

Lenguajes de Bases de Datos Relacionales-SC-504

Estudiantes:

Gamboa Rodríguez Yoswell, Monge Gamboa Nicole,
Madrigal Miranda Wimmer, Serrano Barquero Dennis

Documento final de proyecto.

Profesor:

Clarence Ricketts Torres.

III CUATRIMESTRE

Diciembre 2022.

Script del proyecto

Creación de las tablas con sus respectivos inserts y alter table

TABLA DE CLIENTES

```
CREATE TABLE clientes (  
    id          NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,  
    nombre      VARCHAR2(30) NOT NULL,  
    primer_apellido VARCHAR2(50) NOT NULL,  
    segundo_apellido VARCHAR2(50) NOT NULL,  
    correo      VARCHAR2(100) NOT NULL,  
    telefono    NUMBER NOT NULL  
);  
  
ALTER TABLE clientes ADD CONSTRAINT clientes_pk PRIMARY KEY ( id );
```

Inserts de la tabla

```
insert into clientes(id,nombre,primer_apellido, segundo_apellido,  
correo, telefono) values (1,'David', 'Montero',  
'Gomez','david@gmail.com',89563663);  
insert into clientes(id,nombre,primer_apellido, segundo_apellido,  
correo, telefono) values (2,'Tommy', 'Hernandez',  
'Soto','tommy@gmail.com',98895463);  
insert into clientes(id,nombre,primer_apellido, segundo_apellido,  
correo, telefono) values (3,'Jerry', 'Sojo',  
'Morales','jerry@gmail.com',41526398);  
insert into clientes(id,nombre,primer_apellido, segundo_apellido,  
correo, telefono) values (4,'Santi', 'Gonzalez',  
'Sanabria','santi@gmail.com',7485912);  
insert into clientes(id,nombre,primer_apellido, segundo_apellido,  
correo, telefono) values (5,'Josefina', 'Fernandez',  
'Monge','josefi@gmail.com',63652541);
```

TABLA DE PRODUCTOS NUEVOS

```
CREATE TABLE productos_nuevos (  
    id          NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,  
    codigo      VARCHAR2(50) NOT NULL,  
    nombre      VARCHAR2(50) NOT NULL,  
    cantidad_stock NUMBER NOT NULL,  
    accesorios  VARCHAR2(50) NOT NULL,  
    estado      VARCHAR2(50),  
    precio      FLOAT NOT NULL  
);  
  
ALTER TABLE productos_nuevos ADD CONSTRAINT productos_usadosv1_pk  
PRIMARY KEY ( id );
```

Script del proyecto

Inserts de la tabla

```
insert into productos_nuevos(id, codigo, nombre, cantidad_stock,
accesorios, estado, precio) values (1,'A56D','Control de
XBox',25,'Control para el XBox', 'Nuevo', 15000);
insert into productos_nuevos(id, codigo, nombre, cantidad_stock,
accesorios, estado, precio) values (2,'F53C','Nintendo Wii', 20,
'Cable de video', 'Nuevo', 150000);
insert into productos_nuevos(id, codigo, nombre, cantidad_stock,
accesorios, estado, precio) values (3,'L09P','Play Station 5', 15,
'Mando de la play5', 'Nuevo', 25000);
insert into productos_nuevos(id, codigo, nombre, cantidad_stock,
accesorios, estado, precio) values (4,'G23R','Play Station 4', 50,
'Mando de la Play4', 'Nuevo', 15000);
insert into productos_nuevos(id, codigo, nombre, cantidad_stock,
accesorios, estado, precio) values (5,'N11M','Monitor Dell',
50,'Panalla Dell', 'Nuevo', 69000);
```

TABLA DE PRODUCTOS USADOS

```
CREATE TABLE productos_usados (
  id          NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
  codigo      VARCHAR2(50) NOT NULL,
  nombre      VARCHAR2(50) NOT NULL,
  cantidad_stock NUMBER NOT NULL,
  accesorios  VARCHAR2(50) NOT NULL,
  estado      VARCHAR2(50),
  precio      FLOAT NOT NULL
);

ALTER TABLE productos_usados ADD CONSTRAINT productos_usados_pk
PRIMARY KEY ( id );
```

Inserts de la tabla

```
insert into from productos_usados(id, codigo, nombre,
cantidad_stock, accesorios, estado, precio) values
(6,'H56Y','Consola Play Station 5', 2,'Control', 'Usado sin ningun
problema', 15000);
insert into productos_usados(id, codigo, nombre, cantidad_stock,
accesorios, estado, precio) values (7,'W23A','Teclado pc de USB',
1,'Teclado', 'Con un pequeño golpe en la caratula',5000);
insert into productos_usados(id, codigo, nombre, cantidad_stock,
accesorios, estado, precio) values (8,'D45S','Cable de video',
5,'Cable', 'Cable sin ningun daño', 5000);
insert into productos_usados(id, codigo, nombre, cantidad_stock,
accesorios, estado, precio) values (9,'Y76G','Control de consola
XBox', 2, 'Control','Sin ningun inconveniente', 15000);
insert into productos_usados(id, codigo, nombre, cantidad_stock,
accesorios, estado, precio) values (10,'J78B','Monitor HP', 2,
'Monitor y cable de video', 'Monitor con su cable de video en
```

Script del proyecto

```
excelente estado', 25000);
```

TABLA DE REPUESTOS NUEVOS

```
CREATE TABLE repuestos_nuevos (  
    id          NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,  
    codigo      VARCHAR2(50) NOT NULL,  
    nombre      VARCHAR2(50) NOT NULL,  
    cantidad_stock NUMBER NOT NULL,  
    accesorios  VARCHAR2(50) NOT NULL,  
    estado      VARCHAR2(50),  
    precio      FLOAT NOT NULL  
);  
  
ALTER TABLE repuestos_nuevos ADD CONSTRAINT repuestos_usadosv1_pk  
PRIMARY KEY ( id );
```

Inserts de la tabla

```
insert into repuestos_nuevos(id, codigo, nombre, cantidad_stock,  
accesorios, estado, precio) values (11,'G23R','Play Station 4', 50,  
'Mando de la Play4', 'Nuevo', 15000);  
insert into repuestos_nuevos(id, codigo, nombre, cantidad_stock,  
accesorios, estado, precio) values (12,'F53C','Nintendo Wii', 20,  
'Cable de video', 'Nuevo', 150000);  
insert into repuestos_nuevos(id, codigo, nombre, cantidad_stock,  
accesorios, estado, precio) values (13,'A56D','Control de  
XBox',25,'Control para el XBox', 'Nuevo', 15000);  
insert into repuestos_nuevos(id, codigo, nombre, cantidad_stock,  
accesorios, estado, precio) values (14,'ZPF','Tablet Alcatel', 13,  
'Tableta con sus respectivos cables', 'Nuevo', 79000);  
insert into repuestos_nuevos(id, codigo, nombre, cantidad_stock,  
accesorios, estado, precio) values (15,'N11M','Monitor Dell',  
50,'Panalla Dell', 'Nuevo', 69000);
```

TABLA DE REPUESTOS USADOS

```
CREATE TABLE repuestos_usados (  
    id          NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,  
    codigo      VARCHAR2(50) NOT NULL,  
    nombre      VARCHAR2(50) NOT NULL,  
    cantidad_stock NUMBER NOT NULL,  
    accesorios  VARCHAR2(50) NOT NULL,  
    estado      VARCHAR2(50),  
    precio      FLOAT NOT NULL  
);  
  
ALTER TABLE repuestos_usados ADD CONSTRAINT productos_usadosv1_pkv1  
PRIMARY KEY ( id );
```

Script del proyecto

Inserts de la tabla

```
insert into repuestos_usados(id, codigo, nombre, cantidad_stock,
accesorios, estado, precio) values (17,'F53C','Nintendo Wii', 20,
'Cable de video', 'Cable en perfecto estado', 150000);
insert into repuestos_usados(id, codigo, nombre, cantidad_stock,
accesorios, estado, precio) values (18,'W23A','Teclado pc de USB',
1,'Teclado', 'Con un pequeño golpe en la caratula',5000);
insert into repuestos_usados(id, codigo, nombre, cantidad_stock,
accesorios, estado, precio) values (19,'N11M','Monitor Dell',
50,'Panalla Dell', 'Pantalla sin ningun inconveniente', 69000);
insert into repuestos_usados(id, codigo, nombre, cantidad_stock,
accesorios, estado, precio) values (20,'Y76G','Control de consola
XBox', 2, 'Control', 'Sin ningun inconveniente', 15000);
insert into repuestos_usados(id, codigo, nombre, cantidad_stock,
accesorios, estado, precio) values (8,'D45S','Cable de video',
5,'Cable', 'Cable sin ningun daño', 5000);
```

TABLA DE REPORTE TÉCNICO

```
CREATE TABLE reporte_tecnico (
    id          NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
    problema_diagnostico VARCHAR2(100) NOT NULL,
    descripcion    VARCHAR2(100) NOT NULL,
    costo          FLOAT NOT NULL
);

ALTER TABLE  reporte_tecnico  ADD CONSTRAINT  reporte_tecnico_pk
PRIMARY KEY ( id );
```

Inserts de la tabla

```
insert into reporte_tecnico(id, problema_diagnostico, descripcion,
costo) values (22,'Sin funcionamiento en el monitor, no hay video',
'Cable de video sin reflejo',15000);
insert into reporte_tecnico(id, problema_diagnostico, descripcion,
costo) values (23,'Control de consola Play Station 5 sin respuesta
al dispositivo','Control de la consola Play Station 5 sin fucion de
los botones', 25000);
insert into reporte_tecnico(id, problema_diagnostico, descripcion,
costo) values (24,'Tarjeta madre sin funcionamiento', 'CPU no
corre', 35000);
insert into reporte_tecnico(id, problema_diagnostico, descripcion,
costo) values (25,'Bateeria de la consola sin funcion', 'Nnintedo
Switch sin respuesta luego de encender', 25000);
insert into reporte_tecnico(id, problema_diagnostico, descripcion,
costo) values (26,'Portatil HP sin respuesta al encender por
problemas en la bateria', 'Portatil HP no enciende', 45000);
```

Script del proyecto

TABLA DE USUARIOS

```
CREATE TABLE usuarios (  
    id          NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,  
    usuario     VARCHAR2(30) NOT NULL,  
    contrasena  VARCHAR2(30) NOT NULL  
);  
  
ALTER TABLE usuarios ADD CONSTRAINT usuarios_pk PRIMARY KEY ( id );
```

Inserts de la tabla

```
insert into usuarios(id, usuario, contrasena) values  
(1,'monsty12','123');  
insert into usuarios(id, usuario, contrasena) values  
(2,'morali65','3215');  
insert into usuarios(id, usuario, contrasena) values  
(3,'monchi11','456');
```

TABLA DE ESTADO

```
CREATE TABLE estado (  
    id          NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,  
    clientes_id NUMBER NOT NULL,  
    descripcion VARCHAR2(100) NOT NULL  
);  
  
ALTER TABLE estado ADD CONSTRAINT estado_pk PRIMARY KEY ( id );  
  
ALTER TABLE estado  
    ADD CONSTRAINT estado_clientes_fk FOREIGN KEY ( clientes_id )  
        REFERENCES clientes ( id );
```

Inserts de la tabla

```
insert into estado(id,clientes_id,descripcion) values (0,1,'Cable de  
video sin reflejo');  
insert into estado(id,clientes_id,descripcion) values (1,2,'Control  
de la consola Play Station 5 sin fucion de los botones');  
insert into estado(id,clientes_id,descripcion) values (2,3,'CPU no  
corre');  
insert into estado(id,clientes_id,descripcion) values (3,4,'Nnintedo  
Switch sin respuesta luego de encender');
```

Script del proyecto

```
insert into estado(id,clientes_id,descripcion) values (4,5,'Portatil  
HP no enciende');
```

TABLA DE TIENDA

```
CREATE TABLE tienda (  
    id          NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,  
    taller_id   NUMBER NOT NULL,  
    inventario_id NUMBER NOT NULL,  
    clientes_id NUMBER NOT NULL  
);  
  
ALTER TABLE tienda ADD CONSTRAINT tienda_pk PRIMARY KEY ( id );  
  
ALTER TABLE tienda  
    ADD CONSTRAINT tienda_taller_fk FOREIGN KEY ( taller_id )  
        REFERENCES taller ( id );  
  
ALTER TABLE tienda  
    ADD CONSTRAINT tienda_inventario_fk FOREIGN KEY ( inventario_id  
    )  
        REFERENCES inventario ( id );  
  
ALTER TABLE tienda  
    ADD CONSTRAINT tienda_clientes_fk FOREIGN KEY ( clientes_id )  
        REFERENCES clientes ( id );
```

Inserts de la tabla

```
insert into tienda(id, taller_id, inventario_id, clientes_id) values  
(5,15,20,1);  
insert into tienda(id, taller_id, inventario_id, clientes_id) values  
(6,16,21,2);  
insert into tienda(id, taller_id, inventario_id, clientes_id) values  
(7,17,22,3);  
insert into tienda(id, taller_id, inventario_id, clientes_id) values  
(8,18,23,4);  
insert into tienda(id, taller_id, inventario_id, clientes_id) values  
(9,19,24,5);
```

TABLA DE TALLER

```
CREATE TABLE taller (  
    id          NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
```

Script del proyecto

```
        equipos_id NUMBER NOT NULL,  
        clientes_id NUMBER NOT NULL  
    );  
  
    ALTER TABLE taller ADD CONSTRAINT taller_pk PRIMARY KEY ( id );  
  
    ALTER TABLE taller  
        ADD CONSTRAINT taller_clientes_fk FOREIGN KEY ( clientes_id )  
            REFERENCES clientes ( id );  
  
    ALTER TABLE taller  
        ADD CONSTRAINT taller_equipos_fk FOREIGN KEY ( equipos_id )  
            REFERENCES equipos ( id );
```

Inserts de la tabla

```
insert into taller(id, equipos_id, clientes_id) values (100,85,1);  
insert into taller(id, equipos_id, clientes_id) values (101,86,2);  
insert into taller(id, equipos_id, clientes_id) values (102,87,3);  
insert into taller(id, equipos_id, clientes_id) values (103,88,4);  
insert into taller(id, equipos_id, clientes_id) values (104,89,5);
```

TABLA DE INVENTARIO

```
CREATE TABLE inventario (  
    id          NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,  
    productos_nuevos_id NUMBER NOT NULL,  
    productos_usados_id NUMBER NOT NULL,  
    repuestos_usados_id NUMBER NOT NULL,  
    repuestos_nuevos_id NUMBER NOT NULL  
);  
  
ALTER TABLE inventario ADD CONSTRAINT inventario_pk PRIMARY KEY ( id );  
  
ALTER TABLE inventario  
    ADD CONSTRAINT inventario_productos_nuevos_fk FOREIGN KEY (  
        productos_nuevos_id )  
        REFERENCES productos_nuevos ( id );  
  
ALTER TABLE inventario  
    ADD CONSTRAINT inventario_productos_usados_fk FOREIGN KEY (  
        productos_usados_id )  
        REFERENCES productos_usados ( id );  
  
ALTER TABLE inventario  
    ADD CONSTRAINT inventario_repuestos_nuevos_fk FOREIGN KEY (  
        repuestos_nuevos_id )  
        REFERENCES repuestos_nuevos ( id );
```


Script del proyecto

```
ALTER TABLE inventario
    ADD CONSTRAINT inventario_repuestos_usados_fk FOREIGN KEY (
repuestos_usados_id )
    REFERENCES repuestos_usados ( id );
```

Inserts de la tabla

```
insert into inventario(id, productos_nuevos_id, productos_usados_id,
repuestos_usados_id, repuestos_nuevos_id) values (96,1,6,11,17);
insert into inventario(id, productos_nuevos_id, productos_usados_id,
repuestos_usados_id, repuestos_nuevos_id) values (97,2,7,12,18);
insert into inventario(id, productos_nuevos_id, productos_usados_id,
repuestos_usados_id, repuestos_nuevos_id) values (98,3,8,13,19);
insert into inventario(id, productos_nuevos_id, productos_usados_id,
repuestos_usados_id, repuestos_nuevos_id) values (99,4,9,14,20);
insert into inventario(id, productos_nuevos_id, productos_usados_id,
repuestos_usados_id, repuestos_nuevos_id) values (100,5,10,15,21);
```

TABLA DE FACTURA

```
CREATE TABLE factura (
    id          NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
    clientes_id NUMBER NOT NULL,
    descripcion VARCHAR2(100) NOT NULL,
    costo       FLOAT NOT NULL,
    iva         FLOAT NOT NULL,
    total       FLOAT NOT NULL
);
```

```
ALTER TABLE factura ADD CONSTRAINT factura_pk PRIMARY KEY ( id );
```

```
ALTER TABLE factura
    ADD CONSTRAINT factura_clientes_fk FOREIGN KEY ( clientes_id )
    REFERENCES clientes ( id );
```

Inserts de la tabla

```
insert into factura(id, clientes_id, descripcion, costo, iva, total)
values (150,0,'Cable de video sin reflejo', 15000, 13, 16000);
insert into factura(id, clientes_id, descripcion, costo, iva, total)
values (151,1,'Control de la consola Play Station 5 sin fucion de
los botones', 25000, 13, 26000);
insert into factura(id, clientes_id, descripcion, costo, iva, total)
values (152,2,'CPU no corre', 35000, 13, 36000);
insert into factura(id, clientes_id, descripcion, costo, iva, total)
values (153,3,'Nnintedo Switch sin respuesta luego de encender',
25000, 13, 26000);
insert into factura(id, clientes_id, descripcion, costo, iva, total)
values (154,4,'Portatil HP no enciende', 45000, 13, 46000);
```

Script del proyecto

TABLA DE EQUIPOS

```
CREATE TABLE equipos (  
    id          NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,  
    reporte_tecnico_id  NUMBER NOT NULL,  
    estado_id      NUMBER NOT NULL,  
    numero_serie     VARCHAR2(25) NOT NULL,  
    modelo          VARCHAR2(25) NOT NULL,  
    descripcion_problema VARCHAR2(100) NOT NULL,  
    accesorios       VARCHAR2(50) NOT NULL  
);  
  
ALTER TABLE equipos ADD CONSTRAINT equipos_pk PRIMARY KEY ( id );  
  
ALTER TABLE equipos  
    ADD CONSTRAINT equipos_estado_fk FOREIGN KEY ( estado_id )  
        REFERENCES estado ( id );  
  
ALTER TABLE equipos  
    ADD CONSTRAINT equipos_reporte_tecnico_fk FOREIGN KEY (   
reporte_tecnico_id )  
        REFERENCES reporte_tecnico ( id );
```

Inserts de la tabla

```
insert into equipos(id,reporte_tecnico_id, estado_id, numero_serie,  
modelo,          descripcion_problema,          accesorios)          values  
(85,22,0,'56Nf','Modelo 500','Sin funcionamiento de a tarjeta  
madre','Tarjeta');  
insert into equipos(id,reporte_tecnico_id, estado_id, numero_serie,  
modelo,          descripcion_problema,          accesorios)          values  
(86,23,1,'57hf','Modelo 600','Control de consola Xbox sin respuesta  
al dispositivo','Control de consola Xbox');  
insert into equipos(id,reporte_tecnico_id, estado_id, numero_serie,  
modelo,          descripcion_problema,          accesorios)          values  
(87,24,2,'23Ni','Modelo 700','Luces del Nintendo DSi sin  
funcionamiento','Caratula y luces');  
insert into equipos(id,reporte_tecnico_id, estado_id, numero_serie,  
modelo,          descripcion_problema,          accesorios)          values  
(88,25,3,'78lk','Modelo 800','Nintendo Switch con bateria de muy  
poca duracion','Cargador');  
insert into equipos(id,reporte_tecnico_id, estado_id, numero_serie,  
modelo,          descripcion_problema,          accesorios)          values  
(89,26,4,'98fh','Modelo 900','Monitor Dell sin respuesta de  
video','Cable de video');
```

Script del proyecto

TABLA DE COLA

```
CREATE TABLE cola (  
    id          NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,  
    equipos_id  NUMBER NOT NULL,  
    guia_cola   VARCHAR2(25) NOT NULL  
);  
  
ALTER TABLE cola ADD CONSTRAINT cola_pk PRIMARY KEY ( id );
```

Inserts de la tabla

```
insert into cola(id, equipos_id, guia_cola) values (150,85,'primero  
en atender');  
insert into cola(id, equipos_id, guia_cola) values (151,86,'segundo  
en atender');  
insert into cola(id, equipos_id, guia_cola) values (152,87,'tercero  
en atender');  
insert into cola(id, equipos_id, guia_cola) values (153,88,'cuarto  
en atender');  
insert into cola(id, equipos_id, guia_cola) values (154,89,'quinto  
en atender');  
  
ALTER TABLE cola  
    ADD CONSTRAINT cola_equipos_fk FOREIGN KEY ( equipos_id )  
        REFERENCES equipos ( id );
```

TABLA DE AUDITORÍA

```
CREATE TABLE log_clientes (  
    id NUMBER NOT NULL,  
    nombre VARCHAR2(30) NOT NULL,  
    primer_apellido VARCHAR2(50) NOT NULL,  
    segundo_apellido VARCHAR2(50) NOT NULL,  
    correo VARCHAR2(100) NOT NULL,  
    telefono NUMBER NOT NULL,  
    accion_realizada VARCHAR2(50),  
    usuario_elimina VARCHAR2(50),  
    fecha_elimina TIMESTAMP  
);
```

Script del proyecto

PROCEDIMIENTOS ALMACENADOS

CRUD CLIENTES

CREATE

```
CREATE OR REPLACE PROCEDURE INSERTclientes (  
    p_id CLIENTES.ID%TYPE,  
    p_nombre CLIENTES.NOMBRE%TYPE,  
    p_primer_apellido CLIENTES.PRIMER_APELLIDO%TYPE,  
    p_segundo_apellido CLIENTES.SEGUNDO_APELLIDO%TYPE,  
    p_correo CLIENTES.CORREO%TYPE,  
    p_telefono CLIENTES.TELEFONO%TYPE)  
IS  
BEGIN  
    INSERT INTO CLIENTES (ID, NOMBRE, PRIMER_APELLIDO,  
SEGUNDO_APELLIDO, CORREO, TELEFONO)  
VALUES (p_id, p_nombre, p_primer_apellido, p_segundo_apellido,  
p_correo, p_telefono);  
    COMMIT;  
END;
```

TEST

```
EXECUTE  
INSERTclientes(8,'Elci','Garro','Mata','egarrom@gmail.com',33333333)  
;
```

UPDATE

```
CREATE OR REPLACE PROCEDURE UPDATEclientes (  
    p_id IN CLIENTES.ID%TYPE,  
    p_nombre IN CLIENTES.NOMBRE%TYPE,  
    p_primer_apellido IN CLIENTES.PRIMER_APELLIDO%TYPE,  
    p_segundo_apellido IN CLIENTES.SEGUNDO_APELLIDO%TYPE,  
    p_correo IN CLIENTES.CORREO%TYPE,  
    p_telefono IN CLIENTES.TELEFONO%TYPE)  
IS  
BEGIN  
    UPDATE clientes SET ID = p_id, NOMBRE = p_nombre,  
PRIMER_APELLIDO = p_primer_apellido,  
SEGUNDO_APELLIDO = p_segundo_apellido, CORREO =  
p_correo, TELEFONO = p_telefono  
WHERE ID = p_id;  
    COMMIT;  
END;
```

SELECT

```
CREATE OR REPLACE PROCEDURE SELECTclientes (  
    p_id IN CLIENTES.ID%TYPE,  
    p_clientes_display OUT SYS_REFCURSOR)  
IS  
BEGIN  
    OPEN p_clientes_display for SELECT ID, NOMBRE,
```

Script del proyecto

```
PRIMER_APELLIDO,  
        SEGUNDO_APELLIDO, CORREO, TELEFONO FROM CLIENTES  
WHERE ID = p_id;  
END;
```

DELETE

```
CREATE OR REPLACE PROCEDURE DELETEclientes (p_id IN  
CLIENTES.ID%TYPE)  
IS  
BEGIN  
    DELETE FROM CLIENTES WHERE ID = p_id;  
END;
```

Test

```
EXECUTE DELETEclientes(8);
```

CRUD PRODUCTOS_NUEVOS

CREATE

```
CREATE OR REPLACE PROCEDURE INSERTproductos_nuevos (  
p_id PRODUCTOS_NUEVOS.ID%TYPE,  
p_codigo PRODUCTOS_NUEVOS.CODIGO%TYPE,  
p_nombre PRODUCTOS_NUEVOS.NOMBRE%TYPE,  
p_cantidad_stock PRODUCTOS_NUEVOS.CANTIDAD_STOCK%TYPE,  
p_accesorios PRODUCTOS_NUEVOS.ACCESORIOS%TYPE,  
p_estado PRODUCTOS_NUEVOS.ESTADO%TYPE,  
p_precio PRODUCTOS_NUEVOS.PRECIO%TYPE  
)  
IS  
BEGIN  
    insert into productos_nuevos(codigo, nombre, cantidad_stock,  
    accesorios, estado, precio)  
VALUES (p_codigo, p_nombre, p_cantidad_stock, p_accesorios,  
p_estado, p_precio);  
COMMIT;  
END;
```

TEST

```
EXECUTE INSERTproductos_nuevos(96, 'Tablet', 5, 'cargador', 'nuevo',  
25000);
```

UPDATE

```
CREATE OR REPLACE PROCEDURE UPDATEproductos_nuevos (  
p_id IN PRODUCTOS_NUEVOS.ID%TYPE,  
p_codigo IN PRODUCTOS_NUEVOS.CODIGO%TYPE,  
p_nombre IN PRODUCTOS_NUEVOS.NOMBRE%TYPE,  
p_cantidad_stock IN PRODUCTOS_NUEVOS.CANTIDAD_STOCK%TYPE,  
p_accesorios IN PRODUCTOS_NUEVOS.ACCESORIOS%TYPE,
```

Script del proyecto

```
p_estado IN PRODUCTOS_NUEVOS.ESTADO%TYPE,
p_precio IN PRODUCTOS_NUEVOS.PRECIO%TYPE
)
IS
BEGIN
UPDATE productos_nuevos SET CODIGO = p_codigo, NOMBRE = p_nombre,
                        CANTIDAD_STOCK = p_cantidad_stock, ACCESORIOS =
p_accesorios, ESTADO = p_estado,
                        PRECIO = p_precio
                        WHERE ID = p_id;

COMMIT;
END;

EXECUTE UPDATEproductos_nuevos(96, 'CELULAR', 5, 'cargador',
'nuevo', 25000);
```

SELECT

```
CREATE OR REPLACE PROCEDURE SELECTproductos_nuevos (
p_id IN PRODUCTOS_NUEVOS.ID%TYPE,
p_productos_nuevos_display OUT SYS_REFCURSOR
)
IS
BEGIN
OPEN p_productos_nuevos_display for SELECT id, codigo, nombre,
cantidad_stock,
accesorios, estado, precio FROM PRODUCTOS_NUEVOS
                        WHERE ID = p_id;

END;
```

DELETE

```
CREATE OR REPLACE PROCEDURE DELETEproductos_nuevos (
p_id IN PRODUCTOS_NUEVOS.ID%TYPE
)
IS
BEGIN
DELETE FROM PRODUCTOS_NUEVOS WHERE ID = p_id;
END;

--test--
EXECUTE DELETEproductos_nuevos(8);
```

CRUD PRODUCTOS_USADOS

CREATE

```
CREATE OR REPLACE PROCEDURE INSERTproductos_usados (
```

Script del proyecto

```
p_id PRODUCTOS_USADOS.ID%TYPE,  
p_codigo PRODUCTOS_USADOS.CODIGO%TYPE,  
p_nombre PRODUCTOS_USADOS.NOMBRE%TYPE,  
p_cantidad_stock PRODUCTOS_USADOS.CANTIDAD_STOCK%TYPE,  
p_accesorios PRODUCTOS_USADOS.ACESORIOS%TYPE,  
p_estado PRODUCTOS_USADOS.ESTADO%TYPE,  
p_precio PRODUCTOS_USADOS.PRECIO%TYPE  
)  
IS  
BEGIN  
insert into productos_usados(codigo, nombre, cantidad_stock,  
accesorios, estado, precio)  
VALUES (p_codigo, p_nombre, p_cantidad_stock, p_accesorios,  
p_estado, p_precio);  
COMMIT;  
END;
```

TEST

```
EXECUTE INSERTproductos_usados(96, 'Tablet', 5, 'cargador', 'nuevo',  
25000);
```

UPDATE

```
CREATE OR REPLACE PROCEDURE UPDATEproductos_usados (  
p_id IN PRODUCTOS_USADOS.ID%TYPE,  
p_codigo IN PRODUCTOS_USADOS.CODIGO%TYPE,  
p_nombre IN PRODUCTOS_USADOS.NOMBRE%TYPE,  
p_cantidad_stock IN PRODUCTOS_USADOS.CANTIDAD_STOCK%TYPE,  
p_accesorios IN PRODUCTOS_USADOS.ACESORIOS%TYPE,  
p_estado IN PRODUCTOS_USADOS.ESTADO%TYPE,  
p_precio IN PRODUCTOS_USADOS.PRECIO%TYPE  
)  
IS  
BEGIN  
UPDATE productos_usados SET CODIGO = p_codigo, NOMBRE = p_nombre,  
CANTIDAD_STOCK = p_cantidad_stock, ACCESORIOS =  
p_accesorios, ESTADO = p_estado,  
PRECIO = p_precio  
WHERE ID = p_id;  
  
COMMIT;  
END;  
  
EXECUTE UPDATEproductos_usados(96, 'CELULAR', 5, 'cargador',  
'nuevo', 25000);
```

SELECT

```
CREATE OR REPLACE PROCEDURE SELECTproductos_usados (  
p_id IN PRODUCTOS_USADOS.ID%TYPE,  
p_productos_usados_display OUT SYS_REFCURSOR  
)  
IS  
BEGIN  
OPEN p_productos_usados_display for SELECT id, codigo, nombre,
```

Script del proyecto

```
cantidad_stock,  
accesorios, estado, precio FROM PRODUCTOS_USADOS  
WHERE ID = p_id;  
END;
```

DELETE

```
CREATE OR REPLACE PROCEDURE DELETEproductos_usados (  
p_id IN PRODUCTOS_USADOS.ID%TYPE  
)  
IS  
BEGIN  
DELETE FROM PRODUCTOS_USADOS WHERE ID = p_id;  
END;
```

Test

```
EXECUTE DELETEproductos_usados(8);
```

CRUD REPUESTOS_NUEVOS

CREATE

```
CREATE OR REPLACE PROCEDURE INSERTrepuestos_nuevos (  
p_id REPUESTOS_NUEVOS.ID%TYPE,  
p_codigo REPUESTOS_NUEVOS.CODIGO%TYPE,  
p_nombre REPUESTOS_NUEVOS.NOMBRE%TYPE,  
p_cantidad_stock REPUESTOS_NUEVOS.CANTIDAD_STOCK%TYPE,  
p_accesorios REPUESTOS_NUEVOS.ACCESORIOS%TYPE,  
p_estado REPUESTOS_NUEVOS.ESTADO%TYPE,  
p_precio REPUESTOS_NUEVOS.PRECIO%TYPE  
)  
IS  
BEGIN  
insert into repuestos_nuevos(codigo, nombre, cantidad_stock,  
accesorios, estado, precio)  
VALUES (p_codigo, p_nombre, p_cantidad_stock, p_accesorios,  
p_estado, p_precio);  
COMMIT;  
END;
```

TEST

```
EXECUTE INSERTrepuestos_nuevos(96, 'Tablet', 5, 'cargador', 'nuevo',  
25000);
```

UPDATE

```
CREATE OR REPLACE PROCEDURE UPDATErepuestos_nuevos (  
p_id IN REPUESTOS_NUEVOS.ID%TYPE,  
p_codigo IN REPUESTOS_NUEVOS.CODIGO%TYPE,  
p_nombre IN REPUESTOS_NUEVOS.NOMBRE%TYPE,  
p_cantidad_stock IN REPUESTOS_NUEVOS.CANTIDAD_STOCK%TYPE,
```


Script del proyecto

```
p_accesorios IN REPUESTOS_NUEVOS.ACCESESORIOS%TYPE,
p_estado IN REPUESTOS_NUEVOS.ESTADO%TYPE,
p_precio IN REPUESTOS_NUEVOS.PRECIO%TYPE
)
IS
BEGIN
UPDATE repuestos_nuevos SET CODIGO = p_codigo, NOMBRE = p_nombre,
                        CANTIDAD_STOCK = p_cantidad_stock, ACCESORIOS =
p_accesorios, ESTADO = p_estado,
                        PRECIO = p_precio
                        WHERE ID = p_id;

COMMIT;
END;

EXECUTE UPDATerepuestos_nuevos(96, 'CELULAR', 5, 'cargador',
'nuevo', 25000);
```

```
SELECT
CREATE OR REPLACE PROCEDURE SELECTrepuestos_nuevos (
p_id IN REPUESTOS_NUEVOS.ID%TYPE,
p_repuestos_nuevos_display OUT SYS_REFCURSOR
)
IS
BEGIN
OPEN p_repuestos_nuevos_display for SELECT id, codigo, nombre,
cantidad_stock,
accesorios, estado, precio FROM REPUESTOS_NUEVOS
                        WHERE ID = p_id;

END;
```

```
DELETE
CREATE OR REPLACE PROCEDURE DELETerepuestos_nuevos (
p_id IN REPUESTOS_NUEVOS.ID%TYPE
)
IS
BEGIN
DELETE FROM REPUESTOS_NUEVOS WHERE ID = p_id;

END;
```

```
Test
EXECUTE DELETerepuestos_nuevos(8);
```

CRUD REPUESTOS_USADOS

```
CREATE

CREATE OR REPLACE PROCEDURE INSERTrepuestos_usados (
p_id REPUESTOS_USADOS.ID%TYPE,
p_codigo REPUESTOS_USADOS.CODIGO%TYPE,
p_nombre REPUESTOS_USADOS.NOMBRE%TYPE,
```

Script del proyecto

```
p_cantidad_stock REPUESTOS_USADOS.CANTIDAD_STOCK%TYPE,  
p_accesorios REPUESTOS_USADOS.ACCESORIOS%TYPE,  
p_estado REPUESTOS_USADOS.ESTADO%TYPE,  
p_precio REPUESTOS_USADOS.PRECIO%TYPE  
)  
IS  
BEGIN  
insert into repuestos_usados(codigo, nombre, cantidad_stock,  
accesorios, estado, precio)  
VALUES (p_codigo, p_nombre, p_cantidad_stock, p_accesorios,  
p_estado, p_precio);  
COMMIT;  
END;
```

TEST

```
EXECUTE INSERTrepuestos_usados(96, 'Tablet', 5, 'cargador', 'nuevo',  
25000);
```

UPDATE

```
CREATE OR REPLACE PROCEDURE UPDATErepuestos_nuevos (  
p_id IN REPUESTOS_USADOS.ID%TYPE,  
p_codigo IN REPUESTOS_USADOS.CODIGO%TYPE,  
p_nombre IN REPUESTOS_USADOS.NOMBRE%TYPE,  
p_cantidad_stock IN REPUESTOS_USADOS.CANTIDAD_STOCK%TYPE,  
p_accesorios IN REPUESTOS_USADOS.ACCESORIOS%TYPE,  
p_estado IN REPUESTOS_USADOS.ESTADO%TYPE,  
p_precio IN REPUESTOS_USADOS.PRECIO%TYPE  
)  
IS  
BEGIN  
UPDATE repuestos_usados SET CODIGO = p_codigo, NOMBRE = p_nombre,  
CANTIDAD_STOCK = p_cantidad_stock, ACCESORIOS =  
p_accesorios, ESTADO = p_estado,  
PRECIO = p_precio  
WHERE ID = p_id;  
  
COMMIT;  
END;
```

```
EXECUTE UPDATErepuestos_usados(96, 'CELULAR', 5, 'cargador',  
'nuevo', 25000);
```

SELECT

```
CREATE OR REPLACE PROCEDURE SELECTrepuestos_usados (  
p_id IN REPUESTOS_USADOS.ID%TYPE,  
p_repuestos_usados_display OUT SYS_REFCURSOR  
)  
IS  
BEGIN  
OPEN p_repuestos_usados_display for SELECT id, codigo, nombre,  
cantidad_stock,  
accesorios, estado, precio FROM REPUESTOS_USADOS  
WHERE ID = p_id;
```

Script del proyecto

END;

DELETE

```
CREATE OR REPLACE PROCEDURE DELETErepuestos_usados (  
  p_id IN REPUESTOS_USADOS.ID%TYPE  
)  
IS  
BEGIN  
  DELETE FROM REPUESTOS_USADOS WHERE ID = p_id;  
END;
```

Test

```
EXECUTE DELETEproductos_usados(8);
```

Script del proyecto

Triggers

Creación de Triggers de Auditoría para Clientes

```
CREATE OR REPLACE TRIGGER trg_borrar_clientes BEFORE DELETE ON
clientes FOR EACH ROW
DECLARE
    v_username VARCHAR2(50);
BEGIN
    SELECT USER INTO v_username FROM dual;
    INSERT INTO log_clientes(id, nombre, primer_apellido,
segundo_apellido,
correo, telefono, accion_realizada, usuario_elimina,
fecha_elimina)
VALUES (:OLD.id, :OLD.nombre, :OLD.primer_apellido,
:OLD.segundo_apellido, :OLD.correo,
:OLD.telefono, 'Se eliminó el registro de clientes',
v_username, SYSDATE);
END;
```

```
CREATE OR REPLACE TRIGGER trg_actualizar_clientes AFTER UPDATE ON
clientes FOR EACH ROW
DECLARE
    v_username VARCHAR2(50);
BEGIN
    SELECT USER INTO v_username FROM dual;
    INSERT INTO log_clientes(id, nombre, primer_apellido,
segundo_apellido,
correo, telefono, accion_realizada, usuario_elimina,
fecha_elimina)
VALUES (:NEW.id, :NEW.nombre, :NEW.primer_apellido,
:NEW.segundo_apellido, :NEW.correo,
:NEW.telefono, 'Se actualizó el registro de clientes',
v_username, SYSDATE);
END;
```

```
-- Pruebas
SELECT * FROM CLIENTES;
SELECT * FROM log_clientes;
EXECUTE
INSERTclientes(6,'PRUEBA','PRUEBA','PRUEBA','PRUEBA@gmail.com',00000
0000);
EXECUTE DELETEclientes(6);
```

Script del proyecto

Paquetes

Creación de paquetes

```
SET SERVEROUTPUT ON;

--DROP PACKAGE FACTURAS;

CREATE OR REPLACE PACKAGE FACTURAS
IS
    FUNCTION IMPUESTO(COSTO NUMBER) RETURN DECIMAL;
END;
```

CREAR BODY PARTE 2 IMPLEMENTACION

```
CREATE OR REPLACE PACKAGE BODY FACTURAS
IS

    FUNCTION IMPUESTO(COSTO NUMBER) RETURN DECIMAL
    IS

        IMPUESTO_IVA NUMBER(10,2);
    BEGIN
        IMPUESTO_IVA := COSTO * 0.13;
        RETURN IMPUESTO_IVA;
    END;
END;
```

PRUEBA DE FUNCION

```
DECLARE
    IMP_IVA DECIMAL;
BEGIN
    IMP_IVA := FACTURAS.IMPUESTO(10000);
    DBMS_OUTPUT.PUT_LINE('El iva del 13% es.: ' || IMP_IVA);
END;
```