

----- Globals -----

```
create or replace package TEST.Globals
as
```

```
    procedure SetMandant (p_Mandant varchar2) ;
    procedure SetWerk    (p_Werk    varchar2) ;
```

```
    function GetMandant return varchar2 ;
    function GetWerk    return varchar2 ;
```

```
end ;
/
```

```
create or replace package body TEST.Globals
as
```

```
    cMandant varchar2(3) ;
    cWerk    varchar2(3) ;
```

```
    procedure SetMandant (p_Mandant varchar2) is
    begin
        cMandant := p_Mandant ;
    end ;
```

```
    procedure SetWerk    (p_Werk    varchar2) is
    begin
        cWerk := p_Werk ;
    end ;
```

```
    function GetMandant return varchar2 is
    begin
        return cMandant ;
    end ;
```

```
    function GetWerk    return varchar2 is
    begin
        return cWerk ;
    end ;
```

```
end ;
/
```

----- AGENTUR -----

```
create sequence TEST.AGENTUR_ID_SEQ ;
```

```
CREATE TABLE TEST.AGENTUR (
    ID                NUMBER NOT NULL,
    MANDANT_ID        NUMBER NULL,
    AGT_GRP            VARCHAR2(10) NOT NULL,
    AGT_ID             VARCHAR2(20) NOT NULL,
    GEAENDERT_AM       DATE NULL,
    GEAENDERT_DURCH    VARCHAR2(12) NULL,
    VBNR               VARCHAR2(5) NULL,
    STELLE             VARCHAR2(5) NULL,
    CONSTRAINT Agentur_PK
        PRIMARY KEY (ID),
    CONSTRAINT Agentur_AK
        UNIQUE (
            AGT_GRP,
            AGT_ID
        )
);
```

```
CREATE INDEX TEST.Agentur_Mandant_IX ON TEST.AGENTUR
(
    MANDANT_ID
);
```

```
create or replace trigger TEST.AGENTUR_BRIU
before insert or update on TEST,AGENTUR
for each row
begin
    select nvl (:NEW.ID, AGENTUR_ID_SEQ.NextVal)
    into :NEW.ID
    from Dual ;
end ;
/
```

----- GESCHAEFTSBEREICH -----

```
create sequence TEST.GESCHAEFTSBEREICH_ID_SEQ ;
```

```
CREATE TABLE TEST.GESCHAEFTSBEREICH (  
    WERK1_ID          NUMBER NULL,  
    WERK2             VARCHAR2(3) NULL,  
    BEZEICHNUNG        VARCHAR2(120) NULL,  
    PERSONALBEREICH    VARCHAR2(4) NULL,  
    NAC                VARCHAR2(3) NULL,  
    GESCHAEFTSBEREICH VARCHAR2(4) NULL,  
    BUCHUNGSKREIS      VARCHAR2(4) NULL,  
    JUR_PERSON         VARCHAR2(4) NULL,  
    ID                 NUMBER NOT NULL,  
    CONSTRAINT Geschaeftsbereich_PK  
        PRIMARY KEY (ID)  
);
```

```
CREATE INDEX TEST.Geschaeftsbereich_Werk_IX ON  
TEST.GESCHAEFTSBEREICH  
(  
    WERK1_ID  
);
```

```
CREATE INDEX TEST.Geschaeftsbereich_IX ON  
TEST.GESCHAEFTSBEREICH  
(  
    GESCHAEFTSBEREICH          ASC  
);
```

```
create or replace trigger TEST.GESCHAEFTSBEREICH_BRIU  
before insert or update on TEST,GESCHAEFTSBEREICH  
for each row  
begin  
    select nvl(:NEW.ID,GESCHAEFTSBEREICH_ID_SEQ.NextVal)  
    into :NEW.ID  
    from Dual ;  
end ;  
/
```

----- MANDANT -----

```
create sequence TEST.MANDANT_ID_SEQ ;
```

```
CREATE TABLE TEST.MANDANT (  
    ID                 NUMBER NOT NULL,  
    MANDANT            VARCHAR2(3) NOT NULL,  
    BEZEICHNUNG        VARCHAR2(50) NULL,  
    CONSTRAINT Mandant_PK  
        PRIMARY KEY (ID),  
    CONSTRAINT Mandant_AK  
    UNIQUE (  
        MANDANT  
    )  
);
```

```
create or replace trigger TEST.MANDANT_BRIU  
before insert or update on TEST,MANDANT  
for each row  
begin  
    select nvl (:NEW.ID, MANDANT_ID_SEQ.NextVal)  
    into :NEW.ID  
    from Dual ;  
end ;  
/
```

```
create or replace view TEST.TableName_VW  
as  
select T.*  
from TEST.MANDANT T  
where T.MANDANT = TEST.Globals.GetMANDANT() ;
```

----- PERSON -----

```
create sequence TEST.PERSON_ID_SEQ ;
```

```
CREATE TABLE TEST.PERSON (
  ID                NUMBER NOT NULL,
  UNTERNEHMUNG_ID   NUMBER NULL,
  PERSONALNR        VARCHAR2(15) NOT NULL,
  STATUS            VARCHAR2(1) NULL,
  AENDERUNG_STATUS  DATE NULL,
  BUCHUNGSKREIS     VARCHAR2(4) NULL,
  KOSTENSTELLE      VARCHAR2(10) NULL,
  GESCHAEFTSBEREICH VARCHAR2(3) NULL,
  AENDERUNG_KOSTENSTELLE DATE NULL,
  VORNAME           VARCHAR2(30) NULL,
  NACHNAME          VARCHAR2(30) NULL,
  AENDERUNG_NAME    DATE NULL,
  SPERRKENNZEICHEN  VARCHAR2(3) NULL,
  FUEHRUNGSEBENE    VARCHAR2(3) NULL,
  MANAGEMENTLEVEL   VARCHAR2(3) NULL,
  AENDERUNG_FUEHRUNG DATE NULL,
  WERK1             VARCHAR2(10) NULL,
  WERK2             VARCHAR2(10) NULL,
  CONSTRAINT Person_PK
    PRIMARY KEY (ID),
  CONSTRAINT Person_AK
    UNIQUE (PERSONALNR));
```

```
CREATE INDEX TEST.Person_Unternehmung_IX
ON TEST.PERSON (UNTERNEHMUNG_ID);
```

```
CREATE INDEX TEST.Person_Name_IX
ON TEST.PERSON (NACHNAME ASC,
               VORNAME  ASC);
```

```
create or replace trigger TEST.PERSON_BRIU
before insert or update on TEST,PERSON
for each row
begin
  select nvl (:NEW.ID, PERSON_ID_SEQ.NextVal)
into :NEW.ID
from Dual ;
end ;
/
```

----- UNTERNEHMUNG -----

```
create sequence TEST.UNTERNEHMUNG_ID_SEQ ;
```

```
CREATE TABLE TEST.UNTERNEHMUNG (
  ID                NUMBER NOT NULL,
  KOSTENSTELLE      VARCHAR2(20) NULL,
  GESCHAEFTSBEREICH_ID NUMBER NULL,
  CONSTRAINT Unternehmung_PK
    PRIMARY KEY (ID)
);
```

```
CREATE INDEX TEST.Unternehmung_Geschbereich_IX ON
TEST.UNTERNEHMUNG
(
  GESCHAEFTSBEREICH_ID
);
```

```
CREATE INDEX TEST.Unternehmung_Kostenstelle_IX ON
TEST.UNTERNEHMUNG
(
  KOSTENSTELLE          ASC
);
```

```
create or replace trigger TEST.UNTERNEHMUNG_BRIU
before insert or update on TEST,UNTERNEHMUNG
for each row
begin
  select nvl (:NEW.ID, UNTERNEHMUNG_ID_SEQ.NextVal)
into :NEW.ID
from Dual ;
end ;
/
```

----- VERTRAG -----

```
create sequence TEST.VERTRAG_ID_SEQ ;

CREATE TABLE TEST.VERTRAG (
    ID                NUMBER NOT NULL,
    PERSON_ID         NUMBER NULL,
    TYP               VARCHAR2(3) NULL,
    VERTRAGSNR        VARCHAR2(30) NOT NULL,
    VERTRAGSART        VARCHAR2(10) NULL,
    GUELTIG_BIS        DATE NULL,
    GUELTIG_VON        DATE NULL,
    BUERO              VARCHAR2(120) NULL,
    FIFOE             VARCHAR2(10) NULL,
    STATUS             VARCHAR2(10) NULL,
    FIRMA              VARCHAR2(120) NULL,
    VERTRAGSNR_UEBERGEORDNET VARCHAR2(30) NULL,
    UNTERNEHMUNG_ID    NUMBER NULL,
    CONSTRAINT Vertrag_PK
        PRIMARY KEY (ID),
    CONSTRAINT Vertrag_AK
        UNIQUE (VERTRAGSNR)
);
CREATE INDEX TEST.Vertrag_Person_IX
ON TEST.VERTRAG (PERSON_ID);

CREATE INDEX TEST.Vertrag_Unternehmung_IX
ON TEST.VERTRAG (UNTERNEHMUNG_ID);

CREATE INDEX TEST.Vertrag_Vertragsart_IX
ON TEST.VERTRAG (VERTRAGSART);

create or replace trigger TEST.VERTRAG_BRIU
before insert or update on TEST,VERTRAG
for each row
begin
    select nvl (:NEW.ID, VERTRAG_ID_SEQ.NextVal)
    into :NEW.ID
    from Dual ;
end ;
/
```

----- WERK -----

```
create sequence TEST.WERK_ID_SEQ ;

CREATE TABLE TEST.WERK (
    ID                NUMBER NOT NULL,
    WERK              VARCHAR2(3) NULL,
    BEZEICHNUNG        VARCHAR2(120) NULL,
    CONSTRAINT Werk_PK
        PRIMARY KEY (ID),
    CONSTRAINT Werk_AK
        UNIQUE (
            WERK
        )
);

create or replace trigger TEST.WERK_BRIU
before insert or update on TEST,WERK
for each row
begin
    select nvl (:NEW.ID, WERK_ID_SEQ.NextVal)
    into :NEW.ID
    from Dual ;
end ;
/

create or replace view TEST.TableName_VW
as
select T.*
from TEST.WERK T
where T.WERK = TEST.Globals.GetWERK() ;
```

```

ALTER TABLE TEST.AGENTUR
  ADD ( CONSTRAINT AGENTUR_MANDANT_FK
        FOREIGN KEY (MANDANT_ID)
              REFERENCES TEST.MANDANT ) ;

ALTER TABLE TEST.GESCHAEFTSBEREICH
  ADD ( CONSTRAINT GESCHAEFTSBEREICH_WERK_FK
        FOREIGN KEY (WERK1_ID)
              REFERENCES TEST.WERK ) ;

ALTER TABLE TEST.PERSON
  ADD ( CONSTRAINT PERSON_UNTERNEHMUNG_FK
        FOREIGN KEY (UNTERNEHMUNG_ID)
              REFERENCES TEST.UNTERNEHMUNG ) ;

ALTER TABLE TEST.UNTERNEHMUNG
  ADD ( CONSTRAINT UNTERNEHMUNG_GESCHAEFTSBER_FK
        FOREIGN KEY (GESCHAEFTSBEREICH_ID)
              REFERENCES TEST.GESCHAEFTSBEREICH ) ;

ALTER TABLE TEST.VERTRAG
  ADD ( CONSTRAINT VERTRAG_UNTERNEHMUNG_FK
        FOREIGN KEY (UNTERNEHMUNG_ID)
              REFERENCES TEST.UNTERNEHMUNG ) ;

ALTER TABLE TEST.VERTRAG
  ADD ( CONSTRAINT VERTRAG_PERSON_FK
        FOREIGN KEY (PERSON_ID)
              REFERENCES TEST.PERSON ) ;

```