Федеральное агентство связи

Сибирский Государственный Университет Телекоммуникаций и Информатики

**Межрегиональный центр переподготовки специалистов**

# Контрольная работа

# По дисциплине: Сетевые базы данных

**Выполнил**:

**Группа**:

**Вариант**: 5

**--списано**

**--процедура seeder\_clear очистки таблиц не сработает**

**Оценка –незачет**

Новосибирск, 2019

Создать две таблицы, имеющие уникальные поля и первичные ключи. Таблицы должны быть связаны с помощью внешнего ключа.

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

Создать представление для связанных таблиц, которое позволяет вывести только часть полей и часть строк таблиц.

Создать пакет PL/SQL, содержащий процедуру начального заполнения таблиц данными (по 5-7 записей в таблице) и процедуру очистки таблиц (удаления записей).

Добавить в пакет процедуру (см. свой вариант задания) 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.

Поместить в таблицы информацию о Теннисных турнирах и Теннисистах. В каждом турнире принимает участие не менее четырех теннисистов. Турниры должны иметь номер, наименование и страну проведения. Теннисисты должны иметь номер, имя и страну, за которую он выступает. Процедура должна удалить турниры с количеством участников, менее заданного в параметре. Триггер должен запрещать удаление после 25-го числа. Включить в пакет еще одну процедуру, которая, используя представление, выводит турниры и имена их участников; в вывод должны попадать турниры с количеством участников, заданном в параметре.

**Выполнение**

создаем таблицы

create table tournaments

 (t\_num number(4) PRIMARY KEY,

 t\_name varchar2(15) NOT NULL,

 t\_country varchar2(15) NOT NULL);

create table players

 (p\_num number(4) PRIMARY KEY,

 p\_name varchar2(15) NOT NULL,

 p\_country varchar2(15) NOT NULL);

--связь многие ко многим через связующую таблицу

create table list

 (l\_id number(4) PRIMARY KEY,

 p\_num number(4) NOT NULL,

 t\_num number(4) NOT NULL);

--определение внешнего ключа таблицы cust

ALTER TABLE list

 ADD (CONSTRAINT list\_fk\_p\_num FOREIGN KEY (p\_num)

 REFERENCES players(p\_num),

 CONSTRAINT list\_fk\_t\_num FOREIGN KEY (t\_num)

 REFERENCES tournaments(t\_num));

триггер запрещает удаление после 25-го числа, делаем для players:

CREATE OR REPLACE TRIGGER only\_befor\_25\_del

 BEFORE DELETE ON players

 FOR EACH ROW

 WHEN (TO\_CHAR(sysdate,'dd')>'25')

 BEGIN

 raise\_application\_error(-20400, 'разрешено удалять игроков только до 25 числа любого месяца');

END only\_befor\_25\_del;



-- Создать пакет PL/SQL, содержащий процедуру начального заполнения таблиц данными (по 5-7 записей в таблице)

-- и процедуру очистки таблиц (удаления записей).

CREATE OR REPLACE PACKAGE kr\_pack AS

 PROCEDURE seeder\_add;

 PROCEDURE seeder\_clear;

END kr\_pack;

CREATE OR REPLACE PACKAGE BODY kr\_pack AS

 PROCEDURE seeder\_add IS

 BEGIN

 -- таблица tournaments

 INSERT INTO tournaments VALUES (1001, 'Big Helm', 'Franch');

 INSERT INTO tournaments VALUES (1002, 'ATP', 'Italy');

 INSERT INTO tournaments VALUES (1003, 'WTA', 'USA');

 INSERT INTO tournaments VALUES (1004, 'Davis cup', 'Franch');

 INSERT INTO tournaments VALUES (1005, 'Kreml cup', 'Russia');

 -- таблица players

 INSERT INTO players VALUES (2001, 'Joan', 'Franch');

 INSERT INTO players VALUES (2002, 'Pier', 'Franch');

 INSERT INTO players VALUES (2003, 'Борис', 'Russia');

 INSERT INTO players VALUES (2004, 'Анна', 'Russia');

 INSERT INTO players VALUES (2005, 'Steave', 'USA');

 INSERT INTO players VALUES (2006, 'Carlos', 'Italy');

 -- таблица связей list

 INSERT INTO list VALUES (3001, 2001, 1001);

 INSERT INTO list VALUES (3002, 2002, 1001);

 INSERT INTO list VALUES (3003, 2003, 1001);

 INSERT INTO list VALUES (3004, 2004, 1001);

 INSERT INTO list VALUES (3005, 2001, 1002);

 INSERT INTO list VALUES (3006, 2003, 1002);

 INSERT INTO list VALUES (3007, 2004, 1002);

 INSERT INTO list VALUES (3008, 2006, 1002);

 INSERT INTO list VALUES (3009, 2005, 1002);

 INSERT INTO list VALUES (3010, 2001, 1003);

 INSERT INTO list VALUES (3011, 2004, 1003);

 INSERT INTO list VALUES (3012, 2002, 1003);

 INSERT INTO list VALUES (3013, 2006, 1003);

 INSERT INTO list VALUES (3014, 2001, 1004);

 INSERT INTO list VALUES (3015, 2004, 1004);

 INSERT INTO list VALUES (3016, 2005, 1004);

 INSERT INTO list VALUES (3017, 2006, 1004);

 INSERT INTO list VALUES (3018, 2002, 1004);

 INSERT INTO list VALUES (3019, 2003, 1004);

 INSERT INTO list VALUES (3020, 2001, 1005);

 INSERT INTO list VALUES (3021, 2003, 1005);

 INSERT INTO list VALUES (3022, 2002, 1005);

 INSERT INTO list VALUES (3023, 2005, 1005);

 DBMS\_OUTPUT.PUT\_LINE('записи добавлены');

 COMMIT;

 EXCEPTION

 WHEN OTHERS THEN

 dbms\_output.put\_line('ошибка при добавлении записей в таблицу');

 dbms\_output.put\_line('error code: '||SQLCODE);

 ROLLBACK;

 END seeder\_add;

 PROCEDURE seeder\_clear IS

 BEGIN

 delete from tournaments;

 delete from players;

 delete from list; **--не удалится, т.к. связаны внешним ключом**

 DBMS\_OUTPUT.PUT\_LINE('таблицы очищены');

 COMMIT;

 EXCEPTION

 WHEN OTHERS THEN

 dbms\_output.put\_line('ошибка при очистке таблиц');

 dbms\_output.put\_line('error code: '||SQLCODE);

 ROLLBACK;

 END seeder\_clear;

END kr\_pack;



заполняем таблицу

BEGIN

 kr\_pack.seeder\_add;

END



Вид таблицы игроков



Вид таблицы турниров



Вид таблицы связей



Добавим процедуру, которая должна удалить турниры с количеством участников, менее заданного в параметре

 PROCEDURE del\_low\_popup\_tournaments(cuant IN number) IS

 CURSOR all\_tournamets IS

 SELECT t\_num FROM tournaments;

 v\_t\_num number;

 number\_of\_participants number;

 BEGIN

 OPEN all\_tournamets;

 LOOP

 FETCH all\_tournamets INTO v\_t\_num;

 EXIT WHEN (all\_tournamets%NOTFOUND);

 -- считаем количество участников

 SELECT count(\*) INTO number\_of\_participants FROM list WHERE t\_num = v\_t\_num;

 IF number\_of\_participants < cuant THEN

 DELETE FROM list WHERE t\_num = v\_t\_num;

 DELETE FROM tournaments WHERE t\_num = v\_t\_num;

 DBMS\_OUTPUT.PUT\_LINE('турнир ' || v\_t\_num || ' удален, тк слишком мало участников');

 END IF;

 END LOOP;

 CLOSE all\_tournamets;

 COMMIT;

 EXCEPTION

 WHEN OTHERS THEN

 dbms\_output.put\_line('ошибка при удалении турниров');

 dbms\_output.put\_line('error code: '||SQLCODE);

 ROLLBACK;

 END del\_low\_popup\_tournaments;

удалим турниры с менее чем 5 участниками

BEGIN

 kr\_pack.del\_low\_popup\_tournaments(5);

END

результат



--Создаем представление которое выводит турниры и имена игроков

CREATE OR REPLACE VIEW tour\_player AS

 SELECT \* FROM (tournaments JOIN list USING (t\_num)) JOIN players USING (p\_num);

**--все строки, а нужно часть**



Включим в пакет еще одну процедуру, которая, используя представление, выводит турниры и имена их участников; в вывод попадают турниры с количеством участников, заданном в параметре

PROCEDURE show\_tournaments(cuant IN number) IS

 CURSOR all\_players\_in\_tournament(toutnament\_num varchar2) IS

 SELECT p\_name, t\_name FROM tour\_player WHERE t\_num = toutnament\_num ORDER BY t\_name;

 CURSOR all\_tournamets IS

 SELECT t\_num FROM tournaments;

 v\_t\_num number;

 number\_of\_participants number;

 v\_p\_name players.p\_name%TYPE;

 v\_t\_name tournaments.t\_name%TYPE;

 BEGIN

 OPEN all\_tournamets;

 LOOP

 FETCH all\_tournamets INTO v\_t\_num;

 EXIT WHEN (all\_tournamets%NOTFOUND);

 -- считаем количество участников

 SELECT count(\*) INTO number\_of\_participants FROM list WHERE t\_num = v\_t\_num;

 IF number\_of\_participants = cuant THEN

 dbms\_output.put\_line('в турнире : ' || v\_t\_num || ' ' || number\_of\_participants || ' участников');

 OPEN all\_players\_in\_tournament(v\_t\_num);

 LOOP

 FETCH all\_players\_in\_tournament INTO v\_p\_name, v\_t\_name;

 EXIT WHEN (all\_players\_in\_tournament%NOTFOUND);

 dbms\_output.put\_line('игрок: ' || v\_p\_name || ', турнир: ' || v\_t\_name);

 END LOOP;

 CLOSE all\_players\_in\_tournament;

 END IF; **--открытие курсора и обращение к БД в цикле**

 END LOOP;

 CLOSE all\_tournamets;

 COMMIT;

 EXCEPTION

 WHEN OTHERS THEN

 dbms\_output.put\_line('ошибка при выводе турниров и игроков');

 dbms\_output.put\_line('error code: '||SQLCODE);

 ROLLBACK;

 END show\_tournaments;

Выведем информацию по турнирам с 5 участниками

BEGIN

 kr\_pack.show\_tournaments(5);

END

результат:

