



## *Institut für Informatik - Lehrstuhl Datenbanken*

### **Einführung in Datenbanken**

Wintersemester 22/23 - Prof. Dr. Stefan Brass, PD Dr. Alexander Hinneburg

#### **Übung 5: SQL-Anfragen Joins**

Abgabe bis Mo. 28.11.2022, 18:00 Uhr,

Übungsplattform: Studip → Einführung in Datenbanken → Übungsplattform

#### **Aufgabe 5.1:**

6 Punkte

Geben Sie Daten über Komponisten, deren Stücke, Aufnahmen, CDs und Solisten, die in den Aufnahmen spielen. Loggen Sie sich in die PostgreSQL-Datenbank über StudIP ein und wählen Sie das Schema `komponist_public`. Die Tabellen haben folgenden Aufbau:

- `KOMPONIST`(`KNR`, `NAME`, `VORNAME`, `GEBOREN`, `GESTORBEN`)
- `STUECK`(`SNR`, `KNR` → `KOMPONIST`, `TITEL`, `TONART`, `OPUS`)
- `CD`(`CDNR`, `NAME`, `HERSTELLER`, `ANZ_CDS`, `GESAMTSPIELZEIT`)
- `AUFNAHME`(`CDNR` → `CD`, `SNR` → `STUECK`, `ORCHESTER`, `LEITUNG`)
- `SOLIST`((`CDNR`, `SNR`) → `AUFNAHME`, `NAME`, `INSTRUMENT`)

Geben Sie für die folgenden Anfragen entsprechende SQL-Statements an. Für jede Anfrage soll nur ein SQL-Statement angegeben werden. Falls die Tutoren eine SQL-Anfrage nicht schnell verstehen können, kann es Punktabzug geben. Sie können das durch gute Formatierung und hilfreiche Kommentare (wo nötig) vermeiden. Beachten Sie aber, dass die Tutoren SQL natürlich sehr gut kennen. Vermeiden Sie triviale Kommentare. SQL-Statements mit Syntax-Fehlern werden mit Null Punkten bewertet. Geben Sie jede Teilaufgabe als separate Text-Datei mit der Endung `.sql` ab (kein Word). Unter Windows können Sie Text-Dateien mit Notepad editieren. Das Programm Notepad++ (<http://notepad-plus-plus.org>) bietet mehr Funktionen.

- a) Gesucht sind Aufnahmen von Stücken von Vivaldi und Telemann (d.h. der Name des Komponisten ist Vivaldi oder Telemann). Geben Sie den Komponisten-Namen, den Titel des Stücks, die Nummer des Stücks, den Namen der CD, und die Gesamtspielzeit der CD (bzw. des CD-Packs) aus. Nennen Sie die Spalte mit der Spielzeit "min".
- b) Gesucht sind alle Komponisten in der Datenbank, die Zeitgenossen von Wolfgang Amadeus Mozart sind, deren Lebenszeit also mit der von Mozart überlappt. Es sollen dabei auch Komponisten ausgegeben werden, die in dem Jahr gestorben sind, in dem Mozart geboren wurde, bzw. die in dem Jahr geboren wurden, in dem Mozart gestorben ist (wenn die Überlappung also etwas grenzwertig ist).

„Wolfgang Amadeus Mozart“ selbst soll in der Auflistung nicht erscheinen. (Sie können voraussetzen, dass Name und Vorname des Komponisten zusammen einen Alternativschlüssel darstellen, dass es also nicht zwei verschiedene Einträge mit gleichem Namen und Vornamen gibt.)

Geben Sie Name, Vorname, Geburts- und Todesjahr der Komponisten aus, sowie das „Alter“ des Komponisten im Geburtsjahr von Mozart (das kann natürlich negativ sein, wenn der andere Komponist erst nach Mozart geboren wurde).

Sie dürfen nicht die Lebensdaten von Mozart in die Anfrage einsetzen. Zwar werden die sich nicht mehr ändern, aber der Name „Wolfgang Amadeus Mozart“ soll nur ein Beispiel sein. Ihre Anfrage könnte z.B. später in einem Programm verwendet werden, mit Parametern für den tatsächlichen Namen.

- c) Gesucht sind alle CDs, die Stücke von mehr als einem Komponisten enthalten (also mindestens zwei verschiedenen Komponisten). Geben Sie jeweils den Namen der CD aus.

Bei dieser Anfrage werden Sie wahrscheinlich Duplikate erhalten. Verwenden Sie ggf. `SELECT DISTINCT`, um diese zu eliminieren.

Für die Lösung dieser Aufgabe verwenden Sie bitte kein `GROUP BY` und keine Aggregationsfunktionen wie `COUNT`, selbst wenn Sie das schon kennen sollten (in der Vorlesung war es noch nicht dran).

### Aufgabe 5.2:

4 Punkte

Loggen Sie sich in die PostgreSQL-Datenbank über StudIP ein und wählen Sie das Schema `president_public`. Die Tabellen haben folgenden Aufbau:

- `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→president, 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)`

Geben Sie für die folgenden Anfragen entsprechende SQL-Statements an. Für jede Anfrage soll nur ein SQL-Statement angegeben werden. Falls die Tutoren eine SQL-Anfrage nicht schnell verstehen können, kann es Punktabzug geben. Sie können das durch gute Formatierung und hilfreiche Kommentare (wo nötig) vermeiden. Beachten Sie aber, dass die Tutoren SQL natürlich sehr gut kennen. Vermeiden Sie triviale Kommentare. SQL-Statements mit Syntax-Fehlern werden mit Null Punkten bewertet. Geben Sie jede Teilaufgabe als separate Text-Datei mit der Endung `.sql` ab (kein Word). Unter Windows können Sie Text-Dateien mit Notepad editieren. Das Programm Notepad++ (<http://notepad-plus-plus.org>) bietet mehr Funktionen.

- a) Wer hat eine Wahl verloren (Wert 'L' für `winner_loser_indic` in `election`) und später noch eine Wahl gewonnen ('W')?

Falls Sie `DISTINCT` verwenden, schreiben Sie als Kommentar eine kurze Begründung dazu, warum die Anfrage sonst Duplikate liefern könnte.

- b) Welche Präsidenten haben die Hobbies `Riding` und `Golf`? Gesucht sind nur Präsidenten die beide Hobbies haben, nicht etwa Präsidenten, die das eine oder das andere Hobby haben. Geben Sie das Geburtsjahr des Präsidenten mit aus.

Sie dürfen bei dieser Anfrage keine Unteranfragen verwenden.

Erklären Sie wieder, warum Sie `DISTINCT` verwenden.