## Ejercicio de evaluación Módulo 2 día 5 semana 7

Una parte importante del lenguaje de consultas de bases de datos es su aplicación a las diversas situaciones en los sistemas.

**Objetivo**

Aplicar de forma conceptual la información proporcionada en el módulo 2 pudiendo realizar consultas simples a bases de datos para obtener resultados desde las tablas según los enunciados.

-

**Información proporcionada**

* Tabla Productos
* Datos en tabla
* Tipo de dato de columna
* Enunciados para elaborar consulta

**Modo de resolución**

Según la tabla entregada realice las siguientes consultas SQL para solicitar o realizar operaciones en los datos según los enunciados:

**Tabla Productos**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ProductoID (int) | NombreProducto (varchar) | Descripcion (int) | Precio (int) | Stock (in) |
| 1 | Camiseta | Camiseta negra simple de talla única | 10 | 16 |
| 2 | Pantalón | Pantalón largo azul tipo chino | 20 | 24 |
| 3 | Gorra | Gorra azul con el logo de los Yankees | 15 | 32 |
| 4 | Zapatillas | Zapatillas de running de color blanco y verde | 35 | 13 |

**Enunciados**

1.- Recuperar TODOS los datos de la tabla Productos (sin condición)

select \* from productos;

2.- Recuperar los datos de la fila donde  el nombre del producto sea “Camiseta”.

Select \* from productos where nombreproducto=’Camiseta’;

3.- Recuperar los datos de las filas donde Stock sea menor que 20

select \* from Productos where stock<20;

4.- Recuperar los datos de la fila con ProductoID 3

select \* from productos where productoid=3

5.- Eliminar las filas de productos donde el stock sea mayor o igual a 20

delete \* from productos where stock>=20

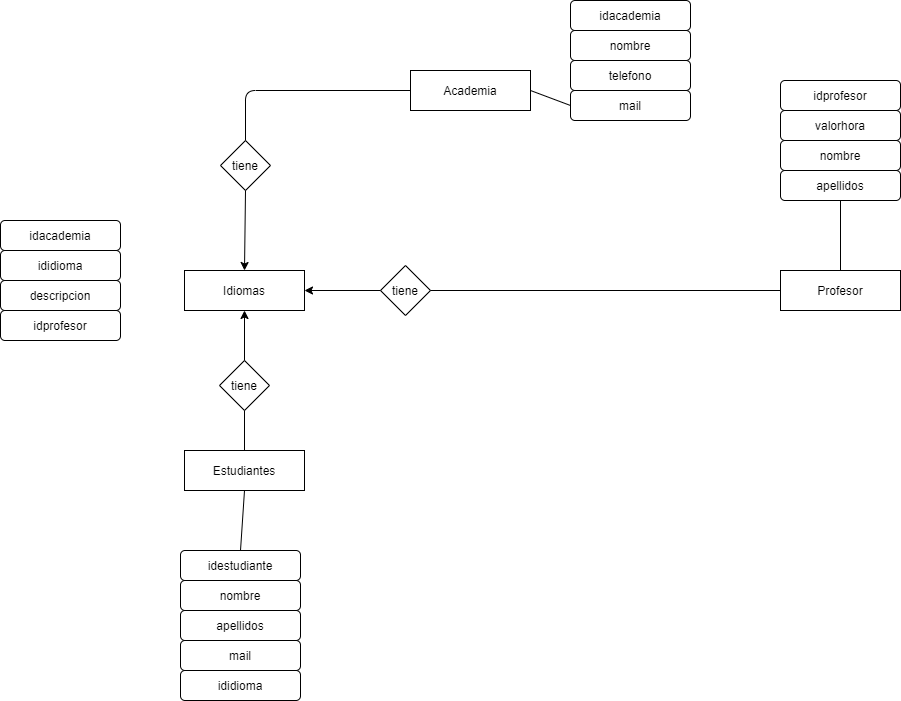
Formato de entrega

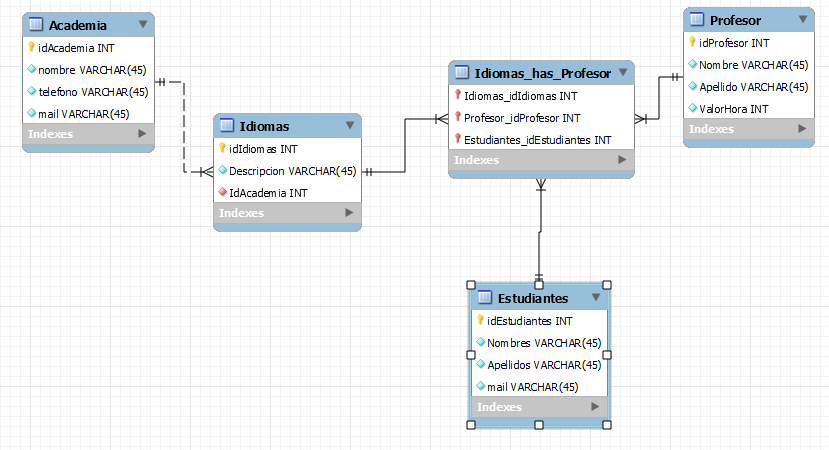
Para entregar este trabajo solo debes subir 1 archivo comprimido en formato .ZIP

Problematica

una academia de idiomas (languagelife) imparte muchos idiomas, cada idioma tiene muchos profesores y cada profesor puede impartir mas de un idioma

un estudiante, puede tener uno o mas idiomas





-- 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 control\_language

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

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

-- Schema control\_language

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

CREATE SCHEMA IF NOT EXISTS `control\_language` DEFAULT CHARACTER SET utf8 ;

USE `control\_language` ;

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

-- Table `control\_language`.`Academia`

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

CREATE TABLE IF NOT EXISTS `control\_language`.`Academia` (

`idAcademia` INT NOT NULL AUTO\_INCREMENT,

`nombre` VARCHAR(45) NOT NULL,

`telefono` VARCHAR(45) NOT NULL,

`mail` VARCHAR(45) NOT NULL,

PRIMARY KEY (`idAcademia`))

ENGINE = InnoDB;

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

-- Table `control\_language`.`Profesor`

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

CREATE TABLE IF NOT EXISTS `control\_language`.`Profesor` (

`idProfesor` INT NOT NULL AUTO\_INCREMENT,

`Nombre` VARCHAR(45) NOT NULL,

`Apellido` VARCHAR(45) NOT NULL,

`ValorHora` INT NOT NULL,

PRIMARY KEY (`idProfesor`))

ENGINE = InnoDB;

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

-- Table `control\_language`.`Idiomas`

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

CREATE TABLE IF NOT EXISTS `control\_language`.`Idiomas` (

`idIdiomas` INT NOT NULL AUTO\_INCREMENT,

`Descripcion` VARCHAR(45) NOT NULL,

`IdAcademia` INT NOT NULL,

PRIMARY KEY (`idIdiomas`),

INDEX `academia\_idx` (`IdAcademia` ASC),

CONSTRAINT `academia`

FOREIGN KEY (`IdAcademia`)

REFERENCES `control\_language`.`Academia` (`idAcademia`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `control\_language`.`Estudiantes`

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

CREATE TABLE IF NOT EXISTS `control\_language`.`Estudiantes` (

`idEstudiantes` INT NOT NULL AUTO\_INCREMENT,

`Nombres` VARCHAR(45) NOT NULL,

`Apellidos` VARCHAR(45) NOT NULL,

`mail` VARCHAR(45) NOT NULL,

PRIMARY KEY (`idEstudiantes`))

ENGINE = InnoDB;

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

-- Table `control\_language`.`Idiomas\_has\_Profesor`

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

CREATE TABLE IF NOT EXISTS `control\_language`.`Idiomas\_has\_Profesor` (

`Idiomas\_idIdiomas` INT NOT NULL,

`Profesor\_idProfesor` INT NOT NULL,

`Estudiantes\_idEstudiantes` INT NOT NULL,

PRIMARY KEY (`Idiomas\_idIdiomas`, `Profesor\_idProfesor`, `Estudiantes\_idEstudiantes`),

INDEX `fk\_Idiomas\_has\_Profesor\_Profesor1\_idx` (`Profesor\_idProfesor` ASC),

INDEX `fk\_Idiomas\_has\_Profesor\_Idiomas1\_idx` (`Idiomas\_idIdiomas` ASC),

INDEX `fk\_Idiomas\_has\_Profesor\_Estudiantes1\_idx` (`Estudiantes\_idEstudiantes` ASC),

CONSTRAINT `fk\_Idiomas\_has\_Profesor\_Idiomas1`

FOREIGN KEY (`Idiomas\_idIdiomas`)

REFERENCES `control\_language`.`Idiomas` (`idIdiomas`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Idiomas\_has\_Profesor\_Profesor1`

FOREIGN KEY (`Profesor\_idProfesor`)

REFERENCES `control\_language`.`Profesor` (`idProfesor`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Idiomas\_has\_Profesor\_Estudiantes1`

FOREIGN KEY (`Estudiantes\_idEstudiantes`)

REFERENCES `control\_language`.`Estudiantes` (`idEstudiantes`)

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;

select \* from academia;  
insert into academia(nombre,telefono,mail)values('Languagelife','982307788','982307788');  
  
select \* from idiomas;  
insert into idiomas(descripcion,idacademia) values('ingles',1);  
insert into idiomas(descripcion,idacademia) values('frances',1);  
insert into idiomas(descripcion,idacademia) values('ruso',1);  
insert into idiomas(descripcion,idacademia) values('español',1);  
  
select \* from profesor;  
insert into profesor (nombre,apellido,valorhora)values('inna','monroy','15000');  
insert into profesor (nombre,apellido,valorhora)values('francisco','monroy','12000');  
insert into profesor (nombre,apellido,valorhora)values('roxana','ivanova','10000');  
  
  
select \* from estudiantes;  
insert into estudiantes(nombres,apellidos,mail)values('camila','osorio','xxx@renolit.cl');  
insert into estudiantes(nombres,apellidos,mail)values('valentina','ivanova','xxx@sushi.cl');  
insert into estudiantes(nombres,apellidos,mail)values('kurt','russel','xxx@kurt.cl');  
insert into estudiantes(nombres,apellidos,mail)values('irina','kurnikova','xxx@rr.cl');  
insert into estudiantes(nombres,apellidos,mail)values('valeria','xxx','xxx@x.cl');  
insert into estudiantes(nombres,apellidos,mail)values('alina','ivanova','xxx@ruuu.cl');  
insert into estudiantes(nombres,apellidos,mail)values('carmen','pradenas','xxx@rr.cl');  
insert into estudiantes(nombres,apellidos,mail)values('jj','xx','xxx@xx.cl');  
insert into estudiantes(nombres,apellidos,mail)values('maria','herrera','xxx@xx.cl');  
  
  
  
select \* from idiomas\_has\_profesor;  
insert into idiomas\_has\_profesor (idiomas\_ididiomas,profesor\_idprofesor,estudiantes\_idestudiantes)values(1,1,1);  
insert into idiomas\_has\_profesor (idiomas\_ididiomas,profesor\_idprofesor,estudiantes\_idestudiantes)values(3,1,2);  
insert into idiomas\_has\_profesor (idiomas\_ididiomas,profesor\_idprofesor,estudiantes\_idestudiantes)values(1,1,2);  
insert into idiomas\_has\_profesor (idiomas\_ididiomas,profesor\_idprofesor,estudiantes\_idestudiantes)values(4,2,4);  
insert into idiomas\_has\_profesor (idiomas\_ididiomas,profesor\_idprofesor,estudiantes\_idestudiantes)values(4,3,5);  
insert into idiomas\_has\_profesor (idiomas\_ididiomas,profesor\_idprofesor,estudiantes\_idestudiantes)values(4,2,6);  
insert into idiomas\_has\_profesor (idiomas\_ididiomas,profesor\_idprofesor,estudiantes\_idestudiantes)values(1,1,6);  
insert into idiomas\_has\_profesor (idiomas\_ididiomas,profesor\_idprofesor,estudiantes\_idestudiantes)values(2,1,7);  
insert into idiomas\_has\_profesor (idiomas\_ididiomas,profesor\_idprofesor,estudiantes\_idestudiantes)values(1,1,7);  
insert into idiomas\_has\_profesor (idiomas\_ididiomas,profesor\_idprofesor,estudiantes\_idestudiantes)values(1,1,8);  
insert into idiomas\_has\_profesor (idiomas\_ididiomas,profesor\_idprofesor,estudiantes\_idestudiantes)values(3,3,8);  
insert into idiomas\_has\_profesor (idiomas\_ididiomas,profesor\_idprofesor,estudiantes\_idestudiantes)values(3,3,9);

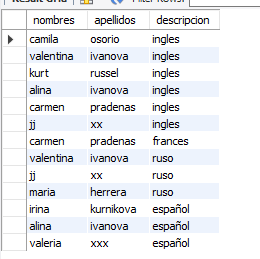
2.-

consultas:

-- listado de estudiantes   
select \* from estudiantes;



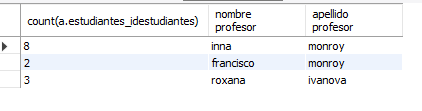
-- estuadiantes idiomas  
select nombres,apellidos,descripcion from estudiantes a , idiomas\_has\_profesor b, idiomas c  
where a.idEstudiantes=b.Estudiantes\_idEstudiantes  
and b.Idiomas\_idIdiomas=c.ididiomas;



-- estudiantes idiomas y profesor  
select nombres,apellidos,descripcion,Nombre as 'nombre profesor', apellido as 'apellido profesor'  
 from estudiantes a , idiomas\_has\_profesor b, idiomas c, profesor d  
where a.idEstudiantes=b.Estudiantes\_idEstudiantes  
and b.Idiomas\_idIdiomas=c.ididiomas  
and b.profesor\_idProfesor=d.idProfesor;



-- cuantos estudiantes por profesor  
select count(a.estudiantes\_idestudiantes), Nombre as 'nombre profesor', apellido as 'apellido profesor'   
from idiomas\_has\_profesor a, profesor b  
where a.Profesor\_idProfesor=b.idProfesor  
group by Profesor\_idProfesor;

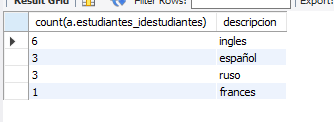


vistas:

estadistica cantidad de estudiantes por idiomas

create view estadistica\_de\_idiomas as  
select count(a.estudiantes\_idestudiantes), descripcion   
from idiomas\_has\_profesor a, idiomas b  
where a.idiomas\_ididiomas=b.ididiomas  
group by descripcion  
order by count(a.estudiantes\_idestudiantes) desc  
;

commit;  
  
select \* from estadistica\_de\_idiomas;



estudiantes con mas de un idioma cursando

create view estudiantes\_mas\_cursos as  
select count(Idiomas\_idIdiomas), concat(nombres,' ', apellidos)   
from idiomas\_has\_profesor a, estudiantes b  
where a.Estudiantes\_idEstudiantes=b.idEstudiantes  
group by Estudiantes\_idEstudiantes  
having count(Idiomas\_idIdiomas)>1  
order by count(Idiomas\_idIdiomas) desc;  
commit;  
select \* from estudiantes\_mas\_cursos;

