

Lehrveranstaltung " Datenbanksysteme II"

5. Aufgabenkomplex für das SQL-Praktikum – Oracle

Client

: SQL Developer

Arbeit unter Linux : Start des SQL Developers mit folgenden Befehlen:

```
cd /opt/sql/sqldeveloper/  
./sqldeveloper.sh
```

0 Kurze Beschreibung der Problemstellung

In einem Unternehmen werden Bauteile von unterschiedlichen Herstellern in verschiedene Fahrzeugtypen eingebaut.

Zu den Fahrzeugtypen werden die Bezeichnung, das Gewicht und Listenpreis in der Datenbank vermerkt. Der Listenpreis enthält sowohl den Netto-, als auch den Bruttopreis und erlaubt, dass der Preis mit einem Umrechnungsfaktor in beliebige andere Währungen umgerechnet werden kann.

Zum Bauteil sind der Teilename und die Einbauzeit zu speichern. Außerdem ist festzuhalten, von welchem Hersteller das entsprechende Bauteil bezogen wird. Bauteile können zu Baugruppen zugefasst werden. Dabei ist festgelegt, dass jedes Bauteil nur zu genau einer Baugruppe gehören darf, weshalb es ausreicht das entsprechende „Elternteil“ mit seiner Bauteil-Nummer zu notieren. Jedes Bauteil kann in mehreren Fahrzeugtypen verbaut werden.

Über den Hersteller werden der Name und die Adressinformationen (Strasse, Plz und Ort) gespeichert.

In ein Fahrzeug kann nur ein Bauteil verbaut werden, dass nicht zu einer Baugruppe gehört (und somit selbst eine Baugruppe darstellt).

Skizzieren Sie das Datenmodell:

05.01.2017

1 Anlegen der Relationen

1.1 Richten Sie folgende Tabellen in Ihrer Datenbank ein:

Fahrzeug

Spaltenname	Typ	Constraints
FzNr	number (5)	Primary Key
Bezeichnung	varchar2(15)	Not Null
Gewicht	float(10)	

Bauteil

Spaltenname	Typ	Constraints
BtNr	number (5)	Primary Key
Teilenamen	varchar2(20)	Not Null
Einbauzeit	number(5)	
HstNr	varchar2(10)	
Baugruppe	number (5)	Foreign Key (auf Bauteil(BtNr))

Einbau

Spaltenname	Typ	Constraints
EbNr	varchar2(10)	Primay Key
BtNr	number (5)	Not Null, Foreign Key
FzNr	number (5)	Not Null, Foreign Key mit Löschweitergabe
Anzahl	number (5)	

1.2 Stellen Sie sicher, dass die Kombination von **BtNr** und **FzNr** in der Tabelle **Einbau** eindeutig ist.

primärer constraint oder unique index

1.3 Legen Sie die Tabelle **Hersteller** an. Kopieren Sie hierzu die Tabellendefinition und die Daten aus der Tabelle **Hersteller** des Datenbankbenutzers **db01**.

1.4 Beim Kopieren der Tabelle werden nicht alle Constraints und Spalteneigenschaften kopiert. Ermitteln Sie die Eigenschaften der Tabelle **db01.Hersteller** und ergänzen Sie Ihre Tabelle **Hersteller** um diese. Zur Recherche nutzen Sie bitte folgende Abfragen. (Bitte das Abfrageergebnis in der vollständigen Breite betrachten!)

```
select * from all_constraints where owner = 'DB01' and table_name = 'HERSTELLER'
```

```
select * from ALL_TAB_COLUMNS where owner = 'DB01' and table_name = 'HERSTELLER'
```

```
select * from ALL_CONS_COLUMNS where owner = 'DB01' and table_name = 'HERSTELLER'
```

1.5 Fügen Sie folgenden Datensatz in die Tabelle **Hersteller** ein:

HstNr	HstName	Strasse	Plz	Ort	Kontaktaufnahme
693253	Tower Zwickau	Kopernikusstr. 60	08056	Zwickau	heute

- 1.6 Geben Sie Namen, PLZ und Ort Ihrer **Hersteller** aus, geordnet nach der PLZ.
- 1.7 Geben Sie alle **Hersteller** aus, zu denen in den letzten 5 Jahren der Kontakt aufgebaut wurde
- 1.8 Definieren Sie in der Tabelle **Bauteil** den Fremdschlüssel auf die Tabelle **Hersteller**.

2 Objekte, Methoden und Eigenschaften

Um den Listenpreis gemäß der Vorgaben zu definieren, wird ein Objekttyp mit entsprechenden Attributen und Methoden benötigt.

- 2.1 Erstellen Sie einen neuen Objekttyp **TPreis** mit folgenden Attributen und Methoden:

Attribut	Typ
Preis	Number(10,2)

Methode	Parameter	Rückgabewert
Netto	--	Number
Brutto	--	Number
Umrechnung	Faktor (number)	Number

- 2.2 Implementieren Sie den Funktionsrumpf:

Dabei soll gelten:

- Netto** – gibt den im Attribut Preis gespeicherten Wert zurück
- Brutto** – gibt den im Attribut Preis gespeicherten Wert zurück + 19% Mwst.
- Umrechn.** – multipliziert den Preis mit einem Faktor und gibt diesen Wert zurück

- 2.3 Ergänzen Sie die Tabelle **Fahrzeug** um die Spalte **Listenpreis** vom Typ **TPreis**.

3 Array Typen

- 3.1 Erstellen Sie einen Array-Typ **AnzTueren** der Größe 5, welcher Elemente vom Typ **char(10)** aufnimmt.

- 3.2 Ergänzen Sie die Tabelle **Fahrzeug** um eine neue Spalte **Tuerzahl** von Typ **AnzTueren**.

- 3.3 Fügen Sie folgende Daten in die Tabelle **Fahrzeug** ein:

FzNr	Bezeichnung	Gewicht	Listenpreis	Tuerzahl
10000	BMW Z4 Roadstar	900	60000	3-Türer
10001	VW Golf GTI	800	25000	3-Türer, 5-Türer
10002	Audi A3	850	30000	3-Türer

- 3.4 Ermitteln Sie die Bruttopreise Ihrer Fahrzeuge. Welcher Preis wäre zu zahlen, wenn der Umrechnungskurs 1:1,5 lauten würde?

4 Nested Table

Die Preise von Bauteilen bleiben nicht konstant. Um die Preisentwicklung über einen längeren Zeitraum verfolgen zu können, soll diese mit einer Nested Table in der Tabelle *Bauteil* festgehalten werden.

- 4.1 Erzeugen Sie einen Objekttyp namens **tPreisentwicklung** von Typ **object** mit folgenden Spalten:

Spaltenname	Typ
PeNr	varchar2(10)
Netto	number(10,2)
Datum	Date

- 4.2 Erzeugen Sie einen Objekttyp **ntPreisentwicklung** vom Typ **table**, der aus dem Typ **tPreisentwicklung** besteht.

- 4.3 Ergänzen Sie die Tabelle *Bauteil* um eine weitere Spalte **Preis** vom Typ **ntPreisentwicklung**.

- 4.4 Fügen Sie folgende Daten in die Tabelle *Bauteil* ein:

BtNr	Teilename	Einbauz	HstNr	Baugruppe	Preis
5000	Tuer links	20	134556	NULL	(7007, 900, 12.10.2015)
5001	Spiegel rechts	10	588797	NULL	(7008, 100, 12.10.2015)
5002	Auspuff	30	693253	NULL	(7009, 2000, 12.10.2015)

- 4.5 Führen Sie das SQL-Script **PRAKT/GRAEFE/bauteil.sql** aus, um Ihr Tabellen mit zusätzlichen Daten zu füllen.

Überprüfen Sie danach den Inhalt der Tabelle *Bauteil* (BtNr, Teilename, Einbau, HstNr, Baugruppe).

- 4.6 Geben Sie Baugruppen mit ihren zugehörigen Bauteilen in folgender Form aus:

```
<baugruppe>
  <bauteil>
    <untergeordnetes bauteil>
```

- 4.7 Geben Sie für jedes Bauteil seine Preisentwicklung aus (BtNr, Teilename, Netto, Datum).

- 4.8 Lesen Sie den Namen der ersten fünf Datensätze aus der Tabelle *Bauteil*, geordnet nach der BtNr, aus. Nutzen Sie hierzu ROWNUM.

- 4.9 Ermitteln Sie für jede Baugruppe das Bauteil mit der geringsten Einbauzeit. Nutzen Sie hierzu die Funktion RANK() OVER.

5 Objektview

5.1 Erstellen Sie einen Objekttyp **TAdresse** mit den Attributen Strasse, Plz und Ort analog der Definition der Tabelle **Hersteller**.

5.2 Erstellen Sie die Tabelle **Lieferant**:

Spaltenname	Typ	Bemerkung
LiefNr	number(6)	
Name	varchar2(20)	Primary Key
Adresse	TAdresse	Not Null

5.3 Erstellen Sie eine View (**Lieferant_OV**) über die Tabelle **Lieferant**. Lösen Sie hierbei die im Typ **TAdresse** enthaltenen Informationen in einzelne Spalten (Strasse, PLZ, Ort) auf.

5.4 Testen Sie, ob sich folgender Datensatz über die View (**Lieferant_OV**) in die Tabelle **Lieferant** einfügen lässt.

LiefNr	Name	Strasse	Plz	Ort
752443	Laepple Teublitz	Maxstr. 12	93158	Teublitz

6 Sicherung der semantischen Integrität

6.1 Nennen Sie Sicherungsmaßnahmen, die Sie bereits beim Erstellen der Tabellen vorgenommen haben.

6.2 Um die View **Lieferant_OV** zum Einfügen von Daten in die Tabelle **Lieferant** nutzen zu können, benötigen Sie einen Instead-Trigger. Erstellen Sie diesen. Testen Sie den Trigger indem Sie die Daten aus der Aufgabe 5.4 erneut eintragen.

6.3 Für die Tabelle **Einbau** soll der Primärschlüssel automatisch vergeben werden. Er soll sich aus einer fortlaufenden Nummer und dem vorangestellten Buchstaben „E“ bestehen (z.B. E2010). Hierzu wird eine Sequence angelegt, die eine fortlaufende Zahl erzeugt und mittels eines Triggers beim Hinzufügen von Datensätzen in die Tabelle übernommen.

- Legen Sie eine Sequence **sq_einbau** an, die einen Startwert von 2000 hat und sich jedes Mal um den Wert 1 erhöht.
- Erstellen Sie einen Trigger für die Tabelle **Einbau**, der vor dem Einfügen für jede Zeile den nächsten Wert aus der Sequence **sq_einbau** ermittelt, diesem Wert ein „E“ voransetzt und dem Feld **EbNr** zuweist. Bei Änderungen in der Tabelle soll der Trigger verhindern, dass der Primärschlüssel geändert werden kann.
- Testen Sie Ihren Trigger, indem Sie folgende Datensatz in die Tabelle einfügen und sich den Inhalt der Tabelle anzeigen lassen.

EbNr	BtNr	FzNr	Anzahl
NULL	5000	10000	1

6.4 Ändern Sie Ihren Trigger und stellen Sie sicher, dass nur solche Bauteile in die Tabelle **Einbau** enthalten sein können, bei denen es sich um vollständige Baugruppen handelt (Bauteil.Baugruppe ist NULL). Sollte das Einfügen nicht möglich sein, geben Sie eine Exception aus.

6.5 Testen Sie Ihren Trigger, indem Sie die folgenden Datensätzen in die Tabelle **Einbau** einfügen und sich den Inhalt der Tabelle anzeigen lassen.

FbNr	BtNr	FzNr	Anzahl
NULL	5001	10001	1
NULL	5010	10002	2

6.6 Löschen Sie den Datensatz mit der FzNr ,10001' aus der Tabelle **Fahrzeug**. Überprüfen Sie vor und nach dem Löschen die Inhalte der Tabellen **Fahrzeug** und **Einbau**!

7 Stored Procedures

Um die Ergebnisse der Prozeduren sehen zu können, müssen Sie für jede Session das Serveroutput einschalten.

```
set serveroutput on
```

7.1 Schreiben Sie eine Prozedur, die alle Fahrzeuge auflistet, die ein Gewicht besitzen, welches kleiner oder gleich einem bestimmten Wert ist. Dieser Wert soll der Prozedur als Parameter übergeben werden.

Die Ausgabe soll folgendes Aussehen haben:

```
FzNr          : 10000
Bezeichnung   : BMW Z4 Roadstar
Gewicht       : 900
-----
```

Hinweis: Nutzen Sie für die Lösung einen Cursor.

7.2 Kopieren Sie sich die Datei **einbau.sql** in das Verzeichnis C:/temp und führen Sie das SQL-Script aus. Es fügt der Tabelle **Einbau** weitere Datensätze hinzu.

Entwickeln Sie eine Prozedur, die Inhalt der Tabelle **Einbau** wie folgt ausliest:

- Cursor c1 liest die Bezeichnung der Fahrzeuge aus der Tabelle **Fahrzeug** für alle Fahrzeuge, die in der Tabelle **Einbau** referenziert werden.
- Cursor c2 liest die Teilennamen der Bauteile aus der Tabelle **Bauteil**, für alle Bauteile, die in ein bestimmtes Fahrzeug eingebaut wurden.

Die Ausgabe soll folgendes Aussehen haben:

```
Audi A3
.   Tuer links
.   Spiegel rechts
BMW Z4 Roadstar
.   Tuer links
.   Spiegel rechts
.   Auspuff
```


- 7.3 Erweitern Sie die Prozedur aus 7.3 um einen weiteren Cursor und geben Sie zusätzlich alle Bauteile, die zu den verbauten Baugruppen gehören, aus.
Die Ausgabe soll folgendes Aussehen haben:

```
Audi A3
. 5000 - Tuer links
. 5003 - Tuergriff links
. 5004 - Schlossriegel links
. 5005 - Türgummi
. 5001 - Spiegel rechts
. 5009 - Spiegelglas
. 5010 - Gehäuse
```

8 Zugriffsschutz

Aufgabe zu den Rechten: Lösen Sie die folgenden Aufgaben wechselseitig mit einem Partner in Form eines Rollenspiels.

- 8.1 Erteilen Sie Ihrem Partner die Leserechte für Ihre Tabelle **Hersteller**.
- 8.2 Ihr Partner soll versuchen, den Inhalt Ihrer Tabelle **Hersteller** zu lesen.
- 8.3 Fordern Sie Ihren Partner auf, einen Wert in **Hersteller** zu ändern.
- 8.4 Nehmen Sie von Ihrem Partner die Rechte an der Tabelle **Hersteller** wieder zurück.