МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

федеральное государственное бюджетное образовательное учреждение высшего образования

**«Вологодский государственный университет»**

**Институт математики, естественных и компьютерных наук**

**Информатика и вычислительная техника**

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3,4,5**

Разработка хранимого кода. Триггеры и пакеты.

Дисциплина: «Базы данных»

Направление подготовки: 09.03.01. Информатика и вычислительная техника

|  |  |
| --- | --- |
| Руководитель | Ржеуцкая С.Ю. |
| Выполнили студенты | Пчелкина О.С. |
| Группа, курс | ВМ-31 |
| Дата сдачи | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Дата защиты | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись преподавателя)* |

Вологда

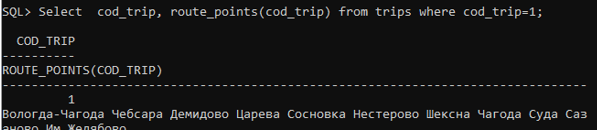
2022 г.

***Цель работы:*** получить первое представление о разработке и отладке хранимых процедур и функций, изучить на практике основные конструкции языка PL/SQL. Закрепить навыки разработки и отладки хранимого программного кода, освоить использование курсоров в хранимом коде. Освоить технику создания и использования триггеров в базе данных. Познакомиться с техникой создания пакетов.

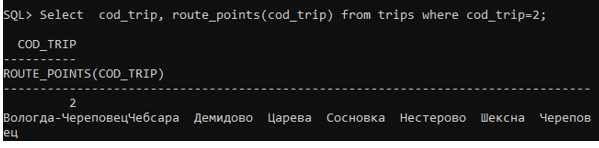
## Хранимый код для базы данных Автовокзал

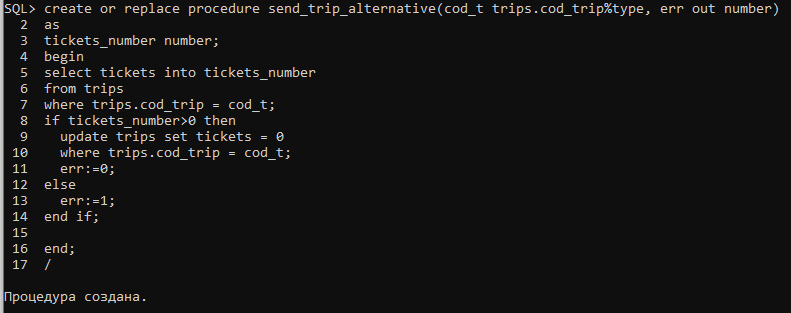
Напишите хранимую функцию (единственный входной параметр «код рейса»), которая возвращает строку, содержащую название маршрута, по которому идет этот рейс, и все остановки этого маршрута (например: Маршрут: список названий остановок через пробел или запятую).

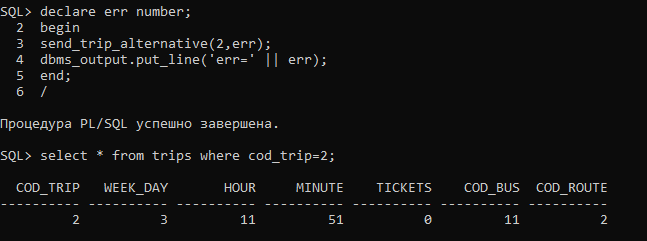
### CREATE or replace Function route\_points (codTrip trips.cod\_trip%type) RETURN varchar as cursor c is select points.name\_point from points, trips, points\_routes, routes where trips.cod\_trip = codTrip AND trips.cod\_route = points\_routes.cod\_route AND points.cod\_point = points\_routes.cod\_point GROUP BY points.name\_point; res varchar(10000); namePoint points.name\_point%type; BEGIN select routes.name\_route into res from trips, routes where trips.cod\_trip = codTrip AND trips.cod\_route = routes.cod\_route; open c; loop fetch c into namePoint; EXIT WHEN NOT c%found; res:=res||' '||namePoint; end loop; CLOSE c; RETURN res ; END;



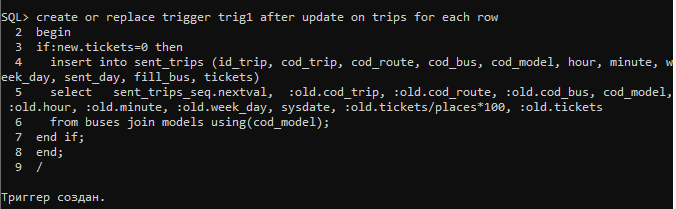
### CREATE or replace Function route\_points (codTrip trips.cod\_trip%type) RETURN varchar as res varchar(10000); BEGIN select routes.name\_route into res from trips, routes where trips.cod\_trip = codTrip AND trips.cod\_route = routes.cod\_route; for crec in (select points.name\_point from points, trips, points\_routes, routes where trips.cod\_trip = codTrip AND trips.cod\_route = points\_routes.cod\_route AND points.cod\_point = points\_routes.cod\_point GROUP BY points.name\_point) loop res:=res||crec.name\_point||' '; end loop; RETURN res ; END;

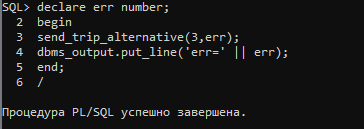


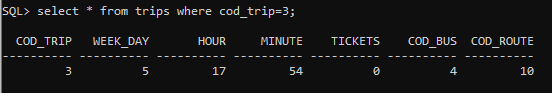
Напишите альтернативную хранимую процедуру «Отправка рейса» (параметр «код рейса»), которая проверяет, продан ли на рейс хотя бы один билет, и в этом случае обнуляет количество проданных билетов в таблице **trips,** но в архив она данные не заносит.

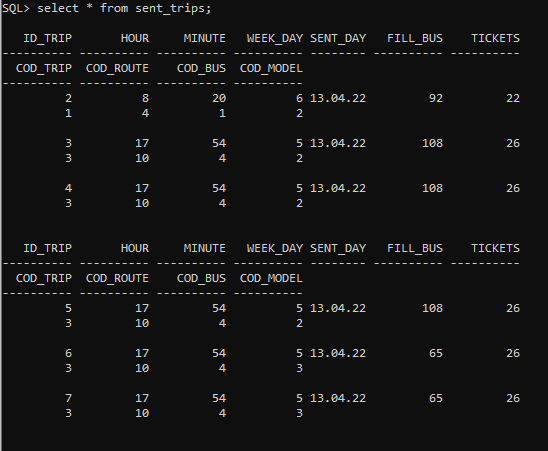


Напишите триггер на обновление таблицы **trips,** который перед обнулением количества проданных билетов (при работе процедуры «отправка рейса») заполнит строку таблицы **sent\_trips** из хранилища данными об отправленном рейсе. Не забудьте при реализации, что этот триггер должен срабатывать только на обнуление количества билетов при отправке рейса, и не реагировать на действия кассиров, продающих билеты (проверьте условие :new.tickets=0).

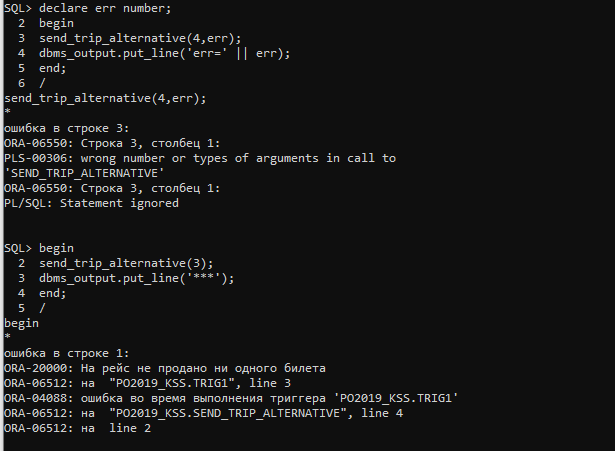




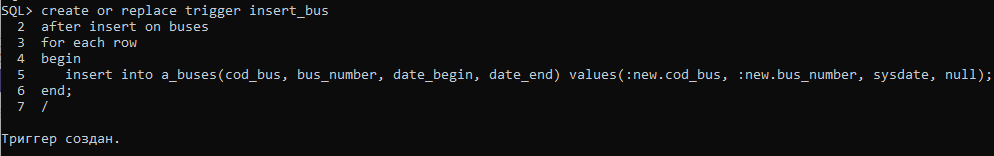


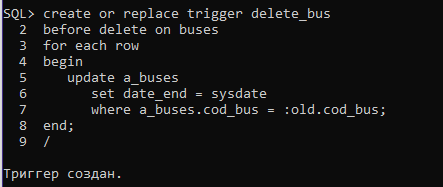


Дополните триггер обработкой исключительной ситуации, которая при обнаружении нулевого количества проданных билетов отменяет запрос на обнуление и возвращает клиенту сообщение «на этот рейс не продано ни одного билета».



В качестве ещё одного примера использования триггеров создайте триггеры для поддержки в актуальном состоянии справочников хранилища данных – это последняя часть файла Примеры хранимого кода (Лабораторная работа 3). В этом файле приведены примеры триггеров для синхронизации таблицы a\_trips с таблицей trips, а вы создайте аналогичные триггеры для таблицы buses, которые будут выполнять нужные изменения в a\_buses.





### *Хранимый код для индивидуальной базы данных*

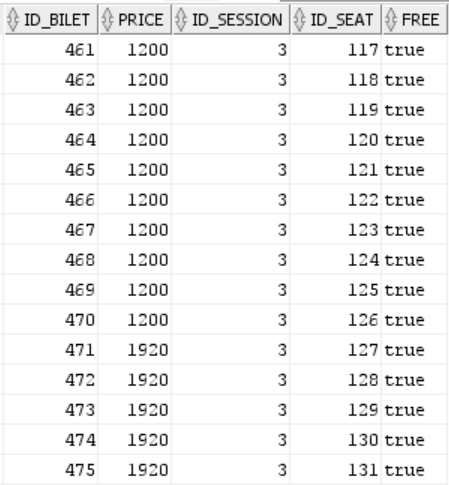
Функция для расчета стоимости билета (При добавлении билета в таблицу стоимость рассчитывается благодаря функции автоматически):

CREATE OR REPLACE FUNCTION GetPriceBilet  
 (price NUMBER, seatStatus CHAR, sessionFactor VARCHAR2, weekDaySession NUMBER, hourSession NUMBER,  
 factorSession FLOAT, factorWeekDay FLOAT, factorSeat FLOAT, factorTime FLOAT)

RETURN NUMBER  
IS priceBilet NUMBER;  
BEGIN

priceBilet:= price;  
IF sessionFactor = 'yes' THEN priceBilet := priceBilet \* factorSession; END IF;  
IF seatStatus = 'vip' THEN priceBilet := priceBilet \* factorSeat; END IF;  
IF weekDaySession > 5 THEN priceBilet := priceBilet \* factorWeekDay; END IF;  
IF hourSession > 17 THEN priceBilet := priceBilet \* factorTime; END IF;

RETURN priceBilet;

END;

Функция, возвращающая количество посещений на заданный фильм за каждый прокат:

Создаем тип array для возвращения функцией нескольких строк:

create type array  
as table of number;

create or replace function GetRoomOccupance(nameFilm VARCHAR2)

return array

PIPELINED

as

begin

for crec in (Select SUM(100 - sessions.number\_free\_seats) occ

from films, rents\_films, sessions where films.title\_film = nameFilm

AND films.id\_film = rents\_films.id\_film AND sessions.id\_rent = rents\_films.id\_rent Group By rents\_films.id\_rent)

loop

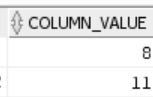
pipe row(crec.occ);

end loop;

return;

end;

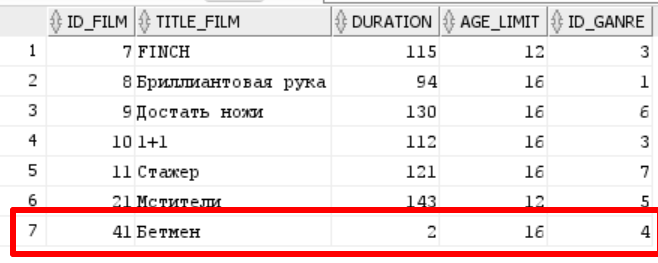
select \* from TABLE(GetRoomOccupanceTest('1+1'));



Триггер по заполнению таблицы ARCH\_FILMS при добавлении в таблицу FILMS нового фильма:

create or replace trigger insert\_film  
after insert on films  
for each row   
begin   
insert into arch\_films(id\_film, title\_film, duration,age\_limit, id\_ganre)  
values(:new.id\_film, :new.title\_film, :new.duration, :new.age\_limit, :new.id\_genres);  
end;

******Проверка:

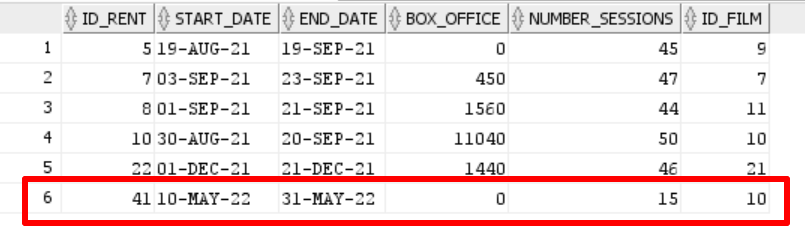
Данные таблицы ARCH\_FILM:

Триггер по заполнению таблицы ARCH\_RENT\_FILMS при удалении проката из таблицы RENTS\_FILMS:

******create or replace trigger insert\_rent  
before delete on rents\_films  
for each row   
begin   
insert into arch\_rent\_films(id\_rent, start\_date, end\_date, box\_office, number\_sessions, id\_film)  
values(:old.id\_rent, :old.start\_date, :old.end\_date, :old.box\_office, :old.number\_sessions, :old.id\_film);  
end;

Проверка:

******delete from RENTS\_FILMS WHERE ID\_RENT = 41;

Данные таблицы ARCH\_RENT\_FILMS:

***Выводы по работе***

В ходе лабораторной работы, получили первое представление о разработке и отладке хранимых процедур и функций, изучить на практике основные конструкции языка PL/SQL. Закрепили навыки разработки и отладки хранимого программного кода, освоили использование курсоров в хранимом коде. Освоили технику создания и использования триггеров в базе данных.