*--Таблицы:*

lu *--Справочники*

service\_group *-- разделы услуг*

srvdep *-- услуги*

price *--Период цен для услуги*

price\_list *--Цена в конкретном МЦ*

doctor *-- Общие карточки врача*

docdep *-- Должности врача*

dep *--Отделения*

docserv *-- Услуги врача по должности*

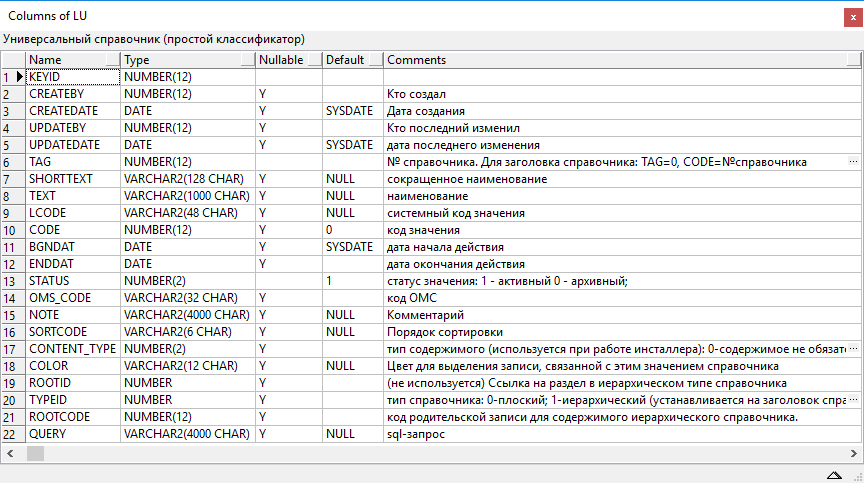
agr *-- Шифры*

discount *-- Скидки на услуги*

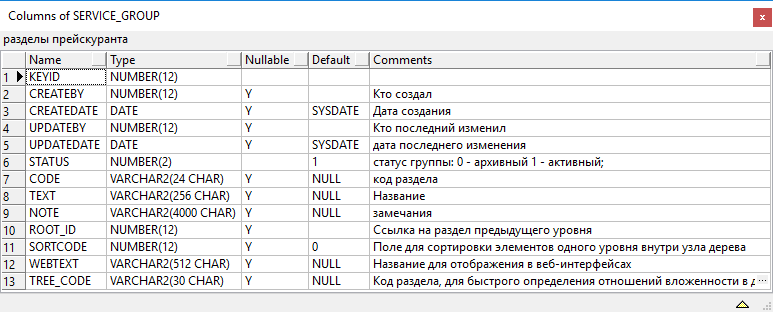
rnumb *–Номерки ??*

*------------------Краткое описание----------*

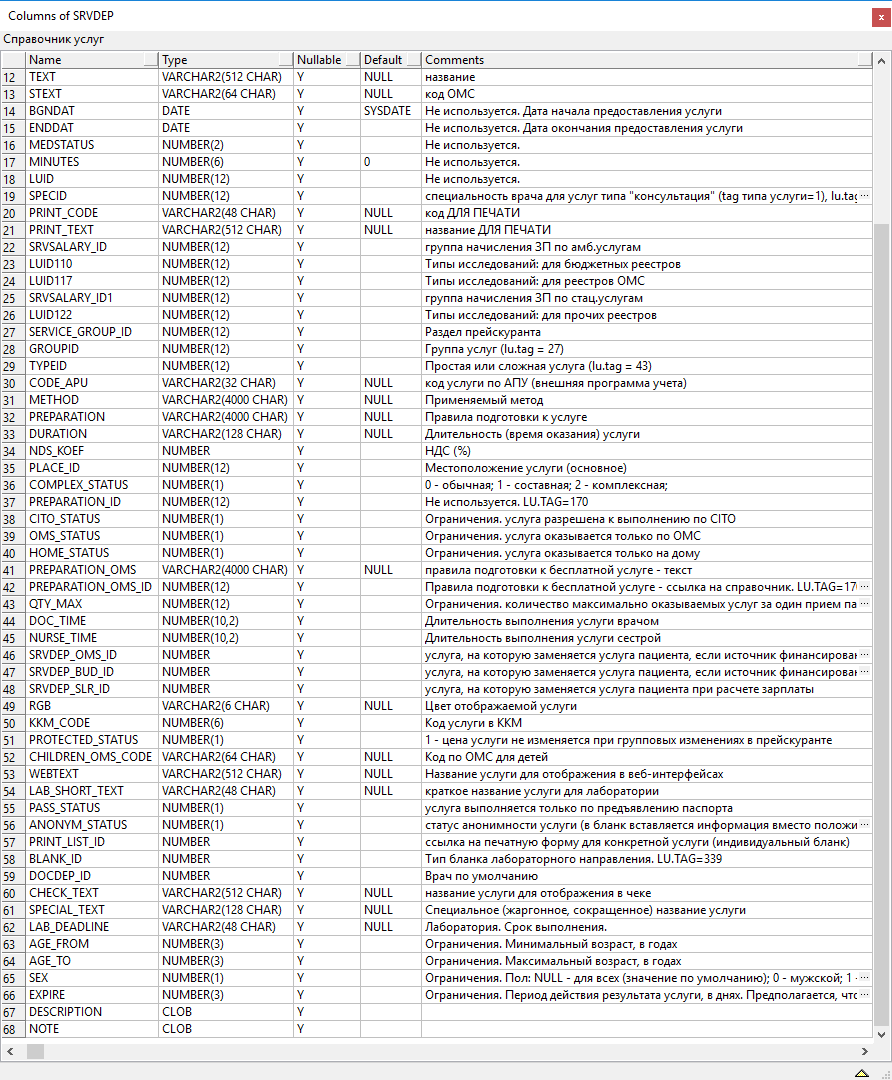
*LU:*



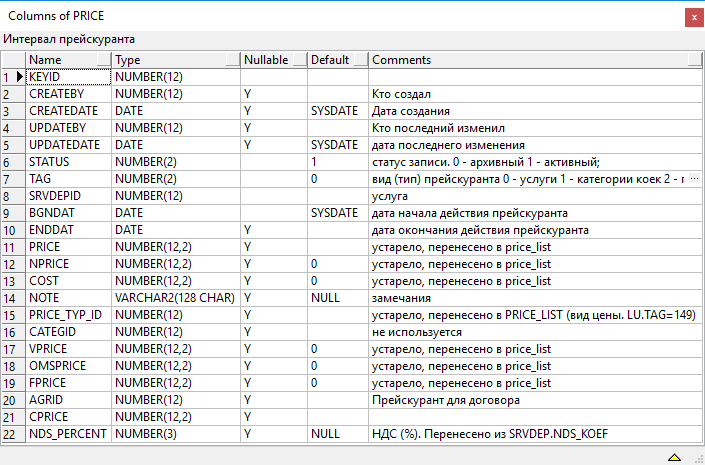
*Service\_group:*



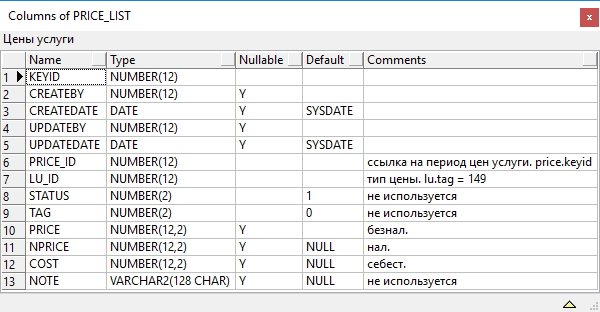
*Srvdep:*



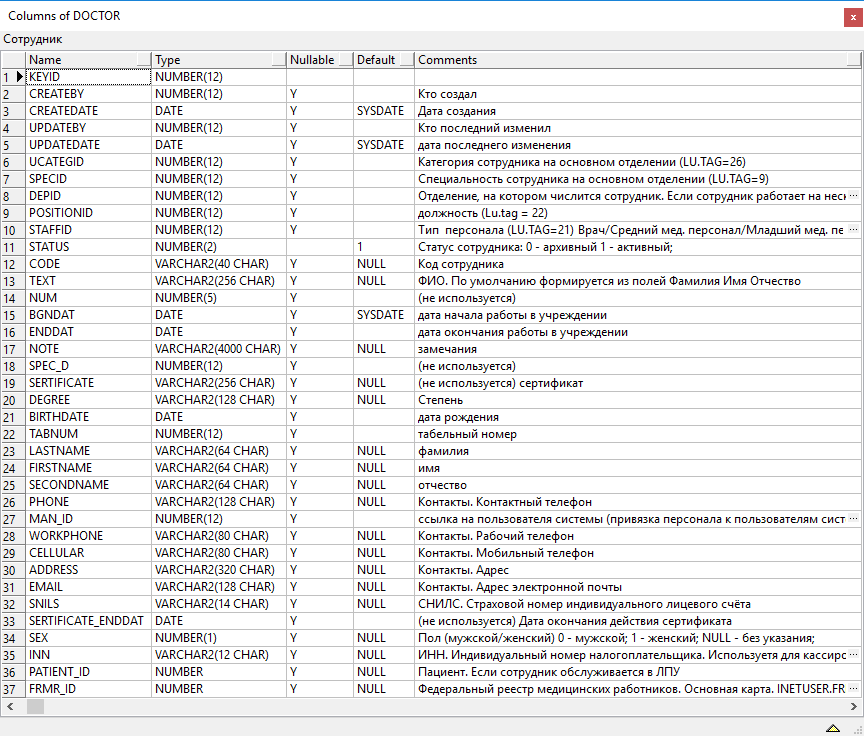
*Price:*



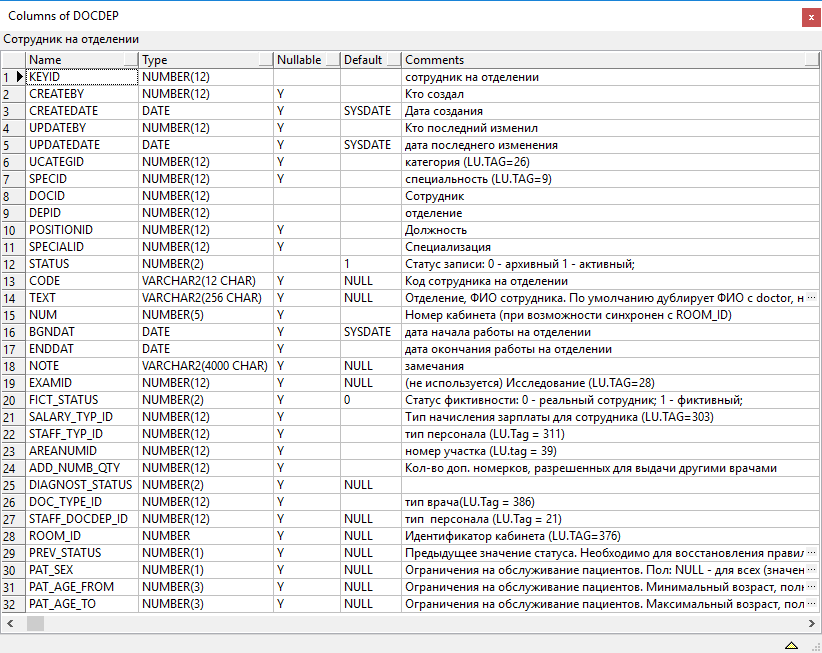
*Price\_list:*



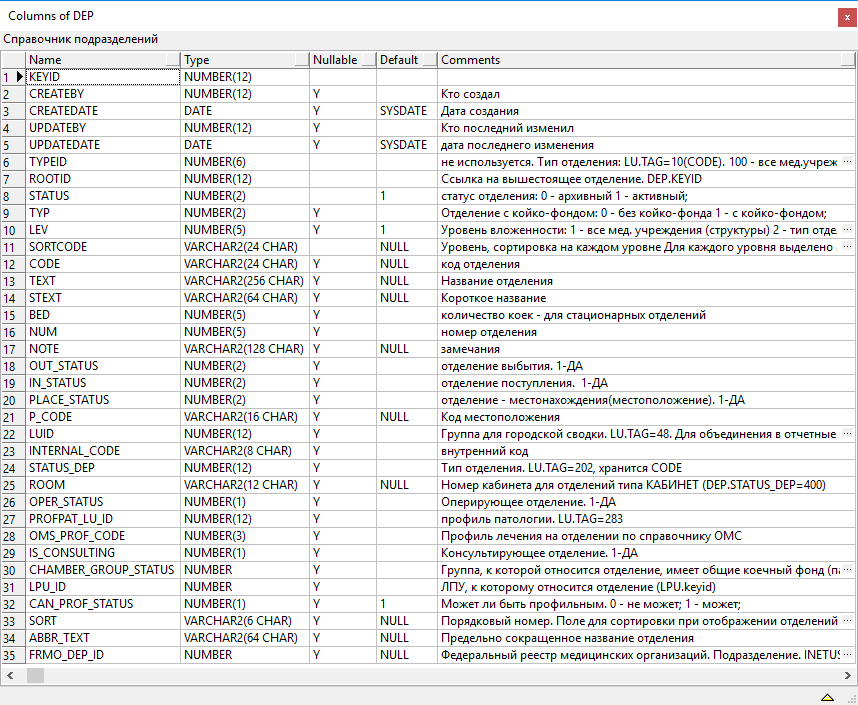
*Doctor:*



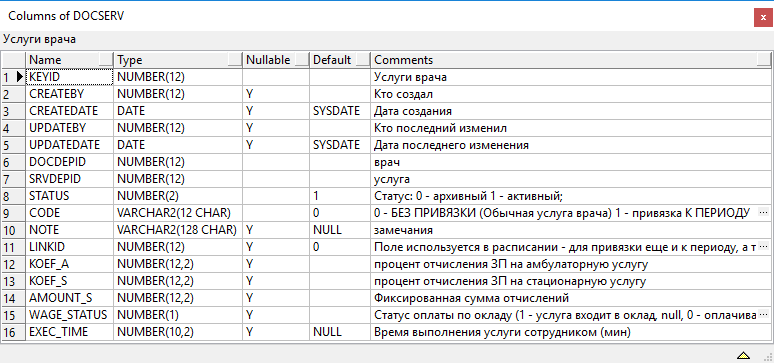
*Docdep:*



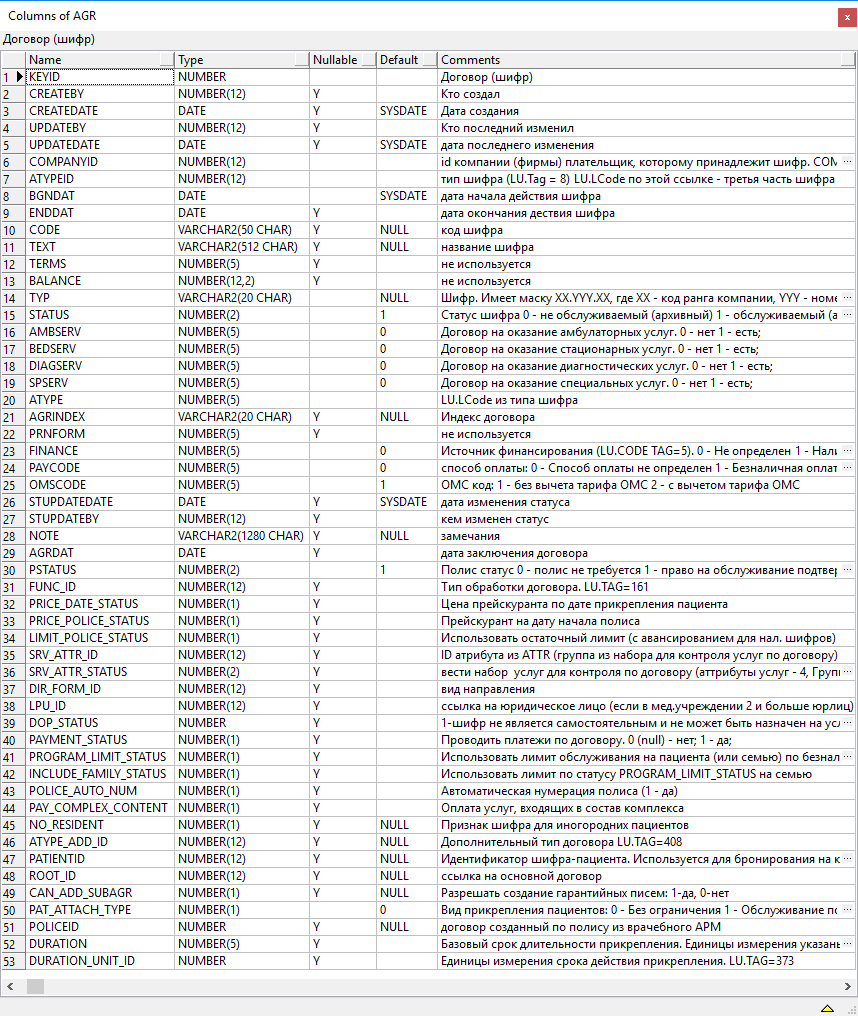
*Dep:*



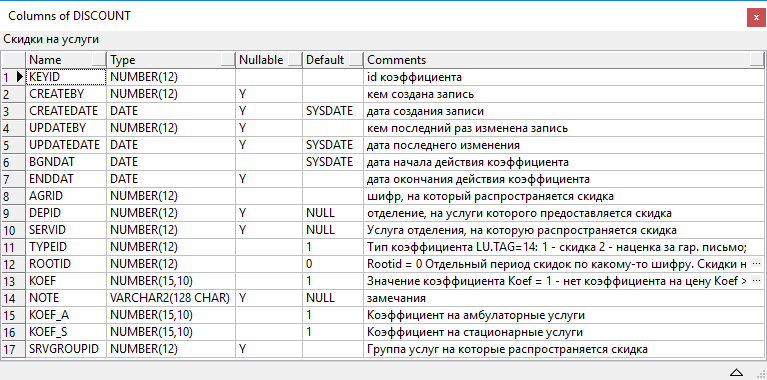
*Docserv:*



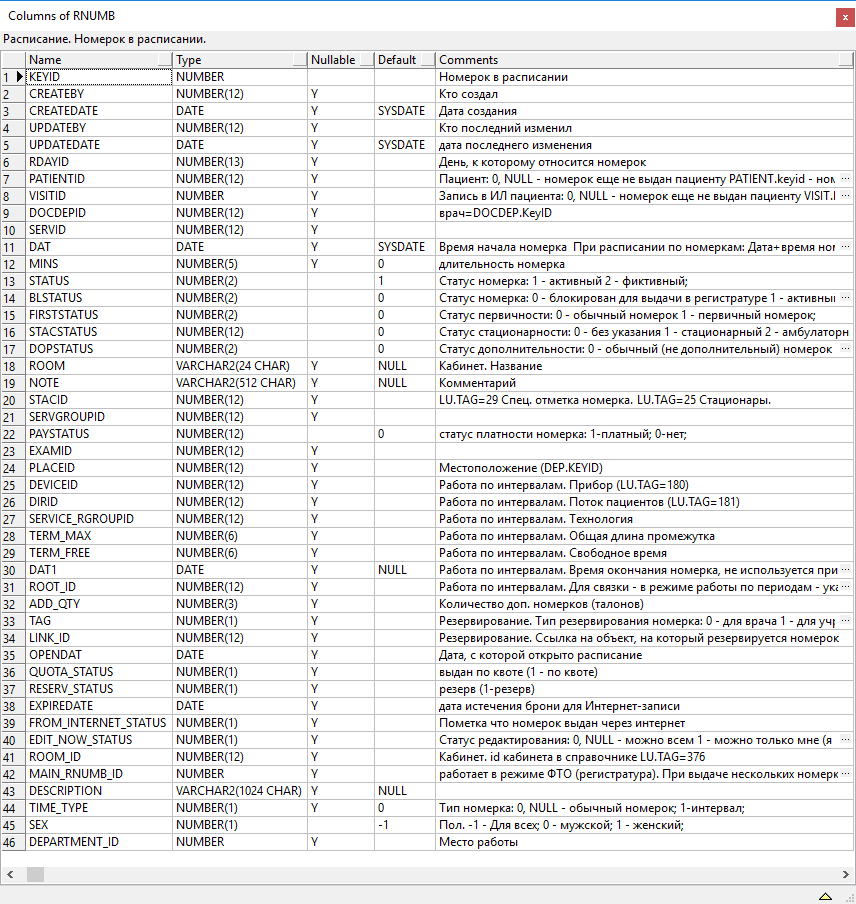
*Agr:*



*Discount:*



*Rnumb:*



*------------------Связи-------------*

*service\_group.keyid = srvdep.service\_groupid*

*price.srvdepid = srvdep.keyid*

*price\_list.price\_id = price.keyid*

*doctor.keyid = docdep.docid*

*docdep.depid = dep.keyid*

*docserv.docdepid = docdep.keyid + docserv.srvdepid = srvdep.keyid*

*discount.agrid = agr.keyid + (discount.servid = srvdep.keyid OR discount.srvgroupid = service\_group.keyid OR discount.depid = dep.keyid (не используется))*

*Код SQL:*

*-- Create table*

create table SERVICE\_GROUP

(

keyid NUMBER(12) not null,

createby NUMBER(12),

createdate DATE default SYSDATE,

updateby NUMBER(12),

updatedate DATE default SYSDATE,

status NUMBER(2) default 1 not null,

code VARCHAR2(24 CHAR),

text VARCHAR2(256 CHAR),

note VARCHAR2(4000 CHAR),

root\_id NUMBER(12),

sortcode NUMBER(12) default 0,

webtext VARCHAR2(512 CHAR),

tree\_code VARCHAR2(30 CHAR)

)

tablespace MED

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

*-- Add comments to the table*

comment on table SERVICE\_GROUP

is 'разделы прейскуранта';

*-- Add comments to the columns*

comment on column SERVICE\_GROUP.createby

is 'Кто создал';

comment on column SERVICE\_GROUP.createdate

is 'Дата создания';

comment on column SERVICE\_GROUP.updateby

is 'Кто последний изменил';

comment on column SERVICE\_GROUP.updatedate

is 'дата последнего изменения';

comment on column SERVICE\_GROUP.status

is 'статус группы:

0 - архивный

1 - активный;';

comment on column SERVICE\_GROUP.code

is 'код раздела';

comment on column SERVICE\_GROUP.text

is 'Название';

comment on column SERVICE\_GROUP.note

is 'замечания';

comment on column SERVICE\_GROUP.root\_id

is 'Ссылка на раздел предыдущего уровня';

comment on column SERVICE\_GROUP.sortcode

is 'Поле для сортировки элементов одного уровня внутри узла дерева';

comment on column SERVICE\_GROUP.webtext

is 'Название для отображения в веб-интерфейсах';

comment on column SERVICE\_GROUP.tree\_code

is 'Код раздела, для быстрого определения отношений вложенности в дереве. по три символа на уровень';

*-- Create/Recreate indexes*

create index I\_SERVICE\_GROUP\_ROOT\_ID on SERVICE\_GROUP (ROOT\_ID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_SERVICE\_GROUP\_TEXT on SERVICE\_GROUP (TEXT)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 768K

next 1M

minextents 1

maxextents unlimited

);

create index I\_SERVICE\_GROUP\_TREE\_CODE on SERVICE\_GROUP (TREE\_CODE)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

*-- Create/Recreate primary, unique and foreign key constraints*

alter table SERVICE\_GROUP

add constraint PK\_SERVICE\_GROUP primary key (KEYID)

using index

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

*-- Create table*

create table SRVDEP

(

keyid NUMBER(12) not null,

createby NUMBER(12),

createdate DATE default SYSDATE,

updateby NUMBER(12),

updatedate DATE default SYSDATE,

servid NUMBER(12),

depid NUMBER(12) not null,

categid NUMBER(12),

status NUMBER(2) default 1 not null,

tag NUMBER(10),

code VARCHAR2(48 CHAR) not null,

text VARCHAR2(512 CHAR),

stext VARCHAR2(64 CHAR),

bgndat DATE default SYSDATE,

enddat DATE,

medstatus NUMBER(2),

minutes NUMBER(6) default 0,

luid NUMBER(12),

specid NUMBER(12),

print\_code VARCHAR2(48 CHAR),

print\_text VARCHAR2(512 CHAR),

srvsalary\_id NUMBER(12),

luid110 NUMBER(12),

luid117 NUMBER(12),

srvsalary\_id1 NUMBER(12),

luid122 NUMBER(12),

service\_group\_id NUMBER(12),

groupid NUMBER(12),

typeid NUMBER(12),

code\_apu VARCHAR2(32 CHAR),

method VARCHAR2(4000 CHAR),

preparation VARCHAR2(4000 CHAR),

duration VARCHAR2(128 CHAR),

nds\_koef NUMBER,

place\_id NUMBER(12),

complex\_status NUMBER(1),

preparation\_id NUMBER(12),

cito\_status NUMBER(1),

oms\_status NUMBER(1),

home\_status NUMBER(1),

preparation\_oms VARCHAR2(4000 CHAR),

preparation\_oms\_id NUMBER(12),

qty\_max NUMBER(12),

doc\_time NUMBER(10,2),

nurse\_time NUMBER(10,2),

srvdep\_oms\_id NUMBER,

srvdep\_bud\_id NUMBER,

srvdep\_slr\_id NUMBER,

rgb VARCHAR2(6 CHAR),

kkm\_code NUMBER(6),

protected\_status NUMBER(1),

children\_oms\_code VARCHAR2(64 CHAR),

webtext VARCHAR2(512 CHAR),

lab\_short\_text VARCHAR2(48 CHAR),

pass\_status NUMBER(1),

anonym\_status NUMBER(1),

print\_list\_id NUMBER,

blank\_id NUMBER,

docdep\_id NUMBER,

check\_text VARCHAR2(512 CHAR),

special\_text VARCHAR2(128 CHAR),

lab\_deadline VARCHAR2(48 CHAR),

age\_from NUMBER(3),

age\_to NUMBER(3),

sex NUMBER(1),

expire NUMBER(3),

description CLOB,

note CLOB

)

tablespace MED

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 704K

next 1M

minextents 1

maxextents unlimited

);

*-- Add comments to the table*

comment on table SRVDEP

is 'Справочник услуг';

*-- Add comments to the columns*

comment on column SRVDEP.keyid

is 'Услуга';

comment on column SRVDEP.createby

is 'Кто создал';

comment on column SRVDEP.createdate

is 'Дата создания';

comment on column SRVDEP.updateby

is 'Кто последний изменил';

comment on column SRVDEP.updatedate

is 'дата последнего изменения';

comment on column SRVDEP.servid

is 'Не используется. Устарело. Услуга. SERVICE.KEYID';

comment on column SRVDEP.depid

is 'отделение';

comment on column SRVDEP.categid

is 'FCATEG.KEYID (TAG = 3- категория коек)

0 для остальных;';

comment on column SRVDEP.status

is 'Статус записи:

0 - архивная

1 - активная; 2 - еще не активная; 3 - уже не активная;';

comment on column SRVDEP.tag

is 'Тип услуги (lu.code из справочника lu.tag = 23)';

comment on column SRVDEP.code

is 'код';

comment on column SRVDEP.text

is 'название';

comment on column SRVDEP.stext

is 'код ОМС';

comment on column SRVDEP.bgndat

is 'Не используется. Дата начала предоставления услуги';

comment on column SRVDEP.enddat

is 'Не используется. Дата окончания предоставления услуги';

comment on column SRVDEP.medstatus

is 'Не используется.';

comment on column SRVDEP.minutes

is 'Не используется.';

comment on column SRVDEP.luid

is 'Не используется.';

comment on column SRVDEP.specid

is 'специальность врача для услуг типа "консультация" (tag типа услуги=1), lu.tag=9';

comment on column SRVDEP.print\_code

is 'код ДЛЯ ПЕЧАТИ';

comment on column SRVDEP.print\_text

is 'название ДЛЯ ПЕЧАТИ';

comment on column SRVDEP.srvsalary\_id

is 'группа начисления ЗП по амб.услугам';

comment on column SRVDEP.luid110

is 'Типы исследований: для бюджетных реестров';

comment on column SRVDEP.luid117

is 'Типы исследований: для реестров ОМС';

comment on column SRVDEP.srvsalary\_id1

is 'группа начисления ЗП по стац.услугам';

comment on column SRVDEP.luid122

is 'Типы исследований: для прочих реестров';

comment on column SRVDEP.service\_group\_id

is 'Раздел прейскуранта';

comment on column SRVDEP.groupid

is 'Группа услуг (lu.tag = 27)';

comment on column SRVDEP.typeid

is 'Простая или сложная услуга (lu.tag = 43)';

comment on column SRVDEP.code\_apu

is 'код услуги по АПУ (внешняя программа учета)';

comment on column SRVDEP.method

is 'Применяемый метод';

comment on column SRVDEP.preparation

is 'Правила подготовки к услуге';

comment on column SRVDEP.duration

is 'Длительность (время оказания) услуги';

comment on column SRVDEP.nds\_koef

is 'НДС (%)';

comment on column SRVDEP.place\_id

is 'Местоположение услуги (основное)';

comment on column SRVDEP.complex\_status

is '0 - обычная; 1 - составная; 2 - комплексная;';

comment on column SRVDEP.preparation\_id

is 'Не используется. LU.TAG=170';

comment on column SRVDEP.cito\_status

is 'Ограничения. услуга разрешена к выполнению по CITO';

comment on column SRVDEP.oms\_status

is 'Ограничения. услуга оказывается только по ОМС';

comment on column SRVDEP.home\_status

is 'Ограничения. услуга оказывается только на дому';

comment on column SRVDEP.preparation\_oms

is 'правила подготовки к бесплатной услуге - текст';

comment on column SRVDEP.preparation\_oms\_id

is 'Правила подготовки к бесплатной услуге - ссылка на справочник. LU.TAG=170';

comment on column SRVDEP.qty\_max

is 'Ограничения. количество максимально оказываемых услуг за один прием пациенту';

comment on column SRVDEP.doc\_time

is 'Длительность выполнения услуги врачом';

comment on column SRVDEP.nurse\_time

is 'Длительность выполнения услуги сестрой';

comment on column SRVDEP.srvdep\_oms\_id

is 'услуга, на которую заменяется услуга пациента, если источник финансирования по ОМС';

comment on column SRVDEP.srvdep\_bud\_id

is 'услуга, на которую заменяется услуга пациента, если источник финансирования по бюджету';

comment on column SRVDEP.srvdep\_slr\_id

is 'услуга, на которую заменяется услуга пациента при расчете зарплаты';

comment on column SRVDEP.rgb

is 'Цвет отображаемой услуги';

comment on column SRVDEP.kkm\_code

is 'Код услуги в ККМ';

comment on column SRVDEP.protected\_status

is '1 - цена услуги не изменяется при групповых изменениях в прейскуранте';

comment on column SRVDEP.children\_oms\_code

is 'Код по ОМС для детей';

comment on column SRVDEP.webtext

is 'Название услуги для отображения в веб-интерфейсах';

comment on column SRVDEP.lab\_short\_text

is 'краткое название услуги для лаборатории';

comment on column SRVDEP.pass\_status

is 'услуга выполняется только по предъявлению паспорта';

comment on column SRVDEP.anonym\_status

is 'статус анонимности услуги (в бланк вставляется информация вместо положительного результата)';

comment on column SRVDEP.print\_list\_id

is 'ссылка на печатную форму для конкретной услуги (индивидуальный бланк)';

comment on column SRVDEP.blank\_id

is 'Тип бланка лабораторного направления. LU.TAG=339';

comment on column SRVDEP.docdep\_id

is 'Врач по умолчанию';

comment on column SRVDEP.check\_text

is 'название услуги для отображения в чеке';

comment on column SRVDEP.special\_text

is 'Специальное (жаргонное, сокращенное) название услуги';

comment on column SRVDEP.lab\_deadline

is 'Лаборатория. Срок выполнения.';

comment on column SRVDEP.age\_from

is 'Ограничения. Минимальный возраст, в годах';

comment on column SRVDEP.age\_to

is 'Ограничения. Максимальный возраст, в годах';

comment on column SRVDEP.sex

is 'Ограничения. Пол: NULL - для всех (значение по умолчанию); 0 - мужской; 1 - женский;';

comment on column SRVDEP.expire

is 'Ограничения. Период действия результата услуги, в днях. Предполагается, что в этот период можно не делать услугу повторно.';

*-- Create/Recreate indexes*

create index I\_SRVDEP\_CATEGID on SRVDEP (CATEGID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 128K

next 1M

minextents 1

maxextents unlimited

);

create index I\_SRVDEP\_CODE on SRVDEP (CODE)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_SRVDEP\_DEPID on SRVDEP (DEPID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 192K

next 1M

minextents 1

maxextents unlimited

);

create index I\_SRVDEP\_DOCDEP\_ID on SRVDEP (DOCDEP\_ID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_SRVDEP\_GROUPID on SRVDEP (GROUPID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_SRVDEP\_PLACE\_ID on SRVDEP (PLACE\_ID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_SRVDEP\_SERVICE\_GROUP\_ID on SRVDEP (SERVICE\_GROUP\_ID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_SRVDEP\_SERVID on SRVDEP (SERVID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 192K

next 1M

minextents 1

maxextents unlimited

);

create index I\_SRVDEP\_SRVSALARY1\_ID on SRVDEP (SRVSALARY\_ID1)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_SRVDEP\_SRVSALARY\_ID on SRVDEP (SRVSALARY\_ID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_SRVDEP\_STATUS on SRVDEP (STATUS)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

*-- Create/Recreate primary, unique and foreign key constraints*

alter table SRVDEP

add constraint PK\_SRVDEP primary key (KEYID)

using index

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 128K

next 1M

minextents 1

maxextents unlimited

);

alter table SRVDEP

add constraint FK\_SERVICE\_HAS\_SRVDEPS foreign key (SERVID)

references SERVICE (KEYID);

alter table SRVDEP

add constraint FK\_SRVDEP\_DEP foreign key (DEPID)

references DEP (KEYID);

alter table SRVDEP

add constraint FK\_SRVDEP\_DEP\_PLACE foreign key (PLACE\_ID)

references DEP (KEYID) on delete set null;

alter table SRVDEP

add constraint FK\_SRVDEP\_DOCDEP foreign key (DOCDEP\_ID)

references DOCDEP (KEYID) on delete set null;

alter table SRVDEP

add constraint FK\_SRVDEP\_SERVICE\_GROUP foreign key (SERVICE\_GROUP\_ID)

references SERVICE\_GROUP (KEYID);

alter table SRVDEP

add constraint FK\_SRVDEP\_SRVSALARY\_ID foreign key (SRVSALARY\_ID)

references SRVSALARY (KEYID);

alter table SRVDEP

add constraint FK\_SRVDEP\_SRVSALARY\_ID1 foreign key (SRVSALARY\_ID1)

references SRVSALARY (KEYID);

*-- Create table*

create table PRICE

(

keyid NUMBER(12) not null,

createby NUMBER(12),

createdate DATE default SYSDATE,

updateby NUMBER(12),

updatedate DATE default SYSDATE,

status NUMBER(2) default 1 not null,

tag NUMBER(2) default 0 not null,

srvdepid NUMBER(12) not null,

bgndat DATE default SYSDATE not null,

enddat DATE,

price NUMBER(12,2),

nprice NUMBER(12,2) default 0,

cost NUMBER(12,2) default 0,

note VARCHAR2(128 CHAR),

price\_typ\_id NUMBER(12),

categid NUMBER(12),

vprice NUMBER(12,2) default 0,

omsprice NUMBER(12,2) default 0,

fprice NUMBER(12,2) default 0,

agrid NUMBER(12),

cprice NUMBER(12,2),

nds\_percent NUMBER(3)

)

tablespace MED

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

*-- Add comments to the table*

comment on table PRICE

is 'Интервал прейскуранта';

*-- Add comments to the columns*

comment on column PRICE.createby

is 'Кто создал';

comment on column PRICE.createdate

is 'Дата создания';

comment on column PRICE.updateby

is 'Кто последний изменил';

comment on column PRICE.updatedate

is 'дата последнего изменения';

comment on column PRICE.status

is 'статус записи.

0 - архивный

1 - активный;';

comment on column PRICE.tag

is 'вид (тип) прейскуранта

0 - услуги

1 - категории коек

2 - профили (коек для стационара)

3 - категории врачей;';

comment on column PRICE.srvdepid

is 'услуга';

comment on column PRICE.bgndat

is 'дата начала действия прейскуранта';

comment on column PRICE.enddat

is 'дата окончания действия прейскуранта';

comment on column PRICE.price

is 'устарело, перенесено в price\_list';

comment on column PRICE.nprice

is 'устарело, перенесено в price\_list';

comment on column PRICE.cost

is 'устарело, перенесено в price\_list';

comment on column PRICE.note

is 'замечания';

comment on column PRICE.price\_typ\_id

is 'устарело, перенесено в PRICE\_LIST (вид цены. LU.TAG=149)';

comment on column PRICE.categid

is 'не используется';

comment on column PRICE.vprice

is 'устарело, перенесено в price\_list';

comment on column PRICE.omsprice

is 'устарело, перенесено в price\_list';

comment on column PRICE.fprice

is 'устарело, перенесено в price\_list';

comment on column PRICE.agrid

is 'Прейскурант для договора';

comment on column PRICE.nds\_percent

is 'НДС (%). Перенесено из SRVDEP.NDS\_KOEF';

*-- Create/Recreate indexes*

create index I\_PRICE\_AGRID on PRICE (AGRID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_PRICE\_BGNDAT on PRICE (BGNDAT)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 704K

next 1M

minextents 1

maxextents unlimited

);

create index I\_PRICE\_ENDDAT on PRICE (ENDDAT)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_PRICE\_SRVDEPID on PRICE (SRVDEPID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 384K

next 1M

minextents 1

maxextents unlimited

);

*-- Create/Recreate primary, unique and foreign key constraints*

alter table PRICE

add constraint PK\_PRICE primary key (KEYID)

using index

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 384K

next 1M

minextents 1

maxextents unlimited

);

alter table PRICE

add constraint FK\_PRICE\_AGR foreign key (AGRID)

references AGR (KEYID)

disable

novalidate;

alter table PRICE

add constraint FK\_PRICE\_SRVDEP foreign key (SRVDEPID)

references SRVDEP (KEYID);

*-- Create table*

create table PRICE\_LIST

(

keyid NUMBER(12) not null,

createby NUMBER(12),

createdate DATE default SYSDATE,

updateby NUMBER(12),

updatedate DATE default SYSDATE,

price\_id NUMBER(12) not null,

lu\_id NUMBER(12) not null,

status NUMBER(2) default 1 not null,

tag NUMBER(2) default 0 not null,

price NUMBER(12,2),

nprice NUMBER(12,2),

cost NUMBER(12,2),

note VARCHAR2(128 CHAR)

)

tablespace MED

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 1M

next 1M

minextents 1

maxextents unlimited

);

*-- Add comments to the table*

comment on table PRICE\_LIST

is 'Цены услуги';

*-- Add comments to the columns*

comment on column PRICE\_LIST.price\_id

is 'ссылка на период цен услуги. price.keyid';

comment on column PRICE\_LIST.lu\_id

is 'тип цены. lu.tag = 149';

comment on column PRICE\_LIST.status

is 'не используется';

comment on column PRICE\_LIST.tag

is 'не используется';

comment on column PRICE\_LIST.price

is 'безнал.';

comment on column PRICE\_LIST.nprice

is 'нал.';

comment on column PRICE\_LIST.cost

is 'себест.';

comment on column PRICE\_LIST.note

is 'не используется';

*-- Create/Recreate indexes*

create index I\_PRICE\_LIST\_LU\_ID on PRICE\_LIST (LU\_ID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 3M

next 1M

minextents 1

maxextents unlimited

);

create index I\_PRICE\_LIST\_PRICE\_ID on PRICE\_LIST (PRICE\_ID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 3M

next 1M

minextents 1

maxextents unlimited

);

*-- Create/Recreate primary, unique and foreign key constraints*

alter table PRICE\_LIST

add constraint PK\_PRICE\_LIST primary key (KEYID)

using index

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 2M

next 1M

minextents 1

maxextents unlimited

);

alter table PRICE\_LIST

add constraint UK\_PRICE\_LIST unique (PRICE\_ID, LU\_ID)

using index

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

alter table PRICE\_LIST

add constraint FK\_PRICE\_LIST\_LU foreign key (LU\_ID)

references LU (KEYID);

alter table PRICE\_LIST

add constraint FK\_PRICE\_LIST\_PRICE foreign key (PRICE\_ID)

references PRICE (KEYID) on delete cascade;

*-- Create table*

create table DOCTOR

(

keyid NUMBER(12) not null,

createby NUMBER(12),

createdate DATE default SYSDATE,

updateby NUMBER(12),

updatedate DATE default SYSDATE,

ucategid NUMBER(12),

specid NUMBER(12),

depid NUMBER(12),

positionid NUMBER(12),

staffid NUMBER(12) not null,

status NUMBER(2) default 1 not null,

code VARCHAR2(40 CHAR),

text VARCHAR2(256 CHAR),

num NUMBER(5),

bgndat DATE default SYSDATE,

enddat DATE,

note VARCHAR2(4000 CHAR),

spec\_d NUMBER(12),

sertificate VARCHAR2(256 CHAR),

degree VARCHAR2(128 CHAR),

birthdate DATE,

tabnum NUMBER(12),

lastname VARCHAR2(64 CHAR),

firstname VARCHAR2(64 CHAR),

secondname VARCHAR2(64 CHAR),

phone VARCHAR2(128 CHAR),

man\_id NUMBER(12),

workphone VARCHAR2(80 CHAR),

cellular VARCHAR2(80 CHAR),

address VARCHAR2(320 CHAR),

email VARCHAR2(128 CHAR),

snils VARCHAR2(14 CHAR),

sertificate\_enddat DATE,

sex NUMBER(1),

inn VARCHAR2(12 CHAR),

patient\_id NUMBER,

frmr\_id NUMBER

)

tablespace MED

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 2M

next 1M

minextents 1

maxextents unlimited

);

*-- Add comments to the table*

comment on table DOCTOR

is 'Сотрудник';

*-- Add comments to the columns*

comment on column DOCTOR.createby

is 'Кто создал';

comment on column DOCTOR.createdate

is 'Дата создания';

comment on column DOCTOR.updateby

is 'Кто последний изменил';

comment on column DOCTOR.updatedate

is 'дата последнего изменения';

comment on column DOCTOR.ucategid

is 'Категория сотрудника на основном отделении (LU.TAG=26)';

comment on column DOCTOR.specid

is 'Специальность сотрудника на основном отделении (LU.TAG=9)';

comment on column DOCTOR.depid

is 'Отделение, на котором числится сотрудник. Если сотрудник работает на нескольких отделениях, то это отделение его основного места работы.';

comment on column DOCTOR.positionid

is 'должность (Lu.tag = 22)';

comment on column DOCTOR.staffid

is 'Тип персонала (LU.TAG=21)

Врач/Средний мед. персонал/Младший мед. персонал/...';

comment on column DOCTOR.status

is 'Статус сотрудника:

0 - архивный

1 - активный;';

comment on column DOCTOR.code

is 'Код сотрудника';

comment on column DOCTOR.text

is 'ФИО. По умолчанию формируется из полей Фамилия Имя Отчество';

comment on column DOCTOR.num

is '(не используется)';

comment on column DOCTOR.bgndat

is 'дата начала работы в учреждении';

comment on column DOCTOR.enddat

is 'дата окончания работы в учреждении';

comment on column DOCTOR.note

is 'замечания';

comment on column DOCTOR.spec\_d

is '(не используется)';

comment on column DOCTOR.sertificate

is '(не используется) сертификат';

comment on column DOCTOR.degree

is 'Степень';

comment on column DOCTOR.birthdate

is 'дата рождения';

comment on column DOCTOR.tabnum

is 'табельный номер';

comment on column DOCTOR.lastname

is 'фамилия';

comment on column DOCTOR.firstname

is 'имя';

comment on column DOCTOR.secondname

is 'отчество';

comment on column DOCTOR.phone

is 'Контакты. Контактный телефон';

comment on column DOCTOR.man\_id

is 'ссылка на пользователя системы (привязка персонала к пользователям системы)';

comment on column DOCTOR.workphone

is 'Контакты. Рабочий телефон';

comment on column DOCTOR.cellular

is 'Контакты. Мобильный телефон';

comment on column DOCTOR.address

is 'Контакты. Адрес';

comment on column DOCTOR.email

is 'Контакты. Адрес электронной почты';

comment on column DOCTOR.snils

is 'СНИЛС. Страховой номер индивидуального лицевого счёта';

comment on column DOCTOR.sertificate\_enddat

is '(не используется) Дата окончания действия сертификата';

comment on column DOCTOR.sex

is 'Пол (мужской/женский)

0 - мужской; 1 - женский; NULL - без указания;';

comment on column DOCTOR.inn

is 'ИНН. Индивидуальный номер налогоплательщика. Используетя для кассиров.';

comment on column DOCTOR.patient\_id

is 'Пациент. Если сотрудник обслуживается в ЛПУ';

comment on column DOCTOR.frmr\_id

is 'Федеральный реестр медицинских работников. Основная карта. INETUSER.FRMR\_GENERAL.ID';

*-- Create/Recreate indexes*

create index I\_DOCTOR\_DEPID on DOCTOR (DEPID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DOCTOR\_FRMR\_ID on DOCTOR (FRMR\_ID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DOCTOR\_MANID on DOCTOR (MAN\_ID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DOCTOR\_PATIENT\_ID on DOCTOR (PATIENT\_ID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DOCTOR\_POSITIONID on DOCTOR (POSITIONID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DOCTOR\_SPECID on DOCTOR (SPECID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DOCTOR\_STAFFID on DOCTOR (STAFFID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DOCTOR\_TEXT on DOCTOR (TEXT)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 448K

next 1M

minextents 1

maxextents unlimited

);

*-- Create/Recreate primary, unique and foreign key constraints*

alter table DOCTOR

add constraint PK\_DOCTOR primary key (KEYID)

using index

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 192K

next 1M

minextents 1

maxextents unlimited

);

alter table DOCTOR

add constraint FK\_DOCTOR\_DEP foreign key (DEPID)

references DEP (KEYID);

alter table DOCTOR

add constraint FK\_DOCTOR\_MAN foreign key (MAN\_ID)

references MAN (KEYID);

alter table DOCTOR

add constraint FK\_DOCTOR\_PATIENT foreign key (PATIENT\_ID)

references PATIENT (KEYID);

*-- Create table*

create table DOCDEP

(

keyid NUMBER(12) not null,

createby NUMBER(12),

createdate DATE default SYSDATE,

updateby NUMBER(12),

updatedate DATE default SYSDATE,

ucategid NUMBER(12),

specid NUMBER(12),

docid NUMBER(12) not null,

depid NUMBER(12) not null,

positionid NUMBER(12),

specialid NUMBER(12),

status NUMBER(2) default 1 not null,

code VARCHAR2(12 CHAR),

text VARCHAR2(256 CHAR),

num NUMBER(5),

bgndat DATE default SYSDATE,

enddat DATE,

note VARCHAR2(4000 CHAR),

examid NUMBER(12),

fict\_status NUMBER(2) default 0,

salary\_typ\_id NUMBER(12),

staff\_typ\_id NUMBER(12),

areanumid NUMBER(12),

add\_numb\_qty NUMBER(12),

diagnost\_status NUMBER(2),

doc\_type\_id NUMBER(12),

staff\_docdep\_id NUMBER(12),

room\_id NUMBER,

prev\_status NUMBER(1),

pat\_sex NUMBER(1),

pat\_age\_from NUMBER(3),

pat\_age\_to NUMBER(3)

)

tablespace MED

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 896K

next 1M

minextents 1

maxextents unlimited

);

*-- Add comments to the table*

comment on table DOCDEP

is 'Сотрудник на отделении';

*-- Add comments to the columns*

comment on column DOCDEP.keyid

is 'сотрудник на отделении';

comment on column DOCDEP.createby

is 'Кто создал';

comment on column DOCDEP.createdate

is 'Дата создания';

comment on column DOCDEP.updateby

is 'Кто последний изменил';

comment on column DOCDEP.updatedate

is 'дата последнего изменения';

comment on column DOCDEP.ucategid

is 'категория (LU.TAG=26)';

comment on column DOCDEP.specid

is 'специальность (LU.TAG=9)';

comment on column DOCDEP.docid

is 'Сотрудник';

comment on column DOCDEP.depid

is 'отделение';

comment on column DOCDEP.positionid

is 'Должность';

comment on column DOCDEP.specialid

is 'Специализация';

comment on column DOCDEP.status

is 'Статус записи:

0 - архивный

1 - активный;';

comment on column DOCDEP.code

is 'Код сотрудника на отделении';

comment on column DOCDEP.text

is 'Отделение, ФИО сотрудника. По умолчанию дублирует ФИО с doctor, но может быть вручную изменено.';

comment on column DOCDEP.num

is 'Номер кабинета (при возможности синхронен с ROOM\_ID)';

comment on column DOCDEP.bgndat

is 'дата начала работы на отделении';

comment on column DOCDEP.enddat

is 'дата окончания работы на отделении';

comment on column DOCDEP.note

is 'замечания';

comment on column DOCDEP.examid

is '(не используется) Исследование (LU.TAG=28)';

comment on column DOCDEP.fict\_status

is 'Статус фиктивности: 0 - реальный сотрудник; 1 - фиктивный;';

comment on column DOCDEP.salary\_typ\_id

is 'Тип начисления зарплаты для сотрудника (LU.TAG=303)';

comment on column DOCDEP.staff\_typ\_id

is 'тип персонала (LU.Tag = 311)';

comment on column DOCDEP.areanumid

is 'номер участка (LU.tag = 39)';

comment on column DOCDEP.add\_numb\_qty

is 'Кол-во доп. номерков, разрешенных для выдачи другими врачами';

comment on column DOCDEP.doc\_type\_id

is 'тип врача(LU.Tag = 386)';

comment on column DOCDEP.staff\_docdep\_id

is 'тип персонала (LU.Tag = 21)';

comment on column DOCDEP.room\_id

is 'Идентификатор кабинета (LU.TAG=376)';

comment on column DOCDEP.prev\_status

is 'Предыдущее значение статуса. Необходимо для восстановления правильного статуса у записи, при изменении статуса у сотрудника.';

comment on column DOCDEP.pat\_sex

is 'Ограничения на обслуживание пациентов. Пол: NULL - для всех (значение по умолчанию); 0 - мужской; 1 - женский;';

comment on column DOCDEP.pat\_age\_from

is 'Ограничения на обслуживание пациентов. Минимальный возраст, полных лет.';

comment on column DOCDEP.pat\_age\_to

is 'Ограничения на обслуживание пациентов. Максимальный возраст, полных лет.';

*-- Create/Recreate indexes*

create index I\_DOCDEP\_DEPID on DOCDEP (DEPID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 128K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DOCDEP\_DOCID on DOCDEP (DOCID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 128K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DOCDEP\_DOC\_TYPE\_ID on DOCDEP (DOC\_TYPE\_ID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DOCDEP\_POSITIONID on DOCDEP (POSITIONID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DOCDEP\_ROOM\_ID on DOCDEP (ROOM\_ID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DOCDEP\_SPECIALID on DOCDEP (SPECIALID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DOCDEP\_SPECID on DOCDEP (SPECID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 128K

next 1M

minextents 1

maxextents unlimited

);

*-- Create/Recreate primary, unique and foreign key constraints*

alter table DOCDEP

add constraint PK\_DOCDEP primary key (KEYID)

using index

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 192K

next 1M

minextents 1

maxextents unlimited

);

alter table DOCDEP

add constraint FK\_DOCDEP\_DEP foreign key (DEPID)

references DEP (KEYID);

alter table DOCDEP

add constraint FK\_DOCDEP\_DOCTOR foreign key (DOCID)

references DOCTOR (KEYID);

alter table DOCDEP

add constraint FK\_DOCDEP\_LU\_SPEC foreign key (SPECID)

references LU (KEYID);

*-- Create table*

create table DEP

(

keyid NUMBER(12) not null,

createby NUMBER(12),

createdate DATE default SYSDATE,

updateby NUMBER(12),

updatedate DATE default SYSDATE,

typeid NUMBER(6) not null,

rootid NUMBER(12) not null,

status NUMBER(2) default 1 not null,

typ NUMBER(2),

lev NUMBER(5) default 1,

sortcode VARCHAR2(24 CHAR) not null,

code VARCHAR2(24 CHAR),

text VARCHAR2(256 CHAR),

stext VARCHAR2(64 CHAR),

bed NUMBER(5),

num NUMBER(5),

note VARCHAR2(128 CHAR),

out\_status NUMBER(2),

in\_status NUMBER(2),

place\_status NUMBER(2),

p\_code VARCHAR2(16 CHAR),

luid NUMBER(12),

internal\_code VARCHAR2(8 CHAR),

status\_dep NUMBER(12),

room VARCHAR2(12 CHAR),

oper\_status NUMBER(1),

profpat\_lu\_id NUMBER(12),

oms\_prof\_code NUMBER(3),

is\_consulting NUMBER(1),

chamber\_group\_status NUMBER,

lpu\_id NUMBER,

can\_prof\_status NUMBER(1) default 1,

sort VARCHAR2(6 CHAR),

abbr\_text VARCHAR2(64 CHAR),

frmo\_dep\_id NUMBER

)

tablespace MED

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

*-- Add comments to the table*

comment on table DEP

is 'Справочник подразделений';

*-- Add comments to the columns*

comment on column DEP.createby

is 'Кто создал';

comment on column DEP.createdate

is 'Дата создания';

comment on column DEP.updateby

is 'Кто последний изменил ';

comment on column DEP.updatedate

is 'дата последнего изменения';

comment on column DEP.typeid

is 'не используется. Тип отделения:

LU.TAG=10(CODE).

100 - все мед.учреждение

101 - амбулаторные

102 - стационарные

103 - диагностические

104 - остальные

105 - специальные;';

comment on column DEP.rootid

is 'Ссылка на вышестоящее отделение. DEP.KEYID';

comment on column DEP.status

is 'статус отделения:

0 - архивный

1 - активный;';

comment on column DEP.typ

is 'Отделение с койко-фондом:

0 - без койко-фонда

1 - с койко-фондом;';

comment on column DEP.lev

is 'Уровень вложенности:

1 - все мед. учреждения (структуры)

2 - тип отделения

3 -.. прочие отделения;';

comment on column DEP.sortcode

is 'Уровень, сортировка на каждом уровне

Для каждого уровня выделено 3 символа для порядкового номера.

Значение ''''''''002005'''''''' означает что отделение находится на втором уровне 5-м по порядку, а вышестоящий уровень 2-й по порядку.

Классификатор отделений допускает 8 уровней.';

comment on column DEP.code

is 'код отделения';

comment on column DEP.text

is 'Название отделения';

comment on column DEP.stext

is 'Короткое название';

comment on column DEP.bed

is 'количество коек - для стационарных отделений';

comment on column DEP.num

is 'номер отделения';

comment on column DEP.note

is 'замечания';

comment on column DEP.out\_status

is 'отделение выбытия. 1-ДА';

comment on column DEP.in\_status

is 'отделение поступления. 1-ДА';

comment on column DEP.place\_status

is 'отделение - местонахождения(местоположение). 1-ДА';

comment on column DEP.p\_code

is 'Код местоположения';

comment on column DEP.luid

is 'Группа для городской сводки. LU.TAG=48. Для объединения в отчетные группы для города';

comment on column DEP.internal\_code

is 'внутренний код';

comment on column DEP.status\_dep

is 'Тип отделения. LU.TAG=202, хранится CODE';

comment on column DEP.room

is 'Номер кабинета для отделений типа КАБИНЕТ (DEP.STATUS\_DEP=400)';

comment on column DEP.oper\_status

is 'Оперирующее отделение. 1-ДА';

comment on column DEP.profpat\_lu\_id

is 'профиль патологии. LU.TAG=283';

comment on column DEP.oms\_prof\_code

is 'Профиль лечения на отделении по справочнику ОМС';

comment on column DEP.is\_consulting

is 'Консультирующее отделение. 1-ДА';

comment on column DEP.chamber\_group\_status

is 'Группа, к которой относится отделение, имеет общие коечный фонд (палаты и койки)';

comment on column DEP.lpu\_id

is 'ЛПУ, к которому относится отделение (LPU.keyid)';

comment on column DEP.can\_prof\_status

is 'Может ли быть профильным. 0 - не может; 1 - может;';

comment on column DEP.sort

is 'Порядковый номер. Поле для сортировки при отображении отделений в одной ветке';

comment on column DEP.abbr\_text

is 'Предельно сокращенное название отделения';

comment on column DEP.frmo\_dep\_id

is 'Федеральный реестр медицинских организаций. Подразделение. INETUSER.FRMO\_DEP.ID';

*-- Create/Recreate indexes*

create index I\_DEP\_FRMO\_DEP\_ID on DEP (FRMO\_DEP\_ID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DEP\_LPU\_ID on DEP (LPU\_ID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DEP\_ROOTID on DEP (ROOTID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DEP\_STATUS\_DEP on DEP (STATUS\_DEP)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DEP\_STRUCT on DEP (SUBSTR(SORTCODE,1,3))

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DEP\_TYPEID on DEP (TYPEID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

*-- Create/Recreate primary, unique and foreign key constraints*

alter table DEP

add constraint PK\_DEP primary key (KEYID)

using index

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

alter table DEP

add constraint UK\_DEP\_SORTCODE unique (SORTCODE)

using index

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

alter table DEP

add constraint FK\_DEP\_LPU foreign key (LPU\_ID)

references LPU (KEYID);

*-- Create table*

create table DOCSERV

(

keyid NUMBER(12) not null,

createby NUMBER(12),

createdate DATE default SYSDATE,

updateby NUMBER(12),

updatedate DATE default SYSDATE,

docdepid NUMBER(12) not null,

srvdepid NUMBER(12) not null,

status NUMBER(2) default 1 not null,

code VARCHAR2(12 CHAR) default 0 not null,

note VARCHAR2(128 CHAR),

linkid NUMBER(12) default 0,

koef\_a NUMBER(12,2),

koef\_s NUMBER(12,2),

amount\_s NUMBER(12,2),

wage\_status NUMBER(1),

exec\_time NUMBER(10,2)

)

tablespace MED

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 2M

next 1M

minextents 1

maxextents unlimited

);

*-- Add comments to the table*

comment on table DOCSERV

is 'Услуги врача';

*-- Add comments to the columns*

comment on column DOCSERV.keyid

is 'Услуги врача';

comment on column DOCSERV.createby

is 'Кто создал';

comment on column DOCSERV.createdate

is 'Дата создания';

comment on column DOCSERV.updateby

is 'Кто последний изменил';

comment on column DOCSERV.updatedate

is 'Дата последнего изменения';

comment on column DOCSERV.docdepid

is 'врач';

comment on column DOCSERV.srvdepid

is 'услуга';

comment on column DOCSERV.status

is 'Статус:

0 - архивный

1 - активный;';

comment on column DOCSERV.code

is '0 - БЕЗ ПРИВЯЗКИ (Обычная услуга врача)

1 - привязка К ПЕРИОДУ (RPERIOD)

2 - привязка к НОМЕРКУ (RNUMB)

Запись с кодом 2 создается ТОЛЬКО когда количество оказываемых услуг на номерок разнится со стандартным!

3 - приявязка к RDAY

4 - FDAY

5 - FNUMB;';

comment on column DOCSERV.note

is 'замечания';

comment on column DOCSERV.linkid

is 'Поле используется в расписании - для привязки еще и к периоду, а так же к номерку. 0 - в эксперте';

comment on column DOCSERV.koef\_a

is 'процент отчисления ЗП на амбулаторную услугу';

comment on column DOCSERV.koef\_s

is 'процент отчисления ЗП на стационарную услугу';

comment on column DOCSERV.amount\_s

is 'Фиксированная сумма отчислений';

comment on column DOCSERV.wage\_status

is 'Статус оплаты по окладу (1 - услуга входит в оклад, null, 0 - оплачивается отдельно)';

comment on column DOCSERV.exec\_time

is 'Время выполнения услуги сотрудником (мин)';

*-- Create/Recreate indexes*

create index I\_DOCSERV\_CODE on DOCSERV (CODE)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 576K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DOCSERV\_DOCDEPID on DOCSERV (DOCDEPID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 320K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DOCSERV\_LINKID on DOCSERV (LINKID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 256K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DOCSERV\_SRVDEPID on DOCSERV (SRVDEPID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 320K

next 1M

minextents 1

maxextents unlimited

);

*-- Create/Recreate primary, unique and foreign key constraints*

alter table DOCSERV

add constraint PK\_DOCSERV primary key (KEYID)

using index

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 640K

next 1M

minextents 1

maxextents unlimited

);

alter table DOCSERV

add constraint FK\_DOCSERV\_DOCDEP foreign key (DOCDEPID)

references DOCDEP (KEYID)

novalidate;

alter table DOCSERV

add constraint FK\_DOCSERV\_SRVDEP foreign key (SRVDEPID)

references SRVDEP (KEYID)

novalidate;

*-- Create table*

create table AGR

(

keyid NUMBER not null,

createby NUMBER(12),

createdate DATE default SYSDATE,

updateby NUMBER(12),

updatedate DATE default SYSDATE,

companyid NUMBER(12) not null,

atypeid NUMBER(12) not null,

bgndat DATE default SYSDATE not null,

enddat DATE,

code VARCHAR2(50 CHAR),

text VARCHAR2(512 CHAR),

terms NUMBER(5),

balance NUMBER(12,2),

typ VARCHAR2(20 CHAR) not null,

status NUMBER(2) default 1 not null,

ambserv NUMBER(5) default 0 not null,

bedserv NUMBER(5) default 0 not null,

diagserv NUMBER(5) default 0 not null,

spserv NUMBER(5) default 0 not null,

atype NUMBER(5) not null,

agrindex VARCHAR2(20 CHAR),

prnform NUMBER(5),

finance NUMBER(5) default 0 not null,

paycode NUMBER(5) default 0 not null,

omscode NUMBER(5) default 1 not null,

stupdatedate DATE default SYSDATE,

stupdateby NUMBER(12),

note VARCHAR2(1280 CHAR),

agrdat DATE,

pstatus NUMBER(2) default 1 not null,

func\_id NUMBER(12),

price\_date\_status NUMBER(1),

price\_police\_status NUMBER(1),

limit\_police\_status NUMBER(1),

srv\_attr\_id NUMBER(12),

srv\_attr\_status NUMBER(2),

dir\_form\_id NUMBER(12),

lpu\_id NUMBER(12),

dop\_status NUMBER,

payment\_status NUMBER(1),

program\_limit\_status NUMBER(1),

include\_family\_status NUMBER(1),

police\_auto\_num NUMBER(1),

pay\_complex\_content NUMBER(1),

no\_resident NUMBER(1),

atype\_add\_id NUMBER(12),

patientid NUMBER(12),

root\_id NUMBER(12),

can\_add\_subagr NUMBER(1),

pat\_attach\_type NUMBER(1) default 0 not null,

policeid NUMBER,

duration NUMBER(5),

duration\_unit\_id NUMBER

)

tablespace MED

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 192K

next 1M

minextents 1

maxextents unlimited

);

*-- Add comments to the table*

comment on table AGR

is 'Договор (шифр)';

*-- Add comments to the columns*

comment on column AGR.keyid

is 'Договор (шифр)';

comment on column AGR.createby

is 'Кто создал';

comment on column AGR.createdate

is 'Дата создания';

comment on column AGR.updateby

is 'Кто последний изменил';

comment on column AGR.updatedate

is 'дата последнего изменения';

comment on column AGR.companyid

is 'id компании (фирмы)

плательщик, которому принадлежит шифр.

COMPANY.Code - это первая и вторая часть шифра';

comment on column AGR.atypeid

is 'тип шифра (LU.Tag = 8)

LU.LCode по этой ссылке - третья часть шифра';

comment on column AGR.bgndat

is 'дата начала действия шифра';

comment on column AGR.enddat

is 'дата окончания дествия шифра';

comment on column AGR.code

is 'код шифра';

comment on column AGR.text

is 'название шифра';

comment on column AGR.terms

is 'не используется';

comment on column AGR.balance

is 'не используется';

comment on column AGR.typ

is 'Шифр.

Имеет маску XX.YYY.XX, где XX - код ранга компании, YYY - номер компании, ZZ - тип шифра';

comment on column AGR.status

is 'Статус шифра

0 - не обслуживаемый (архивный)

1 - обслуживаемый (активный)';

comment on column AGR.ambserv

is 'Договор на оказание амбулаторных услуг.

0 - нет

1 - есть;';

comment on column AGR.bedserv

is 'Договор на оказание стационарных услуг.

0 - нет

1 - есть;';

comment on column AGR.diagserv

is 'Договор на оказание диагностических услуг.

0 - нет

1 - есть;';

comment on column AGR.spserv

is 'Договор на оказание специальных услуг. 0 - нет

1 - есть;';

comment on column AGR.atype

is 'LU.LCode из типа шифра';

comment on column AGR.agrindex

is 'Индекс договора';

comment on column AGR.prnform

is 'не используется';

comment on column AGR.finance

is 'Источник финансирования (LU.CODE TAG=5). 0 - Не определен

1 - Наличные

2 - Договоры

3 - ДМС

4 - Бюджет

5 - ОМС

...';

comment on column AGR.paycode

is 'способ оплаты:

0 - Способ оплаты не определен

1 - Безналичная оплата в рублях

2 - Наличная оплата в рублях

3 - Безналичная оплата в валюте

4 - Наличная оплата в валюте

5 - Оплата по тарифу ОМС;';

comment on column AGR.omscode

is 'ОМС код:

1 - без вычета тарифа ОМС

2 - с вычетом тарифа ОМС';

comment on column AGR.stupdatedate

is 'дата изменения статуса';

comment on column AGR.stupdateby

is 'кем изменен статус';

comment on column AGR.note

is 'замечания';

comment on column AGR.agrdat

is 'дата заключения договора';

comment on column AGR.pstatus

is 'Полис статус

0 - полис не требуется

1 - право на обслуживание подтверждается полисом';

comment on column AGR.func\_id

is 'Тип обработки договора. LU.TAG=161';

comment on column AGR.price\_date\_status

is 'Цена прейскуранта по дате прикрепления пациента';

comment on column AGR.price\_police\_status

is 'Прейскурант на дату начала полиса';

comment on column AGR.limit\_police\_status

is 'Использовать остаточный лимит (с авансированием для нал. шифров)';

comment on column AGR.srv\_attr\_id

is 'ID атрибута из ATTR (группа из набора для контроля услуг по договору)';

comment on column AGR.srv\_attr\_status

is 'вести набор услуг для контроля по договору (аттрибуты услуг - 4, Группировка услуг для контроля по договору - 60 )';

comment on column AGR.dir\_form\_id

is 'вид направления';

comment on column AGR.lpu\_id

is 'ссылка на юридическое лицо (если в мед.учреждении 2 и больше юрлиц)';

comment on column AGR.dop\_status

is '1-шифр не является самостоятельным и не может быть назначен на услугу или визит';

comment on column AGR.payment\_status

is 'Проводить платежи по договору. 0 (null) - нет; 1 - да;';

comment on column AGR.program\_limit\_status

is 'Использовать лимит обслуживания на пациента (или семью) по безнал. оплате';

comment on column AGR.include\_family\_status

is 'Использовать лимит по статусу PROGRAM\_LIMIT\_STATUS на семью';

comment on column AGR.police\_auto\_num

is 'Автоматическая нумерация полиса (1 - да)';

comment on column AGR.pay\_complex\_content

is 'Оплата услуг, входящих в состав комплекса';

comment on column AGR.no\_resident

is 'Признак шифра для иногородних пациентов';

comment on column AGR.atype\_add\_id

is 'Дополнительный тип договора LU.TAG=408';

comment on column AGR.patientid

is 'Идентификатор шифра-пациента. Используется для бронирования на компанию, а не на пациента. У шифра может только один связанный с ним "специальный" пациент.';

comment on column AGR.root\_id

is 'ссылка на основной договор';

comment on column AGR.can\_add\_subagr

is 'Разрешать создание гарантийных писем: 1-да, 0-нет';

comment on column AGR.pat\_attach\_type

is 'Вид прикрепления пациентов:

0 - Без ограничения

1 - Обслуживание по списку

2 - Семейный договор

3 - Персональный договор;';

comment on column AGR.policeid

is 'договор созданный по полису из врачебного АРМ';

comment on column AGR.duration

is 'Базовый срок длительности прикрепления. Единицы измерения указаны в DURATION\_UNIT\_ID';

comment on column AGR.duration\_unit\_id

is 'Единицы измерения срока действия прикрепления. LU.TAG=373';

*-- Create/Recreate indexes*

create index I\_AGR\_ATYPEID on AGR (ATYPEID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_AGR\_COMPANYID on AGR (COMPANYID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_AGR\_FINANCE on AGR (FINANCE)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_AGR\_PATIENTID on AGR (PATIENTID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_AGR\_POLICEID on AGR (POLICEID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_AGR\_ROOTID on AGR (ROOT\_ID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_AGR\_TYP on AGR (TYP)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

*-- Create/Recreate primary, unique and foreign key constraints*

alter table AGR

add constraint PK\_AGR primary key (KEYID)

using index

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

alter table AGR

add constraint FK\_AGR\_COMPANY foreign key (COMPANYID)

references COMPANY (KEYID)

disable

novalidate;

alter table AGR

add constraint FK\_AGR\_LU foreign key (ATYPEID)

references LU (KEYID);

alter table AGR

add constraint FK\_AGR\_PATIENT foreign key (PATIENTID)

references PATIENT (KEYID);

alter table AGR

add constraint FK\_AGR\_ROOT\_AGR foreign key (ROOT\_ID)

references AGR (KEYID);

*-- Create/Recreate check constraints*

alter table AGR

add constraint CKC\_AMBSERV\_AGR

check (AmbServ in (0,1))

disable

novalidate;

alter table AGR

add constraint CKC\_BEDSERV\_AGR

check (BedServ in (0,1))

disable

novalidate;

alter table AGR

add constraint CKC\_DIAGSERV\_AGR

check (DiagServ in (0,1))

disable

novalidate;

alter table AGR

add constraint CKC\_FINANCE\_AGR

check (Finance in (0,1,2,3,4,5,6))

disable

novalidate;

alter table AGR

add constraint CKC\_OMSCODE\_AGR

check (OmsCode in (1,2))

disable

novalidate;

alter table AGR

add constraint CKC\_PAYCODE\_AGR

check (PayCode in (0,1,2,3,4,5))

disable

novalidate;

alter table AGR

add constraint CKC\_SPSERV\_AGR

check (SpServ in (0,1))

disable

novalidate;

*-- Create table*

create table DISCOUNT

(

keyid NUMBER(12) not null,

createby NUMBER(12),

createdate DATE default SYSDATE,

updateby NUMBER(12),

updatedate DATE default SYSDATE,

bgndat DATE default SYSDATE not null,

enddat DATE,

agrid NUMBER(12) not null,

depid NUMBER(12),

servid NUMBER(12),

typeid NUMBER(12) default 1 not null,

rootid NUMBER(12) default 0 not null,

koef NUMBER(15,10) default 1 not null,

note VARCHAR2(128 CHAR),

koef\_a NUMBER(15,10) default 1 not null,

koef\_s NUMBER(15,10) default 1 not null,

srvgroupid NUMBER(12)

)

tablespace MED

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 256K

next 1M

minextents 1

maxextents unlimited

);

*-- Add comments to the table*

comment on table DISCOUNT

is 'Скидки на услуги';

*-- Add comments to the columns*

comment on column DISCOUNT.keyid

is 'id коэффициента';

comment on column DISCOUNT.createby

is 'кем создана запись';

comment on column DISCOUNT.createdate

is 'дата создания записи';

comment on column DISCOUNT.updateby

is 'кем последний раз изменена запись';

comment on column DISCOUNT.updatedate

is 'дата последнего изменения';

comment on column DISCOUNT.bgndat

is 'дата начала действия коэффициента';

comment on column DISCOUNT.enddat

is 'дата окончания действия коэффициента';

comment on column DISCOUNT.agrid

is 'шифр, на который распространяется скидка';

comment on column DISCOUNT.depid

is 'отделение, на услуги которого предоставляется скидка';

comment on column DISCOUNT.servid

is 'Услуга отделения, на которую распространяется скидка';

comment on column DISCOUNT.typeid

is 'Тип коэффициента LU.TAG=14:

1 - скидка

2 - наценка за гар. письмо;';

comment on column DISCOUNT.rootid

is 'Rootid = 0

Отдельный период скидок по какому-то шифру. Скидки на услуги по данному шифру будут ссылаться на эту запись и иметь период действия указанный в полях BGNDAT и ENDDAT

ROOTID <>0 - скидки на услуги за конкретный период (указанный в rootid) по конкретному договору';

comment on column DISCOUNT.koef

is 'Значение коэффициента

Koef = 1 - нет коэффициента на цену

Koef > 1 - наценка

Koef < 1 - скидка';

comment on column DISCOUNT.note

is 'замечания';

comment on column DISCOUNT.koef\_a

is 'Коэффициент на амбулаторные услуги';

comment on column DISCOUNT.koef\_s

is 'Коэффициент на стационарные услуги';

comment on column DISCOUNT.srvgroupid

is 'Группа услуг на которые распространяется скидка';

*-- Create/Recreate indexes*

create index I\_DISCOUNT\_AGRID on DISCOUNT (AGRID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DISCOUNT\_BGNDAT\_ENDDAT on DISCOUNT (BGNDAT, ENDDAT)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DISCOUNT\_DEPID on DISCOUNT (DEPID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DISCOUNT\_ROOTID on DISCOUNT (ROOTID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DISCOUNT\_SERVID on DISCOUNT (SERVID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DISCOUNT\_SRVGROUPID on DISCOUNT (SRVGROUPID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_DISCOUNT\_TYPEID on DISCOUNT (TYPEID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

*-- Create/Recreate primary, unique and foreign key constraints*

alter table DISCOUNT

add constraint PK\_DISCOUNT primary key (KEYID)

using index

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 128K

next 1M

minextents 1

maxextents unlimited

);

alter table DISCOUNT

add constraint FK\_DISCOUNT\_AGR foreign key (AGRID)

references AGR (KEYID)

disable

novalidate;

alter table DISCOUNT

add constraint FK\_DISCOUNT\_DEP foreign key (DEPID)

references DEP (KEYID)

novalidate;

alter table DISCOUNT

add constraint FK\_DISCOUNT\_SERVICE\_GROUP foreign key (SRVGROUPID)

references SERVICE\_GROUP (KEYID);

*-- Create table*

create table LU

(

keyid NUMBER(12) not null,

createby NUMBER(12),

createdate DATE default SYSDATE,

updateby NUMBER(12),

updatedate DATE default SYSDATE,

tag NUMBER(12) not null,

shorttext VARCHAR2(128 CHAR),

text VARCHAR2(1000 CHAR),

lcode VARCHAR2(48 CHAR),

code NUMBER(12) default 0,

bgndat DATE default SYSDATE,

enddat DATE,

status NUMBER(2) default 1 not null,

oms\_code VARCHAR2(32 CHAR),

note VARCHAR2(4000 CHAR),

sortcode VARCHAR2(6 CHAR),

content\_type NUMBER(2),

color VARCHAR2(12 CHAR),

rootid NUMBER,

typeid NUMBER,

rootcode NUMBER(12),

query VARCHAR2(4000 CHAR)

)

tablespace MED

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 2M

next 1M

minextents 1

maxextents unlimited

);

*-- Add comments to the table*

comment on table LU

is 'Универсальный справочник (простой классификатор)';

*-- Add comments to the columns*

comment on column LU.createby

is 'Кто создал';

comment on column LU.createdate

is 'Дата создания';

comment on column LU.updateby

is 'Кто последний изменил';

comment on column LU.updatedate

is 'дата последнего изменения';

comment on column LU.tag

is '№ справочника. Для заголовка справочника: TAG=0, CODE=№справочника';

comment on column LU.shorttext

is 'сокращенное наименование';

comment on column LU.text

is 'наименование';

comment on column LU.lcode

is 'системный код значения';

comment on column LU.code

is 'код значения';

comment on column LU.bgndat

is 'дата начала действия';

comment on column LU.enddat

is 'дата окончания действия';

comment on column LU.status

is 'статус значения: 1 - активный

0 - архивный;';

comment on column LU.oms\_code

is 'код ОМС';

comment on column LU.note

is 'Комментарий';

comment on column LU.sortcode

is 'Порядок сортировки';

comment on column LU.content\_type

is 'тип содержимого (используется при работе инсталлера): 0-содержимое не обязательно; 1-произвольное; 2-фиксированное;';

comment on column LU.color

is 'Цвет для выделения записи, связанной с этим значением справочника';

comment on column LU.rootid

is '(не используется) Ссылка на раздел в иерархическом типе справочника';

comment on column LU.typeid

is 'тип справочника: 0-плоский; 1-иерархический (устанавливается на заголовок справочника: запись с TAG=0)';

comment on column LU.rootcode

is 'код родительской записи для содержимого иерархического справочника.';

comment on column LU.query

is 'sql-запрос';

*-- Create/Recreate indexes*

create index I\_LU\_LCODE on LU (LCODE)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 448K

next 1M

minextents 1

maxextents unlimited

);

create index I\_LU\_SHORT\_TEXT on LU (SHORTTEXT)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 448K

next 1M

minextents 1

maxextents unlimited

);

create index I\_LU\_TAG on LU (TAG)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 448K

next 1M

minextents 1

maxextents unlimited

);

create index I\_LU\_TEXT on LU (TEXT)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 960K

next 1M

minextents 1

maxextents unlimited

);

*-- Create/Recreate primary, unique and foreign key constraints*

alter table LU

add constraint PK\_LU primary key (KEYID)

using index

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 448K

next 1M

minextents 1

maxextents unlimited

);

alter table LU

add constraint UK\_LU\_TAG\_CODE unique (TAG, CODE)

using index

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

)

novalidate;

*-- Create table*

create table RNUMB

(

keyid NUMBER not null,

createby NUMBER(12),

createdate DATE default SYSDATE,

updateby NUMBER(12),

updatedate DATE default SYSDATE,

rdayid NUMBER(13),

patientid NUMBER(12),

visitid NUMBER,

docdepid NUMBER(12),

servid NUMBER(12),

dat DATE default SYSDATE,

mins NUMBER(5) default 0,

status NUMBER(2) default 1 not null,

blstatus NUMBER(2) default 0 not null,

firststatus NUMBER(2) default 0 not null,

stacstatus NUMBER(12) default 0 not null,

dopstatus NUMBER(2) default 0 not null,

room VARCHAR2(24 CHAR),

note VARCHAR2(512 CHAR),

stacid NUMBER(12),

servgroupid NUMBER(12),

paystatus NUMBER(12) default 0 not null,

examid NUMBER(12),

placeid NUMBER(12),

deviceid NUMBER(12),

dirid NUMBER(12),

service\_rgroupid NUMBER(12),

term\_max NUMBER(6),

term\_free NUMBER(6),

dat1 DATE,

root\_id NUMBER(12),

add\_qty NUMBER(3),

tag NUMBER(1),

link\_id NUMBER(12),

opendat DATE,

quota\_status NUMBER(1),

reserv\_status NUMBER(1),

expiredate DATE,

from\_internet\_status NUMBER(1),

edit\_now\_status NUMBER(1),

room\_id NUMBER(12),

main\_rnumb\_id NUMBER,

description VARCHAR2(1024 CHAR),

time\_type NUMBER(1) default 0,

sex NUMBER(1) default -1 not null,

department\_id NUMBER

)

tablespace MED

pctfree 10

initrans 1

maxtrans 255

storage

(

initial 208M

next 1M

minextents 1

maxextents unlimited

);

*-- Add comments to the table*

comment on table RNUMB

is 'Расписание. Номерок в расписании.';

*-- Add comments to the columns*

comment on column RNUMB.keyid

is 'Номерок в расписании';

comment on column RNUMB.createby

is 'Кто создал';

comment on column RNUMB.createdate

is 'Дата создания';

comment on column RNUMB.updateby

is 'Кто последний изменил';

comment on column RNUMB.updatedate

is 'дата последнего изменения';

comment on column RNUMB.rdayid

is 'День, к которому относится номерок';

comment on column RNUMB.patientid

is 'Пациент: 0, NULL - номерок еще не выдан пациенту

PATIENT.keyid - номерок выдан пациенту

-1 - номерок зарезервирован;';

comment on column RNUMB.visitid

is 'Запись в ИЛ пациента:

0, NULL - номерок еще не выдан пациенту

VISIT.keyid - номерок выдан пациенту

-1 - номерок зарезервирован;';

comment on column RNUMB.docdepid

is 'врач=DOCDEP.KeyID';

comment on column RNUMB.dat

is 'Время начала номерка

При расписании по номеркам: Дата+время номерка

При составлении расписания по периодам: Дата начала промежутка';

comment on column RNUMB.mins

is 'длительность номерка

';

comment on column RNUMB.status

is 'Статус номерка:

1 - активный

2 - фиктивный;';

comment on column RNUMB.blstatus

is 'Статус номерка:

0 - блокирован для выдачи в регистратуре

1 - активный

2 - выдан пациенту

3 - зарезервирован;';

comment on column RNUMB.firststatus

is 'Статус первичности:

0 - обычный номерок

1 - первичный номерок;';

comment on column RNUMB.stacstatus

is 'Статус стационарности:

0 - без указания

1 - стационарный

2 - амбулаторный;';

comment on column RNUMB.dopstatus

is 'Статус дополнительности:

0 - обычный (не дополнительный) номерок

1 - дополнительный номерок

2 - создан через диалог граф. выдачи;';

comment on column RNUMB.room

is 'Кабинет. Название';

comment on column RNUMB.note

is 'Комментарий';

comment on column RNUMB.stacid

is 'LU.TAG=29 Спец. отметка номерка.

LU.TAG=25 Стационары. ';

comment on column RNUMB.paystatus

is 'статус платности номерка: 1-платный; 0-нет;';

comment on column RNUMB.placeid

is 'Местоположение (DEP.KEYID)';

comment on column RNUMB.deviceid

is 'Работа по интервалам. Прибор (LU.TAG=180)';

comment on column RNUMB.dirid

is 'Работа по интервалам. Поток пациентов (LU.TAG=181)';

comment on column RNUMB.service\_rgroupid

is 'Работа по интервалам. Технология';

comment on column RNUMB.term\_max

is 'Работа по интервалам. Общая длина промежутка';

comment on column RNUMB.term\_free

is 'Работа по интервалам. Свободное время';

comment on column RNUMB.dat1

is 'Работа по интервалам. Время окончания номерка, не используется при работе по номеркам, нужно при работе по периодам';

comment on column RNUMB.root\_id

is 'Работа по интервалам. Для связки - в режиме работы по периодам - указывает на изначальный период, при заведении конкретного номерка на пациента';

comment on column RNUMB.add\_qty

is 'Количество доп. номерков (талонов)';

comment on column RNUMB.tag

is 'Резервирование. Тип резервирования номерка:

0 - для врача

1 - для учреждения заказчика

2 - для других ЛПУ (ссылка в LINK\_ID на LU - Справочник мед. учреждений)

3 - для страховой компании (ссылка в LINK\_ID на COMPANY)

4 - для гос. здрав. упр.;';

comment on column RNUMB.link\_id

is 'Резервирование. Ссылка на объект, на который резервируется номерок';

comment on column RNUMB.opendat

is 'Дата, с которой открыто расписание';

comment on column RNUMB.quota\_status

is 'выдан по квоте (1 - по квоте)';

comment on column RNUMB.reserv\_status

is 'резерв (1-резерв)';

comment on column RNUMB.expiredate

is 'дата истечения брони для Интернет-записи';

comment on column RNUMB.from\_internet\_status

is 'Пометка что номерок выдан через интернет';

comment on column RNUMB.edit\_now\_status

is 'Статус редактирования:

0, NULL - можно всем

1 - можно только мне (я редактирую сейчас, больше никому нельзя)

2 - нельзя никому (запись закрыта для изменений);';

comment on column RNUMB.room\_id

is 'Кабинет. id кабинета в справочнике LU.TAG=376';

comment on column RNUMB.main\_rnumb\_id

is 'работает в режиме ФТО (регистратура). При выдаче нескольких номерков на одну услугу, записывается keyid первого номерка';

comment on column RNUMB.time\_type

is 'Тип номерка: 0, NULL - обычный номерок; 1-интервал;';

comment on column RNUMB.sex

is 'Пол. -1 - Для всех; 0 - мужской; 1 - женский;';

comment on column RNUMB.department\_id

is 'Место работы';

*-- Create/Recreate indexes*

create index I\_RNUMB\_BLSTATUS on RNUMB (BLSTATUS)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 37M

next 1M

minextents 1

maxextents unlimited

);

create index I\_RNUMB\_DAT on RNUMB (DAT)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 35M

next 1M

minextents 1

maxextents unlimited

);

create index I\_RNUMB\_DOCDATSTATUS on RNUMB (DOCDEPID, DAT, STATUS, BLSTATUS, DOPSTATUS)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 120M

next 1M

minextents 1

maxextents unlimited

);

create index I\_RNUMB\_DOCDEPID on RNUMB (DOCDEPID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 38M

next 1M

minextents 1

maxextents unlimited

);

create index I\_RNUMB\_DOCDEPID\_DAT on RNUMB (DOCDEPID, DAT)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 59M

next 1M

minextents 1

maxextents unlimited

);

create index I\_RNUMB\_DOCDEPID\_TDAT on RNUMB (DOCDEPID, TRUNC(DAT))

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_RNUMB\_DOPSTATUS on RNUMB (DOPSTATUS)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 27M

next 1M

minextents 1

maxextents unlimited

);

create index I\_RNUMB\_EXPIREDATE on RNUMB (EXPIREDATE)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_RNUMB\_FIRSTSTATUS on RNUMB (FIRSTSTATUS)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 27M

next 1M

minextents 1

maxextents unlimited

);

create index I\_RNUMB\_LINK\_ID on RNUMB (LINK\_ID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_RNUMB\_MAIN\_RNUMB\_ID on RNUMB (MAIN\_RNUMB\_ID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_RNUMB\_PATIENTID on RNUMB (PATIENTID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_RNUMB\_PLACEID on RNUMB (PLACEID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_RNUMB\_RDAYID on RNUMB (RDAYID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 56M

next 1M

minextents 1

maxextents unlimited

);

create index I\_RNUMB\_ROOM\_ID on RNUMB (ROOM\_ID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_RNUMB\_ROOT\_ID on RNUMB (ROOT\_ID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 64K

next 1M

minextents 1

maxextents unlimited

);

create index I\_RNUMB\_STACID on RNUMB (STACID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 512K

next 1M

minextents 1

maxextents unlimited

);

create index I\_RNUMB\_STATUS on RNUMB (STATUS)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 32M

next 1M

minextents 1

maxextents unlimited

);

create index I\_RNUMB\_TRUNC\_DAT on RNUMB (TRUNC(DAT))

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 31M

next 1M

minextents 1

maxextents unlimited

);

create index I\_RNUMB\_VISITID on RNUMB (VISITID)

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 36M

next 1M

minextents 1

maxextents unlimited

);

*-- Create/Recreate primary, unique and foreign key constraints*

alter table RNUMB

add constraint PK\_RNUMB primary key (KEYID)

using index

tablespace MED\_IDX

pctfree 10

initrans 2

maxtrans 255

storage

(

initial 50M

next 1M

minextents 1

maxextents unlimited

);

alter table RNUMB

add constraint FK\_RNUMB\_RDAYID foreign key (RDAYID)

references RDAY (KEYID);