



DEPARTAMENTO:	DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN	CARRERA:	INGENIERÍA EN SOFTWARE		
ASIGNATURA:	SISTEMAS DE BASE DE DATOS	PERIODO LECTIVO:	MAYO – SEPTIEMBRE 2023	NIVEL:	4to
DOCENTE:	ING. ELEANA JEREZ, MSc.	NRC:	9752	PRÁCTICA N°:	2
TEMA DE LA PRÁCTICA:	Modelo lógico y físico.				

INTRODUCCIÓN:

En esta práctica de laboratorio, exploraremos tres aspectos fundamentales en el diseño y manejo de bases de datos: las consultas, el modelo lógico y el modelo físico. Estos elementos desempeñan un papel crucial en el desarrollo de sistemas de información eficientes y confiables.

En primer lugar, nos adentraremos en el mundo de las consultas, que nos permiten extraer información relevante de una base de datos. Aprenderemos a diseñar consultas utilizando lenguajes como SQL para obtener resultados personalizados y adaptados a nuestras necesidades.

Luego, nos sumergiremos en el modelo lógico de una base de datos, que representa la estructura y las relaciones entre las entidades. Exploraremos conceptos clave como tablas, claves primarias y foráneas, y relaciones uno a uno, uno a muchos y muchos a muchos. Aprenderemos a diseñar un modelo lógico coherente y bien estructurado que refleje con precisión los datos y las relaciones que deseamos almacenar.

Finalmente, abordaremos el modelo físico de la base de datos, que se centra en la implementación concreta en un sistema de gestión de bases de datos específico. Discutiremos aspectos como la elección del motor de base de datos, la creación de tablas y la optimización del rendimiento. Aprenderemos a traducir el modelo lógico en un esquema físico que cumpla con los requisitos de almacenamiento y permita un acceso eficiente a los datos.

OBJETIVOS:

Objetivo General

Aplicar los conocimientos adquiridos en clase sobre el entorno de trabajo con bases de datos MySQL, incluyendo phpMyAdmin y MySQLWorkbench, con el fin de diseñar, administrar y ejecutar consultas efectivas en una máquina local. Esto permitirá crear un entorno amigable y confiable para el manejo de bases de datos, facilitando la generación de scripts, la elaboración de modelos lógicos y conceptuales, y la obtención de resultados precisos en las consultas.

Objetivos Específicos

- Comprender el entorno de phpMyAdmin para de esta manera poder generar los scripts de los ejercicios del laboratorio.
- Comprender y trabajar en la base de datos MySQLWorkbench para de esta manera elaborar tanto el modelo lógico como el modelo conceptual de las bases de datos planteadas en los ejercicios del laboratorio.
- Realizar las consultas solicitadas en los ejercicios aplicando los conocimientos obtenidos en clase para de esta manera obtener los resultados correctos de las consultas.



MATERIALES:

EQUIPOS:

- Jose: PC (Hp Omen, I5 de 7 generación, 8 GB de ram, Gtx 1050).
- Martin: PC (Asus Tuf F17, I5 de 11ava, 16 GB de ram, Gtx 1650).
- Ricardo: PC (Dell, I7 de 11ava generación, 8 GB ram).
- Angelo: PC (Lenovo, I5 de 11ava generación, 8GB de ram, GTX 1050).

INSTRUCCIONES:

EJERCICIO 1

Se tiene una base de datos que contiene información sobre las observaciones realizadas a distintas especies de aves. Las tablas se muestran a continuación:

AVES			
Id_ave	Id_grupo	Nom_cientifico	Nom_vulgar
001	001	harpia	águila
002	001	Morphnus	águila monera
003	002	Milvago	Caracara
004	003	Polysoma	Gavilán
005	002	Ictinia	Elanio

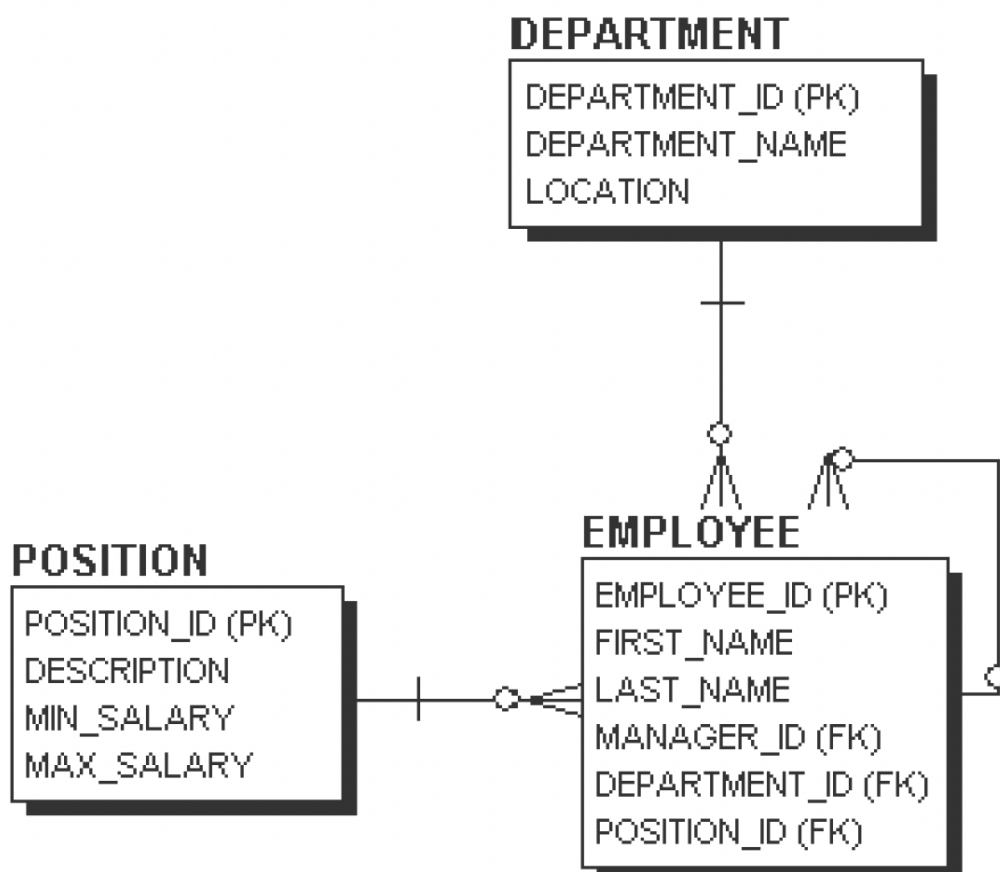
Grupos Ornitológicos			
Id_grupo	Patas	Picos	Dedos
001	Fuertes_patas	Fuertes_picos	Fuertes_dedos
002	Medio_patas	Medio_picos	Medio_dedos
003	Debil_patas	Debil_picos	Debil_dedos

Observadores				
Id_observador	Id_asociacion	Estado	Nombre	Direccion
001	001	Activo	Joel	Calderón
002	001	Pasivo	Steven	Tumbaco
003	002	Activo	Erick	Magdalena
004	002	Pasivo	Marley	Pifo
005	003	Activo	Angelo	Tumbaco

1. Montar un servidor de base de datos MySQL en una máquina local.
2. Utilizar phpmyadmin como SGBD para manipular los datos.
3. Utilizar MySQLWorkbench para elaborar el modelo lógico de la base de datos (Incluir las tablas que faltan en el gráfico y que se desarrolló en el proyecto de la Unidad 1) y generar el script en SQL del modelo conceptual.
4. Importar el script generado a phpmyadmin y obtener el modelo físico.
5. Ingresar registros para cada una de las tablas.
6. Realizar las siguientes consultas SQL a la base de datos, utilizando phpmyadmin.
 - a) Obtener el nombre científico de las aves que pertenecen al grupo ornitológico 001.
 - b) Obtener los nombres de los observadores que pertenecen a la asociación 002.
 - c) Obtener el nombre científico de las aves.
 - d) Obtener el nombre y la dirección de los observadores que están en estado pasivo.
 - e) Obtener los datos del águila.
 - f) Obtener el nombre científico del gavilán.
 - g) Obtener estado de Steven.
 - h) Dirección de los observadores que pertenecen a la asociación 002.
 - i) Obtener el grupo ornitológico de ictinia.

EJERCICIO 2

Se tiene una base de datos que contiene información sobre los empleados y departamentos de una empresa. Las tablas se muestran a continuación:

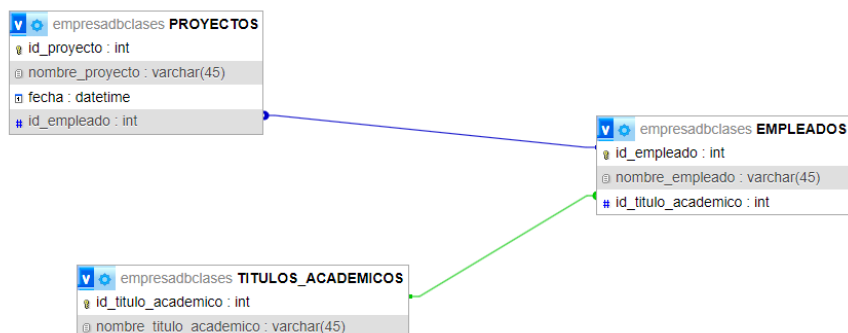


1. Utilizar MySQLWorkbench para elaborar el modelo lógico de la base de datos y generar el script SQL.
2. Importar el script generado a phpmyadmin y obtener el modelo físico.
3. Realizar las siguientes consultas SQL a la base de datos, utilizando MySQLWorkbench (cliente de base de datos).
 - a) Listar todos los datos de las tablas creadas.
 - b) Obtener un reporte de los empleados por orden alfabético.
 - c) Listar los valores mínimos y máximos en salario que pueden percibir los empleados por código de posición.
 - d) Listar los nombres de los departamentos y su localización.
 - e) Listar todos los departamentos cuya localización está en Quito o Guayaquil.
 - f) Listar los empleados que tienen en su apellido la letra “i”.
 - g) Mostrar la descripción de las posiciones de trabajo cuyos minimos salarios sean menores a \$1000.
 - h) Listar los empleados que no son de los departamentos 10, 20 o 30.

EJERCICIO 3

Se tiene una base de datos que contiene información sobre proyectos y empleados de una empresa. Las tablas se muestran a continuación:

1. Utilizar MySQLWorkbench para elaborar el modelo lógico de la base de datos y generar el script SQL.



2. Importar el script generado a phpmyadmin y obtener el modelo físico.
3. Realizar las siguientes consultas SQL a la base de datos, utilizando MySQLWorkbench (cliente de base de datos).
 - a) Mostrar los nombres de los empleados con sus títulos académicos
 - b) Mostrar el nombre del empleado el título académico el nombre del proyecto la fecha de ejecución del proyecto
 - c) Cuantos proyectos se ejecutaron en marzo del 2023

ACTIVIDADES POR DESARROLLAR EN EL LABORATORIO:

EJERCICIO 1

Se tiene una base de datos que contiene información sobre las observaciones realizadas a distintas especies de aves. Las tablas se muestran a continuación:

AVES			
Id_ave	Id_grupo	Nom_cientifico	Nom_vulgar
001	001	harpia	águila
002	001	Morphnus	águila monera
003	002	Milvago	Caracara
004	003	Polysoma	Gavilán
005	002	Ictinia	Elanio

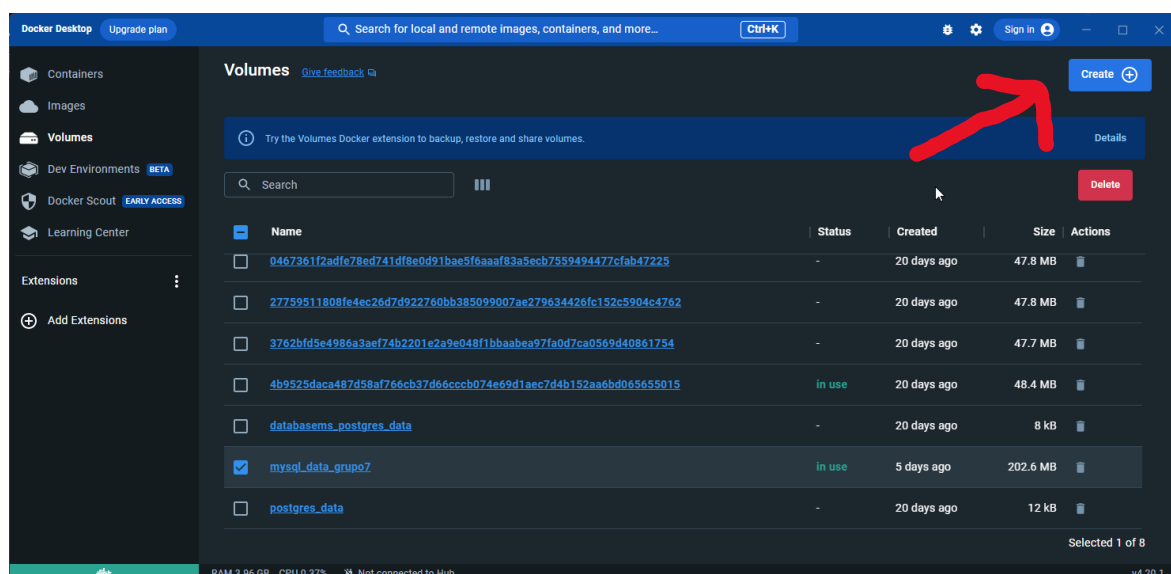
Grupos Ornitológicos			
Id_grupo	Patas	Picos	Dedos
001	Fuertes patas	Fuertes picos	Fuertes dedos
002	Medio patas	Medio picos	Medio dedos
003	Debil patas	Debil picos	Debil dedos

Observadores				
Id_observador	Id_asociacion	Estado	Nombre	Direccion
001	001	Activo	Joel	Calderón
002	001	Pasivo	Steven	Tumbaco
003	002	Activo	Erick	Magdalena
004	002	Pasivo	Marley	Pifo
005	003	Activo	Angelo	Tumbaco

1. Montar un servidor de base de datos MySQL en una máquina local.

Se procedió a montar el servidor de base de datos MySQL utilizando la herramienta de Docker. Para ello se siguieron estos pasos.

- Descargar Docker para Windows desde su sitio web '<https://www.docker.com/>', instalarlo en configuración default y crear un volumen con el nombre 'mysql_data_grupo7'.





- b) Crear un archivo docker-compose.yml con los siguientes parámetros dentro del archivo.

```
version: '24.0'

services:
  database:
    image: mysql:8.0
    environment:
      - MYSQL_DATABASE=grupo7db
      - MYSQL_USER=grupo7
      - MYSQL_PASSWORD=pass
      - MYSQL_ROOT_PASSWORD=pass
    ports:
      - '3306:3306'
    volumes:
      - mysql_data_grupo7:/var/lib/mysql

volumes:
  mysql_data_grupo7:
    external: true
```

- c) Ejecutar el siguiente comando *'docker-compose up -d database'*, en la ruta donde se encuentre el archivo, esperar a que se configure y se descargue la imagen de MySQL, en la primera instalación es necesario ejecutar este comando dos veces.

```
> docker-compose up -d database
[+] Building 0.0s (0/0)
[+] Running 1/0
✓ Container basededatos-database-1 Running
```

- d) Ejecutar el siguiente comando *'docker-compose exec database bash'*.

```
> docker-compose exec database bash
bash-4.4#
```

- e) Ahora, el usuario creado *'grupo7'* no tiene privilegios de super usuario, para otorgar esos privilegios se realizará lo siguiente:

- Ejecutar el comando `'mysql -u root -p'` y utilizar la contraseña `'pass'`, esto para acceder como usuario root en la base de datos.

```
bash-4.4# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 239
Server version: 8.0.33 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> |
```

- Ejecutar el comando `'GRANT SELECT ON *.* TO 'grupo7';'`

```
mysql> GRANT SELECT ON *.* TO 'grupo7';
Query OK, 0 rows affected (0.00 sec)
```

- Si todo salió bien al ejecutar el comando `'show grants for 'grupo7';'` se debe visualizar lo siguiente.

```
mysql> show grants for 'grupo7';
+-----+
| Grants for grupo7@% |
+-----+
|
+-----+
| GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, SHUTDOWN, PROCESS, FILE, REFERENCES, INDEX, ALTER, SHOW DATABASES, SUPER, CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION SLAVE, REPLICATION CLIENT, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER, EVENT, TRIGGER, CREATE TABLESPACE, CREATE ROLE, DROP ROLE ON *.* TO 'grupo7'@'%' |
+-----+
| GRANT APPLICATION_PASSWORD_ADMIN, AUDIT_ABORT_EXEMPT, AUDIT_ADMIN, AUTHENTICATION_POLICY_ADMIN, BACKUP_ADMIN, BINLOG_ADMIN, BINLOG_ENCRYPTION_ADMIN, CLONE_ADMIN, CONNECTION_ADMIN, ENCRYPTION_KEY_ADMIN, FIREWALL_EXEMPT, FLUSH_OPTIMIZER_COSTS, FLUSH_STATUS, FLUSH_TABLES, FLUSH_USER_RESOURCES, GROUP_REPLICATION_ADMIN, GROUP_REPLICATION_STREAM, INNODB_REDO_LOG_ARCHIVE, INNODB_REDO_LOG_ENABLE, PASSWORDLESS_USER_ADMIN, PERSIST_RO_VARIABLES_ADMIN, REPLICATION_APPLIER, REPLICATION_SLAVE_ADMIN, RESOURCE_GROUP_ADMIN, RESOURCE_GROUP_USER, ROLE_ADMIN, SENSITIVE_VARIABLES_OBSERVER, SERVICE_CONNECTION_ADMIN, SESSION_VARIABLES_ADMIN, SET_USER_ID, SHOW_ROUTINE, SYSTEM_USER, SYSTEM_VARIABLES_ADMIN, TABLE_ENCRYPTION_ADMIN, TALENTRY_LOG_ADMIN, XA_RECOVER_ADMIN ON *.* TO 'grupo7'@'%' |
+-----+
| GRANT ALL PRIVILEGES ON grupo7db.* TO 'grupo7'@'%' |
+-----+
3 rows in set (0.00 sec)
```

- Finalmente ejecutar el comando 'exit' e ingresar las credenciales del usuario grupo7.

```
mysql> exit
Bye
bash-4.4# mysql -u grupo7 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 254
Server version: 8.0.33 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

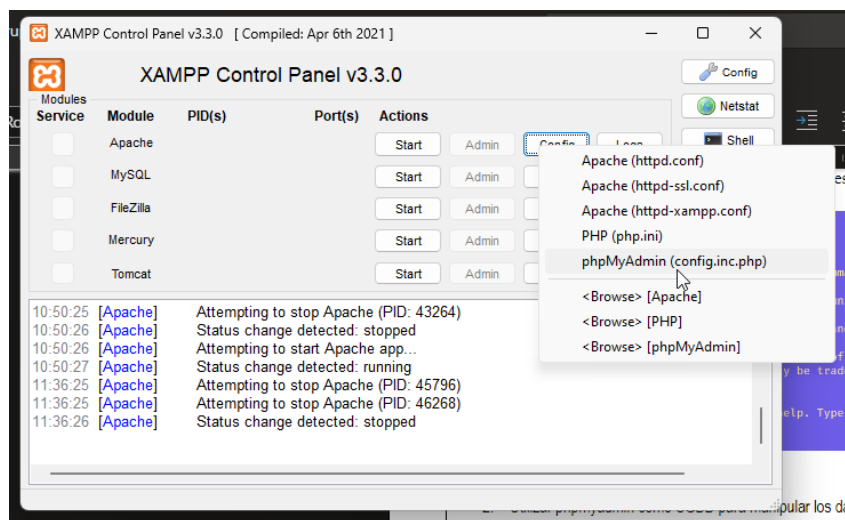
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

2. Utilizar phpmyadmin como SGBD para manipular los datos.

- a) Ahora se realiza la configuración de phpmyadmin para que reconozca al servidor MySQL levantado con Docker. Se ejecutará XAMP y se abrirá el archivo de configuración de phpmyadmin.



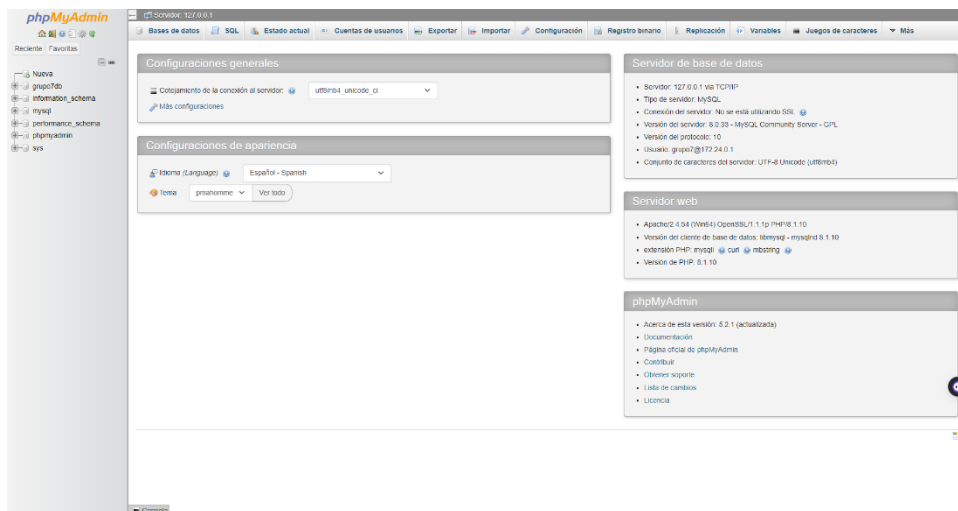
- b) En el archivo de configuración editar las siguientes entradas de tal forma que quede de la siguiente manera.

```
/* Authentication type and info */
$cfg['Servers'][$i]['auth_type'] = 'config';
$cfg['Servers'][$i]['user'] = 'grupo7';
$cfg['Servers'][$i]['password'] = 'pass';
$cfg['Servers'][$i]['extension'] = 'mysqli';
$cfg['Servers'][$i]['AllowNoPassword'] = true;
$cfg['Lang'] = '';

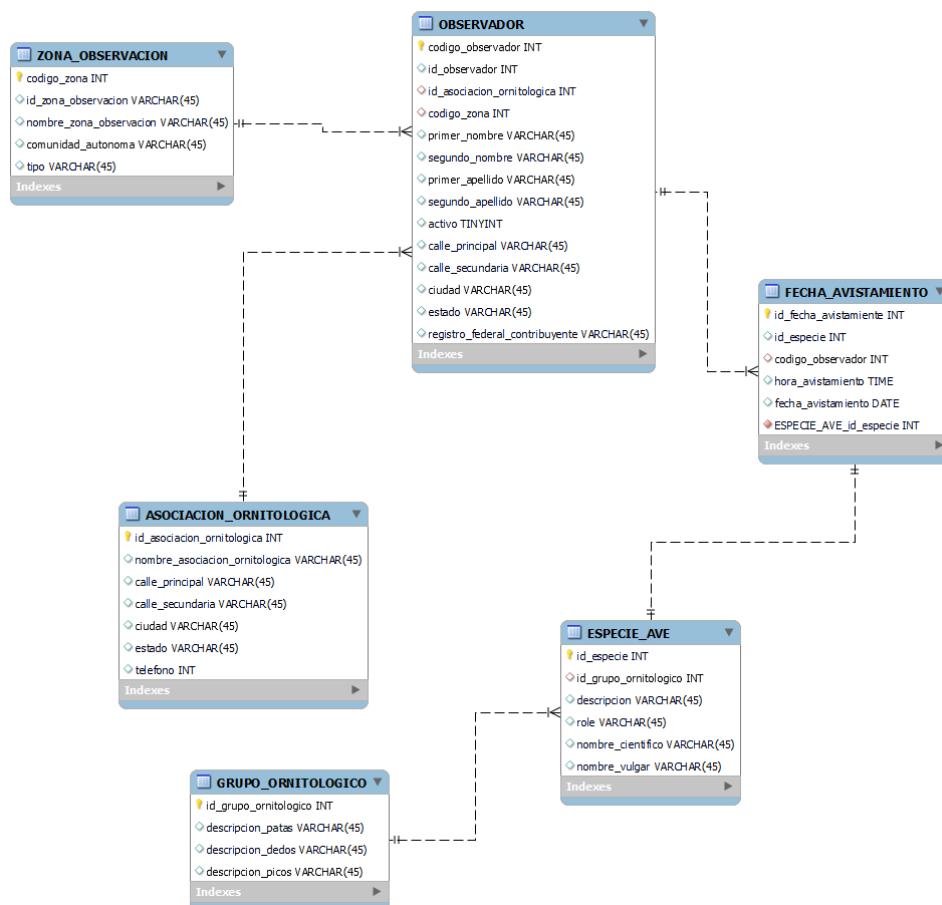
/* Bind to the localhost ipv4 address and tcp */
$cfg['Servers'][$i]['host'] = '127.0.0.1';
$cfg['Servers'][$i]['connect_type'] = 'tcp';

/* User for advanced features */
$cfg['Servers'][$i]['controluser'] = 'root';
$cfg['Servers'][$i]['controlpass'] = 'pass';
```


- c) Finalmente ejecutar únicamente Apache de XAMP (ya no es necesario ejecutar MySQL de XAMP) y abrir phpmyadmin



3. Utilizar MySQLWorkbench para elaborar el modelo lógico de la base de datos (Incluir las tablas que faltan en el gráfico y que se desarrolló en el proyecto de la Unidad 1) y generar el script en SQL del modelo conceptual.





```
-- MySQL Script generated by MySQL Workbench
-- Sat Jul 8 12:15:03 2023
-- Model: New Model Version: 1.0
-- 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 mydb
-- -----

-- -----
-- Schema mydb
-- -----

CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
USE `mydb` ;

-- -----
-- Table `mydb`.`ZONA_OBSERVACION`
-- -----
CREATE TABLE IF NOT EXISTS `mydb`.`ZONA_OBSERVACION` (
  `codigo_zona` INT NOT NULL,
  `id_zona_observacion` VARCHAR(45) NULL,
  `nombre_zona_observacion` VARCHAR(45) NULL,
  `comunidad_autonoma` VARCHAR(45) NULL,
  `tipo` VARCHAR(45) NULL,
  PRIMARY KEY (`codigo_zona`))
ENGINE = InnoDB;

-- -----
-- Table `mydb`.`GRUPO_ORNITOLOGICO`
-- -----
CREATE TABLE IF NOT EXISTS `mydb`.`GRUPO_ORNITOLOGICO` (
  `id_grupo_ornitologico` INT NOT NULL,
  `descripcion_patas` VARCHAR(45) NULL,
  `descripcion_dedos` VARCHAR(45) NULL,
  `descripcion_picos` VARCHAR(45) NULL,
  PRIMARY KEY (`id_grupo_ornitologico`))
ENGINE = InnoDB;

-- -----
-- Table `mydb`.`ESPECIE_AVE`
-- -----
CREATE TABLE IF NOT EXISTS `mydb`.`ESPECIE_AVE` (
  `id_especie` INT NOT NULL,
  `id_grupo_ornitologico` INT NULL,
  `descripcion` VARCHAR(45) NULL,
  `role` VARCHAR(45) NULL,
  `nombre_cientifico` VARCHAR(45) NULL,
  `nombre_vulgar` VARCHAR(45) NULL,
  PRIMARY KEY (`id_especie`),
```



```
INDEX `fk_ESPECIE_AVE_GRUPO_ORNITOLOGICO1_idx` (`id_grupo_ornitologico` ASC) VISIBLE,  
CONSTRAINT `fk_ESPECIE_AVE_GRUPO_ORNITOLOGICO1`  
    FOREIGN KEY (`id_grupo_ornitologico`)  
    REFERENCES `mydb`.`GRUPO_ORNITOLOGICO` (`id_grupo_ornitologico`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;  
  
-----  
-- Table `mydb`.`ASOCIACION_ORNITOLOGICA`  
-----  
CREATE TABLE IF NOT EXISTS `mydb`.`ASOCIACION_ORNITOLOGICA` (  
    `id_asociacion_ornitologica` INT NOT NULL,  
    `nombre_asociacion_ornitologica` VARCHAR(45) NULL,  
    `calle_principal` VARCHAR(45) NULL,  
    `calle_secundaria` VARCHAR(45) NULL,  
    `ciudad` VARCHAR(45) NULL,  
    `estado` VARCHAR(45) NULL,  
    `telefono` INT NULL,  
    PRIMARY KEY (`id_asociacion_ornitologica`),  
    UNIQUE INDEX `telefono_UNIQUE` (`telefono` ASC) VISIBLE)  
ENGINE = InnoDB;  
  
-----  
-- Table `mydb`.`OBSERVADOR`  
-----  
CREATE TABLE IF NOT EXISTS `mydb`.`OBSERVADOR` (  
    `codigo_observador` INT NOT NULL,  
    `id_observador` INT NULL,  
    `id_asociacion_ornitologica` INT NULL,  
    `codigo_zona` INT NULL,  
    `primer_nombre` VARCHAR(45) NULL,  
    `segundo_nombre` VARCHAR(45) NULL,  
    `primer_apellido` VARCHAR(45) NULL,  
    `segundo_apellido` VARCHAR(45) NULL,  
    `activo` TINYINT NULL,  
    `calle_principal` VARCHAR(45) NULL,  
    `calle_secundaria` VARCHAR(45) NULL,  
    `ciudad` VARCHAR(45) NULL,  
    `estado` VARCHAR(45) NULL,  
    `registro_federal_contribuyente` VARCHAR(45) NULL,  
    PRIMARY KEY (`codigo_observador`),  
    INDEX `fk_OBSERVADOR_ZONA_OBSERVACION1_idx` (`codigo_zona` ASC) VISIBLE,  
    INDEX `fk_OBSERVADOR_ASOCIACION_ORNITOLOGICA1_idx` (`id_asociacion_ornitologica` ASC)  
    VISIBLE,  
    CONSTRAINT `fk_OBSERVADOR_ZONA_OBSERVACION1`  
        FOREIGN KEY (`codigo_zona`)  
        REFERENCES `mydb`.`ZONA_OBSERVACION` (`codigo_zona`)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION,  
    CONSTRAINT `fk_OBSERVADOR_ASOCIACION_ORNITOLOGICA1`  
        FOREIGN KEY (`id_asociacion_ornitologica`)  
        REFERENCES `mydb`.`ASOCIACION_ORNITOLOGICA` (`id_asociacion_ornitologica`)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```



```
-- Table `mydb`.`FECHA_AVISTAMIENTO`  
-----  
CREATE TABLE IF NOT EXISTS `mydb`.`FECHA_AVISTAMIENTO` (  
  `id_fecha_avistamiento` INT NOT NULL,  
  `id_especie` INT NULL,  
  `codigo_observador` INT NULL,  
  `hora_avistamiento` TIME NULL,  
  `fecha_avistamiento` DATE NULL,  
  `ESPECIE_AVE_id_especie` INT NOT NULL,  
  PRIMARY KEY (`id_fecha_avistamiento`),  
  INDEX `fk_FECHA_AVISTAMIENTO_OBSERVADOR1_idx` (`codigo_observador` ASC) VISIBLE,  
  INDEX `fk_FECHA_AVISTAMIENTO_ESPECIE_AVE1_idx` (`ESPECIE_AVE_id_especie` ASC) VISIBLE,  
  CONSTRAINT `fk_FECHA_AVISTAMIENTO_OBSERVADOR1`  
    FOREIGN KEY (`codigo_observador`)  
    REFERENCES `mydb`.`OBSERVADOR` (`codigo_observador`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_FECHA_AVISTAMIENTO_ESPECIE_AVE1`  
    FOREIGN KEY (`ESPECIE_AVE_id_especie`)  
    REFERENCES `mydb`.`ESPECIE_AVE` (`id_especie`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- Table `mydb`.`ZONA_OBSERVACION`  
-----  
CREATE TABLE IF NOT EXISTS `mydb`.`ZONA_OBSERVACION` (  
  `codigo_zona` INT NOT NULL,  
  `id_zona_observacion` VARCHAR(45) NULL,  
  `nombre_zona_observacion` VARCHAR(45) NULL,  
  `comunidad_autonoma` VARCHAR(45) NULL,  
  `tipo` VARCHAR(45) NULL,  
  PRIMARY KEY (`codigo_zona`))  
ENGINE = InnoDB;
```

```
-- Table `mydb`.`ASOCIACION_ORNITOLOGICA`  
-----  
CREATE TABLE IF NOT EXISTS `mydb`.`ASOCIACION_ORNITOLOGICA` (  
  `id_asociacion_ornitologica` INT NOT NULL,  
  `nombre_asociacion_ornitologica` VARCHAR(45) NULL,  
  `calle_principal` VARCHAR(45) NULL,  
  `calle_secundaria` VARCHAR(45) NULL,  
  `ciudad` VARCHAR(45) NULL,  
  `estado` VARCHAR(45) NULL,  
  `telefono` INT NULL,  
  PRIMARY KEY (`id_asociacion_ornitologica`),  
  UNIQUE INDEX `telefono_UNIQUE` (`telefono` ASC) VISIBLE)  
ENGINE = InnoDB;
```



-- Table `mydb`.`OBSERVADOR`

```
CREATE TABLE IF NOT EXISTS `mydb`.`OBSERVADOR` (  
  `codigo_observador` INT NOT NULL,  
  `id_observador` INT NULL,  
  `id_asociacion_ornitologica` INT NULL,  
  `codigo_zona` INT NULL,  
  `primer_nombre` VARCHAR(45) NULL,  
  `segundo_nombre` VARCHAR(45) NULL,  
  `primer_apellido` VARCHAR(45) NULL,  
  `segundo_apellido` VARCHAR(45) NULL,  
  `activo` TINYINT NULL,  
  `calle_principal` VARCHAR(45) NULL,  
  `calle_secundaria` VARCHAR(45) NULL,  
  `ciudad` VARCHAR(45) NULL,  
  `estado` VARCHAR(45) NULL,  
  `registro_federal_contribuyente` VARCHAR(45) NULL,  
  PRIMARY KEY (`codigo_observador`),  
  INDEX `fk_OBSERVADOR_ZONA_OBSERVACION1_idx` (`codigo_zona` ASC) VISIBLE,  
  INDEX `fk_OBSERVADOR_ASOCIACION_ORNITOLOGICA1_idx` (`id_asociacion_ornitologica` ASC)  
  VISIBLE,  
  CONSTRAINT `fk_OBSERVADOR_ZONA_OBSERVACION1`  
    FOREIGN KEY (`codigo_zona`)  
    REFERENCES `mydb`.`ZONA_OBSERVACION` (`codigo_zona`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_OBSERVADOR_ASOCIACION_ORNITOLOGICA1`  
    FOREIGN KEY (`id_asociacion_ornitologica`)  
    REFERENCES `mydb`.`ASOCIACION_ORNITOLOGICA` (`id_asociacion_ornitologica`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-- Table `mydb`.`GRUPO_ORNITOLOGICO`

```
CREATE TABLE IF NOT EXISTS `mydb`.`GRUPO_ORNITOLOGICO` (  
  `id_grupo_ornitologico` INT NOT NULL,  
  `descripcion_patas` VARCHAR(45) NULL,  
  `descripcion_dedos` VARCHAR(45) NULL,  
  `descripcion_picos` VARCHAR(45) NULL,  
  PRIMARY KEY (`id_grupo_ornitologico`))  
ENGINE = InnoDB;
```

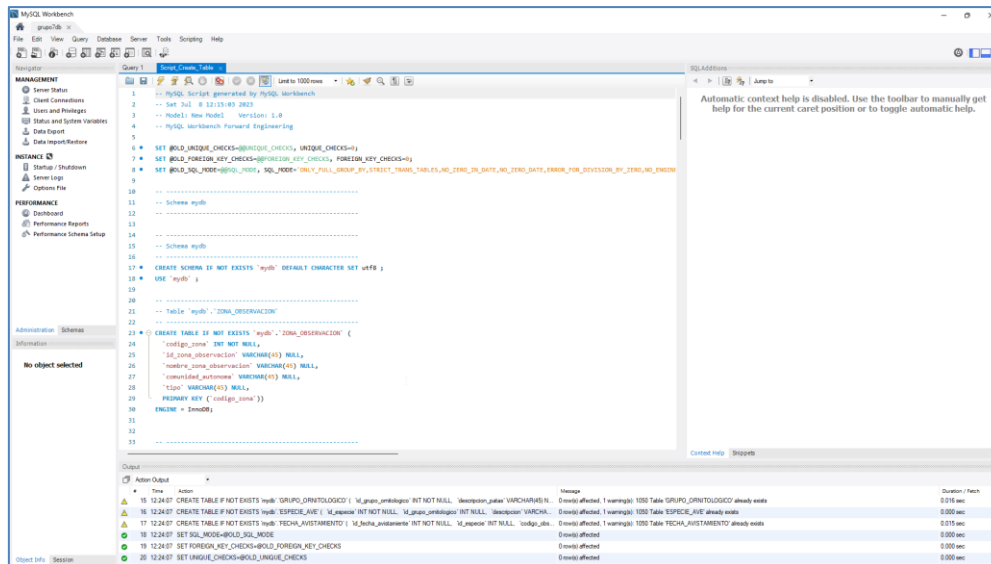
-- Table `mydb`.`ESPECIE_AVE`

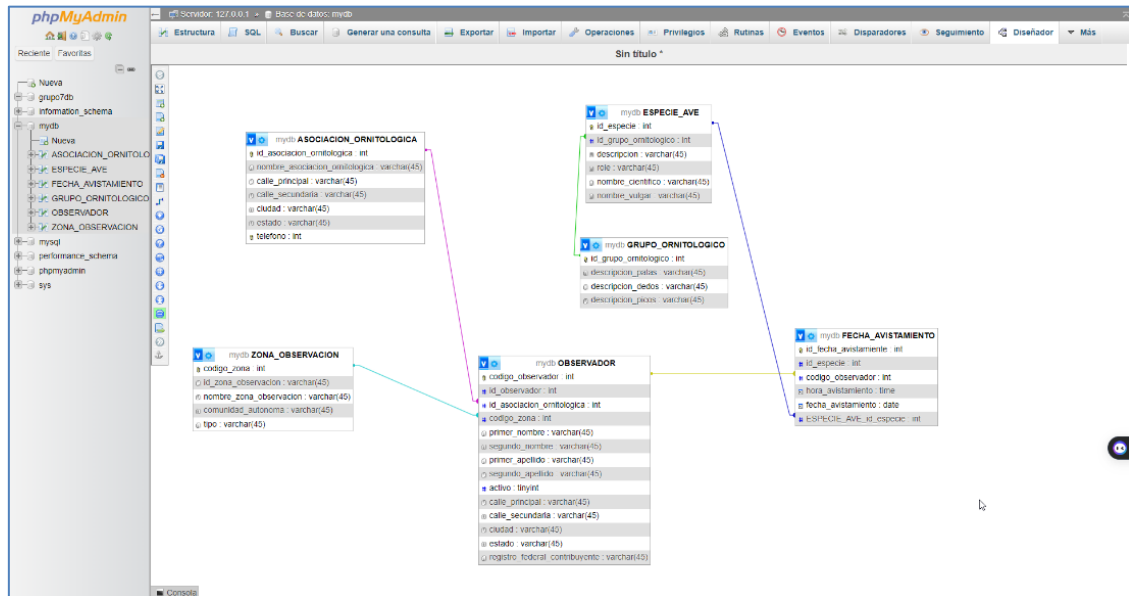
```
CREATE TABLE IF NOT EXISTS `mydb`.`ESPECIE_AVE` (  
  `id_especie` INT NOT NULL,  
  `id_grupo_ornitologico` INT NULL,  
  `descripcion` VARCHAR(45) NULL,  
  `role` VARCHAR(45) NULL,  
  `nombre_cientifico` VARCHAR(45) NULL,  
  `nombre_vulgar` VARCHAR(45) NULL,  
  PRIMARY KEY (`id_especie`),  
  INDEX `fk_ESPECIE_AVE_GRUPO_ORNITOLOGICO1_idx` (`id_grupo_ornitologico` ASC) VISIBLE,
```



```
CONSTRAINT `fk_ESPECIE_AVE_GRUPO_ORNITOLOGICO1`  
  FOREIGN KEY (`id_grupo_ornitologico`)  
  REFERENCES `mydb`.`GRUPO_ORNITOLOGICO` (`id_grupo_ornitologico`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)  
ENGINE = InnoDB;  
  
-----  
-- Table `mydb`.`FECHA_AVISTAMIENTO`  
-----  
CREATE TABLE IF NOT EXISTS `mydb`.`FECHA_AVISTAMIENTO` (  
  `id_fecha_avistamiento` INT NOT NULL,  
  `id_especie` INT NULL,  
  `codigo_observador` INT NULL,  
  `hora_avistamiento` TIME NULL,  
  `fecha_avistamiento` DATE NULL,  
  `ESPECIE_AVE_id_especie` INT NOT NULL,  
  PRIMARY KEY (`id_fecha_avistamiento`),  
  INDEX `fk_FECHA_AVISTAMIENTO_OBSERVADOR1_idx` (`codigo_observador` ASC) VISIBLE,  
  INDEX `fk_FECHA_AVISTAMIENTO_ESPECIE_AVE1_idx` (`ESPECIE_AVE_id_especie` ASC) VISIBLE,  
  CONSTRAINT `fk_FECHA_AVISTAMIENTO_OBSERVADOR1`  
    FOREIGN KEY (`codigo_observador`)  
    REFERENCES `mydb`.`OBSERVADOR` (`codigo_observador`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_FECHA_AVISTAMIENTO_ESPECIE_AVE1`  
    FOREIGN KEY (`ESPECIE_AVE_id_especie`)  
    REFERENCES `mydb`.`ESPECIE_AVE` (`id_especie`)  
    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;
```

4. Importar el script generado a phpmyadmin y obtener el modelo físico.





5. Ingresar registros para cada una de las tablas.

Tabla “asociación_ornitologica”










		id_asociacion_ornitologica	nombre_asociacion_ornitologica	calle_principal	calle_secundaria	ciudad	estado	telefono
<input type="checkbox"/>	 Editar  Copiar  Borrar	1	Asociación de Observadores de Aves	Av. de las Aves	Calle de los Pájaros	Quito	Pichincha	912345678
<input type="checkbox"/>	 Editar  Copiar  Borrar	2	Observadores de Aves del Guayas	Calle de los Colibríes	Avenida de las Especies	Guayaquil	Guayas	987654321
<input type="checkbox"/>	 Editar  Copiar  Borrar	3	Los Pajarracos	Av. de los colibres	Av. de los rapazes	Quito	Pichincha	981473030

Tabla “especie_ave”

			id_especie	id_grupo_ornitologico	descripcion	role	nombre_cientifico	nombre_vulgar
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	1	1 Grande	Carroñero	harpia	águila
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	2	1 Pequeña	Depredador	Morphnus	águila monera
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	3	2 Mediana	Depredador	Milvago	Caracara
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	4	3 Grande	Carroñero	Polysoma	Gavilán
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	5	2 Pequeña	Rapaz	Ictinia	Elanio

Tabla “fecha_avistamiento”




		id_fecha_avistamiento	id_especie	codigo_observador	hora_avistamiento	fecha_avistamiento	ESPECIE_AVE_id_especie
<input type="checkbox"/>	 Editar  Copiar  Borrar	1	1	4	15:08:48	2023-07-05	1
<input type="checkbox"/>	 Editar  Copiar  Borrar	2	2	5	15:08:48	2023-07-08	2

Tabla “grupo_ornitologico”

<div>← T →</div>		<div>▼ id_grupo_ornitologico</div>	descripcion_patas	descripcion_dedos	descripcion_picos
<input type="checkbox"/>	<div>✎ Editar</div>	<div>📄 Copiar</div>	<div>🗑 Borrar</div>	1 Fuertes patas	Fuertes dedos Fuertes picos
<input type="checkbox"/>	<div>✎ Editar</div>	<div>📄 Copiar</div>	<div>🗑 Borrar</div>	2 Medio patas	Medio dedos Medio Picos
<input type="checkbox"/>	<div>✎ Editar</div>	<div>📄 Copiar</div>	<div>🗑 Borrar</div>	3 Débil patas	Débil dedos Débil picos

Tabla “observador”

	codigo_observador	id_observador	id_asociacion_ornitologica	codigo_zona	primer_nombre	segundo_nombre	primer_apellido	segundo_apellido	activo	calle_principal	calle_secundaria	ciudad
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar		1	1	1	Joel	Carlos	Perez	Gomez	1	Calderón	Avenida de las Aves	Quito
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar		2	2	1	Steven	Fernando	Rodríguez	García	0	Tumbaco	Avenida de los Pájaros	Quito
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar		3	3	2	Erick	Andres	López	Martínez	1	Magdalena	Avenida de los Colibríes	Quito
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar		4	4	2	Marley	Gabriela	Castro	Hernández	0	Pifo	Calle de los Pájaros	Quito
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar		5	5	3	Angelo	Antonio	Ruiz	Sánchez	1	Tumbaco	Calle de las Aves	Quito

Tabla “zona_observacion”

	codigo_zona	id_zona_observacion	nombre_zona_observacion	comunidad_autonoma	tipo
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	1	1	Islas Galápagos	Galápagos	Marino
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	2	2	Mindo Cloud Forest	Pichincha	Bosque humedo
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	3	3	Reserva Ecológica Antisana	Napo y Pichincha	Montaña
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	4	4	Parque Nacional Yasuni	Orellana y Pastaza	Bosque humedo
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	5	5	Reserva de Producción Faunística Cuyabeno	Sucumbíos	Manglar

6. Realizar las siguientes consultas SQL a la base de datos, utilizando phpmyadmin.

- a. Obtener el nombre científico de las aves que pertenecen al grupo ornitológico 001.

✓ Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0008 segundos.)

```
SELECT nombre_cientifico FROM ESPECIE_AVE WHERE id_grupo_ornitologico = 1;
```

☐ Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ninguna

Opciones extra

	nombre_cientifico
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	harpia
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	Morphnus

- b. Obtener los nombres de los observadores que pertenecen a la asociación 002.

✓ Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0003 segundos.)

```
SELECT primer_nombre, segundo_nombre, primer_apellido, segundo_apellido FROM observador WHERE id_asociacion_ornitologica=002;
```

☐ Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ninguna

Opciones extra

	primer_nombre	segundo_nombre	primer_apellido	segundo_apellido
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	Erick	Andres	López	Martínez
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	Marley	Gabriela	Castro	Hernández

c. Obtener el nombre científico de las aves.

✓ Mostrando filas 0 - 4 (total de 5, La consulta tardó 0,0003 segundos.)

```
SELECT nombre_cientifico FROM ESPECIE_AVE;
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Ordenar según la clave: Ninguna

Opciones extra

	nombre_cientifico
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	harpia
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	Morphnus
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	Milvago
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	Polysoma
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	Ictinia

☐ Seleccionar todo Para los elementos que están marcados: ☐ Editar ☐ Copiar ☐ Borrar ☐ Exportar

d. Obtener el nombre y la dirección de los observadores que están en estado pasivo.

✓ Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0003 segundos.)

```
SELECT primer_nombre, calle_principal FROM OBSERVADOR WHERE activo=0;
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Ordenar según la clave: Ninguna

Opciones extra

	primer_nombre	calle_principal
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	Steven	Tumbaco
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	Marley	Pifo

☐ Seleccionar todo Para los elementos que están marcados: ☐ Editar ☐ Copiar ☐ Borrar ☐ Exportar

e. Obtener los datos del águila.

✓ Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0003 segundos.)

```
SELECT id_especie, id_grupo_ornitologico, descripcion, role, nombre_cientifico, nombre_vulgar FROM ESPECIE_AVE WHERE nombre_vulgar = 'águila';
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas:

Opciones extra

	id_especie	id_grupo_ornitologico	descripcion	role	nombre_cientifico	nombre_vulgar
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	1	1	Grande	Carroñero	harpia	águila

☐ Seleccionar todo Para los elementos que están marcados: ☐ Editar ☐ Copiar ☐ Borrar ☐ Exportar

☐ Mostrar todo | Número de filas: 25 | Filtrar filas:

f. Obtener el nombre científico del gavián.

✓ Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0003 segundos.)

```
SELECT nombre_cientifico FROM ESPECIE_AVE WHERE nombre_vulgar = 'Gavián';
```

☐ Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla

Opciones extra

← T → nombre_cientifico

☐ Editar ☐ Copiar ☐ Borrar Polysoma

↑ ☐ Seleccionar todo Para los elementos que están marcados: ☐ Editar ☐ Copiar ☐ Borrar ☐ Exportar

g. Obtener estado de Steven.

```
SELECT estado FROM observador WHERE primer_nombre="Steven";
```

☐ Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla

Opciones extra

← T → estado

☐ Editar ☐ Copiar ☐ Borrar Pichincha

h. Dirección de los observadores que pertenecen a la asociación 002.

```
SELECT calle_principal FROM observador WHERE id_asociacion_ornitologica=002;
```

☐ Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla

Opciones extra

← T → calle_principal

☐ Editar ☐ Copiar ☐ Borrar Magdalena

☐ Editar ☐ Copiar ☐ Borrar Pifo

i. Obtener el grupo ornitológico de ictinia.

```
SELECT id_grupo_ornitologico FROM especie_ave WHERE nombre_cientifico="Ictinia";
```

☐ Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla

Opciones extra

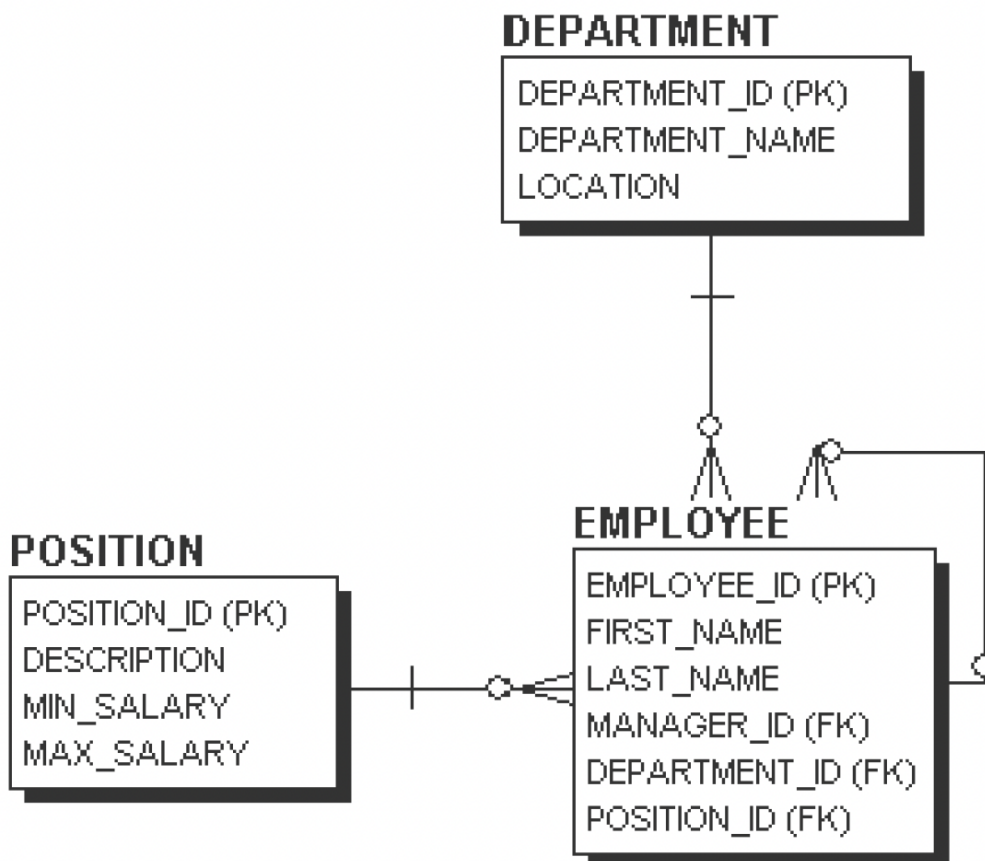
← T → id_grupo_ornitologico

☐ Editar ☐ Copiar ☐ Borrar 2

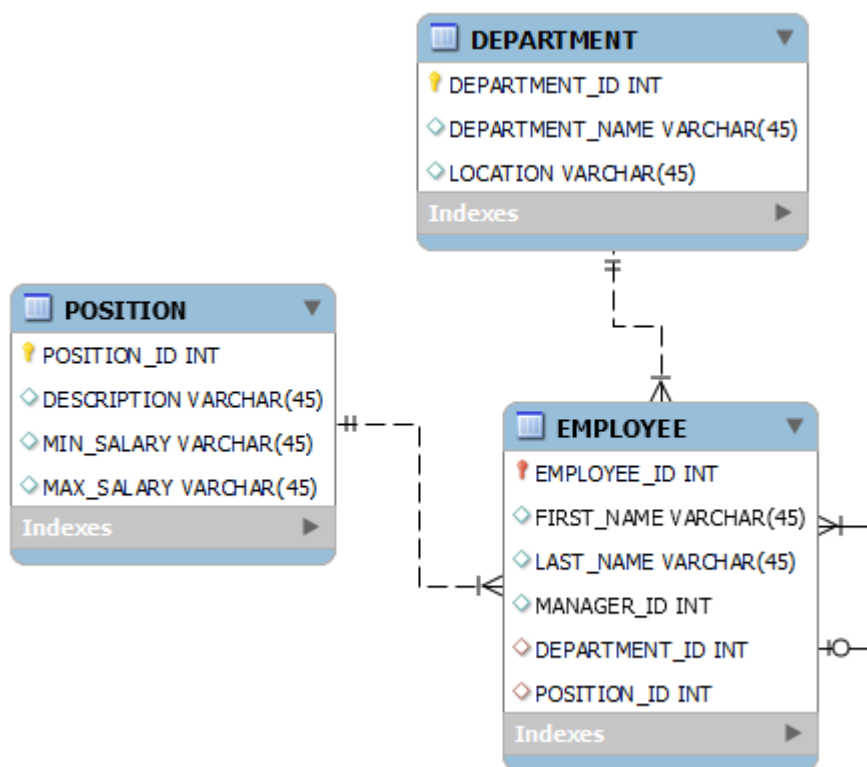


EJERCICIO 2

Se tiene una base de datos que contiene información sobre los empleados y departamentos de una empresa. Las tablas se muestran a continuación:



1. Utilizar MySQL Workbench para elaborar el modelo lógico de la base de datos y generar el script SQL.



Código:

MySQL Script generated by MySQL Workbench

-- Sat Jul 8 15:44:50 2023

-- Model: New Model Version: 1.0

-- 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,ERR
OR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
  
```

```

-- Schema empresadb
  
```

```

-- Schema empresadb
  
```

```

CREATE SCHEMA IF NOT EXISTS `empresadb` DEFAULT CHARACTER SET utf8 ;
USE `empresadb` ;
  
```

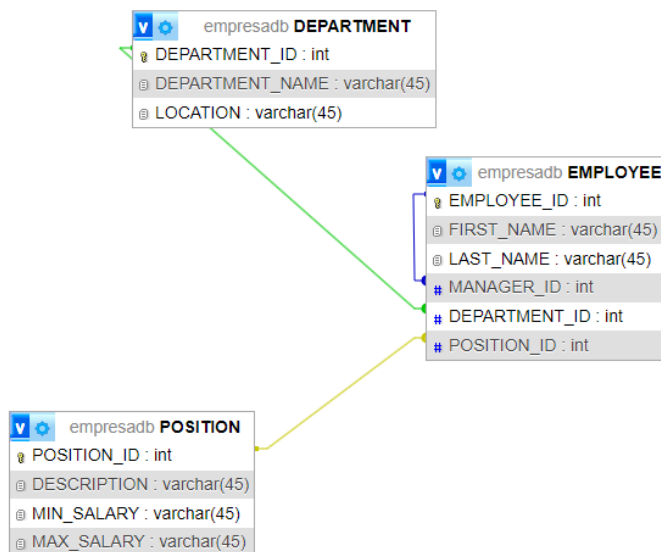



```
-----  
-- Table `empresadb`.`DEPARTMENT`  
-----  
CREATE TABLE IF NOT EXISTS `empresadb`.`DEPARTMENT` (  
  `DEPARTMENT_ID` INT NOT NULL,  
  `DEPARTMENT_NAME` VARCHAR(45) NULL,  
  `LOCATION` VARCHAR(45) NULL,  
  PRIMARY KEY (`DEPARTMENT_ID`))  
ENGINE = InnoDB;  
  
-----  
-- Table `empresadb`.`POSITION`  
-----  
CREATE TABLE IF NOT EXISTS `empresadb`.`POSITION` (  
  `POSITION_ID` INT NOT NULL,  
  `DESCRIPTION` VARCHAR(45) NULL,  
  `MIN_SALARY` VARCHAR(45) NULL,  
  `MAX_SALARY` VARCHAR(45) NULL,  
  PRIMARY KEY (`POSITION_ID`))  
ENGINE = InnoDB;  
  
-----  
-- Table `empresadb`.`EMPLOYEE`  
-----  
CREATE TABLE IF NOT EXISTS `empresadb`.`EMPLOYEE` (  
  `EMPLOYEE_ID` INT NOT NULL,  
  `FIRST_NAME` VARCHAR(45) NULL,  
  `LAST_NAME` VARCHAR(45) NULL,  
  `MANAGER_ID` INT NULL,  
  `DEPARTMENT_ID` INT NULL,  
  `POSITION_ID` INT NULL,  
  PRIMARY KEY (`EMPLOYEE_ID`),  
  INDEX `fk_EMPLOYEE_POSITION1_idx` (`POSITION_ID` ASC) VISIBLE,  
  INDEX `fk_EMPLOYEE_DEPARTMENT1_idx` (`DEPARTMENT_ID` ASC) VISIBLE,  
  CONSTRAINT `fk_EMPLOYEE_POSITION1`  
    FOREIGN KEY (`POSITION_ID`)  
    REFERENCES `empresadb`.`POSITION` (`POSITION_ID`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_EMPLOYEE_EMPLOYEE1`  
    FOREIGN KEY (`MANAGER_ID`)  
    REFERENCES `empresadb`.`EMPLOYEE` (`EMPLOYEE_ID`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_EMPLOYEE_DEPARTMENT1`  
    FOREIGN KEY (`DEPARTMENT_ID`)  
    REFERENCES `empresadb`.`DEPARTMENT` (`DEPARTMENT_ID`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)
```

```
ENGINE = InnoDB;
```

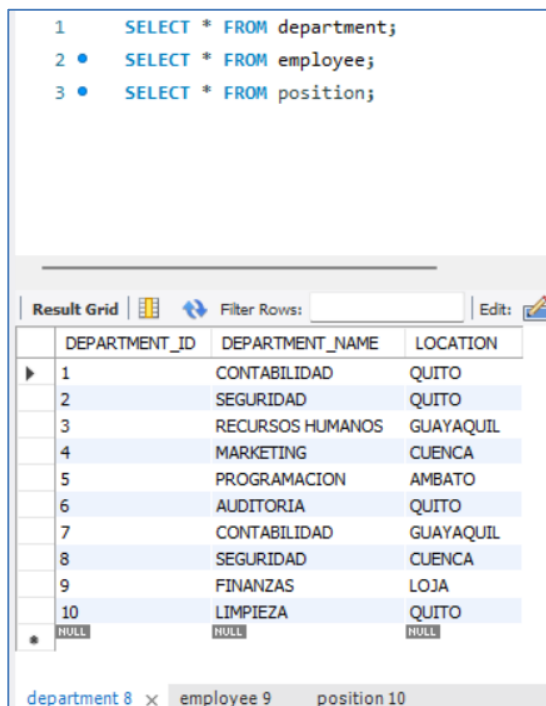
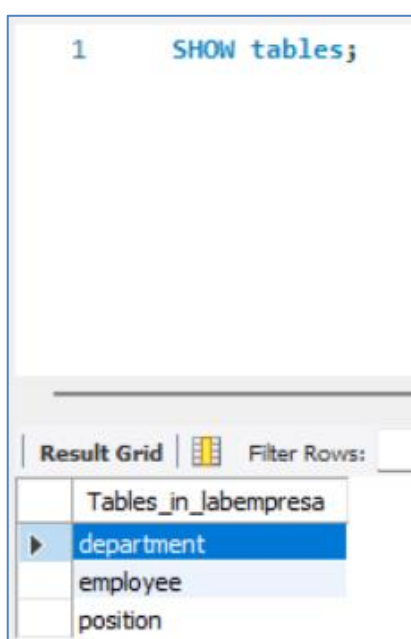
```
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

2. Importar el script generado a phpmyadmin y obtener el modelo físico.



3. Realizar las siguientes consultas SQL a la base de datos, utilizando MySQLWorkbench (cliente de base de datos).

a) Listar todos los datos de las tablas creadas.





Result Grid | Filter Rows: | Edit: | Export/Import:

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	MANAGER_ID	DEPARTMENT_ID	POSITION_ID
▶	40	MARTIN	MEDINA	40	1	26
	41	JOSE	BANDERITAS	40	2	29
	42	ANGELO	YANACALLO	42	3	22
	43	RICARDO	RIVADENEIRA	43	5	25
	44	CAMILO	FERRIN	43	6	24
	45	MARCELO	ALBAN	40	7	27
	46	MATIAS	GOMEZ	42	8	23
	47	JUAN	PONCE	41	9	20
	48	DAYANA	LASCANO	41	10	21
*	NULL	NULL	NULL	NULL	NULL	NULL

department 8 employee 9 × position 10

	POSITION_ID	DESCRIPTION	MIN_SALARY	MAX_SALARY
▶	20	GERENTE	444	750
	21	APOYO	430	694
	22	PASANTE	250	400
	23	GUARDIA	467	721
	24	LICENCIADO	476	803
	25	INGENIERO SOFTWARE	497	923
	26	ECONOMISTA	490	867
	27	CONTADOR	365	763
	28	LICENCIADO	376	719
	29	SECRETARIO	366	710
*	NULL	NULL	NULL	NULL

department 8 employee 9 position 10 ×

b) Obtener un reporte de los empleados por orden alfabético.

Reporte Ordenado Alfabéticamente por la inicial de su nombre.

1 • `SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME FROM employee ORDER BY FIRST_NAME;`

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell C

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME
▶	42	ANGELO	YANACALLO
	44	CAMILO	FERRIN
	48	DAYANA	LASCANO
	41	JOSE	BANDERITAS
	47	JUAN	PONCE
	45	MARCELO	ALBAN
	40	MARTIN	MEDINA
	46	MATIAS	GOMEZ
	43	RICARDO	RIVADENEIRA
*	NULL	NULL	NULL

1 • `SELECT * FROM employee ORDER BY FIRST_NAME;`

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	MANAGER_ID	DEPARTMENT_ID	POSITION_ID
42	ANGELO	YANACALLO	42	3	22
44	CAMILO	FERRIN	43	6	24
48	DAYANA	LASCANO	41	10	21
41	JOSE	BANDERITAS	40	2	29
47	JUAN	PONCE	41	9	20
45	MARCELO	ALBAN	40	7	27
40	MARTIN	MEDINA	40	1	26
46	MATIAS	GOMEZ	42	8	23
43	RICARDO	RIVADENEIRA	43	5	25
NULL	NULL	NULL	NULL	NULL	NULL

Reporte Ordenado Alfabéticamente por la inicial de su apellido.

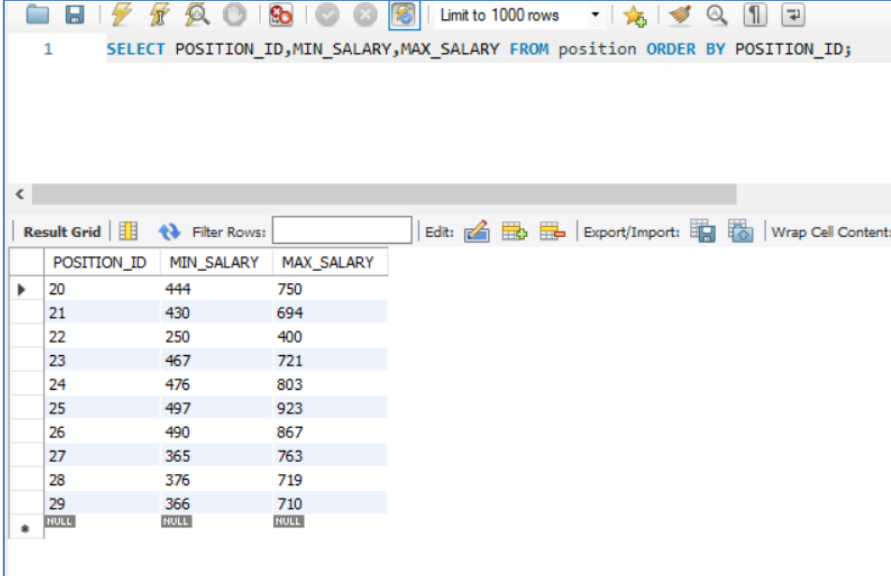
1 • `SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME FROM employee ORDER BY LAST_NAME;`

EMPLOYEE_ID	FIRST_NAME	LAST_NAME
45	MARCELO	ALBAN
41	JOSE	BANDERITAS
44	CAMILO	FERRIN
46	MATIAS	GOMEZ
48	DAYANA	LASCANO
40	MARTIN	MEDINA
47	JUAN	PONCE
43	RICARDO	RIVADENEIRA
42	ANGELO	YANACALLO
NULL	NULL	NULL

1 • `SELECT * FROM employee ORDER BY LAST_NAME;`

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	MANAGER_ID	DEPARTMENT_ID	POSITION_ID
45	MARCELO	ALBAN	40	7	27
41	JOSE	BANDERITAS	40	2	29
44	CAMILO	FERRIN	43	6	24
46	MATIAS	GOMEZ	42	8	23
48	DAYANA	LASCANO	41	10	21
40	MARTIN	MEDINA	40	1	26
47	JUAN	PONCE	41	9	20
43	RICARDO	RIVADENEIRA	43	5	25
42	ANGELO	YANACALLO	42	3	22
NULL	NULL	NULL	NULL	NULL	NULL

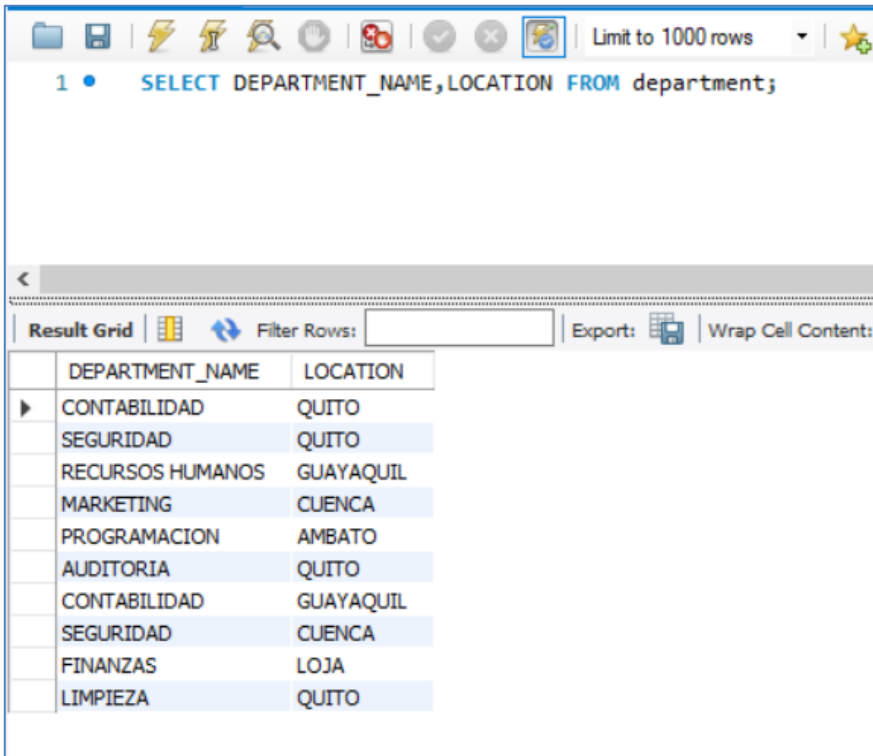
- c) Listar los valores mínimos y máximos en salario que pueden percibir los empleados por código de posición.



SQL Query: `SELECT POSITION_ID, MIN_SALARY, MAX_SALARY FROM position ORDER BY POSITION_ID;`

POSITION_ID	MIN_SALARY	MAX_SALARY
20	444	750
21	430	694
22	250	400
23	467	721
24	476	803
25	497	923
26	490	867
27	365	763
28	376	719
29	366	710
NULL	NULL	NULL

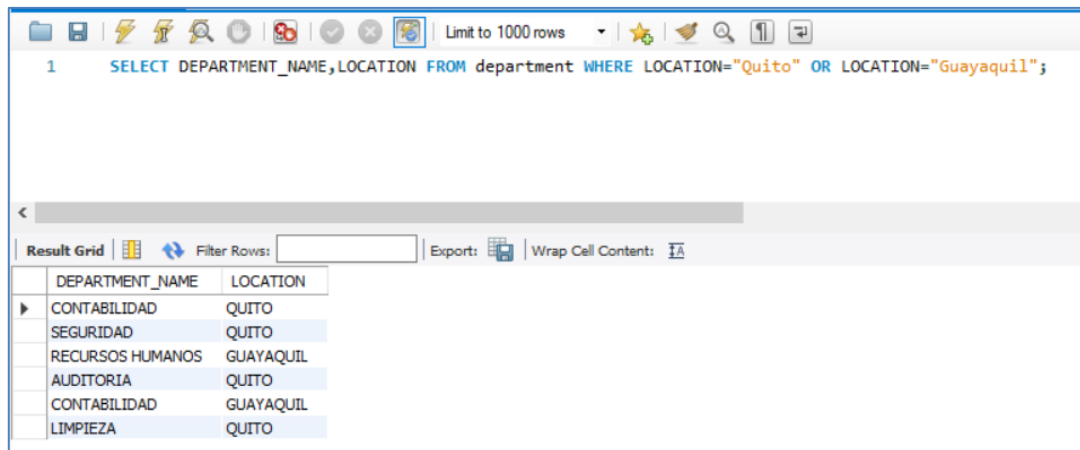
- d) Listar los nombres de los departamentos y su localización.



SQL Query: `SELECT DEPARTMENT_NAME, LOCATION FROM department;`

DEPARTMENT_NAME	LOCATION
CONTABILIDAD	QUITO
SEGURIDAD	QUITO
RECURSOS HUMANOS	GUAYAQUIL
MARKETING	CUENCA
PROGRAMACION	AMBATO
AUDITORIA	QUITO
CONTABILIDAD	GUAYAQUIL
SEGURIDAD	CUENCA
FINANZAS	LOJA
LIMPIEZA	QUITO

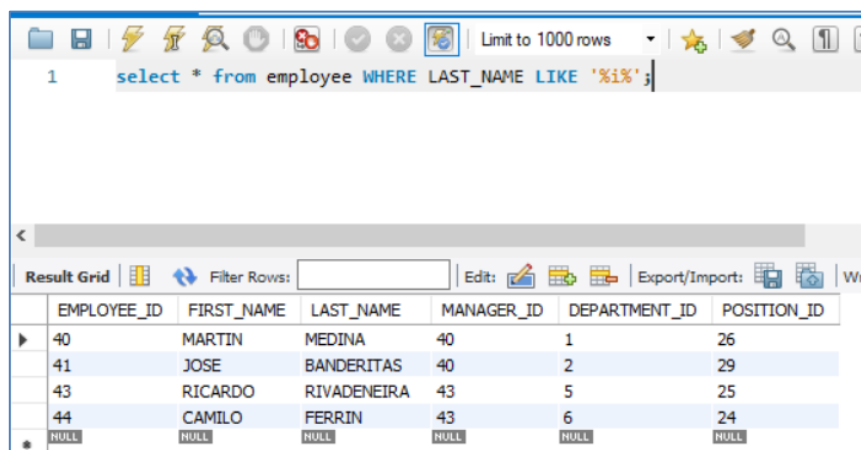
e) Listar todos los departamentos cuya localización está en Quito o Guayaquil.



1 `SELECT DEPARTMENT_NAME, LOCATION FROM department WHERE LOCATION="Quito" OR LOCATION="Guayaquil";`

DEPARTMENT_NAME	LOCATION
CONTABILIDAD	QUITO
SEGURIDAD	QUITO
RECURSOS HUMANOS	GUAYAQUIL
AUDITORIA	QUITO
CONTABILIDAD	GUAYAQUIL
LIMPIEZA	QUITO

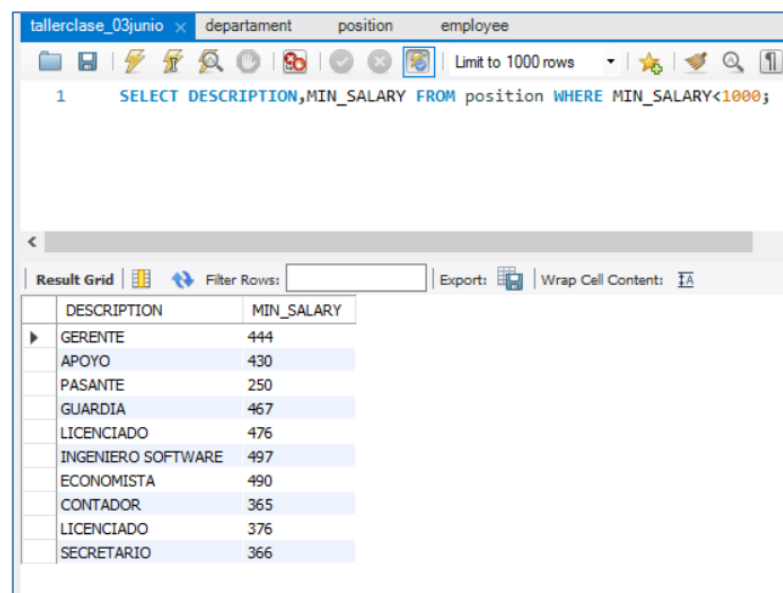
f) Listar los empleados que tienen en su apellido la letra “i”.



1 `select * from employee WHERE LAST_NAME LIKE '%i%';`

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	MANAGER_ID	DEPARTMENT_ID	POSITION_ID
40	MARTIN	MEDINA	40	1	26
41	JOSE	BANDERITAS	40	2	29
43	RICARDO	RIVADENEIRA	43	5	25
44	CAMILO	FERRIN	43	6	24
NULL	NULL	NULL	NULL	NULL	NULL

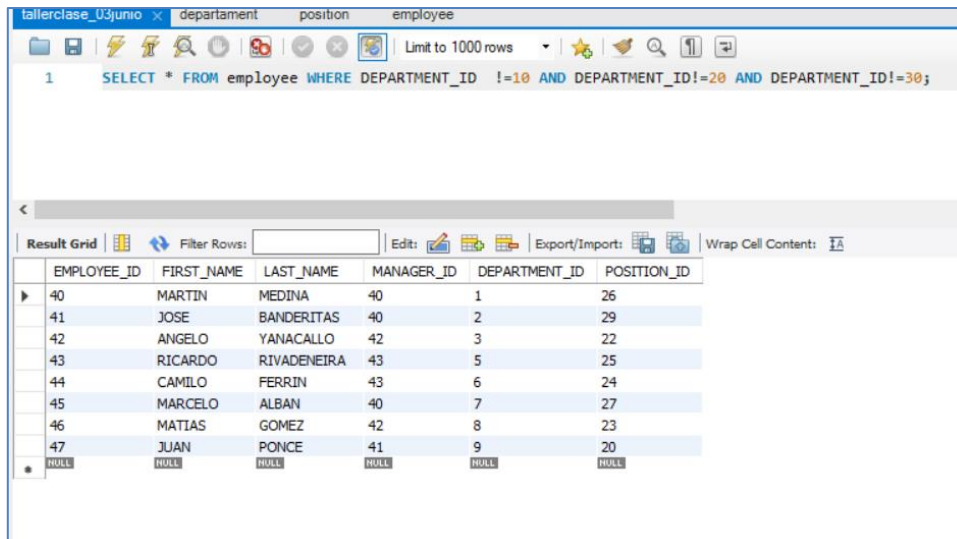
g) Mostrar la descripción de las posiciones de trabajo cuyos minimos salarios sean menores a \$1000.



1 `SELECT DESCRIPTION, MIN_SALARY FROM position WHERE MIN_SALARY<1000;`

DESCRIPTION	MIN_SALARY
GERENTE	444
APOYO	430
PASANTE	250
GUARDIA	467
LICENCIADO	476
INGENIERO SOFTWARE	497
ECONOMISTA	490
CONTADOR	365
LICENCIADO	376
SECRETARIO	366

h) Listar los empleados que no son de los departamentos 10, 20 o 30.



SQL Query:

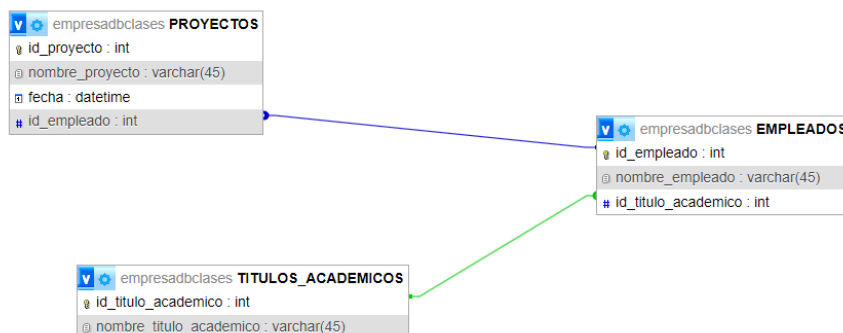
```
SELECT * FROM employee WHERE DEPARTMENT_ID !=10 AND DEPARTMENT_ID!=20 AND DEPARTMENT_ID!=30;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	MANAGER_ID	DEPARTMENT_ID	POSITION_ID
40	MARTIN	MEDINA	40	1	26
41	JOSE	BANDERITAS	40	2	29
42	ANGELO	YANACALLO	42	3	22
43	RICARDO	RIVADENEIRA	43	5	25
44	CAMILO	FERRIN	43	6	24
45	MARCELO	ALBAN	40	7	27
46	MATIAS	GOMEZ	42	8	23
47	JUAN	PONCE	41	9	20

EJERCICIO 3

Se tiene una base de datos que contiene información sobre proyectos y empleados de una empresa. Las tablas se muestran a continuación:

1. Utilizar MySQLWorkbench para elaborar el modelo lógico de la base de datos y generar el script SQL.



2. Importar el script generado a phpmyadmin y obtener el modelo físico.

```
-- MySQL Script generated by MySQL Workbench
-- Mon Jul 10 10:04:26 2023
-- Model: New Model    Version: 1.0
-- 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 empresadbclases
```

```
-- Schema empresadbclases
```

```
CREATE SCHEMA IF NOT EXISTS `empresadbclases` DEFAULT CHARACTER SET utf8 ;  
USE `empresadbclases` ;
```

```
-- Table `empresadbclases`.`TITULOS_ACADEMICOS`
```

```
CREATE TABLE IF NOT EXISTS `empresadbclases`.`TITULOS_ACADEMICOS` (  
  `id_titulo_academico` INT NOT NULL,  
  `nombre_titulo_academico` VARCHAR(45) NULL,  
  PRIMARY KEY (`id_titulo_academico`))  
ENGINE = InnoDB;
```

```
-- Table `empresadbclases`.`EMPLEADOS`
```

```
CREATE TABLE IF NOT EXISTS `empresadbclases`.`EMPLEADOS` (  
  `id_empleado` INT NOT NULL,  
  `nombre_empleado` VARCHAR(45) NULL,  
  `id_titulo_academico` INT NULL,  
  PRIMARY KEY (`id_empleado`),  
  INDEX `fk_EMPLADOS_TITULOS_ACADEMICOS_idx` (`id_titulo_academico` ASC) VISIBLE,  
  CONSTRAINT `fk_EMPLADOS_TITULOS_ACADEMICOS`  
    FOREIGN KEY (`id_titulo_academico`)  
    REFERENCES `empresadbclases`.`TITULOS_ACADEMICOS` (`id_titulo_academico`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- Table `empresadbclases`.`PROYECTOS`
```

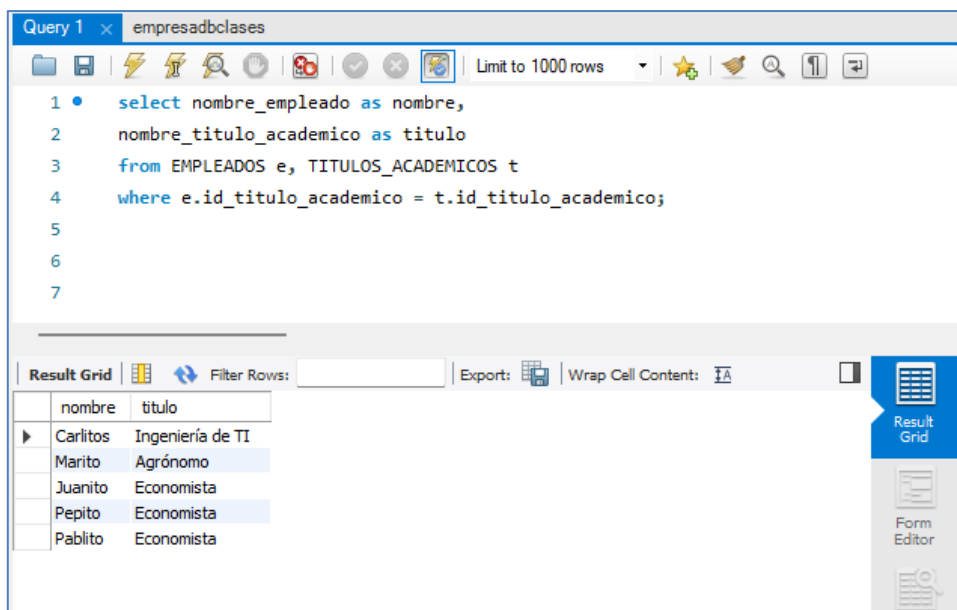
```
CREATE TABLE IF NOT EXISTS `empresadbclases`.`PROYECTOS` (  
  `id_proyecto` INT NOT NULL,  
  `nombre_proyecto` VARCHAR(45) NULL,  
  `fecha` DATETIME NULL,  
  `id_empleado` INT NULL,  
  PRIMARY KEY (`id_proyecto`),  
  INDEX `fk_PROYECTOS_EMPLADOS1_idx` (`id_empleado` ASC) VISIBLE,  
  CONSTRAINT `fk_PROYECTOS_EMPLADOS1`  
    FOREIGN KEY (`id_empleado`)  
    REFERENCES `empresadbclases`.`EMPLEADOS` (`id_empleado`)  
    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;
```

3. Realizar las siguientes consultas SQL a la base de datos, utilizando MySQLWorkbench (cliente de base de datos).

a) Mostrar los nombres de los empleados con sus títulos académicos



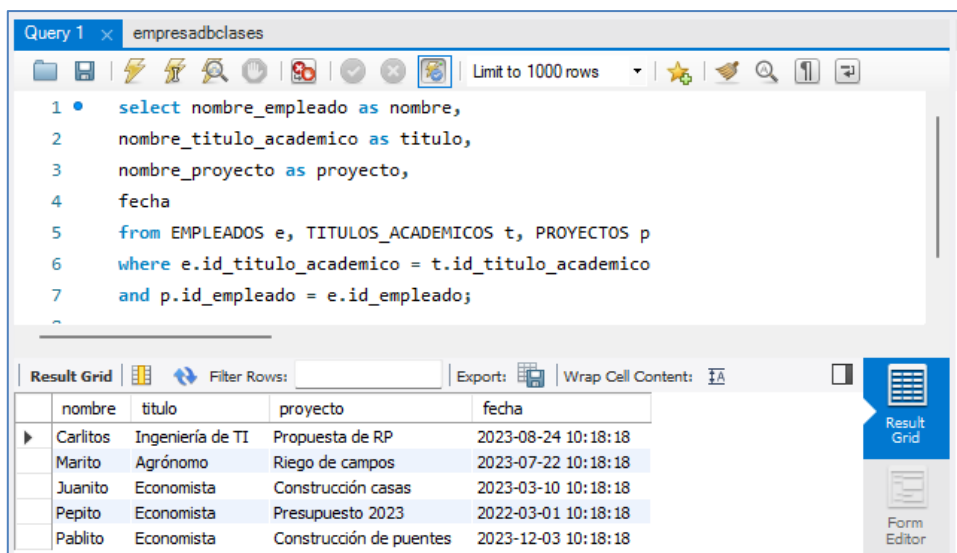
The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
1 • select nombre_empleado as nombre,  
2     nombre_titulo_academico as titulo  
3     from EMPLEADOS e, TITULOS_ACADEMICOS t  
4     where e.id_titulo_academico = t.id_titulo_academico;  
5  
6  
7
```

The results are displayed in the Result Grid below the query editor:

nombre	titulo
Carlitos	Ingeniería de TI
Marito	Agrónomo
Juanito	Economista
Pepito	Economista
Pablito	Economista

b) Mostrar el nombre del empleado el título académico el nombre del proyecto la fecha de ejecución del proyecto



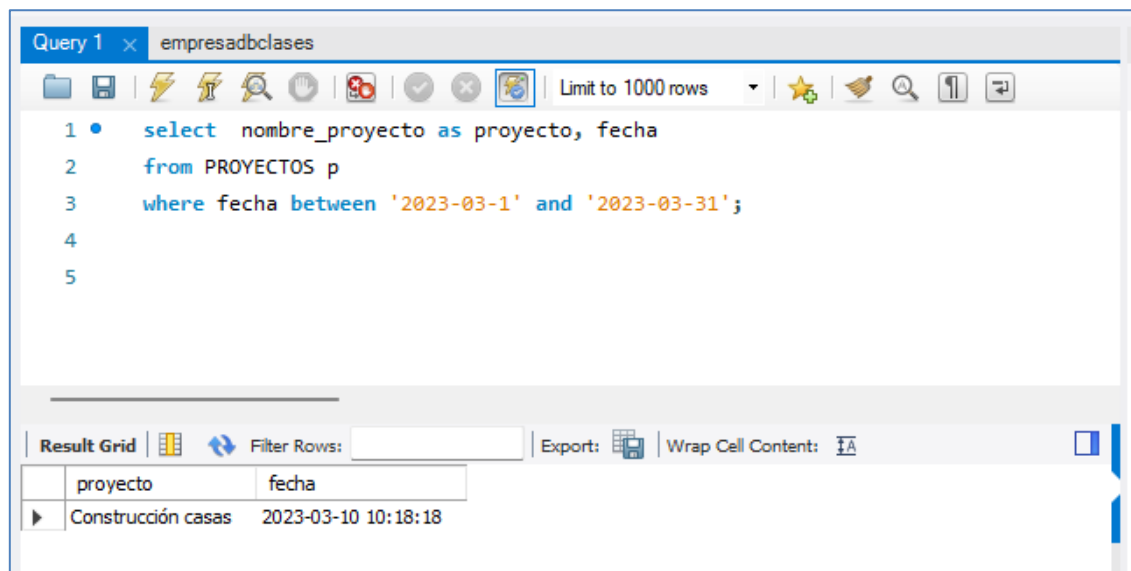
The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
1 • select nombre_empleado as nombre,  
2     nombre_titulo_academico as titulo,  
3     nombre_proyecto as proyecto,  
4     fecha  
5     from EMPLEADOS e, TITULOS_ACADEMICOS t, PROYECTOS p  
6     where e.id_titulo_academico = t.id_titulo_academico  
7     and p.id_empleado = e.id_empleado;  
8
```

The results are displayed in the Result Grid below the query editor:

nombre	titulo	proyecto	fecha
Carlitos	Ingeniería de TI	Propuesta de RP	2023-08-24 10:18:18
Marito	Agrónomo	Riego de campos	2023-07-22 10:18:18
Juanito	Economista	Construcción casas	2023-03-10 10:18:18
Pepito	Economista	Presupuesto 2023	2022-03-01 10:18:18
Pablito	Economista	Construcción de puentes	2023-12-03 10:18:18

c) Cuantos proyectos se ejecutaron en marzo del 2023



RESULTADOS OBTENIDOS:

Aprendimos a utilizar operadores especiales y técnicas de filtrado para obtener resultados específicos según nuestros criterios de búsqueda. Por ejemplo, pudimos ordenar los resultados alfabéticamente utilizando la cláusula "ORDER BY", lo cual nos permitió presentar la información de manera organizada y de fácil lectura.

Los demás resultados de las actividades de laboratorio fueron satisfactorios en cuanto a la creación de servidores locales y la configuración de bases de datos. Logramos establecer con éxito un entorno de servidor local utilizando la herramienta Workbench y un docker, lo cual nos permitió crear y administrar bases de datos utilizando MySQL.

También obtuvimos resultados satisfactorios en la elaboración de los modelos lógicos y físicos para los ejercicios de aves y empresa. Mediante el uso de Workbench pudimos identificar y representar adecuadamente las entidades, atributos y relaciones pertinentes a cada dominio.

CONCLUSIONES:

- Luego de comprender el entorno de phpMyAdmin, se logró generar exitosamente los scripts de los ejercicios del laboratorio. Esto demuestra que, al familiarizarse con esta herramienta, se puede aprovechar su potencial para crear scripts precisos y adecuados para las bases de datos.
- Al comprender y trabajar en la base de datos MySQLWorkbench, se pudo elaborar tanto el modelo lógico como el modelo conceptual de las bases de datos propuestas en los ejercicios del laboratorio. Esto indica que el dominio de MySQLWorkbench permite una visualización clara y una representación adecuada de la estructura y relaciones de las bases de datos.
- Mediante la aplicación de los conocimientos obtenidos en clase, se logró realizar las consultas solicitadas en los ejercicios y obtener los resultados correctos. Esto evidencia que los conocimientos adquiridos son fundamentales para la correcta ejecución de consultas en bases de datos, y que su aplicación adecuada lleva a obtener resultados precisos y relevantes.

RECOMENDACIONES:

- ✓ Antes de comenzar a realizar las consultas y sus tablas resultantes se debe tener un previo conocimiento de las cláusulas principales (FROM y SELECT), ya que son los que se van a usar para dar solución a los enunciados de los ejercicios planteados.
- ✓ Para realizar las diferentes consultas acordes al enunciado de cada literal de cada ejercicio, se debe leer las veces que sean necesarias hasta entender el contexto que nos pide, es decir, revisar una y otra vez las tablas de las bases de datos y comprender de que tablas nos pide el enunciado extraer la información o los datos necesarios para construir la expresión en SQL correctamente
- ✓ Para realizar las tablas resultantes de las consultas mediante SQL es conveniente empezar analizando desde las expresiones más pequeñas hasta las expresiones más grandes, de esta manera se consigue dibujar las tablas resultantes sin tantas complicaciones.

ELABORADO POR:

Rivadeneira Gómez Ricardo Xavier
Medina Armijos Martin Andrés
Yanacallo Monta Angelo Josué
Imbaquinga Guaña Jose Ricardo

ESTUDIANTE