

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

Кафедра прикладной математики и кибернетики

Сетевые базы данных
Расчетно-Графическое Задание
Вариант №9

Выполнил:
Студент IV курса ИВТ,

Работу проверил:
Старший преподаватель ПМиК
Грязнов Николай Геннадьевич

Новосибирск 2020 г.

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 и привилегию 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.

Поместить в таблицы информацию об Университетах и Специальностях. Каждый университет готовит специалистов по нескольким специальностям. Процедура должна удалять университет из таблицы. Триггер должен регистрировать удаление, сохраняя удаленную запись и указывая время выполнения операции. Включить в пакет еще одну процедуру, которая выводит два университета с максимальным числом специальностей. Выборку данных производить в коллекцию типа VARRAY.

2. Описание работы

В результате работы были созданы две таблицы `university` и `speciality`. Первая таблица содержит поля `univ_id`, `uname`, где `univ_id` – является `primary key` (первичным ключом). Вторая таблица содержит поля `spec_id`, `sname`, `univid_val`, где `spec_id` – первичный ключ, а `univid_val` – внешний ключ, который ссылается на ID университета из первой таблицы. Для правильного распределения идентификаторов, были созданы две последовательности `uid_seq` – для университетов и `sid_seq` – для специальностей.

Первый пакет tablePack содержит в себе процедуры для добавления и удаления данных из таблицы.

Триггер del_trigger заносит данные об удаленном университете – тип операции, название удаленного университета и время выполнения в новую таблицу с записями удалений delete_register, которая состоит из полей: command_name, deleted_record и time_of_event.

Представление my_view отображает университеты, и связанные с ними специальности:

select * from my_view			
UNIV_ID	UNAME	SPEC_ID	SNAME
1	SibSUTIS	1	IVT
1	SibSUTIS	2	MRM
1	SibSUTIS	3	MTS
1	SibSUTIS	4	AVS
2	NSU	5	IVT
2	NSU	6	Economics
2	NSU	7	MassMedia
3	NSPU	8	IVT
3	NSPU	9	English
3	NSPU	10	PhysMath
3	NSPU	11	French
4	NSTU	12	IVT
4	NSTU	13	English
4	NSTU	14	Economics
4	NSTU	15	MTS
5	MSU	16	IVT
5	MSU	17	Economics
5	MSU	18	MassMedia
5	MSU	19	Mec.-Math
5	MSU	20	Physics
5	MSU	21	Chemical
6	MSTU	22	IVT
6	MSTU	23	English
6	MSTU	24	Economics
6	MSTU	25	PhysMath
6	MSTU	26	MTS
7	MSPU	27	IVT
7	MSPU	28	English
7	MSPU	29	PhysMath
7	MSPU	30	French
30 rows selected. 0.03 seconds			

Далее был реализован второй пакет, `pcg_2`, содержащий в себе две процедуры и один тип `varchar`. Тип `varchar` из 2 элементов необходим для использования в процедуре `view_top2`, в переменную данного объявленного типа будут записываться названия первых двух университетов, с наибольшим количеством специальностей. И затем значения этой переменной будут выводиться на экран. В случае какой-либо ошибки (например, если данные не были получены из БД) существует исключение.

```
Statement processed. 0.04 seconds

begin      pcg_2.view_top2; end;

First max: MSU
Second max: MSTU

Statement processed. 0.02 seconds
```

Вторая процедура `del_university` принимает в себя аргумент – название университета для удаления. Сначала проверяется есть ли такой университет в таблице, в случае ошибки будет вызвано исключение. Иначе данные об этом университете удаляются из БД.

Попробуем удалить университет «НГУ»:

```
Package Body created. 0.03 seconds

begin      pcg_2.del_university('NSU'); end;

Deleted successfully.

Statement processed. 0.02 seconds
```

, так же триггером была добавлена советующая запись:

Results Explain Describe Saved SQL History		
COMMAND_NAME	DELETED_RECORD	TIME_OF_EVENT
Delete	NSU	23-MAR-21 08.48.43.251304 AM +00:00
1 rows returned in 0.01 seconds Download		

Если попытаться удалить несуществующий университет – выведется соответствующее сообщение:

```
Package Body created. 0.03 seconds

begin      pcg_2.del_university('123125345');      pcg_2.view_top2; end;

Wrong university to delete
```

Всем пользователям были предоставлены привилегии на использование пакета `pcg_2` и представления `my_view`.

3. Скриншоты

SPECIALITY			
Table	Data	Indexes	Model
Constraints	Grants	Statistics	UI Defaults
Triggers	Dependencies	SQL	REST
Sample Queries			
Query	Count Rows	Insert Row	Load Data
EDIT	SPEC_ID	SNAME	UNIVID_VAL
	1	IVT	1
	2	MRM	1
	3	MTS	1
	4	AVS	1
	5	IVT	2
	6	Economics	2
	7	MassMedia	2
	8	IVT	3
	9	English	3
	10	PhysMath	3
	11	French	3
	12	IVT	4
	13	English	4
	14	Economics	4
	15	MTS	4

Рис.1 – таблица speciality.

EDIT	SPEC_ID	SNAME	UNIVID_VAL
	16	IVT	5
	17	Economics	5
	18	MassMedia	5
	19	Mec.-Math	5
	20	Physics	5
	21	Chemical	5
	22	IVT	6
	23	English	6
	24	Economics	6
	25	PhysMath	6
	26	MTS	6
	27	IVT	7
	28	English	7
	29	PhysMath	7
	30	French	7

Рис.2 – продолжение таблицы speciality.

UNIVERSITY		
Table	Data	Indexes
Model	Constraints	Grants
Statistics	UI Defaults	Triggers
Dependencies		
Query	Count Rows	Insert Row
Load Data		
EDIT	UNIV_ID	UNAME
	1	SibSUTIS
	2	NSU
	3	NSPU
	4	NSTU
	5	MSU
	6	MSTU
	7	MSPU

Рис.3 – таблица university.

4. Исходный код

```
drop sequence uid_seq;
drop sequence sid_seq;

create sequence uid_seq start with 1 increment by 1;
create sequence sid_seq start with 1 increment by 1;

drop table speciality;
drop table university;

create table university (
    univ_id number(3) not null,
    uname varchar2(50) not null,

    constraint univid_uk unique(univ_id),
    constraint uname primary key(uname)
);

create table speciality (
    spec_id number(3) not null,
    sname varchar2(50) not null,
    univid_val number(3) not null,

    constraint specid_pk primary key(spec_id),
    constraint univid_fk foreign key(univid_val) references university(univ_id)
);

create or replace package tablePack is
    procedure add_data;
    procedure clear_data;
end;
/
create or replace package body tablePack is
    procedure add_data is
    begin
        insert into university values (uid_seq.nextval, 'SibSUTIS');
        insert into speciality values (sid_seq.nextval, 'IVT', uid_seq.currval);
        insert into speciality values (sid_seq.nextval, 'MRM', uid_seq.currval);
        insert into speciality values (sid_seq.nextval, 'MTS', uid_seq.currval);
        insert into speciality values (sid_seq.nextval, 'AVS', uid_seq.currval);

        insert into university values (uid_seq.nextval, 'NSU');
        insert into speciality values (sid_seq.nextval, 'IVT', uid_seq.currval);
        insert into speciality values (sid_seq.nextval, 'Economics', uid_seq.curr
val);
        insert into speciality values (sid_seq.nextval, 'MassMedia', uid_seq.curr
val);

        insert into university values (uid_seq.nextval, 'NSPU');
        insert into speciality values (sid_seq.nextval, 'IVT', uid_seq.currval);
```

```

        insert into speciality values (sid_seq.nextval, 'English', uid_seq.currval);
    1);
        insert into speciality values (sid_seq.nextval, 'PhysMath', uid_seq.currval);
    al);
        insert into speciality values (sid_seq.nextval, 'French', uid_seq.currval);
    );

        insert into university values (uid_seq.nextval, 'NSTU');
        insert into speciality values (sid_seq.nextval, 'IVT', uid_seq.currval);
        insert into speciality values (sid_seq.nextval, 'English', uid_seq.currval);
    1);
        insert into speciality values (sid_seq.nextval, 'Economics', uid_seq.currval);
    val);
        insert into speciality values (sid_seq.nextval, 'MTS', uid_seq.currval);

        insert into university values (uid_seq.nextval, 'MSU');
        insert into speciality values (sid_seq.nextval, 'IVT', uid_seq.currval);
        insert into speciality values (sid_seq.nextval, 'Economics', uid_seq.currval);
    val);
        insert into speciality values (sid_seq.nextval, 'MassMedia', uid_seq.currval);
    val);
        insert into speciality values (sid_seq.nextval, 'Mec.-
Math', uid_seq.currval);
        insert into speciality values (sid_seq.nextval, 'Physics', uid_seq.currval);
    1);
        insert into speciality values (sid_seq.nextval, 'Chemical', uid_seq.currval);
    al);

        insert into university values (uid_seq.nextval, 'MSTU');
        insert into speciality values (sid_seq.nextval, 'IVT', uid_seq.currval);
        insert into speciality values (sid_seq.nextval, 'English', uid_seq.currval);
    1);
        insert into speciality values (sid_seq.nextval, 'Economics', uid_seq.currval);
    val);
        insert into speciality values (sid_seq.nextval, 'PhysMath', uid_seq.currval);
    al);
        insert into speciality values (sid_seq.nextval, 'MTS', uid_seq.currval);

        insert into university values (uid_seq.nextval, 'MSPU');
        insert into speciality values (sid_seq.nextval, 'IVT', uid_seq.currval);
        insert into speciality values (sid_seq.nextval, 'English', uid_seq.currval);
    1);
        insert into speciality values (sid_seq.nextval, 'PhysMath', uid_seq.currval);
    al);
        insert into speciality values (sid_seq.nextval, 'French', uid_seq.currval);
    );
end add_data;

procedure clear_data is
begin
    delete from university;

```

```

        delete from speciality;
        execute immediate 'drop sequence uid_seq';
        execute immediate 'drop sequence sid_seq';
    end clear_data;
end;
/
begin
    tablePack.add_data;
end;
/

drop table delete_register;
create table delete_register (
    command_name varchar2(20) not null,
    deleted_record varchar2(50) not null,
    time_of_event TIMESTAMP WITH TIME ZONE
);
commit;

create or replace trigger del_trigger
    before delete on university
    for each row
begin
    if deleting then
        insert into delete_register(command_name, deleted_record, time_of_event)
            values ('Delete', :old.uname, CURRENT_TIMESTAMP);
    end if;
end;
/

create or replace view my_view as
    select university.univ_id, university.uname, speciality.spec_id, speciality.s
name
    from university, speciality where university.univ_id = speciality.univid_val;
select * from my_view;

create or replace package pcg_2 is
    type top2 is varray(2) of varchar2(50);
    procedure del_university(to_delete in varchar2 default null);
    procedure view_top2;
end pcg_2;
/

create or replace package body pcg_2 is
    type t_top2 is varray(2) of varchar2(50);
    p_top2 t_top2 := t_top2('first', 'second');

    procedure del_university(to_delete in varchar2 default null) is
        tmp varchar2(50);
    begin
        select uname into tmp from university where uname = to_delete;

```



```

        delete from speciality where univid_val in (select univ_id from universit
y where uname = to_delete);
        delete from university where uname = to_delete;
        commit;

        exception when no_data_found then
            dbms_output.put_line('Wrong university to delete');

    end del_university;

    procedure view_top2 is
        u_first_max varchar2(50);
        u_second_max varchar2(50);
        no_fetch exception;
    begin
        select uname into u_first_max from (select uname, (select count(*) from s
peciality where univ_id = univid_val) as num
            from university order by num desc) where rownum = 1;
        select uname into u_second_max from (select a.*, rownum rnum from (select
uname, (select count(*) from speciality where univ_id = univid_val) as num
            from university order by num desc) a where rownum <= 2) where rnum =
2;

        if u_first_max = '' or u_second_max = '' then
            raise no_fetch;
        end if;

        p_top2(1) := u_first_max;
        p_top2(2) := u_second_max;

        dbms_output.put_line('First min: ' || p_top2(1));
        dbms_output.put_line('Second min: ' || p_top2(2));

        exception when no_fetch then
            dbms_output.put_line('No data was fetched');
            raise;

    end view_top2;
end pcg_2;
/

begin
    pcg_2.del_university('123125345');
end;
/
begin
    pcg_2.view_top2;
end;

```