Министерство связи и массовых коммуникаций Российской Федерации (Минкомсвязь РФ)

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(ФГБОУ ВО «СибГУТИ»)

Кафедра прикладной математики и кибернетики (ПМ и К)

Расчетно-графическое задание по дисциплине «Сетевые базы данных»

Вариант №8.

Выполнил**:** студент группы ИП-413

Горелов Р.С.

Проверил**:** старший преподаватель кафедры ПМ и К

Грязнов Н.Г.

Новосибирск,

2018 г.

**Оглавление**

1. Постановка задачи ………………………………………………….3
2. Листинг ……………………………………………………………...5
   1. Создание таблиц ……………………………………………..5
   2. Создание представления …………………………………….6
   3. Создание триггера …………………………………………...7
   4. Раздача прав ………………………………………………….8
   5. Создание пакетов вместе с процедурами и функциями …..9
3. Результаты …………………………………………………………14
4. Вывод ………………………………………………………………17
5. **Постановка задачи**

В рамка данного РГЗ было предусмотрено следующее задание:

Создать две таблицы, каждая из которых должна иметь первичный ключ и, по 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.

Личный вариант задавал следующие условия:

* Таблицы должны содержать данные о Ресторанах и Технологическом оборудовании. Каждый ресторан может закупать оборудование разных типов.
* Процедура должна изменять на 10% стоимость оборудования для указанного ресторана.
* Триггер должен разрешать изменение до 20-числа.
* Включить в пакет две процедуры, одна выбирает из БД в запись, созданную на основе курсора, количество и суммарную стоимость покупок для заданного ресторана. Другая выводит на экран содержимое записи, которую получает как параметр.

1. **Листинг**
   1. Создание таблиц

drop table RESTAURANTS CASCADE CONSTRAINTS;

drop table TECH\_EQUIP CASCADE CONSTRAINTS;

-- создание таблицы банков

create table RESTAURANTS (

ID NUMBER NOT NULL,

REST\_NAME VARCHAR2(60) NOT NULL

);

-- добавление индекса и первичного ключа

create unique index REST\_PK\_ID ON RESTAURANTS (ID);

alter table RESTAURANTS add constraint D\_REST\_PK\_ID primary key (ID) enable;

-- создание таблицы Банкоматов

create table TECH\_EQUIP (

ID NUMBER NOT NULL,

REST\_ID NUMBER,

EQUIP\_NAME VARCHAR2(50) NOT NULL,

N\_COST NUMBER

);

-- также добавляем индекс и ключи для таблицы Банкоматов

create unique index UNIQUE\_RECORD ON TECH\_EQUIP (ID);

alter table TECH\_EQUIP add constraint D\_EQUIP\_PK\_ID primary key (ID) enable;

alter table TECH\_EQUIP add constraint D\_EQUIP\_FK\_PID foreign key (REST\_ID) references RESTAURANTS (ID) enable;

* 1. Создание представления

create or replace view RESTAURANTS\_AND\_QUIPMENT as

select r.\*,

te.EQUIP\_NAME

from RESTAURANTS r

join TECH\_EQUIP te on r.ID = te.REST\_ID

where r.REST\_NAME <> 'RAGU';

* 1. Создание триггера

create or replace trigger ALLOW\_CHANGE\_EQUIP\_COST

before update on TECH\_EQUIP

for each row

declare

date\_error exception;

n\_day NUMBER;

begin

select extract(day from sysdate)

into n\_day

from dual;

if n\_day >= 20 then

raise date\_error;

end if;

exception

when date\_error then

dbms\_output.put\_line('Изменять стоимость возможно только до 20 числа.');

when others then

raise\_application\_error(-20001, 'Возникла следующая ошибка со следующим кодом-' || SQLCODE || ' Текст ошибки - ' || SQLERRM);

end;

* 1. Раздача прав

grant select on RESTAURANTS\_AND\_QUIPMENT to PUBLIC;

grant execute on PKG\_CHANGE\_DATA.INCREASE\_COST to UP1;

* 1. Создание пакетов вместе с процедурами и функциями
* Листинг первого пакета:

drop sequence GEN\_ID;

create sequence GEN\_ID increment by 1

start with 1;

/

create or replace package PKG\_CHANGE\_TABLES is

procedure ADD;

procedure DEL;

end PKG\_CHANGE\_TABLES;

/

create or replace package body PKG\_CHANGE\_TABLES is

temp1 NUMBER;

temp2 NUMBER;

procedure ADD is

begin

-- add records to RESTAURANTS

insert into RESTAURANTS

values (GEN\_ID.NEXTVAL, 'Peoples');

temp1 := GEN\_ID.CURRVAL;

insert into RESTAURANTS

values (GEN\_ID.NEXTVAL, 'Шашлыкофф');

insert into RESTAURANTS

values (GEN\_ID.NEXTVAL, 'Дрова Мука');

insert into RESTAURANTS

values (GEN\_ID.NEXTVAL, 'Аджикинежаль');

insert into RESTAURANTS

values (GEN\_ID.NEXTVAL, 'RAGU');

insert into RESTAURANTS

values (GEN\_ID.NEXTVAL, 'Парк кафе');

insert into RESTAURANTS

values (GEN\_ID.NEXTVAL, 'Небо');

temp2 := GEN\_ID.CURRVAL;

-- add records to TECH\_EQUIP

insert into TECH\_EQUIP

values (GEN\_ID.NEXTVAL, trunc(dbms\_random.value(temp1, temp2)), 'Духовой шкаф Bosh', trunc(dbms\_random.value(20000, 500000)));

insert into TECH\_EQUIP

values (GEN\_ID.NEXTVAL, trunc(dbms\_random.value(temp1, temp2)), 'Гриль Webber', trunc(dbms\_random.value(20000, 500000)));

insert into TECH\_EQUIP

values (GEN\_ID.NEXTVAL, trunc(dbms\_random.value(temp1, temp2)), 'Тандыр', trunc(dbms\_random.value(20000, 500000)));

insert into TECH\_EQUIP

values (GEN\_ID.NEXTVAL, trunc(dbms\_random.value(temp1, temp2)), 'Посудомоечная машина Bosh', trunc(dbms\_random.value(20000, 500000)));

insert into TECH\_EQUIP

values (GEN\_ID.NEXTVAL, trunc(dbms\_random.value(temp1, temp2)), 'Каменная печь', trunc(dbms\_random.value(20000, 500000)));

insert into TECH\_EQUIP

values (GEN\_ID.NEXTVAL, trunc(dbms\_random.value(temp1, temp2)), 'Стойки для разлива пива', trunc(dbms\_random.value(20000, 500000)));

insert into TECH\_EQUIP

values (GEN\_ID.NEXTVAL, trunc(dbms\_random.value(temp1, temp2)), 'Плита', trunc(dbms\_random.value(20000, 500000)));

insert into TECH\_EQUIP

values (GEN\_ID.NEXTVAL, trunc(dbms\_random.value(temp1, temp2)), 'Кухонный комбайн', trunc(dbms\_random.value(20000, 500000)));

commit;

exception when others then

raise\_application\_error(-20001, 'Возникла следующая ошибка со следующим кодом-' || SQLCODE || ' Текст ошибки - ' || SQLERRM);

end ADD;

procedure DEL is

begin

delete from TECH\_EQUIP;

delete from RESTAURANTS;

commit;

dbms\_output.put\_line('Таблицы RESTAURANTS и TECH\_EQUIP очищены.');

exception when others then

raise\_application\_error(-20001, 'Возникла следующая ошибка со следующим кодом-' || SQLCODE || ' Текст ошибки - ' || SQLERRM);

end DEL;

end PKG\_CHANGE\_TABLES;

* Создание второго пакета:

create or replace package PKG\_CHANGE\_DATA is

procedure ADD;

procedure DEL;

procedure INCREASE\_COST (REST VARCHAR2);

procedure CREATE\_REC\_REST (REST in VARCHAR2);

end PKG\_CHANGE\_DATA;

/

create or replace package body PKG\_CHANGE\_DATA is

procedure ADD is

begin

PKG\_CHANGE\_TABLES.ADD();

end ADD;

procedure DEL is

begin

PKG\_CHANGE\_TABLES.DEL();

end DEL;

-----------------------------------------------------

procedure INCREASE\_COST(REST VARCHAR2) is

begin

update TECH\_EQUIP

set N\_COST = N\_COST \* 0.1 + N\_COST

where REST\_ID = (select r.ID

from RESTAURANTS r

where r.REST\_NAME = REST);

commit;

exception when others then

raise\_application\_error(-20001, 'Возникла следующая ошибка со следующим кодом -' || SQLCODE || ' Текст ошибки - ' || SQLERRM);

end INCREASE\_COST;

-------------------------------------------------------

procedure CREATE\_REC\_REST(REST in VARCHAR2) is

-- создание курсора

cursor res is

select r.REST\_NAME,

count(te.EQUIP\_NAME) cnt,

sum(te.N\_COST) summ

from RESTAURANTS r

join TECH\_EQUIP te on r.ID = TE.REST\_ID

where r.REST\_NAME = REST

group by r.REST\_NAME;

-- создание записи на основе курсора

r\_rest res%ROWTYPE;

data\_error exception;

-- дополнительная процедура для вывода записи

procedure print\_rest (rest res%ROWTYPE) is

begin

dbms\_output.put\_line('Ресторна - ' || rest.REST\_NAME || ' количество оборудования - ' || rest.cnt || ' общая стоимость - ' || rest.summ);

end print\_rest;

begin

open res;

fetch res into r\_rest;

if res%NOTFOUND then

raise data\_error;

end if;

close res;

print\_rest(r\_rest);

exception

when data\_error then

dbms\_output.put\_line('Для данного ресторана ничего не найден. Возможно вы неверное указали название.');

when others then

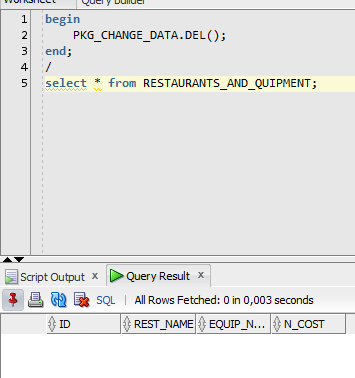
raise\_application\_error(-20001, 'Возникла следующая ошибка со следующим кодом -' || SQLCODE || ' Текст ошибки - ' || SQLERRM);

end CREATE\_REC\_REST;

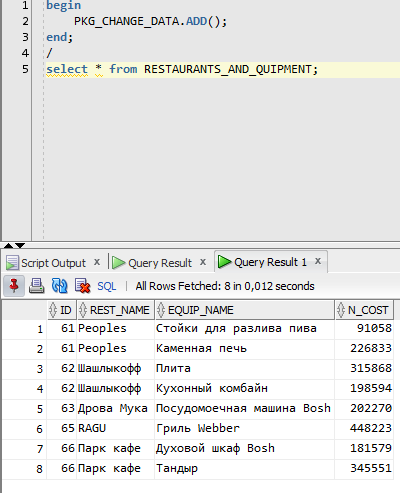
end PKG\_CHANGE\_DATA;

1. **Результаты**

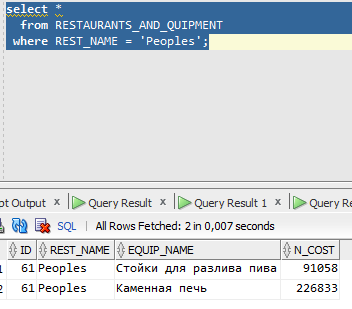
Вывод представления после очистки таблиц.



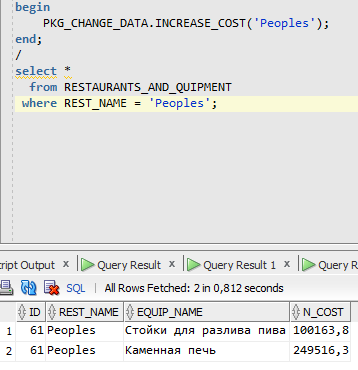
После заполнения таблиц выводим представление, чтобы удостовериться в том, что данные были добавлены.



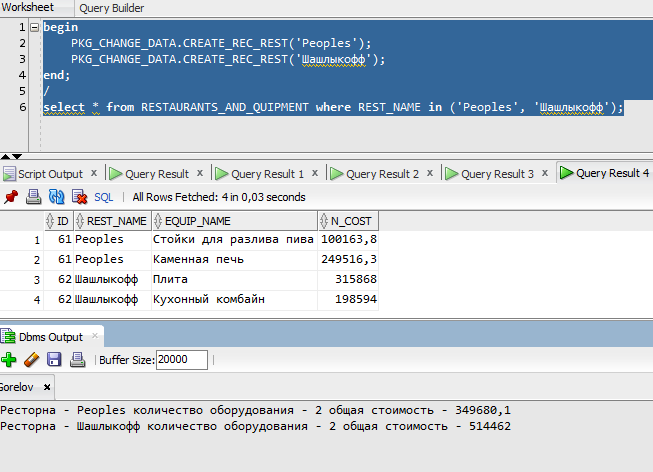
Стоимость оборудования ресторана до увеличения его стоимости на 10%.



После увеличения стоимости.



Вывод информации о ресторане (количество и суммарную стоимость покупок для заданного ресторана).



1. Вывод

Курс «Сетевых баз данных» предполагал знакомство с базой данных Oracle, а также с процедурным расширением языка sql – pl/sql. Данное расчетно – графическое задание являлось показателем изученного материала из данного курса. Материал включал такие темы, как создание хранимых процедур и функций, триггеров, создание пакетов и включение в них процедур и функций, создание пользователей и раздача прав для них и т.п

В ходе проделанной работы были изучены ранее упоминавшиеся механизмы, а также основы самого язык pl/sql: различные условные конструкции, объявление переменных и курсоров, создание коллекций на основе них и т.д