Übungsaufgaben zu Kapitel 5

Edwin Schicker

Die Aufgaben beziehen sich auf die Relationen der Beispieldatenbank *Bike* im Anhang.

Aufgabe 1

Schreiben Sie alle *Create-Table*-Befehle zum Erzeugen der Beispieldatenbank *Bike*. Geben Sie alle Entitäts- und Referenz-Integritätsregeln in Form von Tabellen- und Spaltenbedingungen an. Ergänzen Sie diese Angaben durch sinnvolle weitere Integritätsbedingungen (*Unique*, *Not Null*, *Check*).

Aufgabe 2

Fügen Sie zur Relation *Auftragsposten* das Attribut *Einzelpreis* hinzu. Füllen Sie dieses Attribut mit Daten auf, ermittelt aus den Attributen *Anzahl* und *Gesamtpreis*. In welcher Normalform befindet sich jetzt die Relation *Auftragsposten*?

Aufgabe 3

Erzeugen Sie in der Beispieldatenbank *Bike* eine Sicht *VPers*, die der Relation *Personal* ohne die Attribute *Gehalt* und *Beurteilung* entspricht. Weiter sind in dieser Sicht nur die Personen aufzunehmen, denen ein Vorgesetzter zugeordnet ist. Liegt eine änderbare Sicht vor?

Aufgabe 4

Die Relation *Auftragsposten* enthält aus Redundanzgründen nur den Gesamtpreis jedes einzelnen Auftragspostens. Schreiben Sie daher eine Sicht *VAuftragsposten*, die alle Daten der Relation *Auftragsposten* enthält und zusätzlich ein Attribut *Einzelpreis*. Ist diese Sicht änderbar?

Aufgabe 5

In SQL gibt es keinen *Alter-Schema*-Befehl. Überlegen Sie, wie in ein existierendes Schema weitere Schemaelemente aufgenommen werden können. Zeigen Sie dies an einem Beispiel.

Aufgabe 6

Beim Einfügen und Ändern von Artikeln soll automatisch aus dem Nettopreis die Mehrwertsteuer (19%) und der Gesamtpreis ermittelt werden. Schreiben Sie einen geeigneten Trigger. Testen Sie den Trigger.

Aufgabe 7

Alle neuen Kunden sollen automatisch mit einer Kundennummer versehen werden. Diese Nummern beginnen bei 21. Es sollen nur ungeradzahlige Kundennummern vergeben werden. Schreiben Sie eine geeignete Sequenz. Testen Sie diese Sequenz durch Hinzufügen von neuen Kunden.

Aufgabe 8

In MySQL gibt es die Spaltenbedingung *AutoIncrement*. Damit erhält dieses Attribut immer eine eindeutige automatische Nummer. Bilden Sie diese Funktion mittels Sequenzen und Trigger für das Attribut *Persnr* der Relation *Personal* nach (für Oracle und SQL-Server).

Aufgabe 9

In einem sicheren Datenbankverwaltungssystem wird vor jedem Zugriff auf eine Relation überprüft, ob der Benutzer die erforderlichen Zugriffsrechte besitzt. Ein Datenbankzugriff besteht demnach faktisch aus zwei Zugriffen, Sicherheitsüberprüfung und eigentlicher Zugriff. Stimmt diese Aussage wirklich?

Aufgabe 10

Schreiben Sie einen Befehl, der dem Benutzer *Gast* Änderungsrechte auf die Attribute *Bestand*, *Reserviert* und *Bestellt* der Relation *Lager* und Leserechte auf die gesamte Relation einräumt.

Aufgabe 11

Entziehen Sie dem Benutzer *Gast* die in der vorherigen Aufgabe gewährten Rechte wieder.

Aufgabe 12

Schreiben Sie alle notwendigen Befehle, damit der Benutzer *Gast* nur Leserechte auf die Attribute *Artnr*, *Lagerort* und *Bestand* der Relation *Lager* bekommt. Weiter darf er Tupel dieser Relation nicht sehen, falls Mindestbestand plus reservierte Teile größer als der tatsächliche Bestand ist. Diese Rechte darf der Benutzer *Gast* auch weiterreichen.

Aufgabe 13

Um die Integrität zu optimieren, sollen die Attribute *GebDatum*, *Stand*, *Gehalt* und *Beurteilung* der Relation *Personal* auf zulässige Werte überprüft werden. Es ist bekannt, dass alle Mitarbeiter zwischen 1970 und 2010 geboren sind, entweder ledig, verheiratet, geschieden oder verwitwet sind, das Gehalt zwischen 500 und 6000 Euro liegt und die Beurteilung entweder *Null* oder einen Wert zwischen 1 und 10 besitzt. Fügen Sie diese Bedingungen mittels geeigneter *Alter-Table*-Befehle hinzu, wobei sicherzustellen ist, dass diese Bedingungen, falls gewünscht, auch wieder entfernt werden können (bitte Constraintnamen vergeben!).