Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

Институт информатики и вычислительной техники

09.03.01 "Информатика и вычислительная техника"

профиль "Программное обеспечение средств вычислительной техники и автоматизированных систем"

Кафедра прикладной математики и кибернетики

**Расчетно-графическая работа по дисциплине  
 Сетевые базы данных**

Вариант 10

Выполнил:

Cтудент гр. ИП-813 \_\_\_\_\_\_\_\_\_\_\_\_\_/./

ФИО студента

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Проверил:

Ассистент кафедры ПМиК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Грязнов Н.Г./

ФИО преподавателя

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г. Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Новосибирск 2022 г.

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 и привилегию на выполнение процедуры изменения данных.

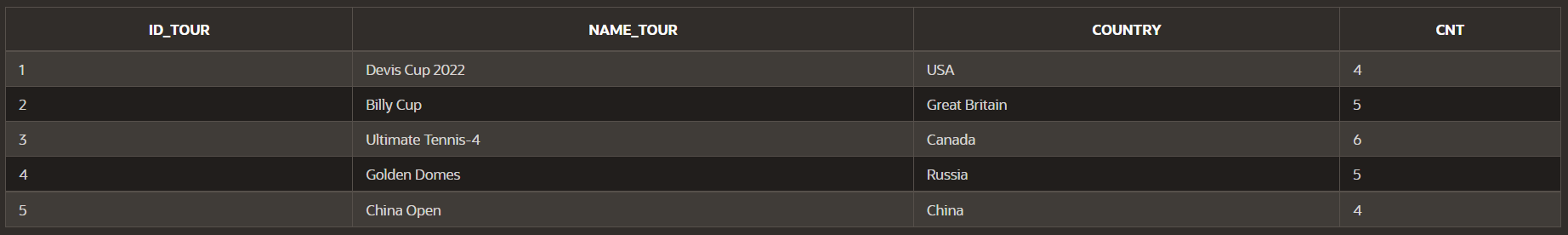
Поместить в таблицы информацию о Теннисных турнирах и Теннисистах. В каждом турнире принимает участие не менее четырех теннисистов. Процедура должна изменять страну-организатор турнира. Триггер должен регистрировать изменение и запрещать его, если изменений было уже не менее двух. Включить в пакет еще одну процедуру, которая выводит турниры с числом теннисистов более указанного в параметре. Выборку данных производить в коллекцию.

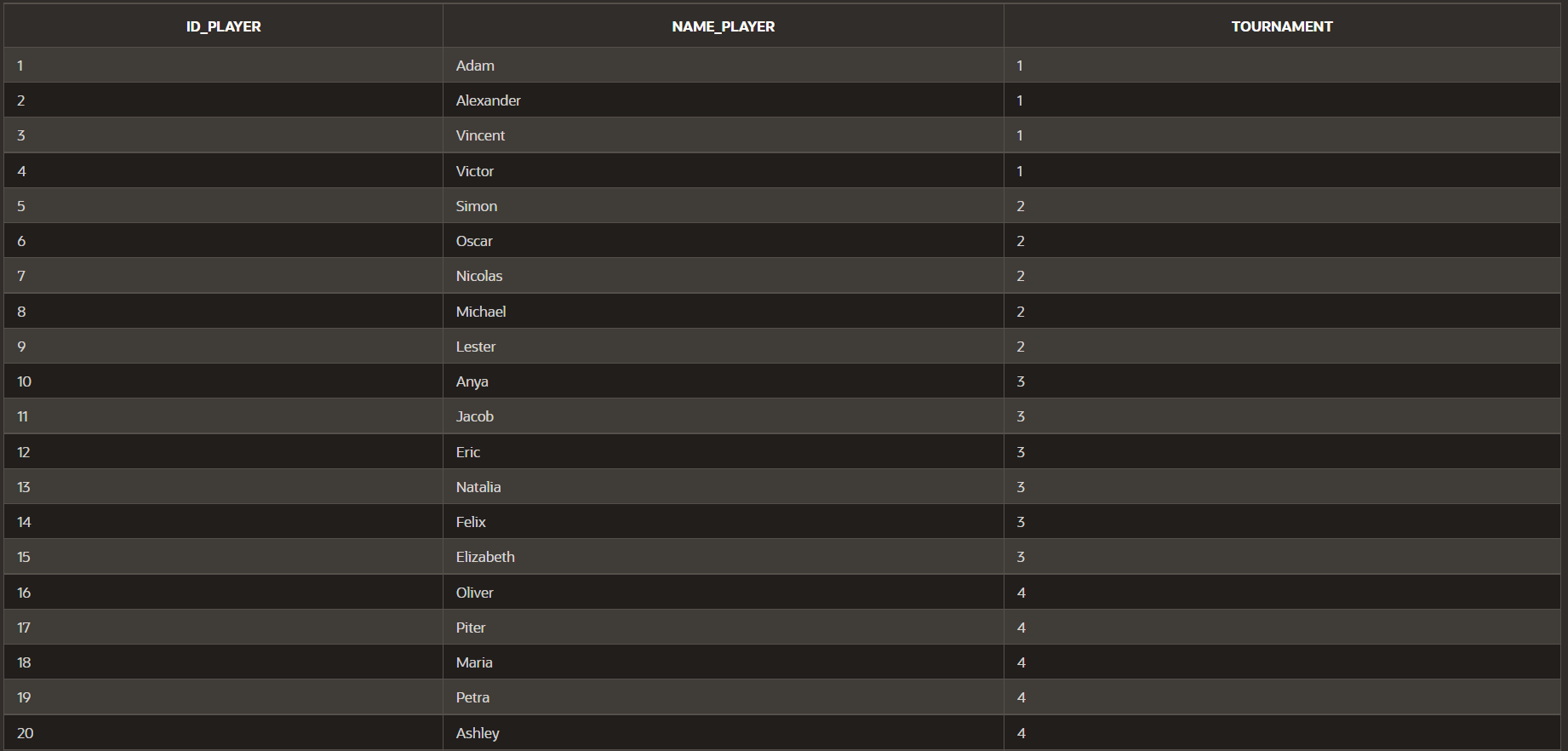
Описание работы

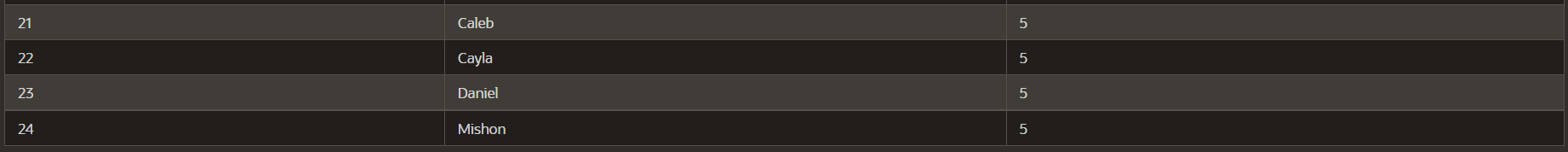
В результате работы было создано 2 таблицы tennis\_players и tennis\_tour. Первая таблица содержит информацию о теннисных турнирах, где определены следующие поля: id\_tour – первичный ключ, name\_tour, country, cnt. Вторая содержит информацию о теннисистах, поля таблицы: id\_player – первичный ключ, name\_player, tournament. Последнее поле tournament – это внешний ключ, который ссылается на id турнира из первой таблицы. Для корректного распределения идентификаторов было создано 2 последовательности: seq\_tour – для таблицы туров, seq\_players – для таблицы теннисистов.

Первый пакет WorkTablePack содержит процедуры для заполнения и очистки таблиц.

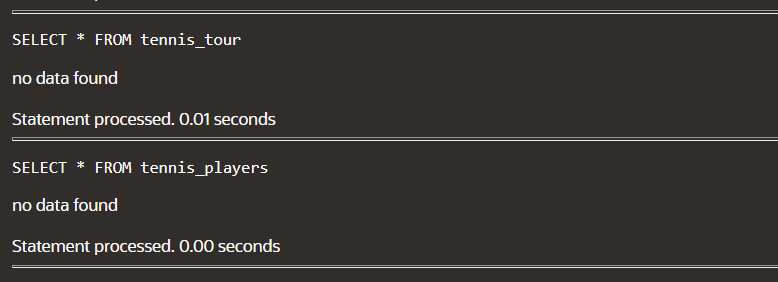
Заполнение таблиц:



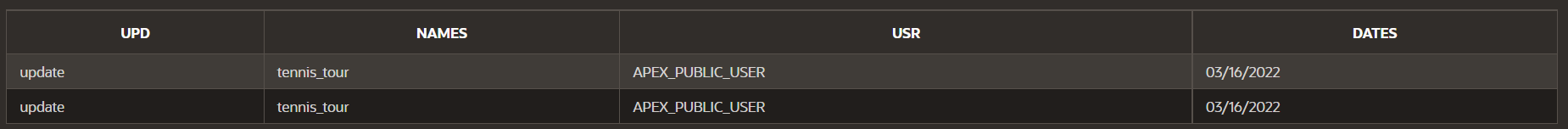




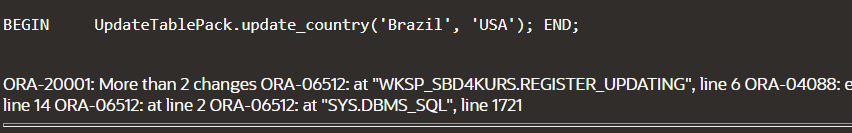
Очищение таблиц:



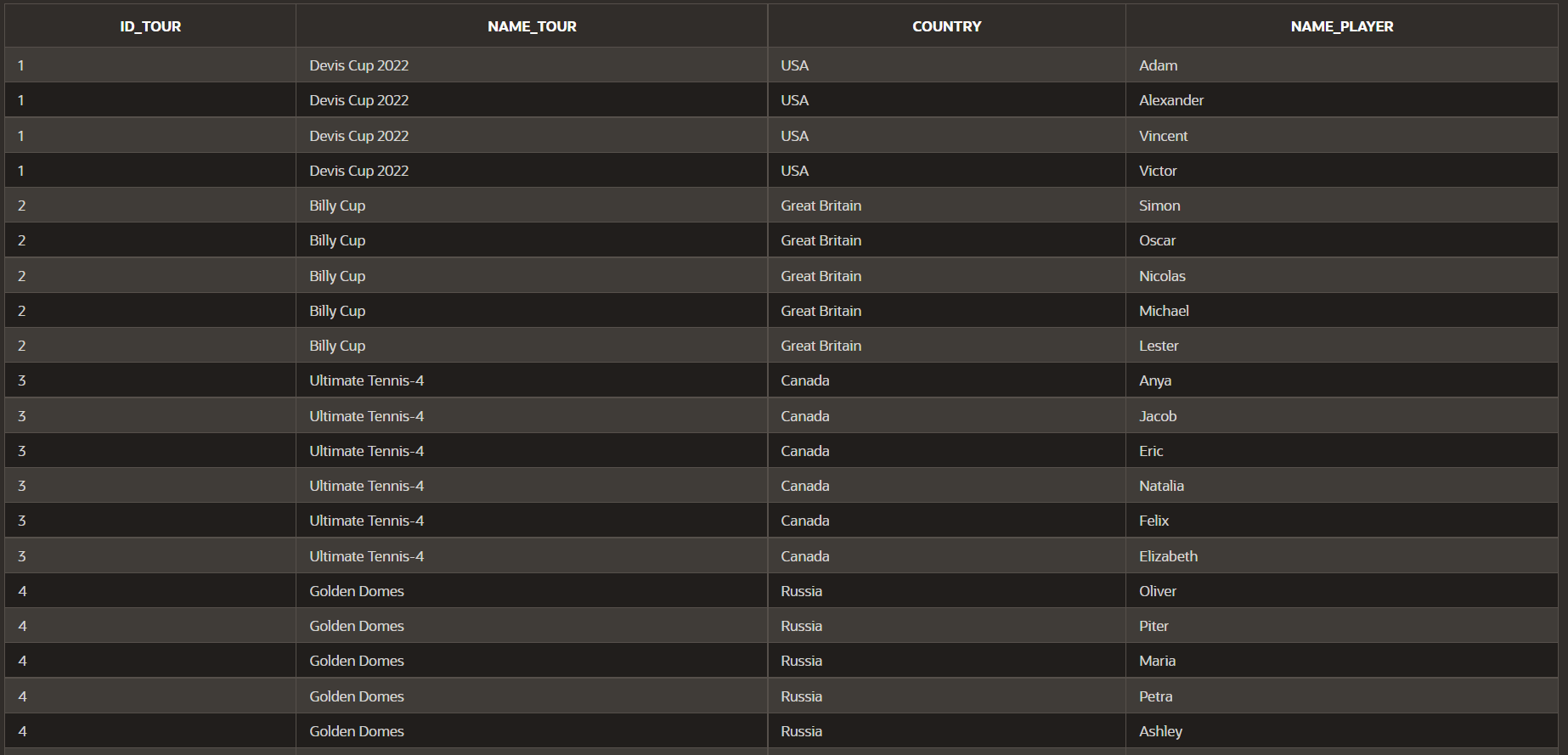
Также для таблицы tennis\_tour был создан триггер register\_updating, который регистрирует изменения и записывает их в таблицу reg\_triggers:



Для ограничения на изменения в количестве не более двух раз, я просто считаю количество записей в триггере и, если их более двух, выдаю исключение:



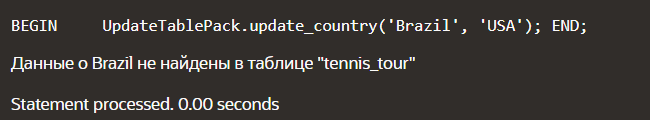
Представление view\_tables отображает названия турниров и связанных с ними игроков:



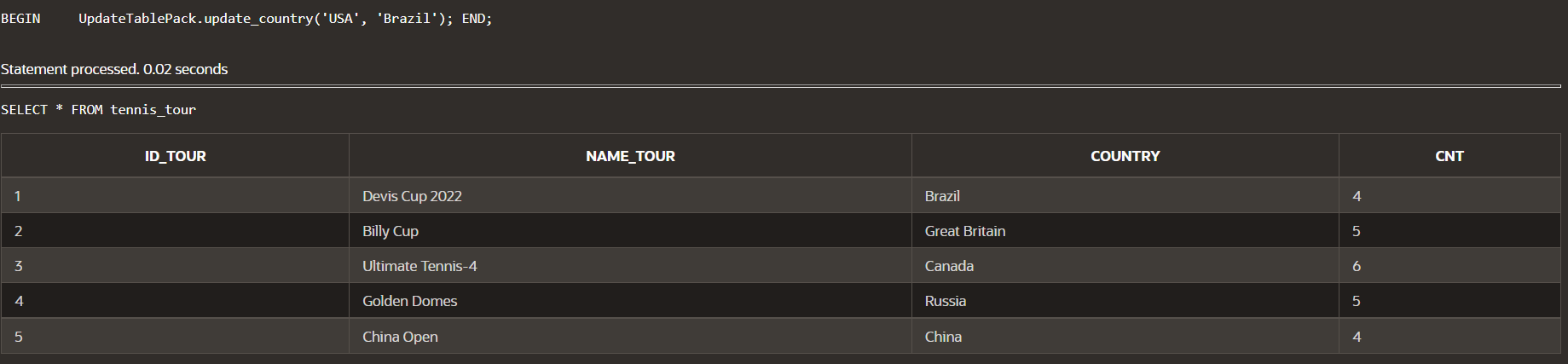


Второй пакет UpdateTablePack содержит:

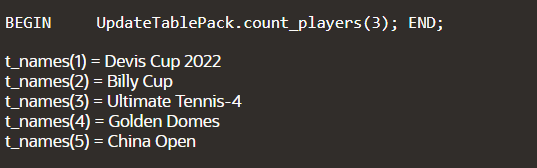
1. Процедуру upd\_data, которая включает в себя вызов процедур и первого пакета;
2. Процедуру update\_country с двумя параметрами – существующую страну и новую, которая изменяет данные в таблице турниров, а точнее страну. Если существующей страны нет в таблице – выдается исключение:



Если ошибок нет, то в таблице изменяется название страны:



1. Тип TABLE, куда будет заноситься выборка турниров с числом теннисистов;
2. Процедуру count\_players с одним параметром, которая осуществляет заполнение коллекции. В качестве параметра мы передаем тип NUMBER. Процедура заполняет и выводит коллекцию и числом теннисистов более, чем указано в параметре:



Листинг

DROP SEQUENCE seq\_tour;

DROP SEQUENCE seq\_players;

CREATE SEQUENCE seq\_tour;

CREATE SEQUENCE seq\_players;

DROP TABLE tennis\_players;

DROP TABLE tennis\_tour;

DROP TABLE reg\_triggers;

CREATE TABLE tennis\_tour (

id\_tour NUMBER(4, 0) NOT NULL,

name\_tour VARCHAR2(20) NOT NULL,

country VARCHAR2(20) NOT NULL,

cnt NUMBER(2, 0) NOT NULL,

CONSTRAINT id\_tour PRIMARY KEY(id\_tour)

);

CREATE TABLE tennis\_players (

id\_player NUMBER(4, 0) NOT NULL,

name\_player VARCHAR2(20) NOT NULL,

tournament NUMBER(4, 0) NOT NULL,

CONSTRAINT id\_player PRIMARY KEY(id\_player),

CONSTRAINT tournament FOREIGN KEY(tournament) REFERENCES tennis\_tour(id\_tour)

);

CREATE TABLE reg\_triggers (

upd VARCHAR2(20),

names VARCHAR2(20),

usr VARCHAR2(20),

dates DATE

);

COMMIT;

CREATE OR REPLACE PACKAGE WorkTablePack IS

PROCEDURE add\_data;

PROCEDURE delete\_data;

END;

/

CREATE OR REPLACE PACKAGE BODY WorkTablePack IS

PROCEDURE add\_data IS

BEGIN

INSERT INTO tennis\_tour VALUES (seq\_tour.NEXTVAL, 'Devis Cup 2022', 'USA', 4);

INSERT INTO tennis\_players VALUES (seq\_players.NEXTVAL, 'Adam', seq\_tour.CURRVAL);

INSERT INTO tennis\_players VALUES (seq\_players.NEXTVAL, 'Alexander', seq\_tour.CURRVAL);

INSERT INTO tennis\_players VALUES (seq\_players.NEXTVAL, 'Vincent', seq\_tour.CURRVAL);

INSERT INTO tennis\_players VALUES (seq\_players.NEXTVAL, 'Victor', seq\_tour.CURRVAL);

INSERT INTO tennis\_tour VALUES (seq\_tour.NEXTVAL, 'Billy Cup', 'Great Britain', 5);

INSERT INTO tennis\_players VALUES (seq\_players.NEXTVAL, 'Simon', seq\_tour.CURRVAL);

INSERT INTO tennis\_players VALUES (seq\_players.NEXTVAL, 'Oscar', seq\_tour.CURRVAL);

INSERT INTO tennis\_players VALUES (seq\_players.NEXTVAL, 'Nicolas', seq\_tour.CURRVAL);

INSERT INTO tennis\_players VALUES (seq\_players.NEXTVAL, 'Michael', seq\_tour.CURRVAL);

INSERT INTO tennis\_players VALUES (seq\_players.NEXTVAL, 'Lester', seq\_tour.CURRVAL);

INSERT INTO tennis\_tour VALUES (seq\_tour.NEXTVAL, 'Ultimate Tennis-4', 'Canada', 6);

INSERT INTO tennis\_players VALUES (seq\_players.NEXTVAL, 'Anya', seq\_tour.CURRVAL);

INSERT INTO tennis\_players VALUES (seq\_players.NEXTVAL, 'Jacob', seq\_tour.CURRVAL);

INSERT INTO tennis\_players VALUES (seq\_players.NEXTVAL, 'Eric', seq\_tour.CURRVAL);

INSERT INTO tennis\_players VALUES (seq\_players.NEXTVAL, 'Natalia', seq\_tour.CURRVAL);

INSERT INTO tennis\_players VALUES (seq\_players.NEXTVAL, 'Felix', seq\_tour.CURRVAL);

INSERT INTO tennis\_players VALUES (seq\_players.NEXTVAL, 'Elizabeth', seq\_tour.CURRVAL);

INSERT INTO tennis\_tour VALUES (seq\_tour.NEXTVAL, 'Golden Domes', 'Russia', 5);

INSERT INTO tennis\_players VALUES (seq\_players.NEXTVAL, 'Oliver', seq\_tour.CURRVAL);

INSERT INTO tennis\_players VALUES (seq\_players.NEXTVAL, 'Piter', seq\_tour.CURRVAL);

INSERT INTO tennis\_players VALUES (seq\_players.NEXTVAL, 'Maria', seq\_tour.CURRVAL);

INSERT INTO tennis\_players VALUES (seq\_players.NEXTVAL, 'Petra', seq\_tour.CURRVAL);

INSERT INTO tennis\_players VALUES (seq\_players.NEXTVAL, 'Ashley', seq\_tour.CURRVAL);

INSERT INTO tennis\_tour VALUES (seq\_tour.NEXTVAL, 'China Open', 'China', 4);

INSERT INTO tennis\_players VALUES (seq\_players.NEXTVAL, 'Caleb', seq\_tour.CURRVAL);

INSERT INTO tennis\_players VALUES (seq\_players.NEXTVAL, 'Cayla', seq\_tour.CURRVAL);

INSERT INTO tennis\_players VALUES (seq\_players.NEXTVAL, 'Daniel', seq\_tour.CURRVAL);

INSERT INTO tennis\_players VALUES (seq\_players.NEXTVAL, 'Mishon', seq\_tour.CURRVAL);

END add\_data;

PROCEDURE delete\_data IS

BEGIN

DELETE FROM tennis\_players;

DELETE FROM tennis\_tour;

EXECUTE IMMEDIATE 'DROP SEQUENCE seq\_tour';

EXECUTE IMMEDIATE 'DROP SEQUENCE seq\_players';

EXECUTE IMMEDIATE 'CREATE SEQUENCE seq\_tour';

EXECUTE IMMEDIATE 'CREATE SEQUENCE seq\_players';

END delete\_data;

END;

/

CREATE OR REPLACE TRIGGER register\_updating

BEFORE UPDATE ON tennis\_tour

FOR EACH ROW

DECLARE

count\_update NUMBER(2);

BEGIN

SELECT count(\*) INTO count\_update FROM reg\_triggers;

IF count\_update >= 2 THEN

RAISE\_APPLICATION\_ERROR(-20001,'More than 2 changes');

ELSE

INSERT INTO reg\_triggers (upd, names, usr, dates)

VALUES ('update', 'tennis\_tour', user, sysdate);

END IF;

END;

/

CREATE OR REPLACE PACKAGE UpdateTablePack IS

PROCEDURE upd\_data;

PROCEDURE update\_country (current\_country IN VARCHAR2, new\_country IN VARCHAR2);

TYPE tour\_names IS TABLE OF VARCHAR2(20);

PROCEDURE count\_players (counter IN NUMBER);

END;

/

CREATE OR REPLACE PACKAGE BODY UpdateTablePack IS

TYPE t\_tour\_names IS TABLE OF VARCHAR2(20) INDEX BY PLS\_INTEGER;

t\_names t\_tour\_names;

PROCEDURE upd\_data IS

BEGIN

WorkTablePack.delete\_data;

WorkTablePack.add\_data;

END upd\_data;

PROCEDURE update\_country (current\_country IN VARCHAR2, new\_country IN VARCHAR2) AS

ccountry VARCHAR2(20);

BEGIN

SELECT country INTO ccountry FROM tennis\_tour WHERE country = current\_country;

UPDATE tennis\_tour SET country = new\_country WHERE country = current\_country;

COMMIT;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Данные о ' || current\_country || ' не найдены в таблице "tennis\_tour"');

-- WHEN OTHERS THEN

-- DBMS\_OUTPUT.PUT\_LINE('Ошибка при изменении страны');

END update\_country;

PROCEDURE count\_players (counter IN NUMBER) AS

invalid\_count EXCEPTION;

BEGIN

SELECT name\_tour BULK COLLECT INTO t\_names

FROM tennis\_tour WHERE cnt > counter;

-- Проверить коллекцию на пустоту

IF t\_names.COUNT = 0 THEN

RAISE invalid\_count;

ELSE

FOR n IN 1..t\_names.COUNT LOOP

-- Проверить индекс на пустоту

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('Collection is EMPTY');

END count\_players;

END;

/

-- Заполнение таблиц данными

BEGIN

WorkTablePack.add\_data;

END;

/

SELECT \* FROM tennis\_tour;

SELECT \* FROM tennis\_players;

BEGIN

WorkTablePack.delete\_data;

END;

/

SELECT \* FROM tennis\_tour;

SELECT \* FROM tennis\_players;

BEGIN

UpdateTablePack.upd\_data;

END;

/

SELECT \* FROM tennis\_tour;

SELECT \* FROM tennis\_players;

-- Представление таблиц

CREATE OR REPLACE VIEW view\_tables AS

SELECT tennis\_tour.id\_tour, tennis\_tour.name\_tour, tennis\_tour.country, tennis\_players.name\_player

FROM tennis\_tour, tennis\_players

WHERE tennis\_tour.id\_tour = tennis\_players.tournament;

SELECT \* FROM view\_tables;

GRANT SELECT ON view\_tables TO public;

GRANT EXECUTE ON UpdateTablePack TO public;

-- Должно сработать исключение

BEGIN

UpdateTablePack.update\_country('Brazil', 'USA');

END;

/

SELECT \* FROM tennis\_tour;

BEGIN

UpdateTablePack.update\_country('USA', 'Brazil');

END;

/

SELECT \* FROM tennis\_tour;

BEGIN

UpdateTablePack.update\_country('China', 'Korea');

END;

/

SELECT \* FROM tennis\_tour;

BEGIN

UpdateTablePack.update\_country('Brazil', 'USA');

END;

/

SELECT \* FROM tennis\_tour;

SELECT \* FROM reg\_triggers;

BEGIN

UpdateTablePack.count\_players(3);

END;

/