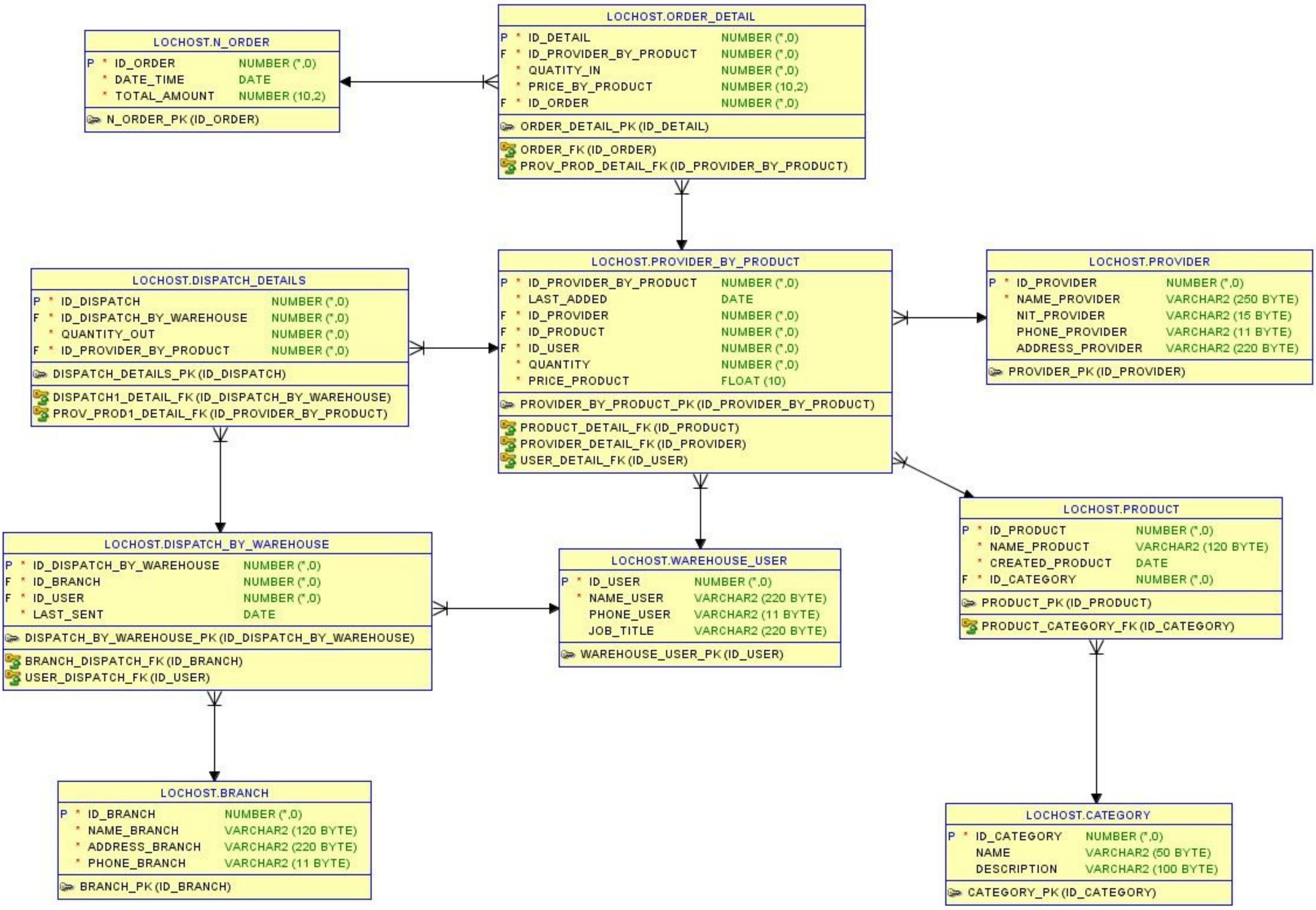
Universidad Mariano Gálvez de Guatemala
Ingeniería en Sistemas de Información y Ciencias de la Computación
Bases de Datos II Sección "A"
Fecha: Guatemala, 30 Mayo de 2020
Catedrático: Ing. Eddy Hernandez.

Proyecto Warehouse.

Anthony Joshua Vanegas Catalán 0900-17-12336
Celso Javier Rojas Villegas 0900-17-6674
Fernando Jose Argueta Martinez 0900-17-645
Pablo Daniel Villagrán Casasola 0900-16-24159
Pedro Rolando Flores Duque 0900-17-1417

1. Modelado de Datos.



LOCHOST.N_ORDER			
P	*	ID_ORDER	NUMBER (*,0)
	*	DATE_TIME	DATE
	*	TOTAL_AMOUNT	NUMBER (10,2)
N_ORDER_PK (ID_ORDER)			

LOCHOST.ORDER_DETAIL			
P	*	ID_DETAIL	NUMBER (*,0)
F	*	ID_PROVIDER_BY_PRODUCT	NUMBER (*,0)
	*	QUATITY_IN	NUMBER (*,0)
	*	PRICE_BY_PRODUCT	NUMBER (10,2)
F	*	ID_ORDER	NUMBER (*,0)
ORDER_DETAIL_PK (ID_DETAIL)			
ORDER_FK (ID_ORDER)			
PROV_PROD_DETAIL_FK (ID_PROVIDER_BY_PRODUCT)			

LOCHOST.DISPATCH_DETAILS			
P	*	ID_DISPATCH	NUMBER (*,0)
F	*	ID_DISPATCH_BY_WAREHOUSE	NUMBER (*,0)
	*	QUANTITY_OUT	NUMBER (*,0)
F	*	ID_PROVIDER_BY_PRODUCT	NUMBER (*,0)
DISPATCH_DETAILS_PK (ID_DISPATCH)			
DISPATCH1_DETAIL_FK (ID_DISPATCH_BY_WAREHOUSE)			
PROV_PROD1_DETAIL_FK (ID_PROVIDER_BY_PRODUCT)			

LOCHOST.PROVIDER_BY_PRODUCT			
P	*	ID_PROVIDER_BY_PRODUCT	NUMBER (*,0)
	*	LAST_ADDED	DATE
F	*	ID_PROVIDER	NUMBER (*,0)
F	*	ID_PRODUCT	NUMBER (*,0)
F	*	ID_USER	NUMBER (*,0)
	*	QUANTITY	NUMBER (*,0)
	*	PRICE_PRODUCT	FLOAT (10)
PROVIDER_BY_PRODUCT_PK (ID_PROVIDER_BY_PRODUCT)			
PRODUCT_DETAIL_FK (ID_PRODUCT)			
PROVIDER_DETAIL_FK (ID_PROVIDER)			
USER_DETAIL_FK (ID_USER)			

LOCHOST.PROVIDER			
P	*	ID_PROVIDER	NUMBER (*,0)
	*	NAME_PROVIDER	VARCHAR2 (250 BYTE)
		NIT_PROVIDER	VARCHAR2 (15 BYTE)
		PHONE_PROVIDER	VARCHAR2 (11 BYTE)
		ADDRESS_PROVIDER	VARCHAR2 (220 BYTE)
PROVIDER_PK (ID_PROVIDER)			

LOCHOST.DISPATCH_BY_WAREHOUSE			
P	*	ID_DISPATCH_BY_WAREHOUSE	NUMBER (*,0)
F	*	ID_BRANCH	NUMBER (*,0)
F	*	ID_USER	NUMBER (*,0)
	*	LAST_SENT	DATE
DISPATCH_BY_WAREHOUSE_PK (ID_DISPATCH_BY_WAREHOUSE)			
BRANCH_DISPATCH_FK (ID_BRANCH)			
USER_DISPATCH_FK (ID_USER)			

LOCHOST.WAREHOUSE_USER			
P	*	ID_USER	NUMBER (*,0)
	*	NAME_USER	VARCHAR2 (220 BYTE)
		PHONE_USER	VARCHAR2 (11 BYTE)
		JOB_TITLE	VARCHAR2 (220 BYTE)
WAREHOUSE_USER_PK (ID_USER)			

LOCHOST.PRODUCT			
P	*	ID_PRODUCT	NUMBER (*,0)
	*	NAME_PRODUCT	VARCHAR2 (120 BYTE)
	*	CREATED_PRODUCT	DATE
F	*	ID_CATEGORY	NUMBER (*,0)
PRODUCT_PK (ID_PRODUCT)			
PRODUCT_CATEGORY_FK (ID_CATEGORY)			

LOCHOST.BRANCH			
P	*	ID_BRANCH	NUMBER (*,0)
	*	NAME_BRANCH	VARCHAR2 (120 BYTE)
	*	ADDRESS_BRANCH	VARCHAR2 (220 BYTE)
	*	PHONE_BRANCH	VARCHAR2 (11 BYTE)
BRANCH_PK (ID_BRANCH)			

LOCHOST.CATEGORY			
P	*	ID_CATEGORY	NUMBER (*,0)
		NAME	VARCHAR2 (50 BYTE)
		DESCRIPTION	VARCHAR2 (100 BYTE)
CATEGORY_PK (ID_CATEGORY)			

2. **ACID**

La solución asegura su calidad y estabilidad en base a las reglas definidas en el acrónimo ACID. Los motores de base de datos de la actualidad son estandarizados de tal manera que resulta improbable que se rompan los conceptos que se mencionan en la normativa. Sin embargo, como proveedores, el equipo se asegura que todas las transacciones se atengan a estas propiedades por medio de los siguientes puntos:

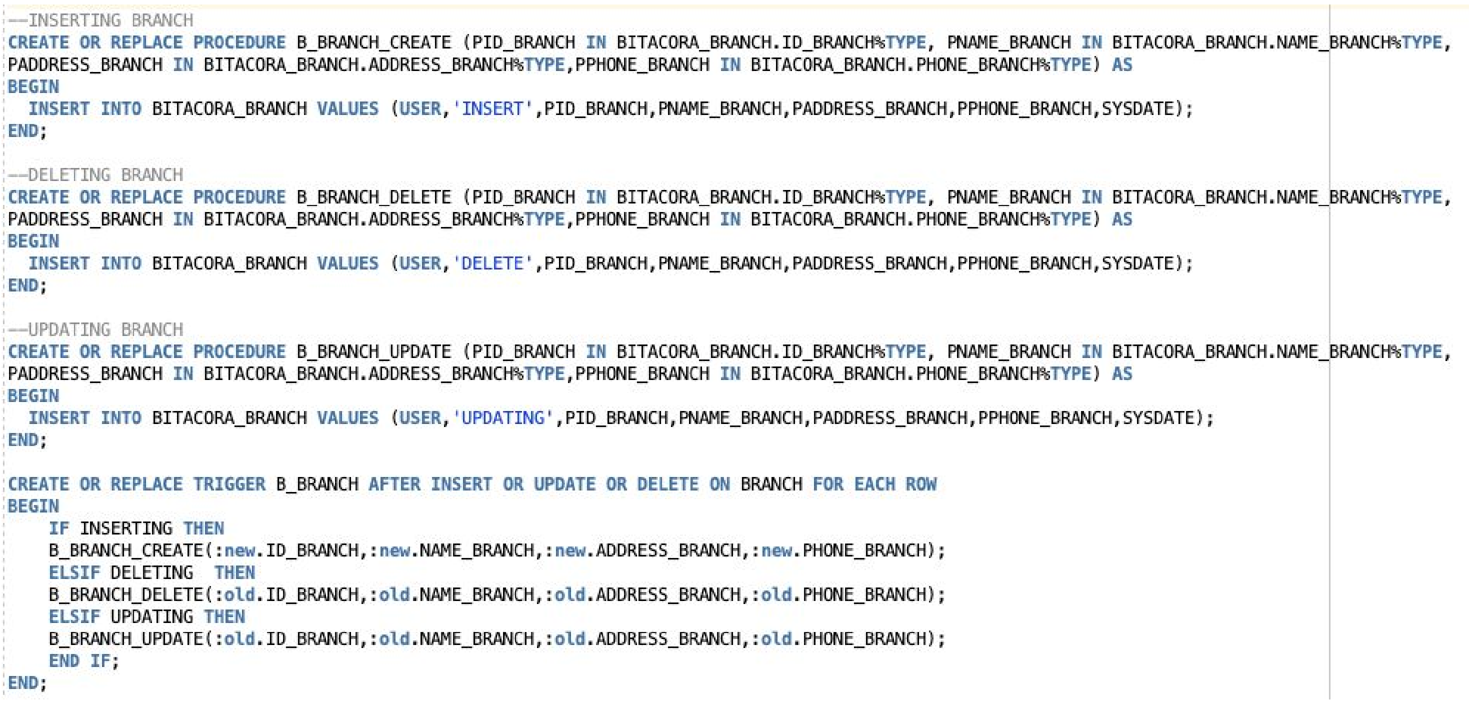
- Por medio del llamado de procedimientos el proceso asegura que todas las transacciones se completen de manera individual y atómica.
- Existen tablas de catálogo con las que se asegura la integridad de la información a manipular y asegurar que no existan datos que no correspondan al esquema por medio de sus respectivas llaves foráneas.
- El llamado de las transacciones a base de datos a nivel de aplicación se realiza de manera individual y cada transacción se ejecuta bajo commit automático al finalizar.
- Los procedimientos utilizados permiten que los eventos transaccionales se ejecuten a nivel de base de datos y por lo tanto se asume que al recibir la confirmación de éxito la data queda asegurada en el disco principal.
- Se garantiza la ejecución de procesos de back-up bajo estas mismas reglas a manera de asegurar la información.

3. Monitoreo de Transacciones.

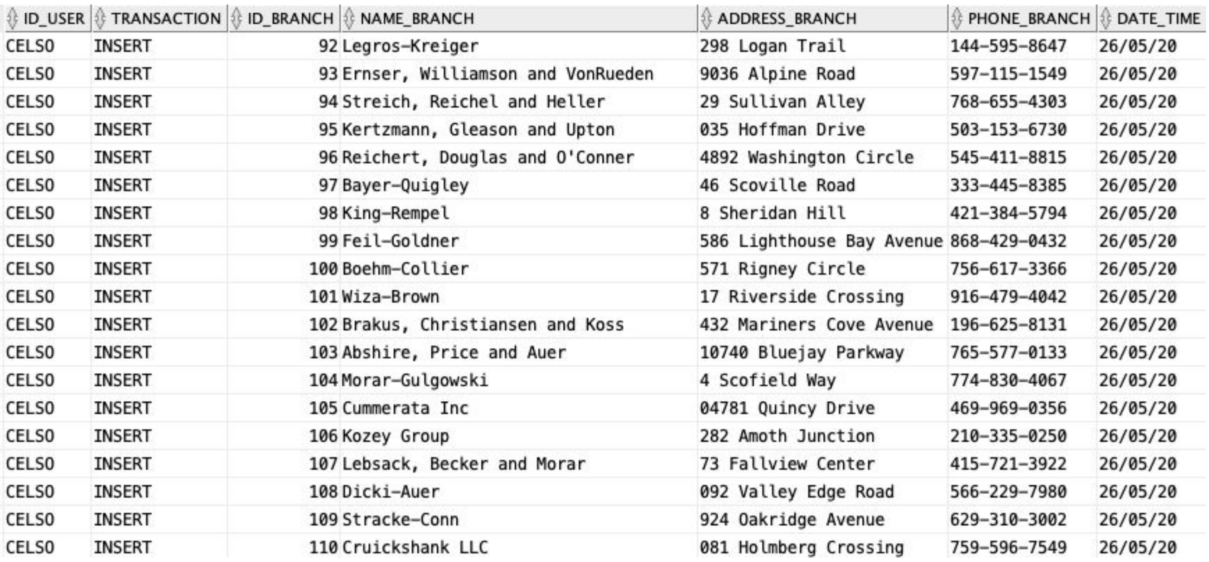
BITÁCORA

Se implementaron tablas Bitácora por cada objeto en el modelado de datos, estas bitácoras fueron pobladas por medio de procedimientos almacenados y Triggers para el control de transacciones.
Controlando los ingresos , cambios y borrado de datos en las tablas. Registrando usuario, fecha y acción.

Para las Tablas que cuentan con valores numéricos como montos, totales, stock se almaceno el :old de cada atributo según la tabla.



```
--INSERTING BRANCH
CREATE OR REPLACE PROCEDURE B_BRANCH_CREATE (PID_BRANCH IN BITACORA_BRANCH.ID_BRANCH%TYPE, PNAME_BRANCH IN BITACORA_BRANCH.NAME_BRANCH%TYPE,
PADDRESS_BRANCH IN BITACORA_BRANCH.ADDRESS_BRANCH%TYPE,PPHONE_BRANCH IN BITACORA_BRANCH.PHONE_BRANCH%TYPE) AS
BEGIN
  INSERT INTO BITACORA_BRANCH VALUES (USER,'INSERT',PID_BRANCH,PNAME_BRANCH,PADDRESS_BRANCH,PPHONE_BRANCH,SYSDATE);
END;

--DELETING BRANCH
CREATE OR REPLACE PROCEDURE B_BRANCH_DELETE (PID_BRANCH IN BITACORA_BRANCH.ID_BRANCH%TYPE, PNAME_BRANCH IN BITACORA_BRANCH.NAME_BRANCH%TYPE,
PADDRESS_BRANCH IN BITACORA_BRANCH.ADDRESS_BRANCH%TYPE,PPHONE_BRANCH IN BITACORA_BRANCH.PHONE_BRANCH%TYPE) AS
BEGIN
  INSERT INTO BITACORA_BRANCH VALUES (USER,'DELETE',PID_BRANCH,PNAME_BRANCH,PADDRESS_BRANCH,PPHONE_BRANCH,SYSDATE);
END;

--UPDATING BRANCH
CREATE OR REPLACE PROCEDURE B_BRANCH_UPDATE (PID_BRANCH IN BITACORA_BRANCH.ID_BRANCH%TYPE, PNAME_BRANCH IN BITACORA_BRANCH.NAME_BRANCH%TYPE,
PADDRESS_BRANCH IN BITACORA_BRANCH.ADDRESS_BRANCH%TYPE,PPHONE_BRANCH IN BITACORA_BRANCH.PHONE_BRANCH%TYPE) AS
BEGIN
  INSERT INTO BITACORA_BRANCH VALUES (USER,'UPDATING',PID_BRANCH,PNAME_BRANCH,PADDRESS_BRANCH,PPHONE_BRANCH,SYSDATE);
END;

CREATE OR REPLACE TRIGGER B_BRANCH AFTER INSERT OR UPDATE OR DELETE ON BRANCH FOR EACH ROW
BEGIN
    IF INSERTING THEN
    B_BRANCH_CREATE(:new.ID_BRANCH,:new.NAME_BRANCH,:new.ADDRESS_BRANCH,:new.PHONE_BRANCH);
    ELSIF DELETING  THEN
    B_BRANCH_DELETE(:old.ID_BRANCH,:old.NAME_BRANCH,:old.ADDRESS_BRANCH,:old.PHONE_BRANCH);
    ELSIF UPDATING THEN
    B_BRANCH_UPDATE(:old.ID_BRANCH,:old.NAME_BRANCH,:old.ADDRESS_BRANCH,:old.PHONE_BRANCH);
    END IF;
END;
```

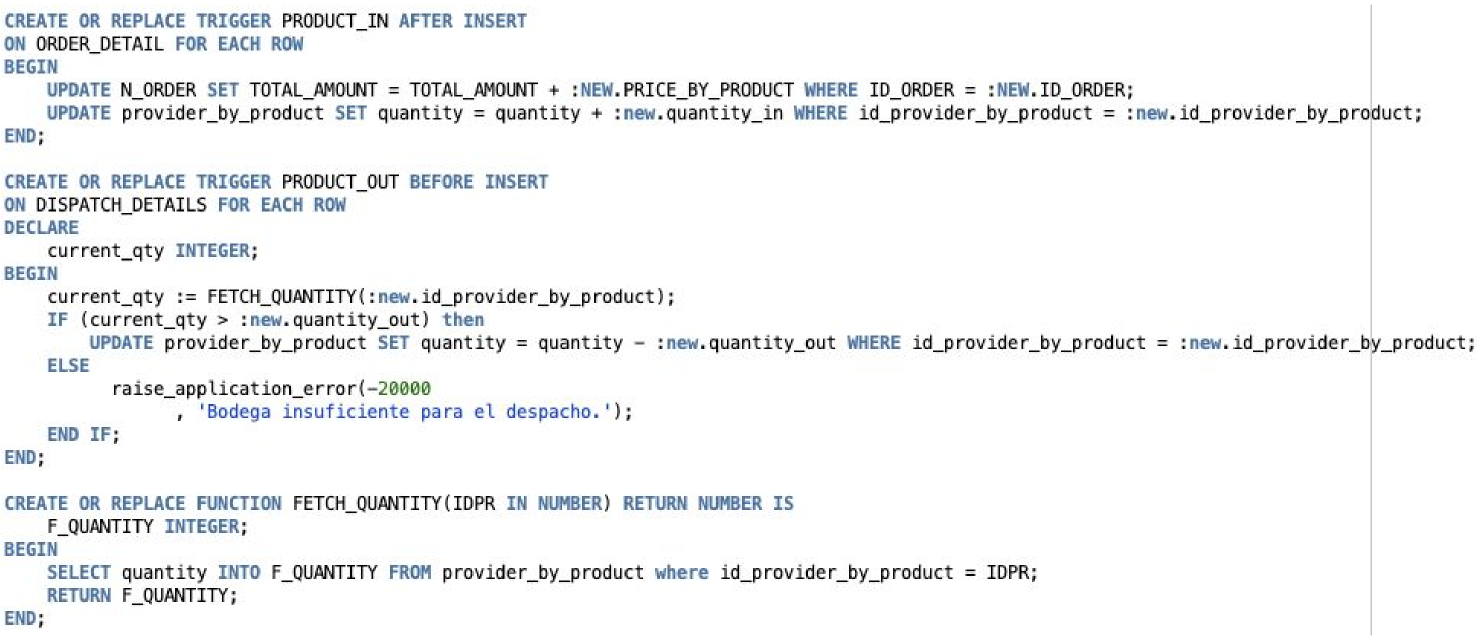


ID_USER	TRANSACTION	ID_BRANCH	NAME_BRANCH	ADDRESS_BRANCH	PHONE_BRANCH	DATE_TIME
CELSO	INSERT	92	Legros-Kreiger	298 Logan Trail	144-595-8647	26/05/20
CELSO	INSERT	93	Ernser, Williamson and VonRueden	9036 Alpine Road	597-115-1549	26/05/20
CELSO	INSERT	94	Streich, Reichel and Heller	29 Sullivan Alley	768-655-4303	26/05/20
CELSO	INSERT	95	Kertzmann, Gleason and Upton	035 Hoffman Drive	503-153-6730	26/05/20
CELSO	INSERT	96	Reichert, Douglas and O'Conner	4892 Washington Circle	545-411-8815	26/05/20
CELSO	INSERT	97	Bayer-Quigley	46 Scoville Road	333-445-8385	26/05/20
CELSO	INSERT	98	King-Rempel	8 Sheridan Hill	421-384-5794	26/05/20
CELSO	INSERT	99	Feil-Goldner	586 Lighthouse Bay Avenue	868-429-0432	26/05/20
CELSO	INSERT	100	Boehm-Collier	571 Rigney Circle	756-617-3366	26/05/20
CELSO	INSERT	101	Wiza-Brown	17 Riverside Crossing	916-479-4042	26/05/20
CELSO	INSERT	102	Brakus, Christiansen and Koss	432 Mariners Cove Avenue	196-625-8131	26/05/20
CELSO	INSERT	103	Abshire, Price and Auer	10740 Bluejay Parkway	765-577-0133	26/05/20
CELSO	INSERT	104	Morar-Gulgowski	4 Scofield Way	774-830-4067	26/05/20
CELSO	INSERT	105	Cummerata Inc	04781 Quincy Drive	469-969-0356	26/05/20
CELSO	INSERT	106	Kozey Group	282 Amoth Junction	210-335-0250	26/05/20
CELSO	INSERT	107	Lebsack, Becker and Morar	73 Fallview Center	415-721-3922	26/05/20
CELSO	INSERT	108	Dicki-Auer	092 Valley Edge Road	566-229-7980	26/05/20
CELSO	INSERT	109	Stracke-Conn	924 Oakridge Avenue	629-310-3002	26/05/20
CELSO	INSERT	110	Cruickshank LLC	081 Holmberg Crossing	759-596-7549	26/05/20

4. Funciones, Procedimientos, Excepciones, Secuencias. (DDL , DML, DCL , TCL).

SCRIPTS PARA ENTRADA Y SALIDA DE INVENTARIO

Se implementó el uso de Triggers para manejar el sistema de inventarios al haber una orden y un despacho, así como validación de no despachar más de la cantidad permitida.

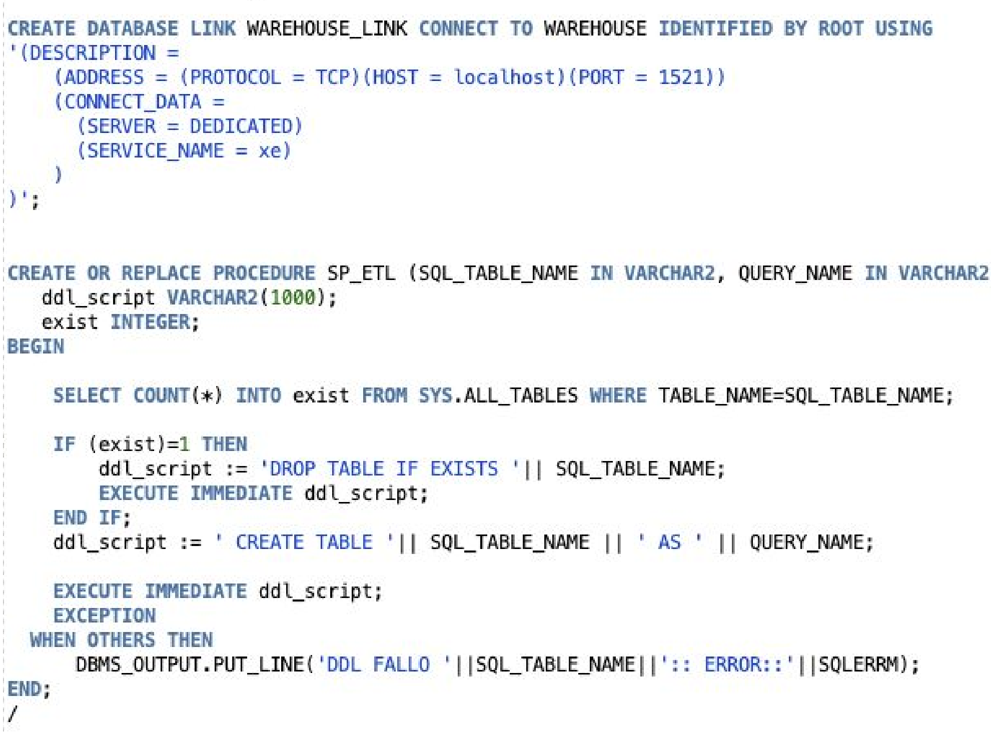


```
CREATE OR REPLACE TRIGGER PRODUCT_IN AFTER INSERT
ON ORDER_DETAIL FOR EACH ROW
BEGIN
    UPDATE N_ORDER SET TOTAL_AMOUNT = TOTAL_AMOUNT + :NEW.PRICE_BY_PRODUCT WHERE ID_ORDER = :NEW.ID_ORDER;
    UPDATE provider_by_product SET quantity = quantity + :new.quantity_in WHERE id_provider_by_product = :new.id_provider_by_product;
END;

CREATE OR REPLACE TRIGGER PRODUCT_OUT BEFORE INSERT
ON DISPATCH_DETAILS FOR EACH ROW
DECLARE
    current_qty INTEGER;
BEGIN
    current_qty := FETCH_QUANTITY(:new.id_provider_by_product);
    IF (current_qty > :new.quantity_out) then
        UPDATE provider_by_product SET quantity = quantity - :new.quantity_out WHERE id_provider_by_product = :new.id_provider_by_product;
    ELSE
          raise_application_error(-20000
                , 'Bodega insuficiente para el despacho.');
    END IF;
END;

CREATE OR REPLACE FUNCTION FETCH_QUANTITY(IDPR IN NUMBER) RETURN NUMBER IS
    F_QUANTITY INTEGER;
BEGIN
    SELECT quantity INTO F_QUANTITY FROM provider_by_product where id_provider_by_product = IDPR;
    RETURN F_QUANTITY;
END;
```

SCRIPT ETL

Se realizó un procedimiento almacenado en la base de datos Warehouse Reports, cuya finalidad es obtener el reporte de la base de datos Warehouse y así poder hacer uso en el módulo de BI.

Este procedimiento se adaptó a un requerimiento del cliente, para poder hacer cualquier reporte, de forma que se parametrizo el nombre de la tabla y el query a transformar y extraer, con esta funcionalidad se puede reutilizar el bloque para cualquier query deseado.



```
CREATE DATABASE LINK WAREHOUSE_LINK CONNECT TO WAREHOUSE IDENTIFIED BY ROOT USING
'(DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = xe)
    )
)';


CREATE OR REPLACE PROCEDURE SP_ETL (SQL_TABLE_NAME IN VARCHAR2, QUERY_NAME IN VARCHAR2) IS
   ddl_script VARCHAR2(1000);
   exist INTEGER;
BEGIN

    SELECT COUNT(*) INTO exist FROM SYS.ALL_TABLES WHERE TABLE_NAME=SQL_TABLE_NAME;

    IF (exist)=1 THEN
        ddl_script := 'DROP TABLE IF EXISTS '|| SQL_TABLE_NAME;
        EXECUTE IMMEDIATE ddl_script;
    END IF;
    ddl_script := ' CREATE TABLE '|| SQL_TABLE_NAME || ' AS ' || QUERY_NAME;

    EXECUTE IMMEDIATE ddl_script;
    EXCEPTION
  WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('DDL FALLO '||SQL_TABLE_NAME||':: ERROR::'||SQLERRM);
END;
/
```

DDL: Data definition language.
Es útil para crear y modificar las estructuras de una base de datos. algunos ejemplos son:

CREATE: se utiliza para crear nuevas tablas, campos e índices.

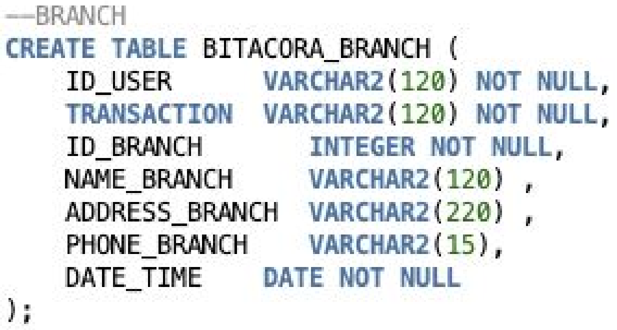
ALTER: se utiliza para editar las tablas agregando campos o cambiando su definición.

DROP: se utiliza para eliminar registros , tablas e índices.

RENAME: se utiliza para renombrar objetos.

TRUNCATE: se utiliza para eliminar todos los elementos de una tabla.

COMMENT: se utiliza para añadir comentarios en el diccionario de datos.



```
--BRANCH
CREATE TABLE BITACORA_BRANCH (
    ID_USER      VARCHAR2(120) NOT NULL,
    TRANSACTION  VARCHAR2(120) NOT NULL,
    ID_BRANCH       INTEGER NOT NULL,
    NAME_BRANCH     VARCHAR2(120) ,
    ADDRESS_BRANCH  VARCHAR2(220) ,
    PHONE_BRANCH    VARCHAR2(15),
    DATE_TIME    DATE NOT NULL
);
```

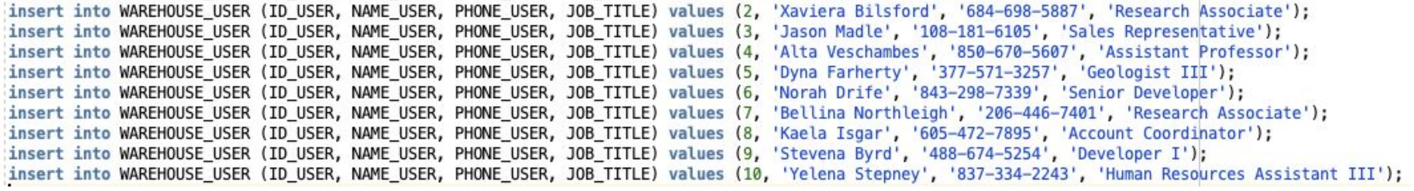
DML: Data manipulation language.
Es útil para el ingreso de datos, para realizar tareas de consulta o de modificación sobre los datos en la Base de datos. algunos ejemplos son:

SELECT: se utiliza para realizar consulta sobre los datos.

INSERT: se utiliza para insertar valores en nuestra base de datos.

UPDATE: se utiliza para modificar y alterar los valores de los registros de la base de datos.

DELETE: se utiliza para eliminar registros de la base de datos.

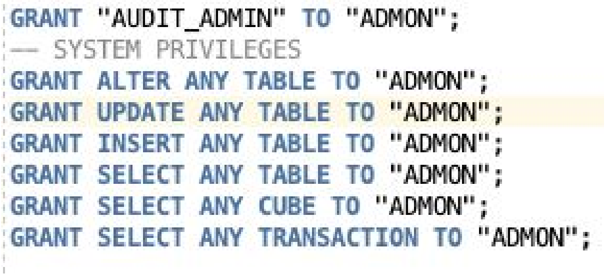


```
insert into WAREHOUSE_USER (ID_USER, NAME_USER, PHONE_USER, JOB_TITLE) values (1, 'Karl Calcraft', '287-306-3425', 'Senior Editor');
insert into WAREHOUSE_USER (ID_USER, NAME_USER, PHONE_USER, JOB_TITLE) values (2, 'Xaviera Bilsford', '684-698-5887', 'Research Associate');
insert into WAREHOUSE_USER (ID_USER, NAME_USER, PHONE_USER, JOB_TITLE) values (3, 'Jason Madle', '108-181-6105', 'Sales Representative');
insert into WAREHOUSE_USER (ID_USER, NAME_USER, PHONE_USER, JOB_TITLE) values (4, 'Alta Veschambes', '850-670-5607', 'Assistant Professor');
insert into WAREHOUSE_USER (ID_USER, NAME_USER, PHONE_USER, JOB_TITLE) values (5, 'Dyna Farherty', '377-571-3257', 'Geologist III');
insert into WAREHOUSE_USER (ID_USER, NAME_USER, PHONE_USER, JOB_TITLE) values (6, 'Norah Drife', '843-298-7339', 'Senior Developer');
insert into WAREHOUSE_USER (ID_USER, NAME_USER, PHONE_USER, JOB_TITLE) values (7, 'Bellina Northleigh', '206-446-7401', 'Research Associate');
insert into WAREHOUSE_USER (ID_USER, NAME_USER, PHONE_USER, JOB_TITLE) values (8, 'Kaela Isgar', '605-472-7895', 'Account Coordinator');
insert into WAREHOUSE_USER (ID_USER, NAME_USER, PHONE_USER, JOB_TITLE) values (9, 'Stevena Byrd', '488-674-5254', 'Developer I');
insert into WAREHOUSE_USER (ID_USER, NAME_USER, PHONE_USER, JOB_TITLE) values (10, 'Yelena Stepney', '837-334-2243', 'Human Resources Assistant III');
```

DCL: Data control language.
Es utilizado para administrar los roles dentro de determinadas tareas en la base de datos. ya sean permisos para permitir o denegar roles así también como controlar accesos y permisos que manipulan el sistema como crear tablas, eliminar registros, etc.

GRANT: se utiliza para otorgar permisos dentro de la base de datos.

REVOKE: es utilizado para eliminar los permisos otorgados.



```
GRANT "AUDIT_ADMIN" TO "ADMON";
-- SYSTEM PRIVILEGES
GRANT ALTER ANY TABLE TO "ADMON";
GRANT UPDATE ANY TABLE TO "ADMON";
GRANT INSERT ANY TABLE TO "ADMON";
GRANT SELECT ANY TABLE TO "ADMON";
GRANT SELECT ANY CUBE TO "ADMON";
GRANT SELECT ANY TRANSACTION TO "ADMON";
```

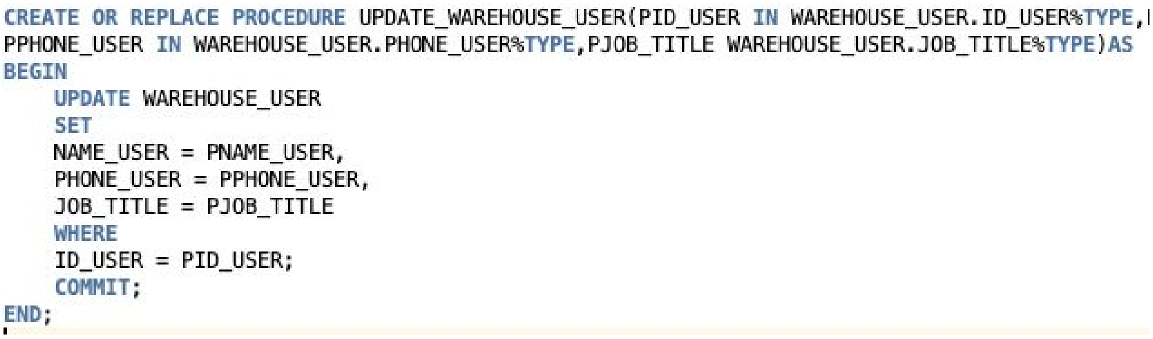
TCL: Transactional control language.
Es utilizado para el control de transacciones que ocurren dentro de la base de datos. algunos ejemplos son:

COMMIT: es empleado para guardar un trabajo realizado.

ROLLBACK: es utilizado para deshacer (volver) a la última modificación que se hizo antes del último COMMIT.

SAVEPOINT: identifica un punto en una transacción a la que más tarde se puede revertir.

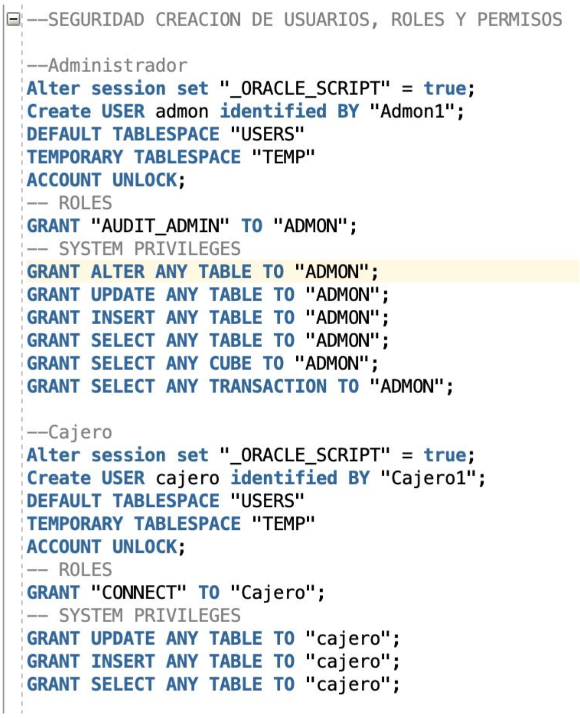
SET TRANSACTION: cambia las opciones de transacción a modo de aislamiento y que segmento de cancelación utiliza.



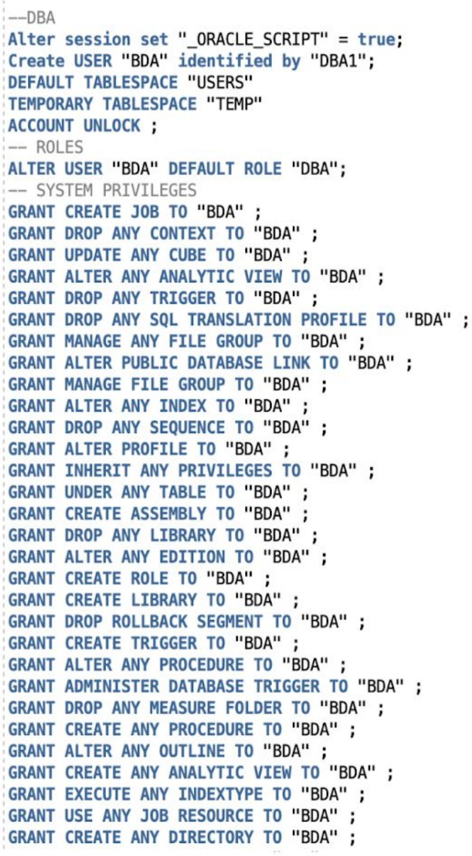
```
CREATE OR REPLACE PROCEDURE UPDATE_WAREHOUSE_USER(PID_USER IN WAREHOUSE_USER.ID_USER%TYPE,
PPHONE_USER IN WAREHOUSE_USER.PHONE_USER%TYPE,PJOB_TITLE WAREHOUSE_USER.JOB_TITLE%TYPE)AS
BEGIN
    UPDATE WAREHOUSE_USER
    SET
    NAME_USER = PNAME_USER,
    PHONE_USER = PPHONE_USER,
    JOB_TITLE = PJOB_TITLE
    WHERE
    ID_USER = PID_USER;
    COMMIT;
END;
```

5. Seguridad.

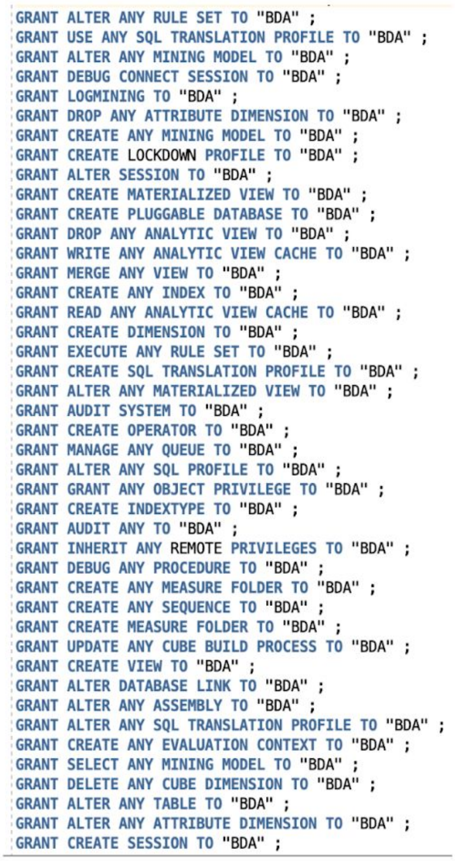
Para asegurar la seguridad de la base de datos se crearon usuarios, roles se les asignan permisos, esto es necesario ya que así nos aseguramos que la integridad de la base de datos se mantenga intacta, asignamos los permisos basados en el rol que cumple cada uno dentro de la empresa.



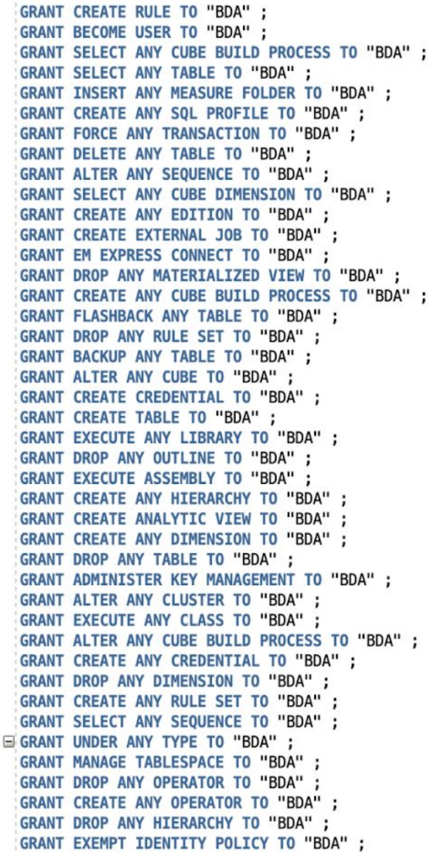
```
--SEGURIDAD CREACION DE USUARIOS, ROLES Y PERMISOS

--Administrador
Alter session set "_ORACLE_SCRIPT" = true;
Create USER admon identified BY "Admon1";
DEFAULT TABLESPACE "USERS"
TEMPORARY TABLESPACE "TEMP"
ACCOUNT UNLOCK;
-- ROLES
GRANT "AUDIT_ADMIN" TO "ADMON";
-- SYSTEM PRIVILEGES
GRANT ALTER ANY TABLE TO "ADMON";
GRANT UPDATE ANY TABLE TO "ADMON";
GRANT INSERT ANY TABLE TO "ADMON";
GRANT SELECT ANY TABLE TO "ADMON";
GRANT SELECT ANY CUBE TO "ADMON";
GRANT SELECT ANY TRANSACTION TO "ADMON";

--Cajero
Alter session set "_ORACLE_SCRIPT" = true;
Create USER cajero identified BY "Cajero1";
DEFAULT TABLESPACE "USERS"
TEMPORARY TABLESPACE "TEMP"
ACCOUNT UNLOCK;
-- ROLES
GRANT "CONNECT" TO "Cajero";
-- SYSTEM PRIVILEGES
GRANT UPDATE ANY TABLE TO "cajero";
GRANT INSERT ANY TABLE TO "cajero";
GRANT SELECT ANY TABLE TO "cajero";
```



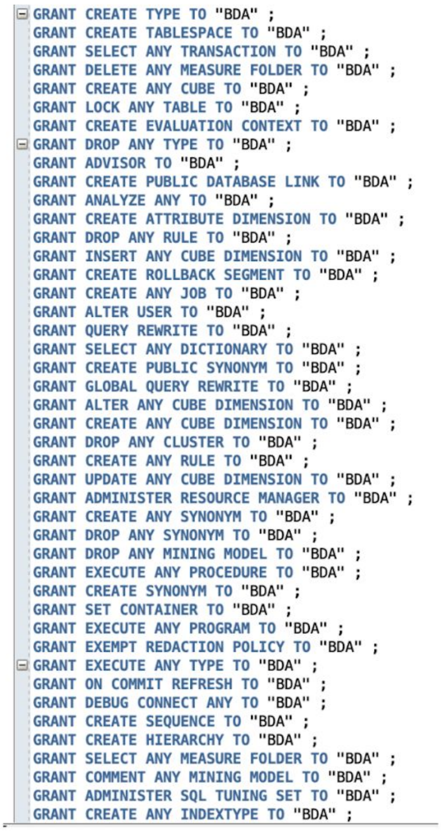
```sql
--DBA
Alter session set "_ORACLE_SCRIPT" = true;
Create USER "BDA" identified by "DBA1";
DEFAULT TABLESPACE "USERS"
TEMPORARY TABLESPACE "TEMP"
ACCOUNT UNLOCK ;
-- ROLES
ALTER USER "BDA" DEFAULT ROLE "DBA";
-- SYSTEM PRIVILEGES
GRANT CREATE JOB TO "BDA" ;
GRANT DROP ANY CONTEXT TO "BDA" ;
GRANT UPDATE ANY CUBE TO "BDA" ;
GRANT ALTER ANY ANALYTIC VIEW TO "BDA" ;
GRANT DROP ANY TRIGGER TO "BDA" ;
GRANT DROP ANY SQL TRANSLATION PROFILE TO "BDA" ;
GRANT MANAGE ANY FILE GROUP TO "BDA" ;
GRANT ALTER PUBLIC DATABASE LINK TO "BDA" ;
GRANT MANAGE FILE GROUP TO "BDA" ;
GRANT ALTER ANY INDEX TO "BDA" ;
GRANT DROP ANY SEQUENCE TO "BDA" ;
GRANT ALTER PROFILE TO "BDA" ;
GRANT INHERIT ANY PRIVILEGES TO "BDA" ;
GRANT UNDER ANY TABLE TO "BDA" ;
GRANT CREATE ASSEMBLY TO "BDA" ;
GRANT DROP ANY LIBRARY TO "BDA" ;
GRANT ALTER ANY EDITION TO "BDA" ;
GRANT CREATE ROLE TO "BDA" ;
GRANT CREATE LIBRARY TO "BDA" ;
GRANT DROP ROLLBACK SEGMENT TO "BDA" ;
GRANT CREATE TRIGGER TO "BDA" ;
GRANT ALTER ANY PROCEDURE TO "BDA" ;
GRANT ADMINISTER DATABASE TRIGGER TO "BDA" ;
GRANT DROP ANY MEASURE FOLDER TO "BDA" ;
GRANT CREATE ANY PROCEDURE TO "BDA" ;
GRANT ALTER ANY OUTLINE TO "BDA" ;
GRANT CREATE ANY ANALYTIC VIEW TO "BDA" ;
GRANT EXECUTE ANY INDEXTYPE TO "BDA" ;
GRANT USE ANY JOB RESOURCE TO "BDA" ;
GRANT CREATE ANY DIRECTORY TO "BDA" ;
```



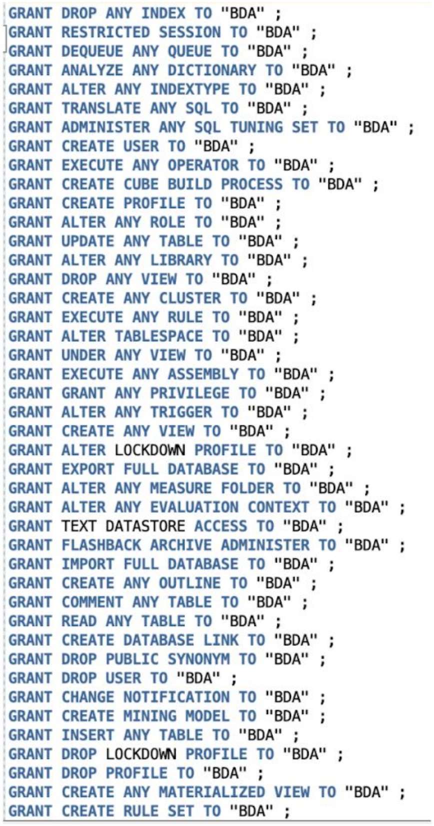
```sql
GRANT ALTER ANY RULE SET TO "BDA" ;
GRANT USE ANY SQL TRANSLATION PROFILE TO "BDA" ;
GRANT ALTER ANY MINING MODEL TO "BDA" ;
GRANT DEBUG CONNECT SESSION TO "BDA" ;
GRANT LOGMINING TO "BDA" ;
GRANT DROP ANY ATTRIBUTE DIMENSION TO "BDA" ;
GRANT CREATE ANY MINING MODEL TO "BDA" ;
GRANT CREATE LOCKDOWN PROFILE TO "BDA" ;
GRANT ALTER SESSION TO "BDA" ;
GRANT CREATE MATERIALIZED VIEW TO "BDA" ;
GRANT CREATE PLUGGABLE DATABASE TO "BDA" ;
GRANT DROP ANY ANALYTIC VIEW TO "BDA" ;
GRANT WRITE ANY ANALYTIC VIEW CACHE TO "BDA" ;
GRANT MERGE ANY VIEW TO "BDA" ;
GRANT CREATE ANY INDEX TO "BDA" ;
GRANT READ ANY ANALYTIC VIEW CACHE TO "BDA" ;
GRANT CREATE DIMENSION TO "BDA" ;
GRANT EXECUTE ANY RULE SET TO "BDA" ;
GRANT CREATE SQL TRANSLATION PROFILE TO "BDA" ;
GRANT ALTER ANY MATERIALIZED VIEW TO "BDA" ;
GRANT AUDIT SYSTEM TO "BDA" ;
GRANT CREATE OPERATOR TO "BDA" ;
GRANT MANAGE ANY QUEUE TO "BDA" ;
GRANT ALTER ANY SQL PROFILE TO "BDA" ;
GRANT GRANT ANY OBJECT PRIVILEGE TO "BDA" ;
GRANT CREATE INDEXTYPE TO "BDA" ;
GRANT AUDIT ANY TO "BDA" ;
GRANT INHERIT ANY REMOTE PRIVILEGES TO "BDA" ;
GRANT DEBUG ANY PROCEDURE TO "BDA" ;
GRANT CREATE ANY MEASURE FOLDER TO "BDA" ;
GRANT CREATE ANY SEQUENCE TO "BDA" ;
GRANT CREATE MEASURE FOLDER TO "BDA" ;
GRANT UPDATE ANY CUBE BUILD PROCESS TO "BDA" ;
GRANT CREATE VIEW TO "BDA" ;
GRANT ALTER DATABASE LINK TO "BDA" ;
GRANT ALTER ANY ASSEMBLY TO "BDA" ;
GRANT ALTER ANY SQL TRANSLATION PROFILE TO "BDA" ;
GRANT CREATE ANY EVALUATION CONTEXT TO "BDA" ;
GRANT SELECT ANY MINING MODEL TO "BDA" ;
GRANT DELETE ANY CUBE DIMENSION TO "BDA" ;
GRANT ALTER ANY TABLE TO "BDA" ;
GRANT ALTER ANY ATTRIBUTE DIMENSION TO "BDA" ;
GRANT CREATE SESSION TO "BDA" ;
```



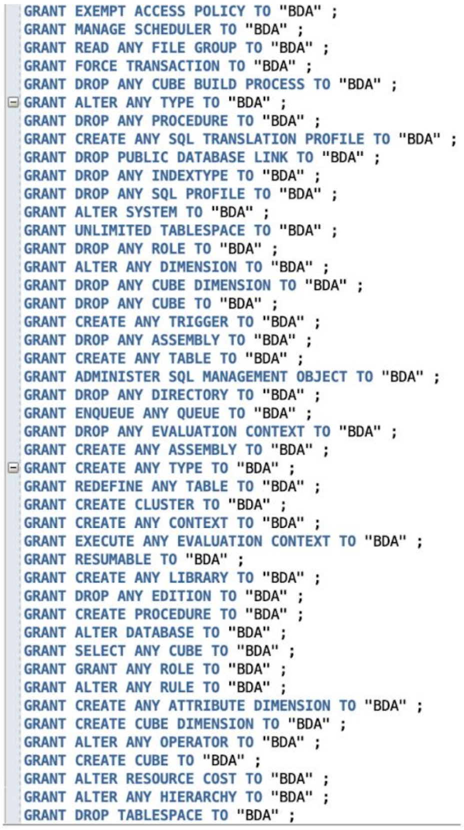
```sql
GRANT CREATE RULE TO "BDA" ;
GRANT BECOME USER TO "BDA" ;
GRANT SELECT ANY CUBE BUILD PROCESS TO "BDA" ;
GRANT SELECT ANY TABLE TO "BDA" ;
GRANT INSERT ANY MEASURE FOLDER TO "BDA" ;
GRANT CREATE ANY SQL PROFILE TO "BDA" ;
GRANT FORCE ANY TRANSACTION TO "BDA" ;
GRANT DELETE ANY TABLE TO "BDA" ;
GRANT ALTER ANY SEQUENCE TO "BDA" ;
GRANT SELECT ANY CUBE DIMENSION TO "BDA" ;
GRANT CREATE ANY EDITION TO "BDA" ;
GRANT CREATE EXTERNAL JOB TO "BDA" ;
GRANT EM EXPRESS CONNECT TO "BDA" ;
GRANT DROP ANY MATERIALIZED VIEW TO "BDA" ;
GRANT CREATE ANY CUBE BUILD PROCESS TO "BDA" ;
GRANT FLASHBACK ANY TABLE TO "BDA" ;
GRANT DROP ANY RULE SET TO "BDA" ;
GRANT BACKUP ANY TABLE TO "BDA" ;
GRANT ALTER ANY CUBE TO "BDA" ;
GRANT CREATE CREDENTIAL TO "BDA" ;
GRANT CREATE TABLE TO "BDA" ;
GRANT EXECUTE ANY LIBRARY TO "BDA" ;
GRANT DROP ANY OUTLINE TO "BDA" ;
GRANT EXECUTE ASSEMBLY TO "BDA" ;
GRANT CREATE ANY HIERARCHY TO "BDA" ;
GRANT CREATE ANALYTIC VIEW TO "BDA" ;
GRANT CREATE ANY DIMENSION TO "BDA" ;
GRANT DROP ANY TABLE TO "BDA" ;
GRANT ADMINISTER KEY MANAGEMENT TO "BDA" ;
GRANT ALTER ANY CLUSTER TO "BDA" ;
GRANT EXECUTE ANY CLASS TO "BDA" ;
GRANT ALTER ANY CUBE BUILD PROCESS TO "BDA" ;
GRANT CREATE ANY CREDENTIAL TO "BDA" ;
GRANT DROP ANY DIMENSION TO "BDA" ;
GRANT CREATE ANY RULE SET TO "BDA" ;
GRANT SELECT ANY SEQUENCE TO "BDA" ;
GRANT UNDER ANY TYPE TO "BDA" ;
GRANT MANAGE TABLESPACE TO "BDA" ;
GRANT DROP ANY OPERATOR TO "BDA" ;
GRANT CREATE ANY OPERATOR TO "BDA" ;
GRANT DROP ANY HIERARCHY TO "BDA" ;
GRANT EXEMPT IDENTITY POLICY TO "BDA" ;
```



```sql
GRANT CREATE TYPE TO "BDA" ;
GRANT CREATE TABLESPACE TO "BDA" ;
GRANT SELECT ANY TRANSACTION TO "BDA" ;
GRANT DELETE ANY MEASURE FOLDER TO "BDA" ;
GRANT CREATE ANY CUBE TO "BDA" ;
GRANT LOCK ANY TABLE TO "BDA" ;
GRANT CREATE EVALUATION CONTEXT TO "BDA" ;
GRANT DROP ANY TYPE TO "BDA" ;
GRANT ADVISOR TO "BDA" ;
GRANT CREATE PUBLIC DATABASE LINK TO "BDA" ;
GRANT ANALYZE ANY TO "BDA" ;
GRANT CREATE ATTRIBUTE DIMENSION TO "BDA" ;
GRANT DROP ANY RULE TO "BDA" ;
GRANT INSERT ANY CUBE DIMENSION TO "BDA" ;
GRANT CREATE ROLLBACK SEGMENT TO "BDA" ;
GRANT CREATE ANY JOB TO "BDA" ;
GRANT ALTER USER TO "BDA" ;
GRANT QUERY REWRITE TO "BDA" ;
GRANT SELECT ANY DICTIONARY TO "BDA" ;
GRANT CREATE PUBLIC SYNONYM TO "BDA" ;
GRANT GLOBAL QUERY REWRITE TO "BDA" ;
GRANT ALTER ANY CUBE DIMENSION TO "BDA" ;
GRANT CREATE ANY CUBE DIMENSION TO "BDA" ;
GRANT DROP ANY CLUSTER TO "BDA" ;
GRANT CREATE ANY RULE TO "BDA" ;
GRANT UPDATE ANY CUBE DIMENSION TO "BDA" ;
GRANT ADMINISTER RESOURCE MANAGER TO "BDA" ;
GRANT CREATE ANY SYNONYM TO "BDA" ;
GRANT DROP ANY SYNONYM TO "BDA" ;
GRANT DROP ANY MINING MODEL TO "BDA" ;
GRANT EXECUTE ANY PROCEDURE TO "BDA" ;
GRANT CREATE SYNONYM TO "BDA" ;
GRANT SET CONTAINER TO "BDA" ;
GRANT EXECUTE ANY PROGRAM TO "BDA" ;
GRANT EXEMPT REDACTION POLICY TO "BDA" ;
GRANT EXECUTE ANY TYPE TO "BDA" ;
GRANT ON COMMIT REFRESH TO "BDA" ;
GRANT DEBUG CONNECT ANY TO "BDA" ;
GRANT CREATE SEQUENCE TO "BDA" ;
GRANT CREATE HIERARCHY TO "BDA" ;
GRANT SELECT ANY MEASURE FOLDER TO "BDA" ;
GRANT COMMENT ANY MINING MODEL TO "BDA" ;
GRANT ADMINISTER SQL TUNING SET TO "BDA" ;
GRANT CREATE ANY INDEXTYPE TO "BDA" ;
```



```sql
GRANT DROP ANY INDEX TO "BDA" ;
GRANT RESTRICTED SESSION TO "BDA" ;
GRANT DEQUEUE ANY QUEUE TO "BDA" ;
GRANT ANALYZE ANY DICTIONARY TO "BDA" ;
GRANT ALTER ANY INDEXTYPE TO "BDA" ;
GRANT TRANSLATE ANY SQL TO "BDA" ;
GRANT ADMINISTER ANY SQL TUNING SET TO "BDA" ;
GRANT CREATE USER TO "BDA" ;
GRANT EXECUTE ANY OPERATOR TO "BDA" ;
GRANT CREATE CUBE BUILD PROCESS TO "BDA" ;
GRANT CREATE PROFILE TO "BDA" ;
GRANT ALTER ANY ROLE TO "BDA" ;
GRANT UPDATE ANY TABLE TO "BDA" ;
GRANT ALTER ANY LIBRARY TO "BDA" ;
GRANT DROP ANY VIEW TO "BDA" ;
GRANT CREATE ANY CLUSTER TO "BDA" ;
GRANT EXECUTE ANY RULE TO "BDA" ;
GRANT ALTER TABLESPACE TO "BDA" ;
GRANT UNDER ANY VIEW TO "BDA" ;
GRANT EXECUTE ANY ASSEMBLY TO "BDA" ;
GRANT GRANT ANY PRIVILEGE TO "BDA" ;
GRANT ALTER ANY TRIGGER TO "BDA" ;
GRANT CREATE ANY VIEW TO "BDA" ;
GRANT ALTER LOCKDOWN PROFILE TO "BDA" ;
GRANT EXPORT FULL DATABASE TO "BDA" ;
GRANT ALTER ANY MEASURE FOLDER TO "BDA" ;
GRANT ALTER ANY EVALUATION CONTEXT TO "BDA" ;
GRANT TEXT DATASTORE ACCESS TO "BDA" ;
GRANT FLASHBACK ARCHIVE ADMINISTER TO "BDA" ;
GRANT IMPORT FULL DATABASE TO "BDA" ;
GRANT CREATE ANY OUTLINE TO "BDA" ;
GRANT COMMENT ANY TABLE TO "BDA" ;
GRANT READ ANY TABLE TO "BDA" ;
GRANT CREATE DATABASE LINK TO "BDA" ;
GRANT DROP PUBLIC SYNONYM TO "BDA" ;
GRANT DROP USER TO "BDA" ;
GRANT CHANGE NOTIFICATION TO "BDA" ;
GRANT CREATE MINING MODEL TO "BDA" ;
GRANT INSERT ANY TABLE TO "BDA" ;
GRANT DROP LOCKDOWN PROFILE TO "BDA" ;
GRANT DROP PROFILE TO "BDA" ;
GRANT CREATE ANY MATERIALIZED VIEW TO "BDA" ;
GRANT CREATE RULE SET TO "BDA" ;
```

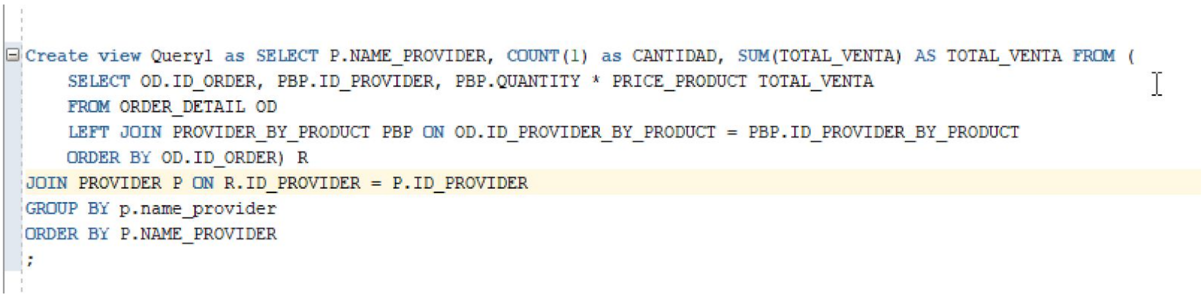


```sql
GRANT EXEMPT ACCESS POLICY TO "BDA" ;
GRANT MANAGE SCHEDULER TO "BDA" ;
GRANT READ ANY FILE GROUP TO "BDA" ;
GRANT FORCE TRANSACTION TO "BDA" ;
GRANT DROP ANY CUBE BUILD PROCESS TO "BDA" ;
GRANT ALTER ANY TYPE TO "BDA" ;
GRANT DROP ANY PROCEDURE TO "BDA" ;
GRANT CREATE ANY SQL TRANSLATION PROFILE TO "BDA" ;
GRANT DROP PUBLIC DATABASE LINK TO "BDA" ;
GRANT DROP ANY INDEXTYPE TO "BDA" ;
GRANT DROP ANY SQL PROFILE TO "BDA" ;
GRANT ALTER SYSTEM TO "BDA" ;
GRANT UNLIMITED TABLESPACE TO "BDA" ;
GRANT DROP ANY ROLE TO "BDA" ;
GRANT ALTER ANY DIMENSION TO "BDA" ;
GRANT DROP ANY CUBE DIMENSION TO "BDA" ;
GRANT DROP ANY CUBE TO "BDA" ;
GRANT CREATE ANY TRIGGER TO "BDA" ;
GRANT DROP ANY ASSEMBLY TO "BDA" ;
GRANT CREATE ANY TABLE TO "BDA" ;
GRANT ADMINISTER SQL MANAGEMENT OBJECT TO "BDA" ;
GRANT DROP ANY DIRECTORY TO "BDA" ;
GRANT ENQUEUE ANY QUEUE TO "BDA" ;
GRANT DROP ANY EVALUATION CONTEXT TO "BDA" ;
GRANT CREATE ANY ASSEMBLY TO "BDA" ;
GRANT CREATE ANY TYPE TO "BDA" ;
GRANT REDEFINE ANY TABLE TO "BDA" ;
GRANT CREATE CLUSTER TO "BDA" ;
GRANT CREATE ANY CONTEXT TO "BDA" ;
GRANT EXECUTE ANY EVALUATION CONTEXT TO "BDA" ;
GRANT RESUMABLE TO "BDA" ;
GRANT CREATE ANY LIBRARY TO "BDA" ;
GRANT DROP ANY EDITION TO "BDA" ;
GRANT CREATE PROCEDURE TO "BDA" ;
GRANT ALTER DATABASE TO "BDA" ;
GRANT SELECT ANY CUBE TO "BDA" ;
GRANT GRANT ANY ROLE TO "BDA" ;
GRANT ALTER ANY RULE TO "BDA" ;
GRANT CREATE ANY ATTRIBUTE DIMENSION TO "BDA" ;
GRANT CREATE CUBE DIMENSION TO "BDA" ;
GRANT ALTER ANY OPERATOR TO "BDA" ;
GRANT CREATE CUBE TO "BDA" ;
GRANT ALTER RESOURCE COST TO "BDA" ;
GRANT ALTER ANY HIERARCHY TO "BDA" ;
GRANT DROP TABLESPACE TO "BDA" ;
```

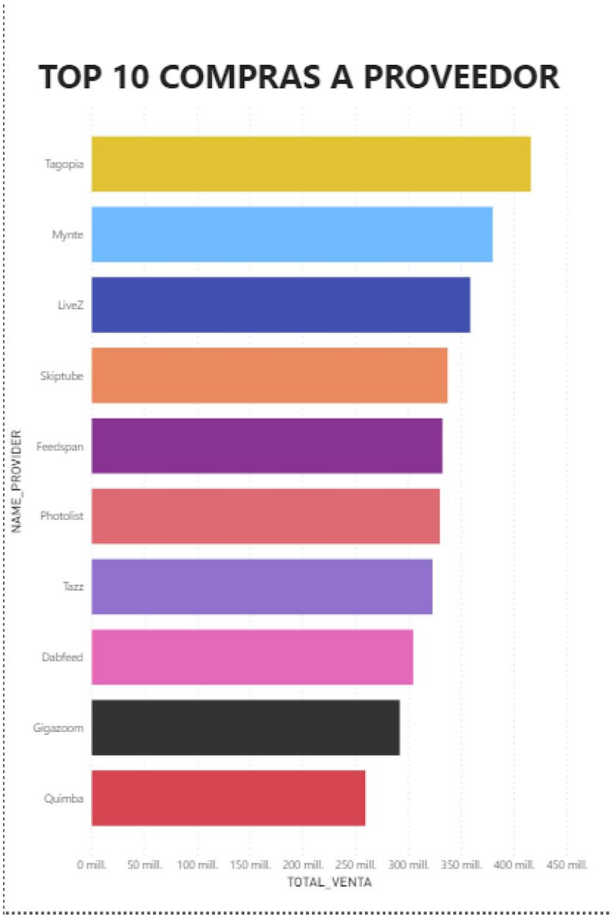
6. Módulo de Business Intelligence BI.

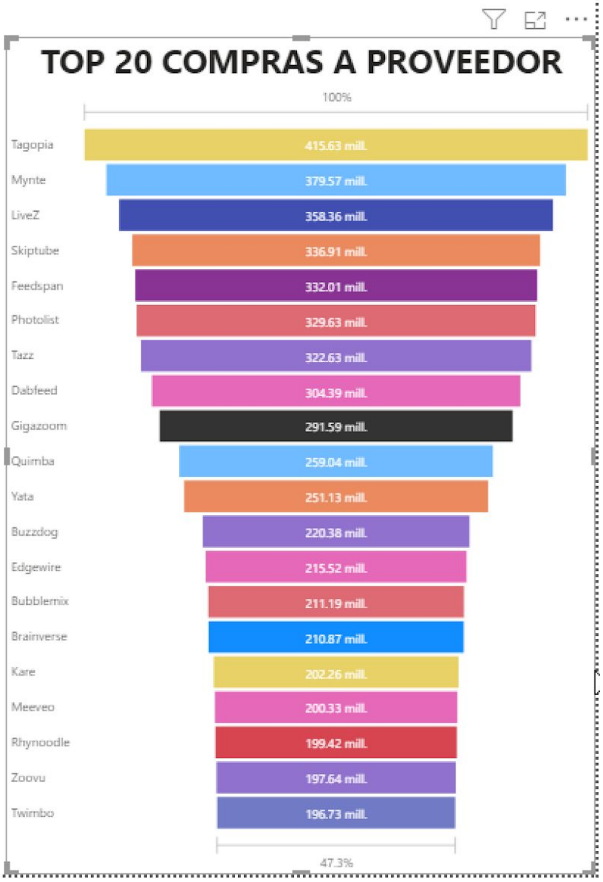
Para implementar el módulo de inteligencia de negocio Business Intelligence, utilizamos las tablas temporales creadas en nuestro ETL. Estas tablas fueron ordenadas en una vistas para poder obtener datos ordenados para implementarlos en Power BI.

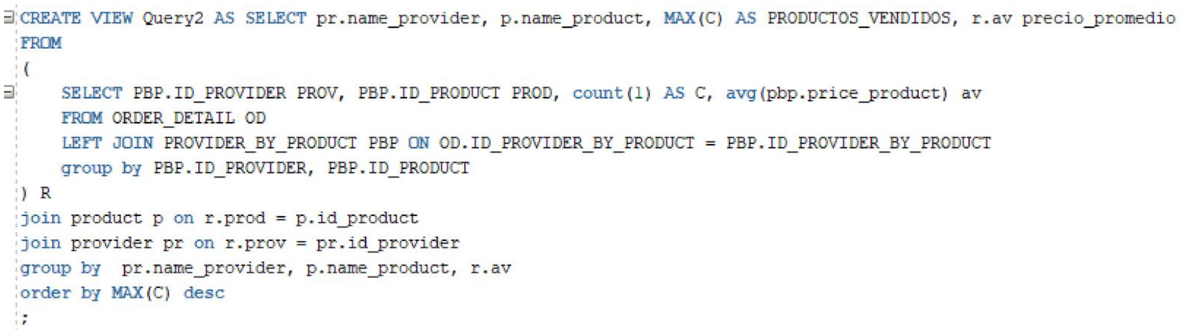
Creamos cuatro tablas temporales las cuales nos retornan valores importantes para la toma de decisiones en el negocio. por medio de la herramienta de Microsoft Power Bi pudimos implementar gráficas, estas gráficas están conectadas directamente a nuestra base de datos de reportes.



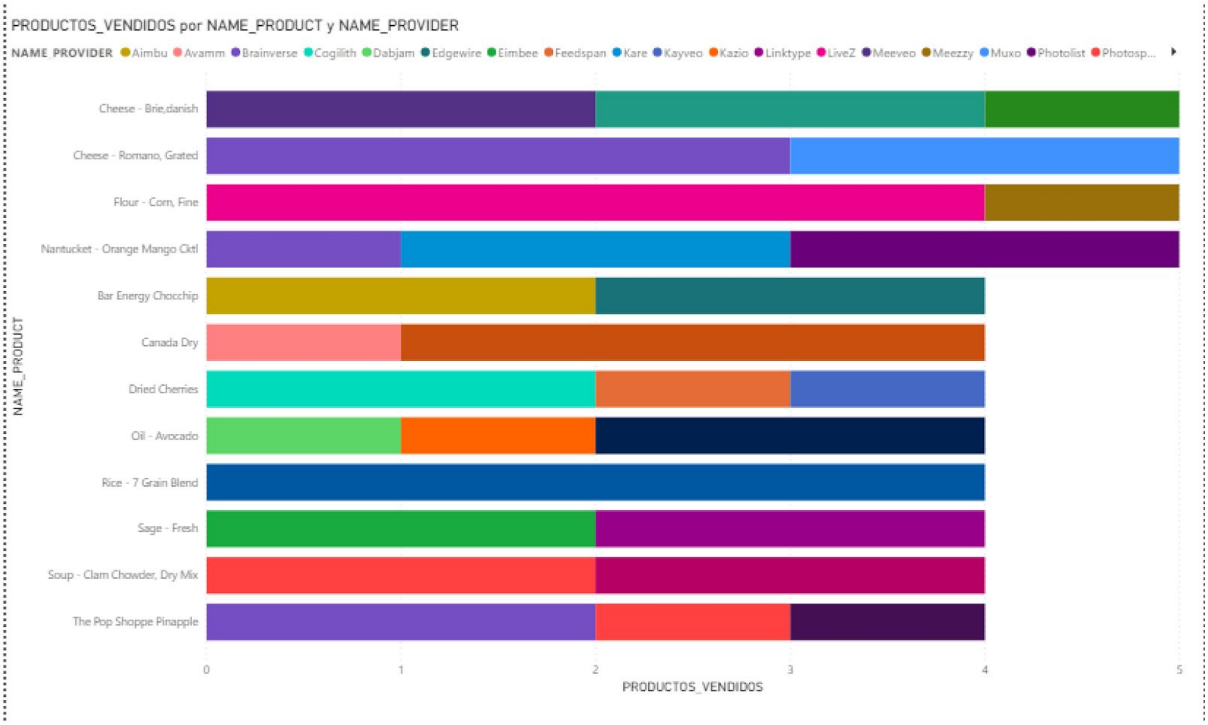
```sql
Create view Query1 as SELECT P.NAME_PROVIDER, COUNT(1) as CANTIDAD, SUM(TOTAL_VENTA) AS TOTAL_VENTA FROM (
    SELECT OD.ID_ORDER, PBP.ID_PROVIDER, PBP.QUANTITY * PRICE_PRODUCT TOTAL_VENTA
    FROM ORDER_DETAIL OD
    LEFT JOIN PROVIDER_BY_PRODUCT PBP ON OD.ID_PROVIDER_BY_PRODUCT = PBP.ID_PROVIDER_BY_PRODUCT
    ORDER BY OD.ID_ORDER) R
JOIN PROVIDER P ON R.ID_PROVIDER = P.ID_PROVIDER
GROUP BY p.name_provider
ORDER BY P.NAME_PROVIDER
;
```

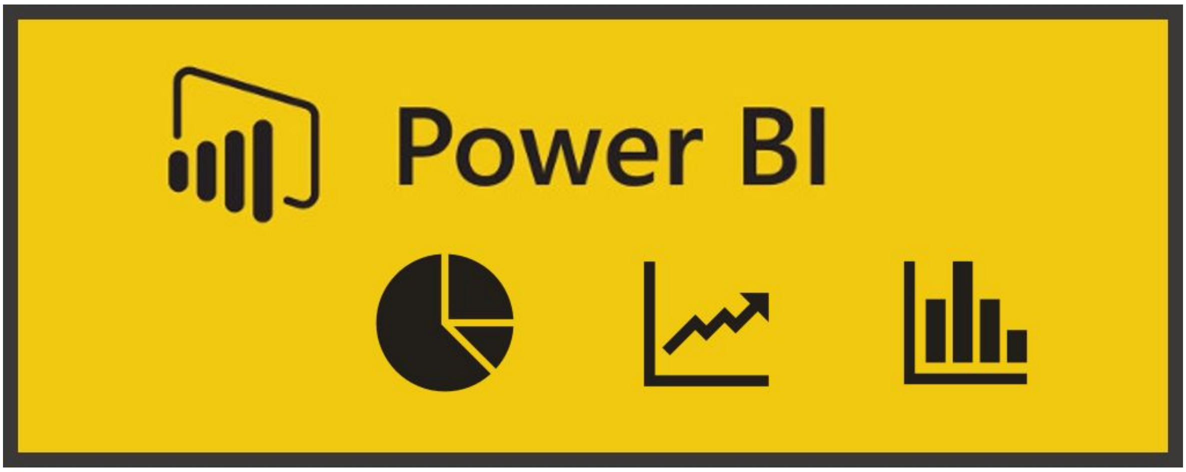


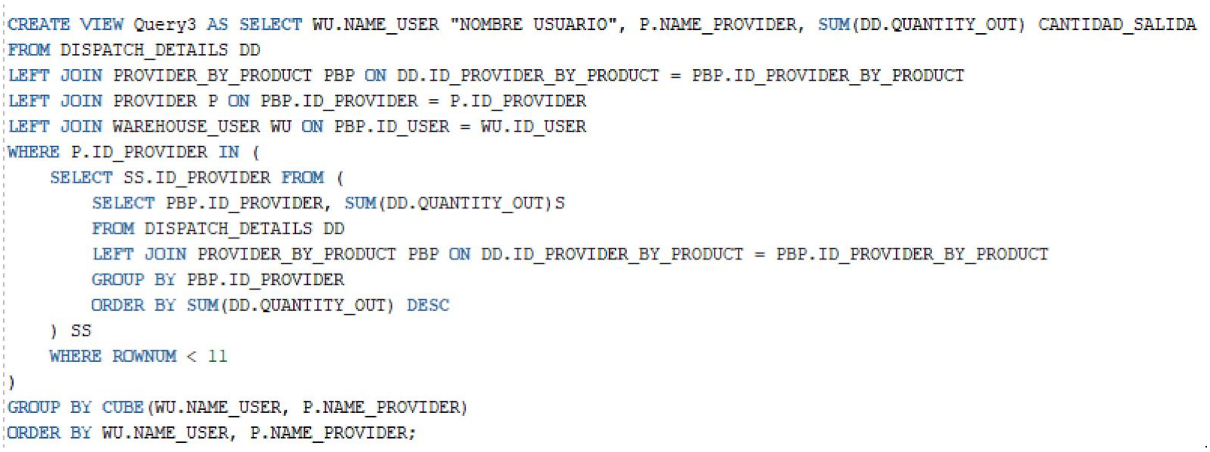




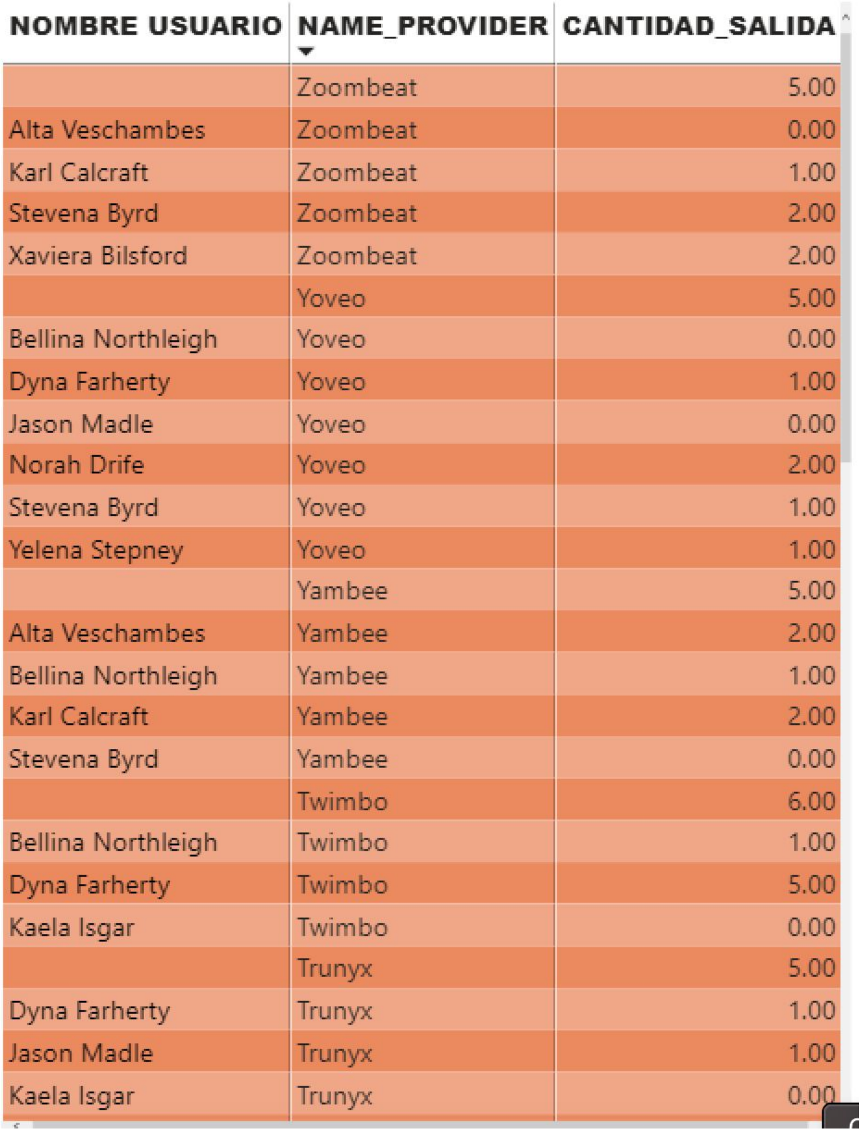
```
CREATE VIEW Query2 AS SELECT pr.name_provider, p.name_product, MAX(C) AS PRODUCTOS_VENDIDOS, r.av precio_promedio
FROM
(
    SELECT PBP.ID_PROVIDER PROV, PBP.ID_PRODUCT PROD, count(1) AS C, avg(pbp.price_product) av
    FROM ORDER_DETAIL OD
    LEFT JOIN PROVIDER_BY_PRODUCT PBP ON OD.ID_PROVIDER_BY_PRODUCT = PBP.ID_PROVIDER_BY_PRODUCT
    group by PBP.ID_PROVIDER, PBP.ID_PRODUCT
) R
join product p on r.prod = p.id_product
join provider pr on r.prov = pr.id_provider
group by  pr.name_provider, p.name_product, r.av
order by MAX(C) desc
;
```



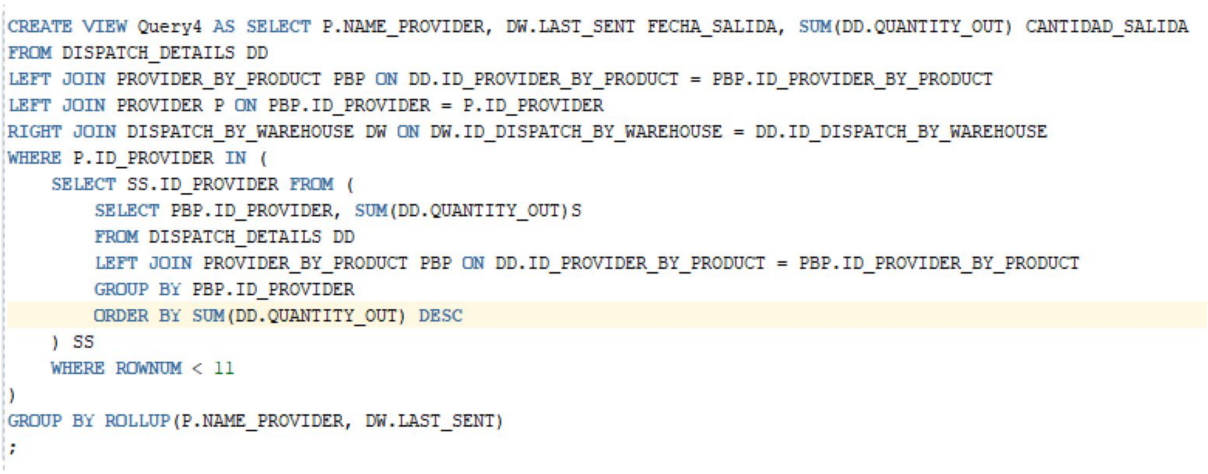




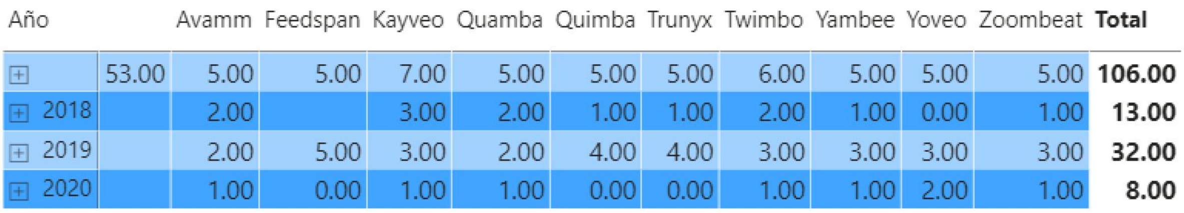
```
CREATE VIEW Query3 AS SELECT WU.NAME_USER "NOMBRE USUARIO", P.NAME_PROVIDER, SUM(DD.QUANTITY_OUT) CANTIDAD_SALIDA
FROM DISPATCH_DETAILS DD
LEFT JOIN PROVIDER_BY_PRODUCT PBP ON DD.ID_PROVIDER_BY_PRODUCT = PBP.ID_PROVIDER_BY_PRODUCT
LEFT JOIN PROVIDER P ON PBP.ID_PROVIDER = P.ID_PROVIDER
LEFT JOIN WAREHOUSE_USER WU ON PBP.ID_USER = WU.ID_USER
WHERE P.ID_PROVIDER IN (
    SELECT SS.ID_PROVIDER FROM (
        SELECT PBP.ID_PROVIDER, SUM(DD.QUANTITY_OUT)S
        FROM DISPATCH_DETAILS DD
        LEFT JOIN PROVIDER_BY_PRODUCT PBP ON DD.ID_PROVIDER_BY_PRODUCT = PBP.ID_PROVIDER_BY_PRODUCT
        GROUP BY PBP.ID_PROVIDER
        ORDER BY SUM(DD.QUANTITY_OUT) DESC
    ) SS
    WHERE ROWNUM < 11
)
GROUP BY CUBE(WU.NAME_USER, P.NAME_PROVIDER)
ORDER BY WU.NAME_USER, P.NAME_PROVIDER;
```



NOMBRE USUARIO	NAME_PROVIDER	CANTIDAD_SALIDA
	Zoombeat	5.00
Alta Veschambes	Zoombeat	0.00
Karl Calcraft	Zoombeat	1.00
Stevena Byrd	Zoombeat	2.00
Xaviera Bilsford	Zoombeat	2.00
	Yoveo	5.00
Bellina Northleigh	Yoveo	0.00
Dyna Farherty	Yoveo	1.00
Jason Madle	Yoveo	0.00
Norah Drife	Yoveo	2.00
Stevena Byrd	Yoveo	1.00
Yelena Stepney	Yoveo	1.00
	Yambee	5.00
Alta Veschambes	Yambee	2.00
Bellina Northleigh	Yambee	1.00
Karl Calcraft	Yambee	2.00
Stevena Byrd	Yambee	0.00
	Twimbo	6.00
Bellina Northleigh	Twimbo	1.00
Dyna Farherty	Twimbo	5.00
Kaela Isgar	Twimbo	0.00
	Trunyx	5.00
Dyna Farherty	Trunyx	1.00
Jason Madle	Trunyx	1.00
Kaela Isgar	Trunyx	0.00



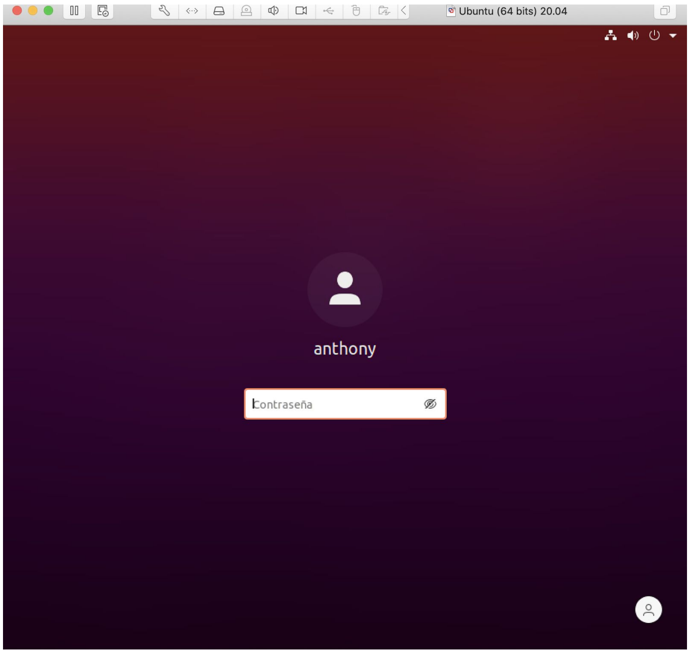
```
CREATE VIEW Query4 AS SELECT P.NAME_PROVIDER, DW.LAST_SENT FECHA_SALIDA, SUM(DD.QUANTITY_OUT) CANTIDAD_SALIDA
FROM DISPATCH_DETAILS DD
LEFT JOIN PROVIDER_BY_PRODUCT PBP ON DD.ID_PROVIDER_BY_PRODUCT = PBP.ID_PROVIDER_BY_PRODUCT
LEFT JOIN PROVIDER P ON PBP.ID_PROVIDER = P.ID_PROVIDER
RIGHT JOIN DISPATCH_BY_WAREHOUSE DW ON DW.ID_DISPATCH_BY_WAREHOUSE = DD.ID_DISPATCH_BY_WAREHOUSE
WHERE P.ID_PROVIDER IN (
    SELECT SS.ID_PROVIDER FROM (
        SELECT PBP.ID_PROVIDER, SUM(DD.QUANTITY_OUT)S
        FROM DISPATCH_DETAILS DD
        LEFT JOIN PROVIDER_BY_PRODUCT PBP ON DD.ID_PROVIDER_BY_PRODUCT = PBP.ID_PROVIDER_BY_PRODUCT
        GROUP BY PBP.ID_PROVIDER
        ORDER BY SUM(DD.QUANTITY_OUT) DESC
    ) SS
    WHERE ROWNUM < 11
)
GROUP BY ROLLUP(P.NAME_PROVIDER, DW.LAST_SENT)
;
```



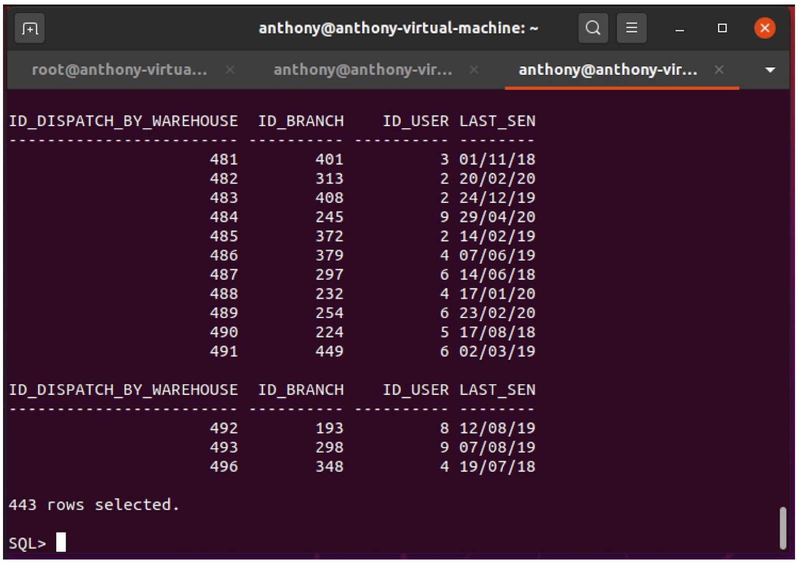
Año		Avamm	Feedspan	Kayveo	Quamba	Quimba	Trunyx	Twimbo	Yambee	Yoveo	Zoombeat	**Total**
⊞	53.00	5.00	5.00	7.00	5.00	5.00	5.00	6.00	5.00	5.00	5.00	**106.00**
⊞ 2018		2.00		3.00	2.00	1.00	1.00	2.00	1.00	0.00	1.00	**13.00**
⊞ 2019		2.00	5.00	3.00	2.00	4.00	4.00	3.00	3.00	3.00	3.00	**32.00**
⊞ 2020		1.00	0.00	1.00	1.00	0.00	0.00	1.00	1.00	2.00	1.00	**8.00**

7. Sitios de Contingencia (Backup)

Se instaló Ubuntu sobre una máquina virtual, para utilizarse como un sitio de contingencia y realizar el backup.

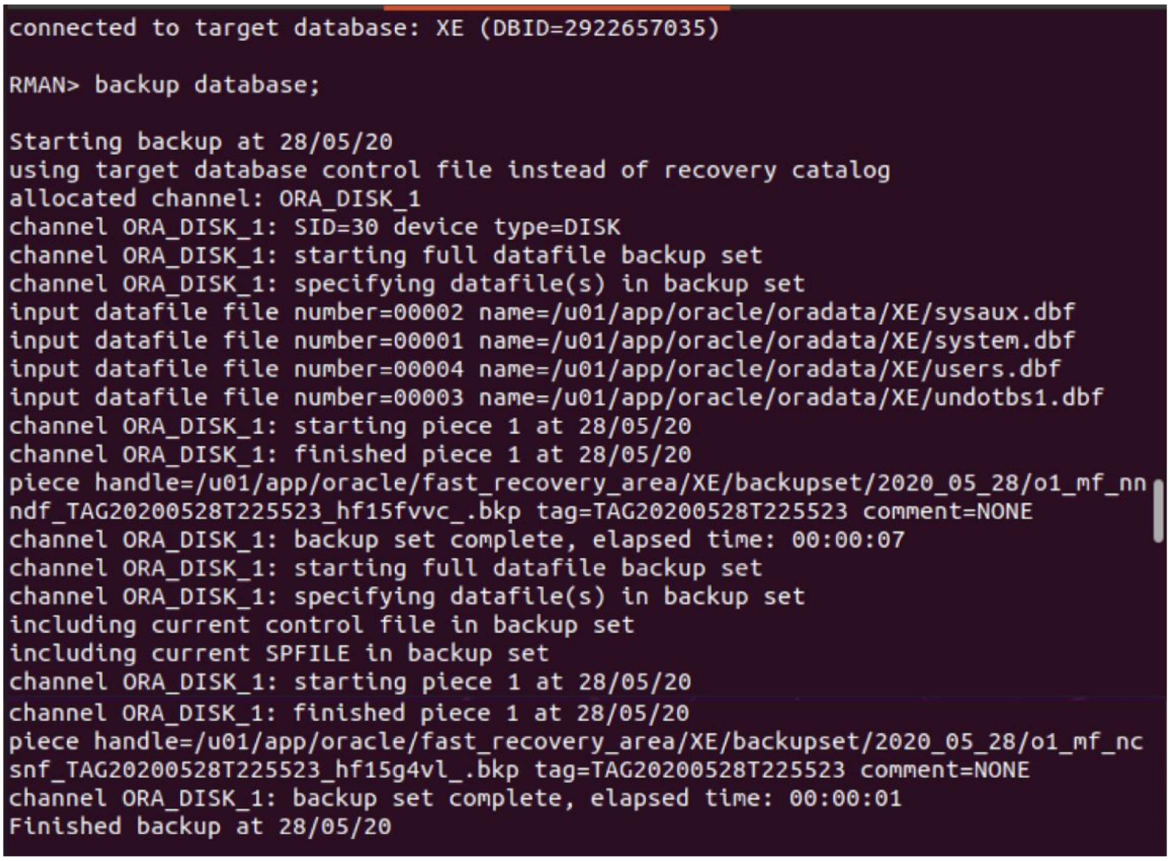


Instalamos Oracle 11g XE



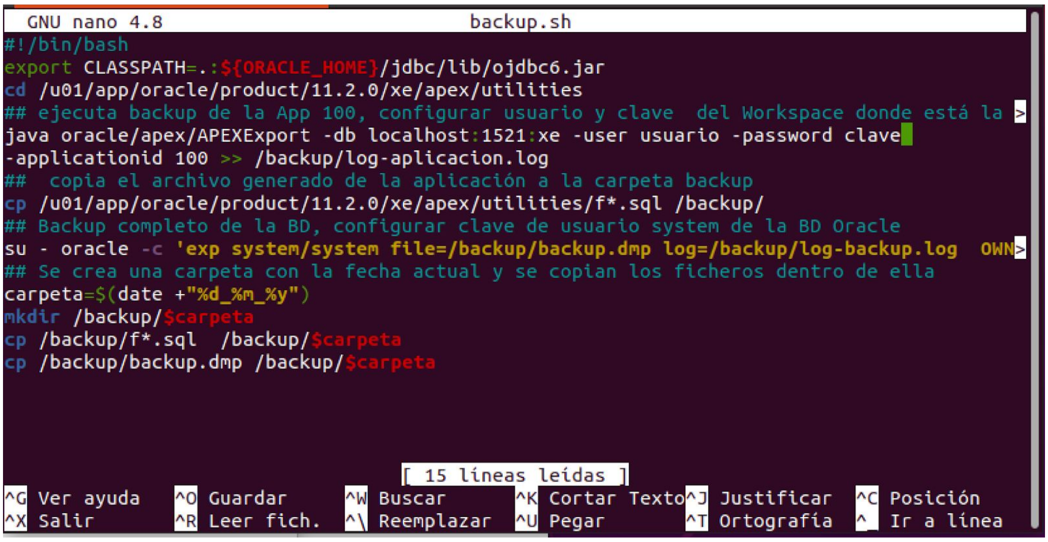
```
ID_DISPATCH_BY_WAREHOUSE  ID_BRANCH    ID_USER LAST_SEN
------------------------ ---------- ---------- --------
                     481        401          3 01/11/18
                     482        313          2 20/02/20
                     483        408          2 24/12/19
                     484        245          9 29/04/20
                     485        372          2 14/02/19
                     486        379          4 07/06/19
                     487        297          6 14/06/18
                     488        232          4 17/01/20
                     489        254          6 23/02/20
                     490        224          5 17/08/18
                     491        449          6 02/03/19

ID_DISPATCH_BY_WAREHOUSE  ID_BRANCH    ID_USER LAST_SEN
------------------------ ---------- ---------- --------
                     492        193          8 12/08/19
                     493        298          9 07/08/19
                     496        348          4 19/07/18

443 rows selected.

SQL>
```

Se utilizó RMAN para probar el backup



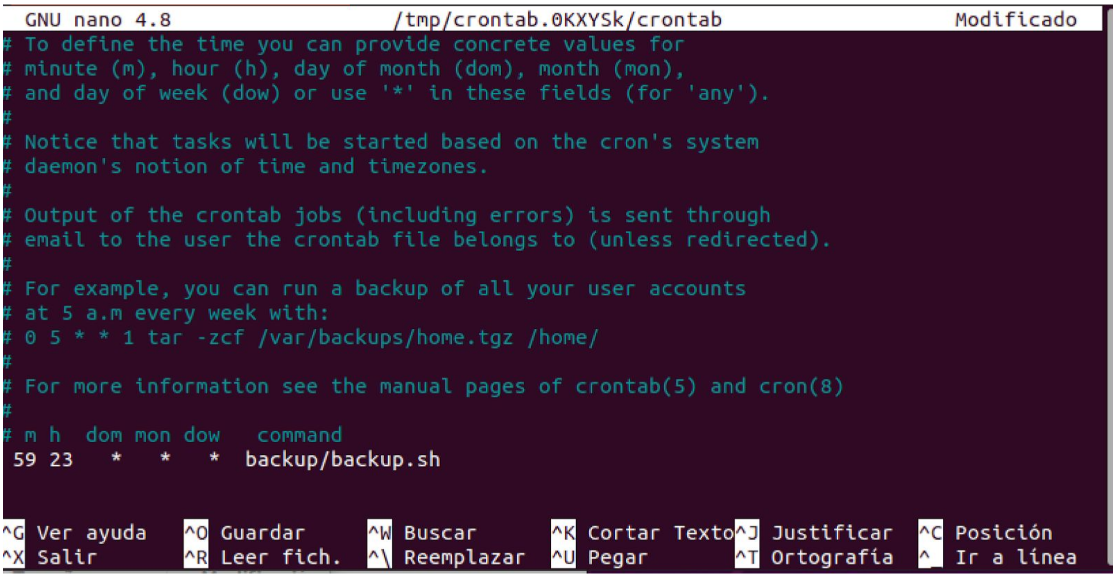
```
connected to target database: XE (DBID=2922657035)

RMAN> backup database;

Starting backup at 28/05/20
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=30 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00002 name=/u01/app/oracle/oradata/XE/sysaux.dbf
input datafile file number=00001 name=/u01/app/oracle/oradata/XE/system.dbf
input datafile file number=00004 name=/u01/app/oracle/oradata/XE/users.dbf
input datafile file number=00003 name=/u01/app/oracle/oradata/XE/undotbs1.dbf
channel ORA_DISK_1: starting piece 1 at 28/05/20
channel ORA_DISK_1: finished piece 1 at 28/05/20
piece handle=/u01/app/oracle/fast_recovery_area/XE/backupset/2020_05_28/o1_mf_nn
ndf_TAG20200528T225523_hf15fvvc_.bkp tag=TAG20200528T225523 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:07
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current control file in backup set
including current SPFILE in backup set
channel ORA_DISK_1: starting piece 1 at 28/05/20
channel ORA_DISK_1: finished piece 1 at 28/05/20
piece handle=/u01/app/oracle/fast_recovery_area/XE/backupset/2020_05_28/o1_mf_nc
snf_TAG20200528T225523_hf15g4vl_.bkp tag=TAG20200528T225523 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 28/05/20
```

Se, creó y configuró backup.sh

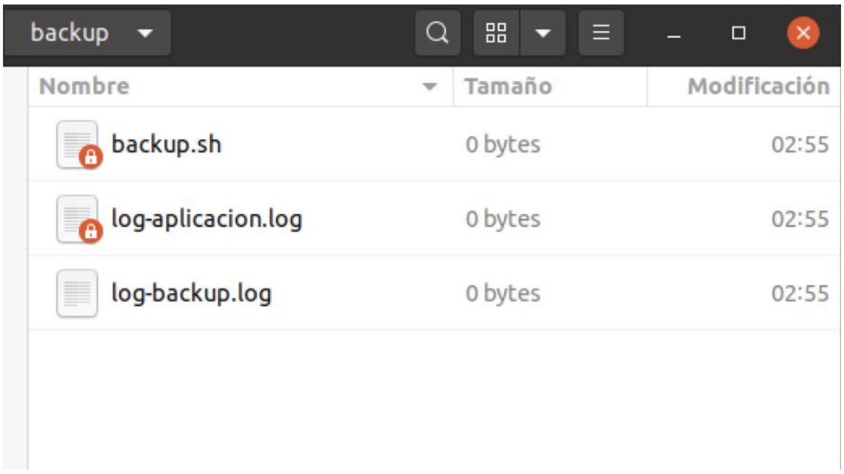


```
  GNU nano 4.8                              backup.sh
#!/bin/bash
export CLASSPATH=.:${ORACLE_HOME}/jdbc/lib/ojdbc6.jar
cd /u01/app/oracle/product/11.2.0/xe/apex/utilities
## ejecuta backup de la App 100, configurar usuario y clave  del Workspace donde está la >
java oracle/apex/APEXExport -db localhost:1521:xe -user usuario -password clave
-applicationid 100 >> /backup/log-aplicacion.log
##  copia el archivo generado de la aplicación a la carpeta backup
cp /u01/app/oracle/product/11.2.0/xe/apex/utilities/f*.sql /backup/
## Backup completo de la BD, configurar clave de usuario system de la BD Oracle
su - oracle -c 'exp system/system file=/backup/backup.dmp log=/backup/log-backup.log  OWN>
## Se crea una carpeta con la fecha actual y se copian los ficheros dentro de ella
carpeta=$(date +"%d_%m_%y")
mkdir /backup/$carpeta
cp /backup/f*.sql  /backup/$carpeta
cp /backup/backup.dmp /backup/$carpeta



                                   [ 15 líneas leídas ]
^G Ver ayuda    ^O Guardar      ^W Buscar       ^K Cortar Texto^J Justificar    ^C Posición
^X Salir        ^R Leer fich.   ^\ Reemplazar   ^U Pegar       ^T Ortografía    ^_ Ir a línea
```

Se configuró el backup para realizaste una vez al día



```
  GNU nano 4.8                  /tmp/crontab.0KXYSk/crontab                        Modificado
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
 59 23   *    *   *  backup/backup.sh

^G Ver ayuda      ^O Guardar       ^W Buscar        ^K Cortar Texto^J Justificar    ^C Posición
^X Salir          ^R Leer fich.    ^\ Reemplazar    ^U Pegar       ^T Ortografía    ^_ Ir a línea
```

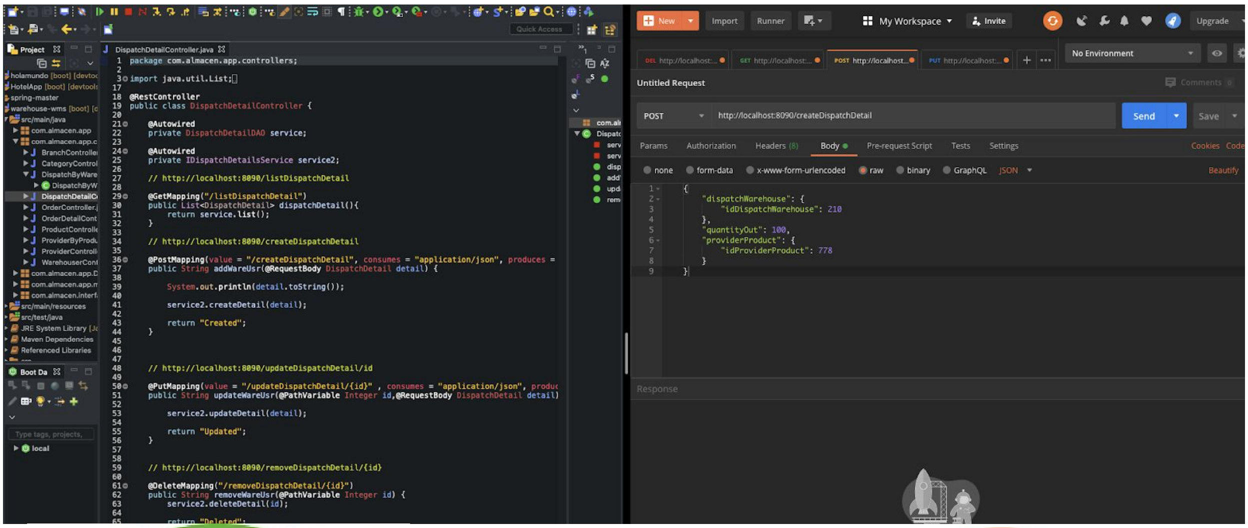


Nombre	Tamaño	Modificación
backup.sh	0 bytes	02:55
log-aplicacion.log	0 bytes	02:55
log-backup.log	0 bytes	02:55

Aspectos Técnicos:

1. **Capas de Servidores.**
   El proyecto se distribuye en un servidor de aplicaciones tomcat, el uso de docker para virtualizar oracle y windows.

2. **Nivel Web.**
   Se implementó un servicio Restful construido en java usando el framework Spring boot.



```java
package com.almacen.app.controllers;

import java.util.List;

@RestController
public class DispatchDetailController {

    @Autowired
    private DispatchDetailDAO service;

    @Autowired
    private IDispatchDetailsService service2;

    // http://localhost:8090/listDispatchDetail

    @GetMapping("/listDispatchDetail")
    public List<DispatchDetail> dispatchDetail(){
        return service.list();
    }

    // http://localhost:8090/createDispatchDetail

    @PostMapping(value = "/createDispatchDetail", consumes = "application/json", produces =
    public String addWareUsr(@RequestBody DispatchDetail detail) {

        System.out.println(detail.toString());

        service2.createDetail(detail);

        return "Created";
    }



    // http://localhost:8090/updateDispatchDetail/id

    @PutMapping(value = "/updateDispatchDetail/{id}" , consumes = "application/json", produc
    public String updateWareUsr(@PathVariable Integer id,@RequestBody DispatchDetail detail)

        service2.updateDetail(detail);

        return "Updated";
    }


    // http://localhost:8090/removeDispatchDetail/{id}

    @DeleteMapping("/removeDispatchDetail/{id}")
    public String removeWareUsr(@PathVariable Integer id) {
        service2.deleteDetail(id);

        return "Deleted";
```

Untitled Request

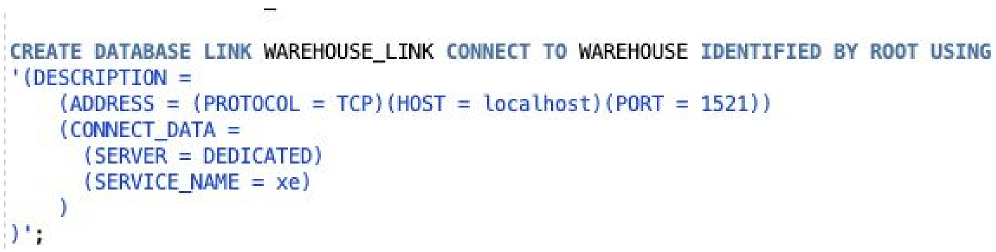
POST http://localhost:8090/createDispatchDetail

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

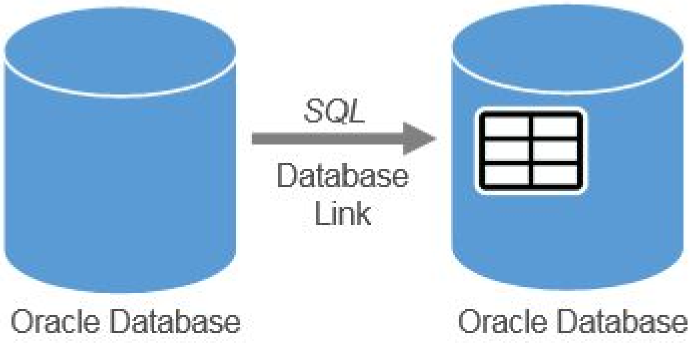
```json
{
    "dispatchWarehouse": {
        "idDispatchWarehouse": 210
    },
    "quantityOut": 100,
    "providerProduct": {
        "idProviderProduct": 778
    }
}
```

3. **DB-Links.**

Se crea un DBLINK Oracle a Oracle, para la creación de ETL para la reportería en el modulo BI. El DBLINK es creado desde la base de datos warehouse reportes a la base de datos principal.



```
CREATE DATABASE LINK WAREHOUSE_LINK CONNECT TO WAREHOUSE IDENTIFIED BY ROOT USING
'(DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = xe)
    )
)';
```



Detalle de horas hombre y costos del proyecto

Costos del proyecto

Nombre Servicio.	Precio mensual en dólares.
PowerBI	$9.99
Oracle 12c	$350
Linux VPS	$20
Total	**$379.99**

Detalle horas hombre
Costo por hora $18 dólares

Tarea	Duración horas
Análisis	5 hrs
E.R	3 hrs
Restful	10 hrs
Procedimientos ABC	5 hrs
Triggers inventario	3 hrs
DBLINKS	2 hrs
PowerBI	8 hrs
Sitios de contención y seguridad	12 hrs
ETL	2 hrs
Manejo de excepciones	4 hrs
Pruebas	5 hrs
Documentación	3 hrs
Despliegue	2 hrs
Total	64 hrs

GIT:
https://github.com/Celsorojasv/warehouse-wms-

SQL ENTREGA:
https://github.com/Celsorojasv/warehouse-wms-/tree/master/SQL%20ENTREGA