

UNIVERSIDAD PRIVADA DEL NORTE

Diseño de una base de datos para los procesos de Delivery en el local “La casa del liquor” usando ORACLE DATABASE

INTEGRANTES

Acosta Fernández Ricardo H., Castillo García Mayte A., Palacios Martel Anthony I., Tafur Cabrera Paul J.

Estudiantes de la Carrera de Ingeniería de Sistemas Computacionales de la Universidad Privada del Norte (UPN) – Sede Los Olivos

DOCENTE:

Chavez Bardales Victor I.

Lima, Perú

Septiembre – 2022

Índice

I. NOMBRE DE LA EMPRESA	3
II. ESENCIA DEL NEGOCIO	3
III. PROCESO A IMPLEMENTAR	3
IV. DESCRIPCION DEL CASO	3
V. REGLAS DE NEGOCIO	4
VI. GLOSARIO DE TERMINOS.....	5
VII. SCRIPTS	6
1. CREACION DE PDBS Y TABLESPACE	6
2. CREACION ROLES, PERFILES, USUARIOS Y PERMISOS	8
3. CREACION DE TABLAS Y RESTRICCIONES	13
VIII. CONCLUSIONES.....	45

I. NOMBRE DE LA EMPRESA

1.1. Nombre de la empresa

Local "La casa del liquor" (Atención al público +18)

1.2. Datos de la empresa

Giro: Licorería

Ubicación: Av. Jose Carlos Mariategui Mz I Lt 14 - Ancon

Ruc:

1.3. Situación actual

La empresa inicio sus actividades en septiembre del 2021 y se encuentra activo hasta la actualidad.

1.4. Área o actividad a implementar

Implementaremos de una base de datos orientada hacia el Control de distribución de pedidos en la modalidad de delivery.

II. ESENCIA DEL NEGOCIO

La empresa "La casa del liquor", brinda servicios de venta licores al publico +18 en general de manera organizada.

III. PROCESO A IMPLEMENTAR

Implementar el control de la distribución de licor en la modalidad delivery que cuenta con un proceso que tiene como finalidad controlar y mejorar el flujo que se tiene sobre sus recursos y el personal. Esto es debido a que habrá una mejora en dicho proceso, que es un ciclo, que se inicia desde que en la tienda recibe el pedido de las personas a las cual despachar, hasta que se retiran el personal que realiza la gestión del delivery.

IV. DESCRIPCION DEL CASO

La Casa Del Liquor es una licorería dedicada al servicio público para mayores de +18 años. La capacidad de la atención al cliente dependerá del aforo ya declarado.

El personal de almacén del establecimiento son los encargados de la recepción de la mercadería por parte de los proveedores y la atención al cliente, además, de la preparación de etiquetas y la limpieza del lugar.

El área de recepción se encarga de atender y/o recibir los pedidos solicitados vía telefónica y/o presencial, ellos también se encargan de la revisión final, previa a su despacho. Asimismo, el personal que se encarga de la distribución de los pedidos son denominados repartidores(delivery).

Por otro lado, la labor es compartida entre las dos áreas, ya que cuando el paquete (pack de la promoción) se encuentre listo, el personal denominado cajero se encargará del cobro del producto pedido y entrega al cliente o al repartidor para distribuir el paquete al cliente.

V. REGLAS DE NEGOCIO

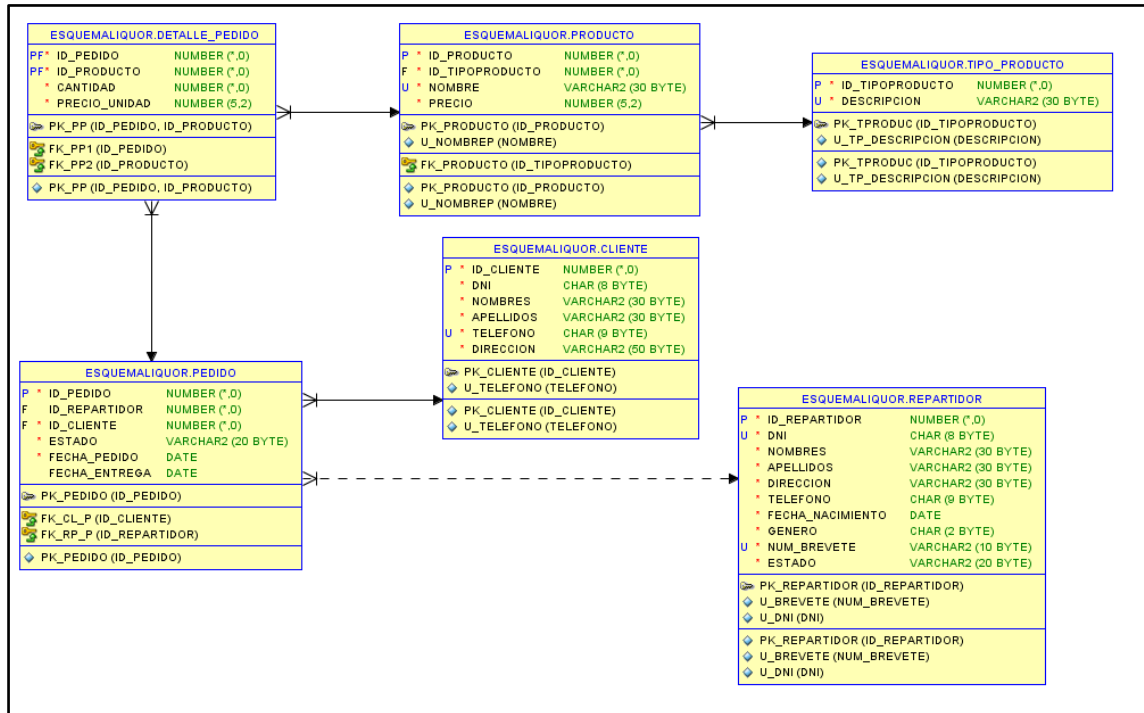
1. Un empleado solo puede pertenecer a una sola área.
2. Cada área solo tiene un encargado y un encargado solo lo puede ser de una sola área.
3. Deben mantener su uniforme de trabajo en perfecto estado tanto si están de cara al público. Los hombres bien afeitados y peinados y las mujeres sin excesos de joyas, perfume o maquillaje.
4. El nombre de un licor no se puede repetir.
5. La cantidad de un licor en un pedido debe ser mayor a 0.
6. EL DNI de un cliente debe ser único.
7. El teléfono de un cliente debe ser único.
8. La dirección de un cliente puede pertenecer a otros clientes.
9. Un repartidor obtendrá un bono adicional según los pedidos que entregue.
10. El estado de los Repartidores solo puede ser "disponible" o "ocupado", solo un repartidor que esté disponible podrá realizar el envío, una vez que se confirmó la entrega del pedido el estado del empleado pasara a disponible nuevamente.
11. El Pedido solo puede tener 3 estados: pendiente, enviado y entregado.
12. Solo se podrá confirmar el envío si el estado del pedido es "enviado", una vez confirmado el envío el estado pasara a "entregado" y se actualizara la fecha de entrega.
13. Se podrá asignar al pedido un repartidor si su estado está en "pendiente", una vez asignado, el estado se actualizará a "enviado".

VI. GLOSARIO DE TERMINOS

Término	Descripción
Cliente	Cliente que acude a comprar una buena oferta.
Orden	Es cuando el cliente o comensal realiza un pedido de una promoción hacia el recepcionista que le estará tomando su orden.
Encargado	Es el personal a cargo de una determinada área.
Empleado	Es la generalización aplicada a todos los empleados de la Empresa.
Delivery	Hace referencia a los datos usados en el transporte de los Motorizados.
Pedido	Es la expresión vía telefónica de los deseos del cliente.

VII. SCRIPTS

MODELO RELACIONAL



1. CREACION DE PDBS Y TABLESPACE

Script de la creación de PDB y Tablespace: por ejemplo, puede definir 3 tablespaces (DATOS, INDICES Y Temporal), con sus respectivos datafiles y tempfile.

CREACION DE PDB

```
CREATE PLUGGABLE DATABASE PROYECTO
ADMIN USER ADMIN IDENTIFIED BY DATABASE
FILE_NAME_CONVERT=('D:\Oracle\oradata\XE\pdbseed',
'D:\Oracle\oradata\XE\PROYECTO');
```

```
ALTER PLUGGABLE DATABASE PROYECTO OPEN;
ALTER SESSION SET CONTAINER=PROYECTO;
```

```
SQL> CREATE PLUGGABLE DATABASE PROYECTO
  2  ADMIN USER ADMIN IDENTIFIED BY DATABASE
  3  FILE_NAME_CONVERT=('D:\Oracle\oradata\XE\pdbseed', 'D:\Oracle\oradata\XE\PROYECTO');

Base de datos de conexi3n creada.

SQL> ALTER PLUGGABLE DATABASE PROYECTO OPEN;

Base de datos de conexi3n modificada.

SQL> ALTER SESSION SET CONTAINER=PROYECTO;

Sesi3n modificada.

SQL> SHOW CON_NAME

CON_NAME
-----
PROYECTO
SQL>
```

CREACION DE TABLESPACE

```
CREATE TABLESPACE TBLIQUOR_DATOS
DATAFILE 'D:\Oracle\oradata\XE\PROYECTO\PDBLIQUOR01.dbf'
SIZE 100M
AUTOEXTEND ON NEXT 5M
MAXSIZE 300M;
```

```
CREATE TEMPORARY TABLESPACE TBLIQUOR_TEMP
TEMPFILE 'D:\Oracle\oradata\XE\PROYECTO\TEMPLIQUOR01'
SIZE 50M
AUTOEXTEND ON NEXT 5M
MAXSIZE 100M;
```

```
SQL> CREATE TABLESPACE TBLIQUOR_DATOS
  2  DATAFILE 'D:\Oracle\oradata\XE\PROYECTO\PDBLIQUOR01.dbf'
  3  SIZE 100M
  4  AUTOEXTEND ON NEXT 5M
  5  MAXSIZE 300M;

Tablespace creado.

SQL> CREATE TEMPORARY TABLESPACE TBLIQUOR_TEMP
  2  TEMPFILE 'D:\Oracle\oradata\XE\PROYECTO\TEMPLIQUOR01'
  3  SIZE 50M
  4  AUTOEXTEND ON NEXT 5M
  5  MAXSIZE 100M;

Tablespace creado.

SQL> _
```

2. CREACION ROLES, PERFILES, USUARIOS Y PERMISOS

Script para la creación de roles, perfiles, usuarios y permisos: Por ejemplo, puede definir dos roles (admin, consulta y esquema), 2 perfiles (esquema y visor) recuerde que esto permite gestionar las contraseñas. Para los usuarios puede crear los que crea convenientes, uno de esquema, un dba, etc.

ROLES

```
CREATE ROLE rol_admin;  
GRANT DBA TO rol_admin;
```

```
SQL> CREATE ROLE rol_admin;  
Rol creado.  
SQL> GRANT DBA TO rol_admin;  
Concesión terminada correctamente.  
SQL>
```

```
CREATE ROLE rol_esquema;  
GRANT CREATE SESSION, RESOURCE TO rol_esquema;
```

```
SQL> CREATE ROLE rol_esquema;  
Rol creado.  
SQL> GRANT CREATE SESSION, RESOURCE TO rol_esquema;  
Concesión terminada correctamente.  
SQL>
```



```
CREATE ROLE rol_consulta;  
grant create session to rol_consulta;  
GRANT SELECT ANY TABLE TO rol_consulta;
```

```
SQL> CREATE ROLE rol_consulta;  
Rol creado.  
SQL> grant create session to rol_consulta;  
Concesi n terminada correctamente.  
SQL> GRANT SELECT ANY TABLE TO rol_consulta;  
Concesi n terminada correctamente.  
SQL>
```

PERFILES

```
CREATE PROFILE perfil_esquema LIMIT  
FAILED_LOGIN_ATTEMPTS 6  
IDLE_TIME 60  
SESSIONS_PER_USER 1  
PASSWORD_LIFE_TIME 150  
CONNECT_TIME 180;
```

```
SQL> CREATE PROFILE perfil_esquema LIMIT  
2 FAILED_LOGIN_ATTEMPTS 6  
3 IDLE_TIME 60  
4 SESSIONS_PER_USER 1  
5 PASSWORD_LIFE_TIME 150  
6 CONNECT_TIME 180;  
Perfil creado.  
SQL>
```

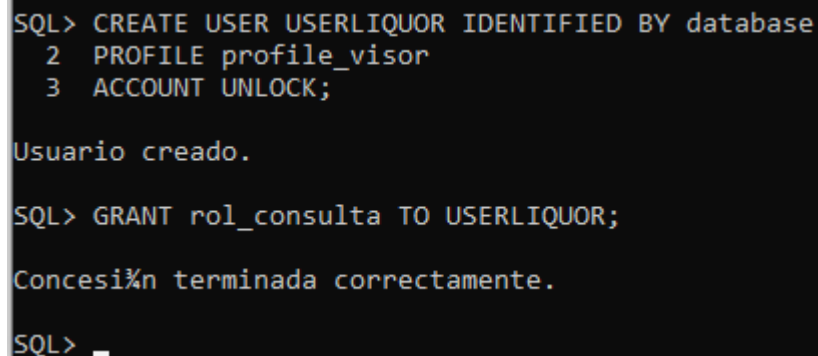
```
CREATE PROFILE profile_visor LIMIT  
FAILED_LOGIN_ATTEMPTS 3  
PASSWORD_LOCK_TIME 2  
PASSWORD_LIFE_TIME 60  
PASSWORD_GRACE_TIME 2  
SESSIONS_PER_USER 2  
IDLE_TIME 30  
CONNECT_TIME 60;
```

```
SQL> CREATE PROFILE profile_visor LIMIT  
2  FAILED_LOGIN_ATTEMPTS 3  
3  PASSWORD_LOCK_TIME 2  
4  PASSWORD_LIFE_TIME 60  
5  PASSWORD_GRACE_TIME 2  
6  SESSIONS_PER_USER 2  
7  IDLE_TIME 30  
8  CONNECT_TIME 60;  
  
Perfil creado.  
  
SQL>
```

USUARIOS

```
CREATE USER USERLIQUOR IDENTIFIED BY database  
PROFILE profile_visor  
ACCOUNT UNLOCK;  
GRANT rol_consulta TO USERLIQUOR;
```

Este usuario solo podrá realizar consultas a todas las tablas del BD, con el fin de constatar la información.



```
SQL> CREATE USER USERLIQUOR IDENTIFIED BY database  
2  PROFILE profile_visor  
3  ACCOUNT UNLOCK;  
  
Usuario creado.  
  
SQL> GRANT rol_consulta TO USERLIQUOR;  
  
Concesión terminada correctamente.  
  
SQL>
```

```
CREATE USER ESQUEMALIQUOR IDENTIFIED BY database  
DEFAULT TABLESPACE TBLIQUOR_DATOS  
TEMPORARY TABLESPACE TBLIQUOR_TEMP  
PROFILE perfil_esquema  
ACCOUNT UNLOCK;
```

```
ALTER USER ESQUEMALIQUOR QUOTA 50M ON  
TBLIQUOR_DATOS;  
GRANT rol_esquema to ESQUEMALIQUOR;
```

Este usuario se encargará de realizar el esquema de la base de datos del restaurante, podrá ingresar registros y crear funciones, vistas y procedimientos almacenados.

```
SQL> CREATE USER ESQUEMALIQUOR IDENTIFIED BY database
2  DEFAULT TABLESPACE TBLIQUOR_DATOS
3  TEMPORARY TABLESPACE TBLIQUOR_TEMP
4  PROFILE perfil_esquema
5  ACCOUNT UNLOCK;

Usuario creado.

SQL> ALTER USER ESQUEMALIQUOR QUOTA 50M ON TBLIQUOR_DATOS;

Usuario modificado.

SQL> GRANT rol_esquema to ESQUEMALIQUOR;

Concesi n terminada correctamente.

SQL> _
```

```
CREATE USER ADMINLIQUOR IDENTIFIED BY database
ACCOUNT UNLOCK;
GRANT rol_admin TO USERLIQUOR;
```

Este usuario tendr  todos los permisos para hacer cambios en la BD si son requeridos, adem s de habilitar el acceso a los otros usuarios.

```
SQL> CREATE USER ADMINLIQUOR IDENTIFIED BY database
2  ACCOUNT UNLOCK;

Usuario creado.

SQL> GRANT rol_admin TO USERLIQUOR;

Concesi n terminada correctamente.

SQL> _
```

3. CREACION DE TABLAS Y RESTRICCIONES

CREACION DE TABLAS

Script para la creación de las tablas también le agregamos sus restricciones.

--TABLA REPARTIDOR

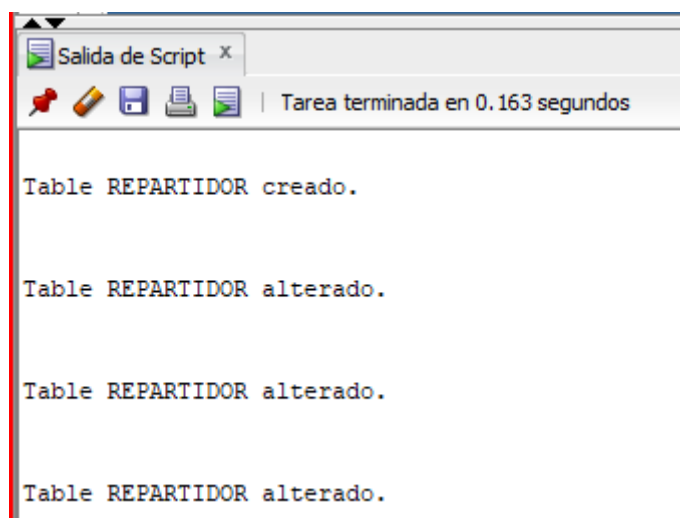
```
CREATE TABLE Repartidor(  
  id_repartidor INT,  
  dni CHAR(8) NOT NULL,  
  nombres VARCHAR2(30) NOT NULL,  
  apellidos VARCHAR2(30) NOT NULL,  
  direccion VARCHAR2(30) NOT NULL,  
  telefono CHAR(9) NOT NULL,  
  fecha_nacimiento DATE NOT NULL,  
  genero CHAR(2) NOT NULL,  
  num_brevete VARCHAR2(10) NOT NULL,  
  estado VARCHAR(20) DEFAULT 'disponible' NOT NULL  
);
```

--Primary Key

```
ALTER TABLE REPARTIDOR ADD CONSTRAINT PK_REPARTIDOR PRIMARY  
KEY(ID_REPARTIDOR);
```

--Restricciones

```
ALTER TABLE REPARTIDOR ADD CONSTRAINT u_brevete UNIQUE  
(NUM_BREVETE);  
ALTER TABLE REPARTIDOR ADD CONSTRAINT u_dni UNIQUE (DNI);  
ALTER TABLE REPARTIDOR ADD CONSTRAINT ck_estadoR CHECK  
(ESTADO='disponible' OR ESTADO='ocupado');  
ALTER TABLE REPARTIDOR ADD CONSTRAINT ck_generoR CHECK (GENERO  
= 'M' OR GENERO='F');
```



--TABLA CLIENTE

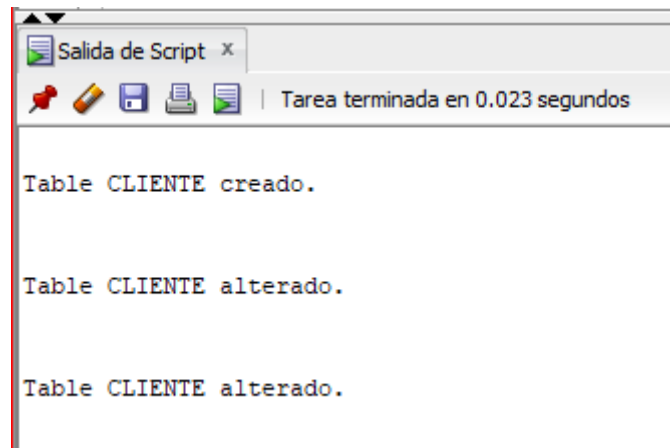
```
CREATE TABLE Cliente(  
id_cliente INT,  
dni CHAR(8) NOT NULL,  
nombres VARCHAR2(30) NOT NULL,  
apellidos VARCHAR2(30) NOT NULL,  
telefono CHAR(9) NOT NULL,  
direccion VARCHAR2(15) NOT NULL  
);
```

--PRIMARY KEY

```
ALTER TABLE CLIENTE ADD CONSTRAINT pk_cliente PRIMARY  
KEY(ID_CLIENTE);
```

--Restricciones

```
ALTER TABLE CLIENTE ADD CONSTRAINT u_telefono UNIQUE  
(TELEFONO);
```



--TABLE TIPO PRODUCTO

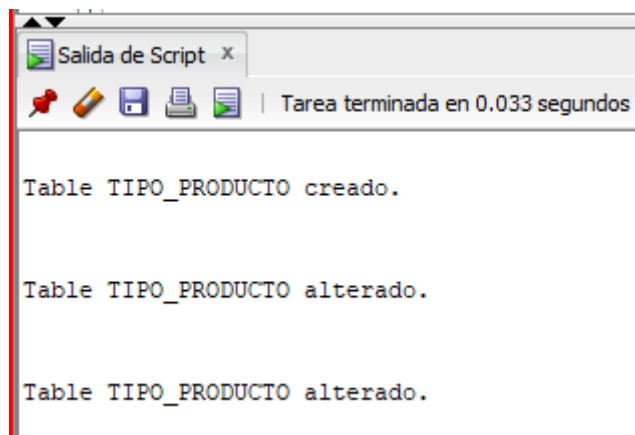
```
CREATE TABLE TIPO_PRODUCTO (  
ID_TIPOPRODUCTO INT,  
DESCRIPCION VARCHAR2(30) NOT NULL  
);
```

--PRYMARY KEY

```
ALTER TABLE TIPO_PRODUCTO ADD CONSTRAINT pk_tproduc PRIMARY  
KEY(ID_TIPOPRODUCTO);
```

--Restricciones

```
ALTER TABLE TIPO_PRODUCTO ADD CONSTRAINT u_tp_descripcion  
UNIQUE(DESCRIPCION);
```



--TABLA PRODUCTO

```
CREATE TABLE PRODUCTO(  
  id_producto INT,  
  id_tipoproducto INTEGER NOT NULL,  
  nombre VARCHAR2(30) NOT NULL,  
  precio number(5,2) NOT NULL  
);
```

--PRIMARY KEY

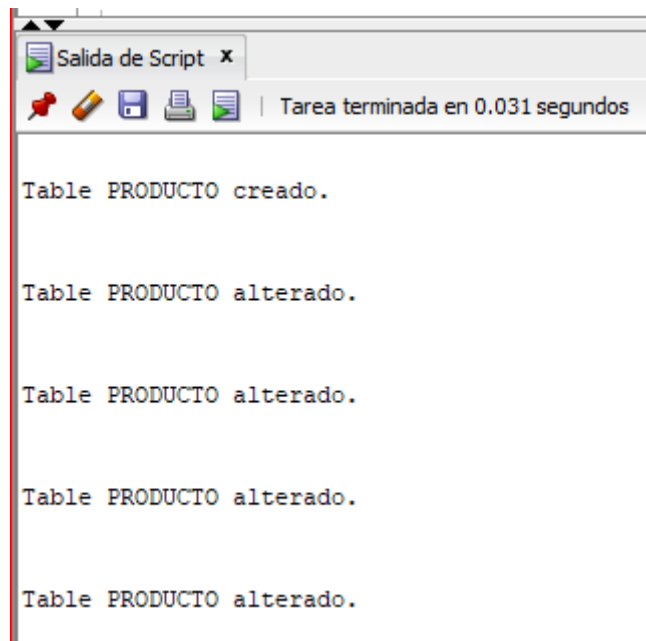
```
ALTER TABLE PRODUCTO ADD CONSTRAINT pk_producto PRIMARY  
KEY(ID_PRODUCTO);
```

--FOREING KEY

```
ALTER TABLE PRODUCTO ADD CONSTRAINT fk_producto FOREIGN  
KEY(ID_TIPOPRODUCTO) REFERENCES TIPO_PRODUCTO;
```

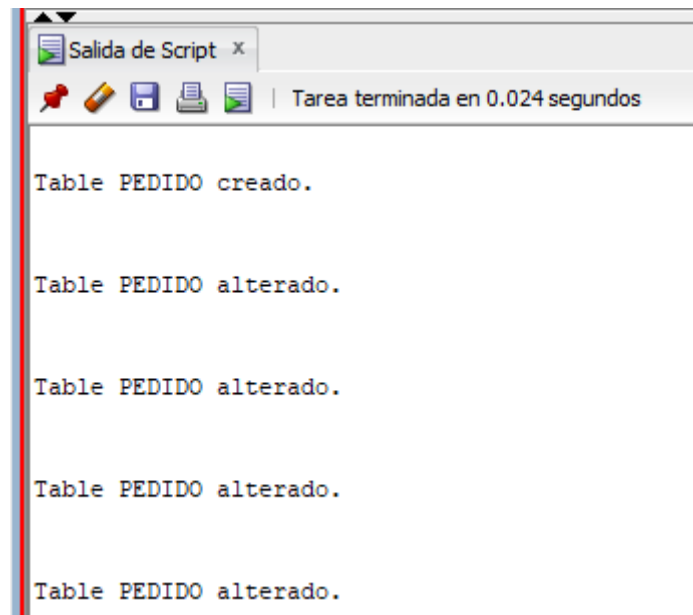
--Restricciones

```
ALTER TABLE PRODUCTO ADD CONSTRAINT ck_precio CHECK (PRECIO >  
0);  
ALTER TABLE PRODUCTO ADD CONSTRAINT u_nombreP UNIQUE  
(NOMBRE);
```



--TABLA PEDIDO

```
CREATE TABLE Pedido(  
  id_pedido INT,  
  id_repartidor INT,  
  id_cliente INT NOT NULL,  
  estado VARCHAR2(20) DEFAULT 'pendiente' NOT NULL,  
  fecha_pedido DATE DEFAULT SYSDATE NOT NULL,  
  fecha_entrega DATE NOT NULL  
);  
  
--PRIMARY KEY  
ALTER TABLE PEDIDO ADD CONSTRAINT pk_pedido PRIMARY  
KEY(ID_PEDIDO);  
  
--FOREING KEY  
ALTER TABLE PEDIDO ADD CONSTRAINT fk_cl_p FOREIGN  
KEY(ID_CLIENTE) REFERENCES CLIENTE;  
ALTER TABLE PEDIDO ADD CONSTRAINT fk_rp_p FOREIGN  
KEY(ID_REPARTIDOR) REFERENCES REPARTIDOR;  
  
--RESTRICCIÓN  
ALTER TABLE PEDIDO ADD CONSTRAINT ck_estado CHECK  
(ESTADO='pendiente' OR ESTADO='enviado' OR  
ESTADO='entregado');
```



--TABLA DETALLE_PEDIDO

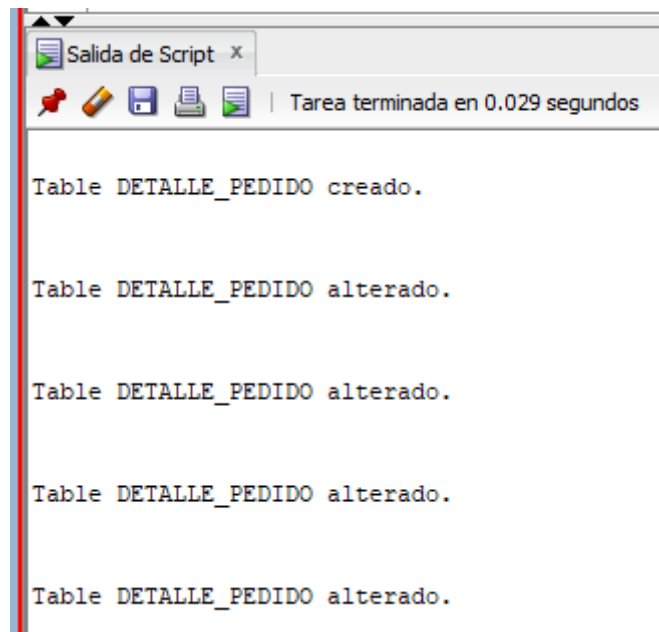
```
CREATE TABLE Detalle_pedido(  
  id_pedido INT,  
  id_producto INT NOT NULL,  
  cantidad INT NOT NULL,  
  precio_unidad NUMBER(5,2) NOT NULL,  
  CONSTRAINT PK_PP PRIMARY KEY (id_pedido, id_producto)  
);
```

--PRIMARY KEY

```
ALTER TABLE DETALLE_PEDIDO ADD CONSTRAINT fk_pp1 FOREIGN KEY  
(ID_PEDIDO) REFERENCES PEDIDO;  
ALTER TABLE DETALLE_PEDIDO ADD CONSTRAINT fk_pp2 FOREIGN KEY  
(ID_PRODUCTO) REFERENCES PRODUCTO;
```

--RESTRICCIONES

```
ALTER TABLE DETALLE_PEDIDO ADD CONSTRAINT ck_precioU_pd CHECK  
(PRECIO_UNIDAD > 0);  
ALTER TABLE DETALLE_PEDIDO ADD CONSTRAINT ck_cantidad_pd  
CHECK (cantidad > 0);
```



a. Registros

--DATOS REPARTIDORES

```
INSERT INTO REPARTIDOR VALUES(1,'60580441','Miguel Angel','Gonzales
  Marquez','av.los
  jirasoles','984123655','10/09/1998','M','T156632548',DEFAULT);
INSERT INTO REPARTIDOR VALUES(2,'17576226','Javier Vergara','James
  Velazquez','Ramón Castilla 685 CC
  Aurora','948216895','06/01/1990','M','T156678963',DEFAULT);
INSERT INTO REPARTIDOR VALUES(3,'64018550','Baltasar
  Soria','Patrick Barrett','av.angamos este
  1560','978456811','05/05/1995','F','T666632548',DEFAULT);
INSERT INTO REPARTIDOR VALUES(4,'41036589','Candido
  Leon','Rodriguez Preston','av.lima
  1556','956515556','05/08/1992','M','T154938548',DEFAULT);
INSERT INTO REPARTIDOR VALUES(5,'11845438','Anthony Ivan','Palacios
  Martel','av.maria parado
  bellido','940205924','15/08/1999','M','T555532548',DEFAULT);
INSERT INTO REPARTIDOR VALUES(6,'12883026','David
  Blazquez','Padilla Lopez','jr.ramon
  castillo','981237894','11/02/1995','M','T156632888',DEFAULT);
INSERT INTO REPARTIDOR VALUES(7,'87264713','Mayte
  Alessandra','Moran Salas','jr.gamara
  1861','951356478','09/09/1995','F','T156444548',DEFAULT);
INSERT INTO REPARTIDOR VALUES(8,'46192396','Diego Chico','Moreno
  Brandt','av.miraflores norte
  cc','912567459','21/09/1989','M','T152232548',DEFAULT);
INSERT INTO REPARTIDOR VALUES(9,'74079725','Pedro Jose','Zavala
  Moran','jr.pilora
  485','988156334','26/05/1999','M','T156631234',DEFAULT);
INSERT INTO REPARTIDOR VALUES(10,'89821736','Antonio
  Manuel','Herrera
  Juarez','av.venezuela','917712368','10/12/1997','M','T156541248'
  ,DEFAULT);
COMMIT;
```

ID_REPARTIDOR	DNI	NOMBRES	APELLIDOS	DIRECCION	TELEFONO	FECHA_NACIMIENTO	GENERO	NUM_BREVETE	ESTADO
1	160580441	Miguel Angel	Gonzales Marquez	av.los jirasoles	984123655	10/09/98	M	T156632548	disponible
2	217576226	Javier Vergara	James Velazquez	Ramón Castilla 685 CC Aurora	948216895	06/01/90	M	T156678963	disponible
3	364018550	Baltasar Soria	Patrick Barrett	av.angamos este 1560	978456811	05/05/95	F	T666632548	disponible
4	441036589	Candido Leon	Rodriguez Preston	av.lima 1556	956515556	05/08/92	M	T154938548	disponible
5	511845438	Anthony Ivan	Palacios Martel	av.maria parado bellido	940205924	15/08/99	M	T555532548	disponible
6	612883026	David Blazquez	Padilla Lopez	jr.ramon castillo	981237894	11/02/95	M	T156632888	disponible
7	787264713	Mayte Alessandra	Moran Salas	jr.gamara 1861	951356478	09/09/95	F	T156444548	disponible
8	846192396	Diego Chico	Moreno Brandt	av.miraflores norte cc	912567459	21/09/89	M	T152232548	disponible
9	974079725	Pedro Jose	Zavala Moran	jr.pilora 485	988156334	26/05/99	M	T156631234	disponible
10	1089821736	Antonio Manuel	Herrera Juarez	av.venezuela	917712368	10/12/97	M	T156541248	disponible

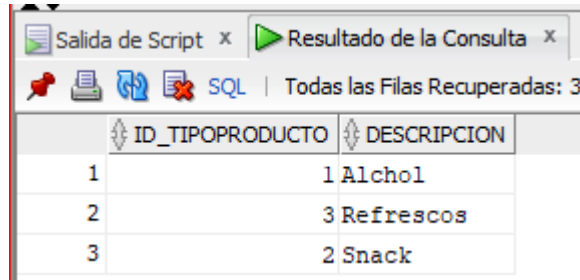
--Datos Clientes

```
INSERT INTO CLIENTE VALUES(1,'08612345','Nayara Gallardo','Kelley
  Marquez','932104412','Av.Peru con Universitaria');
INSERT INTO CLIENTE VALUES(2,'00681684','Pascual Bolaños','James
  Sampson','898262265','Ovalo Jose Granda');
INSERT INTO CLIENTE VALUES(3,'20125515','Rayan Torralba','Becker
  Barrett','332285507','Av.Abancay, Cercado de Lima');
INSERT INTO CLIENTE VALUES(4,'02369845','Felipa Camacho','Rodriguez
  Gutierrez','446565898','Calle Santa Martina 384-EL CERCADO-
  Lima');
INSERT INTO CLIENTE VALUES(5,'01681684','Virgilio Tejedor','Ewing
  Villanueva','755854384','Jr.Jorge Chávez 1685 Breña');
INSERT INTO CLIENTE VALUES(6,'12354681','Amelia Alba','Levine
  Lopez','353530261','Jr.Cajamarca 371 Rimac');
INSERT INTO CLIENTE VALUES(7,'12368600','Lazaro Latorre','Moran
  Marquez','658947134','Av.La Paz Lt.30-Huachipa');
INSERT INTO CLIENTE VALUES(8,'12354952','Adam Cid','Vega
  Brandt','215423968','Calle Perusa 1480-Lima');
INSERT INTO CLIENTE VALUES(9,'00238515','Marcelina Gracia','Zavala
  Pacheco','486297235','Psje.Vicus L 56-La Castellana-Surco');
INSERT INTO CLIENTE VALUES(10,'13519645','Ruben Sales','Ayala
  Juarez','256217367','G.Varela 1878 - Breña');
COMMIT;
```

ID_CLIENTE	DNI	NOMBRES	APELLIDOS	TELEFONO	DIRECCION
1	108612345	Nayara Gallardo	Kelley Marquez	932104412	Av.Peru con Universitaria
2	200681684	Pascual Bolaños	James Sampson	898262265	Ovalo Jose Granda
3	320125515	Rayan Torralba	Becker Barrett	332285507	Av.Abancay, Cercado de Lima
4	402369845	Felipa Camacho	Rodriguez Gutierrez	446565898	Calle Santa Martina 384-EL CERCADO-Lima
5	501681684	Virgilio Tejedor	Ewing Villanueva	755854384	Jr.Jorge Chávez 1685 Breña
6	612354681	Amelia Alba	Levine Lopez	353530261	Jr.Cajamarca 371 Rimac
7	712368600	Lazaro Latorre	Moran Marquez	658947134	Av.La Paz Lt.30-Huachipa
8	812354952	Adam Cid	Vega Brandt	215423968	Calle Perusa 1480-Lima
9	900238515	Marcelina Gracia	Zavala Pacheco	486297235	Psje.Vicus L 56-La Castellana-Surco
10	1013519645	Ruben Sales	Ayala Juarez	256217367	G.Varela 1878 - Breña

--Datos TIPO PRODUCTO

```
INSERT INTO TIPO_PRODUCTO VALUES(1,'Alchol');
INSERT INTO TIPO_PRODUCTO VALUES(2,'Snack');
INSERT INTO TIPO_PRODUCTO VALUES(3,'Refrescos');
COMMIT;
```



ID_TIPOPRODUCTO	DESCRIPCION
1	1 Alchol
2	3 Refrescos
3	2 Snack

--Datos PRODUCTO

--Alcohol

```
INSERT INTO PRODUCTO VALUES(1,1,'Pisco Queirolo Quebranta',32.90);
INSERT INTO PRODUCTO VALUES(2,1,'Whisky Black Label 750
ml',149.90);
INSERT INTO PRODUCTO VALUES(3,1,'Ron Cartavio 750 ml',189.90);
INSERT INTO PRODUCTO VALUES(4,1,'Cerveza Pilsen Six Pack',20.90);
INSERT INTO PRODUCTO VALUES(5,1,'Cerveza Corona Six Pack',35);
INSERT INTO PRODUCTO VALUES(6,1,'Vodka Absolut 750 ml',45.90);
INSERT INTO PRODUCTO VALUES(7,1,'Vino Santiago Queirolo 750
ml',17.90);
INSERT INTO PRODUCTO VALUES(8,1,'Jagermeister 700 ml',69.90);
INSERT INTO PRODUCTO VALUES(9,1,'Cerveza Budweiser Six
Pack',12.90);
INSERT INTO PRODUCTO VALUES(10,1,'Tequila Jose Cuervo 375
ml',37.50);
```


--Snack

```
INSERT INTO PRODUCTO VALUES(11,2,'Camote Villa Natura x 150
g',5.40);
INSERT INTO PRODUCTO VALUES(12,2,'Chifles V.Natura Salados 150
g',4.90);
INSERT INTO PRODUCTO VALUES(13,2,'Piqueo Snax 280 Gr',12.90);
INSERT INTO PRODUCTO VALUES(14,2,'Snack Doritos Dippas 200
g',10.90);
INSERT INTO PRODUCTO VALUES(15,2,'Papas LAYS Clásicas 210 g',8.50);
```

--Refrescos

```
INSERT INTO PRODUCTO VALUES(16,3,'Coca Cola 3 Lt',12.50);
INSERT INTO PRODUCTO VALUES(17,3,'Inka Kola 3 Lt',12.50);
INSERT INTO PRODUCTO VALUES(18,3,'B.Evervess Carbonatada 1.5
Lt',6.90);
INSERT INTO PRODUCTO VALUES(19,3,'Agua San Carlos sin Gas 3 Lt',5);
INSERT INTO PRODUCTO VALUES(20,3,'Cifrut Citrus Punch 1.5
Lt',6.50);
INSERT INTO PRODUCTO VALUES(21,3,'Monster Energy 473 ml',7.50);
COMMIT;
```

Salida de Script x Resultado de la Consulta x

 Todas las Filas Recuperadas: 21 en 0.009 segundos

ID_PRODUCTO	ID_TIPOPRODUCTO	NOMBRE	PRECIO
1	1	1 Pisco Queirolo Quebranta	32.9
2	2	1 Whisky Black Label 750 ml	149.9
3	3	1 Ron Cartavio 750 ml	189.9
4	4	1 Cerveza Pilsen Six Pack	20.9
5	5	1 Cerveza Corona Six Pack	35
6	6	1 Vodka Absolut 750 ml	45.9
7	7	1 Vino Santiago Queirolo 750 ml	17.9
8	8	1 Jagermeister 700 ml	69.9
9	9	1 Cerveza Budweiser Six Pack	12.9
10	10	1 Tequila Jose Cuervo 375 ml	37.5
11	11	2 Camote Villa Natura x 150 g	5.4
12	13	2 Piqueo Snax 280 Gr	12.9
13	14	2 Snack Doritos Dippas 200 g	10.9
14	12	2 Chifles V.Natura Salados 150 g	4.9
15	15	2 Papas LAYS Clásicas 210 g	8.5
16	16	3 Coca Cola 3 Lt	12.5
17	17	3 Inka Kola 3 Lt	12.5
18	19	3 Agua San Carlos sin Gas 3 Lt	5
19	20	3 Cifrut Citrus Punch 1.5 Lt	6.5
20	21	3 Monster Energy 473 ml	7.5
21	18	3 B.Evervess Carbonatada 1.5 Lt	6.9

--Datos Pedido

```

INSERT INTO PEDIDO VALUES(1,1,1,'entregado',DEFAULT,
    TO_DATE('22/10/2022 13:12:00','DD/MM/YY HH24:MI:SS'));
INSERT INTO PEDIDO VALUES(2,2,2,'entregado',DEFAULT,
    TO_DATE('22/10/2022 14:12:00','DD/MM/YY HH24:MI:SS'));
INSERT INTO PEDIDO VALUES(3,null,3,DEFAULT,DEFAULT, null);
INSERT INTO PEDIDO VALUES(4,3,4,'entregado',DEFAULT,
    TO_DATE('22/10/2022 15:12:00','DD/MM/YY HH24:MI:SS'));
INSERT INTO PEDIDO VALUES(5,3,5,'entregado',DEFAULT,
    TO_DATE('23/10/2022 13:10:00','DD/MM/YY HH24:MI:SS'));
INSERT INTO PEDIDO VALUES(6,null,6,DEFAULT,DEFAULT, null);
INSERT INTO PEDIDO VALUES(7,5,7,'entregado',DEFAULT,
    TO_DATE('24/10/2022 15:00:00','DD/MM/YY HH24:MI:SS'));
INSERT INTO PEDIDO VALUES(8,null,8,DEFAULT,DEFAULT, null);
INSERT INTO PEDIDO VALUES(9,5,9,'entregado',DEFAULT,
    TO_DATE('22/10/2022 17:00:00','DD/MM/YY HH24:MI:SS'));
INSERT INTO PEDIDO VALUES(10,5,10,'entregado',DEFAULT,
    TO_DATE('26/10/2022 12:30:00','DD/MM/YY HH24:MI:SS'));
INSERT INTO PEDIDO
    VALUES(11,7,8,'entregado',DEFAULT,TO_DATE('22/10/2022
    18:12:00','DD/MM/YY HH24:MI:SS'));
INSERT INTO PEDIDO VALUES(12,1,4,'entregado',TO_DATE('20/10/2022
    15:00:00','DD/MM/YY HH24:MI:SS'),TO_DATE('20/10/2022
    18:00:00','DD/MM/YY HH24:MI:SS'));
INSERT INTO PEDIDO VALUES(13,9,9,'entregado',TO_DATE('20/10/2022
    16:00:00','DD/MM/YY HH24:MI:SS'),TO_DATE('20/10/2022
    18:00:00','DD/MM/YY HH24:MI:SS'));
INSERT INTO PEDIDO VALUES(14,NULL,1,DEFAULT,DEFAULT,NULL);
INSERT INTO PEDIDO VALUES(15,9,2,'entregado',TO_DATE('21/10/2022
    14:00:00','DD/MM/YY HH24:MI:SS'),TO_DATE('22/10/2022
    16:00:00','DD/MM/YY HH24:MI:SS'));
INSERT INTO PEDIDO VALUES(16,4,8,'entregado',TO_DATE('22/10/2022
    16:00:00','DD/MM/YY HH24:MI:SS'),TO_DATE('22/10/2022
    17:00:00','DD/MM/YY HH24:MI:SS'));
INSERT INTO PEDIDO VALUES(17,NULL,7,DEFAULT,DEFAULT,NULL);
INSERT INTO PEDIDO VALUES(18,5,4,'entregado',TO_DATE('21/10/2022
    16:00:00','DD/MM/YY HH24:MI:SS'),TO_DATE('21/10/2022
    18:00:00','DD/MM/YY HH24:MI:SS'));
INSERT INTO PEDIDO VALUES(19,6,2,'entregado',TO_DATE('23/10/2022
    16:00:00','DD/MM/YY HH24:MI:SS'),TO_DATE('23/10/2022
    17:00:00','DD/MM/YY HH24:MI:SS'));
INSERT INTO PEDIDO VALUES(20,NULL,1,DEFAULT,DEFAULT,NULL);
COMMIT;

```

Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 20 en 0.002 segundos

	ID_PEDIDO	ID_REPARTIDOR	ID_CLIENTE	ESTADO	FECHA_PEDIDO	FECHA_ENTREGA
1	1	1	1	entregado	20/10/22	22/10/22
2	2	2	2	entregado	20/10/22	22/10/22
3	3	(null)	3	pendiente	20/10/22	(null)
4	4	3	3	entregado	20/10/22	22/10/22
5	5	3	3	entregado	20/10/22	23/10/22
6	6	(null)	6	pendiente	20/10/22	(null)
7	7	5	5	entregado	20/10/22	24/10/22
8	8	(null)	8	pendiente	20/10/22	(null)
9	9	5	5	entregado	20/10/22	22/10/22
10	10	5	5	entregado	20/10/22	26/10/22
11	11	7	7	entregado	20/10/22	22/10/22
12	12	1	1	entregado	20/10/22	20/10/22
13	13	9	9	entregado	20/10/22	20/10/22
14	14	(null)	1	pendiente	20/10/22	(null)
15	15	9	9	entregado	21/10/22	22/10/22
16	16	4	4	entregado	22/10/22	22/10/22
17	17	(null)	7	pendiente	20/10/22	(null)
18	18	5	5	entregado	21/10/22	21/10/22
19	19	6	6	entregado	23/10/22	23/10/22
20	20	(null)	1	pendiente	20/10/22	(null)

--DATOS DETALLE_PEDIDO

```
INSERT INTO DETALLE_PEDIDO VALUES (1,1,2,65.80);
INSERT INTO DETALLE_PEDIDO VALUES (1,11,3,16.20);
INSERT INTO DETALLE_PEDIDO VALUES (2,2,1,149.90);
INSERT INTO DETALLE_PEDIDO VALUES (2,12,2,9.80);
INSERT INTO DETALLE_PEDIDO VALUES (3,3,3,569.70);
INSERT INTO DETALLE_PEDIDO VALUES (3,13,4,51.60);
INSERT INTO DETALLE_PEDIDO VALUES (4,4,4,83.60);
INSERT INTO DETALLE_PEDIDO VALUES (4,14,2,21.80);
INSERT INTO DETALLE_PEDIDO VALUES (5,5,5,175);
INSERT INTO DETALLE_PEDIDO VALUES (5,15,1,8.50);
INSERT INTO DETALLE_PEDIDO VALUES (6,6,1,45.90);
INSERT INTO DETALLE_PEDIDO VALUES (6,16,3,37.50);
INSERT INTO DETALLE_PEDIDO VALUES (7,7,3,53.70);
INSERT INTO DETALLE_PEDIDO VALUES (7,17,2,25);
INSERT INTO DETALLE_PEDIDO VALUES (8,8,3,209.70);
INSERT INTO DETALLE_PEDIDO VALUES (8,18,1,6.90);
INSERT INTO DETALLE_PEDIDO VALUES (9,5,1,35);
INSERT INTO DETALLE_PEDIDO VALUES (9,17,1,12.50);
INSERT INTO DETALLE_PEDIDO VALUES (10,2,1,149.90);
INSERT INTO DETALLE_PEDIDO VALUES (10,14,4,43.60);
INSERT INTO DETALLE_PEDIDO VALUES (11,4,4,83.60);
INSERT INTO DETALLE_PEDIDO VALUES (11,12,5,24.50);
INSERT INTO DETALLE_PEDIDO VALUES (12,8,1,69.90);
INSERT INTO DETALLE_PEDIDO VALUES (12,16,2,25);
INSERT INTO DETALLE_PEDIDO VALUES (13,2,1,149.90);
INSERT INTO DETALLE_PEDIDO VALUES (13,17,1,12.50);
INSERT INTO DETALLE_PEDIDO VALUES (14,8,4,279.60);
INSERT INTO DETALLE_PEDIDO VALUES (14,19,3,15);
INSERT INTO DETALLE_PEDIDO VALUES (15,7,3,53.70);
INSERT INTO DETALLE_PEDIDO VALUES (15,16,3,37.50);
INSERT INTO DETALLE_PEDIDO VALUES (16,2,1,149.90);
INSERT INTO DETALLE_PEDIDO VALUES (16,20,3,19.50);
INSERT INTO DETALLE_PEDIDO VALUES (17,3,2,379.80);
INSERT INTO DETALLE_PEDIDO VALUES (17,18,3,20.70);
INSERT INTO DETALLE_PEDIDO VALUES (18,5,5,175);
INSERT INTO DETALLE_PEDIDO VALUES (18,12,2,9.80);
INSERT INTO DETALLE_PEDIDO VALUES (19,2,1,149.90);
INSERT INTO DETALLE_PEDIDO VALUES (19,19,3,15);
INSERT INTO DETALLE_PEDIDO VALUES (20,10,1,37.50);
INSERT INTO DETALLE_PEDIDO VALUES (20,20,1,6.50);
COMMIT;
```

Salida de Script x Resultado de la Consulta x				
SQL Todas las Filas Recuperadas: 40 en 0.007 segundos				
	ID_PEDIDO	ID_PRODUCTO	CANTIDAD	PRECIO_UNIDAD
14	7	17	2	25
15	8	8	3	209.7
16	8	18	1	6.9
17	9	5	1	35
18	9	17	1	12.5
19	10	2	1	149.9
20	10	14	4	43.6
21	11	4	4	83.6
22	11	12	5	24.5
23	12	8	1	69.9
24	12	16	2	25
25	13	2	1	149.9
26	13	17	1	12.5
27	14	8	4	279.6
28	14	19	3	15
29	15	7	3	53.7
30	15	16	3	37.5
31	16	2	1	149.9
32	16	20	3	19.5
33	17	3	2	379.8
34	17	18	3	20.7
35	18	5	5	175
36	18	12	2	6.8

b. Funciones

--Función Convertir Soles

```
CREATE OR REPLACE FUNCTION CONVERT_SOLES(F_MONEY NUMBER)
RETURN VARCHAR2
IS
BEGIN
    RETURN TO_CHAR(F_MONEY,'L999,999.99','nls_currency=S/');
END;
```

--Función Formato Fecha

```
CREATE OR REPLACE FUNCTION FORM_FECHA(F_FECHA DATE)
RETURN VARCHAR2
IS
BEGIN
    RETURN TO_CHAR(F_FECHA,'DD/MM/YYYY "a las" HH24:MI');
END;
```

--Función Nombre Mes

```
CREATE OR REPLACE FUNCTION NOMBRE_MES(P_NUMERO INT)
RETURN VARCHAR2
IS
BEGIN
    RETURN Rtrim(TO_CHAR(TO_DATE(P_NUMERO, 'MM'), 'MONTH'));
END;
```

END;

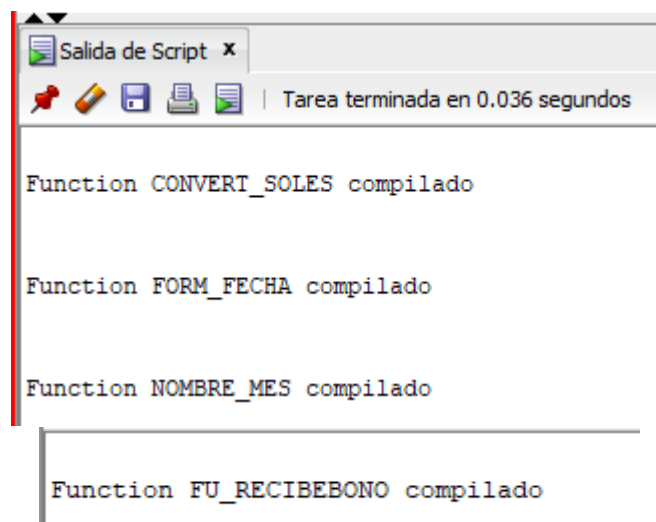
--Función Recibe Bono

```
CREATE OR REPLACE FUNCTION FU_recibebono(P_ID_REPARTIDOR
REPARTIDOR.ID_REPARTIDOR%TYPE,P_MONTH NUMBER)
RETURN VARCHAR2
IS
  V_RESPUESTA VARCHAR2(50);
  V_CANT_PEDIDO NUMBER:=0;
BEGIN
  SELECT COUNT(ID_REPARTIDOR)
  INTO V_CANT_PEDIDO
  FROM PEDIDO
  WHERE ID_REPARTIDOR=P_ID_REPARTIDOR AND EXTRACT(MONTH FROM
PEDIDO.FECHA_PEDIDO)=P_MONTH;
  IF V_CANT_PEDIDO >1 THEN
    V_RESPUESTA:='OBTIENE BONO';
  ELSE
    V_RESPUESTA:='NO OBTIENE BONO';
  END IF;
RETURN
  V_RESPUESTA;
END;
```

--FUNCION PRODUCTO VENTAS

```
CREATE OR REPLACE FUNCTION FU_PRODUCTO_VENTAS(P_ID INT, P_MONTH
NUMBER, P_YEAR NUMBER) RETURN NUMBER
IS
  V_CANT NUMBER:=0;
BEGIN
  SELECT SUM(PP.CANTIDAD)
  INTO V_CANT
  FROM PEDIDO p
  INNER JOIN DETALLE_PEDIDO PP ON P.ID_PEDIDO=PP.ID_PEDIDO
  WHERE PP.ID_PRODUCTO=P_ID AND EXTRACT(MONTH FROM
FECHA_ENTREGA)=P_MONTH AND EXTRACT(YEAR FROM
FECHA_ENTREGA)=P_YEAR;

RETURN
  V_CANT;
END;
```



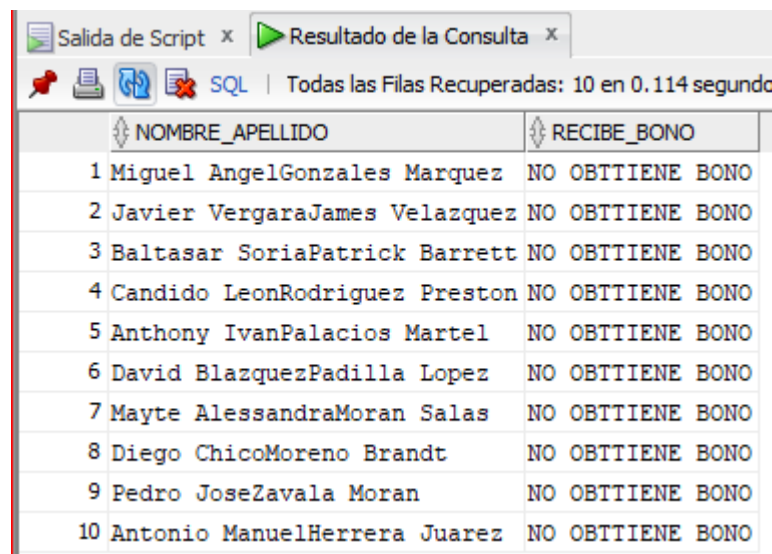
Function FU_PRODUCTO_VENTAS compilado

c. Vistas

--VISTA RECIBE BONO

```
CREATE OR REPLACE VIEW VW_APROVACION_BONO
AS
SELECT REPARTIDOR.nombres || REPARTIDOR.apellidos AS
NOMBRE_APELLIDO ,
FU_recibebono(ID_REPARTIDOR,EXTRACT(MONTH FROM SYSDATE)) AS
RECIBE_BONO
FROM REPARTIDOR;

--PRUEBA
SELECT *FROM VW_APROVACION_BONO;
```




	NOMBRE_APELLIDO	RECIBE_BONO
1	Miguel AngelGonzales Marquez	NO OBTIENE BONO
2	Javier VergaraJames Velazquez	NO OBTIENE BONO
3	Baltasar SoriaPatrick Barrett	NO OBTIENE BONO
4	Candido LeonRodriguez Preston	NO OBTIENE BONO
5	Anthony IvanPalacios Martel	NO OBTIENE BONO
6	David BlazquezPadilla Lopez	NO OBTIENE BONO
7	Mayte AlessandraMoran Salas	NO OBTIENE BONO
8	Diego ChicoMoreno Brandt	NO OBTIENE BONO
9	Pedro JoseZavala Moran	NO OBTIENE BONO
10	Antonio ManuelHerrera Juarez	NO OBTIENE BONO

--Vista reporte de productos

```
CREATE OR REPLACE VIEW VW_PRODUCTOS_CATALOGO
AS
SELECT P.ID_PRODUCTO, TP.DESCRIPCION, P.NOMBRE,
CONVERT_SOLES(P.PRECIO) AS
PRECIO
FROM PRODUCTO P
INNER JOIN TIPO_PRODUCTO TP
ON P.ID_TIPOPRODUCTO=TP.ID_TIPOPRODUCTO
ORDER BY P.ID_PRODUCTO ASC;

--PRUEBA
SELECT *FROM VW_PRODUCTOS_CATALOGO;
```

Salida de Script x Resultado de la Consulta x

 SQL | Todas las Filas Recuperadas: 21 en 0.04 segundos

	ID_PRODUCTO	DESCRIPCION	NOMBRE	PRECIO
1	1	Alcohol	Pisco Queirolo Quebranta	S/32.90
2	2	Alcohol	Whisky Black Label 750 ml	S/149.90
3	3	Alcohol	Ron Cartavio 750 ml	S/189.90
4	4	Alcohol	Cerveza Pilsen Six Pack	S/20.90
5	5	Alcohol	Cerveza Corona Six Pack	S/35.00
6	6	Alcohol	Vodka Absolut 750 ml	S/45.90
7	7	Alcohol	Vino Santiago Queirolo 750 ml	S/17.90
8	8	Alcohol	Jagermeister 700 ml	S/69.90
9	9	Alcohol	Cerveza Budweiser Six Pack	S/12.90
10	10	Alcohol	Tequila Jose Cuervo 375 ml	S/37.50
11	11	Snack	Camote Villa Natura x 150 g	S/5.40
12	12	Snack	Chifles V.Natura Salados 150 g	S/4.90
13	13	Snack	Piqueo Snax 280 Gr	S/12.90
14	14	Snack	Snack Doritos Dippas 200 g	S/10.90
15	15	Snack	Papas LAYS Clásicas 210 g	S/8.50
16	16	Refrescos	Coca Cola 3 Lt	S/12.50
17	17	Refrescos	Inka Kola 3 Lt	S/12.50
18	18	Refrescos	B.Evervess Carbonatada 1.5 Lt	S/6.90
19	19	Refrescos	Agua San Carlos sin Gas 3 Lt	S/5.00
20	20	Refrescos	Cifrut Citrus Punch 1.5 Lt	S/6.50
21	21	Refrescos	Monster Energy 473 ml	S/7.50

d. Procedimientos

--Imprimir un determinado pedido

```
CREATE OR REPLACE PROCEDURE SP_IMPRIME_PEDIDO(V_ID_PEDIDO
PEDIDO.ID_PEDIDO%TYPE)
IS
    CURSOR C_PEDIDO IS (
        SELECT          PR.NOMBRE,TP.DESCRIPCION,                PP.CANTIDAD,
        PP.PRECIO_UNIDAD
        FROM DETALLE_PEDIDO PP
        INNER JOIN PRODUCTO PR ON PP.ID_PRODUCTO=PR.ID_PRODUCTO
        INNER JOIN  tipo_producto TP ON pr.id_tipoproducto=
        tp.id_tipoproducto
        INNER JOIN PEDIDO P ON PP.ID_PEDIDO=P.ID_PEDIDO
        WHERE P.ID_PEDIDO = V_ID_PEDIDO);
    V_NRO_PEDIDO PEDIDO.ID_PEDIDO%TYPE;
    V_NAME_CLIENTE VARCHAR(50);
    V_ID_REPARTIDOR repartidor.id_repartidor%TYPE:=0;
    V_NAME_REPARTIDOR VARCHAR(50);
    V_FECHA_PEDIDO VARCHAR2(40);
    V_ESTADO_PEDIDO.ESTADO%TYPE;
    V_FECHA_ENTREGA VARCHAR2(40);
    V_SUBTOTAL NUMBER:=0;
    V_TOTALP NUMBER:=0;
BEGIN
    SELECT          P.ID_PEDIDO,P.ESTADO,                C.NOMBRES||'
'||C.APELLIDOS,NVL(p.id_repartidor,0)
```

```
, FORM_FECHA(P.FECHA_PEDIDO), FORM_FECHA(P.FECHA_ENTREGA)
INTO V_NRO_PEDIDO, V_ESTADO, V_NAME_CLIENTE, V_ID_REPARTIDOR,
V_FECHA_PEDIDO, V_FECHA_ENTREGA
FROM PEDIDO P
INNER JOIN CLIENTE C ON P.ID_CLIENTE=C.ID_CLIENTE
WHERE ID_PEDIDO = V_ID_PEDIDO;

IF(V_ID_REPARTIDOR=0) THEN
V_NAME_REPARTIDOR:=NULL;
ELSE
SELECT NOMBRES ||' ' || APELLIDOS
INTO V_NAME_REPARTIDOR
FROM REPARTIDOR
WHERE id_repartidor= V_ID_REPARTIDOR;
END IF;
DBMS_OUTPUT.PUT_LINE('');
DBMS_OUTPUT.PUT_LINE(LPAD('La casa del Loquor',45,' '));
DBMS_OUTPUT.PUT_LINE(LPAD('Av Jose Carlos Mariategui Mz E Lote
7',55,' '));

DBMS_OUTPUT.PUT_LINE('=====
=====');
DBMS_OUTPUT.PUT_LINE(LPAD('NRO PEDIDO: ' ||V_ID_PEDIDO,71,'
'));
DBMS_OUTPUT.PUT_LINE('NOMBRE DE CLIENTE: ' ||V_NAME_CLIENTE);
DBMS_OUTPUT.PUT_LINE(RPAD('FECHA PEDIDO: ' ||
V_FECHA_PEDIDO,56) || 'ESTADO: ' ||v_estado);
DBMS_OUTPUT.PUT_LINE(RPAD('REPARTIDOR: ' ||
nvl(V_NAME_REPARTIDOR,'NO ASIGNADO'),45));
DBMS_OUTPUT.PUT_LINE(RPAD('FECHA ENTREGA: ' ||
nvl(V_FECHA_ENTREGA,'NO ENTREGADO'),45));

DBMS_OUTPUT.PUT_LINE('=====
=====');
DBMS_OUTPUT.PUT_LINE(LPAD('-----',40,' '));
DBMS_OUTPUT.PUT_LINE(LPAD('DETALLES DEL PEDIDO',40,' '));
DBMS_OUTPUT.PUT_LINE(LPAD('-----',40,' '));

DBMS_OUTPUT.PUT_LINE(' ' ||'PRODUCTO CANTIDAD
TIPO c/u SUBTOTAL P');
DBMS_OUTPUT.PUT_LINE(' ' ||'----- ---
-----');
FOR AUX_P IN C_PEDIDO LOOP
V_SUBTOTAL:=AUX_P.CANTIDAD*AUX_P.PRECIO_UNIDAD;
DBMS_OUTPUT.PUT_LINE(' ' ||RPAD(AUX_P.NOMBRE,35,' ') ||
RPAD(AUX_P.CANTIDAD,6,' ') ||RPAD(AUX_P.DESCRIPCION,10,' ') ||
LPAD(CONVERT_SOLES(AUX_P.PRECIO_UNIDAD),5,'
') || LPAD(CONVERT_SOLES(V_SUBTOTAL),13,' '));
V_TOTALP:= V_TOTALP + V_SUBTOTAL;
END LOOP;

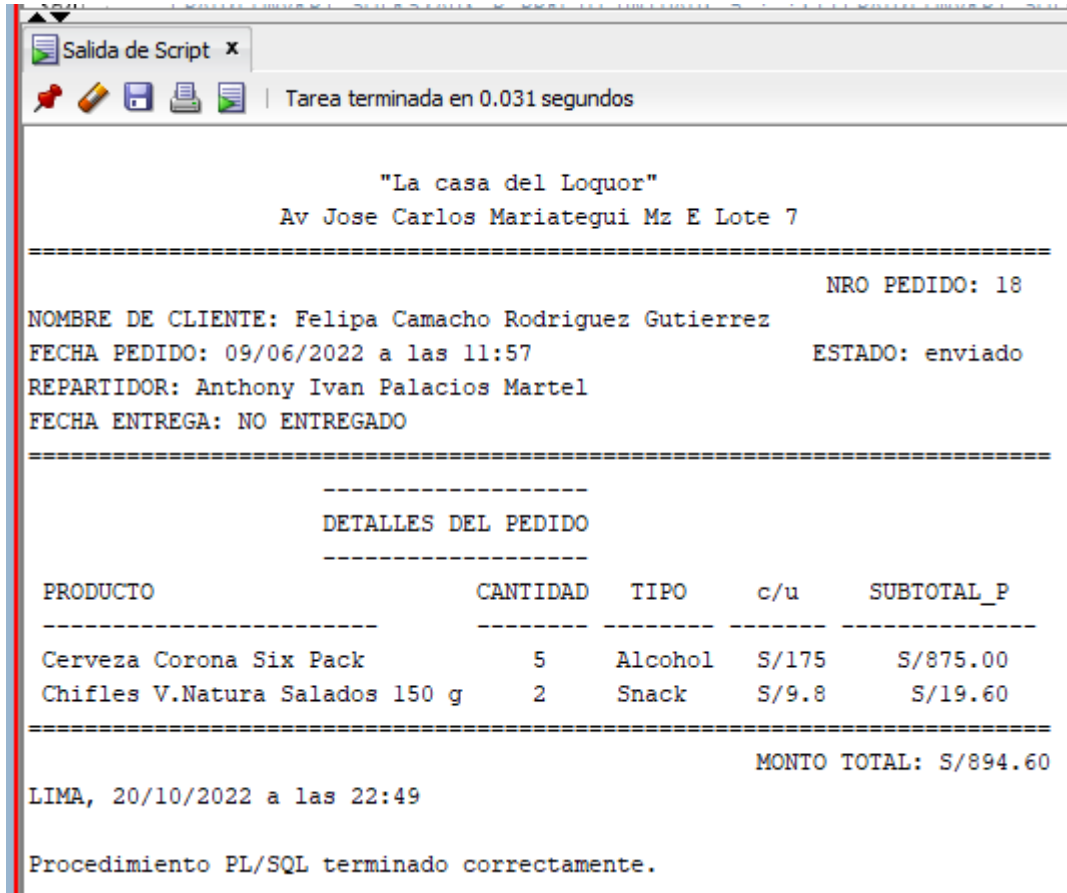
DBMS_OUTPUT.PUT_LINE('=====
=====');
DBMS_OUTPUT.PUT_LINE(LPAD('MONTO TOTAL:
' ||CONVERT_SOLES(V_TOTALP),73,' '));
DBMS_OUTPUT.PUT_LINE('LIMA, ' ||FORM_FECHA(SYSDATE));
EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('PEDIDO NO ENCONTRADO');
```

```

WHEN OTHERS THEN
  DBMS_OUTPUT.PUT_LINE ('OCURRIO UN ERROR '||SQLERRM);
END;

--PRUEBA
EXECUTE SP_IMPRIME_PEDIDO(18);

```



Salida de Script x | Tarea terminada en 0.031 segundos

```

"La casa del Loquor"
Av Jose Carlos Mariategui Mz E Lote 7
=====
NRO PEDIDO: 18
NOMBRE DE CLIENTE: Felipa Camacho Rodriguez Gutierrez
FECHA PEDIDO: 09/06/2022 a las 11:57 ESTADO: enviado
REPARTIDOR: Anthony Ivan Palacios Martel
FECHA ENTREGA: NO ENTREGADO
=====
DETALLES DEL PEDIDO
=====
PRODUCTO                                CANTIDAD  TIPO    c/u    SUBTOTAL_P
-----
Cerveza Corona Six Pack                 5         Alcohol S/175  S/875.00
Chifles V.Natura Salados 150 g          2         Snack   S/9.8   S/19.60
=====
MONTO TOTAL: S/894.60
LIMA, 20/10/2022 a las 22:49

Procedimiento PL/SQL terminado correctamente.

```

--Procedimiento de reporte de venta de un mes

```
CREATE OR REPLACE PROCEDURE PR_REPORTER_VENTA(P_MONTH NUMBER,
P_YEAR NUMBER)
IS
    CURSOR C_VENTAS IS SELECT R.NOMBRES ||' '|| R.APELLIDOS AS
NOMBRE_COMPLETO ,
    P.ID_PEDIDO, C.NOMBRES ||' '|| C.APELLIDOS AS NOMBRE_CLIENTE,
    SUM(PP.CANTIDAD*PP.PRECIO_UNIDAD) AS MONTO_TOTAL
    FROM REPARTIDOR R
    INNER JOIN PEDIDO P ON R.ID_REPARTIDOR=P.ID_REPARTIDOR
    INNER JOIN CLIENTE C ON P.ID_CLIENTE=C.ID_CLIENTE
    INNER JOIN DETALLE_PEDIDO PP ON P.ID_PEDIDO=PP.ID_PEDIDO
    WHERE EXTRACT(MONTH FROM P.FECHA_ENTREGA)=P_MONTH AND
EXTRACT(YEAR FROM
    P.FECHA_ENTREGA)=P_YEAR
    GROUP BY R.NOMBRES||' '|| R.APELLIDOS, P.ID_PEDIDO, C.NOMBRES
    ||' '||
    C.APELLIDOS;
    V_SUMA NUMBER:=0;
BEGIN

    dbms_output.put_line('=====REPORTER
DE VENTAS=====');
    dbms_output.put_line(LPAD('MES: '||NOMBRE_MES(P_MONTH),30,'
')||LPAD('AÑO: '||P_YEAR,30,' '));

    dbms_output.put_line('=====
=====');
    dbms_output.put_line(RPAD('NOMBRE DE REPARTIDOR',31,'
')||RPAD('NRO PEDIDO',15,' ')||RPAD('NOMBRE CLIENTE',35,'
')||'MONTO');
    dbms_output.put_line(RPAD('-----',31,'
')||RPAD('-----',15,' ')||RPAD('-----',35,'
')||'-----');
    FOR AUX IN C_VENTAS LOOP
        V_SUMA:=V_SUMA+AUX.MONTO_TOTAL;
        DBMS_OUTPUT.PUT_LINE(RPAD(AUX.NOMBRE_COMPLETO,35,' ') ||
        RPAD(AUX.ID_PEDIDO,11,' ') || RPAD(AUX.NOMBRE_CLIENTE,35,' ')
        ||
        CONVERT_SALES(AUX.MONTO_TOTAL));
    END LOOP;

    dbms_output.put_line('=====
=====');
    dbms_output.put_line('MONTO GENERAL: '||
CONVERT_SALES(V_SUMA));
    dbms_output.put_line('FECHA DE EJECUCIÓN: '||
to_char(sysdate,'dd "de" Month"del" YYYY'));
exception
    when no_data_found then
        dbms_output.put_line('REGISTROS NO ENCONTRADOS');
END;
--PRUEBA
EXECUTE PR_REPORTER_VENTA(10,2022);
```


=====REPORTE DE VENTAS=====			
MES: OCTUBRE		AÑO: 2022	
NOMBRE DE REAPRTIDOR	NRO PEDIDO	NOMBRE CLIENTE	MONTO
Javier Vergara James Velazquez	2	Pascual Bolaños James Sampson	S/169.50
Baltasar Soria Patrick Barrett	4	Felipa Camacho Rodriguez Gutierrez	S/378.00
Candido Leon Rodriguez Preston	16	Adam Cid Vega Brandt	S/208.40
Anthony Ivan Palacios Martel	10	Ruben Sales Ayala Juarez	S/324.30
Miguel Angel Gonzales Marquez	1	Nayara Gallardo Kelley Marquez	S/180.20
Baltasar Soria Patrick Barrett	5	Virgilio Tejedor Ewing Villanueva	S/883.50
Anthony Ivan Palacios Martel	18	Felipa Camacho Rodriguez Gutierrez	S/894.60
David Blazquez Padilla Lopez	19	Pascual Bolaños James Sampson	S/194.90
Pedro Jose Zavala Moran	13	Marcelina Gracia Zavala Pacheco	S/162.40
Anthony Ivan Palacios Martel	7	Lazaro Latorre Moran Marquez	S/211.10
Mayte Alessandra Moran Salas	11	Adam Cid Vega Brandt	S/456.90
Pedro Jose Zavala Moran	15	Pascual Bolaños James Sampson	S/273.60
Miguel Angel Gonzales Marquez	12	Felipa Camacho Rodriguez Gutierrez	S/119.90
Anthony Ivan Palacios Martel	9	Marcelina Gracia Zavala Pacheco	S/47.50
=====			
MONTO GENERAL: S/4,504.80			
FECHA DE EJECUCIÓN: 03 de Noviembre del 2022			
Procedimiento PL/SQL terminado correctamente.			

e. Paquetes

--Reporte de Productos

```

CREATE OR REPLACE PACKAGE PKG_REPORTES_PRODUCTOS
AS
    PROCEDURE SP_VENTAS_ALCOHOL(V_MONTH NUMBER, V_YEAR NUMBER);
    PROCEDURE SP_VENTAS_SNACK(V_MONTH NUMBER, V_YEAR NUMBER);
    PROCEDURE SP_VENTAS_REFRESCOS(V_MONTH NUMBER, V_YEAR NUMBER);
    PROCEDURE SP_PRODUCTOS_MAS_VENDIDOS(V_MONTH NUMBER, V_YEAR
NUMBER);

END PKG_REPORTES_PRODUCTOS;

CREATE OR REPLACE PACKAGE BODY PKG_REPORTES_PRODUCTOS
AS
    FUNCTION FU_PRODUCTO_VENTA_MES(P_ID INT, P_MONTH NUMBER, P_YEAR
NUMBER) RETURN
NUMBER
    IS
        V_CANT NUMBER:=0;
    BEGIN
        SELECT SUM(PP.CANTIDAD)
        INTO V_CANT
        FROM PEDIDO P
        INNER JOIN DETALLE_PEDIDO PP ON P.ID_PEDIDO=PP.ID_PEDIDO
        WHERE PP.ID_PRODUCTO=P_ID AND EXTRACT(MONTH FROM
FECHA_ENTREGA)=P_MONTH AND
EXTRACT(YEAR FROM FECHA_ENTREGA)=P_YEAR;
        RETURN
        V_CANT;
    END;

```

--PROCEDIMIENTO VENTAS ALCOHOL

```

PROCEDURE SP_VENTAS_ALCOHOL(V_MONTH NUMBER, V_YEAR NUMBER)
IS
CURSOR C_ALCOHOL IS(
SELECT ID_PRODUCTO, NOMBRE
FROM PRODUCTO
WHERE ID_TIPOPRODUCTO=1
);
BEGIN
DBMS_OUTPUT.PUT_LINE('REPORTE DE VENTAS (ALCOHOL) DEL MES
'||NOMBRE_MES(V_MONTH)||' AÑO '||V_YEAR);

DBMS_OUTPUT.PUT_LINE('=====
=====');
DBMS_OUTPUT.PUT_LINE(RPAD('ID',5,' ')||RPAD('NOMBRE
PRODUCTO',28,' ')||'CANTIDAD');
FOR PL IN C_ALCOHOL LOOP
DBMS_OUTPUT.PUT_LINE(RPAD(PL.ID_PRODUCTO,5,'
')||RPAD(PL.NOMBRE,32,'
')||NVL(FU_PRODUCTO_VENTA_MES(PL.ID_PRODUCTO,V_MONTH,V_YEAR),0))
;
END LOOP;
END;
```

--PROCEDIMIENTO VENTAS SNACK

```

PROCEDURE SP_VENTAS_SNACK(V_MONTH NUMBER, V_YEAR NUMBER)
IS
CURSOR C_SNACK IS(
SELECT ID_PRODUCTO, NOMBRE
FROM PRODUCTO
WHERE ID_TIPOPRODUCTO=2
);
BEGIN
DBMS_OUTPUT.PUT_LINE('REPORTE DE VENTAS (SNACK) DEL MES
'||NOMBRE_MES(V_MONTH)||' AÑO '||V_YEAR);

DBMS_OUTPUT.PUT_LINE('=====
=====');
DBMS_OUTPUT.PUT_LINE(RPAD('ID',5,' ')||RPAD('NOMBRE
PRODUCTO',28,' ')||'CANTIDAD');
FOR PL IN C_SNACK LOOP
DBMS_OUTPUT.PUT_LINE(RPAD(PL.ID_PRODUCTO,5,'
')||RPAD(PL.NOMBRE,32,'
')||NVL(FU_PRODUCTO_VENTA_MES(PL.ID_PRODUCTO,V_MONTH,V_YEAR),0))
;
END LOOP;
END;
```

--PROCEDIMIENTO VENTAS REFRESCOS

```

PROCEDURE SP_VENTAS_REFRESCOS(V_MONTH NUMBER, V_YEAR NUMBER)
IS
CURSOR C_REFRESCOS IS(
SELECT ID_PRODUCTO, NOMBRE
FROM PRODUCTO
WHERE ID_TIPOPRODUCTO=3
);
BEGIN
DBMS_OUTPUT.PUT_LINE('REPORTE DE VENTAS (REFRESCOS) DEL MES
'||NOMBRE_MES(V_MONTH)||' AÑO '||V_YEAR);
```

```
DBMS_OUTPUT.PUT_LINE('=====');
=====');
    DBMS_OUTPUT.PUT_LINE(RPAD('ID',5,' ') || RPAD('NOMBRE
PRODUCTO',28,' ') || 'CANTIDAD');
    FOR PL IN C_REFRESCOS LOOP
        DBMS_OUTPUT.PUT_LINE(RPAD(PL.ID_PRODUCTO,5,'
') || RPAD(PL.NOMBRE,32,'
') || NVL(FU_PRODUCTO_VENTA_MES(PL.ID_PRODUCTO,V_MONTH,V_YEAR),0))
    ;
    END LOOP;
END;
```

--LISTADO DE PRODUCTOS

```
PROCEDURE SP_PRODUCTOS_MAS_VENDIDOS(V_MONTH NUMBER, V_YEAR
NUMBER)
IS
```

```
    CURSOR C_ALCOHOL IS
    SELECT
    ID_PRODUCTO,NOMBRE,PRECIO,NVL(FU_PRODUCTO_VENTAS(ID_PRODUCTO,V_M
ONTH,V_YEAR),0) AS CANTIDAD
    FROM PRODUCTO;
```

```
    CURSOR C_SNACK IS
    SELECT
    ID_PRODUCTO,NOMBRE,PRECIO,NVL(FU_PRODUCTO_VENTAS(ID_PRODUCTO,V_M
ONTH,V_YEAR),0) AS CANTIDAD
    FROM PRODUCTO;
```

```
    CURSOR C_REFRESCOS IS
    SELECT
    ID_PRODUCTO,NOMBRE,PRECIO,NVL(FU_PRODUCTO_VENTAS(ID_PRODUCTO,V_M
ONTH,V_YEAR),0) AS CANTIDAD
    FROM PRODUCTO;
```

```
    T_ALCOHOL C_ALCOHOL%ROWTYPE;
    T_SNACK C_SNACK%ROWTYPE;
    T_REFRESCOS C_REFRESCOS%ROWTYPE;
```

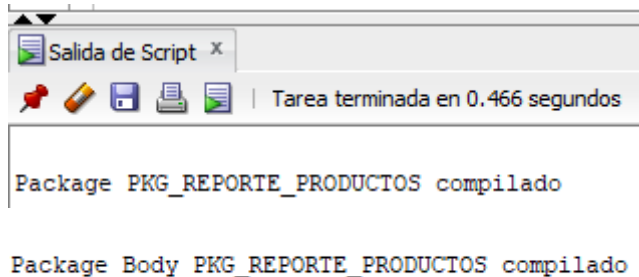
```
BEGIN
    SELECT
    ID_PRODUCTO, NOMBRE,
    PRECIO,NVL(FU_PRODUCTO_VENTAS(ID_PRODUCTO,V_MONTH,V_YEAR),0) AS
    CANTIDAD
    INTO T_ALCOHOL
    FROM PRODUCTO
    WHERE ID_TIPOPRODUCTO=1
    ORDER BY CANTIDAD DESC
    FETCH FIRST 1 ROWS ONLY;
```

```
    SELECT
    ID_PRODUCTO, NOMBRE,
    PRECIO,NVL(FU_PRODUCTO_VENTAS(ID_PRODUCTO,V_MONTH,V_YEAR),0) AS
    CANTIDAD
    INTO T_SNACK
    FROM PRODUCTO
    WHERE ID_TIPOPRODUCTO=2
    ORDER BY CANTIDAD DESC
    FETCH FIRST 1 ROWS ONLY;
```

```
    SELECT
    ID_PRODUCTO, NOMBRE,
    PRECIO,NVL(FU_PRODUCTO_VENTAS(ID_PRODUCTO,V_MONTH,V_YEAR),0) AS
    CANTIDAD
    INTO T_REFRESCOS
```

```
FROM PRODUCTO
WHERE ID_TIPOPRODUCTO=3
ORDER BY CANTIDAD DESC
FETCH FIRST 1 ROWS ONLY;
```

```
DBMS_OUTPUT.PUT_LINE('PRODUCTOS MAS VENDIDOS');
DBMS_OUTPUT.PUT_LINE('=====');
DBMS_OUTPUT.PUT_LINE('CATEGORIA ALCOHOL');
DBMS_OUTPUT.PUT_LINE('-----');
DBMS_OUTPUT.PUT_LINE('ID PRODUCTO: '||T_ALCOHOL.ID_PRODUCTO);
DBMS_OUTPUT.PUT_LINE('NOMBRE: '||T_ALCOHOL.NOMBRE);
DBMS_OUTPUT.PUT_LINE('PRECIO:
'||CONVERT_SOLES(T_ALCOHOL.PRECIO));
DBMS_OUTPUT.PUT_LINE('CATEGORIA SNACK');
DBMS_OUTPUT.PUT_LINE('-----');
DBMS_OUTPUT.PUT_LINE('ID PRODUCTO: '||T_SNACK.ID_PRODUCTO);
DBMS_OUTPUT.PUT_LINE('NOMBRE: '||T_SNACK.NOMBRE);
DBMS_OUTPUT.PUT_LINE('PRECIO:
'||CONVERT_SOLES(T_SNACK.PRECIO));
DBMS_OUTPUT.PUT_LINE('CATEGORIA REFRESCOS');
DBMS_OUTPUT.PUT_LINE('-----');
DBMS_OUTPUT.PUT_LINE('ID PRODUCTO: '||T_REFRESCOS.ID_PRODUCTO);
DBMS_OUTPUT.PUT_LINE('NOMBRE: '||T_REFRESCOS.NOMBRE);
DBMS_OUTPUT.PUT_LINE('PRECIO:
'||CONVERT_SOLES(T_REFRESCOS.PRECIO));
END;
END PKG_REPORTE_PRODUCTOS;
```

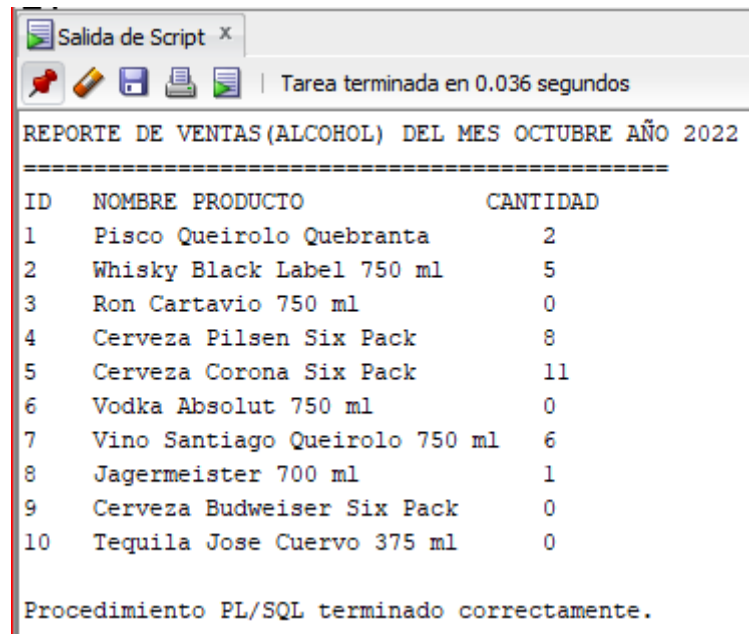


--ALCOHOL

BEGIN

PKG_REPORTE_PRODUCTOS.SP_VENTAS_ALCOHOL(6,2022);

END;



Salida de Script x

Tarea terminada en 0.036 segundos

REPORTE DE VENTAS (ALCOHOL) DEL MES OCTUBRE AÑO 2022

=====

ID	NOMBRE PRODUCTO	CANTIDAD
1	Pisco Queirolo Quebranta	2
2	Whisky Black Label 750 ml	5
3	Ron Cartavio 750 ml	0
4	Cerveza Pilsen Six Pack	8
5	Cerveza Corona Six Pack	11
6	Vodka Absolut 750 ml	0
7	Vino Santiago Queirolo 750 ml	6
8	Jagermeister 700 ml	1
9	Cerveza Budweiser Six Pack	0
10	Tequila Jose Cuervo 375 ml	0

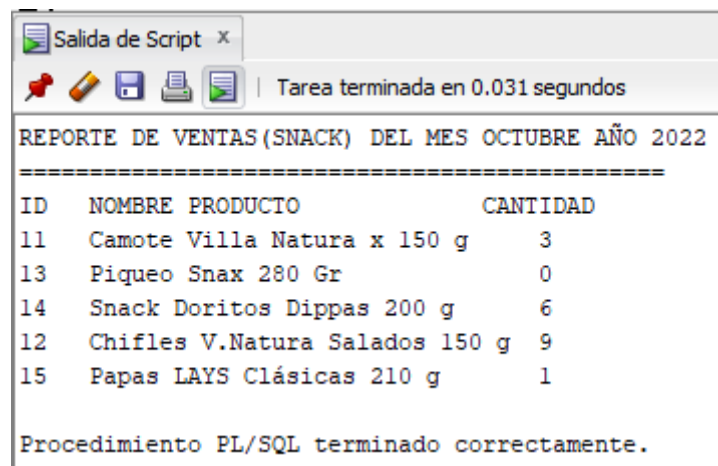
Procedimiento PL/SQL terminado correctamente.

--SNACK

BEGIN

PKG_REPORTE_PRODUCTOS.SP_VENTAS_SNACK(6,2022);

END;



Salida de Script x

Tarea terminada en 0.031 segundos

REPORTE DE VENTAS (SNACK) DEL MES OCTUBRE AÑO 2022

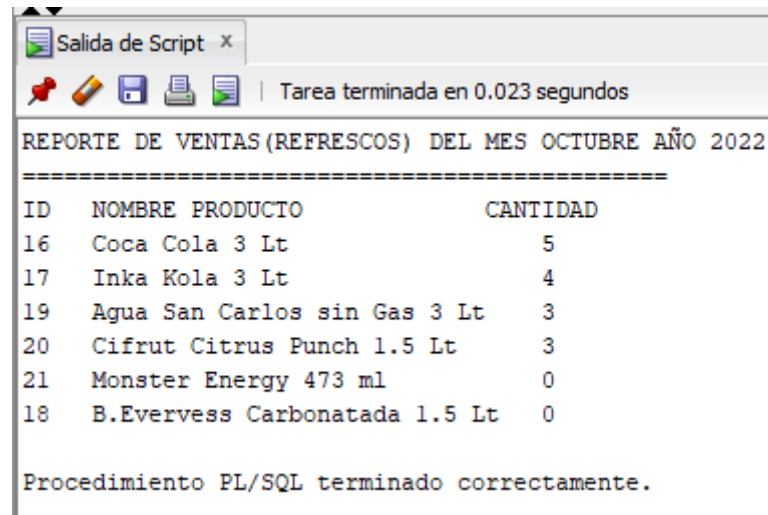
=====

ID	NOMBRE PRODUCTO	CANTIDAD
11	Camote Villa Natura x 150 g	3
13	Piqueo Snax 280 Gr	0
14	Snack Doritos Dippas 200 g	6
12	Chifles V.Natura Salados 150 g	9
15	Papas LAYS Clásicas 210 g	1

Procedimiento PL/SQL terminado correctamente.

--REFRESCOS

```
BEGIN
    PKG_REPORTE_PRODUCTOS.SP_VENTAS_REFRESCOS(6,2022);
END;
```



Salida de Script x | Tarea terminada en 0.023 segundos


REPORTE DE VENTAS(REFRESCOS) DEL MES OCTUBRE AÑO 2022

ID	NOMBRE PRODUCTO	CANTIDAD
16	Coca Cola 3 Lt	5
17	Inka Kola 3 Lt	4
19	Agua San Carlos sin Gas 3 Lt	3
20	Cifrut Citrus Punch 1.5 Lt	3
21	Monster Energy 473 ml	0
18	B.Evervess Carbonatada 1.5 Lt	0

Procedimiento PL/SQL terminado correctamente.

--MAS VENDIDOS

```
BEGIN
    PKG_REPORTE_PRODUCTOS.SP_PRODUCTOS_MAS_VENDIDOS(6,2022);
END;
```



Salida de Script x | Tarea terminada en 0.036 segundos

PRODUCTOS MAS VENDIDOS

CATEGORIA ALCOHOL

ID PRODUCTO: 5
NOMBRE: Cerveza Corona Six Pack
PRECIO: S/35.00

CATEGORIA SNACK

ID PRODUCTO: 12
NOMBRE: Chifles V.Natura Salados 150 g
PRECIO: S/4.90

CATEGORIA REFRESCOS

ID PRODUCTO: 16
NOMBRE: Coca Cola 3 Lt
PRECIO: S/12.50

Procedimiento PL/SQL terminado correctamente.

f. Trigger

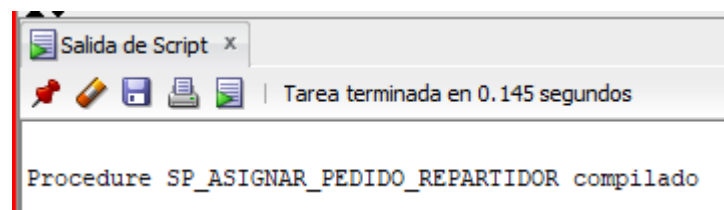
Procedimientos utilizados en conjunto con los trigger

--Procedimiento Asignar repartidor a un pedido

```
CREATE OR REPLACE PROCEDURE
SP_ASIGNAR_PEDIDO_REPARTIDOR(P_ID_REPARTIDOR INT,P_ID_PEDIDO
INT,P_MENSAJE OUT VARCHAR2)
IS
    V_ESTADO_RE REPARTIDOR.ESTADO%TYPE;
    V_ESTADO_P VARCHAR2(10);
    V_EXC_DISPONIBILIDAD EXCEPTION;
    V_EXC_PE_ESTADO EXCEPTION;
BEGIN
    SELECT ESTADO
    INTO V_ESTADO_RE --OBTENER ESTADO DEL REPARTIDOR
    FROM REPARTIDOR
    WHERE ID_REPARTIDOR = P_ID_REPARTIDOR;

    SELECT ESTADO
    INTO V_ESTADO_P --VERIFICAR SI YA ESTA
    FROM PEDIDO
    WHERE ID_PEDIDO = P_ID_PEDIDO;

    IF V_ESTADO_P = 'enviado' OR V_ESTADO_P = 'entregado' THEN
        RAISE V_EXC_PE_ESTADO;
    ELSIF V_ESTADO_RE='ocupado' then
        RAISE V_EXC_DISPONIBILIDAD;
    ELSE
        UPDATE PEDIDO
        SET ID_REPARTIDOR = P_ID_REPARTIDOR
        WHERE ID_PEDIDO = P_ID_PEDIDO;
        COMMIT;
        P_MENSAJE:='SE ASIGNO EL REPARTIDOR AL PEDIDO '
        ||P_ID_PEDIDO ;
    END IF;
EXCEPTION
    WHEN V_EXC_PE_ESTADO THEN
        P_MENSAJE:='PEDIDO YA DESPACHADO';
    WHEN V_EXC_DISPONIBILIDAD THEN
        P_MENSAJE:='REPARTIDOR NO DISPONIBLE';
    WHEN OTHERS THEN
        P_MENSAJE := 'ERROR ENCONTRADO: '||SQLCODE||' Mensaje:
        '||SQLERRM;
END;
```

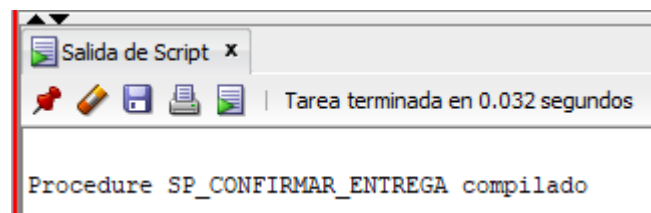


--Procedimiento para confirmar entrega

```
CREATE OR REPLACE PROCEDURE SP_CONFIRMAR_ENTREGA(P_ID_PEDIDO
INT,P_MENSAJE OUT VARCHAR2)
IS
    V_ESTADO_R REPARTIDOR.ESTADO%TYPE;
    V_ESTADO_P PEDIDO.ESTADO%TYPE;
    V_EXC_NO_CONF EXCEPTION;
```

```
BEGIN
  SELECT R.ESTADO, P.ESTADO
  INTO V_ESTADO_R,V_ESTADO_P
  FROM PEDIDO P
  INNER JOIN REPARTIDOR R ON P.ID_REPARTIDOR=r.id_repartidor
  WHERE P.ID_PEDIDO=P_ID_PEDIDO;

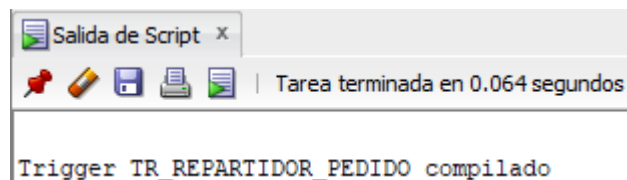
  IF (V_ESTADO_R='ocupado' and V_ESTADO_P='enviado' ) then
    UPDATE PEDIDO
    SET FECHA_ENTREGA=SYSDATE
    WHERE ID_PEDIDO=P_ID_PEDIDO;
    COMMIT;
    P_MENSAJE:='CONFIRMADA ENTREGA';
  ELSE
    RAISE V_EXC_NO_CONF;
  END IF;
EXCEPTION
  WHEN V_EXC_NO_CONF THEN
    P_MENSAJE:='IMPOSIBLE CONFIRMACION';
  WHEN OTHERS THEN
    P_MENSAJE := 'ERROR ENCONTRADO: '||SQLCODE||' Mensaje:
    '||SQLERRM;
END;
```



Trigger disparados por los procedimientos

--Disparador para cambiar estado del repartidor a ocupado y el estado del pedido a enviado

```
CREATE OR REPLACE TRIGGER TR_REPARTIDOR_PEDIDO
BEFORE UPDATE OF ID_REPARTIDOR ON PEDIDO
FOR EACH ROW
DECLARE
BEGIN
  :NEW.ESTADO:='enviado';
  UPDATE REPARTIDOR
  SET ESTADO='ocupado'
  where ID_REPARTIDOR =:NEW.ID_REPARTIDOR;
END;
```



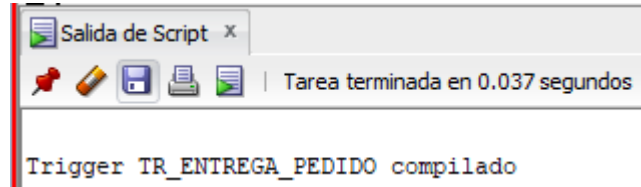
--Disparador para cambiar estado del repartidor a disponible y el estado del pedido a entregado

```
CREATE OR REPLACE TRIGGER TR_ENTREGA_PEDIDO
BEFORE UPDATE OF FECHA_ENTREGA ON PEDIDO
FOR EACH ROW
BEGIN
  IF :OLD.ID_REPARTIDOR IS NULL THEN
```



```

        RAISE_APPLICATION_ERROR(-20000, 'ERROR...EL PEDIDO NO HA
SIDO TOMADO POR UN REPARTIDOR');
    ELSE
        :NEW.ESTADO:='entregado';
        UPDATE REPARTIDOR
        SET ESTADO='disponible'
        where ID_REPARTIDOR =:NEW.ID_REPARTIDOR;
    END IF;
END;
```

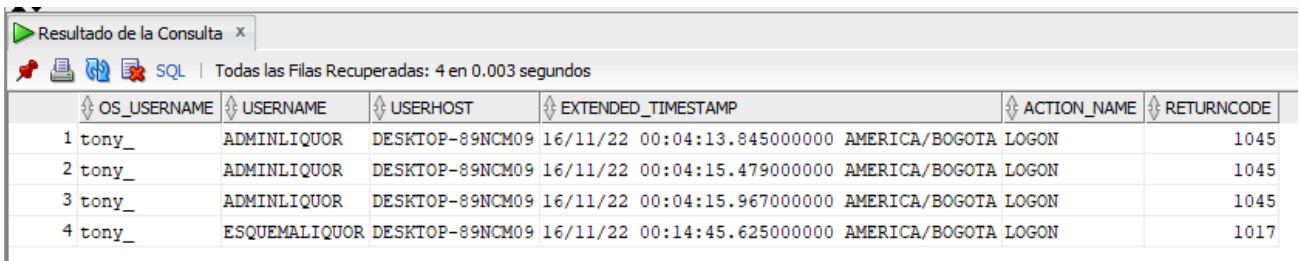


g. Auditorias

--Auditar sesiones fallidas.

```

AUDIT SESSION BY USERLIQUOR, ESQUEMALIQUOR, ADMINLIQUOR whenever
not successful;
```



OS_USERNAME	USERNAME	USERHOST	EXTENDED_TIMESTAMP	ACTION_NAME	RETURNCODE
1 tony_	ADMINLIQUOR	DESKTOP-89NCM09	16/11/22 00:04:13.845000000 AMERICA/BOGOTA	LOGON	1045
2 tony_	ADMINLIQUOR	DESKTOP-89NCM09	16/11/22 00:04:15.479000000 AMERICA/BOGOTA	LOGON	1045
3 tony_	ADMINLIQUOR	DESKTOP-89NCM09	16/11/22 00:04:15.967000000 AMERICA/BOGOTA	LOGON	1045
4 tony_	ESQUEMALIQUOR	DESKTOP-89NCM09	16/11/22 00:14:45.625000000 AMERICA/BOGOTA	LOGON	1017

--Auditoria de select fallidos por todos los usuarios.

```

AUDIT SELECT ON ESQUEMALIQUOR.PRODUCTO WHENEVER NOT SUCCESSFUL;
AUDIT SELECT ON ESQUEMALIQUOR.PEDIDO WHENEVER NOT SUCCESSFUL;
```

Audit correcto.

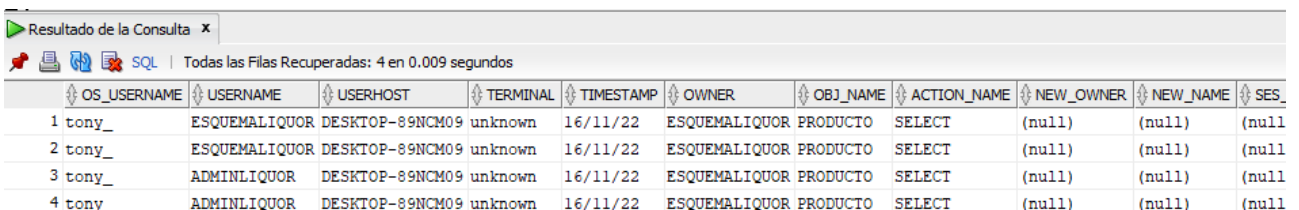
Audit correcto.

```

SELECT ID_TIPO FROM ESQUEMALIQUOR.producto;
```

```

SELECT *FROM DBA_AUDIT_OBJECT;
```



OS_USERNAME	USERNAME	USERHOST	TERMINAL	TIMESTAMP	OWNER	OBJ_NAME	ACTION_NAME	NEW_OWNER	NEW_NAME	SES_
1 tony_	ESQUEMALIQUOR	DESKTOP-89NCM09	unknown	16/11/22	ESQUEMALIQUOR	PRODUCTO	SELECT	(null)	(null)	(null)
2 tony_	ESQUEMALIQUOR	DESKTOP-89NCM09	unknown	16/11/22	ESQUEMALIQUOR	PRODUCTO	SELECT	(null)	(null)	(null)
3 tony_	ADMINLIQUOR	DESKTOP-89NCM09	unknown	16/11/22	ESQUEMALIQUOR	PRODUCTO	SELECT	(null)	(null)	(null)
4 tony_	ADMINLIQUOR	DESKTOP-89NCM09	unknown	16/11/22	ESQUEMALIQUOR	PRODUCTO	SELECT	(null)	(null)	(null)

--Política de auditoría de acciones DML de los usuarios con rol DBA

```
CREATE AUDIT POLICY POL_DML_LIQUOR
ACTIONS
    DELETE ON ESQUEMALIQUOR.PEDIDO,
    INSERT ON ESQUEMALIQUOR.PEDIDO,
    UPDATE ON ESQUEMALIQUOR.PEDIDO,
    DELETE ON ESQUEMALIQUOR.CLIENTE,
    INSERT ON ESQUEMALIQUOR.CLIENTE,
    UPDATE ON ESQUEMALIQUOR.CLIENTE,
    DELETE ON ESQUEMALIQUOR.DETALLE_PEDIDO,
    INSERT ON ESQUEMALIQUOR.DETALLE_PEDIDO,
    UPDATE ON ESQUEMALIQUOR.DETALLE_PEDIDO,
    DELETE ON ESQUEMALIQUOR.PRODUCTO,
    INSERT ON ESQUEMALIQUOR.PRODUCTO,
    UPDATE ON ESQUEMALIQUOR.PRODUCTO,
    DELETE ON ESQUEMALIQUOR.REPARTIDOR,
    INSERT ON ESQUEMALIQUOR.REPARTIDOR,
    UPDATE ON ESQUEMALIQUOR.REPARTIDOR,
    DELETE ON ESQUEMALIQUOR.TIPO_PRODUCTO,
    INSERT ON ESQUEMALIQUOR.TIPO_PRODUCTO,
    UPDATE ON ESQUEMALIQUOR.TIPO_PRODUCTO;
```

```
AUDIT POLICY POL_DML_LIQUOR BY USERS WITH GRANTED ROLES DBA;
```

```
SELECT *FROM AUDIT_UNIFIED_POLICIES
WHERE POLICY_NAME = 'POL_DML_LIQUOR';
```

Resultado de la Consulta x

Todas las Filas Recuperadas: 18 en 1.51 segundos

POLICY_NAME	AUDIT_CONDITION	CONDITION_EVAL_OPT	AUDIT_OPTION	AUDIT_OPTION_TYPE	OBJECT_SCHEMA	OBJECT_NAME	OBJECT_TYPE	COMMON	INHERITED
1 POL_DML_LIQUOR	NONE	NONE	DELETE	OBJECT ACTION	ESQUEMALIQUOR	REPARTIDOR	TABLE	NO	NO
2 POL_DML_LIQUOR	NONE	NONE	INSERT	OBJECT ACTION	ESQUEMALIQUOR	REPARTIDOR	TABLE	NO	NO
3 POL_DML_LIQUOR	NONE	NONE	UPDATE	OBJECT ACTION	ESQUEMALIQUOR	REPARTIDOR	TABLE	NO	NO
4 POL_DML_LIQUOR	NONE	NONE	DELETE	OBJECT ACTION	ESQUEMALIQUOR	CLIENTE	TABLE	NO	NO
5 POL_DML_LIQUOR	NONE	NONE	INSERT	OBJECT ACTION	ESQUEMALIQUOR	CLIENTE	TABLE	NO	NO
6 POL_DML_LIQUOR	NONE	NONE	UPDATE	OBJECT ACTION	ESQUEMALIQUOR	CLIENTE	TABLE	NO	NO
7 POL_DML_LIQUOR	NONE	NONE	DELETE	OBJECT ACTION	ESQUEMALIQUOR	TIPO_PRODUCTO	TABLE	NO	NO
8 POL_DML_LIQUOR	NONE	NONE	INSERT	OBJECT ACTION	ESQUEMALIQUOR	TIPO_PRODUCTO	TABLE	NO	NO
9 POL_DML_LIQUOR	NONE	NONE	UPDATE	OBJECT ACTION	ESQUEMALIQUOR	TIPO_PRODUCTO	TABLE	NO	NO


```
SELECT os_username,userhost,DBUSERNAME,client_program_name,event_timestamp, action_name,return_code,
object_schema,object_name,sql_text,unified_audit_policies FROM UNIFIED_AUDIT_TRAIL
WHERE unified_audit_policies='POL_DML_LIQUOR' order by event_timestamp desc;
```

Resultado de la Consulta 1 x

Todas las Filas Recuperadas: 1 en 0.012 segundos

OS_USERNAME	USERHOST	DBUSERNAME	CLIENT_PROGRAM_NAME	EVENT_TIMESTAMP	ACTION_NAME	RETURN_CODE	OBJECT_SCHEMA	OBJECT_NAME	SQL_TEXT
1 tony_	DESKTOP-89NCM09	ADMINLIQUOR	SQL Developer	16/11/22 02:01:22.661000000	INSERT	0	ESQUEMALIQUOR	PRODUCTO	INSERT INTO ESQUEMALIQUOR.

--Política de Auditoria de acciones DDL

```
create audit policy POL_DLL_LIQUOR
ACTIONS
    CREATE PACKAGE, ALTER PACKAGE, DROP PACKAGE,
    CREATE PACKAGE BODY, ALTER PACKAGE BODY, DROP PACKAGE BODY,
    CREATE PROCEDURE, DROP PROCEDURE, ALTER PROCEDURE,
    CREATE FUNCTION, DROP FUNCTION, ALTER FUNCTION,
    CREATE TRIGGER, ALTER TRIGGER, DROP TRIGGER,
    CREATE TABLE, ALTER TABLE, DROP TABLE, TRUNCATE TABLE
;

AUDIT POLICY POL_DLL_LIQUOR BY ESQUEMALIQUOR;
```

SELECT os_username, userhost, DBUSER_NAME, client_program_name, event_timestamp, action_name, return_code, object_schema, object_name, sql_text, unified_audit_policies FROM UNIFIED_AUDIT_TRAIL WHERE unified_audit_policies='POL_DLL_LIQUOR' order by event_timestamp desc;

OS_USERNAME	USERHOST	DBUSER_NAME	CLIENT_PROGRAM_NAME	EVENT_TIMESTAMP	ACTION_NAME	RETURN_CODE	OBJECT_SCHEMA	OBJECT_NAME	SQL_TEXT
1 tony_	DESKTOP-89NCM09	ESQUEMALIQUOR	SQL Developer	16/11/22 02:24:24.282000000	CREATE PROCEDURE	0	ESQUEMALIQUOR	PR_REPORTES_VENTA	CREATE OR REPLACE P
2 tony_	DESKTOP-89NCM09	ESQUEMALIQUOR	SQL Developer	16/11/22 02:24:01.788000000	CREATE FUNCTION	0	ESQUEMALIQUOR	CONVERT_SOLES	CREATE OR REPLACE F

h. Backups

Habilitar Backups en caliente

Verificar el modo log de la database

```
SQL> show con_name

CON_NAME
-----
CDB$ROOT
SQL> archive log list
Modo log de la base de datos      Modo de No Archivado
Archivado automatico              Desactivado
Destino del archivo                D:\Oracle\dbhomeXE\RDBMS
Secuencia de log en línea mbs antigua  113
Secuencia de log actual            115
SQL>
```

Bajar todas las instancias de la database

```
SQL> shutdown immediate
Base de datos cerrada.
Base de datos desmontada.
Instancia ORACLE cerrada.
SQL>
```

Levantar la database en estado MOUNT

```
SQL> startup mount;
Instancia ORACLE iniciada.

Total System Global Area 1610610664 bytes
Fixed Size                 9029608 bytes
Variable Size             788529152 bytes
Database Buffers          805306368 bytes
Redo Buffers              7745536 bytes
Base de datos montada.
SQL> _
```

Cambiamos a modo de archivado

```
SQL> alter database archivelog;

Base de datos modificada.

SQL> _
```

Abrir la base de datos

```
SQL> alter database open;

Base de datos modificada.

SQL>
```

Configurar políticas de redundancia

```
RMAN> CONFIGURE RETENTION POLICY TO REDUNDANCY 2;

se utiliza el archivo de control de la base de datos destino en lugar del cat logo de recuperaci n
par metros de configuraci n de RMAN nuevos:
CONFIGURE RETENTION POLICY TO REDUNDANCY 2;
se han almacenado correctamente los par metros de configuraci n de RMAN nuevos
```

Configurar guardado del backup(controlfile)

```
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO
'D:\Oracle\DBO\ControlBackup\CONTROLPROYECTO_%.F';
```

```
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO'D:\Oracle\DBO\ControlBackup\CONTROLLIQUOR_%.F';

se utiliza el archivo de control de la base de datos destino en lugar del cat logo de recuperaci n
par metros de configuraci n de RMAN de antiguos:
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO 'D:\Oracle\DBO\ControlBackup\CONTROLLIQUOR_%.F';
par metros de configuraci n de RMAN nuevos:
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO 'D:\Oracle\DBO\ControlBackup\CONTROLLIQUOR_%.F';
se han almacenado correctamente los par metros de configuraci n de RMAN nuevos
```

Generar BackUp con ruta específica

```
BACKUP AS COMPRESSED BACKUPSET PLUGGABLE DATABASE PROYECTO FORMAT  
'D:\Oracle\DBO\PluggableBackup\BK_PROYECTO_%d_%s_%T.bkp';
```

```
RMAN> BACKUP AS COMPRESSED BACKUPSET PLUGGABLE DATABASE PROYECTO FORMAT 'D:\Oracle\DBO\PluggableBackup\BK_PROYECTO_%d_%s_%T.bkp';

Empezando backup a las 17/11/22
canal asignado: ORA_DISK_1
canal ORA_DISK_1: SID=1226 tipo de dispositivo=DISK
canal ORA_DISK_1: iniciando juego de copias de seguridad de archivo de datos completo comprimido
canal ORA_DISK_1: especificando archivo(s) de datos en el juego de copias de seguridad
n-mero de archivo de datos de entrada=00032 nombre=D:\ORACLE\ORADATA\XE\PROYECTO\SYSTEM01.DBF
n-mero de archivo de datos de entrada=00031 nombre=D:\ORACLE\ORADATA\XE\PROYECTO\SYSTEM01.DBF
n-mero de archivo de datos de entrada=00033 nombre=D:\ORACLE\ORADATA\XE\PROYECTO\UNDOTBS01.DBF
n-mero de archivo de datos de entrada=00034 nombre=D:\ORACLE\ORADATA\XE\PROYECTO\PDBLIQUOR01.DBF
canal ORA_DISK_1: iniciando parte 1 en 17/11/22
canal ORA_DISK_1: finalizando parte 1 en 17/11/22
manejador de parte=D:\ORACLE\DBO\PLUGGABLEBACKUP\BK_PROYECTO_XE_23_20221117.BKP etiqueta=TAG20221117T222120 comentario=NONE
canal ORA_DISK_1: juego de copias de seguridad terminado, tiempo transcurrido: 00:00:25
Se ha finalizado backup a las 17/11/22

Empezando Control File and SPFILE Autobackup a las 17/11/22
manejador de parte=D:\ORACLE\DBO\CONTROLBACKUP\CONTROLPROYECTO_C-2995382763-20221117-00 comentario=NONE
Se ha finalizado Control File and SPFILE Autobackup a las 17/11/22
```

Archivos creados

Nuevo vol (D:) > Oracle > DBO > ControlBackup				
Nombre	Fecha de modificación	Tipo	Tamaño	
CONTROLPROYECTO_C-2995382763-202...	17/11/2022 10:21 p. m.	Archivo	18,400 KB	

Nuevo vol (D:) > Oracle > DBO > PluggableBackup				
Nombre	Fecha de modificación	Tipo	Tamaño	
BK_PROYECTO_XE_23_20221117.BKP	17/11/2022 10:21 p. m.	Archivo BKP	179,008 KB	

VIII. CONCLUSIONES

La Base de Datos Cumple con las reglas del negocio y requerimientos de la licorería “Liquor”, permitirá a los trabajadores y al gerente tener un mejor control de sus ventas por delivery, control de los repartidores que llevan los pedidos y los clientes que consume en el local.

Se aplico distintas estrategias de vistas, funciones, procedimientos y paquetes para la correcta gestión de la base de datos de la licorería.

Los Trigger implementados en la base de datos permitirán actualizar registro según se requiera, además de cumplir ciertas reglas de negocio del restaurante.

Las Auditorias y BackUps permitirán tener un registro de quien se conecta y visualiza la base de datos, además de tener un respaldo en caso de perdidas.

INTEGRANTES PARTICIPACION

Acosta Fernández Ricardo H.,	100%
Castillo García Mayte A.,	100%
Palacios Martel Anthony I.,	100%
Tafur Cabrera Paul J	100%.