

Informe de Migración de Base de Datos PostgreSQL a Oracle

1. Resumen del Proceso

La migración se realizó desde **PostgreSQL 17.6**. hacia **Oracle Database 23.9 Free**
El procedimiento constó de:

- Instalación de ambas bases de datos.
- Creación de las tablas en Oracle ajustando la estructura (tipos de datos equivalentes).
- Exportación de datos desde Postgre en formato **CSV**.
- Importación de los CSV a Oracle.
- Conversión de valores incompatibles (ejemplo: campos booleanos **PostgreSQL 17.6**. T/F en Postgre ajustados a 1/0 en **Oracle**).
- Validación y testeo final con consultas.

2. Instalación y Configuración

Oracle Database

1. Descargar e instalar **Oracle Database 23.9 Free**.
2. Crear un usuario de migración (ejemplo: mig_user).
3. Crear una base de datos destino, en nuestro caso, retail
Comandos SQL:
CREATE DATABASE retail;
4. Otorgarle permisos necesarios de conexión al nuevo usuario y importación.

```
CREATE USER C##migrador1 IDENTIFIED BY migrador
```

```
DEFAULT TABLESPACE users
```

```
TEMPORARY TABLESPACE temp
```

```
QUOTA UNLIMITED ON users;
```

```
GRANT
```

```
CREATE SESSION,
```

```
CREATE TABLE,
```

```
CREATE VIEW,
```

```
CREATE SEQUENCE,  
CREATE TRIGGER,  
CREATE PROCEDURE,  
CREATE SYNONYM,  
TO C##migrador1;
```

```
GRANT READ, WRITE ON DIRECTORY DATA_PUMP_DIR TO  
C##migrador1;PostgreSQL
```

1. Descargar e instalar **PostgreSQL 15** y **pgAdmin 4**.
2. Tener las tablas y los datos que se van a migrar.

3. Ajustar la estructura de Tablas

Se identificaron diferencias en tipos de datos:

- VARCHAR2 (Oracle) → VARCHAR (PostgreSQL)
- NUMBER → NUMERIC o INTEGER
- DATE → DATE
- CHAR(1) para booleanos → BOOLEAN

Ejemplo de tabla ajustada:

-- Oracle

```
CREATE TABLE clientes (  
id_cliente NUMBER PRIMARY KEY,  
nombre VARCHAR2(100),  
activo CHAR(1)  
);
```

-- PostgreSQL

```
CREATE TABLE clientes (  
id_cliente SERIAL PRIMARY KEY,  
nombre VARCHAR(100),  
activo BOOLEAN
```

);

4. Migración con CSV

1. Exportación desde Postgresql en formato CSV.
2. Importación de datos en Oracle con la herramienta importación de datos

Par ello usamos:

-- Categorías (con booleano estado convertido a 1/0)

```
\COPY (  
    SELECT id_categoria, nombre, descripcion,  
           CASE WHEN estado = true THEN 1 ELSE 0 END AS estado  
    FROM categorias  
) TO 'C:\data\categorias.csv' DELIMITER ';' CSV HEADER;
```

```
\COPY proveedores TO 'C:\data\proveedores.csv' DELIMITER ';' CSV HEADER;
```

```
\COPY (  
    SELECT id_producto, nombre, descripcion, id_categoria, id_proveedor,  
           precio_compra, precio_venta, stock_minimo,  
           CASE WHEN estado = true THEN 1 ELSE 0 END AS estado  
    FROM productos  
) TO 'C:\data\productos.csv' DELIMITER ';' CSV HEADER;
```

```
\COPY tiendas TO 'C:\data\tiendas.csv' DELIMITER ';' CSV HEADER;
```

```
\COPY inventario_tienda TO 'C:\data\inventario_tienda.csv' DELIMITER ';' CSV  
HEADER;
```

```
\COPY clientes TO 'C:\data\clientes.csv' DELIMITER ';' CSV HEADER;
```

```
\COPY empleados TO 'C:\data\empleados.csv' DELIMITER ';' CSV HEADER;
```

```
\COPY ventas TO 'C:\data\ventas.csv' DELIMITER ';' CSV HEADER;
```

```
\COPY detalle_venta TO 'C:\data\detalle_venta.csv' DELIMITER ';' CSV HEADER;
```

```
\COPY movimientos_inventario TO 'C:\data\movimientos_inventario.csv'  
DELIMITER ';' CSV HEADER;
```

\COPY auditoria_precios TO 'C:\data\auditoria_precios.csv' DELIMITER ',' CSV
HEADER;

- Estos comandos harán que los datos se exporten automáticamente en archivos
csv en la carpeta asignada

5. Comparativo Oracle vs PostgreSQL

Aspecto	Oracle	PostgreSQL
Tipos de datos boolean	No nativo (CHAR(1) o NUMBER)	Nativo (BOOLEAN)
Auto incremento	SEQUENCE + TRIGGER	SERIAL o GENERATED
Herramienta de gestión	SQL Developer	pgAdmin
Exportación de datos	Data Pump, SQL*Loader	COPY, pg_dump, CSV

6. Testeo con Consultas

Una vez migrados los datos, se validaron con consultas utilizadas en Postgre, pero
transformadas a la sintáxis de Oracle, por ejemplo:

/*Top 10 productos más vendidos por tienda*/

SELECT *

FROM (

SELECT

t.nombre AS tienda,

p.nombre AS producto,

SUM(dv.cantidad) AS total_vendido

FROM detalle_venta dv

JOIN productos p ON dv.id_producto = p.id_producto

JOIN ventas v ON dv.id_venta = v.id_venta

JOIN tiendas t ON v.id_tienda = t.id_tienda

GROUP BY t.nombre, p.nombre

ORDER BY t.nombre, total_vendido DESC

)

WHERE ROWNUM <= 10;

/*Análisis de rentabilidad por categoría de producto*/

SELECT

c.nombre AS categoria,

SUM(dv.cantidad * dv.precio_unitario) AS ingresos,

SUM(dv.cantidad * p.precio_compra) AS costos,

SUM(dv.cantidad * dv.precio_unitario) - SUM(dv.cantidad * p.precio_compra) AS
rentabilidad

FROM detalle_venta dv

JOIN productos p ON dv.id_producto = p.id_producto

JOIN categorias c ON p.id_categoria = c.id_categoria

GROUP BY c.nombre

ORDER BY rentabilidad DESC;

7. Conclusión

La migración fue exitosa tras ajustar tipos de datos y valores booleanos. PostgreSQL mostró compatibilidad con la mayoría de estructuras de Oracle, aunque requirió ajustes manuales en campos booleanos y auto-incrementales.

8. LINK DEL VIDEO DE MIGRACIÓN:

[https://youtu.be/LweStLAC](https://youtu.be/LweStLACyko)

[yko](https://youtu.be/LweStLACyko)