

**Informe Final A+S:**  
Creación de Sistema de BD para Préstamos de Libros en Colegios

**MÓDULO:** TALLER DE BASES DE DATOS – INFORME DE AVANCE A+S

**SEMANA**: 9 (ENE 2021)

**Docente:** PABLO CELEDÓN

**Estudiantes:** ROBERTO ARÉVALO / VICTOR TORO / ANDRÉS GATICA

# **Índice**

[1. Índice 2](#_Toc94087601)

[2. Diagnóstico 3](#_Toc94087602)

[a) Descripción de la problemática encontrada 3](#_Toc94087603)

[b) Antecedentes 3](#_Toc94087604)

[c) Problema 4](#_Toc94087605)

[3. Anexo 4](#_Toc94087606)

[4. Bibliografía 15](#_Toc94087607)



Imagen referencial: *www.portalprogramas.com/bibliotecas-empresa*

# **Diagnóstico**

Se ha identificado una necesidad de transformación digital en la comunidad de escuelas y colegios nacionales, en relación a la gestión de préstamos de libros a sus respectivos alumnos. Esta demanda de servicio se ha visto incrementado de manera exponencial debido a los efectos de la pandemia del COVID-19 en Chile, que ha obligado a millones de estudiantes a la opción de conectarse vía internet desde sus computadores a los establecimientos educacionales. Y resulta vital conseguir un libro físico que les permita apoyarse en las materias que allí se encuentran contenidas.

A continuación, se exponen sus tres etapas que ayudan a entender su génesis y contexto.

# **Descripción de la problemática encontrada**

Existen muchos colegios que aún mantienen sistemas de catálogos y control de préstamos de material educativo de forma manual en fichas y kardex, etc. O con sistemas adaptados para el control de stock y préstamos, tales como planillas electrónicas. No es lo óptimo si se considera que existe un riesgo de pérdida de datos o tiempos de respuesta en búsqueda y control efectivo.

El desafío que significa dar solución al problema arriba planteado quiere decir que se debe crear un diseño entidad relación que permita gestionar los datos de una biblioteca de modo que los alumnos disponen de un código de matricula que además les da acceso a la biblioteca y también se necesitara almacenar su Id\_Alumno, Nombre, Rut, Teléfono, Dirección y Correo. Ello supone la presencia de notebook y computadores estacionarios para que los operadores gestionen.

Además, se necesita identificar al bibliotecario que prestara los libros en diferentes turnos almacenado los siguientes datos, Id\_Bibliotecario, Nombre, Rut, Teléfono.

La biblioteca almacena libros que presta a los alumnos y alumnas, de ellos se almacenara su Id Libro, Autor, Genero, Año y Titulo.

Se quiere controlar cada préstamo que se realiza almacenando su Id\_prestamo, Numero Préstamo, Fecha Préstamo, Fecha Devolución.

# **Antecedentes**

El sistema escolar chileno dispone de 12.000 establecimientos educacionales, de los cuales una gran cantidad de ellos carece de un sistema de control de préstamos de libros informático, siendo actualmente su gestión llevada manualmente en tarjetas de cartón.

Para solucionar esto se pide diseñar un modelo de base de datos a través del cual se obtenga la información solicitada de préstamos de libros, mediante consultas SQL a la base de datos. Basar el proyecto en sistema de base de datos de la licencia GPL y poner a disposición pública todo el proyecto y la documentación en GitHub:  
b1) Consultas DDL b2) Manipulación DML b3) Agrupación b4) Vistas

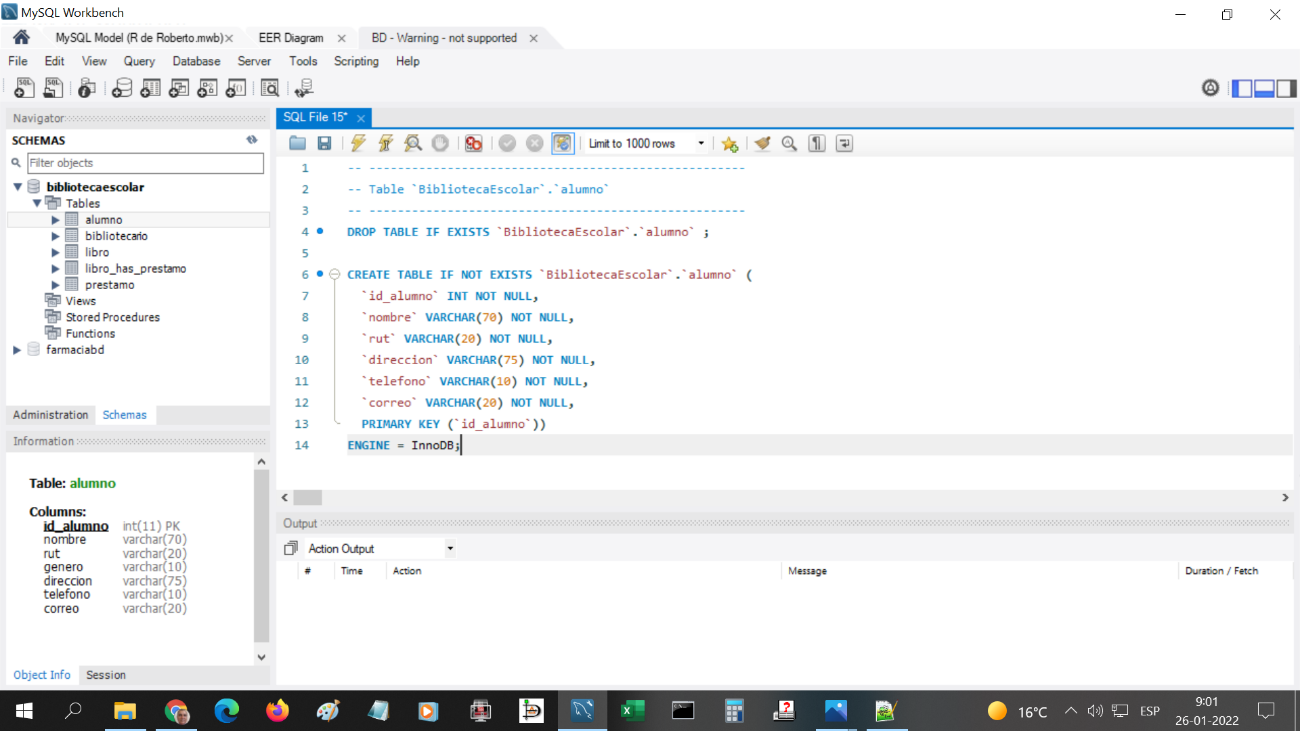
# **Problema**

Es nuestra estimación que este tipo de proyectos debe ser realizado al amparo de una licitación que el Ministerio de Educación impulse. Y es mediante la Dirección de Compras y Contratación Pública, Dirección ChileCompra, es que se puede materializar este tipo de proyectos. Porque la solución requerida para la gestión de préstamos de libros requiere, además del programa de bases de datos indicado junto a un SGBD robusto, de una infraestructura TI no menor: equipos informáticos, licencias de software, internet y mantención continua del hardware y software involucrado en el proyecto nacional.

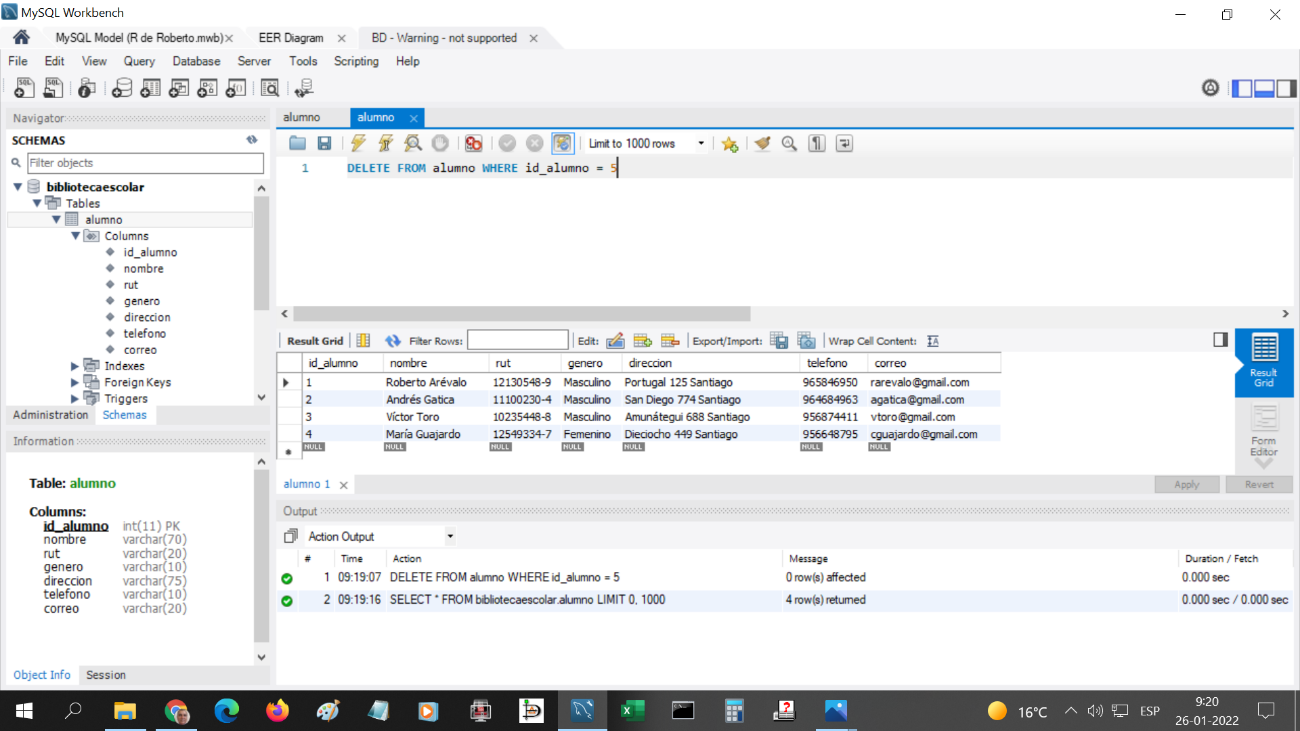
# **Anexo**

Contiene la secuencia gráfica de los pantallazos obtenidos en la creación de la base de datos A+S Proyecto Biblioteca Escolar.

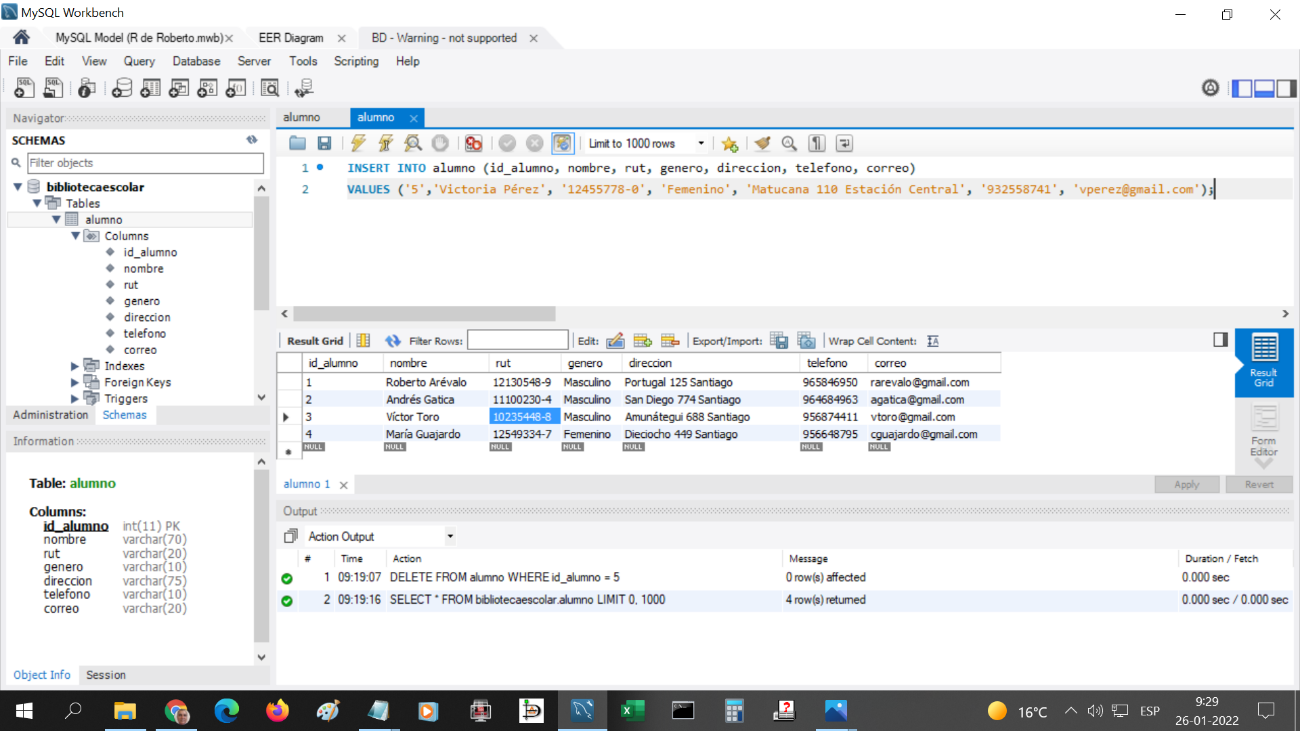
1) CREATE Tabla ALUMNO



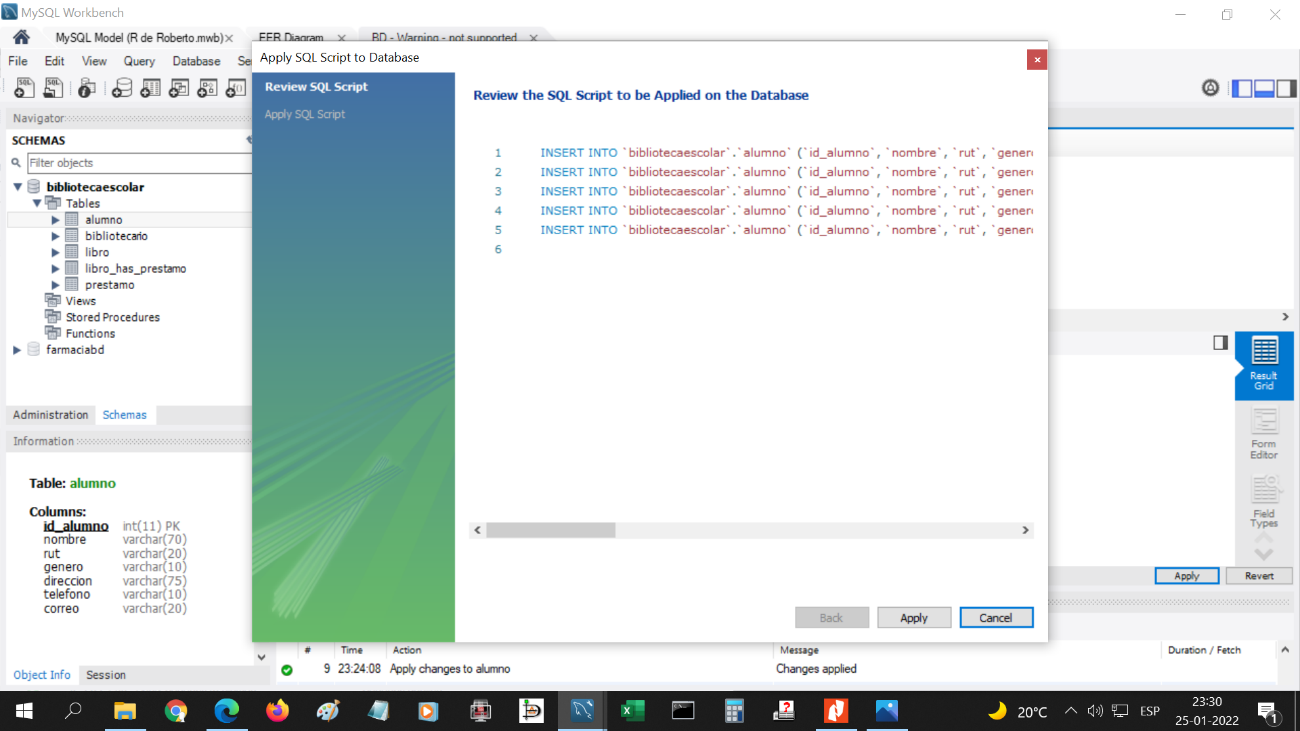
2) DELETE Tabla ALUMNO



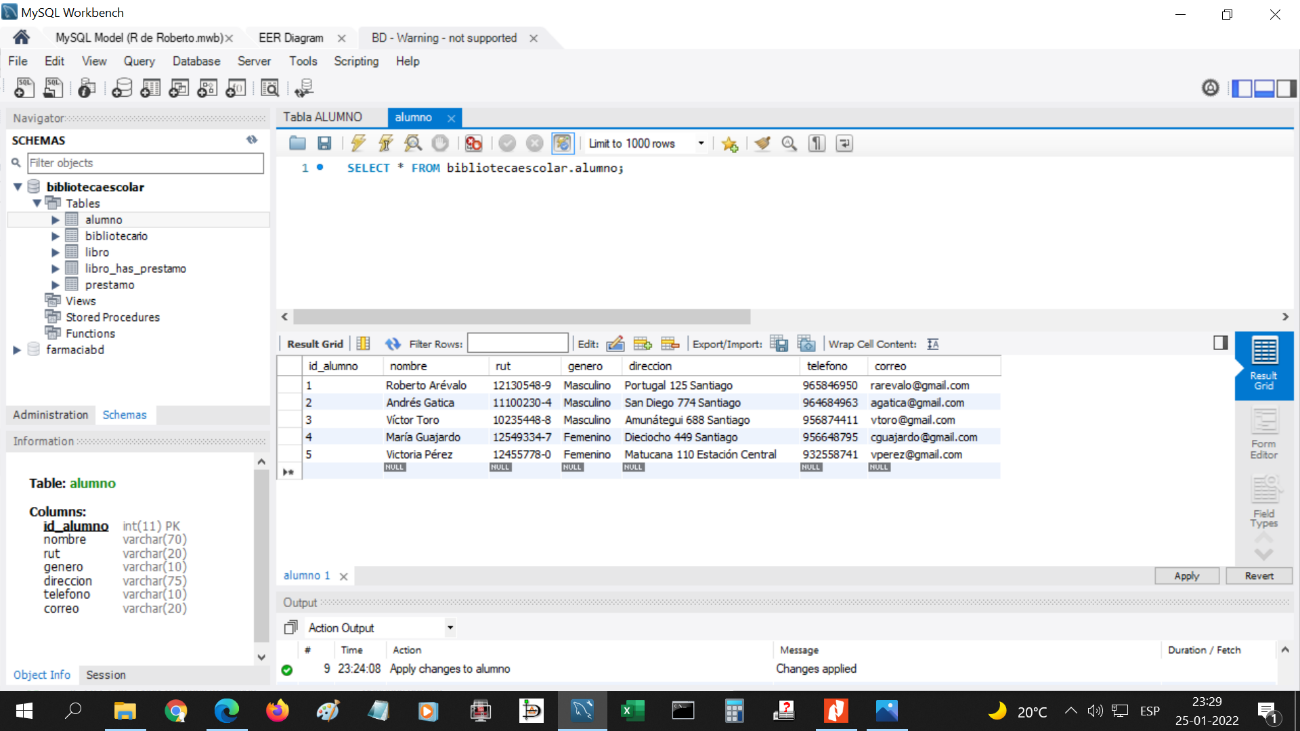
3) INSERT (registro borrado) Tabla ALUMNO



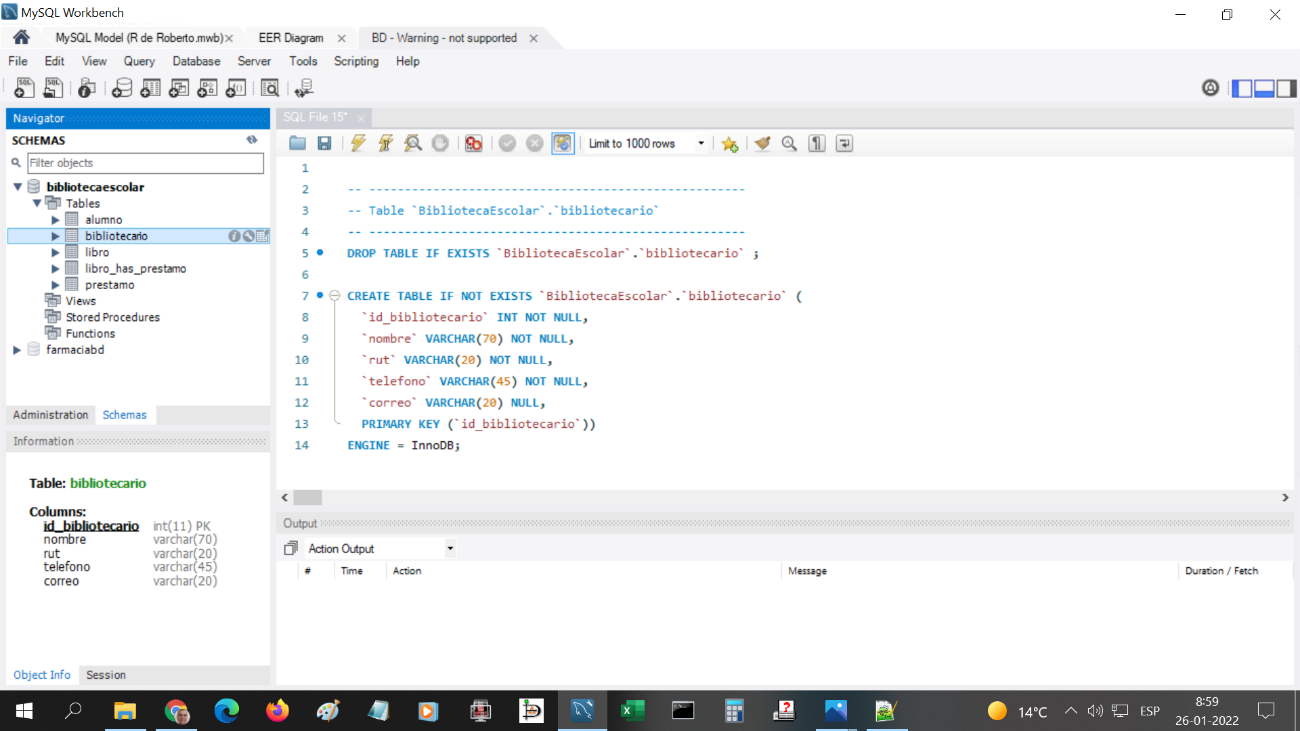
4) INSERT Tabla ALUMNO



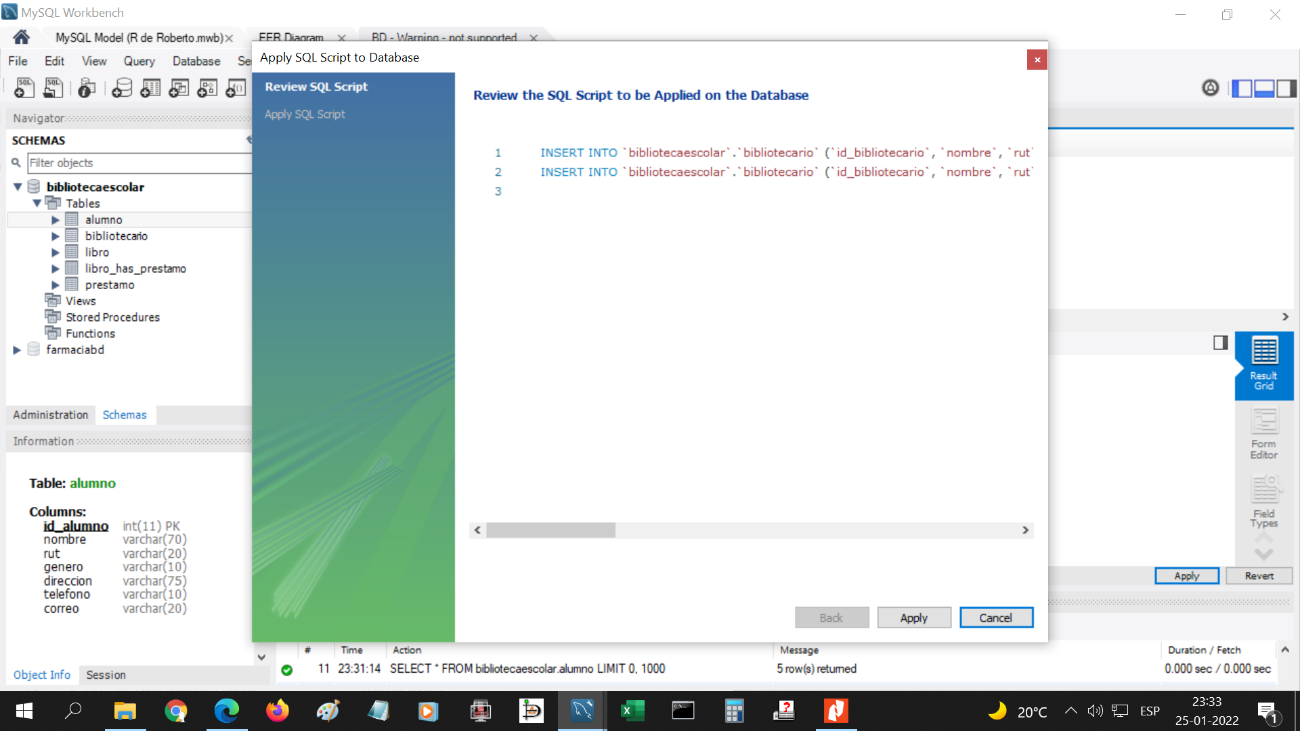
5) SELECT Tabla ALUMNO



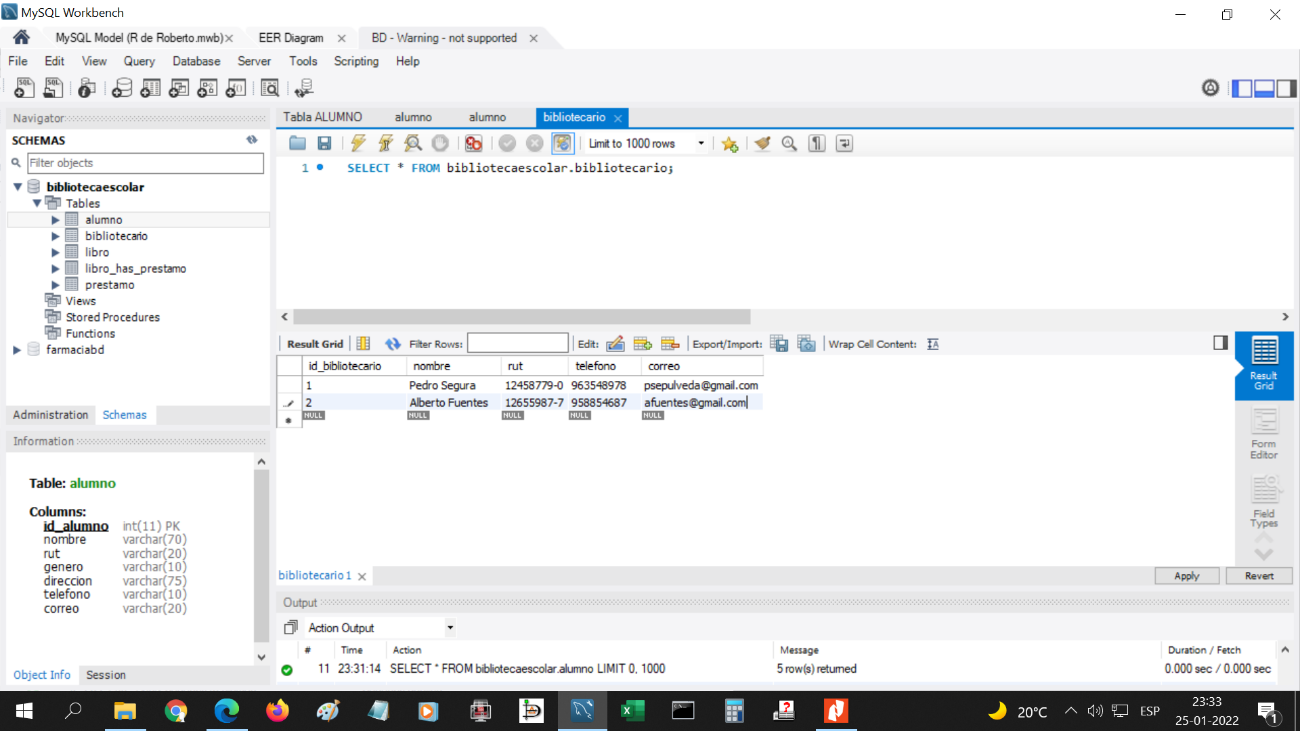
6) CREATE Tabla BIBLIOTECARIO



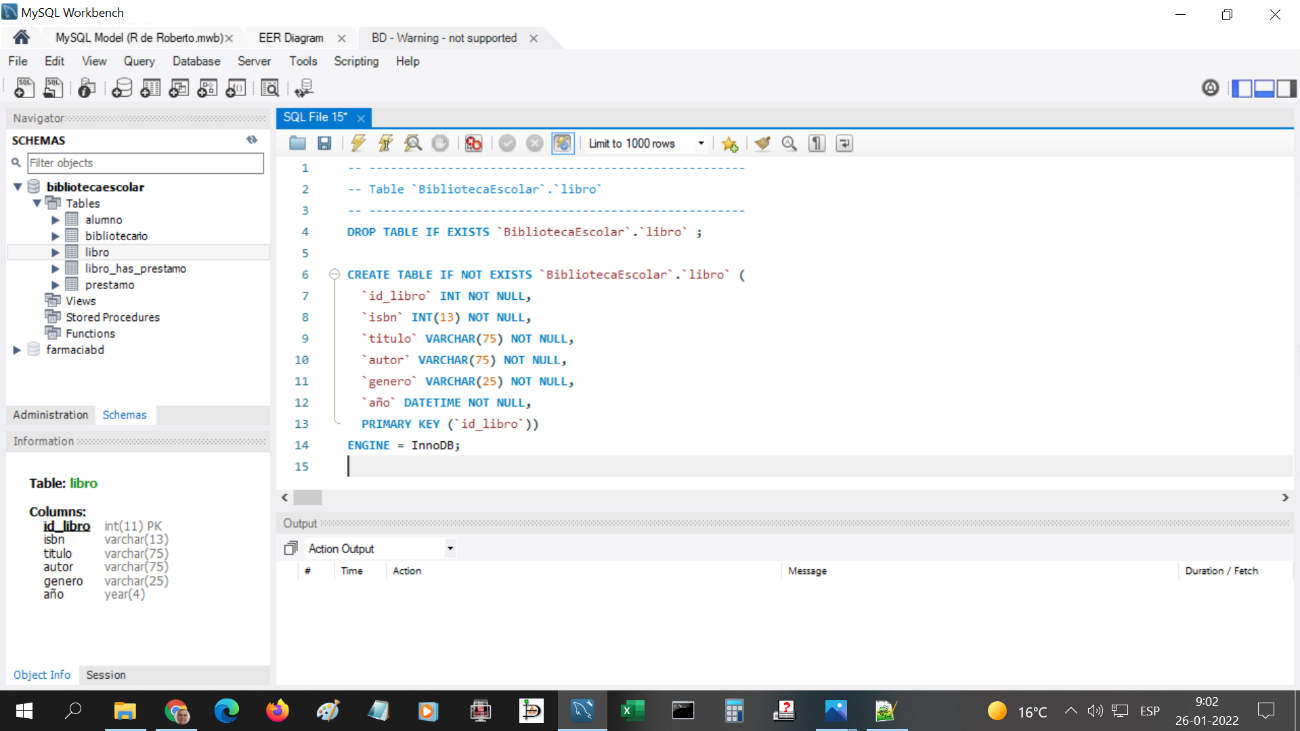
7) INSERT Tabla BIBLIOTECARIO



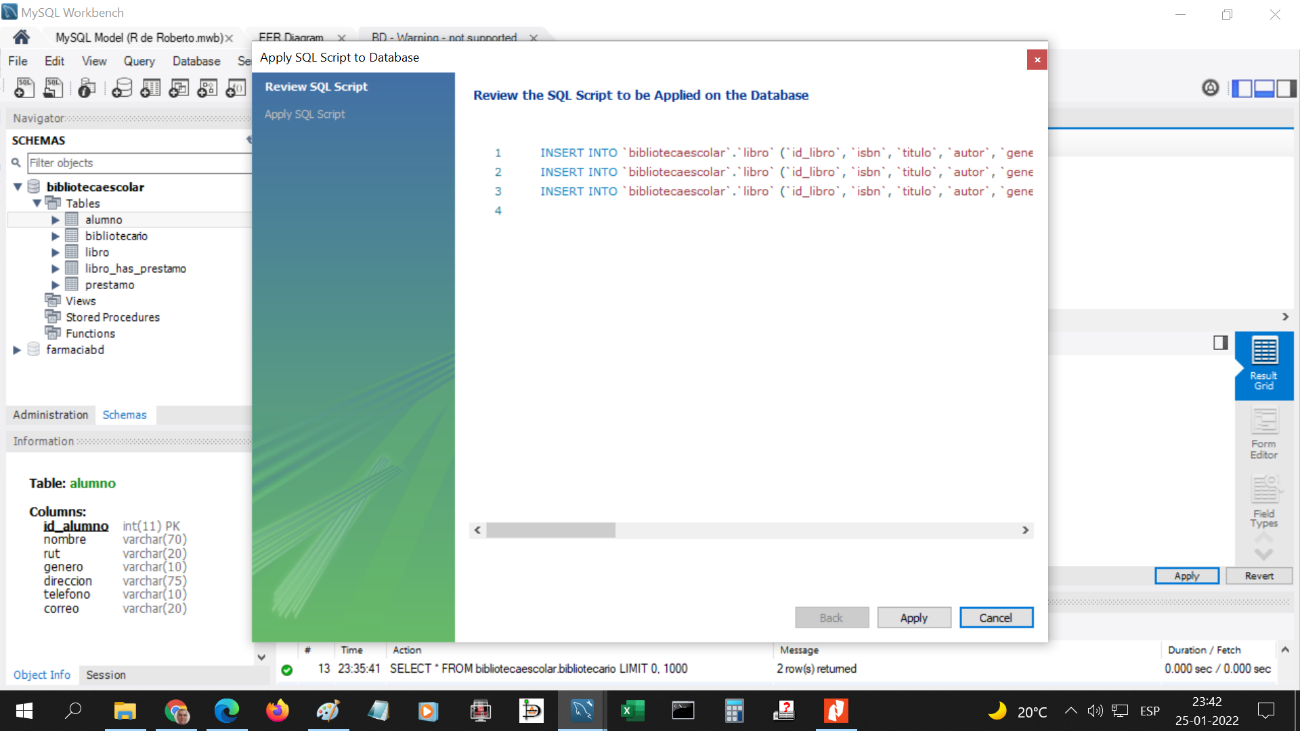
8) SELECT Tabla BIBLIOTECARIO



9) CREATE Tabla LIBRO



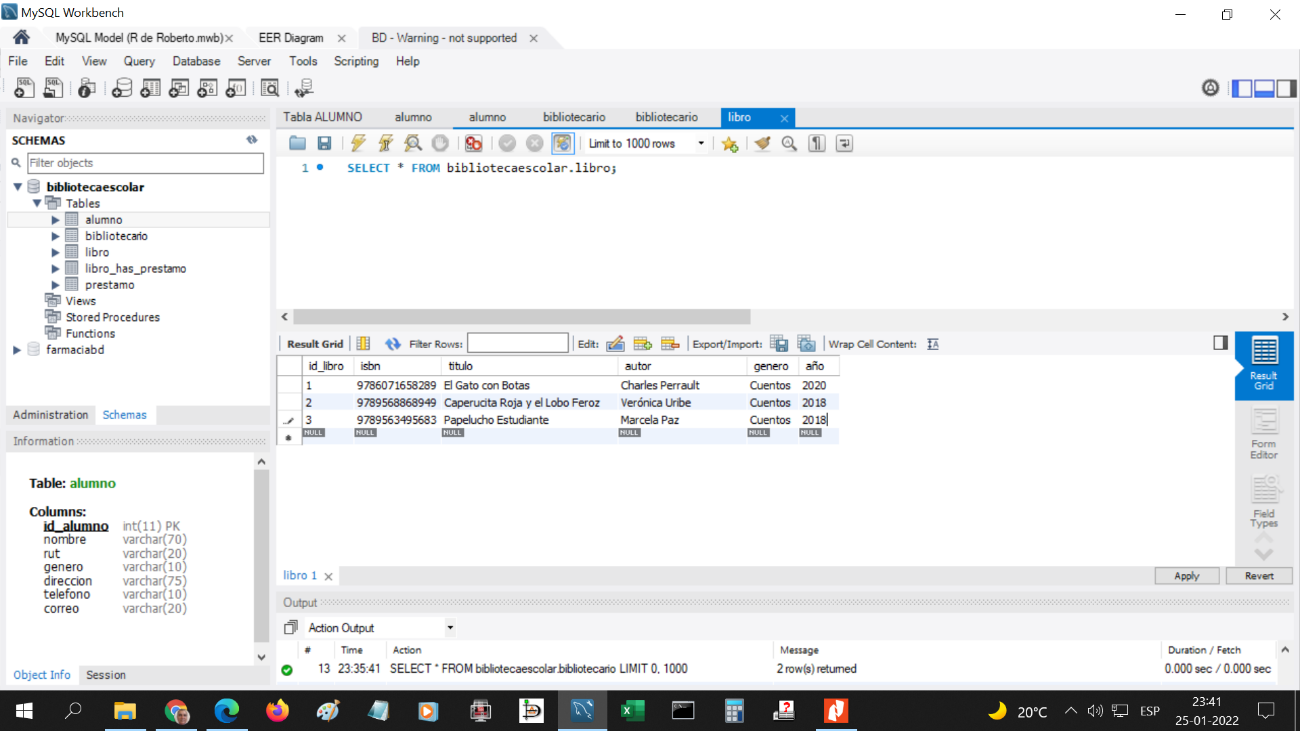
10) INSERT Tabla LIBRO



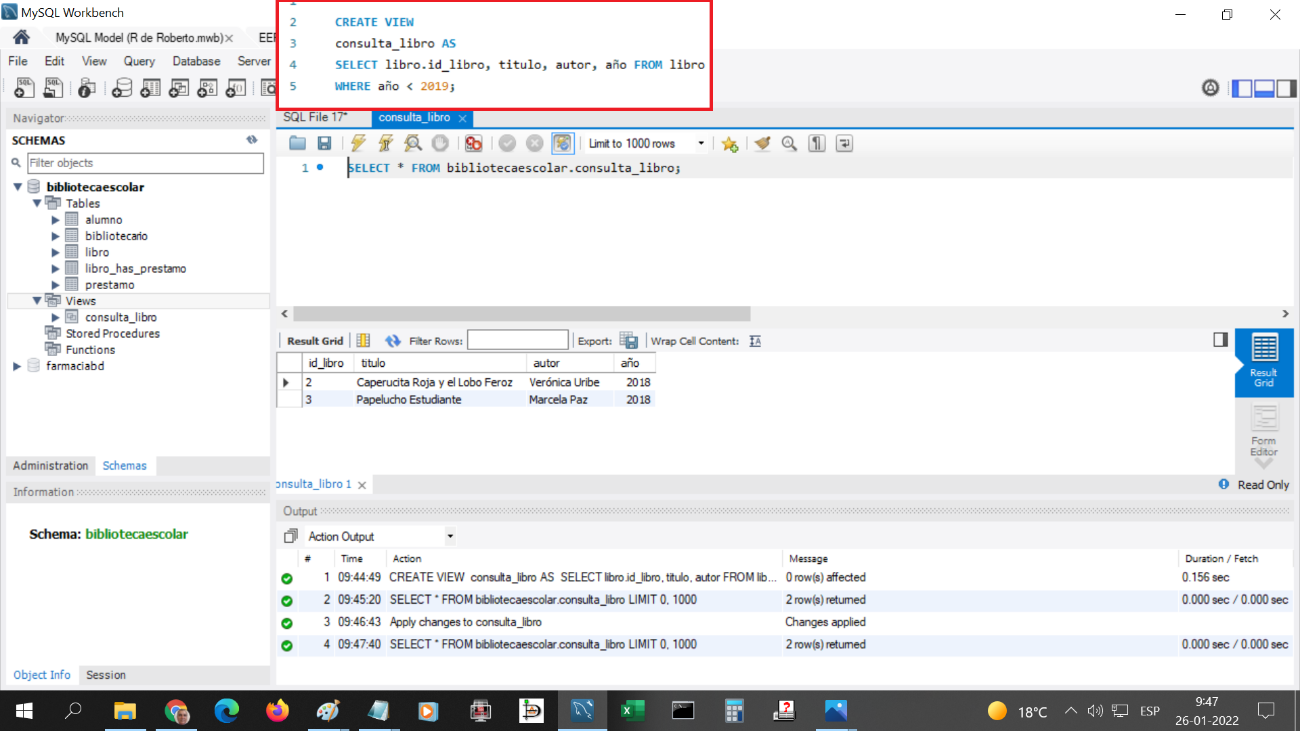
11) ORDER BY Tabla LIBRO



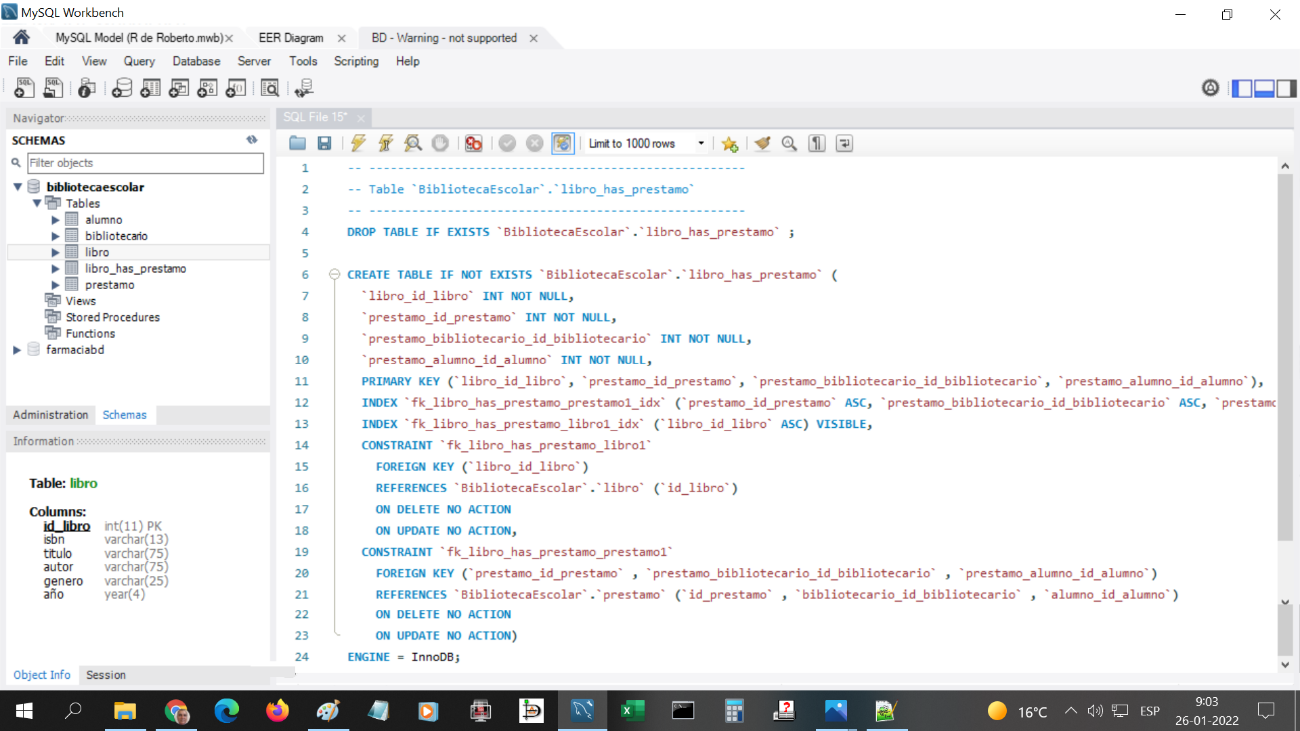
12) SELECT Tabla LIBRO



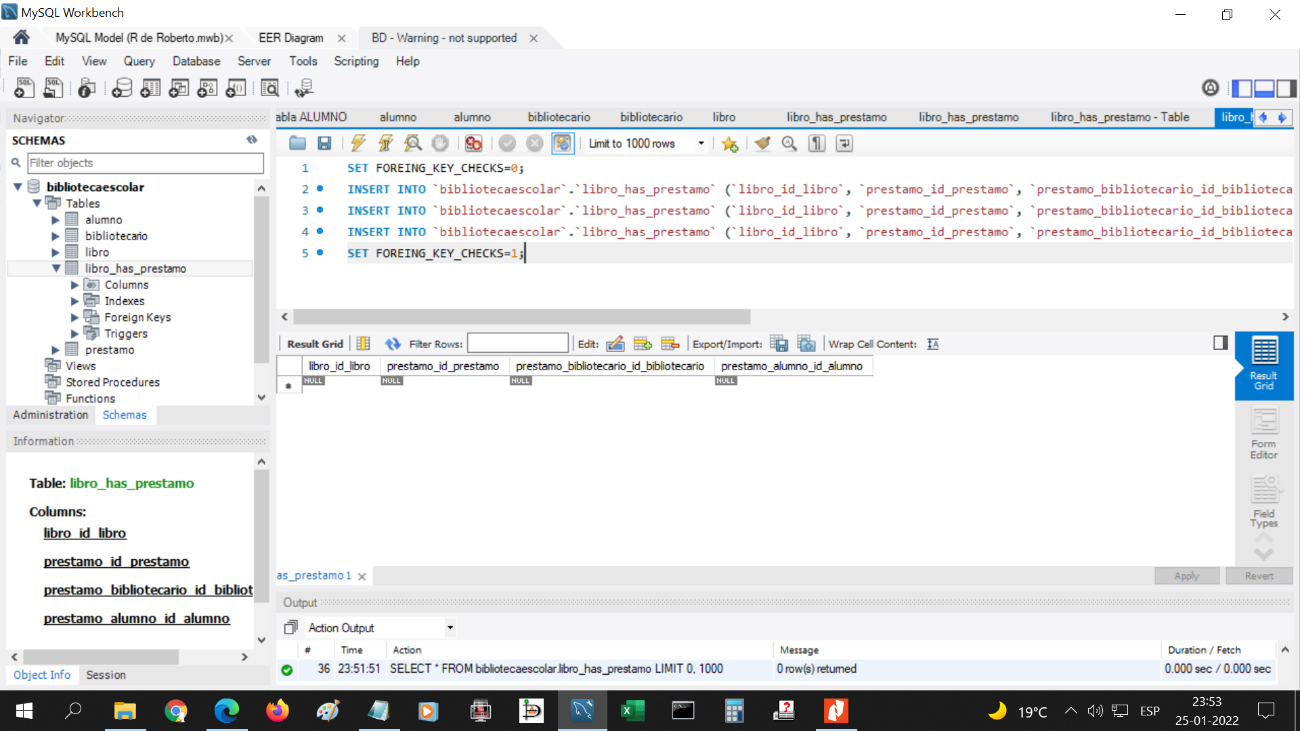
13) VISTA Tabla LIBRO



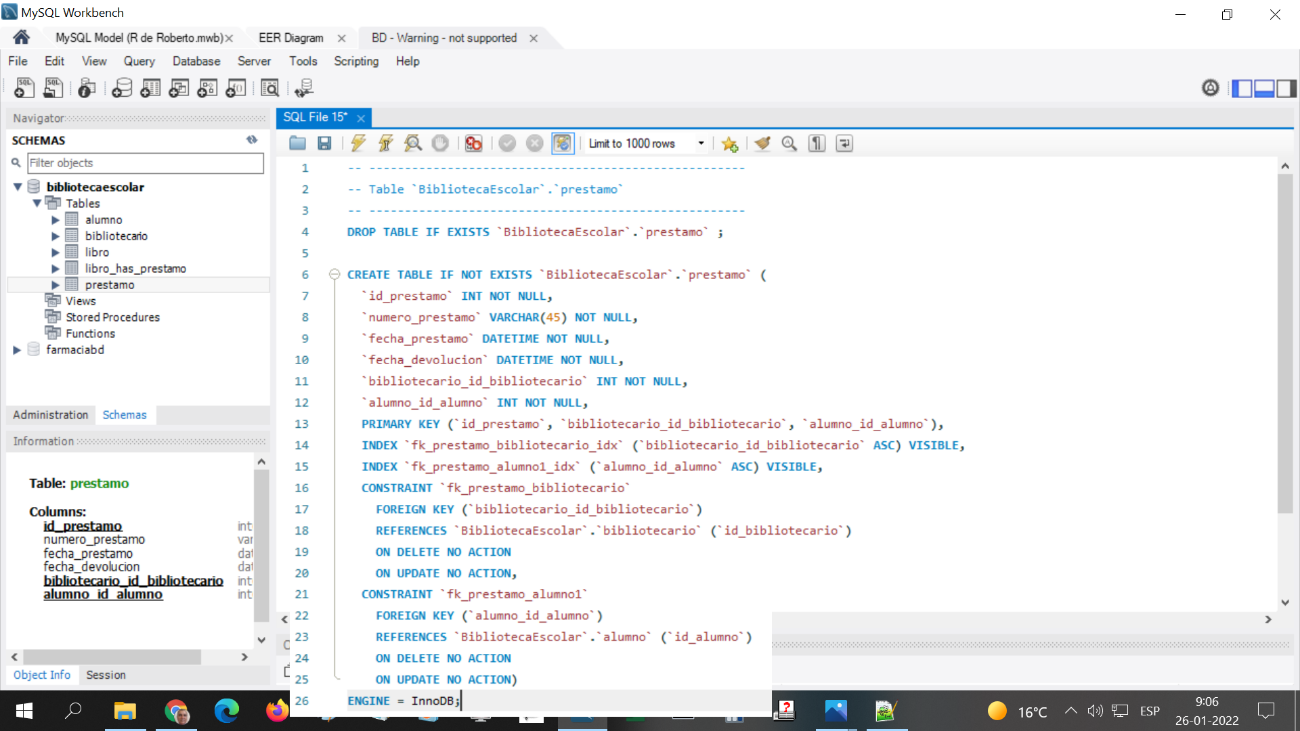
14) CREATE Tabla LIBRO\_HAS\_PRESTAMO



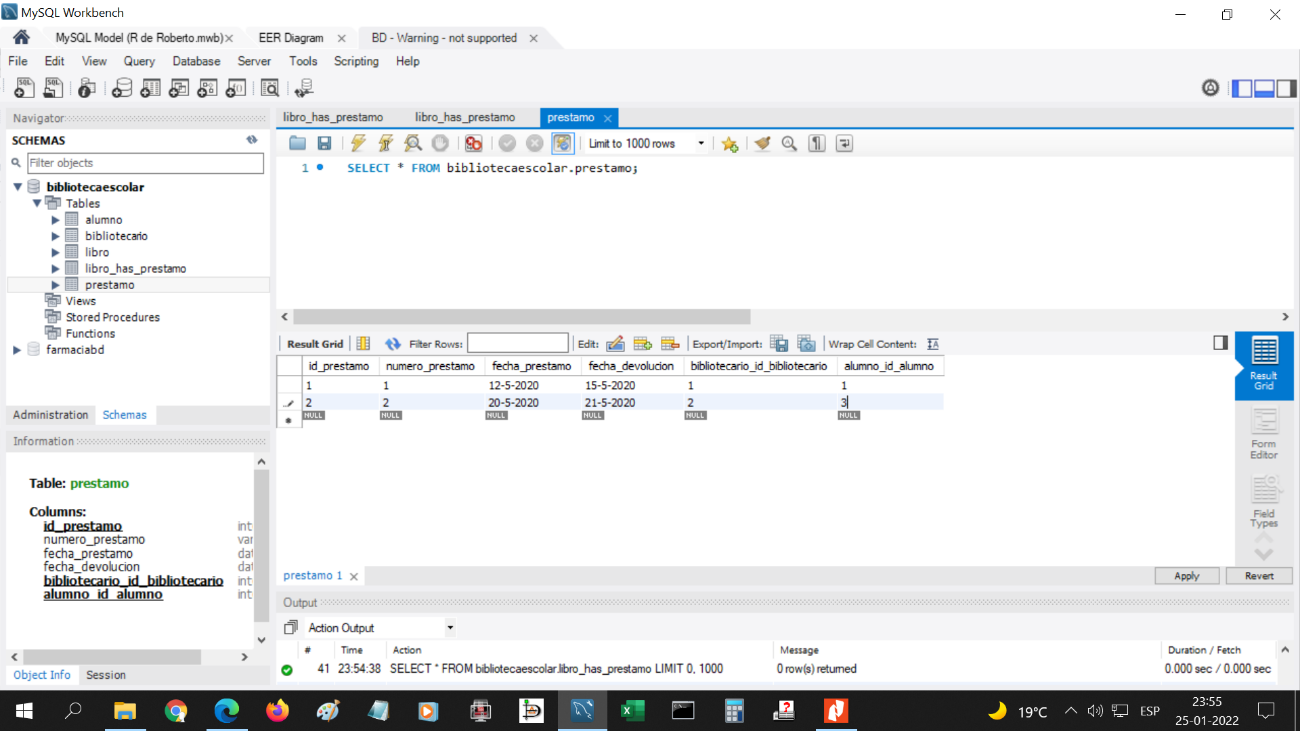
15) INSERT Tabla LIBRO\_HAS\_PRESTAMO



16) CREATE Tabla PRESTAMO



17) SELECT Tabla PRESTAMO



Script Base de Datos BibliotecaEscolar

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema BibliotecaEscolar

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema BibliotecaEscolar

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `BibliotecaEscolar` DEFAULT CHARACTER SET utf8 ;

USE `BibliotecaEscolar` ;

-- -----------------------------------------------------

-- Table `BibliotecaEscolar`.`bibliotecario`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `BibliotecaEscolar`.`bibliotecario` ;

CREATE TABLE IF NOT EXISTS `BibliotecaEscolar`.`bibliotecario` (

`id\_bibliotecario` INT NOT NULL,

`nombre` VARCHAR(70) NOT NULL,

`rut` VARCHAR(20) NOT NULL,

`telefono` VARCHAR(45) NOT NULL,

`correo` VARCHAR(20) NULL,

PRIMARY KEY (`id\_bibliotecario`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `BibliotecaEscolar`.`alumno`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `BibliotecaEscolar`.`alumno` ;

CREATE TABLE IF NOT EXISTS `BibliotecaEscolar`.`alumno` (

`id\_alumno` INT NOT NULL,

`nombre` VARCHAR(70) NOT NULL,

`rut` VARCHAR(20) NOT NULL,

`direccion` VARCHAR(75) NOT NULL,

`telefono` VARCHAR(10) NOT NULL,

`correo` VARCHAR(20) NOT NULL,

PRIMARY KEY (`id\_alumno`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `BibliotecaEscolar`.`prestamo`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `BibliotecaEscolar`.`prestamo` ;

CREATE TABLE IF NOT EXISTS `BibliotecaEscolar`.`prestamo` (

`id\_prestamo` INT NOT NULL,

`numero\_prestamo` VARCHAR(45) NOT NULL,

`fecha\_prestamo` DATETIME NOT NULL,

`fecha\_devolucion` DATETIME NOT NULL,

`bibliotecario\_id\_bibliotecario` INT NOT NULL,

`alumno\_id\_alumno` INT NOT NULL,

PRIMARY KEY (`id\_prestamo`, `bibliotecario\_id\_bibliotecario`, `alumno\_id\_alumno`),

INDEX `fk\_prestamo\_bibliotecario\_idx` (`bibliotecario\_id\_bibliotecario` ASC) VISIBLE,

INDEX `fk\_prestamo\_alumno1\_idx` (`alumno\_id\_alumno` ASC) VISIBLE,

CONSTRAINT `fk\_prestamo\_bibliotecario`

FOREIGN KEY (`bibliotecario\_id\_bibliotecario`)

REFERENCES `BibliotecaEscolar`.`bibliotecario` (`id\_bibliotecario`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_prestamo\_alumno1`

FOREIGN KEY (`alumno\_id\_alumno`)

REFERENCES `BibliotecaEscolar`.`alumno` (`id\_alumno`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `BibliotecaEscolar`.`libro`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `BibliotecaEscolar`.`libro` ;

CREATE TABLE IF NOT EXISTS `BibliotecaEscolar`.`libro` (

`id\_libro` INT NOT NULL,

`isbn` INT(13) NOT NULL,

`titulo` VARCHAR(75) NOT NULL,

`autor` VARCHAR(75) NOT NULL,

`genero` VARCHAR(25) NOT NULL,

`año` DATETIME NOT NULL,

PRIMARY KEY (`id\_libro`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `BibliotecaEscolar`.`libro\_has\_prestamo`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `BibliotecaEscolar`.`libro\_has\_prestamo` ;

CREATE TABLE IF NOT EXISTS `BibliotecaEscolar`.`libro\_has\_prestamo` (

`libro\_id\_libro` INT NOT NULL,

`prestamo\_id\_prestamo` INT NOT NULL,

`prestamo\_bibliotecario\_id\_bibliotecario` INT NOT NULL,

`prestamo\_alumno\_id\_alumno` INT NOT NULL,

PRIMARY KEY (`libro\_id\_libro`, `prestamo\_id\_prestamo`, `prestamo\_bibliotecario\_id\_bibliotecario`, `prestamo\_alumno\_id\_alumno`),

INDEX `fk\_libro\_has\_prestamo\_prestamo1\_idx` (`prestamo\_id\_prestamo` ASC, `prestamo\_bibliotecario\_id\_bibliotecario` ASC, `prestamo\_alumno\_id\_alumno` ASC) VISIBLE,

INDEX `fk\_libro\_has\_prestamo\_libro1\_idx` (`libro\_id\_libro` ASC) VISIBLE,

CONSTRAINT `fk\_libro\_has\_prestamo\_libro1`

FOREIGN KEY (`libro\_id\_libro`)

REFERENCES `BibliotecaEscolar`.`libro` (`id\_libro`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_libro\_has\_prestamo\_prestamo1`

FOREIGN KEY (`prestamo\_id\_prestamo` , `prestamo\_bibliotecario\_id\_bibliotecario` , `prestamo\_alumno\_id\_alumno`)

REFERENCES `BibliotecaEscolar`.`prestamo` (`id\_prestamo` , `bibliotecario\_id\_bibliotecario` , `alumno\_id\_alumno`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

# **Bibliografía**

“12.000 establecimientos”. Tomado de:

<https://www.t13.cl/noticia/nacional/mineduc-espera-que-mas-de-3-millones-de-estudiantes-regresan-a-clases-esta-semana>

“Entendiendo las Bibliotecas CRA”. Tomado de:

<https://bibliotecas-cra.cl/sites/default/files/publicaciones/estandarescra.pdf>

“Funciones y formatos de SQL Convert Date”. Tomado de:

<https://www.sqlshack.com/es/funciones-y-formatos-de-sql-convert-date>

**---\*---**