KNW M-128 Datenbank Administrieren

IAP 12-16 B Jérôme Röthlisberger

Kompetenz-Nachweis Auftrag, Eine Datenbank der Gewerbeschule muss administriert werden. Dazu werden entsprechende Stored-Procedures entwickelt

**Inhaltsverzeichnis**

[Projekt-Info 3](#__RefHeading___Toc6523_192616982)

[Dokument-Historie 3](#__RefHeading___Toc6525_192616982)

[Projekt Zusammenfassung 3](#__RefHeading___Toc6527_192616982)

[Ausgangslage 3](#__RefHeading___Toc6529_192616982)

[Umsetzung 3](#__RefHeading___Toc6531_192616982)

[1Teilprojekt Datenbank Migration 3](#__RefHeading___Toc6533_192616982)

[1.1Definition der Namenskonvention 3](#__RefHeading___Toc6535_192616982)

[1.1.1Globale Konvention 3](#__RefHeading___Toc6537_192616982)

[1.1.2Zusätzliche Definitionen 4](#__RefHeading___Toc6539_192616982)

[1.1.3Datenkatalog 5](#__RefHeading___Toc6541_192616982)

[1.1.4Tabelle Klassen 5](#__RefHeading___Toc6543_192616982)

[1.1.5Tabelle Lehrbetriebe 5](#__RefHeading___Toc6545_192616982)

[1.1.6Tabelle Lernende 6](#__RefHeading___Toc6547_192616982)

[1.1.7Tabelle Berechtigungen\_log 6](#__RefHeading___Toc6549_192616982)

[1.1.8Tabelle module 7](#__RefHeading___Toc6551_192616982)

[1.1.9Tabelle noten 7](#__RefHeading___Toc6553_192616982)

[1.1.10Tabelle Fachrichtungen 7](#__RefHeading___Toc6555_192616982)

[1.2Stored Procedure sp\_migration\_starten 8](#__RefHeading___Toc6557_192616982)

[1.3Stored Procedure Quelltext 8](#__RefHeading___Toc6559_192616982)

[2Teilprojekt Ortschaftstbelle erstellen 15](#__RefHeading___Toc6561_192616982)

[2.1Stored Procedure sp\_orte\_extrahiern 15](#__RefHeading___Toc6563_192616982)

[2.1.1Datenkatalog 15](#__RefHeading___Toc6565_192616982)

[2.1.2Tabelle Orte 15](#__RefHeading___Toc6567_192616982)

[2.1.3Tabelle Lehrbetriebe 16](#__RefHeading___Toc6569_192616982)

[2.1.4Tabelle Lernende 16](#__RefHeading___Toc6571_192616982)

[2.2Stored Procedure Quelltext 17](#__RefHeading___Toc6573_192616982)

[3Teilprojekt Benutzer erstellen 20](#__RefHeading___Toc6575_192616982)

[3.1Stored Procedure sp\_benutzer\_erstellen 20](#__RefHeading___Toc6577_192616982)

[3.2Aufrufbeispiele 20](#__RefHeading___Toc6579_192616982)

[3.3Eingabe Validation 20](#__RefHeading___Toc14171_192616982)

[3.4Datenkatalog 21](#__RefHeading___Toc6581_192616982)

[3.4.1Tabelle Berechtigungen\_log 21](#__RefHeading___Toc6583_192616982)

[3.5Stored Procedure Quelltext 22](#__RefHeading___Toc6585_192616982)

[4Teilprojekt Lernende archivieren 24](#__RefHeading___Toc6587_192616982)

[4.1Stored Procedure sp\_lernende\_archivieren 24](#__RefHeading___Toc6589_192616982)

[4.2Datenkatalog 24](#__RefHeading___Toc6591_192616982)

[4.2.1Tabelle schoolinfo\_archiv.lernende\_archiv 24](#__RefHeading___Toc6593_192616982)

[4.3Stored Procedure Quelltext 25](#__RefHeading___Toc6595_192616982)

Projekt-Info

|  |  |
| --- | --- |
|  |  |
| *Autor* | Jérôme Röthlisberger |
| *E-Mail Adresse* | [jerome.roethlisberger@gibmit.ch](mailto:jerome.roethlisberger@gibmit.ch) |
| *Projektmanager* | T. Bögli, Lehrer der GIBM |
| *Projekt E-Mail* | [t.boegli@gibmit.ch](mailto:t.boegli@gibmit.ch) |
| *Abgabe-Datum* | 17.05.2016 |
|  |  |

Dokument-Historie

|  |  |  |  |
| --- | --- | --- | --- |
| *Version* | *Datum* | *Autor* | *Änderung* |
| *1* | *15.05.16* | *ROJE* | *Erstellt* |
| *2* | *16.05.16* | *ROJE* | *Bearbeitet* |
| *3* | *17.05.2016* | *ROJE* | *Abgabe* |

Projekt Zusammenfassung

Ausgangslage

Die Applikation für Schuladministration der IT-Abteilung der Gewerblich Industriellen Berufsschule Muttenz ist soweit gewachsen dass eine Migration in eine neue Datenbank nötig ist.

Damit die Weiterentwicklung gewährleistet werden kann müssen Teile der Datenbank verändert werden.

Zu diesem Zweck werden entsprechende Stored-Procedures entwickelt mit denen einen Datenbank-Migration durchgeführt werden kann.

Umsetzung

Das Projekt ist in 4 Aufgabenbereiche gegliedert. Jedes dieser Aufgabenbereiche befasst sich mit einem Stored-Procedure.

Folgende Stored-Procedures werden umgesetzt

* sp\_migration\_starten
* sp\_ortschaft\_extrahieren
* sp\_benutzer\_erstellen
* sp\_lernende\_archivieren

1. Teilprojekt Datenbank Migration

Befasst sich mit Stored-Procedure: sp\_migration\_starten

* 1. Definition der Namenskonvention
     1. Globale Konvention
* Datenbank Name
  + schoolinfo\_neu
* Tabellennamen
  + Alphanumerisch
  + nur die Zeichen a-z und 0-9
  + Snake-Case sprich, Lehrzeichen durch Unterstriche ersetzt
  + kein Präfix
  + plural
  + deutsch
* Kolumnen
  + Alphanumerisch
  + nur die Zeichen a-z und 0-9
  + Snake-Case sprich, Lehrzeichen durch Unterstriche ersetzt
  + kein Präfix
  + singular
  + deutsch
* Tabellenaufbau
  + IMMER nur eine Kolumne ist Primärschlüssel
  + Primärschlüssel-Kolumne ist immer
    - Name: id
    - Datentyp: INTEGER(10) NOT NULL AUTOINCREMENT
  + Fremdschlüssel Kolumne sind immer
    - Konvention: Kolumn name ist IMMER singular.
    - Name: fremdtabelle\_id  
       fremndtabelle\_kontext\_id
    - Datentyp: INTEGER(10)
  + Fremdschlüssel
    - Konvention: Fremdschlüssel sind IMMER singular und zeigen IMMER auf eine id.
    - Name eigentabelle\_fremdtabelle\_id  
       eigentabelle\_fremdtabelle\_kontext\_id
    1. Zusätzliche Definitionen

Kolumnen die Text enthalten werden grundsätzlich auf VARCHAR(50) gesetzt. Ausnahmen entstehen bei Felder die in der Quelldatenbank mit mehr als 50 Zeichen definiert wurden. Diese werden grundsätzlich auf VARCHAR(255) gesetzt.

Postleitzahlen werden auf VARCHAR(8) gesetzt um somit den deutschen Postleitzahlen Rechnung zu tragen.

Situativ können auch VARCHAR(2) verwendet werden.

* + 1. Datenkatalog

Datenbank: schoolinfo\_neu

* + 1. Tabelle Klassen

|  |  |  |  |
| --- | --- | --- | --- |
| ***Tabelle Klassen*** | | | |
| **id** | INT(10) | NOT NULL | AUTO INCREMENT |
| **lehrer\_id** | INT(10) |  | DEFAULT NULL |
| **name** | VARCHAR(50) | NOT NULL |  |
| **beschreibung** | VARCHAR(255) |  | DEFAULT NULL |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***SCHLÜSSEL*** | ***Name*** | ***Referenztabelle*** | ***Feld*** | ***Referenz Feld*** |
| **PKY** | id |  |  | id |
| **FKY** | klassen\_lehrer\_id | lehrer | id | lehrer\_id |
|  |  |  |  |  |

* + 1. Tabelle Lehrbetriebe

|  |  |  |  |
| --- | --- | --- | --- |
| ***Tabelle lehrbetriebe*** | | | |
| **id** | INT(10) | NOT NULL | AUTO INCREMENT |
| **name** | VARCHAR(50) |  | DEFAULT NULL |
| **strasse** | VARCHAR(50) | NOT NULL |  |
| **haus\_nr** | VARCHAR(50) |  | DEFAULT NULL |
| **plz** | VARCHAR(8) | NOT NULL |  |
| **ort** | VARCHAR(50) | NOT NULL |  |
| **kanton\_code** | VARCHAR(2) |  | DEFAULT NULL |
| **land\_code** | VARCHAR(2) |  | DEFAULT NULL |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***SCHLÜSSEL*** | ***Name*** | ***Referenztabelle*** | ***Feld*** | ***Referenz Feld*** |
| **PKY** | id |  |  | id |
|  |  |  |  |  |

* + 1. Tabelle Lernende

|  |  |  |  |
| --- | --- | --- | --- |
| ***Tabelle lernende*** |  |  |  |
| **id** | INT(10) | NOT NULL | AUTO INCREMENT |
| **anrede** | VARCHAR(50) |  | DEFAULT NULL |
| **name** | VARCHAR(50) | NOT NULL |  |
| **vorname** | VARCHAR(50) | NOT NULL |  |
| **geschlecht** | VARCHAR(2) | NOT NULL | “M“ = Männlich,  “F“ = Weiblich |
| **klasse\_id** | INT(10) | NOT NULL |  |
| **ist\_bm** | TINYINT(1) | NOT NULL | 0 = Nein  1 = Ja |
| **fachrichtung\_id** | INT(10) | NOT NULL |  |
| **lehrbetrieb\_id** | INT(10) | NOT NULL |  |
| **strasse** | VARCHAR(50) | NOT NULL |  |
| **plz** | VARCHAR(8) | NOT NULL |  |
| **ort** | VARCHAR(50) | NOT NULL |  |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***SCHLÜSSEL*** | ***Name*** | ***Referenztabelle*** | ***Feld*** | ***Referenz Feld*** |
| **PKY** | id |  |  |  |
| **FKY** | lernende\_fachrichtung\_id | fachrichtungen | id | fachrichtung\_id |
| **FKY** | lernende\_klasse\_id | klassen | id | klasse\_id |
| **FKY** | lernende\_lehrbetrieb\_id | lehrbetriebe | id | lehrbetrieb\_id |
|  |  |  |  |  |

* + 1. Tabelle Berechtigungen\_log

|  |  |  |  |
| --- | --- | --- | --- |
| ***Tabelle berechtigungen\_log*** | | | |
| **id** | INT(10) | NOT NULL | AUTO INCREMENT |
| **benutzer** | VARCHAR(50) | NOT NULL |  |
| **zeitpunkt** | DATETIME | NOT NULL | DEFAULT CURRENT\_TIMESTAMP |
| **grund** | VARCHAR(50) | NOT NULL |  |
| **berechtigung** | VARCHAR(50) | NOT NULL |  |
| **tabelle** | VARCHAR(50) | NOT NULL |  |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***SCHLÜSSEL*** | ***Name*** | ***Referenztabelle*** | ***Feld*** | ***Referenz Feld*** |
| **PKY** | id |  |  | id |
|  |  |  |  |  |

* + 1. Tabelle module

|  |  |  |  |
| --- | --- | --- | --- |
| ***Tabelle module*** | | | |
| **id** | INT(10) | NOT NULL | AUTO INCREMENT |
| **benutzer** | VARCHAR(50) | NOT NULL |  |
| **zeitpunkt** | DATETIME | NOT NULL | DEFAULT CURRENT\_TIMESTAMP |
| **grund** | VARCHAR(50) | NOT NULL |  |
| **berechtigung** | VARCHAR(50) | NOT NULL |  |
| **tabelle** | VARCHAR(50) | NOT NULL |  |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***SCHLÜSSEL*** | ***Name*** | ***Referenztabelle*** | ***Feld*** | ***Referenz Feld*** |
| **PKY** | id |  |  | id |
|  |  |  |  |  |

* + 1. Tabelle noten

|  |  |  |  |
| --- | --- | --- | --- |
| ***Tabelle noten*** | | | |
| **id** | INT(10) | NOT NULL | AUTO INCREMENT |
| **lernende\_id** | INT(10) | NOT NULL |  |
| **modul\_id** | INT(10) | NOT NULL |  |
| **erf\_note** | DOUBLE(15,2) | NOT NULL |  |
| **erf\_datum** | DATETIME | NOT NULL |  |
| **knw\_note** | DOUBLE(15,2) | NOT NULL |  |
| **knw\_datum** | DATETIME | NOT NULL |  |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***SCHLÜSSEL*** | ***Name*** | ***Referenztabelle*** | ***Feld*** | ***Referenz Feld*** |
| **PKY** | id |  |  | id |
| **FKY** | noten\_lernende\_id | lernende | id | lernende\_id |
| **FKY** | noten\_modul\_id | module | id | modul\_id |
|  |  |  |  |  |

* + 1. Tabelle Fachrichtungen

|  |  |  |  |
| --- | --- | --- | --- |
| ***Tabelle fachrichtungen*** | | | |
| **id** | INT(10) | NOT NULL | AUTO INCREMENT |
| **name** | VARCHAR(50) | NOT NULL |  |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***SCHLÜSSEL*** | ***Name*** | ***Referenztabelle*** | ***Feld*** | ***Referenz Feld*** |
| **PKY** | id |  |  | id |
|  |  |  |  |  |

* 1. Stored Procedure sp\_migration\_starten
* Name
  + sp\_migration\_starten
* Eingabeparameter
  + keine
* Ausgabeparameter
  + keine
* Resultat
  + Datenbank schoolinfo\_neu ist erstellt
  + Folgende Tabellen sind erstellt
    - klassen
    - lehrbetriebe
    - lernende
    - berechtigunen\_log
    - module
    - noten
    - fachrichtungen
  + Daten aus Datenbank schoolinfo12802016 sind in die Tabellen von schoolinfo\_neu importiert.
  1. Stored Procedure Quelltext

***DELIMITER //***

**CREATE** **PROCEDURE** sp\_migration\_starten

**BEGIN**

**DROP** **DATABASE** **IF** **EXISTS** `schoolinfo\_neu`;

**CREATE** **DATABASE** **IF** **NOT** **EXISTS** `schoolinfo\_neu`

**DEFAULT** **CHARACTER** **SET** latin1;

**USE** `schoolinfo\_neu`;

**SET** **SESSION** SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

**START** **TRANSACTION**;

**DROP** **TABLE** **IF** **EXISTS** `klassen`;

**CREATE** **TABLE** `klassen` (

`**id**` **INT**(10) **NOT** **NULL** AUTO\_INCREMENT,

`lehrer\_id` **INT**(10) **DEFAULT** **NULL**,

`name` **VARCHAR**(50)

**DEFAULT** **NULL**,

`beschreibung` **VARCHAR**(255)

**DEFAULT** **NULL**,

**PRIMARY** **KEY** (`**id**`)

# **FOREIGN** **KEY** (klassen\_lehrer\_id) **REFERENCES** lehrer(**id**)

)

ENGINE = InnoDB

**DEFAULT** CHARSET = utf8

**COLLATE** = utf8\_unicode\_ci;

**SELECT** "Table klassen has been created";

**COMMIT**;

**START** **TRANSACTION**;

**LOCK** **TABLES** `klassen` **WRITE**;

**ALTER** **TABLE** `klassen`

**DISABLE** KEYS;

**INSERT** **INTO** `klassen` (`**id**`, `lehrer\_id`, `name`, `beschreibung`)

**SELECT**

`idklasse` **AS** `**id**`,

`klassenlehrer` **AS** `lehrer\_id`,

`name` **AS** `name`,

`realname` **AS** `beschreibung`

**FROM** `schoolinfo12802016`.`klasse`;

**ALTER** **TABLE** `klassen`

**ENABLE** KEYS;

**UNLOCK** **TABLES**;

**SELECT** "Data for klassen has been imported";

**COMMIT**;

**START** **TRANSACTION**;

**DROP** **TABLE** **IF** **EXISTS** `lehrbetriebe`;

**CREATE** **TABLE** `lehrbetriebe` (

`**id**` **INT**(10) **NOT** **NULL** AUTO\_INCREMENT,

`name` **VARCHAR**(50) **NOT** **NULL**,

`strasse` **VARCHAR**(50) **NOT** **NULL**,

`haus\_nr` **VARCHAR**(8) **DEFAULT** **NULL**,

`plz` **VARCHAR**(8) **NOT** **NULL**,

`ort` **VARCHAR**(50) **NOT** **NULL**,

`kanton\_code` **VARCHAR**(2) **DEFAULT** **NULL**,

`land\_code` **VARCHAR**(2) **DEFAULT** **NULL**,

**PRIMARY** **KEY** (`**id**`)

)

ENGINE = InnoDB

**DEFAULT** CHARSET = utf8

**COLLATE** = utf8\_unicode\_ci;

**SELECT** "Table lehrbetriebe has been created";

**COMMIT**;

**START** **TRANSACTION**;

**LOCK** **TABLES** `lehrbetriebe` **WRITE**;

**ALTER** **TABLE** `lehrbetriebe`

**DISABLE** KEYS;

**INSERT** **INTO** `lehrbetriebe` (

`**id**`, `name`, `strasse`, `haus\_nr`, `plz`, `ort`, `kanton\_code`, `land\_code`

)

**SELECT**

`id\_Lehrbetrieb` **AS** `**id**`,

`FName` **AS** `name`,

`FStrasse` **AS** `strasse`,

`FHausNr` **AS** `haus\_nr`,

`FPlz` **AS** `plz`,

`FOrt` **AS** `ort`,

`FKanton` **AS** `kanton\_code`,

`FLand` **AS** `land\_code`

**FROM** `schoolinfo12802016`.`lehrbetriebe`;

**ALTER** **TABLE** `lehrbetriebe`

**ENABLE** KEYS;

**UNLOCK** **TABLES**;

**SELECT** "Data for lehrbetriebe has been imported";

**COMMIT**;

**START** **TRANSACTION**;

**DROP** **TABLE** **IF** **EXISTS** `lernende`;

**CREATE** **TABLE** `lernende` (

`**id**` **INT**(10) **NOT** **NULL** AUTO\_INCREMENT,

`anrede` **VARCHAR**(25) **DEFAULT** **NULL**,

`name` **VARCHAR**(50) **DEFAULT** **NULL**,

`vorname` **VARCHAR**(50) **DEFAULT** **NULL**,

`geschlecht` **VARCHAR**(50) **DEFAULT** **NULL**,

`klasse\_id` **INT**(10) **DEFAULT** **NULL**,

`ist\_bm` TINYINT(1) **NOT** **NULL**,

`fachrichtung\_id` **INT**(10) **DEFAULT** **NULL**,

`lehrbetrieb\_id` **INT**(10) **DEFAULT** **NULL**,

`strasse` **VARCHAR**(50) **DEFAULT** **NULL**,

`plz` **VARCHAR**(50) **DEFAULT** **NULL**,

`ort` **VARCHAR**(50) **DEFAULT** **NULL**,

**PRIMARY** **KEY** (`**id**`),

**FOREIGN** **KEY** (lernende\_klasse\_id) **REFERENCES** klassen (**id**),

**FOREIGN** **KEY** (lernende\_fachrichtung\_id) **REFERENCES** fachrichtungen (**id**),

**FOREIGN** **KEY** (lernende\_lehrbetrieb\_id) **REFERENCES** lehrbetriebe (**id**)

)

ENGINE = InnoDB

**DEFAULT** CHARSET = utf8

**COLLATE** = utf8\_unicode\_ci;

**SELECT** "Table lernende has been created";

**COMMIT**;

**START** **TRANSACTION**;

**LOCK** **TABLES** `lernende` **WRITE**;

**ALTER** **TABLE** `lernende`

**DISABLE** KEYS;

**INSERT** **INTO** `lernende` (

`**id**`, `anrede`, `name`, `vorname`,

`geschlecht`, `klasse\_id`, `ist\_bm`,

`fachrichtung\_id`, `lehrbetrieb\_id`,

`strasse`, `plz`, `ort`

)

**SELECT**

`Lern\_id` **AS** `**id**`,

`anrede` **AS** `anrede`,

`name` **AS** `name`,

`vorname` **AS** `vorname`,

`geschlecht` **AS** `geschlecht`,

`klasse` **AS** `klasse\_id`,

`bm` **AS** `ist\_bm`,

`richtung` **AS** `fachrichtung\_id`,

`lehrbetrieb` **AS** `lehrbetrieb\_id`,

`strasse` **AS** `strasse`,

`plz` **AS** `plz`,

`ort` **AS** `ort`

**FROM** `schoolinfo12802016`.`lernende`;

**ALTER** **TABLE** `lernende`

**ENABLE** KEYS;

**UNLOCK** **TABLES**;

**SELECT** "Data for lernende has been imported";

**COMMIT**;

**START** **TRANSACTION**;

**DROP** **TABLE** **IF** **EXISTS** `berechtigungen\_log`;

**CREATE** **TABLE** `berechtigungen\_log` (

`**id**` **INT**(10) **NOT** **NULL** AUTO\_INCREMENT,

`benutzer` **VARCHAR**(50) **NOT** **NULL**,

`zeitpunkt` DATETIME **NOT** **NULL** **DEFAULT** **CURRENT\_TIMESTAMP**,

`grund` **VARCHAR**(50) **NOT** **NULL**,

`berechtigung` **VARCHAR**(50) **NOT** **NULL**,

`tabelle` **VARCHAR**(255) **NOT** **NULL**,

**PRIMARY** **KEY** (`**id**`)

)

ENGINE = InnoDB

**DEFAULT** CHARSET = latin1;

**SELECT** "Table berechtigungen\_log has been created";

**COMMIT**;

**START** **TRANSACTION**;

**LOCK** **TABLES** `berechtigungen\_log` **WRITE**;

**ALTER** **TABLE** `berechtigungen\_log`

**DISABLE** KEYS;

**INSERT** **INTO** `berechtigungen\_log` (

`**id**`, `benutzer`, `zeitpunkt`, `grund`, `berechtigung`, `tabelle`)

**SELECT**

`**id**` **AS** `**id**`,

`benutzer` **AS** `benutzer`,

`**timestamp**` **AS** `zeitpunkt`,

`wofuer` **AS** `grund`,

`berechtigung` **AS** `berechtigung`,

`fuer` **AS** `tabelle`

**FROM** `schoolinfo12802016`.`log\_berechtigung`;

**ALTER** **TABLE** `berechtigungen\_log`

**ENABLE** KEYS;

**UNLOCK** **TABLES**;

**SELECT** "Data for berechtigungen\_log has been imported";

**COMMIT**;

**START** **TRANSACTION**;

**DROP** **TABLE** **IF** **EXISTS** `**module**`;

**CREATE** **TABLE** `**module**` (

`**id**` **INT**(10) **NOT** **NULL** AUTO\_INCREMENT,

`name` **VARCHAR**(50) **DEFAULT** **NULL**,

`beschreibung` **VARCHAR**(255) **DEFAULT** **NULL**,

**PRIMARY** **KEY** (`**id**`)

)

ENGINE = InnoDB

**DEFAULT** CHARSET = utf8

**COLLATE** = utf8\_unicode\_ci;

**SELECT** "Table module has been created";

**COMMIT**;

**START** **TRANSACTION**;

**LOCK** **TABLES** `**module**` **WRITE**;

**ALTER** **TABLE** `**module**`

**DISABLE** KEYS;

**INSERT** **INTO** `**module**` (

`**id**`, `name`, `beschreibung`

)

**SELECT**

`**id**` **AS** `**id**`,

`m\_name` **AS** `name`,

`modulname` **AS** `beschreibung`

**FROM** `schoolinfo12802016`.`modul`;

**ALTER** **TABLE** `**module**`

**ENABLE** KEYS;

**UNLOCK** **TABLES**;

**SELECT** "Data for module has been imported";

**COMMIT**;

**START** **TRANSACTION**;

**DROP** **TABLE** **IF** **EXISTS** `noten`;

**CREATE** **TABLE** `noten` (

`**id**` **INT**(10) **NOT** **NULL** AUTO\_INCREMENT,

`lernende\_id` **INT**(10) **DEFAULT** **NULL**,

`modul\_id` **INT**(10) **DEFAULT** **NULL**,

`erf\_note` **DOUBLE**(15, 2) **DEFAULT** **NULL**,

`erf\_datum` DATETIME **DEFAULT** **NULL**,

`knw\_note` **DOUBLE**(15, 2) **DEFAULT** **NULL**,

`knw\_datum` DATETIME **DEFAULT** **NULL**,

**PRIMARY** **KEY** (`**id**`)

)

ENGINE = InnoDB

**DEFAULT** CHARSET = utf8

**COLLATE** = utf8\_unicode\_ci;

**SELECT** "Table noten has been created";

**COMMIT**;

**START** **TRANSACTION**;

**LOCK** **TABLES** `noten` **WRITE**;

**ALTER** **TABLE** `noten`

**DISABLE** KEYS;

**INSERT** **INTO** `noten` (

`lernende\_id`,

`modul\_id`,

`erf\_note`,

`erf\_datum`,

`knw\_note`,

`knw\_datum`

)

**SELECT**

`lernende\_idlernende` **AS** `lernende\_id`,

`module\_idmodule` **AS** `modul\_id`,

`erf\_note` **AS** `erfahrungsnote`,

`erf\_datum` **AS** `erfahrungsnote\_datum`,

`knw\_note` **AS** `knw\_note`,

`knw\_datum` **AS** `knw\_datum`

**FROM** `schoolinfo12802016`.`noten`;

**ALTER** **TABLE** `noten`

**ENABLE** KEYS;

**UNLOCK** **TABLES**;

**SELECT** "Data noten has been imported";

**COMMIT**;

**START** **TRANSACTION**;

**DROP** **TABLE** **IF** **EXISTS** `fachrichtungen`;

**CREATE** **TABLE** `fachrichtungen` (

`**id**` **INT**(10) **NOT** **NULL** AUTO\_INCREMENT,

`name` **VARCHAR**(50) **DEFAULT** **NULL**,

**PRIMARY** **KEY** (`**id**`)

)

ENGINE = InnoDB

**DEFAULT** CHARSET = utf8

**COLLATE** = utf8\_unicode\_ci;

**SELECT** "Table fachrichtungen has been created";

**COMMIT**;

**START** **TRANSACTION**;

**LOCK** **TABLES** `fachrichtungen` **WRITE**;

**ALTER** **TABLE** `fachrichtungen`

**DISABLE** KEYS;

**INSERT** **INTO** `fachrichtungen` (

`**id**`, `name`

)

**SELECT**

`**id**` **AS** `idrichtung`,

`richtung` **AS** `name`

**FROM** `schoolinfo12802016`.`richtung`;

**ALTER** **TABLE** fachrichtungen

**ENABLE** KEYS;

**UNLOCK** **TABLES**;

**SELECT** "Data fachrichtungen has been imported";

**COMMIT**;

**END** //

DELIMITER ;

1. Teilprojekt Ortschaftstbelle erstellen
   1. Stored Procedure sp\_orte\_extrahiern

* Name
  + sp\_orte\_extrahieren
* Eingabeparameter
  + keine
* Ausgabeparameter
  + keine
* Resultat
  + Tabelle schoolinfo\_neu.orte ist erstellt
  + Daten sind aus folgenden Tabellen zusammengeführt
    - schoolinfo\_neu.lernende
    - schoolinfo\_neu.lehrbetriebe
  + Folgende Tabellen sind verändert
    - schoolinfo\_neu.lernende
    - schoolinfo\_neu.lehrbetriebe
    1. Datenkatalog
    2. Tabelle Orte

|  |  |  |  |
| --- | --- | --- | --- |
| ***Tabelle orte*** | | | |
| **id** | INT(10) | NOT NULL | AUTO INCREMENT |
| **plz** | VARCHAR(8) | NOT NULL |  |
| **ort** | VARCHAR(50) |  |  |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***SCHLÜSSEL*** | ***Name*** | ***Referenztabelle*** | ***Feld*** | ***Referenz Feld*** |
| **PKY** | id |  |  | id |
|  |  |  |  |  |

* + 1. Tabelle Lehrbetriebe

|  |  |  |  |
| --- | --- | --- | --- |
| ***Tabelle lehrbetriebe*** | | | |
| **id** | INT(10) | NOT NULL | AUTO INCREMENT |
| **name** | VARCHAR(50) |  | DEFAULT NULL |
| **strasse** | VARCHAR(50) | NOT NULL |  |
| **haus\_nr** | VARCHAR(50) |  | DEFAULT NULL |
| **ort\_id** | INT(10) | NOT NULL |  |
| **kanton\_code** | VARCHAR(2) |  | DEFAULT NULL |
| **land\_code** | VARCHAR(2) |  | DEFAULT NULL |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***SCHLÜSSEL*** | ***Name*** | ***Referenztabelle*** | ***Feld*** | ***Referenz Feld*** |
| **PKY** | id |  |  | id |
| **FKY** | lehrbetriebe\_ort\_id | orte | id | ort\_id |
|  |  |  |  |  |

* + 1. Tabelle Lernende

|  |  |  |  |
| --- | --- | --- | --- |
| ***Tabelle lernende*** |  |  |  |
| **id** | INT(10) | NOT NULL | AUTO INCREMENT |
| **anrede** | VARCHAR(50) |  | DEFAULT NULL |
| **name** | VARCHAR(50) | NOT NULL |  |
| **vorname** | VARCHAR(50) | NOT NULL |  |
| **geschlecht** | VARCHAR(2) | NOT NULL | “M“ = Männlich,  “F“ = Weiblich |
| **klasse\_id** | INT(10) | NOT NULL |  |
| **ist\_bm** | TINYINT(1) | NOT NULL | 0 = Nein  1 = Ja |
| **fachrichtung\_id** | INT(10) | NOT NULL |  |
| **lehrbetrieb\_id** | INT(10) | NOT NULL |  |
| **strasse** | VARCHAR(50) | NOT NULL |  |
| **ort\_id** | INT(10) | NOT NULL |  |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***SCHLÜSSEL*** | ***Name*** | ***Referenztabelle*** | ***Feld*** | ***Referenz Feld*** |
| **PKY** | id |  |  |  |
| **FKY** | lernende\_fachrichtung\_id | fachrichtungen | id | fachrichtung\_id |
| **FKY** | lernende\_klasse\_id | klassen | id | klasse\_id |
| **FKY** | lernende\_lehrbetrieb\_id | lehrbetriebe | id | lehrbetrieb\_id |
| **FKY** | lernende\_ort\_id | orte | id | ort\_id |
|  |  |  |  |  |

* 1. Stored Procedure Quelltext

***DELIMITER //***

**CREATE** **PROCEDURE** sp\_orte\_extrahieren

**BEGIN**

**START** **TRANSACTION**;

**DROP** **TABLE** **IF** **EXISTS** `orte`;

**CREATE** **TABLE** `orte` (

`id` INT(10) NOT NULL AUTO\_INCREMENT,

`plz` VARCHAR(6) NOT NULL,

`ort` VARCHAR(50) NOT NULL,

`region\_code` VARCHAR(2) DEFAULT NULL,

'land\_code' VARCHAR(2) DEFAULT NULL,

**PRIMARY** **KEY** (`id`)

)

ENGINE = InnoDB

DEFAULT CHARSET = utf8

COLLATE = utf8\_unicode\_ci;

**SELECT** "Table ort has been created";

**COMMIT**;

**START** **TRANSACTION**;

**LOCK** **TABLES** `orte` **WRITE**;

**ALTER** **TABLE** `orte` DISABLE **KEYS**;

**INSERT** **INTO** `orte` (`plz`, `ort`)

**SELECT** **DISTINCT**

plz, ort

**FROM** (

**SELECT**

TRIM(**TRAILING** "\r" **FROM** `plz`

) **as** plz,

TRIM(**TRAILING** " 1" **FROM**

TRIM(**TRAILING** " AG" **FROM**

TRIM(**TRAILING** " SO" **FROM**

TRIM(**TRAILING** " BL" **FROM**

TRIM(**TRAILING** "\r" **FROM** `ort`)

)

)

)

) **as** ort

**FROM** `lernende` **GROUP** **BY** `plz`, `ort`

**UNION**

**SELECT**

TRIM(**TRAILING** "\r" **FROM** `plz`

) **as** plz,

TRIM(**TRAILING** " 1" **FROM**

TRIM(**TRAILING** " AG" **FROM**

TRIM(**TRAILING** " SO" **FROM**

TRIM(**TRAILING** " BL" **FROM**

TRIM(**TRAILING** "\r" **FROM** `ort`)

)

)

)

) **as** ort

**FROM** `lehrbetriebe` **GROUP** **BY** `plz`, `ort`) **as** tmp;

**ALTER** **TABLE** `orte` ENABLE **KEYS**;

**UNLOCK** **TABLES**;

**SELECT** "Data for orte has been imported";

**COMMIT**;

**START** **TRANSACTION**;

**LOCK** **TABLES** `lernende` **WRITE**;

**ALTER** **TABLE** `lernende`

**ADD** **COLUMN** `ort\_id` INT(10) NULL DEFAULT NULL **AFTER** `strasse`;

**SELECT** "ort collumn has been created for lernende table";

**UPDATE** `lernende`

**SET** `lernende`.`ort\_id` = `orte`.`id`

**WHERE**

`lernende`.`plz` **LIKE** CONCAT('%', `orte`.`plz` ,'%') AND

`lernende`.`ort` **LIKE** CONCAT('%', `orte`.`ort` ,'%');

**SELECT** "ort\_id has been set in lernende table";

**ALTER** **TABLE** `lernende`

**DROP** **COLUMN** `plz`,

**DROP** **COLUMN** `ort`,

**ADD** **FOREIGN** **KEY** (`lernende\_ort\_id`)

**REFERENCES** `orte`(`id`);

**SELECT** "lernende\_ort\_id foreign key has been established";

**UNLOCK** **TABLES**;

**COMMIT**;

**START** **TRANSACTION**;

**LOCK** **TABLES** `lehrbetriebe` **WRITE**;

**ALTER** **TABLE** `lehrbetriebe`

**ADD** **COLUMN** `ort\_id` INT(10) NULL DEFAULT NULL **AFTER** `haus\_nr`;

**SELECT** "ort collumn has been created for lehrbetriebe table";

**UPDATE** `lehrbetriebe`

**SET** `lehrbetriebe`.`ort\_id` = `orte`.`id`

**WHERE**

`lehrbetriebe`.`plz` **LIKE** CONCAT('%', `orte`.`plz` ,'%') AND

`lehrbetriebe`.`ort` **LIKE** CONCAT('%', `orte`.`ort` ,'%');

**SELECT** "ort\_id has been set in lehrbetriebe table";

**ALTER** **TABLE** `lehrbetriebe`

**DROP** **COLUMN** `plz`,

**DROP** **COLUMN** `ort`,

**ADD** **FOREIGN** **KEY** (`lernende\_ort\_id`)

**REFERENCES** `orte`(`id`);

**SELECT** "lehrbetriebe\_ort\_id foreign key has been established";

**UNLOCK** **TABLES**;

**COMMIT**;

**END** ***//***

DELIMITER ;

1. Teilprojekt Benutzer erstellen
   1. Stored Procedure sp\_benutzer\_erstellen

* Name
  + sp\_benutzer\_erstellen
* Eingabeparameter
  + IN this\_user VARCHAR(50)
  + IN this\_host VARCHAR(50)
  + IN this\_pass VARCHAR(50)
  + IN this\_table VARCHAR(50)
  + IN this\_permission VARCHAR(50)
* Ausgabeparameter
  + keine
* Resultat
  + Benutzer ist erstellt.
  + Benutzer hat die Berechtigung für die entsprechende Tabelle gesetzt
  + Logtabelle erstellt sofern diese nicht bereits existiert.
  + Logeintrag in Tabelle schoolinfo\_neu.berechtigungen\_log erstellt
  + Bei Fehleingaben wird eine Exception geworfen
  1. Aufrufbeispiele

Benutzer erstellen und Leseberechtigung für Tabelle schoolinfo\_neu.lernende erteilen

sp\_bentuzer\_erstellen("jerome","localhost","1234","lernende","SELECT");

* 1. Eingabe Validation

Die Eingabeparameter werden validiert. Die maximale Zeichenlänge von 50 Zeichen pro Parameter wird berücksichtigt. Ausserdem wird nach potentiell schädlichen Zeichen geprüft. Diese Prüfung erfolgt mittels REGEX

Siehe Persönliches Lernjournal

[http://wiki.steampilot.ch](http://wiki.steampilot.ch/) → Suche → Regex

* 1. Datenkatalog
     1. Tabelle Berechtigungen\_log

|  |  |  |  |
| --- | --- | --- | --- |
| ***Tabelle berechtigungen\_log*** | | | |
| **id** | INT(10) | NOT NULL | AUTO INCREMENT |
| **benutzer** | VARCHAR(50) | NOT NULL |  |
| **zeitpunkt** | DATETIME | NOT NULL | DEFAULT CURRENT\_TIMESTAMP |
| **grund** | VARCHAR(50) | NOT NULL |  |
| **berechtigung** | VARCHAR(50) | NOT NULL |  |
| **tabelle** | VARCHAR(50) | NOT NULL |  |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***SCHLÜSSEL*** | ***Name*** | ***Referenztabelle*** | ***Feld*** | ***Referenz Feld*** |
| **PKY** | id |  |  | id |
|  |  |  |  |  |

* 1. Stored Procedure Quelltext

***DELIMITER //***

**CREATE** **PROCEDURE** sp\_benutzer\_erstellen(

**IN** this\_user VARCHAR(50),

**IN** this\_host VARCHAR(50),

**IN** this\_pass VARCHAR(50),

**IN** this\_table VARCHAR(50),

**IN** this\_permission VARCHAR(50)

)

**BEGIN**

**START** **TRANSACTION**;

**IF** (LENGTH(

this\_user OR

this\_host OR

this\_pass OR

this\_table OR

this\_permission

) > 50)

**THEN**

SIGNAL SQLSTATE '45000'

**SET** MESSAGE\_TEXT = 'Eingabe zu lange';

ELSEIF (

this\_user OR

this\_host OR

this\_pass OR

this\_table OR

this\_permission

REGEXP '|^((?![;$<>{}\[\]]).)\*$|')

**THEN**

SIGNAL SQLSTATE '45000'

**SET** MESSAGE\_TEXT = 'Eingabe enthaelt unerlaubte Zeichen. Nur a-z A-Z 0-9 sind erlaubt';

**END** **IF**;

DECLARE query1 VARCHAR(255);

**SET** query1 = CONCAT('

CREATE USER IF NOT EXISTS "', this\_user, '"@"', this\_host,

'" IDENTIFIED BY "', this\_pass, '" ');

PREPARE statement1 **FROM** query1;

EXECUTE statement1;

DEALLOCATE PREPARE statement1;

**SELECT** CONCAT(' User ', this\_user, ' at ', this\_host, ' has been created.');

**INSERT** **INTO** `schoolinfo\_neu`.`berechtigungen\_log`

(`benutzer`, `grund`, `berechtigung`,`tabelle`)

**VALUES** (CONCAT(this\_user, "@", this\_host), "CREATE", "NONE", "NONE");

**COMMIT**;

**START** **TRANSACTION**;

DECLARE query2 VARCHAR(255);

**SET** query2 = CONCAT(

'GRANT ', this\_permission, ' ON `schoolinfo.neu`.', this\_table,

' TO "', this\_user, '"@"', this\_host, '" IDENTIFIED BY "', this\_pass, '" ');

PREPARE statement2 **FROM** query2;

EXECUTE statement2;

DEALLOCATE PREPARE statement2;

**SELECT** CONCAT(' User ', this\_user, ' at ', this\_host,

' has been granted ',this\_permission,

' permisson on table ',this\_table,'.');

**INSERT** **INTO** `schoolinfo\_neu`.`berechtigungen\_log`

(`benutzer`, `grund`, `berechtigung`, `tabelle`)

**VALUES** (CONCAT(this\_user, "@", this\_host), "Granted permission", this\_permission, this\_table);

**COMMIT**;

**END** ***//***

DELIMITER ;

1. Teilprojekt Lernende archivieren
   1. Stored Procedure sp\_lernende\_archivieren

* Name
  + sp\_lernende\_archivieren
* Eingabeparameter
  + IN this\_klasse\_id INT(10)
* Ausgabeparameter
  + keine
* Resultat
  + Datenbank schoolinfo\_archiev ist erstellt sofern diese nicht bereits existiert
  + Tabelle schoolinfo\_archiv.lernende\_archiv ist erstellt sofern diese nicht bereits existiert
  + Archieveinträge aus den zusammengezogenen Daten aus Datenbank schoolinfo\_neu ist in Tabelle lernende\_archiv geschrieben
  1. Datenkatalog
     1. Tabelle schoolinfo\_archiv.lernende\_archiv

|  |  |  |  |
| --- | --- | --- | --- |
| ***Tabelle lernende\_archiv*** | | | |
| **id** | INT(10) | NOT NULL | AUTO INCREMENT |
| **nachname** | VARCHAR(50) | NOT NULL |  |
| **vorname** | VARCHAR(80) | NOT NULL |  |
| **bm** | BOOLEAN | NOT NULL |  |
| **strasse** | VARCHAR(50) | NOT NULL |  |
| **plz** | VARCHAR(8) | NOT NULL |  |
| **ort** | VARCHAR(50) | NOT NULL |  |
| **richtung** | VARCHAR(50) | NOT NULL |  |
| **klasse** | VARCHAR(50) | NOT NULL |  |
| **lehrbetrieb** | VARCHAR(50) | NOT NULL |  |
| **lbstrasse** | VARCHAR(50) | NOT NULL |  |
| **lbhausnr** | VARCHAR(8) | NOT NULL |  |
| **lbplz** | VARCHAR(8) | NOT NULL |  |
| **lbort** | VARCHAR(50) | NOT NULL |  |
| **lbland** | VARCHAR(2) | NOT NULL |  |
| **modulname** | VARCHAR(50) | NOT NULL |  |
| **note\_erf** | DOUBLE(15,2) | NOT NULL |  |
| **note\_knw** | DOUBLE(15,2) | NOT NULL |  |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***SCHLÜSSEL*** | ***Name*** | ***Referenztabelle*** | ***Feld*** | ***Referenz Feld*** |
| **PKY** | id |  |  | id |
|  |  |  |  |  |

* 1. Stored Procedure Quelltext

***DELIMITER //***

**CREATE** **PROCEDURE** sp\_lernende\_archivieren(

**IN** this\_klasse\_id INT(10)

)

**BEGIN**

**CREATE** DATABASE **IF** NOT **EXISTS** schoolinfo\_archiv;

**USE** schoolinfo\_archiv;

**CREATE** **TABLE** **IF** NOT **EXISTS** lernende\_archiv (

id INT **PRIMARY** **KEY** AUTO\_INCREMENT,

nachname VARCHAR(50) NOT NULL,

vorname VARCHAR(50) NOT NULL,

bm TINYINT(1) NOT NULL,

strasse VARCHAR(50) NOT NULL,

plz VARCHAR(8) NOT NULL,

ort VARCHAR(50) NOT NULL,

richtung VARCHAR(30) NOT NULL,

klasse VARCHAR(10) NOT NULL,

lehrbetrieb VARCHAR(100) NOT NULL,

lbstrasse VARCHAR(50) NOT NULL,

lbhausnr VARCHAR(8) NOT NULL,

lbplz VARCHAR(8) NOT NULL,

lbort VARCHAR(50) NOT NULL,

lbland VARCHAR(2) NOT NULL,

modulname VARCHAR(50) NOT NULL,

note\_erf DOUBLE(15, 2) DEFAULT NULL,

note\_knw DOUBLE(15, 2) DEFAULT NULL,

timestamp DATETIME NOT NULL DEFAULT current\_timestamp

)

ENGINE = ARCHIVE;

**INSERT** **INTO** schoolinfo\_archiv.lernende\_archiv

**SELECT**

LD.`name` **AS** nachname,

LD.`vorname` **AS** vorname,

LD.`ist\_bm` **AS** bm,

LD.`strasse` **AS** strasse,

LT.`plz` **AS** plz,

LT.`ort` **AS** ort,

LD.`fachrichtung` **AS** richtung,

KL.`name` **AS** klasse,

BB.`name` **AS** lehrbetrieb,

BB.`strasse` **AS** lbstrasse,

BB.`haus\_nr` **AS** lbhausnr,

BT.`plz` **AS** lbplz,

BT.`ort` **AS** lbort,

BB.`land\_code` **AS** lbland,

MD.`name` **AS** modulname,

NT.`erf\_note` **AS** note\_erf,

NT.`knw\_note` **AS** note\_knw

**FROM** schoolinfo\_neu.lernende **AS** LD

LEFT **JOIN** schoolinfo\_neu.orte **AS** LT

**ON** LD.ort\_id = LT.id

LEFT **JOIN** schoolinfo\_neu.klasse **AS** KL

**ON** LD.klasse\_id = KL.id

LEFT **JOIN** schoolinfo\_neu.lehrbetriebe **AS** BB

**ON** LD.lehrbetrieb\_id = BB.id

LEFT **JOIN** schoolinfo\_neu.orte **AS** BT

**ON** BB.ort\_id = BT.id

RIGHT **OUTER** **JOIN** schoolinfo\_neu.note **AS** NT

**ON** NT.lernende\_id = LD.id

LEFT **JOIN** schoolinfo\_neu.module **AS** MD

**ON** NT.modul\_id = MD.id

**WHERE** LD.klasse\_id = this\_klasse\_id;

**DELETE** t1 **FROM** schoolinfo\_neu.noten t1

**INNER** **JOIN** schoolinfo\_neu.lernende t2 **ON** (t1.lernende\_id = t2.id)

**INNER** **JOIN** schoolinfo\_neu.klasse t3 **ON** (t2.klasse\_id = t3.id)

**WHERE** t3.id = this\_klasse\_id;

**DELETE** **FROM** schoolinfo\_neu.lernende **AS** LD

**WHERE** LD.klasse\_id = this\_klasse\_id;

**END** ***//***

DELIMITER ;