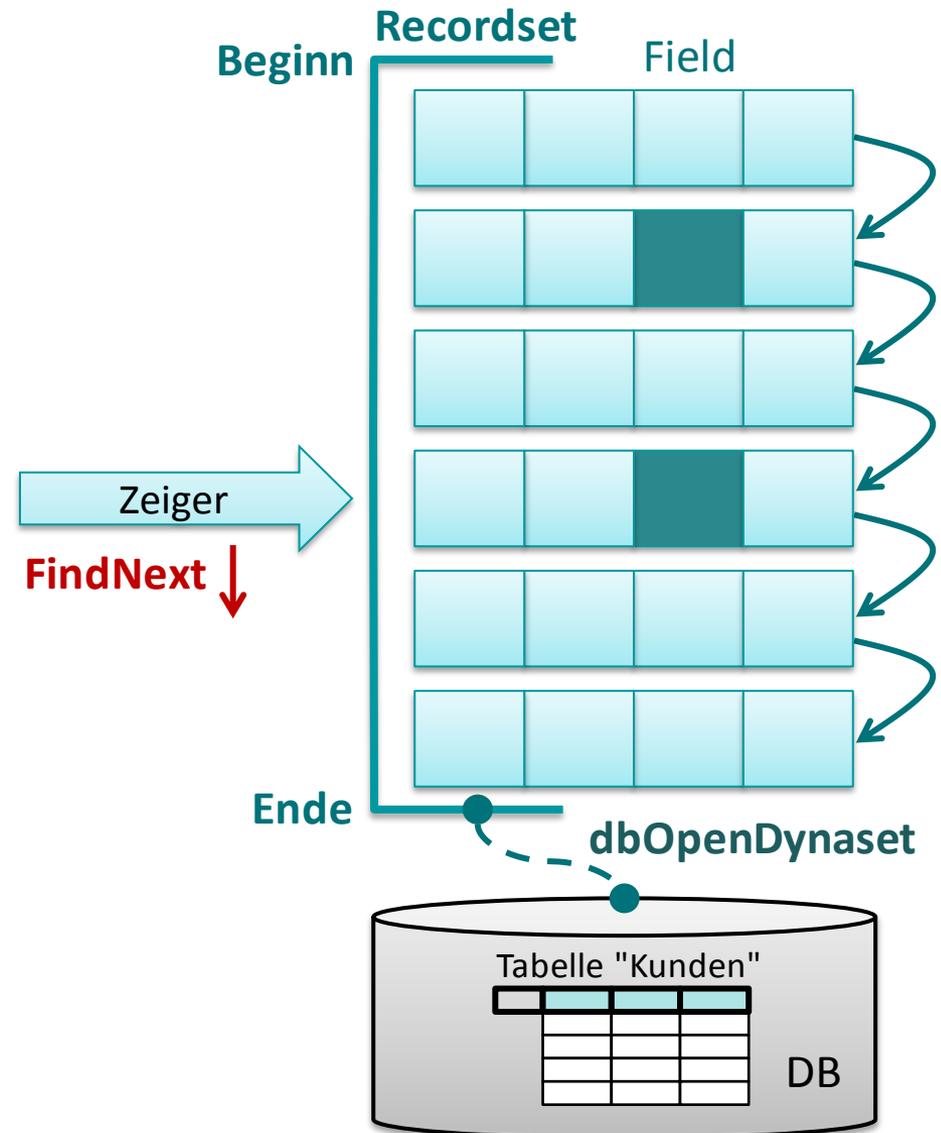


Suchen und Finden im Recordset



Navigationsmöglichkeiten

- ersten Datensatz finden, der ein bestimmtes Kriterium erfüllt
`<einRecordset>.FindFirst()`
- nächsten Datensatz finden, der ein bestimmtes Kriterium erfüllt
`<einRecordset>.FindNext()`



Suchen und Finden im Recordset

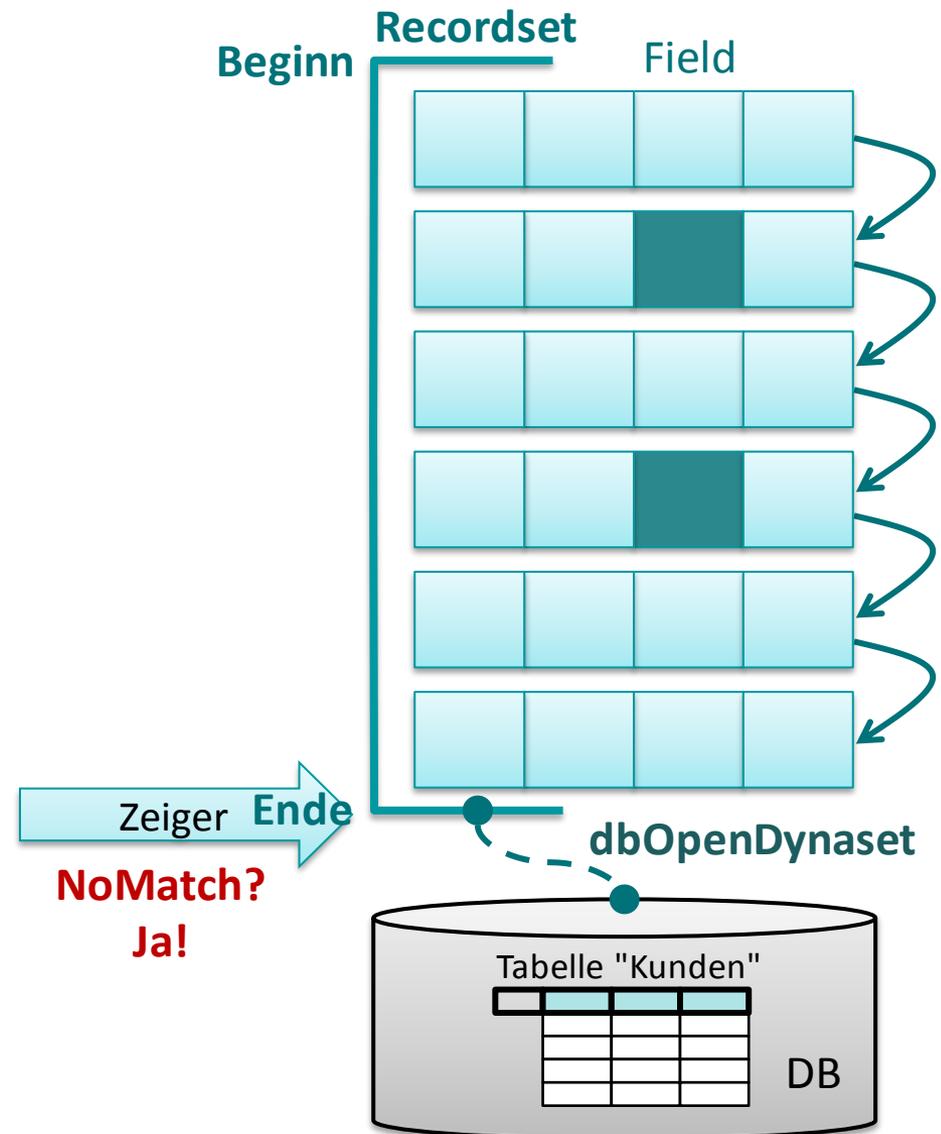


Navigationsmöglichkeiten

- ersten Datensatz finden, der ein bestimmtes Kriterium erfüllt
`<einRecordset>.FindFirst()`
- nächsten Datensatz finden, der ein bestimmtes Kriterium erfüllt
`<einRecordset>.FindNext()`

Positionsbestimmung, ob Datensätze gefunden werden konnten

`<einRecordset>.NoMatch`



Suchen und Finden im Recordset

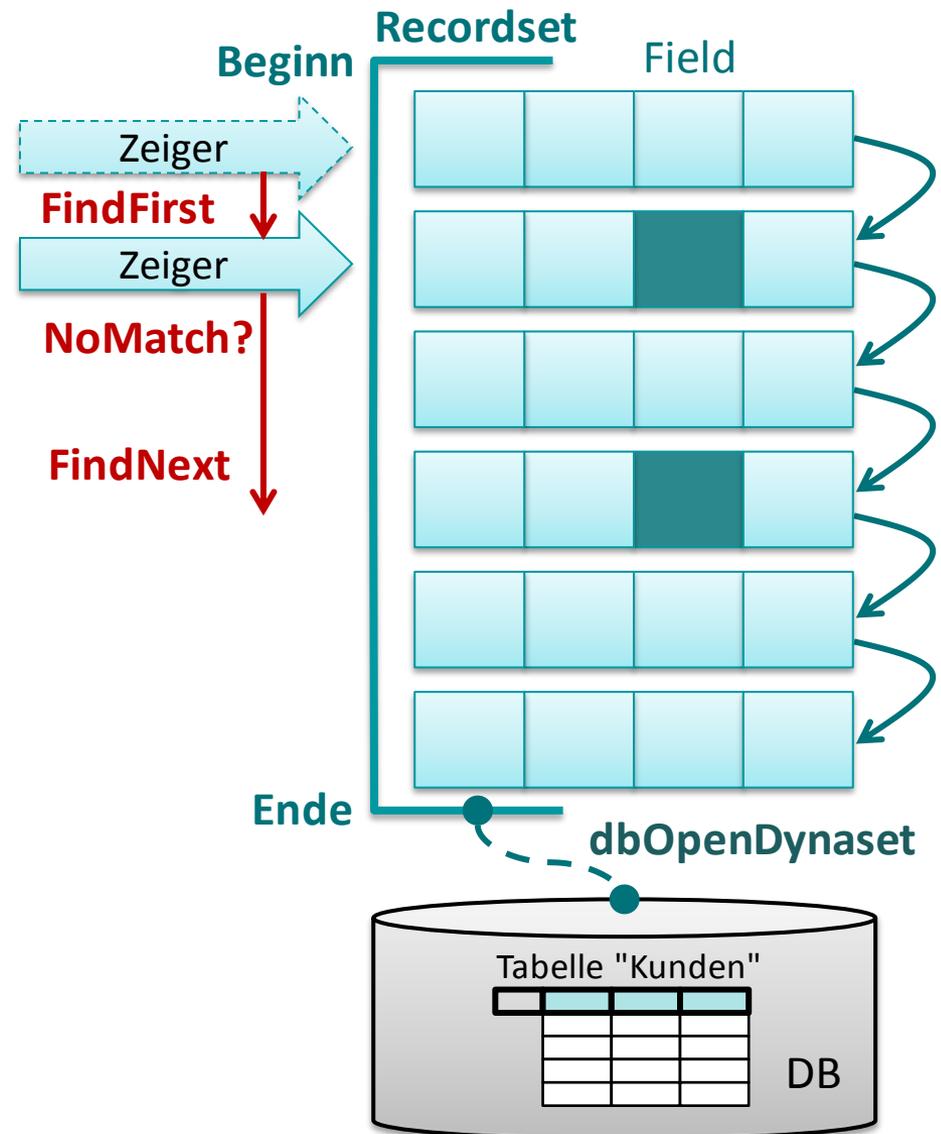


Navigationsmöglichkeiten

- ersten Datensatz finden, der ein bestimmtes Kriterium erfüllt
`<einRecordset>.FindFirst()`
- nächsten Datensatz finden, der ein bestimmtes Kriterium erfüllt
`<einRecordset>.FindNext()`

Positionsbestimmung, ob Datensätze gefunden werden konnten

`<einRecordset>.NoMatch`



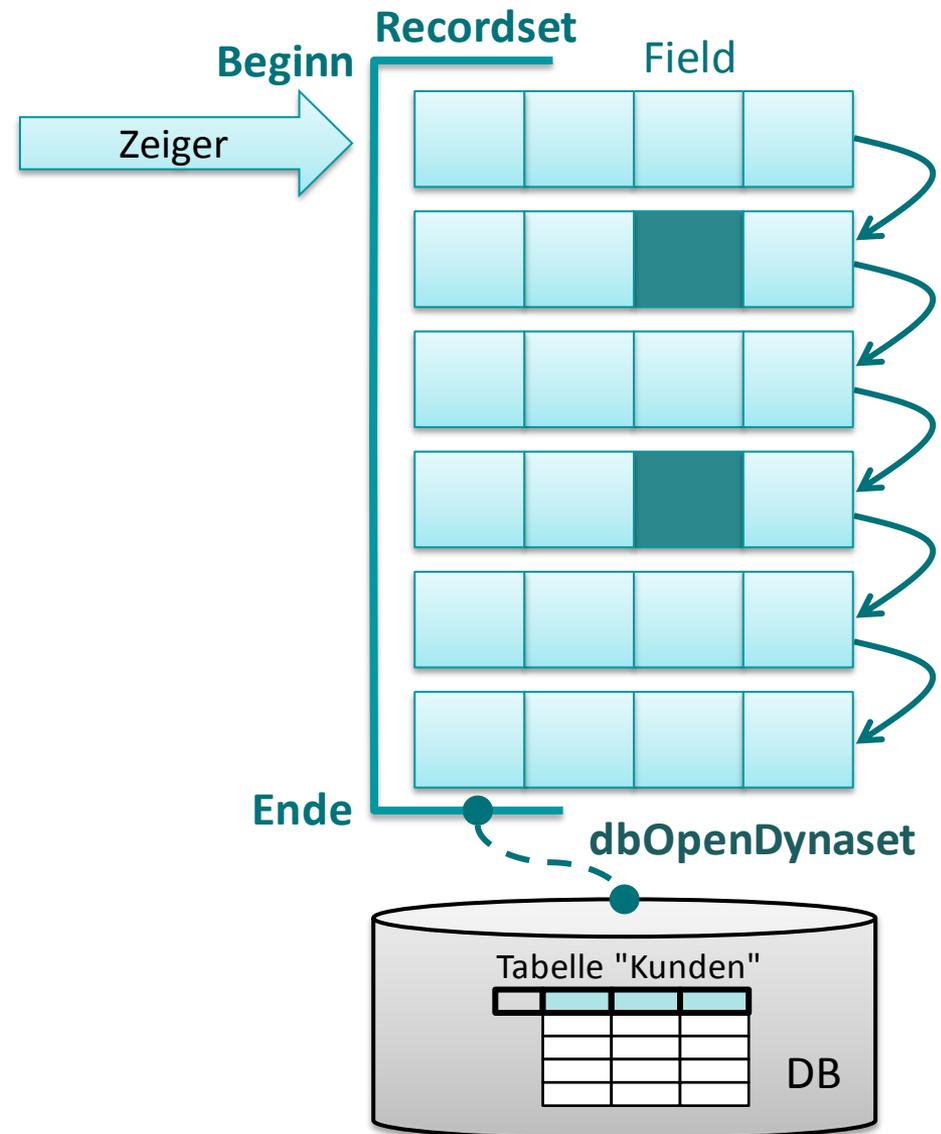
Suchen und Finden im Recordset



Ziel: Aus dem Recordset aller Personen sollen die Kunden gefunden und ausgegeben werden, die "Meier" heißen.

Ansatz

- Recordset als Dynaset befüllen, Zeiger steht auf erstem Datensatz oder wenn leer, dann auf Ende



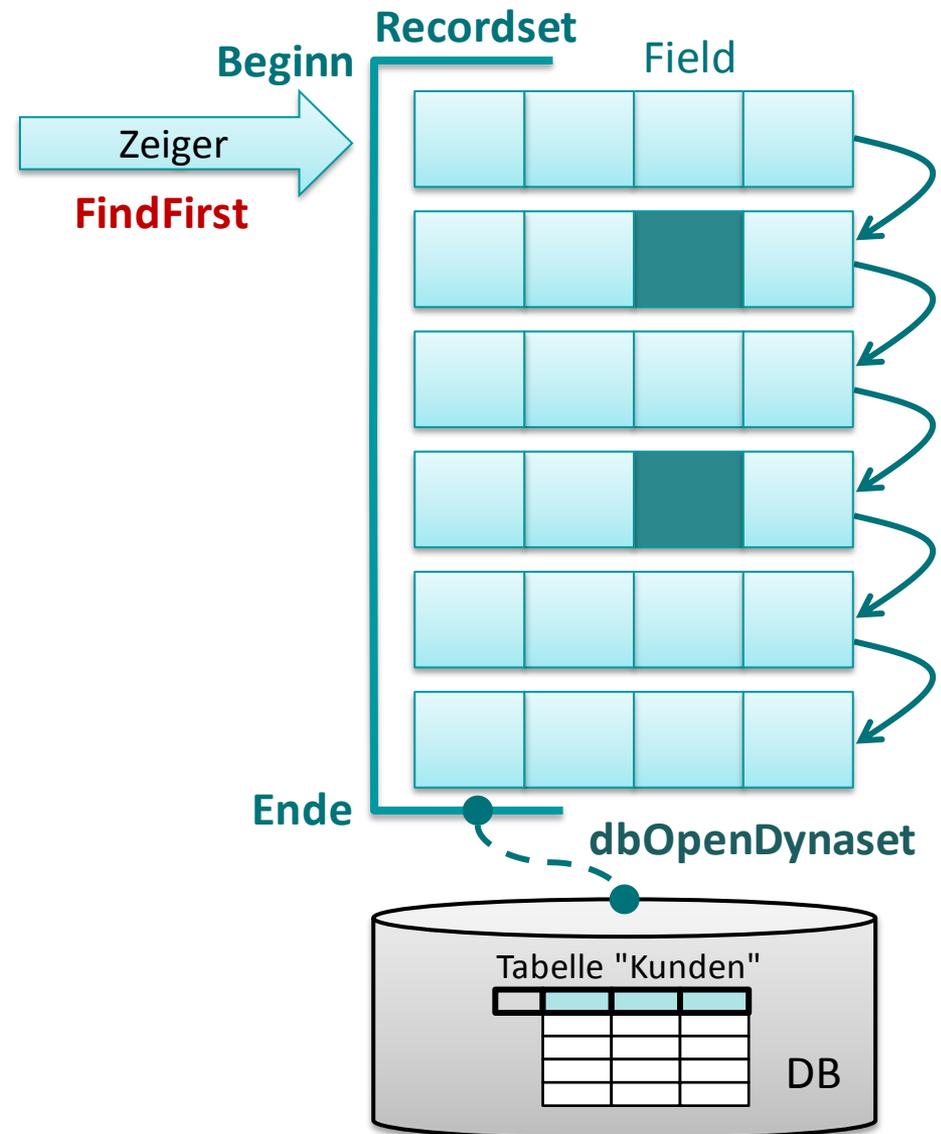
Suchen und Finden im Recordset



Ziel: Aus dem Recordset aller Personen sollen die Kunden gefunden und ausgegeben werden, die "Meier" heißen.

Ansatz

- Recordset als Dnyaset befüllen, Zeiger steht auf erstem Datensatz oder wenn leer, dann auf Ende
- Recordset nach erstem Kunden "Meier" durchsuchen



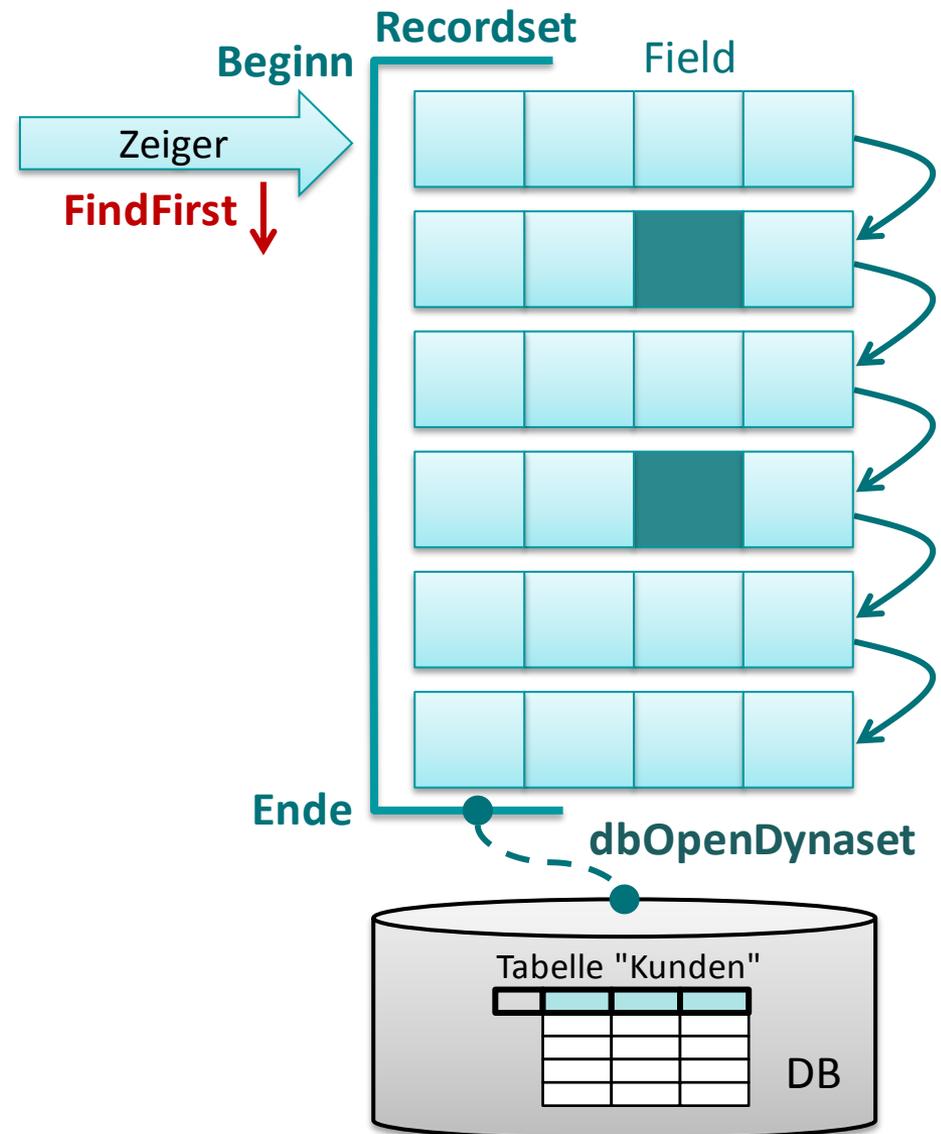
Suchen und Finden im Recordset



Ziel: Aus dem Recordset aller Personen sollen die Kunden gefunden und ausgegeben werden, die "Meier" heißen.

Ansatz

- Recordset als Dynaset befüllen, Zeiger steht auf erstem Datensatz oder wenn leer, dann auf Ende
- Recordset nach erstem Kunden "Meier" durchsuchen
- Wenn gefunden, dann Kunden ausgeben



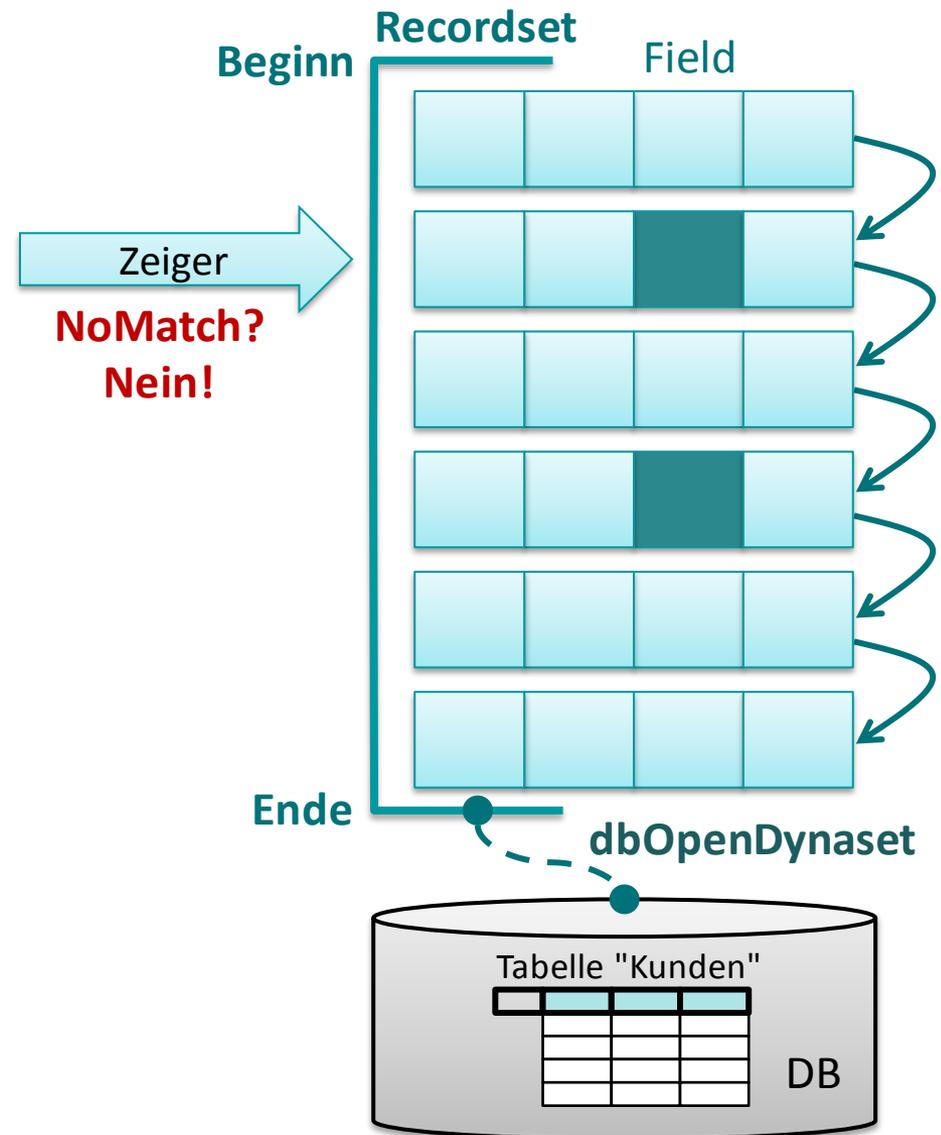
Suchen und Finden im Recordset



Ziel: Aus dem Recordset aller Personen sollen die Kunden gefunden und ausgegeben werden, die "Meier" heißen.

Ansatz

- Recordset als Dnyaset befüllen, Zeiger steht auf erstem Datensatz oder wenn leer, dann auf Ende
- Recordset nach erstem Kunden "Meier" durchsuchen
- Wenn gefunden, dann Kunden ausgeben
- Weitersuchen in Schleife, bis keine Treffer (mehr)



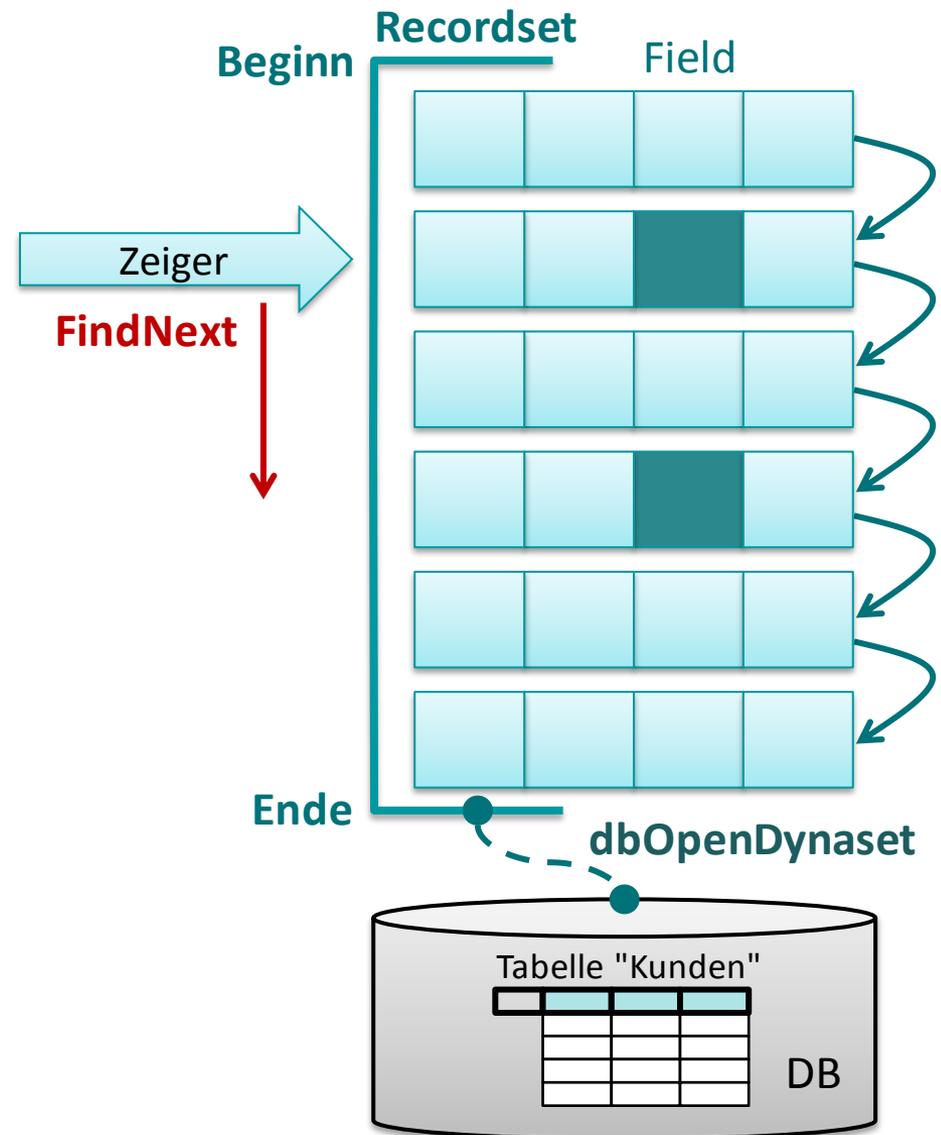
Suchen und Finden im Recordset



Ziel: Aus dem Recordset aller Personen sollen die Kunden gefunden und ausgegeben werden, die "Meier" heißen.

Ansatz

- Recordset als Dynaset befüllen, Zeiger steht auf erstem Datensatz oder wenn leer, dann auf Ende
- Recordset nach erstem Kunden "Meier" durchsuchen
- Wenn gefunden, dann Kunden ausgeben
- Weitersuchen in Schleife, bis keine Treffer (mehr)



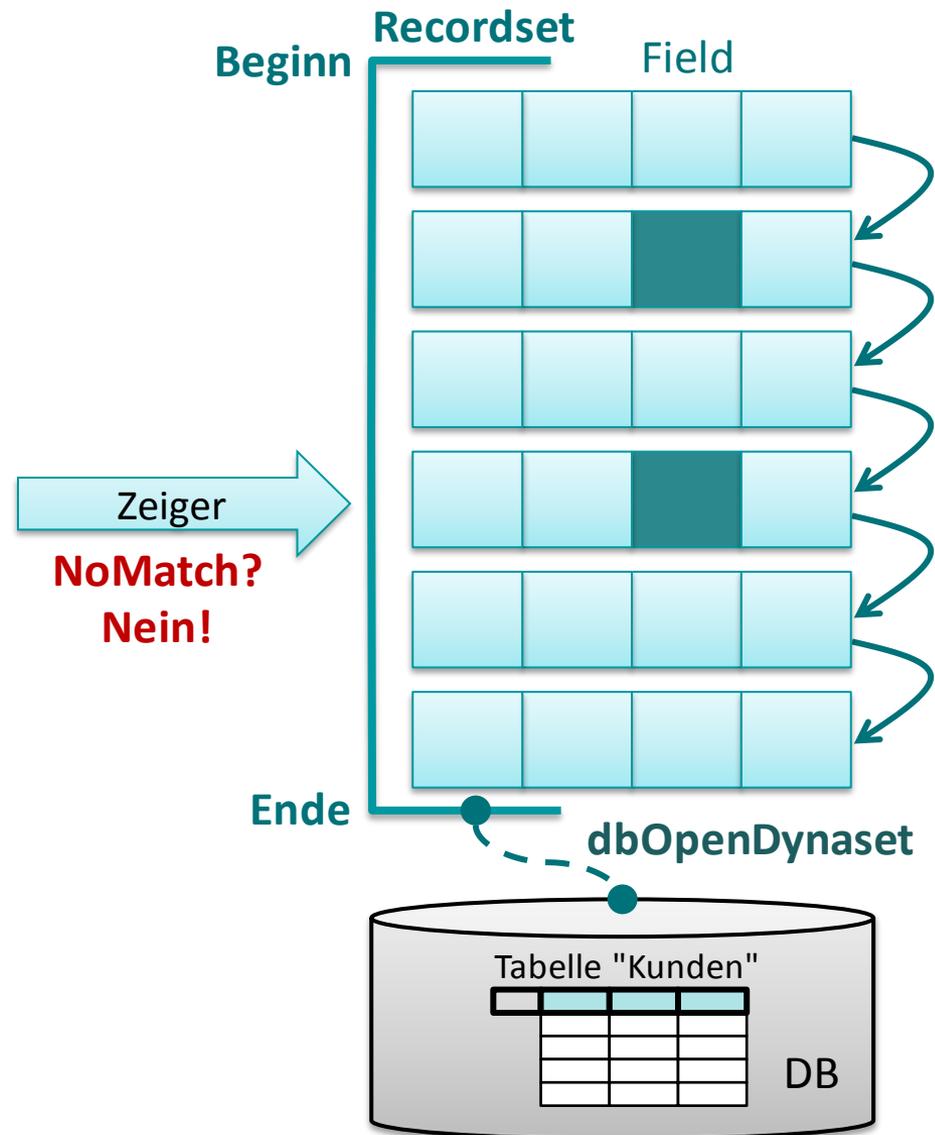
Suchen und Finden im Recordset



Ziel: Aus dem Recordset aller Personen sollen die Kunden gefunden und ausgegeben werden, die "Meier" heißen.

Ansatz

- Recordset als Dynaset befüllen, Zeiger steht auf erstem Datensatz oder wenn leer, dann auf Ende
- Recordset nach erstem Kunden "Meier" durchsuchen
- Wenn gefunden, dann Kunden ausgeben
- Weitersuchen in Schleife, bis keine Treffer (mehr)



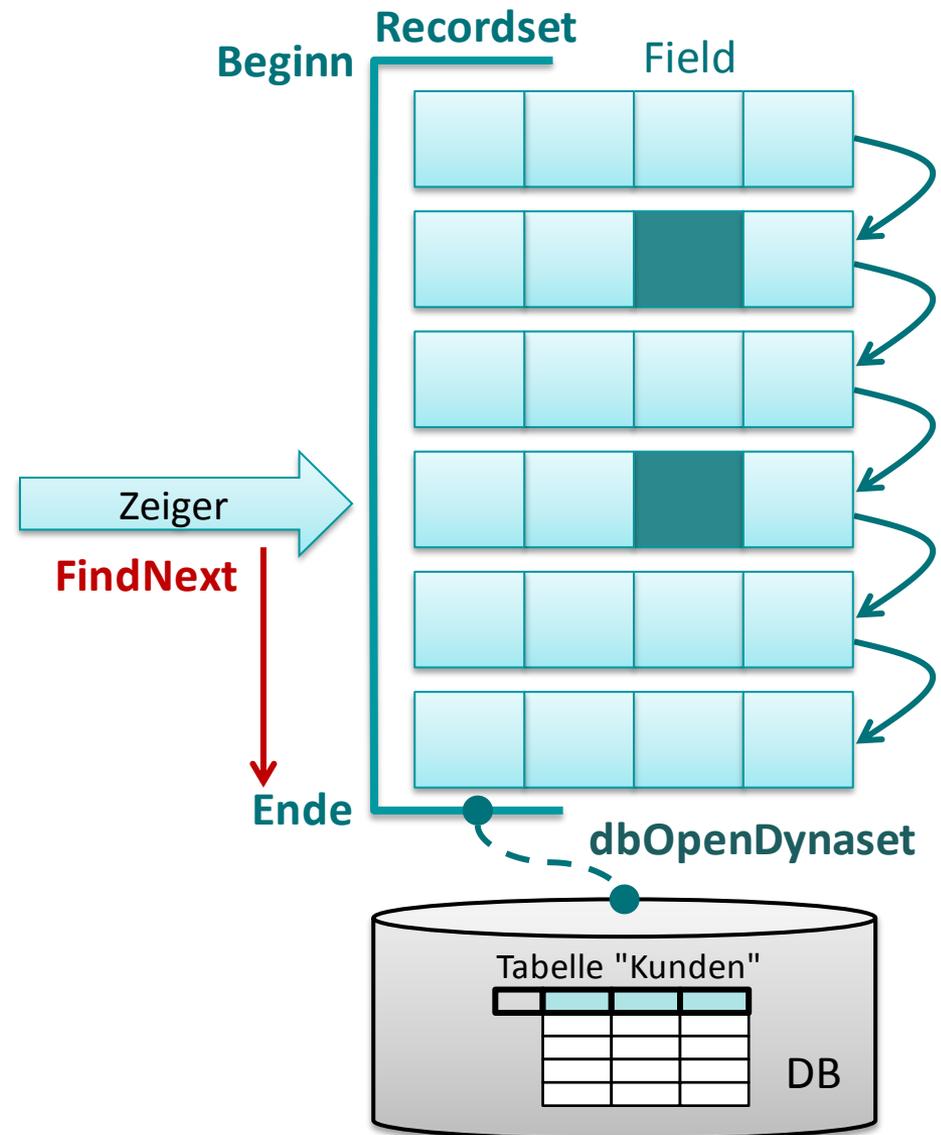
Suchen und Finden im Recordset



Ziel: Aus dem Recordset aller Personen sollen die Kunden gefunden und ausgegeben werden, die "Meier" heißen.

Ansatz

- Recordset als Dynaset befüllen, Zeiger steht auf erstem Datensatz oder wenn leer, dann auf Ende
- Recordset nach erstem Kunden "Meier" durchsuchen
- Wenn gefunden, dann Kunden ausgeben
- Weitersuchen in Schleife, bis keine Treffer (mehr)



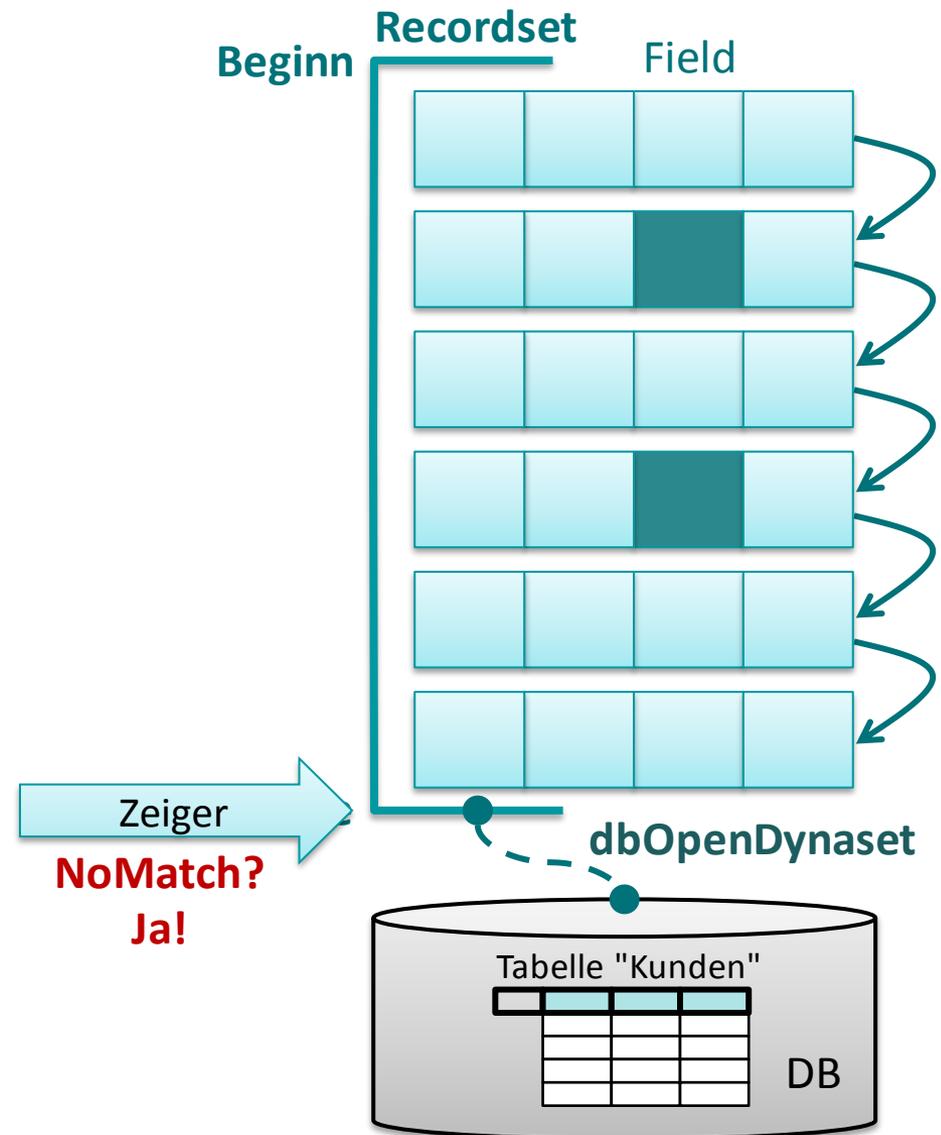
Suchen und Finden im Recordset



Ziel: Aus dem Recordset aller Personen sollen die Kunden gefunden und ausgegeben werden, die "Meier" heißen.

Ansatz

- Recordset als Dynaset befüllen, Zeiger steht auf erstem Datensatz oder wenn leer, dann auf Ende
- Recordset nach erstem Kunden "Meier" durchsuchen
- Wenn gefunden, dann Kunden ausgeben
- Weitersuchen in Schleife, bis keine Treffer (mehr)



Suchen und Finden im Recordset



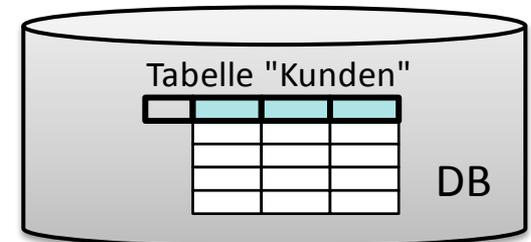
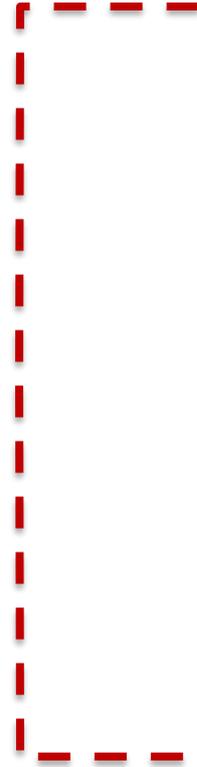
Modul

```
Option Compare Database
Option Explicit

Sub sucheKundenMeier()
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb
' Recordset deklarieren, initialisieren
Dim rcsKnd As Recordset
```

```
End Sub
```

Recordset



Suchen und Finden im Recordset

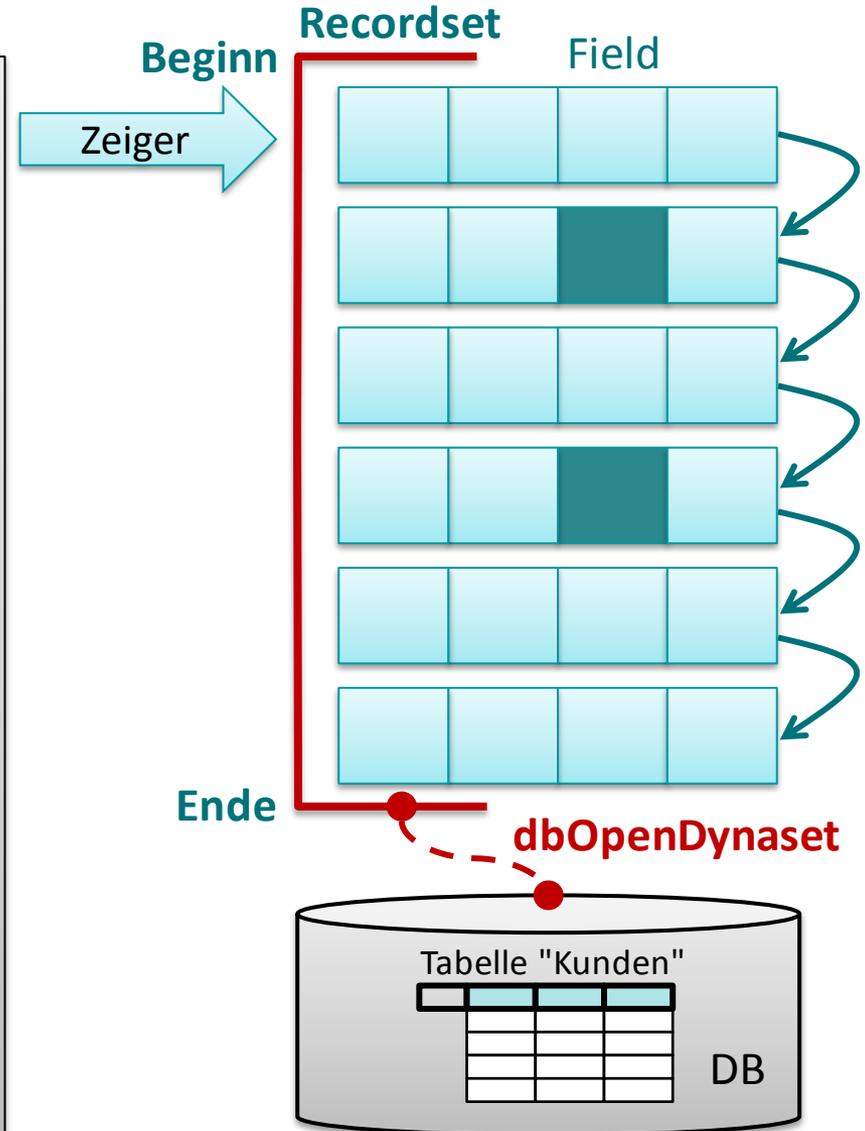


Modul

```
Option Compare Database
Option Explicit

Sub sucheKundenMeier()
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb
' Recordset deklarieren, initialisieren
Dim rcsKnd As Recordset
Set rcsKnd=db.OpenRecordset ("tblKunden",
                             dbOpenDynaset)

End Sub
```



Suchen und Finden im Recordset

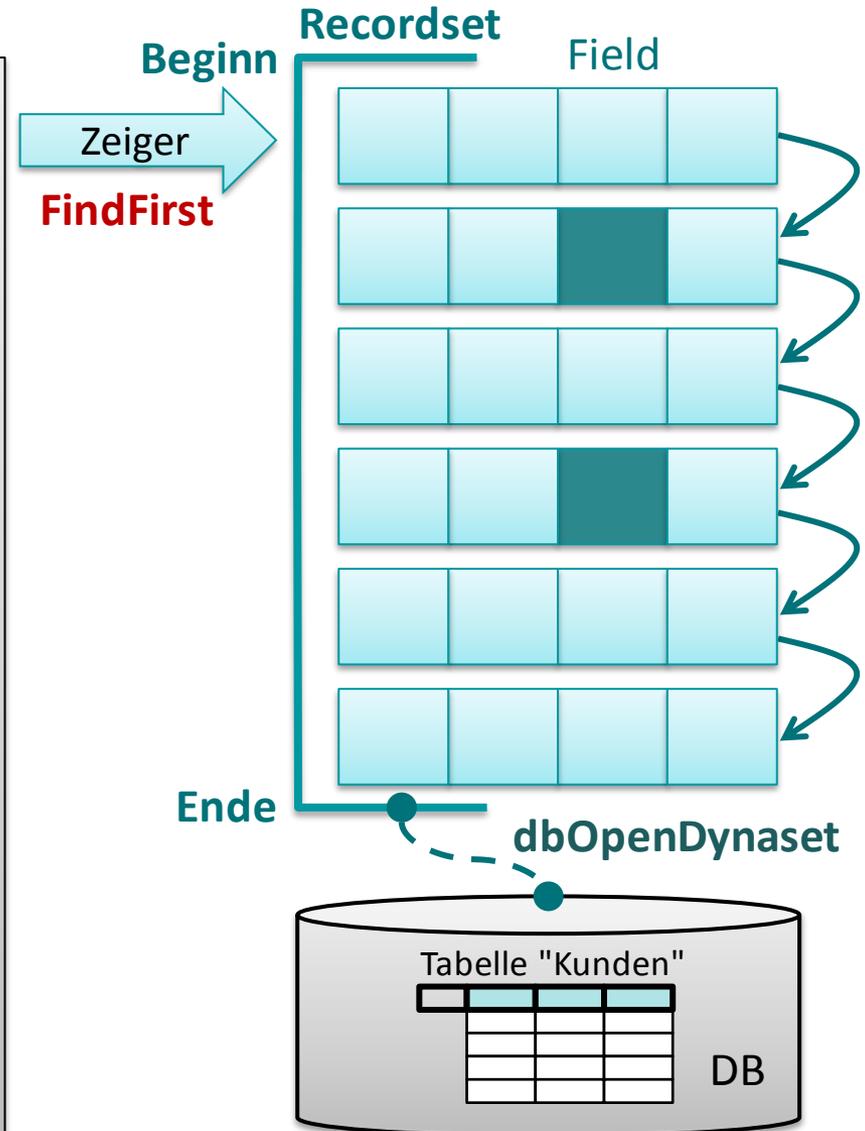


Modul

```
Option Compare Database
Option Explicit

Sub sucheKundenMeier()
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb
' Recordset deklarieren, initialisieren
Dim rcsKnd As Recordset
Set rcsKnd=db.OpenRecordset ("tblKunden",
                             dbOpenDynaset)
' Suchen nach Kunden "Meier"
rcsKnd.FindFirst ("kndName = 'Meier'")

End Sub
```



Suchen und Finden im Recordset

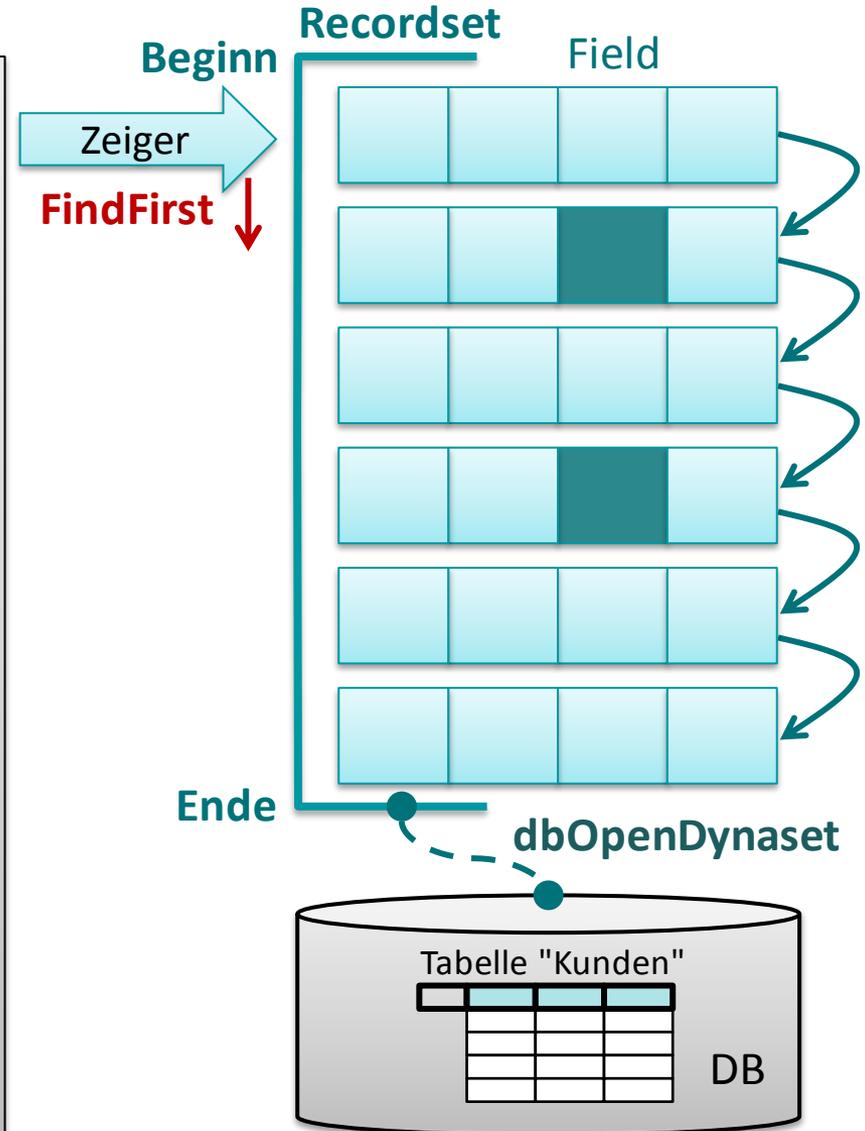


Modul

```
Option Compare Database
Option Explicit

Sub sucheKundenMeier()
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb
' Recordset deklarieren, initialisieren
Dim rcsKnd As Recordset
Set rcsKnd=db.OpenRecordset ("tblKunden",
                             dbOpenDynaset)
' Suchen nach Kunden "Meier"
rcsKnd.FindFirst ("kndName = 'Meier'")

End Sub
```



Suchen und Finden im Recordset



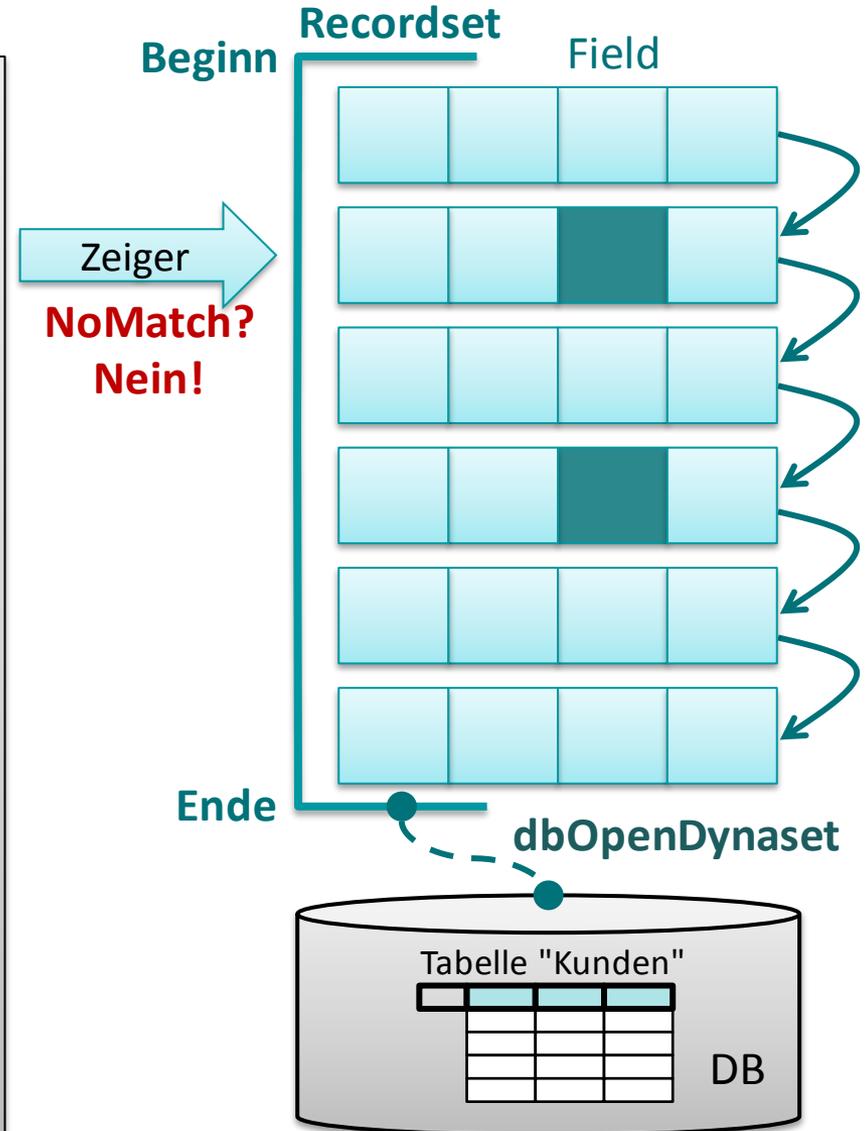
Modul

```
Option Compare Database
Option Explicit

Sub sucheKundenMeier()
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb
' Recordset deklarieren, initialisieren
Dim rcsKnd As Recordset
Set rcsKnd=db.OpenRecordset("tblKunden",
                           dbOpenDynaset)
' Suchen nach Kunden "Meier"
rcsKnd.FindFirst("kndName = 'Meier'")

' Schleife bis zum Ende des Recordset
Do Until rcsKnd.NoMatch

Loop
End Sub
```



Suchen und Finden im Recordset



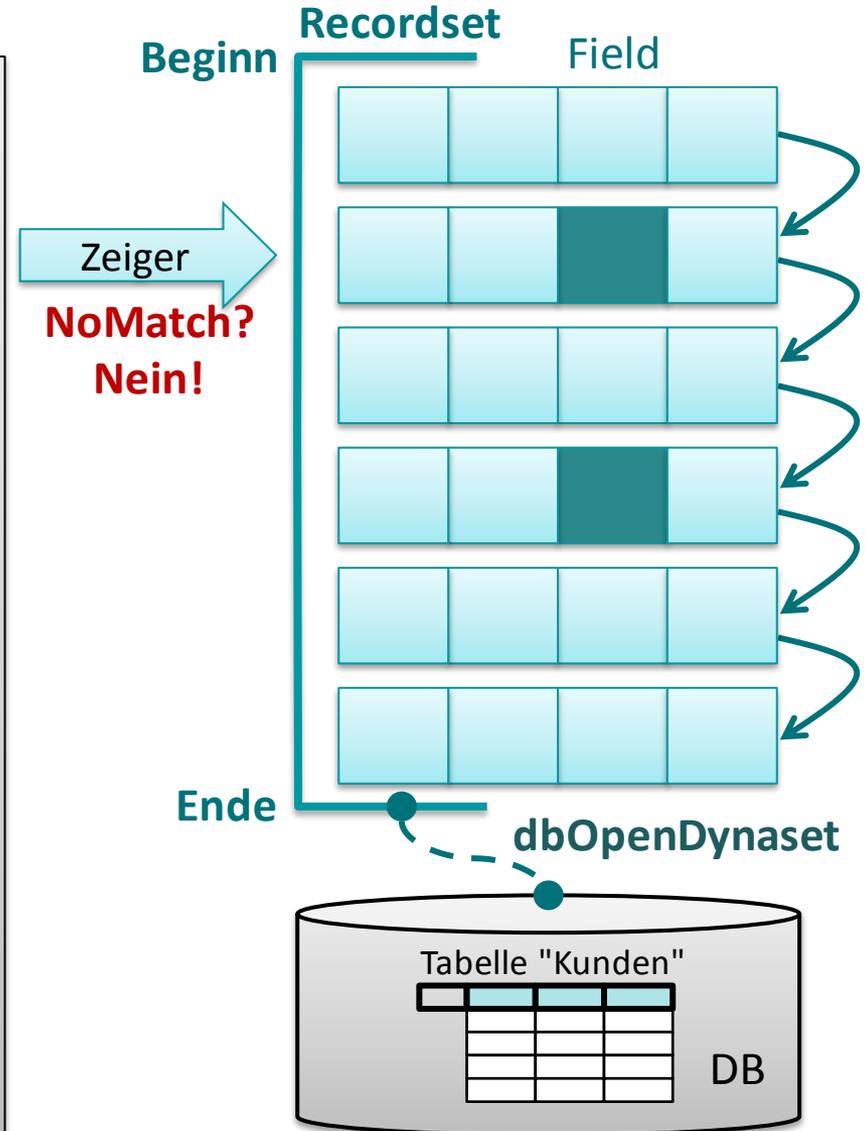
Modul

```
Option Compare Database
Option Explicit

Sub sucheKundenMeier()
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb
' Recordset deklarieren, initialisieren
Dim rcsKnd As Recordset
Set rcsKnd=db.OpenRecordset("tblKunden",
    dbOpenDynaset)
' Suchen nach Kunden "Meier"
rcsKnd.FindFirst("kndName = 'Meier'")

' Schleife bis zum Ende des Recordset
Do Until rcsKnd.NoMatch
' Name und Vorname ausgeben
Debug.Print rcsKnd.Fields("kndName") & _
    ", " & rcsKnd.Fields("kndVorname")

Loop
End Sub
```



Suchen und Finden im Recordset



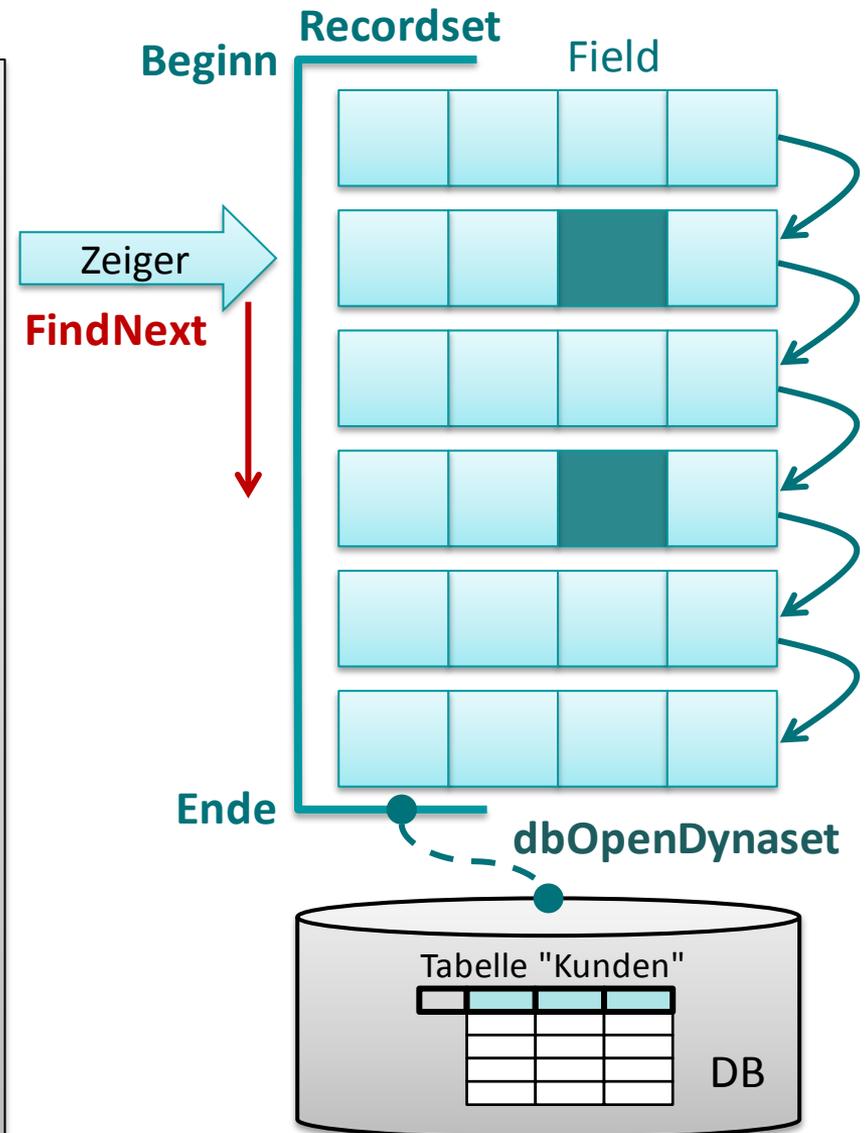
Modul

```
Option Compare Database
Option Explicit

Sub sucheKundenMeier()
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb
' Recordset deklarieren, initialisieren
Dim rcsKnd As Recordset
Set rcsKnd=db.OpenRecordset("tblKunden",
                           dbOpenDynaset)
' Suchen nach Kunden "Meier"
rcsKnd.FindFirst ("kndName = 'Meier'")

' Schleife bis zum Ende des Recordset
Do Until rcsKnd.NoMatch
' Name und Vorname ausgeben
Debug.Print rcsKnd.Fields("kndName") & _
            ", " & rcsKnd.Fields("kndVorname")
' Nächsten finden
rcsKnd.FindNext ("kndName = 'Meier'")

Loop
End Sub
```



Suchen und Finden im Recordset

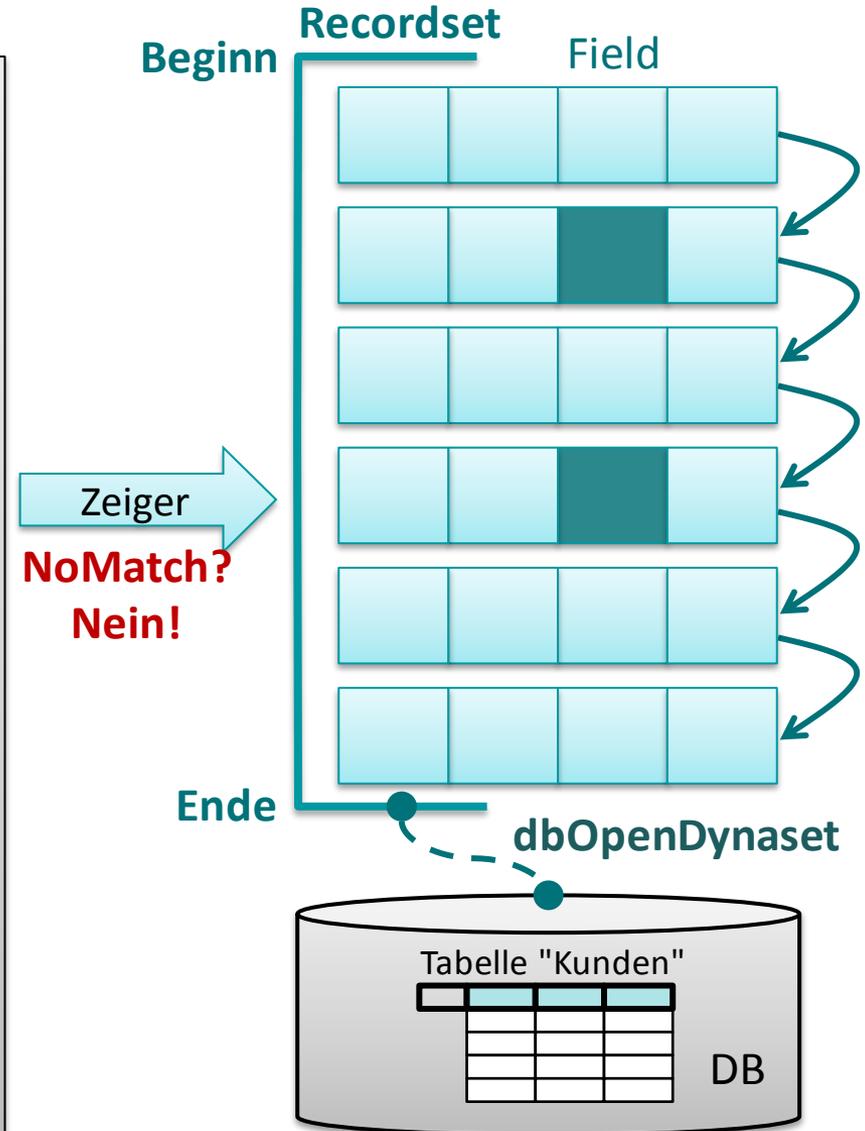


Modul

```
Option Compare Database
Option Explicit

Sub sucheKundenMeier()
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb
' Recordset deklarieren, initialisieren
Dim rcsKnd As Recordset
Set rcsKnd=db.OpenRecordset ("tblKunden",
                             dbOpenDynaset)
' Suchen nach Kunden "Meier"
rcsKnd.FindFirst ("kndName = 'Meier'")

' Schleife bis zum Ende des Recordset
Do Until rcsKnd.NoMatch
' Name und Vorname ausgeben
Debug.Print rcsKnd.Fields("kndName") & _
            ", " & rcsKnd.Fields("kndVorname")
' Nächsten finden
rcsKnd.FindNext ("kndName = 'Meier'")
' Weiterer Schleifendurchlauf
Loop
End Sub
```



Suchen und Finden im Recordset

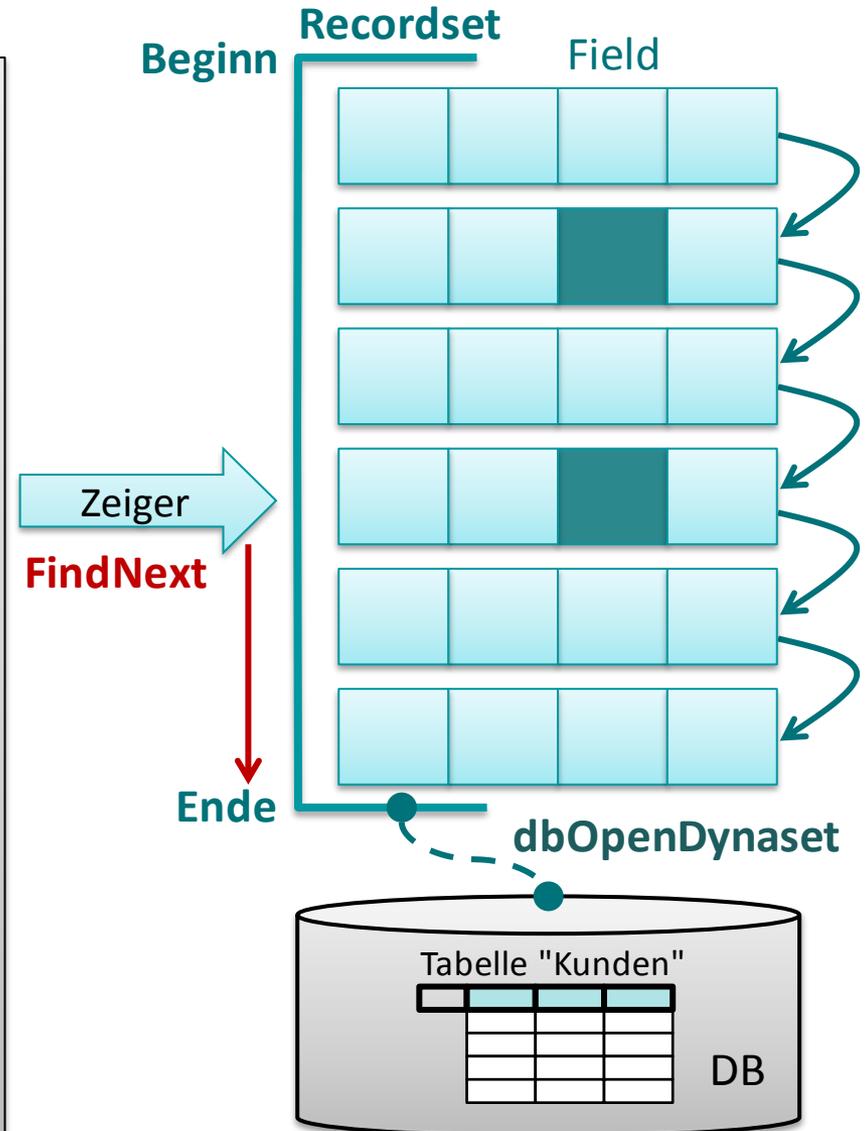


Modul

```
Option Compare Database
Option Explicit

Sub sucheKundenMeier()
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb
' Recordset deklarieren, initialisieren
Dim rcsKnd As Recordset
Set rcsKnd=db.OpenRecordset ("tblKunden",
                             dbOpenDynaset)
' Suchen nach Kunden "Meier"
rcsKnd.FindFirst ("kndName = 'Meier'")

' Schleife bis zum Ende des Recordset
Do Until rcsKnd.NoMatch
' Name und Vorname ausgeben
Debug.Print rcsKnd.Fields("kndName") & _
            ", " & rcsKnd.Fields("kndVorname")
' Nächsten finden
rcsKnd.FindNext ("kndName = 'Meier'")
' Weiterer Schleifendurchlauf
Loop
End Sub
```



Suchen und Finden im Recordset

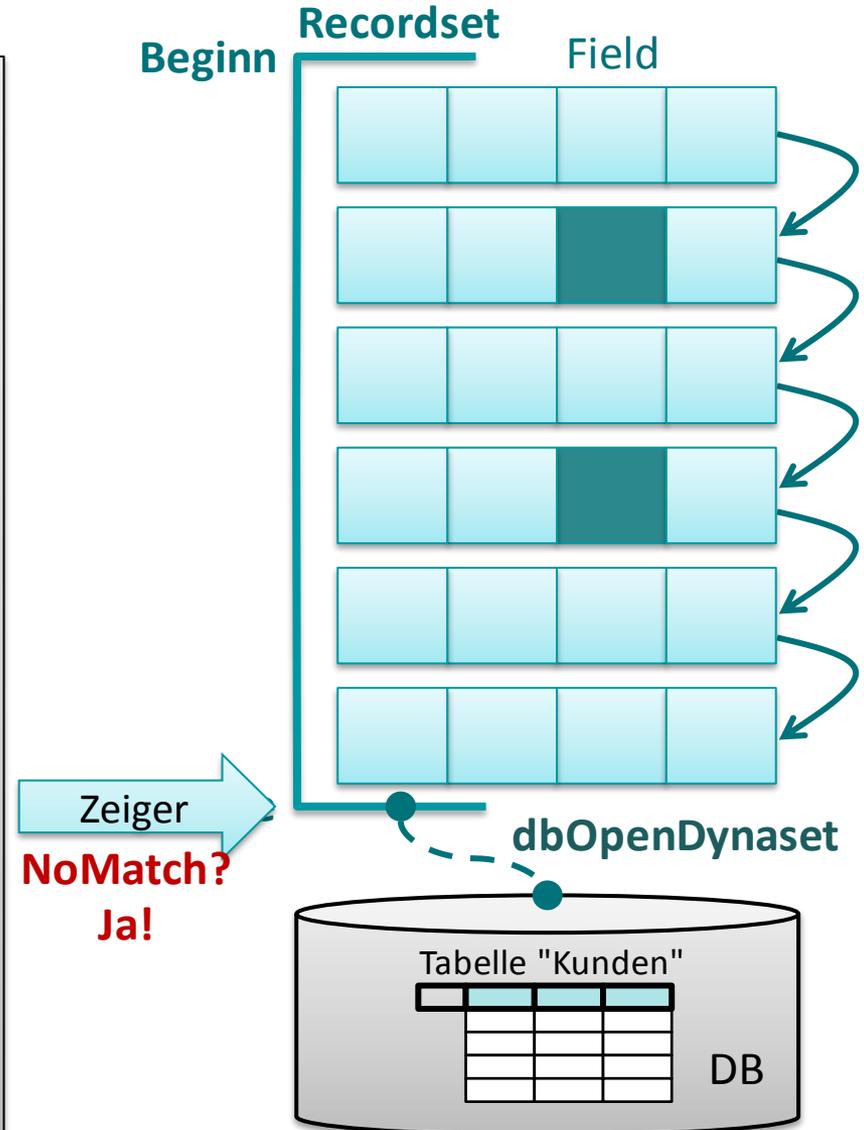


Modul

```
Option Compare Database
Option Explicit

Sub sucheKundenMeier()
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb
' Recordset deklarieren, initialisieren
Dim rcsKnd As Recordset
Set rcsKnd=db.OpenRecordset("tblKunden",
    dbOpenDynaset)
' Suchen nach Kunden "Meier"
rcsKnd.FindFirst ("kndName = 'Meier'")

' Schleife bis zum Ende des Recordset
Do Until rcsKnd.NoMatch
' Name und Vorname ausgeben
Debug.Print rcsKnd.Fields("kndName") & _
    ", " & rcsKnd.Fields("kndVorname")
' Nächsten finden
rcsKnd.FindNext ("kndName = 'Meier'")
' Weiterer Schleifendurchlauf
Loop
End Sub
```



Suchen und Finden im Recordset

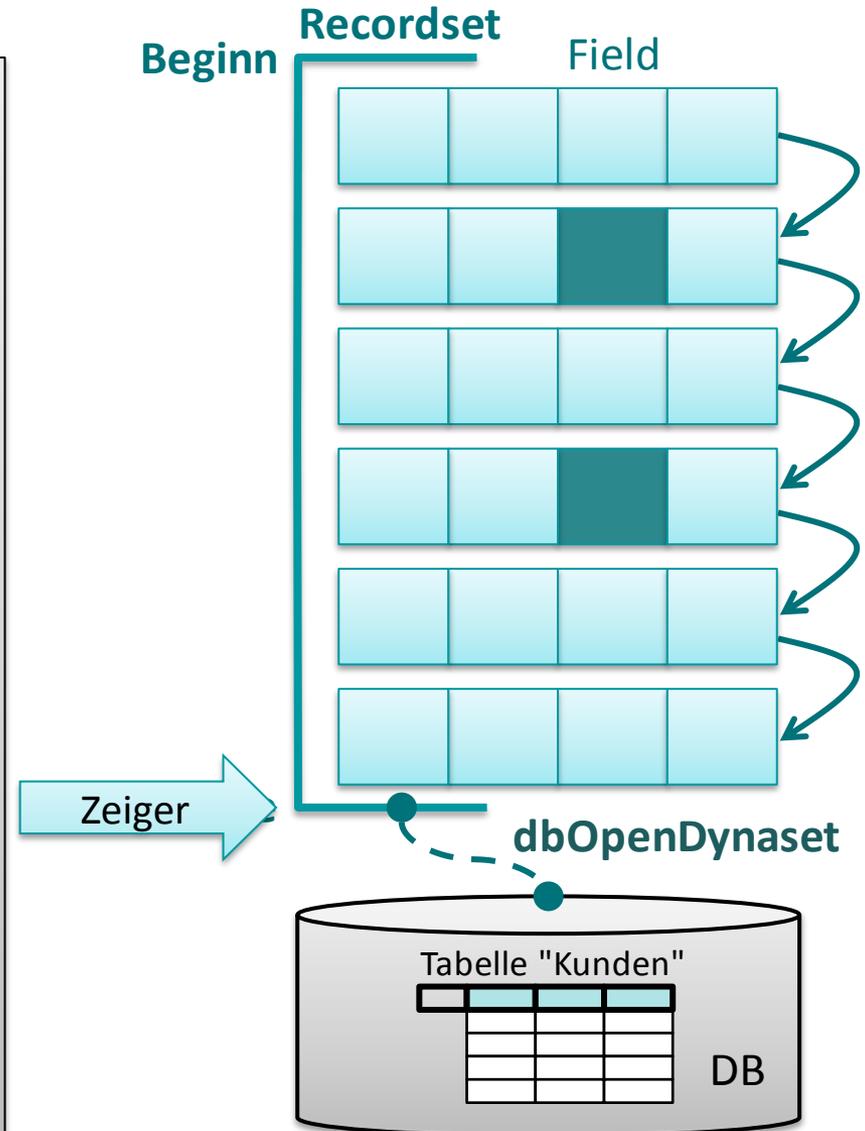


Modul

```
Option Compare Database
Option Explicit

Sub sucheKundenMeier()
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb
' Recordset deklarieren, initialisieren
Dim rcsKnd As Recordset
Set rcsKnd=db.OpenRecordset("tblKunden",
    dbOpenDynaset)
' Suchen nach Kunden "Meier"
rcsKnd.FindFirst ("kndName = 'Meier'")

' Schleife bis zum Ende des Recordset
Do Until rcsKnd.NoMatch
' Name und Vorname ausgeben
Debug.Print rcsKnd.Fields("kndName") & _
    ", " & rcsKnd.Fields("kndVorname")
' Nächsten finden
rcsKnd.FindNext ("kndName = 'Meier'")
' Weiterer Schleifendurchlauf
Loop
End Sub
```



Suchen und Finden im Recordset



Modul

```
(Allgemein) ausgebenAlleKunden

Option Compare Database
Option Explicit

Sub ausgebenKundenMeier()
' Recordset deklarieren
Dim rcsKnd As Recordset

' Verbindung zur Datenbank vorbereiten
Dim db As Database
Set db = CurrentDb

' Recordset initialisieren und dadurch füllen
Set rcsKnd = db.OpenRecordset("tblKunden", _
    dbOpenDynaset)

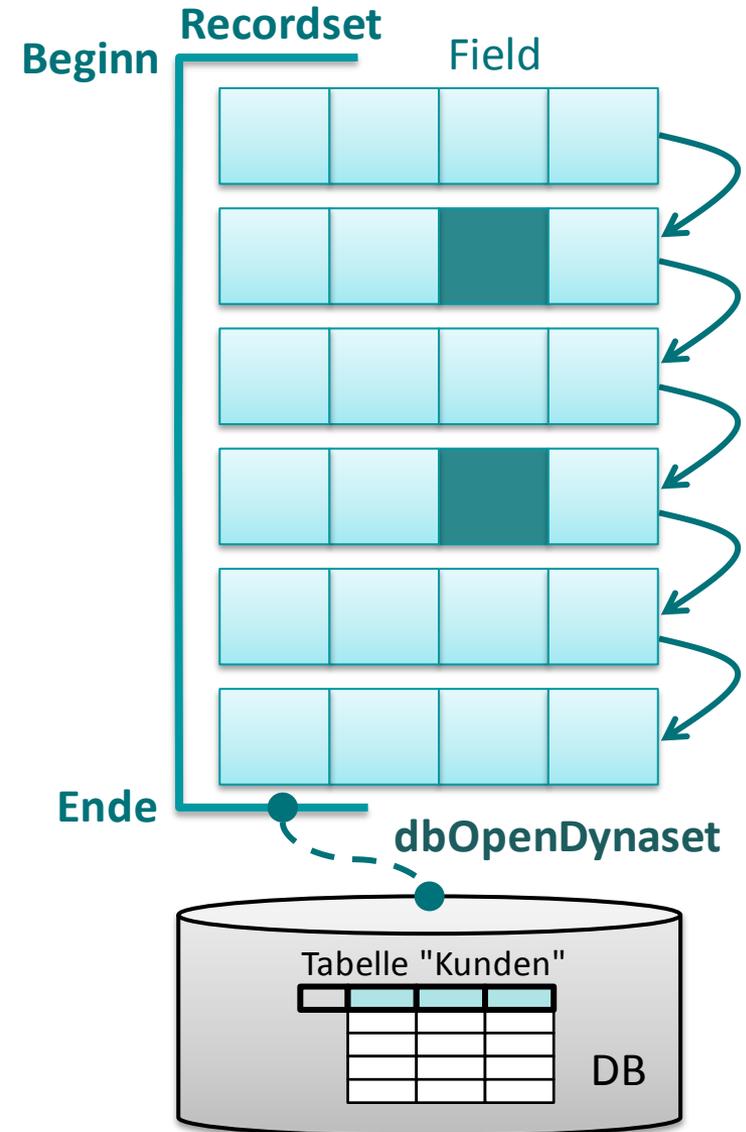
' Suchen nach Kunden "Meier"
rcsKnd.FindFirst ("kndName = 'Meier'")

' Schleife bis zum Ende des Recordset
Do Until rcsKnd.NoMatch

' Name und Vorname ausgeben
Debug.Print rcsKnd.Fields("kndName") & _
    ", " & rcsKnd.Fields("kndVorname")

' Nächsten finden
rcsKnd.FindNext ("kndName = 'Meier'")

Loop
End Sub
```



Suchen und Finden im Recordset



Modul

```
(Allgemein) ausgebenAlleKunden

Option Compare Database
Option Explicit

Sub ausgebenKundenMeier()
' Recordset deklarieren
Dim rcsKnd As Recordset

' Verbindung zur Datenbank vorbereiten
Dim db As Database
Set db = CurrentDb

' Recordset initialisieren und dadurch füllen
Set rcsKnd = db.OpenRecordset("tblKunden", _
    dbOpenDynaset)

' Suchen nach Kunden "Meier"
rcsKnd.FindFirst ("kndName = 'Meier'")

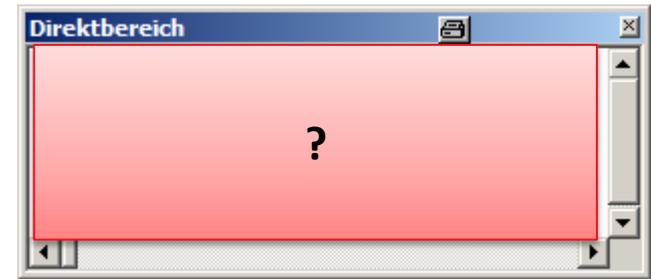
'Schleife bis zum Ende des Recordset
Do Until rcsKnd.NoMatch

' Name und Vorname ausgeben
Debug.Print rcsKnd.Fields("kndName") & _
    ", " & rcsKnd.Fields("kndVorname")

' Nächsten finden
rcsKnd.FindNext ("kndName = 'Meier'")

Loop

End Sub
```



tblKunden			
	kndIdPk	kndName	kndVorname
+	1	Meier	Thomas
+	2	Müller	Sven
+	3	Meier	Mike
+	4	Müller	Silke

Suchen und Finden im Recordset



Modul

```
(Allgemein) ausgebenAlleKunden

Option Compare Database
Option Explicit

Sub ausgebenKundenMeier()
' Recordset deklarieren
Dim rcsKnd As Recordset

' Verbindung zur Datenbank vorbereiten
Dim db As Database
Set db = CurrentDb

' Recordset initialisieren und dadurch füllen
Set rcsKnd = db.OpenRecordset("tblKunden", _
    dbOpenDynaset)

' Suchen nach Kunden "Meier"
rcsKnd.FindFirst ("kndName = 'Meier'")

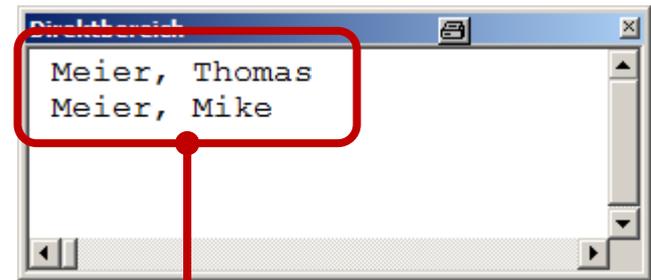
'Schleife bis zum Ende des Recordset
Do Until rcsKnd.NoMatch

' Name und Vorname ausgeben
Debug.Print rcsKnd.Fields("kndName") & _
    ", " & rcsKnd.Fields("kndVorname")

' Nächsten finden
rcsKnd.FindNext ("kndName = 'Meier'")

Loop

End Sub
```



tblKunden			
	kndIdPk	kndName	kndVorname
+	1	Meier	Thomas
+	2	Müller	Sven
+	3	Meier	Mike
+	4	Müller	Silke

Suchen und Finden im Recordset

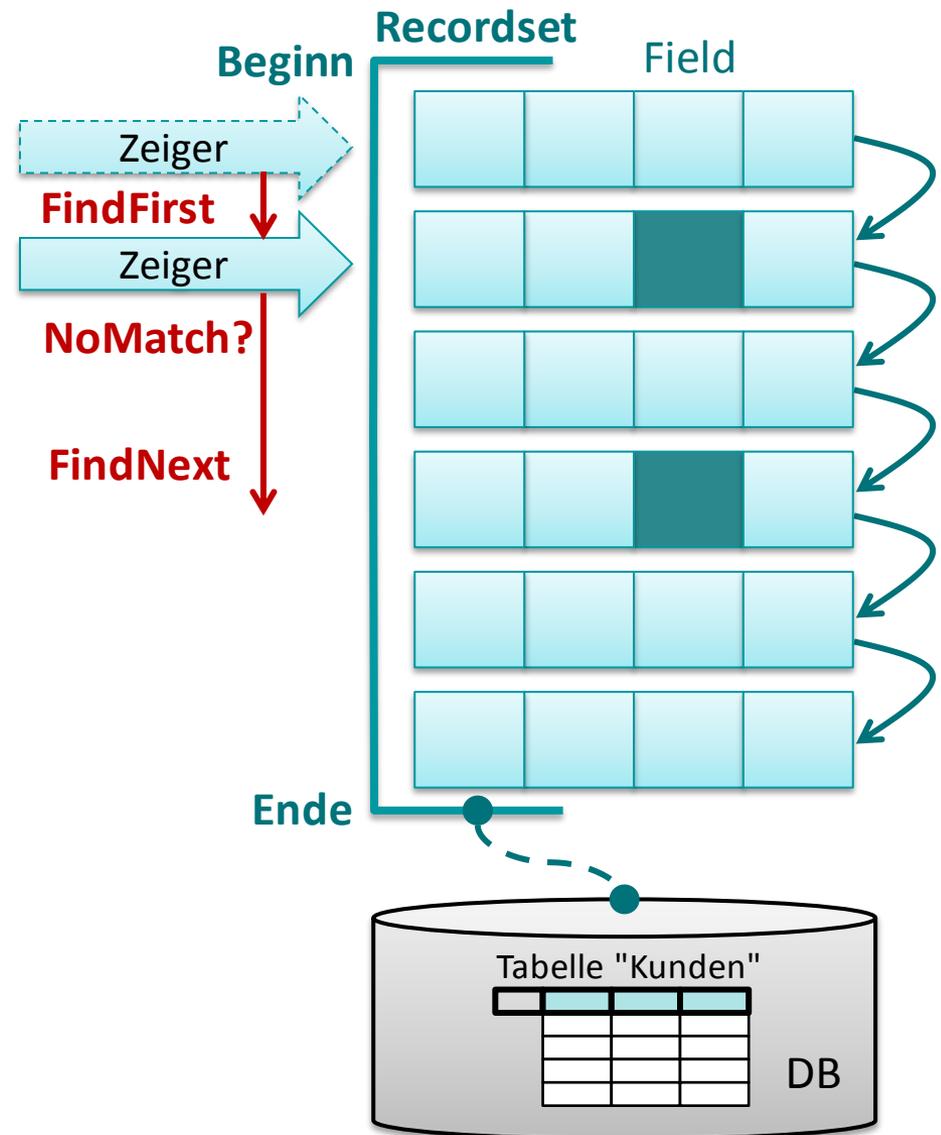


Such-/Findemöglichkeiten

- ersten Datensatz finden, der ein bestimmtes Kriterium erfüllt
`<einRecordset>.FindFirst`
- nächsten Datensatz finden, der ein bestimmtes Kriterium erfüllt
`<einRecordset>.FindNext`

Positionsbestimmung, ob Datensätze gefunden werden konnten

`<einRecordset>.NoMatch`



Ändern und Hinzufügen: Demo 07.02



D07.02: Verwenden Sie ein Recordset, um

- die Bezeichnung und den Preis aller Produkte im Direktbereich auszugeben, deren Preis größer als 2 EUR ist
- Ausgangspunkt Tabelle "tblProdukte" mit Spalte "prdPreis"

tblProdukte	tblProduktKategorie	prdIdPk	prdBezeichnung	prdBeschreibung	prdPreis
		1	Blumenzwiebeln Sorte XY	Rote Blumen, sehr schön	0,59 €
		2	Blumensamen, Sorte ABC	Gelbe Blumen, klein	0,99 €
		3	Sonnenblumensamen, Sorte XXL	Große gelbe Blüten, sehr schön	1,99 €
		4	Rasendünger Maxi	5kg, Lanzeitdünger	5,99 €
		5	Rasendünger Mega	10kg, Sofortwirkung	9,99 €
		6	Blumendünger "Sabine"	500 ml, Grün- und Zimmerpflanzen	0,99 €
		7	Blumendünger "Dörte"	500 ml, Bio-Dünger	2,99 €
		8	Rindenmulch fein	6 kg, feiner Mulch	2,49 €
		9	Rindenmulch grob	6 kg, grober Mulch	2,69 €
		10	Rasenbesen "Bert"	Markenprodukt, lange Haltbarkeit	5,99 €



Inhalt

Ziel und Einordnung

Rückblick

Ausgangssituation

Datenbankschnittstelle

Recordsets

- Was sind Recordsets
- Mit Recordsets Daten lesen
- Navigation im Recordset per Zeiger
- Suchen und Finden im Recordset
- Ändern und Hinzufügen im Recordset
- Kombination mit grafischen Abfragen und SQL

Weitere VBA-Funktionen

Fallstricke

Zusammenfassung und Ausblick



Ändern und Hinzufügen im Recordset



Bearbeitungsmodus

- Änderungsmodus aktivieren
`<einRecordset>.Edit`
- Einfügemodus aktivieren
`<einRecordset>.AddNew`
- Bearbeitungsmodus abschließen
`<einRecordset>.Update`
- Löschmodus aktivieren
`<einRecordset>.Delete`

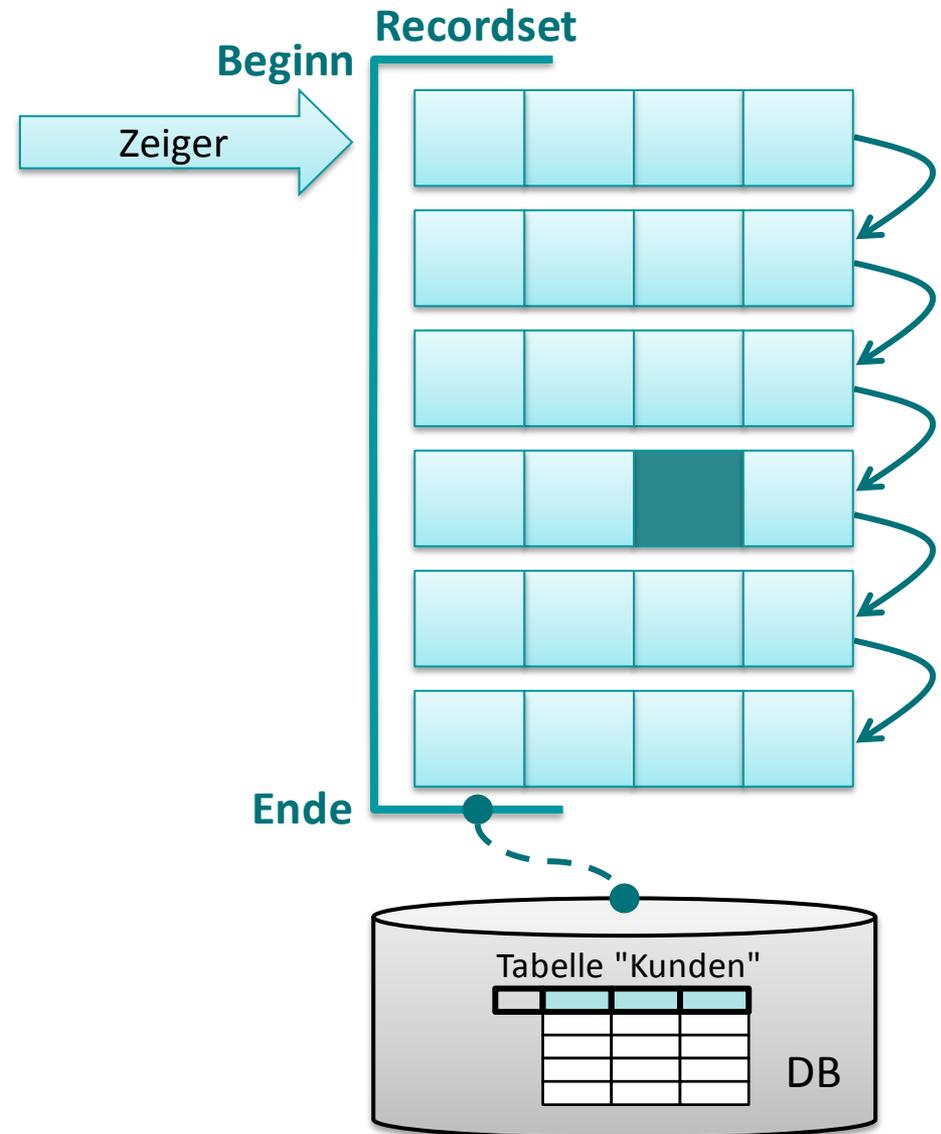
Ändern und Hinzufügen im Recordset



Ziel: Ein Datensatz des Recordset soll geändert werden

Ansatz

- Recordset als Dynaset füllen



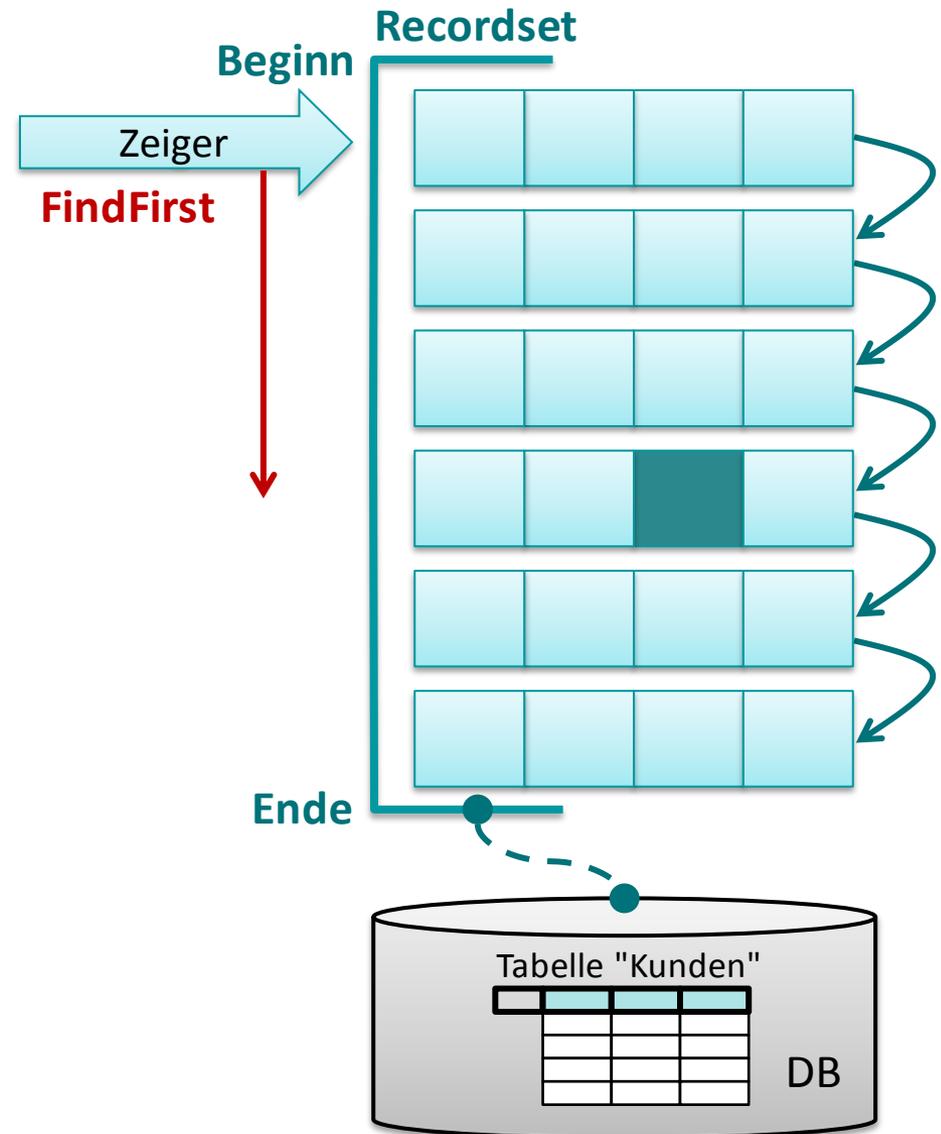
Ändern und Hinzufügen im Recordset



Ziel: Ein Datensatz des Recordset soll geändert werden

Ansatz

- Recordset als Dynaset füllen
- nach zu änderndem Datensatz suchen



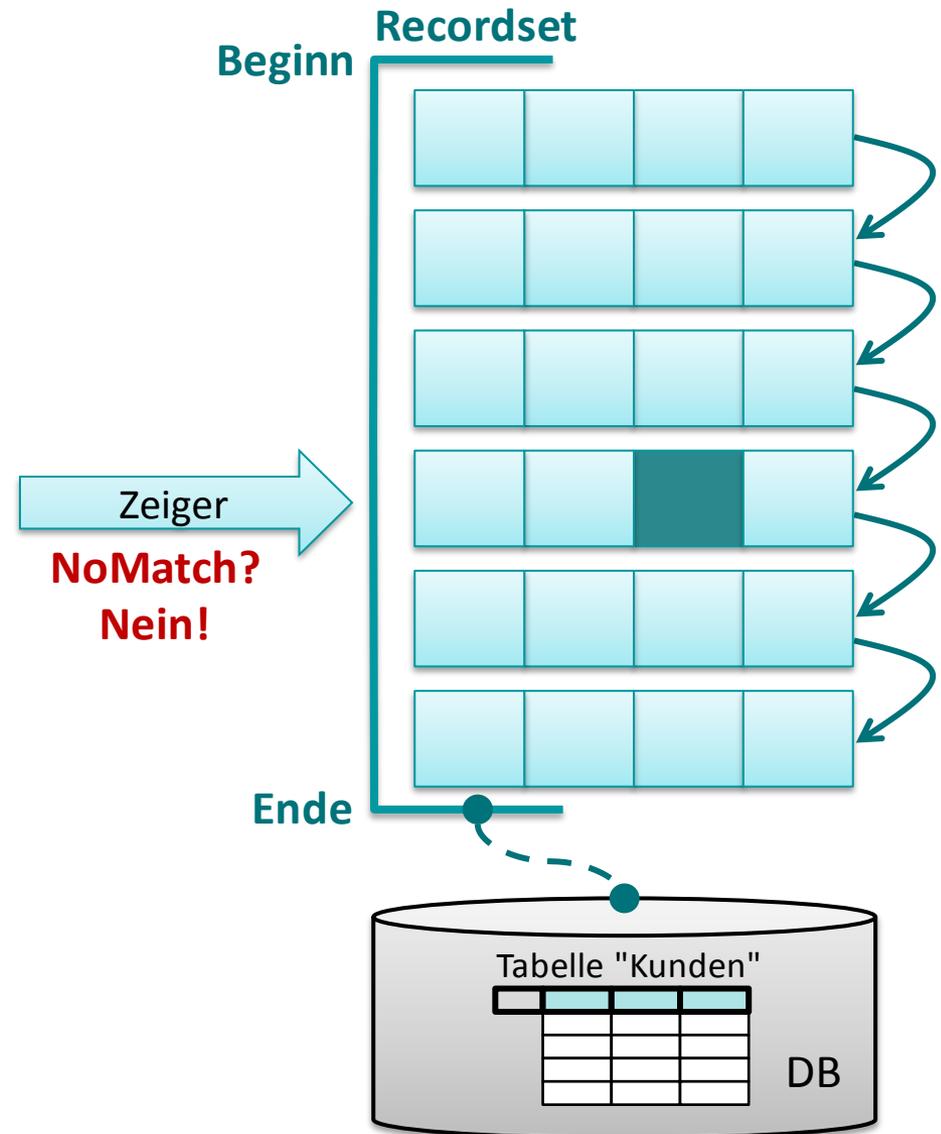
Ändern und Hinzufügen im Recordset



Ziel: Ein Datensatz des Recordset soll geändert werden

Ansatz

- Recordset als Dynaset füllen
- nach zu änderndem Datensatz suchen
- Prüfen, ob Datensatz gefunden



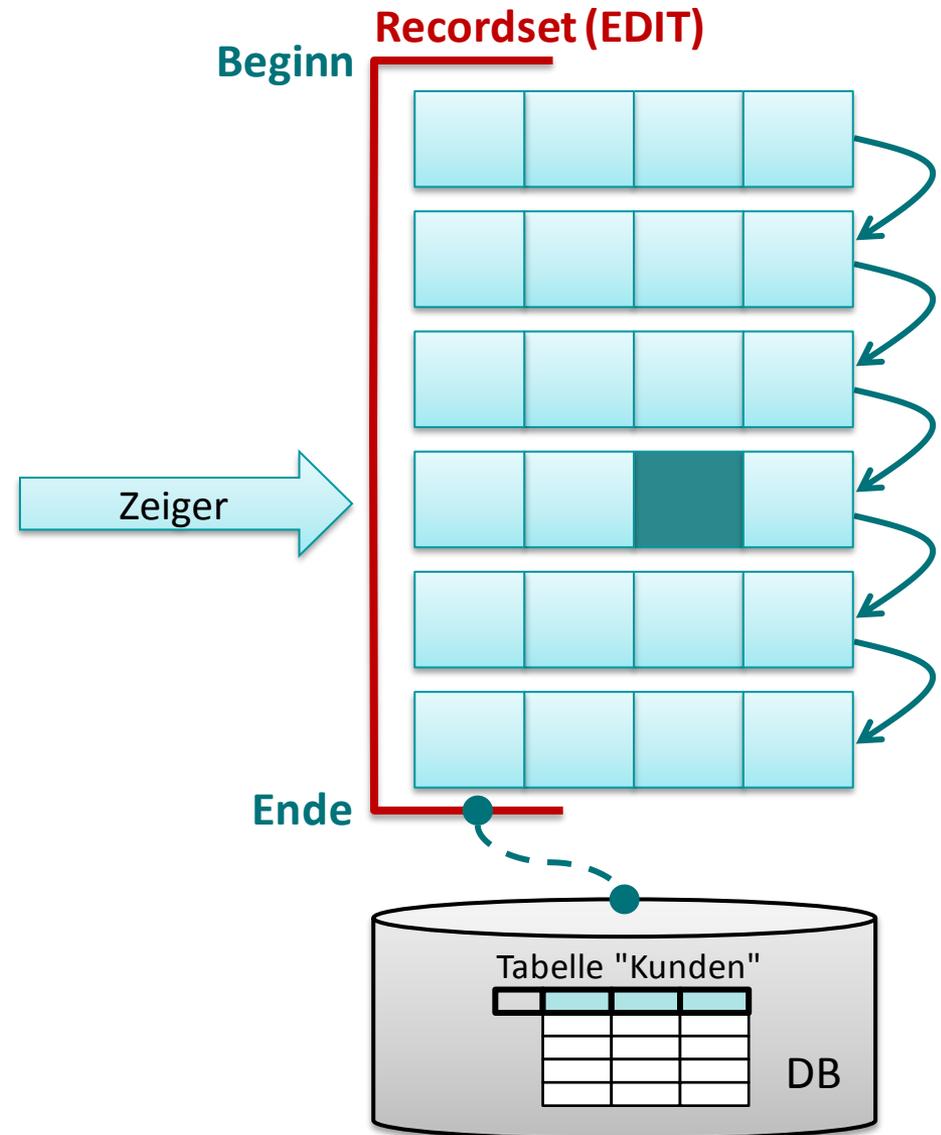
Ändern und Hinzufügen im Recordset



Ziel: Ein Datensatz des Recordset soll geändert werden

Ansatz

- Recordset als Dynaset füllen
- nach zu änderndem Datensatz suchen
- Prüfen, ob Datensatz gefunden
- Wenn ja, dann Änderungsmodus des Recordset aktivieren



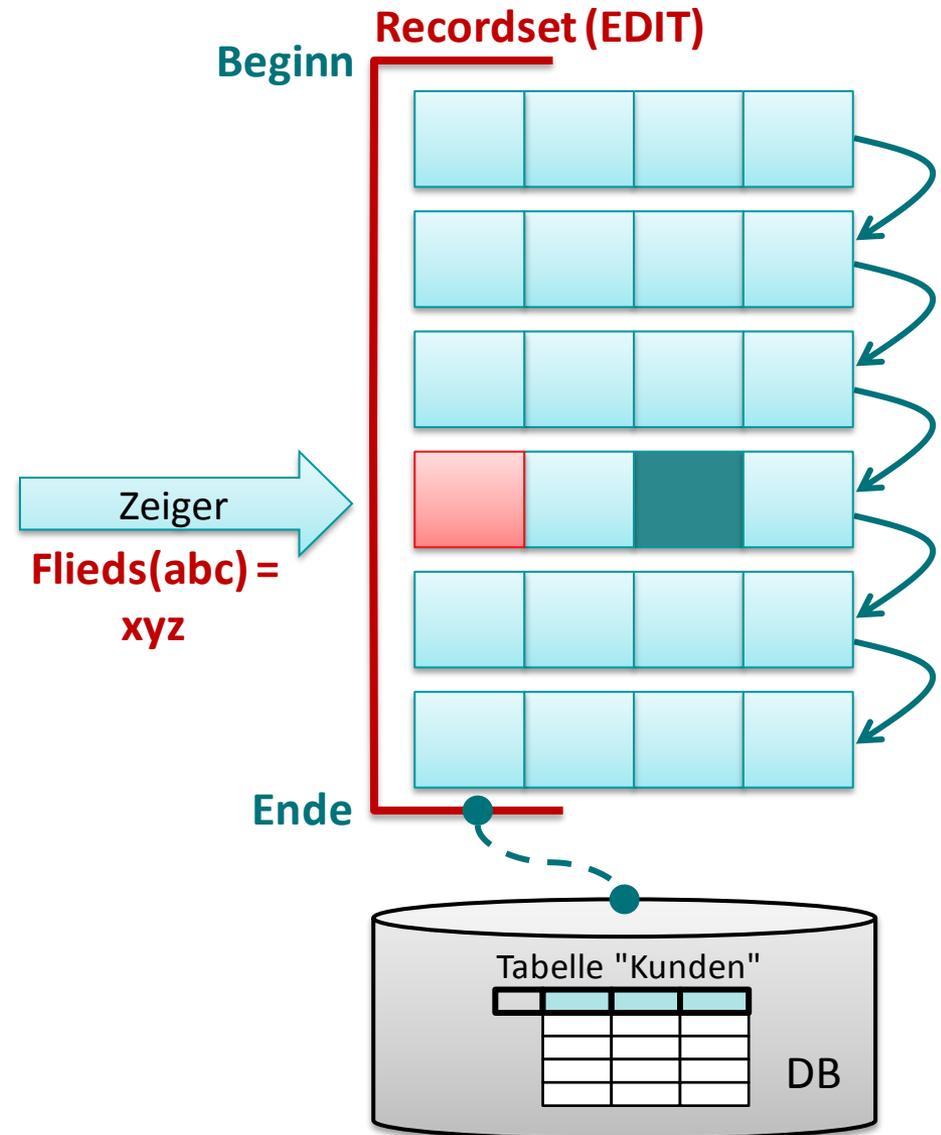
Ändern und Hinzufügen im Recordset



Ziel: Ein Datensatz des Recordset soll geändert werden

Ansatz

- Recordset als Dynaset füllen
- nach zu änderndem Datensatz suchen
- Prüfen, ob Datensatz gefunden
- Wenn ja, dann Änderungsmodus des Recordset aktivieren
- Änderung des Feldes vornehmen



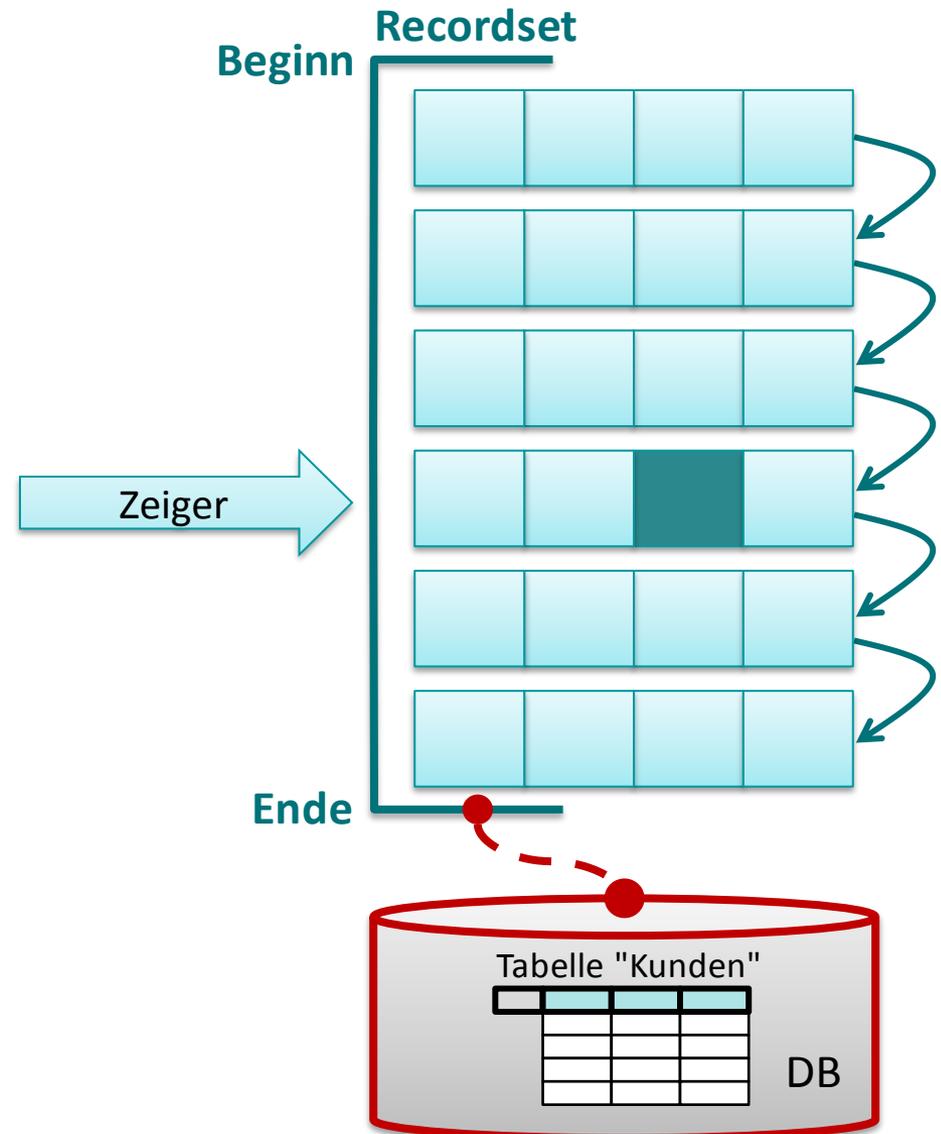
Ändern und Hinzufügen im Recordset



Ziel: Ein Datensatz des Recordset soll geändert werden

Ansatz

- Recordset als Dynaset füllen
- nach zu änderndem Datensatz suchen
- Prüfen, ob Datensatz gefunden
- Wenn ja, dann Änderungsmodus des Recordset aktivieren
- Änderung des Feldes vornehmen
- Aktualisierung abschließen



Ändern und Hinzufügen im Recordset



Ziel: Ein Datensatz des Recordset soll geändert werden

Ansatz

- Recordset als Dynaset füllen
- nach zu änderndem Datensatz suchen
- Prüfen, ob Datensatz gefunden
- Wenn ja, dann Änderungsmodus des Recordset aktivieren
- Änderung des Feldes vornehmen
- Aktualisierung abschließen

Modul

```
Option Compare Database
Option Explicit

Sub aendernKunde()
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb
' Recordset deklarieren, initialisieren
Dim rcsKnd As Recordset
Set rcsKnd=db.OpenRecordset("tblKunden",
                             dbOpenDynaset)
' Suchen nach Kunde mit ID4
rcsKnd.FindFirst ("kndIdPk = 4")
If rcsKnd.NoMatch Then
    Debug.Print "Kein Datensatz!"
Else
    ' Änderungsmodus aktivieren
    rcsKnd.Edit
    ' Änderung vornehmen
    rcsKnd.Fields("kndName") = "Schmidt"
    ' Aktualisierung durchführen
    rcsKnd.Update
End If
End Sub
```

Ändern und Hinzufügen im Recordset



Vorher

tblKunden			
	kndIdPk	kndName	kndVornam
+	1	Meier	Thomas
+	2	Müller	Sven
+	3	Meier	Mike
+	4	Müller	Silke

Nachher



```
(Allgemein) | ausgebenKundenMeier
Option Compare Database
Option Explicit

Sub aendernKunde()
' Recordset deklarieren
Dim rcsKnd As Recordset

' Verbindung zur Datenbank vorbereiten
Dim db As Database
Set db = CurrentDb

' Recordset initialisieren und dadurch füllen
Set rcsKnd = db.OpenRecordset("tblKunden", _
                             dbOpenDynaset)

' Suchen nach Kunde mit ID4
rcsKnd.FindFirst ("kndIdPk = 4")

'Schleife bis zum Ende des Recordset
If rcsKnd.NoMatch Then
    Debug.Print "Keinen Datensatz gefunden."
Else
    ' Änderungsmodus aktivieren
    rcsKnd.Edit
    ' Änderung vornehmen
    rcsKnd.Fields("kndName") = "Schmidt"
    ' Aktualisierung durchführen
    rcsKnd.Update
End If
End Sub
```

Ändern und Hinzufügen im Recordset



Vorher

tblKunden			
	kndIdPk	kndName	kndVorname
+	1	Meier	Thomas
+	2	Müller	Sven
+	3	Meier	Mike
+	4	Müller	Silke

Nachher

tblKunden			
	kndIdPk	kndName	kndVorname
+	1	Meier	Thomas
+	2	Müller	Sven
+	3	Meier	Mike
+	4	Schmidt	Silke

```
(Allgemein) | ausgebenKundenMeier
Option Compare Database
Option Explicit

Sub aendernKunde()
    ' Recordset deklarieren
    Dim rcsKnd As Recordset

    ' Verbindung zur Datenbank vorbereiten
    Dim db As Database
    Set db = CurrentDb

    ' Recordset initialisieren und dadurch füllen
    Set rcsKnd = db.OpenRecordset("tblKunden", _
        dbOpenDynaset)

    ' Suchen nach Kunde mit ID4
    rcsKnd.FindFirst ("kndIdPk = 4")

    'Schleife bis zum Ende des Recordset
    If rcsKnd.NoMatch Then
        Debug.Print "Keinen Datensatz gefunden."
    Else
        ' Änderungsmodus aktivieren
        rcsKnd.Edit
        ' Änderung vornehmen
        rcsKnd.Fields("kndName") = "Schmidt"
        ' Aktualisierung durchführen
        rcsKnd.Update
    End If
End Sub
```

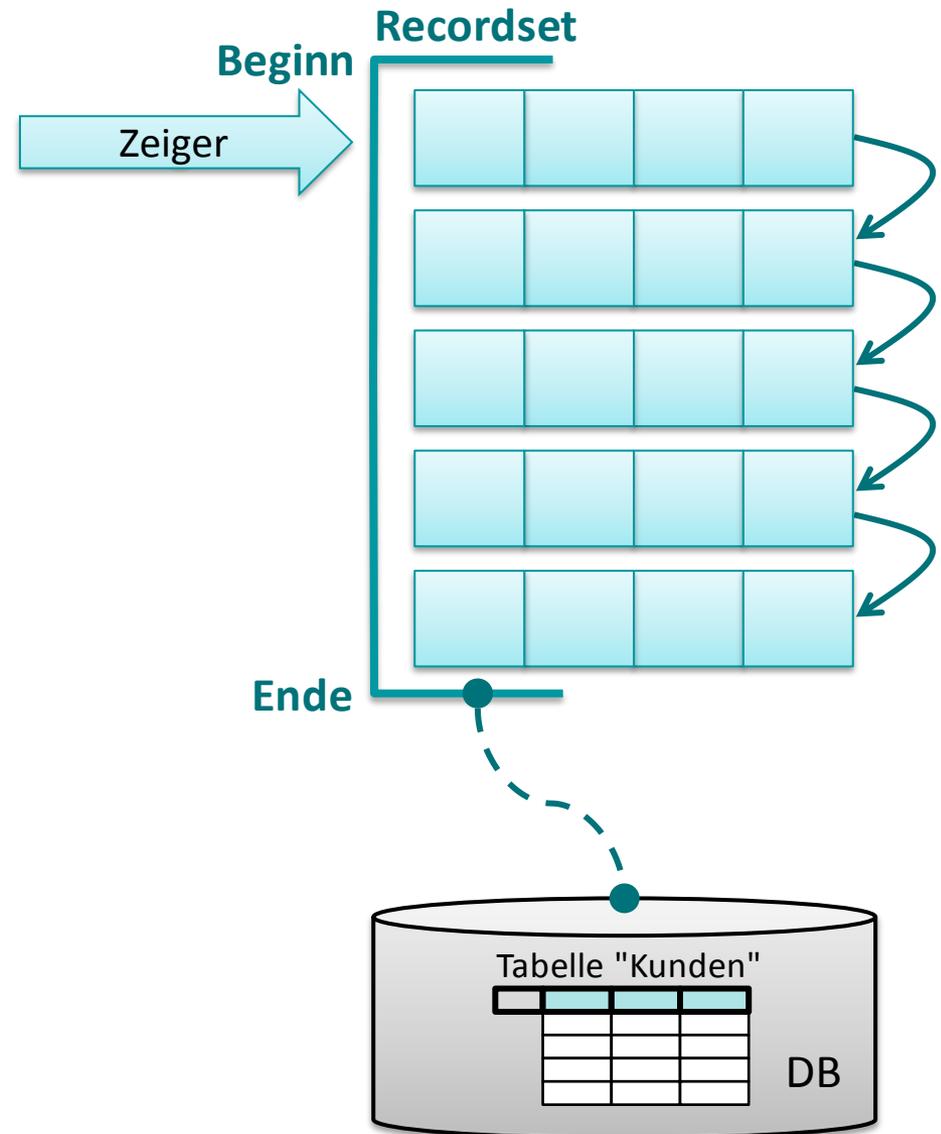
Ändern und Hinzufügen im Recordset



**Ziel: Ein Datensatz soll
Hinzugefügt werden**

Ansatz

- Recordset als Dnyaset befüllen
- Anfügemodus des Recordset aktivieren
- Neuen Eintrag feldweise belegen (Reihenfolge egal)
- Aktualisierung abschließen



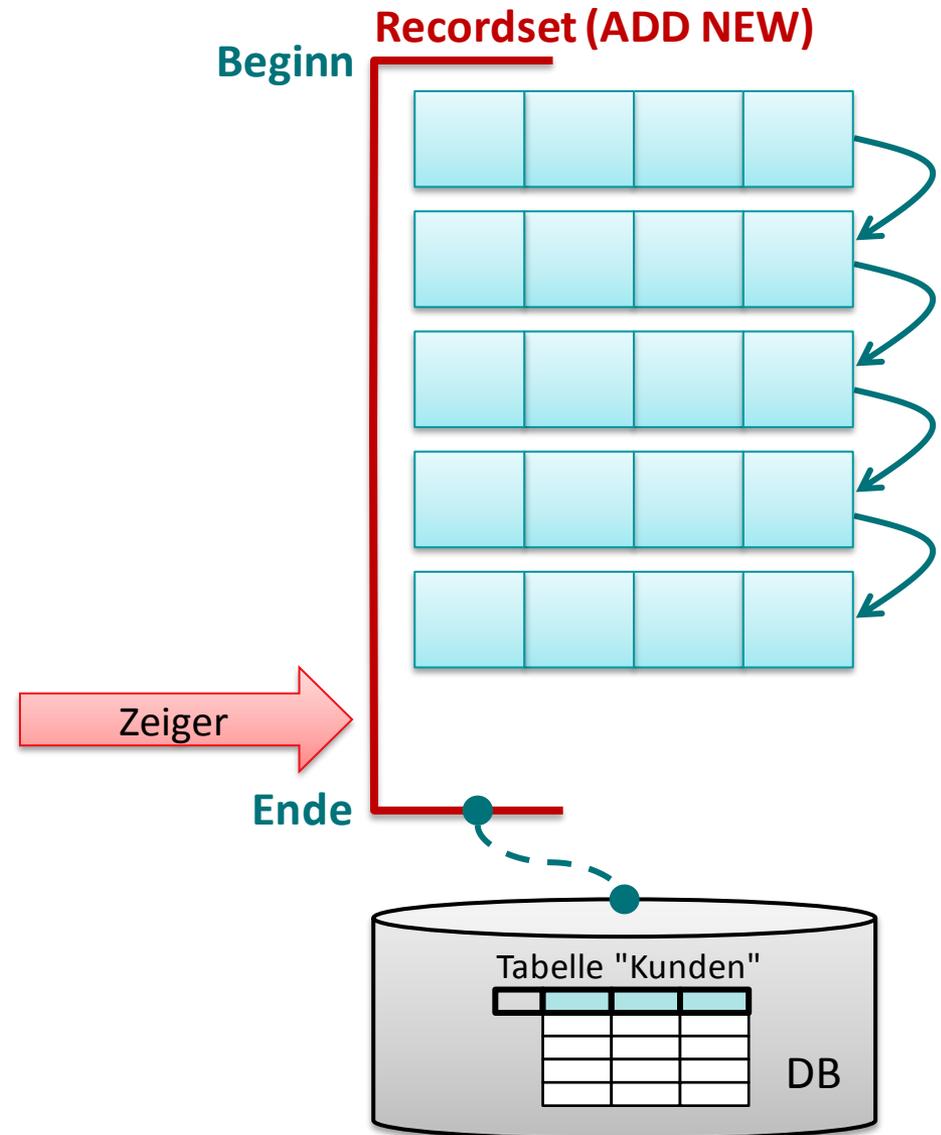
Ändern und Hinzufügen im Recordset



Ziel: Ein Datensatz soll Hinzugefügt werden

Ansatz

- Recordset als Dnyaset befüllen
- Anfügemodus des Recordset aktivieren
- Neuen Eintrag feldweise belegen (Reihenfolge egal)
- Aktualisierung abschließen



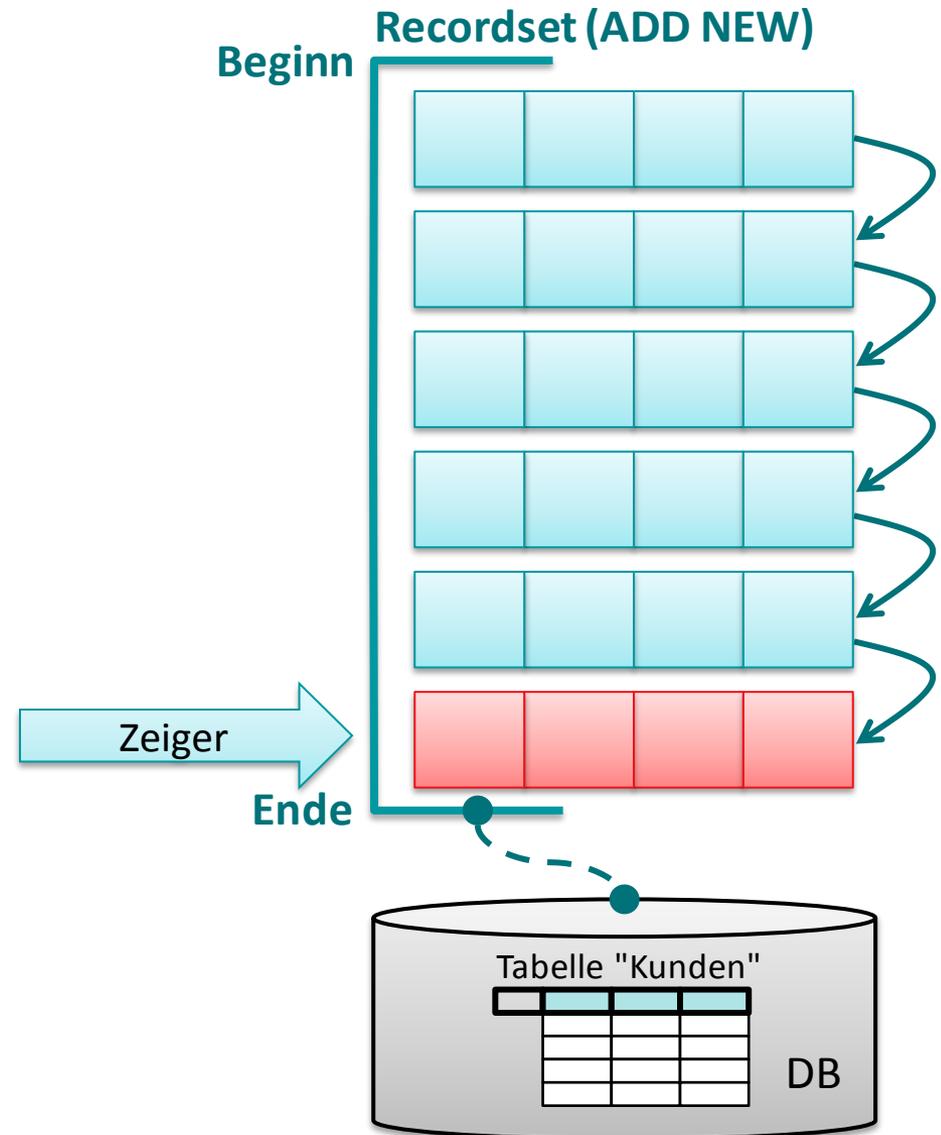
Ändern und Hinzufügen im Recordset



**Ziel: Ein Datensatz soll
Hinzugefügt werden**

Ansatz

- Recordset als Dnyaset befüllen
- Anfügemodus des Recordset aktivieren
- Neuen Eintrag feldweise belegen (Reihenfolge egal)
- Aktualisierung abschließen



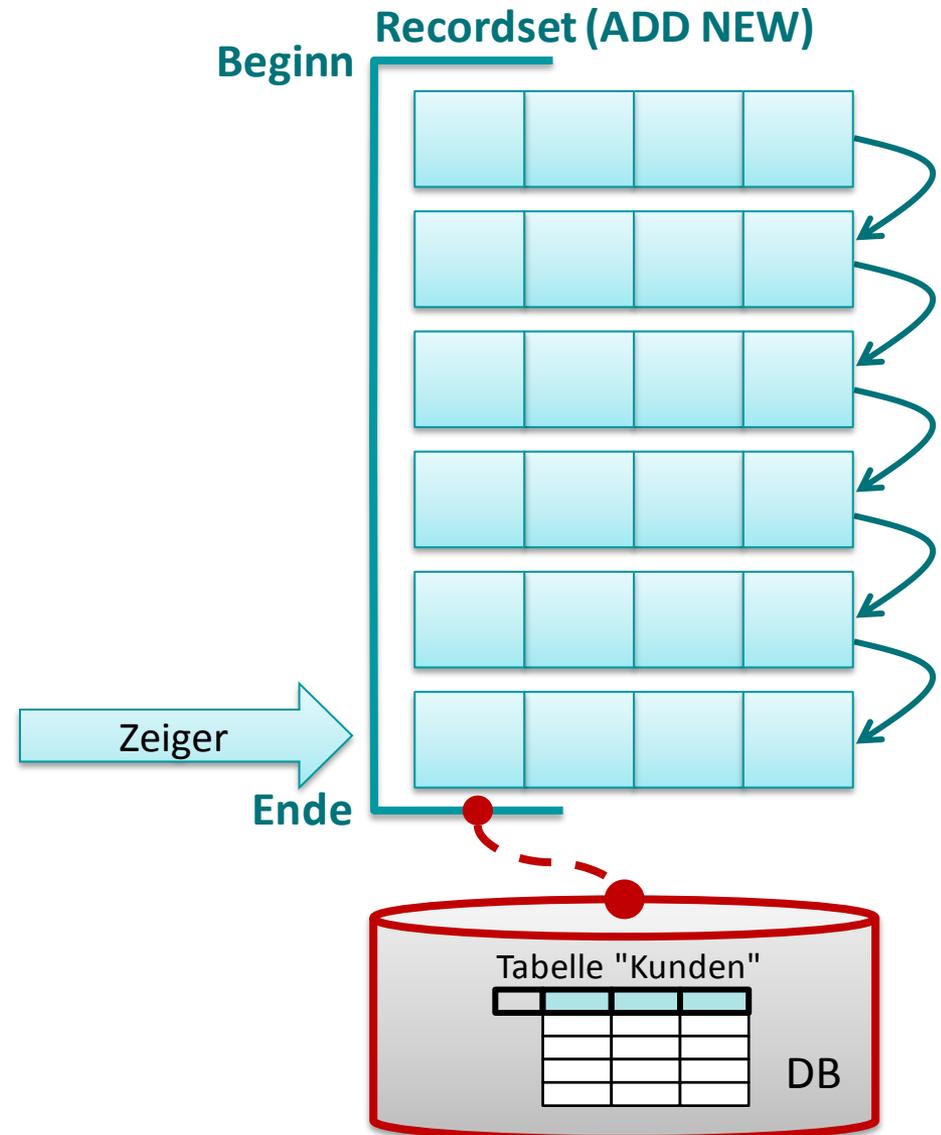
Ändern und Hinzufügen im Recordset



Ziel: Ein Datensatz soll Hinzugefügt werden

Ansatz

- Recordset als Dnyaset befüllen
- Anfügemodus des Recordset aktivieren
- Neuen Eintrag feldweise belegen (Reihenfolge egal)
- Aktualisierung abschließen



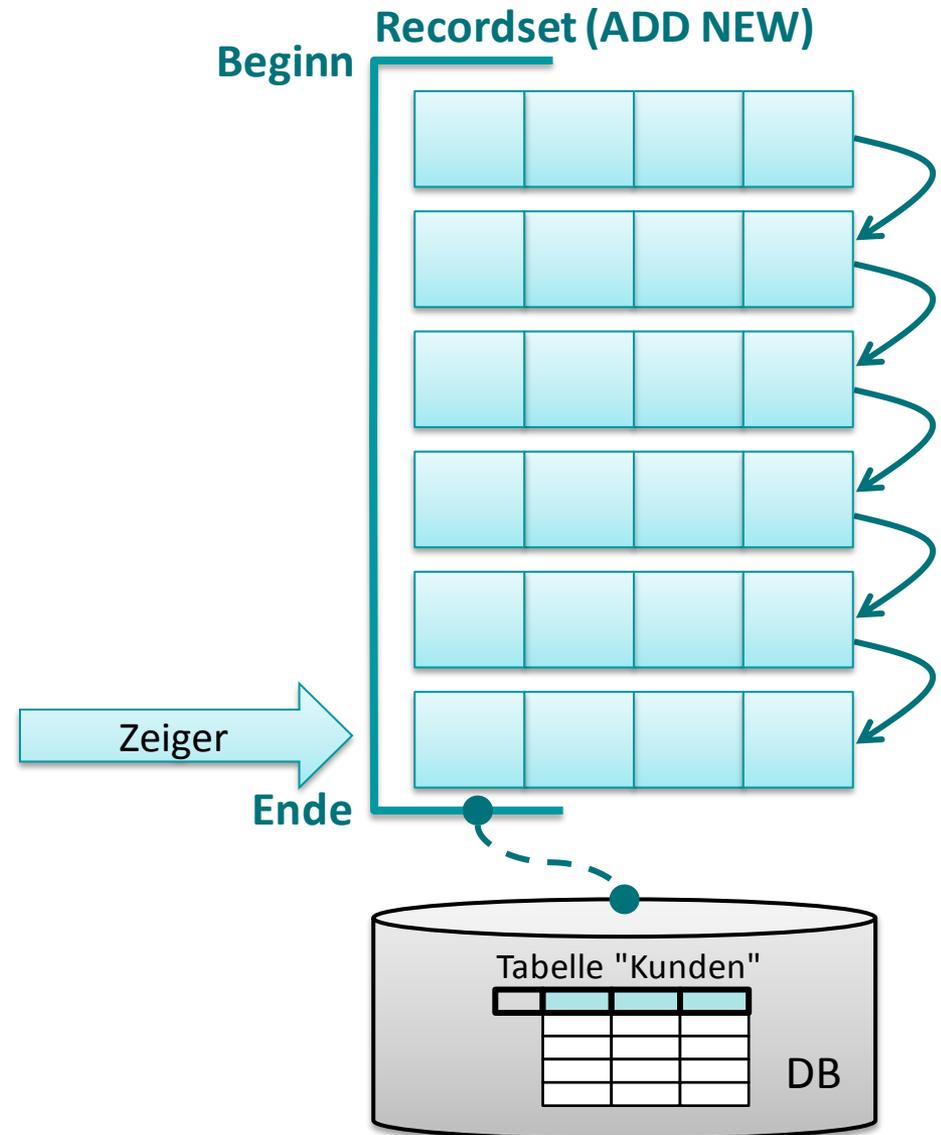
Ändern und Hinzufügen im Recordset



Ziel: Ein Datensatz soll Hinzugefügt werden

Ansatz

- Recordset als Dnyaset befüllen
- Anfügemodus des Recordset aktivieren
- Neuen Eintrag feldweise belegen (Reihenfolge egal)
- Aktualisierung abschließen



Ändern und Hinzufügen im Recordset



**Ziel: Ein Datensatz soll
Hinzugefügt werden**

Ansatz

- Recordset als Dnyaset befüllen
- Anfügemodus des Recordset aktivieren
- Neuen Eintrag feldweise belegen (Reihenfolge egal)
- Aktualisierung abschließen

Modul

```
Option Compare Database
Option Explicit

Sub hinzufuegenKunde()
' Verbindung zur DB vorbereiten
Dim db As Database
Set db = CurrentDb
' Recordset deklarieren, initialisieren
Dim rcsKnd As Recordset
Set rcsKnd=db.OpenRecordset("tblKunden",
                             dbOpenDynaset)

' Anfügemodus aktivieren
rcsKnd.AddNew

' Felder des neuen Eintrags belegen
rcsKnd.Fields("kndName") = "Schmidt"
rcsKnd.Fields("kndVorname") = "Udo"

' Aktualisierung durchführen
rcsKnd.Update

End Sub
```

Ändern und Hinzufügen im Recordset



Vorher

tblKunden				
	kndIdPk	kndName	kndVorname	Zu
+	1	Meier	Thomas	
+	2	Müller	Sven	
+	3	Meier	Mike	
+	4	Schmidt	Silke	
*	(Neu)			

```
(Allgemein) hinzufuegenKunde

Option Compare Database
Option Explicit

Sub hinzufuegenKunde()
' Recordset deklarieren
Dim rcsKnd As Recordset

' Verbindung zur Datenbank vorbereiten
Dim db As Database
Set db = CurrentDb

' Recordset initialisieren und dadurch füllen
Set rcsKnd = db.OpenRecordset("tblKunden", _
    dbOpenDynaset)

' Anfügemodus aktivieren
rcsKnd.AddNew
' Felder belegen
rcsKnd.Fields("kndName") = "Schmidt"
rcsKnd.Fields("kndVorname") = "Udo"

' Aktualisierung durchführen
rcsKnd.Update

End Sub
```

Nachher



Ändern und Hinzufügen im Recordset



Vorher

tblKunden			
	kndIdPk	kndName	kndVorname
+	1	Meier	Thomas
+	2	Müller	Sven
+	3	Meier	Mike
+	4	Schmidt	Silke
*	(Neu)		

```
(Allgemein) hinzufuegenKunde

Option Compare Database
Option Explicit

Sub hinzufuegenKunde()
' Recordset deklarieren
Dim rcsKnd As Recordset

' Verbindung zur Datenbank vorbereiten
Dim db As Database
Set db = CurrentDb

' Recordset initialisieren und dadurch füllen
Set rcsKnd = db.OpenRecordset("tblKunden", _
    dbOpenDynaset)

' Anfügemodus aktivieren
rcsKnd.AddNew
' Felder belegen
rcsKnd.Fields("kndName") = "Schmidt"
rcsKnd.Fields("kndVorname") = "Udo"

' Aktualisierung durchführen
rcsKnd.Update

End Sub
```

Nachher

tblKunden			
	kndIdPk	kndName	kndVorname
+	1	Meier	Thomas
+	2	Müller	Sven
+	3	Meier	Mike
+	4	Schmidt	Silke
+	6	Schmidt	Udo
*	(Neu)		



Ändern und Hinzufügen im Recordset

Bearbeitungsmodus

- Änderungsmodus aktivieren
`<einRecordset>.Edit`
- Einfügemodus aktivieren
`<einRecordset>.AddNew`
- Bearbeitungsmodus abschließen
`<einRecordset>.Update`
- Löschmodus aktivieren
`<einRecordset>.Delete`

Ändern und Hinzufügen: Demo 07.03



D07.03: Verwenden Sie ein Recordset, um

- die Preise aller Produkte in Tabelle "tblProdukte" um 0,50 EUR zu erhöhen
- Ausgangspunkt Tabelle "tblProdukte" mit Spalte "prdPreis"

tblProdukte	tblProduktKategorie		
prdIdPk	prdBezeichnung	prdBeschreibung	prdPreis
1	Blumenzwiebeln Sorte XY	Rote Blumen, sehr schön	0,59 €
2	Blumensamen, Sorte ABC	Gelbe Blumen, klein	0,99 €
3	Sonnenblumensamen, Sorte XXL	Große gelbe Blüten, sehr schön	1,99 €
4	Rasendünger Maxi	5kg, Lanzeitdünger	5,99 €
5	Rasendünger Mega	10kg, Sofortwirkung	9,99 €
6	Blumendünger "Sabine"	500 ml, Grün- und Zimmerpflanzen	0,99 €
7	Blumendünger "Dörte"	500 ml, Bio-Dünger	2,99 €
8	Rindenmulch fein	6 kg, feiner Mulch	2,49 €
9	Rindenmulch grob	6 kg, grober Mulch	2,69 €
10	Rasenbesen "Bert"	Markenprodukt, lange Haltbarkeit	5,99 €



Inhalt

Ziel und Einordnung

Rückblick

Ausgangssituation

Datenbankschnittstelle

Recordsets

- Was sind Recordsets
- Mit Recordsets Daten lesen
- Navigation im Recordset per Zeiger
- Suchen und Finden im Recordset
- Ändern und Hinzufügen im Recordset
- Kombination mit grafischen Abfragen und SQL

Weitere VBA-Funktionen

Fallstricke

Zusammenfassung und Ausblick



Recordsets mit SQL-Abfragen kombinieren



Initialisierung von Recordsets

- mit einer Tabelle: lädt alle Datensätze aus der Tabelle

```
' Alle Kunden  
Set rcsKunden = db.OpenRecordset("tblKunden")
```

- mit einer grafischen Abfrage

```
' Alle Kunden  
Set rcsKunden = db.OpenRecordset("qryKundenAusBerlin")
```

- mit einer SQL-Abfrage

```
' Alle Kunden  
Set rcsKunden = db.OpenRecordset("SELECT * FROM tblKunden")  
' Kunden aus Berlin (auf Leerzeichen in SQL-Abfrage achten)  
Set rcsKunden = db.OpenRecordset("SELECT * FROM tblKunden" & _  
                                " WHERE ort='Berlin'")
```



Inhalt

Ziel und Einordnung

Rückblick

Ausgangssituation

Datenbankschnittstelle

Recordsets

- Was sind Recordsets
- Mit Recordsets Daten lesen
- Navigation im Recordset per Zeiger
- Suchen und Finden im Recordset
- Ändern und Hinzufügen im Recordset
- Kombination mit grafischen Abfragen und SQL

Weitere VBA-Funktionen

Fallstricke

Zusammenfassung und Ausblick





Inhalt

Ziel und Einordnung

Rückblick

Ausgangssituation

Datenbankschnittstelle

Recordsets

- Was sind Recordsets
- Mit Recordsets Daten lesen
- Navigation im Recordset per Zeiger
- Suchen und Finden im Recordset
- Ändern und Hinzufügen im Recordset
- Kombination mit grafischen Abfragen und SQL

Weitere VBA-Funktionen

Fallstricke

Zusammenfassung und Ausblick

Weitere VBA-Funktionen zum Datenzugriff



Bisher: Zugriff auf mehrere Datensätze und deren Werte mittels Recordsets

Jetzt: Ermittlung eines Ergebniswertes aus der Datenbank mittels Domänenfunktionen

- verschiedene Funktionen, z.B.
 - zum Ermitteln eines Wertes aus einem Datensatz
 - zum Ermitteln eines Wertes des ersten/letzten Datensatzes
 - zum Zählen von Datensätzen (anhand einer nicht leeren Spalte)

```
' Generelle Syntax  
Let <Var> = DLookup(<Spalte>, <Tabelle>, <Bedingung>)  
Let <Var> = DFirst(<Spalte>, <Tabelle>, <Bedingung>)  
Let <Var> = DLast(<Spalte>, <Tabelle>, <Bedingung>)  
Let <Var> = DCount(<PrimärschlüsselSpalte>, <Tab>, <Beding>)
```

- weitere Funktionen in Zusammenhang mit Auswertung (z.B. Minimum, Maximum) in LE09

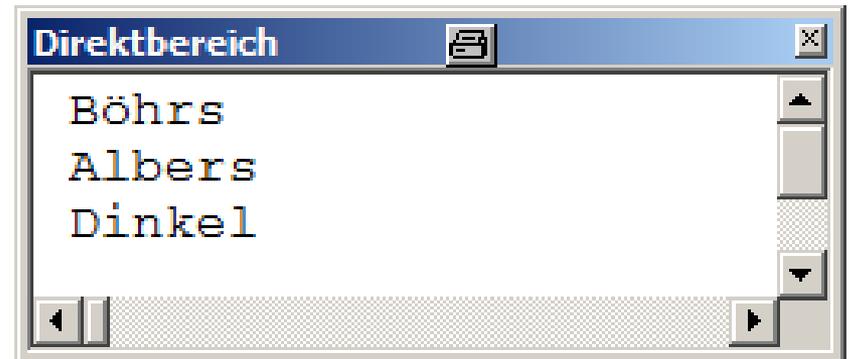
Weitere VBA-Funktionen zum Datenzugriff



Beispiele

```
' Deklaration
Dim strName As String
Dim strErsterName As String
Dim strLetzterName As String
' Initialisierung
Let strName = DLookup("kndName", "tblKunden", "kndIdPk=2")
Let strErsterName = DFirst("kndName", "tblKunden")
Let strLetzterName = DLast("kndName", "tblKunden")
' Ausgabe
Debug.Print strName
Debug.Print strErsterName
Debug.Print strLetzterName
```

tblKunden			
	kndIdPk	kndName	kndVorname
+	1	Albers	Willi
+	2	Böhrs	Thomas
+	3	Dinkel	Ulrike
*	(Neu)		



Weitere VBA-Funktionen zum Datenzugriff



Beispiele

' Deklaration

```
Dim lngAnzahl As Long  
Dim lngNurKat2 As Long
```

' Initialisierung

```
Let lngAnzahl = DCount("prdIdPk", "tblProdukte")  
Let lngNurKat2 = DCount("prdIdPk", "tblProdukte", "prdkatIdFk=2")
```

' Ausgabe

```
Debug.Print "Anzahl Produkte: " & lngAnzahl  
Debug.Print "Anzahl Produkte in Kategorie 2: " & lngNurKat2
```

```
Direktbereich  
Anzahl Produkte: 11  
Anzahl Produkte in Kategorie 2: 4
```

tblProdukte	prdIdPk	prdBezeichnung	prdBeschrei	prdPreis		prdlagerbe:	prdkatIdFk
	1	Blumenzwiebeln Sorte	Rote Blumen, :	0,59 €	🕒(1)	12	1
	2	Blumensamen, Sorte A	Gelbe Blumen,	0,99 €	🕒(1)	32	1
	3	Sonnenblumensamen,	Große gelbe Bl	1,99 €	🕒(1)	9	1
	4	Rasendünger Maxi	5kg, Lanzeitdü	5,99 €	🕒(1)	3	2
	5	Rasendünger Mega	10kg, Sofortwi	9,99 €	🕒(1)	4	2
11	6	Blumendünger "Sabine	500 ml, Grün- t	0,99 €	🕒(1)	45	2
	7	Blumendünger "Dörte"	500 ml, Bio-Dü	2,99 €	🕒(1)	23	2
	8	Rindenmulch fein	6 kg, feiner Mu	2,49 €	🕒(1)	23	
	9	Rindenmulch grob	6 kg, grober Mi	2,69 €	🕒(1)	2	
	10	Rasenbesen "Bert"	Markenproduk	5,99 €	🕒(1)	45	3
	11	Harke "Boris"	Markenproduk	6,99 €	🕒(1)	32	3

4



Inhalt

Ziel und Einordnung

Rückblick

Ausgangssituation

Datenbankschnittstelle

Recordsets

- Was sind Recordsets
- Mit Recordsets Daten lesen
- Navigation im Recordset per Zeiger
- Suchen und Finden im Recordset
- Ändern und Hinzufügen im Recordset
- Kombination mit grafischen Abfragen und SQL

Weitere VBA-Funktionen

Fallstricke

Zusammenfassung und Ausblick





Inhalt

Ziel und Einordnung

Rückblick

Ausgangssituation

Datenbankschnittstelle

Recordsets

- Was sind Recordsets
- Mit Recordsets Daten lesen
- Navigation im Recordset per Zeiger
- Suchen und Finden im Recordset
- Ändern und Hinzufügen im Recordset
- Kombination mit grafischen Abfragen und SQL

Weitere VBA-Funktionen

Fallstricke

Zusammenfassung und Ausblick



Fallstricke

Endlosschleifen vermeiden

- Vergessen Sie nicht MoveNext oder FindNext, wenn Sie in einer Do Until-Schleife arbeiten

Keine Zählerschleife

- Verwenden Sie keine For-Schleife, um Recordsets vom Typ Dynaset (anhand der zuvor gezählten Elemente) zu bearbeiten.
- Anzahl im Recordset kann sich ändern, u.a. durch Einfügen und Löschen

Aufräumen nicht vergessen

- Denken Sie daran, das Recordset mit Close zu schließen, wenn Sie es nicht mehr benötigen, die Prozedur aber noch nicht endet.
- ...



Fallstricke

RecordCount-Eigenschaft

- liefert nur dann den genauen Wert der enthaltenen Datensätze, wenn zuvor mit MoveLast an das Ende des Recordsets gegangen wurde

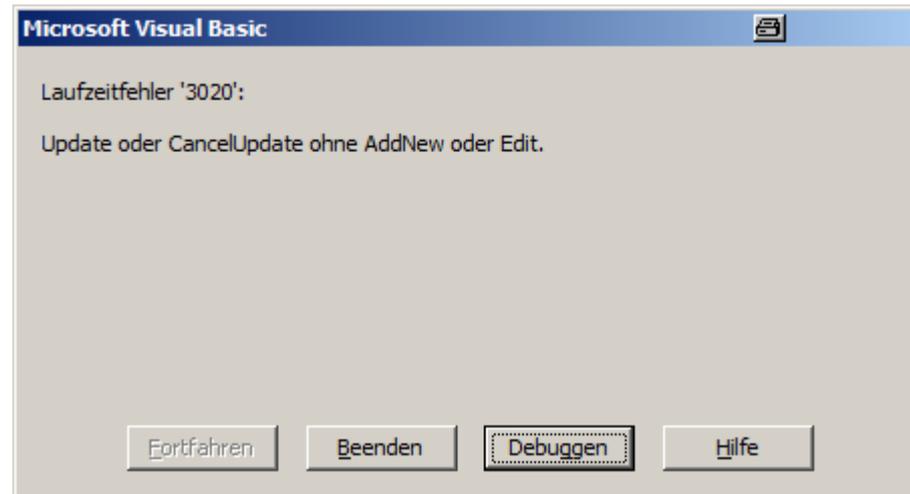
```
' Achtung: Fallstricke bei Recordset: Genaue Anzahl  
' der Datensätze erst nach MoveLast in RecordCount  
If rcsBenutzer.RecordCount > 0 Then  
  ' MoveLast nur, wenn initial der Recordcount > 0 ist  
  ' (sonst Fehler)  
  rcsBenutzer.MoveLast  
End If  
' Jetzt liefert RecordCount die genau Anzahl Datensätze
```

Fallstricke



Edit in einer Schleife

- Änderungen müssen vor dem MoveNext, FindNext usw. mit Update abgeschlossen werden, (unmittelbar) vorher Änderungsmodus aktivieren



– ...

Fallstricke



Edit in einer Schleife

- Änderungen müssen vor dem MoveNext, FindNext usw. mit Update abgeschlossen werden, (unmittelbar) vorher Änderungsmodus aktivieren

```
' Variable für Geburtsjahr deklarieren
Dim intGebJahr As Integer
' Recordset deklarieren, initialisieren
Dim rcsKnd As Recordset
Set rcsKnd = CurrentDb.OpenRecordset("tblKunden", dbOpenDynaset)
'rcsKnd.Edit 'geht hier schief
' Schleife bis zum Ende des Recordset
Do Until rcsKnd.EOF
' rcsKnd.Edit ' So geht es
Let intGebJahr = rcsKnd.Fields("kndGebJahr")
rcsKnd.Edit ' So gehts es auch
rcsKnd.Fields("kndAlter") = 2010 - intGebJahr
rcsKnd.Update ' So gehts
rcsKnd.MoveNext
'rcsKnd.Update 'geht hier schief
Loop
'rcsKnd.Update geht hier auch schief
```



Fallstricke

Bisher ...

- Recordsets mit allen Daten einer Tabelle initialisiert

allerdings ist es nicht sinnvoll,

- Recordset mit sehr vielen Daten füllen und darin per FindNext zu suchen

stattdessen...

- Recordsets mit Abfragen kombinieren
 - Nutzung gespeicherter grafischer Abfragen
 - Nutzung von SQL-Abfragen

aber auch nicht sinnvoll per Recordset nachzubauen

- Standardfunktionen von Access (z.B. zum Füllen von Oberflächenelementen)
- SQL-Funktionen (wenn diese beherrscht werden)



Inhalt

Ziel und Einordnung

Rückblick

Ausgangssituation

Datenbankschnittstelle

Recordsets

- Was sind Recordsets
- Mit Recordsets Daten lesen
- Navigation im Recordset per Zeiger
- Suchen und Finden im Recordset
- Ändern und Hinzufügen im Recordset
- Kombination mit grafischen Abfragen und SQL

Weitere VBA-Funktionen

Fallstricke

Zusammenfassung und Ausblick





Inhalt

Ziel und Einordnung

Rückblick

Ausgangssituation

Datenbankschnittstelle

Recordsets

- Was sind Recordsets
- Mit Recordsets Daten lesen
- Navigation im Recordset per Zeiger
- Suchen und Finden im Recordset
- Ändern und Hinzufügen im Recordset
- Kombination mit grafischen Abfragen und SQL

Weitere VBA-Funktionen

Fallstricke

Zusammenfassung und Ausblick

Zusammenfassung



Was stimmt an der Definition von „Recordset“ in der Wikipedia nicht?

The screenshot shows the Wikipedia article for 'Recordset'. On the left is the Wikipedia logo and navigation menu. The main content area has tabs for 'Artikel', 'Diskussion', 'Lesen', 'Bearbeiten', and 'Versionsgeschichte'. The 'Diskussion' tab is selected. Below the tabs is a search bar. The article title 'Recordset' is displayed in a large font. The main text defines a Recordset as a copy of data records from a database. A category box indicates 'Kategorie: Datenbank'. At the bottom, a timestamp states the page was last edited on March 1, 2008.

[Anmelden](#) / [Benutzerkonto erstellen](#)

Artikel **Diskussion** Lesen **Bearbeiten** Versionsgeschichte

Recordset

Ein **Recordset** ist in der [Datenbanksprache](#) eine Kopie von [Datensätzen](#) (engl. *data record*) aus einer oder mehreren [Tabellen](#) zur Anzeige auf dem Bildschirm und eventuellen Änderung. Ein solcher Recordset bleibt stabil, auch wenn die betroffenen Datensätze sich in der Zwischenzeit verändern.

Kategorie: [Datenbank](#)

Diese Seite wurde zuletzt am 1. März 2008 um 05:28 Uhr geändert.

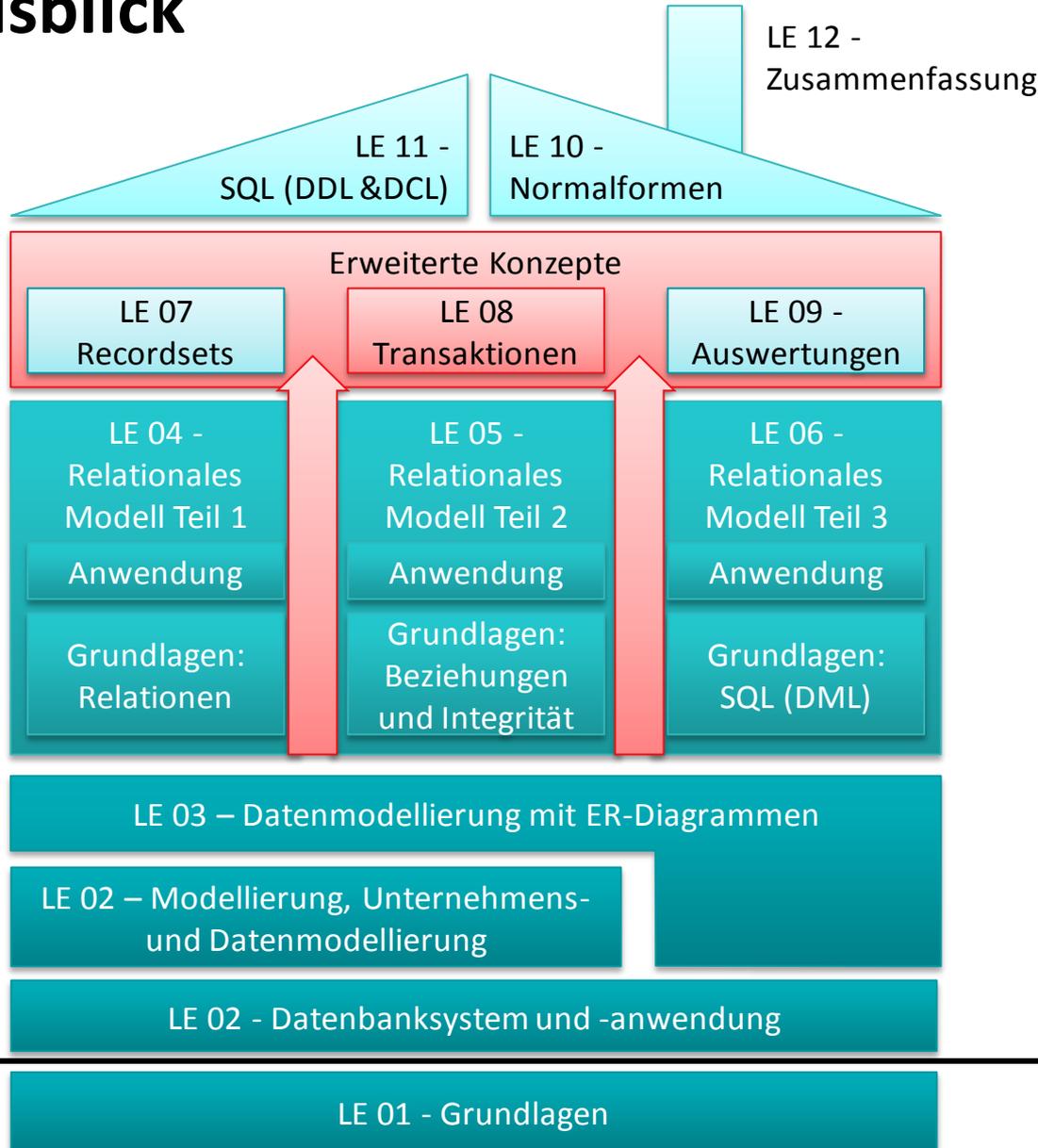
Zusammenfassung



Ein Recordset

- ist eine geordnete Menge von Datensätzen, die aus einer oder mehreren Tabelle einer Datenbank geladen werden.
- kann abhängig von seinem Typ
 - einen Schnappschuss des Datenbestandes repräsentieren
 - mit der Datenbank verbunden sein und Auswirkungen von Änderungen widerspiegeln
- besitzt einen Zeiger, mit dem über Datensätze navigiert werden kann (MoveNext, MovePrevious, ...)
- bietet Möglichkeit auf Werte des Elementes zuzugreifen, auf das der Zeiger zeigt (Fields)
- lässt nach Datensätzen suchen (FindFirst, FindNext, ...)
- kann in verschiedene Modus geschaltet werden
 - Änderungsmodus (Edit)
 - Hinzufüge-Modus (AddNew)
 - Löschmodus (Delete)

Ausblick





BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

Wirtschaftsinformatik 2

LE 07 –Recordsets als Datenbankschnittstelle

Prof. Dr. Thomas Off

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