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

Сетевые базы данных

Расчетно-графическое задание

Вариант 12

Выполнил: студент 4 курса

Ф. ИВТ, группа: ИП-813

Огорелков А. А.

Проверил:

старший преподаватель кафедры ПМиК

Грязнов Николай Геннадьевич

Новосибирск, 2022

**Содержание**

1. Текст задания**……………………………………………………….……….3**
2. Описание алгоритма**……………………………………………….........……….4**
3. Результаты работы**………………………………………………………...........…9**

**Текст задания**

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

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. 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 и привилегию на выполнение процедуры изменения данных.

Внести в таблицы данные о Морских круизах и Портах.

Любой круиз может включать заход в несколько портов.

Процедура должна удалять круиз. Триггер должен

запрещать удаление, если число круизов менее трех.

Включить в пакет процедуру, которая выбирает указанный

в параметре круиз и его порты; выборка должна выполняться в

коллекцию. Включить в пакет еще одну процедуру, которая выводит переданную в параметре коллекцию с данными о круизе и портах.

**Описание алгоритма**

1. Создание таблиц

CREATE TABLE CRUISE(

CRUISE\_ID NUMBER(4),

CRUISE\_NAME VARCHAR2(100) NOT NULL,

CONSTRAINT ports\_pk PRIMARY KEY (CRUISE\_ID)

);

CREATE TABLE PORTS(

PORTS\_ID NUMBER(4),

CRUISE\_ID NUMBER(4),

PORTS\_NAME VARCHAR2(100) NOT NULL,

CONSTRAINT cruise\_pk PRIMARY KEY (PORTS\_ID),

CONSTRAINT cruise\_fk FOREIGN KEY (CRUISE\_ID) REFERENCES CRUISE(CRUISE\_ID)

);

В данном сценарии создаются две таблицы – таблица типов круизов (cruise) и самих портов (PORTS).

1. Создание пакета №1

CREATE OR REPLACE PACKAGE Pack\_1\_Ogo IS

PROCEDURE Ports\_tables\_init;

PROCEDURE Ports\_tables\_delete;

END Pack\_1\_Ogo;

/

CREATE OR REPLACE PACKAGE BODY Pack\_1\_Ogo IS

PROCEDURE Ports\_tables\_init IS

BEGIN

INSERT INTO CRUISE VALUES (1, 'Вокруг света');

INSERT INTO CRUISE VALUES (2, 'До Австралии');

INSERT INTO CRUISE VALUES (3, 'По Европе');

INSERT INTO CRUISE VALUES (4, 'По Азии');

INSERT INTO CRUISE VALUES (5, 'По Америке');

INSERT INTO CRUISE VALUES (6, 'По чёрному морю');

INSERT INTO CRUISE VALUES (7, 'По Океану');

INSERT INTO CRUISE VALUES (8, 'По миру');

INSERT INTO PORTS VALUES (1, 3, 'Новосибирский');

INSERT INTO PORTS VALUES (2, 2, 'Городской');

INSERT INTO PORTS VALUES (3, 6, 'Сочинский');

INSERT INTO PORTS VALUES (4, 3, 'Новый');

INSERT INTO PORTS VALUES (5, 1, 'Российский');

INSERT INTO PORTS VALUES (6, 1, 'Московский');

INSERT INTO PORTS VALUES (7, 2, 'Омский');

INSERT INTO PORTS VALUES (8, 5, 'Волшебный');

COMMIT;

EXCEPTION

WHEN others THEN

DBMS\_OUTPUT.PUT\_LINE(SQLERRM);

ROLLBACK;

END;

PROCEDURE Ports\_tables\_delete IS

BEGIN

DELETE FROM PORTS;

DELETE FROM CRUISE;

COMMIT;

EXCEPTION

WHEN others THEN

DBMS\_OUTPUT.put\_line(SQLERRM);

ROLLBACK;

END;

END Pack\_1\_Ogo;

/

Пакет №1 содержит 2 процедуры – вставка и удаление данных из таблиц.

1. Создание триггера

CREATE OR REPLACE TRIGGER PORTS\_TRIGGER

BEFORE DELETE ON CRUISE

DECLARE

var NUMERIC(4);

BEGIN

SELECT COUNT(\*) INTO var FROM CRUISE;

IF var < 3 THEN

raise\_application\_error(-20001, 'CRUISE < 3');

END IF;

END;

/

Данный триггер запрещает удаление круизов, если их меньше 3.

1. Создание представления

CREATE OR REPLACE TRIGGER PORTS\_TRIGGER

BEFORE DELETE ON CRUISE

DECLARE

var NUMERIC(4);

BEGIN

SELECT COUNT(\*) INTO var FROM CRUISE;

IF var < 3 THEN

raise\_application\_error(-20001, 'CRUISE < 3');

END IF;

END;

1. Создание пакета №2

CREATE OR REPLACE PACKAGE BODY Pack\_2\_Ogo IS

TYPE test\_table IS TABLE OF VARCHAR2(100) INDEX BY PLS\_INTEGER;

my\_test test\_table;

TYPE cruise\_table2 IS TABLE OF VARCHAR2(100) INDEX BY PLS\_INTEGER;

my\_cruise cruise\_table2;

zap NUMBER(10);

count\_zap NUMBER(10);

counter NUMBER(10);

my NUMBER(10);

PROCEDURE Ports\_tables\_init\_call AS

BEGIN

Pack\_1\_Ogo.Ports\_tables\_init;

END Ports\_tables\_init\_call;

PROCEDURE Ports\_tables\_delete\_call AS

BEGIN

Pack\_1\_Ogo.Ports\_tables\_delete;

END Ports\_tables\_delete\_call;

PROCEDURE Delete\_cruise(name\_cruise IN VARCHAR2) AS

BEGIN

DELETE FROM CRUISE WHERE CRUISE\_NAME = name\_cruise;

COMMIT;

EXCEPTION

when OTHERS then

DBMS\_OUTPUT.PUT\_LINE('Ошибка');

END Delete\_cruise;

PROCEDURE Collection\_add(name\_cruise1 IN VARCHAR2) AS

BEGIN

my := 0;

for jj in (SELECT \* FROM CRUISE) loop

--DBMS\_OUTPUT.PUT\_LINE(jj.CRUISE\_NAME);

if jj.CRUISE\_NAME = name\_cruise1 then

zap := jj.CRUISE\_ID;

my := my + 1;

my\_test(my) := ('Круиз - '||jj.CRUISE\_NAME);

end if;

end loop;

--DBMS\_OUTPUT.PUT\_LINE(zap);

--SELECT count(\*) INTO count\_zap FROM PORTS WHERE CRUISE\_ID = zap;

for jj in (SELECT \* FROM PORTS) loop

--DBMS\_OUTPUT.PUT\_LINE(jj.CRUISE\_NAME);

if jj.CRUISE\_ID = zap then

--DBMS\_OUTPUT.PUT\_LINE(jj.PORTS\_NAME);

my := my + 1;

my\_test(my) := ('Порт - '||jj.PORTS\_NAME);

end if;

end loop;

FOR i IN my\_test.FIRST..my\_test.LAST LOOP

IF my\_test.EXISTS(i) THEN

DBMS\_OUTPUT.PUT\_LINE(my\_test(i));

END IF;

END LOOP;

EXCEPTION

when OTHERS then

DBMS\_OUTPUT.PUT\_LINE('Ошибка');

END Collection\_add;

PROCEDURE Collection\_view(number\_id IN NUMBER,collection IN VARCHAR2) AS

BEGIN

--DBMS\_OUTPUT.PUT\_LINE(number\_id||' '||collection);

my\_cruise(number\_id) := collection;

DBMS\_OUTPUT.PUT\_LINE(my\_cruise(number\_id));

/\*

FOR i IN my\_cruise.FIRST..my\_cruise.LAST LOOP

IF my\_cruise.EXISTS(i) THEN

DBMS\_OUTPUT.PUT\_LINE(my\_cruise(i));

END IF;

END LOOP;

\*/

END Collection\_view;

END Pack\_2\_Ogo;

/

1. Предоставление привилегий

GRANT SELECT ON PORTS TO PUBLIC;

GRANT EXECUTE ON Pack\_2\_ogo TO PUBLIC;

**Результаты работы**

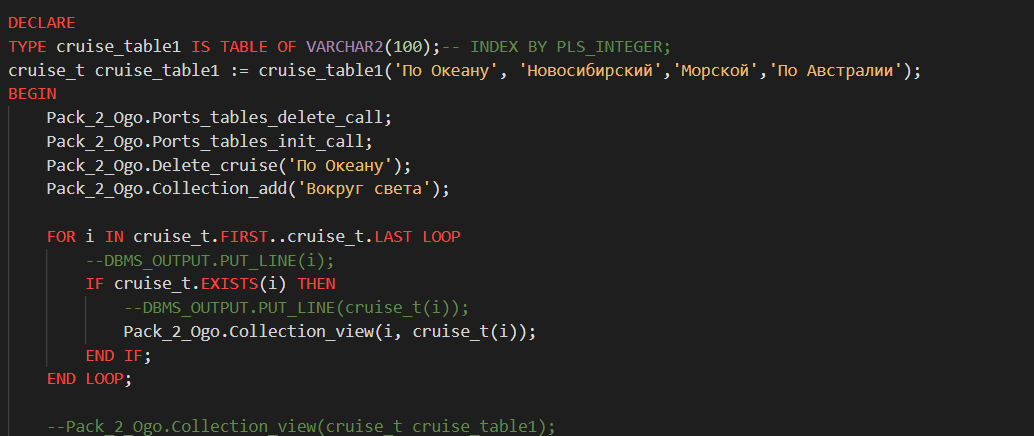


Рисунок 1. Демонстрация работ всех функций.

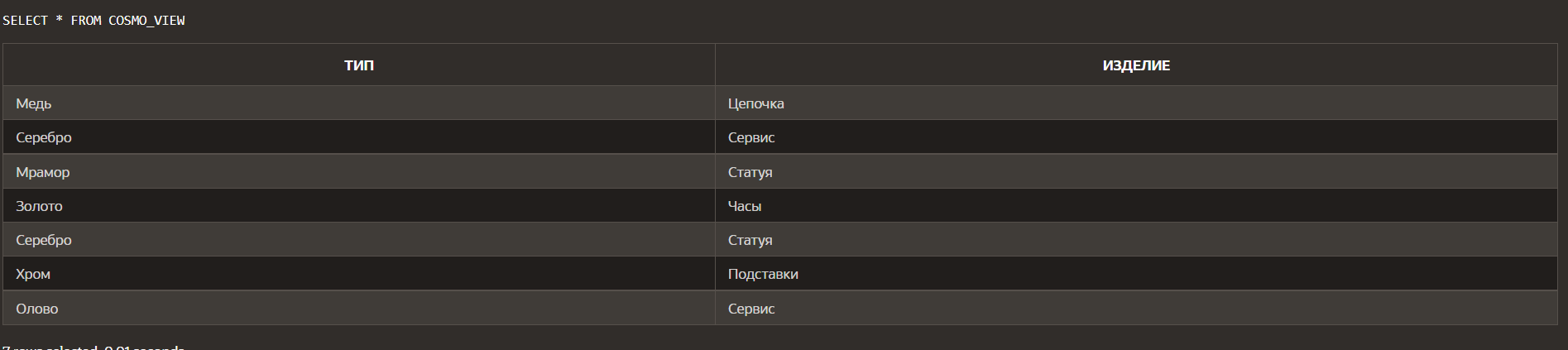
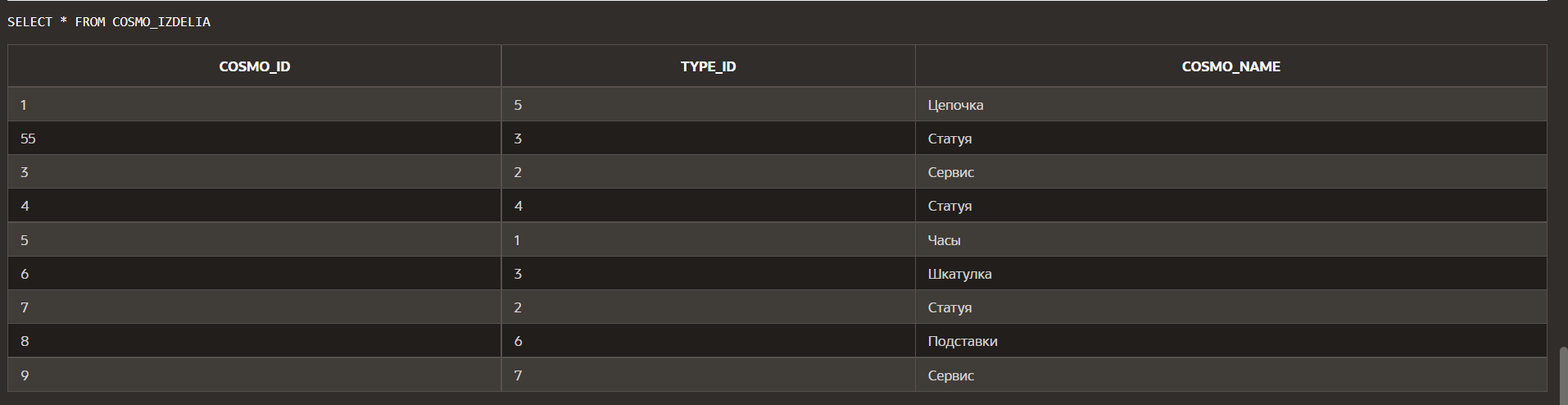


Рисунок 3. Представление



Рисунок 4. Таблица CRUISE

Рисунок 5. Таблица PORTS с изменённым первичным ключом.