

Praktische Übung “DB-Änderungen mit SQL“

1. Führen Sie das Skript p1.sql aus, um die Tabelle MY_EMPLOYEE aufzubauen, die für nachfolgende Übungen gebraucht wird:

```
SET ECHO OFF
SET FEEDBACK OFF
PROMPT Creating the MY_EMPLOYEE table. Please wait ...
CREATE TABLE my_employee
(id          NUMBER(4) NOT NULL,
last_name   VARCHAR2(25),
first_name  VARCHAR2(25),
userid      VARCHAR2(8),
salary      NUMBER(9,2));
PROMPT Table MY_EMPLOYEE has been created
SET FEEDBACK ON
SET ECHO ON
```

2. Zeigen Sie sich die Struktur der Tabelle MY_EMPLOYEE an.
3. Fügen Sie den ersten Datensatz in die Tabelle MY_EMPLOYEE ein. Nutzen Sie die hierfür angegebenen Beispieldaten. Schreiben Sie den INSERT-Befehl ohne Bezugnahme auf Spaltennamen.

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	795
2	Dancs	Betty	bdancs	860
3	Biri	Ben	bbiri	1100
4	Newman	Chad	cnewman	750
5	Ropeburn	Audry	aropebur	1550

4. Schreiben Sie ein Skript p4.sql, um die Tabelle mit den restlichen Datensätzen zu füllen. Geben Sie dieses Mal die Spaltennamen in der INSERT-Anweisung explizit an. Prüfen Sie nach Ausführung, ob die Tabelle die Datensätze enthält.
5. Ändern Sie den Namen (LAST_NAME) des Angestellten 3 auf Drexler.
6. Erhöhen Sie das Gehalt auf 1000 bei all den Angestellten, die weniger als 900 verdienen. Überprüfen Sie, ob Ihre Änderung erfolgreich war.
7. Löschen Sie Betty Dancs aus der Tabelle.

8. Löschen Sie die gesamte Tabelle.
9. Legen Sie eine Tabelle DEPARTMENT an mit folgenden Spalten: ID (numerisch, 7 Stellen), Name (Zeichenkette, max. 25 Stellen). Schreiben Sie dafür ein SQL-Skript (p9.sql), das Sie anschließend ausführen. Definieren Sie den Primärschlüssel.
Hinweis: Oracle verwendet den Typ VARCHAR2 für variabel lange Zeichenketten.
10. Füllen Sie die Tabelle DEPARTMENT mit Daten aus der Tabelle DEPT. Verwenden Sie dabei nur jene Spalten, die Sie auch benötigen. Prüfen Sie, ob die Tabelle angelegt wurde.
11. Erzeugen Sie eine neue Tabelle EMPLOYEE, die auf der Struktur der EMP-Tabelle basiert, wobei nur die Spalten EMPNO, ENAME und DEPTNO übernommen werden sollen. Benennen Sie die Spalten in der neuen Tabelle ID, LAST_NAME und DEPT_ID. Definieren Sie Primär- und Fremdschlüssel, um abzusichern, daß kein Angestellter einer Abteilung zugeordnet werden kann, die nicht existiert. Weiterhin soll dafür gesorgt werden, daß beim Löschen einer Abteilung auch alle dort arbeitenden Angestellten entfernt werden (Spaltendefinition ergänzen um: ON DELETE CASCADE). Füllen Sie Ihre neue Tabelle mit Daten aus der EMP-Tabelle.
12. Erzeugen Sie ein Sequenz-Objekt, um es für die fortlaufende Numerierung der Abteilungsnummern in der Tabelle DEPARTMENT zu verwenden. Die Sequenz soll mit 60 beginnen bis zu einem Maximalwert von 200. Dabei soll jeweils um 10 erhöht werden. Nennen Sie das Sequenz-Objekt DEPT_ID_SEQ.
Verwenden Sie den Befehl `CREATE SEQUENCE name`
`START WITH start_wert`
`INCREMENT BY increment_wert`
`MAXVALUE end_wert ;`
13. Fügen Sie zwei neue Abteilungen ein, genannt "Education" und "Administration". Lassen Sie sich die Abteilungsnummer automatisch generieren, indem Sie die Pseudo-Spalte NEXTVAL Ihres Sequenz-Objekts verwenden:
`sequenz_objekt.NEXTVAL`
14. Fügen Sie zwei neue Angestellte ein, die in den beiden neuen Abteilungen arbeiten.
15. Testen Sie die Wirkungsweise Ihrer DELETE-Option bei der Definition des Fremdschlüssels in der EMPLOYEE-Tabelle, indem Sie die beiden neuen Abteilungen Education und Administration aus der Tabelle DEPARTMENT wieder löschen. Wie verändert sich der Inhalt der Tabelle EMPLOYEE?
16. Erzeugen Sie eine Sicht EMP_VU auf der Tabelle EMP, die Nummer und Name des Angestellten sowie die Nummer seiner Abteilung enthält. Benennen Sie dabei die Spalte ENAME in EMPLOYEE um.
17. Lassen Sie sich den Inhalt der Sicht ausgeben.
18. Lassen Sie sich die Namen aller Angestellten zusammen mit ihrer Abteilungsnummer über die Sicht EMP_VU ausgeben.