

Projekt Aufgaben PHP, MySQL für 2 Tage

1. Schreibt bitte eine SQL-Abfrage (SELECT), die eine Liste mit den Namen aller Autoren (Vor- und Nachname zusammen), und der Summe der Preise aller ihnen zugehörigen Movies ausgibt. Benutzt hier für: INNER JOIN, GROUP BY, SUM-Funktion.

Siehe:

https://www.w3schools.com/mysql/mysql_join_inner.asp

https://www.w3schools.com/mysql/mysql_groupby.asp

https://www.w3schools.com/mysql/mysql_count_avg_sum.asp

2. Erstellt im Movie-Start Projekt analog zu den Autoren die gesamte CRUD-Funktionalität, um über einen neuen Menüpunkt (Movies) alle Movies ausgeben und verwalten zu können.

Also Listen-, Einzelansicht, Formular um neue Movies anzulegen oder bestehende ändern zu können. Löschen von Movies. Alle Schreibaktionen setzen einen eingeloggten User voraus.

3. In unseren Controller-Klassen greifen wir, um mit der DB zu arbeiten, immer auf vorbereitete Model-Klassen zurück, um sie in jeder Controller-Funktion in 2 Schritten zu benutzen:

1. Instanziierung des Model als Objekt-Variable (z.B. `$model = new Author`)
2. Benutzung des Model-Objekts, um Funktionen auszuführen (z.B.: `$data = $model->all()`)

In der Controller-Klasse (Controller/Controller.php) ist bereits eine Klassen-Variable `$model` vorhanden (Ohne Wertzuweisung, also NULL). Versucht eine Möglichkeit zu finden, die Controller-Klassen so anzupassen, dass man statt der 2. Schritte wie oben, nur noch folgende Syntax verwenden braucht, um z.B alle Datensätze auszulesen: `$data = $this->model->all()`;

4. OPTIONAL: Versucht in der Klasse `inc/Model.php` die noch leere Klassen-Funktion (Methode) „insert“ funktionstüchtig zu machen. Als Parameter soll sie folgendes übergeben bekommen:

- assoziatives Array aller Daten, die per INSERT neu in die Tabelle (`$this->table`) eingetragen werden sollen.

Baut anhand dieser Parameter den notwendigen SQL-String zusammen und übergeben diesen inklusive Daten-Parameter an die MyDB-Funktion **`prepareAndExecute($query, $options)`**. Beachtet hierbei aber die Platzhalter-Syntax für prepared Statements (Tip: Variante mit Doppelpunkt und KeyName verwenden, also nicht das Fragezeichen).

Nützlich PHP-Funktionen hierfür:

<https://www.php.net/manual/en/function.array-keys>

<https://www.php.net/manual/en/function.implode>

5. OPTIONAL: Versucht in der Klasse inc/Model.php die noch leere Klassen-Funktion (Methode) „update“ funktionstüchtig zu machen. Als Parameter soll sie folgendes übergeben bekommen:

- assoziatives Array aller Daten, die per UPDATE in der Tabelle (\$this->table) geändert werden sollen.
- id

Baut anhand dieser Parameter den notwendigen SQL-String zusammen und übergebt diesen inklusive Daten-Parameter an die MyDB-Funktion **prepareAndExecute(\$query, \$options)**. Beachtet hierbei aber die Platzhalter-Syntax für prepared Statements (Tip: Variante mit Doppelpunkt und KeyName verwenden, also nicht das Fragezeichen).

Nützlich PHP-Funktionen hierfür:

<https://www.php.net/manual/en/function.array-keys>

<https://www.php.net/manual/en/function.implode>