

Einführung in Datenbanken

Übung 8: ORDER BY, Duplikate, Logik, NOT EXISTS

Prof. Dr. Stefan Brass

PD Dr. Alexander Hinneburg

Martin-Luther-Universität Halle-Wittenberg

Wintersemester 2023/24

<http://www.informatik.uni-halle.de/~brass/db23/>

Inhalt

- 1 Organisatorisches
- 2 Präsenzaufg. 4
- 3 ORDER BY
- 4 Logik
- 5 Übungsblatt 7
- 6 Präsenzaufg. 5

Organisatorisches (1)

- In der letzten Vorlesung (gestern) waren 5 Studierende.
- In StudIP eingetragen sind 118.
- Zur Erinnerung die durchschnittliche Notenverteilung der letzten drei Jahre beim ersten Klausurtermin:
 - ca. 15% bekamen eine 1.0,
 - ca. 25% sind durchgefallen.

„Nicht erschienen“ ist dabei nicht mitgerechnet.
- Es gibt Verbesserungspotential!

Nehmen Sie sich die Zeit, sich mit Datenbanken zu beschäftigen, und insbesondere auch SQL-Anfragen auszuprobieren. Seien Sie nicht zufrieden, wenn die Anfrage funktioniert, sondern probieren Sie es noch einmal anders. Bauen Sie bewusst Fehler ein und beobachten Sie die Reaktion von PostgreSQL.

Organisatorisches (2)

- Ich hatte offenbar vergessen, die Übungsfolien der letzten zwei Wochen auf die Webseite zu stellen.
- Es hat aber auch niemand danach gefragt.
- Ich schließe daraus, dass es
 - entweder kein Problem ist, wenn es diese Folien nicht gibt,

Aus meiner Sicht sollte man die Folien sehr zeitnah anschauen, wenn man die Übung verpasst hat. Wenn man bei der Übung da war, braucht man sie wohl nicht.
 - oder Sie schon so viel Kummer gewöhnt sind, dass Sie das eben so akzeptiert haben.
- Gibt es Fragen?

Inhalt

- 1 Organisatorisches
- 2 Präsenzaufg. 4
- 3 ORDER BY
- 4 Logik
- 5 Übungsblatt 7
- 6 Präsenzaufg. 5

Präsenzaufgabe 4: Duplikate (1)

- Kann diese Anfrage Duplikate liefern?

```
SELECT S.VORNAME, S.NACHNAME
FROM   STUDENTEN S, BEWERTUNGEN B
WHERE  S.SID = B.SID
AND    B.PUNKTE = 10
AND    B.ATYP = 'H'
```

- Laden Sie eine `.txt`-Datei hoch mit der Antwort „Ja“ oder „Nein“ und außerdem:
 - Falls Nein: Beweisen Sie die Aussage, z.B. mit dem obigen Algorithmus.
 - Falls Ja: Geben Sie einen Datenbankzustand an, in dem die Anfrage ein Duplikat liefert.

Präsenzaufgabe 4: Duplikate (2)

Lösung:

- Ja! (Student/in mit 10 Punkten für HA 1 und HA 2)
- Also z.B. Duplikat in folgendem DB-Zustand:

STUDENTEN			
<u>SID</u>	VORNAME	NACHNAME	EMAIL
101	Lisa	Weiss	weiss@acm.org

BEWERTUNGEN			
<u>SID</u>	<u>ATYP</u>	<u>ANR</u>	PUNKTE
101	H	1	10
101	H	2	10

Präsenzaufgabe 4: Duplikate (3)

Lösung, Forts.:

- Variablenbelegung $\mathcal{A}_1$ erzeugt Ausgabe „Lisa Weiss“:

STUDENTEN			
<u>SID</u>	VORNAME	NACHNAME	EMAIL
S → 101	Lisa	Weiss	weiss@acm.org

BEWERTUNGEN			
<u>SID</u>	<u>ATYP</u>	<u>ANR</u>	PUNKTE
B → 101	H	1	10
	H	2	10

Präsenzaufgabe 4: Duplikate (4)

Lösung, Forts.:

- Variablenbelegung $\mathcal{A}_2$ erzeugt auch Ausgabe „Lisa Weiss“:

					STUDENTEN			
					<u>SID</u>	VORNAME	NACHNAME	EMAIL
S →					101	Lisa	Weiss	weiss@acm.org

					BEWERTUNGEN			
					<u>SID</u>	<u>ATYP</u>	<u>ANR</u>	PUNKTE
B →					101	H	1	10
					101	H	2	10

Präsenzaufgabe 4: Duplikate (5)

- Der Algorithmus kann nicht beweisen, dass es keine Duplikate gibt und liefert aber einen Hinweis auf das Problem:

```
SELECT S.VORNAME, S.NACHNAME
FROM   STUDENTEN S, BEWERTUNGEN B
WHERE  S.SID = B.SID
AND    B.PUNKTE = 10
AND    B.ATYP = 'H'
```

- $\mathcal{K} := \{S.VORNAME, S.NACHNAME\}$
- $\mathcal{K} := \mathcal{K} \cup \{B.PUNKTE, B.ATYP\}$
- $\mathcal{K} := \mathcal{K} \cup \{S.SID, S.EMAIL\}$

Weil $\{S.VORNAME, S.NACHNAME\} \subseteq \mathcal{K}$ ein Schlüssel von S ist.

- $\mathcal{K} := \mathcal{K} \cup \{B.SID\}$

Präsenzaufgabe 4: Duplikate (6)

- Die finale Menge $\mathcal{K}$ enthält also:
 - Alle Attribute von S :
 $S.VORNAME$, $S.NACHNAME$, $S.SID$, $S.EMAIL$.
 - Von B nur folgende Attribute:
 $B.PUNKTE$, $B.ATYP$, $B.SID$.
- Damit enthält $\mathcal{K}$ keinen Schlüssel von B .
- Wenn es überhaupt möglich ist, ein Duplikat zu erzeugen, braucht man zwei verschiedene Werte für das fehlende Attribut $B.ANR$.

Wenn der Algorithmus (wie hier) nicht beweisen kann, dass es keine Duplikate gibt, könnte es natürlich trotzdem sein, dass tatsächlich keine Duplikate möglich sind (wegen komplexen WHERE-Bedingungen oder CHECK-Constraints).

Aber es gilt natürlich, dass die Elemente der Menge $\mathcal{K}$ für ein Ergebnistupel nur einen Wert haben können. Das schränkt die Suche ein.

Inhalt

- 1 Organisatorisches
- 2 Präsenzaufg. 4
- 3 ORDER BY**
- 4 Logik
- 5 Übungsblatt 7
- 6 Präsenzaufg. 5

ORDER BY (1)

- Gegeben sei folgende Tabelle R:

R		
A	B	C
a	1	10
b	1	20
c	2	10
d	2	20
e	3	20

- Was gibt diese SQL-Anfrage aus (Folge der A-Werte reicht)?

```
SELECT *  
FROM R  
WHERE B < 3  
ORDER BY B, C DESC
```

ORDER BY (2)

Lösung:

- Die Anfrage war:

```
SELECT *  
FROM R  
WHERE B < 3  
ORDER BY B, C DESC
```

- B ist das Haupt-Sortierkriterium (aufsteigend),
bei gleichem B-Wert wird nach C sortiert (absteigend):

A	B	C
b	1	20
a	1	10
d	2	20
c	2	10

ORDER BY (3)

STUDENTEN

<u>SID</u>	VORNAME	NACHNAME	EMAIL
101	Lisa	Weiss	...
102	Michael	Grau	NULL
103	Daniel	Sommer	...
104	Iris	Winter	...

AUFGABEN

<u>ATYP</u>	<u>ANR</u>	THEMA	MAXPT
H	1	ER	10
H	2	SQL	10
Z	1	SQL	14

BEWERTUNGEN

<u>SID</u>	<u>ATYP</u>	<u>ANR</u>	PUNKTE
101	H	1	10
101	H	2	8
101	Z	1	12
102	H	1	9
102	H	2	9
102	Z	1	10
103	H	1	5
103	Z	1	7

ORDER BY (4)

Aufgabe:

- Ist diese Anfrage syntaktisch korrekt (man beachte, dass beide Tupelvariablen ein Attribut SID haben)?

```
SELECT S.SID, S.VORNAME, S.NACHNAME,  
       B.PUNKTE  
FROM   STUDENTEN S, BEWERTUNGEN B  
WHERE  S.SID = B.SID AND B.ATYP = 'H'  
ORDER BY SID
```

Zum Ausprobieren (Mogeln):

- [\[Link zum Adminer\]](#)

ORDER BY (5)

Aufgabe:

- Ist diese Anfrage syntaktisch korrekt (man beachte, dass beide Tupelvariablen ein Attribut SID haben)?

```
SELECT S.SID, S.VORNAME, S.NACHNAME,  
       B.PUNKTE  
FROM   STUDENTEN S, BEWERTUNGEN B  
WHERE  S.SID = B.SID AND B.ATYP = 'H'  
ORDER  BY SID
```

Lösung:

- Ja, die Anfrage ist korrekt.
- Offiziell kann man nur nach Ergebnisspalten sortieren. Die Ergebnisspalte heißt **SID**.

ORDER BY (6)

Aufgabe:

- Wird diese Anfrage von PostgreSQL und anderen DBMS akzeptiert?

```
SELECT S.SID, S.VORNAME, S.NACHNAME,  
       B.PUNKTE  
FROM   STUDENTEN S, BEWERTUNGEN B  
WHERE  S.SID = B.SID AND B.ATYP = 'H'  
ORDER BY B.SID, S.EMAIL
```

ORDER BY (7)

Aufgabe:

- Wird diese Anfrage von PostgreSQL und anderen DBMS akzeptiert?

```
SELECT S.SID, S.VORNAME, S.NACHNAME,  
       B.PUNKTE  
FROM   STUDENTEN S, BEWERTUNGEN B  
WHERE  S.SID = B.SID AND B.ATYP = 'H'  
ORDER BY B.SID, S.EMAIL
```

Lösung:

- Ja, man kann nach jedem Term sortieren, der unter SELECT gestanden haben könnte.

ORDER BY (8)

Aufgabe:

- Wird diese Anfrage von PostgreSQL akzeptiert?

```
SELECT DISTINCT S.SID, S.VORNAME, S.NACHNAME,  
                B.PUNKTE  
FROM    STUDENTEN S, BEWERTUNGEN B  
WHERE   S.SID = B.SID AND B.ATYP = 'H'  
ORDER  BY B.SID
```

ORDER BY (9)

Aufgabe:

- Wird diese Anfrage von PostgreSQL akzeptiert?

```
SELECT DISTINCT S.SID, S.VORNAME, S.NACHNAME,  
                B.PUNKTE  
FROM    STUDENTEN S, BEWERTUNGEN B  
WHERE   S.SID = B.SID AND B.ATYP = 'H'  
ORDER  BY B.SID
```

Lösung:

- Nein, Fehlermeldung: „for SELECT DISTINCT, ORDER BY expressions must appear in select list“
- Dagegen wird ORDER BY S.SID akzeptiert.

Inhalt

- 1 Organisatorisches
- 2 Präsenzaufg. 4
- 3 ORDER BY
- 4 Logik**
- 5 Übungsblatt 7
- 6 Präsenzaufg. 5

Logischer Vergleich von Anfragen (1)

- Die Tutoren haben mir gesagt, dass die Hausaufgabe zum logischen Vergleich von Anfragen besonders schlecht ausgefallen ist.
- Sie müssen damit rechnen, dass Ankreuzaufgaben von diesem Typ in der Klausur vorkommen.

In den letzten Klausuren waren 5–6 Aufgaben dieses Typs, wobei es einen Punkt pro Aufgabe gab. Das waren ungefähr 10% der Punkte. Für jede von den 5 SQL-Anfragen gab es 5 Punkte.

- Es ist auch ein ganz abstraktes Schema möglich:
 - $R(\underline{A}, B)$
 - $S(\underline{C}, A \rightarrow R, D)$
 - Alle Attribute seien numerisch.

Logischer Vergleich von Anfragen (2)

Aufgabe (zu vergleichende Anfragen):

- Anfrage 1:

```
SELECT DISTINCT R.A
FROM    R, S
WHERE   R.A = S.A
```

- Anfrage 2:

```
SELECT A
FROM    R
```

Schema:

- $R(\underline{A}, B)$
- $S(\underline{C}, A \rightarrow R, D)$

Logischer Vergleich von Anfragen (3)

- Welche der folgenden Aussagen ist korrekt? Wenn mehrere korrekt sein sollten, wählen Sie die erste korrekte Aussage.
 - A. Die beiden Anfragen liefern immer die gleiche Antwort (äquivalent)
 - B. Die beiden Anfragen liefern bis auf Duplikate die gleiche Antwort
 - C. Das Ergebnis von Anfrage 1 ist immer leer (inkonsistent)
 - D. Das Ergebnis von Anfrage 2 ist immer leer (inkonsistent)
 - E. Anfrage 1 liefert immer eine Obermenge ($\supseteq$) von Anfrage 2
 - F. Anfrage 1 liefert immer eine Teilmenge ($\subseteq$) von Anfrage 2
 - G. Keine der Aussagen trifft zu

Bei „B.“ ist gemeint, dass die Tupelmengen (nach Duplikat-Eliminierung) identisch sind. Die Teile „E.“ und „F.“ beziehen sich auch auf die Tupel-Mengen ohne Berücksichtigung eventueller Duplikate.

Logischer Vergleich von Anfragen (4)

Aufgabe (zu vergleichende Anfragen):

- Anfrage 1:

```
SELECT DISTINCT R.A
FROM   R, S
WHERE  R.A = S.A
```

- Anfrage 2:

```
SELECT A
FROM R
```

Lösung:

- Korrekt ist Antwort F: Anfrage 1 liefert immer eine Teilmenge von Anfrage 2 (R-Tupel mit Join-Partner in S).

Logischer Vergleich von Anfragen (5)

Aufgabe (zu vergleichende Anfragen):

- Anfrage 1:

```
SELECT R.A
FROM   R, S
WHERE  R.A = S.A
```

- Anfrage 2:

```
SELECT A
FROM   S
```

Schema:

- $R(\underline{A}, B)$
- $S(\underline{C}, A \rightarrow R, D)$

Logischer Vergleich von Anfragen (6)

Aufgabe (zu vergleichende Anfragen):

- Anfrage 1:

```
SELECT R.A
FROM   R, S
WHERE  R.A = S.A
```

- Anfrage 2:

```
SELECT A
FROM   S
```

Lösung:

- Korrekt ist Antwort A: Die Anfragen sind äquivalent. Aufgrund des Fremdschlüssels hat jedes S-Tupel (genau) einen Join-Partner in R.

Logischer Vergleich von Anfragen (7)

Aufgabe (zu vergleichende Anfragen):

- Anfrage 1:

```
SELECT DISTINCT A
FROM    R
```

- Anfrage 2:

```
SELECT A
FROM    R
```

Schema:

- $R(\underline{A}, B)$
- $S(\underline{C}, A \rightarrow R, D)$

Logischer Vergleich von Anfragen (8)

Aufgabe (zu vergleichende Anfragen):

- Anfrage 1:

```
SELECT DISTINCT A
FROM    R
```

- Anfrage 2:

```
SELECT A
FROM    R
```

Lösung:

- Korrekt ist Antwort A: Die Anfragen sind äquivalent (mit Berücksichtigung von Duplikaten). Weil Spalte **A** Schlüssel von **R** ist, gibt es keine Duplikate.

Logischer Vergleich von Anfragen (9)

Aufgabe (zu vergleichende Anfragen):

- Anfrage 1:

```
SELECT A
FROM R
WHERE A > 5
```

- Anfrage 2:

```
SELECT A
FROM R
WHERE B = 0 AND A > 10 OR A > 5
```

Schema:

- $R(\underline{A}, B)$
- $S(\underline{C}, A \rightarrow R, D)$

Logischer Vergleich von Anfragen (10)

Aufgabe (zu vergleichende Anfragen):

- Anfrage 1:

```
SELECT A
FROM R
WHERE A > 5
```

- Anfrage 2:

```
SELECT  A
FROM    R
WHERE   B = 0 AND A > 10 OR A > 5
```

Lösung:

- Korrekt ist Antwort A: Die Anfragen sind äquivalent.
Wenn $B = 0$ AND $A > 10$ wahr ist, ist auch $A > 5$ wahr.

Inhalt

- 1 Organisatorisches
- 2 Präsenzaufg. 4
- 3 ORDER BY
- 4 Logik
- 5 Übungsblatt 7**
- 6 Präsenzaufg. 5

Schema der Präsidenten-Datenbank

- Schema **president_public** im Adminer:
 - `state(state_name, admin_entered, year_entered)`
 - `president(pres_name, birth_year, years_serv, death_age, party, state_born→state)`
 - `pres_hobby(pres_name→president, hobby)`
 - `administration(admin_nr, pres_name→president, year_inaugurated)`
 - `admin_pr_vp((admin_nr, pres_name)→administration, vice_pres_name)`
 - `pres_marriage(pres_name→president, spouse_name, pr_age, sp_age, nr_children, mar_year)`
 - `election(election_year, candidate, votes, winner_loser_indic)`

Hausaufgabe 7.1: Intervall-Bedingung (2)

Lösung:

- Offensichtlich benötigt man die Tabelle `pres_marriage`.
- Da auch das Geburtsjahr des Präsidenten mit ausgegeben werden soll, benötigt man auch die Tabelle `president`.
- Mögliche Lösung:

```
SELECT p.pres_name, p.birth_year,  
       m.pr_age, m.spouse_name, m.sp_age  
FROM   pres_marriage m, president p  
WHERE  m.pres_name = p.pres_name  
AND    m.sp_age - 1 <= m.pr_age  
AND    m.pr_age <= m.sp_age + 1  
ORDER BY p.birth_year
```

Hausaufgabe 7.1: Intervall-Bedingung (3)

Lösung, Forts.:

- Auch möglich ist die Verwendung von BETWEEN:

```
SELECT p.pres_name, p.birth_year,  
       m.pr_age, m.spouse_name, m.sp_age  
FROM   pres_marriage m, president p  
WHERE  m.pres_name = p.pres_name  
AND    m.pr_age BETWEEN m.sp_age - 1  
              AND m.sp_age + 1  
ORDER BY p.birth_year
```

Beachte: Das AND der dreistelligen Bedingung „... BETWEEN ... AND ...“ hat nichts mit der logischen Verknüpfung AND zu tun.

Hausaufgabe 7.1: Intervall-Bedingung (4)

Lösung, Forts.:

- Auch möglich (aber etwas umständlich) ist eine Disjunktion mit den drei möglichen Werten:

```
SELECT p.pres_name, p.birth_year,  
       m.pr_age, m.spouse_name, m.sp_age  
FROM   pres_marriage m, president p  
WHERE  m.pres_name = p.pres_name  
AND    ( pr_age = m.sp_age - 1  
        OR pr_age = m.sp_age  
        OR pr_age = m.sp_age + 1)  
ORDER BY p.birth_year
```

Hausaufgabe 7.1: Intervall-Bedingung (5)

Lösung, Forts.:

- Noch eine andere Lösung:

```
SELECT p.pres_name, p.birth_year,  
       m.pr_age, m.spouse_name, m.sp_age  
FROM   pres_marriage m, president p  
WHERE  m.pres_name = p.pres_name  
AND    ABS(pr_age - m.sp_age) <= 1
```

- Oder so:

```
SELECT p.pres_name, p.birth_year,  
       m.pr_age, m.spouse_name, m.sp_age  
FROM   pres_marriage m, president p  
WHERE  m.pres_name = p.pres_name  
AND    pr_age - m.sp_age BETWEEN -1 AND 1
```

Hausaufgabe 7.1: Intervall-Bedingung (6)

Bitte:

- Join unter FROM erst benutzen, wenn er auch in der Vorlesung dran war (probieren Sie es mal aus ...).
 - Wenn Sie schon jahrelang SQL schreiben, und sich dieser Stil bei Ihnen verfestigt hat, müssen Sie wohl dabei bleiben.
Obwohl Sie vielleicht gerade dann fit genug wären, auch mal aus Ihrer Komfort-Zone herauszugehen, und etwas Neues zu probieren.
 - Ich wundere mich nur, dass immerhin 38 von 71 es so machen, obwohl es in der Vorlesung nicht dran war. Woher?
 - Ich hoffe, für den **LEFT JOIN** wird ein Punkt abgezogen.
Er ist hier äquivalent zu einem gewöhnlichen Join.
Ich kann das leider nicht erklären, es war ja noch nicht dran.
Wenn Sie schon etwas verwenden, was in der Vorlesung nicht dran war, sollten Sie sich sehr gut damit auskennen.

Hausaufgabe 7.1: Intervall-Bedingung (7)

Weitere Bitten:

- Schreiben Sie keine Zeilen länger als 80 Zeichen!

Das ist unhöflich dem Leser gegenüber. 80 Zeichen sind eine übliche Breite des Editor-Fensters. Man sollte Ihre Anfragen lesen können, ohne den horizontalen Scrollbar benutzen zu müssen. In meinem Editor gibt es keinen horizontalen Scrollbar, sondern die Zeilen werden einfach umgebrochen (abgeschnitten und in der nächsten Zeile fortgesetzt). Das sieht nicht gut aus, aber so übersehe ich nicht, dass es außerhalb des Blickfeldes rechts noch etwas gegeben hätte.

- Setzen Sie nicht notwendige Klammern nur, wenn es die Lesbarkeit wirklich erhöht — auch für andere Leser.
- Delimited Identifier "**president**" für Tabellennamen sind noch immer schlecht (Portabilität, später JDBC).

Hausaufgabe 7.1: Intervall-Bedingung (8)

Beobachtung:

- Die Anfrage funktioniert, auch wenn man die Tupelvariable `p` vor `birth_year` im `ORDER BY` weglässt:

```
SELECT p.pres_name, p.birth_year,  
       m.pr_age, m.spouse_name, m.sp_age  
FROM   pres_marriage m, president p  
WHERE  m.pres_name = p.pres_name  
AND    m.pr_age BETWEEN m.sp_age - 1  
                        AND m.sp_age + 1  
ORDER  BY birth_year
```

- Die Ausgabespalte heisst `birth_year` und nicht `p.birth_year`. Es ist so sogar „korrekter“.

Interessanterweise geht sogar `ORDER BY pres_name`.

Hausaufgabe 7.2: IN-Bedingung (1)

- Welche Präsidenten sind in einem der folgenden Bundesstaaten geboren?
 - Maine,
 - New Hampshire,
 - Massachusetts,
 - Rhode Island,
 - Connecticut,
 - New York,
 - New Jersey
- Diese Staaten liegen alle an der Ostküste, im nördlichen Teil.

Zweck war eine längere Liste von Werten, die ein gutes Anwendungsbeispiel von IN-Bedingungen ist.

Hausaufgabe 7.2: IN-Bedingung (2)

- Geben Sie den Namen des Präsidenten, das Jahr der Ernennung zum Präsidenten und den Bundesstaat aus, in dem er geboren wurde.

Falls ein Präsident mehrere Amtsperioden hatte, soll er entsprechend mehrfach ausgegeben werden. Das Jahr der Ernennung finden Sie in der Tabelle `administration`.

- Das Ergebnis sollte so aussehen:

pres_name	year_inaugurated	state_born
Adams J	1797	Massachusetts
⋮	⋮	⋮
Bush G W	2001	Connecticut
Bush G W	2005	Connecticut
Trump D J	2017	New York

Hausaufgabe 7.2: IN-Bedingung (3)

Lösung:

- SQL-Anfrage mit IN:

```
SELECT    p.pres_name, a.year_inaugurated,
          p.state_born
FROM      president p, administration a
WHERE     p.pres_name = a.pres_name
AND       p.state_born in ('Maine',
                           'New Hampshire',
                           'Massachusetts',
                           'Rhode Island',
                           'Connecticut',
                           'New York',
                           'New Jersey')

ORDER BY year_inaugurated
```

Hausaufgabe 7.2: IN-Bedingung (4)

Lösung, Forts.:

- Alternative:

```
SELECT    p.pres_name, a.year_inaugurated,
          p.state_born
FROM      president p, administration a
WHERE     p.pres_name = a.pres_name
AND       (    p.state_born = 'Maine'
            OR p.state_born = 'New Hampshire'
            OR p.state_born = 'Massachusetts'
            OR p.state_born = 'Rhode Island'
            OR p.state_born = 'Connecticut'
            OR p.state_born = 'New York'
            OR p.state_born = 'New Jersey')
ORDER BY year_inaugurated
```

Hausaufgabe 7.3: Selbstverbund (1)

- In welcher Administration (Zeitintervall nach einer Wahl, identifiziert über die Administrations-Nr) gab es zwei Präsidenten *A* und *B*?
- Vermutlich ist der erste Präsident *A* verstorben (oder musste zurücktreten).
- Die Tabelle *administration* hat öfters das gleiche Jahr der Amtseinführung für beide Präsidenten.
- Um festzustellen, wer der erste Präsident war, kann man in der Tabelle *election* nachschauen, wer bei der Wahl ein Jahr zuvor angetreten ist.

Wenn die Wahl im Jahr j ist, ist die Amtseinführung im Jahr $j + 1$). Sie brauchen nicht zu prüfen, dass die Wahl auch gewonnen wurde —
Präsident und Vizepräsident treten schon als Team an, nicht gegeneinander.

Hausaufgabe 7.3: Selbstverbund (2)

- Geben Sie folgende Daten aus:
 - die gemeinsame Administrations-Nr (`admin_nr`) beider Präsidenten,
 - das Jahr der Amtseinführung des ersten Präsidenten *A*,
 - den Namen des ersten Präsidenten *A*,
 - das ungefähre Todesjahr von *A* (berechnet als Geburtsjahr plus erreichtes Alter)
 - und den Namen des Präsidenten *B* (der nachgerückt ist).

Hausaufgabe 7.3: Selbstverbund (3)

- Das Ergebnis sollte so aussehen:

Nr	Start	Erster	Tod (ca.)	Zweiter
14	1841	Harrison W H	1841	Tyler J
16	1849	Taylor Z	1849	Fillmore M
20	1865	Lincoln A	1865	Johnson A
24	1881	Garfield J A	1880	Arthur C A
29	1901	McKinley W	1901	Roosevelt T
34	1921	Harding W G	1922	Coolidge C
40	1945	Roosevelt F D	1945	Truman H S
44	1961	Kennedy J F	1963	Johnson L B
47	1973	Nixon R M	1994	Ford G R

Die Ausgabespalten sollen so heißen wie hier gezeigt.

Im Fall von Präsident Garfield sieht man, dass die Berechnung des Todesjahres offensichtlich falsch ist. Wenn ein Präsident im Jahr seines Todes nicht mehr Geburtstag hatte, ist das berechnete Jahr um 1 zu klein.

Hausaufgabe 7.4: Selbstverbund (1)

- In welchen Staaten sind mindestens drei Präsidenten geboren?
- Lösen Sie diese Aufgabe ohne Aggregationsfunktionen wie **COUNT** (die in der Vorlesung auch noch nicht behandelt wurden).
- Das erwartete Ergebnis ist:

state_born
Massachusetts
New York
Ohio
Virginia

- Sortieren Sie die Ausgabe nach dem Namen des Staates.

Hausaufgabe 7.4: Selbstverbund (2)

- Mögliche Lösung:

```
SELECT DISTINCT p1.state_born
FROM   president p1, president p2, president p3
WHERE  p1.state_born = p2.state_born
AND    p2.state_born = p3.state_born
AND    p1.pres_name < p2.pres_name
AND    p2.pres_name < p3.pres_name
ORDER BY state_born;
```

- Die Präsidenten-Namen müssen paarweise verschieden sein.
Wenn man <> verwendet, braucht man drei Bedingungen.

Hausaufgabe 7.5: Selbstverbund (1)

- Welche Präsidenten haben sowohl „Golf“ als auch „Riding“ als Hobby (d.h. der gleiche Präsident hat beide Hobbys)?
- Geben Sie das Geburtsjahr des Präsidenten mit aus und sortieren Sie absteigend nach dem Geburtsjahr (d.h. das späteste Geburtsjahr zuerst).
- Das erwartete Ergebnis ist:

pres_name	birth_year
Harding W G	1865
Taft W H	1857
Wilson W	1856

Hausaufgabe 7.5: Selbstverbund (2)

- Mögliche Lösung:

```
SELECT p.pres_name, p.birth_year
FROM president p, pres_hobby h1, pres_hobby h2
WHERE p.pres_name = h1.pres_name
AND h1.pres_name = h2.pres_name
AND h1.hobby = 'Golf'
AND h2.hobby = 'Riding'
ORDER BY birth_year DESC
```

Inhalt

- 1 Organisatorisches
- 2 Präsenzaufg. 4
- 3 ORDER BY
- 4 Logik
- 5 Übungsblatt 7
- 6 Präsenzaufg. 5**

Präsenzaufgabe: Unverheiratete Präsidenten

- Schema `president_public` im Adminer:
 - `president(pres_name, birth_year, years_serv, death_age, party, state_born→state)`
 - `administration(admin_nr, pres_name→president, year_inaugurated)`
 - `pres_marriage(pres_name→president, spouse_name, pr_age, sp_age, nr_children, mar_year)`
- Aufgabe: Schreiben Sie eine Anfrage, die Präsidenten findet, die niemals geheiratet haben.
- Geben Sie den Namen und das Jahr der Amtseinführung aus.

pres_name	year_inaugurated
Buchanan J	1857