**Práctica 02**

**PLANTILLA DE RESPUESTAS**

**Nombres: Luis Morales**

Traducción de modelo lógico a DDL y creación de la base de datos

|  |
| --- |
| **Diseño físico para Oracle** |

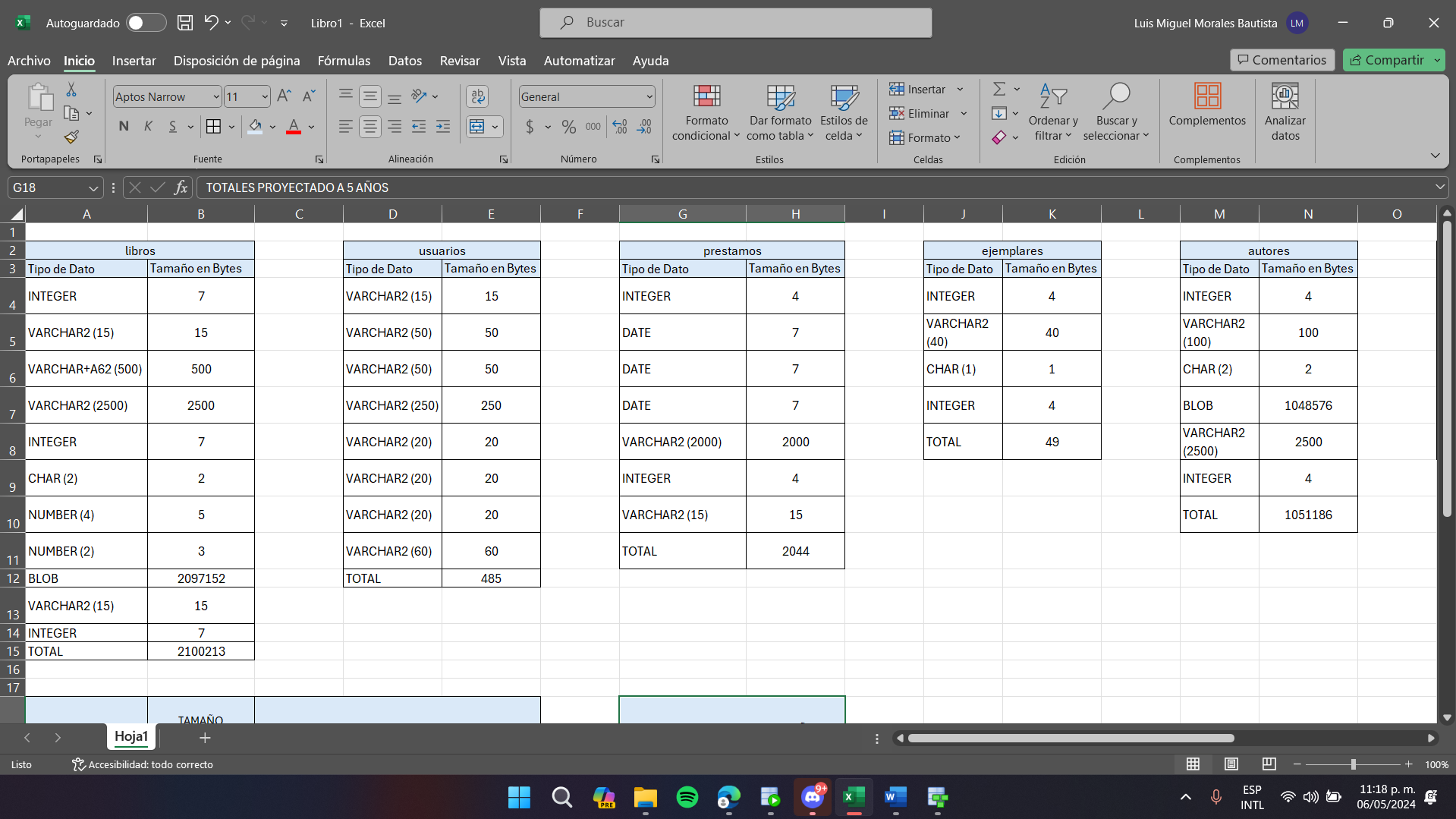
**Script DDL**

|  |
| --- |
| -- Generado por Oracle SQL Developer Data Modeler 23.1.0.087.0806  -- en: 2024-05-06 18:41:07 COT  -- sitio: Oracle Database 11g  -- tipo: Oracle Database 11g  -- predefined type, no DDL - MDSYS.SDO\_GEOMETRY  -- predefined type, no DDL - XMLTYPE  CREATE TABLE areas\_conocimientp (  codareas VARCHAR2(15) NOT NULL,  nombrearea VARCHAR2(250) NOT NULL,  codareapadre VARCHAR2(15)  );  ALTER TABLE areas\_conocimientp ADD CONSTRAINT areas\_conocimientp\_pk PRIMARY KEY ( codareas );  CREATE TABLE autores (  idautor INTEGER NOT NULL,  nombreautor VARCHAR2(100) NOT NULL,  pais CHAR(2) NOT NULL,  foto BLOB,  bio VARCHAR2(2500),  numlibros INTEGER NOT NULL  );  ALTER TABLE autores ADD CONSTRAINT autores\_pk PRIMARY KEY ( idautor );  CREATE TABLE autorias (  orden SMALLINT NOT NULL,  idautor INTEGER NOT NULL,  idlibros INTEGER NOT NULL  );  ALTER TABLE autorias ADD CONSTRAINT autorias\_pk PRIMARY KEY ( idautor,  idlibros );  CREATE TABLE editoriales (  idedt INTEGER NOT NULL,  nombreedt VARCHAR2(100) NOT NULL,  pais VARCHAR2(100) NOT NULL,  web VARCHAR2(200)  );  ALTER TABLE editoriales ADD CONSTRAINT editoriales\_pk PRIMARY KEY ( idedt );  CREATE TABLE ejemplares (  idejemplar INTEGER NOT NULL,  ubicacion VARCHAR2(40),  status CHAR(1) NOT NULL,  idlibro INTEGER NOT NULL  );  ALTER TABLE ejemplares ADD CONSTRAINT ejemplares\_pk PRIMARY KEY ( idejemplar );  CREATE TABLE libros (  idlibro INTEGER NOT NULL,  isbn VARCHAR2(15),  titulo VARCHAR2(500) NOT NULL,  sinopsis VARCHAR2(2500),  numpaginas INTEGER NOT NULL,  idioma CHAR(2) NOT NULL,  anio NUMBER(4) NOT NULL,  nroedicion NUMBER(2) NOT NULL,  imgportada BLOB,  codarea VARCHAR2(15) NOT NULL,  idedt INTEGER NOT NULL  );  ALTER TABLE libros ADD CONSTRAINT libros\_pk PRIMARY KEY ( idlibro );  ALTER TABLE libros ADD CONSTRAINT libros\_\_un UNIQUE ( isbn );  CREATE TABLE prestamos (  idprestamos INTEGER NOT NULL,  "fechaPrestamo " DATE NOT NULL,  fechavencimiento DATE NOT NULL,  fechadevolucion DATE,  observacion VARCHAR2(2000) NOT NULL,  idejemplar INTEGER NOT NULL,  cedula VARCHAR2(15) NOT NULL  );  ALTER TABLE prestamos ADD CONSTRAINT prestamos\_pk PRIMARY KEY ( idprestamos );  CREATE TABLE usuarios (  cedula VARCHAR2(15) NOT NULL,  apellidos VARCHAR2(50) NOT NULL,  nombres VARCHAR2(50) NOT NULL,  direccion VARCHAR2(250),  telefono1 VARCHAR2(20),  telefono2 VARCHAR2(20),  telefono3 VARCHAR2(20),  email VARCHAR2(60)  );  ALTER TABLE usuarios ADD CONSTRAINT usuarios\_pk PRIMARY KEY ( cedula );  ALTER TABLE usuarios ADD CONSTRAINT usuarios\_\_un UNIQUE ( email );  ALTER TABLE prestamos  ADD CONSTRAINT cedula FOREIGN KEY ( cedula )  REFERENCES usuarios ( cedula );  ALTER TABLE libros  ADD CONSTRAINT codarea FOREIGN KEY ( codarea )  REFERENCES areas\_conocimientp ( codareas );  ALTER TABLE areas\_conocimientp  ADD CONSTRAINT codareapadre FOREIGN KEY ( codareapadre )  REFERENCES areas\_conocimientp ( codareas );  ALTER TABLE autorias  ADD CONSTRAINT idautor FOREIGN KEY ( idautor )  REFERENCES autores ( idautor );  ALTER TABLE libros  ADD CONSTRAINT idedt FOREIGN KEY ( idedt )  REFERENCES editoriales ( idedt );  ALTER TABLE prestamos  ADD CONSTRAINT idejemplar FOREIGN KEY ( idejemplar )  REFERENCES ejemplares ( idejemplar );  ALTER TABLE autorias  ADD CONSTRAINT idlibros FOREIGN KEY ( idlibros )  REFERENCES libros ( idlibro );  ALTER TABLE ejemplares  ADD CONSTRAINT idlibrosv2 FOREIGN KEY ( idlibro )  REFERENCES libros ( idlibro );    ALTER TABLE prestamos ADD CONSTRAINT chk\_fechaDevolucion CHECK (fechadevolucion >= fechaprestamo);  ALTER TABLE libros ADD CONSTRAINT chk\_numPaginas CHECK (numpaginas > 50 AND numpaginas < 2000);  ALTER TABLE libros ADD CONSTRAINT chk\_nroEdicion CHECK (nroedicion > 0);  ALTER TABLE libros ADD CONSTRAINT chk\_anio CHECK (anio > 1800);  ALTER TABLE ejemplares ADD CONSTRAINT chk\_status CHECK (status IN ('D', 'P', 'B'));  ALTER TABLE autores ADD CONSTRAINT chk\_numLibros CHECK (numlibros >= 0);  ALTER TABLE usuarios ADD CONSTRAINT chk\_email CHECK (email LIKE '%\_\_@\_\_%' );  -- Informe de Resumen de Oracle SQL Developer Data Modeler:  --  -- CREATE TABLE 8  -- CREATE INDEX 0  -- ALTER TABLE 18  -- CREATE VIEW 0  -- ALTER VIEW 0  -- CREATE PACKAGE 0  -- CREATE PACKAGE BODY 0  -- CREATE PROCEDURE 0  -- CREATE FUNCTION 0  -- CREATE TRIGGER 0  -- ALTER TRIGGER 0  -- CREATE COLLECTION TYPE 0  -- CREATE STRUCTURED TYPE 0  -- CREATE STRUCTURED TYPE BODY 0  -- CREATE CLUSTER 0  -- CREATE CONTEXT 0  -- CREATE DATABASE 0  -- CREATE DIMENSION 0  -- CREATE DIRECTORY 0  -- CREATE DISK GROUP 0  -- CREATE ROLE 0  -- CREATE ROLLBACK SEGMENT 0  -- CREATE SEQUENCE 0  -- CREATE MATERIALIZED VIEW 0  -- CREATE MATERIALIZED VIEW LOG 0  -- CREATE SYNONYM 0  -- CREATE TABLESPACE 0  -- CREATE USER 0  --  -- DROP TABLESPACE 0  -- DROP DATABASE 0  --  -- REDACTION POLICY 0  --  -- ORDS DROP SCHEMA 0  -- ORDS ENABLE SCHEMA 0  -- ORDS ENABLE OBJECT 0  --  -- ERRORS 0  -- WARNINGS 0 |

**Restricciones o requerimientos que no se pueden implementar con DDL**

|  |  |
| --- | --- |
| **Restricción / Requerimiento** | **¿Cómo se debería controlar?** |
| Autores.numLibros, es un campo derivado, corresponde al número de libros registrados por cada  autor. Equivale a contar las filas asociadas a cada autor en la tabla autorias. Debe estar siempre  actualizado. | Esta restricción se debería implementar a nivel de código, el cual debe realizar el conteo de filas, y velicar el estado que siempre debe estar actualizado. |
| Cada libro debe tener al menos un autor. | Esta restricción se debería implementar a nivel de código, al momento de ingresar un libro al sistema también debe agregarse un autor. |
| En la tabla autorias, un mismo libro no puede tener dos o más autores con el mismo orden (el  orden es la posición en la que aparece el autor en la lista de autores del libro). | Esta restricción se debería implementar a nivel de código, verificando el momento en el que fue ingresado el autor para controlar el orden luego. |
| En una misma fecha a un mismo usuario no se le puede prestar dos ejemplares\* del mismo libro. | Esta restricción se debería implementar a nivel de código, verificando los id de los ejemplares y sus libros |

**Cálculo del espacio en disco**



Pantalla de computadora

Descripción generada automáticamente

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

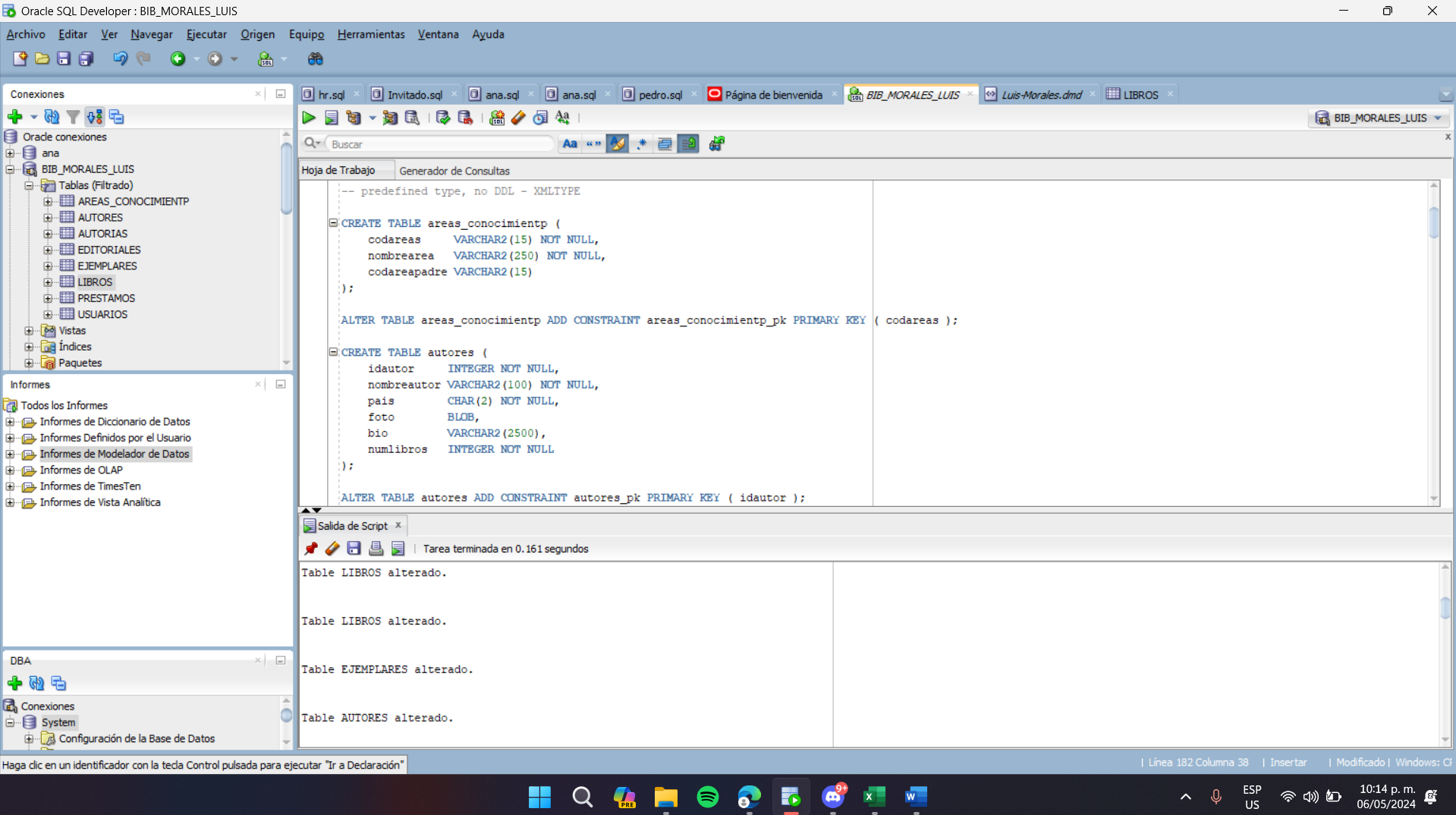
|  |
| --- |
| **Implementación de la base de datos en Oracle** |

Para validar la implementación de la base de datos realice las siguientes capturas de pantalla de la herramienta **SQL Developer**.

***Realice la captura de toda la pantalla de manera que se vea la fecha y hora en la que se realizó.***

**Captura(s) de pantalla 1**

(Que muestre el esquema de base de datos creado con la lista de tablas)



**Captura(s) de pantalla 2**

(Que muestre las columnas y constraints de la tabla LIBROS)

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

.-