

Praktische Übung “Programmentwicklung mit PL/SQL: Programmeinheiten“

1. Studieren Sie die Möglichkeiten des Oracle SQL Developers zur Programmierung von Prozeduren.

2. Stored Procedures: Einfügen

Schreiben Sie eine Prozedur, um ein neues Produkt in eine Tabelle PRODUCT einzufügen. Schauen Sie sich zu diesem Zweck die Tabelle PRODUCT in Ihrer Datenbank an.

Compilieren Sie den Code der Prozedur, rufen Sie die Prozedur auf und kontrollieren Sie, wie sich der Inhalt der Tabelle verändert hat.

Rufen Sie die Prozedur erneut auf mit der Product-ID 100860. Was passiert dabei und warum?

3. Stored Procedures: Ändern

Schreiben Sie eine Prozedur UPD_PROD, um ein Produkt in der Produkte-Tabelle zu modifizieren. Diese soll eine Fehlerbehandlung beinhalten, wenn die Operation nicht ausgeführt werden konnte.

Compilieren Sie den Code, rufen Sie die Prozedur auf und prüfen Sie das Ergebnis in der Produkte-Tabelle. Überprüfen Sie auch die Wirkungsweise Ihrer Fehlerbehandlung, indem Sie versuchen, ein Produkt zu ändern, das gar nicht in der Tabelle existiert.

4. Stored Procedures: Löschen

Schreiben Sie eine Prozedur DEL_PROD, um ein Produkt aus der Produkte-Tabelle zu löschen. Diese soll eine Fehlerbehandlung beinhalten, wenn die Operation nicht erfolgreich ausgeführt werden konnte.

Compilieren Sie den Code, rufen Sie die Prozedur auf und prüfen Sie das Ergebnis in der Produkte-Tabelle. Überprüfen Sie auch die Wirkungsweise Ihrer Fehlerbehandlung, indem Sie versuchen, ein Produkt zu löschen, das gar nicht in der Tabelle existiert.

5. Stored Function

Schreiben Sie eine Stored Function ANNUAL_COMP, die mit einem gegebenen Monatsgehalt und einer Kommission das jährliche Gehalt berechnet. Dabei sind eventuell auftretende NULL-Values zu berücksichtigen. Die Funktion ist in einem SELECT-Statement beim Auslesen der Angestellten-Daten zur Ermittlung des Jahresgehalts zu benutzen.

6. Trigger zur Zugriffskontrolle

Änderungen in der Datenbank sind nur auf Tabellen erlaubt während der normalen Bürozeiten zwischen 8:45 und 17:30 Uhr, Montag bis Freitag.

Erzeugen Sie eine Stored Procedure `SECURE_DML`, die außerhalb dieser Zeiten eine Fehlermeldung produziert: "Änderungen nur während der normalen Arbeitszeit möglich".

Erzeugen Sie einen Trigger, der als Aktion die Prozedur `SECURE_DML` aufruft.

Testen Sie Trigger bzw. Prozedur durch eine temporäre Modifikation des zu testenden Zeitintervalls in der Prozedur und versuchen Sie danach, einen neuen Datensatz in die Tabelle `PRODUCT` einzufügen.

7. Trigger zur Update Propagation

Die Kommission eines Vertriebsmitarbeiters (Salesman) sollte sich ändern bei neuen Bestellungen oder Änderungen von bestehenden Bestellungen. Ein Vertriebsmitarbeiter ist einem bestimmten Kunden in der Tabelle `CUSTOMER` zugeordnet.

Erzeugen Sie eine Prozedur zur Änderung der Kommission eines Vertriebsmitarbeiters. Verwenden Sie Parameter zur Übergabe von Kunden-ID, altem und neuem Gesamtbetrag der Bestellung (`ORDER.total`) vom aufrufenden Trigger. Die Prozedur muss dann die zugehörige Angestellten-Nr. des Vertreters aus der `CUSTOMER`-Tabelle finden und dessen Datensatz der `EMP`-Tabelle ändern. Die Änderung der Kommission geschieht durch Addition der neuen Kommission zum bestehenden Wert. Nehmen Sie für diese Übung eine fixe Kommissionsrate von 5 % an.

Erzeugen Sie einen Trigger auf der Tabelle `ORD`, der die obige Prozedur aufruft und ihr die notwendigen Parameter übergibt.

Verändern Sie zum Test von Prozedur und Trigger die Tabelle `ORD` (erzeugen Sie sich am besten eine Sicherheits-Kopie von `ORD`), konkret die Bestellung Nr. 610. Fügen Sie eine neue Bestellposition hinzu: Position 4, Produkt-ID 102130, Preis 3.4, Menge 1. Reduzieren Sie die bestellte Anzahl in Position 2 um 1.

Überprüfen, wie sich die Kommission des Angestellten `WARD` nach der Modifikation der Bestellung 610 geändert hat.