

Projekt "ShopShoee"

letzte Änderung: 7. Juni 2018

Erstellen Sie als Projektaufgabe die Datenbank zum Webshop "ShopShoee".

Kriterien

- **Regel Nr 1: Lesen Sie die Aufgaben vollständig durch bevor Sie beginnen!**
- **Die Abgabe erfolgt spätestens am Montag den 02.07.2018.** Sie können maximal zu viert abgeben. Sprechen Sie sich gut ab, damit Sie nicht gegenseitig Ihre Änderungen überschreiben.
- Das Modell muss in einer **MySQL Workbench Datei** für Windows vorliegen. Das Diagramm soll möglichst übersichtlich und selbsterklärend sein.
- Die modellierte **Datenbank muss mit aussagekräftigen, korrekten Beispieldaten gefüllt online sein.** Nutzen Sie Ihre Projektdatenbank, die Sie von mir bekommen.
- Das **Beispielskript** (siehe Abb. 1) soll lauffähig sein.
- **Dokumentieren Sie Ihr Ergebnis:** Geben Sie mindestens einen Beispielaufruf für jede Prozedur an, möglichst so, dass gezeigt wird, dass die Prozedur ihren Zweck bei der Konsistenzwahrung erfüllt. Der Quellcode soll gut nachvollziehbar sein. Kommentieren Sie den Code der Prozeduren.
- Nutzen Sie **Forward Engineering.** Legen Sie Grunddaten in den Tabellen des Modells ab. Lassen Sie Ihre Prozeduren komplexere Daten erzeugen. Nutzen Sie dazu eine SQL Datei im Modell, die beim Einspielen in die Datenbank automatisch ausgeführt wird. Forward Engineering muss **einen vollständigen, konsistenten Zustand** ergeben.
- Wählen Sie **gute Bezeichnungen** für Tabellen, Spalten und Variablen. Nutzen Sie $t_$, $p_$ und $v_$
- **Beachten Sie die Integritätsregeln für Tabellen.** Achten Sie darauf, dass die Normalformen erfüllt sind, und dass es möglichst wenige Redundanzen und NULL-Spalten gibt. Erstellen Sie keine überflüssigen Tabellen. Fremdschlüsselbeschränkungen sollen korrekt gesetzt sein.
- **Beachten Sie alle Integritätsbedingungen für Prozeduren,** die sich direkt aus dem Text ergeben. Erfinden Sie keine zusätzlichen. Fehler sollen abgefangen werden. Benutzen Sie Transaktionen und Exceptions.
- **Das Ergebnis wird bewertet und zählt als 50% Ihrer Kursnote.**
Ich werde Sie ggf. zu Ihrem Entwurf und Ihrer Umsetzung befragen.

Bewertung: Sie können 180 Punkte erreichen. 140 Punkte werden als 100% gewertet. Sie benötigen 70 Punkte zum Bestehen des Projektes.

Punkte	Thema
60	Tabellenstruktur
20	Daten
30	Sichten
30	Prozeduren
10	Event
10	Funktion
10	Trigger
10	Analyse
180	Summe

Einleitung

Ein Schuhgeschäft möchte Produkte online verkaufen. Produkte sind Schuhe, deren Modelle in verschiedenen Varianten vorhanden sind.

Dazu gehört ebenfalls ein Shopsystem. Kunden können Schuhe über einen Warenkorb bestellen.

Zur Datenbank gehört auch ein Analysesystem zur Untersuchung der Verkaufszahlen.

Die Datenbank soll genutzt werden um

- das Angebot anzeigen zu können,
- fehlerhafte Daten zu finden,
- Umsätze auswerten zu können.

Sie soll flexibel genug sein, um später leicht erweitert oder verändert werden zu können.

Schuhe

Jedes Schuhmodell kann in verschiedenen Varianten vorhanden sein. Jedes Modell hat einen Namen und einen Hersteller, von dem Name und Land gespeichert werden müssen, und mindestens eine Kategorie.

Schuhe haben einen (Verkaufs-)Preis in Euro, der von der Variante abhängen kann. Die Varianten unterscheiden sich in Größe, Material, Typ und Farbe. Die Typen können Herren-, Damen- oder Kinderschuhe sein. Ein Schuh kann mehrere Farben haben, die alle auch gespeichert werden. Beim Material wird zwischen Schuhmaterial und Sohlenmaterial unterschieden. Es soll eine feste Umrechnung für Schuhgrößen hinterlegt sein. Es genügt vorerst, die Umrechnung deutsche Größe (z.B. 41) und amerikanische Größe (z.B. 8,5) zu speichern¹.

Ein Modell / eine Variante gibt nur dann als vollständig korrekt, wenn es zum Modell mindestens eine Variante gibt, ein Preis > 0 hinterlegt ist, eine Größe aus einer vorgegebenen erlaubten Liste hinterlegt ist, ein vollständiger Hersteller angegeben ist, mindestens eine Farbe und sowohl Schuh- als auch Sohlenmaterial.

Shopsystem

Kunden können Schuhe bestellen. Eine Bestellung kann beliebige Schuhvarianten in beliebiger Menge enthalten, aber nur Modelle / Varianten, die fehlerfrei sind. Sie wird zuerst leer begonnen (als leerer Warenkorb), und die Kunden fügen nach und nach Schuhe hinzu, solange die Bestellung nicht abgeschlossen ist. Bei jeder Bestellung ist wichtig, ob sie abgeschlossen wurde, d.h. ausgeliefert werden kann, und ob sie bezahlt ist. Es soll der Zeitpunkt gespeichert werden, wenn die Bestellung angefangen, abgeschlossen und bezahlt wurde. Zu jeder Bestellung gehört ein eindeutiger Kunde mit einer Lieferadresse. Außerdem hat jede Bestellung einen Gesamtwert. Eine Bestellung kann nur einmal bezahlt werden und nur, wenn sie abgeschlossen ist. Ein Kunde kann nur dann einen neuen Warenkorb anlegen, wenn er keinen offenen Warenkorb mehr hat. Warenkörbe können gelöscht werden.

Kundensystem

Kunden haben eine eindeutige Emailadresse. Jeder Kunde muss eine Lieferadresse haben und ein Geburtsdatum. Eine Adresse besteht in jedem Fall aus Vorname, Nachname, Straße, Postleitzahl und Ort.

Analysesystem

Das Analysesystem soll im Sternschema organisiert sein. Die Fakten sind Anzahl und Preis der gekauften und bezahlten Schuhe. Es soll außerdem in Dimensionstabellen gespeichert werden: Alter (in Jahren) und Wohnort der Kunden, Modellname, Typ, Herstellername und -land, Material des Schuhs und der Sohle, Schuhgröße, das Bestelldatum (auch als Wochentag, als Monat und als Jahr). Bei Bezahlen eines Warenkorbes soll eine Kopie aller zugehörigen relevanten Daten im Analysesystem angelegt werden.

Aufgaben

Tabellen (60 + 20 Punkte)

Erstellen Sie die benötigten Tabellen und füllen Sie sie mit konsistenten Beispieldaten. Sie benötigen:

- mindestens 5 Modelle mit insgesamt mindestens 10 vollständig korrekten Varianten

¹Ignorieren Sie bei der Umrechnung der Einfachheit halber das Geschlecht

- mindestens 3 Kunden
- mindestens 5 Bestellungen, davon mindestens 2 bezahlte

Sichten (30 Punkte)

Erstellen Sie je eine Sicht

- `v_allevarianten` (10 Punkte)
Alle korrekten Modelle in allen Varianten mit sämtlichen Detailinformationen.
- `v_variantenkurz` (10 Punkte)
Alle korrekten Modelle in allen Varianten mit sämtlichen Detailinformationen. Hier soll es je Variante genau eine Zeile geben. Sollten mehrere Kategorien / Farben hinterlegt sein, so sollen die Informationen hier in einer Zelle zusammengefasst sein (etwa `farben: schwarz, weiß`).
- `v_fehlermodelle` (10 Punkte)
Alle nicht korrekten Modelle / Varianten. Es soll eine zusätzliche Spalte geben, in der einer der Fehler beschrieben ist, die dazu führen, dass das Modell nicht korrekt ist (z.B. kein Preis).

Erstellen Sie ggf. Hilfssichten, die Ihnen die Arbeit erleichtern.

Prozeduren (30 Punkte)

Erstellen Sie

- eine Prozedur `p_neuer_warenkorb` (IN `pKunde`, OUT `pWarenkorb`) (6 Punkte)
Ein neuer leerer Warenkorb ohne Posten wird zu einem Kunden angelegt. Der PK des Warenkorbes wird zurückgegeben.
- eine Prozedur `p_neuesWarenkorbelement` (IN `pWarenkorb`, IN `pVariante`, IN `pAnzahl`) (12 Punkte)
Eine Schuhvariante wird in einer gegebenen Anzahl einem Warenkorb hinzugefügt.
- eine Prozedur `p_warenkorb_abschliessen` (IN `pWarenkorb`) (6 Punkte)
Ein Warenkorb wird abgeschlossen.
- eine Prozedur `p_warenkorb_bezahlen` (IN `pWarenkorb`) (6 Punkte)
Ein Warenkorb wird bezahlt.

Verwenden Sie Transaktionen und Exceptions, wo möglich. Fangen Sie alle Fehler ab.

Beispielsweise darf `p_neuer_warenkorb` (IN `pKunde`, OUT `pBestellung`) keinen Fehler ergeben, falls der Kunde zum Parameter `pKunde` nicht existiert.

Erstellen Sie ggf. Hilfsprozeduren, die Ihnen die Arbeit erleichtern.

Event (10 Punkte)

Programmieren Sie ein Event, das einmal pro Minute eine Bestellung macht über ein zufälliges Produkt für einen zufälligen Kunden.

Hinweis: Einen zufälligen Datensatz wählen Sie mit `SELECT ... ORDER BY RAND() LIMIT 1`

Funktionen (10 Punkte)

Programmieren Sie eine Funktion, die ein Geburtsdatum in ein Alter (in Jahren) umrechnet.

Trigger (10 Punkte)

Implementieren Sie mit einem Trigger, dass eine Änderung an einem Preis eines Modells automatisch den Gesamtwert aller offenen Warenkörbe mit diesem Modell angepasst.

Analyse (10 Punkte)

Verbinden Sie über Power Pivot für Excel mit ihrem Analysesystem und erstellen Sie zwei Charts zur Auswertung der Daten.



Beispielskript

```
# Neuer Warenkorb, Kunde 1
CALL p_neuerWarenkorb(1, @warenkorb);

# 3 Paar der Schuhe mit Nummer 2 hinzufügen
CALL p_neuesWarenkorbelement(@warenkorb, 2, 3);
# 2 Paar der Schuhe mit Nummer 1 hinzufügen
CALL p_neuesWarenkorbelement(@warenkorb, 1, 2);

# Warenkorb abschließen
CALL p_warenkorb_abschliessen(@warenkorb);

# Warenkorb bezahlen
CALL p_warenkorb_bezahlen(@warenkorb);
```

Tabelle 1: Beispiel Skript

Tipps

Sie machen Forward Engineering. Nutzen Sie die Vorteile! Löschen Sie im Falle von größeren Problemen alles in Ihrer Projektdatenbank. Ein neues Einspielen kann viel leichter sein als die Korrektur von etwas Fehlerhaftem.

Generell gilt

- Zeichnen Sie zuerst ein ER-Diagramm auf Papier. Überlegen Sie sich, wie die Tabellen dazu aussehen würden, und welche Daten sie dort speichern würden.
- Packen Sie so viele Konsistenzbedingungen wie möglich in die Tabellenstruktur.
- Grundlegende Änderungen an Tabellen (Umbenennen, Änderungen an PK oder FK) sind bei DBS unbeliebt und nicht immer ohne Gegenwehr möglich.
- Notieren Sie sich vorab, was die Prozeduren mit den Daten machen sollen.
- Legen Sie dann die Tabellen an und füllen Sie sie mit Beispieldaten. Verwenden Sie `NOT NULL`, wenn nicht anders naheliegend.
- Sichten erstellen Sie am besten, indem Sie zuerst das passende `SELECT` direkt auf dem Server konstruieren und die Sicht erst anschließend anlegen. Achtung: Bei Sichten dürfen (anders als bei Ergebnistabellen) keine Spaltennamen doppelt vorkommen und Sie dürfen keine Subqueries verwenden.
- Es kann günstig sein, zuerst die Sichten anzulegen und dann die Prozeduren; dann sind die Sichten in den Prozeduren nutzbar.
- Bei Prozeduren erstellen Sie ggf. zuerst eine Grobversion, die das Wesentliche macht, und fügen danach die Konsistenzbedingungen ein.
- Prozeduren bearbeiten Sie ggf. am einfachsten direkt auf dem Server. Sie können so testen, ohne jedesmal das Modell einspielen zu müssen.
- Fügen Sie in Prozeduren Exceptions erst ganz zum Schluss ein. So können Ihnen Fehlermeldungen bei der Entwicklung helfen.
- Legen Sie Beispieldaten, sofern möglich, am besten durch Ihre Prozeduren an. Dadurch werden manche Konsistenzbedingungen automatisch eingehalten.
- Prüfen Sie ggf. am Ende Ihre Beispieldaten.
- Wenn Sie an einer Stelle unsicher sind, wie die Aufgabe zu verstehen ist, fragen Sie frühzeitig in der Übung nach.