Федеральное агентство связи

Сибирский Государственный Университет Телекоммуникаций и Информатики

**Межрегиональный центр переподготовки специалистов**

# Курсовая работа

# По дисциплине: «Сетевые базы данных»

**Выполнил**: Шадт Р.А.

**Группа**: ПБТ-41

**Вариант:** 5

**Проверил**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**--в процедуре deleteFirm лишнее обращение к БД**

**Оценка – «неудовлетворительно»**

Новосибирск, 2017 г

**ЗАДАНИЕ НА КУРСОВОЕ ПРОЕКТИРОВАНИЕ**

**1. Рекомендации по выполнению курсовой работы**

Прежде, чем приступать к выполнению курсовой работы, необходимо выполнить все лабораторные работы данного курса.

Курсовая работа состоит из задания, которое выбирается согласно своему варианту. Если работа выполнена в соответствии с заданием, в ней нет грубых ошибок, то вы получаете положительную оценку по курсовой работе. В противном случае работа отправляется вам на доработку.

**!!! Номер варианта курсовой работы определяется по последней цифре пароля**

Работу следует выполнять в среде программирования Oracle 10g XE.

**2. Требования к оформлению работы**

Необходимо представить следующее:

1. Текст задания ( с подставленными данными своего варианта);
2. Текст программы (текстовый файл или файл с расширением .sql) ;
3. Результат работы программы (т.е. все данные, которые выводятся на экран, – текстовый файл или файл, содержащий скрин-шоты панели вывода среды разработки Oracle XE).

**3. Общая часть задания**

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.

**4. Варианты задания на выполнение контрольной работы**

**5.** Поместить в таблицу информацию о Компьютерных фирмах и Заказчиках. Компьютерная фирма может обслуживать несколько заказчиков. Фирма имеет название, тип поставляемых компьютеров и годовой бюджет. Заказчик имеет название и количество приобретаемых компьютеров. Процедура должна удалять из таблицы указанную фирму; название фирмы передается в процедуру как параметр. Триггер должен регистрировать операцию удаления с указнием пользователя, даты и времени удаления и данных о фирме. Включить в пакет еще одну процедуру, выводящую фирмы, количество заказанных компьютеров у которых находится в диапазоне, заданном в параметрах.

# Таблица фирм

|  |  |  |  |
| --- | --- | --- | --- |
| **fNum** | **fName** | **fProductType** | **fMoney** |
| 1001 | Asus | PC | 9900.500 |
| 1002 | Dell | PC | 960.500 |
| 1003 | Dns | Hardware | 6050.300 |
| 1004 | HP | Notebooks | 2250.000 |
| 1005 | MSI | Devices | 1450.500 |

Первичный ключ fNum.

# Таблица заказчиков

|  |  |  |  |
| --- | --- | --- | --- |
| **cNum** | **cName** | **cProductCount** | **fNum** |
| 1001 | Jerry Miller | 56 | 1001 |
| 1002 | Samanta Greys | 7 | 1001 |
| 1003 | Anna Fray | 25 | 1001 |
| 1004 | Kelvin Kline | 41 | 1002 |
| 1005 | Jeremy Masking | 145 | 1002 |
| 1006 | Mark Twen | 3 | 1003 |
| 1007 | Bop Marly | 1 | 1003 |
| 1008 | Nina Richi | 74 | 1004 |
| 1009 | Stanislav Zhukov | 19 | 1004 |
| 1010 | Tom Polaris | 58 | 1003 |
| 1011 | Gudvin Subox | 300 | 1001 |
| 1012 | Antonio Banderas | 159 | 1002 |
| 1013 | Pavel Durov | 951 | 1001 |
| 1014 | Carlos Kinich | 137 | 1001 |
| 1015 | Viper Loginov | 11 | 1005 |

Первичный ключ cNum. Внешний fNum.

Сценарий создания таблиц

CREATE TABLE firm

 (fNum NUMBER(4),

 fName VARCHAR2(50) NOT NULL,

 fProductType VARCHAR2(50) NOT NULL,

 fMoney NUMBER(7,3) NOT NULL);

CREATE TABLE cust

 (cNum NUMBER(4) ,

 cName VARCHAR2(50) NOT NULL,

 cProductCount NUMBER(7) NOT NULL,

 fNum NUMBER(4));

CREATE TABLE logs

 (userName VARCHAR2(50) NOT NULL,

 operationDate VARCHAR(20) NOT NULL,

 operationType VARCHAR2(50) NOT NULL,

 firmName VARCHAR2(50) NOT NULL,

 firmNum NUMBER(4) NOT NULL);

ALTER TABLE firm

 ADD (CONSTRAINT firm\_pk\_snum PRIMARY KEY (fnum));

ALTER TABLE cust

 ADD (CONSTRAINT cust\_pk\_cnum PRIMARY KEY (cnum),

 CONSTRAINT cust\_fk\_snum FOREIGN KEY (fnum)

 REFERENCES firm(fnum));

В таблицу logs будет добавляться информация об удаляемой фирме, регистрируемая с помощью триггера.

Пакет добавления и удаления записей

CREATE OR REPLACE PACKAGE firm\_tools\_pack

AS

PROCEDURE fillByDefault;

PROCEDURE clear;

END firm\_tools\_pack;

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

CREATE OR REPLACE PACKAGE BODY firm\_tools\_pack

AS

PROCEDURE fillByDefault IS

BEGIN

 INSERT INTO firm

 VALUES (1001, 'Asus', 'PC', 9900.500);

 INSERT INTO firm

 VALUES (1002, Dell', 'PC', 960.500);

...

 INSERT INTO cust

 VALUES (1001, ' Jerry Miller ', 56, 1001);

 INSERT INTO cust

...

 COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Data filled successfully');

EXCEPTION

 WHEN OTHERS THEN

 DBMS\_OUTPUT.PUT\_LINE('Ошибка!');

END fillByDefault;

PROCEDURE clear IS

BEGIN

 DELETE FROM cust;

 DELETE FROM firm;

 COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Data cleared successfully');

END clear;

END firm\_tools\_pack;

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

BEGIN

firm\_tools\_pack.fillByDefault();

END;

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

BEGIN

firm\_tools\_pack.clear();

END;

Представление для связанных таблиц

Отобразим информацию о заказчиках той компании, у которой заказано больше всего единиц товаров.

Для начала выведем вспомогательную информацию о сумме заказанных товаров для каждой компании:

SELECT SUM(cproductcount) SaleProducts, fnum FROM cust GROUP BY fnum ORDER BY fnum



Получим номер компании, у которой заказано больше всего товаров (1476 единиц):

CREATE OR REPLACE VIEW getFirmNumWithMaxSale

AS SELECT f FROM

 (SELECT SUM(cproductcount) s, fnum f FROM cust GROUP BY fnum)

 WHERE s =

 (SELECT MAX(s) FROM (SELECT SUM(cproductcount) s FROM cust GROUP BY fnum))



Отобразим данные о заказчиках полученной компании 1001:

SELECT \* FROM cust c, firm f WHERE c.fnum = f.fnum AND f.fnum =

(SELECT FROM getFirmNumWithMaxSale)



Отображение компаний с заданным диапазоном заказанных товаров

PROCEDURE getFirmsWith(minSaleProducts IN NUMBER,
                       maxSaleProducts IN NUMBER) IS
CURSOR firms IS
    SELECT firm.\*, SUM(cNum) sp FROM firm, cust WHERE firm.fnum = cust.fnum
    GROUP BY firm.fnum, fName, fProductType, fMoney, cust.fnum
    HAVING SUM(cNum) >= minSaleProducts AND

 SUM(cNum) <= maxSaleProducts;
BEGIN
FOR info IN firms
  LOOP
   DBMS\_OUTPUT.PUT\_LINE('Фирма №: '|| info.fnum ||
                       ', название: '|| info.fname ||
                       ', тип товара: '|| info.fProductType ||
                       ', количество: '|| info.sp ||
                       ', бюджет: '|| info.fMoney);
  END LOOP;
EXCEPTION
  WHEN OTHERS THEN
    DBMS\_OUTPUT.PUT\_LINE('Произошла ошибка.');
    ROLLBACK;
END;
/

BEGIN
    getFirmsWith(1000, 3021);
END;



Удаление компании

Создадим триггер:

CREATE OR REPLACE TRIGGER deleteFirmTrigger
    BEFORE DELETE ON firm
    FOR EACH ROW
DECLARE
BEGIN
INSERT INTO logs

 VALUES(USER, to\_char(sysdate,'dd.mm.yyyy hh24:mm'), 'Delete', :old.fname, :old.fnum);

END deleteFirmTrigger;

Внесем изменения в пакет:

CREATE OR REPLACE PACKAGE firm\_tools\_pack

AS

PROCEDURE fillByDefault;

PROCEDURE clear;

PROCEDURE getFirmsWith(minSaleProducts IN NUMBER,

 maxSaleProducts IN NUMBER);

PROCEDURE deleteFirm(name IN VARCHAR2);

END firm\_tools\_pack;

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

CREATE OR REPLACE PACKAGE BODY firm\_tools\_pack

AS

PROCEDURE deleteFirm(name IN VARCHAR2) IS
  deletingFirmNum NUMBER;
BEGIN
    SELECT fnum into deletingFirmNum FROM firm WHERE fname = name;
    DELETE FROM cust WHERE fnum = deletingFirmNum;
    DELETE FROM firm WHERE fnum = deletingFirmNum;
    COMMIT;

**--отдельная команда SELECT – здесь лишнее обращение к БД. Используйте подзапрос**
END;

/

Пример использования:

BEGIN
    firm\_tools\_pack.deleteFirm('Asus');

END;
/

Результаты:







Итоговый вид пакета обработки данных таблиц

CREATE OR REPLACE PACKAGE firm\_tools\_pack

AS

PROCEDURE fillByDefault;

PROCEDURE clear;

END firm\_tools\_pack;

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

CREATE OR REPLACE PACKAGE BODY firm\_tools\_pack

AS

PROCEDURE fillByDefault IS

BEGIN

 INSERT INTO firm

 VALUES (1001, 'Asus', 'PC', 9900.500);

 INSERT INTO firm

 VALUES (1002, Dell', 'PC', 960.500);

...

 INSERT INTO cust

 VALUES (1001, ' Jerry Miller ', 56, 1001);

 INSERT INTO cust

...

 COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Data filled successfully');

EXCEPTION

 WHEN OTHERS THEN

 DBMS\_OUTPUT.PUT\_LINE('Ошибка!');

END fillByDefault;

PROCEDURE clear IS

BEGIN

 DELETE FROM cust;

 DELETE FROM firm;

 COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Data cleared successfully');

END clear;

PROCEDURE getFirmsWith(minSaleProducts IN NUMBER,
                       maxSaleProducts IN NUMBER) IS
CURSOR firms IS
    SELECT firm.\*, SUM(cNum) sp FROM firm, cust WHERE firm.fnum = cust.fnum
    GROUP BY firm.fnum, fName, fProductType, fMoney, cust.fnum
    HAVING SUM(cNum) >= minSaleProducts AND

 SUM(cNum) <= maxSaleProducts;
BEGIN
FOR info IN firms
  LOOP
   DBMS\_OUTPUT.PUT\_LINE('Фирма №: '|| info.fnum ||
                       ', название: '|| info.fname ||
                       ', тип товара: '|| info.fProductType ||
                       ', количество: '|| info.sp ||
                       ', бюджет: '|| info.fMoney);
  END LOOP;
EXCEPTION
  WHEN OTHERS THEN
    DBMS\_OUTPUT.PUT\_LINE('Произошла ошибка.');
    ROLLBACK;
END;
PROCEDURE deleteFirm(name IN VARCHAR2) IS
  deletingFirmNum NUMBER;
BEGIN
    SELECT fnum into deletingFirmNum FROM firm WHERE fname = name;
    DELETE FROM cust WHERE fnum = deletingFirmNum;
    DELETE FROM firm WHERE fnum = deletingFirmNum;
    COMMIT;
END;