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 5

Выполнил: студент 4 курса
ИВТ, гр. ИП-911
Давыдов И.П.
Проверил: Старший преподаватель кафедры ПМиК
Дьячкова Ирина Сергеевна
.

Новосибирск 2023

Оглавление

1. Задание

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". Создать пакет, содержащий процедуру начального заполнения таблиц данными (по 7-10 записей в таблице) и процедуру очистки таблиц (удаления записей).

Для одной из таблиц разработать триггер для обеспечения дополнительных ограничений на изменение данных таблицы (см. свой вариант задания).

Создать представление, которое позволяет запрашивать данные из обеих (связанных) таблиц. Представление должно ограничивать доступ к данным по столбцам и строкам.

Написать второй пакет, в состав которого включить вызовы процедур из первого пакета. В пакет также поместить процедуру изменения данных в таблицах (см. свой вариант задания). 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.

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

Отчет должен отвечать всем требованиям к оформлению курсовых работ и содержать текст задания, тексты сценариев, пакетов, содержимое таблиц и результаты запросов и выполнения процедур.

Таблицы должны содержать данные о Цветочных магазинах и Видах цветов. Каждый вид цветов можно купить в разных магазинах. Процедура должна добавлять новый вид цветов и магазин для его продажи. Триггер должен регистрировать в регистрационной таблице добавление данных с указанием даты и времени операций. Включить в пакет еще одну процедуру, которая выводит магазины, не продающие заданный вид цветов. Выборку данных производить в ассоциативный массив.

2. Описание работы

В результате работы были созданы две таблицы Flower_Types и Flower_Shops.

Первая таблица содержит поля TypeID, NameType, где TypeID – является primary key (первичным ключом). Вторая таблица содержит поля ShopID, NameShop, FlowerTypeID, где ShopID – является primary key (первичным ключом) и FlowerTypeID – является foreign key (внешним ключом), который ссылается на ID продукта из первой таблицы. Для правильного распределения идентификаторов, были созданы две последовательности FlowerTypeSeq – для цветов и FlowerShopSeq – для магазинов.

Первый пакет Flower_Shop_Package содержит в себе две процедуры:

- Init_data – процедура инициализации данных в таблицах.
- Clear_data – процедура очистки таблиц.

Далее был разработан триггер Flower_Types_Trigger, который реагирует на добавление данных в таблицу.

Было создано представление view_table_flowers, которое выводит две таблицы, ограниченные по строкам и столбцам.

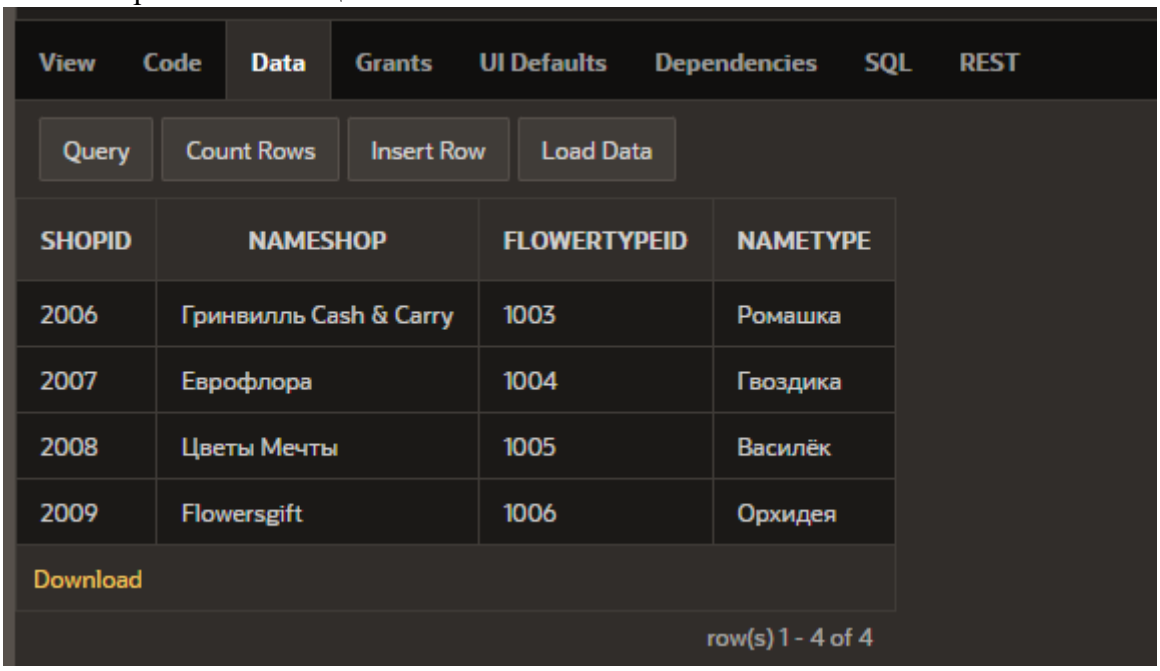

Рис. 1 – Представление view_table_flowers (Flower_Types.TypeID > 1002)

Второй пакет Flower_Shop_Package2 содержит в себе три процедуры:

- create_clear_table; – процедура, которая вызывает процедуры из первого пакета;
- add_new_data_table – процедура, которая добавляет данные в таблицы;
- withdrawal_of_stores_not_used – процедура, которая выводит магазины, не продающие заданный вид цветов (выборка производилась в ассоциативный массив (BULK COLLECT INTO t_names, где t_names – ассоциативный массив)).

Были предоставлены привилегии всем пользователям базы данных Oracle на использование представления для просмотра данных.

Были предоставлены привилегии конкретному пользователю на выполнение процедуры изменения данных.

```
GRANT SELECT ON view_table_flowers TO public

Statement processed. 0.01 seconds
```

Рис. 2 – Привилегии всем пользователям базы данных Oracle на использование представления для просмотра данных

```
GRANT EXECUTE ON Flower_Shop_Package2 TO public

Statement processed. 0.00 seconds
```

Рис. 3 – Привилегии конкретному пользователю на использование второго пакета.

```
BEGIN      Flower_Shop_Package2.withdrawal_of_stores_not_used('Роза'); END;

t_names(1) = Флорист Новосибирский
t_names(2) = ProЦветы
t_names(3) = БукетОпт
t_names(4) = Амур
t_names(5) = Гринвилль Cash & Carry
t_names(6) = Еврофлора
t_names(7) = Цветы Мечты
t_names(8) = Flowersgift

Statement processed. 0.02 seconds
```

Рис. 4 – Результат работы процедуры withdrawal_of_stores_not_used. Параметр ‘роза’.

3. Результаты работы

SHOPID	NAMESHOP	FLOWERTYPEID
2000	La Rose	1000
2001	Моулин роз	1000
2002	Флорист Новосибирский	1001
2003	ProЦветы	1001
2004	БукетОпт	1002
2005	Амур	1002
2006	Гринвилль Cash & Carry	1003
2007	Еврофлора	1004
2008	Цветы Мечты	1005
2009	Flowersgift	1006

Рис. 7 – Вывод таблицы Магазины

TYPEID	NAMETYPE
1000	Роза
1001	Тюльпан
1002	Нарцисс
1003	Ромашка
1004	Гвоздика
1005	Василёк
1006	Орхидея
1010	Хризантема

Рис. 8 – Вывод таблицы Цветы

4. Листинг программы

```
DROP SEQUENCE FlowerTypeSeq;
DROP SEQUENCE FlowerShopSeq;
DROP TRIGGER Flower_Types_Trigger;
CREATE SEQUENCE FlowerTypeSeq start with 1000 increment by 1;
CREATE SEQUENCE FlowerShopSeq start with 2000 increment by 1;


DROP TABLE Flower_Shops;
DROP TABLE Flower_Types;
DROP TABLE Registration_Log;
-- таблицы
CREATE TABLE Flower_Types (
  TypeID NUMBER(4, 0) not null,
  NameType VARCHAR2(50) NOT NULL,
  CONSTRAINT PK_TypeID PRIMARY KEY(TypeID)
);
/
CREATE TABLE Flower_Shops (
  ShopID NUMBER(4, 0) not null,
  NameShop VARCHAR2(50) not null,
  FlowerTypeID NUMBER(4, 0) not null,
  CONSTRAINT PK_ShopID PRIMARY KEY(ShopID),
  CONSTRAINT FK_FlowerTypeID FOREIGN KEY(FlowerTypeID) REFERENCES
Flower_Types(TypeID) ON DELETE CASCADE
);
/
CREATE TABLE Registration_Log (
  NameTable VARCHAR2(50) NOT NULL,
  TypeOperation VARCHAR2(10) NOT NULL,
  TimeOperation TIMESTAMP WITH TIME ZONE
);
/
CREATE OR REPLACE PACKAGE Flower_Shop_Package IS
```

```
    PROCEDURE Init_Data;

    PROCEDURE Clear_Data;

END;

/

CREATE OR REPLACE PACKAGE BODY Flower_Shop_Package IS

  PROCEDURE Init_Data IS

  BEGIN

    INSERT INTO Flower_Types VALUES (FlowerTypeSeq.NEXTVAL, 'Роза');

    INSERT INTO Flower_Shops VALUES (FlowerShopSeq.NEXTVAL, 'La Rose',
FlowerTypeSeq.CURRVAL);

    INSERT INTO Flower_Shops VALUES (FlowerShopSeq.NEXTVAL, 'Моулин роз',
FlowerTypeSeq.CURRVAL);

    INSERT INTO Flower_Types VALUES (FlowerTypeSeq.NEXTVAL, 'Тюльпан');

    INSERT INTO Flower_Shops VALUES (FlowerShopSeq.NEXTVAL, 'Флорист
Новосибирский', FlowerTypeSeq.CURRVAL);

    INSERT INTO Flower_Shops VALUES (FlowerShopSeq.NEXTVAL, 'ProЦветы',
FlowerTypeSeq.CURRVAL);

    INSERT INTO Flower_Types VALUES (FlowerTypeSeq.NEXTVAL, 'Нарцисс');

    INSERT INTO Flower_Shops VALUES (FlowerShopSeq.NEXTVAL, 'БукетОпт',
FlowerTypeSeq.CURRVAL);

    INSERT INTO Flower_Shops VALUES (FlowerShopSeq.NEXTVAL, 'Амур',
FlowerTypeSeq.CURRVAL);

    INSERT INTO Flower_Types VALUES (FlowerTypeSeq.NEXTVAL, 'Ромашка');

    INSERT INTO Flower_Shops VALUES (FlowerShopSeq.NEXTVAL, 'Гринвилль Cash &
Carry', FlowerTypeSeq.CURRVAL);

    INSERT INTO Flower_Types VALUES (FlowerTypeSeq.NEXTVAL, 'Гвоздика');

    INSERT INTO Flower_Shops VALUES (FlowerShopSeq.NEXTVAL, 'Еврофлора',
FlowerTypeSeq.CURRVAL);

    INSERT INTO Flower_Types VALUES (FlowerTypeSeq.NEXTVAL, 'Василёк');

    INSERT INTO Flower_Shops VALUES (FlowerShopSeq.NEXTVAL, 'Цветы Мечты',
FlowerTypeSeq.CURRVAL);

    INSERT INTO Flower_Types VALUES (FlowerTypeSeq.NEXTVAL, 'Орхидея');

    INSERT INTO Flower_Shops VALUES (FlowerShopSeq.NEXTVAL, 'Flowersgift',
FlowerTypeSeq.CURRVAL);

    COMMIT;

  END Init_Data;
```

```
  PROCEDURE Clear_Data IS
  BEGIN
    DELETE FROM Flower_Types;
    DELETE FROM Flower_Shops;
    EXECUTE IMMEDIATE 'DROP SEQUENCE FlowerTypeSeq'; --начальное значение
последовательности и удаление
    EXECUTE IMMEDIATE 'DROP SEQUENCE FlowerShopSeq';
    EXECUTE IMMEDIATE 'CREATE SEQUENCE FlowerTypeSeq start with 1000
increment by 1 ';
    EXECUTE IMMEDIATE 'CREATE SEQUENCE FlowerShopSeq start with 2000
increment by 1';
    COMMIT;
  END Clear_Data;
END;
/
CREATE OR REPLACE PACKAGE Flower_Shop_Package2 IS
    PROCEDURE create_clear_table;
    PROCEDURE withdrawal_of_stores_not_used (NameTypeFlowers IN VARCHAR2); --
вывод магазинов не прод
    PROCEDURE add_new_data_table (idFlower IN NUMBER, nameFlower IN VARCHAR2,
idShop IN NUMBER, nameShop IN VARCHAR2);
END;
/
CREATE OR REPLACE PACKAGE BODY Flower_Shop_Package2 IS
    TYPE t_flower_shop IS TABLE OF Flower_Shops.NameShop%TYPE; --асоц массив
    t_names t_flower_shop;

    PROCEDURE create_clear_table IS
    BEGIN
        Flower_Shop_Package.Clear_Data;
        Flower_Shop_Package.Init_Data;
    END create_clear_table;

    PROCEDURE withdrawal_of_stores_not_used (NameTypeFlowers IN VARCHAR2) AS
    invalid_count EXCEPTION;
```

```
    BEGIN
        SELECT NameShop BULK COLLECT INTO t_names
        FROM Flower_Shops
        WHERE ShopID NOT IN (
            SELECT ShopID FROM Flower_Shops, Flower_Types
            WHERE Flower_Shops.FlowerTypeID = Flower_Types.TypeID AND
Flower_Types.NameType = NameTypeFlowers
        );
        IF t_names.COUNT = 0 THEN
            RAISE invalid_count;
        ELSE
            FOR n IN 1..t_names.COUNT LOOP
                IF t_names.EXISTS(n) THEn
                    DBMS_OUTPUT.PUT_LINE('t_names(' || n || ') = ' ||
t_names(n));
                ELSE
                    DBMS_OUTPUT.PUT_LINE('t_names(' || n || ') = empty');
                END IF;
            END LOOP;
        END IF;
        EXCEPTION
        WHEN invalid_count THEN
            DBMS_OUTPUT.PUT_LINE('AssocArray is EMPTY');
            ROLLBACK;
    END withdrawal_of_stores_not_used;


    PROCEDURE add_new_data_table (idFlower IN NUMBER, nameFlower IN VARCHAR2,
idShop IN NUMBER, nameShop IN VARCHAR2) AS --переделать с параметрами
    BEGIN
        INSERT INTO Flower_Types VALUES (idFlower, nameFlower); --1010
        INSERT INTO Flower_Shops VALUES (idShop, nameShop, idFlower); --2011
        COMMIT;
        EXCEPTION
        WHEN OTHERS THEN
```

```
        DBMS_OUTPUT.PUT_LINE('Ошибка.');
    END add_new_data_table;


END;
/
BEGIN
    Flower_Shop_Package2.create_clear_table;
END;
/
CREATE OR REPLACE TRIGGER Flower_Types_Trigger
BEFORE INSERT ON Flower_Shops
FOR EACH ROW
BEGIN
    IF (INSERTING) THEN
    INSERT INTO Registration_Log (NameTable, TypeOperation, TimeOperation)
        VALUES ('Flower_Shops', 'INSERT', CURRENT_TIMESTAMP);
        COMMIT;
    END IF;
END;
/
CREATE OR REPLACE VIEW view_table_flowers AS
    SELECT Flower_Shops.ShopID, Flower_Shops.NameShop,
Flower_Shops.FlowerTypeID, Flower_Types.NameType FROM
    Flower_Shops, Flower_Types WHERE Flower_Shops.FlowerTypeID =
Flower_Types.TypeID
    AND Flower_Types.TypeID > 1002;
SELECT * FROM view_table_flowers;


GRANT SELECT ON view_table_flowers TO public;
GRANT EXECUTE ON Flower_Shop_Package2 TO public;
/
BEGIN
    Flower_Shop_Package2.withdrawal_of_stores_not_used('Роза');
END;
```

```
/

BEGIN

    Flower_Shop_Package2.add_new_data_table(1010, 'Хризантема', 2011,
'Сфинкс');

END;

/

BEGIN

    Flower_Shop_Package2.withdrawal_of_stores_not_used('Тюльпан');

END;

/
```