FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

Datenbanken und Webtechnologien

Dipl. Inf. Marcel Remmy

Meilenstein 1

- Paket 1: HTML Grundlagen
- Paket 2: Data Definition Language

Support-Termine
- KW45
- KW46

Abnahme
- **KW47**

Abgabe bis Sa, 18.11. 23:55 Uhr in Ilias

SQL
Data Definition Language

Paket 2

In diesem Paket lernen Sie, wie Sie ein Entity-Relationship-Diagramm in das Schema und Tabellen eines relationalen DBMS überführen.

Öffnen Sie HeidiSQL und wählen Sie die Datenbank aus Meilenstein 0 (namens Praktikum) in der MariaDB Instanz auf dem ihrem Rechner.

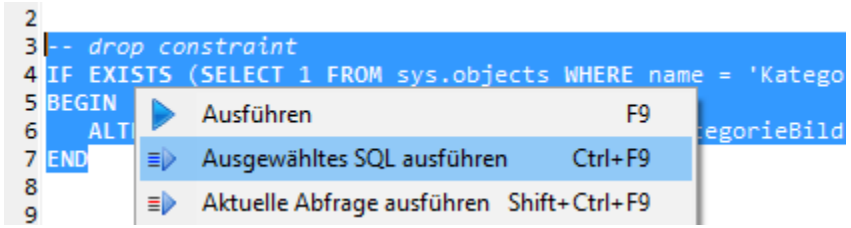
In ILIAS befindet sich das ER-Modell, das wir dieses Paket verwenden werden, und eine SQL-Batchdatei mit ersten CREATE TABLE-Statements für dieses Praktikum. Lesen Sie auch die Business Rules auf den Seiten 6-8. SQL wird in KW46 in der Vorlesung vorgestellt.

Sie sollten – im eigenen Interesse – alle ihre SQL Anweisungen in eigene Textdateien (.txt oder .sql) speichern oder ihrem Praktikums-Dossier hinzufügen. Beantworten Sie auch die Fragen im Dossier.

Paket 2

HeidiSQL[2]

- Mit STRG+T können Sie ein Abfrage-Tab öffnen
- SQL Anweisungen im Abfrage-Tab werden auf der aktuell ausgewählten Datenbank durchgeführt
- Sie können die Ausführung der Auswahl von SQL-Befehlen mit STRG+F9 anfordern, die komplette Abfrage mit F9

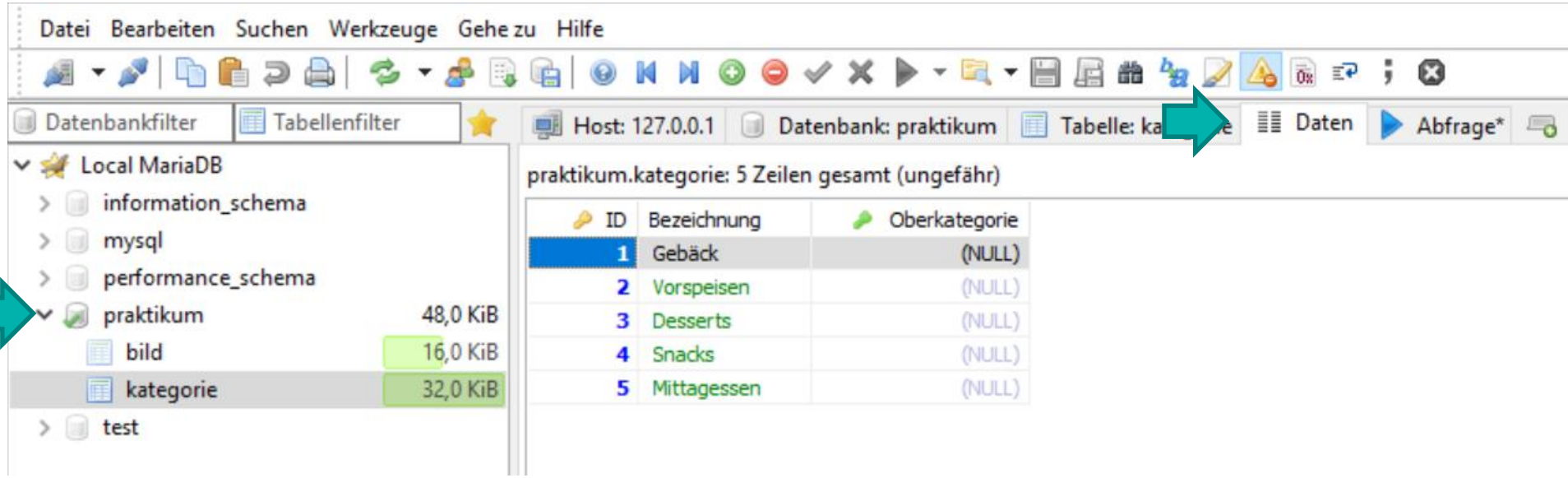


Hinweis: Sie können mit dem HeidiSQL Client die Tabellen zwar auch über Wizards erstellen und die SQL Scripte „reverse engineeren". Jedoch hilft Ihnen die dadurch fehlende Routine nur selten oder überhaupt nicht bei den Aufgaben (und der Klausur!)

Paket 2

HeidiSQL[2]

- Achten Sie darauf, dass stets die richtige Datenbank gewählt ist!
- Neu angelegte Tabellen erscheinen in der Baumansicht links nur, wenn Sie aktualisieren (F5)
- Im Tab „Daten" sehen Sie, welche Daten die ausgewählte Tabelle enthält



Paket 2 -- E-Mensa Business Rules

Das zu implementierende System soll für verschiedene Benutzergruppen Produkte anzeigen und erwerbbar machen. FH-angehörige, die Studierende und/oder Mitarbeiter sein können, und externe Gäste bilden die Benutzergruppen. Alle Benutzer zusammen bilden die Gruppe der FE-Nutzer.

Für alle FE-Benutzer sind Name und E-Mail wichtig und ein Flag, ob der Account aktiv ist. Der Anlege-Zeitpunkt und der Zeitpunkt des letzten Login muss gespeichert werden können.

Das Login geschieht über einen frei vergebbaren Benutzernamen und ein Passwort. In der Datenbank liegen aus Sicherheitsgründen pro Benutzer die Authentifizierungsinformationen als Salt, Hash, Hashing-Verfahren und Anzahl der Stretches vor.

Mitarbeiter weisen eine Mitarbeiternummer und eventuell Telefonnummer und Büro auf. Studierende haben eine Matrikelnummer und einen Studiengang. Für Gäste (nicht Teil der FH-angehörigen) wird der Anlegegrund und ein Ablaufdatum gespeichert.

Paket 2 -- E-Mensa Business Rules

Produkte sind immer Speisen, welche einen Namen, eine Beschreibung, ein Bild und Zutaten aufweisen. Die Zutaten haben jeweils einen Namen, jeweils ein Flag ob die Zutat bio, vegan, vegetarisch, glutenfrei und eine Beschreibung.

Je nach Benutzergruppe können unterschiedliche Bepreisungen der Produkte vorliegen. Produkte sind immer Kategorien zugeordnet. Kategorien haben Bezeichnungen und können Unterkategorien haben. Sowohl die Produkte als auch Kategorien können ein Bild verwenden.

Die Bilddaten werden in der DB gespeichert und müssen darüber hinaus einen Titel und einen Alt-Text aufweisen, und optional eine Bildunterschrift (ein maximal 80 Zeichen langer Text).

Das System soll getätigte Bestellungen speichern können. Es reicht dabei das Bestelldatum, die Anzahl und Art der Produkte und der beteiligte Benutzer.

Paket 2 -- E-Mensa Business Rules

Optionales Feature:

FH-Angehörige dürfen Produkte bewerten, Gäste nicht. Es soll festgehalten werden können, welcher Benutzer wann welches Produkt bewertet hat (pro Nutzer und Produkt nur eine Bewertung).

Bewertungen enthalten eine Note und eine Bemerkung. Außerdem soll mitgeschrieben werden, wie oft eine Bewertung gesehen wurde.

Aufgabe 2.1 – Entitäten überführen (Tabellen)

Die bereitstellte SQL-Datei ist unvollständig und muss von Ihnen ergänzt bzw. geändert werden. Das Script ist so angelegt, dass beim Ausführen die definierten Tabellen gelöscht werden, sofern Sie vorhanden sind (`IF EXIST` Anweisungen). Dies ermöglicht Ihnen ein einfacheres Testen ihrer Tabellendefinitionen.

Erzeugen Sie diese Anweisungen für neu angelegte Tabellen selbst und beachten Sie die Reihenfolge.

Die SQL Data Definition Referenz für MariaDB finden Sie unter [1].

!! In der Abgabe muss Ihr komplettes SQL-Script mehrfach ohne Fehler ausführbar sein. Lassen Sie die Prüfung auf referentielle Integrität zu jeder Zeit intakt (das ist Standard).

Aufgabe 2.1 – Entitäten überführen (Tabellen)

Modifizieren Sie zunächst die `CREATE TABLE` Anweisungen der Entitäten wie folgt ab und achten Sie darauf, die Attribute exakt so zu benennen, wie es das ER-Diagramm vorgibt.

- Wählen Sie immer passende Datentypen (z.B. für `` `Bild`.`Binärdaten` `` bietet sich Binary Large Object an)
- Überprüfen Sie optionale Attribute und Primärschlüssel
- Nicht-optionale Attribute dürfen keine NULL-Werte erlauben
- Verwenden Sie wenn möglich sinnvolle DEFAULT Werte

Wählen Sie für den Primärschlüssel die passenden Datentypen und lassen Sie ggfs. die Surrogate automatisch generieren (mit `AUTO_INCREMENT`)

Aufgabe 2.1 – Entitäten überführen (Tabellen)

Wählen Sie den bestmöglichen Datentyp für jedes Attribut (siehe [3] und [4])

- Definieren Sie abhängig der Domäne des Attributs die reservierte Speichergröße und setzen Sie sich mit den Datentypen auseinander
- 'Flags' sind binär und werden mit dem Datentyp `bool` abgebildet
- Wenn Sie zeitliche Angaben speichern müssen, nutzen Sie dazu auch geeignete temporale Datentypen

Annahmen, die Sie treffen, dokumentieren Sie bitte in einem SQL-Kommentar in der Nähe des `CREATE` Statements

Aufgabe 2.1 – Entitäten überführen (Tabellen)

- Die FE-Benutzer-Entität weist Auth-Attribute auf, die erst in einer späteren Vorlesung erklärt werden, hier jedoch die Angabe der zu speichernden Daten:
    - `Stretch` ist eine positive Ganzzahl
    - `Algorithmus` kann folgende Werte annehmen: 'sha1' oder 'sha256'
    - `Salt` ist immer eine 32 Zeichen lange Zeichenfolge
    - `Hash` ist immer eine 24 Zeichen lange Zeichenfolge
- Die Attribute `` `FE-Nutzer`.`Aktiv` ``, `` `Zutat`.`Vegetarisch` ``, `` `Zutat`.`Vegan` ``, `` `Zutat`.`Glutenfrei` `` und `` `Zutat`.`Bio` `` sind Flags

Aufgabe 2.2 – Relationen

Prinzipiell *kann* jede Relation in eine eigene Tabelle überführt werden, mit je einem Fremdschlüssel für die verknüpften Entitätstypen. Überführen Sie die Relationen in die Datenbank.

- Erweitern Sie die passenden `CREATE TABLE` Statements der Tabellen um `FOREIGN KEY` Constraints

- Die Spezialisierung/Generalisierung (z.B. für FE-Benutzer {FH-angehörige, Gäste} ist bereits mit den vorhandenen Tabellendefinitionen angedeutet.
  Was müssen Sie ändern, um diese besondere Beziehung abzudecken? Welche Vor-/Nachteile hat die von Ihnen ausgewählte Abbildung?

- Was bewirkt das Semikolon am Ende der Anweisung?
  Dokumentieren Sie die kurzen Antworten im Dossier.

Aufgabe 2.3 – Constraints

Neben Primär-/Fremdschlüssel existieren weitere *Constraints*, um zu manipulierende Daten vom DBMS zuvor validieren zu lassen [5].

- Stellen Sie sicher, dass jede *E-Mail* bei FH-angehörige und jede *Matrikelnummer* bei Studenten nur einmal in der entsprechenden Tabelle vorkommt (Eindeutigkeit)

- Ändern Sie das `CREATE TABLE` Statement der *Student*-Tabelle so ab, dass nur *Matrikelnummern* zwischen 10000 und 9999999 eingetragen werden können.

- Das Attribut `bis` der Entität Gast muss dem heutigen Datum entsprechen oder in der Zukunft liegen, wenn der Gast angelegt wird. Das soll das DBMS selbst tun (Default-Wert)

Aufgabe 2.3 – Constraints

Nutzen Sie die Möglichkeit benannter Constraints, wie im Beispiel gezeigt für *KfzEindeutig*.

```
CREATE TABLE Kfz
(
  kennzeichen varchar(10) NOT NULL,
  CONSTRAINT KfzEindeutig UNIQUE (kennzeichen)
);
```

Damit ist es leicht, den Constraint nachträglich zu löschen und zu ändern.

Aufgabe 2.4 – Kaskadiertes Löschen

Legen Sie per INSERT Befehle vier Frontend-Benutzer an.

- Spezialisieren Sie einen FE-Benutzer als *Mitarbeiter* und zwei als *Student*.

Setzen Sie für spezialisierte FE-Benutzer kaskadierendes Löschen um. Es muss also möglich sein, aus der Tabelle `FE-Nutzer` Einträge zu löschen, obwohl sie spezialisiert wurden.

- Wie müssen die spezialisierten Tabellen abgeändert werden, um kaskadierendes Löschen abzubilden? Setzen Sie dies um.

- Testen Sie Ihre Änderung, indem Sie einen FH-Angehörigen löschen:

```
DELETE FROM `FE-Nutzer` WHERE ID= {Nutzer-ID};
```

Paket 2 – In´s Dossier

Dokumentieren Sie im Dossier allgemein, …

- … wie Sie die binären Relationstypen (1:N, N:M) abgebildet haben
- …welche Constraints in MariaDB es gibt und welchem Zweck sie dienen
- … wie Sie den Aufzählungsdatentyp ENUM, den MariaDB unterstützt, per CHECK Constraint auch in anderen DBMS nachbilden könnten
- … was Sie tun mussten, um die Spezialisierung in der Datenbank abbilden zu können (welche INSERT Befehle, welche Reihenfolge)

Paket 2 – Lernziele

Nach Bearbeitung des Pakets haben Sie…

- ✓ ein gegebenes ER-Diagramm in ein Datenbankschema überführt und können abschätzen, inwiefern dessen Vorgaben im DBMS umsetzbar sind
- ✓ Tabellen mit SQL DDL Befehlen erstellt und können erklären…
  - ✓ … wie generalisiert bzw. spezialisiert wird
  - ✓ … wie optionale Attribute, Relationen mit Attributen und PK und FK definiert werden können
  - ✓ … welchen Zweck Constraints haben und wie sie eingesetzt werden
- ✓ einen Einblick in die Datentypen von MariaDB erhalten
- ✓ einen SQL Client (z.B. HeidiSQL) kennengelernt

Paket 2 - Referenzen

SQL DDL in MariaDB
[1] https://mariadb.com/kb/en/library/data-definition/

HeidiSQL Client
[2] https://www.heidisql.com/

SQL Datentypen
[3] https://mariadb.com/kb/en/the-mariadb-library/data-types/
[4] https://www.w3schools.com/sql/sql_datatypes.asp

Constraints
[5] https://www.w3schools.com/sql/sql_constraints.asp

FH Aachen
Fachbereich Elektrotechnik und Informationstechnik

Dipl. Inf. Marcel Remmy
Eupener Straße 70
52064 Aachen
T +49. 241. 6009 52198
remmy@fh-aachen.de
www.m2c-lab.fh-aachen.de

Sprechstunde nach Vereinbarung