

```
/* SQL Manager for InterBase and Firebird 5.5.3.52046 */
/* ----- */
/* Host      : localhost */
/* Database  : D:\Gabriela Fragancias\Base de datos\BASE.FDB */
```

```
CREATE DATABASE 'D:\Gabriela Fragancias\Base de datos\BASE.FDB'
  USER 'SYSDBA'
  PAGE_SIZE = 4096
  DEFAULT CHARACTER SET ASCII
  COLLATION NONE;
```

```
/* Definition for the `ID_VENTAD_GEN` generator : */
```

```
CREATE GENERATOR ID_VENTAD_GEN;
```

```
SET GENERATOR ID_VENTAD_GEN TO 3;
```

```
/* Definition for the `ENTERO` domain : */
```

```
CREATE DOMAIN ENTERO AS
  INTEGER;
```

```
/* Definition for the `NUMERO` domain : */
```

```
CREATE DOMAIN NUMERO AS
  DOUBLE PRECISION;
```

```
/* Structure for the `VENTAD` table : */
```

```
CREATE TABLE VENTAD (
  ID_VENTAD ENTERO NOT NULL,
  ID_VENTA ENTERO,
  ID_PRODUCTO ENTERO,
  CANTIDAD NUMERO,
  PRECIO NUMERO,
  COSTO NUMERO);
```

```
ALTER TABLE VENTAD ADD CONSTRAINT PK_VENTAD PRIMARY KEY (ID_VENTAD);
```

```
/* Definition for the `AI_VENTAD_ID_VENTAD` trigger : */
```

```
SET TERM ^ ;
```

```
CREATE TRIGGER AI_VENTAD_ID_VENTAD FOR VENTAD
  ACTIVE BEFORE
  INSERT
  POSITION 0
  AS
  BEGIN
    NEW.ID_VENTAD = GEN_ID(ID_VENTAD_GEN, 1);
  END^
```

```

SET TERM ; ^

/* Definition for the `AI` domain : */

CREATE DOMAIN AI AS
    SMALLINT;

/* Definition for the `DESCRIPTION_SHORT` domain : */

CREATE DOMAIN DESCRIPTION_SHORT AS
    VARCHAR(35);

/* Structure for the `GRUPO_CLIENTE` table : */

CREATE TABLE GRUPO_CLIENTE (
    ID_GRUPO_CLIENTE AI NOT NULL,
    DESCRIPCION DESCRIPTION_SHORT,
    ID_VENDEDOR ENTERO);

ALTER TABLE GRUPO_CLIENTE ADD CONSTRAINT PK_GRUPO_CLIENTE PRIMARY KEY
(ID_GRUPO_CLIENTE);

/* Definition for the `CODIGO_AREA` domain : */

CREATE DOMAIN CODIGO_AREA AS
    VARCHAR(6);

/* Structure for the `PAIS` table : */

CREATE TABLE PAIS (
    ID_PAIS AI NOT NULL,
    NOMBRE DESCRIPTION_SHORT,
    NACIONALIDAD DESCRIPTION_SHORT,
    AREA_CODIGO CODIGO_AREA);

ALTER TABLE PAIS ADD CONSTRAINT PK_PAIS PRIMARY KEY (ID_PAIS);

/* Structure for the `CIUDAD` table : */

CREATE TABLE CIUDAD (
    ID_CIUADAD AI NOT NULL,
    NOMBRE DESCRIPTION_SHORT,
    CODIGOAREA CODIGO_AREA,
    ID_PAIS AI);

ALTER TABLE CIUDAD ADD CONSTRAINT PK_CIUADAD PRIMARY KEY (ID_CIUADAD);

/* Definition for the `DESCRIPTION_LARGE` domain : */

CREATE DOMAIN DESCRIPTION_LARGE AS
    VARCHAR(80);

```

```

/* Definition for the `FECHA` domain : */

CREATE DOMAIN FECHA AS
    DATE;

/* Definition for the `MEMO` domain : */

CREATE DOMAIN MEMO AS
    VARCHAR(500);

/* Definition for the `NOMBRE_CORTO` domain : */

CREATE DOMAIN NOMBRE_CORTO AS
    VARCHAR(35);

/* Definition for the `NOMBRE_LARGO` domain : */

CREATE DOMAIN NOMBRE_LARGO AS
    VARCHAR(100);

/* Definition for the `PICTURE` domain : */

CREATE DOMAIN PICTURE AS
    BLOB SEGMENT SIZE 100;

/* Definition for the `SI_NO_NO` domain : */

CREATE DOMAIN SI_NO_NO AS
    CHAR(1)
    DEFAULT 'N'
    NOT NULL
    CHECK (VALUE IN ('S','N'));

/* Structure for the `CLIENTE` table : */

CREATE TABLE CLIENTE (
    AI_CLIENTE ENTERO NOT NULL,
    ID_GRUPO_CLIENTE AI,
    ID_CIUADAI AI,
    NOMBRE DESCRIPTION_SHORT,
    APELLIDO DESCRIPTION_LARGE,
    TELEFONO DESCRIPTION_SHORT,
    DIRECCION DESCRIPTION_LARGE,
    RUC DESCRIPTION_SHORT,
    IMAGEN PICTURE,
    MEMO MEMO,
    BARRIO DESCRIPTION_SHORT,
    CONTACTO DESCRIPTION_SHORT,
    CONTACTO2 DESCRIPTION_SHORT,
    DOCUMENTO_1 DESCRIPTION_SHORT,
    DOCUMENTO_2 DESCRIPTION_SHORT,
    CELULAR DESCRIPTION_SHORT,
    TELEFONO_1 DESCRIPTION_SHORT,

```

```

TELEFONO_2 DESCRIPTION_SHORT,
ORDEN_VISITA AI,
EMAIL NOMBRE_LARGO,
OBSERVACION MEMO,
CALIFICACION NOMBRE_CORTO,
ACTIVO ENTERO,
RECIBIRPEDIDO SI_NO_NO NOT NULL,
LATITUD DESCRIPTION_SHORT,
LONGITUD DESCRIPTION_SHORT,
FECHA_NACIMIENTO FECHA,
DIRECCIONTRABAJO MEMO,
ID_VENDEDOR ENTERO,
LUGARTRABAJO DESCRIPTION_LARGE,
LATITUD1 DESCRIPTION_SHORT,
LONGITUD1 DESCRIPTION_SHORT,
FEC_NAC DATE);

```

```

ALTER TABLE CLIENTE ADD CONSTRAINT PK_CLIENTE PRIMARY KEY (AI_CLIENTE);

```

```

/* Definition for the `ANDROIDBUSCACLIENTE` procedure : */

```

```

SET TERM ^ ;

```

```

CREATE OR ALTER PROCEDURE ANDROIDBUSCACLIENTE(
    BUSCADOR VARCHAR(135),
    VAR VARCHAR(3))
RETURNS(
    AI_CLIENTE INTEGER,
    RAZONSOCIAL VARCHAR(117),
    RUC VARCHAR(35))

```

```

AS

```

```

BEGIN

```

```

    if (VAR = '1') then

```

```

        BEGIN

```

```

            FOR

```

```

                SELECT
                    CLIENTE.AI_CLIENTE,
                    CLIENTE.NOMBRE || ', ' || CLIENTE.APELLIDO AS RAZONSOCIAL,
                    CLIENTE.RUC
                FROM
                    CLIENTE
                WHERE
                    (CLIENTE.ai_cliente = :BUSCADOR)
                INTO
                    :AI_CLIENTE,
                    :RAZONSOCIAL,
                    :RUC

```

```

            DO

```

```

                BEGIN

```

```

                    SUSPEND;

```

END

```
FOR
  SELECT
    CLIENTE.AI_CLIENTE,
    CLIENTE.NOMBRE || ', ' || CLIENTE.APELLIDO AS RAZONSOCIAL,
    CLIENTE.RUC
  FROM
    CLIENTE
  WHERE
    (CLIENTE.RUC = :BUSCADOR)
  INTO
    :AI_CLIENTE,
    :RAZONSOCIAL,
    :RUC
DO
BEGIN
  SUSPEND;
END
```

END

```
  if (VAR = '2') then
BEGIN
```

```
BUSCADOR = '%' || BUSCADOR || '%';
  FOR
    SELECT
      CLIENTE.AI_CLIENTE,
      CLIENTE.NOMBRE || ', ' || CLIENTE.APELLIDO AS RAZONSOCIAL,
      CLIENTE.RUC
    FROM
      CLIENTE
    WHERE
      (CLIENTE.nombre LIKE :BUSCADOR)
    INTO
      :AI_CLIENTE,
      :RAZONSOCIAL,
      :RUC
  DO
  BEGIN
    SUSPEND;
  END
```

```

    FOR
    SELECT
      CLIENTE.AI_CLIENTE,
      CLIENTE.NOMBRE || ', ' || CLIENTE.APELLIDO AS RAZONSOCIAL,
      CLIENTE.RUC
```

```

        FROM
        CLIENTE
    WHERE
        (CLIENTE.apellido LIKE :BUSCADOR)
    INTO
        :AI_CLIENTE,
        :RAZONSOCIAL,
        :RUC
    DO
    BEGIN
        SUSPEND;
    END

```

end

end^

SET TERM ; ^

/\* Definition for the `LISTACLIENTE` view : \*/

```

CREATE VIEW LISTACLIENTE(
    AI_CLIENTE,
    RAZONSOCIAL,
    RUC,
    DESCRIPCIONGRUPO,
    CELULAR,
    TELEFONO_1)
AS
SELECT
    CLIENTE.AI_CLIENTE,
    upper(CLIENTE.NOMBRE || ' ' || CLIENTE.APELLIDO) AS RAZONSOCIAL,
    CLIENTE.RUC,
    GRUPO_CLIENTE.DESCRIPCION AS DESCRIPCIONGRUPO,
    CLIENTE.CELULAR,
    CLIENTE.TELEFONO_1
FROM
    CLIENTE
    INNER JOIN GRUPO_CLIENTE ON
    (CLIENTE.ID_GRUPO_CLIENTE=GRUPO_CLIENTE.ID_GRUPO_CLIENTE);

```

/\* Definition for the `BUSCARCLIENTE` procedure : \*/

SET TERM ^ ;

```

CREATE OR ALTER PROCEDURE BUSCARCLIENTE(
    LETRAABUSCAR VARCHAR(150))
RETURNS (
    AI_CLIENTE INTEGER,
    RAZONSOCIAL VARCHAR(117),

```

```

RUC VARCHAR(35),
DESCRIPCIONGRUPO VARCHAR(35),
CELULAR VARCHAR(35),
TELEFONO_1 VARCHAR(35))
AS
BEGIN
LETRAABUSCAR = '%' || LETRAABUSCAR || '%';

FOR
SELECT
LISTACLIENTE.AI_CLIENTE,
LISTACLIENTE.RAZONSOCIAL,
LISTACLIENTE.RUC,
LISTACLIENTE.DESCRIPCIONGRUPO,
LISTACLIENTE.CELULAR,
LISTACLIENTE.TELEFONO_1
FROM
LISTACLIENTE
WHERE
(upper(LISTACLIENTE.RAZONSOCIAL) LIKE :LETRAABUSCAR)
INTO
:AI_CLIENTE,
:RAZONSOCIAL,
:RUC,
:DESCRIPCIONGRUPO,
:CELULAR,
:TELEFONO_1
DO
BEGIN
SUSPEND;
END

FOR
SELECT
LISTACLIENTE.AI_CLIENTE,
LISTACLIENTE.RAZONSOCIAL,
LISTACLIENTE.RUC,
LISTACLIENTE.DESCRIPCIONGRUPO,
LISTACLIENTE.CELULAR,
LISTACLIENTE.TELEFONO_1
FROM
LISTACLIENTE
WHERE
(upper(LISTACLIENTE.ruc) LIKE :LETRAABUSCAR)
INTO
:AI_CLIENTE,
:RAZONSOCIAL,
:RUC,
:DESCRIPCIONGRUPO,
:CELULAR,
:TELEFONO_1
DO
BEGIN
SUSPEND;

```

END

```
FOR
  SELECT
    LISTACLIENTE.AI_CLIENTE,
    LISTACLIENTE.RAZONSOCIAL,
    LISTACLIENTE.RUC,
    LISTACLIENTE.DESCRIPCIONGRUPO,
    LISTACLIENTE.CELULAR,
    LISTACLIENTE.TELEFONO_1
  FROM
    LISTACLIENTE
  WHERE
    (upper(LISTACLIENTE.celular) LIKE :LETRAABUSCAR)
  INTO
    :AI_CLIENTE,
    :RAZONSOCIAL,
    :RUC,
    :DESCRIPCIONGRUPO,
    :CELULAR,
    :TELEFONO_1
  DO
  BEGIN
  SUSPEND;
  END
```

```
FOR
  SELECT
    LISTACLIENTE.AI_CLIENTE,
    LISTACLIENTE.RAZONSOCIAL,
    LISTACLIENTE.RUC,
    LISTACLIENTE.DESCRIPCIONGRUPO,
    LISTACLIENTE.CELULAR,
    LISTACLIENTE.TELEFONO_1
  FROM
    LISTACLIENTE
  WHERE
    (upper(LISTACLIENTE.telefono_1) LIKE :LETRAABUSCAR)
  INTO
    :AI_CLIENTE,
    :RAZONSOCIAL,
    :RUC,
    :DESCRIPCIONGRUPO,
    :CELULAR,
    :TELEFONO_1
  DO
  BEGIN
  SUSPEND;
  END
```

FOR



```

SELECT
LISTACLIENTE.AI_CLIENTE,
LISTACLIENTE.RAZONSOCIAL,
LISTACLIENTE.RUC,
LISTACLIENTE.DESCRIPCIONGRUPO,
LISTACLIENTE.CELULAR,
LISTACLIENTE.TELEFONO_1
FROM
LISTACLIENTE
WHERE
    (upper(LISTACLIENTE.descripciongrupo) LIKE :LETRAABUSCAR)
INTO
    :AI_CLIENTE,
    :RAZONSOCIAL,
    :RUC,
    :DESCRIPCIONGRUPO,
    :CELULAR,
    :TELEFONO_1
DO
BEGIN
SUSPEND;
END

```

```

FOR
SELECT
LISTACLIENTE.AI_CLIENTE,
LISTACLIENTE.RAZONSOCIAL,
LISTACLIENTE.RUC,
LISTACLIENTE.DESCRIPCIONGRUPO,
LISTACLIENTE.CELULAR,
LISTACLIENTE.TELEFONO_1
FROM
LISTACLIENTE
WHERE
    (upper(LISTACLIENTE.ai_cliente) LIKE :LETRAABUSCAR)
INTO
    :AI_CLIENTE,
    :RAZONSOCIAL,
    :RUC,
    :DESCRIPCIONGRUPO,
    :CELULAR,
    :TELEFONO_1
DO
BEGIN
SUSPEND;
END

```

END^

SET TERM ; ^

```

/* Definition for the `ALIAS` domain : */

CREATE DOMAIN ALIAS AS
    VARCHAR(15);

/* Structure for the `CALIFICACION` table : */

CREATE TABLE CALIFICACION (
    ID_CALIFICACION ENTERO NOT NULL,
    DESCRIPCION NOMBRE_CORTO,
    DESDE ENTERO,
    HASTA ENTERO,
    SIMBOLO ALIAS);

ALTER TABLE CALIFICACION ADD CONSTRAINT PK_CALIFICACION PRIMARY KEY
(ID_CALIFICACION);

/* Definition for the `YES_NO` domain : */

CREATE DOMAIN YES_NO AS
    VARCHAR(1)
    CHECK (VALUE IN ('S','N'));

/* Structure for the `VENDEDOR` table : */

CREATE TABLE VENDEDOR (
    ID_VENDEDOR AI NOT NULL,
    NOMBRE DESCRIPTION_SHORT,
    APELLIDO_PATERNO DESCRIPTION_SHORT,
    APELLIDO_MATERNO DESCRIPTION_SHORT,
    TELEFONO DESCRIPTION_SHORT,
    EMAIL DESCRIPTION_LARGE,
    DIRECCION DESCRIPTION_LARGE,
    RUC DESCRIPTION_SHORT,
    FECHA_NACIMIENTO DATE,
    PORCENTAJE_VENTA NUMERO,
    ACTIVO YES_NO,
    ID_USUARIO ENTERO,
    CLAVEACCESO NOMBRE_CORTO);

ALTER TABLE VENDEDOR ADD CONSTRAINT PK_VENDEDOR PRIMARY KEY
(ID_VENDEDOR);

/* Definition for the `DIRECCION` domain : */

CREATE DOMAIN DIRECCION AS
    VARCHAR(250);

/* Structure for the `RECIBO_CLIENTE` table : */

CREATE TABLE RECIBO_CLIENTE (

```

```

ID_COBRANZA ENTERO NOT NULL,
ID_VENDEDOR ENTERO NOT NULL,
ID_USUARIO ENTERO NOT NULL,
FECHA DATE,
COMENTARIO DIRECION,
GUARDADO YES_NO,
TOTAL NUMERO,
COMISION NUMERO,
ID_GRUPO ENTERO,
ANA_ITEMS_CLIENTES ENTERO,
ANA_ITEMS_CLIENTES_COBRADO ENTERO,
TOTALACOBRAR NUMERO);

```

```

ALTER TABLE RECIBO_CLIENTE ADD CONSTRAINT PK_RECIBO_CLIENTE PRIMARY KEY
(ID_COBRANZA);

```

```

/* Structure for the `RECIBO_CLIENTE_DETALLE` table : */

```

```

CREATE TABLE RECIBO_CLIENTE_DETALLE (
    ID_COBRANZA_DETALLE ENTERO NOT NULL,
    ID_COBRANZA ENTERO NOT NULL,
    ID_CLIENTE ENTERO NOT NULL,
    MONTO NUMERO,
    COMSION NUMERO,
    NRORECIBO ENTERO);

```

```

ALTER TABLE RECIBO_CLIENTE_DETALLE ADD CONSTRAINT
PK_RECIBO_CLIENTE_DETALLE PRIMARY KEY (ID_COBRANZA_DETALLE);

```

```

/* Definition for the `DATE` domain : */

```

```

CREATE DOMAIN "DATE" AS
    DATE;

```

```

/* Definition for the `NUMERO_MEDIO` domain : */

```

```

CREATE DOMAIN NUMERO_MEDIO AS
    INTEGER;

```

```

/* Structure for the `CUENTA_CLIENTE` table : */

```

```

CREATE TABLE CUENTA_CLIENTE (
    ID_CUENTA_CLIENTE NUMERO_MEDIO NOT NULL,
    ID_CLIENTE ENTERO,
    ID_COMPROBANTE NUMERO_MEDIO,
    NRO NOMBRE_CORTO,
    FECHA "DATE",
    DESCRIPCION DESCRIPTION_SHORT,
    CREDITO NUMERO,
    DEBITO NUMERO,
    VENCIMIENTO "DATE",
    REFERENCIA NUMERO_MEDIO,

```

```
ID_PERSONAL AI,  
DE_HASTA ALIAS,  
ID_COBRANZA_DETALLE ENTERO,  
ID_VENTA ENTERO,  
ID_FORMA_PAGO ENTERO,  
ID_METO_PAGO ENTERO);
```

```
ALTER TABLE CUENTA_CLIENTE ADD CONSTRAINT PK_CUENTA_CLIENTE PRIMARY KEY  
(ID_CUENTA_CLIENTE);
```

```
CREATE INDEX IDX_CUENTA_CLIENTE ON CUENTA_CLIENTE(REFERENCIA);
```

```
CREATE INDEX IDX_CUENTA_CLIENTE1 ON CUENTA_CLIENTE(ID_CLIENTE);
```

```
CREATE INDEX IDX_CUENTA_CLIENTE2 ON CUENTA_CLIENTE(ID_VENTA);
```

```
/* Definition for the `CLIENTE_ANALISIS_CUENTA` procedure : */
```

```
SET TERM ^ ;
```

```
CREATE OR ALTER PROCEDURE CLIENTE_ANALISIS_CUENTA(  
    CODIGOCLIENTE INTEGER)
```

```
RETURNS (
```

```
    ID_CUENTA_CLIENTE INTEGER,  
    DEBITO DOUBLE PRECISION,  
    PAGOS DOUBLE PRECISION,  
    SALDO DOUBLE PRECISION,  
    VENCIMIENTO DATE,  
    FECHAPAGO DATE,  
    DIASVENCIDO INTEGER)
```

```
AS
```

```
BEGIN
```

```
    FOR
```

```
        SELECT
```

```
            CUENTA_CLIENTE.ID_CUENTA_CLIENTE,  
            CUENTA_CLIENTE.DEBITO,  
            CUENTA_CLIENTE.VENCIMIENTO
```

```
        FROM
```

```
            CUENTA_CLIENTE
```

```
        WHERE
```

```
            (CUENTA_CLIENTE.ID_CLIENTE = :CODIGOCLIENTE) AND  
            (CUENTA_CLIENTE.REFERENCIA IS NULL)
```

```
        ORDER BY
```

```
            CUENTA_CLIENTE.ID_CUENTA_CLIENTE
```

```
        INTO
```

```
            :ID_CUENTA_CLIENTE,  
            :DEBITO,  
            :VENCIMIENTO
```

```
    DO
```

```
    BEGIN
```

```
    SELECT
```

```

SUM(CUENTA_CLIENTE.CREDITO) AS PAGOS
FROM
CUENTA_CLIENTE
WHERE
(CUENTA_CLIENTE.ID_CLIENTE = :CODIGOCLIENTE) AND
(CUENTA_CLIENTE.REFERENCIA = :ID_CUENTA_CLIENTE)
INTO
:PAGOS;
if (PAGOS IS NULL) then
BEGIN
PAGOS = 0;
END

if (PAGOS > 0) then
BEGIN
SELECT DISTINCT
MAX(CUENTA_CLIENTE.FECHA) AS FECHAPAGO
FROM
CUENTA_CLIENTE
WHERE
(CUENTA_CLIENTE.ID_CLIENTE = :CODIGOCLIENTE) AND
(CUENTA_CLIENTE.REFERENCIA = :ID_CUENTA_CLIENTE)
INTO
:FECHAPAGO;
DIASVENCIDO = FECHAPAGO - VENCIMIENTO;
END

if (PAGOS = 0) then
begin
DIASVENCIDO = current_date - VENCIMIENTO;
end
SALDO = DEBITO - PAGOS;

SUSPEND;
FECHAPAGO = null;
DIASVENCIDO = null;

END
END^

SET TERM ; ^

/* Definition for the `CLIENTECALIFICACION` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE CLIENTECALIFICACION(
BUSCADOR INTEGER)
RETURNS (
AI_CLIENTE INTEGER,
RAZONSOCIAL VARCHAR(117),
RUC VARCHAR(35),

```

```

        CELULAR VARCHAR(35),
        CALIFICACION VARCHAR(35),
        CALIFICACIONACTUAL VARCHAR(15),
        DESCRIPCIONCALIFICACIONACTUAL VARCHAR(35),
        DIASVENCIDO INTEGER,
        ERROR CHAR(1))
AS
BEGIN
    FOR
        SELECT
            CLIENTE.AI_CLIENTE,
            CLIENTE.NOMBRE || ', ' || CLIENTE.APELLIDO AS RAZONSOCIAL,
            CLIENTE.RUC,
            CLIENTE.CELULAR,
            CLIENTE.CALIFICACION
        FROM
            CLIENTE
        WHERE
            (CLIENTE.id_ciudad = :BUSCADOR) AND
            (CLIENTE.ACTIVO = 1)
        ORDER BY
            CLIENTE.AI_CLIENTE

        INTO
            :AI_CLIENTE,
            :RAZONSOCIAL,
            :RUC,
            :CELULAR,
            :CALIFICACION
    DO
        BEGIN
            ERROR = 'N';

SELECT DISTINCT
    MAX(CLIENTE_ANALISIS_CUENTA.DIASVENCIDO) AS DIAMAXIMO
FROM
    CLIENTE_ANALISIS_CUENTA(:AI_CLIENTE)
WHERE
    (CLIENTE_ANALISIS_CUENTA.SALDO > 0)
INTO
    :DIASVENCIDO;

if (DIASVENCIDO IS NULL) then
BEGIN
DIASVENCIDO = 0;
END

if (DIASVENCIDO < 0) then
BEGIN
DIASVENCIDO = 0;
END

```

```

SELECT DISTINCT
    CALIFICACION.SIMBOLO,
    CALIFICACION.descripcion
FROM
    CALIFICACION
WHERE
    (:DIASVENCIDO BETWEEN CALIFICACION.DESDE AND CALIFICACION.HASTA)
    INTO
    :CALIFICACIONACTUAL,
    :DESCRIPCIONCALIFICACIONACTUAL;

if (CALIFICACION <> CALIFICACIONACTUAL) then
BEGIN
ERROR = 'S';
END
    SUSPEND;
    DIASVENCIDO = 0;

    END
END^

SET TERM ; ^

/* Definition for the `CLIENTE_ACTUALIZAR_DELETE` trigger : */

SET TERM ^ ;

CREATE TRIGGER CLIENTE_ACTUALIZAR_DELETE FOR CLIENTE
ACTIVE AFTER
    DELETE
POSITION 0
AS
BEGIN
    /* Trigger body */
    POST_EVENT 'NUEVOCLIENTE';
END^

SET TERM ; ^

/* Definition for the `CLIENTE_ACTUALIZAR_EDIT` trigger : */

SET TERM ^ ;

CREATE TRIGGER CLIENTE_ACTUALIZAR_EDIT FOR CLIENTE
ACTIVE AFTER
    UPDATE
POSITION 0
AS
BEGIN
    /* Trigger body */
    POST_EVENT 'NUEVOCLIENTE';

END^

```

```
SET TERM ; ^
```

```
/* Definition for the `CLIENTE_ACTUALIZAR_INSERT` trigger : */
```

```
SET TERM ^ ;
```

```
CREATE TRIGGER CLIENTE_ACTUALIZAR_INSERT FOR CLIENTE
ACTIVE AFTER
INSERT
POSITION 0
AS
BEGIN
/* Trigger body */
POST_EVENT 'NUEVOCLIENTE';
END^
```

```
SET TERM ; ^
```

```
/* Structure for the `COBRANZAMOVIL` table : */
```

```
CREATE TABLE COBRANZAMOVIL (
ID_COBRANZAMOVIL ENTERO NOT NULL,
ID_USUARIO ENTERO,
ID_VENDEDOR ENTERO,
FECHA DATE,
ID_GRUPO_CLIENTE ENTERO,
CERRADO SI_NO_NO NOT NULL,
APLICADO SI_NO_NO NOT NULL,
PASS ALIAS,
MONTOACOBRAR NUMERO,
CANTIDADCLIENTE ENTERO,
META NUMERO);
```

```
ALTER TABLE COBRANZAMOVIL ADD CONSTRAINT PK_COBRANZAMOVIL PRIMARY KEY
(ID_COBRANZAMOVIL);
```

```
/* Definition for the `DESCRIPCION_MEDIO` domain : */
```

```
CREATE DOMAIN DESCRIPCION_MEDIO AS
VARCHAR(50);
```

```
/* Structure for the `COBRANZAMOVILOBSERVACION` table : */
```

```
CREATE TABLE COBRANZAMOVILOBSERVACION (
ID_COBRANZAMOVILOBSERVACION ENTERO NOT NULL,
DESCRIPCION DESCRIPCION_MEDIO);
```

```
ALTER TABLE COBRANZAMOVILOBSERVACION ADD CONSTRAINT
PK_COBRANZAMOVILOBSERVACION PRIMARY KEY (ID_COBRANZAMOVILOBSERVACION);
```

```
/* Definition for the `HORA` domain : */
```



```
CREATE DOMAIN HORA AS  
    TIME;
```

```
/* Structure for the `COBRANZAMOVILD` table : */
```

```
CREATE TABLE COBRANZAMOVILD (  
    ID_COBRANZAMOVILD ENTERO NOT NULL,  
    ID_COBRANZAMOVIL ENTERO,  
    ID_CLIENTE ENTERO,  
    CUOTASVENCIDA ENTERO,  
    MONTOSVENCIDO NUMERO,  
    MONTOPAGADO NUMERO,  
    NRORECIBO ALIAS,  
    COBRO_LATITUD DESCRIPTION_SHORT,  
    COBRO_LONGITUD DESCRIPTION_SHORT,  
    ID_COBRANZAMOVIL OBSERVACION ENTERO,  
    PASS ALIAS,  
    FECHAULTIMOPAGO FECHA,  
    METODODEPAGO DESCRIPTION_LARGE,  
    HORA HORA,  
    SALDOACTUAL NUMERO,  
    ID_COBRANZA_DETALLE ENTERO,  
    VENDEDORA DESCRIPTION_LARGE,  
    MONTOCUOTA NUMERO,  
    NRO_COMPROBANTE NOMBRE_CORTO);
```

```
ALTER TABLE COBRANZAMOVILD ADD CONSTRAINT PK_COBRANZAMOVILD PRIMARY KEY  
(ID_COBRANZAMOVILD);
```

```
/* Definition for the `COBRANZAMOVILD_AU` trigger : */
```

```
SET TERM ^ ;
```

```
CREATE TRIGGER COBRANZAMOVILD_AU FOR COBRANZAMOVILD  
ACTIVE AFTER  
    UPDATE  
POSITION 0  
AS  
BEGIN  
    /* Trigger body */  
    POST_EVENT 'COBROMOVILACTIVO';  
END^
```

```
SET TERM ; ^
```

```
/* Structure for the `UCTABUSERS` table : */
```

```
CREATE TABLE UCTABUSERS (  
    UCIDUSER INTEGER,  
    UCUSERNAME VARCHAR(30),  
    UCLOGIN VARCHAR(30),  
    UCPASSWORD VARCHAR(250),  
    UCPASSEXPIRED CHAR(10),
```

```

UCUSEREXPIRED INTEGER,
UCUSERDAYSSUN INTEGER,
UCEMAIL VARCHAR(150),
UCPRIVILEGED INTEGER,
UCTYPEREC CHAR(1),
UCPROFILE INTEGER,
UCKEY VARCHAR(250));

/* Definition for the `COBRANZAMOVIL_LISTADO` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE COBRANZAMOVIL_LISTADO(
    DESDE DATE,
    HASTA DATE)
RETURNS (
    ID_COBRANZAMOVIL INTEGER,
    ID_USUARIO INTEGER,
    ID_VENDEDOR INTEGER,
    FECHA DATE,
    ID_GRUPO_CLIENTE INTEGER,
    CERRADO CHAR(1),
    APLICADO CHAR(1),
    PASS VARCHAR(15),
    MONTOACOBRAR DOUBLE PRECISION,
    CANTIDADCLIENTE INTEGER,
    UCUSERNAME VARCHAR(30),
    NOMBRE VARCHAR(35),
    APELLIDO_PATERNO VARCHAR(35),
    DESCRIPCION VARCHAR(35),
    MONTOCOBRADO DOUBLE PRECISION,
    META DOUBLE PRECISION,
    MONTOCUOTA DOUBLE PRECISION)
AS
BEGIN
    FOR
        SELECT
            COBRANZAMOVIL.ID_COBRANZAMOVIL,
            COBRANZAMOVIL.ID_USUARIO,
            COBRANZAMOVIL.ID_VENDEDOR,
            COBRANZAMOVIL.FECHA,
            COBRANZAMOVIL.ID_GRUPO_CLIENTE,
            COBRANZAMOVIL.CERRADO,
            COBRANZAMOVIL.APLICADO,
            COBRANZAMOVIL.PASS,
            COBRANZAMOVIL.MONTOACOBRAR,
            COBRANZAMOVIL.CANTIDADCLIENTE,
            UCTABUSERS UCUSERNAME,
            VENDEDOR.NOMBRE,
            VENDEDOR.APELLIDO_PATERNO,
            GRUPO_CLIENTE.DESCRIPCION,
            COBRANZAMOVIL.META
        FROM
            COBRANZAMOVIL

```

```

        INNER JOIN UCTABUSERS ON
(COBRANZAMOVIL.ID_USUARIO=UCTABUSERS.UCIDUSER)
        INNER JOIN VENDEDOR ON
(COBRANZAMOVIL.ID_VENDEDOR=VENDEDOR.ID_VENDEDOR)
        INNER JOIN GRUPO_CLIENTE ON
(COBRANZAMOVIL.ID_GRUPO_CLIENTE=GRUPO_CLIENTE.ID_GRUPO_CLIENTE)
WHERE
        (COBRANZAMOVIL.FECHA BETWEEN :DESDE AND :HASTA)
ORDER BY
        COBRANZAMOVIL.ID_COBRANZAMOVIL DESC

INTO
        :ID_COBRANZAMOVIL,
        :ID_USUARIO,
        :ID_VENDEDOR,
        :FECHA,
        :ID_GRUPO_CLIENTE,
        :CERRADO,
        :APLICADO,
        :PASS,
        :MONTOACOBRAR,
        :CANTIDADCLIENTE,
        :UCUSERNAME,
        :NOMBRE,
        :APELLIDO_PATERNO,
        :DESCRIPCION,
        :META

DO
BEGIN
SELECT
        SUM(COBRANZAMOVILD.MONTOPAGADO) AS MONTOCOBRADO,
        SUM(COBRANZAMOVILD.montocuota) AS MONTOCUOTA
FROM
        COBRANZAMOVILD
WHERE
        (COBRANZAMOVILD.ID_COBRANZAMOVIL = :ID_COBRANZAMOVIL)
INTO
        :MONTOCOBRADO,
        :MONTOCUOTA;

        SUSPEND;
END
END^

SET TERM ; ^

/* Definition for the `COBRANZAPORLINEAL` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE COBRANZAPORLINEAL(
        CODIGOZONA INTEGER)
RETURNS (

```

```

AI_CLIENTE INTEGER,
RAZONSOCIAL VARCHAR(117),
DESCRIPCION VARCHAR(35),
RUC VARCHAR(35),
DIRECCION VARCHAR(80),
CELULAR VARCHAR(35),
TELEFONO VARCHAR(35))
AS
BEGIN
    FOR
        SELECT
            CLIENTE.AI_CLIENTE,
            CLIENTE.NOMBRE || ', ' || CLIENTE.APELLIDO AS RAZONSOCIAL,
            GRUPO_CLIENTE.DESCRIPCION,
            CLIENTE.RUC,
            CLIENTE.DIRECCION,
            CLIENTE.CELULAR,
            CLIENTE.telefono
        FROM
            CLIENTE
            INNER JOIN GRUPO_CLIENTE ON
(CLIENTE.ID_GRUPO_CLIENTE=GRUPO_CLIENTE.ID_GRUPO_CLIENTE)
        WHERE
            (CLIENTE.ID_GRUPO_CLIENTE = :CODIGOZONA)
        ORDER BY
            CLIENTE.ORDEN_VISITA

        INTO
            :AI_CLIENTE,
            :RAZONSOCIAL,
            :DESCRIPCION,
            :RUC,
            :DIRECCION,
            :CELULAR,
            :TELEFONO
    DO
        BEGIN
            SUSPEND;
        END
    END^

```

```

SET TERM ; ^

```

```

/* Structure for the `FORMA_PAGO` table : */

```

```

CREATE TABLE FORMA_PAGO (
    ID_FORMA_PAGO ENTERO NOT NULL,
    DESCRIPCION NOMBRE_CORTO);

```

```

ALTER TABLE FORMA_PAGO ADD CONSTRAINT PK_FORMA_PAGO PRIMARY KEY
(ID_FORMA_PAGO);

```

```

/* Definition for the `COBRANZAPORLINEALDETALLE` procedure : */

```

```
SET TERM ^ ;
```

```
CREATE OR ALTER PROCEDURE COBRANZAPORLINEALDETALLE(  
    CODIGOCLIENTE INTEGER,  
    HASTAFECHA DATE)
```

```
RETURNS (  
    ID_CUENTA_CLIENTE INTEGER,  
    FECHA DATE,  
    DESCRIPCION VARCHAR(35),  
    DEBITO DOUBLE PRECISION,  
    VENCIMIENTO DATE,  
    PAGOS DOUBLE PRECISION,  
    CANCELADO CHAR(1),  
    CANTIDADVECIDO INTEGER,  
    MONTOVENCIDO DOUBLE PRECISION,  
    MONTOAVENCER DOUBLE PRECISION,  
    NROCOMPROBANTE VARCHAR(35),  
    NOMBREVENDEDOR VARCHAR(35),  
    FORMADEPAGO VARCHAR(35))
```

```
AS
```

```
BEGIN
```

```
    CANCELADO = 'N';  
    CANTIDADVECIDO = 0;  
    MONTOVENCIDO = 0;  
    MONTOAVENCER = 0;  
    FOR
```

```
SELECT
```

```
    CUENTA_CLIENTE.ID_CUENTA_CLIENTE,  
    CUENTA_CLIENTE.FECHA,  
    CUENTA_CLIENTE.DESCRIPCION,  
    CUENTA_CLIENTE.DEBITO,  
    CUENTA_CLIENTE.VENCIMIENTO,  
    CUENTA_CLIENTE.NRO,  
    VENDEDOR.NOMBRE,  
    FORMA_PAGO.DESCRIPCION
```

```
FROM
```

```
    CUENTA_CLIENTE
```

```
    INNER JOIN VENDEDOR ON (CUENTA_CLIENTE.ID_PERSONAL=VENDEDOR.ID_VENDEDOR)
```

```
    INNER JOIN FORMA_PAGO ON
```

```
(CUENTA_CLIENTE.ID_FORMA_PAGO=FORMA_PAGO.ID_FORMA_PAGO)
```

```
WHERE
```

```
    (CUENTA_CLIENTE.ID_CLIENTE = :CODIGOCLIENTE) AND  
    (CUENTA_CLIENTE.VENCIMIENTO <= :HASTAFECHA) AND  
    (CUENTA_CLIENTE.REFERENCIA IS NULL)
```

```
ORDER BY
```

```
    CUENTA_CLIENTE.ID_CUENTA_CLIENTE
```

```
INTO
```

```
    :ID_CUENTA_CLIENTE,  
    :FECHA,  
    :DESCRIPCION,  
    :DEBITO,  
    :VENCIMIENTO,
```

```

        :nrocomprobante,
        :NOMBREVENDEDOR,
        :FORMADEPAGO
DO
BEGIN

SELECT
SUM(CUENTA_CLIENTE.CREDITO) AS SUMAPAGOS
FROM
CUENTA_CLIENTE
WHERE
((CUENTA_CLIENTE.id_cliente = :CODIGOCLIENTE) AND
(CUENTA_CLIENTE.REFERENCIA = :ID_CUENTA_CLIENTE))
INTO
:PAGOS;
if (PAGOS IS NULL) then
BEGIN
    PAGOS = 0;
END

if (PAGOS >= DEBITO) then
BEGIN
    CANCELADO = 'S';
END
ELSE
BEGIN
    if (VENCIMIENTO < current_date) then
    begin
        CANTIDADVECIDO = CANTIDADVECIDO + 1;
        MONTOVENCIDO = (MONTOVENCIDO + (DEBITO - PAGOS));
    end
    MONTOAVENCER = MONTOAVENCER + (DEBITO - PAGOS);
END
if (DEBITO > PAGOS) then
BEGIN
    SUSPEND;
END

    CANCELADO = 'N';
    CANTIDADVECIDO = 0;
    MONTOVENCIDO = 0;
    MONTOAVENCER = 0;
END
END^

SET TERM ; ^

/* Definition for the `COMERCIAL_ULTIMO_PAGO_CLIENTE` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE COMERCIAL_ULTIMO_PAGO_CLIENTE (
    CODIGOCLIENTE INTEGER)
RETURNS (

```

```

        FECHA DATE,
        CREDITO DOUBLE PRECISION)
AS
BEGIN
    FOR
SELECT
    CUENTA_CLIENTE.FECHA,
    SUM(CUENTA_CLIENTE.CREDITO) AS FIELD_1
FROM
    CUENTA_CLIENTE
WHERE
    (CUENTA_CLIENTE.REFERENCIA IS NOT NULL) AND
    (CUENTA_CLIENTE.ID_CLIENTE = :CODIGOCLIENTE)
GROUP BY
    CUENTA_CLIENTE.FECHA
ORDER BY
    CUENTA_CLIENTE.FECHA

        INTO
            :FECHA,
            :CREDITO
DO
BEGIN
END
    SUSPEND;
END^

SET TERM ; ^

/* Definition for the `COBRANZAPORLINEALCABECERA` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE COBRANZAPORLINEALCABECERA(
    CODIGOZONA INTEGER,
    HASTAFECHA DATE)
RETURNS (
    AI_CLIENTE INTEGER,
    RAZONSOCIAL VARCHAR(117),
    DESCRIPCION VARCHAR(35),
    RUC VARCHAR(35),
    DIRECCION VARCHAR(80),
    CELULAR VARCHAR(35),
    PRODUCTO VENDIDO VARCHAR(35),
    CANTIDAD VECIDO INTEGER,
    MONTO VENCIDO DOUBLE PRECISION,
    MONTO AVENCER DOUBLE PRECISION,
    TELEFONO VARCHAR(35),
    MEMORANDU VARCHAR(500),
    FECHA DE VENTA DATE,
    MONTO PAGOS DOUBLE PRECISION,
    DEUDA TOTAL DOUBLE PRECISION,
    SALDO ACTUAL DOUBLE PRECISION,
    FECHA ULTIMO PAGO DATE,

```

```

MONTOULTIMOPAGO DOUBLE PRECISION,
MONTOCUOTAS DOUBLE PRECISION,
NOMBREVEDE VARCHAR(35),
FORMADEPA VARCHAR(35),
ORDENVISITA INTEGER,
CALIFICIACION VARCHAR(35),
DIRECCIONLABORAL VARCHAR(500),
LUGARDETRABAJO VARCHAR(85),
NRO_COMPROBANTEVENTA VARCHAR(35))
AS
BEGIN
    FOR

        SELECT
            CLIENTE.AI_CLIENTE,
            CLIENTE.NOMBRE || ', ' || CLIENTE.APELLIDO AS RAZONSOCIAL,
            GRUPO_CLIENTE.DESCRIPCION,
            CLIENTE.RUC,
            CLIENTE.DIRECCION,
            CLIENTE.CELULAR,
            CLIENTE.telefono,
            CLIENTE.memo,
            cliente.orden_visita,
            cliente.calificacion,
            cliente.direcciontrabajo,
            CLIENTE.lugartrabajo
        FROM
            CLIENTE
            INNER JOIN GRUPO_CLIENTE ON
(CLIENTE.ID_GRUPO_CLIENTE=GRUPO_CLIENTE.ID_GRUPO_CLIENTE)
        WHERE
            (CLIENTE.ID_GRUPO_CLIENTE = :CODIGOZONA)
        ORDER BY
            CLIENTE.ORDEN_VISITA
        INTO
            :AI_CLIENTE,
            :RAZONSOCIAL,
            :DESCRIPCION,
            :RUC,
            :DIRECCION,
            :CELULAR,
            :TELEFONO,
            :MEMORANDU,
            :ordenvisita,
            :calificiacion,
            :direccionlaboral,
            :LUGARDETRABAJO

    DO
BEGIN

    SELECT
    SUM(CUENTA_CLIENTE.DEBITO) AS DEBITOS,
    SUM(CUENTA_CLIENTE.CREDITO) AS CREDITOS,
    SUM(CUENTA_CLIENTE.DEBITO - CUENTA_CLIENTE.CREDITO) AS SALDOACTUAL

```



```

FROM
  CUENTA_CLIENTE
WHERE
  (CUENTA_CLIENTE.ID_CLIENTE = :AI_CLIENTE)
  INTO
    :DEUDATOTAL,
    :MONTOPAGOS,
    :SALDOACTUAL;

SELECT
  COMERCIAL_ULTIMO_PAGO_CLIENTE.FECHA,
  COMERCIAL_ULTIMO_PAGO_CLIENTE.CREDITO
FROM
  COMERCIAL_ULTIMO_PAGO_CLIENTE(:AI_CLIENTE)
  into
    :fechaultimopago,
    :montoultimopago;

FOR

SELECT
  COBRANZAPORLINEALDETALLE.DESCRIPCION,
  COBRANZAPORLINEALDETALLE.FECHA,
  COBRANZAPORLINEALDETALLE.DEBITO,
  COBRANZAPORLINEALDETALLE.NOMBREVENDEDOR,
  COBRANZAPORLINEALDETALLE.FORMADEPAGO,
  COBRANZAPORLINEALDETALLE.nrocomprobante,
  SUM(COBRANZAPORLINEALDETALLE.CANTIDADVECIDO) AS CANTIDADVECIDO,
  SUM(COBRANZAPORLINEALDETALLE.MONTOVENCIDO) AS MONTOVENCIDO,
  SUM(COBRANZAPORLINEALDETALLE.MONTOAVENCER) AS MONTOAVENCER
FROM
  COBRANZAPORLINEALDETALLE(:AI_CLIENTE, :HASTAFECHA)
GROUP BY
  COBRANZAPORLINEALDETALLE.DESCRIPCION,
  COBRANZAPORLINEALDETALLE.FECHA,
  COBRANZAPORLINEALDETALLE.DEBITO,
  COBRANZAPORLINEALDETALLE.NOMBREVENDEDOR,
  COBRANZAPORLINEALDETALLE.FORMADEPAGO,
  COBRANZAPORLINEALDETALLE.nrocomprobante
ORDER BY
  COBRANZAPORLINEALDETALLE.DEBITO
  INTO
    :PRODUCTOVENDIDO,
    :FECHADEVENTA,
    :MONTOCUOTAS,
    :NOMBREVEDE,
    :FORMADEPA,
    :NRO_COMPROBANTEVENTA,
    :CANTIDADVECIDO,
    :MONTOVENCIDO,
    :MONTOAVENCER

DO

```

```

BEGIN

END
if (MONTOCUOTAS > 0) then
begin
    SUSPEND;
    PRODUCTOVENDIDO = null;
    FECHADEVENTA = null;
    MONTOCUOTAS = null;
    NOMBREVEDE = null;
    FORMADEPA = null;
    CANTIDADVECIDO = null;
    MONTOVENCIDO = null;
    MONTOAVENCER = null;
end

END
END^

SET TERM ; ^

/* Definition for the `COBRANZA_LISTADO` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE COBRANZA_LISTADO(
    DESDE DATE,
    HASTA DATE,
    ANALISIS CHAR(1))
RETURNS (
    ID_COBRANZA INTEGER,
    ID_VENDEDOR INTEGER,
    NOMBRE VARCHAR(35),
    ID_USUARIO INTEGER,
    UCUSERNAME VARCHAR(30),
    FECHA DATE,
    COMENTARIO VARCHAR(250),
    GUARDADO VARCHAR(1),
    TOTAL DOUBLE PRECISION,
    COMISION DOUBLE PRECISION,
    DESCRIPCIONGRUPO VARCHAR(35),
    ITEMSDELGRUPO INTEGER,
    ITEMSDELGRUPOCOBRADO INTEGER,
    PORCENTAJECOBRA DO DOUBLE PRECISION,
    IDCIUDAD INTEGER,
    MONTOACOBRA DO DOUBLE PRECISION,
    PORCENTAJECOBRA DO EN MONTO DOUBLE PRECISION)
AS
BEGIN
    PORCENTAJECOBRA DO = 0;
    PORCENTAJECOBRA DO EN MONTO = 0;

```

```

FOR
SELECT
    RECIBO_CLIENTE.ID_COBRANZA,
    RECIBO_CLIENTE.ID_VENDEDOR,
    VENDEDOR.NOMBRE,
    RECIBO_CLIENTE.ID_USUARIO,
    UCTABUSERS.UCUSERNAME,
    RECIBO_CLIENTE.FECHA,
    RECIBO_CLIENTE.COMENTARIO,
    RECIBO_CLIENTE.GUARDADO,
    CIUDAD.NOMBRE,
    RECIBO_CLIENTE.id_grupo,
    RECIBO_CLIENTE.totalacobrar
FROM
    RECIBO_CLIENTE
    INNER JOIN VENDEDOR ON (RECIBO_CLIENTE.ID_VENDEDOR=VENDEDOR.ID_VENDEDOR)
    INNER JOIN UCTABUSERS ON (RECIBO_CLIENTE.ID_USUARIO=UCTABUSERS.UCIDUSER)
    INNER JOIN CIUDAD ON (RECIBO_CLIENTE.ID_GRUPO=CIUDAD.ID_CIUADAD)
WHERE
    (RECIBO_CLIENTE.FECHA BETWEEN :DESDE AND :HASTA)
ORDER BY
    RECIBO_CLIENTE.ID_COBRANZA DESC

```

```

    INTO
        :ID_COBRANZA,
        :ID_VENDEDOR,
        :NOMBRE,
        :ID_USUARIO,
        :UCUSERNAME,
        :FECHA,
        :COMENTARIO,
        :GUARDADO,
        :DESCRIPCIONGRUPO,
        :IDCIUDAD,
        :MONTOACOBRAR
DO
BEGIN

    PORCENTAJECOBRAADO = 0;
    PORCENTAJECOBRAADOENMONTO = 0;

    select count(recibo_cliente_detalle.id_cobranza_detalle) from
    recibo_cliente_detalle where recibo_cliente_detalle.id_cobranza =
    :ID_COBRANZA INTO :ITEMSDELGRUPOCOBRADO;

```

```

if (ITEMSDELGRUPO IS NULL) THEN
BEGIN
    ITEMSDELGRUPO = 0;
END
if (ITEMSDELGRUPOCOBRADO IS NULL) then
BEGIN
    ITEMSDELGRUPOCOBRADO = 0;

```

```

END

SELECT
SUM(RECIBO_CLIENTE_DETALLE.MONTO) AS COBRANZA,
SUM(RECIBO_CLIENTE_DETALLE.COMSION) AS COMISION
FROM
RECIBO_CLIENTE_DETALLE
WHERE
(RECIBO_CLIENTE_DETALLE.ID_COBRANZA = :ID_COBRANZA)
INTO
:TOTAL,
:COMISION;

if (MONTOACOBRAR IS NULL) then
BEGIN
MONTOACOBRAR = 0;
END
if (MONTOACOBRAR > 0) then
BEGIN
    PORCENTAJECOBRAO = ((TOTAL * 100) / MONTOACOBRAR);
    PORCENTAJECOBRAOENMONTO = PORCENTAJECOBRAO;
END

    SUSPEND;
END
END^

SET TERM ; ^

/* Definition for the `COBRANZA_LISTADO_WEB` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE COBRANZA_LISTADO_WEB(
    CODIGOCOBRAO INTEGER)
RETURNS (
    ID_COBRANZA INTEGER,
    ID_VENDEDOR INTEGER,
    NOMBRE VARCHAR(35),
    ID_USUARIO INTEGER,
    UCUSERNAME VARCHAR(30),
    FECHA DATE,
    COMENTARIO VARCHAR(250),
    GUARDADO VARCHAR(1),
    TOTAL DOUBLE PRECISION,
    DESCRIPCIONGRUPO VARCHAR(35))
AS
BEGIN
    FOR
    SELECT

```

```

    RECIBO_CLIENTE.ID_COBRANZA,
    RECIBO_CLIENTE.ID_VENDEDOR,
    VENDEDOR.NOMBRE,
    RECIBO_CLIENTE.ID_USUARIO,
    UCTABUSERS.UCUSERNAME,
    RECIBO_CLIENTE.FECHA,
    RECIBO_CLIENTE.COMENTARIO,
    RECIBO_CLIENTE.GUARDADO,
    GRUPO_CLIENTE.DESCRIPCION
FROM
    RECIBO_CLIENTE
    INNER JOIN VENDEDOR ON (RECIBO_CLIENTE.ID_VENDEDOR=VENDEDOR.ID_VENDEDOR)
    INNER JOIN UCTABUSERS ON (RECIBO_CLIENTE.ID_USUARIO=UCTABUSERS.UCIDUSER)
    INNER JOIN GRUPO_CLIENTE ON
    (RECIBO_CLIENTE.ID_GRUPO=GRUPO_CLIENTE.ID_GRUPO_CLIENTE)
WHERE
    (RECIBO_CLIENTE.guardado = 'N') and (recibo_cliente.id_vendedor =
:codigocobrador)
ORDER BY
    RECIBO_CLIENTE.ID_COBRANZA DESC
    INTO
        :ID_COBRANZA,
        :ID_VENDEDOR,
        :NOMBRE,
        :ID_USUARIO,
        :UCUSERNAME,
        :FECHA,
        :COMENTARIO,
        :GUARDADO,
        :DESCRIPCIONGRUPO
DO
BEGIN
SELECT
    SUM(RECIBO_CLIENTE_DETALLE.MONTO) AS COBRANZA
FROM
    RECIBO_CLIENTE_DETALLE
WHERE
    (RECIBO_CLIENTE_DETALLE.ID_COBRANZA = :ID_COBRANZA)
    INTO
        :TOTAL;

    SUSPEND;
END
END^

SET TERM ; ^

/* Definition for the `COBRODETALLEACTIVOS` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE COBRODETALLEACTIVOS(
    CODIGOGRUPO INTEGER,
    FECHACOBRO DATE)

```

```

RETURNS (
    AI_CLIENTE INTEGER,
    ID_CUENTA_CLIENTE INTEGER,
    FECHA DATE,
    VENCIMIENTO DATE,
    DEBITO DOUBLE PRECISION,
    CREDITO DOUBLE PRECISION,
    CANCELAR CHAR(1),
    RAZONSOCIAL VARCHAR(135),
    RUC VARCHAR(35))
AS
BEGIN
    FOR
        SELECT
            CLIENTE.AI_CLIENTE,
            CUENTA_CLIENTE.ID_CUENTA_CLIENTE,
            CUENTA_CLIENTE.FECHA,
            CUENTA_CLIENTE.VENCIMIENTO,
            CUENTA_CLIENTE.DEBITO
        FROM
            CUENTA_CLIENTE
        INNER JOIN CLIENTE ON (CUENTA_CLIENTE.ID_CLIENTE=CLIENTE.AI_CLIENTE)
        WHERE
            (CLIENTE.ID_GRUPO_CLIENTE = :CODIGOGRUPO) AND
            (CUENTA_CLIENTE.REFERENCIA IS NULL) AND
            (CUENTA_CLIENTE.VENCIMIENTO <= :FECHACOBRO) AND
            (CUENTA_CLIENTE.DEBITO > 0)

        INTO
            :AI_CLIENTE,
            :ID_CUENTA_CLIENTE,
            :FECHA,
            :VENCIMIENTO,
            :DEBITO
    DO
        BEGIN
            CANCELAR = 'N';

            SELECT
                SUM(CUENTA_CLIENTE.CREDITO) AS CREDITO
            FROM
                CUENTA_CLIENTE
            WHERE
                ((CUENTA_CLIENTE.REFERENCIA = :ID_CUENTA_CLIENTE)
                AND
                (CUENTA_CLIENTE.fecha < :FECHACOBRO))
            INTO
                :CREDITO;
            if (CREDITO IS NULL) then
                BEGIN
                    CREDITO = 0;
                END
            END
        END
    END

```

```

if (CREDITO >= DEBITO) then
BEGIN
    CANCELAR = 'S';
END

if ((CANCELAR = 'N') or (VENCIMIENTO = FECHACOBRO)) then
BEGIN
    SELECT
    CLIENTE.RUC,
    CLIENTE.NOMBRE || ', ' || CLIENTE.APELLIDO AS RAZONSOCIAL
FROM
    CLIENTE
WHERE
    (CLIENTE.AI_CLIENTE = :AI_CLIENTE)
    INTO
    :RUC,
    :RAZONSOCIAL;
    SUSPEND;
END

END
END^

SET TERM ; ^

/* Definition for the `COBROSRESUMENCONITECLIACOB1` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE COBROSRESUMENCONITECLIACOB1(
    CODIGOGRUPO INTEGER,
    FECHACOBRO DATE)
RETURNS (
    AI_CLIENTE INTEGER,
    ID_CUENTA_CLIENTE INTEGER,
    FECHA DATE,
    VENCIMIENTO DATE,
    DEBITO DOUBLE PRECISION,
    CREDITO DOUBLE PRECISION,
    CANCELAR CHAR(1))
AS
BEGIN
    FOR
        SELECT
            CLIENTE.AI_CLIENTE,
            CUENTA_CLIENTE.ID_CUENTA_CLIENTE,
            CUENTA_CLIENTE.FECHA,
            CUENTA_CLIENTE.VENCIMIENTO,
            CUENTA_CLIENTE.DEBITO
        FROM
            CUENTA_CLIENTE
        INNER JOIN CLIENTE ON (CUENTA_CLIENTE.ID_CLIENTE=CLIENTE.AI_CLIENTE)
        WHERE
            (CLIENTE.ID_GRUPO_CLIENTE = :CODIGOGRUPO) AND

```

```

        (CUENTA_CLIENTE.REFERENCIA IS NULL) AND
        (CUENTA_CLIENTE.VENCIMIENTO <= :FECHACOBRO) AND
        (CUENTA_CLIENTE.DEBITO > 0) and
        (CUENTA_CLIENTE.descripcion <> 'Entrega')

    INTO
        :AI_CLIENTE,
        :ID_CUENTA_CLIENTE,
        :FECHA,
        :VENCIMIENTO,
        :DEBITO
DO
BEGIN
    CANCELAR = 'N';

    SELECT
        SUM(CUENTA_CLIENTE.CREDITO) AS CREDITO
FROM
    CUENTA_CLIENTE
WHERE
    ((CUENTA_CLIENTE.REFERENCIA = :ID_CUENTA_CLIENTE)
    AND
    (CUENTA_CLIENTE.fecha < :FECHACOBRO))
    INTO
        :CREDITO;
    if (CREDITO IS NULL) then
    BEGIN
        CREDITO = 0;
    END

    if (CREDITO >= DEBITO) then
    BEGIN
        CANCELAR = 'S';
    END

    if (CANCELAR = 'N') then
    BEGIN
        SUSPEND;
    END

    END
END^

SET TERM ; ^

/* Definition for the `COBROSRESUMENCONITECLIACOB2` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE COBROSRESUMENCONITECLIACOB2 (
    CODIGOGRUPO INTEGER,
    FECHACOBRO DATE)
RETURNS (

```



```

    CONTADOR INTEGER)
AS
declare variable AI_CLIENTE INTEGER;
BEGIN
CONTADOR = 0;

    FOR
        SELECT
            COBROSRESUMENCONITECLIACOB1.AI_CLIENTE
        FROM
            COBROSRESUMENCONITECLIACOB1 (:CODIGOGRUPO, :FECHACOBRO)
        GROUP BY
            COBROSRESUMENCONITECLIACOB1.AI_CLIENTE
        INTO
            :AI_CLIENTE
    DO
        BEGIN
CONTADOR = CONTADOR + 1;

        END
        SUSPEND;

END^

SET TERM ; ^

/* Definition for the `COBROSRESUMENCONITECLIACOB` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE COBROSRESUMENCONITECLIACOB(
    ID_GRUPO_CLIENTE INTEGER,
    FECHACOBRO DATE)
RETURNS (
    CONTADOR INTEGER)
AS
BEGIN
CONTADOR = 0;

    SELECT
        COBROSRESUMENCONITECLIACOB2.CONTADOR
    FROM
        COBROSRESUMENCONITECLIACOB2 (:ID_GRUPO_CLIENTE, :FECHACOBRO)
    INTO
        :CONTADOR;

        SUSPEND;
        CONTADOR = CONTADOR + 1;

END^

```

```

SET TERM ; ^

/* Definition for the `COBROSRESUMEN` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE COBROSRESUMEN(
    BUSCADOR INTEGER,
    FECHADECOBRO DATE)
RETURNS (
    ID_GRUPO_CLIENTE SMALLINT,
    DESCRIPCION VARCHAR(35),
    COBRO DOUBLE PRECISION,
    CANTIDADCLIENTECOBRAO INTEGER,
    CANTIDADCLIENTEDELGRUPO INTEGER,
    PORCENTAJECOBRAO DOUBLE PRECISION)
AS
BEGIN
    FOR
        SELECT
            GRUPO_CLIENTE.ID_GRUPO_CLIENTE,
            GRUPO_CLIENTE.DESCRIPCION,
            SUM(RECIBO_CLIENTE_DETALLE.MONTO) AS COBRO,
            COUNT(RECIBO_CLIENTE_DETALLE.ID_CLIENTE) AS CANTIDADCLIENTECOBRAO
        FROM
            RECIBO_CLIENTE_DETALLE
            INNER JOIN CLIENTE ON
                (RECIBO_CLIENTE_DETALLE.ID_CLIENTE=CLIENTE.AI_CLIENTE)
            INNER JOIN GRUPO_CLIENTE ON
                (CLIENTE.ID_GRUPO_CLIENTE=GRUPO_CLIENTE.ID_GRUPO_CLIENTE)
        WHERE
            (RECIBO_CLIENTE_DETALLE.ID_COBRANZA = :BUSCADOR)
        GROUP BY
            GRUPO_CLIENTE.ID_GRUPO_CLIENTE,
            GRUPO_CLIENTE.DESCRIPCION

    INTO
        :ID_GRUPO_CLIENTE,
        :DESCRIPCION,
        :COBRO,
        :CANTIDADCLIENTECOBRAO
    DO
    BEGIN

        SELECT COBROSRESUMENCONITECLIACOB.contador
        FROM COBROSRESUMENCONITECLIACOB(:ID_GRUPO_CLIENTE, :FECHADECOBRO)
        INTO
            :CANTIDADCLIENTEDELGRUPO;

        if (CANTIDADCLIENTEDELGRUPO > 0) then
            BEGIN

```

```

        PORCENTAJECOBRAO = ((CANTIDADCLIENTECOBRAO * 100) /
CANTIDADCLIENTEDELGRUPO);
    END

    SUSPEND;
    CANTIDADCLIENTEDELGRUPO = 0;

END
END^

SET TERM ; ^

/* Definition for the `COMERCIAL_LISTADO_CLIENTE` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE COMERCIAL_LISTADO_CLIENTE(
    MOSTRARESTADODECUENTA CHAR(1))
RETURNS (
    AI_CLIENTE INTEGER,
    RAZON_SOCIAL VARCHAR(117),
    DESCRIPCION VARCHAR(35),
    TELEFONO VARCHAR(35),
    RUC VARCHAR(35),
    EMAIL VARCHAR(100),
    BARRIO VARCHAR(35),
    CELULAR VARCHAR(35),
    CALIFICACION VARCHAR(35),
    ACTIVO INTEGER,
    TOTALDEUDA DOUBLE PRECISION,
    TOTALPAGOS DOUBLE PRECISION,
    SALDOACTUAL DOUBLE PRECISION,
    PORCENTAJEPAGO DOUBLE PRECISION,
    NOMBRECLIENTE VARCHAR(35),
    APELLIDOCLIENTE VARCHAR(80),
    DIASVENCIDO INTEGER,
    DIRECCION VARCHAR(100),
    ORDENVISITA INTEGER,
    FECHANACIMIENTO DATE,
    DIRECCIONTRABAJO VARCHAR(500),
    ID_VENDEDOR INTEGER,
    NOMBREVENDEDOR VARCHAR(35),
    NOMBRECIUDAD VARCHAR(35),
    LUGARDETRABAJO VARCHAR(80),
    FECHAULTIMOPAGO DATE,
    MONTOULTIMOPAGO DOUBLE PRECISION,
    OBSERVACION VARCHAR(500))
AS
BEGIN
    FOR
    SELECT
        CLIENTE.AI_CLIENTE,
        CLIENTE.NOMBRE || ', ' || CLIENTE.APELLIDO AS RAZON_SOCIAL,
        GRUPO_CLIENTE.DESCRIPCION,

```

```

    CLIENTE.TELEFONO,
    CLIENTE.RUC,
    CLIENTE.EMAIL,
    CLIENTE.BARRIO,
    CLIENTE.CELULAR,
    CLIENTE.CALIFICACION,
    CLIENTE.ACTIVO,
    CLIENTE.NOMBRE,
    CLIENTE.APELLIDO,
    CLIENTE.DIRECCION,
    CLIENTE.ORDEN_VISITA,
    CLIENTE.FECHA_NACIMIENTO,
    CLIENTE.DIRECCIONTRABAJO,
    CLIENTE.ID_VENDEDOR,
    CIUDAD.NOMBRE,
    CLIENTE.lugartrabajo,
    CLIENTE.observacion
FROM
    CLIENTE
    INNER JOIN GRUPO_CLIENTE ON
    (CLIENTE.ID_GRUPO_CLIENTE=GRUPO_CLIENTE.ID_GRUPO_CLIENTE)
    INNER JOIN CIUDAD ON (CLIENTE.ID_CIUADAD=CIUDAD.ID_CIUADAD)
ORDER BY
    CLIENTE.AI_CLIENTE
    INTO
    :AI_CLIENTE,
    :RAZON_SOCIAL,
    :DESCRIPCION,
    :TELEFONO,
    :RUC,
    :EMAIL,
    :BARRIO,
    :CELULAR,
    :CALIFICACION,
    :ACTIVO,
    :NOMBRECLIENTE,
    :APELLIDOCIENTE,
    :DIRECCION,
    :ORDENVISITA,
    :FECHANACIMIENTO,
    :DIRECCIONTRABAJO,
    :ID_VENDEDOR,
    :NOMBRECIUDAD,
    :LUGARDETRABAJO,
    :OBSERVACION
DO
BEGIN

    TOTALDEUDA = 0;
    TOTALPAGOS = 0;
    SALDOACTUAL = 0;
    PORCENTAJEPAGO = 0;

```

```

        if (MOSTRARESTADODECUENTA = 'S') then
        BEGIN
        SELECT
        SUM(CUENTA_CLIENTE.DEBITO) AS TOTALDEUDA,
        SUM(CUENTA_CLIENTE.CREDITO) AS TOTALPAGOS
FROM
        CUENTA_CLIENTE
WHERE
        (CUENTA_CLIENTE.ID_CLIENTE = :AI_CLIENTE)
        INTO
        :TOTALDEUDA,
        :TOTALPAGOS;
        if (TOTALPAGOS IS NULL) then
        BEGIN
        TOTALPAGOS = 0;
        END
        if (TOTALDEUDA IS NULL) then
        BEGIN
        TOTALDEUDA = 0;
        END
        SALDOACTUAL = TOTALDEUDA - TOTALPAGOS;

        if (TOTALDEUDA > 0) then
        BEGIN
                PORCENTAJEPAGO = TOTALPAGOS * 100 / TOTALDEUDA;
        END

                SELECT
                COMERCIAL_ULTIMO_PAGO_CLIENTE.FECHA,
                COMERCIAL_ULTIMO_PAGO_CLIENTE.CREDITO
FROM
        COMERCIAL_ULTIMO_PAGO_CLIENTE(:AI_CLIENTE)
        into
        :FECHAULTIMOPAGO,
        :MONTOULTIMOPAGO;

        SELECT DISTINCT
        MAX(CLIENTE_ANALISIS_CUENTA.DIASVENCIDO) AS DIASVENCIDO
FROM
        CLIENTE_ANALISIS_CUENTA(:AI_CLIENTE)
        into
        :DIASVENCIDO;

        END

        if (ID_VENDEDOR IS NOT NULL) then
        BEGIN
        SELECT
        VENDEDOR.NOMBRE
FROM

```

```

VENDEDOR
WHERE
  (VENDEDOR.ID_VENDEDOR = :ID_VENDEDOR)
  INTO
  :NOMBREVENDEDOR;

END

SUSPEND;

TOTALDEUDA = 0;
TOTALPAGOS = 0;
SALDOACTUAL = 0;
PORCENTAJEPAGO = 0;
ID_VENDEDOR = NULL;
NOMBREVENDEDOR = NULL;

END
END^

SET TERM ; ^

/* Definition for the `COMERCIAL_MAXIMOS_PARA_FICHA` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE COMERCIAL_MAXIMOS_PARA_FICHA(
  CODIGOCLIENTE INTEGER)
RETURNS (
  DIASVENCIDO INTEGER,
  SALDO DOUBLE PRECISION,
  MONTOTOTALVENCIDO DOUBLE PRECISION)
AS
BEGIN
  FOR
    SELECT DISTINCT
      CLIENTE_ANALISIS_CUENTA.DIASVENCIDO,
      CLIENTE_ANALISIS_CUENTA.SALDO
    FROM
      CLIENTE_ANALISIS_CUENTA(:CODIGOCLIENTE)
    WHERE
      (CLIENTE_ANALISIS_CUENTA.SALDO > 0)
    ORDER BY
      CLIENTE_ANALISIS_CUENTA.DIASVENCIDO asc

    INTO
      :DIASVENCIDO,
      :SALDO

  DO
  BEGIN
  END

  SELECT DISTINCT
  SUM(CLIENTE_ANALISIS_CUENTA.SALDO) AS MONTOTOTALVENCIDO

```

```

FROM
  CLIENTE_ANALISIS_CUENTA(:CODIGOCLIENTE)
WHERE
  (CLIENTE_ANALISIS_CUENTA.SALDO > 0) AND
  (CLIENTE_ANALISIS_CUENTA.DIASVENCIDO > 0)
INTO
  :MONTOTOTALVENCIDO;

```

```

      SUSPEND;

```

```

END^

```

```

SET TERM ; ^

```

```

/* Structure for the `ALMACEN_MOVI` table : */

```

```

CREATE TABLE ALMACEN_MOVI (
  ID_ALMACEN_MO ENTERO NOT NULL,
  DESCRIPCION DESCRIPTION_LARGE,
  FECHA DATE,
  NATU AI,
  GUARDADO YES_NO);

```

```

ALTER TABLE ALMACEN_MOVI ADD CONSTRAINT PK_ALMACEN_MOVI PRIMARY KEY
(ID_ALMACEN_MO);

```

```

/* Structure for the `ALMACEN_MOVI_D` table : */

```

```

CREATE TABLE ALMACEN_MOVI_D (
  ID_DETALLE ENTERO NOT NULL,
  ID_ALMACEN_MO NUMERO_MEDIO,
  ID_PRODUCTO AI,
  CANTIDAD NUMERO,
  PRECIO NUMERO,
  SALIDA NUMERO);

```

```

ALTER TABLE ALMACEN_MOVI_D ADD CONSTRAINT PK_ALMACEN_MOVI_D PRIMARY KEY
(ID_DETALLE);

```

```

/* Structure for the `PRODUCTO` table : */

```

```

CREATE TABLE PRODUCTO (
  ID_PRODUCTO ENTERO NOT NULL,
  DESCRIPCION NOMBRE_CORTO,
  CODIGO_BARRAS NOMBRE_CORTO,
  PRECIO1 NUMERO,
  PRECIO2 NUMERO,
  PRECIO3 NUMERO,
  PESO NUMERO,
  OFERTAR SI_NO_NO NOT NULL,

```

```

PRECIO_COSTO NUMERO,
ID_TIPO_PRODUCTO ENTERO,
ID_MARCA_PRODUCTO ENTERO,
ID_PRODUCTO_SEXO ENTERO);

/* Definition for the `STOCKACTUALDEUNPRODUCTO` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE STOCKACTUALDEUNPRODUCTO(
    IDPRODUCTO INTEGER)
RETURNS (
    SALDOACTUAL DOUBLE PRECISION,
    CAPITAL DOUBLE PRECISION)
AS
BEGIN
    FOR
    SELECT
        SUM(ALMACEN_MOVI_D.CANTIDAD - ALMACEN_MOVI_D.SALIDA) AS SALDOACTUAL,
        SUM(((ALMACEN_MOVI_D.CANTIDAD - ALMACEN_MOVI_D.SALIDA) *
ALMACEN_MOVI_D.PRECIO)) AS CAPITAL
    FROM
        ALMACEN_MOVI_D
    INNER JOIN ALMACEN_MOVI ON
        (ALMACEN_MOVI_D.ID_ALMACEN_MO=ALMACEN_MOVI.ID_ALMACEN_MO)
    WHERE
        (ALMACEN_MOVI.NATU = 1) AND
        (ALMACEN_MOVI.GUARDADO = 'S') AND
        (ALMACEN_MOVI_D.CANTIDAD > ALMACEN_MOVI_D.SALIDA) AND
        (ALMACEN_MOVI_D.ID_PRODUCTO = :IDPRODUCTO)
    INTO
        :SALDOACTUAL,
        :CAPITAL
    DO
    BEGIN
        if (SALDOACTUAL IS NULL) then
        BEGIN
            SALDOACTUAL = 0;
            CAPITAL = 0;
        END
        SUSPEND;
    END
END^

SET TERM ; ^

```

```

/* Definition for the `COMPROBARSTOCKPARADESCONTAR` procedure : */

```

```

SET TERM ^ ;

```

```

CREATE OR ALTER PROCEDURE COMPROBARSTOCKPARADESCONTAR(
    IDTRANSACTION INTEGER)
RETURNS (
    ID_PRODUCTO INTEGER,

```



```

    CANTIDADVENDIDO DOUBLE PRECISION,
    STOCKACTUAL DOUBLE PRECISION,
    SEPUEDEDESCONTAR CHAR(1),
    DESCRIPCIONDELPRODUCTO VARCHAR(35))
AS
BEGIN
    FOR
SELECT
    ALMACEN_MOVI_D.ID_PRODUCTO,
    SUM(ALMACEN_MOVI_D.CANTIDAD) AS CANTIDADVENDIDO,
    PRODUCTO.DESCRIPCION
FROM
    ALMACEN_MOVI_D
    INNER JOIN PRODUCTO ON (ALMACEN_MOVI_D.ID_PRODUCTO=PRODUCTO.ID_PRODUCTO)
WHERE
    (ALMACEN_MOVI_D.ID_ALMACEN_MO = :IDTRANSACION)
GROUP BY
    ALMACEN_MOVI_D.ID_PRODUCTO,
    PRODUCTO.DESCRIPCION

    INTO
        :ID_PRODUCTO,
        :CANTIDADVENDIDO,
        :DESCRIPCIONDELPRODUCTO
DO
BEGIN
    SELECT stockactualdeunproducto.saldoactual FROM
stockactualdeunproducto(:ID_PRODUCTO) INTO :STOCKACTUAL;
    if (CANTIDADVENDIDO > STOCKACTUAL) then
        BEGIN
            SEPUEDEDESCONTAR = 'N';
        END
        ELSE
        BEGIN
            SEPUEDEDESCONTAR = 'S';
        END
        SUSPEND;
    END
END^

SET TERM ; ^

/* Structure for the `VENTAMADE` table : */

CREATE TABLE VENTAMADE (
    ID_VENTAMADE ENTERO NOT NULL,
    ID_VENTAMAYO ENTERO,
    ID_PRODUCTO ENTERO,
    CANTIDAD NUMERO,
    UNITARIO NUMERO,
    DESCONTADO SI_NO_NO NOT NULL,
    COSTO NUMERO);

```

```

ALTER TABLE VENTAMADE ADD CONSTRAINT PK_VENTAMADE PRIMARY KEY
(ID_VENTAMADE);

/* Definition for the `CONTROLVENTACANTISTOCKDIS` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE CONTROLVENTACANTISTOCKDIS (
    CODIGOVENTA INTEGER)
RETURNS (
    ID_VENTAMADE INTEGER,
    ID_PRODUCTO INTEGER,
    DESCRIPCION VARCHAR(35),
    CANTIDAD DOUBLE PRECISION,
    STOCKACTUAL DOUBLE PRECISION,
    DESCONTABLE CHAR(1))
AS
BEGIN
    FOR
        SELECT
            VENTAMADE.ID_VENTAMADE,
            VENTAMADE.ID_PRODUCTO,
            PRODUCTO.DESCRIPCION,
            VENTAMADE.CANTIDAD
        FROM
            VENTAMADE
            INNER JOIN PRODUCTO ON (VENTAMADE.ID_PRODUCTO=PRODUCTO.ID_PRODUCTO)
        WHERE
            (VENTAMADE.ID_VENTAMAYO = :CODIGOVENTA)

        INTO
            :ID_VENTAMADE,
            :ID_PRODUCTO,
            :DESCRIPCION,
            :CANTIDAD

    DO
        BEGIN
            DESCONTABLE = 'N';

            SELECT
                SUM(ALMACEN_MOVI_D.CANTIDAD - ALMACEN_MOVI_D.SALIDA) AS STOCKACTUAL
            FROM
                ALMACEN_MOVI_D
                INNER JOIN ALMACEN_MOVI ON
                (ALMACEN_MOVI_D.ID_ALMACEN_MO=ALMACEN_MOVI.ID_ALMACEN_MO)
            WHERE
                (ALMACEN_MOVI.NATU = 1) AND
                (ALMACEN_MOVI.GUARDADO = 'S') AND
                (ALMACEN_MOVI_D.CANTIDAD > ALMACEN_MOVI_D.SALIDA) AND
                (ALMACEN_MOVI_D.ID_PRODUCTO = :ID_PRODUCTO)
            INTO
                :STOCKACTUAL;
            if (STOCKACTUAL IS NULL) then
                BEGIN

```

```

STOCKACTUAL = 0;
END

if (STOCKACTUAL >= CANTIDAD) then
    BEGIN
        DESCONTABLE = 'S';
    END

    SUSPEND;
END
END^

SET TERM ; ^

/* Definition for the `CORRECCIONERROR1` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE CORRECCIONERROR1
RETURNS (
    ID_CUENTA_CLIENTE INTEGER,
    ID_CLIENTE INTEGER,
    FECHA DATE,
    DESCRIPCION VARCHAR(35),
    CREDITO DOUBLE PRECISION,
    DEBITO DOUBLE PRECISION,
    VENCIMIENTO DATE,
    REFERENCIA INTEGER)
AS
BEGIN
    FOR
        SELECT
            CUENTA_CLIENTE.ID_CUENTA_CLIENTE,
            CUENTA_CLIENTE.ID_CLIENTE,
            CUENTA_CLIENTE.FECHA,
            CUENTA_CLIENTE.DESCRIPCION,
            CUENTA_CLIENTE.DEBITO,
            CUENTA_CLIENTE.VENCIMIENTO,
            CUENTA_CLIENTE.REFERENCIA
        FROM
            CUENTA_CLIENTE
        WHERE
            (CUENTA_CLIENTE.REFERENCIA IS NULL)
        ORDER BY
            CUENTA_CLIENTE.ID_CUENTA_CLIENTE

    INTO
        :ID_CUENTA_CLIENTE,
        :ID_CLIENTE,
        :FECHA,
        :DESCRIPCION,
        :DEBITO,
        :VENCIMIENTO,
        :REFERENCIA

```

```

DO
BEGIN

    SELECT
    SUM(CUENTA_CLIENTE.CREDITO) AS CREDITO
FROM
    CUENTA_CLIENTE
WHERE
    (CUENTA_CLIENTE.REFERENCIA = :ID_CUENTA_CLIENTE)
    INTO
    :CREDITO;

    if (CREDITO > DEBITO ) then
    BEGIN
        update cuenta_cliente
        set cuenta_cliente.debito = :credito
        where cuenta_cliente.id_cuenta_cliente = :ID_CUENTA_CLIENTE;

        SUSPEND;

    END

    END
END^

SET TERM ; ^

/* Structure for the `METO_PAGO` table : */

CREATE TABLE METO_PAGO (
    ID_METO_PAGO ENTERO NOT NULL,
    DESCRIPCION NOMBRE_CORTO);

ALTER TABLE METO_PAGO ADD CONSTRAINT PK_METO_PAGO PRIMARY KEY
(ID_METO_PAGO);

/* Definition for the `LISTADO_CUENTA_CLIENTE` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE LISTADO_CUENTA_CLIENTE(
    CODIGOCLIENTE INTEGER)
RETURNS (
    ID_CUENTA_CLIENTE INTEGER,
    ID_COMPROBANTE INTEGER,
    NR0 VARCHAR(35),
    FECHA DATE,
    CREDITO DOUBLE PRECISION,
    DEBITO DOUBLE PRECISION,
    DESCRIPCION VARCHAR(35),
    VENCECIMIENTO DATE,
    CANCELADO CHAR(1),
    ID_CLIENTE INTEGER,

```

```

    DESDE_HASTA VARCHAR(15),
    DIASVENCIDA INTEGER,
    SALDOLINEAL DOUBLE PRECISION,
    IDDEVENTA INTEGER,
    SALDOINCREMENTAL DOUBLE PRECISION,
    ID_FORMADEPAGO INTEGER,
    ID_VENDEDOR INTEGER,
    NOMBREVENDEDOR VARCHAR(35),
    APAGAR DOUBLE PRECISION,
    ID_FORMA_PAGO INTEGER,
    DESCRIPCIONFORMADEPAGO VARCHAR(35))
AS
declare variable fechaultimopago date;
BEGIN

diasvencida = 0;
SALDOINCREMENTAL = 0;

APAGAR = 0;

FOR
SELECT
    CUENTA_CLIENTE.ID_CUENTA_CLIENTE,
    CUENTA_CLIENTE.ID_COMPROBANTE,
    CUENTA_CLIENTE.NRÖ,
    CUENTA_CLIENTE.FECHA,
    CUENTA_CLIENTE.DEBITO,
    CUENTA_CLIENTE.CREDITO,
    CUENTA_CLIENTE.DESCRIPCION,
    CUENTA_CLIENTE.VENCIMIENTO,
    CUENTA_CLIENTE.ID_CLIENTE,
    CUENTA_CLIENTE.DE_HASTA,
    CUENTA_CLIENTE.ID_VENTA,
    CUENTA_CLIENTE.ID_FORMA_PAGO,
    CUENTA_CLIENTE.ID_PERSONAL,
    CUENTA_CLIENTE.ID_METO_PAGO,
    METO_PAGO.DESCRIPCION
FROM
    CUENTA_CLIENTE
    INNER JOIN METO_PAGO ON
(CUENTA_CLIENTE.ID_METO_PAGO=METO_PAGO.ID_METO_PAGO)
WHERE
    (CUENTA_CLIENTE.ID_CLIENTE = :CODIGOCLIENTE) AND
    (CUENTA_CLIENTE.REFERENCIA IS NULL)
ORDER BY
    CUENTA_CLIENTE.ID_CUENTA_CLIENTE DESC
    INTO
        :ID_CUENTA_CLIENTE,
        :ID_COMPROBANTE,
        :NRÖ,
        :FECHA,
        :DEBITO,

```

```

        :CREDITO,
        :DESCRIPCION,
        :VENCECIMIENTO,
        :ID_CLIENTE,
        :DESDE_HASTA,
        :IDDEVENTA,
        :ID_FORMADEPAGO,
        :ID_VENDEDOR,
        :ID_FORMA_PAGO,
        :DESCRIPCIONFORMADEPAGO
DO
BEGIN

    SELECT
    VENDEDOR.NOMBRE
FROM
    VENDEDOR
WHERE
    (VENDEDOR.ID_VENDEDOR = :ID_VENDEDOR)
    into
    :NOMBREVENDEDOR;

if (credito = 0) then
begin

    SELECT
    SUM(CUENTA_CLIENTE.CREDITO) AS CREDITO
FROM
    CUENTA_CLIENTE
WHERE
    (CUENTA_CLIENTE.REFERENCIA = :ID_CUENTA_CLIENTE)
    INTO
    :CREDITO;
END

    if (credito is null) then
    begin
    credito = 0;
    end

    if (credito >= debito) then
    begin
    CANCELADO = 'S';
        SELECT DISTINCT
        MAX(CUENTA_CLIENTE.FECHA) AS fechaultimopago
        FROM
        CUENTA_CLIENTE
        WHERE
        (CUENTA_CLIENTE.REFERENCIA = :ID_CUENTA_CLIENTE)
        into
        :fechaultimopago;
        diasvencida = VENCECIMIENTO - fechaultimopago;
        SALDOLINEAL = 0;
    end
    end

```

```

end
ELSE
    begin
        CANCELADO = 'N';
        if (VENCECIMIENTO < current_date) then
            BEGIN
                diasvencida = CURRENT_DATE - VENCECIMIENTO;
            END
            SALDOLINEAL = DEBITO - CREDITO;
        end
        SALDOINCREMENTAL = SALDOINCREMENTAL + (DEBITO - CREDITO);
        SUSPEND;
        SALDOLINEAL = 0;
        diasvencida = 0;
    END
END^

SET TERM ; ^

/* Definition for the `CUENTA_A_VENCER_DE_CLIENTES_GRU` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE CUENTA_A_VENCER_DE_CLIENTES_GRU(
    CODIGO_GRUPO INTEGER,
    FIN DATE)
RETURNS (
    ID_CUENTA_CLIENTE INTEGER,
    FACTURA VARCHAR(35),
    NRO VARCHAR(35),
    FECHA DATE,
    DESCRIPCION VARCHAR(35),
    DEBITO DOUBLE PRECISION,
    CREDITO DOUBLE PRECISION,
    VENCIMIENTO DATE,
    RUC VARCHAR(35),
    ID_CLIENTE INTEGER,
    DIRECCION VARCHAR(80),
    MEMOCLI VARCHAR(500),
    CLIENTECOMPLETO VARCHAR(130),
    TELEFONO VARCHAR(35),
    CELULAR VARCHAR(35),
    SUMA_DEBITO DOUBLE PRECISION,
    SUMA_CREDITO DOUBLE PRECISION,
    DIFERENCIA_SUMA DOUBLE PRECISION,
    FECHAULTIMOPAGO DATE,
    MONTOULTIMOPAGO DOUBLE PRECISION,
    PUNTOSOBTENIDOS INTEGER,
    NRODECUOTA VARCHAR(15))
AS
BEGIN
    SUMA_DEBITO = 0;
    SUMA_CREDITO = 0;
    DIFERENCIA_SUMA = 0;

```

```
PUNTOSOBTENIDOS = 0;
```

```
FOR
SELECT
  CUENTA_CLIENTE.ID_CUENTA_CLIENTE,
  CUENTA_CLIENTE.NRO,
  CUENTA_CLIENTE.FECHA,
  CUENTA_CLIENTE.DESCRIPCION,
  CUENTA_CLIENTE.DEBITO,
  CUENTA_CLIENTE.VENCIMIENTO,
  CLIENTE.nombre || ',' || CLIENTE.apellido AS CLIENTECOMPLETO,
  CUENTA_CLIENTE.ID_CLIENTE,
  CLIENTE.DIRECCION,
  CLIENTE.MEMO,
  CLIENTE.TELEFONO,
  CLIENTE.CELULAR,
  CLIENTE.RUC,
  CUENTA_CLIENTE.de_hasta
FROM
  CUENTA_CLIENTE
  RIGHT OUTER JOIN CLIENTE ON
(CUENTA_CLIENTE.ID_CLIENTE=CLIENTE.AI_CLIENTE)
WHERE
  (CLIENTE.ID_GRUPO_CLIENTE = :CODIGO_GRUPO) AND
  (CUENTA_CLIENTE.VENCIMIENTO <= :FIN) AND
  (CUENTA_CLIENTE.REFERENCIA IS NULL)
ORDER BY
  CLIENTE.ORDEN_VISITA,
  CUENTA_CLIENTE.descripcion
```

```
INTO
  :ID_CUENTA_CLIENTE,
  :NRO,
  :FECHA,
  :DESCRIPCION,
  :DEBITO,
  :VENCIMIENTO,
  :CLIENTECOMPLETO,
  :ID_CLIENTE,
  :DIRECCION,
  :MEMOCLI,
  :TELEFONO,
  :CELULAR,
  :RUC,
  :NRODECUOTA
```

```
DO
begin
```

```
FACTURA = 'Credito';
```

```
if (PUNTOSOBTENIDOS IS NULL) then
BEGIN
  PUNTOSOBTENIDOS = 0;
```



```

END

SELECT
SUM(LISTADO_CUENTA_CLIENTE.CREDITO) AS SUMA_CREDITO,
SUM(LISTADO_CUENTA_CLIENTE.DEBITO) AS SUMA_DEBITO,
SUM(LISTADO_CUENTA_CLIENTE.DEBITO - LISTADO_CUENTA_CLIENTE.CREDITO) AS
DIFERENCIA_SUMA
FROM
LISTADO_CUENTA_CLIENTE(:ID_CLIENTE)
INTO
:SUMA_CREDITO,
:SUMA_DEBITO,
:DIFERENCIA_SUMA;

begin

SELECT
SUM(CUENTA_CLIENTE.CREDITO) AS CREDITO
FROM
CUENTA_CLIENTE
WHERE
(CUENTA_CLIENTE.REFERENCIA = :ID_CUENTA_CLIENTE)
INTO
:CREDITO;
if (CREDITO is null) then
begin
CREDITO =0;
end

if (DEBITO > CREDITO) then
BEGIN
execute procedure COMERCIAL_ULTIMO_PAGO_CLIENTE :id_cliente
returning_values FECHAULTIMOPAGO, MONTOULTIMOPAGO;
SUSPEND;
END
end

end
end^

SET TERM ; ^

/* Definition for the `CUENTA_CLIENTE_ELIMINAR` trigger : */

SET TERM ^ ;

CREATE TRIGGER CUENTA_CLIENTE_ELIMINAR FOR CUENTA_CLIENTE
ACTIVE AFTER
DELETE
POSITION 0
AS
BEGIN
/* Trigger body */
POST_EVENT 'NUEVOCUENTA';

```

END^

SET TERM ; ^

/\* Definition for the `CUENTA\_CLIENTE\_MODIFICAR` trigger : \*/

SET TERM ^ ;

CREATE TRIGGER CUENTA\_CLIENTE\_MODIFICAR FOR CUENTA\_CLIENTE  
ACTIVE AFTER

UPDATE

POSITION 0

AS

BEGIN

/\* Trigger body \*/

POST\_EVENT 'NUEVOCUENTA';

END^

SET TERM ; ^

/\* Definition for the `CUENTA\_CLIENTE\_RE\_EDIT` trigger : \*/

SET TERM ^ ;

CREATE TRIGGER CUENTA\_CLIENTE\_RE\_EDIT FOR RECIBO\_CLIENTE  
ACTIVE AFTER

UPDATE

POSITION 2

AS

BEGIN

/\* Trigger body \*/

POST\_EVENT 'NUEVOCUENTA';

END^

SET TERM ; ^

/\* Definition for the `CUENTA\_CLIENTE\_RE\_INS` trigger : \*/

SET TERM ^ ;

CREATE TRIGGER CUENTA\_CLIENTE\_RE\_INS FOR RECIBO\_CLIENTE  
ACTIVE AFTER

INSERT

POSITION 3

AS

BEGIN

POST\_EVENT 'NUEVOCUENTA';

END^

SET TERM ; ^

/\* Definition for the `CUENTA\_DE\_CLIENTE\_POR\_GRUPO\_DE` procedure : \*/

SET TERM ^ ;

```

CREATE OR ALTER PROCEDURE CUENTA_DE_CLIENTE_POR_GRUPO_DE (
    CODIGOCLIENTE INTEGER,
    FECHAVENCIMIENTO DATE)
RETURNS (
    ID_CUENTA_CLIENTE INTEGER,
    DEBITO DOUBLE PRECISION,
    VENCIMIENTO DATE,
    CANTIDADDECUOTASVENCIDA INTEGER,
    CREDITO DOUBLE PRECISION,
    CANCELADO VARCHAR(1),
    DIASVENCIDOS INTEGER,
    SALDO DOUBLE PRECISION,
    DESCRIPCION VARCHAR(35),
    TOTALDEUDA DOUBLE PRECISION,
    TOTALPAGOS DOUBLE PRECISION,
    TOTALSALDO DOUBLE PRECISION)
AS
BEGIN
    DIASVENCIDOS = 0;
    CANTIDADDECUOTASVENCIDA = 0;
    SALDO = 0;

    SELECT
        SUM(CUENTA_CLIENTE.CREDITO) AS TOTALPAGOS,
        SUM(CUENTA_CLIENTE.DEBITO) AS TOTALDEUDA
    FROM
        CUENTA_CLIENTE
    WHERE
        (CUENTA_CLIENTE.ID_CLIENTE = :codigocliente)
    into
        :TOTALPAGOS,
        :TOTALDEUDA;
    if (TOTALDEUDA IS NULL) then
        BEGIN
            TOTALDEUDA = 0;
        END
    if (TOTALPAGOS IS NULL) then
        BEGIN
            TOTALPAGOS = 0;
        END

    TOTALSALDO = TOTALDEUDA - TOTALPAGOS;

    FOR
        SELECT
            CUENTA_CLIENTE.ID_CUENTA_CLIENTE,
            CUENTA_CLIENTE.DEBITO,
            CUENTA_CLIENTE.VENCIMIENTO,
            CUENTA_CLIENTE.descripcion
        FROM
            CUENTA_CLIENTE
        WHERE
            (CUENTA_CLIENTE.ID_CLIENTE = :CODIGOCLIENTE) AND

```

```

        (CUENTA_CLIENTE.REFERENCIA IS NULL) AND
        (CUENTA_CLIENTE.VENCIMIENTO <= :FEHAVENCIMIENTO)
    INTO
        :ID_CUENTA_CLIENTE,
        :DEBITO,
        :VENCIMIENTO,
        :DESCRIPCION
    DO
    BEGIN
        SELECT
            SUM(CUENTA_CLIENTE.CREDITO) AS CREDITO
    FROM
        CUENTA_CLIENTE
    WHERE
        (CUENTA_CLIENTE.REFERENCIA = :ID_CUENTA_CLIENTE)
    INTO
        :CREDITO;
    if (CREDITO IS NULL) then
    BEGIN
        CREDITO = 0;
    END

    SALDO = DEBITO - CREDITO;

    if (credito >= debito) then
    begin
        CANCELADO = 'S';
        DIASVENCIDOS = 0;
    end
    ELSE
        begin
            CANCELADO = 'N';
            if (VENCIMIENTO < FEHAVENCIMIENTO) then
            BEGIN
                DIASVENCIDOS = FEHAVENCIMIENTO - VENCIMIENTO;
                CANTIDADDECUOTASVENCIDA = CANTIDADDECUOTASVENCIDA + 1;
            END
            SUSPEND;
            DIASVENCIDOS = 0;
        end

    END
END^

SET TERM ; ^

/* Definition for the `CUENTA_DE_CLIENTE_POR_GRUPO` procedure : */

SET TERM ^ ;

```

```

CREATE OR ALTER PROCEDURE CUENTA_DE_CLIENTE_POR_GRUPO (
    CODIGOGRUPO INTEGER,
    FECHADECOBRO DATE)
RETURNS (
    AI_CLIENTE INTEGER,
    ID_GRUPO_CLIENTE SMALLINT,
    DESCRIPCION VARCHAR(35),
    RAZON_SOCIAL VARCHAR(120),
    TELEFONO VARCHAR(35),
    CELULAR VARCHAR(35),
    RUC VARCHAR(35),
    DIRECCION VARCHAR(80),
    MEMO VARCHAR(500),
    ORDEN_VISITA SMALLINT,
    DEUDAALAFECHA DOUBLE PRECISION,
    CANTIDADDECUOTASVENCIDA INTEGER,
    CANTIDADDEDIASVENCIDOHASTACOBRO INTEGER,
    DESCRIPCIONCUENTA VARCHAR(35),
    TOTALDEUDA DOUBLE PRECISION,
    TOTALPAGOS DOUBLE PRECISION,
    TOTALSALDO DOUBLE PRECISION,
    FECHAAIMPRIMIR DATE)
AS
declare variable NOMBRE VARCHAR (120) CHARACTER SET NONE;
declare variable APELLIDO VARCHAR (35) CHARACTER SET NONE;
BEGIN
FECHAAIMPRIMIR = FECHADECOBRO;
FOR
    SELECT
        CLIENTE.AI_CLIENTE,
        GRUPO_CLIENTE.ID_GRUPO_CLIENTE,
        GRUPO_CLIENTE.DESCRIPCION,
        CLIENTE.NOMBRE,
        CLIENTE.APELLIDO,
        CLIENTE.TELEFONO,
        CLIENTE.CELULAR,
        CLIENTE.RUC,
        CLIENTE.DIRECCION,
        CLIENTE.MEMO,
        CLIENTE.ORDEN_VISITA
    FROM
        CLIENTE
    INNER JOIN GRUPO_CLIENTE ON
(CLIENTE.ID_GRUPO_CLIENTE=GRUPO_CLIENTE.ID_GRUPO_CLIENTE)
    WHERE
        (CLIENTE.ID_GRUPO_CLIENTE = :CODIGOGRUPO)
    ORDER BY
        CLIENTE.ORDEN_VISITA

    INTO
        :AI_CLIENTE,
        :ID_GRUPO_CLIENTE,
        :DESCRIPCION,
        :NOMBRE,

```

```

        :APELLIDO,
        :TELEFONO,
        :CELULAR,
        :RUC,
        :DIRECCION,
        :MEMO,
        :ORDEN_VISITA
DO
BEGIN

    RAZON_SOCIAL = NOMBRE || ', ' || APELLIDO;
FOR
SELECT
    SUM(CUENTA_DE_CLINETE_POR_GRUPO_DE.SALDO) AS DEUDAALAFECHA,
    MAX(CUENTA_DE_CLINETE_POR_GRUPO_DE.CANTIDADDECUOTASVENCIDA) AS
CANTIDADDECUOTASVENCIDA,
    MAX(CUENTA_DE_CLINETE_POR_GRUPO_DE.DIASVENCIDOS) AS
CANTIDADDEDIASVENCIDOHASTACOBRO,
    CUENTA_DE_CLINETE_POR_GRUPO_DE.DESCRIPCION,
    CUENTA_DE_CLINETE_POR_GRUPO_DE.TOTALDEUDA,
    CUENTA_DE_CLINETE_POR_GRUPO_DE.TOTALPAGOS,
    CUENTA_DE_CLINETE_POR_GRUPO_DE.TOTALSALDO
FROM
    CUENTA_DE_CLINETE_POR_GRUPO_DE(:AI_CLIENTE, :FECHADECOBRO)
GROUP BY
    CUENTA_DE_CLINETE_POR_GRUPO_DE.DESCRIPCION,
    CUENTA_DE_CLINETE_POR_GRUPO_DE.TOTALDEUDA,
    CUENTA_DE_CLINETE_POR_GRUPO_DE.TOTALPAGOS,
    CUENTA_DE_CLINETE_POR_GRUPO_DE.TOTALSALDO
INTO
    :DEUDAALAFECHA,
    :CANTIDADDECUOTASVENCIDA,
    :CANTIDADDEDIASVENCIDOHASTACOBRO,
    :DESCRIPCIONCUENTA,
    :TOTALDEUDA,
    :TOTALPAGOS,
    :TOTALSALDO
do
begin

    if (TOTALSALDO > 0) then
    BEGIN
        SUSPEND;
    END
    end

    END
END^

SET TERM ; ^

/* Definition for the `DETALLEDEVENTA` procedure : */

SET TERM ^ ;

```

```

CREATE OR ALTER PROCEDURE DETALLEDEVENTA(
    ID_VENTA INTEGER)
RETURNS (
    ID_CUENTA_CLIENTE INTEGER,
    ID_CLIENTE INTEGER,
    ID_COMPROBANTE INTEGER,
    NRÖ VARCHAR(35),
    FECHA DATE,
    DESCRIPCION VARCHAR(35),
    CREDITO DOUBLE PRECISION,
    DEBITO DOUBLE PRECISION,
    VENCIMIENTO DATE,
    DE_HASTA VARCHAR(15),
    SALDOLINEAL DOUBLE PRECISION,
    NRODERECIBO INTEGER,
    RAZONCLIENTE VARCHAR(120))
AS
BEGIN

    SALDOLINEAL = 0;
    FOR
    SELECT
        CUENTA_CLIENTE.ID_CUENTA_CLIENTE,
        CUENTA_CLIENTE.ID_CLIENTE,
        CUENTA_CLIENTE.ID_COMPROBANTE,
        CUENTA_CLIENTE.NRÖ,
        CUENTA_CLIENTE.FECHA,
        CUENTA_CLIENTE.DESCRIPCION,
        CUENTA_CLIENTE.CREDITO,
        CUENTA_CLIENTE.DEBITO,
        CUENTA_CLIENTE.VENCIMIENTO,
        CUENTA_CLIENTE.DE_HASTA,
        CUENTA_CLIENTE.ID_VENTA,
        CLIENTE.NOMBRE || ', ' || CLIENTE.APELLIDO AS RAZONCLIENTE
    FROM
        CUENTA_CLIENTE
    INNER JOIN CLIENTE ON (CUENTA_CLIENTE.ID_CLIENTE=CLIENTE.AI_CLIENTE)
    WHERE
        (CUENTA_CLIENTE.ID_VENTA = :ID_VENTA)
    ORDER BY
        CUENTA_CLIENTE.ID_CUENTA_CLIENTE
    INTO
        :ID_CUENTA_CLIENTE,
        :ID_CLIENTE,
        :ID_COMPROBANTE,
        :NRÖ,
        :FECHA,
        :DESCRIPCION,
        :CREDITO,
        :DEBITO,
        :VENCIMIENTO,
        :DE_HASTA,
        :ID_VENTA,

```

```

        :RAZONCLIENTE
DO
BEGIN

SALDOLINEAL = SALDOLINEAL + (DEBITO - CREDITO);

if (DE_HASTA IS NULL) then
BEGIN
    for
        SELECT DISTINCT
        RECIBO_CLIENTE_DETALLE.NRORECIBO
        FROM
        RECIBO_CLIENTE_DETALLE
        INNER JOIN RECIBO_CLIENTE ON
(RECIBO_CLIENTE_DETALLE.ID_COBRANZA=RECIBO_CLIENTE.ID_COBRANZA)
        WHERE
            (RECIBO_CLIENTE.ID_COBRANZA = :NRO) AND
            (RECIBO_CLIENTE_DETALLE.ID_CLIENTE = :ID_CLIENTE)
        INTO
        :NRODERECIBO
    do
        begin
        end

END

        SUSPEND;
END

END^

SET TERM ; ^

/* Definition for the `ID_CUENTA_CLIENTE_GEN` generator : */
CREATE GENERATOR ID_CUENTA_CLIENTE_GEN;

SET GENERATOR ID_CUENTA_CLIENTE_GEN TO 21;

/* Definition for the `DISPARADORCUENTACLIENTE` trigger : */

SET TERM ^ ;

CREATE TRIGGER DISPARADORCUENTACLIENTE FOR CUENTA_CLIENTE
ACTIVE BEFORE
INSERT
POSITION 0
AS

```



```

BEGIN
    NEW.ID_CUENTA_CLIENTE = GEN_ID(ID_CUENTA_CLIENTE_GEN, 1);
    POST_EVENT 'NUEVOCUENTA';
END^

SET TERM ; ^

/* Definition for the `ID_VENTA_GEN` generator : */

CREATE GENERATOR ID_VENTA_GEN;

SET GENERATOR ID_VENTA_GEN TO 3;

/* Structure for the `VENTA` table : */

CREATE TABLE VENTA (
    ID_VENTA ENTERO NOT NULL,
    ID_VENDEDOR ENTERO,
    ID_CLIENTE ENTERO,
    FECHA "DATE",
    METODOPAGO DESCRIPCION_MEDIO,
    ANULADO SI_NO_NO NOT NULL,
    ID_FORMA_PAGO ENTERO,
    NRO_PAGARE ENTERO,
    COMENTARIO DESCRIPTION_LARGE,
    ID_USUARIO ENTERO);

ALTER TABLE VENTA ADD CONSTRAINT PK_VENTA PRIMARY KEY (ID_VENTA);

/* Definition for the `DISPARADORVENTA` trigger : */

SET TERM ^ ;

CREATE TRIGGER DISPARADORVENTA FOR VENTA
ACTIVE BEFORE
INSERT
POSITION 0
AS
BEGIN
    NEW.ID_VENTA = GEN_ID(ID_VENTA_GEN, 1);
END^

SET TERM ; ^

/* Definition for the `EXTRACTO_CLIENTE_DEUDA` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE EXTRACTO_CLIENTE_DEUDA(
    CODIGOCLIENTE INTEGER)
RETURNS (
    ID_CUENTA_CLIENTE INTEGER,
    NRO VARCHAR(35),

```

```

    FECHA DATE,
    DESCRIPCION VARCHAR(35),
    MONTO DOUBLE PRECISION,
    DE_HASTA VARCHAR(15),
    VENCI DATE)
AS
BEGIN
    FOR
        SELECT
            CUENTA_CLIENTE.ID_CUENTA_CLIENTE,
            CUENTA_CLIENTE.NRO,
            CUENTA_CLIENTE.fecha,
            CUENTA_CLIENTE.DESCRIPCION,
            CUENTA_CLIENTE.DEBITO,
            CUENTA_CLIENTE.DE_HASTA,
            CUENTA_CLIENTE.vencimiento
        FROM
            CUENTA_CLIENTE
        WHERE
            (CUENTA_CLIENTE.ID_CLIENTE = :CODIGOCLIENTE) AND
            (CUENTA_CLIENTE.REFERENCIA IS NULL)
        ORDER BY
            CUENTA_CLIENTE.ID_CUENTA_CLIENTE

        INTO
            :ID_CUENTA_CLIENTE,
            :NRO,
            :FECHA,
            :DESCRIPCION,
            :MONTO,
            :DE_HASTA,
            :VENCI
    DO
        BEGIN

            SUSPEND;
        END
    END^

    SET TERM ; ^

    /* Definition for the `EXTRACTO_CLIENTE_PAGO` procedure : */

    SET TERM ^ ;

    CREATE OR ALTER PROCEDURE EXTRACTO_CLIENTE_PAGO (
        CODIGOCLIENTE INTEGER)
    RETURNS (
        ID_CUENTA_CLIENTE INTEGER,
        NRORECIBO VARCHAR(35),
        FECHA DATE,
        DESCRIPCION VARCHAR(35),
        MONTO DOUBLE PRECISION,

```

```

    DE_HASTA VARCHAR(15))
AS
declare variable Nota VARCHAR(80);
BEGIN
Nota = '%Nota de c%';
FOR
    SELECT
        RECIBO_CLIENTE.ID_COBRANZA,
        RECIBO_CLIENTE_DETALLE.NRORECIBO,
        RECIBO_CLIENTE.FECHA,
        RECIBO_CLIENTE_DETALLE.MONTO
    FROM
        RECIBO_CLIENTE_DETALLE
        INNER JOIN RECIBO_CLIENTE ON
(RECIBO_CLIENTE_DETALLE.ID_COBRANZA=RECIBO_CLIENTE.ID_COBRANZA)
    WHERE
        (RECIBO_CLIENTE_DETALLE.ID_CLIENTE = :CODIGOCLIENTE)
    ORDER BY
        RECIBO_CLIENTE.ID_COBRANZA

    INTO
        :ID_CUENTA_CLIENTE,
        :NRORECIBO,
        :FECHA,
        :MONTO
DO
BEGIN
DESCRIPCION = 'Pago';
DE_HASTA     = '1';
    SUSPEND;
END
FOR
    SELECT
        CUENTA_CLIENTE.ID_CUENTA_CLIENTE,
        CUENTA_CLIENTE.NRO,
        CUENTA_CLIENTE.FECHA,
        CUENTA_CLIENTE.DESCRIPCION,
        CUENTA_CLIENTE.CREDITO,
        CUENTA_CLIENTE.DE_HASTA
    FROM
        CUENTA_CLIENTE
    WHERE
        (CUENTA_CLIENTE.ID_CLIENTE = :CODIGOCLIENTE)
        AND
        (CUENTA_CLIENTE.DESCRIPCION LIKE :Nota)
    ORDER BY
        CUENTA_CLIENTE.ID_CUENTA_CLIENTE
    INTO
        :ID_CUENTA_CLIENTE,
        :NRORECIBO,
        :FECHA,
        :DESCRIPCION,
        :MONTO,
        :DE_HASTA

```

```

DO
BEGIN
    SUSPEND;
END
END^

SET TERM ; ^

/* Definition for the `EXTRACTO_CLIENTE` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE EXTRACTO_CLIENTE(
    CODIGOCLIENTE INTEGER)
RETURNS (
    ID_CUENTA_CLIENTE INTEGER,
    NRO VARCHAR(35),
    FECHA DATE,
    DESCRIPCION VARCHAR(35),
    MONTO DOUBLE PRECISION,
    DE_HASTA VARCHAR(15),
    NATURALEZA INTEGER,
    CREDITO DOUBLE PRECISION,
    DEBITO DOUBLE PRECISION,
    VENCI DATE)
AS
BEGIN
    FOR
        SELECT
            EXTRACTO_CLIENTE_DEUDA.ID_CUENTA_CLIENTE,
            EXTRACTO_CLIENTE_DEUDA.NRO,
            EXTRACTO_CLIENTE_DEUDA.FECHA,
            EXTRACTO_CLIENTE_DEUDA.DESCRIPCION,
            EXTRACTO_CLIENTE_DEUDA.MONTO,
            EXTRACTO_CLIENTE_DEUDA.DE_HASTA,
            EXTRACTO_CLIENTE_DEUDA.venci
        FROM
            EXTRACTO_CLIENTE_DEUDA(:CODIGOCLIENTE)

        INTO
            :ID_CUENTA_CLIENTE,
            :NRO,
            :FECHA,
            :DESCRIPCION,
            :MONTO,
            :DE_HASTA,
            :VENCI
    DO
    BEGIN
        if (DESCRIPCION = 'Entrega') then
        begin
            NATURALEZA = 0;
            CREDITO = MONTO;
            DEBITO = MONTO;

```

```

end
else
begin
NATURALEZA = 1;
DEBITO = MONTO;
CREDITO = 0;
end

        SUSPEND;
END

FOR
SELECT
EXTRACTO_CLIENTE_PAGO.ID_CUENTA_CLIENTE,
EXTRACTO_CLIENTE_PAGO.NRORECIBO,
EXTRACTO_CLIENTE_PAGO.FECHA,
EXTRACTO_CLIENTE_PAGO.DESCRIPCION,
EXTRACTO_CLIENTE_PAGO.MONTO,
EXTRACTO_CLIENTE_PAGO.DE_HASTA
FROM
EXTRACTO_CLIENTE_PAGO(:CODIGOCLIENTE)
INTO
        :ID_CUENTA_CLIENTE,
        :NRO,
        :FECHA,
        :DESCRIPCION,
        :MONTO,
        :DE_HASTA

DO
BEGIN
NATURALEZA = 2;
CREDITO = MONTO;
DEBITO = 0;
VENCI = NULL;

SUSPEND;
END

END^

SET TERM ; ^

/* Definition for the `EXTRACTO_GENERAL` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE EXTRACTO_GENERAL(
        CODIGOCLIENTE INTEGER)
RETURNS (
        ID_CUENTA_CLIENTE INTEGER,
        NRO VARCHAR(35),
        FECHA DATE,
        DESCRIPCION VARCHAR(35),

```

```

    MONTO DOUBLE PRECISION,
    DE_HASTA VARCHAR(15),
    NATURALEZA INTEGER,
    SALDOLINEAL DOUBLE PRECISION,
    ID_CLIENTE INTEGER,
    DEBITO DOUBLE PRECISION,
    CREDITO DOUBLE PRECISION,
    VENCI DATE)
AS
BEGIN
    SALDOLINEAL = 0;
    ID_CLIENTE = CODIGOCLIENTE;

    FOR
        SELECT
            EXTRACTO_CLIENTE.ID_CUENTA_CLIENTE,
            EXTRACTO_CLIENTE.NRO,
            EXTRACTO_CLIENTE.FECHA,
            EXTRACTO_CLIENTE.DESCRIPCION,
            EXTRACTO_CLIENTE.MONTO,
            EXTRACTO_CLIENTE.DE_HASTA,
            EXTRACTO_CLIENTE.NATURALEZA,
            EXTRACTO_CLIENTE.debito,
            EXTRACTO_CLIENTE.credito,
            EXTRACTO_CLIENTE.venci
        FROM
            EXTRACTO_CLIENTE(:CODIGOCLIENTE)
        ORDER BY EXTRACTO_CLIENTE.fecha

    INTO
        :ID_CUENTA_CLIENTE,
        :NRO,
        :FECHA,
        :DESCRIPCION,
        :MONTO,
        :DE_HASTA,
        :NATURALEZA,
        :DEBITO,
        :CREDITO,
        :VENCI

    DO
    BEGIN
        if (NATURALEZA = 0) then
        BEGIN
            SALDOLINEAL = SALDOLINEAL;
        END
        if (NATURALEZA = 1) then
        BEGIN
            SALDOLINEAL = SALDOLINEAL + MONTO;
        END
        if (NATURALEZA = 2) then
        BEGIN
            SALDOLINEAL = SALDOLINEAL - MONTO;
        END
    END

```

```

END

        SUSPEND;
END
END^

SET TERM ; ^

/* Structure for the `VENTA_FICHAS` table : */

CREATE TABLE VENTA_FICHAS (
    ID_VENTA_FICHAS ENTERO NOT NULL,
    ID_VENTA1 ENTERO,
    ID_VENTA2 ENTERO,
    ID_VENTA3 ENTERO,
    ID_VENTA4 ENTERO,
    FECHA "DATE",
    ACTIVA SI_NO_NO NOT NULL);

ALTER TABLE VENTA_FICHAS ADD CONSTRAINT PK_VENTA_FICHAS PRIMARY KEY
(ID_VENTA_FICHAS);

/* Definition for the `IMPRESIONFICHAS` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE IMPRESIONFICHAS(
    IDFICHA INTEGER)
RETURNS (
    ID_VENTA_FICHAS INTEGER,
    ID_VENTA1 INTEGER,
    ID_VENTA2 INTEGER,
    ID_VENTA3 INTEGER,
    ID_VENTA4 INTEGER,
    ID_CLIENTE1 INTEGER,
    NOMBREVENDEDOR1 VARCHAR(35),
    FECHA1 DATE,
    PRODUCTO1 VARCHAR(80),
    FORMAPAGO1 VARCHAR(50),
    TOTALVENTA1 DOUBLE PRECISION,
    TELEFONO1 VARCHAR(35),
    DIRECCION1 VARCHAR(80),
    RUC1 VARCHAR(35),
    BARRIO1 VARCHAR(35),
    CELULAR1 VARCHAR(35),
    NOMBRECLIENTE1 VARCHAR(35),
    APELLIDOCLIENTE1 VARCHAR(80),
    CIUDAD1 VARCHAR(35),
    ENTREGA1 DOUBLE PRECISION,
    ID_CLIENTE2 INTEGER,
    NOMBREVENDEDOR2 VARCHAR(35),
    FECHA2 DATE,
    PRODUCTO2 VARCHAR(80),

```

```

FORMAPAGO2 VARCHAR(50),
TOTALVENTA2 DOUBLE PRECISION,
TELEFONO2 VARCHAR(35),
DIRECCION2 VARCHAR(80),
RUC2 VARCHAR(35),
BARRIO2 VARCHAR(35),
CELULAR2 VARCHAR(35),
NOMBRECLIENTE2 VARCHAR(35),
APELLIDOCIENTE2 VARCHAR(80),
CIUDAD2 VARCHAR(35),
ENTREGA2 DOUBLE PRECISION,
ID_CLIENTE3 INTEGER,
NOMBREVENDEDOR3 VARCHAR(35),
FECHA3 DATE,
PRODUCTO3 VARCHAR(80),
FORMAPAGO3 VARCHAR(50),
TOTALVENTA3 DOUBLE PRECISION,
TELEFONO3 VARCHAR(35),
DIRECCION3 VARCHAR(80),
RUC3 VARCHAR(35),
BARRIO3 VARCHAR(35),
CELULAR3 VARCHAR(35),
NOMBRECLIENTE3 VARCHAR(35),
APELLIDOCIENTE3 VARCHAR(80),
CIUDAD3 VARCHAR(35),
ENTREGA3 DOUBLE PRECISION,
ID_CLIENTE4 INTEGER,
NOMBREVENDEDOR4 VARCHAR(35),
FECHA4 DATE,
PRODUCTO4 VARCHAR(80),
FORMAPAGO4 VARCHAR(50),
TOTALVENTA4 DOUBLE PRECISION,
TELEFONO4 VARCHAR(35),
DIRECCION4 VARCHAR(80),
RUC4 VARCHAR(35),
BARRIO4 VARCHAR(35),
CELULAR4 VARCHAR(35),
NOMBRECLIENTE4 VARCHAR(35),
APELLIDOCIENTE4 VARCHAR(80),
CIUDAD4 VARCHAR(35),
ENTREGA4 DOUBLE PRECISION)
AS
BEGIN

SELECT
    cliente.ai_cliente,
    VENTA_FICHAS.ID_VENTA_FICHAS,
    VENTA_FICHAS.ID_VENTA1,
    VENTA_FICHAS.ID_VENTA2,
    VENTA_FICHAS.ID_VENTA3,
    VENTA_FICHAS.ID_VENTA4,
    VENDEDOR.NOMBRE AS NOMBREVENDEDOR1,
    VENTA.FECHA AS FECHA1,
    venta.comentario AS PRODUCTO1,

```



```

VENTA.METODOPAGO AS FORMAPAGO1,
CLIENTE.TELEFONO AS TELEFONO1,
CLIENTE.DIRECCION AS DIRECCION1,
CLIENTE.RUC AS RUC1,
CLIENTE.BARRIO AS BARRIO1,
CLIENTE.CELULAR AS CELULAR1,
CLIENTE.NOMBRE AS NOMBRECLIENTE1,
CLIENTE.APELLIDO AS APELLIDOCIENTE1,
CIUDAD.NOMBRE as ciudad1
FROM
VENTA_FICHAS
INNER JOIN VENTA ON (VENTA_FICHAS.ID_VENTA1=VENTA.ID_VENTA)
INNER JOIN VENDEDOR ON (VENTA.ID_VENDEDOR=VENDEDOR.ID_VENDEDOR)
INNER JOIN CLIENTE ON (VENTA.ID_CLIENTE=CLIENTE.AI_CLIENTE)
INNER JOIN CIUDAD ON (CLIENTE.ID_CIUADAD=CIUDAD.ID_CIUADAD)
WHERE
(VENTA_FICHAS.ID_VENTA_FICHAS = :IDFICHA)
    INTO
        :id_cliente1,
        :ID_VENTA_FICHAS,
        :ID_VENTA1,
        :ID_VENTA2,
        :ID_VENTA3,
        :ID_VENTA4,
        :NOMBREVENDEDOR1,
        :FECHA1,
        :PRODUCTO1,
        :FORMAPAGO1,
        :TELEFONO1,
        :DIRECCION1,
        :RUC1,
        :BARRIO1,
        :CELULAR1,
        :NOMBRECLIENTE1,
        :APELLIDOCIENTE1,
        :CIUDAD1;

SELECT
    SUM(VENTAD.CANTIDAD * VENTAD.PRECIO) AS TOTALVENTA1
FROM
    VENTAD
WHERE
    (VENTAD.ID_VENTA = :ID_VENTA1)
    INTO
        :TOTALVENTA1;
    if (TOTALVENTA1 IS NULL) then
    BEGIN
        TOTALVENTA1 = 0;
    END

SELECT
    CUENTA_CLIENTE.CREDITO
FROM

```

```

CUENTA_CLIENTE
WHERE
  (CUENTA_CLIENTE.ID_VENTA = :ID_VENTA1) AND
  (CUENTA_CLIENTE.DE_HASTA = 'Pago')
into
:entrega1;
if (entrega1 is null) then
begin
entrega1 = 0;
end

SELECT
  CLIENTE.ai_cliente,
  VENDEDOR.NOMBRE AS NOMBREVENDEDOR2,
  VENTA.FECHA AS FECHA1,
  venta.comentario AS PRODUCTO2,
  VENTA.METODOPAGO AS FORMAPAGO2,
  CLIENTE.TELEFONO AS TELEFONO2,
  CLIENTE.DIRECCION AS DIRECCION2,
  CLIENTE.RUC AS RUC2,
  CLIENTE.BARRIO AS BARRIO2,
  CLIENTE.CELULAR AS CELULAR2,
  CLIENTE.NOMBRE AS NOMBRECLIENTE2,
  CLIENTE.APELLIDO AS APELLIDOCIENTE2,
  CIUDAD.NOMBRE AS CIUDAD2
FROM
  VENTA
  INNER JOIN VENDEDOR ON (VENTA.ID_VENDEDOR=VENDEDOR.ID_VENDEDOR)
  INNER JOIN CLIENTE ON (VENTA.ID_CLIENTE=CLIENTE.AI_CLIENTE)
  INNER JOIN VENTA_FICHAS ON (VENTA_FICHAS.ID_VENTA2=VENTA.ID_VENTA)
  INNER JOIN CIUDAD ON (CLIENTE.ID_CIUDAD=CIUDAD.ID_CIUDAD)
WHERE
  (VENTA_FICHAS.ID_VENTA_FICHAS = :IDFICHA)
  INTO
  :ID_CLIENTE2,
  :NOMBREVENDEDOR2,
  :FECHA2,
  :PRODUCTO2,
  :FORMAPAGO2,
  :TELEFONO2,
  :DIRECCION2,
  :RUC2,
  :BARRIO2,
  :CELULAR2,
  :NOMBRECLIENTE2,
  :APELLIDOCIENTE2,
  :CIUDAD2;

SELECT
  SUM(VENTAD.CANTIDAD * VENTAD.PRECIO) AS TOTALVENTA2
FROM
  VENTAD
WHERE

```

```

(VENTAD.ID_VENTA = :ID_VENTA2)
INTO
:TOTALVENTA2;
if (TOTALVENTA2 IS NULL) then
BEGIN
TOTALVENTA2 = 0;
END

```

```

SELECT
CUENTA_CLIENTE.CREDITO
FROM
CUENTA_CLIENTE
WHERE
(CUENTA_CLIENTE.ID_VENTA = :ID_VENTA2) AND
(CUENTA_CLIENTE.DE_HASTA = 'Pago')
into
:entrega2;
if (entrega2 is null) then
begin
entrega2 = 0;
end

```

```

SELECT
cliente.ai_cliente,
VENDEDOR.NOMBRE AS NOMBREVENDEDOR3,
VENTA.FECHA AS FECHA3,
venta.comentario AS PRODUCTO3,
VENTA.METODOPAGO AS FORMAPAGO3,
CLIENTE.TELEFONO AS TELEFONO3,
CLIENTE.DIRECCION AS DIRECCION3,
CLIENTE.RUC AS RUC3,
CLIENTE.BARRIO AS BARRIO3,
CLIENTE.CELULAR AS CELULAR3,
CLIENTE.NOMBRE AS NOMBRECLIENTE3,
CLIENTE.APELLIDO AS APELLIDOCIENTE3,
CIUDAD.NOMBRE
FROM
VENTA
INNER JOIN VENDEDOR ON (VENTA.ID_VENDEDOR=VENDEDOR.ID_VENDEDOR)
INNER JOIN CLIENTE ON (VENTA.ID_CLIENTE=CLIENTE.AI_CLIENTE)
INNER JOIN VENTA_FICHAS ON (VENTA_FICHAS.ID_VENTA3=VENTA.ID_VENTA)
INNER JOIN CIUDAD ON (CLIENTE.ID_CIUDAD=CIUDAD.ID_CIUDAD)
WHERE
(VENTA_FICHAS.ID_VENTA_FICHAS = :IDFICHA)
INTO
:ID_CLIENTE3,
:NOMBREVENDEDOR3,
:FECHA3,
:PRODUCTO3,
:FORMAPAGO3,
:TELEFONO3,

```

```
:DIRECCION3,  
:RUC3,  
:BARRIO3,  
:CELULAR3,  
:NOMBRECLIENTE3,  
:APELLIDOCIENTE3,  
:CIUDAD3;
```

```
SELECT  
    SUM(VENTAD.CANTIDAD * VENTAD.PRECIO) AS TOTALVENTA3  
FROM  
    VENTAD  
WHERE  
    (VENTAD.ID_VENTA = :ID_VENTA3)  
    INTO  
    :TOTALVENTA3;  
if (TOTALVENTA3 IS NULL) then  
BEGIN  
TOTALVENTA3 = 0;  
END
```

```
SELECT  
    CUENTA_CLIENTE.CREDITO  
FROM  
    CUENTA_CLIENTE  
WHERE  
    (CUENTA_CLIENTE.ID_VENTA = :ID_VENTA3) AND  
    (CUENTA_CLIENTE.DE_HASTA = 'Pago')  
    into  
    :entrega3;  
if (entrega3 is null) then  
begin  
entrega3 = 0;  
end
```

```
SELECT  
    cliente.ai_cliente,  
    VENDEDOR.NOMBRE AS NOMBREVENDEDOR4,  
    VENTA.FECHA AS FECHA4,  
    VENTA.comentario AS PRODUCTO4,  
    VENTA.METODOPAGO AS FORMAPAGO4,  
    CLIENTE.TELEFONO AS TELEFONO4,  
    CLIENTE.DIRECCION AS DIRECCION4,  
    CLIENTE.RUC AS RUC4,  
    CLIENTE.BARRIO AS BARRIO4,  
    CLIENTE.CELULAR AS CELULAR4,  
    CLIENTE.NOMBRE AS NOMBRECLIENTE4,  
    CLIENTE.APELLIDO AS APELLIDOCIENTE4,  
    CIUDAD.NOMBRE  
FROM  
    VENTA
```

```

INNER JOIN VENDEDOR ON (VENTA.ID_VENDEDOR=VENDEDOR.ID_VENDEDOR)
INNER JOIN CLIENTE ON (VENTA.ID_CLIENTE=CLIENTE.AI_CLIENTE)
INNER JOIN VENTA_FICHAS ON (VENTA_FICHAS.ID_VENTA4=VENTA.ID_VENTA)
INNER JOIN CIUDAD ON (CLIENTE.ID_CIUDAD=CIUDAD.ID_CIUDAD)
WHERE
    (VENTA_FICHAS.ID_VENTA_FICHAS = :IDFICHA)

    INTO
    :ID_CLIENTE4,
    :NOMBREVENDEDOR4,
    :FECHA4,
    :PRODUCTO4,
    :FORMAPAGO4,
    :TELEFONO4,
    :DIRECCION4,
    :RUC4,
    :BARRIO4,
    :CELULAR4,
    :NOMBRECLIENTE4,
    :APELLIDOCIENTE4,
    :CIUDAD4;

SELECT
    SUM(VENTAD.CANTIDAD * VENTAD.PRECIO) AS TOTALVENTA4
FROM
    VENTAD
WHERE
    (VENTAD.ID_VENTA = :ID_VENTA4)
    INTO
    :TOTALVENTA4;
    if (TOTALVENTA4 IS NULL) then
    BEGIN
    TOTALVENTA4 = 0;
    END

    SELECT
    CUENTA_CLIENTE.CREDITO
FROM
    CUENTA_CLIENTE
WHERE
    (CUENTA_CLIENTE.ID_VENTA = :ID_VENTA4) AND
    (CUENTA_CLIENTE.DE_HASTA = 'Pago')
    into
    :entrega4;
    if (entrega4 is null) then
    begin
    entrega4 = 0;
    end

    SUSPEND;

END^

```

```

SET TERM ; ^

/* Definition for the `MEMO_CORTO` domain : */

CREATE DOMAIN MEMO_CORTO AS
    VARCHAR(250);

/* Structure for the `EMAIL_PENDIENTES` table : */

CREATE TABLE EMAIL_PENDIENTES (
    ID_EMAIL_PENDIENTES ENTERO NOT NULL,
    EMAIL DESCRIPTION_LARGE,
    ASUNTO DESCRIPTION_LARGE,
    ADJUNTO MEMO_CORTO,
    CONTENIDO MEMO,
    ENVIADO YES_NO);

ALTER TABLE EMAIL_PENDIENTES ADD CONSTRAINT PK_EMAIL_PENDIENTES PRIMARY
KEY (ID_EMAIL_PENDIENTES);

/* Definition for the `ID_EMAIL_PENDIENTES_GEN` generator : */

CREATE GENERATOR ID_EMAIL_PENDIENTES_GEN;

SET GENERATOR ID_EMAIL_PENDIENTES_GEN TO 167;

/* Definition for the `INCREMENTADOREMAILPENDIENTE` trigger : */

SET TERM ^ ;

CREATE TRIGGER INCREMENTADOREMAILPENDIENTE FOR EMAIL_PENDIENTES
ACTIVE BEFORE
INSERT
POSITION 0
AS
BEGIN
    NEW.ID_EMAIL_PENDIENTES = GEN_ID(ID_EMAIL_PENDIENTES_GEN, 1);
END^

SET TERM ; ^

/* Structure for the `MOVIMIENTO` table : */

CREATE TABLE MOVIMIENTO (
    ID_MOVIMIENTO ENTERO NOT NULL,
    ID_VENDEDOR ENTERO,
    FECHA FECHA,
    ENTREGADO SI_NO_NO NOT NULL,
    ENVIADO SI_NO_NO NOT NULL);

/* Structure for the `MOVIMIENTO_DETALLE` table : */

CREATE TABLE MOVIMIENTO_DETALLE (

```

```

ID_MOVIMIENTO_DETALLE ENTERO NOT NULL,
ID_MOVIMIENTO ENTERO NOT NULL,
ID_PRODUCTO ENTERO,
CANTIDAD ENTERO,
PRECIO1 NUMERO,
PRECIO2 NUMERO,
PRECIO3 NUMERO,
PESO NUMERO,
DEVUELTO SI_NO_NO NOT NULL,
CODIGO_BARRAS NOMBRE_CORTO,
CANT_DEVUELTO ENTERO);

```

```

ALTER TABLE MOVIMIENTO_DETALLE ADD CONSTRAINT PK_MOVIMIENTO_DETALLE
PRIMARY KEY (ID_MOVIMIENTO_DETALLE);

```

```

/* Definition for the `LISTADEMALETINES` procedure : */

```

```

SET TERM ^ ;

```

```

CREATE OR ALTER PROCEDURE LISTADEMALETINES(
    DESDE DATE,
    HASTA DATE)
RETURNS (
    ID_MOVIMIENTO INTEGER,
    ID_VENDEDOR SMALLINT,
    FECHA DATE,
    ENVIADO CHAR(1),
    ENTREGADO CHAR(1),
    INFOVENDEDOR VARCHAR(110),
    CANTIDADENVIADO DOUBLE PRECISION,
    CANTIDADDEVULETO DOUBLE PRECISION,
    CANTIDADVENDIDO DOUBLE PRECISION,
    VALORPLANILLA_ENVIADO DOUBLE PRECISION,
    VALORPLANILLA_VENDIDO DOUBLE PRECISION,
    VALORPLANILLA_DEVUELTO DOUBLE PRECISION)

```

```

AS

```

```

declare variable NOMBRE VARCHAR (50) CHARACTER SET NONE;
declare variable APELLIDO VARCHAR (50) CHARACTER SET NONE;
BEGIN
CANTIDADENVIADO      = 0;
CANTIDADDEVULETO     = 0;
CANTIDADVENDIDO      = 0;

```

```

FOR

```

```

    SELECT
        MOVIMIENTO.ID_MOVIMIENTO,
        MOVIMIENTO.ID_VENDEDOR,
        VENDEDOR.NOMBRE,
        VENDEDOR.APELLIDO_PATERNO,
        MOVIMIENTO.FECHA,
        MOVIMIENTO.ENVIADO,
        MOVIMIENTO.ENTREGADO

```

```

FROM

```

```

        MOVIMIENTO
        INNER JOIN VENDEDOR ON (MOVIMIENTO.ID_VENDEDOR=VENDEDOR.ID_VENDEDOR)
WHERE
        (MOVIMIENTO.FECHA BETWEEN :DESDE AND :HASTA)
ORDER BY
        MOVIMIENTO.ID_MOVIMIENTO DESC

INTO
        :ID_MOVIMIENTO,
        :ID_VENDEDOR,
        :NOMBRE,
        :APELLIDO,
        :FECHA,
        :ENVIADO,
        :ENTREGADO
DO
BEGIN
        INFOVENDEDOR = NOMBRE || ', ' || APELLIDO;
SELECT
        SUM(MOVIMIENTO_DETALLE.CANTIDAD) AS CANTIDADENVIADO,
        SUM(MOVIMIENTO_DETALLE.CANTIDAD * MOVIMIENTO_DETALLE.PRECIO1) AS
VALORPLANILLA_ENVIADO
FROM
        MOVIMIENTO_DETALLE
WHERE
        (MOVIMIENTO_DETALLE.ID_MOVIMIENTO = :ID_MOVIMIENTO)
        INTO
        :CANTIDADENVIADO,
        :VALORPLANILLA_ENVIADO;

        if (CANTIDADENVIADO IS NULL) then
        BEGIN
                CANTIDADENVIADO          = 0;
                VALORPLANILLA_ENVIADO    = 0;
        END

SELECT
        SUM(MOVIMIENTO_DETALLE.CANT_DEVUELTO) AS CANTIDADDEVULETO,
        SUM(MOVIMIENTO_DETALLE.CANT_DEVUELTO * MOVIMIENTO_DETALLE.PRECIO1) AS
FIELD_1
FROM
        MOVIMIENTO_DETALLE
WHERE
        (MOVIMIENTO_DETALLE.ID_MOVIMIENTO = :ID_MOVIMIENTO)
        INTO
        :CANTIDADDEVULETO,
        :VALORPLANILLA_DEVUELTO;

        if (CANTIDADDEVULETO IS NULL) then
        BEGIN
                CANTIDADDEVULETO          = 0;
                VALORPLANILLA_DEVUELTO    = 0;

```



```

END

CANTIDADVENDIDO          = CANTIDADENVIADO - CANTIDADDEVUELTO;
VALORPLANILLA_VENDIDO    = VALORPLANILLA_ENVIADO -
VALORPLANILLA_DEVUELTO;
    SUSPEND;
END
END^

SET TERM ; ^

/* Definition for the `LISTADEMOROSOSD` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE LISTADEMOROSOSD(
    CODIGOCLIENTE INTEGER)
RETURNS (
    ID_CUENTA_CLIENTE INTEGER,
    FECHA DATE,
    DESCRIPCION VARCHAR(35),
    DEBITO DOUBLE PRECISION,
    VENCIMIENTO DATE,
    PAGOS DOUBLE PRECISION,
    CANCELADO CHAR(1),
    CANTIDADVECIDO INTEGER,
    MONTOVENCIDO DOUBLE PRECISION,
    MONTOAVENCER DOUBLE PRECISION,
    NROCOMPROBANTE VARCHAR(35),
    NOMBREVENDEDOR VARCHAR(35),
    FORMADEPAGO VARCHAR(35),
    DIASVENCIDO INTEGER)
AS
BEGIN
    CANCELADO = 'N';
    CANTIDADVECIDO = 0;
    MONTOVENCIDO = 0;
    MONTOAVENCER = 0;
    DIASVENCIDO = 0;
    FOR
SELECT
    CUENTA_CLIENTE.ID_CUENTA_CLIENTE,
    CUENTA_CLIENTE.FECHA,
    CUENTA_CLIENTE.DESCRIPCION,
    CUENTA_CLIENTE.DEBITO,
    CUENTA_CLIENTE.VENCIMIENTO,
    CUENTA_CLIENTE.NRO,
    VENDEDOR.NOMBRE,
    FORMA_PAGO.DESCRIPCION
FROM
    CUENTA_CLIENTE
    INNER JOIN VENDEDOR ON (CUENTA_CLIENTE.ID_PERSONAL=VENDEDOR.ID_VENDEDOR)
    INNER JOIN FORMA_PAGO ON
(CUENTA_CLIENTE.ID_FORMA_PAGO=FORMA_PAGO.ID_FORMA_PAGO)

```

```

WHERE
    (CUENTA_CLIENTE.ID_CLIENTE = :CODIGOCLIENTE) AND
    (CUENTA_CLIENTE.REFERENCIA IS NULL)
ORDER BY
    CUENTA_CLIENTE.ID_CUENTA_CLIENTE

    INTO
        :ID_CUENTA_CLIENTE,
        :FECHA,
        :DESCRIPCION,
        :DEBITO,
        :VENCIMIENTO,
        :nrocomprobante,
        :NOMBREVENDEDOR,
        :FORMADEPAGO

DO
BEGIN

    SELECT
        SUM(CUENTA_CLIENTE.CREDITO) AS SUMAPAGOS
FROM
    CUENTA_CLIENTE
WHERE
    (CUENTA_CLIENTE.REFERENCIA = :ID_CUENTA_CLIENTE)
    INTO
        :PAGOS;
    if (PAGOS IS NULL) then
    BEGIN
        PAGOS = 0;
    END

    if (PAGOS >= DEBITO) then
    BEGIN
        CANCELADO = 'S';
    END
    ELSE
    BEGIN
        if (VENCIMIENTO < current_date) then
        begin
            CANTIDADVECIDO = CANTIDADVECIDO + 1;
            MONTOVENCIDO = (MONTOVENCIDO + (DEBITO - PAGOS));
            DIASVENCIDO = current_date - VENCIMIENTO;
        end
        MONTOAVENCER = MONTOAVENCER + (DEBITO - PAGOS);
    END

    if (debito > pagos) then
    begin
        suspend;
    end

        CANCELADO = 'N';
        CANTIDADVECIDO = 0;

```

```

    MONTOVENCIDO = 0;
    MONTOAVENCER = 0;
    END
END^

SET TERM ; ^

/* Definition for the `LISTADEMOROSOS` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE LISTADEMOROSOS (
    CODIGOCIUDAD INTEGER,
    CODIGOVENDEDOR INTEGER,
    CALI VARCHAR(35),
    CALI1 VARCHAR(35))
RETURNS (
    AI_CLIENTE INTEGER,
    DESCRIPCION VARCHAR(35),
    NOMBRE VARCHAR(35),
    RAZONSOCIAL VARCHAR(117),
    DIRECCION VARCHAR(80),
    RUC VARCHAR(35),
    BARRIO VARCHAR(35),
    CELULAR VARCHAR(35),
    LUGARTRABAJO VARCHAR(80),
    OBSERVACION VARCHAR(500),
    CALIFICACION VARCHAR(35),
    DIASVENCIDO INTEGER,
    MONTOVENCIDO DOUBLE PRECISION,
    CUOTASVENCIDA INTEGER,
    SALDOACTUAL DOUBLE PRECISION)
AS
BEGIN
    FOR
        SELECT
            CLIENTE.AI_CLIENTE,
            GRUPO_CLIENTE.DESCRIPCION,
            CIUDAD.NOMBRE,
            CLIENTE.NOMBRE || ', ' || CLIENTE.APELLIDO AS RAZONSOCIAL,
            CLIENTE.DIRECCION,
            CLIENTE.RUC,
            CLIENTE.BARRIO,
            CLIENTE.CELULAR,
            CLIENTE.LUGARTRABAJO,
            CLIENTE.OBSERVACION,
            CLIENTE.CALIFICACION
        FROM
            CLIENTE
            INNER JOIN GRUPO_CLIENTE ON
                (CLIENTE.ID_GRUPO_CLIENTE=GRUPO_CLIENTE.ID_GRUPO_CLIENTE)
            INNER JOIN CIUDAD ON (CLIENTE.ID_CIUAD=CIUDAD.ID_CIUAD)
        WHERE
            ((CLIENTE.ACTIVO = 1) AND

```

```

        (CLIENTE.ID_CIUDAD = :CODIGOCIUDAD) AND
        (CLIENTE.ID_VENDEDOR = :CODIGOVENDEDOR)) AND (
        (CLIENTE.CALIFICACION = :CALI) OR
        (CLIENTE.CALIFICACION = :CALI1))
ORDER BY
    CLIENTE.AI_CLIENTE

INTO
    :AI_CLIENTE,
    :DESCRIPCION,
    :NOMBRE,
    :RAZONSOCIAL,
    :DIRECCION,
    :RUC,
    :BARRIO,
    :CELULAR,
    :LUGARTRABAJO,
    :OBSERVACION,
    :CALIFICACION
DO
BEGIN

SELECT
MAX (LISTADEMOROSOSD.DIASVENCIDO) AS DIASVENCIDO,
SUM (LISTADEMOROSOSD.MONTOVENCIDO) AS MONTOVENCIDO,
SUM (LISTADEMOROSOSD.CANTIDADVECIDO) AS CANTIDADVECIDO,
SUM (LISTADEMOROSOSD.DEBITO - LISTADEMOROSOSD.PAGOS) AS SALDOACTUAL
FROM
LISTADEMOROSOSD (:AI_CLIENTE)
INTO
:DIASVENCIDO,
:MONTOVENCIDO,
:CUOTASVENCIDA,
:SALDOACTUAL;

SUSPEND;
END
END^

SET TERM ; ^

/* Definition for the `LISTADODECUMPLEANOS` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE LISTADODECUMPLEANOS
RETURNS (
    AI_CLIENTE INTEGER,
    RUC VARCHAR(35),
    FECHA_NACIMIENTO DATE,

```

```

    CELULAR VARCHAR(35),
    RAZONSOCIAL VARCHAR(118),
    ACTIVO INTEGER,
    EDAD INTEGER,
    ANONACIMIENTO INTEGER,
    ANOACTUAL INTEGER)
AS
BEGIN

FOR
    SELECT
        CLIENTE.AI_CLIENTE,
        CLIENTE.RUC,
        CLIENTE.FECHA_NACIMIENTO,
        CLIENTE.CELULAR,
        CLIENTE.NOMBRE || ' , ' || CLIENTE.APELLIDO AS RAZONSOCIAL,
        CLIENTE.activo
    FROM
        CLIENTE
    where extract(month from CLIENTE.FECHA_NACIMIENTO) = extract(month
from current_date) and
        extract(day from CLIENTE.FECHA_NACIMIENTO) = extract(day from
current_date)
    ORDER BY
        CLIENTE.AI_CLIENTE

    INTO
        :AI_CLIENTE,
        :RUC,
        :FECHA_NACIMIENTO,
        :CELULAR,
        :RAZONSOCIAL,
        :ACTIVO

DO
BEGIN
    if (FECHA_NACIMIENTO IS NOT NULL) then
    BEGIN
        ANONACIMIENTO = extract(year FROM FECHA_NACIMIENTO);
        ANOACTUAL      = extract(year FROM current_date);
    EDAD = ANOACTUAL - ANONACIMIENTO;
    END

    SUSPEND;
    END
END^

SET TERM ; ^

/* Definition for the `CEILING` UDF : */

DECLARE EXTERNAL FUNCTION "CEILING"

```

```

    DOUBLE PRECISION
    RETURNS DOUBLE PRECISION BY VALUE
    ENTRY_POINT 'IB_UDF_ceiling' MODULE_NAME 'ib_udf';

/* Definition for the `FLOOR` UDF : */

DECLARE EXTERNAL FUNCTION "FLOOR"
    DOUBLE PRECISION
    RETURNS DOUBLE PRECISION BY VALUE
    ENTRY_POINT 'IB_UDF_floor' MODULE_NAME 'ib_udf';

/* Definition for the `REDONDEAR` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE REDONDEAR(
    VALORAPASAR FLOAT)
    RETURNS (
        LAVERDAD FLOAT)
AS
declare variable HOLA float;
declare variable HOLA2 float;
BEGIN
    BEGIN
        HOLA = FLOOR(VALORAPASAR);
        HOLA2 = (VALORAPASAR - HOLA);
        if (HOLA2 >= 0.5) then
            BEGIN
                LAVERDAD = ceiling(VALORAPASAR);
            END
        if (HOLA2 < 0.5) then
            BEGIN
                LAVERDAD = FLOOR(VALORAPASAR);
            END
        END
    SUSPEND;
END
END^

SET TERM ; ^

/* Definition for the `LISTADODEVENTA` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE LISTADODEVENTA(
    DESDE DATE,
    HASTA DATE)
    RETURNS (
        ID_VENTA INTEGER,
        ID_VENDEDOR INTEGER,
        ID_CLIENTE INTEGER,
        FECHA DATE,
        NOMBREVENDEDOR VARCHAR(35),
        NOMBRECLIENTE VARCHAR(35),

```

```

APELLIDOCIENTE VARCHAR(80),
NOMBRECIUDAD VARCHAR(35),
TELEFONO VARCHAR(35),
RUC VARCHAR(35),
BARRIO VARCHAR(35),
CELULAR VARCHAR(35),
TELEFONO_1 VARCHAR(35),
OBSERVACION VARCHAR(500),
CALIFICACION VARCHAR(35),
GRUPOCLIENTE VARCHAR(35),
ANULADO CHAR(1),
DESCRIPCION VARCHAR(35),
NRO_PAGARE INTEGER,
COMENTARIO VARCHAR(80),
COSTODEVENTA DOUBLE PRECISION,
TOTALVENTA DOUBLE PRECISION,
PAGOSOBTENIDOS DOUBLE PRECISION,
PORCENTAJEUTILIDAD DOUBLE PRECISION,
PORCENTAJEPAGADO DOUBLE PRECISION,
SALDOACTUAL DOUBLE PRECISION,
UTILIDAD DOUBLE PRECISION,
USUARIO VARCHAR(30))
AS
BEGIN
    FOR
SELECT
    VENTA.ID_VENTA,
    VENTA.ID_VENDEDOR,
    VENTA.ID_CLIENTE,
    VENTA.FECHA,
    VENDEDOR.NOMBRE AS NOMBREVENDEDOR,
    CLIENTE.NOMBRE AS NOMBRECLIENTE,
    CLIENTE.APELLIDO AS APELLIDOCIENTE,
    CIUDAD.NOMBRE AS NOMBRECIUDAD,
    CLIENTE.TELEFONO,
    CLIENTE.RUC,
    CLIENTE.BARRIO,
    CLIENTE.CELULAR,
    CLIENTE.TELEFONO_1,
    CLIENTE.OBSERVACION,
    CLIENTE.CALIFICACION,
    GRUPO_CLIENTE.DESCRIPCION AS GRUPOCLIENTE,
    VENTA.ANULADO,
    FORMA_PAGO.DESCRIPCION,
    VENTA.NRO_PAGARE,
    VENTA.COMENTARIO,
    UCTABUSERS.UCUSERNAME
FROM
    VENTA
    INNER JOIN VENDEDOR ON (VENTA.ID_VENDEDOR=VENDEDOR.ID_VENDEDOR)
    INNER JOIN CLIENTE ON (VENTA.ID_CLIENTE=CLIENTE.AI_CLIENTE)
    INNER JOIN CIUDAD ON (CLIENTE.ID_CIUAD=CIUDAD.ID_CIUAD)
    INNER JOIN GRUPO_CLIENTE ON
(CIENTE.ID_GRUPO_CLIENTE=GRUPO_CLIENTE.ID_GRUPO_CLIENTE)

```

```

INNER JOIN FORMA_PAGO ON (VENTA.ID_FORMA_PAGO=FORMA_PAGO.ID_FORMA_PAGO)
INNER JOIN UCTABUSERS ON (VENTA.ID_USUARIO=UCTABUSERS.UCIDUSER)
WHERE
    (VENTA.FECHA BETWEEN :DESDE AND :HASTA)
ORDER BY
    VENTA.ID_VENTA DESC

```

```

    INTO
        :ID_VENTA,
        :ID_VENDEDOR,
        :ID_CLIENTE,
        :FECHA,
        :NOMBREVENDEDOR,
        :NOMBRECLIENTE,
        :APELLIDOCLIENTE,
        :NOMBRECIUDAD,
        :TELEFONO,
        :RUC,
        :BARRIO,
        :CELULAR,
        :TELEFONO_1,
        :OBSERVACION,
        :CALIFICACION,
        :GRUPOCLIENTE,
        :ANULADO,
        :DESCRIPCION,
        :NRO_PAGARE,
        :COMENTARIO,
        :USUARIO

```

```

DO
BEGIN

```

```

    SELECT
        SUM(VENTAD.COSTO) AS COSTODEVENTA
FROM
    VENTAD
WHERE
    (VENTAD.ID_VENTA = :ID_VENTA)
    INTO
        :COSTODEVENTA;
    if (COSTODEVENTA IS NULL) then
    BEGIN
        COSTODEVENTA = 0;
    END

```

```

        SELECT
            SUM(CUENTA_CLIENTE.debito) AS TOTALVENTA
FROM
    CUENTA_CLIENTE
WHERE
    (CUENTA_CLIENTE.ID_VENTA = :ID_VENTA) AND
    (CUENTA_CLIENTE.REFERENCIA IS NULL)
    INTO
        :TOTALVENTA;

```



```

if (TOTALVENTA IS NULL) then
BEGIN
TOTALVENTA = 0;
END

        SELECT
        SUM(CUENTA_CLIENTE.credito) AS PAGOSOBTENIDOS
FROM
CUENTA_CLIENTE
WHERE
(CUENTA_CLIENTE.ID_VENTA = :ID_VENTA) AND
(CUENTA_CLIENTE.REFERENCIA IS NOT NULL)
into
:PAGOSOBTENIDOS;

if (PAGOSOBTENIDOS IS NULL) then
BEGIN
PAGOSOBTENIDOS = 0;
END

UTILIDAD      = TOTALVENTA - COSTODEVENTA;

if (COSTODEVENTA > 0) then
begin
    PORCENTAJEUTILIDAD = ((UTILIDAD * 100) / COSTODEVENTA);
    SELECT
    REDONDEAR.LAVERDAD
    FROM
    REDONDEAR(:PORCENTAJEUTILIDAD)
    INTO
    :PORCENTAJEUTILIDAD;
end

if (PAGOSOBTENIDOS > 0) then
BEGIN
PORCENTAJEPAGADO = ((PAGOSOBTENIDOS * 100) / TOTALVENTA);
END
ELSE
BEGIN
PORCENTAJEPAGADO = 0;
END

SALDOACTUAL = TOTALVENTA - PAGOSOBTENIDOS;

        SUSPEND;
        UTILIDAD = 0;
        PORCENTAJEPAGADO = 0;
        SALDOACTUAL = 0;
        PORCENTAJEUTILIDAD = 0;

```

```

        COSTODEVENTA = 0;

    END
END^

SET TERM ; ^

/* Structure for the `CLIENTE_NOTA_CREDITO` table : */

CREATE TABLE CLIENTE_NOTA_CREDITO (
    ID_CLIENTE_NOTA_CREDITO ENTERO NOT NULL,
    ID_CLIENTE ENTERO,
    FECHA DATE,
    ID_USUARIO ENTERO,
    COMENTARIO MEMO,
    GUARDADO SI_NO_NO NOT NULL,
    ACREDITADO SI_NO_NO NOT NULL,
    NRO_COMPROBANTE ENTERO);

ALTER TABLE CLIENTE_NOTA_CREDITO ADD CONSTRAINT PK_CLIENTE_NOTA_CREDITO
PRIMARY KEY (ID_CLIENTE_NOTA_CREDITO);

/* Structure for the `CLIENTE_NOTA_CREDITO_D` table : */

CREATE TABLE CLIENTE_NOTA_CREDITO_D (
    ID_NOTA_CREDITO_D ENTERO NOT NULL,
    ID_CLIENTE_NOTA_CREDITO ENTERO,
    ID_PRODUCTO ENTERO,
    CANTIDAD NUMERO,
    PRECIO NUMERO,
    COSTO NUMERO);

ALTER TABLE CLIENTE_NOTA_CREDITO_D ADD CONSTRAINT
PK_CLIENTE_NOTA_CREDITO_D PRIMARY KEY (ID_NOTA_CREDITO_D);

/* Definition for the `LISTADODEVOLUCIONES` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE LISTADODEVOLUCIONES(
    DESDE DATE,
    HASTA DATE)
RETURNS (
    ID_CLIENTE_NOTA_CREDITO INTEGER,
    ID_CLIENTE INTEGER,
    RAZONSOCIAL VARCHAR(117),
    DESCRIPCION VARCHAR(35),
    NOMBRE VARCHAR(35),
    RUC VARCHAR(35),
    CELULAR VARCHAR(35),

```

```

TOTALDEVOLUCION DOUBLE PRECISION,
FECHA DATE,
GUARDADO CHAR(1),
ACREDITADO CHAR(1),
NROCOMPROBANTE INTEGER)
AS
BEGIN
    FOR
        SELECT
            CLIENTE_NOTA_CREDITO.ID_CLIENTE_NOTA_CREDITO,
            CLIENTE_NOTA_CREDITO.ID_CLIENTE,
            CLIENTE.NOMBRE || ', ' || CLIENTE.APELLIDO AS RAZONSOCIAL,
            GRUPO_CLIENTE.DESCRIPCION,
            CIUDAD.NOMBRE,
            CLIENTE.RUC,
            CLIENTE.CELULAR,
            CLIENTE_NOTA_CREDITO.fecha,
            CLIENTE_NOTA_CREDITO.guardado,
            CLIENTE_NOTA_CREDITO.acreditado,
            CLIENTE_NOTA_CREDITO.nro_comprobante
        FROM
            CLIENTE_NOTA_CREDITO
            INNER JOIN CLIENTE ON
(CLIENTE_NOTA_CREDITO.ID_CLIENTE=CLIENTE.AI_CLIENTE)
            INNER JOIN GRUPO_CLIENTE ON
(CLIENTE.ID_GRUPO_CLIENTE=GRUPO_CLIENTE.ID_GRUPO_CLIENTE)
            INNER JOIN CIUDAD ON (CLIENTE.ID_CIUAD=CIUDAD.ID_CIUAD)
        WHERE
            (CLIENTE_NOTA_CREDITO.FECHA BETWEEN :DESDE AND :HASTA)
        ORDER BY
            CLIENTE_NOTA_CREDITO.ID_CLIENTE_NOTA_CREDITO DESC
        INTO
            :ID_CLIENTE_NOTA_CREDITO,
            :ID_CLIENTE,
            :RAZONSOCIAL,
            :DESCRIPCION,
            :NOMBRE,
            :RUC,
            :CELULAR,
            :FECHA,
            :GUARDADO,
            :ACREDITADO,
            :NROCOMPROBANTE
    DO
        BEGIN
            SELECT
                SUM(CLIENTE_NOTA_CREDITO_D.CANTIDAD * CLIENTE_NOTA_CREDITO_D.PRECIO) AS
TOTALDEVOLUCION
            FROM
                CLIENTE_NOTA_CREDITO_D
            WHERE
                (CLIENTE_NOTA_CREDITO_D.ID_CLIENTE_NOTA_CREDITO =
:ID_CLIENTE_NOTA_CREDITO)
            INTO

```

```

        :TOTALDEVOLUCION;

        SUSPEND;
    END
END^

SET TERM ; ^

/* Definition for the `LISTADOMOVIMIENTOALMACEN` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE LISTADOMOVIMIENTOALMACEN(
    DESDE DATE,
    HASTA DATE)
RETURNS (
    ID_ALMACEN_MO INTEGER,
    DESCRIPCION VARCHAR(80),
    FECHA DATE,
    NATU SMALLINT,
    GUARDADO VARCHAR(1),
    TOTALMOVIMIENTO DOUBLE PRECISION,
    NATURALEZA VARCHAR(25))
AS
BEGIN
    FOR
        SELECT
            ALMACEN_MOVI.ID_ALMACEN_MO,
            ALMACEN_MOVI.DESCRIPCION,
            ALMACEN_MOVI.FECHA,
            ALMACEN_MOVI.NATU,
            ALMACEN_MOVI.GUARDADO
        FROM
            ALMACEN_MOVI
        WHERE
            (ALMACEN_MOVI.FECHA BETWEEN :DESDE AND :HASTA)
        ORDER BY
            ALMACEN_MOVI.ID_ALMACEN_MO DESC

    INTO
        :ID_ALMACEN_MO,
        :DESCRIPCION,
        :FECHA,
        :NATU,
        :GUARDADO

    DO
        BEGIN
            if (NATU = 1) then
            begin
                NATURALEZA = 'Entrada';
            end
            else
            begin
                NATURALEZA = 'Salida';
            end
        end
    end
END

```

```

end
    SELECT
    SUM(ALMACEN_MOVI_D.CANTIDAD * ALMACEN_MOVI_D.PRECIO) AS TOTALENTRADA
FROM
    ALMACEN_MOVI_D
WHERE
    (ALMACEN_MOVI_D.ID_ALMACEN_MO = :ID_ALMACEN_MO)
    INTO
    :TOTALMOVIMIENTO;
    if (TOTALMOVIMIENTO IS NULL) then
    BEGIN
    TOTALMOVIMIENTO = 0;
    END
    SUSPEND;
    END
END^

```

```

SET TERM ; ^

```

```

/* Definition for the `LISTADOPARACOBranza` procedure : */

```

```

SET TERM ^ ;

```

```

CREATE OR ALTER PROCEDURE LISTADOPARACOBranza(
    CODIGOGRUPO INTEGER,
    FECHADECOBRO DATE)

```

```

RETURNS (
    ID_CUENTA_CLIENTE INTEGER,
    ID_CLIENTE INTEGER,
    FECHA DATE,
    NRO VARCHAR(35),
    DESCRIPCION VARCHAR(35),
    DEBITO DOUBLE PRECISION,
    CREDITO DOUBLE PRECISION,
    VENCIMIENTO DATE,
    ID_VENDEDOR SMALLINT,
    NOMBRE VARCHAR(35),
    DE_HASTA VARCHAR(15),
    DESCRIPCION1 VARCHAR(35),
    NOMBRE1 VARCHAR(35),
    APELLIDO VARCHAR(80),
    DIRECCION VARCHAR(80),
    RUC VARCHAR(35),
    CELULAR VARCHAR(35),
    DESCRIPCION2 VARCHAR(35),
    MEMO VARCHAR(500),
    BARRIO VARCHAR(35),
    CALIFICACION VARCHAR(35),
    OBSERVACION VARCHAR(500),
    ORDEN_VISITA SMALLINT,
    FECHA_NACIMIENTO DATE,
    DIRECCIONTRABAJO VARCHAR(500))

```

```

AS
BEGIN

```

```

FOR
  SELECT
    CUENTA_CLIENTE.ID_CUENTA_CLIENTE,
    CUENTA_CLIENTE.ID_CLIENTE,
    CUENTA_CLIENTE.FECHA,
    CUENTA_CLIENTE.NRO,
    CUENTA_CLIENTE.DESCRIPCION,
    CUENTA_CLIENTE.DEBITO,
    CUENTA_CLIENTE.VENCIMIENTO,
    VENDEDOR.ID_VENDEDOR,
    VENDEDOR.NOMBRE,
    CUENTA_CLIENTE.DE_HASTA,
    GRUPO_CLIENTE.DESCRIPCION,
    CLIENTE.NOMBRE,
    CLIENTE.APELLIDO,
    CLIENTE.DIRECCION,
    CLIENTE.RUC,
    CLIENTE.CELULAR,
    FORMA_PAGO.DESCRIPCION,
    CLIENTE.MEMO,
    CLIENTE.BARRIO,
    CLIENTE.CALIFICACION,
    CLIENTE.OBSERVACION,
    CLIENTE.ORDEN_VISITA,
    CLIENTE.FECHA_NACIMIENTO,
    CLIENTE.DIRECCIONTRABAJO
  FROM
    CUENTA_CLIENTE
    INNER JOIN CLIENTE ON (CUENTA_CLIENTE.ID_CLIENTE=CLIENTE.AI_CLIENTE)
    INNER JOIN GRUPO_CLIENTE ON
(CUENTA_CLIENTE.ID_PERSONAL=VENDEDOR.ID_VENDEDOR)
    INNER JOIN VENDEDOR ON
(CUENTA_CLIENTE.ID_PERSONAL=VENDEDOR.ID_VENDEDOR)
    INNER JOIN FORMA_PAGO ON
(CUENTA_CLIENTE.ID_FORMA_PAGO=FORMA_PAGO.ID_FORMA_PAGO)
  WHERE
    (CLIENTE.ID_GRUPO_CLIENTE = :CODIGOGRUPO) AND
    (CUENTA_CLIENTE.VENCIMIENTO <= :FECHADECOBRO) AND
    (CUENTA_CLIENTE.REFERENCIA IS NULL)
  ORDER BY
    CUENTA_CLIENTE.ID_CLIENTE

  INTO
    :ID_CUENTA_CLIENTE,
    :ID_CLIENTE,
    :FECHA,
    :NRO,
    :DESCRIPCION,
    :DEBITO,
    :VENCIMIENTO,
    :ID_VENDEDOR,
    :NOMBRE,
    :DE_HASTA,
    :DESCRIPCION1,

```

```

        :NOMBRE1,
        :APELLIDO,
        :DIRECCION,
        :RUC,
        :CELULAR,
        :DESCRIPCION2,
        :MEMO,
        :BARRIO,
        :CALIFICACION,
        :OBSERVACION,
        :ORDEN_VISITA,
        :FECHA_NACIMIENTO,
        :DIRECCIONTRABAJO
DO
BEGIN
SELECT
    SUM(CUENTA_CLIENTE.CREDITO) AS CREDITO
FROM
    CUENTA_CLIENTE
WHERE
    (CUENTA_CLIENTE.REFERENCIA = :ID_CUENTA_CLIENTE)
    INTO
    :CREDITO;
    if (CREDITO IS NULL) then
    BEGIN
    CREDITO = 0;
    END

    if (DEBITO > CREDITO) then
    BEGIN
        SUSPEND;
    END

END

END
END^

SET TERM ; ^

/* Definition for the `LISTADOPORGRUPOACTIVOINACTIVO` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE LISTADOPORGRUPOACTIVOINACTIVO (
    ID_GRUPO_CLIENTE INTEGER,
    OPCION INTEGER)
RETURNS (
    AI_CLIENTE INTEGER,
    RAZONSOCIAL VARCHAR(118),
    RUC VARCHAR(35),
    BARRIO VARCHAR(35),
    CELULAR VARCHAR(35),
    EMAIL VARCHAR(100),
    SALDODELCLIENTE DOUBLE PRECISION)

```

```

AS
BEGIN
    FOR
        SELECT
            CLIENTE.AI_CLIENTE,
            CLIENTE.NOMBRE || ' ' , ' ' || CLIENTE.APELLIDO AS RAZONSOCIAL,
            CLIENTE.RUC,
            CLIENTE.BARRIO,
            CLIENTE.CELULAR,
            CLIENTE.EMAIL
        FROM
            CLIENTE
        WHERE
            (CLIENTE.ID_GRUPO_CLIENTE = :ID_GRUPO_CLIENTE) AND
            (CLIENTE.ACTIVO = :OPCION)
        ORDER BY
            CLIENTE.AI_CLIENTE

        INTO
            :AI_CLIENTE,
            :RAZONSOCIAL,
            :RUC,
            :BARRIO,
            :CELULAR,
            :EMAIL
    DO
        BEGIN
            SELECT
                SUM(CUENTA_CLIENTE.DEBITO - CUENTA_CLIENTE.CREDITO) AS SALDOACTUAL
            FROM
                CUENTA_CLIENTE
            WHERE
                (CUENTA_CLIENTE.ID_CLIENTE = :AI_CLIENTE)
            INTO
                :SALDODELCLIENTE;

            SUSPEND;
        END
    END^

    SET TERM ; ^

    /* Definition for the `ENTERO_CORTO` domain : */

    CREATE DOMAIN ENTERO_CORTO AS
        SMALLINT;

    /* Definition for the `NO_YES_YES` domain : */

    CREATE DOMAIN NO_YES_YES AS
        VARCHAR(1)
        DEFAULT 'S'
        CHECK (VALUE IN ('S','N'));

```



/\* Structure for the `EMPRESA` table : \*/

```
CREATE TABLE EMPRESA (  
  ID_EMPRESA AI NOT NULL,  
  RAZON DESCRIPTION_LARGE,  
  DIRECCION DIRECCION,  
  ID_CIUADAD AI,  
  TELEFONO1 DESCRIPTION_SHORT,  
  TELEFONO2 DESCRIPTION_SHORT,  
  FAX1 DESCRIPTION_SHORT,  
  FAX2 DESCRIPTION_SHORT,  
  EMAIL DESCRIPTION_LARGE,  
  WEB DESCRIPTION_LARGE,  
  REGISTRO DESCRIPTION_LARGE,  
  LINEA1 DIRECCION,  
  LINEA2 DIRECCION,  
  IDEOLOGIA DIRECCION,  
  IMAGEN PICTURE,  
  MEMBRETE_HACIENDA YES_NO,  
  NOMBRE DESCRIPTION_SHORT,  
  OFRECER_PROMOCION NO_YES_YES,  
  PROMOCION_LINEA MEMO,  
  PROMOCION_LINEA2 MEMO,  
  MAXIMO_ITEMS_VENTA ENTERO,  
  MAXIMO_ITEMS_TRANFERENCIA ENTERO,  
  MAXIMO_ITEMS_ENTRADA AI,  
  MAXIMO_ITEMS_SALIDA AI,  
  MAXIMO_ITEMS_COMPRA AI,  
  PERMITIR_AUTO_CARGA YES_NO,  
  PANTALLA_MODAL YES_NO,  
  COMPROBAR_BA_PRO YES_NO,  
  COMPROBAR_DO_CLI YES_NO,  
  CAN_INSERT_INCOM YES_NO,  
  GUARDAR_EVENTOSS YES_NO,  
  PAGARE PICTURE,  
  ID_LINGUAJE ENTERO_CORTO,  
  NOMBRE_SERVIDOR DESCRIPTION_LARGE,  
  PUERTO_SERVIDOR ENTERO,  
  CONTRASE_EMAIL DESCRIPCION_MEDIO,  
  EMAILDESALDOBIENVENIDA SI_NO_NO NOT NULL,  
  CONTROLARSTOCK SI_NO_NO NOT NULL);
```

```
ALTER TABLE EMPRESA ADD CONSTRAINT PK_EMPRESA PRIMARY KEY (ID_EMPRESA);
```

/\* Structure for the `PRODUCTO\_MARCA` table : \*/

```
CREATE TABLE PRODUCTO_MARCA (  
  ID_MARCA_PRODUCTO ENTERO NOT NULL,  
  DESCRIPCION NOMBRE_CORTO);
```

```
ALTER TABLE PRODUCTO_MARCA ADD CONSTRAINT PK_PRODUCTO_MARCA PRIMARY KEY  
(ID_MARCA_PRODUCTO);
```

```

/* Definition for the `DESCRIPCION_CORTO` domain : */

CREATE DOMAIN DESCRIPCION_CORTO AS
    VARCHAR(25);

/* Structure for the `PRODUCTO_SEXO` table : */

CREATE TABLE PRODUCTO_SEXO (
    ID_PRODUCTO_SEXO ENTERO NOT NULL,
    DESCRIPCION DESCRIPCION_CORTO);

ALTER TABLE PRODUCTO_SEXO ADD CONSTRAINT PK_PRODUCTO_SEXO PRIMARY KEY
(ID_PRODUCTO_SEXO);

/* Structure for the `PRODUCTO_TIPO` table : */

CREATE TABLE PRODUCTO_TIPO (
    ID_TIPO_PRODUCTO ENTERO NOT NULL,
    DESCRIPCION NOMBRE_CORTO);

ALTER TABLE PRODUCTO_TIPO ADD CONSTRAINT PK_PRODUCTO_TIPO PRIMARY KEY
(ID_TIPO_PRODUCTO);

/* Definition for the `LISTADOSTOCK` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE LISTADOSTOCK
RETURNS (
    ID_PRODUCTO INTEGER,
    DESCRIPCION VARCHAR(35),
    CODIGO_BARRAS VARCHAR(35),
    PRECIO1 DOUBLE PRECISION,
    PRECIO2 DOUBLE PRECISION,
    PRECIO3 DOUBLE PRECISION,
    OFERTAR CHAR(1),
    PESO DOUBLE PRECISION,
    PRECIO_COSTO DOUBLE PRECISION,
    STOCKACTUAL DOUBLE PRECISION,
    CAPITAL DOUBLE PRECISION,
    DESCRIPCIONTIPO VARCHAR(35),
    DESCRIPCIONMARCA VARCHAR(35),
    DESCRIPCIONSEXO VARCHAR(35))
AS
declare variable CONTROSTOCK VARCHAR(1);
BEGIN
    FOR
SELECT
    PRODUCTO.ID_PRODUCTO,
    PRODUCTO.DESCRIPCION,
    PRODUCTO.CODIGO_BARRAS,

```

```

PRODUCTO.PRECIO1,
PRODUCTO.PRECIO2,
PRODUCTO.PRECIO3,
PRODUCTO.OFERTAR,
PRODUCTO.PESO,
PRODUCTO.PRECIO_COSTO,
PRODUCTO_TIPO.DESCRIPCION,
PRODUCTO_MARCA.DESCRIPCION,
PRODUCTO_SEXO.DESCRIPCION
FROM
PRODUCTO
INNER JOIN PRODUCTO_TIPO ON
(PRODUCTO.ID_TIPO_PRODUCTO=PRODUCTO_TIPO.ID_TIPO_PRODUCTO)
INNER JOIN PRODUCTO_MARCA ON
(PRODUCTO.ID_MARCA_PRODUCTO=PRODUCTO_MARCA.ID_MARCA_PRODUCTO)
INNER JOIN PRODUCTO_SEXO ON
(PRODUCTO.ID_PRODUCTO_SEXO=PRODUCTO_SEXO.ID_PRODUCTO_SEXO)
ORDER BY
PRODUCTO.ID_PRODUCTO DESC
INTO
:ID_PRODUCTO,
:DESCRIPCION,
:CODIGO_BARRAS,
:PRECIO1,
:PRECIO2,
:PRECIO3,
:OFERTAR,
:PESO,
:PRECIO_COSTO,
:DESCRIPCIONTIPO,
:DESCRIPCIONMARCA,
:DESCRIPCIONSEXO

DO
BEGIN
SELECT
EMPRESA.CONTROLARSTOCK
FROM
EMPRESA
INTO
:CONTROSTOCK;

if (CONTROSTOCK = 'S') then
BEGIN
SELECT
STOCKACTUALDEUNPRODUCTO.SALDOACTUAL,
STOCKACTUALDEUNPRODUCTO.CAPITAL
FROM
STOCKACTUALDEUNPRODUCTO (:ID_PRODUCTO)
INTO
:STOCKACTUAL,
:CAPITAL;
END

SUSPEND;

```

```

END
END^

SET TERM ; ^

/* Definition for the `PLANILLAPARAREVENTA` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE PLANILLAPARAREVENTA(
    CODIGOCIUDAD INTEGER,
    CODIGOVENDEDOR INTEGER)
RETURNS (
    AI_CLIENTE INTEGER,
    RAZONSOCIAL VARCHAR(117),
    RUC VARCHAR(35),
    BARRIO VARCHAR(35),
    CELULAR VARCHAR(35),
    DIRECCION VARCHAR(80),
    DIRECCIONTRABAJO VARCHAR(500),
    DESCRIPCION VARCHAR(35),
    SALDOACTUAL DOUBLE PRECISION,
    ULTIMAFECHADEVENTA DATE,
    DIASULTIMAVENTA INTEGER,
    LUGARDETRABAJO VARCHAR(80),
    MEMORANDUN VARCHAR(500),
    MONTOVENCIDO DOUBLE PRECISION)
AS
BEGIN
    FOR
        SELECT
            CLIENTE.AI_CLIENTE,
            CLIENTE.NOMBRE || ', ' || CLIENTE.APELLIDO AS RAZONSOCIAL,
            CLIENTE.RUC,
            CLIENTE.BARRIO,
            CLIENTE.CELULAR,
            CLIENTE.DIRECCION,
            CLIENTE.DIRECCIONTRABAJO,
            GRUPO_CLIENTE.DESCRIPCION,
            CLIENTE.lugartrabajo,
            cliente.memo
        FROM
            CLIENTE
        INNER JOIN GRUPO_CLIENTE ON
            (CLIENTE.ID_GRUPO_CLIENTE=GRUPO_CLIENTE.ID_GRUPO_CLIENTE)
        WHERE
            (CLIENTE.ACTIVO = 1) AND
            (CLIENTE.CALIFICACION = 'E') AND
            (CLIENTE.ID_CIUAD = :CODIGOCIUDAD) AND
            (CLIENTE.ID_VENDEDOR = :CODIGOVENDEDOR)
        ORDER BY
            CLIENTE.ID_GRUPO_CLIENTE

    INTO

```

```

        :AI_CLIENTE,
        :RAZONSOCIAL,
        :RUC,
        :BARRIO,
        :CELULAR,
        :DIRECCION,
        :DIRECCIONTRABAJO,
        :DESCRIPCION,
        :LUGARDETRABAJO,
        :MEMORANDUN
DO
BEGIN
SELECT
SUM(CUENTA_CLIENTE.DEBITO - CUENTA_CLIENTE.CREDITO) AS SALDOACTUAL
FROM
CUENTA_CLIENTE
WHERE
(CUENTA_CLIENTE.ID_CLIENTE = :AI_CLIENTE)
INTO
:SALDOACTUAL;

if (SALDOACTUAL IS NULL) THEN
BEGIN
SALDOACTUAL = 0;
END

SELECT
SUM(LISTADEMOROSOSD.MONTOVENCIDO) AS MONTOVENCIDO
FROM
LISTADEMOROSOSD(:AI_CLIENTE)
INTO
:MONTOVENCIDO;

if (MONTOVENCIDO IS NULL) then
BEGIN
MONTOVENCIDO = 0;
END

SELECT
MAX(VENTA.FECHA) AS FIELD_1
FROM
VENTA
WHERE
(VENTA.ID_CLIENTE = :AI_CLIENTE)
INTO
:ULTIMAFECHADEVENTA;

if (ULTIMAFECHADEVENTA IS NOT NULL) then
BEGIN
DIASULTIMAVENTA = current_date - ULTIMAFECHADEVENTA;
END

```

```

        SUSPEND;
    END
END^

SET TERM ; ^

/* Definition for the `PLANILLAPARAREVENTAINACTIVO` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE PLANILLAPARAREVENTAINACTIVO(
    CODIGOCIUDAD INTEGER)
RETURNS (
    AI_CLIENTE INTEGER,
    RAZONSOCIAL VARCHAR(117),
    RUC VARCHAR(35),
    BARRIO VARCHAR(35),
    CELULAR VARCHAR(35),
    DIRECCION VARCHAR(80),
    DIRECCIONTRABAJO VARCHAR(500),
    DESCRIPCION VARCHAR(35),
    LUGARDETRABAJO VARCHAR(80))
AS
BEGIN
    FOR
        SELECT
            CLIENTE.AI_CLIENTE,
            CLIENTE.NOMBRE || ', ' || CLIENTE.APELLIDO AS RAZONSOCIAL,
            CLIENTE.RUC,
            CLIENTE.BARRIO,
            CLIENTE.CELULAR,
            CLIENTE.DIRECCION,
            CLIENTE.DIRECCIONTRABAJO,
            GRUPO_CLIENTE.DESCRIPCION,
            CLIENTE.lugartrabajo
        FROM
            CLIENTE
        INNER JOIN GRUPO_CLIENTE ON
        (CLIENTE.ID_GRUPO_CLIENTE=GRUPO_CLIENTE.ID_GRUPO_CLIENTE)
        WHERE
            (CLIENTE.ACTIVO = 0) AND
            (CLIENTE.CALIFICACION = 'E') AND
            (CLIENTE.ID_CIUADAD = :CODIGOCIUDAD)
        ORDER BY
            CLIENTE.ID_GRUPO_CLIENTE

    INTO
        :AI_CLIENTE,
        :RAZONSOCIAL,
        :RUC,
        :BARRIO,
        :CELULAR,

```

```

        :DIRECCION,
        :DIRECCIONTRABAJO,
        :DESCRIPCION,
        :LUGARDETRABAJO
DO
BEGIN

        SUSPEND;
END
END^

SET TERM ; ^

/* Definition for the `PROCOBRANZAMOVIL` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE PROCOBRANZAMOVIL(
    BUSCADOR INTEGER)
RETURNS (
    ID_COBRANZAMOVIL INTEGER,
    FECHA DATE,
    DESCRIPCION VARCHAR(35),
    PASS VARCHAR(15),
    MONTOACOBRAR DOUBLE PRECISION,
    CANTIDADCLIENTE INTEGER,
    MONTOCOBRADO DOUBLE PRECISION,
    CANTIDADCLIENTECOBRADO DOUBLE PRECISION)
AS
BEGIN
    FOR
        SELECT
            COBRANZAMOVIL.ID_COBRANZAMOVIL,
            COBRANZAMOVIL.FECHA,
            GRUPO_CLIENTE.DESCRIPCION,
            COBRANZAMOVIL.PASS,
            COBRANZAMOVIL.MONTOACOBRAR,
            COBRANZAMOVIL.CANTIDADCLIENTE
        FROM
            COBRANZAMOVIL
        INNER JOIN GRUPO_CLIENTE ON
            (COBRANZAMOVIL.ID_GRUPO_CLIENTE=GRUPO_CLIENTE.ID_GRUPO_CLIENTE)
        WHERE
            (COBRANZAMOVIL.ID_VENDEDOR = :BUSCADOR) AND
            (COBRANZAMOVIL.CERRADO = 'N')

    INTO
        :ID_COBRANZAMOVIL,
        :FECHA,
        :DESCRIPCION,
        :PASS,

```

```

        :MONTOACOBRAR,
        :CANTIDADCLIENTE
DO
BEGIN

SELECT
SUM(COBRANZAMOVILD.MONTOPAGADO) AS MONTOPAGADO,
COUNT(COBRANZAMOVILD.ID_COBRANZAMOVILD) AS FIELD_1
FROM
COBRANZAMOVILD
WHERE
(COBRANZAMOVILD.ID_COBRANZAMOVIL = :ID_COBRANZAMOVIL) AND
(COBRANZAMOVILD.MONTOPAGADO > 0)
INTO
:MONTOCOBRADO,
:CANTIDADCLIENTECOBRADO;

SUSPEND;
END
END^

SET TERM ; ^

/* Definition for the `PROCOBRANZAMOVILSIN` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE PROCOBRANZAMOVILSIN
RETURNS (
ID_COBRANZAMOVIL INTEGER,
FECHA DATE,
DESCRIPCION VARCHAR(35),
PASS VARCHAR(15),
MONTOACOBRAR DOUBLE PRECISION,
CANTIDADCLIENTE INTEGER,
MONTOCOBRADO DOUBLE PRECISION,
CANTIDADCLIENTECOBRADO DOUBLE PRECISION,
ID_GRUPO_CLIENTE INTEGER,
ID_VENDEDOR INTEGER,
META DOUBLE PRECISION)
AS
BEGIN
FOR
SELECT
COBRANZAMOVIL.ID_COBRANZAMOVIL,
COBRANZAMOVIL.FECHA,
GRUPO_CLIENTE.DESCRIPCION,
COBRANZAMOVIL.PASS,
COBRANZAMOVIL.MONTOACOBRAR,
COBRANZAMOVIL.CANTIDADCLIENTE,
COBRANZAMOVIL.id_grupo_cliente,
COBRANZAMOVIL.id_vendedor,
COBRANZAMOVIL.META
FROM

```



```

        COBRANZAMOVIL
        INNER JOIN GRUPO_CLIENTE ON
(COBRANZAMOVIL.ID_GRUPO_CLIENTE=GRUPO_CLIENTE.ID_GRUPO_CLIENTE)
        WHERE
            (COBRANZAMOVIL.CERRADO = 'N')
            ORDER BY COBRANZAMOVIL.ID_COBRANZAMOVIL

    INTO
        :ID_COBRANZAMOVIL,
        :FECHA,
        :DESCRIPCION,
        :PASS,
        :MONTOACOBRAR,
        :CANTIDADCLIENTE,
        :id_grupo_cliente,
        :ID_VENDEDOR,
        :META

DO
BEGIN

    SELECT
    SUM(COBRANZAMOVILD.MONTOPAGADO) AS MONTOPAGADO,
    COUNT(COBRANZAMOVILD.ID_COBRANZAMOVILD) AS FIELD_1
FROM
    COBRANZAMOVILD
WHERE
    (COBRANZAMOVILD.ID_COBRANZAMOVIL = :ID_COBRANZAMOVIL) AND
    (COBRANZAMOVILD.MONTOPAGADO > 0)
    INTO
        :MONTOCOBRADO,
        :CANTIDADCLIENTECOBRADO;

        SUSPEND;
    END
END^

SET TERM ; ^

/* Definition for the `PRODUCTO_BI` trigger : */

SET TERM ^ ;

CREATE TRIGGER PRODUCTO_BI FOR PRODUCTO
ACTIVE BEFORE
INSERT
POSITION 0
AS
BEGIN
    /* Trigger body */
    if (NEW.ID_PRODUCTO_SEXO is null) then
        begin
            NEW.ID_PRODUCTO_SEXO = 1;
        end
    end

```

END^

SET TERM ; ^

/\* Definition for the `PRODUCTO\_ELIMINAR\_PRODUCTO` trigger : \*/

SET TERM ^ ;

CREATE TRIGGER PRODUCTO\_ELIMINAR\_PRODUCTO FOR PRODUCTO  
ACTIVE AFTER

DELETE  
POSITION 0  
AS

BEGIN

POST\_EVENT 'NUEVOPRODUCTO';  
/\* Trigger body \*/  
END^

SET TERM ; ^

/\* Definition for the `PRODUCTO\_MODIFICAR\_PRODUCTO` trigger : \*/

SET TERM ^ ;

CREATE TRIGGER PRODUCTO\_MODIFICAR\_PRODUCTO FOR PRODUCTO  
ACTIVE AFTER

UPDATE  
POSITION 0  
AS

BEGIN

POST\_EVENT 'NUEVOPRODUCTO';  
/\* Trigger body \*/  
END^

SET TERM ; ^

/\* Definition for the `PRODUCTO\_NUEVO\_PRODUCTO` trigger : \*/

SET TERM ^ ;

CREATE TRIGGER PRODUCTO\_NUEVO\_PRODUCTO FOR PRODUCTO  
ACTIVE AFTER

INSERT  
POSITION 0  
AS

BEGIN

POST\_EVENT 'NUEVOPRODUCTO';

/\* Trigger body \*/  
END^

SET TERM ; ^

```

/* Definition for the `RECIBOANALISIS` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE RECIBOANALISIS(
    CODIGOCIUDAD INTEGER,
    FECHACOBRO DATE,
    CODIGOVENDEDOR INTEGER)
RETURNS (
    AI_CLIENTE INTEGER,
    ID_CUENTA_CLIENTE INTEGER,
    FECHA DATE,
    VENCIMIENTO DATE,
    DEBITO DOUBLE PRECISION,
    CREDITO DOUBLE PRECISION,
    CANCELAR CHAR(1))
AS
BEGIN
    FOR
    SELECT
        CLIENTE.AI_CLIENTE,
        CUENTA_CLIENTE.ID_CUENTA_CLIENTE,
        CUENTA_CLIENTE.FECHA,
        CUENTA_CLIENTE.VENCIMIENTO,
        CUENTA_CLIENTE.DEBITO
    FROM
        CUENTA_CLIENTE
        INNER JOIN CLIENTE ON (CUENTA_CLIENTE.ID_CLIENTE=CLIENTE.AI_CLIENTE)
        INNER JOIN GRUPO_CLIENTE ON
        (CLIENTE.ID_GRUPO_CLIENTE=GRUPO_CLIENTE.ID_GRUPO_CLIENTE)
    WHERE
        (CLIENTE.ID_CIUADAD = :CODIGOCIUDAD) AND
        (GRUPO_CLIENTE.id_vendedor = :CODIGOVENDEDOR) AND
        (CUENTA_CLIENTE.REFERENCIA IS NULL) AND
        (CUENTA_CLIENTE.VENCIMIENTO <= :FECHACOBRO) AND
        (CUENTA_CLIENTE.DEBITO > 0) and
        (CUENTA_CLIENTE.descripcion <> 'Entrega')
    INTO
        :AI_CLIENTE,
        :ID_CUENTA_CLIENTE,
        :FECHA,
        :VENCIMIENTO,
        :DEBITO
    DO
    BEGIN
        CANCELAR = 'N';

        SELECT
            SUM(CUENTA_CLIENTE.CREDITO) AS CREDITO
        FROM
            CUENTA_CLIENTE
        WHERE
            ((CUENTA_CLIENTE.REFERENCIA = :ID_CUENTA_CLIENTE)

```

```

AND
(CUENTA_CLIENTE.fecha < :FECHACOBRO))
INTO
:CREDITO;
if (CREDITO IS NULL) then
BEGIN
CREDITO = 0;
END

if (CREDITO >= DEBITO) then
BEGIN
    CANCELAR = 'S';
END

if (CANCELAR = 'N') then
BEGIN
    SUSPEND;
END

END
END^

SET TERM ; ^

/* Definition for the `RECIBOANALISISISCUE` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE RECIBOANALISISISCUE(
    CODIGOCLIENTE INTEGER)
RETURNS (
    ID_CUENTA_CLIENTE INTEGER,
    DEBITO DOUBLE PRECISION,
    PAGOS DOUBLE PRECISION,
    SALDO DOUBLE PRECISION)
AS
BEGIN
    FOR
        SELECT
            CUENTA_CLIENTE.ID_CUENTA_CLIENTE,
            CUENTA_CLIENTE.DEBITO
        FROM
            CUENTA_CLIENTE
        WHERE
            (CUENTA_CLIENTE.ID_CLIENTE = :CODIGOCLIENTE) AND
            (CUENTA_CLIENTE.REFERENCIA IS NULL)
        ORDER BY
            CUENTA_CLIENTE.ID_CUENTA_CLIENTE asc

        INTO
            :ID_CUENTA_CLIENTE,
            :DEBITO
    DO

```

```

BEGIN

SELECT
SUM(CUENTA_CLIENTE.CREDITO) AS PAGOS
FROM
CUENTA_CLIENTE
WHERE
(CUENTA_CLIENTE.ID_CLIENTE = :CODIGOCLIENTE) AND
(CUENTA_CLIENTE.REFERENCIA = :ID_CUENTA_CLIENTE)
INTO
:PAGOS;

if (PAGOS IS NULL) then
BEGIN
PAGOS = 0;
END

SALDO = DEBITO - PAGOS;
if (SALDO > 0) then
BEGIN
SUSPEND;
END

END

END^

SET TERM ; ^

/* Definition for the `RECIBOANALISIS1` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE RECIBOANALISIS1(
CODIGOCIUDAD INTEGER,
FECHACOBRO DATE,
CODIGOVENDEDOR INTEGER)
RETURNS (
AI_CLIENTE INTEGER,
MONTOACOBRAR DOUBLE PRECISION)
AS
declare variable IDMAXIMOID INTEGER;
BEGIN
FOR
SELECT distinct
RECIBOANALISIS.AI_CLIENTE
FROM
RECIBOANALISIS(:CODIGOCIUDAD, :FECHACOBRO, :CODIGOVENDEDOR)
ORDER BY RECIBOANALISIS.AI_CLIENTE
into

```

```

:AI_CLIENTE
DO
BEGIN

    SELECT
    MIN(RECIBOANALISISISCUE.ID_CUENTA_CLIENTE) AS IDMAXIMOID
FROM
RECIBOANALISISISCUE(:AI_CLIENTE)
INTO
:IDMAXIMOID;

SELECT
RECIBOANALISISISCUE.SALDO
FROM
RECIBOANALISISISCUE(:AI_CLIENTE)
WHERE
    (RECIBOANALISISISCUE.ID_CUENTA_CLIENTE = :IDMAXIMOID)
into
:MONTOACOBRAR;

    SUSPEND;
END

END^

SET TERM ; ^

/* Definition for the `RECIBO_GENERAR_MENSA` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE RECIBO_GENERAR_MENSA(
    CODIGOCOBranza INTEGER)
RETURNS (
    ID_COBRANZA_DETALLE INTEGER,
    RAZONSOCIAL VARCHAR(117),
    MONTO DOUBLE PRECISION,
    AI_CLIENTE INTEGER,
    NRORECIBO INTEGER,
    CELULAR VARCHAR(35),
    SALDOACTUAL DOUBLE PRECISION)
AS
BEGIN
    FOR
        SELECT
            RECIBO_CLIENTE_DETALLE.ID_COBRANZA_DETALLE,
            CLIENTE.NOMBRE || ', ' || CLIENTE.APELLIDO AS RAZONSOCIAL,
            RECIBO_CLIENTE_DETALLE.MONTO,
            CLIENTE.AI_CLIENTE,
            RECIBO_CLIENTE_DETALLE.NRORECIBO,
            CLIENTE.CELULAR

```

```

FROM
    RECIBO_CLIENTE_DETALLE
    INNER JOIN CLIENTE ON
(RECIBO_CLIENTE_DETALLE.ID_CLIENTE=CLIENTE.AI_CLIENTE)
WHERE
    (RECIBO_CLIENTE_DETALLE.ID_COBRANZA = :CODIGOCOBranza) AND
    (CLIENTE.celular IS NOT NULL)
ORDER BY
    RECIBO_CLIENTE_DETALLE.ID_COBRANZA_DETALLE

INTO
    :ID_COBRANZA_DETALLE,
    :RAZONSOCIAL,
    :MONTO,
    :AI_CLIENTE,
    :NRORECIBO,
    :CELULAR
DO
BEGIN
SELECT
    SUM(CUENTA_CLIENTE.DEBITO - CUENTA_CLIENTE.CREDITO) AS SALDOACTUAL
FROM
    CUENTA_CLIENTE
WHERE
    (CUENTA_CLIENTE.ID_CLIENTE = :AI_CLIENTE)
INTO
    :SALDOACTUAL;
if (SALDOACTUAL IS NULL) then
BEGIN
    SALDOACTUAL = 0;
END
    SUSPEND;
END
END^

SET TERM ; ^

/* Definition for the `RESUMENDEUNAVENTA` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE RESUMENDEUNAVENTA(
    IDVENTA INTEGER)
RETURNS (
    ID_CUENTA_CLIENTE INTEGER,
    ID_COMPROBANTE INTEGER,
    NRO VARCHAR(35),
    FECHA DATE,
    CREDITO DOUBLE PRECISION,
    DEBITO DOUBLE PRECISION,
    DESCRIPCION VARCHAR(35),
    VENCECIMIENTO DATE,
    CANCELADO CHAR(1),
    ID_CLIENTE INTEGER,

```

```

    DESDE_HASTA VARCHAR(15),
    DIASVENCIDA INTEGER,
    SALDOLINEAL DOUBLE PRECISION)
AS
BEGIN

diasvencida = 0;

FOR
    SELECT
        CUENTA_CLIENTE.ID_CUENTA_CLIENTE,
        CUENTA_CLIENTE.ID_COMPROBANTE,
        CUENTA_CLIENTE.NRO,
        CUENTA_CLIENTE.FECHA,
        CUENTA_CLIENTE.debito,
        CUENTA_CLIENTE.CREDITO,
        CUENTA_CLIENTE.DESCRIPCION,
        CUENTA_CLIENTE.vencimiento,
        CUENTA_CLIENTE.ID_CLIENTE,
        CUENTA_CLIENTE.de_hasta
    FROM
        CUENTA_CLIENTE
    WHERE
        (CUENTA_CLIENTE.id_venta = :IDVENTA) AND
        (CUENTA_CLIENTE.REFERENCIA IS NULL)
    order by cuenta_cliente.ID_CUENTA_CLIENTE desc

    INTO
        :ID_CUENTA_CLIENTE,
        :ID_COMPROBANTE,
        :NRO,
        :FECHA,
        :DEBITO,
        :CREDITO,
        :DESCRIPCION,
        :VENCECIMIENTO,
        :ID_CLIENTE,
        :DESDE_HASTA
DO
    BEGIN
if (credito = 0) then
begin
    SELECT
        SUM(CUENTA_CLIENTE.CREDITO) AS CREDITO
    FROM
        CUENTA_CLIENTE
    WHERE
        (CUENTA_CLIENTE.REFERENCIA = :ID_CUENTA_CLIENTE)
    INTO
        :CREDITO;
END

    if (credito is null) then
begin

```



```

credito = 0;
end

if (credito >= debito) then
begin
CANCELADO = 'S';
diasvencida = 0;
SALDOLINEAL = 0;
end
ELSE
begin
CANCELADO = 'N';
if (VENCECIMIENTO < current_date) then
BEGIN
diasvencida = CURRENT_DATE - VENCECIMIENTO;
END
SALDOLINEAL = DEBITO - CREDITO;
end
SUSPEND;
SALDOLINEAL = 0;
diasvencida = 0;
END
END^

SET TERM ; ^

/* Definition for the `RESUMEN_CUENTA_POR_VENTA` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE RESUMEN_CUENTA_POR_VENTA(
    BUSCADOR INTEGER)
RETURNS (
    ID_CUENTA_CLIENTE INTEGER,
    DEBITO DOUBLE PRECISION,
    DESCRIPCION VARCHAR(35),
    PAGOS DOUBLE PRECISION,
    VENCIMIENTO DATE,
    SALDO DOUBLE PRECISION,
    DIASVENCIDO INTEGER,
    MONTOVENCIDO DOUBLE PRECISION)
AS
BEGIN
FOR
SELECT
    CUENTA_CLIENTE.ID_CUENTA_CLIENTE,
    CUENTA_CLIENTE.DEBITO,
    CUENTA_CLIENTE.DESCRIPCION,
    CUENTA_CLIENTE.vencimiento
FROM
    CUENTA_CLIENTE
WHERE
    (CUENTA_CLIENTE.ID_VENTA = :buscador) AND
    (CUENTA_CLIENTE.REFERENCIA IS NULL)

```

```

ORDER BY CUENTA_CLIENTE.ID_CUENTA_CLIENTE

INTO
    :ID_CUENTA_CLIENTE,
    :DEBITO,
    :DESCRIPCION,
    :VENCIMIENTO
DO
BEGIN
PAGOS = 0;
MONTOVENCIDO = 0;
SALDO = 0;
DIASVENCIDO = 0;

SELECT
SUM(CUENTA_CLIENTE.CREDITO) AS PAGOS
FROM
CUENTA_CLIENTE
WHERE
(CUENTA_CLIENTE.REFERENCIA = :ID_CUENTA_CLIENTE)
INTO
:PAGOS;

if (PAGOS IS NULL) then
BEGIN
PAGOS = 0;
END

if (PAGOS < DEBITO) then
BEGIN
if (CURRENT_DATE > VENCIMIENTO) then
BEGIN
DIASVENCIDO = current_date - VENCIMIENTO;
MONTOVENCIDO = (DEBITO - PAGOS);

END
END

SUSPEND;
END
END^

SET TERM ; ^

/* Definition for the `UPDATECLIENTEVENDEDOR` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE UPDATECLIENTEVENDEDOR
RETURNS (
    AI_CLIENTE INTEGER,
    ID_VENDEDOR INTEGER)
AS

```

```

BEGIN
  FOR
    SELECT
      CLIENTE.AI_CLIENTE
    FROM
      CLIENTE
    ORDER BY
      CLIENTE.AI_CLIENTE

    INTO
      :AI_CLIENTE
  DO
    BEGIN
      SELECT
        VENTA.ID_VENDEDOR
      FROM
        VENTA
      WHERE
        (VENTA.ID_CLIENTE = :AI_CLIENTE)
      INTO
        :ID_VENDEDOR;

      SUSPEND;
    END
  END^

SET TERM ; ^

/* Definition for the `updatecliente` procedure : */

SET TERM ^ ;

CREATE OR ALTER PROCEDURE "updatecliente"
RETURNS (
  ID_VENTA INTEGER,
  ID_CLIENTE INTEGER,
  ID_VENDEDOR INTEGER)
AS
BEGIN
  FOR
    SELECT
      VENTA.ID_VENTA,
      VENTA.ID_CLIENTE,
      VENTA.ID_VENDEDOR
    FROM
      VENTA
    ORDER BY
      VENTA.ID_VENTA

    INTO
      :ID_VENTA,
      :ID_CLIENTE,

```

```

        :ID_VENDEDOR
DO
BEGIN
UPDATE CLIENTE
SET CLIENTE.id_vendedor = :ID_VENDEDOR
WHERE CLIENTE.ai_cliente = :ID_CLIENTE;

        SUSPEND;
END
END^

SET TERM ; ^

/* Structure for the `CUENTA_CLIENTE_1` table : */

CREATE TABLE CUENTA_CLIENTE_1 (
    ID_CUENTA_CLIENTE NUMERO_MEDIO NOT NULL,
    ID_CLIENTE ENTERO,
    ID_COMPROBANTE NUMERO_MEDIO,
    NRO NOMBRE_CORTO,
    FECHA DATE,
    DESCRIPCION DESCRIPTION_SHORT,
    CREDITO NUMERO,
    DEBITO NUMERO,
    VENCIMIENTO DATE,
    REFERENCIA NUMERO_MEDIO,
    ID_PERSONAL AI,
    DE_HASTA ALIAS,
    ID_COBRANZA_DETALLE ENTERO,
    ID_VENTA ENTERO);

ALTER TABLE CUENTA_CLIENTE_1 ADD CONSTRAINT PK_CUENTA_CLIENTE_1 PRIMARY
KEY (ID_CUENTA_CLIENTE);

/* Structure for the `EMPRESA_CLAVE` table : */

CREATE TABLE EMPRESA_CLAVE (
    ID_EMPRESA AI,
    CLAVE DESCRIPCION_CORTO,
    PRODUCTO DESCRIPCION_CORTO,
    FECHA DATE);

/* Structure for the `SIS` table : */

CREATE TABLE SIS (
    PRODUCTO DESCRIPCION_CORTO NOT NULL,
    "1" AI,
    "2" AI,
    "3" AI,
    "4" AI,
    "5" AI,
    "6" AI,
    "7" AI,

```

```
"8" AI,  
"9" AI,  
"10" AI);
```

```
ALTER TABLE SIS ADD CONSTRAINT PK_SIS PRIMARY KEY (PRODUCTO);
```

```
/* Structure for the `CLIENTE_LLAMADA` table : */
```

```
CREATE TABLE CLIENTE_LLAMADA (  
  ID_CLIENTE_LLAMADA ENTERO NOT NULL,  
  ID_CLIENTE ENTERO,  
  ID_MOTIVO_LLAMADA ENTERO,  
  FECHA "DATE",  
  HORA HORA,  
  MEMO MEMO,  
  AGENDAR SI_NO_NO NOT NULL,  
  AGENDAR_FECHA "DATE",  
  TERMINADO SI_NO_NO NOT NULL,  
  NUMERO ALIAS,  
  ARCHIVO PICTURE,  
  ID_USUARIO ENTERO NOT NULL);
```

```
ALTER TABLE CLIENTE_LLAMADA ADD CONSTRAINT PK_CLIENTE_LLAMADA PRIMARY KEY  
(ID_CLIENTE_LLAMADA);
```

```
/* Structure for the `COMPRA` table : */
```

```
CREATE TABLE COMPRA (  
  ID_COMPRA ENTERO NOT NULL,  
  ID_USUARIO ENTERO,  
  ID_PROVEEDOR ENTERO,  
  FECHA FECHA,  
  NRO_FACTURA ALIAS,  
  RECIBIDO SI_NO_NO NOT NULL,  
  CONTADO SI_NO_NO NOT NULL);
```

```
ALTER TABLE COMPRA ADD CONSTRAINT PK_COMPRA PRIMARY KEY (ID_COMPRA);
```

```
/* Structure for the `PROVEEDOR` table : */
```

```
CREATE TABLE PROVEEDOR (  
  ID_PROVEEDOR ENTERO NOT NULL,  
  RUC ALIAS,  
  RAZONSOCIAL NOMBRE_LARGO,  
  TELEFONO NOMBRE_CORTO,  
  CELULAR NOMBRE_CORTO,  
  DIRECCION MEMO,  
  CONTACTO NOMBRE_CORTO,  
  CONTACTO_DATOS DESCRIPCION_CORTO);
```

```
ALTER TABLE PROVEEDOR ADD CONSTRAINT PK_PROVEEDOR PRIMARY KEY
(ID_PROVEEDOR);
```

```
/* Structure for the `TIPO_PRECIO` table : */
```

```
CREATE TABLE TIPO_PRECIO (
  ID_TIPO_PRECIO ENTERO NOT NULL,
  DESCRIPCION ALIAS);
```

```
ALTER TABLE TIPO_PRECIO ADD CONSTRAINT PK_TIPO_PRECIO PRIMARY KEY
(ID_TIPO_PRECIO);
```

```
/* Structure for the `VENTAAUX` table : */
```

```
CREATE TABLE VENTAAUX (
  ID_VENTAAUX ENTERO NOT NULL,
  ID_CLIENTE ENTERO,
  ID_VENDEDOR ENTERO,
  ID_PRODUCTO ENTERO,
  DESCRIPCIONPRO NOMBRE_LARGO,
  SALDOACTUAL NUMERO,
  PAGOS NUMERO,
  MONTOCUOTA NUMERO,
  ID_FORMA_PAGO ENTERO,
  FECHA FECHA,
  FORMA ALIAS,
  NROPAGARE ALIAS);
```

```
ALTER TABLE VENTAAUX ADD CONSTRAINT PK_VENTAAUX PRIMARY KEY
(ID_VENTAAUX);
```

```
/* Structure for the `VENTAMAYO` table : */
```

```
CREATE TABLE VENTAMAYO (
  ID_VENTAMAYO ENTERO NOT NULL,
  ID_CLIENTE ENTERO,
  ID_USUARIO ENTERO,
  ID_VENDEDOR ENTERO,
  FECHA FECHA,
  NROCOMPROBANTE ALIAS,
  CONFIRMADO SI_NO_NO NOT NULL,
  IMPRESO SI_NO_NO NOT NULL,
  TOTAL NUMERO,
  COSTO NUMERO,
  GANANCIA NUMERO,
  CONTADO SI_NO_NO NOT NULL,
  ID_PRECIO ENTERO);
```

```
ALTER TABLE VENTAMAYO ADD CONSTRAINT PK_VENTAMAYO PRIMARY KEY
(ID_VENTAMAYO);
```

/\* Structure for the `PEDIDO` table : \*/

```
CREATE TABLE PEDIDO (  
  ID_PEDIDO ENTERO NOT NULL,  
  ID_VENDEDOR ENTERO,  
  FECHA "DATE",  
  CANTIDAD NUMERO,  
  PRODUCTO DESCRIPCION_MEDIO,  
  ID_CLIENTE ENTERO NOT NULL,  
  ENTREGADO SI_NO_NO NOT NULL);
```

```
ALTER TABLE PEDIDO ADD CONSTRAINT PK_PEDIDO PRIMARY KEY (ID_PEDIDO);
```

/\* Structure for the `PREVENTA` table : \*/

```
CREATE TABLE PREVENTA (  
  IDPREVENTA ENTERO NOT NULL,  
  IDVENDEDOR ENTERO,  
  FECHA "DATE",  
  CANTIDAD NUMERO,  
  PRECIO NUMERO,  
  ENTREGA NUMERO,  
  CUOTA NUMERO,  
  FORMADEPAGO ENTERO,  
  NROCEDULA DESCRIPCION_CORTO,  
  NOMBRE DESCRIPTION_SHORT,  
  APELLIDO DESCRIPTION_SHORT,  
  CELULAR DESCRIPTION_SHORT,  
  PRODUCTO DESCRIPCION_MEDIO,  
  ID_VENTA ENTERO);
```

```
ALTER TABLE PREVENTA ADD CONSTRAINT PK_PREVENTA PRIMARY KEY (IDPREVENTA);
```

/\* Structure for the `CLIENTE\_SEGUI` table : \*/

```
CREATE TABLE CLIENTE_SEGUI (  
  ID_CLIENTE_SEGUI ENTERO NOT NULL,  
  ID_CLIENTE ENTERO,  
  ID_USUARIO ENTERO,  
  FECHA FECHA,  
  HORA HORA,  
  OBSERVACION DESCRIPTION_LARGE,  
  LATITUD DESCRIPCION_CORTO,  
  LONGITUD DESCRIPCION_CORTO);
```

```
ALTER TABLE CLIENTE_SEGUI ADD CONSTRAINT PK_CLIENTE_SEGUI PRIMARY KEY  
(ID_CLIENTE_SEGUI);
```

/\* Structure for the `UCTABPERFIL` table : \*/

```
CREATE TABLE UCTABPERFIL (  

```

```
ID_CONFIGURACION ENTERO NOT NULL,  
ID_USUARIO ENTERO,  
FORM DESCRIPTION_LARGE,  
GRILLA NOMBRE_CORTO,  
CAMPO NOMBRE_LARGO,  
VISIBLE SI_NO_NO NOT NULL,  
ANCHO ENTERO,  
POSICION ENTERO);
```

```
ALTER TABLE UCTABPERFIL ADD CONSTRAINT PK_UCTABPERFIL PRIMARY KEY  
(ID_CONFIGURACION);
```

```
/* Structure for the `COMPRAD` table : */
```

```
CREATE TABLE COMPRAD (  
    ID_COMPRAD ENTERO NOT NULL,  
    ID_COMPRA ENTERO,  
    ID_PRODUCTO NUMERO,  
    CANTIDAD NUMERO,  
    PRECIO NUMERO);
```

```
ALTER TABLE COMPRAD ADD CONSTRAINT PK_COMPRAD PRIMARY KEY (ID_COMPRAD);
```

```
/* Structure for the `MOTIVO_LLAMADA` table : */
```

```
CREATE TABLE MOTIVO_LLAMADA (  
    ID_MOTIVO_LLAMADA ENTERO NOT NULL,  
    DESCRIPCION NOMBRE_CORTO);
```

```
ALTER TABLE MOTIVO_LLAMADA ADD CONSTRAINT PK_MOTIVO_LLAMADA PRIMARY KEY  
(ID_MOTIVO_LLAMADA);
```

```
/* Structure for the `PRODUCTO_PRECIO_PROVE` table : */
```

```
CREATE TABLE PRODUCTO_PRECIO_PROVE (  
    ID_PRODUCTO_PRECIO_PROVE ENTERO NOT NULL,  
    ID_PRODUCTO ENTERO,  
    ID_PROVEEDOR ENTERO,  
    PRECIO NUMERO,  
    FECHA DATE,  
    ID_USUARIO ENTERO);
```

```
ALTER TABLE PRODUCTO_PRECIO_PROVE ADD CONSTRAINT PK_PRODUCTO_PRECIO_PROVE  
PRIMARY KEY (ID_PRODUCTO_PRECIO_PROVE);
```

```
/* Definition for the `SPREGISTROS` procedure : */
```

```
SET TERM ^ ;
```

```
CREATE OR ALTER PROCEDURE SPREGISTROS
```



```

RETURNS (
    TABLA VARCHAR(31),
    CANTIDAD INTEGER)
AS
Begin
    /* spRegistros. Procedimiento almacenado para obtener la cantidad de
       registros que contiene cada una de las tablas de la base de datos.
    */

    For Select RDB$Relation_Name From RDB$Relations
    Where (RDB$System_Flag = 0) And (RDB$View_BLR Is Null)
    Order By RDB$Relation_Name Into :Tabla Do
    Begin
        Execute Statement 'Select Count (*) From ' || :Tabla Into :Cantidad;
        Suspend;
    End
End^

SET TERM ; ^

/* Definition for the `FIB$BOOLEAN` domain : */

CREATE DOMAIN FIB$BOOLEAN AS
    SMALLINT
    DEFAULT 1
    NOT NULL
    CHECK (VALUE IN (0,1));

/* Structure for the `FIB$DATASETS_INFO` table : */

CREATE TABLE FIB$DATASETS_INFO (
    DS_ID INTEGER NOT NULL,
    DESCRIPTION VARCHAR(40),
    SELECT_SQL BLOB SUB_TYPE 1,
    UPDATE_SQL BLOB SUB_TYPE 1,
    INSERT_SQL BLOB SUB_TYPE 1,
    DELETE_SQL BLOB SUB_TYPE 1,
    REFRESH_SQL BLOB SUB_TYPE 1,
    NAME_GENERATOR VARCHAR(68),
    KEY_FIELD VARCHAR(68),
    UPDATE_TABLE_NAME VARCHAR(68),
    UPDATE_ONLY_MODIFIED_FIELDS FIB$BOOLEAN NOT NULL,
    CONDITIONS BLOB SUB_TYPE 1,
    FIB$VERSION INTEGER);

ALTER TABLE FIB$DATASETS_INFO ADD CONSTRAINT PK_FIB$DATASETS_INFO PRIMARY
KEY (DS_ID);

/* Definition for the `NROTEL` domain : */

CREATE DOMAIN NROTEL AS
    VARCHAR(20);

```

```

/* Definition for the `PRECIO` domain : */

CREATE DOMAIN PRECIO AS
    DOUBLE PRECISION
    DEFAULT 0;

/* Definition for the `SERIAL` domain : */

CREATE DOMAIN SERIAL AS
    VARCHAR(25);

/* Definition for the `YES_NO_NO` domain : */

CREATE DOMAIN YES_NO_NO AS
    VARCHAR(1)
    DEFAULT 'S'
    CHECK (VALUE IN ('S','N'));

/* Structure for the `EVENTO` table : */

CREATE TABLE EVENTO (
    IDUSER INTEGER,
    MSG VARCHAR(250),
    "DATA" VARCHAR(14),
    NIVEL INTEGER);

/* Structure for the `UCTABRIGHTS` table : */

CREATE TABLE UCTABRIGHTS (
    UCIDUSER INTEGER,
    UCMODULE VARCHAR(50),
    UCCOMPNAME VARCHAR(50),
    UCKEY VARCHAR(250));

/* Structure for the `UCTABRIGHTSEX` table : */

CREATE TABLE UCTABRIGHTSEX (
    UCIDUSER INTEGER,
    UCMODULE VARCHAR(50),
    UCCOMPNAME VARCHAR(50),
    UCFORMNAME VARCHAR(50),
    UCKEY VARCHAR(250));

/* Structure for the `UCTABUSERSLOGGED` table : */

CREATE TABLE UCTABUSERSLOGGED (
    UCIDLOGON CHAR(38),
    UCIDUSER INTEGER,
    UCAPPLICATIONID VARCHAR(50),
    UCMACHINENAME VARCHAR(50),
    UCDATA VARCHAR(14));

/* Definition for the `FIB$FIELD_INFO_VERSION` generator : */

```

```

CREATE GENERATOR FIB$FIELD_INFO_VERSION;

/* Definition for the `ID_MENSAJE_GEN` generator : */

CREATE GENERATOR ID_MENSAJE_GEN;

/* Definition for the `Auto Carga` exception : */

CREATE EXCEPTION "Auto Carga" 'No se ha definido transacción para esté
deposito';

/* Definition for the `Credito por comprobante` exception : */

CREATE EXCEPTION "Credito por comprobante" 'Está operación ya ha sido
registrado';

/* Definition for the `NUEVA_IMPRESION` exception : */

CREATE EXCEPTION NUEVA_IMPRESION 'Se ha realizado una nueva impresion';

/* Definition for the `ABS` UDF : */

DECLARE EXTERNAL FUNCTION "ABS"
    DOUBLE PRECISION
    RETURNS DOUBLE PRECISION BY VALUE
    ENTRY_POINT 'IB_UDF_abs' MODULE_NAME 'ib_udf';

/* Definition for the `ACOS` UDF : */

DECLARE EXTERNAL FUNCTION "ACOS"
    DOUBLE PRECISION
    RETURNS DOUBLE PRECISION BY VALUE
    ENTRY_POINT 'IB_UDF_acos' MODULE_NAME 'ib_udf';

/* Definition for the `ADDDAY` UDF : */

DECLARE EXTERNAL FUNCTION ADDDAY
    TIMESTAMP,
    INTEGER
    RETURNS TIMESTAMP
    ENTRY_POINT 'addDay' MODULE_NAME 'fbudf';

/* Definition for the `ADDDHOUR` UDF : */

DECLARE EXTERNAL FUNCTION ADDHOUR
    TIMESTAMP,
    INTEGER
    RETURNS TIMESTAMP
    ENTRY_POINT 'addHour' MODULE_NAME 'fbudf';

/* Definition for the `ADDMILLISECOND` UDF : */

DECLARE EXTERNAL FUNCTION ADDMILLISECOND
    TIMESTAMP,

```

```

    INTEGER
    RETURNS TIMESTAMP
    ENTRY_POINT 'addMilliSecond' MODULE_NAME 'fbudf';

/* Definition for the `ADDMINUTE` UDF : */

DECLARE EXTERNAL FUNCTION ADDMINUTE
    TIMESTAMP,
    INTEGER
    RETURNS TIMESTAMP
    ENTRY_POINT 'addMinute' MODULE_NAME 'fbudf';

/* Definition for the `ADDMONTH` UDF : */

DECLARE EXTERNAL FUNCTION ADDMONTH
    TIMESTAMP,
    INTEGER
    RETURNS TIMESTAMP
    ENTRY_POINT 'addMonth' MODULE_NAME 'fbudf';

/* Definition for the `ADDSECOND` UDF : */

DECLARE EXTERNAL FUNCTION ADDSECOND
    TIMESTAMP,
    INTEGER
    RETURNS TIMESTAMP
    ENTRY_POINT 'addSecond' MODULE_NAME 'fbudf';

/* Definition for the `ADDWEEK` UDF : */

DECLARE EXTERNAL FUNCTION ADDWEEK
    TIMESTAMP,
    INTEGER
    RETURNS TIMESTAMP
    ENTRY_POINT 'addWeek' MODULE_NAME 'fbudf';

/* Definition for the `ADDYEAR` UDF : */

DECLARE EXTERNAL FUNCTION ADDYEAR
    TIMESTAMP,
    INTEGER
    RETURNS TIMESTAMP
    ENTRY_POINT 'addYear' MODULE_NAME 'fbudf';

/* Definition for the `ASCII_CHAR` UDF : */

DECLARE EXTERNAL FUNCTION "ASCII_CHAR"
    INTEGER
    RETURNS CSTRING(1) FREE_IT
    ENTRY_POINT 'IB_UDF_ascii_char' MODULE_NAME 'ib_udf';

/* Definition for the `ASCII_VAL` UDF : */

DECLARE EXTERNAL FUNCTION "ASCII_VAL"

```

```

    CHAR(1)
    RETURNS INTEGER BY VALUE
    ENTRY_POINT 'IB_UDF_ascii_val' MODULE_NAME 'ib_udf';

/* Definition for the `ASIN` UDF : */

DECLARE EXTERNAL FUNCTION "ASIN"
    DOUBLE PRECISION
    RETURNS DOUBLE PRECISION BY VALUE
    ENTRY_POINT 'IB_UDF_asin' MODULE_NAME 'ib_udf';

/* Definition for the `ATAN` UDF : */

DECLARE EXTERNAL FUNCTION "ATAN"
    DOUBLE PRECISION
    RETURNS DOUBLE PRECISION BY VALUE
    ENTRY_POINT 'IB_UDF_atan' MODULE_NAME 'ib_udf';

/* Definition for the `ATAN2` UDF : */

DECLARE EXTERNAL FUNCTION "ATAN2"
    DOUBLE PRECISION,
    DOUBLE PRECISION
    RETURNS DOUBLE PRECISION BY VALUE
    ENTRY_POINT 'IB_UDF_atan2' MODULE_NAME 'ib_udf';

/* Definition for the `BIN_AND` UDF : */

DECLARE EXTERNAL FUNCTION "BIN_AND"
    INTEGER,
    INTEGER
    RETURNS INTEGER BY VALUE
    ENTRY_POINT 'IB_UDF_bin_and' MODULE_NAME 'ib_udf';

/* Definition for the `BIN_OR` UDF : */

DECLARE EXTERNAL FUNCTION "BIN_OR"
    INTEGER,
    INTEGER
    RETURNS INTEGER BY VALUE
    ENTRY_POINT 'IB_UDF_bin_or' MODULE_NAME 'ib_udf';

/* Definition for the `BIN_XOR` UDF : */

DECLARE EXTERNAL FUNCTION "BIN_XOR"
    INTEGER,
    INTEGER
    RETURNS INTEGER BY VALUE
    ENTRY_POINT 'IB_UDF_bin_xor' MODULE_NAME 'ib_udf';

/* Definition for the `COS` UDF : */

DECLARE EXTERNAL FUNCTION "COS"
    DOUBLE PRECISION

```

```

RETURNS DOUBLE PRECISION BY VALUE
ENTRY_POINT 'IB_UDF_cos' MODULE_NAME 'ib_udf';

/* Definition for the `COSH` UDF : */

DECLARE EXTERNAL FUNCTION "COSH"
    DOUBLE PRECISION
RETURNS DOUBLE PRECISION BY VALUE
ENTRY_POINT 'IB_UDF_cosh' MODULE_NAME 'ib_udf';

/* Definition for the `COT` UDF : */

DECLARE EXTERNAL FUNCTION "COT"
    DOUBLE PRECISION
RETURNS DOUBLE PRECISION BY VALUE
ENTRY_POINT 'IB_UDF_cot' MODULE_NAME 'ib_udf';

/* Definition for the `DIV` UDF : */

DECLARE EXTERNAL FUNCTION DIV
    INTEGER,
    INTEGER
RETURNS DOUBLE PRECISION BY VALUE
ENTRY_POINT 'IB_UDF_div' MODULE_NAME 'ib_udf';

/* Definition for the `DNULLIF` UDF : */

DECLARE EXTERNAL FUNCTION DNULLIF
    DOUBLE PRECISION BY DESCRIPTOR,
    DOUBLE PRECISION BY DESCRIPTOR
RETURNS DOUBLE PRECISION BY DESCRIPTOR
ENTRY_POINT 'dNullIf' MODULE_NAME 'fbudf';

/* Definition for the `DNVL` UDF : */

DECLARE EXTERNAL FUNCTION DNVL
    DOUBLE PRECISION BY DESCRIPTOR,
    DOUBLE PRECISION BY DESCRIPTOR
RETURNS DOUBLE PRECISION BY DESCRIPTOR
ENTRY_POINT 'idNvl' MODULE_NAME 'fbudf';

/* Definition for the `DOW` UDF : */

DECLARE EXTERNAL FUNCTION DOW
    TIMESTAMP,
    VARCHAR(15)
RETURNS PARAMETER 2
ENTRY_POINT 'DOW' MODULE_NAME 'fbudf';

/* Definition for the `I64NULLIF` UDF : */

DECLARE EXTERNAL FUNCTION I64NULLIF
    NUMERIC(18, 4) BY DESCRIPTOR,
    NUMERIC(18, 4) BY DESCRIPTOR

```

```

RETURNS NUMERIC(18, 4) BY DESCRIPTOR
ENTRY_POINT 'iNullIf' MODULE_NAME 'fbudf';

/* Definition for the `I64NVL` UDF : */

DECLARE EXTERNAL FUNCTION I64NVL
    NUMERIC(18, 0) BY DESCRIPTOR,
    NUMERIC(18, 0) BY DESCRIPTOR
RETURNS NUMERIC(18, 0) BY DESCRIPTOR
ENTRY_POINT 'idNvl' MODULE_NAME 'fbudf';

/* Definition for the `INULLIF` UDF : */

DECLARE EXTERNAL FUNCTION INULLIF
    INTEGER BY DESCRIPTOR,
    INTEGER BY DESCRIPTOR
RETURNS INTEGER BY DESCRIPTOR
ENTRY_POINT 'iNullIf' MODULE_NAME 'fbudf';

/* Definition for the `INVL` UDF : */

DECLARE EXTERNAL FUNCTION INVL
    INTEGER BY DESCRIPTOR,
    INTEGER BY DESCRIPTOR
RETURNS INTEGER BY DESCRIPTOR
ENTRY_POINT 'idNvl' MODULE_NAME 'fbudf';

/* Definition for the `LN` UDF : */

DECLARE EXTERNAL FUNCTION "LN"
    DOUBLE PRECISION
RETURNS DOUBLE PRECISION BY VALUE
ENTRY_POINT 'IB_UDF_ln' MODULE_NAME 'ib_udf';

/* Definition for the `LOG` UDF : */

DECLARE EXTERNAL FUNCTION "LOG"
    DOUBLE PRECISION,
    DOUBLE PRECISION
RETURNS DOUBLE PRECISION BY VALUE
ENTRY_POINT 'IB_UDF_log' MODULE_NAME 'ib_udf';

/* Definition for the `LOG10` UDF : */

DECLARE EXTERNAL FUNCTION "LOG10"
    DOUBLE PRECISION
RETURNS DOUBLE PRECISION BY VALUE
ENTRY_POINT 'IB_UDF_log10' MODULE_NAME 'ib_udf';

/* Definition for the `LOWER` UDF : */

DECLARE EXTERNAL FUNCTION "LOWER"
    CSTRING(80)
RETURNS CSTRING(80) FREE_IT

```

```

ENTRY_POINT 'IB_UDF_lower' MODULE_NAME 'ib_udf';

/* Definition for the `LPAD` UDF : */

DECLARE EXTERNAL FUNCTION "LPAD"
  CSTRING(80),
  INTEGER,
  CSTRING(1)
RETURNS CSTRING(80) FREE_IT
ENTRY_POINT 'IB_UDF_lpad' MODULE_NAME 'ib_udf';

/* Definition for the `LTRIM` UDF : */

DECLARE EXTERNAL FUNCTION LTRIM
  CSTRING(80)
RETURNS CSTRING(80) FREE_IT
ENTRY_POINT 'IB_UDF_ltrim' MODULE_NAME 'ib_udf';

/* Definition for the `MOD` UDF : */

DECLARE EXTERNAL FUNCTION "MOD"
  INTEGER,
  INTEGER
RETURNS DOUBLE PRECISION BY VALUE
ENTRY_POINT 'IB_UDF_mod' MODULE_NAME 'ib_udf';

/* Definition for the `PI` UDF : */

DECLARE EXTERNAL FUNCTION "PI"
RETURNS DOUBLE PRECISION BY VALUE
ENTRY_POINT 'IB_UDF_pi' MODULE_NAME 'ib_udf';

/* Definition for the `RAND` UDF : */

DECLARE EXTERNAL FUNCTION "RAND"
RETURNS DOUBLE PRECISION BY VALUE
ENTRY_POINT 'IB_UDF_rand' MODULE_NAME 'ib_udf';

/* Definition for the `RPAD` UDF : */

DECLARE EXTERNAL FUNCTION "RPAD"
  CSTRING(80),
  INTEGER,
  CSTRING(1)
RETURNS CSTRING(80) FREE_IT
ENTRY_POINT 'IB_UDF_rpad' MODULE_NAME 'ib_udf';

/* Definition for the `RTRIM` UDF : */

DECLARE EXTERNAL FUNCTION RTRIM
  CSTRING(80)
RETURNS CSTRING(80) FREE_IT
ENTRY_POINT 'IB_UDF_rtrim' MODULE_NAME 'ib_udf';

```



```

/* Definition for the `SDOW` UDF : */

DECLARE EXTERNAL FUNCTION SDOW
    TIMESTAMP,
    VARCHAR(5)
RETURNS PARAMETER 2
ENTRY_POINT 'SDOW' MODULE_NAME 'fbudf';

/* Definition for the `SIGN` UDF : */

DECLARE EXTERNAL FUNCTION "SIGN"
    DOUBLE PRECISION
RETURNS INTEGER BY VALUE
ENTRY_POINT 'IB_UDF_sign' MODULE_NAME 'ib_udf';

/* Definition for the `SIN` UDF : */

DECLARE EXTERNAL FUNCTION "SIN"
    DOUBLE PRECISION
RETURNS DOUBLE PRECISION BY VALUE
ENTRY_POINT 'IB_UDF_sin' MODULE_NAME 'ib_udf';

/* Definition for the `SINH` UDF : */

DECLARE EXTERNAL FUNCTION "SINH"
    DOUBLE PRECISION
RETURNS DOUBLE PRECISION BY VALUE
ENTRY_POINT 'IB_UDF_sinh' MODULE_NAME 'ib_udf';

/* Definition for the `SNULLIF` UDF : */

DECLARE EXTERNAL FUNCTION SNULLIF
    VARCHAR(100) BY DESCRIPTOR,
    VARCHAR(100) BY DESCRIPTOR,
    VARCHAR(100) BY DESCRIPTOR
RETURNS PARAMETER 3
ENTRY_POINT 'sNullIf' MODULE_NAME 'fbudf';

/* Definition for the `SNVL` UDF : */

DECLARE EXTERNAL FUNCTION SNVL
    VARCHAR(100) BY DESCRIPTOR,
    VARCHAR(100) BY DESCRIPTOR,
    VARCHAR(100) BY DESCRIPTOR
RETURNS PARAMETER 3
ENTRY_POINT 'sNvl' MODULE_NAME 'fbudf';

/* Definition for the `SQRT` UDF : */

DECLARE EXTERNAL FUNCTION "SQRT"
    DOUBLE PRECISION
RETURNS DOUBLE PRECISION BY VALUE
ENTRY_POINT 'IB_UDF_sqrt' MODULE_NAME 'ib_udf';

```

```

/* Definition for the `SRIGHT` UDF : */

DECLARE EXTERNAL FUNCTION SRIGHT
  VARCHAR(100) BY DESCRIPTOR,
  SMALLINT,
  VARCHAR(100) BY DESCRIPTOR
RETURNS PARAMETER 3
ENTRY_POINT 'right' MODULE_NAME 'fbudf';

/* Definition for the `STRLEN` UDF : */

DECLARE EXTERNAL FUNCTION STRLEN
  CSTRING(32767)
RETURNS INTEGER BY VALUE
ENTRY_POINT 'IB_UDF_strlen' MODULE_NAME 'ib_udf';

/* Definition for the `SUBSTR` UDF : */

DECLARE EXTERNAL FUNCTION SUBSTR
  CSTRING(80),
  SMALLINT,
  SMALLINT
RETURNS CSTRING(80) FREE_IT
ENTRY_POINT 'IB_UDF_substr' MODULE_NAME 'ib_udf';

/* Definition for the `SUBSTRLEN` UDF : */

DECLARE EXTERNAL FUNCTION SUBSTRLEN
  CSTRING(80),
  SMALLINT,
  SMALLINT
RETURNS CSTRING(80) FREE_IT
ENTRY_POINT 'IB_UDF_substrlen' MODULE_NAME 'ib_udf';

/* Definition for the `TAN` UDF : */

DECLARE EXTERNAL FUNCTION "TAN"
  DOUBLE PRECISION
RETURNS DOUBLE PRECISION BY VALUE
ENTRY_POINT 'IB_UDF_tan' MODULE_NAME 'ib_udf';

/* Definition for the `TANH` UDF : */

DECLARE EXTERNAL FUNCTION "TANH"
  DOUBLE PRECISION
RETURNS DOUBLE PRECISION BY VALUE
ENTRY_POINT 'IB_UDF_tanh' MODULE_NAME 'ib_udf';

/* Data for the 'VENTAD' table (Records 1 - 2) */

INSERT INTO VENTAD
VALUES (2, 2, 2, 1, 280000, 150000);

INSERT INTO VENTAD

```

```
VALUES (3, 3, 37, 1, 280000, 0);
```

```
COMMIT WORK;
```

```
/* Data for the 'GRUPO_CLIENTE' table (Records 1 - 8) */
```

```
INSERT INTO GRUPO_CLIENTE  
VALUES (7, 'InformConf', 6);
```

```
INSERT INTO GRUPO_CLIENTE  
VALUES (14, 'Local', NULL);
```

```
INSERT INTO GRUPO_CLIENTE  
VALUES (8, 'Zona 1', NULL);
```

```
INSERT INTO GRUPO_CLIENTE  
VALUES (9, 'Zona 2', NULL);
```

```
INSERT INTO GRUPO_CLIENTE  
VALUES (10, 'Zona 3', NULL);
```

```
INSERT INTO GRUPO_CLIENTE  
VALUES (11, 'Zona 4', NULL);
```

```
INSERT INTO GRUPO_CLIENTE  
VALUES (12, 'Zona 5', NULL);
```

```
INSERT INTO GRUPO_CLIENTE  
VALUES (13, 'SISCO', NULL);
```

```
COMMIT WORK;
```

```
/* Data for the 'PAIS' table (Records 1 - 3) */
```

```
INSERT INTO PAIS  
VALUES (1, 'Paraguay', 'Paraguaya', '592');
```

```
INSERT INTO PAIS  
VALUES (2, 'Argentina', 'Argentino', '002');
```

```
INSERT INTO PAIS  
VALUES (3, 'Brasil', 'Brasilero', '0');
```

```
COMMIT WORK;
```

```
/* Data for the 'CIUDAD' table (Records 1 - 185) */
```

```
INSERT INTO CIUDAD  
VALUES (1, 'Asunción', '021', 1);
```

```
INSERT INTO CIUDAD  
VALUES (2, 'Luque', '023', 1);
```

```
INSERT INTO CIUDAD
```

```
VALUES (3, 'Itá', '024', 1);

INSERT INTO CIUDAD
VALUES (4, 'Villeta', '025', 1);

INSERT INTO CIUDAD
VALUES (5, 'Villa Hayes', '026', 1);

INSERT INTO CIUDAD
VALUES (6, 'Ypane', '0275', 1);

INSERT INTO CIUDAD
VALUES (7, 'Capiatá', '028', 1);

INSERT INTO CIUDAD
VALUES (8, 'Areguá', '0291', 1);

INSERT INTO CIUDAD
VALUES (9, 'Guarambaré', '0293', 1);

INSERT INTO CIUDAD
VALUES (10, 'Nueva Italia', '02938', 1);

INSERT INTO CIUDAD
VALUES (11, 'Itaugúa', '02942', 1);

INSERT INTO CIUDAD
VALUES (12, 'Limpio', '0297', 1);

INSERT INTO CIUDAD
VALUES (13, 'Concepción', '031', 1);

INSERT INTO CIUDAD
VALUES (14, 'Horqueta', '032', 1);

INSERT INTO CIUDAD
VALUES (15, 'Vallemí', '0351', 1);

INSERT INTO CIUDAD
VALUES (16, 'Pedro Juan Caballero', '036', 1);

INSERT INTO CIUDAD
VALUES (17, 'Alberdi', '0780', 1);

INSERT INTO CIUDAD
VALUES (18, 'Capitán Bado', '037-', 1);

INSERT INTO CIUDAD
VALUES (19, 'Bella Vista del Norte', '038', 1);

INSERT INTO CIUDAD
VALUES (20, 'C', '039', 1);

INSERT INTO CIUDAD
```

```
VALUES (21, 'Itacurubí del Rosario', '041', 1);

INSERT INTO CIUDAD
VALUES (22, 'San Estanislao', '04320', 1);

INSERT INTO CIUDAD
VALUES (23, 'General Aquino', '0438', 1);

INSERT INTO CIUDAD
VALUES (24, 'Villa del Rosario', '044', 1);

INSERT INTO CIUDAD
VALUES (25, 'Salto del Guairá', '046', 1);

INSERT INTO CIUDAD
VALUES (26, 'Puente Kyjhá', '047', 1);

INSERT INTO CIUDAD
VALUES (27, 'Caacupé', '0511', 1);

INSERT INTO CIUDAD
VALUES (28, 'San Bernardino', '0512', 1);

INSERT INTO CIUDAD
VALUES (29, 'Ypacarí', '0513', 1);

INSERT INTO CIUDAD
VALUES (30, 'Eusebio Ayala', '0514', 1);

INSERT INTO CIUDAD
VALUES (31, 'Piribebuy', '0515', 1);

INSERT INTO CIUDAD
VALUES (32, 'Tobatí', '0516', 1);

INSERT INTO CIUDAD
VALUES (33, 'Itacurubí de la Cordillera', '0518', 1);

INSERT INTO CIUDAD
VALUES (34, 'Coronel Oviedo', '0521', 1);

INSERT INTO CIUDAD
VALUES (35, 'Caaguazú', '0522', 1);

INSERT INTO CIUDAD
VALUES (36, 'Juan Manuel Frutos (Pastoreo)', '0524', 1);

INSERT INTO CIUDAD
VALUES (37, 'José Domingo Ocampo', '0527', 1);

INSERT INTO CIUDAD
VALUES (39, 'Paraguarí', '0531', 1);

INSERT INTO CIUDAD
```

```
VALUES (40, 'Carapeguá', '0532', 1);

INSERT INTO CIUDAD
VALUES (41, 'Yaguarón', '0533', 1);

INSERT INTO CIUDAD
VALUES (42, 'Ybycuí', '0534', 1);

INSERT INTO CIUDAD
VALUES (43, 'Quiindy', '0536', 1);

INSERT INTO CIUDAD
VALUES (44, 'La Colmena', '0537', 1);

INSERT INTO CIUDAD
VALUES (45, 'Villarrica', '0541', 1);

INSERT INTO CIUDAD
VALUES (46, 'Iturbe', '0546', 1);

INSERT INTO CIUDAD
VALUES (47, 'Yuty', '0547', 1);

INSERT INTO CIUDAD
VALUES (48, 'Fulgencio Yegros', '0545', 1);

INSERT INTO CIUDAD
VALUES (49, 'Independecia', '0548', 1);

INSERT INTO CIUDAD
VALUES (50, 'Hernandarias', '061', 1);

INSERT INTO CIUDAD
VALUES (51, 'Puerto Pdte. Franco', '061', 1);

INSERT INTO CIUDAD
VALUES (52, 'Ciudad del Este', '0631', 1);

INSERT INTO CIUDAD
VALUES (53, 'Colonia Yguazú', '0632', 1);

INSERT INTO CIUDAD
VALUES (54, 'Yguazú', '0632-', 1);

INSERT INTO CIUDAD
VALUES (55, 'Minga Guazú', '0644', 1);

INSERT INTO CIUDAD
VALUES (56, 'Santa Rita', '0673', 1);

INSERT INTO CIUDAD
VALUES (57, 'Juan León Mallorquín', '0675', 1);

INSERT INTO CIUDAD
```

```
VALUES (58, 'Encarnación', '071', 1);

INSERT INTO CIUDAD
VALUES (59, 'Ayolas', '072', 1);

INSERT INTO CIUDAD
VALUES (60, 'San Cosme', '073', 1);

INSERT INTO CIUDAD
VALUES (61, 'General Delgado', '0740', 1);

INSERT INTO CIUDAD
VALUES (62, 'Coronel Bogado', '0741', 1);

INSERT INTO CIUDAD
VALUES (63, 'Hohenau', '075', 1);

INSERT INTO CIUDAD
VALUES (64, 'Fram', '0761', 1);

INSERT INTO CIUDAD
VALUES (65, 'Carmen del Paraná', '0762', 1);

INSERT INTO CIUDAD
VALUES (66, 'María Auxiliadora', '0764', 1);

INSERT INTO CIUDAD
VALUES (67, 'Bella Vista del Sur', '0767', 1);

INSERT INTO CIUDAD
VALUES (68, 'Colonia Pirapó', '0768', 1);

INSERT INTO CIUDAD
VALUES (69, 'San Juan Bautista Misiones', '081', 1);

INSERT INTO CIUDAD
VALUES (70, 'San Ignacio', '082', 1);

INSERT INTO CIUDAD
VALUES (71, 'Villa Florida', '083', 1);

INSERT INTO CIUDAD
VALUES (72, 'Santa Rosa (Misiones)', '0858', 1);

INSERT INTO CIUDAD
VALUES (74, 'Pilar', '086', 1);

INSERT INTO CIUDAD
VALUES (75, 'San Lorenzo', '58', 1);

INSERT INTO CIUDAD
VALUES (76, 'Mariano Roque Alonso', '75', 1);

INSERT INTO CIUDAD
```

```
VALUES (77, 'Ñemby', '960', 1);

INSERT INTO CIUDAD
VALUES (78, 'San Antonio', '990', 1);

INSERT INTO CIUDAD
VALUES (79, 'Benjamín Aceval', '0271', 1);

INSERT INTO CIUDAD
VALUES (80, 'C.P.C. - Asunción', '0941', 1);

INSERT INTO CIUDAD
VALUES (84, 'Tebicuary', '0553', 1);

INSERT INTO CIUDAD
VALUES (90, 'Neuland', '0493', 1);

INSERT INTO CIUDAD
VALUES (91, 'Filadelfia', '0491', 1);

INSERT INTO CIUDAD
VALUES (92, 'Capiibary', '0453', 1);

INSERT INTO CIUDAD
VALUES (93, 'Loma Plata', '0492', 1);

INSERT INTO CIUDAD
VALUES (96, 'Buenos Aires', '11', 1);

INSERT INTO CIUDAD
VALUES (97, 'Santiago', '2', 1);

INSERT INTO CIUDAD
VALUES (98, 'Concepción', 'SIMBO', 1);

INSERT INTO CIUDAD
VALUES (99, 'Mato Grosso do Sur', '67', 1);

INSERT INTO CIUDAD
VALUES (100, 'Mato Grosso', '66', 1);

INSERT INTO CIUDAD
VALUES (104, 'Savigliano', '172', 1);

INSERT INTO CIUDAD
VALUES (105, 'Saluzzo', '0175', 1);

INSERT INTO CIUDAD
VALUES (106, 'Temuco', '45', 1);

INSERT INTO CIUDAD
VALUES (107, 'Osono', '64', 1);

INSERT INTO CIUDAD
```



```
VALUES (108, 'Paysandú', '747', 1);

INSERT INTO CIUDAD
VALUES (109, 'Paisandú', '72', 1);

INSERT INTO CIUDAD
VALUES (110, 'Los Andes', '34', 1);

INSERT INTO CIUDAD
VALUES (111, 'Bra', '172', 1);

INSERT INTO CIUDAD
VALUES (112, 'Clorinda', '298', 1);

INSERT INTO CIUDAD
VALUES (115, 'Alba', '173', 1);

INSERT INTO CIUDAD
VALUES (116, 'Nusco', '827', 1);

INSERT INTO CIUDAD
VALUES (118, 'Cuneo', '171', 1);

INSERT INTO CIUDAD
VALUES (119, 'Posadas', '03752', 1);

INSERT INTO CIUDAD
VALUES (121, 'Ubatuba', '12', 1);

INSERT INTO CIUDAD
VALUES (122, 'Ubatuba', '19', 1);

INSERT INTO CIUDAD
VALUES (124, 'Foz de Yguazú', '045', 1);

INSERT INTO CIUDAD
VALUES (125, 'Montevideo', '02', 1);

INSERT INTO CIUDAD
VALUES (126, 'Puerto Rico', '3743', 1);

INSERT INTO CIUDAD
VALUES (127, 'Soca', '37', 1);

INSERT INTO CIUDAD
VALUES (128, 'La Paz', '22', 1);

INSERT INTO CIUDAD
VALUES (129, 'Santa Cruz', '719', 1);

INSERT INTO CIUDAD
VALUES (130, 'Altavilla Vicentina', '0444', 1);

INSERT INTO CIUDAD
```

```
VALUES (133, 'Roma', '1', 1);

INSERT INTO CIUDAD
VALUES (134, 'Ciudad de México (Dist. Federal)', '11', 1);

INSERT INTO CIUDAD
VALUES (135, 'Cochabamba', '42', 1);

INSERT INTO CIUDAD
VALUES (270, 'Lambaré', '021', 1);

INSERT INTO CIUDAD
VALUES (271, 'Miami', '305', 1);

INSERT INTO CIUDAD
VALUES (272, 'Los Angeles', '2', 1);

INSERT INTO CIUDAD
VALUES (273, 'Washington', '1', 1);

INSERT INTO CIUDAD
VALUES (274, '3 de mayo', '05478', 1);

INSERT INTO CIUDAD
VALUES (275, 'Acahay', '0535', 1);

INSERT INTO CIUDAD
VALUES (276, 'Arroyos y Esteros', '0516', 1);

INSERT INTO CIUDAD
VALUES (277, 'Atyra', '0520', 1);

INSERT INTO CIUDAD
VALUES (278, 'Bella Vista del Sur (Celular Rural)', '0873', 1);

INSERT INTO CIUDAD
VALUES (279, 'Caapucú', '05368', 1);

INSERT INTO CIUDAD
VALUES (280, 'Caazapá', '0542', 1);

INSERT INTO CIUDAD
VALUES (281, 'Caballero', '0526', 1);

INSERT INTO CIUDAD
VALUES (282, 'Capitán Miranda', '071', 1);

INSERT INTO CIUDAD
VALUES (283, 'Capitán Miranda (Celular Rural)', '0873', 1);

INSERT INTO CIUDAD
VALUES (284, 'Caraguatay', '0517', 1);

INSERT INTO CIUDAD
```

```
VALUES (285, 'Cedrales', '0633', 1);

INSERT INTO CIUDAD
VALUES (286, 'Colonia Lolita', '0424', 1);

INSERT INTO CIUDAD
VALUES (287, 'Colonia Naranjal', '0676', 1);

INSERT INTO CIUDAD
VALUES (288, 'Colonia Para Todo', '0425', 1);

INSERT INTO CIUDAD
VALUES (289, 'Santa Rosa del Monday', '0678', 1);

INSERT INTO CIUDAD
VALUES (290, 'Colonia Volendam', '0451', 1);

INSERT INTO CIUDAD
VALUES (291, 'Emboscada', '0529', 1);

INSERT INTO CIUDAD
VALUES (292, 'General Artigas', '0743', 1);

INSERT INTO CIUDAD
VALUES (293, 'General Diaz', '0787', 1);

INSERT INTO CIUDAD
VALUES (294, 'J. Augusto Saldivar', '0295', 1);

INSERT INTO CIUDAD
VALUES (296, 'Juan E. O'''Leary', '0674', 1);

INSERT INTO CIUDAD
VALUES (297, 'Isla Pucu', '0525', 1);

INSERT INTO CIUDAD
VALUES (298, 'Kaa Rendy', '0873', 1);

INSERT INTO CIUDAD
VALUES (299, 'Kresburgo', '0672', 1);

INSERT INTO CIUDAD
VALUES (300, 'La Paz', '0763', 1);

INSERT INTO CIUDAD
VALUES (301, 'La Paz (Celular Rural)', '0873', 1);

INSERT INTO CIUDAD
VALUES (302, 'Mauricio José Troche', '0550', 1);

INSERT INTO CIUDAD
VALUES (303, 'Mayor Otaño', '0671', 1);

INSERT INTO CIUDAD
```

```
VALUES (304, 'Loreto', '033', 1);

INSERT INTO CIUDAD
VALUES (305, 'Maciel', '05428', 1);

INSERT INTO CIUDAD
VALUES (306, 'Mariscal Estigarribia', '0494', 1);

INSERT INTO CIUDAD
VALUES (307, 'Obligado', '0717', 1);

INSERT INTO CIUDAD
VALUES (308, 'Paso de Patria', '0785', 1);

INSERT INTO CIUDAD
VALUES (309, 'Paso Yobai', '0552', 1);

INSERT INTO CIUDAD
VALUES (310, 'Pirayu', '0519', 1);

INSERT INTO CIUDAD
VALUES (311, 'Pozo Colorado', '0493', 1);

INSERT INTO CIUDAD
VALUES (312, 'Roque Gonzalez de Santa Cruz', '0538', 1);

INSERT INTO CIUDAD
VALUES (313, 'San Alberto', '0677', 1);

INSERT INTO CIUDAD
VALUES (314, 'San José de los Arroyos', '0523', 1);

INSERT INTO CIUDAD
VALUES (315, 'San Juan Neembucu', '0784', 1);

INSERT INTO CIUDAD
VALUES (316, 'San Juan Nepomuceno', '0544', 1);

INSERT INTO CIUDAD
VALUES (317, 'San Miguel Misiones', '081', 1);

INSERT INTO CIUDAD
VALUES (318, 'Santiago', '0782', 1);

INSERT INTO CIUDAD
VALUES (319, 'Sapucaí', '0539', 1);

INSERT INTO CIUDAD
VALUES (320, 'Curuguaty', '048', 1);

INSERT INTO CIUDAD
VALUES (321, 'San Pedro', '042', 1);

INSERT INTO CIUDAD
```

```
VALUES (222, 'San Joaquin', '0522', 1);

INSERT INTO CIUDAD
VALUES (322, 'Yataity', '0549', 1);

INSERT INTO CIUDAD
VALUES (323, 'Fernando de la Mora', '021', 1);

INSERT INTO CIUDAD
VALUES (324, 'VilaJapao', '011', 1);

INSERT INTO CIUDAD
VALUES (325, 'Brasilia (DF)', '61', 3);

INSERT INTO CIUDAD
VALUES (326, 'Curitiva (PR)', '41', 3);

INSERT INTO CIUDAD
VALUES (327, 'Blumenau (SC)', '47', 3);

INSERT INTO CIUDAD
VALUES (328, 'Porto Alegre (RS)', '51', 3);

INSERT INTO CIUDAD
VALUES (329, 'Jaragua do Sul (SC)', '47', 3);

INSERT INTO CIUDAD
VALUES (330, 'Ourinhos (SP)', '14', 3);

INSERT INTO CIUDAD
VALUES (331, 'Sao Ludgrero', '48', 3);

INSERT INTO CIUDAD
VALUES (334, 'Fernando de la Mora', '021', 1);

INSERT INTO CIUDAD
VALUES (335, 'Trinidad', '021', 1);

INSERT INTO CIUDAD
VALUES (336, 'J. E. Estigarribia', '0528', 1);

INSERT INTO CIUDAD
VALUES (337, 'Cruce Liberación', '00', 1);

INSERT INTO CIUDAD
VALUES (338, 'Campo 9', '0528', 1);

INSERT INTO CIUDAD
VALUES (339, 'pastoreo', '05271', 1);

INSERT INTO CIUDAD
VALUES (340, 'MALLORQUIN', '014', 1);

INSERT INTO CIUDAD
```

```
VALUES (341, 'OLEARY', '210', 1);
```

```
INSERT INTO CIUDAD  
VALUES (342, 'KM60', '210', 1);
```

```
INSERT INTO CIUDAD  
VALUES (343, 'Santa Ni', '00', 1);
```

```
COMMIT WORK;
```

```
/* Data for the 'CLIENTE' table (Records 1 - 500) */
```

```
INSERT INTO CLIENTE (AI_CLIENTE, ID_GRUPO_CLIENTE, ID_CIUADAD, NOMBRE,  
APELLIDO, TELEFONO, DIRECCION, RUC, MEMO, BARRIO, CONTACTO, CONTACTO2,  
DOCUMENTO_1, DOCUMENTO_2, CELULAR, TELEFONO_1, TELEFONO_2, ORDEN_VISITA,  
EMAIL, OBSERVACION, CALIFICACION, ACTIVO, RECIBIRPEDIDO, LATITUD,  
LONGITUD, FECHA_NACIMIENTO, DIRECCIONTRABAJO, ID_VENDEDOR, LUGARTRABAJO,  
LATITUD1, LONGITUD1, FEC_NAC)  
VALUES (346, 8, 56, 'ARNALDO', 'NUÑEZ ROJAS', NULL, '', '6330876', NULL,  
'Tavapy', NULL, NULL, NULL, NULL, '0993327621', NULL, NULL, 165, NULL,  
NULL, NULL, 1, 'N', NULL, NULL, '1994-04-01', NULL, NULL, 'Metal Fibra',  
NULL, NULL, NULL);
```

```
COMMIT WORK;
```

```
/* Data for the 'CALIFICACION' table (Records 1 - 4) */
```

```
INSERT INTO CALIFICACION  
VALUES (1, 'Excelente', 0, 45, 'E');
```

```
INSERT INTO CALIFICACION  
VALUES (2, 'Moroso', 46, 90, 'M');
```

```
INSERT INTO CALIFICACION  
VALUES (3, 'Moroso Grave', 91, 180, 'MG');
```

```
INSERT INTO CALIFICACION  
VALUES (4, 'Demanda', 181, 999999, 'Dem');
```

```
COMMIT WORK;
```

```
/* Data for the 'VENDEDOR' table (Records 1 - 9) */
```

```
INSERT INTO VENDEDOR  
VALUES (2, 'Orlando Javier', 'Martínez', 'Orué', '+595972156371', 'ojmo-  
pbw@outlook.com', 'Calle San Miguel 105', '2309387-1', '1977-03-10',  
NULL, 'S', 1, NULL);
```

```
INSERT INTO VENDEDOR
```

```
VALUES (1, 'Rodrigo Ariel', 'Martínez', 'Orué', '+595972260891',  
'pirineta@hotmail.es', 'Calle San Miguel', '4974638', '1991-08-26', NULL,  
'S', NULL, NULL);
```

```
INSERT INTO VENDEDOR  
VALUES (3, 'Marian', 'González', NULL, '+595975420385', NULL, NULL,  
'4549216', '1996-05-30', NULL, 'S', NULL, NULL);
```

```
INSERT INTO VENDEDOR  
VALUES (4, 'Graciela Maricel', 'Rojas', 'Cáceres', '+595993581723', NULL,  
NULL, '4289333', '1987-07-06', NULL, 'S', NULL, NULL);
```

```
INSERT INTO VENDEDOR  
VALUES (5, 'Patrocinia', 'Denis', NULL, NULL, NULL, NULL, NULL, '1960-10-  
10', NULL, 'S', NULL, NULL);
```

```
INSERT INTO VENDEDOR  
VALUES (6, 'Dahiana Analía', 'Rolón', 'Argüello', NULL, NULL, 'San Ramón  
- Caaguazú', '5733408', '1995-03-25', NULL, 'S', NULL, NULL);
```

```
INSERT INTO VENDEDOR  
VALUES (7, 'Pablo Andrés', 'Marecos', 'Velázquez', '+595976471063', NULL,  
NULL, '4386260', '1986-06-29', NULL, 'S', NULL, NULL);
```

```
INSERT INTO VENDEDOR  
VALUES (8, 'Lilian Zunilda', 'González', 'Aguilera', NULL, NULL, NULL,  
'3407619', '1987-07-02', NULL, 'S', NULL, NULL);
```

```
INSERT INTO VENDEDOR  
VALUES (9, 'Noelia Inés', 'Caballero', 'Armoa', '0975371601', NULL, NULL,  
'3374860', '1977-02-11', NULL, 'S', NULL, NULL);
```

```
COMMIT WORK;
```

```
/* Data for the 'RECIBO_CLIENTE' table (Records 1 - 2) */
```

```
INSERT INTO RECIBO_CLIENTE  
VALUES (1, 1, 2, '2017-08-12', NULL, 'S', 25000, 255000, 35, 0, 1, NULL);
```

```
INSERT INTO RECIBO_CLIENTE  
VALUES (2, 3, 2, '2018-03-19', NULL, 'S', 50000, 180000, 56, 0, 1, NULL);
```

```
COMMIT WORK;
```

```
/* Data for the 'RECIBO_CLIENTE_DETALLE' table (Records 1 - 2) */
```

```
INSERT INTO RECIBO_CLIENTE_DETALLE  
VALUES (1, 1, 2, 25000, 255000, 36);
```

```
INSERT INTO RECIBO_CLIENTE_DETALLE  
VALUES (2, 2, 7, 50000, 180000, 69);
```

```
COMMIT WORK;
```

```

/* Data for the 'CUENTA_CLIENTE' table  (Records 1 - 19) */

INSERT INTO CUENTA_CLIENTE
VALUES (1, 2, 2, '2', '2017-08-12', 'Blues + 25 + Semanal', 0, 25000,
'2017-08-19', NULL, 2, '1', NULL, 2, 2, 1);

COMMIT WORK;

/* Data for the 'COBRANZAMOVIL' table  (Records 1 - 1) */

INSERT INTO COBRANZAMOVIL
VALUES (1, 2, 1, '2017-08-31', 1, 'S', 'N', '789', 25000, 1, 150000);

COMMIT WORK;

/* Data for the 'COBRANZAMOVILOBSERVACION' table  (Records 1 - 12) */

INSERT INTO COBRANZAMOVILOBSERVACION
VALUES (1, 'Pagado');

INSERT INTO COBRANZAMOVILOBSERVACION
VALUES (2, 'No visitado');

INSERT INTO COBRANZAMOVILOBSERVACION
VALUES (3, 'Visita 1');

INSERT INTO COBRANZAMOVILOBSERVACION
VALUES (4, 'Visita 2');

INSERT INTO COBRANZAMOVILOBSERVACION
VALUES (5, 'Visita 3');

INSERT INTO COBRANZAMOVILOBSERVACION
VALUES (6, 'Visita 4');

INSERT INTO COBRANZAMOVILOBSERVACION
VALUES (7, 'No pagara');

INSERT INTO COBRANZAMOVILOBSERVACION
VALUES (8, 'Desaparecio');

INSERT INTO COBRANZAMOVILOBSERVACION
VALUES (9, 'Notif. Requerimiento');

INSERT INTO COBRANZAMOVILOBSERVACION
VALUES (10, 'Nofif. Pre Demanda');

INSERT INTO COBRANZAMOVILOBSERVACION
VALUES (11, 'Demandar');

INSERT INTO COBRANZAMOVILOBSERVACION
VALUES (12, 'Supervision');

```



```
COMMIT WORK;
```

```
/* Data for the 'COBRANZAMOVILD' table (Records 1 - 1) */
```

```
INSERT INTO COBRANZAMOVILD  
VALUES (1, 1, 2, 0, 0, 0, NULL, NULL, NULL, 2, NULL, '1899-12-30', 'S',  
NULL, 280000, NULL, 'Orlando', 25000, '2');
```

```
COMMIT WORK;
```

```
/* Data for the 'UCTABUSERS' table (Records 1 - 2) */
```

```
INSERT INTO UCTABUSERS  
VALUES (1, 'Admin', 'Admin', 'jU/GVqfSaD', '16/06/2012', 0, 30,  
'qmd@usercontrol.com.br', -1, 'U', 0, 'xo56NZisv5nLzxU+dA');
```

```
INSERT INTO UCTABUSERS  
VALUES (2, 'Usuario Principal', 'Usuario', '', '11/09/2017', 0, 10, '', -  
1, 'U', 0, 'y8P6oKz09rN');
```

```
COMMIT WORK;
```

```
/* Data for the 'FORMA_PAGO' table (Records 1 - 4) */
```

```
INSERT INTO FORMA_PAGO  
VALUES (1, 'D');
```

```
INSERT INTO FORMA_PAGO  
VALUES (2, 'S');
```

```
INSERT INTO FORMA_PAGO  
VALUES (3, 'Q');
```

```
INSERT INTO FORMA_PAGO  
VALUES (4, 'M');
```

```
COMMIT WORK;
```

```
/* Data for the 'ALMACEN_MOVI' table (Records 1 - 5) */
```

```
INSERT INTO ALMACEN_MOVI  
VALUES (1, 'dfgsdfgdfgdfg', '2017-08-12', 1, 'S');
```

```
INSERT INTO ALMACEN_MOVI  
VALUES (2, '5425017734321', '2017-08-16', 2, 'S');
```

```
INSERT INTO ALMACEN_MOVI  
VALUES (3, 'Maletín Marian', '2017-08-16', 2, 'S');
```

```
INSERT INTO ALMACEN_MOVI  
VALUES (4, NULL, '2017-08-24', 1, 'N');
```

```
INSERT INTO ALMACEN_MOVI  
VALUES (5, 'Compra 1noicai', '2018-03-19', 1, 'S');
```

COMMIT WORK;

/\* Data for the 'ALMACEN\_MOVI\_D' table (Records 1 - 2) \*/

INSERT INTO ALMACEN\_MOVI\_D  
VALUES (1, 1, 2, 10, 150000, 3);

INSERT INTO ALMACEN\_MOVI\_D  
VALUES (2, 5, 37, 50, 75000, 0);

COMMIT WORK;

/\* Data for the 'PRODUCTO' table (Records 1 - 400) \*/

INSERT INTO PRODUCTO  
VALUES (2, 'Blue Seduction', '8411061636275', 280000, 250000, 145000, 50,  
'N', 150000, 2, 2, 1);

INSERT INTO PRODUCTO  
VALUES (3, 'Spirit for Men (promo)', '8411061645406', 280000, 238000,  
224000, 75, 'N', 28000, 2, 2, 1);

INSERT INTO PRODUCTO  
VALUES (4, 'Spirit for Men', '8411061552063', 280000, 238000, 224000, 50,  
'N', 62000, 2, 2, 1);

INSERT INTO PRODUCTO  
VALUES (5, 'Spirit for Women', '8411061561249', 280000, 238000, 224000,  
50, 'N', 62000, 2, 2, 2);

INSERT INTO PRODUCTO  
VALUES (66, 'Arsenal Blue', '3700608300011', 380000, 323000, 304000, 100,  
'N', 100000, 2, 33, 1);

INSERT INTO PRODUCTO  
VALUES (6, 'Mediterráneo', '8411061292488', 250000, 212500, 200000, 50,  
'N', 53000, 2, 2, 1);

INSERT INTO PRODUCTO  
VALUES (7, 'Diavolo', '8411061771419', 250000, 212500, 200000, 50, 'N',  
48000, 2, 2, 1);

INSERT INTO PRODUCTO  
VALUES (8, 'Her Golden Secret', '8411061770801', 280000, 238000, 224000,  
50, 'N', 73000, 2, 2, 2);

INSERT INTO PRODUCTO  
VALUES (9, 'The Golden Secret', '8411061727935', 280000, 238000, 224000,  
50, 'N', 73000, 2, 2, 1);

INSERT INTO PRODUCTO  
VALUES (10, 'Blue Seduction for Women', '8411061636206', 280000, 238000,  
224000, 50, 'N', 76000, 2, 2, 2);

```
INSERT INTO PRODUCTO
VALUES (11, 'Mediterráneo', '8411061027011', 350000, 297500, 280000, 100,
'N', 81000, 2, 2, 1);
```

```
INSERT INTO PRODUCTO
VALUES (12, 'King of Seduction', '8411061784280', 280000, 238000, 224000,
50, 'N', 82000, 2, 2, 1);
```

```
INSERT INTO PRODUCTO
VALUES (13, 'Seduction in Black', '8411061698594', 280000, 238000,
224000, 50, 'N', 73000, 2, 2, 1);
```

```
INSERT INTO PRODUCTO
VALUES (14, 'Unic', '5425017734321', 260000, 221000, 208000, 100, 'N',
48000, 2, 3, 1);
```

```
INSERT INTO PRODUCTO
VALUES (15, 'Prestige US Army', '5425017733560', 260000, 221000, 208000,
100, 'N', 50000, 2, 3, 1);
```

```
INSERT INTO PRODUCTO
VALUES (16, 'Prestige Gold for Men', '5425017734277', 260000, 221000,
208000, 100, 'N', 50000, 2, 3, 1);
```

```
INSERT INTO PRODUCTO
VALUES (17, 'Prestige 4 Women Delicious', '5425017736431', 260000,
221000, 208000, 100, 'N', 48000, 2, 3, 2);
```

```
INSERT INTO PRODUCTO
VALUES (18, 'Prestige Gold for Women', '5425017734314', 260000, 221000,
208000, 100, 'N', 50000, 2, 3, 2);
```

```
INSERT INTO PRODUCTO
VALUES (19, 'Prestige Invincible for Men', '5425039220161', 260000,
221000, 208000, 100, 'N', 50000, 2, 3, 1);
```

```
INSERT INTO PRODUCTO
VALUES (20, 'Prestige Ego Gold for Men', '5425017735113', 260000, 221000,
208000, 100, 'N', 50000, 2, 3, 1);
```

```
INSERT INTO PRODUCTO
VALUES (21, 'Essence Pink Gold for Women', '5425039220109', 260000,
221000, 208000, 100, 'N', 56000, 2, 5, 2);
```

```
INSERT INTO PRODUCTO
VALUES (22, 'Fluo Pink for Women', '5425017735151', 260000, 221000,
208000, 100, 'N', 48000, 2, 3, 2);
```

```
INSERT INTO PRODUCTO
VALUES (23, 'Prestige Extasia for Women', '5425017731016', 260000,
221000, 208000, 100, 'N', 48000, 2, 3, 2);
```

```
INSERT INTO PRODUCTO
```

```
VALUES (24, 'In Love for Women', '5425017734468', 260000, 221000, 208000,
100, 'N', 44000, 2, 3, 2);
```

```
INSERT INTO PRODUCTO
VALUES (25, 'Elixir for Women', '5425017735939', 260000, 221000, 208000,
75, 'N', 56000, 2, 4, 2);
```

```
INSERT INTO PRODUCTO
VALUES (26, 'O de la Vie for Women', '5425017736004', 260000, 221000,
208000, 80, 'N', 56000, 2, 4, 2);
```

```
INSERT INTO PRODUCTO
VALUES (27, 'Prestige Velvet for Women', '5425017735137', 260000, 221000,
208000, 100, 'N', 48000, 2, 3, 2);
```

```
INSERT INTO PRODUCTO
VALUES (28, 'Cabotine', '7640111494324', 240000, 204000, 192000, 30, 'N',
56000, 2, 6, 2);
```

```
INSERT INTO PRODUCTO
VALUES (29, 'Gabriela Sabatini', '4004711262401', 260000, 221000, 208000,
30, 'N', 75500, 2, 7, 2);
```

```
INSERT INTO PRODUCTO
VALUES (30, 'Lapidus pour Homme', '3355992000260', 340000, 289000,
272000, 100, 'N', 95000, 2, 8, 1);
```

```
INSERT INTO PRODUCTO
VALUES (31, 'Lapidus pour Homme', '3355992000246', 260000, 221000,
208000, 30, 'N', 66000, 2, 8, 1);
```

```
INSERT INTO PRODUCTO
VALUES (32, 'pour lui', '3355992002905', 340000, 289000, 272000, 100,
'N', 95000, 2, 8, 1);
```

```
INSERT INTO PRODUCTO
VALUES (33, 'Rumba', '3355992004596', 460000, 391000, 368000, 100, 'N',
950000, 2, 8, 2);
```

```
INSERT INTO PRODUCTO
VALUES (34, 'Azzaro pour Homme', '3351500981700', 290000, 246500, 232000,
30, 'N', 95000, 2, 9, 1);
```

```
INSERT INTO PRODUCTO
VALUES (35, 'Night', '3326240027407', 260000, 221000, 208000, 100, 'N',
56000, 2, 10, 1);
```

```
INSERT INTO PRODUCTO
VALUES (36, 'Team Five Special Edition', '3607346551035', 260000, 221000,
208000, 100, 'N', 53000, 2, 11, 1);
```

```
INSERT INTO PRODUCTO
VALUES (37, 'Scuderia Ferrari', '8002135088870', 280000, 238000, 224000,
40, 'N', 75000, 2, 12, 1);
```

```
INSERT INTO PRODUCTO
VALUES (38, 'Scuderia Ferrari Black', '8002135112018', 280000, 238000,
224000, 40, 'N', 81000, 2, 12, 1);
```

```
INSERT INTO PRODUCTO
VALUES (39, 'Dance', '8411061823033', 290000, 246500, 232000, 50, 'N',
81000, 2, 13, 2);
```

```
INSERT INTO PRODUCTO
VALUES (40, 'Love Rock!', '8411061810606', 280000, 238000, 224000, 50,
'N', 81000, 2, 13, 2);
```

```
INSERT INTO PRODUCTO
VALUES (41, 'S (Clásico)', '8411061697290', 280000, 238000, 224000, 50,
'N', 81000, 2, 13, 2);
```

```
INSERT INTO PRODUCTO
VALUES (42, 'Cuba Original Gold for Men', '5425017732389', 240000,
204000, 192000, 100, 'N', 34000, 2, 14, 1);
```

```
INSERT INTO PRODUCTO
VALUES (43, 'Halloween', '8431754340616', 260000, 221000, 208000, 30,
'N', 72500, 2, 15, 2);
```

```
INSERT INTO PRODUCTO
VALUES (44, 'Doriane', '3442151005829', 340000, 289000, 272000, 100, 'N',
81000, 2, 16, 2);
```

```
INSERT INTO PRODUCTO
VALUES (45, 'Chifon', '6291103661791', 340000, 289000, 272000, 100, 'N',
81000, 2, 17, 2);
```

```
INSERT INTO PRODUCTO
VALUES (46, 'SaGa', '6291103661784', 340000, 289000, 272000, 100, 'N',
81000, 2, 17, 2);
```

```
INSERT INTO PRODUCTO
VALUES (47, 'Jacques Saint Prés Indra', '3326240126612', 260000, 221000,
208000, 100, 'N', 50000, 2, 10, 2);
```

```
INSERT INTO PRODUCTO
VALUES (48, 'Privé Coral pour Femme', '6291103661050', 290000, 246500,
232000, 95, 'N', 72500, 2, 17, 2);
```

```
INSERT INTO PRODUCTO
VALUES (49, 'Eternal Dreams', '6295124016752', 260000, 221000, 208000,
100, 'N', 53000, 2, 18, 2);
```

```
INSERT INTO PRODUCTO
VALUES (50, 'Diva', '8411061759066', 280000, 238000, 224000, 80, 'N',
70000, 2, 19, 2);
```

```
INSERT INTO PRODUCTO
```

```
VALUES (51, 'Aubana', '3243655312099', 340000, 289000, 272000, 100, 'N',  
94000, 2, 20, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (52, 'Beautiful Pink', '8715658997795', 260000, 221000, 208000,  
100, 'N', 78000, 2, 21, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (53, 'Pino Silvestre Original', '679602212434', 260000, 221000,  
208000, 40, 'N', 70000, 2, 25, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (54, 'Animale for Men', '892456000150', 260000, 221000, 208000,  
30, 'N', 75500, 2, 27, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (55, 'Animale for Women', '892456000013', 260000, 221000, 208000,  
30, 'N', 81000, 2, 27, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (56, 'Intenso by Café', '3331432800102', 260000, 221000, 208000,  
30, 'N', 51000, 2, 30, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (57, 'Agua Brava', '8411061401705', 280000, 238000, 224000, 100,  
'N', 81000, 2, 24, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (58, '1 Million', '3349666007891', 590000, 501500, 472000, 50,  
'N', 288000, 2, 31, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (59, '212 Men NYC', '8411061341803', 390000, 331500, 312000, 30,  
'N', 120000, 2, 23, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (60, 'Cheap and Chic Moschino Chic Petals', '8011003814305',  
590000, 501500, 472000, 100, 'N', 215000, 2, 29, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (61, 'Eternity Aqua', '3607348556700', 650000, 552500, 520000,  
100, 'N', 220500, 2, 22, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (62, 'Red Door', '085805114435', 360000, 306000, 288000, 30, 'N',  
116000, 2, 26, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (63, 'Brand Collection N° 001', '6936711314751', 240000, 204000,  
192000, 25, 'N', 48000, 2, 28, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (64, 'Brand Collection N° 005', '6936711314799', 240000, 204000,  
192000, 25, 'N', 48000, 2, 28, 1);
```

```
INSERT INTO PRODUCTO
VALUES (65, 'Brand Collection N° 008', '6936711314829', 240000, 204000,
192000, 25, 'N', 48000, 2, 28, 1);
```

```
INSERT INTO PRODUCTO
VALUES (67, 'Arsenal Gold', '3700608300010', 380000, 323000, 304000, 100,
'N', 100000, 2, 33, 1);
```

```
INSERT INTO PRODUCTO
VALUES (68, 'Oh La La!', '8410225518235', 290000, 246500, 232000, 100,
'N', 78000, 2, 32, 2);
```

```
INSERT INTO PRODUCTO
VALUES (69, 'Sweet Dreams', '6295124016745', 260000, 221000, 208000, 100,
'N', 53000, 2, 18, 2);
```

```
INSERT INTO PRODUCTO
VALUES (71, 'The Golden Secret', '8411061722756', 380000, 323000, 304000,
100, 'N', 95000, 2, 2, 1);
```

```
INSERT INTO PRODUCTO
VALUES (70, 'Self pour Homme', '855714004000', 340000, 289000, 272000,
100, 'N', 81000, 2, 34, 1);
```

```
INSERT INTO PRODUCTO
VALUES (72, 'Antonio (Tester)', '8411061587911', 280000, 238000, 224000,
100, 'N', 73000, 2, 2, 1);
```

```
INSERT INTO PRODUCTO
VALUES (73, 'CHHC Privée', '8411061863954', 650000, 552500, 520000, 50,
'N', 274000, 2, 23, 2);
```

```
INSERT INTO PRODUCTO
VALUES (74, 'Spirit for Men (After Shave)', '8411061645407', 150000,
127500, 120000, 30, 'N', 28000, 2, 2, 1);
```

```
INSERT INTO PRODUCTO
VALUES (226, 'Laloa Sexy', '3700608300102', 260000, 221000, 208000, 100,
'N', 42150, 2, 56, 2);
```

```
INSERT INTO PRODUCTO
VALUES (319, 'Herrera for Men', '8411061081600', 650000, 552500, 520000,
100, 'N', 273000, 2, 23, 1);
```

```
INSERT INTO PRODUCTO
VALUES (76, 'Club N° 1', '00002', 260000, 221000, 208000, 100, 'N',
37758, 2, 9, 1);
```

```
INSERT INTO PRODUCTO
VALUES (88, 'La Prière', '855714004017', 290000, 246500, 232000, 100,
'N', 84000, 2, 34, 2);
```

```
INSERT INTO PRODUCTO
```

```
VALUES (247, 'Miss Dior', '3348901016278', 850000, 722500, 680000, 50, 'N', 377000, 2, 50, 2);
```

```
INSERT INTO PRODUCTO
VALUES (78, 'Cuba Royal for Men', '5425017736363', 240000, 204000, 192000, 100, 'N', 33540, 2, 14, 1);
```

```
INSERT INTO PRODUCTO
VALUES (79, 'Her Golden Secret Kit', '8411061833421', 420000, 357000, 336000, 80, 'N', 114595, 2, 2, 2);
```

```
INSERT INTO PRODUCTO
VALUES (80, 'The Secret Temptation', '8411061860564', 280000, 238000, 224000, 50, 'N', 81055, 2, 2, 1);
```

```
INSERT INTO PRODUCTO
VALUES (81, 'Self pour Homme', '855714004000', 340000, 289000, 272000, 100, 'N', 92235, NULL, 34, 1);
```

```
INSERT INTO PRODUCTO
VALUES (82, 'Elixir for Women', '5425017735939', 260000, 221000, 208000, 75, 'N', 55620, 2, 4, NULL);
```

```
INSERT INTO PRODUCTO
VALUES (89, 'Halloween Man Rock On', '8431754501512', 360000, 306000, 288000, 75, 'N', 165000, 2, 15, 1);
```

```
INSERT INTO PRODUCTO
VALUES (90, '212 Sexy', '8411061558423', 390000, 331500, 312000, 30, 'N', 165000, 2, 23, 2);
```

```
INSERT INTO PRODUCTO
VALUES (91, 'Prestige Princess Dreaming', '5425017735076', 260000, 221000, 208000, 100, 'N', 51000, 2, 3, 2);
```

```
INSERT INTO PRODUCTO
VALUES (92, 'Chance', '3145891264500', 750000, 637500, 600000, 50, 'N', 377700, 2, 35, 2);
```

```
INSERT INTO PRODUCTO
VALUES (93, 'Eau Florale', '8411061722176', 280000, 238000, 224000, 50, 'N', 82360, 2, 13, 2);
```

```
INSERT INTO PRODUCTO
VALUES (94, 'Pure Pink for Women', '3700134404381', 360000, 306000, 288000, 50, 'N', 97000, 2, 36, 2);
```

```
INSERT INTO PRODUCTO
VALUES (95, 'Animale for Men', '892456000174', 560000, 476000, 448000, 100, 'N', 130500, 2, 27, 1);
```

```
INSERT INTO PRODUCTO
VALUES (96, 'Essence for Women', '5425039220079', 260000, 221000, 208000, 100, 'N', 56500, 2, 5, 2);
```



```
INSERT INTO PRODUCTO
VALUES (97, 'Essence for Men', '5425039220062', 260000, 221000, 208000,
100, 'N', 60000, 2, 5, 1);
```

```
INSERT INTO PRODUCTO
VALUES (98, 'Play in Black Seduction', '8411061854761', 380000, 323000,
304000, 100, 'N', 105000, 2, 2, 1);
```

```
INSERT INTO PRODUCTO
VALUES (99, 'Mediterráneo', '8411061705919', 320000, 272000, 256000, 200,
'N', 88000, 2, 2, 1);
```

```
INSERT INTO PRODUCTO
VALUES (100, 'King of Seduction Absolute', '8411061819623', 280000,
238000, 224000, 50, 'N', 82500, 2, 2, 1);
```

```
INSERT INTO PRODUCTO
VALUES (101, 'Legend', '6291103661258', 360000, 306000, 288000, 100, 'N',
80000, 2, 17, 1);
```

```
INSERT INTO PRODUCTO
VALUES (102, 'Les Belles de Nina', '3137370205029', 560000, 476000,
448000, 30, 'N', 158400, 2, 37, 2);
```

```
INSERT INTO PRODUCTO
VALUES (103, 'Cheap and Chic Moschino I Love Love', '8011003991143',
520000, 442000, 416000, 50, 'N', 167265, 2, 29, 2);
```

```
INSERT INTO PRODUCTO
VALUES (104, 'café - café', '3331432180037', 320000, 272000, 256000, 100,
'N', 90500, 2, 30, 2);
```

```
INSERT INTO PRODUCTO
VALUES (105, 'Hugo Boss Energise', '737052139906', 590000, 501500,
472000, 75, 'N', 181500, 2, 7, 1);
```

```
INSERT INTO PRODUCTO
VALUES (248, 'Carolina Herrera New York', '8411061061107', 620000,
527000, 496000, 100, 'N', 329700, 2, 23, 1);
```

```
INSERT INTO PRODUCTO
VALUES (107, 'Quorum', '8411061411919', 390000, 331500, 312000, 100, 'N',
108000, 2, 24, 1);
```

```
INSERT INTO PRODUCTO
VALUES (108, 'Paco Rabanne pour Homme', '3349668021345', 590000, 501500,
472000, 100, 'N', 247000, 2, 31, 1);
```

```
INSERT INTO PRODUCTO
VALUES (109, 'Café', '3331430066043', 320000, 272000, 256000, 100, 'N',
90500, 2, 30, 2);
```

```
INSERT INTO PRODUCTO
```

```
VALUES (110, 'UDV for Men', '3326240816698', 260000, 221000, 208000, 100, 'N', 50000, 2, 10, NULL);
```

```
INSERT INTO PRODUCTO  
VALUES (154, 'Lou Lou', '3360373000074', 390000, 331500, 312000, 30, 'N', 138000, 2, 43, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (112, 'Cabotine Kit (3 piezas)', '7640163970692', 420000, 357000, 336000, 100, 'N', 150255, 2, 6, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (113, 'Azzaro pour Homme', '3351500980406', 490000, 416500, 392000, 100, 'N', 162000, 2, 9, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (114, 'Hugo Boss Man', '737052319995', 380000, 323000, 304000, 40, 'N', 1390000, 2, 7, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (115, '212 Vip NYC', '8411061711774', 390000, 331500, 312000, 30, 'N', 150255, 2, 23, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (116, 'Scuderia Ferrari Black', '8002135111998', 380000, 323000, 304000, 75, 'N', 91000, 2, 12, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (117, 'Blue Seduction', '8411061636268', 380000, 323000, 304000, 100, 'N', 90500, 2, 2, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (118, 'Les Belles de Nina', '3137370180333', 650000, 552500, 520000, 50, 'N', 273500, 2, 37, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (119, 'SaGa Pink', '6291103661005', 340000, 289000, 272000, 100, 'N', 81000, 2, 17, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (120, 'Colonia Corporal Barbie', '663350052374', 260000, 221000, 208000, 200, 'N', 47600, 2, 39, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (121, 'Colonia Corporal Sofia the First', '663350063523', 260000, 221000, 208000, 200, 'N', 47600, 2, 39, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (122, 'Colonia Fresca Cars', '663350052411', 260000, 221000, 208000, 200, 'N', 47600, 2, 39, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (123, 'Colonia Fresca Super Wings', '8411114079660', 260000, 221000, 208000, 200, 'N', 47600, 2, 39, 1);
```

```
INSERT INTO PRODUCTO
VALUES (124, '212 Sexy Men', '8411061670989', 390000, 331500, 312000, 30,
'N', 162000, 2, 23, 1);
```

```
INSERT INTO PRODUCTO
VALUES (125, 'Eternity for Men', '088300105519', 490000, 416500, 392000,
100, 'N', 201000, 2, 22, 1);
```

```
INSERT INTO PRODUCTO
VALUES (126, 'One', '3607343811835', 490000, 416500, 392000, 100, 'N',
128800, 2, 22, 1);
```

```
INSERT INTO PRODUCTO
VALUES (127, 'Fantasy Britney Spears', '719346065399', 360000, 306000,
288000, 50, 'N', 120400, 2, 40, 2);
```

```
INSERT INTO PRODUCTO
VALUES (128, 'Forever', '5425017730620', 260000, 221000, 208000, 100,
'N', 43680, 2, 3, 2);
```

```
INSERT INTO PRODUCTO
VALUES (129, 'Black XS', '3349668161546', 490000, 416500, 392000, 50,
'N', 193200, 2, 31, 1);
```

```
INSERT INTO PRODUCTO
VALUES (130, 'Chrome', '3351500920006', 290000, 246500, 232000, 30, 'N',
102000, 2, 9, 1);
```

```
INSERT INTO PRODUCTO
VALUES (131, 'CHHC', '8411061607169', 650000, 552500, 520000, 50, 'N',
288400, 2, 23, 2);
```

```
INSERT INTO PRODUCTO
VALUES (132, '212 VIP Men', '8411061869253', 590000, 501500, 472000, 50,
'N', 243600, 2, 23, 1);
```

```
INSERT INTO PRODUCTO
VALUES (133, 'Finding Dory', '663350055368', 260000, 221000, 208000, 100,
'N', 47600, 2, 39, 2);
```

```
INSERT INTO PRODUCTO
VALUES (134, '212 Men NYC', '8411061341704', 590000, 501500, 472000, 50,
'N', 238000, 2, 23, 1);
```

```
INSERT INTO PRODUCTO
VALUES (135, 'Blue Seduction (Tester)', '8411061636329', 350000, 297500,
280000, 80, 'N', 67200, 2, 2, 1);
```

```
INSERT INTO PRODUCTO
VALUES (136, '212 Men Sexy', '8411061602539', 590000, 501500, 472000, 50,
'N', 246120, 2, 23, 1);
```

```
INSERT INTO PRODUCTO
```

VALUES (137, 'Scuderia Ferrari', '8002135088894', 380000, 323000, 304000, 75, 'N', 98000, 2, 12, 1);

INSERT INTO PRODUCTO

VALUES (138, 'Fantasy Britney Spears kit', '719346610957', 360000, 306000, 288000, 30, 'N', 112000, 2, 40, 2);

INSERT INTO PRODUCTO

VALUES (155, 'Animale Black', '878813001275', 260000, 221000, 208000, 30, 'N', 60952.5, 2, 27, 1);

INSERT INTO PRODUCTO

VALUES (140, 'The Secret', '8411061701041', 280000, 238000, 224000, 50, 'N', 70000, 2, 2, 1);

INSERT INTO PRODUCTO

VALUES (141, 'The Golden Secret (Tester)', '8411061722794', 350000, 297500, 280000, 100, 'N', 67200, 2, 2, 1);

INSERT INTO PRODUCTO

VALUES (142, 'l''''Oriental White Edition', '3700134405968', 340000, 289000, 272000, 100, 'N', 88480, 2, 41, 1);

INSERT INTO PRODUCTO

VALUES (143, 'l''''Oriental Men', '3700134405180', 340000, 289000, 272000, 100, 'N', 88480, 2, 41, 1);

INSERT INTO PRODUCTO

VALUES (144, 'l''''Oriental Icon', '3700134409034', 340000, 289000, 272000, 100, 'N', 88480, 2, 41, 1);

INSERT INTO PRODUCTO

VALUES (145, '212 Vip Rosé', '8411061777183', 650000, 552500, 520000, 50, 'N', 277200, 2, 23, 2);

INSERT INTO PRODUCTO

VALUES (146, 'Taxi', '3331430740035', 260000, 221000, 208000, 100, 'N', 53200, 2, 42, 1);

INSERT INTO PRODUCTO

VALUES (147, 'Animale for Men', '892456000167', 360000, 306000, 288000, 50, 'N', 103600, 2, 27, 1);

INSERT INTO PRODUCTO

VALUES (148, 'Invictus', '3349668515660', 750000, 637500, 600000, 100, 'N', 333200, 2, 31, 1);

INSERT INTO PRODUCTO

VALUES (156, 'Queen of Seduction', '8411061820117', 280000, 238000, 224000, 50, 'N', 76545, 2, 2, 2);

INSERT INTO PRODUCTO

VALUES (149, 'UDV for Men', '3326240816698', 260000, 221000, 208000, 100, 'N', 50000, 2, 10, 1);

```
INSERT INTO PRODUCTO
VALUES (150, 'Cuba Original Winner', '5425017736134', 260000, 221000,
208000, 100, 'N', 30800, 2, 14, 1);
```

```
INSERT INTO PRODUCTO
VALUES (151, 'Versace Blue Jeans Man', '8018365260757', 490000, 416500,
392000, 75, 'N', 110565, 2, 29, 1);
```

```
INSERT INTO PRODUCTO
VALUES (152, 'Eternity', '8018737922', 620000, 527000, 496000, 100, 'N',
229635, 2, 22, 2);
```

```
INSERT INTO PRODUCTO
VALUES (153, 'Privé', '8411061754467', 260000, 221000, 208000, 100, 'N',
53865, 2, 19, 2);
```

```
INSERT INTO PRODUCTO
VALUES (157, 'Her Secret Temptation', '8411061860441', 280000, 238000,
224000, 50, 'N', 70875, 2, 2, 2);
```

```
INSERT INTO PRODUCTO
VALUES (158, 'tommy', '022548024317', 520000, 442000, 416000, 50, 'N',
133245, 2, 44, 1);
```

```
INSERT INTO PRODUCTO
VALUES (159, 'Herrera for Men', '8411061081709', 520000, 442000, 416000,
50, 'N', 226516.5, 2, 23, 1);
```

```
INSERT INTO PRODUCTO
VALUES (160, 'Euphoria', '088300162505', 650000, 552500, 520000, 100,
'N', 240975, 2, 22, 2);
```

```
INSERT INTO PRODUCTO
VALUES (161, 'Fassio Essence', '6291103660398', 290000, 246500, 232000,
100, 'N', 73710, 2, 17, 2);
```

```
INSERT INTO PRODUCTO
VALUES (162, 'Amarige', '3274878122509', 390000, 331500, 312000, 30, 'N',
137500, 2, 45, 2);
```

```
INSERT INTO PRODUCTO
VALUES (163, 'Bleu', '3145891074505', 890000, 756500, 712000, 50, 'N',
412450, 2, 35, 1);
```

```
INSERT INTO PRODUCTO
VALUES (164, 'Eternity', '088300101306', 490000, 416500, 392000, 50, 'N',
186167.5, 2, 22, 2);
```

```
INSERT INTO PRODUCTO
VALUES (165, 'Escape', '088300608409', 650000, 552500, 520000, 100, 'N',
214700, 2, 22, 2);
```

```
INSERT INTO PRODUCTO
```

```
VALUES (166, '212 NYC', '8411061894026', 650000, 552500, 520000, 60, 'N',  
257075, 2, 23, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (167, '212 Men NYC', '8411061896259', 590000, 501500, 472000, 50,  
'N', 251425, 2, 23, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (168, 'Britney Spears Fantasy', '719346065399', 490000, 416500,  
392000, 50, 'N', 150000, 2, 40, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (169, 'the one', '737052036649', 650000, 552500, 520000, 100, 'N',  
327417.5, 2, 46, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (170, 'Colonia Corporal Minnie', '663350053647', 260000, 221000,  
208000, 200, 'N', 45200, 2, 39, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (171, 'Colonia Fresca PJMasks', '8411114079707', 260000, 221000,  
208000, 200, 'N', 45200, 2, 39, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (172, 'tommy', '022548024324', 650000, 552500, 520000, 100, 'N',  
149725, 2, 44, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (173, 'Euphoria', '088300162543', 560000, 476000, 448000, 50, 'N',  
205200, 2, 22, NULL);
```

```
INSERT INTO PRODUCTO  
VALUES (174, '212 CH MEN Prive', '8411061786345', 590000, 501500, 472000,  
50, 'N', 227715, NULL, 23, NULL);
```

```
INSERT INTO PRODUCTO  
VALUES (175, 'Euphoria', '088300162543', 560000, 476000, 448000, 50, 'N',  
203400, 2, 22, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (176, 'CHHC Men Privé', '8411061786345', 590000, 501500, 472000,  
50, 'N', 225717.5, 2, 23, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (177, 'Hugo Boss Man', '737052664026', 490000, 416500, 392000, 75,  
'N', 219000, 2, 7, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (178, 'One', '088300607686', 380000, 323000, 304000, 50, 'N',  
105000, 2, 22, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (179, 'café', '3331430066029', 260000, 221000, 208000, 60, 'N',  
60000, 2, 30, 2);
```

```
INSERT INTO PRODUCTO
VALUES (180, 'Quorum', '8411061411919', 390000, 331500, 312000, 100, 'N',
114000, 2, NULL, 1);
```

```
INSERT INTO PRODUCTO
VALUES (181, 'Her Secret', '8411061738009', 280000, 238000, 224000, 50,
'N', 75000, 2, 2, 2);
```

```
INSERT INTO PRODUCTO
VALUES (182, 'Diavolo', '8411061771907', 350000, 297500, 280000, 100,
'N', 72000, 2, 2, 1);
```

```
INSERT INTO PRODUCTO
VALUES (184, 'Manifesto', '3365440226708', 850000, 722500, 680000, 90,
'N', 418100, 2, 47, 1);
```

```
INSERT INTO PRODUCTO
VALUES (183, 'Gentlemen Only', '3274870012143', 650000, 552500, 520000,
50, 'N', 211875, 2, 45, 1);
```

```
INSERT INTO PRODUCTO
VALUES (185, 'Be', '088300104406', 490000, 416500, 392000, 100, 'N',
129950, 2, 22, 1);
```

```
INSERT INTO PRODUCTO
VALUES (186, 'Colonia Fresca Minions', '663350062403', 260000, 221000,
208000, 200, 'N', 45200, 2, 39, 1);
```

```
INSERT INTO PRODUCTO
VALUES (187, 'Colonia Fresca Super Wings', '8411114079660', 260000,
221000, 208000, 200, 'N', 45200, 2, 39, 1);
```

```
INSERT INTO PRODUCTO
VALUES (188, 'S (Clásico)', '8411061697283', 390000, 331500, 312000, 80,
'N', 95767.5, 2, 13, 2);
```

```
INSERT INTO PRODUCTO
VALUES (189, 'Me L'''Absolu', '3386460062688', 650000, 552500, 520000,
50, 'N', 271395, 2, 48, 2);
```

```
INSERT INTO PRODUCTO
VALUES (190, 'Invictus', '3349668515653', 590000, 501500, 472000, 50,
'N', 250000, 2, 31, 1);
```

```
INSERT INTO PRODUCTO
VALUES (191, 'Cuba Original', '5425017732396', 260000, 221000, 208000,
100, 'N', 30250, 2, 14, 1);
```

```
INSERT INTO PRODUCTO
VALUES (192, 'Invictus', '3349668543021', 720000, 612000, 576000, 100,
'N', 460650, 2, 31, 1);
```

```
INSERT INTO PRODUCTO
```

```
VALUES (193, 'Cuba Black for Men', '5425017732402', 240000, 204000,
192000, 100, 'N', 30525, 2, 14, 1);
```

```
INSERT INTO PRODUCTO
VALUES (194, 'Jacques Saint Press Isa', '3326240726676', 260000, 221000,
208000, 100, 'N', 48840, 2, 10, 2);
```

```
INSERT INTO PRODUCTO
VALUES (195, 'Blue', '3326240003852', 260000, 221000, 208000, 100, 'N',
71872.5, 2, 10, 1);
```

```
INSERT INTO PRODUCTO
VALUES (196, 'Boum Sport', '3430750040174', 260000, 221000, 208000, 100,
'N', 50000, 2, 49, 1);
```

```
INSERT INTO PRODUCTO
VALUES (197, 'CHHC Men', '8411061665039', 650000, 552500, 520000, 50,
'N', 213675, 2, 23, 1);
```

```
INSERT INTO PRODUCTO
VALUES (198, 'J''''Adore', '3348900417885', 850000, 722500, 680000, 50,
'N', 380175, 2, 50, 2);
```

```
INSERT INTO PRODUCTO
VALUES (199, 'Halloween', '8431754341019', 360000, 306000, 288000, 50,
'N', 110175, 2, 15, 2);
```

```
INSERT INTO PRODUCTO
VALUES (200, 'Love Animale', '892456000976', 380000, 323000, 304000, 50,
'N', 109000, 2, 27, 2);
```

```
INSERT INTO PRODUCTO
VALUES (201, 'Fashionista', '5425039220338', 260000, 221000, 208000, 100,
'N', 47770, 2, 3, 2);
```

```
INSERT INTO PRODUCTO
VALUES (202, 'Cuba Original Set Gold for Men', '5425017732273', 280000,
238000, 224000, 100, 'N', 42150, 2, 14, 1);
```

```
INSERT INTO PRODUCTO
VALUES (203, 'Chantage', '6295124005497', 260000, 221000, 208000, 100,
'N', 80000, 2, 53, 2);
```

```
INSERT INTO PRODUCTO
VALUES (249, '212 Sexy', '8411061865460', 620000, 527000, 496000, 60,
'N', 265845, 2, 23, 1);
```

```
INSERT INTO PRODUCTO
VALUES (205, 'Princesse Clasique', '3494800042044', 290000, 246500,
232000, 30, 'N', 80000, 2, 51, 2);
```

```
INSERT INTO PRODUCTO
VALUES (206, 'K''''men for Men', '802822001650', 260000, 221000, 208000,
100, 'N', 70000, 2, 3, 1);
```



```

INSERT INTO PRODUCTO
VALUES (207, '212 Vip Men NYC Kit', '8411061831304', 750000, 637500,
600000, 100, 'N', 311910, 2, 23, 1);

INSERT INTO PRODUCTO
VALUES (208, 'Dance Kit', '8411061886649', 320000, 272000, 256000, 50,
'N', 83600, 2, 13, NULL);

INSERT INTO PRODUCTO
VALUES (209, 'Eternity', '088300101405', 650000, 552500, 520000, 100,
'N', 226800, 2, 22, 2);

INSERT INTO PRODUCTO
VALUES (210, 'Blue Diamond for Women', '5425039220178', 260000, 221000,
208000, 100, 'N', 53200, 2, 4, 2);

INSERT INTO PRODUCTO
VALUES (211, 'Paw Patrol', '663350073560', 260000, 221000, 208000, 200,
'N', 44800, 2, 39, 1);

INSERT INTO PRODUCTO
VALUES (212, 'Her Golden Secret Kit', '8411061876817', 420000, 357000,
336000, 80, 'N', 110880, 2, 2, 2);

INSERT INTO PRODUCTO
VALUES (214, 'Euphoria', '088300178278', 490000, 416500, 392000, 100,
'N', 190400, 2, 22, 1);

INSERT INTO PRODUCTO
VALUES (213, '212 Men NYC', '8411061853160', 750000, 637500, 600000, 100,
'N', 305200, 2, 23, 1);

INSERT INTO PRODUCTO
VALUES (215, 'Cabotine Kit 100 200', '7640111492030', 420000, 357000,
336000, 100, 'N', 148400, 2, 6, 2);

INSERT INTO PRODUCTO
VALUES (216, 'The Secret', '8411061701034', 380000, 323000, 304000, 100,
'N', 92400, 2, 2, 1);

INSERT INTO PRODUCTO
VALUES (217, 'The Golden Secret', '8411061722756', 380000, 323000,
304000, 100, 'N', 92400, 2, 2, 1);

INSERT INTO PRODUCTO
VALUES (218, 'Bleu - Powerful', '737052413204', 650000, 552500, 520000,
100, 'N', 229320, 2, 54, 1);

INSERT INTO PRODUCTO
VALUES (219, 'Predator Paris for Men', '3473460001035', 260000, 221000,
208000, 100, 'N', 50580, 2, 55, 1);

INSERT INTO PRODUCTO

```

```
VALUES (220, 'Lady Million', '3349668508488', 650000, 552500, 520000, 50, 'N', 269479, 2, 31, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (221, 'Animale Intense', '892456000334', 360000, 306000, 288000, 50, 'N', 95259, 2, 27, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (222, 'Ferrari Uomo', '8002135069329', 260000, 221000, 208000, 30, 'N', 53390, 2, 12, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (223, 'Azzaro pour Homme', '3351500980802', 390000, 331500, 312000, 50, 'N', 132070, 2, 9, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (224, '5th Avenue', '085805390402', 360000, 306000, 288000, 30, 'N', 105656, 2, 26, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (225, 'Britney Spears Fantasy Twist 2 in 1', '719346159357', 390000, 331500, 312000, 30, 'N', 103970, 2, 40, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (227, 'Good Girl (crema)', '6949063982769', 180000, 153000, 144000, 200, 'N', 64630, 2, 57, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (228, 'Fantasy (crema)', '6949063982768', 180000, 153000, 144000, 200, 'N', 42150, 2, 57, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (229, 'Ultraviolet kit', '3349668513970', 730000, 620500, 584000, 80, 'N', 297860, 2, 31, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (230, 'Lady Million', '3349668508471', 540000, 459000, 432000, 30, 'N', 221990, 2, 31, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (231, '1 Million', '3349666007921', 750000, 637500, 600000, 100, 'N', 327000, 2, 31, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (232, 'ange ou démon', '3274870373251', 790000, 671500, 632000, 50, 'N', 365019, 2, 45, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (233, 'Burberry for Men', '5045252667392', 620000, 527000, 496000, 100, 'N', 224519, 2, 58, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (250, 'Gabriela Sabatini', '8005610325507', 360000, 306000, 288000, 60, 'N', 99622.5, 2, 7, 2);
```

```
INSERT INTO PRODUCTO
VALUES (234, 'Olympea (crema)', '6949063982721', 180000, 153000, 144000,
200, 'N', 67159, 2, 57, 2);
```

```
INSERT INTO PRODUCTO
VALUES (235, 'The Secret Temptation (tester)', '8411061860533', 360000,
306000, 288000, 100, 'N', 67440, 2, 2, 1);
```

```
INSERT INTO PRODUCTO
VALUES (236, 'Miami Seduction', '8411061921081', 360000, 306000, 288000,
100, 'N', 67440, 2, 2, 1);
```

```
INSERT INTO PRODUCTO
VALUES (237, 'Her Golden Secret', '8411061770795', 380000, 323000,
304000, 80, 'N', 92730, 2, 2, 2);
```

```
INSERT INTO PRODUCTO
VALUES (313, 'Visit', '3351500950027', 490000, 416500, 392000, 100, 'N',
174000, 2, 9, 1);
```

```
INSERT INTO PRODUCTO
VALUES (240, 'Red Door', '085805558420', 540000, 459000, 432000, 100,
'N', 175725, 2, 26, 2);
```

```
INSERT INTO PRODUCTO
VALUES (241, 'Ice Dive', '3412242610089', 260000, 221000, 208000, 100,
'N', 49225, 2, 11, 1);
```

```
INSERT INTO PRODUCTO
VALUES (245, 'Baby Touch', '5045254676101', 590000, 501500, 472000, 100,
'N', 180000, 2, 58, 1);
```

```
INSERT INTO PRODUCTO
VALUES (243, 'L1212 Blanc Pure', '737052413174', 620000, 527000, 496000,
100, 'N', 264000, 2, 54, 1);
```

```
INSERT INTO PRODUCTO
VALUES (246, 'the one', '8330233101020', 720000, 612000, 576000, 50, 'N',
310750, 2, 46, 1);
```

```
INSERT INTO PRODUCTO
VALUES (251, 'Halloween Man Rock On', '8431754501017', 360000, 306000,
288000, 50, 'N', 16650, 2, 15, 1);
```

```
INSERT INTO PRODUCTO
VALUES (253, 'Good Girl', '8411061819838', 890000, 756500, 712000, 50,
'N', 388222.5, 2, 23, 2);
```

```
INSERT INTO PRODUCTO
VALUES (252, 'Gentleman', '3274871935861', 650000, 552500, 520000, 100,
'N', 241000, 2, 45, 1);
```

```
INSERT INTO PRODUCTO
```

```
VALUES (254, 'Armani Code', '3360372100515', 690000, 586500, 552000, 50, 'N', 277222.5, 2, 60, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (255, 'Barbie Pack', '663350059830', 260000, 221000, 208000, 50, 'N', 69375, 2, 39, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (256, 'Flower by Kenzo', '3352818517803', 620000, 527000, 496000, 50, 'N', 241425, 2, 61, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (257, '212 Vip Rose (Crema)', '6949063982707', 180000, 153000, 144000, 200, 'N', 63825, 2, 57, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (258, 'Monaco', '5425017732792', 260000, 221000, 208000, 100, 'N', 43290, 2, 3, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (259, 'Eternity for Men', '088300605514', 490000, 416500, 392000, 100, 'N', 185925, 2, 22, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (260, 'Air Force Army', '5425017735854', 260000, 221000, 208000, 100, 'N', 43290, 2, 3, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (244, 'Drakkar Noir', '3360372009436', 490000, 416500, 392000, 100, 'N', 145750, 2, 59, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (261, 'One', '088300107407', 490000, 416500, 392000, 100, 'N', 135975, 2, 22, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (262, 'Monaco for Men', '5425017732785', 260000, 221000, 208000, 100, 'N', 43290, 2, 3, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (263, 'Seduction in Black', '8411061695951', 380000, 323000, 304000, 100, 'N', 77700, 2, 2, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (360, 'Wanted', '3351500002696', 590000, 501500, 472000, 50, 'N', 169097.5, 2, 9, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (264, 'One Shock', '3607342401105', 520000, 442000, 416000, 100, 'N', 158175, 2, 22, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (265, 'SXY', '0840430138', 240000, 204000, 192000, 50, 'N', 36075, 2, 62, 2);
```

```

INSERT INTO PRODUCTO
VALUES (266, 'Allure Homme sport', '3145891236200', 850000, 722500,
680000, 50, 'N', 349372.5, 2, 35, 1);

INSERT INTO PRODUCTO
VALUES (268, 'CHHC', '8411061607152', 850000, 722500, 680000, 100, 'N',
360472.5, 2, 23, 2);

INSERT INTO PRODUCTO
VALUES (269, 'Spirit for Women', '8411061561232', 380000, 323000, 304000,
100, 'N', 76590, 2, 2, 2);

INSERT INTO PRODUCTO
VALUES (267, 'Agua Brava Kit', '8411061807842', 360000, 306000, 288000,
100, 'N', 86025, 2, 24, 1);

INSERT INTO PRODUCTO
VALUES (270, 'Pink Diamond for Women', '5425017736424', 260000, 221000,
208000, 100, 'N', 56217.5, 2, 4, 2);

INSERT INTO PRODUCTO
VALUES (271, 'Britney Spears Cosmic Radiance', '719346145879', 390000,
331500, 312000, 30, 'N', 111870, 2, 40, 2);

INSERT INTO PRODUCTO
VALUES (272, 'Britney Spears Fantasy Kit 100', '719346633772', 590000,
501500, 472000, 100, 'N', 202000, 2, 40, 2);

INSERT INTO PRODUCTO
VALUES (273, 'Rojo', '737052074597', 490000, 416500, 392000, 25, 'N',
179550, 2, 46, 1);

INSERT INTO PRODUCTO
VALUES (274, 'Mercedes Benz for Men', '3595471021021', 490000, 416500,
392000, 75, 'N', 186000, 2, 63, 1);

INSERT INTO PRODUCTO
VALUES (275, '212 Sexy', '8411061865439', 950000, 807500, 760000, 100,
'N', 369000, 2, 23, 2);

INSERT INTO PRODUCTO
VALUES (279, 'Real Madrid', '663350065725', 340000, 289000, 272000, 100,
'N', 87000, 2, 39, 1);

INSERT INTO PRODUCTO
VALUES (276, 'Chrome', '3351500920037', 490000, 416500, 392000, 100, 'N',
219000, 2, 9, 1);

INSERT INTO PRODUCTO
VALUES (277, 'Someday by Justin Bieber', '852539003027', 490000, 416500,
392000, 100, 'N', 126000, 2, 64, 1);

INSERT INTO PRODUCTO

```

```
VALUES (278, 'Cadillac Black', '45X123400T12339', 460000, 391000, 368000,
100, 'N', 15000, 2, 65, 1);
```

```
INSERT INTO PRODUCTO
VALUES (280, 'Mango Temptation', '667538582004', 260000, 221000, 208000,
250, 'N', 65700, 2, 66, 2);
```

```
INSERT INTO PRODUCTO
VALUES (281, 'Pi', '3274878222568', 720000, 612000, 576000, 100, 'N',
306000, 2, 45, 2);
```

```
INSERT INTO PRODUCTO
VALUES (282, 'Le Male', '8435415012638', 590000, 501500, 472000, 75, 'N',
257700, 2, 67, 1);
```

```
INSERT INTO PRODUCTO
VALUES (283, 'Dance Diamonds', '8411061876039', 290000, 246500, 232000,
50, 'N', 81000, 2, 13, 2);
```

```
INSERT INTO PRODUCTO
VALUES (284, 'Barbie', '663350050530', 290000, 246500, 232000, 100, 'N',
87000, 2, 39, 2);
```

```
INSERT INTO PRODUCTO
VALUES (285, 'Animale for Women', '892456000037', 460000, 391000, 368000,
100, 'N', 165000, 2, 27, 2);
```

```
INSERT INTO PRODUCTO
VALUES (286, '212 VIP Men NYC', '8411061723777', 590000, 501500, 472000,
50, 'N', 261000, 2, 23, 1);
```

```
INSERT INTO PRODUCTO
VALUES (287, 'Mont Blanc Presence', '3386460028332', 490000, 416500,
392000, 50, 'N', 143700, 2, 68, 1);
```

```
INSERT INTO PRODUCTO
VALUES (288, 'One Red Edition', '3607342771215', 490000, 416500, 392000,
100, 'N', 168000, 2, 22, 1);
```

```
INSERT INTO PRODUCTO
VALUES (289, 'Queen of Seduction kit', '8411061820186', 420000, 357000,
336000, 50, 'N', 99000, 2, 2, 2);
```

```
INSERT INTO PRODUCTO
VALUES (290, 'Eternity', '088300601301', 490000, 416500, 392000, 50, 'N',
173700, 2, 22, 1);
```

```
INSERT INTO PRODUCTO
VALUES (291, 'Hugo Boss Woman', '737052893914', 620000, 527000, 496000,
75, 'N', 256800, 2, 7, 2);
```

```
INSERT INTO PRODUCTO
VALUES (292, 'Hugo Boss In Motion Orange', '737052852034', 590000,
501500, 472000, 90, 'N', 219000, 2, 7, 1);
```

```

INSERT INTO PRODUCTO
VALUES (293, 'Ricci Ricci', '3137370310099', 690000, 586500, 552000, 80,
'N', 2937000, 2, 37, 2);

INSERT INTO PRODUCTO
VALUES (294, '212 Sexy (crema)', '6949063982745', 180000, 153000, 144000,
200, 'N', 69000, 2, 57, 2);

INSERT INTO PRODUCTO
VALUES (295, 'Hugo Boss Man', '737052713984', 590000, 501500, 472000,
125, 'N', 255000, 2, 7, 1);

INSERT INTO PRODUCTO
VALUES (296, 'XS Excess pour Homme', '3349668111541', 590000, 501500,
472000, 50, 'N', 255000, 2, 31, 1);

INSERT INTO PRODUCTO
VALUES (297, 'Play in Black Seduction (tester)', '8411061854853', 320000,
272000, 256000, 100, 'N', 60000, 2, 2, 1);

INSERT INTO PRODUCTO
VALUES (298, 'The Golden Secret (tester)', '8411061889534', 320000,
272000, 256000, 80, 'N', 60000, 2, 2, 1);

INSERT INTO PRODUCTO
VALUES (299, 'Power of Seduction', '8411061917510', 280000, 238000,
224000, 50, 'N', 77700, 2, 2, 1);

INSERT INTO PRODUCTO
VALUES (300, 'Police Woman Passion', '679602461108', 490000, 416500,
392000, 100, 'N', 147000, 2, 25, 2);

INSERT INTO PRODUCTO
VALUES (301, 'Flash', '3326240040024', 260000, 221000, 208000, 100, 'N',
52800, 2, 10, 1);

INSERT INTO PRODUCTO
VALUES (302, 'United Dreams', '8433982002250', 360000, 306000, 288000,
100, 'N', 120000, 2, 69, 1);

INSERT INTO PRODUCTO
VALUES (303, 'Essence Gold for Women', '5425039220093', 260000, 221000,
208000, 100, 'N', 59700, 2, 5, 2);

INSERT INTO PRODUCTO
VALUES (304, '212 VIP Rosé', '8411061781197', 390000, 331500, 312000, 30,
'N', 179700, 2, 23, 2);

INSERT INTO PRODUCTO
VALUES (305, 'Britney Spears Fantasy', '719346065405', 590000, 501500,
472000, 100, 'N', 147000, 2, 40, 2);

INSERT INTO PRODUCTO

```

```
VALUES (368, 'NY Scents Acqua Gio', '6915924303370', 280000, 238000, 224000, 100, 'N', 76800, 2, 79, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (307, 'FC Barcelona', '663350073362', 340000, 289000, 272000, 100, 'N', 63000, 2, 39, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (308, 'Black XS', '3349666002506', 620000, 527000, 496000, 50, 'N', 267000, 2, 31, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (309, 'Wanted', '3351500002702', 650000, 552500, 520000, 100, 'N', 291000, 2, 9, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (310, 'Light Blue', '737052074313', 590000, 501500, 472000, 50, 'N', 261000, 2, 46, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (311, 'L''''Eau D''''Issey', '3423470300154', 520000, 442000, 416000, 50, 'N', 219000, 2, 70, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (312, 'Eternity Moment', '088300139507', 650000, 552500, 520000, 100, 'N', 210000, 2, 22, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (314, '212 VIP NYC', '8411061711804', 650000, 552500, 520000, 50, 'N', 293700, 2, 23, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (315, 'Débutante', '875990000596', 260000, 221000, 208000, 100, 'N', 63000, 2, 71, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (316, 'Elixir kit', '8411061769249', 340000, 289000, 272000, 50, 'N', 93000, 2, 13, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (317, '212 VIP Men', '8411061723760', 720000, 612000, 576000, 100, 'N', 327000, 2, 23, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (318, 'Her Secret Temptation', '8411061860410', 380000, 323000, 304000, 80, 'N', 105000, 2, 2, 2);
```

```
INSERT INTO PRODUCTO  
VALUES (320, 'L''''Eau D''''Issey', '3423470311365', 750000, 637500, 600000, 100, 'N', 309000, 2, 70, 1);
```

```
INSERT INTO PRODUCTO  
VALUES (321, '212 NYC', '8411061251706', 690000, 586500, 552000, 60, 'N', 287700, 2, 23, 1);
```



```

INSERT INTO PRODUCTO
VALUES (322, 'Britney Spears Fantasy Anniversary', '719346601887',
590000, 501500, 472000, 100, 'N', 143750, 2, 40, 2);

INSERT INTO PRODUCTO
VALUES (323, 'CHHC Men', '8411061665022', 750000, 637500, 600000, 100,
'N', 338963, 2, 23, 1);

INSERT INTO PRODUCTO
VALUES (324, 'Lady Million', '3349668508587', 850000, 722500, 680000, 80,
'N', 353625, 2, 31, 2);

INSERT INTO PRODUCTO
VALUES (325, 'L''''Eau D''''Issey pour Homme Sport', '3423474867059',
560000, 476000, 448000, 50, 'N', 198375, 2, 70, 1);

INSERT INTO PRODUCTO
VALUES (326, 'Versace Eros', '8011003809202', 590000, 501500, 472000, 50,
'N', 215625, 2, 29, 1);

INSERT INTO PRODUCTO
VALUES (327, 'Spirit for Men', '8411061137017', 380000, 323000, 304000,
100, 'N', 79350, 2, 2, 1);

INSERT INTO PRODUCTO
VALUES (328, 'Pink Topaz', '3454090002890', 260000, 221000, 208000, 100,
'N', 63000, 2, 72, 2);

INSERT INTO PRODUCTO
VALUES (329, 'Billion Woman', '3454090002838', 260000, 221000, 208000,
100, 'N', 63000, 2, 72, 2);

INSERT INTO PRODUCTO
VALUES (330, 'Blue Seduction for women', '8411061839669', 380000, 323000,
304000, 80, 'N', 95700, 2, 2, 2);

INSERT INTO PRODUCTO
VALUES (331, '212 NYC', '8411061865408', 650000, 552500, 520000, 100,
'N', 357000, 2, 23, 1);

INSERT INTO PRODUCTO
VALUES (332, '212 Men NYC', '8411061341605', 650000, 552500, 520000, 100,
'N', 309000, 2, 23, 1);

INSERT INTO PRODUCTO
VALUES (333, 'Dania women', '3700134406040', 360000, 306000, 288000, 100,
'N', 114000, 2, 73, 2);

INSERT INTO PRODUCTO
VALUES (334, 'Cute', '5425017737087', 260000, 221000, 208000, 100, 'N',
46200, 2, 3, 2);

INSERT INTO PRODUCTO

```

VALUES (335, 'Madly', '3274871939784', 490000, 416500, 392000, 50, 'N',  
177000, 2, 61, 2);

INSERT INTO PRODUCTO  
VALUES (336, 'Light Blue', '3423473020516', 850000, 722500, 680000, 125,  
'N', 333000, 2, 46, 2);

INSERT INTO PRODUCTO  
VALUES (337, '212 VIP Rosé', '8911061777176', 850000, 722500, 680000, 80,  
'N', 381000, 2, 23, 2);

INSERT INTO PRODUCTO  
VALUES (338, 'La Vie est Belle', '3605532612768', 890000, 756500, 712000,  
50, 'N', 371700, 2, 74, 2);

INSERT INTO PRODUCTO  
VALUES (339, 'Rayo McQueen', '8411114073866', 290000, 246500, 232000,  
100, 'N', 102000, 2, 39, 1);

INSERT INTO PRODUCTO  
VALUES (340, 'Stronger with You', '3605522040281', 590000, 501500,  
472000, 50, 'N', 264000, 2, 75, 1);

INSERT INTO PRODUCTO  
VALUES (341, 'Lolita Lempicka', '3595200113768', 590000, 501500, 472000,  
100, 'N', 252000, 2, 76, 2);

INSERT INTO PRODUCTO  
VALUES (342, 'Man', '783320971525', 590000, 501500, 472000, 100, 'N',  
250000, 2, 77, 1);

INSERT INTO PRODUCTO  
VALUES (343, 'Elixir', '8411061741696', 280000, 238000, 224000, 50, 'N',  
84000, 2, 13, 2);

INSERT INTO PRODUCTO  
VALUES (344, 'Coconut Passion', '0667538086168', 260000, 221000, 208000,  
250, 'N', 65700, 2, 66, 2);

INSERT INTO PRODUCTO  
VALUES (345, 'Polo blue', '3360377022911', 650000, 552500, 520000, 75,  
'N', 291000, 2, 78, 1);

INSERT INTO PRODUCTO  
VALUES (346, 'Ricci Ricci', '3137370208303', 540000, 459000, 432000, 50,  
'N', 231000, 2, 37, 2);

INSERT INTO PRODUCTO  
VALUES (347, 'Spirit for men', '8411061924693', 280000, 238000, 224000,  
50, 'N', 77700, 2, 2, 1);

INSERT INTO PRODUCTO  
VALUES (348, 'Arsenal red', '3700608320230', 390000, 331500, 312000, 100,  
'N', 100000, 2, 33, 1);

```
INSERT INTO PRODUCTO
VALUES (349, '1 Million Dollar', '6944348501871', 280000, 238000, 224000,
100, 'N', 100000, 2, 79, 1);
```

```
INSERT INTO PRODUCTO
VALUES (350, '777 VVIP men', '6915924304964', 280000, 238000, 224000,
100, 'N', 100000, 2, 79, 1);
```

```
INSERT INTO PRODUCTO
VALUES (351, '717 Men NYC', '6915924302700', 280000, 238000, 224000, 100,
'N', 100000, 2, 79, 1);
```

```
INSERT INTO PRODUCTO
VALUES (352, 'NYC Scents Invictus', '6915924303134', 280000, 238000,
224000, 100, 'N', 100000, 2, 79, 1);
```

```
INSERT INTO PRODUCTO
VALUES (353, 'NYC Scents Olympea', '6915924304988', 280000, 238000,
224000, 80, 'N', 100000, 2, 79, 2);
```

```
INSERT INTO PRODUCTO
VALUES (354, 'Essence Platinum for men', '5425039220086', 260000, 221000,
208000, 100, 'N', 80000, 2, 5, 1);
```

```
INSERT INTO PRODUCTO
VALUES (355, 'NYC Scents Acqua Gio', '6915924303370', 280000, 238000,
224000, 100, 'N', 100000, 2, 79, 1);
```

```
INSERT INTO PRODUCTO
VALUES (356, 'NYC Scents J''''Adore', '6915924303189', 280000, 238000,
224000, 100, 'N', 100000, 2, 79, 2);
```

```
INSERT INTO PRODUCTO
VALUES (357, 'NYC Scents CH', '6915924303035', 280000, 238000, 224000,
100, 'N', 100000, 2, 79, 2);
```

```
INSERT INTO PRODUCTO
VALUES (358, 'Nice Shoe CH Good Girl', '6915924306616', 280000, 238000,
224000, 80, 'N', 100000, 2, 79, 2);
```

```
INSERT INTO PRODUCTO
VALUES (361, 'Bottled Night', '737052352060', 650000, 552500, 520000,
100, 'N', 249000, 2, 38, 1);
```

```
INSERT INTO PRODUCTO
VALUES (359, 'NYC Scents Women Million', '6915924303288', 280000, 238000,
224000, 100, 'N', 76800, 2, 79, 2);
```

```
INSERT INTO PRODUCTO
VALUES (362, 'Scuderia Ferrari Red', '8002135111363', 340000, 289000,
272000, 75, 'N', 102208.5, 2, 12, 1);
```

```
INSERT INTO PRODUCTO
```

```
VALUES (363, 'Eau de Toilette in Minion Box', '663350057850', 260000,
221000, 208000, 100, 'N', 93465, 2, 39, 1);
```

```
INSERT INTO PRODUCTO
VALUES (364, 'Cabotine', '7640111494027', 390000, 331500, 312000, 100,
'N', 81000, 2, 6, 2);
```

```
INSERT INTO PRODUCTO
VALUES (365, 'Only Me! Passion kit', '3442152259122', 380000, 323000,
304000, 100, 'N', 117000, 2, 16, 2);
```

```
INSERT INTO PRODUCTO
VALUES (366, 'Dream', '8411061916964', 290000, 246500, 232000, 50, 'N',
77700, 2, 13, 2);
```

```
INSERT INTO PRODUCTO
VALUES (367, 'One', '088300607402', 490000, 416500, 392000, 100, 'N',
141000, 2, 22, 1);
```

```
INSERT INTO PRODUCTO
VALUES (369, 'Beauty Girl', '6936711318254', 280000, 238000, 224000, 80,
'N', 87000, 2, 79, 1);
```

```
INSERT INTO PRODUCTO
VALUES (370, 'Romantic', '667538086519', 260000, 221000, 208000, 236,
'N', 57000, 2, 66, 1);
```

```
INSERT INTO PRODUCTO
VALUES (371, 'Light Blue', '737052074306', 490000, 416500, 392000, 25,
'N', 213000, 2, 46, 2);
```

```
INSERT INTO PRODUCTO
VALUES (372, 'Pour Homme', '3423473020783', 590000, 501500, 472000, 75,
'N', 245700, 2, 46, 1);
```

```
INSERT INTO PRODUCTO
VALUES (373, 'ange ou démon', '3274870373268', 850000, 722500, 680000,
100, 'N', 372000, 2, 45, 2);
```

```
INSERT INTO PRODUCTO
VALUES (374, 'Diavolo', '8411061849378', 250000, 212500, 200000, 50, 'N',
51000, 2, 2, 1);
```

```
INSERT INTO PRODUCTO
VALUES (375, 'Cabotine kit 2 piezas', '7640163970708', 360000, 306000,
288000, 100, 'N', 141000, 2, 6, 2);
```

```
INSERT INTO PRODUCTO
VALUES (376, 'Olympea', '3349668568093', 750000, 637500, 600000, 50, 'N',
330000, 2, 31, 2);
```

```
INSERT INTO PRODUCTO
VALUES (377, 'Hugo Boss Energise', '737052139890', 750000, 637500,
600000, 125, 'N', 252000, 2, 7, 1);
```

```
INSERT INTO PRODUCTO
VALUES (378, 'Light Blue', '3423473020264', 590000, 501500, 472000, 50,
'N', 261000, 2, 46, 2);
```

```
INSERT INTO PRODUCTO
VALUES (379, '212 Men Sexy', '8411061865613', 590000, 501500, 472000, 50,
'N', 239700, 2, 23, 1);
```

```
INSERT INTO PRODUCTO
VALUES (380, '212 VIP Party Fever', '8411061924099', 850000, 722500,
680000, 80, 'N', 330000, 2, 23, 2);
```

```
INSERT INTO PRODUCTO
VALUES (381, 'Temptation', '667538086359', 260000, 221000, 208000, 236,
'N', 59700, 2, 66, 2);
```

```
INSERT INTO PRODUCTO
VALUES (382, 'Love Love Love', '8410225524595', 260000, 221000, 208000,
50, 'N', 630000, 2, 32, 2);
```

```
INSERT INTO PRODUCTO
VALUES (383, 'Minions Alcañcia', '663350062847', 280000, 238000, 224000,
100, 'N', 106800, 2, 39, 1);
```

```
INSERT INTO PRODUCTO
VALUES (384, 'Eau de Lacoste for her', '737052624471', 620000, 527000,
496000, 90, 'N', 257700, 2, 54, 2);
```

```
INSERT INTO PRODUCTO
VALUES (388, 'Good Girl', '8411061823514', 620000, 527000, 496000, 30,
'N', 263700, 2, 23, 2);
```

```
INSERT INTO PRODUCTO
VALUES (389, 'Spirit for women kit', '8411061918661', 280000, 238000,
224000, 50, 'N', 63000, 2, 2, 2);
```

```
INSERT INTO PRODUCTO
VALUES (386, 'Scuderia Ferrari Black', '8002135111974', 420000, 357000,
336000, 125, 'N', 108000, 2, 12, 1);
```

```
INSERT INTO PRODUCTO
VALUES (390, 'Her Secret Temptation kit', '8411061925065', 280000,
238000, 224000, 50, 'N', 81000, 2, 2, 2);
```

```
INSERT INTO PRODUCTO
VALUES (387, 'Pour Homme', '737052074436', 490000, 416500, 392000, 40,
'N', 183000, 2, 46, 1);
```

```
INSERT INTO PRODUCTO
VALUES (391, 'Acqua di Gio', '3360372058861', 590000, 501500, 472000, 50,
'N', 244800, 2, 60, 1);
```

```
INSERT INTO PRODUCTO
```

VALUES (392, 'Very Irresistible', '3274870352362', 650000, 552500, 520000, 75, 'N', 277090, 2, 45, 2);

INSERT INTO PRODUCTO  
VALUES (393, 'Blue Seduction kit', '8411061918302', 280000, 238000, 224000, 50, 'N', 81675, 2, 2, 1);

INSERT INTO PRODUCTO  
VALUES (394, 'The Golden Secret kit', '8411061920312', 280000, 238000, 224000, 50, 'N', 81675, 2, 2, 1);

INSERT INTO PRODUCTO  
VALUES (395, 'Smash', '8002135144033', 460000, 391000, 368000, 100, 'N', 132797.5, 2, 80, 1);

INSERT INTO PRODUCTO  
VALUES (396, 'Coco Mademoiselle', '3145891164206', 1100000, 935000, 880000, 50, 'N', 466800, 2, 35, 2);

INSERT INTO PRODUCTO  
VALUES (397, 'L''''Interdit', '3274872372153', 1100000, 935000, 880000, 80, 'N', 408000, 2, 45, 2);

INSERT INTO PRODUCTO  
VALUES (398, '1 Million Privé', '3349668535484', 650000, 552500, 520000, 50, 'N', 273000, 2, 31, 1);

INSERT INTO PRODUCTO  
VALUES (399, 'Aqva for Men', '783320913228', 460000, 391000, 368000, 30, 'N', 137700, 2, 77, 1);

INSERT INTO PRODUCTO  
VALUES (400, 'Burberry for Men', '5045252667453', 460000, 391000, 368000, 30, 'N', 123000, 2, 58, 1);

INSERT INTO PRODUCTO  
VALUES (403, 'Kouros', '3365440003842', 520000, 442000, 416000, 50, 'N', 215700, 2, 47, 1);

INSERT INTO PRODUCTO  
VALUES (401, 'Ulric de Varens for men', '3326240816698', 260000, 221000, 208000, 100, 'N', 46200, 2, 10, 1);

INSERT INTO PRODUCTO  
VALUES (402, 'Olympea', '3349668528660', 750000, 637500, 600000, 50, 'N', 263700, 2, 31, 2);

INSERT INTO PRODUCTO  
VALUES (404, 'Minnie', '663350009736', 260000, 221000, 208000, 100, 'N', 81000, 2, 39, 2);

INSERT INTO PRODUCTO  
VALUES (405, 'Britney Spears Fantasy kit', '719346198592', 360000, 306000, 288000, 50, 'N', 110700, 2, 40, 2);

```
INSERT INTO PRODUCTO
VALUES (406, 'Mediterráneo kit', '8411061918487', 280000, 238000, 224000,
50, 'N', 81000, 2, 2, 1);
```

```
INSERT INTO PRODUCTO
VALUES (407, 'Her Golden Secret', '8411061948903', 300000, 255000,
240000, 50, 'N', 87227, 2, 2, 2);
```

```
INSERT INTO PRODUCTO
VALUES (408, 'Blue Seduction for Women', '8411061982617', 300000, 255000,
240000, 50, 'N', 87227, 2, 2, 2);
```

```
INSERT INTO PRODUCTO
VALUES (409, 'Blue Seduction Wave', '8411061029657', 400000, 340000,
320000, 100, 'N', 123144, 2, 2, 1);
```

```
INSERT INTO PRODUCTO
VALUES (410, 'Romantic', '667548800501', 280000, 238000, 224000, 250,
'N', 89426, 2, 66, 2);
```

```
INSERT INTO PRODUCTO
VALUES (411, 'Rock!', '8411061783375', 300000, 255000, 240000, 50, 'N',
87227, 2, 13, 2);
```

```
INSERT INTO PRODUCTO
VALUES (412, 'Story of New Brand', '5425039220482', 280000, 238000,
224000, 100, 'N', 54975, 2, 3, 1);
```

```
INSERT INTO PRODUCTO
VALUES (413, 'Hola', '5425039221489', 280000, 238000, 224000, 100, 'N',
54975, 2, 3, 2);
```

```
INSERT INTO PRODUCTO
VALUES (414, 'Prestige The NB', '5425039220925', 280000, 238000, 224000,
100, 'N', 54975, 2, 3, 1);
```

```
INSERT INTO PRODUCTO
VALUES (415, 'Toy 2', '8011003864089', 1150000, 977500, 920000, 100, 'N',
465400, 2, 81, 2);
```

```
INSERT INTO PRODUCTO
VALUES (416, 'Cabotine kit', '7640171193458', 450000, 382500, 360000,
100, 'N', 150265, 2, 6, 2);
```

```
INSERT INTO PRODUCTO
VALUES (417, 'Cuba Silver', '5425039222141', 260000, 221000, 208000, 100,
'N', 47645, 2, 14, 1);
```

```
COMMIT WORK;
```

```
/* Data for the 'METO_PAGO' table (Records 1 - 5) */
```

```
INSERT INTO METO_PAGO
```

```
VALUES (1, 'Efectivo');
```

```
INSERT INTO METO_PAGO  
VALUES (2, 'Descuento');
```

```
INSERT INTO METO_PAGO  
VALUES (3, 'Devolucion');
```

```
INSERT INTO METO_PAGO  
VALUES (4, 'Otros');
```

```
INSERT INTO METO_PAGO  
VALUES (5, 'Sin pagos');
```

```
COMMIT WORK;
```

```
/* Data for the 'VENTA' table (Records 1 - 2) */
```

```
INSERT INTO VENTA  
VALUES (2, 2, 2, '2017-08-12', 'Semanal', 'N', 2, 2, 'Blues + 25 +  
Semanal', 2);
```

```
INSERT INTO VENTA  
VALUES (3, 1, 7, '2017-12-18', 'Quincenal', 'N', 3, 33, 'Ferrari', 2);
```

```
COMMIT WORK;
```

```
/* Data for the 'MOVIMIENTO' table (Records 1 - 500) */
```

```
INSERT INTO MOVIMIENTO  
VALUES (777, 1, '2019-02-16', 'N', 'S');
```

```
COMMIT WORK;
```

```
/* Data for the 'MOVIMIENTO_DETALLE' table (Records 1 - 500) */
```

```
INSERT INTO MOVIMIENTO_DETALLE  
VALUES (1, 1, 2, 1, 280000, 250000, 145000, 50, 'N', '!8411061636275',  
0);
```

```
INSERT INTO MOVIMIENTO_DETALLE  
VALUES (24377, 777, 406, 1, 280000, 238000, 224000, 50, 'N',  
'8411061918487', 0);
```

```
COMMIT WORK;
```

```
/* Data for the 'EMPRESA' table (Records 1 - 1) */
```

```
INSERT INTO EMPRESA (ID_EMPRESA, RAZON, DIRECCION, ID_CIUDAD, TELEFONO1,  
TELEFONO2, FAX1, FAX2, EMAIL, WEB, REGISTRO, LINEA1, LINEA2, IDEOLOGIA,
```



```
MEMBRETE_HACIENDA, NOMBRE, OFRECER_PROMOCION, PROMOCION_LINEA,
PROMOCION_LINEA2, MAXIMO_ITEMS_VENTA, MAXIMO_ITEMS_TRANFERENCIA,
MAXIMO_ITEMS_ENTRADA, MAXIMO_ITEMS_SALIDA, MAXIMO_ITEMS_COMPRA,
PERMITIR_AUTO_CARGA, PANTALLA_MODAL, COMPROBAR_BA_PRO, COMPROBAR_DO_CLI,
CAN_INSERT_INCOM, GUARDAR_EVENTOSS, ID LENGUAJE, NOMBRE_SERVIDOR,
PUERTO_SERVIDOR, CONTRASE_EMAIL, EMAILDESALDOBIENVENIDA, CONTROLARSTOCK)
VALUES (1, 'Gabriela Fragancias', 'Calle San Miguel 105', 35,
'+595972260891', '+595986794774', NULL, NULL, NULL, NULL, NULL, 'Vuelva
siempre', NULL, 'Por favor si los datos suyos no concuerdan avisar a los
telefonos 0971 399430.\r\nSu Nombre completo y su fecha de
cumpleaños.\r\nGracias', 'S', NULL, 'S', 'Super promoción de puntos',
NULL, 250000, 1000, 1000, 1000, 60, 'N', 'N', 'S', 'S', 'S', 'S', 1,
'mail.erp-solucion.com', 465, 'erpsolucion', 'N', 'S');
```

```
COMMIT WORK;
```

```
/* Data for the 'PRODUCTO_MARCA' table (Records 1 - 79) */
```

```
INSERT INTO PRODUCTO_MARCA
VALUES (2, 'Antonio Banderas');
```

```
INSERT INTO PRODUCTO_MARCA
VALUES (3, 'New Brand Perfumes');
```

```
INSERT INTO PRODUCTO_MARCA
VALUES (4, 'Chic '''N Glam Perfumes');
```

```
INSERT INTO PRODUCTO_MARCA
VALUES (32, 'Ágata Ruiz de la Prada');
```

```
INSERT INTO PRODUCTO_MARCA
VALUES (5, 'Master of New Brand');
```

```
INSERT INTO PRODUCTO_MARCA
VALUES (6, 'Grès');
```

```
INSERT INTO PRODUCTO_MARCA
VALUES (7, 'P&G Prestige Beauté');
```

```
INSERT INTO PRODUCTO_MARCA
VALUES (8, 'Ted Lapidus');
```

```
INSERT INTO PRODUCTO_MARCA
VALUES (9, 'Azzaro');
```

```
INSERT INTO PRODUCTO_MARCA
VALUES (10, 'Ulric de Varens');
```

```
INSERT INTO PRODUCTO_MARCA
VALUES (11, 'Adidas');
```

```
INSERT INTO PRODUCTO_MARCA
VALUES (12, 'Ferrari Fragrances');
```

```
INSERT INTO PRODUCTO_MARCA
VALUES (13, 'Shakira');

INSERT INTO PRODUCTO_MARCA
VALUES (14, 'Parfums Des Champs');

INSERT INTO PRODUCTO_MARCA
VALUES (15, 'Jesús Del Pozo');

INSERT INTO PRODUCTO_MARCA
VALUES (16, 'Yves de Sistelle');

INSERT INTO PRODUCTO_MARCA
VALUES (17, 'Emper');

INSERT INTO PRODUCTO_MARCA
VALUES (18, 'Alta Moda');

INSERT INTO PRODUCTO_MARCA
VALUES (19, 'Pacha Ibiza');

INSERT INTO PRODUCTO_MARCA
VALUES (20, 'Christine Darvin');

INSERT INTO PRODUCTO_MARCA
VALUES (21, 'Omerta');

INSERT INTO PRODUCTO_MARCA
VALUES (22, 'Calvin Klein');

INSERT INTO PRODUCTO_MARCA
VALUES (33, 'Gilles Cantuel');

INSERT INTO PRODUCTO_MARCA
VALUES (23, 'Carolina Herrera');

INSERT INTO PRODUCTO_MARCA
VALUES (24, 'Puig');

INSERT INTO PRODUCTO_MARCA
VALUES (25, 'Mavive');

INSERT INTO PRODUCTO_MARCA
VALUES (26, 'Elizabeth Arden');

INSERT INTO PRODUCTO_MARCA
VALUES (27, 'Animale Group');

INSERT INTO PRODUCTO_MARCA
VALUES (28, 'Carlotta Zuofun');

INSERT INTO PRODUCTO_MARCA
VALUES (29, 'Euroitalia');
```

```
INSERT INTO PRODUCTO_MARCA
VALUES (30, 'Parfums Café');

INSERT INTO PRODUCTO_MARCA
VALUES (31, 'Paco Rabanne');

INSERT INTO PRODUCTO_MARCA
VALUES (34, 'Stella Dustin');

INSERT INTO PRODUCTO_MARCA
VALUES (35, 'Chanel');

INSERT INTO PRODUCTO_MARCA
VALUES (36, 'Karen Low');

INSERT INTO PRODUCTO_MARCA
VALUES (37, 'Nina Ricci');

INSERT INTO PRODUCTO_MARCA
VALUES (38, 'Hugo Boss');

INSERT INTO PRODUCTO_MARCA
VALUES (39, 'Air-Val International');

INSERT INTO PRODUCTO_MARCA
VALUES (40, 'EA Fragrances');

INSERT INTO PRODUCTO_MARCA
VALUES (41, 'Estelle Ewen');

INSERT INTO PRODUCTO_MARCA
VALUES (42, 'Cofinluxe');

INSERT INTO PRODUCTO_MARCA
VALUES (43, 'Cacharel');

INSERT INTO PRODUCTO_MARCA
VALUES (44, 'Tommy Hilfiger');

INSERT INTO PRODUCTO_MARCA
VALUES (45, 'Givenchy');

INSERT INTO PRODUCTO_MARCA
VALUES (46, 'Dolce & Gabbana');

INSERT INTO PRODUCTO_MARCA
VALUES (47, 'Yves Saint Laurent');

INSERT INTO PRODUCTO_MARCA
VALUES (48, 'Lanvin Parfums');

INSERT INTO PRODUCTO_MARCA
VALUES (49, 'Jeanne Arthes');
```

```
INSERT INTO PRODUCTO_MARCA
VALUES (50, 'Christian Dior');

INSERT INTO PRODUCTO_MARCA
VALUES (51, 'Marina de Bourbon');

INSERT INTO PRODUCTO_MARCA
VALUES (54, 'Lacoste');

INSERT INTO PRODUCTO_MARCA
VALUES (53, 'Sapil');

INSERT INTO PRODUCTO_MARCA
VALUES (55, 'Fragantia Secrets');

INSERT INTO PRODUCTO_MARCA
VALUES (56, 'Via Paris Parfums');

INSERT INTO PRODUCTO_MARCA
VALUES (57, 'Sex in the City');

INSERT INTO PRODUCTO_MARCA
VALUES (59, 'Guy Laroche');

INSERT INTO PRODUCTO_MARCA
VALUES (60, 'Giorgio Armani');

INSERT INTO PRODUCTO_MARCA
VALUES (58, 'Burberry Fragances');

INSERT INTO PRODUCTO_MARCA
VALUES (61, 'Kenzo');

INSERT INTO PRODUCTO_MARCA
VALUES (62, 'Unforgettable Moments');

INSERT INTO PRODUCTO_MARCA
VALUES (63, 'INCC');

INSERT INTO PRODUCTO_MARCA
VALUES (64, 'Give Back Brands');

INSERT INTO PRODUCTO_MARCA
VALUES (65, 'Cadillac');

INSERT INTO PRODUCTO_MARCA
VALUES (66, 'Victoria''''s Secret');

INSERT INTO PRODUCTO_MARCA
VALUES (67, 'Jean Paul Gaultier');

INSERT INTO PRODUCTO_MARCA
VALUES (68, 'Interparfums');
```

```
INSERT INTO PRODUCTO_MARCA
VALUES (69, 'Benetton');

INSERT INTO PRODUCTO_MARCA
VALUES (70, 'Issey Miyake');

INSERT INTO PRODUCTO_MARCA
VALUES (71, 'Nuparfums Group');

INSERT INTO PRODUCTO_MARCA
VALUES (72, 'Paris Elysees');

INSERT INTO PRODUCTO_MARCA
VALUES (73, 'LAK Paris');

INSERT INTO PRODUCTO_MARCA
VALUES (74, 'Lancome');

INSERT INTO PRODUCTO_MARCA
VALUES (75, 'Emporio Armani');

INSERT INTO PRODUCTO_MARCA
VALUES (76, 'Pacific Creation');

INSERT INTO PRODUCTO_MARCA
VALUES (77, 'Bulgari');

INSERT INTO PRODUCTO_MARCA
VALUES (78, 'Ralph Lauren');

INSERT INTO PRODUCTO_MARCA
VALUES (79, 'Parfums');

INSERT INTO PRODUCTO_MARCA
VALUES (80, 'Sergio Tacchini');

INSERT INTO PRODUCTO_MARCA
VALUES (81, 'Moschino');
```

```
COMMIT WORK;
```

```
/* Data for the 'PRODUCTO_SEXO' table (Records 1 - 2) */
```

```
INSERT INTO PRODUCTO_SEXO
VALUES (1, 'MASCULINO');

INSERT INTO PRODUCTO_SEXO
VALUES (2, 'FEMENINO');
```

```
COMMIT WORK;
```

```
/* Data for the 'PRODUCTO_TIPO' table (Records 1 - 2) */
```

```
INSERT INTO PRODUCTO_TIPO
```

```

VALUES (1, 'General');

INSERT INTO PRODUCTO_TIPO
VALUES (2, 'Perfumes');

COMMIT WORK;

/* Data for the 'SIS' table (Records 1 - 3) */

INSERT INTO SIS
VALUES ('Comercial', 2, 7, 3, 6, 7, 9, 5, 4, 3, 8);

INSERT INTO SIS
VALUES ('Financiera', 5, 6, 7, 9, 2, 3, 7, 9, 13, 5);

INSERT INTO SIS
VALUES ('Contabilidad', 8, 2, 5, 4, 9, 23, 55, 8, 23, 4);

COMMIT WORK;

/* Data for the 'CLIENTE_LLAMADA' table (Records 1 - 1) */

INSERT INTO CLIENTE_LLAMADA (ID_CLIENTE_LLAMADA, ID_CLIENTE,
ID_MOTIVO_LLAMADA, FECHA, HORA, MEMO, AGENDAR, AGENDAR_FECHA, TERMINADO,
NUMERO, ID_USUARIO)
VALUES (1, 2, 1, '2017-08-12', '17:58:21',
'sdjfñlsadjñfsjadñfljsdñlfsjd\r\nfsdjfklasjdñf', 'N', NULL, 'S',
'09785785', 2);

COMMIT WORK;

/* Data for the 'PROVEEDOR' table (Records 1 - 4) */

INSERT INTO PROVEEDOR
VALUES (1, '123456789', 'Lucas Perfumes', '0522-40761', NULL, 'Avda.
Gral. Bernardino Caballero c/ Prof. Delfín Chamorro', 'Lorenzo Molinas',
NULL);

INSERT INTO PROVEEDOR
VALUES (3, '2315678', 'Elyzée', NULL, NULL, 'Galería Jebai
Center\r\nAvda. Carlos Antonio López y Monseñor Rodríguez\r\nCiudad del
Este', 'Nilda', NULL);

INSERT INTO PROVEEDOR
VALUES (4, '80044020-0', 'Mile Perfumería', '0336-274356', NULL, '14 de
Mayo N° 142 e/ Mcal. López y Carlos Antonio López.\r\nPedro Juan
Caballero - Amambay', 'Terese del Niño Jesús Ayala Barreto', NULL);

INSERT INTO PROVEEDOR
VALUES (5, '80051827-6', 'Ñanduti Representaciones', '061-502090', NULL,
'Galería Jebai Center\r\nAvda. Carlos Antonio López y Monseñor
Rodríguez\r\n3er Piso Salones 2204-2205\r\n1er Piso Salones 2005-
2006\r\nCiudad del Este', NULL, NULL);

```

```
COMMIT WORK;
```

```
/* Data for the 'TIPO_PRECIO' table (Records 1 - 3) */
```

```
INSERT INTO TIPO_PRECIO  
VALUES (1, 'Precio 1');
```

```
INSERT INTO TIPO_PRECIO  
VALUES (2, 'Precio 2');
```

```
INSERT INTO TIPO_PRECIO  
VALUES (3, 'Precio 3');
```

```
COMMIT WORK;
```

```
/* Data for the 'CLIENTE_SEGUI' table (Records 1 - 4) */
```

```
INSERT INTO CLIENTE_SEGUI  
VALUES (1, 2, 2, '2017-08-12', '17:53:47', 'para la proxima quiere pagar  
todo', NULL, NULL);
```

```
INSERT INTO CLIENTE_SEGUI  
VALUES (2, 2, 2, '2017-08-12', '17:57:23', 'No pago todo', NULL, NULL);
```

```
INSERT INTO CLIENTE_SEGUI  
VALUES (3, 7, 2, '2018-03-19', '10:17:20', 'Dijo que pagarianfgrhjr',  
NULL, NULL);
```

```
INSERT INTO CLIENTE_SEGUI  
VALUES (4, 7, 2, '2018-03-19', '10:30:12', 'pagara el doble la proxima',  
NULL, NULL);
```

```
COMMIT WORK;
```

```
/* Data for the 'UCTABPERFIL' table (Records 1 - 96) */
```

```
INSERT INTO UCTABPERFIL  
VALUES (51, 2, 'FrmListadodeLlamadas', 'GrillaResultado',  
'GrillaResultadoDBTableView1ID_CLIENTE_LLAMADA', 'S', 40, 0);
```

```
INSERT INTO UCTABPERFIL  
VALUES (52, 2, 'FrmListadodeLlamadas', 'GrillaResultado',  
'GrillaResultadoDBTableView1DESCRIPCION1', 'S', 78, 1);
```

```
INSERT INTO UCTABPERFIL  
VALUES (53, 2, 'FrmListadodeLlamadas', 'GrillaResultado',  
'GrillaResultadoDBTableView1NOMBRE1', 'S', 50, 2);
```

```
INSERT INTO UCTABPERFIL  
VALUES (54, 2, 'FrmListadodeLlamadas', 'GrillaResultado',  
'GrillaResultadoDBTableView1Column1', 'S', 143, 3);
```

```
INSERT INTO UCTABPERFIL
```

```
VALUES (55, 2, 'FrmListadodeLlamadas', 'GrillaResultado',  
'GrillaResultadoDBTableView1Column3', 'S', 95, 4);
```

```
INSERT INTO UCTABPERFIL  
VALUES (56, 2, 'FrmListadodeLlamadas', 'GrillaResultado',  
'GrillaResultadoDBTableView1DESCRIPCION', 'S', 82, 5);
```

```
INSERT INTO UCTABPERFIL  
VALUES (57, 2, 'FrmListadodeLlamadas', 'GrillaResultado',  
'GrillaResultadoDBTableView1FECHA', 'S', 61, 6);
```

```
INSERT INTO UCTABPERFIL  
VALUES (58, 2, 'FrmListadodeLlamadas', 'GrillaResultado',  
'GrillaResultadoDBTableView1HORA', 'S', 70, 7);
```

```
INSERT INTO UCTABPERFIL  
VALUES (59, 2, 'FrmListadodeLlamadas', 'GrillaResultado',  
'GrillaResultadoDBTableView1RUC', 'S', 48, 8);
```

```
INSERT INTO UCTABPERFIL  
VALUES (60, 2, 'FrmListadodeLlamadas', 'GrillaResultado',  
'GrillaResultadoDBTableView1MEMO', 'S', 263, 9);
```

```
INSERT INTO UCTABPERFIL  
VALUES (61, 2, 'FrmListadodeLlamadas', 'GrillaResultado',  
'GrillaResultadoDBTableView1AGENDAR', 'S', 62, 10);
```

```
INSERT INTO UCTABPERFIL  
VALUES (62, 2, 'FrmListadodeLlamadas', 'GrillaResultado',  
'GrillaResultadoDBTableView1AGENDAR_FECHA', 'S', 101, 11);
```

```
INSERT INTO UCTABPERFIL  
VALUES (63, 2, 'FrmListadodeLlamadas', 'GrillaResultado',  
'GrillaResultadoDBTableView1TERMINADO', 'S', 85, 12);
```

```
INSERT INTO UCTABPERFIL  
VALUES (64, 2, 'FrmListadodeLlamadas', 'GrillaResultado',  
'GrillaResultadoDBTableView1CELULAR', 'S', 49, 13);
```

```
INSERT INTO UCTABPERFIL  
VALUES (65, 2, 'FrmListadodeLlamadas', 'GrillaResultado',  
'GrillaResultadoDBTableView1TELEFONO_1', 'S', 55, 14);
```

```
INSERT INTO UCTABPERFIL  
VALUES (66, 2, 'FrmListadodeLlamadas', 'GrillaResultado',  
'GrillaResultadoDBTableView1TELEFONO_2', 'S', 62, 15);
```

```
INSERT INTO UCTABPERFIL  
VALUES (67, 2, 'FrmListadodeLlamadas', 'GrillaResultado',  
'GrillaResultadoDBTableView1EMAIL', 'S', 78, 16);
```

```
INSERT INTO UCTABPERFIL  
VALUES (68, 2, 'FrmListadodeLlamadas', 'GrillaResultado',  
'GrillaResultadoDBTableView1Column2', 'S', 114, 17);
```



```

INSERT INTO UCTABPERFIL
VALUES (79, 1, 'FrmListadodeClientes', 'cxGrid1',
'cxGrid1DBTableView1AI_CLIENTE', 'S', 59, 0);

INSERT INTO UCTABPERFIL
VALUES (80, 1, 'FrmListadodeClientes', 'cxGrid1',
'cxGrid1DBTableView1OrdenVisita', 'S', 75, 1);

INSERT INTO UCTABPERFIL
VALUES (81, 1, 'FrmListadodeClientes', 'cxGrid1',
'cxGrid1DBTableView1RUC', 'S', 108, 2);

INSERT INTO UCTABPERFIL
VALUES (82, 1, 'FrmListadodeClientes', 'cxGrid1',
'cxGrid1DBTableView1RazonSocial', 'S', 182, 3);

INSERT INTO UCTABPERFIL
VALUES (83, 1, 'FrmListadodeClientes', 'cxGrid1',
'cxGrid1DBTableView1BARRIO', 'S', 98, 4);

INSERT INTO UCTABPERFIL
VALUES (16, 2, 'FrmCuentaporGrupodeClientes', 'GrillaResultado',
'GrillaResultadoDBTableView1AI_CLIENTE', 'S', 52, 0);

INSERT INTO UCTABPERFIL
VALUES (17, 2, 'FrmCuentaporGrupodeClientes', 'GrillaResultado',
'GrillaResultadoDBTableView1Column7', 'S', 71, 1);

INSERT INTO UCTABPERFIL
VALUES (18, 2, 'FrmCuentaporGrupodeClientes', 'GrillaResultado',
'GrillaResultadoDBTableView1RAZONSOCIAL', 'S', 168, 2);

INSERT INTO UCTABPERFIL
VALUES (19, 2, 'FrmCuentaporGrupodeClientes', 'GrillaResultado',
'GrillaResultadoDBTableView1DESCRIPCION', 'S', 126, 3);

INSERT INTO UCTABPERFIL
VALUES (20, 2, 'FrmCuentaporGrupodeClientes', 'GrillaResultado',
'GrillaResultadoDBTableView1RUC', 'S', 53, 4);

INSERT INTO UCTABPERFIL
VALUES (21, 2, 'FrmCuentaporGrupodeClientes', 'GrillaResultado',
'GrillaResultadoDBTableView1CELULAR', 'S', 92, 5);

INSERT INTO UCTABPERFIL
VALUES (22, 2, 'FrmCuentaporGrupodeClientes', 'GrillaResultado',
'GrillaResultadoDBTableView1DIRECCION', 'S', 158, 6);

INSERT INTO UCTABPERFIL
VALUES (23, 2, 'FrmCuentaporGrupodeClientes', 'GrillaResultado',
'GrillaResultadoDBTableView1PRODUCTOVENDIDO', 'S', 134, 7);

INSERT INTO UCTABPERFIL

```

```
VALUES (24, 2, 'FrmCuentaporGrupodeClientes', 'GrillaResultado',  
'GrillaResultadoDBTableView1Column6', 'S', 49, 8);
```

```
INSERT INTO UCTABPERFIL  
VALUES (25, 2, 'FrmCuentaporGrupodeClientes', 'GrillaResultado',  
'GrillaResultadoDBTableView1CANTIDADVECIDO', 'S', 51, 9);
```

```
INSERT INTO UCTABPERFIL  
VALUES (26, 2, 'FrmCuentaporGrupodeClientes', 'GrillaResultado',  
'GrillaResultadoDBTableView1MONTOVENCIDO', 'S', 63, 10);
```

```
INSERT INTO UCTABPERFIL  
VALUES (27, 2, 'FrmCuentaporGrupodeClientes', 'GrillaResultado',  
'GrillaResultadoDBTableView1MONTOAVENCER', 'S', 93, 11);
```

```
INSERT INTO UCTABPERFIL  
VALUES (28, 2, 'FrmCuentaporGrupodeClientes', 'GrillaResultado',  
'GrillaResultadoDBTableView1Column1', 'S', 64, 12);
```

```
INSERT INTO UCTABPERFIL  
VALUES (29, 2, 'FrmCuentaporGrupodeClientes', 'GrillaResultado',  
'GrillaResultadoDBTableView1Column2', 'S', 64, 13);
```

```
INSERT INTO UCTABPERFIL  
VALUES (30, 2, 'FrmCuentaporGrupodeClientes', 'GrillaResultado',  
'GrillaResultadoDBTableView1Column3', 'S', 63, 14);
```

```
INSERT INTO UCTABPERFIL  
VALUES (31, 2, 'FrmCuentaporGrupodeClientes', 'GrillaResultado',  
'GrillaResultadoDBTableView1Column4', 'S', 64, 15);
```

```
INSERT INTO UCTABPERFIL  
VALUES (32, 2, 'FrmCuentaporGrupodeClientes', 'GrillaResultado',  
'GrillaResultadoDBTableView1Column5', 'S', 64, 16);
```

```
INSERT INTO UCTABPERFIL  
VALUES (33, 2, 'FrmCuentaporGrupodeClientes', 'GrillaResultado',  
'GrillaResultadoDBTableView1Column8', 'S', 59, 17);
```

```
INSERT INTO UCTABPERFIL  
VALUES (34, 2, 'FrmCuentaporGrupodeClientes', 'GrillaResultado',  
'GrillaResultadoDBTableView1Column9', 'S', 56, 18);
```

```
INSERT INTO UCTABPERFIL  
VALUES (35, 2, 'FrmCuentaporGrupodeClientes', 'GrillaResultado',  
'GrillaResultadoDBTableView1Column10', 'S', 67, 19);
```

```
INSERT INTO UCTABPERFIL  
VALUES (36, 2, 'FrmCuentaporGrupodeClientes', 'GrillaResultado',  
'GrillaResultadoDBTableView1Column12', 'S', 229, 21);
```

```
INSERT INTO UCTABPERFIL  
VALUES (37, 2, 'FrmCuentaporGrupodeClientes', 'GrillaResultado',  
'GrillaResultadoDBTableView1Column13', 'S', 88, 22);
```

```

INSERT INTO UCTABPERFIL
VALUES (84, 1, 'FrmListadodeClientes', 'cxGrid1',
'cxGrid1DBTableView1FechaNacimiento', 'S', 22, 5);

INSERT INTO UCTABPERFIL
VALUES (85, 1, 'FrmListadodeClientes', 'cxGrid1',
'cxGrid1DBTableView1DESCRIPCIONGRUPO', 'S', 109, 6);

INSERT INTO UCTABPERFIL
VALUES (86, 1, 'FrmListadodeClientes', 'cxGrid1',
'cxGrid1DBTableView1TELEFONO', 'S', 111, 7);

INSERT INTO UCTABPERFIL
VALUES (87, 1, 'FrmListadodeClientes', 'cxGrid1',
'cxGrid1DBTableView1EMAIL', 'S', 82, 8);

INSERT INTO UCTABPERFIL
VALUES (88, 1, 'FrmListadodeClientes', 'cxGrid1',
'cxGrid1DBTableView1Deuda', 'S', 64, 13);

INSERT INTO UCTABPERFIL
VALUES (69, 2, 'FrmExtractodeunCliente', 'cxGrid1',
'cxGrid1DBTableView1ID_CLIENTE', 'S', 58, 0);

INSERT INTO UCTABPERFIL
VALUES (70, 2, 'FrmExtractodeunCliente', 'cxGrid1',
'cxGrid1DBTableView1FECHA', 'S', 103, 1);

INSERT INTO UCTABPERFIL
VALUES (71, 2, 'FrmExtractodeunCliente', 'cxGrid1',
'cxGrid1DBTableView1DESCRIPCION', 'S', 245, 2);

INSERT INTO UCTABPERFIL
VALUES (72, 2, 'FrmExtractodeunCliente', 'cxGrid1',
'cxGrid1DBTableView1NRO', 'S', 146, 3);

INSERT INTO UCTABPERFIL
VALUES (73, 2, 'FrmExtractodeunCliente', 'cxGrid1',
'cxGrid1DBTableView1DEBITO', 'S', 94, 4);

INSERT INTO UCTABPERFIL
VALUES (74, 2, 'FrmExtractodeunCliente', 'cxGrid1',
'cxGrid1DBTableView1CREDITO', 'S', 91, 5);

INSERT INTO UCTABPERFIL
VALUES (75, 2, 'FrmExtractodeunCliente', 'cxGrid1',
'cxGrid1DBTableView1VENCIMIENTO', 'S', 115, 6);

INSERT INTO UCTABPERFIL
VALUES (76, 2, 'FrmExtractodeunCliente', 'cxGrid1',
'cxGrid1DBTableView1DE_HASTA', 'S', 154, 7);

INSERT INTO UCTABPERFIL

```

```

VALUES (77, 2, 'FrmExtractodeunCliente', 'cxGrid1',
'cxGrid1DBTableView1SALDOLINEAL', 'S', 93, 8);

INSERT INTO UCTABPERFIL
VALUES (38, 2, 'FrmListadeRecibosdeClientes', 'GrillaResultado',
'GrillaResultadoDBTableView1ID_COBRANZA', 'S', 106, 0);

INSERT INTO UCTABPERFIL
VALUES (39, 2, 'FrmListadeRecibosdeClientes', 'GrillaResultado',
'GrillaResultadoDBTableView1FECHA', 'S', 69, 1);

INSERT INTO UCTABPERFIL
VALUES (40, 2, 'FrmListadeRecibosdeClientes', 'GrillaResultado',
'GrillaResultadoDBTableView1Column2', 'S', 119, 2);

INSERT INTO UCTABPERFIL
VALUES (41, 2, 'FrmListadeRecibosdeClientes', 'GrillaResultado',
'GrillaResultadoDBTableView1NOMBRE', 'S', 157, 3);

INSERT INTO UCTABPERFIL
VALUES (42, 2, 'FrmListadeRecibosdeClientes', 'GrillaResultado',
'GrillaResultadoDBTableView1COMENTARIO', 'S', 276, 4);

INSERT INTO UCTABPERFIL
VALUES (43, 2, 'FrmListadeRecibosdeClientes', 'GrillaResultado',
'GrillaResultadoDBTableView1Column5', 'S', 108, 5);

INSERT INTO UCTABPERFIL
VALUES (44, 2, 'FrmListadeRecibosdeClientes', 'GrillaResultado',
'GrillaResultadoDBTableView1TOTAL', 'S', 120, 6);

INSERT INTO UCTABPERFIL
VALUES (45, 2, 'FrmListadeRecibosdeClientes', 'GrillaResultado',
'GrillaResultadoDBTableView1Column6', 'S', 114, 7);

INSERT INTO UCTABPERFIL
VALUES (46, 2, 'FrmListadeRecibosdeClientes', 'GrillaResultado',
'GrillaResultadoDBTableView1UCUSERNAME', 'S', 108, 8);

INSERT INTO UCTABPERFIL
VALUES (47, 2, 'FrmListadeRecibosdeClientes', 'GrillaResultado',
'GrillaResultadoDBTableView1GUARDADO', 'S', 76, 9);

INSERT INTO UCTABPERFIL
VALUES (48, 2, 'FrmListadeRecibosdeClientes', 'GrillaResultado',
'GrillaResultadoDBTableView1Column1', 'S', 74, 10);

INSERT INTO UCTABPERFIL
VALUES (49, 2, 'FrmListadeRecibosdeClientes', 'GrillaResultado',
'GrillaResultadoDBTableView1Column3', 'S', 100, 11);

INSERT INTO UCTABPERFIL
VALUES (50, 2, 'FrmListadeRecibosdeClientes', 'GrillaResultado',
'GrillaResultadoDBTableView1Column4', 'S', 87, 12);

```

```
INSERT INTO UCTABPERFIL
VALUES (78, 2, 'FrmExtractodeunCliente', 'cxGrid1',
'cxGrid1DBTableView1NRODERECIBO', 'S', 143, 9);
```

```
INSERT INTO UCTABPERFIL
VALUES (89, 1, 'FrmListadodeClientes', 'cxGrid1',
'cxGrid1DBTableView1Pagos', 'S', 76, 14);
```

```
INSERT INTO UCTABPERFIL
VALUES (90, 1, 'FrmListadodeClientes', 'cxGrid1',
'cxGrid1DBTableView1PagosPorcentaje', 'S', 76, 15);
```

```
INSERT INTO UCTABPERFIL
VALUES (91, 1, 'FrmListadodeClientes', 'cxGrid1',
'cxGrid1DBTableView1Saldo', 'S', 83, 16);
```

```
INSERT INTO UCTABPERFIL
VALUES (92, 1, 'FrmListadodeClientes', 'cxGrid1',
'cxGrid1DBTableView1Column10', 'S', 119, 18);
```

```
INSERT INTO UCTABPERFIL
VALUES (93, 1, 'FrmListadodeClientes', 'cxGrid1',
'cxGrid1DBTableView1Column1', 'S', 291, 23);
```

```
INSERT INTO UCTABPERFIL
VALUES (94, 1, 'FrmListadodeClientes', 'cxGrid1',
'cxGrid1DBTableView1Column2', 'S', 373, 24);
```

```
INSERT INTO UCTABPERFIL
VALUES (95, 1, 'FrmListadodeClientes', 'cxGrid1',
'cxGrid1DBTableView1Column5', 'S', 78, 25);
```

```
INSERT INTO UCTABPERFIL
VALUES (96, 1, 'FrmListadodeClientes', 'cxGrid1',
'cxGrid1DBTableView1Column6', 'S', 64, 26);
```

```
INSERT INTO UCTABPERFIL
VALUES (97, 2, 'FrmListadodeClientes', 'cxGrid1',
'cxGrid1DBTableView1AI_CLIENTE', 'S', 44, 0);
```

```
INSERT INTO UCTABPERFIL
VALUES (98, 2, 'FrmListadodeClientes', 'cxGrid1',
'cxGrid1DBTableView1OrdenVisita', 'S', 55, 1);
```

```
INSERT INTO UCTABPERFIL
VALUES (99, 2, 'FrmListadodeClientes', 'cxGrid1',
'cxGrid1DBTableView1RUC', 'S', 80, 2);
```

```
INSERT INTO UCTABPERFIL
VALUES (100, 2, 'FrmListadodeClientes', 'cxGrid1',
'cxGrid1DBTableView1RazonSocial', 'S', 264, 3);
```

```
INSERT INTO UCTABPERFIL
```

```
VALUES (101, 2, 'FrmListadodeClientes', 'cxGrid1',  
'cxGrid1DBTableView1BARRIO', 'S', 116, 4);
```

```
INSERT INTO UCTABPERFIL  
VALUES (102, 2, 'FrmListadodeClientes', 'cxGrid1',  
'cxGrid1DBTableView1DESCRIPCIONGRUPO', 'S', 66, 6);
```

```
INSERT INTO UCTABPERFIL  
VALUES (103, 2, 'FrmListadodeClientes', 'cxGrid1',  
'cxGrid1DBTableView1TELEFONO', 'S', 69, 7);
```

```
INSERT INTO UCTABPERFIL  
VALUES (104, 2, 'FrmListadodeClientes', 'cxGrid1',  
'cxGrid1DBTableView1CELULAR', 'S', 71, 9);
```

```
INSERT INTO UCTABPERFIL  
VALUES (105, 2, 'FrmListadodeClientes', 'cxGrid1',  
'cxGrid1DBTableView1Deuda', 'S', 40, 13);
```

```
INSERT INTO UCTABPERFIL  
VALUES (106, 2, 'FrmListadodeClientes', 'cxGrid1',  
'cxGrid1DBTableView1Pagos', 'S', 44, 14);
```

```
INSERT INTO UCTABPERFIL  
VALUES (107, 2, 'FrmListadodeClientes', 'cxGrid1',  
'cxGrid1DBTableView1PagosPorcentaje', 'S', 46, 15);
```

```
INSERT INTO UCTABPERFIL  
VALUES (108, 2, 'FrmListadodeClientes', 'cxGrid1',  
'cxGrid1DBTableView1Saldo', 'S', 48, 16);
```

```
INSERT INTO UCTABPERFIL  
VALUES (109, 2, 'FrmListadodeClientes', 'cxGrid1',  
'cxGrid1DBTableView1Column10', 'S', 71, 18);
```

```
INSERT INTO UCTABPERFIL  
VALUES (110, 2, 'FrmListadodeClientes', 'cxGrid1',  
'cxGrid1DBTableView1Column1', 'S', 173, 23);
```

```
INSERT INTO UCTABPERFIL  
VALUES (111, 2, 'FrmListadodeClientes', 'cxGrid1',  
'cxGrid1DBTableView1Column2', 'S', 222, 24);
```

```
COMMIT WORK;
```

```
/* Data for the 'MOTIVO_LLAMADA' table (Records 1 - 2) */
```

```
INSERT INTO MOTIVO_LLAMADA  
VALUES (1, 'Venta telefonica');
```

```
INSERT INTO MOTIVO_LLAMADA  
VALUES (2, 'Requerimiento');
```

```
COMMIT WORK;
```

```

/* Data for the 'UCTABRIGHTS' table (Records 1 - 86) */

INSERT INTO UCTABRIGHTS
VALUES (1, 'Tarjetas', 'actusuario', 'xorPZCKnmwSqYsL');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Tarjetas', 'actperfil', 'xorPZWanUlU3CB');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Tarjetas', 'actcambiarcontrasena',
'xorPZaLzgAMiXrl4oLFT1eLEnwS9');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'Ayuda', 'ys+AQROu');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'CambioContrasena', 'ykOvwhAyjlWxoHHxr7eg4PN');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'CierreSension', 'ykOtMTbmYnQI3c5gFoB');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'Empresa', 'y8e2uV8ynRK');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'Eventos', 'y8uw37Dd93E');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'Perfiles', 'yov9RhQSM4Fl');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'Usuarios', 'y8P6oKz09rNX');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'Clientes', 'ykesXpvePbfu');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'Empleados', 'y8e2uttDS2M8hC');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'Grupo_Cliente', 'y0eZqEQcalRKOEpfFcXC');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'MapeoClientes', 'yceIcuoYDCgynAMfIBO');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'ModificarCliente', 'yc+LS1tu734VEWMibRmmv7P');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'OrdenVisita', 'yUu1Of5vKJXq9JeN');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'Producto', 'yof4bBb0x0QS');

```

```

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'Proveedores', 'yof4bJqPy8yMy4uW');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'CalificarCliente', 'ykOvxFlfdwhb7/eTL7sLu9N');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'Clientesencumpleanos',
'ykesXpvePbfuOEPA3e9gKvJgbrXY');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'ImpresiondeFichas', 'yMuaQ0m2gE3i+Cs+1BU0Q9E5');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'RegistroLlamadas', 'yg/k9BNBuVbvnkLISwFjleO');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'Clienteactivoporgrupo',
'ykesXpvePb/qC9A8Sn3RvizRvokOKC');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'ClientesInactivoGrupo',
'ykesXpvePbfuiEXxTUg2XaEhb9Y+rB');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'ExtractoCliente', 'y8OzW7M9kKTGDEMvWc/kkB');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'ListadeVentas', 'yY+GI880jl7BZ+DL0A');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'PlanillaCobranza', 'yo//Fo/5dB0snaSYLEaBkcK');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'PlanillaInactivo', 'yo//Fo/5dB0st60gTg1lZtO');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'PlanillaMoroso', 'yo//Fo/5dB0spmofklQc');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'PlanillaparaReVentas',
'yo//Fo/5dB0sUa5rK0kKBB0LsqyX');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'Cobranza', 'ykusvF+waTEi');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'CobranzaMovil', 'ykusvF+waTEiqqMcQJO');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'Compras', 'ykusgtQdNxD');

INSERT INTO UCTABRIGHTS

```



```

VALUES (2, 'Comercial', 'ControldePlanilla', 'ykusjJJwl6G5X4I+6haYdMoa');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'Controldestock', 'ykusjJJwl6G5X0652V8N');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'DevolucionVentas', 'y4OgJLUOACGjOqSOATqoEZD');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'ListadodePedidos', 'yY+GI880jdQh+KcntJBmDBO');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Tarjetas', 'Cerrar_Sesion', 'xg5ldB2PkNHHPx2iL6H');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'VentaMayorista', 'ywvSIRGQpfRbz+SvwDoG');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'Ventas', 'ywvSIRGQXD');

INSERT INTO UCTABRIGHTS
VALUES (2, 'Comercial', 'VentasPorWeb', 'ywvSIRGQXPqcjRnqGD');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Punto de Venta', 'Usuarios', 'x4Ykk836A5LK');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Punto de Venta', 'Perfiles', 'xsoErhGLWnpX');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Punto de Venta', 'CambiarContrasena',
'xg5kY6FDiS2DAMTNxoYv4z3E');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Punto de Venta', 'CerrarSesion', 'xg5ldB2Pk9XmBtXTgB');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Finanzas', 'actusuario', 'xorPZCKnmwSqYsL');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Finanzas', 'actperfil', 'xorPZWanUlU3CB');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Finanzas', 'actcambiarcontrasena',
'xorPZaLzgAMiXrl4oLFTleLEnWS9');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Finanzas', 'actcerrar_sesion', 'xorPZaLyhBNaPWoiAk9fmVL');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'Ayuda', 'xop9LSZ8');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'CambioContrasena', 'xg5kY6FDsWWTJMU6OREOkXO');

```

```

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'CierreSension', 'xg5mMkloiotqtCLuKkD');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'Empresa', 'x4Jp5wT1cCM');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'Eventos', 'x45vqRXuIlM');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'Perfiles', 'xsoErhGLWnpx');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'Usuarios', 'x4Ykk836A5LK');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'Clientes', 'xgpn9G9Y3LT3');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'Empleados', 'x4Jp5IC3EGTdED');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'Grupo_Cliente', 'xwpXWX0YD0V46HM3GKH');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'MapeoClientes', 'xY5NhfUfUdRCm0KNcKL');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'ModificarCliente', 'xYZOK1/X1Ow7RWMtPVBi3iL');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'OrdenVisita', 'xQ5yQ31xy26GNGkX');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'Producto', 'xsYBxDCg4Afe');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'Proveedores', 'xsYBxLTHGUqhmmBr');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'CalificarCliente', 'xg5kZOWYn7srRAiwn5TDa3A');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'Clientesencumpleanos',
'xgpn9G9Y3LT3Q8eekYUWCVK7uVUN');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'ImpresiondeFichas', 'xIZcdSy/1YIAE+mxgeLfZXAO');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'RegistroLlamadas', 'xk4vDsnl6ctzeoB2w8odO7P');

INSERT INTO UCTABRIGHTS

```

```

VALUES (1, 'Comercial', 'Clienteactivoporgrupo',
'xgpn9G9Y3LzzIN0tQqthl0PRxMZMgD');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'ClientesInactivoGrupo',
'xgpn9G9Y3LT38IpxcDLSVbWV/ueQ8A');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'ExtractoCliente', 'x4ZsxuJoLsYt5vw3XEvl5D');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'ListadeVentas', 'xcZTaeBInZcv7QUncMI');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'PlanillaCobranza', 'xs4G6cZqU7Mmq/dEV0cIicG');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'PlanillaInactivo', 'xs4G6cZqU7MmgH/jegUFyyD');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'PlanillaMoroso', 'xs4G6cZqU7Mmkb7t1F//');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'PlanillaparaReVentas',
'xs4G6cZqU7MmZv5LtLDyOnZkW6db');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'Cobranza', 'xgZnkRywXGaW');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'CobranzaMovil', 'xgZnkRywXGaWslUxwaP');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'Compras', 'xgZnrtomSrO');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'ControldePlanilla', 'xgZnot/GN5rn7STKnOfNPPoe');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'Controldestock', 'xgZnot/GN5rn7exlUCqw');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'DevolucionVentas', 'x8p8/uMv4MdyHv9Xiuk+lpB');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'ListadodePedidos', 'xcZTaeBInxspVME0is+fbMJ');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'VentaMayorista', 'x0YcH2RVlGjgi9GxxoGY');

INSERT INTO UCTABRIGHTS
VALUES (1, 'Comercial', 'Ventas', 'x0YcH2RVbC');

INSERT INTO UCTABRIGHTS

```

```
VALUES (1, 'Comercial', 'VentasPorWeb', 'x0Ych2RVb+Wnre10ED');
```

```
COMMIT WORK;
```

```
/* Data for the 'UCTABUSERSLOGGED' table (Records 1 - 500) */
```

```
INSERT INTO UCTABUSERSLOGGED
```

```
VALUES ('{3F3FC141-67FB-43B7-B951-149072BE6AE0}', 1, 'Comercial', 'P5Q-SE', '12/08/17 17:26');
```

```
/* Data for the 'UCTABUSERSLOGGED' table (Records 501 - 518) */
```

```
INSERT INTO UCTABUSERSLOGGED
```

```
VALUES ('{0BC6572A-D4B6-4C47-908B-62E7389B0D01}', 2, 'Comercial', 'HP-PC', '01/08/23 08:09');
```

```
COMMIT WORK;
```

```
/* Structure for the `ALMACEN_MOVI_D` table : */
```

```
ALTER TABLE ALMACEN_MOVI_D ADD CONSTRAINT FK_ID_ALMACEN_MO FOREIGN KEY (ID_ALMACEN_MO) REFERENCES ALMACEN_MOVI(ID_ALMACEN_MO) ON UPDATE CASCADE;
```

```
/* Structure for the `CIUDAD` table : */
```

```
ALTER TABLE CIUDAD ADD CONSTRAINT FK_ID_PAIS FOREIGN KEY (ID_PAIS) REFERENCES PAIS(ID_PAIS) ON UPDATE CASCADE;
```

```
/* Structure for the `CLIENTE` table : */
```

```
ALTER TABLE CLIENTE ADD CONSTRAINT FK_ID_CIUADAD FOREIGN KEY (ID_CIUADAD) REFERENCES CIUDAD(ID_CIUADAD) ON UPDATE CASCADE;
```

```
ALTER TABLE CLIENTE ADD CONSTRAINT FK_ID_GRUPO_CLIENTE FOREIGN KEY (ID_GRUPO_CLIENTE) REFERENCES GRUPO_CLIENTE(ID_GRUPO_CLIENTE) ON UPDATE CASCADE;
```

```
/* Structure for the `CLIENTE_NOTA_CREDITO_D` table : */
```

```
ALTER TABLE CLIENTE_NOTA_CREDITO_D ADD CONSTRAINT FK_ID_CLIENTE_NOTA_CREDITO FOREIGN KEY (ID_CLIENTE_NOTA_CREDITO) REFERENCES CLIENTE_NOTA_CREDITO(ID_CLIENTE_NOTA_CREDITO) ON DELETE CASCADE ON UPDATE CASCADE;
```

```
/* Structure for the `COBRANZAMOVIL` table : */
```

```
ALTER TABLE COBRANZAMOVIL ADD CONSTRAINT COBRANZAMOCONVENDEDOR FOREIGN KEY (ID_VENDEDOR) REFERENCES VENDEDOR(ID_VENDEDOR) ON UPDATE CASCADE;
```

/\* Structure for the `COBRANZAMOVILD` table : \*/

```
ALTER TABLE COBRANZAMOVILD ADD CONSTRAINT COBRANZAMOVILCONCLIENTE FOREIGN  
KEY (ID_CLIENTE) REFERENCES CLIENTE(AI_CLIENTE) ON UPDATE CASCADE;
```

```
ALTER TABLE COBRANZAMOVILD ADD CONSTRAINT FK_ID_COBRANZAMOVIL FOREIGN KEY  
(ID_COBRANZAMOVIL) REFERENCES COBRANZAMOVIL(ID_COBRANZAMOVIL) ON DELETE  
CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE COBRANZAMOVILD ADD CONSTRAINT FK_ID_COBRANZAMOVILOBSERVACION  
FOREIGN KEY (ID_COBRANZAMOVILOBSERVACION) REFERENCES  
COBRANZAMOVILOBSERVACION(ID_COBRANZAMOVILOBSERVACION) ON UPDATE CASCADE;
```

/\* Structure for the `CUENTA\_CLIENTE` table : \*/

```
ALTER TABLE CUENTA_CLIENTE ADD CONSTRAINT FK_ID_CLIENTE FOREIGN KEY  
(ID_CLIENTE) REFERENCES CLIENTE(AI_CLIENTE) ON UPDATE CASCADE;
```

```
ALTER TABLE CUENTA_CLIENTE ADD CONSTRAINT FK_ID_COBRANZA_DETALLE FOREIGN  
KEY (ID_COBRANZA_DETALLE) REFERENCES  
RECIBO_CLIENTE_DETALLE(ID_COBRANZA_DETALLE) ON DELETE CASCADE ON UPDATE  
CASCADE;
```

/\* Structure for the `RECIBO\_CLIENTE` table : \*/

```
ALTER TABLE RECIBO_CLIENTE ADD CONSTRAINT FK_ID_VENDEDOR FOREIGN KEY  
(ID_VENDEDOR) REFERENCES VENDEDOR(ID_VENDEDOR) ON UPDATE CASCADE;
```

/\* Structure for the `RECIBO\_CLIENTE\_DETALLE` table : \*/

```
ALTER TABLE RECIBO_CLIENTE_DETALLE ADD CONSTRAINT FK_ID_COBRANZA FOREIGN  
KEY (ID_COBRANZA) REFERENCES RECIBO_CLIENTE(ID_COBRANZA) ON UPDATE  
CASCADE;
```

/\* Structure for the `VENTA` table : \*/

```
ALTER TABLE VENTA ADD CONSTRAINT VENTASCONCLIENTES FOREIGN KEY  
(ID_CLIENTE) REFERENCES CLIENTE(AI_CLIENTE) ON UPDATE CASCADE;
```

```
ALTER TABLE VENTA ADD CONSTRAINT VENTASCONVENDEDOR FOREIGN KEY  
(ID_VENDEDOR) REFERENCES VENDEDOR(ID_VENDEDOR);
```

/\* Privileges for the `ALMACEN\_MOVI` : \*/

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON ALMACEN_MOVI TO  
SYSDBA WITH GRANT OPTION;
```

/\* Privileges for the `ALMACEN\_MOVI\_D` : \*/

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON ALMACEN_MOVI_D TO  
SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `CALIFICACION` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON CALIFICACION TO  
SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `CIUDAD` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON CIUDAD TO ERP WITH  
GRANT OPTION;
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON CIUDAD TO SYSDBA;
```

```
/* Privileges for the `CLIENTE` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON CLIENTE TO ERP WITH  
GRANT OPTION;
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON CLIENTE TO SYSDBA;
```

```
/* Privileges for the `CLIENTE_LLAMADA` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON CLIENTE_LLAMADA TO  
SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `CLIENTE_NOTA_CREDITO` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON CLIENTE_NOTA_CREDITO  
TO SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `CLIENTE_NOTA_CREDITO_D` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON  
CLIENTE_NOTA_CREDITO_D TO SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `CLIENTE_SEGUI` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON CLIENTE_SEGUI TO  
SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `COBRANZAMOVIL` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON COBRANZAMOVIL TO  
SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `COBRANZAMOVILD` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON COBRANZAMOVILD TO  
SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `COBRANZAMOVILOBSERVACION` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON  
COBRANZAMOVILOBSERVACION TO SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `COMPRA` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON COMPRA TO SYSDBA WITH  
GRANT OPTION;
```

```
/* Privileges for the `COMPRAD` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON COMPRAD TO SYSDBA  
WITH GRANT OPTION;
```

```
/* Privileges for the `CUENTA_CLIENTE` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON CUENTA_CLIENTE TO ERP  
WITH GRANT OPTION;
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON CUENTA_CLIENTE TO  
SYSDBA;
```

```
/* Privileges for the `CUENTA_CLIENTE_1` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON CUENTA_CLIENTE_1 TO  
SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `EMAIL_PENDIENTES` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON EMAIL_PENDIENTES TO  
SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `EMPRESA` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON EMPRESA TO ERP WITH  
GRANT OPTION;
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON EMPRESA TO SYSDBA;
```

```
/* Privileges for the `EMPRESA_CLAVE` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON EMPRESA_CLAVE TO ERP  
WITH GRANT OPTION;
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON EMPRESA_CLAVE TO  
SYSDBA;
```

```
/* Privileges for the `EVENTO` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON EVENTO TO SYSDBA WITH  
GRANT OPTION;
```

```
/* Privileges for the `FIB$DATASETS_INFO` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON FIB$DATASETS_INFO TO  
ERP WITH GRANT OPTION;
```

```
GRANT SELECT ON FIB$DATASETS_INFO TO PUBLIC;
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON FIB$DATASETS_INFO TO  
SYSDBA;
```

```
/* Privileges for the `FORMA_PAGO` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON FORMA_PAGO TO SYSDBA  
WITH GRANT OPTION;
```

```
/* Privileges for the `GRUPO_CLIENTE` : */
```



```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON GRUPO_CLIENTE TO ERP  
WITH GRANT OPTION;
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON GRUPO_CLIENTE TO  
SYSDBA;
```

```
/* Privileges for the `METO_PAGO` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON METO_PAGO TO SYSDBA  
WITH GRANT OPTION;
```

```
/* Privileges for the `MOTIVO_LLAMADA` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON MOTIVO_LLAMADA TO  
SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `MOVIMIENTO` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON MOVIMIENTO TO SYSDBA  
WITH GRANT OPTION;
```

```
/* Privileges for the `MOVIMIENTO_DETALLE` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON MOVIMIENTO_DETALLE TO  
SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `PAIS` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON PAIS TO ERP WITH  
GRANT OPTION;
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON PAIS TO SYSDBA;
```

```
/* Privileges for the `PEDIDO` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON PEDIDO TO SYSDBA WITH  
GRANT OPTION;
```

```
/* Privileges for the `PREVENTA` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON PREVENTA TO SYSDBA  
WITH GRANT OPTION;
```

```
/* Privileges for the `PRODUCTO` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON PRODUCTO TO SYSDBA  
WITH GRANT OPTION;
```

```
/* Privileges for the `PRODUCTO_MARCA` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON PRODUCTO_MARCA TO  
SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `PRODUCTO_PRECIO_PROVE` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON PRODUCTO_PRECIO_PROVE  
TO SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `PRODUCTO_SEXO` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON PRODUCTO_SEXO TO  
SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `PRODUCTO_TIPO` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON PRODUCTO_TIPO TO  
SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `PROVEEDOR` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON PROVEEDOR TO SYSDBA  
WITH GRANT OPTION;
```

```
/* Privileges for the `RECIBO_CLIENTE` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON RECIBO_CLIENTE TO  
SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `RECIBO_CLIENTE_DETALLE` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON  
RECIBO_CLIENTE_DETALLE TO SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `SIS` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON SIS TO ERP WITH GRANT  
OPTION;
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON SIS TO SYSDBA;
```

```
/* Privileges for the `TIPO_PRECIO` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON TIPO_PRECIO TO SYSDBA  
WITH GRANT OPTION;
```

```
/* Privileges for the `UCTABPERFIL` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON UCTABPERFIL TO SYSDBA  
WITH GRANT OPTION;
```

```
/* Privileges for the `UCTABRIGHTS` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON UCTABRIGHTS TO SYSDBA  
WITH GRANT OPTION;
```

```
/* Privileges for the `UCTABRIGHTSEX` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON UCTABRIGHTSEX TO  
SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `UCTABUSERS` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON UCTABUSERS TO SYSDBA  
WITH GRANT OPTION;
```

```
/* Privileges for the `UCTABUSERSLOGGED` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON UCTABUSERSLOGGED TO  
SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `VENDEDOR` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON VENDEDOR TO SYSDBA  
WITH GRANT OPTION;
```

```
/* Privileges for the `VENTA` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON VENTA TO SYSDBA WITH  
GRANT OPTION;
```

```
/* Privileges for the `VENTA_FICHAS` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON VENTA_FICHAS TO  
SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `VENTAAUX` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON VENTAAUX TO SYSDBA  
WITH GRANT OPTION;
```

```
/* Privileges for the `VENTAD` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON VENTAD TO SYSDBA WITH  
GRANT OPTION;
```

```
/* Privileges for the `VENTAMADE` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON VENTAMADE TO SYSDBA  
WITH GRANT OPTION;
```

```
/* Privileges for the `VENTAMAYO` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON VENTAMAYO TO SYSDBA  
WITH GRANT OPTION;
```

```
/* Privileges for the `LISTACLIENTE` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON LISTACLIENTE TO  
SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `ANDROIDBUSCACLIENTE` : */
```

```
GRANT EXECUTE ON PROCEDURE ANDROIDBUSCACLIENTE TO SYSDBA;
```

```
/* Privileges for the `BUSCARCLIENTE` : */
```

```
GRANT EXECUTE ON PROCEDURE BUSCARCLIENTE TO SYSDBA;
```

```
/* Privileges for the `CLIENTE_ANALISIS_CUENTA` : */
```

```
GRANT EXECUTE ON PROCEDURE CLIENTE_ANALISIS_CUENTA TO SYSDBA;
```

```
/* Privileges for the `CLIENTECALIFICACION` : */
```

```
GRANT EXECUTE ON PROCEDURE CLIENTECALIFICACION TO SYSDBA;
```

```
/* Privileges for the `COBRANZA_LISTADO` : */
```

```
GRANT EXECUTE ON PROCEDURE COBRANZA_LISTADO TO SYSDBA;
```

```
/* Privileges for the `COBRANZA_LISTADO_WEB` : */
```

```
GRANT EXECUTE ON PROCEDURE COBRANZA_LISTADO_WEB TO SYSDBA;
```

```
/* Privileges for the `COBRANZAMOVIL_LISTADO` : */
```

```
GRANT EXECUTE ON PROCEDURE COBRANZAMOVIL_LISTADO TO SYSDBA;
```

```
/* Privileges for the `COBRANZAPORLINEAL` : */
```

```
GRANT EXECUTE ON PROCEDURE COBRANZAPORLINEAL TO SYSDBA;
```

```
/* Privileges for the `COBRANZAPORLINEALCABECERA` : */
```

GRANT EXECUTE ON PROCEDURE COBRANZAPORLINEALCABECERA TO SYSDBA;

/\* Privileges for the `COBRANZAPORLINEALDETALLE` : \*/

GRANT EXECUTE ON PROCEDURE COBRANZAPORLINEALDETALLE TO SYSDBA;

/\* Privileges for the `COBRODETALLEACTIVOS` : \*/

GRANT EXECUTE ON PROCEDURE COBRODETALLEACTIVOS TO SYSDBA;

/\* Privileges for the `COBROSRESUMEN` : \*/

GRANT EXECUTE ON PROCEDURE COBROSRESUMEN TO SYSDBA;

/\* Privileges for the `COBROSRESUMENCONITECLIACOB` : \*/

GRANT EXECUTE ON PROCEDURE COBROSRESUMENCONITECLIACOB TO SYSDBA;

/\* Privileges for the `COBROSRESUMENCONITECLIACOB1` : \*/

GRANT EXECUTE ON PROCEDURE COBROSRESUMENCONITECLIACOB1 TO SYSDBA;

/\* Privileges for the `COBROSRESUMENCONITECLIACOB2` : \*/

GRANT EXECUTE ON PROCEDURE COBROSRESUMENCONITECLIACOB2 TO SYSDBA;

/\* Privileges for the `COMERCIAL\_LISTADO\_CLIENTE` : \*/

GRANT EXECUTE ON PROCEDURE COMERCIAL\_LISTADO\_CLIENTE TO SYSDBA;

/\* Privileges for the `COMERCIAL\_MAXIMOS\_PARA\_FICHA` : \*/

GRANT EXECUTE ON PROCEDURE COMERCIAL\_MAXIMOS\_PARA\_FICHA TO SYSDBA;

/\* Privileges for the `COMERCIAL\_ULTIMO\_PAGO\_CLIENTE` : \*/

```
GRANT EXECUTE ON PROCEDURE COMERCIAL_ULTIMO_PAGO_CLIENTE TO SYSDBA;
```

```
/* Privileges for the `COMPROBARSTOCKPARADESCONTAR` : */
```

```
GRANT EXECUTE ON PROCEDURE COMPROBARSTOCKPARADESCONTAR TO SYSDBA;
```

```
/* Privileges for the `CONTROLVENTACANTISTOCKDIS` : */
```

```
GRANT EXECUTE ON PROCEDURE CONTROLVENTACANTISTOCKDIS TO SYSDBA;
```

```
/* Privileges for the `CORRECCIONERROR1` : */
```

```
GRANT EXECUTE ON PROCEDURE CORRECCIONERROR1 TO SYSDBA;
```

```
/* Privileges for the `CUENTA_A_VENCER_DE_CLIENTES_GRU` : */
```

```
GRANT EXECUTE ON PROCEDURE CUENTA_A_VENCER_DE_CLIENTES_GRU TO SYSDBA;
```

```
/* Privileges for the `CUENTA_DE_CLIENTE_POR_GRUPO` : */
```

```
GRANT EXECUTE ON PROCEDURE CUENTA_DE_CLIENTE_POR_GRUPO TO SYSDBA;
```

```
/* Privileges for the `CUENTA_DE_CLINETE_POR_GRUPO_DE` : */
```

```
GRANT EXECUTE ON PROCEDURE CUENTA_DE_CLINETE_POR_GRUPO_DE TO SYSDBA;
```

```
/* Privileges for the `DETALLEDEVENTA` : */
```

```
GRANT EXECUTE ON PROCEDURE DETALLEDEVENTA TO SYSDBA;
```

```
/* Privileges for the `EXTRACTO_CLIENTE` : */
```

```
GRANT EXECUTE ON PROCEDURE EXTRACTO_CLIENTE TO SYSDBA;
```

```
/* Privileges for the `EXTRACTO_CLIENTE_DEUDA` : */
```

```
GRANT EXECUTE ON PROCEDURE EXTRACTO_CLIENTE_DEUDA TO SYSDBA;
```

```
/* Privileges for the `EXTRACTO_CLIENTE_PAGO` : */
```

```
GRANT EXECUTE ON PROCEDURE EXTRACTO_CLIENTE_PAGO TO SYSDBA;
```

```
/* Privileges for the `EXTRACTO_GENERAL` : */
```

```
GRANT EXECUTE ON PROCEDURE EXTRACTO_GENERAL TO SYSDBA;
```

```
/* Privileges for the `IMPRESIONFICHAS` : */
```

```
GRANT EXECUTE ON PROCEDURE IMPRESIONFICHAS TO SYSDBA;
```

```
/* Privileges for the `LISTADEMALETINES` : */
```

```
GRANT EXECUTE ON PROCEDURE LISTADEMALETINES TO SYSDBA;
```

```
/* Privileges for the `LISTADEMOROSOS` : */
```

```
GRANT EXECUTE ON PROCEDURE LISTADEMOROSOS TO SYSDBA;
```

```
/* Privileges for the `LISTADEMOROSOSD` : */
```

```
GRANT EXECUTE ON PROCEDURE LISTADEMOROSOSD TO SYSDBA;
```

```
/* Privileges for the `LISTADO_CUENTA_CLIENTE` : */
```

```
GRANT EXECUTE ON PROCEDURE LISTADO_CUENTA_CLIENTE TO SYSDBA;
```

```
/* Privileges for the `LISTADODECUMPLEANOS` : */
```

```
GRANT EXECUTE ON PROCEDURE LISTADODECUMPLEANOS TO SYSDBA;
```

```
/* Privileges for the `LISTADODEVENTA` : */
```



GRANT EXECUTE ON PROCEDURE LISTADODEVENTA TO SYSDBA;

/\* Privileges for the `LISTADODEVOLUCIONES` : \*/

GRANT EXECUTE ON PROCEDURE LISTADODEVOLUCIONES TO SYSDBA;

/\* Privileges for the `LISTADOMOVIMIENTOALMACEN` : \*/

GRANT EXECUTE ON PROCEDURE LISTADOMOVIMIENTOALMACEN TO SYSDBA;

/\* Privileges for the `LISTADOPARACOBranza` : \*/

GRANT EXECUTE ON PROCEDURE LISTADOPARACOBranza TO SYSDBA;

/\* Privileges for the `LISTADOPORGRUPOACTIVOINACTIVO` : \*/

GRANT EXECUTE ON PROCEDURE LISTADOPORGRUPOACTIVOINACTIVO TO SYSDBA;

/\* Privileges for the `LISTADOSTOCK` : \*/

GRANT EXECUTE ON PROCEDURE LISTADOSTOCK TO SYSDBA;

/\* Privileges for the `PLANILLAPARAREVENTA` : \*/

GRANT EXECUTE ON PROCEDURE PLANILLAPARAREVENTA TO SYSDBA;

/\* Privileges for the `PLANILLAPARAREVENTAINACTIVO` : \*/

GRANT EXECUTE ON PROCEDURE PLANILLAPARAREVENTAINACTIVO TO SYSDBA;

/\* Privileges for the `PROCOBRANZAMOVIL` : \*/

GRANT EXECUTE ON PROCEDURE PROCOBRANZAMOVIL TO SYSDBA;

/\* Privileges for the `PROCOBRANZAMOVILSIN` : \*/

```
GRANT EXECUTE ON PROCEDURE PROCOBRANZAMOVILSIN TO SYSDBA;
```

```
/* Privileges for the `RECIBO_GENERAR_MENSA` : */
```

```
GRANT EXECUTE ON PROCEDURE RECIBO_GENERAR_MENSA TO SYSDBA;
```

```
/* Privileges for the `RECIBOANALISIS` : */
```

```
GRANT EXECUTE ON PROCEDURE RECIBOANALISIS TO SYSDBA;
```

```
/* Privileges for the `RECIBOANALISIS1` : */
```

```
GRANT EXECUTE ON PROCEDURE RECIBOANALISIS1 TO SYSDBA;
```

```
/* Privileges for the `RECIBOANALISISCUE` : */
```

```
GRANT EXECUTE ON PROCEDURE RECIBOANALISISCUE TO SYSDBA;
```

```
/* Privileges for the `REDONDEAR` : */
```

```
GRANT EXECUTE ON PROCEDURE REDONDEAR TO SYSDBA;
```

```
/* Privileges for the `RESUMEN_CUENTA_POR_VENTA` : */
```

```
GRANT EXECUTE ON PROCEDURE RESUMEN_CUENTA_POR_VENTA TO SYSDBA;
```

```
/* Privileges for the `RESUMENDEUNAVENTA` : */
```

```
GRANT EXECUTE ON PROCEDURE RESUMENDEUNAVENTA TO SYSDBA;
```

```
/* Privileges for the `SPREGISTROS` : */
```

```
GRANT EXECUTE ON PROCEDURE SPREGISTROS TO SYSDBA;
```

```
/* Privileges for the `STOCKACTUALDEUNPRODUCTO` : */
```

```
GRANT EXECUTE ON PROCEDURE STOCKACTUALDEUNPRODUCTO TO SYSDBA;
```

```
/* Privileges for the `updatecliente` : */
```

```
GRANT EXECUTE ON PROCEDURE "updatecliente" TO SYSDBA;
```

```
/* Privileges for the `UPDATECLIENTEVENDEDOR` : */
```

```
GRANT EXECUTE ON PROCEDURE UPDATECLIENTEVENDEDOR TO SYSDBA;
```

```
/* SQL Manager for InterBase and Firebird 5.5.3.52046 */
/* ----- */
/* Host      : localhost */
/* Database  : D:\Gabriela Fragancias\Base de datos\EMPRESA.FDB */
```

```
CREATE DATABASE 'D:\Gabriela Fragancias\Base de datos\EMPRESA.FDB'
  USER 'SYSDBA'
  PAGE_SIZE = 1024
  DEFAULT CHARACTER SET NONE;
```

```
/* Definition for the `FIB$BOOLEAN` domain : */
```

```
CREATE DOMAIN FIB$BOOLEAN AS
  SMALLINT
  DEFAULT 1
  NOT NULL
  CHECK (VALUE IN (0,1));
```

```
/* Structure for the `FIB$FIELDS_INFO` table : */
```

```
CREATE TABLE FIB$FIELDS_INFO (
  TABLE_NAME VARCHAR(31) NOT NULL,
  FIELD_NAME VARCHAR(31) NOT NULL,
  DISPLAY_LABEL VARCHAR(25),
  VISIBLE FIB$BOOLEAN DEFAULT 1 NOT NULL,
  DISPLAY_FORMAT VARCHAR(15),
  EDIT_FORMAT VARCHAR(15),
  TRIGGERED FIB$BOOLEAN DEFAULT 0 NOT NULL,
  DISPLAY_WIDTH INTEGER DEFAULT 0,
  FIB$VERSION INTEGER);
```

```
ALTER TABLE FIB$FIELDS_INFO ADD CONSTRAINT PK_FIB$FIELDS_INFO PRIMARY KEY
(TABLE_NAME, FIELD_NAME);
```

```
/* Definition for the `FIB$FIELD_INFO_VERSION` generator : */
```

```
CREATE GENERATOR FIB$FIELD_INFO_VERSION;
```

```
/* Definition for the `FIB$FIELDS_INFO_BI` trigger : */
```

```
SET TERM ^ ;
```

```
CREATE TRIGGER FIB$FIELDS_INFO_BI FOR FIB$FIELDS_INFO  
ACTIVE BEFORE
```

```
    INSERT  
POSITION 0  
AS
```

```
begin  
new.fib$version=gen_id(fib$field_info_version,1);  
end^
```

```
SET TERM ; ^
```

```
/* Definition for the `FIB$FIELDS_INFO_BU` trigger : */
```

```
SET TERM ^ ;
```

```
CREATE TRIGGER FIB$FIELDS_INFO_BU FOR FIB$FIELDS_INFO  
ACTIVE BEFORE
```

```
    UPDATE  
POSITION 0  
AS
```

```
begin  
new.fib$version=gen_id(fib$field_info_version,1);  
end^
```

```
SET TERM ; ^
```

```
/* Definition for the `DIRECCION` domain : */
```

```
CREATE DOMAIN DIRECCION AS  
    VARCHAR(500);
```

```
/* Definition for the `ID` domain : */
```

```
CREATE DOMAIN ID AS  
    SMALLINT;
```

```
/* Definition for the `NOMBRE` domain : */
```

```
CREATE DOMAIN NOMBRE AS  
    VARCHAR(50);
```

```
/* Structure for the `EMPRESA_EMAIL` table : */
```

```
CREATE TABLE EMPRESA_EMAIL (  
    ID_EMPRESA ID NOT NULL,  
    RAZON NOMBRE,  
    DIRECCION DIRECCION,  
    EMAIL NOMBRE,  
    NOMBRE_SERVIDOR NOMBRE,
```

```

    PUERTO_SERVIDOR ID,
    CONTRASE_EMAIL NOMBRE);

/* Structure for the `GRUPO_ENVIO` table : */

CREATE TABLE GRUPO_ENVIO (
    ID_GRUPO_ENVIO ID NOT NULL,
    DESCRIPCION DIRECCION);

ALTER TABLE GRUPO_ENVIO ADD CONSTRAINT PK_GRUPO_ENVIO PRIMARY KEY
(ID_GRUPO_ENVIO);

/* Definition for the `ENTERO` domain : */

CREATE DOMAIN ENTERO AS
    INTEGER;

/* Definition for the `NOMBRE_LARGO` domain : */

CREATE DOMAIN NOMBRE_LARGO AS
    VARCHAR(100);

/* Structure for the `CONFI_MENSAJE` table : */

CREATE TABLE CONFI_MENSAJE (
    ID_CONFIGURACION ENTERO NOT NULL,
    ID_CALIFICACION ENTERO,
    DESCRIPCION NOMBRE,
    UBICACION_SONIDO NOMBRE_LARGO);

ALTER TABLE CONFI_MENSAJE ADD CONSTRAINT PK_CONFI_MENSAJE PRIMARY KEY
(ID_CONFIGURACION);

/* Structure for the `GRUPO_ENVIO_D` table : */

CREATE TABLE GRUPO_ENVIO_D (
    ID_GRUPO_ENVIO_D ID NOT NULL,
    ID_GRUPO_ENVIO ID,
    ID_CLIENTE ENTERO);

ALTER TABLE GRUPO_ENVIO_D ADD CONSTRAINT PK_GRUPO_ENVIO_D PRIMARY KEY
(ID_GRUPO_ENVIO_D);

/* Definition for the `HORA` domain : */

CREATE DOMAIN HORA AS
    TIME;

/* Definition for the `NOMBRE_CORTO` domain : */

CREATE DOMAIN NOMBRE_CORTO AS

```

```

    VARCHAR(50);

/* Structure for the `COPIA_SEGURIDAD` table : */

CREATE TABLE COPIA_SEGURIDAD (
    ID_COPIA ID NOT NULL,
    SERVIDOR NOMBRE_CORTO,
    NOMBRE_BD NOMBRE_LARGO,
    BACK_DESTINO NOMBRE_LARGO,
    HORA_BACK HORA);

ALTER TABLE COPIA_SEGURIDAD ADD CONSTRAINT PK_COPIA_SEGURIDAD PRIMARY KEY
(ID_COPIA);

/* Structure for the `CONFIGURACION` table : */

CREATE TABLE CONFIGURACION (
    INI_DIRECCION VARCHAR(500));

/* Structure for the `EMPRESA` table : */

CREATE TABLE EMPRESA (
    NOMBRE VARCHAR(50),
    UBICACION VARCHAR(500),
    IP CHAR(25));

/* Structure for the `REPMAN_GROUPS` table : */

CREATE TABLE REPMAN_GROUPS (
    GROUP_CODE INTEGER NOT NULL,
    GROUP_NAME VARCHAR(50),
    PARENT_GROUP INTEGER NOT NULL);

ALTER TABLE REPMAN_GROUPS ADD PRIMARY KEY (GROUP_CODE) USING INDEX
RDB$PRIMARY2;

/* Structure for the `REPMAN_REPORTS` table : */

CREATE TABLE REPMAN_REPORTS (
    REPORT_NAME VARCHAR(50) NOT NULL,
    REPORT_BLOB,
    REPORT_GROUP INTEGER,
    USER_FLAG INTEGER);

ALTER TABLE REPMAN_REPORTS ADD PRIMARY KEY (REPORT_NAME) USING INDEX
RDB$PRIMARY1;

/* Definition for the `CEILING` UDF : */

DECLARE EXTERNAL FUNCTION "CEILING"
    DOUBLE PRECISION

```

```

RETURNS DOUBLE PRECISION BY VALUE
ENTRY_POINT 'IB_UDF_ceiling' MODULE_NAME 'ib_udf';

/* Definition for the `FLOOR` UDF : */

DECLARE EXTERNAL FUNCTION "FLOOR"
    DOUBLE PRECISION
RETURNS DOUBLE PRECISION BY VALUE
ENTRY_POINT 'IB_UDF_floor' MODULE_NAME 'ib_udf';

/* Data for the 'EMPRESA_EMAIL' table (Records 1 - 1) */

INSERT INTO EMPRESA_EMAIL
VALUES (1, 'Gabriela Fragancias', 'Calle San Miguel 105 / 2',
'gabriela.fragancias@gmail.com', 26, '3581961cabeza');

COMMIT WORK;

/* Data for the 'CONFI_MENSAJE' table (Records 1 - 2) */

INSERT INTO CONFI_MENSAJE
VALUES (1, 1, 'Escriba una descripción', 'D:\\Gabriela
Fragancias\\Recibir Email\\Sonidos\\Primero.wav');

INSERT INTO CONFI_MENSAJE
VALUES (2, 4, 'Escriba una descripción', 'D:\\Gabriela
Fragancias\\Recibir Email\\Sonidos\\demandar.wav');

COMMIT WORK;

/* Data for the 'COPIA_SEGURIDAD' table (Records 1 - 1) */

INSERT INTO COPIA_SEGURIDAD
VALUES (2, '127.0.0.1', 'BASE', 'C:\\', '11:30:00');

COMMIT WORK;

/* Data for the 'CONFIGURACION' table (Records 1 - 1) */

INSERT INTO CONFIGURACION
VALUES ('D:\\Gabriela Fragancias\\Base de Datos\\Empresa\\');

COMMIT WORK;

/* Data for the 'EMPRESA' table (Records 1 - 8) */

INSERT INTO EMPRESA
VALUES ('Original', 'BASE', '127.0.0.1');

INSERT INTO EMPRESA
VALUES ('Gabriela Fragancias', 'D:\\Gabriela Fragancias\\Base de
Datos\\BASE.FDB', '127.0.0.1');

COMMIT WORK;

```

```

/* Data for the 'REPMAN_REPORTS' table (Records 1 - 1) */

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('FACTURA LEGAL', 0, NULL);

COMMIT WORK;

/* Structure for the `GRUPO_ENVIO_D` table : */

ALTER TABLE GRUPO_ENVIO_D ADD CONSTRAINT FK_ID_GRUPO_ENVIO FOREIGN KEY
(ID_GRUPO_ENVIO) REFERENCES GRUPO_ENVIO(ID_GRUPO_ENVIO) ON DELETE CASCADE
ON UPDATE CASCADE;

/* Privileges for the `CONFI_MENSAJE` : */

GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON CONFI_MENSAJE TO
SYSDBA WITH GRANT OPTION;

/* Privileges for the `CONFIGURACION` : */

GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON CONFIGURACION TO
SYSDBA WITH GRANT OPTION;

/* Privileges for the `COPIA_SEGURIDAD` : */

GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON COPIA_SEGURIDAD TO
SYSDBA WITH GRANT OPTION;

/* Privileges for the `EMPRESA` : */

GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON EMPRESA TO SYSDBA
WITH GRANT OPTION;

/* Privileges for the `EMPRESA_EMAIL` : */

GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON EMPRESA_EMAIL TO
SYSDBA WITH GRANT OPTION;

/* Privileges for the `FIB$FIELDS_INFO` : */

GRANT SELECT ON FIB$FIELDS_INFO TO PUBLIC;

```



```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON FIB$FIELDS_INFO TO
SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `GRUPO_ENVIO` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON GRUPO_ENVIO TO SYSDBA
WITH GRANT OPTION;
```

```
/* Privileges for the `GRUPO_ENVIO_D` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON GRUPO_ENVIO_D TO
SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `REPMAN_GROUPS` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON REPMAN_GROUPS TO
SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `REPMAN_REPORTS` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON REPMAN_REPORTS TO
SYSDBA WITH GRANT OPTION;
```

```
/* SQL Manager for InterBase and Firebird 5.5.3.52046          */
/* ----- */
/* Host      : localhost                                     */
/* Database  : D:\Gabriela Fragancias\Base de datos\INFORME.FDB */
```

```
CREATE DATABASE 'D:\Gabriela Fragancias\Base de datos\INFORME.FDB'
  USER 'SYSDBA'
  PAGE_SIZE = 1024
  DEFAULT CHARACTER SET ASCII;
```

```
/* Structure for the `REPMAN_GROUPS` table : */
```

```
CREATE TABLE REPMAN_GROUPS (
  GROUP_CODE INTEGER NOT NULL,
  GROUP_NAME VARCHAR(100),
  PARENT_GROUP INTEGER NOT NULL);
```

```
ALTER TABLE REPMAN_GROUPS ADD PRIMARY KEY (GROUP_CODE) USING INDEX
RDB$PRIMARY2;
```

```
/* Structure for the `REPMAN_REPORTS` table : */
```

```
CREATE TABLE REPMAN_REPORTS (  
    REPORT_NAME VARCHAR(50) NOT NULL,  
    REPORT_BLOB,  
    REPORT_GROUP INTEGER,  
    USER_FLAG INTEGER);
```

```
ALTER TABLE REPMAN_REPORTS ADD PRIMARY KEY (REPORT_NAME) USING INDEX  
RDB$PRIMARY1;
```

```
/* Data for the 'REPMAN_GROUPS' table (Records 1 - 27) */
```

```
INSERT INTO REPMAN_GROUPS  
VALUES (1, 'PRODUCTOS', 0);
```

```
INSERT INTO REPMAN_GROUPS  
VALUES (5, 'RESUMEN PUNTO DE VENTA', 0);
```

```
INSERT INTO REPMAN_GROUPS  
VALUES (6, 'PROVEEDORES', 0);
```

```
INSERT INTO REPMAN_GROUPS  
VALUES (2, 'ALMACEN', 0);
```

```
INSERT INTO REPMAN_GROUPS  
VALUES (4, 'CAJA', 0);
```

```
INSERT INTO REPMAN_GROUPS  
VALUES (7, 'CLIENTES', 0);
```

```
INSERT INTO REPMAN_GROUPS  
VALUES (3, 'PRECIO', 0);
```

```
INSERT INTO REPMAN_GROUPS  
VALUES (8, 'VENTAS', 0);
```

```
INSERT INTO REPMAN_GROUPS  
VALUES (9, 'CONTABILIDAD', 0);
```

```
INSERT INTO REPMAN_GROUPS  
VALUES (10, 'LIBRO MAYOR', 9);
```

```
INSERT INTO REPMAN_GROUPS  
VALUES (11, 'DIARIO', 9);
```

```
INSERT INTO REPMAN_GROUPS  
VALUES (12, 'PLAN DE CUENTA', 9);
```

```
INSERT INTO REPMAN_GROUPS  
VALUES (13, 'GANANCIA', 0);
```

```
INSERT INTO REPMAN_GROUPS  
VALUES (14, 'BALANCE', 9);
```

```
INSERT INTO REPMAN_GROUPS
VALUES (15, 'COMPRAS', 0);
```

```
INSERT INTO REPMAN_GROUPS
VALUES (16, 'LIBRO COMPRA Y VENTA', 0);
```

```
INSERT INTO REPMAN_GROUPS
VALUES (17, 'COMPARATIVO', 14);
```

```
INSERT INTO REPMAN_GROUPS
VALUES (18, 'VENDEDOR', 0);
```

```
INSERT INTO REPMAN_GROUPS
VALUES (19, 'FLUJO EFECTIVO', 9);
```

```
INSERT INTO REPMAN_GROUPS
VALUES (20, 'LIBRO FISCAL', 9);
```

```
INSERT INTO REPMAN_GROUPS
VALUES (21, 'ANÁLISIS', 0);
```

```
INSERT INTO REPMAN_GROUPS
VALUES (22, 'USUARIO', 0);
```

```
INSERT INTO REPMAN_GROUPS
VALUES (23, 'ERP_SOLUCION', 0);
```

```
INSERT INTO REPMAN_GROUPS
VALUES (24, 'Finanzas', 0);
```

```
INSERT INTO REPMAN_GROUPS
VALUES (25, 'PROMOCION', 0);
```

```
INSERT INTO REPMAN_GROUPS
VALUES (26, 'MODELO DE FACTURAS', 0);
```

```
INSERT INTO REPMAN_GROUPS
VALUES (27, 'ACADEMICO', 0);
```

```
COMMIT WORK;
```

```
/* Data for the 'REPMAN_REPORTS' table (Records 1 - 251) */
```

```
INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO_PRODUCTO', 1, NULL);
```

```
INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('CAJA DE UN USUARIO', 4, NULL);
```

```
INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('INFORMACIÓN DE UN ARTICULO', 1, NULL);
```

```
INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
```

```

VALUES ('MOVIMIENTO EN ALMACEN', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('MOVIMIENTO POR CONCEPTO EN UN ALMACEN', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO UN ALMACEN', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN DE VENTA DE UN USUARIO POR FECHA', 5, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN DE VENTA', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE VENTA POR FECHAS', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('VENTAS', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN DE CUENTA DE UN CURSO POR CONCEPTO', 27, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE CLIENTES', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('VENTA POR GRUPO CLIENTE', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('ALMACEN LISTADO STOCK CON MULTI UNIDADES', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('CUENTAS A VENCER CLIENTE POR GRUPO 1', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Análisis de venta de productos', 21, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('PRODUCTO CON CUOTA TIPO PRESUPUESTO', 1, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('MOVIMIENTOS A CONFIRMAR POR SEMANA', 24, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Detalle operaciones por estado', 24, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('CLIENTES PAGOS REALIZADOS', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Análisis de compra de clientes previo', 21, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)

```

```

VALUES ('EXISTENCIA POR GRUPO DE UN ALMACEN SIN 0', 1, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN DE VENTAS POR FECHAS', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('CODIGO DE BARRAS', 1, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO STOCK MINIMO', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('CLIENTE FIEL', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Listado de Proveedor Desde Hasta', 6, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN DE VENTAS DEL DIA', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Análisis de venta en almacenes', 21, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN DE CAJA DE UN USUARIO POR FECHA RESUMEN', 5, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN VENTA DE UN ARTICULO', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Libro Mayor', 10, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('IMPRESION DE PAGARE', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('UN MOVIMIENTO CAJA GENERAL', 4, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO POR CLASE PARA INVENTARIO', 1, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Libro diario', 11, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN DE GANANCIAS POR FECHA', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE VENTA DE CLIENTE POR UN COMPROBANTE', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('VENTAS DETALLADA POR ARTICULO', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)

```

```

VALUES ('VENTAS RESUMEN DE UN USUARIO', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('GANANCIA DE UN ALMACEN', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('CUENTAS A VENCER DE UN CLIENTE', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('FICHA DE NUEVO CLIENTE', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('MOVIMIENTO EN CAJA GENERAL POR FECHA', 4, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN DE CUENTA CLIENTE', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN DE COMPRAS POR COMPROBANTE', 15, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LIBRO VENTA POR ALMACEN', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('GANANCIAS DETALLADO POR FECHAS', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO_PRODUCTO_COMPATIBILIDAD', 1, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Libro Mayor continuo', 10, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('PRODUCTO MAS VENDIDO POR CLASE', 1, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('UN MOVIMIENTO DE CUENTA', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE VENTA POR COMPROBANTE', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE RECIPIENTE', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO_PRODUCTO_GRUPO', 1, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Diario continuo', 11, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('VENTA CON VUELTO', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)

```

```

VALUES ('RESUMEN DE LAS CUENTAS', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RECETAS DE PRODUCTO', 1, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('GANANCIA POR FACTURA POR FECHA', 13, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Librio Diario por id', 20, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('GANANCIA DEL DIA RESUMEN', 13, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO STOCK + UN PRECIO', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('PRODUCTO A VENCER', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('VENTAS POR FECHA DE UN PRODUCTO', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN DE VENTA DE UN USUARIO POR FECHA RESUMEN', 5, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('TRANSFERENCIA', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('CIERRE CAJA', 4, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Suma y saldo', 14, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('PAGARE', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE CLIENTE POR PRECIO', 3, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN DE CUENTA DETALLADA PROVEEDOR', 6, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('PRODUCTO A VENCER ENTRE FECHA EN ALMACENES', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE VENTA RESUMEN', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('CUENTAS A VENCER DE UN PROVEEDOR', 6, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)

```

```

VALUES ('PRODUCTO VENCIDO EN ALMACENES', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE STOCK ALMACEN POR SESION', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE STOCK + INFORMACION', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('ALMACEN LISTADO STOCK CON MULTI UNIDADES C', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO POR TOTAL PRECIO VENTA', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO PRODUCTO POR UN GRUPO', 1, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Estado de resultado', 14, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('CODIGO DE BARRA DE PRODUCTOS', 1, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Balance General Comparativo', 17, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('compra por mes', 20, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('VENTAS DE PRODUCTO DE UN USUARIO', 22, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Libro venta por mes', 20, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RECIBO POR PAGO', 27, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO PRODUCTO POR UN MARCA', 1, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('VENTAS POR GRUPO DE CLIENTE CON GANANCIA', 18, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('CUENTA VENCIDA DE LOS CLIENTES', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('VENTAS DE USUARIO POR PRODUCTO', 22, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('DOCUMENTO DE TERCEROS VENCIDOS', 24, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)

```



```

VALUES ('RECIBO DE PAGO POR FACTURA CLIENTE', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Análisis de venta de productos previo', 21, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('CUENTAS A VENCER CLIENTE POR GRUPO', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTA_PRODUCTO_CLIENTE_POR_UN_PRECIO', 3, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Análisis de almacen resumen', 21, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN DE CUENTAS ABIERTOS', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('PRODUCTO MAS VENDIDO EN DEPOSITOS', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN DE GANANCIAS DEL DIA', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('UN MOVIMIENTO DE CUENTA PROVEEDOR', 6, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('FACTURA_MATRICIAL', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('VENTAS DE USUARIO POR FACTURA', 22, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('VENTAS DE USUARIO POR PRODUCTO AL', 22, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Plan de cuenta', 12, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN DE CUENTA DETALLADA CLIENTE', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('ALMACEN LISTADO STOCK CON MULTI UNIDADES M', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('ANALISIS PROVEEDOR EN COMPRA Y CUENTA', 21, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN DE LAS CUENTAS PROVEEDOR', 6, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Análisis de venta en almacenes 1', 21, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)

```

```

VALUES ('RESUMEN DE COMPRAS POR FECHAS', 15, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('PRECIOS DE PRODUCTOS', 23, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('CLIENTES CON PUNTOS ACUMULADOS', 25, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('CUENTA VENCIDA DE LOS PROVEEDORES', 6, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Estado de Resultado', 17, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN DE COBROS DE CLIENTES', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('GANANCIAS DETALLADO DEL DÍA', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Flujo Efectivo', 19, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('FACTURA DE MAC MOVIDA MATRICIAL', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN DE VENTAS POR COMPROBANTE', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE PRODUCTO POR PRECIO', 3, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('VENTAS POR ARTICULO DE VENDEDORES', 18, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO_CLIENTE_CON_CUENTA_POR_GRUPO', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Balance General', 14, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN DE COMPRAS POR ALMACEN', 15, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('COBROS REALIZADOS POR FECHA DE UN VENDEDOR', 18, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('INFORMACIÓN DE UN PROVEEDOR', 6, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('GANANCIA POR DIA', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)

```

```

VALUES ('LISTADO DE PUNTOS DE UN CLIENTE', 25, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('de 8 columnas', 14, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('CUENTAS A VENCER DE LOS CLIENTES', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('FACTURA NUEVA VERSION', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE VENTA DE UN VENDEDOR', 18, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN COMPRA DE UN ARTICULO', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Flujo Efectivo Comparativo', 19, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN DE CUENTA PROVEEDOR', 6, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('VENTA CON UTILIDAD', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('GANANCIA DETALLADO DEL DIA', 13, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('PAGOS REALIZADOS DE CLIENTES', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Analisis Venta por fecha resumen', 21, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('FACTURA COMPRA COMPLETA', 15, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('CUENTAS A VENCER DE LOS PROVEEDORES', 6, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('VENTAS DE USUARIO POR FECHA', 22, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('GANANCIAS POR DIA POR FECHA', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Analisis Ventas a clientes por fecha', 21, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('VENTAS POR FECHA DE VENEDORES', 18, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)

```

```

VALUES ('GANANCIA POR ALMACEN RESUMEN POR DIA', 13, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Análisis Ventas por listado de Clientes', 21, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN DE COMPRAS POR PROVEEDOR', 15, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE PRODUCTO POR IMPUESTO', 1, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Análisis cliente en venta y cuenta', 21, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('EPSON GENERICA', 26, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE CLIENTE POR GRUPO', 3, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Análisis Ventas por grupo de cliente', 21, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Ventas por comprobante de un Almacen', 21, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('saldo cuenta', 24, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO POR TIPO DE PRECIO', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('CHE Y DEPÓSITO A CONFIRMAR SIN LIMITE DE FECHA', 24, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('cheques y deposito a confirmar', 24, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE PRODUCTOS POR PRECIOS', 3, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN MOVIMIENTO DE CAJA POR FECHA', 4, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO POR MARCA PARA INVENTARIO', 1, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE PRODUCTO POR MARCA', 1, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('VENTAS DETALLADA POR ARTICULO SUMADO', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)

```

```

VALUES ('RESUMEN DE VENTAS POR ALMACEN COMPROBANTE', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('GANANCIA POR COMPROBANTE POR FECHA', 13, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('CHEQUES POR VENCIMIENTO', 24, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Análisis de almacen resumen grafico', 21, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO COMPRA POR ALMACÉN', 15, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('DOCUMENTO DE TERCEROS EN PODER', 24, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('detalle operacion x concepto', 24, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Detalle de operaciones por conceptos', 24, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('operacion detalle', 24, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Análisis Ventas en Cuenta Cliente', 21, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('DOCUMENTO DE TERCEROS TITULAR EN PODER', 24, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('movimiento x concepto', 24, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('FACTURA A ENVIAR', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO PROVEEDOR CON MOVÍ COMPRO', 6, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('CARNET DE CLIENTES', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE PRODUCTO VENDIDO POR VENDEDOR', 18, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Detalle de una promocion', 25, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LIBRO DE VENTA', 16, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)

```

```

VALUES ('FACTURA RECIBIDA', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE CLIENTES POR NOMBRE', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('GANANCIA POR COMPROBANTE', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Suma y saldo analitico', 14, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('VENTAS RESUMEN DE USUARIOS', 22, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('PRODUCTOS CON STOCK MINIMO EN ALMACENES', 1, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO POR GRUPO', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE PRODUCTO POR TIPO DE PRECIO Y GRUPO', 1, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN DE VENTAS POR CLIENTE', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LIBRO DE COMPRA', 16, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('VENTAS DE USUARIOS POR FECHA', 22, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE STOCK ALMACEN POR CLASE', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN DE VENTAS POR ALMACEN', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO POR CIUDAD DE CLIENTE', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('de 8 columna analitico', 14, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE VENTA POR GRUPO', 18, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('CLIENTES CON PUNTOS SOLO GANADORES', 25, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE PROVEEDOR', 6, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)

```

```

VALUES ('LISTADO DE COMPRAS', 15, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('GANANCIA POR COMPROBANTE POR TOTAL FECHA', 13, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('VENTA POR GRUPO ALMACEN COMPROBANTE DE CLIENTE', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('CLIENTES POR GRUPO PUNTOS ACUMULADOS', 25, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Análisis Ventas por ciudad de cliente', 21, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('PAGOS REALIZADOS DE CLIENTE DE UN GRUPO', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE PROVEEDOR POR CIUDAD', 6, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Ganadores de una promoción', 25, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('VENTA POR PRECIO ALMACEN COMPROBANTE DE CLIENTE', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('CLIENTES POR GRUPO SOLO GANADORES', 25, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Análisis Venta en Cuenta Cliente por almacen', 21, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('VENTA POR CIUDAD ALMACEN COMPROBANTE DE CLIENTE', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Operaciones Confirmada', 24, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Operaciones Anuladas', 24, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Análisis Venta en Cuenta Cliente por vendedor', 21, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('Análisis Venta en Cuenta Cliente por cliente', 21, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('VENTA DE UN GRUPO DE CLIENTE', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('VENTA DE UN PRECIO DE CLIENTE', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)

```

VALUES ('Análisis Venta en Cuenta Cliente por grupo cliente', 21, NULL);

INSERT INTO REPMAN\_REPORTS (REPORT\_NAME, REPORT\_GROUP, USER\_FLAG)  
VALUES ('INVENTARIO', 2, NULL);

INSERT INTO REPMAN\_REPORTS (REPORT\_NAME, REPORT\_GROUP, USER\_FLAG)  
VALUES ('DOCUMENTOS DE TERCEROS TITULAR REMITIDOS', 24, NULL);

INSERT INTO REPMAN\_REPORTS (REPORT\_NAME, REPORT\_GROUP, USER\_FLAG)  
VALUES ('DOCUMENTO DE TERCEROS REMITIDOS', 24, NULL);

INSERT INTO REPMAN\_REPORTS (REPORT\_NAME, REPORT\_GROUP, USER\_FLAG)  
VALUES ('RESUMEN DE VENTAS POR VENDEDORES POR FECHA', 18, NULL);

INSERT INTO REPMAN\_REPORTS (REPORT\_NAME, REPORT\_GROUP, USER\_FLAG)  
VALUES ('DOCUMENTOS DE TERCEROS REMITIDOS POR FECHA', 24, NULL);

INSERT INTO REPMAN\_REPORTS (REPORT\_NAME, REPORT\_GROUP, USER\_FLAG)  
VALUES ('DOCUMENTOS DE TERCEROS PORTADOR REMITIDOS', 24, NULL);

INSERT INTO REPMAN\_REPORTS (REPORT\_NAME, REPORT\_GROUP, USER\_FLAG)  
VALUES ('DOCUMENTO DE TERCEROS A VENDECER', 24, NULL);

INSERT INTO REPMAN\_REPORTS (REPORT\_NAME, REPORT\_GROUP, USER\_FLAG)  
VALUES ('DOCUMENTO DE TERCEROS PORTADOR EN PODER', 24, NULL);

INSERT INTO REPMAN\_REPORTS (REPORT\_NAME, REPORT\_GROUP, USER\_FLAG)  
VALUES ('DOCUMENTO DE TERCERO SELECCIONADO', 24, NULL);

INSERT INTO REPMAN\_REPORTS (REPORT\_NAME, REPORT\_GROUP, USER\_FLAG)  
VALUES ('RESUMEN DE COBROS POR VENDEDORES POR FECHA', 18, NULL);

INSERT INTO REPMAN\_REPORTS (REPORT\_NAME, REPORT\_GROUP, USER\_FLAG)  
VALUES ('LISTADO DE VENTA DE PRODUCTOS POR CLIENTE', 7, NULL);

INSERT INTO REPMAN\_REPORTS (REPORT\_NAME, REPORT\_GROUP, USER\_FLAG)  
VALUES ('RESUMEN DE COBROS A CLIENTES POR GRUPO', 7, NULL);

INSERT INTO REPMAN\_REPORTS (REPORT\_NAME, REPORT\_GROUP, USER\_FLAG)  
VALUES ('LISTADO DE VENTA POR GRUPO PRODUCTO', 7, NULL);

INSERT INTO REPMAN\_REPORTS (REPORT\_NAME, REPORT\_GROUP, USER\_FLAG)  
VALUES ('LISTADO DE STOCK ALMACEN POR MARCA', 2, NULL);

INSERT INTO REPMAN\_REPORTS (REPORT\_NAME, REPORT\_GROUP, USER\_FLAG)  
VALUES ('MODELO MAC MOVIDA', 26, NULL);

INSERT INTO REPMAN\_REPORTS (REPORT\_NAME, REPORT\_GROUP, USER\_FLAG)  
VALUES ('MOVIMIENTO POR CONCEPTO EN UN ALMACEN POR CLASE', 2, NULL);

INSERT INTO REPMAN\_REPORTS (REPORT\_NAME, REPORT\_GROUP, USER\_FLAG)  
VALUES ('MODELO NUEVA VERSION', 26, NULL);

INSERT INTO REPMAN\_REPORTS (REPORT\_NAME, REPORT\_GROUP, USER\_FLAG)



```

VALUES ('MODELO TIPO MATRICIAL', 26, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO RECIPIENTE POR PRODUCTO', 7, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('MOVIMIENTO POR CONCEPTO EN UN ALMACEN POR MARCA', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE STOCK ALMACEN POR TIPO', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('DEVOLUCION', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('VENTAS POR FORMA DE PAGO', 8, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('GANANCIA POR PROVEEDOR GENERAL', 13, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE STOCK ALMACEN POR UNA CLASE', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE STOCK ALMACEN POR UNA MARCA', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE STOCK ALMACEN POR UNA SESION', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE STOCK CON SALDO 0', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('LISTADO DE STOCK ALMACEN POR UNA TIPO', 2, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('RESUMEN DE PRODUCCION', 1, NULL);

INSERT INTO REPMAN_REPORTS (REPORT_NAME, REPORT_GROUP, USER_FLAG)
VALUES ('GANANCIA POR PROVEEDOR POR TIPO', 13, NULL);

COMMIT WORK;

/* Privileges for the `REPMAN_GROUPS` : */

GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON REPMAN_GROUPS TO
SYSDBA WITH GRANT OPTION;

/* Privileges for the `REPMAN_REPORTS` : */

```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON REPMAN_REPORTS TO  
SYSDBA WITH GRANT OPTION;
```

```
/* SQL Manager for InterBase and Firebird 5.5.3.52046 */  
/* ----- */  
/* Host : localhost */  
/* Database : D:\Gabriela Fragancias\Base de datos\IMAGEN.FDB */
```

```
CREATE DATABASE 'D:\Gabriela Fragancias\Base de datos\IMAGEN.FDB'  
USER 'SYSDBA'  
PAGE_SIZE = 1024  
DEFAULT CHARACTER SET NONE;
```

```
/* Definition for the `DESCRIPCION` domain : */
```

```
CREATE DOMAIN DESCRIPCION AS  
VARCHAR(75);
```

```
/* Definition for the `ENTERO` domain : */
```

```
CREATE DOMAIN ENTERO AS  
INTEGER;
```

```
/* Definition for the `FECHA` domain : */
```

```
CREATE DOMAIN FECHA AS  
DATE;
```

```
/* Definition for the `IMAGEN` domain : */
```

```
CREATE DOMAIN IMAGEN AS  
BLOB SEGMENT SIZE 100;
```

```
/* Structure for the `IMAGEN_PRODUCTO` table : */
```

```
CREATE TABLE IMAGEN_PRODUCTO (  
ID_IMAGEN_PRODUCTO ENTERO NOT NULL,  
FECHA FECHA,  
IMAGEN IMAGEN,  
ID_PRODUCTO ENTERO,  
DESCRIPCION DESCRIPCION);
```

```
ALTER TABLE IMAGEN_PRODUCTO ADD CONSTRAINT PK_IMAGEN_PRODUCTO PRIMARY KEY  
(ID_IMAGEN_PRODUCTO);
```

```
/* Definition for the `IMAGEN_PRODUCTO_GEN` generator : */
```

```
CREATE GENERATOR IMAGEN_PRODUCTO_GEN;
```

```
SET GENERATOR IMAGEN_PRODUCTO_GEN TO 9;
```

```
/* Definition for the `DISPARADORPRODUCTO` trigger : */
```

```

SET TERM ^ ;

CREATE TRIGGER DISPARADORPRODUCTO FOR IMAGEN_PRODUCTO
ACTIVE BEFORE
INSERT
POSITION 0
AS
BEGIN
    NEW.ID_IMAGEN_PRODUCTO = GEN_ID(IMAGEN_PRODUCTO_GEN, 1);
END^

SET TERM ; ^

/* Structure for the `IMAGEN_CLIENTE` table : */

CREATE TABLE IMAGEN_CLIENTE (
    ID_IMAGEN_CLIENTE ENTERO NOT NULL,
    ID_CLIENTE ENTERO,
    FECHA FECHA,
    DESCRIPCION DESCRIPCION,
    IMAGEN IMAGEN,
    ID_USUARIO ENTERO);

ALTER TABLE IMAGEN_CLIENTE ADD CONSTRAINT PK_IMAGEN_CLIENTE PRIMARY KEY
(ID_IMAGEN_CLIENTE);

/* Structure for the `IMAGEN_FINANCIERO` table : */

CREATE TABLE IMAGEN_FINANCIERO (
    ID_IMAGEN_FINANCIERO ENTERO NOT NULL,
    ID_OPERACION_FINANCIERO ENTERO NOT NULL,
    DESCRIPCION DESCRIPCION,
    IMAGEN IMAGEN);

ALTER TABLE IMAGEN_FINANCIERO ADD CONSTRAINT PK_IMAGEN_FINANCIERO PRIMARY
KEY (ID_IMAGEN_FINANCIERO);

/* Definition for the `FIB$BOOLEAN` domain : */

CREATE DOMAIN FIB$BOOLEAN AS
    SMALLINT
    DEFAULT 1
    NOT NULL
    CHECK (VALUE IN (0,1));

/* Definition for the `CEILING` UDF : */

DECLARE EXTERNAL FUNCTION "CEILING"
    DOUBLE PRECISION
    RETURNS DOUBLE PRECISION BY VALUE
    ENTRY_POINT 'IB_UDF_ceiling' MODULE_NAME 'ib_udf';

```

```

/* Definition for the `FLOOR` UDF : */

DECLARE EXTERNAL FUNCTION "FLOOR"
  DOUBLE PRECISION
RETURNS DOUBLE PRECISION BY VALUE
ENTRY_POINT 'IB_UDF_floor' MODULE_NAME 'ib_udf';

/* Data for the 'IMAGEN_CLIENTE' table (Records 1 - 8) */

INSERT INTO IMAGEN_CLIENTE (ID_IMAGEN_CLIENTE, ID_CLIENTE, FECHA,
DESCRIPCION, ID_USUARIO)
VALUES (1, 3, '2014-06-23', 'Cedula', 2);

INSERT INTO IMAGEN_CLIENTE (ID_IMAGEN_CLIENTE, ID_CLIENTE, FECHA,
DESCRIPCION, ID_USUARIO)
VALUES (2, 3, '2014-06-23', 'Cedula atras', 2);

INSERT INTO IMAGEN_CLIENTE (ID_IMAGEN_CLIENTE, ID_CLIENTE, FECHA,
DESCRIPCION, ID_USUARIO)
VALUES (3, 3, '2014-06-23', 'Otros', 2);

INSERT INTO IMAGEN_CLIENTE (ID_IMAGEN_CLIENTE, ID_CLIENTE, FECHA,
DESCRIPCION, ID_USUARIO)
VALUES (4, 3, '2014-06-23', 'capaz', 2);

INSERT INTO IMAGEN_CLIENTE (ID_IMAGEN_CLIENTE, ID_CLIENTE, FECHA,
DESCRIPCION, ID_USUARIO)
VALUES (5, 980, '2014-06-23', 'Cedula', 2);

INSERT INTO IMAGEN_CLIENTE (ID_IMAGEN_CLIENTE, ID_CLIENTE, FECHA,
DESCRIPCION, ID_USUARIO)
VALUES (6, 504, '2016-05-02', 'Cara', 1);

INSERT INTO IMAGEN_CLIENTE (ID_IMAGEN_CLIENTE, ID_CLIENTE, FECHA,
DESCRIPCION, ID_USUARIO)
VALUES (7, 504, '2016-05-02', 'Reverso', 1);

INSERT INTO IMAGEN_CLIENTE (ID_IMAGEN_CLIENTE, ID_CLIENTE, FECHA,
DESCRIPCION, ID_USUARIO)
VALUES (8, 1, '2017-05-04', 'foto', 5);

COMMIT WORK;

/* Privileges for the `IMAGEN_CLIENTE` : */

GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON IMAGEN_CLIENTE TO
SYSDBA WITH GRANT OPTION;

/* Privileges for the `IMAGEN_FINANCIERO` : */

```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON IMAGEN_FINANCIERO TO
SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `IMAGEN_PRODUCTO` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON IMAGEN_PRODUCTO TO
SYSDBA WITH GRANT OPTION;
```

```
/* SQL Manager for InterBase and Firebird 5.5.3.52046      */
/* ----- */
/* Host      : localhost */
/* Database : D:\Gabriela Fragancias\Base de datos\SMS.FDB */
```

```
CREATE DATABASE 'D:\Gabriela Fragancias\Base de datos\SMS.FDB'
  USER 'SYSDBA'
  PAGE_SIZE = 1024
  DEFAULT CHARACTER SET NONE;
```

```
/* Definition for the `DESCRIPCION` domain : */
```

```
CREATE DOMAIN DESCRIPCION AS
  VARCHAR(60);
```

```
/* Definition for the `ENTERO` domain : */
```

```
CREATE DOMAIN ENTERO AS
  INTEGER;
```

```
/* Definition for the `FECHA` domain : */
```

```
CREATE DOMAIN FECHA AS
  DATE;
```

```
/* Definition for the `HORA` domain : */
```

```
CREATE DOMAIN HORA AS
  TIME;
```

```
/* Definition for the `IMAGEN` domain : */
```

```
CREATE DOMAIN IMAGEN AS
  BLOB SEGMENT SIZE 100;
```

```
/* Definition for the `MENSAJE` domain : */
```

```
CREATE DOMAIN MENSAJE AS
  VARCHAR(500);
```

```
/* Definition for the `SI_NO_SI` domain : */
```

```
CREATE DOMAIN SI_NO_SI AS
```

```

CHAR(1);

/* Structure for the `EMAIL` table : */

CREATE TABLE EMAIL (
    ID_EMAIL ENTERO NOT NULL,
    EMAIL DESCRIPCION,
    ASUNTO DESCRIPCION,
    ADJUNTO IMAGEN,
    CONTENIDO MENSAJE,
    ENVIADO SI_NO_SI,
    FECHA FECHA,
    HORA HORA,
    MODULO ENTERO,
    REFERENCIA ENTERO);

ALTER TABLE EMAIL ADD CONSTRAINT PK_EMAIL PRIMARY KEY (ID_EMAIL);

/* Definition for the `EMAIL_BI` trigger : */

SET TERM ^ ;

CREATE TRIGGER EMAIL_BI FOR EMAIL
ACTIVE AFTER
    INSERT
POSITION 0
AS
BEGIN
    /* Trigger body */
    post_event 'NUEVOEMAIL';

END^

SET TERM ; ^

/* Definition for the `NUMERODESTINATARIO` domain : */

CREATE DOMAIN NUMERODESTINATARIO AS
    VARCHAR(15);

/* Structure for the `SMS` table : */

CREATE TABLE SMS (
    ID_SMS ENTERO NOT NULL,
    DESTINATARIO NUMERODESTINATARIO,
    MENSAJE MENSAJE,
    ENVIADO SI_NO_SI,
    FECHA FECHA,
    HORA HORA,
    MODULO ENTERO,
    REFERENCIA ENTERO,
    FECHAENVIADO FECHA,
    ID_CLIENTE ENTERO);

```

```

ALTER TABLE SMS ADD CONSTRAINT PK_SMS PRIMARY KEY (ID_SMS);

/* Definition for the `SMS_NEWSMS` trigger : */

SET TERM ^ ;

CREATE TRIGGER SMS_NEWSMS FOR SMS
ACTIVE AFTER
INSERT
POSITION 0
AS
BEGIN
    /* Trigger body */
    post_event 'NUEVOSMS';
END^

SET TERM ; ^

/* Structure for the `ALDO` table : */

CREATE TABLE ALDO (
    NUM_ALDO DESCRIPCION NOT NULL);

ALTER TABLE ALDO ADD CONSTRAINT PK_ALDO PRIMARY KEY (NUM_ALDO);

/* Structure for the `TIPO_CONEXION` table : */

CREATE TABLE TIPO_CONEXION (
    ID_TIPO_CONEXION ENTERO NOT NULL,
    DESCRIPCION DESCRIPCION);

ALTER TABLE TIPO_CONEXION ADD CONSTRAINT PK_TIPO_CONEXION PRIMARY KEY
(ID_TIPO_CONEXION);

/* Structure for the `CONFIGURACION` table : */

CREATE TABLE CONFIGURACION (
    ID_CONFIGURACION ENTERO NOT NULL,
    DESCRIPCION DESCRIPCION,
    ACTIVO SI_NO_SI,
    ID_TIPO_CONEXION ENTERO,
    PROTOCOLO ENTERO,
    PARAMETROCONEXION DESCRIPCION,
    CONEXIONPING DESCRIPCION,
    INTERVALOAGOTADO ENTERO,
    INTERVALOENVIO ENTERO,
    INTERVALOREINTENTO ENTERO,
    VALIDEZMENSAJE ENTERO,
    CONCATENARSMS ENTERO);

```

```
ALTER TABLE CONFIGURACION ADD CONSTRAINT PK_CONFIGURACION PRIMARY KEY
(ID_CONFIGURACION);
```

```
/* Structure for the `SMS_ENTRADA` table : */
```

```
CREATE TABLE SMS_ENTRADA (
  ID_SMS_ENTRADA ENTERO NOT NULL,
  REMITENTE NUMERODESTINATARIO,
  MENSAJE MENSAJE,
  LEIDO SI_NO_SI,
  FECHA FECHA,
  HORA HORA,
  ID_CLIENTE ENTERO);
```

```
ALTER TABLE SMS_ENTRADA ADD CONSTRAINT PK_SMS_ENTRADA PRIMARY KEY
(ID_SMS_ENTRADA);
```

```
/* Definition for the `FIB$BOOLEAN` domain : */
```

```
CREATE DOMAIN FIB$BOOLEAN AS
  SMALLINT
  DEFAULT 1
  NOT NULL
  CHECK (VALUE IN (0,1));
```

```
/* Structure for the `UCTABRIGHTS` table : */
```

```
CREATE TABLE UCTABRIGHTS (
  UCIDUSER INTEGER,
  UCMODULE VARCHAR(50),
  UCCOMPNAME VARCHAR(50),
  UCKEY VARCHAR(250));
```

```
/* Structure for the `UCTABRIGHTSEX` table : */
```

```
CREATE TABLE UCTABRIGHTSEX (
  UCIDUSER INTEGER,
  UCMODULE VARCHAR(50),
  UCCOMPNAME VARCHAR(50),
  UCFORMNAME VARCHAR(50),
  UCKEY VARCHAR(250));
```

```
/* Structure for the `UCTABUSERS` table : */
```

```
CREATE TABLE UCTABUSERS (
  UCIDUSER INTEGER,
  UCUSERNAME VARCHAR(30),
  UCLOGIN VARCHAR(30),
  UCPASSWORD VARCHAR(250),
  UCASSEXPRIED CHAR(10),
  UCUSEREXPIED INTEGER,
  UCUSERDAYSSUN INTEGER,
```



```

UCEMAIL VARCHAR(150),
UCPRIVILEGED INTEGER,
UCTYPEREC CHAR(1),
UCPROFILE INTEGER,
UCKEY VARCHAR(250));

/* Definition for the `CEILING` UDF : */

DECLARE EXTERNAL FUNCTION "CEILING"
  DOUBLE PRECISION
RETURNS DOUBLE PRECISION BY VALUE
ENTRY_POINT 'IB_UDF_ceiling' MODULE_NAME 'ib_udf';

/* Definition for the `FLOOR` UDF : */

DECLARE EXTERNAL FUNCTION "FLOOR"
  DOUBLE PRECISION
RETURNS DOUBLE PRECISION BY VALUE
ENTRY_POINT 'IB_UDF_floor' MODULE_NAME 'ib_udf';

/* Data for the 'EMAIL' table (Records 1 - 22) */

INSERT INTO EMAIL (ID_EMAIL, EMAIL, ASUNTO, CONTENIDO, ENVIADO, FECHA,
HORA, MODULO, REFERENCIA)
VALUES (1, 'mariomartinez@erp-solucion.com', 'Pago de cuota', 'Señor/a:
MARIO MARTINEZ ZAPORITTI se ha registrado un pago de : 1550000 ENRRIQUE
CACERES S.R.L.', 'N', '2012-06-27', NULL, 2, 1);

INSERT INTO EMAIL (ID_EMAIL, EMAIL, ASUNTO, CONTENIDO, ENVIADO, FECHA,
HORA, MODULO, REFERENCIA)
VALUES (7, 'vilmaflorentin@hotmail.com', 'Pago de cuota', 'Señor/a: VILMA
ADELA FLORENTIN ROMERO se ha registrado un pago de : 100000 ENRRIQUE
CACERES S.R.L.', 'N', '2012-07-06', NULL, 2, 1);

COMMIT WORK;

/* Data for the 'SMS' table (Records 1 - 500) */

INSERT INTO SMS
VALUES (3642, '0982-623-903', 'Prueba...', 'N', '2017-05-29', NULL, 0, 0,
NULL, 3722);

COMMIT WORK;

/* Data for the 'ALDO' table (Records 1 - 500) */

INSERT INTO ALDO
VALUES ('0981862674');

INSERT INTO ALDO

```

```
VALUES ('0986231313');
```

```
/* Data for the 'ALDO' table (Records 501 - 1000) */
```

```
INSERT INTO ALDO  
VALUES ('0986226676');
```

```
/* Data for the 'ALDO' table (Records 20001 - 20001) */
```

```
INSERT INTO ALDO  
VALUES ('');
```

```
COMMIT WORK;
```

```
/* Data for the 'TIPO_CONEXION' table (Records 1 - 4) */
```

```
INSERT INTO TIPO_CONEXION  
VALUES (1, 'Physical/Virtual COM Port');
```

```
INSERT INTO TIPO_CONEXION  
VALUES (2, 'Infrarojo');
```

```
INSERT INTO TIPO_CONEXION  
VALUES (3, 'Bluetooth');
```

```
INSERT INTO TIPO_CONEXION  
VALUES (4, 'USB');
```

```
COMMIT WORK;
```

```
/* Data for the 'CONFIGURACION' table (Records 1 - 1) */
```

```
INSERT INTO CONFIGURACION  
VALUES (1, 'ZTE Modem', 'S', 1, 1, 'COM29, 9600', NULL, 1, 28, 1, 5,  
NULL);
```

```
COMMIT WORK;
```

```
/* Data for the 'SMS_ENTRADA' table (Records 1 - 1) */
```

```
INSERT INTO SMS_ENTRADA  
VALUES (1, '981268140', 'Recibido', 'S', '2014-07-08', '08:25:00', 1513);
```

```
COMMIT WORK;
```

```
/* Data for the 'UCTABRIGHTS' table (Records 1 - 20) */
```

```
INSERT INTO UCTABRIGHTS  
VALUES (2, 'ERPSMSSSEND', 'Configuraciones', 'ykusjB4R68Qy8mkoyso41D');
```

```
INSERT INTO UCTABRIGHTS
```

```

VALUES (2, 'ERPSMSSSEND', 'SMSEnviado', 'ykfi3vQPL6IC33L');

INSERT INTO UCTABRIGHTS
VALUES (2, 'ERPSMSSSEND', 'EnvioSMS', 'y8u2aWKyZQVc');

INSERT INTO UCTABRIGHTS
VALUES (2, 'ERPSMSSSEND', 'GenerardeCuentaClientes',
'y0ucluTvSzKVgCRl5NzDohATN4e48QgC');

INSERT INTO UCTABRIGHTS
VALUES (2, 'ERPSMSSSEND', 'MensajeaLista', 'yceJIhuMftIFkBru8RF');

INSERT INTO UCTABRIGHTS
VALUES (2, 'ERPSMSSSEND', 'MensajesLibres', 'yceJIhuMftoBSeRtfvvE');

INSERT INTO UCTABRIGHTS
VALUES (2, 'ERPSMSSSEND', 'CerrarSesion', 'ykOuR6ebPJK8v3qbqD');

INSERT INTO UCTABRIGHTS
VALUES (2, 'ERPSMSSSEND', 'CombioPass', 'ykusglBOB09DgXI');

INSERT INTO UCTABRIGHTS
VALUES (2, 'ERPSMSSSEND', 'Perfiles', 'yov9RhQSMclJiD');

INSERT INTO UCTABRIGHTS
VALUES (2, 'ERPSMSSSEND', 'Usuarios', 'y8P6oKz09rNX');

INSERT INTO UCTABRIGHTS
VALUES (1, 'ERPSMSSSEND', 'Configuraciones', 'xgZnolOjCkEMzGLAnWo/gA');

INSERT INTO UCTABRIGHTS
VALUES (1, 'ERPSMSSSEND', 'SMSEnviado', 'xgYQjtnAeQud4kO');

INSERT INTO UCTABRIGHTS
VALUES (1, 'ERPSMSSSEND', 'EnvioSMS', 'x45pC/vK3nDx');

INSERT INTO UCTABRIGHTS
VALUES (1, 'ERPSMSSSEND', 'GenerardeCuentaClientes',
'xwZSaHK2Z3OdHIT1+XVFxKh4XPQnsWur');

INSERT INTO UCTABRIGHTS
VALUES (1, 'ERPSMSSSEND', 'MensajeaLista', 'xY5M2YK9l232oD+a5eB');

INSERT INTO UCTABRIGHTS
VALUES (1, 'ERPSMSSSEND', 'MensajesLibres', 'xY5M2YK9l2Xysh991b6d');

INSERT INTO UCTABRIGHTS
VALUES (1, 'ERPSMSSSEND', 'CerrarSesion', 'xg5ldB2Pk9XmBtXTgB');

INSERT INTO UCTABRIGHTS
VALUES (1, 'ERPSMSSSEND', 'CombioPass', 'xgZnrl5W38LBL5C');

INSERT INTO UCTABRIGHTS
VALUES (1, 'ERPSMSSSEND', 'Perfiles', 'xsoErhGLWDpF0D');

```

```

INSERT INTO UCTABRIGHTS
VALUES (1, 'ERPSMSSSEND', 'Usuarios', 'x4Ykk836A5LK');

COMMIT WORK;

/* Data for the 'UCTABUSERS' table (Records 1 - 2) */

INSERT INTO UCTABUSERS
VALUES (1, 'admin', 'admin', 'jU/GVqfSaD', '24/06/2012', 0, 30,
'soporte@hoserpsolucion.com', -1, 'U', 0, 'xobOIWXveVL9x3I9DC');

INSERT INTO UCTABUSERS
VALUES (2, 'Mario Martinez', 'Mario2', '', '11/12/2015', 0, 10, '', -1,
'U', 0, 'yceIeO+VdB');

COMMIT WORK;

/* Structure for the `CONFIGURACION` table : */

ALTER TABLE CONFIGURACION ADD CONSTRAINT FK_ID_TIPO_CONEXION FOREIGN KEY
(ID_TIPO_CONEXION) REFERENCES TIPO_CONEXION(ID_TIPO_CONEXION) ON UPDATE
CASCADE;

/* Privileges for the `ALDO` : */

GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON ALDO TO SYSDBA WITH
GRANT OPTION;

/* Privileges for the `CONFIGURACION` : */

GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON CONFIGURACION TO
SYSDBA WITH GRANT OPTION;

/* Privileges for the `EMAIL` : */

GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON EMAIL TO SYSDBA WITH
GRANT OPTION;

/* Privileges for the `SMS` : */

GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON SMS TO SYSDBA WITH
GRANT OPTION;

/* Privileges for the `SMS_ENTRADA` : */

```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON SMS_ENTRADA TO SYSDBA  
WITH GRANT OPTION;
```

```
/* Privileges for the `TIPO_CONEXION` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON TIPO_CONEXION TO  
SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `UCTABRIGHTS` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON UCTABRIGHTS TO SYSDBA  
WITH GRANT OPTION;
```

```
/* Privileges for the `UCTABRIGHTSEX` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON UCTABRIGHTSEX TO  
SYSDBA WITH GRANT OPTION;
```

```
/* Privileges for the `UCTABUSERS` : */
```

```
GRANT SELECT, INSERT, DELETE, REFERENCES, UPDATE ON UCTABUSERS TO SYSDBA  
WITH GRANT OPTION;
```