

Praktikum – Einführung in Datenbanksysteme (DBS I)
SS 14

Aufgabenblatt 5:
SQL & JDBC

Abgabetermin: 13.07.2014

Für die Bewertung der Aufgaben ist es erforderlich, dass Sie sich im Goya-System für die Veranstaltung eingetragen haben und einer Gruppe zugeordnet sind. Geben Sie Ihre Lösung zur Aufgabe 1.1 als Textdatei, Word- oder PDF-Dokument und zu den Aufgaben 1.2 und 2 in den angegebenen Dateien mittels Goya ab.

Die maximale Punkteanzahl beträgt 54 Punkte.

Im Rahmen dieser Aufgabe soll der Zugriff auf eine einfache Musikdatenbank programmiert werden. Die Daten wurden mit Hilfe des Amazon Webservices geladen und liegen als XML-Dateien vor. Die Aufgabe dieses Praktikums ist, das gegebene ER-Modell in ein Relationenmodell zu transformieren und alle notwendigen SQL-DDL-Anweisungen zu generieren (inkl. Constraints und Trigger). Dazu soll das gegebene Java-Projekt MusicDB für den JDBC-Zugriff auf die Datenbank vervollständigt werden. Das Programm soll die Möglichkeit bieten, neue CDs aus den XML-Dateien zu importieren und bereits geladene CDs anzuzeigen, zu verändern oder zu löschen. Die Methoden zum Zugriff auf die XML-Daten sind bereits vorgegeben.

Alle benötigten Dateien sind in einem [gezippten Tar-Archiv](#) gegeben.

Aufgabe 1: Modellieren und Erstellen der Datenbank (16 Punkte)

Hinweis: Jedes Item stellt eine kaufbare CD dar, die eine eindeutige Nummer (ASIN) besitzt, von einem Künstler (Artist) oder Autor ist, auf einem Label produziert wurde, eine Diskanzahl, einen Titel, einen Neu-Preis und einen Gebracht-Preis hat. Jede CD besteht aus mindestens einer Disk, die eine Nummer hat. Jede Disk wiederum hat mindestens einen Track mit Nummer und Titelnamen. Im Modell muss nicht zwischen Künstler und Autor unterschieden werden.

1. Überführen Sie das gegebene ER-Modell in ein Relationenmodell. **(4 Punkte)**
2. Erstellen Sie die von Ihnen definierten Relationen. Fügen Sie dazu im Verzeichnis *MusicDB/sql* in die Datei *createMusicRelation.sql* alle Create-Table-Anweisungen ein. Ergänzen Sie Schlüssel-Constraints und Integritätsbedingungen, die Folgendes sicherstellen.
 - Alle Disk- und Tracknummern sind immer positiv.

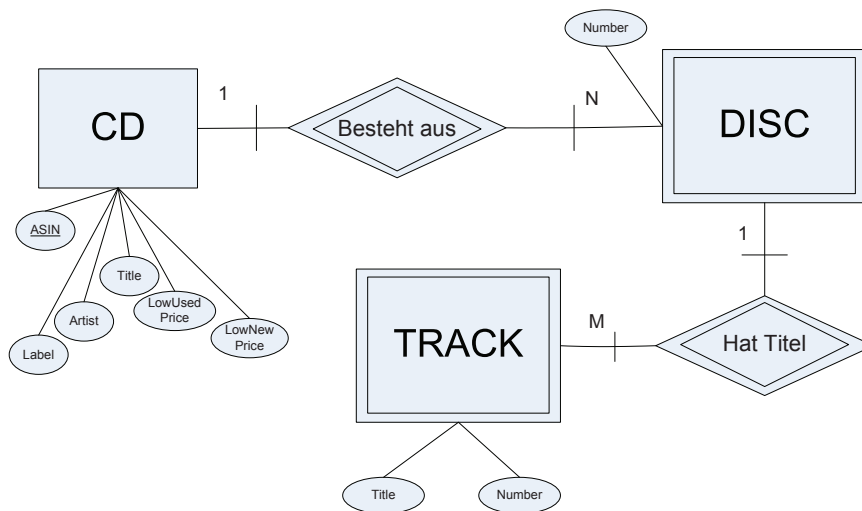


Abbildung 1: ERM

- Alle Preise sind immer nicht-negativ.
- Bei jeder Veränderung des Neu-Preises soll auch der Gebraucht-Preis mit dem gleichen Faktor aktualisiert werden.

Löschen Sie alle von Ihnen erstellten Objekte mittels SQL-Anweisungen in der Datei *dropMusicRelation.sql*. Geben Sie Ihre Lösung in den beiden Dateien *createMusicRelation.sql* und *dropMusicRelation.sql* ab. **(12 Punkte)**

Aufgabe 2: JDBC **(38 Punkte)**

Analysieren Sie das gegebene Programmskelett für einen dynamischen Zugriff auf die Datenbank. An den gekennzeichneten Stellen im Quelltext soll von Ihnen die geforderte Funktionalität der Applikation implementiert und getestet werden. Beachten Sie bei Ihrer Lösung folgende Hinweise:

- Update-Operationen sind nur bezüglich des Neu-Preises möglich (Methode *MusicDB::updateNewPrice*).
- Alle CD-Daten, die die gegebene DTD *musicDBLight.dtd* verletzen (z. B. durch das Fehlen bestimmter Elemente), sollen nicht in die Datenbank eingetragen werden (*ParseMusicXML::loadXMLMusicContentIntoDB*). Die Dateien *response_x.xml* im Verzeichnis *MusicDB/data* folgen einer komplexeren DTD und haben neben den Elementen der gegebenen DTD zusätzliche Elemente. Hier sollen nur die Daten, die in der gegebenen DTD definiert werden (d.h. nur die Attribute, die im ER-Modell aufgeführt werden), in die Datenbank aufgenommen werden.
- Es kann davon ausgegangen werden, dass die Diskanzahl (*NumberOfDiscs*) jeder CD korrekt die vorhandenen Disks angibt.

- Sie müssen in der Datenbank nicht zwischen Künstler (Artist) und Autor (Author) unterscheiden.

Folgende Methoden sind von Ihnen zu ergänzen:

- *MusicDB::createDBConnection* (4 Punkte)
- *MusicDB::closeDBConnection* (2 Punkte)
- *MusicDB::printResult* (4 Punkte)
- *MusicDB::showAllCDs* (3 Punkte)
- *MusicDB::showSingleCD* (6 Punkte)
- *MusicDB::showGroupAllPrices* (3 Punkte)
- *MusicDB::showGroupPricesArtist* (3 Punkte)
- *MusicDB::updateNewPrice* (2 Punkte)
- *MusicDB::deleteSingleCD* (3 Punkte)
- *ParseMusicXML::loadXMLMusicContentIntoDB* (8 Punkte)

Geben Sie Ihre Lösung in den beiden Dateien *MusicDB.java* und *ParseMusicXML.java* ab.

WICHTIG: Es werden nur solche Lösungen bewertet, die auf den Vogelrechnern der Studentenpools (rabe, star, ...) compilierbar sind.