Datenbanksysteme
Sommersemester 2017 - SQL-Übungen

Inhaltsverzeichnis

1 Generelle Anmerkungen

Dieses Übungsblatt ist für den Erwerb eines Klausurbonus für die Klausur zur Vorlesung Datenbanksysteme relevant und wird mit maximal 80 Punkten bewertet. Um den Klausurbonus zu erhalten, müssen Sie 60 Punkte erreichen. Der Klausurbonus verbessert die Note in der Klausur um eine Stufe, also z.B. von 2.7 auf 2.3 oder von 1.3 auf 1.0, etc. Sollte das Klausurergebnis nicht bestanden, also 4.3 oder schlechter sein,findet der Klausurbonus keine Anwendung.
Der Klausurbonus kann nur genutzt werden für die Haupt- und die Nachklausur der Vorlesung, in der der Klausurbonus erworben wurde, d.h. für die Hauptklausur am 04. 08. 2017 oder für die Nachklausur im Februar/März 2018.

Abgabe

Die Abgabe erfolgt elektronisch über das Portal:

https://sql-uebung.ipd.kit.edu/

Dieses Portal wird bis zu folgendem Abgabetermin zur Verfügung stehen:

Letztes Datum der Abgabe: 25. Juli 2017

Wir erwarten, dass die Server-Last unmittelbar vor dem Abgabetermin hoch sein wird. Es liegt in **Ihrer Verantwortlichkeit**, dass die Lösungen **rechtzeitig** vorliegen.

SQL-Konstrukte

Alle Aufgaben lassen sich mit SQL-Konstrukten lösen, die in der Vorlesung vorgestellt wurden, ergänzt um einige zusätzliche Konstrukte wie z.B. ROWNUM, ROUND, CASE. Die Lösungen müssen dem SQL-Standard folgen und unter ORACLE 11g ausführbar sein. Ansonsten bekommen Sie für die Aufgabe keine Punkte.

Offline-Datenbank zum Testen

Um eine sehr große Belastung auf dem Portal und dem Datenbankserver zu vermeiden, stellen wir die Datenbank (dbsysteme) als Download zur Verfügung (Link siehe Ankündigungen auf dem Portal). So können Sie lokal (offline) mit SQLite[1] frei Anfragen an die Datenbank stellen. Um einen besseren Überblick über die Daten zu bekommen, empfehlen wir Ihnen, das graphische Interface[2] für SQLite zu benutzen. Das Schema ist auf den Vorlesungsseiten mit den Übungsmaterialien zu finden. Es ist zu empfehlen, die Anfragen zunächst lokal zu testen und danach das Portal vor allem für die Einreichung der Lösung zu benutzen. Bitte beachten Sie, dass SQLite teilweise vom für die Lösung der Aufgaben geforderten SQL-Standard von ORACLE 11g abweicht. Daher sollten Sie

[1] http://sqlite.org/about.html
[2] http://sqlitebrowser.org/

die Tipps auf dem Online-Portal unbedingt lesen. Ein vorsätzliches Überlasten des Portals wird mit Punktabzug bestraft.

Betrugsversuche

Nicht eigenständig erstellte Lösungen werden mit null Punkten bewertet und führen im Allgemeinen zum Verlust des Prüfungsanspruchs.

2 Datenbankbeschreibung

2.1 ER-Modell

Abbildung 1 zeigt das ER-Modell zu dem Szenario Filme und Ausleihe von Filmen. Dieses Szenario liegt der für die SQL-Übung verwendeten relationalen Datenbank zu Grunde. Folgende Eigenschaften werden in der Modellierung berücksichtigt:

- In einem Film spielen ein oder mehrere Schauspieler mit. Jedem Film wird eine Kategorie zugeordnet. Für einen Film ist die Originalsprache bekannt, in der der Film gedreht wurde. Es kann auch weitere Versionen des Films geben, die in einer anderen Sprache vorliegen. Zu einem Film gibt es ausleihbare Kopien, die im Bestand (Inventory) eines Geschäfts ()Stores) verfügbar sind.
- Ein Kunde hat eine eindeutige Adresse. Darüber hinaus hat ein Kunde ein Geschäft, in dem der Kunde hauptsächlich Filme ausleiht, der sog. Main Store. Jeder Kunde kann ein oder mehrere Filme ausleihen (Rental) und muss für die Ausleihe bezahlen (Payment).
- Einer Ausleihe (rental) ist ein Kunde sowie eine Filmkopie (aus dem Inventory) zugeordnet. Ein Film kann so oft ausgeliehen werden, wie es Kopien in den Beständen der Geschäfte gibt (inventory). Weiterhin wird dem Ausleihvorgang ein Mitarbeiter (Verkäufer) zugeordnet sowie die Bezahlung der Ausleihe.
- Ein Mitarbeiter hat eine Adresse. Er gehört eindeutig zu einem Store, kann aber auch zeitweise in anderen Stores arbeiten.
- Eine Zahlung für eine Ausleihe wird von einem Kunden getätigt und hat Auswirkungen auf den Bestand des Films in einem Inventory.
- Ein Store hat einen Bestand aus auszuleihenden Filmkopien (Inventory), sowie eine eindeutige Adresse.

Abbildung 1 enthält das entsprechende ER-Modell ohne Angabe von Attributen. Es werden Standardkardinalitäten angegeben.

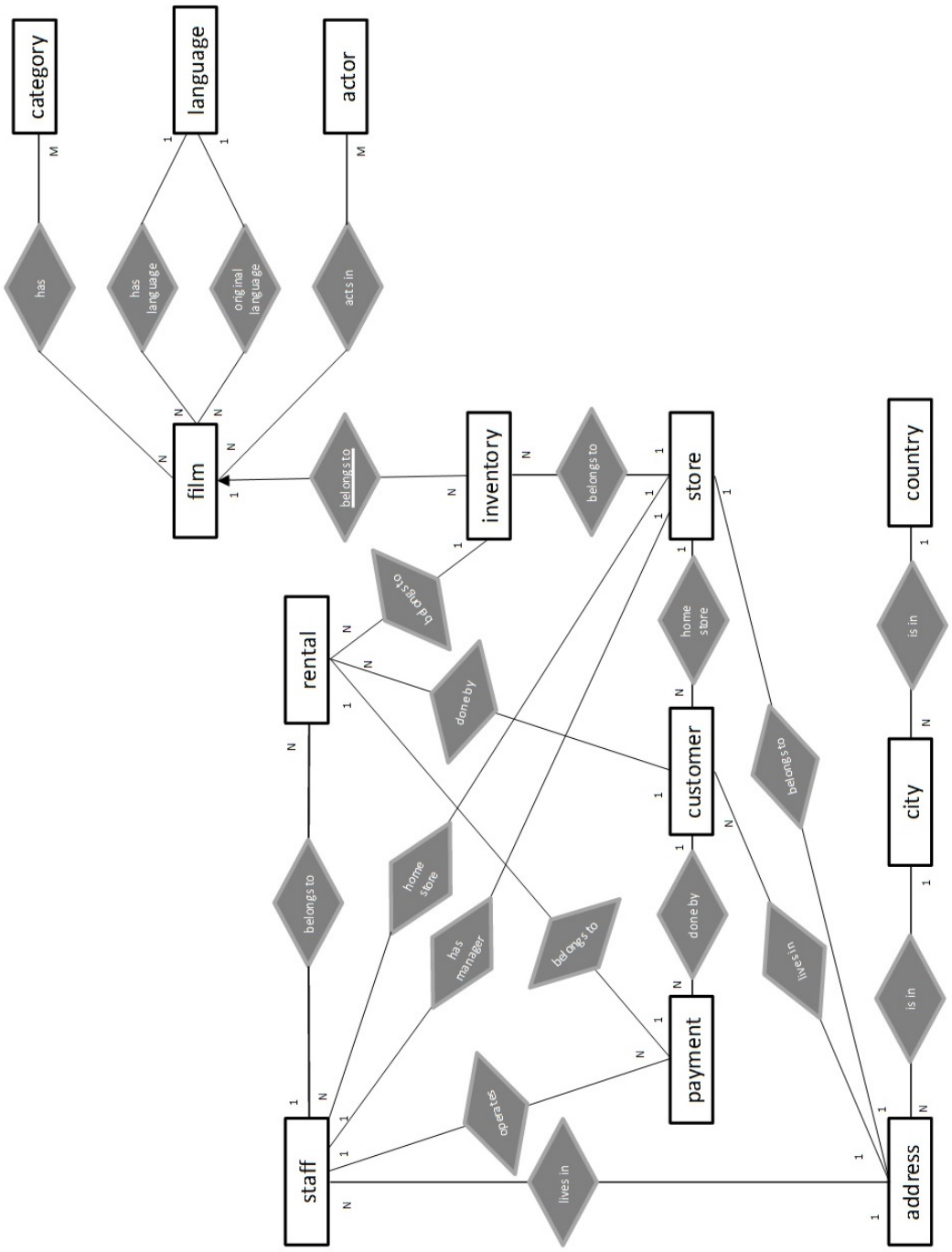

Abbildung 1: ER-Modell

2.2 Relationenmodell

Im folgenden werden die Relationen der Datenbank mit ihren Attributen und Integritätsbedingungen beschrieben.

2.2.1 Tabelle:actor

Die Relation actor beinhaltet alle Informationen über die Schauspieler.

Tabelle: actor

Name	Typ	NULL	Beschreibung
actor_id	INTEGER	Nein	Eindeutige Kennung des Schauspielers (Primärschlüssel)
first_name	VARCHAR(45)	Nein	Vorname des Schauspielers
last_name	VARCHAR(45)	Nein	Nachname des Schauspielers
last_update	TIMESTAMP	Nein	Zeit der Erstellung oder des letzten Updates des Tupels

2.2.2 Tabelle:address

Die Relation address beinhaltet alle Informationen zu Adressen. Der Primärschlüssel dieser Relation ist Fremdschlüssel in den Relationen staff, customer und store. Die Relation adressiert die zugehörige Stadt durch einen Fremdschlüssel.

Tabelle: address

Name	Typ	NULL	Beschreibung
address_id	INTEGER	Nein	Eindeutige Kennung (Primärschlüssel)
address	VARCHAR(50)	Nein	Erste Zeile der Adressangabe
address2	VARCHAR(50)	Ja	Optionale zweite Zeile der Adressangabe
district	VARCHAR(20)	Nein	Region der Adresse
city_id	INTEGER	Nein	ID der Stadt (Fremdschlüssel)
postal_code	VARCHAR(10)	Ja	Postleitzahl
phone	VARCHAR(20)	Nein	Telefonnummer
last_update	TIMESTAMP	Nein	Zeit der Erstellung oder des letzten Updates des Tupels

2.2.3 Tabelle:category_sakila

Die Relation category_sakila listet alle Kategorien für Filme.

Tabelle: category_sakila			
Name	Typ	NULL	Beschreibung
category_id	SMALL INTEGER	Nein	Eindeutige Kennung (Primärschlüssel)
name	VARCHAR(25)	Nein	Kategoriename
last_update	TIMESTAMP	Nein	Zeit der Erstellung oder des letzten Updates des Tupels

2.2.4 Tabelle:city

Die Relation city enthält Informationen über die Städte. Die Relation city wird über einen Fremdschlüssel in der Relation address adressiert und verweist selbst auf die Relation country.

Tabelle: city			
Name	Typ	NULL	Beschreibung
city_id	INTEGER	Nein	Eindeutige Kennung (Primärschlüssel)
city	VARCHAR(50)	Nein	Stadtname
country_id	SMALL INTEGER	Nein	das Land, in dem die Stadt liegt (Fremdschlüssel)
last_update	TIMESTAMP	Nein	Zeit der Erstellung oder des letzten Updates des Tupels

2.2.5 Tabelle:country

Die Relation country Informationen zu den Ländern. Die Relation country wird über einen Fremdschlüssel in der Relation city adressiert.

Tabelle: country			
Name	Typ	NULL	Beschreibung
country_id	SMALL INTEGER	Nein	Eindeutige Kennung (Primärschlüssel)
country	VARCHAR(50)	Nein	Name des Landes
last_update	TIMESTAMP	Nein	Zeit der Erstellung oder des letzten Updates des Tupels

2.2.6 Tabelle:customer

Die Relation customer enthält Informationen über die Kunden. Die Relation customer wird über einen Fremdschlüssel in den Relationen payment und rental adressiert und verweist selbst auf die Relationen address und store.

Tabelle: customer			
Name	Typ	NULL	Beschreibung
customer_id	INTEGER	Nein	Eindeutige Kennung (Primärschlüssel)
store_id	INTEGER	Nein	Store, bei dem der Kunde hauptsächlich Filme leiht (d.h. der Home-Store des Kunden). Ein Kunde kann aber auch bei einem anderen Store Filme leihen. (Fremdschlüssel)
first_name	VARCHAR(45)	Nein	Vorname
last_name	VARCHAR(45)	Nein	Nachname
email	VARCHAR(50)	Ja	E-Mail-Adresse
address_id	INTEGER	Nein	Adresse des Kunden (Fremdschlüssel)
active	CHAR(1)	Nein	Flag, ob Kunde aktiv ist. Ein nicht-aktiver Kunde kann keine Filme ausleihen
create_date	TIMESTAMP	Nein	Zeit der Erstellung des Tupels
last_update	TIMESTAMP	Nein	Zeit der Erstellung oder des letzten Updates des Tupels

2.2.7 Tabelle:film

Die Relation film beinhaltet Informationen über die Filme. Die Relation film wird über einen Fremdschlüssel in den Relationen film_category, film_actor und inventory adressiert

und verweist selbst auf die Relation language.

Tabelle: film

Name	Typ	NULL	Beschreibung
film_id	INTEGER	Nein	Eindeutige Kennung (Primärschlüssel)
title	VARCHAR(255)	Nein	Titel des Films
description	BLOB	Ja	Beschreibung des Films
release_year	VARCHAR(4)	Ja	Erscheinungsjahr
language_id	SMALL INTEGER	Nein	Sprache des Films (Fremdschlüssel)
original_language_id	SMALL INTEGER	Ja	ggfs. Originalsprache des Films (Fremdschlüssel)
rental_duration	SMALL INTEGER	Nein	vorgesehene Dauer der Ausleihe
rental_rate	DECIMAL	Nein	Kosten der Ausleihe (für die vorgesehende Dauer)
length	SMALL INTEGER	Ja	Filmdauer
replacement_cost	DECIMAL	Nein	Kosten der Wiederbeschaffung bei Nicht-Rückgabe oder Beschädigung einer Filmkopie dieses Films
rating	VARCHAR(10)	Ja	Bewertung; mögliche Werte sind: G, PG, PG-13, R oder NC-17
special_features	VARCHAR(100)	Ja	Spezielle Ergänzungen des Films, wie z.B. Trailers, Commentaries, Deleted Scenes, Behind the Scenes
last_update	TIMESTAMP	Nein	Zeit der Erstellung oder des letzten Updates des Tupels

2.2.8 Tabelle:film_actor

Die Relation film_actor verknüpft die Schauspieler zu den gegebenen Filmen.

Tabelle: film_actor			
Name	Typ	NULL	Beschreibung
actor_id	INTEGER	Nein	Eindeutige Kennung des Schauspielers (Teil des Primärschlüssels)
film_id	INTEGER	Nein	Eindeutige Kennung des Films (Teil des Primärschlüssels)
last_update	TIMESTAMP	Nein	Zeit der Erstellung oder des letzten Updates des Tupels

2.2.9 Tabelle:film_category

Die Relation film_category weist den Filmen jeweils eine (oder mehrere) Kategorie(n) zu.

Tabelle: film_category			
Name	Typ	NULL	Beschreibung
film_id	INTEGER	Nein	Eindeutige Kennung des Films (Teil des Primärschlüssels)
category_id	SMALL INTEGER	Nein	Eindeutige Kennung der Kategorie (Teil des Primärschlüssels)
last_update	TIMESTAMP	Nein	Zeit der Erstellung oder des letzten Updates des Tupels

2.2.10 Tabelle:film_text

Die Relation film_text gibt den Titel und die Beschreibung der Filme an. Die zu der bereits in der Relation film enthaltenen redundanten Information zu Titel und Beschreibung des Films wird hier mit Indexen für den schnelleren Zugriff auf die Textattribute versehen.

Tabelle: film_text			
Name	Typ	NULL	Beschreibung
film_id	SMALL INTEGER	Nein	Eindeutige Kennung des Films (Primärschlüssel)
title	VARCHAR(255)	Nein	Titel des Films
description	BLOB	Ja	Beschreibung des Films

2.2.11 Tabelle:inventory

Die Relation inventory enthält die Information über die Filmkopien mit Angabe des Stores, in dem die jeweilige Kopie verfügbar ist. Alle Filmkopien eines Stores bilden dessen Gesamtbestand. Die Relation inventory wird über einen Fremdschlüssel in der Relation rental adressiert und verweist selbst auf die Relationen film und store.

Tabelle: inventory

Name	Typ	NULL	Beschreibung
inventory_id	INTEGER	Nein	Eindeutige Kennung (Primärschlüssel)
film_id	INTEGER	Nein	Film der zu verleihenden Filmkopie (Fremdschlüssel)
store_id	INTEGER	Nein	Store, in dem die Filmkopie verliehen wird (Fremdschlüssel)
last_update	TIMESTAMP	Nein	Zeit der Erstellung oder des letzten Updates des Tupels

2.2.12 Tabelle:language

Die Relation language liefert eine Liste mit möglichen Sprachen für einen Film. Die Relation language wird über Fremdschlüssel in der Relation film adressiert.

Tabelle: language

Name	Typ	NULL	Beschreibung
language_id	SMALL INTEGER	Nein	Eindeutige Kennung der Sprache (Primärschlüssel)
name	CHAR(20)	Nein	Sprachenname
last_update	TIMESTAMP	Nein	Zeit der Erstellung oder des letzten Updates des Tupels

2.2.13 Tabelle:payment

Die Relation payment enthält Informationen über die Bezahlung der Ausleihe einer Filmkopie durch einen Kunden. Die Relation payment verweist über Fremdschlüssel auf die Relationen customer, rental und staff.

Tabelle: payment			
Name	Typ	NULL	Beschreibung
payment_id	INTEGER	Nein	Eindeutige Kennung (Primärschlüssel)
customer_id	INTEGER	Nein	Kunde, der bezahlt (Fremdschlüssel)
staff_id	SMALL INTEGER	Nein	Angestellter, welcher die Bezahlung abwickelt (Fremschlüssel)
rental_id	INTEGER	Ja	Ausleihe, die bezahlt wird. Diese Angabe ist optional, da es auch Bezahlungen gibt, die nicht direkt zu einer Ausleihe gehören, wie z.B. ausstehende Zahlungen (Fremdschlüssel)
amount	DECIMAL	Nein	Betrag der Bezahlung
payment_date	TIMESTAMP	Nein	Datum, an dem die Bezahlung stattfand (Datum und Uhrzeit)
last_update	TIMESTAMP	Nein	Zeit der Erstellung oder des letzten Updates des Tupels

2.2.14 Tabelle:rental

Die Relation rental enthält Informationen über die Ausleihe einer Filmkopie. Die Relation rental wird über einen Fremdschlüssel in der Relation payment adressiert und verweist selbst auf die Relationen inventory, customer und staff.

Tabelle: rental			
Name	Typ	NULL	Beschreibung
rental_id	INTEGER	Nein	Eindeutige Kennung (Primärschlüssel)
rental_date	TIMESTAMP	Nein	Datum und Uhrzeit der Ausleihe
inventory_id	INTEGER	Nein	Filmkopie der Ausleihe (Fremdschlüssel)
customer_id	INTEGER	Nein	Kunde, der ausleiht (Fremdschlüssel)
return_date	TIMESTAMP	Ja	Datum, zu dem die Ausleihe zurückgegeben wurde
staff_id	SMALL INTEGER	Nein	Angestellter, der die Ausleihe betreut (Fremdschlüssel)
last_update	TIMESTAMP	Nein	Zeit der Erstellung oder des letzten Updates des Tupels

2.2.15 Tabelle:staff

Die Relation staff listet alle Mitarbeiter der Stores. Die Relation staff wird über Fremdschlüssel in den Relationen rental, payment und store adressiert und verweist selbst auf die Relationen store und address.

Tabelle: staff			
Name	Typ	NULL	Beschreibung
staff_id	SMALL INTEGER	Nein	Eindeutige Kennung (Primärschlüssel)
first_name	VARCHAR(45)	Nein	Vorname
last_name	VARCHAR(45)	Nein	Nachname
address_id	INTEGER	Nein	Adresse des Mitarbeiters (Fremdschlüssel)
picture	BLOB	Ja	Bild des Mitarbeiters
email	VARCHAR(50)	Ja	E-Mail-Adresse des Mitarbeiters
store_id	INTEGER	Nein	Store, dem der Mitarbeiter primär zugordnet ist ("home store"). Der Mitarbeiter kann auch (teilweise) in anderen Stores arbeiten (Fremdschlüssel)
active	SMALL INTEGER	Nein	Angabe, ob der Mitarbeiter aktueller Angestellter ist
username	VARCHAR(16)	Nein	Benutzername im Verleihsystem
password	VARCHAR(40)	Ja	Passwort im Verleihsystem
last_update	TIMESTAMP	Nein	Zeit der Erstellung oder des letzten Updates des Tupels

2.2.16 Tabelle:store

Die Relation store listet alle Stores. Die Relation store wird über Fremdschlüssel in den Relationen staff, customer und inventory adressiert und verweist selbst auf die Relationen staff und address.

Tabelle: store

Name	Typ	NULL	Beschreibung
store_id	INTEGER	Nein	Eindeutige Kennung (Primärschlüssel)
manager_staff_id	SMALL INTEGER	Nein	Manager des Stores (Fremdschlüssel)
address_id	INTEGER	Nein	Adresse des Stores (Fremdschlüssel)
last_update	TIMESTAMP	Nein	Zeit der Erstellung oder des letzten Updates des Tupels

3 SQL-Aufgaben

(80 Punkte) Ziel ist es, dass Sie gezielt bestimmte SQL-Konzepte üben, die in der Vorlesung vorgestellt wurden.

3.1 Bewertung

Nachdem Sie die Lösung im Portal einreichen, wird diese sofort automatisch geprüft. Eine korrekte Lösung wird mit voller Punktzahl bewertet, eine fehlerhafte Bearbeitung bekommt keine Punkte. Sie können so viele Lösungen einreichen, wie Sie wollen; es wird jeweils die letzte Einreichung gewertet. Ihre Gesamtpunktzahl wird über das Portal angezeigt. Bitte beachten Sie, dass wir die Anfragen nach dem Abgabedatum individuell inspizieren werden, um die Lösungen auf Brute-Force-Ansätze zu überprüfen. Sollten Ihre Einreichungen Brute-Force-Ansätze verwenden, werden die gesamten SQL-Aufgaben mit null Punkten bewertet.

3.2 Struktur der Lösung

Sehr wichtig für die Abgabe im Portal ist, dass Sie die Reihenfolge der Attribute in der Projektionsliste aus der Aufgabenstellung einhalten.

3.3 Aufgaben

Teilaufgabe 1:

Bestimmen Sie den durchschnittlichen Betrag aller Zahlungen, die von Kunden ausgeführt wurden, deren ID zwischen jeweils einschließlich "..." und "..." liegt. Betrachten Sie nur die Zahlungen, welche den Betrag von 5.00 Geldeinheiten übersteigen, und die von einem aktuell angestellten Mitarbeiter abgewickelt wird . Runden Sie das Ergebnis auf 3 Nachkommastellen.

Projektionsliste: Durchschnitt_Betrag

(2.5 Punkte)

Teilaufgabe 2:

Finden Sie alle Kunden, die aus dem Land "..." kommen.

Projektionsliste: country, first_name, last_name

(2.5 Punkte)

Teilaufgabe 3:

Finden Sie alle Filme, welche aus der Kategorie mit der ID "..." stammen.
Projektionsliste: Filmtitel, Filmkategoriename

(5 Punkte)

Teilaufgabe 4:

Finden Sie die 10 Länder, aus welchen die größte Anzahl an Kunden stammt. Geben Sie im Ergebnis die Anzahl der Kunden sowie den Namen des entsprechenden Landes an.

Projektionsliste: Land, AnzahlKunden

(5 Punkte)

Teilaufgabe 5:

Finden Sie alle Namen von Kunden, die im Durchschnitt mehr als 5 Geldeinheiten für einen ausgeliehenen Film bezahlt haben. Geben Sie in Ihrem Ergebnis den Vor-und Nachnamen des Kunden, sowie dessen durchschnittlich bezahlte Leihgebühr (leihschnitt) an. Runden Sie die Leihgebühr auf zwei Stellen hinter dem Komma. Ordnen Sie das Ergebnis absteigend nach der durchschnittlich bezahlten Leihgebühr.

Projektionsliste: Vorname, Nachname, leihschnitt

(5 Punkte)

Teilaufgabe 6:

Finden Sie heraus, wie häufig gleiche Filme im Store mit der ID = "..." auf Lager sind. Geben Sie im Ergebnis die Häufigkeit gleicher Filme im Lager (Haeufigkeit) sowie die Anzahl der unterschiedlichen Filme (Anzahl) an, welche zu der gegebenen Häufigkeit auf Lager sind. Sortieren Sie das Ergebnis absteigend nach der Häufigkeit gleicher Filme.

Projektionsliste: Haeufigkeit, Anzahl

(7.5 Punkte)

Teilaufgabe 7:

Finden Sie den jeweils längsten Film eines Schauspielers und geben Sie den Namen des Schauspielers und die Filmlänge für die 10 größten Längen an. Geben Sie im Ergebnis jeweils die Länge des Films in absteigender Reihenfolge sowie den Vor- und Nachnamen des Schauspielers an.

Projektionsliste: Vorname, Nachname, Filmlaenge

(7.5 Punkte)

Teilaufgabe 8:

Finden Sie alle Kunden mit dem Home-Store "...", deren Vor- und Nachname den gleichen Anfangsbuchstaben haben und die mindestens einen Film ausgeliehen haben, dessen Titel mit dem Buchstaben "..." endet und dessen Rating "*PG*" ist. Sortieren Sie das Ergebnis aufsteigend nach dem Nachnamen und nach dem store_id.

Projektionsliste: first_name, last_name, home_store_id
(10 Punkte)

Teilaufgabe 9:

Bestimmen Sie für jeden Store die Anzahl der Filme (d.h. Filmkopien), die am "..." zurückgegeben wurden und die Anzahl der Filme, die an diesem Datum ausgeliehen waren. Beachten Sie für Letzeres , dass dies der Fall ist, falls der Film vor oder an diesem Datum ausgeliehen wurde und dann entweder erst nach dem Datum zurückgegeben wurde oder noch ausgeliehen ist.
Sortieren Sie absteigend nach der Anzahl der zurückgegebenen Filme.

Projektionsliste: store_id, return_date, Anzahl_Filme_Rueckgabe, Anzahl_Filme_Ausgeliehen

(10 Punkte)

Teilaufgabe 10:

Geben Sie für ein Geschäft mit store_id = 1 eine Liste aller Filme und die Anzahl der verfügbaren Filmkopien aus und zwar für Filme, die mit dem Buchstaben "..." enden. Eine Filmkopie ist verfügbar, d.h. sie steht zur Ausleihe bereit, wenn sie bisher noch nie ausgeliehen wurde oder wenn es keine Ausleihe (rental) gibt, die noch nicht zurückgegeben wurde. Sortieren Sie die Liste aufsteigend nach store_id und absteigend nach dem Filmtitel.
Projektionsliste: store_id, title, AnzAusleihbar

(10 Punkte)

Teilaufgabe 11:

Geben Sie für den Kunden mit der ID = "..." den Betrag an, den dieser am Ende des 30.07.2005 in den Filmverleihgeschäften noch zu zahlen hat (offener Rechnungsbetrag zu diesem Tag). Dabei werden Ausleihen, die erst nach dem vorgegebenen Datum durchgeführt werden, sowie Zahlungen, die erst nach diesem Datum erfolgen, bei der Berechnung nicht berücksichtigt.
Hinweise: Zu beachten sind die Leihgebühren für alle Ausleihen bis zu dem angegebenen Tag; eine Geldeinheit für jeden Tag, den vorherige Ausleihen überfällig waren; falls ein Film mehr als die vorgeschriebene Ausleihdauer (RENTAL_DURATION) überfällig ist, die Ersetzungskosten (REPLACEMENT_COST) sowie alle Zahlungen bis zum angegebenen Datum. Sollte ein Betrag offen stehen, dann soll dieser als Minusbetrag ausgegeben werden.
Sollten Sie für die Berechnung eine Fallunterscheidung benötigen, kann in SQL-Anweisungen ein CASE verwendet werden:
CASE
WHEN ... ((THEN ...) (ELSE ...))
(WHEN ...)
END
Die Klammern geben optionale Teile an. Es sind mehrere WHEN-Anweisungen möglich.
Projektionsliste: customer_id, customer_balance

(15 Punkte)