Número: 8 APELIDOS e Nome:: Formoso Costa Sergio

# 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 nadal 2023.zip

1. el fichero zip con tu solución se entregará en el **MOODLE** de la asignatura.
2. **No puedes utilizar ningún material aparte del que se te entrega específicamente para este examen (no puedes usar ni apuntes, ni ejercicios, ni ....)**

# Consultas

Pon la solución a cada consulta debajo del número indicado:

#1 select count(cliente\_no) from clientes;

#2 select min(salario) as 'salario mas bajo', max(salario) as 'salario mas alto',avg(salario) as 'salario medio' from empleados;

#3 select apellido1,salario,oficio from empleados where (oficio='vendedor' or oficio='director') and salario < 250000;

#4 select empleados.\*,dnombre from empleados inner join departamentos on empleados.dep\_no = departamentos.dep\_no ;

#5 select localidad,cliente\_no,debe from clientes order by localidad desc,nombre asc;

#6 select nombre,descripcion from clientes inner join pedidos on clientes.CLIENTE\_NO = pedidos.CLIENTE\_NO inner join productos on productos.PRODUCTO\_NO;

#7 select \* from productos order by PRECIO\_ACTUAL asc limit 3;

#8 select \* from productos where DESCRIPCION like ('%mesa%despacho%');

#9 select \* from empleados where FECHA\_ALTA between '1981-01-01' and '1981-06-30';

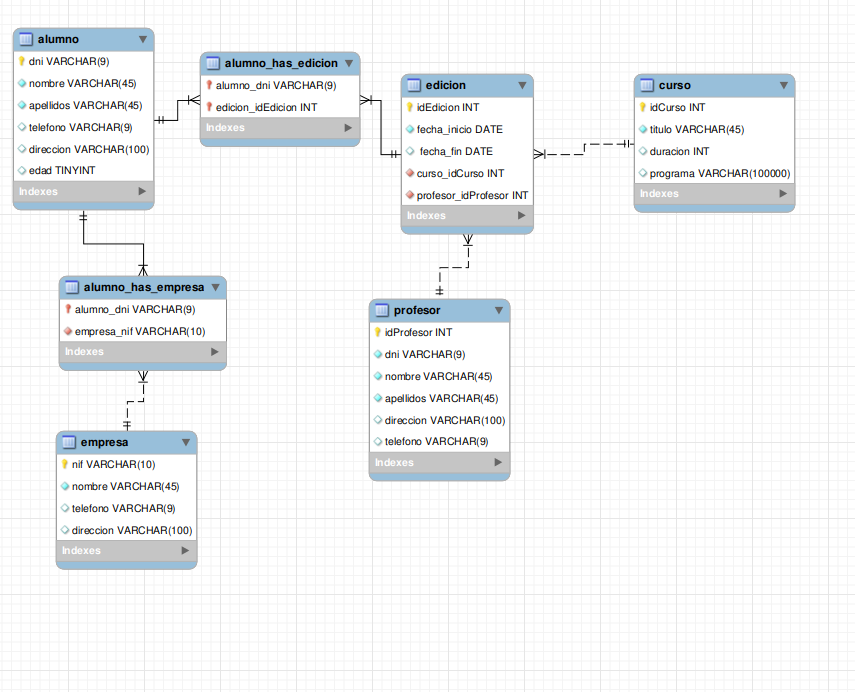
#10 select nombre, (ifnull(salario,0)+ifnull(comision,0)) from empleados;

#11 select count(\*) from empleados inner join departamentos on departamentos.dep\_no = empleados.dep\_no where oficio='vendedor';

#12 select count(NOMBRE)as 'EMPLEADO' ,count(JEFE) as 'JEFE' from empleados;

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

1. Foto del diseño de la BD que has realizado en Workbench



1. Comandos del que se genera al realizar el FORWARD ENGINEER al diagrama que has creado.

-- 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 ejercicio 4

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

DROP SCHEMA IF EXISTS `ejercicio 4` ;

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

-- Schema ejercicio 4

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

CREATE SCHEMA IF NOT EXISTS `ejercicio 4` ;

SHOW WARNINGS;

USE `ejercicio 4` ;

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

-- Table `alumno`

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

DROP TABLE IF EXISTS `alumno` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `alumno` (

`dni` VARCHAR(9) NOT NULL,

`nombre` VARCHAR(45) NOT NULL,

`apellidos` VARCHAR(45) NOT NULL,

`telefono` VARCHAR(9) NULL,

`direccion` VARCHAR(100) NULL,

`edad` TINYINT NULL,

PRIMARY KEY (`dni`))

ENGINE = InnoDB;

SHOW WARNINGS;

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

-- Table `curso`

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

DROP TABLE IF EXISTS `curso` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `curso` (

`idCurso` INT NOT NULL AUTO\_INCREMENT,

`titulo` VARCHAR(45) NOT NULL,

`duracion` INT NULL,

`programa` VARCHAR(100000) NULL,

PRIMARY KEY (`idCurso`))

ENGINE = InnoDB;

SHOW WARNINGS;

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

-- Table `profesor`

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

DROP TABLE IF EXISTS `profesor` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `profesor` (

`idProfesor` INT NOT NULL,

`dni` VARCHAR(9) NOT NULL,

`nombre` VARCHAR(45) NOT NULL,

`apellidos` VARCHAR(45) NOT NULL,

`direccion` VARCHAR(100) NULL,

`telefono` VARCHAR(9) NULL,

PRIMARY KEY (`idProfesor`))

ENGINE = InnoDB;

SHOW WARNINGS;

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

-- Table `edicion`

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

DROP TABLE IF EXISTS `edicion` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `edicion` (

`idEdicion` INT NOT NULL AUTO\_INCREMENT,

`fecha\_inicio` DATE NOT NULL,

` fecha\_fin` DATE NULL,

`curso\_idCurso` INT NOT NULL,

`profesor\_idProfesor` INT NOT NULL,

PRIMARY KEY (`idEdicion`),

INDEX `fk\_edicion\_curso1\_idx` (`curso\_idCurso` ASC) VISIBLE,

INDEX `fk\_edicion\_profesor1\_idx` (`profesor\_idProfesor` ASC) VISIBLE,

CONSTRAINT `fk\_edicion\_curso1`

FOREIGN KEY (`curso\_idCurso`)

REFERENCES `curso` (`idCurso`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_edicion\_profesor1`

FOREIGN KEY (`profesor\_idProfesor`)

REFERENCES `profesor` (`idProfesor`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

SHOW WARNINGS;

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

-- Table `empresa`

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

DROP TABLE IF EXISTS `empresa` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `empresa` (

`nif` VARCHAR(10) NOT NULL,

`nombre` VARCHAR(45) NOT NULL,

`telefono` VARCHAR(9) NULL,

`direccion` VARCHAR(100) NULL,

PRIMARY KEY (`nif`))

ENGINE = InnoDB;

SHOW WARNINGS;

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

-- Table `alumno\_has\_edicion`

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

DROP TABLE IF EXISTS `alumno\_has\_edicion` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `alumno\_has\_edicion` (

`alumno\_dni` VARCHAR(9) NOT NULL,

`edicion\_idEdicion` INT NOT NULL,

PRIMARY KEY (`alumno\_dni`, `edicion\_idEdicion`),

INDEX `fk\_alumno\_has\_edicion\_edicion1\_idx` (`edicion\_idEdicion` ASC) VISIBLE,

INDEX `fk\_alumno\_has\_edicion\_alumno\_idx` (`alumno\_dni` ASC) VISIBLE,

CONSTRAINT `fk\_alumno\_has\_edicion\_alumno`

FOREIGN KEY (`alumno\_dni`)

REFERENCES `alumno` (`dni`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_alumno\_has\_edicion\_edicion1`

FOREIGN KEY (`edicion\_idEdicion`)

REFERENCES `edicion` (`idEdicion`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

SHOW WARNINGS;

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

-- Table `alumno\_has\_empresa`

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

DROP TABLE IF EXISTS `alumno\_has\_empresa` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `alumno\_has\_empresa` (

`alumno\_dni` VARCHAR(9) NOT NULL,

`empresa\_nif` VARCHAR(10) NOT NULL,

PRIMARY KEY (`alumno\_dni`),

INDEX `fk\_alumno\_has\_empresa\_empresa1\_idx` (`empresa\_nif` ASC) VISIBLE,

INDEX `fk\_alumno\_has\_empresa\_alumno1\_idx` (`alumno\_dni` ASC) VISIBLE,

CONSTRAINT `fk\_alumno\_has\_empresa\_alumno1`

FOREIGN KEY (`alumno\_dni`)

REFERENCES `alumno` (`dni`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_alumno\_has\_empresa\_empresa1`

FOREIGN KEY (`empresa\_nif`)

REFERENCES `empresa` (`nif`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

SHOW WARNINGS;

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

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

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