

Abgabetermin: 24.11.2021 (G3)

25.11.2021 (G1, G2)

<input type="checkbox"/> DES3UEG1: Niklas	Name	Angelos Angelis	Aufwand in h	5
<input checked="" type="checkbox"/> DES3UEG2: Niklas				
<input type="checkbox"/> DES3UEG3: Traxler	Punkte		Kurzzeichen Tutor	

Ziel dieser Übung ist die Vertiefung von PL/SQL in Paketen, dem Cursor-Konzept und Behandlung von Exceptions.

1. Datenbankpakete (PL/SQL-Packages)**(10 Punkte – 4+6 Pkt)**

Mit Hilfe von PL/SQL-Packages können Sie zusammengehörige PL/SQL-Typen, Variablen, Datenstrukturen, Exceptions und Unterprogramme in einer Bibliothek zusammenfassen. Packages bestehen normalerweise aus zwei Komponenten (Spezifikation und Body), die separat in der Datenbank gespeichert werden. Das Package selbst kann nicht aufgerufen, parametrisiert oder verschachtelt werden.

```
CREATE OR REPLACE PACKAGE top_actor_pkg AS
```

```
    ...  
END;  
/
```

```
CREATE OR REPLACE PACKAGE BODY top_actor_pkg AS
```

```
    ...  
END;  
/
```

1. Erstellen Sie ein Datenbankpaket top_actor_pkg mit der Funktion GetFilmcount und speichern Sie die Package Spezifikation und den Package Body gemeinsam ab.

Die Funktion GetFilmcount ermittelt für eine*n gegebenen Schauspieler*in der Sakila-Datenbank in einem bestimmten Zeitraum die Anzahl der Filme, in denen er/sie mitgespielt hat. Es werden nur Filme gezählt, die mindestens 60 Minuten lang sind. Erstellen Sie die Funktion mit drei Eingabeparameter:

- actor_id
- begin_year
- end_year

Führen Sie einen Testaufruf (zB für actor_id = 100) durch.

2. Implementieren Sie nun im Package top_actor_pkg eine zusätzliche Prozedur GetTopNActors, die drei Eingabeparameter verlangt:

- die Anzahl der Schauspieler*innen n_count, um eine Top-N Liste der n_count fleißigsten Schauspieler*innen ausgeben zu können
- begin_year
- end_year

D.h. es sollen die Anzahl (n_count) Schauspieler*innen ausgegeben werden, die die meisten Filme zwischen begin_date und end_date bespielt haben (bei Ties/Gleichstand dürfen mehr als n Schauspieler ausgegeben werden). Wird keine Anzahl an Schauspieler angegeben, so sollen

default-mäßig die Top-20 Schauspieler*innen ausgegeben werden. Geben Sie auch für die anderen Parameter Default-Werte an, um einen Aufruf ohne Parameter zu ermöglichen.

Verwenden Sie für die zu erstellende Prozedur eine CURSOR FOR LOOP, die Prozedur DBMS_OUTPUT.PUT_LINE und verwenden Sie die bereits vorhandenen Funktion. Sortieren Sie nach der Anzahl absteigend. Sie können hier FETCH FIRST ... WITH TIES verwenden. Geben Sie die Package Spezifikation, den Package Rumpf sowie den Testaufruf (mit und ohne Parameter) an.

Beispielausgabe

```
...
The top 20 actors from 2000 to 2010 are:
WALTER TORN: 16 films
JADA RYDER: 16 films
...
```

2. Cursor mit FOR-UPDATE

(10 Punkte – 1.-2. je 3 Pkt und 3.-4. je 2 Pkt)

Wenn mehrere Sessions für eine einzelne Datenbank vorhanden sind, besteht die Möglichkeit, dass die Zeilen einer bestimmten Tabelle aktualisiert wurden, nachdem Sie den Cursor geöffnet haben. Sie sehen die aktualisierten Daten nur, wenn Sie den Cursor erneut öffnen. Es ist daher günstiger, die Zeilen zu sperren, bevor Sie Zeilen aktualisieren oder löschen. Sie können zum Sperren der Zeilen die FOR-UPDATE-Klausel in der Cursor-Abfrage verwenden.

1. Erstellen Sie eine Tabelle top_actor, die actor_id und die Anzahl der Filme enthält. Speichern Sie in der Tabelle auch den User und das Datum an dem der jeweilige Datensatz eingefügt wurde. Eine weitere Spalte enthält das Datum an dem der Datensatz ‚deaktiviert‘ wurde (dh. die Person nicht mehr zu den Top Schauspieler*innen gehört), für aktive Datensätze ist dieser Eintrag NULL.
Erweitern Sie Ihre Prozedur GetTopNActors dahingehend, dass zusätzlich zur Ausgabe mit DBMS_OUTPUT auch Datensätze in die Tabelle top_actor eingefügt werden.
2. Fügen Sie dem Datenbankpaket top_actor_pkg eine weitere Prozedur DeactivateTopActors hinzu, die Datensätze in der Tabelle top_actor mit weniger als einer angegebenen Anzahl an Filmen deaktiviert (das aktuelle Datum einträgt). Testen Sie Ihre Prozedur.

Hinweise: Verwenden Sie beim Cursor die FOR-UPDATE und beim UPDATE die WHERE CURRENT OF-Klausel. Geben Sie die Package Spezifikation, den Package Body sowie den Testaufruf an.

3. Öffnen Sie eine zweite Datenbank-Session (Strg+Shift+N) und führen Sie die Prozedur DeactivateTopActors mit den gleichen Parametern in jeder Session aus, ohne ein COMMIT auszuführen. Wählen Sie die Parameter so, damit zumindest ein Satz aus top_actor selektiert wird. Was passiert in der zweiten Session? Führen Sie in der ersten ein COMMIT aus und beschreiben Sie die Auswirkungen.
4. Erweitern Sie nun den Cursor in der Prozedur DeactivateTopActors um die NOWAIT-Klausel. Wiederholen Sie den Test und erläutern Sie den Unterschied. Geben Sie die Package Spezifikation, den Package Rumpf sowie den Testaufruf an.

3. EXCEPTIONS

(4 Punkte – 1. 2 Pkt und 2.-3. je 1 Pkt)

Diese Aufgabe behandelt die Verwendung von vordefinierten Exceptions.

1. Erstellen Sie eine PL/SQL-Prozedur mit dem Parameter `name_part` (vom Typ `customer.last_name`), die anhand eines angegebenen Namens(-teils) jene Kunden auswählt, deren Namen damit beginnen. Hinweis: Verwenden Sie keinen expliziten Cursor und vergleichen Sie mit `LIKE`. Wenn der an die Prozedur übergebene Namensteil nur eine Zeile zurückgibt, fügen Sie in die Tabelle `messages` den Namen des Kunden (Vor- und Nachname und `customer_id`) ein. Die Tabelle `messages` soll aus einer Spalte `results` vom Typ `VARCHAR2(100)` bestehen. Testen Sie die Prozedur anhand folgender Fälle: a) Keine Zeile wird selektiert. b) Eine Zeile wird selektiert. c) Mehrere Zeilen werden selektiert. Was stellen Sie fest?
2. Wenn der eingegebene Name keine Zeilen zurückgibt, behandeln Sie die Exception mit einem entsprechenden Exception Handler, und fügen Sie in die Tabelle `messages` die Meldung „No customer found where last name begins with <name>“ ein.
3. Behandeln Sie beliebige weitere Exceptions mit einem entsprechenden Exception Handler, und fügen Sie in die Tabelle `messages` die Meldung „An undefined error occurred“ ein.

Aufgabe 1)

```

CREATE OR REPLACE PACKAGE top_actor_pkg AS
    FUNCTION GetFilmCount(actor_IDv ACTOR.ACTOR_ID%TYPE, begin_year
NUMBER, end_year NUMBER) RETURN NUMBER;
    PROCEDURE GetTopNActors(n_count NUMBER DEFAULT 20, begin_year NUMBER
DEFAULT 1900, end_year NUMBER DEFAULT 2000);
END;
/

-- TODO Definition des Paktet-Rumpfes/Implementierung
CREATE OR REPLACE PACKAGE BODY top_actor_pkg AS
    --TODO Funktion hineinkopieren
    FUNCTION GetFilmCount(actor_IDv ACTOR.ACTOR_ID%TYPE, begin_year
NUMBER, end_year NUMBER)
        RETURN NUMBER
    IS
        cnt NUMBER;
    BEGIN
        SELECT COUNT(*) INTO cnt
        FROM ACTOR
        JOIN FILM_ACTOR USING (ACTOR_ID)
        JOIN FILM USING (FILM_ID)
        WHERE ACTOR_ID = actor_IDv
        AND RELEASE_YEAR BETWEEN begin_year AND end_year
        AND LENGTH > 60;
        RETURN cnt;
    END;

    PROCEDURE GetTopNActors(n_count NUMBER DEFAULT 20, begin_year NUMBER
DEFAULT 1900, end_year NUMBER DEFAULT 2000)
    IS
        CURSOR act_cur IS
            SELECT First_Name, Last_name, Actor_id, COUNT(*) AS CNT
            FROM ACTOR
            JOIN FILM_ACTOR USING (ACTOR_ID)
            JOIN FILM USING (FILM_ID)
            WHERE RELEASE_YEAR BETWEEN begin_year AND end_year
            AND LENGTH > 60
            GROUP BY First_Name, Last_name, Actor_id
            ORDER BY cnt DESC
            fetch FIRST n_count rows with ties;
    BEGIN
        DBMS_OUTPUT.PUT_LINE('The top 20 actors from '||begin_year||' TO
'||end_year ||' are:');
        FOR actr IN act_cur
        LOOP
            DBMS_OUTPUT.PUT_LINE(actr.First_Name||' '||actr.Last_name||':
'||actr.cnt||' films');
        END LOOP;
    END;
END;
/

```

```
SELECT top actor pkg.GetFilmCount(100,2000,2010) FROM dual;
```

TOP_ACTOR_PKG.GETFILMCOUNT(100,2000,2010) ÷	
1	11

```
Call top actor pkg.GetTopNActors();
```

```
[2021-11-25 11:21:47] completed in 35 ms
[2021-11-25 11:21:47] The top 20 actors from 1900 TO 2000 are:
[2021-11-25 11:21:47] MARY KEITEL: 27 films
[2021-11-25 11:21:47] SANDRA KILMER: 26 films
[2021-11-25 11:21:47] DARYL WAHLBERG: 26 films
[2021-11-25 11:21:47] MATTHEW CARREY: 26 films
[2021-11-25 11:21:47] HENRY BERRY: 25 films
[2021-11-25 11:21:47] WALTER TORN: 25 films
[2021-11-25 11:21:47] JULIANNE DENCH: 24 films
[2021-11-25 11:21:47] KENNETH TORN: 24 films
[2021-11-25 11:21:47] SCARLETT DAMON: 24 films
[2021-11-25 11:21:47] UMA WOOD: 24 films
[2021-11-25 11:21:47] KIRSTEN AKROYD: 24 films
[2021-11-25 11:21:47] WARREN NOLTE: 24 films
[2021-11-25 11:21:47] SUSAN DAVIS: 23 films
[2021-11-25 11:21:47] HUMPHREY GARLAND: 23 films
[2021-11-25 11:21:47] VAL BOLGER: 23 films
[2021-11-25 11:21:47] RIVER DEAN: 23 films
[2021-11-25 11:21:47] JAYNE NOLTE: 22 films
[2021-11-25 11:21:47] MICHAEL BOLGER: 22 films
[2021-11-25 11:21:47] ED MANSFIELD: 22 films
[2021-11-25 11:21:47] RENEE TRACY: 22 films
[2021-11-25 11:21:47] WILL WILSON: 22 films
[2021-11-25 11:21:47] DAN HARRIS: 22 films
```

Aufgabe 2)

```

CREATE TABLE top_actors
(
  actor_id NUMBER(5) PRIMARY KEY,
  film_count NUMBER,
  createdBy VARCHAR2(30) DEFAULT USER,
  dateCreated TIMESTAMP DEFAULT SYSDATE,
  dateDeactivated TIMESTAMP DEFAULT NULL
);

CREATE OR REPLACE PACKAGE top_actor_pkg AS
  FUNCTION GetFilmCount(actor_ID ACTOR.ACTOR_ID%TYPE, begin_year
NUMBER, end_year NUMBER) RETURN NUMBER;
  PROCEDURE GetTopNActors(n_count NUMBER DEFAULT 20, begin_year NUMBER
DEFAULT 1900, end_year NUMBER DEFAULT 2000);
  PROCEDURE DeactivateTopActors(films NUMBER);
END;
/

-- TODO Definition des Paket-Rumpfes/Implementierung
CREATE OR REPLACE PACKAGE BODY top_actor_pkg AS
  --TODO Funktion hineinkopieren
  FUNCTION GetFilmCount(actor_ID ACTOR.ACTOR_ID%TYPE, begin_year
NUMBER, end_year NUMBER)
  RETURN NUMBER
  IS
    cnt NUMBER;
  BEGIN
    SELECT COUNT(*) INTO cnt
    FROM ACTOR
    JOIN FILM_ACTOR USING (ACTOR_ID)
    JOIN FILM USING (FILM_ID)
    WHERE actor_id = 100
    AND RELEASE_YEAR BETWEEN begin_year AND end_year
    AND LENGTH > 60;
    RETURN cnt;
  END;

  PROCEDURE GetTopNActors(n_count NUMBER DEFAULT 20, begin_year NUMBER
DEFAULT 1900, end_year NUMBER DEFAULT 2000)
  IS
    CURSOR act_cur IS
      SELECT First_Name, Last_name, Actor_id, COUNT(*) AS CNT
      FROM ACTOR
      JOIN FILM_ACTOR USING (ACTOR_ID)
      JOIN FILM USING (FILM_ID)
      WHERE RELEASE_YEAR BETWEEN begin_year AND end_year
      AND LENGTH > 60
      GROUP BY First_Name, Last_name, Actor_id
      ORDER BY cnt DESC
      fetch FIRST n_count rows with ties;
  BEGIN
    DBMS_OUTPUT.PUT_LINE('The top 20 actors from ' || begin_year || ' TO
' || end_year || ' are:');
    FOR actr IN act_cur
    LOOP

```

```

        DBMS_OUTPUT.PUT_LINE(actr.First_Name||' '||actr.Last_name||':
'||actr.cnt||' films');
        INSERT INTO top_actors(actor_id,film_count)
        VALUES (actr.actor_ID,actr.cnt);
    END LOOP;
END;

PROCEDURE DeactivateTopActors(films NUMBER)
IS
    CURSOR act_cur IS
        SELECT * FROM top_actors
        FOR UPDATE NOWAIT;
BEGIN
    FOR actr IN act_cur
    LOOP
        IF (actr.film_count < films) THEN
            UPDATE top_actors
            SET datedeactivated = SYSDATE
            WHERE CURRENT OF act_cur;
        END IF;
    END LOOP;
END;
/

drop table TOP_ACTORS;
ROLLBACK;
Call top_actor_pkg.GetTopNActors();
Call top_actor_pkg.deactivatetopactors(24);
COMMIT;
SELECT* FROM top_actors

```

	ACTOR_ID	FILM_COUNT	CREATEDBY	DATECREATED	DATEDEACTIVATED
1	198	27	S2010307048	2021-11-25 11:29:07.000000	<null>
2	23	26	S2010307048	2021-11-25 11:29:07.000000	<null>
3	95	26	S2010307048	2021-11-25 11:29:07.000000	<null>
4	181	26	S2010307048	2021-11-25 11:29:07.000000	<null>
5	60	25	S2010307048	2021-11-25 11:29:07.000000	<null>
6	102	25	S2010307048	2021-11-25 11:29:07.000000	<null>
7	123	24	S2010307048	2021-11-25 11:29:07.000000	<null>
8	94	24	S2010307048	2021-11-25 11:29:07.000000	<null>
9	81	24	S2010307048	2021-11-25 11:29:07.000000	<null>
10	13	24	S2010307048	2021-11-25 11:29:07.000000	<null>
11	92	24	S2010307048	2021-11-25 11:29:07.000000	<null>
12	108	24	S2010307048	2021-11-25 11:29:07.000000	<null>
13	101	23	S2010307048	2021-11-25 11:29:07.000000	<null>
14	184	23	S2010307048	2021-11-25 11:29:07.000000	<null>
15	37	23	S2010307048	2021-11-25 11:29:07.000000	<null>
16	143	23	S2010307048	2021-11-25 11:29:07.000000	<null>
17	150	22	S2010307048	2021-11-25 11:29:07.000000	<null>
18	185	22	S2010307048	2021-11-25 11:29:07.000000	<null>
19	136	22	S2010307048	2021-11-25 11:29:07.000000	<null>
20	117	22	S2010307048	2021-11-25 11:29:07.000000	<null>
21	168	22	S2010307048	2021-11-25 11:29:07.000000	<null>
22	56	22	S2010307048	2021-11-25 11:29:07.000000	<null>

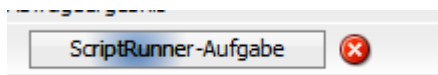
	⚙️ ACTOR_ID ⚙️	📊 FILM_COUNT 📊	👤 CREATEDBY 👤	📅 DATECREATED 📅	📅 DATEDEACTIVATED 📅
1	198	27	S2010307048	2021-11-25 11:29:07.000000	<null>
2	23	26	S2010307048	2021-11-25 11:29:07.000000	<null>
3	95	26	S2010307048	2021-11-25 11:29:07.000000	<null>
4	181	26	S2010307048	2021-11-25 11:29:07.000000	<null>
5	60	25	S2010307048	2021-11-25 11:29:07.000000	<null>
6	102	25	S2010307048	2021-11-25 11:29:07.000000	<null>
7	123	24	S2010307048	2021-11-25 11:29:07.000000	<null>
8	94	24	S2010307048	2021-11-25 11:29:07.000000	<null>
9	81	24	S2010307048	2021-11-25 11:29:07.000000	<null>
10	13	24	S2010307048	2021-11-25 11:29:07.000000	<null>
11	92	24	S2010307048	2021-11-25 11:29:07.000000	<null>
12	108	24	S2010307048	2021-11-25 11:29:07.000000	<null>
13	101	23	S2010307048	2021-11-25 11:29:07.000000	2021-11-25 11:29:46.000000
14	184	23	S2010307048	2021-11-25 11:29:07.000000	2021-11-25 11:29:46.000000
15	37	23	S2010307048	2021-11-25 11:29:07.000000	2021-11-25 11:29:46.000000
16	143	23	S2010307048	2021-11-25 11:29:07.000000	2021-11-25 11:29:46.000000
17	150	22	S2010307048	2021-11-25 11:29:07.000000	2021-11-25 11:29:46.000000
18	185	22	S2010307048	2021-11-25 11:29:07.000000	2021-11-25 11:29:46.000000
19	136	22	S2010307048	2021-11-25 11:29:07.000000	2021-11-25 11:29:46.000000
20	117	22	S2010307048	2021-11-25 11:29:07.000000	2021-11-25 11:29:46.000000
21	168	22	S2010307048	2021-11-25 11:29:07.000000	2021-11-25 11:29:46.000000
22	56	22	S2010307048	2021-11-25 11:29:07.000000	2021-11-25 11:29:46.000000

2.3)

⚙️ ACTOR_ID ⚙️	📊 FILM_COUNT 📊	👤 CREATEDBY 👤	📅 DATECREATED 📅	📅 DATEDEACTIVATED 📅
1	198	27 S2010307048	25.11.21 11:44:59,000000000	(null)
2	23	26 S2010307048	25.11.21 11:44:59,000000000	(null)
3	95	26 S2010307048	25.11.21 11:44:59,000000000	(null)
4	181	26 S2010307048	25.11.21 11:44:59,000000000	(null)
5	60	25 S2010307048	25.11.21 11:44:59,000000000	25.11.21 11:45:53,000000000
6	102	25 S2010307048	25.11.21 11:44:59,000000000	25.11.21 11:45:53,000000000
7	123	24 S2010307048	25.11.21 11:44:59,000000000	25.11.21 11:45:53,000000000
8	94	24 S2010307048	25.11.21 11:44:59,000000000	25.11.21 11:45:53,000000000
9	81	24 S2010307048	25.11.21 11:44:59,000000000	25.11.21 11:45:53,000000000
10	13	24 S2010307048	25.11.21 11:44:59,000000000	25.11.21 11:45:53,000000000
11	92	24 S2010307048	25.11.21 11:44:59,000000000	25.11.21 11:45:53,000000000
12	108	24 S2010307048	25.11.21 11:44:59,000000000	25.11.21 11:45:53,000000000
13	101	23 S2010307048	25.11.21 11:44:59,000000000	25.11.21 11:45:53,000000000
14	184	23 S2010307048	25.11.21 11:44:59,000000000	25.11.21 11:45:53,000000000
15	37	23 S2010307048	25.11.21 11:44:59,000000000	25.11.21 11:45:53,000000000
16	143	23 S2010307048	25.11.21 11:44:59,000000000	25.11.21 11:45:53,000000000
17	150	22 S2010307048	25.11.21 11:44:59,000000000	25.11.21 11:45:53,000000000
18	185	22 S2010307048	25.11.21 11:44:59,000000000	25.11.21 11:45:53,000000000
19	136	22 S2010307048	25.11.21 11:44:59,000000000	25.11.21 11:45:53,000000000
20	117	22 S2010307048	25.11.21 11:44:59,000000000	25.11.21 11:45:53,000000000
21	168	22 S2010307048	25.11.21 11:44:59,000000000	25.11.21 11:45:53,000000000

Ergebnis in Session 1

In Session zwei funktioniert das dann nicht. Es lädt dann unendlich lang.



Mit dem NOWAIT passiert folgendes in Session zwei

```

Fehler beim Start in Zeile: 4 in Befehl -
Call top_actor_pkg.deactivatetopactors(26)
Fehlerbericht -
ORA-00054: Ressource belegt und Anforderung mit NOWAIT angegeben oder Timeout abgelaufen
ORA-06512: in "S2010307048.TOP_ACTOR_PKG", Zeile 43
ORA-06512: in "S2010307048.TOP_ACTOR_PKG", Zeile 46
ORA-06512: in Zeile 1
00054. 00000 - "resource busy and acquire with NOWAIT specified or timeout expired"
*Cause: Interested resource is busy.
*Action: Retry if necessary or increase timeout.

```