CITY

create table City

(

id\_city int primary key,

name\_city char(20) not null unique

)

create sequence city\_sequ start with 1;

create or replace trigger city\_bir

before insert on City

for each row

begin

select city\_sequ.nextval

into :new.id\_city

from dual;

end;

CLIENT

create table Client

(

telephone\_cl number(9) primary key,

fio varchar(30) not null,

sale number(2) default 0 check(sale in(0,5,10,15)),

id\_city int not null,

constraint fk\_city foreign key(id\_city) references City(id\_city)

);

CLIENT\_AUTO

create table Client\_auto

(

telephone\_cl number(9) not null,

password\_cl varchar (25) not null,

constraint fk\_client foreign key(telephone\_cl) references Client(telephone\_cl),

constraint pk\_client\_auto primary key(telephone\_cl)

)

TYPEWORKER

create table TypeWorker

(

typeWorker char(40) not null primary key,

salary number(5) not null

);

STAFF

create table Staff

(

id\_staffer int primary key,

fio\_staff varchar(30) not null,

telephone\_st number(9) not null unique,

birthday date not null,

typeWorker char(40) not null,

constraint fk\_typeW foreign key(typeWorker) references TypeWorker(typeWorker)

)

create sequence staff\_se start with 1;

create or replace trigger staff\_bir

before insert on Staff

for each row

begin

select staff\_se.nextval

into :new.id\_staffer

from dual;

end;

STAFF\_AUTO

create table Staff\_auto

(

telephone\_st number(9) not null,

password\_st varchar (25) not null,

constraint fk\_staff foreign key(telephone\_st) references Staff(telephone\_st),

constraint pk\_staff\_auto primary key(telephone\_st, password\_st)

)

SERVICE

create table Service

(

data timestamp with local time zone not null,

id\_staffer int not null,

telephone\_cl int not null,

constraint fk\_id\_sta foreign key(id\_staffer) references Staff(id\_staffer),

constraint fk\_c foreign key(telephone\_cl) references Client(telephone\_cl),

constraint pk\_se primary key(data, telephone\_cl)

)

REVIEW

create table Review

(

id\_review int primary key,

telephone\_cl number(9) not null,

note varchar(250) not null,

marks int default 5 check(marks in(1,2,3,4,5)) not null,

constraint fk\_client\_tel foreign key(telephone\_cl) references Client(telephone\_cl)

)

create sequence review\_sequ start with 1;

create or replace trigger review\_bir

before insert on Review

for each row

begin

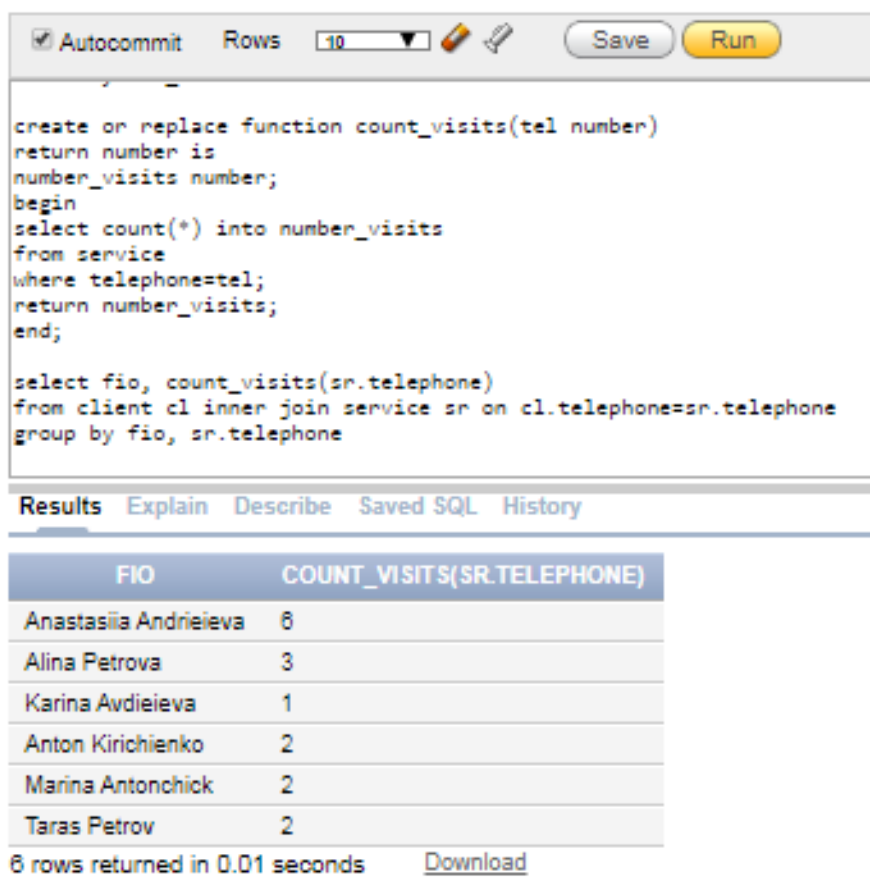
select review\_seq.nextval

into :new.id\_review

from dual;

end;

Функция, которая подсчитывает количество визитов каждого клиента:



Триггер 1, который при добавлении или обновлении таблицы Service проверяет введенные данные даты и телефон. Если введенная дата больше сегодняшней даты, то триггер возвращает текущую дату. Если в веденном номере телефона более 8 символом, то записывается только 8.

CREATE OR REPLACE TRIGGER trigg1

BEFORE INSERT OR UPDATE ON SERVICE

FOR EACH ROW

BEGIN

IF (:new.data > SYSDATE) THEN :new.data := SYSDATE; END IF;

IF (:new.telephone\_cl < 100000000) THEN :new.telephone\_cl := :new.telephone\_cl\*10; END IF;

END;

CREATE OR REPLACE TRIGGER trigg12

BEFORE INSERT OR UPDATE ON STAFF

FOR EACH ROW

BEGIN

IF (:new.data > SYSDATE) THEN :new.data := SYSDATE; END IF;

IF (:new.telephone\_st < 100000000) THEN :new.telephone\_st := :new.telephone\_st\*10; END IF;

END;

Триггер 2, который при удалении определенной записи из таблицы Service записывает данные в новую таблицу ServArch. Этот триггер работает как запись в архив.

CREATE OR REPLACE TRIGGER trigg2

AFTER DELETE ON SERVICE

FOR EACH ROW

BEGIN

insert into Serv\_Arch values(:old.data, :old.id\_staffer, :old.telephone\_cl);

END;

Serv\_Arch

CREATE TABLE Serv\_Arch

(

data timestamp with local time zone not null,

id\_staffer int not null,

telephone\_cl int not null,

primary key(data, id\_staffer, telephone\_cl)

);

Триггер 3, который при удалении определенной строки из таблицы Client удаляет все записи из таблиц, которые ссылаются на эту таблицу. Этот триггер работает как каскадное удаление.

Триггер 3:

CREATE OR REPLACE TRIGGER trigg3

BEFORE DELETE ON Client

FOR EACH ROW

DECLARE

PRAGMA AUTONOMOUS\_TRANSACTION;

BEGIN

DELETE FROM Service WHERE telephone\_cl = :old.telephone\_cl;

COMMIT;

END;

CREATE OR REPLACE TRIGGER trigg31

BEFORE DELETE ON Client

FOR EACH ROW

DECLARE

PRAGMA AUTONOMOUS\_TRANSACTION;

BEGIN

DELETE FROM Client\_auto WHERE telephone\_cl = :old.telephone\_cl;

COMMIT;

END;

CREATE OR REPLACE TRIGGER trigg32

BEFORE DELETE ON Staff

FOR EACH ROW

DECLARE

PRAGMA AUTONOMOUS\_TRANSACTION;

BEGIN

DELETE FROM Service WHERE id\_staffer = :old.id\_staffer;

COMMIT;

END;

CREATE OR REPLACE TRIGGER trigg33

BEFORE DELETE ON Staff

FOR EACH ROW

DECLARE

PRAGMA AUTONOMOUS\_TRANSACTION;

BEGIN

DELETE FROM Staff\_auto WHERE id\_staffer = :old.id\_staffer;

COMMIT;

END;

create or replace procedure test\_proc is

newExc exception;

num1 number;

num2 number;

numSale number;

randomSale number;

fioStaffer VARCHAR2(20);

newFioStaffer VARCHAR2(20);

res numeric(3,1);

numStr number;

str VARCHAR2(20);

resNumStr number;

begin

--ZERO\_DIVIDE!

dbms\_random.seed(SYSTIMESTAMP);

num1 := floor(dbms\_random.value(0, 15));

num2 := floor(dbms\_random.value(0, 15));

res := num1/num2;

--INVALID\_NUMBER!

insert into client(telephone\_cl,fio,sale,id\_city)

values(998745455, 'Oslov Alexand Petrovich', 'five', 3);

--NO\_DATA\_FOUND!

randomSale := floor(dbms\_random.value(0, 16));

select distinct sale into numSale

from client

where sale = randomSale;

--TOO\_MANY\_ROWS!

select fio\_staff into fioStaffer

from staff

where typeworker like 'waiter%';

--NEW\_EXCEPTION!

if randomSale = 1 or randomSale = 2 then raise newExc;

end if;

insert into city(name\_city)

values('Kharkiv');

insert into client(telephone\_cl,fio,sale,id\_city)

values(991213123,'Ivanov Ivan Ivanovich', 0, 1);

insert into client\_auto(telephone\_cl, password\_cl)

values(991213123,'user1');

insert all

into TypeWorker (typeWorker, salary)

values ('manager', 20000)

into TypeWorker (typeWorker, salary)

values ('waiterA', 15000)

into TypeWorker (typeWorker, salary)

values ('waiterB', 11000)

into TypeWorker (typeWorker, salary)

values ('waiterC', 9000)

into TypeWorker (typeWorker, salary)

values ('newbier', 7000)

select \* from dual;

insert into staff(fio\_staff, telephone\_st,birthday,typeworker)

values('Avdeeva Anna Sergeevna', 506689696, '22/12/78', 'waiterA');

insert into Staff\_auto(telephone\_st,password\_st)

values(506689696, 'waitera1');

insert into service(data,id\_staffer,telephone\_cl)

values(TO\_TIMESTAMP('2020-04-02 17:34:00', 'YYYY-MM-DD HH24:MI:SS'),1, 991213123);

insert into review(telephone\_cl,note,marks)

values(991213123, 'It is beautiful place with very taste dishes)))', 5);

update city

set name\_city = 'Kiev'

where name\_city = 'Kharkiv';

update client

set fio = 'Ivanova Irina Ivanovna'

where telephone\_cl = 991213123;

update client\_auto

set password\_cl = 'user2'

where telephone\_cl = 991213123;

update typeworker

set salary = 7500

where typeworker = 'newbier';

update staff

set typeworker = 'manager'

where telephone\_st = 506689696;

update staff\_auto

set password\_st = 'manager1'

where telephone\_st = 506689696;

update service

set data = TO\_TIMESTAMP('2019-04-02 17:34:00', 'YYYY-MM-DD HH24:MI:SS')

where data = TO\_TIMESTAMP('2020-04-02 17:34:00', 'YYYY-MM-DD HH24:MI:SS');

delete from review

where telephone\_cl = 991213123 and note = 'It is beautiful place with very taste dishes)))' and marks = 5;