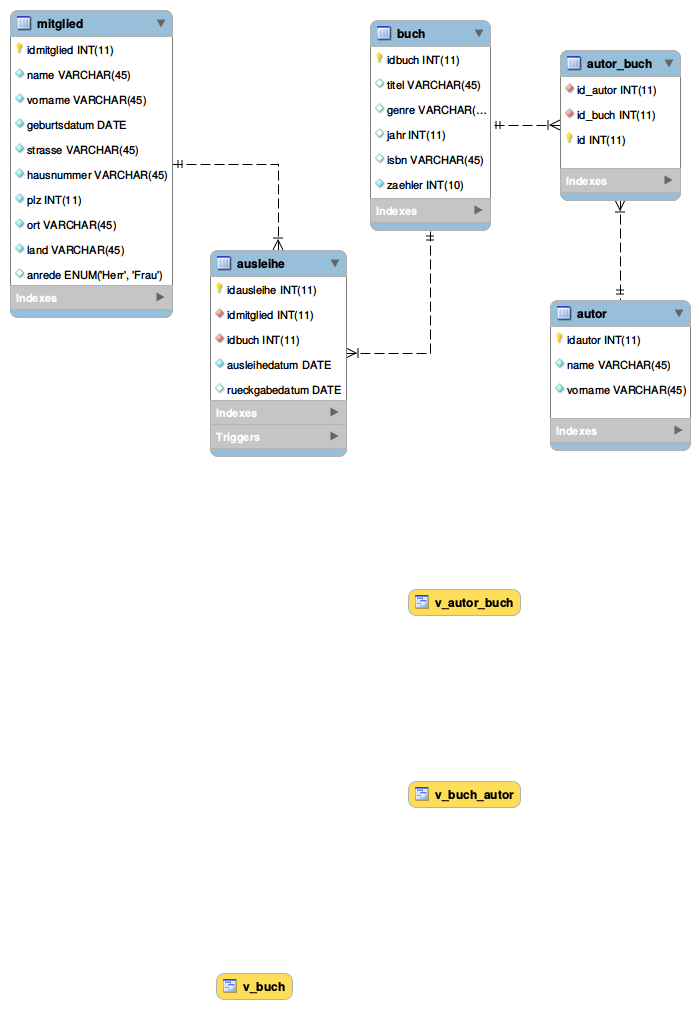
# 61 Prüfungsvorbeitung II Musterlösung

|  |  |
| --- | --- |
| **Erwünschtes Ergebnis** | Am Beispiel einer Bibliotheks-Datenbank repetieren Sie, die Regeln der referenziellen Integrität zu testen und umzusetzen. |
| **Zeitaufwand** | 60 Minuten |
| **Ausgangslage** | Für ein Bibliothekssystem wurde das Skript **m104\_bibo.sql** erstellt.  Mit dieser Datenbank sollen alle Bücher, Ausleihvorgänge und Biblitheksmitglieder verwaltet werden.  Als Anforderung wurden folgende Punkte festgehalten:   * Für jedes Mitglied muss der vollständige Name, die Adresse und das Geburtsdatum gespeichert werden. * Für jedes Buch müssen mindestens Titel, Genre, Erscheinungsjahr gespeichert werden. * Für jeden Ausleihvorgang müssen Buch, Mitglied und Ausleihdatum gespeichert werden. Das Rückgabedatum soll erst bei der Rückgabe eingegeben werden.   **Hinweis**: Durch Deaktivierung der Überwachung der referentiellen Integrität oder Fehler in der Implementierung des ERD ist es auch möglich, ungültige Daten einzugeben. Die Überwachung sollte deshalb nicht deaktiviert werden. |
| **Aufgabe** | 1. Schauen Sie sich das Skript im Editor kurz an (ideal Notepad++) 2. Importieren Sie die Datenbank in Ihr DBMS. 3. Analysieren Sie den Aufbau der Datenbank und erstellen Sie das zugehörige ERD. 4. Erstellen Sie einen Testplan mit ca. 3 Abfragen, der obige Anforderungen und die referentielle Integrität der Daten prüft. 5. Testen Sie die Datenbank anhand Ihres Testplans. 6. Verbessern Sie das Datenbank-Skript so, dass obige Anforderungen erfüllt werden und die referentielle Integrität sichergestellt ist. |
| **Hinweis** | Lösen Sie die Aufgabe allein oder in 2er-Gruppen. Bei Unklarheiten fragen Sie Ihren Dozenten. |
| **Ergebnis** | **Selbstkontrolle:**   * Ich habe das ERD gezeichnet und den Testplan erstellt. * Ich habe Stellen gefunden, wo die Anforderungen nicht erfüllt bzw. die referenzielle Integrität verletzt wurde. * Ich habe das Skript zur Erstellung der Datenbank verbessert und erneut getestet. |

**Lösungsvorschlag:**

* 1. ERD
  2. Testplan
  3. Fehler gefunden und behoben:
* Foreign Key constraints fehlen
* Attribute der Tabelle *Buch* und *Mitglied* können noch auf NOT NULL geändert werden. Damit wird einfach sichergestellt, dass Werte eingegeben werden, d.h. dass das Mitglied z.B. einen Namen bekommt bzw. das Feld nicht leer sein darf.
  1. Fertiges Skript siehe m104\_bibo\_loesung.sql

******

|  |  |  |  |
| --- | --- | --- | --- |
| **Schritt Nr.**  **1** | **Titel des Tests**  **Positiv-Test: Eingabe valider Mitglieder-Daten** | | |
|  | **Beschreibung der Durchführung**  **INSERT INTO `mitglied` ( `name`, `vorname`, `geburtsdatum`, `strasse`, `hausnummer`, `plz`, `ort`, `land`, `anrede`) VALUES**  **('Baumann', 'Daniel', '1955-05-23', 'Eggerstander Strasse', '15', 9050, 'Appenzell', 'Schweiz', NULL);**  **SELECT \* FROM mitglied;** | | |
| **Erwartetes Ergebnis**  **Daniel Baumann wird angezeigt** | **Effektives Ergebnis**  **Daniel Baumann wird angezeigt** | **Erfüllt**  **ok** |
| **Bewertung und Massnahmen** | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Schritt Nr.**  **2** | **Titel des Tests**  **Negativ-Test: Eingabe ungültiger Mitglieder-Daten** | | |
|  | **Beschreibung der Durchführung**  **INSERT INTO `mitglied` ( `name`, `vorname) VALUES**  **('Baumann', 'Daniel');**  **SELECT \* FROM mitglied;** | | |
| **Erwartetes Ergebnis**  **Fehlermeldung beim INSERT** | **Effektives Ergebnis**  **Daniel Baumann wird angezeigt** | **Erfüllt**  **NOK** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Schritt Nr.**  **3** | **Titel des Tests**  **Negativ-Test: Eingabe ungültiger Fremdschlüssel für Ausleihe** | | |
|  | **Beschreibung der Durchführung**  **INSERT INTO `ausleihe` ( `idmitglied`, `idbuch`) VALUES**  **(30, 7);**  **SELECT \* FROM ausleihe;** | | |
| **Erwartetes Ergebnis**  **Fehlermeldung beim INSERT** | **Effektives Ergebnis**  **Ausleih-Eintrag wird angezeigt** | **Erfüllt**  **NOK** |
| **Bewertung und Massnahmen**  **FK implementieren** | | |