

Práctica 02

Por: Oliver Roberto Saraguro Remache

Fecha: 06 de mayo, 2024.

PLANTILLA DE RESPUESTAS

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

Diseño físico para Oracle

Script DDL

```
-- Generated by Oracle SQL Developer Data Modeler 23.1.0.087.0806
--   at:      2024-05-06 18:46:58 ECT
--   site:    Oracle Database 11g
--   type:    Oracle Database 11g

-- predefined type, no DDL - MDSYS.SDO_GEOMETRY

-- predefined type, no DDL - XMLTYPE

CREATE TABLE areas_conocimiento (
    codarea      VARCHAR2(15) NOT NULL,
    nombrearea   VARCHAR2(250) NOT NULL,
    codareapadre VARCHAR2(15)
);

ALTER TABLE areas_conocimiento ADD CONSTRAINT areas_conocimiento_pk PRIMARY KEY ( codarea );

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 );

-- AUTORES NUMLIBROS NO PUEDER SER NEGATIVO.
ALTER TABLE autores ADD CONSTRAINT numlibros_ck CHECK (numlibros > 0);

CREATE TABLE autorias (
    idlibro      INTEGER NOT NULL,
    idautor      INTEGER NOT NULL,
    orden        SMALLINT NOT NULL
);

ALTER TABLE autorias ADD CONSTRAINT autorias_pk PRIMARY KEY ( idlibro,
                                                                idautor );

-- En la tabla autorias, un mismo libro no puede tener dos o más autores con el mismo orden.
ALTER TABLE autorias ADD CONSTRAINT orden_unico UNIQUE (idlibro, orden);

CREATE TABLE editoriales (
    idedit       INTEGER NOT NULL,
    nombreadit   VARCHAR2(100) NOT NULL,
    pais         VARCHAR2(100) NOT NULL,
    web          VARCHAR2(200)
);

ALTER TABLE editoriales ADD CONSTRAINT editoriales_pk PRIMARY KEY ( idedit );
```

```

CREATE TABLE ejemplares (
    idejemplar INTEGER NOT NULL,
    idlibro     INTEGER NOT NULL,
    ubicacion  VARCHAR2(40),
    status      CHAR(1) NOT NULL
);

ALTER TABLE ejemplares ADD CONSTRAINT ejemplares_pk PRIMARY KEY ( idejemplar );

-- EJEMPLARES ESTATUS TRES POSIBLES VALORES "D" DISPONIBLE, "P" PRESTADO, "B" DADO DE BAJA
ALTER TABLE ejemplares ADD CONSTRAINT status_ck CHECK (status in('D', 'P', 'B'));

CREATE TABLE libros (
    idlibro     INTEGER NOT NULL,
    isbm        VARCHAR2(15) NOT NULL,
    titulo      VARCHAR2(500) NOT NULL,
    sinopsis    VARCHAR2(2500),
    numpaginas  INTEGER NOT NULL,
    idioma      CHAR(2),
    codarea     VARCHAR2(15) NOT NULL,
    anio        NUMBER(4) NOT NULL,
    nroedicion  NUMBER(4) NOT NULL,
    idedit      INTEGER NOT NULL,
    imgpotada   BLOB
);

ALTER TABLE libros ADD CONSTRAINT libros_pk PRIMARY KEY ( idlibro );

ALTER TABLE libros ADD CONSTRAINT "libros __UN" UNIQUE ( isbm );

-- libros numpagina Debe ser mayor a 50 y menor a 2000.
ALTER TABLE libros ADD CONSTRAINT numPagina_ck CHECK (numpaginas > 50 AND numpaginas < 2000);

-- libros nroedicion Debe ser mayor a 0.
ALTER TABLE libros ADD CONSTRAINT nroedicion_ck CHECK (nroedicion > 0);

-- libros anio Debe ser mayor a 1800.
ALTER TABLE libros ADD CONSTRAINT anio_ck CHECK (anio > 1800);

CREATE TABLE prestamos (
    idprestamo  INTEGER NOT NULL,
    idejemplar  INTEGER NOT NULL,
    cedula      VARCHAR2(15) NOT NULL,
    fechaprestamo  DATE NOT NULL,
    fechavencimiento DATE NOT NULL,
    fechadevolucion DATE,
    observacion VARCHAR2(2000)
);

ALTER TABLE prestamos ADD CONSTRAINT prestamos_pk PRIMARY KEY ( idprestamo );

-- prestamos fechaDevolucion No debe ser inferior a la fecha de prestamo.
ALTER TABLE prestamos ADD CONSTRAINT fechadevolucion_ck CHECK (fechadevolucion >
fechaprestamo);

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 );

-- usuarios email Debe nescesariamente incluir el caracter especial "@" y debe
-- incluir al menos dos caracteres antes y despues del arroba.
ALTER TABLE usuarios ADD CONSTRAINT email_ck CHECK(email LIKE(' _@ _'));

```

```

ALTER TABLE areas_conocimiento
  ADD CONSTRAINT areas_conocimiento FOREIGN KEY ( codareapadre )
    REFERENCES areas_conocimiento ( codarea );

ALTER TABLE autorias
  ADD CONSTRAINT autorias_autores_fk FOREIGN KEY ( idautor )
    REFERENCES autores ( idautor );

ALTER TABLE autorias
  ADD CONSTRAINT "autorias_libros_FK" FOREIGN KEY ( idlibro )
    REFERENCES libros ( idlibro );

ALTER TABLE ejemplares
  ADD CONSTRAINT "ejemplares_libros_FK" FOREIGN KEY ( idlibro )
    REFERENCES libros ( idlibro );

ALTER TABLE libros
  ADD CONSTRAINT "libros_areas_conocimiento_FK" FOREIGN KEY ( codarea )
    REFERENCES areas_conocimiento ( codarea );

ALTER TABLE libros
  ADD CONSTRAINT "libros_editoriales_FK" FOREIGN KEY ( idedit )
    REFERENCES editoriales ( idedit );

ALTER TABLE prestamos
  ADD CONSTRAINT prestamos_ejemplares_fk FOREIGN KEY ( idejemplar )
    REFERENCES ejemplares ( idejemplar );

ALTER TABLE prestamos
  ADD CONSTRAINT prestamos_usuarios_fk FOREIGN KEY ( cedula )
    REFERENCES usuarios ( cedula );

```

```
-- Oracle SQL Developer Data Modeler Summary Report:
```

```

--
-- 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.	Creando una función o método que se encargue de calcular y actualizar el campo autores.numLibros para cada autor. Esta función debe contar el número de libros asociados a cada autor en la tabla autorias.
Cada libro debe tener al menos un autor.	Realizar una validación en el lado del servidor al ingresar datos del libro para asegurar que al menos un autor esté asociado.
En una misma fecha a un mismo usuario no se le puede prestar dos ejemplares del mismo libro.	Implementar una función de validación de la aplicación para evitar préstamos duplicados del mismo libro al mismo usuario en la misma fecha.

Cálculo del espacio en disco

ESPACIO TABLAS PROYECTADO A 5 AÑOS					
Relaciones	Tamaño aproximado de cada registro (en Bytes)	Proyeccion a 5 años		Total a 5 años	
		Registros	Bytes	Bytes	MB
areas_conocimiento	290	120	34800	34800	0,0348
autores	1.051.196	10000	10511960000	10511960000	10511,96
autorias	20	120000	2400000	2400000	2,4
editoriales	414	400	165600	165600	0,1656
ejemplares	59	200.000	11800000	11800000	11,8
libros	2.003.060	40.000	80122400000	80122400000	80122,4
prestamos	2054	400.000	821600000	821600000	821,6
usuarios	495	9400	4653000	4653000	4,653
Resultado	3057588	779920	91475013400	91475013400	91475,0134

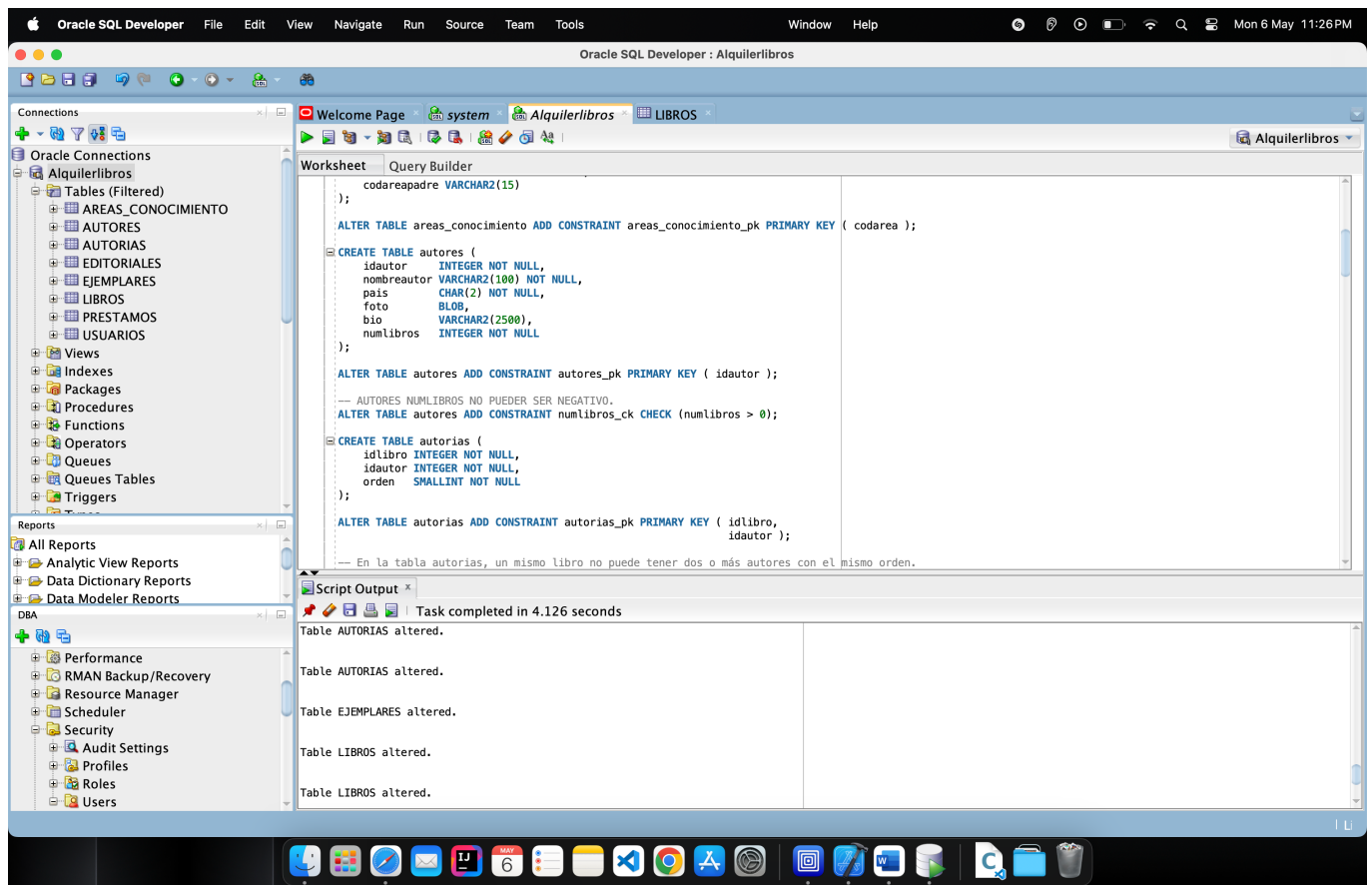
TOTALES PROYECTADOS A 5 AÑOS (EN MBYTES)	
Espacio Tablas	91475,0134
+20% (Actualizacion y Metadata)	18295,0027
Total Tablas	109770,0161
Espacio Adicional (archivos)	0
Total TABLAS + OTROS	109770,0161
+50% (Índices & Otros)	54885,00805
TOTAL	164655,0242

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



Captura(s) de pantalla 2

