

АО «БАРС Груп»

**Расширение функционала настройки системы
БАРС.Здравоохранение-МИС**

Руководство администратора

Версия 02

Казань 2018

3.3.2	Количество амбулаторных карт пациентов.....	14
3.3.3	Количество ИБ.....	14
3.3.4	Количество посещений.....	14
3.3.5	Количество записей на услуги.....	15
3.3.6	Количество амбулаторных талонов.....	15
3.3.7	Количество пациентов с амбулаторными талонами.....	15
3.3.8	Количество пациентов, которым хоть раз была оказана услуга.....	16
3.4	Скрипт для переназначения прав просмотра мед.словарей со специальности на роль.....	16
3.5	Скрипт для переназначения прав просмотра диагнозов МКБ10 со специальности на роль.....	18
3.6	Просмотр прав на меню для Сотрудников.....	20
3.7	Просмотр шаблонов, использующих наш медицинский словарь.....	20
3.8	Просмотр прав на разделы для Сотрудников.....	21
3.9	Просмотр прав на разделы для Сотрудников (с каталогами).....	21
3.10	Заведения Администраторов ЛПУ с назначением Ролей.....	22
3.11	Клонирование пользовательских процедур в другие ЛПУ по каталогу.....	24
3.12	Массовое закрытие ТАПов, открытых до определенной даты.....	26
3.13	Закрытие случаев заболевания прошлого года.....	27
3.14	Копирование типов заявок из одного ЛПУ во все остальные.....	28
3.15	Поиск по отчетам.....	28

Перечень терминов и сокращений

Термин, сокращение	Определение
БД	База данных
КДЛ	Клинико-диагностическая лаборатория
ЛПУ	Лечебно-профилактическое учреждение
Система, МИС	Медицинская информационная система «БАРС.Здравоохранение-МИС»
ТАП	Талон амбулаторного приема
ФЭР	Федеральная электронная регистратура

1 Введение

Существуют скрипты для первичной настройки МИС и оптимизации. Наиболее часто используемые скрипты и пользовательские процедуры описаны в данном руководстве.

2 Расширение настроек МИС

Таблица 1 – Расширение настроек МИС

Действие	Пользовательская процедура	Описание
Полное удаление ролей	DEL_ROLE_FULL	Процедура создана для удаления лишних ролей (например, пришедших со стандартной поставкой). Процедура удаляет роль и ссылки на нее из разделов: назначенные права на разделы, пользователи, главное меню, модули и права записей
Копирование прав на меню	COPY_TO_ROLE_MENU_PRIVS	Вызов из окна «Настройка ролей». Копирует права на меню от отмеченной роли к выбранной из комбобокса
Полное удаление услуг	DEL_SERVICE_FULL	Вызов из окна «Общие услуги». Процедура работает с отмеченными «флажками» записями. Удаления услуги не происходит (ошибки не выводятся), если есть хотя бы одна запись в регистратуру на эту услугу
Удаление всех прав записей с медицинских словарей	DEL_ACS_FROM_ADDDICT	Вызов из настройки медицинских словарей
Добавление роли группе персонала (выделенного «флажками»)	ADD_ROLE_TO_EMPS	Вызов из окна «Настройки персонала»
Копирование услуг всем врачам по специальностям (из одного ЛПУ во множество)	COPY_EMPSERVS_BY_SPEC	Привязана на тестовой базе к разделу «LPUDICT», в меню: «Словари/ Контрагенты/ Список ЛПУ». Отметьте «флажками» ЛПУ, куда требуется скопировать настройки, затем при вызове процедуры выберите ЛПУ-донора настроек. Далее процедурой из корневого каталога «Сотрудники ЛПУ донора» выбираются случайным образом «эталонные» сотрудники по каждой специальности, и по ним происходит копирование оказываемых услуг, сотрудникам с той же специальностью в другие ЛПУ
Добавление анализа и его структуры при настройке состава услуги	ANALYZE_ADD_ON_STR	Вызов при добавлении услуги в состав другой. Процедура создает анализ в КДЛ, исследование и связывает их. Пропишите внутри пакета ссылку на биоматериал, остальные параметры настраиваются сами. После ее работы остается добавить условия забора к анализу и проставить биоматериал в исследованиях

Действие	Пользовательская процедура	Описание
Удаление всех прав на разделы у роли	DEL_ROLE_ALL_PRIVS	Вызов из раздела «Администратор/ Настройка ролей»
Заведение прав на просмотр всех разделов, аналог прав «Минимальная роль»	ADD_ROLE_MIN_ADDDICT	Вызов из раздела «Администратор/ Настройка ролей/ пункт контекстного меню на роли «Расширения/ Пользовательские процедуры/ Минимальная роль»»
Удаление всех прав на главное меню с роли	DEL_ROLE_MENU_ADDDICT	Вызов из раздела «Администратор/Настройка ролей/ пункт контекстного меню на роли «Расширения/ Пользовательские процедуры/ Удаление прав на Главное Меню»»

3 Скрипты и запросы

3.1 Запросы

3.1.1 Спецификация полей указанных таблиц (перечень полей: обязательность, тип, описание, связь с другими таблицами)

```
select
  decode(column_id,0,TABLE_NAME,null) as TABLE_NAME_TITLE,
  column_id, COLUMN_NAME,
  NULLABLE, DATA_TYPE, HAS_DEFAULT,
  REFERENCES_COLUMN,
  COMMENTS,
  TABLE_NAME
from (
  select cc.TABLE_NAME,
  tc.COLUMN_ID, tc.COLUMN_NAME,
  tc.NULLABLE,
  tc.DATA_TYPE ||
  case when tc.DATA_SCALE is not null then '(' || tc.DATA_PRECISION
|| ',' || tc.DATA_SCALE || ')'
  when tc.DATA_PRECISION is not null then '(' || tc.DATA_PRECISION
|| ')'
  when tc.DATA_LENGTH is not null and tc.DATA_TYPE like '%CHAR%'
then '(' || tc.DATA_LENGTH || ')'
  end DATA_TYPE,
  case when tc.DEFAULT_LENGTH is not null then 'Y' end HAS_DEFAULT,

  ( SELECT
  ucc2.TABLE_NAME||'.'||ucc2.column_name
  FROM user_constraints uc ,
  user_cons_columns uccl ,
  user_cons_columns ucc2
  WHERE uc.constraint_name = uccl.constraint_name
  AND uc.r_constraint_name = ucc2.constraint_name
  AND uccl.POSITION = ucc2.POSITION
  AND uc.constraint_type = 'R'
  AND uc.table_name = tc.TABLE_NAME
  AND uccl.column_name = tc.COLUMN_NAME
  ) REFERENCES_COLUMN,

  cc.COMMENTS
  from user_col_comments cc
  INNER JOIN user_tab_cols tc ON (cc.TABLE_NAME = tc.TABLE_NAME and
cc.TABLE_NAME = tc.TABLE_NAME and cc.COLUMN_NAME = tc.COLUMN_NAME)

UNION
select tab.table_name as TABLE_NAME,
```



```

    0 as COLUMN_ID, '' as COLUMN_NAME,
    '' as NULLABLE, '' as DATA_TYPE, Null as HAS_DEFAULT,
    Null as REFERENCES_COLUMN,
    tab.comments as COMMENTS
  from user_tab_comments tab
)
where table_name in
('D_AGENTS','D_AGENT_POLIS','D_INSURANCE_COMPANIES','D_AGENT_POLIS_KIND
S','D_AGENT_SOCIAL_STATES','D_SOCIALSTATES')
order by table_name, column_id;

```

3.1.2 Таблицы, ссылающиеся на указанную

```

select distinct
  upper(t.R_OWNER) OWNER,
  upper(t.TABLE_NAME) NAME,
  'TABLE' TYPE,c.TABLE_NAME
from sys.DBA_CONSTRAINTS t,
  sys.USER_CONSTRAINTS c
where
  t.R_OWNER = 'DEV'
  and t.R_CONSTRAINT_NAME = c.CONSTRAINT_NAME
  and c.TABLE_NAME = 'D_AGENTS'
  and c.CONSTRAINT_TYPE = 'P'

```

3.1.3 Объекты БД и номера строк, в которых присутствует указанный текст

```

select * from user_source where upper(TEXT) like
upper('%GET_PRICE%')

```

3.1.4 Объекты БД, в которых присутствует указанный текст (включая поиск по представлениям)

```

select *
  from D_DDL_OBJECTS
  where upper(OBJ_DDL) like '%GET_STATS%'
  and VERSION = (select ID from (select * from D_DDL_VERSIONS order
by VER_DATE desc) where rownum = 1);

```

3.1.5 Запрос к логам данных

```

--комментируем ненужные фильтры
select *
  from D_LOG t,
  D_LOG_COLS t1
  where t.ID = t1.PID(+)
  and t.DOC_ID = :ID -- фильтр по ID конкретной записи таблицы
  --and t.ACTION = 'D' -- фильтр по действию ('I', 'U', 'D')
  --and trunc(t.LOG_DATE) = to_date('24.07.2012','dd.mm.yyyy') --
фильтр по дате
  --and t.TABLENAME = 'D_VISITS' -- фильтр по таблице

```

```
--and t.USER_NAME = 'DEV' -- фильтр по пользователю БД
--and t.USER_HOST = 'user\MEDPC' -- фильтр по хосту
--and t1.COLUMN_NAME = 'REG_DATE' -- фильтр по полю
измененной/удаленной записи
```

3.1.6 Запрос к логам объектов

```
select t.ID, OBJ_DDL,
to_char(t.MODDATE, 'dd.mm.yyyy hh24:mi:ss') MODDATE_STR,
t.MODDATE,
t.OBJ_NAME,
t.SYSEVENT,
t.USER_HOST,
t.USER_OS_NAME||', '||t.USER_IP USER_INFO
from D_V_SCHEMA_DDLSTORE t
where t.OBJ_NAME = 'D_PKG_DIRECTION_SERVICES'
--and trunc(t.MODDATE) = to_date('23.09.2014', 'dd.mm.yyyy') --
фильтр по дате
--and sysevent = 'CREATE' -- фильтр по действию
order by MODDATE desc;
```

3.1.7 Где используется данная композиция

3.1.7.1 В полях на визитах

```
select vt.VT_CODE||' - '||vt.VT_NAME TEMPLATE, --шаблон
vtb.VT_CODE||' - '||vtb.VT_NAME TAB, --вкладка
tf.F_CODE||' - '||tf.F_NAME TAB_FIELDS, --поле вкладки
c.VC_CODE||' - '||c.VC_NAME CONTAINER, --контейнер
cf.F_CODE||' - '||cf.F_NAME CON_FIELD --поле контейнера
from D_VISIT_CON_FIELDS cf,
D_COMPOSITION cmp,
D_VISIT_CONTAINERS c,
D_VISIT_TAB_FIELDS tf,
D_VISIT_TABS vtb,
D_VIS_TEMP_TABS tt,
D_VISIT_TEMPLATES vt
where cmp.UNITLIST = 'MKB10'
and cmp.CODE = 'DEFAULT'
and cmp.ID = cf.COMPOSITION
and c.ID = cf.PID
and c.ID = tf.CONTAINER
and vtb.ID = tf.PID
and vtb.ID = tt.VISIT_TAB
and vt.ID = tt.PID
union
select vt.VT_CODE||' - '||vt.VT_NAME TEMPLATE, --шаблон
vtb.VT_CODE||' - '||vtb.VT_NAME TAB, --вкладка
tf.F_CODE||' - '||tf.F_NAME TAB_FIELDS, --поле вкладки
null,
null
from D_COMPOSITION cmp,
```

```

D_VISIT_TAB_FIELDS tf,
D_VISIT_TABS vtb,
D_VIS_TEMP_TABS tt,
D_VISIT_TEMPLATES vt
where cmp.UNITLIST = 'MKB10'
and cmp.CODE = 'DEFAULT'
and cmp.ID = tf.COMPOSITION
and vtb.ID = tf.PID
and vtb.ID = tt.VISIT_TAB
and vt.ID = tt.PID
order by 1,2,3,4,5

```

3.1.7.2 В фильтрах

```

select cmp.UNITLIST||' - '||cmp.CODE FILTER_KIND_C, --фильтр по
КОМПОЗИЦИИ
    cmp.FORM_MAIN, --форма раздела
    smc.NAME_SHOW_METHOD_COL, --поле метода показа
    sm.UNITCODE||' - '||sm.CODE SHOW_METHOD, --метод показа
    (select d_stragg_ex(d_tp_stragg_rec(cmp1.UNITLIST||' -
' ||cmp1.CODE||' ' ||cmp1.FORM_MAIN,chr(10),'ASC',null,null))
    from D_COMPOSITION cmp1,
    D_COMPOSITION_METHOD cm
    where cmp1.ID = cm.PID
    and cm.SHOW_METHOD = sm.ID) COMPOSITIONS_HAVE_SM--в каких
КОМПОЗИЦИЯХ есть этот метод показа
    from D_SHOW_METHOD_COLS smc,
    D_COMPOSITION cmp,
    D_SHOW_METHOD sm
    where cmp.ID = smc.FILTER_KIND_C
    and cmp.UNITLIST = 'MKB10'
    and cmp.CODE = 'RETURN_CODE'
    and sm.ID = smc.PID
order by 4,3

```

3.1.8 Где используется данный метод показа

3.1.8.1 В полях на визитах

```

select vt.VT_CODE||' - '||vt.VT_NAME TEMPLATE, --шаблон
vtb.VT_CODE||' - '||vtb.VT_NAME TAB, --вкладка
tf.F_CODE||' - '||tf.F_NAME TAB_FIELDS, --поле вкладки
c.VC_CODE||' - '||c.VC_NAME CONTAINER, --контейнер
cf.F_CODE||' - '||cf.F_NAME CON_FIELD --поле контейнера
from D_VISIT_CON_FIELDS cf,
D_SHOW_METHOD sm,
D_VISIT_CONTAINERS c,
D_VISIT_TAB_FIELDS tf,
D_VISIT_TABS vtb,
D_VIS_TEMP_TABS tt,
D_VISIT_TEMPLATES vt
where sm.UNITCODE = 'DIAGNOSIS_STAGES' --Код раздела
and sm.CODE = 'DEFAULT' --Код Композиции

```

```

and sm.ID = cf.SHOW_METHOD
and c.ID = cf.PID
and c.ID = tf.CONTAINER
and vtb.ID = tf.PID
and vtb.ID = tt.VISIT_TAB
and vt.ID = tt.PID
union
select vt.VT_CODE||' - '||vt.VT_NAME TEMPLATE, --шаблон
vtb.VT_CODE||' - '||vtb.VT_NAME TAB, --вкладка
tf.F_CODE||' - '||tf.F_NAME TAB_FIELDS, --поле вкладки
null,
null
from D_SHOW_METHOD sm,
D_VISIT_TAB_FIELDS tf,
D_VISIT_TABS vtb,
D_VIS_TEMP_TABS tt,
D_VISIT_TEMPLATES vt
where sm.UNITCODE = 'DIAGNOSIS_STAGES' --Код раздела
and sm.CODE = 'DEFAULT' --Код Композиции
and sm.ID = tf.SHOW_METHOD
and vtb.ID = tf.PID
and vtb.ID = tt.VISIT_TAB
and vt.ID = tt.PID
union
select vt.VT_CODE||' - '||vt.VT_NAME TEMPLATE, --шаблон
vtb.VT_CODE||' - '||vtb.VT_NAME TAB, --вкладка
tf.F_CODE||' - '||tf.F_NAME TAB_FIELDS, --поле вкладки
null,
null
from D_SHOW_METHOD sm,
D_VISIT_CONTAINERS c,
D_VISIT_TAB_FIELDS tf,
D_VISIT_TABS vtb,
D_VIS_TEMP_TABS tt,
D_VISIT_TEMPLATES vt
where sm.UNITCODE = 'DIAGNOSIS_STAGES' --Код раздела
and sm.CODE = 'DEFAULT' --Код Композиции
and sm.ID = c.SHOW_METHOD
and c.ID = tf.CONTAINER
and vtb.ID = tf.PID
and vtb.ID = tt.VISIT_TAB
and vt.ID = tt.PID
order by 1,2,3,4,5

```

3.1.8.2 В фильтрах

```

select sm.UNITCODE||' - '||sm.CODE FILTER_KIND_SM, --фильтр по
методу показа
smc.NAME SHOW_METHOD_COL, --поле метода показа
sm1.UNITCODE||' - '||sm1.CODE SHOW_METHOD, --метод показа
sm1.FORM_ADD,
sm1.FORM_UPD,
sm1.FORM_COPY,

```

```

sm1.POPUP_FORM,
(select d_stragg_ex(d_tp_stragg_rec(cmp1.UNITLIST||' -
'||cmp1.CODE||' ' ||cmp1.FORM_MAIN,chr(10),'ASC',null,null))
from D_COMPOSITION cmp1,
D_COMPOSITION_METHOD cm
where cmp1.ID = cm.PID
and cm.SHOW_METHOD = sm1.ID) COMPOSITIONS_HAVE_SM--в каких
композициях есть этот метод показа
from D_SHOW_METHOD_COLS smc,
D_SHOW_METHOD sm,
D_SHOW_METHOD sm1
where sm.ID = smc.FILTER_KIND_SM
and sm.UNITCODE = 'DEPS' --Код раздела
and sm.CODE = 'LIST' --Код метода показа раздела
and sm1.ID = smc.PID
order by 1,2,3

```

3.1.9 Узнать версию МИС

```

select * from (select * from D_DDL_VERSIONS order by VER_DATE desc)
where rownum = 1

```

3.1.10 Проверить блокировки БД

```

select lpad(' ', (level - 1) * 2, ' ') || decode(level, 1, '', '|-
') || s.SID SID,
NVL(s.USERNAME, '(oracle)') USERNAME,
s.OSUSER,
s.TERMINAL,
sq.SQL_TEXT,
s.PROGRAM,
lpad(TO_CHAR(trunc(s.LAST_CALL_ET / 3600)), 2, '0') || ' ч. ' ||
lpad(TO_CHAR(trunc(s.LAST_CALL_ET / 60) - trunc(s.LAST_CALL_ET /
3600)
* 60), 2, '0') || ' мин.' LAST_CALL_ET
FROM GV$SESSION s,
Gv$sql sq
where sq.SQL_ID(+) =s .SQL_ID
and ((CONNECT_BY_ISLEAF = 0) or (level > 1))
start with s.BLOCKING_SESSION is null
connect by prior s.SID = s.BLOCKING_SESSION;

```

3.2 Интеграция с ФЭР - скрипты для размножения соответствий услуг ФЭР в облаке и их импорт

3.2.1 Поиск полей с js-скриптом (позволяет найти на конкретном шаблоне поля, где заполнено поле «Функция (oncreate):»)

```
select vt.VT_CODE,vt.VT_NAME,vt.SCRIPT,
vf.f_code,vf.f_name,vf.oncreate_func
from D_VIS_TEMP_TABS vtt,
D_V_VISIT_TABS vt,
D_V_VISIT_TAB_FIELDS vf
where vf.PID= vt.id
and vtt.visit_tab=vt.id
and vf.oncreate_func is not null
and vtt.PID= <ID шаблона> -- ID шаблона можно узнать с помощью
запроса select * from D_VISIT_TEMPLATES where VT_CODE = 'код шаблона'
поле ID и подставляем в скрипт.
```

3.3 Запросы для определения основных показателей МИС - за весь период работы системы

3.3.1 Количество пациентов (людей в базе на которых есть АК) - если несколько ЛПУ

```
select count(1) from (select distinct(t.agent) from d_persmedcard
t);
```

3.3.2 Количество амбулаторных карт пациентов

```
select count(1) from d_persmedcard t;
```

3.3.3 Количество ИБ

```
select count(1) from d_hosp_histories t;
```

3.3.4 Количество посещений

3.3.4.1 всего услуг

```
select count(1) from d_v_direction_services t1, d_visits t2 where
t2.PID=t1.ID;
```

3.3.4.2 оказанные с типом посещение

```
select count(1) from d_v_direction_services t1, d_visits t2 where
t1.SERVICE_TYPE = 3 and t2.PID=t1.ID;
```

3.3.4.3 оказанные исследования и процедуры

```
select count(1) from d_v_direction_services t1, d_visits t2 where  
t1.SERVICE_TYPE in (1, 0) and t2.PID=t1.ID;
```

3.3.4.4 оказанные операции

```
select count(1) from d_v_direction_services t1, d_visits t2 where  
t1.SERVICE_TYPE = 2 and t2.PID=t1.ID;
```

3.3.4.5 другие услуги

```
select count(1) from d_v_direction_services t1, d_visits t2 where  
t1.SERVICE_TYPE not in (0, 1, 3) and t2.PID=t1.ID;
```

3.3.5 Количество записей на услуги

```
select count(1) from d_direction_services t1;
```

3.3.6 Количество амбулаторных талонов

Тип случая: 0 - заболевание, 1 - осмотр, 2 - диспансеризация, 3 - госпитализация

– всего:

```
select count(1) from d_amb_talons t;
```

– 0 – заболевание:

```
select count(1) from d_amb_talons t where t.at_type=0;
```

– 1 – осмотр:

```
select count(1) from d_amb_talons t where t.at_type=1;
```

– 2 – диспансеризация:

```
select count(1) from d_amb_talons t where t.at_type=2;
```

– 3 – госпитализация:

```
select count(1) from d_amb_talons t where t.at_type=3;
```

3.3.7 Количество пациентов с амбулаторными талонами

Тип случая: 0 - заболевание, 1 - осмотр, 2 - диспансеризация, 3 - госпитализация

– всего:

```
select count(1) from (select distinct(t.PERSMEDCARD) from  
d_amb_talons t);
```

– 0 – заболевание:

```
select count(1) from (select distinct(t.PERSMEDCARD) from  
d_amb_talons t where t.at_type=0);
```

– 1 – осмотр:

```
select count(1) from (select distinct(t.PERSMEDCARD) from  
d_amb_talons t where t.at_type=1);
```

– 2 – диспансеризация:

```
select count(1) from (select distinct(t.PERSMEDCARD) from
d_amb_talons t where t.at_type=2);
```

- 3 – госпитализация:

```
select count(1) from (select distinct(t.PERSMEDCARD) from
d_amb_talons t where t.at_type=3);
```

3.3.8 Количество пациентов, которым хоть раз была оказана услуга

```
select count(1) from (select distinct(t3.Patient) from
d_direction_services t1, d_visits t2, d_directions t3 where
t2.PID=t1.ID and t3.ID = t1.pid);
```

3.4 Скрипт для переназначения прав просмотра мед.словарей со специальности на роль

```
declare
sSTR VARCHAR2(500);
nSPEC NUMBER(17);
nROLE NUMBER(17);
nLPU NUMBER(17);
nRIGHT NUMBER(17);
n1 NUMBER;
n2 NUMBER;
n3 NUMBER;
begin
sSTR := 'Наименование спец - Наименование роли';
--'специальность - роль' Мед словари
begin
select t.ID
into nSPEC
from D_SPECIALITIES t,
D_LPUVERLIST lv
where t.TITLE = trim(substr(sSTR,1,instr(sSTR,'-')-1))
and t.VERSION = lv.VERSION
and lv.LPU = nLPU;
exception when NO_DATA_FOUND then D_P_EXC('Не найдена
специальность : '||trim(substr(sSTR,1,instr(sSTR,'-')-1)));
end;
begin
select g.ID
into nROLE
from D_ROLES g
where g.NAME = trim(substr(sSTR,instr(sSTR,'-')+1));
exception when NO_DATA_FOUND then D_P_EXC('Не найдена роль :
'||trim(substr(sSTR,instr(sSTR,'-')+1)));
end;
-- Call the procedure
--return;
d_pkg_cse_accesses.copy_rights(pnlpu => nLPU,
psunitcode => 'ADD_DIR_ROWS',
```



```

pnsource_from => nSPEC,
pnsource_to => nROLE,
pnstype_from => 1,
pnstype_to => 3,
pnright => nRIGHT, -- отображать на приеме
pnunit_id => null);
select count(1)
into n1
from D_CSE_ACCESSES k,
D_CSE_ACS_SPECS ks,
D_CSE_ACS_SPEC_RIGHTS ksp
where k.UNITCODE = 'ADD_DIR_ROWS'
and k.LPU = nLPU
and ks.PID = k.ID
and ks.SPECIALITY = nSPEC
and ksp.PID = ks.ID
and ksp.RIGHT = nRIGHT;
select count(1)
into n2
from D_CSE_ACCESSES k,
D_CSE_ACS_ROLES ks,
D_CSE_ACS_ROLE_RIGHTS ksp
where k.UNITCODE = 'ADD_DIR_ROWS'
and k.LPU = nLPU
and ks.PID = k.ID
and ks.ROLE = nROLE
and ksp.PID = ks.ID
and ksp.RIGHT = nRIGHT;
for x in (select ksp.ID
from D_CSE_ACCESSES k,
D_CSE_ACS_SPECS ks,
D_CSE_ACS_SPEC_RIGHTS ksp
where k.UNITCODE = 'ADD_DIR_ROWS'
and k.LPU = nLPU
and ks.PID = k.ID
and ks.SPECIALITY = nSPEC
and ksp.PID = ks.ID
and ksp.RIGHT = nRIGHT)
loop
D_PKG_CSE_ACS_SPEC_RIGHTS.DEL(x.ID, nLPU);
end loop;
select count(1)
into n3
from D_CSE_ACCESSES k,
D_CSE_ACS_SPECS ks,
D_CSE_ACS_SPEC_RIGHTS ksp
where k.UNITCODE = 'ADD_DIR_ROWS'
and k.LPU = nLPU
and ks.PID = k.ID
and ks.SPECIALITY = nSPEC
and ksp.PID = ks.ID
and ksp.RIGHT = nRIGHT;
DBMS_OUTPUT.PUT_LINE(n1||'/'||n3||' --- '||n2);

```

end;

3.5 Скрипт для переназначения прав просмотра диагнозов МКБ10 со специальности на роль

```
declare
  sSTR VARCHAR2(500);
  nSPEC NUMBER(17);
  nROLE NUMBER(17);
  nLPU NUMBER(17);
  nRIGHT NUMBER(17);
  sUNITCODE VARCHAR2(30);
  sRIGHT VARCHAR2(1);
  n1 NUMBER;
  n2 NUMBER;
  n3 NUMBER;
begin
  nLPU := 10903;
  sUNITCODE := 'МКБ10';
  sRIGHT := '1';
  sSTR := 'Наименование спец - Наименование роли';
  -- 'специальность - роль' Диагнозы МКБ
  nRIGHT := D_PKG_CSE_RIGHTS.GET_ID(sUNITCODE, nLPU, sRIGHT, 1);
  begin
    select t.ID
    into nSPEC
    from D_SPECIALITIES t,
    D_LPUVERLIST lv
    where t.TITLE = trim(substr(sSTR,1,instr(sSTR,'-')-1))
    and t.VERSION = lv.VERSION
    and lv.LPU = nLPU;
    exception when NO_DATA_FOUND then D_P_EXC('Не найдена
специальность : '||trim(substr(sSTR,1,instr(sSTR,'-')-1)));
  end;
  begin
    select g.ID
    into nROLE
    from D_ROLES g
    where g.NAME = trim(substr(sSTR,instr(sSTR,'-')+1));
    exception when NO_DATA_FOUND then D_P_EXC('Не найдена роль :
'||trim(substr(sSTR,instr(sSTR,'-')+1)));
  end;
  d_pkg_cse_accesses.copy_rights(pnlpu => nLPU,
  psunitcode => sUNITCODE,
  pnsouce_from => nSPEC,
  pnsouce_to => nROLE,
  pntype_from => 1,
  pntype_to => 3,
  pnright => nRIGHT, -- отображать на приеме
  pnunit_id => null);
  select count(1)
```

```

into n1
from D_CSE_ACCESSES k,
D_CSE_ACS_SPECS ks,
D_CSE_ACS_SPEC_RIGHTS ksp
where k.UNITCODE = sUNITCODE
and k.LPU = nLPU
and ks.PID = k.ID
and ks.SPECIALITY = nSPEC
and ksp.PID = ks.ID
and ksp.RIGHT = nRIGHT;
select count(1)
into n2
from D_CSE_ACCESSES k,
D_CSE_ACS_ROLES ks,
D_CSE_ACS_ROLE_RIGHTS ksp
where k.UNITCODE = sUNITCODE
and k.LPU = nLPU
and ks.PID = k.ID
and ks.ROLE = nROLE
and ksp.PID = ks.ID
and ksp.RIGHT = nRIGHT;
for x in (select ksp.ID,
ksp.PID,
(select count(1)
from D_CSE_ACS_SPEC_RIGHTS a
where a.PID = ksp.PID
and a.ID != ksp.ID) CNT
from D_CSE_ACCESSES k,
D_CSE_ACS_SPECS ks,
D_CSE_ACS_SPEC_RIGHTS ksp
where k.UNITCODE = sUNITCODE
and k.LPU = nLPU
and ks.PID = k.ID
and ks.SPECIALITY = nSPEC
and ksp.PID = ks.ID
and ksp.RIGHT = nRIGHT)
loop
D_PKG_CSE_ACS_SPEC_RIGHTS.DEL(x.ID, nLPU);
if x.CNT = 0 then
D_PKG_CSE_ACS_SPECS.DEL(x.PID, nLPU);
end if;
end loop;
select count(1)
into n3
from D_CSE_ACCESSES k,
D_CSE_ACS_SPECS ks,
D_CSE_ACS_SPEC_RIGHTS ksp
where k.UNITCODE = sUNITCODE
and k.LPU = nLPU
and ks.PID = k.ID
and ks.SPECIALITY = nSPEC
and ksp.PID = ks.ID
and ksp.RIGHT = nRIGHT;

```

```

DBMS_OUTPUT.PUT_LINE(n1||'/'||n3||' --- '||n2);
end;

```

3.6 Просмотр прав на меню для Сотрудников

Отображает сотрудника, его роль и доступные пункты главного меню.

```

select e.EMP_FULL_NAME,
       r.NAME ROLE_NAME,
       mm.MENU_CAPTION,
       mm.ID,
       (select SYS_CONNECT_BY_PATH(m.MENU_CAPTION, '|')
        from D_MAIN_MENUS m
        where m.ID = mm.ID
        START WITH m.PID is null
        CONNECT BY PRIOR m.ID = m.PID) MENU_PATH
from D_MAIN_MENUS mm,
     D_MAIN_MENU_ROLES mmr,
     D_ROLES R,
     D_USERROLES ur,
     D_V_EMPLOYERS e,
     D_V_LPU l
where ur.SYSUSER = e.SYSUSER_id
and ur.ROLEID = r.ID
and mmr.ROLEID = r.ID
and mm.ID = mmr.MAIN_MENU
and l.ID = e.LPU

```

3.7 Просмотр шаблонов, использующих наш медицинский словарь

```

select tmp.VT_CODE, tmp.VT_NAME, vt.VT_CODE, c.C_NAME
from D_VISIT_TAB_FIELDS vtf,
     D_VISIT_TABS vt,
     D_VIS_TEMP_TABS vtt,
     D_VISIT_TEMPLATES tmp,
     D_ADD_DIRECTORIES ad,
     D_ADD_DIR_COLUMNS adc,
     D_CATALOGS c
where vtf.PID = vt.ID
and vt.ID = vtt.VISIT_TAB
and vtt.PID = tmp.ID
and c.ID = tmp.CID
and adc.PID = ad.ID
and adc.ID = vtf.ADD_DIR_COLUMN
and ad.AD_CODE like 'ANAM%' - Код мед словаря или часть его кода

```

через %

3.8 Просмотр прав на разделы для Сотрудников

```
select *
  from (select e.ID EMP_ID,e.EMP_FULL_NAME,r.NAME
ROLE_NAME,urp.UNITCODE,nvl(ubp.UNITBP,'VIEW') UNITBP
  from D_V_EMPLOYERS e,
  D_USERROLES usr,
  D_ROLES r,
  D_URPRIVS urp,
  D_UNITBPPRIVS ubp
  where r.ID = urp.ROLEID
  and r.ID = usr.ROLEID
  and usr.SYSUSER = e.SYSUSER_ID
  and ubp.PID(+)=urp.ID
  union
  select e.ID EMP_ID,e.EMP_FULL_NAME,null
ROLE_NAME,urp.UNITCODE,nvl(ubp.UNITBP,'VIEW') UNITBP
  from D_V_EMPLOYERS e,
  D_URPRIVS urp,
  D_UNITBPPRIVS ubp
  where e.SYSUSER=urp.USERNAME
  and ubp.PID(+)=urp.ID
) t where t.UNITCODE like '%' and t.EMP_FULL_NAME like '%Сидор%' --
and t.EMP_ID=10
```

3.9 Просмотр прав на разделы для Сотрудников (с каталогами)

```
select *
  from (select e.ID EMP_ID,e.EMP_FULL_NAME,r.NAME
ROLE_NAME,urp.UNITCODE,nvl(ubp.UNITBP,'VIEW') UNITBP, cat.C_NAME
  from D_V_EMPLOYERS e,
  D_USERROLES usr,
  D_ROLES r,
  D_URPRIVS urp,
  D_UNITBPPRIVS ubp,
  D_CATALOGS cat
  where r.ID = urp.ROLEID
  and r.ID = usr.ROLEID
  and usr.SYSUSER = e.SYSUSER_ID
  and ubp.PID(+)=urp.ID
  and cat.ID(+) = urp.CATALOG
  union
  select e.ID EMP_ID,e.EMP_FULL_NAME,null
ROLE_NAME,urp.UNITCODE,nvl(ubp.UNITBP,'VIEW') UNITBP, cat.C_NAME
  from D_V_EMPLOYERS e,
  D_URPRIVS urp,
  D_UNITBPPRIVS ubp,
  D_CATALOGS cat
  where e.SYSUSER=urp.USERNAME
  and ubp.PID(+) = urp.ID
  and cat.ID(+) = urp.CATALOG
```

```

) t where t.UNITCODE like 'NOM%' and t.EMP_FULL_NAME like '%Сидор%'
--and t.EMP_ID=10

```

3.10 Заведения Администраторов ЛПУ с назначением Ролей

```

declare
  nROLE_ID_MIN NUMBER(17);
  nROLE_ID NUMBER(17);
  nNEW_USER NUMBER(17);
  nNEW_USERROLE NUMBER(17);
  sROLE_NAME VARCHAR2(32) := 'Администратор'; -- Роль, которую
назначим администраторам, помимо минимальной
  sUSER_NAME VARCHAR2(20) := 'ADMIN'; -- Логин пользователя, к нему
прибавляется автоматом "_КодЛПУ"
  sSURNAME VARCHAR2(50) := 'Администратор'; -- Фамилия пользователя,
к ней прибавляется "_КодЛПУ"
  sNAME VARCHAR2(50) := 'Барс'; -- Имя
  sLASTNAME VARCHAR2(50) := 'МИС'; -- Отчество
  sPASS VARCHAR2(10) := '123'; -- Пароль
  nCAT NUMBER(17);
  nNEW_AG NUMBER(17);
  nSPEC NUMBER(17);
  nJOB NUMBER(17);
begin
  select max(r.ID)
  into nROLE_ID
  from D_ROLES r
  where r.NAME = sROLE_NAME;
  for cr in (select * from D_LPU )
  loop
    select max(r.ID)
    into nROLE_ID_MIN
    from D_ROLES r
    where r.NAME = 'Минимальная_' || cr.CODE_LPU;
    select max(t.ID)
    into nNEW_USER
    from D_USERS t
    where t.USERNAME = sUSER_NAME || '_' || cr.CODE_LPU;
    if nNEW_USER is null then
      begin
        D_PKG_USERS.ADD(nNEW_USER,
          cr.ID,
          null,
          sUSER_NAME || '_' || cr.CODE_LPU,
          sPASS,
          sSURNAME || '_' || cr.CODE_LPU || ' ' || sNAME || ' ' || sLASTNAME,
          1,
          1);
      exception when OTHERS then
        nNEW_USER := null;
      end;
    end;
  end;

```

```

end if;
if nROLE_ID_MIN is not null and nNEW_USER is not null then
select max(t.ID)
into nNEW_USERROLE
from D_USERROLES t
where t.ROLEID = nROLE_ID_MIN
and t.SYSUSER = nNEW_USER;
if nNEW_USERROLE is null then
D_PKG_USERROLES.ADD(nNEW_USERROLE,cr.ID,nROLE_ID_MIN,nNEW_USER);
end if;
end if;
if nROLE_ID is not null and nNEW_USER is not null then
select max(t.ID)
into nNEW_USERROLE
from D_USERROLES t
where t.ROLEID = nROLE_ID
and t.SYSUSER = nNEW_USER;
if nNEW_USERROLE is null then
D_PKG_USERROLES.ADD(nNEW_USERROLE,cr.ID,nROLE_ID,nNEW_USER);
end if;
end if;
select max(s.ID)
into nSPEC
from D_SPECIALITIES s
where s.CODE = '1'
and s.VERSION =
D_PKG_VERSIONS.GET_VERSION_BY_LPU(0,cr.ID,'SPECIALITIES');
select max(j.ID)
into nJOB
from D_JOBTTITLES j
where j.CODE = '1'
and j.VERSION =
D_PKG_VERSIONS.GET_VERSION_BY_LPU(0,cr.ID,'JOBTTITLES');
D_PKG_CATALOGS.FIND_ROOT_CATALOG(0,cr.id,'EMPLOYERS',nCAT);
if nNEW_USER is not null then
begin
D_PKG_EMPLOYERS.ADD_FULL(
nNEW_AG,
cr.ID,
nCAT,
null, --ID контрагента
sNAME, --Имя
sSURNAME||'_'||cr.CODE_LPU, --Фамилия
sLASTNAME, --Отчество
'01.01.1990', --Дата рождения
1, --Пол : 0 - женский, 1 - мужской
null, --СНИЛС
nJOB, --Должность
nSPEC, --Специальность
null, --Код врача
null, --Отделение
null, --Регистрационный код
null, --Специальность по образованию

```

```

null, --Квалификационная категория
sysdate,
nNEW_USER, --Пользователь
null, --Подпись врача в отчетах
null --Табельный номер
);
exception when OTHERS then
DBMS_OUTPUT.put_line(sqlerrm||' '||sUSER_NAME||'_'||cr.CODE_LPU);
end;
end if;
end loop;
end;

```

3.11 Клонирование пользовательских процедур в другие ЛПУ по каталогу

```

declare
  nCID D_PKG_STD.tREF;
  nTEMP D_PKG_STD.tREF;
  nTEMP2 D_PKG_STD.tREF;
  nUP_ID D_PKG_STD.tREF;
begin
  for rLPU in (select t.ID
  from D_LPU t
  where t.ID != :LPU_FROM
  --существует каталог ФЕР в пользовательских процедурах
  and exists (select null from D_CATALOGS t1 where t1.C_UNITCODE =
'USERPROCS' and t1.C_NAME = 'ФЕР' and t1.LPU = t.ID)
  --существует пользовательская процедура с кодом ADD_DIR_SERV
  and exists (select null from D_USERPROCS t1 where t1.LPU = t.ID
and t1.PR_CODE = 'ADD_DIR_SERV'))
  loop
  begin
  select t1.ID
  into nCID
  from D_CATALOGS t1
  where t1.C_UNITCODE = 'USERPROCS' and t1.C_NAME = 'ФЕР' and t1.LPU
= rLPU.ID;
  exception when NO_DATA_FOUND then
  D_PKG_CATALOGS.FIND_ROOT_CATALOG(1, rLPU.ID, 'USERPROCS', nTEMP);
  D_PKG_CATALOGS.ADD(nCID, rLPU.ID, nTEMP, 'ФЕР');
  end;
  for rUP in (select t.*, t2.ID UP_ID
  from D_USERPROCS t,
  D_CATALOGS t1,
  D_USERPROCS t2
  where t1.ID = t.CID
  and t1.C_UNITCODE = 'USERPROCS'
  and t1.C_NAME = 'ФЕР'
  and t.LPU = :LPU_FROM
  and t2.LPU(+) = rLPU.ID
  and t2.PR_CODE(+) = t.PR_CODE)

```



```

loop
--Добавление/Исправление ПП
if rUP.UP_ID is not null then
nUP_ID := rUP.UP_ID;
D_PKG_USERPROCS.UPD(pnid => nUP_ID,
pnlpu => rLPU.ID,
pspr_code => rUP.PR_CODE,
pspr_name => rUP.PR_NAME,
pspr_note => rUP.PR_NOTE,
pnpr_type => rUP.PR_TYPE,
pnpr_exec_type => rUP.PR_EXEC_TYPE,
psstoredproc => rUP.STOREDPROC,
psnnmblock => rUP.NNMBLOCK,
psschema => rUP.SCHEMA,
pnoverloadnumb => rUP.OVERLOADNUMB);
else
D_PKG_USERPROCS.ADD(PND_INSERT_ID => nUP_ID,
pnlpu => rLPU.ID,
PNCID => nCID,
pspr_code => rUP.PR_CODE,
pspr_name => rUP.PR_NAME,
pspr_note => rUP.PR_NOTE,
pnpr_type => rUP.PR_TYPE,
pnpr_exec_type => rUP.PR_EXEC_TYPE,
psstoredproc => rUP.STOREDPROC,
psnnmblock => rUP.NNMBLOCK,
psschema => rUP.SCHEMA,
pnoverloadnumb => rUP.OVERLOADNUMB);
end if;
--Добавление связей
delete D_USERPROCS_BPLINKS t where t.PID in (select z.ID from
D_USERPROCS_LINKS z where z.PID = nUP_ID);
delete D_USERPROCS_LINKS t where t.PID = nUP_ID;
for rLINK in (select * from D_USERPROCS_LINKS t where t.PID =
rUP.ID)
loop
D_PKG_USERPROCS_LINKS.ADD(pnd_insert_id => nTEMP,
pnlpu => rLPU.ID,
pnpid => nUP_ID,
psunitcode => rLINK.unitcode,
pspriv_name => rLINK.priv_name,
pntranz_end => rLINK.tranz_end,
pnconfirm => rLINK.confirm,
pstext_confirm => rLINK.text_confirm,
pnrefresh_data => rLINK.refresh_data);
for rBP in (select * from D_USERPROCS_BPLINKS t where t.PID =
rLINK.ID)
loop
D_PKG_USERPROCS_BPLINKS.ADD(pnd_insert_id => nTEMP2,
pnlpu => rLPU.ID,
pnpid => nTEMP,
psunitbp => rBP.unitbp,
pnbef => rBP.bef,

```

```

pnaft => rBP.aft,
pnexec_order => rBP.exec_order);
end loop;
end loop;
end loop;
end loop;
end;

```

3.12 Массовое закрытие ТАПов, открытых до определенной даты

```

declare
  nREF_RESULT NUMBER;
begin
  D_PKG_LOG.SET_FLAG(2);
  select t.ID into nREF_RESULT from D_VISITRESULTS t where VR_CODE =
'13';
  for r in (select t.*, v.VIS_DATE
from D_AMB_TALONS t, D_AMB_TALON_VISITS v
where t.END_DATE is null
and t.ID = v.PID
and v.IS_LAST = 1
and t.AT_DATE <=
to_date('НАПИСАТЬ_ДАТУ', 'dd.mm.yyyy')) loop
  d_pkg_amb_talons.set_close(pnid => r.ID,
  pnlpu => r.lpu,
  pnis_close => 1,
  pdend_date => r.VIS_DATE,
  pnref_result => nREF_RESULT);
  end loop;
  for r in (select t.*
from D_AMB_TALONS t
where t.END_DATE is null
and not exists
(select null
from D_AMB_TALON_VISITS atv
where atv.PID = t.ID)
and t.AT_DATE <=
to_date('НАПИСАТЬ_ДАТУ', 'dd.mm.yyyy')) loop
  d_pkg_amb_talons.set_close(pnid => r.ID,
  pnlpu => r.lpu,
  pnis_close => 1,
  pdend_date => to_date('НАПИСАТЬ_ДАТУ',
  'dd.mm.yyyy'),
  pnref_result => nREF_RESULT);
  end loop;
  D_PKG_LOG.RESET_FLAG;
  commit;
end;

```

3.13 Закрытие случаев заболевания прошлого года

Скрипт закрывает незакрытые случаи заболевания максимальной датой из «максимальная дата визита» или «максимальная дата направления» или «максимальная дата амбулаторного визита» для случаев, открытых ранее 2014 года, и типов случаев заболевания не диспансеризация и не стационар.

```
declare
  nUPD NUMBER(17) := 0;
  nDEL NUMBER(17) := 0;
begin
  for cr in ( select d.ID,
    nvl((select nvl(max(v.VISIT_DATE), max(ds.REC_DATE))
    from D_DIRECTION_SERVICES ds,
    D_VISITS v
    where ds.DISEASECASE = d.ID
    and v.PID(+) = ds.ID),
    (select max(atv.VIS_DATE)
    from D_AMB_TALONS amt,
    D_AMB_TALON_VISITS atv
    where amt.DISEASECASE = d.ID
    and atv.PID = amt.ID)) max_date
    from D_DISEASECASES d
    where d.IS_CLOSE = 0
    and d.DC_CLOSEDATE is null
    and d.DC_OPENDATE < '01.01.2014'
    and d.DC_TYPE not in (2,3))
    loop
    if cr.MAX_DATE is null then
    begin
    delete from D_DISEASECASES where ID = cr.ID;
    nDEL := nDEL + 1;
    exception when OTHERS then
    DBMS_OUTPUT.put_line(sqlerrm);
    end;
    else
    update D_DISEASECASES set IS_CLOSE = 1, DC_CLOSEDATE = cr.MAX_DATE
    where ID = cr.ID;
    nUPD := nUPD + 1;
    end if;
    commit;
    end loop;
    DBMS_OUTPUT.put_line('UPD ' || nUPD || ' DEL ' || nDEL);
end;
```

3.14 Копирование типов заявок из одного ЛПУ во все остальные

Копирование типов заявок из ЛПУ, код которого нужно указать в параметрах внизу, во все ЛПУ Системы:

```
declare
  -- Local variables here
  i integer;
begin
  -- Test statements here
  for r in (select * from D_LPU)
  loop
    for t in (select * from D_APPLICATION_TYPES at where at.LPU=:LPU
and at.AT_NAME not in
  (select att.AT_NAME from D_APPLICATION_TYPES att where
att.LPU=r.ID))
  loop
    D_PKG_APPLICATION_TYPES.ADD(pnd_insert_id => i,
    pnlpu => r.ID,
    psat_code =>t.AT_CODE,
    psat_name =>t.AT_NAME,
    pnap_type =>t.AP_TYPE
    );
  end loop;
end loop;
end;
```

3.15 Поиск по отчетам

Скрипт для поиска по отчетам, созданным через web-конструктор:

```
PL/SQL Developer Test script 3.0
21
declare
  sBUFFER VARCHAR2(4000);
  nLIMIT NUMBER(17) := 0;
  nCUR NUMBER(17);
  nPOS NUMBER(17);
begin
  for r in (select t.ID, t.REP_CODE, t.REP_NAME, t.REP_DATA REP_DATA
from D_REPORTS t)
  loop
    nCUR := 0;
    nLIMIT := trunc((dbms_lob.getlength(r.REP_DATA)+1999) / 2000);
    while nCUR < nLIMIT loop
      sBUFFER := UTL_RAW.CAST_TO_VARCHAR2(DBMS_LOB.SUBSTR(r.REP_DATA,
2000, nCUR * 2000 + 1));
      if INSTR(upper(sBUFFER), upper(:SEARCH)) > 0 then
        DBMS_OUTPUT.put_line(r.ID ||' REP_CODE('||r.REP_CODE||')
'||r.REP_NAME);
      exit;
    end loop;
  end loop;
```

```
else  
nCUR := nCUR + 1;  
end if;  
end loop;  
end loop;  
end;
```

