

# Основы PL\SQL для разработчиков



Лазоренко Александр Андреевич

# 1. О себе

- Место работы ООО «Софт Инжиниринг»
- Возраст – 24 года
- Стаж работы – 2 месяца
- Должность – администратор баз данных
- До поступления на курс не занимался разработкой

## 2. Об обучении

### Что было нового:

- Большинство аспектов разработки, рассмотренных в ходе обучения
- Работа с Git

### Самая важная / интересная тема

- Тема 2 – Основы (типы)
- Тема 4 – Функции (и процедуры)

### 3. Сложности в обучении

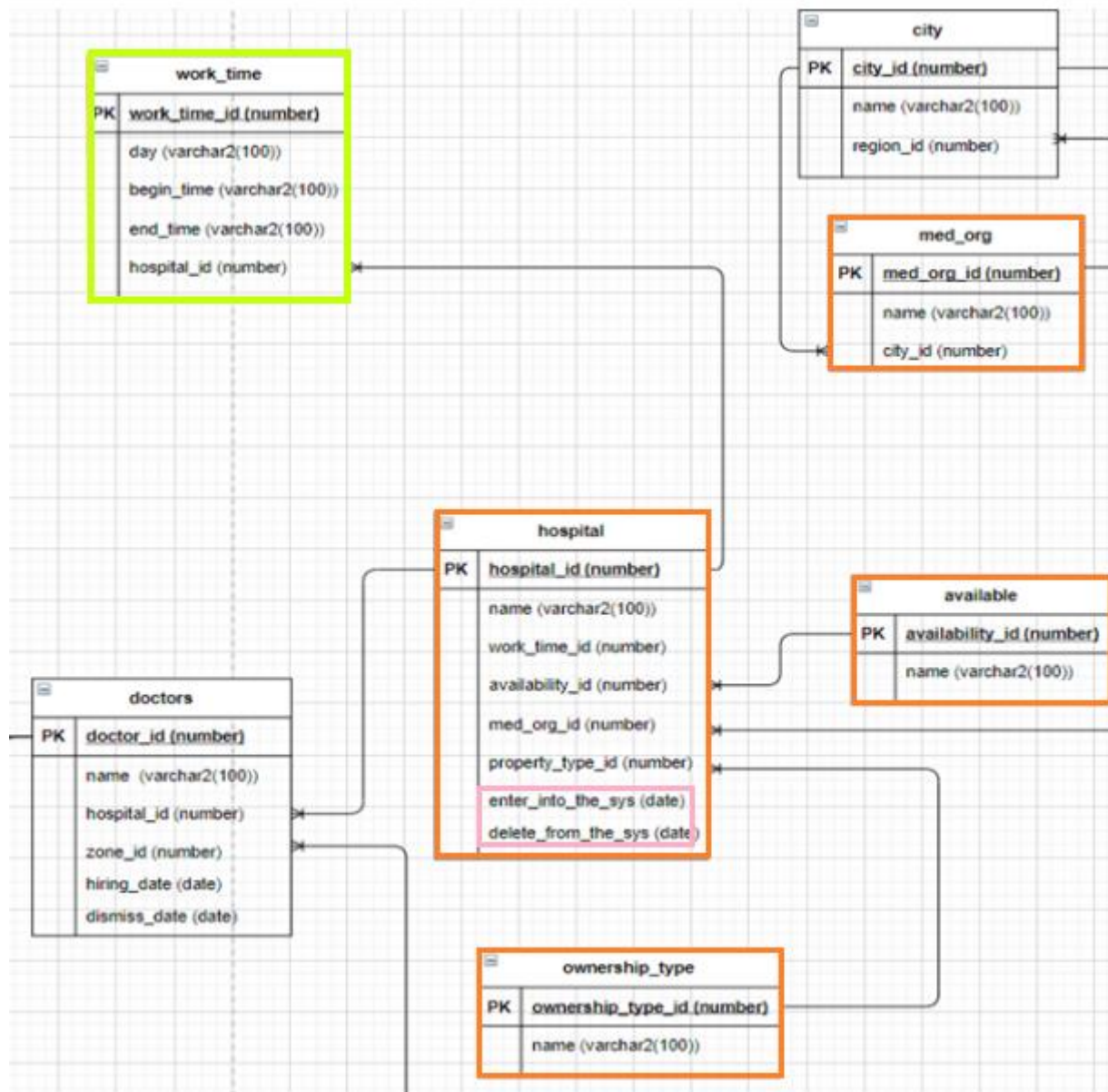
- Выполнение д/з по теме 2 «Основы» - сложности с применением переменных различных типов
- Выполнение д/з по теме 4 «Функции» (**самая интересная тема**) - сложности с увязкой бизнес-логики и создаваемых механизмов

Основная причина – отсутствие подобного опыта

# 4. Что сделано

## ER-модель:

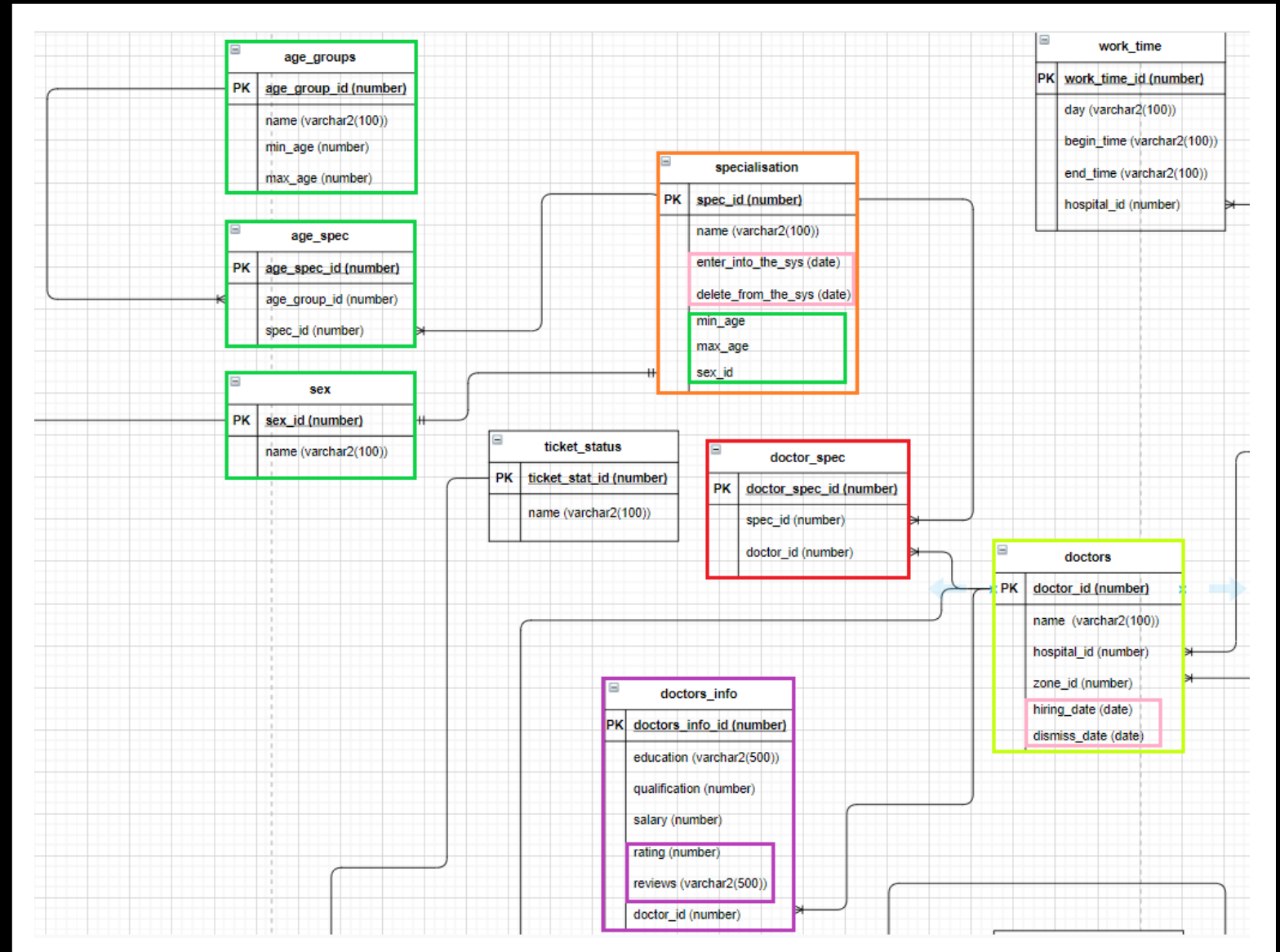
- Больница
- Расписание работы больницы
- Механизм удаления



# 4. Что сделано - продолжение

ER-модель:

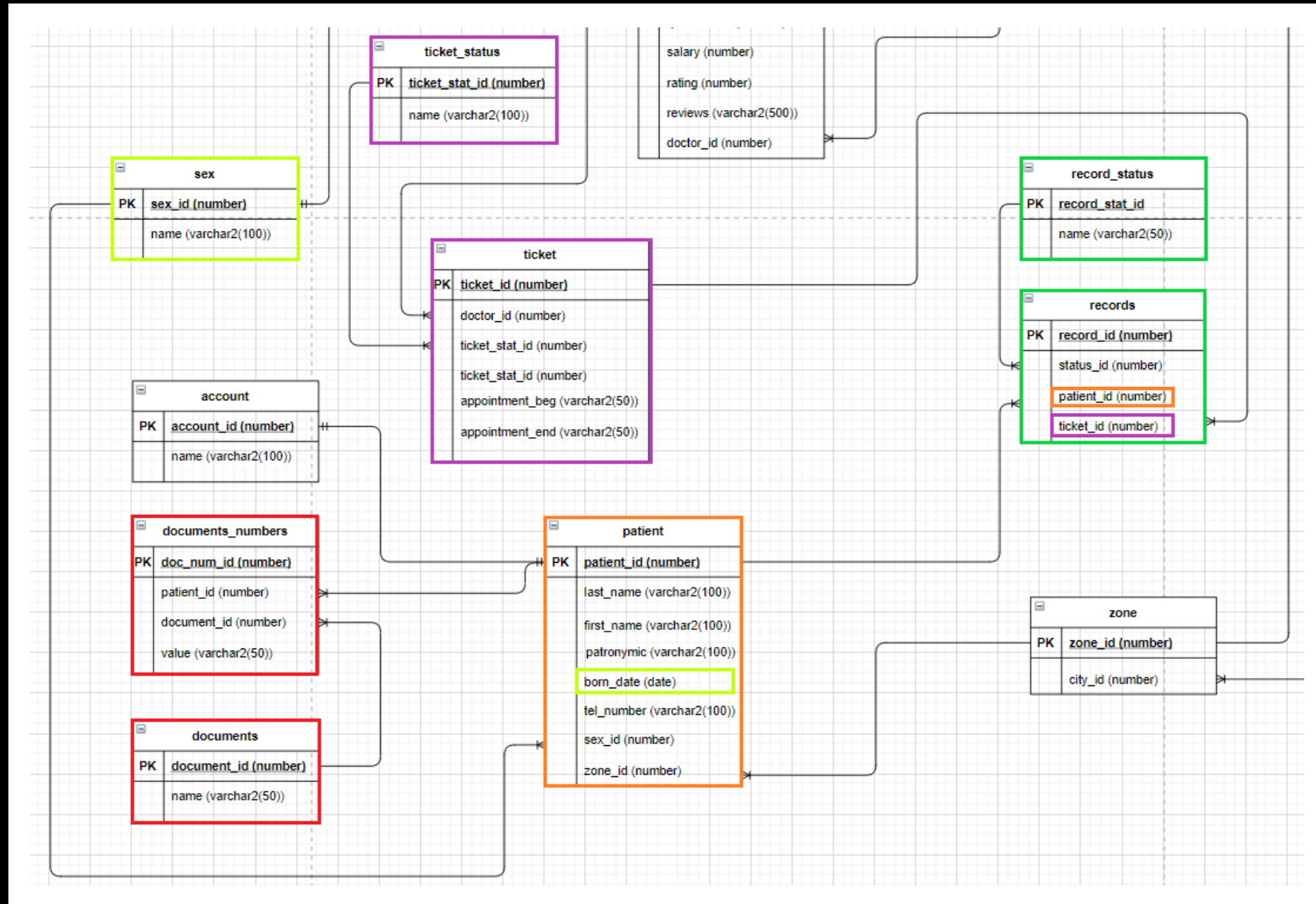
- Специальности
- Доктора
- Связь между докторами и специальностями
- Механизм удаления
- Пол и возраст для специальностей
- Оценка врачей через отзывы



# 4. Что сделано - продолжение

ER-модель:

- Пациенты
- Пол и возраст пациента
- Документы пациента
- Журнал пациента
- Талоны



# 5. Логика БД

## 1. Выдача всех городов и регионов

```
create or replace function lazorenko_al.get_cities_regions(  
p_region_id in number default null)  
return sys_refcursor  
as  
    v_cursor_1 sys_refcursor;  
begin  
    open v_cursor_1 for  
        select c.name, r.name  
        from lazorenko_al.city c inner join lazorenko_al.region r USING(region_id)  
        where p_region_id=region_id or p_region_id is null;  
  
    return v_cursor_1;  
end;
```

```
declare  
    type record1 is record (cname varchar2(100), rname varchar2(100));  
    v_city_region record1;  
    v_cursor_1 sys_refcursor;  
begin  
    v_cursor_1 :=lazorenko_al.get_cities_regions();  
    loop  
        FETCH v_cursor_1 INTO v_city_region;  
        exit when v_cursor_1%notfound;  
        dbms_output.put_line( A: v_city_region.cname ||' - '|| v_city_region.rname);  
    end loop;  
    close v_cursor_1;  
end;
```

completed in 57 ms

Кемерово - Кемеровская область

Новокузнецк - Кемеровская область

Новосибирск - Новосибирская область

Красноярск - Красноярский край



## 2. Выдача специальностей по докторам, по больницам

```
create or replace function lazorenko_al.get_specs(  
  p_doctor_id in number default null,  
  p_hospital_id in number default null)  
return sys_refcursor as  
  v_cursor_1 sys_refcursor;  
begin  
  open v_cursor_1 for  
    select distinct s.name  
  
    from specialisation s inner join doctor_spec using(spec_id)  
    inner join doctor d using(doctor_id)  
    inner join hospital h using(hospital_id)  
  
    where s.delete_from_the_sys is null and d.dismiss_date is null and h.delete_from_the_sys is null  
      and (p_hospital_id=hospital_id or p_hospital_id is null)  
      and (p_doctor_id=doctor_id or p_doctor_id is null);  
  
  return v_cursor_1;  
end;
```

```
declare  
  type record2 is record (spec_name varchar2(50));  
  v_spec_name record2;  
  v_cursor_1 sys_refcursor;  
begin  
  v_cursor_1 :=lazorenko_al.get_specs(P_DOCTOR_ID: 3, P_HOSPITAL_ID: null);  
  loop  
    FETCH v_cursor_1 INTO v_spec_name;  
    exit when v_cursor_1%notfound;  
    dbms_output.put_line( A: v_spec_name.spec_name);  
  end loop;  
  close v_cursor_1;  
end;
```

completed in 13 ms

аллерголог

педиатр

лор

травматолог

уролог

### 3. Выдача больниц по специальностям, сортировки

```
create or replace procedure lazorenko_al.get_doctors_specs(
    p_spec_id in number,
    out_cursor out sys_refcursor)
as
begin
    open out_cursor for
        select h.name, a.name, count(d.doctor_id) as количество_врачей,
        case
            when o.ownership_type_id=1 then 'частная'
            when o.ownership_type_id=2 then 'государственная'
            end as форма_собственности,
        case
            when w.end_time is null then ' - '
            else w.end_time
            end as закрытие

        from hospital h left join work_time w on h.hospital_id=w.hospital_id
        inner join ownership_type o on h.ownership_type_id=o.ownership_type_id
        inner join doctor d on d.hospital_id=h.hospital_id
        inner join doctor_spec ds on d.doctor_id=ds.doctor_id
        inner join available a on h.availability_id=a.availability_id

        where (spec_id=p_spec_id or p_spec_id is null) and h.delete_from_the_sys is null
            and w.day=to_char(sysdate, 'd')

        group by h.name, a.name, o.ownership_type_id, w.end_time
        order by case
            when o.ownership_type_id=1 then 1
            else 0 end desc, количество_врачей desc,
            case
            when w.end_time>TO_CHAR(sysdate, 'hh24:mi:ss') then 1
            else 0
            end desc;
end;
```

```
declare
    v_cursor sys_refcursor;
    type record3 is record (hname varchar2(100), aname varchar2(100), doctor_count number,
        oname varchar2(100), end_time varchar2(100));
    v_hospital_time record3;
begin
    lazorenko_al.get_doctors_specs(p_spec_id, 4, out_cursor:=v_cursor);
    loop
        fetch v_cursor into v_hospital_time;
        exit when v_cursor%notfound;
        dbms_output.put_line ('A3 'название больницы - ' || v_hospital_time.hname || '; сейчас ' || v_hospital_time.aname
            || '; число докторов указанной специальности=' || v_hospital_time.doctor_count ||
            chr(10)||
            '; форма собственности - ' || v_hospital_time.oname || '; закрывается в ' || v_hospital_time.end_time)
    end loop;
    close v_cursor;
end;
```

completed in 22 ms

название больницы - больница2; сейчас не доступна; число докторов указанной специальности=1  
; форма собственности - частная; закрывается в 18:00

название больницы - больница5; сейчас не доступна; число докторов указанной специальности=1  
; форма собственности - частная; закрывается в 18:00

название больницы - больница1; сейчас доступна; число докторов указанной специальности=1  
; форма собственности - частная; закрывается в 18:00

название больницы - больница10; сейчас доступна; число докторов указанной специальности=2  
; форма собственности - государственная; закрывается в 18:00

название больницы - больница6; сейчас доступна; число докторов указанной специальности=2  
; форма собственности - государственная; закрывается в 18:00

название больницы - больница3; сейчас доступна; число докторов указанной специальности=1  
; форма собственности - государственная; закрывается в 18:00

## 4. Выдача врачей по больнице, сортировки

```
create or replace function lazorenko_al.get_doctor(
    p_hospital_id in number,
    p_zone_id in number)
return sys_refcursor
as
    v_cursor_1 sys_refcursor;
begin
    open v_cursor_1 for
        select d.name, s.name, di.qualification

        from doctor d inner join doctor_spec using(doctor_id)
        inner join specialisation s using(spec_id)
        inner join doctors_info di using(doctor_id)
        inner join hospital using(hospital_id)

        where (hospital_id=p_hospital_id or p_hospital_id is null) and d.dismiss_date is null

        order by di.qualification desc,
            case
                when d.zone_id=p_zone_id then 1
                else 0 end desc;

    return v_cursor_1;
end;

declare
    type record4 is record (dname varchar2(100), spname varchar2(100), qualification number);
    v_doctor record4;
    v_cursor_1 sys_refcursor;
begin
    v_cursor_1 :=lazorenko_al.get_doctor( P_HOSPITAL_ID: 6, P_ZONE_ID: 2);
    loop
        FETCH v_cursor_1 INTO v_doctor;
        exit when v_cursor_1%notfound;
        dbms_output.put_line( A: 'ФИО врача - ' || v_doctor.dname || '; специальность - ' ||
            v_doctor.spname || '; квалификация - ' || v_doctor.qualification);
    end loop;
    close v_cursor_1;
end;
```

completed in 23 ms

ФИО врача - Иванов2; специальность - педиатр; квалификация - 8

ФИО врача - Иванов2; специальность - лор; квалификация - 8

ФИО врача - Климова2; специальность - гинеколог; квалификация - 4

ФИО врача - Сидоров2; специальность - травматолог; квалификация - 3

ФИО врача - Петров2; специальность - аллерголог; квалификация - 3

## 5. Талоны по врачу, сортировки

```
create or replace procedure lazorenko_al.get_ticket(
    v_doctor_id in number,
    out_cursor out sys_refcursor)
as
begin
    open out_cursor for
        select t.ticket_id, d.name, t.appointment_beg, t.appointment_end
        from ticket t right join doctor d using(doctor_id)
        where (doctor_id=v_doctor_id or v_doctor_id is null) and t.appointment_beg>to_char(sysdate, 'yyyy-mm-dd hh24:mi:ss')
        order by t.appointment_beg;
end;

declare
    v_cursor sys_refcursor;
    type record5 is record (ticket_id number, dname varchar2(100), appointment_beg varchar2(100), appointment_end varchar2(100));
    v_ticket record5;
begin
    lazorenko_al.get_ticket(V_DOCTOR_ID: 4, OUT_CURSOR: v_cursor);
    loop
        fetch v_cursor into v_ticket;
        exit when v_cursor%notfound;
        dbms_output.put_line ('A: 'id талона - ' ||v_ticket.ticket_id || '; врач - ' ||v_ticket.dname ||'; начало приёма - ' ||
            v_ticket.appointment_beg || '; конец приёма - ' || v_ticket.appointment_end);
    end loop;
    close v_cursor;
end;
```

completed in 12 ms

id талона - 32; врач - Петров; начало приёма - 2021-11-25 9:00:00; конец приёма - 2021-11-25 9:30:00

## 6. Выдача журнала записей

```
create or replace function lazorenko_al.get_records(  
    p_patient_id in number default null,  
    p_record_stat_id in number default null)  
return sys_refcursor as  
    v_cursor_1 sys_refcursor;  
  
begin  
    open v_cursor_1 for  
        select last_name, first_name, petronymic, d.name, record_status.name, appointment_beg, appointment_end  
        from lazorenko_al.patient p left join lazorenko_al.records using(patient_id)  
        inner join record_status using(record_stat_id)  
        inner join ticket using(ticket_id)  
        inner join lazorenko_al.doctor d using(doctor_id)  
  
        where (patient_id=p_patient_id or p_patient_id is null) and (record_stat_id=p_record_stat_id or p_record_stat_id is null);  
  
    return v_cursor_1;  
end;
```

```
declare  
type record8 is record(last_name varchar2(100), first_name varchar2(100),  
    petronymic varchar2(100), dname varchar2(100), rec_stat varchar2(50), appointment_beg varchar2(100), appointment_end varchar2(100));  
v_patient record8;  
v_cursor_1 sys_refcursor;  
  
begin  
    v_cursor_1 :=lazorenko_al.get_records( P_PATIENT_ID: 1, P_RECORD_STAT_ID: null);  
    loop  
        FETCH v_cursor_1 INTO v_patient;  
        exit when v_cursor_1%notfound;  
        dbms_output.put_line( A: '-пациент - ' || v_patient.last_name || ' ' || v_patient.first_name || ' ' || v_patient.petronymic  
            || '; врач - ' || v_patient.dname || '; статус записи - ' || v_patient.rec_stat ||chr(10)||'; начало приёма - '  
            || v_patient.appointment_beg || ' ; конец приёма - ' || v_patient.appointment_end||chr(10));  
    end loop;  
    close v_cursor_1;  
end;
```

completed in 11 ms

-пациент - petrov ivan ivanovich; врач - Иванов; статус записи - действует  
; начало приёма - 2021-11-20 11:30:00; конец приёма - 2021-11-20 12:00:00  
-пациент - petrov ivan ivanovich; врач - Иванов; статус записи - исполнено  
; начало приёма - 2021-10-10 13:00:00; конец приёма - 2021-10-10 13:30:00  
-пациент - petrov ivan ivanovich; врач - Петров; статус записи - исполнено  
; начало приёма - 2021-10-16 14:00:00; конец приёма - 2021-10-16 14:30:00

## 7. Метод записи по условиям – функции проверки условий

```
create or replace function lazorenko_al.ticket_status_check(  
    p_ticket_id in number,  
    p_patient_id number)  
return boolean as  
    v_count number;  
begin  
    select count(*)  
    into v_count  
    from lazorenko_al.ticket t left join lazorenko_al.records r on r.ticket_id=t.ticket_id  
    where (t.ticket_id=p_ticket_id and t.ticket_stat_id=1)  
        or (t.ticket_id=p_ticket_id and t.ticket_stat_id=2 and r.patient_id=p_patient_id);  
  
    return v_count>0;  
end;
```

- Проверка возраста
- Проверка пола
- **Проверка статуса талона**
- Проверка повторной записи
- Проверка времени начала талона
- Проверка на удаление больницы, врача, специальности
- Проверка наличия полиса
- Проверка на соответствие вводимых параметров

## 7. Метод записи по условиям – функция проверки всех условий перед записью

```
create or replace function lazorenko_al.check_for_accept(
    p_ticket_id in number,
    p_patient_id in number,
    p_spec_id in number,
    p_doctor_id in number,
    p_hospital_id in number,
    p_lazy_check in boolean default false,
    out_messages out varchar2
)
return boolean
as
    v_result boolean := true;
begin
    if (not lazorenko_al.check_IN_parameters(
        p_hospital_id => p_hospital_id,
        p_doctor_id => p_doctor_id,
        p_spec_id => p_spec_id,
        p_ticket_id => p_ticket_id
    )) then v_result:=false;
    out_messages:=out_messages||chr(10)
    ||'несоответствие врача и специальности или врача и больницы или выбранного талона и врача';
    if (p_lazy_check) then return v_result; end if;
    end if;

    if (not lazorenko_al.patient_doc_check(
        p_patient_id => p_patient_id
    )) then v_result:=false;
    out_messages:=out_messages||chr(10)
    ||'отсутствуют данные о полисе ОМС пациента';
    end if;
```

**p\_lazy\_check** – позволяет завершить проверку после первой неудачи (несоответствие вносимых параметров)

```
    if (not lazorenko_al.not_deleted_check(
        p_doctor_id => p_doctor_id
    )) then v_result:=false;
    out_messages:=out_messages||chr(10)
    ||'больница, врач или специальность удалены';
    end if;

    if (not lazorenko_al.time_check(
        p_ticket_id => p_ticket_id
    )) then v_result:=false;
    out_messages:=out_messages||chr(10)
    ||'данный приём уже завершился';
    end if;

    if (not lazorenko_al.sex_check(
        p_patient_id => p_patient_id,
        p_spec_id => p_spec_id
    )) then v_result:=false;
    out_messages:=out_messages||chr(10)
    ||'пол пациента не соответствует полу специальности';
    end if;

    if (not lazorenko_al.check_age(
        p_patient_id => p_patient_id,
        p_spec_id => p_spec_id
    )) then v_result:=false;
    out_messages:=out_messages||chr(10)
    ||'возраст пациента не соответствует возрасту специальности';
    end if;

    if (not lazorenko_al.ticket_check(
        p_ticket_id => p_ticket_id,
        p_patient_id => p_patient_id
    )) then v_result:=false;
    out_messages:=out_messages||chr(10)
    ||'пациент уже был записан на этот талон';
    end if;

    if (not lazorenko_al.ticket_status_check(
        p_ticket_id => p_ticket_id,
        p_patient_id => p_patient_id
    )) then v_result:=false;
    out_messages:=out_messages||chr(10)
    ||'талон закрыт';
    end if;

    return v_result;
end;
```

## 7. Метод записи по условиям – функции записи

```
create or replace function write_to_records(  
    p_patient_id in number,  
    p_ticket_id in number)  
return lazorenko_al.records.record_id%type as  
    v_record_id lazorenko_al.records.record_id%type;  
begin  
    insert into lazorenko_al.records(record_id, record_stat_id, patient_id, ticket_id)  
    values (default, 1, p_patient_id, p_ticket_id)  
    returning record_id into v_record_id;  
  
    update lazorenko_al.ticket t set t.ticket_stat_id=2 where t.ticket_id=p_ticket_id;  
  
    commit;  
  
    return v_record_id;  
end;
```

- В журнале появляется действующая запись
- Статус талона меняется на «закрыт»



## 7. Метод записи по условиям – функции записи с проверкой условий

```
create or replace function lazorenko_al.accept_record_by_rules(  
    v_ticket_id number,  
    v_patient_id number,  
    v_spec_id number,  
    v_doctor_id number,  
    v_hospital_id number,  
    v_lazy_check boolean,  
    v_messages in out varchar2,  
    v_result out number)  
return lazorenko_al.records.record_id%type as  
    v_record_id lazorenko_al.records.record_id%type;  
begin  
    if (lazorenko_al.check_for_accept(  
        p_ticket_id => v_ticket_id,  
        p_patient_id => v_patient_id,  
        p_spec_id => v_spec_id,  
        p_doctor_id => v_doctor_id,  
        p_hospital_id => v_hospital_id,  
        p_lazy_check => v_lazy_check,  
        out_messages => v_messages))  
    then v_record_id:=lazorenko_al.write_to_records(  
        p_patient_id => v_patient_id,  
        p_ticket_id => v_ticket_id);  
    dbms_output.put_line(A: v_record_id || ' - ' || 'запись осуществлена успешно');  
else v_result:=sys.diutil.bool_to_int(B: lazorenko_al.check_for_accept(  
    P_TICKET_ID: v_ticket_id,  
    P_PATIENT_ID: v_patient_id,  
    P_SPEC_ID: v_spec_id,  
    P_DOCTOR_ID: v_doctor_id,  
    P_HOSPITAL_ID: v_hospital_id,  
    P_LAZY_CHECK: v_lazy_check,  
    OUT_MESSAGES: v_messages));  
    dbms_output.put_line(A: v_messages);  
end if;  
return v_result;  
end;
```

```
declare  
    v_messages varchar2(500);  
    v_result number;  
begin  
    v_result:=lazorenko_al.accept_record_by_rules(  
        V_TICKET_ID: 27, V_PATIENT_ID: 7, V_SPEC_ID: 6, V_DOCTOR_ID: 3, V_HOSPITAL_ID: 5, V_LAZY_CHECK: false,  
        V_MESSAGES: v_messages, V_RESULT: v_result);  
end;
```

completed in 17 ms

несоответствие врача и специальности или врача и больницы или выбранного талона и врача  
данный приём уже завершился  
пол пациента не соответствует полу специальности  
возраст пациента не соответствует возрасту специальности  
талон закрыт

```
declare  
    v_messages varchar2(500);  
    v_result number;  
begin  
    v_result:=lazorenko_al.accept_record_by_rules(  
        V_TICKET_ID: 33, V_PATIENT_ID: 1, V_SPEC_ID: 2, V_DOCTOR_ID: 3, V_HOSPITAL_ID: 4, V_LAZY_CHECK: false,  
        V_MESSAGES: v_messages, V_RESULT: v_result);  
end;
```

completed in 26 ms

80 - запись осуществлена успешно

## 8. Метод отмены по условиям – функции проверки условий

```
create or replace function lazorenko_al.hospital_time_check(  
    p_hospital_id in number)  
return boolean as  
    v_count number;  
begin  
    select count(*)  
    into v_count  
    from lazorenko_al.work_time w  
    where w.end_time>(TO_CHAR(sysdate+1/12, 'hh24:mi'))  
    and w.day in (to_char(sysdate, 'd')) and w.hospital_id=p_hospital_id;  
  
    return v_count>0;  
end;
```

- Проверка временных ограничений (талон можно отменять , только если эта клиника работает сегодня еще минимум два часа)
- Проверка времени начала талона
- Проверка на соответствие вводимых параметров

## 8. Метод отмены по условиям – функция проверки всех условий перед отменой

```
create or replace function lazorenko_al.check_for_cancel(  
    p_hospital_id in number,  
    p_ticket_id in number,  
    p_patient_id in number,  
    out_messages out varchar2  
)  
return boolean  
as  
    v_result boolean := true;  
begin  
    if (not lazorenko_al.check_IN_parameters2(  
        p_patient_id => p_patient_id,  
        p_ticket_id => p_ticket_id  
    )) then v_result:=false;  
    out_messages:=out_messages||chr(10)  
    ||'у вас отсутствует действующий талон с подобными параметрами или талон закрыт';  
    end if;  
  
    if (not lazorenko_al.ticket_time_check(  
        p_ticket_id => p_ticket_id  
    )) then v_result:=false;  
    out_messages:=out_messages||chr(10)  
    ||'приём уже завершился';  
    end if;  
  
    if (not lazorenko_al.hospital_time_check(  
        p_hospital_id => p_hospital_id  
    )) then v_result:=false;  
    out_messages:=out_messages||chr(10)  
    ||'не соблюдены ограничения по времени работы больницы';  
    end if;  
  
    return v_result;  
end;
```

## 8. Метод отмены по условиям – функция отмены записи

```
create or replace function lazorenko_al.cancel_record(  
    p_ticket_id in number)  
return number as  
    v_record_id number;  
begin  
    update lazorenko_al.ticket t set t.ticket_stat_id=1 where t.ticket_id=p_ticket_id;  
    update lazorenko_al.records r set r.record_stat_id=2 where r.ticket_id=p_ticket_id;  
    commit;  
    return v_record_id;  
end;
```

- Статус талона меняется на «открыт»
- Статус записи меняется на «отменена»

## 8. Метод отмены по условиям – функция отмены записи по условиям

```
create or replace function lazorenko_al.cancel_record_by_rules(  
    v_ticket_id number,  
    v_patient_id number,  
    v_hospital_id number,  
    v_messages in out varchar2,  
    v_result out number)  
return lazorenko_al.records.record_id%type as  
    v_record_id lazorenko_al.records.record_id%type;  
begin  
    if (lazorenko_al.check_for_cancel(  
        p_ticket_id => v_ticket_id,  
        p_patient_id => v_patient_id,  
        p_hospital_id => v_hospital_id,  
        out_messages => v_messages))  
    then v_record_id:=lazorenko_al.cancel_record(p_ticket_id => v_ticket_id);  
    dbms_output.put_line('A: v_record_id ||' - '||'запись отменена');  
  
    else v_result:=sys.diutil.bool_to_int('B: lazorenko_al.check_for_cancel(  
        P_HOSPITAL_ID: v_hospital_id,  
        P_TICKET_ID: v_ticket_id,  
        P_PATIENT_ID: v_patient_id,  
        OUT_MESSAGES: v_messages));  
  
    dbms_output.put_line('A: v_messages');  
end if;  
return v_result;  
end;
```

```
declare  
    v_messages varchar2(500);  
    v_result number;  
begin  
    v_result:=lazorenko_al.cancel_record_by_rules(  
        V_TICKET_ID: 33, V_PATIENT_ID: 1, V_HOSPITAL_ID: 4,  
        V_MESSAGES: v_messages, V_RESULT: v_result);  
end;
```

completed in 35 ms

- запись отменена

```
declare  
    v_messages varchar2(500);  
    v_result number;  
begin  
    v_result:=lazorenko_al.cancel_record_by_rules(  
        V_TICKET_ID: 27, V_PATIENT_ID: 2, V_HOSPITAL_ID: 5,  
        V_MESSAGES: v_messages, V_RESULT: v_result);  
end;
```

completed in 25 ms

у вас отсутствует действующий талон с подобными параметрами или талон закрыт  
приём уже завершился

## 6. Ожидания / Планы

- Применение пакетов в рамках разработанных методов
- Получение полезной информации об администрировании баз данных в ходе прохождения курса
- Получение базовых знаний и навыков в сфере разработки
- Возможность эффективного взаимодействия с разработчиками по вопросам, связанным с внедрением/модификацией приложений в БД