

INSTITUTO TECNOLÓGICO VERACRUZ

Manual Técnico

Taller de Base de Datos

Pérez Muñoz Luis Ángel

Rey Méndez José Alfredo

Márquez Solano José Ramón

12/06/2015

Contenido

Instalación de la base de datos	2
Diccionario de datos.....	29
Modelo Relacional.....	31
Constraints	32
Vistas	34
Triggers.....	35
Procedures	37

Instalación de la base de datos

- Para crear la base de datos, es necesario tener una instancia de Oracle versión 11g o superior, tener acceso a **SYSDBA**, la **instancia de Oracle** debe de tener el nombre por defecto **XE**. Primero que nada hay que crear un usuario con el nombre LUIS que será el esquema que contendrá la BD, lo podemos crear así:

Create user LUIS identified by 1234 default tablespace users;

- Después de crear el usuario, es necesario cederle los permisos necesarios, debido a que el esquema que contiene la base de datos será el administrador de la misma, se le ceden todos los derechos de la siguiente manera:

Grant all privileges to Luis identified by 1234

- Después de crear el usuario, se procede a crear la base de datos, y una tabla independiente llamada log, que servirá para grabar los registros de la misma base de datos, para crearla, se debe de ejecutar los siguientes scripts uno por uno, del orden que aquí está presente.

-- Archivo creado - martes-junio-09-2015

-- DDL for Table LOGS

CREATE TABLE "LUIS"."LOGS"

**("USUARIO" VARCHAR2(40 BYTE),
"TABLA_MODIFICADA" VARCHAR2(40 BYTE),
"FECHA_MOD" DATE**

```

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE
DEFAULT CELL_FLASH_CACHE DEFAULT)

TABLESPACE "USERS" ;

REM INSERTING into LUIS.LOGS

SET DEFINE OFF;

-----

-- Archivo creado - martes-junio-09-2015

-----

-----

-- DDL for Table ALMACEN

-----

CREATE TABLE "LUIS"."ALMACEN"
(
  "IDALMACEN" CHAR(4 BYTE),
  "DESCRIPCIONALMACEN" VARCHAR2(80 BYTE)
) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE
DEFAULT CELL_FLASH_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

REM INSERTING into LUIS.ALMACEN

```

SET DEFINE OFF;

Insert into LUIS.ALMACEN (IDALMACEN,DESCRIPCIONALMACEN) values ('1 ', 'Almacen 1');

-- DDL for Index PK_ALMACEN

CREATE UNIQUE INDEX "LUIS"."PK_ALMACEN" ON "LUIS"."ALMACEN" ("IDALMACEN")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE
DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "SYSTEM" ;

-- Constraints for Table ALMACEN

ALTER TABLE "LUIS"."ALMACEN" ADD CONSTRAINT "PK_ALMACEN" PRIMARY KEY
("IDALMACEN")
USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE
DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "SYSTEM" ENABLE;
ALTER TABLE "LUIS"."ALMACEN" MODIFY ("IDALMACEN" NOT NULL ENABLE);

-- DDL for Trigger TRALMACEN

```
CREATE OR REPLACE TRIGGER "LUIS"."TRALMACEN"
after insert or delete or update on almacen

begin

if inserting or deleting or updating then

    insert into logs (usuario, tabla_modificada, fecha_mod) values (user, 'almacen', SYSDATE);

    dbms_output.put_line( 'Se modifico almacen' );

end if;

end ;

/

ALTER TRIGGER "LUIS"."TRALMACEN" ENABLE;
```

```
-----
-- Archivo creado - martes-junio-09-2015
-----
```

```
-----
-- DDL for Table CLIENTE
-----
```

```
CREATE TABLE "LUIS"."CLIENTE"
(
    "IDCLIENTE" CHAR(4 BYTE),
    "NOMBRECLIENTE" VARCHAR2(45 BYTE),
    "APPATERNO" VARCHAR2(15 BYTE),
    "APMATERNO" VARCHAR2(15 BYTE),
    "DIRECCIONCLIENTE" VARCHAR2(80 BYTE),
```

```

"CORREOCLIENTE" VARCHAR2(40 BYTE),

"TELEFONOCLIENTE" NUMBER(10,0) DEFAULT 0,

"SEXOCLIENTE" CHAR(1 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE
DEFAULT CELL_FLASH_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

REM INSERTING into LUIS.CLIENTE

SET DEFINE OFF;

Insert into LUIS.CLIENTE
(IDCLIENTE,NOMBRECLIENTE,APPATERNO,APMATERNO,DIRECCIONCLIENTE,CORREOCLIENTE,TEL
EFONOCLIENTE,SEXOCLIENTE) values ('2  ', 'Rafael ', 'Marquez', 'Solano', 'Victoria
12', 'rafitamar@gmail.com', 4192382, 'M');

Insert into LUIS.CLIENTE
(IDCLIENTE,NOMBRECLIENTE,APPATERNO,APMATERNO,DIRECCIONCLIENTE,CORREOCLIENTE,TEL
EFONOCLIENTE,SEXOCLIENTE) values ('3  ', 'Francisco', 'Rodriguez', 'Madero', 'Juarez
20', 'pacorodriguez@yahoo.com', 2947726, 'M');

Insert into LUIS.CLIENTE
(IDCLIENTE,NOMBRECLIENTE,APPATERNO,APMATERNO,DIRECCIONCLIENTE,CORREOCLIENTE,TEL
EFONOCLIENTE,SEXOCLIENTE) values ('4  ', 'Hector', 'Moreno', 'Blanco', 'Heroes de Puebla
201', 'hectormorblan@gmail.com', 9820123, 'M');

Insert into LUIS.CLIENTE
(IDCLIENTE,NOMBRECLIENTE,APPATERNO,APMATERNO,DIRECCIONCLIENTE,CORREOCLIENTE,TEL
EFONOCLIENTE,SEXOCLIENTE) values ('1  ', 'Jesus', 'Corona', 'Modelo', 'Carlos Cruz
10', 'chuycorona@hotmail.com', 1234656, 'M');

Insert into LUIS.CLIENTE
(IDCLIENTE,NOMBRECLIENTE,APPATERNO,APMATERNO,DIRECCIONCLIENTE,CORREOCLIENTE,TEL
EFONOCLIENTE,SEXOCLIENTE) values ('5  ', 'Paul', 'Aguilar', 'Cruz', 'Hidalgo
291', 'paulaguilar@hotmail.com', 3783461, 'M');

```

Insert into LUIS.CLIENTE

(IDCLIENTE,NOMBRECLIENTE,APPATERNO,APMATERNO,DIRECCIONCLIENTE,CORREOCLIENTE,TELEFONOCLIENTE,SEXOCLIENTE) values ('6 ','Miguel','Layun','Fernandez','Mancilla 284','todoesculpadelayun@gmail.com','7894622','M');

Insert into LUIS.CLIENTE

(IDCLIENTE,NOMBRECLIENTE,APPATERNO,APMATERNO,DIRECCIONCLIENTE,CORREOCLIENTE,TELEFONOCLIENTE,SEXOCLIENTE) values ('7 ','Jose','Vazquez','Vazquez','qwerty 78','gallovazquez@gmail.com','9187633','M');

Insert into LUIS.CLIENTE

(IDCLIENTE,NOMBRECLIENTE,APPATERNO,APMATERNO,DIRECCIONCLIENTE,CORREOCLIENTE,TELEFONOCLIENTE,SEXOCLIENTE) values ('8 ','Hector','Herrera','Guzman','Arteaga 28','hectorherrera@hotmail.com','3126745','M');

Insert into LUIS.CLIENTE

(IDCLIENTE,NOMBRECLIENTE,APPATERNO,APMATERNO,DIRECCIONCLIENTE,CORREOCLIENTE,TELEFONOCLIENTE,SEXOCLIENTE) values ('9 ','Andres','Guardado','Amione','Independencia 23','andresguarda@hotmail.com','2375423','M');

Insert into LUIS.CLIENTE

(IDCLIENTE,NOMBRECLIENTE,APPATERNO,APMATERNO,DIRECCIONCLIENTE,CORREOCLIENTE,TELEFONOCLIENTE,SEXOCLIENTE) values ('10 ','Giovanni','dos Santos','Aveiro','5 de mayo 56','belindagolosa69@yahoo.com','4736241','M');

Insert into LUIS.CLIENTE

(IDCLIENTE,NOMBRECLIENTE,APPATERNO,APMATERNO,DIRECCIONCLIENTE,CORREOCLIENTE,TELEFONOCLIENTE,SEXOCLIENTE) values ('11 ','Javier','Hernandez','Balcazar','España 89','chicharito14@hotmail.com','2734685','M');

Insert into LUIS.CLIENTE

(IDCLIENTE,NOMBRECLIENTE,APPATERNO,APMATERNO,DIRECCIONCLIENTE,CORREOCLIENTE,TELEFONOCLIENTE,SEXOCLIENTE) values ('12 ','Gabriela','Rodriguez ','Medina','Procuraduria 82','gabyrodriguez@gmail.com','5637426','F');

-- DDL for Index PK_CLIENTE

CREATE UNIQUE INDEX "LUIS"."PK_CLIENTE" ON "LUIS"."CLIENTE" ("IDCLIENTE")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS


```

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE
DEFAULT CELL_FLASH_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

-----

-- DDL for Index U_CORREOCLIENTE

-----

CREATE UNIQUE INDEX "LUIS"."U_CORREOCLIENTE" ON "LUIS"."CLIENTE" ("CORREOCLIENTE")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE
DEFAULT CELL_FLASH_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

-----

-- DDL for Index U_TELEFONOCIENTE

-----

CREATE UNIQUE INDEX "LUIS"."U_TELEFONOCIENTE" ON "LUIS"."CLIENTE"
("TELEFONOCIENTE")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE
DEFAULT CELL_FLASH_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

-----

-- Constraints for Table CLIENTE

```

```
ALTER TABLE "LUIS"."CLIENTE" ADD CONSTRAINT "U_TELEFONOCIENTE" UNIQUE  
("TELEFONOCIENTE")
```

```
USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
```

```
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
```

```
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE  
DEFAULT CELL_FLASH_CACHE DEFAULT)
```

```
TABLESPACE "SYSTEM" ENABLE;
```

```
ALTER TABLE "LUIS"."CLIENTE" ADD CONSTRAINT "U_CORREOCIENTE" UNIQUE  
("CORREOCIENTE")
```

```
USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
```

```
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
```

```
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE  
DEFAULT CELL_FLASH_CACHE DEFAULT)
```

```
TABLESPACE "SYSTEM" ENABLE;
```

```
ALTER TABLE "LUIS"."CLIENTE" ADD CONSTRAINT "CK_SEXO" CHECK (sexocliente = 'M' or  
sexocliente = 'F') ENABLE;
```

```
ALTER TABLE "LUIS"."CLIENTE" ADD CONSTRAINT "PK_CLIENTE" PRIMARY KEY ("IDCLIENTE")
```

```
USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
```

```
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
```

```
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE  
DEFAULT CELL_FLASH_CACHE DEFAULT)
```

```
TABLESPACE "SYSTEM" ENABLE;
```

```
ALTER TABLE "LUIS"."CLIENTE" MODIFY ("SEXOCIENTE" NOT NULL ENABLE);
```

```
ALTER TABLE "LUIS"."CLIENTE" MODIFY ("DIRECCIONCLIENTE" NOT NULL ENABLE);
```

```
ALTER TABLE "LUIS"."CLIENTE" MODIFY ("APMATERNO" NOT NULL ENABLE);
```

```
ALTER TABLE "LUIS"."CLIENTE" MODIFY ("APPATERNO" NOT NULL ENABLE);
```

```
ALTER TABLE "LUIS"."CLIENTE" MODIFY ("NOMBRECLIENTE" NOT NULL ENABLE);
```

```
ALTER TABLE "LUIS"."CLIENTE" MODIFY ("IDCLIENTE" NOT NULL ENABLE);
```

```
-----  
-- DDL for Trigger TRCLIENTE  
-----
```

```
CREATE OR REPLACE TRIGGER "LUIS"."TRCLIENTE"
```

```
after insert or delete or update on cliente
```

```
begin
```

```
if inserting or deleting or updating then
```

```
    insert into logs (usuario, tabla_modificada, fecha_mod) values (user, 'cliente', SYSDATE);
```

```
    dbms_output.put_line( 'Se modifiko cliente' );
```

```
end if;
```

```
end ;
```

```
/
```

```
ALTER TRIGGER "LUIS"."TRCLIENTE" ENABLE;
```

```
-----  
-- Archivo creado - martes-junio-09-2015  
-----  
-----
```

```
-- DDL for Table PRODUCTO
```

```

-----

CREATE TABLE "LUIS"."PRODUCTO"

(
    "IDPRODUCTO" CHAR(4 BYTE),

    "NOMPRODUCTO" VARCHAR2(15 BYTE),

    "PRECIOPRODUCTO" FLOAT(126),

    "DESCRIPCIONPRODUCTO" VARCHAR2(80 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE
DEFAULT CELL_FLASH_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

REM INSERTING into LUIS.PRODUCTO

SET DEFINE OFF;

Insert into LUIS.PRODUCTO
(IDPRODUCTO,NOMPRODUCTO,PRECIOPRODUCTO,DESCRIPCIONPRODUCTO) values ('1
','Playera
',400,'100 % Algodon , Colores: Azul,Blanca,Roja,Amarilla. Tallas G.M.CH.ECH');

Insert into LUIS.PRODUCTO
(IDPRODUCTO,NOMPRODUCTO,PRECIOPRODUCTO,DESCRIPCIONPRODUCTO) values ('2 ','Tenis
',600,'Tallas: 21-28 ');

Insert into LUIS.PRODUCTO
(IDPRODUCTO,NOMPRODUCTO,PRECIOPRODUCTO,DESCRIPCIONPRODUCTO) values ('3 ','Balon
Basquet',250,'Como el que usan en la NBA');

Insert into LUIS.PRODUCTO
(IDPRODUCTO,NOMPRODUCTO,PRECIOPRODUCTO,DESCRIPCIONPRODUCTO) values ('4
','Raqueta',150,'Mango metalico');

```

Insert into LUIS.PRODUCTO

(IDPRODUCTO,NOMPRODUCTO,PRECIOPRODUCTO,DESCRIPCIONPRODUCTO) values ('5', 'Rodilleras',100,'Colores: Blanca , Negro');

Insert into LUIS.PRODUCTO

(IDPRODUCTO,NOMPRODUCTO,PRECIOPRODUCTO,DESCRIPCIONPRODUCTO) values ('6', 'Coderas',100,'Colores: Blaca, Negro');

Insert into LUIS.PRODUCTO

(IDPRODUCTO,NOMPRODUCTO,PRECIOPRODUCTO,DESCRIPCIONPRODUCTO) values ('7', 'Casco',350,'Colores, blanco, negro');

Insert into LUIS.PRODUCTO

(IDPRODUCTO,NOMPRODUCTO,PRECIOPRODUCTO,DESCRIPCIONPRODUCTO) values ('8', 'BandasToallas',200,'90 % algodón 10% poliéster Marca: Nike Colores: Azul Blanco Rosa Rojo');

Insert into LUIS.PRODUCTO

(IDPRODUCTO,NOMPRODUCTO,PRECIOPRODUCTO,DESCRIPCIONPRODUCTO) values ('9', 'SmartBand',2000,'La SmartBand mas smart del mercado');

-- DDL for Index PK_PRODUCTO

CREATE UNIQUE INDEX "LUIS"."PK_PRODUCTO" ON "LUIS"."PRODUCTO" ("IDPRODUCTO")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE
DEFAULT CELL_FLASH_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

-- Constraints for Table PRODUCTO

```
ALTER TABLE "LUIS"."PRODUCTO" ADD CONSTRAINT "PK_PRODUCTO" PRIMARY KEY  
("IDPRODUCTO")
```

```
USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
```

```
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
```

```
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE  
DEFAULT CELL_FLASH_CACHE DEFAULT)
```

```
TABLESPACE "SYSTEM" ENABLE;
```

```
ALTER TABLE "LUIS"."PRODUCTO" MODIFY ("PRECIOPRODUCTO" NOT NULL ENABLE);
```

```
ALTER TABLE "LUIS"."PRODUCTO" MODIFY ("NOMPRODUCTO" NOT NULL ENABLE);
```

```
ALTER TABLE "LUIS"."PRODUCTO" MODIFY ("IDPRODUCTO" NOT NULL ENABLE);
```

```
-----  
-- DDL for Trigger TRPRODUCTO  
-----
```

```
CREATE OR REPLACE TRIGGER "LUIS"."TRPRODUCTO"
```

```
after insert or delete or update on producto
```

```
begin
```

```
if inserting or deleting or updating then
```

```
insert into logs (usuario, tabla_modificada, fecha_mod) values (user, 'producto', SYSDATE);
```

```
dbms_output.put_line( 'Se modifico producto' );
```

```
end if;
```

```
end ;
```

```
/
```

```
ALTER TRIGGER "LUIS"."TRPRODUCTO" ENABLE;
```

-- Archivo creado - martes-junio-09-2015

-- DDL for Table VENDEDOR

```
CREATE TABLE "LUIS"."VENDEDOR"
(
  "IDVENDEDOR" CHAR(4 BYTE),
  "NOMVENDEDOR" VARCHAR2(20 BYTE),
  "APPATERNOVENDEDOR" VARCHAR2(15 BYTE),
  "APMATERNOVENDEDOR" VARCHAR2(15 BYTE),
  "FECHANACVENDEDOR" DATE,
  "CORREOVENDEDOR" VARCHAR2(40 BYTE),
  "DIRECCIONVENDEDOR" VARCHAR2(80 BYTE),
  "SEXOVENDEDOR" VARCHAR2(1 BYTE),
  "FECHAINGRESOVENDEDOR" DATE,
  "SUELDOVENDEDOR" NUMBER
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
        PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE
        DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "SYSTEM" ;

REM INSERTING into LUIS.VENDEDOR

SET DEFINE OFF;
```

Insert into LUIS.VENDEDOR

```
(IDVENDEDOR,NOMVENDEDOR,APPATERNOVENDEDOR,APMATERNOVENDEDOR,FECHANACVENDEDOR,CORREOVENDEDOR,DIRECCIONVENDEDOR,SEXOVENDEDOR,FECHAINGRESOVENDEDOR,SUELDOVENDEDOR) values ('1', 'Fernando', 'Bobadilla', 'Contreras', to_date('05/10/93', 'DD/MM/RR'), 'mbobadilla@outlook.com', 'Carlos 123', 'M', to_date('26/02/13', 'DD/MM/RR'), 30000);
```

Insert into LUIS.VENDEDOR

```
(IDVENDEDOR,NOMVENDEDOR,APPATERNOVENDEDOR,APMATERNOVENDEDOR,FECHANACVENDEDOR,CORREOVENDEDOR,DIRECCIONVENDEDOR,SEXOVENDEDOR,FECHAINGRESOVENDEDOR,SUELDOVENDEDOR) values ('2', 'Jose Ramon', 'Marquez', 'Solano', to_date('06/01/92', 'DD/MM/RR'), 'jrms@gmail.com', 'cordoba 23', 'M', to_date('20/06/13', 'DD/MM/RR'), 15000);
```

Insert into LUIS.VENDEDOR

```
(IDVENDEDOR,NOMVENDEDOR,APPATERNOVENDEDOR,APMATERNOVENDEDOR,FECHANACVENDEDOR,CORREOVENDEDOR,DIRECCIONVENDEDOR,SEXOVENDEDOR,FECHAINGRESOVENDEDOR,SUELDOVENDEDOR) values ('3', 'Ruben', 'Perez', 'Rodriguez', to_date('22/06/94', 'DD/MM/RR'), 'crumac22@gmail.com', 'aeropuerto 6', 'M', to_date('14/02/14', 'DD/MM/RR'), 5000);
```

Insert into LUIS.VENDEDOR

```
(IDVENDEDOR,NOMVENDEDOR,APPATERNOVENDEDOR,APMATERNOVENDEDOR,FECHANACVENDEDOR,CORREOVENDEDOR,DIRECCIONVENDEDOR,SEXOVENDEDOR,FECHAINGRESOVENDEDOR,SUELDOVENDEDOR) values ('4', 'Luis', 'Perez', 'Muñoz', to_date('26/07/94', 'DD/MM/RR'), 'ingluisperez.m@outlook.com', 'Rio Esva 131-4 Lomas de Rio Medio 4', 'M', to_date('04/04/14', 'DD/MM/RR'), 9000);
```

Insert into LUIS.VENDEDOR

```
(IDVENDEDOR,NOMVENDEDOR,APPATERNOVENDEDOR,APMATERNOVENDEDOR,FECHANACVENDEDOR,CORREOVENDEDOR,DIRECCIONVENDEDOR,SEXOVENDEDOR,FECHAINGRESOVENDEDOR,SUELDOVENDEDOR) values ('5', 'Karen', 'Garcia', 'Robles', to_date('02/02/90', 'DD/MM/RR'), 'grkaren@gmail.com', 'Calle de la perdicion #89 Col.Margarita', 'F', to_date('02/02/14', 'DD/MM/RR'), 5555);
```

-- DDL for Index PK_VENDEDOR

CREATE UNIQUE INDEX "LUIS"."PK_VENDEDOR" ON "LUIS"."VENDEDOR" ("IDVENDEDOR")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE
DEFAULT CELL_FLASH_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

-- DDL for Index U_CORREOVENDEDOR

CREATE UNIQUE INDEX "LUIS"."U_CORREOVENDEDOR" ON "LUIS"."VENDEDOR"
("CORREOVENDEDOR")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE
DEFAULT CELL_FLASH_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

-- Constraints for Table VENDEDOR

ALTER TABLE "LUIS"."VENDEDOR" ADD CONSTRAINT "U_CORREOVENDEDOR" UNIQUE
("CORREOVENDEDOR")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE
DEFAULT CELL_FLASH_CACHE DEFAULT)

TABLESPACE "SYSTEM" ENABLE;

ALTER TABLE "LUIS"."VENDEDOR" ADD CONSTRAINT "CK_SEXOVENDEDOR" CHECK
(sexovendedor = 'M' or sexovendedor = 'F') ENABLE;

ALTER TABLE "LUIS"."VENDEDOR" ADD CONSTRAINT "PK_VENDEDOR" PRIMARY KEY
("IDVENDEDOR")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE
DEFAULT CELL_FLASH_CACHE DEFAULT)

TABLESPACE "SYSTEM" ENABLE;

ALTER TABLE "LUIS"."VENDEDOR" MODIFY ("SUELDOVENDEDOR" NOT NULL ENABLE);

ALTER TABLE "LUIS"."VENDEDOR" MODIFY ("FECHAINGRESOVENDEDOR" NOT NULL ENABLE);

ALTER TABLE "LUIS"."VENDEDOR" MODIFY ("SEXOVENDEDOR" NOT NULL ENABLE);

ALTER TABLE "LUIS"."VENDEDOR" MODIFY ("FECHANACVENDEDOR" NOT NULL ENABLE);

ALTER TABLE "LUIS"."VENDEDOR" MODIFY ("APMATERNOVENDEDOR" NOT NULL ENABLE);

ALTER TABLE "LUIS"."VENDEDOR" MODIFY ("APPATERNOVENDEDOR" NOT NULL ENABLE);

ALTER TABLE "LUIS"."VENDEDOR" MODIFY ("NOMVENDEDOR" NOT NULL ENABLE);

ALTER TABLE "LUIS"."VENDEDOR" MODIFY ("IDVENDEDOR" NOT NULL ENABLE);

-- DDL for Trigger TRVENDEDOR

CREATE OR REPLACE TRIGGER "LUIS"."TRVENDEDOR"

after insert or delete or update on vendedor

begin

if inserting or deleting or updating then

insert into logs (usuario, tabla_modificada, fecha_mod) values (user, 'vendedor', SYSDATE);

```
        dbms_output.put_line( 'Se modifico vendedor' );

    end if;

end ;

/

ALTER TRIGGER "LUIS"."TRVENDEDOR" ENABLE;
```

```
-----

-- Archivo creado - martes-junio-09-2015

-----

-----
```

```
-- DDL for Table GUARDA

-----
```

```
CREATE TABLE "LUIS"."GUARDA"

(      "IDGUARDA" CHAR(4 BYTE),

        "IDPRODUCTO" CHAR(4 BYTE),

        "IDALMACEN" CHAR(4 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

        PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE

        DEFAULT CELL_FLASH_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

REM INSERTING into LUIS.GUARDA

SET DEFINE OFF;

Insert into LUIS.GUARDA (IDGUARDA,IDPRODUCTO,IDALMACEN) values ('1 ','1 ','1 ');
```

Insert into LUIS.GUARDA (IDGUARDA,IDPRODUCTO,IDALMACEN) values ('2 ','2 ','1 ');

Insert into LUIS.GUARDA (IDGUARDA,IDPRODUCTO,IDALMACEN) values ('3 ','3 ','1 ');

Insert into LUIS.GUARDA (IDGUARDA,IDPRODUCTO,IDALMACEN) values ('4 ','4 ','1 ');

Insert into LUIS.GUARDA (IDGUARDA,IDPRODUCTO,IDALMACEN) values ('5 ','5 ','1 ');

Insert into LUIS.GUARDA (IDGUARDA,IDPRODUCTO,IDALMACEN) values ('6 ','6 ','1 ');

Insert into LUIS.GUARDA (IDGUARDA,IDPRODUCTO,IDALMACEN) values ('7 ','7 ','1 ');

Insert into LUIS.GUARDA (IDGUARDA,IDPRODUCTO,IDALMACEN) values ('8 ','8 ','1 ');

Insert into LUIS.GUARDA (IDGUARDA,IDPRODUCTO,IDALMACEN) values ('9 ','9 ','1 ');

-- DDL for Index PK_GUARDA

CREATE UNIQUE INDEX "LUIS"."PK_GUARDA" ON "LUIS"."GUARDA" ("IDGUARDA")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE
DEFAULT CELL_FLASH_CACHE DEFAULT)

TABSPACE "SYSTEM" ;

-- Constraints for Table GUARDA

ALTER TABLE "LUIS"."GUARDA" ADD CONSTRAINT "PK_GUARDA" PRIMARY KEY ("IDGUARDA")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE
DEFAULT CELL_FLASH_CACHE DEFAULT)

TABLESPACE "SYSTEM" ENABLE;

ALTER TABLE "LUIS"."GUARDA" MODIFY ("IDALMACEN" NOT NULL ENABLE);

ALTER TABLE "LUIS"."GUARDA" MODIFY ("IDPRODUCTO" NOT NULL ENABLE);

ALTER TABLE "LUIS"."GUARDA" MODIFY ("IDGUARDA" NOT NULL ENABLE);

-- Ref Constraints for Table GUARDA

ALTER TABLE "LUIS"."GUARDA" ADD CONSTRAINT "FK_GUARDA_ALMACEN" FOREIGN KEY
("IDALMACEN")

REFERENCES "LUIS"."ALMACEN" ("IDALMACEN") ENABLE;

ALTER TABLE "LUIS"."GUARDA" ADD CONSTRAINT "FK_GUARDA_PRODUCTO" FOREIGN KEY
("IDPRODUCTO")

REFERENCES "LUIS"."PRODUCTO" ("IDPRODUCTO") ENABLE;

-- DDL for Trigger TRGUARDA

CREATE OR REPLACE TRIGGER "LUIS"."TRGUARDA"

after insert or delete or update on guarda

begin

if inserting or deleting or updating then

insert into logs (usuario, tabla_modificada, fecha_mod) values (user, 'guarda', SYSDATE);

dbms_output.put_line('Se modifiko guarda');

```
end if;

end ;

/

ALTER TRIGGER "LUIS"."TRGUARDA" ENABLE;
```

```
-----

-- Archivo creado - martes-junio-09-2015

-----

-----
```

```
-- DDL for Table CABFACTURA

-----
```

```
CREATE TABLE "LUIS"."CABFACTURA"

(      "IDCABFACTURA" CHAR(4 BYTE),

        "IDCLIENTE" CHAR(4 BYTE),

        "IDVENDEDOR" CHAR(4 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

        PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE

        DEFAULT CELL_FLASH_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

REM INSERTING into LUIS.CABFACTURA

SET DEFINE OFF;

Insert into LUIS.CABFACTURA (IDCABFACTURA,IDCLIENTE,IDVENDEDOR) values ('1  ','1  ','1  ');
```

```
Insert into LUIS.CABFACTURA (IDCABFACTURA,IDCLIENTE,IDVENDEDOR) values ('2 ','2 ','2 ');
Insert into LUIS.CABFACTURA (IDCABFACTURA,IDCLIENTE,IDVENDEDOR) values ('3 ','3 ','3 ');
Insert into LUIS.CABFACTURA (IDCABFACTURA,IDCLIENTE,IDVENDEDOR) values ('4 ','1 ','4 ');
Insert into LUIS.CABFACTURA (IDCABFACTURA,IDCLIENTE,IDVENDEDOR) values ('5 ','2 ','4 ');
Insert into LUIS.CABFACTURA (IDCABFACTURA,IDCLIENTE,IDVENDEDOR) values ('6 ','3 ','3 ');
Insert into LUIS.CABFACTURA (IDCABFACTURA,IDCLIENTE,IDVENDEDOR) values ('7 ','3 ','1 ');
Insert into LUIS.CABFACTURA (IDCABFACTURA,IDCLIENTE,IDVENDEDOR) values ('8 ','3 ','1 ');
Insert into LUIS.CABFACTURA (IDCABFACTURA,IDCLIENTE,IDVENDEDOR) values ('9 ','2 ','1 ');
Insert into LUIS.CABFACTURA (IDCABFACTURA,IDCLIENTE,IDVENDEDOR) values ('10 ','2 ','3 ');
```

```
-----
-- DDL for Index PK_CABFACTURA
-----
```

```
CREATE UNIQUE INDEX "LUIS"."PK_CABFACTURA" ON "LUIS"."CABFACTURA"
("IDCABFACTURA")
```

```
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
```

```
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
```

```
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE
DEFAULT CELL_FLASH_CACHE DEFAULT)
```

```
TABLESPACE "SYSTEM" ;
```

```
-----
-- Constraints for Table CABFACTURA
-----
```

```
ALTER TABLE "LUIS"."CABFACTURA" ADD CONSTRAINT "PK_CABFACTURA" PRIMARY KEY
("IDCABFACTURA")
```

```
USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
```

```

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE
DEFAULT CELL_FLASH_CACHE DEFAULT)

TABLESPACE "SYSTEM" ENABLE;

ALTER TABLE "LUIS"."CABFACTURA" MODIFY ("IDVENDEDOR" NOT NULL ENABLE);

ALTER TABLE "LUIS"."CABFACTURA" MODIFY ("IDCLIENTE" NOT NULL ENABLE);

ALTER TABLE "LUIS"."CABFACTURA" MODIFY ("IDCABFACTURA" NOT NULL ENABLE);

-----

-- Ref Constraints for Table CABFACTURA

-----

ALTER TABLE "LUIS"."CABFACTURA" ADD CONSTRAINT "FK_CABFACTURA_CLIENTE" FOREIGN
KEY ("IDCLIENTE")

REFERENCES "LUIS"."CLIENTE" ("IDCLIENTE") ENABLE;

ALTER TABLE "LUIS"."CABFACTURA" ADD CONSTRAINT "FK_CABFACTURA_VENDEDOR"
FOREIGN KEY ("IDVENDEDOR")

REFERENCES "LUIS"."VENDEDOR" ("IDVENDEDOR") ENABLE;

-----

-- DDL for Trigger TRCABFACTURA

-----

CREATE OR REPLACE TRIGGER "LUIS"."TRCABFACTURA"

after insert or delete or update on cabfactura

begin

if inserting or deleting or updating then

insert into logs (usuario, tabla_modificada, fecha_mod) values (user, 'cabfactura', SYSDATE);

```



```
        dbms_output.put_line( 'Se modifico cabfactura' );  
    end if;  
end ;  
  
/  
  
ALTER TRIGGER "LUIS"."TRCABFACTURA" ENABLE;
```

```
-----  
-- Archivo creado - martes-junio-09-2015  
-----  
-----
```

```
-- DDL for Table DETALLEFACTURA  
-----
```

```
CREATE TABLE "LUIS"."DETALLEFACTURA"  
  
  (    "IDDETALLEFACT" CHAR(4 BYTE),  
    "IDCABFACTURA" CHAR(4 BYTE),  
    "IDPRODUCTO" CHAR(4 BYTE),  
    "CANTIDADPRODUCTO" NUMBER DEFAULT 0,  
    "FECHAVENTA" DATE  
  
  ) SEGMENT CREATION IMMEDIATE  
  
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING  
  
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645  
  
    PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE  
  DEFAULT CELL_FLASH_CACHE DEFAULT)  
  
  TABLESPACE "USERS" ;
```

REM INSERTING into LUIS.DETALLEFACTURA

SET DEFINE OFF;

Insert into LUIS.DETALLEFACTURA

(IDDETALLEFACT,IDCABFACTURA,IDPRODUCTO,CANTIDADPRODUCTO,FECHAVENTA) values ('1','1','1',2,to_date('01/01/13','DD/MM/RR'));

Insert into LUIS.DETALLEFACTURA

(IDDETALLEFACT,IDCABFACTURA,IDPRODUCTO,CANTIDADPRODUCTO,FECHAVENTA) values ('2','1','2',1,to_date('01/01/13','DD/MM/RR'));

Insert into LUIS.DETALLEFACTURA

(IDDETALLEFACT,IDCABFACTURA,IDPRODUCTO,CANTIDADPRODUCTO,FECHAVENTA) values ('3','1','3',1,to_date('01/01/13','DD/MM/RR'));

Insert into LUIS.DETALLEFACTURA

(IDDETALLEFACT,IDCABFACTURA,IDPRODUCTO,CANTIDADPRODUCTO,FECHAVENTA) values ('4','2','6',2,to_date('02/02/13','DD/MM/RR'));

Insert into LUIS.DETALLEFACTURA

(IDDETALLEFACT,IDCABFACTURA,IDPRODUCTO,CANTIDADPRODUCTO,FECHAVENTA) values ('5','2','7',1,to_date('02/02/13','DD/MM/RR'));

Insert into LUIS.DETALLEFACTURA

(IDDETALLEFACT,IDCABFACTURA,IDPRODUCTO,CANTIDADPRODUCTO,FECHAVENTA) values ('6','3','4',2,to_date('03/03/13','DD/MM/RR'));

Insert into LUIS.DETALLEFACTURA

(IDDETALLEFACT,IDCABFACTURA,IDPRODUCTO,CANTIDADPRODUCTO,FECHAVENTA) values ('7','3','8',2,to_date('03/03/13','DD/MM/RR'));

Insert into LUIS.DETALLEFACTURA

(IDDETALLEFACT,IDCABFACTURA,IDPRODUCTO,CANTIDADPRODUCTO,FECHAVENTA) values ('8','3','2',1,to_date('03/03/13','DD/MM/RR'));

Insert into LUIS.DETALLEFACTURA

(IDDETALLEFACT,IDCABFACTURA,IDPRODUCTO,CANTIDADPRODUCTO,FECHAVENTA) values ('9','4','1',2,to_date('05/05/13','DD/MM/RR'));

Insert into LUIS.DETALLEFACTURA

(IDDETALLEFACT,IDCABFACTURA,IDPRODUCTO,CANTIDADPRODUCTO,FECHAVENTA) values ('10','5','7',1,to_date('06/06/13','DD/MM/RR'));

Insert into LUIS.DETALLEFACTURA

(IDDETALLEFACT,IDCABFACTURA,IDPRODUCTO,CANTIDADPRODUCTO,FECHAVENTA) values ('11', '6', '3', '2', to_date('08/08/13', 'DD/MM/RR'));

Insert into LUIS.DETALLEFACTURA

(IDDETALLEFACT,IDCABFACTURA,IDPRODUCTO,CANTIDADPRODUCTO,FECHAVENTA) values ('12', '6', '2', '1', to_date('08/08/13', 'DD/MM/RR'));

Insert into LUIS.DETALLEFACTURA

(IDDETALLEFACT,IDCABFACTURA,IDPRODUCTO,CANTIDADPRODUCTO,FECHAVENTA) values ('13', '6', '8', '2', to_date('08/08/13', 'DD/MM/RR'));

Insert into LUIS.DETALLEFACTURA

(IDDETALLEFACT,IDCABFACTURA,IDPRODUCTO,CANTIDADPRODUCTO,FECHAVENTA) values ('14', '7', '5', '2', to_date('10/10/13', 'DD/MM/RR'));

Insert into LUIS.DETALLEFACTURA

(IDDETALLEFACT,IDCABFACTURA,IDPRODUCTO,CANTIDADPRODUCTO,FECHAVENTA) values ('15', '7', '6', '2', to_date('10/10/13', 'DD/MM/RR'));

Insert into LUIS.DETALLEFACTURA

(IDDETALLEFACT,IDCABFACTURA,IDPRODUCTO,CANTIDADPRODUCTO,FECHAVENTA) values ('16', '8', '9', '1', to_date('12/12/13', 'DD/MM/RR'));

Insert into LUIS.DETALLEFACTURA

(IDDETALLEFACT,IDCABFACTURA,IDPRODUCTO,CANTIDADPRODUCTO,FECHAVENTA) values ('17', '9', '9', '1', to_date('01/01/14', 'DD/MM/RR'));

Insert into LUIS.DETALLEFACTURA

(IDDETALLEFACT,IDCABFACTURA,IDPRODUCTO,CANTIDADPRODUCTO,FECHAVENTA) values ('18', '10', '1', '3', to_date('05/05/14', 'DD/MM/RR'));

Insert into LUIS.DETALLEFACTURA

(IDDETALLEFACT,IDCABFACTURA,IDPRODUCTO,CANTIDADPRODUCTO,FECHAVENTA) values ('19', '10', '2', '1', to_date('05/05/14', 'DD/MM/RR'));

Insert into LUIS.DETALLEFACTURA

(IDDETALLEFACT,IDCABFACTURA,IDPRODUCTO,CANTIDADPRODUCTO,FECHAVENTA) values ('20', '10', '3', '2', to_date('05/05/14', 'DD/MM/RR'));

-- DDL for Index PK_IDDETALLEFACT

CREATE UNIQUE INDEX "LUIS"."PK_IDDETALLEFACT" ON "LUIS"."DETALLEFACTURA"
("IDDETALLEFACT")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE
DEFAULT CELL_FLASH_CACHE DEFAULT)

TABLESPACE "USERS" ;

-- Constraints for Table DETALLEFACTURA

ALTER TABLE "LUIS"."DETALLEFACTURA" ADD CONSTRAINT "PK_IDDETALLEFACT" PRIMARY
KEY ("IDDETALLEFACT")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE
DEFAULT CELL_FLASH_CACHE DEFAULT)

TABLESPACE "USERS" ENABLE;

ALTER TABLE "LUIS"."DETALLEFACTURA" ADD CONSTRAINT "CK_CANTIDADPRODUCTO" CHECK
(cantidadproducto >= 0) ENABLE;

ALTER TABLE "LUIS"."DETALLEFACTURA" MODIFY ("IDDETALLEFACT" NOT NULL ENABLE);

ALTER TABLE "LUIS"."DETALLEFACTURA" MODIFY ("IDCABFACTURA" NOT NULL ENABLE);

ALTER TABLE "LUIS"."DETALLEFACTURA" MODIFY ("IDPRODUCTO" NOT NULL ENABLE);

ALTER TABLE "LUIS"."DETALLEFACTURA" MODIFY ("CANTIDADPRODUCTO" NOT NULL ENABLE);

ALTER TABLE "LUIS"."DETALLEFACTURA" MODIFY ("FECHAVENTA" NOT NULL ENABLE);

-- Ref Constraints for Table DETALLEFACTURA

```
ALTER TABLE "LUIS"."DETALLEFACTURA" ADD CONSTRAINT
"FK_DETALLEFACTURA_CABFACTURA" FOREIGN KEY ("IDCABFACTURA")
    REFERENCES "LUIS"."CABFACTURA" ("IDCABFACTURA") ENABLE;
```

```
ALTER TABLE "LUIS"."DETALLEFACTURA" ADD CONSTRAINT
"FK_DETALLEFACTURA_PRODUCTO" FOREIGN KEY ("IDPRODUCTO")
    REFERENCES "LUIS"."PRODUCTO" ("IDPRODUCTO") ENABLE;
```

-- DDL for Trigger TRDETALLEFACTURA

```
CREATE OR REPLACE TRIGGER "LUIS"."TRDETALLEFACTURA"
after insert or delete or update on detallefactura

begin

    if inserting or deleting or updating then

        insert into logs (usuario, tabla_modificada, fecha_mod) values (user, 'detallefactura',
SYSDATE);

        dbms_output.put_line( 'Se modifiko detallefactura' );

    end if;

end ;

/

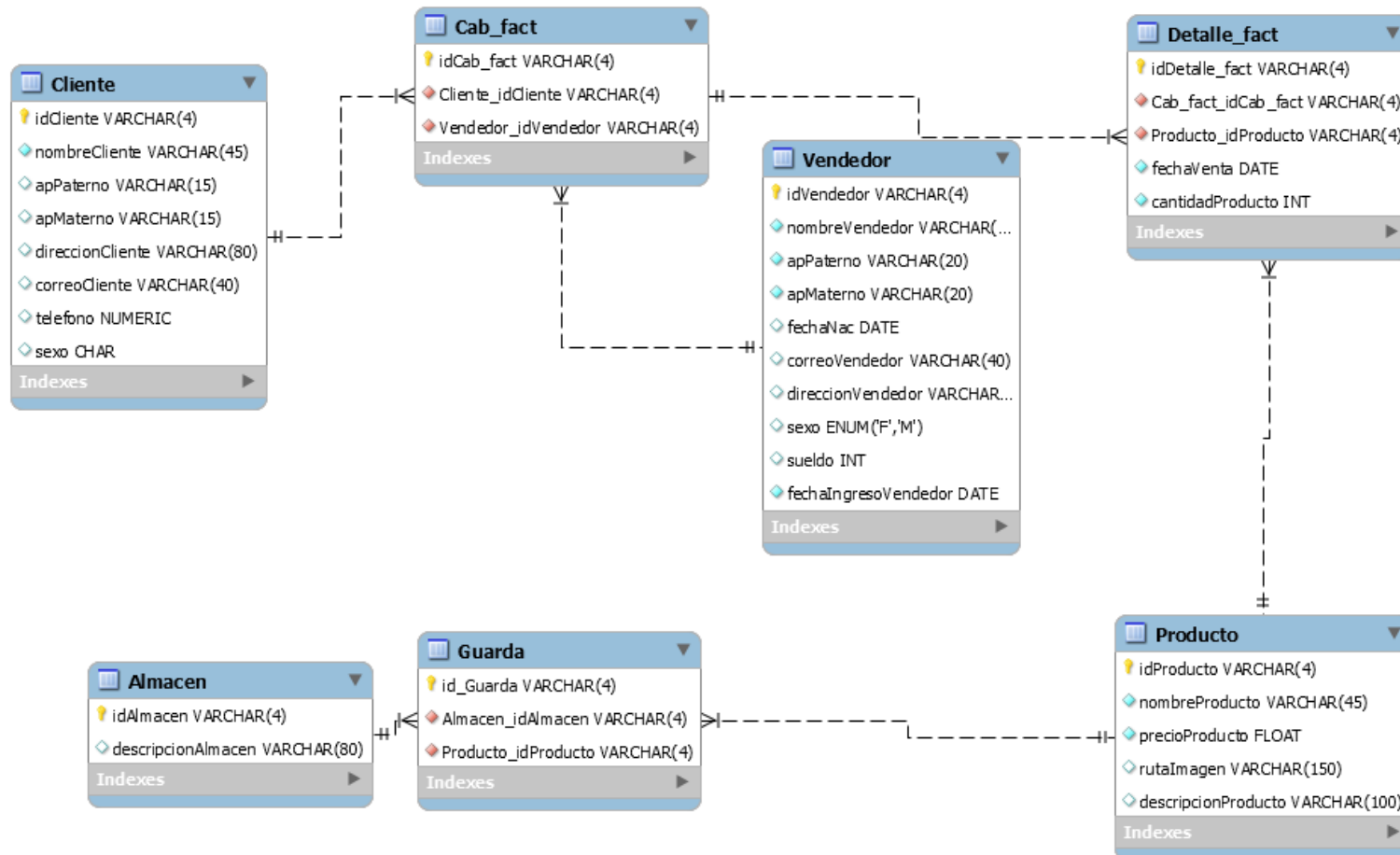
ALTER TRIGGER "LUIS"."TRDETALLEFACTURA" ENABLE
```

Diccionario de datos

Campo	Tamaño	Tipo	Descripción
idAlmacen	4	CHAR	Es la clave de identificacion del almacen
descripcionAlmacen	80	VARCHAR2	Pequeña descripcion acerca del almacen
idCabFactura	4	CHAR	Identificacion de la cabecera de factura
Cliente_idCliente	4	CHAR	Clave externa que representa el idCliente
Vendedor_idVendedor	4	CHAR	Clave externa de tabla Vendedor
fechaFactura	DD/MM/RR	DATE	Fecha en la cual se emitio la factura
idCliente	4	CHAR	Clave de identificacion de Cliente
nombreCliente	45	VARCHAR2	Muestra el nombre del cliente
apPaterno	15	VARCHAR2	Muestra el apellido paterno
apMaterno	15	VARCHAR2	Muestra el apellido materno
direccionCliente	80	VARCHAR2	Muestra la direccion del cliente
correoCliente	40	VARCHAR2	Muestra el correo electronico del cliente
telefonoCliente	10,0	NUMBER	Muestra telefono del cliente
sexoCliente	1	CHAR	Sexo del cliente
idVendedor	4	CHAR	Clave de identificacion del Vendedor
nomVendedor	20	VARCHAR2	Muestra el nombre del vendedor
apPaternoVendedor	15	VARCHAR2	Muestra el apellido paterno vendedor
apMaternoVendedor	15	VARCHAR2	Muestra el apellido materno vendedor
fechaNacVendedor	DD/MM/RR	DATE	Fecha de nacimiento vendedor
correoVendedor	40	VARCHAR2	E-mail del vendedor
direccionVendedor	80	VARCHAR2	Describe la direccion
sexoVendedor	1	VARCHAR2	Sexo del vendedor
fechaIngresoVendedor	DD/MM/RR	DATE	Fecha en la que entro el vendedor
sueldoVendedor	11	NUMBER	Muestra el sueldo del vendedor

idDetalleFact	4	CHAR	Identifiacion del detalle de factura
CabFactura_idCabFactura	4	CHAR	Clave externa cabecera factura
Producto_idProducto	4	CHAR	Clave externa del Producto
cantidadProducto	11	NUMBER	Muestra cuantos productos se vendieron
idGuarda	4	CHAR	Clave de identificacion de Guarda
Almacen_idAlmacen	4	CHAR	Clave externa de Almacen
Producto_idProducto	4	CHAR	Clave externa producto
idProducto	4	CHAR	Identificador de producto
nomProducto	15	VARCHAR2	Informa que producto es
precioProducto	126	FLOAT	Precio del producto con decimales
descripcionProducto	100	VARCHAR	Pequeña descripcion acerca del producto

Modelo Relacional



- **Nota:** Se incluye con éste archivo los .sql que crean las tablas

Constraints

Los constraints que se crearon auxilian a que no se repitan correos electrónicos, teléfonos, y que la identificación del sexo solo se limite a que sea **F** o **M**, y que los precios y las cantidades de productos sean iguales o mayores a cero, nunca menores, a continuación se mostraran los constraints utilizados.

```
/*DEFAULT*/
```

```
alter table cliente
```

```
modify (telefonocliente default 0);
```

```
alter table detallefactura
```

```
modify (cantidadproducto default 0);
```

```
/*unique*/
```

```
alter table cliente
```

```
add constraint u_correocliente
```

```
unique (correocliente);
```

```
alter table cliente
```

```
add constraint u_telefonocliente
```

```
unique (telefonocliente);
```

```
alter table vendedor
```

```
add constraint u_correovendedor
```

```
unique (correovendedor);
```

```
/*check*/
```

```
alter table cliente
```

```
add constraints ck_sexo
```

```
check (sexocliente = 'M' or sexocliente = 'F');
```

- **Nota.** Los constraints que se observan aquí, no necesitan ejecutarse, debido que con los scripts mencionados arriba de las tablas, estos mismos ya contienen los constraints cuando se crea la tabla de manera automática.

Vistas

- En la base de datos se incluyen tres vistas que son algunas de las consultas más importantes, se crean con los siguientes scripts:

```
-----  
-- Archivo creado - martes-junio-09-2015  
-----
```

```
-----  
-- DDL for View CLIENTES_MAS_COMPRAS  
-----
```

```
CREATE OR REPLACE FORCE VIEW "LUIS"."CLIENTES_MAS_COMPRAS" ("IDCLIENTE",  
"NUMERO_COMPRAS", "APPATERNO", "APMATERNO", "NOMBRECLIENTE") AS  
  select distinct cabfactura.IDCLIENTE, count(cabfactura.idcliente) as Numero_Compras,  
  cliente.APPATERNO,  
  cliente.APMATERNO, cliente.NOMBRECLIENTE  
from cliente, cabfactura  
where cabfactura.idcliente = cliente.idcliente  
group by cabfactura.IDCLIENTE, cliente.idcliente, cliente.APPATERNO,  
cliente.APMATERNO, cliente.NOMBRECLIENTE  
order by cabfactura.IDCLIENTE;
```

```
-----  
-- DDL for View PRODUCTOMASVENDIDOS  
-----
```

```
CREATE OR REPLACE FORCE VIEW "LUIS"."PRODUCTOMASVENDIDOS" ("IDPRODUCTO",  
"NOMPRODUCTO", "VENDIDOS") AS  
  SELECT Producto.idProducto, Producto.nomProducto,  
  SUM(Detallefactura.cantidadProducto) vendidos
```

```
FROM Cabfactura, Detallefactura, Producto
```

```
WHERE Cabfactura.idCabfactura = Detallefactura.idCabfactura  
      AND Detallefactura.idProducto = Producto.idProducto
```

```
GROUP BY Producto.idProducto, Producto.nomProducto
```

```
ORDER BY vendidos DESC;
```

```
-----  
-- DDL for View VENDEDORES_MAS_VENTAS  
-----
```

```
CREATE OR REPLACE FORCE VIEW "LUIS"."VENDEDORES_MAS_VENTAS"  
("IDVENDEDOR", "NUMERO_VENTAS", "APPATERNOVENDEDOR",  
"APMATERNOVENDEDOR", "NOMVENDEDOR") AS  
  select cabfactura.idvendedor, count(cabfactura.IDVENDEDOR) as Numero_ventas,  
vendedor.APPATERNOVENDEDOR,  
vendedor.APMATERNOVENDEDOR, VENDEDOR.NOMVENDEDOR  
from vendedor, cabfactura  
where cabfactura.idvendedor = vendedor.idvendedor  
group by cabfactura.idvendedor, vendedor.APPATERNOVENDEDOR,  
vendedor.APMATERNOVENDEDOR, VENDEDOR.NOMVENDEDOR  
order by cabfactura.idvendedor;
```

Triggers

- Los triggers que están presentes en la base de datos se encuentran en cada tabla que se creó previamente, estos triggers, cada vez que se modifica, elimina o inserta algo de las tablas de las bases de datos, se insertan dentro de la tabla logs, aquí se especifica que tabla se modificó, que usuario cambió y cuál fue la fecha en la que se realizó el cambio, los triggers pueden apreciarse a continuación.
- **Nota:** no es necesario ejecutar estos scripts, ya se encuentran incluidos en el script de la creación de las tablas

```
/*disparadores*/
```

```
create table logs (usuario varchar(40), tabla_Modificada varchar(40), fecha_mod date);
```

```
create or replace trigger tralmacen  
after insert or delete or update on almacen
```

```
begin  
  if inserting or deleting or updating then
```

```
insert into logs (usuario, tabla_modificada, fecha_mod) values (user, 'almacen',  
SYSDATE);  
dbms_output.put_line( 'Se modifiko almacen' );  
end if;  
end ;
```

create or replace trigger trcabfactura
after insert or delete or update on cabfactura

```
begin  
if inserting or deleting or updating then  
insert into logs (usuario, tabla_modificada, fecha_mod) values (user, 'cabfactura',  
SYSDATE);  
dbms_output.put_line( 'Se modifiko cabfactura' );  
end if;  
end ;
```

create or replace trigger trcliente
after insert or delete or update on cliente

```
begin  
if inserting or deleting or updating then  
insert into logs (usuario, tabla_modificada, fecha_mod) values (user, 'cliente',  
SYSDATE);  
dbms_output.put_line( 'Se modifiko cliente' );  
end if;  
end ;
```

create or replace trigger trdetallefactura
after insert or delete or update on detallefactura

```
begin  
if inserting or deleting or updating then  
insert into logs (usuario, tabla_modificada, fecha_mod) values (user, 'detallefactura',  
SYSDATE);  
dbms_output.put_line( 'Se modifiko detallefactura' );  
end if;  
end ;
```

create or replace trigger trguarda
after insert or delete or update on guarda

```

begin
  if inserting or deleting or updating then
    insert into logs (usuario, tabla_modificada, fecha_mod) values (user, 'guarda',
SYSDATE);
    dbms_output.put_line( 'Se modifiko guarda' );
  end if;
end ;

create or replace trigger trproducto
after insert or delete or update on producto

begin
  if inserting or deleting or updating then
    insert into logs (usuario, tabla_modificada, fecha_mod) values (user, 'producto',
SYSDATE);
    dbms_output.put_line( 'Se modifiko producto' );
  end if;
end ;

create or replace trigger trvendedor
after insert or delete or update on vendedor

begin
  if inserting or deleting or updating then
    insert into logs (usuario, tabla_modificada, fecha_mod) values (user, 'vendedor',
SYSDATE);
    dbms_output.put_line( 'Se modifiko vendedor' );
  end if;
end ;

```

Procedures

- El esquema contiene procedimientos para crear deletes e inserts, para cada tabla de la base de datos, excepto para la tabla logs, existen procedimientos que sirven para realizar updates, solo que éstos solo se limitan para todos los campos que no sean llaves primarias, se agregaron procedimientos que usan cursores y muestran las vistas de las consultas más importantes, las podemos encontrar en la siguiente lista:

/*Procedimientos*/

set serveroutput on;

commit;

create or replace procedure prModificacion(s char)

is

BEGIN

dbms_output.put_line('Se agrego modificaciones a la tabla log, modificado en: ' || s);

END;

set serveroutput on;

commit;

/*insertar*/

create or replace procedure almacen_ins(a CHAR, b varchar2)

is

BEGIN

insert into almacen (idalmacen, descripcionalmacen) values (a, b);

dbms_output.put_line ('Se inserto correctamente en almacen');

END;

```
create or replace procedure cabfactura_ins (a char, b char, c char,  
d varchar2)
```

```
is
```

```
BEGIN
```

```
insert into cabfactura (idcabfactura, idcliente, idvendedor, fechafactura)  
values (a, b, c, d);
```

```
dbms_output.put_line ('Se inserto correctamente en cabfactura');
```

```
END;
```

```
create or replace procedure cliente_ins (a char, b varchar2, c varchar2,  
d varchar2, e varchar2, f varchar2, g number, h char)
```

```
is
```

```
BEGIN
```

```
insert into cliente (idcliente, nombrecliente, appaterno,  
apmaterno, direccioncliente, correocliente, telefonocliente, sexocliente)  
values
```

```
(a, b, c, d, e, f, g, h);
```

```
dbms_output.put_line ('Se inserto correctamente en cliente');
```


END;

create or replace procedure detallefactura_ins (a char, b char, c char, d number)

is

e date := SYSDATE;

BEGIN

**insert into detallefactura (iddetallefact, idcabfactura, idproducto, cantidadproducto,
fechaventa) values (a, b, c, d, e);**

dbms_output.put_line ('Se inserto correctamente en detallefactura');

END;

create or replace procedure guarda_ins (a char, b char, c char)

is

BEGIN

insert into guarda (idguarda, idproducto, idalmacen) values (a, b, c);

```
dbms_output.put_line ('Se inserto correctamente en guarda');
```

```
end;
```

```
create or replace procedure producto_ins(a char, b varchar2, c float, d varchar2)
```

```
is
```

```
BEGIN
```

```
insert into producto (idproducto, nomproducto, precioproducto, descripcionproducto)
```

```
values (a, b, c, d);
```

```
dbms_output.put_line ('Se inserto correctamente en producto');
```

```
end;
```

```
create or replace procedure vendedor_ins (a char, b varchar2, c varchar2, d varchar2 , e  
varchar2,
```

```
f varchar2, g varchar2, h varchar2, j number)
```

```
is
```

```
e1 date := TO_DATE (e || ' 00:00:00', 'dd/mm/yyyy hh24:mi:ss' );
```

```
i date := SYSDATE;
```

```
BEGIN
```

```
insert into vendedor (idvendedor, nomvendedor, appaternovendedor, apmaternovendedor,  
fechanacvendedor, correoendedor, direccionvendedor, sexovendedor,  
fechaingresovendedor, sueldoendedor) values  
(a, b, c, d, e1, f, g, h, i, j);
```

```
dbms_output.put_line ('Se inserto correctamente en vendedor');
```

```
end;
```

```
/*ejemplo en vendedor*/
```

```
execute vendedor_ins('4', 'JR', 'M', 'S', '22/01/1993', 'jrmsxx@hotmail.com', 'n/a', 'M', 5000);
```

```
/*Eliminar*/
```

```
create or replace procedure prEliminarAlmacen(idc char)
```

```
is
```

```
Begin
```

```
delete from almacen where idalmacen = idc;
```

```
dbms_output.put_line ('Se elimino correctamente el registro');
```

```
end;
```

```
create or replace procedure prEliminarCabFact(idc char)
```

```
is
```

Begin

delete from cabfactura where idcabfactura = idc;

dbms_output.put_line ('Se elimino correctamente el registro');

end;

create or replace procedure prEliminarDetFact(idc char)

is

Begin

delete from detallefactura where iddetallefact = idc;

dbms_output.put_line ('Se elimino correctamente el registro');

end;

create or replace procedure prEliminarGuarda(idc char)

is

Begin

delete from guarda where idguarda = idc;

dbms_output.put_line ('Se elimino correctamente el registro');

end;

create or replace procedure prEliminarProducto(idc char)

is

Begin

delete from producto where idproducto = idc;

```
dbms_output.put_line ('Se elimino correctamente el registro');
```

```
end;
```

```
create or replace procedure prEliminarVendedor(idc char)
```

```
is
```

```
Begin
```

```
delete from vendedor where idvendedor = idc;
```

```
dbms_output.put_line ('Se elimino correctamente el registro');
```

```
end;
```

```
/*update*/
```

```
create or replace procedure prActualizardescalmacen(idc char, val varchar2)
```

```
is
```

```
Begin
```

```
update almacen
```

```
set descripcionalmacen = val
```

```
where idalmacen = idc;
```

```
dbms_output.put_line ('Se actualizo correctamente el registro');
```

```
end;
```

```
create or replace procedure prActualizaridcliente(idc char, val char)
```

is

Begin

update cabfactura

set cabfactura.IDCLIENTE = val

where idcabfactura = idc;

dbms_output.put_line ('Se actualizo correctamente el registro');

end;

create or replace procedure prActualizaridvendedor(idc char, val char)

is

Begin

update cabfactura

set idvendedor = val

where idcabfactura = idc;

dbms_output.put_line ('Se actualizo correctamente el registro');

end;

create or replace procedure prActualizarFechaFactura(idc char, val varchar2)

is

Begin

update cabfactura

set fechafactura = TO_DATE (val || ' 00:00:00', 'dd/mm/yyyy hh24:mi:ss')

where idcabfactura = idc;

dbms_output.put_line ('Se actualizo correctamente el registro');

end;

```
create or replace procedure prActualizarNombreCliente(idc char, val varchar2)
is
Begin
update cliente
set nombreciente = val
where idcliente = idc;
dbms_output.put_line ('Se actualizo correctamente el registro');
end;
```

```
create or replace procedure prActualizarappaterno(idc char, val varchar2)
is
Begin
update cliente
set appaterno = val
where idcliente = idc;
dbms_output.put_line ('Se actualizo correctamente el registro');
end;
```

```
create or replace procedure prActualizarapmaterno(idc char, val varchar2)
is
Begin
update cliente
set apmaterno = val
```

where idcliente = idc;

dbms_output.put_line ('Se actualizo correctamente el registro');

end;

create or replace procedure prActualizardireccioncliente(idc char, val varchar2)

is

Begin

update cliente

set direccioncliente = val

where idcliente = idc;

dbms_output.put_line ('Se actualizo correctamente el registro');

end;

create or replace procedure prActualizarCorreoCliente(idc char, val varchar2)

is

Begin

update cliente

set correocliente = val

where idcliente = idc;

dbms_output.put_line ('Se actualizo correctamente el registro');

end;

create or replace procedure prActualizartelefonocliente(idc char, val number)

is

Begin

update cliente


```
set telefonocliente = val
```

```
where idcliente = idc;
```

```
dbms_output.put_line ('Se actualizo correctamente el registro');
```

```
end;
```

```
create or replace procedure prActualizarSexocliente(idc char, val char)
```

```
is
```

```
Begin
```

```
update cliente
```

```
set sexocliente = val
```

```
where idcliente = idc;
```

```
dbms_output.put_line ('Se actualizo correctamente el registro');
```

```
end;
```

```
create or replace procedure prActualizaridcabfactdetfact(idc char, val char)
```

```
is
```

```
Begin
```

```
update detallefactura
```

```
set idcabfactura = val
```

```
where iddetallefact = idc;
```

```
dbms_output.put_line ('Se actualizo correctamente el registro');
```

end;

create or replace procedure prActualizarProductoDF(idc char, val char)

is

Begin

update detallefactura

set idproducto = val

where iddetallefact = idc;

dbms_output.put_line ('Se actualizo correctamente el registro');

end;

create or replace procedure prActualizarcantidadprod(idc char, val char)

is

Begin

update detallefactura

set CANTIDADPRODUCTO = val

where iddetallefact = idc;

dbms_output.put_line ('Se actualizo correctamente el registro');

end;

create or replace procedure prActualizarFechaFactura(idc char, val varchar2)

is

Begin

update detallefactura

set fechaventa = TO_DATE (val || ' 00:00:00', 'dd/mm/yyyy hh24:mi:ss')

where iddetallefact = idc;

```
dbms_output.put_line ('Se actualizo correctamente el registro');  
end;
```

```
create or replace procedure prActualizaridproductog(idc char, val char)  
is  
Begin  
update guarda  
set idproducto = val  
where idguarda = idc;  
dbms_output.put_line ('Se actualizo correctamente el registro');  
end;
```

```
create or replace procedure prActualizaridalmaceng(idc char, val char)  
is  
Begin  
update guarda  
set idalmacen = val  
where idguarda = idc;  
dbms_output.put_line ('Se actualizo correctamente el registro');  
end;
```

```
create or replace procedure prActualizararnomproducto(idc char, val varchar2)  
is  
Begin  
update producto
```

```
set nomproducto = val
```

```
where idproducto = idc;
```

```
dbms_output.put_line ('Se actualizo correctamente el registro');
```

```
end;
```

```
create or replace procedure prActualizarprecioproducto(idc char, val float)
```

```
is
```

```
Begin
```

```
update producto
```

```
set precioproducto = val
```

```
where idproducto = idc;
```

```
dbms_output.put_line ('Se actualizo correctamente el registro');
```

```
end;
```

```
create or replace procedure prActualizardescripproducto(idc char, val varchar2)
```

```
is
```

```
Begin
```

```
update producto
```

```
set descripcionproducto = val
```

```
where idproducto = idc;
```

```
dbms_output.put_line ('Se actualizo correctamente el registro');
```

```
end;
```

```
describe vendedor
```

```
create or replace procedure prActualizarnomvendedor(idc char, val varchar2)
```

is

Begin

update vendedor

set nomvendedor = val

where idvendedor = idc;

dbms_output.put_line ('Se actualizo correctamente el registro');

end;

create or replace procedure prActualizarAPvendedor(idc char, val varchar2)

is

Begin

update vendedor

set APPATERNOVENDEDOR = val

where idvendedor = idc;

dbms_output.put_line ('Se actualizo correctamente el registro');

end;

create or replace procedure prActualizarAMvendedor(idc char, val varchar2)

is

Begin

update vendedor

set apmaternovendedor = val

where idvendedor = idc;

dbms_output.put_line ('Se actualizo correctamente el registro');

end;

```
create or replace procedure prActualizarFNvendedor(idc char, val varchar2)
is
Begin
update vendedor
set fechanacvendedor = TO_DATE (val || ' 00:00:00', 'dd/mm/yyyy hh24:mi:ss' )
where idvendedor = idc;
dbms_output.put_line ('Se actualizo correctamente el registro');
end;
```

```
create or replace procedure prActualizarcorreovendedor(idc char, val varchar2)
is
Begin
update vendedor
set correovendedor = val
where idvendedor = idc;
dbms_output.put_line ('Se actualizo correctamente el registro');
end;
```

```
create or replace procedure prActualizardirvendedor(idc char, val varchar2)
is
Begin
update vendedor
set direccionvendedor = val
where idvendedor = idc;
dbms_output.put_line ('Se actualizo correctamente el registro');
```

end;

create or replace procedure prActualizarsexovendedor(idc char, val varchar2)

is

Begin

update vendedor

set sexovendedor = val

where idvendedor = idc;

dbms_output.put_line ('Se actualizo correctamente el registro');

end;

create or replace procedure prActualizarFlvendedor(idc char, val varchar2)

is

Begin

update vendedor

set fechaingreso = TO_DATE (val || ' 00:00:00', 'dd/mm/yyyy hh24:mi:ss')

where idvendedor = idc;

dbms_output.put_line ('Se actualizo correctamente el registro');

end;

create or replace procedure prActualizarsueldovendedor(idc char, val number)

is

Begin

update vendedor

set sueldovendedor = val

where idvendedor = idc;

```
dbms_output.put_line ('Se actualizo correctamente el registro');
```

```
end
```

- **Nota:** Se debe de ejecutar cada script para poder usar estos procedures dentro del esquema, junto con éste archivo en la carpeta docs se encuentran los archivos procedures.sql que contienen los procedimientos descritos arriba