Datenbank-Systeme

Von Anfragen zu Abfragen - mit nur einem SQL Befehl

Internationaler Frauenstudiengang Informatik

Renzo Kottmann

Wiederholung

Datenbank-Projekt

Anfragen

Z.B. Wieviel Teilnehmer sind in meinem Kurs?

Stärken relationaler Datenbanken:

1. Persistente, sichere und strukturierte Datenspeicherung
2. Effiziente **Abfragen** um Anfragen beantworten zu können!

Stärken relationaler Datenbanken:

1. Persistente, sichere und strukturierte Datenspeicherung
2. Effiziente **Abfragen** um Anfragen beantworten zu können!

- SQL hat nur einen Befehl dafür:

Stärken relationaler Datenbanken:

1. Persistente, sichere und strukturierte Datenspeicherung
2. Effiziente **Abfragen** um Anfragen beantworten zu können!

- SQL hat nur einen Befehl dafür:

SELECT

Anatomie von `SELECT`

```
SELECT *          -- welche Spalten sollen wie angezeigt werden
  FROM tabelle    -- Daten welcher Tabelle
 WHERE true       -- Selektionesbedingungen: nur Daten, die Kriterium entsprechen
```

Anatomie von `SELECT`

```
SELECT *          -- welche Spalten sollen wie angezeigt werden
  FROM tabelle    -- Daten welcher Tabelle
 WHERE true       -- Selektionesbedingungen: nur Daten, die Kriterium entsprechen
```

Kann gelesen werden als:

```
Zeige mir alle Spalten der Tabelle "tabelle" an und davon alle Zeilen.
```

Anatomie von `SELECT`

```
SELECT *         -- welche Spalten sollen wie angezeigt werden
  FROM tabelle   -- Daten welcher Tabelle
 WHERE true      -- Selektionesbedingungen: nur Daten, die Kriterium entsprechen
```

Kann gelesen werden als:

```
Zeige mir alle Spalten der Tabelle "tabelle" an und davon alle Zeilen.
```

Datenbank interpretiert das in der Reihenfolge `FROM`, `WHERE`, `'*'` (Spalten)

```
Hole aus der Tabelle "tabelle" alle Zeilen die der Bedingung 'true' entsprechen
und zeige davon alle Spalten an.
```

Anatomie von SELECT

```
SELECT *         -- welche Spalten sollen wie angezeigt werden
  FROM tabelle   -- Daten welcher Tabelle
 WHERE true      -- Selektionesbedingungen: nur Daten, die Kriterium entsprechen
```

Es wird immer eine und nur eine Tabelle durch SELECT erzeugt, daher ist das technisch präziser:

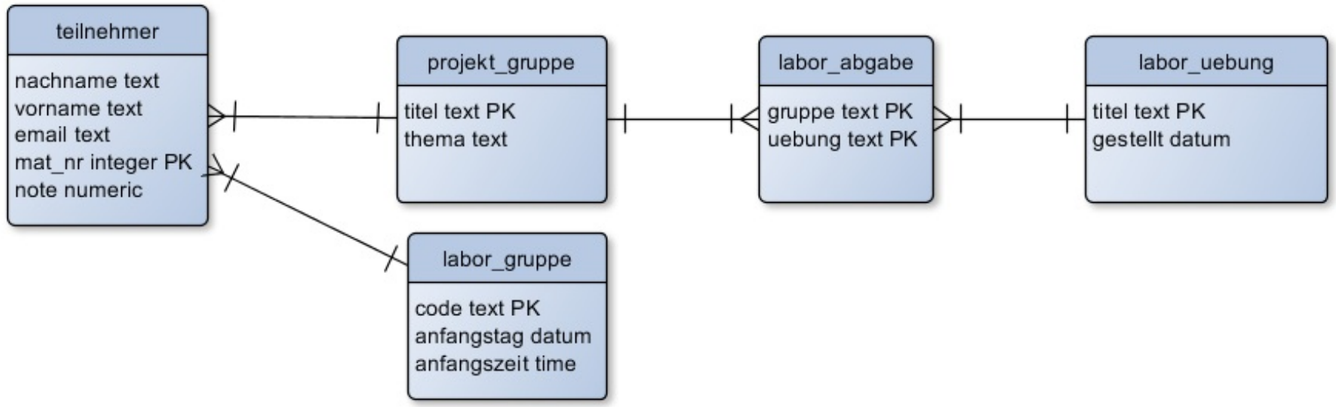
```
 Erzeuge und zeig mir *eine* virtuelle Tabelle, die folgender Anweisung entspricht
 Hole aus der Tabelle "tabelle" alle Zeilen die der Bedingung 'true' entsprechen
 und zeige davon alle Spalten an.
```

Konkretes SELECT

```
SELECT *         -- * (asterisk) heisst alle spalten, wie sie sind
  FROM "teilnehmer"; -- Daten der Tabelle mit dem Namen "order"
```

Boolsche `WHERE` Bedingung kann weggelassen werden, wenn man alle Zeilen will.

Diagramm der Teilnehmer-Datenbank



Source Code der Teilnehmer-Datenbank

- SQL file mit ersten Testdaten

Sammeln von Anfragen

ANFRAGE:

Gib mir alle Daten aller Teilnehmer

ABFRAGE:

```
select *
  from Teilnehmer;
```

Welche Teilnehmer sind im hoeheren Semester als 3

```
select *
  from Teilnehmer
 where semester > 3 ;
```

Datenbank: Query Plan

- Jede Abfrage wird vom DBMS ausgewertet
  - DBMS findet einen optimalen Plan zur Ausführung der Abfrage
  - DBMS bestimmt einen optimalen Algorithmus!

Logische Ausführung vs. Aktuelle Ausführung

- SQL standard legt eine logische Auswertungsreihenfolge fest
- DBMS kann intern davon abweichen

Logische Ausführungsreihenfolge

(1) FROM, JOIN, APPLY

(2) WHERE

(3) CONNECT BY (Oracle)

(4) GROUP BY

(5) AGGREGATIONS

(6) HAVING

(7) WINDOW

(8) MODEL (Oracle)

(9) SELECT

(10) DISTINCT

(11)UNION, INTERCEPT, EXCEPT

(12) ORDER BY

(13) OFFSET

(14) LIMIT

(15) FOR UPDATE

Es geht noch viel mehr mit SQL

(1) FROM, JOIN, APPLY

(2) WHERE

(3) CONNECT BY (Oracle)

(4) GROUP BY

(5) AGGREGATIONS

(6) HAVING

(7) WINDOW

(8) MODEL (Oracle)

(9) SELECT

(10) DISTINCT

(11)UNION, INTERCEPT, EXCEPT

(12) ORDER BY

(13) OFFSET

(14) LIMIT

(15) FOR UPDATE

[1] Gesamtüberblick

[2] SELECT Details

Danke für die Zusammenarbeit