



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
CARRERA DE INGENIERIA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

Laboratorio de:

Fundamentos de Bases de Datos

Práctica No.: 3

Tema: Modelamiento Físico en PowerDesigner - Sistema de Proveedores y Productos

Objetivos:

1. Transformar el modelo lógico del Laboratorio 2 a un modelo físico específico para Oracle 19c
2. Aplicar criterios de diseño físico: tipos de datos, restricciones de integridad, índices y optimización
3. Utilizar PowerDesigner como herramienta CASE para la generación del modelo físico
4. Generar scripts SQL DDL completos y funcionales para Oracle 19c
5. Validar la implementación física mediante pruebas de integridad y rendimiento

Marco teórico:

1. Diferencias Modelo Lógico vs. Modelo Físico

Característica	Modelo Lógico	Modelo Físico
Nivel de abstracción	Alto	Bajo
Qué representa	Entidades, relaciones, atributos	Tablas físicas, índices, tipos de datos, almacenamiento
Enfoque	Diseño del "qué" (estructura)	Implementación del "cómo" (rendimiento, hardware)

2. Consideraciones de Implementación en el Modelo Físico

SGBD	Sintaxis y funciones específicas, tipos soportados
Rendimiento	Índices, particiones, optimización consulta
Almacenamiento	Cómo y dónde se guardan los datos
Seguridad y backup	Definición de roles, respaldos
Integridad y restricciones	A nivel de columnas/campos

3. Tipos de Datos en Oracle 19c vs. Otros SGBD

Tipo Oracle	Descripción	Límite principal	Relación con otros SGBD
VARCHAR2	Cadena variable (caracteres)	4000 bytes	VARCHAR/TEXT (MySQL, SQL Server)
NUMBER	Números y decimales	Precisión: hasta 38 dígitos	NUMERIC, DECIMAL, FLOAT (otros SGBD)
DATE	Fecha (año-mes-día, hora-minuto-segundo)	Hasta el 31-12-9999	DATE, DATETIME (otros SGBD)



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

TIMESTAMP	Igual a DATE + decimales de segundo	Hasta 9 decimales segundos	TIMESTAMP (otros SGBD)
CLOB	Texto muy grande (documentos)	4 GB	TEXT, NTEXT (otros SGBD)
BLOB	Binarios grandes (imágenes, PDFs)	4 GB	IMAGE, VARBINARY (otros SGBD)

4. Restricciones de Integridad en Oracle

Restricción	Uso principal	Sintaxis/Palabra clave	Ejemplo de uso
PRIMARY KEY	Identidad única en cada fila	PRIMARY KEY	PRIMARY KEY (id)
FOREIGN KEY	Relación con clave de otra tabla	FOREIGN KEY (ON DELETE/UPDATE)	FOREIGN KEY (cliente_id) REFERENCES clientes(id) ON DELETE CASCADE
CHECK	Restricción de valores permitidos en columnas	CHECK	CHECK (edad >= 18)
NOT NULL	Impide valores nulos	NOT NULL	nombre VARCHAR2(50) NOT NULL
DEFAULT	Valor por defecto en ausencia de dato	DEFAULT	fecha DATE DEFAULT SYSDATE
UNIQUE	Unicidad de valores en columna(s)	UNIQUE	UNIQUE (correo)

Políticas de integridad referencial:

Política	Efecto
ON DELETE RESTRICT	Prohíbe eliminar registros "padre" referenciados
ON DELETE CASCADE	Elimina en cascada los registros "hijo" relacionados
ON UPDATE RESTRICT	No permite actualizar claves referenciadas

5. Índices en Oracle

Tipo de Índice	Aplicación principal	Características
B-tree (por defecto)	Consultas generales	Eficiente en búsqueda por valores únicos/muchos distintos
Bitmap	Consultas en columnas con baja cardinalidad (pocos valores únicos)	Muy eficiente para combinar múltiples filtros (AND/OR); ocupa poco espacio en columnas aptas
Function-based	Consultas en columnas derivados por función	Permite indexar resultados de funciones aplicadas a columnas
Simple	Una sola columna	
Compuesto	Varias columnas	Orden relevante; útil en combinaciones comunes de búsqueda

6. Impacto de los Índices en el Rendimiento

Operación	¿Mejora con índice?	
SELECT	Sí	Índices permiten búsquedas/matches mucho más rápidas



**ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL**

INSERT	No	Índices deben actualizarse, haciendo el proceso más lento
UPDATE	No	Si columnas indexadas se tocan, el índice se recalcula
DELETE	No	Elimina y actualiza entradas en el índice

Desarrollo de la práctica:

CONTEXTO DEL PROBLEMA

Una empresa comercial desea diseñar una Base de Datos para llevar un control eficiente de sus proveedores, productos y categorías. Los datos significativos a tener en cuenta son:

Un proveedor se define por su código de proveedor (único), razón social, dirección, teléfono, celular, nombre y apellido paterno del representante.

Los productos que comercializa la empresa, cada uno se define por su código de producto (único), nombre del producto, descripción del producto, costo unitario de compra (costoUC), costo unitario de venta (costoUV) y stock disponible.

Hay que tener en cuenta que existen diferentes categorías de productos (Electrónica, Alimentos, Vestimenta, Ferretería, ...), y cada una se define por un código de categoría (único) y el nombre de la categoría.

Consideraciones de diseño

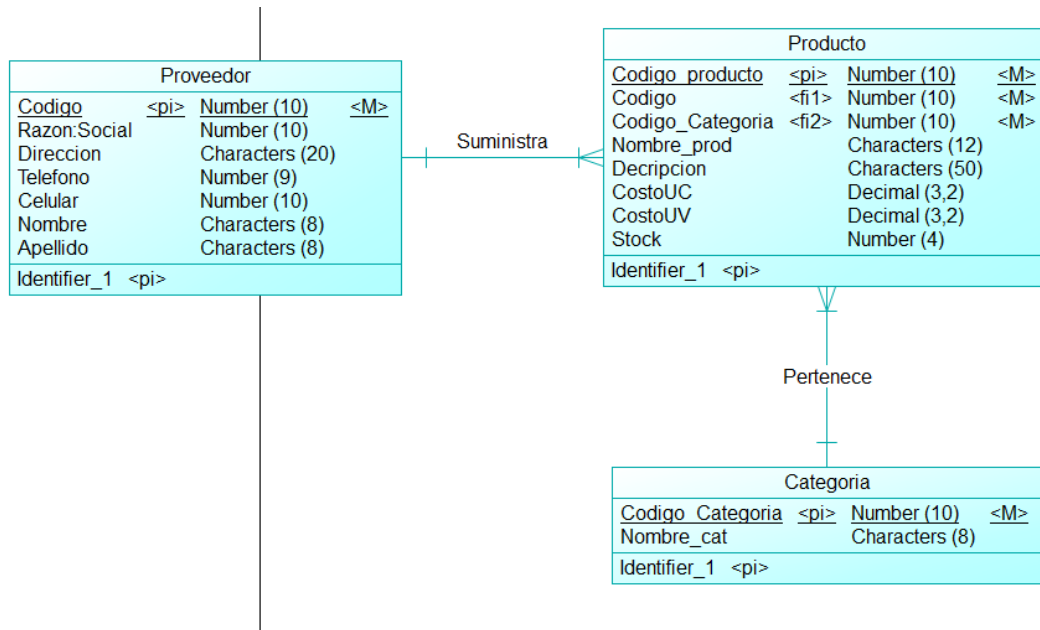
Un proveedor puede proveer varios productos a la empresa. Un producto es suministrado por un único proveedor durante el periodo de relación comercial. Todos los productos deben pertenecer obligatoriamente a una categoría específica. Una categoría agrupa múltiples productos relacionados del mismo tipo. Cada producto debe tener registrado su proveedor correspondiente y su categoría asignada para efectos de inventario y control de stock.

Es importante mantener actualizado el stock de cada producto, así como sus costos de compra y venta para el cálculo de márgenes de ganancia. La información de contacto del proveedor (teléfono, celular y dirección) debe estar disponible para facilitar los procesos de reabastecimiento.

Esquema lógico base



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

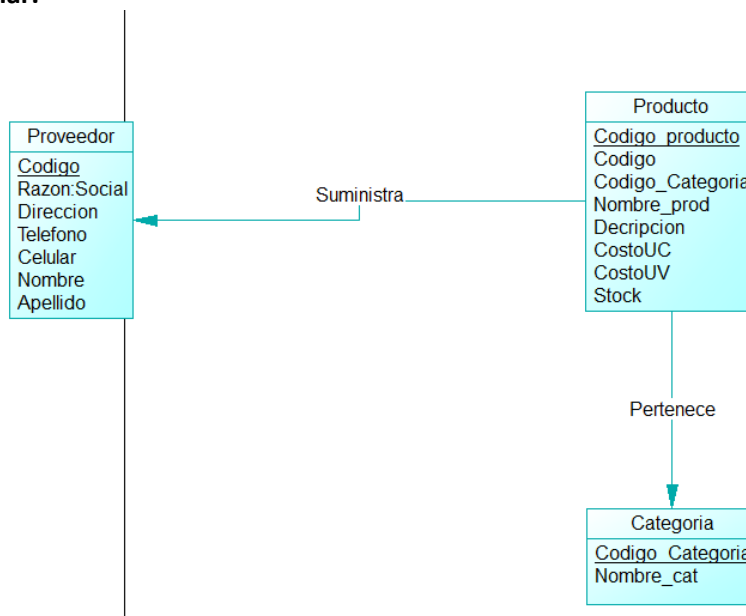


PARTE 1: Generación del Modelo Físico en PowerDesigner

Procedimiento

1. Abra el modelo lógico (LDM) del Sistema de Proveedores del Laboratorio 2
2. Seleccione: Tools → Generate Physical Data Model
3. Configure:
Name: Proveedores_Productos_PDM
DBMS: Oracle Database 19c
☒ Generate new Physical Data Model
4. Click en OK

Resultado: **PowerDesigner** creará un PDM con una transformación inicial básica que deberá **refinar**.





ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

PARTE 2: Definición de Tipos de Datos Físicos

Para cada tabla, defina los tipos de datos apropiados considerando:

- Naturaleza de los datos
- Rango de valores esperados
- Eficiencia de almacenamiento
- Operaciones previstas

Tabla PROVEEDOR

	Name	Code	Data Type	Leng	Preci
1	Cod_proveed	COD_PROVE	VARCHAR2(10)	10	
2	Razon:Social	RAZON_SOC	VARCHAR2(200)	200	
3	Direccion	DIRECCION	VARCHAR2(300)	300	
4	Telefono	TELEFONO	VARCHAR2(15)	15	
5	Celular	CELULAR	VARCHAR2(15)	15	
6	Nombre	NOMBRE	VARCHAR2(100)	100	
→	Apellido	APELLIDO	VARCHAR2(1	100	

Tabla CATEGORIA

	Name	Code	Data Type	Length
1	Codigo_Categoria	CODIGO_CATEGORIA	VARCHAR2(10)	10
→	Nombre_cat	NOMBRE_CAT	VARCHAR2(100)	100

Tabla PRODUCTO

	Name	Code	Data Type
1	Codigo_producto	CODIGO_PRODUCTO	VARCHAR2(10)
2	Cod_proveedor	COD_PROVEEDOR	VARCHAR2(200)
3	Codigo_Categoria	CODIGO_CATEGORIA	VARCHAR2(10)
4	Nombre_prod	NOMBRE_PROD	VARCHAR2(200)
→	Decripcion	DECRIPCION	VARCHAR2(500)
6	CostoUC	COSTOUC	NUMBER(3,2)
7	CostoUV	COSTOUV	NUMBER(3,2)
8	Stock	STOCK	NUMBER(10)

PARTE 3: Restricciones de Integridad

A) Claves Primarias (PRIMARY KEY)

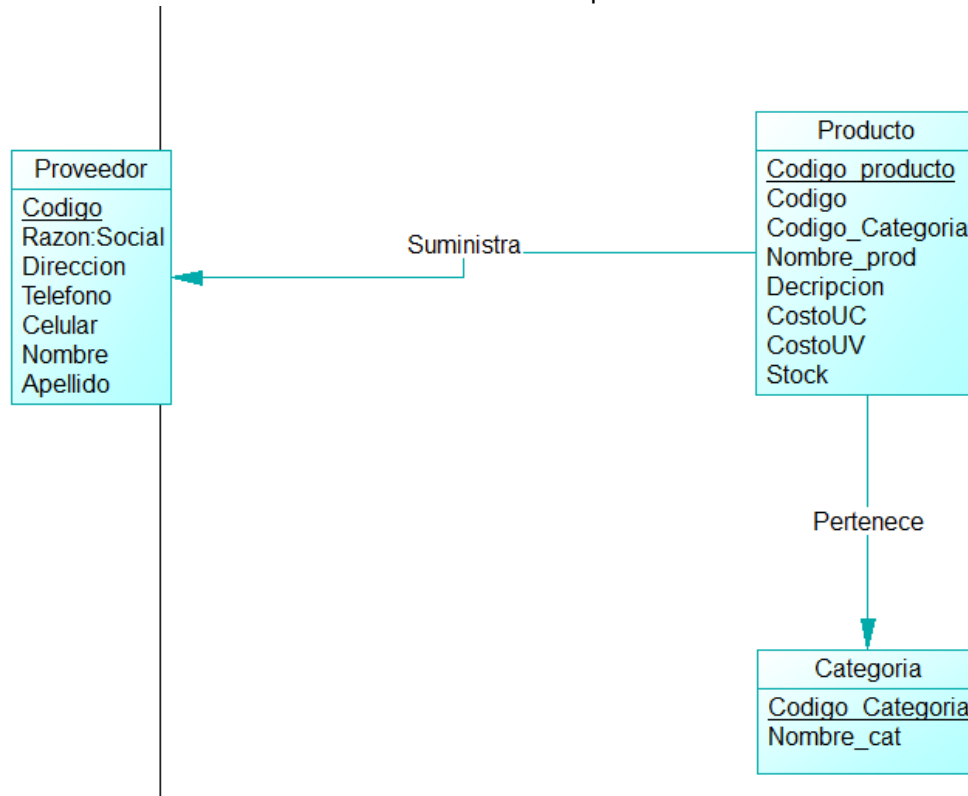
PowerDesigner las genera automáticamente, pero verifique:

En cada tabla:



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

PROVEEDOR: PK_PROVEEDOR en cod_proveedor CATEGORIA: PK_CATEGORIA en cod_categoria
PRODUCTO: PK_PRODUCTO en cod_producto
Oracle crea automáticamente un índice único para cada PK.



Las PK, son los elementos subrayados

B) Claves Foráneas (FOREIGN KEY)

Configurar en PRODUCTO:

FK hacia PROVEEDOR

1. Doble clic en la relación (línea entre PRODUCTO y PROVEEDOR)
2. Configure:
Constraint Name: FK_PRODUCTO_PROVEEDOR
Delete Rule: RESTRICT (o NO ACTION)
Update Rule: CASCADE (opcional en Oracle, se simula con triggers)

Significado:

RESTRICT: No permite eliminar un proveedor si tiene productos asociados

CASCADE: Si se actualiza cod_proveedor, se actualiza automáticamente en productos



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

Child Table	Parent Table
Producto	Proveedor

General Joins Integrity Oracle Notes Rules Preview

Constraint name: FK_PRODUCTO_SUMINISTR_PROVEEDO

Implementation: Trigger Cardinality: 1..*

Update constraint

☐ <None>
☐ Restrict
☒ Cascade
☐ Set Null
☐ Set Default

Delete constraint

☐ <None>
☒ Restrict
☐ Cascade
☐ Set Null
☐ Set Default

☒ Mandatory paren ☐ Check on commi

☒ Change parent allowec

More >> Aceptar Cancelar Aplicar Ayuda

FK hacia CATEGORIA

Constraint Name: FK_PRODUCTO_CATEGORIA

Delete Rule: RESTRICT

Update Rule: CASCADE



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

Reference Properties - Pertenece (PERTENECE)

Child Table: Producto Parent Table: Categoria

Constraint name: FK_PRODUCTO_PERTENECE_CATEGORI

Implementation: Trigger Cardinality: 1..*

Update constraint:

- ☐ <None>
- ☐ Restrict
- ☒ Cascade
- ☐ Set Null
- ☐ Set Default

Delete constraint:

- ☐ <None>
- ☒ Restrict
- ☐ Cascade
- ☐ Set Null
- ☐ Set Default

☒ Mandatory paren ☐ Check on commi

☒ Change parent allowec

More >> Aceptar Cancelar Aplicar Ayuda

C) Restricciones de Dominio (CHECK)

En tabla PRODUCTO, agregue las siguientes restricciones CHECK:

1. Precios positivos

sql

CONSTRAINT CHK_COSTO_COMPRA_POSITIVO

CHECK (costo_unitario_compra > 0)

CONSTRAINT CHK_COSTO_VENTA_POSITIVO

CHECK (costo_unitario_venta > 0)



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

2. *Stock no negativo*

sql

CONSTRAINT CHK_STOCK_NO_NEGATIVO

CHECK (stock_disponible >= 0)

3. *Margen de ganancia*

sql

CONSTRAINT CHK_MARGEN_GANANCIA

CHECK (costo_unitario_venta > costo_unitario_compra)

	Name	Code
1	Precios positivos	PRECIOS_POSITIVOS
2	Stock no negativo	STOCK_NO_NEGATIVO
3	margen de ganancia	MARGEN_DE_GANANCIA

PARTE 4: Generación del Script DDL

Procedimiento

1. En el PDM, seleccione: Database → Generate Database
2. Configure las opciones:

Pestaña General

☒ Check Model
Selection: Select All
Directory: Elija carpeta de destino
File name: Proveedores_Productos_DDL.sql

Pestaña Options

☒ Create tables
☒ Primary keys
☒ Foreign keys
☒ Indexes



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

- ☒ Checks
- ☒ Triggers (si los hay)

- ☒ Generate header comment
- ☒ Generate default values

Pestaña Format

- ☒ Generate SQL only
- Case: UPPER CASE (recomendado para Oracle)
- 3. Click en Preview para revisar el script
- 4. Click en OK para generar

Archivo generado:

```
/*=====*/
/* DBMS name:      ORACLE Version 11g          */
/* Created on:     23/10/2025 3:23:00          */
/*=====*/

-- TYPE PACKAGE DECLARATION
CREATE OR REPLACE PACKAGE PDYPES
AS
    TYPE REF_CURSOR IS REF CURSOR;
END;

-- INTEGRITY PACKAGE DECLARATION
CREATE OR REPLACE PACKAGE INTEGRITYPACKAGE AS
PROCEDURE INITNESTLEVEL;
FUNCTION GETNESTLEVEL RETURN NUMBER;
PROCEDURE NEXTNESTLEVEL;
PROCEDURE PREVIOUSNESTLEVEL;
END INTEGRITYPACKAGE;
```

PARTE 5: Validación y Pruebas

Creación de usuario en base de datos Oracle

Ingresar con usuario system

Create user **bchancusig** identified by **bchancusig**

Default tablespace users

Temporary tablespace temp

Profile default;

Grant connect,resource to **bchancusig**;

Alter user **bchancusig** quota unlimited on users;

A) Ejecución del Script

1. Conecte a Oracle 19c usando SQL Developer o SQL*Plus
2. Ejecute el script DDL completo



**ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL**

3. Verifique la creación exitosa:

```
-- Verificar tablas creadas
SELECT table_name FROM user_tables ORDER BY table_name;

-- Verificar constraints
SELECT constraint_name, constraint_type, table_name, search_condition FROM
user_constraints
WHERE table_name IN ('PROVEEDOR', 'CATEGORIA', 'PRODUCTO')
ORDER BY table_name, constraint_type;

-- Verificar índices
SELECT index_name, table_name, column_name, column_position FROM user_ind_columns
WHERE table_name IN ('PROVEEDOR', 'CATEGORIA', 'PRODUCTO')
ORDER BY table_name, index_name, column_position;
```

Resultado:

```
SQL> -- Verificar tablas creadas
SQL> SELECT table_name FROM user_tables ORDER BY table_name;
```

TABLE_NAME

AQ\$_INTERNET_AGENT_PRIVS
AQ\$_INTERNET_AGENTS
AQ\$_KEY_SHARD_MAP
AQ\$_QUEUES
AQ\$_QUEUE_TABLES
AQ\$_SCHEDULES
CATEGORIA
HELP
LOGMNR_AGE_SPILL\$
LOGMNR_ATTRCOL\$
LOGMNR_ATTRIBUTE\$

TABLE_NAME

LOGMNR_CCOL\$
LOGMNR_CONCOL_GG
LOGMNR_CON_GG
LOGMNR_DBNAME_UID_MAP
LOGMNR_CDEF\$
LOGMNR_GSBA
LOGMNR_GSII
LOGMNR_GTCS
LOGMNR_GTLO
LOGMNR_INDCOL_GG



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

LOGMNRC_IND_GG

TABLE_NAME

LOGMNR_COL\$
LOGMNR_COLTYPE\$
LOGMNR_CON\$
LOGMNR_CONTAINER\$
LOGMNRC_SEQ_GG
LOGMNRC_SHARD_TS
LOGMNRC_TS
LOGMNRC_TSPART
LOGMNR_DICTIONARY\$
LOGMNR_DICTSTATE\$
LOGMNR_DID\$

TABLE_NAME

LOGMNR_ENC\$
LOGMNR_ERROR\$
LOGMNR_FILTER\$
LOGMNRGGC_GTCS
LOGMNRGGC_GTLO
LOGMNR_GLOBAL\$
LOGMNR_GT_TAB_INCLUDE\$
LOGMNR_GT_USER_INCLUDE\$
LOGMNR_GT_XID_INCLUDE\$
LOGMNR_ICOL\$
LOGMNR_IDNSEQ\$

TABLE_NAME

LOGMNR_IND\$
LOGMNR_INDCOMPART\$
LOGMNR_INDPART\$
LOGMNR_INDSUBPART\$
LOGMNR_KOPM\$
LOGMNR_LOB\$
LOGMNR_LOBFRAG\$
LOGMNR_LOG\$
LOGMNR_LOGMNR_BUILDLOG
LOGMNR_NTAB\$
LOGMNR_OBJ\$

TABLE_NAME

LOGMNR_OPQTYPE\$
LOGMNR_PARAMETER\$



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

LOGMNR_PARTOBJ\$
LOGMNRP_CTAS_PART_MAP
LOGMNR_PDB_INFO\$
LOGMNR_PROCESSED_LOG\$
LOGMNR_PROFILE_PLSQL_STAT\$
LOGMNR_PROFILE_TABLE_STAT\$
LOGMNR_PROPS\$
LOGMNR_REFCON\$
LOGMNR_RESTART_CKPT\$

TABLE_NAME

LOGMNR_RESTART_CKPT_TXINFO\$
LOGMNR_SEED\$
LOGMNR_SESSION\$
LOGMNR_SESSION_ACTIONS\$
LOGMNR_SESSION_EVOLVE\$
LOGMNR_SHARD_TS
LOGMNR_SPILL\$
LOGMNR_SUBCOLTYPE\$
LOGMNR_TAB\$
LOGMNR_TABCOMPART\$
LOGMNR_TABPART\$

TABLE_NAME

LOGMNR_TABSUBPART\$
LOGMNR_T_MDDL\$
LOGMNR_TS\$
LOGMNR_TYPE\$
LOGMNR_UID\$
LOGMNR_USER\$
LOGSTDBY\$APPLY_MILESTONE
LOGSTDBY\$APPLY_PROGRESS
LOGSTDBY\$EDS_TABLES
LOGSTDBY\$EVENTS
LOGSTDBY\$FLASHBACK_SCN

TABLE_NAME

LOGSTDBY\$HISTORY
LOGSTDBY\$PARAMETERS
LOGSTDBY\$PLSQL
LOGSTDBY\$SCN
LOGSTDBY\$SKIP
LOGSTDBY\$SKIP_SUPPORT
LOGSTDBY\$SKIP_TRANSACTION
MVIEW\$ADV_AJG



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

MVIEW\$_ADV_Basetable
MVIEW\$_ADV_Clique
MVIEW\$_ADV_Eligible

TABLE_NAME

MVIEW\$_ADV_Exceptions
MVIEW\$_ADV_Filter
MVIEW\$_ADV_FilterInstance
MVIEW\$_ADV_FJG
MVIEW\$_ADV_GC
MVIEW\$_ADV_Info
MVIEW\$_ADV_Journal
MVIEW\$_ADV_Level
MVIEW\$_ADV_Log
MVIEW\$_ADV_Output
MVIEW\$_ADV_Parameters

TABLE_NAME

MVIEW\$_ADV_Plan
MVIEW\$_ADV_Pretty
MVIEW\$_ADV_Rollup
MVIEW\$_ADV_SQLDepend
MVIEW\$_ADV_Temp
MVIEW\$_ADV_Workload
OL\$
OL\$Hints
OL\$Nodes
PRODUCTO
PROVEEDOR

TABLE_NAME

REDO_DB
REDO_LOG
ROLLING\$Connections
ROLLING\$Databases
ROLLING\$Directives
ROLLING\$Events
ROLLING\$Parameters
ROLLING\$Plan
ROLLING\$Statistics
ROLLING\$Status
SCHEDULER_JOB_ARGS_TBL

TABLE_NAME



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

SCHEDULER_PROGRAM_ARGS_TBL
SQLPLUS_PRODUCT_PROFILE

134 filas seleccionadas.

SQL>

SQL> -- Verificar constraints

SQL> SELECT constraint_name, constraint_type, table_name, search_condition FROM
user_constraints

2 WHERE table_name IN ('PROVEEDOR', 'CATEGORIA', 'PRODUCTO')

3 ORDER BY table_name, constraint_type;

CONSTRAINT_NAME

C

-

TABLE_NAME

SEARCH_CONDITION

SYS_C007457

C

CATEGORIA

"CODIGO_CATEGORIA" IS NOT NULL

CONSTRAINT_NAME

C

-

TABLE_NAME

SEARCH_CONDITION

PK_CATEGORIA

P

CATEGORIA

CONSTRAINT_NAME

C

-

TABLE_NAME

SEARCH_CONDITION



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

SYS_C007459

C

PRODUCTO

"CODIGO_PRODUCTO" IS NOT NULL

CONSTRAINT_NAME

C

-

TABLE_NAME

SEARCH_CONDITION

SYS_C007460

C

PRODUCTO

"COD_PROVEEDOR" IS NOT NULL

CONSTRAINT_NAME

C

-

TABLE_NAME

SEARCH_CONDITION

SYS_C007461

C

PRODUCTO

"CODIGO_CATEGORIA" IS NOT NULL

CONSTRAINT_NAME

C

-

TABLE_NAME

SEARCH_CONDITION

PK_PRODUCTO

P

PRODUCTO



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

CONSTRAINT_NAME

C
-
TABLE_NAME

SEARCH_CONDITION

FK_PRODUCTO_PERTENECE_CATEGORI
R
PRODUCTO
CONSTRAINT_NAME

C
-
TABLE_NAME

SEARCH_CONDITION

SYS_C007463
C
PROVEEDOR
"COD_PROVEEDOR" IS NOT NULL
CONSTRAINT_NAME

C
-
TABLE_NAME

SEARCH_CONDITION

PK_PROVEEDOR
P
PROVEEDOR
9 filas seleccionadas.
SQL>
SQL> -- Verificar índices
SQL> SELECT index_name, table_name, column_name, column_position FROM
user_ind_columns



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

2 WHERE table_name IN ('PROVEEDOR', 'CATEGORIA', 'PRODUCTO')

3 ORDER BY table_name, index_name, column_position;

INDEX_NAME

TABLE_NAME

COLUMN_NAME

COLUMN_POSITION

PK_CATEGORIA

CATEGORIA

CODIGO_CATEGORIA

1

INDEX_NAME

TABLE_NAME

COLUMN_NAME

COLUMN_POSITION

PERTENECE_FK

PRODUCTO

CODIGO_CATEGORIA

1

INDEX_NAME

TABLE_NAME

COLUMN_NAME

COLUMN_POSITION

PK_PRODUCTO

PRODUCTO

CODIGO_PRODUCTO

1

INDEX_NAME

TABLE_NAME



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

COLUMN_NAME

COLUMN_POSITION

SUMINISTRA_FK

PRODUCTO

COD_PROVEEDOR

1

INDEX_NAME

TABLE_NAME

COLUMN_NAME

COLUMN_POSITION

PK_PROVEEDOR

PROVEEDOR

COD_PROVEEDOR

1

B) Pruebas de Integridad

Debe ejecutarse correctamente

INSERT INTO PROVEEDOR VALUES ('PR001', 'Distribuidora Tech SA', 'Av. Amazonas 123, Quito', '022345678', '0998765432', 'Juan', 'Pérez García');

INSERT INTO PROVEEDOR VALUES ('PR002', 'Comercial ElectroHogar', 'Calle García Moreno 456', '022876543', '0991234567', 'María', 'López Rodríguez');

INSERT INTO PROVEEDOR VALUES ('PR003', 'Importadora Global', 'Av. Naciones Unidas 789', '022567890', '0989876543', 'Carlos', 'Martínez Silva');

```
SQL> INSERT INTO PROVEEDOR VALUES ('PR001', 'Distribuidora Tech SA', 'Av. Amazonas 123, Quito', '022345678', '0998765432', 'Juan', 'Pérez García');
```

```
1 fila creada.
```

```
SQL> INSERT INTO PROVEEDOR VALUES ('PR002', 'Comercial ElectroHogar', 'Calle García Moreno 456', '022876543', '0991234567', 'María', 'López Rodríguez');
```

```
1 fila creada.
```

```
SQL> INSERT INTO PROVEEDOR VALUES ('PR003', 'Importadora Global', 'Av. Naciones Unidas 789', '022567890', '0989876543', 'Carlos', 'Martínez Silva');
```

```
1 fila creada.
```

INSERT INTO CATEGORIA VALUES ('CAT001', 'Electrónica');

INSERT INTO CATEGORIA VALUES ('CAT002', 'Electrodomésticos');

INSERT INTO CATEGORIA VALUES ('CAT003', 'Computación');



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

```
SQL> INSERT INTO CATEGORIA VALUES ('CAT001', 'Electrónica');
1 fila creada.

SQL> INSERT INTO CATEGORIA VALUES ('CAT002', 'Electrodomésticos');
1 fila creada.

SQL> INSERT INTO CATEGORIA VALUES ('CAT003', 'Computación');
1 fila creada.
```

```
INSERT INTO PRODUCTO VALUES ('PROD001', 'Laptop HP 15', 'Laptop HP 15.6"', 'Intel i5, 8GB RAM, 256GB SSD', 650.00, 899.00, 15, 'PR001', 'CAT001');
INSERT INTO PRODUCTO VALUES ('PROD002', 'Mouse Logitech M185', 'Mouse inalámbrico USB', 8.50, 15.99, 50, 'PR001', 'CAT003');
INSERT INTO PRODUCTO VALUES ('PROD003', 'Refrigeradora LG 500L', 'Refrigeradora No Frost 500 litros', 850.00, 1299.00, 8, 'PR002', 'CAT002');
COMMIT;
```

```
SQL>
SQL> INSERT INTO PRODUCTO (CODIGO_PRODUCTO, COD_PROVEEDOR, CODIGO_CATEGORIA, NOMBRE_PROD, DESCRIPCION, COSTOUC, COSTOUV, STOCK)
  2 VALUES ('PROD001', 'PR001', 'CAT001', 'Laptop HP 15', 'Laptop HP 15.6"', 'Intel i5, 8GB RAM, 256GB SSD', 650.00, 899.00, 15);
1 fila creada.

SQL>
SQL> INSERT INTO PRODUCTO (CODIGO_PRODUCTO, COD_PROVEEDOR, CODIGO_CATEGORIA, NOMBRE_PROD, DESCRIPCION, COSTOUC, COSTOUV, STOCK)
  2 VALUES ('PROD002', 'PR001', 'CAT003', 'Mouse Logitech M185', 'Mouse inalámbrico USB', 8.50, 15.99, 50);
1 fila creada.

SQL>
SQL> INSERT INTO PRODUCTO (CODIGO_PRODUCTO, COD_PROVEEDOR, CODIGO_CATEGORIA, NOMBRE_PROD, DESCRIPCION, COSTOUC, COSTOUV, STOCK)
  2 VALUES ('PROD003', 'PR002', 'CAT002', 'Refrigeradora LG 500L', 'Refrigeradora No Frost 500 litros', 850.00, 1299.00, 8);
1 fila creada.

SQL>
SQL> COMMIT;

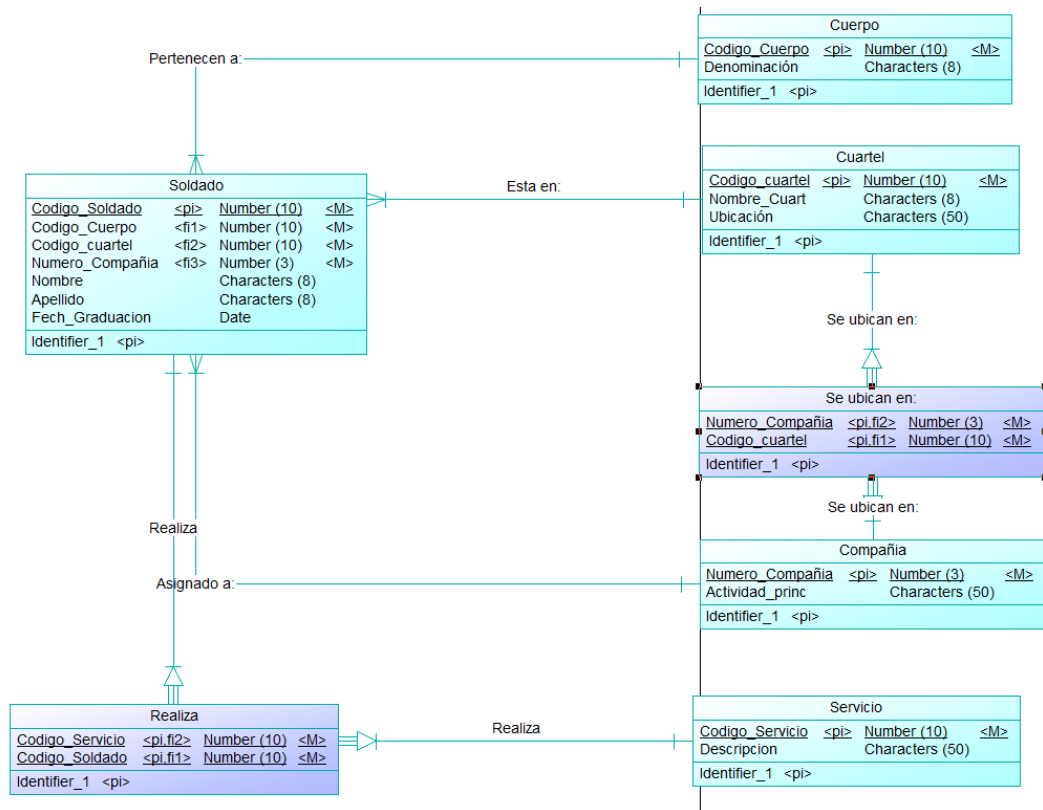
Confirmación terminada.
```

Ejercicio 2: Sistema para milicia

Partimos de:



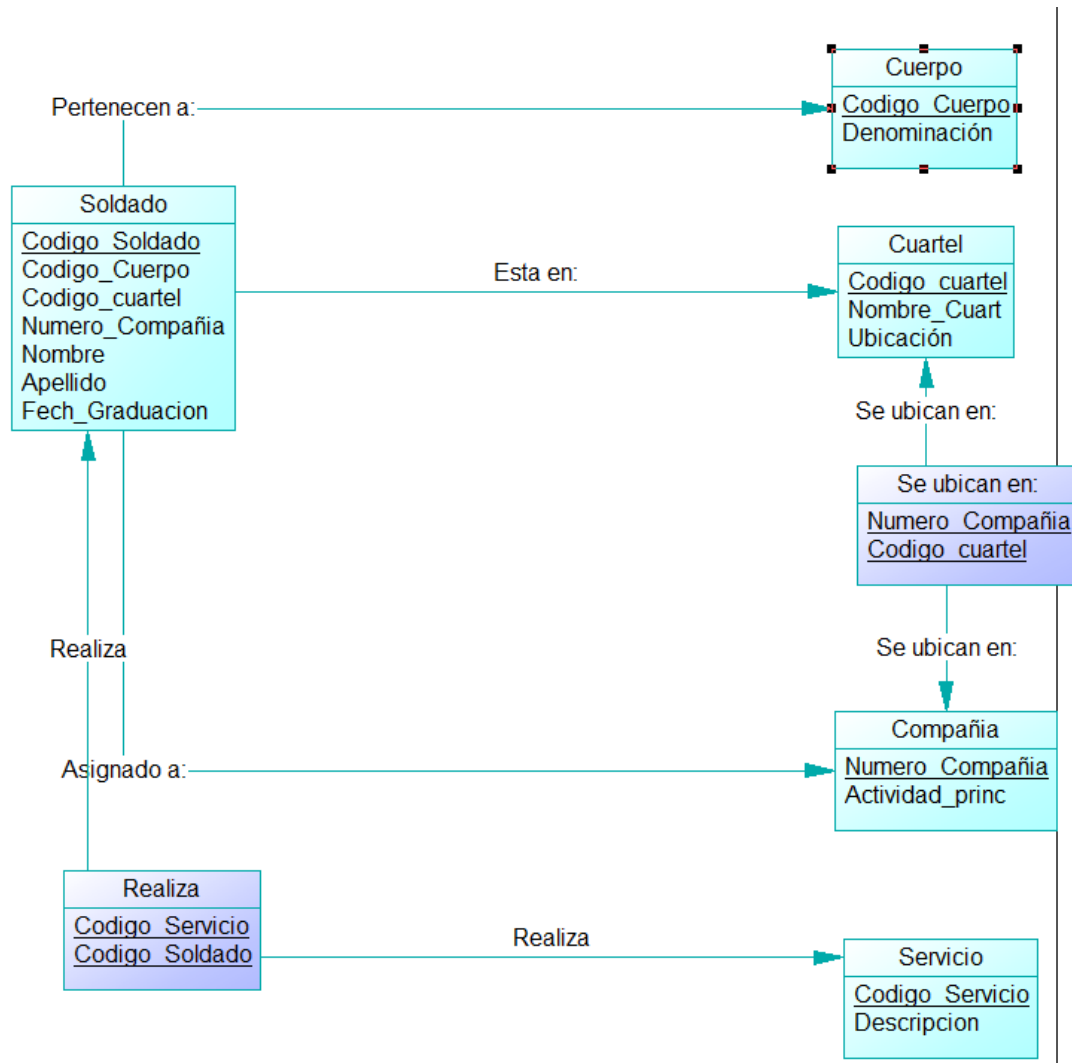
ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL



Su PDM:



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL



Tablas:

Cuerpo:

	Name	Code	Data Type	Lengt	Precision	P	F	N
1	Codigo_Cuerpo	CODIGO_CUERPO	VARCHAR2(10)	10		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	Denominación	DENOMINACION	VARCHAR2(8)	8		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Cuartel:

	Name	Code	Data Type	Lengt	Precision	P	F	N
1	Codigo_cuartel	CODIGO_CUARTEL	VARCHAR2(10)	10		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	Nombre_Cuart	NOMBRE_CUART	VARCHAR2(150)	150		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
→	Ubicación	UBICACION	VARCHAR2(200)	200		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Compañía:

	Name	Code	Data Type	Lengt	Precision	P	F	N
1	Numero_Compañía	NUMERO_COMPANIA	VARCHAR2(10)	10		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
→	Actividad_princ	ACTIVIDAD_PRINC	VARCHAR2(200)	200		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



**ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL**

Servicio:

	Name	Code	Data Type	Length	Precision	P	F	N
1	Codigo_Servicio	CODIGO_SERVICIO	VARCHAR2(10)	10		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
→	Descripcion	DESCRIPCION	VARCHAR2(200)	200		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Soldado:

	Name	Code	Data Type	Length
1	Codigo_Soldado	CODIGO_SOLDADO	VARCHAR2(10)	10
2	Codigo_Cuerpo	CODIGO_CUERPO	VARCHAR2(10)	10
3	Codigo_cuartel	CODIGO_CUARTEL	VARCHAR2(10)	10
4	Numero_Compañi	NUMERO_COMPANIA	VARCHAR2(10)	10
5	Nombre	NOMBRE	VARCHAR2(100)	100
6	Apellido	APELLIDO	VARCHAR2(100)	100
→	Fech_Graduacion	FECH_GRADUACION	VARCHAR2(50)	50

Restricciones:

- Soldado, se asegura que solo se ingresen rangos válidos predefinidos en la columna**
 CONSTRAINT CHK_SOLDADO_GRADUACION
 CHECK (graduacion IN ('RASO', 'CABO SEGUNDO', 'CABO PRIMERO', 'SARGENTO SEGUNDO', 'SARGENTO PRIMERO'))

	Name	Code
→	Confirmar_Graduacion	CONFIRMAR_GRADUACION

- Cuerpo, limita las denominaciones de los cuerpos a una lista específica, similar a los ejemplos provistos.**
 CONSTRAINT CHK_CUERPO_DENOMINACION
 CHECK (denominacion IN ('INFANTERIA', 'ARTILLERIA', 'ARMADA', 'INGENIERIA', 'LOGISTICA', 'COMUNICACIONES'))

	Name	Code
1	Comprobar_Denominacion	COMPROBAR_DENOMINACION

- Soldado_Servicio (Realiza), impide que se registren servicios con fechas futuras ,
 asumiendo que el registro es de servicios ya completados**
 CONSTRAINT CHK_SERVICIO_FECHA_NO_FUTURA
 CHECK (fecha_realizacion <= SYSDATE)

	Name	Code
→	Bloquear_Servicios_Sin_Complet	BLOQUEAR_SERVICIOS_SIN_

Archivo con scripts:



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

```
Milicia_DDL.sql × +
Archivo Editar Ver

/*=====*/
/* DBMS name:      ORACLE Version 11g          */
/* Created on:     23/10/2025 15:17:25         */
/*=====*/

ALTER TABLE REALIZA
  DROP CONSTRAINT FK_REALIZA_REALIZA_SERVICIO;
```

Validación:

Tablas Creadas:

```
SQL> SELECT table_name FROM user_tables ORDER BY table_name;
```

TABLE_NAME

AQ\$_INTERNET_AGENT_PRIVS
AQ\$_INTERNET_AGENTS
AQ\$_KEY_SHARD_MAP
AQ\$_QUEUES
AQ\$_QUEUE_TABLES
AQ\$_SCHEDULES
CATEGORIA
HELP
LOGMNR_AGE_SPILL\$
LOGMNR_ATTRCOL\$
LOGMNR_ATTRIBUTES\$

TABLE_NAME

LOGMNR_CCOL\$
LOGMNR_CONCOL_GG
LOGMNR_CON_GG
LOGMNR_DBNAME_UID_MAP
LOGMNR_CDEF\$
LOGMNR_GSBA
LOGMNR_GSII
LOGMNR_GTCS
LOGMNR_GTLO
LOGMNR_INDCOL_GG
LOGMNR_IND_GG

TABLE_NAME



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

LOGMNR_COL\$
LOGMNR_COLTYPE\$
LOGMNR_CON\$
LOGMNR_CONTAINER\$
LOGMNR_SEQ_GG
LOGMNR_SHARD_TS
LOGMNR_TS
LOGMNR_TSPART
LOGMNR_DICTIONARY\$
LOGMNR_DICTSTATE\$
LOGMNR_DID\$

TABLE_NAME

LOGMNR_ENC\$
LOGMNR_ERROR\$
LOGMNR_FILTER\$
LOGMNRGGC_GTCS
LOGMNRGGC_GTLO
LOGMNR_GLOBAL\$
LOGMNR_GT_TAB_INCLUDE\$
LOGMNR_GT_USER_INCLUDE\$
LOGMNR_GT_XID_INCLUDE\$
LOGMNR_ICOL\$
LOGMNR_IDNSEQ\$

TABLE_NAME

LOGMNR_IND\$
LOGMNR_INDCOMPART\$
LOGMNR_INDPART\$
LOGMNR_INDSUBPART\$
LOGMNR_KOPM\$
LOGMNR_LOB\$
LOGMNR_LOBFRAG\$
LOGMNR_LOG\$
LOGMNR_LOGMNR_BUILDLOG
LOGMNR_NTAB\$
LOGMNR_OBJ\$

TABLE_NAME

LOGMNR_OPQTYPE\$
LOGMNR_PARAMETER\$
LOGMNR_PARTOBJ\$
LOGMNR_CTAS_PART_MAP
LOGMNR_PDB_INFO\$
LOGMNR_PROCESSED_LOG\$



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

LOGMNR_PROFILE_PLSQL_STAT\$
LOGMNR_PROFILE_TABLE_STAT\$
LOGMNR_PROPS\$
LOGMNR_REFCON\$
LOGMNR_RESTART_CKPT\$

TABLE_NAME

LOGMNR_RESTART_CKPT_TXINFO\$
LOGMNR_SEED\$
LOGMNR_SESSION\$
LOGMNR_SESSION_ACTIONS\$
LOGMNR_SESSION_EVOLVE\$
LOGMNR_SHARD_TS
LOGMNR_SPILL\$
LOGMNR_SUBCOLTYPE\$
LOGMNR_TAB\$
LOGMNR_TABCOMPART\$
LOGMNR_TABPART\$

TABLE_NAME

LOGMNR_TABSUBPART\$
LOGMNR_MDDL\$
LOGMNR_TS\$
LOGMNR_TYPE\$
LOGMNR_UID\$
LOGMNR_USER\$
LOGSTDBY\$APPLY_MILESTONE
LOGSTDBY\$APPLY_PROGRESS
LOGSTDBY\$EDS_TABLES
LOGSTDBY\$EVENTS
LOGSTDBY\$FLASHBACK_SCN

TABLE_NAME

LOGSTDBY\$HISTORY
LOGSTDBY\$PARAMETERS
LOGSTDBY\$PLSQL
LOGSTDBY\$SCN
LOGSTDBY\$SKIP
LOGSTDBY\$SKIP_SUPPORT
LOGSTDBY\$SKIP_TRANSACTION
MVIEW\$ADV_AJG
MVIEW\$ADV_Basetable
MVIEW\$ADV_Clique
MVIEW\$ADV_Eligible



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

TABLE_NAME
MVIEW\$_ADV_EXCEPTIONS
MVIEW\$_ADV_FILTER
MVIEW\$_ADV_FILTERINSTANCE
MVIEW\$_ADV_FJG
MVIEW\$_ADV_GC
MVIEW\$_ADV_INFO
MVIEW\$_ADV_JOURNAL
MVIEW\$_ADV_LEVEL
MVIEW\$_ADV_LOG
MVIEW\$_ADV_OUTPUT
MVIEW\$_ADV_PARAMETERS
TABLE_NAME
MVIEW\$_ADV_PLAN
MVIEW\$_ADV_PRETTY
MVIEW\$_ADV_ROLLUP
MVIEW\$_ADV_SQLDEPEND
MVIEW\$_ADV_TEMP
MVIEW\$_ADV_WORKLOAD
OL\$
OL\$HINTS
OL\$NODES
PRODUCTO
PROVEEDOR
TABLE_NAME
REDO_DB
REDO_LOG
ROLLING\$CONNECTIONS
ROLLING\$DATABASES
ROLLING\$DIRECTIVES
ROLLING\$EVENTS
ROLLING\$PARAMETERS
ROLLING\$PLAN
ROLLING\$STATISTICS
ROLLING\$STATUS
SCHEDULER_JOB_ARGS_TBL
TABLE_NAME
SCHEDULER_PROGRAM_ARGS_TBL
SQLPLUS_PRODUCT_PROFILE
134 filas seleccionadas.



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

Verificar Constraints:

```
SQL> SELECT constraint_name, constraint_type, table_name, search_condition
2 FROM user_constraints
3 WHERE table_name IN ('PROVEEDOR', 'CATEGORIA', 'PRODUCTO')
4 ORDER BY table_name, constraint_type;
```

CONSTRAINT_NAME

C

-

TABLE_NAME

SEARCH_CONDITION

SYS_C007457

C

CATEGORIA

"CODIGO_CATEGORIA" IS NOT NULL

CONSTRAINT_NAME

C

-

TABLE_NAME

SEARCH_CONDITION

PK_CATEGORIA

P

CATEGORIA

CONSTRAINT_NAME

C

-

TABLE_NAME

SEARCH_CONDITION

SYS_C007460

C

PRODUCTO

"COD_PROVEEDOR" IS NOT NULL



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

CONSTRAINT_NAME

C

-

TABLE_NAME

SEARCH_CONDITION

SYS_C007459

C

PRODUCTO

"CODIGO_PRODUCTO" IS NOT NULL

CONSTRAINT_NAME

C

-

TABLE_NAME

SEARCH_CONDITION

SYS_C007461

C

PRODUCTO

"CODIGO_CATEGORIA" IS NOT NULL

CONSTRAINT_NAME

C

-

TABLE_NAME

SEARCH_CONDITION

PK_PRODUCTO

P

PRODUCTO

CONSTRAINT_NAME

C

-

TABLE_NAME



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

SEARCH_CONDITION
FK_PRODUCTO_PERTENECE_CATEGORI R PRODUCTO
CONSTRAINT_NAME
C -
TABLE_NAME
SEARCH_CONDITION
SYS_C007463 C PROVEEDOR "COD_PROVEEDOR" IS NOT NULL
CONSTRAINT_NAME
C -
TABLE_NAME
SEARCH_CONDITION
PK_PROVEEDOR P PROVEEDOR
9 filas seleccionadas.

Verificar índices:

SQL> SELECT index_name, table_name, column_name, column_position 2 FROM user_ind_columns 3 WHERE table_name IN ('PROVEEDOR', 'CATEGORIA', 'PRODUCTO') 4 ORDER BY table_name, index_name, column_position;
INDEX_NAME



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

TABLE_NAME

COLUMN_NAME

COLUMN_POSITION

PK_CATEGORIA
CATEGORIA
CODIGO_CATEGORIA
1
INDEX_NAME

TABLE_NAME

COLUMN_NAME

COLUMN_POSITION

PERTENECE_FK
PRODUCTO
CODIGO_CATEGORIA
1
INDEX_NAME

TABLE_NAME

COLUMN_NAME

COLUMN_POSITION

PK_PRODUCTO
PRODUCTO
CODIGO_PRODUCTO
1
INDEX_NAME

TABLE_NAME

COLUMN_NAME

COLUMN_POSITION



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

SUMINISTRA_FK
PRODUCTO
COD_PROVEEDOR
1

INDEX_NAME

TABLE_NAME

COLUMN_NAME

COLUMN_POSITION

PK_PROVEEDOR
PROVEEDOR
COD_PROVEEDOR
1

Pruebas de integridad finales:

SQL> -- Insertar datos en las tablas

SQL> INSERT INTO CUERPO (CODIGO_CUERPO, DENOMINACION)
2 VALUES ('CUE001', 'INFANT');

1 fila creada.

SQL>

SQL> INSERT INTO CUERPO (CODIGO_CUERPO, DENOMINACION)
2 VALUES ('CUE002', 'ARTILL');

1 fila creada.

SQL>

SQL> INSERT INTO CUARTEL (CODIGO_CUARTEL, NOMBRE_CUART, UBICACION)
2 VALUES ('QUA001', 'Cuartel San Jorge', 'Quito');

1 fila creada.

SQL>

SQL> INSERT INTO CUARTEL (CODIGO_CUARTEL, NOMBRE_CUART, UBICACION)
2 VALUES ('QUA002', 'Cuartel Rumiñahui', 'Loja');

1 fila creada.

SQL>

SQL> INSERT INTO COMPANIA (NUMERO_COMPANIA, ACTIVIDAD_PRINC)
2 VALUES ('CIA101', 'Entrenamiento Táctico');



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

1 fila creada.

SQL>

```
SQL> INSERT INTO COMPANIA (NUMERO_COMPANIA, ACTIVIDAD_PRINC)
2 VALUES ('CIA102', 'Logística y Abastecimiento');
```

1 fila creada.

SQL>

```
SQL> INSERT INTO SERVICIO (CODIGO_SERVICIO, DESCRIPCION)
2 VALUES ('SER001', 'Guardia Nocturna 22:00-06:00');
```

1 fila creada.

SQL>

```
SQL> INSERT INTO SERVICIO (CODIGO_SERVICIO, DESCRIPCION)
2 VALUES ('SER002', 'Instructor de Campo');
```

1 fila creada.

SQL>

SQL> -- Insertar datos en la tabla SOLDADO (que depende de las anteriores)

SQL> -- Nota: Tu DDL usa FECH_GRADUACION VARCHAR2(50), usaré ese campo para el rango [cite: 141]

```
SQL> INSERT INTO SOLDADO (CODIGO_SOLDADO, CODIGO_CUERPO, CODIGO_CUARTEL,
NUMERO_COMPANIA, NOMBRE, APELLIDO, FECH_GRADUACION)
2 VALUES ('S00001', 'CUE001', 'QUA001', 'CIA101', 'Juan', 'Perez', 'RASO');
```

1 fila creada.

SQL>

```
SQL> INSERT INTO SOLDADO (CODIGO_SOLDADO, CODIGO_CUERPO, CODIGO_CUARTEL,
NUMERO_COMPANIA, NOMBRE, APELLIDO, FECH_GRADUACION)
2 VALUES ('S00002', 'CUE002', 'QUA002', 'CIA102', 'Ana', 'Gomez', 'CABO');
```

1 fila creada.

SQL>

SQL> -- Insertar en tablas asociativas (Muchos a Muchos)

```
SQL> INSERT INTO REALIZA (CODIGO_SERVICIO, CODIGO_SOLDADO)
2 VALUES ('SER001', 'S00001');
```

1 fila creada.

SQL>

```
SQL> INSERT INTO REALIZA (CODIGO_SERVICIO, CODIGO_SOLDADO)
2 VALUES ('SER002', 'S00002');
```



**ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL**

1 fila creada.

SQL>

SQL> INSERT INTO REALIZA (CODIGO_SERVICIO, CODIGO_SOLDADO)

2 VALUES ('SER001', 'S00002'); -- La soldado Ana Gomez también hace guardia

3

SQL> INSERT INTO SE_UBICAN_EN_ (NUMERO_COMPANIA, CODIGO_CUARTEL)

2 VALUES ('CIA101', 'QUA001');

1 fila creada.

SQL>

SQL> INSERT INTO SE_UBICAN_EN_ (NUMERO_COMPANIA, CODIGO_CUARTEL)

2 VALUES ('CIA102', 'QUA002');

1 fila creada.

SQL>

SQL> -- Confirmar los cambios

SQL> COMMIT;

Confirmación terminada.

SQL>

Análisis de resultados:

Se ha logrado generar el modelo físico en el motor Oracle 19c, haciendo uso de power designer, gracias a las pruebas hechas en los puntos 4 y 5, se ha demostrado que es funcional y coherente con el diseño lógico original, se ha validado correctamente la creación de tablas, restricciones de integridad (claves primarias, foráneas y verificaciones) e índices. Las pruebas de inserción confirmaron que los datos se ajustan a las reglas. Garantizando la consistencia y validez de la información almacenada en el sistema

Conclusiones y recomendaciones:

PowerDesigner optimiza la transición del modelo lógico al físico, ya que gracias al uso de Power Designer y Oracle se hizo la transición exitosamente garantizando la integridad de los datos a través de restricciones y claves definidas, y mejorando el rendimiento de las consultas mediante la creación automática de índices. Podemos concluir que después de todas las pruebas el modelo es funcional y escalable.

Bibliografía:

[1] Oracle Corporation, *Oracle Database SQL Language Reference, 19c*, Oracle Documentation, 2020. [Online]. Available: <https://docs.oracle.com/en/database/oracle/oracle-database/19/sqlrf/>

[2] SAP SE, *PowerDesigner Physical Data Model User Guide*, SAP Documentation, 2020. [Online]. Available: <https://help.sap.com/viewer/product/POWERDESIGNER>

[3] T. J. Teorey, S. S. Lightstone, and T. Nadeau, *Database Modeling and Design: Logical Design*, 5th ed. Burlington, MA: Morgan Kaufmann, 2011.