

MySQL Grundlagen

Christian RAINER

MySQL Datenbank

- Speichert beliebige Daten in tabellarischer Form
- Jede Spalte hat einen fixen Datentyp

A	B	C
	<u>id</u> vorname	<u>geburtstag</u>
	1 Hermann	1976-03-18
	2 Otto	1987-11-23
	4 Fritz	1962-08-02
	5 Ulf	1968-10-11

MySQL Datenbank

- Eine Datenbank umfasst beliebig viele Tabellen
- Tabellen können durchsucht und sortiert werden
- Tabellen können miteinander verknüpft werden
- Effizienter als manuell angelegte Dateien

MySQL: Datentypen

- Numerisch:
 - INTEGER: ~4,29 Mrd. Zahlen (32 Bit)
 - TINYINT: 256 Zahlen (8 Bit)
 - SMALLINT: 65.536 Zahlen (16 Bit)
 - MEDIUMINT: ~8,38 Mio. Zahlen (24 Bit)
 - BIGINT: ~9,22e18 Zahlen (64 Bit)
 - FLOAT/DOUBLE: Gleitkommazahlen, bis zu 63 Stellen, nicht exakt
 - DECIMAL: Exakte Kommazahlen, bis zu 65 Stellen

MySQL: Datentypen

- Datum und Zeit:
 - DATE: Format YYYY-MM-DD
 - DATETIME: Format YYYY-MM-DD HH:MM:SS
 - TIME: HH:MM:SS
- Wertebereich für Datum:
 - 1000-01-01 bis 9999-12-31

MySQL: Datentypen

- Strings:
 - VARCHAR: Text bis zu 255 Zeichen (üblich), max. möglich 64 Kbyte
 - TEXT: Text bis zu 64 Kbyte
 - MEDIUMTEXT: Text bis zu 16 Mbyte
 - LONGTEXT: Text bis zu 4 GByte

MySQL: Datentypen

- Binärdaten:
 - VARBINARY: Bis zu 255 Byte (üblich), max. möglich 64 Kbyte
 - BLOB: Bis zu 64 Kbyte
 - MEDIUMBLOB: Bis zu 16 Mbyte
 - LONGBLOB: Bis zu 4 GByte

MySQL Befehle: Tabellenstruktur

- **Tabelle erstellen:**
 - `CREATE TABLE adressen (
 id INT UNSIGNED,
 vorname VARCHAR(255),
 nachname VARCHAR(255),
 geburtstag DATE
);`
- **Struktur ansehen:**
 - `DESCRIBE TABLE adressen;`
- **Tabelle löschen:**
 - `DROP TABLE adressen;`

MySQL Befehle: Tabellenstruktur

- **Spalte hinzufügen:**

- `ALTER TABLE adressen ADD COLUMN strasse
VARCHAR(255) ;`

- **Spalte entfernen:**

- `ALTER TABLE adressen DROP COLUMN strasse;`

Beispiel: Tabellenstruktur

- Erstelle in phpMyAdmin eine neue Datenbank
- Erstelle darin eine neue Tabelle „benutzer“ mit Spalten für „id“, „benutzername“ und „password“
- Füge im Nachhinein vor „password“ eine Spalte „email“ hinzu

MySQL Befehle: Daten

- **Datensätze einfügen:**

- `INSERT INTO adressen SET id = '1', vorname = 'Christian', nachname = 'Rainer';`

- **Alternative Schreibweise:**

- `INSERT INTO adressen (id, vorname, nachname) VALUES ('1', 'Christian', 'Rainer'), ('2', 'Max', 'Muster');`

MySQL Befehle: Daten

- Eine Zeile ändern:

- `UPDATE adressen SET vorname = 'Martin' WHERE id = 2;`

- Mehrere Zeilen auf einmal ändern:

- `UPDATE adressen
SET vorname = 'Max', nachname = 'Mustermann'
WHERE nachname LIKE 'M%';`

MySQL Befehle: Daten

- Eine Zeile löschen:
 - `DELETE FROM adressen WHERE id = 1;`
- Eine Tabelle leeren:
 - `TRUNCATE TABLE adressen;`

MySQL Befehle: Daten

- **Alle Daten ausgeben:**
 - `SELECT * FROM adressen;`
- **Bestimmte Felder ausgeben:**
 - `SELECT vorname, nachname FROM adressen;`
- **Bestimmte Zeilen ausgeben:**
 - `SELECT * FROM adressen
WHERE vorname = 'Max' AND nachname = 'Muster';`
 - `SELECT * FROM adressen
WHERE nachname LIKE '%a%';`

MySQL Befehle: Daten

- Ergebnis sortieren:

- `SELECT * FROM adressen ORDER BY nachname ASC;`

- Ergebnis limitieren:

- `SELECT * FROM adressen LIMIT 0,3;`

- All in one:

- `SELECT vorname, nachname, FROM adressen
WHERE nachname LIKE '%mann%'
ORDER BY nachname DESC LIMIT 0,3;`

Beispiel: Datenverarbeitung

- Füge mindestens 3 Datensätze in die Benutzertabelle ein
- Verändere die E-Mail Adresse eines Datensatzes
- Lösche einen der Benutzer aus der Tabelle
- Zeige nur einen der Benutzer in der Liste
- Versuche die SQL-Befehle auch selbst zu schreiben

MySQL: Index

- Einen Index wird über eine oder mehrere Spalten angelegt
- Beschleunigt die Suche, wenn viele Datensätze vorhanden sind
- Verlangsamt das speichern von Daten geringfügig, da der Index aktuell gehalten werden muss.

MySQL: Indextypen

- PRIMARY
 - Eindeutige und einmalige Kennzeichnung einer Zeile, meist eine automatisch fortlaufende ID
 - Nur einmal pro Tabelle
- UNIQUE
 - Jeder Wert einer Spalte muss einmalig sein
- INDEX
 - Keine Einschränkungen
- FULLTEXT
 - Bietet Features zum Suchen in längeren Texten

MySQL: Index

- **Index hinzufügen**
 - `ALTER TABLE adressen
ADD INDEX vorname_idx (vorname);`
- **Index über mehrere Spalten hinzufügen**
 - `ALTER TABLE adressen
ADD UNIQUE multi_idx (vorname, nachname);`
- **Index entfernen**
 - `ALTER TABLE adressen DROP INDEX vorname_idx;`

MySQL: Auto Inkrement

- Generiert automatisch eindeutige, fortlaufende Ids für jede Zeile
- Tritt meist in Verbindung mit einem PRIMARY Index auf
- ```
CREATE TABLE adressen (
 id INT AUTO_INCREMENT,
 vorname VARCHAR(255),
 PRIMARY KEY (id)
);
```

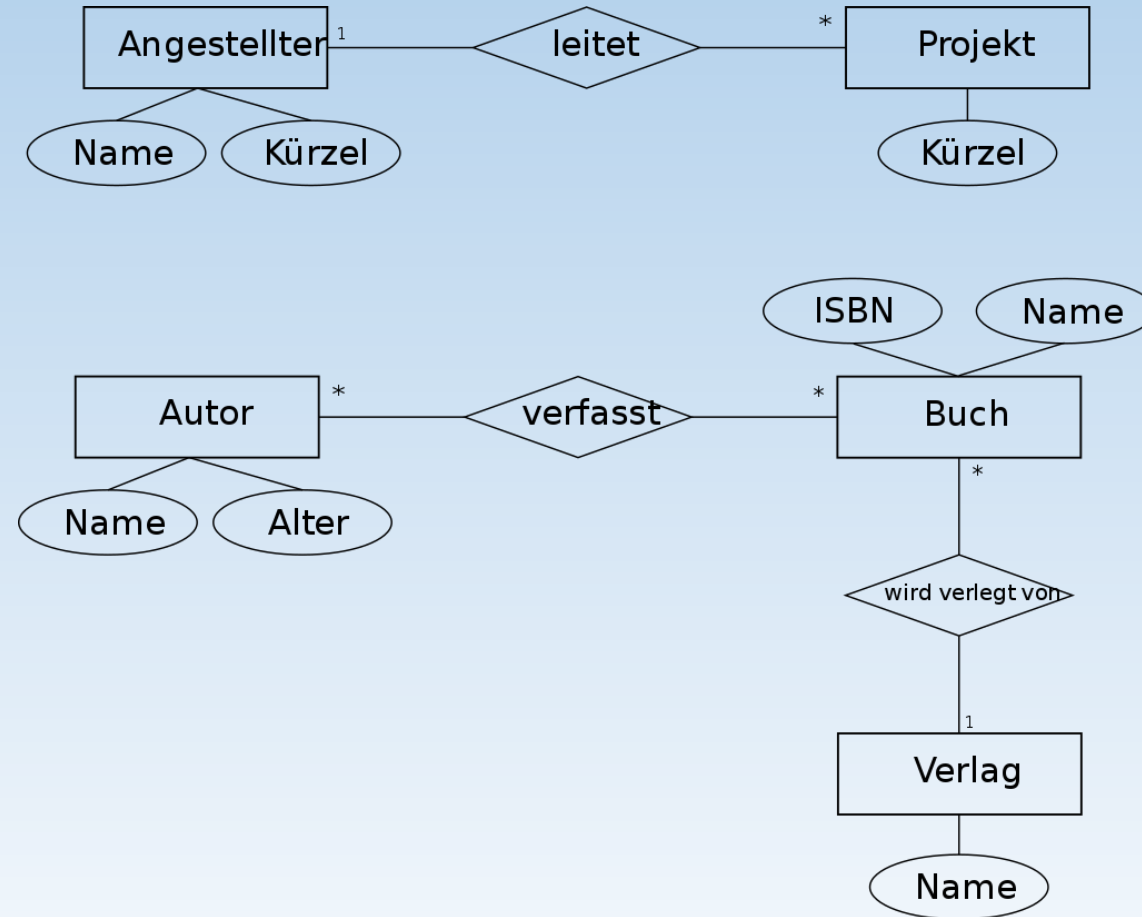
# Beispiel: Index

- Erstelle einen Auto Inkrement auf die id-Spalte der Benutzertabelle
- Erstelle korrekte Indizes auf den übrigen Spalten, sofern sinnvoll

# MySQL: Referenzen

- Verbindet Datensätze zweier Tabellen miteinander
- Werden über id-Spalte verknüpft
- Vermeidet redundant gespeicherte Informationen

# MySQL: Referenzen



# Beispiel: Referenzen

- Erstelle eine Tabelle „rezepte“ mit Spalten „titel“ und „beschreibung“
- Weise jedes Rezept über einen Fremdschlüssel einem Benutzer zu
- Erstelle eine Tabelle „zutaten“
- Erstelle eine weitere Tabelle, die einen n:m Beziehung zwischen „rezepte“ und „zutaten“ herstellt
  - Ein Rezept hat mehrere Zutaten
  - Eine Zutat gehört zu mehreren Rezepten