# **DBWT Dossier** Denis Behrends

## Lizenzen VS 2017, MariaDB, HeidiSQL?

Visual Studio 2017 wird unter einer EULA(End User License Agreement) für propritäre Software vertrieben.

### Microsoft-Imagine-Abonnementvertrag Direkte Abonnements

<https://de.wikipedia.org/wiki/Visual_Studio>

<https://e5.onthehub.com/WebStore/Common/DisplayOrderMessage.aspx?oiomr=edde59d8-e921-e711-9427-b8ca3a5db7a1&o=e8de59d8-e921-e711-9427-b8ca3a5db7a1&ws=fb848721-3b84-e311-93f9-b8ca3a5db7a1&vsro=8>

MariaDB wird unter GPL und LGPL als freie Software entwickelt und verteilt.

<https://de.wikipedia.org/wiki/MariaDB>

HeidiSQL wird unter der GPL Lizenz als Freie Software bereitgestellt.

<https://de.wikipedia.org/wiki/HeidiSQL>

## Welche IDE kenne/benutze ich?

Eclipse, Emacs, VS Code, Vim, Notepad++

## Welchen Port nutzt MariaDB Server im Standard?

Port 3306

Wie würde der Shop umgesetzt werden müssen (statisches HTML), wenn der Kunde 10 Detailseiten für die Produkte fordern würde?

* Jede Detailseite für ein Produkt müsste neu angelegt werden
* Es entsteht viel redundanter HTML Code
* Die Verlinkungen von der Produkte.html müsste manuell für jedes Produkt bearbeitet werden
* Bei kleiner Designänderung auf den Detailseiten müsste die Veränderung auf allen zehn Seiten vorgenommen werden.

Dropdownelemente im HTML Formularen anbieten und mehrfach verschachteln?

* Ein Dropdown kann mit dem <section> Tag eingeleitet werden.
* Die einzelnen Auswahlmöglichkeiten können mit <option> Tags erstellt werden
* Für eine mehrfache Verschachtelung der Auswahlelemente gibt es das <optgroup> Tag

https://wiki.selfhtml.org/wiki/HTML/Formulare/Auswahllisten

Wie kann man <option> Elemente in einem Dropdown nicht auswählbar machen?

* Indem man <option disabled></option> schreibt.

Welche Attribute sind bei <option> noch nützlich?

* Das „selected“ Attribut kann dazu verwendet werden eine Vorauswahl zu treffen
* Andernfalls wird das oberste Dropdownelement genommen

## Was müssen Sie ändern, um diese besondere Beziehung abzudecken?

* Es müssen drei Tabellen angelegt werden die mit FE-Nutzer über Foreign Keys verbunden sind
* Diese spezialisierten Nutzer sind über den Foreign Key nutzerFK mit der Tabelle FE-Nutzer verbunden

## Was bewirkt das Semikolon am Ende der Anweisung?

* Das Semikolon bewirkt die Beendigung eines Statements
* Führt man im DBMS eine markierte Query aus, dann muss das Semikolon nicht mit markiert werden

## Abbildung von binären Relationstypen (1:N, N:M) ?

Bei 1:N Relationen muss bei Entitäten mit N der Foreign Key gesetzt werden um auf das Objekt aus der anderen Tabelle zu zeigen.

Bei N:M mit Zwischentabelle nie Foreign Keys beider Tabellen miteinander zu einem Unique Key verbindet

## Welche Constraints gibt es und wofür werden Sie verwendet?

* Check Constraint zur überprüfung ob ein Wert bestimmte vorgaben erfüllt
* Not null Constraint zur sicherstellung das kein null wert in Spalte eingetragen werden kann
* Foreign Key Constraint um Relation mit anderer Tabelle herzustellen
* Unique Constraint zur sicherstellung auf einzigartige Spaltenwerte
* Primary Key Constraint zur erstellung eines eindeutigen schlüssels
* Index Constraint zur erstellung von Indizes
* Default Constraint zur sicherstellung das immer ein default wert gesetzt wird

## Aufzählungstyp ENUM aus MariaDB in anderen DBMS mit CHECK nachbilden?

Enum kann nachgebildet werden mit einer Aufzählung der CHECK Constraints

Bsp.

**CONSTRAINT** `test` **CHECK**(spalte='a'),**CHECK**(spalte='b'),**CHECK**(spalte='c')

## Wie Spezialisierungen in DBMS abgebildet werden müssen?

* Speizialisierung muss beim Anlegen der Tabellen immer nach dem Anlegen der Generalisierung passieren
* Beim einfügen in die Tabellen müssen die Daten aus den generelleren Tabellen vor den spezielleren Tabellen erstellt werden
* Bei drop oder truncate muss zuerst die spezielleren Datensätze gelöscht werden (es sei denn die speziellen werden direkt mit gelöscht durch ein on delete cascade)

## Wozu dienen die SQL Funktionen COALESCE, IFNULL und NULLIF?

* COALESCE gibt das erste Element das nicht null ist zurück  
  Bsp.  
  SELECT COALESCE(NULL, NULL, NULL, 'W3Schools.com', NULL, 'Example.com');  
  gibt W3Schools.com zurück
* IFNULL gibt eine alternative an wenn der erste Ausdruck null geliefert hat.  
  Bsp.  
  SELECT IFNULL(preis, "Kostenlos") FROM product;
* NULLIF(expr1, expr2) gibt null zurück wenn expr1 und expr2 gleich sind und gibt expr1 zurück wenn nicht.

## Wozu dienen die Schlüsselwörter ALL und ANY bei Subqueries und

## wie kann man sie einsetzen?

* ALL und ANY können in where und having von sql statements verwendet werden.
* Columnname Operator [ANY|ALL] (Subquery)   
    
  Operator kann <, >, =, <>, <=, >=
* Bei ALL muss alle Results von Subquery mit dem vergleichenden Wert übereinstimmen
* Bei ANY muss mind. Eins übereinstimmen.

## Wofür wird having verwendet?

Um Aggregatfunktionen bei der Einschränkung zu verwenden muss having verwendet werden.

## Erstellung von Nutzern einer Datenbank

**USE** `praktikum`;

*/\* Aktiviere Sitzung "local" \*/*

**SHOW** **VARIABLES** **LIKE** 'skip\_name\_resolve';

**FLUSH** **PRIVILEGES**;

**SELECT** `user`, `host`, `authentication\_string` **FROM** `mysql`.`user`;

*/\* Erstellt Nutzer webapp der sich über localhost mit dem angegebenen Passwort einloggen kann \*/*

**CREATE** **USER** 'webapp'@'localhost' **IDENTIFIED** **BY** '[passwort]';

**GRANT** **USAGE** **ON** \*.\* **TO** 'webapp'@'localhost';

*/\* Gewährt dem Nutzer Zugriff auf die Datenbank Praktikum mit angegebenen Zugriffrechten über die localhost Verbindung \*/*

**GRANT** **SELECT**, **EXECUTE**, **SHOW** **VIEW**, **ALTER**, **ALTER** **ROUTINE**, **CREATE**, **CREATE** **ROUTINE**, **CREATE** **TEMPORARY** **TABLES**, **CREATE** **VIEW**, **DELETE**, **DROP**, **EVENT**, **INDEX**, **INSERT**, **REFERENCES**, **TRIGGER**, **UPDATE**, **LOCK** **TABLES** **ON** `praktikum`.\* **TO** 'webapp'@'localhost';

*/\* Schreibt die neuen Privileges in die Informationshema Datenbank\*/*

**FLUSH** **PRIVILEGES**;

**SHOW** **GRANTS** **FOR** 'webapp'@'localhost';

## GRANT Recht

* Das GRANT Recht ermöglicht dem erstellten Nutzer selber Nutzer zu erstellen und den erstellten Nutzern Rechte zu vergeben.
* Da der Benutzer webapp keine Benutzer anlegen soll fällt das Recht somit weg.
* Diese Berechtigung sollte nur an Datenbankadministratoren vergeben werden.

## Was ist alles zu tun um sowohl Anmeldungen als auch Anmeldeversuche

## anzeigen zu können?

* In der Datenbank müsste Fe-Nutzer um eine Spalte Anmeldeversuche vom Typ Integer hinzugefügt wird
* Im FeNutzer Model muss eine Methode updateLoginVersuche() hinzugefügt werden die die Spalte Anmeldeversuche incrementiert und ein Feld Versuche
* Diese Methode wird in der Action Login vom Controller User aufgerufen, die ausgeführt wird wenn der Anmeldeversuch ausgeführt wird.
* Im Controller wird der Wert der FENutzer Instanz ausgelesen und im View angezeigt sobald der Nutzer eingeloggt ist.

## Die Kriterien für Anmeldungen der Art "Backend"

Ein Backend-Nutzer wird aus der Menge der FE-Nutzer zu Backend spezialisiert. Ich habe eine Tabelle für be-nutzer angelegt und im model einfach eine BE\_Nutzer Klasse angelegt die von FE\_Nutzer erbt.

Diese Klasse wurde um das BEid Feld erweitert und eine Methode die testet ob der Benutzer Administratorrechte hat.

Diese werden in der Controller Klasse Admin verwendet