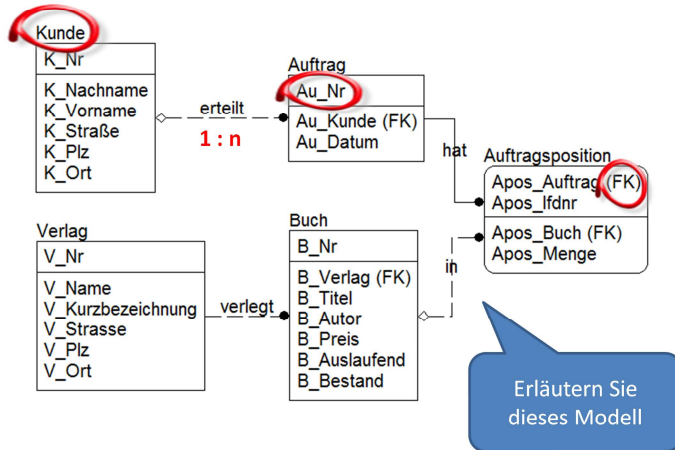


DBIS2 – Datenbanken und Informationssysteme

Lektion 4



Erläutern Sie
dieses Modell



A01

Die Startfolie enthält das Modell (Schema) einer Datenbank

Kunde, Auftrag sind die Namen der Tabellen (stehen oberhalb des Kästchens),
Im Kästchen drinnen stehen Namen der Felder,
der Primary Key (PK) steht oben und ist durch einen Querstrich von den anderen
getrennt

Foreign Keys (FK) sind entsprechend gekennzeichnet.

Die Beziehungen haben eine etwas eigenartige Symbolik,
die Seite mit der schwarzen Kugel ist immer die n Seite der 1:n Beziehungen

Also:

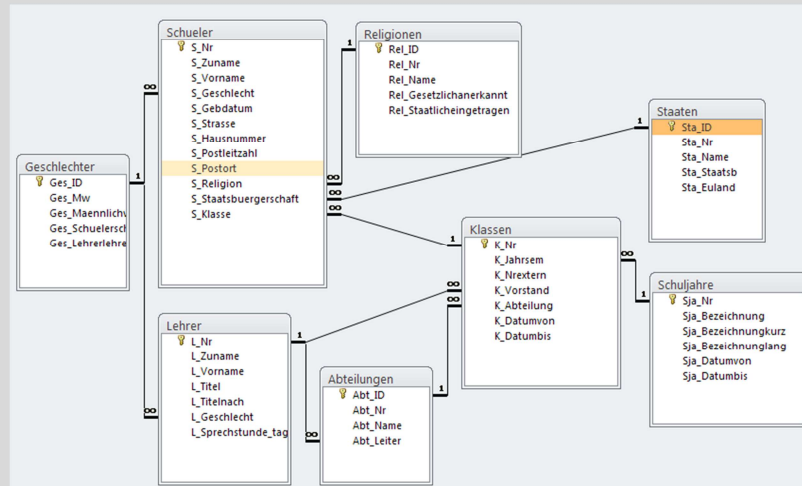
- 1 Kunde erteilt n Aufträge aber 1 Auftrag gehört zu genau einem Kunden
- 1 Verlag verlegt n Bücher aber 1 Buch wird nur von genau einem Verlag verlegt
- 1 Auftrag hat n Auftragspositionen
- 1 Buch wird in n Auftragspositionen verkauft

Zwischen Buch und Auftrag kann man eine n:m Beziehung unterstellen.
Weil es so etwas in Datenbank nicht gibt wurde die
wurde die Tabelle Auftragsposition zur Auflösung der n:m Beziehung geschaffen.

Wiederholung Abfragen (auf 1 Tabelle)



- Man muß die abzufragende Datenbank gut kennen



Ein Bild (genauer Modell) der Datenbank sollte man immer bereit haben

Die Dateninhalt muß man notfalls durch öffnen der Tabelle nachschlagen
z.B. Ges_Mw enthält die Werte „m“ bzw „w“

Wiederholung Abfragen (auf 1 Tabelle) 2



- Alle Übungen vom Ende der letzten Stunde verwendeten genau eine Tabelle (z.B. FROM lehrer), es mußte eine Bedingung formuliert werden (z.B. WHERE S_Name LIKE "S*") und beliebige Spalten ausgegeben werden.
- Lehrer mit Sprechstunde am Montag, sortiert nach Lehrerkürzel

```
SELECT L_Nr, L_Zuname, L_Sprechstunde_tag  
FROM Lehrer  
WHERE L_Sprechstunde_tag="Montag"  
ORDER BY L_Nr
```

The screenshot shows a query tool interface. On the left, a list of fields from the 'Lehrer' table is shown: L_Nr, L_Zuname, L_Vorname, L_Titel, L_Titelnach, and L_Geschlecht. On the right, a table summarizes the query parameters:

Feld:	L_Nr	L_Zuname	L_Sprechstunde_tag
Tabelle:	Lehrer	Lehrer	Lehrer
Sortierung:	Aufsteigend		
Anzeigen:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kriterien:			"Montag"
oder:			

DIE SPENGERGASSE
DER WEG ZUM ERFOLG

DBIS2 - Datenbank und Informationssysteme

3

-- Lehrer mit Sprechstunde am Montag

-- wird in der Entwurfsansicht so erzeugt

```
SELECT Lehrer.[L_Nr], Lehrer.[L_Zuname], Lehrer.[L_Sprechstunde_tag]  
FROM Lehrer
```

```
WHERE (((Lehrer.[L_Sprechstunde_tag])="Montag"))  
ORDER BY Lehrer.[L_Nr];
```

-- genauso ok ist der folgende Befehl (ohne unnütze Zeichen)

```
SELECT L_Nr, L_Zuname, L_Sprechstunde_tag  
FROM Lehrer  
WHERE L_Sprechstunde_tag="Montag"  
ORDER BY L_Nr
```

Wiederholung Abfragen (auf 1 Tabelle) 3



- Welche Lehrer haben keine Sprechstunde eingetragen

```
FROM Lehrer  
WHERE L_Sprechstunde_tag IS NULL
```

- Welche Klassen enden vor dem Schuljahresende
(haben Inhalt in K_Datumbis)

```
From Klassen  
where K_Datumbis is not null
```

Leer oder
nicht leer

- Welche Klassen sind Fachschulklassen (enthalten FID)

```
From Klassen where K_Nr like "*FID*" From Klassen  
where K_Nr like "??FID"
```

- welche Klassen sind 1. jahrgänge

```
From Klassen  
where K_Nr like "1*"
```

Like mog l

DIE SPENGERGASSE
DER WEG ZUM ERFOLG

DBIS2 - Datenbank und Informationssysteme

4

Alle Datenbanken kennen den speziellen Zustand „leer“, dieser muss mit IS NULL abgefragt werden

Like kennt in Access ? (für genau 1 beliebiges Zeichen) bzw * für 0 bis n beliebige Zeichen

In anderen DBs werden statt dessen _ und % verwendet

Erweiterungen auf mächtigere Bedingungen (wie regex) gibt es in manchen DBs

Wiederholung Abfragen (auf 1 Tabelle) 4



- Welche Schuljahre beginnen im Jahr 2009

```
From Schuljahre  
where year(Sja_Datumvon) = 2009
```

Datums-
funktionen
sind praktisch

- Wie lange dauern die verschiedenen Schuljahre

```
SELECT Sja_datumvon, sja_Datumbis,  
       Sja_datumbis - Sja_Datumvon as Dauer  
From Schuljahre
```

- Welche Schüler wohnen an einem "weg" in den Bezirken 2,12 oder 22 (über die Postleitzahl)

```
From Schueler  
where s_postleitzahl in (1020,1120,1220) and S_Strasse like '*weg*'
```

erspart viele OR

Die Kenntnis von Datums- und auch Textfunktionen ist sehr praktisch

Wiederholung Abfragen (auf 1 Tabelle) 5



- Welche Schüler wohnen nicht in Wien und sind vor 1993 geboren

```
From Schueler
where s_postleitzahl >= 2000
and year(s_gebdatum) < 1993
```

- Welche Schüler wohnen in Bezirken, die an die Ringstraße bzw. die 2er-Linie grenzen

```
From Schueler
where S_Postleitzahl between 1010 and 1090
and S_Postleitzahl not in (1050,1020)
```

- where s_geschlecht = month(s_gebdatum)

Burschen im Jänner und Mädchen im Februar geboren

where Instr(S_Hausnummer, '/') = 0

Hausnummer hat keine Stock, Türangabe

Natürlich ist `where s_geschlecht = month(s_gebdatum)`
schon eine abstruse Bedingung, aber Lehrer müssen sich Übungen einfallen lassen

Abfragen in Access mit mehreren Tabellen



- **SQL** wäre sicher nie verwendet worden, wenn man immer nur eine Tabelle abfragen könnte.
- **Die wirkliche Stärke liegt im JOIN mehrerer Tabellen**
- Es können 2 oder mehr Tabellen im FROM kombiniert (mit JOIN verbunden) werden, man muß angeben, welcher Datensatz mit welchen anderen kombiniert werden soll.

FROM Lehrer INNER JOIN Klassen
ON Lehrer.L_Nr = Klassen.K_Vorstand

Tabelle 1 Tabelle 2

Tabelle 1.PK Tabelle2.FK

```
SELECT Lehrer.L_Nr, Lehrer.L_Zuname, Klassen.K_Nr, Klassen.K_Jahrsem  
FROM Lehrer INNER JOIN Klassen  
ON Lehrer.L_Nr = Klassen.K_Vorstand  
ORDER BY Lehrer.L_Nr;
```

INNER JOIN wird benutzt um die Tabellen zu verbinden

ON Klausel definiert, welche Datensätze aus den beiden Tabellen kombiniert werden, praktisch wird hier immer eine existierende Beziehung verwendet, also `tabelle1.PrimaryKey = tabelle2.ForeignKey`

Abfragen in Access mit mehreren Tabellen 2



- Die Entwurfsansicht zeigt den FROM sehr gut an

```
FROM Lehrer INNER JOIN Klassen  
ON Lehrer.L_Nr = Klassen.K_Vorstand
```

SELECT Lehrer.L_Nr,
Lehrer.L_Zuname,
Klassen.K_Nr,
Klassen.K_Jahrsem

Feld:	L_Nr	L_Zuname	K_Nr	K_Jahrsem
Tabelle:	Lehrer	Lehrer	Klassen	Klassen
Sortierung:	Aufsteigend			
Anzeigen:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kriterien:				

ORDER BY Lehrer.L_Nr;

DIE SPENGERGASSE
DER WEG ZUM ERFOLG

DBIS2 - Datenbank und Informationssysteme

8

Im oberen Bereich des Abfrage Entwurfsfensters
wird die FROM Klausel sehr übersichtlich dargestellt

Feld und Tabelle zeigt (wo immer ein Hackerl bei Anzeigen steht) die Select Klausel
hier können Felder aus allen im from stehenden Tabellen aufscheinen

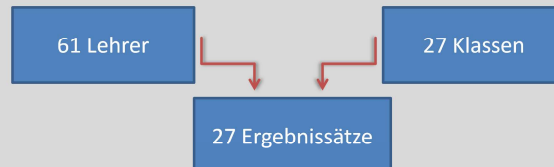
Die Sortierung Zeile regelt den Inhalt des Order by

Abfragen in Access mit mehreren Tabellen 3



- Was passiert beim JOIN?

```
FROM Lehrer INNER JOIN Klassen  
ON Lehrer.L_Nr = Klassen.K_Vorstand
```



- In der Ausgabe sind jene Lehrer enthalten, deren PK Wert (L_Nr) auch im FK der Tabelle Klassen (K_Vorstand) vorkommt. Weil alle Klassen einen KV haben gibt es 27 Ergebnissätze (jede Klasse mit den passenden KV Daten) (Hinweis: Ihre DB kann andere Anzahlen haben)

Die Anzahl der Ausgabedatensätze wird durch die Anzahl der FK Einträge bestimmt. Zu jeder Klasse, die einen Eintrag im FK K_Vorstand hat wird der passende Lehrer-Datensatz dazugejoint.

Lehrer, die nicht KV sind kommen in der Ausgabe nicht vor.

Klassen, die keinen K_Vorstand eingetragen haben würden ebenfalls fehlen

Weil in diesem Fall alle 27 Klassen auch einen eingetragenen Klassenvorstand haben kommt sinngemäss jede Klasse in die Ausgabe, dazu die Daten des Lehrers, welcher KV ist.

Würde eine Klasse keinen KV eingetragen haben, dann gäbe es 26 Ergebnissätze

???? Kann es vorkommen, dass im FK K_Vorstand eine nicht existierendes Lehrerkürzel steht (z.B: xyz) ??????

wenn ja, wie würde sich das auswirken

Abfragen in Access mit mehreren Tabellen 4



FROM Lehrer **INNER JOIN** Klassen
ON Lehrer.L_Nr = Klassen.K_Vorstand

L_Nr	L_Name	K_Nr	K_Vorstand
SKO	Skolud	2CHIF	SKO
PS	Preißl	2AHIF	STJ
DP	Divjak	1BFID	CAM
CAM	Camrda	3BFID	SKO

Nur jene Zeilen in der Ausgabe, wo das Lehrerkürzel auf beiden Seiten existiert !

L_Nr	L_Name	K_Nr	K_Vorstand
SKO	Skolud	2CHIF	SKO
CAM	Camrda	1BFID	CAM
SKO	Skolud	3BFID	SKO

SKO gibt es hier 2mal, daher auch 2 Ausgabezeilen

DIE SPENGERGASSE
DER WEG ZUM ERFOLG

DBIS2 - Datenbank und Informationssysteme

10

Die Lehrer PS und DP kommen nicht in die Ausgabe, weil sie keine KVs sind
 (nie in K_Vorstand vorkommen)

Die Klasse 2AHIF fehlt in der Ausgabe, weil STJ hier nicht in L_Nr vorkommt
 (dies ist in der Datenbank nicht möglich, weil die definierte

Beziehung

zwischen Lehrer und Klassen falsche Werte im FK verhindert)
 allerdings könnte K_Vorstand leer sein, dann fehlt die Zeile auch

Somit sind 2 Lehrer und 3 Klassen in der Ausgabe
 (SKO ist 2mal als KV eingetragen)

Abfragen in Access Übungen für Sie schuldb1_V3.mdb



- In welcher Abteilung ist der Lehrer Bruckner der Abteilungsvorstand (Ausgabe Abteilungsname,...)
- In welchen Klassen (K_Nr, K_jahrsem ausgeben) sind Schüler aus Transdanubien
- Gebe alle Klassen (K_Nr) mit dem Namen der Abteilung aus
- Gebe alle Abteilungen mit Namen und Geschlecht des AV aus
- Welche Schüler haben ein evangelisches Religionsbekenntnis Warum kommt nix raus?

Was ist Transdanubien

In Wien üblicher Ausdruck für die Bezirke (21,22) jenseits der Donau

Abfrage mit der ev. Religion hat keine Ausgabe, weil das Fremdschlüsselfeld (S_Religion) immer leer ist,

dies zeigt recht deutlich, dass die Einträge auf der Fremdschlüsselseite die (Anzahl der) Ausgabesätze bestimmen

Abfragen in Access Übungen für Sie schuldb1_V3.mdb



- Gebe alle Klassen aus, dazu das Beginndatum und das Enddatum (falls dies in der Klasse leer ist verwende den Wert aus dem Schuljahr) Hinweis: Nz-Funktion
- Gebe alle Klassen mit dem Namen des AV aus
- Gebe alle Schülerinnenennamen (also nur weibliche) mit dem Namen des KV aus, falls auch dieser weiblich ist (S_Geschlecht bzw L_Geschlecht muß 2 sein)
- Gebe alle Klassen mit dem Namen des KV und des AV aus (trickreiche Abfrage, benötigt Lehrer 2mal)

Bei Klassen und Schuljahre liegt der Datenbank folgende Überlegung zugrunde.

Im Schuljahr steht auf jeden Fall ein Beginn- und Enddatum (Sja_Datumvon, Sja_Datumbis) Klassen gehören zu genau einem Schuljahr (siehe Beziehung)
In der Klasse gibt es ebenfalls ein Beginn- und Enddatum (K_Datumvon, K_Datumbis)

dies ist aber nur dann ausgefüllt, wenn diese Klasse eine vom Schuljahr abweichende Dauer hat

Will man für eine Klasse das korrekte Beginn- und Enddatum muss man das Schuljahr

im Befehl dazunehmen (JOIN Klasse und Schuljahr) und dann das jeweilige Datumsfeld

aus der Klasse nehmen, falls es befüllt ist – ansonsten das Datum aus dem Schuljahr nehmen

Die Nz Funktion Nz(wert1,wert2) liefert wert1 falls dieser IS NOT NULL, ansonsten wert2

Wenn Sie die Funktion in der Entwurfsansicht benutzen, dann ; statt ,
Nz(wert1;wert2)

Ab der 2. Fragestellung benötigt man mehr als 2 Tabellen

dies ist in der Entwurfsansicht völlig unproblematisch,
schreibt man den Befehl selbst muss im From Zweig der erste Teil geklammert werden (Access Sonderfall)