# Instrucciones

1. Debes **grabar en vídeo la pantalla de tu PC** durante toda la duración del examen. **Tras entregar tu solución** del examen al profesor:

* Detendrás la grabación del vídeo.
* Renombrarás el vídeo para que incluya tus apellidos y nombre.
* Subirás el vídeo a una carpeta del ordenador de profesor que se te indicará.

1. Abre este documento con el **LIBREOFFICE**, no con otro editor.
2. Debes guardar este fichero **sin cambiarlo de formato,** UTILIZA EL FORMATO DOCX **(NO UTILICES ODT** ni otro formato),
3. **Renombra este fichero para que, en lugar de APELLIDOS, NOMBRE ponga tus verdaderos apellidos y nombre.**
4. Al finalizar debes de comprimir tu ficheros de soluciones en **ZIP** no 7z, con el nombre:

**APELLIDOS, NOMBRE -** Examen BD mayo 2024.zip

1. el fichero zip con tu solución se entregará en el **MOODLE** de la asignatura.
2. **Sólo puedes usar el material que entrega el profesor específicamente para realiar el examen.**

# Consultas

Pon la solución a cada consulta debajo del número indicado:

#1

select \*

from empleados natural left join departamentos;

#2

select \*

from empleados

where FECHA\_ALTA between '1981-1-1' and '1981-6-30';

#3

select \*

from empleados natural join departamentos

where foto is null and (localidad = "Sevilla" or localidad="Barcelona");

#4

select distinct jefe.apellido1,jefe.apellido2,jefe.nombre,jefe.oficio

from empleados jefe natural join empleados subordinado order by oficio asc,apellido1;

#5

select count(\*)

from empleados

where comision is null and oficio= 'vendedor' and dep\_no = 30;

#6

select \*

from empleados

where oficio =(

select oficio

from empleados

where apellido1='Cortes');

select \*

from productos

where DESCRIPCION like '%mesa%' and DESCRIPCION like '%despacho%' and PRECIO\_ACTUAL <100000;

#8

select sum(ifnull(salario,0) + ifnull(comision,0)),dnombre,dep\_no

from empleados natural join departamentos

group by dep\_no

having count(\*) > 2;

#9

select \* from empleados

where salario >= (

select max(salario) from empleados);

#10

#11

start transaction;

update empleados

set salario = salario - salario \*5/100

where oficio="director";

commit;

#12

insert into empleados

values(9328,'12345678Z','Concepcion','Arenal','Ponte','abogada',null,'1981-11-17',120000.00,null,10,981456789,null);

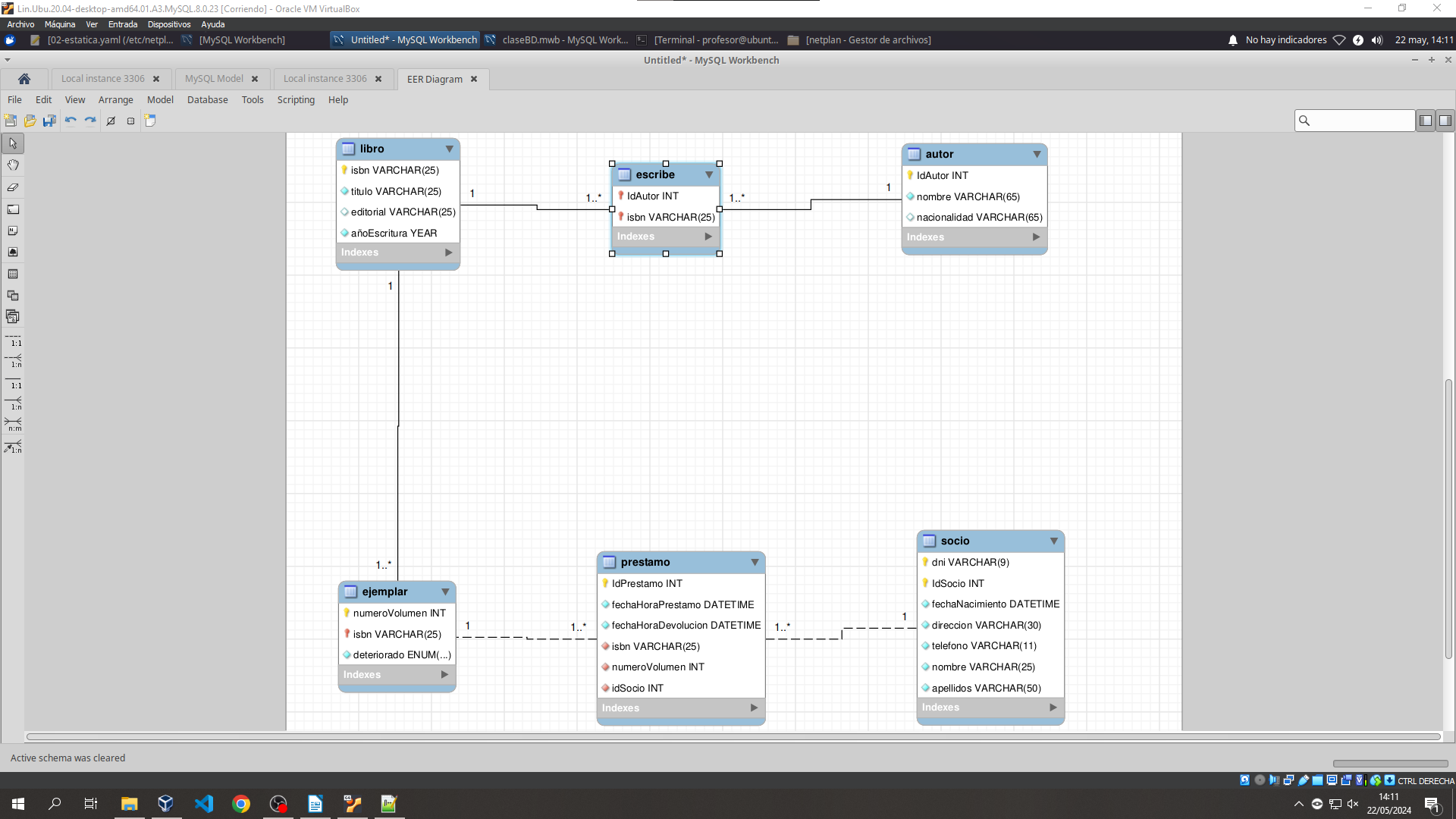
#13

select apellido1,nombre,salario,oficio from empleados natural join departamentos

where dnombre = contabilidad or dnombre= ventas and salario between 100000 and 220.000;

# Pasar de modelo E-R a modelo Relacional (tablas)

1. FOTO con el diseño de la BD que has realizado en Workbench tras realizar el REVERSE ENGINEER. La foto debe permitir revisar claramente la estructura de la BD (nombres de atributos y tablas, relaciones), demodo que el texto de las tablas debe ser grande para leerlo bien.



1. Comandos del script que has escrito para crea la Base de Datos.

drop database if exists biblioteca;

create database biblioteca;

use biblioteca;

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

-- Table autor

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

drop table if exists autor;

CREATE TABLE IF NOT EXISTS autor (

IdAutor int auto\_increment NOT NULL,

nombre VARCHAR(65) NOT NULL,

nacionalidad VARCHAR(65),

PRIMARY KEY (IdAutor)

);

insert into autor

values(null,'pepe','español'),

(null,'sergio','español');

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

-- Table libro

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

drop table if exists libro;

CREATE TABLE IF NOT EXISTS libro (

isbn varchar(25) not null,

titulo varchar(25) not null,

editorial varchar(25) ,

añoEscritura year(4) not null,

PRIMARY KEY (isbn)

);

insert into libro

values('158548G','La historia interminable','Galaxia','2012'),

('545648G','La historia Terminable','Galaxia','2011');

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

-- Table escribe

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

drop table if exists escribe;

CREATE TABLE IF NOT EXISTS escribe (

IdAutor int auto\_increment not null,

isbn varchar (25) not null,

PRIMARY KEY (IdAutor,isbn),

constraint fk\_autor\_escribe

foreign key (IdAutor)

references autor (IdAutor),

constraint fk\_isbn\_libro

foreign key (isbn)

references libro (isbn)

on update cascade

on delete no action

);

insert into escribe

values(null,'545648G'),

(null,'158548G');

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

-- Table ejemplar

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

drop table if exists ejemplar;

CREATE TABLE IF NOT EXISTS ejemplar (

numeroVolumen int not null,

isbn varchar (25) not null,

deteriorado enum('true','false') not null,

PRIMARY KEY (numeroVolumen,isbn),

constraint fk\_isbn\_lib

foreign key (isbn)

references libro (isbn)

on update cascade

on delete no action

);

insert into ejemplar

values(1,'158548G','true'),

(2,'545648G','false');

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

-- Table socio

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

drop table if exists socio;

CREATE TABLE IF NOT EXISTS socio (

dni varchar(9) not null,

IdSocio int auto\_increment not null,

fechaNacimiento datetime not null,

direccion varchar(30) not null,

telefono varchar (11) not null,

nombre varchar(25) not null,

apellidos varchar(50) not null,

PRIMARY KEY (IdSocio),

unique UQ\_SOC\_DNI (dni)

);

insert into socio

values('15854845E',null,'2012-12-12','calle cabañas','677666555','Sergio','Iglesias'),

('15854843J',null,'2012-11-16','calle Flor','677777555','Pedro','Formoso');

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

-- Table prestamo

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

drop table if exists prestamo;

CREATE TABLE IF NOT EXISTS prestamo (

IdPrestamo int auto\_increment not null,

fechaHoraPrestamo datetime not null,

fechaHoraDevolucion datetime not null,

isbn varchar(25) not null,

numeroVolumen int not null,

idSocio int not null,

PRIMARY KEY (IdPrestamo),

constraint fk\_isbn\_libro\_prestamo

foreign key (numeroVolumen,isbn)

references ejemplar(numeroVolumen,isbn)

on update cascade

on delete no action,

constraint fk\_socio\_idS

foreign key (IdSocio)

references socio(IdSocio)

on update cascade

on delete no action

);

insert into prestamo

values(null,'2012-12-12','2012-02-02','158548G',1,1),

(null,'2012-11-02','2012-12-06','545648G',2,2);