

Abgabetermin: FR, 28.11.14 (15:00)

Name _____

☐ DES3UEG1☐ DES3UEG2

Punkte _____

Kurzzeichen Tutor _____

Trigger

Die folgenden Aufgaben beziehen sich auf Beispieltabellen, die mit einem vorgegebenen SQL-Skript gemeinsam mit einigen Datensätzen (Insert-Statements) zu erstellen sind (siehe letzte Übung). Kommentieren sie ihre Programme ausreichend und führen sie alle relevanten Überprüfungen durch. Testen sie ihre Trigger ausführlich.

Aufgabe 1: Business Rules

Erstellen Sie einen Trigger, der sicherstellt, dass die Tabelle PRODUCT nur zu Geschäftszeiten

- Montag – Freitag von
- 8.00-17.59 Uhr

verändert werden kann. Entsprechende Überprüfungen können folgendermaßen durchgeführt werden:

- `TO_CHAR(SYSDATE, 'DY') = 'MO'`
- `TO_CHAR(SYSDATE, 'HH24') = 14`

Außerhalb der Geschäftszeiten sollen entsprechende Statements fehlschlagen. Dazu können sie die vordefinierte Prozedur

```
raise_application_error(error_number, 'message')
```

aufrufen, die eine zwischen -20001 u. -20999 frei wählbare Fehlernummer sowie einen entsprechenden Fehlertext (z.B. „Zugriff am Wochenende verboten“, „Zugriff von 18.00 bis 8.00 verboten“) ausgibt und automatisch ein Rollback des triggernden Statements und etwaiger im Trigger durchgeführter Aktionen ausführt.

Aufgabe 2: Zugriffs-Protokollierung

Realisieren sie mittels Trigger die Überwachung der Tabelle EMP. In einer Tabelle AUDIT_TABLE_EMP sollen folgende Informationen protokolliert werden:

- Benutzer (Systemvariable USER)
- Zugriffszeitpunkt (Systemvariable SYSDATE)
- Zugriffsart
- bei INSERT die neue empno, bei UPDATE und DELETE die alte empno

Bei einem UPDATE auf eines der Attribute sal und/oder deptno sollen zusätzlich in einer weiteren Tabelle AUDIT_TABLE_EMP_VALUES die alten und neuen Werte dieses Attributs/dieser Attribute inklusive Bezeichnung gespeichert werden (AttributName, alterWert, neuerWert).

Weiters soll es möglich sein, die Tabellen `AUDIT_TABLE_EMP` und `AUDIT_TABLE_EMP_VALUES` miteinander zu verknüpfen. Die Werte für die Join-Attribute können dabei mit einer `SEQUENCE` automatisch generiert werden:

- 1) `CREATE SEQUENCE audit_seq;`
- 2) Generierung und Zugriff auf die nächst höhere Sequence-Nummer: `audit_seq.NEXTVAL`
- 3) Zugriff auf die aktuelle Sequence-Nummer: `audit_seq.CURRVAL`

Aufgabe 3: Überprüfung, ob Bestellung geändert werden darf

Stellen Sie sicher, dass eine Bestellung nicht mehr geändert werden kann, sobald sie ausgeliefert wurde.

- a) Erstellen Sie einen Trigger namens `ITEM_ORDER_SHIPPED`.

Der Trigger soll verhindern, dass in der Tabelle `ITEM` Datenmanipulationen durchgeführt werden, wenn die entsprechende Bestellung bereits ausgeliefert wurde (Spalte `SHIPDATE` der Tabelle `ORD` einen Wert aufweist).

- b) Erstellen Sie einen Trigger namens `ORD_ORDER_SHIPPED`.

Dieser Trigger soll verhindern, dass in der Tabelle `ORD` Datenmanipulationen durchgeführt werden, wenn die entsprechende Bestellung bereits ausgeliefert wurde.

Führen Sie Tests für alle möglichen Zugriffe durch, die durch die beiden Trigger verhindert werden sollen, wie z.B. die folgenden Zugriffe nach dem Setzen der Spalte `SHIPDATE`.

```
SELECT * FROM item WHERE ordid = 622;
SELECT * FROM ord  WHERE ordid = 622;

UPDATE item
  SET qty = 10
  WHERE ordid = 622 AND prodid = 100870;

UPDATE ord
  SET custid = 101
  WHERE ordid = 622;

UPDATE ord
  SET shipdate = SYSDATE
  WHERE ordid = 622;

UPDATE item
  SET qty = 100
  WHERE ordid = 622 AND prodid = 100870;

UPDATE ord
  SET custid = 102
  WHERE ordid = 622;
```

Organisatorische Hinweise

Die Lösungen sind von den Studierenden allein auszuarbeiten, Teamarbeit ist nicht erlaubt.

Die Lösungen sind zum Abgabetermin im E-Learning-System online abzugeben. Die Übungen werden von folgenden Tutoren betreut:

- | | | |
|-------------|-------------------|--------------------------------------|
| ▪ DES3UEG1: | Daniel Glaser | S1210307059@students.fh-hagenberg.at |
| ▪ DES3UEG2: | Melanie Mayrhofer | S1010307074@students.fh-hagenberg.at |