Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный университет телекоммуникаций и информатики» (СибГУТИ)

Кафедра ПМиК

Расчетно-графическое задание

по дисциплине

«Сетевые базы данных»

Вариант 7

Выполнил:
студент 4 курса ИВТ,
гр. ИП-013
Копытина Т.А.

Проверил:
Старший преподаватель
кафедры ПМиК
Дьячкова И.С.

Новосибирск, 2024 г.

Содержание

Задание

1. Создать две таблицы, каждая из которых должна иметь первичный ключ и, по крайней мере, один столбец с ограничением NOT NULL. Таблицы должны быть связаны внешним ключом; тип связи - "один-ко-многим". *Таблицы должны содержать данные о Морях и впадающих в них Реках. В любое море может впадать несколько рек.*
2. Создать пакет, содержащий процедуру начального заполнения таблиц данными (по 7-10 записей в таблице) и процедуру очистки таблиц (удаления записей).
3. Для одной из таблиц разработать триггер для обеспечения дополнительных ограничений на изменение данных таблицы. *Триггер должен регистрировать изменение с указанием названия реки, прежнего и нового ее размера и времени изменения.*
4. Создать представление, которое позволяет запрашивать данные из обеих (связанных) таблиц. Представление должно ограничивать доступ к данным по столбцам и строкам.
5. Написать второй пакет, в состав которого включить вызовы процедур из первого пакета.
    - В пакет также поместить процедуру изменения данных в таблицах *Процедура должна изменять длину указанной реки. Включить в пакет еще одну процедуру, которая выводит реки, впадающие в каждое море, кроме моря, указанного в параметре. Команду выборки сформировать, используя внутренний динамический SQL.*
    - Значения изменяемых данных должны передаваться в процедуру как параметры.
    - В процедурах предусмотреть обработку исключений.
    - Обеспечить подтверждение транзакций при их успешном выполнении и откат - в случае возникновения исключительной ситуации.

6. Предоставить привилегии всем пользователям базы данных Oracle на использование представления для просмотра данных.
7. Предоставить привилегию конкретному пользователю на выполнение процедуры изменения данных.

Текст программы

```
BEGIN
    EXECUTE IMMEDIATE 'DROP TABLE Rivers_info';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE != -942 THEN
            RAISE;
        END IF;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'DROP TABLE Seas';
EXCEPTION
    WHEN OTHERS THEN
        IF SQLCODE != -942 THEN
            RAISE;
        END IF;
END;
/

DROP TRIGGER RiverLengthChange;

CREATE TABLE Seas (
    sea_id NUMBER NOT NULL,
    sea_name VARCHAR2(100) NOT NULL,
    CONSTRAINT pk_sea_id PRIMARY KEY(sea_id)
);

CREATE TABLE Rivers_info (
    river_id NUMBER NOT NULL,
    river_name VARCHAR2(100) NOT NULL,
    length_info NUMBER NOT NULL,
    sea_id NUMBER NOT NULL,
    CONSTRAINT pk_river PRIMARY KEY(river_id),
    CONSTRAINT fk_seas FOREIGN KEY (sea_id) REFERENCES Seas(sea_id)
```

```
);

CREATE OR REPLACE PACKAGE RGR_Utils IS
    PROCEDURE FIll;
    PROCEDURE Burn;
END RGR_Utils;
/
CREATE OR REPLACE PACKAGE BODY RGR_Utils IS
    PROCEDURE FIll IS
    BEGIN
        RGR_Utils.Burn;

        INSERT INTO Seas VALUES(0, 'Красное');
        INSERT INTO Seas VALUES(1, 'Белое');
        INSERT INTO Seas VALUES(2, 'Черное');
        INSERT INTO Seas VALUES(3, 'Баренцево');
        INSERT INTO Seas VALUES(4, 'Карибское');
        INSERT INTO Seas VALUES(5, 'Азовское');
        INSERT INTO Seas VALUES(6, 'Байкал');

        INSERT INTO Rivers_info VALUES(0, 'Дон', 1870,5);
        INSERT INTO Rivers_info VALUES(1, 'Днестр', 1352, 2);
        INSERT INTO Rivers_info VALUES(2, 'Дунай', 2850, 2);
        INSERT INTO Rivers_info VALUES(3, 'Нил', 6650, 0);
        INSERT INTO Rivers_info VALUES(4, 'Северная двина', 744, 1);
        INSERT INTO Rivers_info VALUES(5, 'Ориноко', 2140, 4);
        INSERT INTO Rivers_info VALUES(6, 'Магдалена', 1538, 4);
        INSERT INTO Rivers_info VALUES(7, 'Печора', 1809,3);
        INSERT INTO Rivers_info VALUES(8, 'Тыя', 120, 6);
        INSERT INTO Rivers_info VALUES(9, 'Ангара', 1779, 6);
    END FIll;

    PROCEDURE Burn IS
    BEGIN
        DELETE FROM Seas;
        DELETE FROM Rivers_info;
```

```
    END Burn;
END RGR_Utils;
/

CREATE OR REPLACE PACKAGE RGR_Utils2 IS
    PROCEDURE FIll;
    PROCEDURE Burn;
    PROCEDURE changelenght(new_lenght IN VARCHAR2, rivers_name IN VARCHAR2);
    PROCEDURE rivers_in_seas(sea_name IN VARCHAR2);
END RGR_Utils2;
/
CREATE OR REPLACE PACKAGE BODY RGR_Utils2 IS
    PROCEDURE FIll IS
    BEGIN
        RGR_Utils.FIll;
    END FIll;

    PROCEDURE Burn IS
    BEGIN
        RGR_Utils.Burn;
    END Burn;

    PROCEDURE changelenght(new_lenght IN VARCHAR2, rivers_name IN VARCHAR2)
IS
    BEGIN
        UPDATE Rivers_info
        SET length_info = new_lenght
        WHERE river_name = rivers_name;

        IF SQL%NOTFOUND THEN
            DBMS_OUTPUT.PUT_LINE('Данные о ' || rivers_name || ' не
найдены!');
            ROLLBACK;
        ELSE
            COMMIT;
        END IF;
        EXCEPTION
```

```
            WHEN OTHERS THEN
                DBMS_OUTPUT.PUT_LINE('Ошибка при изменении');
                ROLLBACK;
    END changelenght;

    PROCEDURE rivers_in_seas(sea_name IN VARCHAR2) IS
    BEGIN

        EXECUTE IMMEDIATE 'CREATE OR REPLACE VIEW Rivers_in_Seas_View AS
                           SELECT r.river_name AS River, s.sea_name AS Sea
                           FROM Rivers_info r
                           JOIN Seas s ON r.sea_id = s.sea_id
                           WHERE s.sea_name != ''' || sea_name || '''';

        DBMS_OUTPUT.PUT_LINE('Реки, впадающие в каждое море, кроме ' ||
sea_name || ':');

        FOR rec IN (SELECT River, Sea FROM Rivers_in_Seas_View) LOOP
            DBMS_OUTPUT.PUT_LINE('Река: ' || rec.River || ', Море: ' ||
rec.Sea);
        END LOOP;

        EXCEPTION
            WHEN OTHERS THEN
                DBMS_OUTPUT.PUT_LINE('Ошибка');
                ROLLBACK;
    END rivers_in_seas;



END RGR_Utils2;
/

CREATE OR REPLACE TRIGGER RiverLengthChange
AFTER UPDATE OF length_info ON Rivers_info
FOR EACH ROW
DECLARE
```

```
    old_length NUMBER;
BEGIN
    old_length := :OLD.length_info;
    DBMS_OUTPUT.PUT_LINE('Река ' || :NEW.river_name || ' сменила длину с ' ||
old_length || ' на ' || :NEW.length_info || ' - время изменения = ' ||
SYSTIMESTAMP);
END;
/


BEGIN
    RGR_Utils2.FIll;
    RGR_Utils2.changelenght(1111,'Дон');
    RGR_Utils2.rivers_in_seas('Черное');
    COMMIT;
END;
/


CREATE OR REPLACE VIEW viewTable AS
    SELECT Rivers_info.river_id AS river_id,
           Rivers_info.river_name AS river_name,
           Seas.sea_name AS sea_name
    FROM Rivers_info, Seas
    WHERE Rivers_info.sea_id = Seas.sea_id and Rivers_info.sea_id > 0;


SELECT * FROM viewTable;


GRANT SELECT ON viewTable TO public;
GRANT EXECUTE ON RGR_Utils2 TO public;
```

Таблицы

SEAS

SEA_ID	SEA_NAME
0	Красное
1	Белое
2	Черное
3	Баренцево
4	Карибское
5	Азовское
6	Байкал

Таблица морей

RIVERS_INFO

RIVER_ID	RIVER_NAME	LENGTH_INFO	SEA_ID
0	Дон	1870	5
1	Днестр	1352	2
2	Дунай	2850	2
3	Нил	6650	0
4	Северная двина	744	1
5	Ориноко	2140	4
6	Магдалена	1538	4
7	Печора	1809	3
8	Тыя	120	6
9	Ангара	1779	6

Таблица рек

Результаты

```
BEGIN      RGR_Utils2.FIll;      RGR_Utils2.changelenght(1111,'Дон');      COMMIT; END;

Река Дон сменила длину с 1870 на 1111 - время изменения = 19-MAR-24 02.12.07.277548000 PM +00:00

Statement processed. 0.07 seconds
```

Рис1. Работа триггера и процедуры

```
RGR_Utils2.rivers_in_seas('Черное');
```

```
Реки, впадающие в каждое море, кроме Черное:
Река: Дон, Море: Азовское
Река: Нил, Море: Красное
Река: Северная двина, Море: Белое
Река: Ориноко, Море: Карибское
Река: Магдалена, Море: Карибское
Река: Печора, Море: Баренцево
Река: Тыя, Море: Байкал
Река: Ангара, Море: Байкал

Statement processed. 0.06 seconds
```

Рис2-3. Вызов и результат работы процедуры вывода рек морей, кроме указанного моря

```
CREATE OR REPLACE VIEW viewTable AS     SELECT Rivers_info.river_id AS river_id,          Rivers_info.river_name AS river_name,          Seas.sea_name AS sea_name     FROM
Rivers_info, Seas     WHERE Rivers_info.sea_id = Seas.sea_id and Rivers_info.sea_id > 0

View created. 0.04 seconds
```

```
SELECT * FROM viewTable
```

RIVER_ID	RIVER_NAME	SEA_NAME
0	Дон	Азовское
1	Днестр	Черное
2	Дунай	Черное
4	Северная двина	Белое
5	Ориноко	Карибское
6	Магдалена	Карибское
7	Печора	Баренцево
8	Тыя	Байкал
9	Ангара	Байкал

Рис4. Объединенная таблица для просмотра данных

```
9 rows selected. 0.01 seconds
```

```
GRANT SELECT ON viewTable TO public

Statement processed. 0.01 seconds
```

```
GRANT EXECUTE ON RGR_Utils2 TO public

Statement processed. 0.01 seconds
```

Рис5. Выдача привилегий на просмотр и изменение данных.