

Datenbanksysteme 2		Praktikum Blatt 04: PL/SQL Einführung	
Studiengang:	B. Sc. Medieninformatik	Bearbeitung vor dem Praktikum:	Erstellen Sie einen Lösungsentwurf in einer Zweiergruppe
Status:	Fertig	Teilnahmetestat zu Beginn des Praktikums:	Erläuterung Ihres Lösungsentwurfs
Bearbeiter:	Salgert, Lambert, Rakow	Praktikum auf Datenbank-Server:	Anpassungen am Lösungsentwurf; Erstellung des Log-Files zum Durchlauf
		Bearbeitungstestat am Ende des Praktikums:	Log-File des Durchlaufs

Ziele dieses Praktikums

- Variable und Datentypen von PL/SQL kennenlernen
- die Interaktion zwischen PL/SQL und der Datenbank anwenden

Die Bearbeitung

Bearbeiten Sie die folgenden Aufgaben in Ihrer Kleingruppe. Bereiten Sie alle Aufgaben zu Hause vor. Verwenden Sie zur Lösung der Aufgaben im Praktikum einen SQL-Worksheet im **SQL Developer** von Oracle. **Erstellen Sie ein Log-File**. Machen Sie sich mit dem neuen Datenbankschema „HR“ vertraut. Das Skript „hr_main.sql“ zur Erstellung des HR-Schemas finden Sie als Anlage in Moodle.

Beginnen Sie ein Skript immer mit:

```
set serveroutput on;
set echo on;
```

Aufgabe 1 (Grundlagen)

Erstellen Sie einen PL/SQL-Block, der in der departments-Tabelle die höchste Abteilungsnummer wählt und in die Variable `v_max_deptno` speichert.

Zeigen Sie die höchste Abteilungsnummer an. Gehen Sie wie folgt vor:

- a) Deklarieren Sie im deklarativen Bereich eine Variable `v_max_deptno` des `NUMBER`-Datentyps.
- b) Starten Sie den ausführbaren Bereich mit dem `BEGIN`-Schlüsselwort.
Nehmen Sie eine `SELECT`-Anweisung auf, mit der Sie die höchste `department_id` aus der `departments`-Tabelle abrufen können.
- c) Zeigen Sie `v_max_deptno` an, und beenden Sie den ausführbaren Bereich.
- d) Führen Sie Ihr Skript aus. Die Ausgabe kann beispielsweise wie folgt lauten:

```
anonymous block completed  
Die höchste Abteilungsnummer ist: 270
```

Aufgabe 2 (Komplexe Datentypen)

Erstellen Sie einen PL/SQL-Block, der Informationen über **ein** bestimmtes Land ausgibt. Deklarieren Sie einen PL/SQL-Record basierend auf der Struktur der `countries`-Tabelle wie folgt:

- a) Deklarieren Sie eine `v_countryid`-Variable. Weisen Sie `CA` der Variable `v_countryid` zu.
- b) Deklarieren Sie die Variable `v_country_record` mit einer Referenz auf den Typ des `countries`-Datentyps.
- c) Rufen Sie im ausführbaren Bereich mit `v_countryid` alle Daten zu einem Land aus der `countries`-Tabelle ab. Zeigen Sie die ausgewählten Informationen zu dem gewählten Land an. Die Ausgabe kann wie folgt aussehen:

Country Id:	CA
Country Name:	Canada
Region:	2
- d) Erstellen Sie eine Procedure, die die Daten zu einer CountryId anzeigt. Verwenden Sie folgende Signatur: `PROCEDURE SHOW_COUNTRY (p_country_id countries.country_id%type)`. Testen Sie die Procedure für die Länder mit den IDs `DE`, `UK` und `US`.

Aufgabe 3 (Integration ins SELECT)

Erstellen Sie einen PL/SQL-Funktion, die überprüft ob eine IBAN gültig ist. Die Funktion muss nur die Gültigkeit deutscher IBANs erkennen, ausländische IBANs können Sie ignorieren. Führen Sie die SQL-Datei `emp_account.sql` aus, um die Tabelle `emp_account` anzulegen. Schreiben Sie eine SQL-Abfrage die ausgibt, für welche Mitarbeiter die IBAN gültig ist. Die Ausgabe soll in etwa wie folgt aussehen:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	IBAN	SALARY	IS_VALID
198	Donald	OConnell	DE87648045234485030332	2600	OK

...
Hinweis: Die Berechnungsvorschrift zur IBAN finden Sie im Anhang.

Aufgabe 4 (Kontrollstrukturen)

Führen Sie das Skript lab_04_01.sql aus. Dieses Skript erstellt eine emp-Tabelle, die ein Replikat der employees-Tabelle ist. Es fügt eine neue Spalte (`stars`, Datentyp VARCHAR2 und Größe 50) in die emp-Tabelle ein.

Erstellen Sie eine auf dem Datenbankserver gespeicherte PL/SQL-Prozedur (= stored procedure) `set_employee_stars`, die für einen Mitarbeiter pro 1000 Euro des Mitarbeitergehalts ein Sternchen (*) in die `stars`-Spalte einfügt. Dazu können Sie sich an folgenden Schritten orientieren:

- Deklarieren Sie für die Parameter Übergabe der Prozedur eine `v_empno`-Variable des Typs `emp.employee_id`. Deklarieren Sie eine `v_asterisk`-Variable des `emp.stars`-Typs, und initialisieren Sie sie auf `NULL`. Erstellen Sie eine `sal`-Variable des `emp.salary`-Typs.
- Erstellen Sie im ausführbaren Bereich die Logik, um der Zeichenfolge für jede 1000 Euro Gehalt (gerundet auf volle 1000 Euro) ein Sternchen (*) hinzuzufügen. Wenn der ausgewählte Mitarbeiter beispielsweise 8200 Euro verdient, sollte die Zeichenfolge acht Sternchen enthalten. Verdient der Mitarbeiter 12500 Euro, sollte die Zeichenfolge 13 Sternchen enthalten. Komplizieren und speichern Sie das Ergebnis in der Datenbank.
- Rufen Sie für den Mitarbeiter mit der „employee_id“ 176 die zuvor erstellte Prozedur auf, um die Sternzeichenfolge zu setzen.
- Zeigen Sie die Zeile des entsprechenden Mitarbeiters aus der emp-Tabelle an, um zu prüfen, ob der PL/SQL-Block erfolgreich ausgeführt wurde.

Aufgabe 5 (Trigger)

Erstellen Sie einen Trigger, der beim Ändern oder Hinzufügen eines Angestellten automatisch die Spalte `stars` mit dem entsprechenden Wert füllt. Benutzen Sie folgendes Muster für den Trigger:

```
CREATE OR REPLACE TRIGGER emp_set_stars
  BEFORE INSERT OR UPDATE ON emp
  FOR EACH ROW
DECLARE
  --
BEGIN
  --
END;
```

Aufgabe 6 (Trigger)

In der Datenbank kam es wiederholt zu unerlaubten Änderungen der Gehälter von Mitarbeitern. Aus diesem Grund sollen Änderungen an der Spalte Gehalt protokolliert werden und nur zwischen 8 bis 18 Uhr an Werktagen erlaubt werden. Erstellen Sie einen Trigger der diese Aufgabe erfüllt.

Hinweis: Sie können die Funktion TO_CHAR(DATETIME) verwenden.

Aufgabe 7 (Cursor)

Die Angestellten des HR-Schemas sollen ausgegeben werden.

- a) Erstellen Sie in PL/SQL einen anonymen Block, der alle Daten eines bestimmten Angestellten aus der Datenbank holt und anschließend mit Hilfe einer in der Datenbank gespeicherten Prozedur `output_employee` formatiert ausgibt.
- b) Modifizieren Sie die in Teilaufgabe a) erstellte Prozedur so, dass jeweils bei den Fremdschlüsseln anstelle des Identifier für Beruf, Abteilung oder Vorgesetzter ein eindeutiger Text ausgegeben wird. Benennen Sie sie mit Prozedur `output_employee_b`.
- c) Erstellen Sie in PL/SQL eine stored Procedure `output_employees`, die alle Angestellten ausgibt, wobei die in Teilaufgabe b) erstellte Prozedur aufgerufen wird. Hinweis: Benutzen Sie hierfür einen Cursor.
- d) Modifizieren Sie den in Teilaufgabe c) erstellte Prozedur, so dass, die Angestellten in einer **neuen** Prozedur `output_employee_set` bereichsweise mit Anfangs- und Endnummer ausgegeben werden (z.B. 203-205, auf- und absteigend). Auch eine Ausgabe in umgekehrter Reihenfolge soll möglich sein. Beispiel für eine Signatur dieser Prozedur:

```
PROCEDURE output_employee_set (
    from_employee IN NUMBER,
    to_employee IN NUMBER,
    ascending_order IN BOOLEAN);
```

Mitarbeiter Nr.:	101
Name:	Neena Kochhar
E-Mail:	nkochhar@oracle.com
Telefon Nr.:	515.123.4568
Eingestellt am:	21.09.89
Gehalt:	17000 \$
Beruf ID:	AD_VP
Abteilung Nr.:	90
Vorgesetzter Nr.:	100

a) Beispiel für die formatierte Ausgabe eines Mitarbeiters

Aufgabe 8 (Exceptions)

- a) Modifizieren Sie die in Teilaufgabe 7b) erstellte Prozedur derart, dass die Daten eines Angestellten innerhalb der Prozedur - parametrisiert anhand seines Nachnamens – aus der Datenbank geholt und ausgegeben wer-

den. Behandeln Sie mögliche auftretende Exceptions. Erstellen Sie hierfür eine neue Procedur `output_employee_by_name`.

- b) Testen Sie den Aufruf von Exceptions im Falle einer leeren und einer mehrelementigen Resultatmenge.

Das Praktikum

Erstellen Sie für die Aufgaben ein Log-File inklusive der Fragen und Antworten als Kommentar bzw. der Ausgaben der SQL-Queries.

Nach dem Praktikum

Üben Sie Ihre Kenntnisse in PL/SQL an Variationen der Aufgaben dieses Praktikumblattes und auch an selbst ausgedachten Anfragen. Suchen Sie Ähnlichkeiten und Unterschiede von PL/SQL in Ihnen bekannten Programmiersprachen.

Anhang: Validierung einer IBAN

Gegeben sein folgende deutsche IBAN DE~~87~~648045234485030332.

Bei der IBAN steht ~~87~~ für die Prüfziffer. Zur Berechnung der Prüfziffern wird an den [Zahlenteil](#) der IBAN DE00 angehängen, wobei DE als 1314 codiert wird. Dies ergibt: [648045234485030332](#)131400. Dann wird die Zahl Modulo 97 gerechnet: $648045234485030332131400 \% 97 = 11$. Das Ergebnis wird von 98 abgezogen: $98-11=87$. Die IBAN ist gültig, wenn die berechnete Prüfziffer gleich der angegebenen Prüfziffer ist.