



MUSTERLÖSUNG: ÜBUNGSBLATT 6

Musterlösung der dritten Laborübung Datenbanken (DBA), Studiengang IB,
Sven Klaus, s.klaus@hs-mannheim.de, <http://www.informatik.hs-mannheim.de/~klaus/>

AUFGABE 1

- a) `SELECT vname, name FROM t_ma_dt LIMIT 15;`
- b) `SELECT vname, name, plz, ort FROM t_ma_dt LIMIT 15;`
- c) `SELECT vname AS Vorname, name AS Familienname, plz AS Postleitzahl,
ort As Wohnort
FROM t_ma_dt LIMIT 15;`
- d) `SELECT vname AS Vorname, name AS Familienname, plz AS Postleitzahl,
ort As Wohnort
FROM t_ma_dt
WHERE ort IN ('Hamburg','Berlin') AND alt>30
LIMIT 15;`
- e) `SELECT vname AS Vorname, name AS Familienname, plz AS Postleitzahl,
ort As Wohnort
FROM t_ma_dt
WHERE ort IN ('Frankfurt','Offenbach') AND alt>30
ORDER BY name
LIMIT 15;`

AUFGABE 2

- a) `SELECT name, stueckzahl, preis, preis*1.16 AS Bruttopreis FROM t_lager;`
- b) `SELECT stueckzahl, COUNT(name) FROM t_lager GROUP BY stueckzahl;`
- c) `SELECT stueckzahl, COUNT(name) FROM t_lager WHERE stueckzahl<100
GROUP BY stueckzahl;`
- d) `SELECT stueckzahl, COUNT(name) FROM t_lager WHERE stueckzahl<100
GROUP BY stueckzahl
ORDER BY stueck DESC;`

AUFGABE 3

- a) CREATE TABLE t_buecher
 (isbn VARCHAR(13) NOT NULL,
 titel VARCHAR(100),
 autor VARCHAR(100),
 auflage INTEGER,
 preis FLOAT,
 PRIMARY KEY(isbn));
- b) CREATE TABLE t_leser
 (nr INTEGER NOT NULL AUTO_INCREMENT,
 name VARCHAR(30),
 vname VARCHAR(30),
 gebdat DATE,
 adr VARCHAR(100),
 PRIMARY KEY(nr));
- c) CREATE TABLE t_verleih
 (isbn VARCHAR(13) NOT NULL,
 leser INTEGER NOT NULL,
 datum DATE,
 PRIMARY KEY(isbn,leser),
 FOREIGN KEY(isbn) REFERENCES t_buecher(isbn) ON DELETE CASCADE,
 FOREIGN KEY(leser) REFERENCES t_leser(nr) ON DELETE CASCADE);

Beim Einfügen der Datensätze in die Tabelle t_verleih kommt es bei dem letzten Datensatz zu Problemen. Sein Einfügen wird abgewiesen. Aber warum? Nein, es liegt nicht an der zuvor schon verwendeten ISBN, da t_verleih die Umsetzung eines many-to-many-Relationships ist. Es liegt einzig und allein daran, dass es bisher keinen Leser mit der nr = 3 gibt.

Spätestens wenn Sie hier Schwierigkeiten hatten, dies zu erkennen, sollten Sie über eine konsequente Benennung Ihrer Fremdschlüsselbedingungen und Indizes nachdenken.

AUFGABE 4

- a) CREATE INDEX i_autor ON t_buecher (autor);
- b) CREATE INDEX i_titel ON t_buecher (titel);
- c) SELECT * FROM t_buecher WHERE autor LIKE 'B%';
 SELECT * FROM t_buecher ORDER BY titel;
- d) CREATE INDEX i_verleih ON t_verleih (isbn, leser);
- e) SELECT * FROM t_verleih ORDER BY isbn, leser;
- f) CREATE INDEX i_leser_ab ON t_leser (name DESC);
- g) SHOW INDEX FROM t_leser;
- h) DROP INDEX i_leser_ab ON t_leser;

Vorsicht: Eventuell sind einige Befehle unter älteren Versionen von MySQL nicht verfügbar!