

Geodatenbanken

SQL

Adrian Ostermann, Pascal Offermann

Einführung in SQL

- **Was ist SQL?**

- SQL steht für Structured Query Language
- Datenbanksprache zur Definition von Datenstrukturen (DDL) in relationalen Datenbanken

- **Was kann SQL?**

- Befehle zur Datenmanipulation (Ändern, Einfügen, Löschen) und lesendem Zugriff (DML)
- Abfragen aus einer Datenbank (DRL)
- Befehle für die Rechteverwaltung und Transaktionskontrolle (DCL)

- **Wichtige Befehle**

- CREATE TABLE, DROP TABLE, INSERT, SELECT, UPDATE, DELETE

Struktur des SELECT - Befehls

- Der select-Befehl hat im Wesentlichen die Struktur :
 - select ... Auflistung von Attributen oder Ausdrücken mit Attributen und Funktionen
 - from ... Auflistung der beteiligten Tabellen
 - where ... Auswahlbedingung über die Datensätze
 - group by ... Zusammenfassung der Datensätze in Gruppen
 - having ... Auswahlbedingung über die Gruppe
 - order by ... Attribute oder Ausdrücke für die Ausgabesortierung

JOIN Funktionen in SQL

- Mit „Joins“ lassen sich die Spalten verschiedener Tabellen über den PRIMARY KEY verknüpfen:
 - **INNER JOIN:** Gibt alle Zeilen zweier Tabellen wieder, wenn mind. Eine Übereinstimmung besteht
 - **LEFT JOIN:** Gibt alle Zeilen der ersten Tabelle wieder und die zugehörigen aus der zweiten Tabelle
 - **RIGHT JOIN:** Gibt alle Zeilen der zweiten Tabelle wieder und die zugehörigen aus der ersten Tabelle
 - **FULL JOIN:** Gibt alle Zeilen wieder, wenn wenigstens eine Übereinstimmung besteht

SQL Funktionen

- SQL aggregate functions geben einen Wert zurück, der aus den Spaltenwerten berechnet wird
- Wichtige Funktionen:
 - AVG() - Returns the average value
 - COUNT() - Returns the number of rows
 - FIRST() - Returns the first value
 - LAST() - Returns the last value
 - MAX() - Returns the largest value
 - MIN() - Returns the smallest value
 - SUM() - Returns the sum

Beispiel: Abfrage

- Gib alle Professoren mit ihren Personalnummern und der Anzahl der Vorlesungen, die sie lesen, aus – aufsteigend sortiert nach der Anzahl der Vorlesungen.

Select p.Name, p.PersNr, count(v.Titel) as Anzahl

FROM Professoren p **LEFT JOIN** Vorlesungen v

ON p.PersNr = v.GelesenVon

Group by p.Name, p.PersNr

Order by Anzahl

Unterabfragen / Verschachtelte Abfrage

- Zwei verschiedene Arten von Unteranfragen: korrelierte und unkorrelierte
- **unkorreliert:** Unteranfrage bezieht sich nur auf "eigene" Attribute
 - Unteranfrage wird einmal ausgewertet für jedes Tupel der äußeren Anfrage wird geprüft, ob die MatrNr im Ergebnis der Unteranfrage vorkommt
- **korreliert:** Unteranfrage referenziert auch Attribute der äußeren Anfrage
 - Für jedes Tupel der äußeren Anfrage hat innere Anfrage verschiedene Werte
 - das exists-Prädikat wahr, wenn exists ist die Unteranfrage mind. ein Tupel enthält

Beispiel: Unteranfrage

Unkorreliert:

```
select s.*  
from Studenten s  
where s.GebDatum <  
(select max (p.GebDatum)  
from Professoren p);
```

Korreliert:

```
select s.*  
from Studenten s  
where exists  
(select p.*  
from Professoren p  
where p.GebDatum >  
s.GebDatum);
```