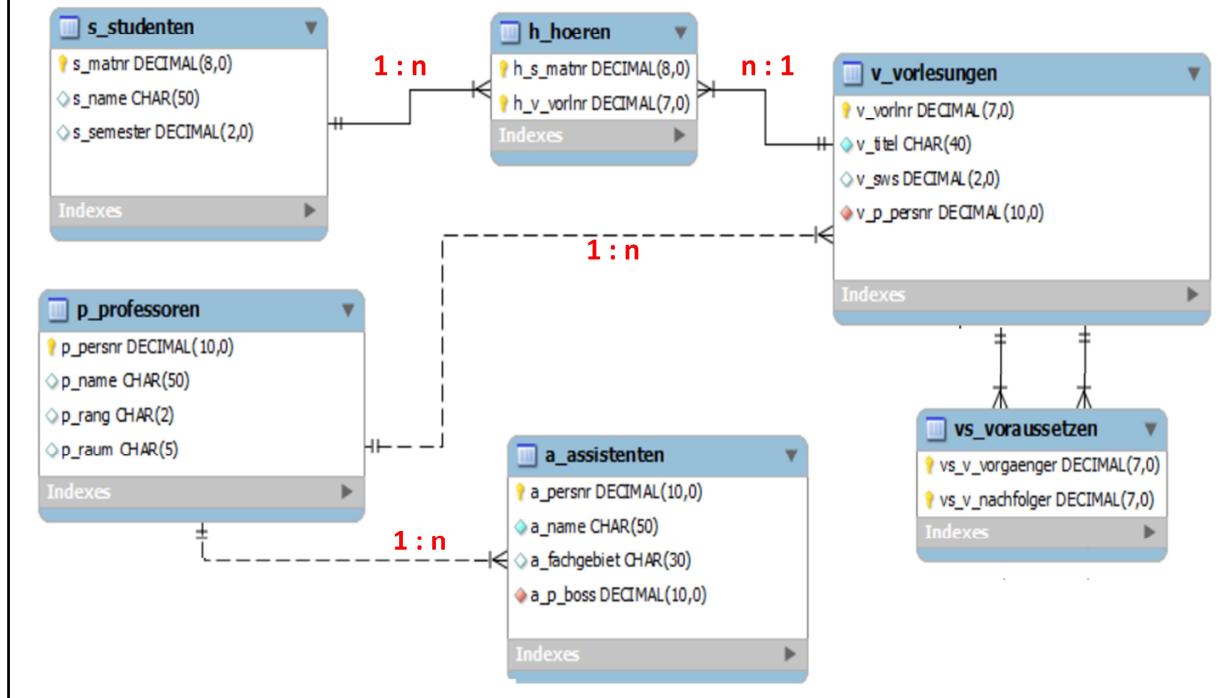


DBIS2 – Datenbanken und Informationssysteme



Die Startfolie enthält wieder ein Modell (gezeichnet mit mysql workbench)

Was bedeuten die Symbole ???? (wie erkennt man Tabelle, Beziehungen, Felder, PK, FK

Wie fast immer: Kästchen sind Tabellen, Striche sind Beziehungen

oben im blauen Bereich der Tabellenname, darunter die Felder

PK Felder mit einem gelben Schlüssel

FK Felder sind rot(rosa) markiert, aber nur dann wenn sie nicht bereits ein PK sind

Die normalen Felder haben hellblau wenn NOT NULL bzw. weiß wenn NULL erlaubt ist

Die Beziehungen sind alle 1 : n, das n Ende wird mit dem sogenannten Krähenfuss dargestellt

Nur dokumentativ sind die quer liegenden kurzen Striche

(die sogenannte Kardinalität = präzisere Aussage zu 1:n))

Inhaltlich geht es um einen Unibetrieb, wo

Professoren mehrere Vorlesungen halten können,

Studenten Vorlesungen besuchen (natürlich mehrere Studenten pro Vorlesung und umgekehrt),

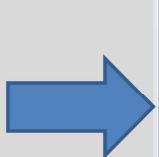
Festgelegt wird, welche Vorlesungen Voraussetzung für andere Vorlesungen sind

- Kann eine Vorlesung von mehreren Professoren gemeinsam abgehalten werden
- Können Sie aus der DB herauslesen, welche Assistenten den Professor bei welchen Vorlesungen unterstützen
- Wie sehen die Datensätze in vs_voraussetzen für die Vorlesung 25/Deutsch1 und 32/Deutsch2 aus wenn Deutsch1 Voraussetzung für Deutsch2 sein soll



- 1) GRUPPIEREN IM DETAIL**
- 2) VERSCHACHTELTER SELECT AN EINEM BEISPIEL**

Gruppierung



The diagram illustrates the grouping process. On the left, a detailed table shows student records with columns 'S_Geschlecht' and 'S_Nr'. An arrow points to the right, where a grouped table shows the count of students per gender. The grouped table has columns 'S_Geschlecht' and 'Anzahl_Schueler'.

S_Geschlecht	S_Nr
1	1
1	2
1	3
1	4
1	5
2	6
2	7
2	8
1	9
1	10
1	11
1	12
1	13

S_Geschlecht	Anzahl_Schueler
1	130
2	23

Gruppenfunktionen:
COUNT(), SUM(),
AVG(), MIN(), MAX()

SQL Schlüsselwörter: GROUP BY, HAVING

3

Mit Gruppierung (oder Summierung)

kann man nach verschiedenen Kriterien

- Zählen COUNT()
- Summieren SUM()
- Durchschnitt AVG()
- kleinster MIN()
- größter MAX()

Das sind die sogenannten Gruppen- oder Aggregatsfunktionen,

die sich deutlich anders verhalten als normale Funktionen

→ sie ermitteln aus mehreren Werten (aus verschiedenen Zeilen einer Spalten) einen Gruppenergebniswert, die detaillierten Einzeldaten werden nicht mehr angezeigt

GROUP BY regelt welche Gruppen gebildet werden

(auf welche Menge von Einzelsätzen jeweils die Gruppenfunktion angewendet wird)

HAVING ist eine Bedingung für die Ergebnisse der Gruppenfunktionen

Abfragen Gruppenfunktionen



- Folgendes gibt nicht die Schüler im Detail sondern nur (in einer Zeile) die Anzahl der Schüler aus

```
SELECT Count(*) as Anzahl FROM Schueler
```

Anzahl
153

- Mit GROUP BY gibt es mehrere Zeilen (abhängig vom Inhalt des Group BY Felds)

```
SELECT S_Geschlecht, Count(s_nr) AS Anz  
FROM Schueler  
GROUP BY S_Geschlecht;
```

S_Geschlecht	Anz
1	130
2	23

- Weil es keine Detailsätze mehr gibt dürfen auch NUR Gruppenfunktionen beim Select geschrieben werden (Ausnahme: Felder beim Group by)

Gruppierung liegt vor, wenn man

GROUP BY oder HAVING und/oder eine der Gruppenfunktionen einsetzt

Wichtiger Hinweis:

Im Fall einer Gruppierung dürfen Sie im Select nur Gruppenfunktionen (und Group by Felder) einsetzen

Es sollte auch auffallen, dass die zugrundeliegenden Detailsätze nicht mehr ausgegeben werden (können)
Stattdessen gibt es „Gruppensummenzeilen“

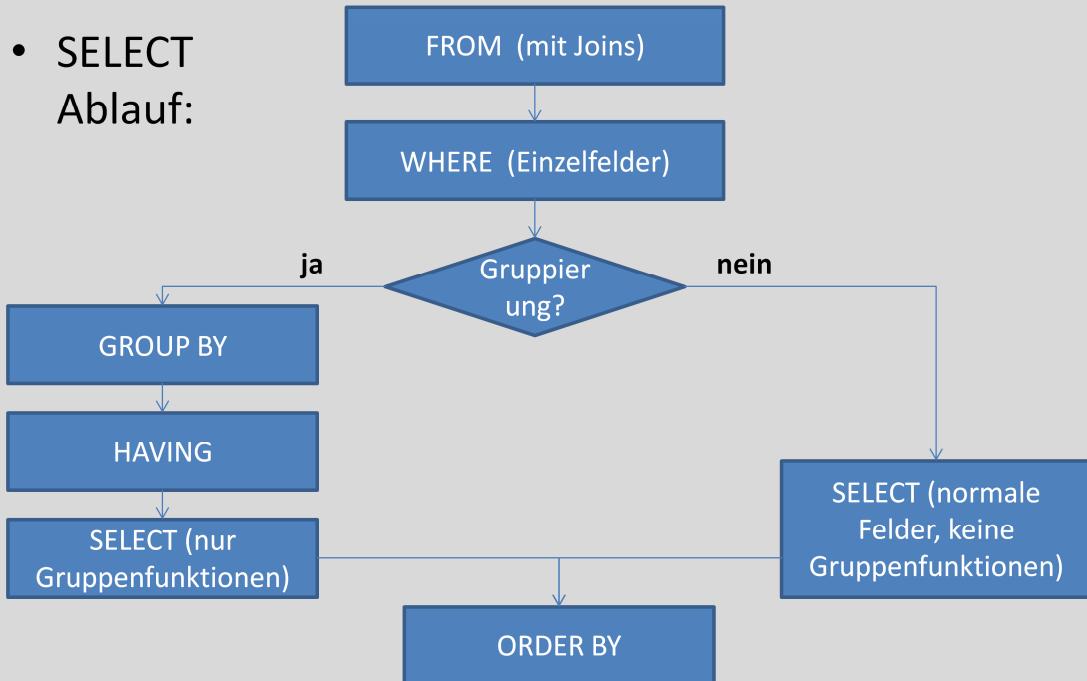
Nur eine einzige Zeile erscheint in der Ausgabe, wenn man kein Group By hat

Mit Einsatz von Group By entscheiden die verschiedenen Werte in den Group by Feldern wie viele Gruppen gebildet werden und somit wie viele Ausgabezeilen entstehen (eine pro Gruppe)

Abfragen Abarbeitungsreihenfolge



- SELECT
Ablauf:



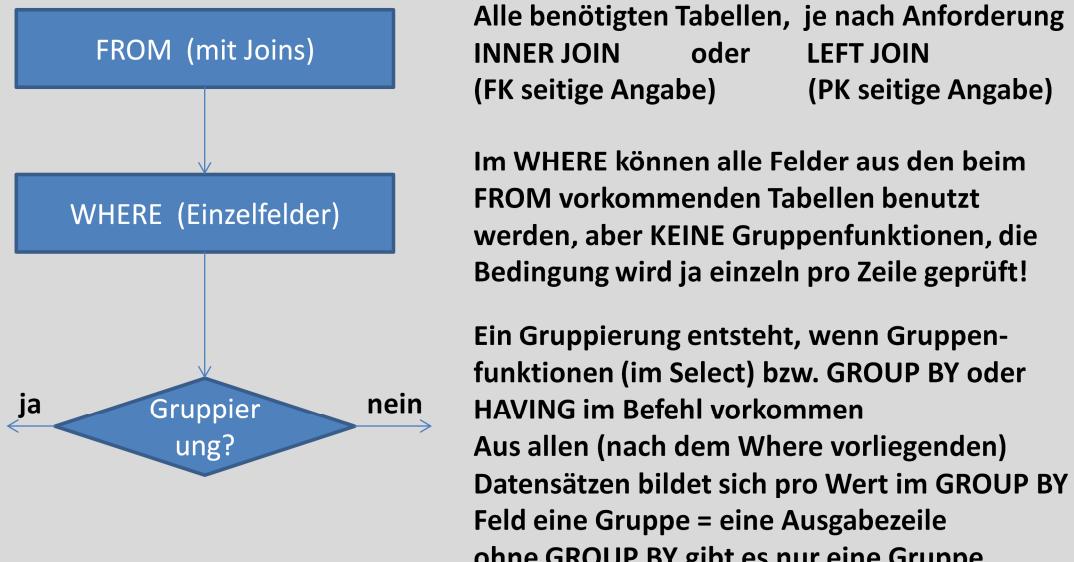
So wird ein SQL Befehl logisch abgearbeitet,

Es wird empfohlen, den Befehl auch in dieser Reihenfolge zu schreiben

Nur SELECT und FROM sind zwingend vorhandene Schlüsselwörter (Klauseln)
die anderen können bei Bedarf eingesetzt werden
werden aber immer in dieser Reihenfolge abgearbeitet



- SELECT Ablauf : in jedem Fall zu Beginn FROM, WHERE



Alle Abfragen beginnen logisch (sowohl in der Abarbeitung, als auch wenn wir sie schreiben) mit FROM und WHERE

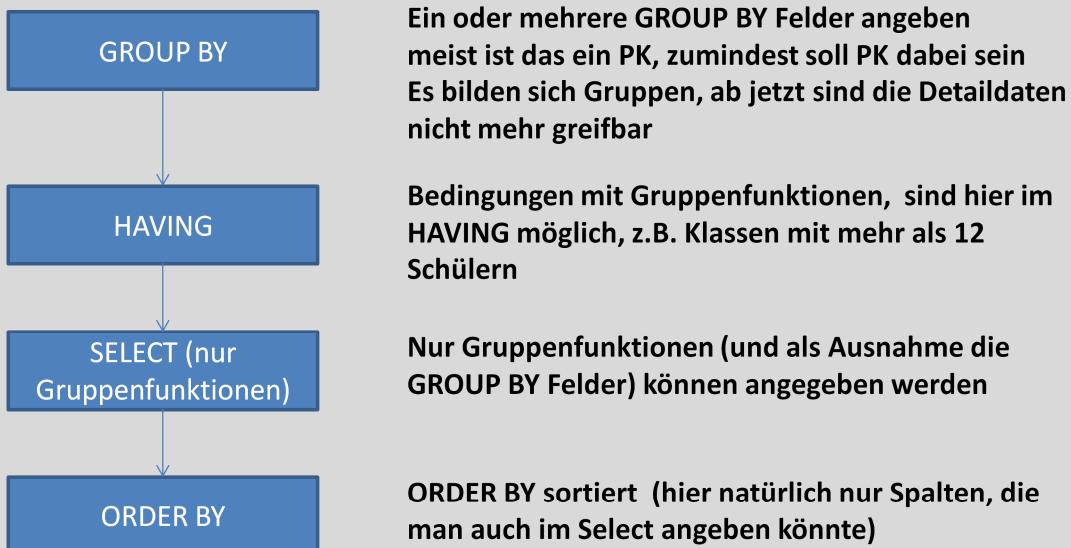
Unabhängig ob später Einzelsätze oder Gruppenergebnisse ausgegeben werden werden hier einzelne Tabellendatensätze beim join kombiniert und danach beim where eventuell ausgeschieden.

Deshalb wird die WHERE Klausel immer so formuliert,
dass sie auf einzelne Datensätze angewendet werden kann
folglich darf sie keine Gruppenfunktionen enthalten

Gruppierung entsteht durch das Vorhandensein von GROUP BY, HAVING
oder durch verwenden einer Gruppenfunktion im SELECT



- SELECT Ablauf bei Gruppierung



Dies ist die Abarbeitungsfolge im Fall von Gruppierung:
Beim GROUP BY kann man ein oder mehrere Felder angeben
Die Inhalte dieser Felder bilden die Gruppen für die Ausgabe
SELECT DISTINCT gruppenfelder FROM
würde genau die sich ergebenden Gruppen ausgeben

Pro Gruppe gibt es dann eine Ausabezeile,
fehlt GROUP BY, so gibt es nur eine einzige Gruppe = nur eine Ausabezeile

Ab diesem Zeitpunkt ist die Verwendung einzelner Felder verboten,
es sind nur mehr Gruppenfunktionen einsetzbar,
eine Ausnahme bilden die beim GROUP BY genannten Gruppenfelder
weil diese innerhalb einer Gruppe zwingend bei allen Zeilen
den gleichen Wert haben kann man sie auch direkt verwenden

HAVING ist eine weitere mögliche Bedingung, wo man mit Gruppenfunktionen arbeiten kann
z.B. HAVING Count(*) > 2 oder HAVING AVG(Note) < 1,5

SELECT im Fall von Gruppierung darf nur mehr Gruppenfunktionen
und als Ausnahme die GROUP BY Felder enthalten



- Wir wollen pro Klasse (oft schreibt man auch je Klasse) die Anzahl der Prüfungen und die Durchschnittsnote ausgeben.
- Natürlich benötigt man die Tabellen Klassen, Schüler und Prüfungen
- Aus der Schülertabelle wird allerdings nichts ausgegeben, sie ist nur als Bindeglied zu den Prüfungen nötig
- INNER JOIN oder OUTER JOIN ?? Anders gefragt, sollen Klassen ohne Schüler bzw. ohne Prüfungen in der Ausgabe erscheinen ??

Wenn in der Angabe das Wort „pro“ oder „je“ vorkommt riecht das sehr nach Gruppierung

Liegt die folgende Beziehung vor:

Tabelle1 ----→ Tabelle2

Und ich will eine Ausgabe pro Tabelle1 (= eine Zeile pro Datensatz in Tabelle1)
geht das nur mittels Gruppierung

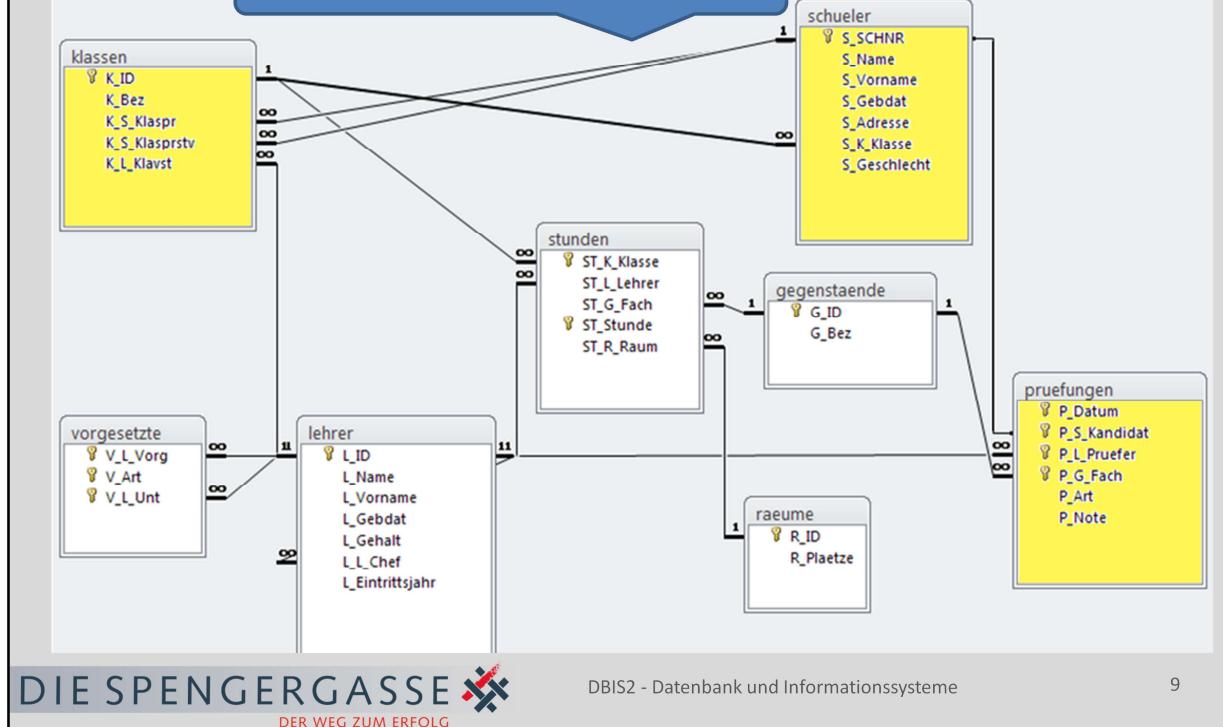
GROUP BY PK-Feld_aus_Tabelle1 ist die richtige Wahl

Felder aus Tabelle2 können dann in Gruppenfunktionen ausgegeben werden

Abfragen Gruppenfunktionen Beispiel die DB



3 Tabelle, richtige Beziehungen wählen



DIE SPENGERGASSE

DER WEG ZUM ERFOLG

DBIS2 - Datenbank und Informationssysteme

9

Die Auswahl der 3 Tabellen sollte jedem klar sein,

Zwischen Klasse und Schüler natürlich nur die eine richtige Beziehung wählen !!!

Fall es jemand auffallen sollte:

Zwischen Schüler und Prüfungen gibt es gar keine echte Beziehung, sondern nur einen Strich

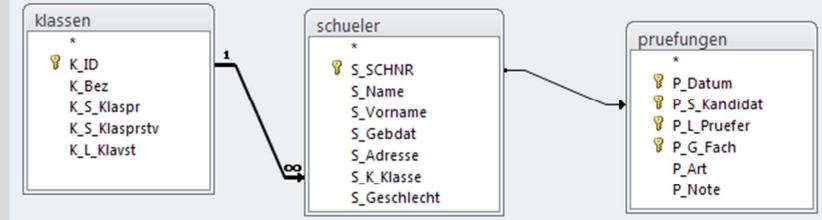
--> beim Erstellen der Beziehung hat man das Häkchen bei referentielle Integrität vergessen.

Das ist zwar sehr schlecht für die DB (es können sinnlose Werte im FK P_S_Kandidat eingetragen werden)
Die Abfrage wird davon nicht gestört

Abfragen Gruppenfunktionen Beispiel FROM



- Entscheidung für outer join, also auch Klassen ohne Prüfungen.



```
SELECT K_ID, S_K_Klasse, S_Schnr, P_S_Kandidat, P_Datum, P_Note
FROM (klassen LEFT JOIN schueler ON klassen.K_ID = schueler.S_K_Klasse)
      LEFT JOIN pruefungen ON schueler.S_SCHNR = pruefungen.P_S_Kandidat
```

- Ausgabe:

K_ID	S_K_Klasse	S_Schnr	P_S_Kandidat	P_Datum	P_Note
01VL	01VL	1111	1111	03.02.2003	3
01VL	01VL	9988			
01VL	01VL	9966			
01VL	01VL	8820	8820	04.04.2003	2
01VL	01VL	7722			
03TA	03TA	1	1	01.04.1989	4
03TA	03TA	1	1	03.03.1990	2
03TA	03TA	1	1	02.02.1990	1
03TA	03TA	1	1	01.04.2003	3

Schüler ohne Prüfungen

Zeigt man mit dem fertigen FROM einige Daten an
sollte man den folgenden GROUP BY besser verstehen.

Wenn in dieser Ausgabe S_K_Klasse (der FK) oder auch S_Schnr (der PK von Schüler)
leer wären, dann hätten wir eine Klassenzeile ohne Schüler (auch ohne Prüfungen)

Wenn P_S_Kandidat (der FK) oder P_Datum (Teil des PK von Prüfungen)

leer sind, dann haben wir Schüler ohne Prüfungen

Beides kann natürlich nur vorkommen, weil wir LEFT JOIN einsetzen

In der Klasse 01VL haben wir 5 Schüler, nur 2 davon haben je eine Püfung,
Darunter beginnt die Klasse 03TA, wo der Schüler (Nr = 1)
gleich mehrere Prüfungen hat

In Folge werden wir eine Ausgabezeile „pro Klasse“ wünschen und daher
eine Gruppierung mit dem PK von Klasse vornehmen

GROUP BY K_ID

Abfragen Gruppenfunktionen Beispiel FROM



- Komplette Ausgabe (sortiert nach K_ID)

K_ID	S_Kl	S_Sch	P_Datum	P_Note
01VL	01VL	1111	03.02.2003	3
01VL	01VL	7722		
01VL	01VL	8820	04.04.2003	2
01VL	01VL	9966		
01VL	01VL	9988		
03TA	03TA	1	01.04.1989	4
03TA	03TA	1	03.03.1990	2
03TA	03TA	9999		
03TB	03TB	77		
03TB	03TB	88		
03TB	03TB	111	04.04.1990	1
03TB	03TB	111	04.04.1990	3
03TB	03TB	111	07.06.1988	1
03TB	03TB	111	15.03.1989	4
03TB	03TB	111	06.02.1988	4
03TB	03TB	111	04.04.1990	1
03TB	03TB	1266	19.10.2002	3
03TB	03TB	1266	22.10.2002	2
03TB	03TB	1266	18.09.2002	2
03TB	03TB	1266	07.06.2001	4
03TB	03TB	1266	20.10.2002	3
03TB	03TB	1266	04.01.1988	3
03TB	03TB	1266	21.10.2002	1
03TB	03TB	7733		
03TB	03TB	7744		
03TB	03TB	7755		
03TB	03TB	8815	04.04.2003	1
03TB	03TB	8818		
03TB	03TB	8819		

GROUP BY K_ID bildet 5 Gruppen

0M5A	0M5A	55	26.02.2003	3
0M5A	0M5A	55	24.12.1987	5
0M5A	0M5A	84		
0M5A	0M5A	122	07.06.2001	5
0M5A	0M5A	122	05.06.1991	2
0M5A	0M5A	122	06.06.2001	1
0M5A	0M5A	1277		
0M5A	0M5A	3333	26.02.2003	2
0M5A	0M5A	4444	26.02.2003	1
0M5A	0M5A	8810	04.04.2003	2
0M5A	0M5A	8810	04.04.2003	2
0M5A	0M5A	8817	05.05.2003	2
0M5A	0M5A	8817	04.04.2003	2
0M5A	0M5A	8817	07.04.2003	5
0M5A	0M5A	8817	04.04.2003	1
0M5A	0M5A	8877		
0M5A	0M5A	8899		
0M5A	0M5A	9933		
0M5A	0M5A	9977		
0M5Q	0M5Q	2222	03.03.2002	2
0M5Q	0M5Q	2222	26.08.2003	1
0M5Q	0M5Q	8811		

Schüler ohne
Prüfungen

Zeigt man mit dem fertigen FROM einige Daten an
sollte man den folgenden GROUP BY besser verstehen.

Hier sieht man (den größten Teil) der Detaildaten,
Zeilen einer Klasse sind jeweils farblich markiert

In Folge werden wir eine Ausgabezeile „pro Klasse“ wünschen und daher
eine Gruppierung mit dem PK von Klasse vornehmen

GROUP BY K_ID

Weil es 5 verschiedene Inhalte in K_ID gibt
erhalten wir genau 5 Gruppen = 5 Ausgabezeilen

Abfragen Gruppenfunktionen Beispiel komplett



- GROUP BY K_ID

```
SELECT K_ID, COUNT(*) AS Zeilen_in_Gruppe,
       COUNT(P_S_Kandidat) AS Anz_Pruefungen,
       MIN(P_Datum)      AS erstes_Pruefungsdatum,
       AVG(P_Note)       AS Notendurchschnitt
  FROM (klassen LEFT JOIN schueler ON klassen.K_ID = schueler.S_K_Klasse)
        LEFT JOIN pruefungen ON schueler.S_SCHNR = pruefungen.P_S_Kandidat
 GROUP BY K_ID
```

- Ausgabe:

K_ID	Zeilen_in_G	Anz_Pruefungen	erstes_Pruefungsdatum	Notendurchschnitt
01VL	5	2	03.02.2003	2,5
03TA	33	18	01.01.1988	2,77777777777778
03TB	21	14	04.01.1988	2,35714285714286
0M5A	19	13	24.12.1987	2,53846153846154
0M5Q	3	2	03.03.2002	1,5

Anzahl Zeilen mit Prüfungen

Wie man schon an den FROM-Daten gesehen hat gibt es 5 verschiedene Werte in K_ID,
daher bildet GROUP BY K_ID fünf Gruppen

COUNT(*) AS Zeilen_in_Gruppe ist in Fall von Outer Join eher sinnlos
zählt eben nur die Datensätze in der Gruppe

COUNT(P_S_Kandidat) zählt hingegen verlässlich die Anzahl der Prüfungen
+-- hier soll der FK (in diesem Fall P_S_Kandidat) angegeben werden

COUNT(DISTINCT S_K_Klasse) zählt die Anzahl der Schüler in der Klasse
geht aber leider in Access nicht, in den großen Server DBs schon

Der Befehl zum kopieren:

```
SELECT K_ID, COUNT(*) AS Zeilen_in_Gruppe,
       COUNT(P_S_Kandidat) AS Anz_Pruefungen,
       MIN(P_Datum)      AS erstes_Pruefungsdatum,
       AVG(P_Note)       AS Notendurchschnitt
  FROM (klassen LEFT JOIN schueler ON klassen.K_ID = schueler.S_K_Klasse)
        LEFT JOIN pruefungen ON schueler.S_SCHNR = pruefungen.P_S_Kandidat
 GROUP BY K_ID
```

Abfragen Gruppenfunktionen Beispiel Having



- HAVING COUNT(P_S_Kandidat) > 5 or AVG(P_Note)

```
SELECT K_ID, COUNT(*) AS Zeilen_in_Gruppe,
       COUNT(P_S_Kandidat) AS Anz_Pruefungen,
       MIN(P_Datum)        AS erstes_Pruefungsdatum,
       AVG(P_Note)         AS Notendurchschnitt
  FROM (klassen LEFT JOIN schueler ON klassen.K_ID = schueler.S_K_Klasse)
        LEFT JOIN pruefungen ON schueler.S_SCHNR = pruefungen.P_S_Kandidat
 GROUP BY K_ID
 HAVING COUNT(P_S_Kandidat) > 5 or AVG(P_Note) < 2
```

K_ID	Zeilen_in_G	Anz_Pruefungen	erstes_Pruefungsdatum	Notendurchschnitt
03TA	33	18	01.01.1988	2,77777777777778
03TB	21	14	04.01.1988	2,35714285714286
0M5A	19	13	24.12.1987	2,53846153846154
0M5Q	3	2	03.03.2002	1,5

Anzahl Zeilen mit Prüfungen

HAVING eignet sich um eine Bedingung für die Ausgabe
der Gruppensummenzeilen zu erstellen,
enthält daher nur Gruppenfunktionen

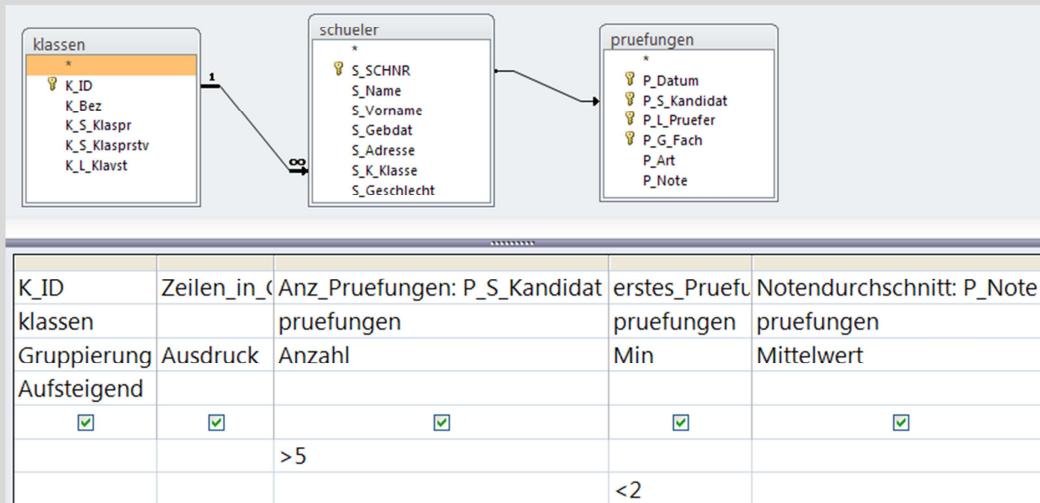
HAVING COUNT(P_S_Kandidat) > 5 or AVG(P_Note) < 2
... Anzahl Prüfungen > 5 oder
Notendurchschnitt kleiner 2

```
SELECT K_ID, COUNT(*) AS Zeilen_in_Gruppe,
       COUNT(P_S_Kandidat) AS Anz_Pruefungen,
       MIN(P_Datum)        AS erstes_Pruefungsdatum,
       AVG(P_Note)         AS Notendurchschnitt
  FROM (klassen LEFT JOIN schueler ON klassen.K_ID = schueler.S_K_Klasse)
        LEFT JOIN pruefungen ON schueler.S_SCHNR = pruefungen.P_S_Kandidat
 GROUP BY K_ID
 HAVING COUNT(P_S_Kandidat) > 5 or AVG(P_Note) < 2
 order by K_Id
```

Abfragen Gruppenfunktionen Beispiel



- In der Entwurfsansicht



Während der FROM Zweig (auch einfache Select Teile)
recht einfach in der Entwurfsansicht erstellt werden können

sollte man danach eher in der SQL Ansicht weiterarbeiten

Abfragen Gruppenfunktionen Beispiel Fragen



- Kann man Felder aus Prüfungen oder Schüler direkt ausgeben (Select S_Name, P_Note, ...) ?
NEIN, nur innerhalb von Gruppenfunktionen
(es ist nicht möglich aus vielen Zeilen mit vielen Werten einen Einzelwert auszugeben)
- Kann man Felder aus Klasse ausgeben, wir verwenden ja GROUP BY PK_von_Klasse ?
Nur innerhalb von Gruppenfunktionen (in allen Zeilen der Gruppe haben die Felder aus Klassen den gleichen Wert, daher ist MAX(K_Bez) oder auch MIN(K_Bez) möglich)
- Kann man weiter Felder zum GROUP BY schreiben ?
Nur Felder aus Klasse, sonst verändern sich die Gruppen

Zusammengefasst:

Nur Gruppenfunktionen im Select !!!

Wenn die Angabe fordert Ausgabezeilen „PRO tabelle1“
dann ist der PK dieser Tabelle das geeignete GROUP BY Feld

Wenn man schon unbedingt (unnützerweise)
mehr Felder in den GROUP BY schreiben will als man braucht
darf man zum PK von tabelle1 weitere Felder aus tabelle1 dazustellen

GROUP BY K_ID bildet die selben Gruppen wie GROUP BY K_ID, K_Bez

GROUP BY K_ID, S_Gebdat ist aber etwas ganz anderes
und auch ziemlich sinnlos

Abfrage „Suche Zeilen mit größten Wert in Spalte x“



- Z.B. „Lehrer mit dem höchsten Gehalt suchen“
man kann sich mit dieser Abfrage behelfen:

```
SELECT L_ID, L_Name, L_Vorname, L_Gehalt
FROM lehrer
ORDER BY L_Gehalt DESC
```
- Besser ist aber ein exakte Ausgabe, dazu zuerst das größte Gehalt ermitteln

```
SELECT MAX(L_Gehalt) FROM lehrer
```
- Dann den (die) Lehrer mit diesem Gehalt

```
SELECT L_ID, L_Name, L_Vorname, L_Gehalt
FROM lehrer
WHERE L_Gehalt = (SELECT MAX(L_Gehalt) FROM lehrer)
```

L_ID	L_Name	L_Vorname	L_Gehalt
B1	Berger	Alfred	400,00
HA	Hanke	Gustav	300,00
LI	Lindner	Kristine	300,00
RX	Renkin	Max	299,00
	Rilek	Hans	298,00

`SELECT MAX(L_Gehalt) FROM lehrer`

`SELECT L_ID, L_Name, L_Vorname, L_Gehalt`

`FROM lehrer`

`WHERE L_Gehalt = (SELECT MAX(L_Gehalt) FROM lehrer)`

L_ID	L_Name	L_Vorname	L_Gehalt
B1	Berger	Alfred	400,00

16

Geschachtelt oder verschachtelt nennt man den Fall,
wo innerhalb eines SQL Befehls ein weiterer innerhalb runder Klammern vorkommt

Für spezielle Fragen ist das notwendig,
aus Performancegründen sollte Verschachtelung aber sparsam eingesetzt werden

Hier ist unser verschachtelter Befehl innerhalb des WHERE Zweigs
weil er im Vergleich = benutzt wird muss er auch genau einen Wert liefern

```
SELECT L_ID, L_Name, L_Vorname, L_Gehalt
FROM lehrer
WHERE L_Gehalt = (SELECT MAX(L_Gehalt) FROM lehrer)
```



- Ermittle die Anzahl der Unterrichtsstunden pro Klasse
- Gibt es Lehrer, die mehr als 10 Stunden unterrichten
- Ermittle die Häufigkeit der einzelnen PrüfungsNoten
- Ermittle pro Chef (L_L_Chef) Anzahl, Durchschnittsgehalt und Durchschnittsalter der Untergebenen
- Wie viele Unterrichtsstunden gibt es pro Wochentag
- Liefere eine Schülerliste der Klasse 03TA, mit Schname, Anzahl Prüfungen, Durchschnittsnote, Anzahl 1er, Anzahl 2er, Anzahl 3er, Anzahl 4er

Basiert auf der ÜbungsDB zum Skriptum (DB des Probetests),
aus der auch das obige GROUP BY Beispiel stammt

Wochentag steht in den ersten 2 Zeichen von ST_Stunde
Funktion Left(St_Stunde,2) ist gut geeignet

Die letzte Frage würde ein SUMMEWENN oder ein DURCHSCHNITTWENN erfordern,
beides existiert nicht

Man behilft sich daher mit einem Rechenausdruck
(in Access mit der WENN Funktion)

IIF(P_Note=1, 1 , 0) bzw. WENN(P_Note=1; 1 ; 0)
in die SUM (.....) hineinschreiben

In der professionellen Server DBs benutzt man Case

Sum (CASE WHEN P_Note = 1 THEN 1 ELSE 0 END)