

Zadania do realizacji

1. Dana jest tabela ciastko(id_ciastrko,nazwa,cena).
Zdefiniuj wyzwalacz, który dla każdego dodawanego ciastka będzie automatycznie przydzielał kolejny wolny identyfikator id_ciastrko. Przykładowo instrukcja:
`INSERT INTO ciastko(nazwa,cena) VALUES ('Beza',3.15);`
powinna poprawnie dodać ciastko, a kolumna id_ciastrko nie może mieć wartości NULL.
Przetestuj działanie utworzonego wyzwalacza.
2. Dana jest tabela dane(id_dane,imie,nazwisko,pesel,data_ur,plec). Po wykonaniu instrukcji:
`INSERT INTO dane(imie,nazwisko,pesel) VALUES ('Jan','Kowalski','75052503553');`
a) Wyzwalacz powinien automatycznie zdefiniować kolejny numer id_dane.
b) Wyzwalacz powinien z numeru pesel wyciągnąć datę urodzenia i wpisać ją do kolumny data_ur.
c) Wyzwalacz powinien z numeru pesel odczytać płeć i wpisać wartości 'K' lub 'M' do kolumny plec.
Przetestuj działanie utworzonego wyzwalacza.
3. Dana jest tabela:
`CREATE TABLE lista(id_slowo INT, slowo VARCHAR2(20));`
i przykładowe rekordy:
`INSERT INTO lista(id_slowo,slowo) VALUES(1,'red');`
`INSERT INTO lista(id_slowo,slowo) VALUES(2,'blue');`
`INSERT INTO lista(id_slowo,slowo) VALUES(3,'green');`
Napisać wyzwalacz, który uniemożliwi usuwanie i modyfikację rekordów w tabeli lista.
Przetestuj działanie utworzonego wyzwalacza.
4. Dane są tabele imiona(id,imie) oraz imiona_historia(id,imie,data_zmiany,zmiana).
Napisać wyzwalacz, który będzie archiwizował modyfikowane i usuwane rekordy z tabeli imiona do tabeli imiona_historia. Dodatkowo w tabeli imiona_historia powinna być zapisana data zmiany i rodzaj zmiany: 'U' (od update), 'D' (od delete). Przetestuj działanie utworzonego wyzwalacza.

Rozwiązania

1. Dana jest tabela ciastko(id_ciastrko,nazwa,cena).

Zdefiniuj wyzwalacz, który dla każdego dodawanego ciastka będzie automatycznie przydzielał kolejny wolny identyfikator id_ciastrko. Przykładowo instrukcja:

INSERT INTO ciastko(nazwa,cena) VALUES ('Beza',3.15);

powinna poprawnie dodać ciastko, a kolumna id_ciastrko nie może mieć wartości NULL.

Przetestuj działanie utworzonego wyzwalacza.

```

1  --DROP TABLE ciastko;
2  CREATE TABLE ciastko(
3      id_ciastrko INT,
4      nazwa VARCHAR2(20),
5      cena NUMBER(10,2)
6  );
7  /
8  --DROP SEQUENCE ciastko_seq_id;
9  CREATE SEQUENCE ciastko_seq_id START WITH 1 INCREMENT BY 1 NOMAXVALUE
10 /
11 CREATE OR REPLACE TRIGGER dodaj_ciastrko BEFORE INSERT ON ciastko
12 FOR EACH ROW
13 BEGIN
14     SELECT ciastko_seq_id.NEXTVAL INTO :NEW.id_ciastrko FROM dual;
15 END;
16 /
17 SELECT * FROM ciastko;
18 INSERT INTO ciastko(nazwa,cena) VALUES ('Beza',3.15);
19 SELECT * FROM ciastko;
20 INSERT INTO ciastko(nazwa,cena) VALUES ('Pączek',2.25);
21 SELECT * FROM ciastko;
22 /

```

ID_CIASTRKO	NAZWA	CENA
-------------	-------	------

ID_CIASTRKO	NAZWA	CENA
1	1 Beza	3,15

ID_CIASTRKO	NAZWA	CENA
1	1 Beza	3,15
2	2 Pączek	2,25

2. Dana jest tabela dane(id_dane,imie,nazwisko,pesel,data_ur,plec). Po wykonaniu instrukcji: INSERT INTO dane(imie,nazwisko,pesel) VALUES ('Jan','Kowalski','75052503553');
- a) Wyzwalacz powinien automatycznie zdefiniować kolejny numer id_dane.
 - b) Wyzwalacz powinien z numeru pesel wyciągnąć datę urodzenia i wpisać ją do kolumny data_ur.
 - c) Wyzwalacz powinien z numeru pesel odczytać płeć i wpisać wartości 'K' lub 'M' do kolumny plec.
- Przetestuj działanie utworzonego wyzwalacza.

```
1  --DROP TABLE dane;
2  CREATE TABLE dane(
3      id_dane INT,
4      imie VARCHAR2(20),
5      nazwisko VARCHAR2(30),
6      pesel CHAR(11),
7      data_ur DATE,
8      plec CHAR(1)
9  );
10 /
11 --DROP SEQUENCE dane_seq_id;
12 CREATE SEQUENCE dane_seq_id START WITH 1 INCREMENT BY 1 NOMAXVALUE;
13 /

14 CREATE OR REPLACE TRIGGER dodaj_dane BEFORE INSERT ON dane
15 FOR EACH ROW
16 DECLARE
17     v_data VARCHAR(10);
18     v_plec CHAR(1);
19 BEGIN
20     v_data:=SUBSTR(:new.pesel,1,2)|| '-' || SUBSTR(:new.pesel,3,2)||
21         '-' || SUBSTR(:new.pesel,5,2);
22     IF TO_NUMBER(SUBSTR(:new.pesel,10,1)) IN (0,2,4,6,8)
23     THEN v_plec := 'K';
24     ELSE v_plec := 'M';
25     END IF;
26     SELECT dane_seq_id.NEXTVAL INTO :new.id_dane FROM dual;
27     SELECT to_date(v_data,'yy-mm-dd') INTO :new.data_ur FROM dual;
28     SELECT v_plec INTO :new.plec FROM dual;
29 END;
30 /
```

```
31 | SET SERVEROUTPUT ON
32 | SELECT * FROM dane;
33 | INSERT INTO dane(imie,nazwisko,pesel)
34 | VALUES ('Jan','Kowalski','75052503553');
35 | SELECT * FROM dane;
36 | /
```

ID_DANE	IMIE	NAZWISKO	PESEL	DATA_UR	PLEC
---------	------	----------	-------	---------	------

ID_DANE	IMIE	NAZWISKO	PESEL	DATA_UR	PLEC
1	Jan	Kowalski	75052503553	75/05/25	M

3. Dana jest tabela:

```
CREATE TABLE lista(id_slowo INT, slowo VARCHAR2(20));
```

i przykładowe rekordy:

```
INSERT INTO lista(id_slowo,slowo) VALUES(1,'red');
```

```
INSERT INTO lista(id_slowo,slowo) VALUES(2,'blue');
```

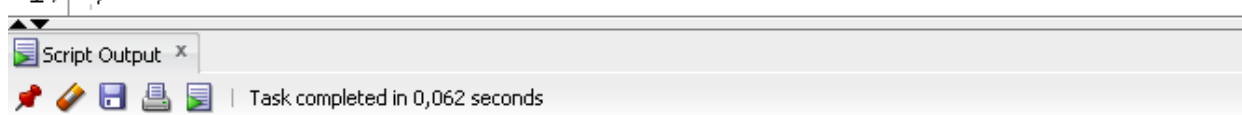
```
INSERT INTO lista(id_slowo,slowo) VALUES(3,'green');
```

Napisać wyzwalacz, który uniemożliwi usuwanie i modyfikację rekordów w tabeli lista.

Przetestuj działanie utworzonego wyzwalacza.

```
1  --DROP TABLE lista;
2  CREATE TABLE lista(id_slowo INT,slowo VARCHAR2(20));
3  /
4  INSERT INTO lista(id_slowo,slowo) VALUES(1,'red');
5  INSERT INTO lista(id_slowo,slowo) VALUES(2,'blue');
6  INSERT INTO lista(id_slowo,slowo) VALUES(3,'green');
7  /
8  CREATE OR REPLACE TRIGGER zabron_delete
9  BEFORE DELETE OR UPDATE ON lista
10 BEGIN
11     IF DELETING THEN RAISE_APPLICATION_ERROR(-20501,'Zakaz usuwania!');
12     ELSE RAISE_APPLICATION_ERROR(-20501,'Zakaz modyfikowania!');
13     END IF;
14 END;
15 /

16 DELETE FROM lista;
17 /
```



Error starting at line : 16 in command -

DELETE FROM lista

Error report -

SQL Error: ORA-20501: Zakaz usuwania!

ORA-06512: at "ROBERT.ZABRON_DELETE", line 2

ORA-04088: error during execution of trigger 'ROBERT.ZABRON_DELETE'

```
18 UPDATE lista SET slowo='black' WHERE id_slowo=1;  
19 /
```

Script Output x

Task completed in 0,015 seconds

Error starting at line : 18 in command -
UPDATE lista SET slowo='black' WHERE id_slowo=1
Error report -
SQL Error: ORA-20501: Zakaz modyfikowania!
ORA-06512: at "ROBERT.ZABRON_DELETE", line 3
ORA-04088: error during execution of trigger 'ROBERT.ZABRON_DELETE'

```
20 SELECT * FROM lista;  
21 /
```

Script Output x Query Result x

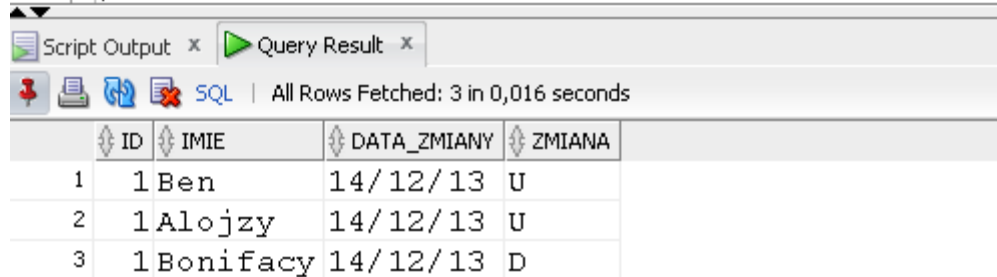
All Rows Fetched: 3 in 0 seconds

ID_SLOWO	SLOWO
1	red
2	blue
3	green

4. Dane są tabele imiona(id,imie) oraz imiona_historia(id,imie,data_zmiany,zmiana). Napisać wyzwalacz, który będzie archiwizował modyfikowane i usuwane rekordy z tabeli imiona do tabeli imiona_historia. Dodatkowo w tabeli imiona_historia powinna być zapisana data zmiany i rodzaj zmiany: 'U' (od update), 'D' (od delete). Przetestuj działanie utworzonego wyzwalacza.

```
1 CREATE TABLE imiona(  
2     id INT,  
3     imie VARCHAR2(20)  
4 );  
5 CREATE TABLE imiona_historia(  
6     id INT,  
7     imie VARCHAR2(20),  
8     data_zmiany DATE,  
9     zmiana CHAR(1)  
10 );  
11 /  
12 INSERT INTO imiona(id,imie) VALUES(1,'Ben');  
13 INSERT INTO imiona(id,imie) VALUES(2,'Ola');  
14 INSERT INTO imiona(id,imie) VALUES(3,'Ala');  
15 INSERT INTO imiona(id,imie) VALUES(4,'Ewa');  
16 /  
  
17 CREATE OR REPLACE TRIGGER imiona_historia_zapisz  
18 BEFORE DELETE OR UPDATE ON imiona FOR EACH ROW  
19 BEGIN  
20     IF DELETING THEN  
21         INSERT INTO imiona_historia(id,imie,data_zmiany,zmiana)  
22         VALUES (:old.id,:old.imie,sysdate,'D');  
23     ELSE  
24         INSERT INTO imiona_historia(id,imie,data_zmiany,zmiana)  
25         VALUES (:old.id,:old.imie,sysdate,'U');  
26     END IF;  
27 END;  
28 /
```

```
29 UPDATE imiona SET imie='Alojzy' WHERE id=1;
30 UPDATE imiona SET imie='Bonifacy' WHERE id=1;
31 DELETE FROM imiona WHERE id=1;
32 /
33 SELECT * FROM imiona_historia;
34 /
```



ID	IMIE	DATA_ZMIANY	ZMIANA
1	1 Ben	14/12/13	U
2	1 Alojzy	14/12/13	U
3	1 Bonifacy	14/12/13	D