

# **Normalización, 3FN y Diagramas ER**

**Isaac Damian Arrieta Mercado**  
**José Alejandro González Ortiz**

**Bases de Datos II**  
**Ruben Enrique Baena Navarro**  
**Ingeniería de Sistemas**  
**Universidad de Córdoba**  
**2020-2**

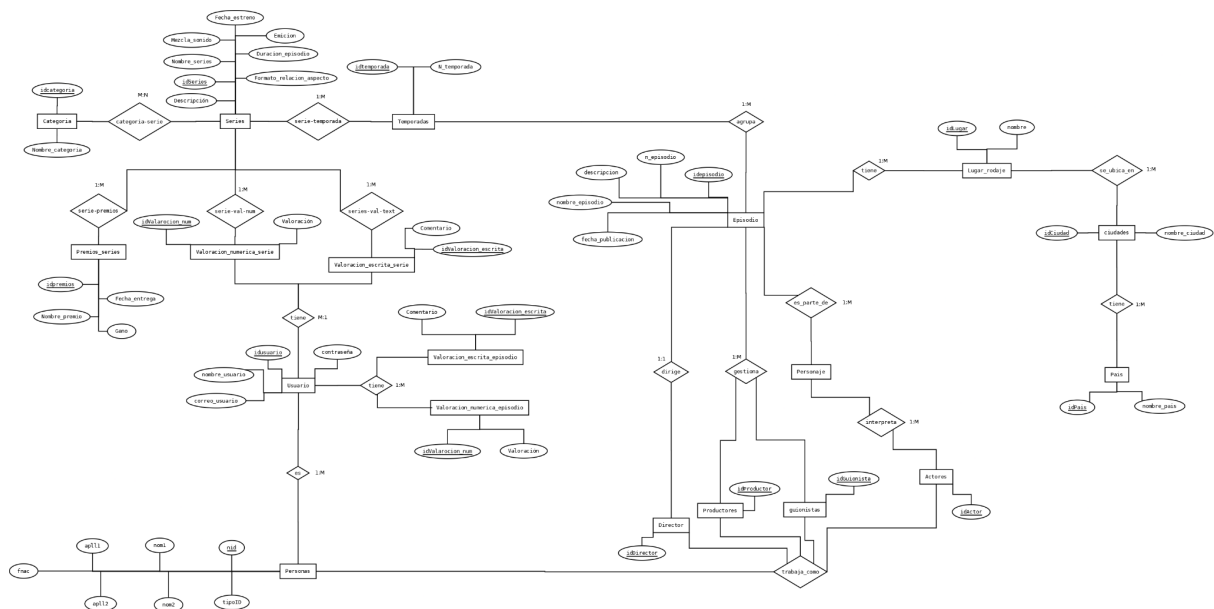
# Catálogo Serie #1

Una serie es una obra audiovisual compuesta de episodios, que mantienen una unidad argumental y con continuidad. Los capítulos se organizan en temporadas, y son estrenados en una fecha concreta. En cada capítulo aparecen diferentes personajes que son interpretados por actores. Además, cada capítulo tendrá un director, uno o más guionistas, uno o más productores, y uno o más lugares de rodaje. Una serie tiene una lista de especificaciones técnicas, duración de los capítulos, mezcla de sonido, formato, relación de aspecto (formato de alto y ancho que se emite).

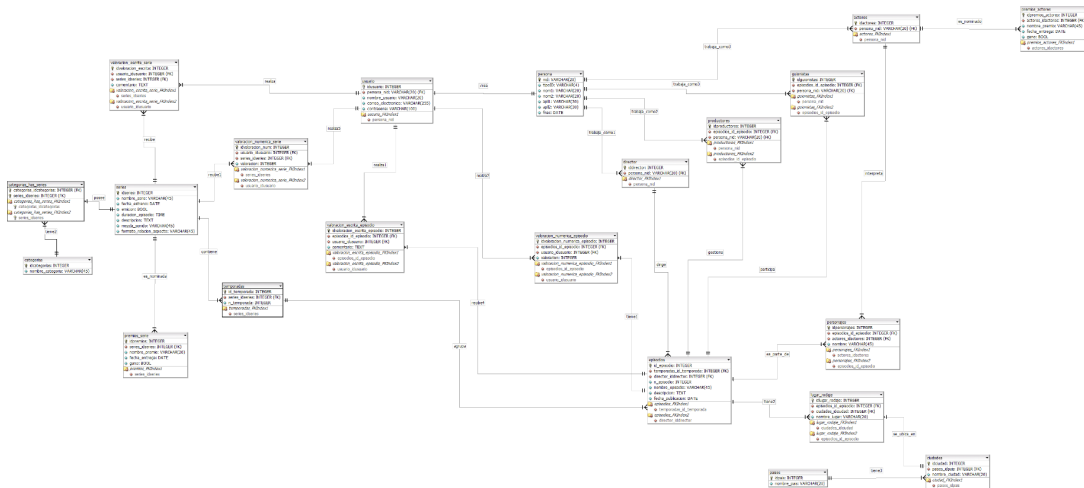
Cada serie puede estar asociada a una o varias categorías: drama, ciencia ficción, etc. Por cada serie se tiene una lista de premios a los que ha sido nominado y los que ha ganado. Los actores de una serie también pueden tener una lista de los premios a los que han sido nominados y los que han ganado por su rol en la serie.

Por cada capítulo de una serie, un usuario podrá hacer una valoración numérica y una escrita. Lo mismo ocurre para la serie en general. Además, el sistema permite realizar valoraciones solo a usuarios registrados. Para los usuarios registrados se almacenan el nombre, apellido, correo electrónico y valoraciones realizadas.

## Diagrama ER



# Diagrama Relacional



## Poblando Base de datos:

## MYSQL

phpMyAdmin

Reciente Favoritas

ejercicio 1

- Nueva
- actores
- categorias
- categorias\_has\_series
- ciudades
- director
- episodios
- guionistas
- lugar\_rodaje
- paises
- persona
- personajes
- premios\_actores
- premios\_serie
- productores
- series
- temporadas
- usuario
- valoracion\_escrita\_episodio
- valoracion\_escrita\_serie
- valoracion\_numerica\_episodio
- valoracion\_numerica\_serie

Servidor: 127.0.0.1 Base de datos: ejercicio 1

Estructura SQL Buscar Generar una consulta Exportar Importar Operaciones Privilegios Más

Importación ejecutada exitosamente. 21 consultas ejecutadas. (mysql1.sql)

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0379 segundos)

```
CREATE TABLE persona ( nid VARCHAR(50) NOT NULL , tipoID VARCHAR(4) NOT NULL , nom1 VARCHAR(50) NOT NULL , nom2 VARCHAR(50) NOT NULL , ap11 VARCHAR(30) NOT NULL , ap112 VARCHAR(30) NOT NULL , fnac DATE NOT NULL , PRIMARY KEY(nid));
```

[ Editar en línea ] [ Editar ] [ Crear código PHP ]

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0238 segundos)

```
CREATE TABLE paises ( idpais INTEGER UNSIGNED NOT NULL AUTO_INCREMENT , nombre_pais VARCHAR(50) NOT NULL , PRIMARY KEY(idpais));
```

[ Editar en línea ] [ Editar ] [ Crear código PHP ]

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0351 segundos)

```
CREATE TABLE series ( idseries INTEGER UNSIGNED NOT NULL AUTO_INCREMENT , nombre_serie VARCHAR(45) NOT NULL , fecha_estreno DATE NOT NULL , emicon BOOL NOT NULL , duracion_episodio TIME NOT NULL , descripcion TEXT NOT NULL , mezcla_sonido VARCHAR(45) NOT NULL , formato_relacion_aspecto VARCHAR(45) NOT NULL , PRIMARY KEY(idseries));
```

[ Editar en línea ] [ Editar ] [ Crear código PHP ]

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0389 segundos)

```
CREATE TABLE categorias ( idcategorias INTEGER UNSIGNED NOT NULL AUTO_INCREMENT , nombre_categoria VARCHAR(45) NOT NULL , PRIMARY KEY(idcategorias));
```

[ Editar en línea ] [ Editar ] [ Crear código PHP ]

Consola

Servidor: 127.0.0.1 Base de datos: ejercicio 1

Estructura SQL Buscar Generar una consulta Exportar Importar Operaciones Privilegios Rutinas Eventos Disparadores Diseñador

Mostrar ventana de consultas SQL

7 filas insertadas. (La consulta tardó 0.0043 segundos)

```
INSERT INTO persona (nid, tipoID, nom1, nom2, ap11, ap112, fnac) VALUES ('132545', 'C.C.', 'Juan', 'Antonio', 'Hernandez', 'Perez', '2000-01-01'), ('2144645', 'C.C.', 'Jhoan', 'Daniel', 'Ramirez', 'Artega', '2002-03-04'), ('514356', 'C.C.', 'Luz', 'Maria', 'Gonzalez', 'Lopez', '2001-08-20'), ('614356', 'C.C.', 'Maria', 'Camila', 'Pereira', 'Lopez', '2003-08-20'), ('714356', 'C.C.', 'Adria', 'Melissa', 'Westra', 'Martinez', '2000-08-20'), ('3215346', 'C.C.', 'Luisa', 'Maria', 'Velazquez', 'Espitia', '2002-05-20'), ('675443', '...');
```

[ Editar ]

4 filas insertadas. (La consulta tardó 0.0025 segundos)

```
INSERT INTO paises (idpais, nombre_pais) VALUES (1, 'Mexico'), (2, 'Colombia'), (3, 'Venezuela'), (4, 'Peru');
```

[ Editar en línea ] [ Editar ] [ Crear código PHP ]

4 filas insertadas. (La consulta tardó 0.0028 segundos)

```
INSERT INTO series (idseries, nombre_serie, fecha_estreno, emicon, duracion_episodio, descripcion, mezcla_sonido, formato_relacion_aspecto) VALUES (1, 'The Walking Dead', '2010-10-20', TRUE, '00:45:00', 'Una serie de ciencia ficción que narra la historia de una población de una región del norte de América, que se enfrenta a una pandemia que amenaza con la supervivencia de la humanidad.', 'TensionSong', '16:9'), (2, 'The Big Bang Theory', '2007-09-24', FALSE, '00:30:00', 'Una serie de televisión que narra la vida de una familia de quienes trabajan en una empresa de informática, donde la mayoría de sus miembros son estudiantes universitarios.', 'AccionSong', '16:9'), (3, 'Los Simpsons', '1989-12-17', TRUE, [...]);
```

[ Editar ]

8 filas insertadas. (La consulta tardó 0.0022 segundos)

```
INSERT INTO categorias (idcategorias, nombre_categoria) VALUES (1, 'Accion'), (2, 'Comedia'), (3, 'Drama'), (4, 'Terror'), (5, 'Fantasia'), (6, 'Anime');
```

[ Editar en línea ] [ Editar ] [ Crear código PHP ]

4 filas insertadas. (La consulta tardó 0.0020 segundos)

```
INSERT INTO usuario (idusuario, persona_nid, nombre_usuario, correo_electronico, contrasena) VALUES (1, 132545, 'Juan', 'juachito@gmail.com', 'Craken'), (2, 2144645, 'Jhoan', 'eljoal@gmail.com', 'surucutanga'), (3, 514356, 'Luz', 'labebeita@gmail.com', 'bebelin'), (4, 614356, 'Maria', 'mariela@gmail.com', 'deusnadeus');
```

[ Editar en línea ] [ Editar ] [ Crear código PHP ]

# PSQL

```
ejercicio1=# \dt
```

List of relations			
Schema	Name	Type	Owner
public	actores	table	postgres
public	categorias	table	postgres
public	categorias_has_series	table	postgres
public	ciudades	table	postgres
public	director	table	postgres
public	episodios	table	postgres
public	guionistas	table	postgres
public	lugar_rodaje	table	postgres
public	países	table	postgres
public	persona	table	postgres
public	personajes	table	postgres
public	premios_actores	table	postgres
public	premios_serie	table	postgres
public	productores	table	postgres
public	series	table	postgres
public	temporadas	table	postgres
public	usuario	table	postgres
public	valoracion_escrita_episodio	table	postgres
public	valoracion_escrita_serie	table	postgres
public	valoracion_numerica_episodio	table	postgres
public	valoracion_numerica_serie	table	postgres

(21 rows)

```
ejercicio1=# \i ./Actividad001/1/data.sql
```

```
INSERT 0 7
INSERT 0 4
INSERT 0 4
INSERT 0 6
INSERT 0 4
INSERT 0 4
INSERT 0 4
INSERT 0 4
INSERT 0 2
INSERT 0 4
INSERT 0 2
INSERT 0 4
INSERT 0 4
INSERT 0 4
INSERT 0 4
INSERT 0 5
INSERT 0 4
INSERT 0 4
INSERT 0 4
INSERT 0 4
INSERT 0 4
INSERT 0 4
INSERT 0 4
```

```
ejercicio1=#
```

## Tribu Arcoch #2

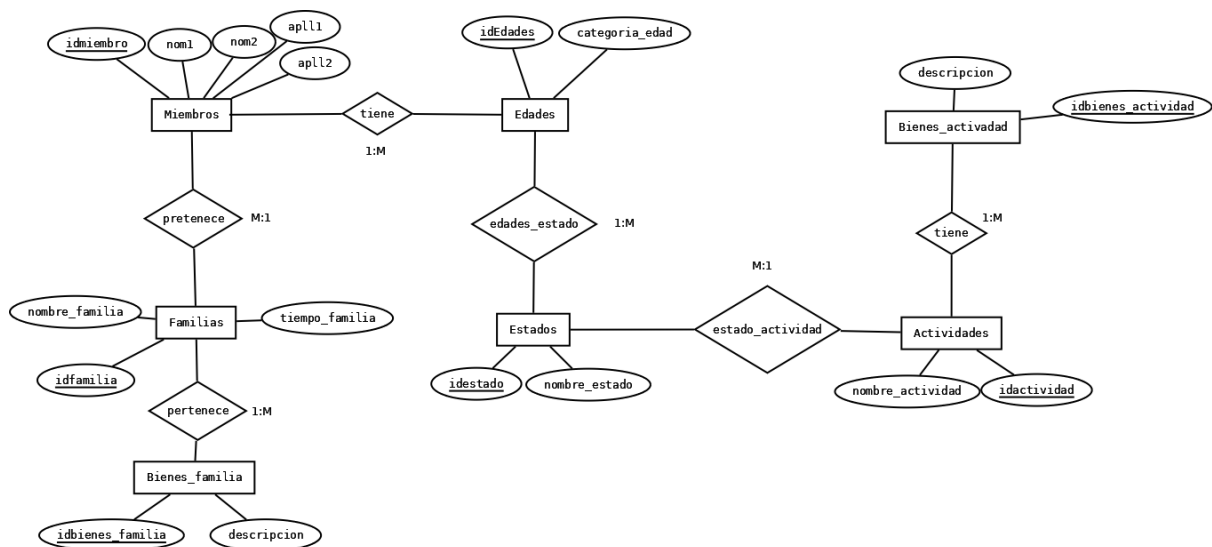
La tribu de los Arcoch, habitantes del Planeta Nak y únicos sobrevivientes de la guerra del año 2084, que destruyó los cuatro planetas de su sistema solar, se han propuesto construir una nueva forma de organización social para proteger su raza, actualmente en extinción.

Cada miembro de la tribu desarrolla alguna actividad, la cual debe aprender y luego perfeccionar. Por ejemplo, cazar, cultivar la tierra, educar, entre otros. Cada niño es educado e inicia alguna actividad en la adolescencia, cada anciano puede dejar de trabajar si lo desea.

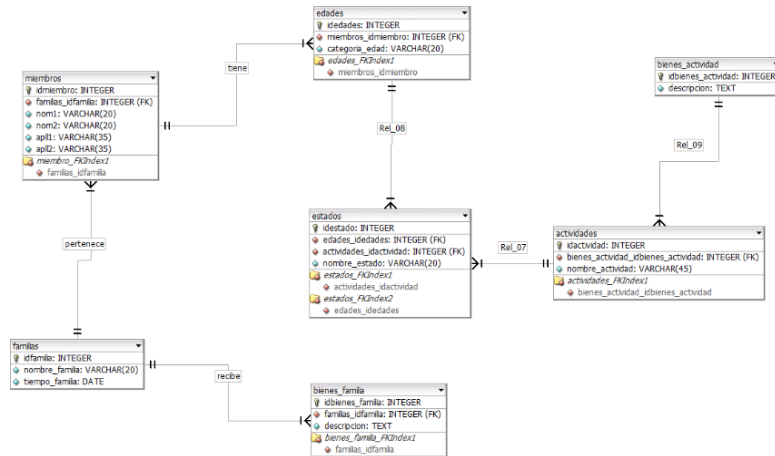
De acuerdo a la actividad que desarrolle cada Arcoch, se le entregan los elementos necesarios. Si se cambia de actividad estos deben ser devueltos. Se conoce que elementos deben ser entregados por actividad.

Tal como se otorgan bienes por actividad, se otorgan bienes a las familias de acuerdo a la edad de la familia (tiempo que lleva constituida la familia) y al número de integrantes que tenga. Por ejemplo, cuando dos Arcoch deciden formar una familia, la Tribu les da una choza y otros utensilios.

## Diagrama ER



# Diagrama Relacional



## Poblando Base de datos:

## MYSQL

Reciente Favoritas

- Nueva
- akeron
- ejercicio 1
- ejercicio2
  - Nueva
  - actividades
  - bienes\_actividad
  - bienes\_familia
  - edades
  - estados
  - familias
  - miembros
- empresa
- information\_schema
- mysql
- performance\_schema
- phpmyadmin
- test

Servidor: 127.0.0.1 Base de datos: ejercicio2

Estructura SQL Buscar Generar una consulta Exportar Importar Operaciones Privilegios Más

Importación ejecutada exitosamente, 7 consultas ejecutadas. (mysql2 sql)

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0334 segundos.)

```
CREATE TABLE familias ( idfamilia INTEGER UNSIGNED NOT NULL AUTO_INCREMENT, nombre_familia VARCHAR(20) NOT NULL, tiempo_familia DATE NOT NULL, PRIMARY KEY(idfamilia));
```

[ Editar en línea ] [ Editar ] [ Crear código PHP ]

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0245 segundos.)

```
CREATE TABLE bienes_actividad ( idbienes_actividad INTEGER UNSIGNED NOT NULL AUTO_INCREMENT, descripcion TEXT NOT NULL, PRIMARY KEY(idbienes_actividad));
```

[ Editar en línea ] [ Editar ] [ Crear código PHP ]

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0262 segundos.)

```
CREATE TABLE miembros ( idmiembro INTEGER UNSIGNED NOT NULL AUTO_INCREMENT, familias_idfamilia INTEGER UNSIGNED NOT NULL, nom1 VARCHAR(20) NOT NULL, nom2 VARCHAR(20) NULL, ap11 VARCHAR(35) NOT NULL, ap12 VARCHAR(35) NOT NULL, PRIMARY KEY(idmiembro), INDEX miembro_FKIndex1(familias_idfamilia), FOREIGN KEY(familias_idfamilia) REFERENCES familias(idfamilia) ON DELETE RESTRICT ON UPDATE CASCADE);
```

[ Editar en línea ] [ Editar ] [ Crear código PHP ]

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0433 segundos.)

```
CREATE TABLE edades ( idedades INTEGER UNSIGNED NOT NULL AUTO_INCREMENT, miembros_idmiembro INTEGER UNSIGNED NOT NULL, categoria_edad VARCHAR(20) NOT NULL, PRIMARY KEY(idedades), INDEX edades_FKIndex1(miembros_idmiembro), FOREIGN KEY(miembros_idmiembro) REFERENCES miembros(idmiembro) ON DELETE RESTRICT ON UPDATE CASCADE);
```

Consola

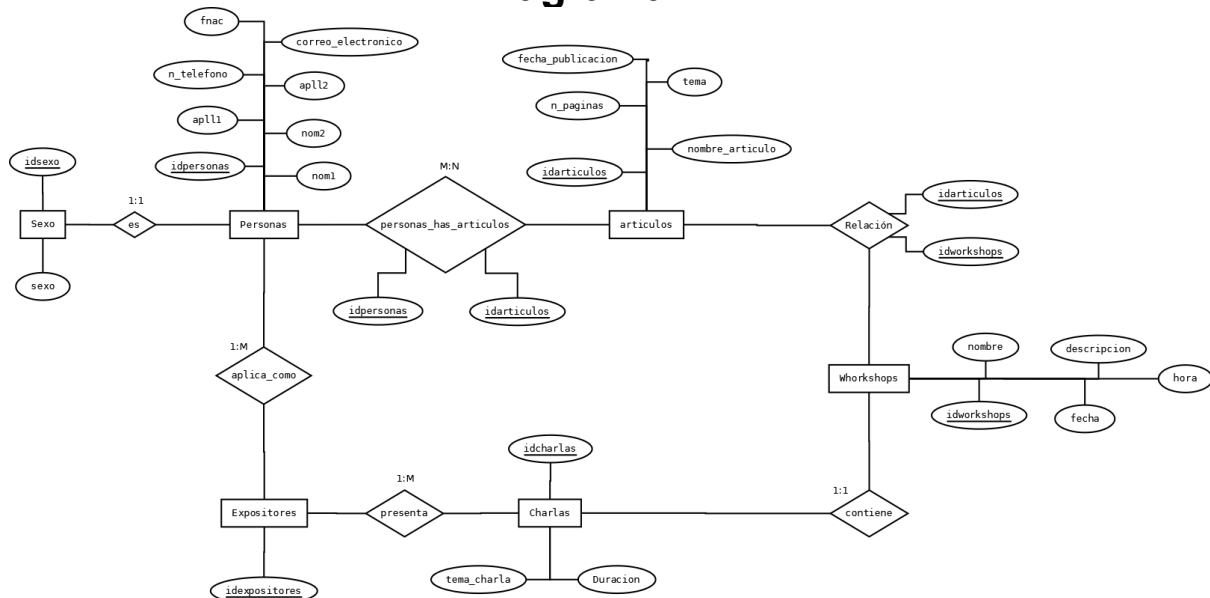
[illegible]

## Jornadas de la Computación #3

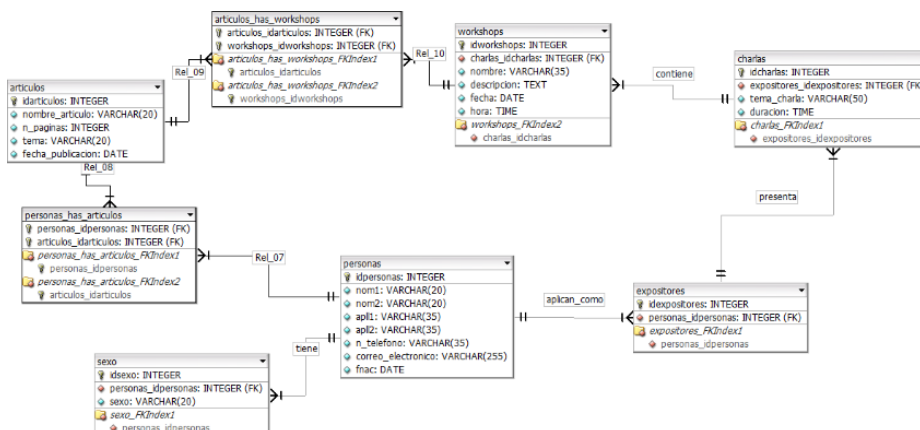
Para las Jornadas de Computación es indispensable tener una base de datos donde se guarde la información necesaria para la correcta organización de actividades.

En esta base de datos se tiene información sobre los diferentes Workshops, los artículos que se van a presentar en cada uno y las charlas que se darán. Cada artículo puede estar escrito por más de una persona a diferencia de las charlas donde solo hay un expositor. Tanto las charlas como los artículos son únicos por cada Workshop.

### Diagrama ER



### Diagrama Relacional





# Poblando Base de datos:

## MYSQL

The screenshot shows the phpMyAdmin interface with the 'ejercicio3' database selected. The left sidebar shows the database structure, including tables like 'personas', 'articulos', 'sexo', and 'expositores'. The main panel displays the SQL queries used to create these tables.

**Importación ejecutada exitosamente, 8 consultas ejecutadas. (mysql3 sql)**

**MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0321 segundos.)**

```
CREATE TABLE personas ( idpersonas INTEGER UNSIGNED NOT NULL AUTO_INCREMENT, nom1 VARCHAR(20) NOT NULL, nom2 VARCHAR(20) NOT NULL, ap11 VARCHAR(35) NOT NULL, ap12 VARCHAR(35) NOT NULL, n_telefono VARCHAR(35) NOT NULL, correo_electronico VARCHAR(255) NOT NULL, fnac DATE NOT NULL, PRIMARY KEY(idpersonas));
```

**MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0354 segundos.)**

```
CREATE TABLE articulos ( idarticulos INTEGER UNSIGNED NOT NULL AUTO_INCREMENT, nombre_articulo VARCHAR(20) NOT NULL, n_paginas INTEGER UNSIGNED NOT NULL, tema VARCHAR(20) NOT NULL, fecha_publicacion DATE NOT NULL, PRIMARY KEY(idarticulos));
```

**MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0209 segundos.)**

```
CREATE TABLE sexo ( idsexo INTEGER UNSIGNED NOT NULL AUTO_INCREMENT, personas_idpersonas INTEGER UNSIGNED NOT NULL, sexo VARCHAR(20) NOT NULL, PRIMARY KEY(idsexo), INDEX sexo_FKIndex1(personas_idpersonas), FOREIGN KEY(personas_idpersonas) REFERENCES personas(idpersonas) ON DELETE RESTRICT ON UPDATE CASCADE);
```

**MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0347 segundos.)**

```
CREATE TABLE expositores ( idexpositores INTEGER UNSIGNED NOT NULL AUTO_INCREMENT, personas_idpersonas INTEGER UNSIGNED NOT NULL, PRIMARY KEY(idexpositores), INDEX expositores_FKIndex1(personas_idpersonas), FOREIGN KEY(personas_idpersonas) REFERENCES personas(idpersonas) ON DELETE RESTRICT ON UPDATE CASCADE);
```

The screenshot shows the phpMyAdmin interface with the 'ejercicio3' database selected. The main panel displays the SQL queries used to insert data into the 'personas' table.

**1 fila insertada. (La consulta tardó 0,0021 segundos.)**

```
INSERT INTO personas (idpersonas, nom1, nom2, ap11, ap12, n_telefono, correo_electronico, fnac) VALUES (159753, 'GABRIEL', 'FELIPE', 'HERRERA', 'MORENO', '55555555', 'MORENO123@correo.com', '2008-01-01');
```

**1 fila insertada. (La consulta tardó 0,0020 segundos.)**

```
INSERT INTO personas (idpersonas, nom1, nom2, ap11, ap12, n_telefono, correo_electronico, fnac) VALUES (258852, 'GLORIA', 'PATRICIA', 'MENDOZA', 'ALVEAR', '45555986', 'ALVEAR23@correo.com', '1999-06-03');
```

**1 fila insertada. (La consulta tardó 0,0020 segundos.)**

```
INSERT INTO personas (idpersonas, nom1, nom2, ap11, ap12, n_telefono, correo_electronico, fnac) VALUES (147741, 'ESTEUIL', 'CARLOS', 'QUESADA', 'CALDERIN', '26845555', 'CALDERIN23@correo.com', '2002-08-10');
```

**1 fila insertada. (La consulta tardó 0,0019 segundos.)**

```
INSERT INTO personas (idpersonas, nom1, nom2, ap11, ap12, n_telefono, correo_electronico, fnac) VALUES (369963, 'CARLOS', 'DIDIER', 'CASTAÑO', 'CONTRERAS', '75655986', 'CONTRERAS3@correo.com', '1999-08-09');
```

**1 fila insertada. (La consulta tardó 0,0017 segundos.)**

```
INSERT INTO personas (idpersonas, nom1, nom2, ap11, ap12, n_telefono, correo_electronico, fnac) VALUES (456654, 'ADRIANA', 'MARCELA', 'REY', 'SANCHEZ', '23568555', 'SANCHEZ2123@correo.com', '2005-10-15');
```

**1 fila insertada. (La consulta tardó 0,0019 segundos.)**

```
INSERT INTO personas (idpersonas, nom1, nom2, ap11, ap12, n_telefono, correo_electronico, fnac) VALUES (159951, 'CINTHYA', 'FERNANDA', 'DUSSAN', 'GUZMAN', '45555986', 'SANCHEZ2123@correo.com', '1999-06-01');
```

# PSQL

```
ejercicio3=# \dt
```

## List of relations

Schema	Name	Type	Owner
public	articulos	table	postgres
public	articulos_has_workshops	table	postgres
public	charlas	table	postgres
public	expositores	table	postgres
public	personas	table	postgres
public	personas_has_articulos	table	postgres
public	sexo	table	postgres
public	workshops	table	postgres

(8 rows)

(8 rows)

ejercicio3=#

```
ejercicio3=# \i ./Actividad001/3/data.sql
```

INSERT 0 1

INSERT 0 1

INSERT 0 1

INSERT 0 1

INSERT 0 1

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
INSERT 0 1
```

```
INSERT 0 1
INSERT 0 1
```

```
INSERT 0 1
INSERT 0 1
```

```
INSERT 0 1
INSERT 0 1
```

```
INSERT 0 1
INSERT 0 1
```

INSERT 0 1  
INSERT 0 1

```
INSERT 0 1
INSERT 0 1
```

```
INSERT 0 1
INSERT 0 1
```

```
INSERT 0 1
INSERT 0 1
```

```
INSERT 0 1
INSERT 0 1
```

```
INSERT 0 1
INSERT 0 1
```

```
INSERT 0 1
INSERT 0 1
```

```
INSERT 0 1
INSERT 0 1
```

```
INSERT 0 1
INSERT 0 1
```

```
INSERT 0 1
INSERT 0 1
```

```
INSERT 0 1
INSERT 0 1
```

```
INSERT 0 1
INSERT 0 1
```

```
INSERT 0 1
INSERT 0 1
```

```
INSERT 0 1
INSERT 0 1
```

```
INSERT 0 1
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

INSERT 0 1

INSERT 0 1

INSERT 0 1

INSERT 0 1

INSERT 0 1

INSERT 0 1

INSERT 0 1

INSERT 0 1

```
INSERT 0 1
```

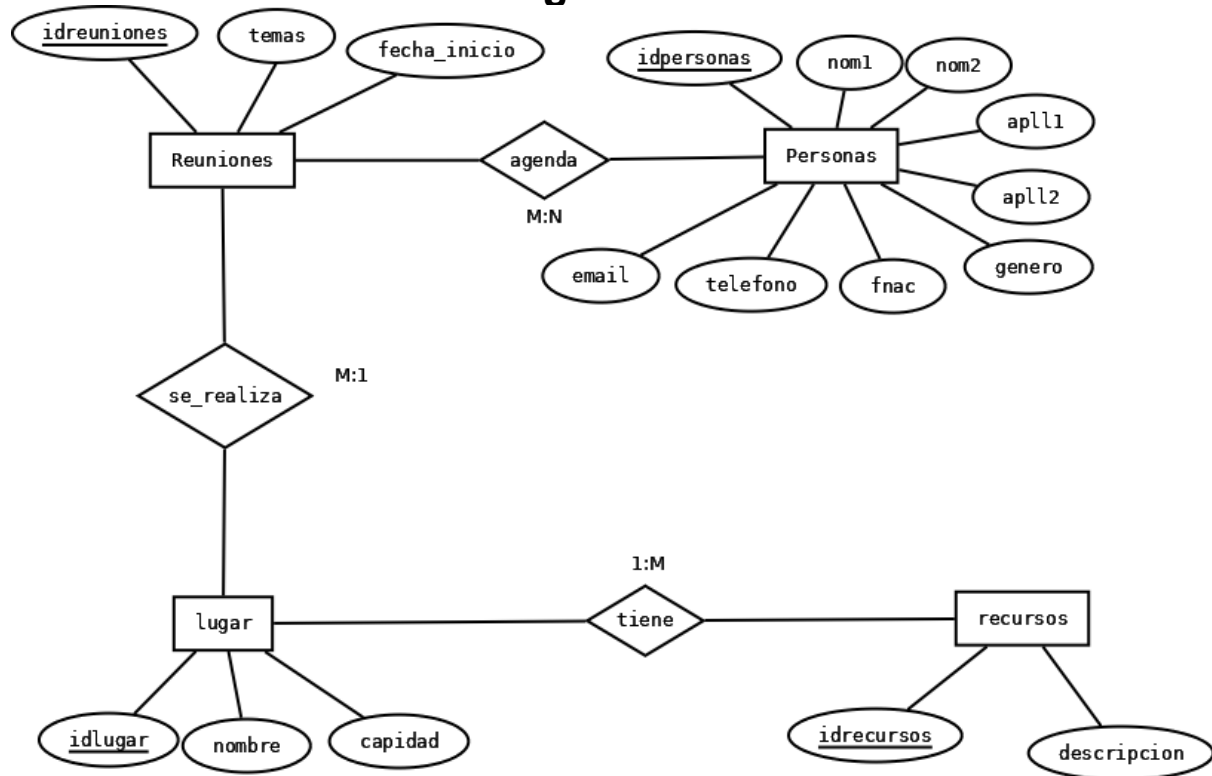
```
INSERT 0 1
```

ejercicio3=#

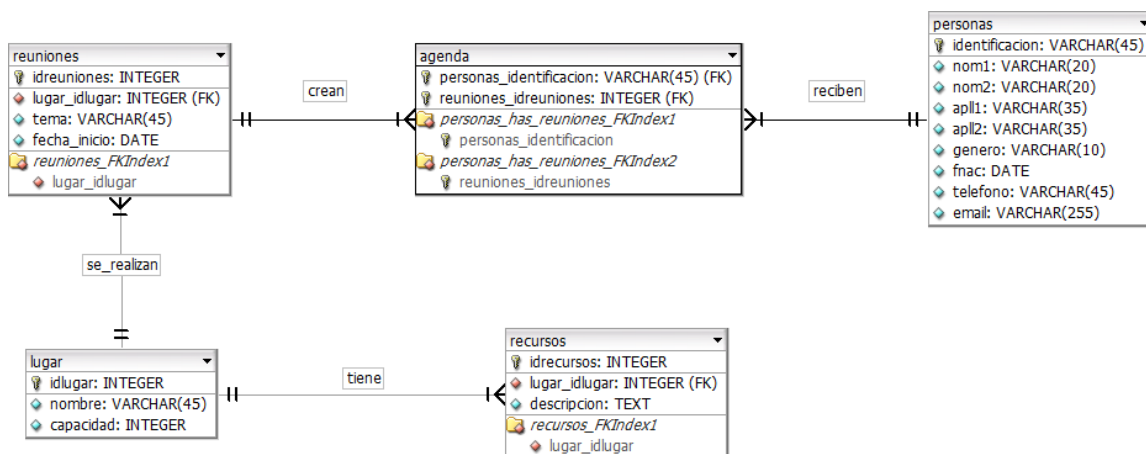
## Reuniones #4

Se desea guardar el registro de reuniones que se realizan en un Hospital. Cada persona es invitada a una o más reuniones, las cuales se realizan en un único lugar. La base de datos permite también saber los recursos que se encuentran en cada lugar.

### Diagrama ER



### Diagrama Relacional



# Poblando Base de datos:

## MYSQL

The screenshot displays the phpMyAdmin interface for a database named 'ejercicio4'. The left sidebar shows the database structure, with 'ejercicio4' selected. The main panel shows the 'Estructura' (Structure) tab, displaying the schema for three tables: 'personas', 'lugar', and 'reuniones'. Each table's structure is shown with its columns, data types, and constraints. Below the structure, the 'SQL' tab is active, showing the SQL queries used to create the tables and insert data. The queries are as follows:

```
CREATE TABLE personas (
  identificacion VARCHAR(45) NOT NULL,
  nom1 VARCHAR(20) NOT NULL,
  nom2 VARCHAR(20) NOT NULL,
  ap11 VARCHAR(35) NOT NULL,
  ap12 VARCHAR(35) NOT NULL,
  genero VARCHAR(10) NOT NULL,
  fnac DATE NOT NULL,
  telefono VARCHAR(45) NOT NULL,
  email VARCHAR(255) NOT NULL,
  PRIMARY KEY(identificacion));
```

```
CREATE TABLE lugar (
  idlugar INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  nombre VARCHAR(45) NOT NULL,
  capacidad INTEGER UNSIGNED NOT NULL,
  PRIMARY KEY(idlugar));
```

```
CREATE TABLE reuniones (
  idreuniones INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  lugar_idlugar INTEGER UNSIGNED NOT NULL,
  tema VARCHAR(45) NOT NULL,
  fecha_inicio DATE NOT NULL,
  PRIMARY KEY(idreuniones),
  INDEX reuniones_FKIndex1(lugar_idlugar),
  FOREIGN KEY(lugar_idlugar) REFERENCES lugar(idlugar) ON DELETE RESTRICT ON UPDATE CASCADE);
```

The 'SQL' tab also shows the 'Importación ejecutada exitosamente' (Importation successfully executed) message, indicating that the data has been successfully imported into the database. Below the SQL queries, the 'Consola' (Console) tab is active, showing the results of the queries. The console displays the following messages:

- 1 fila insertada. (La consulta tardó 0.0023 segundos.)
- 1 fila insertada. (La consulta tardó 0.0012 segundos.)
- 1 fila insertada. (La consulta tardó 0.0012 segundos.)
- 1 fila insertada. (La consulta tardó 0.0010 segundos.)
- 1 fila insertada. (La consulta tardó 0.0010 segundos.)

The console also shows the SQL queries used to insert the data into the 'personas' table:

```
INSERT INTO personas (identificacion, nom1, nom2, ap11, ap12, genero, fnac, telefono, email) VALUES ('29', 'Juan', 'Antonio', 'Hernandez', 'Perez', 'M', '2000-02-19', '3142345435', 'juancho@gmail.com');
```

```
INSERT INTO personas (identificacion, nom1, nom2, ap11, ap12, genero, fnac, telefono, email) VALUES ('30', 'Jhoan', 'Daniel', 'Ramirez', 'Arteaga', 'M', '2001-02-19', '31523445435', 'jhoan@gmail.com');
```

```
INSERT INTO personas (identificacion, nom1, nom2, ap11, ap12, genero, fnac, telefono, email) VALUES ('31', 'Luz', 'Maria', 'Estrella', 'Gonzalez', 'F', '2002-02-19', '3152344513', 'luzn@gmail.com');
```

```
INSERT INTO personas (identificacion, nom1, nom2, ap11, ap12, genero, fnac, telefono, email) VALUES ('32', 'Maria', 'Angelica', 'Gomez', 'Avila', 'F', '2000-05-02', '3217654874', 'avila@gmail.com');
```

```
INSERT INTO personas (identificacion, nom1, nom2, ap11, ap12, genero, fnac, telefono, email) VALUES ('33', 'Katerin', 'Estefani', 'Pereira', 'Lopez', 'F', '2002-04-27', '312756534', 'katerin12@gmail.com');
```

## PSQL

```
ejercicio4=# \dt
```

List of relations

Schema	Name	Type	Owner
public	agenda	table	postgres
public	lugar	table	postgres
public	personas	table	postgres
public	recursos	table	postgres
public	reuniones	table	postgres

(5 rows)

```
ejercicio4=# \i ./Actividad001/4/data.sql
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

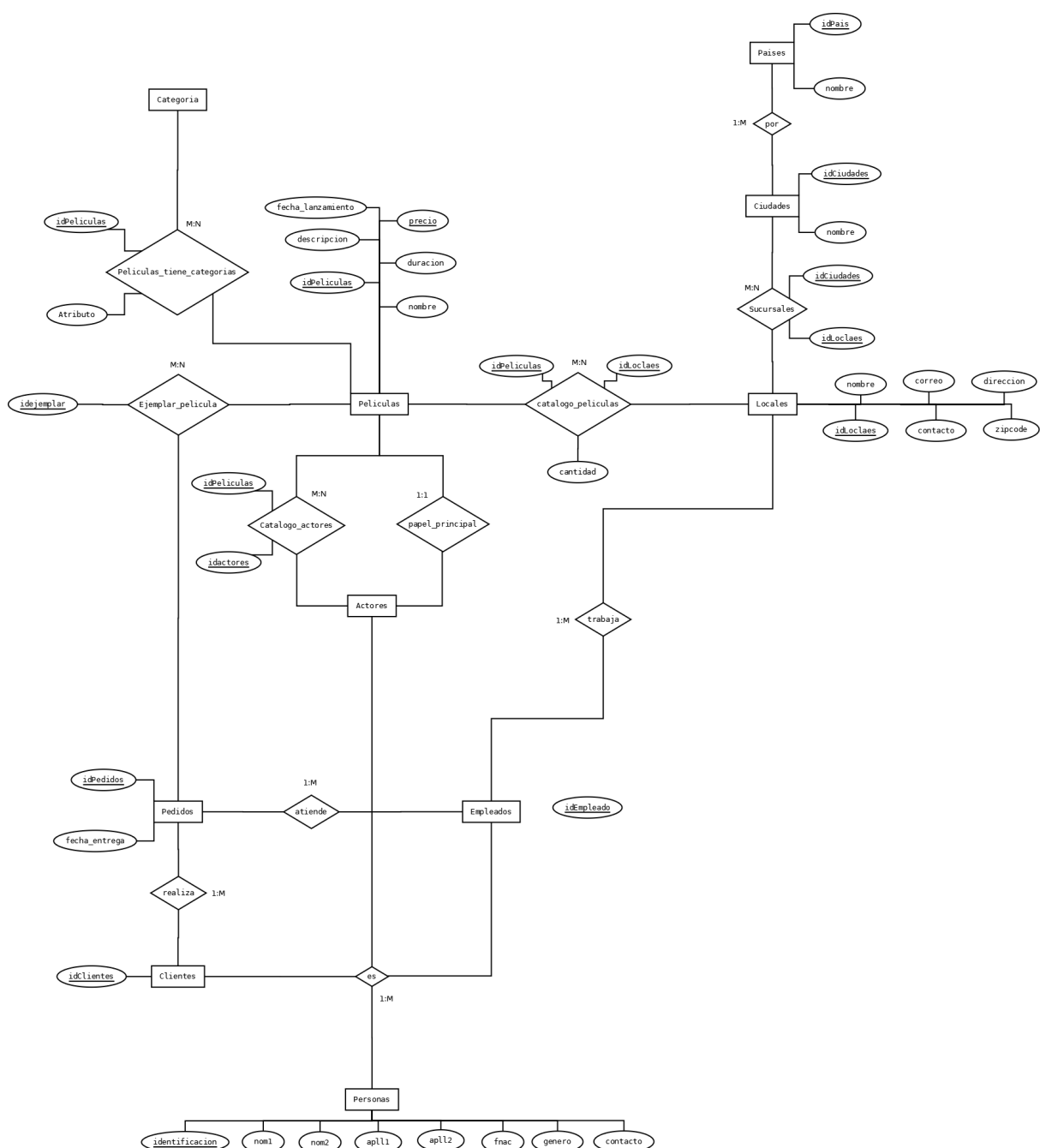
```
ejercicio4=#
```

## Arriendo de Peliculas #5

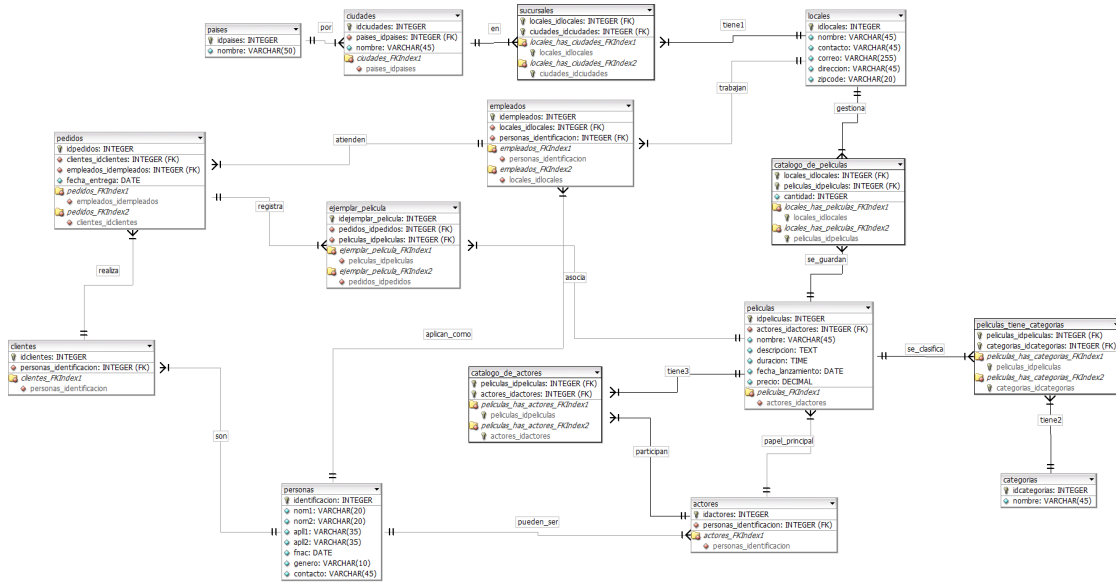
Se necesita una base de datos para la administración de una cadena de locales de arriendo de películas. La cadena de arriendo tiene locales en varios países donde comparten una variedad de películas globalmente, cada local cuenta con un stock propio a través de un inventario. Cada película tiene una o más categorías como uno o más actores. Se debe conocer la categoría y actor principal para ambos casos.

Cuando un usuario realiza un arriendo es necesario saber de qué tienda se realizó, que película y que persona atendió al cliente.

### Diagrama ER



## Diagrama Relacional



## Poblando Base de datos:

# MYSQL

**phpMyAdmin**

Servidor: 127.0.0.1 » Base de datos: ejercicio5 » Tabla: personas

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones Más

Reciente Favoritas

- ejercicio4
- ejercicio5**
  - Nueva
  - actores
  - catalogo\_de\_actores
  - catalogo\_de\_peliculas
  - categorias
  - ciudades
  - clientes
  - ejemplar\_pelicula
  - empleados
  - locales
  - paises
  - pedidos
  - peliculas
  - peliculas\_tiene\_categorias
  - personas
  - sucursales
- empresa
- information\_schema
- mysql
- performance\_schema
- phpmyadmin
- test

+ Opciones

		identificacion	nom1	nom2	apll1	apll2	fnac	genero	contacto
<input type="checkbox"/>	[Editar] [Copiar] [Borrar]	1	Maddie	Arni	Bellenie	Benesevich	2019-09-09	M	11584-0029
<input type="checkbox"/>	[Editar] [Copiar] [Borrar]	2	Carlotta	Meehan	Posnett	Dameisele	2020-08-05	F	42799-217
<input type="checkbox"/>	[Editar] [Copiar] [Borrar]	3	Laura	Molgee	Greenin	Tanton	2013-07-02	F	55154-4727
<input type="checkbox"/>	[Editar] [Copiar] [Borrar]	4	Roddy	Delbert	Miner	Bedin	2004-02-20	M	59746-308
<input type="checkbox"/>	[Editar] [Copiar] [Borrar]	5	Agace	Fredelia	MacKenny	Simeoli	2021-04-25	F	55289-324
<input type="checkbox"/>	[Editar] [Copiar] [Borrar]	6	Patric	Orland	Learnmond	Mc Gee	2003-07-24	M	62257-585
<input type="checkbox"/>	[Editar] [Copiar] [Borrar]	7	Wren	Katuscha	Presshaugh	McFarlane	2016-03-24	F	68084-282
<input type="checkbox"/>	[Editar] [Copiar] [Borrar]	8	Jeralae	Della	Burker	Milesop	2002-06-15	F	36987-2923
<input type="checkbox"/>	[Editar] [Copiar] [Borrar]	9	Elwyn	Renado	Mallison	Carnier	2006-10-25	M	0003-2327
<input type="checkbox"/>	[Editar] [Copiar] [Borrar]	10	Cristiano	Jerome	Dayes	Friman	2011-06-08	M	67475-114
<input type="checkbox"/>	[Editar] [Copiar] [Borrar]	11	Howard	Skipp	Claye	Veazey	2010-04-01	M	58407-625
<input type="checkbox"/>	[Editar] [Copiar] [Borrar]	12	Vyky	Dasi	Rackstraw	Gounod	2007-11-05	F	0143-9660
<input type="checkbox"/>	[Editar] [Copiar] [Borrar]	13	Herta	Haley	Valeri	Bayley	1998-12-30	F	49981-001
<input type="checkbox"/>	[Editar] [Copiar] [Borrar]	14	Riley	Derrek	Pirie	Fames	2021-08-28	M	68786-228
<input type="checkbox"/>	[Editar] [Copiar] [Borrar]	15	Eal	Dore	Davydkov	Yanuk	2019-11-16	M	68786-211
<input type="checkbox"/>	[Editar] [Copiar] [Borrar]	16	Nolan	Camrine	Corriea	Cottel	2017-07-01	M	50268-618

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

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Sort by key: Ninguna

Consola

```
ejercicio5=# \dt
```

Schema	Name	Type	Owner
public	actores	table	postgres
public	catalogo_de_actores	table	postgres
public	catalogo_de_peliculas	table	postgres
public	categorias	table	postgres
public	ciudades	table	postgres
public	clientes	table	postgres
public	ejemplar_pelicula	table	postgres
public	empleados	table	postgres
public	locales	table	postgres
public	países	table	postgres
public	pedidos	table	postgres
public	peliculas	table	postgres
public	peliculas_tiene_categorias	table	postgres
public	personas	table	postgres
public	sucursales	table	postgres

```
(15 rows)
```

```
ejercicio5=#
```



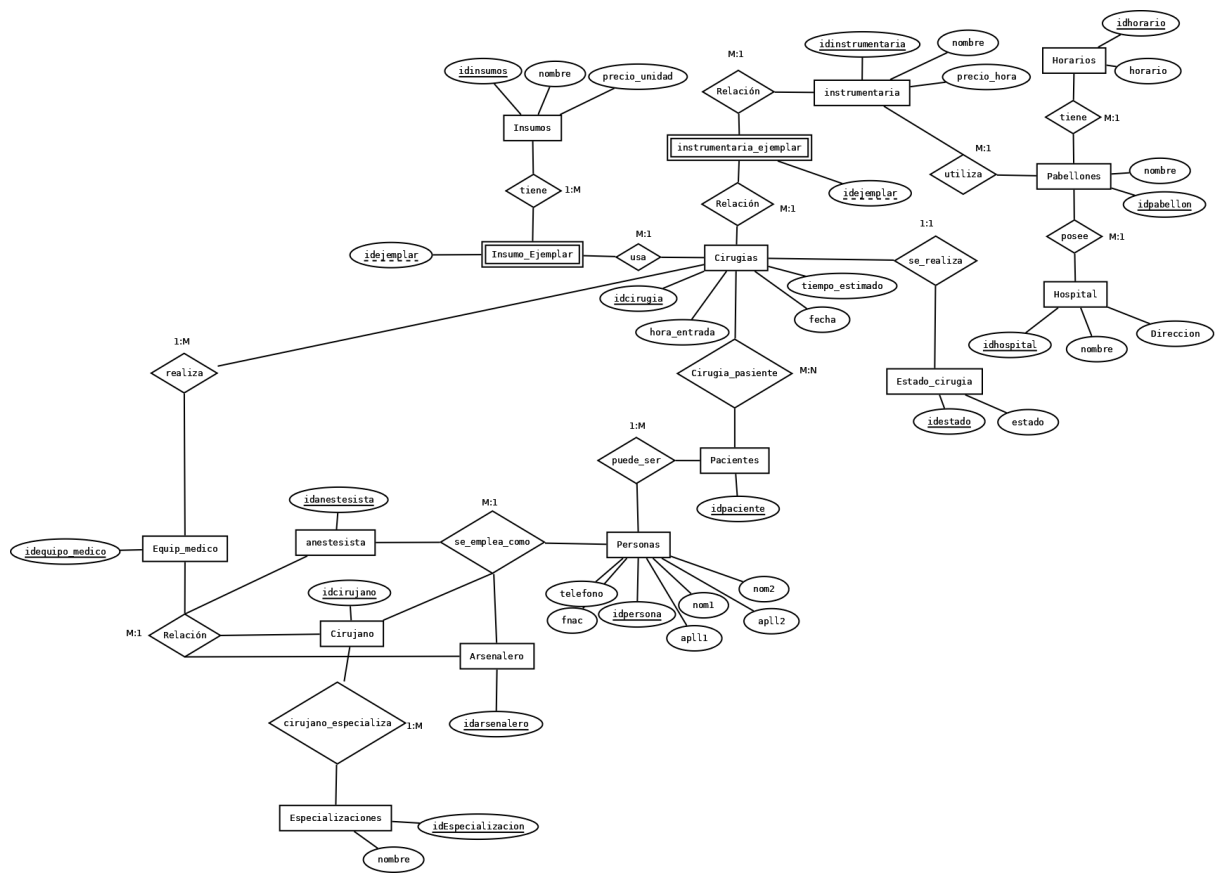
## **Pabellón #6**

El Pabellón Quirúrgico es un recinto especialmente diseñado y equipado para garantizar la seguridad en la atención del paciente, que va a ser sometido a una actividad anestésica o quirúrgica.

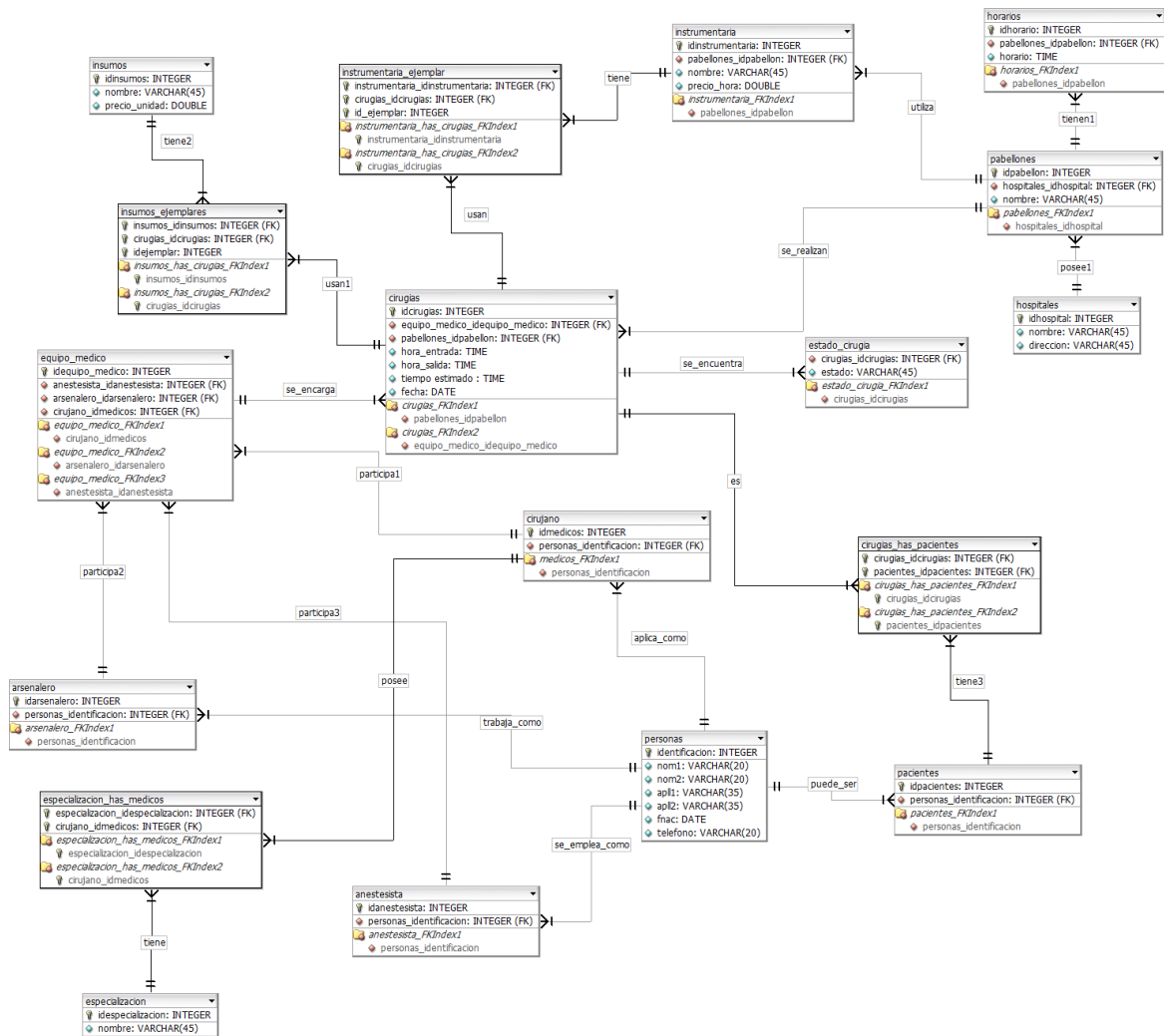
El módulo de pabellón tiene por finalidad gestionar las diversas intervenciones que se van a realizar a los pacientes, el equipo médico que va a trabajar, materiales, horarios de las cirugías entre otros. La especificación de los requisitos se encuentra en el siguiente listado:

1. Cada Hospital tiene un número determinado de pabellones, los cuales poseen instrumentaría especial para los diversos tipos de cirugías.
2. El sistema debe permitir el ingreso de pacientes, médicos, pabellones, instrumentaría, insumos y cirugías. Además, el sistema debe permitir especificar la instrumentaría necesaria para cada cirugía, así como gestionar la instrumentaría existente en cada pabellón.
3. Cuando se desee agendar una cirugía, es necesario especificar el equipo médico (agregados previamente al sistema) y la cirugía a realizar, automáticamente el sistema devolvería los horarios disponibles de los pabellones que cumplan con los requisitos.
4. Al momento de agendar una cirugía, se especifica el tiempo aproximado de duración. Como es posible que ocurran imprevistos, el sistema debe permitir manejar éstos, asignándole mayor tiempo en caso de ser necesario y replanificando las cirugías que seguían a continuación de la que está en transcurso.
5. El sistema debe permitir ver todas las cirugías a la cual fue sometido un paciente, con su respectivo detalle, como la instrumentaría, insumos, equipo médico y duración, con el fin de realizar la facturación una vez finalice.
6. Cada insumo tiene un precio por unidad y la instrumentaría un precio por hora. En el caso especial de los insumos, cuando se realiza el ingreso al sistema se da un stock que se va descontando en cada utilización.
7. Se debe permitir la visualización de todas las cirugías realizadas por los médicos, las cirugías que están en pabellón, las cirugías a realizarse en un futuro, entre otra información que se considere importante para el módulo.
8. Al momento de agendar una cirugía, se debe agregar cirujano, un arsenalero y un anestesista. Si no se encuentran alguno de estos profesionales, la cirugía no podrá ser llevada a cabo.

# Diagrama ER



## Diagrama Relacional



# Poblando Base de datos:

## MYSQL

The image shows two screenshots of the phpMyAdmin interface. The top screenshot displays the 'Base de datos: ejercicio6' window with a list of tables on the left and a console on the right showing the execution of SQL commands to create tables and insert data. A red arrow points to 'ejercicio6' in the table list. The bottom screenshot shows the 'Tabla: personas' window with a table of data.

**Top Screenshot: Database Creation and Data Insertion**

Consola

```
Importación ejecutada exitosamente, 18 consultas ejecutadas. (mysql6.sql)

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0328 segundos.)

CREATE TABLE insumos ( idinsumos INTEGER UNSIGNED NOT NULL AUTO_INCREMENT, nombre VARCHAR(45) NOT NULL , precio_unidad DOUBLE NOT NULL , PRIMARY KEY(idinsumos));

[ Editar en línea ] [ Editar ] [ Crear código PHP ]

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0267 segundos.)

CREATE TABLE hospitales ( idhospital INTEGER UNSIGNED NOT NULL AUTO_INCREMENT, nombre VARCHAR(45) NOT NULL , direccion VARCHAR(45) NOT NULL , PRIMARY KEY(idhospital));

[ Editar en línea ] [ Editar ] [ Crear código PHP ]

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0252 segundos.)

CREATE TABLE especializacion ( idespecializacion INTEGER UNSIGNED NOT NULL AUTO_INCREMENT, nombre VARCHAR(45) NULL , PRIMARY KEY(idespecializacion));

[ Editar en línea ] [ Editar ] [ Crear código PHP ]

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0270 segundos.)

CREATE TABLE personas ( identificacion INTEGER UNSIGNED NOT NULL AUTO_INCREMENT, nom1 VARCHAR(20) NOT NULL , nom2 VARCHAR(20) NOT NULL , ap11 VARCHAR(35) NOT NULL , ap12 VARCHAR(35) NOT NULL , fnac DATE NOT NULL , telefono VARCHAR(20) NOT NULL , PRIMARY KEY(identificacion));

[ Editar en línea ] [ Editar ] [ Crear código PHP ]
```

**Bottom Screenshot: Table Population**

Tabla: personas

SELECT \* FROM `personas`

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

Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Sort by key: Ninguna

	identificacion	nom1	nom2	ap11	ap12	fnac	telefono
<input type="checkbox"/>	24	Julieta	Ponce	Peña	Posada	1987-08-26	312459383
<input type="checkbox"/>	25	Valeria	Elvia	Hernández	Meneses	1986-07-16	321563378
<input type="checkbox"/>	26	Mauricio	Andres	Jiménez	Piedrahita	1986-02-13	312247689
<input type="checkbox"/>	27	Martin	Elias	Acosta	Muñoz	1990-08-21	3143576748
<input type="checkbox"/>	28	Pedro	Andres	Perez	Díaz	1989-04-27	3134345564
<input type="checkbox"/>	29	Juan	Antonio	Hernandez	Perez	1974-02-19	3142345435
<input type="checkbox"/>	30	Jhoan	Daniel	Ramirez	Arteaga	1985-02-19	31523445435
<input type="checkbox"/>	31	Luz	Maria	Estrella	Gonzalez	1978-02-19	3152344513
<input type="checkbox"/>	32	Maria	Angelica	Gomez	Avila	1989-05-02	3217654874
<input type="checkbox"/>	33	Katerin	Estefani	Pereira	Lopez	1980-04-27	312756534
<input type="checkbox"/>	34	Adalia	Melissa	Mestra	Martinez	1970-09-09	316345435
<input type="checkbox"/>	35	Mauricio	Andres	Altamiranda	Alzate	1974-10-09	3231432532

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

# PSQL

```
ejercicio6=# \dt
```

## List of relations

Schema	Name	Type	Owner
public	anestesista	table	postgres
public	arsenalero	table	postgres
public	cirugias	table	postgres
public	cirugias_has_pacientes	table	postgres
public	cirujano	table	postgres
public	equipo_medico	table	postgres
public	especializacion	table	postgres
public	especializacion_has_medicos	table	postgres
public	estado_cirugia	table	postgres
public	horarios	table	postgres
public	hospitales	table	postgres
public	instrumentaria	table	postgres
public	instrumentaria_ejemplar	table	postgres
public	insumos	table	postgres
public	insumos_ejemplares	table	postgres
public	pabellones	table	postgres
public	pacientes	table	postgres
public	personas	table	postgres

(18 rows)

```
ejercicio6=# \i ./Actividad001/6/data.sql
```

```
INSERT 0 1
```

```
INSERT 0 1
```

INSERT 0 1

INSERT 0 1

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

INSERT 0 1

INSERT 0 1

INSERT 0 1

INSERT 0 1

INSERT 0 1

INSERT 0 1

INSERT 0 1

INSERT 0 1

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

INSERT 0 1

INSERT 0 1

```
INSERT 0 1
```

## Gestión Hotelera #7

Un pequeño hotel rural necesita una aplicación software que le permita gestionar su negocio. Las primeras reuniones se han resumido en un documento expresado en lenguaje natural, que recoge a grandes rasgos la lógica de negocio del sistema a construir. El software a construir debe cumplir las siguientes funcionalidades:

1. Gestionar las reservas de habitaciones
2. Gestionar las habitaciones libres/ocupadas
3. Gestionar gastos extras
4. Facturar a los clientes
5. Permitir listados y estadísticas

### Reservas

Las reservas se hacen por teléfono, y el cliente debe dar su nombre, DNI, teléfono de contacto. La reserva tendrá validez hasta las 0:00 horas del día siguiente a la supuesta entrada en el hotel. Un cliente puede reservar tantas habitaciones como sean necesarias.

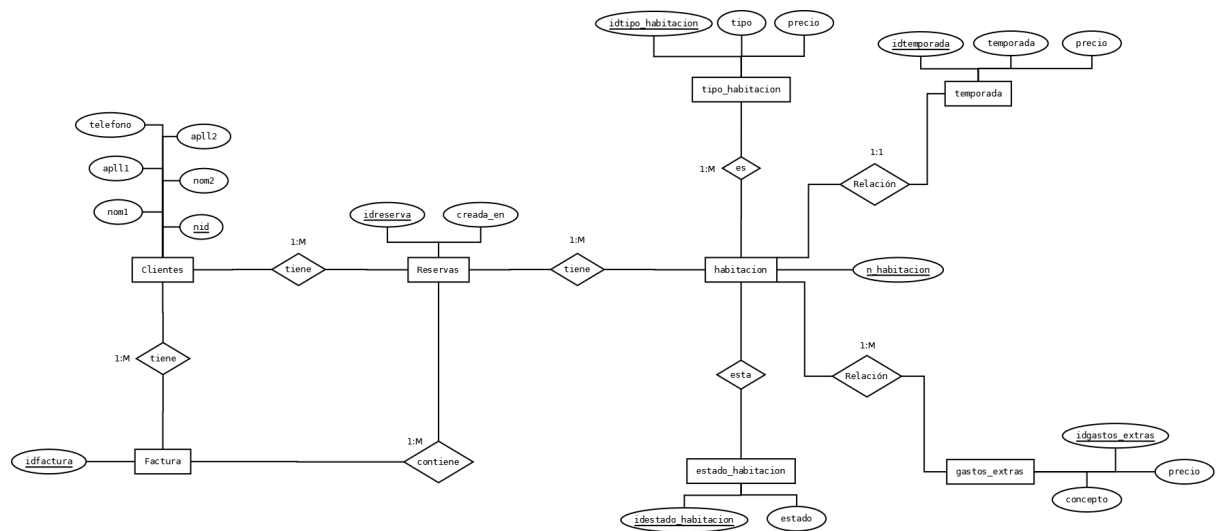
### Habitaciones

El hotel tiene tres pisos con habitaciones para los clientes. Cada habitación tiene un número de habitación y pertenece a un tipo de habitación (los tipos existentes son individual, doble y matrimonial). El precio de la habitación depende del tipo de la habitación y de la temporada. Existen tres temporadas: baja, alta y normal. Las habitaciones pueden estar libres, reservadas u ocupadas. Una habitación está ocupada sólo en el momento que el cliente ha llegado al hotel y hay un responsable de la habitación del cual se necesita su nombre, NID y teléfono de contacto. Si la habitación estaba reservada, la reserva ya no se necesita y puede eliminarse. Sin embargo, si debe existir información histórica del uso del hotel.

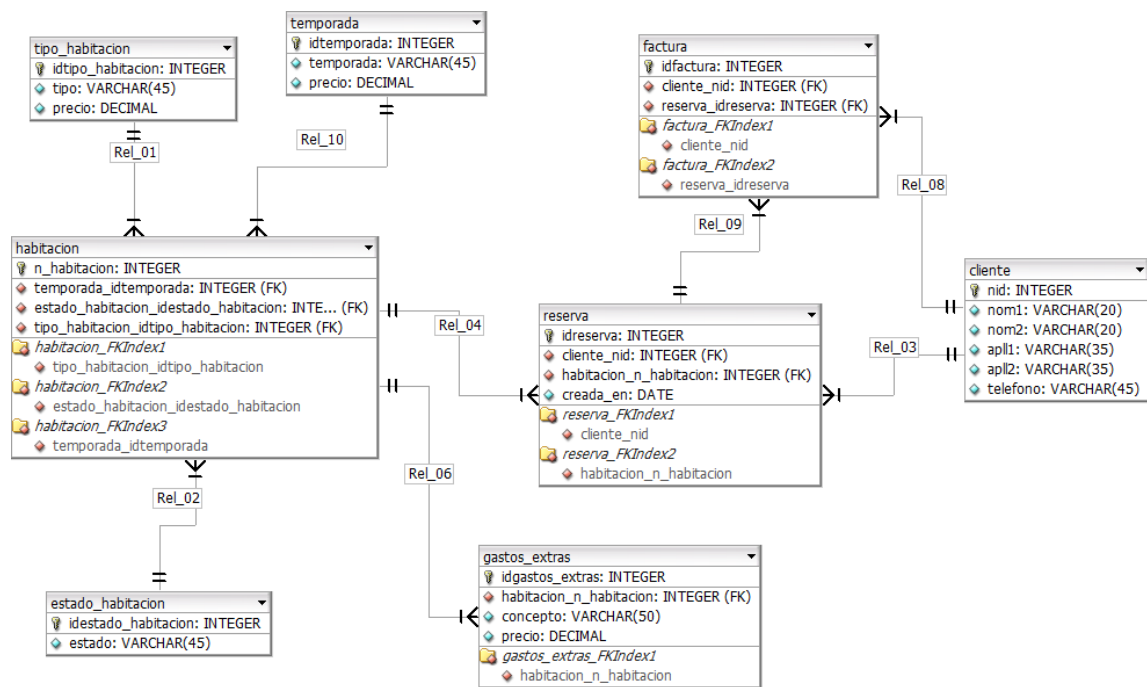
### Gastos extras

Los gastos extras que se van a poder cargar a la habitación son de naturaleza variada, y se denotará por un concepto y valor.

## Diagrama ER



## Diagrama Relacional



# Poblando Base de datos:

## MYSQL

The screenshot shows the phpMyAdmin interface. On the left, the database structure tree is visible, with 'ejercicio7' selected. A red arrow points to the 'ejercicio7' folder. The main panel displays the 'Estructura' (Structure) tab for the 'ejercicio7' database. It shows four tables: 'tipo\_habitacion', 'temporada', 'cliente', and 'estado\_habitacion'. Each table's structure is displayed, including its columns, data types, and primary keys. The 'tipo\_habitacion' table has columns 'idtipo\_habitacion' (INTEGER UNSIGNED NOT NULL AUTO\_INCREMENT), 'tipo' (VARCHAR(45) NOT NULL), and 'precio' (DECIMAL NOT NULL). The 'temporada' table has columns 'idtemporada' (INTEGER UNSIGNED NOT NULL AUTO\_INCREMENT), 'temporada' (VARCHAR(45) NOT NULL), and 'precio' (DECIMAL NOT NULL). The 'cliente' table has columns 'nid' (INTEGER UNSIGNED NOT NULL), 'nom1' (VARCHAR(20) NOT NULL), 'nom2' (VARCHAR(20) NOT NULL), 'apl1' (VARCHAR(35) NOT NULL), 'apl2' (VARCHAR(35) NOT NULL), 'telefono' (VARCHAR(45) NOT NULL), and 'PRIMARY KEY(nid)'. The 'estado\_habitacion' table has columns 'idestado\_habitacion' (INTEGER UNSIGNED NOT NULL AUTO\_INCREMENT), 'estado' (VARCHAR(45) NOT NULL), and 'PRIMARY KEY(idestado\_habitacion)'. The console at the bottom shows the execution of SQL commands for creating these tables.

The screenshot shows the phpMyAdmin interface with the 'Consola' (Console) tab selected. It displays the execution of SQL commands for inserting data into the 'cliente' and 'tipo\_habitacion' tables. The console shows five successful insertions into the 'cliente' table and one successful insertion into the 'tipo\_habitacion' table. Each insertion is followed by a confirmation message: '1 fila insertada. (La consulta tardó 0,0020 segundos.)' for the first client, '1 fila insertada. (La consulta tardó 0,0018 segundos.)' for the second, '1 fila insertada. (La consulta tardó 0,0019 segundos.)' for the third, '1 fila insertada. (La consulta tardó 0,0021 segundos.)' for the fourth, and '1 fila insertada. (La consulta tardó 0,0024 segundos.)' for the fifth. The SQL commands are as follows:

```
INSERT INTO cliente (nid,nom1,nom2,apl1,apl2,telefono) VALUES (13265,'Juan','Perez','Perez','Perez','123456789');
INSERT INTO cliente (nid,nom1,nom2,apl1,apl2,telefono) VALUES (13266,'Pedro','Juan','saens','Perez','1472583692');
INSERT INTO cliente (nid,nom1,nom2,apl1,apl2,telefono) VALUES (13267,'Yan','Fran','Arias','Bohorques','963852741');
INSERT INTO cliente (nid,nom1,nom2,apl1,apl2,telefono) VALUES (13268,'Jose','Alejandro','Antonio','Arias','456123789');
INSERT INTO tipo_habitacion (idtipo_habitacion, tipo, precio) VALUES (1, 'Individual', 100);
```



## PSQL

```
ejercicio7=# \dt
```

### List of relations

Schema	Name	Type	Owner
public	cliente	table	postgres
public	estado_habitacion	table	postgres
public	factura	table	postgres
public	gastos_extras	table	postgres
public	habitacion	table	postgres
public	reserva	table	postgres
public	temporada	table	postgres
public	tipo_habitacion	table	postgres

(8 rows)

```
ejercicio7=# \i ./Actividad001/7/data.sql
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```

```
INSERT 0 1
```