

Notizverwaltung

Schuljahr 2014/15

Christoph Hackenberger

Inhaltsverzeichnis

1. Aufgabenstellung.....	3
2. Zeitaufzeichnung	4
3. Installation Propel 2.0	4
Voraussetzungen	4
Setup	4
4. Design	5
5. Build a Project	5
Describing the Database as XML Schema	5
Many-to-Many Beziehung	7
Configuration	7
Build SQL-Files	7
SQL-Files erzeugen:.....	7
6. Quellen	8

1. Aufgabenstellung

Notizen sollen in einer Software verwaltet werden

Notiz = Betreff + Text + Datum + Besitzer + Projekt (optional)

Projekt = Name + Startdatum + Enddatum + Projektteilnehmer (mind. 1)

Person = Name + Beschreibung

Von Notizen kann es spezialisierte Arten geben:

ToDoNotiz = Notiz + Status [erledigt | offen | in Arbeit] + Priorität

Rezept = Notiz + geordnete Liste von Subnotizen (enthalten die einzelnen Kochanweisungen)

Aufgaben

1. Umsetzung in UML-Klassendiagramm + ERD (oder UML für Datenstruktur)
2. Erstelle eine schema.xml-Datei für Propel aus den Modellen. Vergleiche den Output von propel sql:build mit deinem Modell.
3. Sorge dafür, dass Änderungen an den Tabellen mit Zeitstempeln protokolliert werden (Zeitpunkt der Erstellung, Zeitpunkt des letzten Updates).
In deiner Implementierung darf es auch nicht erlaubt sein, dass das Startdatum eines Projektes nach dem Enddatum liegt. Beschränke den Betreff-Text einer Notiz auf 100 Zeichen und beschränke den ToDo-Status auf die genannten drei Werte.
Schau dir dafür Behaviors genauer an.
4. Wähle eine Form der Vererbungsumsetzung, die Propel anbietet und dokumentiere ausführlich, warum gerade diese deiner Meinung nach für die Aufgabenstellung am Besten geeignet ist.
5. Generiere die PHP-Klassen für dein Modell
6. Verwende diese in einer einfachen PHP-Anwendung, die es erlaubt Notizen hinzuzufügen, zu ändern und zu löschen:
 - eine Listenansicht soll alle Notizen zeigen (mit all Ihren Feldern). Die Detailinformationen des Projekts bzw. des Besitzers sollen ebenfalls gezeigt werden.
 - bei Rezepten soll man per Klick zu der Liste der verknüpften Kochanweisungen kommen (in der richtigen Reihenfolge gelistet!). Hier muss es möglich sein, weitere Kochanweisungen hinzuzufügen (Auswahl aus bestehenden Notizen) oder zu löschen
 - Projekte und Personen dürfen fix in der Datenbank angelegt werden (Insert-Skript dem Projekt hinzufügen), damit sie bei der Notizerstellung/bearbeitung ausgewählt werden können - es sind keine eigenen Formulare notwendig.
 - KEINE HTML-Formulare selbst erstellen

- Validieren der Eingabefelder ist Pflicht
- verwende dafür Libraries wie Symfony Form oder PHP Form Builder Class

Die Verwendung von Symfony-Komponenten bietet sich besonders an, da Propel selbst einige davon verwendet und es etliche Tutorials und Hilfestellungen zur Verknüpfung von Symfony und Propel gibt.

2. Zeitaufzeichnung

	Geplant	Aktuell
Installation Propel	0:45	1:00
Design	0:20	0:25
Erstellen des Schema Behaviors	0:15	0:15
	1:00	
Summe		

3. Installation Propel 2.0

Voraussetzungen

- PHP 5.4 (Cli & PDO Drivers für das jeweilige DBMS)
- Database (MySQL, PostgreSQL, etc.)
- Config : uses in the source code to manage and validate configuration.
- Console : which manage the generators propel uses.
- Yaml
- Validator : a way you manage validations with Propel.
- Finder : uses in the source code to manage the files.

Setup

Erstellen eine Datei composer.json

```
{
    "require": {
        "propel/propel": "~2.0@dev"
    }
}
```

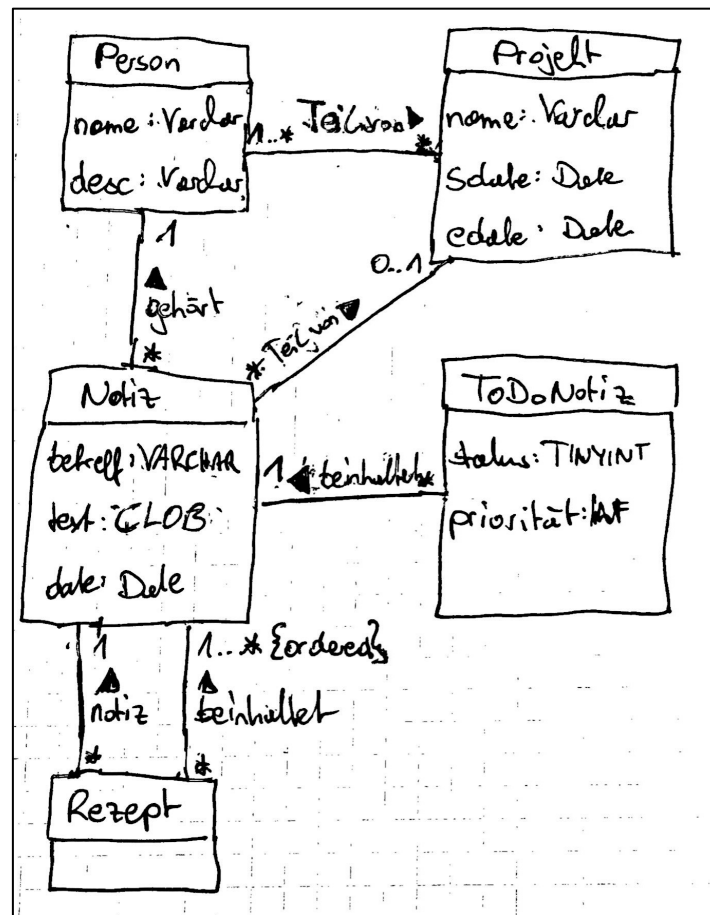
Download Composer mit dem Befehl:

wget http://getcomposer.org/composer.phar

Installation

php composer.phar install

4. Design



5. Build a Project

Describing the Database as XML Schema

```

<?xml version="1.0" encoding="UTF-8"?>
<database name="notizverwaltung" defaultIdMethod="native">
  <behavior name="timestampable" />
  <table name="notiz" phpName="Notiz">
    <column name="id" type="integer" required="true" primaryKey="true"
    autoIncrement="true" />
    <column name="besitzer_id" type="integer" required="true" />
    <column name="projekt_id" type="integer" required="false" />
    <column name="name" type="varchar" required="true" size="255" />
    <column name="text" type="clob" required="true" />
    <foreign-key foreignTable="person" phpName="Besitzer">
      <reference local="besitzer_id" foreign="id" />
    </foreign-key>
    <foreign-key foreignTable="projekt">
      <reference local="projekt_id" foreign="id" />
    </foreign-key>
  </table>
  <table name="projekt" phpName="Projekt">

```

```

    <column name="id" type="integer" required="true" primaryKey="true"
autoIncrement="true" />
    <column name="name" type="varchar" required="true" size="255" />
    <column name="sdate" type="date" required="true" />
    <column name="edate" type="date" required="true" />
</table>
<table name="person" phpName="Person">
    <column name="id" type="integer" required="true" primaryKey="true"
autoIncrement="true" />
    <column name="name" type="varchar" required="true" size="150" />
    <column name="desc" type="varchar" required="true" size="255" />
</table>
<table name="todo_notiz" phpName="ToDoNotiz">
    <column name="id" type="integer" required="true" primaryKey="true"
autoIncrement="true" />
    <column name="notiz_id" type="integer" required="true" />
    <column name="status" type="tinyint" required="true" />
    <column name="prior" type="integer" required="true" />
    <foreign-key foreignTable="notiz" >
        <reference local="notiz_id" foreign="id" />
    </foreign-key>
</table>
<table name="rezept" phpName="Rezept">
    <column name="id" type="integer" required="true" primaryKey="true"
autoIncrement="true" />
    <column name="notiz_id" type="integer" required="true" />
    <foreign-key foreignTable="notiz" >
        <reference local="notiz_id" foreign="id" />
    </foreign-key>
</table>
<table name="person_projekt" isCrossRef="true" >
    <column name="person_id" type="integer" primaryKey="true" />
    <column name="projekt_id" type="integer" primaryKey="true" />
    <foreign-key foreignTable="person" >
        <reference local="person_id" foreign="id" />
    </foreign-key>
    <foreign-key foreignTable="projekt" >
        <reference local="projekt_id" foreign="id" />
    </foreign-key>
</table>
<table name="rezept_notiz" isCrossRef="true" >
    <column name="rezept_id" type="integer" primaryKey="true" />
    <column name="notiz_id" type="integer" primaryKey="true" />
    <foreign-key foreignTable="rezept" >
        <reference local="rezept_id" foreign="id" />
    </foreign-key>
    <foreign-key foreignTable="notiz" >
        <reference local="notiz_id" foreign="id" />
    </foreign-key>
</table>
</database>

```

Many-to-Many Beziehung

isCrossRef Attribut im `<table>` Tag gibt an, dass die Tabelle eine cross-reference Tabelle ist.

Configuration

Config-Datei `/config/propel.php`

```
<?php

return [
    'propel' => [
        'database' => [
            'connections' => [
                'notizverwaltung' => [
                    'adapter' => 'mysql',
                    'classname' => 'Propel\Runtime\Connection\ConnectionWrapper',
                    'dsn' => 'mysql:host=localhost;dbname=notizverwaltung',
                    'user' => 'root',
                    'password' => 'password',
                    'attributes' => []
                ]
            ]
        ],
        'runtime' => [
            'defaultConnection' => 'notizverwaltung',
            'connections' => ['notizverwaltung']
        ],
        'generator' => [
            'defaultConnection' => 'notizverwaltung',
            'connections' => ['notizverwaltung']
        ]
    ]
];
```

Build SQL-Files

SQL-Files erzeugen

`propel sql:build`

Tabellen erzeugen

`propel sql:insert`

Sollte eine Fehlermeldung kommen das die benötigten PDO Drivers nicht gefunden werden können müssen diese für das jeweils verwendete DBMS nach installiert werden.
zB.: `sudo apt-get install php5-mysql`

6. Behaviors

Propel bietet ein Konzept namens Behaviors an. Mit Behaviors ist es möglich wiederverwendbares Verhalten für Objekte zu definieren.

So kann man zB.: mit dem Behavior `<behavior name="timestampable" />` auf database Ebene im Schema allen Tabellen Columns zum speichern des Erstellungsdatum und das Datum des letzten Updates. Außerdem werden so auch gleich die benötigten php Funktionen in der

7. Quellen

- [1] Propel Installations Dokumentation, verfügbar unter:
<http://propelorm.org/documentation/01-installation.html>, [abgerufen am 26.11.14]
- [2] Propel Building Dokumentation, verfügbar unter:
<http://propelorm.org/documentation/02-buildtime.html>, [abgerufen am 26.11.14]
- [3]